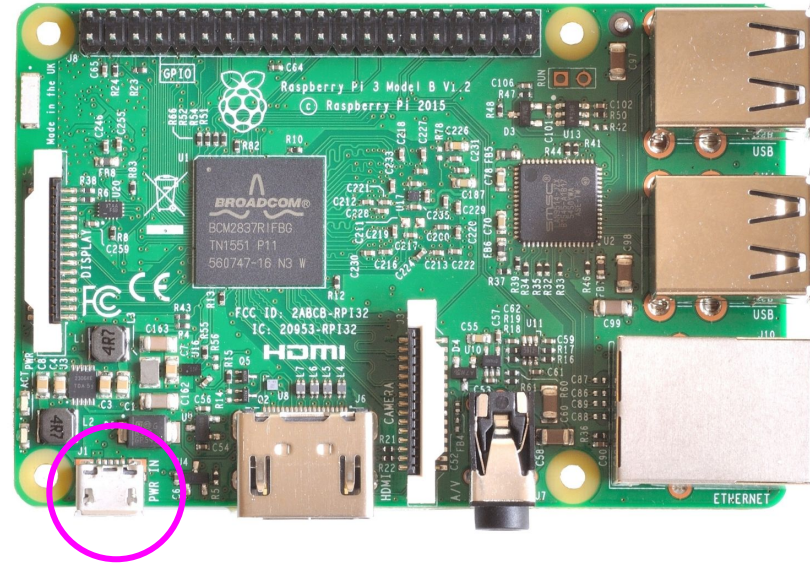
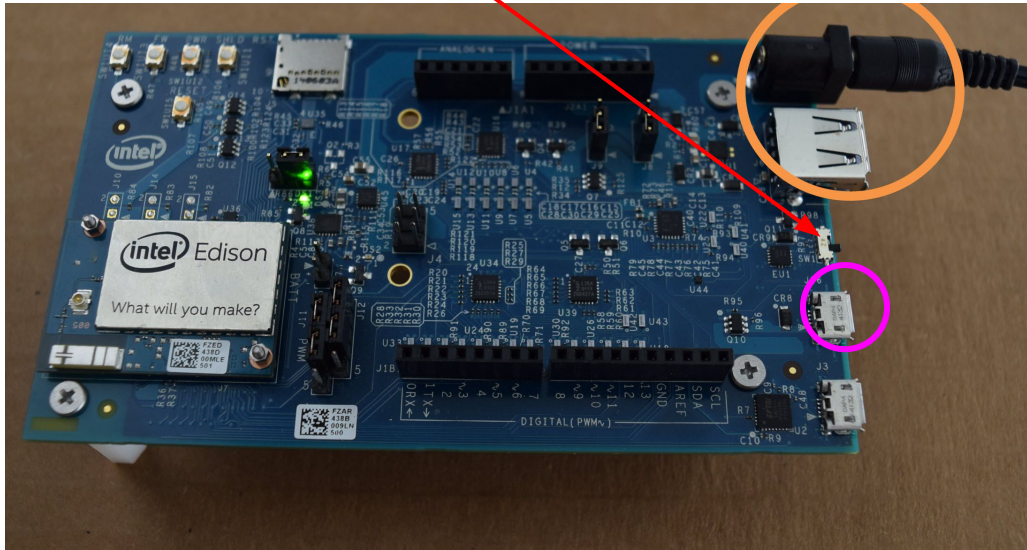


Plug in the Power Supply

Switch to power cable or micro usb cable



7-15V DC/ At least 1500mA

Micro USB cable

Sensor and Actuator Connection for Raspberry Pi 3

Physical Pin

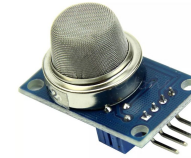


Alternate Function		Physical Pin		Physical Pin		Alternate Function
	3.3V PWR	1		2	5V PWR	
I2C1 SDA	GPIO 2	3		4	5V PWR	
I2C1 SCL	GPIO 3	5		6	GND	
	GPIO 4	7		8	UART0 TX	
	GND	9		10	UART0 RX	
	GPIO 17	11		12	GPIO 18	
	GPIO 27	13		14	GND	
	GPIO 22	15		16	GPIO 23	
	3.3V PWR	17		18	GPIO 24	
SPI0 MOSI	GPIO 10	19		20	GND	
SPI0 MISO	GPIO 9	21		22	GPIO 25	
SPI0 SCLK	GPIO 11	23		24	GPIO 8	SPI0 CS0
	GND	25		26	GPIO 7	SPI0 CS1
	Reserved	27		28	Reserved	
	GPIO 5	29		30	GND	
	GPIO 6	31		32	GPIO 12	
	GPIO 13	33		34	GND	
SPI1 MISO	GPIO 19	35		36	GPIO 16	SPI1 CS0
	GPIO 26	37		38	GPIO 20	SPI1 MOSI
	GND	39		40	GPIO 21	SPI1 SCLK

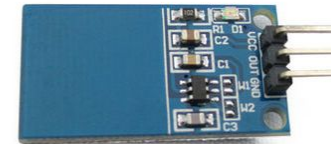
Physical Pin -> Function



2 -> +5V
23 -> CI
19 -> DI
20 -> GND

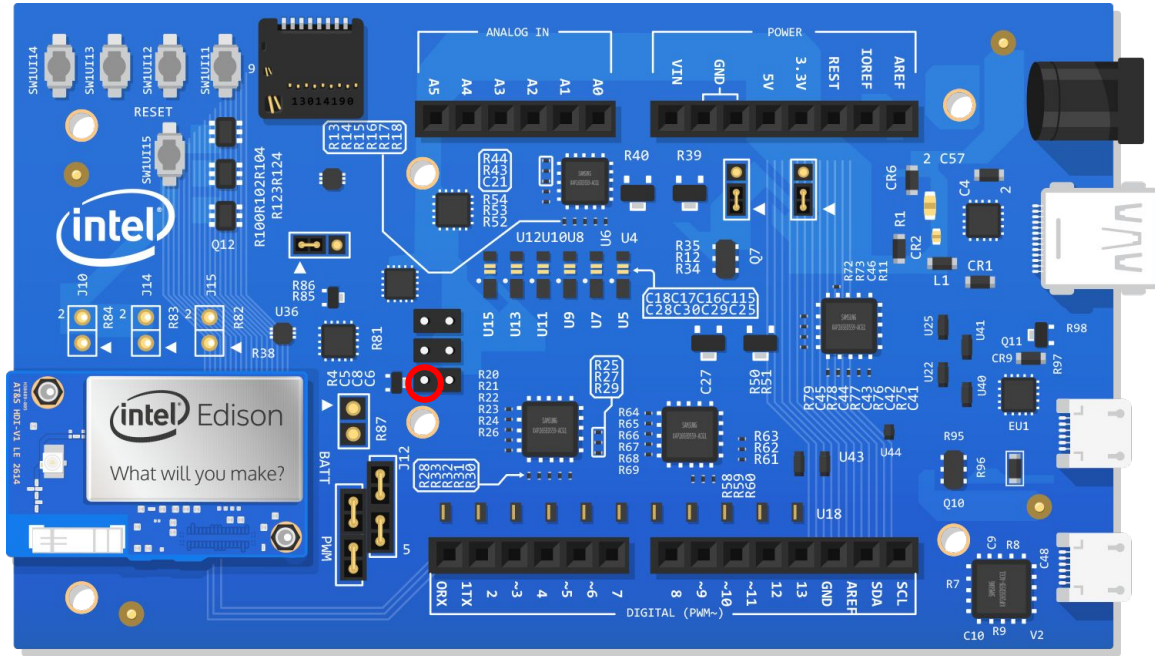


4 -> Vcc
12 -> DO
14 -> GND

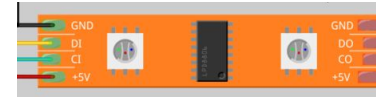


1 -> V
3 -> Signal
9 -> GND

Sensor and Actuator Connection for Intel Edison Board



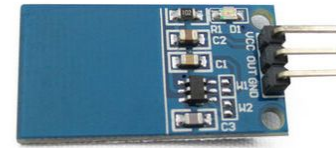
Board Label -> Sensor or Actuator label



5V -> +5V
13 -> CI
11 -> DI
GND -> GND



Circle -> Vcc
12 -> DO
GND -> GND



3.3V -> V
3 -> Signal
GND -> GND

Remove Cache File

- This instruction can be found in the chapter 4.1.5 of Gitbook.
- Remove the compiled Python files, JSON files, and also the changeset, which is a history record of deployment

```
cd ~/wukong-darjeeling/wukong/master  
rm *.pyc *.json change*
```
- Remove the compiled Python files, JSON files, and the devices.pkl, which is a file that includes the discovery result for gateway program.

```
cd <path_source_code>/wukong-darjeeling/wukong/gateway  
rm *.pyc *.json device*
```
- Remove the compiled Python files and the JSON, which includes detailed information of each device.

```
cd <path_source_code>/wukong-darjeeling/wukong/gateway/udpwpkpf  
rm *.pyc *.json
```

Change Branch and Update Source Code

- Please checkout to workshop branch

```
cd ~/wukong-darjeeling
```

```
git checkout workshop
```

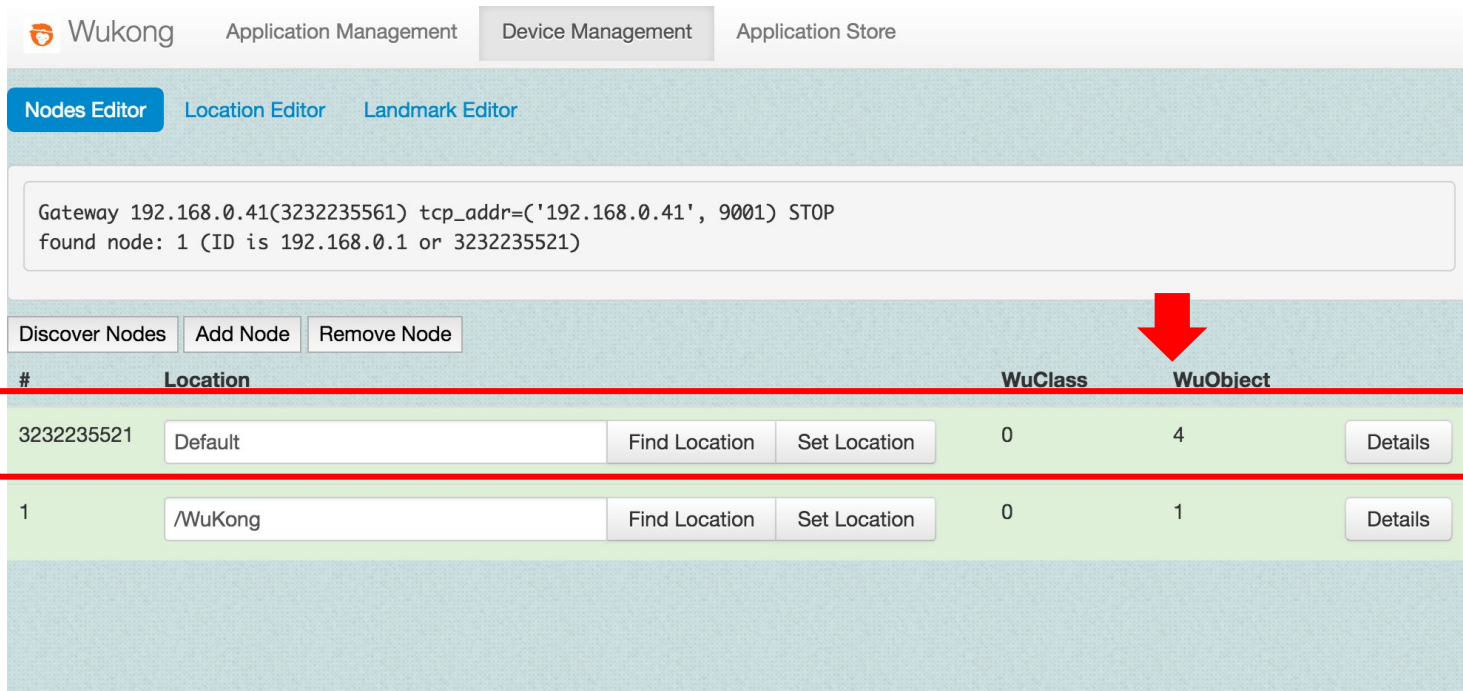
```
git pull
```

- Check current branch

```
git branch
```


Include udpdevice_workshop.py

- Follow the instructions of the chapter 4.1.2 of Gitbook to include udpdevice_workshop.py

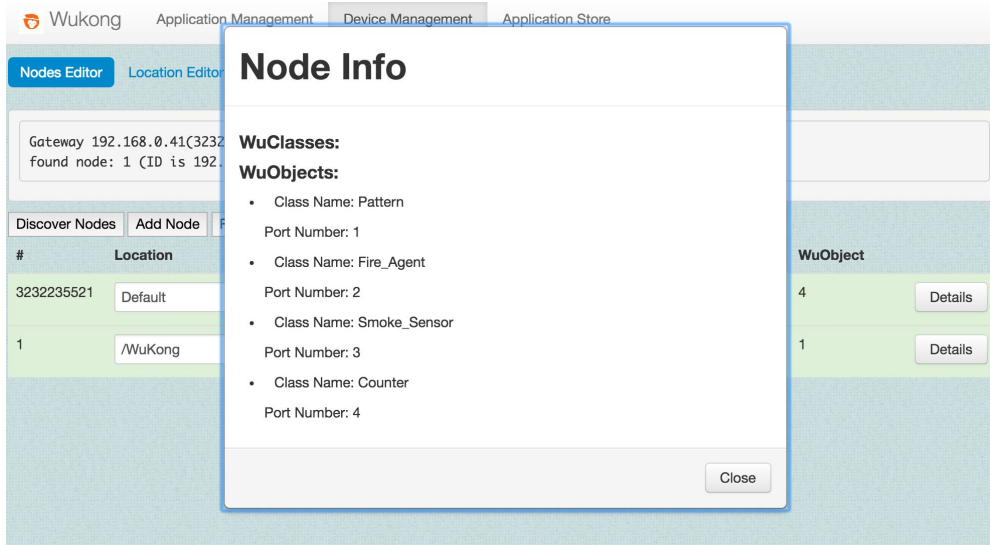


The screenshot shows the Wukong web interface for device management. At the top, there are tabs for 'Application Management', 'Device Management' (which is active), and 'Application Store'. Below these are sub-tabs: 'Nodes Editor' (active), 'Location Editor', and 'Landmark Editor'. A status box displays the message: 'Gateway 192.168.0.41(3232235561) tcp_addr=('192.168.0.41', 9001) STOP found node: 1 (ID is 192.168.0.1 or 3232235521)'. Below the status box are buttons for 'Discover Nodes', 'Add Node', and 'Remove Node'. A table lists the discovered nodes. The first row, with ID 3232235521, is highlighted with a red box and has a red arrow pointing to it from above. The table has columns for '#', 'Location', 'WuClass', and 'WuObject'. Each row also includes 'Find Location' and 'Set Location' buttons, and a 'Details' button.

#	Location	WuClass	WuObject
3232235521	Default	0	4
1	/WuKong	0	1

Compare Node Info with udpdevice_workshop.py

- Press “Detail” button to check which objects are included in the udpdevice_workshop.py

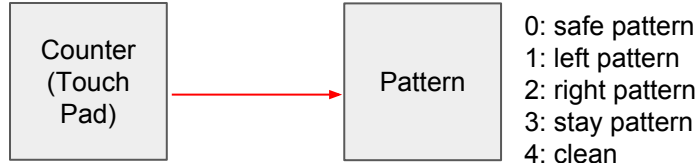


```
9 import udpdevice_pattern_no_thread
10 import udpdevice_fire_agent
11 import udpdevice_smoke_sensor
12 import udpdevice_counter
```

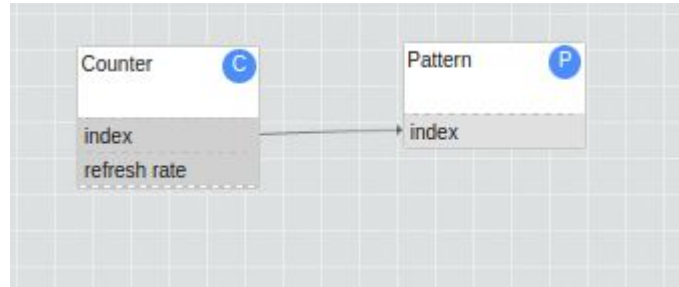
Lab 1

Spec:

Pattern will change according to how many times the counter is pressed.

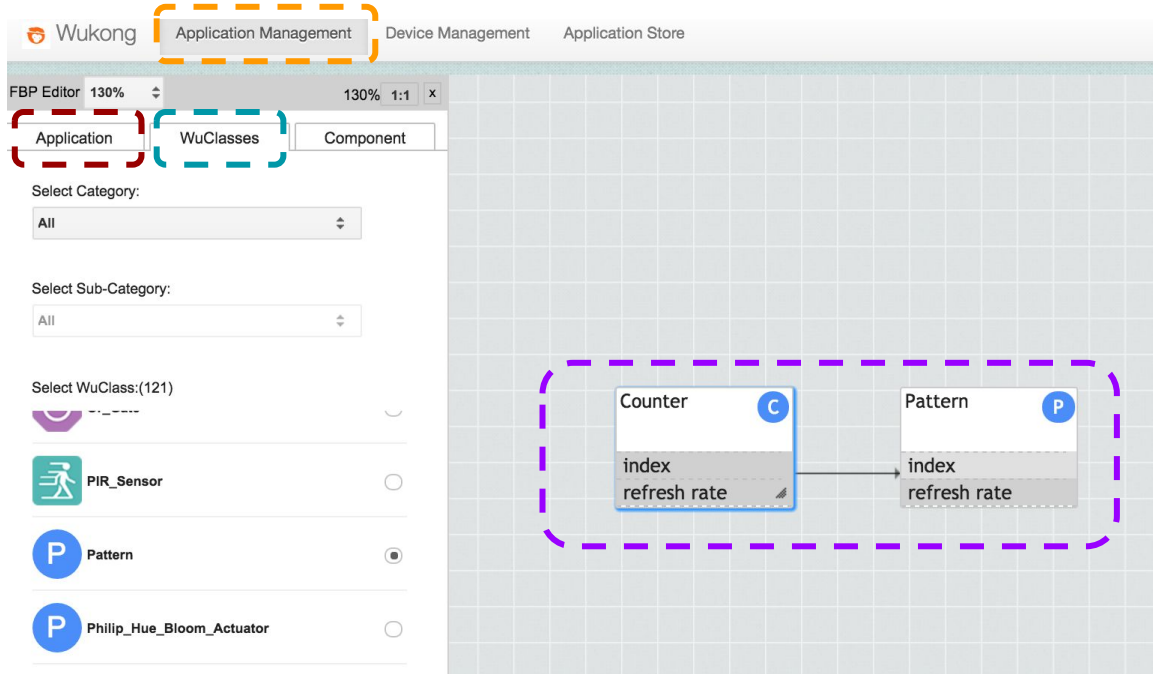


Output: 0, 1, 2, 3, 4



Draw FBP and Deploy Application

- Follow the instructions of chapter 4.1.3 of gitbook to build this application.



Procedures:

Include new devices

Create a new FBP

Add components

Set properties and locations

Add links between components

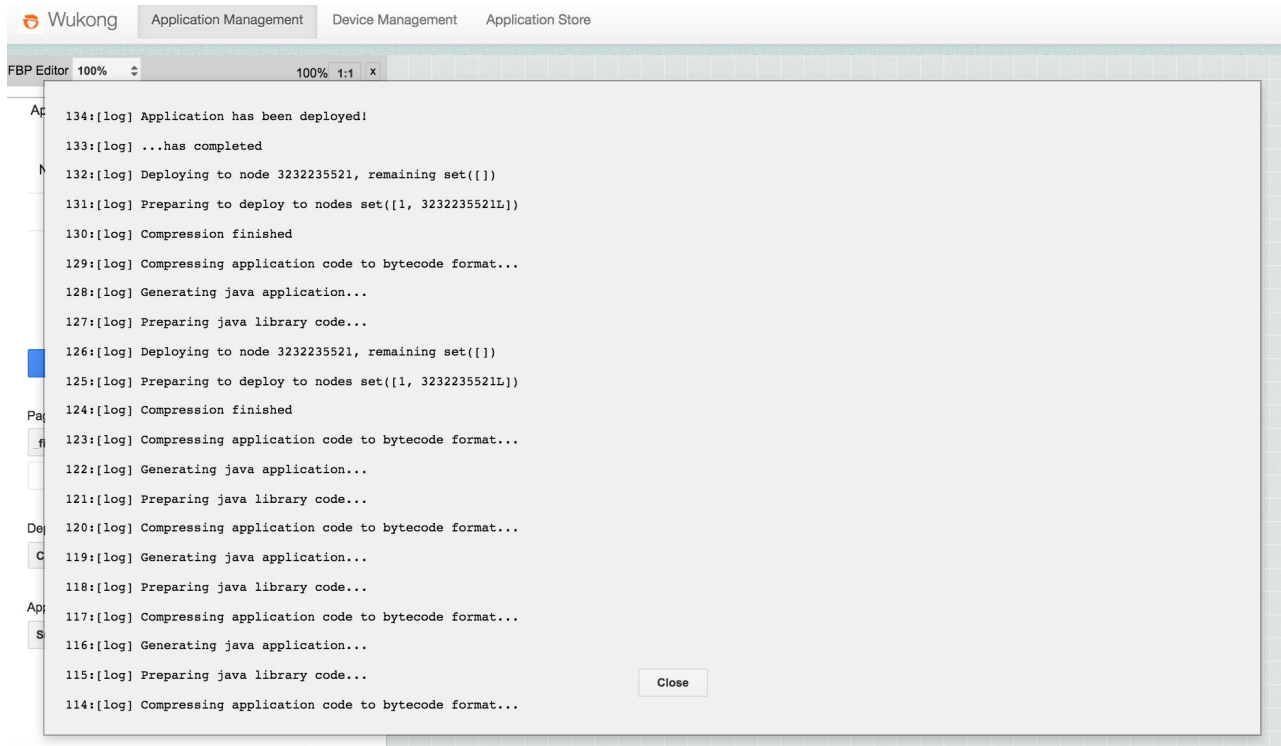
Save FBP

Mapping components to physical devices

Deploy FBP to physical devices

During Deployment ...

- The process of deployment takes a while because the application is compiled by edison or pi.



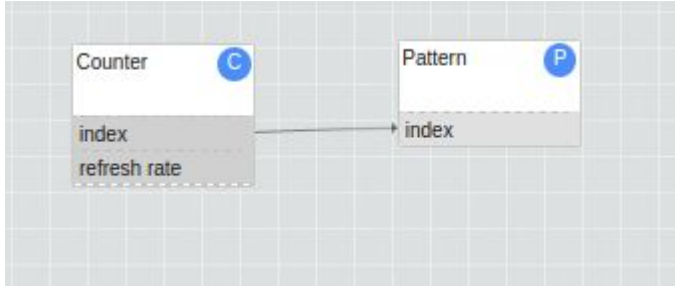
The screenshot shows the Wukong Application Management interface. The top navigation bar includes 'Wukong', 'Application Management' (selected), 'Device Management', and 'Application Store'. Below the navigation bar, there's a tab labeled 'FBP Editor' with a zoom level of '100%' and a window title '100% 1:1 x'. The main content area displays a log of deployment steps, numbered from 114 to 134. The log entries are as follows:

```
134:[log] Application has been deployed!
133:[log] ...has completed
132:[log] Deploying to node 3232235521, remaining set({})
131:[log] Preparing to deploy to nodes set([1, 3232235521L])
130:[log] Compression finished
129:[log] Compressing application code to bytecode format...
128:[log] Generating java application...
127:[log] Preparing java library code...
126:[log] Deploying to node 3232235521, remaining set({})
125:[log] Preparing to deploy to nodes set([1, 3232235521L])
124:[log] Compression finished
123:[log] Compressing application code to bytecode format...
122:[log] Generating java application...
121:[log] Preparing java library code...
120:[log] Compressing application code to bytecode format...
119:[log] Generating java application...
118:[log] Preparing java library code...
117:[log] Compressing application code to bytecode format...
116:[log] Generating java application...
115:[log] Preparing java library code...
114:[log] Compressing application code to bytecode format...
```

A 'Close' button is located at the bottom right of the log window.

Lab 1

- However, after deployment, the pattern won't change no matter how you press the touch pad because the update function of Counter WuClass is EMPTY!!
- Our task is to write update function to simulate the behavior of previous video.



```
<WuClass name="Counter" id="1017" virtual="false" type="hard">
  <property name="index" access="readwrite" datatype="short" default="0" />
  <property name="refresh_rate" access="readwrite" datatype="refresh_rate" default="100" />
</WuClass>
```

```
<WuClass name="Pattern" id="2038" virtual="false" type="hard">
  <property name="index" access="writeonly" datatype="short" />
  <property name="refresh_rate" access="readwrite" datatype="refresh_rate" default="50" />
</WuClass>
```

```
10 class Counter(WuClass):
11     def __init__(self):
12         WuClass.__init__(self)
13         self.loadClass('Counter')
14         self.IO = pin_mode(Counter_Pin, PIN_TYPE_DIGITAL, PIN_MODE_INPUT)
15
16     def update(self,obj,pID=None,val=None):
17         current_value = digital_read(self.IO)
```

EMPTY!!!

https://github.com/wukong-m2m/wukong-darjeeling/blob/workshop/wukong/gateway/udpwpkpf/udpdevice_counter.py#L16

Lab 1

- How to program this lab?

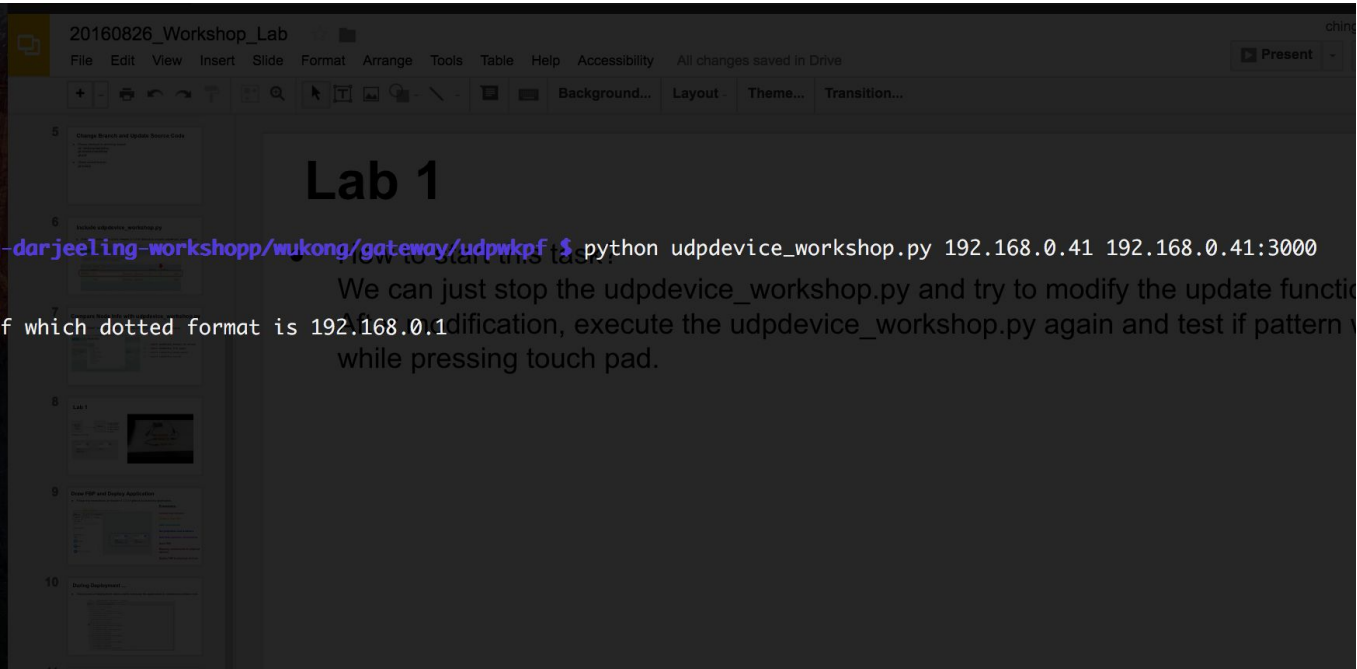
We can just stop the `udpdevice_workshop.py` and try to modify the update function.

After modification, execute the `udpdevice_workshop.py` again and test if pattern will change while pressing touch pad.

Ctrl+c

execute
again

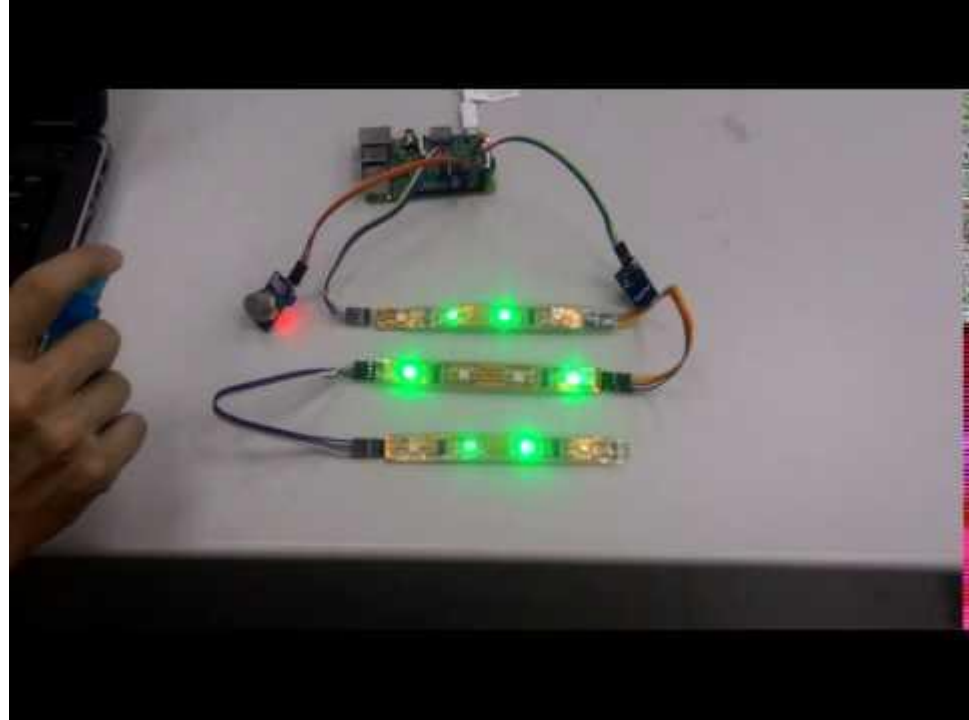
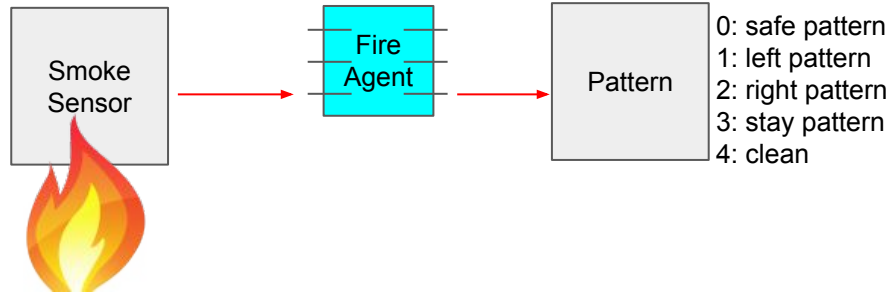
```
Alarm triggered
Pattern 0 start
Alarm triggered
Pattern 0 start
Alarm triggered
^CPattern 0 start
Alarm triggered
Pattern 0 start
Alarm triggered
pi@raspberrypi:~/wukong-darjeeling-workshop/wukong/gateway/udpworkshop$ python udpdevice_workshop.py 192.168.0.41 192.168.0.41:3000
Init: Pattern 0 start
Fire Agent init success
Your ID is 3232235521 of which dotted format is 192.168.0.1
Pattern 0 start
Alarm triggered
Pattern 0 start
Alarm triggered
Pattern 0 start
Alarm triggered
Pattern 0 start
Alarm triggered
Pattern 0 start
Alarm triggered
Pattern 0 start
Alarm triggered
Pattern 0 start
Alarm triggered
```



Lab 2

Brief Spec:

Pattern will change according to where is the smoke. For example, LED strip should show left pattern if the danger happens only in the right side.



Lab 2

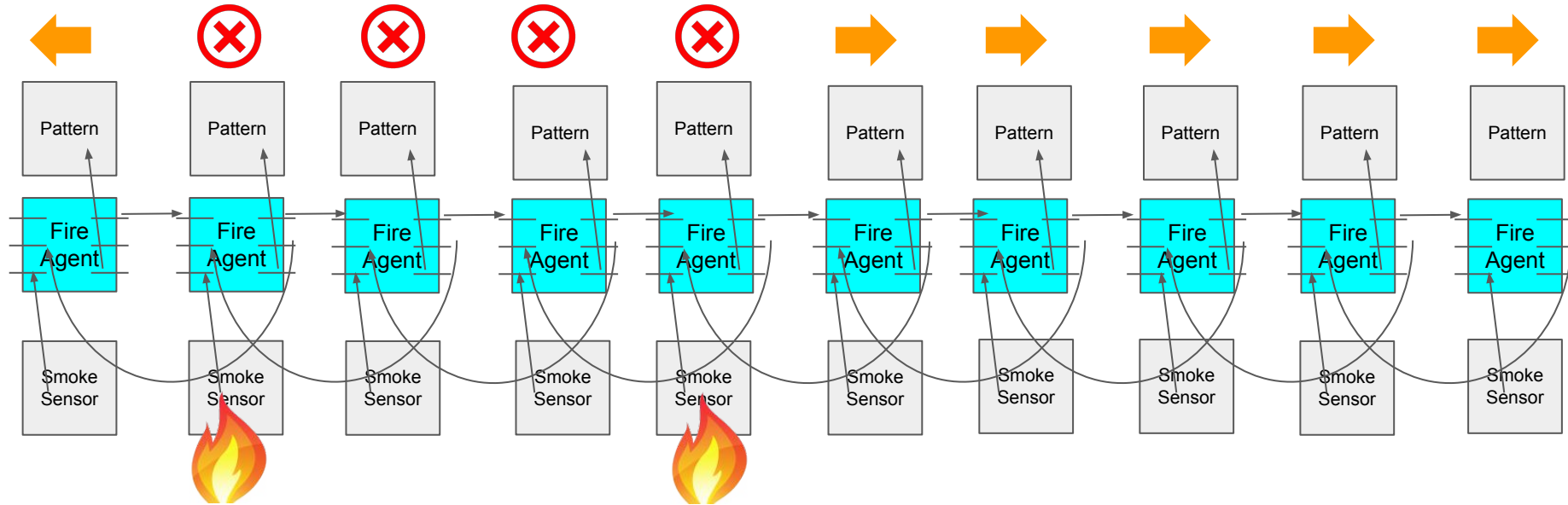
Detailed Spec:

LED strip should show **left** pattern if the danger happens only in the right side.

LED strip should show **right** pattern if the danger happens only in the left side.

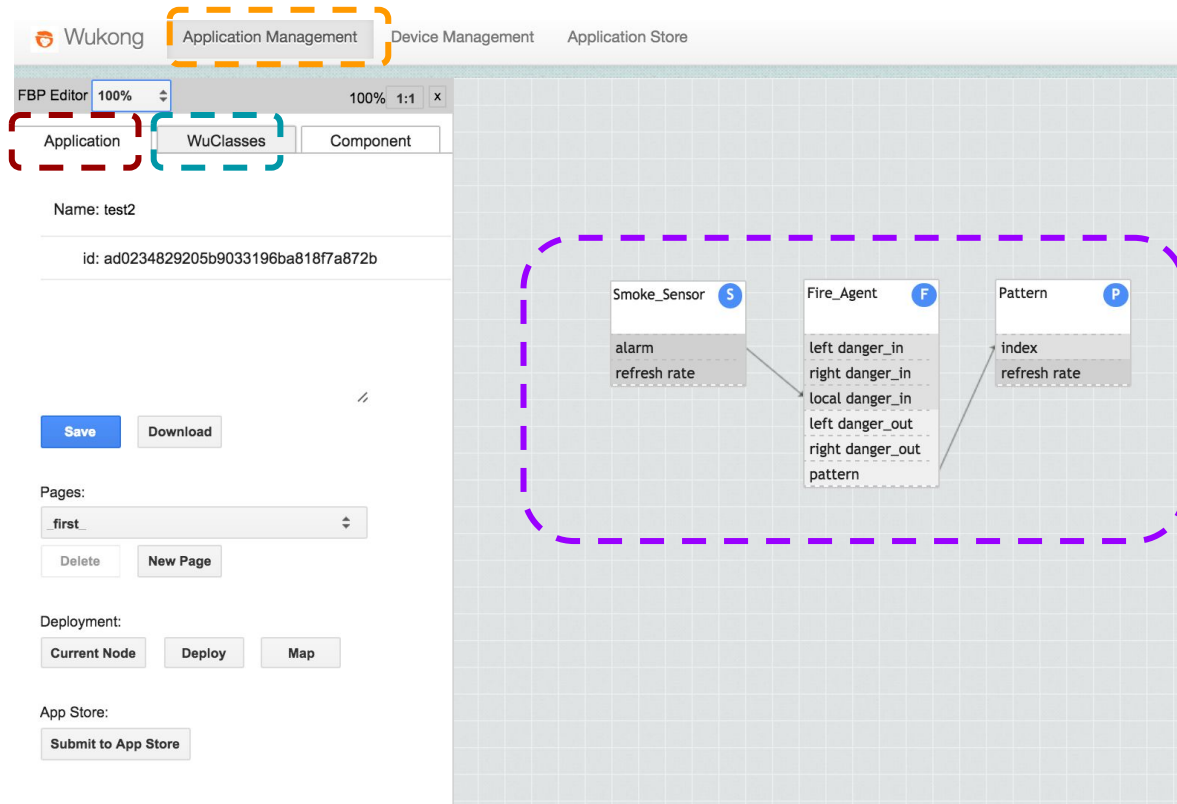
LED strip should show **stay** pattern if the danger happens locally or nearby.

LED strip should show **safe** pattern if there is no danger anywhere.



Draw FBP and Deploy Application

- Follow the instructions of chapter 4.1.3 of gitbook to build this application.



Procedures:

Include new devices

Create a new FBP

Add components

Set properties and locations

Add links between components

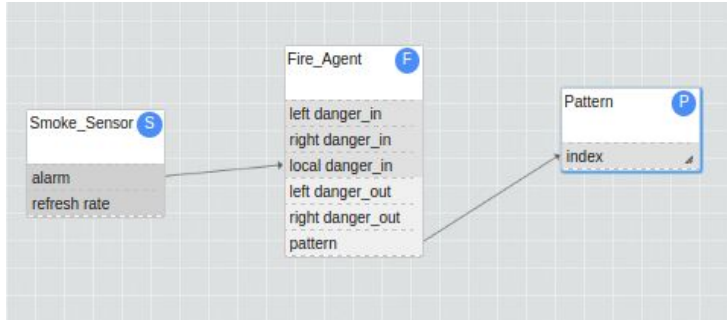
Save FBP

Mapping components to physical devices

Deploy FBP to physical devices

Lab 2

- However, after deployment, the pattern won't change whether there is smoke or not because the update function of Fire_Agent WuClass is EMPTY!!
- Our task is to write update function to simulate the behavior of previous video.



```
<WuClass name="Smoke_Sensor" id="1018" virtual="false" type="hard">
  <property name="alarm" access="readwrite" datatype="boolean" default="false" />
  <property name="refresh_rate" access="readwrite" datatype="refresh_rate" default="100" />
</WuClass>
```

```
<WuClass name="Fire_Agent" id="18" virtual="true" type="soft">
  <property name="left_danger_in" access="writeonly" datatype="boolean" default="false" />
  <property name="right_danger_in" access="writeonly" datatype="boolean" default="false" />
  <property name="local_danger_in" access="writeonly" datatype="boolean" default="false" />
  <property name="left_danger_out" access="readonly" datatype="boolean" default="false" />
  <property name="right_danger_out" access="readonly" datatype="boolean" default="false" />
  <property name="pattern" access="readonly" datatype="short" default="0" />
</WuClass>
```

```
6 class Fire_Agent(WuClass):
7     def __init__(self):
8         WuClass.__init__(self)
9         self.loadClass('Fire_Agent')
10        print "Fire Agent init success"
11    def update(self,obj,pID,val):
12        if pID == 0 or pID == 1 or pID == 2:
13            left_in = obj.getProperty(0)
14            right_in = obj.getProperty(1)
15            local_in = obj.getProperty(2)
16
```

EMPTY!!!

https://github.com/wukong-m2m/wukong-darjeeling/blob/workshop/wukong/gateway/udpwpkf/udpdevice_fire_agent.py#L11

Lab 2

- How to program this lab?

We can just stop the `udpdevice_workshop.py` and try to modify the update function.

After modification, execute the `udpdevice_workshop.py` again and test if pattern will change when alcohol is sprinkling.



**execute
again**



Lab 3

