# Using Philips Hue

without technical knowledge

This tutorial explains how to use the Philips Hue (**www.meethue.com**) connected Things: lights, switches, motion - temperature - luminosity sensors.

Thanks to a visual editor (NodeRED from IBM) and the HueMagic component, you can manage the whole set of Philips Hue products without any technical knowledge.

Palette   Workspace   Display / Info
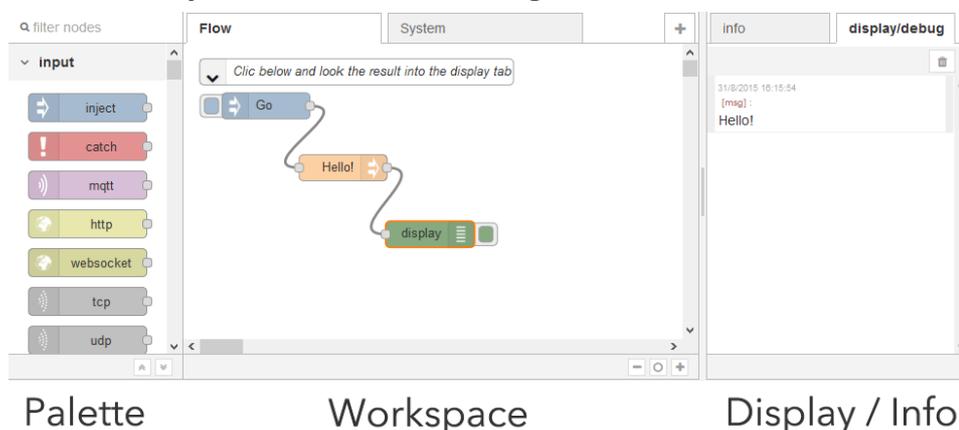
Learn how to manage:

- **A light**

- **A motion Sensor**

- **A brightness Sensor**

- **A temperature Sensor**

- **A tap switch**

- **A dimmer switch**

> Notice that today, brightness and temperature Sensors are included in the motion sensor.

Credits

All this made possible thanks to the outstanding work of **HueMagic** from Foddy.

`Donate` `PayPal` if you use it. (many people give as much as 1 dollar)
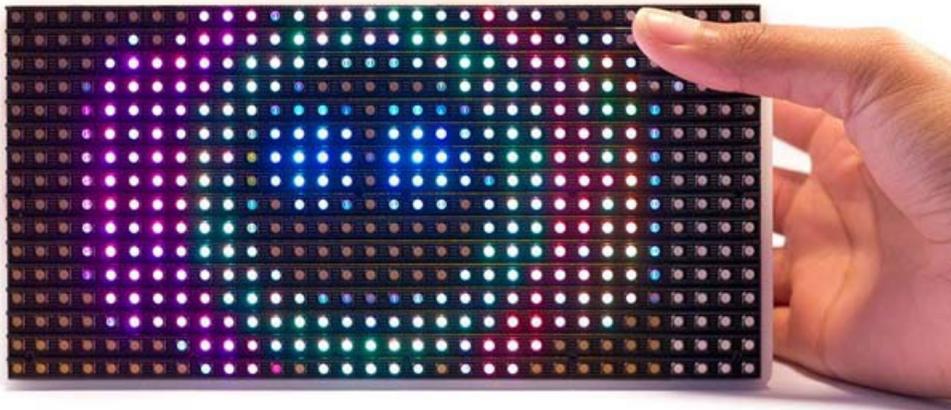
Devices

The following tutorial can be used with two devices of you choice:

- A firmware named **TheThingbox** (**Thingbox.io**): a SDCard image you can download for free and insert into a Raspberry Pi. See below for details.

- Alternatively you can buy a **TimeSquAir** which provide you with an already built device, a NFC reader/writer and a LED Matrix to display informations.
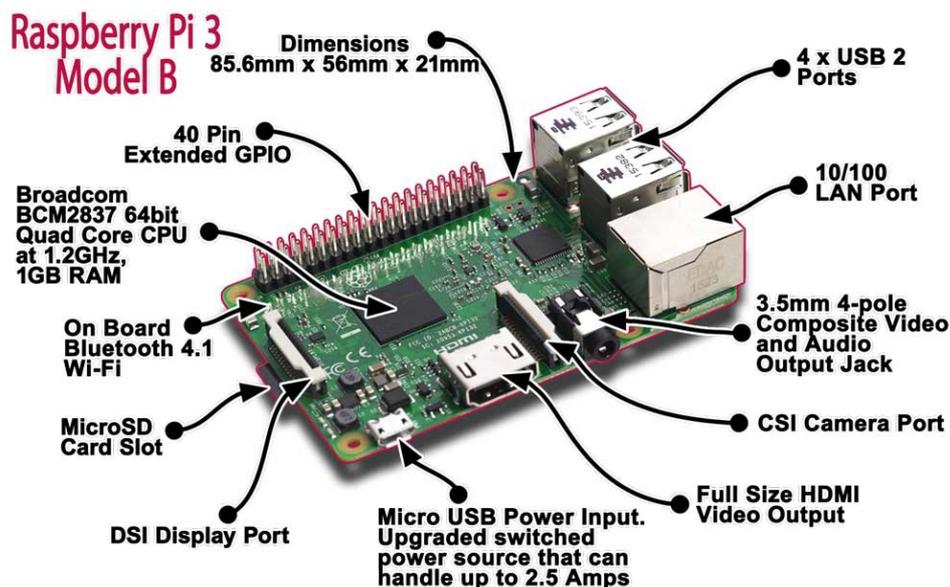

See **TimeSquAir.io**.

Needed components

To build a Thingbox, you'll need the following components:

Discover the Raspberry Pi

https://www.raspberrypi.org/

What do I need ?

- A **Raspberry Pi**, buy **here**

- You can add a box for your Pi **here**

- A **Power supply** : 5V 1A micro USB power supply to power your Raspberry Pi **here** (to know if an adapter can be used check **this link**)

- A **SD card** : 16 Gb is the minimum like this one:

  https://www.amazon.fr/gp/product/B073K14CVB

- An ethernet (RJ45) cable if you don't want to use WiFi

Notice that you will also need a usual computer with a SDCard slot (with adaptator) and a web browser.

Download and Flash

Two steps:

- Download the binay file of the SDCard
- Unzip and write this file to a SDCard

Download the firmware

The firmware is a very big file (many Gb) that is downloaded from the cloud and then written on the SDCard that will then be inserted in the Raspberry Pi. As this firmware runs on the Pi, it is the same if you use Windows or Macintosh.

If you are using a Thingbox: Download the image from **thethingbox.io**

If you are using a TimeSquAir: Download the image from **timesquair.io** . As TimeSquair is shipped with an already in place SDCard, this only applies to rebuild a card.

Write the SDCard

Etcher is the application to write the firmware on the SD card

Etcher (**https://etcher.io/**) turns the whole process of flashing an OS image file into three simple steps: Select Image, Select Drive, and Flash Image. More importantly, the same program, with the same interface, is available on all three types of computer – Windows, Mac, and Linux – which makes it easy for everybody to understand.

More in **https://www.raspberrypi.org/magpi/pi-sd-etcher** :
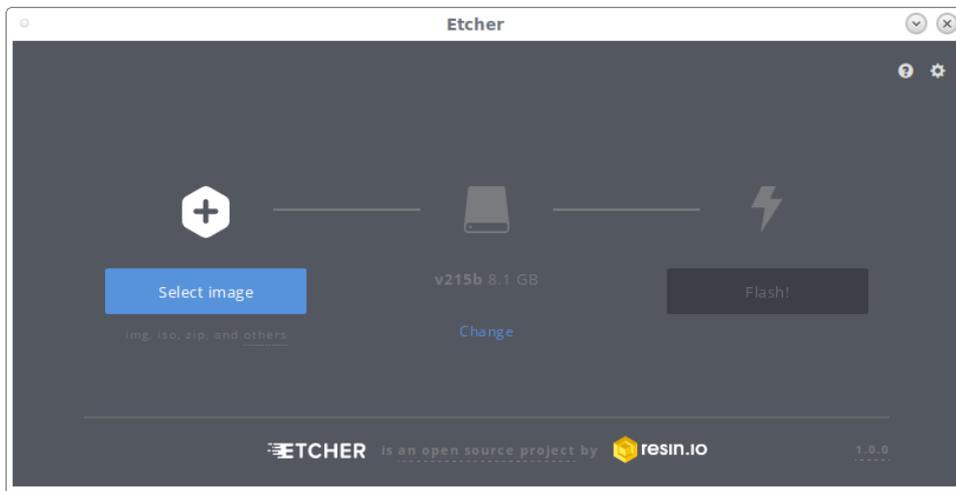
- **Download and install** Etcher from the website.

- **Launch** the application file and follow the Etcher setup wizard.

- **Click "Select Image"** in Etcher. Use the file manager window and locate the zip file you dowloaded (there is no need to unzip it). Click Open. The image will appear under Select Image, and Connect a drive will highlight red.

> **Warning** : the SD card will be totally over-written, so this SD card should be used only for the Raspberry Pi.

- Insert your SD card in the computer. Etcher will select it automatically. Etcher won't write to your hard drives by default, but check that the SD card is listed correctly. Now click Flash! to write the image file to the SD card.

Plug

Etcher will format the SD card, before writing and verifying the image; this is shown by a progress bar. When done, remove the SD card, insert it into your Raspberry Pi, and power it up.

The first boot is longer because it expands the SDCard to use the free part of the SDCard. Then reboot again.

> If you don't want to use WiFi, plug the local network wire BEFORE powering the device.

Initial Setup

There is two ways to connect to internet:

- The first solution is to use an **ethernet wire**. There is nothing more to do that plugging it in a ethernet port.

- The second solution is to use the **WiFi Access Point** that is created at startup (for 20 minutes, if there is no other internet access). It allows to easily set the WiFi parameters and connect the device to the network.

Using the WiFi Access point

When launched, a WiFi Access Point is automatically created for a few minutes. It allows to easily set the WiFi parameters and connect the device to the network.

More precisely, the access point is created when the following conditions are met:

- No Wire connection

- No previous WiFi connection (empty or wrong parameters)

To use it:

- Power-On the device

- Look at the WiFi access points from your PC or your smartphone. It may take a while before it pops. The name is built with the "digitalairways" string concatened with part of the device identifier. It looks like `digitalairways_32bc`.

- Connect to this open access point with your PC or smartphone

Since the device is not yet internet connected, you may get an alert saying just that. Dismiss it.

At this point, your PC or your smartphone may tell you there more info to add and give you the opportunity to switch to a browser based page.

- When connected, the browser should automatically swap to a wizard asking for inital setup.

The setup Wizard

The setup Wizard allows to set * the WiFi parameters the device should connect to, * a new network name (a name should be unique in the same network)

xxx describe the setup here xxx

At this point the device reboots and connects to the new network.

Just **connect your PC to the same network** and navigate to it.

Access the setup

All you need is a computer on the same local network.

Use your favorite internet browser (Google Chrome and Mozilla Firefox are the most tested).

Avoid *Microsoft Edge* which seems to have a very strange behavior). Also some **Android** tablets seems unable to resolve .local addresses.

Enter the network name you chose in the setup folowed by ".local". Suppose you chose "mySuperMachine":

```
mySuperMachine.local/
```

in the address bar.

The final "/" may be omitted, depending of the browser.

If you used the settings page to rename your device, say, to "myBox", the browser adress will be:

```
myBox.local/
```

If it does not work !

Some local networks are not able to resolve .local network names.

- make sure you are on the same local network

- be aware that some electronic router may block that address

- Try the IP address (the four dot separated numbers at are displayed at the launch time) something like `http://192.168.0.23` You can get the IP address in your router parameters in the DHCP list.

Last but not least: plug a Screen on the HDMI port (you may have

to open the box), and look at the console for some errors.

The visual editor

This tutorial introduces the visual editor: a web page (that you access from your internet browser) displaying a graphical interface that allows to create new applications and change preferences.

The visual editor is IBM's Node-Red (**http://node-red.org**) an open source visual editor for using the internet of things.

It is a very dynamic project with an ever growing community.

Discovering Node-RED

Palette   Workspace   Display / Info
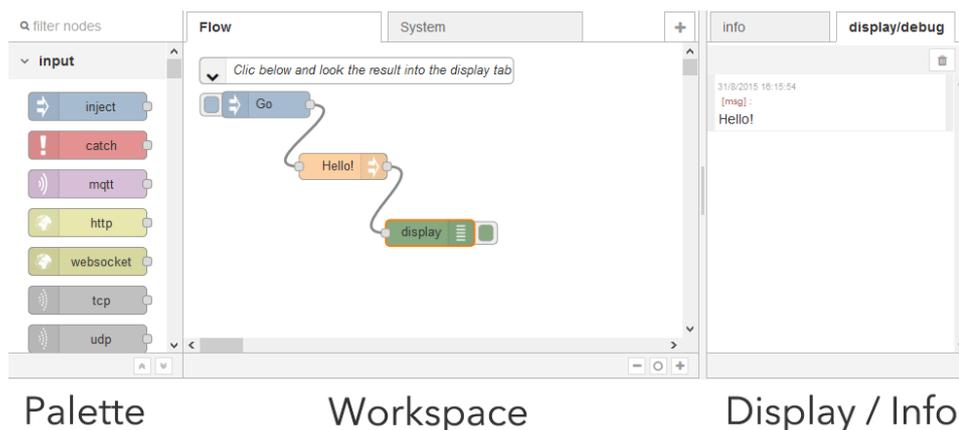
The **Palette** contains all available **nodes**.

Nodes can be wired together to build **flows**.

In the flow above, clicking "Go" goes through "hello" and ends in the display node that prints "Hello" in the display tab.

NodeRED tutorial

Follow the great tutorial "**getting started with Node-RED**"

How to build a flow

Building a flow is easy:

- Drag and drop nodes from the left palette
- Wire them together by drag'n dropping their handles

Activating

Once modified (by adding or removing nodes or wires) the flow should be activated by using the activate top right button.

Push the flow from your desktop web page to the device:

 use "modified flows" only to save time.

Trigger the flow

The flow is triggered by a manual event (as the "Go button" above) or by external events (tweets, NFC tag, ...). When triggered, the execution starts with the trigger node and goes right through the flow. If a Debug or Display node is invoked, a string is displayed in the info/debug tab.

Flows follow the scheme below:
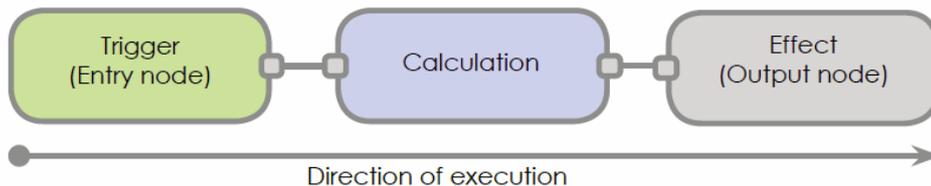
You can click on the left part of the Go node. This injects to the hello node (whose job is to put hello on the main track of the flow, known as the payload), and then to the display node whose job is to display the payload.

If you now select the display tab, you should see the output:

Modify the flow
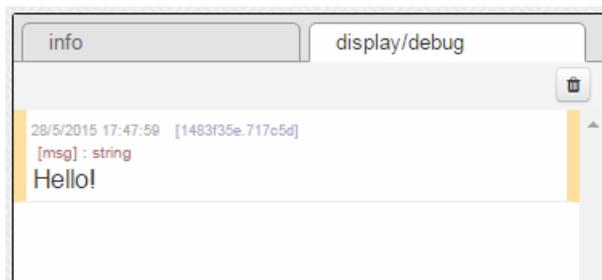
Each node can be tunned by double clicking it. Double click on the Hello node (which was built by dragging a node Set whose job is to change the value of the payload track of the flow):
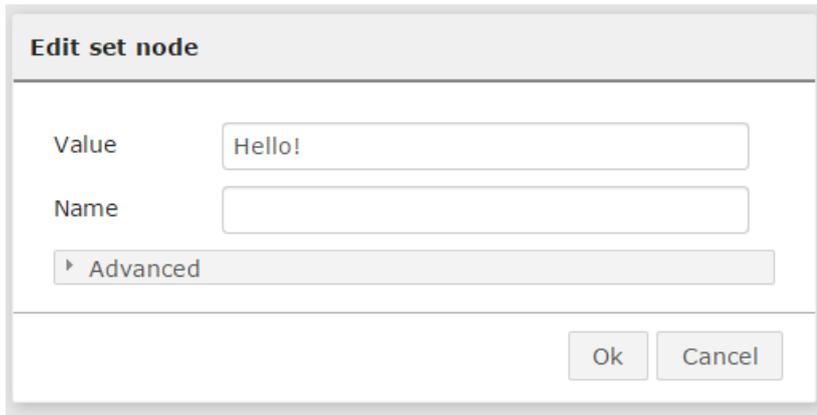
You can change the value to "Hello world" and click ok.

Now a particular point should be understood:the browser you are looking at is running on your computer, not on the Thingbox/TimeSquAir. You just edit the flow on your computer the same way you edit your Facebook page.

When the modification of the flow is complete, it must be sent to the Thingbox/TimeSquAir where it runs:


Now you can hit the Go bouton again!

Important note: the Pi need about 40s to definitively write the flow: do NOT power off or reboot the Pi before waiting this delay, the flow may be lost!!!

Learn more about Node-RED

- First read the IBM's Node-Red websites (**http://nodered.org/**) and **https://developer.ibm.com/open/projects/node-red/**.

- **noderedguide.com** series of lectures designed to get you started with Node-RED

You can also go to Youtube.com to get great video tutorials

- Node-RED innovators Dave Conway-Jones and Nick O'Leary provide an overview, demo, and technical details for their innovation that's making a big splash.
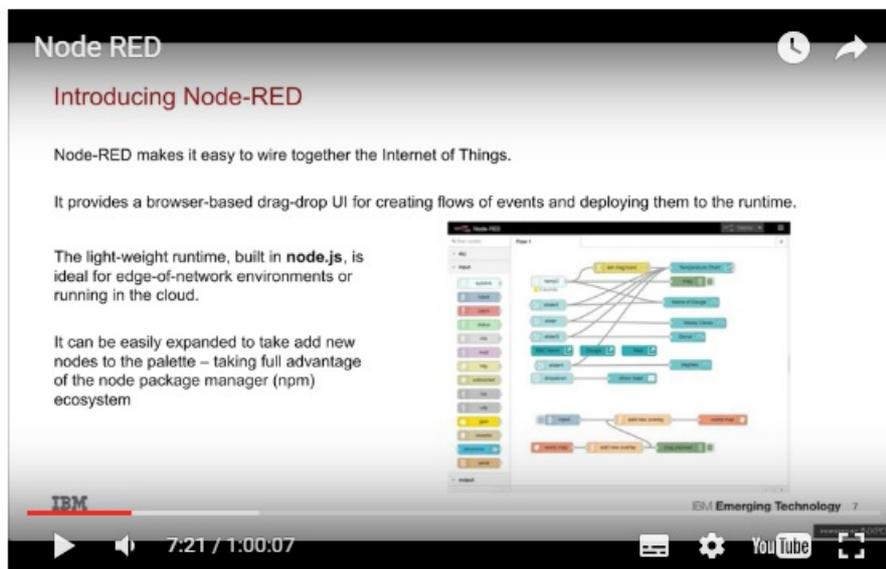
**https://www.youtube.com/results?
search_query=introducing+node+red**

- Read (and contribute to) the forum
  https://discourse.nodered.org/

⟲ How to find new nodes

Browse the IBM's repository: **http://flows.nodered.org**. There is a
great Node-RED community that publish new nodes every day.

Managing Philips Hue

⦿ Introducing Philips Hue bulbs

Philips Hue provides a permanently growing collection of connected things: bulbs, switches, motion sensor, temperature and luminosity.

The hue bulbs can be turned on and off, and change their color dynamically. NodeRED nodes provide a simple access to every control you can have on your hue devices.

⦿ Before you start

Plug in your bridge and it will automatically power up. Connect it to your Wi-Fi router using the network cable provided. **Wait for the three lights to come on** and you are good to go.

Then download the Hue app on your phone and discover the bulbs to add them in the bridge.

The app is a nightmare. They have decided that you have to create a virtual room to monitor your bulbs. Good luck.
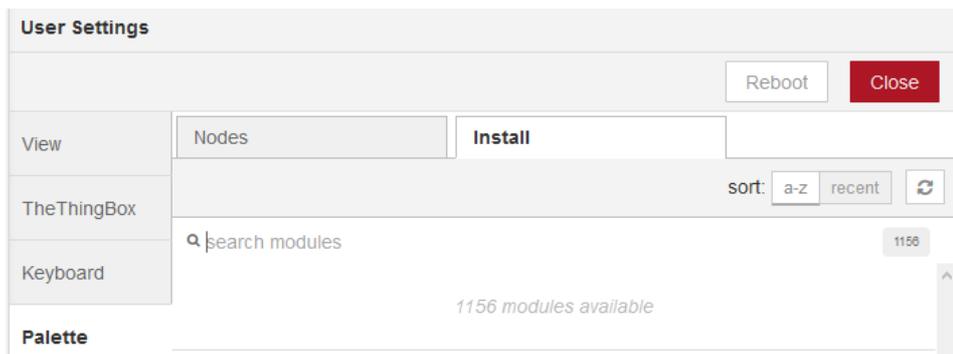
The Hue nodes we will use in this tutorial will only work if you have already configured your Hue Bridge and added the Hue

devices to it as described in the Hue documentation.

For more informations about how to do it, go **on the Phillips Web site**.

Install the nodes

Click the right menu in the Node-RED web page. Choose `Manage Palette` then `Install`.

User Settings				
			Reboot	Close
View	Nodes	Install		
TheThingBox			sort: a-z recent ⟳	
Keyboard	🔍 search modules			1156
	1156 modules available			
Palette				

- Type `ttb-huemagic` in the edit box and click the OK button.

- Wait for the install

> 💬 The installation may take up to 5 minutes (a progress bar shows the progress), depending on what is already installed on your device.

- No need to reboot or to refresh the web browser page.

- Browse the palette and look at the newly added nodes.

Bridge pairing

Each hue node you drag on the NodeRED desktop should be paired with the Hue bridge. If there is only one bridge, it is added automatically as the last one is used, except the first time.

Let's add it for the first node.

Drag'n drop the bridge node:

Hue Bridge

Edit it:

🏷 Name	Hue Bridge Name
🖥 Bridge	Add new hue-bridge... ▾ ✏
⟳ Auto Updates	☑ Activate auto updates

click on the pencil

✏

🏷 Name	Hue Bridge|
🖥 Bridge IP	X.X.X.X[:PORT] 🔍
🔑 API Key	👤
⏱ Poll interval	3000

click on:

🔍

and wait "searching..." to disappear.

Then the pull down menu shows your Hue bridges:

🖥 Bridge IP	▾ 🔍
🔑 API Key	192.168.0.166
	192.168.0.193
⏱ Poll interval	192.168.0.178

💬 Hope you don't have too much since it display addresses and not the names... Fortunately, the name appears if you choose one.

click on:

👤

Please press your Hue Bridge button within the next 20 seconds to create a new user!

And press the bridge central button, resulting in:

🔑 API Key	8pegbopHtaQzWBsXUw7ZbEnhpZSJORI3 👤

Validate then all dialogs.

At this point, The bridge is paired, you can add any hue device:

- **Add lights**
- Add Hue Tap
- Add Hue Motion sensor (also contains temperature and

luminosity)

Hue lights

⦿ Hue prerequisite

Pairing

The Hue nodes we use in this tutorial will only work if you have already configured your Hue Bridge and added the Hue devices to it as described in the Hue documentation.

For more informations about how to do it, go **on the Phillips Web site**.

This also may help: https://www.makeuseof.com/tag/philips-hue-motion-sensor-guide/

💬 Tip: reset of the sensor is done by pressing and holding in the reset button on the back of the sensor until the LED becomes green then oprange (usually 5 seconds). Press the reset again to put sensor into pairing mode.

Remove default behavior

The sensor comes with a default behavior (as switching lights on when lumonosity moves down) you should delete since it usually does not fit your needs.

I didn't find how to do it with the official Hue App. I did it with the alternativ phone App "all 4 hue" (**google Play** and **AppStore**): go to "Show rules" and choose "delete" for each.

Build a simple flow

Drag'n drop any node, a light node for instance:

Edit it: The previously defined bridge should already be set.

Click on:

Nothing seems to be done but then a pull down menu contains the bulbs:

Choose the one and validate the form.

On/Off

Look for intents in the palette:

Build the following flow:

Sending the intent "open" turns the light on, and "close" off.

Changing color

Build the following flow:
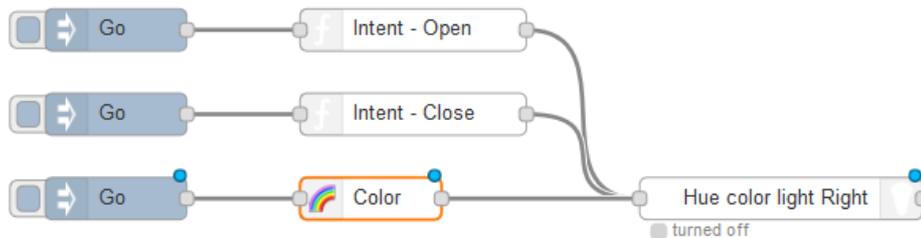
Edit the node color to change the color:

Sending the intent "color" changes the color accordingly. Notice that if the light is off it will also be set to node (Hue makes an error if not).

Clicking on Advanced allows you to enter the color hex code (#FFFFFF for white):
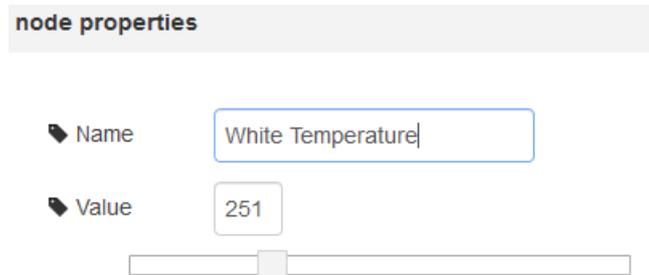

don't check "Use white" for that purpose.

Changing White temperature

Some Hue bulbs only handle white color, warm or cold.

Use the node:

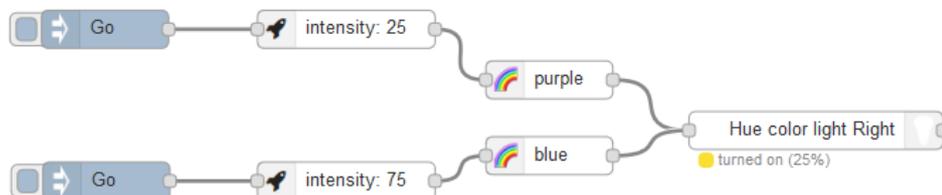If you edit it, there is a slider: left is cold, right is warm:

node properties

🏷 Name   White Temperature

🏷 Value   251

Brightness / intensity

Use the intensity node

Intensity	Advanced

🏷 Name :

🏷 Valeur :   84
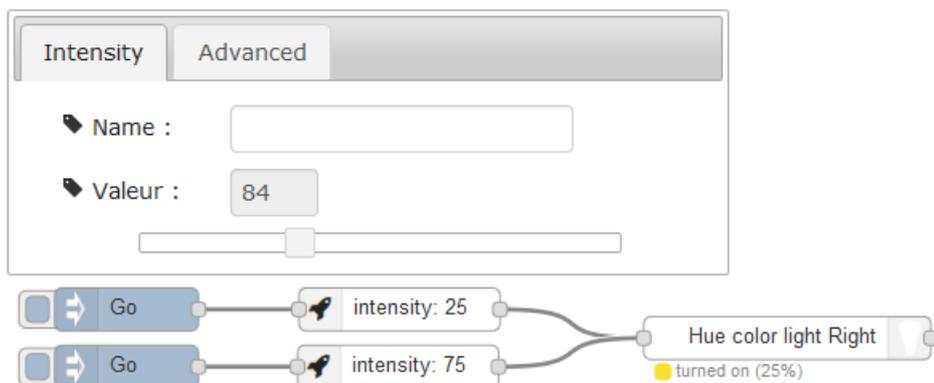
Another possibility to also tuen on le light at the same time:

Transition time

!!! à tester

Edit another intensity node, in the advanced tab:

Change it to:

Reading state

Each command send to the lighty node also gets out with the state of the bulb:

Just add a debug node, edit it to display the complet msg:


```json
{
    "payload": {
        "on": true,
        "brightness": 1,
        "reachable": true,
        "rgb": [255, 114, 246],
        "hex": "ff72f6",
        "color": "violet",
        "updated": "2019-01-11T14:27:26+00:00"
    },
    "info": {
        "id": "2",
        "uniqueId": "00:17:88:01:00:c3:7e:ac-0b",
        "name": "Hue color light Right",
        "type": "Color light",
        "softwareVersion": "5.5.1.9461",
        "model": {
            "id": "LLC012",
            "manufacturer": "Philips",
            "name": "Hue Living Colors Bloom",
            "type": "Color Light",
            "colorGamut": "A",
            "friendsOfHue": true
        }
    },
    "_msgid": "fe7f1cae.e52aa"
}
```

If you just want the status, use a msg with an empty payload:

Motion, Temperature and motion

⊙ Hue prerequisite

Pairing

The Hue nodes we use in this tutorial will only work if you have already configured your Hue Bridge and added the Hue devices to it as described in the Hue documentation.

For more informations about how to do it, go **on the Phillips Web site**.

This also may help: https://www.makeuseof.com/tag/philips-hue-motion-sensor-guide/

💬 Tip: reset of the sensor is done by pressing and holding in the reset button on the back of the sensor until the LED becomes green then oprange (usually 5 seconds). Press the reset again to put sensor into pairing mode.

Remove default behavior

The sensor comes with a default behavior (as switching lights on when lumonosity moves down) you should delete since it usually does not fit your needs.

I didn't find how to do it with the official Hue App. I did it with the alternativ phone App "all 4 hue" (**google Play** and **AppStore**): go to "Show rules" and choose "delete" for each.

Build a simple flow

Drag'n drop a hue motion node:

Edit it: The previously defined bridge should already be set.

Click on:

Nothing seems to be done but then a pull down menu contains the sensors:

Choose the good one and validate the form.

Only devices that were previously added with the hue app on your phone with the Hue app will be present in the menu.
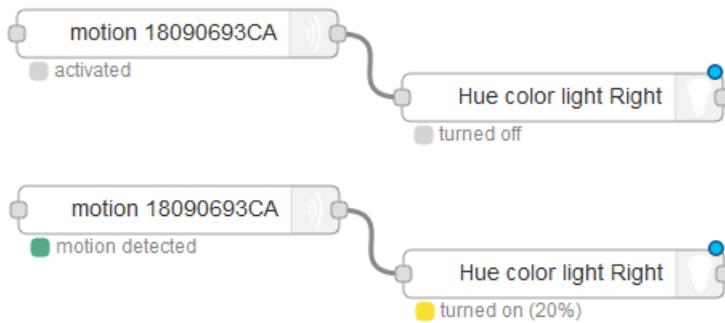
An easy flow is to display if there is motion:
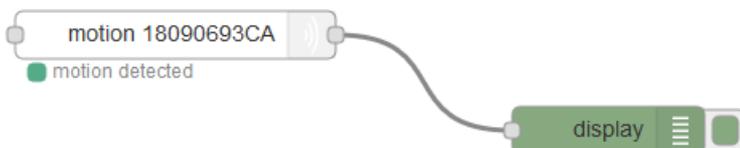
You get the data when you move:

4/29/2019, 3:01:46 PM   node: 9ccee4c3.e8d11

Sensor On

4/29/2019, 3:01:55 PM   node: 9ccee4c3.e8d11

Sensor Off

Switch On a light when you are here

It's easy to switch on a light when a motion is detected:

This works because the motion detector and the light nodes talk the same language (see intents here http://thethingbox.io/docs/Intents.html). You can use a ZWave (or Enocean) motion sensor node the same way, with a hue bulb, as well as a Hue detector wired to a ZWave bulb.

Advanced

Here is a typical output if you display the whole msg:

```
{
   "intent":1,
   "payload":1,
   "message":"Sensor On",
   "battery":100,
   "updated":"2019-04-29T12:54:36+00:00",
   "info":{
      "id":"4",
      "uniqueId":"00:17:88:01:04:b5:4f:d6-02-0406",
      "name":"motion 18090693CA",
      "type":"ZLLPresence",
      "softwareVersion":"6.1.0.18912",
      "battery":100,
      "model":{
         "id":"SML001",
         "manufacturer":"Philips",
         "name":"Hue Motion Sensor",
         "type":"ZLLPresence"
      }
   },
   "_msgid":"f7015ed1.a3c0b"
}
```

Tap Switches

⦾ Hue prerequisite

Pairing

The Hue nodes we use in this tutorial will only work if you have already configured your Hue Bridge and added the Hue devices to it as described in the Hue documentation.

For more informations about how to do it, go **on the Phillips Web site**.

This also may help: https://www.makeuseof.com/tag/philips-hue-motion-sensor-guide/

💬 Tip: reset of the sensor is done by pressing and holding in the reset button on the back of the sensor until the LED becomes green then oprange (usually 5 seconds). Press the reset again to put sensor into pairing mode.

Remove default behavior

The sensor comes with a default behavior (as switching lights on when lumonosity moves down) you should delete since it usually does not fit your needs.

I didn't find how to do it with the official Hue App. I did it with the alternativ phone App "all 4 hue" (**google Play** and **AppStore**): go to "Show rules" and choose "delete" for each.

Build a simple flow

Drag'n drop a hue tap node:

Edit it: The previously defined bridge should already be set.

Click on:

Nothing seems to be done but then a pull down menu contains the bulbs:

Choose the one and validate the form.

Only devices that were previously added with the hue app on your phone with the Hue app will be present in the menu.

Each node output is dedicated to one button.

Just add behavior for each one.

Dimmer Switches

◉ Hue prerequisite

Pairing

The Hue nodes we use in this tutorial will only work if you have already configured your Hue Bridge and added the Hue devices to it as described in the Hue documentation.

For more informations about how to do it, go **on the Phillips Web site**.

This also may help: https://www.makeuseof.com/tag/philips-hue-motion-sensor-guide/

> Tip: reset of the sensor is done by pressing and holding in the reset button on the back of the sensor until the LED becomes green then oprange (usually 5 seconds). Press the reset again to put sensor into pairing mode.

Remove default behavior

The sensor comes with a default behavior (as switching lights on when lumonosity moves down) you should delete since it usually does not fit your needs.

I didn't find how to do it with the official Hue App. I did it with the alternativ phone App "all 4 hue" (**google Play** and **AppStore**): go to "Show rules" and choose "delete" for each.

Build a simple flow

Drag'n drop a hue switch node:
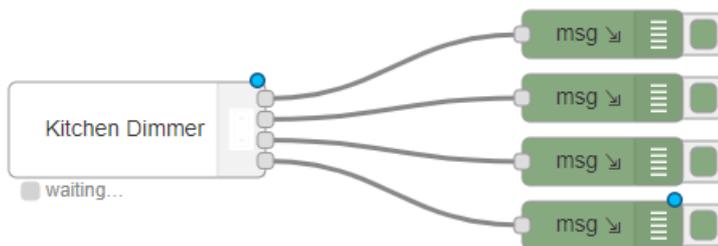
Edit it: The previously defined bridge should already be set.

Click on:

Nothing seems to be done but then a pull down menu now contains the switch:

Choose the right one and validate the form.

Only devices that were previously added with the hue app on your phone (either IOS or Android) will be present in the menu.

When adding the switch to the bridge within the App, don't forget to remove the default behavior of the buttons! I didn't find how to do it with the official Hue App. I did it with the alternativ App "**all 4 hue**" Go to rules and choose delete for each.

Node output

A simple use is to use one output per button. Just plug the action flow after the right output.

As intent are used for this node, the first "I" button means "open" (intent=1), "O" means "close" (intent = 0). There is also "More" and "Less" intents on the other middle buttons.

An advanced use is to analyse the output. There are many possibilities depending on whether you press the button for a long time or not, for example.

Here is an example:

```
▼ object
▼ payload: object
    button: 1003
    buttonnb: 1
    name: "On"
    action: "long released"
    updated: "2019-02-27T15:18:26+00:00"
▼ info: object
    id: "2"
    uniqueId: "00:17:88:01:04:e7:6c:7b-02-fc00"
    name: "ehpad interrupteur"
    type: "ZLLSwitch"
    softwareVersion: "5.45.1.17846"
    battery: 100
  ▶ model: object
    intent: 1
```

Build a simple flow

As an exercise, build a simple flow that switch-on a blub when a button is pressed.

Here is one that monitors my Jazz radio:
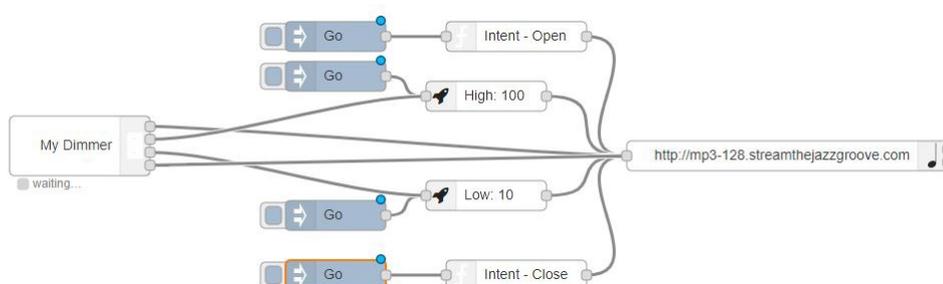
Brightness / light Sensor

The Brightness / light Sensor is embedded into the motion sensor.

Hue prerequisite

Pairing

The Hue nodes we use in this tutorial will only work if you have already configured your Hue Bridge and added the Hue devices to it as described in the Hue documentation.

For more informations about how to do it, go **on the Phillips Web site**.

This also may help: https://www.makeuseof.com/tag/philips-hue-motion-sensor-guide/

Tip: reset of the sensor is done by pressing and holding in the reset button on the back of the sensor until the LED becomes green then oprange (usually 5 seconds). Press the reset again to put sensor into pairing mode.
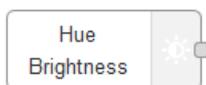
Remove default behavior

The sensor comes with a default behavior (as switching lights on when lumonosity moves down) you should delete since it usually does not fit your needs.

I didn't find how to do it with the official Hue App. I did it with the alternativ phone App "all 4 hue" (**google Play** and **AppStore**): go to "Show rules" and choose "delete" for each.

Build a simple flow

Drag'n drop a hue brightness node:

Edit it: The previously defined bridge should already be set.

Click on:

Nothing seems to be done but then a pull down menu contains the sensors:

Choose the one and validate the form.

Only devices that were previously added with the hue app on your phone with the Hue app will be present in the menu.

An easy flow is to display the brightness:

Edit the display node as follow:

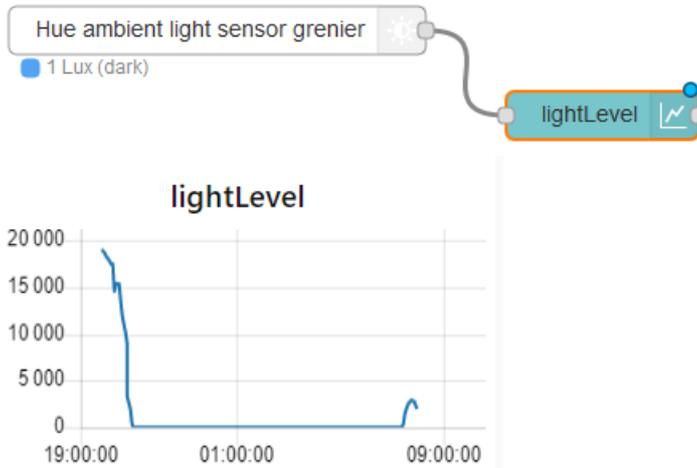| ☰ Template	The brightness is {{{payload}}}
🏷 Name	Name

You get the data after few minutes:

🐞 **debug**

4/28/2019, 10:48:15 PM   node: 3ebf67da.752ec8

The brightness is 0

💬 *Yes it's dark outside..*

Another easy flow is to make a chart using the nodeRED dashboard (should be manually installed in the palette http://thethingbox.io/docs/contrib-ui.html).

⊙ Advanced

Here is a typical output if you display the whole msg:

```
{
   "payload":0,
   "battery":98,
   "updated":"2019-05-01T19:55:24+00:00",
   "data":{
      "lux":1,
      "dark":true,
      "daylight":false
   },
   "info":{
      "id":"4",
      "uniqueId":"00:17:88:01:02:10:2c:06-02-0400",
      "name":"Hue ambient light sensor grenier",
      "type":"ZLLLightLevel",
      "softwareVersion":"6.1.0.18912",
      "model":{
```

```json
        "id":"SML001",
        "manufacturer":"Philips",
        "name":"Hue Motion Sensor",
        "type":"ZLLPresence"
      }
    },
    "_msgid":"8c6f1c52.0900d"
}
```

Temperature Sensor

The Temperature Sensor is embedded into the motion sensor.

⟲ Hue prerequisite

Pairing

The Hue nodes we use in this tutorial will only work if you have already configured your Hue Bridge and added the Hue devices to it as described in the Hue documentation.

For more informations about how to do it, go **on the Phillips Web site**.

This also may help: https://www.makeuseof.com/tag/philips-hue-motion-sensor-guide/

💬 Tip: reset of the sensor is done by pressing and holding in the reset button on the back of the sensor until the LED becomes green then oprange (usually 5 seconds). Press the reset again to put sensor into pairing mode.

Remove default behavior

The sensor comes with a default behavior (as switching lights on when lumonosity moves down) you should delete since it usually does not fit your needs.

I didn't find how to do it with the official Hue App. I did it with the alternativ phone App "all 4 hue" (**google Play** and **AppStore**): go to "Show rules" and choose "delete" for each.

Build a simple flow

Drag'n drop a hue temperature node:
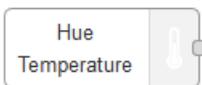
Edit it: The previously defined bridge should already be set.

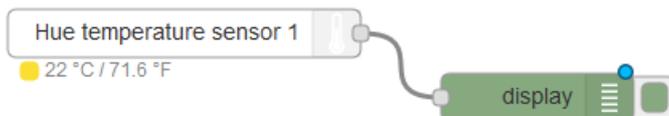Nothing seems to be done but then a pull down menu contains the sensors:

Choose the one and validate the form.

 Only devices that were previously added with the hue app on your phone with the Hue app will be present in the menu.

An easy flow is to display the temperature:
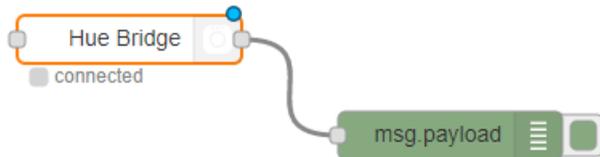

You get the data after few minutes:

Advanced

Here is a typical output if you display the whole msg:

```
xxxx
```

The Bridge node

The Bridge node can be used to monitor the Hue Network.

Here is the node output.

If there is an error, every 10 s, the output is (the message may vary):

```
{
"error":true,
"message":"Error: Huejay: connect EHOSTUNREACH 192.168.0.193:80"
}
```

If there is no error:

```
{
   "error":false,
   "id":"001788FFFEA4E968",
   "name":"DemoRoomBridge",
   "factoryNew":false,
   "replacesBridgeId":null,
   "dataStoreVersion":"76",
   "starterKitId":"",
   "softwareVersion":"1901181309",
   "apiVersion":"1.29.0",
   "zigbeeChannel":11,
   "macAddress":"00:17:88:a4:e9:68",
   "ipAddress":"192.168.0.178",
   "dhcpEnabled":true,
   "netmask":"255.255.255.0",
   "gateway":"192.168.0.254",
   "utcTime":"2019-02-27T14:17:00",
   "timeZone":"Europe/Paris",
   "localTime":"2019-02-27T15:17:00",
   "portalServicesEnabled":true,
   "portalConnected":true,
   "linkButtonEnabled":false,
   "touchlinkEnabled":false,
   "autoUpdatesEnabled":true,
   "model":{
      "id":"BSB002",
      "manufacturer":"Philips",
      "name":"Hue v2"
   }
}
```

This allows to build a monitoring of the Hue network.