SIT225 Data Capture Technologies

Pass Task: Store data to cloud

Overview

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. Data is stored as JSON and synchronized in real-time to every connected client.

Hardware Required

i. Arduino Nano 33 IoT device,
ii. USB cable,
iii. LSM6DS3 module on the Arduino Nano 33 IoT for Gyroscope data.

Software Required

Python 3.

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

Week 5 activities in the unit site.

Task Objective

In this week, you have learned how to use Firebase real-time database and perform operations such as add/update/query data and listen to live changes to your data. In this task, you will need to record Gyroscope readings from the built-in module of Arduino Nano board and pass data to a Python script through Serial communication (or through Arduino IoT Cloud variable synchronisation with Python). The Python script upload data to Firebase real-time database as soon as it receives any. After enough data is stored to capture interesting patterns, for no less than 10 minutes, query the database for data, format it to save in a CSV file and analyse the data. *You may reuse this data next week, so keep your CSV file handy*.

Steps:

1. Connect the sensor for collecting data samples no less than 30 minutes. Justify your data collection decision in terms of what, when and how long to capture with respect to the phenomena you want to capture with Gyroscope sensor.
2. Find and report the data rate of the Gyroscope module and decide how frequently you want to sample data from it. Justify your answer.

3. Consider the number of data variables per sensor, since you will need to transmit them through the serial communication port (or alternatively, through Arduino IoT Cloud). The number of variables for LSM6DS3 Gyroscope is 3 (x, y, z).
4. Write a Python script to continuously run and perform the following actions:
    a. Receive data following the protocol how data is sent from the Arduino program,
    b. Record your computer's current timestamp and prepare a JSON message with timestamp and x, y, z and add this data sample to Firebase.
    c. Repeat step 4 as long as Arduino is sending data, stop both processes once data collection is done.
5. Now query Firebase to receive all data you have sent so far, process the JSON data array to format each sample as CSV and store in a CSV file.
6. Try to clean data variables in the CSV file which includes removing any non-number or empty fields, if exists. This step can be a manual process, or you can use Python tools such as Numpy or Pandas, if you want to. Justify your action.
7. Read the CSV file, plot x, y, z variables separately as 3 line-graphs and a graph with all 3 variables in a single graph.
8. Observe the data changing pattern of the single variable plot and combined plot and try to come up with your comments. Following items are subject to observation, but not limited to, -
    a. Any repeating pattern,
    b. Up and down trends in data variables, or
    c. Relative changes of Gyroscope variables x, y or z.
    d. You should refer to Gyroscope variable semantics in the Arduino document (https://docs.arduino.cc/tutorials/nano-33-iot/imu-gyroscope).

Submission details

Q1. Perform week 5 activities mentioned in the unit site and produce outputs.

Q2. State the hypothesis you can think out of your data. Show the graph created from the sensor data, analyse it and describe if there are any interesting patterns you can observe. Justify if your hypothesis holds, at what level; if not, then what might be the reason?

Q3. Paste Python and Arduino sketch and explain program steps.

Q4. Create a video in Panopto/CloudDeakin showing your program execution, data collection, data upload to Firebase and graph output, share the video link here.

Q5. Create a subdirectory 'week-5' under directory 'SIT225_2024T2' in your drive where you copy the Python script file, Arduino sketch file, 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***.