

```

#!/usr/bin/env python
# -*- coding:utf-8 -*-
# clavier_direction.py
from Tkinter import *
from nanpy import Arduino
from nanpy import SerialManager
from nanpy.pwm import ArduinoPwmPin
from nanpy.arduinotree import ArduinoTree
from time import sleep
# serial_manager.connect('/dev/ttyACM0') # serial connection to Arduino ne ma
rche pas
connection = SerialManager()
# Arduino = Arduino(connection=connection)
EN = 9
IN1 = 10 # sens avant sens à 1
IN2 = 11 # sens arrière sens à 0

# Moteur sur Arduino Pin 9,10,11 avec MLI (with PWM)
# http://python.mesexemples.com/plus-de-python/gui/evenements/gerer-les-evenements-1
ies-aux-touches-de-direction-du-clavier/
# Maggi

class Evenement(Frame):
    def __init__(self, parent=None):
        Frame.__init__( self )
        self.master.title("Touche de direction")
        self.master.geometry("150x100") #La Taille du fenetre
        self.pack(expand=YES, fill=BOTH)
        lbl=Label(self, text="Appuyez sur une touche de direction")
        lbl["bg"]="blue"
        lbl["font"]=('courier', 20, 'bold')
        lbl.pack(expand=YES, fill=BOTH)
        lbl.focus()
        self.valeur_direction = 0
        self.valeur_propulsion = 0
        self.sens=0
        # Lier un événement aux quatre touches de direction du clavier
        lbl.bind("<Left>", self.gauche) # direction gauche '<--'
        lbl.bind("<Right>", self.droite) # direction droite '-->'
        lbl.bind("<Up>", self.avant) # direction avant
        lbl.bind("<Down>", self.arriere) # direction arriere
        lbl.bind("<Escape>", self.quitter) # arrêt
        lbl.bind("<space>", self.arret) # arrêt

    def droite(self,event):
        print "Vous avez appuyé sur une touche de direction droite"
        self.valeur_direction += 1
        print "dir:",self.valeur_direction

    def gauche(self,event):
        #def gauche(self,event):
        print "Vous avez appuyé sur une touche de direction gauche"
        self.valeur_direction -= 1
        print "dir:",self.valeur_direction

    def arriere(self,event):
        print "Vous avez appuyé sur une touche de direction recul"
        self.valeur_propulsion -= 1
        print "avance:",self.valeur_propulsion
        self.traction()

    def avant(self,event):
        print "Vous avez appuyé sur une touche de direction avant"
        self.valeur_propulsion += 1
        print "avance:",self.valeur_propulsion
        self.traction()

    def quitter(self,event):
        print "Vous avez appuyé sur la touche arrêt"
        self.quit()

    def arret(self,event):
        print "Vous avez appuyé sur la touche arrêt"
        self.parar()

    def parar(self):
        Arduino.digitalWrite(IN1,Arduino.LOW)
        Arduino.digitalWrite(IN2,Arduino.LOW)
        Arduino.digitalWrite(EN,Arduino.LOW)

```

```
        self.valeur_propulsion = 0
        self.valeur_direction = 0
        sleep(1.5)
def traction(self):
    if self.valeur_propulsion > 0 :
        # self.valeur_propulsion %= 50
        val=255-self.valeur_propulsion
        if self.sens == 0 : # changement de sens
            Arduino.digitalWrite(IN1,Arduino.HIGH)
            Arduino.digitalWrite(IN2,Arduino.HIGH)
            Arduino.digitalWrite(EN,Arduino.LOW)
            sleep(1.5)
            self.sens = 1
        else :
            Arduino.digitalWrite(EN,Arduino.HIGH)
            Arduino.digitalWrite(IN2,Arduino.HIGH)
            Arduino.analogWrite(IN1, val)
    else :
        val = 255+self.valeur_propulsion
        #self.valeur_propulsion = -val
        if self.sens == 1 : # changement de sens
            Arduino.digitalWrite(IN1,Arduino.LOW)
            Arduino.digitalWrite(IN2,Arduino.LOW)
            Arduino.digitalWrite(EN,Arduino.LOW)
            sleep(1.5)
            self.sens = 0
        else :
            Arduino.digitalWrite(IN1,Arduino.HIGH)
            Arduino.digitalWrite(EN,Arduino.HIGH)
            Arduino.analogWrite(IN2,val)
    print "sens",self.sens
def main():
    #Fonction principale
    a = ArduinoTree(connection=connection)
    Arduino.pinMode(EN, Arduino.OUTPUT)
    Arduino.pinMode(IN1, Arduino.OUTPUT)
    Arduino.pinMode(IN2, Arduino.OUTPUT)
    avant = a.pin.get(IN1)
    # print ('PWM frequencies_available: %s' % avant.pwm.frequencies_available)
    avant.pwm.frequency = 3906.25
    Evenement().mainloop()
    Evenement().destroy()
if __name__=="__main__":
    main()
```