OpenSensorHub

Development Training
Driver Development: Creating Driver Framework and Adding to Build

Lab 3 – 75 minutes
Requirements

• Java Programming Language – Entry Level Experience

• Lab 2 Complete!
Opening Existing Project

Can skip to next section if project already opened...
Opening existing project
Project Desktop
Create Sensor Module
From sensorhub-driver-template
Copy Driver Template
Copy Driver Template...

1. (right mouse button click)
Paste Template Back Into Sensors Directory

1. (right mouse button click)
Name Module

1. Assign new name
2. Click OK button
Successfully Created Module

2. Add files if to be managed by git, cancel otherwise
Current Project State

Note: Files (red) not added to git
Updating Package Name
Navigate to Driver Package
Refactor Package Name

1. Right click on package
2. Select Rename
1. Provide a new name
2. Select Refactor
Refactored Package Name
Updating Driver Build Scripts

Name and Description
Editing Driver build.gradle

Open sensorhub-driver-simulation/build.gradle
1. Edit description – this field contains the name that will be assigned to your driver and visible in OpenSensorHub “Simulated Sensor Driver”

2. Edit the osgi->manifest attribute for the ‘Bundle-Activator’ property to reflect the name of the package to com.sample.impl.sensor.simulated.Activator
About OSGi and the Activator Module

OSGi stands for **Open Service Gateway Initiative**, which is a Java framework for developing and deploying modular software programs and libraries. Although the labs do not cover OSGi Bundles, OSH does support this functionality. OSH allows for additional drivers and modules to be loaded into a deployed instance without rebuilding and redeploying the node instance through OSGi. It is important to update the build.gradle file where the OSGi manifest attributes are concerned. This will ensure that in the future if it is intended to be used, the module can be recognized and loaded by an OSGi enabled node.

1. Edit the osgi->manifest attribute for the ‘Bundle-Activator’ property to reflect the name of the package to com.sample.impl.sensor.simulated.Activator
2. Edit ext.details – this field contains the description assigned to your driver and visible in OpenSensorHub

“Driver Simulation - OpenSensorHub Driver Development Labs”
Update Manifest Details

3. Edit the Details within the `ext.pom` block
   - `id` – nickname, e-mail, etc
   - `name` – self-explanatory
   - `organization` – company name
   - `organizationUrl` – self-explanatory
Adding sensor-driver-simulation to Build Target

Updating and Building
Updating Project – Adding Driver

1. Open project wide build.gradle
2. Dependencies block shown with some examples commented out
3. Add dependency for driver:
   “implementation project('sensorhub-driver-simulation’)”
1. Open `settings.gradle`
Uncomment FileTree Builder
Modify FileTree Builder – Exclude Driver Template

1. Add statement to ignore project folders with having “template” in name
Refresh Gradle

Make sure to resync/refresh Gradle. This will ensure that the directory and its contents are included as part of the project and the folders such as ‘java’ be added as source roots for the module.
sensorhub-driver-simulation is ready to build!
Building Project

1. Execute build task
2. Build console log
3. Target – If build succeeds, creates build/distributions/[name]-[version].zip