IoT Device Configuration Guide for Manufacturers

Overview

This guide explains how to create a YAML configuration file that describes your IoT device's capabilities and interfaces. Once validated and uploaded to our manufacturer portal, your device will be supported by our plant monitoring application.

Configuration Structure

Device Information (Required)

```yaml
device_info:
  manufacturer: string    # Your company name
  model: string          # Model name/number of the device
  firmware_version: string # Current firmware version
  description: string    # (Optional) Brief description of device functionality
```

Network Configuration (Optional)

```yaml
network:
  default_port: 5683     # CoAP port, defaults to 5683 if omitted
  discovery_enabled: boolean # Defaults to true if omitted
```

Sensors Configuration

At least one sensor or actuator is required. Sensors are read-only components that report numerical values.

```yaml
sensors:
  - name: string         # Descriptive name (e.g., "light_sensor")
    type: string        # Recommended sensor type (e.g., "LIGHT", "HUMIDITY")
    measurement:
      unit: string      # Measurement unit (e.g., "lux", "%")
      data_type: enum   # "INTEGER" or "FLOAT"
      range:            # Optional
        min: number     # Minimum possible value
        max: number     # Maximum possible value
    coap_endpoints:
      read:
        path: string    # CoAP endpoint path (e.g., "/sensors/light")
        content_format: string  # Recommended: "application/json"
        value_key: string      # JSON key for the sensor value
    sampling:           # Optional
      interval: number  # Sampling frequency in seconds (default: 60)
```

Actuators Configuration

Actuators are controllable components with changeable states. Supported types: PUMP, LIGHT, BLIND.

```
actuators:
  - type: enum          # "PUMP", "LIGHT", or "BLIND"
    name: string        # Descriptive name (e.g., "water_pump")
    endpoints:
      status: string    # Status query endpoint
      # For PUMP and LIGHT:
      turn_on: string   # Endpoint to activate
      turn_off: string  # Endpoint to deactivate
      # For BLIND:
      open: string      # Endpoint to open
      close: string     # Endpoint to close
    state:
      value_key: string # JSON key for state value in response
      data_type: enum   # "BOOLEAN" or "ENUM"
      possible_values: [string]  # List of possible state values
      # Optional state mappings:
      on_value: string  # For PUMP/LIGHT: custom "on" value
      off_value: string # For PUMP/LIGHT: custom "off" value
      open_value: string # For BLIND: custom "open" value
      closed_value: string # For BLIND: custom "closed" value
```

Requirements and Validation Rules

General Requirements

- File must be valid YAML format
- At least one sensor or actuator must be defined
- All required fields must be present
- All endpoints must be valid URL paths
- All strings must be non-empty

Sensor-specific Rules

- Measurement units must be specified
- Data type must be either INTEGER or FLOAT
- If range is specified, min must be less than max
- CoAP read endpoint must be specified

Actuator-specific Rules

- Type must be one of: PUMP, LIGHT, or BLIND
- All required endpoints must be specified based on type
- State configuration must be valid for the actuator type

- If using ENUM data type with more than 2 values, state mappings are required

Example Configuration

```yaml
device_info:
  manufacturer: "GreenTech Solutions"
  model: "Plant Care Basic"
  description: "Basic plant monitoring and care station"
  firmware_version: "1.0.3"

network:
  default_port: 5683
  discovery_enabled: true

sensors:
  - name: "light_sensor"
    type: "LIGHT"
    measurement:
      unit: "lux"
      data_type: "INTEGER"
      range:
        min: 0
        max: 100000
    coap_endpoints:
      read:
        path: "/sensors/light"
        content_format: "application/json"
        value_key: "light"
    sampling:
      interval: 300

actuators:
  - type: "PUMP"
    name: "water_pump"
    endpoints:
      turn_on: "/actuators/pump/on"
      turn_off: "/actuators/pump/off"
      status: "/actuators/pump"
    state:
      data_type: "BOOLEAN"
      possible_values: ["on", "off"]
      value_key: "status"
```

CoAP Requirements

- All devices must implement CoAP server functionality
- Endpoints must respond to appropriate CoAP methods:
  - Sensor endpoints: GET
  - Actuator status endpoints: GET
  - Actuator control endpoints: PUT
- All responses should be in JSON format
- Status codes should follow CoAP conventions

Best Practices

1. Use descriptive names for sensors and actuators
2. Include meaningful descriptions
3. Specify ranges for sensor values when applicable
4. Use BOOLEAN data type for simple on/off states
5. Use consistent naming conventions for endpoints
6. Include all optional fields that are relevant
7. Document any device-specific behavior in the description
8. Use appropriate sampling intervals based on the sensor type

Validation Process

1. Upload YAML file to manufacturer portal
2. System validates format and requirements
3. If validation fails, detailed error messages are provided
4. After successful validation, device type becomes available
5. End users can create instances of the device

For technical support or questions about the configuration process, please contact our manufacturer support team.