

DRKnob 旋钮用法

DRKnob 旋钮控件提供一个可旋转调节数字量输入的控件，供用户鼠标拖动改变数字值，然后将数值通过回调函数或直连形式传递给下一模块，驱动其运行。



DRKnob 旋钮控件的图形显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRKnob 旋钮控件的命令和语法格式如下：

```
hl=DRKnob(win,x,y,w,h,option=value, ...)
```

其中输入参数 win 为 Tkinter 库创建的 GUI 窗口句柄，[x,y,w,h]四个参数为该控件插入 GUI 窗口 win 时的左上角坐标[x,y]和该控件的宽、高[w,h]。另外是一组可选的输入参数，包括：

- cb: 背景色，默认值='#004466'
- cf: 前景色，默认值='#222222'
- cfl: 前景色 1，默认值='#888888'
- txt: 表头量程显示文本，默认值='0,100'
- mlow: 最小值，默认值=0
- mhigh: 最大值，默认值=100
- val: 初始值，默认值=10
- fsiz: 字符点数，默认值=12

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。

DRKnob 旋钮控件有 2 个属性函数，功能如下：

- hl.setValueSingle(v): 设定控件值
- v=hl.getValueSingle(): 取控件值

下面是一个全参数的 DRKnob 旋钮控件创建实例：

```
hl=dr.DRKnob(win,223,76,120,120,'#004466','#222222','#888888','0,100',0,100,20,12)
```

通常，可以选择设定部分选择参数的形式进行设定，除基本参数 win、[x,y,w,h]和用变量名设定的可选参数外，其余可选参数采用默认值。创建实例如下：

```
hl=dr.DRKnob(win,276,287,120,120,val=50)
```

创建后可以通过返回的控件句柄对其属性函数进行操作。

DRKnob 旋钮控件可以通过回调函数绑定的方式将用户的输入传递给绑定的回调函数，格式为：

```
hl.addCallBackSingle(fun)
```

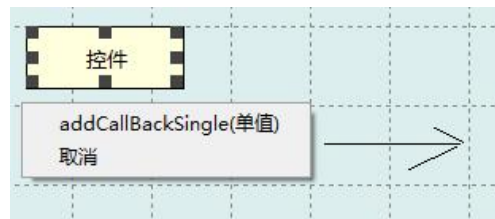
被绑定的回调函数或另一控件的 set 函数是一个单数值函数，收到的是输入框中的数字。回调函数形式为：

```
def setValueSingle(v):
```

.....

通过回调函数绑定方式，就可以将用户的输入传递到自定义的回调函数中或传递给直连的另一控件。

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。旋钮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。控件值可以通过单值端口(addCallBackSingle)输出。



旋钮控件也可以作为单数/字符串输入控件，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。从单值端口(setValueSingle)收到数据流后，则作为控件的显示值。



下面是 DRKnob 旋钮控件的创建和应用示例小程序，功能是在 GUI 界面上插入一个旋钮控件，一个标签控件、一个用户自定义函数，然后通过直连绑定方式，将 DRKnob 旋钮控件的值传递给标签控件和自定义函数。DRKnob 旋钮控件应用示例小程序代码如下：

```
##引用的包
import tkinter as tk
import DRKnob.DRKnobControlls as dr

#用户自定义的回调函数
def muserSingle3(v):
    print(v)

##创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('816x497')
win.config(bg="#ddeeee")
##插入旋钮控件
mKnob=dr.DRKnob(win, 100, 40, 120, 120, val=50)
##插入标签控件
mLab=dr.DRLabel(win, 221, 42, 200, 30, fsize=16)
##绑定用户回调函数
mKnob.addCallBackSingle(muserSingle3)
mKnob.addCallBackSingle(mLab.setValueSingle)
```

```
#==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。在 DRVI 的 pyqt 版本中,DRKnob 继承自 QDial,因此除了用 DRKnob 的参数化外观定义外,还可以直接用 QDial 的样式表来对其进行外观描述,并继承其样式表加载方法:

```
setStyleSheet(styleString)
```