IoT Developer Assignment

Estimate time: 6h

Objective

Build a small Python-based Device Agent that simulates core features of an edge/IoT runtime. This challenge evaluates async I/O, MQTT messaging, configuration reload, and robust service design.

Instructions

Load `device.json` at startup:

```
{
  "device_id": "pv-sim-001",
  "power": 1000
}
```

Every 5 seconds generate a simulated measurement by varying the power ±5%.

Example:

```
{ "timestamp": "2025-01-01T12:00:00Z", "power": 1032.4 }
```

Log errors if the file is missing or malformed.

MQTT publishing

Publish each generated PV message to MQTT under topic:

`device/{device_id}/state`

Payload example:

```
{ "timestamp": "2025-01-01T12:00:00Z", "power": 982.3 }
```

Heartbeat & LWT

Send a heartbeat every 30 seconds and implement Last Will (offline) messages.

Topic: `device/{device_id}/status`

Payload:

```
{"status": "online"} or {"status": "offline"}
```

Configuration File (config.yaml)

Load config such as MQTT host/port and poll_interval. Detect and apply changes without restart. Handle malformed config gracefully.

Robustness & Shutdown

Use structured logging, handle exceptions, disconnect cleanly from MQTT, support Ctrl+C graceful shutdown.

Deliverables

The finished project should be delivered in the form of a publicly available GIT repository on GitHub or other similar tools. The project should include the documentation and the required instruction for running the application.

- Source code
- README with instructions and design notes
- Dockerfile
- Docker compose for local development that includes all relevant dependecies

Evaluation criteria

Code quality and readability

- Async implementation
- MQTT correctness
- Error handling
- Config reload reliability
- Documentation clarity