

```

1  -*-*coding:utf-8*-*
2  #qpy:kivy
3  #=====
4  # Desarrollado por Diego Alberto Parra Garzón
5  # Bogotá D.C., Colombia
6  # noviembre 2015
7  # Esto es software libre con licencia GPL3
8  #=====
9  # clase Bluetooth Arduino referencia a pyserial
10 # El emparejamiento y conexión al puerto serial se realiza
11 # manualmente con bluetooth-manager
12 # dparra@opesai.org
13 import serial
14 import subprocess
15 from commands import getoutput
16 class ArduinoBluetooth:
17
18     def obtenerCorrienteEnchufe(self, Nombre):
19         try:
20             print puerto
21             self.Blue = serial.Serial(puerto, "9600")
22             self.conexion = True
23         except:
24             print "Fallo la función obtenerCorrienteEnchufe del módulo Ardutooth
Linux"
25             print "Recuerde logearse como usuario su"
26             self.conexion = False
27
28
29     # self.ss = serial.Serial()
30
31
32     def Escribir(self, Mensaje, *args):
33         if self.conexion == True:
34             self.Blue.write(Mensaje)
35             print "Mensaje enviado al dispositivo"
36         else:
37             print "Dispositivo no esta conectado"
38
39     def LeerNUMERO(self, *args):
40         CadenaDatos = ""
41         if self.conexion == True:
42             CadenaDatos = self.Blue.readline()
43         return CadenaDatos
44
45
46     def LeerCADENA(self, *args):
47         CadenaDatos = ""
48         if self.conexion == True:
49             CadenaDatos = self.Blue.readline()
50             print "Lectura correcta ", CadenaDatos
51         return CadenaDatos
52
53     def Cerrar(self):
54         if self.conexion:
55             self.Blue.close()
56             print "Dispositivo cerrado"
57
58     def __init__(self):
59         global puerto
60         puerta = getoutput("rfcomm")
61         try:
62             puertas = puerta.split("\n")
63             longitudDeLaPuerta = len(puertas)

```

```

64         if longitudDeLaPuerta == 1:
65             connect = puertas[0]
66             TestEstado = connect.split(" ")
67             estado = TestEstado[6]
68             if estado == "closed":
69                 print "Revise su conexión con el dispositivo en bluemanager e intentelo de nuevo"
70                 if estado == "connected":
71                     #         puerto = puertas[0]
72                     #         print puerta
73                     puertoa = TestEstado[0]
74                     puertop = puertoa.split(":")
75                     puerto = "/dev/"+str(puertop[0])
76                     print puerto
77
78                 if longitudDeLaPuerta > 1:
79                     i = 1
80                     while(i<longitudDeLaPuerta):
81                         connect = puertas[i]
82                         TestEstado = connect.split(" ")
83                         estado = TestEstado[6]
84                         if estado == "closed":
85                             print "Revise su conexión con el dispositivo en bluemanager e intentelo de nuevo"
86                             i = i+1
87                             if estado == "connected":
88                                 #         puerto = puertas[0]
89                                 #         print puerta
90                                 puertoa = TestEstado[0]
91                                 puertop = puertoa.split(":")
92                                 puerto = "/dev/"+str(puertop[0])
93                                 i = longitudDeLaPuerta +1
94
95
96         except:
97             print "Conecte su dispositivo bluetooth con bluemanager"
98
99
100     def __del__(self):
101         print "destructor de la clase ArduinoBluetooth"
102
103     #Arduino = ArduinoBluetooth()
104     #Arduino.obtenerCorrienteEnchufe("HC-05")
105     #Arduino.Escribir('a')
106     #print Arduino.LeerNUMERO()

```