SIT225 Data Capture Technologies

Pass Task: Capture smartphone sensor data

Overview

The Arduino Cloud is a powerful platform that can help you with setting up your own IoT devices within minutes. It is now possible to synchronize your phone's sensor data with the Cloud, using the Arduino IoT Remote App for iOS and Android. Data can be shown in Arduino Cloud dashboard, collected and stored for further analysis.

Hardware Required

i. A compatible smartphone – Android or iPhone.

Software Required

i. Arduino IoT Cloud
ii. Python 3,
iii. Plotly Python library,
iv. Numpy and Pandas Python libary

Pre-requisites: You must do the following before this task

Week 8 activities in the unit site.

Task Objective

In this week, you have learned how to use smartphone's sensor data, create dashboard to visualise data and download it manually for analysis. You have also learned how to create custom Python device to connect to Arduino cloud to synchronise with any cloud variable to receive fresh data from Arduino cloud. You will need to continuously collect smartphone accelerometer readings in Python device (Python script) and create visualisation using Plotly Dash and update the graphics with fresh data, when available.

Steps:

1. Setup Arduino IoT Remote App and Arduino Cloud to receive phone's accelerometer data in Arduino dashboard using variables x, y, and z.
2. Create custom Python device to connect to Arduino Cloud and create its own variables py_x, py_y, and py_z. There will be no things attached to Python, but its variables will be used to link to other Arduino Cloud variables.

3. Synchronise Python variables py_x, py_y, and py_z to phone's corresponding variables x, y, and z. Set it up to receive fresh accelerometer readings from your phone in Python script.
4. Continuous data received in Python script can be stored in lists to buffer and once enough number of readings are gathered (say N number of samples where N equals to 1000 or more, as you choose right), use this data to show in Plotly Dash using suitable graphs.
5. As new data is pouring in, the lists will get filled in, once N number of new samples arrive, update the Plotly Dash to refresh the displayed graphics.
6. The data being received in Python script in lists needs to be managed so that the data used to display a graph should be isolated from the newly received data. Hint: You can maintain 2 lists, one to receive data continuously and the other to copy N samples from the first one to draw graphs. N samples should be moved from the first buffer to the 2nd buffer to avoid drawing the same graph next time.
7. Just before updating Plotly Dash graphs, save the plot with filename using appropriate timestamp, you can also save data corresponding to the plot into a CSV file. Make sure the graph contains enough samples (say 10 seconds samples) to capture a single activity. You should perform at least 2 distinct activities using your phone that are reflected in your Plotly graphs. Repeat these activities multiple times so you can validate each activity with more graphs available.
8. Analyse the graphs of distinct activities you have created and saved in step 7 to find patterns among variables x, y, and z. Justify your answer. You can use corresponding graph's CSV data to calculate statistical measures to support your claim.

Submission details

Q1. Perform week 8 activities mentioned in the unit site and produce outputs.

Q2. Summarise your analysis of accelerometer activity separation describing your method, results and discussion.

Q3. Create a video in Panopto/CloudDeakin showing your program execution, continuous graph updates, and share the video link here.

Q4. Create a subdirectory 'week-8' under directory 'SIT225_2024T2' in your drive where you copy the Python script file which contains Plotly Dash functions, Arduino sketch file if any, data file and the generated graphs. Commit and push to changes to GitHub. Include the link to your repository here with a GitHub page screenshot of weekly folder content. A tutor may try to access your GitHub link, if necessary. Give access to your tutor by adding tutor's email address as a collaborator of your private repository.

Instructions

Consolidate outputs following the submission details above into a single PDF file.

Submit your work

When you are ready, login to OnTrack and submit your pdf which consolidates all the items mentioned in the submission detail section above. Remember to save and backup your work.

Complete your work

After your submission, your OnTrack reviewer (tutor) will review your submission and give you feedback in about 5 business days. Your reviewer may further ask you some questions on the weekly topics and/or about your submissions. You are required to address your OnTrack reviewer’s questions as a form of task discussions. Please frequently login to OnTrack for the task ***Discuss/Demonstrate*** or ***Resubmit*** equivalent to fix your work (if needed) based on the feedback to get your task signed as ***Complete***.