

# Getting Started Guide

---

This guide will help you setup your development environment, build the Soft Sensor Manager component with IoTivity stack and sample applications. Once successfully build the stack, the Soft Sensor Manager Developer's guide will be a good reference to using the Soft Sensor Manager service

## Building Soft Sensor Manager and sample applications in Ubuntu

This guide will provide instructions on how to build the stack. Instructions provided in this guide are tested against Ubuntu 14.10, however, Ubuntu version 12.0.4 and above are supported.

### Build tools and libraries

Open the terminal window and follow the instructions below to install all the necessary tools and libraries in order to build IoTivity project.

#### Git

Git is a source code management software. IoTivity is set as a git project. Git is mandatory in order to get access to the IoTivity source code. Use the following command to download and install git.

```
$ sudo apt-get install git-core
```

#### ssh

Secure Shell is required to connect to GIT repository in order to checkout the IoTivity source code. SSH is typically part of the base operating system and should be include. If, for any reason, it is not available, it can be installed by running the following command in your terminal window.

```
$ sudo apt-get install ssh
```

#### G++

G++ is required in order to build the IoTivity stack. Download and install G++ by running following command in your terminal window

```
$ sudo apt-get install build-essential g++
```

#### Boost Version 1.55

Boost c++ library is required in order to build the IoTivity stack. Download and install Boost libraries by running the following command in your terminal window

```
$ sudo apt-get install libboost-all-dev
```

## Doxygen

Doxygen is a documentation generation tool used to generate API Documentaiton for the IoTivity project. Download and install doxygen by running following command in your terminal window.

```
$ sudo apt-get install doxygen
```

## Checking out the source code

Gerrit is a web-based code review tool built on top of the git version control system. Its main features are the side-by-side difference viewing and inline commenting which makes code reviews quick and simple task. Gerrit allows authorized contributors to submit changes to Git repository, after reviews are done. Contributors can get their code reviewed with a little effort, and get their changes quickly through the system.

[todo: Section that talks about how to checkout the source code from wiki]

## Building Soft Sensor Manager and its Applications

Once the source code is downloaded into a specific folder, **oic** in this context, you may follow the steps to build and execute Soft Sensor Manager and its applications.

### 1. Make sure that the downloaded code structure is as followings;

Two directories for oic-resources; oic-resource and oic- utilities

```
~/oic/oic-resource$ _
```

```
~/oic/oic- utilities$ _
```

The path for Soft Sensor Manager is as following;

```
~/oic/oic-service/Data_Management/SoftSensorManager$ _
```

The SoftSensorManager directory includes following sub directories;

Directories	Description
/SDK	The SDK APIs for applications is located.
/SSMCore	The SSM service codes

/SoftSensorPlugin	The source codes for soft sensors can be located in this folder. An example of soft sensors is <i>DiscomfortIndexSensor</i> .
/Ubuntu_SSMTesterApp	It is the sample application on Ubuntu. Basically, the input and output are displayed in the consol.
/Ubuntu_THSensorApp	A sample Physical sensor codes working with the DiscomfortIndexSensor in SSM, on Ubuntu. Temperature and Humidity are randomly created in the codes.

## 2. Modify environment settings for soft sensor library, <PlatformLayer.h>

Before building the source codes, the directory where soft sensor libraries (\*.so) are deployed should be modified since the path SSM installed will be different from individuals.

The path, SSM refers, is specified in the header file, <PlatformLayer.h>. There are two definitions to be modified; 1) *MODEL\_DIRECTORY* for \*.so files, and 2) *HIGH\_LOCATION* for xml manifest files for the \*.so files.

In the PlatformLayer.h, the two definitions are specified as following;

```
.
.
#define MODEL_DIRECTORY "/home/developer/ssm/"
#define HIGH_LOCATION "/home/developer/ssm/HighContextDictionary.xml"
.
```

They should be changed with the **full path** the source codes of the SoftSensorManager deployed and **‘/outputs/’** appended. Following shows the path provided by the original source code.

```
.
.
#define MODEL_DIRECTORY
    "/oic/oic-service/Data_Management/SoftSensorManager/outputs/"
#define HIGH_LOCATION
    "/oic/oic-service/Data_Management/SoftSensorManager/outputs/
    /HighContextDictionary.xml"
.
```

## 3. Modify the Makefile in the Soft Sensor Manager directory with the local boost path.

Installing Boost is one of prerequisites to use Iotivity Base, and there are references for the Boost path in Makefiles of SoftSensorManager. Following codes shows the default path for Boost and users should modify the exact path in their environment.

```
IOT_BASE=../../../../oic-resource
BOOST=../1_SDK/util/boost_1_56_0
SRC_PATH=.
RST_NAME=Debug.
.
.
.
```

#### 4. Run make

```
~/oic/oic-service/Data_Management/SoftSensorManager$ make
```

#### 5. Execute THSensorApp, SSMService and AppResourceClient

To initiate SSM service, please enter as following;

```
~/oic/oic-service/Data_Management/SoftSensorManager/Outputs$ ./THSensorApp
```

To initiate SSM service, please enter as following;

```
~/oic/oic-service/Data_Management/SoftSensorManager/Outputs$ ./SSMResourceServer
```

To initiate SSM Client Test Application, please enter as following;

```
~/oic/oic-service/Data_Management/SoftSensorManager/Outputs$ ./AppResourceClient
```

Note that the sequence of process initiations should be followed due to the process dependencies.