

# 飞比物联网浏览器软件 V1.0

## 用户手册

深圳市飞比电子科技有限公司

# 目 录

1、简介	
1.1 软件背景.....	1
1.2 功能简介.....	1
2、软件安装.....	5
3、界面介绍	
3.1 主界面.....	7
3.2 配置界面.....	8
4、基本操作	
4.1 启动软件 .....	9
4.2 打开配置文件.....	9
4.3 连接方式配置.....	9
4.4 连接.....	10
4.5 数据分析.....	10
5、场景设计基础	
5.1 进入高级设定状态:.....	12
5.2 修改节点位置及大小: .....	12
5.3 修改节点: .....	12
6、创建应用场景实例	
6.1 设置界面背景.....	15
6.2 创建传感器.....	18
6.3 创建控制对象.....	22
6.4 设置控制指令.....	23
7、通过 LUA 脚本扩展场景功能	
7.1 LUA 脚本简介.....	25
7.2 如何启用 LUA.....	26
8、飞比云服务平台介绍及其 API 详解	
8.1 背景.....	28
8.2 系统结构.....	28
8.3 功能简介.....	29
8.4 帐号申请及简单测试.....	29
8.5 API 详解.....	31
9、飞比云传感协议介绍	
9.1 数据格式.....	34
9.2 格式详解.....	34
9.3 示例分析.....	36
10、FIT Explorer 的“网关”功能介绍	
10.1 网关功能设置.....	37
10.2 远程监控功能.....	38

# 1、简介

## 1.1 软件背景

自 2009 年 8 月温总理提出“感知中国”以来,物联网被正式列为国家五大新兴战略性新兴产业之一, 写入“政府工作报告”, 物联网在中国受到了全社会极大的关注。与此同时, 各个行业中对物联网技术的需求也迅速增长。然而, 物联网涉及到“感知”、“传输”与“应用”各个不同层次的技术, 其系统复杂度往往令中小型技术团队望而却步, 阻碍了创新的想法快速产品化的步伐。

飞比物联网浏览器, 结合了很多实际中的物联网应用项目, 将“应用层”需求进行了提炼, 只需进行简单的设置, 即可创建很多不同的应用场景。并可以作为无线传感网与互联网的网关使用, 大大缩短了项目开发周期。

何为“物联网浏览器”? 相信大家对“浏览器”这个词不会陌生, 它将互联网上的信息一包括图片、文字、声音、视频等, 转为成标准语言 (比如 html), 最终以一个统一的窗口呈现给您, 完成信息的获取或者交互。而“物联网浏览器”所采集的信息, 并不是虚拟信息, 而是真实世界的反映, 比如某个地方的温度、某一个物体的位置, 某个地方是否有人入侵等等; 所使用的语言, 以 FIT Explorer 为例是 xml; 进行交互的对象, 也不是虚拟的数据, 而是真实世界的对象, 比如一个水龙头、一盏电灯等等。

## 1.2 功能简介

下面我们以几个应用案例来说明软件功能:

“开心农场”的“偷菜”游戏, 也许不少人都玩过, 这是一个传统浏览器的典型应用, 在这类应用中, 所有的过程都是虚拟的数字, 真实世界中不会因为您的游戏而多一颗白菜的, 下面看下用飞比的物联网浏览器是如何“种菜”的。

### 例 1、飞比版“开心农场”—现代农业温湿度采集与灌溉控制系统



在这个应用中，菜园（真实，而非虚拟的）中，放了三个 zigbee 传感器监控点，每个监控点均可分别监控该点的温度、湿度信息，并且同时控制一个水龙头，采集信息经过 zigbee 网络传至电脑，可在本地通过 FIT Explorer 进行观测；另外，可以设置报警门限，比如当湿度低于 60% 时，自动打开水龙头进行灌溉，从而实现现代农业中的智能控制。同时，可以将此信息上传至“飞云服务器”，在世界的另外一个角落，通过 FIT Explorer 的网络连接功能进行观测。

### 例 2、“基于 3D 运动传感器的远程智能防盗系统”：



每辆车安装一个带 3D 运动传感器 Zigbee 节点，当车辆启动或受到外力干扰时，睡眠中的 Zigbee 节点会被唤醒，发送一个报警信号，停车场闸口处的采集节点，将报警信号通过 GSM/GPRS 网络发送至服务器，通过电脑版或者手机版 FIT Explorer 为车主报警。

### 例 3、停车场智能车位引导系统



停车场的每个车位上安装一个“车位传感器”，可以检测是否有车辆驶入该车位，Zigbee 节点负责将车位的状态通过 Zigbee 网络发送至停车场管理中心，以便在适当的位置指示车主每个区域的车位情况。如有必要，同样可以利用 zigbee-GPRS 网关将通讯范围进行扩展。

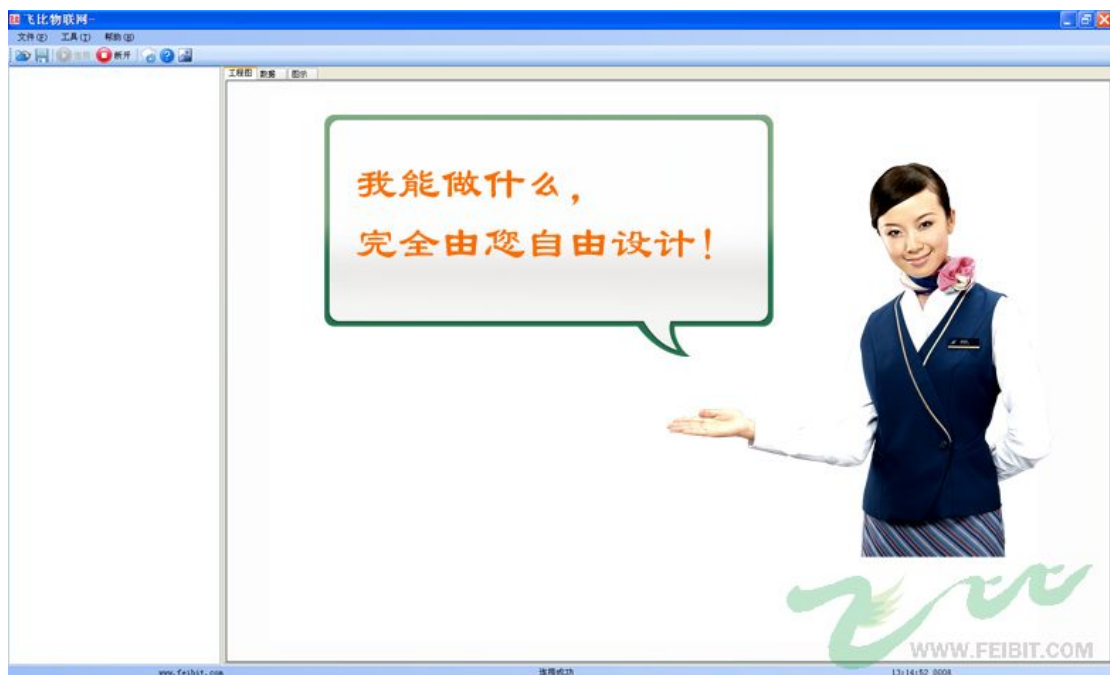
### 例 4、远程智能家居监控系统



在这套系统中：“3D 防盗门锁”安装在大门上，随时检测大门是否被撬动；“人体感应”装置放在阳台上，防止有扒手从阳台入侵；“温湿度采集”装置，随时监控室内温湿度，用以控制空调等设备，保证室内舒适；“电器监控”装置，跟踪每个电器的开关状态，避免不必要的电能消耗；“火灾报警”装置，将通知您家中是否着火，并在第一时间打开灭火装置；“煤气报警”使您和家人远离燃气泄露引起的事故。

区别于常见的“物联网演示平台”，FIT Explorer 并不是一个仅供演示的产品，上面的三个应用，只是我们提供的例子而已，它能做什么，是完全由您来自由设计的！包括传感器的名称、数据定义、背景图与监控对象图片和位置等等，而且支持 gif 动画功能！

### 由您来自由设计应用场景



## 2. 软件安装

1) 本软件需Microsoft .Net Framework 3.5支持。

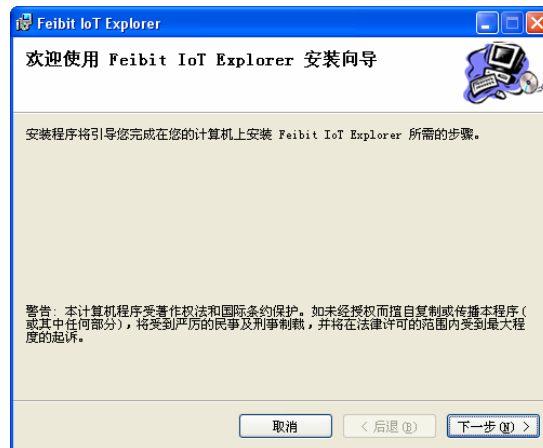
请安装附带的.Net 安装包 或 到微软网站下载

<http://www.microsoft.com/downloads/zh-cn/details.aspx?FamilyID=333325fd-ae52-4e35-b531-508d977d32a6>

2) 双击文件夹中的Setup图标，开始软件安装



3) 启动安装向导



4) 许可协议



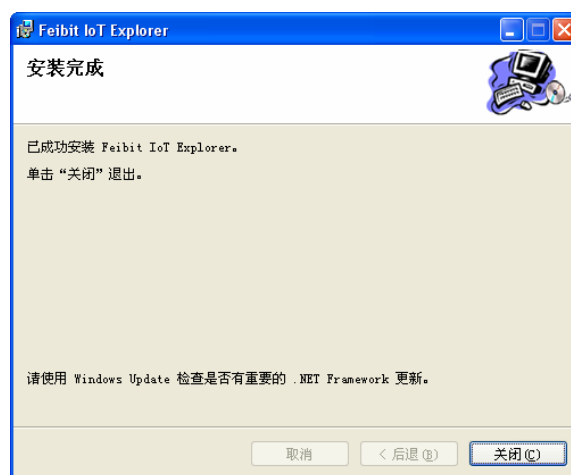
5) 安装文件夹



## 6) 确认安装



## 7) 安装完成



## 3. 界面介绍

飞比物联网浏览器原始界面非常简单，用户完全可以按照自己的想象来设计界面中的内容。

### 3.1 主界面

飞比物联网浏览器主界面有三个区域：

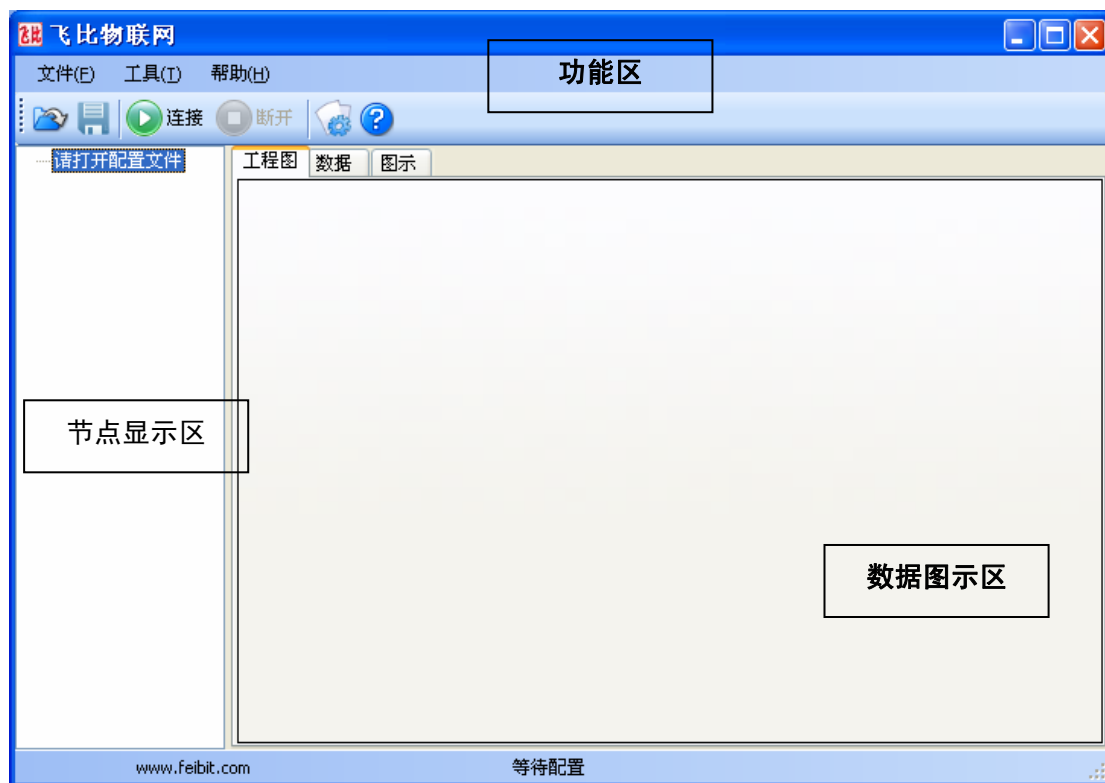


图 1: FIT Explorer界面图

#### 功能区：

功能区提供菜单选项及一些快捷功能按钮。

#### 节点显示区：

该区域显示配置文件描述的节点。

#### 数据图示区：

该区域对节点的分布进行展示，并显示其数据和图示。包括三个子项：工程图、数据、图示。

## 3.2 配置界面

配置界面可以对 FIT Explorer 的工作方式进行配置：

The screenshot shows the '配置' (Configuration) window with the '串口配置' (Serial Port Configuration) tab selected. The '启用串口连接' (Enable serial port connection) checkbox is checked. Under '地址' (Address), '地址确认方式' (Address confirmation method) is set to '自定义ID' (Custom ID). The '数据有效期' (Data validity period) is set to 20 seconds. The '网络配置' (Network Configuration) tab is also visible, showing '启用网络连接' (Enable network connection) checked, '用户名' (Username) as 40, '密码' (Password) as \*\*\*\*\*, and '轮询时间' (Polling time) as 2 seconds. The '本机IP' (Local IP) is 192.168.1.105. The '确定' (OK) and '取消' (Cancel) buttons are at the bottom.

串口工作方式

The screenshot shows the '配置' (Configuration) window with the '网络配置' (Network Configuration) tab selected. The '启用网络连接' (Enable network connection) checkbox is checked. Under '地址' (Address), '地址确认方式' (Address confirmation method) is set to '自定义ID' (Custom ID). The '数据有效期' (Data validity period) is set to 20 seconds. The '串口配置' (Serial Port Configuration) tab is also visible, showing '启用串口连接' (Enable serial port connection) unchecked, '上传到飞云服务器' (Upload to FIT cloud server) checked, and '从飞云服务器获取控制' (Get control from FIT cloud server) checked. The '本机IP' (Local IP) is 192.168.1.105. The '确定' (OK) and '取消' (Cancel) buttons are at the bottom.

网络工作方式

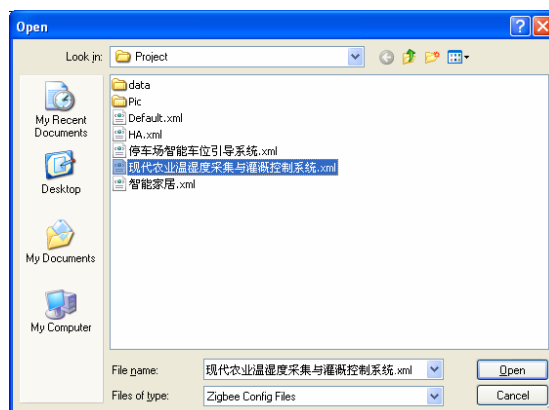
## 4. 基本操作

### 4.1 启动软件：

点击开始菜单中，Feibit IoT Explorer 项，启动应用软件



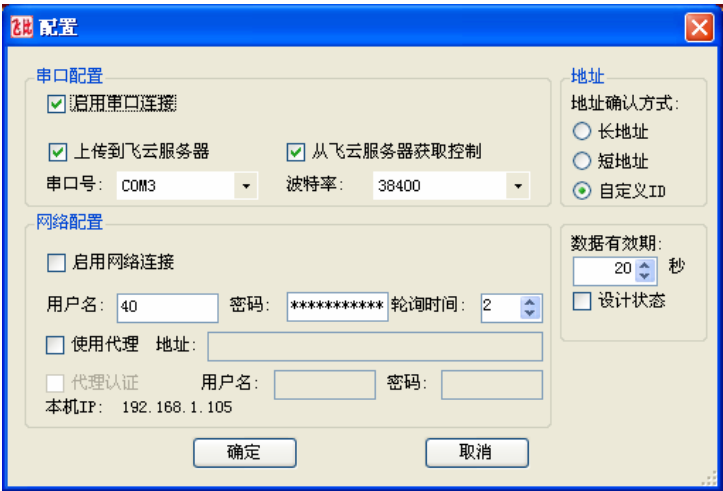
### 4.2 打开配置文件



打开配置文件

### 4.3 连接方式配置





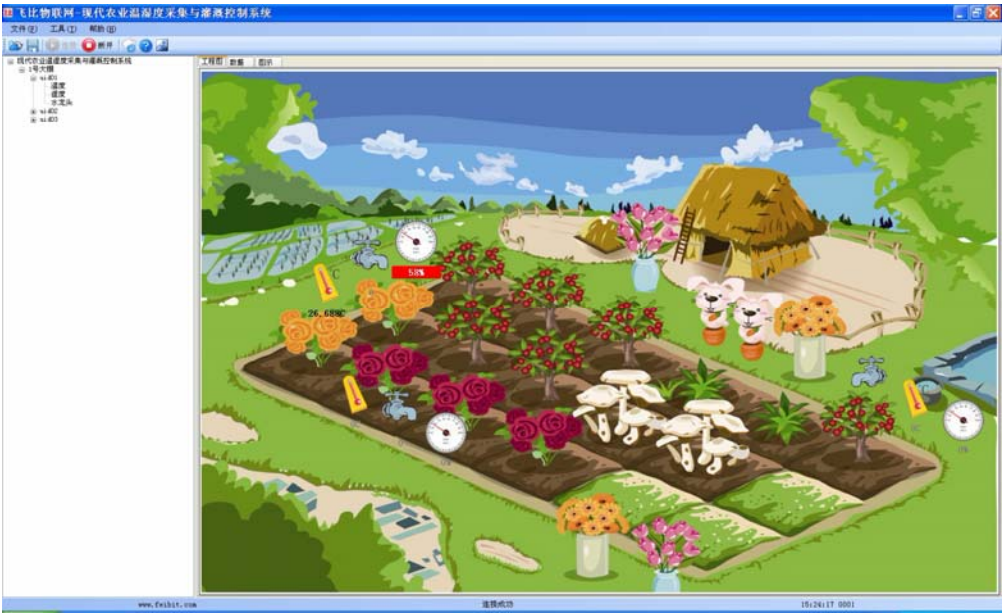
对连接方式进行配置。注：仅需配置一次即可。

#### 4.4 连接

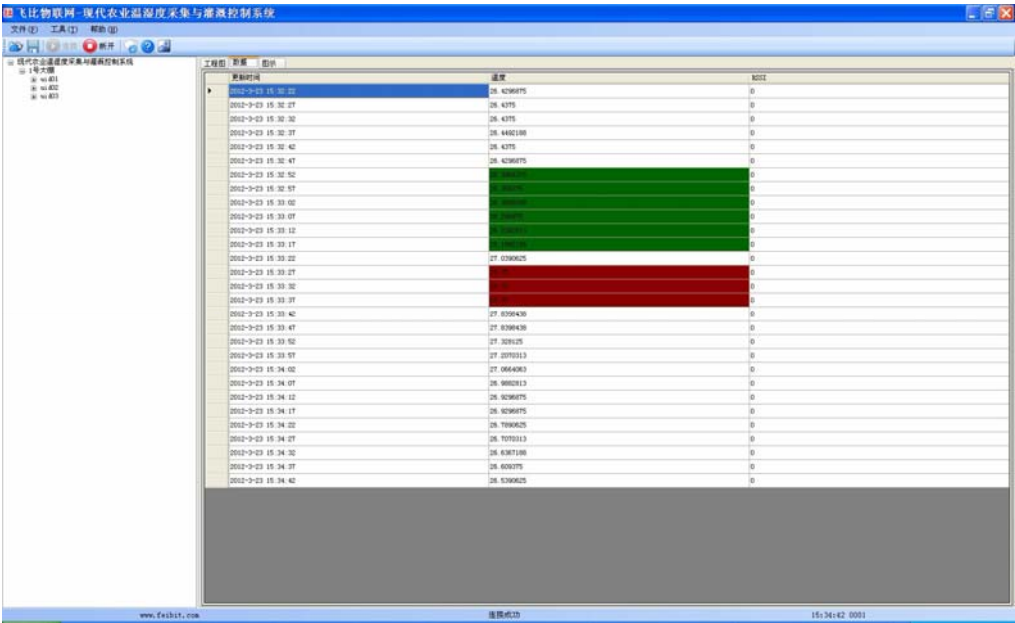


#### 4.5 数据分析

工程图:



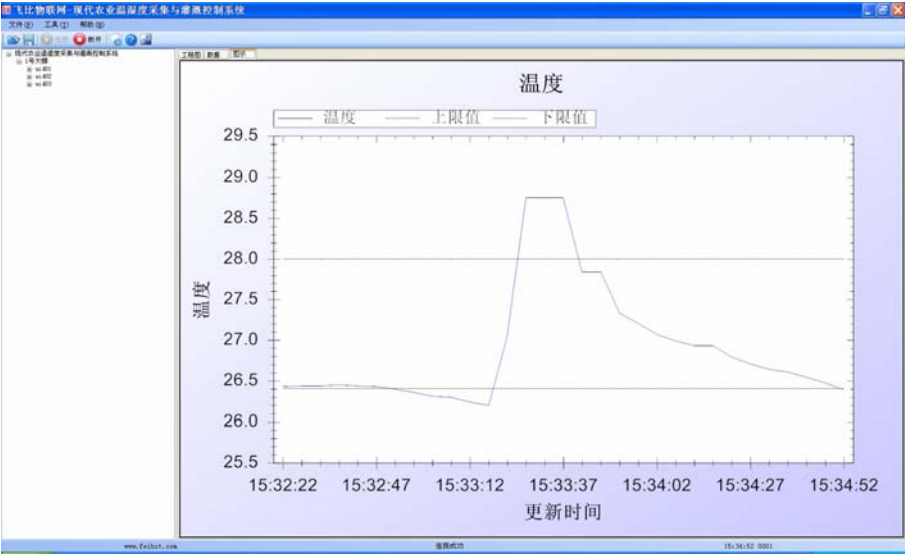
双击节点，即可查看该节点的数据：



该图中，超出设定上限值的被显示为红色，低于设定下限值的被显示为绿色。

更新时间	温度	RSSI
2012-3-23 15:32:22	26.4296875	0

双击“温度”栏，即可转到“图示”页：



## 5. 场景设计基础

### 5.1 进入高级设定状态：



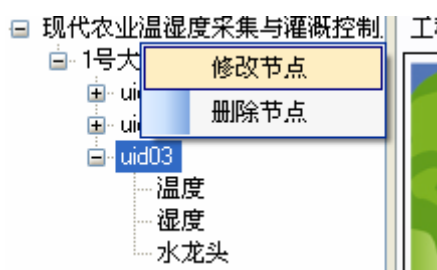
点击设计状态，进入高级设定。

### 5.2 修改节点位置及大小：



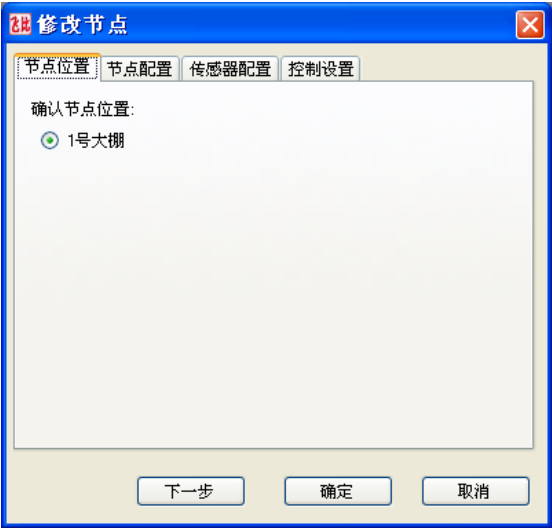
点击节点，拖动即可更改节点的位置和大小。

### 5.3 修改节点：

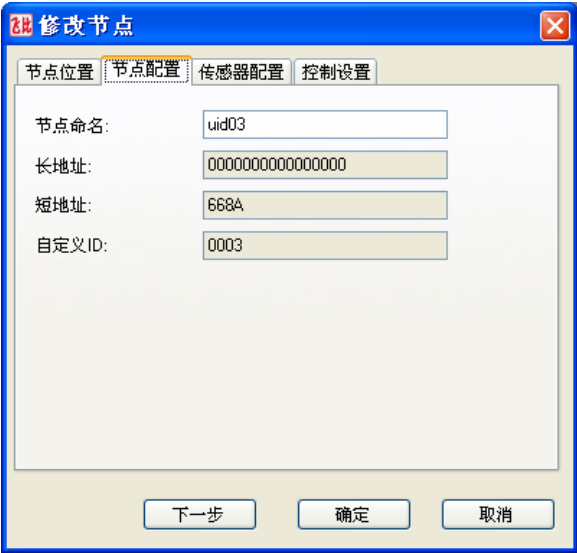


右键点击节点，可以在弹出的菜单中选择修改或删除节点。

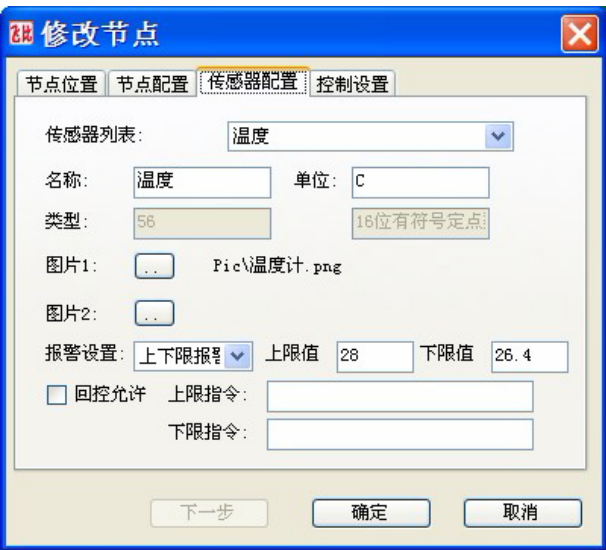
选择“修改节点”：



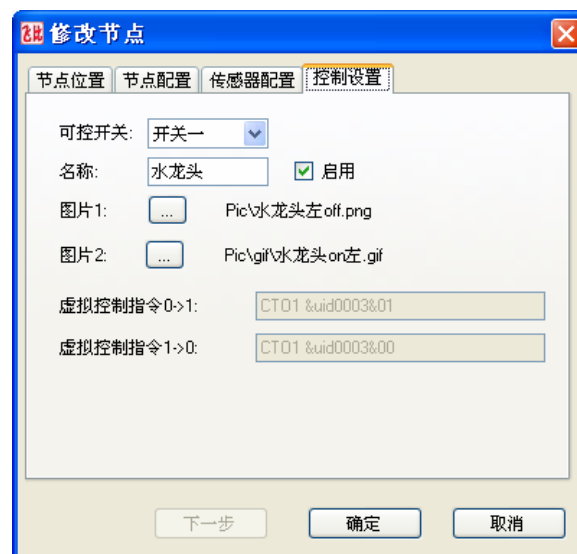
节点位置



节点名称



传感器配置



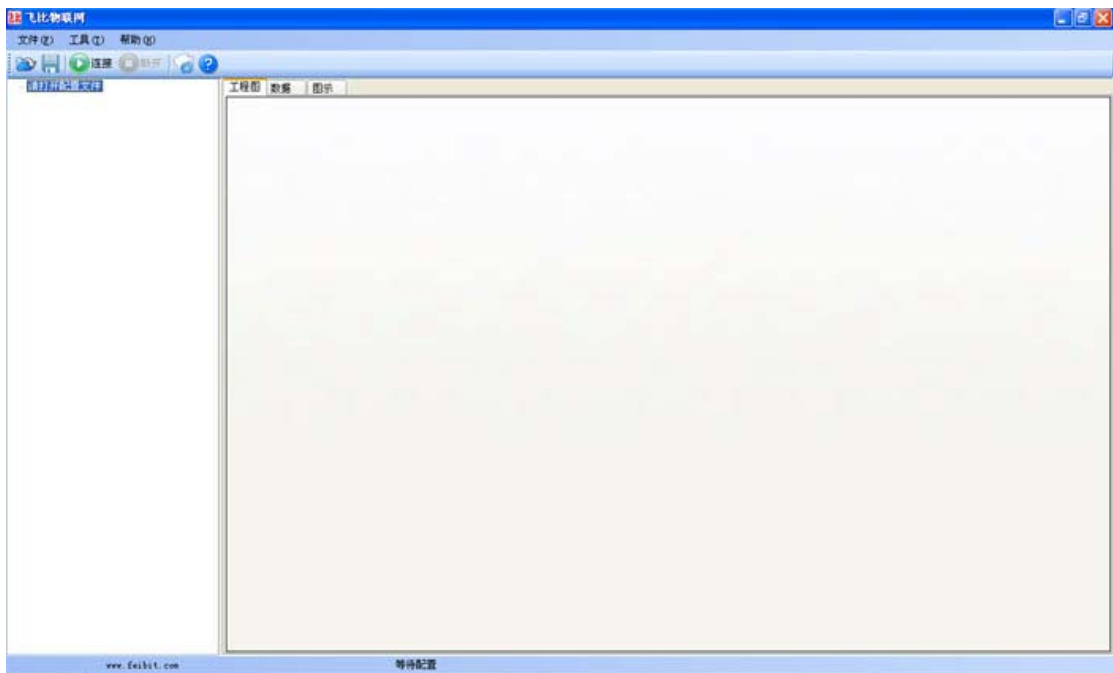
控制设置

## 6、创建应用场景实例

下面我们以创建一个“远程控制 Zigbee 电灯”的应用项目为例，介绍下如何使用用 FIT Explorer 来构建应用场景。

### 6.1 设置界面背景

按上述步骤安装好软件，并打开后。一个简单得像张白纸一样的界面出现在你面前：



没错，它就是一张白纸，却可以画出你想得到的所有东西，而且它可以“动”，动的不但是界面的图片，关键是那些动画下面所连接的是真实的世界，它能告诉你今天风力几级，海水温度多少；你也能通过它来打开家里的空调，为你的宠物喂水... ..

这就是 IoT Explorer 和我们常用的 Internet Explorer 的区别，我们已经把它实现了，并且免费供大家使用，接下来我们能做的，只是做一些 Demo，然后告诉您怎么用。但您的想像力才是它的灵魂，有好的作品，一定记得和我们分享！

创建工作场景，就像一个画家要画一幅画，第一步是要构思，就是先想象一下，理想中的场景是怎样的？作为一个最简单的例子，我们想象有个屋子，客厅里放着一盏电灯，电灯旁边一个开关，当点击开关时，zigbee 节点所控制的电灯动作，并且将结果反馈回控制界面，通过亮、灭两张不同的图片来表示。

按照这个构思，我们先选好素材：

## 1、背景（智能家居.jpg）



## 2、电灯（开灯.png/关灯.png）



## 3、开关按钮（按钮开.png/按钮关.png）



准备工作基本就绪了，但是在 FIT Explorer 中到底是如何将这些元素组合成一个场景的呢？这要借助于 xml 语言，请看下面一段简单的例子：

```
<?xml version="1.0" encoding="GB2312"?>
<Title Name=" 我 家 的 远 程 电 灯 " Background="NULL" ComPort="COM3"
ComBandrate="38400" NetPort="8090" AddressMode="2" DataValidTime="20"
```

```
ConnectType="0" UptoFbCloud="0" GetCTLFbCloud="0" AreaCNT="1" SensorCNT="0">
  <Area0>
    <AreaAttrib AreaName="我的家" Background="pic\智能家居.jpg" NodeCNT="0" />
  </Area0>
</Title>
```

<Title>开始, </Title>结束, <Area0>开始, </Area0>结束, 看过 html 代码的朋友, 可能会觉得有点亲切。至于 xml 语言是什么, 怎么写, 这里不去深究, 有兴趣的朋友可以专门了解下。我们只要知道我们想改的东西在哪里就可以了。


把上面一段代码复制到记事本, 保存成“我家的远程电灯.xml”的文件(注意不是“我家的远程电灯.xml.txt”), 然后在这个 xml 文件所在的目录下建立新的文件夹, 命名为“pic”, 并将前面的所有图片都复制到这个文件夹下。

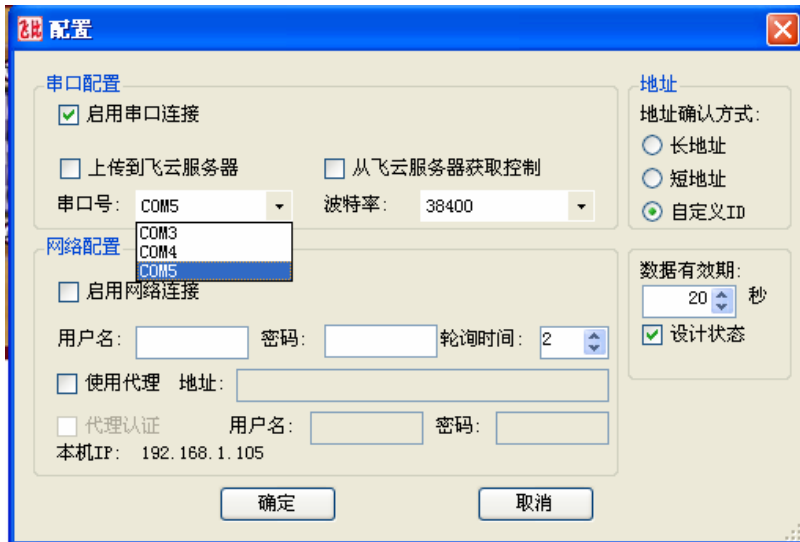
此时打开 FIT Explorer 软件, 点击, 选择刚刚建立的“我家的远程电灯.xml”文件, 此时, 场景的背景即设置成功。




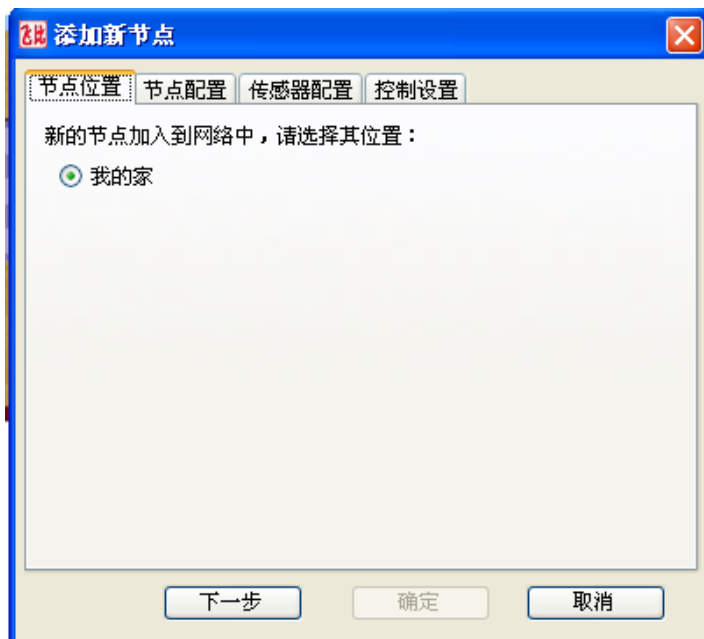
但是电灯、开关在哪里呢? 由于“物联网浏览器”必须与“物”相连, 在继续下面的操作前, 需要将您“符合飞比云传感协议”的设备与电脑间通过串口连接, 获取传感数据。然后, 我们就要把这个数据以可视化的方式, 放到“物联网浏览器”里来。

## 6.2 创建传感器

在保证无线传感网络设备（以 zigbee 设备为例）正确向电脑发送数据后，点击，进入设置界面：

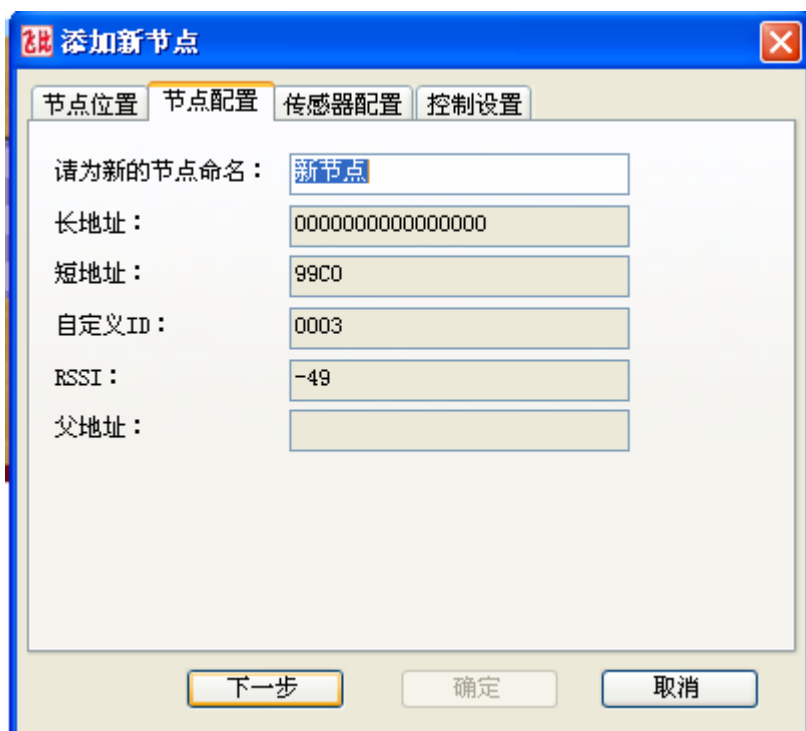


按照上图做好设置后，点“确定”回到主界面，然后点后弹出了如下对话框：



这说明 FIT Explorer 软件已经从 Zigbee 协调器的串口上收到了一组完整的传感数据，在等待您给传感器命名。

点击“下一步”：



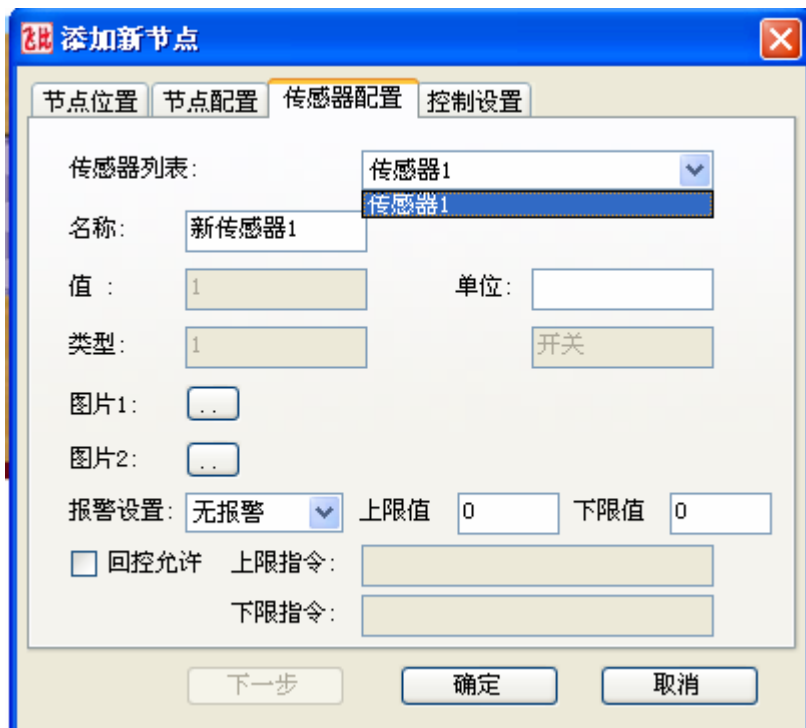
The screenshot shows the 'Add New Node' dialog box with the 'Node Configuration' tab selected. The fields are as follows:

Field	Value
请为新的节点命名:	新节点
长地址:	0000000000000000
短地址:	99C0
自定义ID:	0003
RSSI:	-49
父地址:	

Buttons at the bottom: 下一步 (Next), 确定 (OK), 取消 (Cancel).

我们把这个节点命名为“Zigbee 电灯”，在下面的数据中，我们可以看到更多的信息：长地址，即 IEEE 地址是空的，说明这组数据中没有带长地址；短地址是 99C0；自定义 ID，即 UserID 为 0003（详情还放在后面解释）；RSSI，即信号质量为-49dbm。

继续下一步：

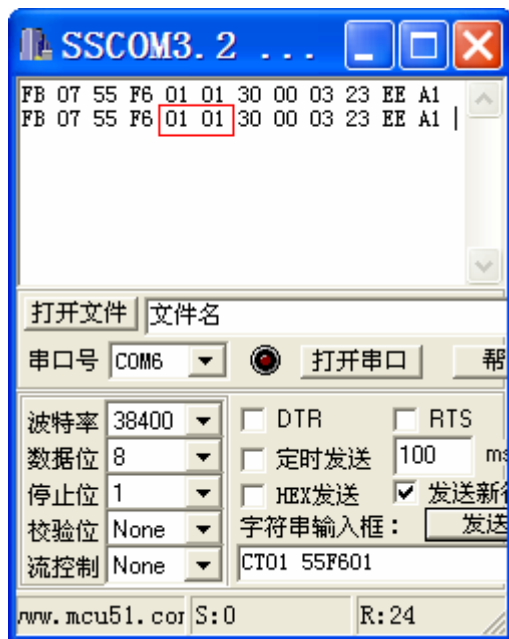


The screenshot shows the 'Add New Node' dialog box with the 'Sensor Configuration' tab selected. The fields are as follows:

Field	Value
传感器列表:	传感器1
名称:	新传感器1
值:	1
单位:	
类型:	1
开关:	开关
图片1:	..
图片2:	..
报警设置:	无报警
上限值:	0
下限值:	0
回控允许:	<input type="checkbox"/>
上限指令:	
下限指令:	

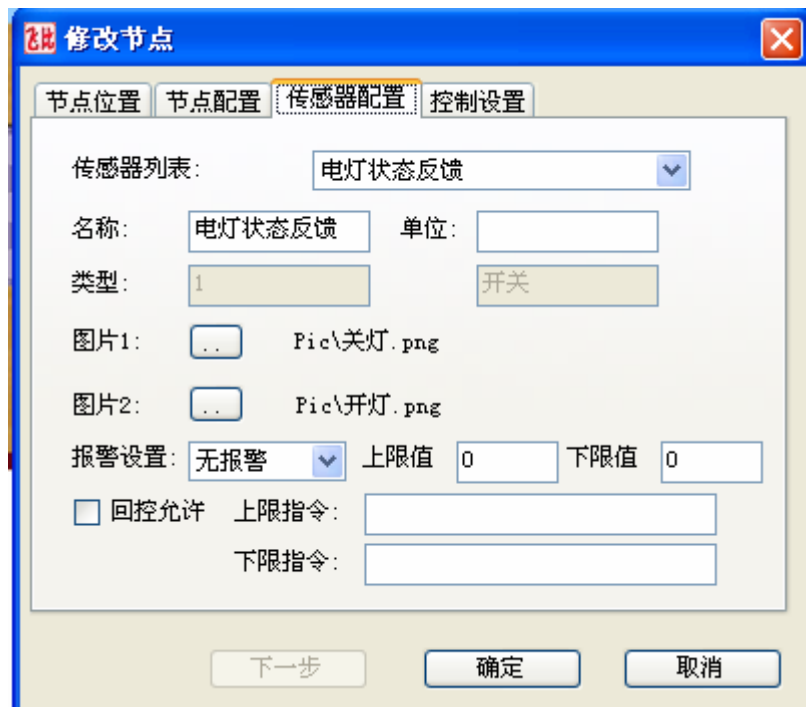
Buttons at the bottom: 下一步 (Next), 确定 (OK), 取消 (Cancel).

从传感器的下拉表中，只有一个选项，说明收到的一组数据中只有一种传感器，按照“飞比云”传感协议，其数据格式如下：（其中 0101 代表传感器返回的高低电平状态）



实际上这一串数据，只有“0101”这一个“传感器数据”，那就是电灯状态的“反馈”信号，其它的数据都是“地址”及“信号质量”等辅助数据。

我们把传感器名称命名为“电灯状态反馈”，然后“图片1”和“图片2”分别指：反馈数据为0和1时，界面中所显示的图片。这里我们指定为素材中的那两个灯泡，按下图进行设置：



点“确定”后，发现主界面上多了一个灯泡：



但灯泡的位置和大小都处于默认状态下，我们看下用 FIT Explorer 是如何实现可视化的场景布置的：

- 1、首先确认设置界面中的 ☒ 设计状态 是否已经勾选的
- 2、通过鼠标点击、选中灯泡后，出现 8 个缩放点，如下图：

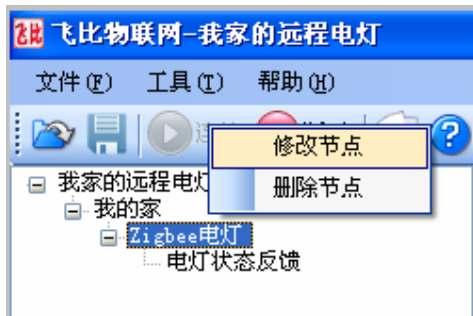


- 3、此时，用鼠标进行大小的缩放、位置的拖动，这一点和普通 windows 软件没什么区别。

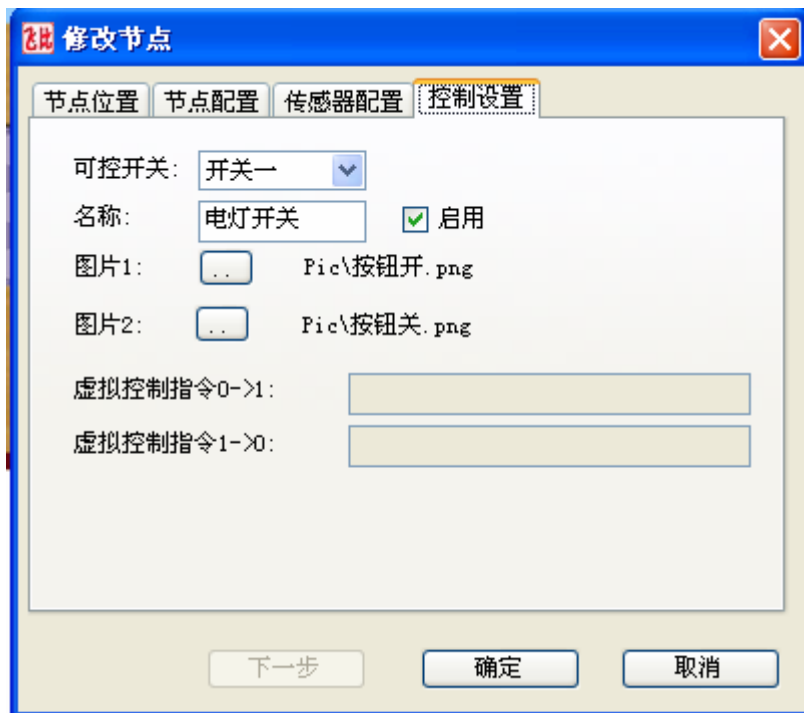
至此，我们完成了灯的状态的可视化采集，为检测我们的成果，您可以将采集 IO 口分别连接高电平或者低电平，看下灯的状态是否发生了变化。

### 6.3 创建控制对象

在左侧对象列表中选择“Zigbee 电灯”，并点击右键，选择“修改节点”，如下图：



然后按如下设置进行配置：



在这里，我们启用了—个可控开关，取名为“电灯开关”，设置了“打开”与“关闭”动作所对应的不同的图片。经过调整后，工程图如下所示：



## 6.4 设置控制指令

首先，我们看下“我家的远程电灯.xml”文件，在我们前面做了一系列的设置后，内容发生了什么变化：

```
<?xml version="1.0" encoding="GB2312"?>
<Title Name=" 我 家 的 远 程 电 灯 " Background="NULL" ComPort="COM5"
ComBandrate="38400" NetPort="8090" AddressMode="2" DataValidTime="20"
ConnectType="0" UptoFbCloud="0" GetCTLFbCloud="0" AreaCNT="1" SensorCNT="2">
  <Area0>
    <AreaAttrib AreaName="我的家" Background="pic\智能家居.jpg" NodeCNT="1" />
    <Zigbee 电灯>
      <NodeAttrib LongAddress="0000000000000000" ShortAddress="99C0"
UserID="0003" NodePicture="" NodeX="0" NodeY="0" NodeType="0" SensorCNT="2" />
      <电灯状态反馈 Type="1" PicName="Pic\关灯.png" PicName2="Pic\开灯.png"
Unit="" SensorX="386" SensorY="420" SensorWidth="76" SensorHeight="211" AlarmType="0"
AlarmUpper="0" AlarmDown="0" AEnable="0" UpAlarm="" DownAlarm="" />
      <电灯开关 Type="4097" PicName="Pic\按钮开.png" PicName2="Pic\按钮关.png"
Unit="" SensorX="484" SensorY="496" SensorWidth="52" SensorHeight="101" AlarmType="0"
AlarmUpper="0" AlarmDown="0" AEnable="0" UpAlarm="" DownAlarm="" />
    </Zigbee 电灯>
  </Area0>
</Title>
```

相较于最初的代码，系统自动增加了一部分内容，增加了传感器与控制对象的相关描述部分。我们重点来看下“电灯开关”这个对象，因为我们下面要对它的控制指令进行设置：

```
<电灯开关 Type="4097" PicName="Pic\按钮开.png" PicName2="Pic\按钮关.png" Unit=""  
SensorX="484" SensorY="496" SensorWidth="52" SensorHeight="101" AlarmType="0"  
AlarmUpper="0" AlarmDown="0" AEnable="0" UpAlarm="" DownAlarm="" />
```

这段代码中的 UpAlarm="" DownAlarm="", 引号中间的内容，分别为点击“打开”与“关闭”两个按钮时，向协调器串口发送的指令字符。

以 FBee™ Zigbee 模块为例，如果我们要控制某节点的 IO 口为高电平，可以向协调器串口发送“CTO1 HHLL01”这样的指令，其中 HHLL 代表的是节点的“短地址”。

但 Zigbee 短地址，有可能在某些条件下发生变化。为了避免这种情况，而且又不用使用过长的 IEEE 地址，我们的默认设置采用了用户自定义 ID，即 UserID 的方式来区分节点。而且 FIT Explorer 软件已经实现了 UserID 与短地址的自动转换，比如 UserID 为 0003 的节点，其短地址为 99C0，如果我们想发送“CTO1 99C001”这样的控制指令，可以转换为“CTO1 &0003&01”。这样，当节点短地址变动时，软件将自动获取新的短地址，并替换&0003&，这样用户就可以不用关注 zigbee 短地址，只关注自己设置的 UserID 号，即可区分不同的设备。

在 xml 语言中，“&”这个符号是用&amp;符号代表的，所以最终的控制代码如下：

```
UpAlarm="CTO1 &amp;uid0003&amp;01" DownAlarm="CTO1 &amp;uid0003&amp;00"
```

将这段代码替换到原 xml 文件中，保存，然后用 FIT Explorer 重新打开，点击界面里的开关按钮，此时与软件连接的 Zigbee 网络中，短地址为 99C0 的节点的第一个可控 IO 口（DIO1）的电平状态将发生相应变化，进而控制“Zigbee 电灯”的亮灭。

## 7、通过 LUA 脚本扩展场景功能

### 7.1 LUA 脚本简介

Lua 是一个小巧的脚本语言。作者是巴西人。该语言的设计目的是为了嵌入应用程序中，从而为应用程序提供灵活的扩展和定制功能。

Lua 脚本可以很容易的被 C/C++ 代码调用，也可以反过来调用 C/C++ 的函数，这使得 Lua 在应用程序中可以被广泛应用。不仅仅作为扩展脚本，也可以作为普通的配置文件，代替 XML、Ini 等文件格式，并且更容易理解和维护。

Lua 由标准 C 编写而成，代码简洁优美，几乎在所有操作系统和平台上都可以编译、运行。一个完整的 Lua 解释器不过 200k，在目前所有脚本引擎中，Lua 的速度是最快的。这一切都决定了 Lua 是作为嵌入式脚本的最佳选择。

在 FIT Explorer 中采用 LUA 脚本的目的是为了客户方便进行功能扩展，比如传感器数据在不同取值范围内，显示不同的图片；传感器原始数据需要进行单位换算再输出到界面中；需要时弹出 Windows 报警窗口；甚至是设置一定的触发条件后，软件自动发送微博、短信等功能，都可以通过 LUA 脚本来进行扩展。

### 7.2 如何启用 LUA

首先，说明一点，在 FIT Explorer 中，每一个传感器都可以对应一个 LUA 脚本文件，对该传感器的界面及行为进行编程。

以一个“风速”的采集为例：界面中用一个动画风车表示风速，风速大时风车转速加快；同时，将采集的风速传感器的 0-3.3V 电压值转换为以 m/s 为单位的的风速值，其转换关系如下表：

--风速(m/s)	风级	AD 值(V)
--0.0-0.2	0 级无风	0.00-0.02
--0.3-1.5	1 级软风	0.03-0.15
--1.6-3.3	2 级轻风	0.16-0.33
--3.4-5.4	3 级微风	0.34-0.54
--5.5-7.9	4 级和风	0.55-0.79
--8.0-10.7	5 级劲风	0.80-1.07
--10.8-13.8	6 级强风	1.08-1.38
--13.9-17.1	7 级疾风	1.39-1.71
--17.2-20.7	8 级大风	1.72-2.07
--20.8-24.4	9 级烈风	2.08-2.44
--24.5-28.4	10 级狂风	2.45-2.84
--28.5-32.6	11 级暴风	2.85-3.26
--32.7-	12 级飓风	3.27-

## 1、为风速传感器指定 LUA 脚本



## 2、在脚本中添加不同值对应的界面 gif 动画文件

```
--GetPicName-----
```

```
--GetPicName (fValue)
```

```
--获得一个图片名，传入参数是当前的传感器值，可能是整数，也可以有小数
```

```
--返回:带路径的文件名 字符串
```

```
function GetPicName ( fValue )
```

```
if(fValue<0.03) then
```

```
    x="Pic\\风速\\00.gif"
```

```
elseif(fValue<0.34) then
```

```
    x="Pic\\风速\\01.gif"
```

```
elseif(fValue<0.80) then
```

```
    x="Pic\\风速\\02.gif"
```

```
elseif(fValue<1.39) then
```

```
    x="Pic\\风速\\03.gif"
```

```
elseif(fValue<2.08) then
```

```
    x="Pic\\风速\\04.gif"
```

```
elseif(fValue<2.85) then
```

```
    x="Pic\\风速\\05.gif"
```

```
else
```

```
    x="Pic\\风速\\06.gif"
```

```
end
```

```
return x
```

```
end
```

### 3、每次传感数据发生变化时，进行原始数据与显示数据的换算：

--ValueChange-----

--ValueChange(fValue,Dw)

--当传感器值改变时调用，传入：当前传感器值，传感器单位。

--返回：字符串，显示在 FIT 传感器图片下方

```
function ValueChange ( fValue,Dw )
wSpeed = fValue*10
if(fValue<0.03) then
    wGradeStr = "0 级无风: "
elseif(fValue<0.16) then
    wGradeStr = "1 级软风: "
elseif(fValue<0.34) then
    wGradeStr = "2 级轻风: "
elseif(fValue<0.55) then
    wGradeStr = "3 级微风: "
elseif(fValue<0.80) then
    wGradeStr = "4 级和风: "
elseif(fValue<1.08) then
    wGradeStr = "5 级劲风: "
elseif(fValue<1.39) then
    wGradeStr = "6 级强风: "
elseif(fValue<1.72) then
    wGradeStr = "7 级疾风: "
elseif(fValue<2.08) then
    wGradeStr = "8 级大风: "
elseif(fValue<2.45) then
    wGradeStr = "9 级烈风: "
elseif(fValue<2.85) then
    wGradeStr = "10 级狂风: "
elseif(fValue<3.27) then
    wGradeStr = "11 级暴风: "
else
    wGradeStr = "12 级飓风: "
end
x=string.format("%s%3.2f%s",wGradeStr,wSpeed,Dw)
return x
end
```

### 4、除了上述几个函数外，当长时间没收到传感器数据，系统就会自动调用如下函数：

function DataExpired ( fValue,Dw )

用户可在此函数内编写 LUA 脚本，实现传感器“断线”时，执行相应的“警示”或其它相关功能。

## 8、飞比云服务平台介绍及其 API 详解

飞比物联网浏览器,在完成上述基本的传感器数据的本地采集与控制对象的本地控制之外,还可以连接“飞比云服务平台”,将传感与控制扩展到互联网上,即实现“远程”的传感器采集与控制,而要实现这一功能,仅需要在飞比物联网浏览器上进行简单的设置即可。下文将介绍其背景、功能及 API 接口。

### 8.1 背景

飞比云服务平台最初只是想解决我们遇到的一个问题:我们在给客户演示我们的 GPRS 网关的时候,经常在客户那里没办法临时搭建一个简易服务器,因为经常用的是内网的电脑,而且没有权限设置局域网路由器;就算有这样的条件,但是网络状况不好的时候,服务器经常掉线,搞得很狼狈。于是,我们就想到了自己搭建一个专业的服务器,保证随时随地都可以稳定工作。这就是“飞比云服务平台”的雏形。

后来,软、硬件不断扩展与完善,现在它除了能完成为 GPRS 网关提供数据服务外,还可以接 TCP/IP/WIFI 等多种网关,将其数据进行上传与下载,完成远程监控的功能;甚至可以不加网关,直接通过“飞比物联网浏览器”将数据上传,进行远程监控。

### 8.2 系统结构



## 8.3 功能简介

“飞比云”是一个开放的平台，可以把它理解成一个传感数据的“容器”，您自己放数据进去，自己再取出来。我们不限您放的是什么数据，只限制每条数据不超过 160 个字符（字符为 0-9，A-F 16 种字符，所以可表示 80 字节的 16 进制数据）

而且您完全可以按自己的想法定义这 80 字节数据，并自行进行解析。所以，它并不限制任何设备的接入。不管这个设备是否是 Zigbee 无线通讯协议，也不管它的通讯频率是多少，只要可以通过任何方式连接互联网，即可与飞比云服务平台对接。

当然，我们推荐设备按“飞比云传感协议”来进行上传，这样就可以用“飞比物联网浏览器”对您的传感器进行图形化浏览。节省开发时间。

为方便用户实现远程的“采集”与“控制”两种功能，飞比云用户数据库，分成两个子库，分别用来存放传感器的“采集”数据（即 upData），与用户对执行装置发出的“控制”数据（即 downData）。采用两套相似的 API 进行管理。

## 8.4 帐号申请及简单测试

1、首先在飞比论坛（bbs.feibit.com）注册一个帐号，登陆后进入 [www.feibit.com/yun](http://www.feibit.com/yun) 这个界面，如下图：



2、点击“申请飞云帐号”，出现如下信息：

- 请记住如下信息，并对其保密：
- 您可使用的“飞云”数据库大小为：256\*80字节
- 您的用户ID号(userID)是：24
- 您的API密钥(API Key)是：~~1-T-S41TQ-R99~~
-

3、保存您的“userID”和“API Key”，在后面将会用到。

4、在没有任何设备的前提下，即可对数据的上传和下载进行简单测试：

进行上传实验：直接访问这样的地址：

<http://www.feibit.com/put.php?userID=xxxx&userKey=xxxxxxxxxxxxxxxx&upData=FB1234567890AA>

如果上传成功，则返回“xxx#upData send successful!”

进行下载实验：直接访问这样的地址：

<http://www.feibit.com/get.php?userID=xxxx&userKey=xxxxxxxxxxxxxxxx&upDataStartNo=1>,

如果下载成功，则依次返回“1#upData=... ..”，“2#upData=... ..”，直至最后一条数据记录。

其中，userID 与 userKey 的取值见上述 2。

5、“反向控制指令”的上传和下载测试：

进行上传实验：直接访问这样的地址：

<http://www.feibit.com/ctput.php?userID=xxxx&userKey=xxxxxxxxxxxxxxxx&downData=CTO101>

如果上传成功，则返回“xxx#downData send successful!”

进行下载实验：直接访问这样的地址：

<http://www.feibit.com/ctget.php?userID=xxxx&userKey=xxxxxxxxxxxxxxxx&downDataStartNo=1>,

如果下载成功，则依次返回“1#downData=... ..”，“2#downData=... ..”，直至最后一条数据记录。

其中，userID 与 userKey 的取值见上述 2。

## 8.5 API 详解

飞比云用户数据库分为采集数据（upData）库与控制数据（downData）两个子库，分别采用两套结构相似的 API 接口。

### 1、upData 数据上传 API

`http://www.feibit.com/put.php?userID=xxx&userKey=xxx&upData=FBXXXXXXXXXXXXAF`

此指令将把数据“FBXXXXXXXXXXXXAF”上传至 userID 对应的数据库中，其序号 upDataNo 将在上一条基础上自动加 1，满“最大数据个数（MaxNo）”后，循环至 0001，upDataPtr 将保存最新数据指针。

如果密码错，返回文本

“userID or key error!”

### 2、upData 数据下载 API

1)、`http://www.feibit.com/get.php?userID=xxx&userKey=xxx&totalNo`

此指令返回当前 userID 的全部数据个数，如果总数据个数小于“MaxNo”，返回 upDataPtr；如果大于 MaxNo，返回 MaxNo

如果密码错，返回文本

“userID or key error!”

2)、`http://www.feibit.com/get.php?userID=xxx&userKey=xxx&upDataNo=xxx`

此指令返回，记录号为 upDataNo 的数据，格式为：

`xxx#upData=FBXXXXXXXXXXXXXXXXXXXX`

如果密码错，返回文本

“userID or key error!”

3)、`http://www.feibit.com/get.php?userID=xxx&userkey=xxx&upDataStartNo=xxx`

返回从 upDataStartNo 号开始，至 totalNo 号结束的所有数据，格式为：

`1#upData=FBXXXXXXXXXXXXXXXXXXXX`

`2#upData=FBXXXXXXXXXXXXXXXXXXXX`

...

`x#upData=FBXXXXXXXXXXXXXXXXXXXX`

直到最后一条记录

如果密码错，返回文本

“userID or key error!”

4)、`http://www.feibit.com/get.php?userID=xxx&userKey=xxx&upDataPtr`

此指令返回当前 userID 的最新数据编号

upDataPtr=xxxx

如果密码错，返回文本

“userID or key error!”

### 3、downData 数据上传 API

http://www.feibit.com/ctput.php?userID=xxx&userKey=xxx&downData=FBXXXXXXXXXX  
XXAF

此指令将把数据“FBXXXXXXXXXXAF”上传至 userID 对应的数据库中，其序号  
downDataNo 将在上一条基础上自动加 1，满“最大数据个数(MaxNo)”后，循环至  
0001，downDataPtr 将保存最新数据指针。

如果密码错，返回文本

“userID or key error!”

### 4、downData 数据下载 API

1)、http://www.feibit.com/ctget.php?userID=xxx&userKey=xxx&totalNo

此指令返回当前 userID 的全部数据个数，如果总数据个数小于“MaxNo”，返回  
downDataPtr；如果大于 MaxNo，返回 MaxNo

如果密码错，返回文本

“userID or key error!”

2)、http://www.feibit.com/ctget.php?userID=xxx&userKey=xxx&downDataNo=xxx

此指令返回，记录号为 upDataNo 的数据，格式为：

xxx#downData=FBXXXXXXXXXXXXXXXXXXXXX

如果密码错，返回文本

“userID or key error!”

3)、http://www.feibit.com/ctget.php?userID=xxx&userkey=xxx&downDataStartNo=xxx

返回从 downDataStartNo 号开始，至 totalNo 号结束的所有数据，格式为：

1#downData=FBXXXXXXXXXXXXXXXXXXXXX

2#downData=FBXXXXXXXXXXXXXXXXXXXXX

...

x#downData=FBXXXXXXXXXXXXXXXXXXXXX

直到最后一条记录

如果密码错，返回文本

“userID or key error!”

4)、http://www.feibit.com/ctget.php?userID=xxx&userKey=xxx&downDataPtr

此指令返回当前 userID 的最新数据编号

downDataPtr=xxxx

如果密码错，返回文本

“userID or key error!”

# 9、飞比云传感协议介绍

本协议规定了无线传感网中的汇总节点（如 zigbee 中的协调器），以何种格式发送至上位机（如 PC 串口），进而由上位机软件进行解析。

此协议的最大特点是：用户可灵活自定义传感数据，扩展性极好，不会产生冗余数据，适合在资源有限的无线传感网设备中使用。

“飞比物联网浏览器”与“飞比云服务平台”采用的即是这种数据格式，用户使用该协议可以轻松实现与“飞比云服务平台”的对接，无须做任何上位机的开发工作，即可实现图形化监控，甚至在不增加“网关”设备的前提下，轻松通过“飞比物联网浏览器（本地）”将数据上传至“飞云服务器”，并在地球的另一端用“飞比物联网浏览器（网络）”进行监控！

## 9.1 数据格式

	帧头	长度	节点短地址	【类型+数据】	【类型+数据】	【...】	CRC
字节数	1	1	2	1+N	1+N	1+N	

注：【】内的数据为可选数据，其余为必备数据

## 9.2 格式详解

### 1、帧头

数据开始标识，固定为 FB

### 2、长度

“可选数据”总长度，即【】内的数据长度

### 3、节点短地址

此数据包的发起节点的 16 位 Zigbee 短地址

### 4、类型

其中高四位，共有 0-15，16 个数字，代表意义如下表：

高四位值	数据类型	数据长度 N
0	开关量	1
1	8 位无符号模拟量	1
2	8 位有符号模拟量	1
3	16 位无符号模拟量	2
4	16 位有符号模拟量	2
5	16 位有符号带定点量	2
6	32 位有符号模拟量	4
7	32 位浮点数	4
8	64 位 IEEE 地址	8
9-15	保留位	... ..

低四位值的意义与数据类型有关：

1)、当数据为“开关量（0/1）”，即高四位为 0 时，代表后续数据“共有几个开关量”，所以一个数据包中最多可以支持  $2^4=16$  个开关量；

2)、当数据为“非开关量”，即高四位大于 0 时，代表后续模拟数据的“序号”，同上，一个数据包中最多可以支持  $2^4=16$  个非开关量

其中，0-5 为系统固定的数据内容，如下：

低四位“序号”	名称
0	用户自定义 ID 号（UserID）
1	节点 IEEE 地址
2	节点父地址
3	信号质量 RSSI
4-5	保留位

6-15 为用户自定义部分，用户可在上、下位机自定义、自命名

### 5、数据

即实际的传感器数据内容（如 IEEE 地址、RSSI 值、温湿度数据等信息）

### 6、CRC

数据包检验和（异或和）

### 9.3 示例分析

某传感节点带一个温度和湿度传感器，带一个开关量采集，同时带 IEEE 地址/RSSI 值信息：

FB 13 HH LL 【81 xx xx xx xx xx xx xx xx】【23 xx】【36 xx xx】【17 xx xx】【01 xx】CRC

解析：

**FB:** 帧头

**13:** 【】内数据总数，共 19 字节

**HH LL:** 分别代表节点地址的高 8 位和低 8 位；xx 代表实际数据

**【81 xx xx xx xx xx xx xx xx】:**

8: 64 位 IEEE 地址

1: 模拟量序号为 1(系统定义)

xx xx xx xx xx xx xx xx: 传感节点的 64 位 IEEE 地址

**【23 xx】:**

2: 8 位有符号模拟量

3: 模拟量序号为 3(系统定义)

xx: 传感节点发送到协调器最后一跳的 rssi 值，单位为 dbm

**【36 xx xx】:**

3: 16 位无符号模拟量

6: 模拟量序号为 6(用户定义)

xx xx: 温度数据，前 8 位代表整数部分，后 8 位代表小数部分，单位为 °C

**【17 xx xx】:**

1: 8 位无符号模拟量

7: 模拟量序号为 7(用户定义)

xx: 湿度数据，单位为%

**【01 xx】:**

0: 开关量

1: 后八位数据中含有一个开关量

xx: 开关量数据，最低位(bit0)代表此开关量数值

**CRC:** 数据包检验和（异或和）

## 10、FIT Explorer 的“网关”功能介绍

FIT Explorer 已经内置了“飞比云服务平台”的连接功能，即可以将无线传感网汇总节点汇报的传感器数据上传至“飞比云服务平台”；同时从飞比云服务平台中获取用户下达的控制指令，转发给无线传感网汇总节点，进而控制某节点动作。

### 10.1 网关功能设置

以上述第 6 章中“Zigbee 电灯”为例子，继续介绍如何用 FIT Explorer 实现“网关”的功能：

- 一、首先要获取飞比云服务器的用户 ID 号与 API Key，申请地址：[www.feibit.com/yun](http://www.feibit.com/yun)
- 二、在物联网浏览器主界面中点击“选项”进入设置界面：



- 三、将获取的用户 ID 号与 API Key 分别填入“用户名”与“密码”中，并勾选“上传至飞云服务器”和“从飞云服务器获取控制”

仅需要这三步，即可在本机实现“网关”的功能——将 zigbee 传感器数据上传至网络服务器；并将服务器上的控制指令转达给 zigbee 协调器！

- 四、我们先用服务器 API 方式测试数据是否正确送达：

打开如下网址：

<http://www.feibit.com/get.php?userID=40&userKey=xxxxxxxxxxxxxxxx&upDataStartNo=1>

注：其中 userID=40&userKey=xxxxxxxxxxxxxxxx 替换为您个人的帐号和 api key

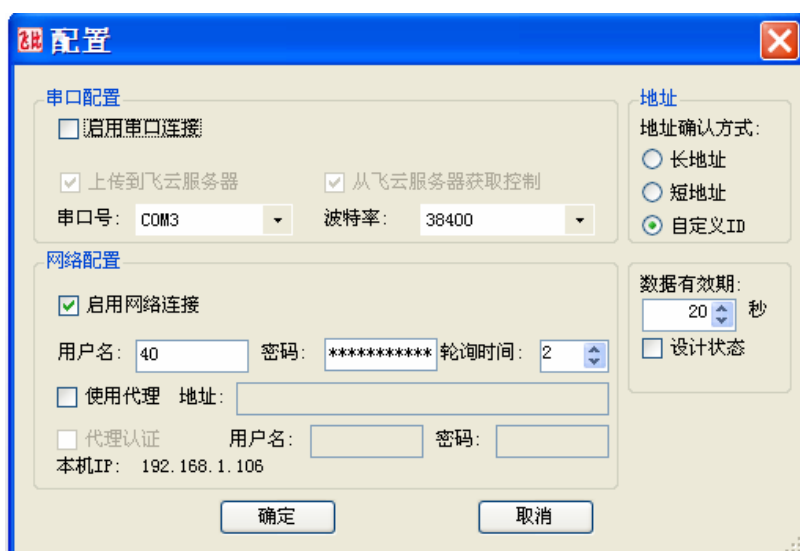
如果网络正常的情况下，将获取如下数据，此数据即为“飞比物联网浏览器”发出，保存至“飞比云服务器”上的传感数据：


← → C www.feibit.com/get.php?userID=40&userKey=XXXXXXXXXXXX&upDataStartNo=1

1#upData=FB0756A8010130000323BAA8 2#upData=FB0756A8010130000323BAA8 3#upDat  
5#upData=FB0756A8010130000323BAA8 6#upData=FB0756A8010130000323BAA8 7#upDat  
9#upData=FB0756A8010130000323B9AB 10#upData=FB0756A8010130000323B9AB 11#upD  
13#upData=FB0756A8010130000323B9AB 14#upData=FB0756A8010130000323B9AB

## 10.2 远程监控功能

同样，如果想通过“飞比物联网浏览器”进行远程互联网监控，设置也非常简单：



仅需要点击“启用网络连接”，输入用户名和密码，此时进入监控界面后，虽然表面看和本地监控没什么不同，但实际上点击后，其监控数据来源于“飞比云服务器”，而并非前面讲的本地串口！