import RPi.GPIO as GPIO

class R2R\_DAC:

def \_\_init\_\_(self, gpio\_bits, dynamic\_range, verbose=False):

self.gpio\_bits = gpio\_bits

self.dynamic\_range = dynamic\_range

self.verbose = verbose

GPIO.setmode(GPIO.BCM)

GPIO.setup(self.gpio\_bits, GPIO.OUT, initial=0)

if self.verbose:

print(f"R2R DAC инициализирован на пинах: {self.gpio\_bits}")

print(f"Динамический диапазон: {self.dynamic\_range} В")

def deinit(self):

GPIO.output(self.gpio\_bits, 0)

GPIO.cleanup()

if self.verbose:

print("R2R DAC отключен, GPIO очищены")

def set\_number(self, number):

"""Устанавливает число на ЦАП в двоичном представлении"""

if number < 0 or number > 255:

if self.verbose:

print(f"Число {number} выходит за диапазон 0-255")

number = max(0, min(255, number))

binary = bin(number)[2:].zfill(8)

binary\_array = [int(bit) for bit in binary]

for i in range(len(self.gpio\_bits)):

GPIO.output(self.gpio\_bits[i], binary\_array[i])

if self.verbose:

print(f"Установлено число: {number}")

print(f"Двоичное представление: {binary}")

def set\_voltage(self, voltage):

"""Устанавливает напряжение на ЦАП"""

if voltage < 0 or voltage > self.dynamic\_range:

if self.verbose:

print(f"Напряжение {voltage:.2f} В выходит за динамический диапазон (0.00 - {self.dynamic\_range:.2f} В)")

print("Устанавливаем 0.0 В")

self.set\_number(0)

return

number = int(voltage / self.dynamic\_range \* 255)

self.set\_number(number)

if self.verbose:

print(f"Установлено напряжение: {voltage:.2f} В")

if \_\_name\_\_ == "\_\_main\_\_":

try:

dac = R2R\_DAC([16, 20, 21, 25, 26, 17, 27, 22], 3.183, True)

while True:

try:

voltage = float(input("Введите напряжение в Вольтах: "))

dac.set\_voltage(voltage)

except ValueError:

print("Вы ввели не число. Попробуйте ещё раз\n")

finally:

dac.deinit()