The User Manual of IoTSimulator

by Fred Chang

[Introduction 1](#_Toc417041224)

[Building and Executing 1](#_Toc417041225)

[Configure file 2](#_Toc417041226)

[**1.** **Sensor support list** 2](#_Toc417041227)

[**2.** **Interface description** 3](#_Toc417041228)

[**3.** **Virtual mote description** 4](#_Toc417041229)

[Programmable value 5](#_Toc417041230)

[**1.** **Random mode** 5](#_Toc417041231)

[**2.** **Programmable mode** 5](#_Toc417041232)

# Introduction

It is difficult to allocate abundant sensors to verify IoT system, so this simulator is used to generate many virtual configurable motes(be a mote manager) to detect unknown bugs between gateway and cloud. In some scenarios, we can use it to simulate a developing sensor and analyze the interaction between them to increase our system compatibility or speed up our development progress. We can also release the simulator to help our customer speedup the development.

# Building and Executing

Modify the cross prefix string in Makefile and make and use the file “libDustWsnDrv.so.[version]”.

# File structure

# Configure file

The configure files be separated into following parts.

1. **Sensor support list**

This file is placed in execute path and its filename is a static name “@mote\_list.ini”. Many kinds of sensors were defined in this file, and those would be used to match the one inside a virtual mote’s ini to get default values and fundamental parameters. The form is shown below.

Sensor\_Type="Temperature"

Sensor\_Default\_Info="n:Temp, u:Cel"

Sensor\_Default\_Value="v:(##), min:0, max:100"

Sensor\_Default\_Trigger\_Mode=”Edge”

* The **Sensor\_Type** indicates the kind of sensor.
* The **Sensor\_Default\_Info** has two values, name(n) and unit(u).
* The **Sensor\_Default\_Value** has three values, value(v), minimum(min) and maximum(max). The sign “(##)” means a programmable value which is generated by simulator, and it will be explained in chapter programmable value.
* The **Sensor\_Default\_Trigger\_Mode** has an alternative option, Edge or Level. Edge is reported by triggering, and Level is reported per second.

1. **Interface description**

This file is placed in a directory named by Communication\_Type like “wsn” and filename is fixed to “@interface.ini”. The type of the interface on gateway was defined in it like Wifi, WSN, Zeebig, Bluetooth and so on. Those had some important values like mac or uuid. The form is shown below.

Communication\_Type="WSN"

[Interface]

Interface\_Number="2"

Interface\_Name="WSN0"

Interface\_Id="0001852CF4B7B0E8"

Interface\_Health="90"

Interface\_Name="WSN1"

Interface\_Id="0001852CF4B7B0E7"

Interface\_Health="0"

* The **Communication\_Type** indicates the type of interface like Wifi, BT, Zigbee and so on.
* In the Interface block, it is used to define the number of interface first, and then define other parameter. Each interface has three parameters. The **Interface\_Name** is the name of interface. The **Interface\_Id** is the mac or the uuid of interface. The **Interface\_Health** is the calculated result of health.

1. **Virtual mote description**

We can make many description files to simulate motes. Those files are placed in a directory named by Communication\_Type like “wsn”, and all files in it will be recognized into the mote description. The mote description file can include many sensors. In other words, a mote means a sensor hub. The example is shown below.

[SensorHub]

Sensor\_Product="WISE1020"

Sensor\_Id="0001000EC6F0F832"

Sensor\_Connect\_To="0001000EC6F0F831"

[Sensor]

Sensor\_Element\_Size="2"

Sensor\_Type="CO2"

Sensor\_Info="n:Environment CO2"

Sensor\_Value="v:25, min:0, max:50"

Sensor\_Trigger\_Mode=“Edge"

Sensor\_Type="Temperature"

Sensor\_Info="n:Room Temp"

Sensor\_Value="min:0, max:100"

Sensor\_Trigger\_Mode=“Level@3"

* **Sensor\_Product** is a user-defined name.
* **Sensor\_Id** is mac or uuid.
* **Sensor\_Connect\_To** is a list of mac connected to this mote.
* **Sensor\_Element\_Size** indicates the number of sensor.
* **Sensor\_Type**、**Sensor\_Info**、**Sensor\_Value** are the same as **Sensor support list**. If the value is not defined in this scope, the default value in **Sensor support list** will be adopted.
* **Sensor\_Trigger\_Mode** has two choices, Edge and Level. Edge is triggered if value is changed. Level is triggered per custom sec, but it is also triggered by that value is changed.

# Programmable value

The sign (##) means it’s a programmable code, there are two methods which is shown below.

1. **Random mode**

If there are no other words in the center of the sign like (##), It is the random mode. The value will updated by random value per second.

1. **Programmable mode**

The syntax is “(#[second]@[value]- [second]@[value]-…and so on#)” like “(#3@5-7@13-9@1#)”. The meaning of “(#3@5-7@13-9@1#)” is shown below chart. The latest second word like 9+1=10 is the period of the code, and the value of the zero second point is the extension of the latest second.