

基于 HarmonyOS 的智能家居 App 设计与实现

孙 健

(天津电子信息职业技术学院, 天津 300350)

摘要:文章采用 HarmonyOS+云的技术方案,设计了一款智能家居 App,可根据室内温度控制空调开关,并实现数据展示、设备控制、智能告警等功能,智能调节室内温度,达到居住环境智能化控制的效果。

关键词:HarmonyOS;智能家居;云平台

中图分类号:TP311 **文献标识码:**A

文章编号:1009-3044(2023)09-0046-04

DOI:10.14004/j.cnki.ckt.2023.0436

开放科学(资源服务)标识码(OSID):



0 引言

HarmonyOS 是华为基于 OpenHarmony 开源项目开发的面向多场景智能设备的商用版本,在遵循 AOSP(Android Open Source Project)的开源许可上实现的新一代智能终端操作系统^[1]。在我国信创产业蓬勃发展的背景下,HarmonyOS 比 Android 在物联网领域具有先天优势^[2]。在物联网诸多应用场景中,智能家居是物联网技术融入生活的典型案例,开发一款基于 HarmonyOS 的智能家居 App 既能够使人们享受到科技发展带来的便捷,又能够发展壮大国产自主可控的鸿蒙生态。

1 需求分析

本文采用移动端 App+云的技术方案^[3],基于 HarmonyOS 的智能家居 App 以智能控制家居环境中的空调设备为例,通过温度传感器获取室内温度信息,并将温度的数据上报给物联网云平台^[4]。智能家居 App 从物联网云平台获取访问认证后,可查看当前各房间的温度信息,当需要时可以通过智能家居 App 远程控制家中的空调设备进行温度调节,具体功能的需求描述如下:

- 1) 进行页面的设计与开发;
- 2) 完成 App 与物联网云平台的交互认证;
- 3) 在页面上绑定设备;
- 4) 数据展示界面能展示卧室、客厅温度传感器的实时数据;
- 5) 设备控制界面能实现自动控制、手动控制两种模式对卧室空调和客厅空调控制工作;
- 6) 能实时监测设备状态(在线、离线、运转情况),有异常时能实现告警提示;

7) 能充分利用了组件的拉伸、占比、延伸和均分能力,配合媒体查询以及多态组件 Swiper,实现在不同的终端设备上有不同的显示方式,适配不同的设备屏幕形态,从而实现应用的“一次开发,多端部署”。

2 系统总体设计

基于 HarmonyOS 的智能家居 App 连接在华为云上用 Docker 容器技术部署的 ThingsBoard 物联网云平台,通过 MQTTBox 软件模拟物联网设备的数据上报给 ThingsBoard,智能家居 App 成功获取到 ThingsBoard 云平台的安全访问认证后,从 ThingsBoard 云平台获取设备数据并展示数据,当需要时进行设备监控和远程控制设备。

3 系统详细设计

3.1 云平台搭建

ThingsBoard 是一种开源物联网云平台^[5],智能家居设备数据要上报到 ThingsBoard 平台,需要在 ThingsBoard 平台上有对应的设备信息,因此需要在 ThingsBoard 平台上添加并配置设备相关的设备配置、设备之间的关联关系等信息。

1) 添加设备配置

设备配置主要用于确定设备类型,可通过设备配置信息对设备进行分类过滤,查找到合适的设备进行页面渲染。智能家居 App 需要添加 2 个设备配置,如表 1 所示。

表 1 智能家居 App 的设备配置信息

序 号	名 称	说 明
1	home_air	智慧家空调设备
2	home_temp	智慧家温度传感器设备

收稿日期:2023-01-15

基金项目:全国高等院校计算机基础教育研究会课题—人工智能+物联网(AIoT)应用开发立体化课程资源建设(2019-AFCEC-071);天津电子信息职业技术学院校级课题:《物联网移动软件开发》课程建设调研课题

作者简介:孙健(1988—),男,天津人,讲师,硕士,主要研究方向为软件开发,物联网技术。

2) 添加设备

在设备配置添加完成后,需要添加设备。ThingsBoard 平台中的设备主要作为真实设备的映射,所有真实设备需要在 ThingsBoard 平台中创建对应的设备。通过侧边栏单击设备切换主页,找到主页右上角“+”号,选择添加新设备,输入设备名称、设备标签、选择添加的设备配置、输入设备说明、单击添加,添加后可在设备列表查看到添加的设备信息。智能家居 App 需要添加 4 个设备,如表 2 所示。

表 2 智能家居 App 的设备信息

序号	名 称	标签	设备配置	说 明
1	home_saloon_temp	客厅温度	home_temp	客厅温度传感器设备
2	home_bedroom_temp	卧室温度	home_temp	卧室温度传感器设备
3	home_saloon_air	客厅空调	home_air	客厅空调设备
4	home_bedroom_air	卧室空调	home_air	卧室空调设备

3) 添加关联关系

创建好设备后,传感器设备和空调设备需相互关联产生联系,根据温度传感器获得的温度控制空调设备。在 ThingsBoard 平台中,单击侧边栏设备,选择并单击空调设备,选择关联,方向选择“从”,单击弹框的右上角“+”号,类型选择设备。找到相关传感器设备,单击添加,即可在关联列表查看到关联设备信息,如表 3 所示。

表 3 智能家居 App 的设备关联信息

序号	名 称	类 型	实体名称
1	home_saloon_air	设 备	home_saloon_temp
2	home_bedroom_air	设 备	home_bedroom_temp

4) 使用 MQTTBox 发送遥测数据

MQTT 是一种轻量级的基于客户端-服务器的消息发布/订阅传输协议,实现 MQTT 协议的通信需要有客户端和服务端。MQTT 协议的通信过程中有三种身份:发布者(Publish)、消息代理(MQTT Broker)、订阅者(Subscribe)。其中,消息的发布者和订阅者都是客户端,消息代理是服务器。消息发布者可以同时是消息的订阅者。MQTT 会自动构建底层网络传输,建立客户端到服务器的连接,提供两者之间的一个有序的、无损的、基于字节流的双向传输。ThingsBoard 物联网平台当接入 MQTT 协议的设备时,此时的 ThingsBoard 充当 MQTT 服务器(MQTT Broker),再使用 MQTTBox 软件充当客户端,客户端模拟真实设备通过 MQTT 协议连接到 ThingsBoard。

在搭建好 ThingsBoard 后,通过 MQTTBox 测试连接到该平台实现设备遥测的测试,按照如下步骤

实现。

第一步:双击打开 MQTTBox 软件,单击 Create MQTT Client(创建 MQTT 客户端),输入 MQTT Client Name(客户端名称),选择传输协议:mqtt/tcp,输入用户名:设备的访问令牌值,输入远程服务器的 IP 和 PORT(Host),选择服务质量(QoS)选择 1-Atleast Once,单击 Save,页面跳转后,出现 Connected 则证明 MQTT-Box 客户端已经连接 ThingsBoard 平台。

第二步:MQTTBox 连接成功后,输入遥测主题,选择服务质量(QoS)为 1-Atleast Once,输入载荷(Payload):“{“temp”:25}”,单击 publish。

第三步:打开 ThingsBoard 平台,通过侧边栏单击设备,找到并单击连接 MQTTBox 复制 Token 的设备,选择最新遥测值,可查看到 MQTTBox 提交的遥测信息。

通过以上方式,按下列参数,将 4 个设备的模拟遥测数据发送到 ThingsBoard 对应的设备中,智能家居 App 设备的主题和负载测试数据如表 4 所示。

表 4 智能家居 App 设备的主题和负载测试数据

序号	设 备	主 题	QoS	负载测试数据
1	home_saloon_temp	v1/devices/me/telemetry	1-Atleast OnceQoS	{“temp”:25}
2	home_bedroom_temp			{“temp”:25}
3	home_bedroom_air			{“value”:false}
4	home_saloon_air			{“value”:false}

3.2 数据展示

1) 设备数据展示页面的 UI 设计

设备数据展示页面主要用来展示传感器的状态信息和设备数据,根据智能家居 App 的功能需求,需要展示温度传感器数据以及空调设备的状态信息。在 view 目录下创建 Device.ets 文件。

2) 创建数据实体类

根据智能家居 App 的设备情况,需创建出温度传感器和空调两类设备对应的数据实体类,用来保存从云平台获取到的设备信息。

3) 封装从云平台获取数据的方法

页面设计好后,智能家居 App 需要从 ThingsBoard 平台获取温度传感器设备和空调设备的信息。在 TbCloud 类中添加获取温度传感器数据的方法、获取空调设备的方法、获取空调设备访问令牌 TOKEN 的方法、获取遥测数据的方法以及获取空调设备关联的温度传感器设备 ID 的方法。

4) 从云平台获取数据并展示在设备页上

在 TbCloud 编写好相应的方法后,则需在 Device

页面初始化的时候去获取设备的各种信息,所以需要在 Device.ets 文件中的 aboutToAppear() 方法中编写代码。首先需要判断是否登录过,接下来创建线程每 7 秒获取一次设备信息以及设备遥测信息。

5) 将设备页放到 Tabs 中

打开 Index.ets 页面,导入设备页,并修改 TabContent 中的设备页为 Device。

3.3 设备控制

1) 手动控制设备

通过单击空调设备的开关图标,达到控制空调设备的开关效果,所以需在 TbCloud 中创建修改设备遥测信息的方法。

2) 联动控制设备

在手动控制完成后,须实现联动控制设备,也就是设置相应温度传感器一个阈值,当前温度超过阈值时,启动空调设备,实现手动和自动的联动控制。因此需先创建弹框,通过单击设备的编辑按钮,弹出弹框对自动控制的温度阈值与自动化/手动化进行切换。故在 view 文件夹下创建 AirDeviceEditDialog.ets 文件。弹框页面最外层是通过 Column 组合而成,里面自上而下分别由文本框、横线、输入框、选择器以及确定和取消按钮组合而成,其中选择器以及确定和取消按钮分别用弹性布局进行包裹渲染。

至此,单击弹框后可将数据进行保存,并再次展示,但若是退出 App 至后台,则无法进行联动控制,故需要在 Device.ets 的初始化线程中添加一个业务,将持久化的数据读取并保存到空调设备集合的全局状态变量中,而当自动化数据可保存后,该数据需在获取到温度数据时判断是否需要开启空调设备,以达到真实的自动化效果。在 Device.ets 的初始化线程中,添加一个业务,判断当前温度是否超过自动化数据中设定的阈值,如果温度超过阈值,空调设备未打开并且已经开启自动化,则打开空调设备

3.4 智能告警

1) 创建告警实体类

告警页的开发首先需通过面向对象进行考虑,所以需在 model 文件夹中创建 AlarmInfo.ets 文件,创建出告警实体类,代码如下。

2) 持久化告警数据

在 common>database 文件夹创建 AlarmInfoDao.ets 用于持久化存储告警内容。所谓数据管理的存储方式,底层主要采用 SQLite 来实现,故首先需初始化数据库文件名、建表语句、表名称以及 RDB 数据库访问对象作为全局变量;在构造函数中通过传入上下文环境对象,获取 RDB 数据库访问对象,并将该对象设置到全局变量中并执行建表语句;通过 RDB 数据库访问

对象即可执行 SQL 实现数据的增删改查。

3) 将告警对象设置成全局变量

在设备初始化时,须先初始化告警对象并保存至全局变量中。在 EntryAbility 的 onCreate() 方法中,初始化告警对象,并设置成全局变量。

4) 查询告警的数据

从数据库中查询出来的数据,放在对应的集合中。在 viewmodel 文件夹中创建 AlarmViewModel.ets 文件,用来保存告警信息的数组。

5) 告警信息页面设计

在 view 目录下创建告警展示页面 Alarm.ets,用来展示告警信息列表。其中对告警项的布局进行封装,在 view 目录下创建 AlarmItem.ets 文件,通过 Flex 自左向右设置告警图标、室内温度和空调设备启动提示以及创建时间,其中室内温度和空调设备启动提醒通过 Flex 布局再次包裹,并在告警页的 ListItem 中进行引入,以便降低代码冗余,如图 1 所示。

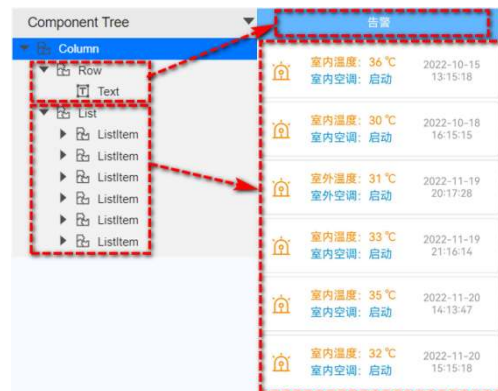


图 1 告警页及告警项布局

4 系统部署

HarmonyOS 的组件提供了不同的能力,用来实现一次开发,多端部署。智能家居 App 在开发过程中使用拉伸、占比、延伸、均分、多态等组件能力,实现一次开发,多端部署。

4.1 组件的拉伸能力

组件的拉伸能力无论在平板或手机屏幕中,当前 UI 组件都能够以相同的宽度或高度展示在屏幕中,也就是在设置 width、height、margin 和 padding 等属性的过程中,通过设置百分比,让该组件可以在不同的屏幕上都保持该宽高和间距,在智能家居 App 的开发过程中,以告警页面为例应用组件的拉伸能力,如图 2 所示。

4.2 设置组件占比能力

组件的占比能力即通过设置组件的权重属性 layoutWeight,将页面中留白的位置在高度或宽度上全部占满,通过此方式的配置不论在任意分辨率的显示屏

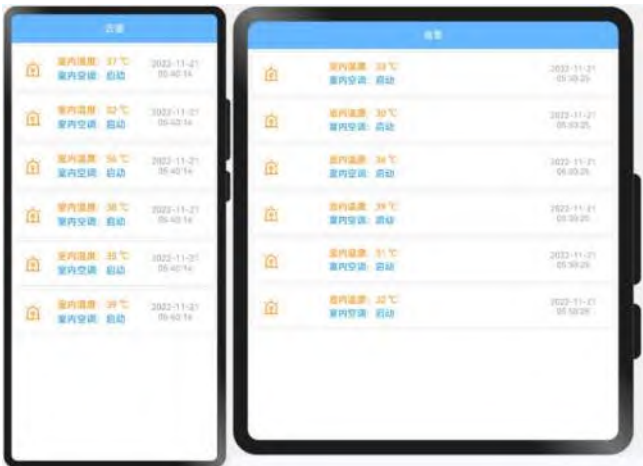


图 2 告警页组件拉伸效果

上,都是占满的效果,以告警页为例应用组件的占比能力。

4.3 组件的延伸能力

组件延伸能力即当页面高度不足以支持页面数据时,能够自动添加滚动条功能以达到在不同分辨率的显示屏上进行滑动来展示数据效果,通过 List 控件自带的效果即可展示出效果无须添加属性,此功能以告警页为例应用组件的延伸能力,如图 3 所示。



图 3 告警页组件延伸效果

4.4 组件的均分能力

组件均分能力即通过 Flex 布局将组件水平或垂直方向地形成合适的摆放效果,以达到在不同的屏幕分辨率下都能够达到组件均分的效果。在智能家居 App 中的告警项就是通过 Flex 的水平均分达到效果,如图 4 所示。



图 4 告警页组件均分效果

4.5 多态组件

多态组件指的是 Swiper 组件即轮播图组件,通过该组件的 displayCount 函数,传入屏幕中需展示的页数数量,达到不同的展示效果。在该功能下,只需计算出当前屏幕的分辨率即可根据不同分辨率展示页数数量,而此功能即媒体查询功能,即通过媒体查询出当前的屏幕分辨率,根据分辨率设置 Swiper 组件的 displayCount 参数,即可实现一次开发多端部署的能力,如图 5 所示。



图 5 多态组件加媒体查询效果

5 结论

本文主要完成了基于 HarmonyOS 的智能家居 App 的设计与实现,结合 ThingsBoard 云平台实现了智能家居数据展示、设备控制、智能告警等工作,并通过多种部署能力达到一次开发,多端部署的效果。

参考文献:

[1] 潘炳征. 基于鸿蒙的分布式创作播报应用研究[J]. 价值工程, 2022,41(23):166-168.
[2] 文丹. 基于 Android 的智能家居交互系统设计与开发[J]. 电子制作,2023,31(1):62-66.
[3] 范晓红. 基于云+Android 的物联网智能家居软件设计与实现[J]. 电子制作,2023,31(2):46-48,67.
[4] 赵宏林,廉小亲,郝宝智,等. 基于物联网云平台的空调远程控制系统[J]. 计算机工程与设计,2017,38(1):265-270.
[5] 张吉圭. 基于开源 AIoT 物联网平台的智慧牧场仿真实现[J]. 智慧农业导刊,2021,1(9):18-21.

【通联编辑:唐一东】