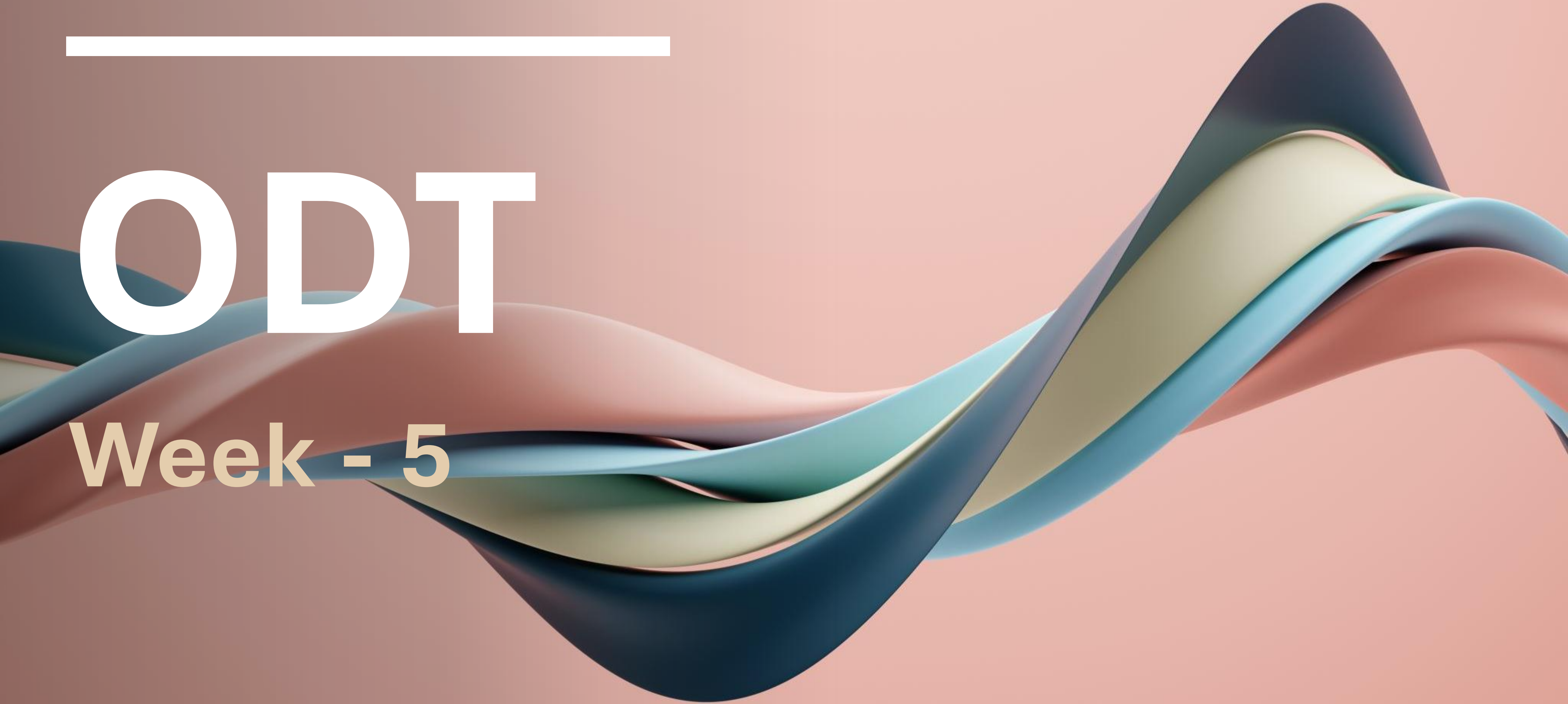
ODT

Week - 5



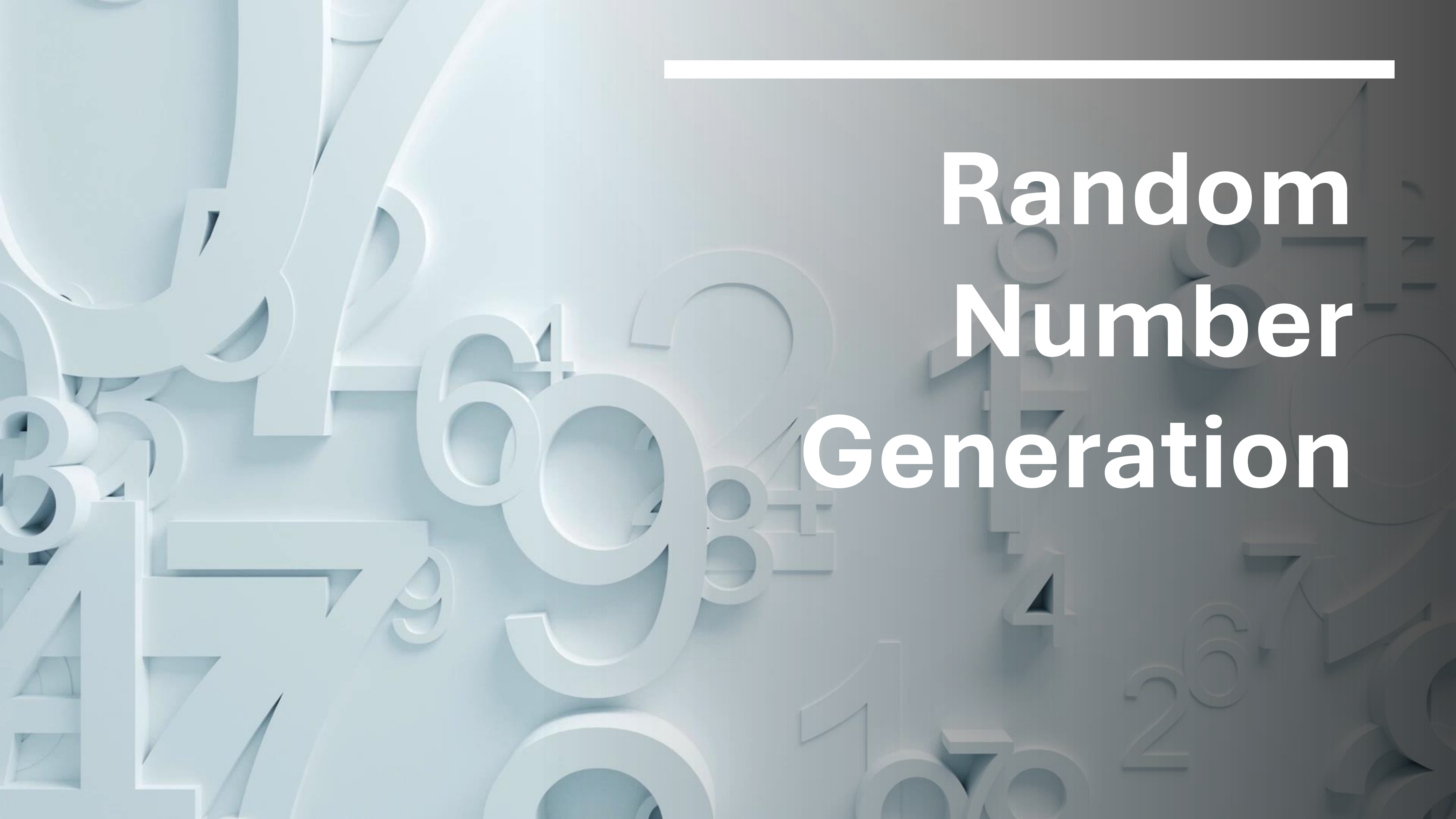
Empowering Designers to Embrace Technology...

Debugging

1. Syntax Error
2. Indentation Error
3. Attribute Error
4. Import Error

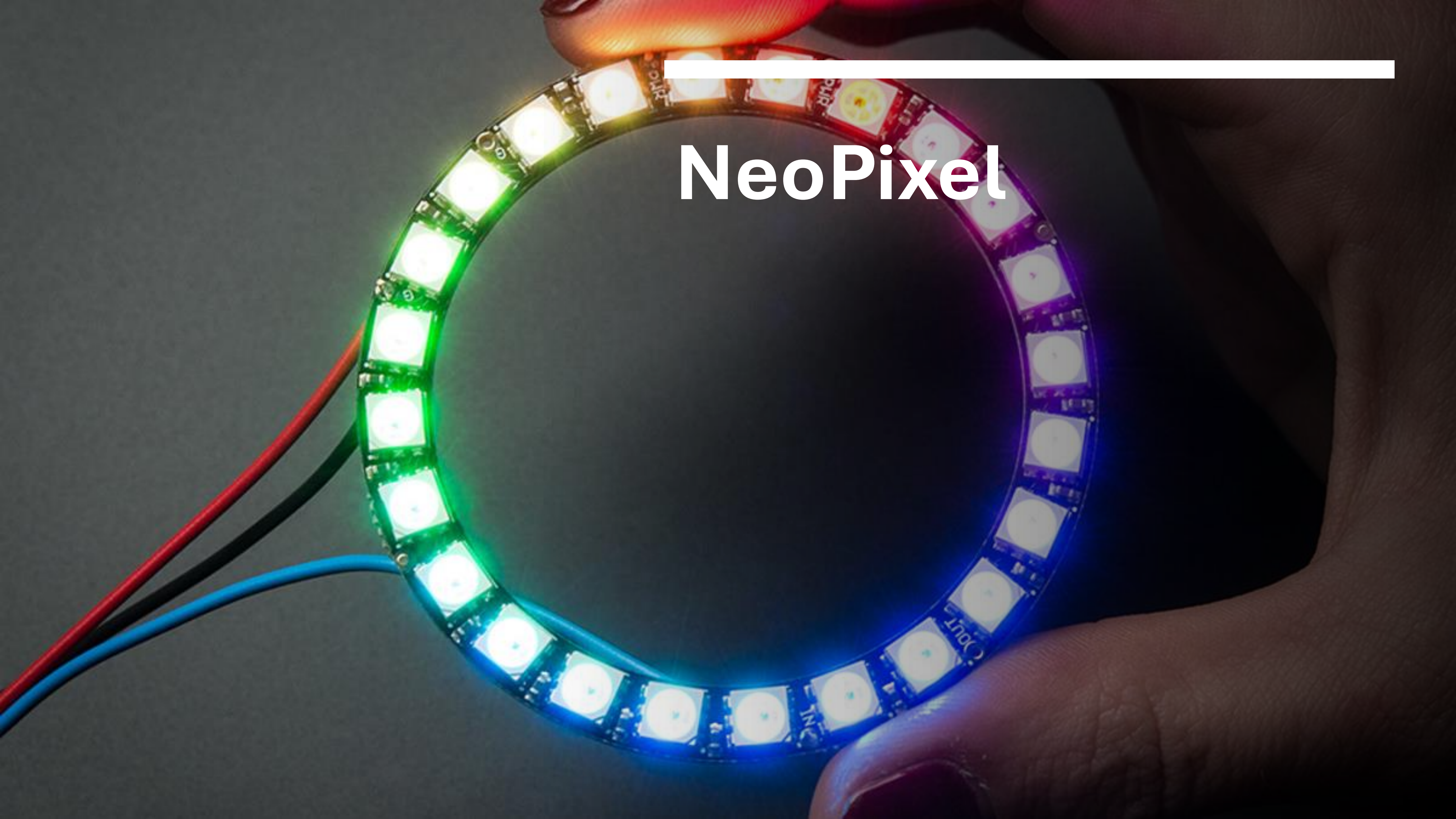
5. Logic Error

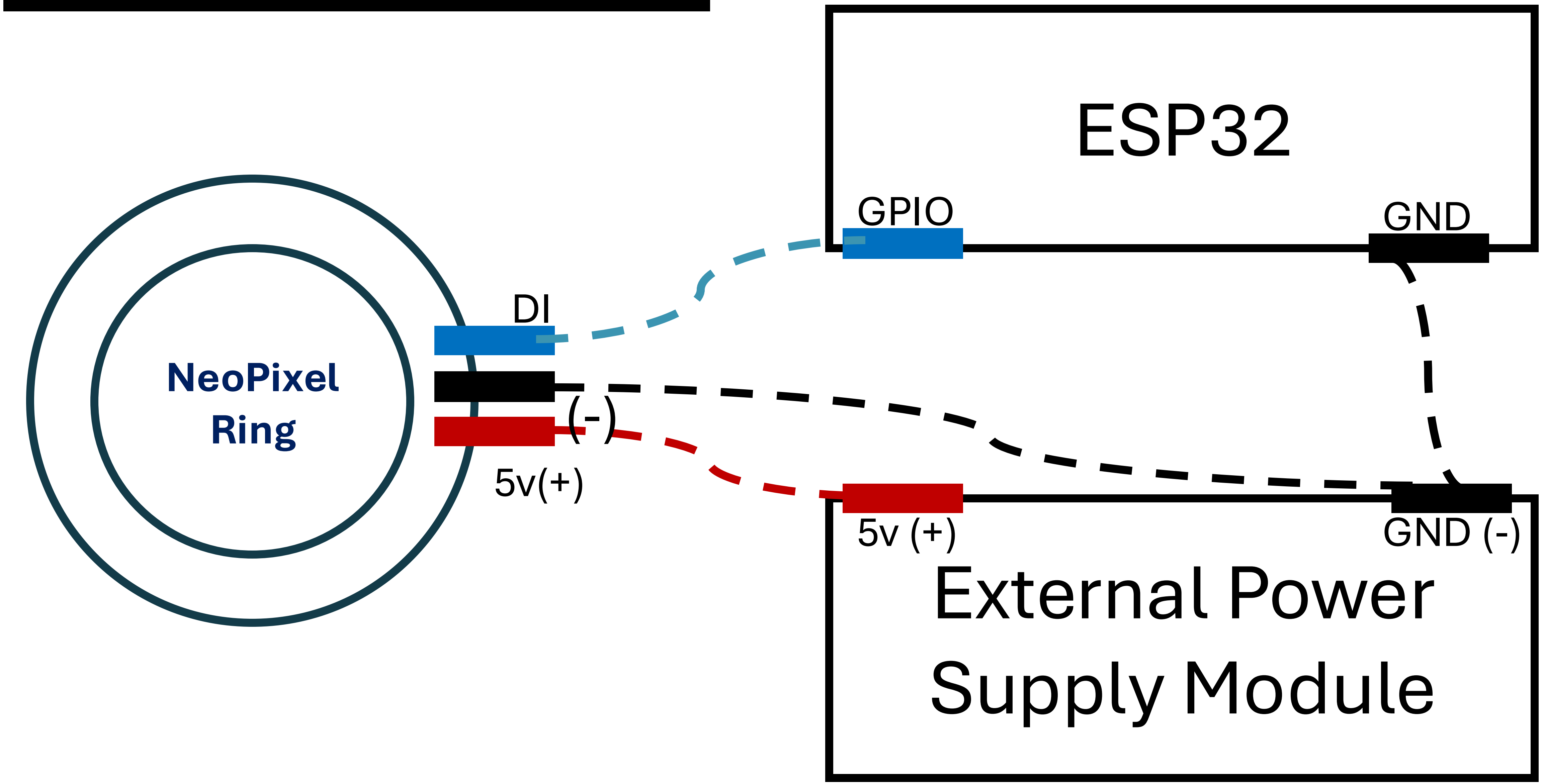
Circuit?



Random Number Generation

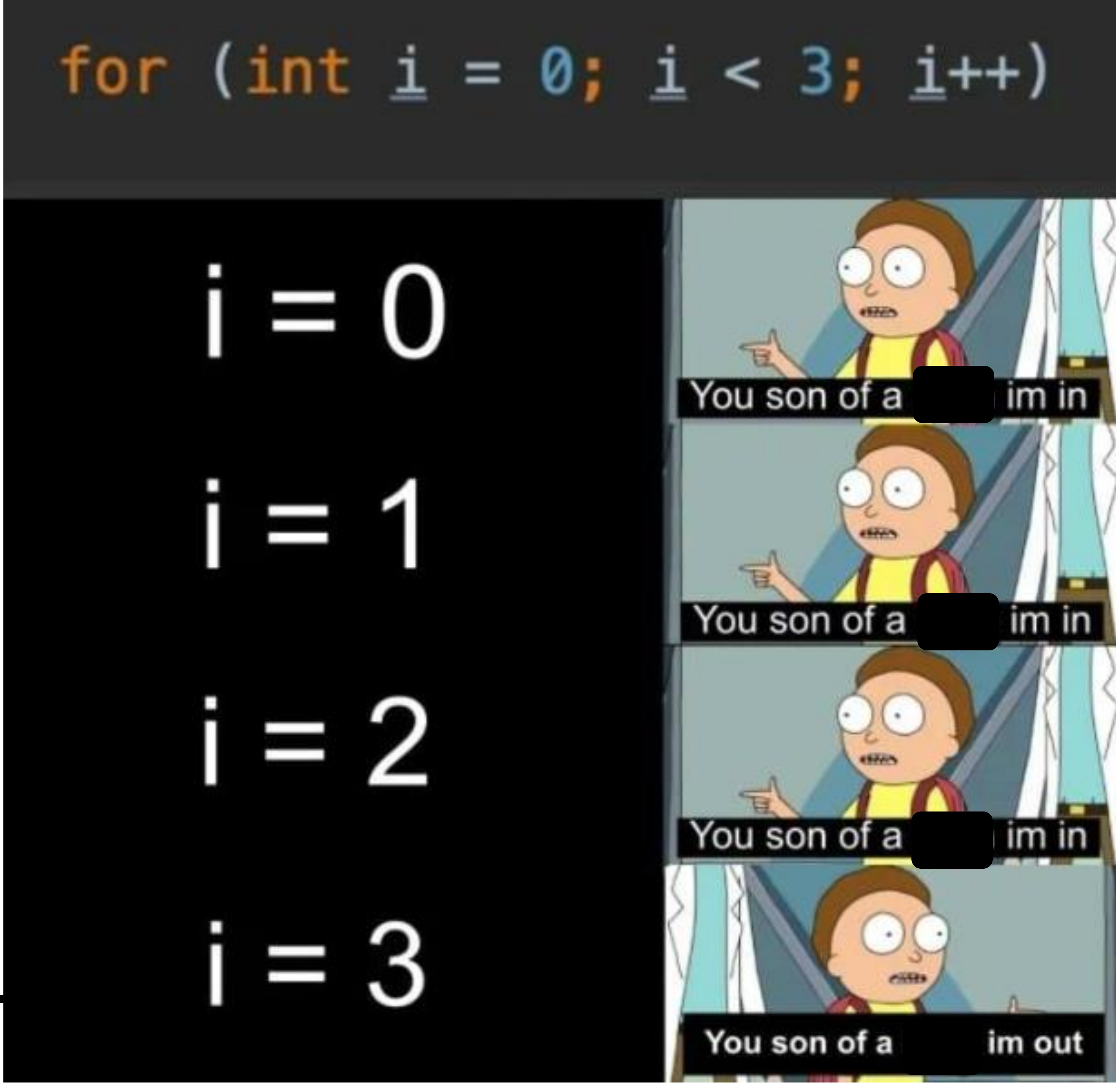
NeoPixel




ESP32
GPIO
GND
DI
NeoPixel
Ring
(-)
5v(+)
5v (+)
GND (-)
External Power
Supply Module

For

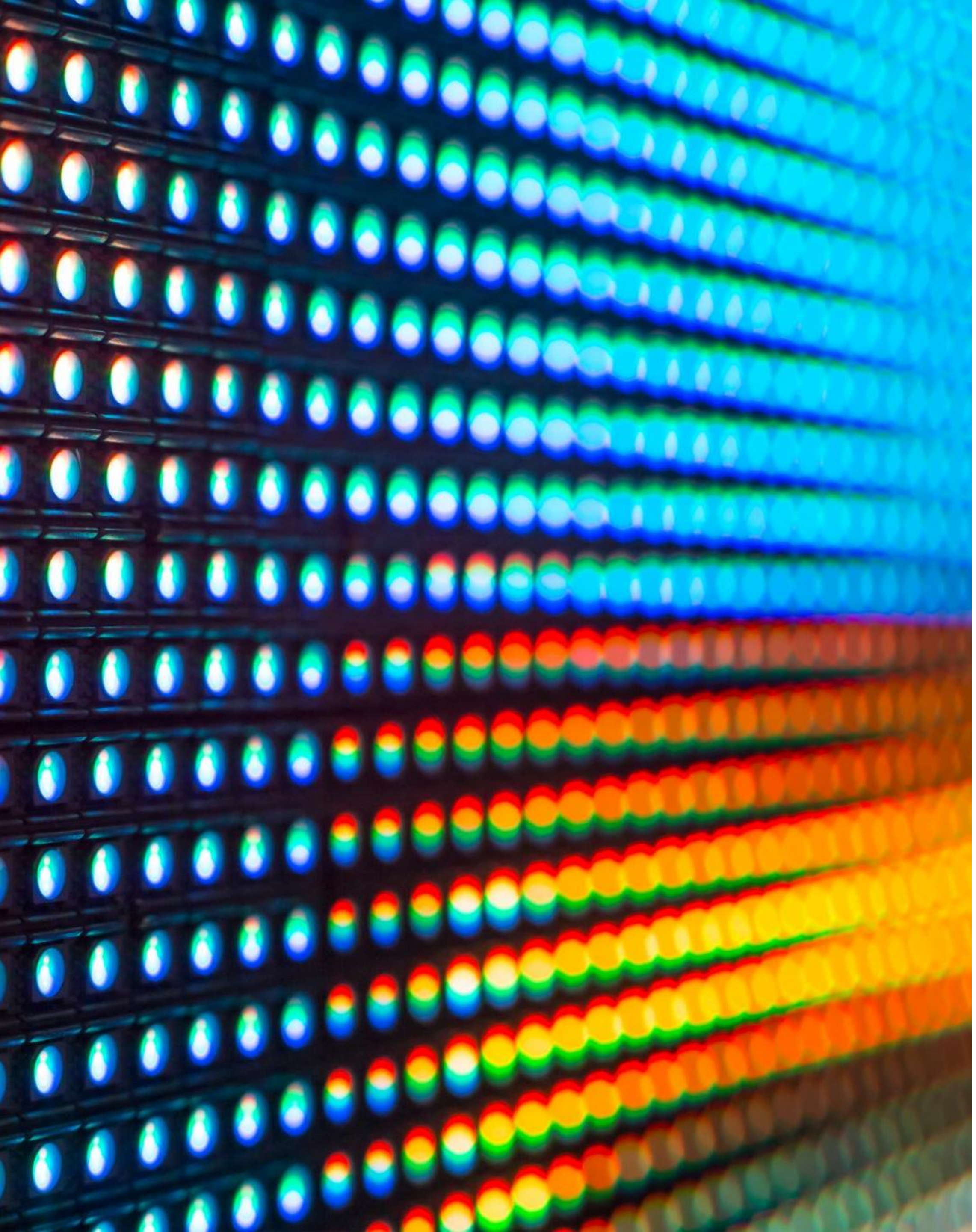
Loop



```
for (int i = 0; i < 3; i++)
```

i = 0
You son of a [redacted] im in
i = 1
You son of a [redacted] im in
i = 2
You son of a [redacted] im in
i = 3
You son of a [redacted] im out

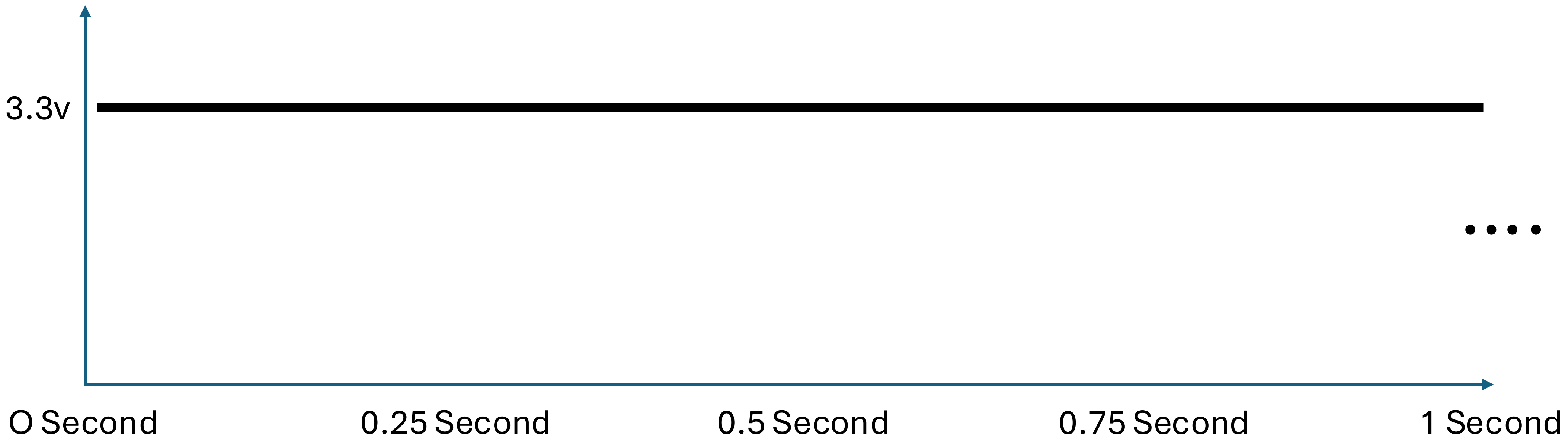
Storing the code in ESP32



LED with ESP32!

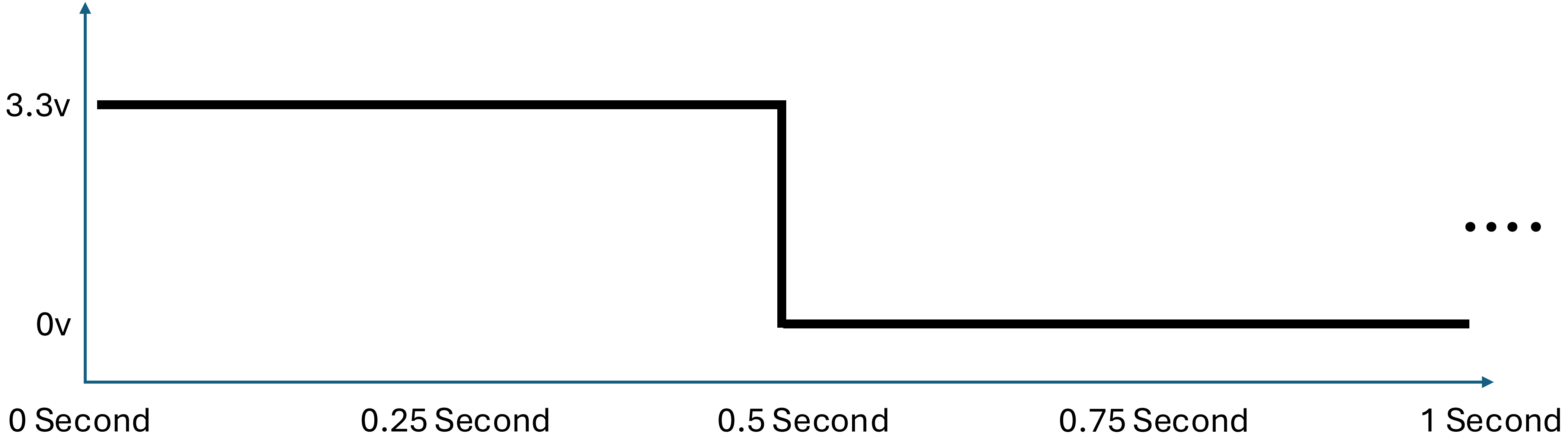
- Voltage across LED timeline

Voltage across LED (0 to 1 second) Case 1



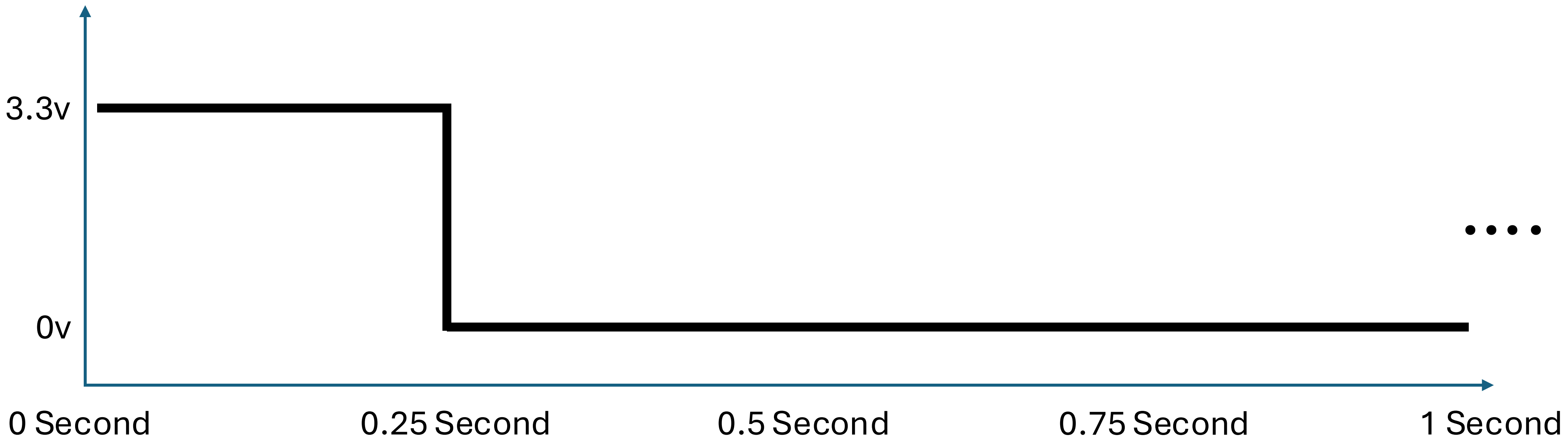
Time Period = 1 second

Voltage across LED (0 to 1 second) Case 2



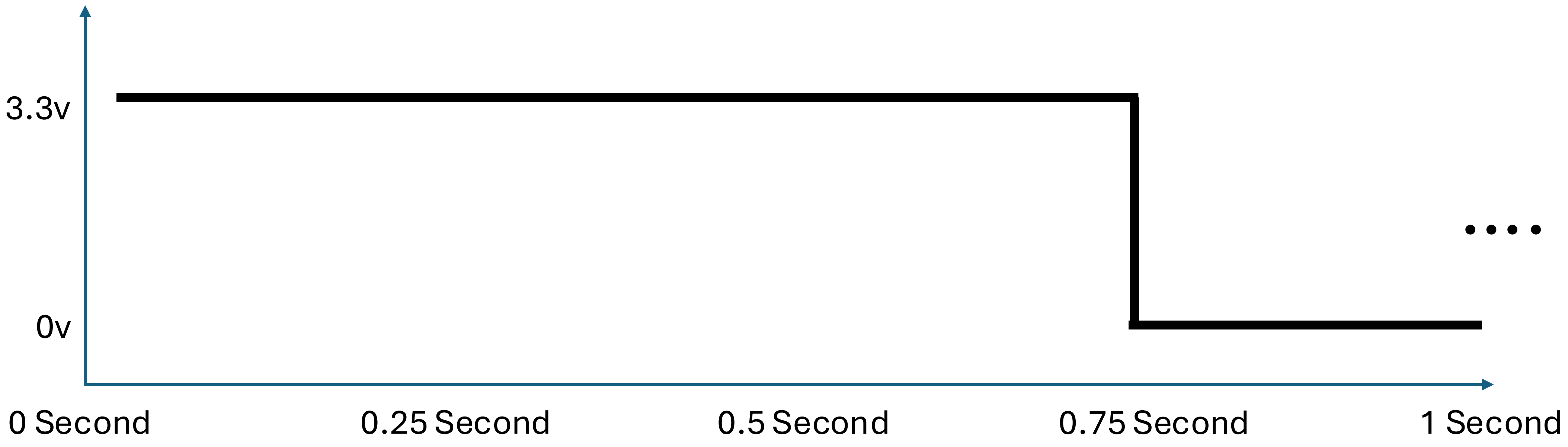
Time Period = 1 second

Voltage across LED (0 to 1 second) Case 3

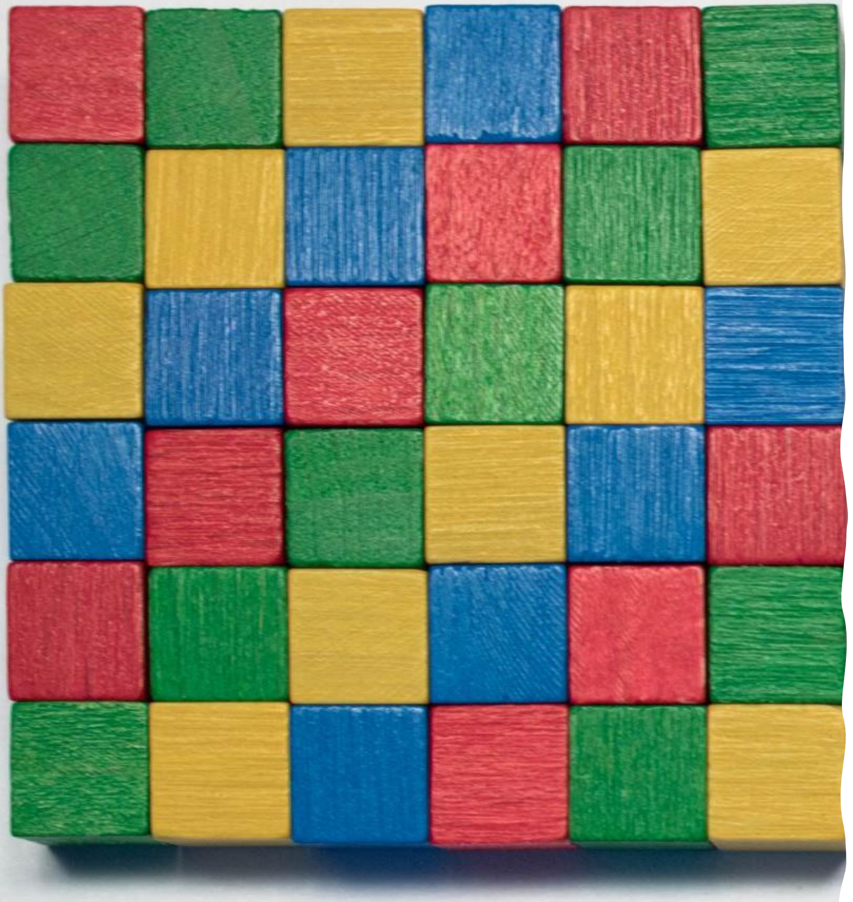


Time Period = 1 second

Voltage across LED (0 to 1 second) Case 4



Time Period = 1 second



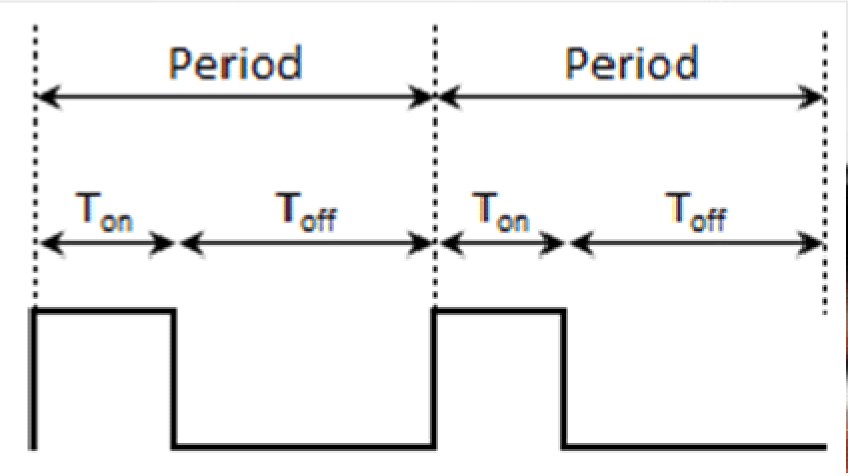
Case 1 to Case 4 ?

Pulse **W**idth **M**odulation

DutyCycle & Frequency

Duty Cycle

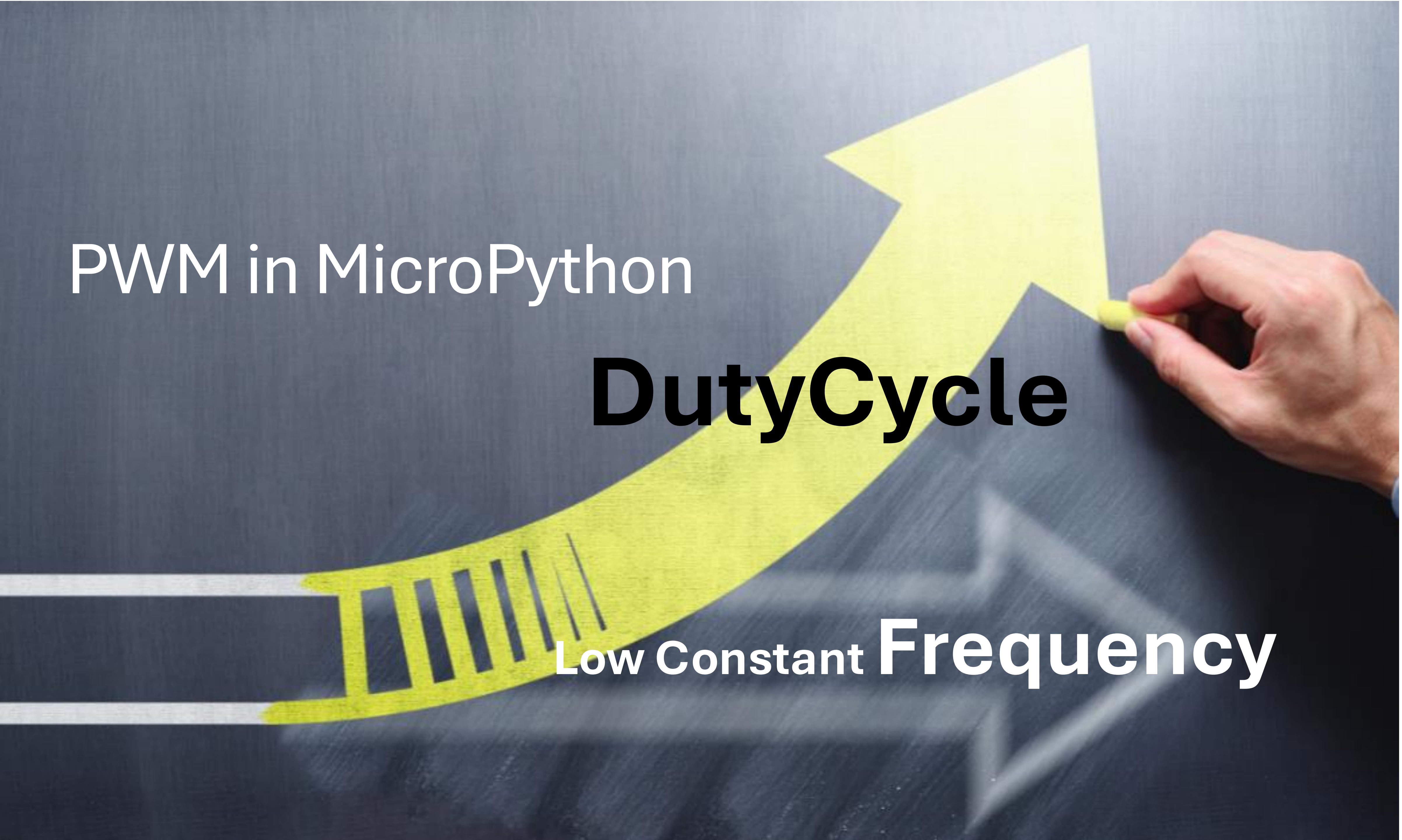
$$Duty\ Cycle\ \% = \frac{On\ time}{On\ time + Off\ time} X100$$



DutyCycle – PWM in MicroPython (ESP32)

DutyCycle	Value
100%	1023
75%	768
50%	512
25%	256
0%	0

PWM resolution of ESP32 = 10 bit (2^10 = 1024)



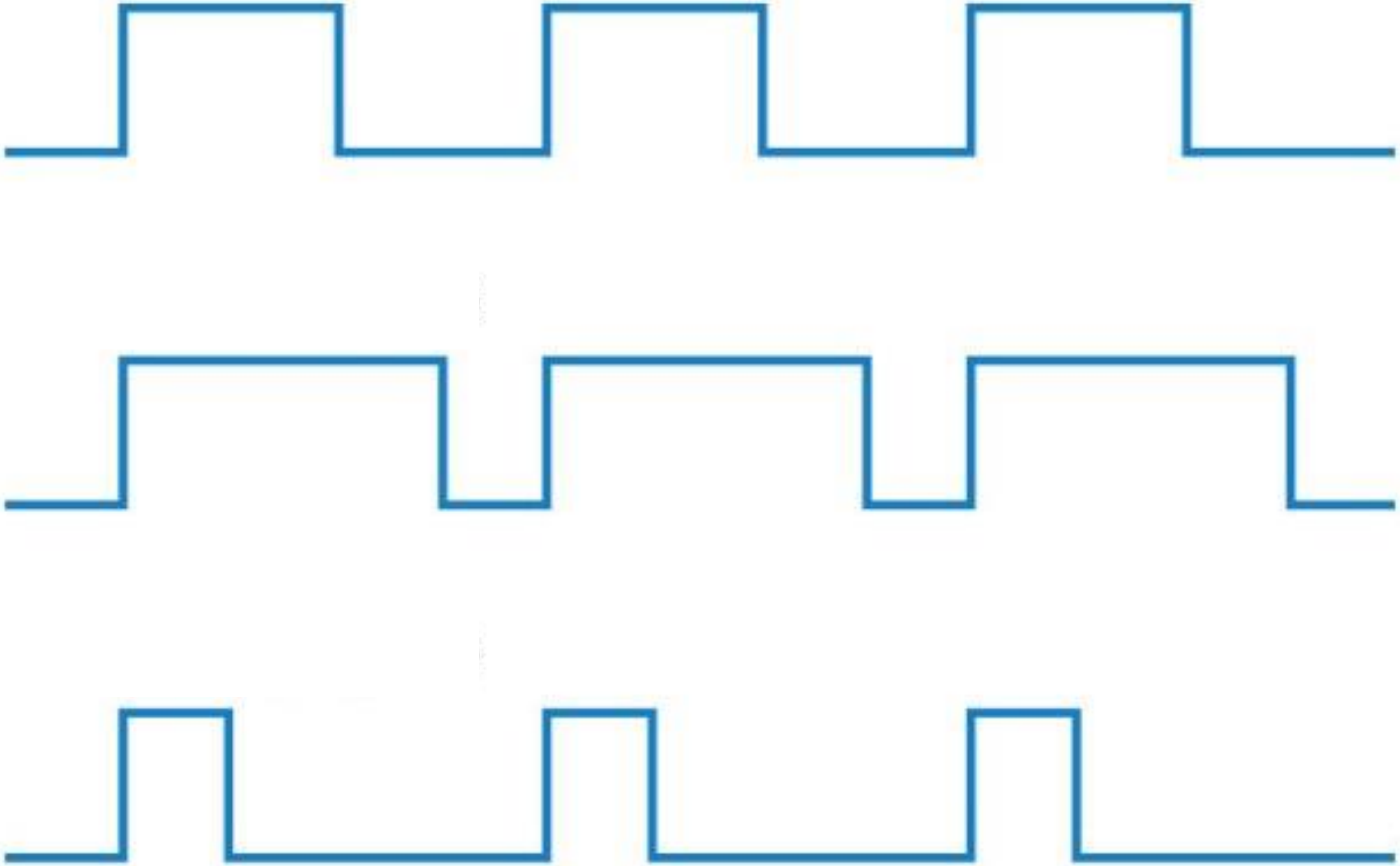
PWM in MicroPython

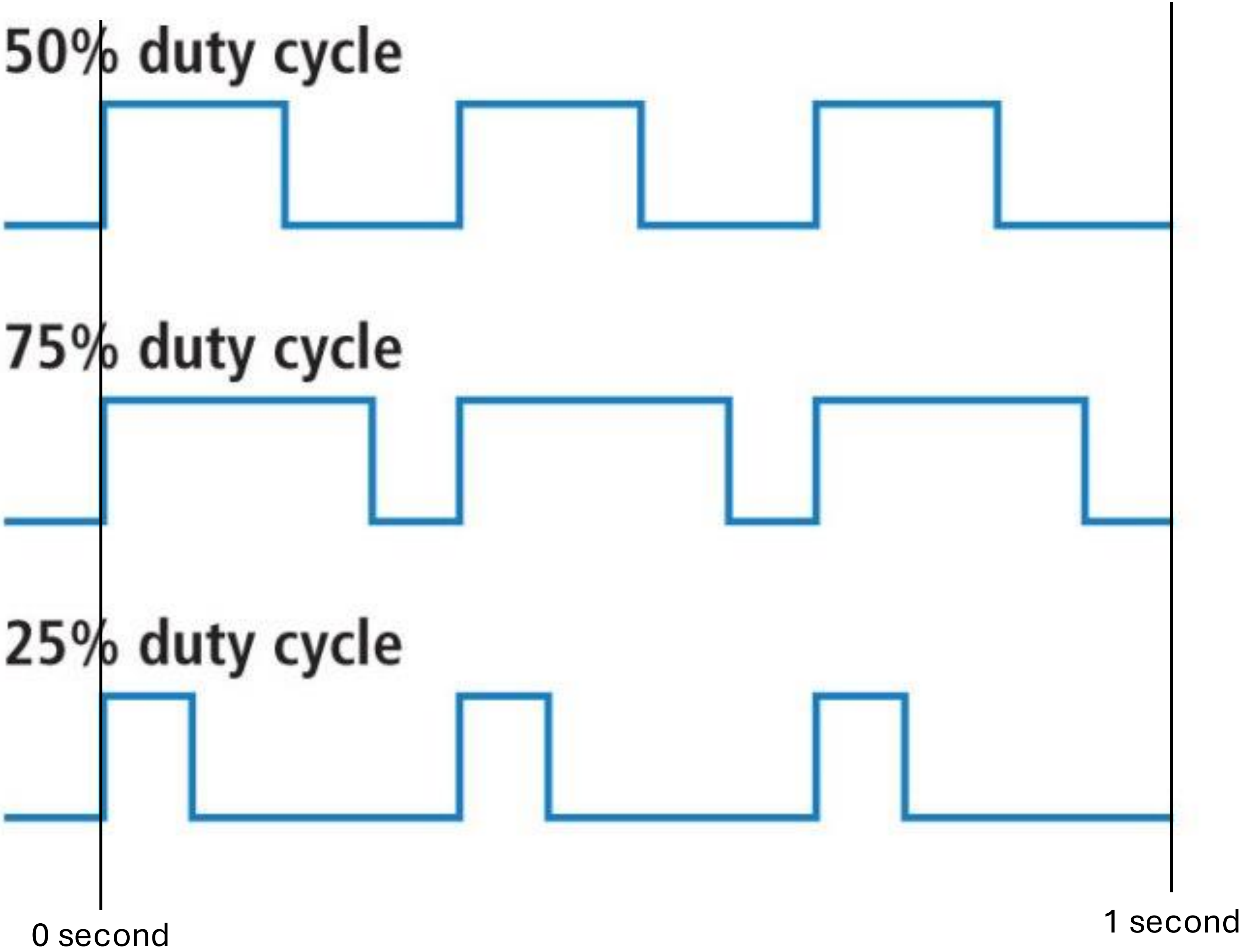
DutyCycle

Low Constant Frequency

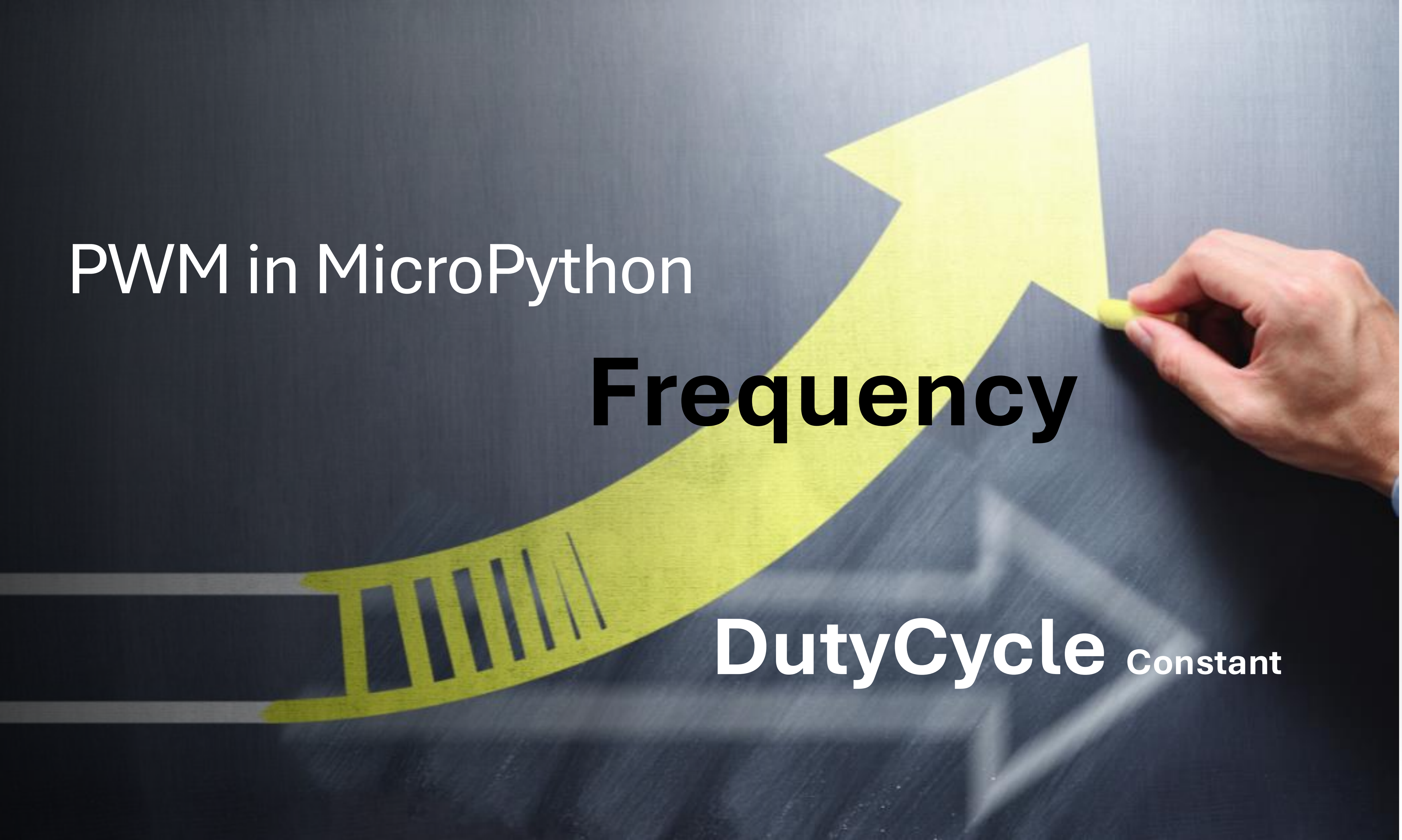
Frequency

No of repetitions
per unit time




50% duty cycle
75% duty cycle
25% duty cycle
0 second
1 second

?

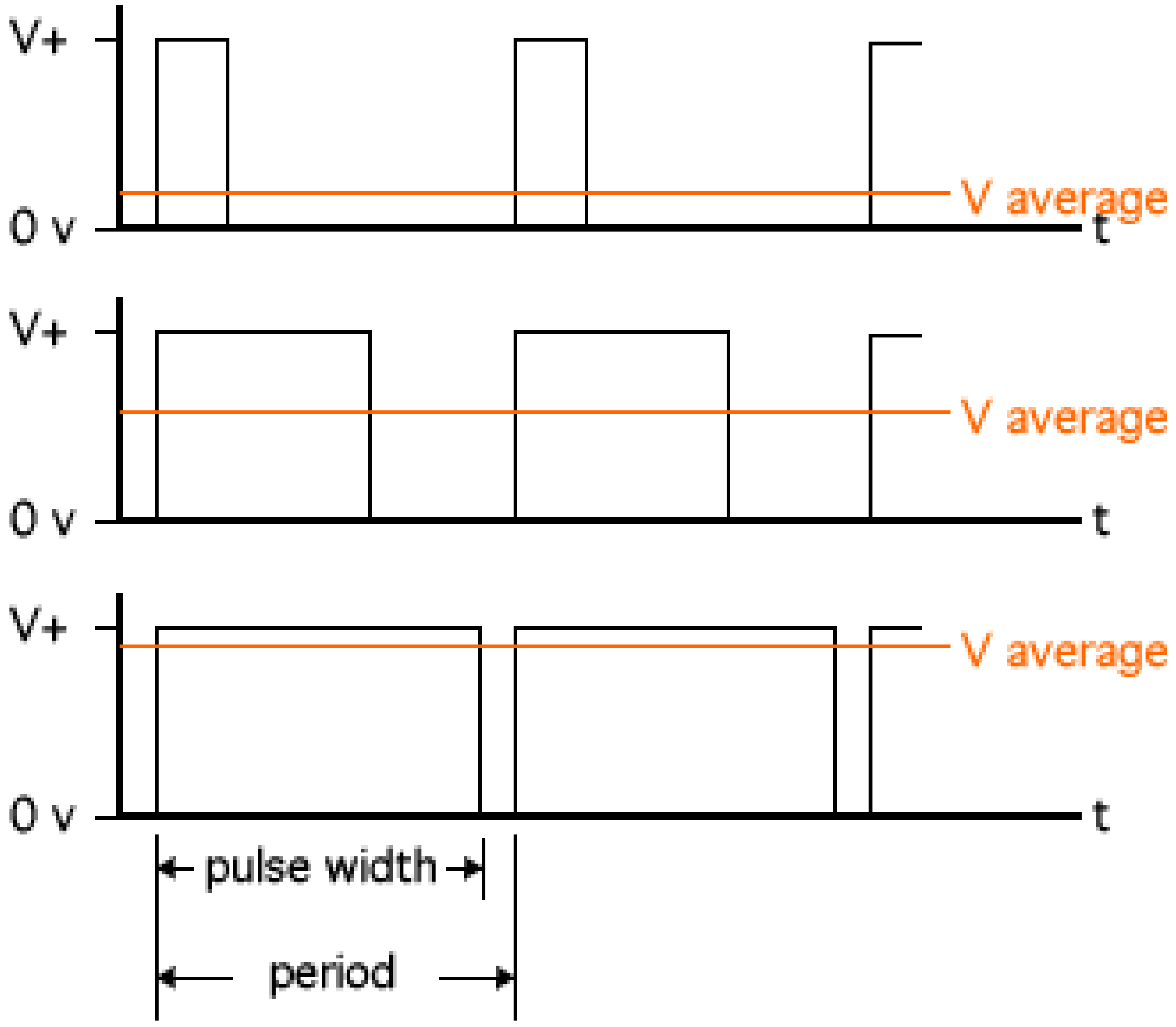


PWM in MicroPython

Frequency

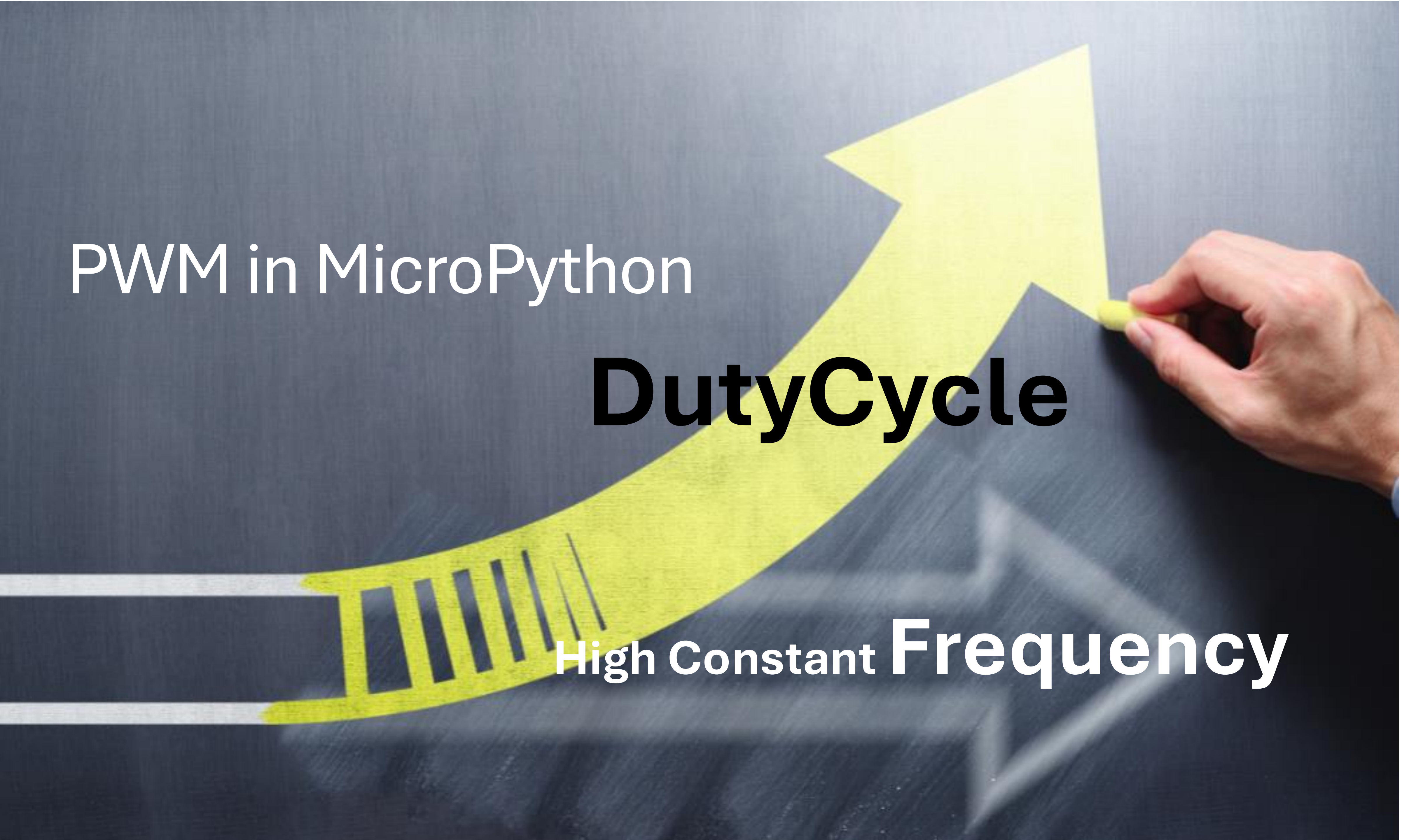
DutyCycle Constant

Duty cycle : averages the output voltage for Higher Frequencies :



$$V_{out} = \text{Duty Cycle} * V_{max}$$

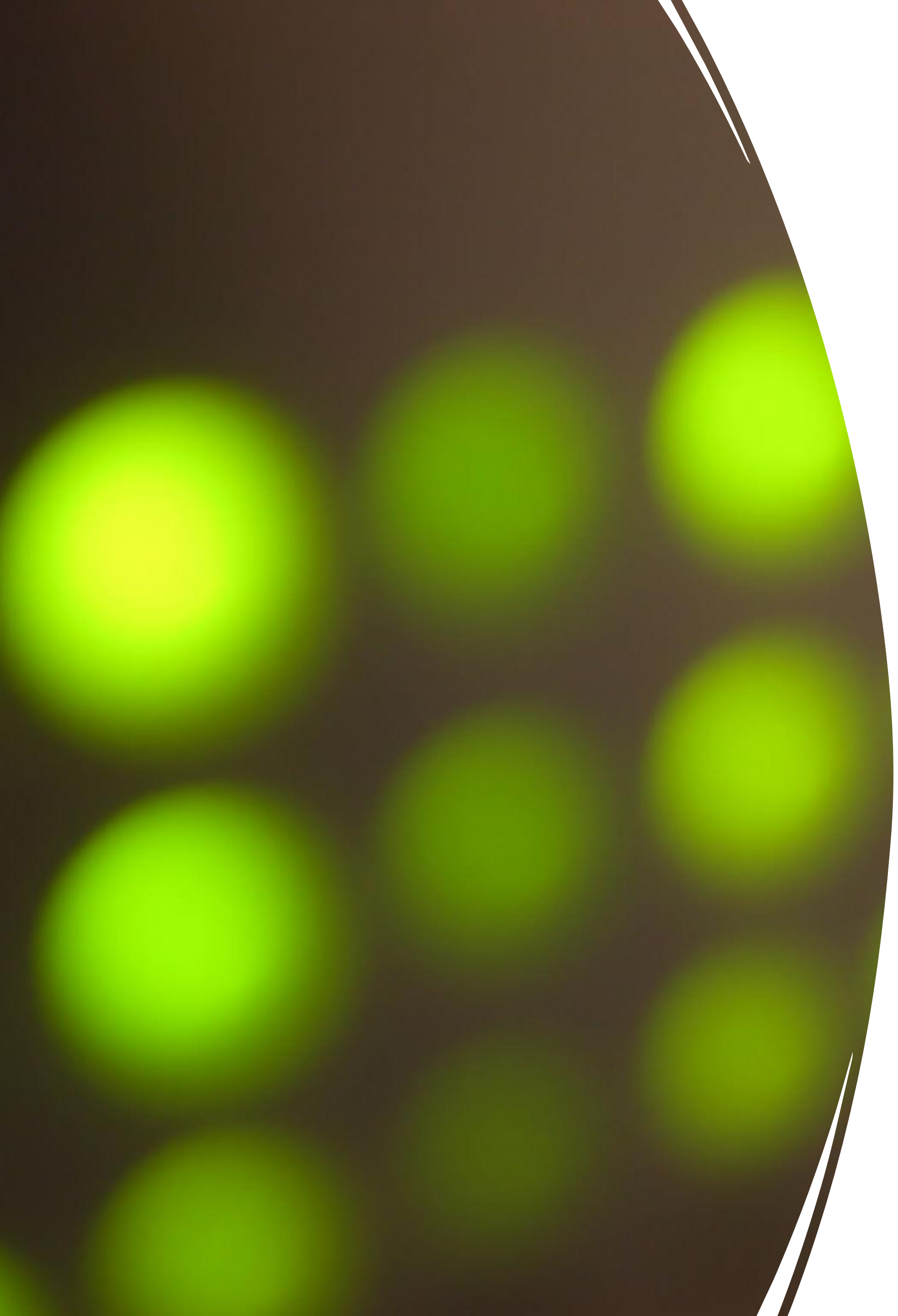
Duty cycle corresponds to brightness of the LED



PWM in MicroPython

DutyCycle

High Constant Frequency



Activity 1

- Gradually increase the LED brightness to its maximum level
- Gradually decrease the LED brightness until it is turned off
- Continuous smooth fading "in and out" LED brightness

Pules Width Modulation

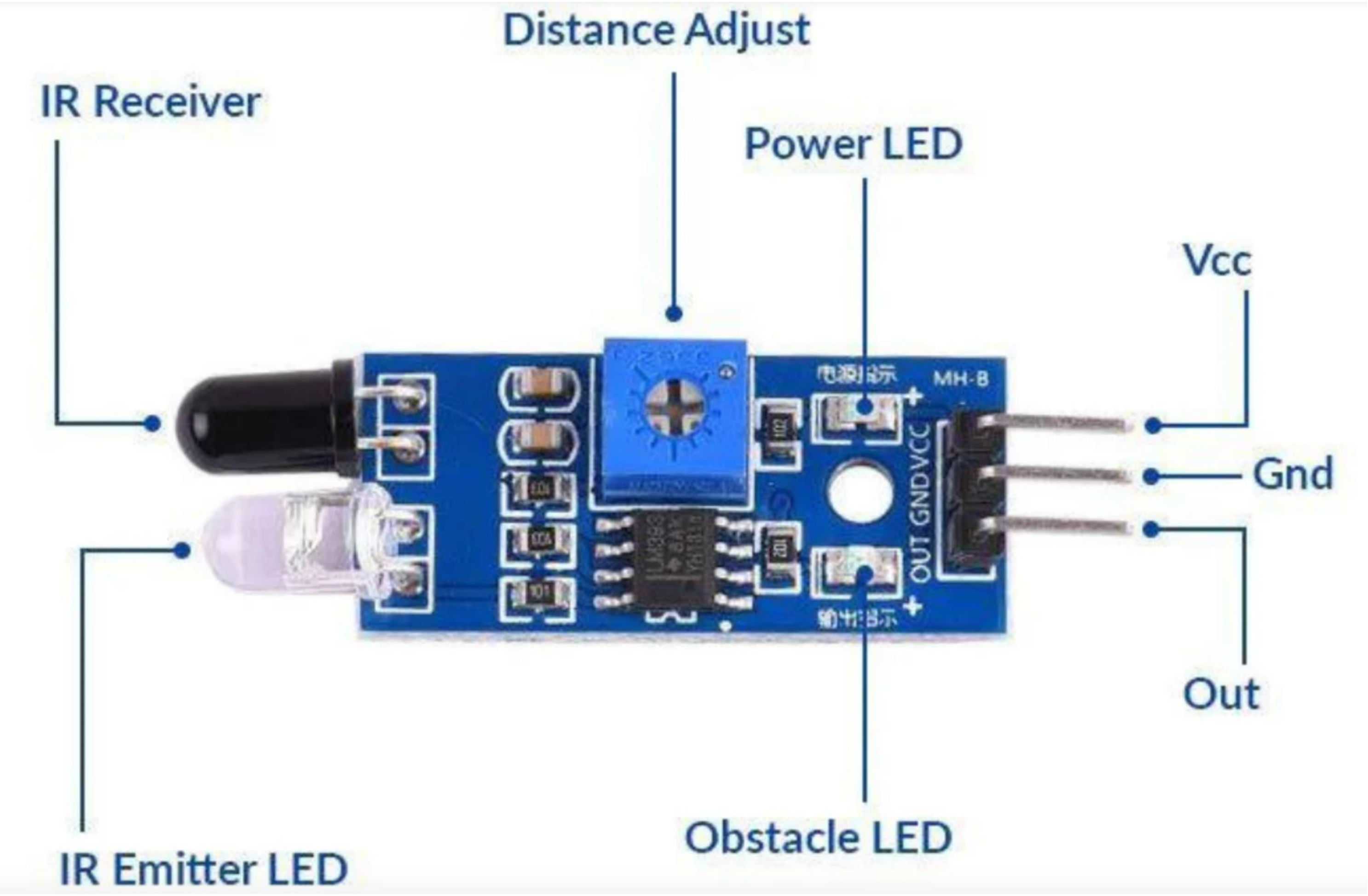
Duty Cycle

Frequency

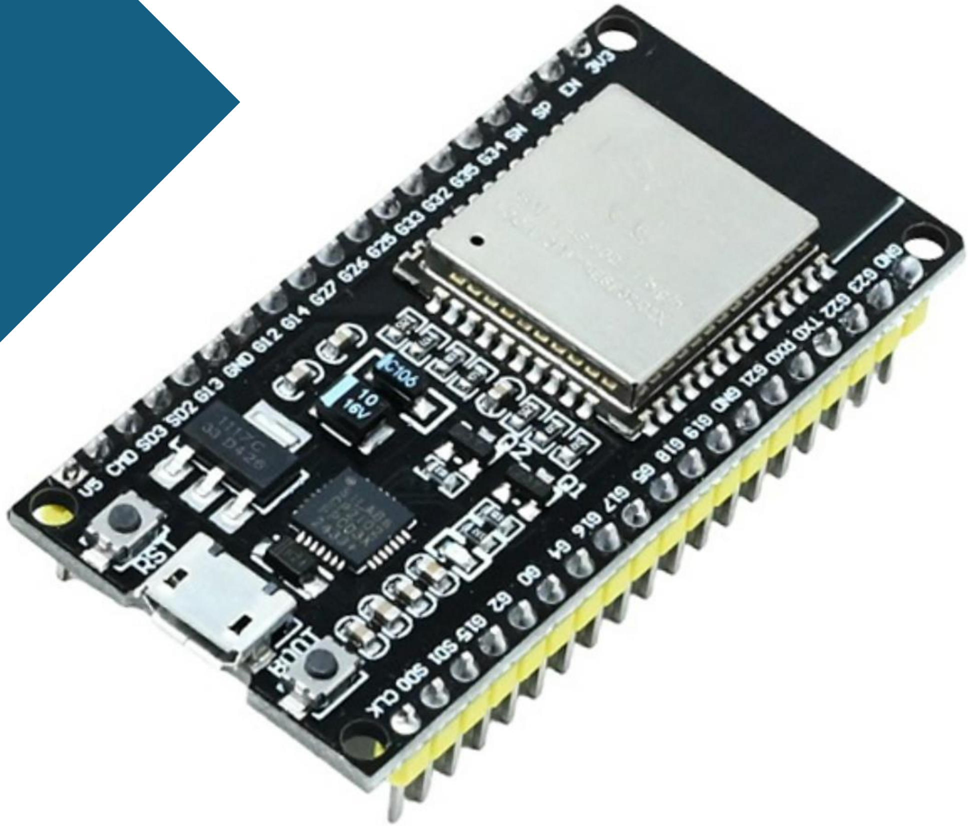
Time Period

IR Obstacle Detection Sensor

ESP32	Sensor
3.3v	Vcc
GND	GND
GPIO 13/14/25	Out



Push Button
Capacitive Touch
IR Sensor



LED(s)
Buzzer
NeoPixel

Interactive System

Learning Outcomes Week 5

- More clarity on Debugging
- Random Number
- For Loop
- Interfacing NeoPixel with ESP32
- Using For loop for NeoPixel
- PWM
- Controlling LED using PWM
- IR Sensor