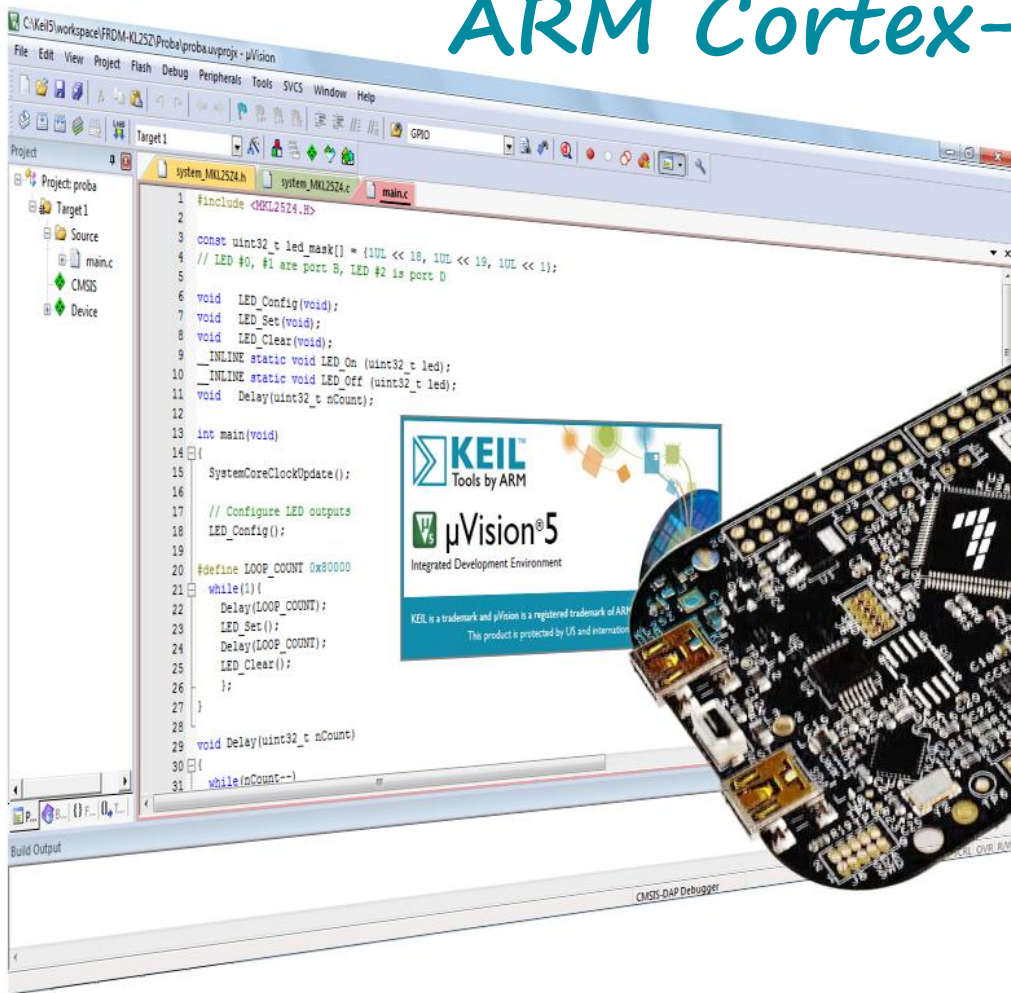


ARM Cortex-M0+ mikrovezérlő programozása KEIL MDK 5 környezetben



Cortex
Intelligent Processors by ARM®

5. Időzítők, számlálók – 2. rész

Felhasznált anyagok, ajánlott irodalom

- ❑ Joseph Yiu: **The Definitive Guide to ARM® Cortex®-M0 and Cortex-M0+ Processors** (2nd Ed.)
- ❑ Muhammad Ali Mazidi, Shujen Chen, Sarmad Naimi, Sepehr Naimi: **Freescale ARM Cortex-M Embedded Programming**
- ❑ ARM University Program: **Course/Lab Material for Teaching Embedded Systems/MCUs** (for the FRDM-KL25Z board)
- ❑ Trevor Martin: **The Designer's Guide to the Cortex-M Processor Family**
- ❑ Freescale: [MKL25Z128VLK4 MCU datasheet](#)
- ❑ Freescale: [KL25 Sub-Family Reference Manual](#)
- ❑ Freescale: [FRDM-KL25Z User Manual](#)
- ❑ Freescale: [FRDM-KL25Z Pinout](#)

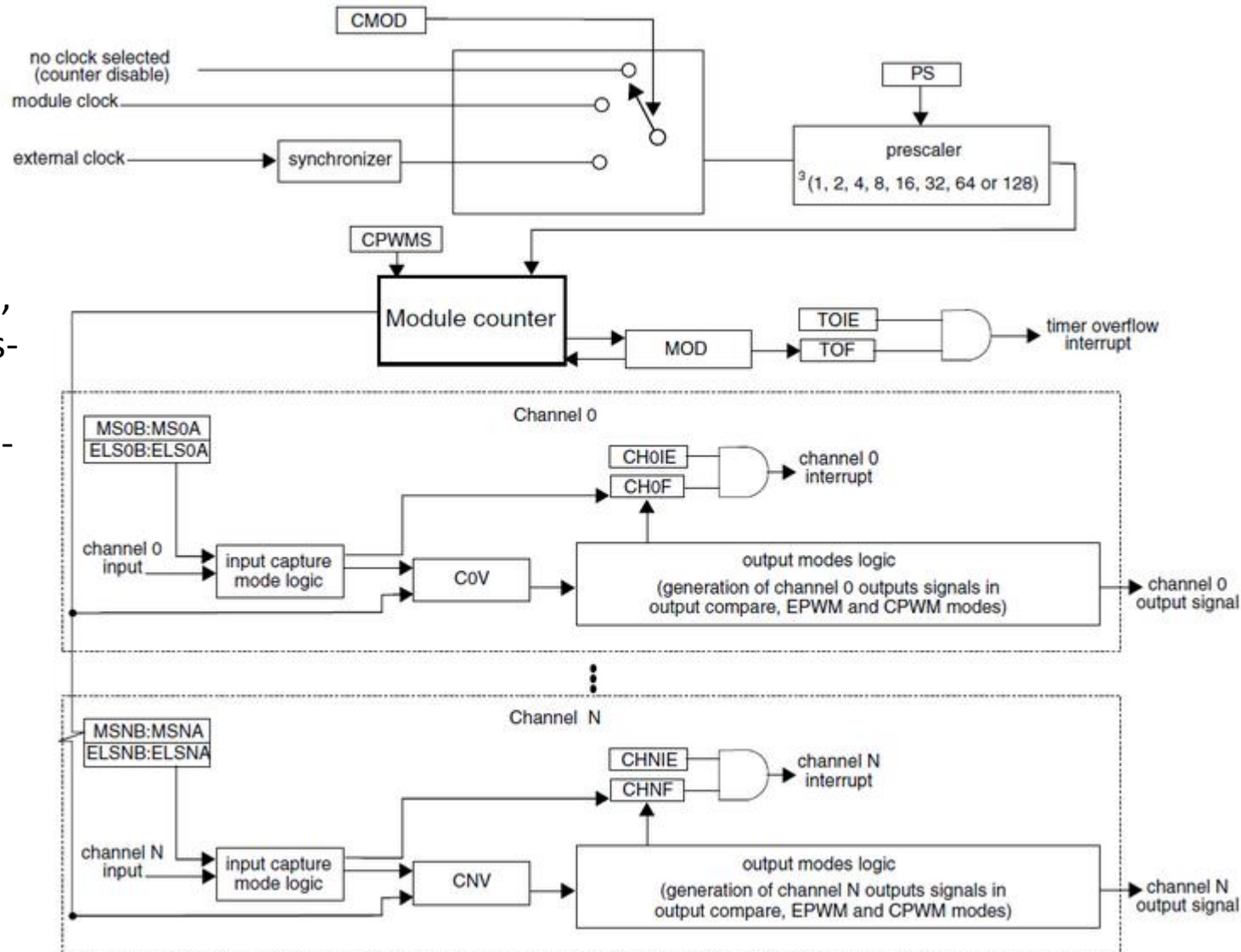
Időzítők/számlálók az MKL25Z128VLK4 mikrovezérlőben

- **Systick** 24-bites (vissza)számláló, az **ARM Cortex-M** mikrovezérlő mag része. Többnyire az operációs rendszerek (pl. RTOS) használják a feladatok ütemezéséhez.
- **TPM0, TPM1, TPM2** általános célú 16-bites számlálók/időzítők, amelyekhez 2-6 db impulzus-szélesség modulációra (PWM), bemeneti jelrögzítésre (Input Capture), vagy digitális komparálásra (Output Compare használható csatorna is tartozik).
- **Periodic Interrupt Timer (PIT)** 2 x 32 bites időzítővel rendelkezik, amelyek egy 64 bites időzítővé is összekapcsolhatók. Periodikus megszakítások előállítására szolgál, de más célra, pl. a legutolsó RESET óta eltelt folyó időt mérésére is használhatjuk.
- **Low-Power Timer (LPTMR)** kismegfogyasztású 16-bites időzítő/számláló. Mindegyik energiatakarékos módban üzemképes, külső események számlálására vagy belső órajellel időzítésre, vagy programozott ébresztésre használható.
- **Real-Time Clock (RTC)** modul, melyben található egy 32-bites másodperc számláló, egy 16-bites előszámláló, egy 16-bites időkompenzációs regiszter és egy 32-bites regiszter az ébresztés beállítására. A **FRDM-KL25Z** kártya esetén a használatot némiképp korlátozza, hogy az RTC órajelét az **OpenSDA** áramkör biztosítja, s az **MKL25Z128VLK4** mikrovezérlő **PTC1** kivezetését (**RTC_CLKIN**) lefoglalja.

A TPM modulok csatornáí

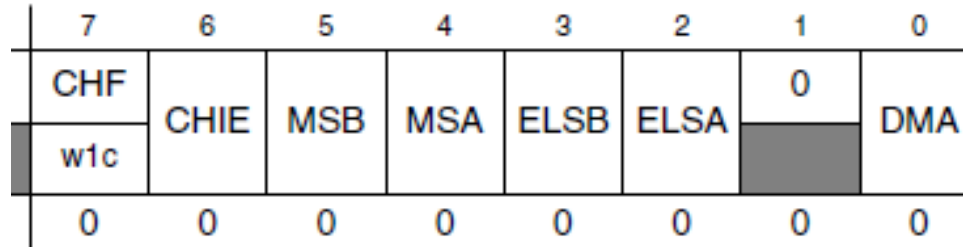
Az **MKL25Z128VLK4** mikrovezérlő általános célú időzítőihez több impulzus-szélesség modulációra (PWM), bemeneti jelrögzítésre (Input Capture), vagy digitális komparálásra (Output Compare) használható csatorna is kapcsolódik:

- TPM0** 6 db,
- TPM1** 2 db,
- TPM2** 2 db csatornával rendelkezik.



Csatorna konfigurációs és adatregiszterek

❖ Konfigurációs regiszter: **TPM_x_CnSC**



- CHF – '1'-be áll, amikor az esemény bekövetkezik
 - CHIE – megszakításkérés engedélyezés
 - MSB:MSA – üzemmód választás
 - ELSB:ELSA – él vagy jelszint választás
 - DMA – DMA átvitel engedélyezése
-
- Adatregiszter: **TPM_x_CnV**
 - 16-bites érték az összehasonlításhoz, vagy bemeneti jelrögzítés eredménye

x – az időzítő sorszáma ($x = 0, 1, 2$)

n – a csatorna sorszáma ($n = 0, \dots, 5$)

Beállítható csatorna üzemmódok

CPWMS	MSB:MSA	ELSB:ELSA	Üzemmód	Konfiguráció
x	00	00	nincs	csatorna letiltva
x	01/10/11	00	komparálás szoftveres vezérléssel	nincs hardveres kimenetvezérlés
0	00	01	Bemeneti jelrögzítés (Input Capture)	rögzítés felfutó élnél
		10		rögzítés lefutól élnél
		11		rögzítés mindkét élnél
	01	01	kimenetvezérlés digitális összehasonlítással (Output Compare)	kimenet átbillentése egyezéskor
		10		kimenet törlése egyezéskor
		11		kimenet '1'-be állítása egyezéskor
	10	10	Él igazított jelű PWM	kimenet törlése egyezéskor
		x1		kimenet '1'-be állítása egyezéskor
	11	10	impulzus módú kimenetvezérlés digitális összehasonlítással (Output Compare)	negatív impulzus egyezéskor
		x1		pozitív impulzus egyezéskor
1	10	10	Középre igazított jelű PWM	pozitív impulzus jelalak
		x1		negatív impulzus jelalak

A TPMx_STATUS regiszter

A **TPMx_STATUS** állapotjelző regiszterben kigyűjtve megtalálható valamennyi csatorna **CHF** jelzőbitjének másolata és a **TPMx** számláló **TOF** túlcsoordulásjelző bitje is.

A **w1c** bejegyzések arra utalnak, hogy mindegyik jelzőbit '1' beírásával törölhető.

A működés szempontjából mindegy, hogy a jelzőbiteket ebben a gyűjtőregiszterben figyeljük/töröljük, vagy az egyes csatornaregiszterekben külön-külön

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R				0				TOF		0	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W								w1c			w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A csatornákhöz rendelhető kivezetések

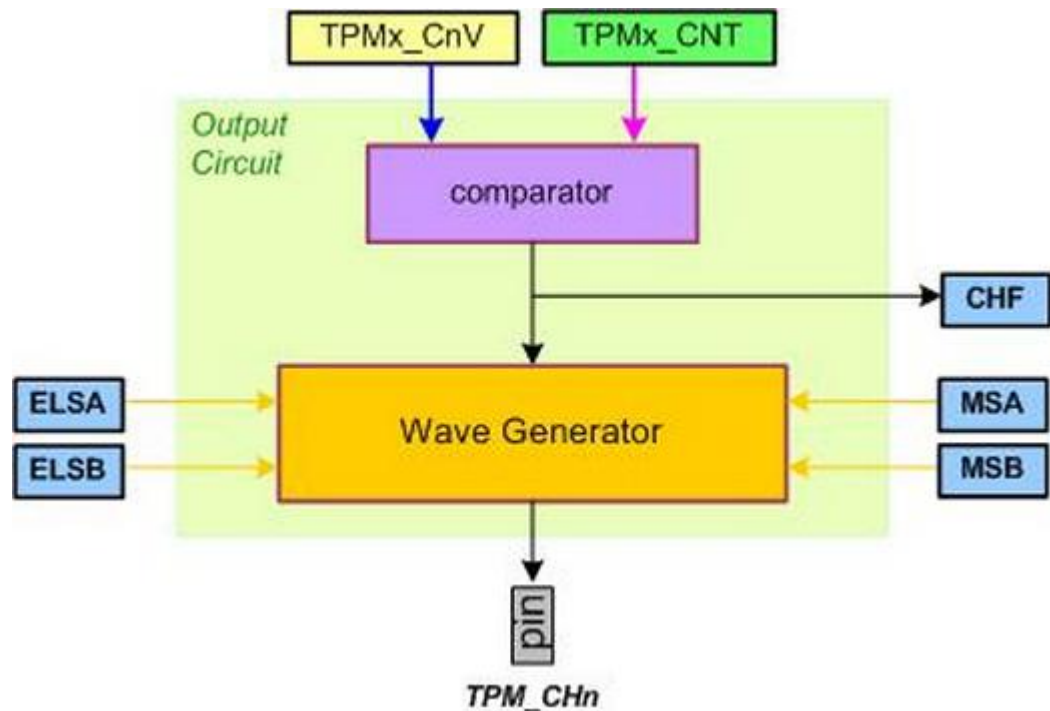
Mindegyik **TPM_x** modul mindegyik csatornájához külön kivezetés rendelhető, s általában egynél több lehetőség közül választhatunk. A választás a megfelelő **PORT_x_PCR_n** portvezérlő regiszter **MUX** bitcsoportjába írt számmal történik (**Alt3** vagy **Alt4** funkciót kell választanunk, ahogy az alábbi táblázat, illetve a [FRDM-KL25Z Pinouts](#) dokumentum jelzi). A hardverütközések kiszűrésével a **FRDM-KL25Z** kártya esetében az alábbi lehetőségek közül választhatunk:

Időzítő	Csatorna	Kivezetés
TPM0	CH0	PTC1 (4), PTD1 (4)
	CH1	PTA4 (3), PTC2 (4), PTD1 (4)
	CH2	PTA5 (3), PTC3 (4), PTD2 (4), PTE29 (3)
	CH3	PTC4 (4), PTD3 (4), PTE30 (3)
	CH4	PTC8 (3), PTD4 (4), PTE31 (3)
	CH5	PTC9 (3), PTD5 (4)
TPM1	CH0	PTA12 (3), PTB0 (3), PTE20 (3)
	CH1	PTA13 (3), PTB1 (3), PTE21 (3)
TPM2	CH0	PTA1 (3), PTB2 (3), PTB18 (3), PTE22 (3)
	CH1	PTA2 (3), PTB3 (3), PTB19 (3), PTE23 (3)

A zárójelbe írt számok az üzemmód beállításához választandó **Alt_n** mód sorszámát jelzik.

Az Output Compare mód

Output Compare (kimenetvezérlés digitális összehasonlítással) módban (**MSB:MSA** = 01 vagy 11), a **TPMx_CNT** számláló 0-tól felfelé számlál, s minden léptetéskor összehasonlításra kerül a csatorna **TPMx_CnV** adatregiszterének tartalmával. Egyezéskor a csatorna **CHF** jelzőbitje '1'-be áll, s a csatornához rendelt kimenet állapota megváltozik az **ELSA:ELSA**-ban kiválasztott módon. A számlálás folytatódik a **TPMx_MOD** regiszterben megadott értékig, majd a számláló túlcsoordul és újratekzi a számlálást.



Az Output Compare mód

Kimenetvezérlési lehetőségek **MSB:MSA = 01** módban:

- ❖ **Kimenet átbillentése egyezéskor** (toggle output on match) - Amikor a **TPMx_CNT** számláló és a **TPMx_CnV** csatorna adatregiszter tartalma megegyezik, akkor a csatornához rendelt kimenet ellenkező állapotba billen.
- ❖ **Kimenet törlése egyezéskor** (clear output on match) - Amikor az üzemmódot beállítjuk, a kimenet '1'-be áll, majd amikor a **TPMx_CNT** számláló és a **TPMx_CnV** csatorna adatregiszter tartalma megegyezik, akkor a csatornához rendelt kimenet '0' állapotba billen.
- ❖ **Kimenet beállítása egyezéskor** (set output on match) - Amikor az üzemmódot beállítjuk, a kimenet '0'-be áll, majd amikor a **TPMx_CNT** számláló és a **TPMx_CnV** csatorna adatregiszter tartalma megegyezik, akkor a csatornához rendelt kimenet '1' állapotba billen.

Kimenetvezérlési lehetőségek **MSB:MSA = 11** módban:

- ❖ **Negatív impulzus egyezéskor** - Amikor az üzemmódot beállítjuk, a kimenet '1'-be áll, majd amikor a **TPMx_CNT** számláló és a **TPMx_CnV** csatorna adatregiszter tartalma megegyezik, akkor a csatornához rendelt kimenet '0' állapotba billen egy órajel idejére.
- ❖ **Pozitív impulzus egyezéskor** - Amikor az üzemmódot beállítjuk, a kimenet '0'-ba áll, majd amikor a **TPMx_CNT** számláló és a **TPMx_CnV** csatorna adatregiszter tartalma megegyezik, akkor a csatornához rendelt kimenet '1' állapotba billen egy órajel idejére.

Program5_8: kimenet billegtetése OC módban

A **TPM0_CH1** csatornát használjuk **Output Compare toggle** módban periodikus kimenő jel előállítására a **PTD1** kimeneten (kék LED). A **TPM** órajel = 20.97 MHz / 128 lesz. A **PTD1** kimenetet (kék LED) **TPM0_CH1** módba állítjuk. Átbillenéskor **TPM0_C1V** értékét 32766-vel megnöveljük, így a következő egyezés $32766 * 128 / 20\,970\,000 \text{ Hz} = \text{kb. } 200 \text{ ms}$ múlva lesz.

A konfigurálás lépései:

1. Engedélyezzük a **D** portot (SIM->SCGC5: bit12 = 1)
2. A **PTD1** kivezetés **TPM0CH1** (Alt4) funkcióját választjuk (PORTD->PCR[1]: MUX = 4)
3. Engedélyezzük a **TPM0** modult (SIM->SCGC6: bit24 = 1)
4. Konfiguráluk a **TPM** modulok közös órajelét (SIM->SOPT2: TPMSRC = 01, PLLFLLSEL = 0)
5. Letiltjuk a **TPM0** modul számlálóját a konfigurálás idejére (TPM0->SC = 0)
6. Beállítjuk **TPM0** előszámlálóját (TPM0_SC: PS = 111)
7. Beállítjuk **TPM0** modulo regiszterét (TPM0_MOD = 0xFFFF)
8. Beállítjuk a **TPM0_C1SC** vezérlő regiszterben az OC toggle módot (MSB:MSA=01, ELSB:ELSA=01)
9. Töröljük a **TPM0_C1SC** regiszterben a **CHF** jelzőt
10. Beállítjuk az első késleltetési ciklust (TPM0_C1V = 32766)
11. Elindítjuk a számlálót (TPM0_SC: CMOD = 01)

A végtelen ciklusban ismételt lépések:

1. Várunk a **TPM0 CH1** csatorna **CHF** jelzőjére (TPM0_C1SC: CHF = 1 eseményre várunk)
2. Töröljük **TPM0 CH1** csatorna **CHF** jelzőjét (TPM0_C1SC: a CHF = 1 beírás töröl)
3. Új időpont beállítása (TPM0_C1V értékét növeljük meg 32766-tal)

```

/* Program5_8: A PTD1 kimenent billegtetése OC módban
 *
 * A program forrása: Mazidi et al., Freescale ARM Cortex-M Embedded Programming
 * http://www.microdigitaled.com/ARM/Freescale_ARM/Code/Freescale_ARM_codes.htm
 */
#include <MKL25Z4.H>

int main (void) {
    SIM->SCGC5 |= 0x1000;           // A D port engedélyezése
    PORTD->PCR[1] = 0x400;        // PTD1 legyen TPM0CH1 kimenet

    SIM->SCGC6 |= 0x01000000;     // A TPM0 időzítő engedélyezése
    SIM->SOPT2 |= 0x01000000;     // MCGFLLCLK legyen a TPM órajel

    TPM0->SC = 0;                 // Konfiguráláshoz letiltjuk az időzítőt
    TPM0->SC = 0x07;              // Előosztási arány 128 legyen
    TPM0->MOD = 0xFFFF;          // Maximális modulo érték
    TPM0->CNT = 0;                // A számláló törlése
    TPM0->CONTROLS[1].CnSC = 0x14; // OC Toggle mód
    TPM0->CONTROLS[1].CnSC |= 0x80; // CHF jelző törlése
    TPM0->CONTROLS[1].CnV = 32766; // 200 ms múlva legyen egyezés
    TPM0->SC |= 0x08;            // TPM0 számláló indítása

    while (1) {
        while(!(TPM0->CONTROLS[1].CnSC & 0x80)) { } // CHF jelzőre várunk
        TPM0->CONTROLS[1].CnSC |= 0x80;           // CHF jelző törlése
        TPM0->CONTROLS[1].CnV += 32766;          // 200 ms múlva újabb egyezés
    }
}

```

OC_flash: LED villantás

Az alábbi programban a **TPM0** modul **CH1** csatornáját használjuk kb. 100 ms szélességű impulzusok keltésére **Output Compare** módban, s a **PTD1** kivezetést (kék LED) villogtatjuk vele. Ahhoz, hogy az impulzus elegendően hosszú (100 ms) lehessen, alacsony frekvenciájú órajelet, a 32.768 kHz-es belső oszcillátor jelét használjuk fel **TPM** órajelként. A következő villantáshoz megvárjuk a szabadonfutó módban számláló **TPM0_CNT** túlcsordulását, így a villantások között 2 s telik el. A közös anódú RGB LED vezérlése negatív logikájú jelekkel történik, ezért a **Kimenet beállítása** (*Set output on match*) módot kell használnunk.

Megjegyzés: Ha ismételni akarjuk az impulzust, akkor először le kell tiltanunk az **Output Compare** módot, meg kell várni ennek érvényesülését (TOF flag vagy számláló változásának figyelésével), majd újra kell konfigurálnunk a kívánt **OC** módot.

A program végtelen ciklusában két esemény bekövetkeztére várunk:

- ❖ Amikor az egyezéskor (a 100 ms-os villantás végén) a **CHF1** jelzőbit bebillen, akkor töröljük a jelzőt és letiltjuk a csatorna **OC** módját.
- ❖ Amikor pedig túlcsordul a számláló, akkor töröljük a bebillen **TOF** jelzőbitet, majd újraindítjuk az **OC Kimenent beállítása** módot (elindítjuk az újabb villantást).

OC_flash: LED villantás

```
#include <MKL25Z4.H>
int main (void) {
    MCG->C1      |= 2;                // IRCLK lehessen MCGIRCLK forrás
    SIM->SCGC5   |= 0x1000;           // A D port engedélyezése
    PORTD->PCR[1] = 0x400;           // PTD1 pin legyen FTM0_CH1 módban
    SIM->SCGC6   |= 0x01000000;       // A TPM0 időzítő engedélyezése
    SIM->SOPT2   |= 0x03000000;       // MCGIRCLK legyen a TPM órajel
    TPM0->SC     = 0;                 // Konfiguráláshoz letiltjuk az időzítőt
    TPM0->MOD    = 0xFFFF0;          // Szabadonfutó mód, periódus = 2 sec
    TPM0->SC     |= 0x80;             // TOF törlése
    TPM0->SC     |= 0x08;             // Számláló engedélyezése (CMOD=01)
    TPM0->CONTROLS[1].CnSC = 0x1C;   // OC, Set high mód
    TPM0->CONTROLS[1].CnSC |= 0x80;   // CHF jelző törlése
    TPM0->CONTROLS[1].CnV = 3276;     // 100 ms múlva legyen egyezés
    while (1) {
        if(TPM0->STATUS&0x02) {      // CHF1-re vártunk...
            TPM0->STATUS |= 0x02;     // CHF1 törlése
            TPM0->CONTROLS[1].CnSC = 0; // Csatorna letiltása
        }
        if(TPM0->SC&0x80) {          // TOF-ra vártunk...
            TPM0->SC |= 0x80;         // TOF törlése
            TPM0->CONTROLS[1].CnV = 3276; // 100 ms múlva legyen egyezés
            TPM0->CONTROLS[1].CnSC = 0x1C; // OC, Set high mód
        }
    }
}
```

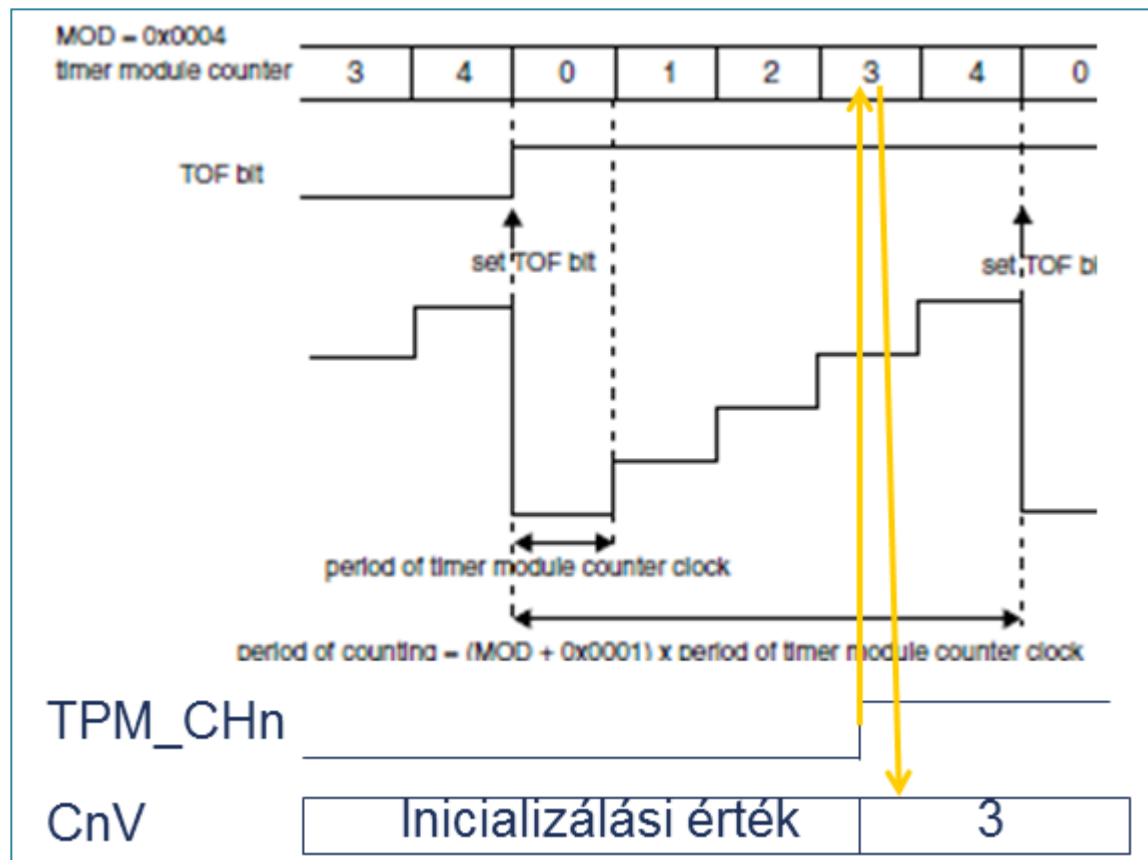
Bemeneti jelrögzítés mód

- ❖ Módbeállításhoz
CPWMS = 0,
MSnB:MSnA = 00

- ❖ **TPM_CHn** kivezetés él-érzékeny módban működik

- **ELSnB:ELSnA**
kiválasztja a felfutó élt (01)
vagy a lefutó élt (10)
vagy mindkettőt (11)

- ❖ Amikor a kiválasztott él megjelenik a **TPM_CHn** bemeneten, akkor:
 - A számláló pillanatnyi értéke eltárolódik a **CnV** regiszterbe
 - Megszakítás keletkezik (ha CHnIE = 1)
 - A CHFn jelzőbit '1'-be áll (3 órajel késleltetéssel)



Program5_9: periódusidő mérése

A programban bemeneti jelrögzés módban használjuk a **TPM2_CHO** csatornát, s a **PTA1** bemenethez rendeljük. Így az **UART0** bejövő jeleit vizsgálhatjuk, ha a bitrátát 9600 bps-re vagy kisebbre állítjuk. TPM órajelnek az előszámlálóval 4-gyel leosztott **MCGFLLCLK** órajelet használjuk: $20.97 \text{ MHz} / 4 = 5\,242\,500 \text{ Hz}$.

A bemeneti jelrögzés konfigurálásának lépései:

1. Engedélyezzük a **A** portot (SIM->SCGC5: bit9 = 1)
2. A **PTA1** kivezetés **TPM2CHO** (Alt3) funkcióját választjuk (PORTA->PCR[1]: MUX = 3)
3. Engedélyezzük a **TPM2** modult (SIM->SCGC6: bit26 = 1)
4. Konfiguráljuk a **TPM** modulok közös órajelét (SIM->SOPT2: TPMSRC = 01, PLLFLLSEL = 0)
5. Letiltjuk a **TPM2** modul számlálóját a konfigurálás idejére (TPM2->SC = 0)
6. Beállítjuk **TPM2** előszámlálóját (TPM2_SC: PS = 010)
7. Beállítjuk **TPM2** modulo regiszterét (TPM2_MOD = 0xFFFF)
8. Beállítjuk a **TPM2_COSC**-benben az „IC up” módot (MSB:MSA=00, ELSB:ELSA=01)
9. Töröljük a **TPM2_COSC** regiszterben a **CHF** jelzőt
10. Elindítjuk a számlálót (TPM2_SC: CMOD = 01)

A végtelen ciklusban ismételt lépések:

1. Várunk **TPM2 CHO** csatorna CHF jelzőjére (TPM2_COSC: CHF = 1 eseményre várunk)
2. Töröljük **TPM2 CHO** csatorna CHF jelzőjét (TPM2_COSC: a CHF = 1 beírás töröl)
3. Az új és az előző **TPM2_COV** értékek különbségének 11..9 bitjeit az RGB LED-en megjelenítjük


```
#include <MKL25Z4.H>
```

Program5_9 1. rész

```
int main (void) {
    unsigned short then = 0;
    unsigned short now = 0;
    unsigned short diff;

    //--- A GPIO kivezetések konfigurálása az RGB LED-hez ----
    SIM->SCGC5 |= 0x400;           // B port engedélyezése
    SIM->SCGC5 |= 0x1000;          // D port engedélyezése
    PORTB->PCR[18] = 0x100;        // PTB18 GPIO módba állítása
    PTB->PDDR |= 0x40000;          // PTB18 kimenet legyen
    PORTB->PCR[19] = 0x100;        // PTB19 GPIO módba állítása
    PTB->PDDR |= 0x80000;          // PTB19 kimenet legyen
    PORTD->PCR[1] = 0x100;         // PTD1 GPIO módba
    PTD->PDDR |= 0x02;            // PTD1 kimenet legyen

    //--- Timer konfigurálás -----
    SIM->SCGC5 |= 0x0200;          // A port engedélyezés
    PORTA->PCR[1] = 0x0300;        // PTA1 TPM2CH0 módba állítása

    SIM->SCGC6 |= 0x04000000;      // TPM2 engedélyezés
    SIM->SOPT2 |= 0x01000000;      // MCGFLLCLK legyen a TPM órajel

    TPM2->SC = 0;                  // Időzítő letiltása konfiguráláshoz
    TPM2->SC = 0x02;               // Előszámláló /4
    TPM2->MOD = 0xFFFF;           // Szabadonfutó mód
    TPM2->CONTROLS[0].CnSC = 0x04; // IC mód, felfutó élre
    TPM2->SC |= 0x08;             // Számláló engedélyezése
```

```
while (1) {
    while(!(TPM2->CONTROLS[0].CnSC & 0x80)); // CHF jelzőre várunk
    TPM2->CONTROLS[0].CnSC |= 0x80;          // CHF törlése
    now = TPM2->CONTROLS[0].CnV;            // A rögzített érték kiolvasása
    diff = now - then;                      // Az időtartam számítása
    then = now;                             // Az új érték lesz a vonatkoztatási pont

//--- LED-e beállítása az időtartam 11..9 bitjei szerint
    diff = diff >> 9;
    if (diff & 1)                          // Piros LED beállítása bit9 szerint
        PTB->PCOR = 0x40000; // LED = be
    else
        PTB->PSOR = 0x40000; // LED = ki

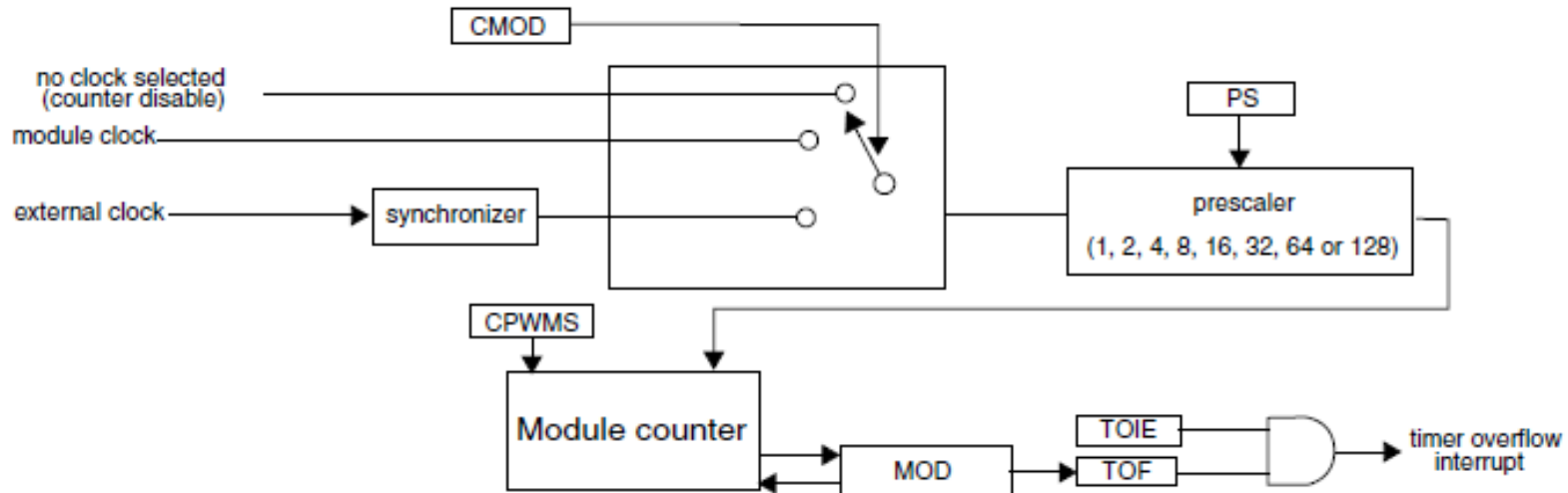
    if (diff & 2)                          // Zöld LED beállítása bit10 szerint
        PTB->PCOR = 0x80000;
    else
        PTB->PSOR = 0x80000;

    if (diff & 4)                          // Kék LED beállítása bit11 szerint
        PTD->PCOR = 0x02;
    else
        PTD->PSOR = 0x02;
}
}
```

Események számlálása

A **TPMx** modul számlálóként működik, ha a **TPMx_SC** vezérlő regiszter **CMOD** bitcsoportjában (bináris) 10-et írunk. Ebben az üzemmódban a számláló a kívülről érkező jel felfutó éleire számol. A bejövő jelet a mikrovezérlő a **TPM** órajelhez szinkronizálja, ezért a külső jelek számlálása esetén is konfigurálnunk kell a közös **TPM** órajelet.

Ahogy az ábrán is látható, a szinkronizált jel a **TPMx** modul előosztójára kerül, s a továbbiakban a számlálás ugyanúgy történik, mint a belső órajelek esetében. Amikor a számláló eléri a modulo regiszterben beállított felső határt, a **TPMx_SC** regiszterben bebillen a TOF jelzőbit, s a számlálás nulla értékről folytatódik.

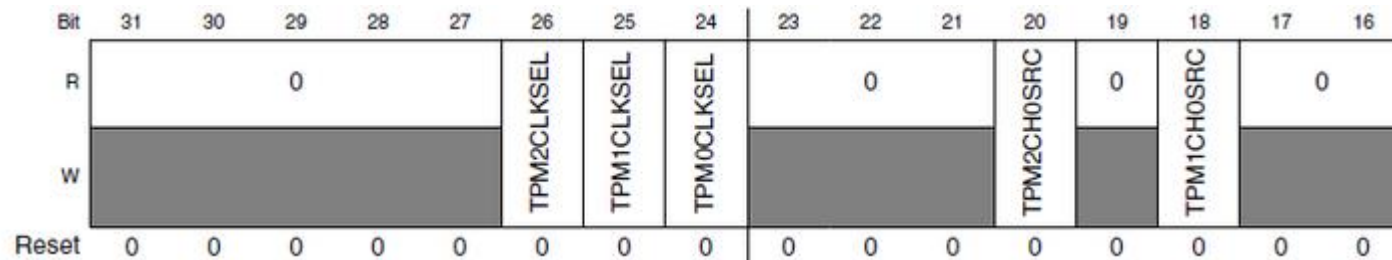


Bemenetválasztás

Külső események számlálásához az **MKL25Z128VLK4** mikrovezérlő három **TPM** moduljához csak két bemenet konfigurálható: **TPM_CLKIN0** és **TPM_CLKIN1**. Ezekhez az alábbi táblázat szerint rendelhetünk kivezetést a kiválasztott lábhoz tartozó **PORTx_PCRn** MUX bitjeinek beállításával (a halványan írt lábak hardverütközés miatt a **FRDM-KL25Z** kártya esetében nem választhatók).

TPM_CLKIN0	TPM_CLKIN1
PTA18	PTA19
PTB16	PTB17
PTC12	PTC13
PTE29	PTE30

A bemenetválasztás következő lépése: a rendszerintegrációs modul **SIM_SOPT4** regiszterében be kell állítani, hogy a használni kívánt **TPM_x** modul melyik bemenetet használja (**0**: TPM_CLKIN0, **1**: TPM_CLKIN1 választás).



Program5_10: Külső események számlálása

Ebben a programban külső eseményeket számlálunk a **PTC12** kivezetés, mint **TPM_CLKINO** bemenet használatával. A **TPM0** modul külső órajel bemenetét a mikrovezérlő **PTC12** kivezetéséhez rendeljük, amelyet **TPM_CLKINO** módba konfigurálunk.

A számlálót rendszeresen kiolvassuk, s a kiolvasott pillanatnyi értékének legalsó három bitjét kijelezzük az RGB LED-en.

TPM órajel gyanánt az alapértelmezett (20.97 MHz) **MCGFLLCLK** órajelet használjuk.

Megjegyzés: Ha összekötjük a **FRDM-KL25Z** kártya csatlakozóin a **PTA1** és **PTC12** pontokat, akkor a számítógépről küldött karakterek (**UART0 RX**) felfutó éleit számoltathatjuk.

Természetesen bármilyen más (alacsony frekvenciás) jelet is számláltathatjuk, vagy egy nyomógomb pergését is megfigyelhetjük...

Program5_10: Külső események számlálása

```
#include <MKL25Z4.H>

int main (void) {
    unsigned short count;
    //--- GPIO kimenetek inicializálása az RGB LED-hez -----
    SIM->SCGC5 |= 0x400;           // B port engedélyezése
    SIM->SCGC5 |= 0x1000;          // D port engedélyezése
    PORTB->PCR[18] = 0x100;        // PTB18 GPIO módba állítása
    PTB->PDDR |= 0x40000;          // PTB18 kimenet legyen
    PORTB->PCR[19] = 0x100;        // PTB19 GPIO módba állítása
    PTB->PDDR |= 0x80000;          // PTB19 kimenet legyen
    PORTD->PCR[1] = 0x100;         // PTD1 GPIO módba
    PTD->PDDR |= 0x02;             // PTD1 kimenet legyen
    //--- TPM0 és PCT12 inicializálása -----
    SIM->SCGC5 |= 0x0800;          // C port engedélyezése
    PORTC->PCR[12] = 0x400;        // PTC12 legyen TPM_CLKIN0 bemenet
    SIM->SOPT4 &= ~0x01000000;     // TPM_CLKIN0 bemenetet használjuk
    SIM->SCGC6 |= 0x01000000;      // TPM0 engedélyezése
    SIM->SOPT2 |= 0x01000000;      // MCGFLLCLK legyen a TPM órajel
    TPM0->SC = 0;                  // Időzítő letiltása konfiguráláshoz
    TPM0->SC = 0x80;               // Előszámláló /1 és TOF törlés
    TPM0->MOD = 0xFFFF;           // Szabadonfutó mód
    TPM0->CNT = 0;                 // Számláló törlése
    TPM0->SC |= 0x10;              // Számlálás indítása külső órajellel
}
```

Program5_10: Külső események számlálása

```
while (1) {
    count = TPM0->CNT;          // Számláló kiolvasása

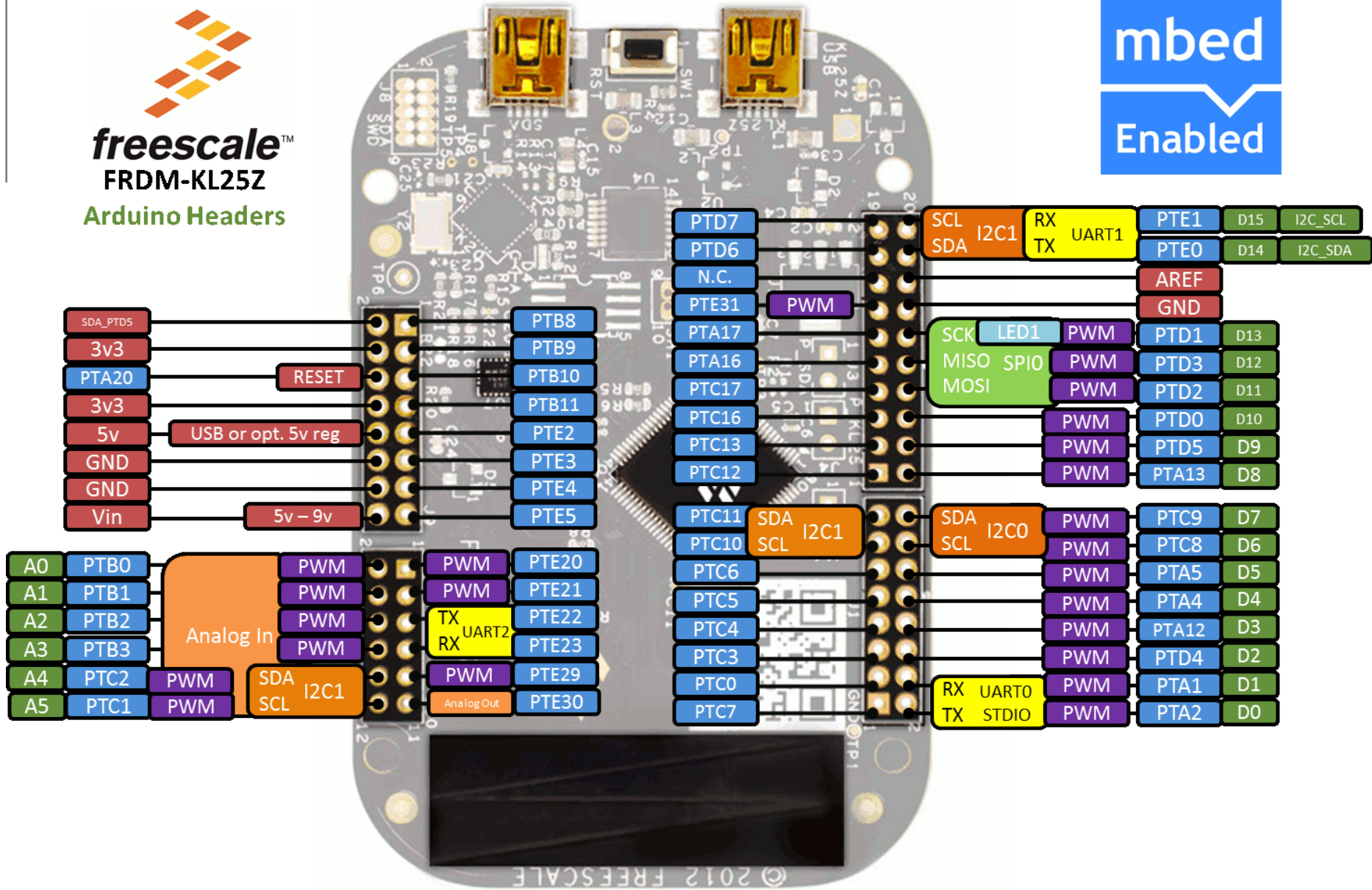
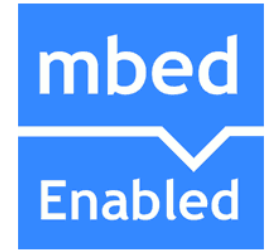
    //--- LED-ek beállítása a számláló legalsó bitjei szerint
    if (count & 1)              // Piros LED beállítása bit0 szerint
        PTB->PCOR = 0x40000; // LED = be
    else
        PTB->PSOR = 0x40000; // LED = ki

    if (count & 2)              // Zöld LED beállítása bit1 szerint
        PTB->PCOR = 0x80000;
    else
        PTB->PSOR = 0x80000;

    if (count & 4)              // Kék LED beállítása bit2 szerint
        PTD->PCOR = 0x02;
    else
        PTD->PSOR = 0x02;
}
}
```



freescale™
FRDM-KL25Z
 Arduino Headers





freescale™
FRDM-KL25Z

Additional Peripherals



- PTE24 SCL
- PTE25 SDA
- PTA14 INT1
- PTA15 INT2

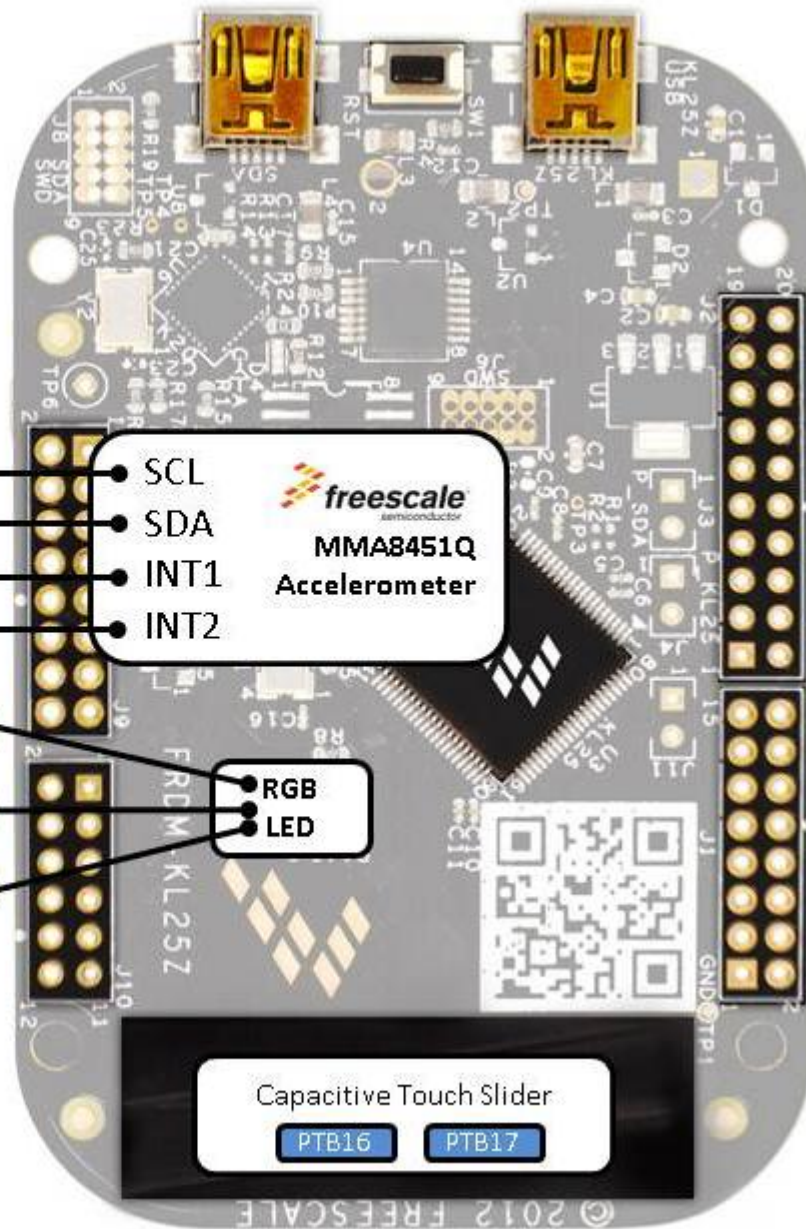
freescale
MMA8451Q
Accelerometer

- LED1 PTB18 PWM
- LED2 PTB19 PWM
- LED3 PTD1 PWM

RGB
LED

Capacitive Touch Slider

- PTB16
- PTB17



© 2012 FREESCALE