

„Ha beérjük annyival, hogy elátkozzuk vagy dicsőítjük a technikát, akkor sohasem jutunk el lényegének a megragadásához.”

Martin Heidegger

MIKROVEZÉRLŐS RENDSZERFEJLESZTÉS

ChibiOS/RT

PWM: Impulzus szélesség moduláció

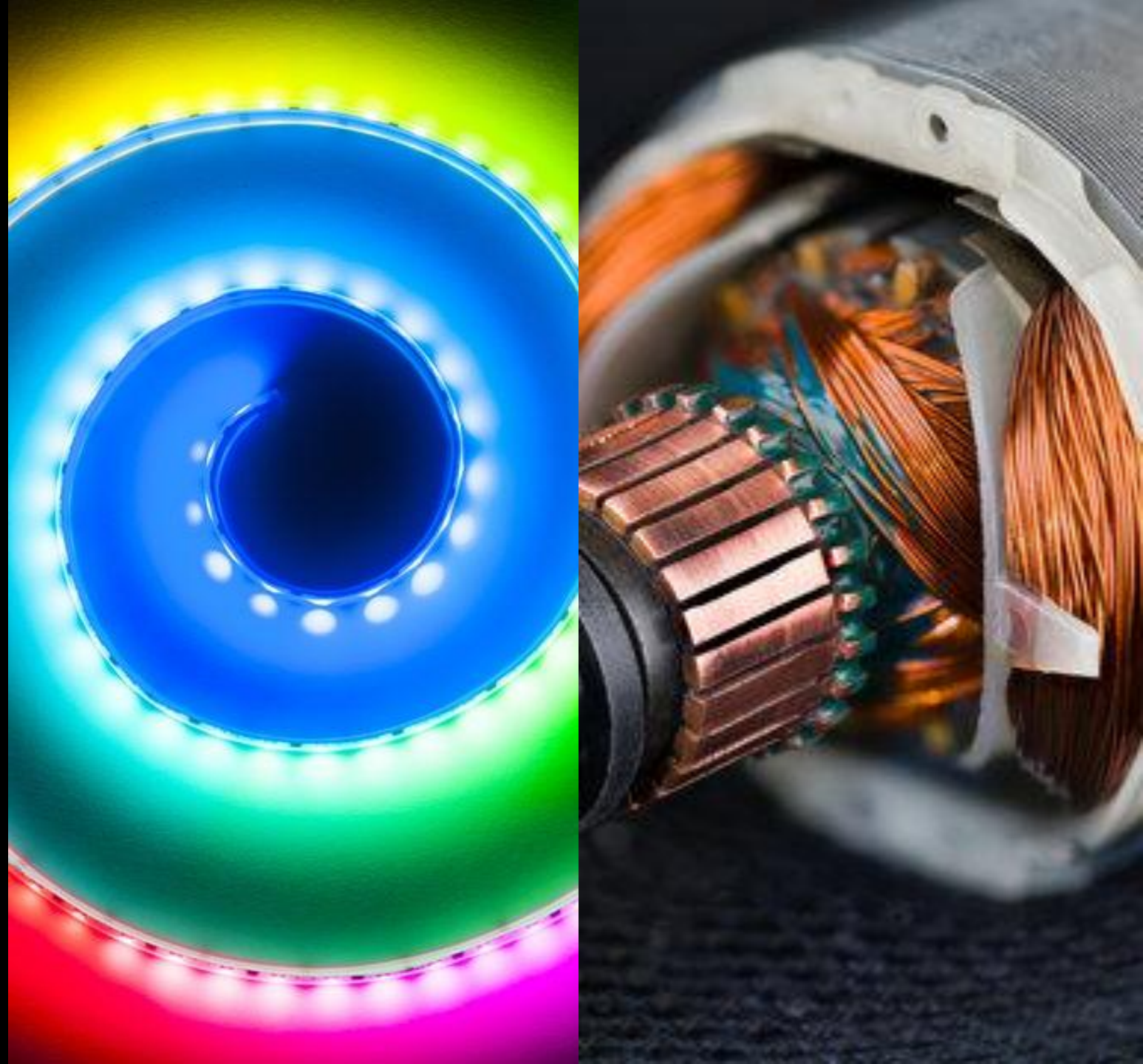
Zsupányi Krisztián

PWM: Impulzus szélesség moduláció

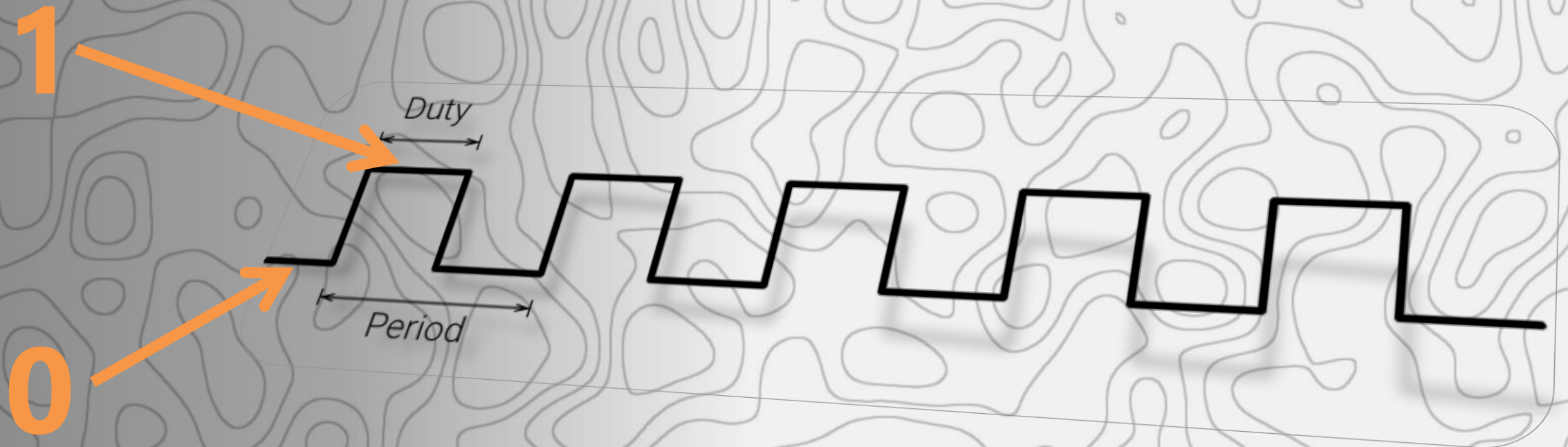


STARTING STARTING STARTING STARTING

PWM: IMPULZUS SZÉLESSÉG MODULÁCIÓ



DIGITÁLIS -> PWM->ANALÓG



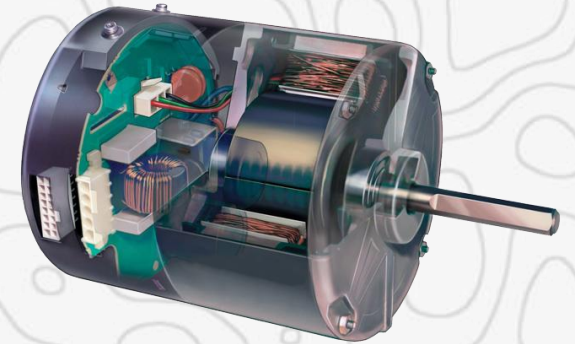
Digitális **ki** és **be** értékekkel **szabályozzuk az analóg** komponensek energiáját.

MIRE HASZNÁLHATÓ?



Vezérlési feladatok

- Motor sebesség vezérlése
- LED fényerő vezérlése



Lényege: minimalizálja az energiavesztést a szabályzás közben

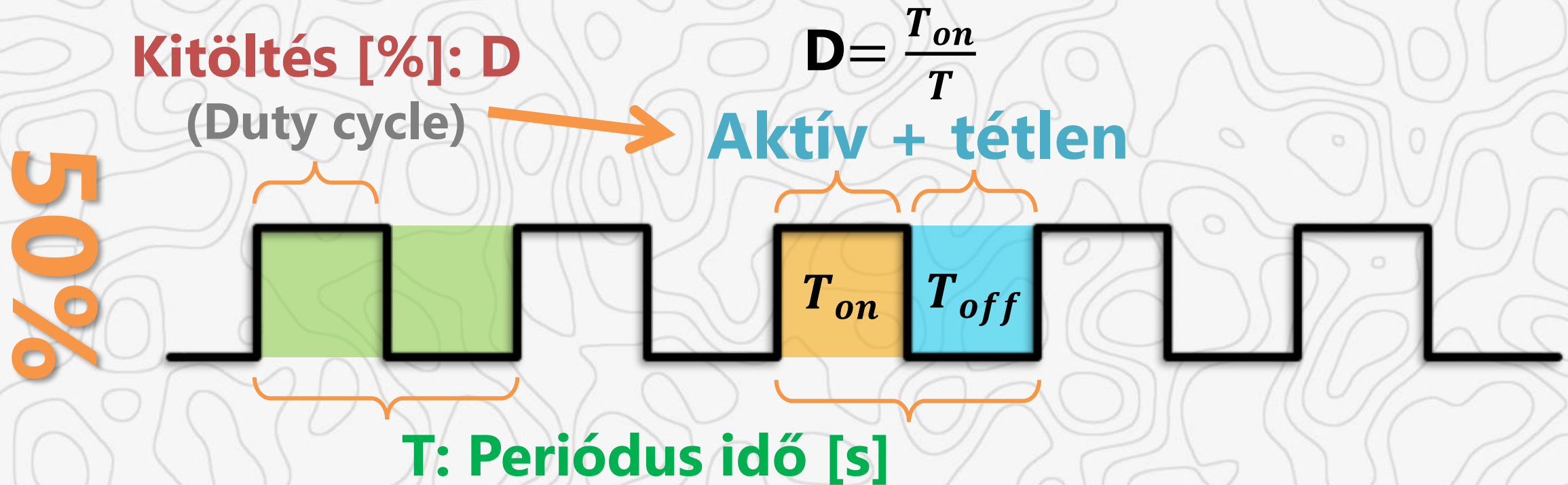


Kommunikáció

- RC járművek, servo
- Távolságmérők, mérési adatai



PWM JEL ÖSSZETEVŐI



Frekvencia [Hz]: $F = \frac{1}{T}$

Periodikusan ismétlődő impulzusok sorozatát hozzuk létre.

PWM JEL: KÜLÖNFÉLE KITÖLTÉSSEL

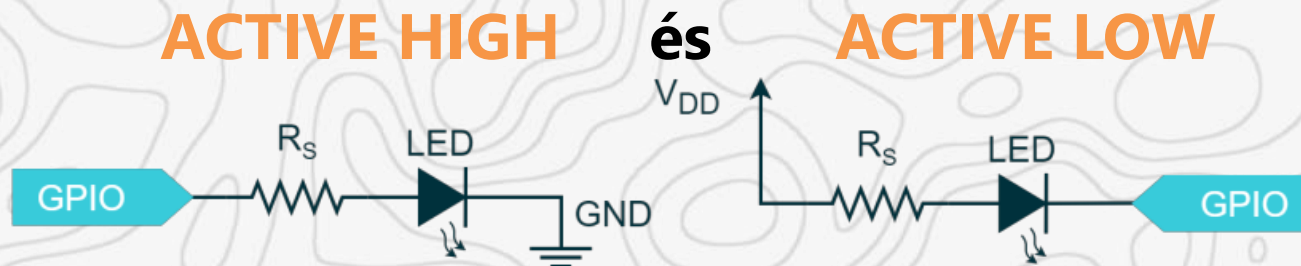
Állapotok időtartama nem kell megegyezzen.

A periódusidő és a frekvencia viszont nem változik!

Aktív állapot nem feltétlen a magas szint.

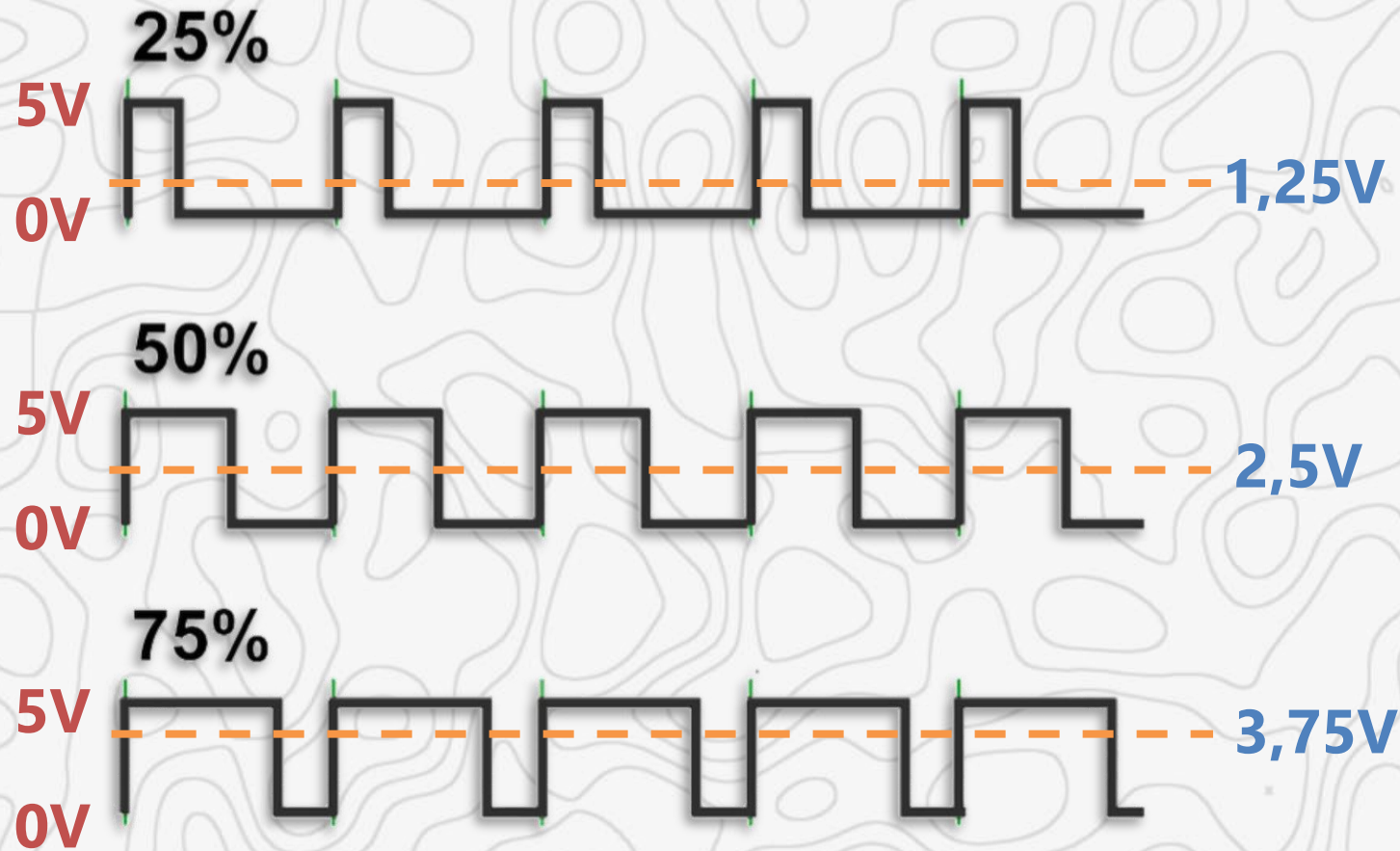


$\longleftrightarrow T$

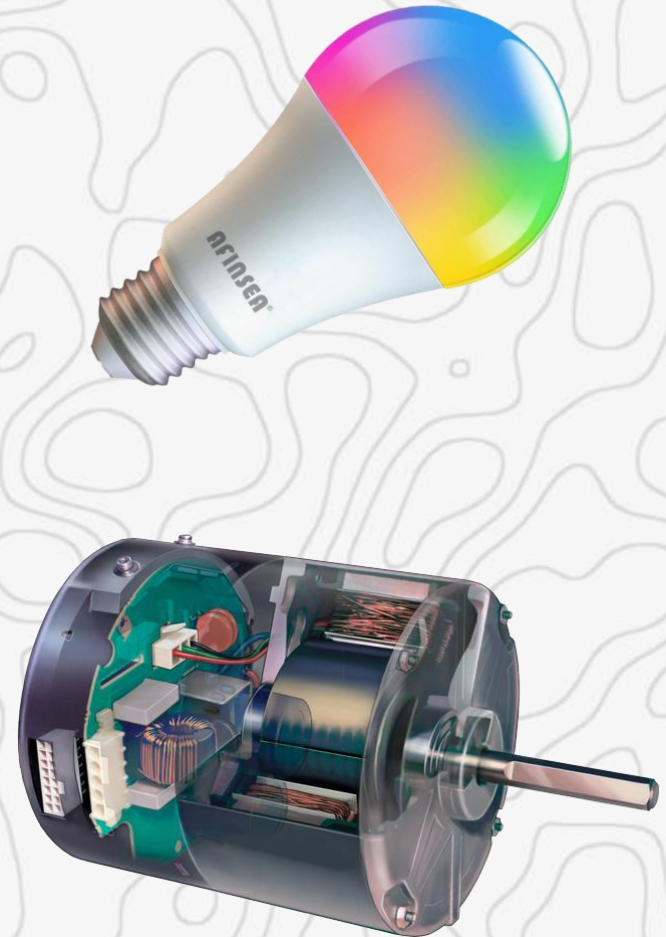


HOGY LESZ EBBŐL VEZÉRLÉS?

Amit vezérelni szeretnénk sok esetben analóg:

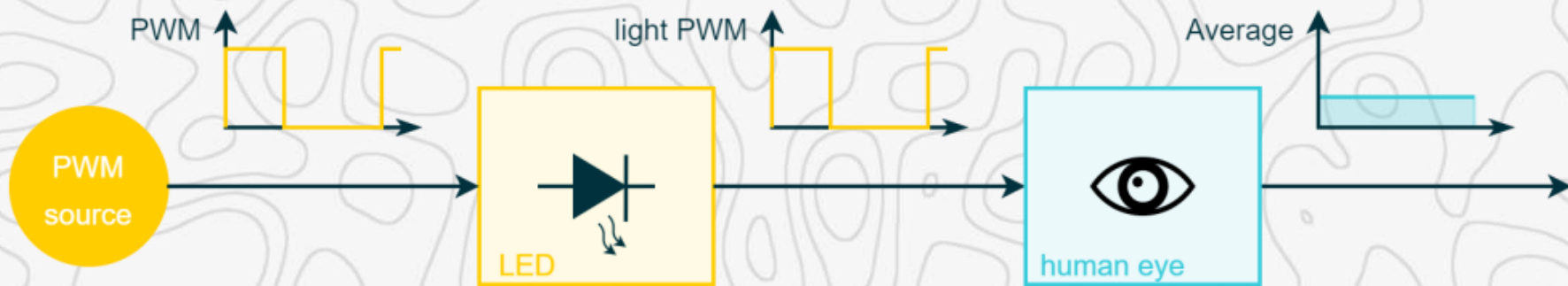


Kitöltési tényező -> átlagérték

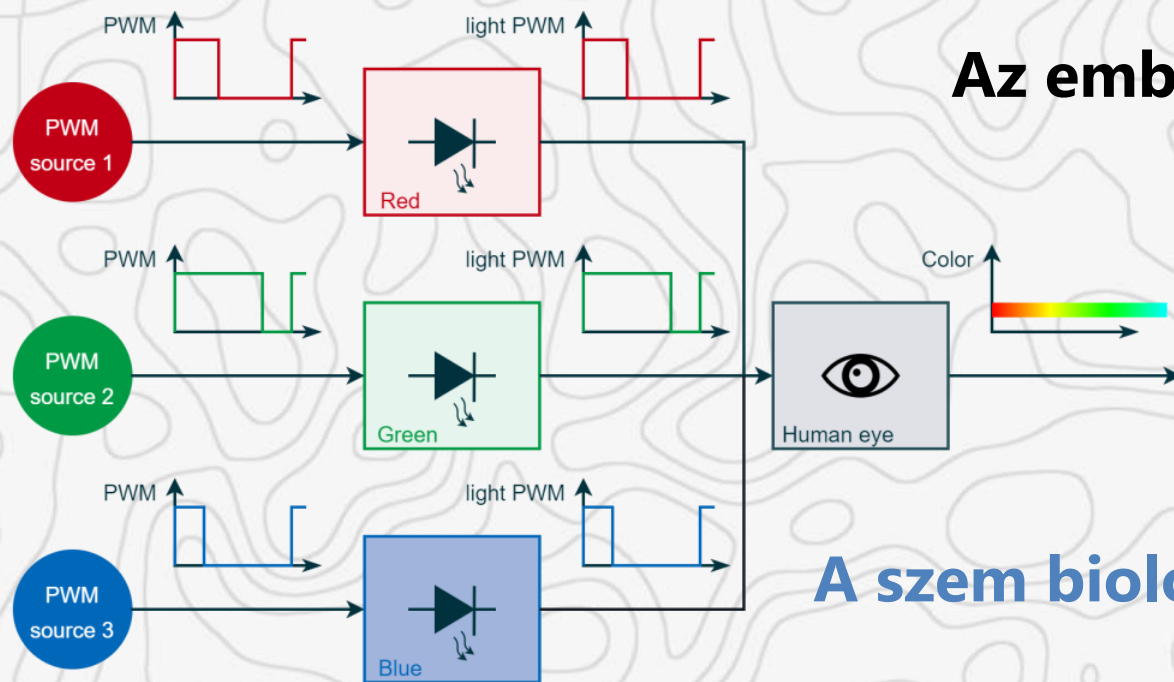


Kvází analóg érték keletkezik

ÁTLAGÉRTÉK A LED ESETÉN



Az emberi szem nem tudja követni a LED gyors villogását.



> 100 Hz

A szem biológiai aluláteresztő szűrő

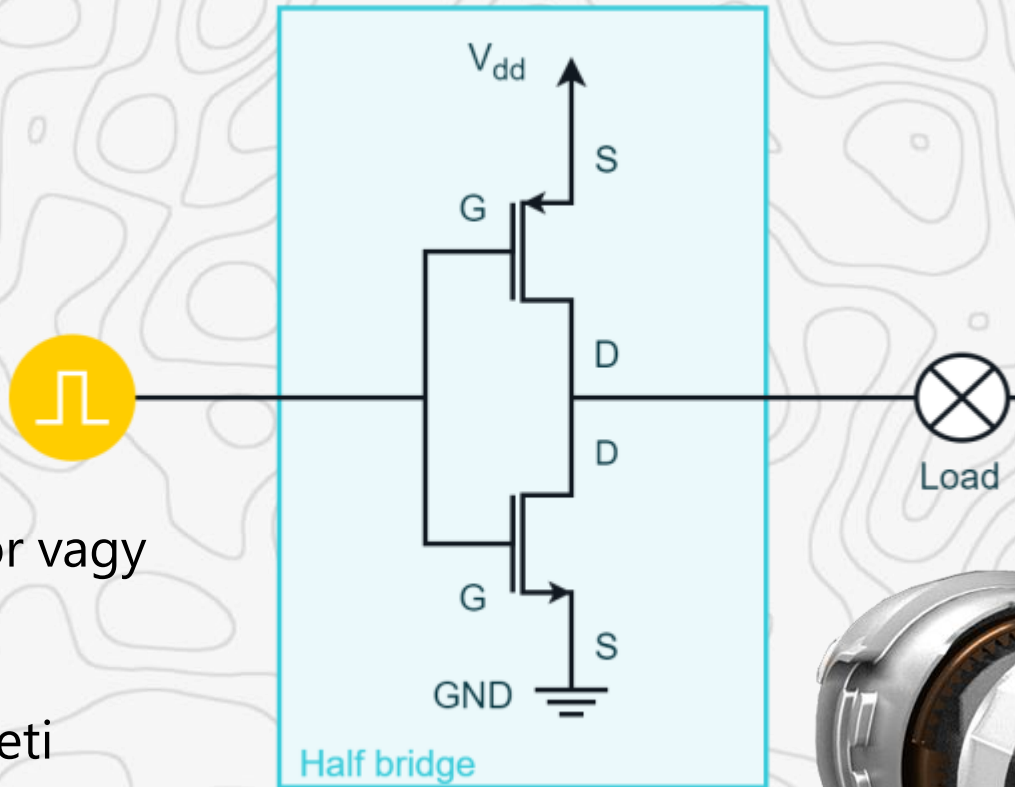


PWM: Impulzus Szélesség Moduláció, Pulse Width Modulation

ÁTLAGÉRTÉK A MOTOR ESETÉN

- **Half bridge**, a motornál az egyik kivezetés fix,
- **2 tranzisztorral vagy föld vagy táp** köthető a motorral ezt PWM vezérli,
- **Egy irányba forog** de a PWM a **sebességet állítja**
- **PWM előnye a hatékonyság**, a tranzisztor vagy teljesen be van kapcsolva vagy ki,
- **„Nincs” átmeneti állapot**, „Nincs” átmeneti ellenállás, kevés hő keletkezik mivel nem esik nagy feszültség a kapcsoló elemen.

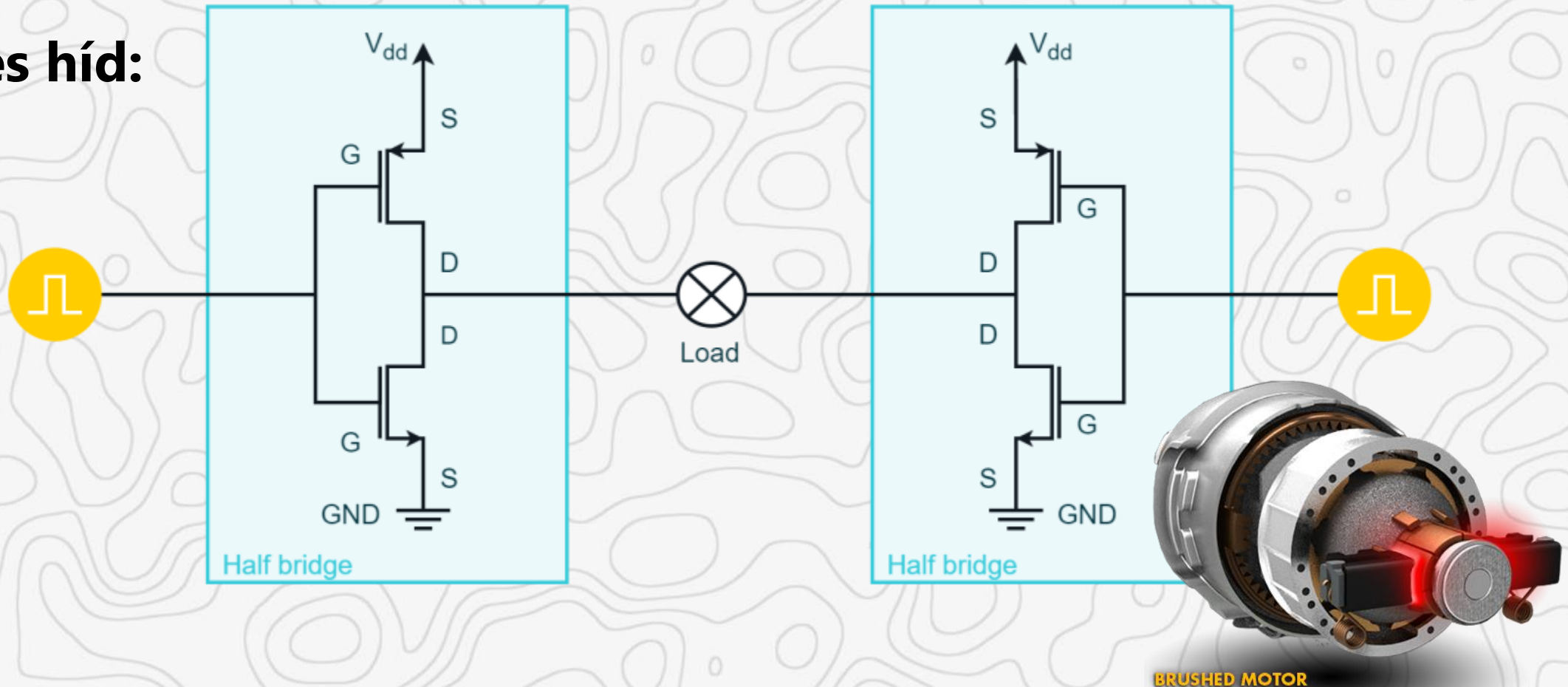
Félhíd:



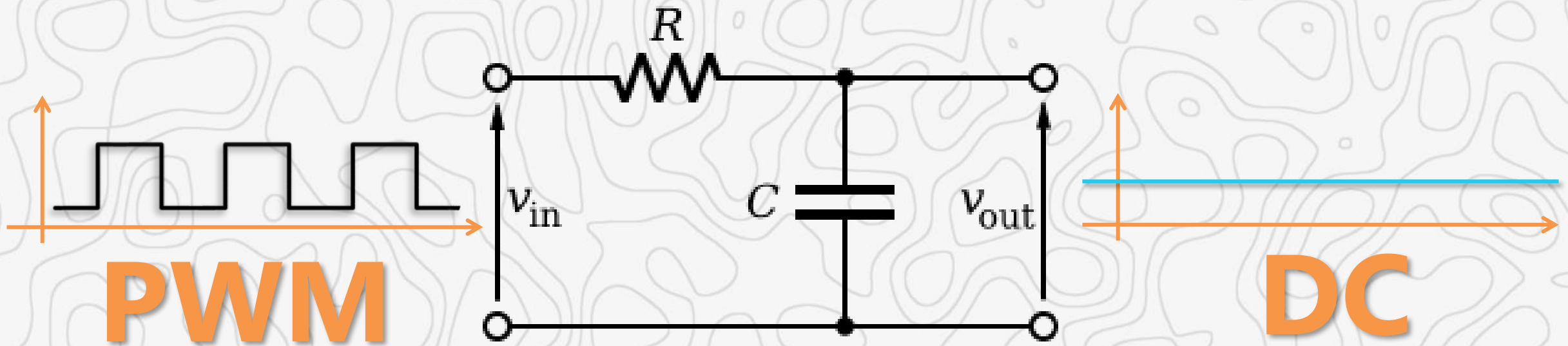
PWM: MOTOR SEBESSÉG ÉS FORGÁSIRÁNY

Full bridge, két félhíd amivel nem csak a sebesség hanem az áram irány változtatható, tehát a forgásirány.

Teljes híd:



PWM: ALULÁTERESZTŐ SZŰRŐ



PWM: MI TÖRTÉNIK A HÁTTÉRBE?

A **PWM-et időzítők generálják**, nem csak számlálásra használhatók. Képes hardveresen **pontos PWM jelet** kiadni a **kívánt frekvencián és kitöltéssel**.

Pl: 84 MHz



APB órajele

**Előosztó
(Prescaler)**

/84



Időzítő órajele:

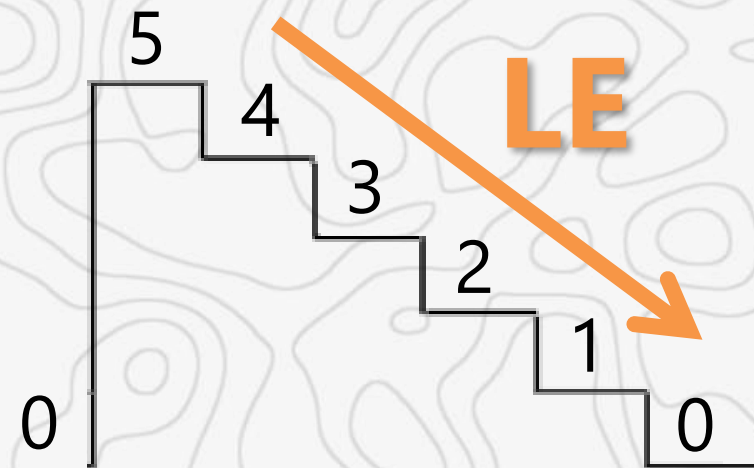
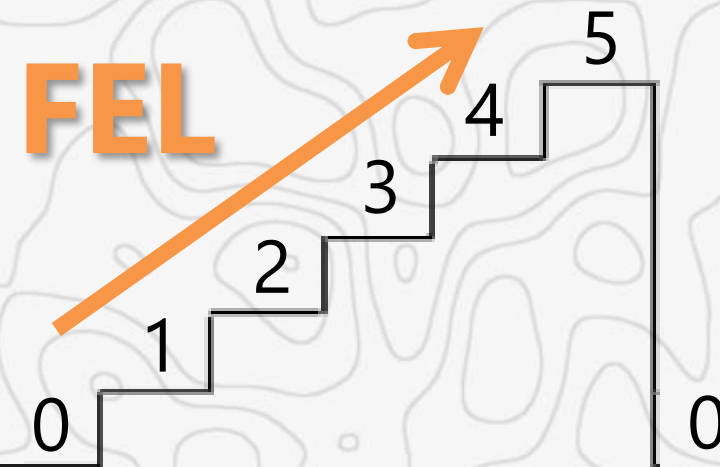
1 MHz

**Számláló
(Timer)**

16 vagy 32 bit

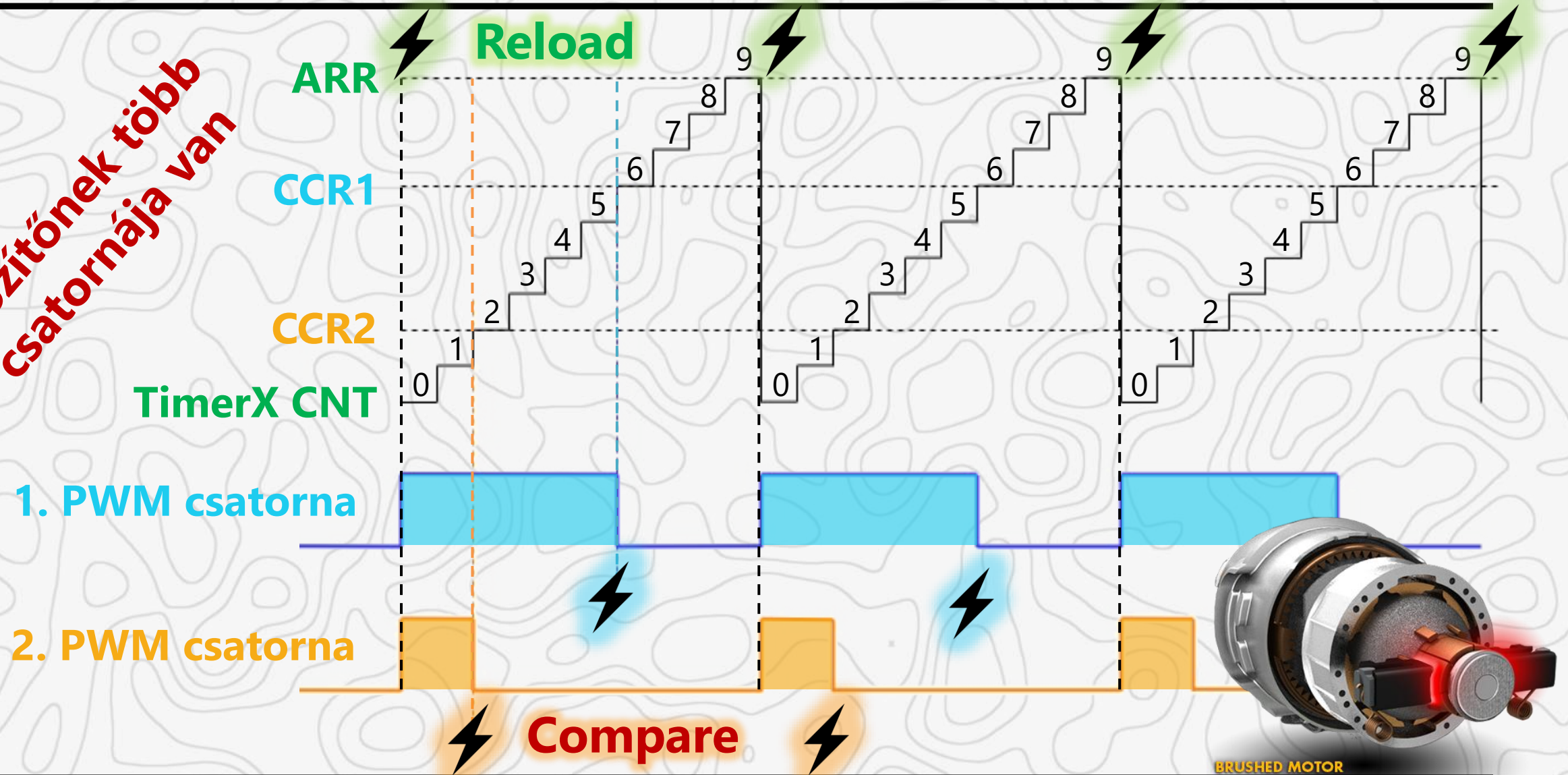
ARR (AutoReload Register): periódusidőt adja, periódus végén 0-ra vagy a beállított értékre áll vissza

Számít, hogy meddig számolunk? ... igen: felbontás



EDGE-ALIGNED PWM – ÁLTALÁBAN ÚGY VAN

Időzítőknek több csatornája van

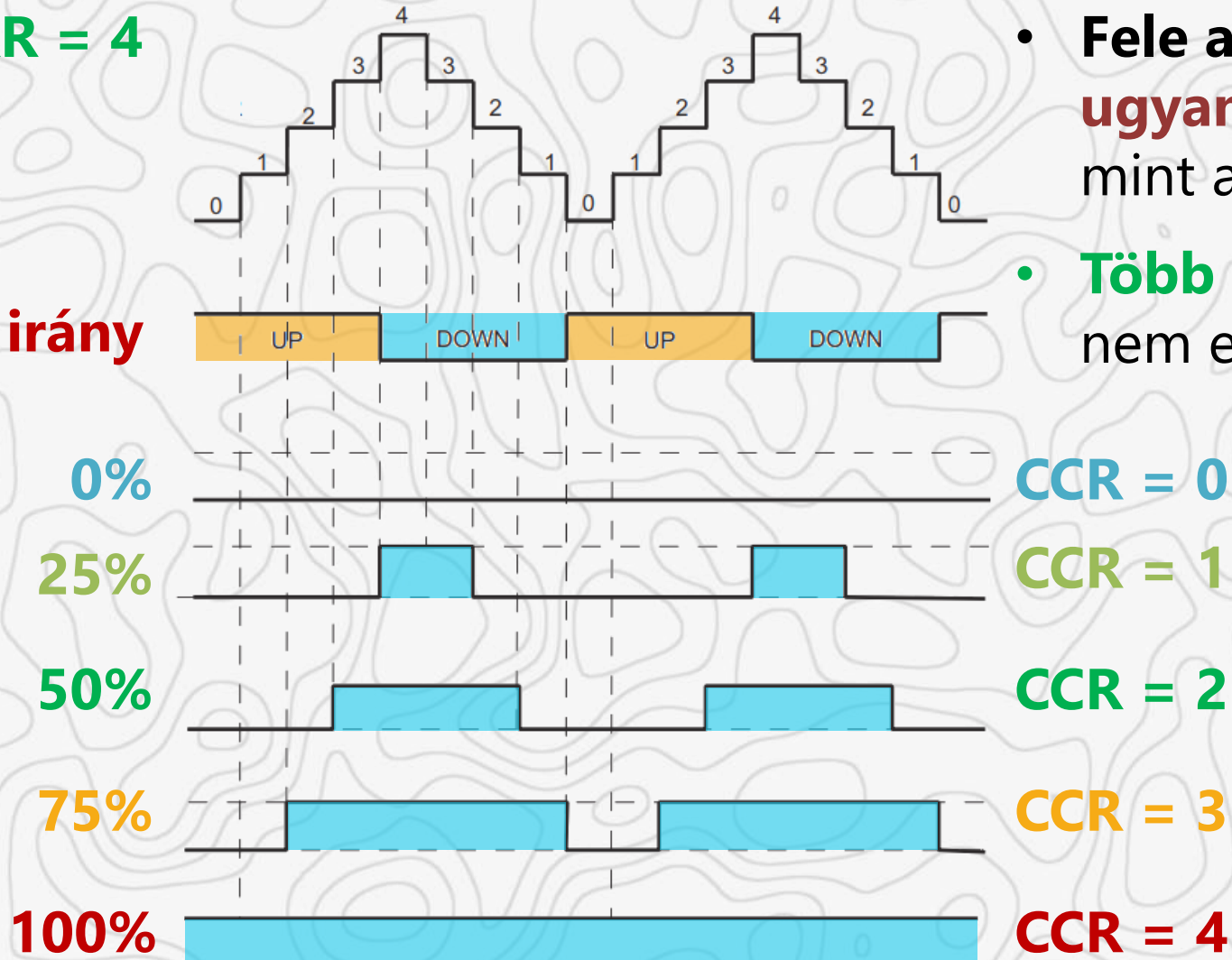


PWM: Impulzus Szélesség Moduláció, Pulse Width Modulation

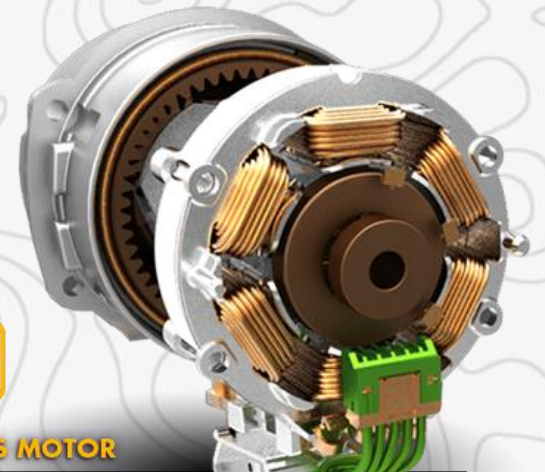
CENTER-ALIGNED PWM

ARR = 4

Számláló irány



- **Fele akkora ARR érték** kell, hogy **ugyanolyan frekvencia** legyen, mint az Edged verziónál,
- **Több csatornánál EMI-t csökkent,** nem egyszerre rántja meg a tápot.

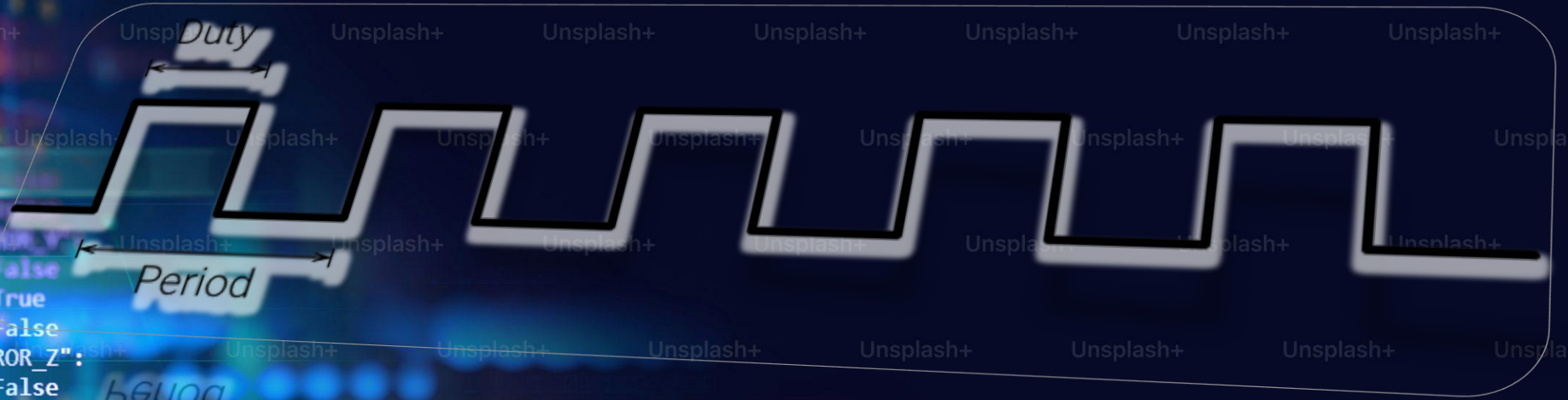


PWM: Impulzus Szélesség Moduláció, Pulse Width Modulation


```
elif operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob

mirror_ob.select = 0
time = bpy.context.selected_objects[0]
bpy.data.objects[time.name].select = 1
```



CHIBIOS: PWM

PWM: Impulzus Szélesség Moduláció, Pulse Width Modulation

PWM: FIZIKAI PORT KIVÁLASZTÁSA



PWM driverek: PWMD1, PWMD2, PWMD3...

- **mcuconf.h** -> PWM periféria engedélyezés
- **halconf.h** -> PWM kezelő kód engedélyezés

! Nincs a board fájlban konfigurálva:

```
palSetPadMode(GPIOD, GPIOD_LED4,  
PAL_MODE_ALTERNATE(2));
```

Driver indítása:

```
pwmStart (PWMDriver *pwmp,  
          const PWMConfig *config );  
pwmStop(PWMDriver *pwmp);
```


PWM: CHIBIOS KONFIGURÁCIÓ

PWM driver konfiguráció:

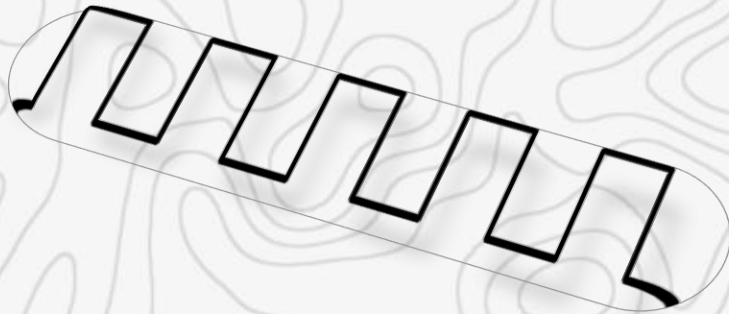
PWM Timer frekvencia:

Periódus hossza, tick-ben:

Timer reload callback:

Csatornák konfigurációja:

- **Aktív jelszint (magas)**
- **PWM compare callback**



```
static PWMConfig pwmcfg = {  
    100000, } 100 000 Hz / 100 = 1000 Hz  
    100,  
    pwmpcb,  
    {  
        { PWM_OUTPUT_ACTIVE_HIGH, pwmc1cb },  
        { PWM_OUTPUT_ACTIVE_HIGH, NULL },  
        { PWM_OUTPUT_ACTIVE_HIGH, NULL },  
        { PWM_OUTPUT_ACTIVE_HIGH, NULL },  
    },  
    0,  
    0,  
};
```

PWM: CHIBIOS FÜGGVÉNYEK

PWM csatorna bekapcsolása és kitöltés beállítása:

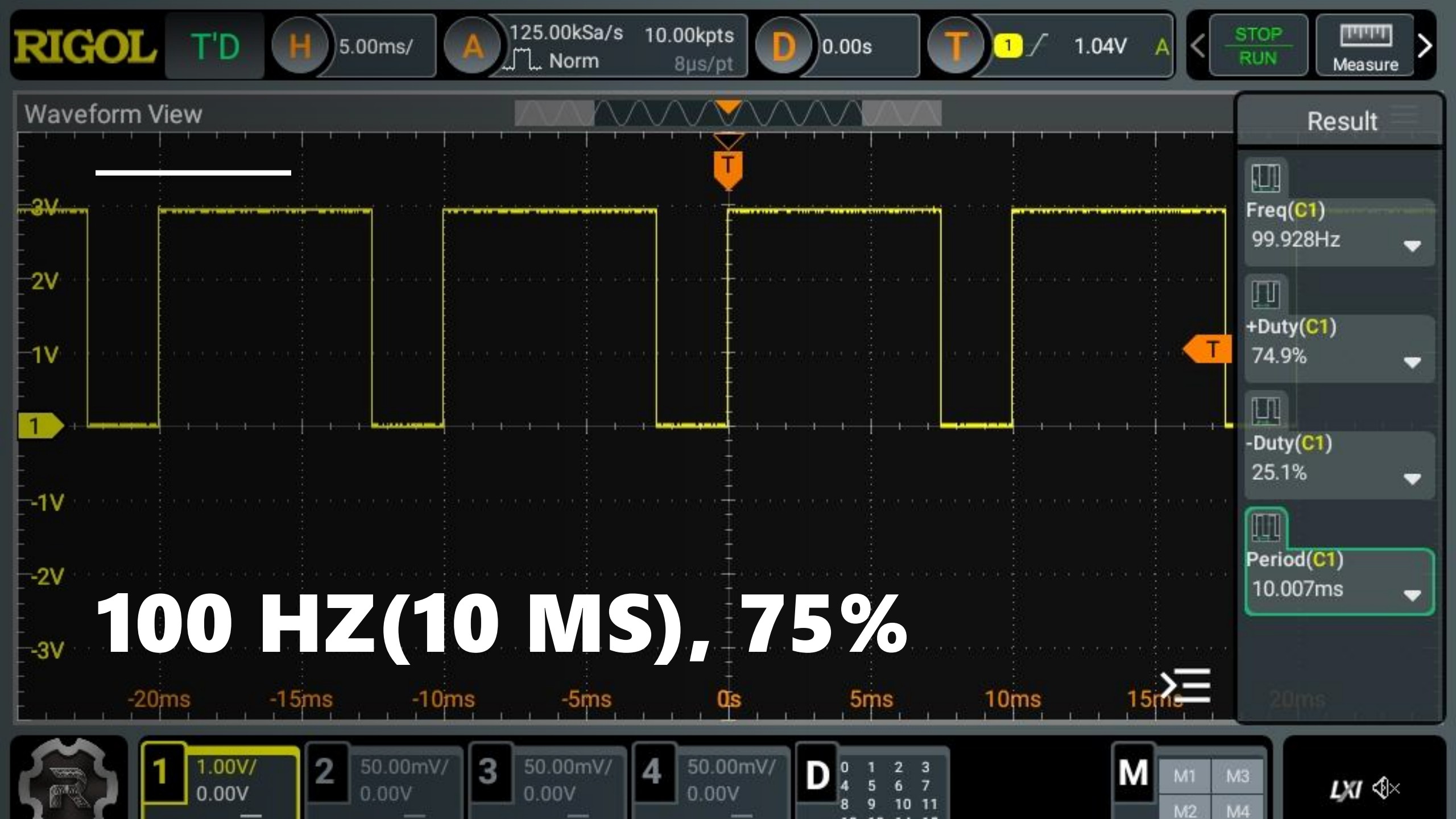
```
void pwmEnableChannel (PWMDriver *pwmp, pwmchannel_t channel,  
                      pwmcnt_t width);  
  
pwmEnableChannel(&PWMD4, 0,  
                PWM_PERCENTAGE_TO_WIDTH(&PWMD4, 7500));
```

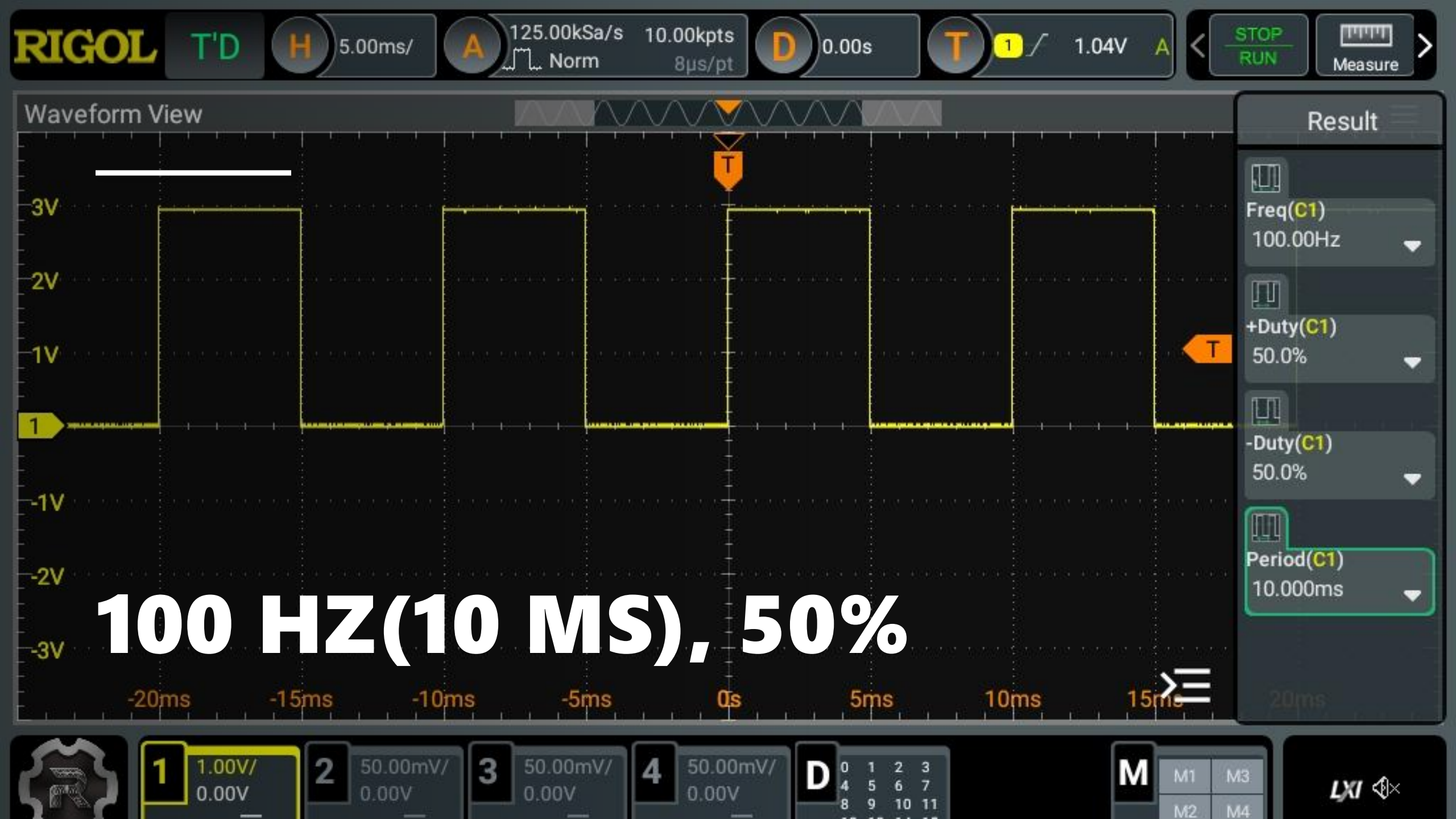
PWM csatorna kikapcsolása :

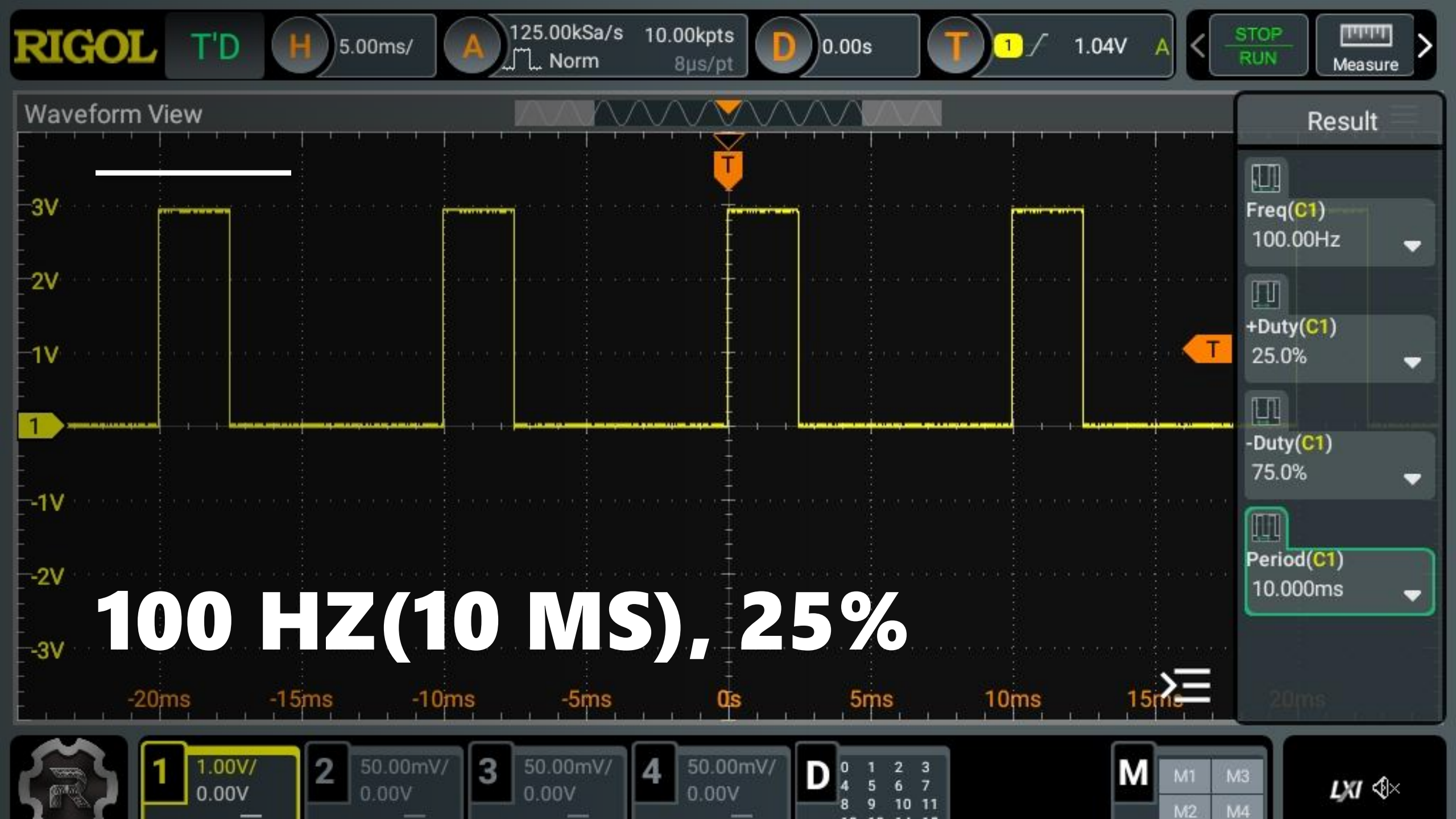
```
void pwmDisableChannel(PWMDriver *pwmp, pwmchannel_t channel);  
  
pwmDisableChannel(&PWMD4, 0);
```

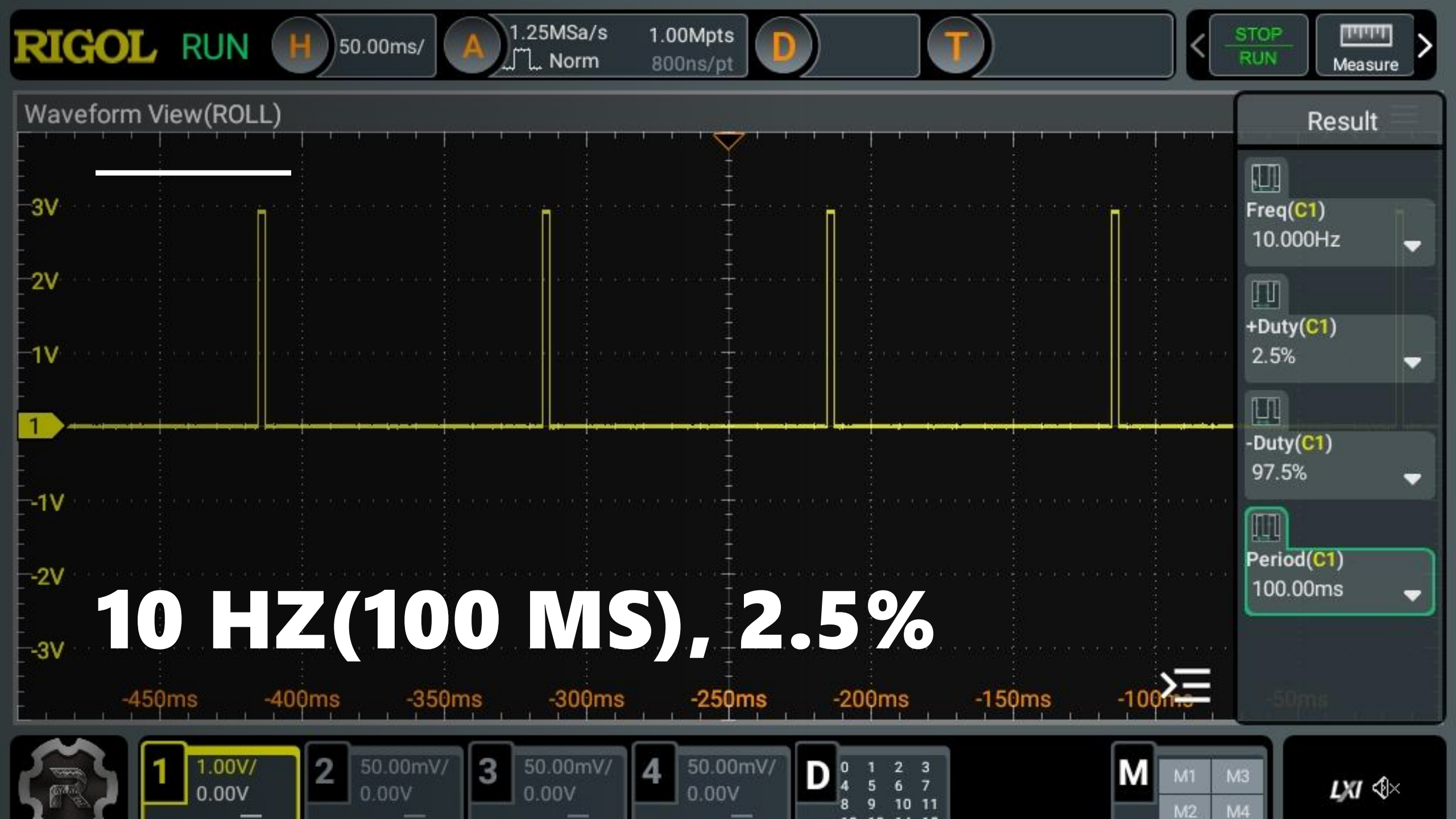
Periódusidő változtatása (tick-ben):

```
void pwmChangePeriod(PWMDriver *pwmp, pwmcnt_t period);  
  
pwmChangePeriod(&PWMD4, 1000);
```







RIGOL

T'D

H

5.00ms/

A

125.00kSa/s
Norm

10.00kpts
8 μ s/pt

D

0.00s

T

1 / 350.00mV A

STOP
RUN

Measure

Waveform View



Result



Period(C1)

10.000ms



+Duty(C1)

9.1%



-Duty(C1)

90.9%



1

1.00V/
-1.50V

2

50.00mV/
0.00V

3

50.00mV/
0.00V

4

50.00mV/
0.00V

D

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

M

M1

M3

M2

M4

LXI

„Ha beérjük annyival, hogy elátkozzuk vagy dicsőítjük a technikát, akkor sohasem jutunk el lényegének a megragadásához.”

Martin Heidegger

KÖSZÖNÖM A FIGYELMET!

Zsupányi Krisztián

SPI: Gyorsulásmérő



ENDING ENDING ENDING ENDING E