# 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 defines a Python application called ConstrainedDeviceApp, aimed at monitoring system performance, particularly focusing on CPU and memory usage. The design uses a modular approach by organizing the functionality into separate classes, improving maintainability and clarity. The key components include:

\* ConstrainedDeviceApp: The core application that brings together different system performance monitoring tasks.

\* SystemPerformanceManager: Manages system performance tasks and ensures they are executed at regular intervals.

\* BaseSystemUtilTask: A base class that provides shared functionality for all system utilization tasks.

\* SystemCpuUtilTask: A subclass of BaseSystemUtilTask, dedicated to gathering CPU usage data.

\* SystemMemUtilTask: Another subclass of BaseSystemUtilTask, responsible for tracking memory usage.

The SystemPerformanceManager coordinates the creation and scheduling of CPU and memory tasks using the APScheduler library.

How does your implementation work?

Application Structure:

The application is organized into a package hierarchy, with the main application components located in ./programmingtheiot/cda/app and system management modules under ./programmingtheiot/cda/system.

Integration of SystemPerformanceManager:

Within the ConstrainedDeviceApp, an instance of the SystemPerformanceManager is created. This instance is responsible for managing the start and stop operations for monitoring system metrics.

Task Management:

The SystemPerformanceManager contains instances of SystemCpuUtilTask and SystemMemUtilTask. It uses the APScheduler library to schedule these tasks at specified intervals, ensuring continuous system monitoring without interrupting the main application workflow.

Scheduled Tasks:

When triggered by the scheduler, each task (CPU and memory) collects the relevant system metrics. The collected data can then be logged or processed according to the application’s needs.

Base Class for Common Functionality:

The BaseSystemUtilTask provides common methods and properties for both CPU and memory tasks, allowing for consistency across tasks and minimizing code duplication.

Start/Stop Methods:

The ConstrainedDeviceApp manages the lifecycle of the SystemPerformanceManager, calling its start method to initiate monitoring and its stop method to gracefully shut down the monitoring process.

Overall, this implementation offers a scalable and efficient solution for monitoring system performance metrics in a constrained device environment, using a modular architecture and leveraging existing libraries.

### Code Repository and Branch

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

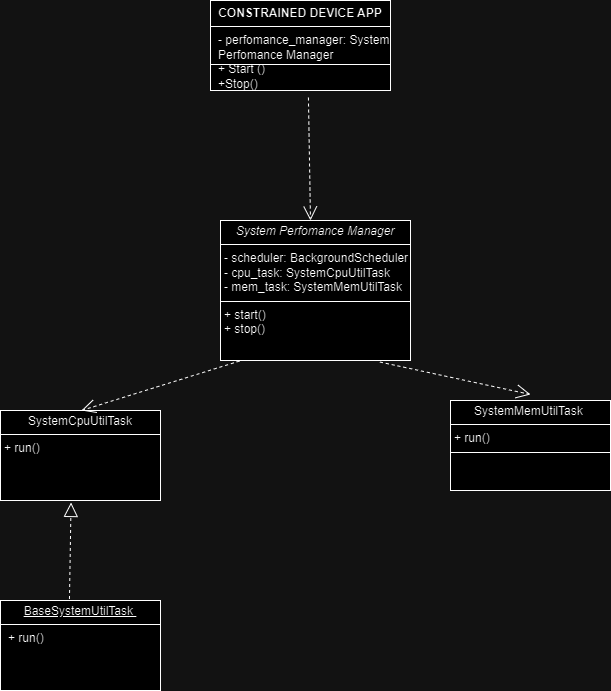
URL: [Elikem161/py-components (github.com)](https://github.com/Elikem161/py-components)

### UML Design Diagram(s)

NOTE: Include one or more UML designs representing your solution. It's expected each

diagram you provide will look similar to, but not the same as, its counterpart in the

book [Programming the IoT](https://learning.oreilly.com/library/view/programming-the-internet/9781492081401/).



### 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.