

Digitális, analóg bemenetek, hőszenzorok olvasása, relék vezérlése, a PICPLC16, PICPLC8 áramkörrel, és IBM PC-n futó kezelő programmal

A következők azoknak a "programozóknak" szólnak, akik egy olyan, saját, PC alapú vezérlőprogramot szeretnének írni, amivel bemeneteket figyelhetnek, analóg méréseket végezhetnek, és külső eszközöket tudnak vezérelni. (Pl. egy PC központú folyamatvezérléshez, kazánszabályozáshoz kell egy be/kimeneti egység, vagy egy PC-ről vezérelt intelligens ház" programmal világítást, fűtést, redőny leeresztést, felhúzást kapcsolgatni, stb.)

Kimenetként 8-16 relé, és összesen 8 bemenet áll rendelkezésünkre. A bemenetek lehetnek analógok vagy digitálisak, vagy hőszenzor bemenetek, keverve is.

Tehát tudunk:

- 8 vagy 16 relét vezérelni (PICPLC8 vagy PICPLC16-on fut a program.)
- beolvasni 1-8 digitális bemenetet
- vagy 1-8 analóg bemenetet
- vagy 1-8 darab hőszenzort (DS1820)

A hardwer:

A programok a PICPLC8 és a PICPLC16 mikrokontrolleres vezérlőpanelre íródtak. Mivel ezek felépítése hasonló, a programok ugyanúgy futtathatóak mindkét mikrogépen. (Az áramkörök részletes leírása a PICPLC8.PDF és PICPLC16.PDF file-okban.) Amíg a PICPLC16 panelen 16, a PICPLC8-on csak 8 relé van. De az utóbbin is elérhető az a nyolc vezérlőjel, ami a nagyobb panelen 9-16 relé vezérlését végzi, így ezek kimenetek felhasználhatóak digitális kimeneti jelnek.

Az áramkörben két, sorba kötött 74HC4094-es léptetőregiszter kimeneteire kötött ULN2803 kapcsolja az áramot a relé meghúzó tekercsekre. A mikrokontrollerbe égetett program tehát fogadja a PC-ről küldött adatokat, és belépteti azokat a reléket vezérlő léptetőregiszterekbe, majd a 4094-ek STR jelével érvényesíti azokat. (Tehát a 16 kimeneti vezérlőjel egy pillanatban jelenik meg.)

A B1-B8 bemenet a mikrokontroller A0-7 portjára csatlakozik. A mikrokontroller ezeket a kivezetéseit tudja digitális, vagy analóg bemenetként kezelni, de akár kimenetként is.

Honnan fogja a program tudni, hogy a B1-8 bemenetet mint analóg, vagy digitális bemenetként kezelje, vagy épp egy DS1820 szenzort próbáljon beolvasni? A PC-ről érkező parancs "A" vagy "D", vagy épp "?" karakteréből, ami meghatározza, hogy az adott bemenetet hogyan kezelje.

A RELE16DS program:

A program PIC 16F871 mikrokontrollerre készült, és 4 megás kvarcot igényel. A PIC-be égetendő tartalom a RELE16DS.HEX .

A program fogadja a PC soros portjáról küldött vezérlőjelet, és azok tartalmától függően kapcsolgatja a reléket, beolvassa, és beküldi a 8 bemenetet állapotát. A bemenetek lehetnek digitális, vagy 0-5 voltos analóg bemenetek, de köthetünk a bemenetekre DS1820 hőszenzort is. A mikrokontroller a PC-ről jövő parancsok megfelelően olvassa be az adott bemenetet, vagy az oda kötött hőszenzort. A vezérlőjelet egy ellenőrző kód egészíti ki.

A program futását egy led villogása jelzi. (A PICPLC8-nál az L9, a PICPLC16-nál a L19)

A relék kapcsolgatása:

A PICPLCx mikrokontrollerébe égetett program egy 16 bites adatot vár a soros portján, amit a vétel után kirak a reléket vezérlő 74HC4094-ekbe. (Pl. ha a 1034 hexadecimális kódot küldjük - binárisan leírva ez 0001000000110100 - akkor a 3., az 5., a 6., és a 13. relé lesz bekapcsolva.) A két bájtot 2400 baudon küldjük, 8 adat bit, 1 stop bit, paritás vizsgálat nincs.

A működés biztonságát a program azonosító, és ellenőrző kóddal biztosítja. Tehát csak akkor történjen relé kapcsolás, ha a vezérlő adatok egy azonosító bájttal kezdődnek, és a PC programban, valamint a PICPLCx-ben képzett ellenőrző kód megegyezik.

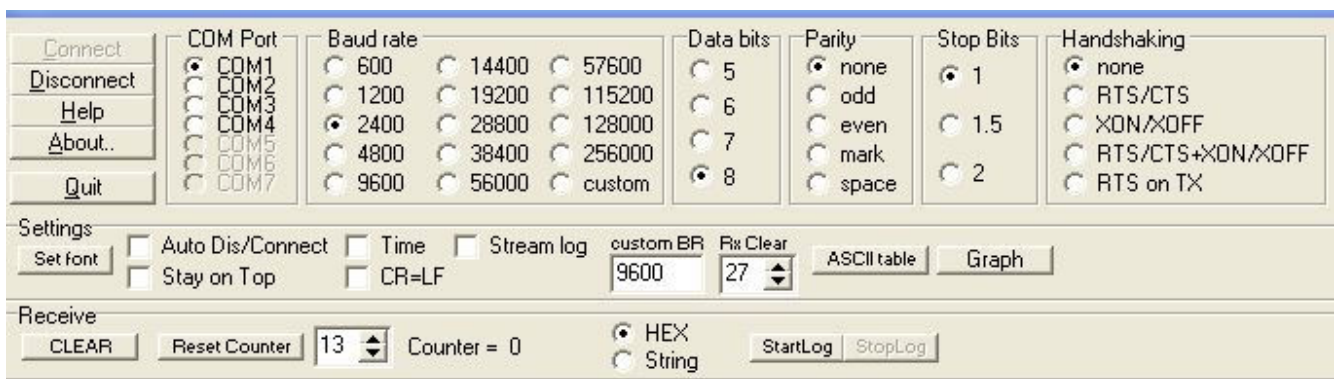
A lényeg, hogy 4byte-ot kell küldeni, az alábbiak szerint:

1. bájtt: 21 (hexa) ami egyébként a "!" kódja, ez egy azonosító byte
2. bájtt: a relék felső (H) byte-ja
3. bájtt: a relék alsó (L) byte-ja
4. bájtt: a 2. és 3. byte összege 1byte-on

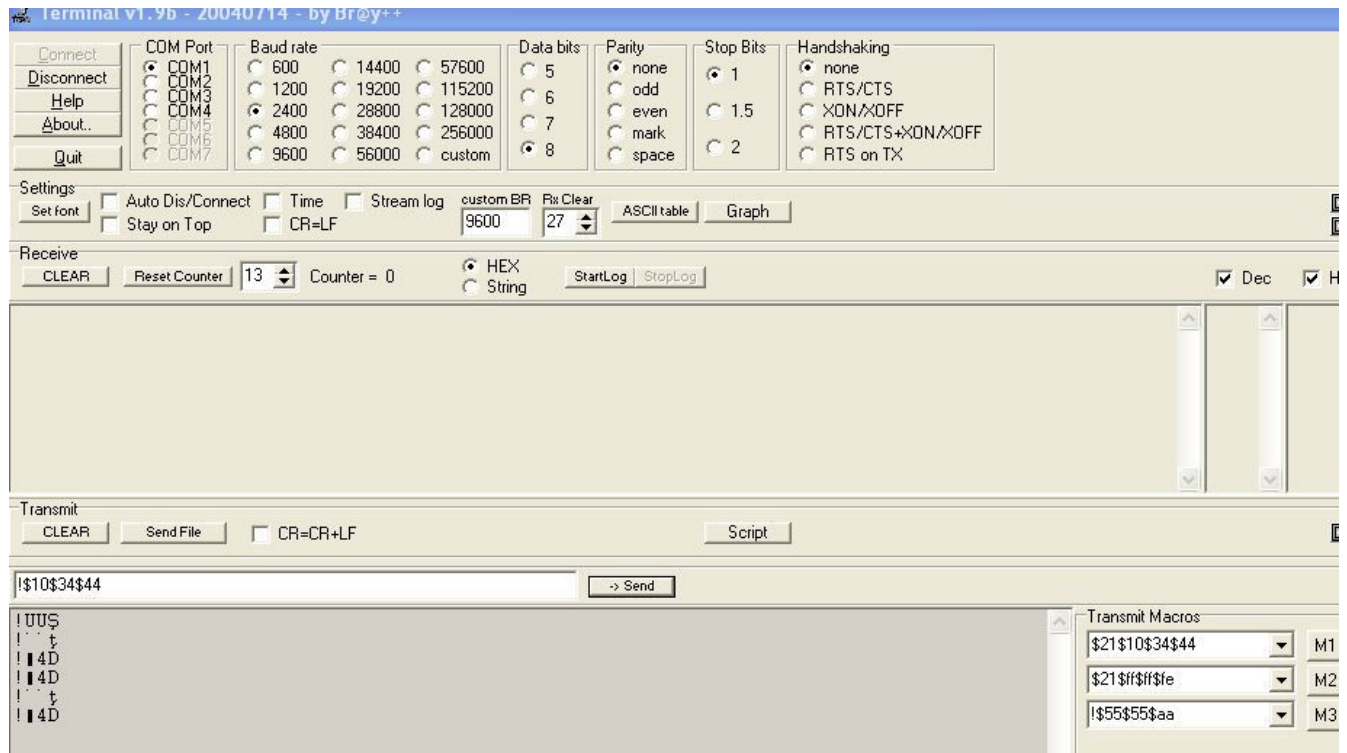
Próbaképpen tehát kapcsoljuk be a 3., az 5., a 6., és a 13. relét. Az előbbiek alapján ez a 10 és 34 hexadecimális vezérlőparancs értékeket jelenti, kiegészítve az indító kóddal, és az ellenőrző kóddal, tehát most az 21 , 10 , 34 , 44 hexadecimális kódokat kell kiküldeni.

Hogy néz ki ez a gyakorlatban ? A működés egy terminál programmal lepróbálható. (Egy nagyon jó terminál program - ami az alábbi példákban is szerepel - rajt van a „mikroklub” CD-n, illetve letölthető a <http://bray.velenje.cx/avr/terminal/> címről.)

A teszthez állítsuk be a soros portot, és az előbb leírt adatátviteli paramétereket - 2400 baud, 8 adatbit, 1 stop bit, paritás vizsgálat nincs - és a vezérlő adatokat küldjük ki a mikroszámítógépbe!



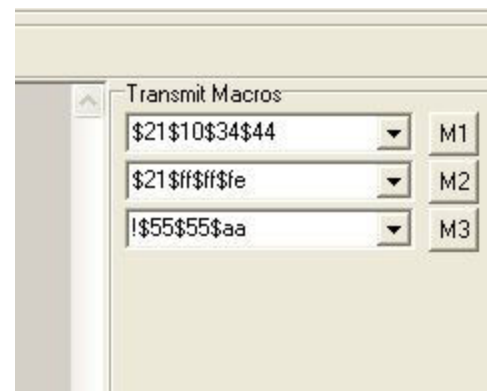
A terminálprogramban gépeljük be a \$21\$10\$34\$44 adatokat, vagy az ezzel egyenértékű !\$10\$34\$44 kódsorozatot. (A \$ jelzi hogy hexadecimális adatokról van szó.) Eztán kattintsunk a Connect , majd a Send gombokra:



Ha minden rendben, bekapcsolnak a kiválasztott kimenetek.

Ha nem, ellenőrizzük le a beállításokat. (Soros port, baud, stb.) Ha a soros porton adat érkezik, akkor az L17 LED felvillan. Ha nem, akkor ellenőrizzük le a soros kábel csatlakozását, valamint hogy a megfelelő portot használjuk-e.

Ugyanígy, az 21h, 55h, 55h, AAh érték minden második - 1. 3. 5. 7. .. 15. - az 21h, FFh, FFh, FEh pedig az összes kimenetet magas szintre kapcsolja.

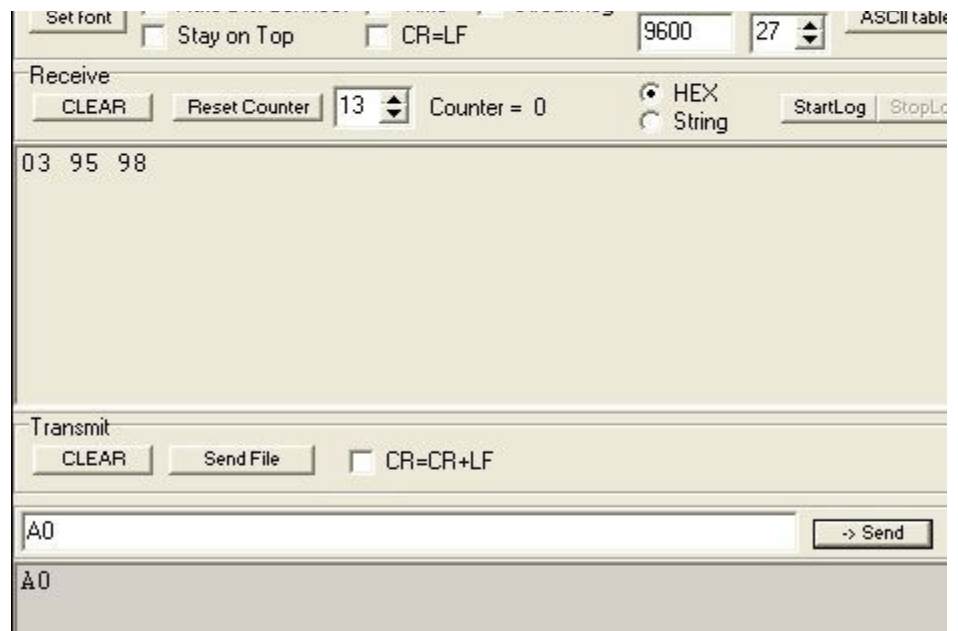


Az A/D-k bekérése:

Két bájtot kell küldünk, az első byte az "A" karakter, majd a csatorna azonosító, 0-7. A mikrokontroller most analóg bemenetként olvassa be a kiválasztott bemenetet, és az eredményt . (A PICPLC8/16 B1-8 bemenete a mikrokontroller A0-7 A/D bemenetére csatlakozik.)

Tehát 2 byte-ot kell a PC-ről küldeni - Axx - és válaszként a bemeneten mért feszültség digitális eredményét kapjuk:

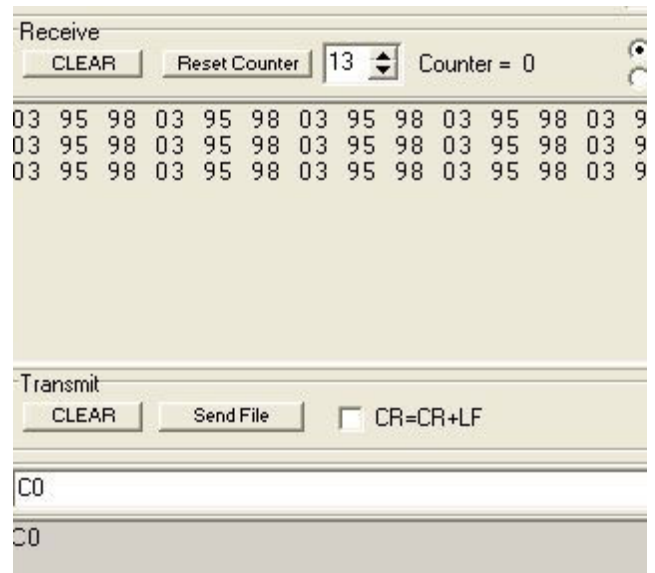
1. bájt: a A/D felső (H) byte-ja
2. bájt: a A/D alsó (L) byte-ja
3. bájt: a 2. és 3. byte összege lbyte-on



Ha egy analóg csatornát folyamatosan akarunk beolvasni, erre is lehetőség van, a C0-C8 a parancsal. (C = continous)

Tehát a mikrogép C0..C7 parancsra 1 másodpercenként küldi az adatot.

Kilépni a folyamatos küldésből úgy lehet, hogy valamelyik másik parancsot küldünk neki, pl. A0.



A B1-8 bemenet mint digitális bemenetek bekérése:

Ha a "DI" parancs megy ki a PC-ről, akkor mint digitális bemenet van értelmezve a B1-8 sorkapocs, és válaszul 3 bájtot kapunk vissza. Az első bájt a B1-8 bemenet állapota. Mivel most ezek digitális bemenetként vannak értelmezve, 0 vagy 1-es értékű bit jelenti az adott bemenet állapotát.

A második bájt - 8 bit - a mikrokontroller RC0-7 portjának állapota. (Ezek nincsenek kivezelve sorkapocsra, a tűkesorról érhetőek el.)

A harmadik bájt az ellenőrző bájt, ami az első két bájt összege.

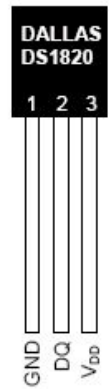
Tehát ha "DI" a soros porton küldött parancs, akkor válaszul ezt kapjuk:

1. bájt: az RA0-5, RE0-2 egy byte-ban
2. bájt: az RC0-7 egy byte-ban
3. bájt: a 2. és 3. byte összege 1byte-on

(RC-nél csak 0..5-ig küldi be, mert a 6-7. bit az rxd,txd. Ezeknél 0-t kapunk.)

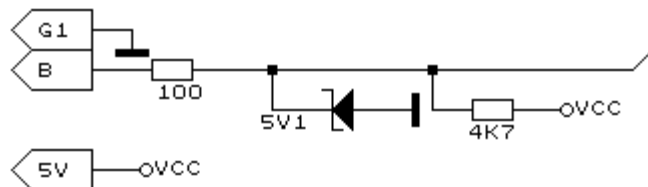
A hőszenzor beolvasás:

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 DS1820 bekötése:

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

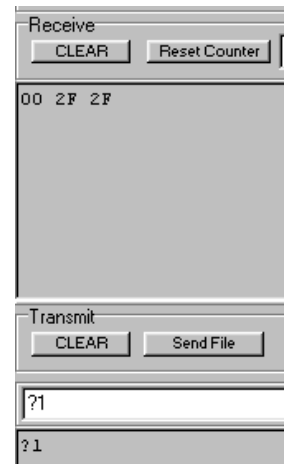


adatkimenetet a B1, vagy B2...B8 bemenetre, ahova akarjuk. A kábel hossz akár több tíz méter is lehet. A kábel típus pár méterig lényegtelen, nagyobb távolságnál a sodrott kábelt ajánlják. A PICPLCx bemenetével soros ellenállás értéke 100 ohm, a felhúzóellenállás 4.7 Kohm legyen.

Lekérdezéskor "?" karaktert, majd a bemenet számát kell kiküldeni. (Pl.: ?5 kóddal kérdezhető le az ötödik bemenetre kötött hőmérő adata.) Erre a mikro válaszként 3 byte-ot küld vissza:

1. bájt: előjel, ha 0 , akkor plusz, ha 1 akkor mínusz fok.
2. bájt: a hőmérséklet értéke 0.5 Celsius fokos egységben
3. bájt: egy ellenőrző összeg, az 1. és 2. bájt összege 1 byte-on

Egy példa: Mondjuk hogy a szenzor a B1 bemeneten van. A terminál programmal ?1 kódot küldünk, a mikrogep a 00, 2F, 2F hexadecimális értékeket küldi vissza. Az első bájt nulla, tehát a fagypont felett vagyunk, a második érték 2Fh ami ugye decimálisan 47, 47×0.5 fok pedig adja az eredményt, ami most 23.5 Celsius fok.



Ha az előjel 1 (mínusz fok), akkor a 2. bájtban a hőmérséklet kettes komplementese van. (pl. 1.bájt=1, 2.bájt = FFh = 255 = -0.5C) Egy táblázat erről, az IC dokumentációjából:

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+85.0°C*	0000 0000 1010 1010	00AAh
+25.0°C	0000 0000 0011 0010	0032h
+0.5°C	0000 0000 0000 0001	0001h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1111	FFFFh
-25.0°C	1111 1111 1100 1110	FFCEh
-55.0°C	1111 1111 1001 0010	FF92h

Ha nincs, vagy nem olvasható a szenzor, akkor a program 2. és 3. bájton FFh értéket küld.

Az LCD/nyomógomb panel lekezelése:

Ha csatlakoztatjuk az LCD/nyomógomb panelt, akkor azt is tudjuk a PC-ről vezérelni, illetve a gombok nyomását érzékelni.

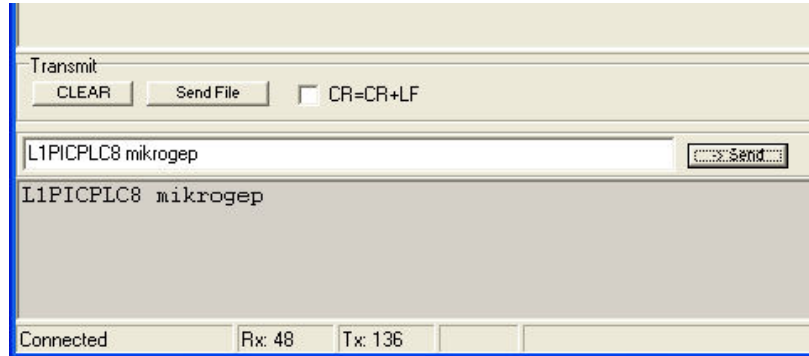
Ha az

Lxyyyyyyyyyyyyyyyyyy

parancsot küldjük ki a soros porton, ahol az y a kiküldendő maximum 16 karakter, az pedig 1 vagy 2 lehet, attól függően, hogy az LCD első, vagy a második sorába akarjuk kiírni.

Pl. az L1PICPLC8 mikrogep parancs az LCD első sorába a PICPLC8 mikrogep szöveget fogja kiírni.

Ha lenyomunk egy nyomógombot akkor a Gx - ahol az x=1, 2, 3 vagy 4 - kódokat küldi be a mikro, attól függően, hogy melyik gombot nyomtuk le a négy közül.



Kapcsolódó dokumentációk:

A PICPLC16, és a PICPLC8 áramkörrel egy-egy külön leírás szól. (PICPLC8.PDF, és PICPLC16.PDF)

A PIC-ek assembly programozásának témájához kapcsolódik a PICDEMO panel dokumentációja, valamint a MICROCHIP PIC ICD leírása. (PICDEMO.PDF, ICD.PDF)

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

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

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