

MPLAB leírás V1.1



A következő összefoglaló az MPLAB használatát próbálja elősegíteni, az angol nyelvet kissé nehezen értők számára. Nem helyettesíti a kézikönyvet, és nem törekszik a teljességre. A menüvel való ismerkedésnek a legjobb módja, azok kipróbálása. Feltételezzük továbbá, hogy az olvasó jártas a Windows alapú környezet használatában. Ez az anyag az előző leírás kismértékű módosításának tekinthető. A változások: az újfajta projektszemlélet, és az ICD (In-Circuit Debugger) flash epromos olcsó fejlesztő eszköz MPLAB-ba illesztése.

Az MPLAB a PIC kontrollerekre írt programok fejlesztő eszköze. Tartalmaz egy szövegszerkesztőt (editort) a programok megírásához, assemblert a programok fordításához, szimulátort a programok teszteléséhez, valamint a Picstart Plus égető kezelőprogramját, amivel a fejlesztett program a EPROM-os vagy OTP-s PIC tokba beégethető. A program a PICmaster emulátort és egyéb eszközöket is tudja kezelni, a leírásban az ehhez kapcsolódó részek kimaradnak. Az anyagba bekerült ellenben az olcsó MPLAB-ICD fejlesztő használatának leírása.

Az ilyen, „mindent az egyben” típusú programot szokták **Integrált fejlesztő környezet**-nek (Integrated Developing Environment = **IDE**) is nevezni.

Az MPLAB parancsainak kiadására három lehetőségünk van:

- a menürendszerből választjuk ki,
- a hozzájuk rendelt billentyűkombinációt használjuk,
- az eszköztár megfelelő ikonjára kattintva aktivizáljuk.

Installálás után egy legördülő menürendszer jelenik meg, amelynek első eleme:

File	Project	Edit	Debug	Picstart
N ew Source				Ctrl+N
O pen Source...				Ctrl+O
V iew...				
S ave				Ctrl+S
S ave As...				
S ave All				
C lose				
C lose All				Shift+F9
I mport				
E xport				
P rint...				Ctrl+P
P rint Setup...				
E xit				Alt+F4
1 c:\mplab\munka\advstim.sti				
2 c:\mplab\munka\advstim.asm				
3 c:\mplab\munka\advbank.asm				

A **File** menüpont a szokásos fájlműveletek (**Save** (=mentés), **Save as...**(=mentés másként), **Save All** (az összes megnyitott fájl mentése), **Close** (=bezárás), **Close All** (az összes megnyitott fájl bezárása), mellett a következő műveleteket is biztosítja:

New Source - Új forrásfájl készítése

Open source- meglévő forrásfájl megnyitása.

Megjegyzés: a fenti két menüpont segítségével a projekten kívüli fájlok kezelésére nyílik lehetőség.

View - a kiválasztott fájl megnézése.

Import - **Download to memory** .HEX kiterjesztésű fájlt tudunk behívni pl. az égető program részére

Export

- **Save Trace Buffer:** a nyomkövetés (ld. később) eredményét menthetjük fájlba pl. további tanulmányozásra
- **Save Hex File:** a lefordított programunkat menthetjük el Intel hex formátumban

Print - A kiválasztott fájl nyomtatása

Print Setup - Nyomtató beállítás

Az utoljára használt fájlok listája van a menüpont végén.

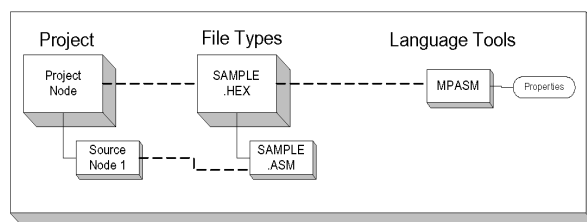
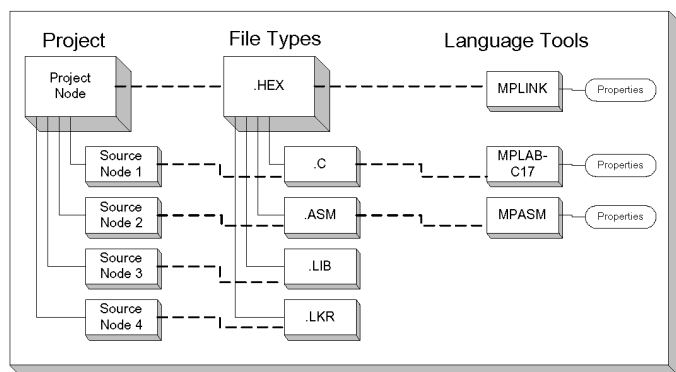
Projektek kezelése

Fontos jellemzője az MPLAB-nek a projekt koncepció: ez azt jelenti, hogy egy adott fejlesztéshez tartozó fájlokat egy ún. projekt fájlban nyilvántartjuk. Ha ezt a fájlt megnyitjuk, a nyilvántartott fájlok a fejlesztéskor azonnal betöltődnek és használhatók. A Project menüvel lehet a projekteket kezelni.

Ez a menü - amióta lehetséges különféle forrásból származó fájlokat összekapcsolva (ez a linkelés) egy összetett .HEX fájlt létrehozni - összetettebb lett.

Az MPLAB-ban a projekt egy projekt objektum (ez a node, azaz csomópont) egy vagy több forrás objektumból (node-ból) épül fel. Ezek a források általában MPASM assembler, illetve magas szintű C vagy BASIC forrásfájlok, előre lefordított könyvár fájl, más tárgykódok (object fájlok), illetve az ezek összekapcsolását végző **linker script** vezérlő szövegfájlok (hasonlók a DOS batch fájljaihoz).

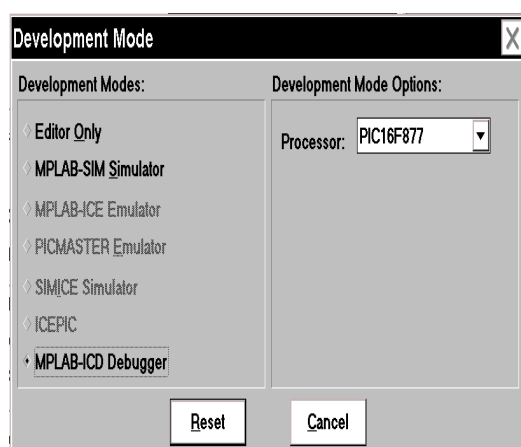
Általában a projektet ugyanabban a könyvtárban helyezük el mint a forrásfájlokat. Linkeléskor az összelinkelni kívánt ún. object fájlkból hozzuk létre a végleges lefordított tárgykódot, vagyis egy .HEX fájlt ilyen node-ok építik fel. Egy node egy forrásprogramot és egy fordítóprogramot (BASIC, C) vagy assembler programot is meghatároz ezeket hívjuk nyelvi eszközöknek (Language Tool = Nyelvi eszköz).



Mi a továbbiakban (mivel ez a leírás az MPLAB-al ismerekedőknek szól), csupán az olya projektek létrehozásának lépéseit írjuk le, amely egy MPASM forrásfájlt tartalmaz. A gyakorlatban is ez a legtöbbször előforduló eset.

1. **Új projekt létrehozásakor** először a program fejlesztési módját, és a használt PIC processzor típusát kell megadni az **Options>Development Mode** menüpontban.

Az itteni választási lehetőségek:



- **Editor Only** - ezt választva csak program írásra és fordításra van lehetőség
- **MPLAB-SIM - Simulator** ezt kell használni ha nincs külső eszköznünk
- **MPLAB-ICE Emulátor**
- **PICMASTER Emulátor**
- **SIMICE Simulator**
- **ICEPIC**
- **MPLAB-ICD Debugger** - ez az új lehetőség !

A megadás után a RESET-re kattintva történik meg a kiválasztás.

2. Utána a **Project>New Project** menüpontnál kell megadni a projekt fájl nevét (pl.: minta.pjt), és a könyvtárát, ahol el fogjuk helyezni.

3. Ezek után az **Edit Project** párbeszédés doboz jeleneik meg, ahol a benne lévő **Project Files** ablakban már látjuk a minta[.hex] bejegyzést. (Itt adhatjuk meg az Include, Library és Linker Script fájlok útvonalát, változathatjuk meg a már beállított fejlesztési módot, controller típust, valamint a használt nyelvi eszközt (fordító, vagy assembler programot)).

4. Ha rákattintunk a **minta[.hex]** bejegyzésre, megjelenik a **NODE Properties** párbeszédés ablak, ahol megnézhetjük illetve átállíthatjuk a nyelvi eszköz beállított alapértelmezéseit:

*Kiválaszthatjuk a programozónak vagy a szimulátornak szóló kimeneti hex fájl formátumát (**Hex Output**). Generáljon-e az assembler hibaüzeneteket tartalmazó fájlt (**Error file**), a forrásnyelvi kódot és a hozzá fordított gépi kódot tartalmazó **lista fájlt**, illetve **kereszt referencia fájlt** (ez azt tartalmazza, hogy a használt szimbólumok melyik program sorokban található.) Mikor nyissa meg a hiba fájlt megtekintésre: csak a hibáknál - hibáknál és figyelmeztetéseknél - bármilyen üzenet esetén. Ezeket a megfelelő jelölőnégyzetre kattintással tudjuk beállítani, amit a négyzet kijelölt volta is jelez, bár a beállításokat eleinte nem érdemes változtatni.*

Az OK-ra kattintás után az előző **Edit Project** ablakban az előzőleg szürke (és ilyen módon nem kiválasztható) **Add Node** gomb kivilágosodik, és rákattintva hozzárendelhetjük a minta.asm fájlt. Természetesen ennek a fájlnak már előzőleg léteznie kell !!! (Azaz az elsőnek megadott projekt fájl nevét már pontosan ez ok miatt adtuk meg minta.pjt néven). Ez azt jelenti hogy új projekt esetén először egy ASM fájlt kell létrehoznunk !!!

Egy új forrásfájl létrehozása a **File>New** menüpont segítségével lehetséges.

5. Ezek után a **Project>Make Project** menüpont aktivizálásával létrehozható a tárgykód.

Project	Edit	Debug	Picstart Plus	Q
New Project...				
Open Project...			Ctrl+F2	
Close Project				
Save Project				
Edit Project...			Ctrl+F3	
Make Project			F10	
Build All			Ctrl+F10	
Build Node			Alt+F10	
Install Language Tool...				
1 c:\progra~1\mplab\munka\primit.pjt				
2 c:\progra~1\mplab\pbl\blink.pjt				
3 c:\progra~1\mplab\pbl\lcd.pjt				

A project fájlok szokásos kezelési lehetőségei, nem részletezzük.

Az **Edit Project** menüpontban szerkesztjük meg a project fájlokat.

A **Make Project** segítségével elvégezzük az assemblálást az MPASM segítségével.

Build All menüpontot aktualizálva a projekt újrafordítása történik meg. *(Ez úgy történik, hogy a szimbólumokat tartalmazó .COD kiterjesztésű fájl létrehozási időpontját hasonlítja a projekt .INC és .ASM kiterjesztésű fájljaihoz. Ha a .COD régebbi mint a hasonlított fájlok valamelyike, (valószínűleg módosítva lett) akkor elvégzi az újrafordítást.)*

A **Build Node** menüpontot használhatjuk egy forrásfájl lefordítása projekt környezet nélkül. Ehhez minden projektet be kell zárni, majd a File>New vagy File>Open menüpontokkal létrehozott illetve beolvasott fájlt **Build Node** menüponttal lefordítjuk.

Az **Install Language Tool...** segítségével új fordítót (pl. C, BASIC) illeszthetünk az MPLAB környezetébe.

Az MPLAB szövegszerkesztője

Edit	Debug	Picstart Plus
Undo		Ctrl+Z
Cut		Ctrl+X
Copy		Ctrl+C
Paste		Ctrl+V
Select All		
Select Word		
Delete Line	Ctrl+Shift+K	
Delete EOL	Ctrl+K	
Goto Line...	Ctrl+G	
Find...	F3	
Replace...	F4	
Repeat Find	Shift+F3	
Repeat Replace	Shift+F4	
Match Brace	Ctrl+B	
Template		
Text		

Az MPLAB szövegszerkesztője lényegében egy átlagos Windows alapú szövegszerkesztő, de olyan részekkel lett kiegészítve, amelyek a programok megírását megkönnyítik

Az Edit menü az MPLAB szövegszerkesztőjéhez kapcsolódik.

Undo - utolsó szerkesztő művelet visszavonása,

Cut - kijelölt rész kivágása és a vágólapra másolása

Copy - kijelölt rész vágólapra másolása

Paste - a kurzorral megjelölt helyre a vágólap tartalmát másoljuk

Select All - a teljes szöveg kijelölése

Select Word - annak a szónak a kijelölése, amelyiken a kurzor áll.

Delete Line - a kurzort tartalmazó sor törlése

Delete EOL - a kurzorpozíciótól kezdve s sor végéig a törlés

Goto line - ugrás adott sorszámú sorra

Find - keresés

Replace - keresés és helyettesítés

Repeat Find - ismételt keresés

Repeat Replace - ismételt helyettesítés

Match Brace - ha a kezdő zárójelre állunk a kurzorral, akkor a parancs segítségével megkereshetjük a hozzá tartozó befejező zárójelet

Template - segítségével létrehozhatunk, kezelhetünk olyan állományokat, melyben a gyakran használt szövegrészeinket tárolhatjuk, és a forrásprogrambeli szöveg megfelelő helyére állítva a kurzort ezeket a szövegrészeket előhívhatjuk és a szövegbe illeszthetjük.

Transpose	Ctrl+T
Upper Case	
Lower Case	
Indent	
Un-indent	

A kurzor bal és jobb oldalán lévő két karaktert felcseréli

Nagybetűssé alakítja a kijelölt szöveget

Kisbetűssé alakítja a kijelölt szöveget

Bekezdést növel a kurzorral kijelölt sorban

Bekezdést csökkent a kurzorral kijelölt sorban

Az MPLAB szimulátora

A szimulátor program arra szolgál, hogy a PIC mikrokontrollerekre írt programot ellenőrzött módon lehessen végrehajtani. Ez azt jelenti, hogy a program végrehajtása közben:

- meg tudjuk vizsgálni és esetleg módosítani a kontroller bármelyik regiszterének tartalmát (watch),
- meg tudjuk a program futását adott helyen állítani, azaz töréspontot tudunk elhelyezni (breakpoint),

- tudjuk a programot lépésenként futtatni (single step),
- a külső környezetből jövő (input) adatokat fájlból beolvasva szimulálni (stimulus),
- illetve kiküldött (output) adatokat fájlba eltárolni,
- a szimuláció során bekövetkező eseményeket a későbbi elemzés céljából gyűjteni (trace).

A **Debug** menüpont az MPLAB legfontosabb része. Segítségével lehet a megírt programunkat tesztelni, szimulálni. Gerjesztő jeleket lehet előállítani, és a hatásukat a szimulációban nyomon követni.

A felhasználónak a program belövése során igen nagy segítséget nyújt, ha a programot egy, még az indítás előtt kijelölt címen meg lehet állítani, azaz a futását felfüggeszteni. Ekkor ugyanis a regiszterek és a programban használt változók tartalmának vizsgálatával könnyen eldönthető, hogy a program eddig a pontig helyesen, vagy hibásan működött. A teljes programot több, logikailag jól elhatárolt részre bontva, a hibásan működő programrész gyorsan meghatározható. Az ilyen leállási feltétel--azaz **töréspont** (angolul **breakpoint** (e.: brékpont)-- a programban több is elhelyezhető. Az is fontos lehet hogy egy program szimulálása közben a futás során a használt regiszterekről, azok tartalmáról adatokat gyűjtsünk. Ez a nyomkövetés, vagy más néven **tracing**.

A **lépésenkénti program végrehajtás (Single Step)** üzemmódban a felhasználói programnak mindig csak egy utasítása hajtódik végre, és ilyenkor mindig lehetőség van a regiszterek vagy a tártartalmak vizsgálatára, vagyis a végrehajtott utasítás hatásának, a program helyes működésének az ellenőrzésére. Lehetőség van szubrutinok egy lépésben történő végrehajtására is.

Debug	Picstart Plus	Options	Tools
R un			
E xecute			
S imulator S timulus			
C enter D eb <u>ug Location</u>			
B reak S ettings...			F2
T race S ettings...			
T rigger O ut S ettings...			
C lear A ll P oints...			
C lear P rogram M emory...			Ctrl+Shift+F2
S ystem R eset			Ctrl+Shift+F3
P ower- O n- R eset...			Ctrl+Shift+F5

A program futásával kapcsolatos tevékenységeket tartalmazza

Execute an Opcode - Egy beírt utasítást végrehajt

Conditional Break - ld. később

Simulátor Stimulus azaz a Gerjesztések - ld. lent

Az aktuális programszámlálóhoz tartozó programrészletet a képernyő közepére teszi

Töréspontok beállítása

Nyomkövetés beállítása

Emulátorhoz....

Minden törés- és nyomkövetési pont törlése

A program memória törlése

Mintha az MPLAB-ot újra indítottuk volna

Mintha a kontrollert éppen bekapcsoltuk volna..

A **Run** menüpont részletesebben:

R un	F9	A program futtatása
R eset	F6	Alaphelyzetbe állítás, programszámlálóba nulla kerül
H alt	F5	A futó program megállítása, programszámláló megáll
H alt T race	Shift+F5	Nyomkövetés megállítása, de a program továbbfut
A nimate	Ctrl+F9	Futtatás, a képernyőn történő frissítéssel, „látjuk a futást”
S tep	F7	Lépésenkénti program végrehajtás
S tep O ver	F8	Lépésenkénti futtatás, a rutinokat egy lépésben hajtja végre.
U ppdate A ll R egisters		Minden regiszter aktualizálása a képernyőn
C hange P rogram C ounter...		Az utasításszámláló megváltoztatása

Conditional Break (Feltételes töréspont)

A menüpont aktivizálásakor egy dialógusdoboz jelenik meg, amivel automatikus lépésenkénti programvégrehajtást indíthatunk. Akkor áll meg a program futása, ha valamelyik megadott feltétel teljesül., (vagy a Halt gombot nyomjuk meg...). A dialógusdobozban szereplő elemek magyarázata:

- **Single Cycle** - a feltételt minden utasítás végrehajtásakor ellenőrzi.
- **Multiple Cycles** - A feltételt csak a normál, definiált töréspontok elérésekor ellenőrzi

- **Update Display** - a feltételes töréspontot végrehajtja, de a disassemblált listát tartalmazó képernyőt nem frissíti

A következő megállási feltételek választhatók:

User Halt - Halt gombra áll meg meg a futás

Number of Cycles - adott számú ciklus után áll meg

Register Value Conditions

Valamelyik fülregiszter tartalmát hasonlítjuk egy konstanshoz (=, <, >, <=, >=)

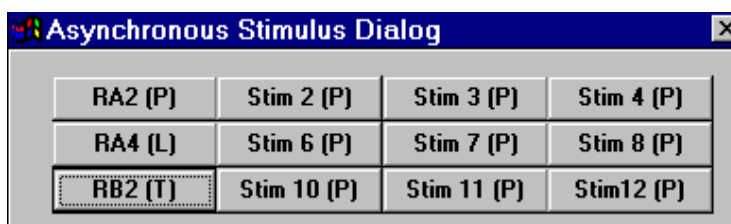
Meg kell adni a regisztert (pl. 02ah) és az értéket (pl. 33h)

- **Trace Data** - A kiválasztott regiszterek tartalma minden alkalommal amikor a processzort leállítjuk, egy listába kerül, ahol látható lesz.
- **Edit** - itt lehet az előbb említett regisztereket kiválasztani
- **Start** - lépésenként hajtja végre a programot
- **Halt** - a feltételes törésponti végrehajtás megállítása
- **Reset** - processzor alapállapotba állítása
- **Break Setting** - Töréspont beállítási dialógus ablak megnyitása
- **Save Buffer** - a törésponti tároló tartalmának *.TB fájlba mentése

A lábakra adható gerjesztések (Stimulusok)

A szimuláció során a legnehezebb feladat a külső környezet figyelembe vétele. A külvilággal való kapcsolat szimulációjának megvalósítására az MPLAB több lehetőséget is felkínál:

Aszinkron stimulus : Segítségével a szimulált controller lábaihoz gerjesztéseket rendelünk, és ezeket a képernyőn lévő nyomógombsor segítségével aktivizálhatjuk. Egy gombhoz hozzá kell rendelnünk a lábat, és a gerjesztés típusát:



Low - a gomb megnyomásakor alacsony szint lesz az adott lábon

High - a gomb megnyomásakor magas szint lesz az adott lábon

Toggle - a gomb megnyomásakor az adott lábon lévő szint ellentétére vált

Pulse - a gomb megnyomásakor egy impulzus jelenik meg az adott lábon

Az aszinkron stimulust csak a program futása közben használhatjuk!

Pin stimulus : Egy szöveges ún. stimulus fájlban adhatjuk meg, hogy adott cikusszám után milyen legyen az adott láb gerjesztése. A fájl első sorában szerepel a STEP szó és utána a szimulált lábak szabványos megnevezése:

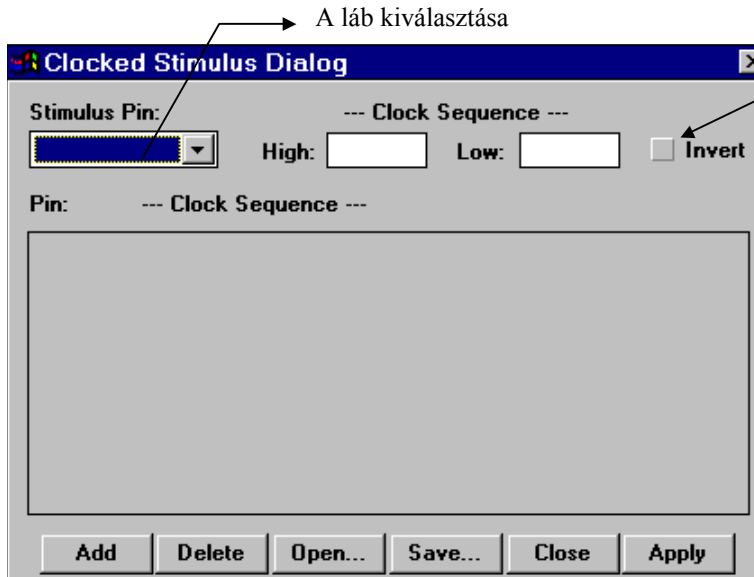
STEP	RB3	RB0
4	0	1
13	1	0

Azaz a 4. ciklusban RB3 lábra 0 RB0-ra 1 kerül, míg a 13.-ban RB3 lesz 1 és RB0 pedig 0.

Clock stimulus:

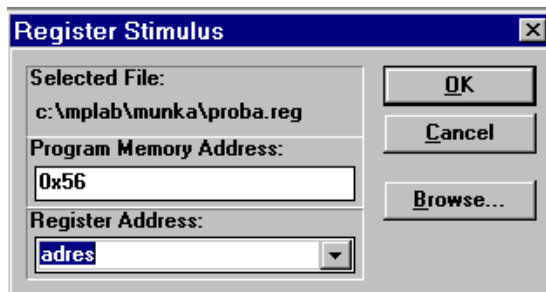
Egy I/O lábhoz ismétlődő periodikus négyszögjelet rendelünk: meg kell adni hogy hány ciklusig legyen a láb állapota magas (high), és hány ciklusig alacsony (low) szintű.

A ciklusszámot decimális számként kell megadni.



Az Invert bejelölésével a jel ellentétes lesz (alacsonnyal kezd). Az alsó sorban lévő menüpontokkal több gerjesztést is használhatunk egyszerre (**Add**), törölhetünk közülük (**Delete**), fájlból tölthetjük (**Open**) fájlba menthetjük (**Save**), a dialógusdobozt bezárhatjuk (**Close**), a kiválasztott jelet aktivizálhatjuk (**Apply**).

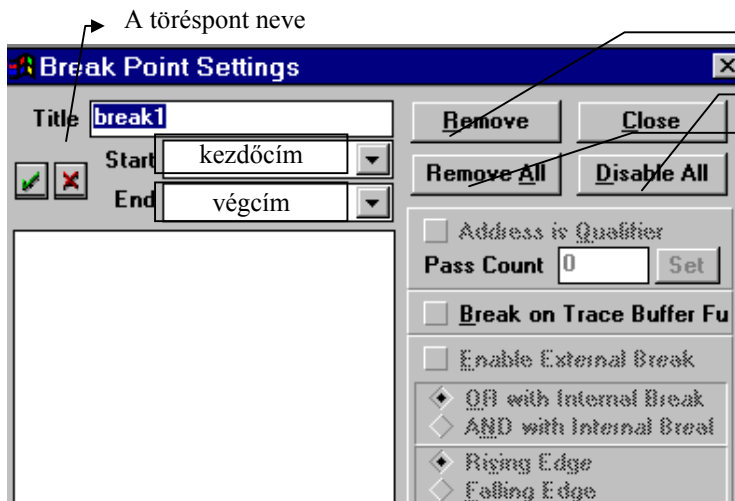
Register Stimulus:



Mikor a szimulációban egy megadott címre érünk, egy megadott regiszterbe egy szövegfájlból egy adott értéket töltünk. Ha többször haladunk át az adott címen, akkor a fájlban szereplő egymás utáni értékek sorban egymás után betöltődnek a regiszterbe. Az ábrán például az A/D átalakítás „eredményét” felhasználhatjuk, az ADRES regiszterbe a 0x56-os címet elérve a proba.reg fájlban szereplő értéket (pl. 0x11) tölti a szimulátor.

A gerjesztéseket tartalmazó fájl keresése.

Töréspontok beállítása



Töréspont eltávolítása

Az összes töréspont kikapcsolása

Az összes töréspont eltávolítása

A maximum 16 töréspontot a dialógusablak segítségével adhatjuk meg. A töréspontnak nevet adhatunk (**Title**). Beállíthatjuk a törésponti tartomány kezdő- és végcímét (ha a kettő azonos, akkor tényleg töréspont).

Address is Qualifier - Ha töréspont létezik, akkor törésponti címhez megadhatjuk (**Pass Count**), hogy hányzori ezen címen történő átfutás után álljon meg a futó program

Break on Trace Buffer Full - a program futása megáll, ha a nyomkövetést tároló puffer megtelik (8 k). Emulátornál külső esemény is

okozhat törésponti megállást (**Enable External Break**, és az alatta lévő pontok).

A definiált töréspontokat mint a projekt része az MPLAB automatikusan elmenti kilépéskor.

Nyomkövetés beállítása: dialógusdobozza majdnem azonos a töréspontiéval. Hasonló módon adhatjuk meg azt a címet illetve tartományt, amely elérésekor a nyomkövetés adatait tárolni kezdjük a töréspont pufferben.

Picstart Plus programozó

A Picstart Plus programozó segítségével tudjuk a fejlesztett programjainkat EPROM-os vagy OTP-s tokba írni, és a valódi hardver környezetben kipróbálni. A programozó minden PIC controller írására képes, az újabb típusok megjelenésekor a MCHIP adja azt a .HEX kiterjesztésű fájlt, amelyet egy PIC17C44-es tokba írva, a programozó aktualizálható (upgrade).

A programozót a számítógép valamelyik soros vonalára kell csatlakoztatni. A programozás során az MPLAB belső memória pufferének tartalmát írjuk a programozóban lévő tokba, a beállított konfigurációs bitekkel együtt. A puffert vagy egy fájlból töltjük fel (**File-> Import ->DownloadToMemory**), vagy az assemblálás során keletkezik. A programozóval pufferbe beolvasott PIC tok tartalmát fájlba is menthetjük (**File->Export->Save .HEX file**)

Picstart Plus	Options	Tool
Enable Programmer		A programozó engedélyezése
Program/Verify		Programozás, a beírt program ellenőrzése
Read Device		A programozóba helyezett PIC tok olvasása
Blank Check All		A tok ürességének (memória + konfigurációs bitek) vizsgálata
Blank Check QTP		OTP tok ürességének vizsgálata
Display Error Log		A programozás vagy ellenőrzés során kiírt hibák megnézése
Erase Program Memory		A program memória puffer törlése
Erase Configuration Bits		Konfigurációs bitek törlése
Reset Programmer		Programozó alaphelyzetbe állítása

Az Intel hexa formátum

Az Intel hexa formátumnál az adatokat rekordokba szervezik. Egy rekord változó hosszúságú lehet.

A rekordban az első mező a **rekordjelző mező (Record Mark Field)**. Ez a mező jelzi a rekord kezdetét és egy ASCII kettőspontot tartalmaz (:).

A második mező a **rekordhossz mező (Record Length Field)**. Ez a mező két ASCII karaktert tartalmaz, melyek jelzik a rekordban lévő adatbájtok számát. A hexadecimálisan adott adatbájtok számát két ASCII karakterre konvertálva adódik ki a mező két karaktere, a magasabb helyiértékű digit szerepel előbb. Egy rekordban maximálisan 255 adatbajt lehet.

A harmadik mező a **betöltési cím mező (Load Address Field)**. Ez a mező négy ASCII karaktert tartalmaz, a rekord hexadecimálisan adott betöltési címének ASCII karakterre konvertált értékét az alábbi sorrendben: a cím felső bájtnak magasabb helyiértékű digitje, a cím felső bájtnak alacsonyabb helyiértékű digitje, a cím alsó bájtnak magasabb helyiértékű digitje, a cím alsó bájtnak alacsonyabb helyiértékű digitje. A rekordban lévő első adatbajt a betöltési címre töltődik, az utána következő adatbájtok a sorban következő címekre. A fájlvége (End Of File, EOF) rekordban ez a mező négy ASCII nullát tartalmaz vagy a program kezdőcímét.

A negyedik mező a **rekord típus mező (Record Type Field)**. A rekord típus adatrekord esetén 00, fájlvége rekord esetén 01. Ez a mező két ASCII karaktert tartalmaz, a rekord típus ASCII karaktereit, a magasabb helyiértékű digit szerepel előbb.

Az ötödik mező az **adat mező (Data Field)**. Ez a mező tartalmazza az aktuális adatokat két-két ASCII karakterre konvertálva, a magasabb helyiértékű digit szerepel előbb. A fájlvége rekordban nincs adat mező.

Az utolsó mező az **ellenőrző összeg mező (Checksum Field)**. Az ellenőrző összeg a második, harmadik, negyedik és az ötödik mező hexadecimális bájtnak 8 bitre csonkított összegének kettes komplemente. Az így kapott összeget két ASCII karakterre konvertálva kapjuk a mező két karakterét, a magasabb helyiértékű digit szerepel előbb.

A leírásból is látható, hogy egy Intel hexa formátumú fájl egy szöveges (ASCII) fájl, így pl. egy szövegszerkesztő vagy listázó program segítségével megvizsgálhatjuk a tartalmát. Példa Intel hexa formátumra:

```
:10200000455A5420455244454D455320564F4C5453
```

```
:0B201000204D454746454A54454E49C7
```

```
:00000001FF
```

A leíráshoz kapcsolódva, az első sor mezői:

1.	2.	3.	4.	5.	6
:	10	2000	00	455A5420455244454D455320564F4C54	53

Opciók

Az Opciók menüpontban az MPLAB beállításait tudjuk megnézni, illetve módosítani.

Options	Tools	Window	Help
Development Mode...			
Window Setup			
Default Editor Modes...			
Current Editor Modes...			
Reset Editor Modes			
Key Mappings...			
Environment Setup... Ctrl+F7			
Colors...			
Processor Setup			
Programmer Options			

Fejlesztési mód: csak editor, vagy szimulátor (vagy emulátor...)
A beállított ablakkonfigurációt tudjuk elmenteni
Alapértelmezés szerinti szerkesztési beállítások
Aktuális szerkesztési beállítások
Szerkesztési mód alaphelyzet
Tevékenységekhez billentyűkombináció rendelése
MPLAB környezeti beállításai
Színbeállítások...
Processzor
Programozó típus választása

A **Development Mode** ablakot a projekt létrehozásakor a leírás elején már megmutattuk. Összefoglalva, az MPLAB több fejlesztési módban képes dolgozni: csak editort használva lehetséges programírás és fordítás. Szimulátor módban lehet a programot szimulátoron futtatni, és lehetséges emulátor alkalmazása is. Mindezen üzemmódok kiválasztásánál kell (kötelező) megadni a használt processzor típusát is.!

Ha az ICD-t használjuk, akkor a **Options>Development Mode** párbeszéd ablakban a MPLAB-ICD Debugger jelű gombot, és a PIC16F877 controller típust kell beállítani. Természetesen először a debuggert soros porton kell csatlakoztatni, majd a megfelelő COM portot és adatátviteli sebességet (pl. 192000) is be kell állítani. Ezek után a a Reset-re kattintva az MPLAB ICD működésre kész, (az esetleges hibákról itt nem írunk...) és megjelenik az MPLAB-ICD párbeszéd ablak.

A felső ablakfél a PIC-eket ismerőknek egyértelmű, lényegében az emulátorban használt 16F877 tok konfigurációja.

A beírt program ellenőrző összege és tokazonosító megadása

Programozási paraméterek: mettől-meddig, és mit programozunk?

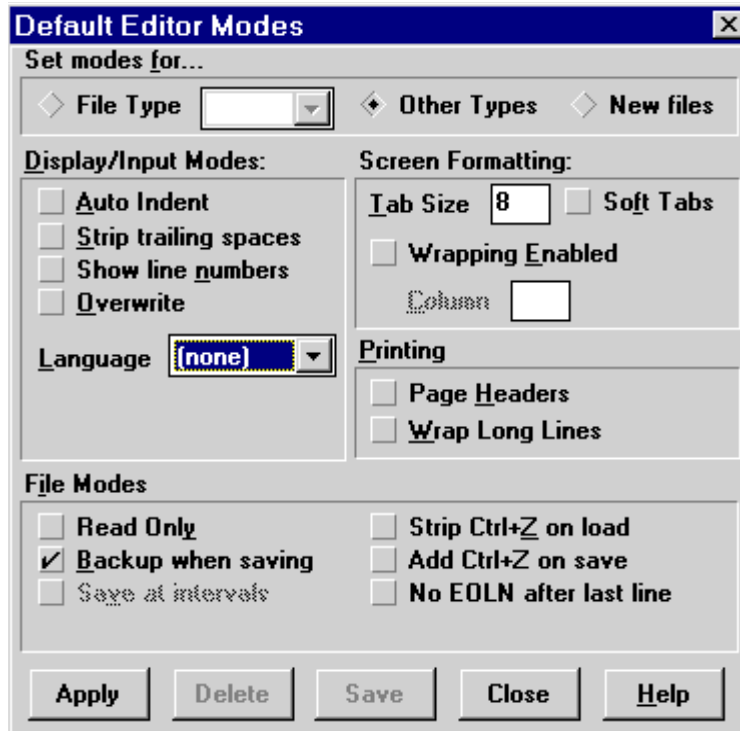
Mit olvasson fel a tokból lépésenként. (vigyázat az itt látható az alapértelmezés, és ez nagyon lelassítja a futást!

Az ICD állapota, alatta balra a soros port és sebesség. Órajel sebessége

Üres teszt Olvasás Programoz Ellenőrzés Törlés ICD önteszt

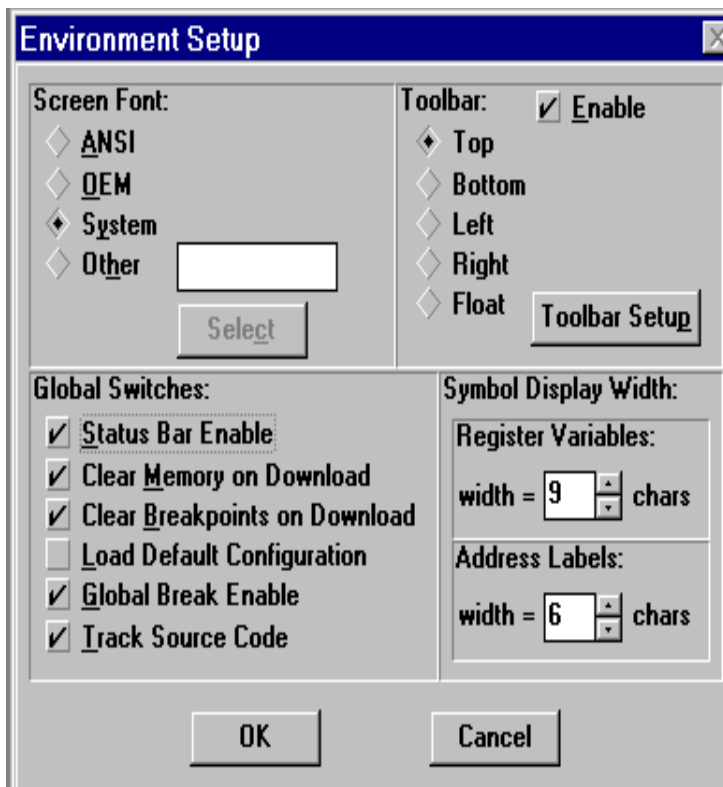
Szövegszerkesztő beállításai

Megadhatjuk, hogy a **Default Editor Modes**, azaz a szövegszerkesztő beállításai milyen fájlokra legyen érvényes: megadott kiterjesztésű fájlokra (File Type), éppen azokra ne (Other Types), illetve csak az új fájlokra (New Files).



FÁJLOK KEZELÉSE:

Read Only	A betöltött fájlok nem módosíthatók, csak olvashatók.
Backup When Saving	Másolatot (backup copy) készít mentéskor az eredeti fajlról.
Save at intervals	A fájl adott tartományát menti.
Strip Ctrl+Z On Load	Betöltéskor automatikusan a Ctrl+Z karaktert a fájl végéről levágja.
Add Ctrl+Z On Save	Mentéskor automatikusan a Ctrl+Z karaktert a fájl végére írja.
No EOLN after last line	Utolsó sor utolsó karaktere után sorvége jelet ír mentéskor a fájl végére.



A regiszterek és a címkék kijelzésénél meg lehet adni azt hogy azok mennyi helyet foglaljanak el.

KIJELZÉSI - BEVITELI MÓD

Auto Indent - automatikus bekezdés: az új sor annyival fog beljebb kezdődni mint az előző

Strip Trailing Spaces - a sor végén lévő jelentéssel úgy sem bíró betűközöket levágja ha Enter-t ütünk a sor végén

Show Line Numbers - a sorok számozva lesznek a képernyőn

Overwrite - felülírás üzemmód

Language - programnyelv választása

KÉPERNYŐ FORMÁZÁS

Tab Size - a tabulátor szélesség megadása (hány betűközt jelent)

Soft Tabs - TAB hatására adott számú betűközt helyettesít a TAB karakter helyett

Wrapping Enabled - sortörés adott sorhosszt elérve automatikusan megtörténik

Wrap Column - hányadik oszlopnál legyen a törés

NYOMTATÁS

Page Headers - új lap tetejére kiírja a fájl nevét, az oldalszámot és a dátumot.

Wrap Long Lines - a hosszú sorokat nyomtatásnál eltöri

Környezeti beállítások: a képernyőn használt font típusa adható meg. Az eszközsáv (**Toolbar**) engedélyezése (**Enable**), és ha igen akkor a képernyő melyik részén legyen. Az eszközsávok (Edit, Projekt, Debug és User) testre is szabhatók (**Toolbar Setup**).

Status Bar Enable - A képernyő alján lévő státusz sor engedélyezése

Clear Memory on Download - az emulátor memóriáját letöltés előtt törli (minden bit 1).

Clear Breakpoints on Download - az emulátorba való letöltés előtt minden töréspontot, nyomkövetési pontot töröl.

Load Default Configuration - az, MPLAB az alapértelmezett felhasználói ablakbeállításokat tölti be. Ennek megváltoztatása az *Options > Window Setup > Save Setup* menüpontban lehetséges. Az MPLAB.CFG az alapértelmezett felhasználói konfigurációt tartalmazó állomány.

Global Break Enable - Ha nincs kiválasztva nem engedélyezi a töréspontokat. A státusz sorban is be- illetve kikapcsolható (a kurzorral ráállva, dupla kattintás a bal egérgombbal).

Track Source Code - A forráskód aktuális sorát frissíti lépésenkénti programvégrehajtáskor.

Tools (Eszközök) menüpont

Tools	Window	Help
DOS Command to Window...	F11	
Repeat DOS Command to Window	Ctrl+F11	
Verify PICMASTER...		
Configure Probe...		

DOS parancs kiadása a futó MPLAB alól.

Ismételt parancskiadás

Emulátor ellenőrzése

Emulátor csatlakozófej (Probe) konfigurálása

Windows

Window	Help
Program Memory	
Trace Memory	
EEPROM Memory	
Absolute Listing	
Stack	
File Registers	
Special Function Registers	
Show Symbol List... Ctrl+F8	
Stopwatch...	
Project	
New Watch Window	
Load Watch Window...	
Modify...	
Tile Horizontal	
Tile Vertical	
Cascade	
Iconize All	
Arrange Icons	

Program memória megjelenítése

Nyomkövetési adatokat tartalmazó memória megjelenítése

EEPROM memória TARTALMÁNAK megjelenítése (16X84)

Teljes listafájl megjelenítése

Veremtartalom megjelenítése

Fájlregiszter - tartalom megjelenítése

Speciális funkciójú regiszterek megjelenítése

Használt szimbólumok listájának a megjelenítése

Stopper...

Projekt ablak megjelenítése

A felhasználó által szimuláció közbeni megfigyelésre kiválasztott regisztereket tartalmazó ablak létrehozása, illetve a már létrehozott(ak) betöltése

Módosítás

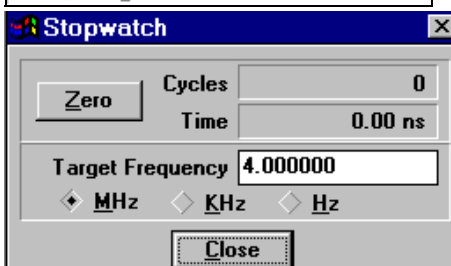
Ablakok vízszintes elrendezése

Ablakok függőleges elrendezése

Ablakok egymás mögötti elrendezése

Ablakok ikonokká zsugorítása

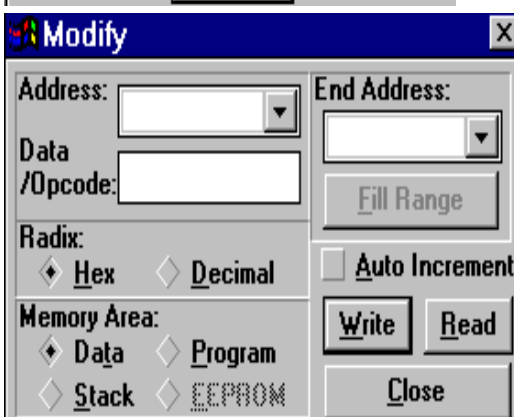
Ikronok rendezése



Tetszőleges programrészlet időbeli végrehajtását követhetjük nyomon. Megadhatjuk a szimulált processzor órajelét (**Target Frequency**) és mértékegységét.

A **Zero** gombbal nullázhatjuk a stoppert.

Kijelzésre kerül a nullázás óta végrehajtott ciklusok (a legtöbb esetben egy utasítás az egy ciklus) száma (**Cycles**), illetve az eltelt idő (**Time**).



Segítségével memóriaterületek (Data, Stack, Program, illetve EEPROM) tartalmát módosíthatjuk.

Ehhez meg kell adni a módosítandó terület kezdő (**Address**) és esetleges végcímét (**End Address**). Címként hexadecimális szám (Radix:Hex), vagy decimális szám (Radix:Decimal), illetve címke is megadható.

Data/Opcode mező - **Read** gombra kattintva kijelzésre kerül a kiválasztott címen lévő adat vagy utasításkód. **Write** gombra kattintva az ablakban lévő (megszerkesztett) érték íródik az adott címre

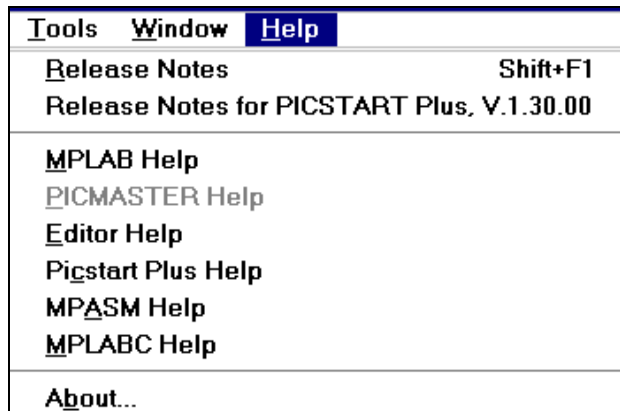
Fill Range - A kezdő és végcím közötti memóriaterületet a a Data/Opcode mező tartalmával tölti fel.

Auto Increment - Kiválasztva minden Read és Write parancs után a cím eggyel nő (inkrementálódik).

Close - A dialógusablak bezárása.

Help

Az MPLAB Help menüje lényegében minden információt tartalmaz a programra vonatkozóan, külön szerepeltetve az MPLAB egyes komponenseit. Az egyes sűgők a jobb eligazodást segitendő tematikus módon alfejezetekre is vannak osztva.



Az aktuális MPLAB verzióhoz kapcsolódó info.
Az aktuális PICSTART verzióhoz kapcsolódó info.

MPLAB program Sűgője
PICMASTER Emulátor Sűgője
Szövegszerkesztő program Sűgője
Picstart Plus programozó Sűgője
MPASM assembler program Sűgője
MPLABC C fordító program Sűgője
Az MPLAB-ról...

Összefoglalás

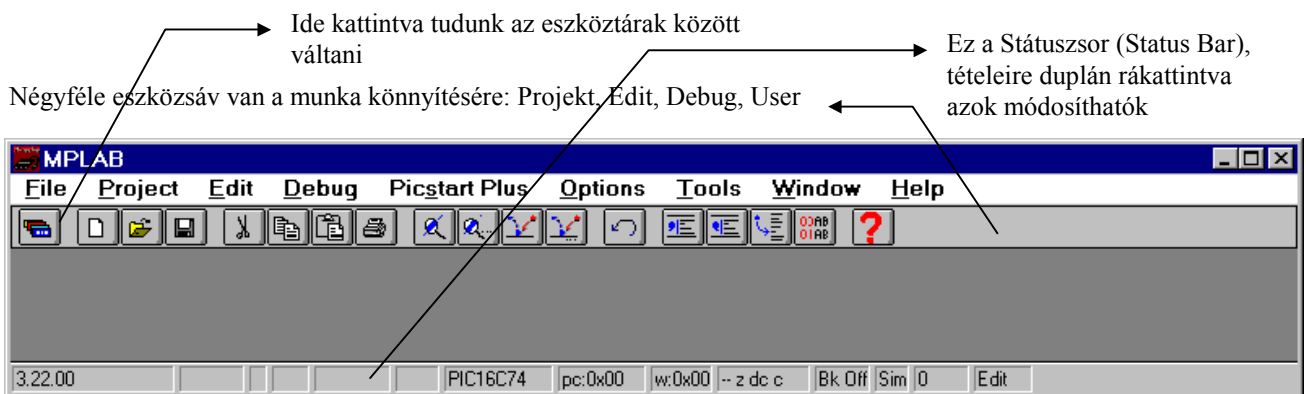
Foglaljuk össze azt, hogy hogyan kell használnunk az MPLAB-ot programjaink fejlesztésére. Első lépésként, ha még nincs projektünk a **File->New Source** menüpontjával létrehozhatunk egy új forrásfájlt. Másik, és valószínűleg jobb megoldás az, hogy a létrehozott új projektünkbe egy átnevezett .ASM fájlt illesztünk, mert azt könnyen módosíthatjuk a közös, szabványos részek meghagyásával. Ha szükséges. Akkor a projektbe INCLUDE fájlokat is beilleszthetünk.

Ezek után a fájlokat a **Projekt->Build All** menüpont segítségével lefordítjuk, az esetleges hibákat kijavítjuk. Ehhez az is segítséget nyújt, hogy a hibafájl megjelenítésekor rákattintva egérgombbal a kurzorral rámutatott hibás sorra, a forrásprogram megfelelő sora jelenik meg a szerkesztőben.

A fájlok futtatása a megfelelő **Debug** menüpont segítségével végezhető. Beállíthatunk töréspontok, nyomkövetést, és gerjesztések (stimulusok) segítségével a külső környezetet is szimulálhatjuk. A **Watch** ablak felhasználásával a futás során fontos regisztereket is figyelemmel kísérhetjük. **Fontos! A megfelelő működéshez a processzor típusát a forrászövegben megadott típushoz azonos módon kötelező megadni!**

Egy ilyen összetett eszköz mint az MPLAB kezelése, használatának elsajátítása szorgalmas munkát igényel. A munkát jelentősen megkönnyíti az eszköztárak és a státusz sor felhasználása. A leírás nem tért ki az egér használatára, de célszerű mindkét egérgombot különböző szituációkban kipróbálni, mert számos funkció valósítható meg az egér segítségével. (Például töréspontok, nyomkövetés beállítása a forrásnyelvi szövegben kurzorral az adott részre mutatva és a jobb egérgombbal kattintva.). Navigációs célzattal befejezésül itt látható az MPLAB képernyő képe.

MPLAB képernyőképe



MPLAB verzió

Proc. Progrszáml. w reg. státusz Break Üzemmod Aktuális eszközsor