

DRRecorder 音频视频信号采集用法

DRRecorder 音频视频信号采集控件提供一个计算机麦克风音频信号和摄像头视频信号(仅 pyqt 版本)采集模块,可以实时获取外部的音频/视频信号,并时通过数据流和回调函数形式返回采集的声音数据。



DRRecorder 控件图形显示效果图

Tkinter 版 DRRecorder 控件在使用前需引入 drviAudio 支持库:

```
import drvi.drviAudio as drAudio
```

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

```
hl=DRRecorder(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='#00aacc'

cf: 前景色, 默认值='#eeee00'

Fs: 音频信号采样频率, 默认值=44100, 单位: Hz

N: 信号采样长度, 默认值=8192, 单位: 点

T: 采样进程定时间隔, 默认值=200, 单位: 毫秒。 $T > 1000 * N / Fs$

fsize: 字符点数, 默认值=11

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。

DRRecorder 控件有 2 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。

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

下面是一个全参数的 DRRecorder 控件创建实例:

```
hl=drAudio.DRRecorder(win,211,27,100,40,'#00aacc','#eeee00',44100,8192,200,11)
```

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.DRRecorder(win,211,35,100,40,T=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。

DRRecorder 控件可以通过绑定的方式将采集的麦克风音频信号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，收到的是音频信号曲线的 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，收到的是音频信号曲线的采样间隔和音频信号信号数据，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。DRRecorder 麦克风录音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：当前的录音数据(addCallBack2D 或 addCallBack2DX)。



DRRecorder 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。输入参数为控件运行开关(setVSONOff)，收到单值数据流后在功能在“开/关上切换”。



下面是 DRRecorder 控件的创建和应用示例小程序，功能是在 GUI 界面上插入 1 个按钮控件，一个 DRRecorder 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麦克风信号采集的启/停；用 2 个 Plot 曲线控件显示录音信号波形和频谱。DRRecorder 控件应用示例小程序代码如下：

```
#==引用的包
import tkinter as tk
import numpy as np
from scipy.fft import rfft,irfft
import drvi.drviControlls as dr
import drvi.drviAudio as drAudio

#用户自定义的回调函数
def muser2d(t,y):
    tArray=t
    yArray=y
    N=len(y)
    N2=int(N/2)
    Fs=1/t[1]
    df=Fs/N;
    A=np.abs(np.fft.rfft(y))/N2
    mPlot3.setValue2DX(tArray,yArray)
    mPlot4.setValue2DX(df,A)

#==创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('1000x497')
win.config(bg="#ddeeee")
#==插入 1 个开关按钮控件
mButton1=dr.DRButton(win,32,40,100,40,txt='Run/Stop')
#==插入麦克风控件
mRecorder2=drAudio.DRRecorder(win,29,112,100,40)
#==插入 2 个曲线控件
mPlot3=dr.DRPlot(win,159,34,800,200)
mPlot4=dr.DRPlot(win,159,250,800,200)
#=直连绑定
mButton1.addCallBackSingle(mRecorder2.setVSOonOff)
mRecorder2.addCallBack2D(muser2d)
#==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，DRRecorder 继承自 QtMultimedia，支持音频与视频采样，其参数与 TK 版的创建参数略有不同，的命令和语法格式如下：

```
hl=DRRecorder(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='#00aacc'
cf: 前景色, 默认值='#eeee00'
fsize: 字符点数, 默认值=11
Fs: 音频信号采样频率, 默认值=44100, 单位: Hz
N: 信号采样长度, 默认值=8192, 单位: 点
nChunk: 采集的音频数据块数, 默认值=4
nFrame: 采样进程定时间隔(数据块数), 默认值=6

当视频采样时, DRRecorder 控件在采集音频信号的同时, 还可以通过绑定的方式将同步采集的图象信号传递给绑定的回调函数或其他控件, 格式为:

```
hl.addCallBackImage(fun)
```

被绑定的回调函数或另一控件的 set 函数是一个图象函数, 收到的是采集的图像帧。回调函数形式为:

```
def setImage(mImage):
```

```
...
```

其余属性参数和连线方式等则与 TK 版相同。

下面是 pyqt 版 DRRecorder 控件图象采集程序样例:

```
# 引用的库
import sys
from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import drvi.drviQTControlls as dr

#创建 GUI 窗体对象
app=QApplication(sys.argv)
app.setStyle("Fusion")
win=QWidget()
win.setWindowTitle('PyQt 程序')
win.resize(800,500)
win.setStyleSheet("background-color : #ddeeee; ")
#在 GUI 窗口插入控件和布局
mButton1=dr.DRButton(win,60,45,100,40,'#00aacc','#eeee00','运行/停止')
mRecorder2=dr.DRRecorder(win,228,46,100,40,'#00aacc','#eeee00')
mImage3=dr.DRImage(win,72,139,420,260,'#dddddd')
#连线
mButton1.addCallBackSingle(mRecorder2.setVSONOff)
mRecorder2.addCallBackImage(mImage3.setValueImage)
#显示 GUI 主窗口界面, 等待用户操作
win.show()
sys.exit(app.exec_())
```