

ASIX Presto



Kézikönyv

ASIX s.r.o.
Staropramenna 4
150 00 Prague
Czech Republic

E-Mail: sales@asix.net (sales inquiries, ordering)

support@asix.net (technical support)

WWW: tools.asix.net (development tools)

www.asix.net (company website)

Tel.: +420-257 312 378

Fax: +420-257 329 116

Magyarországi forgalmazó: www.epromirok.hu

Bevezetés

Ez a kézikönyv az ASIX gyártmányú programozót és annak vezérlő szoftverét mutatja be. Az első fejezet ismerteti magát a PRESTO USB programozót, a telepítését és a lehetséges bekötéseket, alap jellemzőket. A bekötési részben, a képek alatt a lehetséges programozási problémák is fel vannak tüntetve.

Később bemutatásra kerül az UP szoftver is. Ez a szoftver használható az összes ASIX programozóhoz. A programozók ICSP programozást tesznek lehetővé. Az UP vezérelhető parancssorból is, és DLL libraryval.

JTAG csatolóval rendelkező eszközök programozásához a Jtag Player használható. Ennek a használata a harmadik fejezetben kerül ismertetésre.

A negyedik fejezet ismerteti a Precog-ot. A Precog a Cyan Technology eCOG1 microcontrollerek programozását teszi lehetővé. Ez egyszerű hibakeresésre is alkalmas.

A függelékben található meg a támogatott PIC eszközök konfigurációs adatai és az UP_DLL.DLL beállítások. A C függelék az ICSP programozásról szól. Információk találhatóak itt a programozó által biztosítható áramerősségről, előforduló problémák megoldásáról és egyéb hasznos tudnivalókról.

1. PRESTO

PRESTO egy hatékony, gyors és nagyon jó ár/érték aránnyal rendelkező ICSP* programozó, mely számos gyakran előforduló, népszerű chipet támogat, pl microcontroller, serial EEPROM, Flash memóriák, CPLD, stb. A programozó a maximális programozási sebességre lett optimalizálva, mely rendelkezik túláram védelemmel a Vpp-n és Vcc-n, illetve rendelkezik túlfeszültség védelemmel is a Vcc-n.

A készülék USB porton keresztül kommunikál a számítógéppel és nem igényel külső tápegységet. Az ICSP (In-Circuit Serial Programming - Áramkörre építve történő soros programozás vagy) és ISP (In-System Programming - Rendszerben történő) programozás egy nagyon gyakori és hasznos művelet MCU (microcontroller) esetén. A PRESTO ezt a funkciót a direkt erre a célra kialakított csatlakozójával és kábelével tudja biztosítani.

1.1 Használat

A PRESTO arra lett tervezve, hogy az integrált áramköröket közvetlenül a beépítési helyén programozza és tesztelje.

A támogatott eszközök listája:

- **Microchip PIC** microcontrollerek – soros programozással (Flash, EPROM és OTP), minden PIC és dsPICs, kivéve néhány elavult eszközt.
- **Atmel AVR** microcontrollerek – mindegyik támogatja az "SPI alacsony feszültségű soros letöltést", pl. ATtiny12, AT90S8535 or Atmega128.
- **Atmel AVR32** microcontrollerek - pl. AT32UC3A1256
- **Atmel 8051 család** eszközei – támogatják az ISP programozást. pl. AT89S8253, AT89LP4052, AT89LP216 vagy AT89S2051
- **eCOG1** Cyan Technology mikrocontrollerek
- **Texas Instruments** microcontrollerek – flash MSP430 és CCxxxx

- **Cypress** – PSoC
- **Serial EEPROM** - I2C (24LCxx), Microwire (93LCxx) és SPI (25Cxx)
- **JTAG** csatolóval rendelkező eszközökhöz használható SVF vagy XSVF fájlok lehetségesek. Ezek CPLD (pl. Xilinx XC95xx és CoolRunner), FPGA konfigurációs memóriák (pl. Xilinx XC18Vxx és XCFxxS), microkontrollerek (pl. ATmega128) stb.
- **ARM** magra épülő eszközök – AT91SAM7 mikrokontrollerek programozása és hiba keresése az ARMINE szoftver segítségével lehetséges

1.2 Telepítés

A szoftver és a driver telepítéséhez a felhasználónak rendszergazdai jogokkal kell rendelkeznie.

Először telepítsd fel a PRESTO drivereket és csak **UTÁNA** telepítsd fel az UP szoftvert!

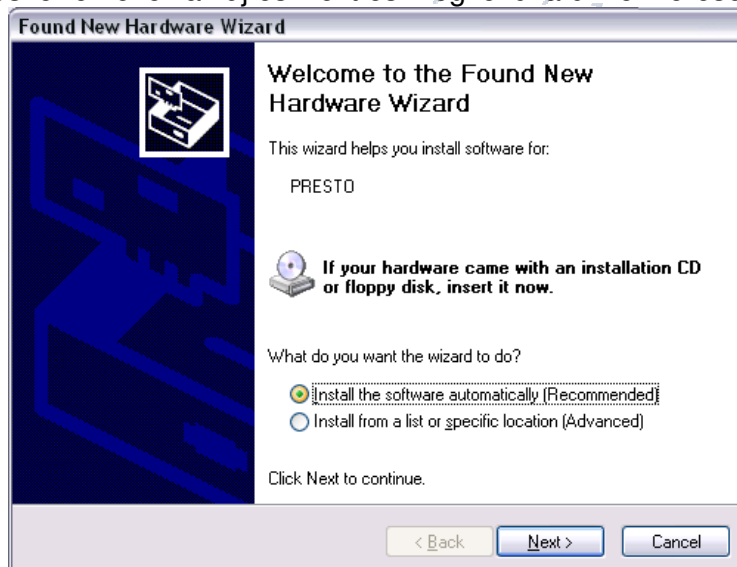
1.3 PRESTO Driver telepítés

Megjegyzése: Windows 98 esetén az új driverből hiányoznak a szükséges .CAT fájlok, ezért ebben az esetben a régebbi driver verziót telepítsd fel.

"ASIX_PRESTO_USB_DRIVERS_2004-04-06.ZIP" letölthető innen:

http://tools.asix.net/dwnld_presto.htm

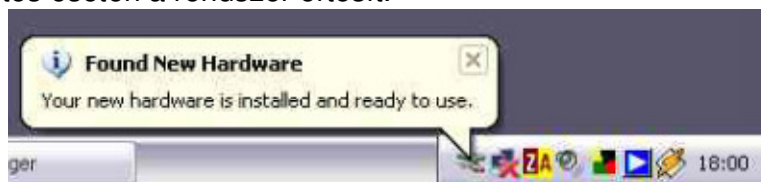
Helyezd be a telepítő lemezt és csatlakoztasd a PRESTO-t az USB portra. Az operációs rendszer érzékeli az új eszközt és megkezdí a driver keresését:



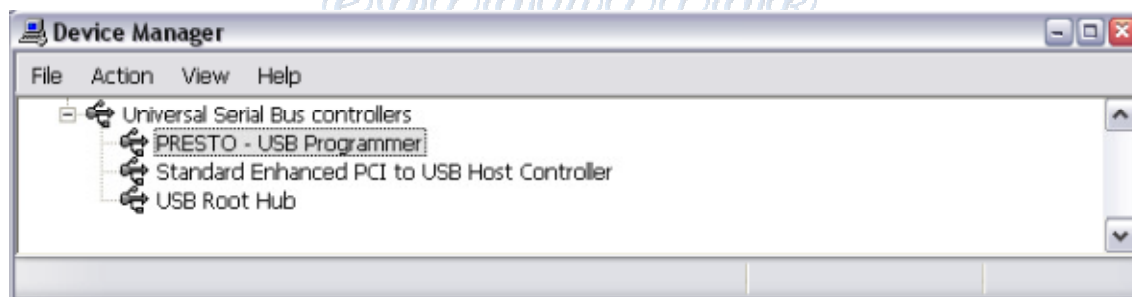
Válaszd a felajánlott automatikus telepítést. Az operációs rendszer a CD-n megtalálja az új drivereket és kéri, hogy hagyd jóvá a telepítését. Kattints a "Tovább" gombra.



Sikeres telepítés esetén a rendszer értesít.



Az eszközkiszolgálóban leellenőrizhető a driver.



Driver telepítés Windows 7 rendszeren

Windows 7 esetén használd a legújabb drivert amit letölthetsz az internetről vagy megtalálsz a PRESTO CD-ROM-ján. Windows 7 rendszeren a driver nem települ fel automatikusan, ezért manuálisan kell elvégezni: Nyisd meg az eszközkiszolgálót és keresd meg a felcsatlakoztatott PRESTO programozót. Kattints rá jobb egérgombbal és válaszd ki az „Illesztőprogram frissítése” menüpontot. Tallózd ki a kicsomagolt drivert tartalmazó mappát.

Ha sikeresen megtörtént a PRESTO driver telepítése, felrakhatod a szükséges szoftvert is. Használhatod pl az alábbiakat:

UP program – mely támogatja a PRESTO programozót és a korábbi ASIX eszközöket is. Ez a program számos funkciót kínál, pl projekt kezelést, parancssoros vezérlést, Windows üzenet kezelést, munkafelület beállítást, testreszabható gyorsbillentyűk, sorozat szám generálást, stb.

JTAG SVF PLAYER – Ez a szoftver JTAG csatolóval rendelkező eszközök esetén használatos

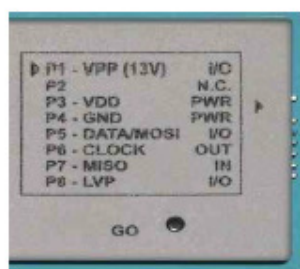
PRECOG – Ez a szoftver eCOG microcontrollerek programozásához használható

ARMINE – ARMINE egy program csomag, mely lehetővé teszi ARM magra épülő microkontrollerek FLASH programozását és hiba keresését. ARMINE csomag egy OpenOCD alapú alkalmazás, mely PRESTO támogatással is rendelkezik, ill. FLASH programozáshoz szükséges kiegészítéseket és egyszerű, de jól használható GUI-t tartalmaz.

Megjegyzés: Ha úgy tűnik, hogy a PRESTO nem működik rendesen, egy letölthető alkalmazás segítségével könnyedén letesztelhető:

http://tools.asix.net/supp_testers.htm

1.4. A programozó csatlakozójának ismertetése



Pin	AVR ³	AVR TPI	8051 arch.	JTAG	eCOG ⁷
P1	Reset	Reset	Reset	USR ⁴	CS
P2					
P3	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵
P4	GND	GND	GND	GND	GND
P5	MOSI	TPIDATA	MOSI	TDI	MOSI
P6	SCK	TPICLK	SCK	TCK	CLK
P7	MISO		MISO	TDO	MISO
P8				TMS	LOADB ⁸

Pin	PIC ICSP	MSP430	MSP430 SBW	CCxxxx	PSoC	I2C	MicroWire	SPI
P1	V _{PP}	TEST		Reset	XRST		CS	#CS
P2								
P3	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵	V _{CC} ⁵
P4	GND	GND	GND	GND	GND	GND	GND	GND
P5	DATA	TDI	SBWTDIO	Debug data	ISSP_SDATA	SDA	DI	SI
P6	CLK	TCK	SBWTCK	Debug clock	ISSP_SCLK	SCK	CLK	SCK
P7		TDO					DO	SO
P8	LVP ²	TMS					ORG ²	

1 – PRESTO beépített lehúzó ellenállása

2 – ez a láb elfordulhat, hogy nem kerül csatlakoztatásra ha az eszköz bekötés úgy kívánja meg

3 – Kristály oszcillátor szükséges ha nincs más órajel forrás

4 – Kiválasztható funkciók TRST, SCK vagy felhasználói

5 – log.0 / Z, Lehúzó ha az áramkör esetén szükséges

6 – log.0 / Z

7 – 32.768 kHz és 5.000 MHz kristály oszcillátor szükséges

8 – Beépített 1 kΩ lehúzó ellenállás

Megjegyzés:

• További információkért nézd át az eszköz bekötési **példákat**.

• Ha a programozandó egység külső tápegységről kap tápot, vagy ha nincs földelve, nagy feszültség különbség lehet a PRESTO földje és a cél eszköz földje között, ami annak meghibásodásához vezethet.

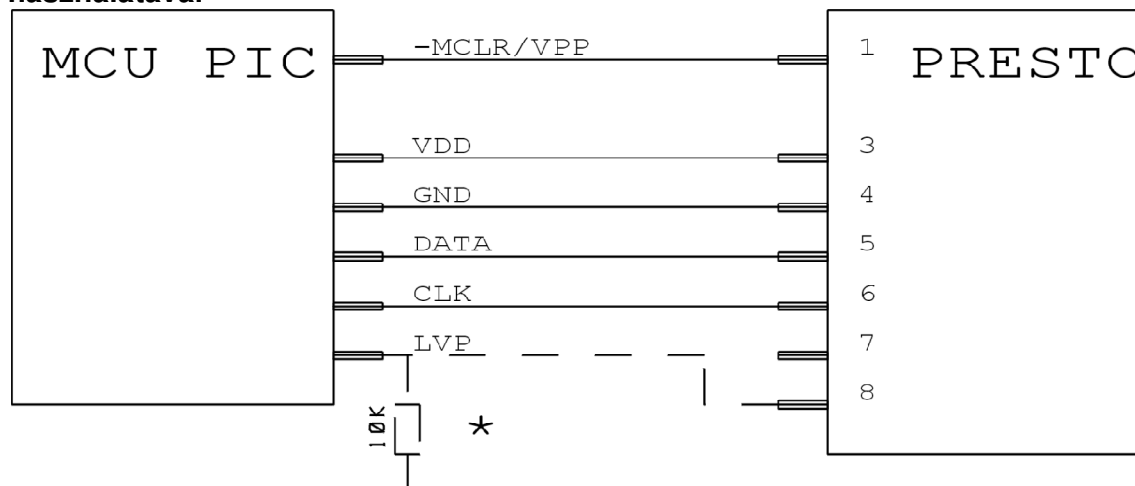
A „helyes” csatlakoztatási mód. Kapcsoljuk össze a PRESTO programozót és a cél eszközt, ezután külső tápegység esetén adjunk tápot a cél eszközre, majd csatlakoztassuk a programozót az USB portra.

A legegyszerűbben úgy kerülhetjük el a föld feszültség különbségből eredő meghibásodást, hogy először a földet csatlakoztatjuk a cél eszközre mielőtt magát a programozót csatlakoztatnánk. – mindig a föld láb legyen az első csatlakoztatott láb. Ha a PRESTO csatlakoztatásra kerül az USB portra, a PC földjére kerül rákötésre.



1.5 Bekötési példák programozáshoz

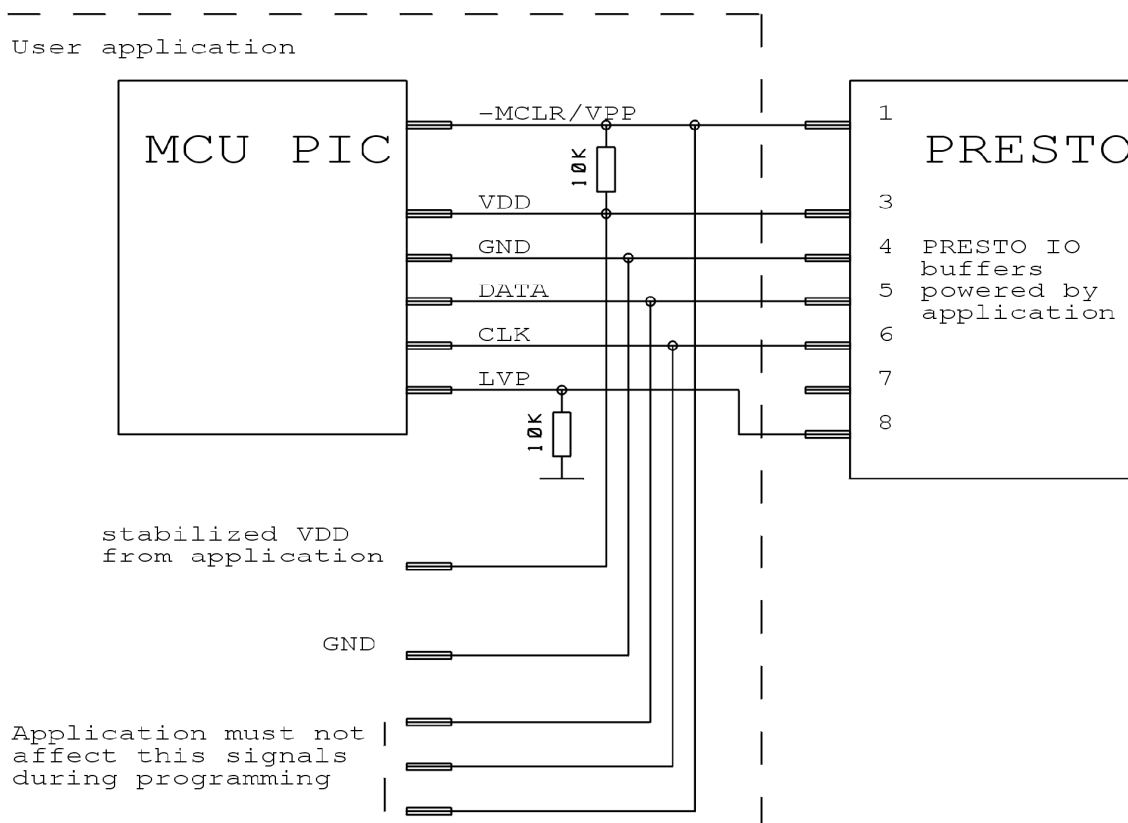
Különálló PIC, beépítés nélkül, magas feszültség (HVP) (13V) mód használatával



* Csatlakoztasd a PRESTO-ra vagy a lehúzó ellenállásra
Lásd a PIC mikrokontrollerekre vonatkozó megjegyzéseket és konfigurációs cím beállításokat.

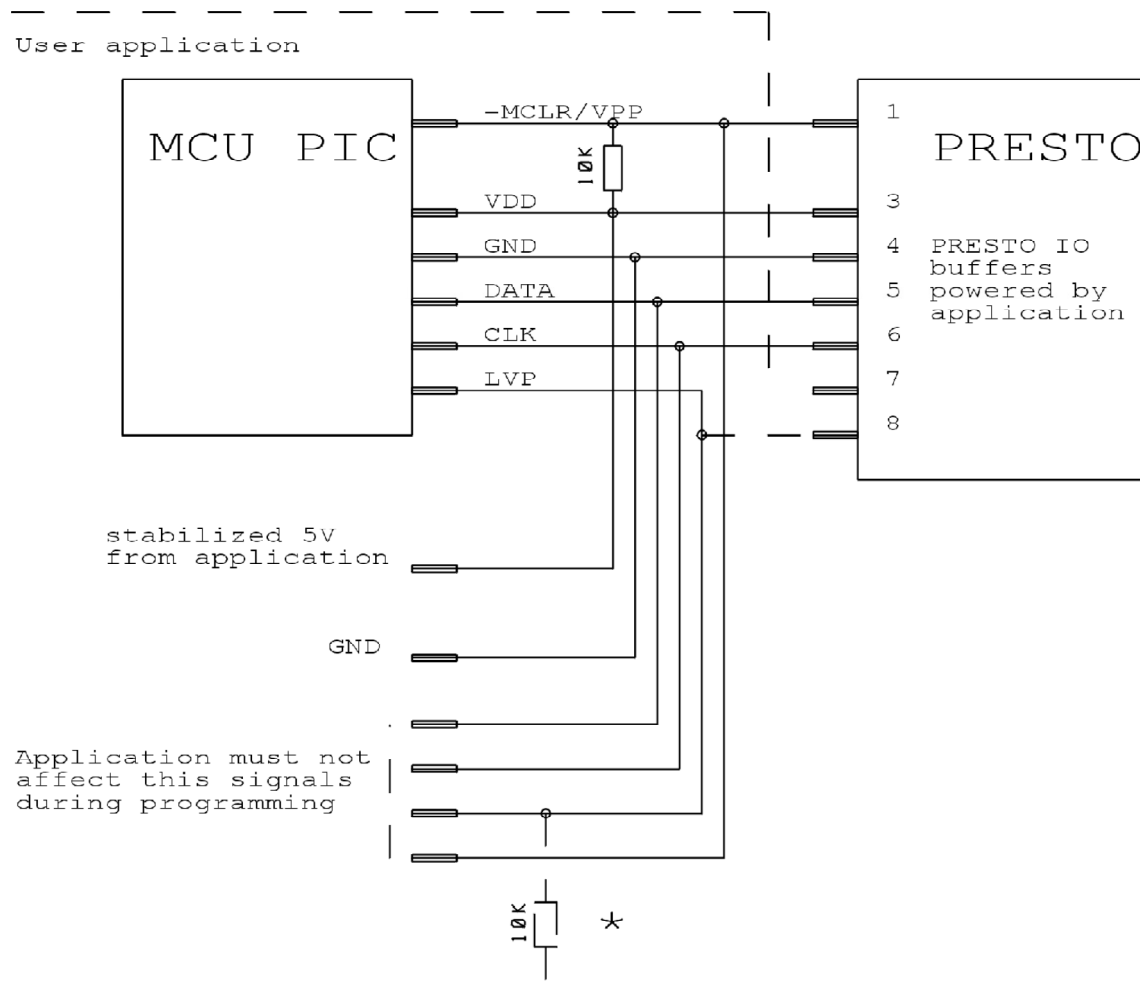
epromírók

Beépített PIC, alacsony feszültségű programozás (LVP (PGM)) mód használatával (nem 13V), tápellátás a cél eszköztől



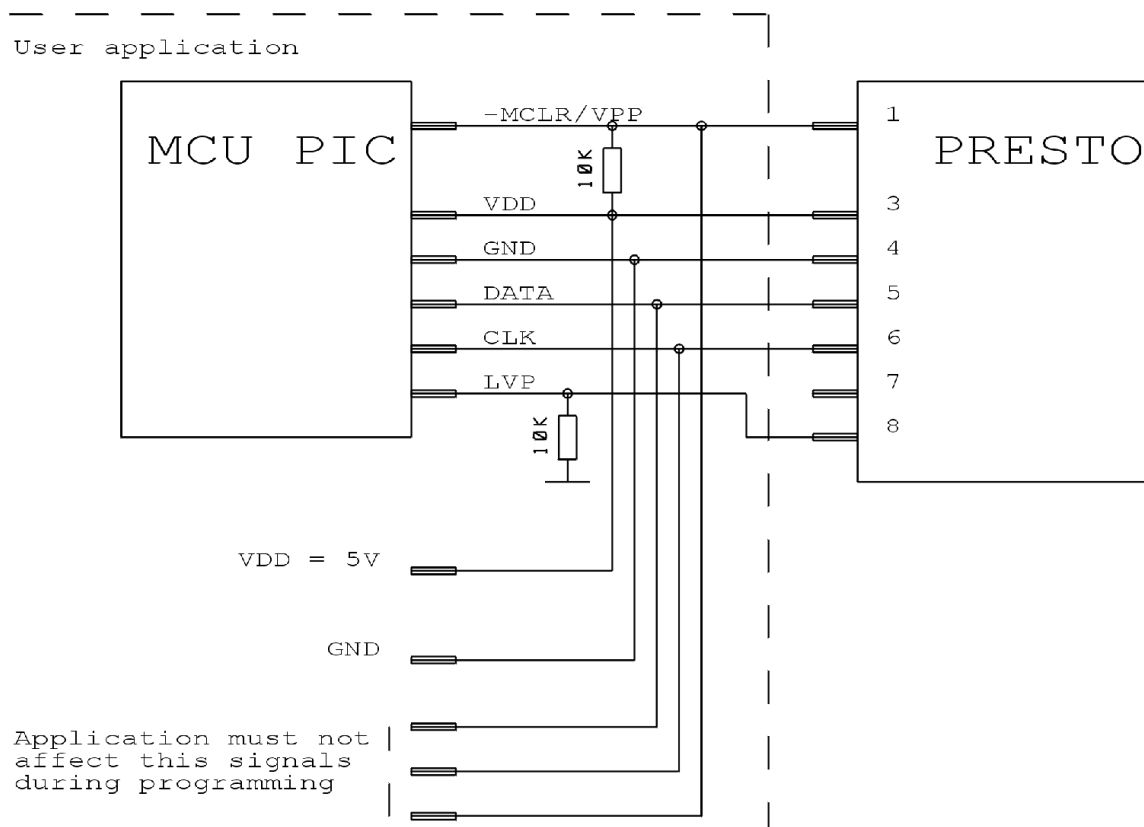
Lásd a PIC mikrokontrollerekre vonatkozó megjegyzéseket és konfigurációs cím beállításokat.

Beépített PIC, magas feszültségű programozás (HVP (13V)) mód használatával, tápellátás a cél eszköztől



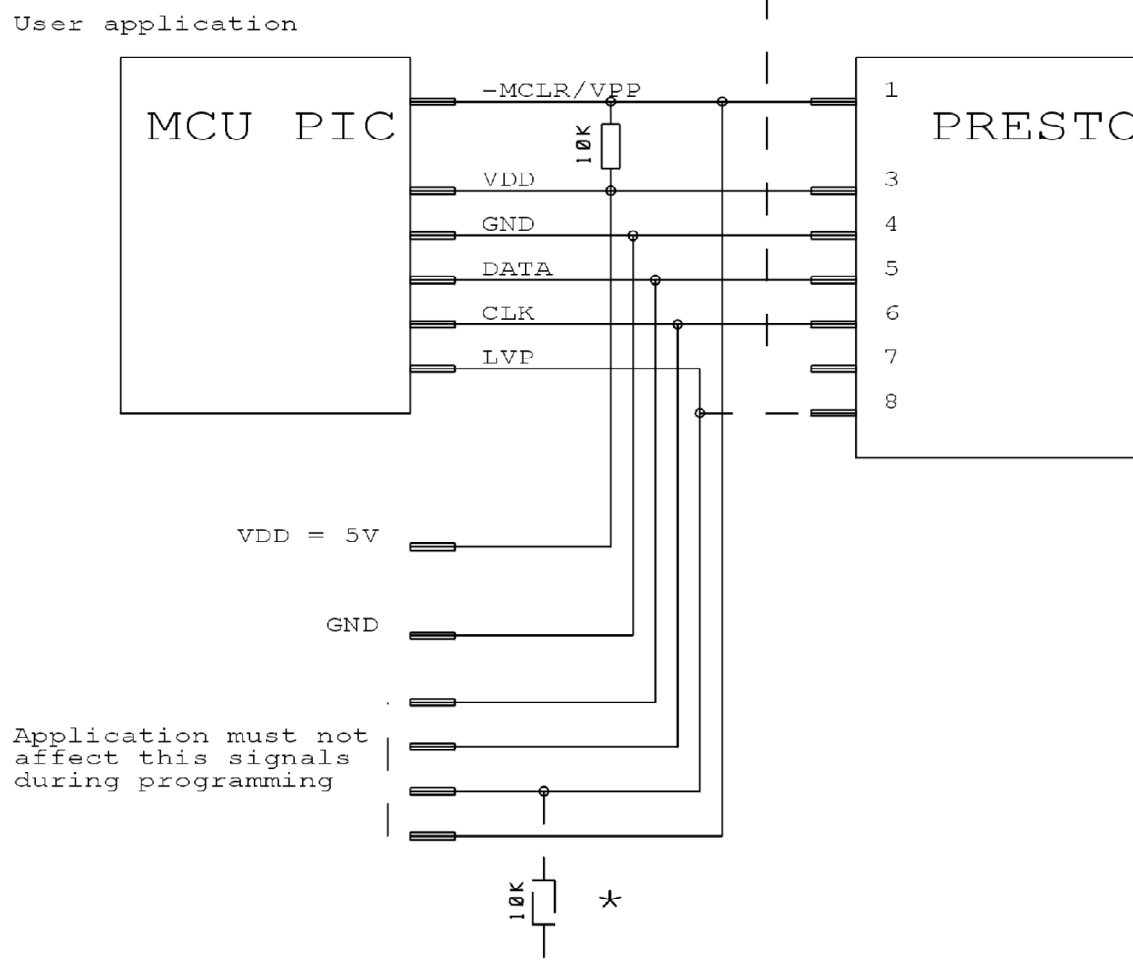
* Csatlakoztasd a PRESTO-ra vagy a lehúzó ellenállásra
Lásd a PIC mikrokontrollerekre vonatkozó megjegyzéseket és konfigurációs cím beállításokat.

Beépített PIC, alacsony feszültségű programozás (LVP (PGM)) mód használatával (nem 13V), tápellátás a PRESTO-ról



Lásd a PIC mikrokontrollerekre vonatkozó megjegyzéseket és konfigurációs cím beállításokat.

Beépített PIC, nagy feszültségű programozás (HVP (13V)) mód, tápellátás a PRESTO-ról



* Csatlakoztasd a PRESTO-ra vagy a lehúzó ellenállásra

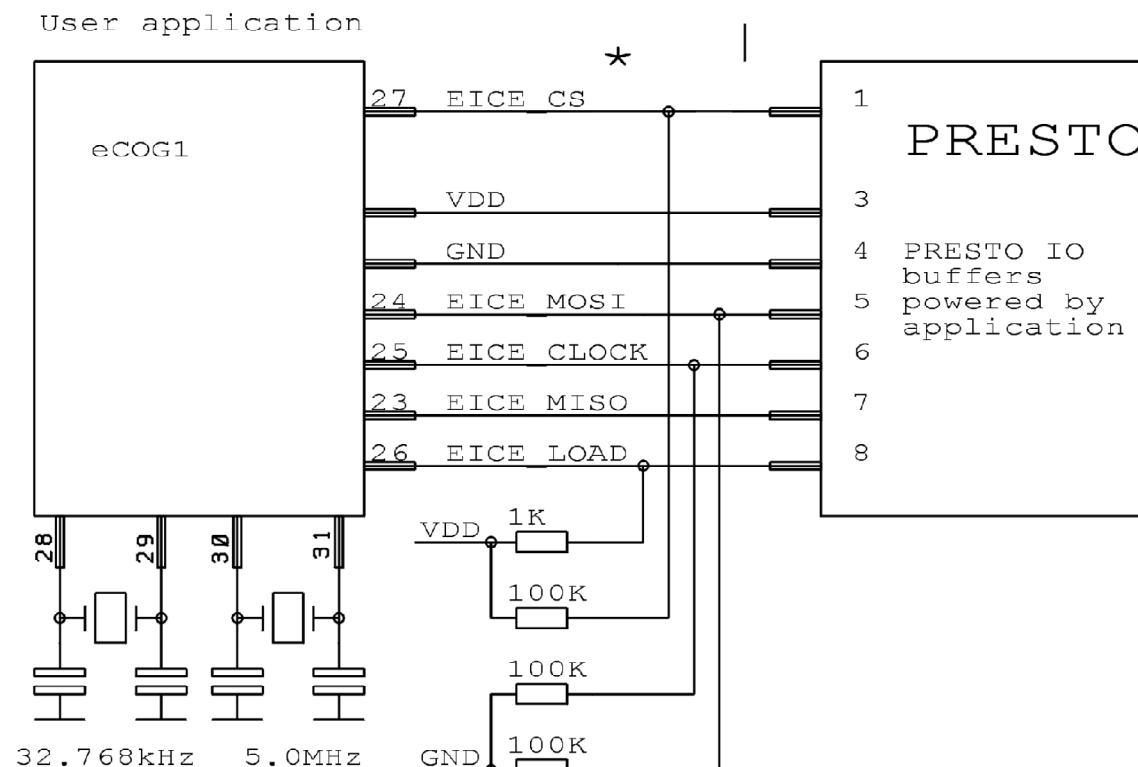
Lásd a PIC mikrokontrollerekre vonatkozó megjegyzéseket és konfigurációs cím beállításokat.

Megjegyzés:

- Ha a programozandó eszköz PIC18FxxJxx microcontroller, az MCLR lábán soha sem lesz 13V, hanem csak 5V.
- Ha a firmware vagy adat CP vagy CPD használatával védett, az eszközt a programozás előtt mindig teljesen törölni kell.
- Az eszköz nem törölhető, ha a CP vagy CPD kisebb mint 5V tápellátású (pl 3.3V).
- Ha a processzor egynél több VDD és GND (föld) lábbal rendelkezik, mindet csatlakoztatni kell, beleértve az AVCC és AGND lábakat is ha rendelkezik ilyennel.
- A cél eszközre a 3V-os tápot külső forrásból kell biztosítani, mert a PRESTO 5V-t képes csak biztosítani.
- Néhány eszköz 13V-nál kisebb maximális feszültséget igényel az MCLR lábra, de a PRESTO 13V-t fog biztosítani. Ha ilyen eszköz kerül kiválasztásakor az UP értesíteni fogja a felhasználót. A felhasználó korlátozhatja a feszültséget egy két ellenállásos feszültségosztóval vagy egy Zener dióda és áram korlátozó ellenállással.

- Magas feszültségű programozás (HVP) módban a microcontroller törlése közben egy LVP fuse (alacsony feszültségű programozás fusebit) is törölhető. Alacsony feszültségű programozás módban történő programozáshoz a felhasználónak az LVP fuse bitet HVP módban kell ismét beállítania.
- PIC32MX eszközök ICSP (áramkörön történő programozás) csatlón keresztül programozhatók egy külső 3V-os táppal.
- ICPORT fuse bittel rendelkező eszközök esetén a dedikált ICSP portot le kell tiltani az LVP programozáshoz.
- PIC24 és dsPIC33 esetén lehetséges programozási módot választani a PE rublika segítségével. A PE jelentése "Programozási parancs használat". Ez a mód gyorsabb mint a többi általános programozási mód.
- Néha ha a PRESTO programozó olyan új PIC-el kerül használatra melyet HVP módban programoztak, egy túláram hibaüzenet figyelmeztet, hogy a programozási feszültségen túláram jelenhet meg. Ha a chip bekötésekor nem történt komolyabb hiba, ez a Microchip új gyártási technológiájának köszönhető. Az új gyártású chippek másként viselkednek mint a régebbiek, még ha ugyanannak a régebbi chip családnak is a tagja. Ebben az esetben a megoldás, hogy egy 1 nF kondenzátor kerül a VPP és GND közé. Ha a probléma továbbra is fennáll, sorosan lehet kötni egy 10 Ω ellenállást a VPP jelre.



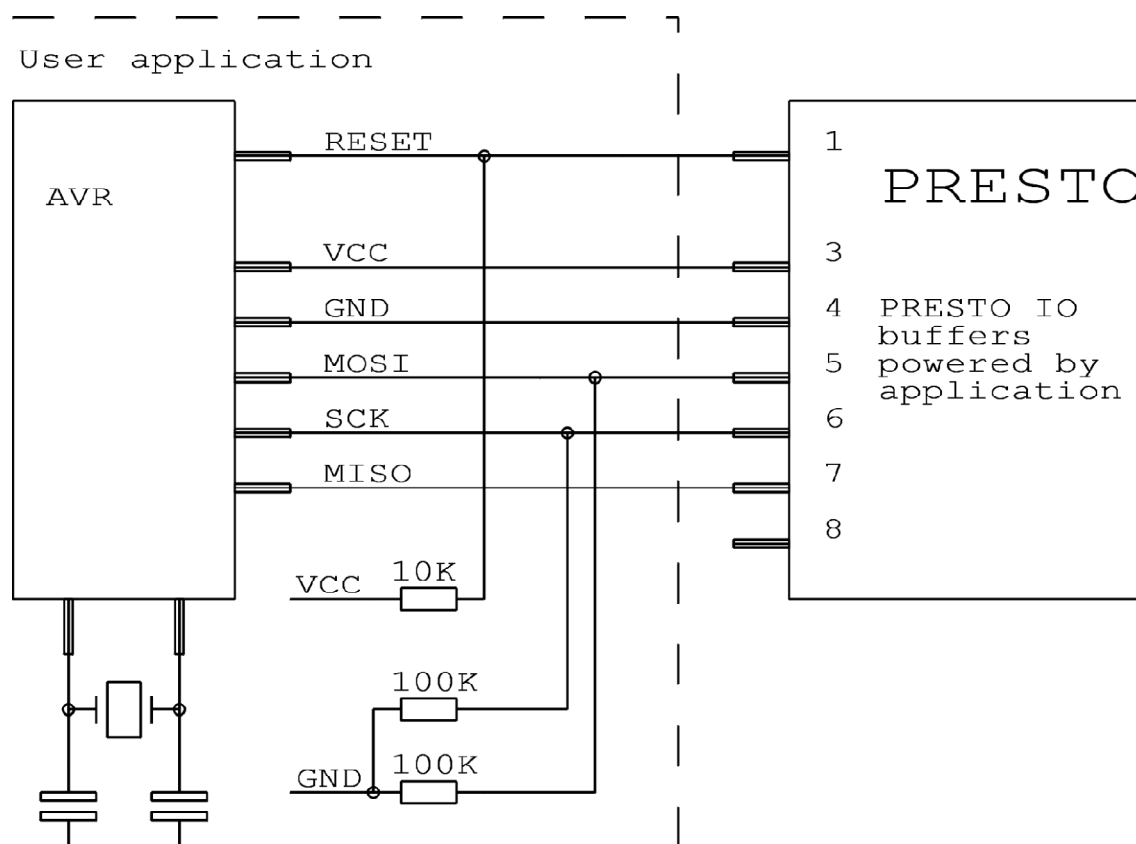
Beépített eCOG, mindig a cél eszköztől kapja a tápot (VDD=3.3V)

* Nem szükséges az EICE_CS a PRESTO-ra történő kötése. Marad log1-ben a programozás alatt.

Megjegyzés:

- VCC 3.3V kell legyen. A tápellátás mindig a cél eszköztől történik.
- Egy eCOG processzor programozásához egy PreCOG alkalmazás tölthető le innen: <http://tools.asix.net/> Megtalálható a CD lemezen is.
- Lásd PreCOGra vonatkozó fejezetet.

Beépített AVR, tápellátás a cél eszköztől

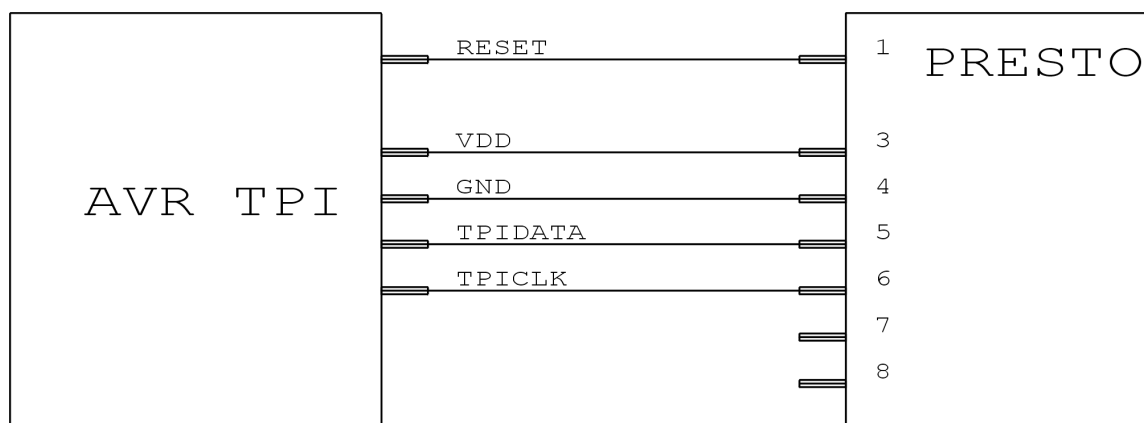


Notes:

- A fuse bitek ATmega és ATtiny esetében alapértelmezésben (gyárilag) belső oszcillátorra vannak állítva. Első alkalomkor az eszköz programozhatóságához az Oszcillátor frekvenciát ">750kHz" vagy kisebb értékre kell beállítani a PRESTO programozó beállítások menüpont alatt. Külső oszcillátor kristály csak abban az esetben szükséges, ha a microprocesszor fusebitje külső oszcillátorra lett állítva programozáskor.
- Nem lehetséges minden AVR microcontrollerhez külső kristályt csatlakoztatni (pl. Attiny13, Attiny15).
- Ha a programozott AVR fuse bitjei megfelelően lettek beállítva, jobb egérgombbal kattints a konfiguráció ablakra és a felugró menüből válaszd ki a „Fuse bitek megismerése” menüpontot. A fuse bit beállítások elmentésre kerülnek a projektben, így ha legközelebb megnyitásra kerül egy .hex fájl, a fuse bitek automatikusan egyformán beállításra kerülnek. (Ha parancssor segítségével kerül programozásra az eszköz, a felhasználónak .ppr fájlt kell megadni a .hex fájl helyett, a fuse bitek elmentéséhez.)
- A **Fájl menüben** használható a **„Hex fájl automatikus megnyitása az adat memóriával”** menüpont. Ez a parancs lehetővé teszi, hogy az EEPROM memória egyszerre kerüljön betöltésre a program fájlal.
- Ha nem kell megváltoztatni az EEPROM memória területét a microcontrollerben, használd az EESAVE fuse-t. Ha ez a fuse bit aktív állapotban van, használd az **Összes programozása EEPROM adat kivételével** parancsot, különben az UP figyelmeztetni fog, hogy az EEPROM memória nem lett törölve.

- Hasznos opció lehet a „**Nincs adat memória törlés annak programozása előtt (csak AVR-nél)**” a **Beállítások -> Program beállítások -> Programozás** menü alatt.
- 3.3V –t igénylő eszközök esetén lehetséges a programozóról történő tápellátás, a **HPR3V3** adapterrel.
- A PRESTO ICSP csatolójának Atmel 10lábú ISP csatlakozóra történő alakításához **HPRAVR** adapter használható.
- Néhány AVR eszköz ISP csatolója más elrendezésű mint az SPI csatoló. Lásd “Soros letöltés” részt az eszköz adatlapján.

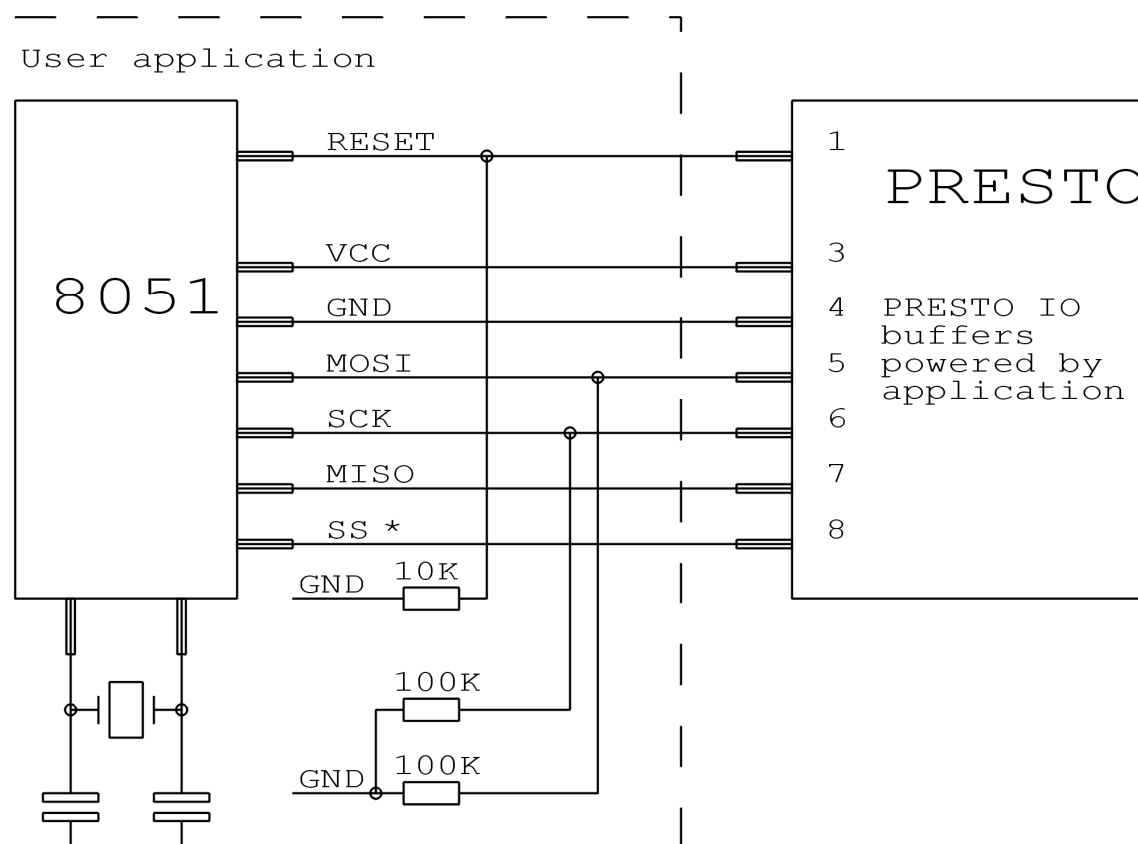


Atmel AVR TPI csatolóval (pl. ATtiny10)**Megjegyzés:**

- A chip RESET lábára 12V rákötés szükséges HVP (magas feszültségű) programozás alatt. A programozó csak 13V-t tud biztosítani, így egy külső áramkör segítségével csökkenteni kell ezt a feszültséget. A chip nem igényel külső kiegészítést, normál LVP (kis feszültségű programozás) mód esetén.

epromírók

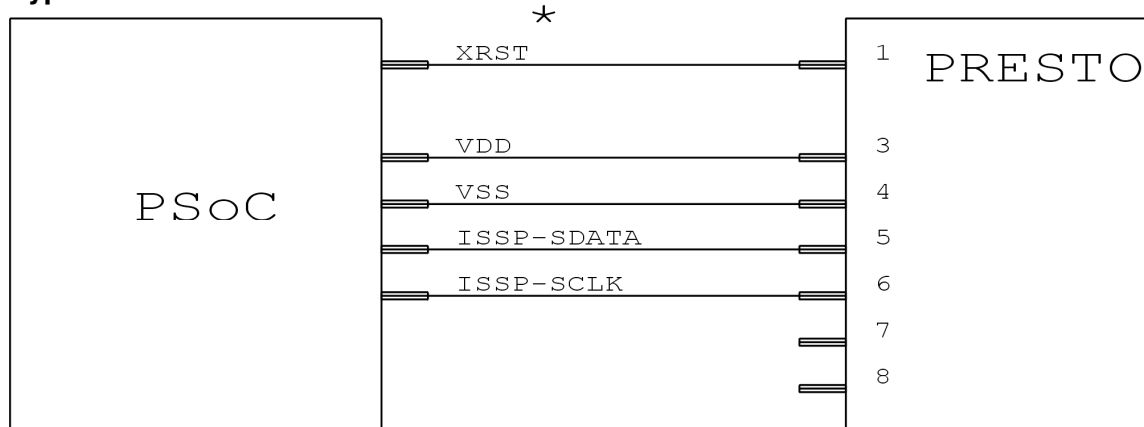
Beépített Atmel 8051 microcontroller, tápellátás a cél eszköztől



* Az SS lábat csak AT89LP2052/4052/213/214/216/428/828/6440/51RD2 esetén kell bekötni.

Megjegyzés:

- AT89LP213, AT89LP214 és AT89LP216 fordított reset logikát használ. A RESET láb ellenállását a VCC lábra kell kötni nem pedig a földre (GND).
- A „C” jelű eszközök nem programozhatók a PRESTO programozóval, de az „S” jelölésűek igen. Illetve néhány „S” típus kompatibilis a „C” típussal. PI AT89C2051 nem támogatott, de az AT89S2051 igen.
- A szoftver azt feltételezi, hogy a POL láb log1-re van kötve AT89LP52 programozása esetén. Ha a POL láb a log0-ra van mégis kötve, akkor a „Fordított RESET” rubrikát ki kell pipálni. AT89LP51RD2 fordított reset logikával rendelkezik, így a szoftver feltételezi, hogy a POL láb a log.0-ra van kötve.

Cypress PSoC eszköz csatlakoztatása

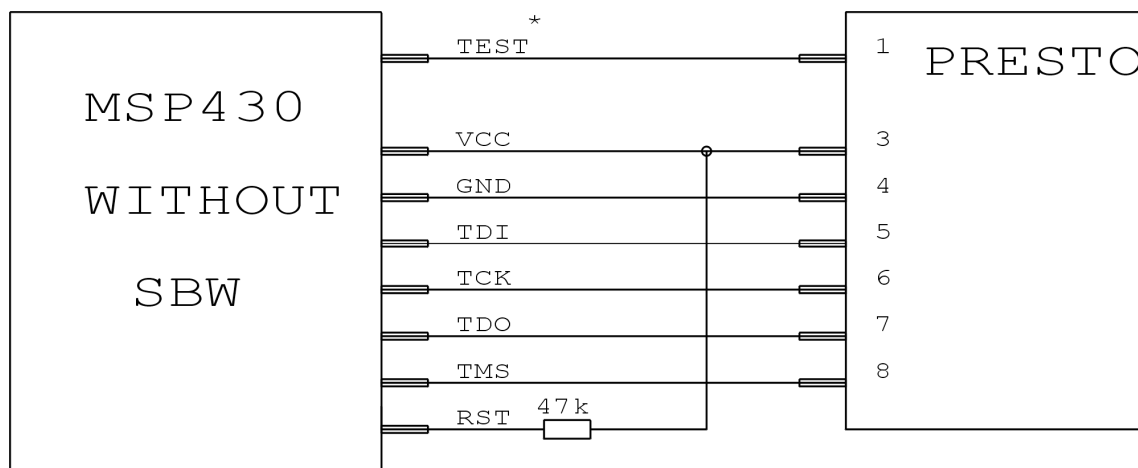
* Nem minden PSoC eszköz rendelkezik XRST lábbal.

Megjegyzés:

- A felhasználónak először a programozási mód „indítási mód”-ját kell kiválasztania, a „PRESTO beállítások” ablakban. XRST lábbal nem rendelkező eszközök esetén csak „power-cycling” mód használható. XRST lábbal rendelkező eszközök esetén mindkét mód használható, de az a jobb, amelyik a reset jel használatával indul (using reset signal), mert az használható külső tápegységgel is.
- A „Programozási algoritmus” a „PRESTO beállítások” ablakban a tápfeszültségének megfelelően legyen kiválasztva.

epromi.rók

MSP430 eszköz csatlakoztatása, ami nem rendelkezik SBW (két eres) csatolóval

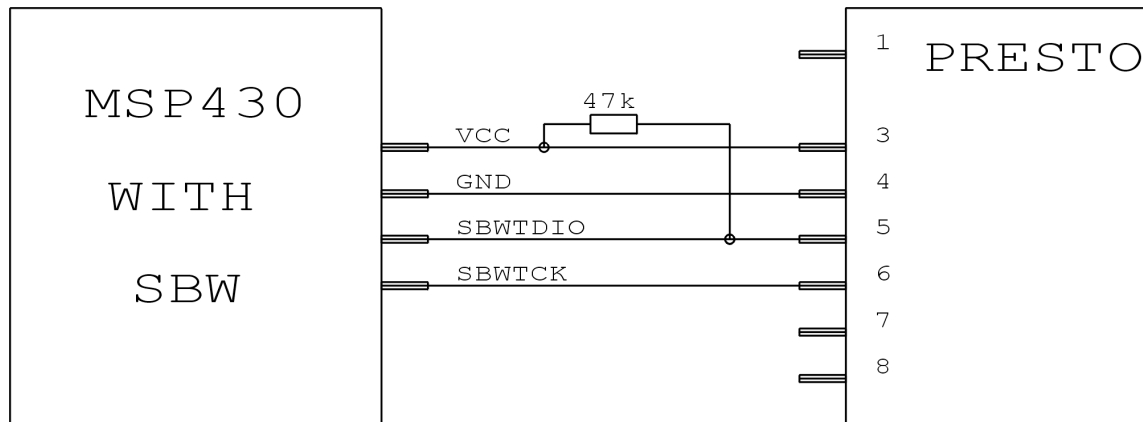


* Nem minden MSP430 eszköz rendelkezik TEST lábbal.

Megjegyzés:

- Ez a csatlakoztatási mód használható pl MSP430F1xx, MSP430F4xx, MSP430F21x1 eszközökkel, de MSP430F20xx vagy MSP430F22xx eszközökkel nem.
- Ha az eszköz rendelkezik Kalibrációs értékkel az információs memóriájában és ezt a memóriát nem szándékozzuk újraprogramozni (törölni), a „**PRESTO beállítások**” ablakban válasszuk a „**Kalibrált belső RC**” oszcillátor beállítást. Egyéb esetben válasszuk a „**Nem kal. belső RC**” oszcillátor beállítást.
- MSP430 processzor esetén a PRESTO programozóval nincs lehetőség fuse biteket állítani.
- Egyedülálló MSP430 processzor (mely nem rendelkezik SBW csatolóval) programozható a PRESTO belső tápjával is a [HPR3V3](#) adapter segítségével.

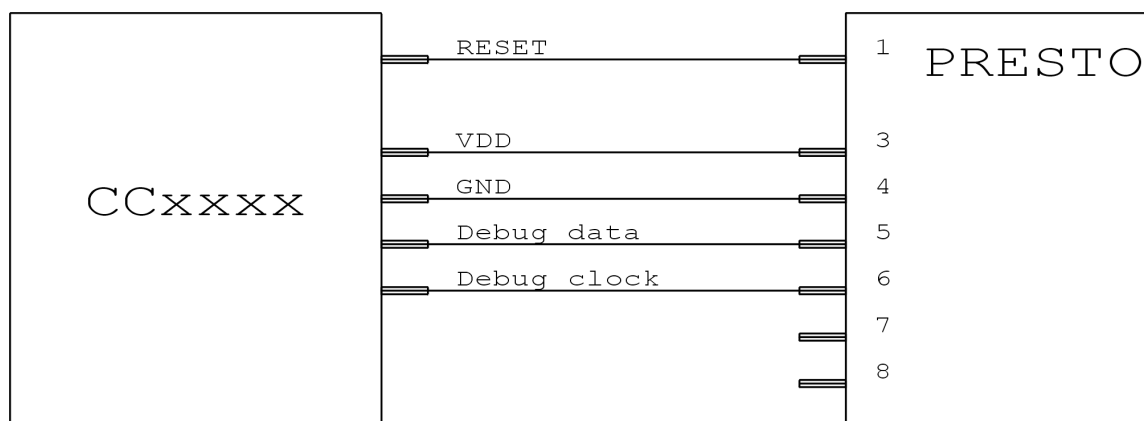
CC430 vagy MSP430 eszközök bekötése melyek rendelkeznek SBW (két eres) csatolóval



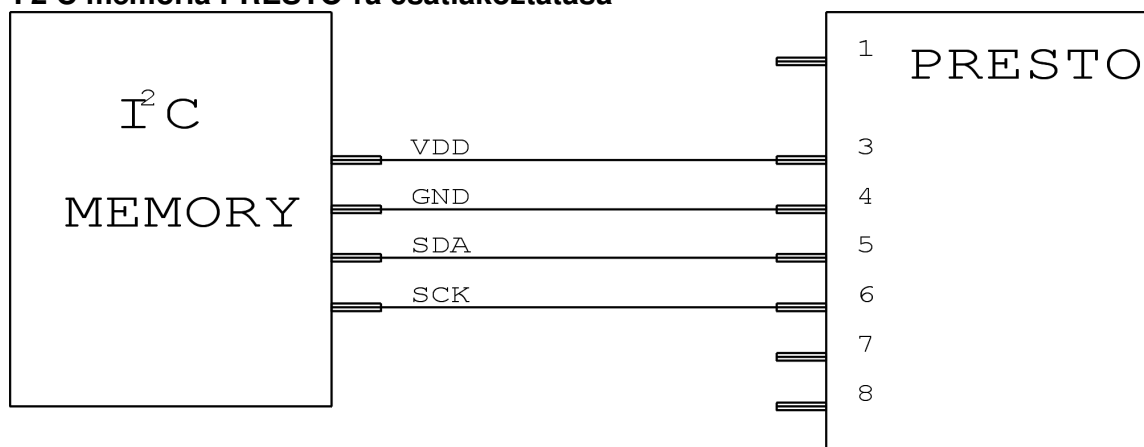
Megjegyzés:

- SBW csatolóval rendelkező eszközök csak ezen a csatolón keresztül programozhatók. Pl. MSP430F20xx, MSP430F22xx vagy MSP430F5xxx.
- Ha az eszköz rendelkezik Kalibrációs értékkel az információs memóriájában és ezt a memóriát nem szándékozzuk újraprogramozni (törölni), a „**PRESTO beállítások**” ablakban válasszuk a „**Kalibrált belső RC**” oszcillátor beállítást. Egyéb esetben válasszuk a „**Nem kal. belső RC**” oszcillátor beállítást. Az oszcillátor típus nem állítható MSP430F5xxx és CC430 esetén.
- MSP430 processzor esetén a PRESTO programozóval nincs lehetőség fuse biteket állítani.
- A „PRESTO beállítások” ablakon található sebesség beállítás hasznos ha kondenzátor van kapcsolva a RESET lábra.
- Azon chipek esetén ahol kalibrációs érték van az információs memóriába mentve, az információs memória törölhető az „A” szegmens nélkül vagy azzal együtt. Ez az „A szegmens törlése” beállítással használható.

Chipcon CCxxxx eszközök bekötése TI-vel



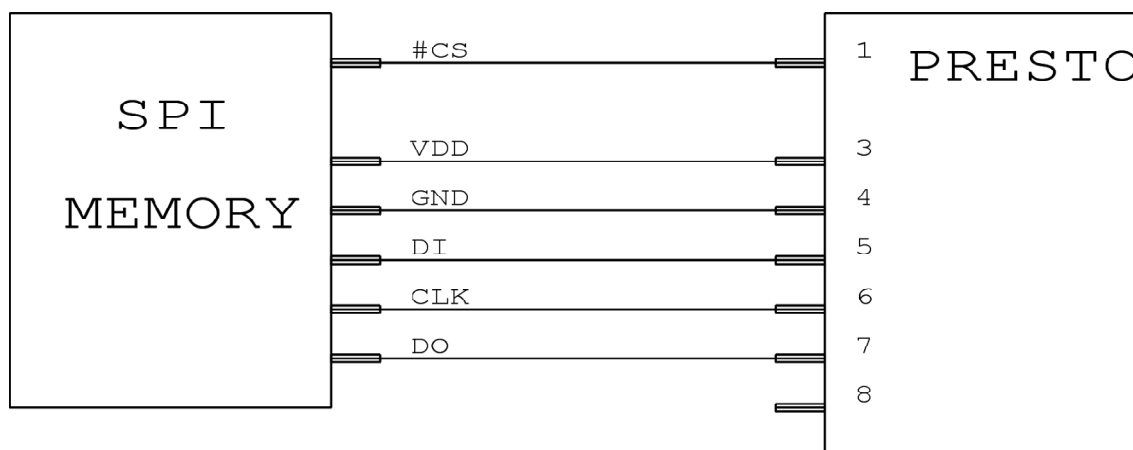
I²C memória PRESTO-ra csatlakoztatása



Megjegyzés:

- A programozó egy belső 2.2kΩ-os lehúzó ellenállást használ az adat jelen (data signal (SDA)) ha I²C buszon dolgozik.
- Ha a programozott eszköz 24LC(S)21A vagy 24LC(S)22A, a VCLK lábat a programozás idejére rá kell kötni a VDD-re.
- 34xx02 nagy feszültséget igényel az A0 lábra SWP és CSWP írás védelem parancs esetén. A nagy feszültség a programozó Vpp lábán kerül előállításra, mely 13V értékű, de a chip kisebb mint 10V-t igényel. A felhasználónak kell a feszültséget a megfelelő értékre csökkentenie. A memória A0, A1 és A2 lábait manuálisan kell beállítani a kiválasztott védelem módnak megfelelően.

SPI memória PRESTO-ra csatlakoztatása

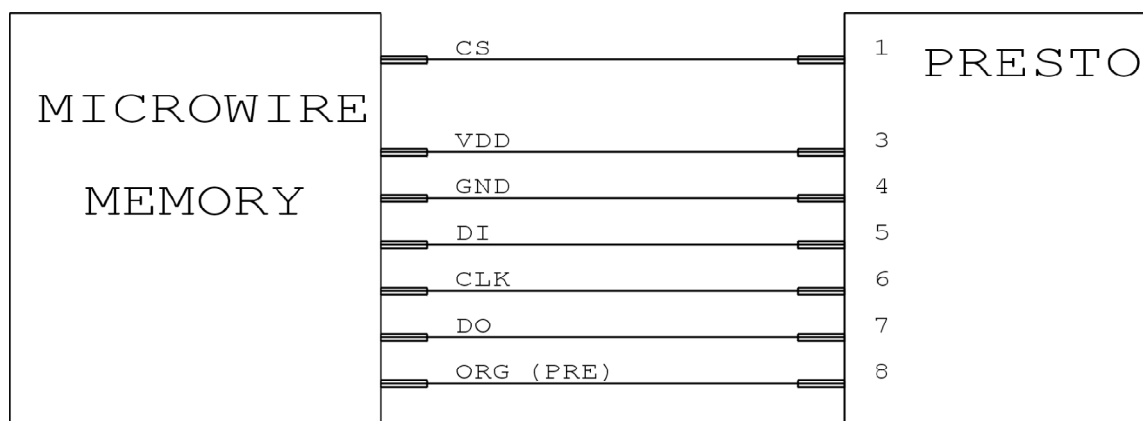


Számos gyártó másképp jelöli az SPI memóriák lábait. Néhány megtalálható az alábbi táblázatban:

Megnevezés képen	a	Atmel, SST	ST
DI		SI	D
DO		SO	Q
CLK		SCK	C
CS		CS, CE	S

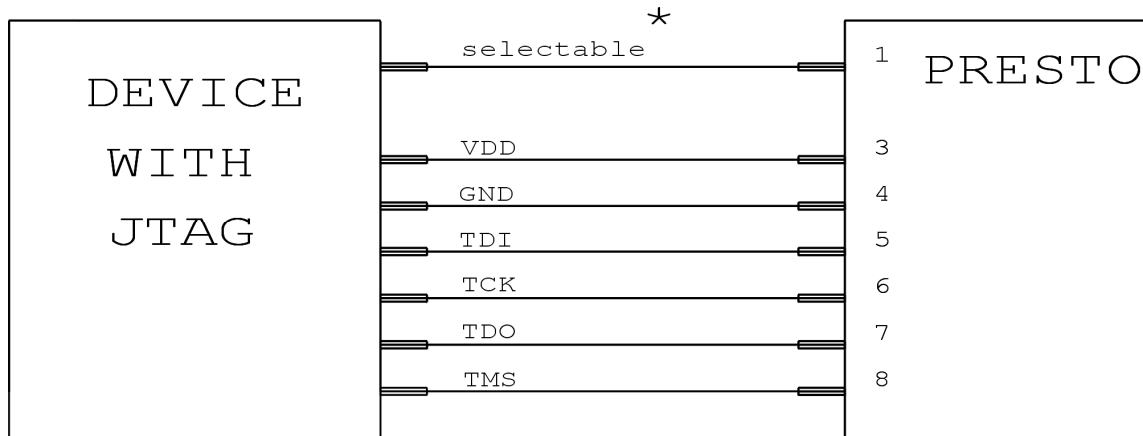
Megjegyzés:

- **Áramkörre épített SPI memória programozás** – A program engedheti és/vagy megtarthatja a lábakat a megfelelő logikai szinten a beépítési helyen. A programozónak a programozás alatt arra van szüksége, hogy minden lábon legyen digitális bemenet vagy legyen leválasztva más áramkörökről multiplexerrel. Ezeknél a memóriáknál a programozó órajel kb 500-1000kHz (az 'AA' eszközök időzírtési igényét kielégítve), így az adatvonalak kapacitív ellenállásával az megfelelő sebességre lassítható.
- 3.3V feszültséget igénylő eszközök esetén lehetséges a programozó által biztosított belső tápellátás, a [HPR3V3](#) adapter segítségével.

Microwire memória PRESTO-ra csatlakoztatása**Megjegyzés:**

- Az **LVP láb** (8 láb) határozza meg a memória szervezést 8-bit vagy 16-bit word egyikén. A felhasználó a PRESTO szoftver segítségével kiválaszthatja a kívánt szervezést, így az ezt a lábat a megfelelő logikai állapotba helyezi. Ha a memória szervezés a hardver áramkörön meghatározásra került, vagy nem lehetséges az LVP láb kiválasztása, ez a láb bekötetlenül marad. Ha programozott eszköz M93Sx6, akkor az LVP lábat össze kell kötni a chip PRE lábával.



PRESTO csatlakoztatása JTAG csatolóval rendelkező eszközre

* A VPP jel viselkedhet SCK-ként vagy TRST-ként is az SVF fájlban meghatározottak szerint, vagy lehet a felhasználó által meghatározott állapotban (log.1, log.0, tristate, különböző a programozás alatt és után)

A programozó P1 lába a RESET lábra kell legyen kötve AVR32 eszközök programozása esetén.

Megjegyzés:

- JTAG esetén a tápellátás mindig a programozandó eszközről történik. JTAG eszközök gyakran használnak 5V-tól eltérő feszültséget.

- Lehetséges az olyan eszközök programozása és tesztelése melyek szoftveres adatai SVF vagy XSVF formátumban vannak.

Az XSVF formátum CPLD Xilinx XC9500 eszközhöz ajánlott csak. Az SVF minden máshoz javasolt. A szoftver PRESTO JTAG csatolón keresztül használatához letölthető innen: <http://tools.asix.net/> és megtalálható a CD lemezen.

- JTAG mód esetén a PRESTO nem biztosít tápot a cél eszközre.

- Ha a JTAG lábak I/O lábként vannak használva, a processzort Resetben kell tartani a programozás alatt.

Ez elvégezhető a PRESTO sárga (VPP) lábának segítségével, mely beprogramozható (JTAG player opciókban), hogy ekként viselkedjen.

- A JTAG csatoló ARM maggal rendelkező processzorok programozására és hiba keresésére is használatos. Az ARM processzorokhoz szükséges **ARMINE** szoftver letölthető a netről. További infókért lásd ARMINE ismertetést.

- A **HPR1V2** adaptert használni kell ha a PRESTO programozóval olyan eszközt akarunk programozni amelyik 2,7V-nál kisebb feszültséget igényel.

- Az AVR32 MCU-k programozhatók JTAG felületen keresztül az UP programmal is. Az alkatrész nem lehet RESET állapotban a programozás alatt.

- Az ATxmega MCU-k programozhatók JTAG felületen keresztül az UP programmal. A P1 láb nem szükséges a programozáshoz. Az ATxmega eszközök a HPR3V3 adapter segítségével kezelhetők.

1.6 A kijelzők és kezelő berendezések ismertetése

Zöld LED (ON-LINE) - PRESTO csatlakoztatva van a számítógéphez

Sárga LED (ACTIVE) – Kommunikáció folyamatban

Gomb (GO) – A programozási folyamat elindítása

1.7 Technikai ismertetés

Figyelmeztetés: Ezen értékek megváltoztatása a programozó vagy a számítógép meghibásodását is okozhatják!

Maximális Vcc feszültség UVCC MAX 7.5 V

Maximális feszültség a többi lábon UIO MAX 5.5 V

Maximális áram felvétel Vcc lábról IVCC MAX 100 mA

Maximális áram felvétel Vpp lábról IVPP MAX 50 mA

Maximális áram felvétel a többi lábról IIO MAX 4 mA

Feszültség ha a tápellátás a cél eszköztől történik UVCC IN 3.0 V to 5.0 V }10%

Feszültség a Vpp lábon programozás alatt UVPP 5 V/13 V

Működési hőmérséklet TOP 0 to 40 °C

Méretek H x SZ x M Kb. 105 x 55 x 25 mm.

ISP kábel maximális megengedett hossza 150mm

2. Adapterek

2.1 HPR3V3

A HPR3V3 egy opcionális kiegészítő a PRESTO programozóhoz 3.3 V-os eszközök kezelhetőségéhez. PI DataFlash Memóriák. A PRESTO programozó 5V-os tápellátást tud biztosítani, azonban néhány eszköz 3,3V-os táp feszültséget és 3,3V-os logikai szinteket igényel a lábain. Ebben az esetben egy HPR3V3 adapter vagy külső tápegység használandó.

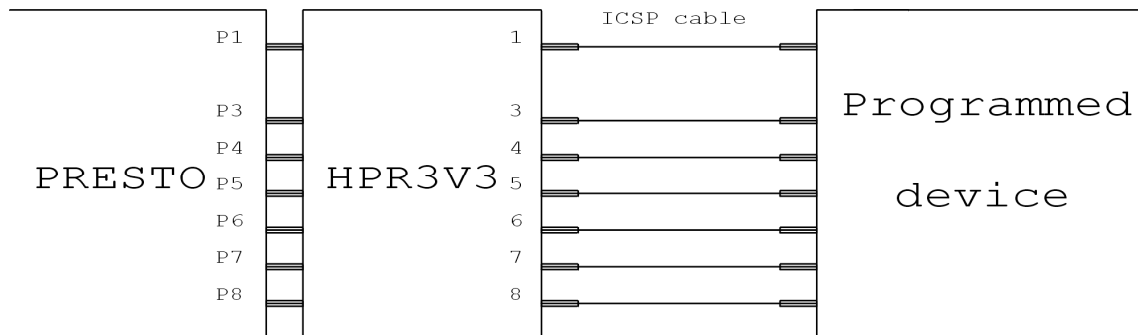
Használat

A HPR3V3 adapter nagyon egyszerűen használható. Csatlakoztasd rá közvetlenül a PRESTO programozóra **(NE A PROGRAMOZANDÓ ESZKÖZ CSATLAKOZÓJÁRA!)**. A 2. láb kulcsként működik, hogy ne lehessen fordítva csatlakoztatni. Ezután csatlakoztasd a programozandó eszközt a HPR3V3-ra az ICSP kábel segítségével. A programozandó eszközre történő csatlakozáshoz az Adapter csatlakozása megegyezik a PRESTO csatlakozójával. Lásd a PRESTO bekötési példákat. Általános bekötési ábra lentebb megtekinthető.

Megjegyzés:

- A HPR3V3 egyirányú adapter, így az adapter használható AVR eszközökhöz, SPI Flash memóriákhoz ill. MSP430 eszközökhöz SBW csatoló nélkül, de PIC eszközökhöz és MSP430 eszközökhöz SBW-vel NEM.

- **Soha se köss külső feszültséget az adapter kimeneti 3.3 V lábára!**



2.2 HPR1V2

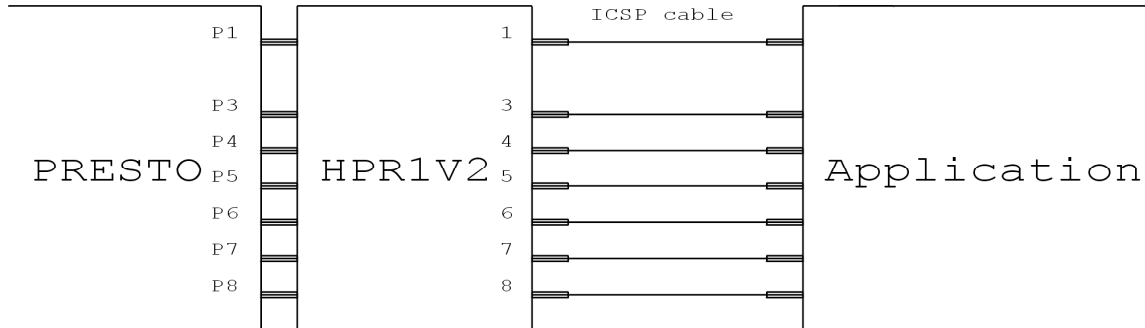
A HPR1V2 egy opcionális kiegészítő a PRESTO programozóhoz, 1,2 V és 3,3V közötti tápot és logikai szintet igénylő eszközök programozásához, pl. Xilinx CoolRunner-II. A PRESTO programozó az eszközöket 3 és 5V } 10 % közötti jelszinten tudja programozni, de néhány eszköz programozásához alacsonyabb jelszint szükséges. Ebben az esetben, a HPR1V2 adapter használandó. Ez az adapter külső, cél eszköztől történő tápellátást igényel. Ebben az esetben a tápellátás nem történhet a programozóról.

Használat

A HPR1V2 adapter nagyon egyszerűen használható. Csatlakoztasd rá közvetlenül a PRESTO programozóra **(NE A PROGRAMOZANDÓ ESZKÖZ CSATLAKOZÓJÁRA!)**. A 2. láb kulcsként működik, hogy ne lehessen fordítva csatlakoztatni. Ezután csatlakoztasd a programozandó eszközt a HPR1V2-re az ICSP kábel segítségével. A programozandó eszközre történő csatlakozáshoz az Adapter csatlakozása megegyezik a PRESTO csatlakozójával. Lásd a PRESTO [bekötési példákat](#). Általános bekötési ábra lentebb megtekinthető.

Megjegyzés:

- A HPR1V2 egyirányú adapter. Használható pl JTAG csatolóval rendelkező eszközökhöz, de olyan eszközök esetén melyek kétirányú lábakat igényelnek pl PIC, NEM.
- A HPR1V2 esetén a programozandó eszköz külső tápellátást igényel.
- **Soha se kapcsolj be a belső, programozóról történő tápellátást ha a HPR1V2 csatlakoztatva van!**

**3. Az UP szoftver**

Az UP szoftver az ASIX programozókhoz készült. Számos funkcióval rendelkezik és részletes felhasználói felügyeletet tesz lehetővé – akár interaktív módon, akár parncssoros mód segítségével távolról vagy Windows üzenetekkel és DLL library segítségével. A szoftver futtatható Windows 95/98/ME/NT/2K/XP/Vista/Win7 alatt.

3.1 Az UP telepítése

A telepítési mód nagyon egyszerű. Futtasd a CD lemezen vagy az interneten is megtalálható telepítőt.

<http://tools.asix.net/> (UP_xxx_EN.EXE, xxx helyett a verzió szám értendő)

A telepítés csupán néhány másodpercet vesz csak igénybe. Az operációs rendszer fájljai közül egy sem kerül módosításra, így nem kell a számítógépet sem újraindítani. A telepítés után azonnal indítható a szoftver (pl. az asztalon található ikon segítségével). Első indításkor az alkalmazás megkér a használandó nyelv, a programozó és a használandó port kiválasztására.

Alapértelmezés szerint angol és cseh nyelv közül lehet választani. Ha a www.epromirok.hu oldalról letölthető magyar nyelvi fájl bemásolásra kerül az UP gyökér mappájába, akkor bármikor kiválasztható később is a magyar nyelv. Alapértelmezett telepítési útvonal: c:\Program Files\ASIX\UP

3.2. Eszköz programozás

Ajánlatos projekteket használni az UP szoftverben. A projekt fájlok tartalmazznak minden beállítást.

Új projekt létrehozható a **Fájl -> Új projekt (File-> New Project)** menü segítségével. Meglévő projekt a **Fájl -> Projekt megnyitás (File-> Open Project)** menü segítségével nyitható meg.

A programozás előtt meg kell határozni a használandó programozót és a programozandó eszköz típusát. Ez elvégezhető a **Beállítások -> Programozó választás (Options-> Select programmer)** és **Eszköz -> Eszköz kiválasztás (Device-> Select device)** menüpontok segítségével, illetve megtehető, ha a bal egérgombbal duplán rákattintasz a Programozó ill. az eszköz nevére a jobb felső sarokban.

A program beállításai a felhasználói igényeknek megfelelően módosíthatók a **Beállítások-> Program beállítások (Options-> Program settings)** menüpont alatt. Ha a PRESTO programozó került kiválasztásra, a PRESTO beállítások ablak jelenik meg, ahol a Tápellátás beállítások és egyéb fontos beállítások végezhetők el.

Programozás : Használd a **Fájl-> Megnyitás (File-> Open)** menüpontot egy .hex fájl megnyitásához. Ha szükséges módosítani a fuse bit beállításokat akkor az könnyedén megtehető a **Konfiguráció (Configuration)** ablakon. A változtatások a **Fájl-> Mentés (File-> Save)** menüvel végezhető el. (Lásd **AVR** és **PIC** programozás.)

A programozás megkezdődik, ha rákattintasz az **Eszköz-> Programoz (Device -> Program)** menüpontra. A programozó az alábbi műveletek fogja elvégezni: Törlés, üresség ellenőrzés, programozás és ellenőrzés. A művelet megkezdése előtt ellenőrizd az eszköz azonosítót és a Kód/adat védelmet.

Ha csak az eszköz egyik részét kell programozni, könnyedén elvégezhető a Programozás menüpont alatti részletezett programozási műveletek egyikével.

Megjegyzés : Ha a „**Minden programozás előtt .hex fájl betöltése**” („load hex file before programming”) menü kipipálásra kerül a **program beállítások** alatt, a szoftver újra betölti a hex fájlt ha megnyomod a Programozás gombot. Ha a fuse bitek nem a hexben vannak tárolva vedd ki innen a pipát vagy válaszd ki a **Program beállítások-> Fájl** menüpont alatt a **Konfig memória törlése fájl beolvasás előtt (erase configuration memory before reading file)** menüpontot. (Ez azt jelenti, hogy azok a fuse bitek melyek nem kerültek a hex fájlban tárolásra alap állapotra kerülnek vissza).

3.3. GO gomb funkció kiválasztása

A PRESTO programozó rendelkezik egy GO gombbal. Ez a gomb lehetővé teszi, hogy a felhasználó egyszerűen, ennek a gombnak a megnyomásával elindíthassa a programozást anélkül, hogy a számítógéphez érne. A LED-ek mutatják a programozó állapotát - zöld LED (ON-LINE) *informál az USB kapcsolatról* a sárga LED (ACTIVE) *mutatja, hogy működik az eszköz* (programoz, olvas, stb...)

A gomb funkciója megváltoztatható a felhasználói igényeknek megfelelően a **Beállítások-> Gyorsbillentyűk (Options□Key shortcuts)** menü segítségével.

Az UP programnak futnia kell ha a GO gomb használhatóságához.

3.4. Sorozat gyártás

Menü: **Eszköz -> Programoz-> Sorozat gyártás (Device-> Program-> Mass production)**

Ez a funkció elérhető az eszközpanelon található „Programozás” gomb alatt is.

A megnyíló sorozat gyártás ablakban a „Programoz” gombra kattintva indítható el a művelet, ami megfelel az „Összes programozása” vagy az „Összes programozása EEPROM nélkül” parancsoknak, attól függően, hogy a „Ne programozd az adat EEPROM-ot” kipipálásra került-e.

A számláló is ezen az ablakon található meg. A beállítások menüben bekapcsolható, hogy ez is megjelenjen az állapot kijelzőn. A számláló a sorozat gyártás módban és normál módon programozott eszközök teljes mennyiségét mutatja.

3.5. Sorozat számok

A **"Sorozat szám"** funkció egy sorozat számot vagy más adat részt programoz a megadott memóriaterületre a programozás előtt vagy után.

A sorozat szám lehet:

- *automatikus*

Az automatikusan létrehozott számok mindig az eszköz memóriaterületére kerülnek. pl lehet kód memória, EEPROM memória vagy azonosító hely. A szám lehet decimális vagy hexadecimális.

- *kiolvasás fájlból*

Ebben az esetben a szám számos memória terület között szétdarabolódhat (pl. a sorozat szám maga a kód memóriában van, az eszköz címek pedig az EEPROM-ban és ugyanez a sorozat szám az azonosító területen, hogy védett eszköz esetén is kiolvasható legyen).

Sorozat szám fájl formátuma

A sorozat szám fájl egy txt fájl, ami könnyedén létrehozható bármely szerkesztő program segítségével. Az ajánlott kiterjesztés .SN vagy .TXT.

- *Üres hely* a szóköz, tab vagy a sor vége (CR+LF).
- *Kommentár* bármely sor mely pontosvesszővel ';' kezdődik.
- *Sorozat szám rekord* egy szöveg
címke: adat rekord, adat rekord, ..., adat rekord; *Kommentár*
- *Címke* egy szöveg ami azonosítja a sorozat számot. A címke használata kötelező.
- *Adat rekord* címek (hexadecimális) és adat mezők (hexadecimális) összessége, ami programozásra kerül a memória megadott kezdő címére, pl. 2100 05 55 54 hatására az EEPROM 05h, 55h, 54h értékeket fog tartalmazni a 00-tól 02 címig.

Meghatározható az a memória ahova a sorozat szám mentésre kerüljön a CODE. vagy PROG. vagy P. (program memóriához) vagy DATA. vagy EE. vagy E. (adat memóriához) vagy ID. vagy I. (azonosító helyhez) használatával. Ezeket követik a megadott memória cím értékei.

pl. EE.00 05 55 54 hatására az EEPROM 05h, 55h, 54h értékeket fog tartalmazni a 00-tól 02 címig.

Megjegyzés:

A fuse bitek helye nem adható meg.

Sorozat szám példa:

; a kezdésre vonatkozó kommentár

sn1: 0000 34 45 56 67,

2100 01 02 03 04; ez az 1. sorozat szám

sn2: 0000 45 56 67 78, 2100 02 02 03 04;

sn3: 0000 56 67 78 89, 2100 03 02 03 04;

; megjegyzés

sn4: 0000 67 78 89 9A, 2100 04 02 03 04;

sn5: 0000 78 89 9A AB, 2100 05 02 03 04;

sn6: 0000 78 89 9A AB, 2100 06 02 03 04;
 sn7: 0000 78 89 9A AB, 2100 07 02 03 04;
 sn8: code.0001 3F00 3F01 3F02 3F03, data.0002 'x' '4' '2';
 sn9: prog.0001 3F00 3F01 3F02 3F03, e.0002 'x' '4' '3';

3.6. Az UP használata parancssorból

A program futtatható parancssoros módban is.

```
up.exe [{/ask | /q}] [{/e eeprom_file.hex | /noe}] [{/p | /pdiff} | /o]
file.hex | file.ppr] [/part part_name] [/erase] [/w[nd] up_window_class] [/cfg]
[/devid] [/blank] [/verify file] [/s programmer_SN]
```

Részletes parancs ismertetést lásd az angol kézikönyvben.

3.7. UP kezelés Windows üzenetekkel

Az UP kezelhető Windows üzenetek használatával is.

Részletes ismertetést lásd az angol kézikönyvben.

3.8 UP futtatása több példányban

Ha a felhasználó több PRESTO programozót is szeretne ugyanahhoz a számítógéphez csatlakoztatni, akkor annyi példányban kell futtatni az UP szoftvert ahány programozót a gépre kíván csatlakoztatni.

Alap esetben csak egy példányban futhat a szoftver. Ha egy újabb példányt szeretnénk futtatni akkor parancssorból indítva egyedi ablak osztály nevet kell annak adni. Ablak osztály név megadható a parancssorban a /w paraméter használatával.

Az első példány elindítható a startmenüből.

A többi példány parancssorból. pl. up /w "another up"

3.9 Programozó elérése több alkalmazással

Egyszerre csak egy alkalmazás tudja kezelni a programozót. Az operációs rendszer nem engedélyezi, hogy egy programozót egy időben több alkalmazás is elérje, így az UP szoftver sem engedi, hogy a PRESTO programozót más szoftver elérje, de lehetséges az UP alkalmazást kényszeríteni, hogy engedje át a készüléket más alkalmazásnak. Amíg a programozó választás ablak megjelenik, a programozó elérhető más alkalmazások számára is, úgy hogy azt nem kell bezárni és nem történik adatvesztés.

3.10 Kalibrációs memória támogatása

A kalibrációs információk kezelése UV EPROM törlő használata esetén

Az alkatrész törlése előtt fontos a kalibrációs információ mentése. Ez a funkció elérhető a "Kalibrációs adat mentése..." menüpont segítségével.

Menü: Fájl -> Kalibrációs adat mentése...

Menü: Fájl -> Kalibrációs adat kiolvasása...

A program felkínálja az alkatrész üresség ellenőrzését is. Ha ez a funkció elindításra kerül, a kalibrációs memória megjelenítésre kerül.

A kalibrációs információk kezelése flash memóriával rendelkező alkatrészek esetén

A kalibrációs memória tartalma megőrzésre kerül egy egyszerű törlés esetén.

Ha mégis le szeretnénk törölni, akkor az „Eszköz teljes törlése (kalibrációs értékekkel)” parancs használandó, mely a törlés gomb alatt található meg.

Figyelmeztetés: Új flash memóriával rendelkező alkatrészek esetén (pl. PIC12F629) a bandgap bitek is a kalibrációs memória részét képezik. Így ezek a bitek is törlésre kerülnek a kalibrációs memóriával együtt az "Eszköz teljes törlése (kalibrációs értékekkel)" használatával.

4. Az alkalmazás menüi

A menü az alábbi részeket tartalmazza:

- Fájl
- Szerkeszt
- Nézet
- Eszköz
- Beállítások
- Súgó
- PRESTO beállítások ablak
- HEX editor ablakok



4.1. Fájl menü

Fájl -> Új (File -> New)

Gyorsbillentyű: Ctrl+N

Új üres fájlt hoz létre. Ha az aktuális fájl még nem lett elmentve a program megkér annak mentésére.

Fájl -> Megnyitás... (File -> Open...)

Gyorsbillentyű: Ctrl+O

Megnyit egy fájlt a normál Windows tallózó segítségével.

Fájl -> Következő fájl megnyitása (File -> Open next file)

A megadható eltolással megnyitja a következő HEX vagy BIN fájlt. Ez a funkció akkor hasznos ha egy második fájlt be kell tölteni pl a chip memóriába.

Fájl -> Aktuális fájl újra betöltése (File -> Reload...)

Gyorsbillentyű: Ctrl+R

Az éppen megnyitott fájlt ismét megnyitja a merevlemezről. Ez a funkció akkor hasznos, ha egy másik alkalmazás által frissítésre került fájl szeretnénk megnyitni.

Ha az „Aktuális .hex fájlhoz automatikus újabb verzió keresés” ("Check for HEX file updates") be lett kapcsolva a Beállítások -> Fájlok fül alatt, akkor a felhasználót meg fogja kérdezni a szoftver, hogy be akarja-e tölteni a frissített fájlt.

Fájl -> Mentés (File -> Save)

Gyorsbillentyű: Ctrl+S

Elmenti a fájlt a merevlemezre. Ha más néven szeretnéd elmenteni, akkor használd a *Mentés másként* menüpontot. A nem használt memória területek átlépésre kerülnek mentéskor.

Fájl -> Mentés másként (File -> Save as...)

Más néven menthető el a fájl. A nem használt memória területek átlépésre kerülnek mentéskor.

Fájl -> EEPROM adat importálása fájlból (File • Import data memory...)

EEPROM memória tartalom importálható be egy különálló fájlból. A fájl a nulla címtől kerül kiolvasásra függetlenül annak tartalmától. Egy compiler által létrehozott fájl nem importálható be ezen a módon.

Ez a funkció a régebbi szoftverrel való kompatibilitást hivatott megoldani, ha egy régebbi szoftverrel történt az EEPROM tartalom külön fájlba mentése. Ez a funkció hamarosan elveszti létjogosultságát mert a mostani alkalmazások egy fájlba mentik az összes memória tartalmát.

Fájl -> Hex fájl automatikus megnyitása az adat memóriával (File -> Open hex file with data automatically)

Ha ez az opció kiválasztásra kerül, az UP szoftver automatikusan betölti a hex fájlt az adat memóriába a kód memóriával együtt. Ez az opció csak akkor aktív ha az adatmemóriának a különálló fájl megnyitásra került.

Fájl -> Új projekt (File -> New project)

Gyorsbillentyű: Shift+Ctrl+N

Új projekt létrehozása. A projektek használata akkor ajánlott különösen, ha sokféle alkatrészt kell programozni. A projekt fájl tartalmazza az összes szükséges beállítást.

Fájl -> Projekt megnyitás (File -> Open project)

Gyorsbillentyű: Shift+Ctrl+O

Egy már létező projekt fájl megnyitása. Ha egy fájl is meg volt nyitva a projekttel, megnyitásra kerül az is.

Fájl -> Projekt mentés (File • Save project)

Gyorsbillentyű: Shift+Ctrl+S

Az elkészített projektet menti el. A mentést új néven is megtehetjük. Ha automatikusan azonos néven szeretnénk elmenteni, akkor a program beállításokban ez megadható.

Fájl -> Projekt bezárás (File -> Close project)

Gyorsbillentyű: Shift+Ctrl+W

Az éppen megnyitott projektet elmenti és bezárja. A szoftver visszaáll a megnyitása előtti állapotra.

Fájl -> Aktuális projektek (File -> Recent projects)

Itt tekinthető meg az utolsó 5 használt projekt. Ha az egyikre rákattintasz az adott projekt megnyílik.

Fájl -> Kalibrációs adat kiolvasása (File • Read calibration information...)

Megnyit egy fájlt és kiolvassa annak kalibrációs információit

Fájl -> Kalibrációs adat mentése (File • Save calibration information...)

Elment egy fájl a kiolvasott kalibrációs információkkal. Ha az alkatrész törlésre került, lehetséges betölteni a kalibrációs információt a Kalibrációs adat kiolvasása parancs segítségével.

Fájl -> Exportálás bin formátumra (File -> Export to binary...)

Létrehoz egy nyers bináris adatot az EEPROM memóriájáról. Az adat szervezés lehet 8 vagy 16 bites is.

Fájl -> Kilépés (File -> Exit)

Gyorsbillentyű: Alt+F4 vagy: Alt+X

Kilépés a programból. Ha az éppen megnyitott fájl megváltozott a program megér annak mentésére.

4.2. Szerkeszt menü (Edit Menu)

Szerkeszt -> Értékkal kitölt (Edit -> Fill with value...)

A kiválasztott memória területet a megadott értékkel kitölti. Ezzel a funkcióval általában üresre vagy nullára állítható a kiválasztott terület, de más esetekben is lehet rá szükség. A memória típus előre kiválasztásra kerül az aktív abak számára. Ha a memória terület kiválasztása már megtörtént, akkor az automatikusan hozzárendelésre kerül a kitöltési művelethez. Egy memória terület kiválasztható a Shift bill nyomva tartásával és az egér mutató használatával (kattintással)

Szerkeszt -> Szöveg beszúrás (Edit -> Insert text...)

Egy ASCII kódolású szöveg illeszthető a memóriába. A sor végi karakter lehet kódolva mint NULL, CR, LF vagy CR+LF.

4.3. Nézet menü

Nézet -> Kód memória (View -> Code memory)

Láthatóvá teszi vagy elrejt a kód memória szerkesztőt.

Nézet -> EEPROM memória (View -> EEPROM memory)

Láthatóvá teszi vagy elrejt az EEPROM memória szerkesztőt.

Nézet -> Konfigurációs memória (View -> Configuration memory)

Láthatóvá teszi vagy elrejt a konfigurációs memória szerkesztőt.

Nézet -> Kód memória megjelenítés (View -> Show code memory)

Gyorsbillentyű: Alt+F10

Megjeleníti a kód memória szerkesztőt és előtérbe helyezi.

Nézet -> EEPROM memória megjelenítés (View -> Show EEPROM memory)

Gyorsbillentyű: Alt+F11

Megjeleníti az EEPROM memória szerkesztőt és előtérbe helyezi.

Nézet -> Konfigurációs memória megjelenítés (View -> Show configuration memory)

Gyorsbillentyű: Alt+F12

Megjeleníti a konfigurációs memória szerkesztőt és előtérbe helyezi.

4.4. Eszköz menü (Device menu)**Eszköz -> Programoz (Device · Program)**

Gyorsbillentyű: Shift+F5

· **Összes programozása (Program All)**

Gyorsbillentyű: F5

Töröl, ürességet ellenőriz, programoz és ellenőriz a teljes alkatrészen. A folyamat elindítása előtt azonosító ellenőrzést végez, és a kód/ adat védelem végrehajtásra kerül.

· **Összes programozása EEPROM adat kivételével (Program all but EEPROM)**

A gyorsbillentyű a beállítások alatt meghatározható.

A művelet teljesen megegyezik az előzővel, csak itt nem történik törlés, programozás, ellenőrzés az EEPROM memória területen. Ez a funkció olyan eszköz esetén nem érhető el amelyik nem rendelkezik EEPROM memóriával, de ehelyett használható az Összes programozása menüpont

· **Kód memória programozása (Program code memory)**

A gyorsbillentyű a beállítások alatt meghatározható.

Törli, ellenőrzi az ürességét, programozza és ellenőrzi a kód memóriát.

· **EEPROM memória programozása (Program EEPROM memory)**

A gyorsbillentyű a beállítások alatt meghatározható.

Törli, ellenőrzi az ürességét, programozza és ellenőrzi az EEPROM memóriát.

· **Konfigurációs memória programozása (Program configuration memory)**

A gyorsbillentyű a beállítások alatt meghatározható.

Törli, ellenőrzi az ürességét, programozza és ellenőrzi a konfigurációs memóriát.

· **Differenciális programozás**

Gyorsbillentyű: Ctrl+F5

Kiolvassa az alkatrész tartalmát és csak azokat a részeket programozza át ahol különbség található. Kód vagy adat védelem esetén nincs jelentősége a differenciális programozásnak, így teljes programozás fog történni. Ez a művelet csak akkor érhető el, ha a programozott eszköz is támogatja ezt a funkciót.

· **EEPROM memória differenciális programozása (Differential program EEPROM)**

A gyorsbillentyű a beállítások alatt meghatározható.

Kiolvassa az EEPROM memóriát és csak azokat a részeit programozza át ahol különbség található. Ugyanúgy működik mint a normál differenciális programozás. AVR eszközök esetén kell ezt a funkciót használni, ha csak az EEPROM memóriát szeretnéd programozni és az előtte nem lett törölve.

· *Sorozat gyártás (Mass production)*

A gyorsbillentyű a beállítások alatt meghatározható.

Ezzel lehet a legegyszerűbben megoldani az olyan feladatokat ahol sok ugyanolyan eszközt kell feltölteni megegyező vagy hasonló tartalommal. (pl különböző sorozat számmal).

Eszköz -> Kiolvasás (Device -> Read)

Gyorsbillentyű: Shift+F6

· *Összes kiolvasása (Read All)*

Gyorsbillentyű: F6

Kiolvassa az egész alkatrészt.

· *Összes kiolvasása kivéve EEPROM*

A gyorsbillentyű a beállítások alatt meghatározható.

Az EEPROM memória kivételével kiolvassa a teljes alkatrészt.

· *Kód memória kiolvasása*

A gyorsbillentyű a beállítások alatt meghatározható.

Kiolvassa a kód memóriát.

· *EEPROM memória kiolvasása (Read EEPROM)*

A gyorsbillentyű a beállítások alatt meghatározható.

Kiolvassa az EEPROM memóriát.

· *Konfigurációs memória kiolvasása (Read configuration memory)*

A gyorsbillentyű a beállítások alatt meghatározható

Kiolvassa a konfigurációs memóriát.

Bizonyos alkatrészek esetén némelyik parancs nem érhető el.

Eszköz -> Ellenőrzés (Device -> Verify)

Gyorsbillentyű: Shift+F7

Az almenük hasonlóan működnek mint kiolvasáskor. Lásd Kiolvasás.

Eszköz -> Törlés (Device • Erase)

Gyorsbillentyű: Shift+F8

Az almenük hasonlóan működnek mint kiolvasáskor. Lásd Kiolvasás.

Eszköz -> Üresség ellenőrzés (Device • Blank check)

Gyorsbillentyű: Shift+F9

Az almenük hasonlóan működnek mint kiolvasáskor. Lásd Kiolvasás.

Eszköz -> Eszköz kiválasztás (Device -> Select part...)

Gyorsbillentyű: F4

Egy kiválasztó ablak nyílik meg, ahol számos memória típus közül választható ki a megfelelő. A kívánt típus kiválasztása után a programozó által támogatott alkatrészek listája jeleneik meg. Ha a kiválasztott alkatrész nem programozható ICSP módban, az ICSP mód automatikusan kikapcsolásra kerül.

Ha olyan alkatrészt szeretnél kiválasztani amit a programozód nem támogat válassz másik programozót.

4.5. Beállítások menü (Options menu)

Beállítások -> Program beállítások (Options · Program settings...)

Gyorsbillentyű: Shift+F10

Beállítások -> Program beállítások-> Programozás fül (Options · Program settings...; Programming tab)

Gyorsbillentyű: Shift+F10

Minden általános beállítás ezen az ablakon végezhető el.

A programozó típustól függő beállítások (pl. kommunikációs port, ICSP mód) megadhatók a konfiguráció során.

· **Minden programozás előtt .hex fájl újra betöltése (Reload .HEX file before every programming)**

Ha ki van jelölve akkor minden programozás előtt kiolvassa a fájl tartalmát. Sorozat gyártás esetén először a fájl kerül beolvasásra, majd az azt követő sorozat szám kerül kiadásra.

· **Kérdezz OTP illetve kód / adat védelem programozás előtt (Ask before programming of OTP / programming of Flash/ Code/Data protection programming/ differential program)**

Ebben az esetben a megadott programozási műveletek evégzése előtt a felhasználó jóváhagyása szükséges. Ha gyorsan kell a programozás elegendő egyszer elvégezni a jóváhagyást, kivéve kód/adat védelem programozás ahol további jóváhagyás szükséges.

· **Fuse bit figyelmeztetések mutatása (Show fuses warning messages)**

Ha bekapcsolásra kerül akkor a felhasználó fuse bit beállítással kapcsolatos figyelmeztető üzeneteket kap. Ajánlatos bekapcsolva hagyni ezt a funkciót.

· **Állapot jelző ablak bezárása (Automatically close status window)**

Ha nem történt a művelet elvégzése során hiba, vagy nem jelent meg egyéb figyelmeztetés, az állapot jelző automatikusan bezáródik.

· **Csippanás, ha sikeresen/nem sikeresen fejeződött be (Beep after succesful/unsuccesful finishing)**

A beállítástól függően csippanás hallatszik ha sikeresen vagy épp nem sikeresen történt meg a művelet elvégzése.

· **A szoftver minden hangjának kikapcsolása (Turn off all sounds of the UP)**

Ha kijelölésre kerül soha nem hallatszik hangjelzés.

· **ICSP programozás (ICSP programming)**

Az ICSP programozás nem használható minden alkatrész esetén, illetve néhány programozó külön programozó adaptereket, alkatrészeket igényelhet.

ICSP kábel használata esetén a „ICSP táp késleltetés (ISCP power up delay)” késleltetéssel jelenik meg a táp a microcontrolleren , pl a szűrő kondenzátorok feltöltődése a cél áramkörön. PICQUICK és PRESTO rendelkezik beépített túláram védelemmel, ami méri az áramerősséget azután, hogy a táp rákötésre került a microcontrollerre. ISCP mód esetén a késleltetés mértéke ettől a beállítástól függ. Ha ez az érték szükségtelenül túl nagyra van növelve nagyobb lehet az alkatrész

sérülésének veszélye nem megfelelő kezelés esetén. Kisebb késleltetés esetén a programozó hamarabb képes érzékelni az esetleges rövidzárt, így biztosítva annak épségét. A túláram határa kb 100 mA mindkét tápon és a programozási feszültségnél is.

• **Lassabb feszültség váltás ICSP esetén (Slower switching of voltage with ICSP)**

Kiválasztása esetén lehetőség nyílik a VDD töltési és kisülési idő megváltoztatására. Ha egy kondenzátor van a VDD lábán, ez a láb lassabban vált feszültség szintet. Ez problémát okozhat a programozás során. Ebben az esetben segíthet a töltési és kisülési idő növelése.

• **Ne végezz üresség ellenőrzést konfigurációs word programozás előtt (Don't any blank check before cfg word programming)**

A legtöbb újraprogramozható alkatrész esetén lehetséges a konfigurációs memória felülírása az egész alkatrész törlése nélkül. Ennek végrehajthatóságához a konfigurációs memória üresség ellenőrzését le kell tiltani. Ez nem végez programozást a teljes alkatrészen teljes törlés végrehajtásakor. Kihagyja az üresség ellenőrzést a törlés után, így a programozási művelet is felgyorsul. A nem megfelelően törölt alkatrész nem programozható fel jól, ami meg is mutatkozik kicsivel később az ellenőrzés során. Nem megfelelő törlés általában több száz esetből egyszer szokott előfordulni.

• **Nem kérek üresség ellenőrzést törlés után (Do not perform blank check after erasing)**

Ha beállításra kerül a programozó nem ellenőriz ürességet a törlés után, ha az megfelelően elvégzésre került. A programozás felgyorsul, de nem megfelelően törölt alkatrész esetén problémát okozhat.

• **Ne töröld az eszközt programozás előtt (Do not erase device before programming)**

Az eszköz a programozás előtt nem kerül törlésre.

• **Nincs adat memória törlés annak programozása előtt (No data memory erase before its programming)**

AVR eszközök programozása esetén hasznos ez a funkció, ahol .hex fájlt kell külön betölteni az EEPROM memóriának. Ha felhasználó nem akarja megváltoztatni az EEPROM memóriát, ez beállítás szükséges.

• **Ne ellenőrizd a nem programozott word-okat a memória végén (Don't verify unprogrammed words at the end of the memory)**

Ha a programozott memória tartományának végén van valamennyi üres terület, az nem kerül ellenőrzésre. Az ellenőrzés ezáltal gyorsabbá válik.

• **Ne ellenőrizd (Do not verify)**

Az ellenőrzés kihagyásra kerül a programozási művelet elvégzése után. Sorozat gyártás esetén nem használható!

Beállítások -> Program beállítások-> Panelek fül (Options • Program settings...; Panels tab)

Gyorsbillentyű: Shift+F10

Itt állítható be az alkalmazás kinézete és, hogy milyen alkotórészei kerüljenek megjelenítésre.

Bővebben lásd angol nyelvű kézikönyv.

Beállítások -> Program beállítások-> Fájlok fül (Options · Program settings...; Files tab)

Gyorsbillentyű: Shift+F10

· Fájl mentés típus (File save style)

A memória részenként történő mentésével kapcsolatos beállításokat lehet itt elvégezni.

· Az aktuális .hex fájlhoz automatikus újabb verzió keresés (Do automatic check for newer versions of actual .hex file)

Szoftver fejlesztés esetén hasznos, mert a fájl újra beolvasásra kerül ha az módosult.

· Rész szám ellenőrzése .hex fájl betöltésekor (Check part number when loading .hex file)

Rész típus azonosítást végez a HEX fájl esetében.

· Rész szám mentése .hex fájlba (Save part number into .hex file)

Ha beállításra kerül, a fájl végéhez egy rész azonosító kerül kiadásra. A legtöbb program Intel HEX fájlokkal dolgozik, mely figyelmen kívül hagyja ezt a sort, azonban az ilyen fájl nem tekinthető Intel HEX fájljal teljesen megegyezőnek.

· BIN fájl betöltési és mentési stípusa (BIN file loading and saving style)

Ennek az opciónak a segítségével megadható, hogy a program rákérdezzen a BIN fájl mentés és betöltés típusára.

· Használt helyek mentése hex fájlba (Save unused positions to .hex file)

Ha kikapcsolásra kerül a fájl kisebb lehet, de ez problémákat is okozhat.

· Adat memória ürítés eszköztől kiolvasáskor (Clear main / data / ID locations before reading.)

Az adott memória minden bitje beállításra kerül fájl beolvasáskor.

· Konfig memória törlése fájl beolvasás előtt (Erase configuration memory before reading)

Ezt az opciót akkor érdemes kikapcsolni, ha a fuse bitek nem a hex fájlban vannak tárolva.

· Azonosító helyek olvasása az eszköztől (Read data memory / ID locations from the device)

Bekapcsolásakor a programozandó alkatrész EEPROM / ID kiolvasása az alkatrészből történik meg, nem a fájlból, ami így megakadályozza ennek az információnak a programozás közbeni felülírását.

Figyelem! Ez a funkció akaratlan hozzáférést tesz lehetővé a programozó felé pl program indításkor.

· Projekt tárolási stílus (Project storing style)

Projekt mentéssel kapcsolatos opciók.

Beállítások -> Program beállítások-> Színek fül (Options · Program settings...; Colors tab)

Gyorsbillentyű: Shift+F10

A hexeditor színei állíthatók be a felhasználói igényeknek megfelelően.

Beállítások -> Program beállítások -> Szerkesztők fül (Options · Program settings...; Editors tab)

Gyorsbillentyű: Shift+F10

· Keskeny program memória szerkesztő (Narrow code memory editor)

A memória szerkesztő soronként nem 8 hanem 16 cellát jelenít meg. Alacsony felbontású képernyő esetén lehet hasznos. Ez az opció akaratlanul is megváltozhat egy másik alaktrész kiválasztása esetén.

- Azonosító helyzet maszkolás (Mask ID positions)

Az azonosító hely kitakarható. Kijelölése esetén a program ezt automatikusan intézi.

- Konfigurációs memória szerkesztő: cfg word mutatása fuse helyett (Configuration memory editor: show cfg word instead of fuses)

Csak gyakorlott felhasználóknak ajánlott. Ez a lehetőség közvetlen konfigurációs szó szerkesztést tesz lehetővé.

Beállítások -> Program beállítások -> Sorozat számok fül (Options · Program settings...; Serial numbers tab)

Gyorsbillentyű: Shift+F10

A sorozat számokkal kapcsolatos információk megtalálhatók a Sorozat számok fejezetben.

Beállítások -> Program beállítások -> Egyebek (Options · Program settings...; Others tab)

Gyorsbillentyű: Nincs

- Frissítés keresési beállítások
- Belső és külső táp ütközés engedélyezés

Beállítások -> Nyelv kiválasztás (Options · Language selection...)

Gyorsbillentyű: Ctrl+L

Egyszerűen kiválasztható a használandó nyelv.

Az epromirok.hu oldalról külön letölthető magyar fájl segítségével a magyar nyelv is kiválasztható. Ehhez a nyelvi fájlt előbb be kell másolni az UP gyökér mappájába. Alapértelmezés szerint: c:\Program Files\ASIX\UP

Beállítások -> Gyors billentyűk (Options · Keyboard shortcuts)

Gyorsbillentyű: Ctrl+K

Műveletekhez rendelhető gyorsbillentyűk beállítása.

4.6. Súgó menü (Help menu)

Súgó -> Program súgó (Help · Help on program)

Gyorsbillentyű: F1

Az F1 billentyű segítségével segítség kérhető.

Súgó -> Támogatott eszközök listája (Help · List of supported devices)

Megtekinthető az összes támogatott eszköz listája.

Súgó -> ASIX-Tools weboldal (Help · ASIX - Tools website)

Megnyitja az ASIX weboldalát.

Súgó -> Frissítés keresése a neten (Help · Check Internet for an update)

A program az interneten keres frissítést.

4.7. PRESTO beállítások ablak (PRESTO programmer settings window)

Tétlen állapot táp (Idle power supply)

- Nincs / Külső (None / External): A programozó nem biztosít tápot a VDD lábra. Csak akkor lehetséges a program indítása ha a külső tápegység elérhető.
- Belső 5V (Internal 5V): A programozó 5V tápot biztosít a VDD lábra. A cél eszköz tápellátása megoldható erről a lábról.

Aktív állapot táp (Active power supply)

- Külső 3-tól 5V-ig: A programozó nem biztosít tápot a cél eszközre, sőt a ki és bemenet áramkör is a cél eszköztől kapja a szükséges tápot.
- Belső 5V: A programozó 5V tápot biztosít a VDD lábra.

PIC-hez kapcsolódó további lehetőségek

-MCLR láb vezérlés (MCLR pin control)

Tétlen állapotban az MCLR láb állapota szabályozható ezekkel a gombokkal.

-Programozási mód (Programming method)

- HVP: Klasszikus programozás 13V biztosításával az -MCLR/VPP-re
- LVP: LVP mód, csak a 0 és 1 logikai értékek kerülnek az -MCLR/VPP-re.

-Programozási algoritmus (Programming algorithm)

- Auto: Az algoritmus a VDD lábon jelen lévő feszültség alapján kerül meghatározásra.
- Vcc=5V: Algoritmus a gyors 5V-os programozáshoz.
- Vcc=2-től 5V-ig: Lassú programozási algoritmus. Működik minden használt feszültséggel.

-PE

PIC24 és dsPIC33 esetén lehetséges programozási módot választani a PE használat rubrika használatával. A PE jelentése "Programming Executive (Parancs használat)". Ez a mód gyorsabb mint az általános programozási mód.

AVR és 8051 microcontrollerekhez kapcsolódó további lehetőségek

Oscillátor frekvencia (Oscillator frequency)

Az AVR microcontrollerek külső vagy belső oszcillátort igényelnek a programozás alatt. A maximális kommunikációs sebesség a használt oszcillátor frekvenciájától függ.

Gyorsabb programozás lassabb órajellel (Faster programming with slow clock)

A chip törlése után a fuse bitek programozásra kerülnek, így a belső oszcillátor maximális frekvenciája kerül használatra. Így a programozó gyorsabban tud kommunikálni az alkatrésszel. A programozó rövidebb programozási időt érhet el alacsony órajel esetén. Ez a funkció csak akkor használható, ha az Összes programozása van kiválasztva, mert az összes fuse bit értéknek vissza kell programozódnia a művelet végén.

Fordított RESET (Inverse RESET)

Kiválasztása esetén a programozó fordított reset jelet állít elő. Ez akkor lehet hasznos ha valamelyik reset áramkör fordított bemeneti szintet igényel a cél eszköz használt kimenetéhez képest amikor a programozó csatlakoztatásra kerül.

HVP

Kiválasztása esetén a programozó nagy feszültséget használ a chippel történő kommunikáció során. Lehetővé teszi a chip programozását akkor is ha a külső RESET jel kikapcsolásra kerül.

I2C memóriákhoz kapcsolódó további lehetőségek**I2C busz sebesség (I2C bus speed)**

Válaszd a maximális I2C busz sebességet. A PRESTO belső 2.2k Ω lehúzó ellenállást használ I2C busz használat esetén.

I2C memória cím (I2C Memory Address)

Válassz egy programozott I2C memória címet.

4.8. Hex editor ablakok

A memória tartalmak megjelenítésére hexeditorok használatosak. A hexeditorban használt színek segítenek megjelölni az adatok eredetét, állapotát, így könnyedén meg lehet állapítani melyik cella került egy fájlból betöltésre, melyik lett beprogramozva, stb. A színek a beállítások alatt teljesen testre szabhatók.

Egy terület kijelölése

A hex editorban egy terület kijelölhető az egérmutató használatával miközben nyomva tartjuk a SHIFT billentyűt. Kijelölés után lehetőség nyílik annak egy megadott értékkel történő kitöltésére vagy az egérrel történő jobb gombos kattintás után az előugró menü segítségével műveletet végezni.

Külön-külön szerkesztő érhető el a Kód memóriának, EEPROM memóriának és Konfig memóriának a Menü-> Nézet menüpont alatt.

Gyorsbillentyűk: F10, F11, F12

Az adott ablak bezárása: Esc

Bővebben lásd angol kézikönyv!

5. JTAG SVF PLAYER

JTAG csatolóval rendelkező eszközök PRESTO-val történő programozására használható. Lehetséges vele alkatrészeket programozni és tesztelni ha az adatok SVF vagy XSVF formátumban vannak.

PI:

CPLD - Xilinx XC9572XL, XCR3256XL, Altera EPM7128AELC, EPM7064SLC, EPM7128SLC stb

FPGA konfigurációs Flash PROM - Xilinx XC3S1000, XC2V1000 stb

memória - XC18V02VQ44I, ...

Atmel ATmega128, ATmega64, ...

Megjegyzés:

- Az XSVF formátum ajánlott CPLD Xilinx XC9500 esetén, SVF formátum az összes többi esetén javasolt.
- JTAG player nem támogatja a .JED fájlokat.
- Lásd Altera programozási példákat.

5.1. Telepítés

A JTAG PLAYER telepítése nagyon egyszerű. A telepítő (JTAG_xxx_EN.EXE) megtalálható a programozó CD lemezén vagy letölthető az ASIX weboldaláról.

<http://tools.asix.net/>. Indítsd el a fájlt és kövesd az utasításokat.

5.2. Egyszerű programozás / Tesztelés

SVF fájlok (Serial Vector Format) a magas szintű IEEE 1149.1 busz műveletek esetén használatos.

SVF (Serial Vector Format) ajánlott minden programozási és tesztelési feladatra, kivéve Xilinx CPLD XC9500.

Xilinx CPLD XC9500 programozására csak az XSVF (Xilinx Serial Vector Format) használható.

- Csatlakoztasd a PRESTO programozót a cél eszköz JTAG csatolójára.
A csatlakoztatási módot lásd feljebb a „PRESTO csatlakoztatása JTAG csatolóval rendelkező eszközre” fejezet alatt.
- Indítsd el a jtagplay.exe alkalmazást
- Válaszd az Open & Process File menüpontot

5.3 SVF/XSVFfájl létrehozási példák

AVR ATmega128 programozása

SVF fájl létrehozható az ATMEL weboldaláról, a Tools & Software of AVR 8-bit RISC rész alól letölthető avrsvf.exe alkalmazással.

Példa :

```
avrsvf -datmega128 -s -e -ifmyfile.hex -pf -vf -ovmyfile.svf -mp
```

létrehoz egy SVF fájlt melynek eredményeként törölni, programozni és ellenőrizni fogja a myfile.hex fájlt.

avrsvf -h utasítással további információk kérhetők.

Xilinx CPLD programozása

Az iMPACT szoftver letölthető a Xilinx weboldaláról, mely létre tud hozni XSVF fájlt. Az Operation Mode Selection (Működési mód választás) alatt mely megjelenik az iMPACT indításakor, válaszd a Prepare Configuration Files -> Boundary-Scan File -> XSVF File menüt.

Idíts minden műveletet (törlés, programozás, ellenőrzés, teszt...) ahogy a programozó (pl.párhuzamos kábel (Parallel Cable)) csatlakoztatásra került és zárd be az iMPACT alkalmazást.

Az XSVF fájlban rögzítésre kerül minden művelet.

SVF fájl használata nem ajánlatos Xilinx CPLD XC9500 programozásához, mert az XC9500(XL) eszközök nem írhatók le SVF fájl segítségével.

Lattice CPLD programozása

Készíts egy JED kimeneti fájlt pl az ispLEVER szoftverrel. Ezután készítsd el az SVF fájlt Universal File Writer (UFW) használatával. Ez a program telepítésre kerül az ispVM-el. Az ispLEVER és ispVM letölthető a Lattice weboldaláról.

Altera CPLD programozása

Az Altera QUARTUS II szoftver tudja elkészíteni az SVF fájlt. Az így létrehozott fájl még nem alkalmas a chip PRESTO-val történő programozására, mert az csak ATE eszközöket támogat (hibás silicon ID). De a fájl manuálisan módosítható, hogy használható lehessen PRESTO programozóval is. Egyszerűen törölni kell a „CHECKING SILICON ID” részt az SVF fájlból.

Bővebben lásd angol kézikönyv.

6. PRECOG

Ez a szoftver a Cyan Technology eCOG1 microcontroller programozásra használható. Rendelkezik alap hibakeresési funkciókkal az eICE csatolón keresztül (Indít, Leállít, Reset)

6.1. Telepítés

A PRECOG telepítése nagyon egyszerű. A telepítő (PRECOG_xxx_EN.EXE) megtalálható a programozó CD lemezén vagy letölthető az ASIX weboldaláról.

<http://tools.asix.net/>. Indítsd el a fájlt és kövesd az utasításokat.

6.2. Programozás

Csatlakoztasd a PRESTO-t az eCOG1-hoz.

Nyisd meg az adat fájlt az Open (Megnyit) gombra kattintva vagy a File/Open menüpont segítségével. A .rom kiterjesztésű fájlokat támogatja.

Nyomd meg a Program gombot vagy a Device/Program menüpontot használd a programozás megkezdéséhez.

6.3 Debugging (Hibakeresés)

Csatlakoztasd a PRESTO-t az eCOG processzorhoz.

Nyomd meg az Attach gombot, vagy használd a Device/Attach menüpontot.

Így használhatók a debugging gombok (Indít, Leállít, Reset) vagy az azonos nevű menüpontok a Debug menü alatt.

7. A presto.dll library

A presto.dll-be ágyazott funkciók engedik szerkeszteni vagy kiolvasni a PRESTO programozó egyes lábainak a logikai értékét. Ezen a módon számos kommunikációs eljárás építhető be.

QSetPins() funkció engedélyezi a kimeneti láb vezérlést.

QGetPins() funkció engedélyezi a bemeneti láb kiolvasást.

QSendByte() funkció engedélyezi a gyors SPI byte küldést az adaton és a órajel lábakon. Ha kiolvasás is szükséges akkor a QSendByte_OutIn() használható.

Számos programozó funkció is létezik a tápellátáshoz, a programozási feszültség szabályozáshoz és a visszatérő értékek kiolvasásához.

A library minden PRESTO programozóval használható, nem függ annak verzió számától.

„A” FÜGGELÉK: Konfigurációs word címek PIC eszközökben

PIC10xxx konfigurációs word címek

Minden PIC10xxx eszköznek az FFFh címen van a konfig word-je

PIC12xxx konfigurációs word címek

FFFh		2007h
PIC16F505	PIC12C509A	PIC12C671
PIC12C508	PIC12F510	PIC12C672
PIC12F508	PIC12CE518	PIC12CE673
PIC12C508A	PIC12CE519	PIC12CE674
PIC12C509	rfPIC12C509Ax	PIC12F629
PIC12F509		PIC12F675
		rfPIC12F675x
		PIC12F635

PIC16xxx konfigurációs word címek

PIC16xxx with cofig mem addr. FFFh			
PIC16C54	PIC16C57	PIC16C54-LP	PIC16C56-LP
PIC16C54A	PIC16C57C	PIC16C55-HS	PIC16C57-HS
PIC16C54B	PIC16C58A	PIC16C55-RC	PIC16C57-RC
PIC16C54C	PIC16C58B	PIC16C55-XT	PIC16C57-XT
PIC16C55	PIC16HV540	PIC16C55-LP	PIC16C57-LP
PIC 16C55A	PIC16C54-HS	PIC16C56-HS	PIC16F54
PIC16C56	PIC16C54-RC	PIC16C56-RC	PIC16F57
PIC16C56A	PIC16C54-XT	PIC16C56-XT	PIC16F59
PIC16C505			

Megjegyzés: Minden más támogatott PIC16xxx eszköz konfigurációs word-je a 2007h címen található.

PIC18xxx konfigurációs word címek

Microcontroller	Cfg. Mem. Addr.	Microcontroller	Cfg. Mem. Addr.	Microcontroller	Cfg. Mem. Addr.
PIC18F24J10	3FF8h	PIC18F63J11	1FF8h	PIC18F83J11	1FF8h
PIC18LF24J10	3FF8h	PIC18F63J90	1FF8h	PIC18F83J90	1FF8h
PIC18F25J10	7FF8h	PIC18F64J11	3FF8h	PIC18F84J11	3FF8h
PIC18LF25J10	7FF8h	PIC18F64J90	3FF8h	PIC18F84J90	3FF8h
PIC18F44J10	3FF8h	PIC18F65J10	7FF8h	PIC18F85J10	7FF8h
PIC18LF44J10	3FF8h	PIC18F65J90	7FF8h	PIC18F85J11	7FF8h
PIC18F45J10	7FF8h	PIC18F65J15	BFF8h	PIC18F85J90	7FF8h
PIC18LF45J10	7FF8h	PIC18F66J10	FFF8h	PIC18F85J15	BFF8h
		PIC18F66J15	17FF8h	PIC18F86J10	FFF8h
		PIC18F67J10	1FFF8h	PIC18F86J15	17FF8h
				PIC18F87J10	1FFF8h

Megjegyzés: Minden más támogatott PIC18xxx eszköz konfigurációs word-je a 300000h címen található.

dsPIC30xxx konfigurációs word címek

Minden dsPIC30xxx eszköznek a 7C0000h címen van a konfigurációs word-je

dsPIC33xxx konfigurációs word címek

Minden dsPIC33xxx eszköznek a 7C0000h címen van a konfigurációs word-je

PIC24xxx konfigurációs word címek

Microcontroller	Cfg. Mem. Addr.
PIC24FJ16GA0xx	2BFCh
PIC24FJ32GA0xx	57FCh
PIC24FJ48GA0xx	83FCh
PIC24FJ64GA0xx	ABFCh
PIC24FJ96GA0xx	FFFCh
PIC24FJ128GA0xx	157FCh

Minden PIC24H eszköznek a 7C0000h címen van a konfigurációs word-je

„B” FÜGGELÉK: UP DLL.DLL nevek és értékek beállítása

Néhány batch fájl példa segítségével mutatjuk be, hogyan működnek ezek a beállítások. Ezek az információk csak gyakorlott felhasználóknak ajánlottak és saját felelősségre használhatók.

Bővebb információk az angol kézikönyvben.

„C” FÜGGELÉK: ICSP használata

ICSP (In-Circuit Serial Programming / Áramkörön történő soros programozás) egy PIC microcontroller programozási mód, mely lehetővé teszi az áramkörre forrasztott alkatrész programozását. Két különböző programozási algoritmus lehetséges:

HVP (+13V a Vpp-n) vagy LVP (LVP láb használat). Az LVP programozás letiltható a konfigurációs word-ben. Az új microcontrollerek esetében alapértelmezés szerint az LVP mód engedélyezve van, így az első programozás során gondoskodni kell az LVP lábról. Az LVP láb log 0 állapotban kell legyen HVP programozás alatt.

A programozás alatt használt lábak:

HVP algoritmus (+13V a Vpp-re)

- -MCLR/VPP láb le kell választva legyen a reset áramkörrel (pl egy 10kΩ ellenállással). A programozási feszültség (VPP) +13V erre a lábra kerül programozás alatt. A jel és feszültség szint emelkedésre nem szabad, hogy hatással legyen a cél áramkör.
- LVP láb (ha az alkatrésznek van ilyen) log.0 állapotban kell legyen!!
- RB6 és RB7 lábra nem szabad, hogy hatással legyen a cél áramkör a programozás alatt.

LVP algoritmus (+13V nélkül)

- RB6, RB7, LVP és -MCLR/VPP lábra nem szabad, hogy hatással legyen a cél áramkör a programozás alatt. A többi láb akármelyik logikai állapotban lehet.
- Az LVP algoritmus csak a PRESTO programozóval támogatott.

A programozó lábain megjelenő maximális áramfelvételek

	PRESTO	PICQUICK	PICCOLO	CAPR-PI
CLK & DATA, VPP @ 0V/5V	24mA	4mA	4mA	8mA
Vpp @ 13V	50mA ¹⁾	50mA ¹⁾	cca 1mA	cca 1mA

1) Csak flash memóriák esetén. OTP (egyszer programozható) alkatrészek esetén maximum csak 1mA lehet. A programozó 50mA-t biztosít a VPP lábra.

Az adat lábakon a frekvencia több MHz is lehet a programozás alatt és a cél eszköz áramköre nem szabad, hogy különösebben hatással legyen a jel esésekre.

Tápellátási opciók

Minden esetben a földnek (GND) csatlakoztatva kell lenni.

A microcontroller kétféle tápellátással programozható:

- külső tápellátás a cél eszközön

- belső tápellátás a programozóról (5V)

Külső tápellátás olyan microcontroller típusok esetén nem használható, ahol az - MCLR/VPP láb ki és bementként (I/O) konfigurálható.

A belső tápellátás csak akkor használható ha a cél áramkör nem vesz fel túl nagy áramot a programozó Vcc lábáról.

Programmer	Maximal current drawn from the programmer
PRESTO	90mA ¹⁾
PICQUICK	10mA ¹⁾
PICCOLO	50mA

A programozó rendelkezik túláram védelemmel. Ha a megengedett áram jelentősen túllépésre kerül egy megadott időre (beállítható a Program beállítások->Programozás fül alatt) akkor a programozó lekapcsolja a tápot. A PRESTO minden esetben figyeli a túláramot ha a tápellátás be van kapcsolva.

A PRESTO programozó hardveresen támogatja a külső tápellátást. Képes felhasználni a Vdd lábon jelen lévő feszültséget az I/O áramkörök ellátására. Ez a feszültség lehet kisebb is mint 5V. Mindenképp győződj meg róla, hogy a programozandó eszköz nem-e csak 5V-nál kisebb feszültségen működik, mert ebben az esetben csak alacsony feszültségen programozható.

Ha a cél áramkör kondenzátorokat is tartalmaz, melyek lelassítják a táp feszültség váltást szükséges lehet nagyobb töltési és kisülési időket beállítani. (Beállítások-> Program beállítások -> Programozás fül -> Lassabb feszültség váltás ICSP esetén)

Programmer	Charging current	Discharging current
PRESTO	corresponds to 50Ω	corresponds to 1kΩ
PICQUICK	corresponds to 50Ω	corresponds to 10kΩ
PICCOLO	corresponds to 50Ω	none

A körülbelüli idő beállítható a szoftverben $t[\mu s] = 2.5 \times C[\mu F] \times R[\Omega]$.

PI abban az esetben ha a cél áramkör of 33 μF kondenzátorral rendelkezik, akkor

$2.5 \times 33 \times 50 = 4125 \mu s$ a szükséges töltési (bekapcsolási) idő és

$2.5 \times 33 \times 1000 = 82.5ms$ a kisülési idő.

Megjegyzés:

- Néha hiba fordul elő, mert az UP nem tudja programozni a kalibrációs word-t vagy eszköz azonosító kiolvasáskor, illetve egy hibaüzenet jelenik meg, hogy a VDD lábon túláram van. Ebben az esetben segíthet a töltési és kisülési idő hosszának megváltoztatása.
- Ha olyan hiba fordul elő, hogy túláram van a VPP lábon, próbálj használni rövidebb ICSP kábelt (leghosszabb 20cm).

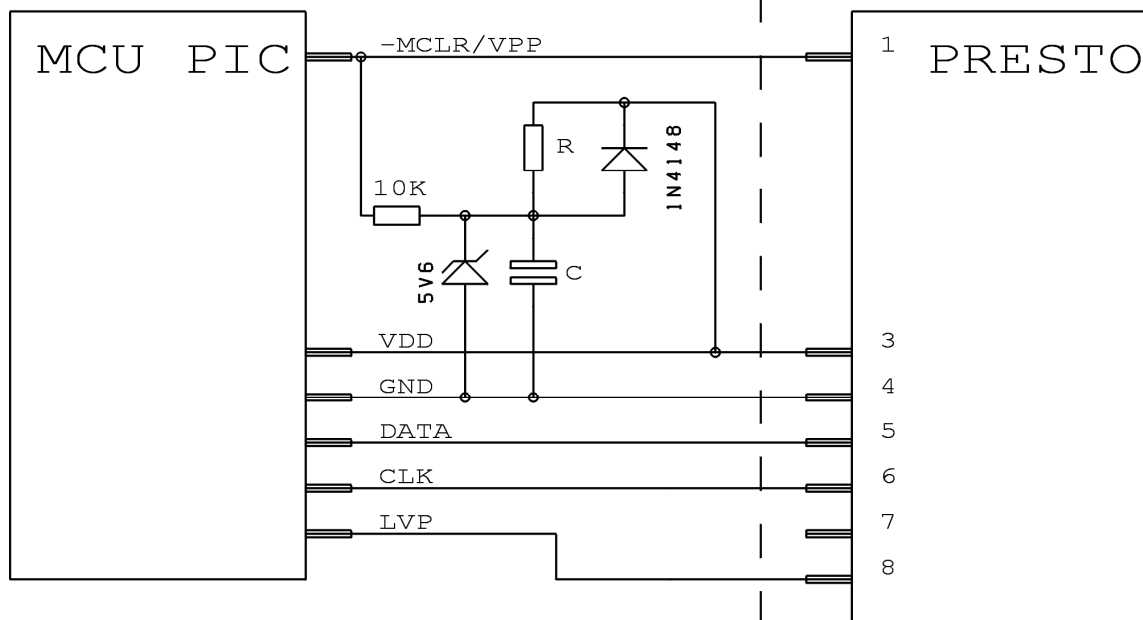
ICSP csatlakozó

Minden ASIX programozó egyesített csatlakozót használ ICSP programozáshoz, mely 2.54mm lábosztással rendelkezik. Ez a csatlakozó 6 vagy 8 lábbal rendelkezik (a programozó típusától függ) 5 (vagy 7) jellel. A csatlakozó bővített verziója (8 lábú) rendelkezik LVP lábbal, mely LVP programozáshoz használható.

Pin number	Signal	Programming connector
1	-MCLR	VPP/-MCLR
2		not used (key)
3	VCC	VCC
4	GND	GND
5	RB7	DATA
6	RB6	CLOCK
7		not used
8	RB3/RB4/RB5	LVP

Az -MCLR/VPP láb javasolt bekötése:

User application



Az ajánlott bekötés figyelembe veszi a Microchip adatlapon megadott ajánlásokat.

Az R és C értéke határozza meg azt az időt amit a processzor reset állapotban töltsön. A dióda teszi lehetővé a kondenzátor gyors kisülését ha a VDD láb lecsatlakoztatásra került. Zener dióda korlátozza a programozó által biztosított programozási feszültséget (+13V).

Az áramkör leegyszerűsíthető az R, C és a dióda elhagyásával. Ekkor az áramkör nem rendelkezik reset állapot tartási idővel.