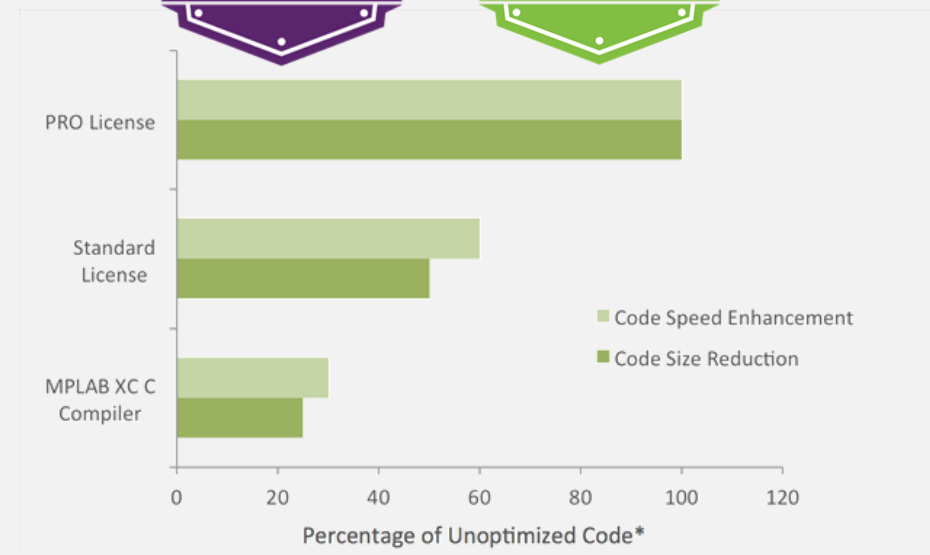




Ismerkedés az eszközökkel

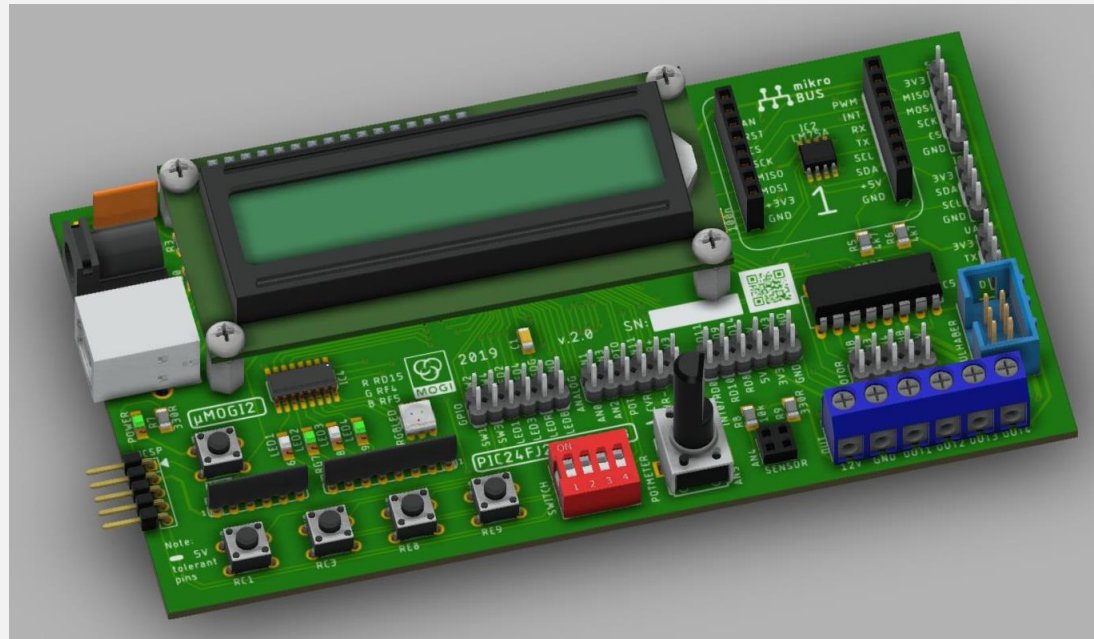
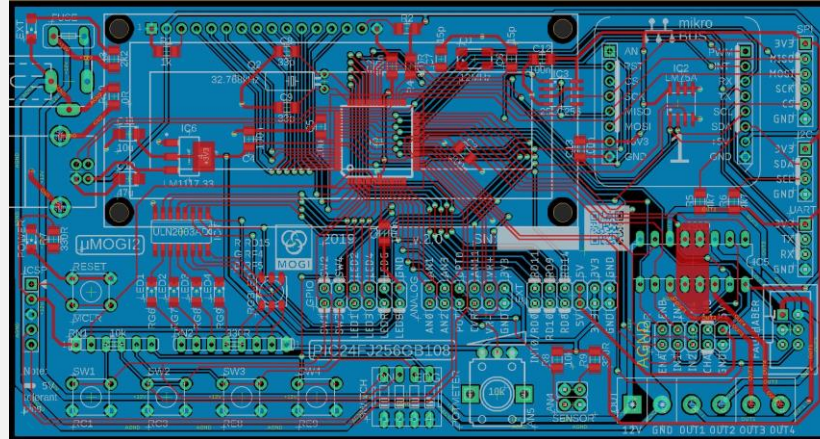
Fejlesztőkörnyezet

- Microchip MPLAB X
 - *Java alapú fejlesztőkörnyezet (NetBeans)*
 - *Platformok: Windows, Linux, Mac*
 - *Két részből áll:*
 - *Intergrated Programming Environment (IPE)*
 - *Integrated Development Environment (IDE)*
 - *Létezik felhőalapú verziója:*
 - *MPLAB® Xpress IDE*
 - www.microchip.com/mplabx
- Fordító (XC16)
 - *16 bites eszközök fordítója*
 - *Az ingyenes verzió 60 napig PRO*
 - www.microchip.com/compilers



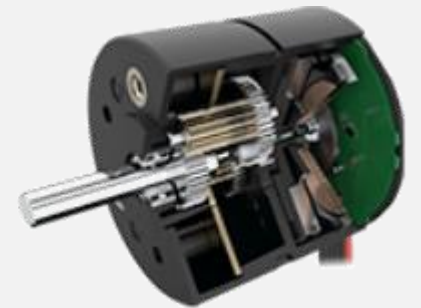
μMOGI² Panel

- I/O
 - 4 nyomógomb / kapcsoló
 - 4 LED
 - 1 RGB LED
 - LCD
- Timer
- PWM
- I²C (LM75A hőmérő)
- SPI (25LC256 EEPROM)
- Komparátor
- ADC
 - potméter
 - reflexiós optokapu
- UART, USB
- Motor H-híd + Encoder



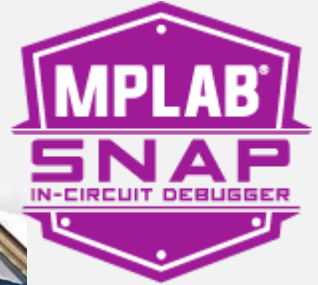
 **FAULHABER**

Bolygóműves DC motor
Enkóderrel

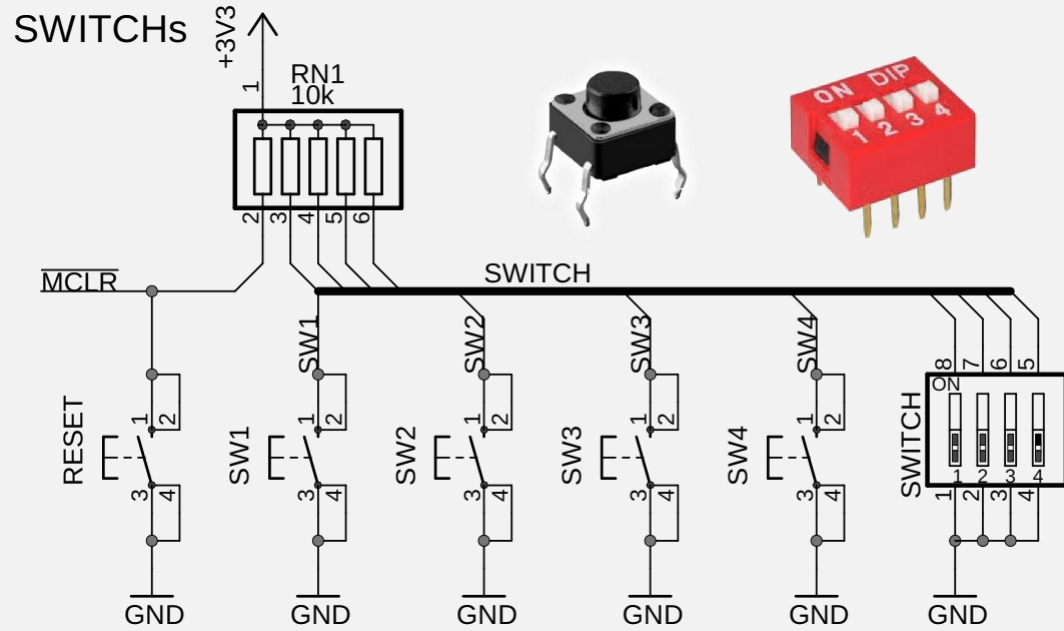


μMOGI² Panel

- SNAP programozó/hibavadász készülék
- MikroElektronika click boards támogatás (1 db mikroBUS foglalat)



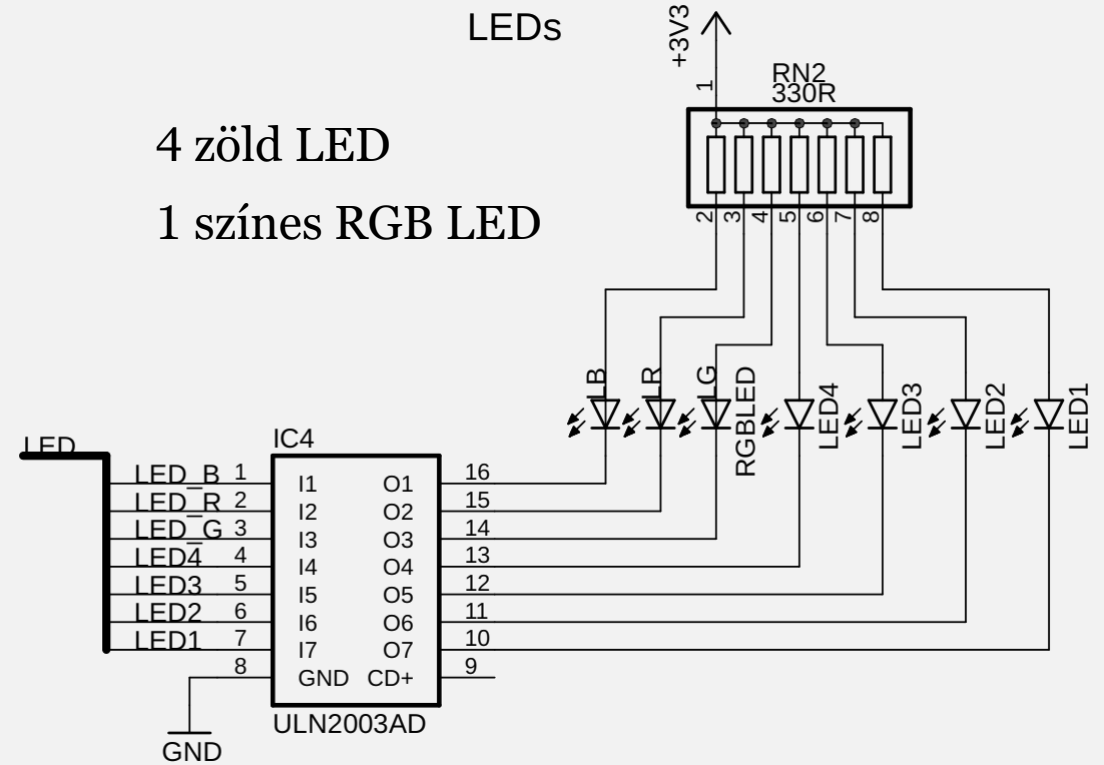
μMOGI² Panel



4 nyomógomb / kapcsoló



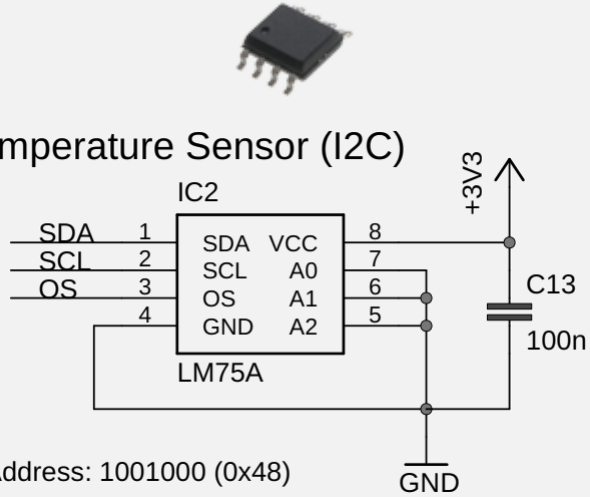
4 zöld LED
1 színes RGB LED



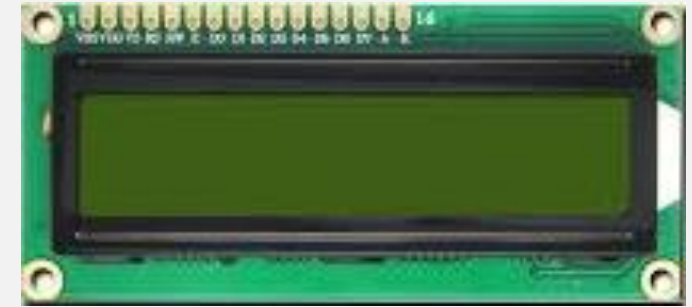
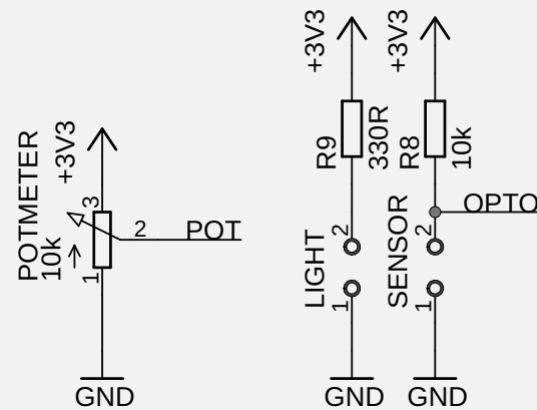


μMOGI² Panel

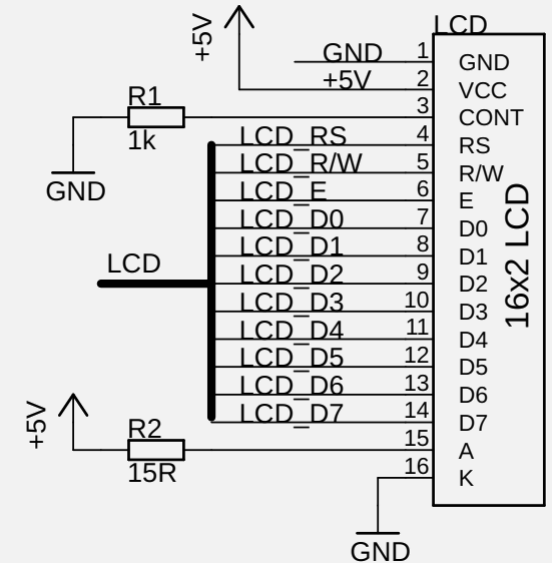
Temperature Sensor (I2C)



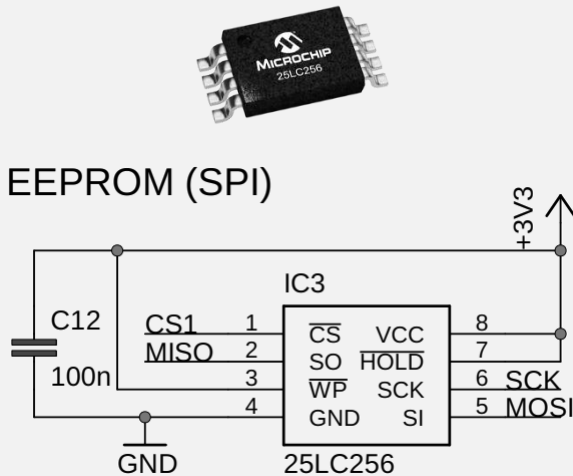
Analog Sensors



2x16 LCD



EEPROM (SPI)



Mikrokontroller

PIC24FJ256GB108

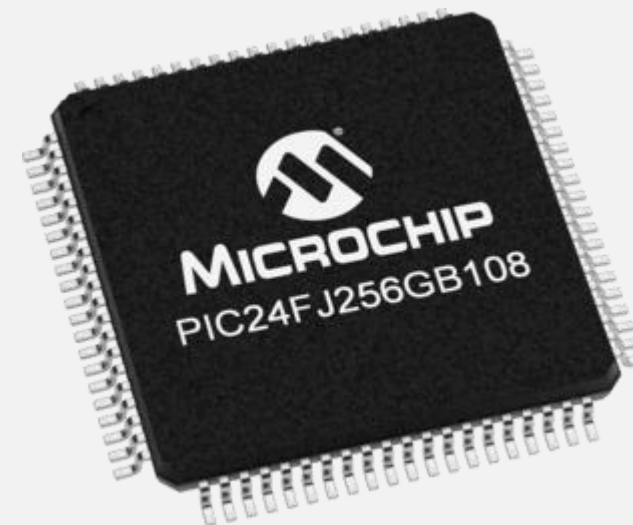
24 bites utasításhossz
16 bites adathossz

16 MIPS

256 kByte program memória

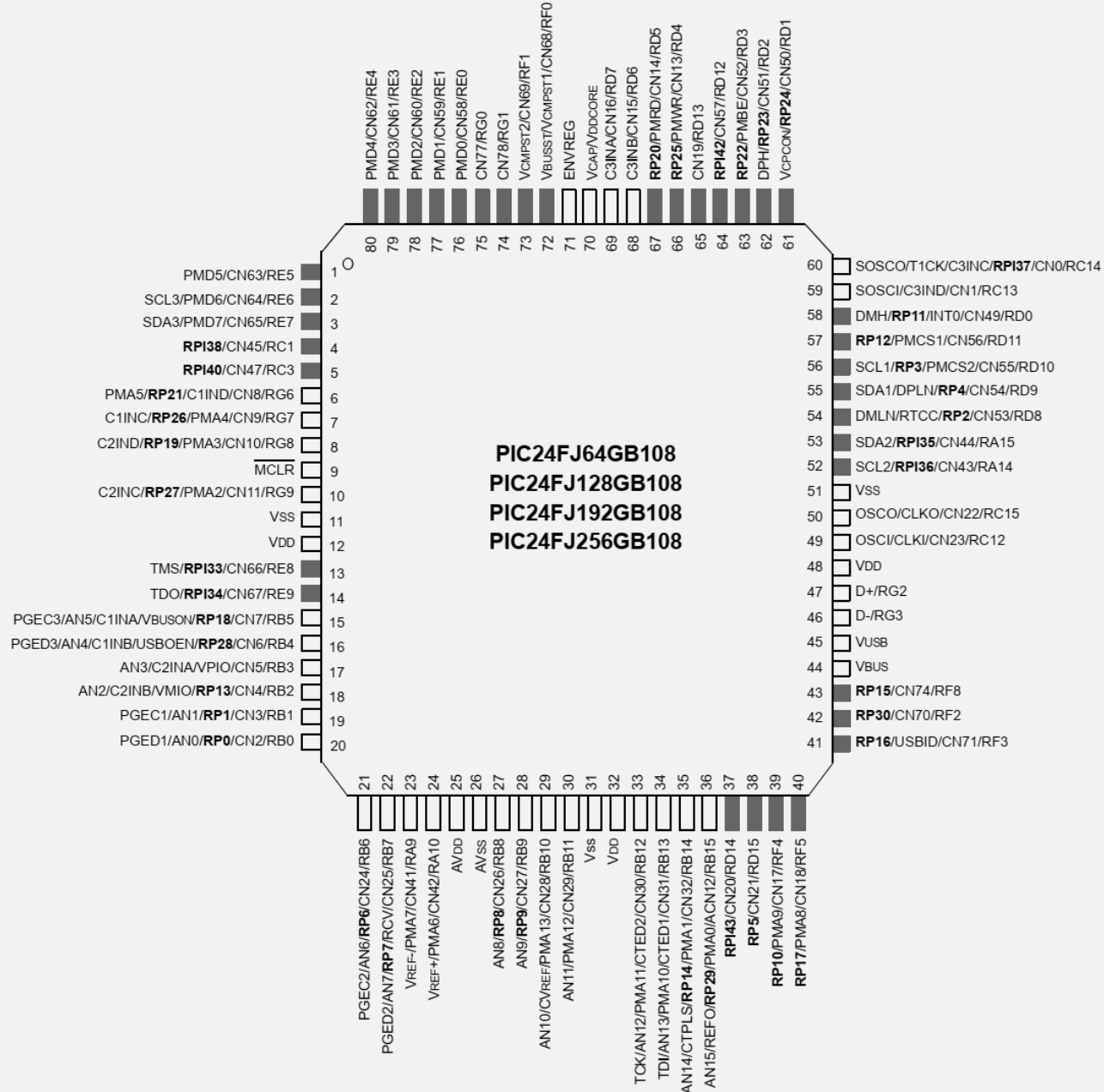
80 láb

A μ MOGI² panelen TQFP tokozású kapott helyet.





Lábak



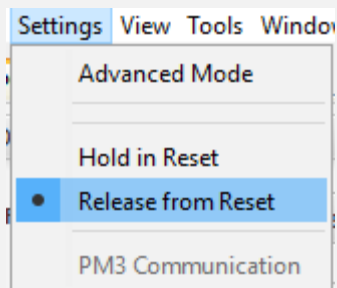


Tesztelés

MPAB X IPE



- Elkészült hex fájlok fel-/letöltésére alkalmas
- Tesztelhetjük vagy törölhetjük a csatlakoztatott eszköz tartalmát
- Production és Advanced módok
- Release from Reset



Mintafájl a Teams-ben

[Youtube link](#)

1 Eszköz: PIC24FJ256GB108

2 Snap S.No : BUR183075070

3 Disconnect

4 C:\uMOGI2.hex

5 Program

Feltöltés

Égető: Snap

Csatlakozás

Csatlakozáskor

HEX fájl betöltése

Égetéskor

```

Currently loaded versions:
Application version.....00.04.59
Boot version.....01.00.00
Script version.....00.04.87
Script build number.....f3faflee5c
Tool pack version .....1.9.685
Target device PIC24FJ256GB108 found.
Device Id Revision = 0x4
Loaded Hex File:C:\uMOGI2.hex
Loading code from C:\uMOGI2.hex...

2022-09-20 12:02:21 +0200 - Programming...

*****

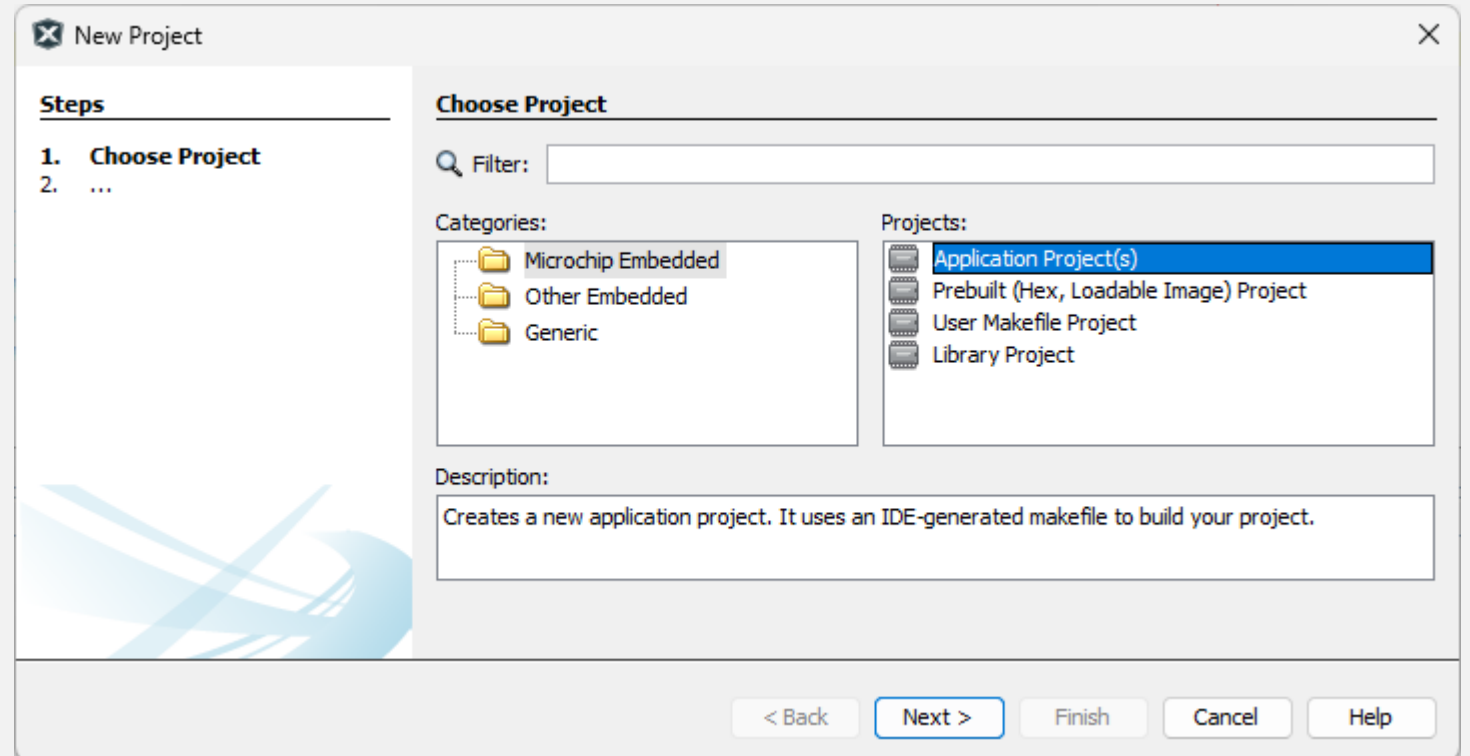
Calculating memory ranges for operation...
Erasing...

The following memory area(s) will be programmed:
program memory: start address = 0x0, end address = 0x5bff
configuration memory
Programming/Verify complete
2022-09-20 12:02:23 +0200 - Programming complete
  
```

C program készítése

- MPLAB X IDE v6.20
- File -> New Project ...
 - Microchip Embedded
 - Application Project(s)

Ctrl+Shift+N



C program készítése

New Project

Steps

1. Choose Project
- 2. Select Device**
3. Select Header
4. Select Plugin Board
5. Select Compiler
6. Select Project Name and Folder
7. (Optional) Add Project

Select Device

Family: All Families

Device: PIC24FJ256GB108

Tool: Snap-SN: BUR.183075070 Show All

MPLAB X IDE

- PIC24-es család
- PIC24FJ256GB108-as eszköz
- Programozó: SNAP

< Back Add Another Project Next > Finish

New Project

Steps

1. Choose Project
2. Select Device
3. Select Header
4. Select Plugin Board
- 5. Select Compiler**
6. Select Project Name and Folder
7. (Optional) Add Project

Select Compiler

Compiler Toolchains

- XC16 [\[Download Latest\]](#)
- XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
- ASM30
- C24
- C30
- HI-TECH DSPICC
- XC-DSC [\[Download Latest\]](#)

MPLAB X IDE

- Fordító:
 - XC16 (v2.10)

< Back Add Another Project Next > Finish

C program készítése

- Projekt név és elérési út megadása
- Karakterkódolás: windows-1250

Steps

1. Choose Project
2. Select Device
3. Select Header
4. Select Plugin Board
5. Select Compiler
- 6. Select Project Name and Folder**
7. (Optional) Add Project

Select Project Name and Folder

Project Name:

Project Location:

Project Folder:

Overwrite existing project.

Also delete sources.

Set as main project

Open MCC on Finish

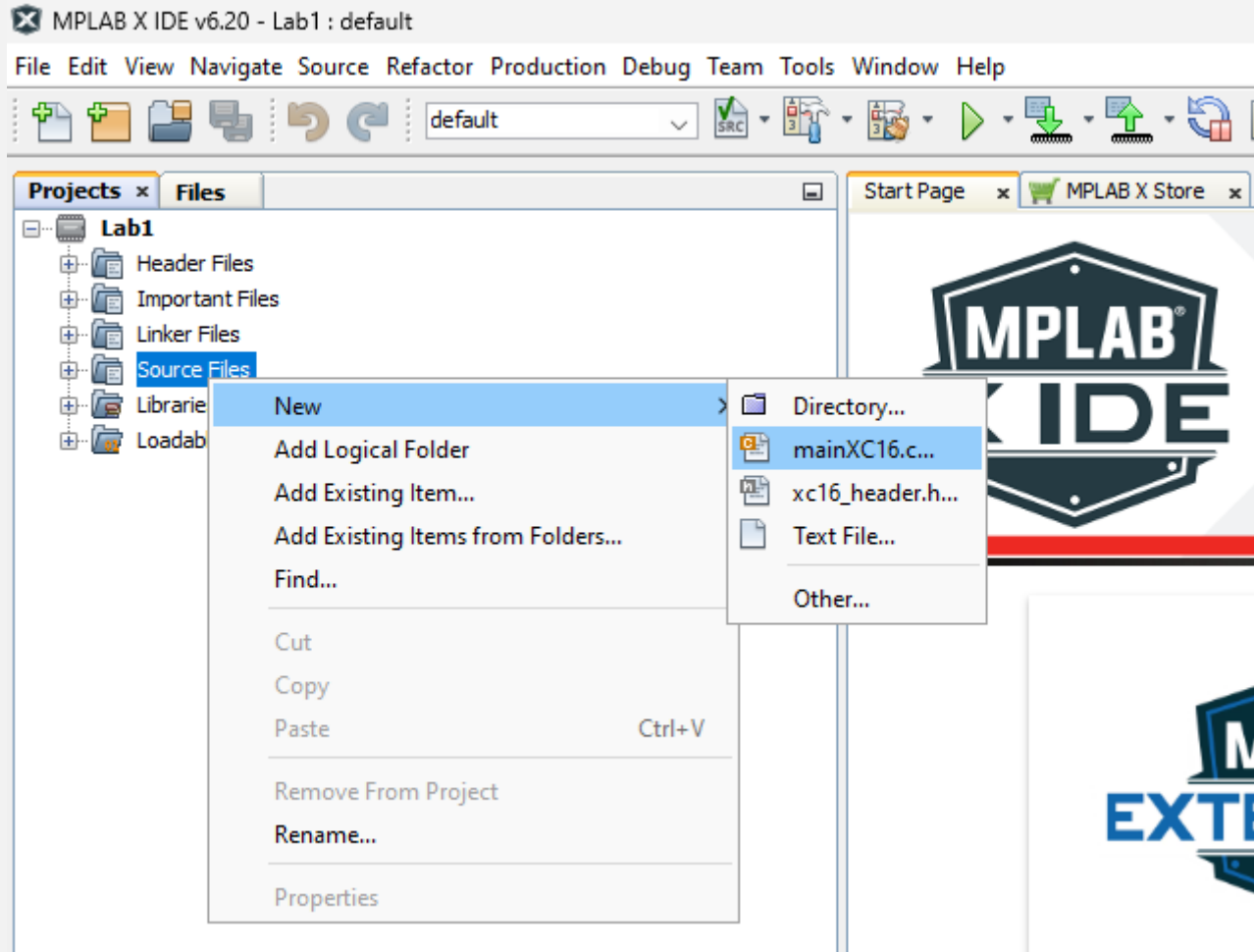
Use project location as the project folder

Encoding:

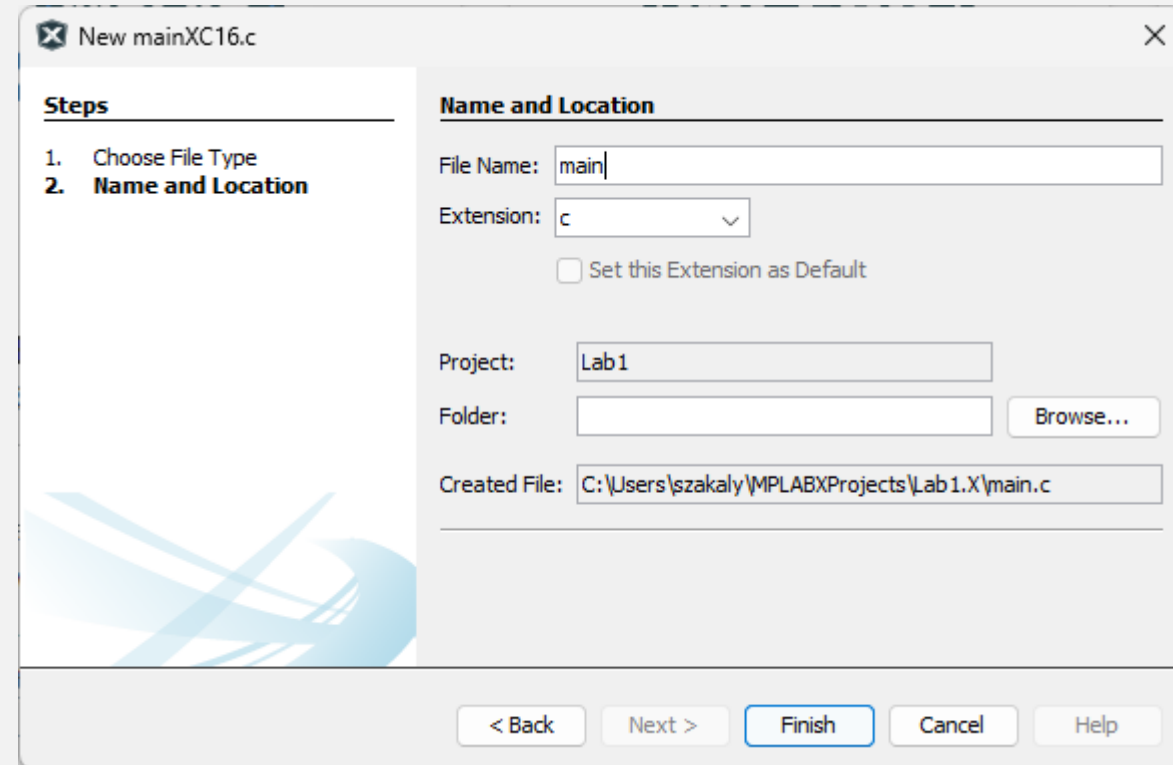
< Back Add Another Project Next > **Finish** Cancel Help

Ne indítsuk el a Code Configurátort

C program készítése



Készítsük el a main.c állományt





C program felépítése

```
#include "xc.h"          /* általános header a PIC24 családhoz */
//egyéb include fájlok

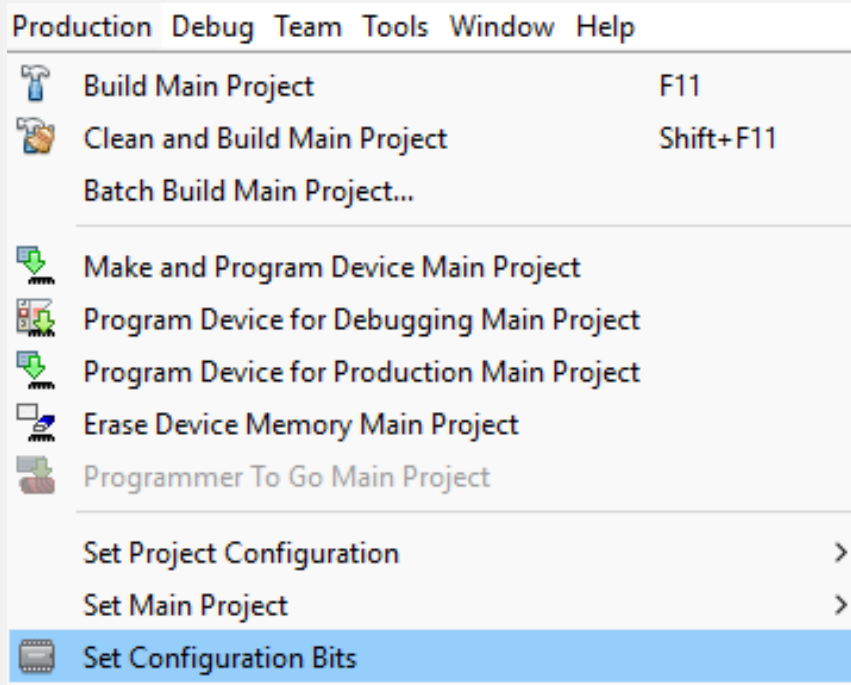
#ifdef __PIC24FJ256GB108__
//konfigurációs bitek beállítása
#endif

//globális változók

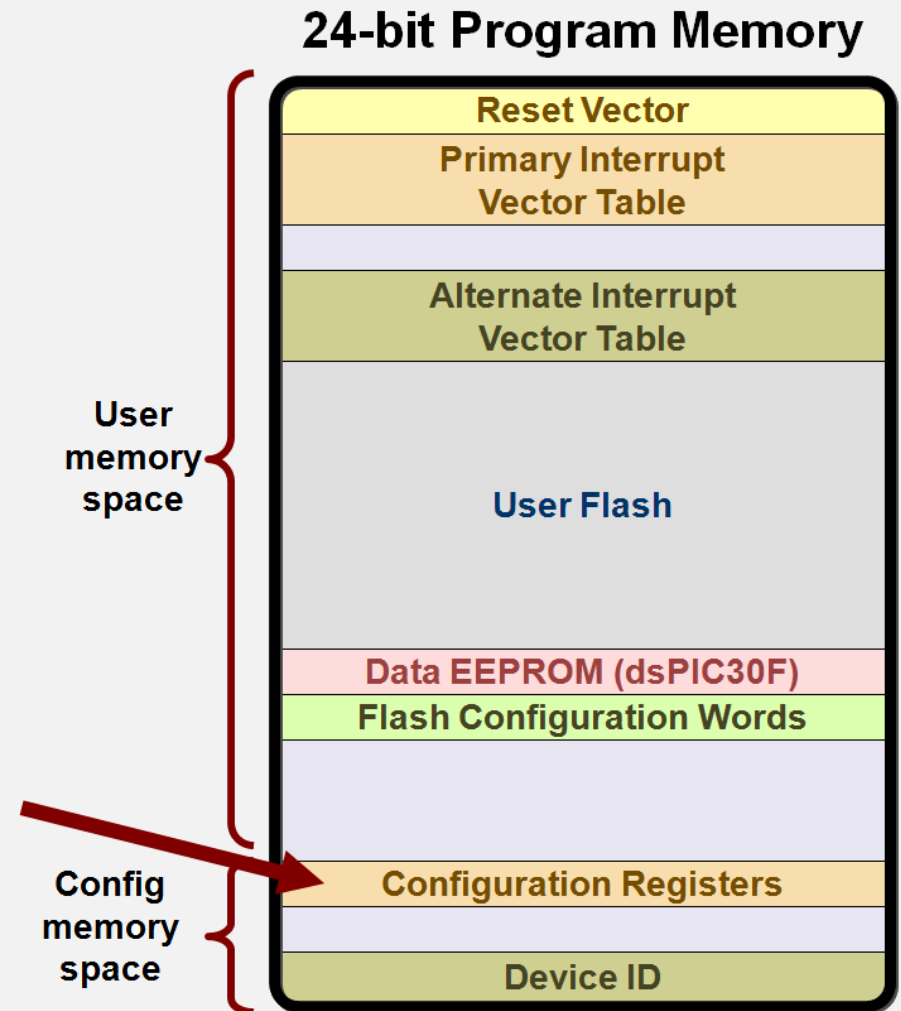
int main(void) {
    //Inicializálás
    //Órajel forrás beállítása
    //Lábak felkonfigurálása

    while (1) //végtelen ciklus
    {
        //ciklikus feladatok elvégzése
    }
    return 0;
}
```

Konfigurációs bitek beállítása



CONFIG1 and CONFIG2 registers located in PM outside the range of executable code space





Konfigurációs bitek beállítása

Konfigurációs bitek beállítása XC16 fordítóval:

#pragma config *CONFIG_BIT_NAME = CONFIG_VALUE*

Address	Name	Value	Field	Option	Category	Setting
2ABFA	CONFIG3	FFFF	-	-	-	-
		1FF	WPFP	WPFP511	Write Protection Flash Page Segment Boundary	Highest Page (same as page 170)
		1	WPDIS	WPDIS	Segment Write Protection Disable bit	Segmented code protection disabled
		1	WPCFG	WPCFGDIS	Configuration Word Code Page Protection Select bit	Last page(at the top of program memory) and Flash configuration words are not protected
		1	WPEND	WPENMEM	Segment Write Protection End Page Select bit	Write Protect from WPFP to the last page of memory
2ABFC	CONFIG2	23F6	-	-	-	-
		2	POSCMOD	HS	Primary Oscillator Select	HS oscillator mode selected
		0	DISUVREG	ON	Internal USB 3.3V Regulator Disable bit	Regulator is enabled
		1	IOL1WAY	ON	IOLOCK One-Way Set Enable bit	Write RP Registers Once
		1	OSCIOFNC	OFF	Primary Oscillator Output Function	OSCO functions as CLKO (FOSC/2)
		3	FCKSM	CSDCMD	Clock Switching and Monitor	Both Clock Switching and Fail-safe Clock Monitor are disabled
		3	FNOSC	PRIPLL	Oscillator Select	Primary oscillator (XT, HS, EC) with PLL module (XTPLL,HSPLL, ECPLL)
		0	PLL_96MHZ	ON	96MHz PLL Disable	Enabled
		2	PLLDIV	DIV3	USB 96 MHz PLL Prescaler Select bits	Oscillator input divided by 3 (12MHz input)
		0	IESO	OFF	Internal External Switch Over Mode	IESO mode (Two-speed start-up)disabled
2ABFE	CONFIG1	3E7F	-	-	-	-
		F	WDTPS	PS32768	Watchdog Timer Postscaler	1:32,768
		1	FWPSA	PR128	WDT Prescaler	Prescaler ratio of 1:128
		1	WINDIS	OFF	Watchdog Timer Window	Standard Watchdog Timer enabled, (Windowed-mode is disabled)
		0	FWDTEN	OFF	Watchdog Timer Enable	Watchdog Timer is disabled
		2	ICS	PGx2	Comm Channel Select	Emulator functions are shared with PGEC2/PGED2
		1	GWRP	OFF	General Code Segment Write Protect	Writes to program memory are allowed
		1	GCP	OFF	General Code Segment Code Protect	Code protection is disabled
		0	JTAGEN	OFF	JTAG Port Enable	JTAG port is disabled

Memory Configuration Bits Format Read/Write **Generate Source Code to Output**

Konfigurációs bitek C kódra fordítása

Filter Address

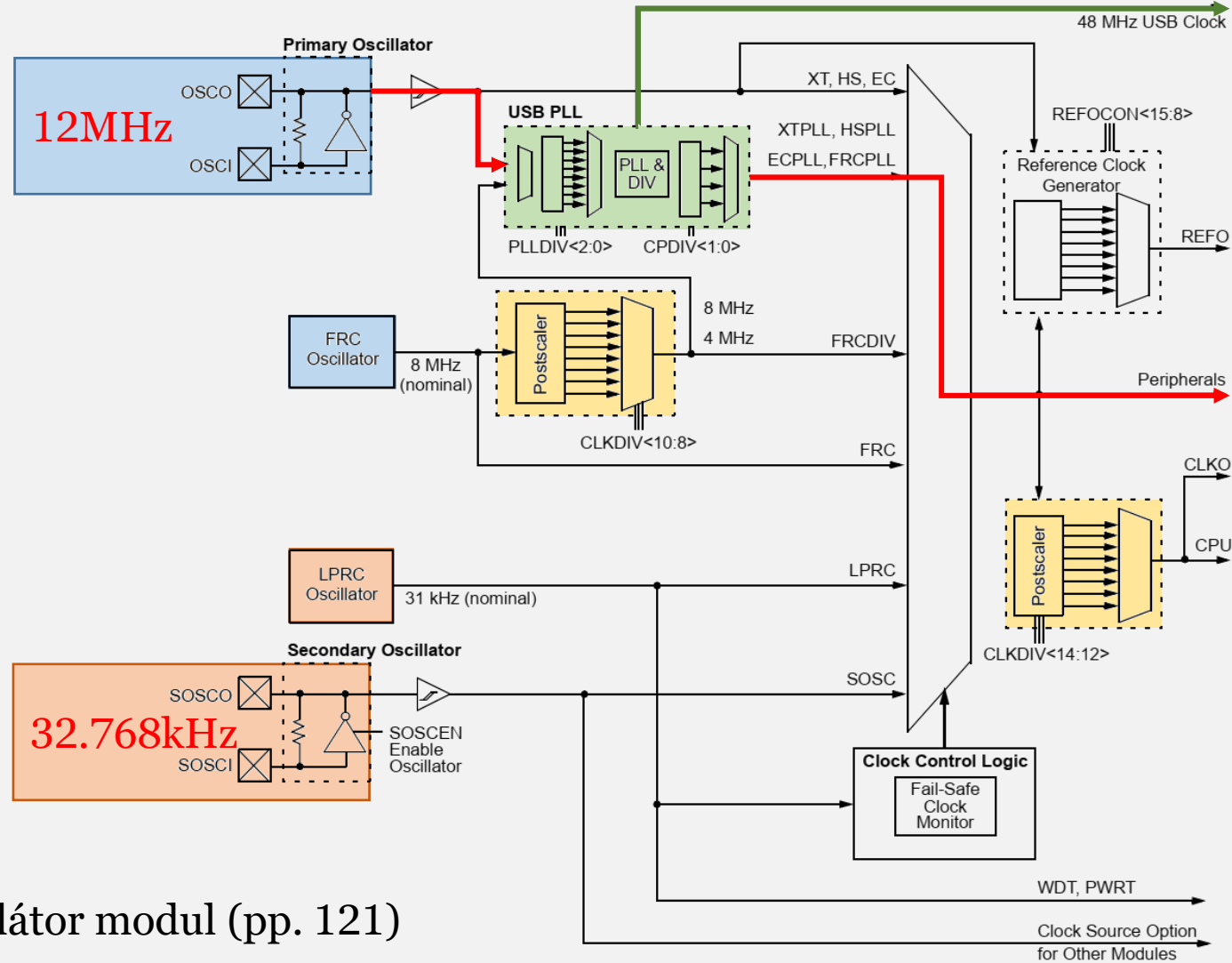


Konfigurációs bitek beállítása - μ MOGI² Panelre

```
// Primary Oscillator Select (HS oscillator mode selected)
#pragma config POSCMOD = HS
// Internal USB 3.3V Regulator Disable bit (Regulator is enabled)
#pragma config DISUVREG = ON
// Oscillator Select (Primary oscillator (XT, HS, EC) with PLL module (XTPLL,HSPLL, ECPLL))
#pragma config FNOSC = PRIPLL
// USB 96 MHz PLL Prescaler Select bits (Oscillator input divided by 3 (12MHz input))
#pragma config PLLDIV = DIV3
// Internal External Switch Over Mode (IESO mode (Two-speed start-up) disabled)
#pragma config IESO = OFF

// Watchdog Timer Enable (Watchdog Timer is disabled)
#pragma config FWDTEN = OFF
// Comm Channel Select (Emulator functions are shared with PGEC2/PGED2)
#pragma config ICS = PGx2
// JTAG Port Enable (JTAG port is disabled)
#pragma config JTAGEN = OFF
```

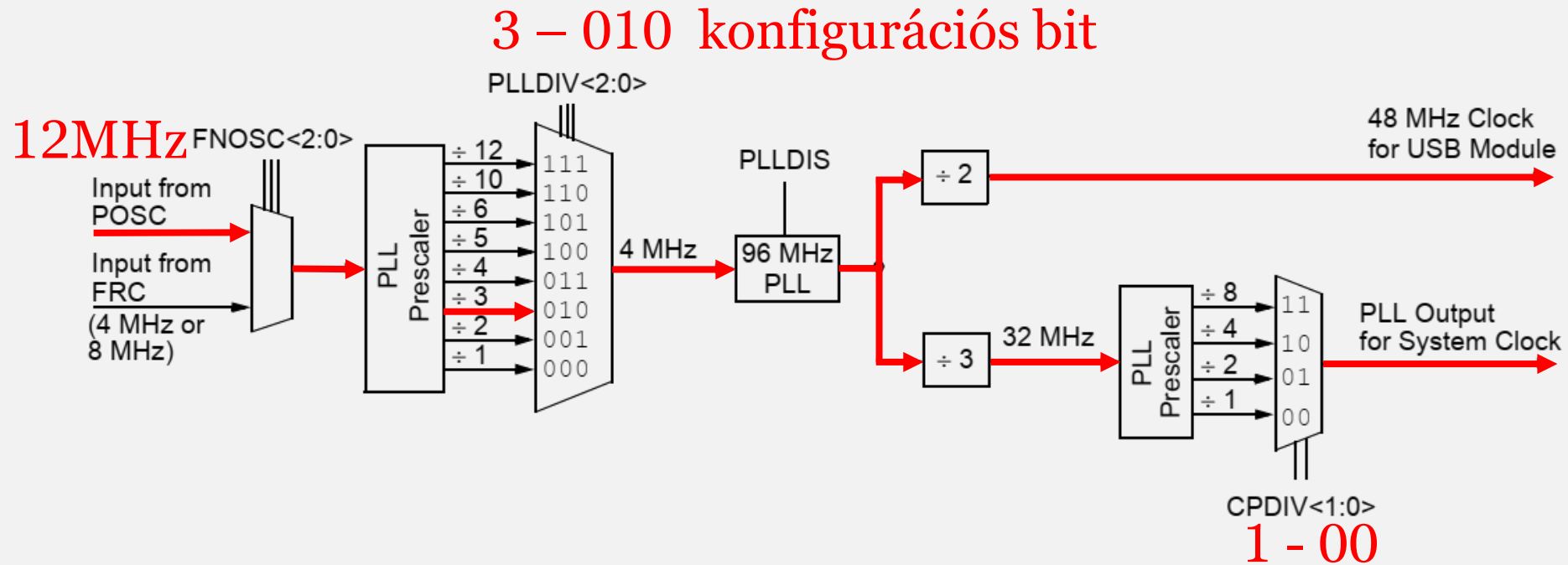
Órajelforrások



Oscillátor modul (pp. 121)

Órajelforrás beállítása

- Az elsődleges kvarc 12MHz-es
- PLL beállítása: (pp. 128)



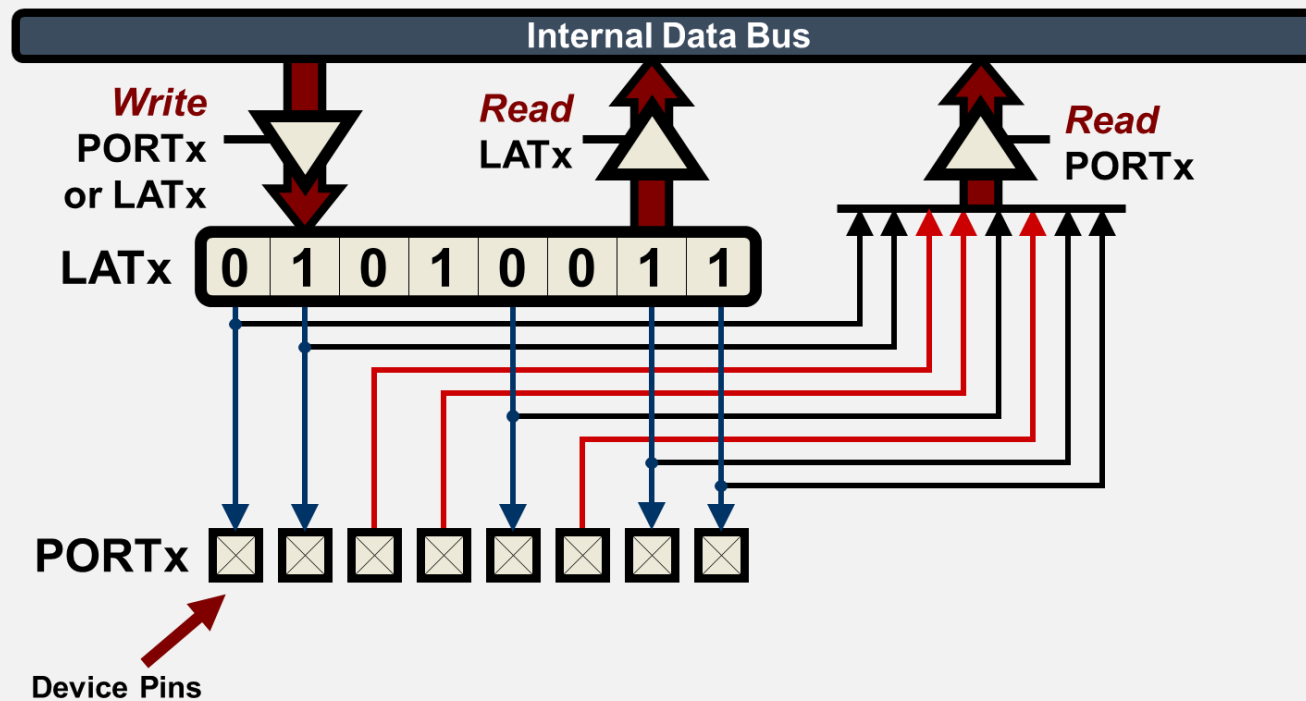


Órajelforrás beállítása

```
/*  
* Külső HS oszcillátor 12MHz  
* USB PLL bekapcsolva  
* pp 128  
*/  
  
// PLL kimenete, System Clock  
CLKDIVbits.CPDIV = 0; //1:1 osztás, 32MHz  
  
// várakozás PLL elkészültéig  
while(!OSCCONbits.LOCK) Nop();  
  
// Watchdog timer ki  
RCONbits.SWDTEN = 0;
```

GPIO használata

- TRIS_x kimenet (0) / bemenet (1)
- PORT_x a láb állapotát lehet lekérdezni / beállítani
- LAT_x az adat retesz állapotát lehet lekérdezni / beállítani
- ODC_x Open-Drain kimenet engedélyező regiszter



GPIO használata

A mikrovezérlő lábai reset után mind bemenetek és amelyik láb rendelkezik analóg funkcióval az analóg bemenetként működik.

Feladatok

1. Keressük meg a piros led lábát, állítsuk be kimenetnek és kapcsoljuk be.
2. Keressük meg az SW1-es nyomógomb lábát, állítsuk be bemenetnek és figyeljük meg a működését. Ha benyomjuk a gombot, akkor világítson a piros led, különben ne.

Hogy tudunk várakozni? Miért nincs delay fvg., mint Arduino-n?

- Timer modul alkalmazásával – következő óra
- Nop() utasításokkal – 1 utasításnyi időt vesz igénybe

Szerencsére nem kell kiszámolnunk hányszor kell meghívni, mert a **libpic30.h** tartalmazza a számítást. Előtte definiálni kell a pontos működési frekvenciát!

```
#define SYS_FREQ          32000000L
#define FCY                SYS_FREQ/2
#include <libpic30.h>
```

Alkalmazható függvények:

```
__delay_ms (d)
__delay_us (d)
```



Gyakorlás

Feladat

1. Keressük meg a kék és zöld ledek lábait is, állítsuk be kimenetnek.
2. A három led felváltva világítson 1 másodperces váltásokkal.
3. A program csak az SW1 benyomásával induljon el.

Feladat

1. Az SW1 benyomásával lépkedjünk az egyes színek között.
2. A 4 db ledre készítsen egy futó fényt