Instructions to follow to make sure that Atlas Thing Middleware works on your Raspberry Pi.

Firstly, I am going to copy and paste the exact instructions that are given in the GitHub repo of the Atlas Thing Middleware(https://github.com/AtlasFramework/AtlasThingMiddleware_RPI), for reference.

Prepare your Atlas smart thing on Raspberry Pi through the following steps:

Step1: run the following linux commands through terminal:

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt-get install gcc-6 g++-6 build-essential //usually install the latest gcc and g++
- sudo apt-get install doxygen
- sudo apt-get install cmake cmake-curses-gui
- sudo apt-get install libboost-all-dev
- sudo apt-get install curl libcurl4-openssl-dev
- sudo apt-get install autogen

Step2: Get the latest version of the middleware:

From Github, download the zip version of the middleware on your RaspberryPi, then unzip the folder.

Step3: Install cppMicroservices library

- unzip the folder named CppMicroServices-development under Atlas-IoT_Thing/lib/ of the middleware, and keep in the lib directory
- cmake CppMicroServices-development/
- sudo make
- sudo make install
- LD_LIBRARY_PATH=/usr/local/include
- export LD_LIBRARY_PATH

- sudo ldconfig
- //the new version installs the library in /usr/local/include/cppmicroservices4/ rather than /usr/local/include/
- sudo mv /usr/local/include/cppmicroservices4/ ~/Desktop/
- sudo mv ~/Desktop/cppmicroservices4/cppmicroservices/ /usr/local/include/

Step4: Install WiringPi library and enable the hardware interfaces

- unzip the folder named WiringPi-master under Atlas-IoT_Thing/lib/WiringPi-master/ of the middleware
- cd to the WiringPi-master folder
- ./build
- sudo apt update
- sudo apt upgrade
- sudo apt install rpi.gpio
- sudo raspi-config
- under “Interfacing Options”, enable both I2C and SPI

Step5: Compile and Build Atlas middleware

- Navigate to the directory of Atlas-IoT_thing (use cd command) and Compile as follows:
- cmake Main/
- make

Step6: Add an IoT-DDL

- Use this builder tool to build an IoT-DDL file for your Atlas thing.
- Navigate to the directory of Atlas-IoT_thing and add the genenerated IoT-DDL.xml file to the /ConfigurationFiles directory (replace the default file)

Step7: Run Atlas middleware

- Through therminal, and under the directlory of the middleware, run the following command:
- ./Atlas

Due to differences between the Raspberry Pi version used to develop/test the Atlas Thing Middleware and the versions most students are using, some changes to the above instructions are necessary.

Note: If anyone is using an older version of the Raspberry Pi (Raspberry Pi 3+), the only edit they need to make is the editing of the service.tpl file as mentioned in the Step 6 below. This needs to be done only if they want to use pigpio. If WiringPi works on their Raspberry Pi, no need to make this change.

These are the changes:

1. Follow the above step 1 and step 2 as is. There is no issue in those steps.
2. In step 3, after unzipping the folder and after running the command "cmake CppMicroServices-development/", download the patch that has been provided to you anywhere on your Rapsberry Pi.
3. Now, run the following command in the terminal of your Raspberry Pi "patch -p1 < <absolute_path_of_patch_file>"
    a. Eg: The absolute path of the patch in my Raspberry Pi was '/home/subhash/Downloads/required_patch.patch'. The command that I used was "patch -p1 < /home/subhash/Downloads/required_patch.patch"
4. After this, complete the rest of the Step 3 provided in the GitHub Readme as is.
5. If you are using WiringPi to connect to your Raspberry Pi, follow the exact same steps given in step 4 of the GitHub Readme file.
6. If you are using pigpio, follow these steps:
    a. Make sure to install the pigpio on your Raspberry Pi. To do this, run the following command "sudo apt install libpigpiod-if-dev"
    b. Now, open the service template file that is available at the following path "/AtlasThingMiddleware_RPI-master/Architecture/GeneratedServices/ServiceTemplate/service.tpl" and replace the "#include <wiringpi.h>" with "#include <pigpio.h>". Save the changes.
7. Now, before you execute the step 5 in the GitHub Readme file, make the following changes:

a. Navigate to the following file "/AtlasThingMiddleware_RPI-master/lib/PahoMQTT/Log.h" and go to line number 43 and delete the text "Log_levels". After you do this, your code block starting from line 35 should look like shown below and save the changes:

```
enum LOG_LEVELS {
        TRACE_MAXIMUM = 1,
        TRACE_MEDIUM,
        TRACE_MINIMUM,
        TRACE_PROTOCOL,
        LOG_ERROR,
        LOG_SEVERE,
        LOG_FATAL,
};
```

i.

8. Note: Make sure to save the edited files before you execute any later commands.
9. Now, you can proceed with steps 5, 6 and 7 given in the GitHub Readme file.