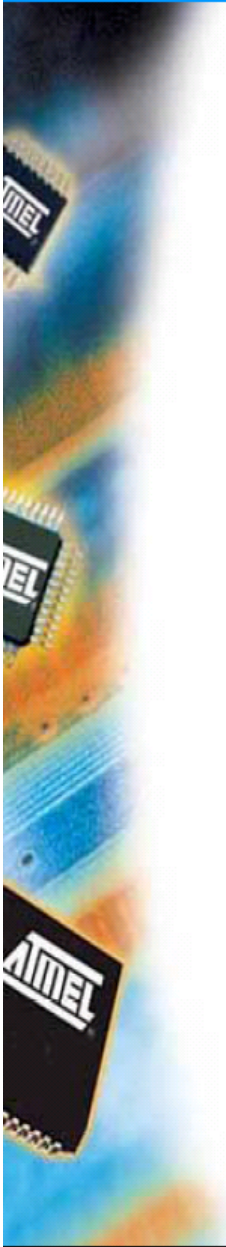


**AVR ATmega128
assembly**

2. mérés

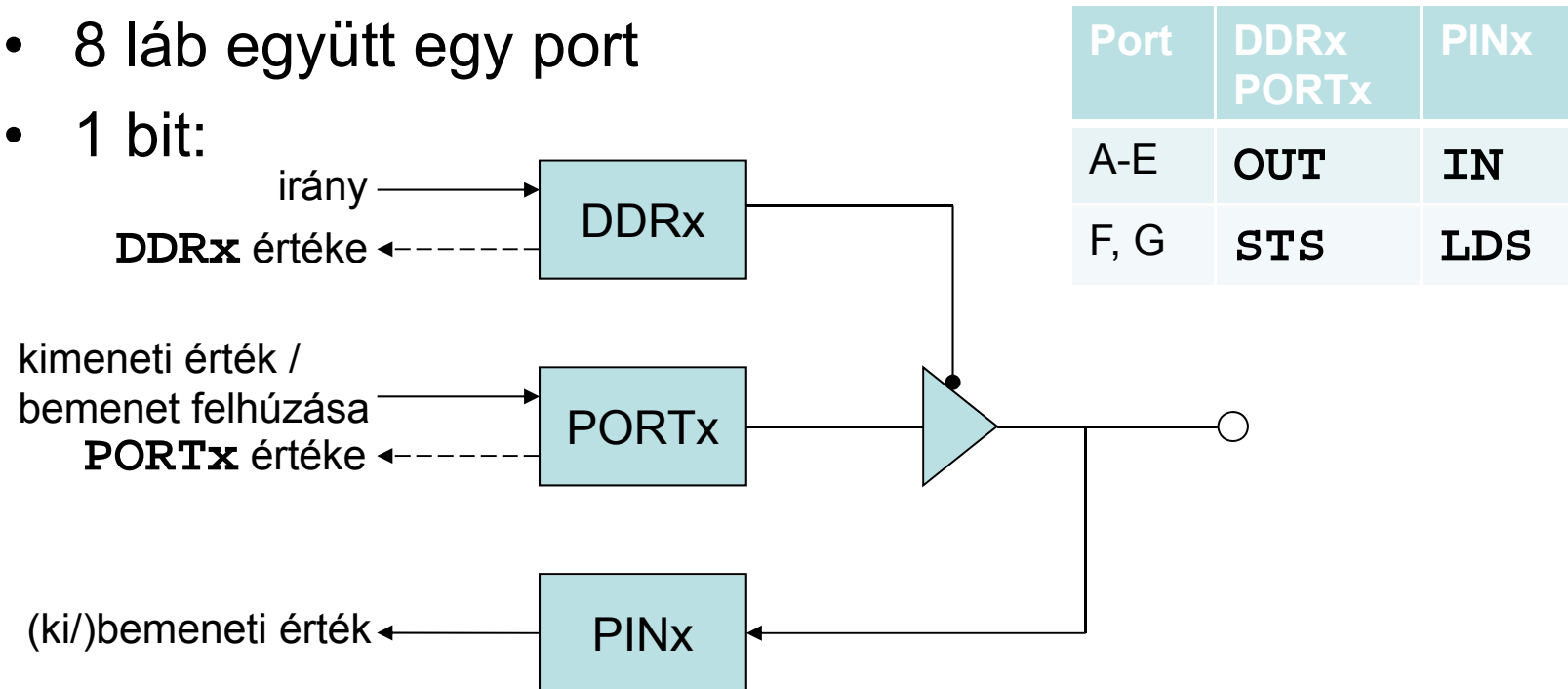


A mérés témái

- **Ismétlés**
- **Időzítés**
- **Mérési feladatok**










I/O portok kezelése

- 8 láb együtt egy port
- 1 bit:

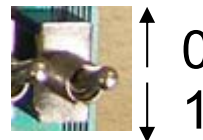


I/O	DDRx (irány)	PORTx (adat ki)	PINx (adat be)
BE	0	0: Hi-Z, 1: felhúzó ell.	Bemeneti érték
KI	1	Kimeneti érték	PORTx értéke

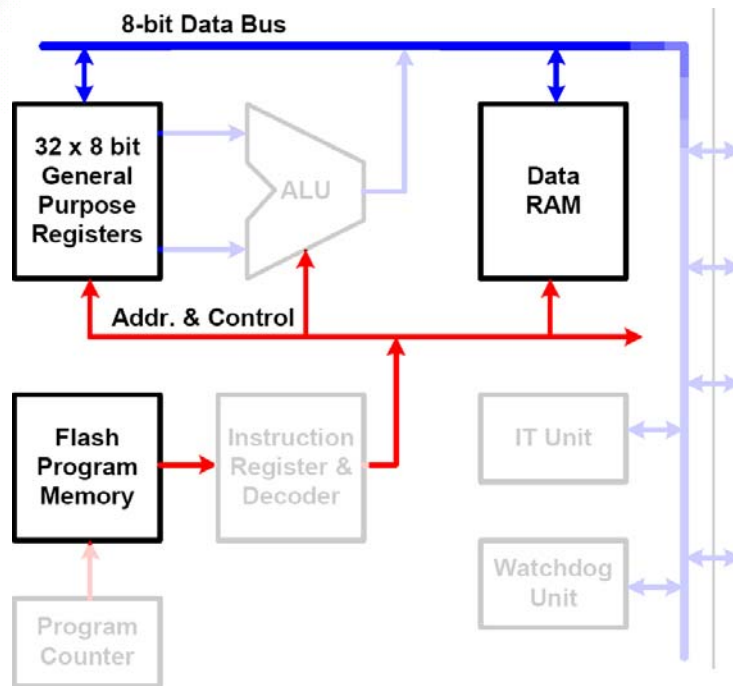
I/O portok kezelése - mérőpanel perifériái

port	7	6	5	4	3	2	1	0
B	BT3							
C	LED7 	LED6 	LED5 	LED4 	LED3 	LED2 	LED1 	LED0 
E	BT2	BT1	BT0	INT				
G				 SW2	SW3		SW1	SW0

- Minta programváz elején szerepelnek
- LED: C port
- Nyomógombok: E port + B
 - alacsony aktív, felengedve 1, lenyomva 0
- Kapcsolók: G port



ATmega128 memóriák



Adatmemória:

- 32 általános célú regiszter
- I/O regiszterek
 - normál + „kibővített”
 - kibővített I/O: memóriába ágyazott, úgy kezelhető, mint az SRAM

Memória:

- 128K Flash (program)
- 4K belső SRAM (adat) | Harvard-architektúra
- 4K EEPROM (nem felejtő)

Programmémória:

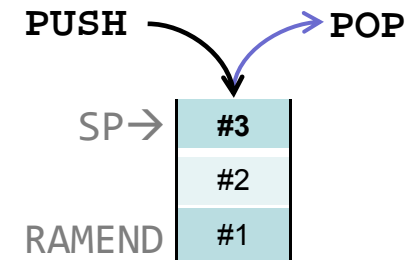
szavas címezés → 16 bittel 128KB

Adatmemória	
\$0000 -	32 regiszter
\$0020 -	64 I/O reg.
\$0060 -	160 ext. I/O reg.
\$0100 -	Belső SRAM (4096 x 8)
RAMEND = \$10FF -	(Külső SRAM)

AVR assembly – speciális regiszterek

Veremmutató (SP, Stack pointer)

- verem: LIFO memóriatartomány a belső SRAM végén, a szubrutinok és a megszakításkezelő rutin visszatérési címei ill. regiszterek átmeneti tárolására
- **SP**: 16 bites reg, verem tetejére mutat (legalacsonyabb című veremadat elé)
- Inicializálás RAM tetejére
- mentés: **PUSH**, visszatöltés: **POP**



```
.def temp = r16
ldi temp, LOW(RAMEND)
out SPL, temp
ldi temp, HIGH(RAMEND)
out SPH, temp
```

SP

```
.org $0050
M_LOOP:
$0050 ---- ldi r16, 0x12
$0051 ---- call STACKDEMO ; 32 bit
$0053 ---- jmp M_LOOP ; 32 bit
STACKDEMO: ; címek folytatódnak!
$0055 ---- push r16 ; r16 mentése
; r16 más célokra...
$0056 ---- pop r16 ; r16 = 0x12
$0057 ---- ret
```

Utasítás	\$10FC	\$10FD	\$10FE	\$10FF
Init	?	?	?	?
CALL	?	?	00	53
PUSH	?	12	00	53
POP	?	12	00	53
RET	?	12	00	53

Veremhasználat példa

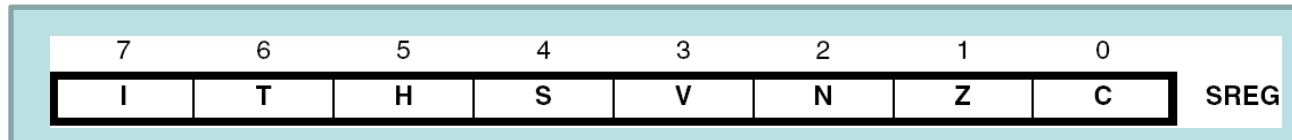
```
.org $0050
M_LOOP:
$0050 -- ldi r16, 0x12
$0051 -- call STACKDEMO
$0053 -- jmp M_LOOP
STACKDEMO:
$0055 -- push r16
;...
$0056 -- pop r16
$0057 -- ret
```

Utasítás	\$10FC	\$10FD	\$10FE	\$10FF
Init	?	?	?	?
CALL	?	?	00	53
PUSH	?	12	00	53
POP	?	12	00	53
RET	?	12	00	53

SP

AVR assembly – speciális regiszterek

Státuszregiszter (SREG) – flagek („állapotjelző bit”)



- I: globális megszakításengedélyezés
- T: „bitmásolás”
- H: half-carry (4 bites műveletek)
- S: előjel (N és V alapján)
- V: kettes komplementes túlcsordulás
- N: eredmény 7. (15.) bitje
- Z: minden bit 0
- C: átvitel-bit

Státuszregiszter példa

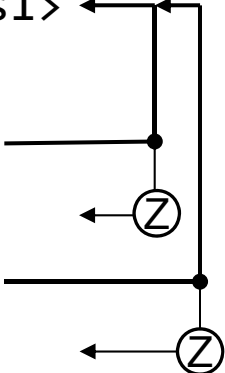
Mnemo.	Operands	Description	Operation	Flags
SEI		Global Interrupt Enable	$I \leftarrow 1$	I
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V
SUBI	Rd, K	Subtract Immediate	$Rd \leftarrow Rd - K$	Z,C,N,V,H

Utasítás	R16	N	Z	C
ldi r16, 254	0xFE	0	0	0
inc r16	0xFF	1	0	0
inc r16	0x00	0	1	0
inc r16	0x01	0	0	0
subi r16, 3	0xFE	1	0	1

Szoftveres időzítés

Több byte-os ciklusszámlálójú ciklus

```
.def i = r16
.def j = r17      ; 300 = 0x12C
    ldi i, HIGH(300)+1 ; =0x1+1
    ldi j, LOW(300)    ; =0x2C
LOOP: <utasítás1> ←
    dec j
    brne LOOP
    dec i
    brne LOOP
<utasítás2> ←
```

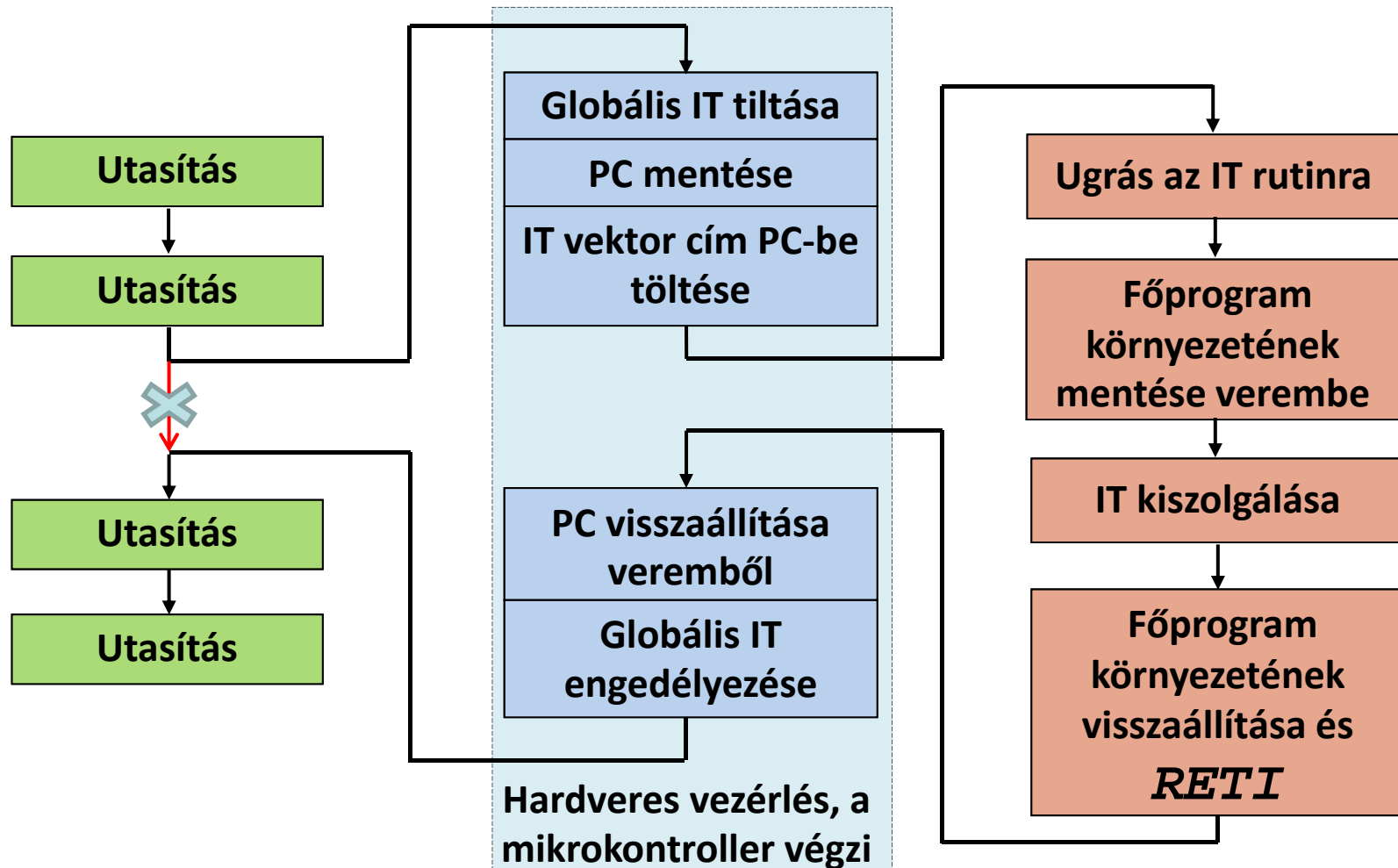


- 11MHz → több millió kör – 2 regiszterben csak $2^{16}=65536$
- pontos ciklusszám nem túl fontos, nem fogjuk pontos időzítésre használni

1. feladat

- LED-ek léptetése (*mindig* 1 led világít, sorban végiglép 0,1..7,0...**szimulátorban**)
- Szoftveres késleltető szubrutin 1-4Hz (**JTAG ICE**)
- Debug kipróbálása + képernyőkép:
 - töréspont,
 - Run/Autostep/Run to Cursor
 - Step Over/Step Into/Step Out közti különbség
 - I/O portok
 - **Watch ablak** (View/Watch), 1-2 regiszter

Megszakításkezelés



Megszakításkezelés példa: BT0

Ugrótábla / vektortábla

4 bájtnyi hely - egy JMP elfér

0000	jmp RESET	RESET
0002	jmp DUMMY_IT	INT0
0004	jmp DUMMY_IT	INT1
...		
000C	jmp BT0_IT	INT5
...		
0044	jmp DUMMY_IT	SPM ready

DUMMY_IT: nem kezelt megszakítás → LED-ek, végtelen ciklus

Nyomógomb-megszakítás konfigurálása

port	7	6	5	4	3	2	1	0
E	BT2	BT1	BT0	INT				

E port nyomógombjai megszakítással is kezelhetők

- **EICRB**: szint/élvezérelt
- **EIFR**: „flag register”
- **EIMSK**: engedélyezett IT-források

```
ldi temp, 0 ; E port bemenet
out DDRE, temp
ldi temp, 0b00001000 ; megszakítási esemény:
; .....10.. ; ISC5=10 (INT5 lefutó él)
;ldi temp, (1<<ISC51) ; ugyanaz másképp → m128def.inc
out EICRB, temp
ldi temp, 0b00100000 ; INT5 megszakítást engedélyezzük
; ..1..... ; INT5=1
;ldi temp, (1<<INT5) ; másképp
out EIMSK, temp
sei ; globális IT engedélyezése
```

Megszakításkezelő rutin

BT0_IT:

; nem kell CLI

push temp

in temp, SREG

push temp

SREG mentése verembe
(ha pl. egy összehasonlítás és
az ez alapján történő elágazás
között futna le az IT rutin)

<...IT-kezelés...>

; visszaállítjuk a státusz, temp regiszert

pop temp

out SREG, temp

pop temp

; nem kell SEI:

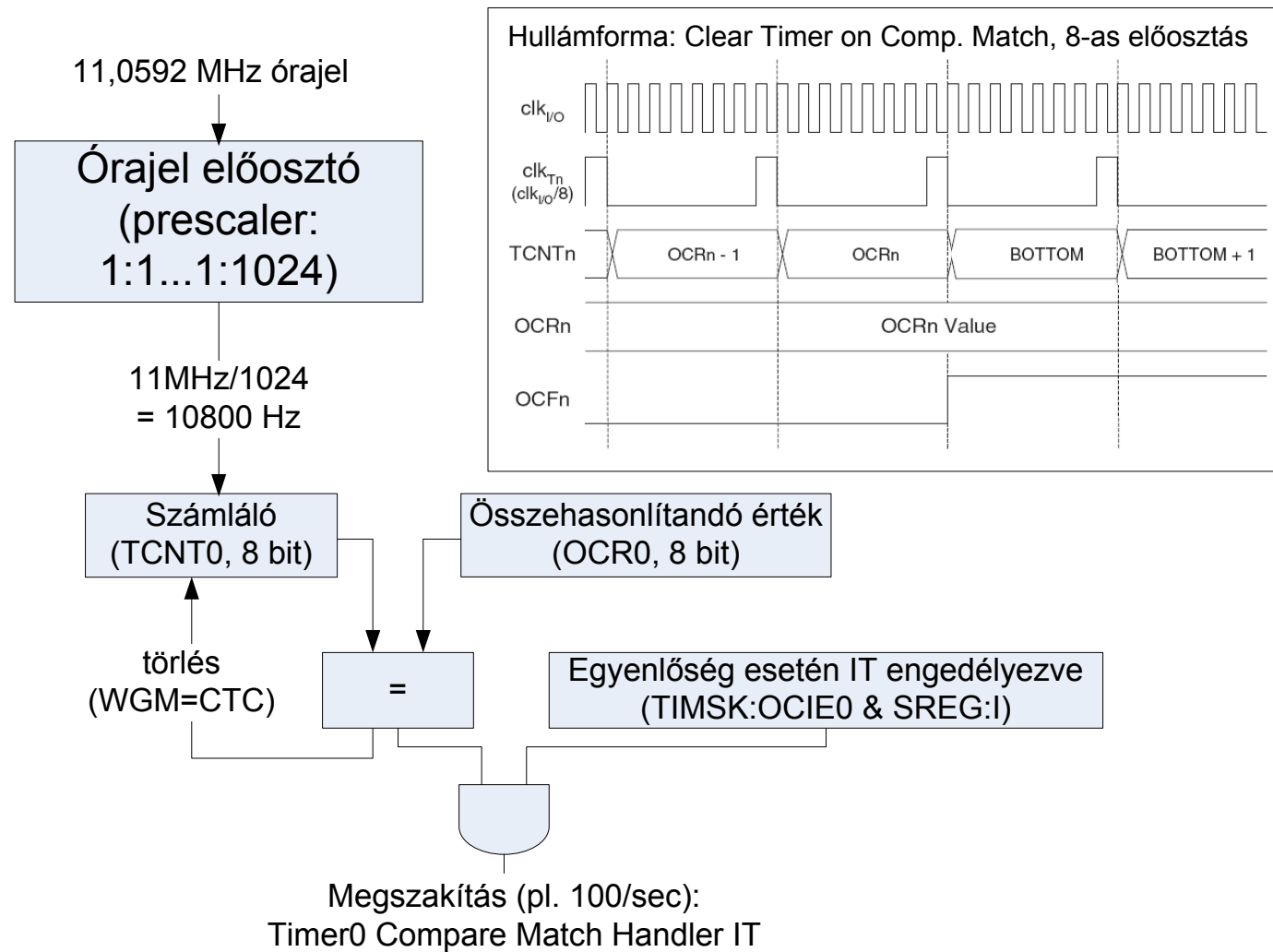
reti ; reti → SREG I bitje visszaáll,

; PC veremből visszatöltődik

2. feladat

- Nyomógombbal megállítható futófény, második lenyomásra tovább fut
 - nyomógombot megszakítással
- Szimulátorban teszt: otthon hogyan tudunk IT-t generálni

Hardveres időzítés – Timer/Counter0

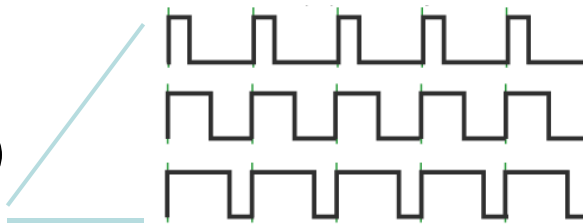


Timer/Counter0 regiszterek

Bit	7	6	5	4	3	2	1	0
TCNT0	TCNT0[7:0]							
OCR0	OCR0[7:0]							
TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0

WGM: Waveform Generation Mode

- Normál: 0..255
- CTC: 0..OCR0
(„Clear Timer on Compare Match”)
- PWM: pulzusszélesség-moduláció
(→ Timer1: OC1A kimenet, fokozatos LCD háttérvilágítás)



CS: Clock Select

OCIE: Output Compare match Interrupt Enable

Időzítés megszakítással

```
; ***** Timer 0 inicializálása *****  
; előosztás (prescaler) beállítása  
ldi temp, 0b00001111  
    ; 0..... ; FOC=0  
    ; .0..1... ; WGM=10 (CTC)  
    ; ..00.... ; COM=00 (kimenet tiltva)  
    ; .....111 ; CS0=111 (CLK/1024)  
out TCCR0,temp ; Timer 0 TCCR0 regiszter  
; compare regiszter beállítása  
ldi temp, 107 ; 11059200Hz/1024 = 108*100  
out OCR0, temp ; Timer 0 OCR0 regiszter  
; Timer 0 IT engedélyezése, többiek tiltása  
ldi temp, 0b00000010  
    ; 000000.. ; Timer2,1 IT tiltva  
    ; .....1. ; OCIE0=1 - match  
    ; .....0 ; TOIE0=0 - overflow  
out TIMSK,temp ; Timer IT Mask regiszter  
sei            ; globális IT engedélyezve
```

dokumentáció segítségével
össze kell tudni állítani

pontos baudrate érdekében

3. feladat

- HW időzítő felprogramozása 10ms periódusidőre
- Késleltető szubrutin cseréje olyanra, ami 25-100 timer IT lefutás után tér vissza
- Jegyzőkönyv, HF