

DRGenerator 信号发生器用法

DRGenerator 信号发生器控件提供一个标准的软件信号源，可以生成正弦波、方波、三角波和白噪声信号，并将所产生的信号通过直连绑定的形式传递给其它控件或用户自定义函数。该控件为后台控件，在 GUI 界面上无显示。

在 Python 程序 Tkinter GUI 界面中创建一个 DRGenerator 信号发生器控件的命令和语法格式如下：

```
hl=DRGenerator(option=value, ...)
```

控件创建时有一组可选的输入参数，包括：

stype: 信号类型，默认值=0；0/1/2/3=正弦波/方波/三角波/白噪声

Fs: 采样频率，默认值=44100

N: 采样长度，默认值=8192

A: 信号幅值，默认值=0.6

F: 信号频率，默认值=100

P: 信号相位，默认值=0

T: 数据刷新定时间隔，默认值=200，单位：毫秒。

返回值 hl 则是该控件创建后的句柄变量名，可以通过 hl 对创建后的控件进行进一步的设定。

DRGenerator 信号发生器控件有 8 个属性函数，其功能定义如下：

hl.setVSONOff(v): 音频信号采集开关，收到数据流后，在启停之间切换。

hl.setVSType(v): 设定信号类型，0/1/2/3=正弦波/方波/三角波/白噪声

hl.setVSFs(v): 设定采样频率

hl.setVSLen(self,v): 设定采样长度

hl.setVSAmp(self,v): 设定信号幅值

hl.setVSFre(self,v): 设定信号频率

hl.setVSPha(self,v): 设定信号相位

dx,yarray=hl.getArrayTwoX(): 获取当前信号采样数据

下面是一个全参数的 DRGenerator 信号发生器控件创建实例：

```
hl=dsp.DRGenerator(0,44100,4096,0.6,100,0,250)
```

通常，可以选择设定部分选择参数的形式进行设定，其余可选参数采用默认值。创建实例如下：

```
hl=dsp.DRGenerator(F=200)
```

创建后可以通过返回的控件句柄对其属性函数进行操作。

DRGenerator 信号发生器控件可以通过绑定的方式将采集的信号数据传递给绑定的回调函数或其他控件，格式为：

```
hl.addCallBack2D(fun)
```

被绑定的回调函数或另一控件的 set 函数是一个二维数组函数，收到的是信号曲线的 x 和 y 数组值。回调函数形式为：

```
def setArrayTwo(xArray,yArray):
```

...

或：

```
hl.addCallBack2DX(fun)
```

被绑定的回调函数或另一控件的 set 函数是单值+一维数组函数，收到的是信号曲线的采样间隔和信号数据，其数据传输效率更高一些。回调函数形式为：

```
def setArrayTwoX(dx,yArray):
```

```
.....
```

当采用 DRPython 进行图形化编程时，控件的创建代码和直连绑定代码的生成均可以由机器实现；控件创建只需要将所需控件拖拽到 GUI 布局区，而直连绑定则需要在 GUI 布局区的控件间进行连线操作。DRGenerator 信号发生器控件可以作为单数值输出控件，在连线状态下在控件上点击鼠标，则弹出端选择；选择完成后，点击另一控件，则弹出输入端选择；若输出端和输入端类型一致，则完成连线。输出参数包括：当前的产生的信号数据(addCallBack2D 或 addCallBack2DX)。



DRGenerator 信号发生器控件也可以作为输入控件，在连线状态下当上一控件已选输出函数后，在控件上点击鼠标，则弹出连线选择菜单；选择完成后，若输出端和输入端类型一致，则完成连线。输入参数包括单数值的控件运行开关(setVSONOff)，收到单值数据流后在功能在“开/关上切换”；单数值的信号类型(setVSType)，单数值的信号采样频率(setVSFs)，单数值的信号采样长度(setVSLen)，单数值的信号幅值(setVSAmp)，单数值的信号频率(setVSFre)，单数值的信号相位(setVSPha)。



下面是 DRGenerator 信号发生器控件的创建和应用示例小程序，功能是在 GUI 界面上插入 2 个按钮控件，一个 DRGenerator 信号发生器控件、2 个 Plot 曲线控件和 1 个用户自定义回调函数，然后分别通过直连绑定的方式，用一个按钮控件控制信号发生器的启/停；用另一个控件获取当前信号采集数据。

DRGenerator 信号发生器控件应用示例小程序代码如下：

```
===引用的包
```

```

import tkinter as tk
import drvi.drviControlls as dr
import drvi.drviDSP as dsp

#用户自定义的回调函数
def muserSingle6(v):
    dx, yarray=mGen2. getArrayTwoX()
    mPlot4. setValue2DX(dx, yarray)

#==创建 Tkinter GUI 窗口
win= tk. Tk()
win. geometry('816x497')
win. config(bg="#ddeeef")
#==插入 2 个开关按钮控件
mButton1=dr. DRButton(win, 32, 40, 100, 40, txt=' Run/Stop')
mButton5=dr. DRButton(win, 27, 257, 100, 40, txt=' GetData')
#==插入信号发生器控件
mGen2=dsp. DRGenerator(F=200)
#==插入 2 个曲线控件
mPlot3=dr. DRPlot(win, 159, 34, 700, 159)
mPlot4=dr. DRPlot(win, 151, 217, 700, 180)
#=直连绑定
mButton1. addCallBackSingle(mGen2. setVSONOff)
mButton5. addCallBackSingle(muserSingle6)
mGen2. addCallBack2D(mPlot3. setValue2D)
#==主程序循环
win. mainloop()

```

pyqt 版本中的用法说明:

DRVI 有继承自 Tkinter 和继承自 pyqt 两个不同库,除引用的 GUI 库(Tkinter/pyqt)和框架程序略有不同外, DRVI 部分控件的创建、布局和连线是完全相同的。因均继承自 Numpy 和 Scipy, 该函数在两个版本中用法无区别。