

## Yeelink简介

最近接触了一个物联网平台Yeelink。这个平台可以使用户通过Http请求的方式上传或查询数据，通过这样的方式可以使用Web方式访问Yeelink平台，通过这种方式获取传感器信息或是控制执行设备。下面就来说说Yeelink中控制执行设备的Http请求格式。

Yeelink平台提供了一些例子，但是硬件平台主要是围绕Arduino。如果想通过STM32辅以uIP或lwIP的方式实现执行设备的远程控制，这就需要单片机/嵌入式系统发送格式正确的HTTP请求。查阅了Yeelink的网页，API函数写得非常详细。对于一些能力受限的设备，正确处理好HTTP及其外围相关协议并非易事。

## 传感器分类

- 设备和设备ID

设备可以包含多个传感器，本例中设备ID为1949。

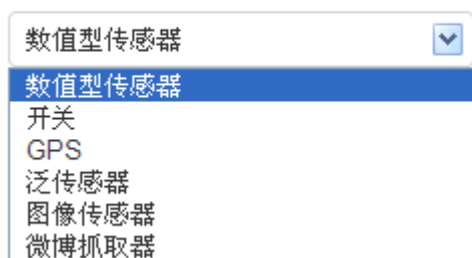
- 传感器和传感器ID

传感器为输入设备或者输出设备，例如开关和等，本例中设备传感器ID为2511。

- 数据类型

数据类型包括数据点datapoint和图像photo，本例中数据类型为数据点datapoint

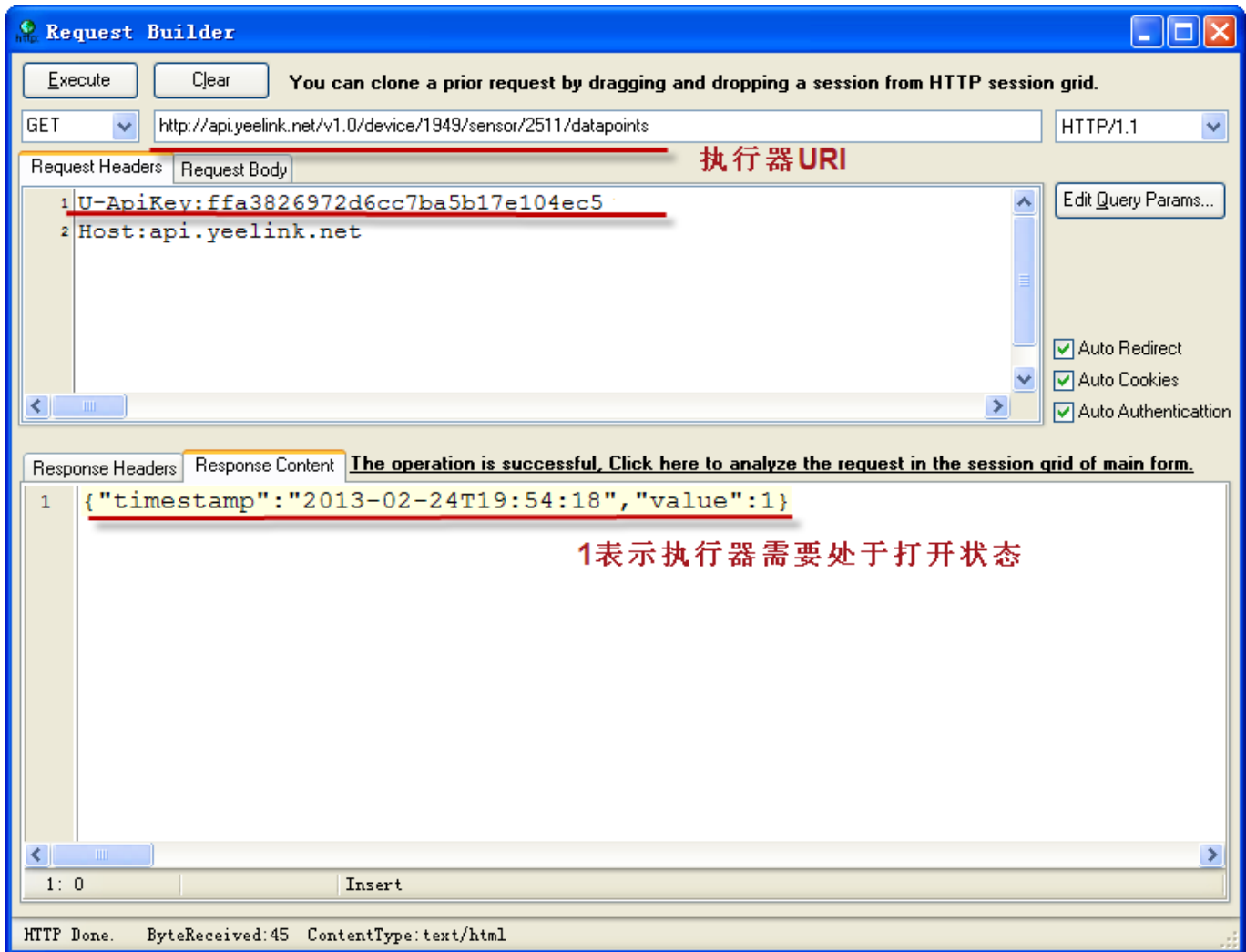
类型



在Yeelink中，首先需要创建一个设备（Device）。这个设备可以理解为PLC，该设备可能具有多种功能，例如传感器输入通道，或是执行器输出通道。一般在PLC中传感器输入称之为AI，执行器控制称之为DO。但是非常遗憾的是Yeelink把执行器的控制端口定义在传感器类型中，在传感器类型中有开关的子类型，如果有一个变量存放该开关的状态的话，这个变量不是0就是1。可能Yeelink的开发者可能不清楚开关和灯并不是一类设备，开关为输入设备，灯为输出设备或执行设备。

## 查询开关状态

查询开关状态需要明确开关的URI和本用户的API-Key。可以通过Http调试软件发送Http请求以获得此时开关量的状态。例如以下多图所示。



```
GET /v1.0/device/1949/sensor/2511/datapoints HTTP/1.1
U-ApiKey: ffa3826972d6cc7ba5b17e104ec5XXXX
Host: api.yeelink.net
.Cookie: CAKEPHP=n2n4g8q1j8om2326eu0cfsegg0
```

## HTTP请求

- 1) 用GET方法，查询开关状态。
- 2) URI中包括设备编号和传感器编号及类型，设备编号为1949，传感器编号为2511，传感器类型为 datapoint。
- 3) U-ApiKey，相当于用户的操作密码。

**测试执行器**

**状态URL**  
http://api.yeelink.net/v1.0/device/1949/sensor/2511/datapoints

**控制操作**  
按动图标以改变开关的状态

**JSON响应:**  
`{"timestamp": "2013-02-24T19:54:18", "value": 1}`

**测试执行器**

**状态URL**  
http://api.yeelink.net/v1.0/device/1949/sensor/2511/datapoints

**控制操作**  
按动图标以改变开关的状态

**JSON响应:**  
`{"timestamp": "2013-02-24T20:05:44", "value": 0}`

## HTTP响应

HTTP响应采用JSON

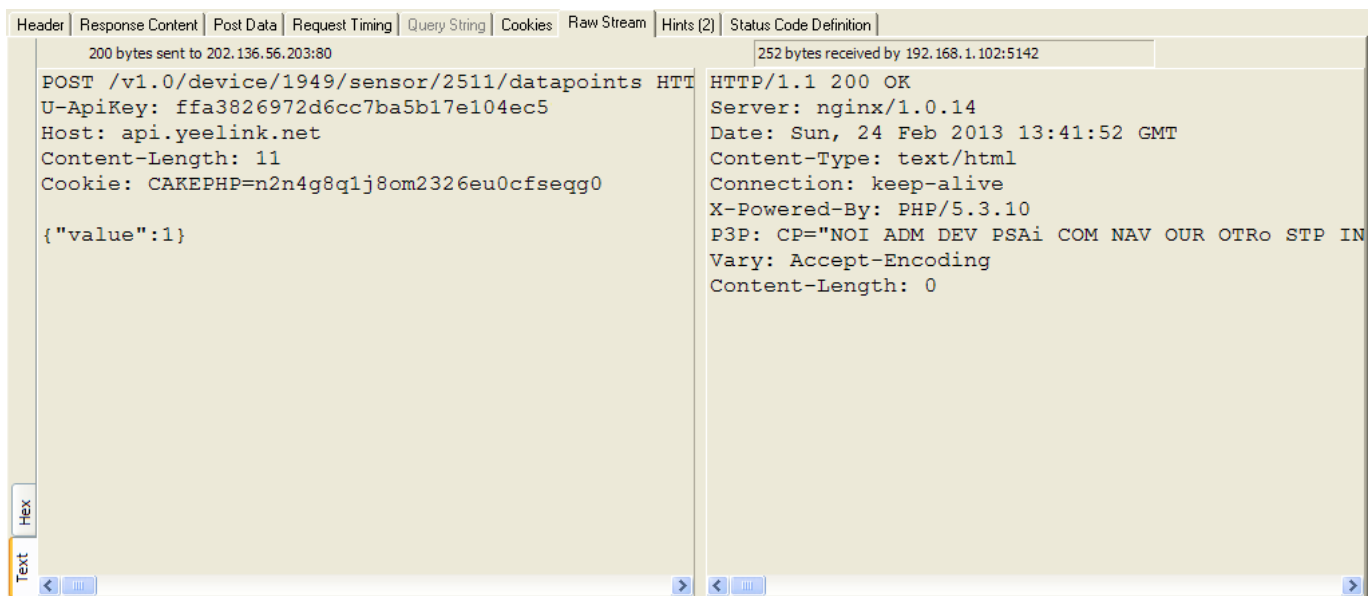
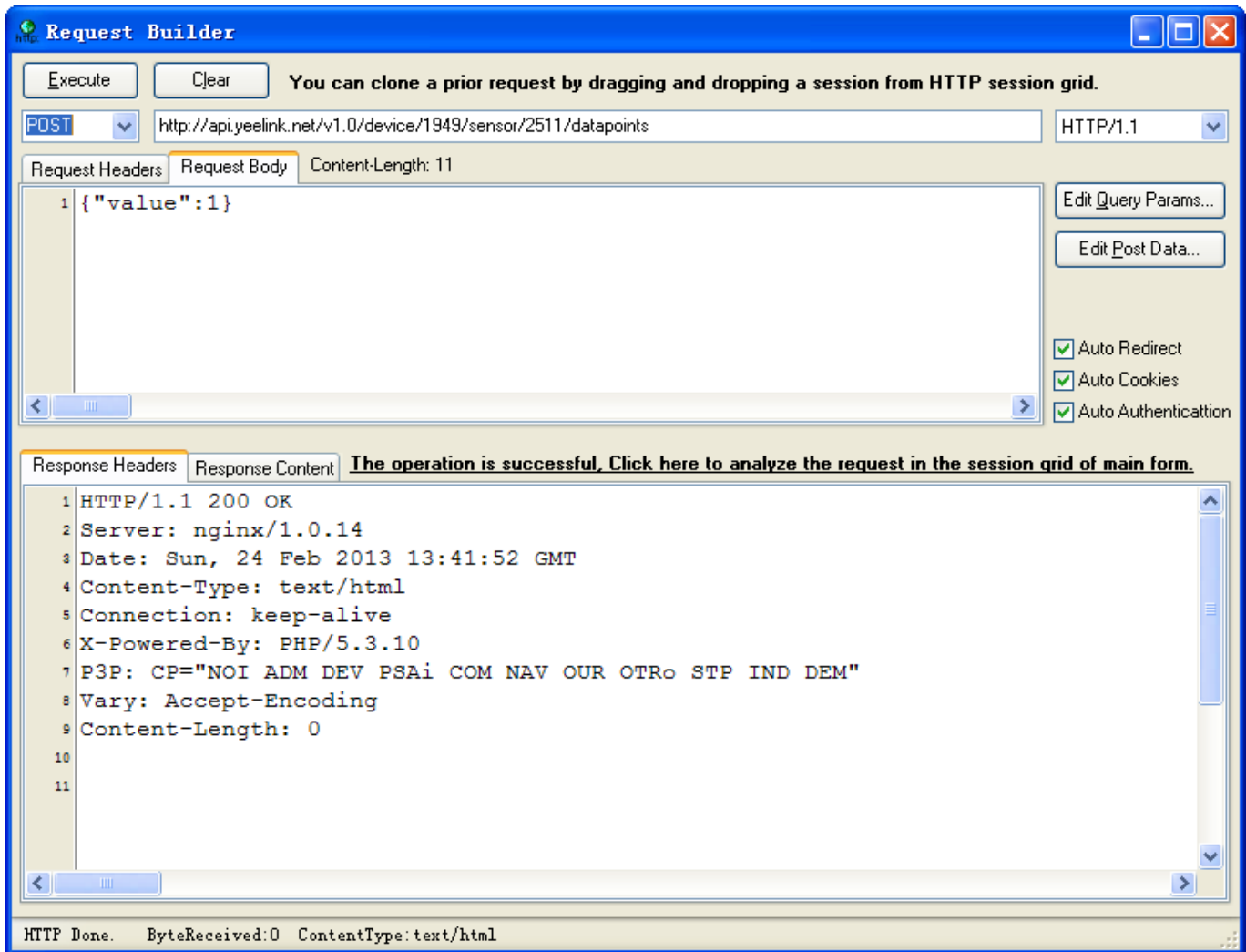
格式，其中包括查询此开关状态的时间和此时的开关状态结果，1表示开关闭合，0表示开关断开，根据开关的状态可以控制响应的执行设备，例如LED灯，或电机等。

```
HTTP/1.1 200 OK
Server: nginx/1.0.14
Date: Sun, 24 Feb 2013 12:07:24 GMT
Content-Type: text/html
Connection: keep-alive
X-Powered-By: PHP/5.3.10
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Vary: Accept-Encoding
Content-Length: 45

{"timestamp": "2013-02-24T20:05:44", "value": 0}
```

## 设置开关状态

除了查询开关状态之外，还可以设置开关状态。例如通过Android手机发送Http请求至yeelink服务器，修改开关的value。另一端，嵌入式系统不断查询该value结果，若该结果为1则控制响应的执行器启动，若该结果为0则关闭响应的执行器。



```
POST /v1.0/device/1949/sensor/2511/datapoints HTTP/1.1
U-ApiKey: ffa3826972d6cc7ba5b17e104ec59fa3
Host: api.yeelink.net
Content-Length: 11
Cookie: CAKEPHP=n2n4g8q1j8om2326eu0cfsegg0

{"value":1}
```

## HTTP请求

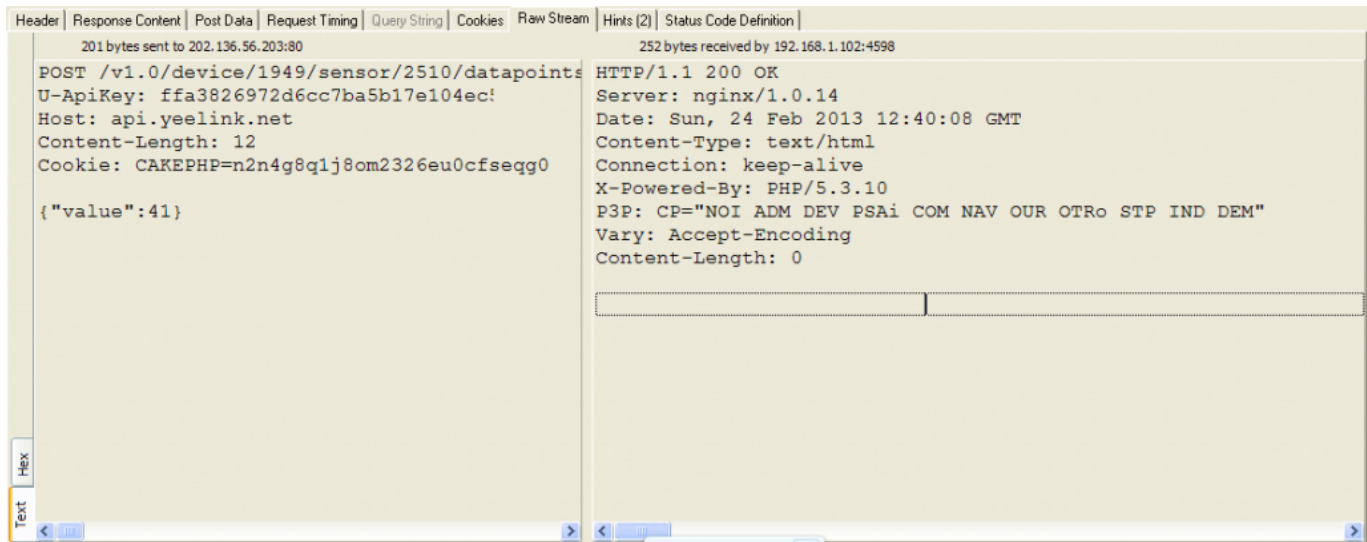
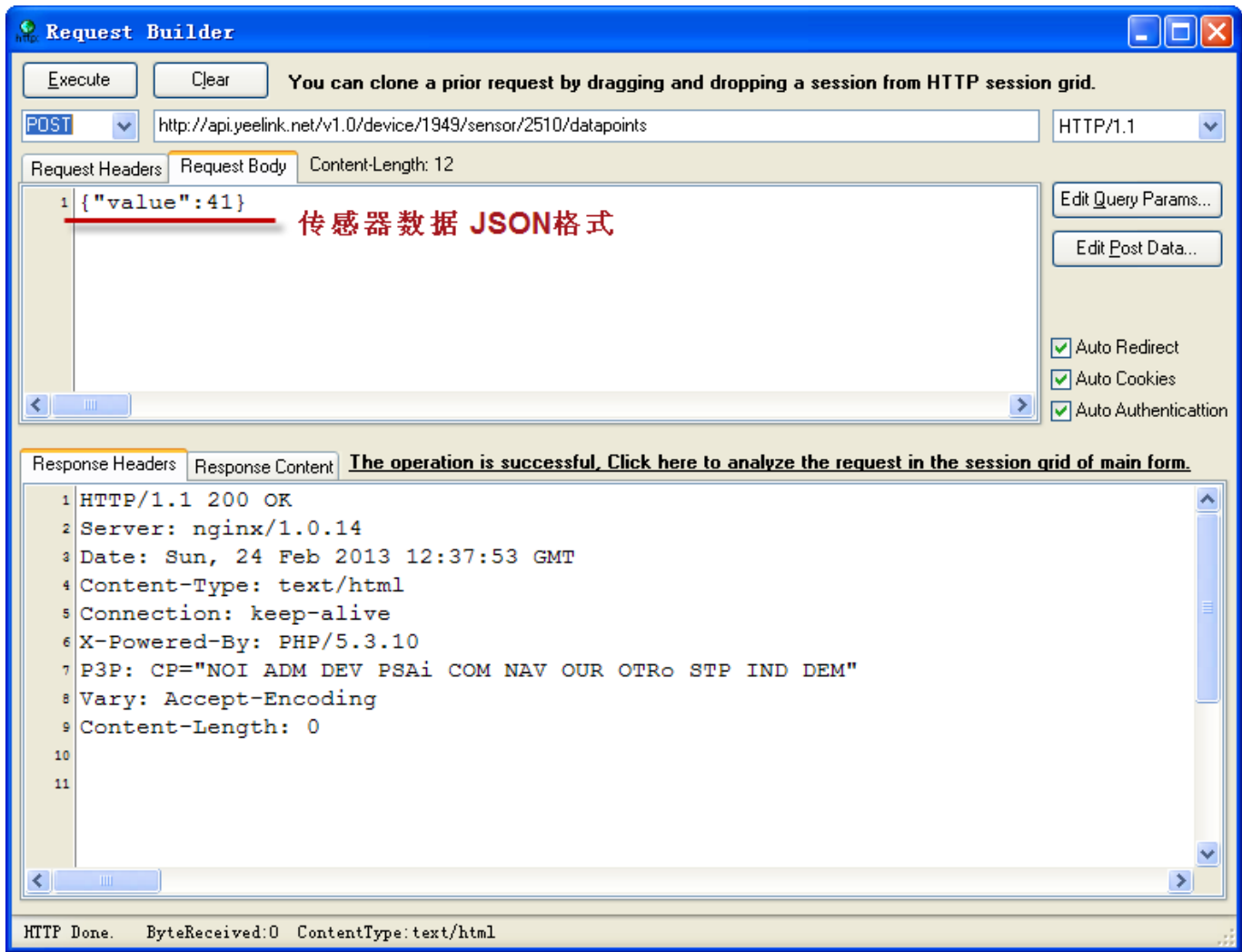
- 1) 使用POST方法。
- 2) URI中包括设备编号和传感器编号及类型，设备编号为1949，传感器编号为2511，传感器类型为datapoint。
- 3) U-ApiKey，相当于用户的操作密码。

## HTTP响应

```
HTTP/1.1 200 OK
Server: nginx/1.0.14
Date: Sun, 24 Feb 2013 13:41:52 GMT
Content-Type: text/html
Connection: keep-alive
X-Powered-By: PHP/5.3.10
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Vary: Accept-Encoding
Content-Length: 0
```

## 推送传感器数据

推送传感器数据的过程也类似。和查询执行器不同的是，推送传感器数据使用POST方法，在HTTP请求中包含了传感器的采集数据，以JSON格式存放。具体操作如下图所示。

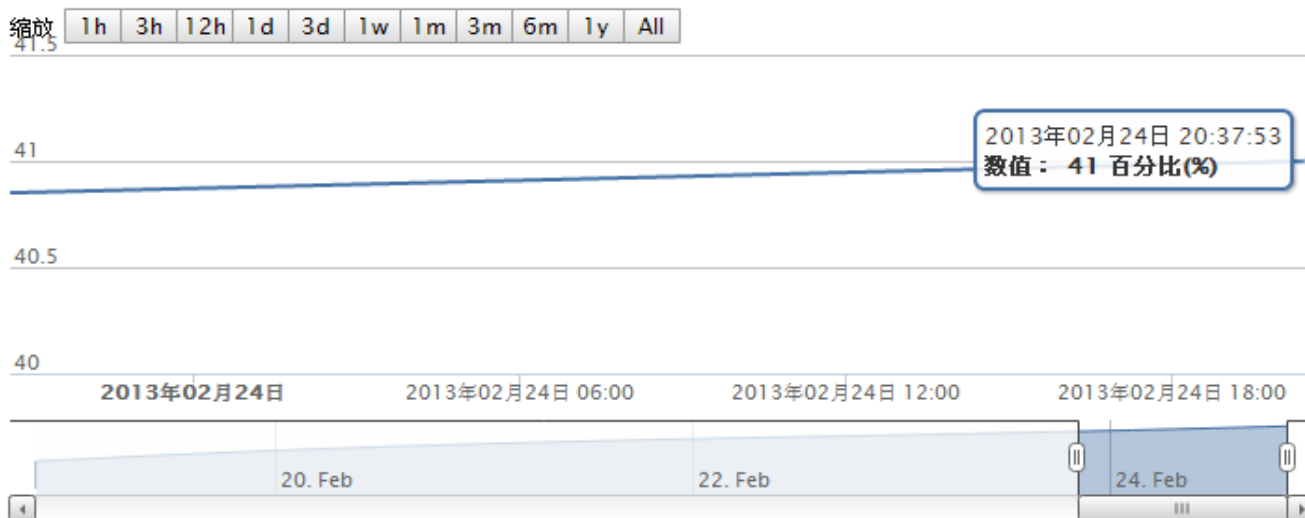


## HTTP请求

```
POST /v1.0/device/1949/sensor/2510/datapoints HTTP/1.1
U-ApiKey: ffa3826972d6cc7ba5b17e104ecXXXX
Host: api.yeelink.net
Content-Length: 12
Cookie: CAKEPHP=n2n4g8qlj8om2326eu0cfsegg0
```

```
{"value":41}
```

- 1) 使用POST方法
- 2) URI中包括设备编号和传感器编号及类型，设备编号为1949，传感器编号为2510，传感器类型为datapoint
- 3) U-ApiKey，相当于用户的操作密码
- 4) 请求内容包括传感器的数据，采用{"value":xx.xx}格式。



从图中的结果可以看出，数据41被成功上传到Yeelink服务器。

## 嵌入式应用的问题

嵌入式系统需要构成HTTP请求，还需要一个TCP首部，IP首部和以太网首部。在IP首部中需要明确Yeelink服务器的IP地址，但是在实际的过程中仅明确该服务器的域名。从域名到IP地址需要有一个DNS过程。如果不需要这个DNS过程也是可以发送HTTP请求的，需要在IP首部中手动添加yeelink服务器的IP地址。之后的工作便是利用STM32和uIP或lwIP，甚至是自己编写的以太网协议栈来测试HTTP请求的发送。

From: <http://www.simpleteam.com/> - SimpleTeam—无线技术无限

Permanent link: [http://www.simpleteam.com/doku.php?id=iot:yeelink\\_http\\_sepecification](http://www.simpleteam.com/doku.php?id=iot:yeelink_http_sepecification)

Last update: 2015/06/10 09:42