

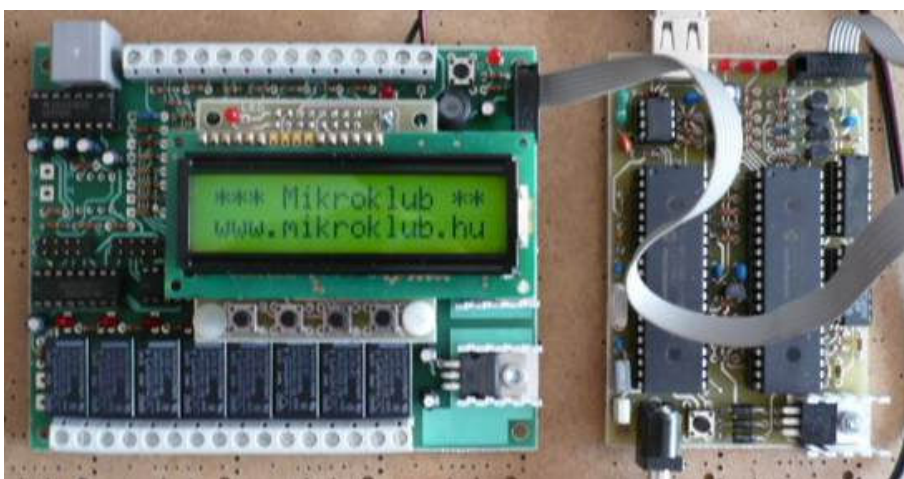
## A PIC-BASIC program használata

A következőkben egy-két gyakorlati példán keresztül próbálom leírni, hogyan használhatjuk a PIC BASIC PRO fordítóprogramot, tölthetjük be az elkészült program kódot az ICD2 programozó/debugger áramkörrel, egy PICPLC16 vagy PICPLC8 áramkörbe.

A PICBASIC PRO programot a Microengineering Labs árulja, leírások, példaprogramok, egyebek a [www.melabs.com](http://www.melabs.com) címen. Ugyanitt letölthető a program demó verziója is, amivel megoldhatunk kisebb feladatokat, valamint kipróbálhatjuk magát a programot.

Az ingyenes, demo verzió megtalálható a "mikroklub CD"-n is a MIKROKLB\PICBASIC könyvtárban. (Telepítsük.)

Egy egyszerű, az LCD-re író, és "relé billegtető" programot fordítunk le az ingyen BASIC-el, töltünk be, és futtatunk a PICPLC8 vagy PICPLC16 mikrokontrolleres vezérlő panelen.



Két BASIC programon keresztül lesz bemutatva az LCD, és a relé kezelés, illetve egy hőszenzor beolvasása. A PLCPROBA.BAS egy egyszerű "relé kattogtató", amivel tesztelhetjük az LCD és a relék működését. A DS1820.BAS pedig egy hőmérő, termosztát program.

### Röviden a PLCPROBA.BAS programról:

Először is megadjuk a BASIC fordítónak, hogy az LCD-nket mely portokon kezeli a mikrokontroller, milyen üzemmódban akarjuk használni - 4 bitesben - és milyen típust. (Két sorost.)

Majd az LCD törlés után kikerül a szöveg az LCD-re.

Ezután a reléket vezérlő 74HC4094-ekbe léptetünk 1010101010101010 biteket - minden második relé bekapcsolva - majd pedig egy másodperc várakozás után kezdődik a 0101010101010101 sorozat beléptetése, és így tovább...

```

plcproba.bas - Jegyzetfőmb
Ejál Szerkesztés Formátum Nézet Súgó

DEFINE LCD_DREG      PORTD      ' Az LCD adatbitek a D porton
DEFINE LCD_DBIT      0          ' Az első bit az RD0
DEFINE LCD_RSREG     PORTD      ' A register select bit a D porton,
DEFINE LCD_RSBIT     4          ' az RD4 port
DEFINE LCD_EREG      PORTD      ' Az enable bit a D porton,
DEFINE LCD_EBIT      5          ' az RD5 portra van kiosztva
DEFINE LCD_BITS      4          ' a 4 bites üzemmódot választjuk
DEFINE LCD_LINES     2          ' 2 soros az LCD

lcdout $fe,1                ' LCD torles, es iras az elso sorba
lcdout "*** Mikroklub ***"
lcdout $fe,$C0               ' LCD torles, es iras a masodik sorba
lcdout "www.mikroklub.hu"

' A relek valtott ki/bekapcsolasa:

W0 var Word
B6 var byte

STROBE var PORTB.5          ' A 74HC4094 kimeneti beiro jele a PORTB.5,
CLK var PORTB.7             ' az orajele a PORTB.7,
DAT var PORTB.6             ' az adat bemenete a PORTB.6-ra van kiosztva.
main: FOR B6 = 1 TO 16      ' Az adatok beleptetese a sorba kotott 4094-ekbe
  Toggle DAT                ' A relek sorban be-ki-be-ki-be... lesznek kapcsol
  HIGH CLK                  ' az orajelbemenet magasra kapcsolva
  Pause 1                   ' egy kicsit magasan tartjuk
  LOW CLK                   ' majd alacsony szintre kapcsolva egy oraimpulzust
  NEXT B6                   ' vissza a ciklusba, amig a 2x8=16 bit nincs belept

  HIGH STROBE                ' A beleptetett adatokat a kimenetre kapcsoljuk,
  Pause 1                   ' a "Strobe"-ra egy beiro jelet adunk.
  LOW STROBE                 ' A 4094 kimeneten megjelennek a beleptetett adatok

  Pause 1000                 ' 1000 msec, azaz 1 masodperc pause

  Toggle PORTA.4             ' Az RA4 porton vezereelt LED ki/bekapcsolgatasa
  Toggle DAT

```

Persze a program működésének megértéséhez kicsit bele kell ásnunk magunk az LCD-k, a 74HC4094, és magának a PIC BASIC utasításainak leírásába. A BASIC leírás PDF-je a telepítő program mellett található, a 74HC4094-é a MIKROKLB\PDF\TTL-CMOS\74HC4094.PDF-ben.

A PBPWDEMO.EXE fordító programnak meg kell adni, milyen processzort akarunk használni - most például 16F877-et - és persze a fordítandó programunk nevét:

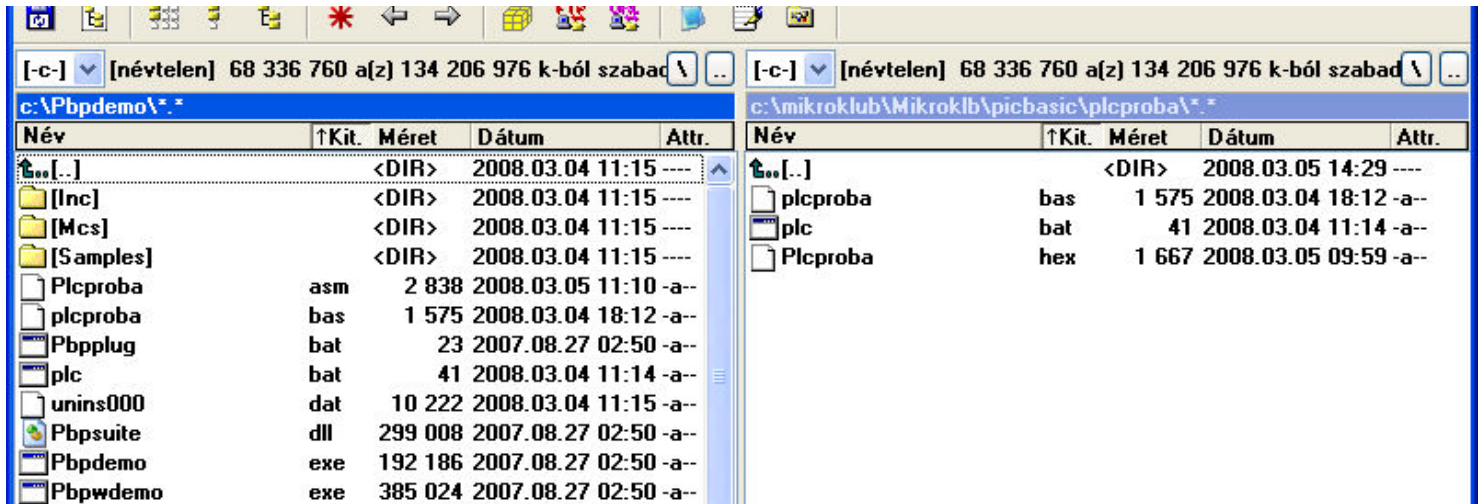
```
pbpdemo.exe -p16f877 plcproba.bas
```

Hogy lássuk is a fordítás eredményét, egy "pause" utasítást érdemes ezután kiadni, és hogy ezt ne kelljen mindig begépelni, egy batch programba beleírni ezeket. Ez lett a PLC.BAT

A mini BASIC programunk, és a PLC.BAT a MIKROKLB\PICBASIC\PLCPROBA könyvtárban található.

### És akkor sorban a lépések:

- Telepítsük a BASIC demo programot.
- Másoljuk be PLCPROBA.BAS és a PLC.BAT programot abba könyvtárba, amibe a BASIC fordító került.



- Most már le tudjuk fordítani a programot, kattintsunk a PLC.BAT-ra, ha minden rendben - és miért ne lenne - ezt látjuk:

```

C:\WINDOWS\system32\cmd.exe

c:\PBPDEMO>pbpwdemo.exe -p16f877 plcproba.bas
PICBASIC PRO(TM) Compiler Demo, (c) 1998, 2007 microEngineering Labs, Inc.
All Rights Reserved.

For more information, including purchase info, for this product,
visit melabs.com or call microEngineering Labs, Inc. at 719-520-5323.

A UDM IPX/SPX támogatást nem lehet betölteni.
PM Assembler 4.08, Copyright (c) 1995, 2006 microEngineering Labs, Inc.
289 words used.

c:\PBPDEMO>pause
A folytatáshoz nyomjon meg egy billentyűt . . . _

```

- Nyomjunk meg egy billentyűt, és meglátjuk, hogy elkészült a 16F877-be töltendő tartalom, a PLCPROBA.HEX

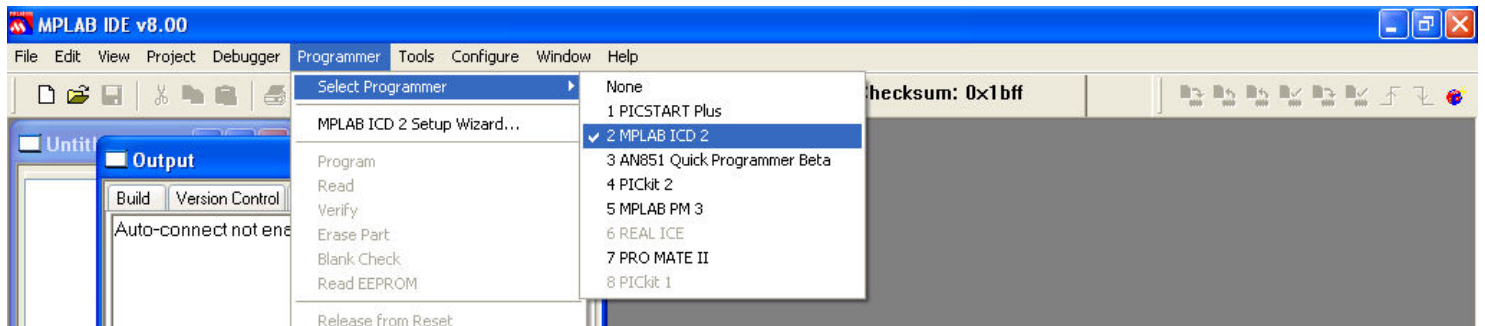
- A letöltéshez persze össze kell állítanunk a hardver-t is, a PICPLC8/16 panelt kössük össze az ICD2-vel, az ICD2-öt a PC-vel, és adjunk nekik tápot.

Logikusnak tűnne, hogy az assembly programok letöltésénél ismerttetett "download" programmal töltsük be az előállt hexe-et a PIC-be. Valóban, ha az assembly által előállított hexet be tudom tölteni, akkor a basic csinálta hexet miért nem? Mert egyrészt úgy kéne fordítani a basicnek a progit, hogy az első 3 bájt üresen maradjon - erre még nem tudtam rávenni - másrészt meg szabadon hagyja azt a területet, ahol a download progit tanyázik. Na ezt se tudom neki "megmondani", ezért **kell egy PIC égető a BASIC programozáshoz.**

- Ha még nem tettük meg, installáljuk az MPLAB programot a gépünkre, olvassuk el a PICDEMO, az MPLAB és az ICD-ről szóló leírásokat.

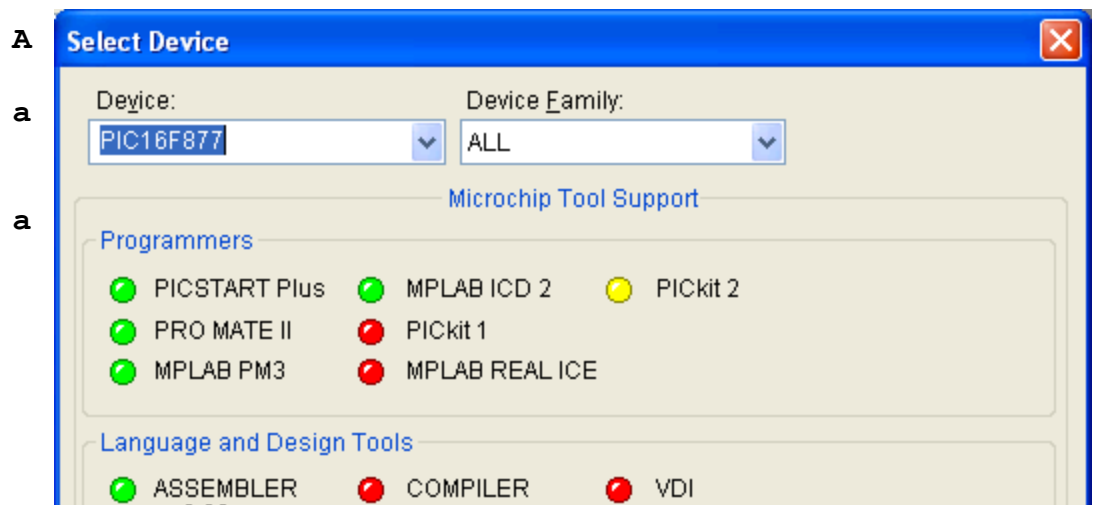
Konkrétan az MPLAB 8.0-ás verzióját használtam, egy XP-s gépen, a telepítő programja a \SHAREWARE\MPLAB8.0 könyvtárban található.

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

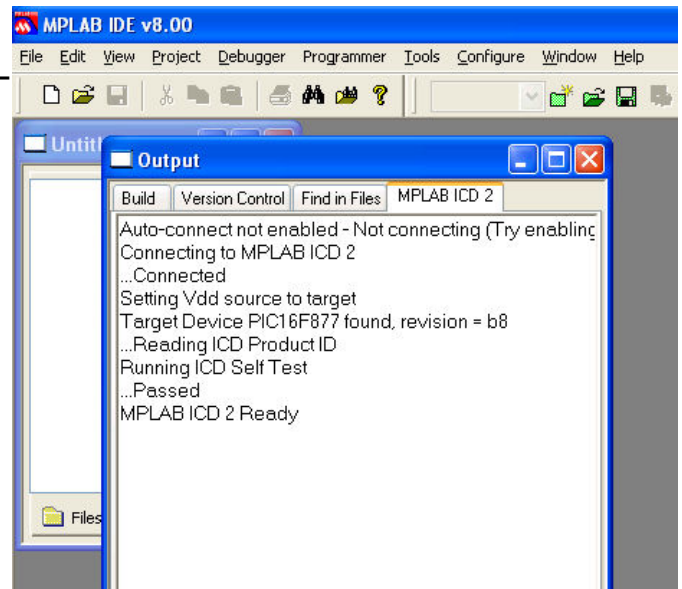


-  
"Settings"-ben  
válasszuk ki  
PIC16F877-et:

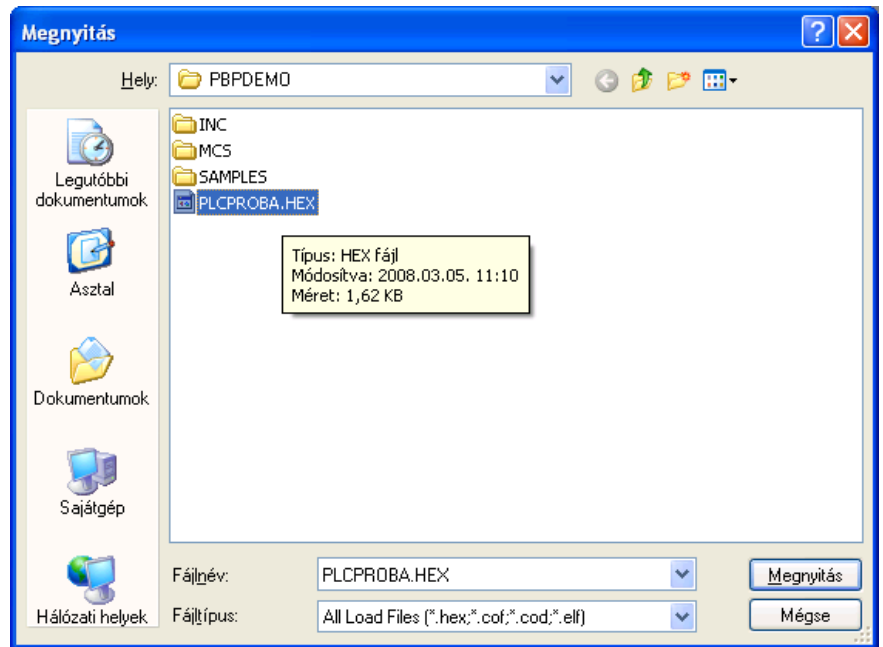
- Kattintsunk a  
"Connect"-re!



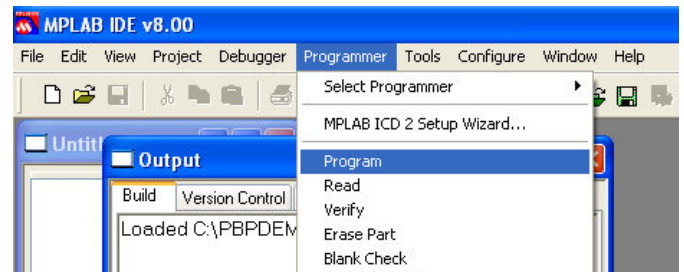
- Az ICD-nk meg is találja a 16F877-es mikrokontrollert:



- Most olvassuk be a már elkészült HEX fájlunkat, azaz "File" menü, "Import", és a PLCPROBA.HEX kijelölése:

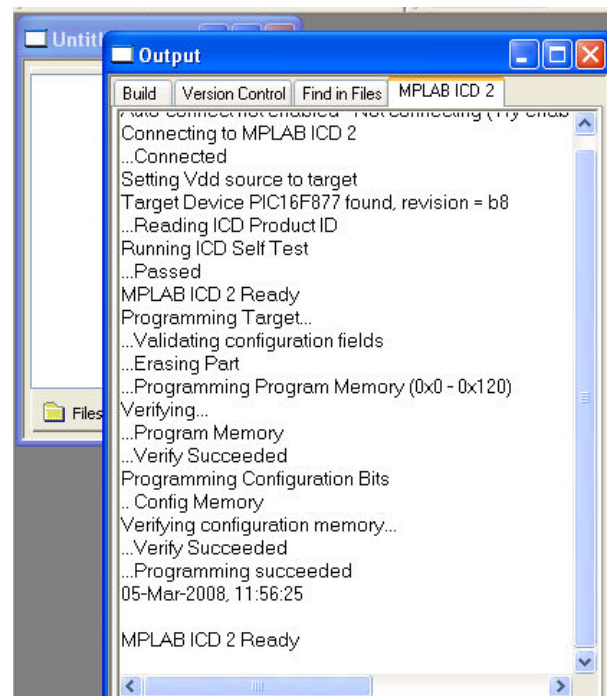


- Jöhet a programozás:



- A programozás sikeresen megtörtént, húzzuk le az ICD2 csatlakozót! (Az ICD2 "fogja" az RB6, RB7 portot, ami a 74HC4094-et is kezelné.)

A relék kattognak, a LED villog, az LCD-n megjelent a felirat. (Ugye?)





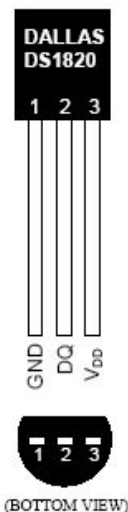
### A DS1820.BAS program:

Egy DS1820 hőszenzort olvasunk be, az eredmény az LCD-re, a relék a hőmérséklettől függően kapcsolgatva.

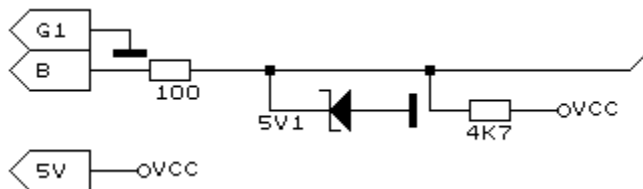
A DS1820/DS18S20 hőszenzorról, bekötéséről:

A DS1820 mindössze 3 kivezetéssel - ebből kettő a tápfeszültség - egy komplett hőmérő. A IC-be mindent beleintegráltak, - hőérzékelő, referenciafeszültség generátor, 9 bites A/D, stb. - ami egy 0.5 tized fokos pontosságú hőmérsékletméréshez szükséges. Az IC -55-től +125 C fokig tud mérni, és kifejezetten a mikrokontrolleres alkalmazásokhoz fejlesztették ki. (A teljes dokumentáció megtalálható a CD-n a MIKROKLUB/PDF/DALLAS könyvtárban.)

A PICPLCx mikrogéphez a DS1820-at egy 3 eres kábellel csatlakoztathatjuk. A negatív tápfeszültség kivezetést - GND - értelemszerűen egy G1 vagy G2, G3.. jelű, a pozitív tápot - Vdd - az 5V sorkapocsba kössük.



A DQ adatkimenetet - most - a B1 bemenetre. A kábel hossz akár több tíz méter is lehet. A kábel típus pár méterig lényegtelen, nagyobb távolságnál a sodrott kábelt ajánlják. A PICPLCx bemenetével soros ellenállás értéke 100 ohm, a felhúzóellenállás 4.7 Kohm legyen.



És akkor a program működéséről:

Először a szükséges konfigurációk. Az LCD a már szokásos beállítás szerint, a DS1820 adatvezetékét az RA0 porton várja a program.

A szenzornak egy parancsot kell küldeni, hogy kezdje meg a konverziót, majd pedig egy másik paranccsal tudjuk beolvasni az eredményt.

(A BASIC-nek az egy vezetékes DALLAS busz kényelmes kezelésére két spec. utasítása van. Adatot beolvasni a OWIN , küldeni a OWOUT utasítással tudunk.)

A beolvasott adatból kiszámoljuk a Celsius fokot, és kiírjuk az LCD-re.

Végül a relék vezérlése. A programban akkor kapcsol be - az egyszerűség kedvéért az összes - relé, ha a meleg eléri a 32 fokot, és akkor kapcsol ki, ha 30 fokra süllyed.

```

DS1820.BAS - Jegyzetömb
Fájl Szerkesztés Formátum Nézet Súgó

' Homerseklet mero, termosztat program PICPLC8 / PICPLC16 mikrogepre, DS1820 szenzorral

DQ      Var      PORTA.0      ' DS1820 adatlaba, a DQ, a PORTA.0 labon
ADCON1 = 7      ' A PORTA es a PORTE most digitalis portkent mukodjon

      DEFINE LCD_DREG      PORTD      ' Megadjuk az LCD vezzerlesi parametereit :
      DEFINE LCD_DBIT      0      ' Az LCD adatbitek a D porton,
      DEFINE LCD_RSREG      PORTD      ' es az első bit az RD0.
      DEFINE LCD_RSBIT      4      ' A register select bit a D porton,
      DEFINE LCD_ERE      PORTD      ' az RD4 port.
      DEFINE LCD_EBIT      5      ' Az enable bit a D porton,
      DEFINE LCD_BITS      4      ' az RD5 portra van kiosztva.
      DEFINE LCD_LINES      2      ' A 4 bites üzemmódot választjuk,
      ' es 2 soros az LCD

hofok      Var      Word      ' A "hofok" változonak egy 2 bájtos helyet foglalok
meres_ok      Var      Byte      ' Allapotjelzo a mereshez

main:      OWOut DQ, 1, [$CC, $44]      ' Felszolitom a DS1820-at, hogy
      ' csinaljon egy homerseklet merest!

meres:      OWIn DQ, 4, [meres_ok]      ' Varom hogy vegezzen... Amig a meres_ok = 0 , nincs kesz!
      If meres_ok = 0 Then meres      ' Csak ha a meres_ok mar nem nullat ad, mehetunk tova.
      OWOut DQ, 1, [$CC, $BE]      ' Most mar beolvashatom a hofokat.
      ' Felszolitom a 1820-at, hogy kuldje az adatokat!

      OWIn DQ, 0, [hofok.LOWBYTE, hofok.HIGHBYTE]      ' Az adatokat "hofok" változoba olvasom.

      ' At kell szamolnom az eredmenyt Celsius fokra:
      ' Egy bit, 0.5 foknak felel meg. (Pl. ha 51-et olvasok ki, az 25.5 C)
      hofok = hofok * 5      ' Ha szorzom 5-el a hofok erteket, akkor homerseklet ertek 10-szereset kapom
      ' Pl. ha 51 volt a hofok változoban, akkor a muvelet utan 255 lesz.
      Lcdout $fe, 1, "Szenzor: ", DEC (hofok / 10), ".", DEC1 hofok, " C"      ' Kirakom az eredmenyt az LCD-re

      Pause 1000      ' Egy masodperc szunet...

      ' Es akkor a relek vezzerlese, a homerseklet szerint:

```

### Epilógus:

A programok működésének teljes megértéséhez persze hosszan kell tanulmányoznunk a BASIC leírását, az LCD, a 74HC4094, és a DS1820 adatlapját - mindezek a CD-n megtalálhatóak - és persze a PIC mikrokontrollerek belső felépítését.

A „mikroklub cd”-n két kisebb BASIC program található a MIKROKLUB\PICBASIC könyvtárban. Az LCDTEST.BAS egy LCD tesztelő, LED villogtató program a PICDEMO panelre, a PLCTEST.BAS egy hasonló a PICPLC8, PICPLC16 áramkörre.

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>