Chapter 8: Class Project

Objective

Time: 3 Hours

Fundamentals

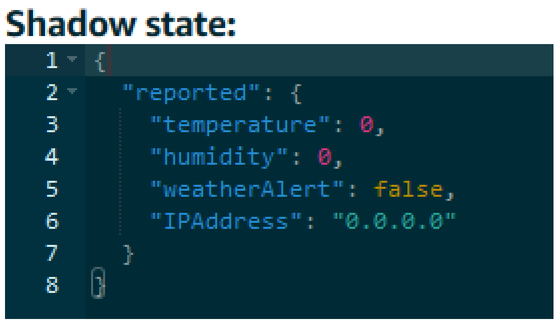
Exercise(s)

Your project is to build an IoT weather station. It will:

1. Measure local temperature and humidity. This information can be read from the analog co-processor shield kit using I2C (see I2C exercises in the peripherals chapter).
2. Display the temperature and humidity on the OLED screen.
3. Connect to a provided MQTT broker:

   amk6m51qrxr2u.iot.us-east-1.amazonaws.com

4. Your *thing* name will be "ww101_<nn>" where <nn> will be a number assigned to you. For example ww101_01.
5. The credential and private key for your *thing* can be found in the class material folder.
   a. Hint: After updating the key files, you should run a "Clean" on the project. Otherwise, the project will not see the new keys.
6. Update the state of the *thing*. The parameters are named "temperature" (float), "humidity" (float), "weatherAlert" (true or false) and "IPAddress" (ipv4 4dot syntax).
   a. Hint: The starting (empty) shadow for your *thing* will look like the following. You will publish JSON messages to the *thing* shadow to provide updates.



Shadow state:

```
{
  "reported": {
    "temperature": 0,
    "humidity": 0,
    "weatherAlert": false,
    "IPAddress": "0.0.0.0"
  }
}
```

7. Implement a serial terminal to allow the following commands (see UART exercises in the peripherals chapter):

    t – read + publish temperature
    h – read + publish humidity
    A – publish weather alert on
    a – publish weather alert off
    S – turn on subscriptions
    s – turn off subscriptions
    P – turn on printing of updates
    p – turn off printing of updates
    x – print the current known state of data
    l – print the list of known things
    c – clear the screen
    ? – print out a help screen
    u – turn off auto updating
    U – turn on auto updating

   For subscriptions, it is easiest to just maintain a list of all of the *things* that have been assigned for the class (i.e. ww101_01, ww101_02, etc.)

It would be cool if you:

1. Used the linked_list library to maintain a local database
2. Used threads
3. Used the console library functions to build the interface
4. Used VT100 escape codes to make a pretty screen (http://ascii-table.com/ansi-escape-sequences-vt-100.php)
5. Used the DCT to write the configuration
6. Created an HTTP server to display all of the information