SIT225 Data Capture Technologies

Pass Task: Working with sensors

Overview

Arduino can connect sensors and read data from them. Often data is stored either in a file in memory card if Arduino has any or sent to cloud storage. You are lucky to have your computer connected to Arduino which enables serial communication using USB. Arduino can write any data through serial port which can be received in computer, you will be using a Python program to receive data and keep storing in a file in your computer to do exciting analysis afterwards.

Hardware Required

i. Arduino Nano 33 IoT device,
ii. USB cable,
iii. Any of the sensors including DHT22 Temperature and Humidity sensor, HC-SR04 Ultrasonic sensor or LSM6DS3 module on the Arduino Nano 33 IoT for accelerometer data.

Software Required

i. Python 3

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

Week 2 activities in the unit site.

Task Objective

In this week, you have learned to use sensors including DHT22, HC-SR04 and LSM6DS3. In this task, you will collect data from any of these sensors for a long time to gather enough data for analysis, 30 minutes or longer, if possible, overnight for DHT22, and continuously store them in a CSV file in your computer with timestamps of each data sample. Once data is gathered, you will need to plot the data in graphs and analyse them. Keep in mind that capturing interesting patterns in data is one of the main goals of this task, so you should plan what to capture and when to capture.

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.

2. Decide the sampling frequency of the data collection. For example, accelerometer data requires high sampling frequency (such as a single sample each second) while other sensors can be sampled at a slower interval (10- or 30-seconds interval). You choose data sampling frequency as reasonable and justify your decision.
3. Consider the number of data variables per sensor, since you will need to transmit them through the serial communication port. The number of variables for DHT22, HC-SR04 and LSM6DS3 accelerometer sensors corresponds to 2 variables (temperature and humidity), 1 variable (distance) and 3 variables (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 store each data sample in a sensor specific filename as comma separated values, data format: ***<timestamp>,<comma separated data items>,*** where timestamp format is ***YearMonthDayHourMinuteSecond*** *(such as 20240517210810 for year 2024, month 05, day 17, hour 21, minute 08 and second 10)*. Append the next data point in a separate line in the same file, so the file grows bigger as new data comes in.
    c. Repeat step 4 as long as Arduino is sending data, stop both processes once data collection is done.
5. Read each CSV file, plot each sensor data separately using Python script. You can use Jupyter notebook for this task. Data plots for either of the sensors can be as follows -
    a. LSM6DS3: Accelerometer data plot x, y, z variables into a single figure to see combined variation,
    b. DHT22: Temperature and humidity variables into a single figure to see combined variation, or
    c. HC-SR04: The distance is a single variable to plot.
6. Try to clean data variables in the CSV file which includes removing any non-number or empty fields, if exists. At this point data cleaning is a manual process, you will be learning how to do it using tools (such as Numpy and Pandas) in the following weeks.
7. Observe the data changing pattern of the sensor 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 variables (such as x, y or z variables in accelerometer data)

Submission details

Q1. Perform week 2 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 and any instruction to be followed in order to run your code, share the video link here.

Q5. Create a subdirectory 'week-2' under directory 'SIT225_2024T2', which you created for week 1 task, 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***.