



Lab Week12

• 2016/11/29

• 周君哲



NTU Iox Center



Wudevice



NTU Iox Center



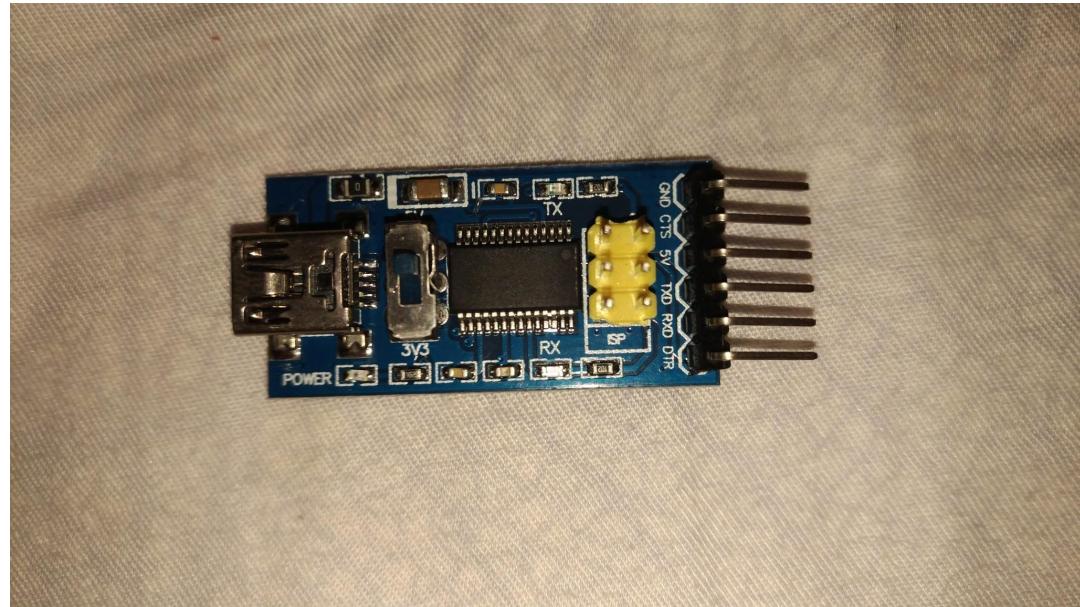
Wudevice

- AVR Atmega2560
 - CPU: 8-bit AVR
 - Flash: 256KB
 - RAM: 8KB
 - EEPROM: 4KB
- Zwave module
- Arduino compatible IO Pins



FTDI connector

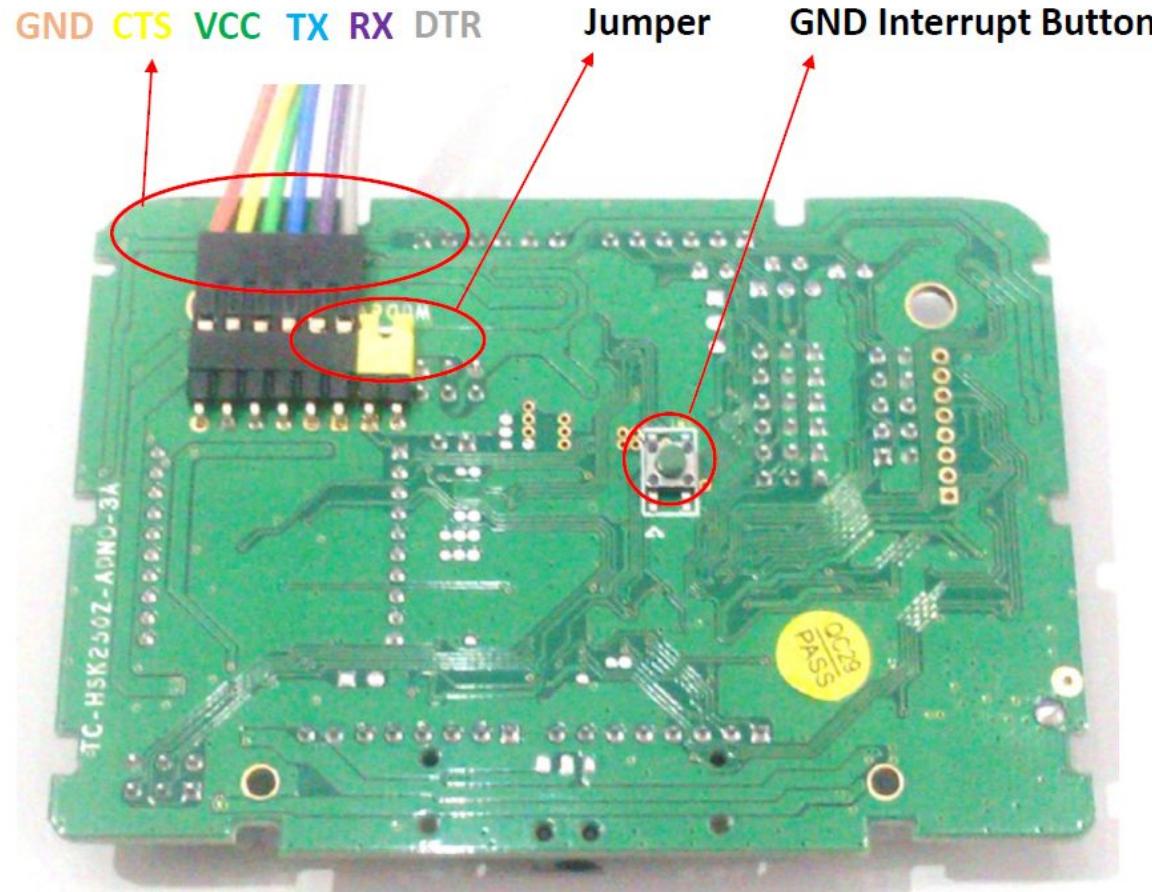
Connect your PC and Wudevice.



NTU lox Center



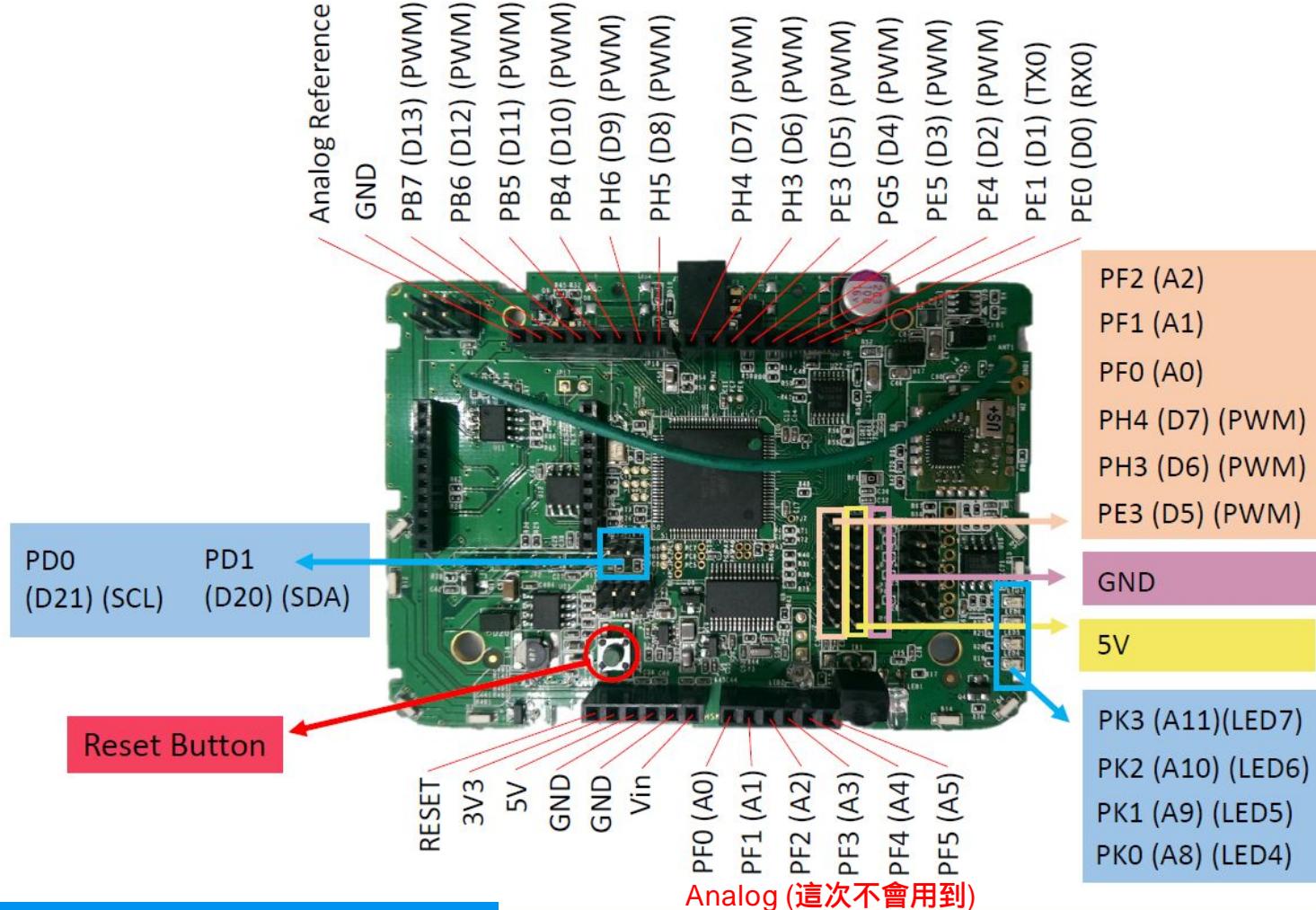
Connect Wudevice with FTDI Connector



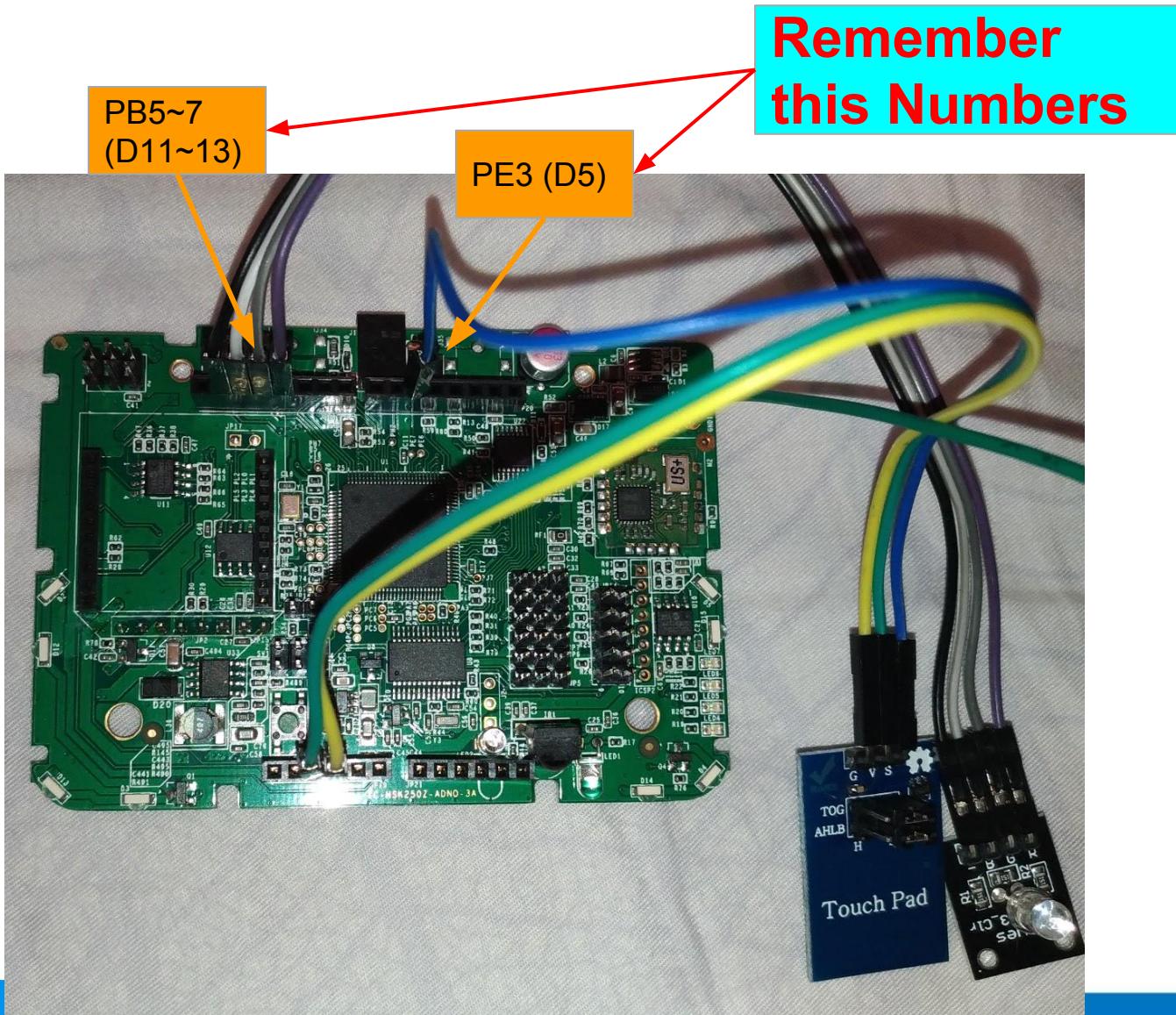
NTU Iox Center



Wudevice layout



Plug in Sensor and Actuator



Connect Your PC and Wudevice

Use this command to check your Wudevice's device number.

```
ls /dev/ttyUSB*
```



Zwave dongle

The wireless radio we use on the Wudevice. Use it to communicate between PC and Wudevice.

After you plug in the dongle. You can use this command to check your dongle's device number.

`ls /dev/ttyACM*` 0 : 就是你安裝的裝置



Install Toolchain

- sudo apt-get install avrdude gcc-avr avr-libc binutils-avr
- sudo adduser <your username> dialout #get the permission of serial ports

And other tools we have already install in:

https://wukongsun.gitbooks.io/wukong-release-0-4/content/Ch2/Ch2_WuKong_Environment_Setup.html



Definition of Wuclass

Here is the two Wuclass we will modify in this lab.

一般on/off的sensor

```
<WuClass name="Binary_Sensor" id="1004" virtual="false" type="hard" privateCData="short">
    <property name="current_value" access="readwrite" datatype="boolean" default="false" />
    <property name="refresh_rate" access="readwrite" datatype="refresh_rate" default="10" />
</WuClass>
```

```
<WuClass name="Light_Actuator" id="2001" virtual="false" type="hard">
    <property name="on_off" access="readwrite" datatype="boolean" default="false" />
</WuClass>
```



Implement Wuclasses in C

寫C code

Digital I/O:

DDRX -> The direction of the pins

PINX -> The input value of the pins

PORTX -> The output value of pins

EX: touch pad (binary sensor) on PE3 (D5)

```
DDRE &= ~(1<<3); //set the third bit of DDRE to 0
```

```
value = PINE & (1<<3); //get the value on the third bit of PINE
```

ref: <https://www.arduino.cc/en/Reference/PortManipulation>



NTU lox Center



Ex: Binary Sensor

https://github.com/wukong-m2m/wukong-darjeeling/blob/release0.4/src/lib/wkpf/c/arduino.wudevice/native_wuclasses/wuclass_binary_sensor_update.c

```
#define set_input(portdir, pin) portdir &= ~(1<<pin)
#define output_high(port, pin) port |= (1<<pin)
#define input_get(port, pin) ((port & (1 << pin)) != 0)

void wuclass_binary_sensor_setup(wuobject_t *wuobject) {
    DEBUG_LOG(DBG_WKPFUPDATE, "WKPFUPDATE(BinarySensor): setup\n");
    set_input(DDRE, 3);
    output_high(PINE, 3);
}

void wuclass_binary_sensor_update(wuobject_t *wuobject) {
    bool currentValue = 0;
    currentValue = input_get(PINE, 3);          Read the value from the
                                                third pin of port E.
                                                Set the sensor value to
                                                the property.
    wkpf_internal_write_property_boolean(wuobject, WKPF_PROPERTY_BINARY_SENSOR_CURRENT_VALUE, currentValue);

    DEBUG_LOG(DBG_WKPFUPDATE, "WKPFUPDATE(BinarySensor): Sensed binary value: %d\n", currentValue);
}
```



Light Actuator

https://github.com/wukong-m2m/wukong-darjeeling/blob/release0.4/src/lib/wkpf/c/arduino.wudevice/native_wuclasses/wuclass_light_actuator_update.c

```
void wuclass_light_actuator_setup(wuobject_t *wuobject) {}

void wuclass_light_actuator_update(wuobject_t *wuobject) {
    bool onOff;
    wkpf_internal_read_property_boolean(wuobject, WKPF_PROPERTY_LIGHT_ACTUATOR_ON_OFF, &onOff);

    // Connect light to port B, bit 4. This maps to pin 3 of JP18 on the WuNode (pin 1 is behind the power connector)
    // Port B, bit 5 (pin 4 on JP18) will reversed: low for on, high for off, for connecting LEDs
    // SETOUPUT
    DDRB |= _BV(4);
    DDRB |= _BV(5);
    DDRB |= _BV(7);
    if (onOff)
        PORTB |= _BV(4);
    else
        PORTB &= ~_BV(4);
    if (onOff)
        PORTB &= ~_BV(5);
    else
        PORTB |= _BV(5);
    if (onOff)
        PORTB |= _BV(7);
    else
        PORTB &= ~_BV(7);
    DEBUG_LOG(DBG_WKPFUPDATE, "WKPFUPDATE(Light): Setting light to: %x\n", onoff);
}
```

$_BV(x) = (1 << x)$
Set PB4 (D10), PB5 (D11), PB7 (D13) as output

If onOff is true set the fourth and seventh pin of port B to high and the fifth pin to low.



Modify Light Actuator Wuclass

這次要做的事情是要改Light Actuator

Make a Wuclass that read the value from a binary sensor and switch to the eight different RGB LED lights (from all dark to all light).

1. Keep the value from “on_off” property
2. If the value change from 0 to 1, change the LED light.
3. The pattern of LED should be 000 -> 001 -> 010 -> ... -> 111 (red -> green -> yellow -> blue -> ... -> white)



Wudevice Setup

```
cd <wkroot>/src/
```

```
cp usersettings.gradle.dist usersettings.gradle
```

```
vim usersettings.gradle
```

```
# Fill in your device name
```

```
# ex: /dev/ttyUSB0
```

```
1 ext.avrdudeprogrammer='/dev/<fill in your device name here>'
```



Enable Wuclasses

```
cd <wkroot>/src/config/wunode
```

```
vim enabled_wuclasses.xml
```

```
# Fill the wuclasses name we need.
```

```
<WuKong>
```

```
  <WuClass name="Binary_Sensor" appCanCreateInstances="false">
```

```
    <CreateInstance />
```

```
  </WuClass>
```

```
  <WuClass name="Light_Actuator" appCanCreateInstances="false">
```

```
    <CreateInstance />
```

```
  </WuClass>
```

```
</WuKong>
```



Upload Wukong Image

```
cd <wkroot>/src/config/wunode
```

```
gradle -b ../../build.gradle avrdude
```

To check the debug messages from wudevice:

```
sudo screen /dev/ttyUSB0 115200
```

```
Clearing leftovers
```

```
My Zwave node_id: 2  
Darjeeling is go!  
Java get node id: 0.0.0.2
```

If the node id is
255.255.255.X, bring
the device to TA



Run wukong master

Same as we used before.

```
cd <wkroot>/wukong/master
```

```
python master_server.py
```



Reset Zwave Dongle

```
cd <wkroot>/wukong/tools/testrtt
```

```
make testrtt
```

```
sudo ./testrtt -d /dev/ttyACM0    #fill in the dongle's  
device number
```

```
#Use command “controller reset”
```

```
#Ctrl-C after you see the “Z-Wave controller is back to  
factory default”
```



Install Python Zwave Library

```
cd <wkroot>/tools/python/pyzwave
```

```
sudo python setup.py install
```



Config wukong gateway

```
cd <wkroot>/wukong
```

```
cp -r gateway gateway_zwave #Create a new gateway
```

```
cd gateway_zwave
```

```
vim gtwconfig.py #we will use zwave as gateway's  
transportation interface
```

```
# set TRANSPORT_INTERFACE_TYPE = 'zwave'
```

```
# set TRANSPORT_INTERFACE_ADDR = '/dev/ttyACM0'
```

```
python start_gateway.py
```



Add WuDevice

1. Press the “add node” button, and wait until it shows “Ready to include/exclude device”. This message might only show up for few seconds. You can continue even it disappeared.
2. Press the GND interrupt button on the back side of the WuDevice to be discovered

Ready to include/exclude device

#	Location	WuClass	WuObject
1	/WuKong	0	1



Add Wudevice

3. Press the stop button after it shows “Node is found”
4. It should show a message “Add: done, Node id: 2”
5. Rediscovery and check the detail of Wudevice

The screenshot shows the 'Nodes Editor' tab selected in a software interface. A message box displays 'Node is found'. Below it, two buttons are visible: 'Discover Nodes' and 'Stop to complete operation'. A table lists a single node entry:

#	Location	WuClass	WuObject	Details
1	/WuKong	0	1	<button>Details</button>

Buttons for 'Find Location' and 'Set Location' are located at the bottom of the table row.



Device Detail

Wukong Application Management Device Management Application Store

Nodes Editor Location Editor Landmark Editor

Gateway 0.0.0.1(1) tcp_addr=('192.168.4.16', 9001)

Discover Nodes Add Node Remove Node

#	Location
2	/D15
1	/WuKong

Node Info

WuClasses:

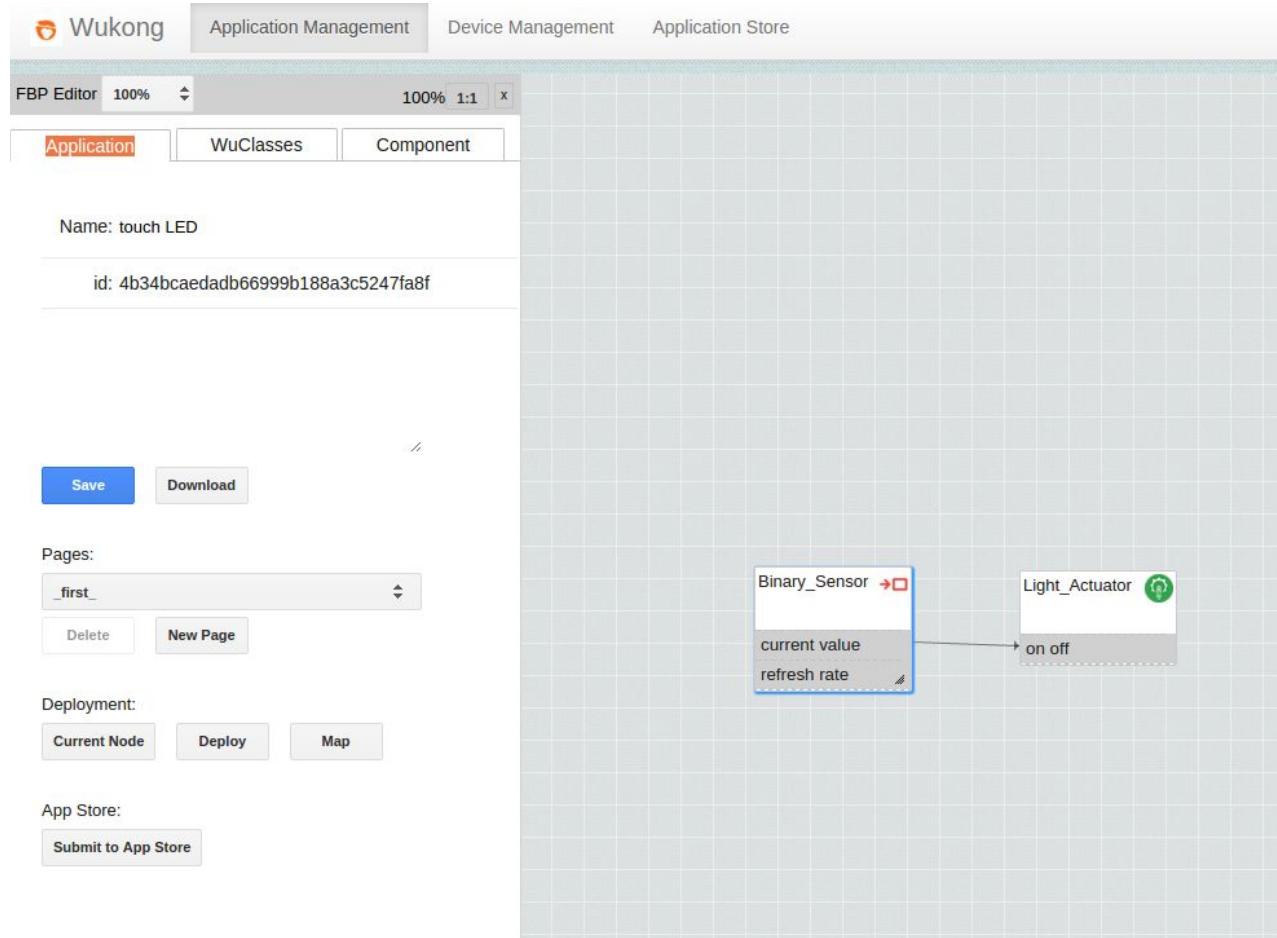
WuObjects:

- Class Name: Binary_Sensor
Port Number: 1
- Class Name: Light_Actuator
Port Number: 2

Close



Make FBP



Map and Deploy

Mapping Progress:

Mapping Results:

115	(Virtual) Binary_Sensor	<192.168.4.16>:2	1
116	(Virtual) Light_Actuator	<192.168.4.16>:2	2
1	(Virtual) Server	<192.168.4.16>:1	1

ukong Application Management Device Management Application Store

100% 100% 1:1 x

```
038:[log] Application has been deployed!
037:[log] ...has completed
036:[log] Deploying to node 2, remaining set({})
035:[log] Preparing to deploy to nodes set([1, 2])
034:[log] Compression finished
033:[log] Compressing application code to bytecode format...
032:[log] Generating java application...
031:[log] Preparing java library code...
030:[log] Deploying to node 2, remaining set({})
029:[log] Preparing to deploy to nodes set([1, 2])
028:[log] Compression finished
027:[log] Compressing application code to bytecode format...
026:[log] Generating java application...
025:[log] Preparing java library code...
024:[log] Deploying to node 2, remaining set({})
023:[log] Preparing to deploy to nodes set([1, 2])
022:[log] Compression finished
021:[log] Compressing application code to bytecode format...
020:[log] Generating java application...
019:[log] Preparing java library code...
```



Run the Result

<https://drive.google.com/open?id=0B7bbdsnbCVsvczVndnIUUGMtdXc>



Hand in

1. A video of your result.
2. Your code of light actuator wuclass.



NTU Iox Center

