

# SpecTalk: Integrating Sensor Specifications into IoT Implementation



林一平(Yi-Bing Lin)  
Winbond Chair Professor  
NYCU, CMU, NCKU, Asia U

# Introduction

- Taiwan Association of Information and Communication Standards (TAICS) is developing an automatic process to allow the IoT applications to be conformed with TAICS specifications.
- TAICS is sponsored by Ministry of Economic Affairs, Taiwan.

# Taiwan Association of Information and Communication Standards (TAICS)

- TAICS is an industry organization founded in June 2015 with the members from industry, research and academia organizations in Taiwan
- The objective of TAICS is to bridge the local industry with global standard initiatives/organizations by contributing the study results or consolidated consensus, it may also develop the local standard or study report per request.
- TAICS is open for the registration by all the companies/organizations with division in Taiwan.



台灣資通產業標準協會

Taiwan Association of Information and Communication Standards

## Technical Committees of TAICS

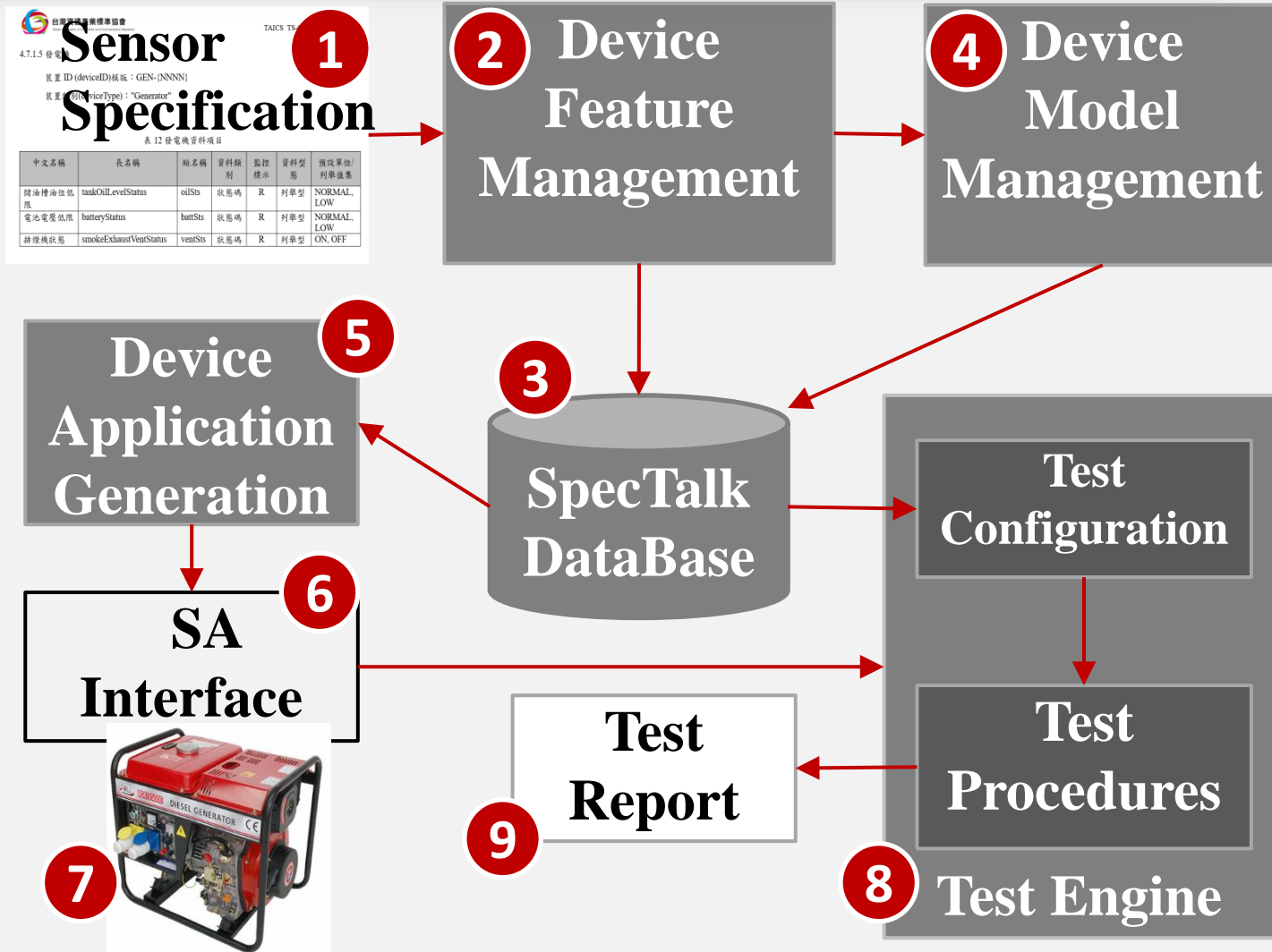
Seven technical committees (TCs) have been chartered focusing on

- Advanced Mobile Communication
- Network Communication
- Device Internetworking
- Audiovisual Services and Communications
- Network and Information Security
- Intelligent Buildings ICT
- Internet of Vehicles (IoV) & Automated Driving

## Bridging the IoT Applications with TAICS Specifications

- Proposed by Yi-Bing Lin, a member of Board of Directors of TAICS, TAICS is developing an automatic process to confirm the IoT applications with TAICS specifications.
- This mechanism was The procedures have been implemented and is being used to inspect the smart buildings in Shuinan Trade and Economic Park, Taichung

# Test an IoT Device with TAICS Specification



# Example: TAICS spec for Generator



## 4.7.1.5 發電機

2 裝置 ID (deviceID) 模版: GEN-{NNNN} **Device ID**

1 裝置類別(deviceType): "Generator" **Device Model (DM)**

### Device Feature (DF)

表 12 發電機資料項目

中文名稱	長名稱	短名稱	資料類別	監控標示	資料型態	預設單位/列舉值集
3 儲油槽油位低限	4 tankOilLevelStatus	5 oilSts	狀態碼	6 R	列舉型	7 NORMAL, LOW
電池電壓低限	batteryStatus	battSts	狀態碼	R	列舉型	NORMAL, LOW
排煙機狀態	smokeExhaustVentStatus	ventSts	狀態碼	R	列舉型	ON, OFF

DF parameter

I/O=I

# Step 1. Device Feature (DF) Creation

中文名稱	長名稱	短名稱	資料類別	監控標示	資料型態	預設單位/列舉值集
儲油槽油位低限	tankOilLevelStatus	oilSts	狀態碼	R	列舉型	NORMAL, LOW

TS-0022

**3**

Device Feature    Device Model    Logout

### Device Feature Window

Type  IDF     ODF    **6** Category test **1**

DF Name **5** oilSts-I **2**

Number of parameters

Type	Min	Max	Unit
boolean	0	1	None

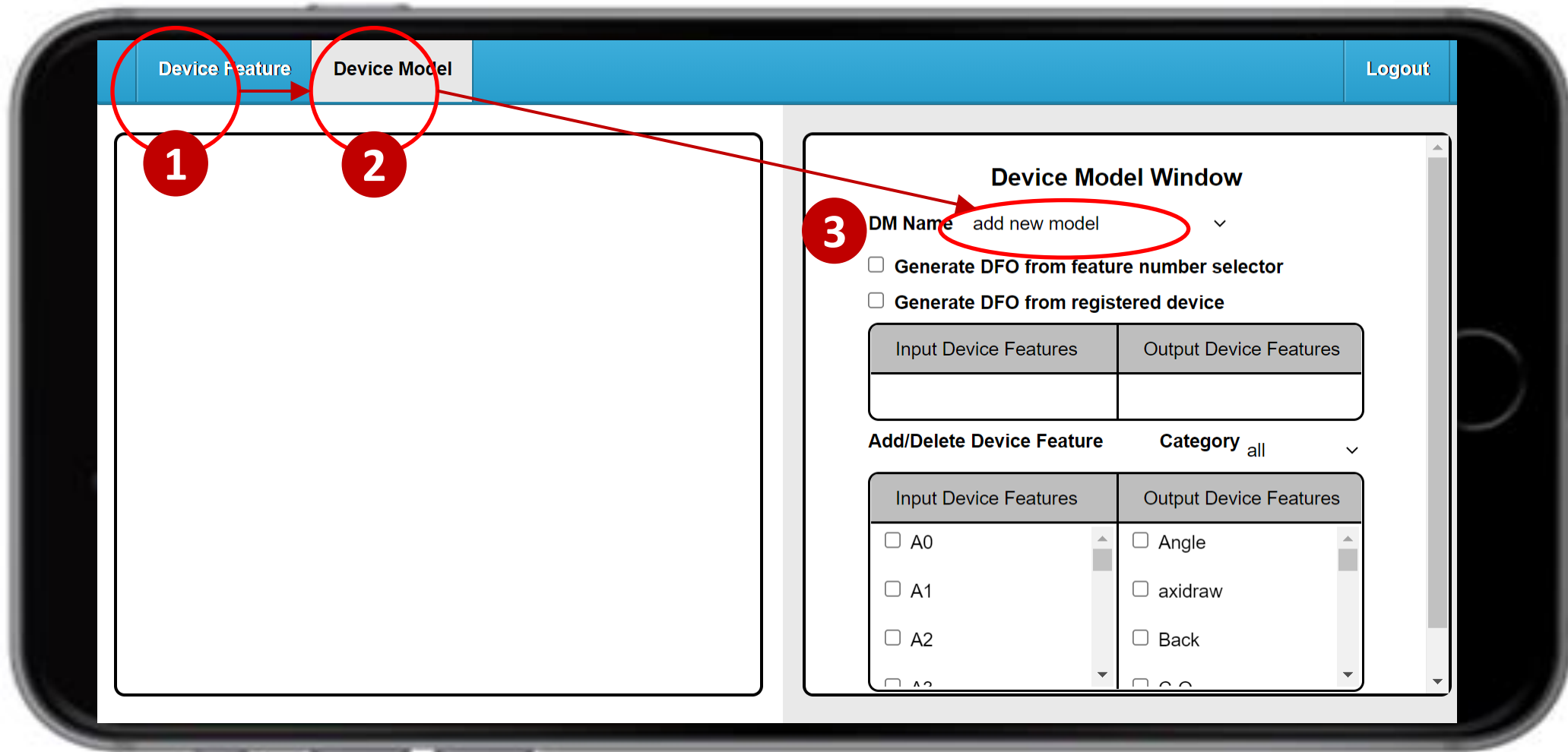
Save    Delete

### Description

儲油槽油位低限; tankOilLevelStatus; {NORMAL, LOW}



## 2.1. Add a new Device Model



## 2.2. Add DF battSts to the DM

**Device Feature Window**

Type  IDF     ODF    Category test    v

DF Name **battSts-I**    v

Number of parameters

Type	Min	Max	Unit
boolean    v	<input type="text" value="0"/>	<input type="text" value="1"/>	None    v

**Device Model Window**

DM Name add new model    v

Generate DFO from feature number selector

Generate DFO from registered device

Input Device Features	Output Device Features
<b>battSts-I</b> No Tag    v	

Add/Delete Device Feature    Category all    v

Input Device Features	Output Device Features
<input checked="" type="checkbox"/> <b>battSts-I</b>	<input type="checkbox"/> Angle
<input type="checkbox"/> Button1	<input type="checkbox"/> axidraw
<input type="checkbox"/> Button2	<input type="checkbox"/> Back

表 12 發電機資料項

中文名稱	長名稱	短名稱
儲油槽油位低限	tankOilLevelStatus	oilSts
電池電壓低限	batteryStatus	<b>battSts</b>
排煙機狀態	smokeExhaustVentStatus	ventSts

## 2.3. Add DF oilSts to the DM

Device Feature      Device Model      Logout

**Device Feature Window**

Type  IDF     ODF    Category test    ▾

DF Name oilSts-I    ▾

Number of parameters

Type	Min	Max	Unit
boolean ▾	<input type="text" value="0"/>	<input type="text" value="1"/>	None ▾

**Device Model Window**

DM Name add new model    ▾

Generate DFO from feature number selector

Generate DFO from registered device

Input Device Features	Output Device Features
battSts-I    No Tag ▾	
oilSts-I    No Tag ▾	

Add/Delete Device Feature    Category all    ▾

Input Device Features	Output Device Features
<input checked="" type="checkbox"/> oilSts-I	<input type="checkbox"/> Angle
<input type="checkbox"/> Orientation	<input type="checkbox"/> axidraw

表 12 發電機資料項

中文名稱	長名稱	短名稱
儲油槽油位低限	tankOilLevelStatus	oilSts
電池電壓低限	batteryStatus	battSts
排煙機狀態	smokeExhaustVentStatus	ventSts

## 2.4. Add DF ventSts to the DM

**Device Feature Window**

Type  IDF     ODF    Category test    v

DF Name ventSts-l    v

Number of parameters 1

Type	Min	Max	Unit
boolean v	0	1	None v

**Device Model Window**

DM Name add new model    v

Generate DFO from feature number selector

Generate DFO from registered device

Input Device Features	Output Device Features
battSts-l No Ta v	
oilSts-l No Ta v	
ventSts-l No Ta v	

Add/Delete Device Feature    Category all    v

Input Device Features	Output Device Features
<input checked="" type="checkbox"/> ventSts	<input type="checkbox"/> Angle

表 12 發電機資料項

中文名稱	長名稱	短名稱
儲油槽油位低限	tankOilLevelStatus	oilSts
電池電壓低限	batteryStatus	battSts
排煙機狀態	smokeExhaustVentStatus	ventSts

## 2.5. Create the DM Name

The Device Model name: Generator

Device Feature

Type  IDF  ODF Category test

DF Name ventSts

Number of parameters 1

Output Device Features

oilSts-l No Ta

ventSts-l No Ta

Add/Delete Device Feature Category all

中文名稱	長名稱	短名稱
儲油槽油位低限	tankOilLevelStatus	oilSts
電池電壓低限	batteryStatus	battSts
排煙機狀態	smokeExhaustVentStatus	ventSts

Input Device Features

ventSts

Volume-l

weather

Output Device Features

Angle

axidraw

Back

Save

表 12 發電機資料項

## 2.6. Generator is added to the DM List

Device Feature      Device Model      Logout



表 12 發電機資料項

中文名稱	長名稱	短名稱
儲油槽油位低限	tankOilLevelStatus	oilSts
電池電壓低限	batteryStatus	battSts
排煙機狀態	smokeExhaustVentStatus	ventSts

**Device Model Window**

DM Name: Generator

- Generator
- Generator

Input Device: Fan

battSts: game

oilSts: **Generator**

ventSts: Glasses

Add/Delete: Intel-Cam

Input Device: Jack

A0: Koala

Lamp

## Step 3. Device Application Generation

The screenshot shows a web browser with three tabs: "IoTtalk Homepage", "IoTtalk", and "DF Management". The address bar displays "iottalk2.tw/connection". The browser's bookmark bar includes "應用程式", "Bookmarks", "Gmail - Inbox (2) ...", "tvt-web", "twc-web", and "iot". The main content area features a blue header with several buttons: "Generator-Test" (circled in red with a red circle containing the number 3), "Model" (circled in red with a red circle containing the number 4), "Flush" (with a green indicator), and "Delete". A dropdown menu is open under the "Model" button, listing various device types: "Dandelion", "Dummy\_Device", "Dummy\_Device\_Tag", "Fan", "game", "Generator" (circled in red with a red circle containing the number 5), "Glasses", and "GPS". A red dashed arrow points from the "DF Management" tab to the "Generator" option in the dropdown menu. Another red arrow points from the "IoTtalk" tab to the "Generator-Test" button. A third red arrow points from the "IoTtalk" tab to the "Model" dropdown menu.

## 3.1. Specify the Generator under Test

The screenshot displays a software interface for configuring a generator. At the top, there is a blue header bar with four buttons: "Generator-Test" (with a dropdown arrow), "Model" (with a dropdown arrow), "Flush" (with a green indicator dot), and "Delete".

Below the header, the interface is split into two main sections:

- Left Section:** A list of generators. The top item is "Generator" with a gear icon. Below it are three items: "battSts-I", "oilSts-I", and "ventSts-I". A red circle highlights this entire list.
- Right Section:** A configuration panel for the selected generator. It is titled "Generator" and contains two sections:
  - Input Device Features:** This section contains three dropdown menus: "battSts" with the value "1", "oilSts" with the value "1", and "ventSts" with the value "1". A red circle highlights these three dropdowns.
  - Output Device Features:** This section is currently empty.

At the bottom of the right section, there is a "Save" button, which is also circled in red. A dashed red arrow points from the "ventSts-I" item in the left list to the "Save" button.



## 3.2. Set up the Parameters of the Generator

Generator-Test ▾

Model ▾

Flush ●

Delete

Simulation

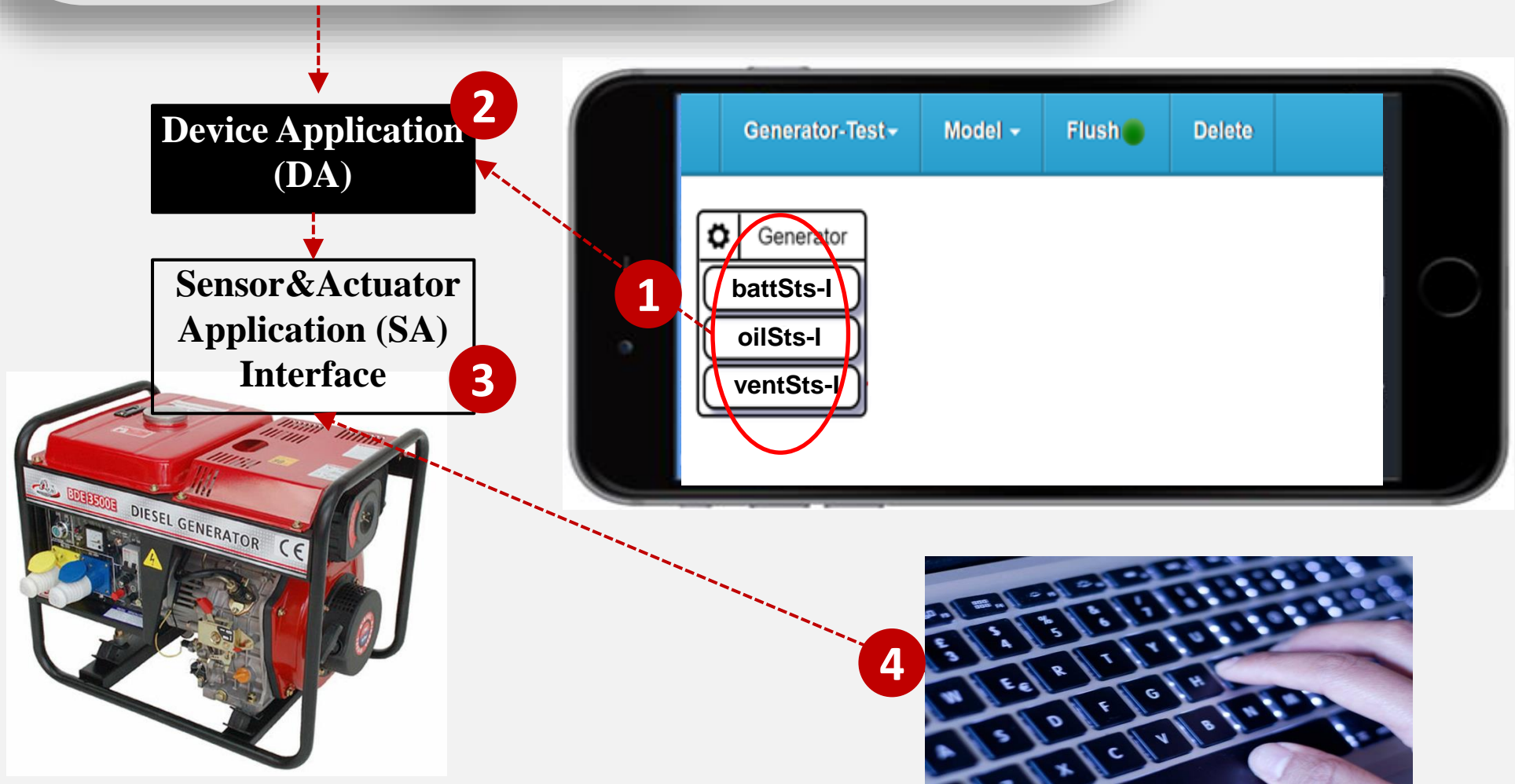
Generator

- battSts-l
- oilSts-l
- ventSts-l**

Generator					
ventSts ✎	Type	Function	Simulation Range		
			Min	Max	
x1	sample ▾	disable ▾	0	1	

Save

## Step 3.3. Automatically Generate the Device Application



**Step 4. The manufacture is responsible for writing the SA interface**

### Step 3. Automatically generates the SA Interface for Generator

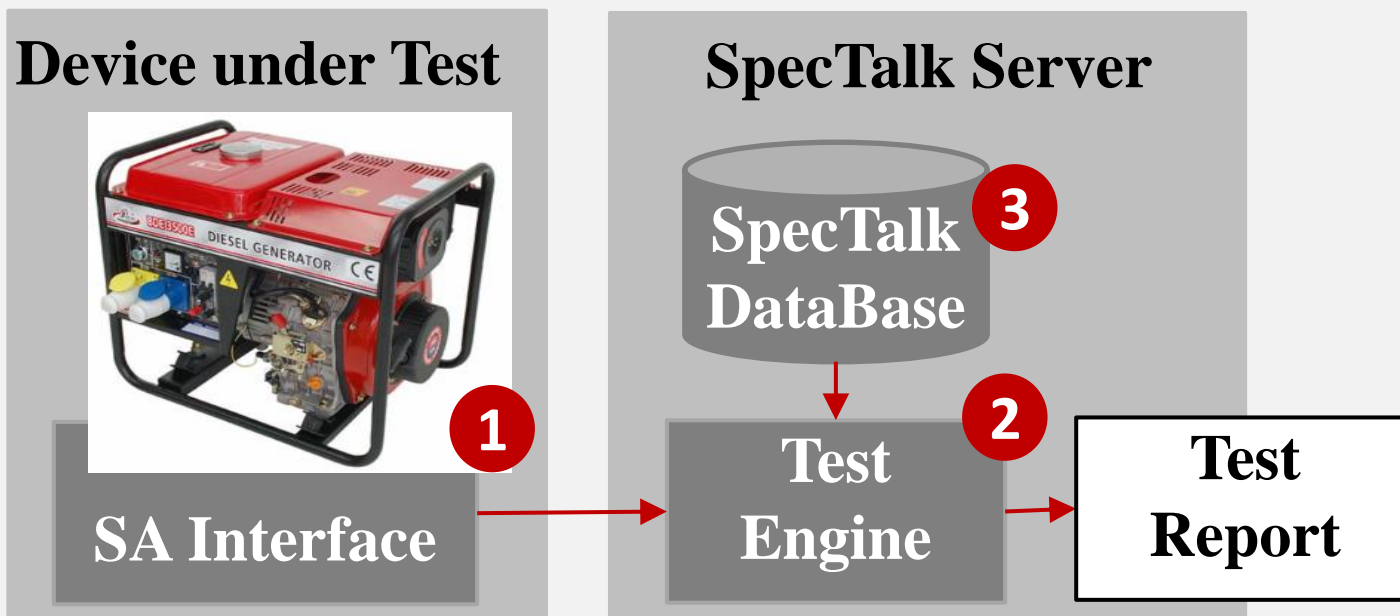
**Generator**

Name	Your-Device-Name
API	Your-API
oilSts	Your-oilSts-function
battSts	Your-battSts-function
ventSts	Your-ventSts-function
File	File-for-SA-code

### Step 4. The manufacturer specifies its generator

```
01. import time, random, requests↵
02. import DA↵
03. import Your-API↵
04. ServerURL = 'https://DomainName' ↵
05. Reg_addr = "GEN-0012"↵
06. DA.profile['dm_name']='Generator'↵
07. DA.profile['df_list']=['oilSts', 'battSts', 'ventSts',]↵
08. DA.profile['d_name']='Your-Device-Name' ↵
09. DA.register(ServerURL, Reg_addr)↵
10. while True:↵
11.     try:↵
12.         oilSts_data = Your-oilSts-function ↵
13.         DA.push('oilSts', oilSts_data) ↵
14.         battSts_data = Your-battSts-function ↵
15.         DA.push('battSts', battSts_data)↵
16.         ventSts_data = Your-ventSts-function ↵
17.         DA.push('ventSts', ventSts_data)↵
18.     except Exception as e:↵
19.         print(e)↵
20.         if str(e).find('mac_addr not found:') != -1:↵
21.             print('Reg_addr is not found. Try to re-register.')↵
22.             DA.register (ServerURL, Reg_addr)↵
23.         else:↵
24.             print('Connection fails.')↵
25.             time.sleep(1) ↵
26.             time.sleep(0.2)↵
```

## Step 5. SpecTalk Tests the Generator



# Step 5. Simulate the Generator

The screenshot displays a simulation software interface. At the top, a blue toolbar contains buttons for 'ator-Test', 'Model' (circled in red with a '1'), 'Flush', 'Delete', 'Simulation' (with a green toggle and circled in red with a '4'), and 'Impo'. Below the toolbar, a 'Generator' block is shown on the left, containing sub-blocks 'battSts-I', 'oilSts-I', and 'ventSts-I' (circled in red with a '3'). A 'Join2' block connects the 'ventSts-I' block to a 'Controller' block. The 'Controller' block contains a 'Display' block (circled in red with a '2'). To the right, two 'Monitor' windows are visible. The 'IDF Monitor' window shows a table with 'Sub-stage: Input' and a 'Continue' button. The 'ODF Monitor' window shows a table with 'Sub-stage: Function' and a 'Table' button. Both monitors display data for times 13:20:35, 13:20:36, 13:20:37, 13:20:38, and 13:20:39. The values for 13:20:35 and 13:20:36 are 1.00, while for 13:20:37, 13:20:38, and 13:20:39, they are 0.00. A red arrow points from the circled '13:20:38' row in the IDF Monitor to the circled '13:20:38' row in the ODF Monitor. A red circle with a '5' is placed near the arrow. The '13:20:38' rows in both monitors are circled in red.

Time	Value
13:20:35	1.00
13:20:36	1.00
13:20:37	0.00
13:20:38	0.00
13:20:39	0.00

Time	Value
13:20:35	1.00
13:20:36	1.00
13:20:37	0.00
13:20:38	0.00
13:20:39	0.00

## Types for Acceptance Test

- **Self-Test**
- **Mutual-Test**
- **Visual-Test**



# Hsinchu International AI Smart Park

- Hsinchu International AI Smart Park will be managed by Accton (one of the largest ICT company in Taiwan, Revenue US\$ 1.85 Billion in 2020) .
- SpecTalk will be used by Accton to verify all smart applications in the park.

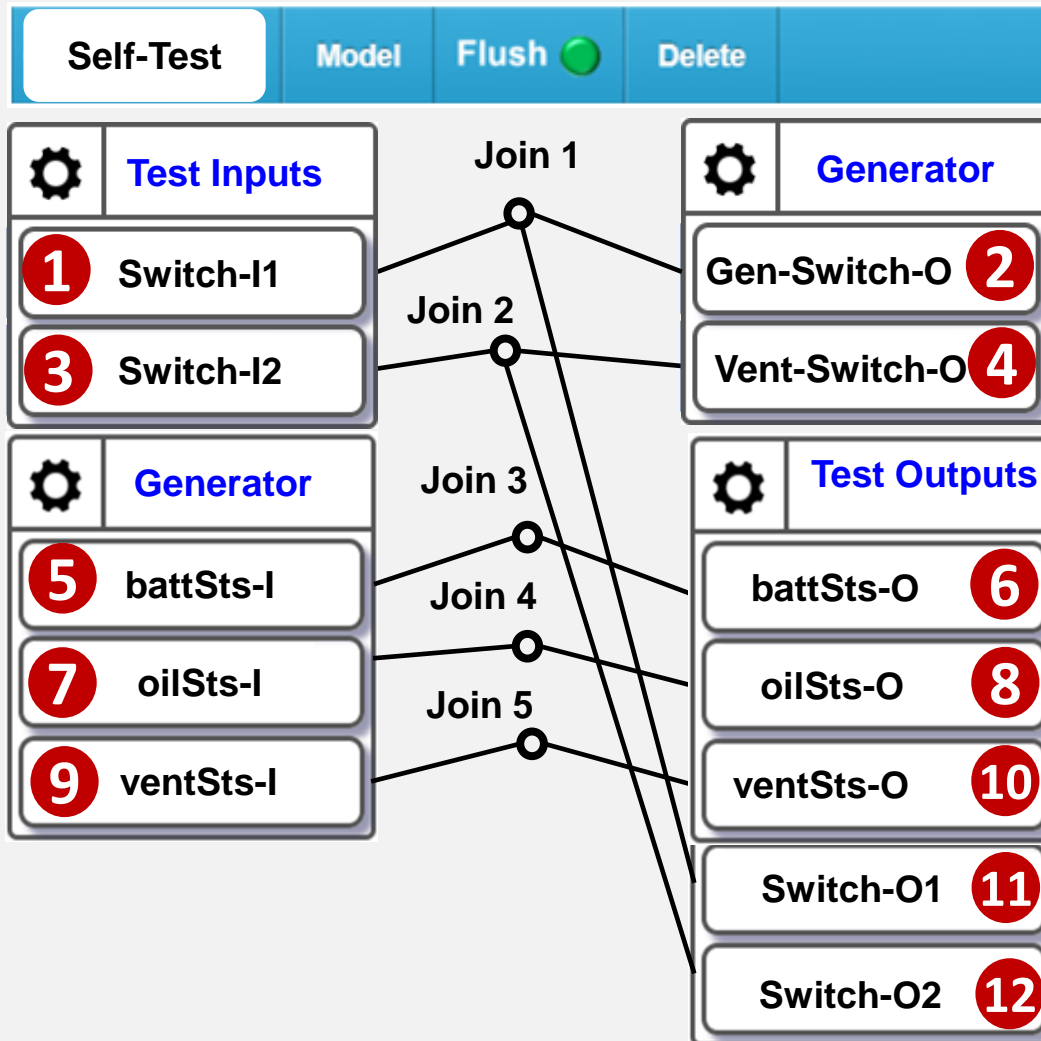


# Self-Test Example: Smart Building of China Medical University in Shuinan Trade and Economic Park, Taichung





# Self-Test Configuration for the Generator in the Smart Building



# The IDFs and ODF for the Water-Cooled Chiller

## Device Feature Window

Type  IDF  ODF Category TS-0022 ▾

DF Name chWF-I **1** ▾

Number of parameters 2

Type	Min	Max	Unit
float ▾	0	10	LPM ▾
float ▾	0	100	Percent ▾

**2**

**3**

## Device Feature Window

Type  IDF  ODF Category TS-0022 ▾

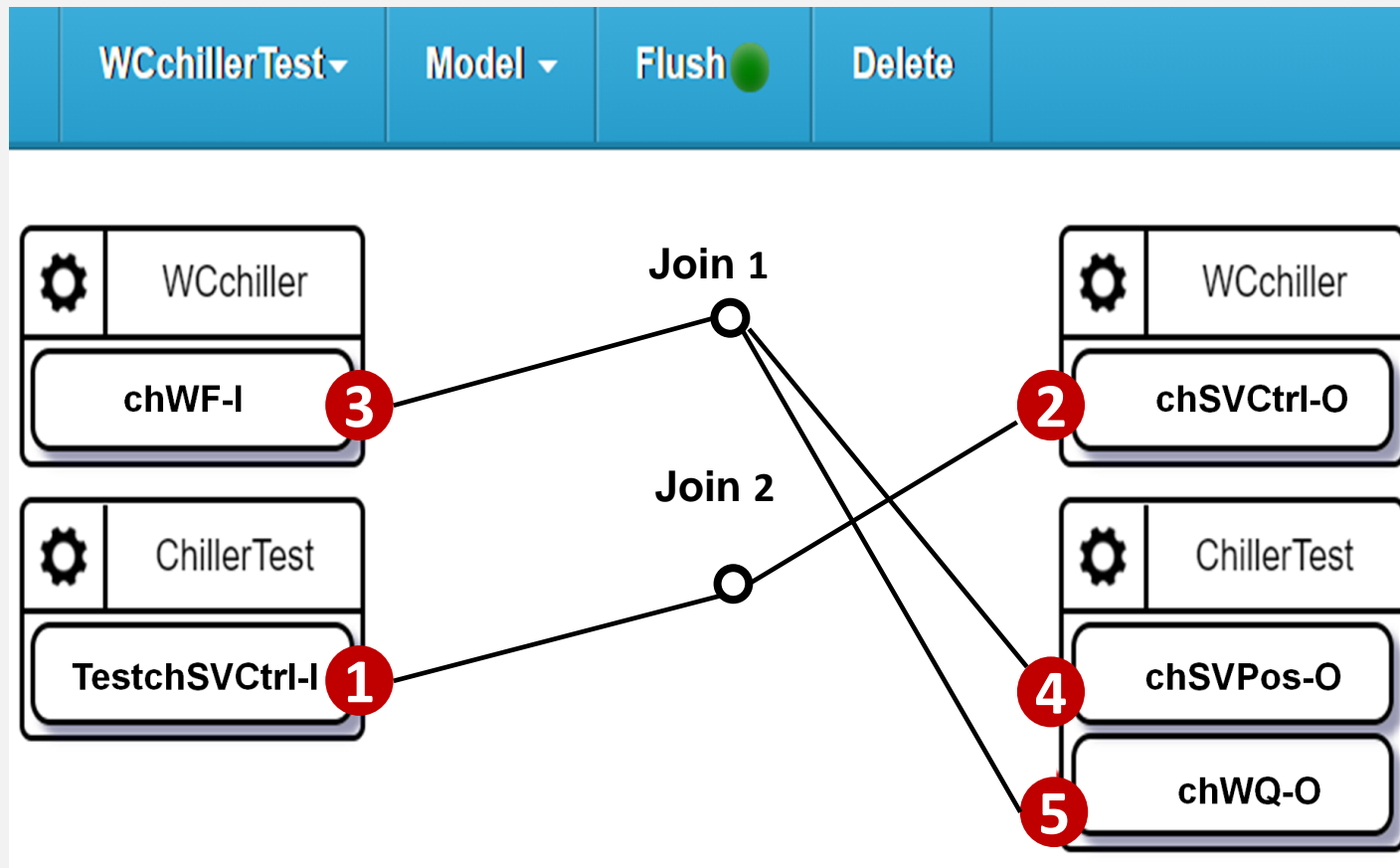
DF Name chSVCtrl-I ▾

Number of parameters 1

Type	Min	Max	Unit
float ▾	0	100	Percent ▾

**4**

# Self-Test Configuration for the Water-Cooled Chiller in the Smart Building



# Mutual-Test for the Window Control

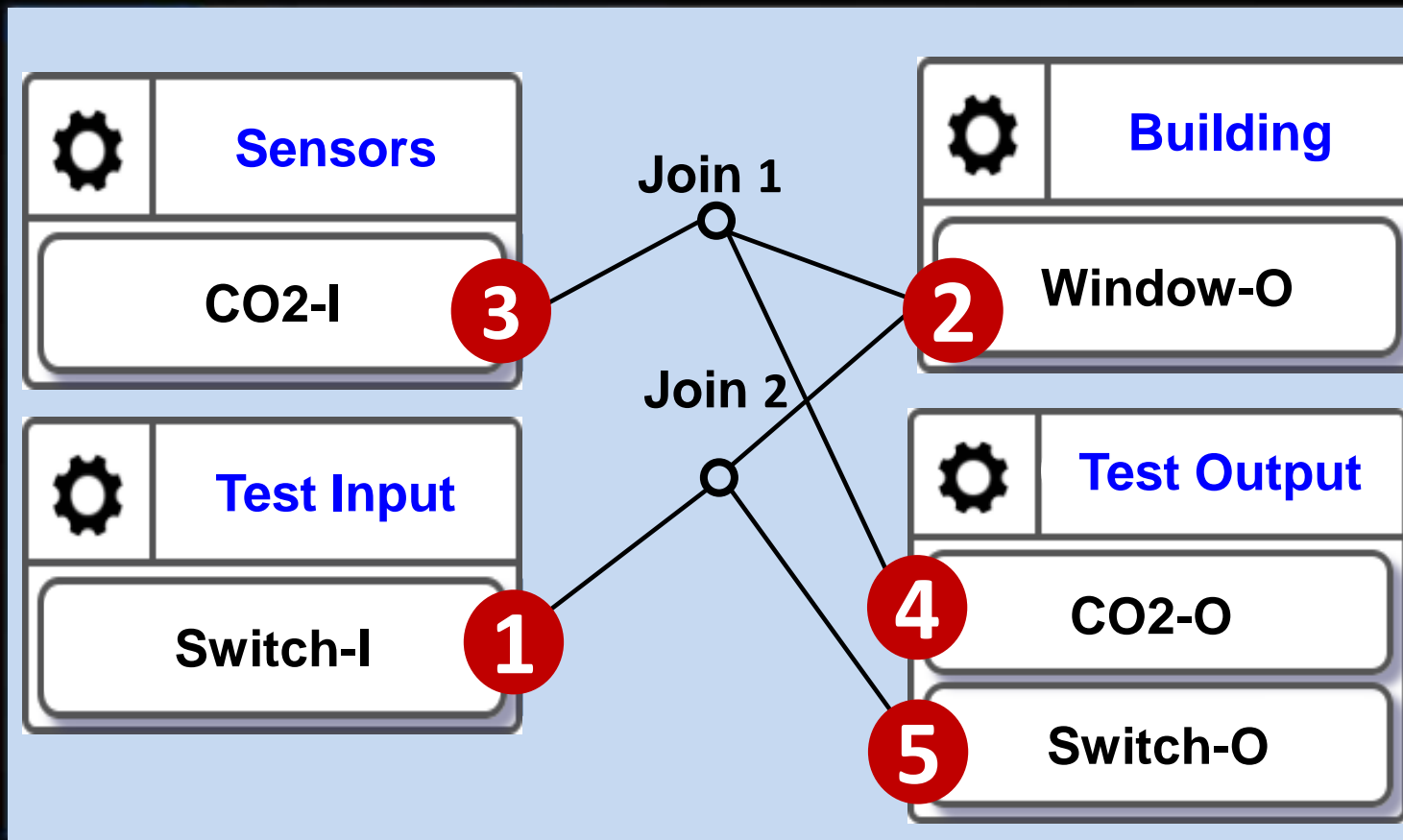


(a) CO<sub>2</sub>, temperature, humidity sensors



(b) Window control

# Mutual-Test Configuration for Window Control

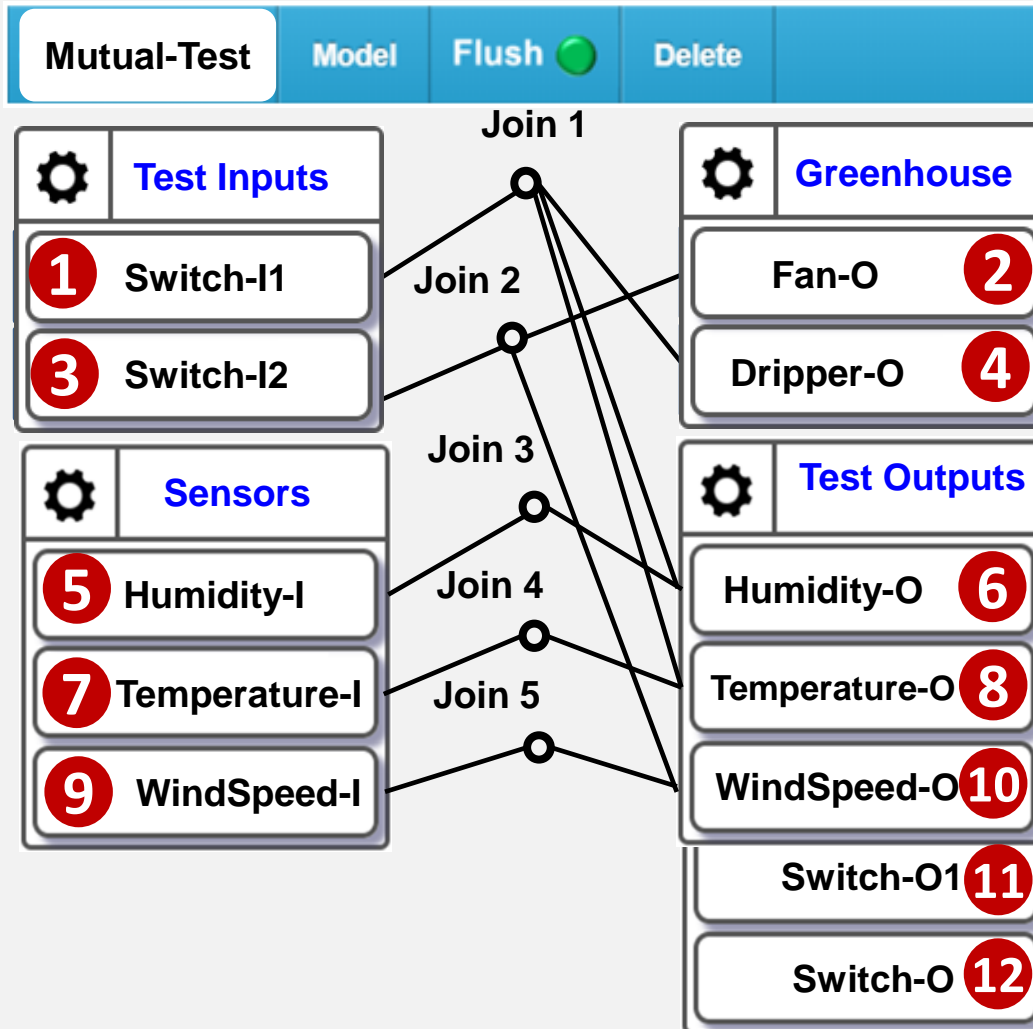


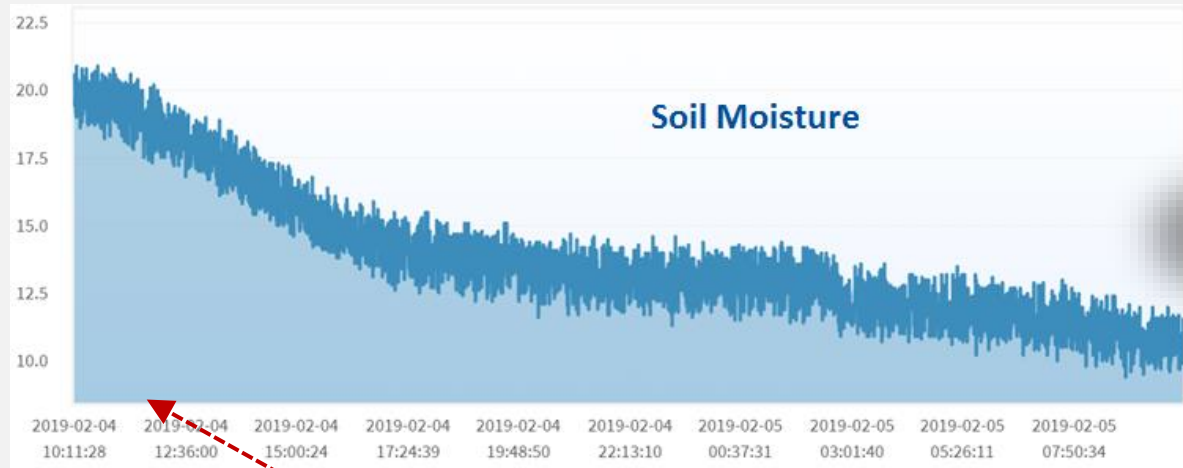


# Mutual-Test Example: Greenhouse in the Bao Mountain, Hsinchu

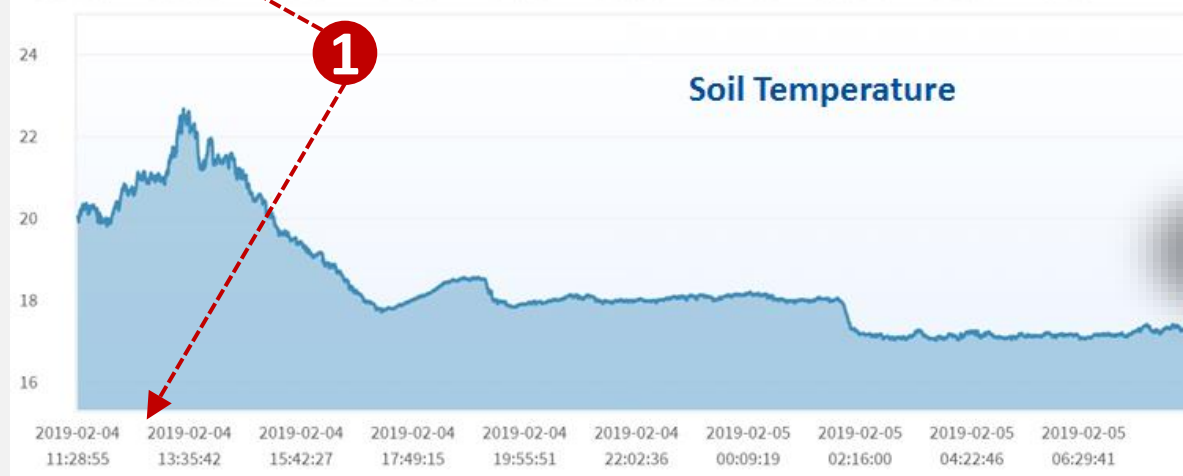


# Mutual-Test Configuration for the Greenhouse





**Fail**



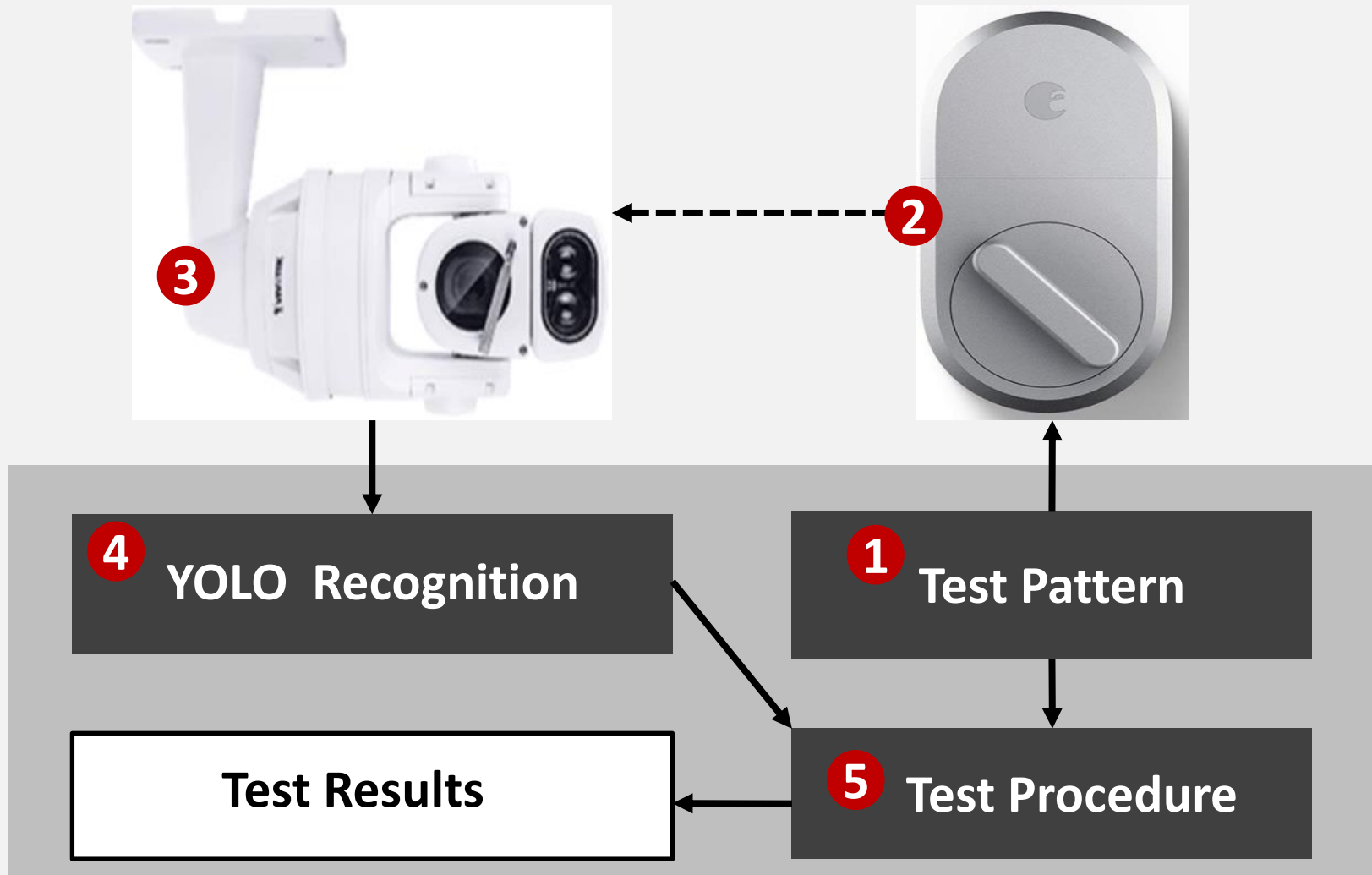
**Pass**



**Pass**



# Visual-Test Example: Door Lock



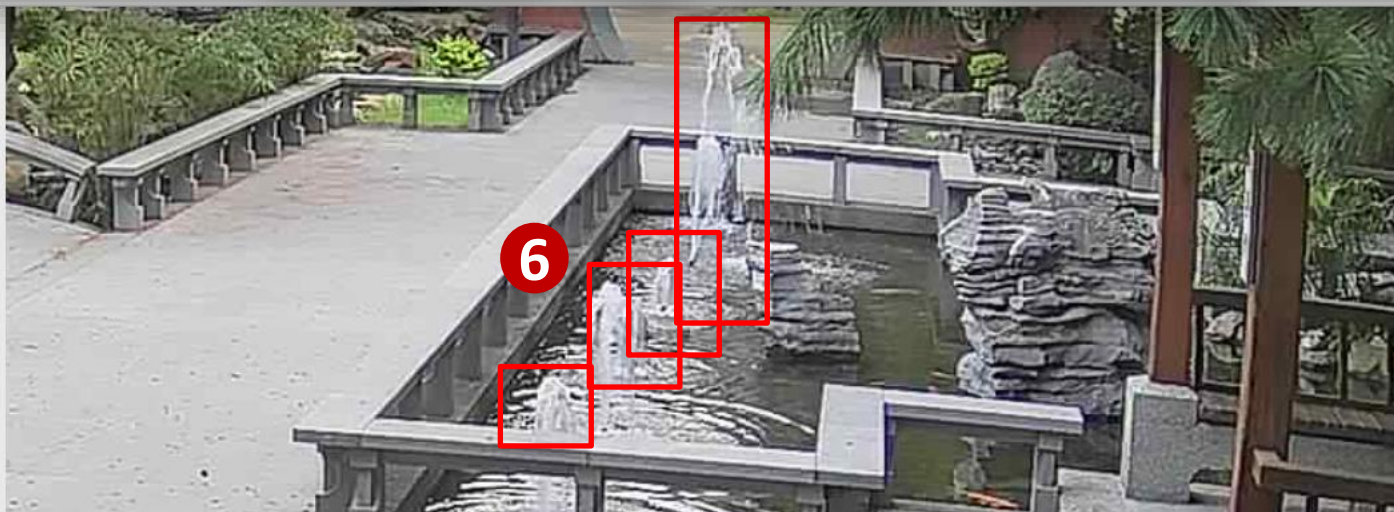


MDHS\_Ctr116

- On 循環馬達
- Off 狀元橋噴
- Off 許願池噴
- Off 瀑布
- Off 晴園立燈
- Off 松道立燈
- Off 涵園立燈
- Off 操場立燈

**Sprinkler**  
**Street Lamp**

## Visual Test Configuration for a Fountain in Ming Der High School, Taichung

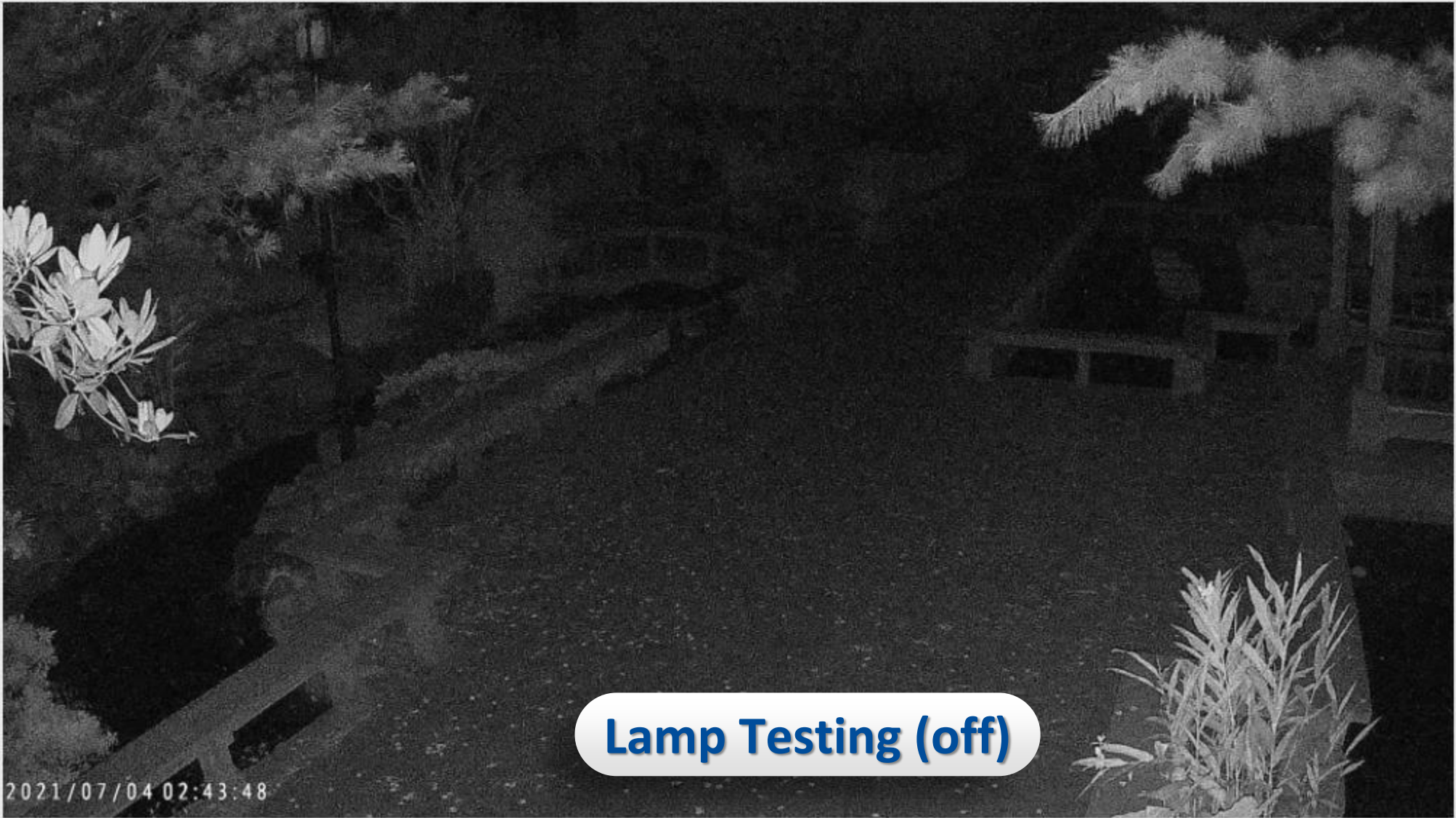


MDHS\_Ctr116

- On 循環馬達
- Off 狀元橋噴泉
- On 許願池噴泉
- Off 瀑布

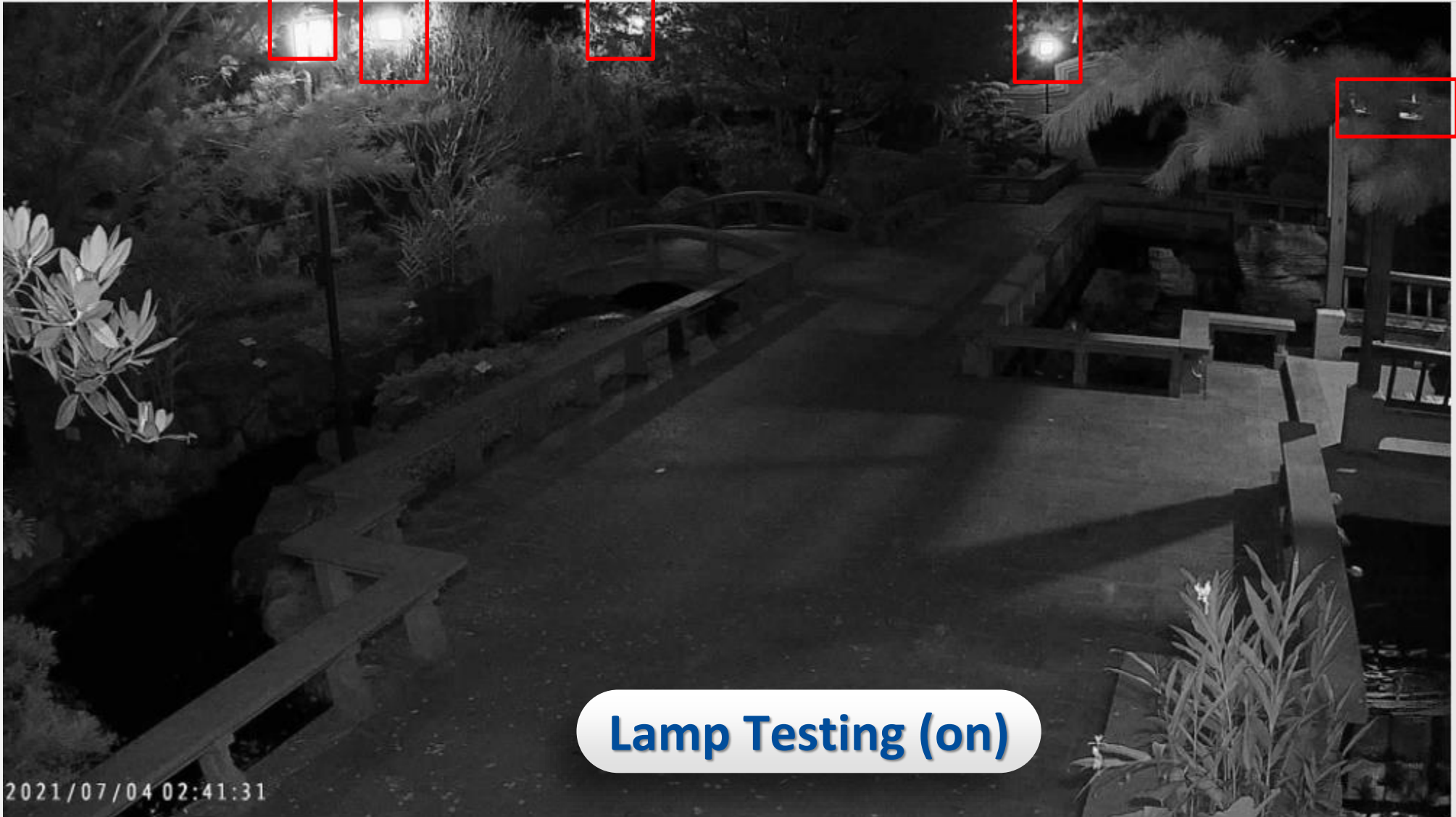
**Sprinkler**





Lamp Testing (off)



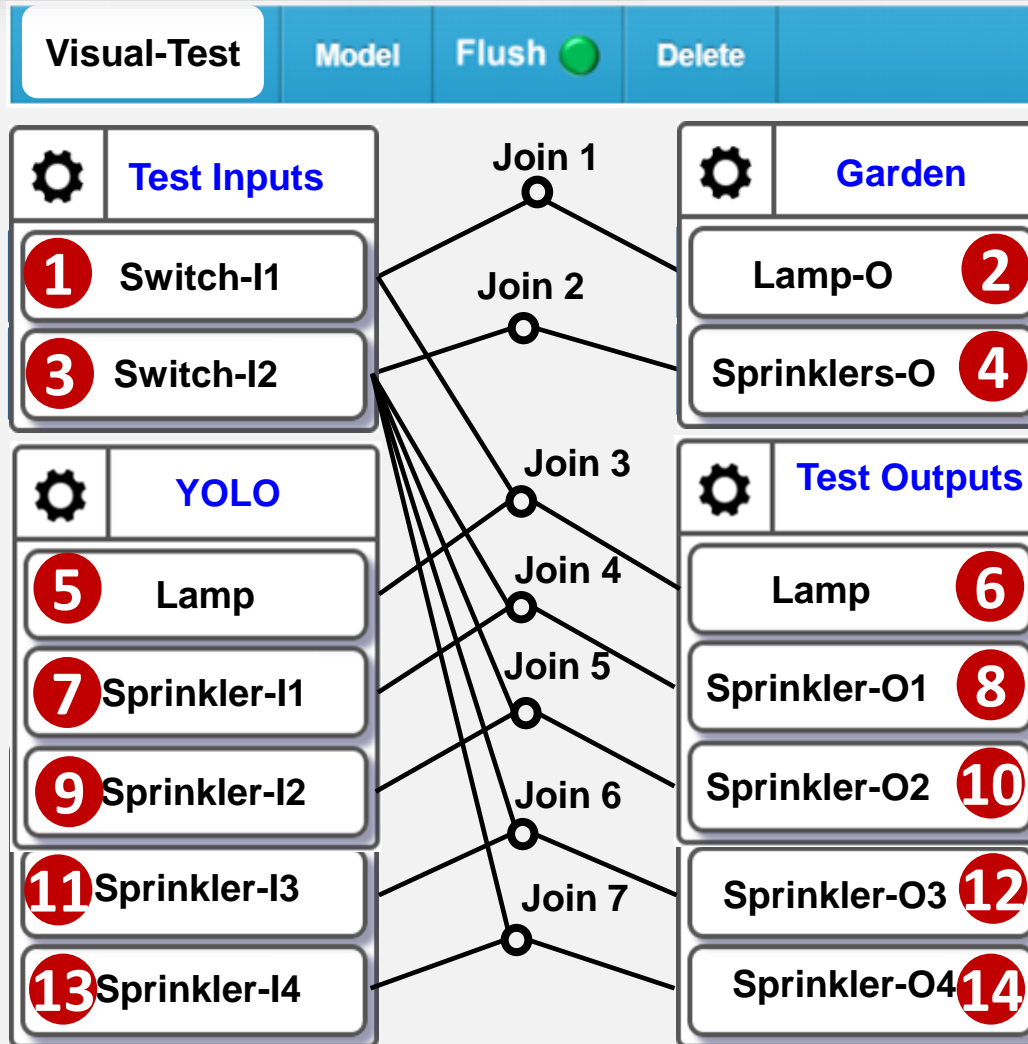


Lamp Testing (on)

2021/07/04 02:41:31

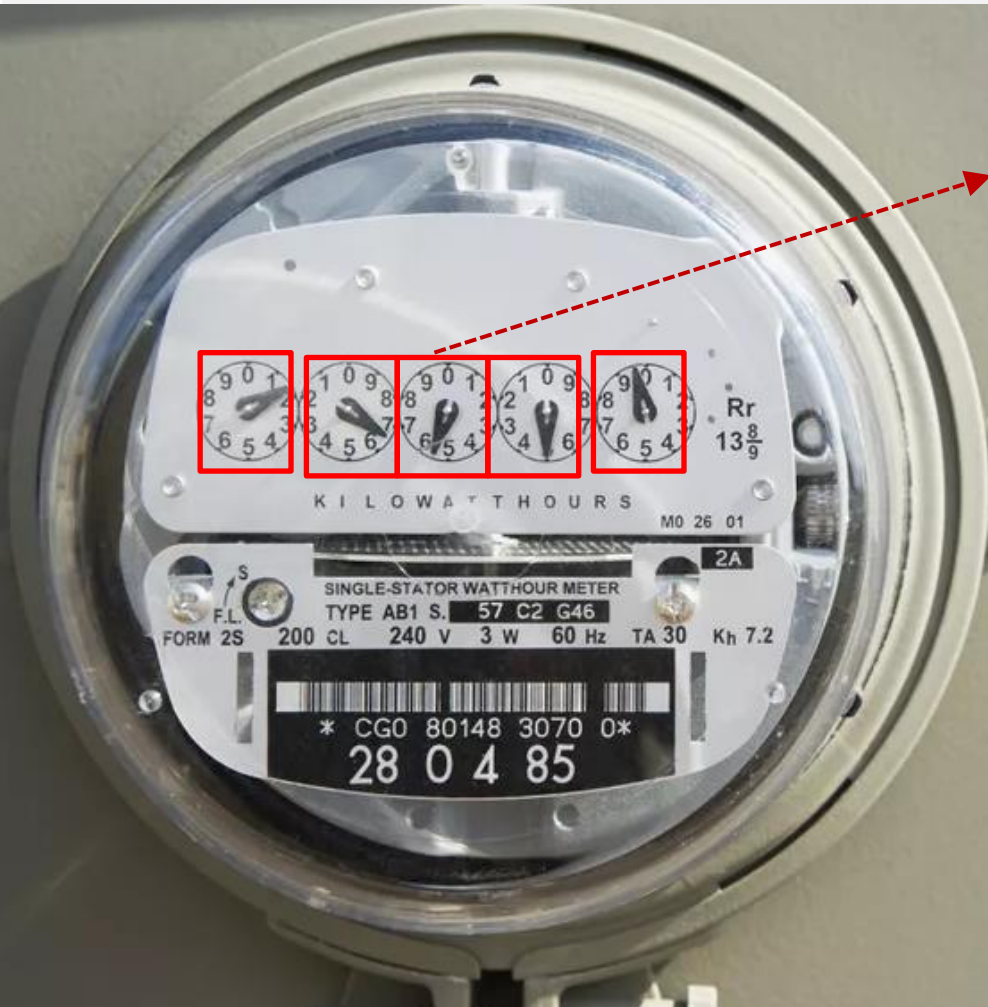


# Visual-Test Configuration for the Fountain in the Ming Der High School





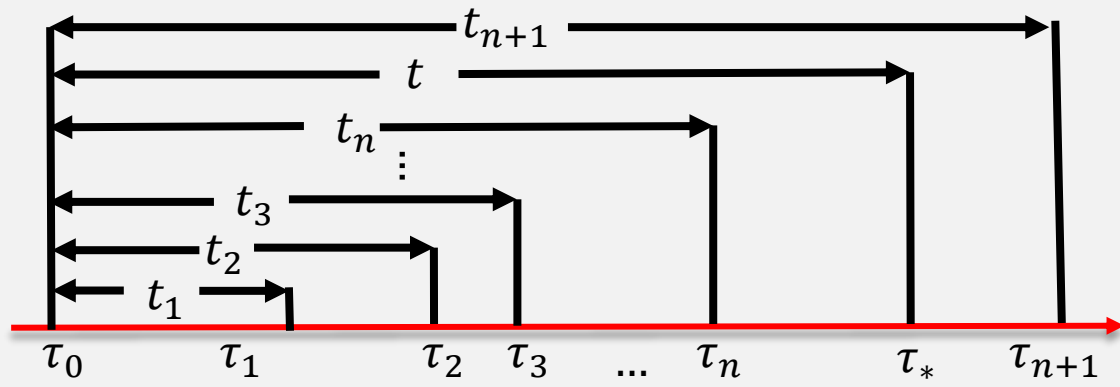
# Visual-Test: Meter Reading



Test Results

YOLO Recognition






## Comments on Visual-Test

- **Lamp test is conducted during nights where the switches are automatically turned on and off.**
- **YOLO already knows the specific spots under test in the picture and therefore can accurately detect the changes on these spots.**
- **We can conduct multiple tests simultaneously in one video screen (for example, lamps and sprinklers in the Ming Der High School)**



# TACIS Partners

- The Telecommunication Technology Committee (TTC), Japan 

- The Telecommunications Industry Association (TIA), USA 

- Institute of Electrical and Electronics Engineers (IEEE) 

- China Communications Standards Association (CCSA), China 

# TACIS Partners

- European Telecommunications Standards Institute (ETSI)



- The Association of Radio Industries and Businesses (ARIB)



- Telecommunications Standards Development Society (TSDSI), India



- Malaysian Technical Standards Forum Bhd (MTSFB), Malaysia



- Telecommunications Technology Association (TTA), Korea



1

# Step 3. Device Application Generation

The screenshot shows a web browser window with the URL `iottalk2.tw/connection`. The browser tabs include 'IoTtalk Homepage', 'IoTtalk', and 'DF Management'. The page header features a navigation bar with buttons: 'Generator-Test', 'Model', 'Flush', 'Delete', 'Simulation' (with a toggle switch), 'Import', 'Export', and 'Logout'. The 'Model' dropdown menu is open, listing several device models: 'Dandelion', 'Dummy\_Device', 'Dummy\_Device\_Tag', 'Fan', 'game', 'Generator', 'Glasses', and 'GPS'. The 'Generator' option is highlighted and circled in red. A red dashed arrow points from the 'DF Management' tab to the 'Generator' option in the dropdown menu. Another red arrow points from the 'IoTtalk' tab to the 'Generator-Test' button. The Windows taskbar on the left shows the time as 08:24 on Tuesday, 2021/6/29.

- Dummy\_Device\_Tag
- Fan
- game
- Generator** 5
- Glasses
- GPS

oilSts 1

ventSts

Output De

**Save** 7

**Generator-Test** **Model** **Flush** **Delete**

**Generator**

battSts-I

oilSts-I

**ventSts-I** 9

Generator				<b>10</b> Simulation	
ventSts	Type	Function	Min		
x1	sample	disable	0		

**Save**

**Generator**

<b>Name</b>	<input type="text" value="Your-Device-Name"/>	Line 8
<b>API</b>	<input type="text" value="Your-API"/>	Line 5
<b>oilSts</b>	<input type="text" value="Your-oilSts-function"/>	Line 12
<b>battSts</b>	<input type="text" value="Your-battSts-function"/>	Line 14
<b>ventSts</b>	<input type="text" value="Your-ventSts-function"/>	Line 16
<b>File</b>	<input type="text" value="File-for-SA-code"/>	3

**1** Device Application Generation

**2** SA Interface



## Generator

<b>Name</b>	<b>Your-Device-Name</b>	<b>Line 8</b>
<b>API</b>	<b>Your-API</b>	<b>Line 5</b>
<b>oilSts</b>	<b>Your-oilSts-function</b>	<b>Line 12</b>
<b>battSts</b>	<b>Your-battSts-function</b>	<b>Line 14</b>
<b>ventSts</b>	<b>Your-ventSts-function</b>	<b>Line 16</b>
<b>File</b>	<b>File-for-SA-code</b>	

Save
Delete

```

01. import time, random, requests↵
02. import DA↵
03. import Your-API↵
04. ServerURL = 'https://DomainName' ↵
05. Reg_addr = "GEN-0012"↵
06. DA.profile['dm_name']='Generator'↵
07. DA.profile['df_list']=['oilSts', 'battSts', 'ventSts',]↵
08. DA.profile['d_name']='Your-Device-Name' ↵
09. DA.register(ServerURL, Reg_addr)↵
10. while True:↵
11.     try:↵
12.         oilSts_data = Your-oilSts-function ↵
13.         DA.push('oilSts', oilSts_data) ↵
14.         battSts_data = Your-battSts-function ↵
15.         DA.push('battSts', battSts_data)↵
16.         ventSts_data = Your-ventSts-function ↵
17.         DA.push('ventSts', ventSts_data)↵
18.     except Exception as e:↵
19.         print(e)↵
20.     if str(e).find('mac_addr not found:') != -1:↵
21.         print('Reg_addr is not found. Try to re-register.')↵
22.         DA.register (ServerURL, Reg_addr)↵
23.     else:↵
24.         print('Connection fails.')↵
25.         time.sleep(1) ↵
26.     time.sleep(0.2)↵

```



# Step 5. Simulate the Generator

Generator-Test Model Flush Delete Simulation Import Export Logout

Generator

- battSts
- oilSts
- ventSts

Join2

Display

Controller

ODF Monitor

Sub-stage: Input

Timestamp	Value
13:20:35	1.00
13:20:36	1.00
13:20:37	0.00
13:20:38	0.00
13:20:39	0.00

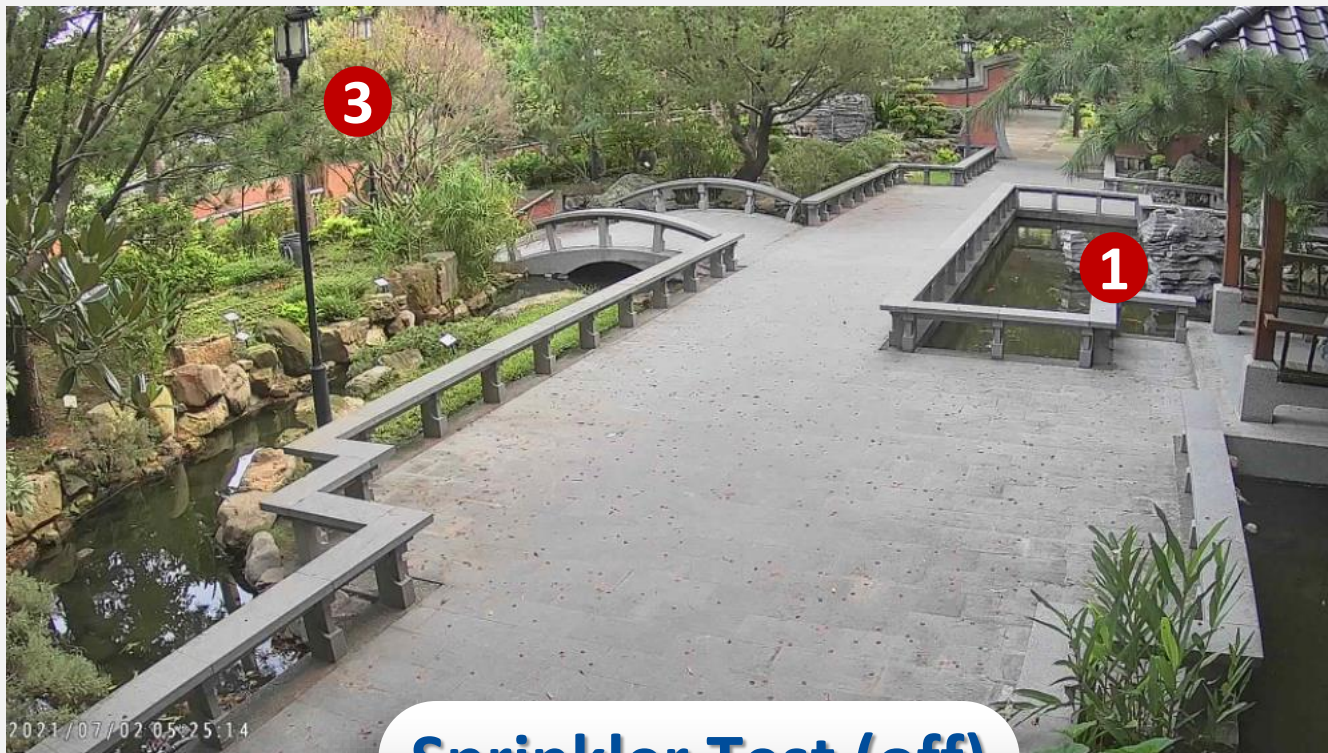
ODF Monitor

Sub-stage: Function

Timestamp	Value
13:20:35	1.00
13:20:36	1.00
13:20:37	0.00
13:20:38	0.00
13:20:39	0.00

Controller

Timestamp	y1
-----------	----



Sprinkler Test (off)



MDHS\_Ctr116

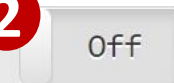


循環馬達



狀元橋噴

2



許願池噴

Sprinkler



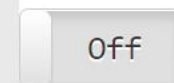
瀑布

4

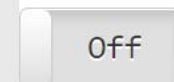


晴園立燈

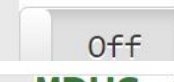
Street Lamp



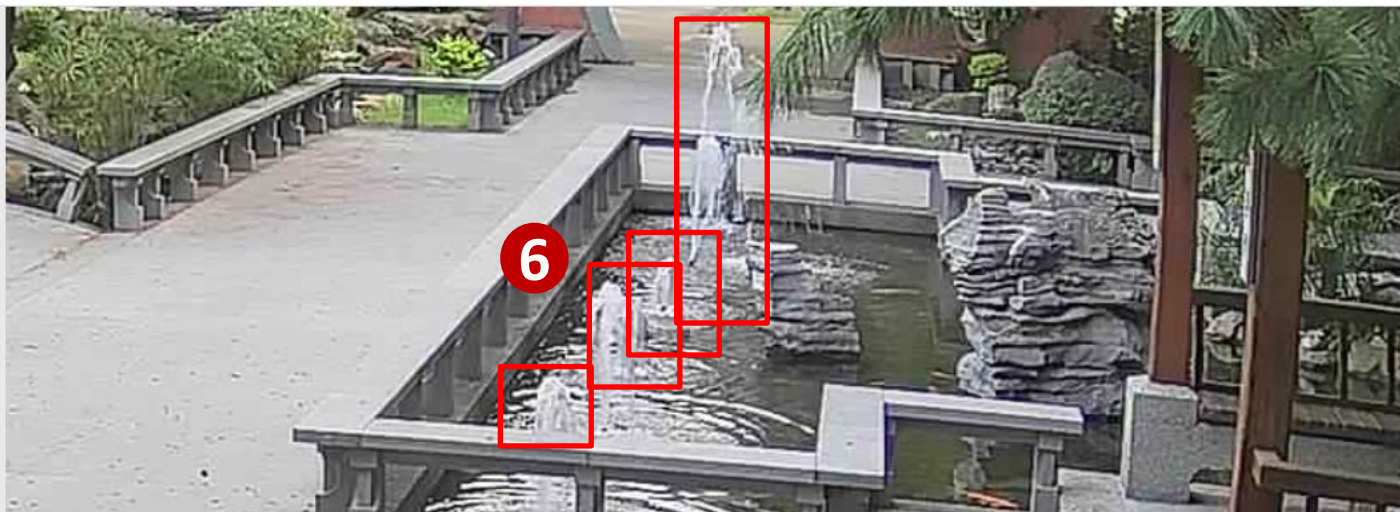
松道立燈



涵園立燈



操場立燈



6

MDHS\_Ctr116



循環馬達



狀元橋噴泉

5



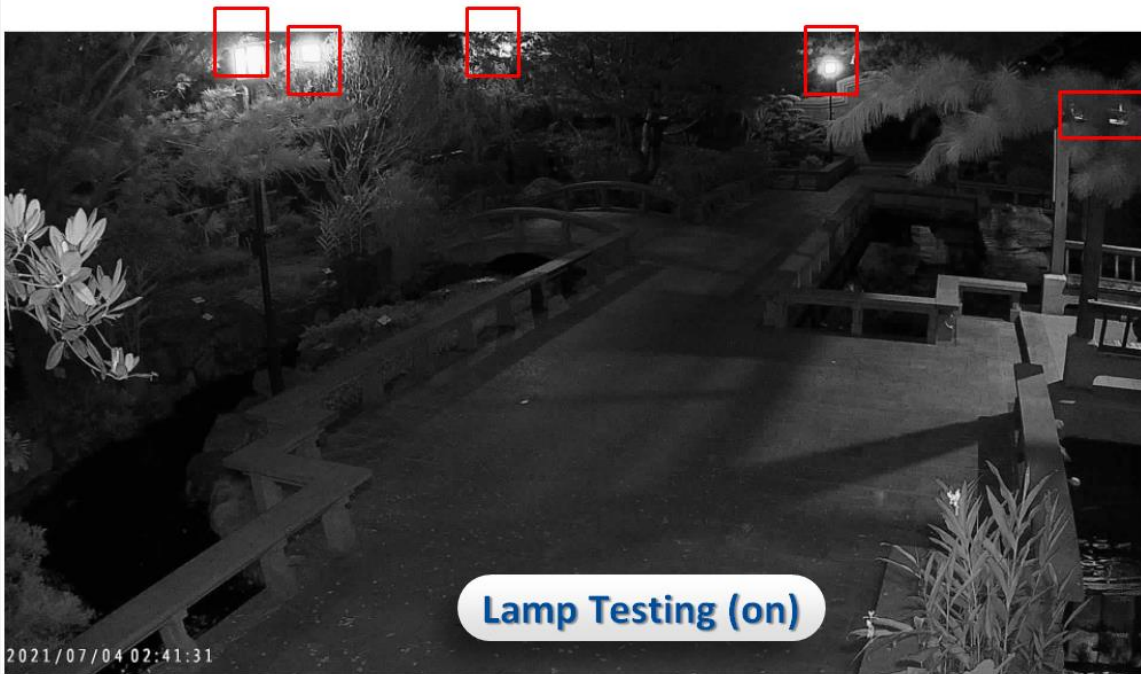
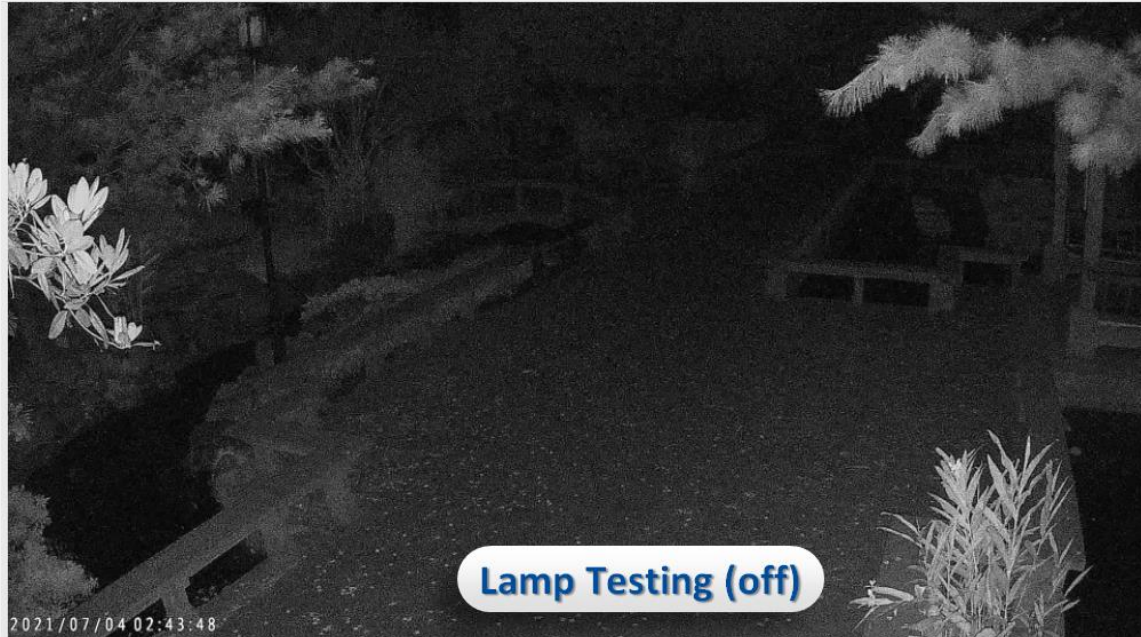
許願池噴泉

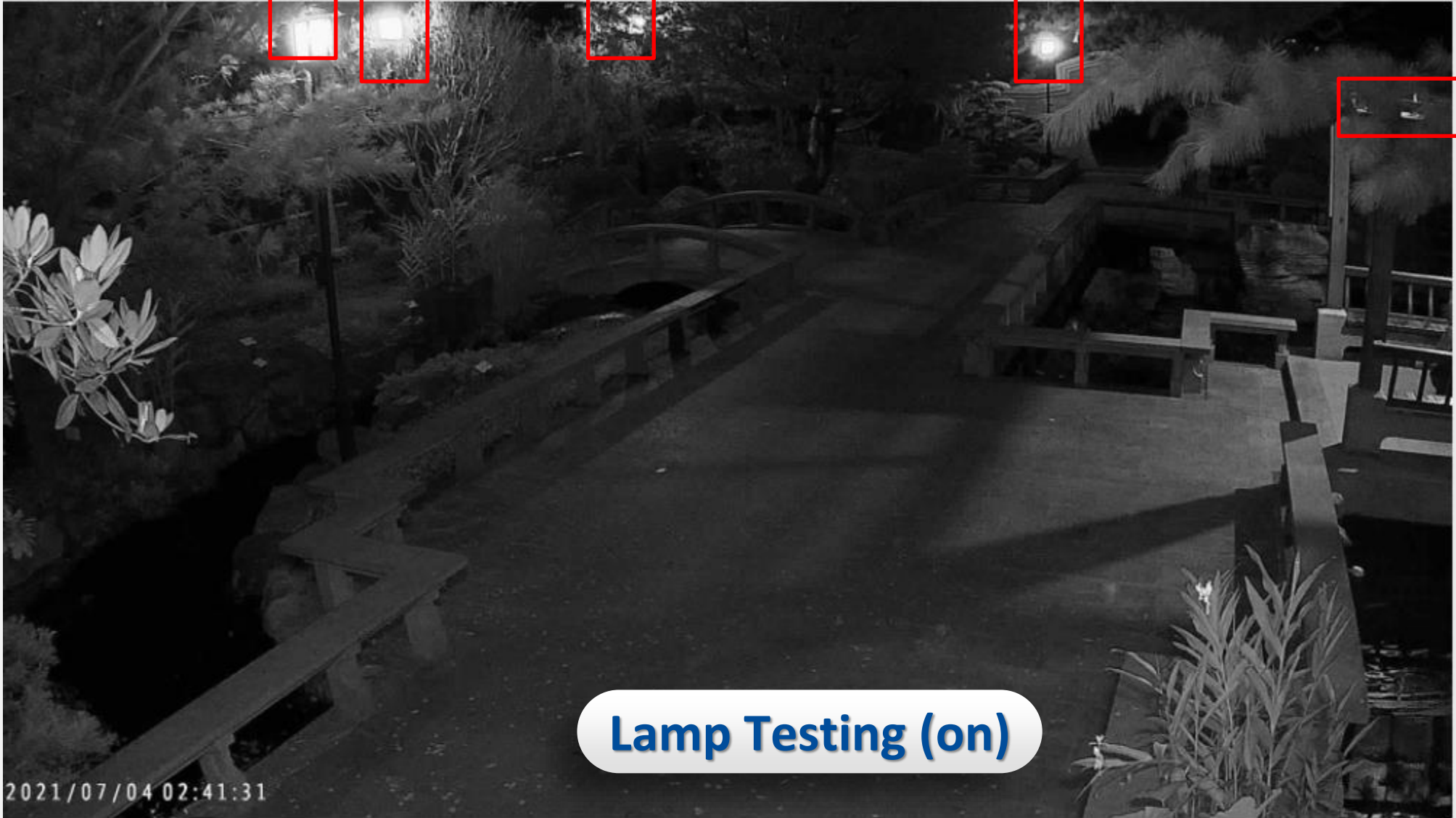
Sprinkler



瀑布







Lamp Testing (on)

2021/07/04 02:41:31





**Generator units**