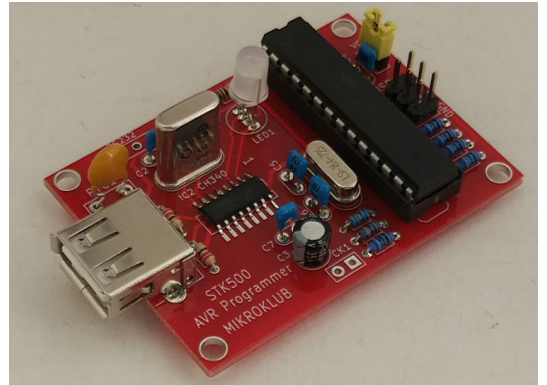


Az STK500 ATMEL mikrokontroller égető

Az újabb kiadású mikrokontrollerek többsége tartalmazza a soros programozás lehetőségét. A lényeg, hogy a betöltendő adatokat sorban, „bitenként” léptetjük be a mikrokontrollerbe. Mivel ez az algoritmus csak pár IC kivezetést igényel, lehetőséget ad arra, hogy a mikrokontrollert az áramkörben - in circuit - programozzuk. Ez nagyon előnyös, pl. programfejlesztésnél, hiszen nem kell a mikrokontrollert minden egyes program verzió kipróbálásához kiemelni a foglalatából az égetéshez. (A felületre szerelt - smd - IC-eket pedig nem is lehet.)

Mivel rengeteg áramkör lelke egy ATMEL mikrokontroller, ezért aztán egy csomó égető is készült hozzájuk.

A párhuzamos, soros portos variációk felett már eljárt az idő. Legyen USB-s! De melyik? Egy olyan égetőt akartam, ami működik az ARDUNIO-val, de az ATMEL7 fejlesztő programmal is használható. Az STK500 programozó tudja ezt.



Ez a programozó eredetileg az STK500-as fejlesztőpanel része volt, onnan lett kiemelve, hiszen önállóan is használható.

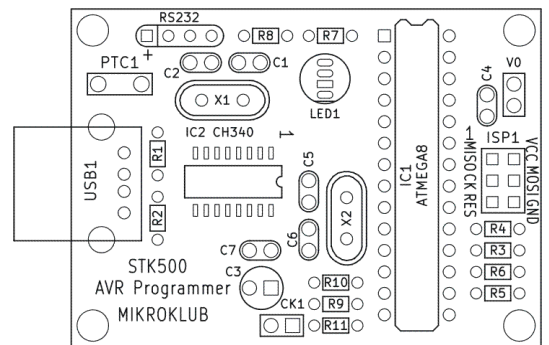
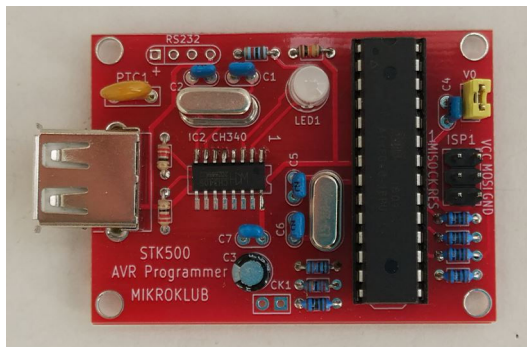
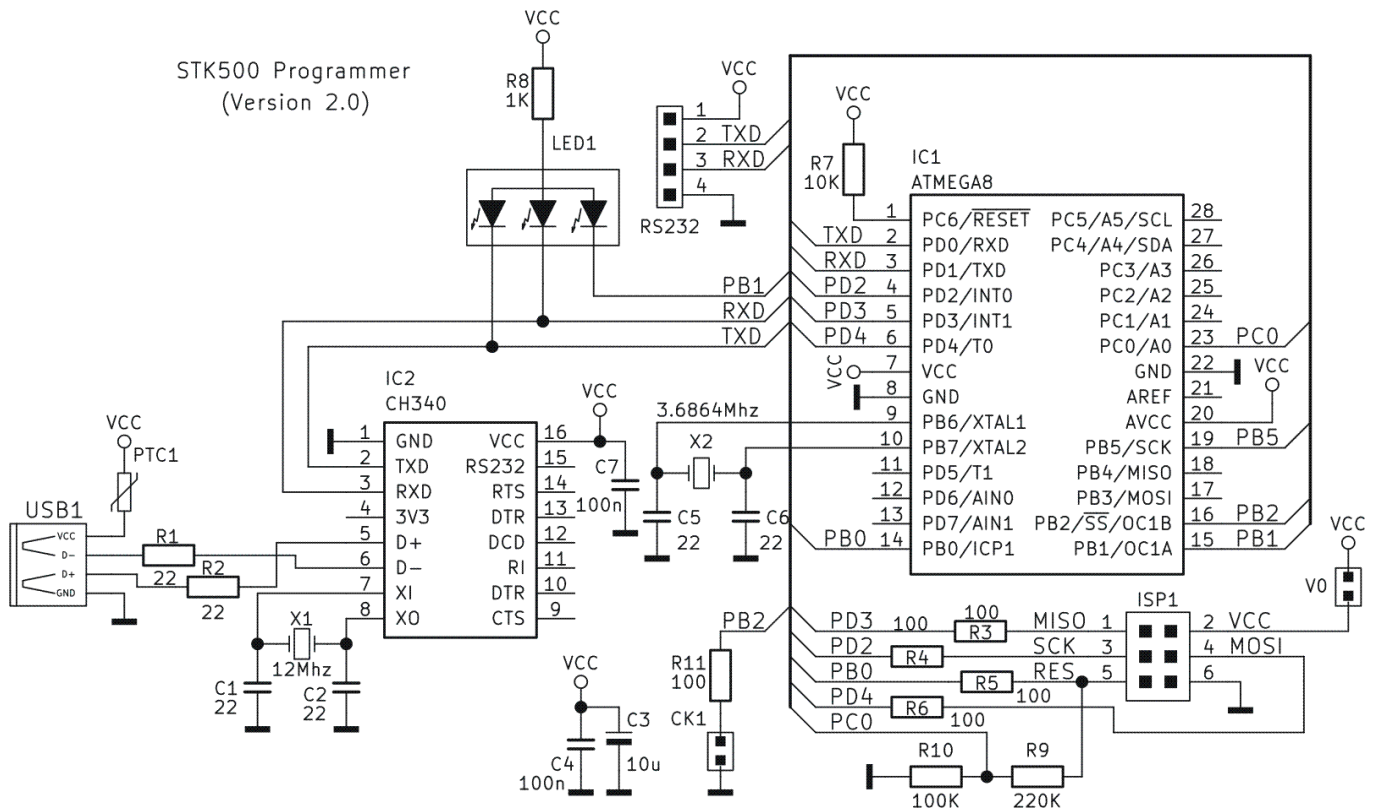
Mivel a "hivatalos" fejlesztőprogram, az ATMEL7 stúdió is kezeli, így amelyik AVR típus tudja az ISP módot - a legtöbb - az programozható vele. Található rajt egy kb. 1MHz-es órajelkimenet is, ha esetleg kizárnánk magunkat az AVR-ből, egy rossz órajel forrás beállítással.

Az „ős” STK500-as még az RS232 porton működött, ami már egyre kevesebb PC-n található meg. Hogy USB-n is tudjunk csatlakozni a számítógéphez, egy CH341-es USB/RS232 csatoló került a panelra.

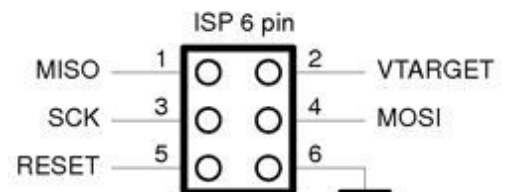
Azért választottam a CH340-et, mert olcsó, a kivezetései nem hajszálvékonyak, és van hozzá driver Windows XP, WIN7, és WIN10 alá is.

Az áramkör:

A kapcsolási, és beültetési rajz:



A programozó csatlakozó kiosztása:



Egy jumper is van a panelen.

A VO jumper a programozó csatlakozó 2. kivezetésére adja ki - vagy veszi le - a plusz 5V tápot.

Az USB driver telepítése:

Windows7, WIN8, WIN10 :

A netre csatlakozott win7, WIN10 rendszeren roppant egyszerűen zajlott a telepítés. Az adaptert rádugva egy USB aljzatra, a windows „új hardvert talált”, és pár perces – legalább 3-4 percre számítsunk – ügymködés után megjelent a megnyugtató üzenet.

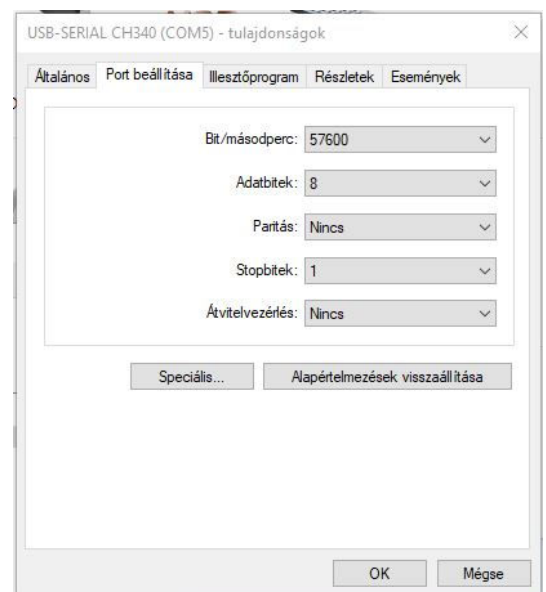
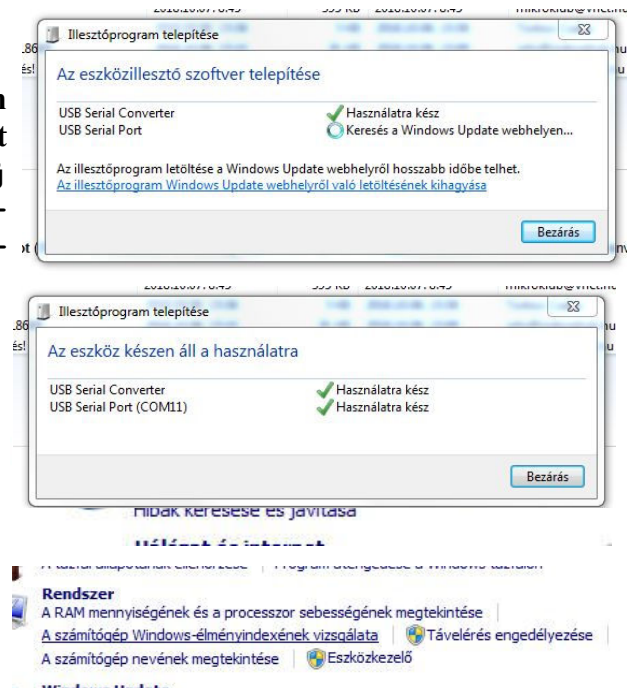
Az eszközközelőben meg is tekinthetjük az új soros portunk tulajdonságait.

Kattintás a „Rendszer és biztonság”-ra

Aztán az „eszközközelőre”

A tulajdonságok-ban állítsuk át a baud sebességet – az alapbeállítás 9600-ról – legalább 57600-ra!

(Hogy a nagyobb sebességű adatforgalom se legyen akadály.)



Ha a windows nem telepíti magától a CH34x_Install_Windows_v3_4 EXE meghajtó programot, futassuk a CD-n található telepítő programot:

Az ATMEL Studio

A programozó hardverhez több működtető program is készült. De a legtöbb típust a „gyári”, azaz az ATMEL által készített ATMEL Stúdió biztosítja.

- Először installáljuk a működtető programot! Ez egy 20-40 perces folyamat lesz, szóval néha rá kell kattintani, de közben nézhetünk TV-t, nevelhetünk gyereket, stb.)



Valamint - ha nincs fenn a gépen, felrakja magának a „Microsoft NET Framework”, és a „Visual Studio” keret programot is ezek windows kiegészítők. (Ez önmagában 20 percig tartott...)

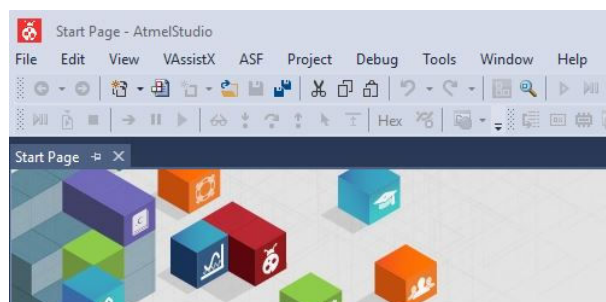
És akkor egy konkrét példán keresztül bemutatom a használatot. Legyen a feladat, egy ATMEGA328P mikrokontroller kiolvasása, beégetése.

Csatlakoztassuk az IC-t az égetőhöz, az égetőt a PC USB portjához!

Nézzük meg, melyik portra rakta magát a CH340 USB/RS232 illesztő!

 USB-SERIAL CH340 (COM5)

Esetünkben ez a COM5 :

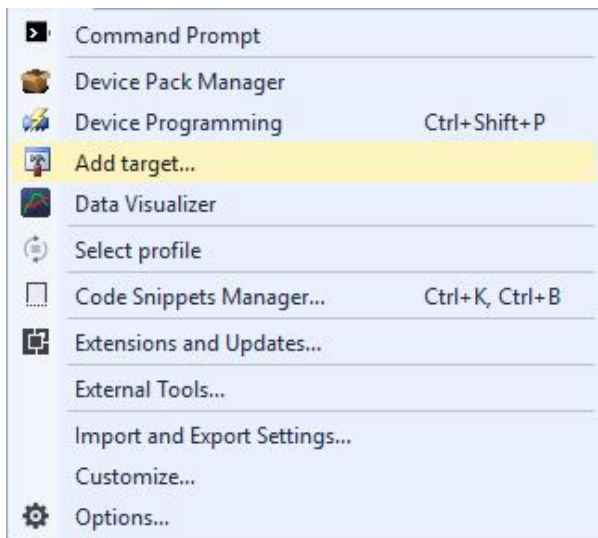
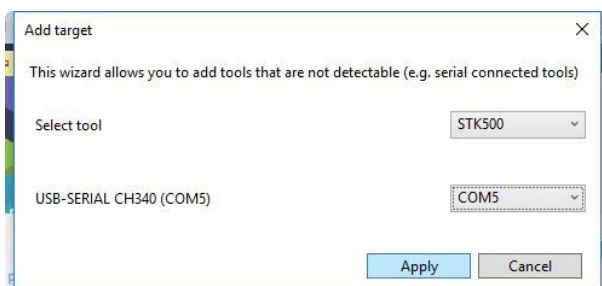


És akkor futassuk a programot:

Most hozzá kell adnunk az STK500-as égetőnket a fejlesztő programhoz.

Lépünk be a Tools menübe, azon belül katt az „Add target”-re.

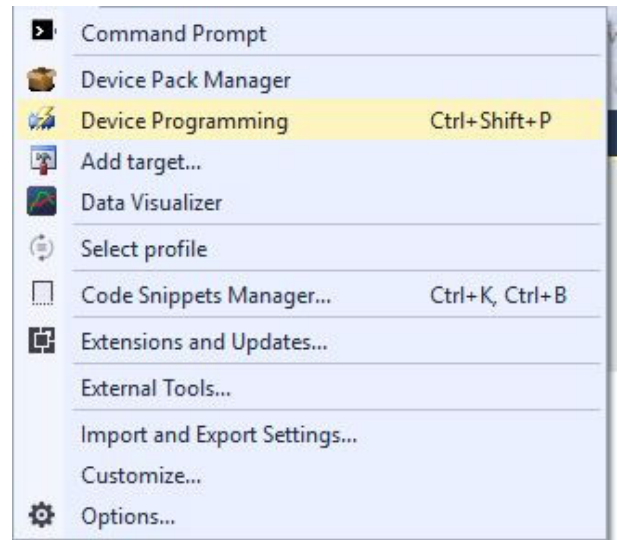
Válasszuk ki az STK500-at! (Nálam ugye ez most a COM5-ön van.)



Végül „Apply”.

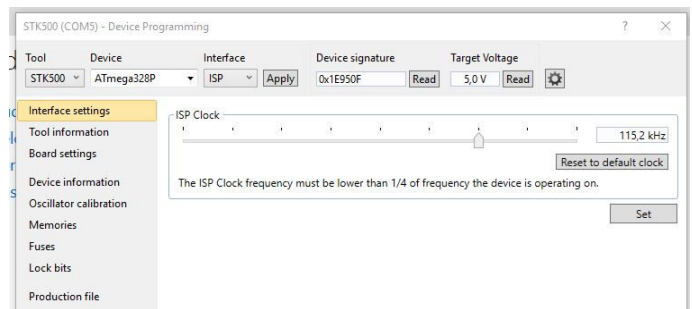
Ha eddig eljutottunk, akkor már van használhatjuk is a programozónk.

Csatlakoztassuk az IC-re, és most már rákattinthatunk a „Device Programming”-ra.



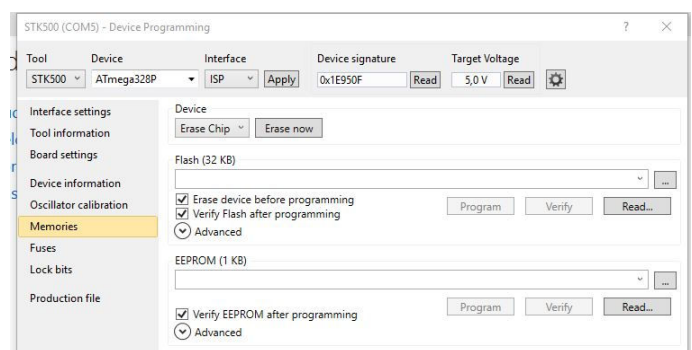
Most akkor állítsuk be, amit be kell. Tehát a „Tool” az STK500, a „Device” az ATmega328P, és ISP.

Próbaképp olvassuk be az IC azonosítóját. (Device signature Read)



Ha az IC azonosító kiolvasás sikeres volt, akkor valószínűleg a programmemória, fuses írás/olvasás is rendben lesz.

Ha hibaüzenetet kapunk, akkor nézzük át a csatlakozást, kap-e az IC tápot, stb. Ha áramkörön belül programozzuk az IC-t, akkor nézzük át, nem csatlakozik-e valamilyen alkatrész, ami megakadályozza a kivezetések vezérlését.

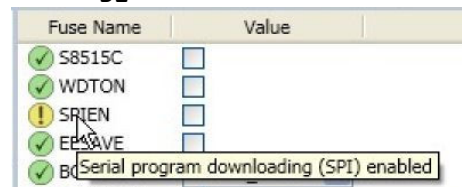
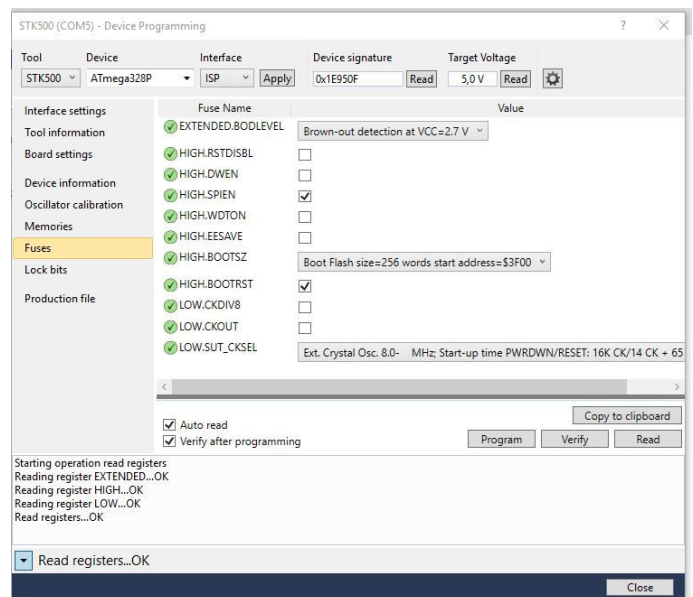


A „Fuse”-ok:

A Fuse - biztosíték - bitek alapvetően határozzák meg a mikrokontroller működését. Ha nem jól állítjuk be pl. a „watchdog timer” - WDTON - vagy az oszcillátor típust - SUT_CKSEL - akkor a beégetett programunk el se indul, vagy épp század másodpercenként „reseteli” a watchdog. Ezért is érthetetlen, hogy az ATMEL miért nem tette lehetővé, hogy a program kóddal együtt ezeket is bele lehessen írni a „HEX” fájlba. (Pl. a microchip PIC-eknél a programkód beolvasásával a fuse bitek is beállnak.)

Hát itt nem. Egyenként kell pipálgatni őket. Persze előtte böngészni a leírást, hogy mit, hogyan.

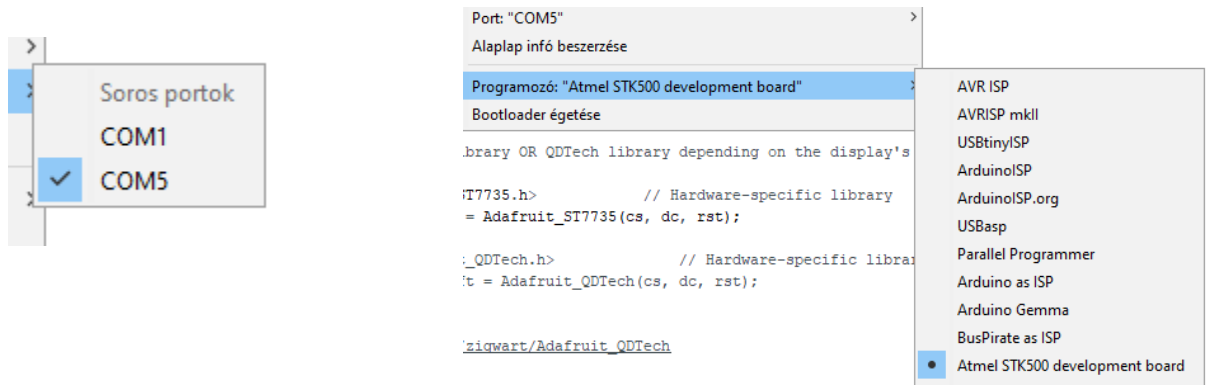
Tovább bonyolítja dolgot, hogy akár ki is tudjuk magunkat csukni a későbbi programozásból, az SPIEN fuse bit átállításával. Ez a „fuse” a soros programozási mód engedélyezése/tiltása. Ha kivesszük a pipát, és így égetjük be a biztikit, akkor többet már nem tudunk variálni, hiszen tiltottuk a soros programozást, és ezért ugye ezt a bitet se tudjuk átírni többé.... (Nem értem mi értelme van ennek, a program kiolvasást a titkosító fuse-ok bebillentésével is megakadályozhatjuk.) Az azért rendes az ATMEL-től, hogy egy figyelmeztető felirattal figyelmeztet erre:



Arduino:

Ez a programozó használható az „Arduino” programból is.

Jelöljük ki hogy melyik soros porton van az égető, és persze hogy az „Atmel STK500 development board” a programozó eszköz.



A 89Cxx lábkompatibilis, 40 lábú IC-k kezelése az ISP csatlakozóról:

Programmer: IC

1. (MISO) - 7.
2. (VCC) - 40.
3. (SCK) - 8.
4. (MOSI) - 6.
5. (RESET) - 9.
6. (GND) - 20.

A 18 - 19 lábra egy kvarcot kell kötni!

A 20 lábú IC-k kezelése:

A programozó, és a programozandó DIL 20-as tokozású IC összekötése:

Programmer: IC

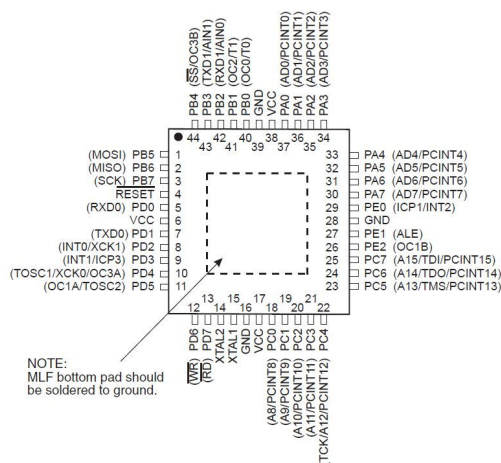
1. (MISO) - 18.
2. (VCC) - 20.
3. (SCK) - 19.
4. (MOSI) - 17.
5. (RESET) - 1.
6. (GND) - 10.

A 4. - 5. lábra egy kvarcot kell kötni!

ISP - TQFP44 IC

1. (MISO) - 2.
2. (VCC) - 6.
3. (SCK) - 3.
4. (MOSI) - 1.
5. (RESET) - 4.
6. (GND) - 16, 28, 39.

Kvarc: 14. - 15.



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.eprom.hu>,
<http://www.mikroklub.hu>