

## A TOP2049 univerzális IC programozó

Egy IC programozó készülék, egyenesen a Kínai Népi Demokratikus Köztársaságból. Nézzük mit tud? Sokat, nagyon sokat...

- több mint 2500 támogatott IC típus. EPROM, EEPROM, mikrokontroller, valamint GAL és PALCE IC-eket is programozhatunk vele.

- 2.5 - 6.5 voltos IC-k támogatása

- USB-s csatlakozás, nagyon gyors, 12Mhz-es adatforgalommal. A tápfeszültséget is az USB adja, de lehetőség van külső táp csatlakoztatására is.

- IC csatlakozás vizsgálat. Az égető teszteli a behelyezett IC kivezetéseit, ha úgy találja, hogy érintkezési hiba van, ezt jelzi. Pl. a „sok lábú”, smd tokos TSOP, PSOP adapterrel égethető IC-knél ez nagyon hasznos.

- A 48 lábú, karos IC foglalatnak köszönhetően, az összes támogatott, DIP tokos IC adapter nélkül égethető. (azaz nem kell külön adapter pl. a mikrokontrollerekhez, a 16 bites epromokhoz, stb.)

- Windows 98/2000/XP alatt működtethető.

- Formatervezett, kicsi, strapabíró műanyag doboz.

- Képes a gyártmány kód kiolvasására.

- A 74xxx, 40xxx, 45xxx TTL és CMOS IC-k, valamint RAM-ok tesztelésére is alkalmas.

- Az égetőt egy mikrokontroller vezérli. Ennek talán a legnagyobb előnye az, hogy stabilak, a windows-tól függetlenek a programozási folyamatot meghatározó időzítések.

### A szoftver, és az USB driver telepítése:

- Először is: még a programozó csatlakoztatása előtt installáljuk a működtető programot! (A mellékelt CD-n a SETUP.EXE)

Mondanom se kell, hogy az angol nyelvet válasszuk. A szokásos program telepítés után újra kell indítani a gépet a helyes működéshez.

- A PC program installálása után, de még a program futtatása előtt csatlakoztassuk az égetőt, egy szabad USB porthoz. Ha USB elosztón - HUB-on - keresztül csatlakoztatjuk a PC-hez az égetőt, akkor szükség lehet egy külső tápegységre. (5V DC)

Szóval a legjobb, a tápellátás szempontjából is, ha egy közvetlen - nem HUB-os - USB aljzatba csatlakoztatjuk az égetőt.

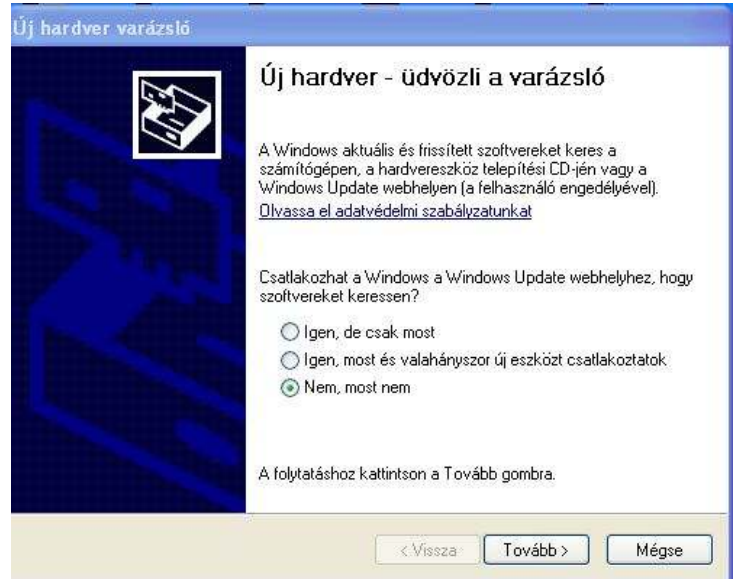


- A windows új hardver-t fog találni:



- „Nem, most nem” kell hogy a windows egy webhelyhez csatlakozzon:

- Aztán telepítés „beavatkozás nélkül”



A végén az eszközvezérlőben meg kell találnunk a programozónkat:



### Fontos dolgok a használatához:

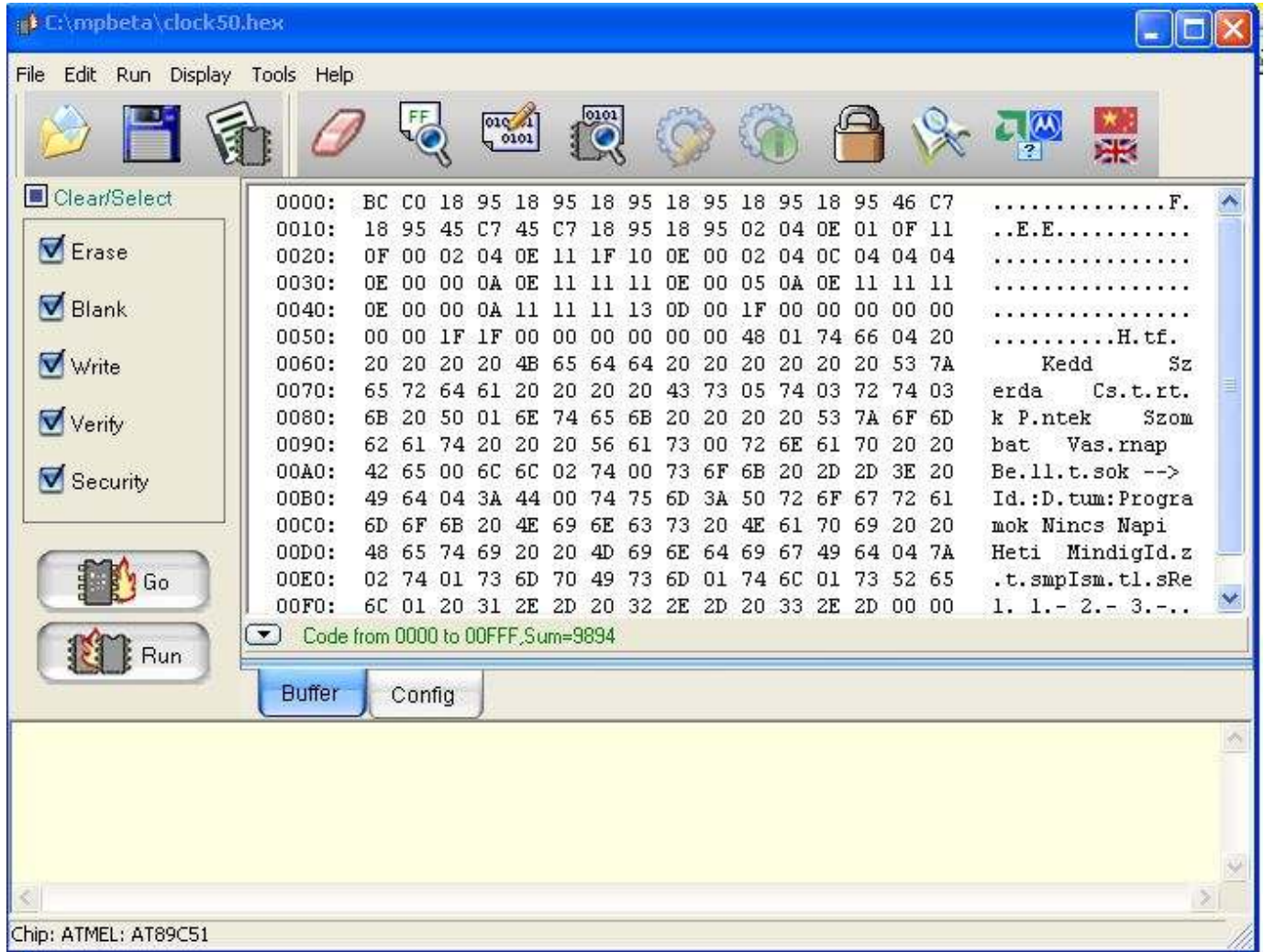
Ne csatlakoztassuk, vagy húzzuk le az égetőt a PC-ről, ha külső táppal használjuk. (Ha külső tápot akarunk használni, előbb csatlakoztassuk az USB-re, dugjuk be a tápcsatlakozót, és csak ezután dugjuk a dugasztápot a konnektorba.)

A bekapcsolás alatt ne legyen IC a foglalatban, illetve ne hagyjunk IC-t a programozó foglalatban, ha kilépünk a programból! Hogy miért is olyan fontos ez? Az égető figyeli a programozott IC tápáram felvételét, ha ez túl magas, beavatkozik. De ha a PC bekapcsolásakor már benn van egy IC a foglalatban, az égető pedig az USB-n keresztül tápot kap, akkor a bekapcsolási folyamatban ellenőrizetlenül juthat ki valamilyen feszültség a foglalatra - az égető honnan is tudná, milyen IC van a foglalatban, ráadásul a vezérlő mikrokontroller, elektronika, feléledéséhez is kell egy kis idő - ami szerencsétlen esetben a foglalatban hagyott IC, de akár az égető meghibásodását is okozhatja!

### A PC program használata:

Az égetőt és a PC-t kössük össze az USB kábellet! A bekapcsolás alatt ne legyen IC a foglalatban!

A Topwin.EXE PC program kezeli, automatikusan felismeri, és ellenőrzi a csatlakoztatott égető hardvert.



### A képernyő felosztás:

A felső ikon sor elemeire kattintva indíthatók a leggyakrabban használt műveletek.



Középen az IC-be irandó, vagy az onnan beolvasott adatok, a „buffer” memória.

Bal oldalt a programozáskor használatos műveletekből állíthatunk össze egy sorozatot. (Pl. IC törlés, törlés ellenőrzés, programozás, stb.)

Alul pedig az „üzenőfal”, a programozási folyamatok, üzenetek láthatóak itt.

### A program "ikonok" és funkciói:



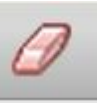
"Load fájl to buffer" : file beolvasás a bufferbe, azaz annak a munkaterületnek, memóriatartománynak a feltöltése adatokkal, ami majd az IC-be kerül.



"Save buffer to a file" : a buffer mentése egy fájlba. Azaz pl. egy IC-ből kiolvasott, és a memóriába - amit itt buffernek nevezünk - elhelyezett adatokat menthetjük el.



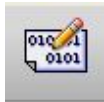
"Select chip" : IC típus választás.



"Erase chip" : az elektromosan törölhető IC-knél adható ki ez az utasítás, és törli a chipet.



"Check blank chip" : üresség teszt, azaz töröltség ellenőrzés. (Első sorban az UV fénnel törölhető IC-knél van jelentősége.)



"Write code to chip" : a bufferben lévő adatok beégetése - beprogramozása, írása - a chipbe.



"Compare chip with the fájl buffer" : az IC-ből kiolvasott adatok összehasonlítása a buffer adatokkal.



"Write Config to chip" : csak azoknál az IC típusoknál van értelme - aktív - ahol működést beállító biteket is lehet programozni. (Pl. Microchip, vagy ATMEEL mikrokontrollerek.)



"Read Config from chip" : a „beállítás kiolvasás a csipből” azoknál az IC típusoknál aktív, ahol vannak működést beállító bitek. (Pl. Microchip, vagy ATMEEL mikrokontrollerek.)



"Lock chip" : titkosítás, lezárás. Vannak IC típusok, amelyeknél tiltható kiolvasás, vagy a programozás. Az ezt beállító „titkosító” biteket programozhatjuk be.



"Read chip to buffer" : az IC tartalmának kiolvasása. (És a „bufferbe” helyezése.)



"Display MFA and type" : elektronikus azonosító kiolvasása a chipből. (Nem minden mikrokontroller, vagy memória rendelkezik kiolvasható azonosítóval.)

De ugyanezek a parancsok kiadhatóak műveletsorként is.

A bal oldalon van egy parancs mező, ahol kiválaszthatjuk a programozás során végrehajtandó művelet sort.

Az „Erase” a törlést, a „Blank” a törlés ellenőrzést, a „Write” a programozást, vagy mondhatjuk úgy is, az írást, a „Verify” pedig a visszaellenőrzést jelenti.

Ha pl. egy 29F010 flash EPROM-ot akarunk törölni, majd égetni, végül ellenőrizni az égetést, akkor a jobb oldali ablakban jelöljük ki ezeket a műveleteket, majd kattintsunk az „Go” ikonra:



Ha sok IC-t kell azonos tartalommal beprogramoznunk, akkor lehet hasznos a Run parancs.

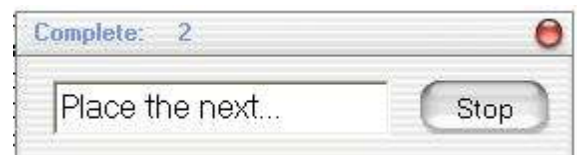


Ekkor kapunk egy parancsot, hogy vegyük ki az IC-t a foglalatból:



Rakjuk be, - ha már benn volt akkor vegyük ki a foglalatból és rakjuk vissza - az IC-t a programozó foglalatba, és már indul is a programozás. (A programozó észleli hogy beraktuk az IC-t.)

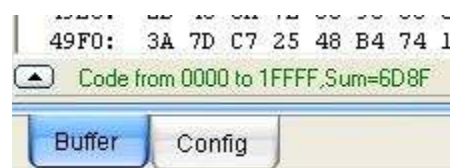
A végén kéri a következőt, egyben jelzi hogy hányadiknál tartunk:



Na igen, itt van egy szépséghibája a programnak. Mégpedig, hogy az első IC-nél, ha már beraktuk a foglalatba, akkor azt ki kell vennünk, majd újra berakni, hogy elinduljon a programozás. De egy sorozat programozásnál tényleg hasznos ez a funkció, egyszerűen csak rakni kell az IC-eket egymás után a foglalatba, és kész.

#### A „buffer” memória:

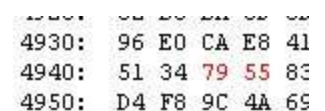
Ha középen a „Buffer”-ra kattintunk, akkor az átmeneti tár - buffer - tartalmát látjuk, azaz az IC-ből vagy egy fájlból beolvasott adatokat.



A beolvasott adatokból egy „Check Summa”-t, azaz ellenőrző összeget is kiszámol a program, ez látható a SUM=xxxx . (Az ellenőrző összeg lényege, hogy valamennyi adat bájtot összegzi a program, aminek az eredménye a „csekksumm”. Ebből következik, hogy ha csak egy bájt értéke is változik, akkor az adott tartalomhoz tartozó ellenőrző szám is más lesz.

A „buffer” tartalmában a PgDown és PgUp gombokkal lapozhatunk, de csak azután, ha az égerrel „belekattintunk” a „buffer ablakba”.

Az adatokat át is tudjuk írni, azaz „editálhatjuk” a tartalmat. Az átírt bájtok színe piros lesz.





### Egy fájl betöltése a „bufferbe”:

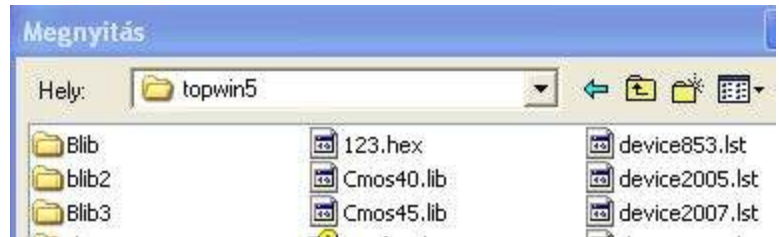
A File/Open paranccsal, vagy az



ikonra kattintva tudunk egy

égetendő tartalmat beolvasni a bufferbe.

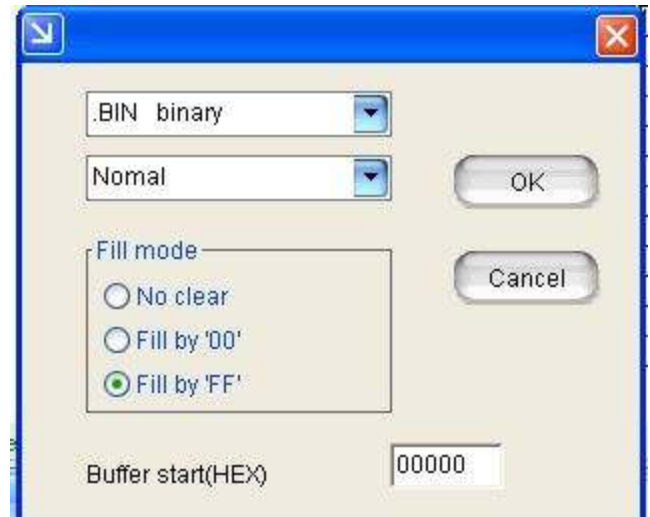
A szokásos böngésző menüben válogathatunk a fájlok között.



A kiválasztás után egy újabb ablak jelenik meg:

Egyrészt megadhatjuk, hogy a beolvasandó fájl egy „sima” adat csomag - ez a bináris, azaz .BIN fájl - vagy valamilyen speciális fájl formátum, mint pl. a .HEX vagy .JED fájl.

A HEX, JED, stb. formátumú adatfájlokat automatikusan feldolgozza - „kicsomagolja” tulajdonképpen egy bináris fájlra alakítja - és „buffer ablakban” már az égetendő adatok látszanak.



Aztán pedig választható egy „Fill mode”, ami - talán - arra szolgál, hogy az üresen maradt helyet milyen adattal töltsse fel, azaz 00-val, vagy FF-el. - (pl. ha egy 32Kb-ajos memóriába 4K-s adatot töltünk, akkor ugye marad 28K üresen.) De én nem láttam különbséget, **hagyjuk az alapértelmezett FF-en.**

És van még egy lehetőség, hogy az adatokat ne a memória elejétől töltsse be, hanem egy adott címtől. Ez a cím adható meg Buffer start(HEX) ablakban. **Normál esetben persze maradjon a 00000 kezdőcím.**

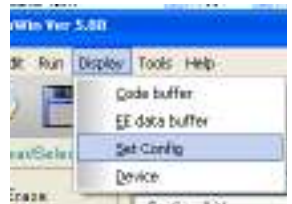
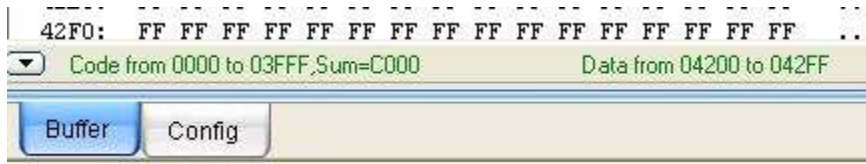
### Az adatok módosítása, az „Edit” menü:

A buffer memóriába került adatokkal különböző műveleteket végezhetünk. Átírhadjuk azokat egyenként, vagy - az EDIT menüt meghívva - kereshetünk benne hexa, vagy ASCII kód szerint, feltölthetjük egy adat byte-tal a munkaterületet, azaz az buffert. (Pl. a "üres EEPROM-ot" csinálunk ha FF-el végig égetjük.)

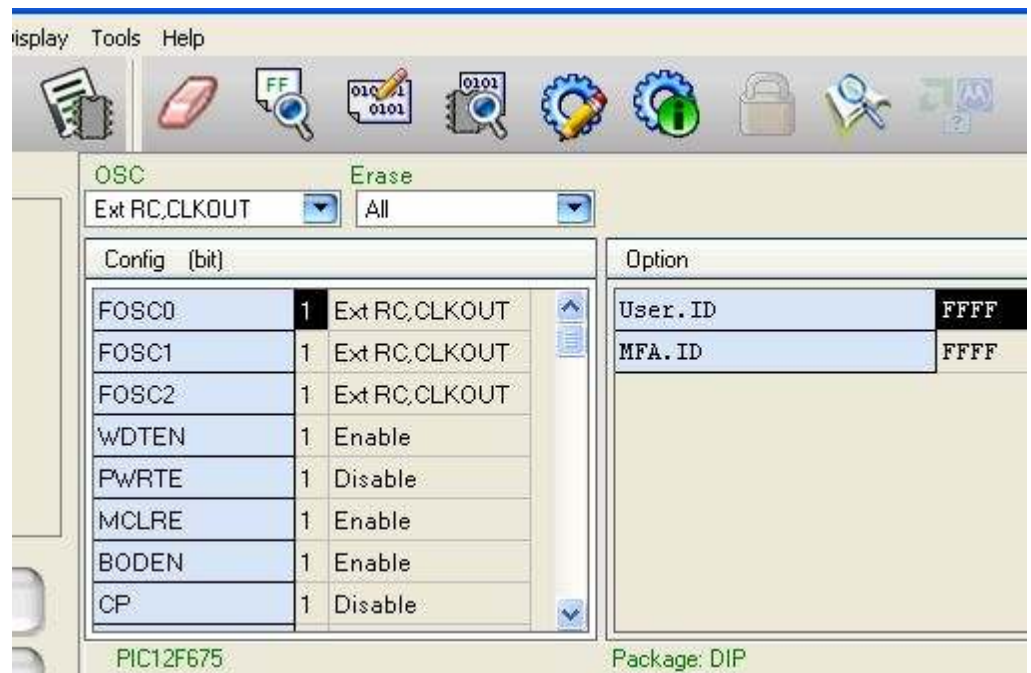


### Adat memória, konfiguráló bitek:

Vannak több programozható területtel rendelkező IC-k, ahol a „Code Buffer” mellett van EEPROM memória is, vagy/és konfiguráló bitek. Ezek a „Display” menü „EEP Data Memory”, és „Set config” almenüjével, vagy a „buffer” memória alatt a „Data from ...” és a Config-ra kattintva tekinthetők meg, illetve módosíthatóak.



Pl. a Microchip PIC-eknél program memória mellett van egy külön EEPROM memória, és egy működést beállító bitmező is. Ezek tartalma tekinthető meg a "EEP Data Memory" és "Config" ikonra kattintva.



És még egy kicsit maradjunk a PIC-eknél.

Az előbbi beállító biteket - „fuse”-okat, ahogy a MICROCHIP nevezi őket - tartalmazhatja a .HEX fájl is. (Nem mindig tartalmazza.) A programozó azonban ezeket nem olvassa ki - komoly hiba - külön kell beállítani azokat.

Erre egy megoldás lehet, hogy egy MICROCHIP-es PIC programozó PC programjába beolvassuk az érintett .HEX fájlt, és ott megnézzük a beállításokat, majd azok szerint beállítjuk.

Vannak IC-k, amelyek tartalma titkosítható - azaz később nem olvasható ki belőlük a program - ezeknél a „Security”, azaz titkosítás is kijelölhető.



### A programozandó IC kiválasztása:

Ahogy arról az előbbiekben már szó volt, a "Device" menüben jelölhető ki, milyen IC-vel akarunk foglalkozni.



Egy egér kattintással ki kell jelölnünk, hogy milyen IC csoportból akarunk választani – EPROM, EEPROM, mikrokontroller, soros EEPROM, stb.

A kezelendő IC típus gyors beállítását segíti a "típus keresés". Tehát, ha pl. egy 24C256 EPROM-ot akarunk égetni, akkor a C256-ot beírva a kereső sorba az összes C256-os azonosítót tartalmazó soros EEPROM IC-t megjeleníti a

választásra, valamint a gyártó cégeket is. Ha ismeretlen a gyártó, esetleg nem szerepel a gyártók listájában, akkor a ??? -et válasszuk.

### Egy „Project” mentése:

Ahogy arról szó volt, vannak olyan programozható IC-k, amik egy csomó programozható beállítási lehetőséget is tartalmaznak.

A TOP2049 programjának van egy nagyon hasznos funkciója, a File menüben az „Open Project” és „Save project”. Itt minden, a programozáshoz szükséges beállítás elmenthető.

Tehát ha pl. beállítottuk egy mikrokontroller típusát, a beállító bitjeit, behívtuk a bele égetendő adatokat, akkor ezt mind egy helyre, egy „project”-be menthetjük. (És persze később beolvashatjuk.)

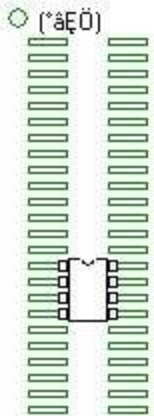


### A programozandó IC behelyezése a foglalatba:

Ahogy a készülék dobozán is látható, az az „alapértelmezés”, hogy a foglalatban lefelé kell sorolni az IC-eket, azaz amennyire csak lehet, lefelé.

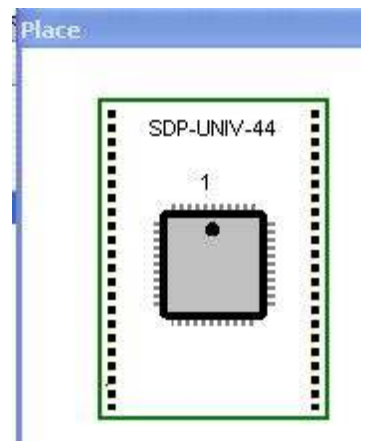
A PC program ezt annyira egyértelműnek veszi, hogy csak akkor jelzi ki az IC helyét a programozó foglalatban, ha az előbbtől eltérő módon kell berakni az IC-t.

(Valóban csak pár olyan IC van, amit nem legalulra kell sorolni, de azért – csak megnyugtatásul – mindig ki rajzolhatná az IC helyét a program.)



Szintén egy ábra fogja megmutatni, ha adapter kell az IC-hez.

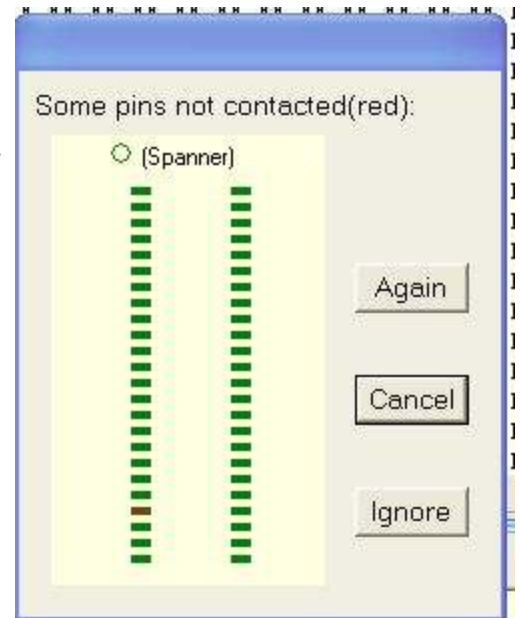
Egy pár szó az adapterekről: minden adapter, a PLCC/DIP és az smd/DIP adapterek is 1:1 bekötésűek, azaz pl. a PSOP44 tokos smd adapternél az smd IC tok 1-es lába a DIP tok 1-es, és így tovább, a 44. a 44.-hez.



A programozó képes meghatározni az esetleges kontakt hibát is a behelyezett IC-nél.

Ha a kiadott művelet – beprogramozás, kiolvasás – sikertelen, akkor megpróbálja azonosítani a lehetséges kontakt hibát, és az ki is jelzi.

Ezt a képet pl. egy 44 lábú smd flash eeprom programozásakor kaptam, a 20. lábon észlelt kontakt hibáról:



### Az IC teszter:

A „Tools” menüben található az „IC tester” menü.

TTL és CMOS logikai IC-k működő képességét állapíthatjuk meg vele.

Két fajta vizsgálatot végezhetünk vele.

Beállíthatjuk a kívánt IC típust, majd a „Test”-re kattintva lepróbálja azt, ha minden rendben, az OK üzenet jelenik meg:

De van egy másik, érdekes funkció is, ha egy „ismeretlen” logikai IC-t csíptetünk a foglalatba, majd a „Search”, azaz kutatás gombot nyomjuk be, akkor a készülék megpróbálja kideríteni a típust:

### Tápellátás:

Mint arról korábban szó volt, a tápot az USB portról veszi az égető. Külső tápegység kizárólag akkor kell, ha olyan USB elosztón - HUB - keresztül csatlakoztatjuk, ami nem ad ki elég tápot.

Még egy esetben lehet szükség tápra: ha olyan, régi EPROM típust kezelünk - pl. 27xxx sorozat - ami több száz mA-ert vesz fel égetéskor.

Ami tetszik, és ami nem:

- Rengeteg EPROM , EEPROM típust tud, a mikrokontrollereknél az ATMEL „egzotikus” típusait is, plusz GAL-okat, PALCE-t is, és mindezt külön adapter nélkül. (A PLCC, smd IC-khez persze kell átalakító, hogy a karos foglalatba csíptethessük.)

- Kijelzi, ha nem ismeri fel az IC-t, hogy hol keressünk kontakt hibát. Ez pl. az smd adaptereknél nagyon hasznos.

- TTL IC-k is tesztelhetők vele.

- Nagyon gyors.

- A MICROCHIP PIC mikrokontrollereknél nem olvassa be a HEX fájlból a „fuse”-okat. Ez nagy baki! Ha sokszor kell MICROCHIP PIC mikrokontrollert égetni, akkor inkább pl. a VP-380 programozót válasszuk... (Vagy, a legjobb, egy külön PIC égető a PIC-eknek. Pl. az USB-s PICKIT2. Erről bővebben a lenti honlapon.) Ellenben ha ezeket beállítjuk, akkor „project”-ként menthetjük. Ez, azaz a beállító bitek mentése pl. az ATMEL mikrokontrollereknél igen csak jól jön.

- Kiolvassa a chip gyári azonosítóját. (Már amelyiknek van ilyen.) De csak a kódot jelzi ki, azt nem hogy ez melyik gyártó melyik chipje.

- A szoftver lehetne kicsit „kezesebb”. Pl. az IC választáskor pl. meg kell adni hogy EEPROM, mikrokontroller, stb. típusokból keressen, vagy csak akkor jelzi ki az IC helyét a foglalatban, ha az eltér a „szokásos” lefele rendezéstől.

- Van sorozat programozás funkciója.

Összefoglalva egy nagy tudású, gyors égető, a kezelő szoftver néhány apróbb szépséghibájával. De megéri az árát, ha pedig az epromokon kívül mikrokontrollereket, 16 bites epromokat, esetleg GAL-okat is kell égetnünk, akkor nagyon.

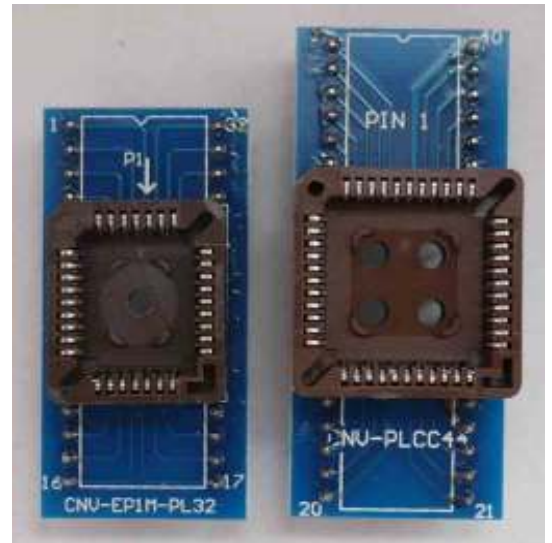
### Adapterek PLCC, smd tokozású EPROM-okhoz:

#### Adapterek PLCC tokozású IC-khez :

Az adapterek lehetővé teszik a PLCC tokos epromok, mikrokontrollerek kiolvasását, égetését. Kiválasztjuk az IC típust - mivel DIP átalakítót használunk, a DIP tokos verziót jelöljük ki - majd az adaptert egyszerűen be kell dugni az égető karos IC foglalatába, aztán ugyanúgy égetjük, vagy olvassuk, mint a normál, DIP tokozású alkatrészt.

A TOP2049 égetőnél a PLCC32-es tokos IC-hez az 1:1-es DIP32-PLCC32 a PLCC44 tokos IC-k többségéhez pedig egy DIP40-PLCC44 adapter kell.

(Ezeket a gyártó jelenleg tartozékként adja.)



#### PLCC kizsedő csipesz:

Amilyen könnyű eltörni a plcc IC foglalatot az IC kifeszése közben, olyan nehéz azt kicserélni a panelen... Az ilyen balesetek elkerülését segíti a plcc kizsedő csipesz:



### Adapterek smd tokos alkatrészekhez:

A 8 lábú DIP IC tokra forrasztott panel végszükségletben, mint "fapados" SMD adapter is használható programozó készülékekhez. Értelmszerűen a DIP tokot csíptessük a karos foglalatba, az SMD tokot pedig az olvasás vagy programozás alatt szorítsuk a panel SMD fóliázatára.



### DIP8-SMD8 ZIF foglalatatos programozó adapter:

Az előbbi egyszerű adapter ZIF IC tokos megfelelője. (A "zero force insert" a könnyített IC behelyezést jelenti most.)



Szintén az SO8 tokos IC-khez használható. A tokozatot felülről megnyomva, felemelkedik a nyolc "lábacska" ami a behelyezett smd alkatrész kivezetéseit fogja meg. Az IC így behelyezhető, a rugós befogást visszaengedve, az aranyozott érintkezők megfogják az IC lábakat.

Szintén az SMD IC-k kezelését könnyítik meg az SMD csipeszek. A csipeszt egy szalagkábeles DIP8-as IC csatlakozóval tudjuk az égetőhöz kötni. Arra figyeljünk, hogy az IC csatlakozóknál a kijövő vezetékek "tükörben" vannak, tehát az első vezeték nem az 1-es, hanem a 8-as. (Tehát 8, 1, 7, 2, 6, 3, 5, 4 lesz a csatlakozóból kijövő vezetékek sorrendje.)



A "csíptetésnél" persze figyeljünk, hogy az smd IC 1-es lába legyen összeköttetésben a programozó foglalat 1-es lábával.

Gyakori kérdés, hogy programozható-e kiforrasztás nélkül, az áramkörben a chip. Esete válogatja, illetve próba szerencse... Szóval ez egyrészt függ az áramköri környezettől, azaz mi csatlakozik az eeprom lábaira - általában egy mikrokontroller - és a programozó tudja-e annak ellenében mozgatni a kivezetések szintjét.

Valamint még a tápellátás "kritikus". Azaz hogy a programozó által adott 5 volt elég-e a panel számára. (Hiszen nem csak az eepromot fogja a tápot terhelni, hanem minden 5 voltra csatlakozó IC.)

Szóval lehet hogy 1-2-3 lábat "fel kell emelni" a panelről. Vagy - ez a biztos - le kell forrasztani az IC-t a panelről, és úgy "meg csípti".



### Adapter PSOP44 tokozású IC-khez:

Az adapterkártya lehetővé teszi az ssop44 tokozású IC-k kiolvasását, beégetését. (Pl. 29F200, 29F400, 29F800) Az adapterbe kerül az EEPROM, majd azt egyszerűen be kell dugni az égető 48-as karos IC foglalatába, ezután pedig ugyanúgy égetjük, mint egy normál, DIP tokozású alkatrészt.

Az adapter teljes leírása a PSOP44.PDF-ben olvasható.



### Adapter TSOP48 tokozású IC-khez:

Az adapterkártya lehetővé teszi a tsop48 tokozású IC-k kiolvasását, beégetését. (Pl. 29F200, 29F400, 29F800) Az adapterbe kerül az EEPROM, majd azt egyszerűen be kell dugni az égető 48-as karos IC foglalatába, ezután pedig ugyanúgy égetjük, mint egy normál, DIP tokozású alkatrészt.

Az adapter teljes leírása a TSOP48.PDF-ben olvasható.



### Kapcsolódó dokumentáció, szakirodalom:

A 27xxxx EPROM-okról, és azok törléséről szól az EPROM.PDF, a soros EEPROM-okról az EEPROMOK.PDF. (A CD-n a MIKROKLB\MIKROPO\DOCUMENT könyvtárban találhatóak.)

Egy UV fényes törlő lámpa leírása található az UVTORLO.PDF file-ban. (A CD-n a MIKROKLB\UVERASER könyvtárban.)

A PLCC EPROM-okhoz használható adapterekről PLCCEP.PDF szól. (A CD-n a MIKROKLB\MIKROPO\ADAPTERS\PLCCEP könyvtárban találhatóak.)

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

Végül nincs más hátra, mint hogy sok sikert kívánjak a használathoz. Torkos Csaba, 8100 Várpalota Táncsics u. 7. Telefon, napközben, 8-16 óráig: 06/88/473-784, vagy - egész nap, 8-22 óráig - a 06/30/9 472-294 számon. Email: [mikroklub@vnet.hu](mailto:mikroklub@vnet.hu) Internet: <http://www.mikroklub.hu>, <http://www.eprom.hu>