

做一个可以远程管理的虚拟语音播报员

郑祥 浙江省温州市第四中学

涉及学科：信息技术

在日常的校园广播任务中,学校广播社的学生们、政教处和团委的老师们需要从教室、行政楼的政教处和团委办公室出发,穿过整个校园到广播室里进行烦琐的设置后才能向全校的师生播送重要的通知。当遇到紧急情况时,他们还要百米冲刺,前往广播室。

于是,笔者便产生了“智能虚拟语音播报员”的想法——只要在校园内,教师通过手机发送通知消息,广播室内的主机就会将消息合成为语音通过校园广播系统播报给全校师生。这个作品的编程重点在于消息的远程传输和接收。

● “虚拟语音播报员”的技术分析

“虚拟语音播报员”作品主要由消息发送端、物联网平台和语音播报端三个部分组成。用户通过“消息发送端”编辑待通知的消息内容,在确认内容无误后将消息发送至“物联网平台”进行存储和转发;“语音播报端”的程序则实时接收物联网平台上的最新消息,并将文字内容转换为语音的形式通过校园广播播报出来。具体流程如图1所示。

● 物联网服务器的选择

物联网(Internet of Things,简称IoT),是指各类事物通过传感器和控制设备,根据约定的标准或开放协议与互联网连接,实现数据的存储、传递和交互。简而言之,物联网可以理解为“物物相连的互联网”,其目标是让万物都能联网实现沟通对话。本案例的作品中就应用了物联网平台的数据信息的存储和传递功能。常见的物联网平台有OneNET(中国移动物联网平台)、阿里云等,由于本案例中的作品主要应用在校园局域网中,因此采用简单、易用的SIoT物联网平台。

SIoT物联网平台的架设:

①下载SIoT物联网文件,下载网址:https://siot.readthedocs.io/zh_CN/latest/2.setup/01_download.html。

②运行SIoT物联网服务器:

将SIoT物联网文件复制到校园网内的计算机上进行解压,并运行文件夹中“SIoT.exe”应用程序,即可完成SIoT物联网服务的架设,如下页图2所示。

③登录SIoT物联网平台:通过服务器的IP地址即可远程访问SIoT物联网平台,成功登录SIoT物联网平台后可以进行“项目/设备”的查看、创建、删除等操作,登录用户名为siot,密码为dfrobot,系统界面如下页图3所示。

● “消息发送端”的开发

“消息发送端”的开发可以有很多形式,如网页、PC端应用程序。本案例中选择了对学校教师来说最为方便的手机App应用程序的形式来实现消息的编辑和发送。手机App应用开发工具很多,笔者选择了可视化编程且十分容易上手的App Inventor2。

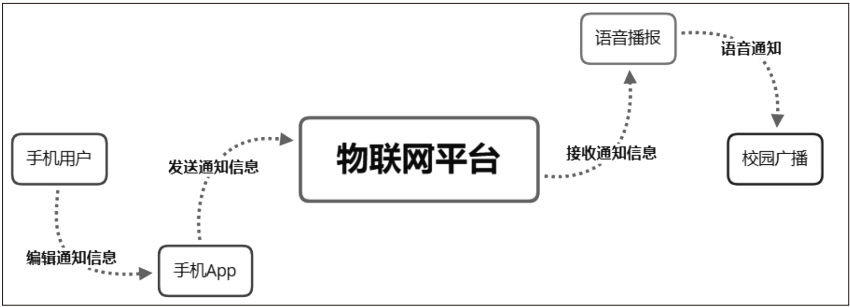


图1 虚拟语音播报员工作流程



图2 运行“SIoT.exe”应用程序



图3 SIoT系统界面



图4 界面设计

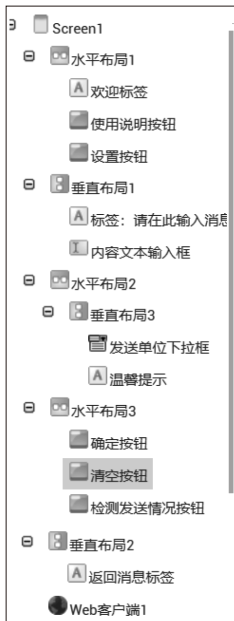


图5 界面设计组件

发送消息	Http://[SIoT的IP]:8080/publish?topic=[项目主题]&msg=[待发送消息]&iname=siot&ipwd=dfrobot
接收消息	Http://[SIoT的IP]:8080/lastmessage?topic=[项目主题]&iname=siot&ipwd=dfrobot



图6 “消息发送端”核心代码

① 新建项目：
虚拟语音播报员。

②消息发送端界面设计:界面设计中主要采用的组件和界面设计效果,其中“Web客户端”组件是非可视组件,也是实现消息发送的关键组件,如图4和图5所示。

③“消息发送端”逻辑设计:在逻辑设计方面,“消息发送端”的核心代码就是消息的发送。SIoT物联网平台提供了WebAPI接口,通过WebAPI接口可以轻松实现消息数据的接收和发送功能,具体代码如下表所示。

通过“Web客户端组件”和“WebAPI接口”就能轻松实现消息的发送功能,具体如图6所示。

“语音播报端”的开发

“语音播报端”的核心功能是从SIoT物联网平台上接收消息数据,并且借助语音合成技术(Text to Speech,TTS),把文本信息转化成语音信息。考虑到Mind+编程软件同时具备了MTQQ消息收发模块和语音合成模块,笔者选择了这个简单的图形编程工具来编写这个程序。

MQTT模块和文字朗读模块在“Mind+”编程软件的“网络服务”中,点击加载即可。

1.界面设计

在Mind+中新建一个列表变量,并设置显示在舞台中,如下页图7所示。

2.“语音播报端”的核心代码

“语音播报端”涉及的MQTT模块和文字朗读模块,需进行初始化,如下页图8、图9所示。其中“04/001”是MQTT消息的主题名称,可以自定义,只要这个主题名称中间有“/”即可。

“语音播报端”的核心代码消息数据接收和语音合成的程序,如下页图10所示。

系统应用和测试

“虚拟语音播报员”的消息发送端和语音播报端都开发完成了,只需在校园广播系统中进行一些简单的设置,就能实现“虚拟语音播报员”了。

1.安装“消息发送端”App应用程序

在安卓手机/平板中安装“消息发送端”的App应用。



图7 “语音播报端”的界面设计

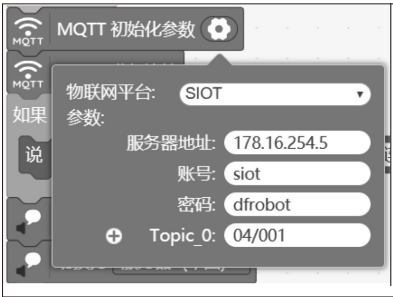


图8 MQTT初始化参数



图9 初始化程序

2. 运行“语音播报端”程序

在一台计算机中运行“语音播报端”程序,等待语音播报的内容更新。

3. 在校园广播系统中接入“语音播报端”的音频输出

将运行“语音播报端”程序的计算机作为音源接入校园广播系统,具体有两种方案。

方案1: 在校园广播系统中,通过定时功能设置播报音源为运行了“语音播报端”程序的计算机音频输出,如图11所示。该计算机通

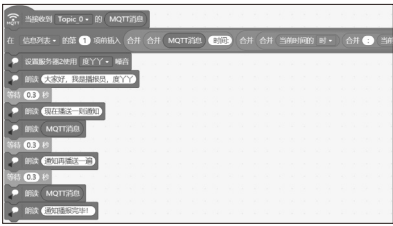


图10 消息数据接收和语音合成的核心代码

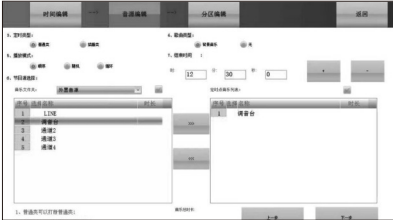


图11 校园广播系统定时点音源设置



图12 “消息发送端”App运行界面



图13 “语音播报端”的运行效果

过调音台作为外置音源接入校园广播系统。

方案2: 在校园广播系统中,找到校园的总音频功放设备,并将运行“语音合成端”程序的计算机音频输

出端接入到总音频功放设备即可。

4. 调试运行

在“消息发送端”App中输入待播放的消息内容,“语音播报端”通过“SIoT物联网平台”接收消息内容,并转换为语音在校园广播系统中播报出来,运行效果如图12、图13所示。

经过测试,“虚拟语音播报员”从消息发出到语音播报的实现,有1秒钟左右的延迟,这取决于校园内网络的拓扑结构,但这完全不影响校园通知的播送效果。

拓展

“虚拟语音播报员”主要运用了物联网技术和语音合成技术,但经过拓展和修改也完全可以应用在其他物联网项目之中,如在消息的接收端进行开源硬件的拓展,也完全能够轻松实现“智慧农场”“智能家居”等远程控制的项目。

这个案例还可以选择其他的技术来实现。例如,“消息发送端”可以用“MQTTTool”之类的任何一款MQTT客户端工具,也可以用网页,甚至可以用Word、PowerPoint之类的软件。“语音播放端”可以用Python,在一个小巧的迷你电脑上就能运行。这个案例的最大价值在于,学生能够用技术去解决一些真实的问题,开发各种个性化的应用,让智慧校园变得名副其实。*e*