

# 智慧烟感案例开发实验手册

## （基于 NB-IoT）



扫一扫，关注我们，获取更多资料

（教程若有误，请与小助手联系，微信号：BearPi\_Helper）

Revision 1.6

2020 年 11 月 5 日

## 目录

首页-技术文档.....	1
1. 场景概述 .....	3
2. 实验准备 .....	3
3. 硬件连接 .....	3
4. 整体流程 .....	4
5. 创建项目 .....	5
6. 创建产品 .....	6
7. Profile 定义.....	7
8. 编解码插件开发.....	10
9. 设备开发 .....	23
10. 应用开发 .....	23
10.1. 开发应用.....	23
10.2. 编辑应用.....	24
11. 业务调试.....	27
11.1. 监控数据.....	27
11.2. 手动控制.....	27
11.3. 设置自动报警规则.....	29
技术支持获取.....	34

## 1. 场景概述

近年来，火灾频发，造成很大的人员及财产损失，国家高度重视，陆续出台相关文件，明确相关要求，落实消防安全责任。同时，人民消防意识逐渐提高，独立烟感得到一定程度的普及，在防火减灾方面起到一定的作用。但由于独立烟感产品存在一定的局限，例如人不在场收不到报警信息，工作状态不能实时掌握，独立烟感没有完全解决这些问题。NB-IoT 智慧烟感解决了传统烟感器布线难、电池使用周期短、维护成本高、无法与业主及消防机构交互等缺点。智慧烟感采用无线通信，具备即插即用、无需布线、易于安装等特点。

在该文档中，将基于华为一站式开发工具平台—开发中心，从设备、平台、应用，端到端构建一款智慧烟感解决方案样例，如图 1-1 所示，带您体验十分钟快速上云。

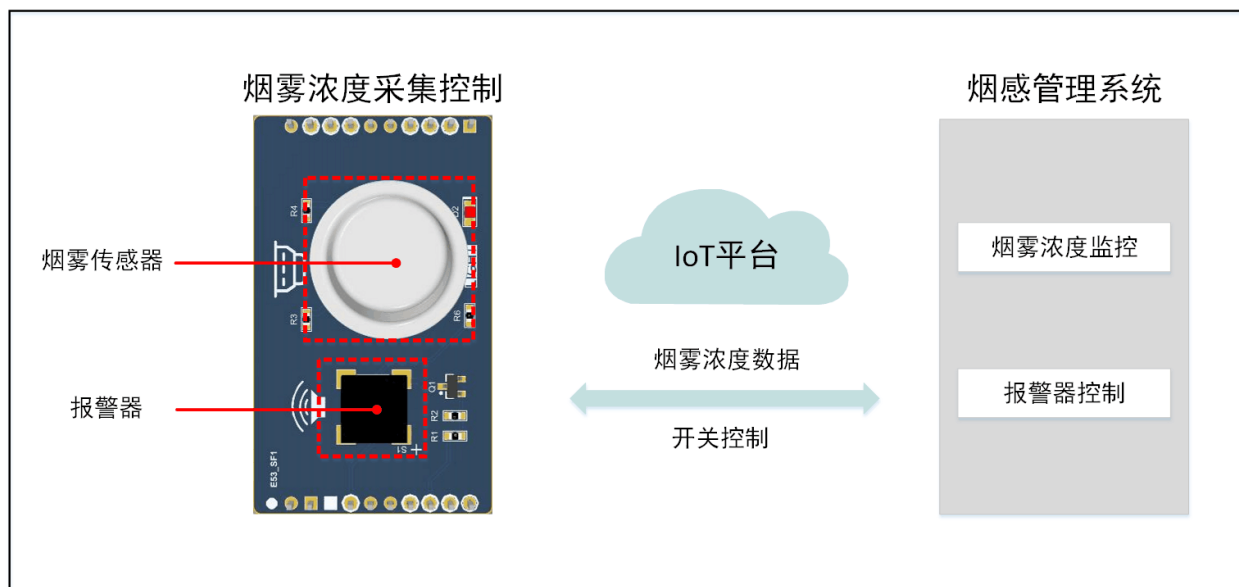


图 1-1 智慧烟感解决方案

## 2. 实验准备

- 开发板：小熊派开发板（含 NB 卡、NB35-A 通信扩展板、E53\_SF1 案例扩展板等）
- IDE：IoT Studio（安装资料包中的版本）
- 平台：华为云账号（需完成实名认证）

## 3. 硬件连接

连接好 E53\_SF1 案例扩展板和 NB35-A 通信扩展板，NB35-A 通信扩展板需要安装 SIM 卡，并注意卡的缺口朝外插入。将串口选择开关拨到 MCU 模式，并用 USB 线将开发板与电脑连接，如图 3-1 所示。

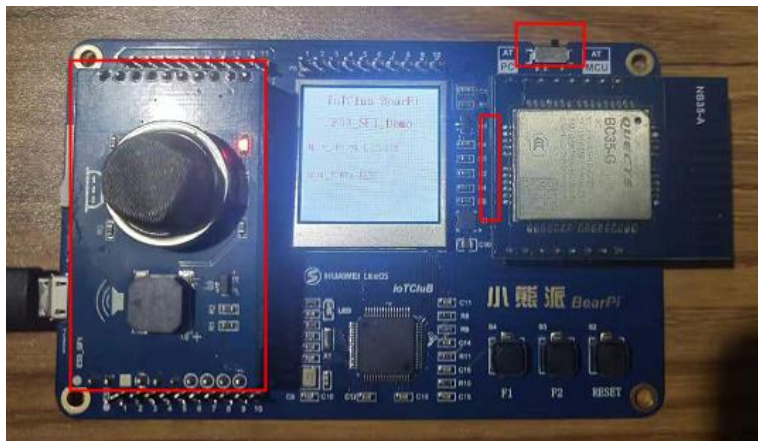


图 3-1 硬件连接

## 4. 整体流程

智慧烟感案例的开发整体流程如图 4-1 所示。

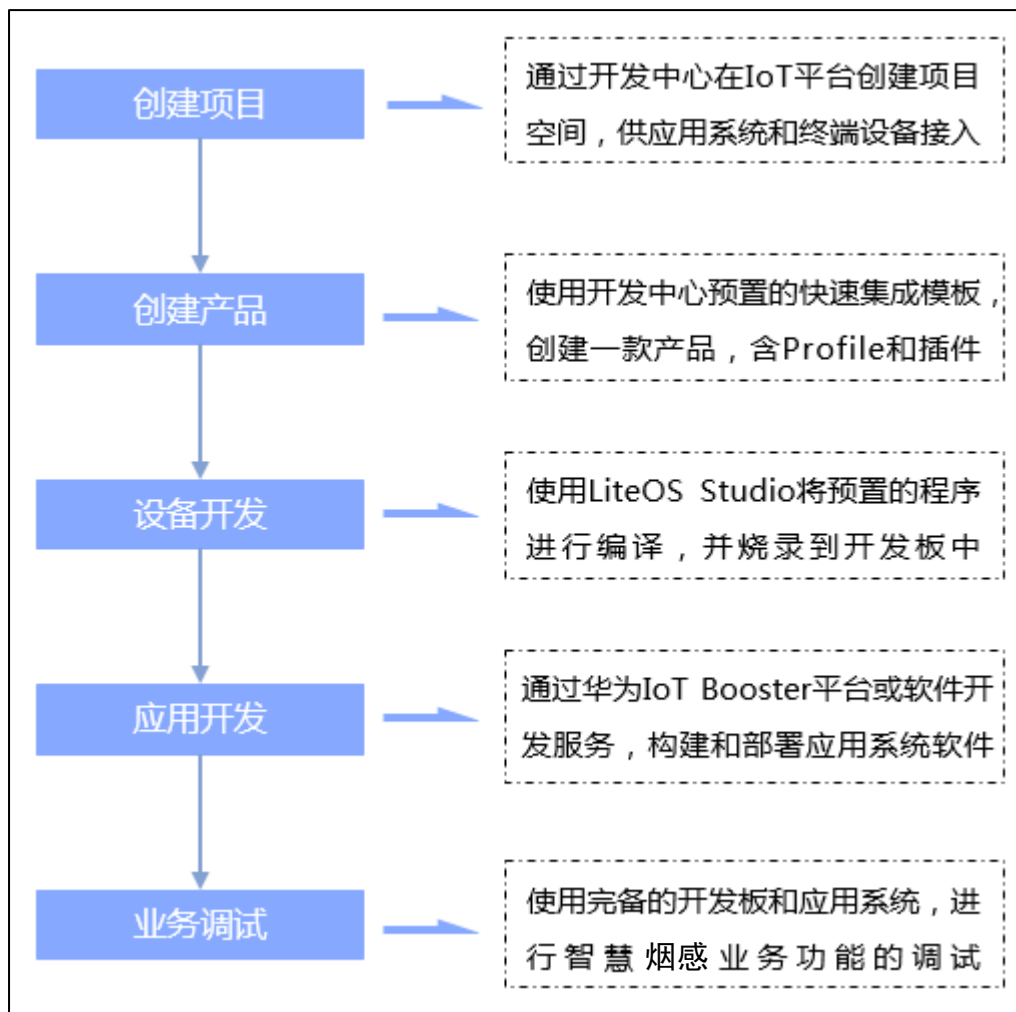


图 4-1 整体流程

## 5. 创建项目

在进行开发之前，开发者需要基于行业属性，创建一个独立的资源空间。在资源空间内，开发者可以开发相应的物联网产品和应用。

**步骤1** 使用华为云账号，登录[物联网应用构建器](#)，如图 5-1 所示。

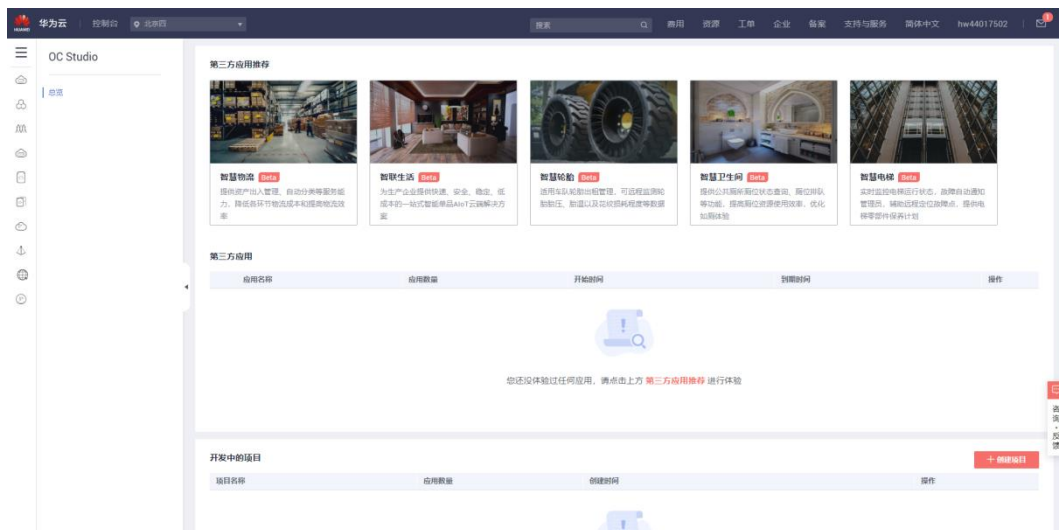


图 5-1 登录物联网应用构建器

**步骤2** 选择右下角的“新建项目”，填写“创建项目”，单击“确定”，如图 5-2 所示。配置示例：

- 项目名称：OceanConnect



图 5-2 新建项目

**步骤3** 创建完成后会生成自己建好的项目之后单击“进入开发”，如图 5-3 所示。



图 5-3 进入开发

**步骤4** 进入“OC Studio”界面下单击“创建应用”填写参数后，单击“确认”  
如图 5-4 所示。配置示例：

- 应用名称：Smoke

图 5-4 创建应用

## 6. 创建产品

某一类具有相同能力或特征的设备的集合称为一款产品。除了设备实体，产品还包含该类设备在物联网能力建设中产生的产品信息、产品模型（Profile）、插件、测试报告 等资源。

**步骤1** 使用华为云账号，登录[设备接入](#)，选择页面左侧的产品，单击右上角下拉框，选择新建产品所属的资源空间（[第 5 节创建的](#)）。如图 6-1 所示。



图 6-1 创建产品

**步骤2** 单击右上角的“创建产品”，创建一个基于 CoAP 协议的产品，填写参数后，单击“立即创建”，如图 6-2 所示。

图 6-2 创建产品

建完后可以发现“产品”里多了“Bearpi\_Smoke”如 6-3 所示，单击“详情”

产品名称	产品ID	设备类型	协议类型	操作
Bearpi_Smoke	5e1bee5ee705930976722c44	Smoke	CoAP	详情
BearPiKit_huawei_model	5e1becd5a705930976722a3b	BearPiKit	CoAP	详情 删除

图 6-3 创建产品

## 7. Profile 定义

在“功能定义”页面下，单击“自定义功能”，配置产品的服务。如图 7-1 所示。



图 7-1 自定义功能

**步骤 1** 进入“新增服务”，填写相关信息后，单击“确认”用来管理路灯的功能。如图 7-2 所示。配置示例：

- 服务名称：Smoke

图 7-2 新增服务

**步骤 2** 在“Smoke”的下拉菜单下点击“添加属性”填写相关信息“Smoke\_Value”，如图 7-3 所示，单击“确认”。





新增属性

\* 属性名称: Smoke\_Value ☒ 必选

\* 数据类型: int(整型)

\* 访问权限: ☒ 可读 ☐ 可写 ☐ 可执行

\* 取值范围: 0 - 65535

步长: 0

单位:

确认 取消

图 7-3 Smoke\_Value

**步骤 3** 在“Smoke”的下拉菜单下点击“添加命令”填写相关信息，如图 7-4 所示：



新增命令

\* 命令名称: Smoke\_Control\_Beep

下发参数: 新增输入参数

参数名称	数据类型	是否必选	操作
无数据			

响应参数: 新增输出参数

参数名称	数据类型	是否必选	操作
无数据			

确认 取消

图 7-4 新增命令 Smoke\_Control\_Beep

**步骤 4** 在“新增命令”里点击“新增输入参数”填写相关信息如图 7-5，再单击“确定”，同样在“新增命令”里点击“新增输出参数”填写相关信息如图 7-6 所示，再单击“确定”，最后在“新增命令”里点“确认”。



The dialog box titled "新增参数" (New Parameter) contains the following fields:

- 参数名称** (Parameter Name): Input field with "Beep". A blue checkmark and "必选" (Required) label are to its right.
- 数据类型** (Data Type): Dropdown menu showing "string(字符串)".
- 长度** (Length): Input field with "3".
- 枚举值** (Enumeration Value): Input field with "ON,OFF". A small icon is at the bottom right of the field.

At the bottom right, there is a "6/1024" character count. At the bottom center are two buttons: "确认" (Confirm) in red and "取消" (Cancel) in white.

图 7-5 新增输入参数



The dialog box titled "新增参数" (New Parameter) contains the following fields:

- 参数名称** (Parameter Name): Input field with "Beep\_State". A blue checkmark and "必选" (Required) label are to its right.
- 数据类型** (Data Type): Dropdown menu showing "int(整型)".
- 取值范围** (Value Range): Two input fields with "0" and "1" separated by a minus sign.
- 步长** (Step Length): Input field with "0".
- 单位** (Unit): Empty input field.

At the bottom center are two buttons: "确认" (Confirm) in red and "取消" (Cancel) in white.

图 7-6 新增输出参数

## 8. 编解码插件开发

**步骤1** 在产品详情插件开发页面，选择“在线开发插件”，单击“在线开发插件”，如图 8-1 所示。



图 8-1 插件开发

**步骤2** 在“在线开发插件”区域，单击“新增消息”，如图 8-2 所示。

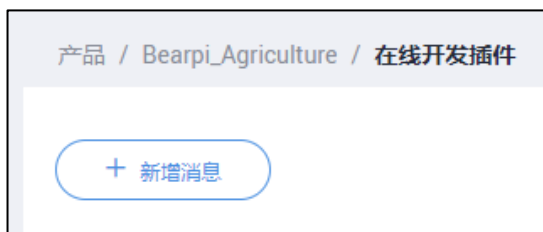


图 8-2 新增消息

**步骤3** 新增消息 Smoke 如图 8-3 所示。

配置示例：

- 消息名：Smoke
- 消息类型：数据上报

新增消息

基本信息

\*消息名

Smoke

\*消息类型

☒ 数据上报
 ☐ 命令下发

☐ 添加响应字段

字段

+ 添加字段

图 8-3 新增消息 Smoke

a) 在“新增消息”界面，单击“添加字段”。

- b) 在“添加字段”界面，勾选“标记为地址域”，然后单击“确认”，添加地址域字段 `messageId`，默认值填 `0x8`，如图 8-4 所示。



**编辑字段** ×

☒ 标记为地址域 ⓘ

\*名字 只有标记为地址域时，名字固定为messageId；其他字段名字不能设置为messageId。

messageId

描述

输入字段描述

数据类型 (大端模式)

int8u (8位无符号整型) ▼

\*长度 ⓘ

1

\*默认值 ⓘ

0x8

偏移值 ⓘ

0-1

完成 取消

图 8-4 添加地址域字段

- c) 在“新增消息”界面，单击“添加字段”，填写相关信息，然后单击“确认”，如图 8-5 所示。

配置示例：

- 名字: `Smoke_Value`
- 数据类型: `int16s`

添加字段

×

☐ 标记为地址域 ?

\*名字
 

Smoke\_Value

描述
 

输入字段描述

数据类型 (大端模式)
 

int16s

\*长度 ?
 

2

默认值 ?

偏移值 ?
 

1-3

确认

取消

图 8-5 添加 Smoke\_Value 字段

d) 在“新增消息”界面，单击“确认”，完成消息 Smoke\_Value 的配置。

**步骤4** 新增消息 **Smoke\_Control\_Beep** 如 8-6 所示。

配置示例：

- 消息名：**Smoke\_Control\_Beep**
- 消息类型：命令下发
- 添加响应字段：是

图 8-6 新增消息 Smoke\_Control\_Beep

a) 在“新增消息”界面，单击“添加字段”，在“添加字段”界面，勾选“标记为地址域”，然后单击“确认”，添加地址域字段 `messageld`，默认值填 `0x9`，如图 8-7 所示。

图 8-7 添加地址域字段

b) 在“新增消息”界面，单击“添加字段”，在“添加字段”界面，勾选“标记响应标识字段”，然后单击“确认”，添加响应标识字段 mid，如图 8-8 所示。



添加字段

☐ 标记为地址域 ⓘ

☒ 标记为响应标识字段 ⓘ

\*名字 只有标记为响应标识字段时，名字固定为mid；其他字段名字不能设置为mid。

mid

描述

输入字段描述

数据类型 (大端模式)

int16u (16位无符号整型)

\*长度 ⓘ

2

默认值 ⓘ

偏移值 ⓘ

1-3

确认 取消

图 8-8 添加响应标识字段 mid

c) 在“新增消息”界面，单击“添加字段”，填写相关信息，然后单击“确认”，如图 8-59 所示。

配置示例：

- 名字：Beep
- 数据类型：string

- 长度：3

添加字段

☐ 标记为地址域 ?
   
☐ 标记为响应标识字段 ?

\*名字

Beep

描述

输入字段描述

数据类型 (大端模式)

string

\*长度 ?

3

默认值 ?

偏移值 ?

3-4

确认

取消

图 8-9 添加 beep 字段

d) 在“新增消息”界面，单击“添加响应字段”，在“添加字段”界面，勾选“标记为地址域”，然后单击“确认”，添加地址域字段 `messageId`，默认值填 `0xa`，如图 8-10 所示。



编辑字段

×

☒ 标记为地址域 ⓘ

☐ 标记为响应标识字段 ⓘ

☐ 标记为命令执行状态字段 ⓘ

\*名字 只有标记为地址域时，名字固定为messageId；其他字段名字不能设置为messageId。

messageId

描述

输入字段描述

数据类型 (大端模式)

int8u (8位无符号整型)

\*长度 ⓘ

1

\*默认值 ⓘ

0xa

偏移值 ⓘ

0-1

完成

取消

图 8-10 添加地址域字段 messageId

e) 在“新增消息”界面，单击“添加响应字段”，在“添加字段”界面，勾选“标记响应标识字段”，然后单击“确认”，添加响应标识字段 mid，如图 8-11 所示。

添加字段

×

☐ 标记为地址域 ②
   
☒ 标记为响应标识字段 ②
   
☐ 标记为命令执行状态字段 ②

\*名字 只有标记为响应标识字段时，名字固定为mid；其他字段名字不能设置为mid。

mid

描述

输入字段描述

数据类型（大端模式）

int16u（16位无符号整型）

\*长度 ②

2

默认值 ②

偏移值 ②

1-3

确认

取消

图 8-11 添加响应标识字段 mid

f) 在“新增消息”界面，单击“添加响应字段”，在“添加字段”界面，勾选“标记命令执行状态字段”，然后单击“确认”，添加命令执行状态字段 `errcode`，如图 8-12 所示。

添加字段

×

☐ 标记为地址域 ①
   
☐ 标记为响应标识字段 ①
   
☒ 标记为命令执行状态字段 ①

\*名字 只有标记为命令执行状态字段时，名字固定为errcode；其他字段名字不能设置为errcode。

errcode

描述

输入字段描述

数据类型（大端模式）

int8u

\*长度 ①

1

默认值 ①

偏移值 ①

3-4

确认

取消

图 8-12 添加命令执行状态字段 errcode

g) 在“新增消息”界面，单击“添加响应字段”，填写相关信息，然后单击“确认”，如图 8-13 所示。

配置示例：

- 名字：Beep\_State

添加字段

×

☐ 标记为地址域 ②
   
☐ 标记为响应标识字段 ②
   
☐ 标记为命令执行状态字段 ②

\*名字

Beep\_State

描述

输入字段描述

数据类型 (大端模式)

int8u

\*长度 ②

1

默认值 ②

偏移值 ②

4-5

确认

取消

图 8-13 添加 Beep\_State 字段

f) 在“新增消息”界面，单击“确认”，完成消息 **Smoke\_Control\_Beep** 的配置。

**步骤5** 拖动右侧“设备模型”区域的属性字段、命令字段和响应字段，与数据上报消息、命令下发消息和命令响应消息的相应字段建立映射关系，如图 8-14 所示。

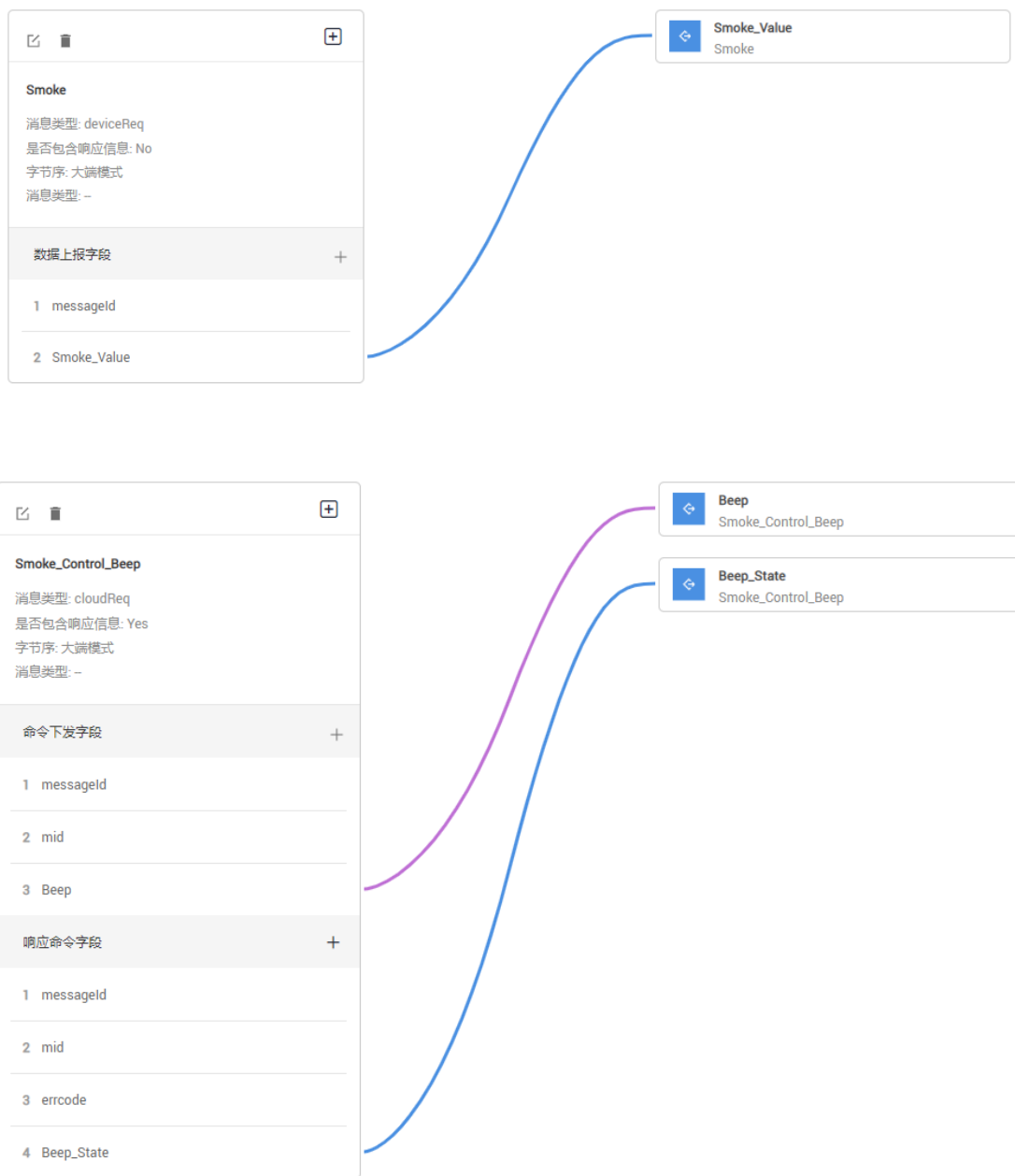


图 8-14 映射 Profile

**步骤6** 单击“保存”，并在插件保存成功后单击“部署”，将编解码插件部署到物联网平台，如图 8-15 所示。



图 8-15 部署

**步骤7** 在“在线调试”下单击“新增调试设备”填写相关信息，列如图 8-16 所示。

配置示例：

- 设备名称：TEST（自定义即可）
- 设备标识码：该设备的 IMEI 号，可在设备上查看，如图 8-17 所示：

图 8-16 新增测试设备

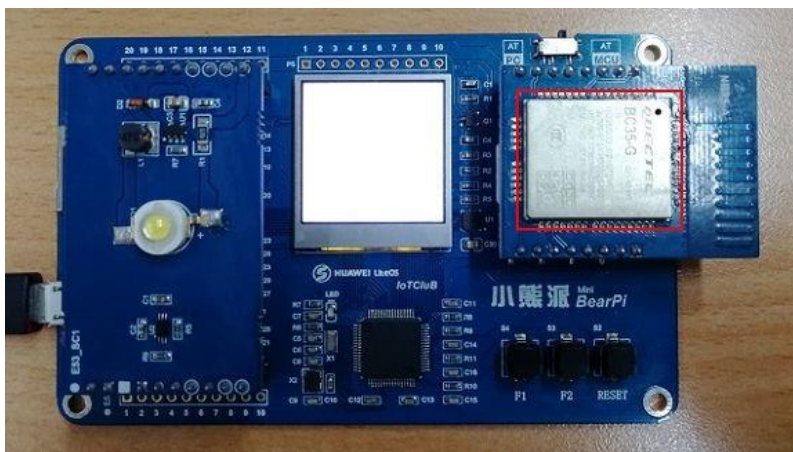


图 8-17 查看 IMEI 号

## 9. 设备开发

请参考《E53\_SF1 智慧烟感案例设备开发手册》

## 10. 应用开发

回到华为“[IoT Studio](#)”里的“web 在线开发”单击之前创建好的应用，进入，如图 10-1所示。

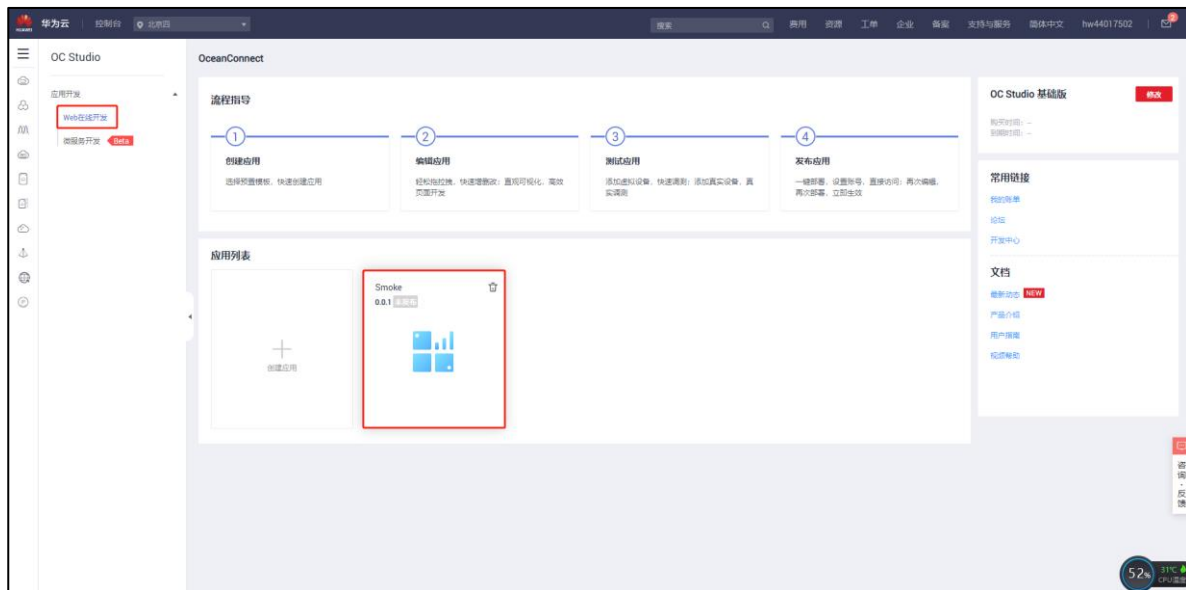


图 10-1 Web 应用开发

### 10.1. 开发应用

**步骤1** 在“开发应用”界面，点击“开发应用”，如图 10-2 所示。



图 10-2 开发应用

## 10.2. 编辑应用

**步骤1** 将鼠标移至“自定义页面 1”上，在弹出的列表中选择“修改”，修改页面信息。  
在弹出的窗口中，修改“菜单名称”为“烟感管理”，其他保持默认，然后点击“确定”，如图 10-3 所示。



图 10-3 修改菜单名称

**步骤2** 选择“烟感管理”页面，设计页面组件布局。

1. 拖动 1 个“选择设备”组件、2 个“监控”组件和 1 个“命令下发”组件至页面中，并按图 10-4 所示的布局进行摆放。

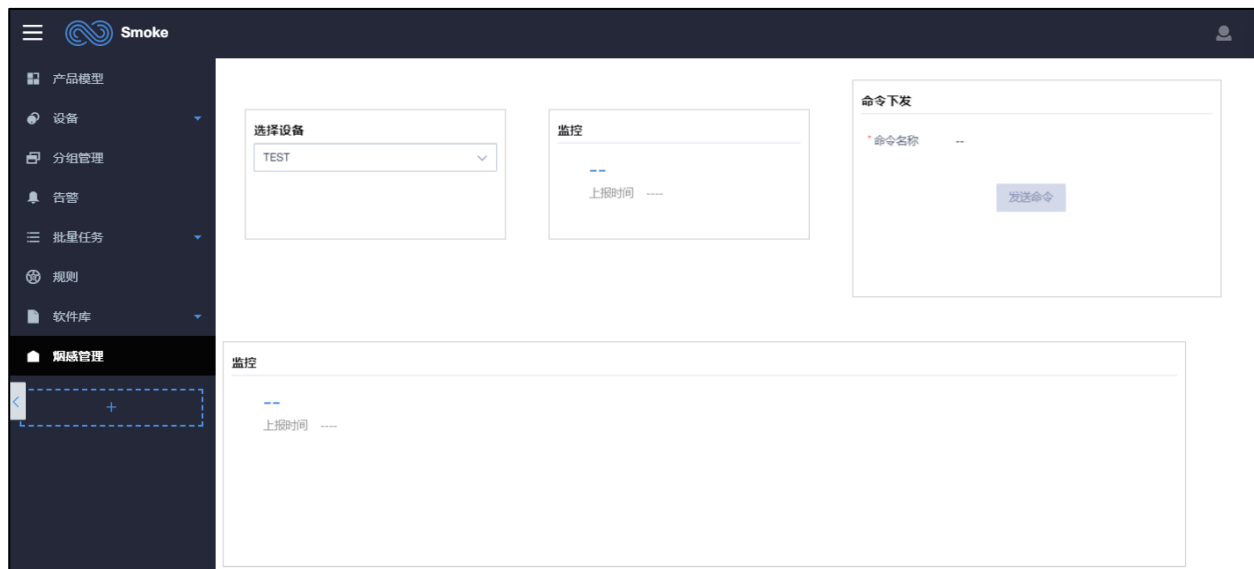


图 10-4 摆放布局



2. 分别单击页面中的“监控”组件，在右侧“配置面板→样式”中设置组件的样式。

标题	显示类型	样式
烟雾浓度监控	简易	保持默认
烟雾浓度变化	图表	保持默认

3. 分别单击页面中的“监控”组件，在右侧“配置面板→数据”中设置组件的数据源。因为 2 个烟雾浓度的“监控”组件都是用于监控环境的烟雾浓度，只是显示方式不同，所以“数据”页面参数设置一样，如图 10-5 所示。

- 产品：选择创建产品中已创建的产品“Bearpi\_Smoke”
- 服务：Smoke
- 属性：Smoke\_Value

图 10-5 配置监控数据源

- 分别单击页面中的“命令下发”组件，在右侧“配置面板”设置对应功能的属性参数，此处以“报警控制”为例，如图 10-6 所示。



图 10-6 配置命令下发参数

**步骤3** 烟雾管理页面构建完成，点击右上角“保存”，然后点击“预览”查看应用页面效果，如图 10-7 所示。



图 10-7 查看页面效果

----结束

## 11. 业务调试

使用已经烧录程序的开发板和构建完成的应用系统，进行智慧烟感业务功能的调试。

### 11.1. 监控数据

开发板上电，在“烟感管理”页面，可以观察烟雾浓度监控和烟雾浓度变化，如图 11-1 所示。

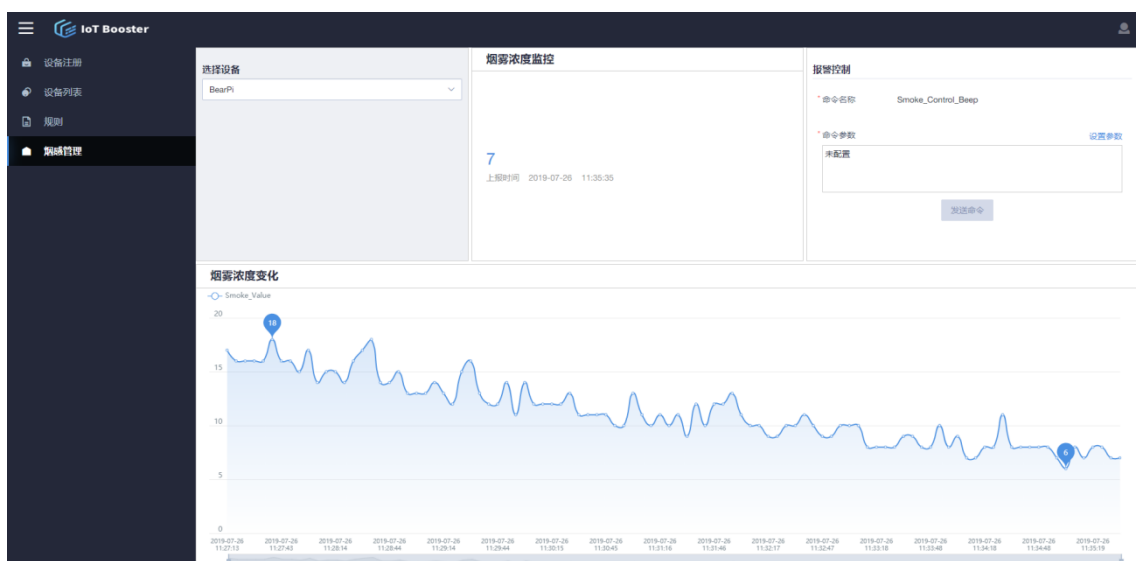
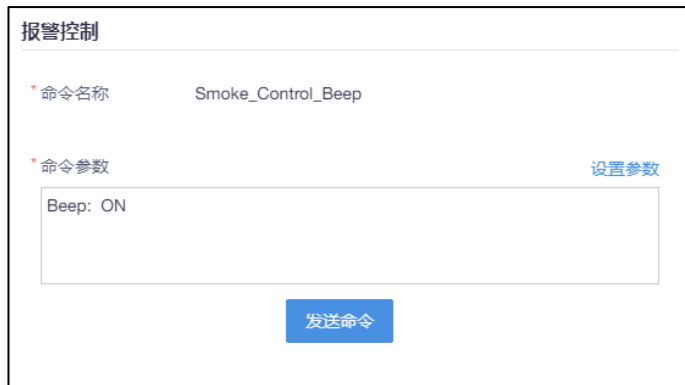


图 11-1 观察烟雾浓度数据

### 11.2. 手动控制

**步骤1** 选择“烟感管理 → 报警控制”，点击“设置参数”，“Beep”选择“ON”，单击“发送命令”，如图 11-2 所示。



The figure shows a modal dialog titled '报警控制' (Alarm Control). It contains the following elements: '命令名称' (Command Name) set to 'Smoke\_Control\_Beep'; '命令参数' (Command Parameters) set to 'Beep: ON'; a '设置参数' (Set Parameters) link; and a blue '发送命令' (Send Command) button at the bottom.

图 11-2 发送开启报警命令

此时扩展板的蜂鸣器会发出报警声，关蜂鸣器命令与开蜂鸣器命令操作步骤一样，差别是“Beep”选择“OFF”。

----结束

### 11.3. 设置自动报警规则

**步骤1** 新建两条规则，分别用于控制照明灯在不同条件下的开和关。选择“规则”，单击“创建规则组”，如图 11-3 所示。



图 11-3 创建规则组

**步骤2** 在“创建规则组”里填写名称（自定义），例如 Beep，如图 11-4 所示：创好后可以发现所有规则里多了一个 Beep 组，之后点亮 Beep 前的框，在点击右上角的“创建规则”选择“设备联动规则”如图 11-5 所示：

A screenshot of the '创建规则组' (Create Rule Group) dialog box. It has a title bar with the text '创建规则组' and a close button 'X'. Inside, there are two input fields: '名称' (Name) with the text 'Beep' and '描述' (Description) with the placeholder text '请输入规则组描述信息。'. At the bottom are two buttons: '确认' (Confirm) and '取消' (Cancel).

图 11-4 创建规则组





图 11-5 创建规则

**步骤3** 在“创建规则”界面，分别填写开、关报警规则信息。

参数	开	关
规则名称	Beep_ON	Beep_OFF
条件		
条件类型	设备类型	
选择设备模型	选择 <a href="#">创建产品</a> 中已创建的产品	
服务类型	Smoke	
属性名字	Smoke_Value	
操作	>	<
值	200	50
动作		
动作类型	设备	
选择设备模型	选择 <a href="#">创建产品</a> 中已创建的产品	
点击选择设备	选择 <a href="#">注册设备</a> 中新增的设备	
服务类型	Smoke	
命令名称	Smoke_Control_Beep	
参数	Beep	
值	ON	OFF
命令状态	启用	
描述	烟雾浓度大于 200 时，开启报警	烟雾浓度小于 50 时，关闭报警

参照上述表格，填写开灯的规则信息。

1. 填写规则名称：Beep\_ON，规则组选择“Beep”如图 11-6 所示。

图 11-6 填写规则名称

2. 设置开灯规则的条件信息。在“条件”模块，点击“设备行为”所在行右侧的“添加”，如图 11-7 所示。

图 11-7 添加条件

3. 按照参数表中开灯规则的条件参数填写条件信息，如图 11-8 所示。

图 11-8 填写条件信息

4. 设置开灯规则的动作。在“动作”模块，点击“设备行动”所在行右侧的“添加”，如图 11-9 所示。

图 11-9 添加动作

5. 按照参数表中开灯规则的动作参数填写动作信息，如图 11-10 所示。

图 11-10 填写动作信息

6. 然后单击右上角的“提交”，开启报警规则创建完成。关闭报警规则的创建操作和开启报警规则的一样，只是规则名称、条件的取值和动作执行不同。

#### 步骤4 测试自动开关报警。

1. 用烟熏烟感传感器(E53\_SF1)，使烟雾浓度>200，查看“烟感管理”中烟雾浓度监控数值如图 11-11 所示和报警器，当烟雾浓度>200 报警器应该自动发出报警声。

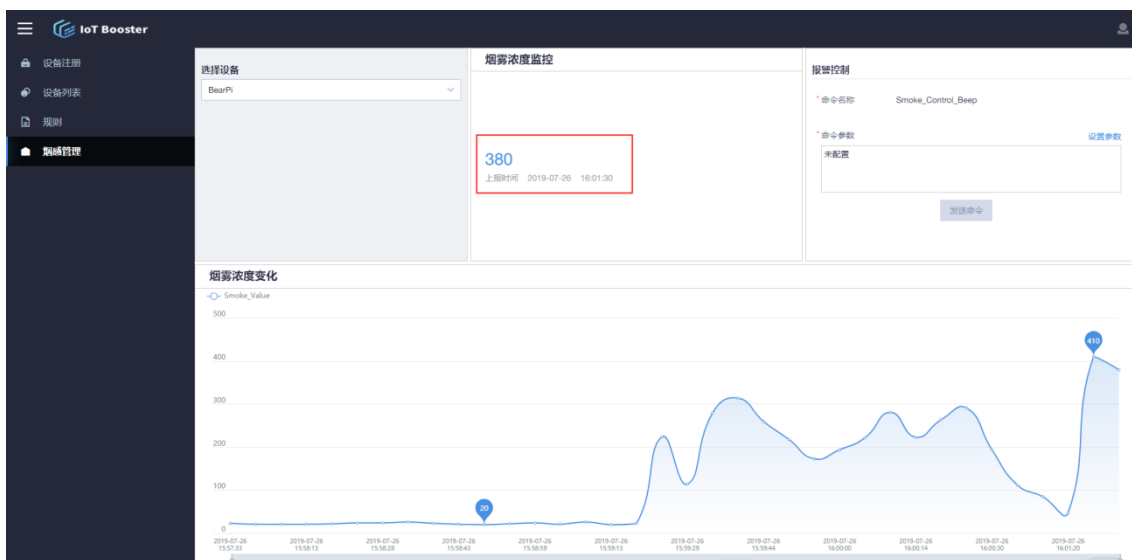


图 11-11 烟雾数值变化

2. 将开发板移至通风处，查看“烟感管理”中烟雾浓度监控数值如图 11-12 所示和报警器，当烟雾浓度<50 报警器应该自动关闭报警声。



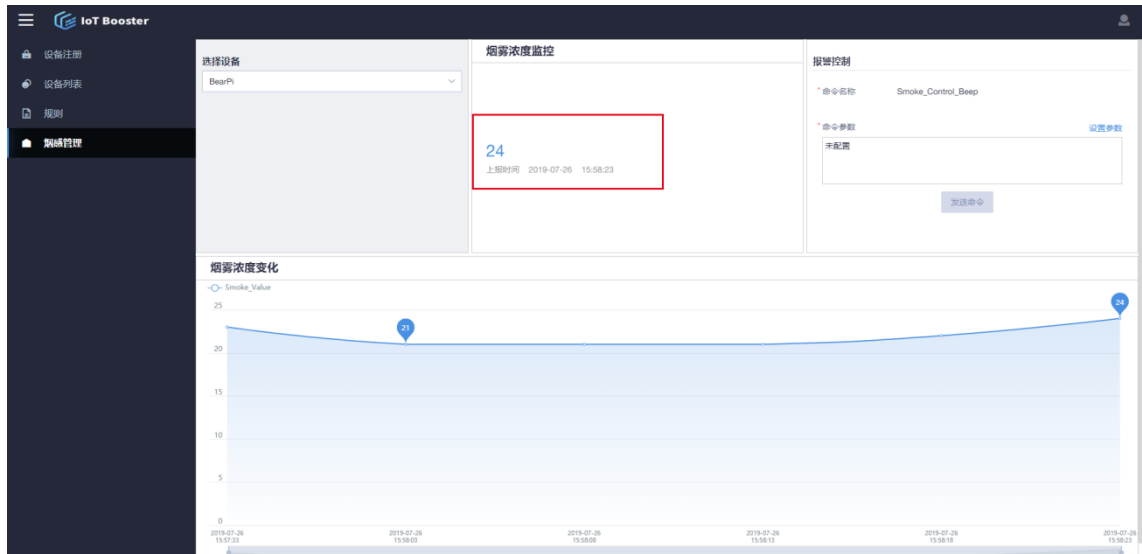


图 11-12 烟雾数值变化

---结束

## 技术支持获取

若学习及开发过程中有任何问题，可以进入华为 IoT 论坛进行提问，或者添加小助手获取帮助。



华为 IoT 论坛-小熊派版块



小熊派小助手微信号: BearPi\_Helper