

双树莓派实现“隔空”同声传译

文/山东省招远第一中学 牟晓东

小时候,我们都玩过手工版的“传声筒”:底部挖孔的两个空铁制罐头盒子,通过一根长长的细管或棉线连接起来,进行“你说、我听”和“我说、你听”的同步“传话”。不过,如果通话双方使用的是不同的语种,这种简易设备就无法实现“同声传译”了。现在,通过树莓派和古德微机器人“积木”模块语句等进行开源硬件编程,再借助于网络进行语音信号的传输及处理,可以非常方便地制作出一个“隔空”同声传译机:一方说中文、另一方讲英文,实现中英文双语的双向语音及文字的互译和传输,从而可以进行无障碍的实时交流。

1. 实验器材及连接

所需的实验器材包括:树莓派及古德微扩展板各两块,全向麦克风两个,音箱两个,LED灯两支。

通讯双方的实验器材及连接方式均完全一致,分别将扩展板正确安装于树莓派上(注意四周要均匀小心用力);然后,将全向麦克风插入树莓派的USB接口,音箱插入树莓派的音频输出圆孔;最后,将LED灯按“长腿正、短腿负”的规则插入扩展板的5号插孔,给树莓派通电,启动操作系统(如图1)。

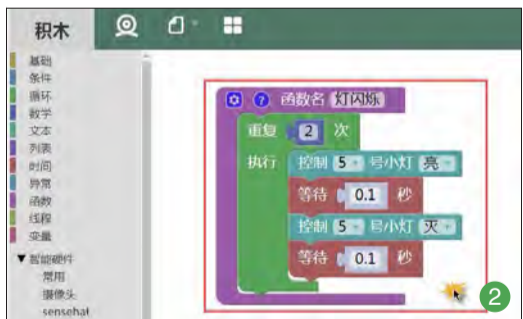


2. 中文通讯端编程

通过浏览器访问古德微机器人网站(<http://www.gdwrobot.cn/>),登录账号(luke007)后点击“设备控制”进入“积木”编程区开始编写程序:

(1) 编写控制LED的“灯闪烁”函数

新建一个名为“灯闪烁”的函数,实现的功能是控制插在5号GPIO接口的LED灯,当程序被语音唤醒后闪烁两次作为响应信号。先从“循环”中选择“重复‘10’次执行……”模块语句,修改为“2”;再从“智能硬件”-“常用”中选择“控制‘2’号小灯‘亮’”,作为循环体的第一个模块语句,并改为“5”号;接着加一个“等待0.1秒”,作用是控制LED灯持续亮0.1秒;最后,再添加一个“控制‘5’号小灯‘灭’”和“等待0.1秒”,实现LED灯“亮0.1秒、灭0.1秒,再亮0.1秒、灭0.1秒”的闪烁功能(如图2)。

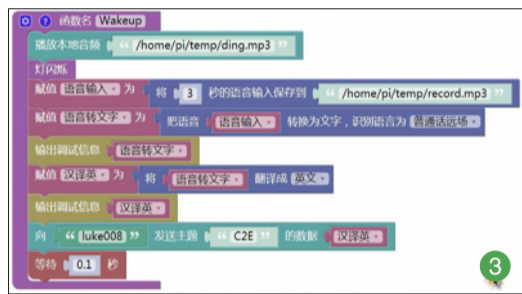


(2) 语音唤醒“Wakeup”函数

新建一个名为“Wakeup”的函数(注意必须是这个名字),其中的第一个模块语句是“播放本地音频‘/home/pi/temp/ding.mp3’”,作用是播放树莓派操作系统/home/pi/temp目录中的ding.mp3音频文件,即发出“叮”的一声;接着,调用刚刚编写的“灯闪烁”函数;然后,分别建立名为“语音输入”和“语音转文字”两个变量,将“语音输入”变量赋值为“将‘3’秒的语音输入保存到‘/home/pi/temp/record.mp3’”,作用是将用户通过全向麦克风输入的语音生成MP3文件并保存至树莓派的temp目录中,命名为“record.mp3”;将“语音转文字”变量赋值为“把语音‘语音输

入’转换为文字,识别语言为‘普通话远场’”,作用是将变量“语音输入”(对应刚刚生成的“record.mp3”音频文件)进行语音转文字的识别操作;接着,通过“输出调试信息‘语音转文字’”模块语句,实现在程序LOG调试区输出显示识别转换出的文字信息。

再建立变量“汉译英”,赋值为“将‘语音转文字’翻译成‘英文’”,作用是将识别出的中文文字信息翻译成英文,然后也是在LOG调试区输出显示;接着,从“物联网”-“常用”中选择并设置好“向‘luke008’发送主题‘C2E’的数据‘汉译英’”模块语句,其中的“luke008”对应着英文通讯端的账号(中文通讯端账号是“luke007”),主题“C2E”的意思是“中文对英文”(Chinese to English),可自定义;该模块语句的作用是将变量“汉译英”中保存的英文翻译文字信息发送至英文通讯端(如图3)。



(3)调用Wakeup函数进行数据处理

首先,从“人工智能”-“语音识别”中选择“小度小度关键词唤醒,请创建一个Wakeup新函数”模块语句,作用是通过“小度小度”关键词来唤醒执行对应的“Wakeup”函数;接着,从“物联网”-“常用”中选择“监听主题‘E2C’并设置初始值‘0’”模块语句,作用是监测英文通讯端发送过来的“E2C”主题数据(English to Chinese);然后建立一个“重复当‘真’执行……”循环结构,通过“如果……执行……”选择结构对“物联网是否收到新数据”进行判断,条件成立的话,建立变量“对方的中文译文”并赋值为“获取主题‘E2C’的数据”,也就是对应英文通讯端在经过“英译汉”后的文字信息;接着,选择“人工智能”-“语音识别”中的“播放语音‘对方的中文译文’发音人为‘0’语调为‘5’语速为‘3’”模块语句,实现对方语音信息译文的语音播放功能(如图4)。



程序编写完毕,点击“保存”按钮将程序保存为“汉语输入端”。

3. 英文通讯端编程

同样是访问古德微机器人网站,并且登录账号(luke008)后进行“积木”编程区开始编写程序。英文通讯端的程序与中文通讯端几乎完全一致,可先通过下载luke007代码后,用上传或分享的方式在编程区打开,然后进行少量的修改。



“灯闪烁”函数保持不变;在“Wakeup”函数中,先将变量“语音转文字”的赋值模块语句中的识别语言改为“英语”;再使用变量“英译汉”来替换之前的“汉译英”,将“发送主题”模块语句改为“向‘luke007’发送主题‘E2C’的数据‘英译汉’”;主程序中的“监听主题”中将“E2C”改为“C2E”,将“对方的中文译文”变量替换为“对方的英文译文”,其值为“获取主题‘C2E’的数据”(如图5)。

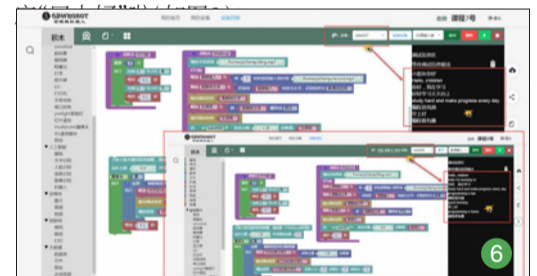
其他的不必作改动,将程序保存为“英语输入端”。

4. 测试树莓派“隔空”同声传译的效果

将两个树莓派分隔开数米的距离,防止同时语音唤醒;接着,分别在两个编程界面中点击“连接设备”按钮,出现五个绿色对钩后表示连接成功,再点击“运行”按钮进行测试。

在中文通讯端对着麦克风喊话:“小度小度”,音箱会先发出“叮”的一声回应,接着LED灯会闪烁两次,说明已经完成了语音唤醒;然后继续喊话:“小朋友,你好!”,这时会先在程序LOG调试区显示语音识别的结果:“小朋友,你好!”,接着又会显示对应的英文翻译:“Hello, children”;在英文通讯端也会很快收到该英文翻译“Hello, children”并同样显示在LOG区,接着将它转换为语音形式并通过音箱进行语音的播放;再对着麦克风喊话:“小度小度”,音箱同样也会先发出“叮”的一声回应、LED灯闪烁两次,然后继续喊话:“Hello, I'm learning to program”。由于编程时设置了3秒的语音输入限制,语音信息后面的单词“program”被忽视,此时LOG显示区只输出了“Hello, I'm learning to”和中文翻译“你好,我在学习”。同样,中文通讯端也是如此。

继续测试,比如中文的“好好学习,天天向上”被翻译为“Study hard and make progress every day”,英文的“Programming is fun”和“Programming is funny”则分别被翻译为“编程是风扇”和“编程很有趣”,“Good morning”当然就对



相关视频关注“壹零社”公众号观看

找不到编程工具和源代码?别着急,关注“壹零社”公众号,点击下方编程相关按钮,就可以用百度网盘下载到Scratch以及其他常用编程工具,还有更多编程文章和源代码等你免费来拿。



加壹零社小助理为好友,随时回答您的各种疑问