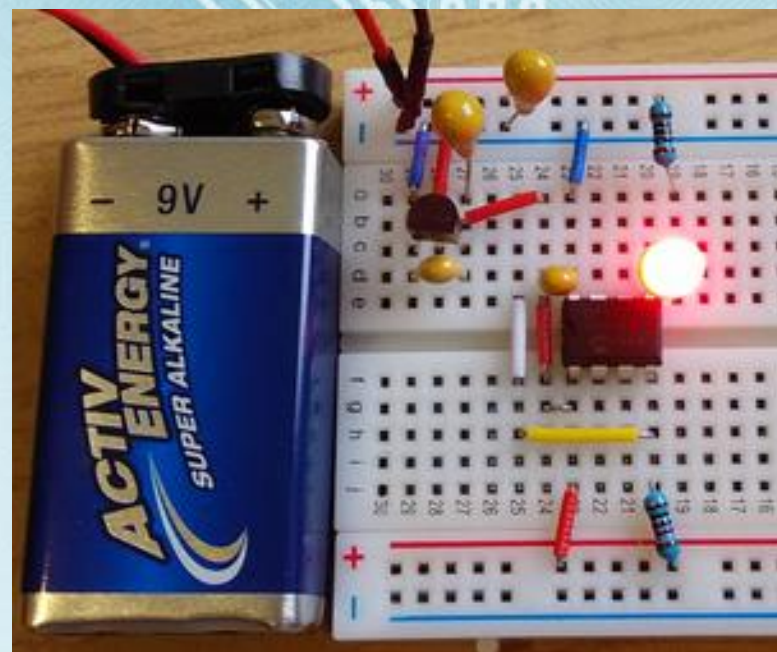
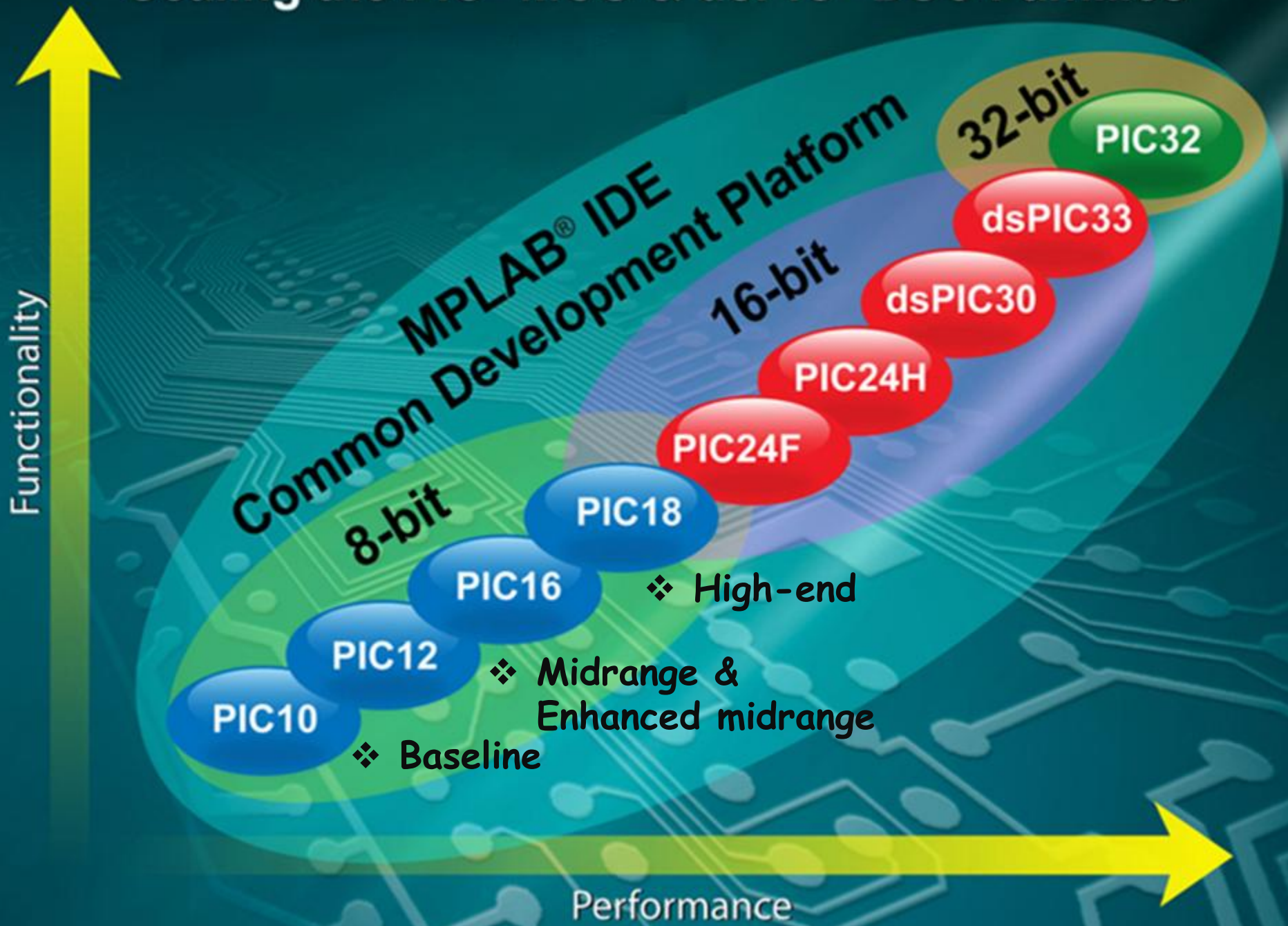


Hobbi Elektronika

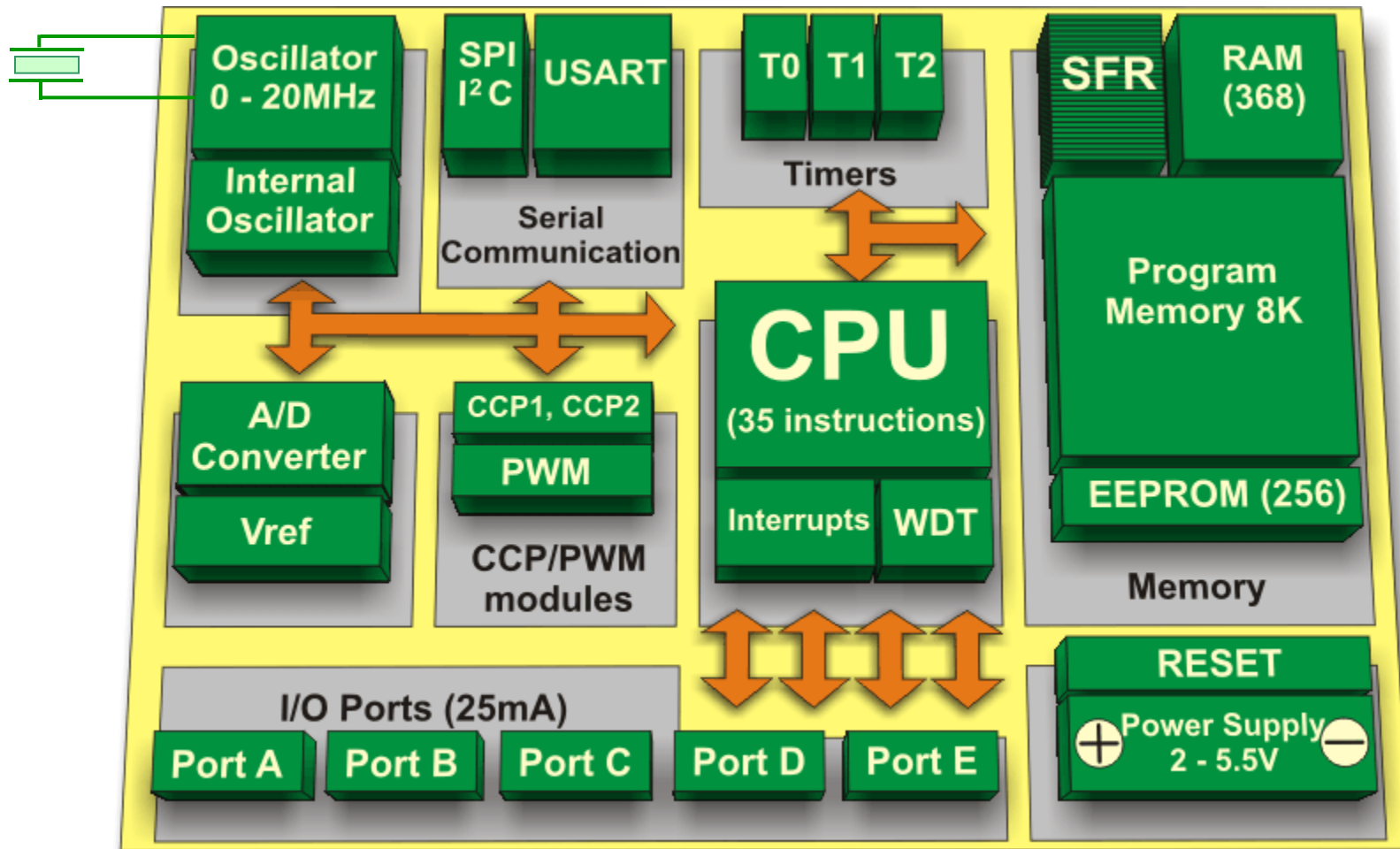


Mikrovezérlők programozása Microchip PIC mikrovezérlők – bevezető előadás

Scaling the PIC[®] MCU & dsPIC[®] DSC Families



8-bites „midrange” PIC felépítése

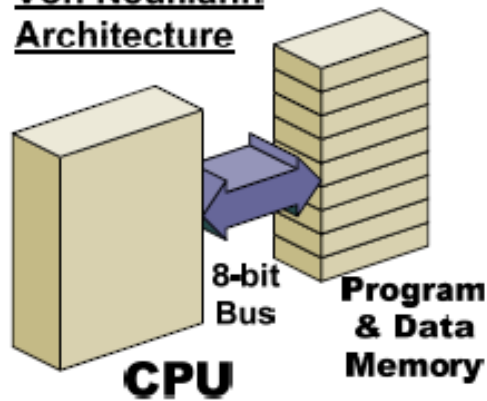


PIC16F887

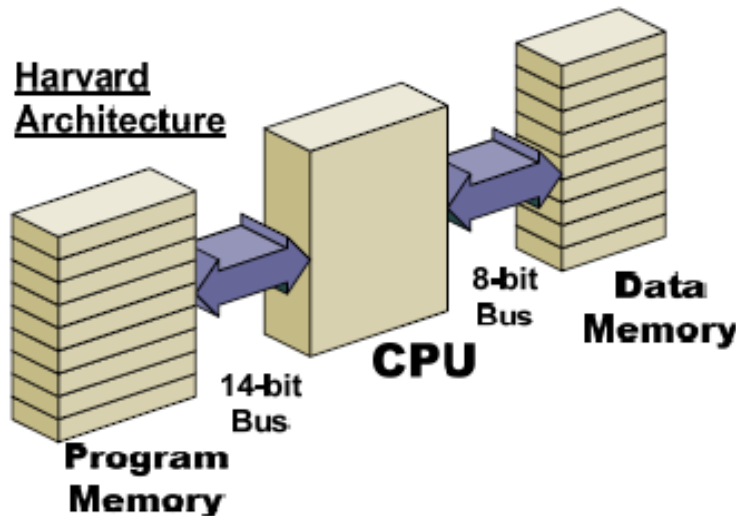
Forrás: www.mikroe.com

Harvard felépítés

Von Neumann Architecture



Harvard Architecture



□ Neumann-felépítés

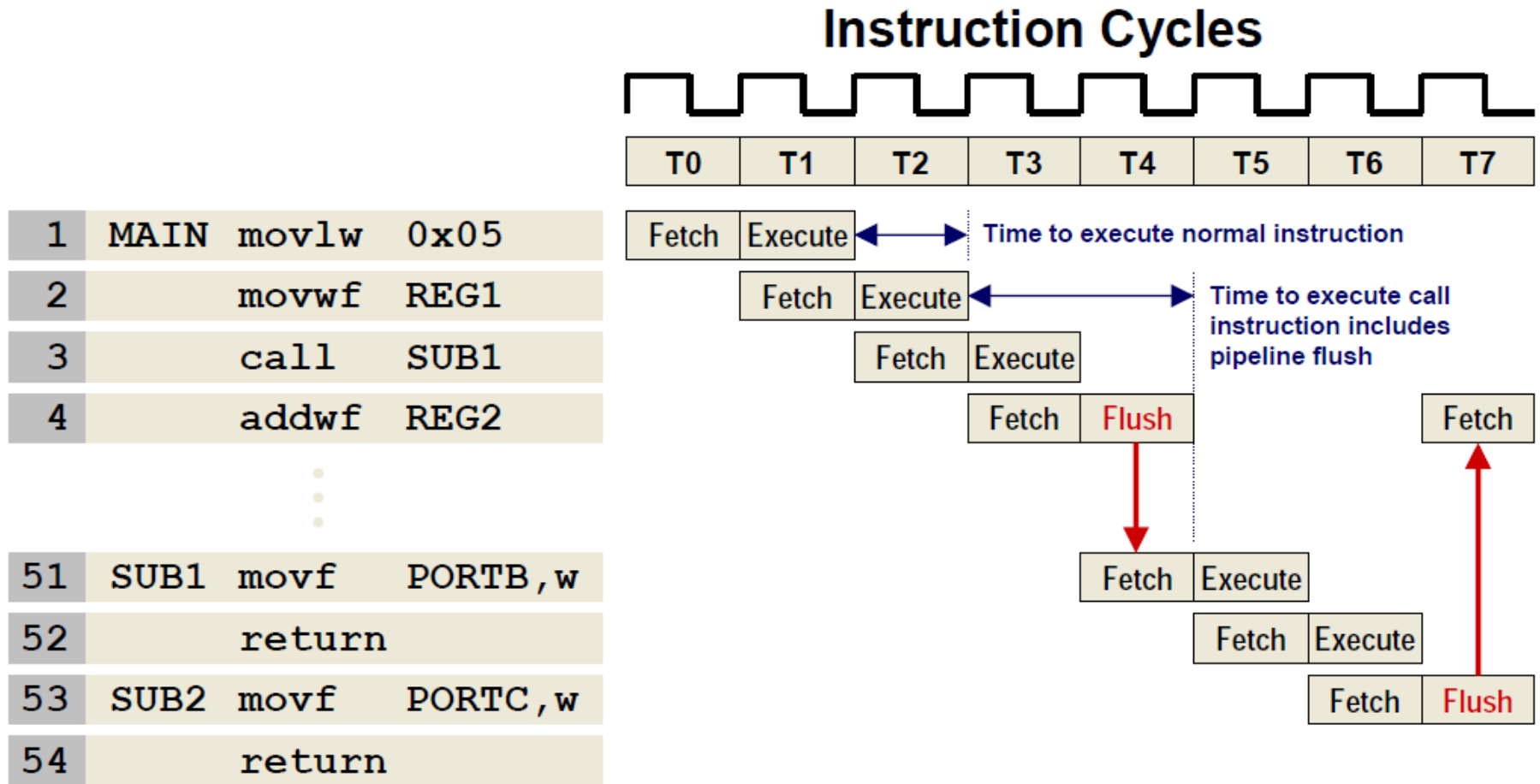
- Az utasításokat és az adatokat ugyanabban a memóriában tárolja
- Korlátozott sávszélesség

□ Harvard felépítés

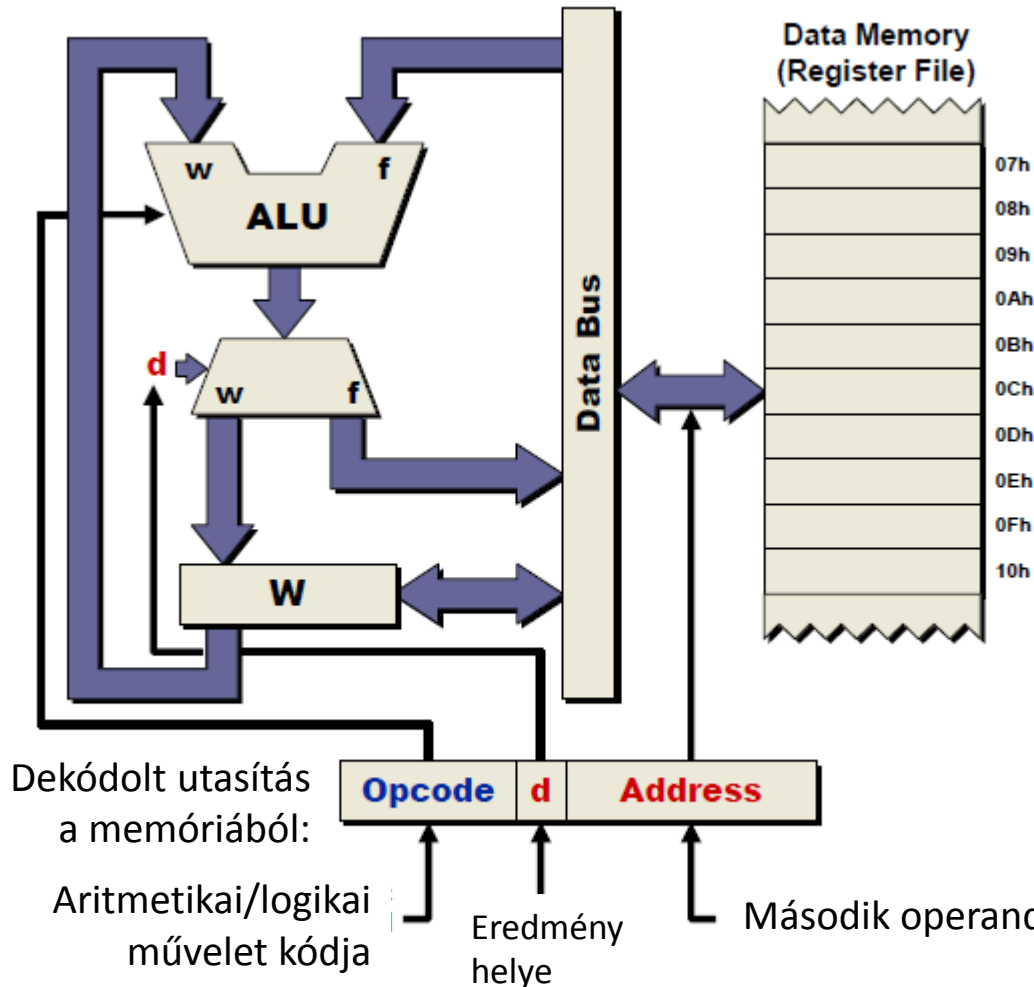
- Az utasításokat és az adatokat külön-külön memóriában tárolja
- A párhuzamos működés miatt nagyobb a sávszélesség
- Lehetőség különböző szélességű kiépítésre (program memória 14-bites, adatmemória 8-bites)

Utasítás futószalag

Az aktuális utasítás végrehajtásával párhuzamosan folyik a következő utasítás elővétele



A regiszter fájl koncepció

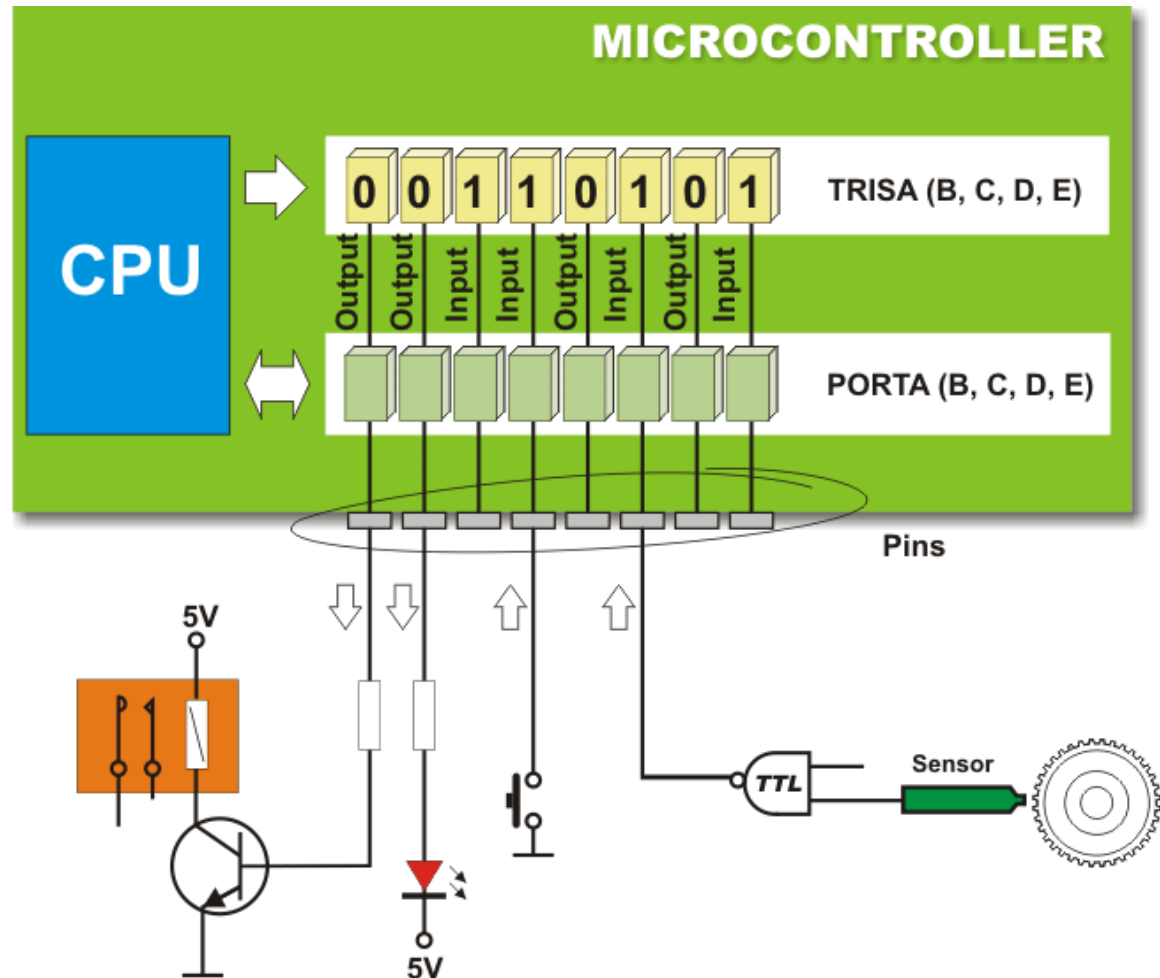


- ❖ A teljes adatmemória a regiszterkészlethez tartozik, tehát az adatok közvetlenül manipulálhatók.
- ❖ Minden periféria regiszter az memóriába van leképezve
- ❖ **Ortogonalis utasításkészlet:** minden utasítás a memória bármely részét kezelheti.
- ❖ **A szószervezésű utasításokkal** közvetlenül címezhetjük a lapon belüli (7-bites címtartomány) regisztereket

A *d* bit beállításától függően az eredmény a regiszterfájlba, vagy a *W* munkaregiszterbe kerül.

Digitális I/O

Egy port általában 8 bites szervezésű, de bitenként is kezelhető, beleértve az adatáramlás irányának megválasztását is.



Forrás: www.mikroe.com

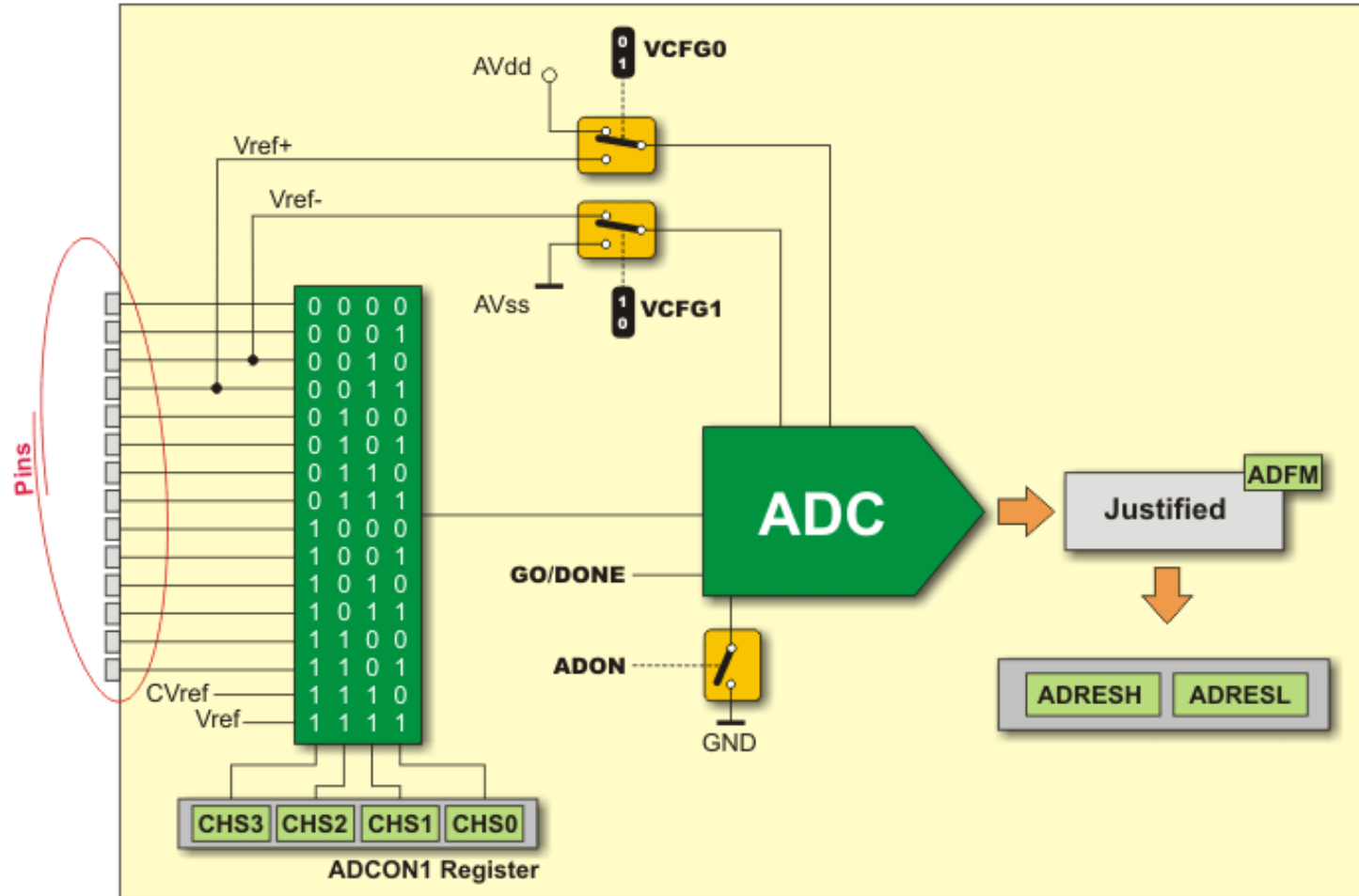
Analóg bemenetek (10-bites ADC)

Referencia: belső vagy külső

Felbontás: 10 bit (0 – 1023)

Felépítés: SAR (fokozatos megközelítés elvén)

Sebesség: 20 k – 1.1 M mintavétel/s (típustól függően)



Forrás: www.mikroe.com

OPTION_REG regiszter

Mielőtt a **Timer0** időzítő/számláló-val megismerkedünk, tisztáznunk kell az OPTION_REG regiszter szerepét!

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features	
OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

RBPU – belső felhúzások engedélyezése

INTEDG – INT megszakításhoz élválasztás (**0**: lefutó élre, **1**: felfutó élre)

T0CS – **Timer0** órajel választás (**0**: belső órajel ($F_{osc}/4$), **1**: külső órajel)

T0SE – **Timer0** órajel élválasztás (**0**: felfutó élre számlál, **1**: lefutó élre számlál)

PSA – Előszámláló hozzárendelés (**0**: TMR1 használja, **1**: WDT használja)

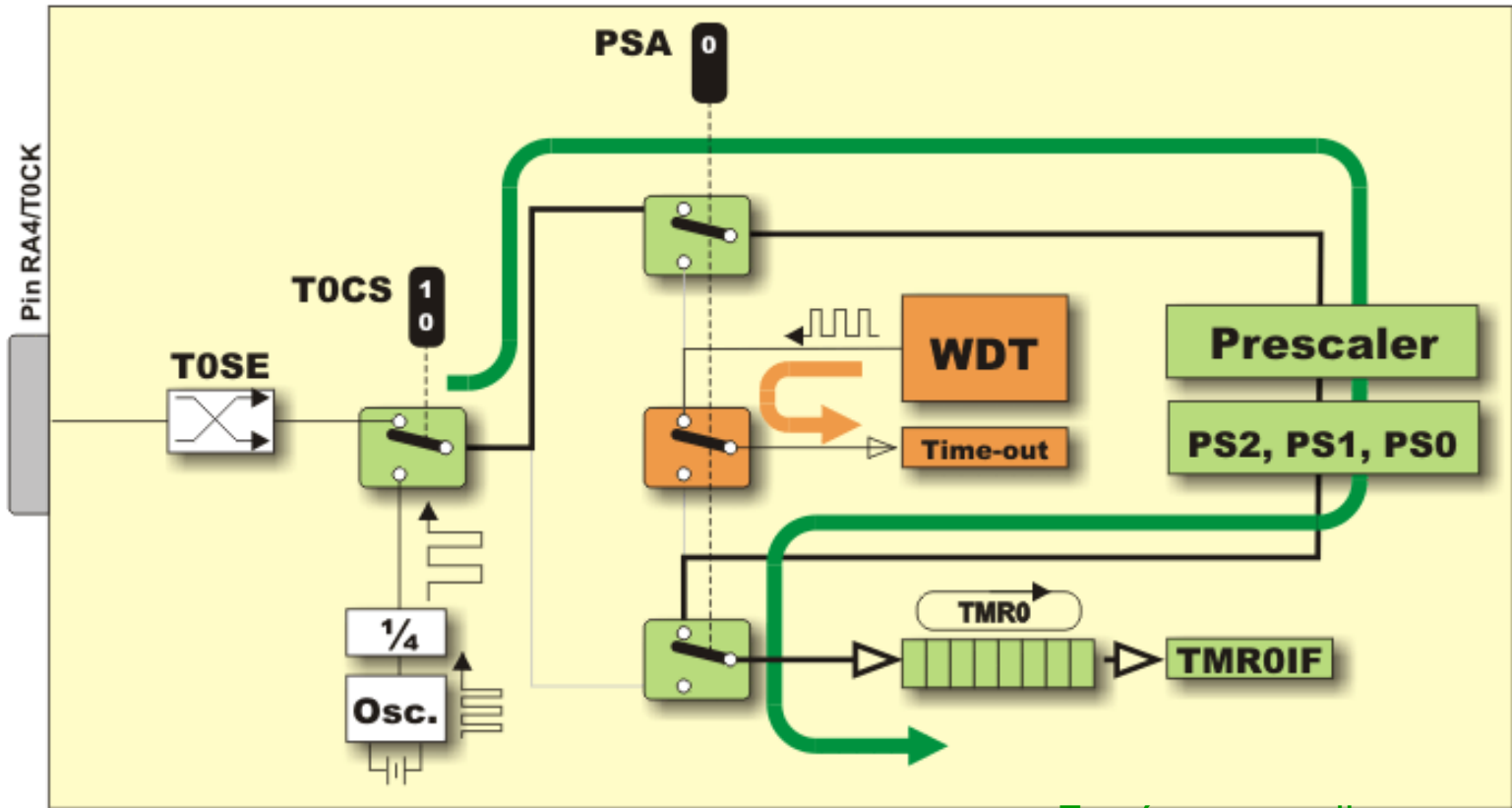
PS[1:0] – előosztási arány beállítása (2^{N+1} TMR0 esetén, illetve 2^N WDT esetén)

Forrás: www.mikroe.com

Timer0 8-bites időzítő/számláló

Timer0 előosztója vagylagosan az időzítőhöz, vagy a **Watchdog**-hoz rendelhető. Az ábrán látható esetben az **OPTION_REG** **PSA** bitje '0', ezért az előosztó **TMR0**-hoz kapcsolódik.

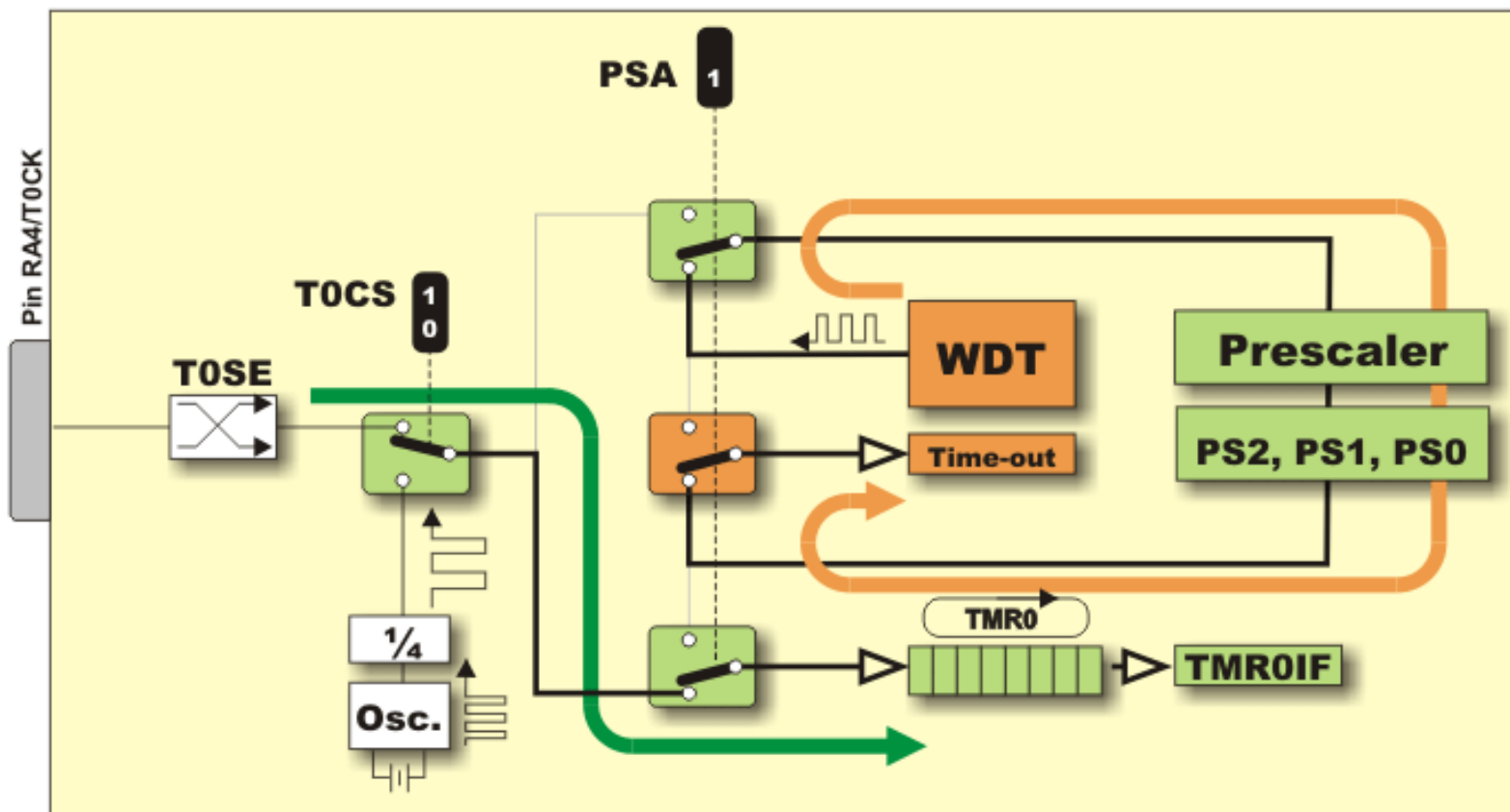
A **Watchdog** az elakadt programok utolsó menedéke – ha nem töröljük rendszeresen, túlcsoordulásakor **RESET**-et okoz.



Forrás: www.mikroe.com

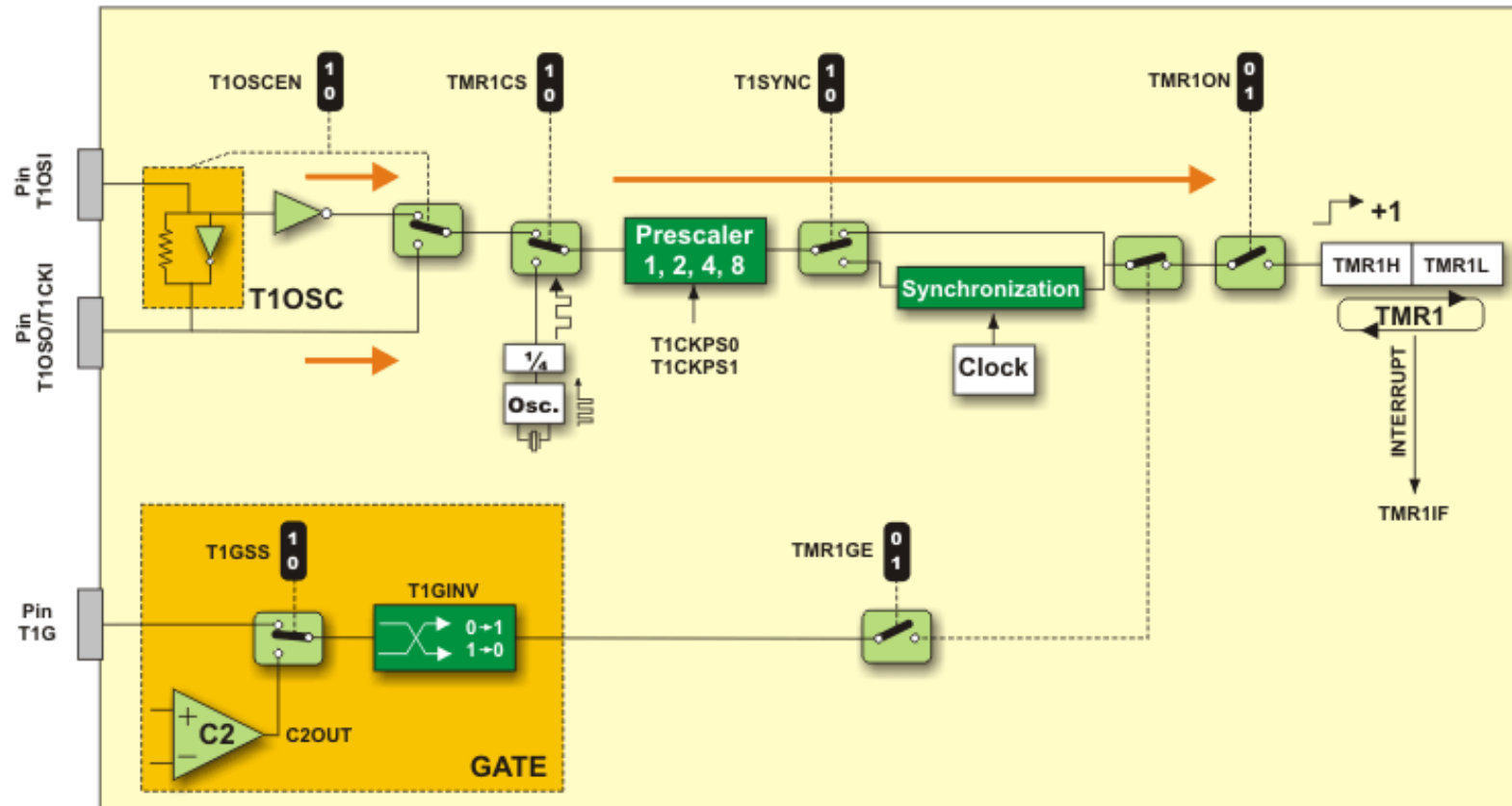
Timer0 8-bites időzítő/számláló

Az alábbi ábrán látható esetben az **OPTION_REG PSA** bitje '1', ezért az előosztó a **WDT**-hez kapcsolódik, **TMRO** pedig előosztó nélkül, közvetlenül a bejövő órajelet számlálja. Ebben az esetben a számláló legkésőbb 256 ciklusonként túlcsoordul.



Forrás: www.mikroe.com

Timer1 16-bites időzítő/számláló



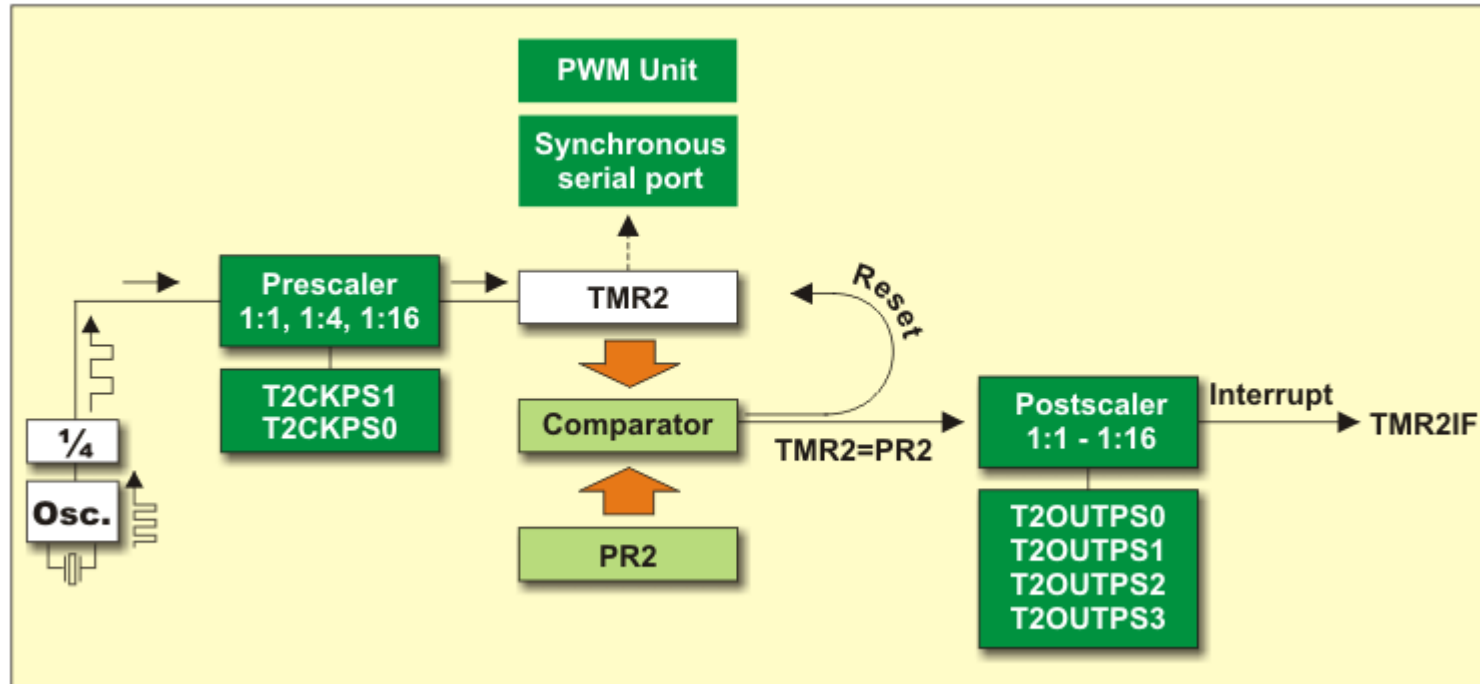
- Külső vagy belső jelforrás
- Szinkron/aszinkron mód
- Kapuzás külső jellel, vagy az analóg komparátor kimenetével

Előnyös tulajdonság, hogy a szinkronizálás az előszámláló kimenetén történik!

Forrás: www.mikroe.com

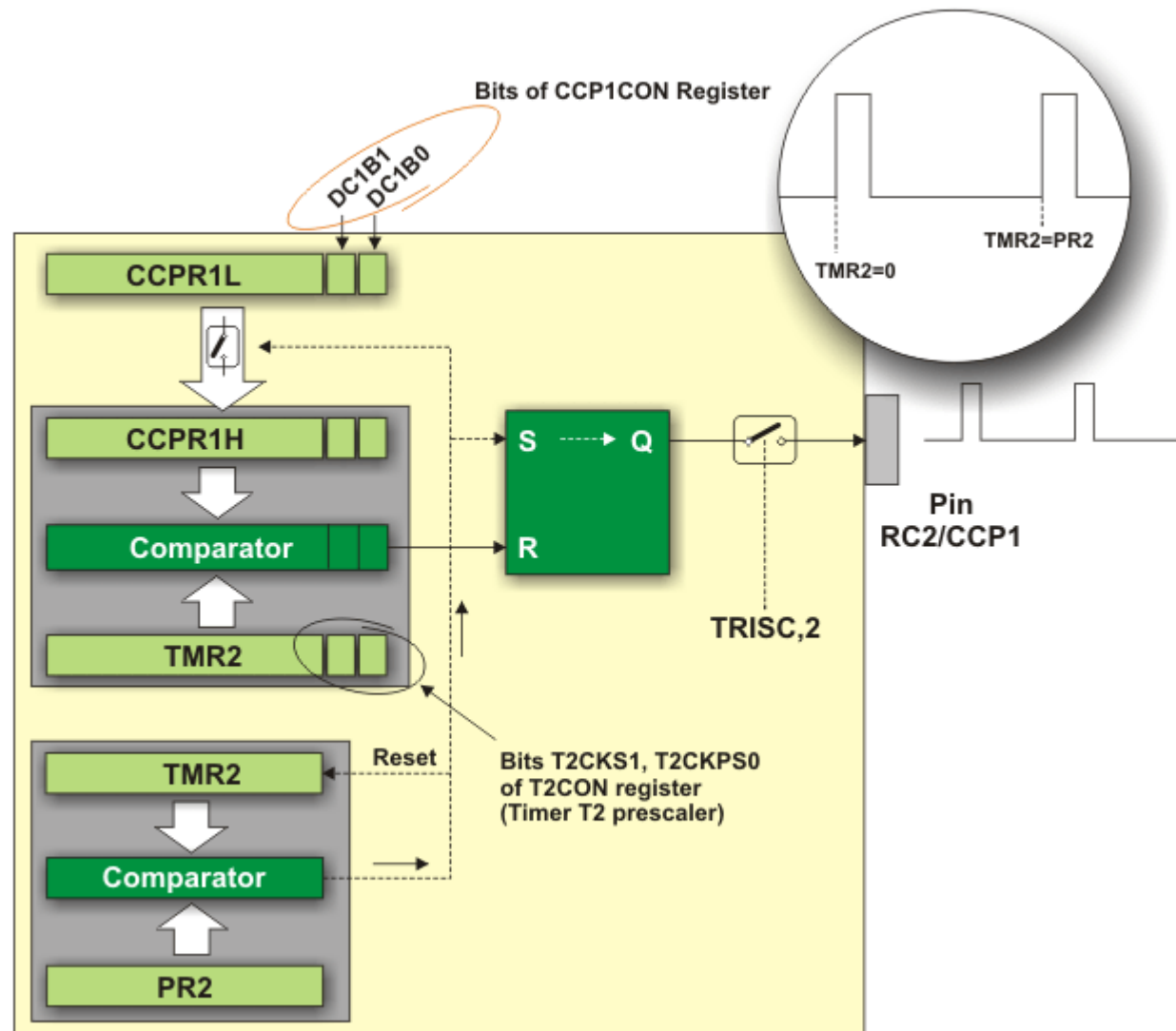
TMR2 8-bites számláló elő- és utóosztással

Általában periodikus időzítésre használjuk: rendszeres időközönkénti megszakításokhoz, vagy PWM jel előállításához (CCP modullal kombinálva).



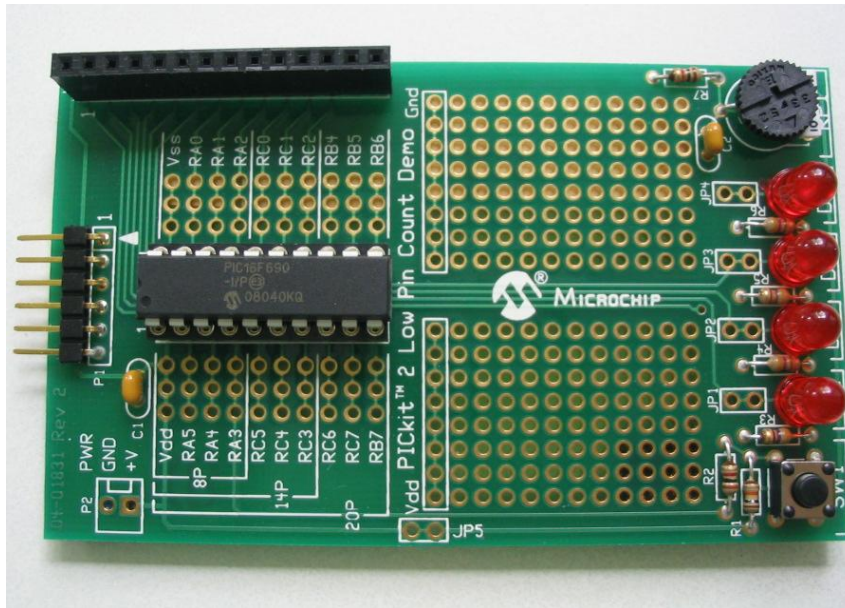
Forrás: www.mikroe.com

PWM jel generálás (CCP1+TMR2)



Forrás: www.mikroe.com

PICkit2 Starter Kit (2007)



Low Pin Count Demo

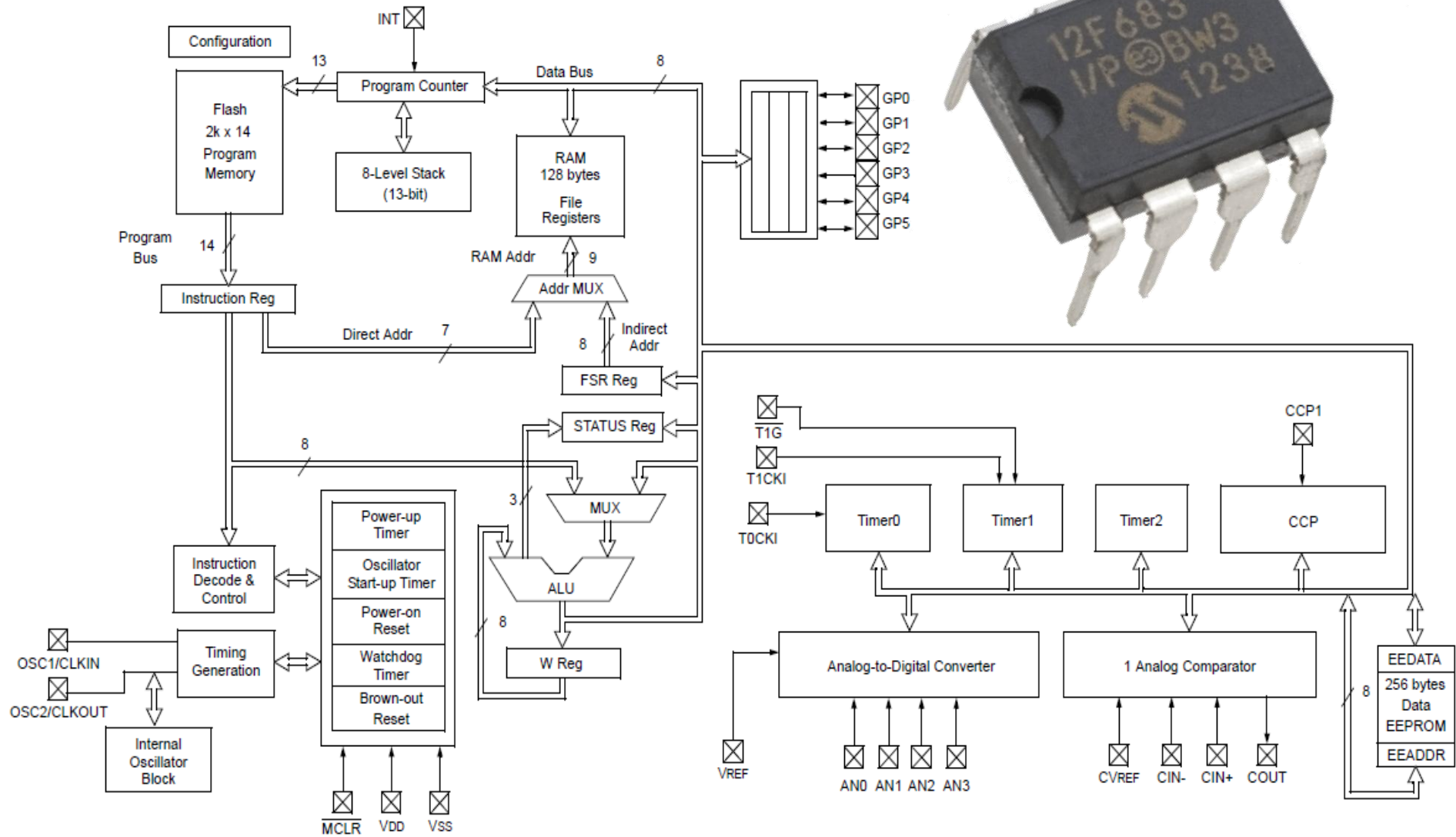
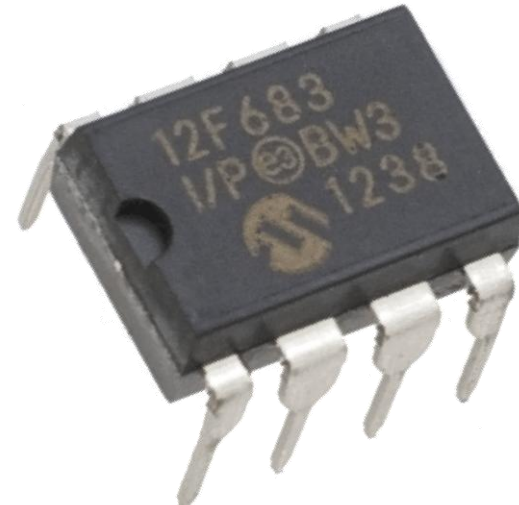
- PIC16F690
- 4 db LED
- nyomógomb
- Potméter
- ICSP csatlakozó
- Protoboard felület
- PICtail bővítő csatlakozó

PICkit2

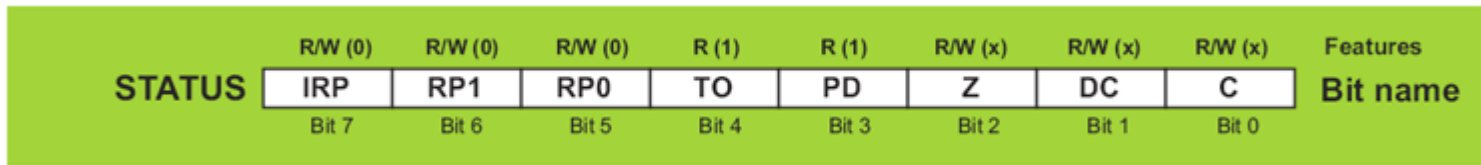
- Programozó
- Nyomkövető
- Logikai analizátor
- Soros kommunikátor



Egy konkrét típus: PIC12F683



A STATUS regiszter



IRP – Regiszter bank választás kiegészítő bitje indirekt címzésnél.

RP1, RP0 – Regiszter bank választása direkt címzés esetén (lásd BANKSEL).

TO – Időtúllépés jelzése (**0**: WDT túlcsordulásakor, **1**: egyébként).

PD – Power-down jelzése (0: SLEEP után, 1: egyébként)

Z – Zero bit (1: ha a művelet eredménye 0)

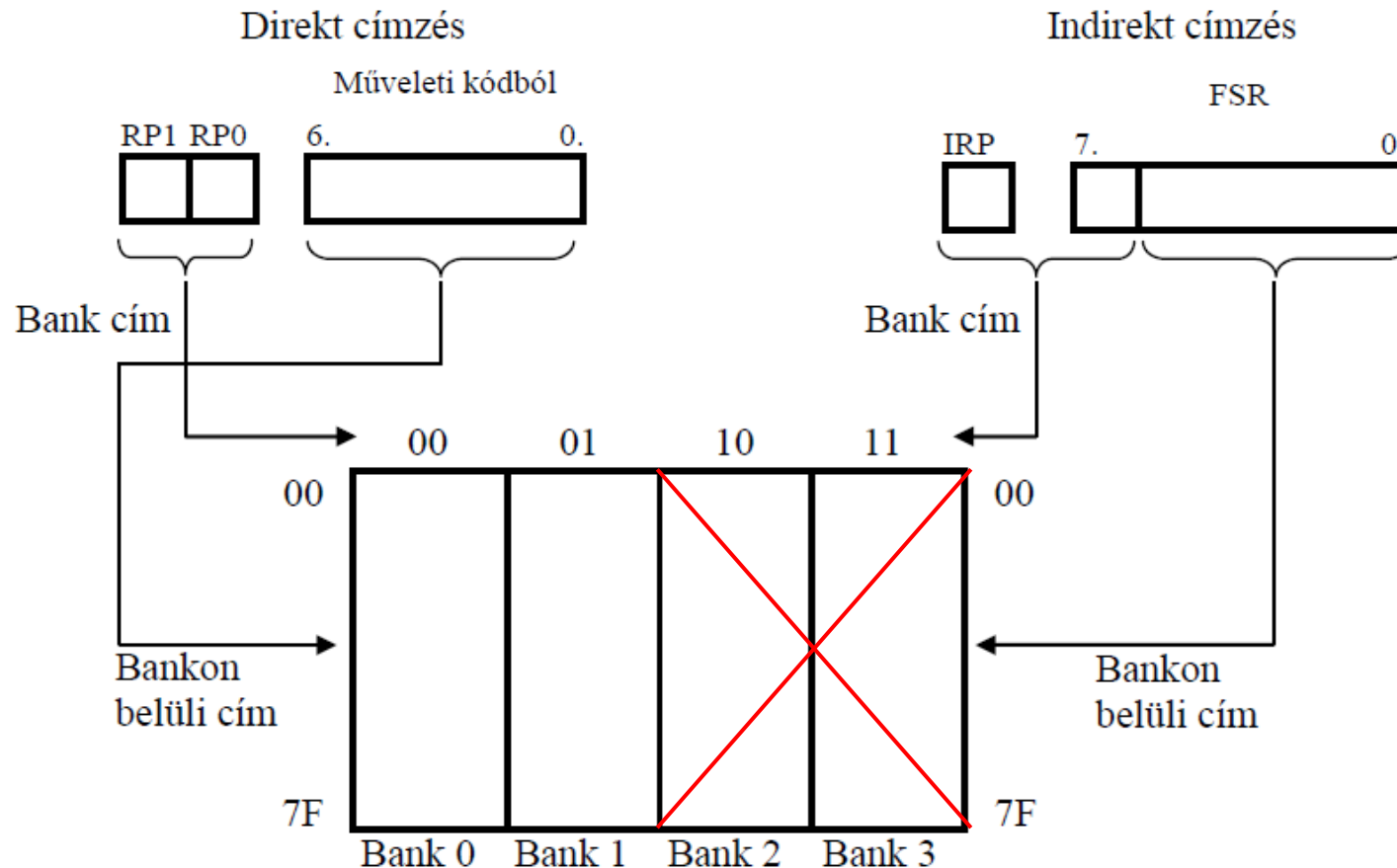
DC – számjegy túlcsordulás/áthozat bit BCD összeadásnál, illetve kivonásnál

C – túlcsordulás jelző bit

Indirekt és direkt címzés

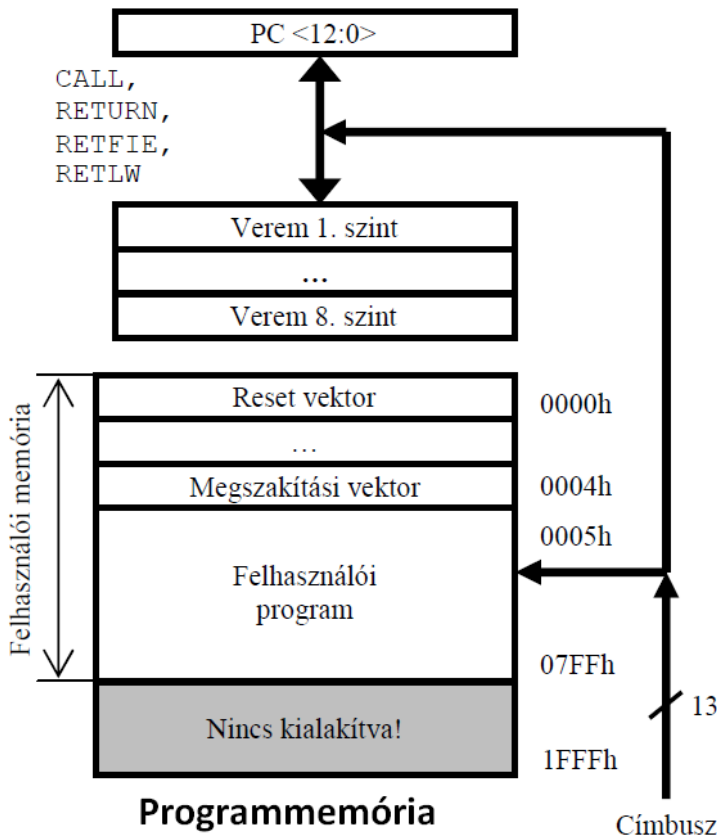
Adatmemória és az **SFR** regiszterek elérésére kétféle címzést használhatunk:

- Direkt címzés az utasítás kódjában tárolt címmel
- Indirekt (indexelt) címzés az FSR regiszter felhasználásával

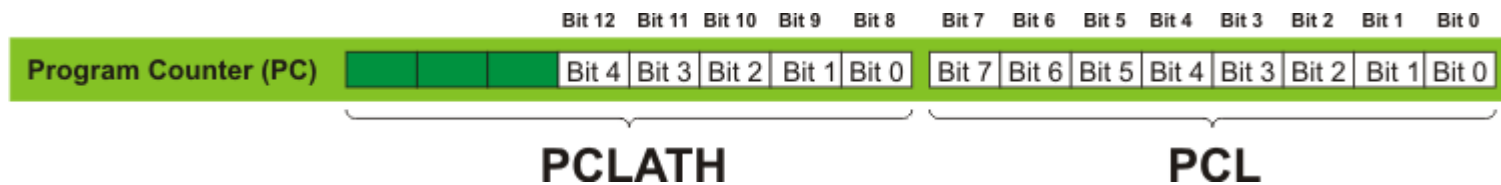


A PIC12F683 esetében csak az első két bank van implementálva.

A programmemória címzése



- ❑ A veremtár 8 szintű, kezelése hardveresen, automatikusan történik, szoftveresen nem hozzáférhető.
 - **Betöltés:** `CALL`, illetve megszakítás hatására.
 - **Visszatöltés:** `RETURN`, `RETLW`, vagy `RETFIE` hatására.
- ❑ A 2k-nál nagyobb memóriájú PIC midrange mikrovezérlők esetében a **GOTO** és **CALL** utasítások csak 2 K lapokban tudják kezelni a memóriát (**PIC12F683** esetén ez nem gond, mert csak 2 K a memória...)
- ❑ **PCL**-t manipuláló parancsoknál ügyelni kell, hogy nincs-e túlcscordulás (amikor **PCLATH** értékét növelni kell).



PIC12F683 konfigurációs bitek

—	—	—	—	FCMEN	IESO	BOREN1	BOREN0
bit 15				bit 8			

$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7				bit 0			

A hardver „viselkedését” befolyásoló, menet közben nem változtatható beállítások, a program beégetésekor kell megadni. Egyelőre csak a színezéssel jelöltekkel foglalkozunk.

FCMEN: Fail-Safe Clock Monitor Enabled bit

IESO: Internal External Switchover enable

BOREN<1:0>: Brown-out Reset Selection

CPD: Data Code Protection bit

CP: Code Protection bit

MCLRE: GP3/MCLR pin function select bit (MCLR láb RESET vagy I/O legyen)

PWRTE: Power-up Timer Enable bit

WDTE: Watchdog Timer Enable bit (Watchdog tiltás/engedélyezése)

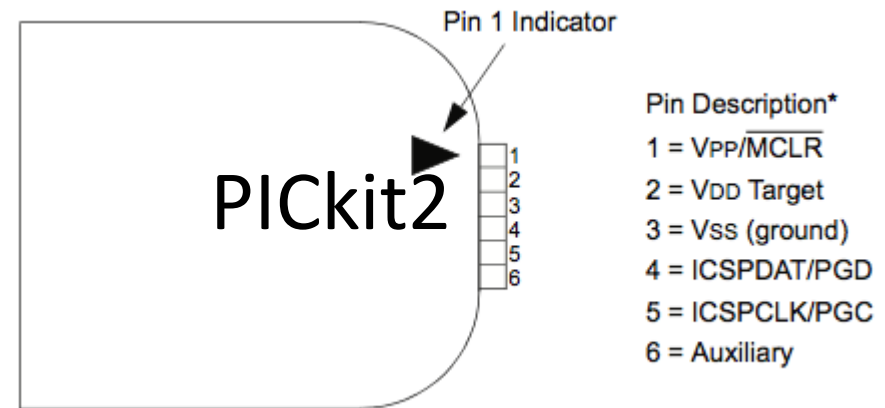
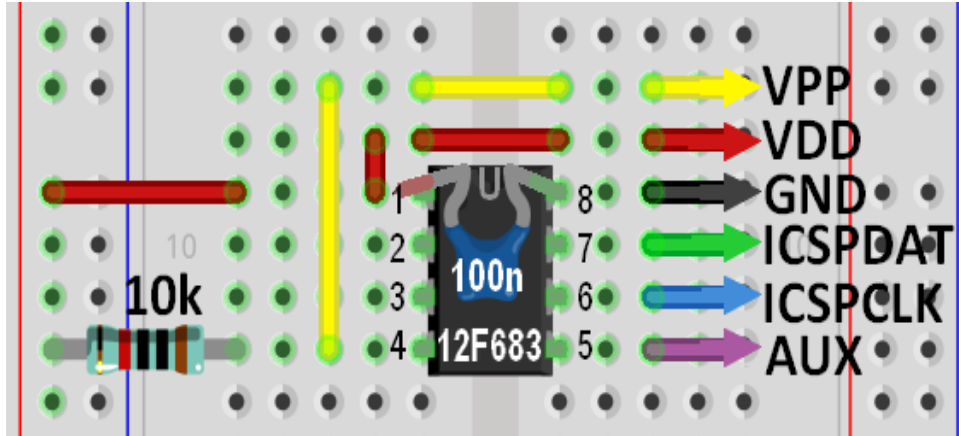
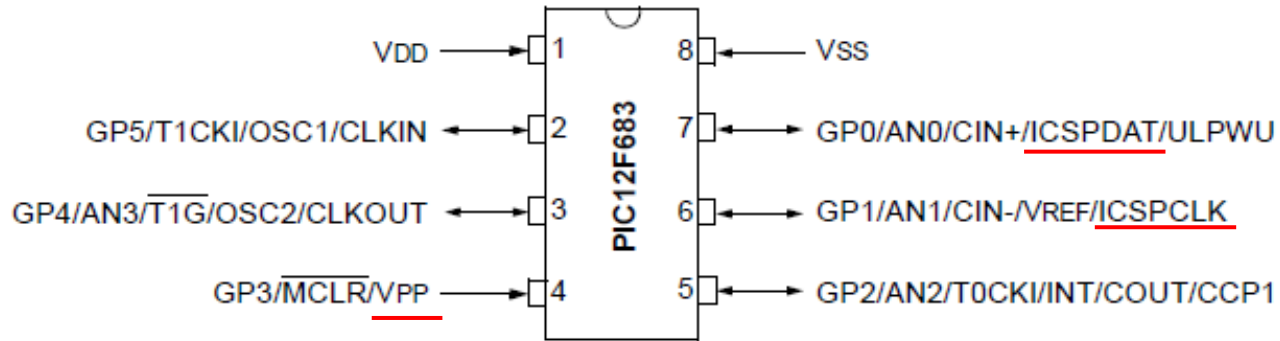
FOSC<2:0>: Oscillator Selection bits (órajelforrás választása)

Programok letöltése

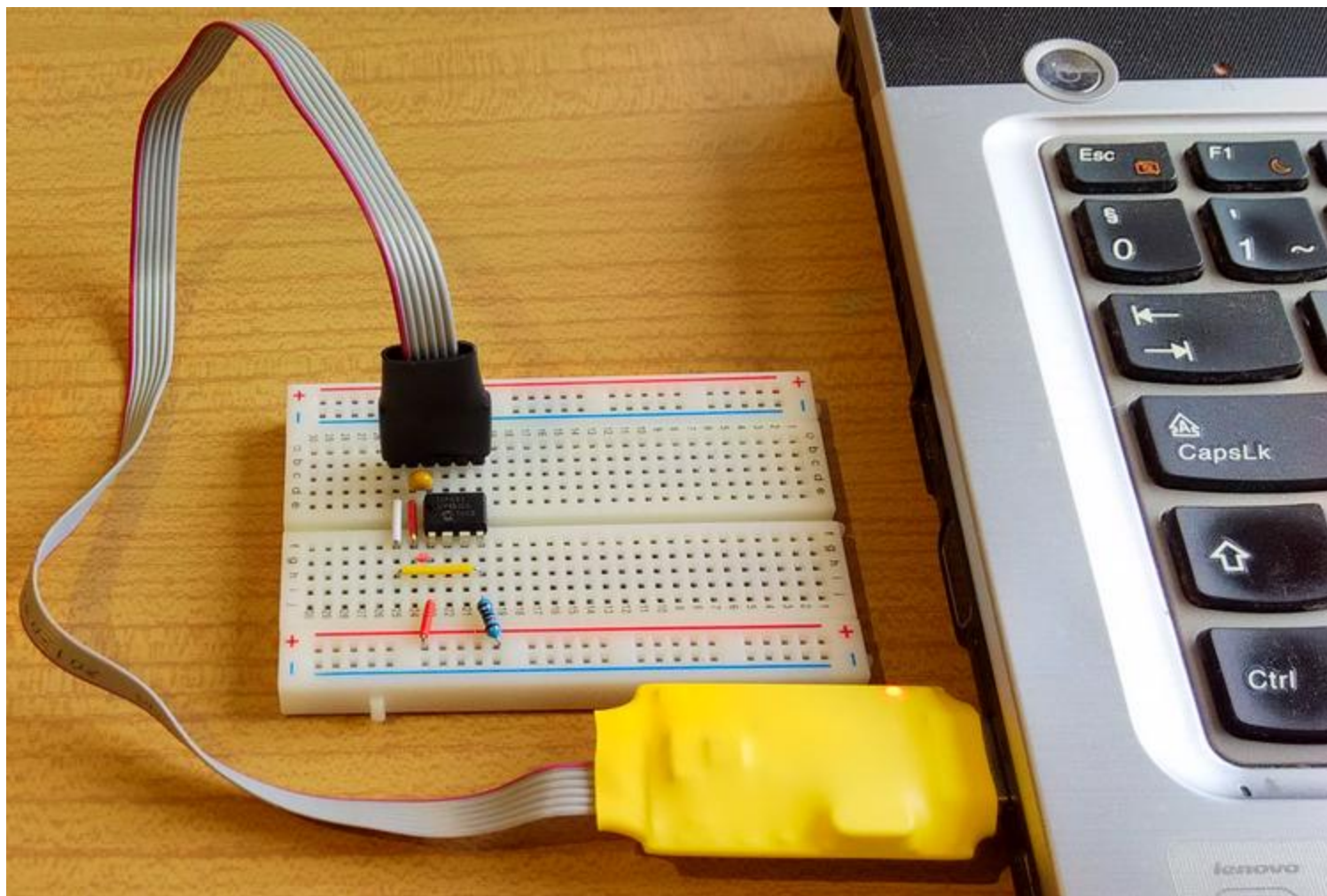
A programokat legegyszerűbben egy **PICkit2** programozóval tölthetjük le, amely saját programletöltő szoftver alkalmazással rendelkezik, tehát MPLAB telepítés nélkül is használható. Az **AUX** lábat esetünkben nem kell bekötni.

Figyelem! Programozás módban a VPP lábán 10 – 13 V-os feszültség jelenik meg!

Programozás közben a kommunikáció az ICSPDAT és ICSPCLK lábakon folyik.



Programok letöltése



Programletöltés iCP02v2 programozóval (PICkit2 klón)

A PICkit 2 V2 önálló alkalmazás

PICkit 2 Programmer - iCP02-V2.0

File Device Family Programmer Tools View Help

Midrange/Standard Configuration

Device: PIC12F683 Configuration: 3FFF

User IDs: FF FF FF FF

Checksum: 07FF OSCCAL: BandGap:

PICkit 2 connected. ID = iCP02-V2.0
PIC Device Found.

MICROCHIP

VDD PICkit 2
 On
 /MCLR 5,0

Read Write Verify Erase Blank Check

Program Memory

Enabled Hex Only Source: None (Empty/Erased)

000	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
008	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
010	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
018	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
020	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
028	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
030	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
038	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
040	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
048	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
050	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
058	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF

EEPROM Data

Enabled Hex Only

00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
10	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
30	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Auto Import Hex + Write Device

Read Device + Export Hex File

PICkit™ 2

PICkit 2 Programmer - iCP02-V2.0

File Device Family Programmer Tools View Help

Midrange/Standard Configuration

Device: PIC12F683 Configuration: 3FC4

User IDs: FF FF FF FF

Checksum: AAF7 OSCCAL: BandGap:

Programming Successful.

MICROCHIP

VDD PICkit 2
 On
 /MCLR 5,0

Read Write Verify Erase Blank Check

Program Memory

Enabled Hex Only Source: C:\...roject\PIC12F683\12f683_ledblink.hex

000	308F	1683	050F	3870	008F	019F	1283	019F	
008	3007	0099	019A	1683	1105	1283	1523	0823	
010	0085	1283	1303	3017	00A0	3067	00A1	3029	
018	00A2	120A	118A	0BA2	2819	120A	118A	0BA1	
020	2817	120A	118A	0BA0	2815	1123	0823	0085	
028	1283	1303	3017	00A0	3067	00A1	3029	00A2	
030	120A	118A	0BA2	2830	120A	118A	0BA1	282E	
038	120A	118A	0BA0	282C	280D	3FFF	3FFF	3FFF	
040	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
048	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
050	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
058	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	

EEPROM Data

Enabled Hex Only

00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
10	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
30	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Auto Import Hex + Write Device

Read Device + Export Hex File

PICkit™ 2

LED villogtatás (JAL nyelven)

A **JAL** (Just Another Language) Pascal-szerű programnyelvet és a fordítóprogramot **Wouter van Ooijen** alkotta meg a Microchip PIC mikrovezérlőihez. A **JALLIB** munkacsoport ezt kiegészítette perifériakönyvtárakkal és mintaprogramokkal. A szoftver ingyenesen használható.

```
1 -----
2 -- LED villogtatás Microchip pic12f683 mikrovezérlővel
3 -----
4 include 12f683                -- PIC céláramkör
5
6 pragma target CLOCK          8_000_000    -- oszcillátor frekvencia
7 pragma target OSC            INTOSC_NOCLKOUT -- belső oszcillátor 8MHz-en
8 pragma target WDT            disabled
9 pragma target PWRTE          enabled
10 pragma target MCLR           internal     -- MCLR láb GPIO legyen
11 OSCCON_IRCF = 0b_111           -- Fosc = 8 MHz beállítása
12
13 enable_digital_io()          -- mindegyik GPIO digitális legyen
14 alias led is pin_A2          -- GPIO2-re kötjük a LED-et
15 pin_A2_direction = output    -- GPIO2 legyen kimenet
16 --
17 forever loop                 -- Végtelen ciklus
18     led = on                  -- LED be (GPIO2 = 1)
19     _usec_delay(250_000)      -- 250 ms várakozás
20     led = off                 -- LED ki (GPIO2 = 0)
21     _usec_delay(250_000)      -- 250 ms várakozás
22 end loop
```

12f683_ledblink.jal

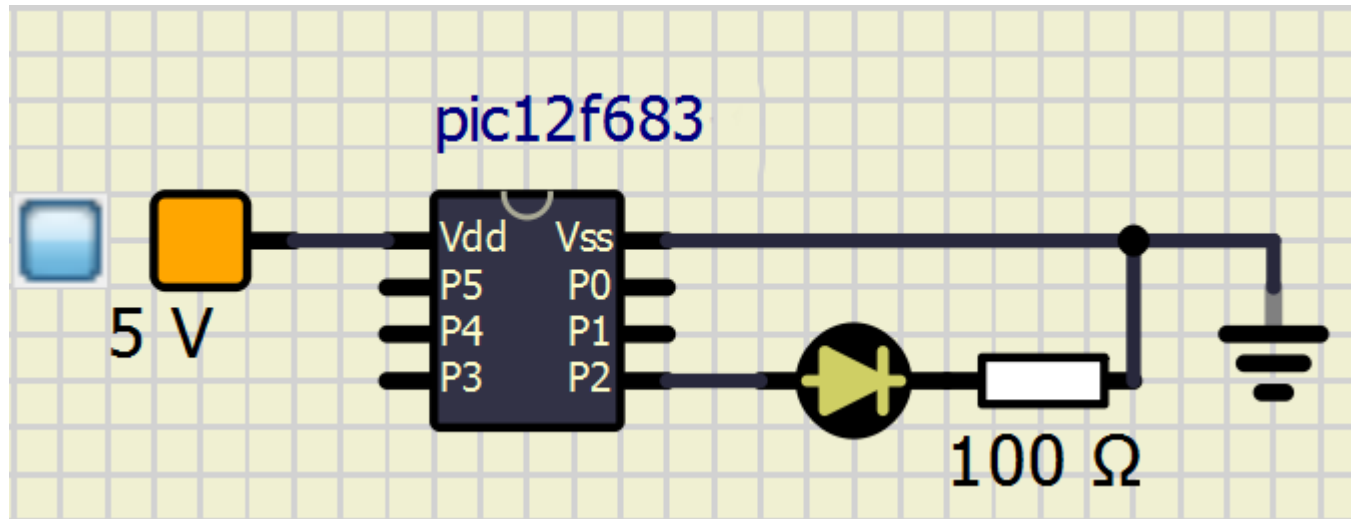
Ez a sor csak a fordítónak szól
Konfigurációs bitek megadása
Ez már utasítás az MCU-nak

Fejlesztői szoftver letöltés (Editor, fordító és periféria könyvtárak): jallib_full-1.2.0.zip
Dokumentáció és oktatási segédlet: www.justanotherlanguage.org

LED villogtatás

- ❖ Kössük egy LED anódját a **GPIO2** (PORTA 2. bit) kivezetésre!
- ❖ Kössük a LED katódját egy áramkorlátozó ellenálláson keresztül a **VSS** lábra!
- ❖ Töltsük be a mikrovezérlőbe a **12f682_ledblink.hex** programot!
- ❖ Indítsuk el a szimulációt és kapcsoljuk be a tápfeszültséget!

A **SimulIDE** környezetbe betölthető állomány: **12f683_ledblink.simu**

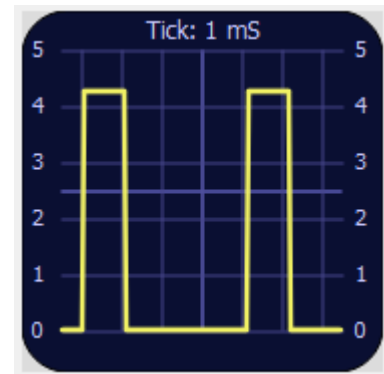
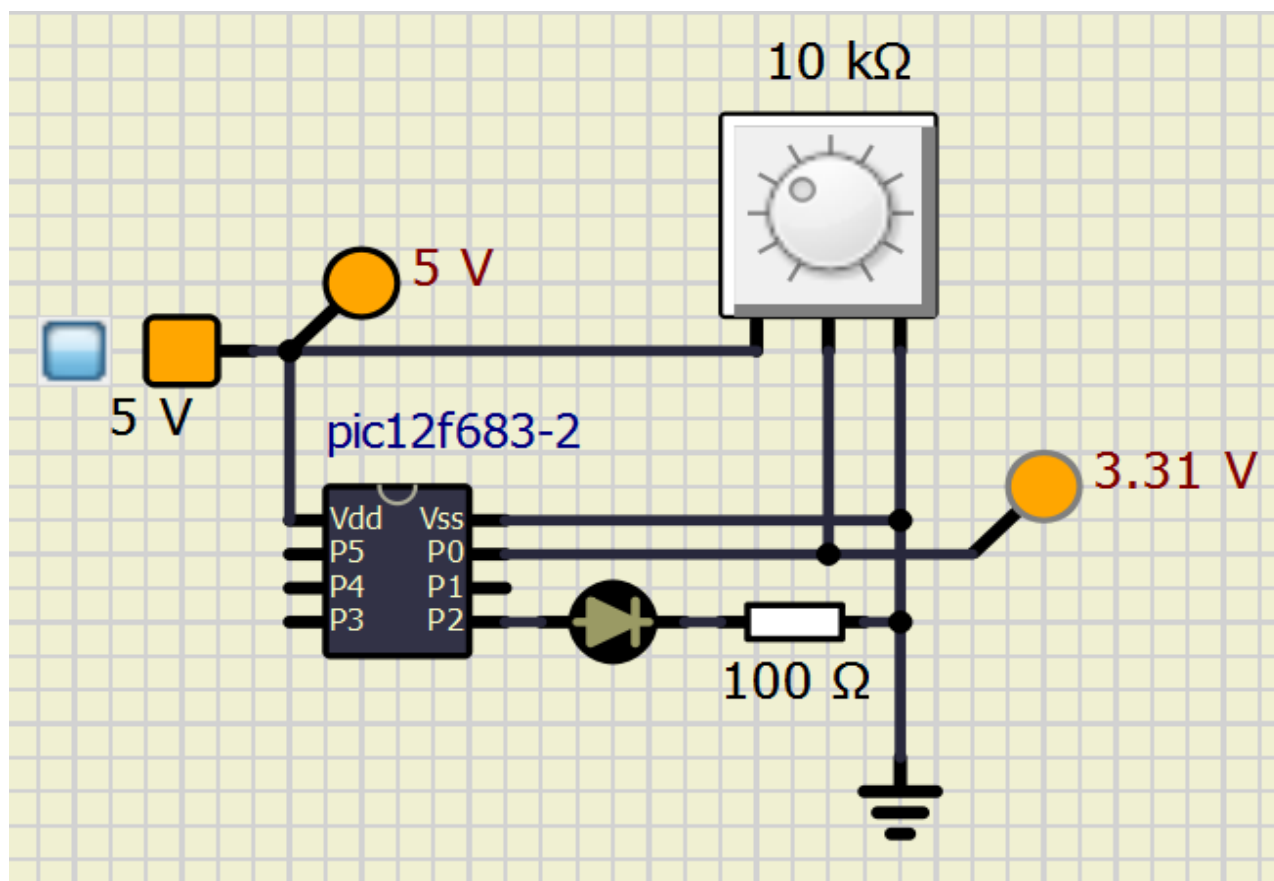


SimulIDE linkek: [Wiki](#) , [Letöltés](#)

PWM jel szabályozása potméterrel

A **GPIO0** (PORTA 0. bit) bemenetre a potméterrel leosztott tápfeszültséget vezetjük, amelyet egy ADC-vel megmérünk. A mért feszültséggel arányosan állítjuk be a kb. 488 Hz-es PWM jel kitöltését. **Megjegyzés: a szimulátor időskálája csali!**

A **SimulIDE** környezetbe betölthető állomány: **12f683_pwm_adc.simu**



PWM jel szabályozása potméterrel

```
-----  
-- Egycsatornás dimmer Microchip pic12f683 mikrovezérlővel  
-- Rob Hamerling mintapéldája, apróbb módosításokkal  
--  
-- Leírás:  
-- Ez a program egy egyszerű, egycsatornás dimmert valósít meg  
-- egy ADC analóg bemenet és egy PWM kimenet felhasználásával.  
-- Az MCU belső oszcillátorát használjuk alkatrész takarékosági okokból.  
-- A PWM szabályozása 256 lépésben bőven elegendő ehhez a feladathoz,  
-- ennek megfelelően a kis ADC felbontás is elegendő.  
-- Fosc = 8 MHz és 16-os előosztás esetén a PWM frekvencia 488 Hz lesz.  
-----  
include 12f683                                -- PIC céláramkör  
  
pragma target CLOCK      8_000_000           -- oszcillátor frekvencia  
pragma target OSC        INTOSC_NOCLKOUT     -- belső oszcillátor 8MHz-en  
pragma target WDT        disabled  
Progma target MCLR       internal  
OSCCON_IRCF    = 0b_111                      -- Fosc = 8 MHz beállítása  
  
enable_digital_io()                          -- mindegyik GPIO digitális legyen
```

PWM jel szabályozása potméterrel

```
-- ADC beállítás -----
const byte ADC_NVREF = ADC_NO_EXT_VREF
const ADC_RSOURCE = 10_000
const ADC_HIGH_RESOLUTION = FALSE
include adc
adc_init()

const byte ADC_CHANNEL = 0
set_analog_pin(ADC_CHANNEL)

-- PWM beállítás -----
include pwm_hardware
pwm_max_resolution(16)
pin_CCP1_direction = output
var byte measure

-- programhurok -----
forever loop
  measure = adc_read_low_res(ADC_CHANNEL)
  pwm1_set_dutycycle(measure)
  _usec_delay(20_000)
end loop
```

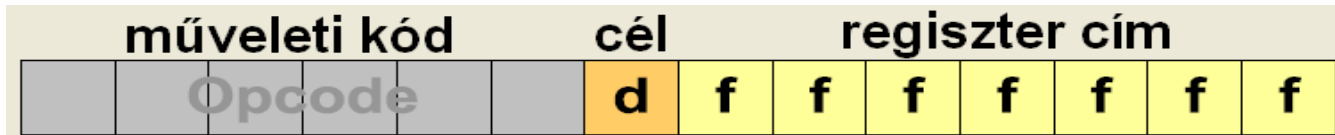
-- nincs külső Vref
-- Bemenet: 10K potméter
-- Kis felbontású ADC is elég!
-- ADC könyvtár becsatolása
-- ADC inicializálás

-- A potméter pin_AN0-hoz kötve
-- a választott csatorna beállítása

-- a PWM könyvtár becsatolása
-- Timer2 előosztási arány
-- a PWM-láb kimenet legyen
-- ADC érték / PWM kitöltés

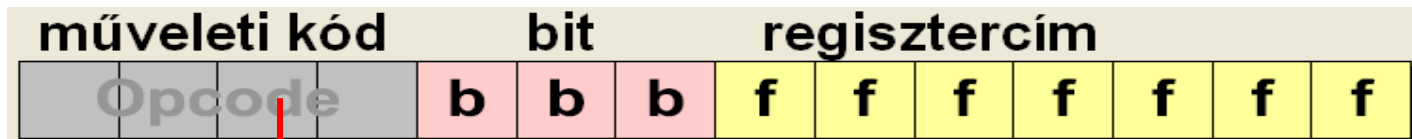
-- ADC kiolvasása
-- PWM kitöltés átírása
-- várunk egy kicsit...

RISC utasításkészlet 1. rész



Mnemonic	Leírás		Ciklus	14 bites műveleti kód				Flag
operandus				Msb				Lsb
Bájt orientált fájlregiszter utasítások								
ADDWF	f, d	$W + f \rightarrow d$	1	00	0111	dfff	ffff	C,DC,Z
ANDWF	f, d	$W \& f \rightarrow d$	1	00	0101	dfff	ffff	Z
CLRF	f	$0 \rightarrow f$	1	00	0001	1fff	ffff	Z
CLRW	-	$0 \rightarrow W$	1	00	0001	0xxx	xxxx	Z
COMF	f, d	$\neg f \rightarrow d$	1	00	1001	dfff	ffff	Z
DECF	f, d	$f - 1 \rightarrow d$	1	00	0011	dfff	ffff	Z
DECFSZ	f, d	$f - 1 \rightarrow d$, átlép ha 0	1 (2)	00	1011	dfff	ffff	
INCF	f, d	$f + 1 \rightarrow d$	1	00	1010	dfff	ffff	Z
INCFSZ	f, d	$f + 1 \rightarrow d$, átlép ha 0	1 (2)	00	1111	dfff	ffff	
IORWF	f, d	$W \parallel f \rightarrow d$	1	00	0100	dfff	ffff	Z
MOVF	f, d	$f \rightarrow d$	1	00	1000	dfff	ffff	Z
MOVWF	f, d	$W \rightarrow f$	1	00	0000	dfff	ffff	
NOP	-	-	1	00	0000	0xx0	0000	
RLF	f, d	$f(n) \rightarrow d(n+1)$, $C \rightarrow d(0)$, $f(7) \rightarrow C$	1	00	1101	dfff	ffff	C
RRF	f, d	$f(n) \rightarrow d(n-1)$, $C \rightarrow d(7)$, $f(0) \rightarrow C$	1	00	1100	dfff	ffff	C
SUBWF	f, d	$f - W \rightarrow d$ [$f + \neg W + 1 \rightarrow d$]	1	00	0010	dfff	ffff	C,DC,Z
SWAPF	f, d	$f(0-3) \leftrightarrow f(4-7)$	1	00	1110	dfff	ffff	
XORWF	f, d	$W \oplus f \rightarrow d$	1	00	0110	dfff	ffff	Z

RISC utasításkészlet 2. rész



Bit orientált fájlregiszter utasítások

Utasítás	Operandumok	Leírás	Bit	Op	Op2	Op1	Op0	Op3	Op4	Op5	Op6
BCF	f, b	0 → f(b)	1	01	00bb	bfff	ffff				
BSF	f, b	1 → f(b)	1	01	01bb	bfff	ffff				
BTFSC	f, b	Ha f(b) = 0 akkor átlép	1 (2)	01	10bb	bfff	ffff				
BTFSS	f, b	Ha f(b) = 1 akkor átlép	1 (2)	01	11bb	bfff	ffff				

Állandót tartalmazó utasítások

Utasítás	Operandumok	Leírás	Bit	Op	Op2	Op1	Op0	Op3	Op4	Op5	Op6
ADDLW	k	$k + W \rightarrow W$	1	11	111x	kkkk	kkkk	C, DC, Z			
ANDLW	k	$k \& W \rightarrow W$	1	11	1001	kkkk	kkkk	Z			
SUBLW	k	$W - k \rightarrow W$	1	11	110x	kkkk	kkkk	C, DC, Z			
IORLW	k	$k \parallel W \rightarrow W$	1	11	1000	kkkk	kkkk	Z			
XORLW	k	$k \oplus W \rightarrow W$	1	11	1010	kkkk	kkkk	Z			
RETLW	k	$k \rightarrow W, TOS \rightarrow PC$	2	11	01xx	kkkk	kkkk				



Vezérlő utasítások

Utasítás	Operandumok	Leírás	Bit	Op	Op2	Op1	Op0	Op3	Op4	Op5	Op6
GOTO	k	$k \rightarrow PC$ (11 bit)	2	10	1kkk	kkkk	kkkk				
CALL	k	$PC + 1 \rightarrow TOS, k \rightarrow PC$	2	10	0kkk	kkkk	kkkk				
RETURN	-	$TOS \rightarrow PC$	2	00	0000	0000	1000				
RETFIE	-	$TOS \rightarrow PC$	2	00	0000	0000	1001				
CLRWDT	-	0 → WDT (az előosztó is, ha van)	1	00	0000	0110	0100	/TO, /PD			
SLEEP	-	0 → WDT (oszillátor stop)	1	00	0000	0110	0011	/TO, /PD			

MOVLW	k	$k \rightarrow W$ (konstans betöltése)	1	11	00xx	kkkk	kkkk				
-------	---	--	---	----	------	------	------	--	--	--	--

Memóriatérkép

File Address	File Address	File Address	File Address
Indirect addr.(1)	00h	Indirect addr.(1)	80h
TMR0	01h	OPTION_REG	81h
PCL	02h	PCL	82h
STATUS	03h	STATUS	83h
FSR	04h	FSR	84h
GPIO	05h	TRISIO	85h
	06h		86h
	07h		87h
	08h		88h
	09h		89h
PCLATH	0Ah	PCLATH	8Ah
INTCON	0Bh	INTCON	8Bh
PIR1	0Ch	PIE1	8Ch
	0Dh		8Dh
TMR1L	0Eh	PCON	8Eh
TMR1H	0Fh	OSCCON	8Fh
T1CON	10h	OSCTUNE	90h
TMR2	11h		91h
T2CON	12h	PR2	92h
CCPR1L	13h		93h
CCPR1H	14h		94h
CCP1CON	15h	WPU	95h
	16h	IOC	96h
	17h		97h
WDTCON	18h		98h
CMCON0	19h	VRCON	99h
CMCON1	1Ah	EEDAT	9Ah
	1Bh	EEADR	9Bh
	1Ch	EECON1	9Ch
	1Dh	EECON2(1)	9Dh
ADRESH	1Eh	ADRESL	9Eh
ADCON0	1Fh	ANSEL	9Fh
General Purpose Registers 96 Bytes	20h	General Purpose Registers 32 Bytes	A0h
			BFh
			C0h
			EFh
		Accesses 70h-7Fh	F0h
			FFh
BANK 0	7Fh	BANK 1	

Közvetlenül egyszerre csak az egyik memória bank érhető el. A bankváltás esetünkben a **STATUS** regiszter **RPO** bitjének átbillentésével történik.

Hordozhatóbb a kód, ha a **BANKSEL** direktívát használjuk. Például:

BANKSEL TRISIO

Vannak átfedő memória-tartományok, illetve regiszterek. Ezek mindig elérhetőek, bankváltás nélkül is.

LED villogtatás assembly programmal

```
list P=12F683, ST=OFF           ; Turnoff Symbol Table in List file.
errorlevel -302                 ; Ignore error message when storing to Bank 1.
#include "p12f683.inc"           ; processor specific variable definitions
```

```
__config _INTOSCIO & _WDT_OFF & _PWRTE_ON & _MCLRE_OFF & _CP_OFF & _CPD_OFF &
        _BOD_NSLEEP & _IESO_OFF & _FCMEN_OFF
```

```
ORG      0x000                   ; processor reset vector
```

MAIN:

```
BANKSEL CMCON0                   ; Switch to Bank 0.
MOVLW   b'00000111'              ; Turn off Analog Comparator.
MOVWF   CMCON0
CLRF    GPIO
BANKSEL TRISIO                   ; Switch to Bank 1.
MOVLW   ~(1<<GP2)                ; configure GP2 (only) as an output
MOVWF   TRISIO
CLRF    ANSEL                    ; Enable digital I/O function
MOVLW   b'01110001'              ; 8MHz Clk, IntOsc, SysClk via IntOsc
MOVWF   OSCCON
banksel GPIO                      ; Switch to Bank 0.
```

LOOP:

```
MOVLW   1<<GP2
XORWF   GPIO,F                   ; toggle LED
CALL    Delay
goto    LOOP
```

A programot a SimulIDE szimulátorban majd a **12f683_ledblink.simu** kapcsolással próbálhatjuk ki.

250 ms késleltető függvény

```
cblock
d1
d2
d3
endc

; Delay = 0.25 seconds
; Clock frequency = 8 MHz
Delay

    movlw    0x03
    movwf   d1
    movlw   0x18
    movwf   d2
    movlw   0x02
    movwf   d3

Delay_0
    decfsz  d1, f
    goto   $+2
    decfsz  d2, f
    goto   $+2
    decfsz  d3, f
    goto   Delay_0
    goto   $+1
    return
```

Delay: 0.25

Instruction cycles
 Seconds

Temporary registers names: d1 d2 d3 d4

Clock frequency: 8 MHz

Generate routine: Delay

Select CPU: PIC SX

Generate code!

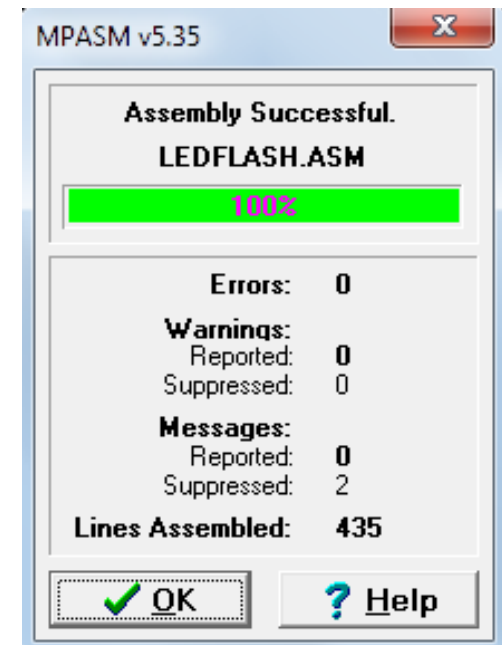
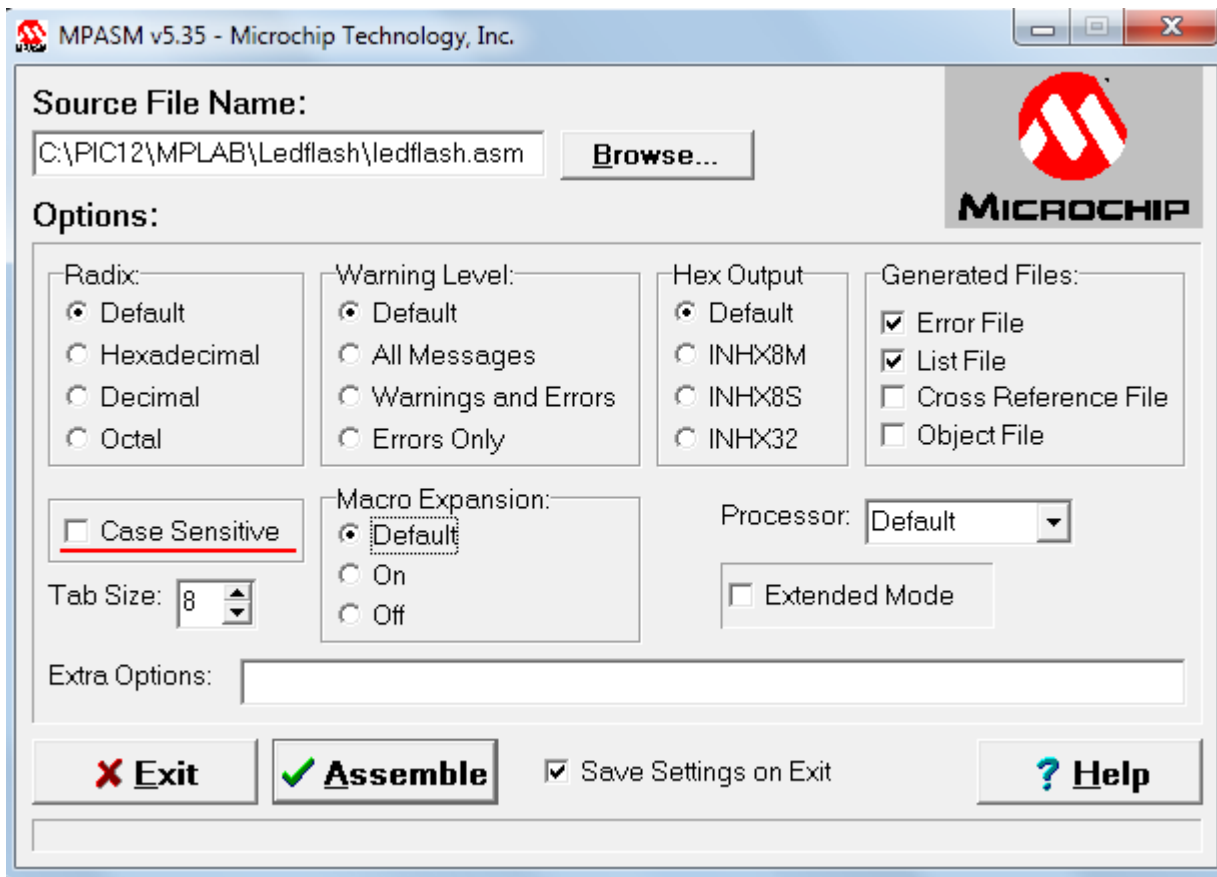
Kód generálás indítása

Az on-line használható, késleltető kódot generáló programot Nyikoláj Golovcsenkó írta PIC illetve SX mikrovezérlőkhöz.
Link: [Microchip PIC Delay Code Generator](#)

MPASM makroassembler

Az **MPASM** makroassembler ingyenes szoftver, az **MPLAB IDE** tartalmazza, de önállóan is használható. Utóbbi esetben gondoskodjunk róla, hogy a használni kívánt mikrovezérlő leíró állománya a program rendelkezésére álljon (esetünkben a **p12f683.inc** állomány).

Kapcsoljuk ki a kis- és nagybetűket megkülönböztető módot, válasszuk ki a forrásállományt és kattintsunk az **Assemble** gombra! Sikeres fordítás esetén zöld sáv jelzi az eredményességet.



A fenti példában a letöltendő fájl neve **ledflash.HEX** lesz.

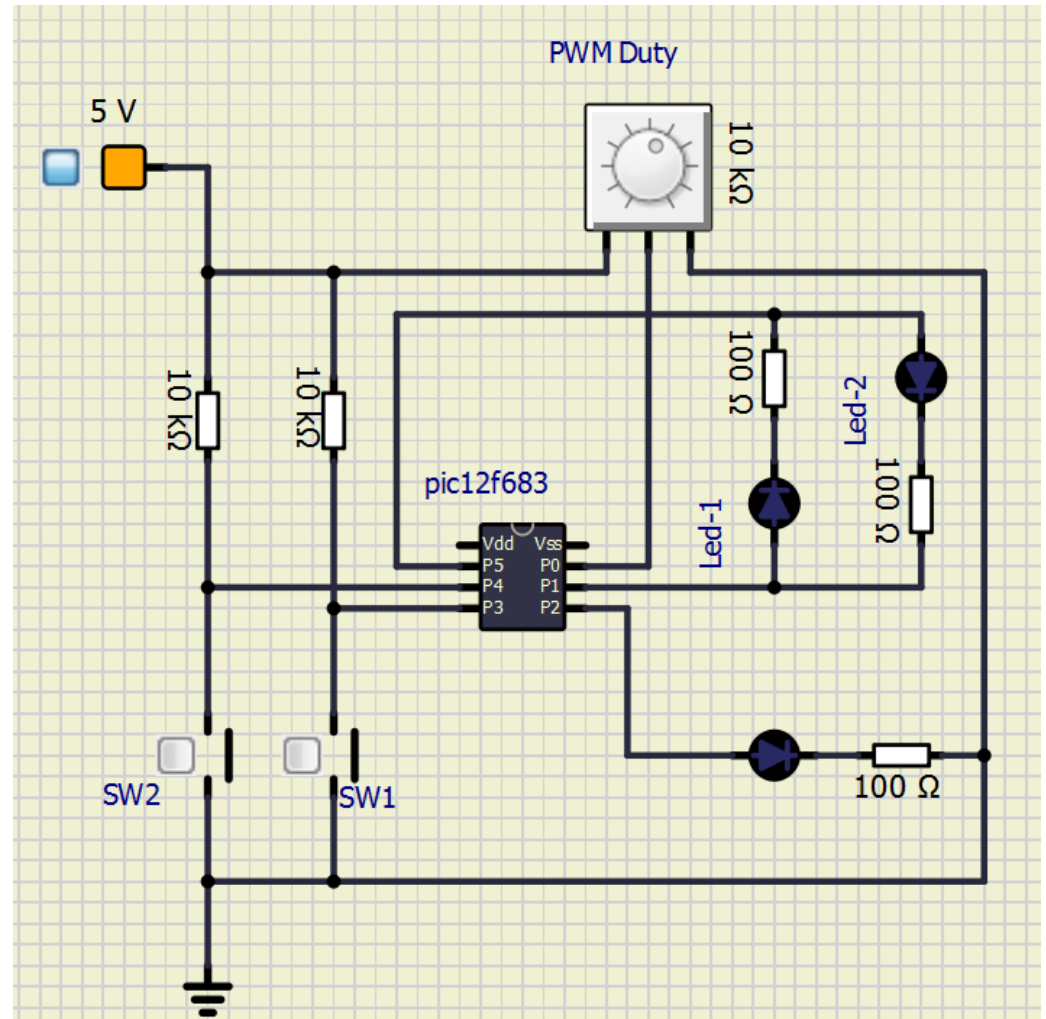
PWM mintapélda (SimulIDE)

Az alábbi kapcsolás és a hozzá tartozó program a **SimulIDE** program egyik mintapéldája.

A **GPIO1** és **GPIO5** kimenetek H-hídként viselkednek, polaritásváltó vezérlésre mutatnak példát. **GPIO1** az **SW1** nyomógommbal, **GPIO5** pedig az **SW2** nyomógommbal kapcsolgatható magas, ill. alacsony szintre.

A **GPIO0** analóg bemenetre kapcsolt feszültséget az ADC-vel megmérjük és a **PWM** jel kitöltését szabályozzuk vele arányosan.

GPIO2 a **PWM** kimenet. Ha **GPIO1** és **GPIO5** egyaránt alacsony szinten van, akkor a **PWM** jelet lekapcsoljuk.



Áramkör: **PWM_Switch_Pic12F683.simu**

Firmware: **PWM_Switch_Pic12F683.asm**

```

list      P=12F683, ST=OFF    ; Turnoff Symbol Table in List file.
#include   <p12f683.inc>
__config _INTOSCIO & _WDT_OFF & _PWRTE_ON & _MCLRE_OFF & _CP_OFF &
_CP_OFF & _BOD_NSLEEP & _IESO_OFF & _FCMEN_OFF

```

```

CBLOCK   0x20                ; Assign each reg. from Bank 0 RAM area.
    count1
    count2
    count3
    count4
    temp
    RESULTHI
    RESULTLO

```

```

ENDC                ; Conclude Bank 0 RAM assignments.

```

```

ORG      0x000          ; processor reset vector
GOTO     setup

```

setup

```

BANKSEL  OSCCON        ; Switch to Bank 1.
MOVLW   b'01100001'   ; 4MHz Clk, IntOsc, SysClk via IntOsc
MOVWF   OSCCON
BANKSEL  CMCON0        ; Switch to Bank 0.
MOVLW   b'00000111'   ; Turn off Comparator.
MOVWF   CMCON0
BANKSEL  ANSEL         ; Switch to Bank 1.
CLRF    ANSEL          ; Set I/O pins to Digital.
CLRF    TRISIO         ; all output
BSF     TRISIO, GP3    ; GP3 input
BSF     TRISIO, GP4
BANKSEL  GPIO          ; Switch to Bank 0.
CLRF    GPIO

```

```

;--- This code block configures the ADC for polling,
    BANKSEL TRISIO ; bank 1
    BSF      TRISIO, 0 ; Set GP0 for ADC input
    BANKSEL ANSEL ; 0x9F P32
    MOVLW   b'01110001' ; ADC Frc clock,
    IORWF   ANSEL ; and GP0 as analog
    BANKSEL ADCON0 ; 0x1F
    MOVLW   b'00000001' ; Left justify,
    MOVWF   ADCON0 ; Vdd Vref, AN0, On
    CALL    delay_1ms ; Acquisiton delay
;--- setup PWM CCP1CON in bank 0 bits 4-5 are LSB of 10-bit PWM
    movlw   b'00000111'
    movwf   T2CON ; turn on TMR2 prescale 16 - frequency
    movlw   d'127' ; duty cycle = TMR2 = CCPR1L:CCP1CON<5:4>
    movwf   CCPR1L
    movlw   b'00001110'
    movwf   CCP1CON ; turn on PWM
loop
    CALL    ADCtoPWM
    BTFSS   GPIO, GP3 ; SW1
    CALL    toggleGP1
    BTFSS   GPIO, GP3
    GOTO    $-1 ; wait for release
    BTFSS   GPIO, GP4
    CALL    toggleGP5
    BTFSS   GPIO, GP4
    GOTO    $-1 ; wait for release
    BTFSC   GPIO, 1
    goto    $+5
    BTFSC   GPIO, 5
    goto    $+3
    CALL    PWMOFF
    goto    loop
    Call    PWMON
    goto    loop

```

```

PWMOFF
    BANKSEL TRISIO
    BSF TRISIO, GP2
    BANKSEL GPIO
    CALL delay_1ms
    RETURN

PWMON
    BANKSEL TRISIO
    BCF TRISIO, GP2
    BANKSEL GPIO
    CALL delay_1ms
    RETURN

; XOR toggle
toggleGP1 ; LED on GP1
    movlw 2
    xorwf GPIO, f
    CALL delay_1ms
    return

; non-XOR toggle
toggleGP5 ; LED on GP5
    BTFSS GPIO, 5
    goto $+3
    BCF GPIO, 5
    goto $+2
    BSF GPIO, 5
    CALL delay_1ms
    return

```

readADC0

```
BSF      ADCON0,GO ;Start conversion
BTFSC   ADCON0,GO ;Is conversion done?
GOTO    $-1 ;No, test again
MOVF    ADRESH,W ;Read upper 8 bits
MOVWF   RESULTHI ;Store in GPR space
; ADRESL bits 6, 7 LSB of ADC
BANKSEL ADRESL
MOVWF   ADRESL
BANKSEL ADRESH ; bank 0
; right shift bits 7, 6 to bits 4, 5
MOVWF   RESULTLO
RRF     RESULTLO, f
RRF     RESULTLO, f
return

;#####
```

ADCtoPWM

```
CALL   readADC0
MOVWF  RESULTHI
MOVWF  CCP1L
; RESULTLO bits 5, 4 LSB of ADC
; CCP1CON bit 5,4 LSB PWM
;MOVLW  b'00001110'
;IORWF  RESULTLO, w
;MOVWF  CCP1CON ; turn on PWM
return

;#####
```

```
; delay routines *****
delay_1ms
; 4 mHz crystal
movlw  D'2'
movwf  count1
aa     movlw  D'167'
decfsz count2, F
goto  $-1 ; two cycles
decfsz count1, F
goto  aa
return

delay_100ms
movlw  d'100'
movwf  count3
bb     call  delay_1ms
decfsz count3, F
goto  bb
return
```

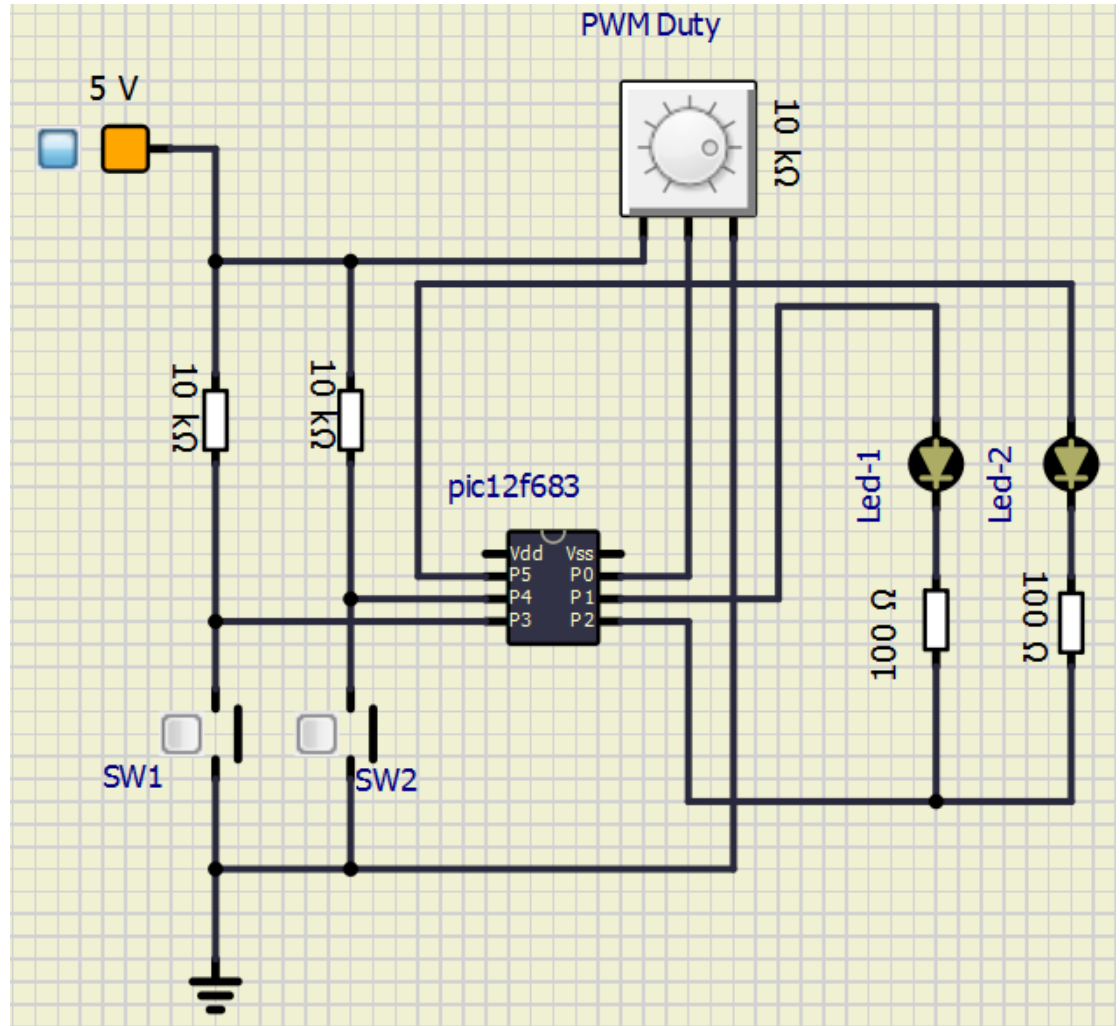
Egy másik kapcsolás...

A **SimulIDE** előzőekben ismertetett mintapéldájához egy másik kapcsolás is tartozik, amely ugyanazzal a firmware programmal, de egy kicsit másként viselkedik:

A **GPIO0** analóg bemenet most is a **GPIO2** kimeneten megjelenő **PWM** jel szabályozására szolgál.

GPIO1 és **GPIO5** az előzőekhez hasonlóan most is az **SW2**, ill. **SW1** nyomógombokkal kapcsolgatható ki-be.

A LED-ek most unipoláris vezérlést kapnak, a **PWM** jel mindkettő áramát azonos módon szabályozza.



Áramkör: **PWM_Switch_Pic12F683-2.simu**
Firmware: **PWM_Switch_Pic12F683.asm**

PIC programozás Arduinoval

PIC12F675
PIC12F683
PIC16F691

```
Rendszergazda: C:\Windows\System32\cmd.exe
Microsoft Windows [verziószám: 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Minden jog fenntartva.

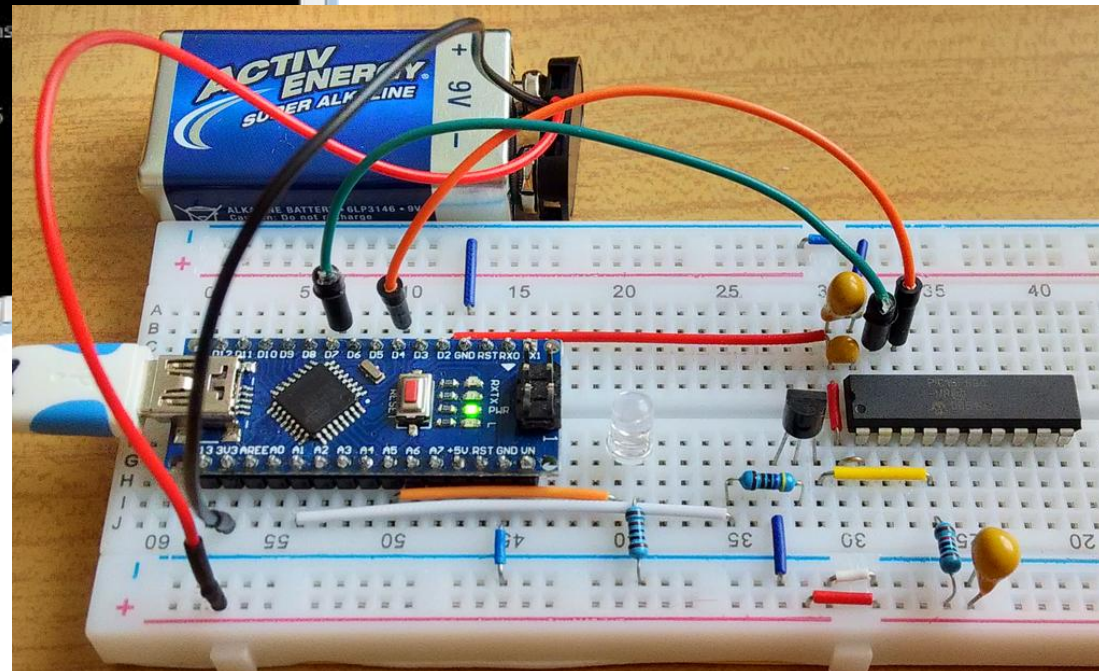
c:\Picprog>PicProgHost.exe -p COM7 --list-devices
PicProgHost version 1.0, Copyright (c) 2015 www.pikoder.com
PicProgHost comes with ABSOLUTELY NO WARRANTY; for details
type 'PicProgHost --warranty'. This is free software,
and you are welcome to redistribute it under certain conditions;
type 'picProgHost --copying' for details.

Initializing programmer ...
Supported devices:
pic12f629*, pic12f675*, pic16f630*, pic16f676*, pic16f84, pic16f84a*,
pic16f87*, pic16f88*, pic16f627*, pic16f627a*, pic16f628*, pic16f628a*,
pic16f648a*, pic16f684*, pic16f690*, pic16f877*, pic16f882*, pic16f883*,
pic16f884*, pic16f886*, pic16f887*
* = autodetected

c:\Picprog>PicProgHost.exe -p COM7 --ihx32 -i Rotate.HEX --erase --burn
PicProgHost version 1.0, Copyright (c) 2015 www.pikoder.com
PicProgHost comes with ABSOLUTELY NO WARRANTY; for details
type 'PicProgHost --warranty'. This is free software,
and you are welcome to redistribute it under certain conditions;
type 'picProgHost --copying' for details.

Initializing programmer ...
Device pic16f690, program memory: 4096 words, data memory: 256
Erasing and removing code protection.
Burning program memory, 16 locations,
burning data memory, 0 locations,
burning id words and fuses, 1 location,
done.

c:\picprog>
```



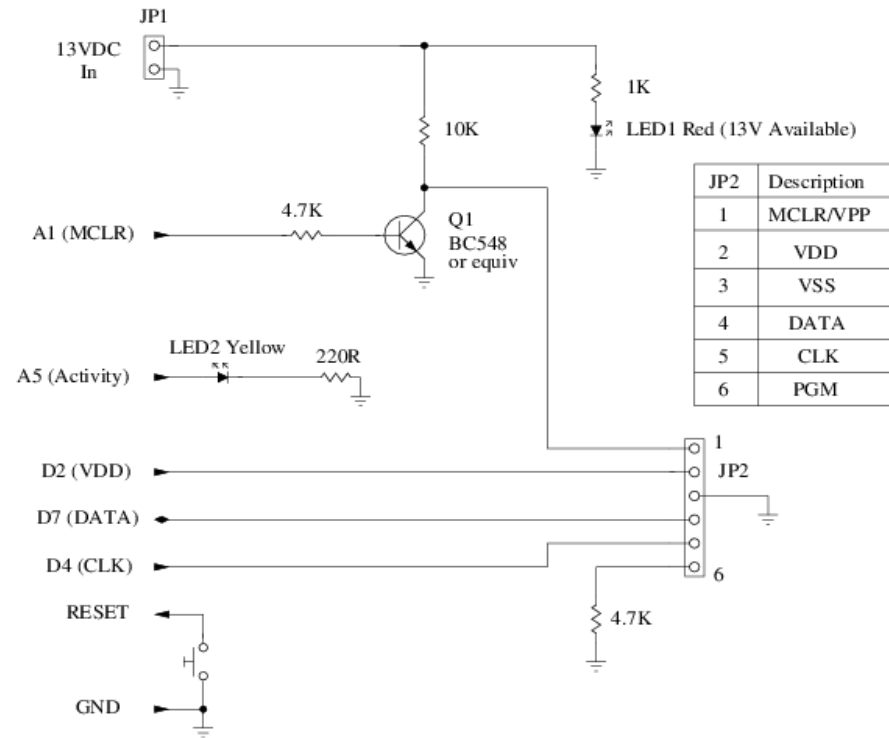
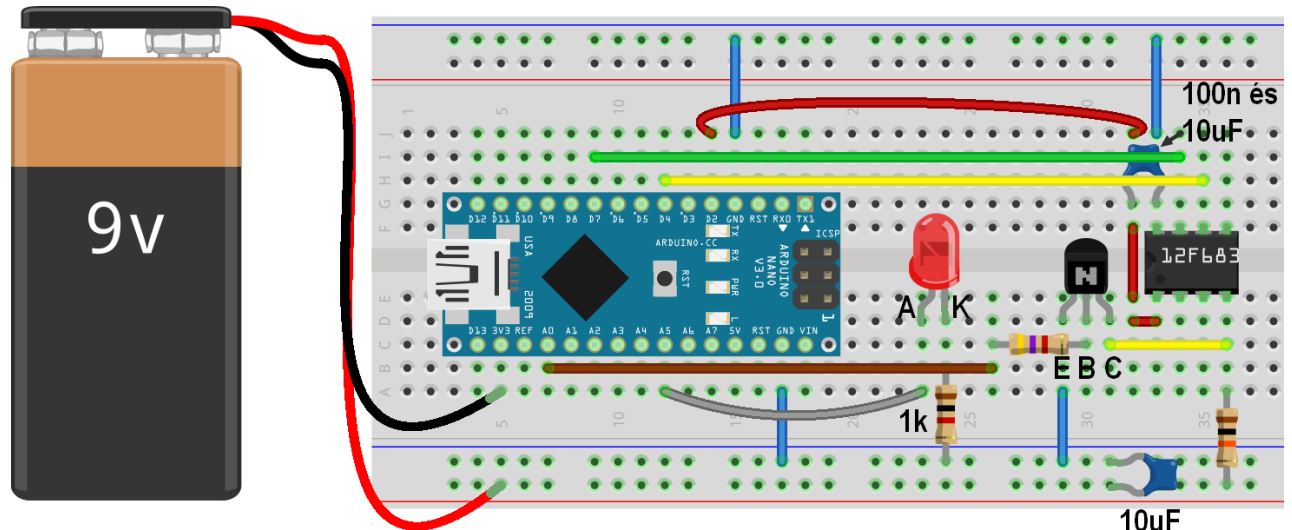
A kapcsolás

A kiindulási alap Rhys Weatherley kapcsolása és programja.

Link: rweather.github.io/ardpicprog/index.html

Módosítás: A kapcsolást csupán annyiban módosítottuk, hogy külső 12 v-os tápegység helyett egy 9 V-os elemet használunk, melyet szükség esetén ráültetünk az Arduino 3,3 V-os kimenetére.

A mellékelt ábra szerint a kapcsolást próbapanelon is gyorsan összerakhatjuk. Ügyeljünk a tranzisztor lábkiosztására! A BC jelzésűeket fordítva (háttal kifelé) kell elhelyezni!



Arduino firmware – 1. változat

Az eredeti firmware a http://rweather.github.io/ardpicprog/programpic_sketch.html címen található (Rhys Weatherley). A programot bővíteni kell, mert az általunk használt típusok közül sem a **PIC12F683**-at, sem a **PIC16F690**-et nem támogatja.

A firmware módosítása

1. A `// Device names, forced out into PROGMEM` szekcióba vegyük fel az alábbi sorokat is!

```
const char s_pic12f683[]  PROGMEM = "pic12f683";  
const char s_pic16f690[]  PROGMEM = "pic16f690";
```

2. A `struct deviceInfo const devices[]` stuktúratömbbe vegyük fel az alábbi sorokat is!

```
// http://ww1.microchip.com/downloads/en/DeviceDoc/40001204J.pdf  
{s_pic12f683, 0x0460, 2048, 0x2000, 0x2100, 8, 256, 0, 0, FLASH4, EEPROM},  
{s_pic16f690, 0x1400, 4096, 0x2000, 0x2100, 8, 256, 0, 0, FLASH4, EEPROM},
```

3. Fordítsuk le és töltsük le a firmware-t egy Arduino (Atmega 328) kártyába!
Lehetőleg 1.05 vagy 1.06 verziójú Arduino IDE-t használjunk!

Arduino firmware – 2. változat

Az előző oldalon bemutatott eredeti firmware-nek található egy módosított változata is:

Link: github.com/makerprojects/ProgramPic. Ez támogat néhány más típust is (többek között a **PIC16F690**-et is), de a **PIC12F683** ebben sincs benne.

A firmware módosítása

1. A `// Device names, forced out into PROGMEM` szekcióba vegyük fel az alábbi sorokat is!

```
const char s_pic12f683[] PROGMEM = "pic12f683";
```

2. A `struct deviceInfo const devices[]` stuktúratömbbe vegyük fel az alábbi sorokat is!

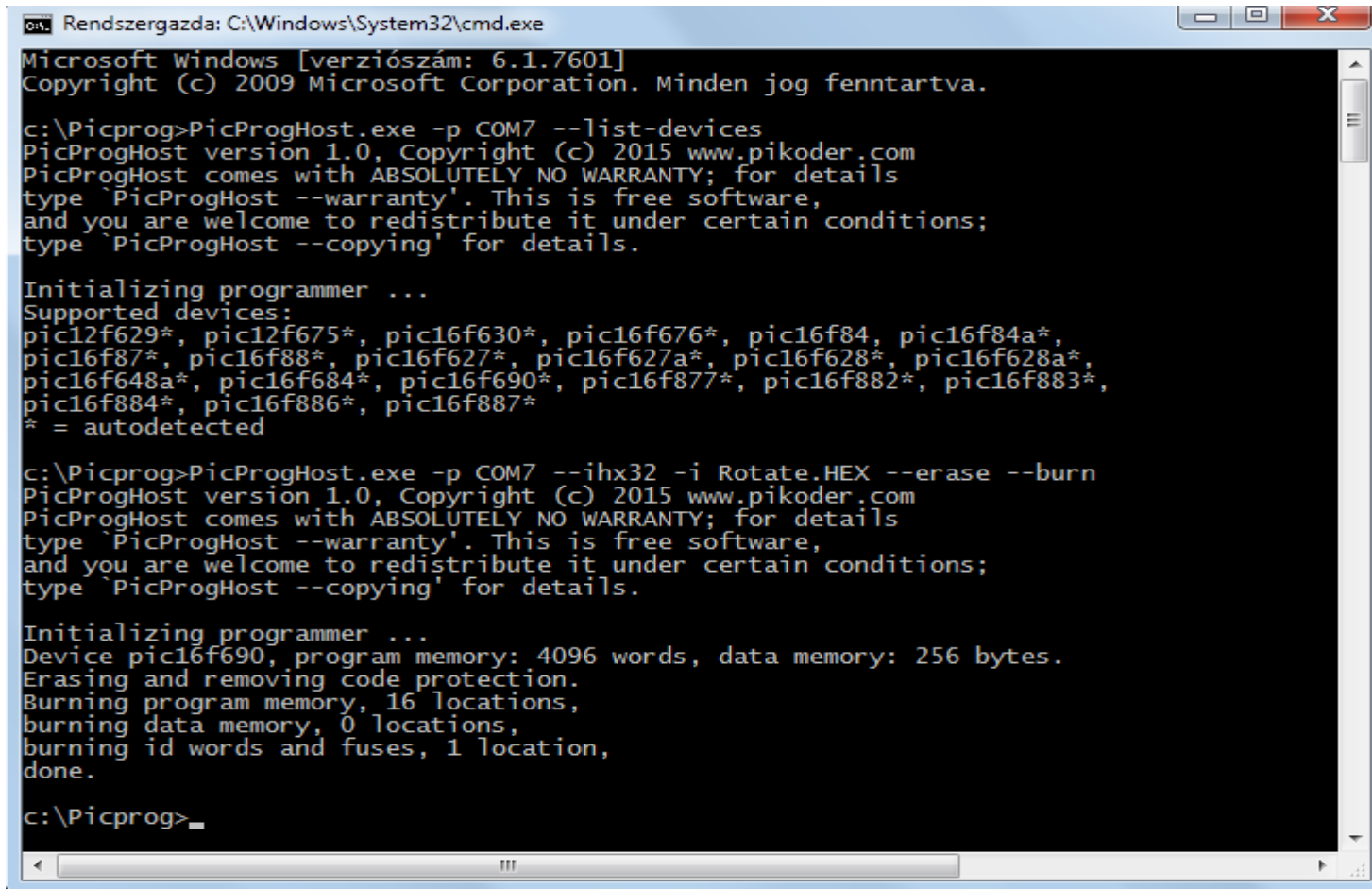
```
// http://ww1.microchip.com/downloads/en/DeviceDoc/40001204J.pdf  
{s_pic12f683, 0x0460, 2048, 0x2000, 0x2100, 8, 256, 0, 0, FLASH4, EEPROM},
```

3. Fordítsuk le és töltsük le a firmware-t egy Arduino (Atmega 328) kártyába!
Ennek lefordításához már 1.6.9 verziójú Arduino IDE-t használtunk!

PC oldali letöltőprogram

Mindkét firmware-hez jól használható a **PicProgHost** parancssoros program.

- Honlap, leírás: www.pikoder.de/ArdPicProgEN.htm
- Futtatható program Windows 7-hez: www.pikoder.de/download/PicProgHost.zip
- Forráskód: github.com/makerprojects/PicProgTerm



```
Microsoft Windows [verziószám: 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Minden jog fenntartva.

c:\Picprog>PicProgHost.exe -p COM7 --list-devices
PicProgHost version 1.0, Copyright (c) 2015 www.pikoder.com
PicProgHost comes with ABSOLUTELY NO WARRANTY; for details
type `PicProgHost --warranty`. This is free software,
and you are welcome to redistribute it under certain conditions;
type `PicProgHost --copying` for details.

Initializing programmer ...
Supported devices:
pic12f629*, pic12f675*, pic16f630*, pic16f676*, pic16f84, pic16f84a*,
pic16f87*, pic16f88*, pic16f627*, pic16f627a*, pic16f628*, pic16f628a*,
pic16f648a*, pic16f684*, pic16f690*, pic16f877*, pic16f882*, pic16f883*,
pic16f884*, pic16f886*, pic16f887*
* = autodetected

c:\Picprog>PicProgHost.exe -p COM7 --ihx32 -i Rotate.HEX --erase --burn
PicProgHost version 1.0, Copyright (c) 2015 www.pikoder.com
PicProgHost comes with ABSOLUTELY NO WARRANTY; for details
type `PicProgHost --warranty`. This is free software,
and you are welcome to redistribute it under certain conditions;
type `PicProgHost --copying` for details.

Initializing programmer ...
Device pic16f690, program memory: 4096 words, data memory: 256 bytes.
Erasing and removing code protection.
Burning program memory, 16 locations,
burning data memory, 0 locations,
burning id words and fuses, 1 location,
done.

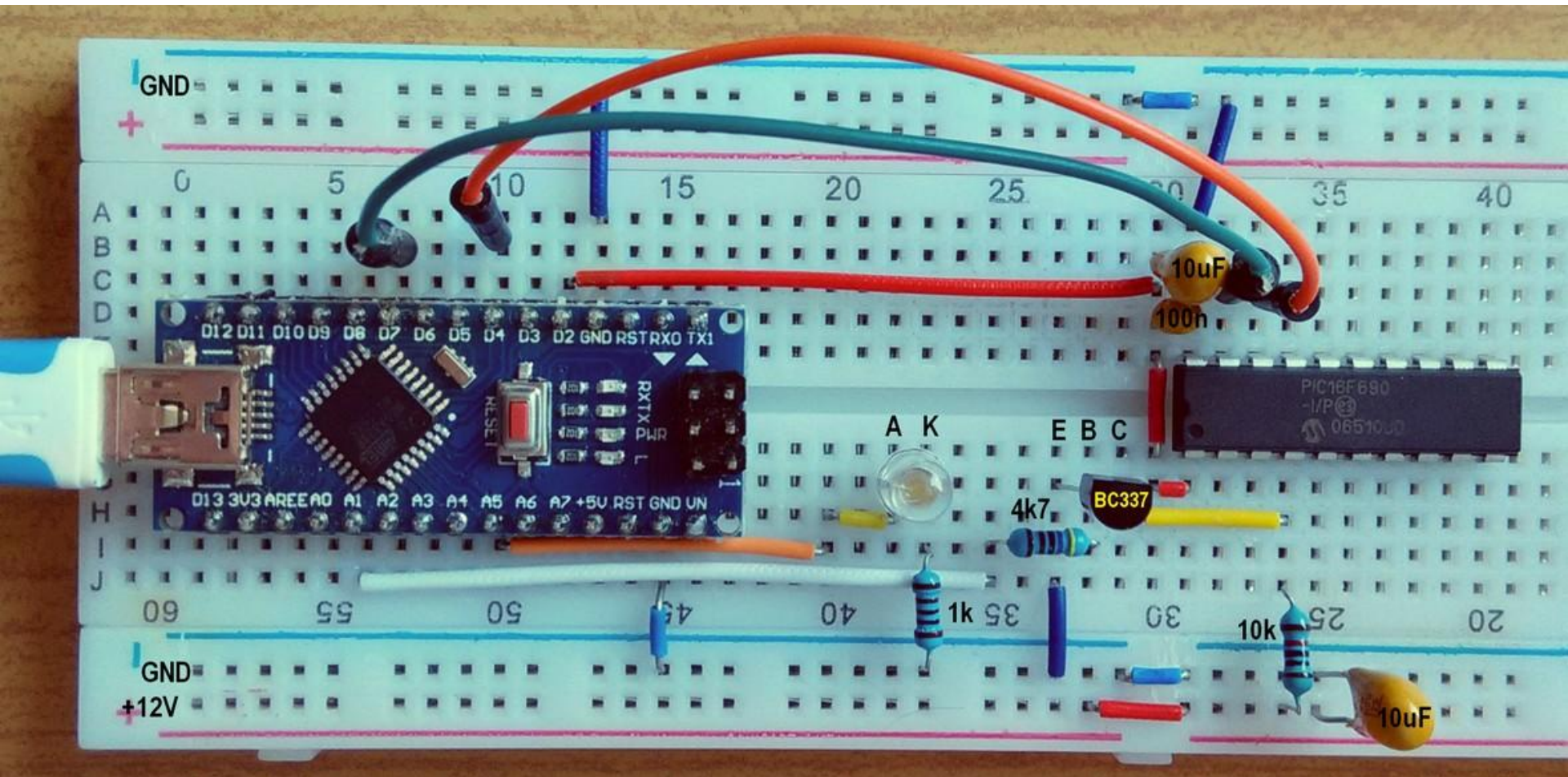
c:\Picprog>
```

PIC16F690

Sok vonatkozásban a **PIC12F683** „nagytesójának” is tekinthetjük.

RAM: 256 bájt, SFR: 4 bank, FLASH: 4 k szó, EEPROM: 256 bájt, 18 GPIO kivezetés

Bővebb periféria-készlet: EUSART, SPI/I2C, 2 analóg komparátor, továbbfejlesztett PWM



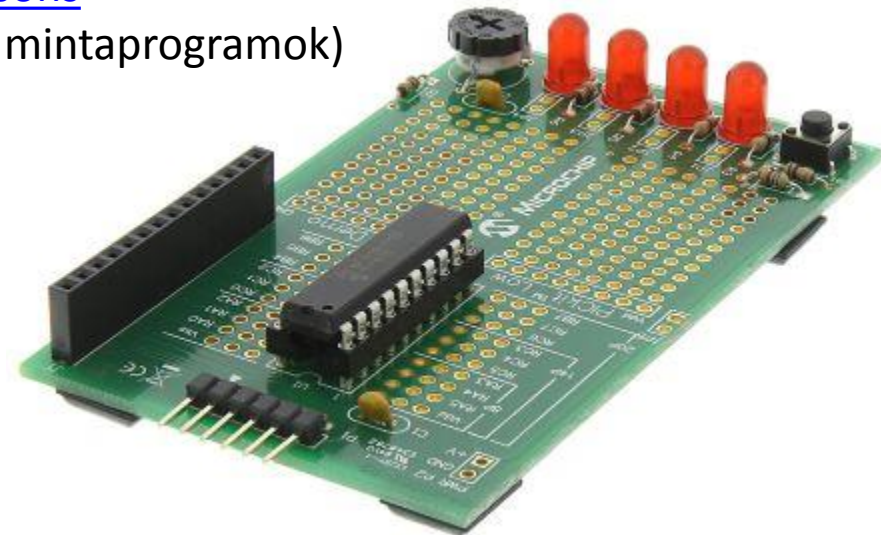
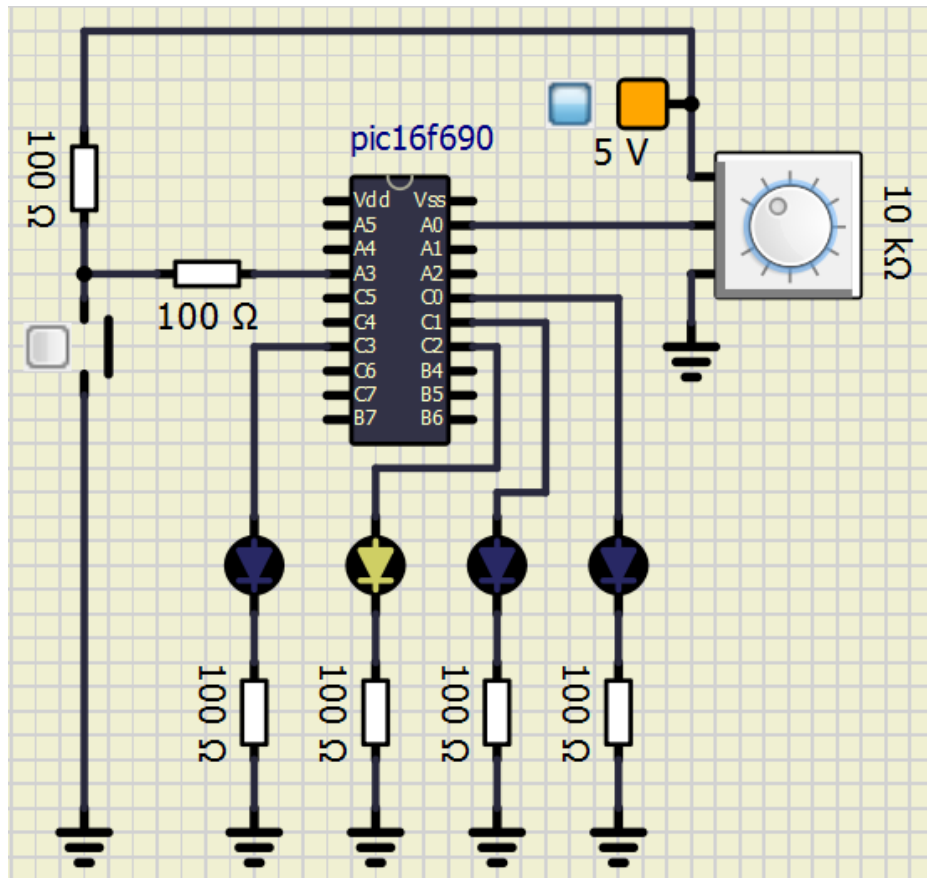
Low Pin Count Demo Board

Egyszerű fejlesztői kártya **PIC16F690** mikrovezérlővel.

Leírás: [Low Pin Count Demo Board User's Guide](#)

Mintaprogramok: [Low Pin Count Demo Board Lessons](#)

(a PICkit2 Starter Kit CD mellékletéről való „gyári” mintaprogramok)



Az `LPC_Demo_board_PIC16F690.simu` állomány betöltése után a **SimulIDE** program segítségével szimulálhatjuk az LPC Demo kártyát és futtathatjuk rajta a fenti linken található mintaprogramokat.