# Constrained Device Application (Connected Devices)

## Lab Module 02

Be sure to implement all the PIOT-CDA-\* issues (requirements) listed at [PIOT-INF-02-001 - Lab Module 02](https://github.com/orgs/programming-the-iot/projects/1#column-9974938).

### Description

NOTE: Include two full paragraphs describing your implementation approach by answering the questions listed below.

What does your implementation do?

The implementation involves creating a Python application called ConstrainedDeviceApp, which is designed to monitor key system performance metrics, specifically CPU and memory usage. The application is structured in a modular way, making it easier to maintain and extend. Here are the main components:

\* ConstrainedDeviceApp: This is the main application that brings together various tasks for monitoring system performance.

\* SystemPerformanceManager: This class oversees the system performance tasks, scheduling them to run at regular intervals.

\* BaseSystemUtilTask: This acts as a base class for all system utilization tasks, providing shared functionality.

\* SystemCpuUtilTask: This subclass of BaseSystemUtilTask is dedicated to retrieving CPU usage metrics.

\* SystemMemUtilTask: Another subclass of BaseSystemUtilTask, this one focuses on monitoring memory usage.

The SystemPerformanceManager is responsible for creating instances of both the CPU and memory tasks and scheduling them using the APScheduler library.

How does your implementation work?

The application is organized into two main package directories: the main application is located in ./programmingtheiot/cda/app, while the system management modules are found in ./programmingtheiot/cda/system.

Integration of SystemPerformanceManager

Inside the ConstrainedDeviceApp, we create an instance of SystemPerformanceManager. This instance is responsible for starting and stopping the monitoring of system performance metrics.

Task Management

The SystemPerformanceManager holds instances of SystemCpuUtilTask and SystemMemUtilTask. It uses the APScheduler library to run these tasks at set intervals, allowing for continuous monitoring without interrupting the main application.

Scheduled Tasks

Each taskùone for CPU utilization and one for memory utilizationùretrieves the relevant system metrics when triggered by the scheduler. The results can then be logged or processed as needed.

Base Class for Common Functionality

The BaseSystemUtilTask serves as a base class that provides shared methods and properties for both the CPU and memory tasks. This helps ensure consistency and reduces code duplication.

Start/Stop Methods

The ConstrainedDeviceApp controls the lifecycle of the SystemPerformanceManager, using its start method to begin monitoring and its stop method to terminate the monitoring process gracefully.

Summary

Overall, this implementation offers a robust and scalable solution for monitoring system performance metrics in a constrained device environment. It leverages existing libraries and follows a clear modular architecture, making it easy to maintain and extend.

### Code Repository and Branch

NOTE: Be sure to include the branch (e.g. https://github.com/programming-the-iot/python-components/tree/alpha001).

URL: https://github.com/Francistapiwa/python-components

### UML Design Diagram(s)

A screenshot of a computer

Description automatically generated

### Unit Tests Executed

NOTE: TA's will execute your unit tests. You only need to list each test case below

(e.g. ConfigUtilTest, DataUtilTest, etc). Be sure to include all previous tests, too,

since you need to ensure you haven't introduced regressions.

- ConfigUtiltest

- SystemCpuUtiltask

- SystemMemUtiltask

### Integration Tests Executed

NOTE: TA's will execute most of your integration tests using their own environment, with

some exceptions (such as your cloud connectivity tests). In such cases, they'll review

your code to ensure it's correct. As for the tests you execute, you only need to list each

test case below (e.g. SensorSimAdapterManagerTest, DeviceDataManagerTest, etc.)

- ConstrainedDeviceApp

- SystemPerformanceManager

EOF.