#!/usr/bin/env python

# -\*- coding: utf8 -\*-

import RPi.GPIO as GPIO

import MFRC522

import signal

continue\_reading = True

import time

OFFSE\_DUTY = 0.5 #define pulse offset of servo

SERVO\_MIN\_DUTY = 2.5+OFFSE\_DUTY #define pulse duty cycle for minimum angle of servo

SERVO\_MAX\_DUTY = 12.5+OFFSE\_DUTY #define pulse duty cycle for maximum angle of servo

servoPin = 12

def map( value, fromLow, fromHigh, toLow, toHigh):

return (toHigh-toLow)\*(value-fromLow) / (fromHigh-fromLow) + toLow

def setup():

global p

GPIO.setmode(GPIO.BOARD) # Numbers GPIOs by physical location

GPIO.setup(servoPin, GPIO.OUT) # Set servoPin's mode is output

GPIO.output(servoPin, GPIO.LOW) # Set servoPin to low

p = GPIO.PWM(servoPin, 50) # set Frequece to 50Hz

p.start(0)

def open():

while True:

for dc in range(0, 181, 1): #make servo rotate from 0 to 180 deg

servoWrite(dc) # Write to servo

time.sleep(0.001)

time.sleep(0.5)

def close():

while True:

for dc in range(180, -1, -1): #make servo rotate from 180 to 0 deg

servoWrite(dc)

time.sleep(0.001)

time.sleep(0.5)

def servoWrite(angle): # make the servo rotate to specific angle (0-180 degrees)

if(angle<0):

angle = 0

elif(angle > 180):

angle = 180

p.ChangeDutyCycle(map(angle,0,180,SERVO\_MIN\_DUTY,SERVO\_MAX\_DUTY))#map the angle to duty cycle and output it

def destroy():

p.stop()

GPIO.cleanup()

if \_\_name\_\_ == '\_\_main\_\_': #Program start from here

print 'Program is starting...'

setup()

try:

open()

close()

except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the child program destroy() will be executed.

destroy()