Update: 2015/03/10
Potang Chen

SUSI IoT Demo Program Instructions

For Windows

Introduction

Advantech SUSI IoT Demo is a demonstrational application that shows the powerful abilities of the Advantech's SUSI IoT library. This app provides a simple and effective user interface to help user to access various sensing modules connected to user's platform. SUSI IoT is designed to be highly scalable and friendly, developers can make their own library/driver/app to compatible with SUSI IoT library by following the program document and porting guide. There are three main purposes to use SUSI IoT Demo:

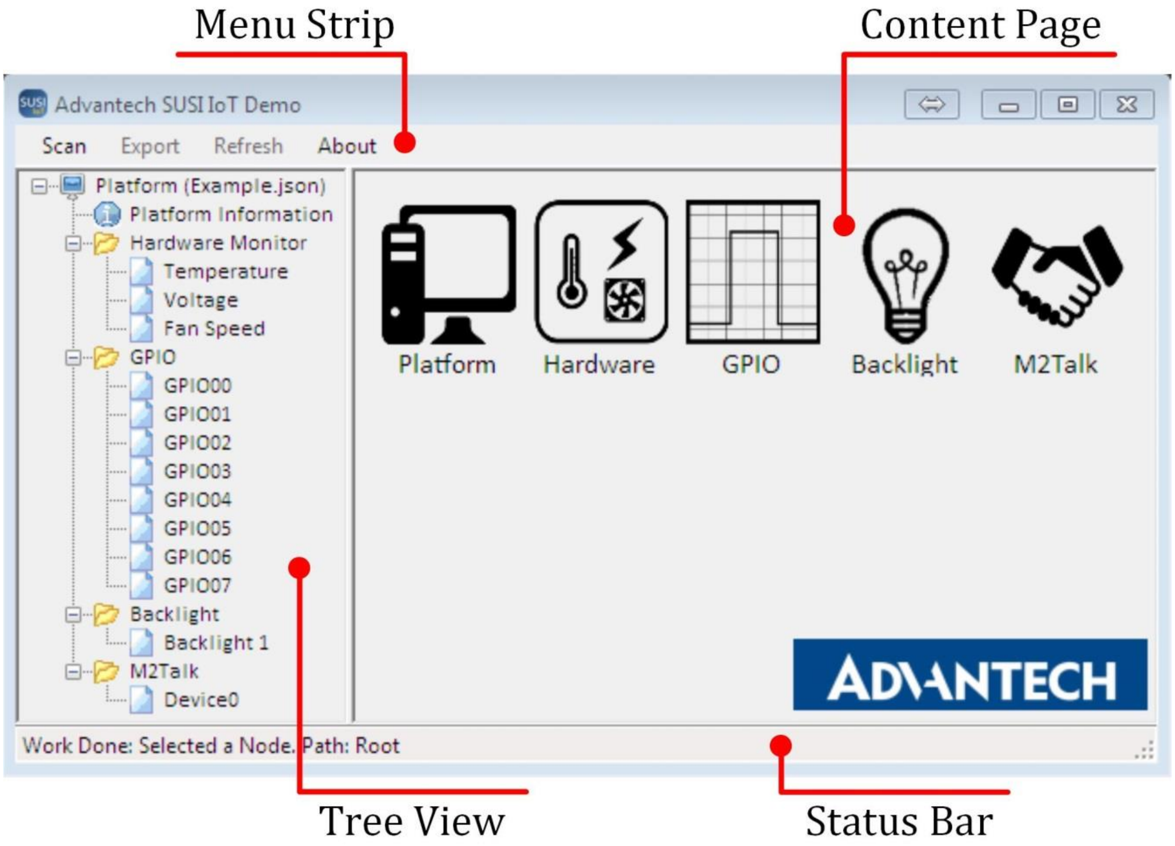
I. Develop: verify developer's compatible library/driver/app
II. Debug: locate which layer an issue happened
III. Monitor/Control: access the connected modules directly

Environment Requirements

Contents

1. User Interface

The graphical user interface of SUSI IoT Demo is shown as Figure 1. The layout is divided simply and clearly into four parts: Menu Strip, Tree View, Content Page and Status Bar.



Work Done: Selected a Node. Path: Root

Figure 1 SUSI IoT Demo User Interface

3.1 Menu Strip

The Menu Strip is on the top of the window. There are four options on this strip, such as: Scan, Export, Refresh and About.

(A) Scan

This option contains two sub-options: “This platform” and “Open File...” as Figure2. Users are capable to choose data source by selecting options.

- Select “This platform” to enter “General Mode”

  This application will then start to scan the real-time generated data by the connected sensors. The detail of General Mode will be illustrated in following chapter.

- Select “Open File…” to enter “Develop Mode”
  User can load a file contained JSON to verifying development work. The detail of Develop Mode will be illustrated in following chapter.

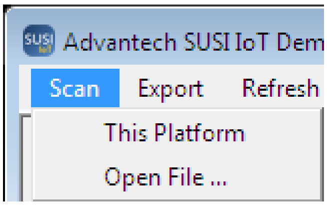


Figure 2 Options of Scan Menu Strip

(B) Export

The data of SUSI IoT Library is transferred as a flexible format designed by Advantech which is based on JSON. For develop purpose, user is able to export the JSON data in two ways. As you can see in figure 3, user can save it into a text file, or simply copy it to clipboard.

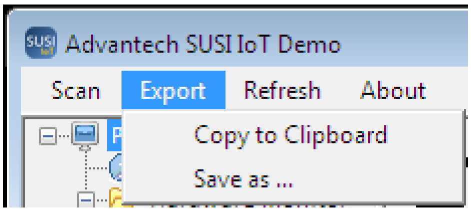


Figure 3 Options of Export Menu Strip

(C) Refresh

The refresh rate of SUSI IoT data can be switched in four different speeds: Fastest, One second, Five second and None.

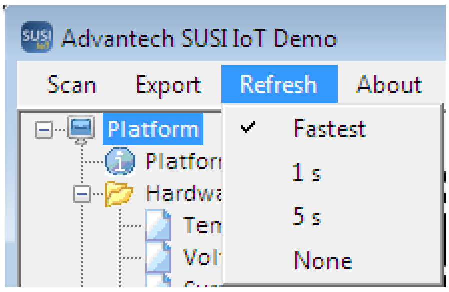


Figure 4 Options of Refresh Menu Strip

(D) About

This option contains the version, license and company information of SUSI IoT Demo application.

3.2 Tree View

The Tree View is placed on the left of the window. This panel shows the hierarchy structure of the data nodes. By selecting a node in Tree View, the SUSI IoT data contained in the selected node will be listed in the Content Page on the right. There are three kinds of node in Tree View:

(A) Root Node

The root node is the root of the tree view. It is always named as “Platform” and shown with a computer icon.

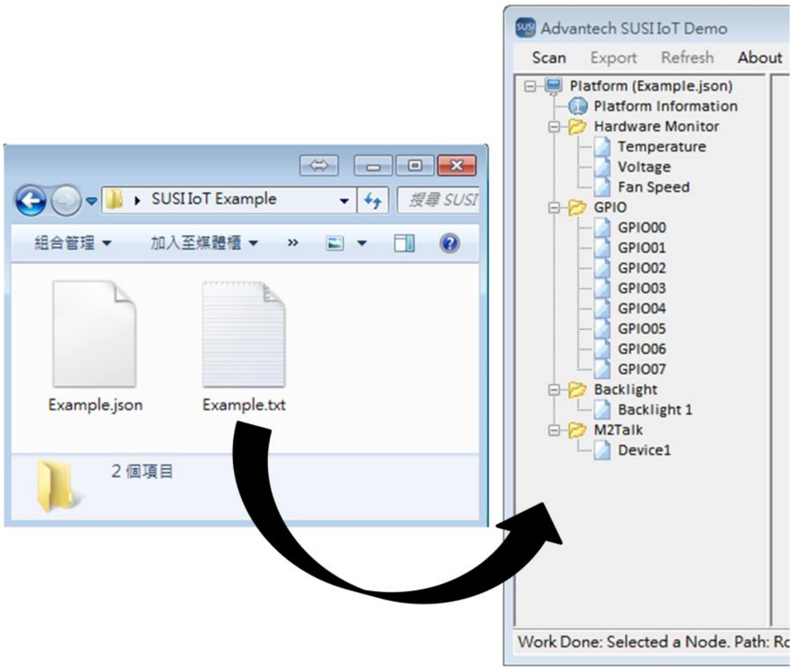
(B) Class Node

A class node is a node containing several group nodes. For instance, a Hardware Monitor class node might contain Temperature, Voltage and Fan Speed group nodes. The icon of a class node is a directory.

(C) Group Node

A group node is a node containing SUSI IoT data items. For example, a Voltage group node might contain VCore, V3.3 and VBattery data items. The icon of a group node is a list document.

Furthermore, as mentioned before in Chapter 1.1 Menu Strip, SUSI IoT Demo supports to open a JSON text file with an open file dialog, it is also supported to drag and drop a file into Tree View to load a file.



Work Done: Selected a Node. Path: Rc

Figure 5 Drag and drop function of the Tree View

3.3 Content Page

The content page is placed on the right of the window. This panel displays the content of a selected tree node in Tree View. There are three kind of pages possibly showed: Root Page, Class Page and Group Page.

(A) Root Page

While the root node in Tree View is selected, the Root Page displays. In this page, SUSI IoT Demo creates one hyper link icon for each available module on user's platform as shown in figure 6. Besides SUSI modules are represented with a certain set of icons, others are represented with a hand-shaking icon uniformly. As soon as user clicks a hyper link icon, the corresponding data content will be shown in Content Page.

(B) Class Page

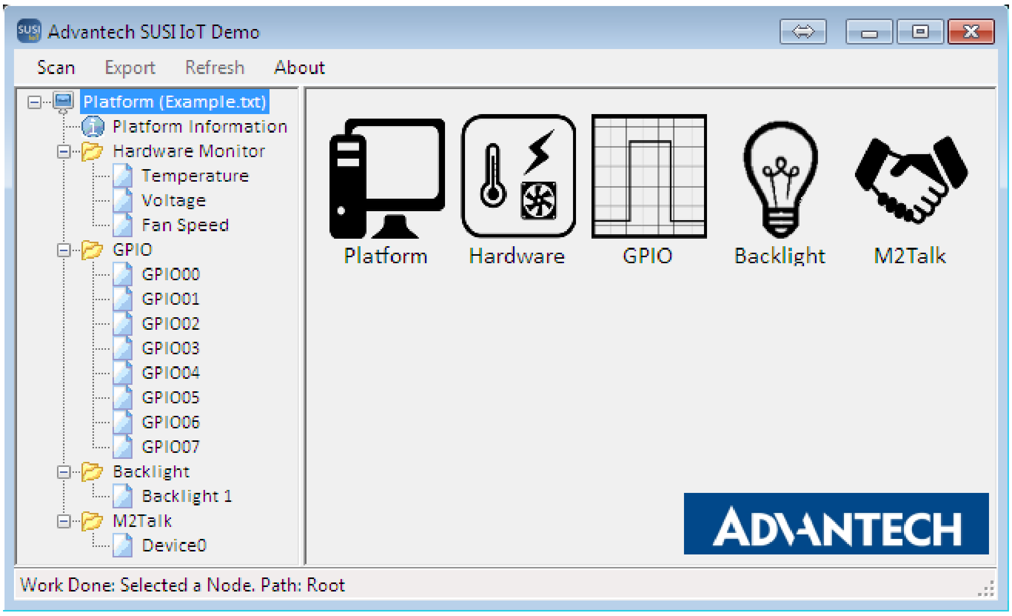
In a Class Page, the data of the group nodes which belong to this class is merged and listed into one form as shown in Figure 7. There are eight columns in the form: Name, ID, Range, Value, Unit, Read, Write and Others.

- Name, ID, Range, Value and Unit

  These columns are pre-defined properties of a data item.
- Read and Write

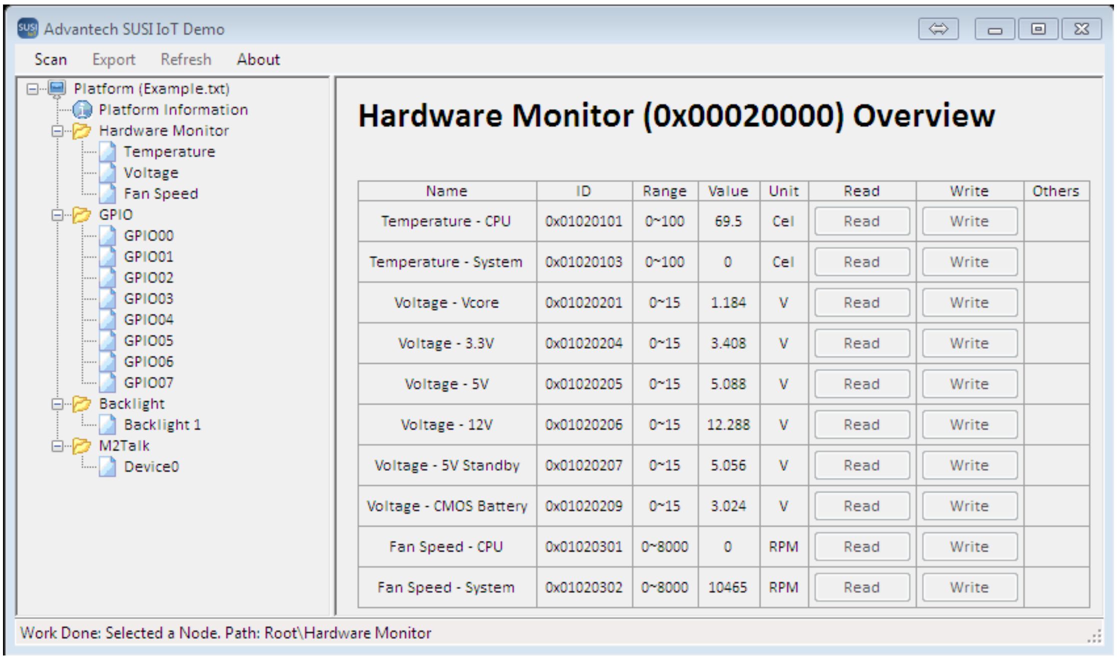
  These columns provide user read/write control buttons of data items. The availabilities of buttons depend on the "rw" property in JSON data.
- Others

  This column shows the extended properties of a data item.



Work Done: Selected a Node. Path: Root

Figure 6 Hyper link icons in Root Page



Hardware Monitor (0x00020000) Overview

Name	ID	Range	Value	Unit	Read	Write	Others
Temperature - CPU	0x01020101	0~100	69.5	Cel	Read	Write	
Temperature - System	0x01020103	0~100	0	Cel	Read	Write	
Voltage - Vcore	0x01020201	0~15	1.184	V	Read	Write	
Voltage - 3.3V	0x01020204	0~15	3.408	V	Read	Write	
Voltage - 5V	0x01020205	0~15	5.088	V	Read	Write	
Voltage - 12V	0x01020206	0~15	12.288	V	Read	Write	
Voltage - 5V Standby	0x01020207	0~15	5.056	V	Read	Write	
Voltage - CMOS Battery	0x01020209	0~15	3.024	V	Read	Write	
Fan Speed - CPU	0x01020301	0~8000	0	RPM	Read	Write	
Fan Speed - System	0x01020302	0~8000	10465	RPM	Read	Write	

Work Done: Selected a Node. Path: Root\Hardware Monitor

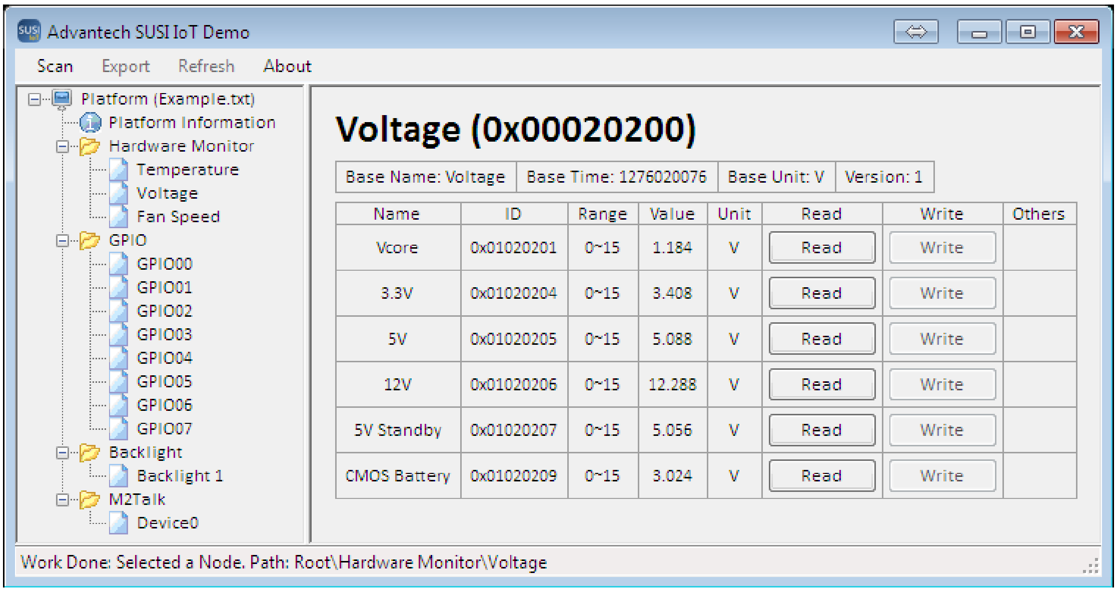
Figure 7 Data List in Class Page

(C) Group Page

In a group page, the data item of this group is listed in a form with the same columns in class page. The group page is shown as figure 8.

3.4 Status Bar

The Status Bar is placed on the bottom of the window. The status message will be shown here to let user knows the current condition of SUSI IoT Demo.



Voltage (0x00020200)

Base Name: Voltage | Base Time: 1276020076 | Base Unit: V | Version: 1

Name	ID	Range	Value	Unit	Read	Write	Others
Vcore	0x01020201	0~15	1.184	V	Read	Write	
3.3V	0x01020204	0~15	3.408	V	Read	Write	
5V	0x01020205	0~15	5.088	V	Read	Write	
12V	0x01020206	0~15	12.288	V	Read	Write	
5V Standby	0x01020207	0~15	5.056	V	Read	Write	
CMOS Battery	0x01020209	0~15	3.024	V	Read	Write	

Work Done: Selected a Node. Path: Root\Hardware Monitor\Voltage

Figure 8 Data List in Group Page

2. Application Modes

SUSI IoT Demo Application provides two modes for users: General Mode and Develop Mode. By selecting the sub-options in the “Scan” option in Menu Strip, user can easily switch between these two modes.

2.1 General Mode

The entering method of General Mode is to select “This platform” in “Scan”. In General Mode, the connected modules are accessed actually. The data signal generated by the modules is encoded and transferred as JSON by the corresponding drivers and libraries. SUSI IoT Demo decodes this JSON data and displays the result with convertible refreshing speed. Also, the writing/setting command is available in General Mode, too.

Using SUSI IoT Demo is quite convenient when a user only needs to monitor and manually control their modules, and the General Mode perfectly fulfills this kind of user scenario which is also the most popular case.

2.2 Develop Mode

There are two methods to enter Develop Mode: one is to open file in “Scan” and the other is to drag and drop a JSON text file into Tree View. In Develop mode, developers are able to verify their own design before spending time and resource to build a driver/library. Developers can load an evaluating JSON data file to do designing and verification. After the developer’s JSON data is verified, the driver and library can be tested in General Mode. The .dll library file should be placed under C:\Windows\SUSI\DLL\.

NOTICE It is restricted to do Read, Write, Export and Refresh in Develop Mode due to the reason that the data source is from text file. The function availability in different mode is sorted out in Table 1.

Mode	Data Source	Export	Refresh	Read/Write
General	Platform	O	O	O
Develop	File	X	X	X

Table 1 Function availability in different modes

3. JSON-Variant GUI

To provide the best flexibility and user experience, various properties are defined in SUSI IoT JSON format. Some of the properties are making the DEMO GUI showing differently, the relative effect of these properties will be explained in this chapter. The complete format rules are illustrated in the SUSI IoT Library document.

3.1 JSON

According to the library document, there are 9 pre-defined properties in a leaf node(data item), such as "n", "min", "max", "u", "ui", "sui", "Id" ,"rw" and for value using "v" or "sv" or "bv". In addition, developers can extend any properties with any key name and any value. The relation between JSON and GUI is specified in Table 2.

JSON Key	GUI Column	Valid Value	Description
n	Name	any	the Name of this lead node
Id	ID	any	the ID of this leaf node
max/min	Range	any	the Range of Value
ui	Value	""/ComboBox /TextBox /Label	the type of GUI Component for showing the Value
v/sv/bv	Value	any	the Value of this leaf node
u	Unit	any	the Unit of the Value
rw	Read/Write	R/W/B	the read/write capability of the Value
sui	--	""/PlotChart	the type of SUSIAccess GUI Component for showing the Value
EXTENDED	Others	any	any extended properties will be listed in Others column

Table 2 Relation between JSON and GUI

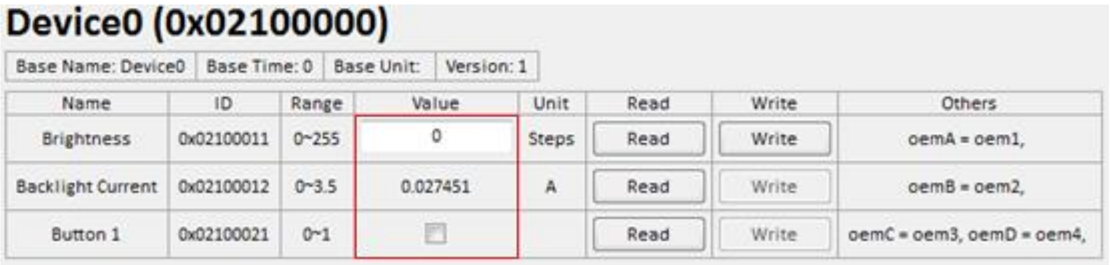
3.2 Example

Figure 9 is an example illustrating how the Example 1 JSON data affects the appearance of SUSI IoT Demo. In this example, three data items are used: Brightness, Backlight Current and Button 1.

For “Value”, the “ui” JSON properties are “TextBox”, “Label” and “CheckBox” in sequence. As you can see the red marked part in figure 9, the GUI components in “Value” column are different.

For “Read” and “Write”, the “rw” JSON properties are “B”, “R” and “R” in sequence. In consequence, the Write buttons of Backlight Current and Button 1 are disabled.

For “Others”, there are several properties prefixed with “oem” in the contents of these three data items. As shown in Figure 9, these “oem” properties are all listed in “Others” column.



Device0 (0x02100000)

Base Name: Device0	Base Time: 0	Base Unit:	Version: 1

Name	ID	Range	Value	Unit	Read	Write	Others
Brightness	0x02100011	0~255	0	Steps	Read	Write	oemA = oem1,
Backlight Current	0x02100012	0~3.5	0.027451	A	Read	Write	oemB = oem2,
Button 1	0x02100021	0~1	□		Read	Write	oemC = oem3, oemD = oem4,

Figure 9 GUI with Different JSON

Example 1

```
{
   "M2Talk": {
      "bn": "M2Talk",
      "Id": 34603008,
      "Device0": {
         "Id": 34603008,
         "e": [
            {
               "n": "Brightness",
               "v": 0,
               "min": 0,
               "sui": "",
               "max": 255,
               "u": "Steps",
               "rw": "B",
               "ui": "TextBox",
               "Id": 34603025,
               "oemA": "oem1"
            },
            {
               "n": "Backlight Current",
               "v": 0.027451,
               "min": 0.0,
               "sui": "",
               "max": 3.5,
               "u": "A",
               "rw": "R",
               "ui": "Label",
               "Id": 34603026,
               "oemB": "oem2"
            },
            {
               "n": "Button 1",
               "v": 0,
               "min": 0,
               "sui": "",
               "max": 1,
               "u": "",
               "rw": "R",
               "ui": "CheckBox",
               "Id": 34603041,
               "oemC": "oem3",
               "oemD": "oem4"
            }
         ],
         "bn": "Device0",
         "bu": "",
         "bt": 0,
         "ver": 1
      }
   }
}
```