

A PIC programozás C nyelven

A C nyelvről már mindenki hallott, aki beleolvasott a számítógép, vagy mikrokontroller programozásról szóló szakirodalomba. És biztos nem véletlen, hogy a C nyelv a talán a legnépszerűbb magas szintű programozási nyelv, mind a PC-k, mind a mikroprocesszorok világában.

Hogy miért is? Mert pl. alig kell törődnünk a mikrokontroller belső felépítésével, kényelmes utasítás készlet áll rendelkezésünkre, és sokkal könnyebben olvasható - és ezért javítható - a programkód, mint ha assembler nyelven programoznánk. Arról meg nem is beszélve, hogy pl. egy gyökvonás C nyelven egy `SQRT(x)` parancs sor, assembly nyelven egy nap, vagy egy hét...

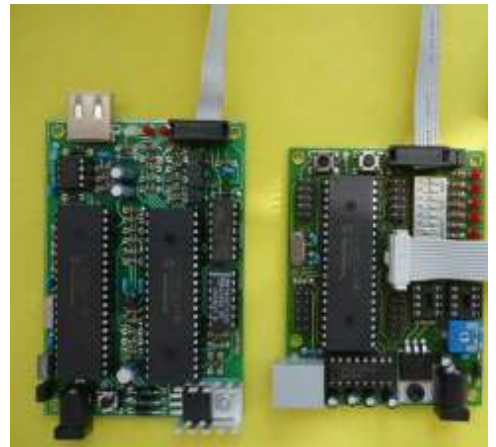
Az előbbieket már persze a gyártók is figyelembe veszik, az újabb mikrokontrollerek belső felépítését már eleve a C nyelvű programozásra optimalizálták.

Addig tehát sokan eljutottunk, hogy C nyelven kéne programozni. De hogyan tovább? Mert azok a kezdőlépések mindig nehezek...

A következőkben egy gyakorlati példán keresztül próbálom leírni, hogyan használhatjuk a MICROCHIP C18 fordítóprogramját, és tölthetjük be az elkészült program kódot, egy PICDEMO áramkörbe.

Egy egyszerű, "LED villogtató" C programot fordítunk le, töltünk be, és futtatunk a PICDEMO mikrokontrolleres vezérlő panelen.

A hardver, amin a program fut, a „mikroklubbos” PICDEMO áramkör lesz, benne egy 18F4580 mikrokontrollerrel, a program beégetésére az ICD2-est használom.



A C fordító a MICROCHIP MPLAB programja „alatt” fut, tehát, ha még nem tettük meg, installáljuk az MPLAB programot a gépünkre.

Konkrétan az MPLAB 8.3-ás verzióját használtam, egy XP-s gépen, aminek a telepítő programja a `\SHAREWARE\MPLAB8.3` könyvtárban található.

Az MCC18 C fordító program:

A MICROCHIP mikrokontrollerekre több cég is készített C fordító programot. Melyiket válasszuk?

Kezdsnek jó lenne egy „ingyen program”. Nézzük a MICROCHIP a „standard/eval” C fordítóit! Ezek 60 napig teljes verzióként működnek – erre utal az eval, evulate, azaz „kiértékelő” jelző – ami alatt minden funkció teljes körűen használható. A 60 nap után – ha nem vesszük meg a licenszet – bizonyos kód optimalizáló funkciók „kikapcsolódnak”, de továbbra is használhatjuk a programot. (Egy kicsit lassabb, és nagyobb lesz a fordított kód, de ez általában semmi problémát nem okoz.)

A MICROCHIP – vajon miért? – minden típus családhoz egy-egy külön fordítót alkotott, azaz van C fordító a 16F, 18F, 24F, és a 32F-es mikrokontrollerekhez.

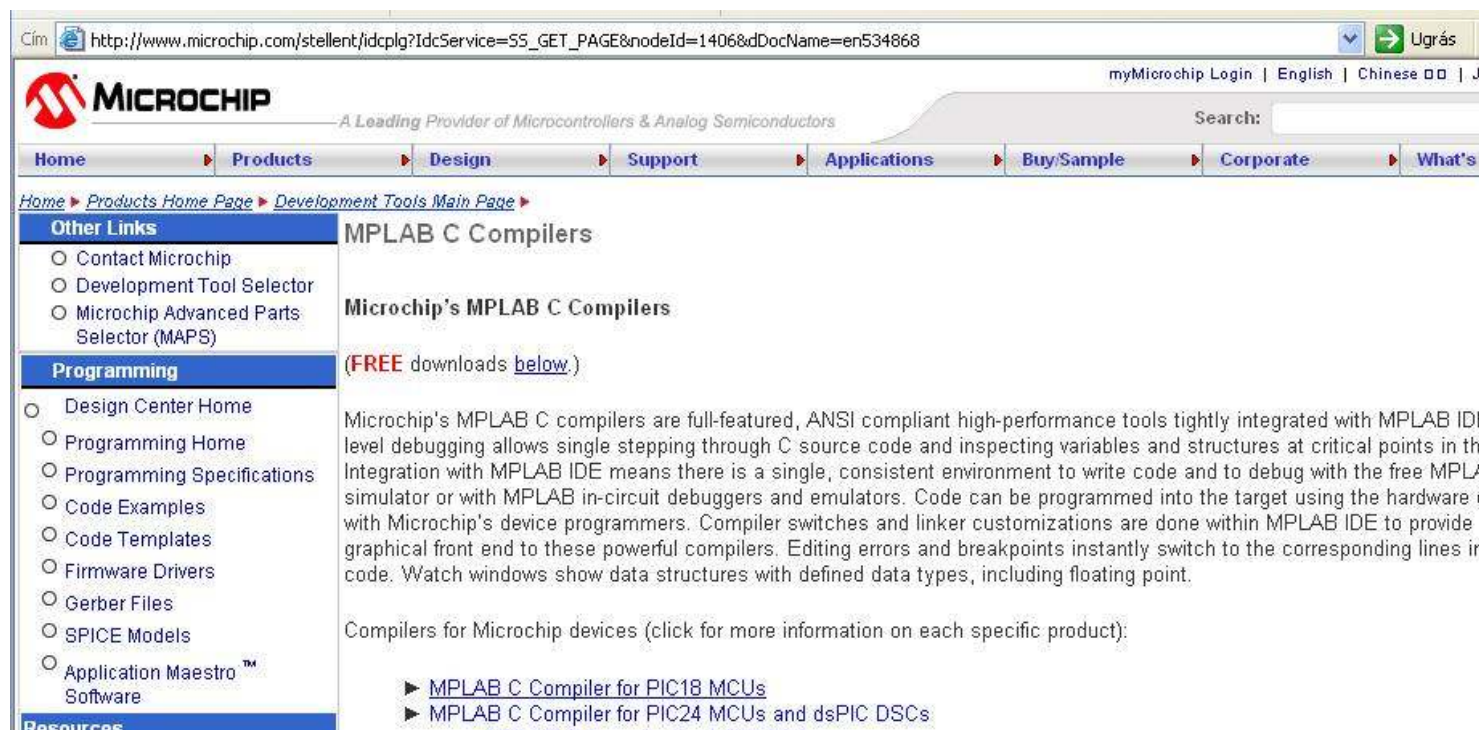
Én a 18F-es családot választottam. A 18F-es mikrokontrollerek gyorsak, a belső felépítésüket eleve a C programozásra igazították, kaphatóak 5 és 3.3 voltos tápfeszültséghez, olcsók, és – ami nagyon fontos szerintem – a „hagyományos” DIL, és smd tokozásban is elérhetőek.

Tehát nekünk az „MPLAB C Compiler for PIC18” kell.

Először is töltsük le a programot. Na ez nem is olyan egyszerű...

Kezdsnek meg kell találnunk a programot. A MICROCHIP oldalán – kicsit elrejtve – keressük meg az MPLAB C Compilers lapot:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en534868



MICROCHIP
— A Leading Provider of Microcontrollers & Analog Semiconductors

myMicrochip Login | English | Chinese | J

Home | Products | Design | Support | Applications | Buy/Sample | Corporate | What's

Home ▶ Products Home Page ▶ Development Tools Main Page ▶

Other Links

- Contact Microchip
- Development Tool Selector
- Microchip Advanced Parts Selector (MAPS)

Programming

- Design Center Home
- Programming Home
- Programming Specifications
- Code Examples
- Code Templates
- Firmware Drivers
- Gerber Files
- SPICE Models
- Application Maestro™ Software

Resources

MPLAB C Compilers

Microchip's MPLAB C Compilers

(FREE downloads [below](#).)

Microchip's MPLAB C compilers are full-featured, ANSI compliant high-performance tools tightly integrated with MPLAB IDE. Level debugging allows single stepping through C source code and inspecting variables and structures at critical points in the integration with MPLAB IDE means there is a single, consistent environment to write code and to debug with the free MPLAB simulator or with MPLAB in-circuit debuggers and emulators. Code can be programmed into the target using the hardware or with Microchip's device programmers. Compiler switches and linker customizations are done within MPLAB IDE to provide a graphical front end to these powerful compilers. Editing errors and breakpoints instantly switch to the corresponding lines in code. Watch windows show data structures with defined data types, including floating point.

Compilers for Microchip devices (click for more information on each specific product):


- ▶ [MPLAB C Compiler for PIC18 MCUs](#)
- ▶ [MPLAB C Compiler for PIC24 MCUs and dsPIC DSCs](#)

Menjünk innen tovább az „MPLAB C Compiler for PIC18 MCUs” lapra!

 Sign in required to download content.

Title	Date Published	Size	D/L
HI-TECH C PRO for PIC10/12/16 MCU Family in Lite mode v9.65PL1 	7/9/2009 4:28:45 PM	5929 KB	
HI-TECH C PRO for the PIC18 MCU Family in Lite mode v9.63PL2 	3/25/2009 2:44:13 PM	24261 KB	
HI-TECH C PRO for the PIC32 MCU Family in Lite mode v9.60PL2 	3/25/2009 2:55:40 PM	21367 KB	
MPLAB C Compiler for dsPIC v3.20 Standard-Eval Version 	8/31/2009 2:39:10 PM	33318 KB	
MPLAB C Compiler for PIC24 v3.20 Standard-Eval Version 	8/31/2009 2:26:19 PM	28967 KB	
MPLAB C Compiler for PIC32 v1.05 Evaluation Version 	3/9/2009 1:37:26 PM	26827 KB	
MPLAB C for PIC18 v3.33 Standard-Eval Version 	8/28/2009 1:00:25 PM	53136 KB	

Na, itt válasszuk ki a „Standard-Eval Version”-t!

[MPLAB C for PIC18 v3.33 Standard-Eval Version](#)  8/28/2009 1:00:25 PM 53136 KB 

Kezdődhet(ne) a letöltés. De ehhez a „Sign in”-re kattintva be kell jelentkezünk.

Sign In

◀ Denotes required field

E-mail address:

Password:

 [Sign In](#)

 [Forgot your password?](#)

Az ehhez szükséges jelszóhoz regisztrálnunk kell a Microchip-nél:

Sign in required to download the content. If you have registered previously with an existing email address and password, please click on the "Sign In" button. If you have not registered before, you may [register](#) now. If you wish not to register, you may click on the "Cancel" button.

Ha ezzel is meg vagyunk, kezdődhet az áhított letöltés:



Szóval nem olyan egyszerű a letöltés, ezért aztán a mellékelt CD-n is megtalálható, a PIC-C könyvtárban a program.

Remek, kezdődhet a program telepítése!

A telepítés során több ablak ad lehetőséget a telepített összetevők, a környezet, stb. változtatására, de én semmit se állítottam át - azaz egyetlen x-et se vettem ki, vagy raktam be - tehát meghagytam mindent az alapbeállításoknál, mindig a „Next”-re kattintva.

Így a program az C:\MCC18 könyvtárba települt.

Legyen a cél, egy egyszerű, C program lefordítása. Mondjuk a PICOKC04.C mintaprogramé, ami egyébként egy futófényt csinál a PICDEMO panel LED-jein.

Röviden a PICOKC04.C programról:

Ez egy 10 darabos, a C programozás alapjait bemutató, valamint a kezdő lépések megkönnyítésére készült minta program csomag 4. tagja. A legelső, pár soros program csak „begyűjt” egy LED-et, a következő már meg is villogtatja, ez a negyedik már egy futófényt csinál a 8 LED-en. (A 9. már az A/D konvertert kezeli, és az LCD-re is ír, stb.)

Magáról a C nyelvről, és a PIC-ek C programozásáról nem akarok írni. A C-ről több remek könyv is megjelent, a MICROCHIP PIC-ek C programozásáról pedig pl. Kónya László „PIC mikrokontrollerek alkalmazástechnikája” c. könyvben olvashatunk.

Szóval csak nagy vonalakban a program működéséről:

Először is megadjuk a C fordítónak, hogy a 18F4580 mikrokontrollert akarjuk használni, majd beállítjuk a „fuse”-okat. (Jó sok van neki.)

Aztán egy „meghívható időzítő” ciklus, NOP utasításokból csinálunk annyit, hogy kb. 0.5 mp-ig teljen az idő.

Ezután a konfiguráljuk, hogy a mikrokontroller RB0-7 portját - ott van a 8 LED - kimenetnek akarjuk használni.

Kezdetnek nullázunk egy számlálót - ez fogja 1-8 között mutatni, hol tartunk LED-ek léptetésében - és betöltjük a 00000001 értéket a PIC RB port kimenet vezérlő regiszterébe.

Ezután jön a program lényege, a futófényt csináló ciklus. A port vezérlő regiszter tartalmát eggyel léptetjük - 00000010 lesz - és eggyel növeljük a „futófény” számláló értékét is.

Ez utóbbit aztán meg kell vizsgálni, hogy nem lett-e több mint 7, mert ha igen, akkor a számlálót nullázni, az RB port vezérlő regisztert pedig alap állapotba kell hozni. (Azaz 00000001-el feltölteni.)

Ezután meghívjuk a fél másodperces időzítést, majd kezdődik minden előlről, és így tovább...

Persze a program működésének megértéséhez kicsit bele kell ásunk magunk a mikrokontroller, és magának a C nyelv utasításainak leírásába.

```

//*****
//          PICOK C 04
//*****
//    4. program : futófény. Egy kezdőérték töltése a PORTB-t
//    vezérlő regiszterbe, aztán kb. 0.5 mp időzítés, ezután a
//    tartalom eggyel siftelve, időzítés, sift ..
//*****
//Processor:   PIC18F4580
//Quartz:      4MHz
//Panel:       PICDEMO
//*****

#include <p18F4580.h>

// Állítsuk be a "fuse-okat" :
#pragma config OSC=XT,FCMEN=OFF,IESO=OFF
#pragma config PWRT=ON,BOREN=OFF //,BORV=42
#pragma config WDT=OFF
#pragma config MCLRE=ON,PBAREN=OFF
#pragma config STVREN=OFF,LVP=OFF,DEBUG=OFF, XINST=OFF
#pragma config CP0=OFF,CP1=OFF,CP2=OFF,CP3=OFF
#pragma config CPB=ON,CPD=OFF
#pragma config WRT0=OFF,WRT1=OFF,WRT2=OFF,WRT3=OFF
#pragma config WRTC=OFF,WRTB=ON,WRTD=OFF
#pragma config EBTR0=OFF,EBTR1=OFF,EBTR2=OFF,EBTR3=OFF
#pragma config EBTRB=ON
//*****
void delay(void)
{
    unsigned int i;    //változó deklarálása

    for(i=0;i<20000;i++) //Ez egy kb 0.5 masodperces időzítésű ciklus
    {
        Nop();
        Nop();
        Nop();
    }
}
//*****
void main(void)
{
    char cnt;    // a cnt változó deklarálása.

    TRISB = 0;    // A portb kimenet legyen!
    cnt=0;    // számláló nullázás
    PORTB = 0x01;    // PORTB kezdőérték: 00000001

    while(1)    // ez egy végtelen ciklus
    {
        PORTB=PORTB<<1;    // a portb-t rotáljuk. (Hogy fusson az a LED...)
        cnt++;    // 1-el növeljük a számlálót
        if(cnt>7)    // ha elérjük a 8-at, kezdjük előlről,
        {
            cnt=0;    // a számlálót nullázzuk,
            PORTB = 0x01;    // a PORTB-be meg újra a 00000001 kezdőérték
        }
        delay();    // késleltetés
    }
}

```


A minta programok a PIC-C\C-PROGIK\ könyvtárban vannak

Csináltam egy „C-PROGIK” alkönyvtárat az MCC18-ban, és bele másoltam a PICOKC04.C minta programot:

c:\Mcc18*.*					c:\Mcc18\c-progik*.*				
Név	Kit.	Méret	↓Dátum	Attr.	Név	Kit.	Méret	↓Dátum	Attr.
↑...[...]			2009.09.01 14:05 ----		↑...[...]			2009.09.01 14:05 ----	
[bin]			2009.09.01 12:27 ----		[picokc04]	c	1 818	2009.08.15 18:13 -a--	
[cpp]			2009.09.01 12:27 ----						
[c-progik]			2009.09.01 14:05 ----						
[doc]			2009.09.01 12:27 ----						
[...]			2009.09.01 12:27 ----						

Na, ha eddig eljutottunk, akkor nyissuk meg az MPLAB programot, és csináljunk egy „project”-et!

Használjuk ehhez a projekt vizardot, azaz a projekt varázslót!



Válasszuk ki a PIC18F4580 mikrokontrollert:

Device:

PIC18F4580

Második lépés,
válasszuk ki, hogy a
C18 fordítóprogramot
akarjuk használni:

Project Wizard

Step Two:
Select a language toolsuite

Active Toolsuite: Microchip C18 Toolsuite

Toolsuite Contents

- MPASM Assembler (mpasmwin.exe)
- MPLINK Object Linker (mplink.exe)
- MPLAB C18 C Compiler (mcc18.exe)
- WINLIB Library (winlib.exe)

Location

Browse...

☒ Store tool locations in project

Help! My Suite Isn't Listed!

☐ Show all installed toolsuites

< Vissza Tovább > Mégse Súgó

Majd adjunk egy
nevet a „project”-nek:

Én a „cproba”
nevet adtam.

Project Wizard

Step Three:
Create a new project, or reconfigure the active project?

☒ Create New Project File

C:\MCC18\c-progik\cproba.mcp

Browse...

☐ Reconfigure Active Project

☐ Make changes without saving

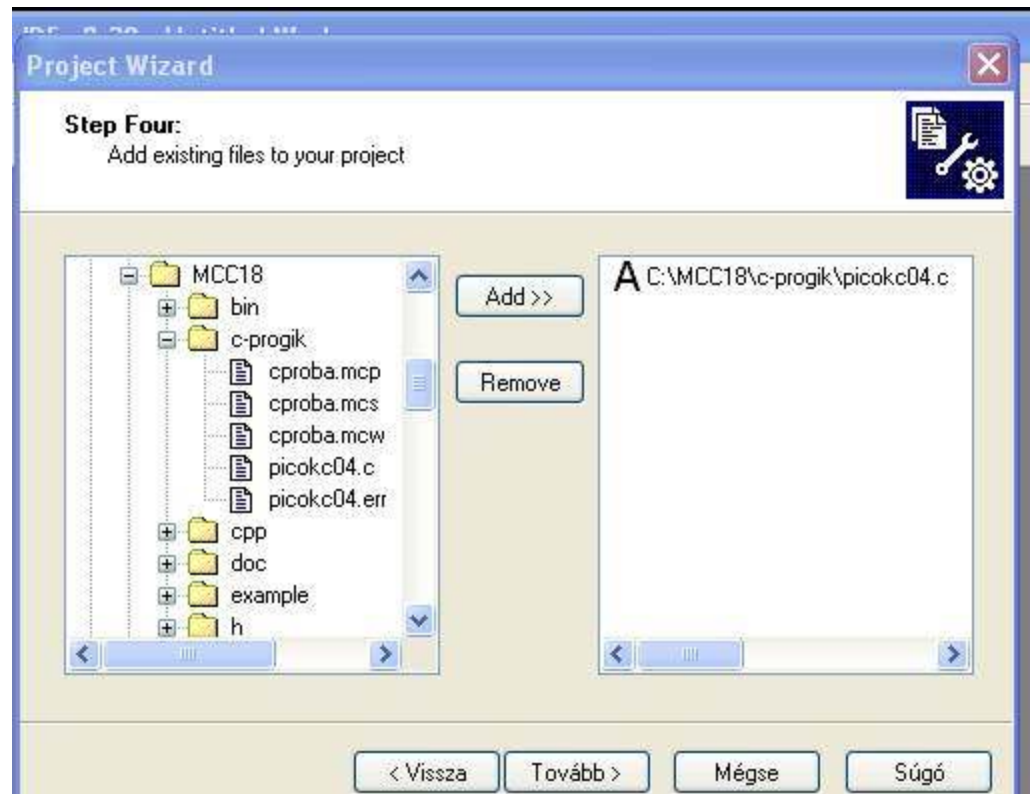
☐ Save changes to existing project file

☐ Save changes to another project file

Browse...

< Vissza Tovább > Mégse Súgó

Adjuk hozzá a
„projecthez” a
PICOKC04.C forrás
programunkat:



Ezzel tulajdonképp meg is lenne a „project”, és ha a Build All menüre kattintunk, akkor lefordulna a programunk. De nem...

Egy „Build Failed” hibaüzenete kapunk:

```
Clean: Deleted file C:\MCC18\c-progik\cproba.mcs
Clean: Done.
Executing: "C:\MCC18\bin\mcc18.exe" -p=18F452 "picokc04.c" -fo="picokc04.o" -D__DEBUG -Ou -Ot -Ob -Op -Or -Od -Opa-
C:\MCC18\c-progik\picokc04.c:15:Error [1027] unable to locate 'p18F4580.h'
MPLAB C18 v3.33 (evaluation)
Copyright 2000-2009 Microchip Technology Inc.
Days remaining until evaluation becomes feature limited: 60
Halting build on first failure as requested.

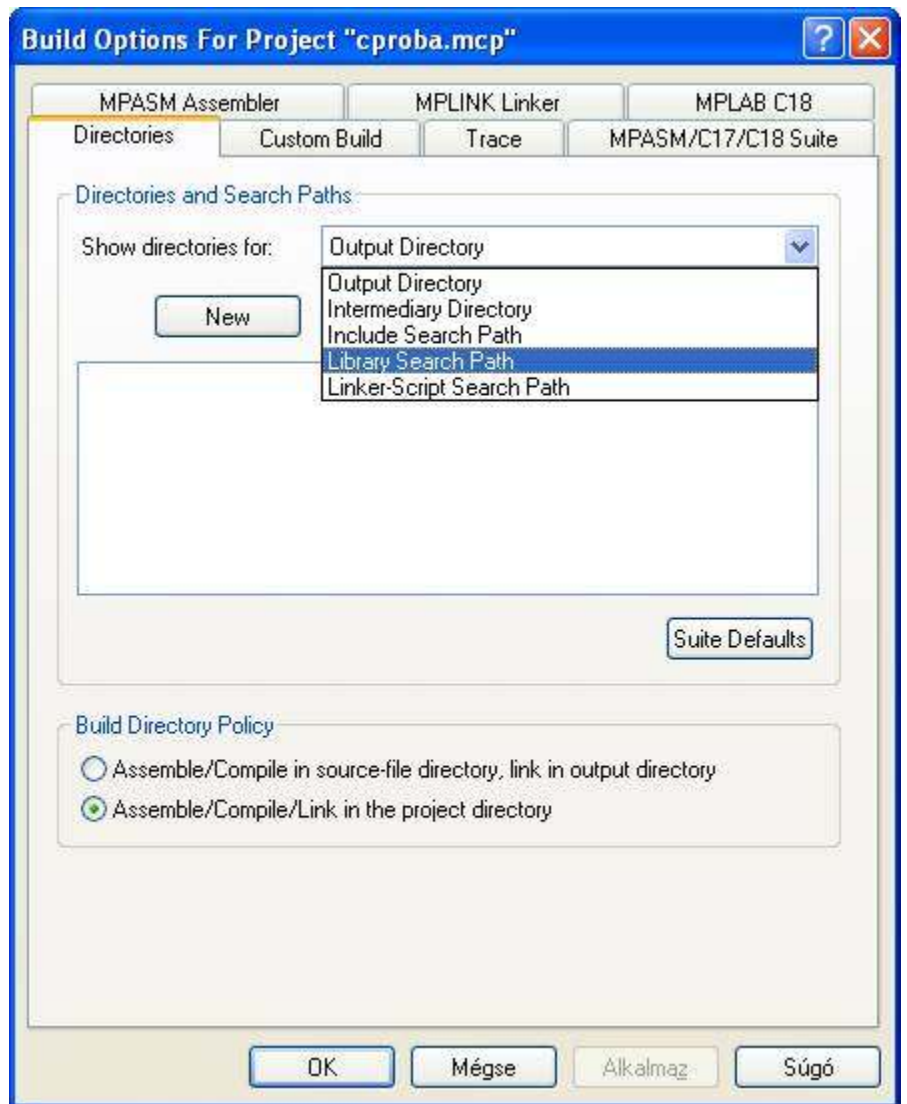
Debug build of project 'C:\MCC18\c-progik\cproba.mcp' failed.
Language tool versions: mcc18.exe v3.33, mplink.exe v4.33, mcc18.exe v3.33
Preprocessor symbol '__DEBUG' is defined.
Tue Sep 01 14:29:17 2009

BUILD FAILED
```

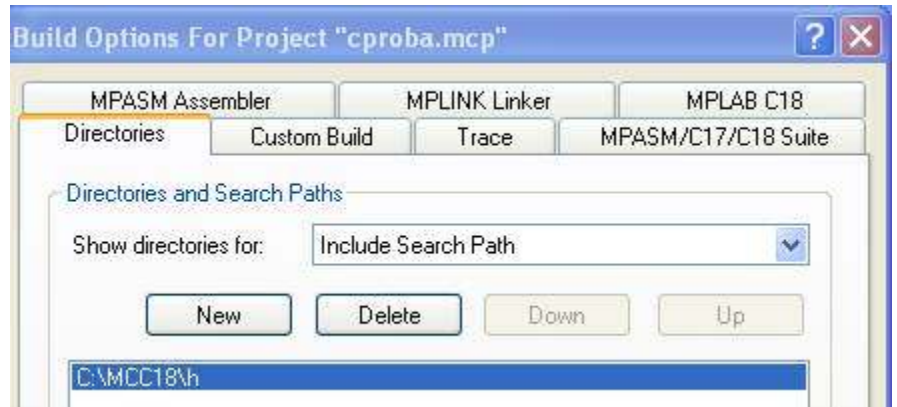
De miért?

A lényeg, hogy az MPLAB nem talál néhány, a fordításhoz szükséges fájlt.

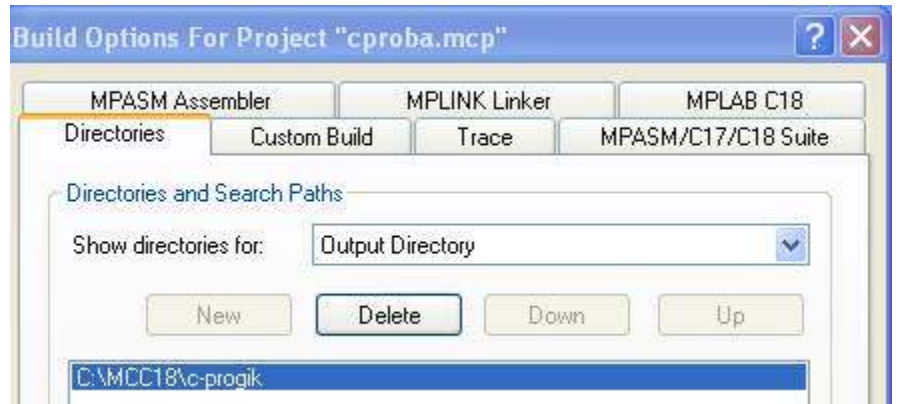
Ezért lépünk be a „Build Options For Project” menübe, és állítsuk be, hogy a „Library” fájlokat, az MCC18\LIB könyvtárban keresse.



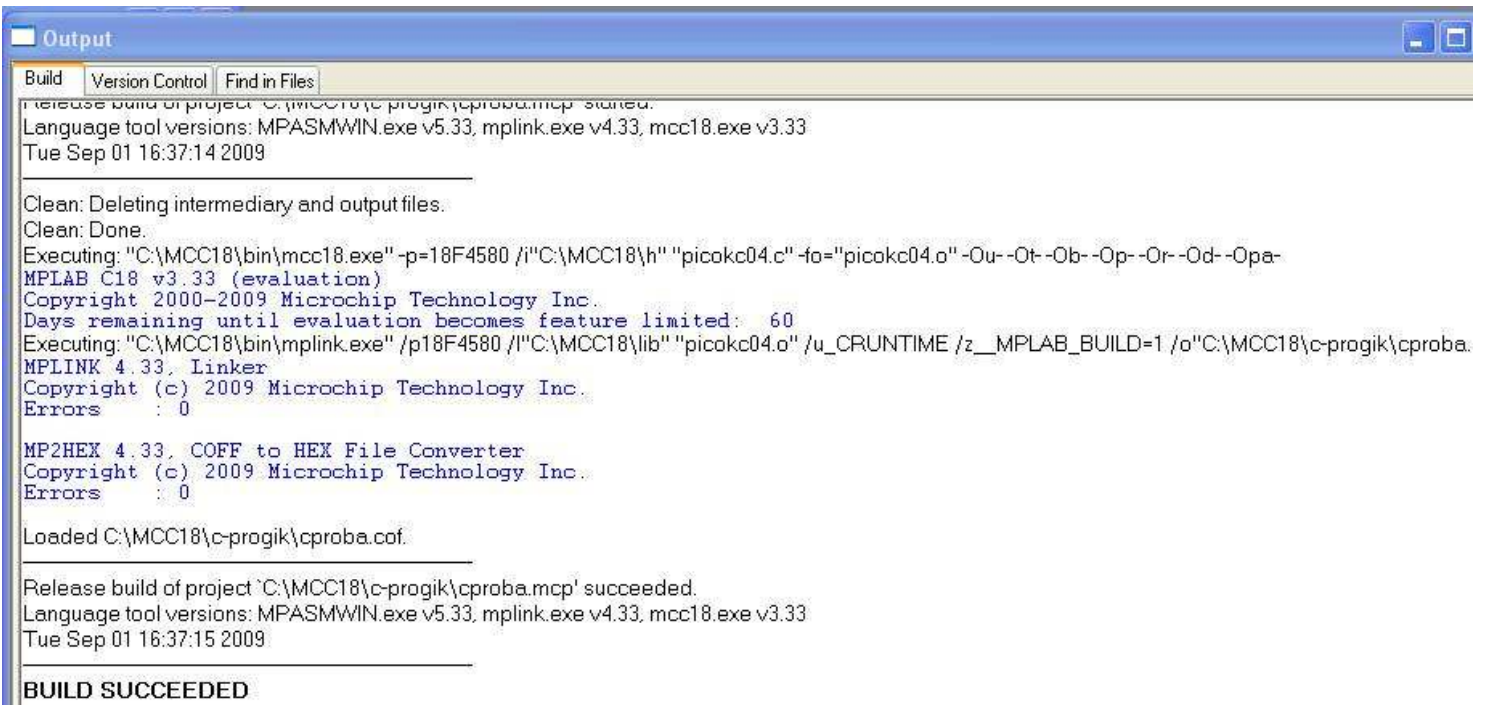
Az előbbihez hasonlóan állítsuk be az „include” fájlok elérését, és azt is



hogy „C-PROG1K” könyvtárunkba rakja a kimenő fájlokat is. (Legyen minden együtt. A forrás is, meg amit abból a fordító csinált.)



Na és akkor egy újabb próba a „Build All”-ra kattintva, és - ha minden rendben - akkor „BUILD SUCCEEDED”, azaz sikeres volt a fordítás.



A „C-PROGIK” könyvtárunkban pedig - néhány másik fájl kíséretében - megjelenik a CPROBA.HEX fájl, ami a 18F4580 mikrokontrollerbe égetendő tartalom.

c:\Mcc18*					c:\Mcc18\c-progik*				
Név	↑Kit.	Méret	Dátum	Attr.	Név	Kit.	Méret	↓Dátum	Attr.
↑...[...]		<DIR>	2009.09.01 14:05 ----		↑...[...]		<DIR>	2009.09.03 15:48 ----	
[bin]		<DIR>	2009.09.01 12:27 ----		[cproba]	cof	30 433	2009.09.03 15:48 -a--	
[cpp]		<DIR>	2009.09.01 12:27 ----		[cproba]	hex	1 284	2009.09.03 15:48 -a--	
[c-progik]		<DIR>	2009.09.03 15:48 ----		[cproba]	map	163 687	2009.09.03 15:48 -a--	
[doc]		<DIR>	2009.09.01 12:27 ----		[picokc04]	o	4 955	2009.09.03 15:48 -a--	
[example]		<DIR>	2009.09.01 12:27 ----		[cproba]	mcp	1 099	2009.09.01 16:39 -a--	
[h]		<DIR>	2009.09.01 12:27 ----		[cproba]	mcw	23 552	2009.09.01 16:39 -a--	
[lib]		<DIR>	2009.09.01 12:27 ----		[picokc04]	c	1 818	2009.08.15 18:13 -a--	
[libtest]		<DIR>	2009.09.01 12:27 ----						

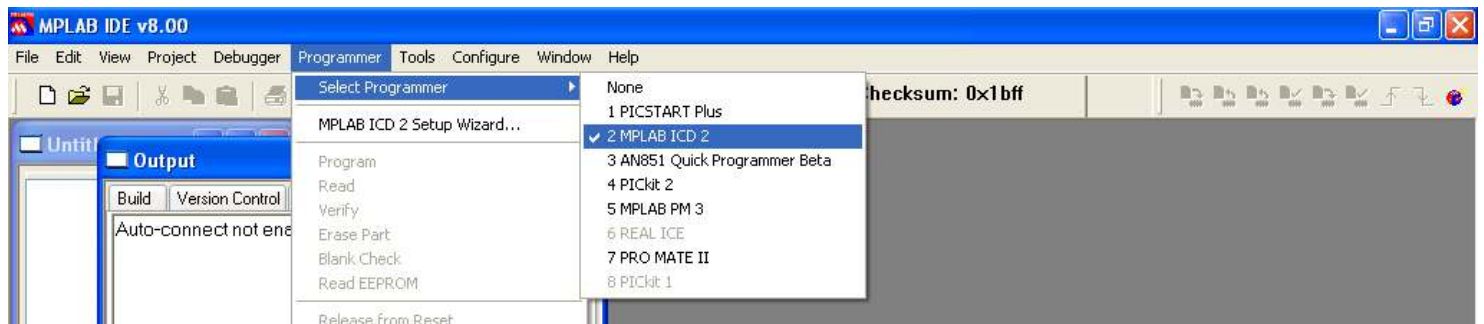
A program beégetése:

Jelen esetben a PIC égetőnek az ICD2-öt fogjuk használni. (Természetesen a soros portos, és az USB-s verzió is jó.)

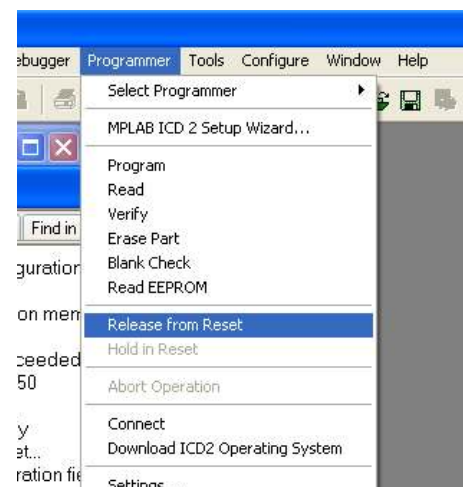
A program beégetéséhez persze össze kell állítanunk a hardwer-t is, a PICDEMO panelt kössük össze az ICD2-vel, az ICD2-öt a PC-vel, és adjunk nekik tápot. (Illetve elég az ICD2-nek, az tud tápot adni a PICDEMO panelnek.)

Az ICD2 használatáról, magáról az égetésről most csak nagy vonalakban, mert ez részletesen le van írva az ICD2, valamint a BASIC programozásról szóló PDF fájlokban is.

Szóval indítsuk az MPLAB-ot, válasszuk ki mint programozó eszközt az ICD2-öt!



- A "Settings"-ben válasszuk ki a PIC18F4580-at
- Kattintsunk a "Connect"-re!
- Most olvassuk be a már ekészült HEX fájlunkat, azaz "File" menü, "Import", és a CPROBA.HEX kijelölése.
- Jöhet a programozás:
- A programozás sikeresen megtörtént, utasítsuk az ICD2-önket, hogy "engedje el" a reset lábát. (Kattintás a Release from Reset-re.)



És - remélhetőleg - futnak a LED-ek. (Az ICD2 "fogja" az RB6, RB7 portot, ami az L6-7 LED-et is kezelné, ezek csak akkor fognak világítani, ha az ICD már nincs csatlakoztatva.)

Kapcsolódó dokumentáció, szakirodalom:

Magyar nyelven olvashatunk a mikrokontrollerek alkalmazásáról, programozásáról általában, és annak gyakorlatáról a PIC mikrokontrollerekre alapozva a PIC mikrovezérlők alkalmazástechnikája c. könyvben.

A könyv az elektronikai, programozási alap elemek - logikai kapuk, kettes számrendszer - ismertetésével indul, folytatva a mikrokontrollerek általános felépítével, majd rátér konkrétan a PIC-ek belsejére, utasítás készletére, végül néhány gyakorlati program példa. A könyvhöz egy CD melléklet, és egy "letöltő kulcs" is jár. (A Microchip CD és az előbbi könyv CD is megvásárolható a lenti címen.)

A MICROCHIP PIC-ek beégetését, de a program fejlesztést is nagyon segíti, a MICROCHIP ICD. (ICD2.PDF)

Egy konkrét példán keresztül, azaz az MPLAB-ban egy assembly nyelvű PIC-es példa program lefordításával, a "project" létrehozásával, annak letöltésének folyamatával foglalkozik az MPLAB.PDF dokumentáció.

A PIC-eket programozhatjuk BASIC nyelven is.

A PIC-BASIC PRO program használatáról, integrálásáról az MPLAB fejlesztői környezetbe, szól a PICBASIC.PDF

A PICDEMO panelnél ismertetett mikrokontrolleres „magra” épülve, vezérlési feladatokra készült a 8 relés kimenetű PICPLC8, és egy nagyobb, 16 relés kimenettel rendelkező változata, a PICPLC16. (PICPLC8.PDF, PICPLC16.PDF)

Az előbbi leírások, programok letölthetőek a lenti honlapcímről, vagy megtalálhatóak a „mikroklub cd”-n.

Végül nincs más hátra, mint hogy sok sikert kívánjak a használatához. Viszontlátásra: Torkos Csaba 8100 Várpalota Táncsics u. 7. Telefon: napközben: 88/473-784, egész nap: 06/30/9472-294, email: mikroklub@vnet.hu Internet: <http://www.mikroklub.hu>, <http://www.eprom.hu>