

物联网协议 **MQTT** 打造智能家居指南：**Homematic + Node-RED + Node-RED-Dashboard**

Phodal Huang

October 24, 2017

## 目录

步骤 1: 软件安装 . . . . .	3
安装 Mosquitto . . . . .	3
Node.js . . . . .	4
步骤 2: 配置 pm2 . . . . .	4
检察日志文件 . . . . .	5
步骤 3: 使用 mqtt-admin . . . . .	5
步骤 3: NODE-RED 流示例 . . . . .	6
添加一个 MQTT 输入节点 . . . . .	6
处理传入的消息 . . . . .	7
等待 5 分钟 . . . . .	8
发送信息关灯 . . . . .	8
连线并启动流程 . . . . .	9
步骤 4: Node-RED-Dashboard 示例 . . . . .	10

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/mqtt-node-red-dashboard-mqtt-admin-build-m>

我使用了一个最小的 Debian Jessie (8.7) 64 位虚拟机创建、测试了这个玩法,但它也应用于其它基于 Debian 的 Linux 发行版,如 Armbian、Ubuntu。

要完成这个玩法,你应该有基本的 Linux 系统管理知识。此外,您应该已经知道 mqtt-smarthome 的概念,以及如何工作,如果尚未完成,我建议您阅读会 [Talk](#) 和 [架构概述](#)。

如果您对此玩法,有任何疑问或遇到问题,请创建一个 [Issue](#), 英语、德语都可以。

## 步骤 1: 软件安装

首先,我们需要安装一系列的软件,并配置他们。

### 安装 Mosquitto

```
1 wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
2 sudo apt-key add mosquitto-repo.gpg.key
3 cd /etc/apt/sources.list.d/
4 sudo wget http://repo.mosquitto.org/debian/mosquitto-jessie.list
5 cd
6 sudo apt-get update
7 sudo apt-get install mosquitto mosquitto-clients
```

### 配置

编辑 Mosquitto 配置文件,以允许在 8080 端口上进行 websocket 连接。这是为了那些基于浏览器的应用程序能连接到代理,我们稍后再回来讨论这具问题。

使用您喜欢的编辑器,将以下 4 行添加到文件 /etc/mosquitto/mosquitto.conf 中:

```
1 listener 1883 0.0.0.0
2 protocol mqtt
3
4 listener 8080 0.0.0.0
5 protocol websockets
```

之后,重启 Mosquitto:

```
1 sudo systemctl restart mosquitto
```

## Node.js

我建议使用 **n** 来安装 **Node.js**。这样可以更新 **Node.js**，甚至并行使用不同的 **Node.js** 版本。为此，我们将需要 **git** 来进行安装：

```
1 sudo apt-get install git make
2 git clone https://github.com/tj/n
3 cd n
4 sudo make install
5 sudo n stable
```

## **pm2**

**pm2** 是一个 **Node.js** 的进程管理器。我们用它来启动系统所需要的软件，以及管理日志文件。

```
1 sudo npm install -g pm2
```

## **hm2mqtt.js**

这是一个将您的 **Homematic CCU** 与 **MQTT Broker** 对接的软件。

```
1 sudo npm install -g hm2mqtt
```

## **Node-RED**

**Node-RED** 是一个令人敬畏的软件，可以让您连接事物 (**things**)，并通过图形用户界面完成自动化规则。

```
1 sudo npm install -g node-red
2 sudo npm install -g node-red-dashboard
```

创建一个用户

我建议创建一个用户，在其下运行所有 **mqtt-smarthome** 相关软件。

```
1 sudo adduser smarthome --shell /bin/bash
2 sudo usermod -aG sudo smarthome
```

现在注销，并使用新创建的用户 **smarthome** 重新登录。

## 步骤 2: 配置 **pm2**

首先，我们要将 **pm2** 作为一个在系统启动时自动启动的服务。然后，我们告诉 **pm2** 它应该为我们启动那些软件。在下面的例子中，替换你的 **CCU** 的 **IP** 地址。最后的 **save**

命令, 会将其持久化在 `pm2` 的设置中。

```
1 sudo -E pm2 startup -u smarthome
2 pm2 start node-red
3 pm2 start hm2mqtt -- --ccu-address 192.168.1.100
4 pm2 save
```

如果你执行 `pm2 list`, 将会看到一下如下的输出结果:

```
1 | hm2mqtt | 0 | fork | 3605 | online | 0 | 11s | 0% | 90.9
2 | node-red | 0 | fork | 3573 | online | 0 | 39s | 0% | 77.1
```

现在, 软件已经安装完成。

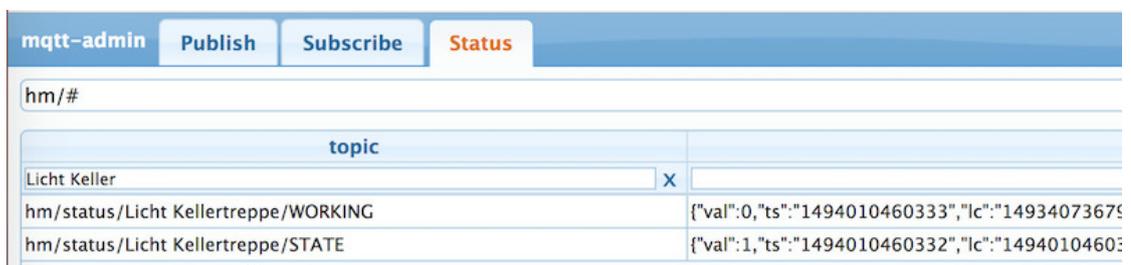
### 检察日志文件

要检查 `hm2mqtt` 和 `node-red` 是否一切正常, 您可以使用 `pm2` 显示其日志文件, 执行 `pm2 logs hm2mqtt`。更多选项, 请参阅 `pm2 help`。

### 步骤 3: 使用 `mqtt-admin`

您可以使用工具 `mqtt-admin`, 来查看您的 MQTT 代理发生了什么事。此工具不需要安装, 您只需在浏览器中打开 <https://hobbyquaker.github.io/mqtt-admin>, 并连接到您的代理。在连接屏幕中, 输入运行 Mosquitto 的主机 IP 地址及其端口 `8080`, 然后单击保存设置 (Save Settings)。当 `connection` 对话框消失时, 您就连接到了 Mosquitto。然后更改标签为 'Status', 并在最上面的输入框中输入 `hm/#`。您应该看到由 `hm2mqtt` 发布的消息的 MQTT 主题列表。您现在可以过滤出一个特定的 Homematic 传感器或执行器, 做一些事情 (例如打开一个灯), 看看会发生什么。

这里有一个我的地下室灯的屏幕截图:



当我打开和关闭这个灯时, 我可以看到在主题 `hm/status/Licht Kellertreppe/STATE` 下 `val` 属性的改变。

您现在还可以通过 MQTT 打开和关闭该灯，使用“Publish”选项并尝试不同的有效载荷（0 -> 关闭，1 -> 打开）

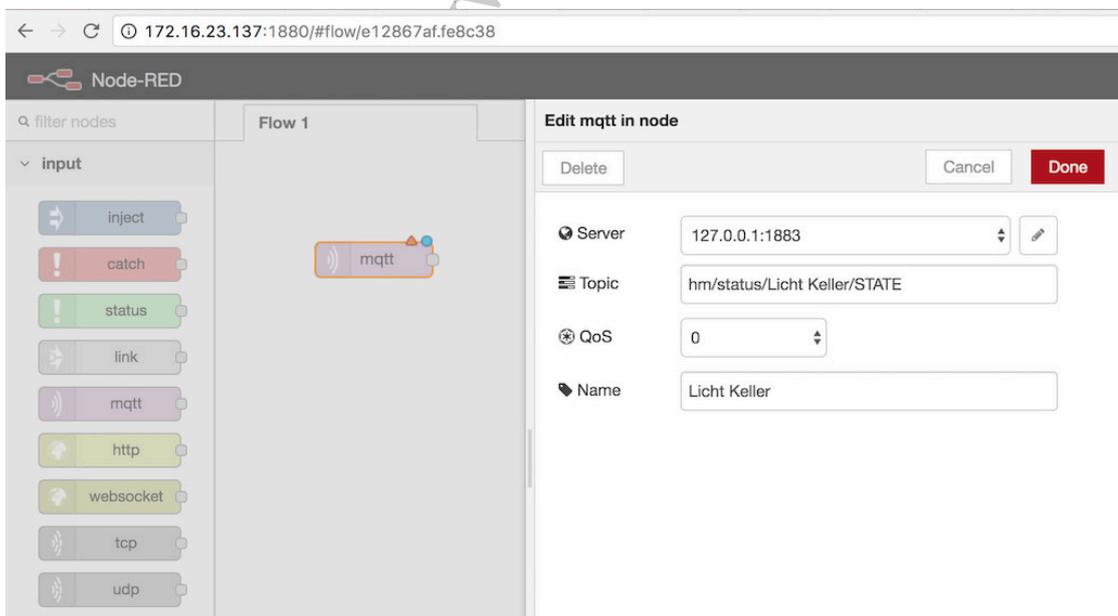


请注意，我们不会在 hm2mqtt 的同一主题上发布状态。我们使用 hm/set/... 来打开和关闭灯，而非 hm/status....

### 步骤 3: NODE-RED 流示例

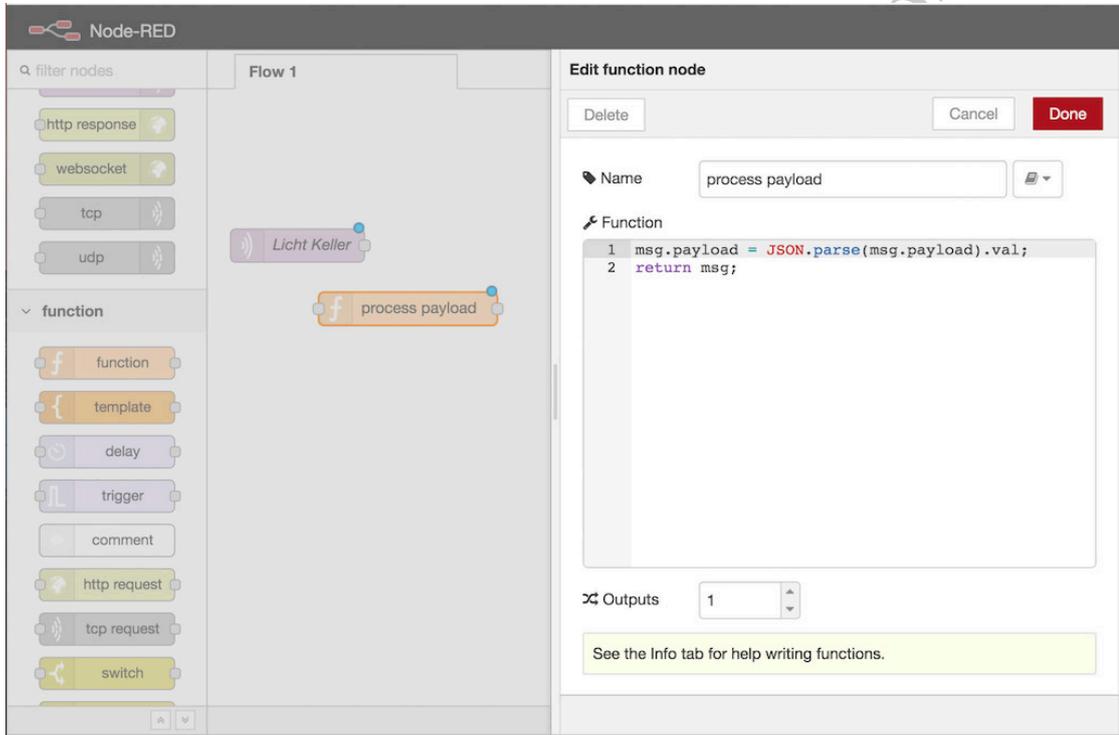
添加一个 MQTT 输入节点

Node-RED 可以通过 http 访问 1880 端口，所以在浏览器中打开 <http://:8080/>。在这个简单的例子中，我将创建一个流程（flow），在开机后 5 分钟切换（开头）我的地下室灯。为此，我们首先需要有一个 mqtt 输入节点。从左到右拖动它。双击它以打开其配置对话框。点击 IP 地址旁边的小铅笔来“添加一个新的 MQTT 节点”。在打开的对话框中，只需输入 127.0.0.1 作为服务器，然后单击“Add”。回到节点配置，我们必须输入一个主题，我再用我的地下室灯作为例子：

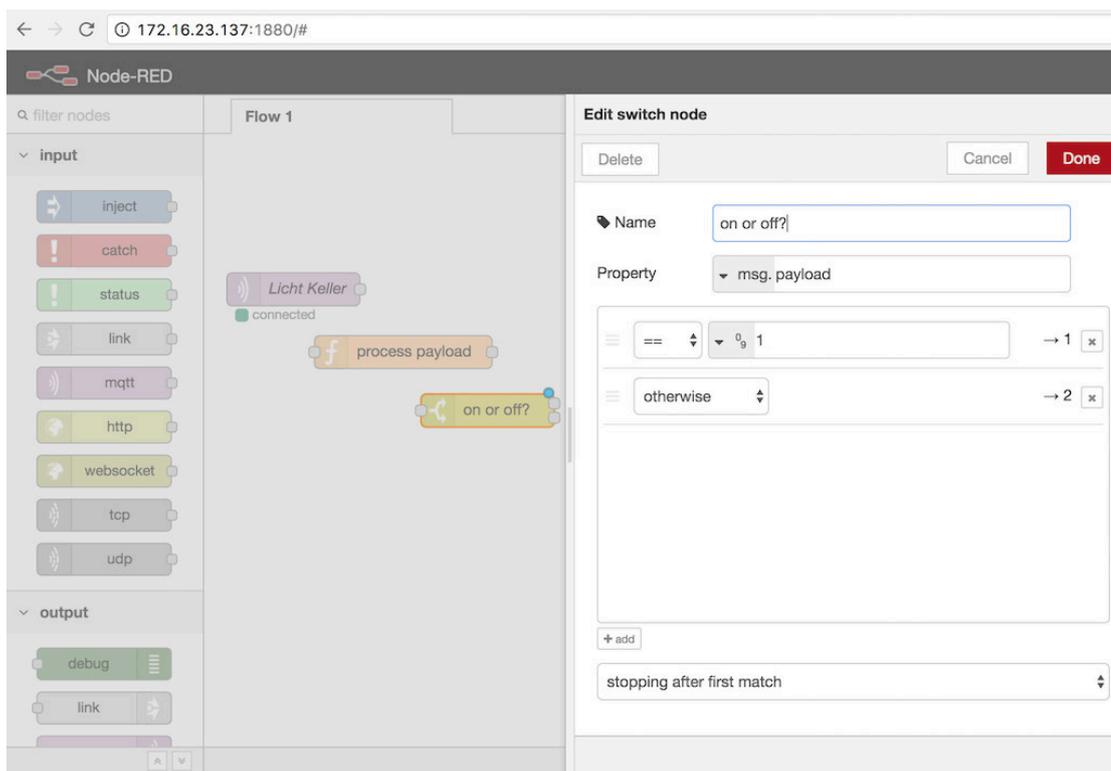


处理传入的消息

因为 `mqtt-smarthome` 确认软件在 `val` 属性中发布一个有效的 `JSON` 载荷和实际值，我们需要一些处理来获取光的实际状态。为此，我们使用一个函数节点，在这个节点中可以使用 `javascript`：

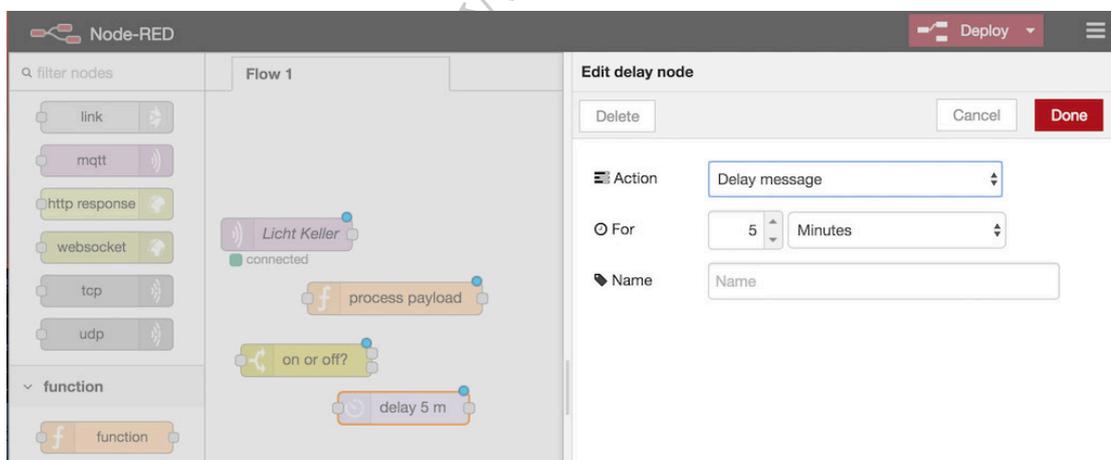


这个节点将只输出 `val` 发生的一个内容，现在我们可以为两个不同输出的交换节点添加一个：如果 `val` 为 `1`，则发出一个消息，如果 `val` 为 `0`，则会发出另外一个消息。



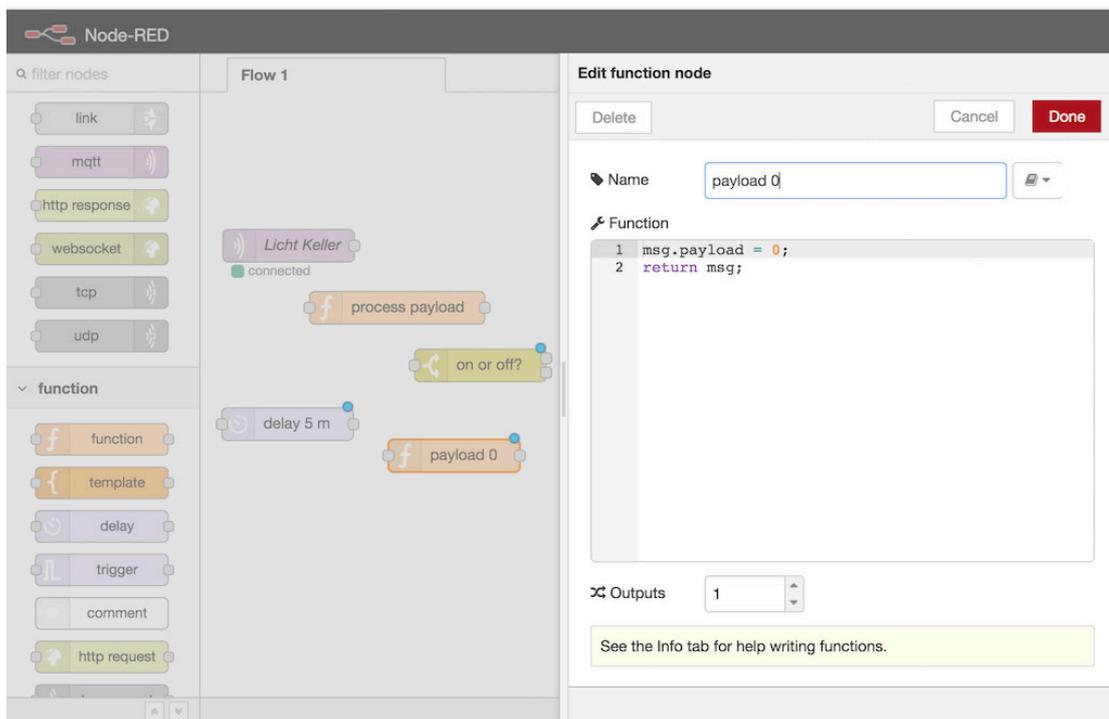
等待 5 分钟

现在，我们添加一个延迟节点，延迟消息转发 5 分钟。

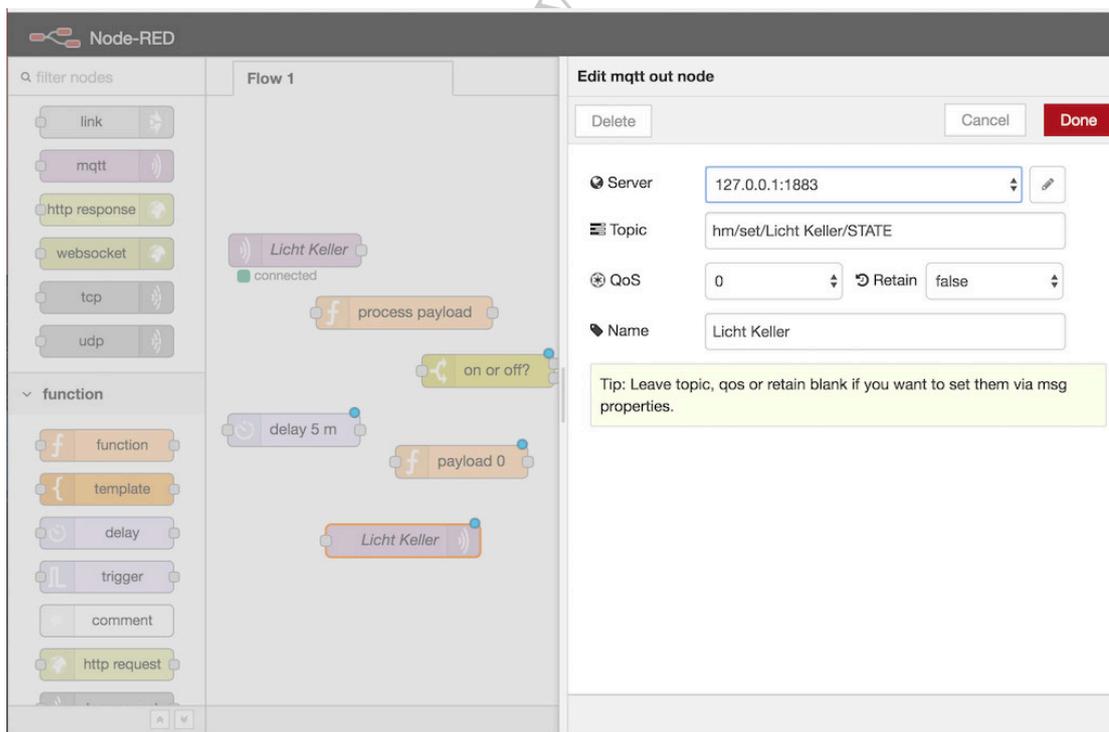


发送信息关灯

在另外一个功能节点上，我们将消息有效载荷设置为 0（我们要关闭灯泡）。

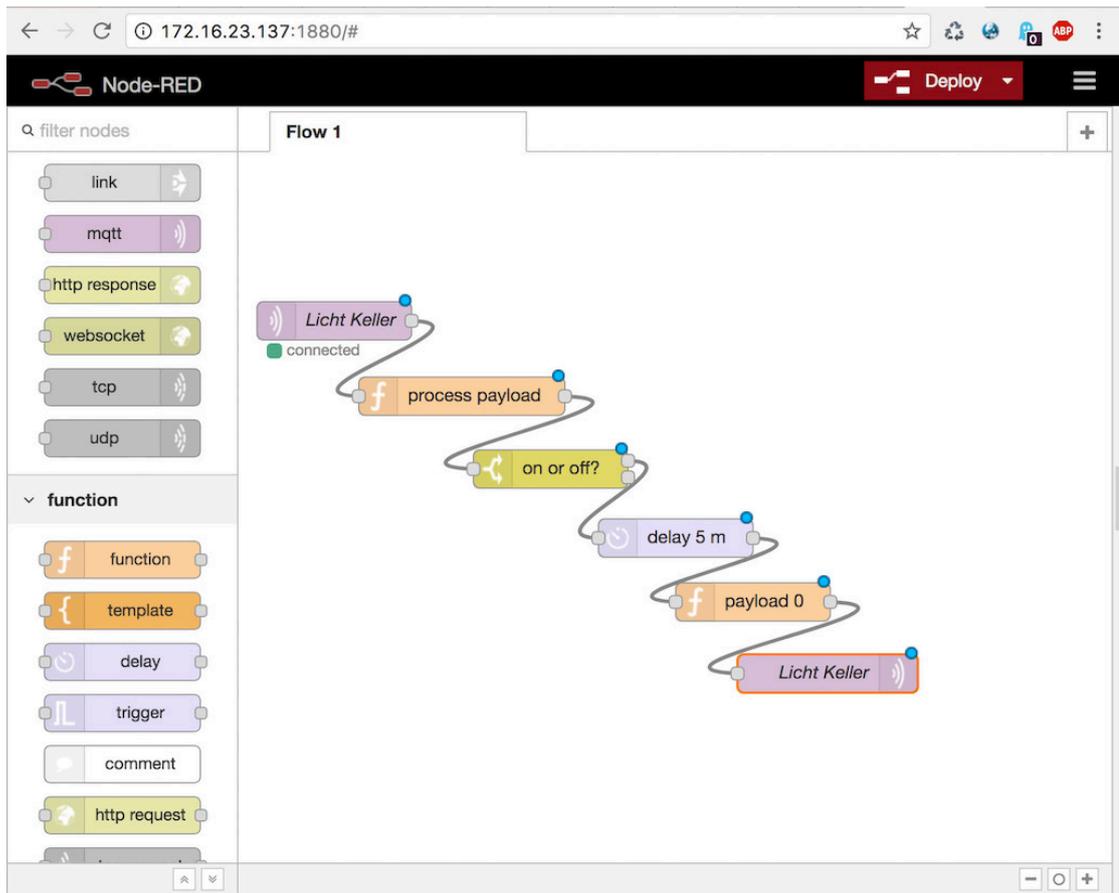


并且，使用一个 **mqtt** 输出来将命令发送到 **hm2mqtt**。再次注意要使用正确的主题，这里我们再次需要 `hm/set/...`。



连线并启动流程

最后一步是将所有节点连接在一起，然后按右上角的红色“Deploy”按钮激活流程。

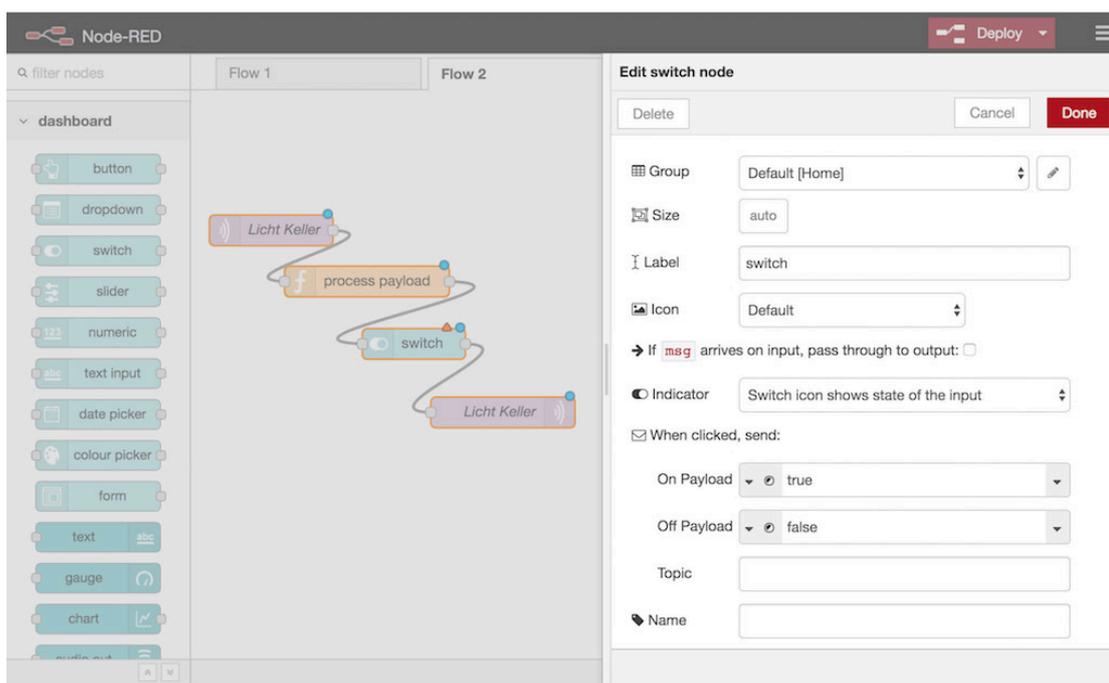


现在你可以通过打开灯光来测试你的第一个流程，等待 5 分钟。如果你忙的话，也可以减少延迟时间。

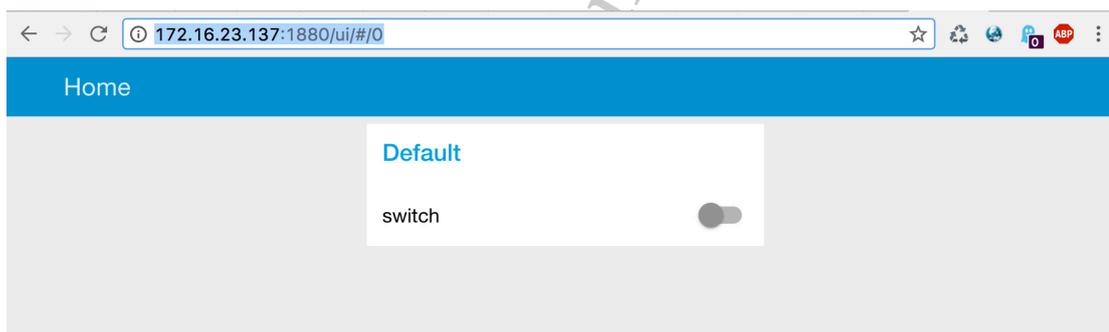
请注意，这个流程并不意味着它是一个真正的智能家居规则，仅仅是简单的一个例子来说明这些原则。它可以做得更好，可以用较少的节点来完成，现在您需要了解如何使用 Node-RED :-)

#### 步骤 4: Node-RED-Dashboard 示例

Node-RED-Dashboard 是一个非常棒的工具，可以轻松创建能在任何浏览器、移动设备上，使用的用户界面。以下示例，将仅显示如何使用地下室灯的开关，来创建一个网站。我们可以从我们第一个流程示例中重用一些节点，您可以复制 `mqtt` 输入节点、`mqtt` 输出节点和“进程有效负载”（`process payload`）功能节点。在“进程有效载荷”（`process payload`）和 `mqtt` 输出节点之间，我们插入一个仪表盘切换节点。请注意，您必须删除 `if msg arrives on input, pass through to output` 复选标记。



在 <http://:1880/ui> 上部署这个新的流节点之后，你的浏览器将会看到这个结果：



现在有一个地下室灯的用户界面，我们可以打开和关闭它。这只是一个非常简单的例子，Node-RED-Dashboard 让你拥有更多的可能性 - 玩得开心! ;-)

原文地址：[Getting started with mqtt-smarthome - Homematic, Node-RED and Node-RED-Dashboard](https://www.wandianshenme.com/play/mqtt-node-red-dashboard-mqtt-admin-build-m)

原文链接：<https://www.wandianshenme.com/play/mqtt-node-red-dashboard-mqtt-admin-build-m>