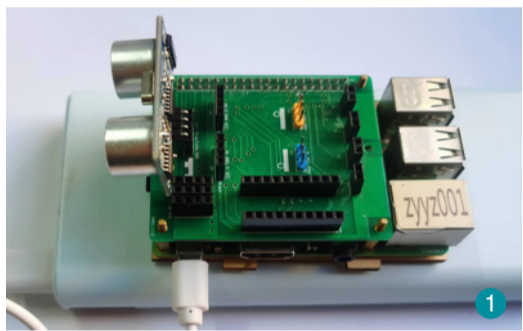


树莓派绘制超声波测距轨迹曲线

文/山东省招远第一中学 牟晓东

在开源硬件编程中,我们都会实时获取测量生成的实验数据,比如温湿度传感器对环境温度和湿度进行监测而产生的数据。通常情况下,我们会先将这些实验数据写入文件(比如CSV格式文件),然后进行数据整理,再绘制拟合图像,最终总结出实验结论。其实,对于一些非常直观的测量数据,我们可以“跳”过数据整理,直接将它们“画”出来,以动态曲线的形式进行展示。以超声波传感器的“测距值”为例,我们在树莓派中分别通过Python代码编程和古德微“积木”图形化编程两种方法,来实现超声波测距轨迹曲线的动态绘制。

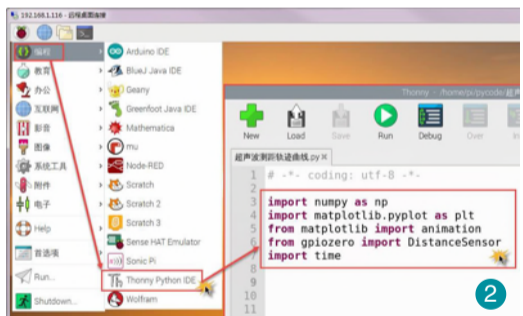
实验器材包括树莓派3B+和古德微扩展板各一块,HC-SR04超声波传感器一个,移动电源一个。首先,将超声波传感器的四个引脚(VCC、Trig、Echo和GND)对应插入扩展板的20和21号引脚,方向朝外;然后,通过数据线将树莓派与移动电源连接,通电,启动操作系统(如图1)。



一、Python代码编程绘制超声波测距轨迹曲线

1. 远程登录树莓派,导入待用的库模块

运行Windows的“远程桌面连接”,输入树莓派的IP地址(比如192.168.1.116),登录成功后点击“编程”-“Thonny Python IDE”菜单,启动Python编辑器,导入库模块:“import numpy as np”、“import matplotlib.pyplot as plt”、“from matplotlib import animation”、“from gpiozero import DistanceSensor”和“import time”(如图2),其中的numpy、matplotlib.pyplot和animation是负责数据生成和绘制曲线图像的,gpiozero库中DistanceSensor类是用来控制超声波传感器的。

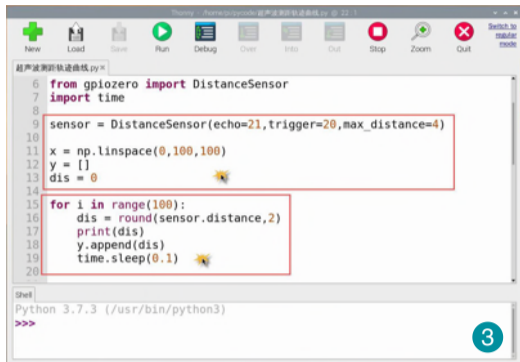


2. 初始化及数据获取

首先,建立变量sensor,对超声波传感器的类进行实例化:“sensor = DistanceSensor(echo=21, trigger=20,max_distance=4)”,注意其中的“echo=21”和“trigger=20”分别对应超声波传感器的信号回收与发射引脚号21和20,而“max_distance=4”的作用是设置超声波传感器的最大检测距离为4(单位是“米”),如果不设置,则最大只能测量至1米的极限值(HC-SR04超声波传感器可提供2cm-400cm的非接触式距离感测功能)。

接着,进行图像绘制前的数据准备,横坐标x是通过numpy库中的linspace()来生成,范围为从0至100:“x = np.linspace(0, 100, 100)”;纵坐标y先设置为一个“空列表”:“y = []”,建立变量dis,作用是保存超声波传感器的“测距”值,初值为0:“dis = 0”;然后建立一个for循环结构(for i in range(100):),其中的第一行语句“dis = round(sensor.distance,2)”作用是为变量dis赋值,通过round()函数来保留两位小数;第二行语句“print(dis)”是将dis保存的测距

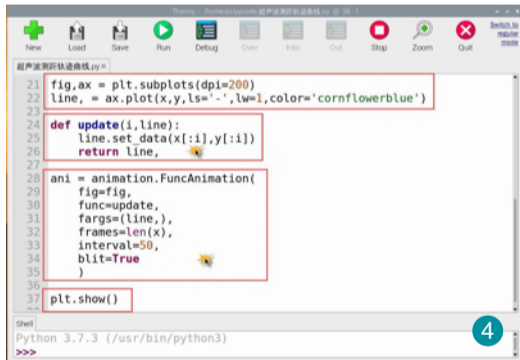
值在屏幕上同步输出显示;第三行语句“y.append(dis)”作用是将测距值追加至列表y中;第四行语句“time.sleep(0.1)”是控制程序暂停0.1秒钟,然后进入下一次循环,获取下一个测距值(如图3)。



3. 绘制“测距值”轨迹动态曲线图

绘制动态(包括静态)曲线图需要创建画布和子图:“fig,ax = plt.subplots(dpi=200)”,其中的“dpi=200”是设置显示分辨率的,可根据情况自行设置;接着,进行曲线的绘制:“line, = ax.plot(x,y,ls='-',lw=1,color='cornflowerblue)”,其中的参数含义分别是横坐标值x、纵坐标值y、绘制线的类型与宽度,以及线的颜色;然后,自定义建立一个帧画布的更新函数update(),其中的第一个语句是“line.set_data(x[:i],y[:i])”,作用是将动态检测生成的每一组数据(横坐标x值和纵坐标y值)进行“切片”操作后,作为数据源提供给后面的animation.FuncAnimation进行曲线绘制;第二个语句“return line”,是将line值返回。

接着,进行曲线动画的生成,建立变量ani,调用animation.FuncAnimation()并进行相关参数的设置,其中包括:语句“fig=fig”,作用是设置绘制动态图的画布名称;语句“func=update”,作用是调用自定义动画函数update(),注意不要带括号;语句“fargs=(line,)”,作用是除frames之外需要向func传递的参数;语句“frames=len(x)”,作用是控制生成的动画长度(一次循环所包含的帧数目);语句“interval=50”,作用是设置更新的频率,单位是ms(毫秒);语句“blit=True”,作用是选择更新所有点还是仅更新产生变化的点;最后,通过语句“plt.show()”将图像进行显示(如图4)。

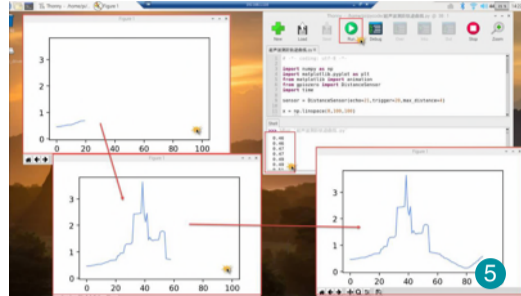


4. 程序的运行测试

将程序保存为“超声波测距轨迹曲线”后点击“运行”按钮,然后控制树莓派带动超声波传感器转动,或朝向墙面等障碍物靠近与远离,大约持续10秒钟后(程序的循环结构:100×0.1=10),此时,就会在屏幕上看到有超声波传感器“测距值”数据不断出现:0.46、0.49……接着,就会弹出一个名为“Figure 1”的图片窗口,绘制的正是动态的超声波测距轨迹曲



线,而且会不断循环播放之前程序所监测获取的100个数据(如图5)。



程序可以重复运行,获取到不同的超声波测距值,对应不同的动态轨迹曲线。而且,我们还可以修改程序中的循环次数(比如300),或是修改曲线绘制的速度(在变量ani调用animation.FuncAnimation()中“interval=50”处设置),当然也包括曲线的颜色、宽度等,都可以尝试。

二、图形化编程绘制超声波测距轨迹曲线

1. 登录古德微机器人平台,完成“积木”图形化编程

访问古德微机器人网站(http://www.gdwrobot.cn),登录进入自己账号后点击“设备控制”进入“积木”界面。

首先,进行物联网服务器的设置,从左侧“物联网”-“常用”中将“设置物联网服务器”功能模块拖动至主界面,保持其默认的服务器“www.gdwrobot.top”和端口“1883”不变,下方的用户名和密码均保持为空;接着,建立一个重复执行100次的循环结构,建立名为“超声波测距”的变量,为其赋值为“智能硬件”-“常用”中的“超声波测距”项,并进行保留两位小数的四舍五入处理;然后通过“输出调试信息‘超声波测距’”,在LOG调试信息区将超声波传感器每次获取的“测距”值进行显示输出;接着,从“物联网”-“常用”中拖动出“发送主题”功能模块,并且设置好对应的信息:“向‘luke007’发送主题‘distance’的数据‘超声波测距’”,其中的“luke007”对应自己的登录账号,主题名为“distance”;最后,添加一条“等待0.3秒”的循环间隔时间,点击“保存”按钮将程序保存为“绘制超声波测距轨迹曲线”(如图6)。



2. 设置动态曲线绘制参数

点击“更多功能”按钮后再点击“采集数据”按钮,进行动态曲线绘制的相关设置:将“采集标题”设置为“绘制超声波测距轨迹曲线”;第二行的服务器地址和端口号同样均保持默认不变(与“积木”图形化编程一致),可尝试点击后面的“测试连接”按钮进行测试,正常的话会出现“连接成功”的提示;第三行的“功能描述”设置为“使用超声波传感器进行绘图”(可自定义),“树莓派编号”处填写为“luke007”、“主题”填写为“distance”,同样是与图形化编程一一对应,后面的数值单位设置为“厘米”;最后,点击“保存”按钮,将本次的数据采集参数设置项目保存(如图7)。

