SIT225 Data Capture Technologies

Credit Task: Trigger an alarm in Arduino IoT Cloud

Overview

Arduino IoT Cloud dashboard is a great way to monitor your sensor data in real-time. It has several UI widgets for data visualisation including Gauge to show data values graphically and LED to turn ON/OFF based on some event. Widgets like LED can be used to trigger an alarm if sensor data is outside a normal range.

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. Arduino IoT Cloud,
ii. Arduino Web IDE

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

Week 3 activities in the unit site and 3.1P task.

Task Objective

You can continue with the setting of 3.1P to stream sensor data to Arduino IoT Cloud and display in cloud dashboard. You will need to process real-time sensor data and assume a minimum or maximum threshold value beyond which any value you consider as a danger value and you want to trigger an alarm to be visible in Arduino Cloud dashboard.

Steps:

1. Setup Arduino Cloud to add Arduino Nano 33 IoT device and Things which are your sensors. This requires you to connect Arduino board using Wi-Fi where you can share your smartphone's Wi-Fi to Arduino as a hotspot.

2. Connect a sensor to Arduino board and keep streaming data to Arduino Cloud. Justify selecting the sensor among available sensors for this task where generation of alarm is important.
3. Identify how you would like to display an alarm and what widgets in Arduino dashboard are suitable for displaying alarm. Come up with at least 2 ways of generating alarm, create the widget and necessary variable, if needs to be linked.
4. State a hypothesis where sensor values can produce extreme values and try to create that environment. For example, for Ultrasonic sensor, a very close distance value, or for accelerometer sensor, a continuous shaking for few seconds can simulate a situation which is ideal case for generating alarm. Justify your own hypothesis of emergency situations and how you want to observe it in data.
5. You will need to add logic in your Arduino code to sense alarming conditions from sensor data. It may require storing several data readings to sense alarming conditions, such as for accelerometer data, few seconds of x, y and z values need to be stored for processing. Describe how you have implemented emergency condition sensing logic.
6. Keep monitoring fresh sensor readings and once an alarming condition is detected, trigger the variable which is linked to alarm widget in Arduino cloud dashboard. Keep the alarm variable set indefinitely until reset manually from the dashboard, if it has option to do so.
7. Repeat step 6 for an alternative alarm widget.

Submission details

Q1. Perform week 3 activities mentioned in the unit site and produce outputs.

Q2. State the hypothesis you can think of would be an ideal situation to be considered as an alarming condition. Describe how to sense the condition in freshly read sensor data. You should include any other justification required for this task such as sensor selection mentioned in step 2.

Q3. Justify your selection of 2 Arduino cloud dashboard widgets for alarm visualisation among other available widgets. How you triggered the widget to enable once an emergency was detected in sensor data.

Q4. Create a video in Panopto/Cloud Deakin showing your Arduino Cloud Dashboard with your Thing's data variables being updated, alarm is generated and share the video link here.

Q5. Create a subdirectory 'week-3.2C' 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***.