

A VP-ISP10 soros IC programozók

A Wellon cég IC programozó készülék családjának legkisebb tagja VP-ISP10 programozó. Nézzük, mit tud:

VP-ISP10 soros programozók:

A VP-ISP soros programozók csak a sorosan, tehát néhány csatlakozóponton keresztül írható, olvasható IC-khez használhatóak. (Általában a soros eepromok, és az újabb mikrokontrollerek tudják a soros program letöltést.)



Az égető hardvere itt a lehető legegyszerűbb, nincs is programozó foglalat, csupán az IC programozáshoz szükséges csatlakozót, és kábeleket biztosítja.

A VP-ISP10:

- 3 - 5 voltos IC-k támogatása
- Windows 98/2000/XP/Vista/Win7 alatt működtethető. (Amúgy ugyanez a PC szoftver használható a nagyobb wellon égetőhöz is, a program felismeri a csatlakoztatott égetőt, és a működését ahhoz igazítja.)
- Kicsi, strapabíró műanyag doboz.
- Az égetőt egy mikrokontroller vezérli. Ennek talán a legnagyobb előnye az, hogy stabilak, a windows-tól függetlenek a programozási folyamatot meghatározó időzítések.
- Ingyenes szoftver frissítés.

A kezelt IC-k:

Az égető tudását két „modul” határozza meg. A gyártó a kezelt IC-k választékát egységekre bontotta, ezekre mint „modulokra” hivatkozik. Minden égető tartalmazza az alapmodult - MOD0 - ami a „szokásos” soros EEPROM típusokat tartalmazza, nem is keveset, kb. 2500 típust. Szóval a 24xx, 93xx, 25xx memória IC-eket.

És van egy másik „modul”, ami további IC-eket ad az eszközlistához. Ezek lehetnek speciális eepromok, vagy sorosan programozható mikrokontrollerek.

És akkor három variáció:

MOD0 + MOD1 : az „alap” eeprom készlet kiegészül az spi flash eepromokkal, pl. 25Pxx, 25Lxx stb.

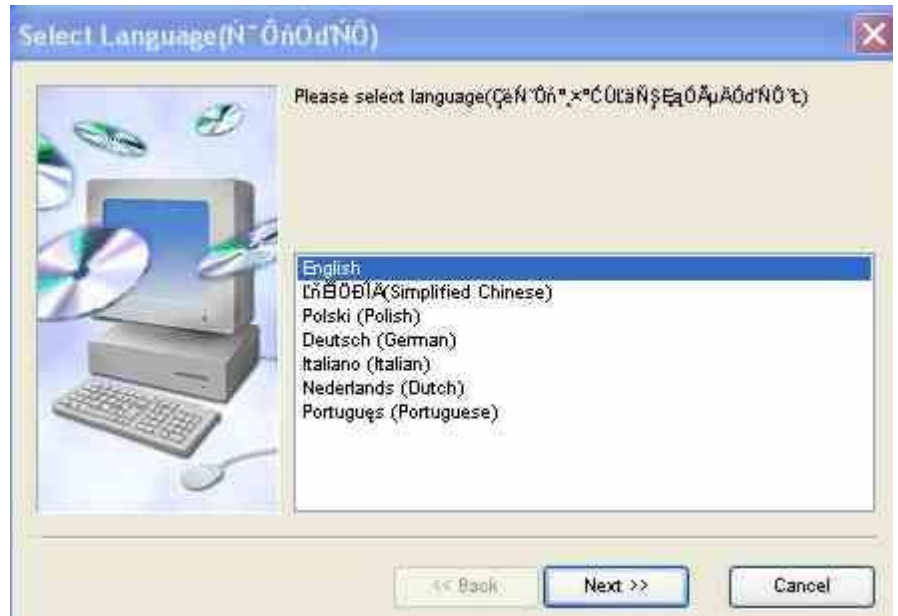
MOD0 + MOD2 : az „alap” eeprom készlet kiegészül az ATMEL 45Dxxx, 45DBxx eepromokkal.

MOD0 + MOD9 : itt pedig az ATMEL 89Sxxx, és az ATMEGA, ATTINY mikrokontrollerekkel bővül az „alap” eeprom készlet.

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.



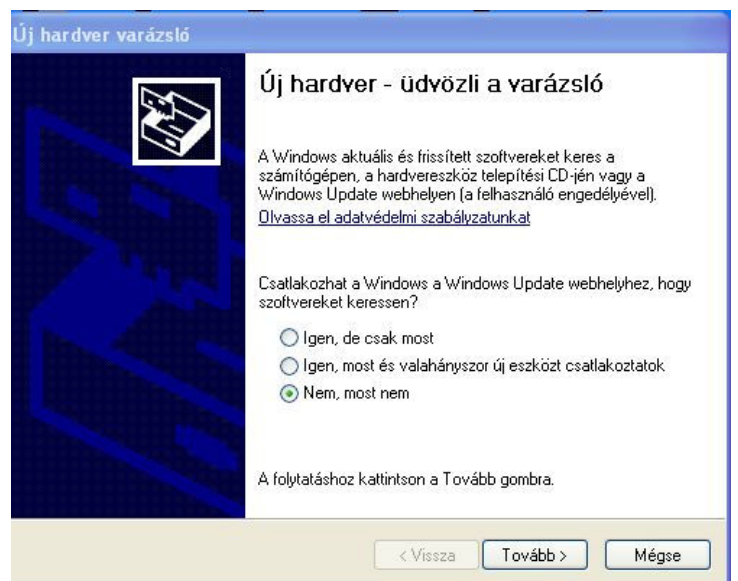
- A PC program installálása után, de még a program futtatása előtt csatlakoztassuk az égetőt, egy szabad USB porthoz.

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

A csatlakoztatás után telepedik fel az égető meghajtó programja, a windows új hardver-t fog találni:



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



- Aztán telepítés „beavatkozás nélkül” :



Ha a hardverhez kapott telepítőlemez (CD-t vagy hajlékonylemez), helyezze be a meghajtóba.

Mit tegyen a varázsló?

- ☒ Telepítse a szoftvert beavatkozás nélkül (ajánlott)
- ☐ Telepítsen listából vagy adott helyről (háladókknak)

A folytatáshoz kattintson a Tovább gombra.

Azzal ne törődjünk, ha az XP figyelmeztet, hogy ez a szoftver nem esett át microsoft tesztjén, folytassuk a telepítést:



Ha minden rendben, a végén az eszközvezérlőben, az „USB-vezérlők” között, meg találhatjuk a programozónkat:



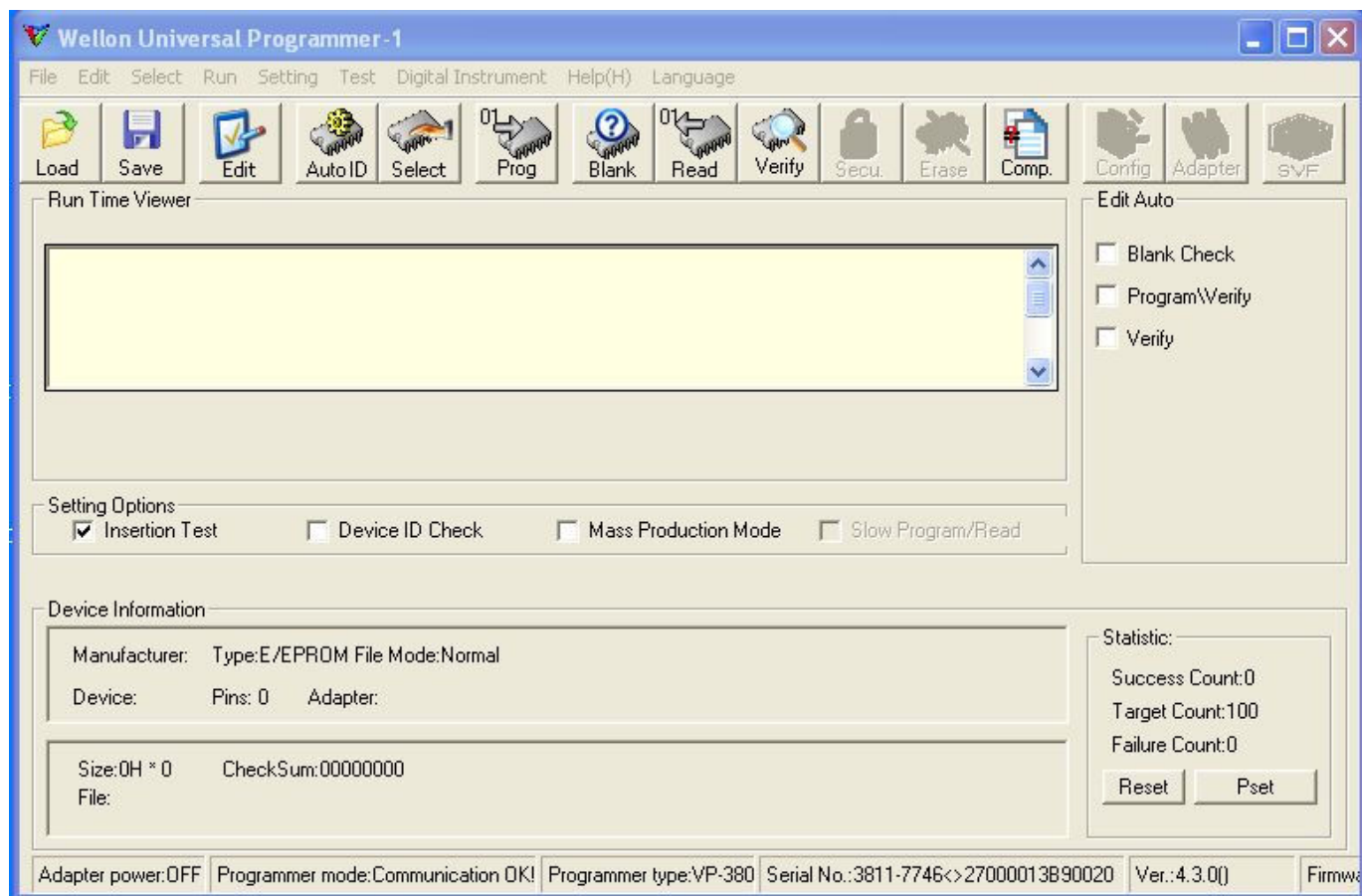
Fontos dolgok a használatához:

A bekapcsolás alatt ne legyen IC a csatlakozó pontokon, 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 PC program automatikusan felismeri, és ellenőrzi a csatlakoztatott égető hardvert. Ugyanaz a PC program kezeli a többi wellon égetőt is, a PC program azonban az induláskor azonosítja a típust, és a működését – pl. az IC típus választásnál – ahhoz igazítja.



A képernyő felosztás:

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



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

Jobb oldalt a programozáskor használatos műveletekből állíthatunk össze egy automatikusan végrehajtódó művelet sorozatot. (Pl. IC törlés, törlés ellenőrzés, programozás, stb.)

A legalsó sorban az égető állapotjelzése, azaz hogy kap-e külső tápot, aztán a kommunikáció OK, majd az égető típusa, gyári száma, program verziója.

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.



"Auto ID" : elektronikus azonosító kiolvasása a chipből. (Nem minden mikrokontroller, vagy memória rendelkezik kiolvasható azonosítóval.)



"Select chip" : IC típus választás.



"Program" : a bufferben lévő adatok beégetése - beprogramozása, írása - a chipbe.



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



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



"Verify" : az IC-ből kiolvasott adatok összehasonlítása a buffer adatokkal.



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



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



Compare : Az IC-ből kiolvasott adatok, és a „buffer” összehasonlítása.



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



Adapter : itt mutatja meg, hogy az adott IC-hez hogyan kell csatlakoznunk a kábelekkel.

A programozáskor használt parancsok kiadhatóak egy műveletsorként is. A jobb oldalon van az automatikus programozás parancs mezője - Edit auto - ahol kiválaszthatjuk a programozás során végrehajtandó művelet sort.


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

Ha pl. egy 93C56 EEPROM-ot akarunk törölni, és utána ellenőrizni is a törlést, majd égetni, végül ellenőrizni az égetést, akkor a jobb oldali ablakban jelöljük ki ezeket a műveleteket:

Edit Auto	
<input checked="" type="checkbox"/>	Erase\Blank Check
<input type="checkbox"/>	Blank Check
<input checked="" type="checkbox"/>	Program\Verify
<input type="checkbox"/>	Verify

Egy fájl betöltése a „bufferbe”:



A File/Open paranccsal, vagy az  ikonra kattintva tudunk egy égetendő tartalmat beolvasni a bufferbe.

Egy „szokásos” böngésző menüben válogathatunk a fájlok között.

A fájl kiválasztása 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.

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

File: B:\DS1820RS.HEX

From File Mode: Normal

To Buffer Mode: Normal

To Buffer Address(HEX): 0

From File Address(HEX): 0

File Size(HEX): 80

Auto Format Detected

- ☐ Binary
- ☒ Intel
- ☐ Motorola S
- ☐ Tektronix
- ☐ Holtek
- ☐ EMC(CDS)
- ☐ Intel Hex16

Clear Buffer Options

- ☐ Disable
- ☐ 0x00
- ☒ 0xFF

OK Cancel

Aztán választható az is, a „Clear Buffer Options” ablakban, hogy az üresen maradt helyet milyen adattal töltsse fel, azaz 00-val, vagy FF-el, vagy ne csináljon semmit a maradék hellyel, „Disable”. Ezt az opciót általában hagyjuk az alapértelmezett FF-en.

Pl. ha egy 32Kbájtos memóriába 4K-s adatot töltünk, akkor ugye marad 28K üresen, ami így FF-el .

És van még egy lehetőség, hogy az adatokat ne a memória elejétől töltsse be, hanem egy adott címtől. Ez a cím adható meg To Buffer Address(HEX) ablakban. Normál esetben persze maradjon a 0 kezdőcím.

De általában is elmondható, hogy egy „szokásos” IC programozásnál ne változtassuk meg az alapbeállításokat.

A „buffer” editor:



Ha az „Edit”-re kattintunk, akkor az átmeneti tár – a buffer – tartalmát látjuk, azaz az IC-ből vagy egy fájlból beolvasott adatokat.

Buffer Editor																
Fill Random Swap Search Next Copy CheckSum Print Show Exit																
ADDRESS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000H	23	28	FF	FF	FF	FF	FF	FF	FE	00	03	0E	83	01	FF	00
00000010H	0C	1D	1E	28	0C	11	64	00	A5	0B	1E	28	0A	30	A5	00
00000020H	01	30	A0	06	A6	0B	1E	28	04	30	A6	00	98	1C	1B	28
00000030H	18	12	00	00	18	16	01	30	86	06	A0	14	7F	0E	83	00
00000040H	FE	0E	7E	0E	09	00	85	01	86	01	07	30	9F	00	83	16
00000050H	8F	30	81	00	30	30	85	00	02	30	86	00	0C	15	18	15
00000060H	19	30	99	00	98	16	83	12	98	17	18	16	20	30	84	00
00000070H	80	01	84	0A	84	1F	38	28	0A	30	A5	00	04	30	A6	00

Az adatokat nem csak nézegethetjük, de át is tudjuk írni, azaz „editálhatjuk” a tartalmat. Az átírt bájtok színe kék lesz.

A „Fill” paranccsal egy adot kóddal tölthető fel a buffer, a „Random” paranccsal pedig véletlen szerű értékekkel. (Gondolkodtam ez utóbbi parancs értelmén, de pl. próba égetéseknél, programozó adapterek tesztjénél jól jön.)

A „Swap” , azaz felcserélés menüre kattintva a szomszédos adat bájtok helyet cserélnek. Azaz az első a másodikkal, a harmadik a negyedikkel, és így tovább. Na, ennek mi értelme? Lehet a jogos kérdés. Pl. a 16 bites epromoknál ugye két egymást követő 8 bites adat adja meg a 16 bit tartalmát. De arra nincs semmi előírás, hogy a bitek sorrendje 15.-8. és 7.-0., vagy előbb vannak az alacsonyabb helyi értékűek, azaz 7.-0. és 15.-8. Ez attól függ, milyen programmal tárolták le azokat, stb. És néha jó lenne megcserélni a sorrendet.

Kereshetünk az adatok közt hexa, vagy ASCII kód szerint – szöveget – a „Search” paranccsal. A „Next”-el folytathatjuk a keresést.

A „Copy”-val egy kijelölt terület tartalmát másolhatjuk egy másik kijelölt területre.

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 „csekkszumma”. Ebből következik, hogy ha csak egy bájt értéke is változik, akkor az adott tartalomhoz tartozó ellenőrző szám is más lesz.

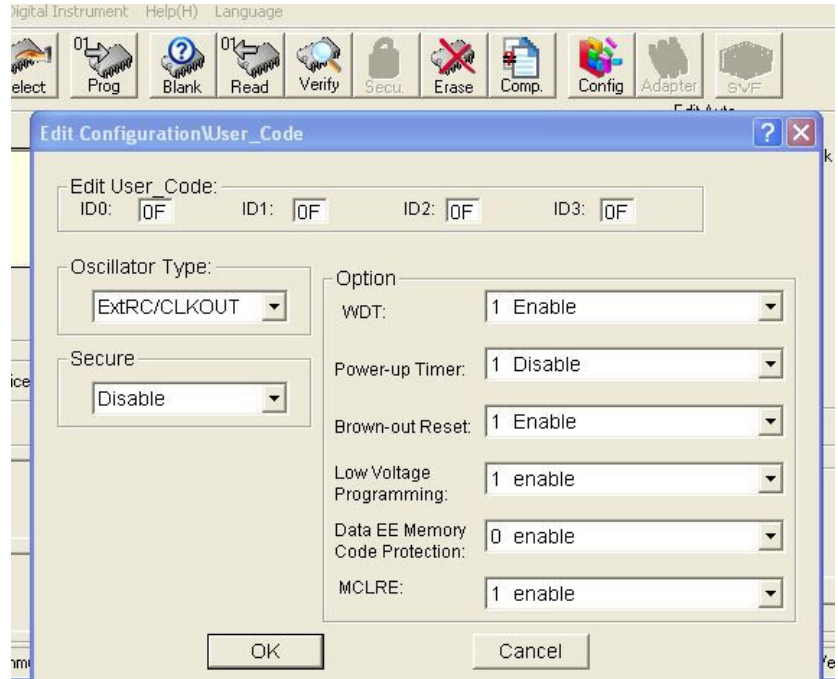
Adat memória, konfiguráló bitek:

Vannak több programozható területtel rendelkező IC-k, ahol a „Code Buffer” mellett van EEPROM memória is, vagy/és konfiguráló bitek. Ezek az „Edit” menü „Edit E_field”, és „Edit Configuration\User_Code” almenük alatt tekinthetők meg, illetve módosíthatóak.

Elvileg. Mert nekem csak a Config biteket sikerült szerkesztenem pl. a Microchip PIC-ek esetében, azoknál a típusoknál is, ahol pedig van eeprom adat memória is.



Pl. a Microchip PIC-eknél program memória mellett van egy működést beállító bitmező is. Ezek tartalma tekinthető meg a „Config” ikonra kattintva.



Vannak IC-k, amelyek tartalma titkosítható – azaz később nem olvasható ki belőlük a program – ezeknél a „Security”, azaz titkosítás is kijelölhető. (A mikrokontrollereknél a konfiguráló biteknél kell beikszelni a program memória kiolvasásának tiltását.)



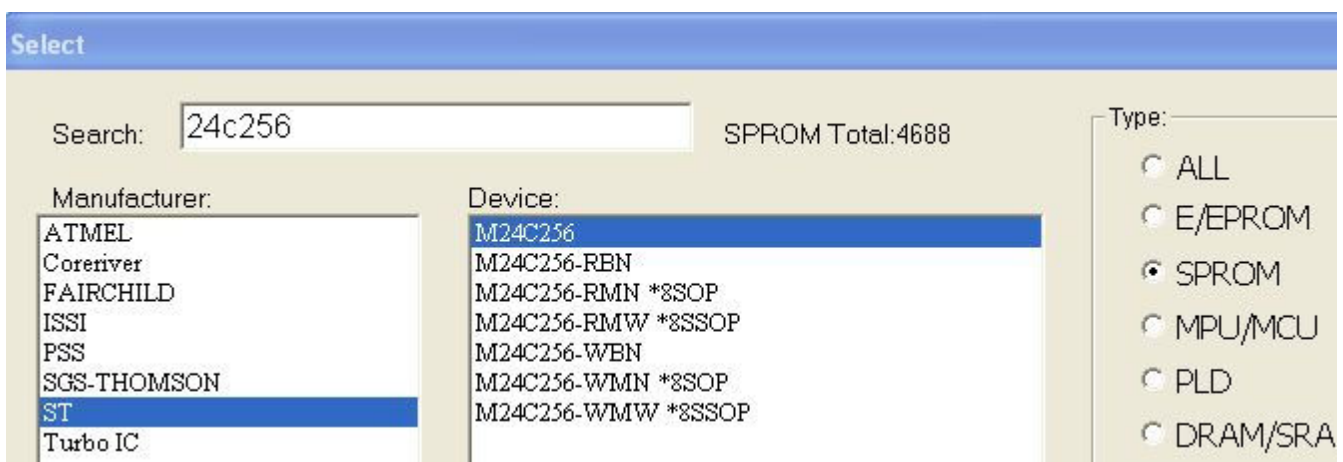
A programozandó IC kiválasztása:

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



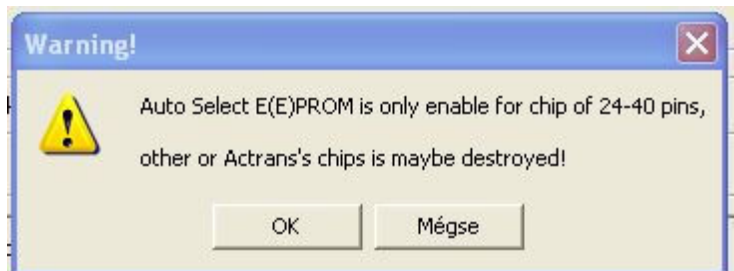
Egy egér kattintással ki kell jelölnünk, hogy milyen IC csoportból akarunk választani - EPROM, EEPROM, mikrokontroller, soros EEPROM, stb.

A kezelendő IC típus gyors beállítását segíti a "típus keresés". Tehát, ha pl. egy ST gyártotta M24C256 EPROM-ot akarunk égetni, akkor a 24C256-ot beírva a kereső sorba, akkor megjeleníti az „SPROM” típust - a Serial Prom rövidítéseként, bár szerintem a serial eeprom jobb megnevezés lenne - és megjeleníti a gyártó cégeket. Ezután tudjuk gyártók típusaira szűkíteni a kínálatot, és végül választani:



EPROM és EEPROM típusoknál kiadható az automatikus chip azonosítás.

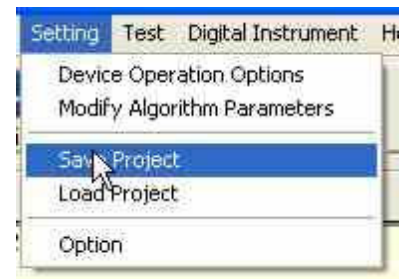
De - erre figyelmeztet a program - csak 24-40 kivezetéses (E)EPROM esetében adjuk ki ezt a parancsot, mert más IC-károsodhatnak!



Egy „Project” mentése:

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

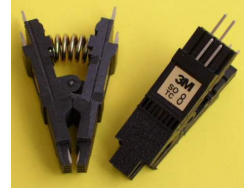
A programnak van egy nagyon hasznos funkciója, a Setting menüben a „Load Project” és „Save project”. Itt minden, a programozáshoz szükséges beállítás elmenthető.



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.)

Az SMD IC-k kezelését könnyítik meg az SMD csipeszek. A csipeszt egy szalagkábeles DIP8-as IC csatlakozóval tudjuk az égetőhöz kötni.

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



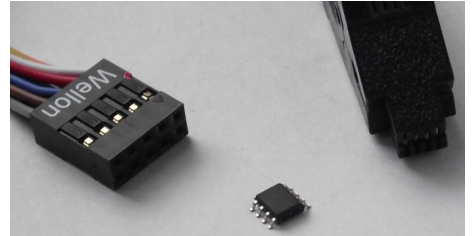
És akkor egy gyakorlati példán keresztül bemutatva a használatot, mondjuk hogy egy smd 24c02-est, egy ST gyártmányút akarunk írni, olvasni.

Az IC csatlakoztatásához egy csipeszt használunk. A programozóhoz mellékelt 10 eres kábel csatlakozó színes vezetőkeinek a végén egy-egy tuskére tolható hüvely van.

Ezek persze ráhúzhatóak az smd csipesz tuskéire.

Hogy melyik kábel hova kerül, azt majd a programozó pc programja kijelzi.

Tehát indítsuk el pc programot, és válasszuk ki az IC típust. Írjuk be a kereső sorba a 24c02-est, majd szűkítsük a választékot az ST gyártmányokra, és válasszuk az ST24C02M *8SO(ISP)

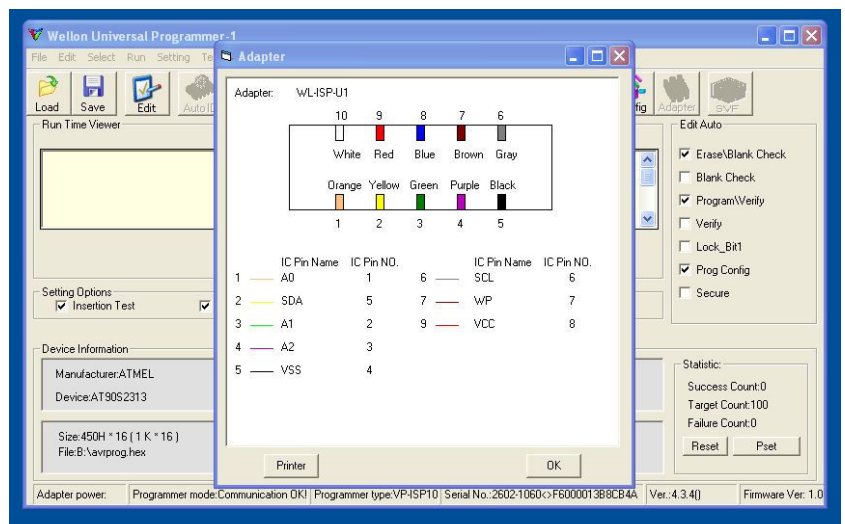


Az OK-ra kattintva előugrik

a „bekötési ábra.”

Fogjuk a programozó 10 eres kábelcsatlakozóját, és kössük rá 8 lábú smd csipeszünket.

Ahogy látszik, az 1-es, narancs vezeték megy az IC - csipesz - 1-es lábára, a 2-es, sárga az 5. csipesz tuskére, és így tovább.



Nézzük át még egyszer a vezetékezésünket, majd nyomjuk a - most már egy csipesszel kiegészített - csatlakozót a programozó 2x5-ös tűskéjére. A WELLON felirat legyen felül. Csípjük meg az IC-t.

És most már kiadhatjuk a READ, vagy PROG parancsot, attól függően, hogy pl. olvasni, vagy programozni akarjuk az IC-t.

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 csípni".

Kapcsolódó dokumentáció, szakirodalom:

A wellon cég nagyobb, univerzális programozói:

VP-380:

- több mint 10.000 (!) támogatott IC típus. EPROM, EEPROM, , Atmel, Intel, Microchip, Signetics, Zilog mikrokontrollerek kezelése, de GAL és PALCE IC-ket is programozhatunk vele. USB-s csatlakozás, nagyon gyors adatforgalommal. A tápfeszültséget is az USB adja, de lehetőség van külső táp csatlakoztatására is.

VP-590:

- A VP-380-nál többet tudó, több mint 25.000 IC típust támogat.
- USB csatlakozás, kapcsolóüzemű külső tápegység. (tartozék)

VP-990:

- a wellon cég csúcs programozója, csaknem 50.000 támogatott IC típus.
- USB csatlakozás, kapcsolóüzemű külső tápegység. (tartozék)

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

Végül nincs más hátra, mint hogy sok sikert kívánjak a használatához.
 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 Internet: <http://www.mikroklub.hu>,
<http://www.eprom.hu>