from time import sleep

import machine

import math

from machine import PWM

from machine import Pin

from machine import ADC

from picobricks import WS2812

import time

from machine import I2C

from picobricks import SSD1306\_I2C

pin\_button = machine.Pin(10, machine.Pin.IN)

import math

pot = machine.ADC(26)

motor\_1 = PWM(Pin(21))

motor\_1.freq(50)

motor\_1.duty\_u16(0)

import random

ws2812 = WS2812(6, brightness = 1)

i2c = I2C(0, scl=Pin(5), sda=Pin(4), freq=200000)

oled = SSD1306\_I2C(128, 64, i2c, addr=0x3c)

def Countdown():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while (pin\_button.value()) == (0) and a >= (4):

 for count in range(sumofvariables):

 while sumofvariables == (1):

 motor\_1.duty\_u16(round(round( pot.read\_u16() - 0 ) \* ( 100 - 0 ) / ( 65535 - 0 ) + 0) \* 650)

 ws2812.pixels\_fill(((random.randint(1, 255)), (random.randint(1, 255)), (random.randint(1, 255))))

 ws2812.pixels\_show()

 time.sleep((1))

 oled.fill(0)

 oled.text("{}".format("~~~~~~~~~~~~~~~~"), 0, 10)

 oled.text("{}".format("~~~~~~~~~~~~~~~~"), 0, 40)

 oled.text("{}".format("Merry Christmas"), 5, 25)

 oled.fill(0)

 oled.text("{}".format("Days"), 0, 5)

 oled.text("{}".format("Hours"), 0, 15)

 oled.text("{}".format("Minutes"), 0, 25)

 oled.text("{}".format("Seconds"), 0, 35)

 remday = sumofvariables % 86400

 subday = sumofvariables - (sumofvariables % 86400)

 Day = subday / (86400)

 remhour = remday % 3600

 subhour = remday - remhour

 hour = subhour / (3600)

 remminute = remhour % 60

 subminute = remhour - remminute

 minute = subminute / (60)

 days()

 hours()

 minutes()

 seconds()

 sumofvariables = sumofvariables - (1)

 speed = round(round( pot.read\_u16() - 0 ) \* ( 100 - 0 ) / ( 65535 - 0 ) + 0)

 motor\_1.duty\_u16(speed \* 650)

 oled.text("{}".format(speed), 0, 45)

 oled.show()

 ws2812.pixels\_fill(((random.randint(1, 255)), (random.randint(1, 255)), (random.randint(1, 255))))

 ws2812.pixels\_show()

 time.sleep((0.9))

def dayset():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while (pin\_button.value()) == (0) and a == (0):

 dayvalue = round(round( pot.read\_u16() - 0 ) \* ( 365 - -1 ) / ( 65535 - 0 ) + -1)

 oled.fill(0)

 oled.text("{}".format("Enter Days"), 24, 5)

 oled.text("{}".format("Left"), 50, 15)

 oled.text("{}".format("---------------------"), 0, 25)

 oled.text("{}".format(dayvalue), 55, 40)

 oled.show()

 a += 1

def MinuteSet():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while (pin\_button.value()) == (0) and a == (2):

 minutevalue = round(round( pot.read\_u16() - 0 ) \* ( 59 - 0 ) / ( 65535 - 0 ) + 0)

 oled.fill(0)

 oled.text("{}".format("Enter Minutes"), 24, 5)

 oled.text("{}".format("Left"), 50, 15)

 oled.text("{}".format("---------------------"), 0, 25)

 oled.text("{}".format(minutevalue), 55, 40)

 oled.show()

 a += 1

def Hour\_Set():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while (pin\_button.value()) == (0) and a == (1):

 hourvalue = round(round( pot.read\_u16() - 0 ) \* ( 23 - 0 ) / ( 65535 - 0 ) + 0)

 oled.fill(0)

 oled.text("{}".format("Enter hours"), 24, 5)

 oled.text("{}".format("Left"), 50, 15)

 oled.text("{}".format("---------------------"), 0, 25)

 oled.text("{}".format(hourvalue), 55, 40)

 oled.show()

 a += 1

def days():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while Day <= (365) and Day >= (100):

 oled.text("{}".format(Day), 105, 5)

 oled.show()

 break

 while Day <= (99) and Day >= (10):

 oled.text("{}".format(Day), 110, 5)

 oled.show()

 break

 while Day <= (9) and Day >= (0):

 oled.text("{}".format(Day), 120, 5)

 oled.show()

 break

def secondset():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while (pin\_button.value()) == (0) and a == (3):

 secondvalue = round(round( pot.read\_u16() - 0 ) \* ( 59 - 0 ) / ( 65535 - 0 ) + 0)

 oled.fill(0)

 oled.text("{}".format("Enter Seconds"), 24, 5)

 oled.text("{}".format("Left"), 50, 15)

 oled.text("{}".format("---------------------"), 0, 25)

 oled.text("{}".format(secondvalue), 55, 40)

 oled.show()

 a += 1

def dataenter():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 a = 0

 dayset()

 time.sleep((1))

 Hour\_Set()

 time.sleep((1))

 MinuteSet()

 time.sleep((1))

 secondset()

 secondvalue = secondvalue - (6)

 time.sleep((1))

 daytosec = (86400) \* dayvalue

 time.sleep((1))

 hourtosec = (3600) \* hourvalue

 time.sleep((1))

 minutetosec = (60) \* minutevalue

 time.sleep((1))

 sumofvariables = (hourtosec + minutetosec) + (daytosec + secondvalue)

 time.sleep((1))

def hours():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while hour <= (99) and hour >= (10):

 oled.text("{}".format(hourvalue), 110, 15)

 oled.show()

 break

 while hour <= (9) and hour >= (0):

 oled.text("{}".format(hour), 120, 15)

 oled.show()

 break

def minutes():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while minute <= (99) and minute >= (10):

 oled.text("{}".format(minute), 110, 25)

 oled.show()

 break

 while minute <= (9) and minute >= (0):

 oled.text("{}".format(minute), 120, 25)

 oled.show()

 break

def seconds():

 global a, dayvalue, minutevalue, hourvalue, secondvalue, sumofvariables, Day, minute, remminute, hour, remday, subday, daytosec, remhour, subhour, hourtosec, minutetosec, subminute, speed

 while remminute <= (99) and remminute >= (10):

 oled.text("{}".format(remminute), 110, 35)

 oled.show()

 break

 while remminute <= (9) and remminute >= (0):

 oled.text("{}".format(remminute), 120, 35)

 oled.show()

 break

dataenter()

while True:

 oled.fill(0)

 oled.text("{}".format("Days"), 0, 5)

 oled.text("{}".format("Hours"), 0, 15)

 oled.text("{}".format("Minutes"), 0, 25)

 oled.text("{}".format("Seconds"), 0, 35)

 oled.show()

 Countdown()