



# *Digitális technika 2.*

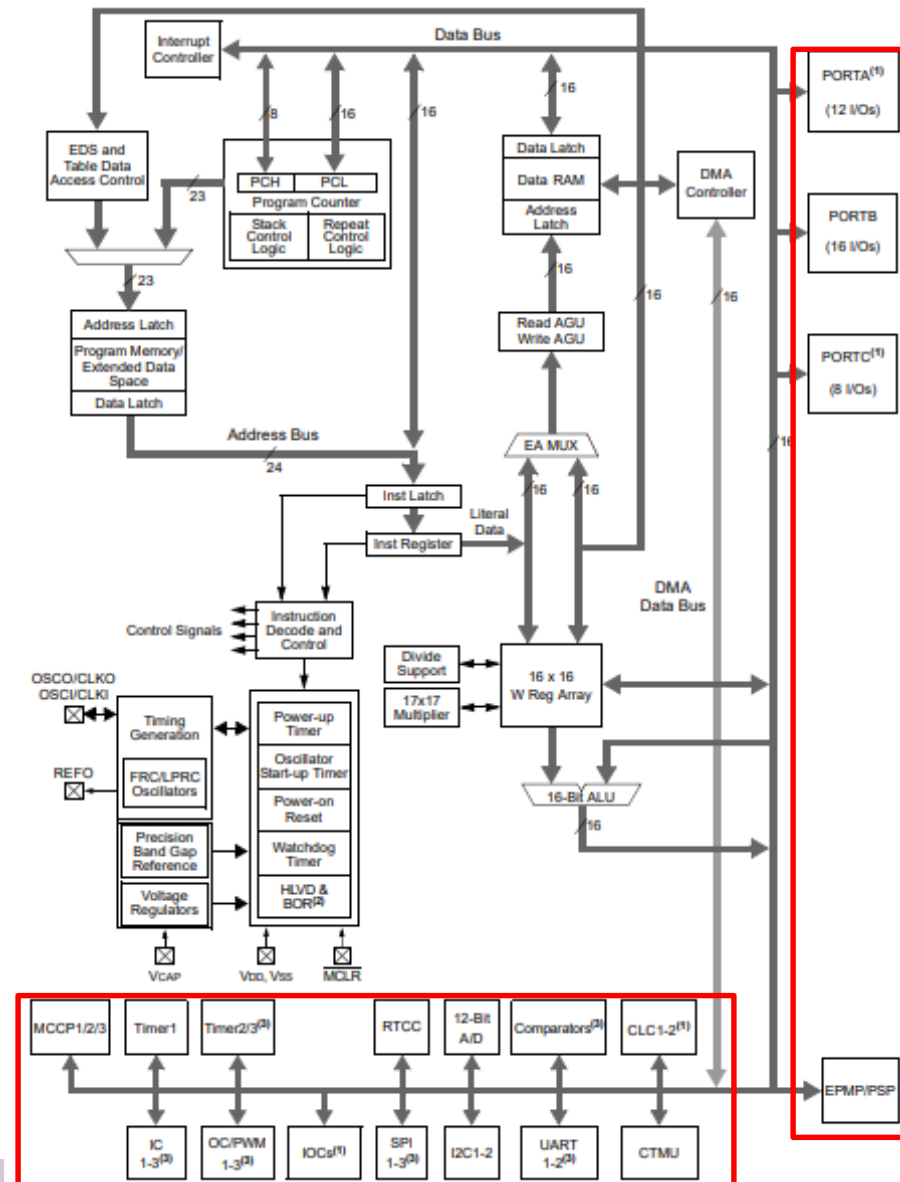
*BMEV8IAA06*

## *8. előadás*

*Időzítő alapú perifériák*

- Megszakítás
  - Speciális függvényhívás
  - Hardver vagy szoftver esemény váltja ki
  - Jobb alternatíva a szoftveres időzítésre
- Időzítő
  - Speciális, általános célú periféria
  - Hardveresen egy egyszerű felfele számláló, választható órajellel és periódussal.
  - Lejárása esetén megszakítást okoz
  - Külső jelek megszámlálására is használható
  - Külső impulzusok hosszának megmérésére is alkalmas

# Perifériák a PIC24FJ256GA705-ben

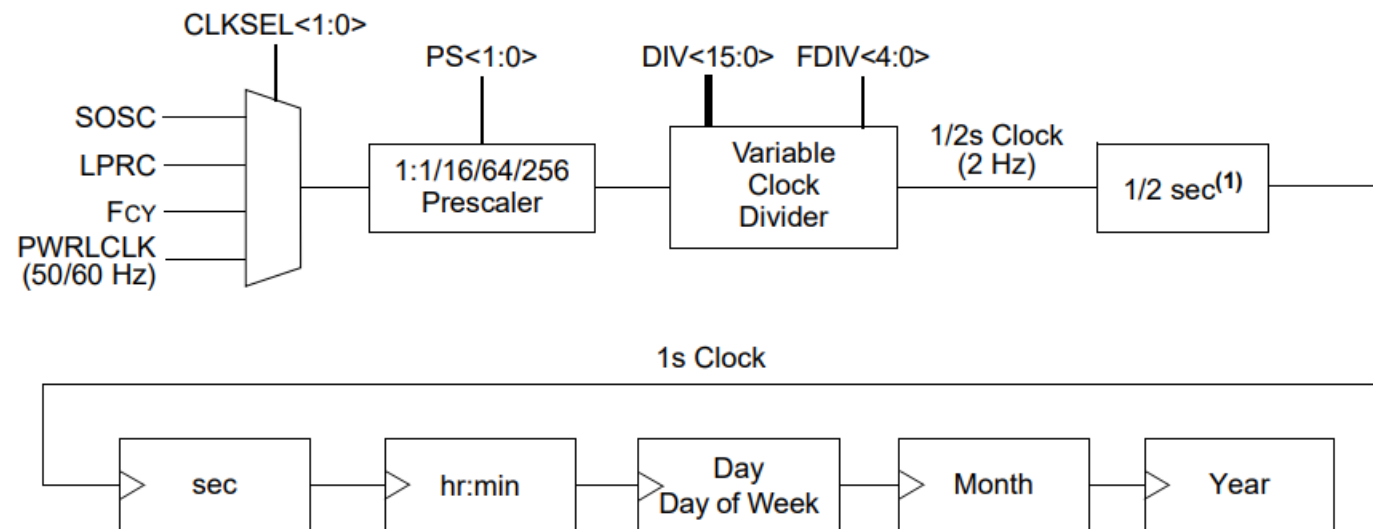


# Mire használhatunk még időzítőt?

- Konkrétan órának
  - » RTCC → real time clock calendar
- Programhiba (timeout) észlelésére
  - » WDT → watchdog timer
- Külső jelek mérésére
  - » ICT → Input capture timer
- Adott hosszú vagy távolságú impulzusok előállítására
  - » OCT → Output compare timer
- Adott kitöltési tényezőjű négyszögjel előállítására
  - » PWM → Pulse width modulation

# Valós idő óra (RTCC)

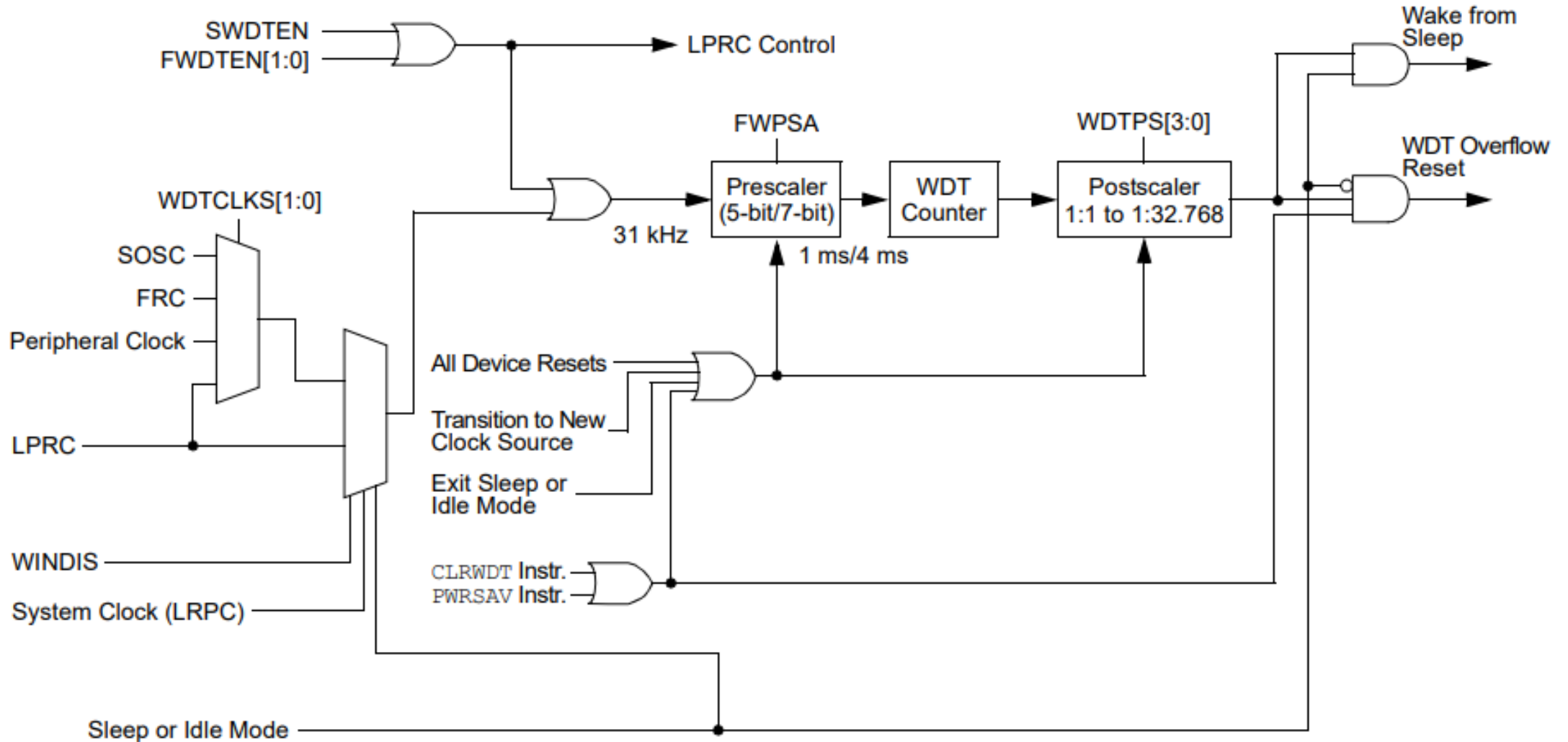
- Speciális timer, ami pont a valós órának megfelelően jár.
- Többféle órajelet választhatunk, de azt az állítható osztóval 2Hz-esre le kell osszuk
  - Ez az egyetlen timer ahol nem csak 2-hatvánnyal oszthatunk az előosztóban
- A többi rész innentől pont jól fogja számolni a másodperceket, perceket, órákat, napokat, hónapokat, éveket.
  - BCD-ben ábrázol
- A hét napjait (hétfő, kedd, stb.) is tudja
- A szökőéveket is tudja (2099-ig)
- Van Alarm funkciója
  - tud szólni adott dátum és idő elérésekor



# Watchdog timer (WDT)

- Speciális számláló
  - Ez is felfele számol
  - Különböző órajelekről mehet, általában a belső lassú órajelről szokott (LPRC, 32kHz)
- Leállítható, de ha el van indítva és túlcsordul, akkor újraindítja a processzort
  - Alvó módban (alacsony fogyasztású mód, ahova a `pwrsav` utasítás küld) nem indít újra, csak „felébreszt” → folytatódik a programvégrehajtás a következő utasításról
- Speciális utasítással törölhető az értéke  
`clrwdt`
- Általában biztonság-kritikus eszközökben arra használjuk, hogy a program „eltévedését” észre vegyük
  - » A programban bizonyos helyeken szándékosan töröljük, ha valamiért nem jut el adott időnként ide a program, akkor valami gond van, „lefagytunk”, akkor újraindulhatunk...
- Néha nagyon kis fogyasztású eszközökben arra használjuk, hogy periodikusan kihozza a processzort az alvó módból
  - » Pl. azért, mert csak percenként vagy óránként kell valamit gyorsan megcsinálni, és utána alhatunk tovább
  - » Itt is véd a programhiba ellen, mert ha valamiért nem aludtunk vissza mire megint fel kéne ébredni, akkor újraindít...

# Watchdog timer (WDT)



# Kitekintés: „Alvó” üzemmódok

- Energiatakarékosság céljára szolgálnak
  - Kettő is van belőle
- Ilyenkor a processzor nem hajt végre utasítást „megáll”
- Speciális utasítással kezdeményezhetők:

```
pwrsav #0 ; SLEEP mód
```

```
pwrsav #1 ; IDLE mód
```

- Különbség:
  - Sleep módban minden órajel megáll (kivéve LPRC 32kHz)
  - Idle módban a processzor órajele megáll, de perifériák tovább működhetnek (általában egyesével kapcsolgathatók ki-be)



`pwrsav #0 ; SLEEP mód`

`pwrsav #1 ; IDLE mód`

- Mindkét módból megszakítás, watchdog, vagy reset hoz ki.
  - Ilyenkor a CPU és minden leállított periféria órajele visszakapcsol
  - Reset esetén természetesen a program előlről kezdődik *(ez nem túl hasznos)*
  - Megszakítás esetén a megszakítási szubrutin lefutása után a pwrsav utasítás után folytatódik a program.
  - Watchdog timeout esetén egyszerűen csak a pwrsav utáni utasításon folytatódik a program.

**Figyelem! Ha egy periféria órajele nem megy, az nem fog megszakítást okozni sem! ☹**

→ Jellemzően Sleep módból csak portláb külső megszakítása (ahhoz nem kell órajel) vagy watchdog (neki saját külön órajele van) tud kihozni.

# Mit tegyünk, ha épp nincs mit csinálni?

- Láttuk:

Vegtelen:

```
bra Vegtelen
```

→ Közben kb. 6-7 mA-t fogyasztunk

- Egyel jobb megoldás:

Vegtelen\_idle:

```
pwrsav #1
```

```
BRA Vegtelen_idle
```

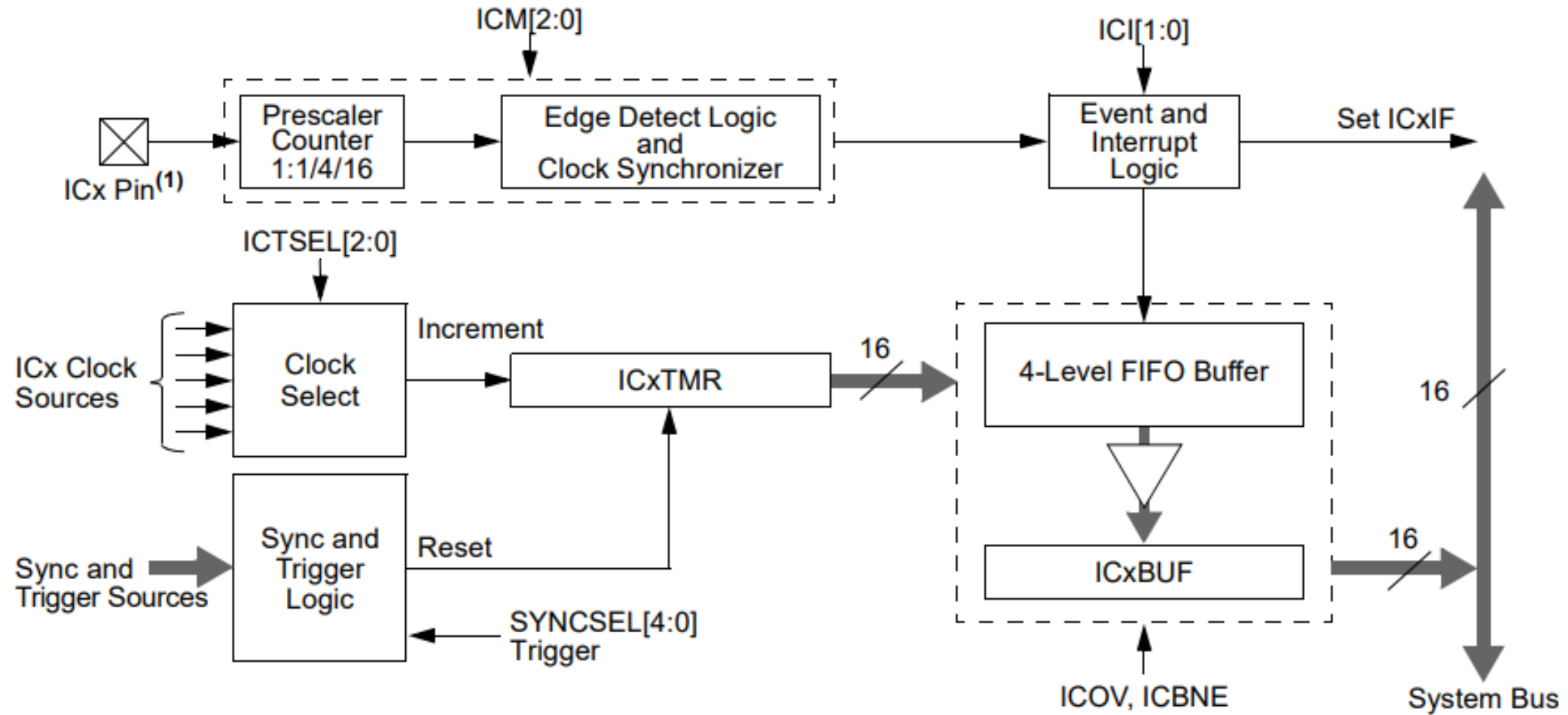
→ Közben kb. 1-2 mA-t fogyasztunk (a bekapcsolt perifériáktól függően)

→ Sleep-ben (pwrsav #0) már csak 15µA-t fogyasztanánk, de akkor gondoskodjunk róla, hogy legyen, ami felébreszt

# Input capture timer (IC vagy ICT)

- Általában belső órajelre lépő számláló
  - Lehet külső órajellel is járatni
- Periódus állítási lehetősége nincs (szabadon túlcsordul)
- Esemény hatására a számláló pillanatnyi értékét egy tárolóba menti
  - » Ettől „capture”
- Az esemény lehet külső portlábon megjelenő felfutó, lefutó, vagy mindkét él.
- Adott számú (1-4 db) esemény eltelte után megszakítást kérhet
- A tároló egy 4 regiszterből álló FIFO
  - » Az időpontok a beérkezés sorrendjében olvashatók ki, akár mind a 4 is.

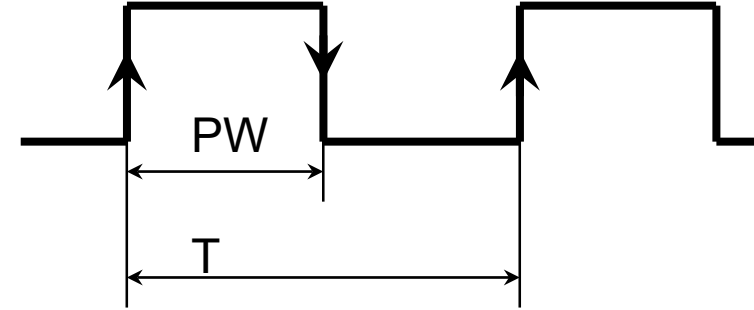
# Input capture timer (IC vagy ICT)



# Input capture alkalmazása

- Impulzus hosszának megmérése

$$PW = CAP2\uparrow - CAP1\downarrow$$



- Jel periódus idejének mérése

$$T = CAP2\uparrow - CAP1\uparrow$$

- Frekvencia mérés N darab minta után:

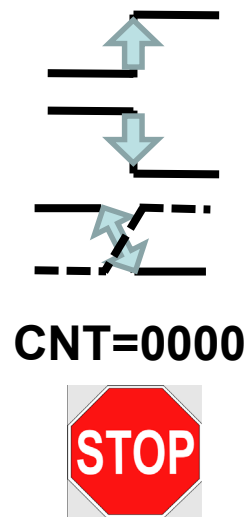
$$F = \frac{N}{Tösszes} = \frac{darabszám}{össz.idő}$$



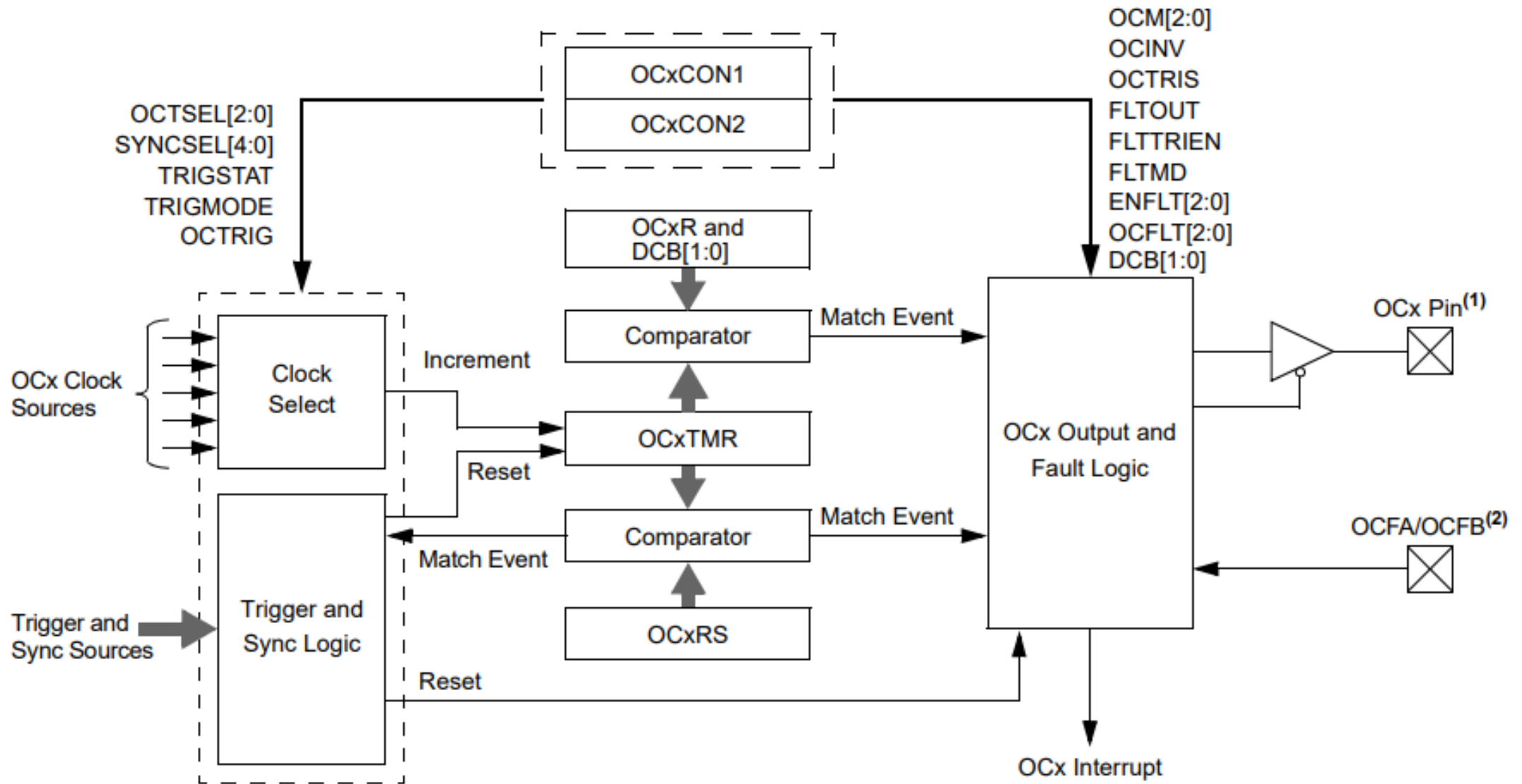
- Időbélyeg előállítása kommunikációhoz

# Output compare timer (OC vagy OCT)

- Ugyanúgy 16 bites számláló
- Két komparátora is van, és mindkettőhöz külön komparálási érték adható meg.
- A komparálási értékek elérésekor lehetőség van:
  - Adott kimenetet 1-be állítani
  - Adott kimenetet 0-ba állítani
  - Adott kimenetet invertálni
  - A számlálót letörölni
  - A számlálást megállítani



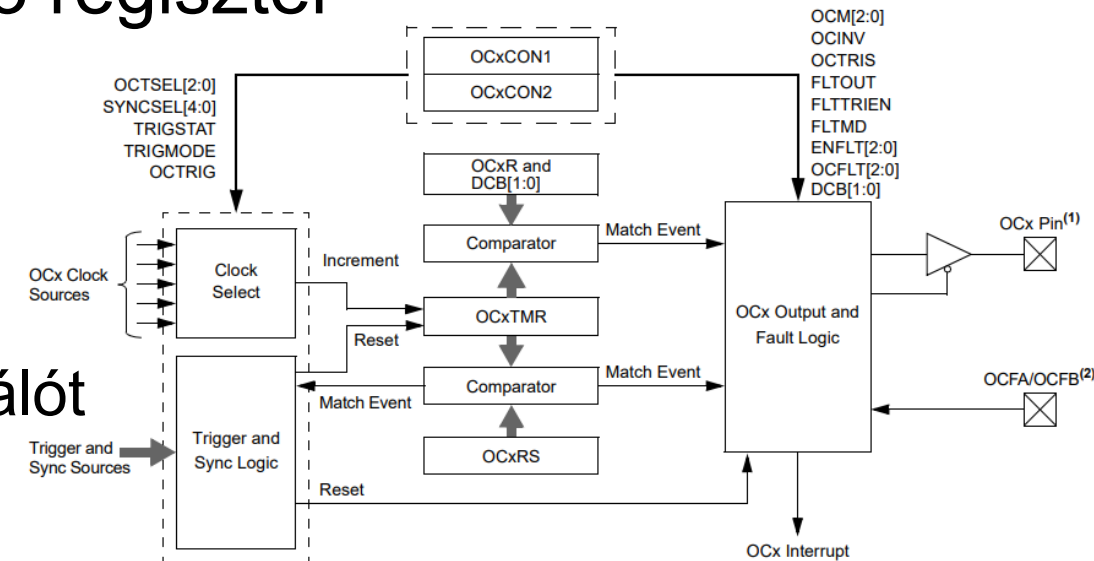
# Output compare timer (OC vagy OCT)



# Output compare timer (OC vagy OCT)

- Három regiszter:
  - OCxTMR: Maga a számláló
  - OCxR: Egyik komparátorhoz tartozó regiszter
  - OCxRS: Másik komparátorhoz tartozó regiszter
- x: a modul sorszáma

→ az adott típusban 1-3-ig
- Sync source:
  - Ezen a bemeneten felfutó élre törli a számlálót
- Trigger source:
  - Ezen a bemeneten felfutó élre indul vagy leáll a számlálás

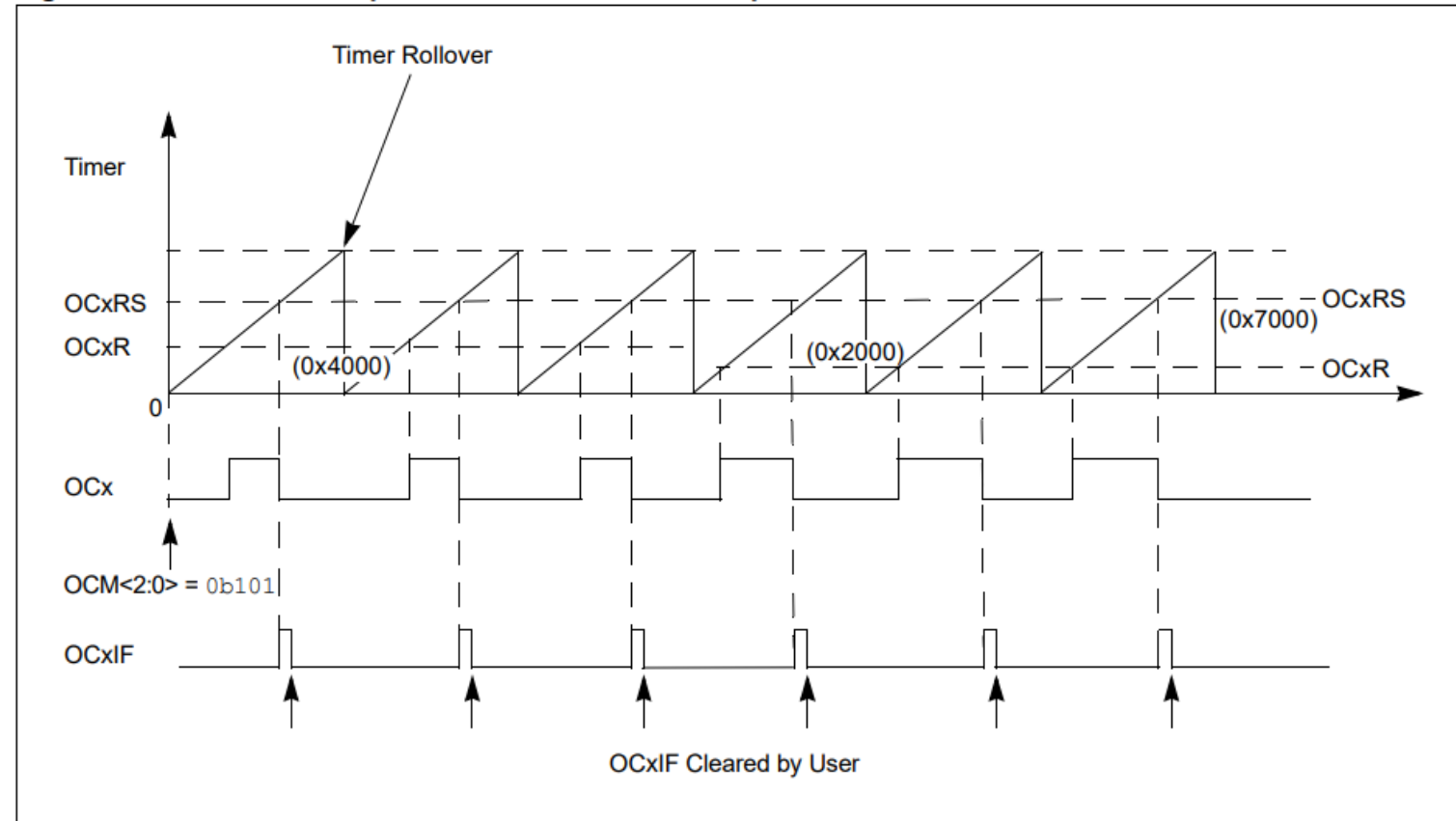




# Output compare alkalmazása

- Adott idejű és adott hosszú pulzus (sorozat) kiadása
  - » Dual compare mód

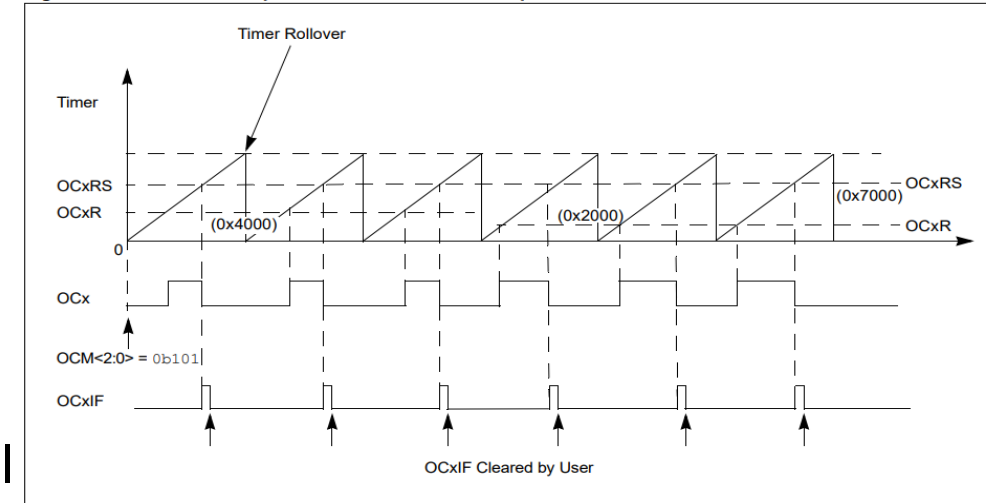
Figure 3-9: Dual Compare Mode – Continuous Output Pulses



# Output compare alkalmazása

- Adott idejű és adott hosszú pulzus (sorozat) kiadása
  - » Dual compare mód
- A modul inicializálásakor a kimenet 0.
- OCxTMR folyamatosan felfele számol
  - » „Single pulse” módban túlcsordulásakor megáll
  - » „Continuous pulses” módban túlcsordulás után is tovább folytatódik
- OCxR értékének elérésekor, a kimenet 1-esbe vált.
- OCxRS értékének elérésekor, a kimenet 0-ba vált.
- A kimenet 0-ba váltásával egyidőben megszakítást is kér.

Figure 3-9: Dual Compare Mode – Continuous Output Pulses

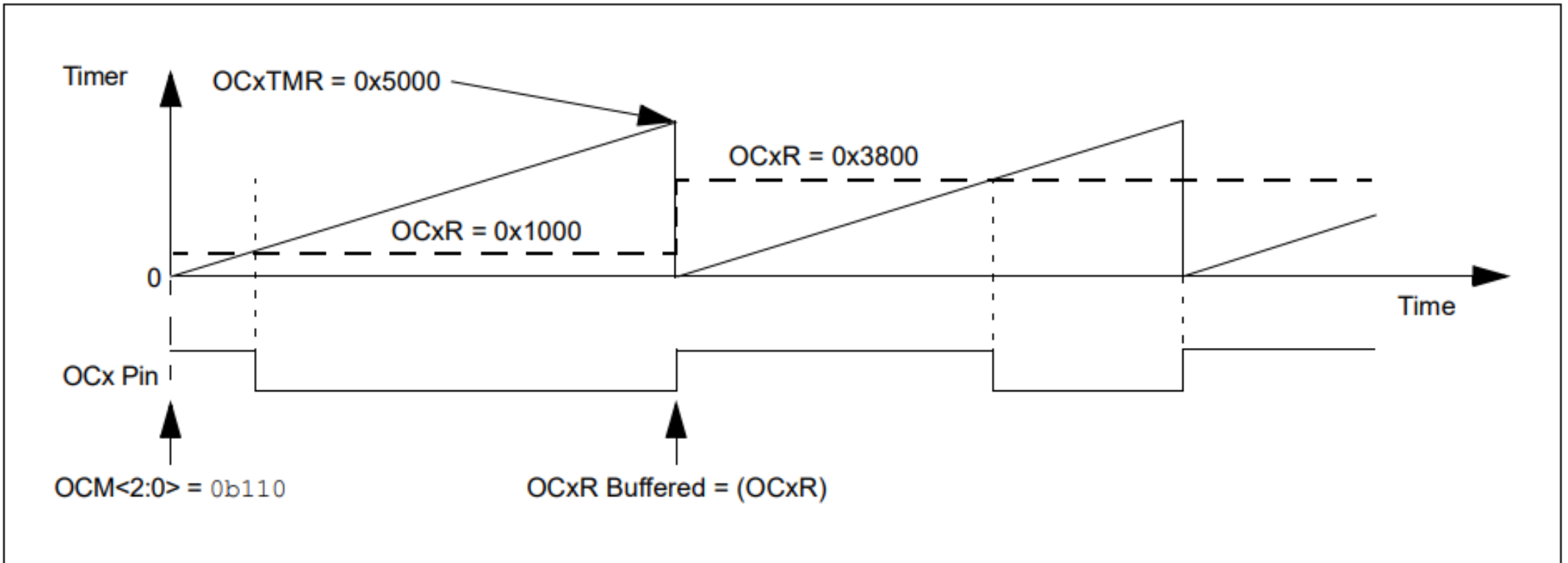


# Output compare alkalmazása

- Pulzusszélesség-moduláció (PWM)

» PWM mód

**Figure 3-16: Edge-Aligned PWM Mode with  $OCR > 0$  – At Module Initialization,  $OCxR = 0x1000$ ,  $OCxRS = 0x5000$**



- Figure 3-16: Edge-Aligned PWM Mode with OCR > 0 – At Module Initialization, OCxR = 0x1000, OCxRS = 0x5000**



- Fix periódus, változó kitöltési tényező:
  - Egyszerű analóg kimenet
    - Külső R-C szűrő „kiátlagolja” a jelet
  - Valamilyen fizikai mennyiség szabályozása gyors ki-be kapcsolással
    - Kijelző/lámpa fényerő
    - Fűtés
    - Motor sebesség
  - Kódolt adattovábbítás (pl.: rádió csatornán)
- Fix kitöltési tényező, változó frekvencia
  - Jelzőhang/zene generálás
  - Kódolt adattovábbítás (pl.: rádió csatornán)

# Univerzális timer: MCCP

- Multiple output Capture Compare and PWM timer
- 1 bemenet, 6db kimenet, 4 órajel lehetőség
- Üzem módok:
  - 1db 32 bites timer
  - 2 független 16 bites timer
  - 32 bites Input capture
  - 6 db szinkronizált (közös időalappal működő) PWM kimenet

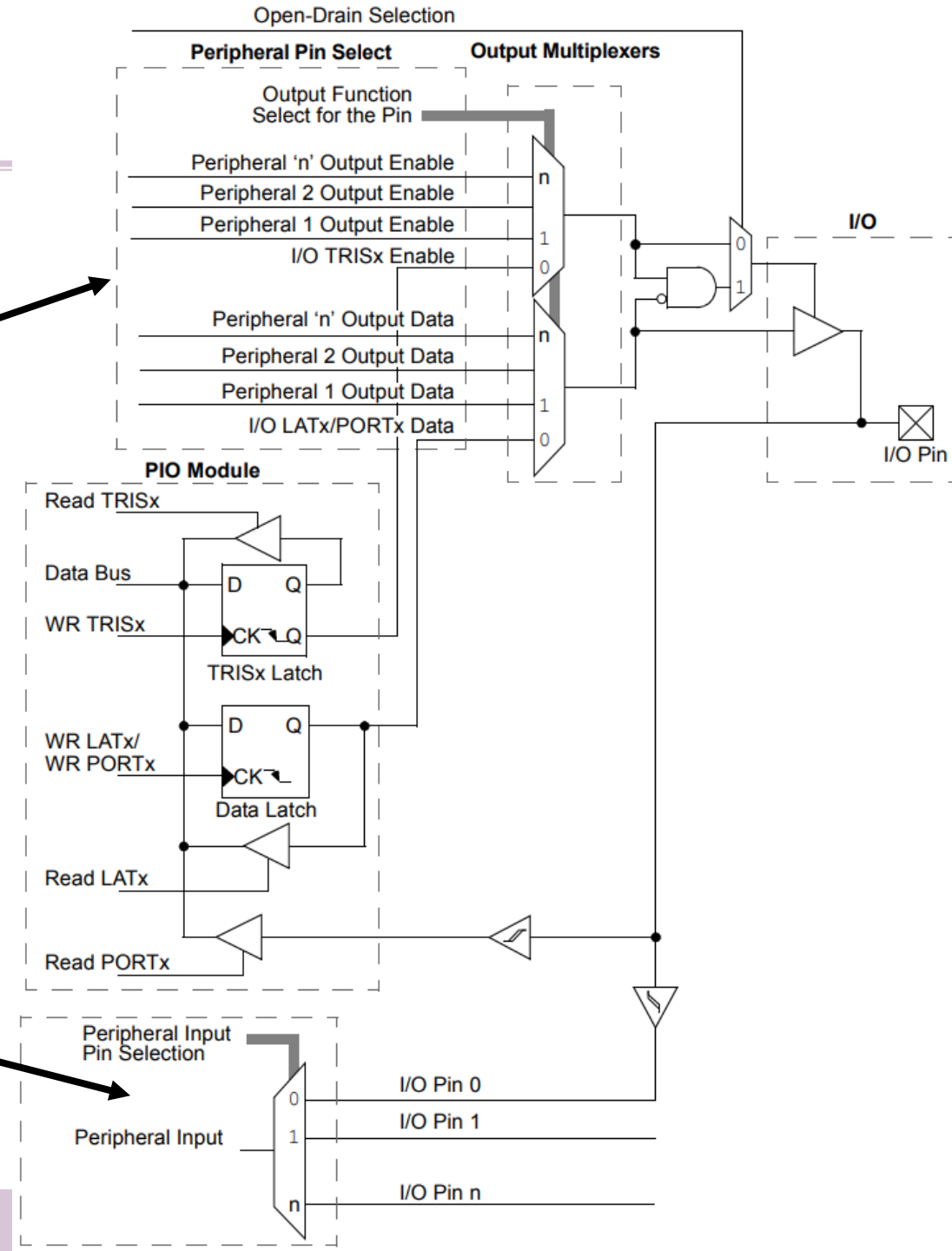
» *Nem fogjuk használni*

# Hogyan lehet a perifériák jeleit elérni? → PPS

- Sokkal több periféria van, mint portláb
- PPS: Peripheral Pin Select → multiplexer
- A belső perifériák jelei nincsenek fixen egy portbithez huzalozva, hanem konfigurálhatók
- Minden bemeneti- és kimeneti ponthoz ki kell választani (hozzá kell rendelni) a portlábát
- Előnyök:
  - a nem használt perifériák nem foglalnak fixen le egy portlábát
  - Sokkal több periféria lehet a mikrokontrollerben, mint portláb

Minden portlábhhoz (az eddig látottakon kívül) tartozik egy-egy multiplexer, ami valamelyik periféria valamelyik jelét kapcsolja a portlábra

Minden periféria majdnem minden bemeneti jeléhez szintén tartozik egy multiplexer, amelyik kiválasztja melyik portlábról jön a bemenet





# PPS – bemenetek választása

- Minden bemenethez tartozik egy regiszter valahány bitje
- A regiszterbe az adott port sorszámát kell beírni

Egy regiszterbe csak egy szám kerülhet

→ Nyilván több lábat ugyanoda kötni nincs értelme

→ Több bemenetre viszont köthetjük ugyanazt a lábat

→ PI: megmérhetjük ugyanazt a jelet többféleképpen

TABLE 11-6: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION)<sup>(1)</sup>

Input Name	Function Name	Register	Function Mapping Bits
Output Compare Trigger 1	OCTRIG1	RPINR0[5:0]	OCTRIG1R[5:0]
External Interrupt 1	INT1	RPINR0[13:8]	INT1R[5:0]
External Interrupt 2	INT2	RPINR1[5:0]	INT2R[5:0]
External Interrupt 3	INT3	RPINR1[13:8]	INT3R[5:0]
External Interrupt 4	INT4	RPINR2[5:0]	INT4R[5:0]
Output Compare Trigger 2	OCTRIG2	RPINR2[13:8]	OCTRIG2R[5:0]
Timer2 External Clock	T2CK	RPINR3[5:0]	T2CKR[5:0]
Timer3 External Clock	T3CK	RPINR3[13:8]	T3CKR[5:0]
Input Capture Mode 1	ICM1	RPINR5[5:0]	ICM1R[5:0]
Input Capture Mode 2	ICM2	RPINR5[13:8]	ICM2R[5:0]
Input Capture Mode 3	ICM3	RPINR6[5:0]	ICM3R[5:0]
Input Capture Mode 4	ICM4	RPINR6[13:8]	ICM4R[5:0]
Input Capture 1	IC1	RPINR7[5:0]	IC1R[5:0]
Input Capture 2	IC2	RPINR7[13:8]	IC2R[5:0]
Input Capture 3	IC3	RPINR8[5:0]	IC3R[5:0]
Output Compare Fault A	OCFA	RPINR11[5:0]	OCFAR[5:0]
Output Compare Fault B	OCFB	RPINR11[13:8]	OCFBR[5:0]
CCP Clock Input A	TCKIA	RPINR12[5:0]	TCKIAR[5:0]
CCP Clock Input B	TCKIB	RPINR12[13:8]	TCKIBR[5:0]
UART1 Receive	U1RX	RPINR18[5:0]	U1RXR[5:0]
UART1 Clear-to-Send	U1CTS	RPINR18[13:8]	U1CTSR[5:0]
UART2 Receive	U2RX	RPINR19[5:0]	U2RXR[5:0]
UART2 Clear-to-Send	U2CTS	RPINR19[13:8]	U2CTSR[5:0]
SPI1 Data Input	SDI1	RPINR20[5:0]	SDI1R[5:0]
SPI1 Clock Input	SCK1IN	RPINR20[13:8]	SCK1R[5:0]

# PPS – kimenetek választása

- Minden portlábhoz tartozik egy regiszter
- Minden funkcióhoz tartozik egy funkciókód
- A regiszterbe a funkciókódot kell írni

Egy regiszterbe csak egy szám kerülhet, de több helyre is írhatjuk ugyanazt a számot

→ Nyilván ugyanarra a lábra két kimenetet nem lehet kötni

→ Viszont ugyanazt a kimenetet több helyre is kihozhatjuk

**REGISTER 11-32: RP0R0: PERIPHERAL PIN SELECT OUTPUT REGISTER 0**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
bit 15				bit 8			

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7				bit 0			

**Legend:**

R = Readable bit      W = Writable bit      U = Unimplemented bit, read as '0'  
 -n = Value at POR      '1' = Bit is set      '0' = Bit is cleared      x = Bit is unknown

bit 15-14      **Unimplemented:** Read as '0'

bit 13-8      **RP1R[5:0]:** RP1 Output Pin Mapping bits  
 Peripheral Output Number n is assigned to pin, RP1 (see [Table 11-7](#) for peripheral function numbers).

bit 7-6      **Unimplemented:** Read as '0'

bit 5-0      **RP0R[5:0]:** RP0 Output Pin Mapping bits  
 Peripheral Output Number n is assigned to pin, RP0 (see [Table 11-7](#) for peripheral function numbers).

**TABLE 11-7: SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)**

Output Function Number	Function	Output Name
0	None (Pin Disabled)	—
1	C1OUT	Comparator 1 Output
2	C2OUT	Comparator 2 Output
3	U1TX	UART1 Transmit
4	U1RTS	UART1 Request-to-Send
5	U2TX	UART2 Transmit
6	U2RTS	UART2 Request-to-Send
7	SDO1	SPI1 Data Output
8	SCK1OUT	SPI1 Clock Output
9	SS1OUT	SPI1 Slave Select Output
10	SDO2	SPI2 Data Output
11	SCK2OUT	SPI2 Clock Output
12	SS2OUT	SPI2 Slave Select Output
13	OC1	Output Compare 1
14	OC2	Output Compare 2
15	OC3	Output Compare 3

# PPS – portlábak sorszámai

Sajnos a portlábak sorszáma (amit a PPS-hez használunk) nem egyezik meg az IO portok nevével

- Pl: Port C 6. bitje → RP22
- Valamennyire logikus azért, az adott kontrolleren a számozás a B port-tal kezdődik (PortB 0 → RP0, PortB15 → RP15) utána a C port-tal folytatódik, majd az A port lábai jönnek össze-vissza... ☹

→ mindig meg kell nézni a táblázatban...

**TABLE 6: COMPLETE PIN FUNCTION DESCRIPTIONS (PIC24FJ256GA705 TQFP)**

Pin	Function	Pin	Function
1	C1INC/C2INC/C3INC/TMPRN/ <b>RP9</b> /SDA1/T1CK/CTED4/PMD3/RB9	25	AN4/C1INB/ <b>RP2</b> /SDA2/CTED13/RB2
2	<b>RP22</b> /PMA1/PMALH/RC6	26	AN5/C1INA/ <b>RP3</b> /SCL2/CTED8/RB3
3	<b>RP23</b> /PMA0/PMALL/RC7	27	AN10/ <b>RP16</b> /PMBE1/RC0
4	<b>RP24</b> /PMA5/RC8	28	AN11/ <b>RP17</b> /PMA15/PMCS2/RC1
5	<b>RP25</b> /CTED7/PMA6/RC9	29	AN12/ <b>RP18</b> /PMACK1/RC2
6	Vss	30	Vdd
7	VCAP	31	Vss
8	<b>RP129</b> /RA11	32	<b>RP131</b> /RA13
9	PGD2/ <b>RP10</b> /OCM1C/CTED11/PMD2/RB10	33	OSCI/CLKI/C1IND/RA2
10	PGC2/REF11/ <b>RP11</b> /CTED9/PMD1/RB11	34	OSCO/CLKO/C2IND/RA3
11	AN8/LVDIN/ <b>RP12</b> /PMD0/RB12	35	TDO/PMA8/RA8
12	AN7/C1INC/ <b>RP13</b> /OCM1D/CTPLS/PMRD/PMWR/RB13	36	SOSCI/ <b>RP4</b> /RB4
13	TMS/ <b>RP28</b> /PMA2/PMALU/RA10	37	SOSCO/PWRLCLK/RA4
14	TCK/PMA7/RA7	38	TDI/PMA9/RA9
15	CVREF/AN6/C3INB/ <b>RP14</b> /CTED5/PMWR/PMENB/RB14	39	AN13/ <b>RP19</b> /PMBE0/RC3
16	AN9/C3INA/ <b>RP15</b> /CTED6/PMA14/PMCS/PMCS1/RB15	40	<b>RP20</b> /PMA4/RC4
17	AVss	41	<b>RP21</b> /PMA3/RC5
18	AVDD	42	Vss
19	MCLR	43	VDD
20	<b>RP130</b> /RA12	44	<b>RP132</b> /RA14
21	VREF+/CVREF+/AN0/C3INC/ <b>RP26</b> /CTED1/RA0	45	PGD3/ <b>RP5</b> /ASDA1/OCM1E/PMD7/RB5
22	VREF-/CVREF-/AN1/C3IND/ <b>RP27</b> /CTED2/RA1	46	PGC3/ <b>RP6</b> /ASCL1/OCM1F/PMD6/RB6
23	PGD1/AN2/CTCMP/C2INB/ <b>RP0</b> /RB0	47	<b>RP7</b> /OCM1A/CTED3/PMD5/INT0/RB7
24	PGC1/AN1-/AN3/C2INA/ <b>RP1</b> /CTED12/RB1	48	<b>RP8</b> /SCL1/OCM1B/CTED10/PMD4/RB8

**Legend:** **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select (PPS) functions.

# Példa: kijelző fényerő változtatása

A PortC 3-ra kötött jel kitöltési tényezőjével arányos egy LED kijelző fényereje.

- Írjunk szubrutint (PWM\_init) amely inicializálja az OC1 modult a megfelelő jel előállítására.
- Írjunk szubrutint (Disp\_brightness), amely képes a fényerő változtatására egy 0..999 közti skálán. A fényerő értékét a W0 regiszterben kapja.

# Példa: kijelző fényerő változtatása

A PortC 3-ra kötött jel kitöltési tényezőjével arányos egy LED kijelző fényereje.

- Írjunk szubrutint (PWM\_init) amely inicializálja az OC1 modult a megfelelő jel előállítására.

TABLE 11-7: SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)

Output Function Number	Function	Output Name
0	None (Pin Disabled)	—

REGISTER 11-41: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 15				bit 8			

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7				bit 0			

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP19R[5:0]:** RP19 Output Pin Mapping bits

Peripheral Output Number n is assigned to pin, RP19 (see Table 11-7 for peripheral function numbers).

Output	38	TDI/PMA9/RA9
Output	39	AN13/RP19/PMBE0/RC3
to-Ser	40	RP20/PMA4/RC4

to-Send

to-Send

to-Send

to-Send

to-Send

to-Send

to-Send

to-Send

to-Send

to-Send

to-Send

RPOR9 regiszter felső bájtjába kell beírni a 13-as számot

# Példa: kijelző fényerő változtatása

A PortC 3-ra kötött jel kitöltési tényezőjével arányos egy LED kijelző fényereje.

- Írunk szubrutint (PWM\_init) amely inicializálja az OC1 modult a megfelelő jel előállítására.

PWM\_init:

```
mov #13, W0
mov.b WREG, RPOR9H ;RPOR9 felső bajtja a 19-es láb funkciója, 13 a pwm OC1

mov #0x001f, w0
mov w0, OC1CON2
```

**REGISTER 15-2: OCxCON2: OUTPUT COMPARE x CONTROL REGISTER 2**

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1 <sup>(3)</sup>	DCB0 <sup>(3)</sup>	OC32
bit 15				bit 8			
R/W-0	HS/R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7				bit 0			

**SYNCSEL[4:0]: Trigger/Synchronization Source Selection bits**

11111 = OCx Sync out<sup>(1)</sup>

11110 = OCTRIG1 pin

11101 = OCTRIG2 pin

11100 = CTMU trigger<sup>(2)</sup>

11011 = A/D interrupt<sup>(2)</sup>

11010 = CMP3 trigger<sup>(2)</sup>

11001 = CMP2 trigger<sup>(2)</sup>

11000 = CMP1 trigger<sup>(2)</sup>

10111 = Not used

10110 = M CCP4 IC/OC interrupt

10101 = M CCP3 IC/OC interrupt

10100 = M CCP2 IC/OC interrupt

10011 = M CCP1 IC/OC interrupt

10010 = IC3 interrupt<sup>(2)</sup>

10001 = IC2 interrupt<sup>(2)</sup>

10000 = IC1 interrupt<sup>(2)</sup>

01111 = Not used

01110 = Not used

01101 = Timer3 match event

01100 = Timer2 match event (default)

01011 = Timer1 match event

01010 = Not used

01001 = Not used

01000 = Not used

00111 = M CCP4 Sync/Trigger out

00110 = M CCP3 Sync/Trigger out

00101 = M CCP2 Sync/Trigger out

00100 = M CCP1 Sync/Trigger out

00011 = Not used

00010 = OC3 Sync/Trigger out<sup>(1)</sup>

00001 = OC1 Sync/Trigger out<sup>(1)</sup>

00000 = Off, Free-Running mode with no synchronization and rollover at FFFFh



# Példa: kijelző fényerő változtatása

A PortC 3-ra kötött jel kitöltési tényezőjével arányos egy LED kijelző fényereje.

- Írjunk szubrutint (PWM\_init) amely inicializálja az OC1 modult a megfelelő jel előállítására.

PWM\_init:

```
mov #13, W0
mov.b WREG, RPOR9H ;RPOR9 felső bajtja a 19-es láb funkciója, 13 a pwm OC1
```

```
mov #0x001f, w0
mov w0, OC1CON2
```

```
mov #0x1C06, w0
mov w0, OC1CON1
```

bit 12-10

**OCTSEL[2:0]: Output Compare x Timer Select bits**

111 = Peripheral clock (Fcy)

110 = Reserved

101 = Reserved

100 = Timer1 clock (only synchronous clock is supported)

011 = Unimplemented

010 = Unimplemented

001 = Timer3 clock

000 = Timer2 clock

bit 2-0

**OCM[2:0]: Output Compare x Mode Select bits<sup>(1)</sup>**

111 = Center-Aligned PWM mode on OCx<sup>(2)</sup>

110 = Edge-Aligned PWM mode on OCx<sup>(2)</sup>

101 = Double-Compare Continuous Pulse mode: Initializes the OCx pin low; toggles the OCx state continuously on alternate matches of OCxR and OCxRS

100 = Double-Compare Single-Shot mode: Initializes the OCx pin low; toggles the OCx state on matches of OCxR and OCxRS for one cycle

011 = Single Compare Continuous Pulse mode: Compare events continuously toggle the OCx pin

010 = Single Compare Single-Shot mode: Initializes OCx pin high; compare event forces the OCx pin low

001 = Single Compare Single-Shot mode: Initializes OCx pin low; compare event forces the OCx pin high

000 = Output compare channel is disabled

**REGISTER 15-1: OCxCON1: OUTPUT COMPARE x CONTROL REGISTER 1**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2 <sup>(2)</sup>	ENFLT1 <sup>(2)</sup>
bit 15				bit 8			

R/W-0	HSC/R/W-0	HSC/R/W-0	HSC/R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0 <sup>(2)</sup>	OCFLT2 <sup>(2,3)</sup>	OCFLT1 <sup>(2,4)</sup>	OCFLT0 <sup>(2,4)</sup>	TRIGMODE	OCM2 <sup>(1)</sup>	OCM1 <sup>(1)</sup>	OCM0 <sup>(1)</sup>
bit 7				bit 0			

# Példa: kijelző fényerő változtatása

A PortC 3-ra kötött jel kitöltési tényezőjével arányos egy LED kijelző fényereje.

- Írjunk szubrutint (PWM\_init) amely inicializálja az OC1 modult a megfelelő jel előállítására.

PWM\_init:

```

mov #13, w0
mov.b WREG, RPOR9H ;RPOR9 felső bajtja a 19-es láb funkciója, 13 a pwm OC1

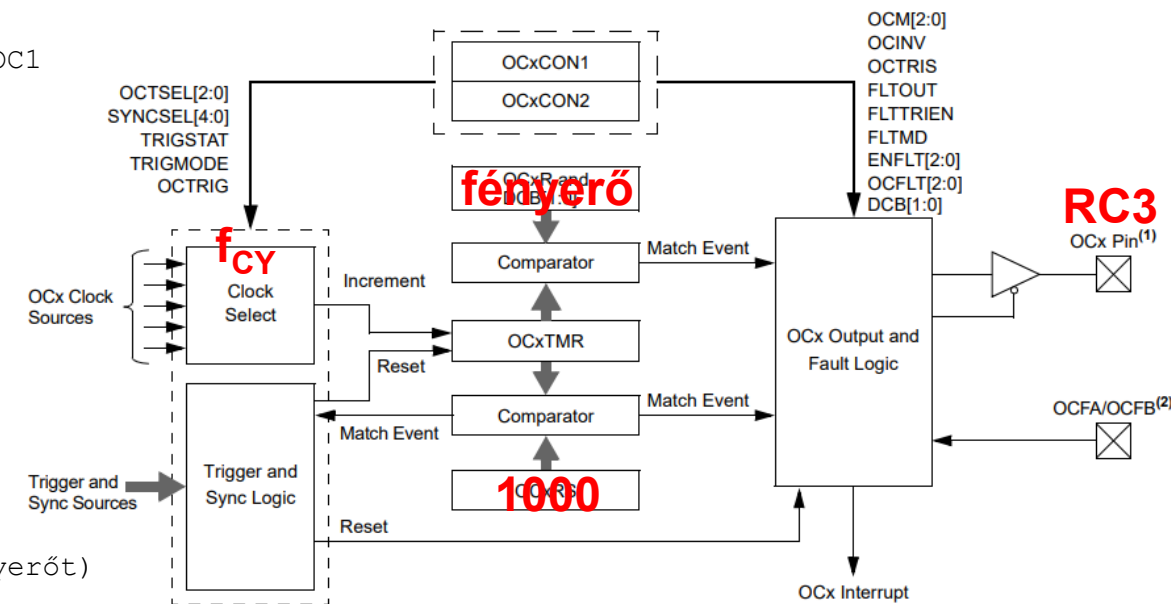
mov #0x001f, w0
mov w0, OC1CON2

mov #0x1C06, w0
mov w0, OC1CON1

mov #1000, w0
mov w0, OC1RS ;a periódus legyen 1000

mov #0, w0
mov w0, OC1R ;kezdetben legyen 0 (de választhatnánk másik kezdeti fényerőt)

return
  
```





# Példa: kijelző fényerő változtatása

A PortC 3-ra kötött jel kitöltési tényezőjével arányos egy LED kijelző fényereje.

- Írjunk szubrutint (Disp\_brightness), amely képes a fényerő változtatására egy 0..999 közti skálán. A fényerő értékét a W0 regiszterben kapja.

Disp\_brightness:

```
mov w0, OC1R ; persze megnézhethetnénk, hogy nagyobb-e mint 999  
return
```