

Návod na obsluhu pre

BeeHive204

Veľmi rýchly univerzálny 4x48 pinový concurrent multiprogramovací systém s možnosťou ISP

BeeHive4+

Rýchly univerzálny 4x48 pinový concurrent multiprogramovací systém s možnosťou ISP

BeeProg2

Veľmi rýchly univerzálny 48 pinový programátor s USB/LPT pripojením a s možnosťou ISP

BeeProg+

Univerzálny 48 pinový programátor s USB/LPT pripojením a s možnosťou ISP

SmartProg2

Univerzálny 40 pinový programátor s USB pripojením a s možnosťou ISP

MEMprog2

Univerzálny 40 pinový programátor pamätí

T51prog2

40 pinový programátor MCS51 a Atmel AVR mikroprocesorov s možnosťou ISP

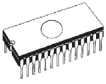
PIKprog2

40 pinový programátor PIC procesorov s možnosťou ISP

SEEpog

Programátor sériových EEPROM

ELNEC s.r.o.
Prešov, Slovensko
9. December 2011



Obsah tohto dokumentu je duševným vlastníctvom firmy ELNEC s.r.o., Prešov. Žiadna časť tohto dokumentu nesmie byť reprodukována, kopírovaná, prekladaná alebo prenášaná v ľubovoľnej forme akoukoľvek cestou bez predchádzajúceho písomného súhlasu firmy ELNEC s.r.o. Prešov.

Rovnako ovládací program k programátorom ELNEC je duševným vlastníctvom firmy ELNEC s.r.o. Prešov. Ovládací program (ani jeho časť) nesmie byť analyzovaný a modifikovaný ľubovoľnou formou na akomkoľvek médiu pre žiaden účel.

Informácie, uvádzané v tomto návode na obsluhu sa snažia byť presné a aktuálne. Firma ELNEC s.r.o. nepreberá zodpovednosť za chybné použitie tohto návodu.

Pretože každý výrobok podlieha vývoju, môže sa stať, že dôjde k určitým vylepšeniam, ktoré táto príručka nezachytáva. Prípadné odchýlky od popisu v tomto návode sú uverejnené na www.elnec.sk

Firma ELNEC s.r.o., Prešov si vyhradzuje právo upravovať a zdokonaľovať výrobok, popísaný v tomto návode kedykoľvek bez predchádzajúceho upozornenia.

V návode sú použité názvy programových produktov, firiem a pod., ktoré môžu byť ochrannými známkami alebo registrovanými ochrannými známkami príslušných vlastníkov.

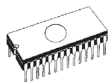
Ako používať tento návod na obsluhu

Tento návod na obsluhu je navrhnutý v snahe pomôcť Vám rýchlo a jednoducho inštalovať a používať Váš programátor. Návod predpokladá, že užívateľ má aspoň základné znalosti a skúsenosti s ovládaním PC aj s inštaláciou technického a programového vybavenia pre PC. Časť návodu, popisujúcu programové vybavenie programátora, odporúčame používať v spojení s integrovanou kontextovo citlivou nápovedou ovládacieho programu. V kontextovo citlivej nápovede sú zahrnuté aj doplnky a dodatky, ktoré vznikli po vydaní tohto návodu na použitie.

Vážený zákazník,

*dakujeme za dôveru, ktorú ste nám prejavili
zakúpením jedného z programátorov firmy ELNEC.*

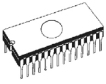
Aktuálna verzia návodu na obsluhu je na
našej internetovej stránke (www.elnec.sk).



Obsah

| | |
|---|-----------|
| Ako používať tento návod na obsluhu..... | 3 |
| Úvod..... | 7 |
| Konfigurácia výrobkov..... | 9 |
| Požiadavky na riadiaci počítač..... | 10 |
| Doplnkové služby:..... | 11 |
| Rýchly štart..... | 12 |
| Podrobný popis..... | 15 |
| BeeHive204 / BeeHive4+..... | 16 |
| Úvod..... | 17 |
| Popis programátora BeeHive204 / BeeHive4+..... | 19 |
| Manipulácia s programovanými obvodmi..... | 20 |
| In-system serial programming pomocou BeeHive204 / BeeHive4+..... | 21 |
| Test a kalibrácia programátora..... | 23 |
| Technická špecifikácia..... | 24 |
| BeeProg2 / BeeProg+..... | 30 |
| Úvod..... | 31 |
| Popis programátora BeeProg2 / BeeProg+..... | 33 |
| Pripojenie BeeProg2 / BeeProg+ k PC..... | 34 |
| Manipulácia s programovanými obvodmi..... | 35 |
| In-system serial programming pomocou BeeProg2 / BeeProg+..... | 35 |
| Multiprogramovanie pomocou BeeProg2 / BeeProg+..... | 37 |
| Test a kalibrácia programátora..... | 37 |
| Technická špecifikácia..... | 38 |
| SmartProg2..... | 44 |
| Úvod..... | 45 |
| Popis programátora SmartProg2..... | 46 |
| Pripojenie SmartProg2 k PC..... | 47 |
| Manipulácia s programovanými obvodmi..... | 47 |
| In-system serial programming pomocou programátora SmartProg2..... | 47 |
| Selftest..... | 49 |
| Technická špecifikácia..... | 49 |
| MEMprog2..... | 53 |
| Úvod..... | 54 |
| Popis programátora MEMprog2..... | 55 |
| Pripojenie MEMprog2 k PC..... | 56 |
| Manipulácia s programovanými obvodmi..... | 56 |
| Selftest..... | 56 |
| Technická špecifikácia..... | 57 |
| T51prog2..... | 60 |
| Úvod..... | 61 |
| Popis programátora T51prog2..... | 62 |
| Pripojenie T51prog2 k PC..... | 63 |
| Manipulácia s programovanými obvodmi..... | 63 |
| In-system serial programming pomocou programátora T51prog2..... | 63 |
| Selftest..... | 65 |
| Technická špecifikácia..... | 65 |
| PIKprog2..... | 68 |
| Úvod..... | 69 |
| Popis programátora PIKprog2..... | 70 |

| | |
|--|------------|
| Pripojenie PIKprog2 k PC | 71 |
| Manipulácia s programovanými obvodmi | 71 |
| In-system serial programming pomocou programátora PIKprog2 | 71 |
| Selftest | 73 |
| Technická špecifikácia | 73 |
| SEEprog | 76 |
| Úvod | 77 |
| Popis programátora SEEprog | 78 |
| Pripojenie SEEprog k PC | 78 |
| Manipulácia s programovanými obvodmi | 79 |
| Technická špecifikácia | 79 |
| Inštalácia | 81 |
| Inštalácia programového vybavenia | 82 |
| Inštalácia hardware | 88 |
| Pg4uw | 92 |
| Pg4uw – ovládací program | 93 |
| Súbor | 96 |
| Buffer | 102 |
| Obvod | 109 |
| Programátor | 135 |
| Nastavenia | 140 |
| Pomocník | 150 |
| Pg4uwMC | 153 |
| Spoločné poznámky | 162 |
| Starostlivosť o programátor | 163 |
| Software | 164 |
| Hardware | 170 |
| ISP (In-System Programming) | 170 |
| Iné | 173 |
| Ak sa vyskytnú problémy Záručné podmienky | 175 |
| Ak sa vyskytnú problémy s programátorom | 176 |
| Ak sa vyskytnú nepodporované obvody | 177 |
| Záručné podmienky | 177 |



Konvencie použité v návode

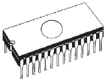
Odkazy na funkcie ovládacieho programu sú tlačené zvyrazneným písmom, napr. **Čítať**, **Súbor**, **Obvod** atď. Odkazy na ovládacie klávesy sú uzavreté do znakov < >, napr. <F1>.

Názvoslovie:

| | |
|---------------------|--|
| Obvod | rôzne typy programovateľných obvodov |
| ZIF päťica | zero insertion force socket = päťica (programátora) s nulovou zasúvacou silou |
| Buffer | oblasť v operačnej pamäti alebo na disku, ktorá slúži na uchovanie dát, potrebných na programovanie obvodov |
| Printer port | paralelný port PC, určený štandardne na pripojenie tlačiarne |
| USB port | typ PC konektora (sériový), ktorý je určený na pripojenie prenosných zariadení a periférií počítača |
| HEX formát | formát zápisu údajov tak, aby bol čitateľný štandardným textovým editorom, teda napr. byte 5AH je zapísaný ako znak 5 a znak A - to znamená 35H a 41H. Jeden riadok záznamu takého súboru obsahuje aj adresu, na ktorej sa údaje nachádzajú a celý záznam je zabezpečený kontrolnou sumou, takže je možné zistiť i prípadné chyby v údajoch. |



Úvod



Tento návod na obsluhu zahrňuje tieto ELNEC programátory: **BeeHive204**, **BeeHive4+**, **BeeProg2**, **BeeProg+**, **SmartProg2**, **MEMprog2**, **T51prog2**, **PIKprog2** a **SEEprog**.

BeeHive204 je veľmi rýchly 4x48 pinový **concurrent multiprogramovací systém** navrhnutý pre programovanie veľkého množstva obvodov s minimálnou záťažou pre obsluhu. Obvody sa programujú rýchlosťou blízkou maximálnej teoretickej rýchlosti. Použitím zabudovaného ISP konektora je možné programovať aj ISP programovateľné obvody v cieľovom zariadení.

BeeHive4+ je rýchly 4x48 pinový **concurrent multiprogramovací systém** navrhnutý pre programovanie veľkého množstva obvodov s minimálnou záťažou pre obsluhu. Obvody sa programujú rýchlosťou blízkou maximálnej teoretickej rýchlosti. Použitím zabudovaného ISP konektora je možné programovať aj ISP programovateľné obvody v cieľovom zariadení.

BeeProg2 je veľmi rýchly, univerzálny programátor a tester logických IO so 48 univerzálnymi pindriverami s USB a LPT pripojením k PC. Má zabudovaný ISP konektor na programovanie obvodov priamo v zariadení. Je navrhnutý tak, aby bolo jednoduché pridávať nové obvody do zoznamu podporovaných obvodov. BeeProg2 je naozaj univerzálny a lacný programátor, ktorý vo svojej triede iste poskytuje najlepší pomer cena/výkon.

BeeProg+ je rýchly, univerzálny programátor a tester logických IO so 48 univerzálnymi pindriverami s USB a LPT pripojením k PC. Má zabudovaný ISP konektor na programovanie obvodov priamo v zariadení. Je navrhnutý tak, aby bolo jednoduché pridávať nové obvody do zoznamu podporovaných obvodov. BeeProg+ je naozaj univerzálny a lacný programátor, ktorý vo svojej triede iste poskytuje najlepší pomer cena/výkon.

SmartProg2 je malý, rýchly a výkonný programátor všetkých typov programovateľných obvodov s USB pripojením. Má zabudovaný ISP konektor na programovanie obvodov priamo v zariadení. Je navrhnutý tak, aby bolo jednoduché pridávať nové obvody do zoznamu podporovaných obvodov. Vo svojej triede poskytuje iste jeden z najlepších pomerov cena/výkon.

MEMprog2 je malý, rýchly a výkonný programátor EPROM, EEPROM, Flash EPROM, NVRAM a sériových EEPROM s USB pripojením k PC. Zároveň je aj testerom statických RAM. MEMprog2 môže byť upgradovaný na SmartProg2.

T51prog2 je malý výkonný a rýchly programátor jedno čipových mikroprocesorov rady MCS51 a Atmel AVR s možnosťou ISP programovania a s USB pripojením k PC. Ako bonus programuje aj sériové EEPROM s rozhraním IIC, Microwire a SPI. T51prog2 môže byť upgradovaný na SmartProg2.

PIKprog+ je malý výkonný a rýchly programátor jednočipových mikroprocesorov PICmicro[®] firmy Microchip s možnosťou ISP programovania a s USB pripojením k PC. Má zabudovaný ISP konektor na programovanie Microchip PICmicro[®] obvodov priamo v zariadení. Ako bonus programuje aj sériové EEPROM s rozhraním IIC, Microwire a SPI. PIKprog2 môže byť upgradovaný na SmartProg2.

SEEprog je univerzálny programátor sériových EEPROM v 8 pinovom DIL puzdre s rozhraním IIC, SPI, Microwire a tiež špeciálť ako napr. digitálne teplomery. Programátor tiež podporuje LV obvody (3,3 V).

Všetky programátory sú periférne zariadenia osobného počítača triedy IBM PC Pentium kompatibilný a vyšších typov. Programátory sa k PC pripájajú cez USB port alebo štandardný paralelný port.

Všetky programátory bezchybne pracujú pod operačným systémom Windows (pozrite časť Požiadavky na riadiaci počítač).

Ovládací program, ktorý je spoločný pre všetky programátory je jednoduchý s komfortným ovládaním pomocou pull-down menu, skratkovými klávesmi a bohatou nápovedou.

Koncepcia programátorov, ochranné obvody a použité súčiastky dovoľujú poskytnúť na programátory BeeHive204, BeeHive4+, BeeProg2, BeeProg+ a SmartProg2 **3 ročnú záruku** a na programátory MEMprog2, T51prog2, PIKprog2 a SEEprog **2 ročnú záruku**, s obmedzením 25 000 cyklov na päťicu ZIF. Tieto záručné podmienky platia pre užívateľov, ktorý si kúpili programátor priamo od ELNEC s.r.o. Záručné podmienky distribútorov môžu byť iné a závislé od zákonov cieľovej krajiny alebo záručnej politiky distribútora.

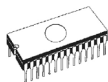
Poznámka: Neodporúčame používať programátory MEMprog2 a SEEprog pre ISP programovanie. Pozrite sekciu FAQ (Otázky a odpovede) na našej internetovej stránke www.elnec.sk.

Konfigurácia výrobkov

Predtým, ako začnete inštalovať a používať Váš programátor, skontrolujte si, prosím dôkladne úplnosť zásielky programátora. Ak nájdete akúkoľvek nezrovnalosť s týmto zoznamom, prípadne, ak je niektorá položka poškodená, obráťte sa ihneď na Vášho distribútora.

| | BeeHive204 BeeHive4+ | BeeProg2 BeeProg+ | SmartProg2 | MEMprog2 | T51prog2 | PIKprog2 | SEEprog |
|------------------------|-------------------------|----------------------|------------|----------|----------|----------|---------|
| programátor | • | • | • | • | • | • | • |
| USB kábel | • | • | • | • | • | • | - |
| LPT kábel | - | * | - | - | - | - | • |
| interný zdroj | • | • | - | - | - | - | - |
| externý zdroj | - | - | • | • | • | • | • |
| diagnostický POD | 1x | • | • | • | • | • | - |
| ISP diagnostický POD | 1x | • | - | - | - | - | - |
| ISP kábel | 4x | • | • | • | • | • | - |
| protiprachový kryt ZIF | 4x | • | • | • | • | • | - |
| software CD | • | • | • | • | • | • | • |
| Návod na obsluhu | • | • | - | - | - | - | - |
| stručný návod | - | - | • | • | • | • | • |
| registračná karta | • | • | • | • | • | • | • |
| prepravný obal | • | • | • | • | • | • | • |

* voľiteľné príslušenstvo



Požiadavky na riadiaci počítač

Minimálne požiadavky

| | 2x BeeHive204 2x BeeHive4+ | BeeHive204 BeeHive4+ | BeeProg2 BeeProg+ | SmartProg2 | MEMprog2 | T51prog2 | PIKprog2 | SEEprog |
|--------------------------|-------------------------------|-------------------------|----------------------|------------|----------|----------|----------|---------|
| OS - Windows | XP | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| CPU | C2D 2,6GHz | P4 | P4 | P4 | P4 | P4 | P4 | P4 |
| RAM [MB] | 1000 | 512 | 256 | 256 | 256 | 256 | 256 | 256 |
| voľné miesto na HDD [MB] | 400 | 200 | 200 | 200 | 200 | 200 | 200 | 200 |
| USB 2.0 high speed | • | • | - | - | - | - | - | - |
| USB 1.1 | - | - | • | • | • | • | • | - |
| LPT | - | - | • | - | - | - | - | • |
| CDROM | • | • | • | • | • | • | • | • |

Odporúčané požiadavky

| | 2x BeeHive204 2x BeeHive4+ | BeeHive204 BeeHive4+ | BeeProg2 BeeProg+ | SmartProg2 | MEMprog2 | T51prog2 | PIKprog2 | SEEprog |
|--------------------------|-------------------------------|-------------------------|----------------------|------------|----------|----------|----------|---------|
| OS - Windows | Win 7 | Win 7 | Win 7 | Win 7 | Win 7 | Win 7 | Win 7 | Win 7 |
| CPU | C2Quad | C2D | C2D | C2D | C2D | C2D | C2D | C2D |
| RAM [MB] | 2000 | 1000 | 1000 | 512 | 512 | 512 | 512 | 512 |
| voľné miesto na HDD [MB] | 2000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| USB 2.0 high speed | • | • | • | • | • | • | • | - |
| LPT IEEE1284 | - | - | • | - | - | - | - | • |

Tieto požiadavky sú platné pre verziu ovládacieho programu 2.84/11.2011. Pre ostatné verzie pozrite **Pomocník/O ovládacom programe**.

Potrebné voľné miesto na hard disku závisí aj na veľkosti použitého obvodu a počte pripojených programovacích siteov. Pre veľké obvody je požadovaná veľkosť voľného miesta **1000MB + 2x veľkosť obvodu x počet programovacích siteov**.

Veľmi jednoduchou indikáciou, či máte počítač s dostatočnou konfiguráciou pre aktuálnu verziu programu a aktuálnu situáciu s Pg4uw / Pg4uwMC je spustiť Windows task manager (Ctrl+Alt+Del) a pozrite sa na kartu Výkon. Zaťaženie procesora má byť max. 80% pri plnom behu programovacieho systému.

Doplnkové služby:

Prečo je dôležité používať najnovšiu verziu ovládacieho programu?

- Výrobcovia polovodičov každú chvíľu uvádzajú na trh nové obvody v nových typoch púzdiar a vyrobené novšími technológiami na podporu flexibilitnosti, kvality a rýchlosti návrhu a výroby výrobkov. Za jeden rok implementujeme do ovládacieho programu podporu na cca 5000 nových obvodov.
- Technológia výroby typického programovateľného obvodu podstúpi niekoľko zmien počas doby životnosti obvodu s cieľom udržať, alebo zlepšiť parametre obvodu a výťažnosť výrobného procesu. Tieto zmeny majú často za následok aj zmenu programovacieho algoritmu (programovací algoritmus je postupnosť príkazov, ktoré hovoria programátoru obvodov, ako naprogramovať dáta do toho ktorého programovateľného obvodu). Použitie najnovších algoritmov pri programovaní je kľúčom k získaniu kvalitného výsledku tohto procesu. V mnohých prípadoch pri programovaní obvodu staršími algoritmiami prebehne programovanie bez chyby, proces však nezaručuje kvalitu programovania, akú je možné dosiahnuť použitím optimálnych algoritmov. Nepoužitie najnovších programovacích algoritmov môže znížiť výťažnosť programovania (zvyšuje sa počet chybne naprogramovaných obvodov), zväčša predlžuje čas programovania a má priamy vplyv na dlhodobé uchovanie dát v programovanom obvode.
- My tiež robíme chyby...

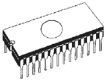
Našou snahou je implementovať podporu pre tieto nové obvody alebo modifikácie starších obvodov tak rýchlo ako sa len dá po ich zverejnení, preto si môžete byť istý, že my používame najnovšie a/alebo optimálne programovacie algoritmy, ktoré boli vytvorené pre tieto obvody.

- bezplatná poradenská služba (hot line) v oblasti programovania
- nárok na bezplatné nové verzie programového vybavenia

Nové verzie ovládacieho programu sú bezplatne k dispozícii na našej internetovej stránke
www.elnec.sk

Novinkami podpory zákazníkov sú služby Keep-Current a AlgOR.

- **Keep-Current** v rámci tejto služby je zasielaná zákazníkovi najnovšia verzia ovládacieho programu a doplnky k návodu na obsluhu a použitie (Keep-Current balíček). Služba Keep-Current zaručuje jednoducho a bez zmätkov aktuálne najvyššiu kvalitu programovania na programátoroch ELNEC. Pre viac informácií pozrite na www.elnec.sk.
- **AlgOR** (Algorithm On Request - algoritmy na požiadanie) je služba, ktorá dáva užívateľovi programátora nástroj, ako vo svoj prospech ovplyvniť počet a sortiment programovateľných obvodov a tiež možnosť podieľať sa na koncepcii a vlastnostiach ovládacieho programu. Pre viac informácií pozrite na www.elnec.sk.



Rýchly štart

Pripojenie programátora k PC

- prepojte USB (alebo paralelný) port programátora a USB (alebo paralelný) port PC dodaným káblom
- priveďte napájanie k programátoru a ak je to potrebné zapnite programátor vypínačom

Inštalácia programového vybavenia

Z dodaného CD s programovým vybavením spustíte inštalačný program (Setup.exe) a riadte sa jeho pokynmi. Nové informácie o hardware a software, ktoré nie sú uvedené v tomto návode nájdete na www.elnec.sk.

Práca s programovým vybavením



Dvakrát kliknite na

Ovládací program po spustení automaticky prehľadá všetky dostupné porty a vyhľadá pripojený ELNEC programátor. Keďže program PG4uw je spoločný pre všetky programátory firmy ELNEC, program zisťuje prítomnosť všetkých podporovaných programátorov

Menu **Súbor** umožňuje základnú manipuláciu so súbormi, nastavenie a prezeranie adresárov, zmenu aktuálneho disku a modifikáciu počiatkovej a koncovkej adresy buffera pre čítanie a ukladanie súborov a načítanie a uloženie projektu.

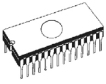
Menu **Buffer** umožňuje prácu s bufferom ako sú blokové operácie, mazanie obsahu, naplnenie reťazcom, 4 druhy kontrolnej sumy, prezeranie a editáciu buffera, ktorá v sebe zahŕňa ďalšie služby (vyhľadávanie reťazcov a ich náhrada, výstup na tlačiareň atď.).

Menu **Obvod** obsahuje funkcie pre prácu so zvoleným programovateľným obvodom - výber obvodu, prečítanie obsahu obvodu do buffera, kontrola vymazania, programovanie, verifikácia naprogramovaných dát, serializácia a nastavenia priradeného súboru.







Menu **Programátor** obsahuje príkazy pre prácu s programátorom.

Menu **Nastavenia** obsahuje príkazy, ktorými je možné prezerat' a menit' rôzne prednastavenia programu.

Menu **Pomocník** slúži na zobrazenie zoznamu podporovaných obvodov a programátorov a tiež obsahuje informácie o verzii programu.

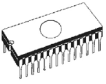


Naprogramovanie obvodu - najkratšia cesta

1. výber obvodu: kliknite na 
2. načítanie dát do buffera:
 - a) zo súboru: kliknite na 
 - b) z obvodu: vložte obvod do ZIF a kliknite na 
3. vložte zvolený obvod do ZIF
4. skontrolujte, či je obvod vymazaný: kliknite na 
5. naprogramovanie obvodu: kliknite na 
6. dodatočná verifikácia obvodu: kliknite na 



Podrobný popis



BeeHive204 / BeeHive4+



Úvod

BeeHive204 je veľmi rýchly 4x48 pinový **concurrent multiprogramovací systém** navrhnutý pre programovanie veľkého množstva obvodov s minimálnou záťažou pre obsluhu. Obvody sa programujú rýchlosťou blízkou maximálnej teoretickej rýchlosti.

BeeHive4+ je rýchly 4x48 pinový **concurrent multiprogramovací systém** navrhnutý pre programovanie veľkého množstva obvodov s minimálnou záťažou pre obsluhu. Obvody sa programujú rýchlosťou blízkou maximálnej teoretickej rýchlosti.

BeeHive204 / BeeHive4+ pozostáva zo štyroch nezávislých izolovaných univerzálnych modulov, založených na programátore BeeProg2 / BeeProg+. Na základe toho, môžu jednotlivé moduly pracovať asynchrónne ("concurrent" programovací mód). Programovací proces každého modulu začína nezávisle od ostatných modulov po detekcii správneho vloženia obvodu do päťce. V konečnom dôsledku počas prevádzky troch modulov môže operátor manipulovať s obvodom štvrtého modulu. Činnosť operátora v tomto prípade spočíva len vo vyberaní a vkladaní obvodov do päťce programátora. Čas školenia operátorov je v tomto prípade minimálny.

Modulárna koncepcia programátora (nezávislá činnosť jednotlivých modulov) umožňuje plynulú prevádzku aj v prípade poruchy jedného modulu. To robí servis jednoduchým a rýchlym.

Asynchrónna a concurrent práca umožňuje začať programovanie obvodov hneď po ich vložení. Operátor iba vyberie naprogramovaný obvod a vloží nový. Zaškoľovanie obsluhy je týmto minimalizované.

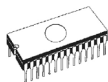
BeeHive204 / BeeHive4+ podporuje všetky typy a technológie programovateľných obvodov dneška aj zajtraška bez použitia špeciálnych prídavných modulov. Na základe toho, že podpora nových obvodov sa deje výlučne **programovo**, prípadne keď to je potrebné aj s použitím jednoduchého konvertora, sú ďalšie **náklady vlastníkov programátorov minimálne**.

Pri použití zabudovaného **ISP konektora**, je možné ISP programovateľné obvody programovať priamo v aplikácii.

BeeHive204 / BeeHive4+ predstavuje vynikajúco technicky spracovaný výrobok, zabezpečujúci spoľahlivé a rýchle programovanie a to za veľmi konkurencieschopnú cenu. Vo svojej triede poskytuje iste najlepší pomer cena/výkon.

BeeHive204 / BeeHive4+ programuje obvody maximálne rýchlo z dôvodu použitia výkonného hardware na báze **vysokorýchlostného FPGA** a tiež vďaka vykonávaniu niektorých časovo náročných činností interne programátorom. BeeHive204 / BeeHive4+ je vo svojej kategórii rýchlejší, ako väčšina konkurenčných programátorov. V konečnom dôsledku, pri použití programátora vo výrobe, je väčšinou čas programovania obvodu kratší ako čas obsluhy programátora operátorom.

BeeHive204 / BeeHive4+ je periférne zariadenie osobného počítača triedy IBM PC, pripojiteľné PC desktop po palmtop pomocou **USB 2.0** portu.



Na **BeeHive204 / BeeHive4+** sa nachádza zdierka pre pripojenie antistatického náramku, čo umožňuje ľahkú implementáciu ESD ochrany.

BeeHive204 / BeeHive4+ má 48 výkonných plne konfigurovateľných TTL pindriverov na báze FPGA, ktoré poskytujú H/L/ pull-up / pull-down a možnosť čítania pre každý pin päťice. Použitie pokročilých kvalitných a vysokorychlostných obvodov poskytujú kvalitné signály bez prekmitov na programovanie všetkých podporovaných obvodov. Pindriver poskytuje napájacie napätie dostatočne nízke na programovanie všetkých dnešných (1.8 V) aj budúcich low-voltage obvodov.

Každý z programovacích modulov umožňuje skontrolovať správnu polohu (posunutie, otočenie) programovaného obvodu v päťici, ako aj to, či má každý pin programovaného obvodu kontakt v päťici. Tieto schopnosti podporené limitáciou programovacích prúdov a kontrolou identifikačných byte programovaného obvodu takmer vylučujú možnosť chyby obsluhy.

BeeHive204 / BeeHive4+ umožňuje kontrolu celkového stavu programátora, vykonaním diagnostických častí obslužného programu.

BeeHive204 / BeeHive4+ má zabezpečovacie obvody, ktoré vylučujú poškodenie programátora a/alebo programovaného obvodu v prípadoch poruchy napájania, výpadku komunikácie s PC alebo chyby operátora. Ochranné obvody na ZIF pinoch chránia programátor aj programovaný obvod pred poškodením v prípade prepätia a elektrostatického výboja až do 15kV.

Programátor **BeeHive204 / BeeHive4+** vykonáva **verifikáciu** naprogramovaných hodnôt pri **marginálnych** hodnotách napájacieho napätia, čo zvyšuje kvalitu programovacieho procesu a dlhodobú stálosť naprogramovaných údajov.

Pre programátor je k dispozícii široký výber konvertorov pre obvody v púzdrach PLCC, SOIC, PSOP, SSOP, TSOP, TSSOP, TQFP, QFN (MLF), SDIP, BGA atď.

K dispozícii je WIN32 ovládací program s prehľadným ovládaním a výkonnými funkciami, podporený bohatou nápovedou. Voľba programovaného obvodu je podľa výrobcov, podľa tried obvodov alebo zadaním fragmentu označenia, uvedeného na obvode.

Štandardné príkazy pre manipuláciu s obvodmi (čítanie, kontrola vymazania, programovanie, verifikácia, výmaz obvodu) sú posilnené o **testovacie funkcie** (test správneho vloženia obvodu do päťice, kontrola elektronickej identifikácie obvodu) a **špeciálne funkcie** (autoinkrementácia a výrobný mód, teda štart programovania ihneď po vložení ďalšieho obvodu do päťice programátora).

Ovládací program podporuje všetky známe formáty súborov. Čítacia funkcia automaticky rozpoznáva formát načítavaného súboru a urobí potrebnú konverziu dát.

Funkcia **autoinkrementácie** zabezpečí modifikáciu sériového čísla v bufferi vždy pred vložením nového obvodu. Autoinkrementačná funkcia umožňuje čítať sériové čísla, resp. individuálnu identifikáciu programovaného obvodu aj zo súboru.

Software poskytuje veľa informácií o programovaných obvodoch, napr. obrázky možných puzdrení obvodu, vysvetlenie popisov na obvode (prefixy a sufíxy) a mnoho ďalších informácií.

Software takisto poskytuje veľa informácií pre ISP programovanie a jeho implementáciu do procesu. Opisuje piny ISP konektora pre zvolený obvod, doporučené zapojenie cieľovej aplikácie pre bezproblémové programovanie zvoleného obvodu a ďalšie dôležité informácie.

Diaľkové ovládanie je vlastnosť, ktorá umožňuje, aby program Pg4uw bol riadený inými aplikáciami – napr. BAT príkazmi, alebo DLL knižnicami. Pre BeeHive204 / beeHive4+ je diaľkové ovládanie obmedzené iba na ISP obvody. Pre automatizované off line programovanie použite BeeHive204AP.

Implementovaný **Jam Player** umožňuje interpretáciu Jam súborov štandardu JEDEC JESD-71 súborov. Jam súbory sú generované softwarom, ktorý dodávajú výrobcovia týchto programovateľných obvodov. Obvody môžu byť programované v ZIF päťici alebo cez ISP konektor (IEEE 1149.1 Joint Test Action Group (JTAG) interface).

Implementovaný **VME Player** umožňuje interpretáciu VME súborov. VME súbor je binárne komprimovaná verzia SVF súboru a obsahuje IEEE 1149.1 busové operácie vyšších úrovní. VME súbory sú generované softwarom, ktorý dodávajú výrobcovia týchto programovateľných obvodov. Obvody môžu byť programované v ZIF päťici alebo cez ISP konektor (IEEE 1149.1 Joint Test Action Group (JTAG) interface).

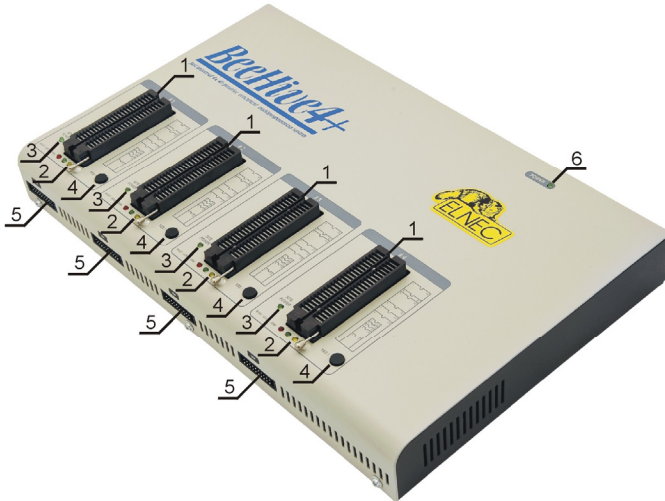
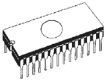
Viacero takýchto obvodov je možné naraz programovať alebo testovať cez JTAG reťaz: JTAG reťaz (ISP-Jam) alebo JTAG reťaz (ISP-VME)

Devízou technického riešenia programátora BeeHive204 / BeeHive4+ je fakt, že podpora na nové obvody sa deje výlučne programovo, pretože BeeHive204 / BeeHive4+ je založený na použití univerzálneho programátora (BeeProg2 / BeeProg+). Pri pružnosti firmy ELNEC sa podpora nových obvodov vykonáva takmer okamžite!

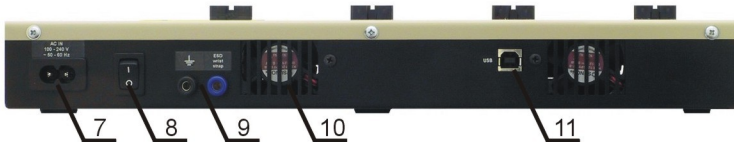
Koncepcia programátora BeeHive204 / BeeHive4+, ochranné obvody a použité súčiastky, ako aj spoľahlivý výrobný proces a zahorovanie dovoľujú poskytnúť na programátor **3-ročnú záruku** s obmedzením 25 000 cyklov na päťice ZIF.

Popis programátora *BeeHive204 / BeeHive4+*

- 1) 48 pinová ZIF (**Z**ero **I**nsertion **F**orce) päťica
- 2) LED indikujúce prácu a výsledok práce s obvodom
- 3) LED indikujúca napájanie a uspanie modulu
- 4) tlačítko YES!
- 5) ISP konektor
- 6) LED indikujúca napájanie programátora



- 7) napájací konektor
- 8) vypínač
- 9) „GND“ konektor môže byť použitý pre uzemnenie programátora
- 10) Teplom riadené ventilátory
- 11) USB konektor typu B na pripojenie USB prepojovacieho kábla PC <-> BeeHive204 / BeeHive4+



Manipulácia s programovanými obvodmi

Po výbere želaného obvodu, vložte ho do ZIF päťice (päčka je hore) a sklopte páčku na vytvorenie kontaktu obvod - ZIF päťica.

Správna orientácia programovaného obvodu v ZIF päťici je znázornená na obrázku vedľa ZIF päťice. Programovaný obvod je nutné vkladať a vyberať zo ZIF päťice iba vtedy, keď nesvieti LED BUSY.

Poznámka: Ochranné obvody programátora chránia programovaný obvod pred krátkodobým aj dlhodobým výpadkom napájacieho napätia, čiastočne aj v prípade poruchy PC. Nezabránia však prípadnému zničeniu programovaného obvodu pri nesprávne zvolených parametroch programovania. Programovaný obvod by nemal byť zničený pri násilnom prerušení behu ovládacieho programu (napr. Reset alebo vypnutie počítača), odpojení prepojovacieho kábla k programátoru, ale môže byť poškodená práve programovaná bunka. Nevyberajte obvod zo ZIF päťice počas práce s obvodom (LED BUSY svieti).

In-system serial programming pomocou BeeHive204 / BeeHive4+

Použitie pokročilé kvalitné a vysokorychlostné obvody pindrivera poskytujú kvalitné signály bez prekmitov na programovanie všetkých podporovaných obvodov. Pindriver poskytuje napájacie napätie dostatočne nízke na programovanie všetkých dnešných (1.8 V) aj budúcich low-voltage obvodov.

BeeHive204 / BeeHive4+ vykonáva **verifikáciu** naprogramovaných hodnôt pri **marginálnych** hodnotách napájacieho napätia, čo zvyšuje kvalitu programovacieho procesu a dlhodobú stálosť naprogramovaných údajov.

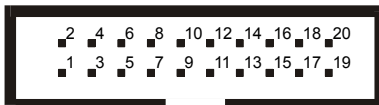
BeeHive204 / BeeHive4+ cez každý ISP konektor umožňuje napájanie cieľového zariadenia, v ktorom programuje cieľový obvod.

Takéto riešenie ISP programovania poskytuje, za veľmi konkurencieschopnú cenu, vynikajúco technicky spracovaný výrobok pre spoľahlivé programovanie.

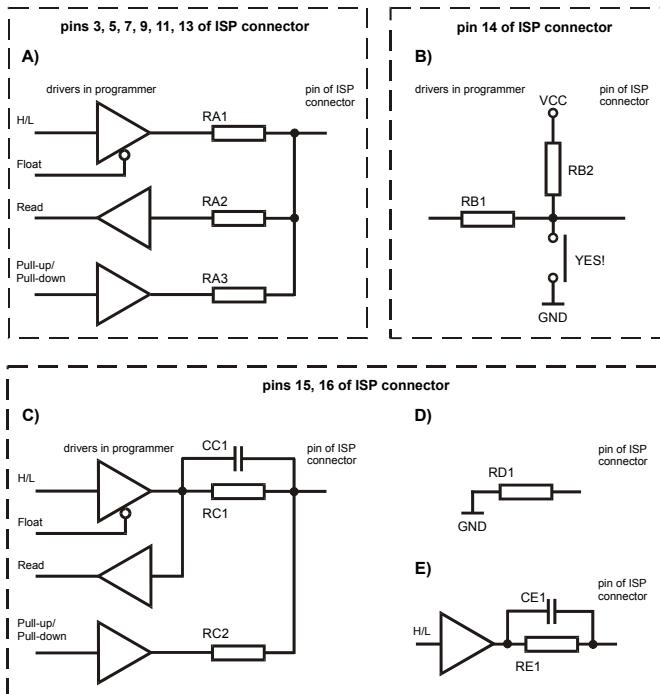
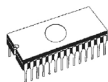
Software poskytuje ucelené informácie pre ISP programovanie: popis jednotlivých pinov ISP konektora, doporučený spôsob pripojenia do cieľového zariadenia s in-circuit programovaným obvodom a ďalšie nevyhnutné informácie.

Definície, odporúčania a pokyny sú popísané v časti **Spoločné poznámky / ISP**.

Opis ISP konektorov programátora BeeHive204 / BeeHive4+



Predný pohľad na ISP konektor programátora.
Driver programátora BeeHive204 / BeeHive4+pre H/L a čítanie



RA1 180R, RA2 1k3, RA3 22k,
RB1 10k, RB2 10k,
CC1 1n, RC1 1k3, RC2 22k,
RD1 22k, CE1 1n, RE1 1k3,

C) Zapojenie pinov 15 a 16, keď sú konfigurované ako logické signály potrebné pre ISP programovanie

D) E) Keď sú piny 15 a 16 konfigurované ako indikátor stavu LED OK a LED ERROR

D) pred prvou akciou s vybratým ISP obvodom

E) po prvej akcii s vybratým ISP obvodom

Poznámka: Keď LED OK alebo LED ERROR svieti, tento stav je prezentovaný ako logické H, úroveň H je 1,8V - 5V v závislosti na H úrovni vybratého ISP obvodu.

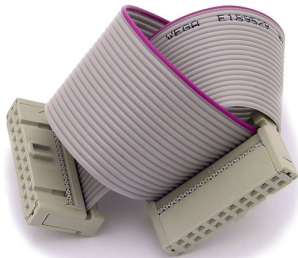
Keď LED OK alebo LED ERROR nesvieti, tento stav je prezentovaný ako logické L, úroveň L je 0V - 0,4V.

Vyššie uvedené hodnoty sú poskytnuté na pochopenie (a tiež na exaktný výpočet) hodnôt rezistorov, ktoré oddelujú programovaný ISP obvod od cieľového systému.

Konkrétny význam pinov ISP konektora je závislý od obvodu, ktorý chcete programovať. Nájdete ho v ovládacom programe Pg4uw, menu **Obvod / Informácie o obvode (Ctrl+F1)**. Skontrolujte, či ste vybrali ISP spôsob programovania želaného obvodu. Je to označené príponou (ISP) za názvom vybratého obvodu.

Význam pinov odpovedá odporúčaniam z aplikačných poznámok, ktoré vydávajú výrobcovia súčiastok. Použité aplikačné poznámky môžete nájsť na www.elnec.sk v časti aplikačné poznámky.

Poznámka: Pin 1 je na konektore ISP káblíka označený trojuholníkovým zárezom.



ISP káblík programátora BeeHive204 / BeeHive4+

Upozornenie:

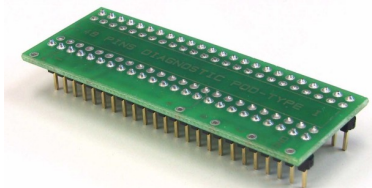
- Používajte iba **priložený ISP káblík**. Použitie iného ISP káblíka (iný materiál, dĺžka...) môže byť príčinou nespoľahlivého programovania.
- **BeeHive204 / BeeHive4+** môže napájať programovaný obvod (pin 1 ISP konektora), a cieľové zariadenie (piny 19 a 20 ISP konektora) s obmedzením (pozri Technická špecifikácia / ISP konektor).
- **BeeHive204 / BeeHive4+** privedie programovacie napätie na cieľový obvod a overí si jeho hodnotu (cieľový systém môže ovplyvňovať programovacie napätie). Ak je programovacie napätie odlišné ako požadované, práca s obvodom bude zrušená.

Test a kalibrácia programátora

Ak máte pocit, že sa Váš programátor nespráva úplne podľa Vašich očakávaní, vykonajte selftest programátora (ISP konektora) pomocou diagnostickej hlavice (Diagnostic POD for ISP connectors #2), dodávanej štandardne s programátorom.

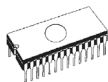
Selftest programátora

- Vložte **48 pins diagnostic POD - type I** do ZIF päťice modulu. **48 pins diagnostic POD - type I** musí byť vložený ako 48 pinový obvod.
- V Pg4uw spustíte selftest programátora (Programátor / Selftest plus)

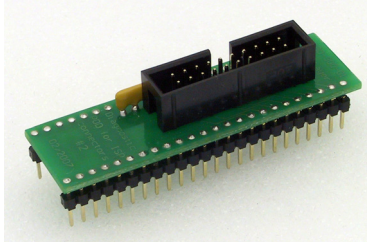


Selftest ISP konektora

- Vložte **Diagnostic POD for ISP connectors #2** do ZIF päťice modulu. **Diagnostic POD for ISP connectors #2** musí byť vložený ako 48 pinový obvod.

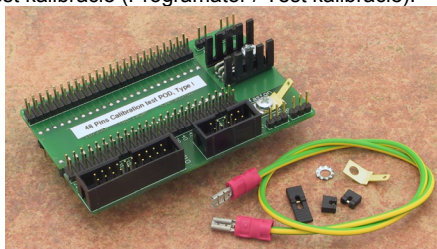


- Prepojte 20 pinový konektor **Diagnostic POD for ISP connectors #2** s ISP konektorom toho istého modulu ISP káblom štandardne dodávaným s programátorom. Skontrolujte správnosť prepojenia (napr. 1-1, 2-2, ..., 20-20).
- V Pg4uw spustíte selftest ISP konektora (Programátor / Selftest ISP konektora)



Test kalibrácie

- Vložte **48 Pins Calibration test POD, Type I** do ZIF päťice modulu. **48 Pins Calibration test POD, Type I** musí byť vložený ako 48 pinový obvod.
- V Pg4uw spustíte test kalibrácie (Programátor / Test kalibrácie).



Technická špecifikácia

Špecifikácia (BeeHive204 / BeeHive4+ systém násobného programovania)

- 4x univerzálny programovací modul (4x päťica 48-pin DIL ZIF)
- indikácia stavov operácií pomocou LED diód, LED power
- port USB 2.0 high speed kompatibilný
- zdierka napájania 100-240VAC/60W max.
- zdierka pre pripojenie antistatického náramku
- zdierka pre pripojenie k uzemňovacej časti

Špecifikácia (platná pre všetky programovacie moduly)

HARDWARE

Programátor, prevodníky

- USB 2.0 high speed kompatibilný port, prenosová rýchlosť do 480 Mbit/s

- vlastná inteligencia tvorená výkonným mikroprocesorom s podporou FPGA stavového automatu
- tri D/A prevodníky pre VCCP, VPP1 a VPP2, s riadenou strmostou nábežnej a zostupnej hrany
- rozsah VCCP: 0..8V/1A
- rozsah VPP1, VPP2: 0..26V/1A
- autokalibrácia
- schopnosť selftestu

Programovacia ZIF päťica, pindrivery

- 48-pinový DIL ZIF (Zero Insertion Force) sokel pre obvody do 48 pinov so šírkou púzdra 300/600 mil
- pindrivery: 48 univerzálne
- VCCP/VPP1/VPP2 pripojiteľné na všetky piny
- kvalitné pripojenie GND pre každý pin
- TTL driver realizovaný pomocou FPGA poskytuje H, L, CLK, pull-up, pull-down na všetkých pinoch pindrivery
- napäťové úrovne analógovej časti pindrivery nastaviteľné od 1.8 V do 26V
- prúdové obmedzenie, vypnutie pri prekročení prúdu a výpadku napájacieho napätia
- ESD prepäťová ochrana všetkých vývodov päťice (IEC1000-4-2: 15kV vzduchom, 8kV kontaktom)
- test vodivosti programátora: všetky piny sú testované pred každou programovacou procedúrou

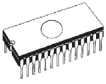
ISP konektor

- 20-pinová vidlica s kľúčom
- 6 TTL pindrivery, poskytujúce H, L, CLK, pull-up, pull-down; úroveň H nastaviteľná v rozmedzí 1.8V až 5V k práci so všetkými (vrátane nízkonapäťovými) zariadeniami
- 1x VCCP napätie (rozsah 2V..7V/100mA), pripojiteľné na piny 1, 3
- programovacie napätie (VCCP) s obidvoma úrovňami a detekciou napätia
- 1x VPP napätie (rozsah 2V..25V/50mA), pripojiteľné na piny 2,3,4,6,8,10
- napájanie cieľovej aplikácie (rozsah 2V..6V/250mA)
- ESD ochrana všetkých vývodov ISP konektora (IEC1000-4-2: 15kV vzduchom, 8kV kontaktom)
- 2 výstupné signály, ktoré indikujú výsledok operácie = LED OK a LED Error (aktívna úroveň min 1,8V)
- Vstupný signál, ktorý je ekvivalent tlačidla YES! (aktívna úroveň max. 0,8V)

PROGRAMOVANÉ OBVODY

Programátor, v ZIF päťici

- EPROM: NMOS/CMOS, 27xxx a 27Cxxx, s 8/16 bitovou dátovou zbernicou, plná podpora LV obvodov
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx, s 8/16 bitovou dátovou zbernicou
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx, od 256Kbit do 1Gbit, s 8/16 bitovou dátovou zbernicou, plná podpora LV obvodov
- Sériové E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 45Dxxx, 59Cxxx, 25Fxxx, 25Pxxx, 85xxx, 93Cxxx, NVM3060, MDAXxx, plná podpora LV obvodov
- Konfigurovateľné (EE)PROM: XCFxxx, XC17xxxx, XC18Vxxx, EPCxxx, AT17xxx, 37LVxx



- 1-Wire E(E)PROM: DS1xxx, DS2xxx
- PROM: AMD, Harris, National, Philips/Signetics, Tesla, TI
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x
- PLD: Altera: MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 7000AE, MAX II
- PLD: Lattice: ispGAL22V10x, ispLSI1xxx, ispLSI1xxxEA, ispLSI2xxx, ispLSI2xxxA, ispLSI2xxxE, ispLSI2xxxV, ispLSI2xxxVE, ispLSI2xxxVL, LC4xxxB/C/V/ZC, M4-xx/xx, M4A3-xx/xx, M4A5-xx/xx, M4LV-xx/xx
- PLD: Xilinx: XC9500, XC9500XL, XC9500XV, CoolRunner XPLA3, CoolRunner-II
- ostatné PLD: SPLD/CPLD series: AMI, Atmel, AMD-Vantis, Gould, Cypress, ICT, Lattice, NS, Philips, STM, VLSI, TI
- Mikroprocesory rady 48: 87x41, 87x42, 87x48, 87x49, 87x50
- Mikroprocesory rady 51: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, všetci výrobcovia, rada Philips LPC
- Mikroprocesory Intel rady 196: 87C196 KB/KC/KD/KT/KR/...
- Mikroprocesory Atmel AVR: AT90Sxxxx, ATtiny, ATmega
- Mikroprocesory Cypress: CY7Cxxxxx, CY8Cxxxxx
- Mikroprocesory ELAN: EM78Pxxx
- Mikroprocesory MDT 1xxx and 2xxx
- Mikroprocesory Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17Cxxx, PIC18xxx, PIC24xxx, dsPIC
- Mikroprocesory Motorola (Freescale): 68HC05, 68HC08, 68HC11, HCS08, HCS12
- Mikroprocesory Myson MTV2xx, 3xx, 4xx a 5xx
- Mikroprocesory National: COP8xxx
- Mikroprocesory NEC: uPD78Fxxx
- Mikroprocesory Novatek: NT68xxx
- Mikroprocesory Scenix (Ubicom): SXxxx
- Mikroprocesory SGS-Thomson: ST6xx, ST7xx, ST10xx, STR7xx
- Mikroprocesory TI: MSP430 a MSC121x
- Mikroprocesory ZILOG: Z86/Z89xxx and Z8xxx
- Mikroprocesory ostatné: EM Microelectronic, Fujitsu, Goal Semiconductor, Hitachi, Holtek, Princeton, Macronix, Winbond, Infineon(Siemens), Samsung, Toshiba, ...

I.C. Tester

- Statické RAM: 6116 .. 624000

Programátor, cez ISP konektor

- Sériové E(E)PROM: rada IIC, rada MW, rada SPI, rada KEELOQ, sériové data Flash, PLD konfigurovateľné pamäte
- Mikroprocesory Atmel: AT89Sxxx, AT90Sxxxx, ATtiny, ATmega
- Mikroprocesory Cypress: CY8C2xxxx
- Mikroprocesory Elan: EM78Pxxx, EM6xxx
- Mikroprocesory EM Microelectronic: 4 a 8 bitové rady
- Mikroprocesory Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17xxx, PIC18xxx, PIC24xxx, dsPIC
- Mikroprocesory Motorola/Freescale: HC11, HC908 (5-wire a All-wire), HCS08, HCS12
- Mikroprocesory NEC: uPD7xxx
- Mikroprocesory Philips: LPC2xxx, LPC, 89xxx
- Mikroprocesory Scenix (Ubicom): SXxxx
- Mikroprocesory TI: MSP430 (JTAG a rada BSL), MSC12xxx
- PLD: Lattice: ispGAL22xV10x, ispLSI1xxxEA, ispLSI2xxxE, ispLSI2xxxV, ispLSI2xxxVE, ispLSI2xxxVL, M4-xx/xx, M4LV-xx/xx, M4A3-xx/xx, M4A5-xx/xx, LC4xxxB/C/V/ZC

- rôzne PLD (tiež s podporou JAM player/JTAG):
- Altera: MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 9000, MAX II
- Xilinx: XC9500, XC9500XL, XC9500XV, CoolRunner XPLA3, CoolRunner-II

Poznámka:

Aktuálny zoznam všetkých podporovaných obvodov je na www.elnec.sk

Podporované púzdra

- podpora pre všetky obvody v DIP púzdrení bez prídavnej redukcie
- podpora púzdrenia DIP, SDIP, PLCC, JLCC, SOIC, SOP, PSOP, SSOP, TSOP, TSOPII, TSSOP, QFP, PQFP, TQFP, VQFP, QFN (MLF), SON, BGA, EBGA, FBGA, VFBGA, UBG, FTBGA, LAP, CSP, SCSP atď.
- ostatné púzdrenia do 48 pinov podporované pomocou univerzálnych konvertorov
- programátor je kompatibilný aj s konvertormi iných výrobcov

Rýchlosť programovania

BeeHive204

| Obvod | Veľkosť [bits] | Operácia | čas |
|-----------------------------------|----------------------|-----------------------------|-----------|
| Am29DL640G (paralelná NOR Flash) | 400080Hx16 (64 Mega) | programovanie a verifikácia | 24 sek |
| K8P6415UQB (paralelná NOR Flash) | 400100Hx16 (64 Mega) | programovanie a verifikácia | 13 sek |
| K9F1G08U0M (paralelná NAND Flash) | 8400000Hx8 (1 Giga) | programovanie a verifikácia | 122.7 sek |
| QB25F640S33 (sériová Flash) | 800200Hx8 (64 Mega) | programovanie a verifikácia | 30.7 sek |
| AT89C51RD2 (mikroprocesor) | 10000Hx8 | programovanie a verifikácia | 14.4 sek |
| PIC32MX360F512L (mikroprocesor) | 80000Hx8 | programovanie a verifikácia | 16.2 sek |

Konfigurácia: P4, 2.4GHz, 512MB RAM, USB2.0, Windows XP

BeeHive4+

| Obvod | Veľkosť [bits] | Operácia | čas |
|-----------------------------------|----------------------|-----------------------------|---------|
| M50FW080 (paralelná Flash) | 100000Hx8 (8 Mega) | programovanie a verifikácia | 22 sek |
| MX28F640C3BT (paralelná Flash) | 400000Hx16 (64 Mega) | programovanie a verifikácia | 57 sek |
| K9F1G08U0M (paralelná NAND Flash) | 8400000Hx8 (1 Giga) | programovanie a verifikácia | 239 sek |
| AT45D081 (sériová Flash) | 108000Hx8 (16 Mega) | programovanie a verifikácia | 36 sek |
| AT89C51RD2 (mikroprocesor) | 10000Hx8 | programovanie a verifikácia | 15 sek |
| PIC18LF452 (mikroprocesor) | 4000Hx16 | programovanie a verifikácia | 4 sek |

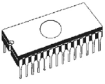
Konfigurácia: P4, 2.4GHz, 512MB RAM, USB2.0, Windows XP

Poznámky:

- Rýchlosti programovania našich výrobkov testujeme na vzorkách náhodných dát, na rozdiel od niektorých konkurenčných výrobcov používajúcich vzorky s "riedkymi" dátami obsahujúcimi len časť kvalitatívnych dát na programovanie a dáta obsahujúce niekoľko nulových bitov (FE, EF, atď.). Takouto úpravou vzoriek je možné do značnej miery znížiť programovacie časy jednotlivých obvodov. Pri porovnávaní produktov z tohto hľadiska si vždy vyžiadajte vzorky dát na ktorých boli tieto produkty testované.
- Rýchlosť programovania čiastočne závisí od rýchlosti PC a od vyťaženia operačného systému.

SOFTWARE

- **Algoritmy:** použité sú výhradne výrobcom obvodov schválené alebo odporúčané algoritmy.
- **Nové verzie SW:** nové verzie programového vybavenia sú k dispozícii približne každé 4 týždne, bezplatne.
- **Základné vlastnosti:** história revízií, protokol práce, on-line nápoveda, informácie o obvodoch a algoritmoch.



Činnosti s obvodom

• štandardné:

- inteligentný výber obvodu podľa typu, výrobcu alebo časti mena obvodu
- automatický výber EPROM/Flash EPROM obvodov podľa ID
- kontrola vymazania, čítanie, verifikácia
- programovanie
- výmaz
- programovanie konfiguračných a zabezpečovacích bitov
- illegal bit test
- kontrolná suma
- interpretuje Jam Standard Test and Programming Language (STAPL), JEDEC standard JESD-71
- interpretuje VME/SVF súbory

• zabezpečovacie

- kontrola správneho vloženia, spätná kontrola vloženia
- kontrola kontaktu ZIF - obvod
- kontrola identifikačných (ID) byte obvodu

• špeciálne

- výrobný mód (automatický štart programovania po vložení obvodu)
- niekoľko serializačných módov (viac typov inkrementálnych módov, from-file mód, custom generator mód)
- štatistika
- count-down mód

Činnosti s bufferom

- view/edit, find/replace
- fill, copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Súbor načítať/uložiť

- žiadny čas na download, pretože programátor je riadený PC
- automatické rozpoznávanie formátu súborov

Akceptované formáty súborov

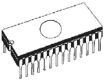
- neformátovaný (raw) binárny
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-SPACE-HEX
- ALTERA POF, JEDEC (ver. 3.0.A), napr. od ABEL, CUPL, PALASM, TANGO PLD, OrCAD PLD, PLD Designer ISDATA, atď.
- JAM (JEDEC STAPL Format), JBC (Jam STAPL Byte Code), STAPL (STAPL File) JEDEC standard JESD-71
- VME (ispVME súbor VME2.0/VME3.0)

VŠEOBECNÉ

- prevádzkové napájanie: 110-250V AC
- spotreba elektrickej energie max. 60W aktívny režim



- rozmery 361x234x56mm
- hmotnosť (programátor) 3.5kg
- pracovný teplotný obsah 5°C + 40°C
- pracovná relatívna vlhkosť 20%..80%, nekondenzujúca



BeeProg2 / BeeProg+



Úvod

BeeProg2 je veľmi rýchly, univerzálny programátor s USB/LPT pripojením, ktorý bol vyvinutý tak aby spĺňal prísne požiadavky všetkých vrstiev vývojárov a programovacích centier na rýchly a spoľahlivý programátor.

BeeProg+ je rýchly, univerzálny programátor s USB/LPT pripojením, ktorý bol vyvinutý tak aby spĺňal prísne požiadavky všetkých vrstiev vývojárov a programovacích centier na rýchly a spoľahlivý programátor.

BeeProg2 / BeeProg+ podporuje všetky typy a technológie programovateľných obvodov dneška aj najzajrjška bez použitia špeciálnych prídavných modulov, čo dáva voľnosť pri použití optimálneho obvodu do vyvíjaného zariadenia. Pri použití zabudovaného **ISP konektora**, je možné ISP programovateľné obvody programovať priamo v aplikácii.

BeeProg2 / BeeProg+ nie je len programátor, ale taktiež **tester** logických integrovaných obvodov a pamätí TTL alebo CMOS, pre ktoré navyiac umožňuje generovať užívateľsky definovateľné testovacie sekvencie.

BeeProg2 / BeeProg+ predstavuje vynikajúco technicky spracovaný výrobok, zabezpečujúci spoľahlivé a rýchle programovanie a to za veľmi konkurencieschopnú cenu. Vo svojej triede poskytujú iste najlepší pomer cena/výkon.

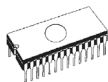
BeeProg2 / BeeProg+ programuje obvody maximálne rýchlo z dôvodu použitia výkonného hardware na báze vysokorýchlostného FPGA a tiež vďaka vykonávaniu niektorých časovo náročných činností interne programátorom. BeeProg2 / BeeProg+ je vo svojej kategórii rýchlejší, ako väčšina konkurencie. V konečnom dôsledku, pri použití programátora vo výrobe, je väčšinou čas programovania obvodu kratší ako čas obsluhy programátora operátorom.

BeeProg2 / BeeProg+ je periférne zariadenie osobného počítača triedy IBM PC, pripojiteľné PC desktop po palmtop pomocou USB (2.0/1.1) portu alebo pomocou štandardného paralelného (printer) portu, prípadne pomocou (ECP/EPP) paralelného portu na PCI zbernici, kompatibilného s IEEE 1284. Podpora USB/LPT portov vám umožňuje vybrať si spôsob pripojenia BeeProg2 / BeeProg+ k ľubovoľnému PC, od najnovších notebookov až po staršie desktopy nepodporujúce USB port.

Na **BeeProg2 / BeeProg+** sa nachádza zdieľka pre pripojenie antistatického náramku, čo umožňuje ľahkú implementáciu ESD ochrany.

BeeProg2 / BeeProg+ má 48 výkonných plne konfigurovateľných TTL pindriverov na báze FPGA, ktoré poskytujú H/L/pull-up/pull-down a možnosť čítania pre každý pin päťice. Použitie pokročilé kvalitné a vysokorýchlostné obvody poskytujú kvalitné signály bez prekmitov na programovanie všetkých podporovaných obvodov. Pindriver poskytuje napájacie napätie dostatočne nízke na programovanie všetkých dnešných (1.8 V) aj budúcich low-voltage obvodov.

BeeProg2 / BeeProg+ umožňuje skontrolovať **správnu polohu** (posunutie, otočenie) programovaného obvodu v päťici, a aj to, či má každý pin programovaného obvodu **kontakt** v



päťici. Tieto schopnosti podporené **limitáciou programovacích prúdov a kontrolou identifikačných byte** programovaného obvodu takmer vylučujú možnosť chyby obsluhy.

BeeProg2 / BeeProg+ umožňuje kontrolu celkového stavu programátora, vykonaním diagnostických častí obsluhujúceho programu.

BeeProg2 / BeeProg+ má zabezpečovacie obvody, ktoré vylučujú poškodenie programátora a/alebo programovaného obvodu v prípadoch poruchy napájania, výpadku komunikácie s PC alebo chyby operátora. Ochranné obvody na ZIF pinoch chránia programátor aj programovaný obvod pred poškodením v prípade prepätia a ESD až do 15kV.

Programátor **BeeProg2 / BeeProg+** vykonáva **verifikáciu** naprogramovaných hodnôt pri **marginálnych** hodnotách napájacieho napätia, čo zvyšuje kvalitu programovacieho procesu a dlhodobú stálosť naprogramovaných údajov.

Pre programátor je k dispozícii široký výber konvertorov pre obvody v púzdrach PLCC, SOIC, PSOP, SSOP, TSOP, TSSOP, TQFP, QFN (MLF), SDIP, BGA atď.

K dispozícii je WIN32 ovládací program s prehľadným ovládaním a výkonnými funkciami, podporený bohatou nápovedou. Voľba programovaného obvodu je podľa výrobcov, podľa tried obvodov alebo zadaním fragmentu označenia, uvedeného na obvode.

Štandardné príkazy pre manipuláciu s obvody (čítanie, kontrola vymazania, programovanie, verifikácia, výmaz obvodu) sú posilnené o **testovacie funkcie** (test správneho vloženia obvodu do päťice, kontrola elektronickej identifikácie obvodu) a **špeciálne funkcie** (autoinkrementácia a výrobný mód, teda štart programovania ihneď po vložení ďalšieho obvodu do päťice programátora).

Ovládací program podporuje všetky známe formáty súborov. Čítacia funkcia automaticky rozpoznáva formát načítavaného súboru a urobí potrebnú konverziu dát.

Funkcia **autoinkrementácie** zabezpečí modifikáciu sériového čísla v bufferi vždy pred vložení nového obvodu. Autoinkrementačná funkcia umožňuje čítať sériové čísla, resp. individuálnu identifikáciu programovaného obvodu aj zo súboru.

Software poskytuje veľa informácií o programovaných obvodoch, napr. obrázky možných puzdrení obvodu, vysvetlenie popisov na obvode (prefixy a sufixy) a mnoho ďalších informácií.

Software takisto poskytuje veľa informácií pre ISP programovanie a jeho implementáciu do procesu. Opisuje piny ISP konektora pre zvolený obvod, doporučené zapojenie cieľovej aplikácie pre bezproblémové programovanie zvoleného obvodu a ďalšie dôležité informácie.

Vzdialená správa je vlastnosť, ktorá umožňuje, aby program Pg4uw bol riadený inými aplikáciami – napr. BAT príkazmi, alebo DLL knižnicami. DLL knižnica, príklady (C/PAS/VBASIC/.NET) a návod na používanie je štandardnou súčasťou software.

Implementovaný **Jam Player** umožňuje interpretáciu Jam súborov štandardu JEDEC JESD-71 súborov. Jam súbory sú generované softwarom, ktorý dodávajú výrobcovia týchto programovateľných obvodov. Obvody môžu byť programované v ZIF päťici alebo cez ISP konektor (IEEE 1149.1 Joint Test Action Group (JTAG) interface).

Implementovaný **VME Player** umožňuje interpretáciu VME súborov. VME súbor je binárne komprimovaná verzia SVF súboru a obsahuje IEEE 1149.1 busové operácie vyšších úrovní. VME súbory sú generované softwarom, ktorý dodávajú výrobcovia týchto programovateľných obvodov. Obvody môžu byť programované v ZIF páťici alebo cez ISP konektor (IEEE 1149.1 Joint Test Action Group (JTAG) interface). Viacero takýchto obvodov je možné naraz programovať alebo testovať cez: JTAG reťaz (ISP-Jam) alebo JTAG reťaz (ISP-VME)

Pripojením viac programátorov **BeeProg2 / BeeProg+** na to isté PC cez USB porty je možné získať výkonný **multiprogramovací systém s rovnakým počtom podporovaných obvodov**, ako BeeProg2 / BeeProg+ a s **vysokou rýchlosťou programovania**. V tomto prípade ide o tzv. "concurrent multiprogramming" t.j. každý programátor pracuje nezávisle a môžu byť (pokiaľ je to potrebné) programované rozdielne obvody.

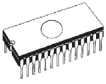
Devízou technického riešenia programátora BeeProg2 / BeeProg+ je fakt, že podpora na nové obvody sa deje výlučne programovo. Pri pružnosti firmy ELNEC sa podpora nových obvodov vykonáva takmer okamžite!

Koncepcia programátora BeeProg2 / BeeProg+, ochranné obvody a použité súčiastky, ako aj spoľahlivý výrobný proces a zahorovanie dovoľujú poskytnúť na programátor **3-ročnú záruku** s obmedzením 25 000 cyklov na páťice ZIF.

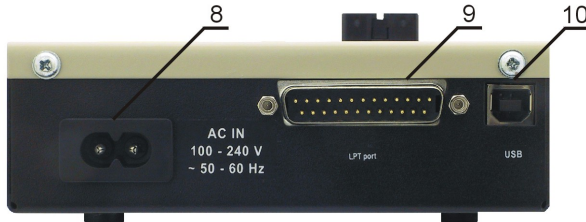
Popis programátora BeeProg2 / BeeProg+

- 1) 48 pinová ZIF (Zero Insertion Force) páťica
- 2) LED indikujúce prácu a výsledok práce s obvodom
- 3) LED indikujúca napájanie a uspanie modulu
- 4) tlačítko YES!
- 5) ISP konektor
- 6) vypínač
- 7) „GND“ konektor môže byť použitý pre uzemnenie programátora
"ESD wrist strap" konektor pre pripojenie antistatického náramku





- 8) napájací konektor
- 9) LPT konektor na pripojenie LPT prepojovacieho kábla PC <-> BeeProg2 / BeeProg+
- 10) USB konektor typu B na pripojenie USB prepojovacieho kábla PC <-> BeeProg2 / BeeProg+



Pripojenie BeeProg2 / BeeProg+ k PC

Cez USB port

Pri tomto pripojení je poradie pripájania USB kábla a napájacieho zdroja nerelevantné.

Cez LPT port

Vypnite PC a prepojovací kábel PC<->programátor, ktorý je súčasťou dodávky programátora BeeProg2 / BeeProg+, zapojte do počítača do konektora, ktorý je určený pre pripojenie tlačiarne (printer port). V prípade, že váš počítač obsahuje iba jeden printer port, odpojte od počítača kábel k tlačiarne a na jeho miesto zapojte kábel programátora. Ak má váš počítač viac printer portov, pripojte programátor na nevyužitý printer port. Opačný koniec kábla pripojte ku programátoru. Obidva konektory je potrebné priskrutkovať do protikusov, zvlášť kritické je to pre konektor vedúci do programátora. Manipulácia zámény kábla k tlačiarne za kábel k programátoru je iste nepríjemná, neodporúčame však prevádzkovať programátor BeeProg2 / BeeProg+ cez mechanický printer switch. Prevádzka cez elektronický prepínač nie je možná. Kvôli pohodlnosti Vám navrhujeme používať ďalšiu I/O kartu obsahujúcu ďalší LPT port, čím budete mať pripojenú tlačiareň na LPT1, zatiaľ čo programátor na LPT2.

Prívodný kábel najprv zapojte do programátora a potom do sieťovej zásuvky. Zapnite PC. Na programátore začne svietiť LED POWER plnou intenzitou a postupne zablikajú všetky indikačné LED na znak selftestu programátora po zapnutí. Keď začne LED POWER svietiť tlme, programátor BeeProg2 / BeeProg+ je pripravený komunikovať s ovládacím programom v PC.

Spustíte ovládací program pre BeeProg2 / BeeProg+.

Upozornenie! Ak sa rozhodnete pri pripájaní alebo odpájaní programátora BeeProg2 / BeeProg+ nevypínať PC (komu by sa chcelo), dodržte prosím nasledovnú postupnosť:

- **Pripájanie:** NAJPRV kábel k PC a POTOM napájanie
- **Odpájanie:** NAJPRV napájanie a POTOM kábel k PC

Z hľadiska programátora BeeProg2 / BeeProg+ je postup pripájania a odpájania káblov neďôležitý - ochranné obvody na vstupoch do programátora zabráni poškodeniu obvodov programátora pri akejkoľvek manipulácii. **Odporúčame ale myslieť aj na Váš PC...**

Problémy prepojenia BeeProg2 / BeeProg+ ⇔ PC a ich odstránenie

Ak máte problémy s prepojením BeeProg2 / BeeProg+ ⇔ PC, prosím pozrite časť **Spoločné poznámky**.

Manipulácia s programovanými obvodmi

Po výbere želaného obvodu, vložte ho do ZIF päťice (páčka je hore) a sklopte páčku na vytvorenie kontaktu obvod - ZIF päťica.

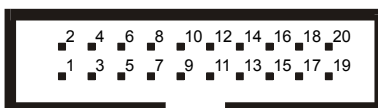
Správna orientácia programovaného obvodu v ZIF päťici je znázornená na obrázku vedľa ZIF päťice. Programovaný obvod je nutné vkladať a vyberať zo ZIF päťice iba vtedy, keď nesvieti LED BUSY.

Poznámka: *Ochranné obvody programátora chránia programovaný obvod pred krátkodobým aj dlhodobým výpadkom napájacieho napätia, čiastočne aj v prípade poruchy PC. Nezabránia však prípadnému zničeniu programovaného obvodu pri nesprávne zvolených parametroch programovania. Programovaný obvod by nemal byť zničený pri násilnom prerušení behu ovládacieho programu (napr. Reset alebo vypnutie počítača), odpojení prepojovacieho kábla k programátoru BeeProg2 / BeeProg+, ale môže byť poškodená práve programovaná bunka. Nevyberajte obvod zo ZIF päťice počas práce s obvodom (LED BUSY svieti).*

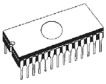
In-system serial programming pomocou BeeProg2 / BeeProg+

Definície, odporúčania a pokyny sú popísané v časti **Spoločné poznámky / ISP**.

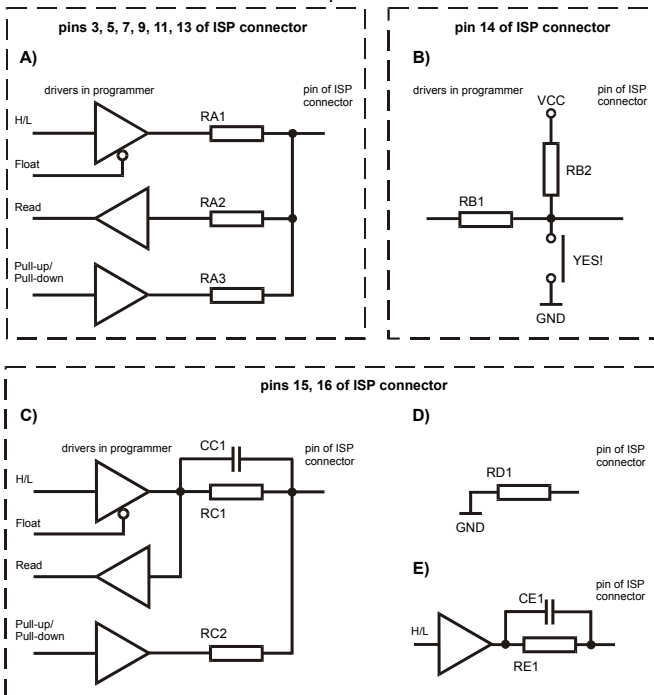
Opis ISP konektorov programátora BeeProg2 / BeeProg+



Predný pohľad na ISP konektor programátora.



Driver programátora BeeHive204 / BeeHive4+pre H/L a čítanie



RA1 180R, RA2 1k3, RA3 22k,
RB1 10k, RB2 10k,
CC1 1n, RC1 1k3, RC2 22k,
RD1 22k, CE1 1n, RE1 1k3,

C) Zapojenie pinov 15 a 16, keď sú konfigurované ako logické signály potrebné pre ISP programovanie

D) E) Keď sú piny 15 a 16 konfigurované ako indikátor stavu LED OK a LED ERROR

D) pred prvou akciou s vybraným ISP obvodom

E) po prvej akcii s vybraným ISP obvodom

Poznámka: Keď LED OK alebo LED ERROR svieti, tento stav je prezentovaný ako logické H, úroveň H je 1,8V - 5V v závislosti na H úrovni vybraného ISP obvodu.

Keď LED OK alebo LED ERROR nesvieti, tento stav je prezentovaný ako logické L, úroveň L je 0V - 0,4V.

Vyššie uvedené hodnoty sú poskytnuté na pochopenie (a tiež na exaktný výpočet) hodnôt rezistorov, ktoré oddeľujú programovaný ISP obvod od cieľového systému.

Konkrétny význam pinov ISP konektora je závislý od obvodu, ktorý chcete programovať. Nájdete ho v ovládacom programe Pg4uw, menu **Obvod / Informácie o obvode (Ctrl+F1)**. Skontrolujte, či ste vybrali ISP spôsob programovania želaného obvodu. Je to označené príponou (ISP) za názvom vybraného obvodu.

Význam pinov odpovedá odporúčaniam z aplikačných poznámok, ktoré vydávajú výrobcovia súčiastok. Použité aplikačné poznámky môžete nájsť na www.elnec.sk v časti aplikačné poznámky.

Poznámka: Pin 1 je na konektore ISP káblíka označený trojuholníkovým zárezom.



ISP káblík programátora BeeProg2 / BeeProg+

Upozornenie:

- Používajte iba **priložený ISP káblík**. Použitie iného ISP káblíka (iný materiál, dĺžka...) môže byť príčinou nespoľahlivého programovania.
- **BeeProg2 / BeeProg+** môže napájať programovaný obvod (pin 1 ISP konektora), a cieľové zariadenie (piny 19 a 20 ISP konektora) s obmedzením (pozri Technická špecifikácia / ISP konektor).
- **BeeProg2 / BeeProg+** privedie programovacie napätie na cieľový obvod a overí si jeho hodnotu (cieľový systém môže ovplyvňovať programovacie napätie). Ak je programovacie napätie odlišné ako požadované, práca s obvodom bude zrušená.

Multiprogramovanie pomocou BeeProg2 / BeeProg+

Počas inštalácie Pg4uw v okne „Vyberte ďalšie úlohy“ skontrolujte, či je povolená inštalácia podpory multiprogramovania.

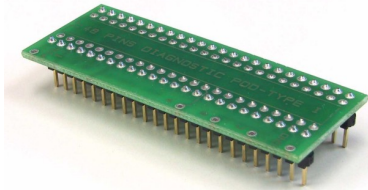
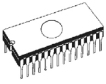
Na spustenie multiprogramovania pomocou BeeProg2 / BeeProg+ je nutné spustiť program **Pg4uwMC.exe**. V tomto programe užívateľ priraduje BeeProg2 / BeeProg+ jednotlivým ovládacím programom, môže načítať projekty pre všetky BeeProg2 / BeeProg+ a spúšťať ovládacie programy pre pripojené a priradené programátory.

Test a kalibrácia programátora

Ak máte pocit, že sa Váš programátor nespráva úplne podľa Vašich očakávaní, vykonajte selftest programátora (ISP konektora) pomocou diagnostickej hlavice 48 pins diagnostic POD –type I (Diagnostic POD for ISP connectors #2), dodávanej štandardne s programátorom.

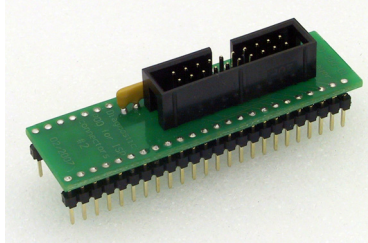
Selftest programátora

- Vložte **48 pins diagnostic POD - type I** do ZIF päťice modulu. **48 pins diagnostic POD - type I** musí byť vložený ako 48 pinový obvod.
- V Pg4uw spustíte selftest programátora (Programátor / Selftest plus)



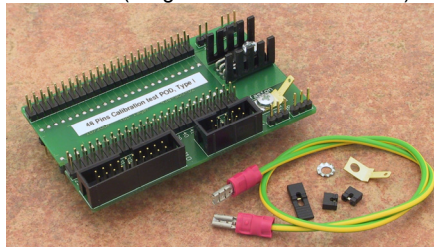
Selftest ISP konektora

- Vložte **Diagnostic POD for ISP connectors #2** do ZIF päťice modulu. **Diagnostic POD for ISP connectors #2** musí byť vložený ako 48 pinový obvod.
- Prepojte 20 pinový konektor **Diagnostic POD for ISP connectors #2** s ISP konektorom programátora ISP káblom štandardne dodávaným s programátorom. Skontrolujte správnosť prepojenia (napr. 1-1, 2-2, ..., 20-20).
- V Pg4uw spustíte selftest ISP konektora (Programátor / Selftest ISP konektora)



Test kalibrácie

- Vložte **48 Pins Calibration test POD, Type I** do ZIF päťice modulu. **48 Pins Calibration test POD, Type I** musí byť vložený ako 48 pinový obvod.
- V Pg4uw spustíte test kalibrácie (Programátor / Test kalibrácie).



Technická špecifikácia

HARDWARE

Programátor, prevodníky

- USB 2.0 high speed kompatibilný port, prenosová rýchlosť do 480 Mbit/s
- pomocou FPGA realizovaný IEEE 1284 slave printer port, prenosová rýchlosť do 1MB/s

- vlastná inteligencia tvorená výkonným mikroprocesorom s podporou FPGA stavového automatu
- tri D/A prevodníky pre VCCP, VPP1 a VPP2, s riadenou strmostou nábežnej a zostupnej hrany
- rozsah VCCP: 0..8V/1A
- rozsah VPP1, VPP2: 0..26V/1A
- autokalibrácia
- schopnosť selftestu
- ochrana proti elektrostatickým výbojom a prepätiam na vstupe napájania aj na komunikačnom konektore
- zdierka pre pripojenie antistatického náramku
- zdierka pre pripojenie k uzemňovacej časti

Programovacia ZIF päťica, pindrivery

- 48-pinová DIL ZIF (Zero Insertion Force) päťica; pre obvody do 48 pinov so šírkou púzdra 300/600 mil
- pindrivery: 48 univerzálny
- VCCP/VPP1/VPP2 pripojiteľné na všetky piny
- kvalitné pripojenie GND pre každý pin
- TTL driver realizovaný pomocou FPGA poskytuje H, L, CLK, pull-up, pull-down na všetkých pinoch pindrivery
- napäťové úrovne analógovej časti pindrivery nastaviteľné od 1.8 V do 26V
- prúdové obmedzenie, vypnutie pri prekročení prúdu a výpadku napájacieho napätia
- ESD prepäťová ochrana všetkých vývodov päťice (IEC1000-4-2: 15kV vzduchom, 8kV kontaktom)
- test vodivosti programátora: všetky piny sú testované pred každou programovacou procedúrou

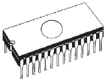
ISP konektor

- 20-pinová vidlica s kľúčom
- 6 TTL pindrivery, poskytujúce H, L, CLK, pull-up, pull-down; úroveň H nastaviteľná v rozmedzí 1.8V až 5V k práci so všetkými (vrátane nízkonapäťovými) zariadeniami
- 1x VCCP napätie (rozsah 2V..7V/100mA), pripojiteľné na dva piny
- programovacie napätie (VCCP) so schopnosťou korigovať napájacie napätie cieľového obvodu smerom hore aj dole a detekciu napätia
- 1x VPP napätie (rozsah 2V..25V/50mA), pripojiteľné na 6 pinov
- napájanie cieľovej aplikácie (rozsah 2V..6V/250mA)
- ESD ochrana všetkých vývodov ISP konektora (IEC1000-4-2: 15kV vzduchom, 8kV kontaktom)
- 2 výstupné signály, ktoré indikujú výsledok operácie = LED OK a LED Error (aktívna úroveň min 1,8V)
- Vstupný signál, ktorý je ekvivalent tlačidla YES! (aktívna úroveň max. 0,8V)

PROGRAMOVANÉ OBVODY

Programátor, v ZIF päťici

- EPROM: NMOS/CMOS, 27xxx a 27Cxxx, s 8/16 bitovou dátovou zbernicou, plná podpora LV obvodov
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx, s 8/16 bitovou dátovou zbernicou



- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx, od 256Kbit do 1Gbit, s 8/16 bitovou dátovou zbernicou, plná podpora LV obvodov
- Sériové E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 45Dxxx, 59Cxxx, 25Fxxx, 25Pxxx, 85xxx, 93Cxxx, NVM3060, MDAXxx, plná podpora LV obvodov
- Konfigurovateľné (EE)PROM: XCFxxx, XC17xxxx, XC18Vxxx, EPCxxx, AT17xxx, 37LVxx
- 1-Wire E(E)PROM: DS1xxx, DS2xxx
- PROM: AMD, Harris, National, Philips/Signetics, Tesla, TI
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x
- PLD: Altera: MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX7000AE, MAX II
- PLD: Lattice: ispGAL22V10x, ispLSI1xxx, ispLSI1xxxEA, ispLSI2xxx, ispLSI2xxxA, ispLSI2xxxE, ispLSI2xxxV, ispLSI2xxxVE, ispLSI2xxxVL, LC4xxxB/C/V/ZC, M4-xx/xx, M4A3-xx/xx, M4A5-xx/xx, M4LV-xx/xx
- PLD: Xilinx: XC9500, XC9500XL, XC9500XV, CoolRunner XPLA3, CoolRunner-II
- ostatné PLD: SPLD/CPLD series: AMI, Atmel, AMD-Vantis, Gould, Cypress, ICT, Lattice, NS, Philips, STM, VLSI, TI
- Mikroprocesory rady 48: 87x41, 87x42, 87x48, 87x49, 87x50
- Mikroprocesory rady 51: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, všetci výrobcovia, rada Philips LPC
- Mikroprocesory Intel rady 196: 87C196 KB/KC/KD/KT/KR/...
- Mikroprocesory Atmel AVR: AT90Sxxxx, ATtiny, ATmega
- Mikroprocesory Cypress: CY7Cxxxxx, CY8Cxxxxx
- Mikroprocesory ELAN: EM78Pxxx
- Mikroprocesory MDT 1xxx and 2xxx
- Mikroprocesory Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17Cxxx, PIC18xxx, PIC24xxx, dsPIC
- Mikroprocesory Motorola (Freescale): 68HC05, 68HC08, 68HC11, HCS08, HCS12
- Mikroprocesory Myson MTV2xx, 3xx, 4xx a 5xx
- Mikroprocesory National: COP8xxx
- Mikroprocesory NEC: uPD78Fxxx
- Mikroprocesory Novatek: NT68xxx
- Mikroprocesory Scenix (Ubicom): SXxxx
- Mikroprocesory SGS-Thomson: ST6xx, ST7xx, ST10xx, STR7xx
- Mikroprocesory TI: MSP430 a MSC121x
- Mikroprocesory ZILOG: Z86/Z89xxx and Z8xxx
- Mikroprocesory ostatné: EM Microelectronic, Fujitsu, Goal Semiconductor, Hitachi, Holtek, Princeton, Macronix, Winbond, Infineon(Siemens), Samsung, Toshiba, ...

I.C. Tester

- TTL typy: 54,74 S/LS/ALS/H/HC/HCT
- CMOS typy: 4000, 4500
- Statické RAM: 6116 .. 62A000
- Generovanie užívateľom definovaných testovacích postupností

Programátor, cez ISP konektor

- Sériové E(E)PROM: rada IIC, rada MW, rada SPI, rada KEELOQ, sériová data Flash, PLD konfigurovateľné pamäte
- Mikroprocesory Atmel: AT89Sxxx, AT90Sxxxx, ATtiny, ATmega
- Mikroprocesory Cypress: CY8C2xxxx
- Mikroprocesory Elan: EM78Pxxx, EM6xxx
- Mikroprocesory EM Microelectronic: 4 a 8 bitové rady

- Mikroprocesory Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17xxx, PIC18xxx, PIC24xxx, dsPIC
- Mikroprocesory Motorola/Freescale: HC11, HC908 (5-wire a All-wire), HCS08, HCS12
- Mikroprocesory NEC: uPD7xxx
- Mikroprocesory Philips: LPC2xxx, LPC, 89xxx
- Mikroprocesory Scenix (Ubicom): SXxxx
- Mikroprocesory TI: MSP430 (JTAG a rada BSL), MSC12xxx
- PLD: Lattice: ispGAL22xV10x, ispLSI1xxxEA, ispLSI2xxxE, ispLSI2xxxV, ispLSI2xxxVE, ispLSI2xxxVL, M4-xx/xx, M4LV-xx/xx, M4A3-xx/xx, M4A5-xx/xx, LC4xxxB/C/V/ZC
- rôzne PLD (tiež s podporou JAM player/JTAG):
- Altera: MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 9000, MAX II
- Xilinx: XC9500, XC9500XL, XC9500XV, CoolRunner XPLA3, CoolRunner-II

Poznámka:

Aktuálny zoznam všetkých podporovaných obvodov je na www.elnec.sk

Podporované púzdra

- podpora pre všetky obvody v DIP púzdrení bez prídavnej redukcie
- podpora púzdrenia DIP, SDIP, PLCC, JLCC, SOIC, SOP, PSOP, SSOP, TSOP, TSOPII, TSSOP, QFP, PQFP, TQFP, VQFP, QFN (MLF), SON, BGA, EBGA, FBGA, VFBGA, UBGA, FTBGA, LAP, CSP, SCSIP atď.
- ostatné púzdrenia do 48 pinov podporované pomocou univerzálnych konvertorov
- programátor je kompatibilný aj s konvertormi iných výrobcov

Rýchlosť programovania

BeeProg2

| Obvod | Veľkosť [bits] | Operácia | čas |
|-----------------------------------|----------------------|-----------------------------|-----------|
| Am29DL640G (paralelná NOR Flash) | 400080Hx16 (64 Mega) | programovanie a verifikácia | 24 sek |
| K8P6415UQB (paralelná NOR Flash) | 400100Hx16 (64 Mega) | programovanie a verifikácia | 13 sek |
| K9F1G08U0M (paralelná NAND Flash) | 8400000Hx8 (1 Giga) | programovanie a verifikácia | 122.7 sek |
| QB25F640S33 (sériová Flash) | 800200Hx8 (64 Mega) | programovanie a verifikácia | 30.7 sek |
| AT89C51RD2 (mikroprocesor) | 10000Hx8 | programovanie a verifikácia | 14.4 sek |
| PIC32MX360F512L (mikroprocesor) | 80000Hx8 | programovanie a verifikácia | 16.2 sek |

Konfigurácia:: P4, 2.4GHz, 512MB RAM, USB2.0, Windows XP

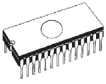
BeeProg+

| Obvod | Veľkosť [bits] | Operácia | čas |
|-----------------------------------|----------------------|-----------------------------|---------|
| M50FW080 (paralelná Flash) | 100000Hx8 (8 Mega) | programovanie a verifikácia | 22 sek |
| MX28F640C3BT (paralelná Flash) | 400000Hx16 (64 Mega) | programovanie a verifikácia | 57 sek |
| K9F1G08U0M (paralelná NAND Flash) | 8400000Hx8 (1 Giga) | programovanie a verifikácia | 239 sek |
| AT45D081 (sériová Flash) | 108000Hx8 (16 Mega) | programovanie a verifikácia | 36 sek |
| AT89C51RD2 (mikroprocesor) | 10000Hx8 | programovanie a verifikácia | 15 sek |
| PIC18LF452 (mikroprocesor) | 4000Hx16 | programovanie a verifikácia | 4 sek |

Konfigurácia: P4, 2.4GHz, 512MB RAM, USB2.0, Windows XP

Poznámky:

- Rýchlosti programovania našich výrobkov testujeme na vzorkách náhodných dát, na rozdiel od niektorých konkurenčných výrobcov používajúcich vzorky s "riedkymi" dátami obsahujúcimi len časť kvalitatívnych dát na programovanie a dáta obsahujúce niekoľko nulových bitov (FE, EF, atď.). Takouto úpravou vzoriek je možné do značnej miery znížiť programovacie časy jednotlivých obvodov. Pri porovnávaní produktov z tohto hľadiska si vždy vyžiadajte vzorky dát na ktorých boli tieto produkty testované.



- *Rýchlosť programovania čiastočne závisí od rýchlosti PC a od vyťaženia operačného systému.*

SOFTWARE

- **Algoritmy:** použité sú výhradne výrobcom obvodov schválené alebo odporúčané algoritmy.
- **Nové verzie SW:** nové verzie programového vybavenia sú k dispozícii približne každé 4 týždne, bezplatne.
- **Základné vlastnosti:** história revízií, protokol práce, on-line nápoveda, informácie o obvodoch a algoritmoch.

Činnosti s obvodom

- **štandardné:**
 - inteligentný výber obvodu podľa typu, výrobcu alebo časti mena obvodu
 - automatický výber EPROM/Flash EPROM obvodov podľa ID
 - kontrola vymazania, čítanie, verifikácia
 - programovanie
 - výmaz
 - programovanie konfiguračných a zabezpečovacích bitov
 - illegal bit test
 - kontrolná suma
 - interpretuje Jam Standard Test and Programming Language (STAPL), JEDEC standard JESD-71
 - interpretuje VME/SVF súbory
- **zabezpečovacie**
 - kontrola správneho vloženia, spätná kontrola vloženia
 - kontrola kontaktu ZIF - obvod
 - kontrola identifikačných (ID) byte obvodu
- **špeciálne**
 - výrobný mód (automatický štart programovania po vložení obvodu)
 - niekoľko serializačných módov (viac typov inkrementálnych módov, from-file mód, custom generator mód)
 - štatistika
 - count-down mód

Činnosti s bufferom

- view/edit, find/replace
- fill, copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Súbor načítať/uložiť

- žiadny čas na download, pretože programátor je riadený PC
- automatické rozpoznávanie formátu súborov

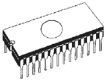
Akceptované formáty súborov

- neformátovaný (raw) binárny

- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-SPACE-HEX
- ALTERA POF, JEDEC (ver. 3.0.A), napr. od ABEL, CUPL, PALASM, TANGO PLD, OrCAD PLD, PLD Designer ISDATA, atď.
- JAM (JEDEC STAPL Format), JBC (Jam STAPL Byte Code), STAPL (STAPL File) JEDEC standard JESD-71
- VME (ispVME súbor VME2.0/VME3.0)

VŠEOBECNÉ

- prevádzkové napájanie: 110-250V AC
- spotreba elektrickej energie max. 20W aktívny režim
- rozmery 195x140x55 mm
- hmotnosť (programátor) 0,9kg
- pracovný teplotný obsah 5°C ÷ 40°C
- pracovná relatívna vlhkosť 20%..80%, nekondenzujúca



SmartProg2



Úvod

SmartProg2 je ďalším členom novej generácie univerzálnych programátorov ELNEC, pracujúcich pod Windows. Programátor bol vyvinutý aby spĺňal požiadavky vývojových pracovísk a servisných centier na univerzálny a ľahko prenosný programátor.

SmartProg2 je malý, výkonný a extrémne rýchly programátor všetkých programovateľných obvodov, určený pre stacionárne aj mobilné použitie. Programátor pracuje bez použitia prídavných modulov so všetkými obvodmi v DIL puzdre do 40 pinov. Programátor je vybavený aj konektorom pre in-circuit sériové programovanie (ISP), čo ďalej zvyšuje jeho úžitkovú hodnotu. SmartProg2 je nielen programátor, ale aj tester RAM pamätí.

Programátor predstavuje vynikajúco technicky spracovaný výrobok, zabezpečujúci spoľahlivé a rýchle programovanie a to za veľmi konkurencieschopnú cenu. Vo svojej triede poskytuje iste jeden z najlepších pomerov cena/výkon.

Programátor programuje obvody maximálne rýchlo z dôvodu použitia výkonného hardware na báze FPGA a podpory USB 2.0 full speed portu.

SmartProg2 je periférne zariadenie osobného počítača triedy IBM PC Pentium kompatibilný a vyšších typov, od PC desktop po palmtop cez USB port, čo je dôležité pre nové typy počítačov bez vstavaného LPT portu (napr. notebook).

SmartProg2 má 40 výkonných TTL pindriverov, ktoré poskytujú H/L/pull-up/pull-down a možnosť čítania pre každý pin päťice. Použitie pokročilé kvalitné a vysokorýchlostné obvody poskytujú kvalitné signály bez prekmitov na programovanie všetkých podporovaných obvodov. Pindriver poskytuje napájacie napätie dostatočne nízke na programovanie všetkých dnešných (1.8V) aj budúcich low-voltage obvodov.

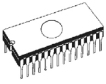
Programovaný obvod je chránený pred poškodením chybou obsluhy výkonnými ochrannými opatreniami: kontrola **správneho zasunutia obvodu** do päťice, kontrola **správneho kontaktu vývodov** a samozrejme kontrola **elektronickej identifikácie obvodu**.

Hardware programátora poskytuje tiež dostatočné prostriedky pre **test programátora**, takže ovládací program má kedykoľvek možnosť pretestovať správnosť fungovania pindriverov, prítomnosť a správnosť nastavenia všetkých napätí, preveriť správne časovanie, ako aj preveriť komunikačný kanál PC - programátor.

Verifikácia naprogramovaných hodnôt je vykonávaná pri **marginálnych hodnotách** napájacieho napätia, čo zvyšuje kvalitu programovacieho procesu a dlhodobú stálosť naprogramovaných údajov.

K dispozícii je WIN32 ovládací program s prehľadným ovládaním a výkonnými funkciami, podporený bohatou nápovedou. Voľba programovaného obvodu je podľa výrobcov, podľa tried obvodov alebo zadaním fragmentu označenia, uvedeného na obvode.

Štandardné príkazy pre manipuláciu s obvodmi (čítanie, kontrola vymazania, programovanie, verifikácia, výmaz obvodu) sú posilnené o **testovacie funkcie** (test správneho vloženia obvodu do päťice, kontrola elektronickej identifikácie obvodu) a **špeciálne funkcie** (autoinkrementácia).



Ovládací program podporuje všetky známe formáty súborov. Čítacia funkcia automaticky rozpoznáva formát načítavaného súboru a urobí potrebnú konverziu dát.

Funkcia **autoinkrementácie** zabezpečí modifikáciu sériového čísla v bufferi vždy pred vložením nového obvodu. Autoinkrementačná funkcia umožňuje čítať sériové čísla, resp. individuálnu identifikáciu programovaného obvodu aj zo súboru.

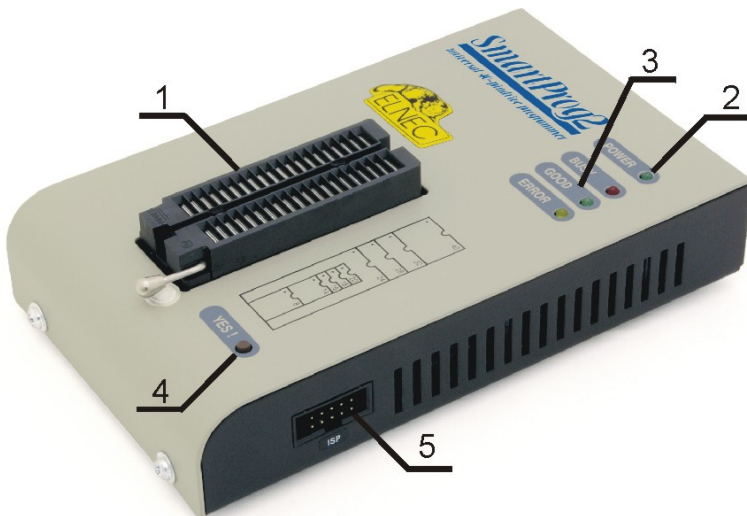
Software poskytuje veľa informácií o programovaných obvodoch, napr. obrázky možných puzdrení obvodu, vysvetlenie popisov na obvode (prefixy a sufixy) a mnoho ďalších informácií.

Na programovanie obvodov v iných ako DIL púzdrach je k dispozícii široký sortiment konvertorov z DIL na PLCC, SOIC, SSOP, TSOP, TSSOP, TQFP, QFN (MLF) a iné puzdra.

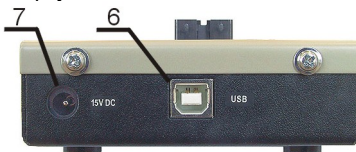
Koncepcia programátora SmartProg2, použité súčiastky, ako aj spoľahlivý výrobný proces a zahorovanie dovoľujú poskytnúť na programátor 3 ročnú záruku, s obmedzením 25 000 cyklov na päťicu ZIF.

Popis programátora SmartProg2

- 1) 40 pinová ZIF (Zero Insertion Force) päťica
- 2) LED indikujúca napájanie
- 3) LED indikujúce prácu a výsledok práce s obvodom
- 4) tlačítko YES!
- 5) ISP konektor



- 6) konektor na pripojenie USB prepojovacieho kábla PC <-> SmartProg2
- 7) konektor na pripojenie napájacieho kábla



napájací konektor



Poznámka: Pretože spotreba programátora v neaktívnom stave je veľmi malá, programátor nemá vypínač napájania. Neaktívny stav je indikovaný tlmenou intenzitou svietenia LED POWER

Pripojenie SmartProg2 k PC

Pre SmartProg2 je poradie pripájania USB kábla a napájacieho zdroja nerelevantné.

Problémy prepojenia SmartProg2 ↔ PC a ich odstránenie

Ak máte problémy s prepojením SmartProg2 ↔ PC, prosím pozrite časť **Spoločné poznámky**.

Manipulácia s programovanými obvodmi

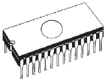
Po výbere želaného obvodu, vložte ho do ZIF päťice (páčka je hore) a sklopte páčku na vytvorenie kontaktu obvod - ZIF päťica.

Správna orientácia programovaného obvodu v ZIF päťici je znázornená na obrázku vedľa ZIF päťice. Programovaný obvod je nutné vkladať a vyberať zo ZIF päťice iba vtedy, keď nesvieti LED BUSY.

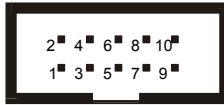
Varovanie: Programátor SmartProg2 nemá ochranné obvody, ktoré chránia obsah programovaného obvodu pri kritických situáciách (napr. výpadky napájania, výpadky komunikácie s PC). Obvod je zvyčajne zničený, ak počas programovania dôjde k násilnému prerušeniu behu ovládacieho programu (RESET, vypnutie počítača...) alebo vybratia obvodu zo ZIF päťice. Nesprávne vloženie obvodu do ZIF päťice môže byť príčinou jeho poškodenia alebo zničenia.

In-system serial programming pomocou programátora SmartProg2

Definície, odporúčania a pokyny sú popísané v časti **Spoločné poznámky / ISP**.



Opis ISP konektora programátora SmartProg2

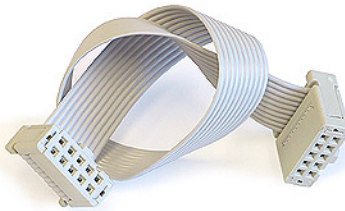


Predný pohľad na ISP konektor programátora.

Konkrétny význam pinov ISP konektora je závislý od obvodu, ktorý chcete programovať. Nájdete ho v ovládacom programe Pg4uw, menu **Obvod / Informácie o obvode (Ctrl+F1)**. Skontrolujte, či ste vybrali ISP spôsob programovania želaného obvodu. Je to označené príponou (ISP) za názvom vybratého obvodu.

Význam pinov odpovedá odporúčaniam z aplikačných poznámok, ktoré vydávajú výrobcovia súčiastok. Použité aplikačné poznámky môžete nájsť na www.elnec.sk v časti aplikačné poznámky.

Poznámka: Pin 1 je na konektore ISP káblíka označený trojuholníkovým zárezom.

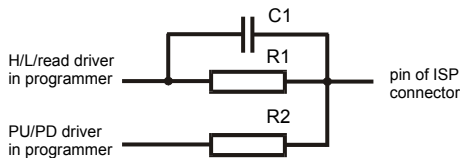


ISP káblík programátora SmartProg2

Upozornenie:

- **Keď používate SmartProg2 ako ISP programátor, nekladajte obvod do ZIF päťice.**
- **Keď programujete obvody v ZIF päťici, nezasúvajte ISP káblík do ISP konektora.**
- **Používajte iba priložený ISP káblík.** Použitie iného ISP káblíka (iný materiál, dĺžka...) môže byť príčinou nespoľahlivého programovania.
- **SmartProg2 môže napájať iba programovaný obvod, ale cieľové zariadenie nemôže napájať SmartProg2.**
- **SmartProg2 privedie programovacie napätie na cieľový obvod a overí si jeho hodnotu (cieľový systém môže ovplyvňovať programovacie napätie).** Ak je programovacie napätie odlišné ako požadované, práca s obvodom bude zrušená.

Poznámka: Driver programátora SmartProg2 pre H/L/ čítanie

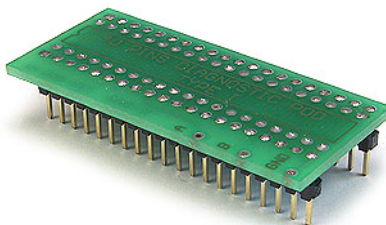


C1=1nF, R1=1k3, R2=22k

Selftest

Ak máte pocit, že sa Váš programátor nespráva úplne podľa Vašich očakávaní, vykonajte selftest programátora pomocou diagnostickej hlavice dodávanej štandardne s programátorom.

- Vložte **40 pins diagnostic POD - type I** do ZIF päťice programátora. **40 pins diagnostic POD - type I** musí byť vložený ako 40 pinový obvod.
- Spustíte selftest programátora v Pg4uw (Programátor / Selftest plus).



Technická špecifikácia

HARDWARE

Programátor

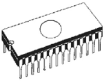
- dva D/A prevodníky pre VCCP a VPP s riadenou strmosťou nábežnej a zostupnej hrany
- rozsah VCCP: 2V až 7V/350mA
- rozsah VPP: 2V až 25V/200mA
- USB 2.0/1.1 kompatibilný interface
- schopnosť selftestu

Programovacia päťica a pindrivery

- 40 pinový DIL ZIF sokel pre obvody do 40 pinov so šírkou púzdra 300/600 mil
- pindrivery: 40 TTL pindrivery, špecializovaný (GND/VCC/VPP) pindriver
- pomocou FPGA realizovaný TTL driver podporuje H, L, CLK, pull-up, pull-down na všetkých pinoch pindrivery, úroveň H nastaviteľná od 1.8 V po 5V
- test vodivosti (ZIF programátora - programovaný obvod) pre každý pin pindrivery

ISP konektor

- 10-pinový konektor s ochranou proti prepólovaniu
- 6 TTL pindrivery podporujúcich H, L, CLK, pull-up, pull-down; úroveň H voliteľná od 1.8V do 5V pre prácu so všetkými (vrátane nízkonapäťovými) zariadeniami
- 1x VCCP napätie (rozsah 2V..7V/100mA) a 1x VPP napätie (rozsah 2V..25V/50mA)



- programovacie napätie (VCCP) so schopnosťou korigovať napájacie napätie cieľového obvodu smerom hore aj dole a detekciou napätia

Note: Programátor nie je určený k napájaniu cieľovej aplikácie pomocou VCCP pinu. Pri požiadavke napájať cieľovú aplikáciu z programátora, použite prosím programátor BeeProg2 / BeeProg+.

PROGRAMOVANÉ OBVODY

Programátor

- EPROM: NMOS/CMOS, 27xxx a 27Cxxx s 8/16 bitovou dátovou zbernicou, vrátane LV verzií (*1*2)
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx s 8/16 bitovou dátovou zbernicou vrátane LV verzií (*1*2)
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx, s 8/16 bitovou dátovou zbernicou vrátane LV verzií (*1*2)
- Sériové E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 45Dxxx, 59Cxxx, 25Fxxx, 25Pxxx, 85xxx, 93Cxxx, plná podpora LV obvodov (*1)
- Konfiguračné (EE)PROM: XCFxxx, 37LVxx, XC17xxxx, EPCxxx, AT17xxx, plná podpora LV obvodov
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x series
- PLD: firiem Atmel, AMD-Vantis, Cypress, ICT, Lattice, NS, ... (*1)
- Mikroprocesory rady MCS51: 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, LPC series od Atmel, Atmel W&M, Intel, Philips, SST, Winbond (*1*2)
- Mikroprocesory Atmel AVR: ATtiny, AT90Sxxx, ATmega (*1*2)
- Mikroprocesory Cypress: CY8Cxxxx
- Mikroprocesory ELAN: EM78Pxxx
- Mikroprocesory EM Microelectronic: 4 a 8 bitové
- Mikroprocesory Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17Cxxx, PIC18xxx, dsPIC 8-40 pinové (*1*2)
- Mikroprocesory Scenix (Ubicom): SXxxx series

Programátor, cez ISP konektor

- Sériové E(E)PROM: IIC rodina
- Mikroprocesory rady Atmel: AT89Sxxx, AT90Sxxxx, ATtiny, ATmega
- Mikroprocesory Cypress: CY8C2xxxx
- Mikroprocesory Elan: EM78Pxxx
- Mikroprocesory EM Microelectronic: 4 a 8 bitové
- Mikroprocesory Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17xxx, PIC18xxx, dsPIC
- Mikroprocesory Philips: rodina LPC

Poznámky:

- (*1) - pre non DIL puzdra sú k dispozícii odpovedajúce konvertory puzdier.
- (*2) - pre viac ako 40-pinové obvody je k dispozícii iba niekoľko adaptérov, preto prosím uvažujte o výkonnejšom programátore (BeeProg2 / BeeProg+), ak potrebujete programovať obvody s viac ako 40 pinmi
- aktuálny zoznam všetkých podporovaných obvodov je na www.elnec.sk

I.C. Tester

- Statické RAM: 6116 .. 624000

Rýchlosť programovania

| Obvod | Činnosť | Mód | Čas |
|------------|-----------------------------|-------|--------|
| 27C010 | programovanie a verifikácia | v ZIF | 28 sek |
| AT29C040A | programovanie a verifikácia | v ZIF | 32 sek |
| AM29F040 | programovanie a verifikácia | v ZIF | 62 sek |
| PIC16C67 | programovanie a verifikácia | v ZIF | 10 sek |
| PIC18F452 | programovanie a verifikácia | v ZIF | 7 sek |
| AT89C52 | programovanie a verifikácia | v ZIF | 16 sek |
| PIC16F876A | programovanie a verifikácia | ISP | 5 sek |
| PIC12C508 | programovanie a verifikácia | ISP | 3 sek |

Konfigurácia *P4, 2,4GHz, USB 2.0, Windows XP, Pg4uw 2.11*

SOFTWARE

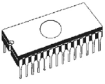
- **Algoritmy:** použité sú výhradne výrobcom obvodov schválené alebo odporúčané algoritmy.
- **Nové verzie SW:** nové verzie programového vybavenia sú k dispozícii približne každé 4 týždne, bezplatne.
- **Základné vlastnosti:** história revízií, protokol práce, on-line nápoveda, informácie o obvodoch a algoritmoch.

Činnosti s obvodom

- **štandardné:**
 - inteligentný výber obvodu podľa typu, výrobcu alebo fragmentu mena obvodu
 - kontrola vymazania, čítanie, verifikácia
 - programovanie
 - porovnanie a zápis rozdielov na disk
 - výmaz
 - programovanie konfiguračných a zabezpečovacích bitov
 - illegal bit test
 - kontrolná suma
- **zabezpečovacie:**
 - kontrola správneho vloženia obvodu do päťice
 - kontrola kontaktu ZIF - obvod
 - kontrola identifikačných (ID) byte obvodu
- **špeciálne:**
 - automatická inkrementácia sériového čísla
 - štatistika
 - count-down mód

Činnosti s bufferom

- view/edit, find/replace
- fill, copy, move, byte swap, word/dword split
- checksum (byte, word)
- print



Súbor načítať/uložiť

- žiadny čas na download, pretože programátor je riadený PC
- automatické rozpoznávanie formátu súborov

Akceptované formáty súborov

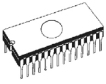
- neformátovaný (raw) binárny
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-SPACE-HEX
- JEDEC (ver. 3.0.A), napr. od ABEL, CUPL, PALASM, TANGO PLD, OrCAD PLD, PLD Designer ISDATA, atď.

VŠEOBECNÉ

- napájanie 15..20V DC, max. 500mA
- spotreba max. 6W aktívny, približne 1,4W neaktívny
- rozmery: 160x95x35 [mm]
- hmotnosť: cca 500g (bez napájacieho zdroja)
- pracovný teplotný rozsah 5°..40°C
- pracovná relatívna vlhkosť 20%..80%, nekondenzujúca

MEMprog2





Úvod

MEMprog2 je ďalším členom novej generácie špecializovaných programátorov ELNEC, pracujúcich pod Windows. Programátor bol vyvinutý tak, aby spĺňal požiadavky vývojových pracovísk a servisných centier na špecializovaný programátor pamätí.

MEMprog2 môže byť upgradovaný na **SmartProg2** použitím **Xprog2 to SmartProg2 upgrade kit** (obj. č.60-0048).

MEMprog2 podporuje prakticky všetky typy pamätí - EPROM, EEPROM, NVRAM, Flash EPROM a sériové EEPROM - vrátane LV verzií. MEMprog2 je nielen programátor, ale aj tester statických pamätí.

Programátor predstavuje vynikajúco technicky spracovaný výrobok, zabezpečujúci spoľahlivé a rýchle programovanie a to za veľmi konkurencieschopnú cenu. Vo svojej triede poskytuje iste jeden z najlepších pomerov cena/výkon.

Programátor programuje obvody maximálne rýchlo z dôvodu použitia výkonného hardware na báze FPGA a podpory USB 2.0 full speed portu.

MEMprog2 je periférne zariadenie osobného počítača triedy IBM PC Pentium kompatibilný a vyšších typov, od PC desktop po palmtop cez USB port, čo je dôležité pre nové typy počítačov bez vstavaného LPT portu (napr. notebook).

MEMprog2 má 40 výkonných TTL pindriverov, ktoré poskytujú H/L/pull-up/pull-down a možnosť čítania pre každý pin päťice. Použitie pokročilé kvalitné a vysokorýchlostné obvody poskytujú kvalitné signály bez prekmitov na programovanie všetkých podporovaných obvodov. Pindriver poskytuje napájacie napätie dostatočne nízke na programovanie všetkých dnešných (1.8V) aj budúcich low-voltage obvodov.

Programovaný obvod je chránený pred poškodením chybou obsluhy výkonnými ochrannými opatreniami: kontrola **správneho zasunutia obvodu** do päťice, kontrola **správneho kontaktu vývodov** a samozrejme kontrola **elektronickej identifikácie obvodu**.

Hardware programátora poskytuje tiež dostatočné prostriedky pre **test programátora**, takže ovládací program má kedykoľvek možnosť pretestovať správnosť fungovania pindriverov, prítomnosť a správnosť nastavenia všetkých napätí, preveriť správne časovanie, ako aj preveriť komunikačný kanál PC - programátor.

Verifikácia naprogramovaných hodnôt je vykonávaná pri **marginálnych hodnotách** napájacieho napätia, čo zvyšuje kvalitu programovacieho procesu a dlhodobú stálosť naprogramovaných údajov.

K dispozícii je WIN32 ovládací program s prehľadným ovládaním a výkonnými funkciami, podporený bohatou nápovedou. Voľba programovaného obvodu je podľa výrobcov, podľa tried obvodov alebo zadaním fragmentu označenia, uvedeného na obvode.

Štandardné príkazy pre manipuláciu s obvody (čítanie, kontrola vymazania, programovanie, verifikácia, výmaz obvodu) sú posilnené o **testovacie funkcie** (test správneho vloženia

obvodu do päťice, kontrola elektronickej identifikácie obvodu) a **špeciálne funkcie** (autoinkrementácia).

Ovládací program podporuje všetky známe formáty súborov. Čítacia funkcia automaticky rozpoznáva formát načítavaného súboru a urobí potrebnú konverziu dát.

Funkcia **autoinkrementácie** zabezpečí modifikáciu sériového čísla v bufferi vždy pred vloženíím nového obvodu. Autoinkrementačná funkcia umožňuje čítať sériové čísla, resp. individuálnu identifikáciu programovaného obvodu aj zo súboru.

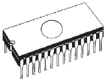
Software poskytuje veľa informácií o programovaných obvodoch, napr. obrázky možných puzdrení obvodu, vysvetlenie popisov na obvode (prefixy a sufíxy) a mnoho ďalších informácií.

Na programovanie obvodov v iných ako DIL púzdrach je k dispozícii široký sortiment konvertorov z DIL na PLCC, SOIC, SSOP, TSOP, TSSOP a iné puzdra.

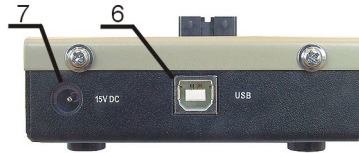
Popis programátora MEMprog2

- 1) 40 pinová ZIF (Zero Insertion Force) päťica
- 2) LED indikujúca napájanie
- 3) LED indikujúca prácu a výsledok práce s obvodom
- 4) tlačítko YES!
- 5) ISP konektor





- 6) konektor na pripojenie USB prepojovacieho kábla PC <-> MEMprog2
- 7) konektor na pripojenie napájacieho kábla



napájací konektor



Poznámka: Pretože spotreba programátora v neaktívnom stave je veľmi malá, programátor nemá vypínač napájania. Neaktívny stav je indikovaný tlmenu intenzitou svietenia LED POWER

Pripojenie MEMprog2 k PC

Pre MEMProg2 je poradie pripájania USB kábla a napájacieho zdroja nerelevantné.

Problémy prepojenia MEMprog2 ↔ PC a ich odstránenie

Ak máte problémy s prepojením MEMprog2 ↔ PC, prosím pozrite časť **Spoločné poznámky**.

Manipulácia s programovanými obvodmi

Po výbere želaného obvodu, vložte ho do ZIF päťice (páčka je hore) a sklopte páčku na vytvorenie kontaktu obvodu ZIF päťica.

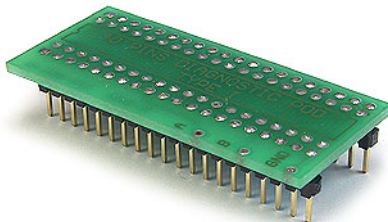
Správna orientácia programovaného obvodu v ZIF päťici je znázornená na obrázku vedľa ZIF päťice. Programovaný obvod je nutné vkladať a vyberať zo ZIF päťice iba vtedy, keď nesvieti LED BUSY.

Varovanie: Programátor MEMprog2 nemá ochranné obvody, ktoré chránia obsah programovaného obvodu pri kritických situáciách (napr. výpadky napájania, výpadky komunikácie s PC). Obvod je zvyčajne zničený, ak počas programovania dôjde k násilnému prerušeniu behu ovládacieho programu (RESET, vypnutie počítača...) alebo vybratia obvodu zo ZIF päťice. Nesprávne vloženie obvodu do ZIF päťice môže byť príčinou jeho poškodenia alebo zničenia.

Selftest

Ak máte pocit, že sa Váš programátor nespráva úplne podľa Vašich očakávaní, vykonajte selftest programátora pomocou diagnostической hlavice dodávanej štandardne s programátorom.

- Vložte **40 pins diagnostic POD - type I** do ZIF päťice programátora. **40 pins diagnostic POD - type I** musí byť vložený ako 40 pinový obvod.
- Spustíte selftest programátora v Pg4uw (Programátor / Selftest plus).



Technická špecifikácia

HARDWARE

Programátor

- dva D/A prevodníky pre VCCP a VPP s riadenou strmou nábežnej a zostupnej hrany
- rozsah VCCP: 2V až 7V/350mA
- rozsah VPP: 2V až 25V/200mA
- USB 2.0/1.1 kompatibilný interface
- schopnosť selftestu

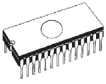
Programovacia päťica a pindrivery

- 40 pinový DIL ZIF sokel pre obvody do 40 pinov so šírkou púzdra 300/600 mil
- pindrivery: 40 TTL pindriverov, špecializovaný (GND/VCC/VPP) pindriver
- pomocou FPGA realizovaný TTL driver podporuje H, L, CLK, pull-up, pull-down na všetkých pinoch pindriveru, úroveň H nastaviteľná od 1.8 V po 5V
- test vodivosti (ZIF programátora - programovaný obvod) pre každý pin pindriveru

ISP konektor (použiteľný až po upgrade na SmartProg2)

- 10-pinový konektor s ochranou proti prepólovaniu
- 6 TTL pindriverov podporujúcich H, L, CLK, pull-up, pull-down; úroveň H voliteľná od 1.8V do 5V pre prácu so všetkými (vrátane nízkonapäťovými) zariadeniami
- 1x VCCP napätie (rozsah 2V..7V/100mA) a 1x VPP napätie (rozsah 2V..25V/50mA)
- programovacie napätie (VCCP) so schopnosťou korigovať napájacie napätie cieľového obvodu smerom hore aj dole a detekciou napätia

Note: Programátor nie je určený k napájaniu cieľovej aplikácie pomocou VCCP pinu. Pri požiadavke napájať cieľovú aplikáciu z programátora, použite prosím programátor BeeProg2 / BeeProg+.



PROGRAMOVANÉ OBVODY

Programátor

- EPROM: NMOS/CMOS, 27xxx a 27Cxxx s 8/16 bitovou dátovou zbernicou, vrátane LV verzii (*1*2)
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx s 8/16 bitovou dátovou zbernicou vrátane LV verzii (*1*2)
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx, s 8/16 bitovou dátovou zbernicou vrátane LV verzii (*1*2)
- Sériové E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 45Dxxx, 59Cxxx, 25Fxxx, 25Pxxx, 85xxx, 93Cxxx, plná podpora LV obvodov (*1)
- Konfiguračné (EE)PROM: XCFxxx, 37LVxx, XC17xxxx, EPCxxx, AT17xxx, plná podpora LV obvodov
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x series

Poznámky:

- (*1) - pre non DIL puzdra sú k dispozícii odpovedajúce konvertory puzdier.
- (*2) - pre viac ako 40-pinové obvody je k dispozícii iba niekoľko adaptérov, preto prosím uvažujte o výkonnejšom programátore (BeeProg2 / BeeProg+), ak potrebujete programovať obvody s viac ako 40 pinmi
- aktuálny zoznam všetkých podporovaných obvodov je na www.elnec.sk

I.C. Tester

- Statické RAM: 6116 .. 624000

Rýchlosť programovania

| Obvod | Činnosť | Čas |
|-----------|-----------------------------|--------|
| 27C010 | programovanie a verifikácia | 28 sek |
| AT29C040A | programovanie a verifikácia | 32 sek |
| AM29F040 | programovanie a verifikácia | 62 sek |

Konfigurácia

P4, 2,4GHz, USB 2.0, Windows XP, Pg4uw 2.11

SOFTWARE

- **Algoritmy:** použité sú výhradne výrobcom obvodov schválené alebo odporúčané algoritmy.
- **Nové verzie SW:** nové verzie programového vybavenia sú k dispozícii približne každé 4 týždne, bezplatne.
- **Základné vlastnosti:** história revízií, protokol práce, on-line nápoveda, informácie o obvodoch a algoritmoch.

Činnosti s obvodom

- **štandardné:**
 - inteligentný výber obvodu podľa typu, výrobcu alebo fragmentu mena obvodu
 - kontrola vymazania, čítanie, verifikácia
 - programovanie
 - porovnanie a zápis rozdielov na disk
 - výmaz
 - programovanie konfiguračných a zabezpečovacích bitov
 - illegal bit test

- kontrolná suma
- **zabezpečovacie:**
 - kontrola správneho vloženia obvodu do päťice
 - kontrola kontaktu ZIF obvod
 - kontrola identifikačných (ID) byte obvodu
- **špeciálne:**
 - automatická inkrementácia sériového čísla
 - štatistika
 - count-down mód

Činnosti s bufferom

- view/edit, find/replace
- fill, copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Súbor načítať/uložiť

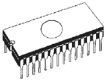
- žiadny čas na download, pretože programátor je riadený PC
- automatické rozpoznávanie formátu súborov

Akceptované formáty súborov

- neformátovaný (raw) binárny
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-SPACE-HEX

VŠEOBECNÉ

- napájanie 15..20V DC, max. 500mA
- spotreba max. 6W aktívny, približne 1,4W neaktívny
- rozmery: 160x95x35 [mm]
- hmotnosť: cca 500g (bez napájacieho zdroja)
- pracovný teplotný rozsah 5°..40°C
- pracovná relatívna vlhkosť 20%..80%, nekondenzujúca



T51prog2



Úvod

T51prog2 je ďalším členom novej generácie špecializovaných programátorov ELNEC, pracujúcich pod Windows. Programátor je schopný podporovať všetky súčasné mikroprocesory rodiny MCS51 (40 pinové) a AVR (8-40 pinov) používajúce sériový a paralelný prístup. **T51prog2** je zameraný na podporu súčasných a budúcich mikroprocesorov rodiny Atmel W&M MCS51 v rámci blízkej spolupráce s firmou Atmel W&M.

T51prog2 môže byť upgradovaný na **SmartProg2** použitím **Xprog2 to SmartProg2 upgrade kit** (obj. č.60-0048).

T51prog2 je malý, veľmi rýchly, výkonný a prenosný programátor mikroprocesorov rodiny MCS51 Atmel AVR. T51prog zároveň podporuje sériové EEPROM s typmi rozhrania IIC (24Cxx), Microwire (93Cxx) a SPI (25Cxx). Vstavaný ISP konektor umožňuje in-circuit sériové programovanie mikroprocesory rodiny MCS51 a Atmel AVR využívajúce sériový prístup.

Programátor predstavuje vynikajúco technicky spracovaný výrobok, zabezpečujúci spoľahlivé a rýchle programovanie a to za veľmi konkurencieschopnú cenu. Vo svojej triede poskytuje iste jeden z najlepších pomerov cena/výkon.

Programátor programuje obvody maximálne rýchlo z dôvodu použitia výkonného hardware na báze FPGA a podpory USB 2.0 full speed portu.

T51prog2 je periférne zariadenie osobného počítača triedy IBM PC Pentium kompatibilný a vyšších typov, od PC desktop po palmtop cez USB port, čo je dôležité pre nové typy počítačov bez vstavaného LPT portu (napr. notebook).

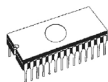
T51prog2 má 40 výkonných TTL pindriverov, ktoré poskytujú H/L/pull-up/pull-down a možnosť čítania pre každý pin päťice. Použitie pokročilé kvalitné a vysokorýchlostné obvody poskytujú kvalitné signály bez prekmitov na programovanie všetkých podporovaných obvodov. Pindriver poskytuje napájacie napätie dostatočne nízke na programovanie všetkých dnešných (1.8V) aj budúcich low-voltage obvodov.

Programovaný obvod je chránený pred poškodením chybou obsluhy výkonnými ochrannými opatreniami: kontrola **správneho zasunutia obvodu** do päťice, kontrola **správneho kontaktu vývodov** a samozrejme kontrola **elektronickej identifikácie obvodu**.

Hardware programátora poskytuje tiež dostatočné prostriedky pre **test programátora**, takže ovládací program má kedykoľvek možnosť pretestovať správnosť fungovania pindriverov, prítomnosť a správnosť nastavenia všetkých napätí, preveriť správne časovanie, ako aj preveriť komunikačný kanál PC - programátor.

Verifikácia naprogramovaných hodnôt je vykonávaná pri **marginálnych hodnotách** napájacieho napätia, čo zvyšuje kvalitu programovacieho procesu a dlhodobú stálosť naprogramovaných údajov.

K dispozícii je WIN32 ovládací program s prehľadným ovládaním a výkonnými funkciami, podporený bohatou nápovedou. Voľba programovaného obvodu je podľa výrobcov, podľa tried obvodov alebo zadaním fragmentu označenia, uvedeného na obvode.



Štandardné príkazy pre manipuláciu s obvody (čítanie, kontrola vymazania, programovanie, verifikácia, výmaz obvodu) sú posilnené o **testovacie funkcie** (test správneho vloženia obvodu do päťice, kontrola elektronickej identifikácie obvodu) a **špeciálne funkcie** (autoinkrementácia).

Ovládací program podporuje všetky známe formáty súborov. Čítacia funkcia automaticky rozpoznáva formát načítavaného súboru a urobí potrebnú konverziu dát.

Funkcia **autoinkrementácie** zabezpečí modifikáciu sériového čísla v bufferi vždy pred vloženíím nového obvodu. Autoinkrementačná funkcia umožňuje čítať sériové čísla, resp. individuálnu identifikáciu programovaného obvodu aj zo súboru.

Software poskytuje veľa informácií o programovaných obvodoch, napr. obrázky možných puzdrení obvodu, vysvetlenie popisov na obvode (prefixy a sufixy) a mnoho ďalších informácií.

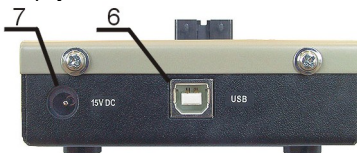
Na programovanie obvodov v iných ako DIL púzdrach je k dispozícii široký sortiment konvertorov z DIL na PLCC, SOIC, SSOP, TSOP, TSSOP, TQFP, QFN (MLF) a iné puzdra.

Popis programátora T51prog2

- 1) 40 pinová ZIF (Zero Insertion Force) päťica
- 2) LED indikujúca napájanie
- 3) LED indikujúce prácu a výsledok práce s obvodom
- 4) tlačítko YES!
- 5) ISP konektor



- 6) konektor na pripojenie USB prepojovacieho kábla PC <-> T51prog2
- 7) konektor na pripojenie napájacieho kábla



napájací konektor



Poznámka: Pretože spotreba programátora v neaktívnom stave je veľmi malá, programátor nemá vypínač napájania. Neaktívny stav je indikovaný tlmenou intenzitou svietenia LED POWER

Pripojenie T51prog2 k PC

Pre T51prog2 je poradie pripájania USB kábla a napájacieho zdroja nerelevantné.

Problémy prepojenia T51prog2 ↔ PC a ich odstránenie

Ak máte problémy s prepojením T51prog2 ↔ PC, prosím pozrite časť **Spoločné poznámky**.

Manipulácia s programovanými obvodmi

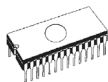
Po výbere želaného obvodu, vložte ho do ZIF päťice (páčka je hore) a sklopte páčku na vytvorenie kontaktu obvod ZIF päťica.

Správna orientácia programovaného obvodu v ZIF päťici je znázornená na obrázku vedľa ZIF päťice. Programovaný obvod je nutné vkladať a vyberať zo ZIF päťice iba vtedy, keď nesvieti LED BUSY.

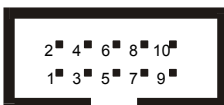
Varovanie: Programátor T51prog2 nemá ochranné obvody, ktoré chránia obsah programovaného obvodu pri kritických situáciách (napr. výpadky napájania, výpadky komunikácie s PC). Obvod je zvyčajne zničený, ak počas programovania dôjde k násilnému prerušeniu behu ovládacieho programu (RESET, vypnutie počítača...) alebo vybratia obvodu zo ZIF päťice. Nesprávne vloženie obvodu do ZIF päťice môže byť príčinou jeho poškodenia alebo zničenia.

In-system serial programming pomocou programátora T51prog2

Definície, odporúčania a pokyny sú popísané v časti **Spoločné poznámky / ISP**.



Opis ISP konektora programátora T51prog2

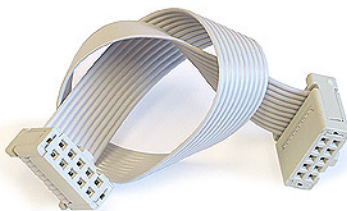


Predný pohľad na ISP konektor programátora.

Konkrétny význam pinov ISP konektora je závislý od obvodu, ktorý chcete programovať. Nájdete ho v ovládacom programe Pg4uw, menu **Obvod / Informácie o obvode (Ctrl+F1)**. Skontrolujte, či ste vybrali ISP spôsob programovania želaného obvodu. Je to označené príponou (ISP) za názvom vybratého obvodu.

Význam pinov odpovedá odporúčaniam z aplikačných poznámok, ktoré vydávajú výrobcovia súčiastok. Použité aplikačné poznámky môžete nájsť na www.elnec.sk v časti aplikačné poznámky.

Poznámka: Pin 1 je na konektore ISP káblíka označený trojuholníkovým zárezom.

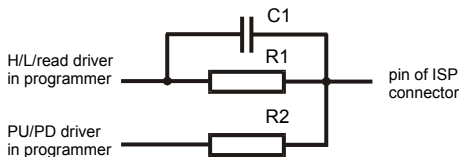


ISP káblík programátora T51prog2

Upozornenie:

- **Keď používate T51prog2 ako ISP programátor, nekladajte obvod do ZIF päťice.**
- **Keď programujete obvody v ZIF päťici, nezasúvajte ISP káblík do ISP konektora.**
- Používajte iba **priložený ISP káblík**. Použitie iného ISP káblíka (iný materiál, dĺžka...) môže byť príčinou nespoľahlivého programovania.
- **T51prog2 môže napájať iba programovaný obvod, ale cieľové zariadenie nemôže napájať T51prog2.**
- **T51prog2 privedie programovacie napätie na cieľový obvod a overí si jeho hodnotu (cieľový systém môže ovplyvňovať programovacie napätie). Ak je programovacie napätie odlišné ako požadované, práca s obvodom bude zrušená.**

Poznámka: Driver programátora T51prog2 pre H/L/ čítanie

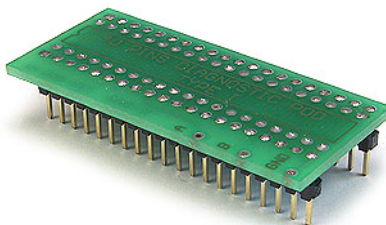


C1=1nF, R1=1k3, R2=22k

Selftest

Ak máte pocit, že sa Váš programátor nespráva úplne podľa Vašich očakávaní, vykonajte selftest programátora pomocou diagnostickej hlavice dodávanej štandardne s programátorom.

- Vložte **40 pins diagnostic POD - type I** do ZIF päťice programátora. **40 pins diagnostic POD - type I** musí byť vložený ako 40 pinový obvod.
- Spustíte selftest programátora v Pg4uw (Programátor / Selftest plus).



Technická špecifikácia

HARDWARE

Programátor

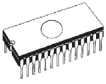
- dva D/A prevodníky pre VCCP a VPP s riadenou strmosťou nábežnej a zostupnej hrany
- rozsah VCCP: 2V až 7V/350mA
- rozsah VPP: 2V až 25V/200mA
- USB 2.0/1.1 kompatibilný interface
- schopnosť selftestu

Programovacia päťica a pindrivery

- 40 pinový DIL ZIF sokel pre obvody do 40 pinov so šírkou púzdra 300/600 mil
- pindrivery: 40 TTL, špecializovaný (GND/VCC/VPP) pindriver pre MCS51 a AVR
- pomocou FPGA realizovaný TTL driver podporuje H, L, CLK, pull-up, pull-down na všetkých pinoch pindriveru, úroveň H nastaviteľná od 1.8 V po 5V
- test vodivosti (ZIF programátora - programovaný obvod) pre každý pin pindriveru

ISP konektor

- 10-pinový konektor s ochranou proti prepólovaniu
- 6 TTL pindriverov podporujúcich H, L, CLK, pull-up, pull-down; úroveň H voliteľná od 1.8V do 5V pre prácu so všetkými (vrátane nízkonapäťovými) zariadeniami
- 1x VCCP napätie (rozsah 2V..7V/100mA) a 1x VPP napätie (rozsah 2V..25V/50mA)



- programovacie napätie (VCCP) so schopnosťou korigovať napájacie napätie cieľového obvodu smerom hore aj dole a detekciou napätia

Note: Programátor nie je určený k napájaniu cieľovej aplikácie pomocou VCCP pinu. Pri požiadavke napájať cieľovú aplikáciu z programátora, použite prosím programátor BeeProg2 / BeeProg+.

PROGRAMOVANÉ OBVODY

Programátor

- mikroprocesory rodiny MCS51: 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, LPC od Atmel, Atmel W&M, Intel, Philips, SST, Winbond, ... 8-40 pinov (*1)
- mikroprocesory Atmel AVR: rodina ATtiny, AT90Sxxx, ATmega (paralelný a sériový prístup), 8-40 pinov (*1)
- Sériové E(E)PROM: rodiny 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx

Programátor, cez ISP konektor

- Mikroprocesory rady Atmel: AT89Sxxx, AT90Sxxxx, ATtiny, ATmega
- Mikroprocesory Philips: rodina LPC

Poznámky:

- (*1) - odpovedajúce konvertory sú použiteľné pre púzdra iné ako DIL a obvody viac ako 40 pinov.
- aktuálny zoznam všetkých podporovaných obvodov je na www.elnec.sk

Rýchlosť programovania

| Obvod | Činnosť | Čas |
|----------|-----------------------------|--------|
| AT89C52 | programovanie a verifikácia | 15 sek |
| T87C5111 | programovanie a verifikácia | 14 sek |

Konfigurácia P4, 2,4GHz, USB 2.0, Windows XP, Pg4uw 2.11

SOFTWARE

- **Algoritmy:** použité sú výhradne výrobcom obvodov schválené alebo odporúčané algoritmy.
- **Nové verzie SW:** nové verzie programového vybavenia sú k dispozícii približne každé 4 týždne, bezplatne.
- **Základné vlastnosti:** história revízií, protokol práce, on-line nápoveda, informácie o obvodoch a algoritmoch.

Činnosti s obvodom

- **štandardné:**
 - inteligentný výber obvodu podľa typu, výrobcu alebo fragmentu mena obvodu
 - kontrola vymazania, čítanie, verifikácia
 - programovanie
 - porovnanie a zápis rozdielov na disk
 - výmaz
 - programovanie konfiguračných a zabezpečovacích bitov
 - illegal bit test
 - kontrolná suma
- **zabezpečovacie:**
 - kontrola správneho vloženia obvodu do päťice
 - kontrola kontaktu ZIF obvod

- kontrola identifikačných (ID) byte obvodu
- **špeciálne:**
 - automatická inkrementácia sériového čísla
 - štatistika
 - count-down mód

Činnosti s bufferom

- view/edit, find/replace
- fill, copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Súbor načítať/uložiť

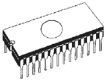
- žiadny čas na download, pretože programátor je riadený PC
- automatické rozpoznávanie formátu súborov

Akceptované formáty súborov

- neformátovaný (raw) binárny
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-SPACE-HEX

VŠEOBECNÉ

- napájanie 15..20V DC, max. 500mA
- spotreba max. 6W aktívny, približne 1,4W neaktívny
- rozmery: 160x95x35 [mm]
- hmotnosť: cca 500g (bez napájacieho zdroja)
- pracovný teplotný rozsah 5°..40°C
- pracovná relatívna vlhkosť 20%..80%, nekondenzujúca



PIKprog2



Úvod

PIKprog2 je ďalším členom novej generácie špecializovaných programátorov ELNEC, pracujúcich pod Windows. Programátor bol vyvinutý tak, aby spĺňal všetky požiadavky firmy Microchip™ pre profesionálne programátory.

PIKprog2 môže byť upgradovaný na **SmartProg2** použitím **Xprog2 to SmartProg2 upgrade kit** (obj. č.60-0048).

PIKprog2 je malý, výkonný a extrémne rýchly servisný programátor všetkých jednočipových mikroprocesorov rady Microchip™ PICmicro® (do 40-pinov DIL), určený pre stacionárne aj mobilné použitie. PIKprog+ programuje aj sériové EEPROM s rozhraním IIC, Microwire a SPI. Programátor je vybavený konektorom pre in-circuit sériové programovanie (ISP), čo ďalej zvyšuje jeho úžitkovú hodnotu. Programátor PIKprog+ je určený všetkým, ktorí pracujú s procesormi rodiny Microchip™ PIC a univerzálny programátor im nevyhovuje z hľadiska ceny, alebo z hľadiska mobilnosti a rozmeru.

Programátor predstavuje vynikajúco technicky spracovaný výrobok, zabezpečujúci spoľahlivé a rýchle programovanie a to za veľmi konkurencieschopnú cenu. Vo svojej triede poskytuje iste jeden z najlepších pomerov cena/výkon.

Programátor programuje obvody maximálne rýchlo z dôvodu použitia výkonného hardware na báze FPGA a podpory USB 2.0 full speed portu.

PIKprog2 je periférne zariadenie osobného počítača triedy IBM PC Pentium kompatibilný a vyšších typov, od PC desktop po palmtop cez USB port, čo je dôležité pre nové typy počítačov bez vstavaného LPT portu (napr. notebook).

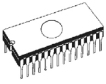
PIKprog2 má 40 výkonných TTL pindriverov, ktoré poskytujú H/L/pull-up/pull-down a možnosť čítania pre každý pin päťice. Použitie pokročilé kvalitné a vysokorýchlostné obvody poskytujú kvalitné signály bez prekmitov na programovanie všetkých podporovaných obvodov. Pindriver poskytuje napájacie napätie dostatočne nízke na programovanie všetkých dnešných (1.8V) aj budúcich low-voltage obvodov.

Programovaný obvod je chránený pred poškodením chybou obsluhy výkonnými ochrannými opatreniami: kontrola **správneho zasunutia obvodu** do päťice, kontrola **správneho kontaktu vývodov** a samozrejme kontrola **elektronickej identifikácie obvodu**.

Hardware programátora poskytuje tiež dostatočné prostriedky pre **test programátora**, takže ovládací program má kedykoľvek možnosť pretestovať správnosť fungovania pindriverov, prítomnosť a správnosť nastavenia všetkých napätí, preveriť správne časovanie, ako aj preveriť komunikačný kanál PC - programátor.

Verifikácia naprogramovaných hodnôt je vykonávaná pri **marginálnych hodnotách** napájacieho napätia, čo zvyšuje kvalitu programovacieho procesu a dlhodobú stálosť naprogramovaných údajov.

K dispozícii je WIN32 ovládací program s prehľadným ovládaním a výkonnými funkciami, podporený bohatou nápovedou. Voľba programovaného obvodu je podľa výrobcov, podľa tried obvodov alebo zadáním fragmentu označenia, uvedeného na obvode.



Štandardné príkazy pre manipuláciu s obvody (čítanie, kontrola vymazania, programovanie, verifikácia, výmaz obvodu) sú posilnené o **testovacie funkcie** (test správneho vloženia obvodu do päťice, kontrola elektronickej identifikácie obvodu) a **špeciálne funkcie** (autoinkrementácia).

Ovládací program podporuje všetky známe formáty súborov. Čítacia funkcia automaticky rozpoznáva formát načítavaného súboru a urobí potrebnú konverziu dát.

Funkcia **autoinkrementácie** zabezpečí modifikáciu sériového čísla v bufferi vždy pred vložím nového obvodu. Autoinkrementačná funkcia umožňuje čítať sériové čísla, resp. individuálnu identifikáciu programovaného obvodu aj zo súboru.

Software poskytuje veľa informácií o programovaných obvodoch, napr. obrázky možných puzdrení obvodu, vysvetlenie popisov na obvode (prefixy a sufixy) a mnoho ďalších informácií.

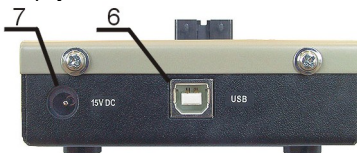
Na programovanie obvodov v iných ako DIL púzdrach je k dispozícii široký sortiment konvertorov z DIL na PLCC, SOIC, SSOP, TSOP, TSSOP, TQFP, QFN (MLF) a iné puzdra.

Popis programátora PIKprog2

- 1) 40 pinová ZIF (Zero Insertion Force) päťica
- 2) LED indikujúca napájanie
- 3) LED indikujúce prácu a výsledok práce s obvodom
- 4) tlačítko YES!
- 5) ISP konektor



- 6) konektor na pripojenie USB prepojovacieho kábla PC <-> PIKprog2
- 7) konektor na pripojenie napájacieho kábla



napájací konektor



Poznámka: Pretože spotreba programátora v neaktívnom stave je veľmi malá, programátor nemá vypínač napájania. Neaktívny stav je indikovaný tlmenou intenzitou svietenia LED POWER

Pripojenie PIKprog2 k PC

Pre PIKprog2 je poradie pripájania USB kábla a napájacieho zdroja nerelevantné.

Problémy prepojenia PIKprog2 ↔ PC a ich odstránenie

Ak máte problémy s prepojením PIKprog2 ↔ PC, prosím pozrite časť **Spoločné poznámky**.

Manipulácia s programovanými obvodmi

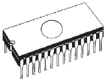
Po výbere želaného obvodu, vložte ho do ZIF päťice (páčka je hore) a sklopte páčku na vytvorenie kontaktu obvod - ZIF päťica.

Správna orientácia programovaného obvodu v ZIF päťici je znázornená na obrázku vedľa ZIF päťice. Programovaný obvod je nutné vkladať a vyberať zo ZIF päťice iba vtedy, keď nesvieti LED BUSY.

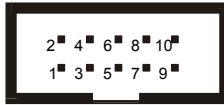
Varovanie: Programátor PIKprog2 nemá ochranné obvody, ktoré chránia obsah programovaného obvodu pri kritických situáciách (napr. výpadky napájania, výpadky komunikácie s PC). Obvod je zvyčajne zničený, ak počas programovania dôjde k násilnému prerušeniu behu ovládacieho programu (RESET, vypnutie počítača...) alebo vybratia obvodu zo ZIF päťice. Nesprávne vloženie obvodu do ZIF päťice môže byť príčinou jeho poškodenia alebo zničenia.

In-system serial programming pomocou programátora PIKprog2

Definície, odporúčania a pokyny sú popísané v časti **Spoločné poznámky / ISP**.



Opis ISP konektora programátora PIKprog2

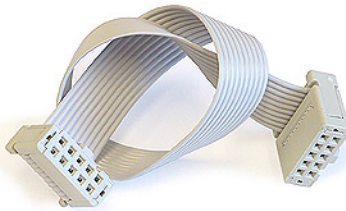


Predný pohľad na ISP konektor programátora.

Konkrétny význam pinov ISP konektora je závislý od obvodu, ktorý chcete programovať. Nájdete ho v ovládacom programe Pg4uw, menu **Obvod / Informácie o obvode (Ctrl+F1)**. Skontrolujte, či ste vybrali ISP spôsob programovania želaného obvodu. Je to označené príponou (ISP) za názvom vybratého obvodu.

Význam pinov odpovedá odporúčaniam z aplikačných poznámok, ktoré vydávajú výrobcovia súčiastok. Použité aplikačné poznámky môžete nájsť na www.elnec.sk v časti aplikačné poznámky.

Poznámka: Pin 1 je na konektore ISP káblíka označený trojuholníkovým zárezom.

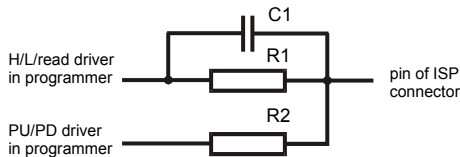


ISP káblík programátora PIKprog2

Upozornenie:

- **Keď používate PIKprog2 ako ISP programátor, nekladajte obvod do ZIF päťice.**
- **Keď programujete obvody v ZIF päťici, nezasúvajte ISP káblík do ISP konektora.**
- Používajte iba **priložený ISP káblík**. Použitie iného ISP káblíka (iný materiál, dĺžka...) môže byť príčinou nespoľahlivého programovania.
- **PIKprog2 môže napájať iba programovaný obvod, ale cieľové zariadenie nemôže napájať PIKprog2.**
- **PIKprog2 privedie programovacie napätie na cieľový obvod a overí si jeho hodnotu (cieľový systém môže ovplyvňovať programovacie napätie). Ak je programovacie napätie odlišné ako požadované, práca s obvodom bude zrušená.**

Poznámka: Driver programátora PIKprog2 pre H/L/ čítanie

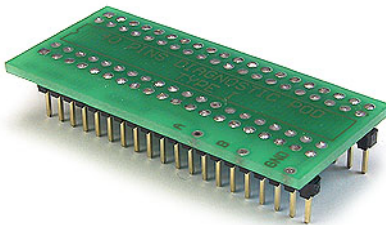


C1=1nF, R1=1k3, R2=22k

Selftest

Ak máte pocit, že sa Váš programátor nespráva úplne podľa Vašich očakávaní, vykonajte selftest programátora pomocou diagnostickej hlavice dodávanej štandardne s programátorom.

- Vložte **40 pins diagnostic POD - type I** do ZIF päťice programátora. **40 pins diagnostic POD - type I** musí byť vložený ako 40 pinový obvod.
- Spustíte selftest programátora v Pg4uw (Programátor / Selftest plus).



Technická špecifikácia

HARDWARE

Programátor

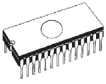
- dva D/A prevodníky pre VCCP a VPP s riadenou strmosťou nábežnej a zostupnej hrany
- rozsah VCCP: 2V až 7V/350mA
- rozsah VPP: 2V až 25V/200mA
- USB 2.0/1.1 kompatibilný interface
- schopnosť selftestu

Programovacia päťica a pindrivery

- 40 pinový DIL ZIF sokel pre obvody do 40 pinov so šírkou púzdra 300/600 mil
- pindrivery: 40 TTL pindrivery, špecializovaný (GND/VCC/VPP) pindriver
- pomocou FPGA realizovaný TTL driver podporuje H, L, CLK, pull-up, pull-down na všetkých pinoch pindrivery, úroveň H nastaviteľná od 1.8 V po 5V
- test vodivosti (ZIF programátora - programovaný obvod) pre každý pin pindrivery

ISP konektor

- 10-pinový konektor s ochranou proti prepólovaniu
- 6 TTL pindrivery podporujúcich H, L, CLK, pull-up, pull-down; úroveň H voliteľná od 1.8V do 5V pre prácu so všetkými (vrátane nízkonapäťovými) zariadeniami
- 1x VCCP napätie (rozsah 2V..7V/100mA) a 1x VPP napätie (rozsah 2V..25V/50mA)



- programovacie napätie (VCCP) so schopnosťou korigovať napájacie napätie cieľového obvodu smerom hore aj dole a detekciou napätia

Note: Programátor nie je určený k napájaniu cieľovej aplikácie pomocou VCCP pinu. Pri požiadavke napájať cieľovú aplikáciu z programátora, použite prosím programátor BeeProg2 / BeeProg+.

PROGRAMOVANÉ OBVODY

Programátor

- Mikroprocesory Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17Cxxx, PIC18xxx, dsPIC 8-40 pinové (*1)
- Sériové E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 45Dxxx, 59Cxxx, 25Fxxx, 25Pxxx, 85xxx, 93Cxxx, plná podpora LV obvodov (*1)

Programátor, cez ISP konektor

- Mikrokontroléry Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17xxx, PIC18xxx, dsPIC

Poznámky:

- (*1) - odpovedajúce konvertory sú použiteľné pre púzdra iné ako DIL a obvody viac ako 40 pinov.
- aktuálny zoznam všetkých podporovaných obvodov je na www.elnec.sk

Rýchlosť programovania

| Obvod | Činnosť | Mód | Čas |
|------------|-----------------------------|-------|--------|
| PIC16C67 | programovanie a verifikácia | v ZIF | 12 sek |
| PIC18F452 | programovanie a verifikácia | v ZIF | 8 sek |
| PIC16F876A | programovanie a verifikácia | ISP | 6 sek |
| PIC12C508 | programovanie a verifikácia | ISP | 4 sek |

Konfigurácia P4, 2,4GHz, USB 2.0, Windows XP, Pg4uw 2.11

SOFTWARE

- **Algoritmy:** použité sú výhradne výrobcom obvodov schválené alebo odporúčané algoritmy.
- **Nové verzie SW:** nové verzie programového vybavenia sú k dispozícii približne každé 4 týždne, bezplatne.
- **Základné vlastnosti:** história revízií, protokol práce, on-line nápoveda, informácie o obvodoch a algoritmoch.

Činnosti s obvodom

- **štandardné:**
 - inteligentný výber obvodu podľa typu, výrobcu alebo fragmentu mena obvodu
 - kontrola vymazania, čítanie, verifikácia
 - programovanie
 - porovnanie a zápis rozdielov na disk
 - výmaz
 - programovanie konfiguračných a zabezpečovacích bitov
 - illegal bit test
 - kontrolná suma
- **zabezpečovacie:**
 - kontrola správneho vloženia obvodu do päťce

- kontrola kontaktu ZIF - obvod
- kontrola identifikačných (ID) byte obvodu
- **špeciálne:**
 - automatická inkrementácia sériového čísla
 - štatistika
 - count-down mód

Činnosti s bufferom

- view/edit, find/replace
- fill, copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Súbor načítať/uložiť

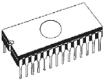
- žiadny čas na download, pretože programátor je riadený PC
- automatické rozpoznávanie formátu súborov

Akceptované formáty súborov

- neformátovaný (raw) binárny
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-SPACE-HEX

VŠEOBECNÉ

- napájanie 15..20V DC, max. 500mA
- spotreba max. 6W aktívny, približne 1,4W neaktívny
- rozmery: 160x95x35 [mm]
- hmotnosť: cca 500g (bez napájacieho zdroja)
- pracovný teplotný rozsah 5°..40°C
- pracovná relatívna vlhkosť 20%..80%, nekondenzujúca



SEEPROM



Úvod

SEEprog je univerzálny programátor sériových EEPROM v 8 pinovom DIL puzdre s rozhraním IIC, SPI, Microwire a tiež špeciálit ako napr. digitálne teplomery. Programátor tiež podporuje LV obvody (3,3 V).

SEEprog je periférne zariadenie osobného počítača triedy IBM PC Pentium kompatibilný a vyšších typov, od PC desktop po palmtop. K PC sa pripája cez štandardný paralelný port, určený na pripojenie tlačiarne. Pri pripájaní SEEprog k PC teda samozrejme nie je potrebné PC vypínať a rozoberať.

K dispozícii je WIN32 ovládací program s prehľadným ovládaním a výkonnými funkciami, podporený bohatou nápovedou. Voľba programovaného obvodu je podľa výrobcov, podľa tried obvodov alebo zadaním fragmentu označenia, uvedeného na obvode.

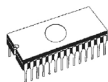
Štandardné príkazy pre manipuláciu s obvodmi (čítanie, kontrola vymazania, programovanie, verifikácia, výmaz obvodu) sú posilnené o **testovacie funkcie** a **špeciálne funkcie** (autoinkrementácia).

Ovládací program podporuje všetky známe formáty súborov. Čítacia funkcia automaticky rozpoznáva formát načítavaného súboru a urobí potrebnú konverziu dát.

Funkcia **autoinkrementácie** zabezpečí modifikáciu sériového čísla v bufferi vždy pred vložením nového obvodu. Autoinkrementačná funkcia umožňuje čítať sériové čísla, resp. individuálnu identifikáciu programovaného obvodu aj zo súboru.

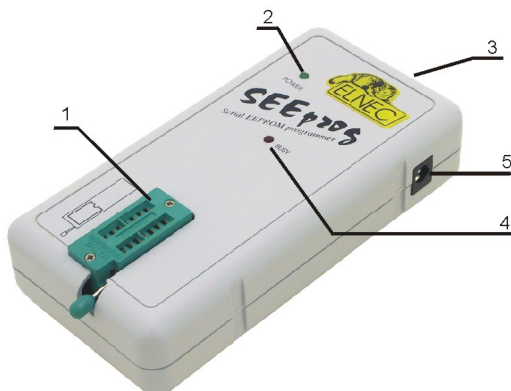
Software poskytuje veľa informácií o programovaných obvodoch, napr. obrázky možných puzdrení obvodu, vysvetlenie popisov na obvode (prefixy a sufixy) a mnoho ďalších informácií.

Pre programátor SEEprog sú k dispozícii konvertory päťc z DIL na SOIC.



Popis programátora SEEprog

- 1) 14 pinová ZIF (Zero Insertion Force) päťica
- 2) LED indikujúca napájanie
- 3) konektor na pripojenie prepojuvacieho kábla PC <-> SEEprog
- 4) LED indikujúca prácu a výsledok práce s obvodom
- 5) konektor na pripojenie napájacieho kábla



napájací konektor



Pripojenie SEEprog k PC

Vypnite PC, prepojuvaci kábel PC - programátor, ktorý je súčasťou dodávky programátora SEEprog zapojte do počítača na konektor, ktorý je určený pre pripojenie tlačiarne (printer port). V prípade, že váš počítač obsahuje iba jeden printer port, odpojte od počítača kábel k tlačiarne a na jeho miesto zapojte kábel programátora. Ak má váš počítač viac printer portov, pripojte programátor na nevyužitý printer port. Opačný koniec kábla pripojte ku programátoru. Obedva konektory je potrebné priskrutkovať do protikusov, zvlášť kritické je to pre konektor vedúci do programátora. Manipulácia zámeny kábla k tlačiarne za kábel k programátoru je iste nepríjemná, neodporúčame však prevádzkovať programátor SEEprog cez mechanický printer switch. Prevádzka cez elektronický prepínač nie je možná.

Napájací zdroj zasuňte do sieťovej zásuvky a konektor zdroja zapojte do programátora. Zapnite PC. Na programátore začne svietiť LED POWER a programátor SEEprog je pripravený komunikovať s ovládacím programom v PC. Spustíte ovládaci program pre SEEprog.

Upozornenie! Ak sa rozhodnete pri pripájaní alebo odpájaní programátora SEEprog nevypínať PC (komu by sa chcelo), dodržte prosím nasledovnú postupnosť:

- **Pripájanie:** *NAJPRV* kábel k PC a *POTOM* napájanie
- **Odpájanie:** *NAJPRV* napájanie a *POTOM* kábel k PC

Problémy prepojenia SEEprog ⇔ PC a ich odstránenie

Ak máte problémy s prepojením SEEprog ⇔ PC, prosím pozrite časť **Spoločné poznámky**.

Manipulácia s programovanými obvodmi

Po výbere želaného obvodu, vložte ho do ZIF päťice (páčka je hore) a sklopte páčku na vytvorenie kontaktu obvod - ZIF päťica.

Správna orientácia programovaného obvodu v ZIF päťici je znázornená na obrázku vedľa ZIF päťice. Programovaný obvod je nutné vkladať a vyberať zo ZIF päťice iba vtedy, keď nesvieti LED BUSY.

Varovanie: Programátor SEEprog nemá ochranné obvody, ktoré chránia obsah programovaného obvodu pri kritických situáciách (napr. výpadky napájania, výpadky komunikácie s PC). Okrem toho, obvod je zvyčajne zničený, ak počas programovania dôjde k násilnému prerušeniu behu ovládacieho programu (RESET, vypnutie počítača...) alebo vybratia obvodu zo ZIF päťice. Nesprávne vloženie obvodu do ZIF päťice môže byť príčinou jeho poškodenia alebo zničenja.

Technická špecifikácia

HARDWARE

Päťica, pin driver

- 14 pinová DIL ZIF päťica pre obvody so šírkou puzdra 300 mil (*1)
- každý pin päťice je možné nastaviť do stavov Low a Pull-up
- z každého pinu päťice je možné čítať
- podpora Low Voltage obvodov (napájanie 3.3V)

Poznámka:

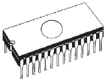
(*1) Pre prácu so sériovými EEPROM je použitých 8 pinov najbližšie k páčke päťice, nepoužitie piny sú nezapojené.

PROGRAMOVANÉ OBVODY

- EEPROM I2C (24Cxxx)
- EEPROM Microwire (93Cxxx)
- EEPROM SPI (25Cxxx)
- špeciálne obvody (digitálne teploměry, ...)

Poznámka:

- Programátor **NIE JE VHODNÝ** na ISP programovanie. Ak predpokladáte potrebu ISP v budúcnosti, uvažujte o programátoroch, ktoré majú ISP konektor (SmartProg2, BeeProg2 / BeeProg+...)
- aktuálny zoznam všetkých podporovaných obvodov je na www.elnec.sk



SOFTWARE

- **Algoritmy:** použité sú výhradne výrobcom obvodov schválené alebo odporúčané algoritmy.
- **Nové verzie SW:** nové verzie programového vybavenia sú k dispozícii približne každé 2 týždne, bezplatne.
- **Základné vlastnosti:** história revízií, protokol práce, on-line nápoveda, informácie o obvodoch a algoritmoch.

Činnosti s obvodom:

- **štandardné:**
 - kontrola vymazania,
 - čítanie,
 - programovanie,
 - verifikácia,
 - výmaz,
 - kontrolná suma

Činnosti s bufferom

- view/edit, find/replace
- fill/copy, move, byte swap, checksum
- checksum (byte, word)
- print

Akceptované formáty súborov

- neformátovaný (raw) binárny
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, ASCII-SPACE-HEX, Tektronix

Požiadavky na PC

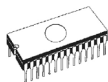
Pozrite časť **Úvod/ Požiadavky na riadiaci počítač**

VŠEOBECNÉ

- napájanie: 8..15V, max. 50mA
- rozmery: 137x65x40 [mm]
- hmotnosť: cca 150g
- pracovný teplotný rozsah: 5 až 40°C
- pracovná relatívna vlhkosť 20%..80%, nekondenzujúca



Inštalácia



Balenie programátora obsahuje CD s ovládacím programom, užitočnými nástrojmi a s doplnkovými informáciami. Za účelom ukázania ako pracujú programátory ELNEC je dovolené kopírovať obsah CD.

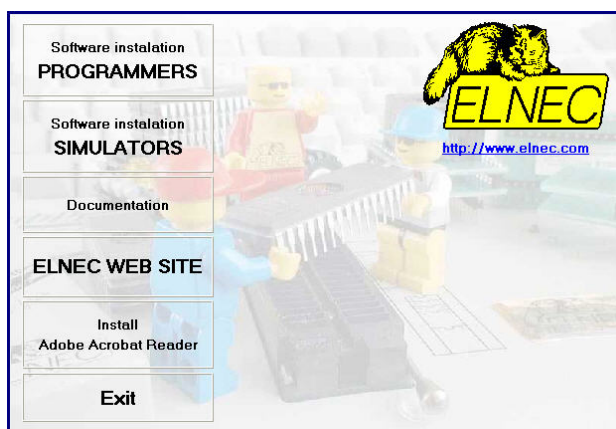
Pre programátory pripájané cez USB (LPT) port, ovládací program vyžaduje správne nainštalovaný USB (LPT) driver.

Aby sa zabránilo nechceným komplikáciám pri inštalácii, odporúčame inštalovať ovládací program pred pripojením programátora ku PC,.

Inštalácia programového vybavenia

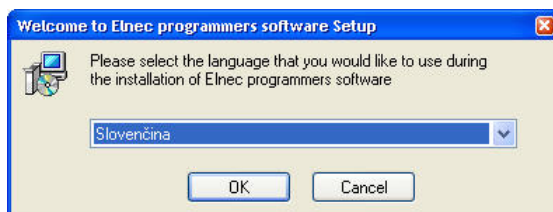
Vložte dodávané CD do vašej CD (DVD) mechaniky a inštalačný program sa spustí automaticky (ak nie, spustíte setup.exe). Inštalačný program Vás prevedie inštalačným procesom a urobí všetky potrebné kroky pred prvým spustením ovládacieho programu.

Krok 1.



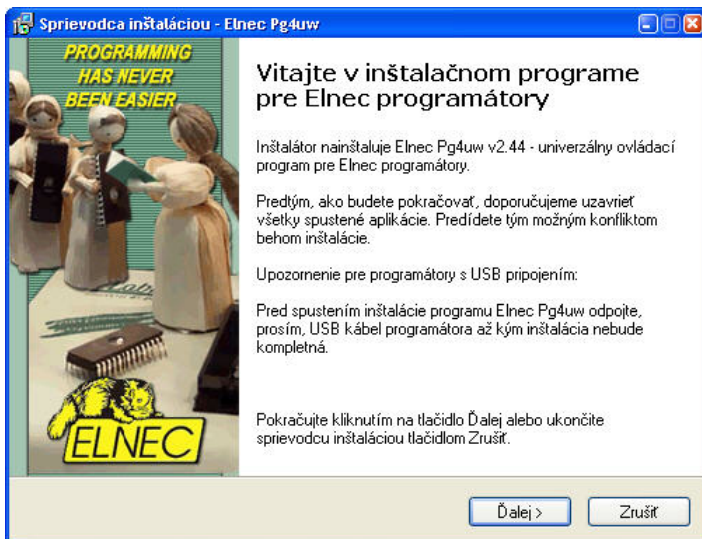
Kliknite na položku "Software installation PROGRAMMERS".

Krok 2.



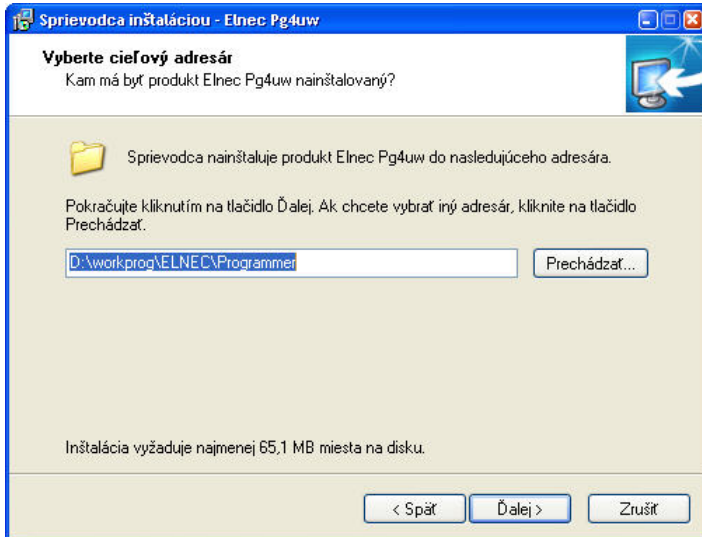
Vyberte si jazyk a potom kliknite na "OK".

Krok 3.

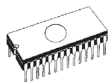
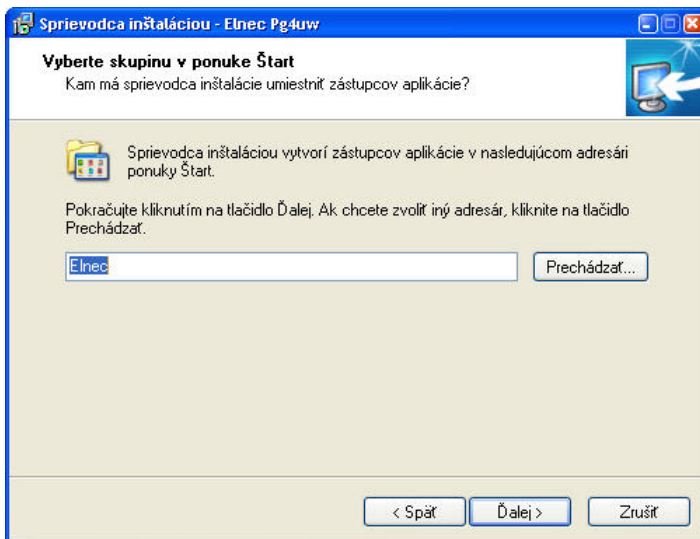


Kliknite na "Ďalej"

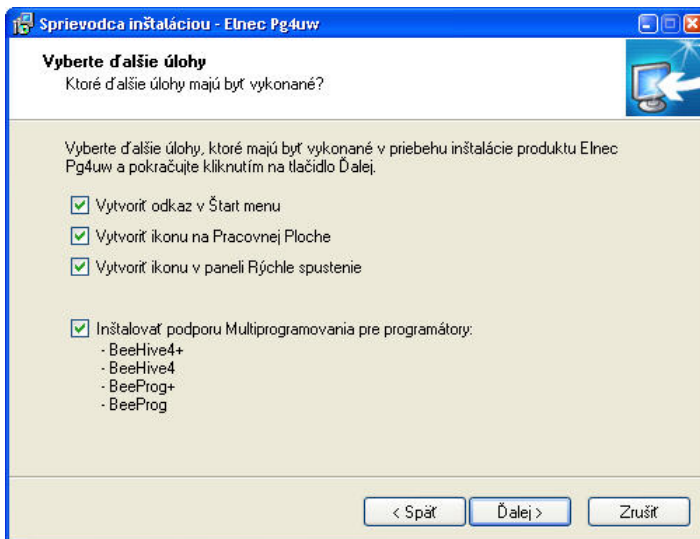
Krok 4.



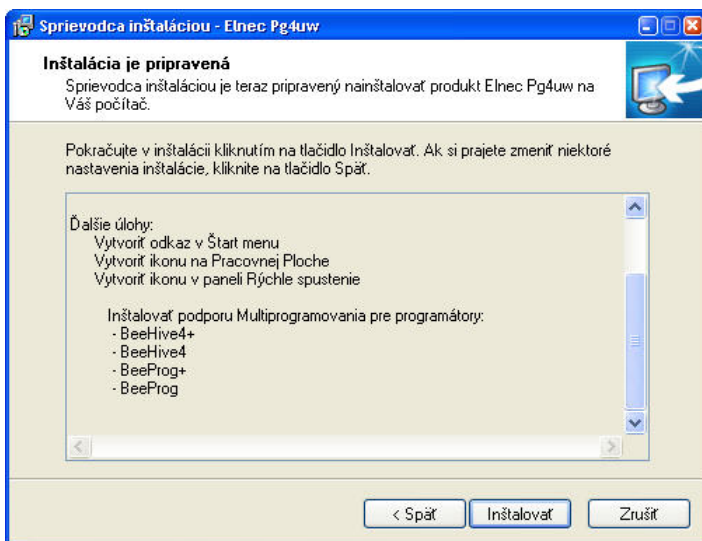
Ak chcete zmeniť prednastavený adresár, kliknite na prechádzať a vyberte si cieľový adresár. Potom kliknite na "Ďalej"

**Krok 5.**

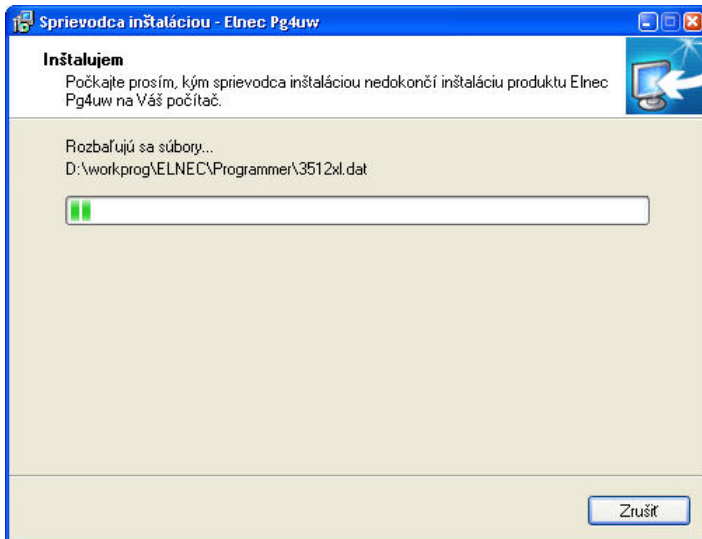
Ak chcete zmeniť prednastavený adresár, kliknite na prechádzať a vyberte si cieľový adresár. Potom kliknite na "Ďalej"

Krok 6.

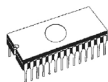
Skontrolujte či je zaškrtnuté "Inštalovať podporu Multiprogramovania pre programátory". Ak chcete, zmeňte prednastavené voľby. Potom kliknite na "Ďalej"

Krok 7.

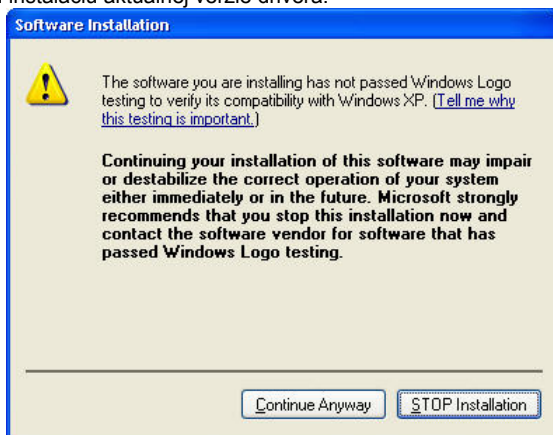
Skontrolujte svoje nastavenia a potom kliknite na "Inštalovať"

Krok 8.

Začne sa inštalácia.

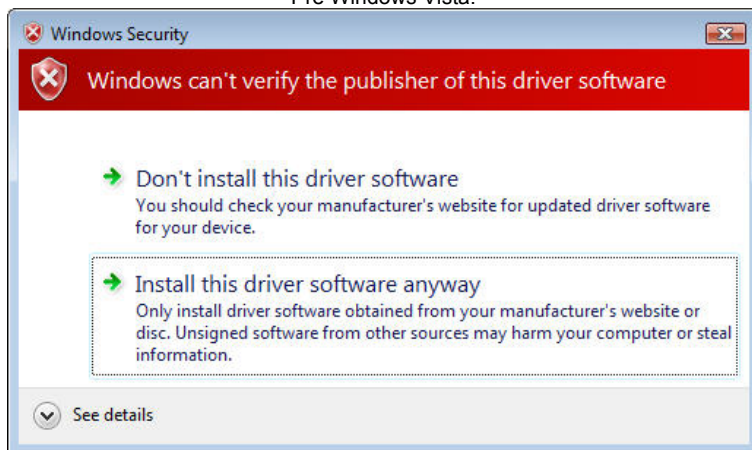
**Krok 9.**

Platí iba pre prvú inštaláciu aktuálnej verzie drivera.

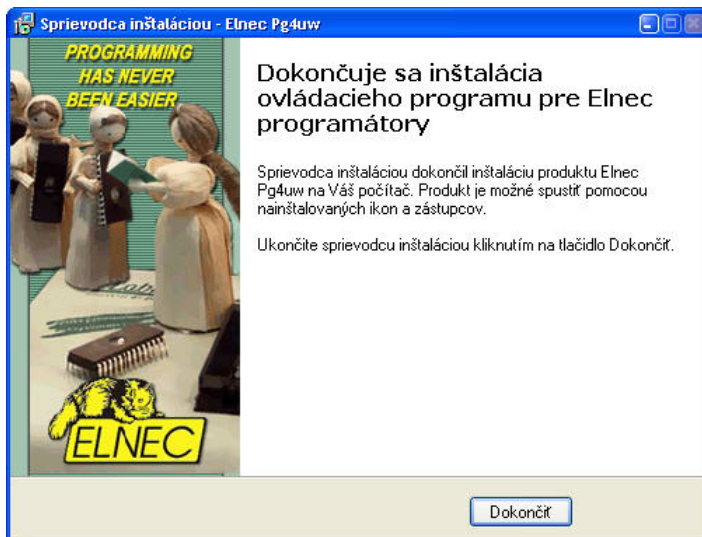


Kliknite na "Continue Anyway".

Pre Windows Vista:



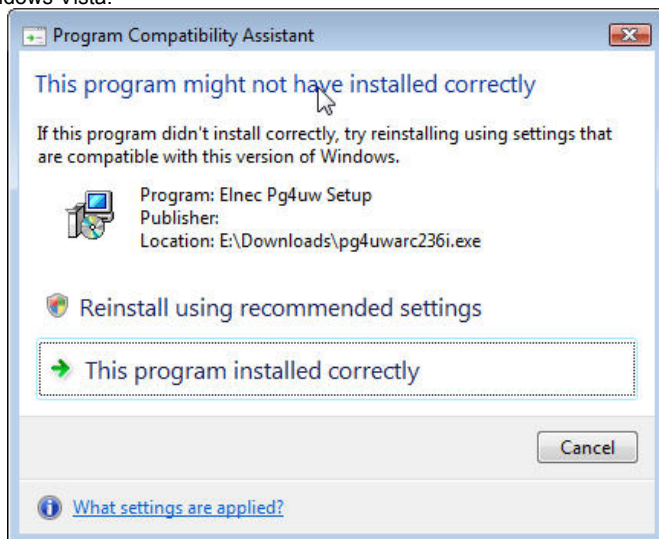
Kliknite na "Install this driver software anyway"

Krok 10.

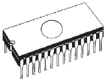
Na dokončenie inštalácie kliknite na "Dokončiť".

Krok 11.

Iba pre Windows Vista:



Kliknite na "This program installed correctly"



Nové verzie ovládacieho programu

Používanie najnovších verzií ovládacieho programu pre programátor je potrebné z hľadiska maximálneho využitia jeho vlastností. Najnovšiu verziu ovládacieho programu (Pg4uwARC.exe) môžete získať na internetovej stránke www.elnec.sk v časti **download**. Súbor je vhodné nakopírovať na disk do temporary katalógu, odpojiť programátor od PC a spustiť ho. Inštalácia začne v **Krok 2** opísanom vyššie.

Inštalácia hardware

Keď je programátor pripojený ku PC skôr ako bol nainštalovaný ovládací program, Windows detekuje nový hardware a dáva užívateľovi na výber medzi automatickou a ručnou inštaláciou drivera. Pre správnu detekciu programátora musí byť v CD mechanike počítača vložené inštaláčn CD a je potrebné vykonať nasledovné kroky:

Krok 1.

Zapojiť USB (LPT) kábel do USB (LPT) portu programátora.

Krok 2.

Zapojiť USB (LPT) kábel do USB (LPT) portu počítača (doporučujeme vysokorychlostné porty).

Krok 3.

Zapojiť konektory napájacieho zdroja do programátora a elektrickej siete.

Krok 4.

Zapnúť programátor. Po zapnutí všetky LED indikujúce prácu s obvodom zablikajú a potom ostanú zhasnuté.

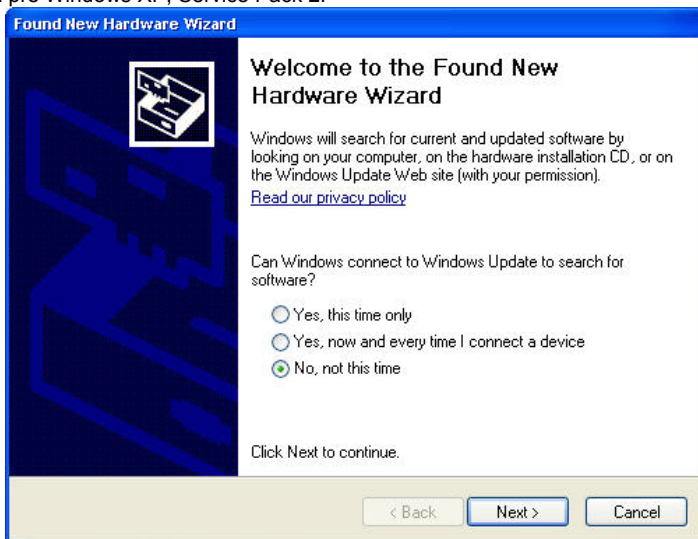
Pre programátory pripojené cez LPT môžete začať prácu s programátorom.

Pre programátory pripojené cez USB pokračujte ďalšími krokmi.

Krok 5.

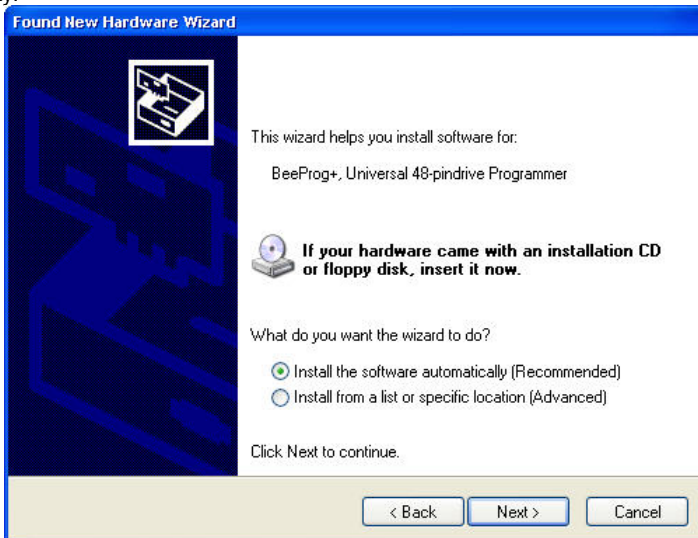
Windows zobrazí "Found new hardware wizard".

Iba pre Windows XP, Service Pack 2:

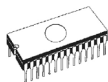


Zvoľte "No, not this time" a potom kliknite na "Next".

Pre všetky:



Zvoľte "Install the software automatically" a potom kliknite na "Next".

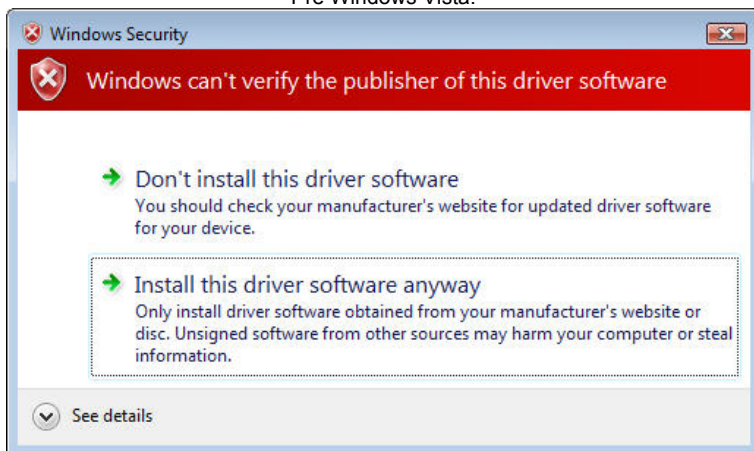


Krok 6.



Kliknite na "Continue Anyway".

Pre Windows Vista:



Kliknite na "Install this driver software anyway"

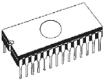
Krok 7.

Kliknutím na "Finish" ukončíte inštaláciu

Krok 8.

Postup inštalácie "Found new hardware wizard" sa spustí pre každý programátor raz (pre BeeHive204 / BeeHive4+ 4 krát). Inštalácia bude pokračovať krokom 5.

Poznámka: Ak použijete iný USB port na Vašom PC pre pripojenie programátora okno "Found new hardware wizard" sa spustí opäť a nainštaluje nový USB ovládač.



Pg4uw

Pg4uw – ovládací program

Program Pg4uw je spoločný ovládací program pre všetky programátory ELNEC, ktorý spoľahlivo funguje pod operačnými systémami Windows a nijako neovplyvňuje operácie bežiacie na pozadí.

Používanie ovládacieho programu

Ovládacie programy pre naše výrobky nie sú v čase expedície infikované žiadnym druhom počítačového vírusu. Pre zvýšenie bezpečnosti je každý náš ovládací program vybavený kontrolným algoritmom, ktorý prípadné budúce napadnutie počítačovým vírusom ohlásí.

Spustenie ovládacieho programu

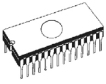
Vo Windows: dvakrát kliknite na ikonu Pg4uw.



Ovládací program po spustení automaticky prehľadá všetky dostupné porty a vyhledá pripojený hocijaký ELNEC programátor. Keďže program Pg4uw je spoločný pre všetky programátory firmy ELNEC, program zisťuje prítomnosť všetkých podporovaných programátorov.

Poznámky: *Ovládací program po spustení kontroluje svoju integritu porovnaním očakávanej a vypočítanej kontrolnej sumy. Ak táto kontrola prebehne úspešne, program zobrazí štandardnú užívateľskú obrazovku a čaká na Vaše pokyny.*

V prípade, že ovládací program nie je schopný komunikácie s programátorom, vypíše hlásenie o chybe s popisom možných zdrojov problémov (programátor nepripojený, zle zasunuté konektory, chyba napájania, printer port je nekompatibilný, alebo je nastavený chybné...). Chybu je potrebné odstrániť a potom stlačiť ľubovoľnú klávesu. Ak sa príčina poruchy neodstráni, po stlačení ľubovoľnej klávesy program pokračuje v činnosti v DEMO móde, teda prístup na programátor nie je možný. Ak sa vám nepodarí zistiť príčinu poruchy, pozrite pokyny v časti **Ak sa vyskytnú problémy**. Rovnako kontrolu komunikácie s programátorom vykonáva ovládací program pred každou činnosťou s programovaným obvodom.



Užívateľská obrazovka

Windows program Pg4uw

The screenshot shows the Pg4uw software interface. The main window is titled "Pg4uw v2.44f11.2007 - univerzálny ovládač program pre Elnece programátory". The interface includes a menu bar (Súbor, Buffer, Obvod, Programátor, Nastavenia, Pomocník), a toolbar with icons for file operations and program control, and a main workspace. A "Log okno" (Log window) is open, displaying a list of status messages for various program models. A "Statistika" (Statistics) panel on the right shows counts for successful, failed, and total operations. Below the log, there are sections for "Adresy (hex)" (Addresses in hex) and "Programátor" (Programmer) details.

Log okno

```
10026: Hľadá sa programátor SI&VRprog ..... Menájdenny.
10027: Hľadá sa programátor BeeHave4, Site #1 .... Menájdenny.
10028: Hľadá sa programátor BeeHave4+, Site #1 ..... Menájdenny.
10029: Hľadá sa programátor BeeHave59, Site #1 .... Menájdenny.
10030: Hľadá sa programátor BeeProg ..... Menájdenny. (EC: 11)
10031: Hľadá sa programátor BeeProg+ ..... Menájdenny. (EC: 01)
10032: Hľadá sa programátor JetProg ..... Menájdenny. (EC: 01)
10033: Hľadá sa programátor LabProg+ ..... Menájdenny.
10034: Hľadá sa programátor LabProg+68IV ..... Menájdenny.
10035: Hľadá sa programátor MEMprog ..... Menájdenny. (EC: 01)
10036: Hľadá sa programátor MEMprog2 ..... Menájdenny.
10037: Hľadá sa programátor MEMprogL ..... Menájdenny.
10038: Hľadá sa programátor PIKprog ..... Menájdenny.
10039: Hľadá sa programátor PIKprog+ ..... Menájdenny. (EC: 01)
10040: Hľadá sa programátor PIKprog2 ..... Menájdenny.
10041: Hľadá sa programátor PREPRON-02 ..... Menájdenny.
10042: Hľadá sa programátor PREPRON-02aLV ..... Menájdenny. (EC:1 [W:66, R:FF])
10043: Hľadá sa programátor SEProg ..... Menájdenny.
10044: Hľadá sa programátor SmartProg ..... Menájdenny. (EC: 01)
10045: Hľadá sa programátor SmartProg2 ..... Menájdenny.
10046: Hľadá sa programátor T51prog ..... Menájdenny. (EC: 01)
10047: Hľadá sa programátor T51prog2 ..... nájdený na porte USB.
10048: Nastavenie komunikácie ... skončené.
10049: Nastavená komunikačná rýchlosť: dosiahnutých 100% z maxima v tomto komunikačnom režime.
10050: Sériové číslo programátora: 576-00320.
```

Statistika

| | |
|----------|---|
| Úspešné: | 0 |
| Chybné: | 0 |
| Celkovo: | 0 |

Reset

Odpočítavanie: **Nepovolené**
Zostáva: 0 z 0
[Naplniť odpočítavanie]

Adresy (hex)

| Obvod | Dig | Veľkosť | Začiatok | Koniec |
|--------|-----|---------|----------|--------|
| Obvod | x8 | 800 | 0 | 7FF |
| Buffer | x8 | 800 | 0 | 7FF |
| Súbor | x8 | - | - | - |

Rozdelenie: **Nie je** Konti.suma: 0007F800h
Serializácia: **Nie je**

Názov súboru:

Programátor

| | | | |
|------|----------|-------|------------|
| Typ: | T51prog2 | Stav: | Prípravený |
| | | Port: | USB |
| | | YES! | Manuálne |

Obvod

| | |
|------|-----------|
| Typ: | nevýbratý |
|------|-----------|

Lišty tlačidiel

Pod hlavným menu sú umiestnené lišty tlačidiel pre rýchle spustenie často používaných menu príkazov. Lišty tlačidiel sú voliteľné a je možné ich skryť pomocou menu **Nastavenia / Zobrazit'**.

Log okno

Obsahuje informácie o priebehu mnohých činností v programe Pg4uw. Činnosti môžu byť:

- spustenie programu Pg4uw
- vyhľadavanie programátora
- načítanie / uloženie súboru alebo projektu
- výber obvodu
- akcie s obvodmi (čítanie, kontrola, programovanie, ...)
- pripojenie / odpojenie aplikácie diaľkového ovládania programu Pg4uw
- a ďalšie činnosti

Obsah Log okna môže byť priebežne ukladaný do súboru. Príslušné nastavenie je k dispozícii cez menu **Nastavenia / Všeobecné nastavenia**.

Panel Adresy

Panel Adresy obsahuje informácie o aktuálnych adresných rozsahoch vybraného obvodu, načítaného súboru a aktuálny stav buffera. U niektorých obvodov je možné nastaviť

užívateľsky adresné rozsahy obvodu a - alebo buffera cez menu **Obvod / Nastavenia súvisiace s obvodom / Operácie s obvodom**.

Panel Adresy ďalej obsahuje informácie o aktuálnom stave funkcií Rozdelenie (Split), Serializácia a kontrolná suma buffera. Pre viac informácií o uvedených funkciách, viď. kapitoly:

- Rozdelenie (Split) Obvod / Nastavenia súvisiace s obvodom / Operácie s obvodom
- Serializácia Obvod / Nastavenia súvisiace s obvodom / Serializácia
- Kontrolná suma Buffer / Kontrolná suma

Panel Programátor

Obsahuje informácie o aktuálne vybratom programátore. Nasledovné informácie o programátore sú zobrazené:

- typ programátora
- port, prostredníctvom ktorého je programátor pripojený k počítaču
- stav programátora, môže nadobúdať niektorú z nasledovných hodnôt:
 - Pripravený - programátor je pripojený, nájdený a pripravený na prácu
 - Nenájdený - programátor nie je nájdený
 - Demo - pokiaľ užívateľ zvolí voľbu Demo v dialógu Hľadať programátor
- nastavenie funkcie YES! - niektoré typy programátorov majú možnosť použiť niekoľko režimov na vykonávanie zvolenej akcie s obvodmi pre dávku viacerých obvodov nasledovne:
 - opakované spúšťanie akcie manuálne z ovládacieho programu pomocou dialógu Opakovať
 - opakované spúšťanie akcie manuálne pomocou tlačidla YES! umiestneného priamo na programátore
 - opakované spúšťanie akcie automaticky - programátor detekuje vybratie obvodu a vloženie ďalšieho obvodu

Pre viac informácií viď. **Programátor / Automatické YES!**

Panel Obvod

Obsahuje informácie o práve vybratom obvode. Informácie o obvode zahŕňajú:

- názov obvodu a výrobca obvodu
- adaptér potrebný pre prácu s obvodom na aktuálne vybratom programátore
- odkaz na podrobnejšie informácie o obvode, tzv. Informácie o obvode, dostupné aj cez menu Obvod / Informácie o obvode
- odkaz na *Špeciálne nastavenia* súvisiace s obvodom, je to len pre niektoré typy obvodov

Panel Štatistika

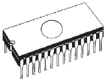
Obsahuje štatistické informácie pre aktuálne vybraný obvod. Štatistické informácie zahŕňajú:

- počet úspešných akcií, neúspešných akcií a celkový počet akcií s obvodom
- stav Odpočítavania indikujúci počet zostávajúcich obvodov

Nastavenia Štatistiky a Odpočítavania sú k dispozícii cez menu **Obvod / Nastavenia súvisiace s obvodom / Štatistika** alebo stlačením pravého tlačidla myši na paneli Štatistika a vybratie menu položky Štatistika.

Panel Súbor

Panel je umiestnený v spodnej časti hlavného okna programu Pg4uw. Panel zobrazuje názov, veľkosť a dátum poslednej zmeny aktuálne načítaného / uloženého súboru alebo projektu.



Zoznam horúcich kláves.

| | | |
|-----------------|----------------------|---|
| <F1> | Obsah | Zobrazenie obsahu pomocníka |
| <F2> | Uložiť | Uloženie buffera na disk |
| <F3> | Čítať | Načítanie súboru do buffera |
| <F4> | Editovať | Zobrazenie/editovanie buffera |
| <F5> | Vybrať z ... | Výber obvodu z posledných 10 vybraných obvodov |
| <Alt+F5> | Vybrať obvod | Výber obvodu podľa triedy, výrobcu |
| <F6> | Kontrola | Kontrola vymazania obvodu |
| <F7> | Čítať | Načítanie obsahu obvodu do buffera |
| <F8> | Verifikovať | Verifikovanie zhodnosti dát z obvodu s bufferom |
| <F9> | Programovať | Programovanie vybraného obvodu |
| <Alt+Q> | Skončiť bez uloženia | Skončenie programu bez uloženia konfigurácie |
| <Alt+X> | Skončiť s uloženíím | Skončenie programu s uloženíím konfigurácie |
| <Ctrl+F1> | Info o obvode | Zobrazenie informácií o vybranom obvode |
| <Ctrl+F2> | Vymazať | Naplnenie buffera zadanými dátami |
| <Ctrl+Shift+F2> | Naplniť dátami | Naplnenie buffera náhodnými dátami |

Súbor

Príkaz umožňuje základnú manipuláciu so súbormi, nastavenie a prezeranie adresárov, zmenu aktuálneho disku a modifikáciu počítačovej a koncovkej adresy buffera pre čítanie a ukladanie súborov v **binárnom**, **MOTOROLA**, **MOS Technology**, **ASCII space**, **Tektronix**, **JEDEC**, **POF** alebo **Intel HEX** formáte. Obsahuje aj položky na načítanie a uloženie projektu.

Súbor / Čítať

Rozpoznáva formát súboru a načítava údaje zo súboru na disku do buffera zvoleným formátom (**binárne**, **MOTOROLA**, **Tektronix**, **MOS Technology**, **ASCII space**, **Intel HEX**, **ASCII space**, **JEDEC** a **POF formát**). Program si pamätá masku, s ktorou sa naposledy pracovalo a túto možno odpamätať do konfiguračného súboru pomocou **Nastavenia / Uložiť nastavenia**.

Stlačením skracovacej klávesy <F3> sa zobrazí toto menu.

Poznámka pre špeciálne x16 formáty:

Intel HEXx16 je Intel Hex file formát s 16 bitovými dátami (wordami) pre TMS320F obvody.

Motorola HEXx16 je Motorola file formát s 16 bitovými dátami (wordami) pre TMS320F obvody.

Zaškrtávacím políčkوم **Automatické rozpoznávanie formátu súboru** sa definuje spôsob určenia formátu načítavaného súboru. Ak je políčko zaškrtnuté, program sa pokúsi automaticky rozpoznať formát súboru ako jeden z podporovaných formátov. Ak sa programu nepodarí zistiť formát súboru, uvažuje sa binárny formát.

Ak políčko nie je zaškrtnuté, tak užívateľ má zadať požadovaný formát súboru zo zoznamu podporovaných formátov. Táto voľba je prednastavená podľa nastavenia v menu **Nastavenia / Všeobecné nastavenia** v záložke **Možnosti načítania súboru** v časti **Formát načítaného súboru**.

Upozornenie: Súbor v ASCII Hex formáte nie je možné identifikovať automaticky. Program ich identifikuje ako binárne. Preto súbor v tomto formáte čítajte s vypnutou voľbou automatického rozpoznávania.

Panel Dodatočné operácie obsahuje nasledovné nastavenia:

Zaškrtnutím políčka **Vymazať buffer pred načítaním** sa aktivuje funkcia vymazania obsahu buffera pred načítaním súboru v binárnom a všetkých HEX formátoch. Toto nastavenie je jednorazové pre každé čítanie súboru. Jeho aktivácia zároveň potláča nastavenie iného mazania buffera pred načítaním v menu **Nastavenia / Všeobecné nastavenia** v záložke **Nastavenia pre Hex súbor** v časti **Vymazať buffer pred načítaním**.

Zaškrtnutím políčka **Swapovať bajty** sa aktivuje/deaktivuje funkcia vzájomného prehodenia susedných bajtov v rámci 16 bitových slov pri čítaní dát zo súboru. Funkcia Swapovať bajty má význam napr. pri čítaní súborov s Motorola reprezentáciou poradia bajtov, tzv. Big Endian.

Poznámka: Big endian a little endian sú termíny udávajúce poradie bajtov, v akom sú bajty ukladané v slovách u multibajtových dátových typov uložených napr. v pamäti počítača. Big endian označuje poradie, kde "big end" – bajt s najväčšou váhou – je uložený ako prvý – na najnižšej adrese. Little endian je poradie, kde "little end" – bajt s najnižšou váhou – je uložený ako prvý – na najvyššej adrese.

Napríklad hexadecimálne číslo 4F52H môže byť uložené v pamäti na adrese 1000H nasledovne:

| Adresa | Big endian | Little endian |
|--------|------------|---------------|
| | system | system |
| 1000H | 4FH | 52H |
| 1001H | 52H | 4FH |

Panel Buffer offset pre načítavanie:

Panel **Buffer offset pre načítavanie** obsahuje nastavenia offsetu pre ukladanie dát do buffera pri čítaní dát zo súboru. Pri každom otvorení okna Načítať súbor sa offset nastaví na voľbu Nie je. Výber položky a zadanie hodnoty offsetu teda platí jednorázovo.

K dispozícii sú nasledovné voľby pre nastavenie offsetu:

Nie je znamená, že sa dáta zo súboru čítajú bez uplatnenia offsetu pre počiatočnú adresu buffera

Kladný offset nastavenie hodnoty offsetu sa pripočíta k aktuálnej adrese pre uloženie dát do buffera. Tento offset sa uplatňuje pre všetky formáty, pričom sa použije organizácia adresovania x8 alebo x16 podľa aktuálne nastaveného zobrazenia buffera.

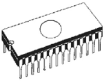
Záporný offset má dve možnosti:

Záporný offset a Automatický záporný offset:

Pre manuálne nastavenie zvolte položku Záporný offset a zadajte hodnotu offsetu.

Pre automatické zistenie záporného offsetu zvolte položku Automatický záporný offset.

Hodnota záporného offsetu sa odčíta od aktuálnej adresy pre uloženie dát do buffera. Záporný offset platí len pre HEX formáty a použije sa vždy s organizáciou x8.

**Poznámky:**

- *Keďže hodnota nastaveného offsetu sa od reálnej adresy odčítava, dbajte na správnosť jeho zadávania. V opačnom prípade sa môže stať, že zmodifikovaná adresa nadobudne zápornú (a teda nepoužiteľnú) hodnotu.*
- *Automatické nastavenie negatívneho offsetu doporučujeme používať len v nevyhnutných prípadoch. Táto voľba má implementovanú heuristickú analýzu, ktorá nemusí korektné spracovať všetky dáta v súbore. Zvlášť kritické to môže byť v súboroch, kde adresný rozsah dát je fragmentovaný a presahuje rámec zvoleného obvodu - niektoré bloky potom môžu byť ignorované.*
- *Automatické nastavenie negatívneho offsetu môže byť výberom niektorých špeciálnych obvodov zablokované. V HEX súboroch súvisiacich s takýmito obvodmi sa rozsah adries musí zachovať a nemožno ho premiestniť ani vypustiť niektoré bloky. V takýchto prípadoch je pre negatívny offset možné len manuálne nastavenie Záporného offsetu.*

Príklad použitia negatívneho offsetu:

Súbor obsahuje dáta v S - formáte fy. Motorola.

Oblasť údajov začína na adrese FFFF0H. Jedná sa o formát S2, teda riadkový záznam obsahuje adresové pole veľkosti 3 byte.

Aby bolo možné prečítať všetky údaje, treba ich do buffera ukladať od adresy 0.

Offset teda nastavíme na hodnotu FFFF0H, čím zabezpečíme, že reálne adresy údajov v súbore budú práve o túto hodnotu znižované a tým zapisované od nulovej adresy do buffera.

Zoznam kódov formátov súborov a chýb

V procese načítavania súboru v niektorom z podporovaných HEX formátov sa môže vyskytnúť chyba. Táto sa vypíše do LOG okna v tvare "Upozornenie: chyba #xyy v riadku rrr", kde xx je kód rozpoznateľného formátu, y je kód chyby a rrr je číslo riadku v dekadickom vyjadrení.

Kódy formátov:

- #00y - binárny
- #10y - ASCII Space
- #20y - Tektronix
- #30y - Extended Tektronix
- #40y - Motorola
- #50y - MOS Technology
- #60y - Intel HEX

Kódy chýb:

- #xx1 - chybný prvý znak - hlavička
- #xx2 - chybný znak v aktuálnom riadku
- #xx3 - chybné CRC
- #xx4 - chybné načítaná adresa
- #xx5 - chybná dĺžka aktuálneho riadku
- #xx6 - príliš veľký negatívny offset
- #xx7 - adresa je mimo rozsah buffera
- #xx8 - chybný typ daného formátu súboru
- #xx9 - súbor nebol načítaný celý

Súbor / Uložiť

Ukladá údaje vytvorené v bufferi, modifikované alebo prečítané z obvodu do súboru na disk zvoleným formátom. Taktiež je možné zadať adresy začiatku a konca bloku buffera, ktorý sa má uložiť.

V súčasnosti sú podporované nasledujúce formáty: **binárny, Intel MOTOROLA, Tektronix, ASCII space, MOS Technology, JEDEC** a **POF** formát. Masky pre zoznam súborov možno prednastaviť príkazom v **Nastavenia / Všeobecné nastavenia** na liste **Možnosti načítania súboru**.

Zaškrtávacím políčkom **Swapovať bajty** sa aktivuje/deaktivuje funkcia vzájomného prehodenia susedných bajtov v rámci 16 bitových slov pri čítaní dát zo súboru. Funkcia Swapovať bajty má význam napr. pri čítaní súborov s Motorola reprezentáciou poradia bajtov, tzv. Big Endian.

Stlačením skracovacej klávesy <F2> sa zobrazí toto menu.

Súbor / Čítať projekt

Tento príkaz slúži na načítanie projektu. Projekt je špeciálny súbor, ktorý obsahuje údaje o obvode a jeho nastaveniach, dáta buffera a niektoré nastavenia programu. Dialóg načítania projektu obsahuje oproti štandardnému dialógu na otvorenie súboru prídavné okno na zobrazenie informácií o aktuálne nalistovanom súbore projektu. Informácie o projekte pozostávajú z nasledovných údajov:

- výrobca a meno obvodu
- dátum a čas vytvorenia projektu
- užívateľom definovaný text na bližší popis projektu

Poznámka: *Serializácia u projektov sa pri načítaní projektu nastaví podľa nasledujúceho postupu:*

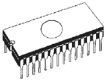
1. *Akceptuje sa nastavenie serializácie uložené v súbore projektu*
2. *Dodatočne sa hľadá externý súbor serializácie priradeného k projektu, a ak sa taký súbor nájde, tak sa prednostne nastaví serializácia vrátane aktuálneho sériového čísla pre najbližšie programovaný obvod podľa údajov v externom súbore serializácie. Nastavenia serializácie z projektu sa v takomto prípade ignorujú.*

Meno externého serializačného súboru sa odvodzuje z mena projektu pridaním prípony ".sn" k menu súboru projektu. Externý serializačný súbor sa predpokladá byť umiestnený v podadresári "serialization\" v adresári, kde je inštalovaný ovládací program.

Príklad: pre projekt s menom "my_project.prj" bude externý serializačný súbor mať meno "my_project.prj.sn" a bude umiestnený v podadresári "...\serialization\" ako "...\serialization\my_project.prj.sn "

Externý súbor serializácie sa vytvára vždy pre konkrétny projekt a to po úspešnej akcii programovania obvodu. Podmienkou je, že bol pred programovaním obvodu načítaný nejaký projekt so zapnutou serializáciou.

Akcia Súbor / Uložiť projekt vykoná úplné zmazanie externého serializačného súboru pre ukladaný projekt, pokiaľ externý serializačný súbor pre ukladaný projekt existuje.



Súbor / Uložiť projekt

Tento príkaz slúži na uloženie projektu. Dialóg uloženia projektu obsahuje oproti štandardnému dialógu na uloženie súboru prídavné okná na zobrazenie informácií o aktuálne ukladanom projekte. Dialóg navyše obsahuje tlačidlo s obrázkom kľúča, ktorým sa môže nastaviť uloženie projektu s heslom. Pre bližšie informácie o projektoch ukladaných s heslom (v tzv. chránenom móde) viď. **Nastavenia / Chránený mód**.

Dialóg **Uložiť projekt** obsahuje dodatočné panely s informáciami a nastaveniami.

V prvom rade obsahuje dve zaškrťavacie políčka a tlačítko s ikonkou kľúča.

Prvé zaškrťavacie políčko s názvom "**Nastaviť Chránený mód pre ukladaný projekt**" a tlačítko so zobrazeným kľúčom slúžia na nastavenie Chráneného módu a hesla pre ukladaný projekt. Projekty uložené v Chránenom móde sú od klasických projektov odlišné v tom, že po ich načítaní do programu Pg4uw sa program prepne do Chráneného módu. Pre bližšie informácie ohľadom Chráneného módu viď. **Nastavenia / Chránený mód**.

Druhé zaškrťavacie políčko s názvom "**Požadovať kontrolnú sumu projektu pred prvým programovaním**" slúži na nastavenie požiadavky na zadanie kontrolnej sumy projektu pred prvým programovaním obvodu z projektu. Dané nastavenie sa prejaví pri, resp. po načítaní projektu. Požiadavka na zadanie kontrolnej sumy projektu je vhodná na dodatočné overenie, že je načítaný správny projekt. Pri použití tejto voľby je vhodné použiť (zaškrtnúť) aj voľbu pre Chránený mód projektu (prvé zaškrťavacie políčko).

Tretie okno odspodu obsahuje informácie a práve vybranom súbore projektu v dialógu **Uložiť projekt**.

Druhé okno odspodu obsahuje informácie o aktuálne vybranom obvode, dátum a čas a verziu programu, v ktorej sa ide projekt uložiť. Tieto informácie teda budú uložené do práve ukladaného projektu.

Spodné okno obsahuje popis projektu vo forme textu, ktorý zvyčajne obsahuje užívateľom definovaný popis projektu, autora projektu a niektoré ďalšie užívateľom zapísané informácie.

Z informácií o projekte užívateľ môže priamo editovať len poslednú časť - užívateľom zadaný ľubovoľný text v spodnom okne dialógu **Uložiť projekt**. Meno a výrobca obvodu, dátum a čas projektu v druhom a treťom okne odspodu sú generované programom automaticky ovládacím programom.

Informácie alebo inak popis projektu slúžia na bližšiu špecifikáciu projektu, čo má pomôcť orientácii užívateľa pri hľadaní požadovaného súboru projektu v dialógu **Čítať projekt**.

Popis projektu sa skladá z nasledujúcich položiek:

- meno a výrobca aktuálne vybratého obvodu v projekte
- dátum a čas vytvorenia projektu
- užívateľom zadaný ľubovoľný text určený na bližší popis projektu

Užívateľ môže priamo editovať len poslednú časť - užívateľom zadaný ľubovoľný text. Meno a výrobca obvodu, dátum a čas projektu a verzia programu sú generované programom automaticky pri každom ukladaní projektu.

Súbor / Znova načítať súbor

Tento príkaz umožňuje znova načítať súbor, ktorý bol prednedávnom použitý. Pokiaľ používate nejaký súbor, tak sa meno daného súboru zarádi do zoznamu posledne používaných súborov v menu Znova načítať súbor. Súborové sú v menu posledne používaných súborov usporiadané podľa času použitia tak, že naposledy používané súbory sú pred dávnejšie používanými súbormi.

Postup pre načítanie súboru zo zoznamu najnovšie používaných súborov:

1. Z menu **Súbor** vyberte položku **Znova načítať súbor**.
2. Rozbalí sa podmenu so zoznamom posledne používaných súborov, v ktorom kliknite súbor, ktorý chcete načítať.

Upozornenie: Pri znova načítaní súboru sa uvažuje formát súboru, v akom bol daný súbor naposledy načítaný alebo uložený. Rovnako sa použije aj funkcia "Swapovať bajty".

Súbor / Znova načítať projekt

Tento príkaz umožňuje znova načítať projekt, ktorý bol prednedávnom použitý. Pokiaľ používate nejaký projekt, tak sa meno daného projektu zarádi do zoznamu posledne používaných projektov v menu **Znova načítať projekt**. Projekty sú v menu posledne používaných projektov usporiadané podľa času použitia tak, že naposledy používané projekty sú pred dávnejšie používanými projektmi.

Postup pre načítanie súboru zo zoznamu najnovšie používaných projektov:

1. Z menu **Súbor** vyberte položku **Znova načítať projekt**.
2. Rozbalí sa podmenu so zoznamom posledne používaných projekt, v ktorom kliknite projekt, ktorý chcete načítať.

Súbor / Vlastnosti projektu

Tento príkaz slúži na zobrazenie/editovanie vlastností aktuálneho projektu. Vlastnosťami projektu sa rozumie popis projektu. Popis projektu pozostáva z nasledujúcich položiek:

- meno a výrobca aktuálne vybraného obvodu v projekte
- dátum a čas vytvorenia projektu
- užívateľom zadaný ľubovoľný text určený na bližší popis projektu

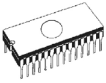
Užívateľ môže priamo editovať len poslednú časť - užívateľom zadaný ľubovoľný text. Meno a výrobca obvodu, dátum a čas projektu a verzia programu sú generované programom automaticky pri každom ukladaní projektu.

Súbor / Načítať kódovaciu tabuľku...

Načítava dáta zo súboru v binárnom tvare z disku a ukladá ich do oblasti pamäte ako obsah kódovacej (encryption, security...) tabuľky.

Súbor / Uložiť kódovaciu tabuľku...

Zapíše obsah kódovacej (encryption, security...) tabuľky do súboru na disk v binárnom tvare.



Súbor / Skončiť bez uloženia konfigurácie...

Príkaz ukončí program, uvoľní alokovanú časť operačnej pamäti, zruší buffer na disku (ak bol vytvorený) a vráti riadenie operačnému systému.

Súbor / Skončiť s uložením konfigurácie...

Príkaz ukončí program, uvoľní alokovanú časť operačnej pamäti, zruší buffer na disku (ak bol vytvorený) a vráti riadenie operačnému systému.

Príkaz navyše uloží aktuálnu konfiguráciu programu na disk.

Buffer

Menu **Buffer** ponúka široké možnosti práce s bufferom ako sú blokové operácie, mazanie obsahu, naplnenie reťazcom, 4 druhy kontrolnej sumy a samozrejme editácia, ktorá v sebe zahŕňa ďalšie služby (vyhladávanie reťazcov a ich náhrada, výstup na tlačiareň atď.).

Buffer / Zobrazit/Editovať

Tento príkaz umožňuje **prezeranie** (view režim) alebo **editáciu** (edit režim) údajov v bufferi. Použitím kurzorových kláves alebo myši je možné pohybovať sa v celom obsahu buffera a tak vybrať príslušný blok na editáciu. Modifikované dáta sú farebne odlišené.

Možno použiť skracovaciu klávesu <F4>.

Zobrazit/Editovať Buffer

- F1** Zobrazí pomocníka k danému oknu.
- F2** Naplní zvolenú časť buffera zadaným reťazcom. Ten je možné zadávať v ASCII alebo HEX tvare. Maximálna dĺžka reťazca je 16 ASCII znakov.
- Ctrl+F2** Vymaže buffer aktuálnou mazacou konštantou (pre EPROM je to 0FFH).
- Ctrl+Shift+F2** Naplní buffer náhodnými dátami.
- F3** Skopíruje zadaný blok buffera na zvolené miesto. Cieľový priestor kde sa blok uloží sa smie prekryvať so zdrojovým.
- F4** Prenesie zdrojový blok buffera do cieľového. Bloky sa smú prekryvať. Zdrojový blok (alebo jeho časť) sa po prenesení vyplní aktuálnou mazacou konštantou (pre EPROM je to 0FFH).
- F5** Vykoná vzájomnú zámenu dvojíc bajtov v zadanom bloku buffera. Predpokladá sa, že oblasť začína na párnej adrese a obsahuje párny počet bajtov. Ak podmienky nie sú splnené, program si adresy upraví sám (spodnú hranicu posunie na nižšiu párnú adresu a/alebo hornú hranicu na vyššiu nepárnu adresu).
- F6** Tlačí buffer.
- F7** Nájde reťazec (max. dĺžka 16 ASCII znakov).
- F8** Nájde a nahradí reťazec (max. 16 ASCII znakov).
- F9** Skočí na zadanú adresu.
- F10** Zmena módu prezeranie / editácia.
- F11** Prepína spôsob zobrazenia buffera medzi 8 bitovým a 16 bitovým zobrazením. Prepínanie sa dá previesť aj myšou kliknutím na tlačidlo vedľa indikátora módu prezerania alebo editácie buffera. Tlačidlo zároveň indikuje aktuálny mód zobrazenia (8 bit alebo 16 bit).

| | |
|-----------------------|---|
| F12 | Dialóg na výpočet kontrolnej sumy umožňuje vypočítať kontrolné súčty rôznych typov na vybranom bloku dát v bufferi. |
| ESC | Ukončí editáciu a prezeranie buffera. |
| Šípky | Kurzor hore, dole, vpravo a vľavo. |
| Home/End | Skočí na začiatok / koniec riadku. |
| PgUp/PgDn | Skočí na predchádzajúcu / novú stranu. |
| Ctrl+PgUp/PgDn | Skočí na začiatok / koniec strany. |
| Ctrl+Home/End | Skočí na začiatok / koniec obvodu. |
| Shift+Home/End | Skočí na začiatok / koniec .buffera |
| Backspace | Posunie kurzor o jednu pozíciu vľavo (späť). |

Poznámka: *znaky 20H - FFH (režim ASCII) a čísla 0..9, A..F (režim HEX) bezprostredne menia obsah editovaného buffera.*

Upozornenie: *editovanie ASCII znakov pre word obvody je zakázané*

Tlačiť buffer

Tento príkaz umožňuje vytlačiť obsah zvolenej časti buffera na tlačiarňu alebo ho zapísať do súboru na disk. Vyžíva sa pritom užívateľom zvolený externý textový editor, v ktorom sa obsah zvolenej časti buffera zobrazí a je možné ho odtiaľ aj vytlačiť. Štandardne je predvolený textový editor **Notepad.exe**, ktorý je bežnou súčasťou všetkých verzií systému Windows.

V dialógu Tlačiť buffer sú nasledujúce nastavenia:

Začiatok bloku

Definuje v bufferi počiatočnú adresu bloku, ktorý je určený na tlač (zobrazenie). Pri bajtovom zobrazení buffera sa adresujú bajty a pri 16 bitovom zobrazení buffera sa adresujú 16 bitové slová.

Koniec bloku

Definuje v bufferi koncovú adresu bloku, ktorý je určený na tlač (zobrazenie). Pri bajtovom zobrazení buffera sa adresujú bajty a pri 16 bitovom zobrazení buffera sa adresujú 16 bitové slová.

Externý editor

Definuje cestu a meno externého programu, ktorý sa má použiť ako textový editor pre zvolený blok buffera. Štandardne je predvolený textový editor Notepad.exe, ktorý je bežnou súčasťou všetkých verzií systému Windows. Užívateľ si môže nastaviť ľubovoľný iný textový editor, napr. Wordpad.exe, ktorý umožňuje pracovať aj s veľkými súbormi. Samotné vytlačenie buffera užívateľ realizuje v zvolenom textovom editore.

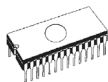
Nastavenie cesty a meno textového editora sa pri skončení ovládacieho programu automaticky ukladá na disk.

Hľadať text

Toto okno slúži na **vyhľadanie** reťazca v bufferi. Do položky **Hľadať text** napíšete reťazec, ktorý chcete vyhľadať a po potvrdení sa začne vyhľadávanie. Po stlačení tlačidla **Zrušiť** sa okno **Hľadať text** zavrie.

Položka **Smer** určuje smer (Nahor alebo Nadol) vyhľadávania od aktuálnej pozície kurzora v móde pozeranie. V móde editácia sa prehľadáva celý buffer.

Položka **Odkiaľ** určuje začiatok (od kurzora alebo v celom bufferi) vyhľadávania.



Nahradiť text

Tento nástroj slúži na nahradenie nájdených reťazcov iným reťazcom. Do položky **Hľadať text** zadajte reťazec, ktorý chcete nahradiť a do položky **Nahradiť čím** reťazec, ktorým ho chcete nahrádzať.

V položke **Nastavenia** nastavíte voľbu, či má program pýtať potvrdenie na nahradenie reťazca alebo nie.

Položka **Odkiaľ** určuje začiatok (od kurzora alebo v celom bufferi) vyhľadávania.

Položka **Smer** určuje smer (**Nahor** alebo **Nadol**) vyhľadávania od aktuálnej pozície kurzora v Edit móde. Vo view móde sa prehľadáva celý buffer.

Stlačením <Esc> alebo kliknutím na tlačidlo **Zrušiť** sa okno Nahradiť text zavrie.

Kliknutím na tlačidlo **Nahradiť** sa okno Nahradiť text zavrie a zobrazí sa okno **Otázka**, v ktorom sú nasledujúce voľby:

| | |
|-------------------------|--|
| Áno | nahradí nájdenú položku a vyhľadá ďalšiu |
| Nie | nájde ďalšiu položku |
| Nahradiť všetko | nahradí všetky nájdené položky |
| Ukončiť hľadanie | ukončí hľadanie a nahrádzanie reťazcov |

Zobrazíť/Editovať buffer pre PLD

| | |
|-----------------------|--|
| Ctrl+F2 | Vymazanie buffera predvolenou konštantou. |
| Ctrl+Shift+F2 | Naplnenie buffera náhodnými dátami. |
| F9 | Skok na adresu... |
| F10 | Zmena módu prehliadanie / editácia. |
| F11 | Prepínanie spôsobu zobrazenia buffera medzi 1 bitovým a 8 bitovým zobrazením. Prepínanie sa dá previesť aj myšou kliknutím na tlačidlo vedľa indikátora módu prezerania alebo editácie buffera. Tlačidlo zároveň indikuje aktuálny mód zobrazenia (1 bit alebo 8 bit). |
| Šípky | Kurzor hore, dole, vpravo a vľavo. |
| Home/End | Skok na začiatok / koniec riadku. |
| PgUp/PgDn | Skok na predchádzajúcu / novú stranu. |
| Ctrl+PgUp/PgDn | Skok na začiatok / koniec buffera. |
| Ctrl+Home/End | Skok na začiatok / koniec strany. |
| Backspace | Vráti kurzor o jednu pozíciu vľavo. |

Poznámka: *Znaky 0 a 1 priamo menia obsah editovanej oblasti.*

Buffer / Vyplniť blok

Príkaz umožňuje naplniť zvolenú časť buffera zadaným reťazcom. Ten je možné zadávať v ASCII alebo HEX tvare. Maximálna dĺžka reťazca je 16 ASCII znakov.

Buffer / Kopírovať blok

Tento príkaz kopíruje zadaný blok buffera na zvolené miesto. Cieľový priestor kde sa blok uloží sa smie prekrývať so zdrojovým.

Buffer / Presunúť blok

Tento príkaz prenáša zdrojový blok buffera do cieľového. Bloky sa smú prekryvať. Zdrojový blok (alebo jeho časť) sa po prenesení vyplní aktuálnou mazacou konštantou (pre EPROM je to 0FFH).

Buffer / Swapovať blok

Vykoná sa vzájomná zámena dvojíc bajtov v zadanom bloku buffera. Predpokladá sa, že oblasť začína na párnej adrese a obsahuje párny počet bajtov. Ak podmienky nie sú splnené, program si adresy upraví sám (spodnú hranicu posunie na nižšiu párnú adresu a/alebo hornú hranicu na vyššiu nepárnu adresu).

Na výber sú nasledujúce módy zámeny bajtov:

- Prehodiť 2-bajty v rámci 16-bit slov prehodiť bajty po dvojiciach v rámci 16-bit slov.
- Prehodiť 4-bajty v rámci 32-bit slov prehodiť bajty po štvoriciach v rámci 32-bit slov.
- Prehodiť nibbly vo vnútri bajtov prehodiť horné a dolné štvorice bitov vnútri bajtov.

Príklady jednotlivých typov zámeny bajtov v bufferi:

Obsahy buffera po vykonaní akcií zámeny bajtov v bloku s adresami 0 až N pre jednotlivé typy zámen sú uvedené v nasledujúcej tabuľke:

| Adresa | Pôvodné dáta | Prehodiť 2-bajty v rámci 16-bit slov | Prehodiť 4-bajty v rámci 32-bit slov | Prehodiť nibbly vo vnútri bajtov |
|--------|--------------|--------------------------------------|--------------------------------------|----------------------------------|
| 0000h | b0 | b1 | b3 | b0m |
| 0001h | b1 | b0 | b2 | b1m |
| 0002h | b2 | b3 | b1 | b2m |
| 0003h | b3 | b2 | b0 | b3m |
| 0004h | b4 | b5 | b7 | b4m |
| 0005h | b5 | b4 | b6 | b5m |
| 0006h | b6 | b7 | b5 | b6m |
| 0007h | b7 | b6 | b4 | b7m |

b0, b1, b2, .. označenie hodnôt jednotlivých bajtov v pôvodnom bufferi na adresách 0, 1, 2, .. Kde b0m, b1m, b2m, ... označujú pôvodné bajty b0, b1, b2, ... bitovo modifikované podľa nasledovných pravidiel:

| | |
|---|---|
| pôvodný Bajte | bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0 |
| modifikovaný Bajte (prehodenia štvoric bitov) | bit 3 bit 2 bit 1 bit 0 bit 7 bit 6 bit 5 bit 4 |

Buffer / Vymazať

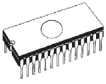
Obsah buffera je vyplnený aktuálnou mazacou konštantou (pre EPROM je to 0FFH).

Možno použiť skracovaciu klávesu <Ctrl+F2>

Buffer / Vyplniť náhodnými dátami

Táto funkcia naplní buffer náhodnými dátami.

Pre túto funkciu je rezervovaný kľúč < Ctrl+Shift+F2>



Buffer / Duplikovať obsah buffera

Tento príkaz umožňuje duplikovať obsah buffera z oblasti zdrojovej pamäte EPROM do oblasti cieľovej EPROM. Duplikovanie buffera je vhodné napr. ak sa má použiť pamäť EPROM 27C512 namiesto pamäte EPROM 27C256.

Poznámka: Pre duplikovanie buffera sa vždy používa adresa začiatku buffera rovná nule.

Buffer / Kontrolná suma

Kontrolná suma dát buffera programu Pg4uw je užitočná pre kontrolu, či dáta v bufferi sú správne.

Pg4uw obsahuje nasledujúce možnosti súvisiace s kontrolnými sumami:

- **Automatický výpočet kontrolnej sumy** so zobrazením aktuálnej kontrolnej sumy v hlavnom okne programu Pg4uw v paneli Adresy a tiež v Log okne programu.
- Dialóg **Kontrolná suma** obsahuje nastavenia a nástroje na výpočet viacerých typov kontrolných súm dát z buffera Pg4uw.

Dialógu **Kontrolná suma** obsahuje nastavenia pre obidva módy. Tento dialóg obsahuje 2 záložky:

- **Počítadlo kontrolnej sumy**
- **Nastavenia hlavnej kontrolnej sumy**
- Záložka **Nastavenia hlavnej kontrolnej sumy** umožňuje nastaviť módy **Automatického výpočtu kontrolnej sumy**:
 - Zaškrtnuté políčko **Použiť užívateľské adresy pre hlavnú kontrolnú sumu**
 - Nezaškrtnuté globálne adresy Buffer začiatok a Buffer koniec budú použité pri výpočte kontrolnej sumy
 - Zaškrtnuté užívateľom nastavené adresy budú použité pri výpočte kontrolnej sumy
 - Polia **Od adresy** a **Po adresu** sa používajú na zadanie rozsahu adries pre výpočet hlavnej kontrolnej sumy. Adresy sú použiteľné iba v prípade zaškrtnutia políčka **Použiť užívateľské adresy pre hlavnú kontrolnú sumu**.
 - Výberové pole **Typ kontrolnej sumy** umožňuje vybrať si požadovaný typ hlavnej kontrolnej sumy
 - Pole **Kontrolná suma** obsahuje aktuálnu hodnotu vypočítanej kontrolnej sumy
 - Tlačítko **Použiť** sa používa na potvrdenie nastavení pre hlavnú kontrolnú sumu. Všimnite si, že stlačením tohto tlačítka sa stratia predchádzajúce nastavenia.
 - Tlačítko **Zavrieť** sa používa na zatvorenie tohto dialógu.
 - Pole **Vynechať dáta z výpočtu kontrolnej sumy** obsahuje nastavenia pre určenie, ktoré bloky buffera sa nebudú zahrňovať do výpočtu kontrolnej sumy. Je to užitočné napríklad pre serializáciu. Serializácia modifikuje dáta na špecifických adresách v bufferi. Preto je problém skontrolovať kontrolnú sumu buffera, keď dáta na niektorých adresách sú menené serializáciou pred programovaním každého obvodu. Ak časť buffera, ktorú používa serializácia bude vynechaná pri výpočte kontrolnej sumy, táto sa už nebude meniť kvôli zmenám urobeným serializáciou. Môže byť špecifikovaný jeden alebo viac výnimkových blokov.
- Záložka **Kontrolná suma** obsahuje počítadlo kontrolnej sumy.

Možné nastavenia:

 - **Od adresy** Štartovacia adresa vybraného bloku pre výpočet kontrolnej sumy v bufferi. Adresa je zadaná v bajtoch.

- **Po adresu** Koncová adresa vybraného bloku pre výpočet kontrolnej sumy v bufferi. Adresa je zadaná v bajtoch.

- Výberové pole **Typ kontrolnej sumy**

| | |
|-------------------|---|
| Byte | Sumovanie po bajtoch do "word". Príznak CY je ignorovaný. |
| Word | Sumovanie po slovách do "word". Príznak CY je ignorovaný. |
| Byte (CY) | Sumovanie po bajtoch do "word". Príznak CY sa pripočítava k výsledku. |
| Word (CY) | Sumovanie po slovách do "word". Príznak CY sa pripočítava k výsledku. |
| CRC-CCITT | Sumovanie po bajtoch do "word" použitím $VÝSELEK = PREDCHÁDZAJÚCI + (x^{16} + x^{12} + x^5 + 1)$ |
| CRC-XModem | Sumovanie po bajtoch do "word" použitím $VÝSELEK = PREDCHÁDZAJÚCI + (x^{16} + x^{15} + x^2 + 1)$ |
| CRC-32 | sumovanie po bajtoch do DWORD-u použitím štandardného CRC-32 algoritmu |

Stípec označený ako **Neg.** je negácia sumy tak, že platí $SUM + NEG. = FFFFH$.

Stípec označený **Doplňok** je doplnok k sume tak, že platí $SUM + SUPPL. = 0$ (+ prenos).

Termín **Word** označuje slovo s dĺžkou 16 bitov.

Termín **DWORD** označuje slovo s dĺžkou 32 bitov.

Sú podporované aj výpočty kontrolných súm typu "hash":

MD5 MD5 hash sa skladá z 32 hexadecimálnych číslic (128 bitov)

SHA-1 "Secure Hash Standard" sa skladá zo 40 hexadecimálnych číslic (160 bitov)

- Pole **Vložiť kontrolnú sumu**

Toto pole obsahuje nastavenia pre príkaz **Vypočítať a vložiť**

- **Vložiť kontrolnú sumu**

Výber typu kontrolnej sumy, ktorý bude zapísaný do buffera, keď sa vykoná príkaz **Vypočítať a vložiť**

- **Vložiť na adresu**

Adresa buffera, kde bude zapísaná kontrolná suma po vykonaní príkazu **Vypočítať a vložiť**. Adresa nemôže byť určená vo vnútri rozsahu <Od adresy> do <Po adresu>. Adresa je zadaná v bajtoch.

- **Veľkosť**

Nastavenie veľkosti výsledku kontrolnej sumy. Veľkosť môže byť **Bajt** (8 bitov), **Word** (16 bitov) alebo **DWORD** (32 bitov). Ak je veľkosť menšia ako vybraná veľkosť kontrolnej sumy potom bude zapísaný do buffera len dolný bajt kontrolnej sumy.

Poznámka: Pri výbere **Word** (16 bitov) bude dolný bajt kontrolnej sumy uložený na miesto určené v položke **Vložiť na adresu a horný bajt na adresu o jedna vyššie**. Podobne to platí aj pre veľkosť **DWORD** (32 bitov).

- **Mód poradia bajtov pre sumy typu Word a Word (CY)**

Obsahuje nastavenia poradia bajtov pri sčítavaní slovných (nie bajtových) kontrolných súm typu **Word** a **Word (CY)**.

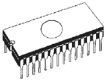
Na výber sú dve možnosti: **Little endian** a **Big Endian**. Popis oboch nastavení nájdete v časti **Word sum Little Endian (x16)** a **Word sum Big Endian (x16)** odstavca **Automaticky výpočet kontrolnej sumy uvedeného vyššie**.

- Tlačítko **Vypočítať**

Tlačítkom sa vykoná výpočet kontrolných súm vybraného bloku dát z buffera. Blok dát je určený adresami **Od adresy...Po adresu**. Obsah buffera sa pri ani po výpočte nemení.

- Tlačítko **Vypočítať a vložiť**

Tlačítkom sa vykoná výpočet kontrolných súm vybraného bloku dát z buffera a po skončení výpočtu sa zvolená vypočítaná kontrolná suma zapíše do buffera na adresu určenú hodnotou zadanou v políčku **Vložiť na adresu**. Funkcia **Vypočítať a vložiť** je



podporovaná len pre kontrolné sumy typu Byte, Word, CRC-CCITT and CRC-XMODEM.

- Tlačítko **Zavrieť**
Zatvorí dialóg Kontrolná suma.

Hlavná kontrolná suma zobrazená v hlavnom okne programu

Kontrolná suma zobrazená v hlavnom okne programu v paneli "Adresy" zobrazuje aktuálnu súčtovú sumu dát v hlavnom bufferi.

- Rozsah adries, z ktorých sa bude počítať kontrolná suma má dve možnosti nastavenia:
 - Kontrolná suma je vypočítaná ako súčet dát v bufferi v rozsahu adries "Začiatok Buffera" a "Koniec Buffera". Hodnoty "Začiatok Buffera" a "Koniec Buffera" sú zobrazené v tabuľke "Adresy" v hlavnom okne programu
 - Užívateľom nastavený rozsah adries.

Nastavenie sa nachádza v dialógu **Kontrolná suma** na karte **Nastavenia hlavnej kontrolnej sumy**, zaškrťavacie políčko **Použiť užívateľské adresy pre hlavnú kontrolnú sumu**. Užívateľské nastavenie rozsahu je indikované reťazcom [od adresy...po adresu] za kontrolnou sumou v hlavnom okne programu v bloku Adresy.

Nastavenie rozsahu adries pre výpočet kontrolnej sumy je priradený aktuálne vybranému obvodu. Po výbere nového obvodu je adresný rozsah nastavený na default hodnoty „Buffer start“ a „Buffer end“ pre vybraný obvod. Rozsah adries pre výpočet kontrolnej sumy sa tiež ukladá do projekt súborov.

- Sú viaceré možnosti nastavenia výpočtu kontrolnej sumy buffera:
 - **Byte sum (x8)** - Dáta v bufferi sú sčítavané po bajtoch, bez ohľadu na aktuálnu organizáciu buffera (x8/x16/x1). Tento mód je indikovaný výpisom reťazca (x8) za hodnotou kontrolnej sumy v hlavnom okne programu.
 - **Word sum Little Endian (x16)** - Dáta v bufferi sú sčítavané po slovách so šírkou 16 bitov. Tento mód je indikovaný výpisom reťazca (x16 LE) za hodnotou kontrolnej sumy v hlavnom okne programu. Výraz Little Endian znamená, že suma sa počíta sčítavaním 16 bitových slov načítavaných z buffera v poradí nižší bajt slova z nižšej adresy a vyšší bajt slova z vyššej adresy buffera.
 - **Word sum Big Endian (x16)** - Dáta v bufferi sú sčítavané po slovách so šírkou 16 bitov. Tento mód je indikovaný výpisom reťazca (x16 BE) za hodnotou kontrolnej sumy v hlavnom okne programu. Výraz Big Endian znamená, že suma sa počíta sčítavaním 16 bitových slov načítavaných z buffera v poradí nižší bajt slova z vyššej adresy a vyšší bajt slova z nižšej adresy buffera.
 - **CRC-CCITT** - Dáta v bufferi sú sčítavané po bajtoch do 16-bitového slova podľa štandardného algoritmu CRC-CCITT.
 - **CRC-XMODEM** - Dáta v bufferi sú sčítavané po bajtoch do 16-bitového slova podľa štandardného algoritmu CRC-XMODEM.
 - **CRC-16** - Dáta v bufferi sú sčítavané po bajtoch do 16-bitového slova podľa štandardného algoritmu CRC-16.
 - **CRC-32** - Dáta v bufferi sú sčítavané po bajtoch do 32-bitového slova podľa štandardného algoritmu CRC-32.
 - **MD5 a SHA-1** - Podpora pre typy hash súm sa pripravuje.

Módy sčítavania kontrolnej sumy sa dajú prepínať v menu, ktoré sa zobrazí po kliknutí na indikátor módu kontrolnej sumy v hlavnom okne programu. Ďalšou možnosťou sú klávesové skratky **Shift+Ctrl+1** pre Byte sum (x8), **Shift+Ctrl+2** pre Word sum Little Endian (x16) a **Shift+Ctrl+3** pre Word sum Big Endian (x16), ... atď.

Taktiež sa typ hlavnej kontrolnej sumy dá zvoliť v nastaveniach v dialógu Kontrolná suma.

Kontrolná suma sa ukladá do konfiguračného a projektového súboru. Nastavenie v projektovom súbore má vyššiu prioritu.

Obvod

Menu **Obvod** obsahuje funkcie pre prácu so zvoleným programovateľným obvodom - výber obvodu, prečítanie obsahu obvodu do buffera, kontrola vymazania, programovanie a verifikácia naprogramovaných dát.

Obvod / Vybrať z prednastavených obvodov

Tento príkaz umožňuje výber obvodu zo zoznamu obvodov, s ktorými sa naposledy pracovalo. Obvody sú uložené v kruhovom bufferi, ktorý si pamätá posledných 20 obvodov včítane nastavenia volieb pre programovanie. V bufferi môžu byť uložené aj rovnaké obvody líšiace sa práve v nastavení volieb pre programovanie. Na výber obvodu použite výberové klávesy. Tento zoznam sa ukladá na disk príkazom **Súbor / Skončiť s uložením konfigurácie....**

Klávesou **<Ctrl+F1>** získate prídavné informácie o obvode, na ktorom sa práve nachádza výberový pásik. Informácie obsahujú pevnú a pružnú časť. Pevná časť obsahuje veľkosť obvodu, organizáciu, programovací algoritmus a zoznam programátorov (včítane požadovaných modulov), ktoré tento obvod podporujú. Pružná časť môže obsahovať informácie o puzdrení a rôzne iné všeobecné informácie o danom obvode.

Klávesou **** môžete vymazať ľubovoľný obvod zo zoznamu, ktorý sa tam dostal napr. omylom. Nie je možné opakovaným volaním tejto služby buffer úplne vyprázdniť. Posledný obvod tam zostane a klávesa **** sa nebude uplatňovať.

Obvod / Vybrať obvod...

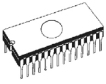
Toto okno umožňuje výber požadovaného obvodu z knižnice podporovaných obvodov aktuálnym programátorom. Obvod je možné vybrať podľa **mena**, podľa **typu** alebo podľa **výrobcu**.

Poznámka: *Názvy podporovaných obvodov v programe neobsahujú všetky znaky zobrazené na obvode alebo uvedené v časti popisujúcej značenie obvodu v dokumentácii k obvodu. Názvy podporovaných obvodov v programe obsahujú len znaky potrebné na rozlíšenie typov obvodov z hľadiska programovania obvodov, a neobsahujú teda napríklad značenia teplotného rozsahu, rýchlosti, typu púzdenia apod.. Pokiaľ sú nepodstatné znaky na konci názvu obvodu, tak sa v programe vôbec neuvádzajú. Pokiaľ sú nepodstatné znaky vnútri názvu obvodu, tak sa nahrádzajú znakmi x.*

Príklady:

- 1) Obvody Am27C512-150, Am27C512-200 a Am27C512-250 sú v programe zobrazené len raz, pod názvom Am27C512
- 2) Obvod S29GL064N11TF1010 je v programe zobrazený ako S29GL064NxxTxx01

Poznámka 2: *Ak je nejaký obvod uvedený v databáze dvakrát, pričom druhýkrát je s dovetkom x16, znamená to, že programovací algoritmus je zrealizovaný pre wordový režim, ktorý je rýchlejší.*



Vybratý obvod je automaticky uložený do buffera obvodov, s ktorými sa naposledy pracovalo (max. 20 obvodov). Pristupuje sa k nemu príkazom **Obvod / Vybrať z prednastavených obvodov**.

Kliknutím na „**Informácie o obvode**“ alebo klávesou **<Ctrl+F1>** získate prídavné informácie o obvode, na ktorom sa práve nachádza výberový pásik. Informácie obsahujú pevnú a pružnú časť. Pevná časť obsahuje veľkosť obvodu, organizáciu, programovací algoritmus a zoznam programátorov (včítane požadovaných modulov), ktoré tento obvod podporujú. Pružná časť môže obsahovať informácie o puzdrení a rôzne iné všeobecné informácie o danom obvode. Aktuálne zobrazený zoznam obvodov je možné uložiť do textového súboru kliknutím na „**Uložiť aktuálne zobrazený zoznam obvodov do súboru**“.

Vybrať obvod... / Všetky

Toto okno umožňuje výber požadovaného obvodu z knižnice podporovaných obvodov aktuálnym programátorom. Podporované obvody sú zobrazené vo forme zoznamu.

Obvod môže byť vybratý dvojklikom na riadok s požadovaným výrobcom a číslom obvodu. Na výber obvodu môžete využiť okrem výberových kláves aj možnosť priameho zadania časti názvu výrobcu a/alebo obvodu vo výberovom riadku (oddeľovací znak je **<Space>**). Výber potvrdíte stlačením **<Enter>** alebo kliknutím na tlačidlo **OK**.

Stlačením klávesy **<Esc>** alebo kliknutím na tlačidlo **Cancel** môžete kedykoľvek zrušiť výber bez vplyvu na aktuálne vybraný obvod.

Vybratý obvod je automaticky uložený do buffera obvodov, s ktorými sa naposledy pracovalo. Pristupuje sa k nemu príkazom **Obvod / Vybrať z prednastavených obvodov**.

Klávesou **<Ctrl+F1>** získate prídavné informácie o obvode, na ktorom sa práve nachádza výberový pásik. Informácie obsahujú pevnú a pružnú časť. Pevná časť obsahuje veľkosť obvodu, organizáciu, programovací algoritmus a zoznam programátorov (včítane požadovaných modulov), ktoré tento obvod podporujú. Pružná časť môže obsahovať informácie o puzdrení a rôzne iné všeobecné informácie o danom obvode.

Vybrať obvod ... / Iba vybrané typy

Tento príkaz umožňuje výber obvodu podľa typu. Obvody podporované aktuálnym programátorom sú logicky rozdelené na triedy a podtriedy. Použitím myši a kurzorových kláves najprv zvolíte triedu (napr. EPROM) a ak existuje aj podtriedu (napr. EPROM 2Kx8 (2716)). Ovládací program poskytne zoznam obvodov prislúchajúcich zvolenej triede a podtriede.

Obvod môže byť vybratý dvojklikom na riadok s požadovaným výrobcom a číslom obvodu. Na výber obvodu môžete využiť okrem výberových kláves aj možnosť priameho zadania časti názvu výrobcu a/alebo obvodu vo výberovom riadku (oddeľovací znak je **<Space>**). Výber potvrdíte stlačením **<Enter>** alebo kliknutím na tlačidlo **OK**.

Stlačením klávesy **<Esc>** alebo kliknutím na tlačidlo **Cancel** môžete kedykoľvek zrušiť výber bez vplyvu na aktuálne vybraný obvod.

Vybratý obvod je automaticky uložený do buffera obvodov, s ktorými sa naposledy pracovalo. Pristupuje sa k nemu príkazom **Obvod / Vybrať z prednastavených obvodov**.

Klávesou **<Ctrl+F1>** získate prídavné informácie o obvode, na ktorom sa práve nachádza výberový pásik. Informácie obsahujú pevnú a pružnú časť. Pevná časť obsahuje veľkosť obvodu, organizáciu, programovací algoritmus a zoznam programátorov (včítane požadovaných modulov), ktoré tento obvod podporujú. Pružná časť môže obsahovať informácie o puzdrení a rôzne iné všeobecné informácie o danom obvode.

Vybrať obvod ... / Iba vybraný výrobca

Týmto príkazom zvolíte potrebný typ obvodu podľa výrobcu. Použitím myši a kurzorových kláves najprv zvolíte výrobcu v zozname výrobcov. Ovládací program poskytne zoznam obvodov prislúchajúcich zvolenému výrobcovi.

Obvod môže byť vybraný dvojklikom na riadok s požadovaným výrobcom a číslom obvodu. Na výber obvodu môžete využiť okrem výberových kláves aj možnosť priameho zadania časti názvu výrobcu a/alebo obvodu vo výberovom riadku (oddeľovací znak je **<Space>**). Výber potvrdíte stlačením **<Enter>** alebo kliknutím na tlačidlo **OK**.

Stlačením klávesy **<Esc>** alebo kliknutím na tlačidlo **Cancel** môžete kedykoľvek zrušiť výber bez vplyvu na aktuálne vybraný obvod.

Vybraný obvod je automaticky uložený do buffera obvodov, s ktorými sa naposledy pracovalo. Pristupuje sa k nemu príkazom **Obvod / Vybrať z prednastavených obvodov**.

Klávesou **<Ctrl+F1>** získate prídavné informácie o obvode, na ktorom sa práve nachádza výberový pásik. Informácie obsahujú pevnú a pružnú časť. Pevná časť obsahuje veľkosť obvodu, organizáciu, programovací algoritmus a zoznam programátorov (včítane požadovaných modulov), ktoré tento obvod podporujú. Pružná časť môže obsahovať informácie o puzdrení a rôzne iné všeobecné informácie o danom obvode.

Obvod / Vybrať EPROM / Flash pomocou ID

Túto službu použijete v prípade automatického výberu EPROM alebo Flash prečítaním jej identifikačných údajov. Jedná sa o prečítanie kódu výrobcu a výrobku, ktoré sú v obvode implementované. Tento spôsob výberu sa dá aplikovať len na obvody, ktoré ID podporujú. Ak obvod nepodporuje ID, program zobrazí varovné hlásenie, že sa jedná o neznámy resp. nepodporovaný obvod.

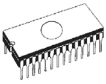
Ak sa detekuje viacero EPROM alebo Flash s rovnakým ID, zobrazí sa zoznam týchto obvodov. Požadovaný obvod z tohto zoznamu vyberiete dvojklikom. Môžete tiež použiť výberové klávesy a voľbu potvrdiť stlačením **<Enter>** alebo kliknutím na tlačidlo **OK** .

Stlačením klávesy **<Esc>** alebo kliknutím na tlačidlo **Cancel** môžete kedykoľvek zrušiť výber bez vplyvu na aktuálne vybraný obvod.

Varovanie: Ovládací program podporuje momentálne automatický výber pre 28 a 32 pinové EPROM a Flash. Programátorom, ktoré nemajú schopnosť automaticky zistiť počet pinov obvodu, treba tento údaj zadať manuálne.

Programátor pripája na príslušné piny obvodu v päťici zvýšené napätie, ktoré je potrebné na povolenie čítania ID bajtov. Nevkladajte preto do päťice iné obvody ako EPROM alebo Flash, pretože zvýšené napätie na príslušných pinoch môže spôsobiť zničenie obvodu.

Neodporúča sa taktiež aplikovať tento spôsob výberu na EPROM typu 2764 a 27128, pretože väčšina z nich ID nepodporuje.



Obvod / Nastavenia súvisiace s obvodom

Tento príkaz ponúka celú škálu nastavení, ktoré riadia určité činnosti vykonávané na obvodoch. Jedná sa o riadenie programovacieho procesu, serializáciu a nastavenie súboru súvisiaceho s aktuálnym obvodom.

Obvod / Nastavenia súvisiace s obvodom / Operácie s obvodom

Všetky nastavenia, ktoré ponúka tento príkaz sú určené na riadenie programovacieho procesu. Je to variabilné prostredie, ktoré obsahuje nastaviteľné položky súvisiace s aktuálnym obvodom a typom programátora. Položky, ktoré sa pre daný obvod môžu uplatniť, ale nie sú podporované aktuálnym programátorom, sú zablokované. Toto nastavenie sa ukladá spolu so súvisiacim obvodom na disk príkazom **Súbor / Skončiť s uložením konfigurácie...**

Použité pojmy sú bližšie vysvetlené v užívateľskom manuáli k programátoru. Niektoré špeciálne termíny tu použité sa môžu vzťahovať ku konkrétnym obvodom. V takom prípade odporúčame prečítať dokumentáciu k obvodu poskytovanú od výrobcu obvodu.

Jedná sa o nasledovné položky (informácie v zátvorke sú prednastavené hodnoty): skupina **Adresy**:

počiatočná adresa obvodu (0)
koncová adresa obvodu (veľkosť obvodu-1)
počiatočná adresa buffera (0)
rozdelenie (0)

Nastavenie špeciálneho módu buffera, v ktorom sa používa modifikované adresovanie dát medzi bufferom a obvodom pri akciách programovania, čítania alebo verifikácie obvodu. Nastavenie rozdelenia má význam najmä pri použití pamäťových obvodov s 8-bitovou organizáciou dát v aplikáciách, kde je potrebná organizácia dát 16 alebo 32 bitová. Spôsob adresovania pre rôzne nastavenia voľby Rozdelenie je uvedený v nasledujúcej tabuľke:

| Rozdelenie | Obvod | Buffer |
|------------|-------------|--------------------|
| Nie je | Obvod[ADDR] | Buffer[ADDR] |
| Párne | Obvod[ADDR] | Buffer[2*ADDR] |
| Nepárne | Obvod[ADDR] | Buffer[1+(2*ADDR)] |
| 1./4 | Obvod[ADDR] | Buffer[4*ADDR] |
| 2./4 | Obvod[ADDR] | Buffer[1+(4*ADDR)] |
| 3./4 | Obvod[ADDR] | Buffer[2+(4*ADDR)] |
| 4./4 | Obvod[ADDR] | Buffer[3+(4*ADDR)] |

Príklad priradenia adres medzi obvodom a bufferom:

| Rozdelenie | adresa v obvode | adresa v bufferi |
|------------|-------------------------------|-------------------------------|
| Nie je | 00 01 02 03 04 05 06 07 08 09 | 00 01 02 03 04 05 06 07 08 09 |
| Párne | 00 01 02 03 04 05 06 07 08 09 | 00 02 04 06 08 0A 0C 0E 10 12 |
| Nepárne | 00 01 02 03 04 05 06 07 08 09 | 01 03 05 07 09 0B 0D 0F 11 13 |
| 1./4 | 00 01 02 03 04 05 06 07 08 09 | 00 04 08 0C 10 14 18 1C 20 24 |
| 2./4 | 00 01 02 03 04 05 06 07 08 09 | 01 05 09 0D 11 15 19 1D 21 25 |
| 3./4 | 00 01 02 03 04 05 06 07 08 09 | 02 06 0A 0E 12 16 1A 1E 22 26 |
| 4./4 | 00 01 02 03 04 05 06 07 08 09 | 03 07 0B 0F 13 17 1B 1F 23 27 |

Vysvetlenie pojmov:

Prístup k dátam obvodu na adrese ADDR je zapísaný ako Obvod[ADDR].

Prístup k dátam buffera na adrese ADDR je zapísaný ako Buffer[ADDR].
Hodnota ADDR môže nadobúdať hodnoty od nuly po veľkosť pamäte obvodu (v bajtoch).
Všetky adresy sú bajtovo orientované.

skupina **Test vloženia obvodu:**

kontrola prítomnosti obvodu v päťci (štandardne Povolený)

Ak je test povolený, tak programátor skontroluje všetky piny vloženého obvodu, či majú správny kontakt v ZIF päťci programátora. Programátor dokáže odhaliť zlé kontakty, posunuté vloženie obvodu a tiež u určitých obvodov aj otočenie obvodu opačne.

kontrola identifikačných údajov obvodu (štandardne Povolený)

Programátor pred každou operáciou s obvodom kontroluje ID kód obvodu definovaný výrobcom obvodu. V prípade chyby ID sa program správa nasledovne:

- ak je nastavenie Povolené, zvolená akcia sa ukončí
- ak je nastavenie Nepovolené, zvolená akcia pokračuje ďalej. Program iba vypíše do LOGu varovnú správu o chybe ID.

Ak je kontrola ID obvodu povolená, programátor kontroluje ID kód obvodu definovaný výrobcom obvodu.

Poznámky:

Niektoré obvody nemajú ID, preto kontrola ID nie je dostupná pre také obvody.

Pre niektoré obvody sa nemusí vykonať kontrola ID, aj keď je v ovládacom programe povolená. Týka sa to obvodov, ktoré v prípade nastavenia ochrany kopírovania nedovoľujú čítať ID obvodu.

skupina **Vykonanie príkazov:**

kontrola vymazania pred programovaním (štandardne Zakázaný)

vymazanie obvodu pred programovaním (štandardne Zakázaný)

verifikácia po prečítaní (štandardne Povolený)

verifikácia (štandardne Jedenkrát)

spôsob verifikácie

umožňuje zvoliť hodnoty hraničných napájacích napätí pre verifikáciu. Má význam len ak je nastavená Verifikácia Dvakrát

Poznámka: *Verifikáciu po naprogramovaní možno nastaviť na jedenkrát (pri nominálnom napájacom napätí) alebo dvakrát (podľa nastaveného spôsobu). Pri verifikácii na dvakrát sa jedná o kontrolu pri zníženom a zvýšenom napájacom napätí podľa zvolenia užívateľa alebo podľa odporúčania výrobcu programovateľného obvodu.*

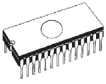
skupina **Parametre napájania cieľovej aplikácie:**

Táto skupina sa zobrazuje len pre niektoré obvody pri ISP programovaní a obsahuje nasledovné nastavenia:

Napájanie cieľovej aplikácie - povolí napájanie cieľového zariadenia z programátora. Napájacie napätie pre cieľové zariadenie je pripojené pred začiatkom akcie na programovanom obvode a je odpojené po ukončení akcie. V prípade ak je povolené Držať signály ISP na definovanej úrovni po akcii, programátor odpojí napájacie napätie aplikácie po deaktivovaní pull-up/pull-down rezistorov.

Napätie - veľkosť napájacieho napätia pre cieľové zariadenie. Napätie musí byť v rozsahu 2V až 6V.

Poznámka: *Hodnota napätia pre cieľový systém tiež závisí na pretekajúcom prúde. Na dosiahnutie presnej hodnoty napätia pre cieľový systém musí byť zadané požadované napätie a maximálny prúd. Hodnota maximálneho prúdu by mala čo najviac odpovedať reálnej spotrebe.*



Max. prúd - najvyšší povolený prúd do cieľového systému (z napájacieho napätia). Prúd musí byť z rozsahu 0 – 300mA.

Doba nábehu napätia - strmosť nábežnej hrany napájacieho napätia cieľovej aplikácie (pripojenie napätia).

Čas ustálenia napájania cieľovej aplikácie - čas, počas ktorého sa musí napájacie napätie cieľového zariadenia ustabilizovať na zadanej hodnote.

Doba dobehu napätia - strmosť spádovej hrany napájacieho napätia cieľovej aplikácie (odpojenie napätia).

Doba ustálenia po odpojení napájania - čas, počas ktorého môže cieľové zariadenie po odpojení napätia z programátora udržiavať zbytkové napájacie napätie (napr. z nabitých kondenzátorov). Po uplynutí tohto času nesmie cieľové zariadenie držať zbytkové napätie a môže byť bezpečne odpojené od programátora.

skupina **Parametre cieľovej aplikácie**

Táto skupina sa zobrazuje len pre niektoré obvody pri ISP programovaní a obsahuje nasledovné nastavenia:

Frekvencia oscilátora (v Hz) - frekvencia oscilátora obvodu (v cieľovom systéme). Riadiaci program nastavuje rýchlosť programovania podľa tohto parametra, preto je potrebné zadať jeho správnu hodnotu.

Napájacie napätie (v mV) - napájacie napätie v cieľovom systéme. Riadiaci program skontroluje alebo nastaví (závisí od typu programátora) zadané napájacie napätie v cieľovom systéme pred každou akciou na obvode.

Nepovoliť test napájacieho napätia - zakáže test napájacieho napätia, nastaveného v Napájacie napätie, programovaného obvodu pred akciou.

Oneskorenie po aktivácii resetu - tento parameter určuje oneskorenie po aktivácii signálu reset do začiatku akcie s obvodom. Oneskorenie závisí od súčiastok použitých v resetovacom obvode a môže byť zvolené z nasledujúcich hodnôt: 10ms, 50ms, 100ms, 500ms alebo 1s.

Držať signály ISP na definovanej úrovni po akcii - povolí držanie definovanej úrovne ISP signálov po ukončení činnosti s programovaným obvodom. Riadiaci program indikuje tento stav zobrazením okna s upozornením (že sú aktívne pull-up/pull-down rezistory). Po zatvorení tohto okna užívateľom program deaktivuje rezistory.

Kľudová úroveň všetkých ISP signálov - tento parameter určuje stav signálov ISP konektora v kľudovom stave (LED Busy nesvieti). Signály ISP konektora môžu byť nastavené buď do Pull-up (signály sú ťahané cez 22k rezistory do úrovne napájacieho napätia) alebo Pull-down (signály sú ťahané cez 22k rezistory do zeme).

skupina **Parametre programovania**

Táto skupina sa zobrazuje len pre niektoré obvody pri ISP programovaní a obsahuje nastavenia, ktorá časť obvodu má byť naprogramovaná, alebo iné špeciálne nastavenia.

skupina **Vymazanie parametre**

Táto skupina sa zobrazuje len pre niektoré obvody pri ISP programovaní a obsahuje špeciálne nastavenia pre módy vymazania vybratého obvodu.

Obvod / Nastavenia súvisiace s obvodom / Serializácia

Serializácia je špeciálny mód programu. Ak je aktívny, špecifikovaná hodnota je automaticky vložená na preddefinovanú adresu do buffera pred programovaním každého obvodu. Keď programujete viacero obvodov jeden za druhým, hodnota sériového čísla je automaticky zmenená pre každý obvod a pred programovaním obvodu vložená do buffera, takže každý obvod má unikátne sériové číslo.

Program poskytuje nasledovné typy serializácie:

- Pripočítavací mód
- mód Zo súboru
- mód Uživateľský generátor

Dialóg **Serializácia** taktiež obsahuje nastavenia pre priradený súbor na serializáciu, ktorý sa bude používať s projektovým súborom, keď bude zapnutá serializácia. Viac informácií nájdete v častiach Serializácia a Projekty.

Pri každom výbere nového obvodu sa serializácia nastaví do neaktívneho módu.

Aktuálne nastavenia serializácie pre aktuálny obvod sa ukladajú spolu so súvisiacim obvodom na disk príkazom **Súbor / Skončiť s uložením konfigurácie...**

Pri aktívnom **Pripočítavacom móde** sa ukladajú nasledujúce aktuálne nastavenia: adresa, veľkosť, sériové číslo, inkrementačný krok a nastavenia módov ASCII / BIN, DEC / HEX a LS Byte / MS Byte skôr.

Pri aktívnom móde **Zo súboru** sa ukladajú nasledujúce aktuálne nastavenia: meno vstupného súboru pre serializáciu a aktuálne návěstie (label) udávajúce riadok s aktuálnym sériovým číslom v súbore.

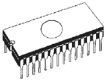
Ak je program v **multiprogramovom** móde (je vybratý programátor s násobnými modulmi), tak v dialógu Serializácia je zobrazená sekcia **Akcia s (kvôli chybe) nenaprogramovanými sériovými číslami**. Tu je možné nastaviť činnosť, ktorá sa má vykonať pre sériové čísla, ktoré neboli naprogramované (najčastejšie kvôli výskytu chyby pri programovaní). Na výber sú dve možnosti:

- Ignorovať nenaprogramované sériové čísla
- Pridať nenaprogramované sériové čísla do súboru

Ignorovať nenaprogramované sériové čísla znamená, že nenaprogramované sériové čísla sa ignorujú a nebude s nimi robená žiadna ďalšia akcia.

Pridať nenaprogramované sériové čísla do súboru znamená, že nenaprogramované sériové čísla sa zapíšu do súboru. Súbor nenaprogramovaných čísel má rovnaký formát ako pri serializácii **zo súboru**. Takto je možné naprogramovať tieto čísla neskôr použitím serializácie **zo súboru**.

Dialóg Serializácia obsahuje aj nastavenia ohľadom použitia serializácie s projektovými súborami (projektmi). Pre bližšie informácie o serializácii a projektoch, viď. osobitnú kapitolu **Serializácia a projekty**.



Poznámky: *Pokiaľ je programovanie prerušené užívateľom, sériové čísla sa nemenia a sú pripravené na naprogramovanie.*

Ignorovanie alebo zápis do súboru nenaprogramovaných sériových čísiel sa uplatní jedine, keď akcia programovania (aj s verifikáciou) prejde celá aspoň pre jeden obvod z danej skupiny obvodov v multipäťcovom module.

Serializácia môže fungovať s hlavným bufferom programu Pg4uw aj s rozšírenými buffermi dostupnými pre niektoré typy obvodov, napr. pre obvody Microchip PIC16Fxxx s pamäťou Data EEPROM. Výber buffera, do ktorého sa majú sériové čísla zapisovať, sa dá realizovať v dialógu Serializácia. Nastavenie buffera sa ignoruje pre serializáciu typu **From-file** playlist. Pre viac informácií viď. popis mód Zo súboru.

Pokiaľ si želáte, aby serializačné dáta boli vynechané z výpočtu kontrolnej sumy buffera, môžete použiť funkciu **Vynechať dáta z výpočtu kontrolnej sumy**, ktorá sa dá nastaviť v dialógu **Kontrolná suma**.

Obvod / Nastavenia súvisiace s obvodom / Serializácia / Pripočítavací mód

Ak je pri hromadnom programovaní potrebné individuálne pre každý programovaný obvod meniť sériové číslo, použite túto funkciu, ktorá zabezpečí inkrementáciu sériového čísla v bufferi vždy pred vloženíím nového obvodu. Pri inkrementačnom serializačnom móde je na začiatku užívateľom zadané sériové číslo, ktoré je zapísané do buffera pred každým programovaním obvodu. Po skončení každého programovania sa hodnota sériového čísla inkrementuje o špecifikovaný krok. Tým je zabezpečené zapísanie vždy nového aktuálneho sériového čísla do buffera pred každým programovaním.

K dispozícii sú nasledujúce nastavenia:

Veľk. sér. čísla

Veľk. sér. čísla určuje počet bajtov sériového čísla, t.j. jeho veľkosť. Pre Bin (binárny) mód je dovolená veľkosť v rozsahu 1 až 8 bajty, pre ASCII mód je dovolená veľkosť 1 až 16 bajtov.

Adresa

Adresa určuje adresu v bufferi, kde sa má zapísať sériové číslo. Pri zadávaní adresy treba zachovať opatrnosť a zadávať adresu vnútri intervalu **začiatok obvodu a koniec obvodu**. Adresa musí byť zadaná tak, aby aj posledný bajt sériového čísla, bol vnútri adresového intervalu začiatok obvodu a koniec obvodu.

Štart. hodnota

Štart. hodnota špecifikuje inicializačnú (prvú) hodnotu sériového čísla. Maximálna možná hodnota sériového čísla je \$1FFFFFFF zobrazených v 32 bitovom slove. V prípade, že sa pri inkrementácii prekročí maximálna dovolená hodnota, tri najvyššie bity sériového čísla sa nastavujú na 0, čím sa zabezpečí aby číslo bolo opäť v rozsahu 0 až \$1FFFFFFF (ide teda o ošetrenie pretečenia).

Krok

Krok udáva inkrementačný krok, o ktorý sa má zvyšovať sériové číslo pri ukončení programovania obvodu. Tým sa pripraví hodnota sériového čísla pre nasledujúci obvod.

Tvar sér. čísla

Tvar sér. čísla definuje tvar, akým sa sériové číslo bude zapisovať do buffera. Sú k dispozícii dve možnosti nastavenia:

- ASCII
- Bin

ASCII - sériové číslo bude zapísané do buffera v tvare ASCII reťazca. Napríklad číslo \$0528CD je v ASCII serializačnom móde zapísané do buffera v tvare 30h 35h 32h 38h 43h 44h ('0' '5' '2' '8' 'C' 'D'), t.j. na šiestich bajtoch.

Bin - znamená, že sériové číslo je zapísané priamo tak ako je v bufferi. Ak sériové číslo má viac ako jeden bajt, môže byť do buffera zapísané dvoma spôsobmi s ohľadom na najvyšší a najnižší bajt (viď. nastavenie **Uloženie v bufferi**).

Štýl

Štýl určuje číselnú sústavu, v ktorej je sériové číslo reprezentované. Sú dve možnosti:

- Decimal
- Hexadecimal.

Decimal (desiatková) - sú použité len desiatkové číslice 0 až 9.

Hexadecimal (šestnástková) - sú použité aj šestnástkové číslice A až F.

Špeciálny prípad je S/N mód nastavený na „Bin“ a Štýl na „Dec“, čo znamená BCD číselné kódovanie. BCD je spôsob zobrazenia desiatkových čísel v číslach šestnástkových tak, že pre štvorbity udávajúce jednotlivé číslice šestnástkového čísla platí, že môžu mať len hodnoty od 0 do 9 (nesmú mať hodnoty od A po F).

Uloženie v bufferi

Uloženie v bufferi umožňuje definovať spôsob, akým sa bude zapisovať sériové číslo do buffera z hľadiska poradia jednotlivých bajtov sériového čísla. Nastavenia v „Uloženie v bufferi“ majú opodstatnenie len pre binárny (Bin) tvar sér. čísla. Sú dve možnosti nastavenia:

- Najmenej významný - je používaný procesormi firmy Intel a znamená, že najmenej významný (najnižší) bajt čísla bude umiestnený na najnižšej adrese v bufferi.
- Najvýznamnejší - je používaný procesormi firmy Motorola a znamená, že najviac významný (najvyšší) bajt čísla bude umiestnený na najnižšej adrese v bufferi.

Rozdeliť sériové číslo na každý N bajt

Voľba umožňuje aktivovať rozdelenie sériového čísla na bajty a zápis sériového čísla do obvodu s preskakovaním buniek po N bajtoch. Význam to môže mať napríklad u obvodov Microchip PIC, kde sa dá realizovať identifikácia obvodu cez sériové číslo uložené v programovej pamäti v rámci inštrukcií RETLW. Príklad použitia je uvedený v časti Príklady ako druhý príklad.

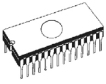
Príklady:

1. Príklad

Máme obvod AT29C040 a chceme na pozícii 7FFFAH ukladať sériové čísla s veľkosťou 4 bajty. Požadujeme štartovaciu hodnotu 1600000H, pripočítavací krok 1, binárny tvar čísla a najnižší bajt čísla do nižšej adresy v obvode.

Pre realizáciu požadovanej serializácie sú nastavenia serializácie v programe nasledovné:

Mód serializácie: Pripočítavací
Veľkosť sériového čísla: 4 bajty
Tvar sériového čísla: Bin



Štýl: Hex
Uloženie v bufferi: Najmenej významný
Adresa: 7FFFCH
Štartovacia hodnota: 16000000H
Krok: 1

Potom sa do obvodu budú zapisovať nasledovné hodnoty (uvádza sa ako výpis obsahu buffera po načítaní obvodu naprogramovaného užívateľskými dátami a serializáciou generovaným sériovým číslom):

1. obvod

Adresa Data

007FFF0 xx xx xx xx xx xx xx xx xx xx 00 00 00 16

2.obvod

Adresa Data

007FFF0 xx xx xx xx xx xx xx xx xx xx 01 00 00 16

3. obvod

Adresa Data

007FFF0 xx xx xx xx xx xx xx xx xx xx 02 00 00 16

atď.

Označenie "xx" sú užívateľské dáta napaľované do obvodu.

Od adresy 7FFFCH po 7FFFFH sú do obvodu zapisované 4 bytové sériové čísla.

2. Príklad

Nasledujúci príklad ukazuje využitie možnosti rozdelenia sériového čísla podľa SQTP serializácie a využitie rozdelenia v rámci RETLW inštrukcií pre obvod Microchip PIC16F628.

Poznámka: *Serial quick turn programming (SQTP) je štandard serializácie špecifikovaný firmou Microchip pre obvody Microchip PIC. Obvody Microchip PIC umožňujú programovať jedinečné sériové číslo pre každý obvod. Toto číslo môže byť použité ako vstupný kód, heslo alebo ako identifikačné číslo.*

SQTP serializácia je realizovaná použitím série inštrukcií RETLW (Return Literal W) s bajtovým sériovým číslom zadaným ako dátový priamy parameter inštrukcie RETLW. Na SQTP serializáciu je možné v ovládacom programe použiť dva druhy serializácie - Pripočítavací mód alebo mód Zo súboru.

Serializácia cez Pripočítavací mód ponúka možnosť rozdelenia sériového čísla na jednotlivé bajty, pričom medzi bajtami sériového čísla sa dodá nejaká konštantná hodnota, napr. kód inštrukcie RETLW.

Serializácia zo súboru používa externý súbor obsahujúci zoznam sériových čísel. Súbor môže obsahovať ľubovoľné sériové čísla, čiže aj sériové čísla v tvare vhodnom pre SQTP serializáciu. Upozornenie: Serializačný súbor má iný formát ako súbor SQTP vygenerovaný programom Microchip MPLAB.

Obvod PIC16F628 má 14 bitovú šírku slova. Inštrukcia RETLW má tvar:

| | | | |
|--------------------------------|-----|-----------------|------|
| Popis | MSB | 14-Bitové slovo | LSB |
| RETLW return with literal in W | 11 | 01xx kkkk | kkkk |

kde xx sa nahradí nulami a "K" sú bity dátové, t.j. bity sériového čísla

kód inštrukcie RETLW teda bude 34KKH kde KK je dátový bajt

Teraz uvažujme, že chceme zapísať sériové číslo s veľkosťou 4 bajty a hodnotou 1234ABCDH. Nech je najvyšší bajt najvýznamnejší. Číslo chceme zapísať do obvodu na adresu 40H a chceme ho rozdeliť do jednotlivých 14 bitových inštrukcií RETLW obvodu PIC. Na rozdelenie čísla je vhodné práve použiť funkciu "Rozdeliť sériové číslo každý N bajt". Štandardne sa totiž pri serializácii číslo 1234ABCDH zapíše do buffera v tvare

```
Adresa  Data
0000080  CD AB 34 12  xx xx xx xx xx xx xx xx xx xx
```

pozn.: *adresa 80H je preto, že adresa 40H je uvažovaná ako wordová, preto v bajtovej organizácii v*

V bufferi v režime view16 to bude mať tvar

```
Adresa  Data
0000040  ABCD 1234 xxxx xxxx xxxx xxxx xxxx
```

My však chceme dostať do buffera a do obvodu data v tvare RETLW inštrukcií nasledovne:

```
Adresa  Data
0000040  34CD 34AB 3434 3412 xxxx xxxx xxxx xxxx
```

to docielime nasledovným postupom:

- do buffera ručne alebo načítaním súboru načítame na prvé štyri adresy dáta v tvare inštrukcie RETLW s tým, že na spodných 8 bitoch nezáleží lebo tie bude dopĺňať serializácia. Obsah buffera môže potom vyzeráť napr. takto:

```
Adresa  Data
0000040  3400 3400 3400 3400 xxxx xxxx xxxx xxxx
```

Pre každú inštrukciu RETLW som zadal spodných 8 bitov ako nuly.

- Nastavenia serializácie nastavíme nasledovne:

Veľk. sér. čísla 4 bajty

Adresa: 40H

Štart hodnota: 1234ABCDH

Krok: 1

Tvar sér. čísla: BIN

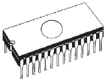
Štýl: HEX

Uloženie v bufferi: Najmenej význ.

Zaškrtnutá voľba "Rozdeliť sériové číslo na každý N bajt" a nastavená hodnota 2, t.j. "Rozdeliť sériové číslo na každý 2 bajt"

Pri programovaní obvodu sa nastaví tesne pred akciou programovania obvodu do buffera sériové číslo nasledovne:

```
Adresa  Data
0000040  34CD 34AB 3434 3412 xxxx xxxx xxxx xxxx
```



To je presne, čo chceme.

3. Príklad

Nasledujúci príklad používa tie isté nastavenia serializácie, ako príklad 2., okrem hodnoty rozdelenia sériového čísla, ktoré je tu nastavené na hodnoty 3 a 4.

Ak je nastavená voľba "Rozdeliť sériové číslo na každý 3 bajt", tak obsah buffera bude nasledovný:

Bajtová organizácia buffera:

Adresa Data

0000080 CD xx xx AB xx xx 34 xx xx 12 xx xx xx xx xx xx

Slovná organizácia buffera (x16):

Adresa Data

0000040 xxCD ABxx xxxx xx34 12xx xxxx xxxx xxxx

Ak je nastavená voľba "Rozdeliť sériové číslo na každý 4 bajt", tak obsah buffera bude nasledovný:

Bajtová organizácia buffera:

Adresa Data

0000080 CD xx xx xx AB xx xx xx 34 xx xx 12

Slovná organizácia buffera (x16):

Adresa Data

0000040 xxCD xxxx xxAB xxxx xx34 xxxx xx12 xxxx

Poznámka:

Ak si nie ste istý aký efekt majú nastavenia serializácie, ktoré nastavíte, je možné overiť, čo sa naozaj do buffera zapíše, nasledovným spôsobom:

- 1. zvolte požadované nastavenia serializácie v dialógu Serializácia a potvrdte ich*
- 2. zakážte v dialógu Operácie s obvodom položky "Test vloženia obvodu" a "Kontrola ID obvodu" (ak je taká položka zobrazená)*
- 3. skontrolujte, že v päťici programátora nie je vložený obvod*
- 4. spustíte akciu programovať obvod (u niektorých obvodov je pred samotným programovaním potrebné nastaviť oblasti obvodu, ktoré sa majú programovať)*
- 5. po skončení akcie (pravdepodobne s chybou, čo však nevadí) sa pozrite na obsah buffera (Pozerať/editovať buffer) a na adresu buffera, na ktorej má byť sériové číslo umiestnené*

Upozornenie:

Adresa sa vzťahuje vždy k aktuálnej organizácii obvodu, resp. buffera v programe. Pri organizácii bajtovej (x8) je adresa zadávaná v bajtoch a pri organizácii väčšej ako x8, napr. x16, je adresa zadávaná v slovách (wordoch).

Obvod / Nastavenia súvisiace s obvodom / Serializácia / Klasický mód zo súboru

Pri použití serializačnej metódy **Klasický mód zo súboru** serializačný súbor obsahuje priamo sériové čísla aj s adresami, na ktoré sa majú sériové čísla zapísať do buffera. Serializačné dáta sú teda čítané priamo zo serializačného súboru. Klasický mód zo

súboru je indikovaný v hlavnom okne programu Pg4uw a v info okne počas akcie s obvodom v tvare "Serialization: From-file" alebo "Serializácia: Zo súboru". Sú dve nastavenia ku Klasickému módu zo súboru:

Štart. návestie

Štart. návestie definuje návestie vo vstupnom súbore. Čítanie sériových čísel zo súboru začne od návestia definovaného v položke Štart. návestie.

Názov súboru

Názov súboru špecifikuje meno vstupného súboru, z ktorého sa majú čítať sériové čísla. Súbor pre Klasický mód zo súboru je v textovom tvare a musí mať správny formát.

Formát súboru

Vstupný súbor serializácie v Klasickom móde zo súboru má definovaný textový formát, v ktorom sú definované polia bajtov udávajúce dáta určené na zápis do buffera. Súbor pre Klasický mód zo súboru si teda vytvára sám používateľ.

Formát súboru je nasledovný:

```
[label1] addr byte0 byte1 .. byten  
...  
[labeln] addr byte0 byte1 . bytem, addr byte0 byte1 ... bytek
```

└──────────┬──────────┘ └──────────┬──────────┘
 | |
základná časť nepovinná časť

; Komentár

význam je nasledovný:

základná časť

Základná časť obsahuje adresu a pole bajtov, ktoré sa má zapísať do buffera. Základná časť sa musí povinne zadávať za návestím v riadku.

nepovinná časť

Nepovinná časť umožňuje definovať v príslušnom riadku druhú adresu a pole údajov na zápis do buffera.

label1, labeln - návestia

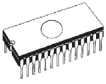
Návestia sú identifikátormi každého riadku v súbore. Používajú sa na adresovanie jednotlivých riadkov súboru, preto by sa mená návestí nemali v súbore opakovať. Adresovanie riadkov v súbore znamená napr., že čítanie sériových čísel začne od riadku s užívateľom zadaným štartovacím návestím.

addr

Addr definuje adresu v bufferi, na ktorú sa majú zapísať dáta nasledujúce za adresou.

byte0..byten, byte0..bytem, byte0..bytek

Polia bajtov byte0..byten, byte0..bytem a byte0..bytek definujú samotné dáta, ktoré sa majú zapísať do buffer. Max. počet bajtov pre jedno pole údajov nasledujúce za adresou je 64 bajtov. Bajty sú do buffera zapisované postupne od adresy addr nasledovne:



byte0 na addr
byte1 na addr + 1
byte2 na addr + 2
....
byten na addr + n

Nepovinná časť sa zadáva za základnou časťou, pričom je od základnej časti oddelená čiarkou a jej štruktúra je rovnaká ako u prvej časti, t.j. adresa a za ňou nasledujúce pole bajtov.

Znaky so špeciálnym významom:

[] - návestia musia byť definované v hranatých zátvorkách

'-' – čiarka slúži na oddelenie poľa údajov základnej časti a nep povinnej časti

;' – bodkočiarka definuje začiatok komentára v riadku, t.j. všetky znaky v riadku počnúc bodkočiarkou až do konca riadku sú ignorované a prechádza sa na nasledujúci riadok. Komentár môže byť na samostatnom riadku alebo nasledovať za dátovými hodnotami.

Poznámky:

- *Mená návestí môžu obsahovať ľubovoľné znaky okrem znakov "[“ a ”]“. Nerozlišujú sa veľké a malé písmená v návestiach.*
- *Všetky adresy a bajtové hodnoty sú zadávané ako šestnástkové čísla.*
- *Povolená veľkosť adresy v bajtoch je 1 až 4.*
- *Povolená veľkosť dátových polí je 1 až 64 bajtov, pričom ak sú v jednom riadku zadane dve dátové polia súčet ich veľkostí v bajtoch môže byť maximálne 80 bajtov.*
- *Je potrebné zachovať opatrnosť pri zadávaní adries. Adresy musia byť definované vnútri intervalu začiatok obvodu a koniec obvodu. V prípade, že ktorákoľvek adresa je mimo rozsahu, zobrazí sa okno upozorňujúce na tento stav a serializácia sa prepne do neaktívneho módu (mód Neaktívne).*
- *Adresa sa vzťahuje vždy k aktuálnej organizácii obvodu, resp. buffera v programe. Pri organizácii buffera v programe bajtovej (x8) je adresa zadávaná v bajtoch a pri organizácii väčšej ako x8, napr. x16, je adresa zadávaná v slovách (wordoch).*

Príklad typického vstupného súboru pre Klasický mód zo súboru:

[nav1] A7890 78 89 56 02 AB CD ; komentár1

[nav2] A7890 02 02 04 06 08 0A

[nav3] A7890 08 09 0A 0B A0 C0 ; komentár2

[nav4] A7890 68 87 50 02 0B 8D

[nav5] A7890 A8 88 59 02 AB 7D

;nasledujúci riadok obsahuje aj druhú definíciu

[nav6] A7890 18 29 36 42 5B 6D , FFFF6 44 11 22 33 99 88 77 66 55 16

;toto je komentár : posledný riadok – koniec súboru

V príklade je definovaných šesť sériových čísel s návestiami „nav1“, „nav2“, ...“nav6“. Každé z čísel je zapísané do buffera na adresu \$A7890. Všetky hodnoty majú veľkosť

šesť bajtov. Riadok s návěstím „nav6“ obsahuje navyše definíciu druhej hodnoty, ktorá môže znamenať napríklad CRC súčet a je zapísaná do buffera na adresu \$FFFF6. Veľkosť hodnoty je 10 bajtov, čiže posledný bajt hodnoty bude zapísaný do buffera na adresu \$FFFFFF.

Upozornenie:

Adresy zadávané v súbore pre serializáciu "Zo súboru" sa vzťahujú vždy k aktuálnej organizácii obvodu, resp. buffera v programe. Pri organizácii bajtovej (x8) je adresa zadávaná v bajtoch a pri organizácii väčšej ako x8, napr. x16, je adresa zadávaná v slovách (wordoch).

Obvod / Nastavenia súvisiace s obvodom / Serializácia / Playlist mód zo súboru

Pri použití serializačnej metódy **Playlist mód zo súboru** serializačný súbor neobsahuje priamo sériové čísla, ale zoznam súborov, ktoré obsahujú jednotlivé sériové čísla. Serializačné dáta sú teda načítavané až z externých súborov uvedených v serializačnom súbore. Každý externý súbor reprezentuje jedno sériové číslo (jeden naprogramovaný obvod). Playlist mód zo súboru je indikovaný v hlavnom okne programu Pg4uw a v info okne počas akcie s obvodom v tvare "*Serialization: From-file-pl*" alebo "*Serializácia: Zo súboru-pl*".

Formát súboru

Serializačný súbor playlist serializácie obsahuje zoznam mien súborov, ktoré obsahujú serializačné dáta. Formát playlist súboru je podobný ako u klasickej serializácie zo súboru. U playlist serializačných súborov sú nasledovné rozdiely:

1. playlist súbor musí mať ako prvý riadok uvedenú nasledovnú hlavičku
FILETYPE=Pg4uw SERIALIZATION PLAYLIST FILE
2. každý záznam serializačnej hodnoty je reprezentovaný na samostatnom riadku vo formáte
[label x] datafilename

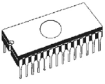
label/x - reprezentuje návestie príslušnej serializačnej hodnoty

Návestia sú teda jedinečné identifikátory každej serializačnej hodnoty. Slúžia na jednoznačnú identifikáciu každého serializačného záznamu. Vďaka návestiam môže užívateľ nastaviť štartovaciu pozíciu serializácie v serializačnom súbore.

datafilename - definuje meno dátového súboru, ktorý obsahuje serializačné dáta. Ak systém serializácie dá požiadavku na načítanie sériového čísla, resp. serializačného záznamu, tak sa príslušný dátový súbor načíta do buffera pomocou štandardnej funkcie "Čítať súbor". Formát dátového serializačného súboru môže byť binárny alebo Hex formát (napr. IntelHex). Systém rozpoznávania formátu súborov nastaví správny formát pre načítanie dát zo súboru automaticky. Meno dátového súboru je zadávané vrátane cesty k súboru, pričom cesta je vzťahovaná relatívne k hlavnému (playlist) serializačnému súboru.

Príklad serializačného súboru pre Playlist mód zo súboru:

```
;---- following file header is required -----  
FILETYPE=PG4UW SERIALIZATION PLAYLIST FILE
```



;----- references to serialization data files

[nav1] file1.dat

[nav2] file2.dat

[nav3] file3.dat

...

[label n] filex.dat

;----- end of file -----

Detailnejší a komplexnejší príklad serializácie typu Playlist mód zo súboru, viď. príklad reálnych serializačných súborov v podadresári "Examples" umiestnenom v adresári, kde je inštalovaný program Pg4uw. Súboru sú zbalené v .zip archíve fromfileplaylistexample.zip. Dajte prosím dôraz na to, aby bola zachovaná štruktúra adresárov tak, ako je v .zip archíve.

Uvedené serializačné súbory môžete otestovať v reálnej situácii nasledovnými krokmi:

- 1.rozbalte súbory z archívu fromfileplaylistexample.zip do ľubovoľného vhodného adresára
- 2.spustíte PG4UW
- 3.potrebuje mať pripojený a úspešne vyhľadán programátor v Pg4uw
- 4.vyberte požadovaný obvod, na ktorom si želáte vyskúšať serializáciu (najlepšie na skúšanie sú obvody s mazateľnou pamäťou - FLASH, EEPROM, a nie OTP obvody, napr. (EPROM))
- 5.otvorte dialóg Serializácia dostupný cez menu **Obvod / Nastavenia súvisiace s obvodom / Serializácia**
- 6.nastavte mód "Zo súboru" a v paneli "Nastavenia módu zo súboru" zadajte playlist serializačný súbor fromfile_playlist.ser
- 7.stlačte tlačidlo OK na potvrdenie nových nastavení serializácie
- 8.spustíte operáciu "Programovať" obvod

Môžete si pozrieť indikátory serializácie v hlavnom okne Pg4uw v paneli Adresy a taktiež počas akcie programovania na obvode v okne Info.

Dodatočná akcia s použitými súbormi

V tejto skupine si môžete vybrať jednu z troch volieb, čo urobiť s použitými súbormi.

- **Nerobiť nič** – program neurobí žiadnu operáciu s použitými súbormi.
- **Presunúť použitý súbor do zvoleného adresára** – program presunie použité súbory do užívateľom zvoleného adresára.
- **Zmazať použitý súbor** – program zmaže použité súbory

Obvod / Nastavenia súvisiace s obvodom / Serializácia / mód Užívateľský generátor

Mód serializácie Užívateľský generátor poskytuje maximálnu flexibilitu serializácie, pretože užívateľ má celú serializáciu vo vlastných rukách.

Keď je vybraný serializačný mód Užívateľský generátor, sériové čísla sú generované užívateľom napísaným programom „on the fly“ pred naprogramovaním každého obvodu. Tento mód umožňuje užívateľovi generovať jedinečné sekvencie potrebných sériových čísel. Sériové čísla môžu byť zvyšované ako lineárne sekvencie alebo úplne nelineárne sekvencie.

Podrobnosti o užívateľom napísanom programe na generovanie sériových čísel budú opísané v časti **Užívateľský generátor – program**.

Príklad:

Súbory s príkladmi serializácie `examples.exe` a C/C++ zdrojový kód sa nachádzajú v adresári, kde sa inštaluje Pg4uw v podadresári `Examples`.

```
<Pg4uw_inst_dir>\Examples\Serialization\customgenerator_example\
```

Pri štandardnom umiestnení Pg4uw:

```
C:\Program Files\Elnec_sw\Programmer\Examples\Serialization\customgenerator_example\
```

V dialógu Serializácia zvolíte Užívateľský generátor. Zobrazia sa nasledovné možnosti:

Serializačný dátový súbor

Určuje cestu a meno súboru, ktorý bude obsahovať aktuálne sériové číslo. Pred programovaním obvodu Pg4uw zavolá užívateľský generátor, ktorý aktualizuje dátový súbor. Doporučená prípona dátového súboru je `.dat`.

Pretože veľa našich zákazníkov, ktorí tiež používajú programátory BP Microsystem nás žiadalo o možnosť používať rovnaký serializačný software, je dátový súbor kompatibilný s `.dat` súbormi „Complex serialization“ systému používaného v BP Micro software.

Poznámka: *Dátový súbor je úplne a pravidelne prepisovaný počas programovania s týmto typom serializácie. Uistite sa, že ste zadali správne meno `.dat` súboru, ktorý chcete používať. Napríklad `“c:\serial_files\serial.dat”`.*

Serializačný generátor

Určuje meno a cestu ku spustiteľnému súboru, ktorý bude generovať serializačný dátový súbor.

Prvé sériové číslo

Táto voľba je potrebná k určeniu počiatočného sériového čísla, ktoré bude vložené do užívateľského generátora. Číslo sa vkladá a zobrazuje v hexadecimálnom tvare.

Posledné sériové číslo

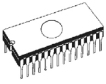
Táto voľba určuje maximálnu hodnotu povolených sériových čísel. Ak je to nenulová hodnota, bude vložená do užívateľského generátora. Užívateľský generátor testuje hodnotu posledného sériového čísla a generuje dátový súbor s odpovedajúcou chybovou informáciou v serializačnom dátovom súbore v prípade, že aktuálne sériové číslo je väčšie ako zadané posledné sériové číslo. Ak je hodnota posledného sériového čísla nula, hodnota nebude použitá.

Zaškrtnuté políčko **Volat' generátor s parametrom -RESULT po dobehnutí akcie na obvode**

Táto nová voľba má špeciálny zmysel. Ak existuje požiadavka na volanie serializačného generátora so špeciálnym parametrom `-RESULT`, tak by ste mali zaškrtnúť toto políčko, pretože štandardne je nezaškrtnuté. Ak je toto políčko zaškrtnuté užívateľský generátor je volaný programom Pg4uw po ukončení každej operácie na obvode bez ohľadu na výsledok OK alebo Error. Parametre pre generátor sú tvorené serializáciou v programe Pg4uw. Sú možné dva parametre:

`-RESULT[n]=TRUE | FALSE`

kde *n* je voliteľný – číslo programovacieho site v prípade multiprogramovania.



TRUE znamená, že akcia s obvodom skončila OK. FALSE znamená, že akcia s obvodom skončila s chybou.

-N<sériové číslo>

Určuje aktuálne sériové číslo rovnakým spôsobom ako pri normálnom volaní serializačného generátora.

Užívateľský generátor - program

Užívateľský generátor – program alebo užívateľský generátor je program, ktorý bude generovať jedinečnú postupnosť sériových čísel a zapisovať ich do serializačného dátového súboru. Tento program si vytvára užívateľ. Meno a cesta ku tomuto programu musí byť zadaná v dialógu Nastavenia serializácie, v časti Užívateľský generátor.

Tento program bude volaný z Pg4uw vždy keď sa bude generovať nové sériové číslo, t.j. obvyčajne pred programovaním obvodu. Pg4uw vloží do príkazového riadku parametre pre serializačný program a ten vygeneruje serializačný dátový súbor, ktorý bude Pg4uw čítať. V príkazovom riadku sa používajú nasledovné parametre:

-N<sériové číslo> určuje aktuálne sériové číslo.

-E<sériové číslo> určuje posledné sériové číslo. Tento parameter sa použije iba vtedy, keď hodnota Posledného sériového čísla určeného v dialógu Nastavenia serializácie je nenulová. Serializačný program vráti chybový záznam T06 v serializačnom dátovom súbore, ak aktuálne sériové číslo je väčšie ako posledné sériové číslo. Pre viac informácií pozrite časť formát serializačného dátového súboru.

Formát serializačného dátového súboru

Serializačný dátový súbor generovaný serializačným generátorom musí dodržiavať nasledovný formát.

Serializačný dátový súbor sa skladá z časti záznamov a z časti sériových čísel.

Záznam je riadok, ktorý začína reťazcom Txx, ako je to opísané nižšie. Hodnota „xx“ predstavuje kód záznamu. Záznamy sa používajú na informovanie Pg4uw o stave serializácie (aktuálne a posledné sériové číslo, serializačné dáta a ich formát, chyby atď.) Nutnými záznamami sú záznamy T01, T02, T03 a T04. Ostatné záznamy sú voliteľné.

T01:< sériové číslo > Obsahuje aktuálne sériové číslo vložené do generátora z príkazového riadku *-N< sériové číslo >*.

T02:< sériové číslo > Obsahuje nasledovné sériové číslo, ktoré Pg4uw použije v nasledovnom serializačnom cykle. Táto hodnota je generovaná serializačným generátorom a informuje Pg4uw, aké sériové číslo bude nasledovať za aktuálnym.

T03:<kód formátu dát> Určuje formát serializačných dát. Nasledovné formáty sú teraz podporované:

T03:50 alebo T03:55 ASCII Space dátový formát

T03:99 - Intel Hex dátový formát

T04: označuje, že serializačné dáta budú nasledovať od nasledujúceho riadku do konca súboru. Serializačné dáta sú uložené v jednom zo štandardných ASCII dátových formátov, napr. Intel HEX, ASCII Space atď. Použitý dátový formát musí byť určený záznamom T03.

Príklad: Typický serializačný dátový súbor:

T01:000005
T02:001006
T03:99
T04:
:0300000000096B89
:03000300000005F5
:02000C005A0197
:01003F004F71
:0000001FF

Súbor obsahuje nasledovné informácie:

riadok T01 – aktuálne sériové číslo je 000005h

riadok T02 – koncové sériové číslo je 001006h

riadok T03 – serializačné dáta za riadkom T04 sú vo formáte Intel Hex

riadok T04 – serializačné dáta, ktoré budú načítané do buffera Pg4uw pred programovaním obvodu, dáta sú Intel HEX formáte

Nepovinné záznamy sú:

T05:<správa> Varovné alebo chybové hlásenie. Tento záznam reprezentuje zastavenie serializácie a zároveň sa v Pg4uw zobrazí varovné alebo chybové hlásenie.

T06: Aktuálne sériové číslo je väčšie ako limit.
Tento záznam reprezentuje zastavenie serializácie a zobrazenie varovného alebo chybového hlásenia v Pg4uw, pretože aktuálne sériové číslo je väčšie ako povolené maximálne sériové číslo. Tento záznam sa používa keď je použitý -E parameter príkazového riadku, t.j. je zadaná nenulová hodnota posledného sériového čísla v dialógu Nastavenia serializácie.

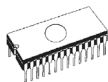
T11:<správa> Menej dôležité upozornenia a správy. Serializácia nebude prerušená.

Typický priebeh programovania obvodov je nasledovný:

1. Štart programovacej postupnosti
2. Insertion test obvodu
3. Serializačná sekvencia pozostáva zo 4 krokov:
 - Zavolanie serializačného generátora so správnym parametrom príkazového riadka na vygenerovanie serializačného .dat súboru
 - Čakanie, kým bude serializačný .dat súbor k dispozícii
 - Načítanie dát zo serializačného .dat súboru do buffera programátora (dáta budú použité pre programovanie obvodu)
 - Vymazanie serializačného .dat súboru po prečítaní dát z neho
4. Programovanie obvodu
5. Verifikácia obvodu
6. Kontrola výsledku operácie.

Táto je plne riadená programom Pg4uw. Serializačný program nemôže robiť nič podľa výsledku operácie. Ovládací program zavolá serializačný generátor požadovaným parametrom príkazového riadku.

OK - Pg4uw pošle požiadavku na ďalšie sériové číslo. Ďalšie sériové číslo bude prečítané z .dat súboru ako v kroku 3. Volanie serializačného generátora prinesie ďalšie sériové číslo špecifikované v príkazovom riadku.



ERROR - Pg4uw nedá žiadosť na nové sériové číslo. Posledné sériové číslo bude použité pre nasledujúci obvod. Nasledujúce volanie serializačného generátora prinesie posledné sériové číslo špecifikované v príkazovom riadku.

7. Zopakovať programovanie s ďalším obvodom?

Áno chod' na krok 2.

Nie chod' na krok 8.

8. Koniec programovacej postupnosti

Poznámky:

V prípade chyby pri programovaní, posledné sériové číslo je použité, ale generátor bude zavolaný ako v kroku 3 aj napriek tomu, že rovnaké sériové číslo je použité pre predchádzajúci obvod.

Ak sa zistí chyba serializačného .dat súboru, program Pg4uw hlási serializačnú chybu a hneď zastaví programovacie postupnosť.

Obvod / Nastavenia súvisiace s obvodom / Štatistika

Štatistika slúži na informáciu o aktuálnom počte operácií, ktoré boli vykonané na obvodoch práve vybraného typu. Ak jednému obvodu bude zodpovedať jedna operácia, napr. programovanie, tak počet operácií bude rovný počtu naprogramovaných obvodov.

Ďalšou funkciou štatistiky je možnosť nastaviť počet obvodov, na ktorých sa má vykonať určitá operácia, pričom po každom úspešnom skončení operácie na obvode sa dekrementuje tzv. **Počítadlo nadol** alebo počítadlo, ktoré na začiatku obsahuje užívateľom zadaný počet obvodov. Keď Počítadlo nadol dosiahne nulu, znamená to, že boli vykonané operácie na príslušnom počte obvodov a užívateľovi sa o tom vypíše informácia. Zároveň sa už nezobrazí otázka na opakovanie operácie.

Dialóg **Štatistika** obsahuje nasledujúce položky:

Zaškrtávacie políčka **Programovať**, **Verifikovať**, **Kontr. Vymaz.**, **Vymazať** a **Čítať** slúžia na určenie akcií, po ktorých sa majú inkrementovať hodnoty štatistiky.

Akákoľvek vybratá a uskutočnená operácia s obvodom bude inkrementovať počítadlo **Celkovo** a jedno z počítadiel **Úspešne** alebo **Chybne** v závislosti na výsledku operácie. Kombinácia čiastkových operácií sa berie ako jedna operácia. Napríklad: operácia Čítanie obsahuje aj operáciu Verifikácia po čítaní, ale berie sa to ako jedna operácia. Operácia Programovanie obsahuje Mazanie a/alebo Verifikácia a tiež je to len jedna operácia.

Zaškrtávacie políčko **Počítadlo nadol** určuje, či je odpočítavanie aktívne alebo neaktívne. Do editačného okienka nasledujúce za položkou Počítadlo nadol sa udáva počiatočný počet obvodov. Od tejto hodnoty začne odpočítavanie.

Ak sa ukazovateľ myši nachádza v paneli **Štatistika**, je možné tento dialóg otvoriť aj stlačením pravého tlačidla myši a potvrdením voľby Štatistika.

Aktuálny stav štatistiky sa zobrazuje v hlavnom okne ovládacieho programu v paneli Štatistika.

Panel Štatistika obsahuje tri hodnoty štatistických veličín: **Úspechy**, **Chyby**, **Celkovo** a dve hodnoty informujúce o stave odpočítavania **Odpočítavanie**: a **Zostáva**.

Ich význam je nasledovný:

| | |
|----------------------|--|
| Úspechy | udáva počet úspešne vykonaných operácií |
| Chyby | udáva počet neúspešne vykonaných operácií |
| Celkovo | udáva celkový počet operácií. |
| Odpočítavanie | informuje, či je odpočítavanie aktívne alebo neaktívne |
| Zostáva | informuje, koľko operácií s obvodom sa má ešte vykonať |

Poznámka: Pri výbere nového typu obvodu sa všetky ukazovatele štatistiky vynulujú a **Odpočítavanie** sa nastaví na **Neaktívne**.

Tlačidlo **Reset** v paneli **Štatistika** slúži na vynulovanie indikátorov štatistiky.

Tlačidlo **Naplniť odpočítavanie** v paneli **Štatistika** slúži na nastavenie počítadla na počiatočný počet obvodov.

Pri ukončení programu Pg4uw, tento ukladá informácie o štatistike do Log okna (a aj do Log súboru, keď je táto možnosť zvolená).

Pre multiprogramovanie pri ukončení programu Pg4uwMC, tento ukladá informácie o štatistike do Job Report súboru.

Obvod / Nastavenia súvisiace s obvodom / Priradený súbor

Tento príkaz umožňuje manuálne zadať plný názov súboru, ktorý súvisí s aktuálnym obvodom. Jedná sa o súbor, ktorý môže byť automaticky načítaný do buffera pri výbere tohto obvodu z prednastavených obvodov (t.j. služba <F5>), alebo pri spustení programu po spracovaní súboru uloženého na disku. Editovať názov súboru možno v každom prípade, t.j. aj vtedy, keď automatické čítanie takto zadaného súboru je zablokované. Priradený súbor a povolenie na jeho automatické čítanie pri výbere obvodu sa uchováva na disk príkazom **Súbor / Skončiť s uložením konfigurácie....**

Výberom nového obvodu sa táto voľba prednastavuje na prázdne meno súboru a zablokovanie automatického čítania.

Obvod / Nastavenia súvisiace s obvodom / Špeciálne voľby

Použité termíny a pojmy sú presne podľa špecifikácií uvádzaných jednotlivými výrobcami obvodov. Vysvetlenie týchto termínov sa uvádza v dokumentácii poskytnutej jednotlivými výrobcami pre každý obvod.

Keď názov položky menu začína "View/Edit...", potom operácia "Čítať" obvod číta aj obsah konfiguračných špeciálnych oblastí obvodu, ktoré môžu byť prezerané a editované cez túto menu položku.

Obvod / Skontrolovať vymazanie

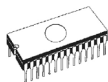
Príkaz umožňuje skontrolovať celý obvod (alebo jeho časť) či je vymazaný. Ovládací program ohlásí výsledok činnosti výpisom hlásenia na obrazovku.

Ak chcete kontrolovať iba časť obvodu, nastavte v menu **Obvod / Nastavenia súvisiace s obvodom / Operácie s obvodom** počiatočnú a koncovú adresu aktuálnej oblasti za predpokladu, že pre daný obvod je táto služba podporovaná.

Obvod / Čítať

Príkazom sa prečíta celý obsah obvodu (alebo jeho časť) a uloží sa do buffera ovládacieho programu. Operácia čítania dokáže prečítať tiež špeciálne oblasti obvodu alebo konfiguračné nastavenia obvodu (pokiaľ sa dajú z obvodu prečítať). Takéto nastavenia sa potom dajú v programe prezerat' a/alebo editovať pomocou menu príkazov **Buffer / Zobrazit'/Editovat'** a **Obvod / Nastavenia súvisiace s obvodom / Špeciálne voľby (Alt+S)**.

Ovládací program ohlásí výsledok činnosti výpisom hlásenia na obrazovku.



Ak chcete prečítať iba časť obvodu, použite menu **Obvod / Nastavenia súvisiace s obvodom / Operácie s obvodom** na nastavenie počiatočnej a koncovej adresy aktuálnej oblasti za predpokladu, že pre daný obvod je táto služba podporovaná. Nastaviť môžete taktiež automatickú **verifikáciu prečítaných údajov**, čo vedie k zvýšenej spoľahlivosti pri čítaní obvodu.

Obvod / Verifikovať

Príkaz kontroluje správnosť naprogramovaných dát porovnaním obsahu obvodu s obsahom buffera ovládacieho programu v platnom rozsahu nastavených adries. Ovládaci program ohlásí výsledok činnosti výpisom hlásenia na obrazovku.

Platný rozsah adries nastavte v menu **Obvod / Nastavenia súvisiace s obvodom / Operácie s obvodom** za predpokladu, že pre daný obvod je táto služba podporovaná.

V menu **Nastavenia / Zobrazit' chyby** nastavte spôsob zobrazenia vzniknutých chýb. Chyby sa môžu zobrazovať do súboru VERIFY.ERR alebo na obrazovku. Pri zobrazení chýb na obrazovku sa ich však zobrazí iba prvých 45.

Poznámka:

Operácia Verifikovať porovnáva obsah celého obvodu s dátami v programe, preto sa môže stať, v prípade programovania nie celého obvodu, že verifikácia po programovaní neukáže chyby, ale operácia Verifikácia áno.

Operácia Verifikovať bude hlásiť chyby aj v prípade chránených obvodov, ktoré majú zapnutú ochranu proti čítaniu.

Obvod / Programovať

Príkaz vykoná naprogramovanie celého obvodu (časti) dátami z buffera. Ovládaci program ohlásí výsledok činnosti výpisom hlásenia na obrazovku.

Riadenie programovacieho procesu zabezpečujú nastavenia v menu **Obvod / Nastavenia súvisiace s obvodom / Operácie s obvodom**. Je to variabilné prostredie, ktoré obsahuje nastaviteľné položky súvisiace s aktuálnym obvodom a typom programátora.

Obvod / Vymazať

Vymazanie celého programovateľného obvodu. Ovládaci program ohlásí výsledok činnosti výpisom hlásenia na obrazovku.

Pre obvody, u ktorých nie je možné overiť výsledok akcie Vymazať inak, sa po akcii vykoná kontrola vymazania obvodu spustením akcie **Obvod / Skontrolovať vymazanie**.

Obvod / Testovať

Táto položka umožňuje spustenie testu obvodu vybraného z knižnice podporovaných obvodov (napr. statickej RAM), ak je tento test podporovaný aktuálnym programátorom.

Obvod / Test IO

Tento príkaz aktivuje nadstavbu pre testovanie integrovaných obvodov rozdelených podľa kompatibility do niekoľkých knižníc, ktoré sú súčasťou ovládacieho programu. Príkaz umožňuje výber príslušnej knižnice, želaného obvodu a režimu spracovania testovacích

vektorov pre vybraný obvod (**cyklicky** alebo **krokovanim**). Postupnosť a výsledky vykonávaných testov sú zobrazované do Záznamového okna (Log okno). V prípade potreby je možné taktiež definovanie testovacích vektorov priamo užívateľom. Podrobný popis syntaxe a spôsobu vytvárania testovacích vektorov je popísaný v súbore `exampl_s.lib`, ktorý sa nachádza v adresári programu.

Poznámka: *Pretože nábežná a dobežná hrana signálov programátora je nastavená na programovanie obvodov, môže sa stať, že testy niektorých obvodov neprejdú (napríklad čítače). To však neznamená, že tieto obvody sú chybné.*

Obvod / JAM/VME/... Prehrávač

Jam STAPL bol vytvorený inžiniermi firmy Altera® a je podporovaný združením veľkovýrobcov programovateľných logických obvodov (PLD), výrobcami programovateľných zariadení a veľkovýrobcami testovacích zariadení.

Jam™ Standard Test and Programming Language (STAPL), JEDEC štandard JESD-71, je štandardným súborovým formátom pre ISP (In-System Programming) účely. Jam STAPL je voľne licencovaný otvorený štandard. Podporuje programovanie alebo konfigurovanie programovateľných zariadení a testovanie elektronických systémov, ktoré používajú IEEE 1149.1 Joint Test Action Group (JTAG) rozhranie. Obvod môže byť programovaný alebo verifikovaný, ale Jam STAPL zvyčajne neposkytuje iné funkcie ako napr. čítanie obvodu.

Programové riešenie Jam STAPL pozostáva z dvoch súčastí: Jam generátora a Jam Prehrávača.

Jam generátor je program, zvyčajne napísaný dodávateľom programovacej logiky, ktorý vytvorí Jam súbor (.jam), ktorý obsahuje užívateľské údaje a programovací algoritmus potrebný k naprogramovaniu návrhu do obvodu.

Jam Prehrávač je program, ktorý číta Jam súbor a aplikuje príkazy pre programované a testované zariadenia JTAG reťaze.

Zariadenia môžu byť programované v ZIF päťici programátora alebo v cieľovom systéme cez ISP konektor. To je indikované [PLCC44](Jam), (ISP-Jam) príponou za menom vybraného obvodu. Viac obvodov je možné programovať pomocou JTAG zreťazenia: JTAG chain (ISP-Jam).

Viac informácií na web stránke: <http://www.altera.com>

Pozrite prosím aplikačné poznámky:

"AN 425: Using the Jam Player to Program Altera Devices",

"AN 100: In-System Programmability Guidelines",

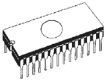
"AN 122: Using Jam STAPL for ISP & ICR via an Embedded Processor".

a v prípade potreby aj ďalšie dostupné aplikačné poznámky.

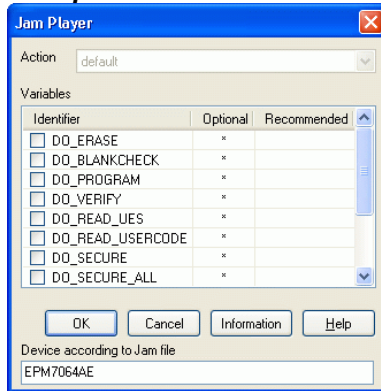
Programové nástroje:

Altera: MAX+plus II, Quartus II, SVF2Jam utilita (konvertuje serial vektor súbor na Jam súbor), LAT2Jam utilita (konvertuje ispLSI3256A JEDEC súbor na Jam súbor)

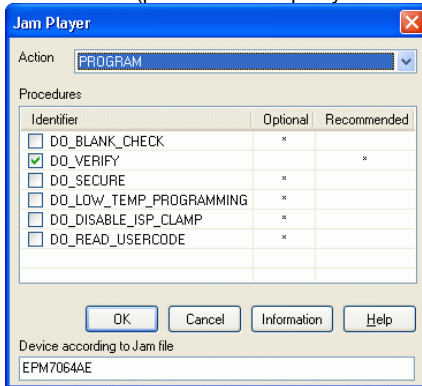
Xilinx: Xilinx ISE Webpack or Foundation software (vygeneruje STAPL súbor alebo SVF súbor pre použitie utilitou SVF2Jam)



informácie o dialógu JAM prehrávač



Jam Prehrávač verzie 1 (pozri ovládacie prvky Akcia a Premenné)



Jam Prehrávač verzie 2 (pozri ovládacie prvky Akcia a Procedúry)

Akcia

Zvoľte požadovanú akciu pre vykonanie.

Jam súbor verzie 2 pozostáva z akcií. Akcia pozostáva z volania procedúr, ktoré sú vykonávané.

Jam súbor verzie 1 nepozná výrazy 'action' a 'procedure', preto nie je dostupná voľba Akcie. Programový tok začína vykonávať inštrukcie podľa booleovských premenných s predponou DO_niečo. Ak potrebujete nejaké nové booleovské premenné s predponou DO_niečo, tak nás kontaktujte.

Procedúry

Program vykonáva výrazy každej procedúry. Procedúry môžu byť voliteľné a odporúčané. Odporúčané procedúry sú predznačené. Procedúry môžete povoľovať a zakazovať podľa potreby. Jam Prehrávač vykonáva len označené procedúry. Zvyšné procedúry sú ignorované. Počet procedúr je rozličný, závisí od Jam súboru.

Premenné

Jam súbor verzie 1 nepozná výrazy 'action' a 'procedure'. Programový tok začína vykonávať inštrukcie podľa booleovskej premennej s predponou DO_niečo. Jam Prehrávač vykonáva všetky označené DO_niečo vetvy v algoritme. Počet premenných (procedúr) je konštantný, nezávisí od Jam súboru. Ak potrebujete nejaké nové booleovské premenné s predponou DO_niečo, tak nás kontaktujte.

OK

Vykoná zvolenú akciu s príslušnými procedúrami, ktoré sú označené.

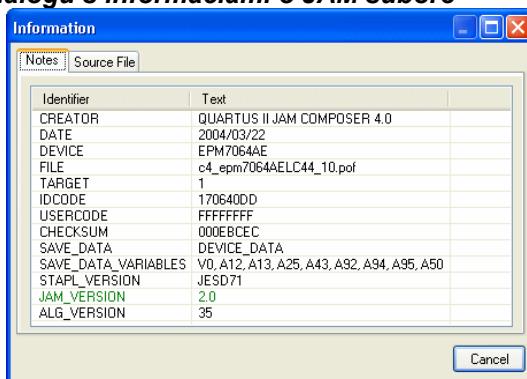
Informácie

Zobrazí informácie o Jam súbore. V dialógu môžete prezerať poznámky a zdrojový súbor.

Obvod podľa Jam súboru

Jam súbor je vytvorený pre určitý obvod. Názov obvodu sa nachádza v Jam súbore v časti Poznámky, identifikátor DEVICE. Názov obvodu musí byť zhodný s názvom vybratého obvodu v dialógu Výber obvodu. Keď sú obvody rôzne, program indikuje danú situáciu varovnou správou pri spustení Jam Prehrávača.

Informácie o dialógu s informáciami o JAM súbore



Poznámky

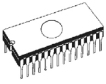
Poznámkové výrazy sú použité k uloženiu informácií týkajúcich sa Jam súboru. Informácie uložené v poliach Poznámky môžu obsahovať ľubovoľné dokumentačné typy alebo atribúty súvisiace s príslušným Jam programom.

Zdrojový súbor

Obsahuje program v Jam jazyku. Jam program sa skladá zo sekvencie výrazov. Jam výraz pozostáva z návestia, ktoré je voliteľné, inštrukcie a argumentov ukončených bodkočiarkou (;).

Argumentmi môžu byť konštanty, premenné alebo výrazy vracajúce očakávaný dátový typ (Boolean alebo integer). Výraz zvyčajne zaberá jeden riadok Jam programu, ale nie je to podmienkou. Nové riadky nie sú podstatné pre syntax jazyka Jam s výnimkou ukončenia komentárov. Znak apostrof (') môže byť použitý k vyznačeniu komentára, ktorý je ignorovaný interpretom. Jazyk nešpecifikuje žiaden limit pre dĺžku riadku, výrazu alebo veľkosť programu.

Jam súbor s príponou .jbc je Jam STAPL Bajtovo kódovaný formát, ktorý nie je viditeľný.



Konvertovanie JED súboru na Jam STAPL súbor pre obvody Xilinx:

1. nainštalujte program Xilinx Integrated Software Environment (ISE) 6.3i, k voľnému stiahnutiu: WebPACK_63_fcfull_i.exe + 6_3_02i_pc.exe (315MB)
2. spustíte Xilinx ISE 6/Accessories/iMPACT
 - v dialógu „Operation Mod Selection: What do you want to do first?“ zvolíte: „Prepare Configuration Files“,
 - v dialógu „Prepare Configuration Files: I want create a:“ zvolíte: „Boundary-Scan File“,
 - v dialógu „Prepare Boundary-Scan File: I want create a:“ zvolíte: „STAPL File“,
 - v dialógu „Create a New STAPL File“ napíšete meno Jam súboru s príponou .stapl,
 - v dialógu „Add Device“ zvolíte JED súbor s príponou .jed,
 - vo vytvorenej jtag reťazi zvolíte obvod napr.: XC2C32A (ľavé tlačidlo myši) a zvolíte postupnosť operácií (napr.: Erase, Blank, Program, Verify; pravé tlačidlo myši),
 - v menu vyberete položku “Output/Stapl file/Stop writing to Stapl file”.
3. spustíte Pg4uw, vyberete obvod napr.: Xilinx XC2x32A [QFG32](Jam), nahrajte Jam súbor (Typ súboru: vyberte STAPL File)
4. vyberete “Operácie s obvodom Alt+O” stlačíte tlačidlo “Jam konfigurácia”. Upozornenie “Vybrať obvod z menu “Vybrať obvod” a Jam súboru je pravdepodobne odlišný! Pokračovať?” vyberte Áno. (Xilinx softvér nevkladá riadok: NOTE "DEVICE" "XC2x32A"; do Jam súboru). V dialógu “Jam Prehrávač” vyberete akciu a procedúry, ukončíte dialógy, stlačíte tlačidlo “Jam” z nástrojovej lišty a sledujte Log okno.

ispVM Virtual Machine

ispVM Virtual Machine je virtuálny stroj optimalizovaný pre programovanie kompatibilných obvodov podľa štandardu IEEE 1149.1 pre Boundary Scan testovanie. ispVM EMBEDDED nástroj spája schopnosti ispVM Virtual MachineTM s priemyselným štandardom Serial Vector Format (SVF), jazyka pre Boundary Scan programovanie a testovanie.

Softvér ispVM System generuje VME súbory zo súborov JTAG, ktoré spĺňajú IEEE 1149.1 štandard a sú zapísané vo SVF alebo IEEE 1532 formáte. Zariadenia môžu byť programované v ZIF päťci programátora alebo v cieľovom systéme cez ISP konektor. To je indikované [PLCC44](VME), (ISP-VME) príponou za menom vybraťého obvodu. Viac obvodov je možné programovať pomocou JTAG zariadenia: JTAG chain (ISP-Jam).

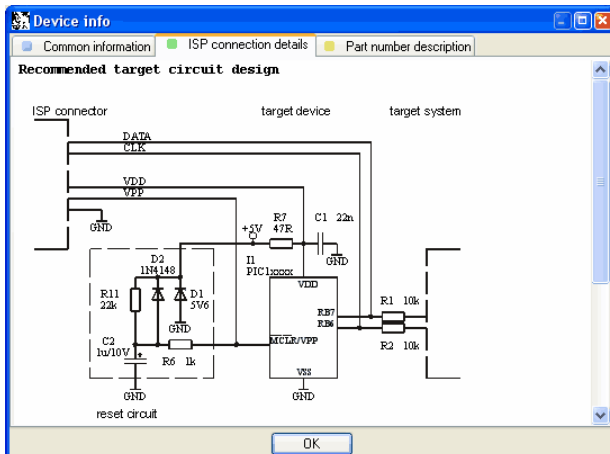
Viac informácií na web stránke: <http://www.latticesemi.com>

Programové nástroje:

Lattice: ispLEVER, ispVM System ISP Programming Software, PAC-Designer Software, svf2vme utility (konvertuje SVF súbor na VME súbor)

Obvod / Informácie o obvode

Príkaz poskytuje prídavné informácie o aktuálnom obvode. Informácie obsahujú pevnú a pružnú časť. Pevná časť obsahuje veľkosť obvodu, organizáciu, programovací algoritmus a zoznam programátorov (včítane požadovaných modulov), ktoré tento obvod podporujú. Pružná časť môže obsahovať informácie o puzdrení a rôzne iné všeobecné informácie o danom obvode.



Na aktiváciu tohto príkazu možno použiť skracovaciu klávesu <Ctrl+F1>.

Programátor

Menu **Programátor** obsahuje príkazy pre prácu s programátorom.

Programátor / Hľadať programátor

Tento príkaz umožňuje počas behu programu nastaviť nový typ programátora a parametre komunikácie. Obsahuje nasledujúce položky:

Programátor - Nastaví nový typ programátora, ktorý sa bude vyhľadávať. Ak je nastavené Hľadať všetky, potom ovládací program bude hľadať všetky podporované programátory.

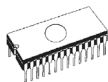
Nadviazanie komunikácie - Umožňuje manuálne alebo automatické nastavenie komunikačnej rýchlosti pre vybraný programátor.

Rýchlosť - V tejto položke sa nastavuje rýchlosť, ak je povolené ručné nastavenie komunikačnej rýchlosti, ktorou bude PC posielat' údaje do programátora. Rýchlosť je udávaná v percentách z maximálnej rýchlosti.

Možnosť modifikovať komunikačnú rýchlosť má význam u počítačov s málo výkonnými obvody paralelného portu, neschopnými dostatočne rýchlo budiť prepojovací kábel (laptop, notebook, ...). Použite túto možnosť v prípade, že sa programátor - aj keď pripojený k PC - vôbec neprihlási, alebo ak je komunikácia s programátorom nespoľahlivá.

Ak je nastavené automatické nastavenie komunikačnej rýchlosti, ovládací program nastaví maximálnu možnú rýchlosť.

Port - umožňuje vybrať port, na ktorom bude nový programátor hľadaný. Ak je zvolené Všetky porty, potom ovládací program bude hľadať programátor na všetkých štandardných portoch, ktoré sú dostupné na adresách.



Adresa špeciálneho portu - umožňuje nastaviť adresu paralelného portu ak bol vybraný špeciálny port.

Stlačením klávesy **<Enter>** alebo tlačítka **OK** sa aktivuje hľadanie želaného programátora podľa nastavených parametrov. Vykoná sa rovnaká činnosť ako pri spustení ovládacieho programu. Vymaže sa zoznam obvodov, s ktorými sa naposledy pracovalo okrem aktuálneho, ak ho nový programátor podporuje.

Toto nastavenie sa ukladá na disk príkazom **Programátor / Uložiť nastavenia**.

Programátor / Hľadať znova programátor

Tento príkaz umožňuje znova vyhľadať, t.j. znova nadviazať komunikáciu s aktuálne zvoleným programátorom.

Pre výber želaného typu programátora, komunikačných nastavení pre programátor a prvotné nadviazanie komunikácie s programátorom sa používa menu položka **Programátor / Hľadať programátor**.

Programátor / Handler

V dialógu **Handler** sa dá zvoliť typ handlera a nastavenia komunikácie handlera s ovládacím programom. Handler je externé zariadenie pre riadenie akcií na obvodoch v ovládacom programe. Keď je vybraný typ handlera "Nie je", tak program sa nachádza v štandardnom móde ovládania akcií na obvodoch, čo znamená, že akcie na obvodoch riadi priamo užívateľ pomocou klávesnice, resp. myši. Ak je zvolený konkrétny handler, tak ovládací program sa nachádza v špeciálnom móde, kde akcie na obvodoch sú riadené automaticky v kooperácii s handlerom.

Dialóg **Handler** obsahuje nasledujúce položky:

Vybraný handler – nastavenie požadovaného handlera

Hľadať na porte - umožňuje vybrať sériový port, na ktorom bude požadovaný handler hľadaný

Stlačením klávesy **<Enter>** alebo tlačítka **OK** sa aktivuje hľadanie želaného handlera podľa nastavených parametrov.

Toto nastavenie sa ukladá na disk príkazom **Nastavenia / Uložiť nastavenia**.

Programátor / Nastavenia modulu

Tento príkaz sa používa u násobných programátorov na nastavenie **MASTER** päťice a aktivity jednotlivých päťíc. V časti **MASTER** päťica sa nastavuje päťica, ktorá sa používa pre operáciu čítania z obvodu. V časti **Povolit' / Zakázat' päťice** sa nastavuje povolenie alebo zakázanie jednotlivých päťíc. Zakázané päťice sú ignorované pri všetkých činnostiach na obvodoch.

Programátor / Automatické YES!

Príkaz slúži na nastavenie módu **Automatické YES!**. V tomto móde stačí na zopakovanie naposledy vykonanej operácie len vložiť do ZIF päťice nový obvod. Program automaticky detekuje vloženie nového obvodu a spustí vykonávanie operácie bez potreby stlačiť nejakú

klávesu, resp. tlačítko. Vloženie obvodu do ZIF päťice je zobrazené na obrazovke. Opakované vykonávanie operácie je možné ukončiť stlačením klávesy <Esc> počas čakania na vloženie/vybratie obvodu do/zo ZIF päťice.

Po ukončení akcie s obvodom, program zasvieti jednu zo stavových LED programátora (OK alebo ERROR) v závislosti od výsledku ukončenej akcie a zároveň začne blikať LED BUSY. Keď program detekuje vybratie obvodu zo ZIF päťice, zhasne stavovú LED, ale nechá blikať LED BUSY, čo indikuje pripravenosť programu zopakovať operáciu s novým obvodom.

Hneď ako program indikuje jeden alebo viac pinov (nového) obvodu v ZIF päťici, rozsvieti LED BUSY trvale. Od tohto okamihu program čaká prednastavený čas na úplné vloženie nového obvodu do ZIF päťice programátora. Ak sa tak v nastavenom časovom intervale (**Čakanie na kompletné vloženie obvodu**) nestane, program zasvieti na programátore LED ERROR, aby indikoval tento stav. Po úspešnom vložení nového obvodu do ZIF programátora, program zhasne všetky stavové LED, okrem LED BUSY a začne vykonávať opakovanú operáciu s novým obvodom.

Povolenie alebo zakázanie tohto módu je možné položkou **Automatické YES!**. Výberom nového programátora pomocou **Programátor / Najst' programátor** sa tento mód zakáže.

V položke **Čas odozvy** je možné predĺžiť časový interval, počas ktorého musí byť v ZIF päťici detekovaný obvod, aby program akceptoval založenie nového obvodu. Prednastavený je štandardný časový interval. Pri použití konvertora puzdra sa odporúča zvoliť predĺžený interval.

Pole **Použitý programovací adaptér** ukazuje názov použitého adaptéra potrebného na prácu s obvodom.

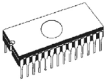
V položke **Piny ZIF päťice programátora, ktoré nebudú detekované** je možné zadať zoznam pinov, ktoré budú ignorované počas testovania Automatic YES!. Väčšinou je dôvodom pripojený kondenzátor na daný pin.

Stlačením tlačidla **Nastavenie parametrov pre Automatické YES!** sa spustí wizard, ktorý môže detekovať trvale pripojené piny (piny s kondenzátormi) a nastaví tieto piny do zoznamu pinov, na ktorých nemá prebiehať detekcia. Po výbere obvodu sa tento zoznam nastaví ako prednastavený pre aktuálne vybraný adaptér. Ak si užívateľ pridá alebo odoberie kondenzátory na adaptéri, je dôležité spustiť wizard **Nastavenie parametrov pre Automatické YES!** znova, aby došlo k prepísaniu prednastavených pinov a detekovaniu ďalších pinov.

V položke **Čakanie po výbere obvodu** je možné nastaviť časový interval, počas ktorého nebude program po vybratí obvodu zo ZIF päťice testovať vloženie nového obvodu. Tento interval je zadávaný v sekundách a jeho hodnota môže byť od 1 do 120 (hodnota je prednastavená na 2 sekundy).

V položke **Čakanie na kompletné vloženie obvodu** je možné nastaviť časový interval, počas ktorého musia byť detekované všetky piny (nového) obvodu v ZIF päťici programátora po detekovaní prvého pinu v ZIF. Tento interval je zadávaný v sekundách a jeho hodnota môže byť od 1 do 120 (hodnota je prednastavená na 5 sekúnd).

V položke **Dočasné prerušenie pri chybe** je možné zvoliť si správanie sa funkcie Automatické YES! po detekovaní chyby počas práce s obvodom. Nastavením tejto položky



môže byť funkcia buď dočasne prerušená, aby bolo možné prezrieť si výsledok akcie alebo bude pokračovať ďalej (funkcia Automatické YES!) bez prerušenia.

Výberom nového obvodu pomocou **Obvod / Vybrať obvod** alebo **Obvod / Vybrať z prednastavených obvodov**, je tento zoznam vymazaný.

Toto nastavenie sa ukladá na disk príkazom **Nastavenia / Uložiť nastavenia**.

Programátor / Test programátora

Procedúra vykoná test aktuálne vybraného programátora bez použitia diagnostickej hlavice. Pre zvýšenie spoľahlivosti testovania odporúčame vykonať aj **Programátor / Test programátora plus**. Obyčajný test programátora (bez diagnostickej hlavice) totiž nemusí odhaliť niektoré druhy hardvérových chýb.

Programátor / Test programátora plus

Procedúra vykoná test aktuálne vybraného programátora použitím diagnostickej hlavice, dodávanej štandardne s programátorom.

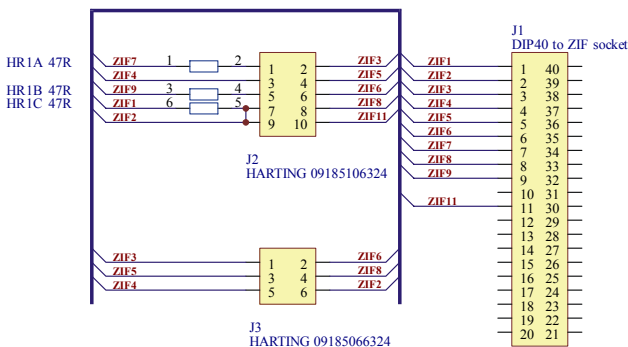
Odporúčania ako často spustiť Selftest plus nájdete v časti **Starostlivosť o programátor**.

Programátor / Test ISP konektora

Procedúra vykoná test ISP konektora aktuálne vybraného programátora použitím diagnostickej hlavice pre ISP konektory.

Diagnostic POD for ISP connectors slúži na otestovanie 6 a 10 pinových ISP konektorov programátorov a je k dispozícii ako doplnkové príslušenstvo pre programátory s 6 a 10 pinovým ISP konektorom (BeeProg, SmartProg2, SmartProg, PIKprog2, PIKprog+, T51prog2 and T51prog). Objednávacie číslo diagnostickej hlavice pre ISP je 70-0208.

Schéma **Diagnostic POD for ISP connectors**:



Postup testovania 6 pinového ISP konektora programátora:

1. Diagnostic POD for ISP connectors založiť do ZIF päťice programátora. Diagnostic POD for ISP connectors sa do ZIF päťice zakladá ako 40 pinový obvod.

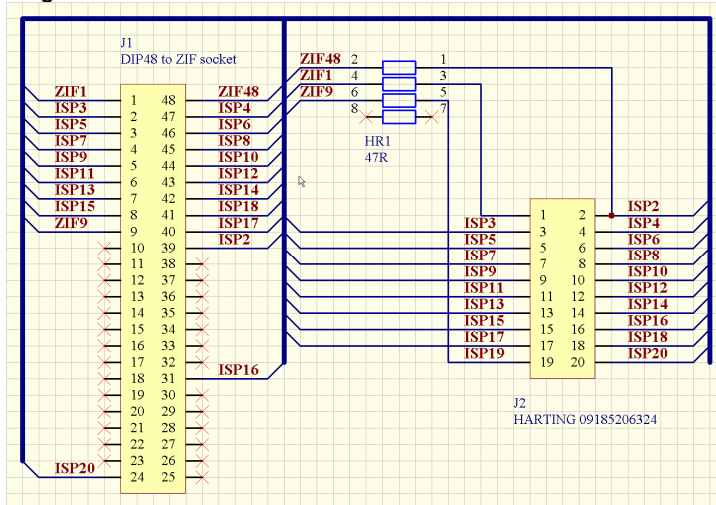
2. ISP konektor programátora prepojiť so 6 pinovým konektorom Diagnostic POD for ISP connectors pomocou ISP kábla, ktorý je obsahom dodávky programátora. Pri prepojovaní je nutné dbať na to, aby boli piny správne prepojené (t.j. 1-1, 2-2, ..., 6-6).
3. Spustiť vykonanie selftestu ISP konektora v programe Pg4uw (Diagnostika / Test ISP konektora).

Postup testovania 10 pinového ISP konektora programátora:

1. Diagnostic POD for ISP connectors založiť do ZIF päťice programátora. Diagnostic POD for ISP connectors sa do ZIF päťice zakladá ako 40 pinový obvod.
2. ISP konektor programátora prepojiť s 10 pinovým konektorom Diagnostic POD for ISP connectors pomocou ISP kábla, ktorý je obsahom dodávky programátora. Pri prepojovaní je nutné dbať na to, aby boli piny správne prepojené (t.j. 1-1, 2-2, ..., 10-10).
3. Spustiť vykonanie selftestu ISP konektora v programe Pg4uw (Diagnostika / Test ISP konektora).

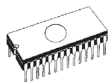
Diagnostic POD for ISP connectors #2 slúži na otestovanie 20 pinových ISP konektorov programátorov a je k dispozícii ako štandardné doplnkové príslušenstvo pre programátory s 20 pinovým ISP konektorom (BeeHive208S, BeeHive8S, BeeHive204, BeeHive4+ BeeProg2 a BeeProg+). Objednávacie číslo Diagnostic POD for ISP connectors #2 je 70-0680.

Schéma Diagnostic POD for ISP connectors #2:



Postup testovania 20 pinového ISP konektora programátora:

4. Diagnostic POD for ISP connectors #2 založiť do ZIF päťice programátora. Diagnostic POD for ISP connectors #2 sa do ZIF päťice zakladá ako 40 pinový obvod.
5. ISP konektor programátora prepojiť s 20 pinovým konektorom Diagnostic POD for ISP connectors #2 pomocou ISP kábla, ktorý je obsahom dodávky programátora. Pri prepojovaní je nutné dbať na to, aby boli piny správne prepojené (t.j. 1-1, 2-2, ..., 20-20).



6. Spustiť vykonanie selftestu ISP konektora v programe Pg4uw (Diagnostika / Test ISP konektora).

Odporúčame tento test vykonávať raz za 6 mesiacov.

Programátor / Test kalibrácie

Procedúra vykoná test kalibračných hodnôt programátora pomocou **48 Pins Calibration test POD, Type I**, ktorý je voliteľným príslušenstvom pre BeeHive208S, BeeHive8S, BeeHive204, BeeHive4+, BeeHive4, BeeProg2, BeeProg+ a BeeProg. Objednávacie číslo 48 Pins Calibration test POD, Type I je 70-0438.

Počas testu sú skontrolované úrovne pre TTL a správnosť nastavenia napätí pre VCCP, VPP1 a VPP2 na každom pine ZIF pátky. Výsledok Testu Kalibrácie môže byť uložený do súboru na disku a/alebo vytlačený na tlačiarňu pre neskoršie použitie.

Programátor / Všeobecné poznámky

Sériové čísla

Nasledujúca tabuľka uvádza rozdiely medzi starým a novým spôsobom číslovania programátorov sériovými číslami:

| nové číslovanie | staré číslovanie |
|-----------------|------------------|
| 050-00xxx | 050-xxx |
| 050-01xxx | 950-xxx |
| 050-02xxx | 850-xxx |
| 050-03xxx | 750-xxx |

Nastavenia

Menu obsahuje príkazy, ktorými je možné prezerat' a menit' rôzne prednastavenia programu.

Nastavenia / Všeobecné nastavenia

Dialóg Všeobecné nastavenia môže užívateľ nastavovat' a menit' rôzne nastavenia programu Pg4uw. Tieto nastavenia budú uložené do konfiguračného súboru pri ukončení programu Pg4uw, alebo kedykoľvek príkazom Nastavenia / Uložit' nastavenia.

Načítanie súborov

Táto stránka obsahuje nastavenia týkajúce sa prípon súborov v jednotlivých formátoch, spôsobu znovunačítania aktuálneho súboru pri jeho modifikácii inými aplikáciami a rozpoznavania formátov načítavaných súborov.

V paneli **Vymazať buffer pred načítaním** sa nastavuje možnosť automatického vymazania dát v bufferi zvolenou hodnotou pred každým načítaním nového súboru.

Panel **Ak je aktuálny súbor modifikovaný inou aplikáciou** umožňuje nastaviť spôsob znovunačítania aktuálneho súboru, ak bol súbor zmenený inou aplikáciou. Je možné vybrať z troch možností.

- Otázka pred znovu - načítaním zmeneného súboru
- Automatické znovu - načítanie
- Ignorovanie zmien

Kontrola na zmeny súboru a následná akcia sa vykonávajú v troch prípadoch:

- pri prepnutí do ovládacieho programu z inej aplikácie

- pri zvolení akcie **Kontrolovať** alebo **Programovať** obvod pred samotným spustením akcie
- pri znovu - opakovaní akcie na obvode potvrdením voľby **Áno** v okne "Opakovať?"

Formát načítaného súboru riadi načítanie zvoleného súboru z disku do buffera s automatickým alebo manuálnym (užívateľom daným) rozpoznaním formátu. Pri načítaní súboru z disku do buffera automaticky rozpoznaným formátom sa program pokúsi analyzovať formát načítavaného súboru, pričom testuje všetky programom podporované formáty. Ak súbor je v niektorom z daných formátov, načíta sa súbor príslušným spôsobom do buffera. Manuálny výber formátu pre čítanie súboru z disku do buffera umožňuje užívateľovi vo výberovom okne formátov zvoliť formát podľa želania. Samozrejme ak súbor nie je vo formáte, ktorý definoval užívateľ, môže byť načítanie súboru neúspešné, najmä pokiaľ ide o formáty v ASCII tvare s normovanými formátmi danými štandardami.

Položka **Zobraziť dialóg "Načítať vybraný projekt" pri štarte programu** umožňuje nastaviť zobrazenie dialógu pri spustení programu Pg4uw. Dialóg **Načítať** aktuálny projekt obsahuje zoznam posledne čítaných projektov (históriu projektov) a dovoľuje užívateľovi rýchly výber a načítanie projektu pri spustení programu.

Prípory súborov

Táto stránka umožňuje nastaviť masky súborov.

V okienkach **Masky formátov** súborov sa nastavujú masky, ktorými sa filtruje zobrazovaný zoznam súborov v okne pre uloženie **Súbor / Uložiť** a načítanie **Súbor / Čítať** (pre uloženie má každý formát vlastnú masku). Masky musí obsahovať aspoň jeden z používaných pseudoznakov (*,?) a musí byť zadaná syntakticky správne, ináč sa nebude akceptovať.

Poznámka: Viaceré masky môžu byť priradené každému formátu. Bodkočiarka sa používa ako oddeľovač týchto prípon.

Príklad: *Motorola: *.MOT;*.S19*

*2 masky *.MOT a *.S19 sú priradené pre Motorola formát*

Prípory súborov projektov umožňuje nastaviť príponu pre projekty, ktoré používa ovládací program. Prípory sa používa v dialógoch **Súbor / Čítať** projekt a **Súbor / Uložiť** projekt.

Buffer

Táto stránka umožňuje nastaviť **Zmazať buffer pred výberom nového obvodu**. To môže byť užitočné pre niektoré typy obvodov, ktoré vyžadujú špeciálny formát dát na presných adresách v prípadoch, keď tieto dáta nie sú súčasťou načítaného súboru s dátami pre tento obvod.

Buffer môže byť vymazaný (naplnený) s prednastavenou „blank“ hodnotou pre vybraný obvod alebo užívateľom definovanou hodnotou. Toto nastavenie sa realizuje v skupine **Mazacia hodnota** a v poli **Definovaná mazacia hodnota**.

Poznámka: *Nedoporučujeme používať túto funkciu pre veľké obvody (viac ako 8MB), pretože to môže trvať dlhšie ako príkaz Výmaz buffera. Nastavenie je uložené do konfiguračného súboru. Nie je uložené do projektového súboru.*

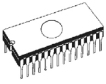
Jazyk

Táto voľba umožňuje nastaviť rozličné jazyky pre užívateľské prostredie programu, ako sú menu, tlačidlá, dialógy, správy a chybové správy. Taktiež umožňuje zvoliť help súbor v rôznych jazykoch. Pre podporu konkrétneho jazyka sú potrebné príslušné externé jazykové súbory.

Zvuk

Na stránke **Zvuk** sa dá nastaviť spôsob generovania zvukov programom. Program generuje zvuky po ukončení určitých činností, napr. činností na obvode (programovanie, kontrola atď.). Program tiež generuje zvuky pri výpise upozornení alebo chybových hlásení.

Spôsoby generovania zvuku sú:



- Windows zvuky (vyžaduje zvukovú kartu)
- dlhé zvuky cez systémový reproduktor.
- skrátene zvuky cez systémový reproduktor.
- žiadne zvuky

Panel **Povolit' zvuk pre nasledujúce operácie** obsahuje nasledovné položky:

- **Úspešná operácia** – Keď je vybratá táto voľba, zvuk sa bude generovať po úspešnom ukončení operácie s obvodom. Keď nie je vybratá, nebudú sa generovať žiadne zvuky.
- **V prípade chyby** - Keď je vybratá táto voľba, zvuk sa bude generovať po operácii, ktorá skončila s chybou. Keď nie je vybratá, nebudú sa generovať žiadne zvuky.

V paneli **Nastavenia zvuku pre reproduktor zabudovaný v programátore** je možné u niektorých programátorov nastaviť generovanie zvuku výsledku operácie na obvode. Zvuk sa generuje interným reproduktorom zabudovaným v programátore a to len po vykonaní akcie na obvode, úspešnej alebo neúspešnej.

Chyby

Táto voľba súvisí s ukladaním chýb vzniknutých pri verifikácii naprogramovaných údajov do súboru na disku. Zápis verifikačných chýb na obrazovku niekedy nestačí, navyše sa zobrazuje max. 45 prvých rozdielov. V sekcii **Uložit' výpis chýb kontroly obvodu do súboru** môžete voľbu vypnúť alebo nastaviť tak, že sa buď verifikačné údaje každej verifikačnej akcie kumulujú do toho istého súboru alebo sa pre každú akciu pôvodný súbor vymaže a vytvorí sa nový len pre data z poslednej akcie. Meno súboru, do ktorého sa majú chyby zapisovať je nastaviteľné v editačnom riadku **Meno súboru chýb**. K dispozícii sú teda nasledujúce nastavenia:

- **Nie** (predvolená) zápis verifikačných chýb do súboru je vypnuté (funguje len zápis na obrazovku)
- **Nový** do súboru sa uložia len chyby z poslednej verifikácie
- **Pripojiť** do súboru sa kumulujú chyby všetkých verifikačných akcií

V sekcii **Obmedzenie veľkosti súboru chýb** je možné nastaviť:

Ukončenie verifikácie po dosiahnutí max. počtu chýb

Zaškrtnutá - verifikácia sa ukončí hneď, ako sa do súboru zapíše nastavený počet chýb.
Nezaškrtnutá - voľba znamená, že sa do súboru uložia všetky verifikačné chyby.

Max. počet chýb špecifikuje množstvo chýb, ktoré sa majú do súboru zachytiť.

Nastavenie sa ukladá na disk príkazom **Nastavenia / Uložit' nastavenia**.

Log súbor

Tento príkaz umožňuje nastaviť voľbu, ktorá veľmi úzko súvisí s použitím **Záznamového okna** (Log window). Všetky správy, ktoré sa do tohto okna zapisujú môžu byť zapísané súčasne aj do Log súboru. Log súbor má štandardne meno Report.rep a vytvára sa v aktuálnom adresári. Meno aj adresár Log súboru sa dá zmeniť v editačnom riadku **Meno Log súboru**.

K dispozícii sú nasledujúce nastavenia:

- **Nie** predvolená, hlásenia z Log okna sa nekopírujú do súboru
- **Nový** hlásenia z Log okna sa kopírujú súboru, pričom sa pri každom spustení programu vymaže starý Log súbor, ak existuje a vytvorí sa nový súbor
- **Pripojiť** hlásenia z Log okna sa kopírujú do existujúceho súboru na koniec súboru, ak súbor neexistuje, tak bude vytvorený

Zaškrtnuté políčko **Pridať dátum do mena Log súboru** umožňuje v prípade zaškrtnutia aktivovať automatické dodanie aktuálneho dátumu k užívateľom zvolenému menu Log súboru. Dátum sa k menu Log súboru dodá nasledujúcim spôsobom:

Predpokladajme, že užívateľom definované meno Log súboru má tvar:

```
<meno_log_súboru>.<prípona_mena_log_súboru>
```

Výsledné meno Log súboru s pridaným dátumom bude:

```
<meno_log_súboru><-yyyy-mmm-dd>.<prípona_mena_log_súboru>
```

Nová časť <-yyyy-mmm-dd> reprezentujúca aktuálny dátum je vsunutá do mena súboru, pričom význam častí je yyyy - rok, mmm - mesiac a je dd -deň.

Príklad: Meno log súboru zadané užívateľom nech je `c:\logs\myfile.log`

Uvažujme dátum 7.november 2006. Potom výsledné meno Log súboru aj s pridaným dátumom bude:

```
c:\logs\myfile-2006-nov-07.log
```

Pokiaľ si želáte, aby meno Log súboru obsahovalo len dátum a príponu, je potrebné zadať meno Log súboru len vo forme "cesta + prípona" nasledovne:

```
.<prípona_mena_log_súboru> - bodka je prvým znakom mena súboru
```

Príklad: Nech meno log súboru zadané užívateľom nech je `c:\logs\myfile.log`

Potom výsledné meno Log súboru s pridaným dátumom 7.novembra 2006 bude:

```
c:\logs\2006-nov-07.log
```

Ďalšie nastavenia týkajúce sa Log súboru:

- voľba **Orezať Log súbor, pokiaľ veľkosť súboru dosiahne nastavený limit** - ak je voľba zaškrtnutá, tak obmedzenie veľkosti Log súboru je zapnuté. znamená to, že keď Log súbor dosiahne špecifikovanú veľkosť, potom časť textu z Log súboru sa vyhodí, čím sa zmenší veľkosť Log súboru. Pokiaľ je voľba nezaškrtnutá, tak veľkosť Log súboru nie je limitovaná, iba ak veľkosťou voľného miesta na disku.
- voľba **Maximálna veľkosť Log súboru** určuje maximálnu dovolenú veľkosť Log súboru v kilobajtoch.
- voľba **Podiel vyhodeneného textu** udáva percentuálne množstvo textu, ktorý sa má z Log súboru odstrániť pri zmenšovaní veľkosti Log súboru pri dosiahnutí maximálnej povolenej veľkosti Log súboru. Vyššia hodnota znamená viac vyhodeneného textu.

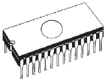
Zmeny nastavení sa uplatnia pri ďalšom štarte programu. Nastavenia pre Log súbor sa ukládajú na disk príkazom **Nastavenia / Uložiť nastavenia**.

Job Report

Job Report reprezentuje celkový opis operácii na obode v nedávnom čase. Job je spojený s projektovým súborom a teda operácie začínajú načítaním projektového súboru a trvajú do načítania nového projekt súboru alebo do ukončenia programu Pg4uw.

Job Report obsahuje nasledovné informácie:

- názov projektu
- dátum projektu
- stav Chráneného módu
- verziu programu Pg4uw
- typ a sériové číslo programátora
- čas štartu vykonávania Jobu (t.j. kedy bol načítaný projekt)



- čas konca vykonávania Jobu (t.j. čas vytvorenia Job Reportu)
- názov obvodu
- typ obvodu
- kontrolná suma
- nastavenia súvisiace s obvodom
- informácie o serializácii
- informácie o štatistike

Job Report sa generuje v nasledovných prípadoch:

- pri zvolení príkazu Načítať projekt
- pri zvolení príkazu zatvorenie alebo odpojenie programovacieho site
- pri ukončení programu Pg4uw
- pri dosiahnutí 0 pre Počítadlo nadol (konečný stav)
- manuálne, keď užívateľ zvolí príkaz Súbor / Vytvoriť Job Report

Job Report sa vygeneruje pre aktuálne načítaný projektový súbor iba vtedy, keď hodnota Celkovo v štatistike je väčšia ako 0. To znamená, že aspoň jedna operácia s obvodom (programovanie, verifikácia) musela byť uskutočnená.

Pre Job Report sú dostupné nasledovné možnosti:

Zaškrtávacie políčko **Povolit' funkciu Job Report** – keď je zaškrtnuté funkcia je povolená. V opačnom prípade nie je povolená.

Zaškrtávacie políčko **Automaticky ukladať súbor Job Report** – keď je zaškrtnuté Job Report sa bude automaticky ukladať do adresára špecifikovaného v položke **Adresár súborov Job Report** s názvom vytvoreným nasledovným spôsobom:

```
job_report_<poradové číslo>_<názov projektu>.jrp
```

kde

<poradové číslo> je poradové číslo súboru v desiatkovej sústave. Ak v danom adresári existujú Job report súbory s rovnakým názvom, poradové číslo bude inkrementované o 1 oproti poslednému číslu.

<názov projektu> je názov projektového súboru aktuálneho projektu bez prípony.

Príklad 1:

Použijeme projektový súbor c:\myproject.epj a adresár pre Job Reporty nech je d:\job_reports\. V danom adresári nie je žiaden Job Report.

Job Report súbor sa bude volať:

```
d:\job_reports\job_report_000_myproject.jrp
```

Príklad 2:

Použijeme podmienky z príkladu 1, ale predpokladajme, že Job Report adresári už je jeden Job Report s menom: job_report_000_myproject.jrp

Nový Job Report súbor sa bude volať:

```
d:\job_reports\job_report_001_myproject.jrp
```

Všimnite si, že poradové číslo vnútri mena sa zvýšilo o 1.

Keď je nastavená voľba **Automatically ukladať súbor Job Report** už sa nebude zobrazovať dialóg na generovanie Job Reportu. Nový Job Report bude uložený do súboru bez akéhokoľvek dialógu alebo správy (ak nenastane chyba pri ukladaní do súboru).

Keď nie je nastavená voľba **Automatically ukladať súbor Job Report** program Pg4uw zobrazí dialóg Job Report vždy keď to bude potrebné.

V dialógu Job Report si užívateľ môže zvoliť, čo chce urobiť s Job Reportom. Môže si ho dať vytlačiť alebo uložiť na disk. Ak zvolí Zavrieť, Job Report sa zapíše iba do Log okna.

Diaľkové ovládanie (Remote control)

Diaľkové ovládanie programu Pg4uw umožňuje ovládať niektoré funkcie programu Pg4uw z inej aplikácie. To je výhodné pre aplikácie integrujúce programátory obvodov do výrobných zariadení - handlerov apod. Diaľkové ovládanie nájde uplatnenie aj v iných aplikáciách, pre ktoré je výhodné riadiť program na diaľku.

Vzdialená aplikácia ovládajúca Pg4uw funguje ako Server. Program Pg4uw funguje ako klient. Komunikácia medzi vzdialeným programom a Pg4uw je realizovaná cez TCP protokol, čo dovoľuje mať Pg4uw inštalovaný na inom počítači ako aplikácia ovládajúca na diaľku program Pg4uw. Jedinou podmienkou je, aby počítače boli vzájomne prepojené do siete s podporou TCP protokolu.

Predvolené nastavenia TCP komunikácie sú nasledovné:

Port: **telnet** Adresa: 127.0.0.1 alebo localhost

Nastavenie adresy má uplatnenie len v programe Pg4uw (klienta). Ide o adresu vzdialeného programu (servera), na ktorý sa má Pg4uw pripojiť.

Nastavenie komunikačného portu má uplatnenie pre Pg4uw aj pre vzdialenú aplikáciu.

Predvolené nastavenia umožňujú použiť diaľkové riadenie programu Pg4uw v rámci jedného počítača, t.j. Pg4uw aj vzdialená aplikácia sú nainštalované na tom istom počítači.

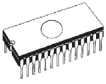
Upozornenie: *Pokiaľ je na počítači aktívny firewall, môže sa pri pokuse o pripojenie Pg4uw a/alebo vzdialenej aplikácie na ovládanie Pg4uw, zobraziť upozornenie aplikácie firewall s otázkou, či dovoliť alebo zamietnuť požiadavku o pripojenie. Treba zvoliť možnosť "dovoliť", inak by vzdialené riadenie programu Pg4uw nefungovalo.*

Pre ďalšie informácie o diaľkovom ovládaní viď. aplikačnú poznámku **remotemanual.pdf**, ktorá sa nachádza v podadresári \RemoteCtrl umiestnenom v adresári, kde je inštalovaný program Pg4uw. Manuál je dostupný taktiež cez odkaz vo Windows menu Štart / Programy vytvorený počas inštalácie Pg4uw.

Možnosti uloženia

Umožňuje zvoliť spôsob uloženia aktuálnych nastavení pri skončení programu tromi spôsobmi:

- **Neuložiť nastavenia** - neukladať nastavenia pri skončení programu ani sa nepýtať na uloženie.
- **Automatické uloženie nastavenia** - ukladať automaticky bez opýtania nastavenia pri skončení programu.
- **Výzva na uloženie nastavenia** - pri skončení programu sa zobrazí otázka umožňujúca pred skončením programu výber z možností uložiť alebo neuložiť aktuálnu konfiguráciu.



Ďalšie

Na stránke **Ďalšie** je možné nastavovať rôzne vlastnosti programu.

V paneli **Priorita aplikácie** sa dá nastaviť priorita programu, čím je možné mierne ovplyvniť výkon programátora (časy programovania), najmä pokiaľ je v systéme spustených viac aplikácií.

V paneli **Počiatočný adresár** sa nastavuje spôsob určenia štartovacieho adresára pri spustení programu. Voľba **Predvolený adresár** znamená, že ako štartovací adresár programu sa bude brať adresár, z ktorého program je volaný pri spustení. Voľba **Adresár, s ktorým program bol naposledy ukončený** znamená, že ako štartovací adresár programu pri spustení sa nastaví ten adresár, v ktorom bol program nastavený pri poslednom ukončení. Ide teda vlastne o prvý (najaktuálnejší) adresár zo zoznamu histórie adresárov.

V paneli **Tlačidlá na paneloch tlačidiel** sa dá nastaviť ploché zobrazenie tlačidiel a zobrazenie bublinovej pomoci pre tlačidlá v hlavnom okne programu.

Nastavenia / Zobrazit'

Príkazy tohto menu slúžia na zobrazenie alebo schovanie rôznych líšt tlačidiel programu.(toolbarov).

Menu obsahuje nasledujúce položky:

Nastavenia / Zobrazit' / Hlavná lišta tlačidiel

Týmto príkazom zobrazíte alebo skryjete hlavnú lištu tlačidiel.

Nastavenia / Zobrazit' / Prídavná lišta tlačidiel

Týmto príkazom zobrazíte alebo skryjete prídavnú lištu tlačidiel.

Nastavenia / Zobrazit' / Nastavenie obvodu pred akciou na obvode

Týmto príkazom sa povolí/zakáže zobrazenie okna nastavenia obvodu pred akciou na obvode, t.j. pri voľbe menu akcie na obvode, napr. Programovať.

Nastavenia / Chránený mód

Chránený mód je špeciálny mód programu, keď sú v programe zablokované tie činnosti, ktorými je možné meniť obsah buffera a meniť nastavenia aktuálne vybraného obvodu. Chránený mód slúži na to, aby sa nestalo, že užívateľ nedopatrením zmodifikuje buffer alebo nastavenia obvodu, najmä pri programovaní väčšieho počtu rovnakých obvodov za sebou, čo môže oslabiť pozornosť operátora.

Funkcia Chránený mód je prístupná nezávisle pre sólo programovanie pomocou programu Pg4uw a pre multiprogramovanie pomocou programu Pg4uwMC.

Chránený mód v Pg4uw

Program sa môže prepnúť do chráneného módu jedným z dvoch nasledujúcich spôsobov:

- 1.pomocou menu položky **Nastavenia / Chránený mód**, kde sa zobrazí dialóg vyzývajúci užívateľa, aby zadal heslo, ktoré slúži na ochranu pred neželaným prepnutím z chráneného módu do normálneho módu. Heslo pri prepínaní do chráneného módu je potrebné potvrdiť zadaním dvakrát.
- 2.pri načítaní projektu, ktorý bol uložený v chránenom móde (pre bližšie informácie vid' **Súbor / Uložiť projekt**).

Program sa z chráneného módu do normálneho módu dá prepnúť pomocou menu položky **Nastavenia / Normálny mód**, kde sa zobrazí dialóg vyzývajúci užívateľa na zadanie hesla.

Po zadaní a potvrzení správného hesla sa zruší chránený mód programu a program bude pracovať v normálnom (štandardnom) režime.

Ďalšou možnosťou ako zrušiť chránený mód programu, je jeho ukončenie. Program sa totiž v chránenom móde nachádza len do jeho skončenia. Pri ďalšom spustení je program štandardne v normálnom móde (to neplatí, pokiaľ je v príkazovom riadku pri spustení programu zadaný chránený projekt, t.j. projekt, ktorý bol uložený v chránenom móde a pri načítaní ktorého, sa program prepne do chráneného režimu).

Chránený mód v Pg4uwMC

Program Pg4uwMC má chránený mód veľmi podobný ako program Pg4uw. Rozdiel je, že Chránený mód môže byť aktivovaný pomocou príkazu z menu, ale nemôže byť aktivovaný načítaním projektového súboru. Ďalší rozdiel je, že pre Pg4uwMC sa nastavenia Chráneného módu v Pg4uwMC ukladajú do konfiguračného súboru pri ukončení programu. Pri ďalšom štarte programu Pg4uwMC sa tieto nastavenia obnovia z konfiguračného súboru.

Príkazom menu - **Options / Protected mode** – sa zapína Chránený mód v programe Pg4uwMC. Po zvolení tohto príkazu sa zobrazí dialóg na zadanie hesla. Heslo sa zadáva dvakrát, aby bolo potvrdené, že je správne. Po úspešnom zadaní hesla sa program prepne do Chráneného módu.

Nastavenia Chráneného módu v Pg4uwMC ukladajú do konfiguračného súboru pri ukončení programu. Pri ďalšom štarte programu Pg4uwMC sa tieto nastavenia obnovia z konfiguračného súboru.

Špeciálna voľba - **Keep "Load project" operation allowed**, ktorá je štandardne nezaškrtnutá, slúži na zablokovanie tlačidla a príkazu menu Load project, pokiaľ je program v Chránenom móde.

Ak túto zvolíme túto voľbu (zaškrtneme) tak tlačidlo a príkaz menu Load project budú povolené v Chránenom móde.

Na prepnutie programu z Chráneného módu späť do Normálneho módu sa používa príkaz menu **Options / Normal mode**. Po jeho zvolení sa zobrazí okno požadujúce heslo. Užívateľ má zadať rovnaké heslo aké naposledy zadával, keď prepínal program do Chráneného módu.

Keď je program v Chránenom móde, na vrchu Log okna je zobrazený nadpis "Protected mode".

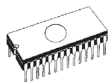
Poznámka: *Niekedy pri prechode z Chráneného módu do Normálneho módu sa môže stať, že niektoré príkazy (napríklad "Load project") zostanú nepovolené. Táto situácia môže byť vyriešená kliknutím na tlačidlo **Stop ALL**.*

Multi-projects

Multi-project je špeciálna vlastnosť, ktorá bola vyvinutá na uľahčenie programovania **multichip obvodov**.

Multichip obvod je obvod, ktorý obsahuje dva alebo viac nezávislých obvodov v jednom púzde.

Sub-device – Sub-obvod - jedna nezávislá časť multichip obvodu. Sub-device je vybrateľný zo zoznamu podporovaných obvodov programu Pg4uw. Keď je zvolený, dá sa s ním normálne pracovať. Je možné definovať, testovať a ukladať projekt pre tento sub-obvod.



Master device – Master obvod – je multichip obvod, ktorý obsahuje sub-devices. Master device je tiež vybrateľný zo zoznamu podporovaných obvodov programu Pg4uw. Po výbere sa môže použiť Multi-project Wizard na urobenie Multi-project súboru z jednotlivých projektových súborov a potom sa môže uložiť, načítať a vykonať.

Multi-project file – multi-projekt súbor – je špeciálny súbor, ktorý obsahuje užívateľom vybrané projekty. Multi-project súbor môže obsahovať jeden alebo viac projektov. Projekty obsiahnuté v Multi-project sa tiež volajú sub-projects.

Project file – projektový súbor – je špeciálny súbor, ktorý obsahuje dáta buffera, nastavenia práce s obvodom, špeciálne nastavenia a niečo z ochranných nastavení. Kompletne definuje postupnosť práce s obvodom. Keď sa raz uloží, môže sa kedykoľvek načítať a operácie budú kompletne zopakované.

Multi-project Wizard – je pomocník na urobenie Multi-project súboru. Wizard umožňuje užívateľovi vybrať projekty, ktoré majú byť v Multi-project a uložiť ich do jedného Multi-project súboru. Proces ukladania vybraných projektových súborov do jedného Multi-project súboru sa volá skladanie Multi-project súboru. Wizard tiež umožňuje spustiť operácie s obvodom v zhode s projektami obsiahnutými v Multi-project.

Multi-project Wizard

Aby sa dala funkcia Multi-project použiť, musí sa vytvoriť Multi-project súbor, ktorý obsahuje projekty zodpovedajúce jednotlivým čipom multichip obvodu. Multi-project súbor môže byť vytvorený v Multi-project Wizard. Wizard má nasledovné hlavné funkcie:

- Výber sub-project a vytvorenie výsledného Multi-project súboru
- Načítanie existujúceho Multi-project súboru
- Spustenie práce s obvodom s daným Multi-project súborom

Poznámka: *Takýto výsledný Multi-project je potom možné načítať cez príkaz Load project v hlavnom okne Pg4uw alebo v Multi-project Wizardovi.*

Multi-project Wizard obsahuje nasledovné ovládacie prvky:

- **Tlačítko Load multi-prj** - Slúži na načítanie existujúceho súboru multi-projektu.
- **Tlačítko Build Multi-project** - Uloží Multi-project file s obsahom projektov definovaných v tabuľke Sub-projects.
- **Tabuľka Table 1: Sub-projects** - Obsahuje zoznam projektov, ktoré majú byť súčasťou Multi-projectu.
- **Tlačítko Add project** - Slúži na pridanie ďalšieho projektu do zoznamu vybraných projektov uvedených v Table 1.
- **Tlačítko Remove project** - Slúži na odobratie vybraného projektu z tabuľky projektov Table 1.
- **Tlačítka Move up a Move down** - Tlačítka slúžia na presunutie vybraného projektu v tabuľke o 1 pozíciu vyššie, alebo nižšie.
- **Tlačítko Help** - Zobrazí tento help.
- **Tlačítka akcií na obvode Blank, Verify, Program, Erase** - Tlačítka slúžia na spustenie vybranej akcie na všetkých čipoch (projektoch) uvedených v tabuľke Sub-projects. V rámci vybranej akcie sa vykoná zvolená akcia postupne na všetkých čipoch (sub-device-och) master obvodu.

Popíšeme si dve základné etapy súvisiace s Multi-projectami:

- Vytvorenie Multi-projektu

- Použitie Multi-projektu na programovanie Multichip obvodu

Vytvorenie Multi-projektu

Nasledovné kroky sú odporúčané pre tvorbu Multi-project súboru.

- Vytvoriť projekty pre jednotlivé čipy multichip obvodu.
 - Projekty sa vytvárajú v programe Pg4uw rovnako ako projekty pre klasické obvody:
 - vybrať sub-device prislúchajúci čipu master obvodu *1
 - nastaviť parametre obvodu a načítať do buffera požadované data
 - vykonať testovanie vykonaním akcie na obvode
 - keď všetko sedí, potom príkazom Save project uložiť nastavenia obvodu do projektu
- Vybrať Master obvod, pre ktorý sa má vytvoriť multi-project. Po výbere multichip obvodu sa automaticky zobrazí Multi-project Wizard.
- V Multi-project Wizardovi tlačítkom Add project vybrať vytvorené projekty zodpovedajúce jednotlivým sub-devices.
- Po skončení výberu projektov spustiť Build Multi-project, ktorý po zadaní mena výsledného multi-projectu vytvorí Multi-project file obsahujúci všetky projekty (sub-projekty) z tabuľky Sub-projects v Wizardovi.

Poznámky: Je možné vytvárať Multi-project z ľubovoľného klasického projektu. Teda asociácia s Master device nie je povinná. Je len na užívateľovi ako skombinuje správne sub-device do jedného Multi-project. Táto funkcia môže byť špeciálne užitočná pri ISP programovaní viacerých obvodov zreťazovaných v JTAG reťazi s rôznymi projektami.

Multi-project Wizard môže byť spustený jednou z nasledovných akcií:

- Výberom Master device z dialógu Výber obvodu v Pg4uw
- Načítaním Multi-project súboru
- Otvorením dialógu Multi-project wizard priamo z Pg4uw príkazom menu Nastavenia / Asistent Multi projekt

*1Pravidlá pre názvy Master-device a Sub-device obvodov v zozname podporovaných obvodov Pg4uw:

Master-device: Multichip_originálny_názov [púzdro_typ]

Sub-devices: Multichip_originálny_názov [púzdro_typ] (časť1)

Multichip_originálny_názov [púzdro_typ] (časť2)

...

Multichip_originálny_názov [púzdro_typ] (časť n)

Príklad:

Master-device: TV0057A002CAGD [FBGA107]

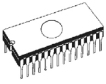
Sub-devices #1 TV0057A002CAGD [FBGA107] (NAND)

#2 TV0057A002CAGD [FBGA107] (NOR)

Použitie Multi-projektu na programovanie Multichip obvodu

Typický postup použitia existujúceho Multi-projectu je nasledovný.

- Pre single programovanie programom Pg4uw:
 - Načítať Multi-project file cez príkaz Load project v hlavnom okne Pg4uw alebo v Multi-project Wizardovi. Po úspešnom načítaní Multi-project je vždy zobrazený Multi-project Wizard
 - Spustiť požadovanú akciu na obvode niektorým z tlačidiel operácie na obvode. Najčastejšie to bude zrejme akcia Program.
 - Zvolená akcia sa vykoná postupným spracovaním projektov zahrnutých v Multi-project. Výsledkom je, že pre každý sub-device definovaný sub-projectom z Multi-project sa daná akcia vykoná. Ide o zautomatizovanie programovania viacerých sub-devices obvodov jediným spustením. Dôsledkom tohto postupu je, že indikácia priebehu akcie



na obvode začína pre každý sub-device od 0, preto priebeh akcie v Info okne "skáče" na 0 pri začatí programovania každého sub-device.

- Po naprogramovaní všetkých sub-devices (alebo i v prípade výskytu chyby) sa zobrazí klasické "Repeat" okno. V tej chvíli je možné obvod vybrať a vložiť nový obvod. Potvrdením okna "Repeat" alebo tlačítkom YES! na programátore * sa spustí nová sekvencia programovania sub-devices multichip obvodu. Pokiaľ je zvolená funkcia Automatic YES!, tak sa namiesto okna Repeat zobrazí po skončení akcie okno zobrazujúce stav päťice programátora s výzvou na výmenu obvodu. Po výbere obvodu a vložení nového obvodu sa automaticky spustí nová sekvencia programovania multichip obvodu. bližšie informácie a funkciu Automatic YES! nájdete v Programmer / Automatic YES!.
- Pre multi-programovanie programom Pg4uwMC alebo na standalone programátore:
 - Načítať Multi-project file cez menu Load project
 - Spustiť požadovanú akciu na obvode. Najčastejšie to bude zrejme akcia Program.
 - Zvolená akcia sa vykoná postupným spracovaním projektov zahrnutých v Multi-project. Výsledkom je, že pre každý sub-device definovaný sub-projektom z Multi-project sa daná akcia vykoná. Ide o zautomatizovanie programovania viacerých sub-devices obvodov jediným spustením.
 - Dôsledkom tohto postupu je, že indikácia priebehu akcie na obvode začína pre každý sub-device od 0, preto priebeh akcie v Status tabuľke "skáče" na 0 pri začatí programovania každého sub-device.
 - Po naprogramovaní všetkých sub-devices (alebo i v prípade výskytu chyby) sa zobrazí informácia o skončení akcie. V tej chvíli je možné obvod vybrať a vložiť nový obvod. Potvrdením tlačítka akcie alebo tlačítkom YES! na programátore * sa spustí nová sekvencia programovania sub-devices multichip obvodu. Pokiaľ je zvolená funkcia Automatic YES!, tak spustenie akcie je automatizované. Po výbere obvodu a vložení nového obvodu sa automaticky spustí nová sekvencia programovania multichip obvodu. bližšie informácie a funkciu Automatic YES! nájdete v Programmer / Automatic YES!.

Poznámky:

- zatiaľ nejde serializácia v režime multiprogramovania (v single programovaní serializácia funguje)
- nefunguje funkcia Count-down

Nastavenia / Uložiť nastavenia

Tento príkaz umožňuje nahráť všetky nastavenia programu, ktorých uloženie je podporované v súčasnosti. Nastavenia sú uložené aj keď je vypnuté automatické ukladanie nastavení pri skončení programu. Nastavenia, ktoré budú nahrávané: nastavenia v menu Nastavenia, desať posledne vybraných obvodov, poloha a rozmery hlavného okna programu, história posledne načítavaných a ukladaných súborov.

Pomocník

Stlačením klávesy <F1> spustíte obsah Pomocníka. Ak vyberáte nejakú položku z menu a vtedy stlačíte <F1>, zobrazí sa kontextovo citlivý pomocník. Počas činnosti s programátorom program nereaguje na stlačenie klávesy <F1>.

Nasledujúce položky Pomocníka sú zvýraznené:

- slová označujúce klávesy, o ktorých informuje aktuálna nápoveda

- všetky ostatné, významom dôležité slová
- aktuálna krížová referencia, kliknutím na ňu dostanete o nej ďalšie informácie.

Pretože obsah Pomocníka sa mení spolu s ovládacím programom, môže sa stať, že obsahuje informácie, ktoré nie sú uvedené v tomto návode.

Podrobnejšie informácie o jednotlivých položkách menu môžete nájsť v integrovanom on-line Pomocníkovi.

Poznámka: *Prípadné odchýlky, vylepšenia atď. tohto návodu sú na www.elnec.sk.*

Pomocník / Podporované obvody

Tento príkaz zobrazí zoznam všetkých obvodov, ktoré sú podporované aspoň jedným z podporovaných programátorov.

Prefix "g_" pred menom obvodu znamená, že daný obvod je podporovaný multinásobným programátorom.

Pomocník / Podporované programátory

Týmto príkazom získate informácie o programátoroch podporovaných ovládacím programom.

Pomocník / Zoznam podporovaných obvodov (aktuálny programátor)

Tento príkaz vygeneruje zoznam obvodov, ktoré sú podporované aktuálnym programátorom a uloží ho v textovej forme do súboru `?????DEV.TXT` a vo formáte HTML do súborov `?????DEV.HTM` v adresári, kde sa nachádza ovládací program .

Pomocník / Zoznam podporovaných obvodov (všetky programátory)

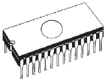
Tento príkaz vygeneruje zoznamy podporovaných obvodov pre všetky programátory a uloží ich v textovej forme do súborov `?????DEV.TXT` a vo formáte HTML do súborov `?????DEV.HTM` v adresári, kde sa nachádza ovládací program . Znak `?????` sú nahradené skrátenými názvami jednotlivých programátorov, pre ktoré sa zoznamy vytvárajú.

Poznámka: *Po vykonaní príkazu stratí program všetky informácie o aktuálnom obvode. Treba ho znovu vybrať použitím niektorej z metód pre výber obvodu.*

Pomocník / Zoznam podporovaných obvodov (krížová referencia)

Tento príkaz vygeneruje zoznam všetkých obvodov, ktoré sú podporované programátormi dostupnými na trhu a podporovanými aktuálnym ovládacím programom pre programátory (ďalej už len ovládací program). Výsledný zoznam má HTML formát a skladá sa z nasledujúcich súborov:

- jeden hlavný súbor `TOP_DEV.htm` obsahuje výrobcov podporovaných obvodov
- ďalšie HTML súbory obsahujúce zoznamy podporovaných obvodov jednotlivých výrobcov



Hlavný HTML súbor je umiestnený v adresári, v ktorom sa nachádza tento ovládací program.

Ďalšie HTML súbory sú umiestnené do podadresára **DEV_HTML**, ktorý je umiestnený do adresára, v ktorom sa nachádza ovládací program.

Programátor / Vytvoriť report správu o probléme

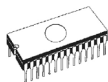
Príkaz Vytvoriť report správu o probléme sa používa na vygenerovanie diagnostických informácií (nastavenia obvodu, programátora apod.) do Log okna a skopírovanie aktuálneho obsahu celého Log okna do schránky, odkiaľ je možné vložiť obsah Log okna do ľubovoľného textového editora. Diagnostický výpis je vhodné použiť vtedy, ak sa v ovládacom programe alebo na programátore vyskytla chyba, ktorú je potrebné riešiť s výrobcom programátora. V takom prípade je pri posielaní správy o chybe od zákazníka výrobcovi dobré poslať aj diagnostický výpis, čo môže pomôcť skôr lokalizovať príčinu chyby a následne ju odstrániť.

Pomocník / O programe

Tento príkaz vygeneruje okno obsahujúce základné informácie o verzii programu, "copyright" a adresu dodávateľa.



Pg4uwMC



Program **Pg4uwMC** sa používa na riadenie práce viacerých programátorov alebo multiprogramátorov pripojených k jednému PC cez USB, ktoré takto tvoria concurrent programovací systém (obvody sa programujú paralelne, ale nezávisle jeden od druhého).

Pg4uwMC slúži na jednoduché monitorovanie veľkoobjemového programovania. Uživatelsky prívetivé prostredie Pg4uwMC kombinuje množstvo funkcií s jednoduchým používaním a poskytuje prehľad všetkých dôležitých činností a výsledkov bez zaťažovania obsluhy nepodstatnými detailmi.

Pg4uwMC používa projektový súbor na riadenie multiprogramovacieho systému. Projektový súbor obsahuje užívateľské data, nastavenia programovania, nastavenia pre programovaný obvod, nastavenie programovacej postupnosti atď. Chyby obsluhy programovania sú minimalizované, pretože projekt je vytvorený a odskúšaný návrhom a až potom odovzdaný obsluhu. Na zamedzenie nechcenej zmeny projektového súboru, môže projekt nastaviť aj prepnutie ovládacieho programu do chráneného módu.

Každý obvod môže byť programovaný s rôznymi dátami – sériové číslo, konfigurácia a kalibrácia.

Program Pg4uwMC pozostáva z nasledovných hlavných okien:

- Hlavné okno
- Okno pre nastavenia
- Okno „Search for programmers“

Užívateľská obrazovka Pg4uwMC

Site #1 | **Site #2** | **Site #3** | **Site #4**

100% | 81% | 22% | 0%

| Site | Status | Serialization |
|------|----------------------------|---------------|
| #1 | Programming device - O.K. | |
| #2 | Programming device ... 81% | |
| #3 | Programming device ... 22% | |
| #4 | Erasing device ... 0% | |

Log window:
L0124: Device blank checking ...
L0125: Programming device ...
L0126: Verifying device with buffer ...
L0127: Programming device - O.K.
L0128: Elapsed time: 0:00:47.1
L0129: ===== SITE #1 =====
L0130: ==> 03.04.2007, 10:15:08
L0131: Programming device: Macronix MX29F1615.
L0132: Device insertion test ...
L0133: Checking device ID ...
L0134: Erasing device ...
L0135: Device blank checking ...
L0136: Programming device ...
L0137: Verifying device with buffer ...
L0138: Programming device - O.K.
L0139: Elapsed time: 0:00:45.2

Základné okno Pg4uwMC

Základné okno Pg4uwMC obsahuje nasledovné časti:

Menu a tlačidlá nástrojov

Cez menu a tlačidlá nástrojov je prístupných veľa funkcií Pg4uwMC.

Tlačidlo Settings

Kliknutím na toto tlačidlo sa otvorí okno Pg4uwMC Settings, ktoré je opísané nižšie.

Panely Site #1, Site #2,...

Panely informujú o:

- výber programovacieho modulu
- činnosti programovacieho modulu
- stav akcie na obvode a/alebo výsledok

Každý panel tiež obsahuje tlačidlá **Run** a **YES!**, používané na začatie operácie s obvodom.

Box Statistics

Box Statistics informuje o množstve dobre, zle a celkovo naprogramovaných obvodov.

Checksum

Checksum je jednoduchá kontrolná suma dát načítaných z projektového súboru.

Panel Status window

Panel Status window informuje o aktuálnom stave každého modulu. Stav môže byť:

Blank modul nie je aktívny

Ready modul je aktívny a pripravený na prácu. Programátor je pripojený. Neprebíha žiadna akcia s obvodom.

and other information aktuálne bežiaci operácia s obvodom, výsledok, stav pripájania programátora atď...

Log window na pravej strane Status window

Log window obsahuje informácie o pripájaní a odpájaní programátorov (modulov), výsledky akcií s obvody a ďalšie informácie

Tlačidlo Connect programmers

Tlačidlo sa používa na pripojenie vybratých programovacích modulov. Toto tlačidlo sa obvyčajne používa hneď na začiatku po štarte Pg4uwMC.

Tlačidlo Disconnect programmers

Tlačidlo sa používa na odpojenie pripojených programovacích modulov a ukončenie ovládacích programov pre jednotlivé moduly. Odpojenie je možné realizovať jedine vtedy, keď na žiadnom programovacom module nebeží akcia s obvodom.

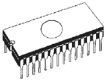
Tlačidlo Run <operation>

Tlačidlo sa používa na začatie akcie s obvodom na všetkých programovacích moduloch v tom istom čase. Pod pojmom <operation> sa myslia nasledovné operácie: Program, Verify, Blank check, Erase.

Tlačidlo Stop ALL

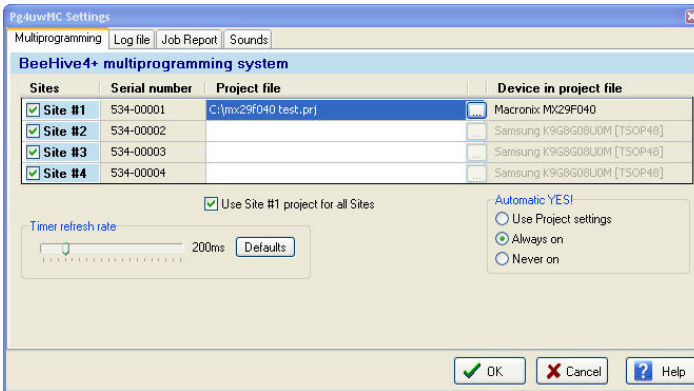
Tlačidlo sa používa na ukončenie akcie prebiehajúcej s obvodom na všetkých pripojených programovacích moduloch.

Tlačidlo Help



Tlačidlo zobrazí Pomocníka pre Pg4uwMC.

Okno Pg4uwMC Settings



Okno Pg4uwMC Settings slúži na nastavenie rôznych volieb:

- tabuľka obsahujúca informácie / nastavenia pre programovacie sity: poradové číslo, sériové číslo, priradený projektový súbor.
- zaškrťavacie políčko Use Site #1 project for all Sites
- panel Automatic YES!
- panel Timer refresh rate settings
- panel Log file settings

Tabuľka obsahujúca informácie a nastavenia pre programovacie moduly

Na vrchu je tabuľka obsahujúca 3 stĺpce :

- stĺpec SITES obsahuje zaškrťavacie políčka s číslom modulu #1, #2,...#8, ktorým sa povoľuje/zakazuje použitie programovacieho modulu. Na 1 PC môže byť naraz pripojených až 8 programovacích modulov.
- stĺpec Serial number obsahuje sériové čísla programovacích modulov
- stĺpec Project file obsahuje editovateľné riadky, do ktorých sa zadáva adresa projektového súboru pre programovacie moduly #1, #2, #3, #4. Projektové súbory môžu byť vybrané aj pomocou dialógu Select project file, ktorý sa otvorí kliknutím na tlačidlo „...“ umiestnené na pravej strane riadku. Ak je tento riadok prázdny, nenačíta sa žiaden projektový súbor.

Zaškrťavacie políčko Use Site #1 project for all Sites

Zaškrťavacie políčko Use Site #1 project for all Sites je umiestnené pod stĺpcom Project file.

Keď je požiadavka na programovanie rovnakých obvodov s rovnakými dátami, zaškrťavacie políčko by malo byť zaškrtnuté.

- Ak je zaškrťavacie políčko zaškrtnuté, projektový súbor vybraný pre modul #1 bude použitý aj pre ostatné moduly. V tomto móde všetky programovacie moduly používajú rovnaký zdieľaný buffer dát a programuje sa ten istý typ obvodu.
- Ak zaškrťavacie políčko nie je zaškrtnuté, každý programovací modul používa vlastný projektový súbor určený v stĺpci Project file pre každý modul. V tomto móde každý programovací modul používa vlastný buffer dát, čo umožňuje programovať v každom module rôzne obvody rôznymi dátami.

Panel Automatic YES!

V poli Automatic YES! Si môžete vybrať nastavenie tejto vlastnosti:

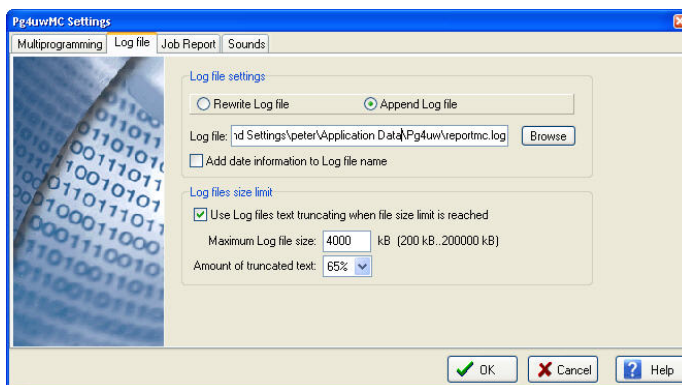
- nastavenie podľa projektu
- vždy zapnuté
- vždy vypnuté

Pre viac informácií o Automatic YES! pozrite **Programmer / Automatic YES**.

Panel Timer refresh rate

Timer refresh rate definuje ako často bude Pg4uwMC žiadať od programovacích modulov informácie: aktuálne prebiehajúcu akciu s obvodom, priebeh akcie, jej výsledok a pod. Tieto informácie sú potom zobrazované v hlavnom okne Pg4uwMC. Prednastavená hodnota je 200ms. Ak chcete častejšie obnovovať stavové informácie, zvolte kratší čas. Ak pri tom spozorujete pokles výkonu systému, nastavte dlhší čas. Na počítačoch triedy Pentium 4 nie je skoro žiadne zníženie výkonu pri častejšom obnovovaní informácií, ale na pomalších počítačoch je vhodné nastaviť dlhší čas.

Panel Log file settings slúži na nastavenie správania Log súboru.

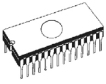


Log súbor je textový súbor, ktorý obsahuje informácie o celom procese práce s obvodom a Pg4uwMC, t.j. obsahuje informácie o projektových súboroch, akciách s obvodom i ich výsledkami. Multiprogramovací systém generuje niekoľko Log súborov. Jeden hlavný pre Pg4uwMC a pre každý aktívny programovací modul ďalší Log súbor, t.j. každý programovací modul má vlastný Log súbor. Meno Log súborov pre programovacie moduly sa skladá z názvu špecifikovaného v editovateľnom riadku Log file, za ktorým nasleduje číslo modulu vo formáte `_#<Snum>`.

Príklad:

Meno Log súboru špecifikovaného užívateľom je „report.log“. Mená Log súborov budú: Pg4uwMC Log súbor „report.log“

Modul #1 Log súbor „report_#1.log“
Modul #2 Log súbor „report_#2.log“
Modul #3 Log súbor „report_#3.log“
Modul #4 Log súbor „report_#4.log“
atď...



Nasledujúce voľby môžu byť nastavené pre tvorbu log súboru.

- **Append Log file** zapne vytváranie Log súboru. Log súbor bude vytvorený po prvom reštarte Pg4uwMC. Pri každom ďalšom štarte Pg4uwMC predchádzajúce dáta ostanú zachované v súbore a nové dáta sa do súboru doplnia.
- **Rewrite Log file** zapne vytváranie Log súboru. Log súbor bude vytvorený po prvom reštarte Pg4uwMC. Pri každom ďalšom štarte Pg4uwMC bude existujúci Log súbor prepísaný novým Log súborom a predchádzajúce dáta budú vymazané.

Zaškrtávacie políčko **Add date information to Log file name** umožňuje užívateľovi vložiť dátum do názvu Log súboru, ktorý je špecifikovaný v riadku Log file name. Ak si užívateľ zvolí túto voľbu, program automaticky vloží do názvu súboru reťazec reprezentujúci aktuálny dátum podľa nasledovných pravidiel:

Ak užívateľom špecifikovaný názov Log súboru má tvar

<užívateľské_meno_Log_súboru>.<prípona_Log_súboru>

Názov Log súboru s pridaným dátumom bude:

<užívateľské_meno_Log_súboru><rrrr-mmm-dd>.<prípona_Log_súboru>

Nová časť reprezentuje dátum a obsahuje: rrrr – rok, mmm – mesiac, dd – deň

Príklad: *užívateľské meno Log súboru:*

c:\logs\myfile.log

Meno s pridaným dátumom bude (dátum: 7.11.2006):

c:\logs\myfile-2006-nov-07.log

Ak chcete mať názov Log súboru tvorený iba dátumom, môžete to urobiť nasledovne:

.<prípona_Log_súboru> bodka bude prvá v názve súboru, t.j. nezadáte meno

Príklad: *užívateľské meno Log súboru:*

c:\logs\log

Meno s pridaným dátumom bude (dátum: 7.11.2006):

c:\logs\2006-nov-07.log

Panel Log file size limit

- Zaškrtávacie políčko **Use Log file text truncating when file size limit is reached** slúži na nastavenia limitu veľkosti Log súboru. Keď je zaškrtnuté, pri dosiahnutí nastavenej veľkosti bude časť textu vymazaná a súbor potom bude mať definovanú veľkosť. Ak táto voľba nie je zaškrtnutá, veľkosť Log súboru je obmedzená len veľkosťou voľného miesta na disku.
- Voľba **Maximum Log file size** určuje maximálnu veľkosť Log súboru.
- Voľba **Amount of truncated text specifies** určuje koľko percent súboru bude vymazané pri dosiahnutí limitu veľkosti Log súboru.

Záložka **Job Report** slúži na nastavenie módu Job Report súboru.

Job Report celkovo opisuje priebeh celej práce na obvode – od načítania projektu až po koniec práce s daným typom obvodu.

Job Report obsahuje nasledovné informácie:

- názov projektového súboru
- dátum vytvorenia projektového súboru
- stav Protected módu
- verziu Pg4uwMC
- typ programátora a jeho sériové číslo
- čas, kedy začala práca, t.j. čas kedy bol načítaný projekt, ktorý sa bude vykonávať
- čas, kedy skončila práca, t.j. čas vytvorenia Job Report súboru
- názov obvodu
- trieda obvodov

- kontrolná suma
- nastavenia súvisiace s obvodom
- informácie o serializácii
- štatistické informácie

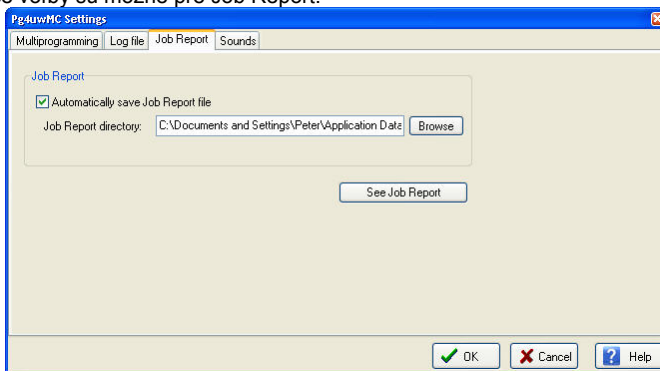
Job Report bude vytvorený v nasledovných prípadoch:

- pri zvolení príkazu Load project
- pri zvolení príkazu disconnect programmers (programmer site)
- ukončení programu Pg4uwMC
- keď počítadlo Count down dosiahne „0“ (koniec úlohy)
- pri zvolení príkazu File / Make Job Report

Job Report je generovaný pre aktuálne načítaný projekt iba vtedy, keď hodnota Total v paneli Statistics je väčšia ako 0, t.j. musela byť uskutočnená aspoň jedna akcia s obvodom (programovanie, verifikácia...).

Nastavenia pre Job Report sa nachádzajú na záložke Job Report v okne Pg4uwMC Settings (menu Options / Settings).

Nasledujúce voľby sú možné pre Job Report:



Keď je zaškrtnuté políčko Automatically save Job Report file, Job Report bude automaticky ukladaný do adresára špecifikovaného v riadku Job Report directory a bude mať názov vytvorený nasledovne:

job_report_<ordnum>_<prjname>.jrp

kde

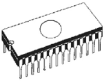
<ordnum> je decimálne číslo súboru. Ak existuje viacero súborov s rovnakým menom, potom ordnum bude zväčšené o počet takýchto súborov

<prjname> je názov aktuálne používaného projektu, ale bez prípony

Príklad 1: *Nech názov projektu je c:\myproject.eprj a adresár pre ukladanie Job Report je d:\jobreports\. V adresári nie je žiaden Job report súbor.*

Meno Job reportu bude: d:\job_reports\job_report_000_myproject.jrp

Príklad 2: *Nech názov projektu je c:\myproject.eprj a adresár pre ukladanie Job Report je d:\jobreports\. V adresári je Job Report: job_report_000_myproject.jrp.*



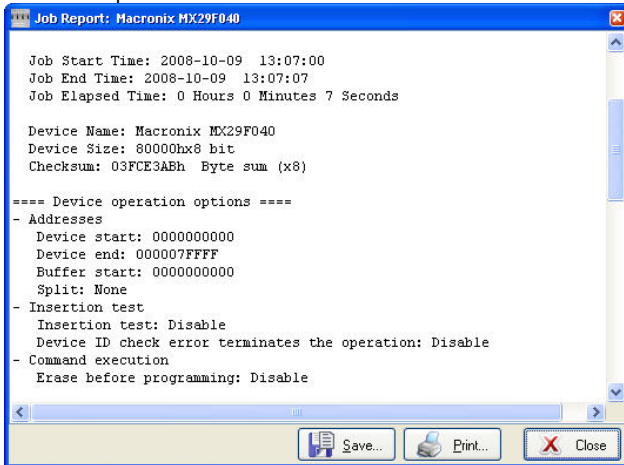
Meno Job reportu bude: `d:\job_reports\job_report_001_myproject.jrp`

Všimnite si, že číslo v názve súboru sa zmenilo o 1.

Keď je nastavené **Automatically save Job Report file** pri ukladaní Job Reportu sa nezobrazí žiaden dialóg ani správa (ak nenastane chyba počas ukladania súboru).

Ak nie je nastavené **Automatically save Job Report file** potom pri ukladaní súboru Pg4uwMC ukáže dialógové okno.

V tomto dialógovom okne si užívateľ môže zvoliť, čo chce urobiť s Job Reportom. Ak si užívateľ nevyberie žiadnu akciu, klikne na Cancel Job Report bude uložený iba do Pg4uwMC Log okna. Príklad Job Reportu:



Okno "Search for Programmers"



Dialóg umožňuje vyhľadať všetky pripojené programátory cez USB port a vybrať multiprogramovací systém. Po skončení vyhľadávania sa v okne Search result zobrazia

nájdené programátory. Pre niektoré multiprogramovacie systémy užívateľ môže zmeniť počet programovacích modulov alebo vylúčiť nechcené programovacie moduly. Keď je nájdený aspoň jeden programátor, tlačidlo Accept sa stane aktívnym a užívateľ kliknutím na neho akceptuje nové nastavenie.

Parametre príkazového riadku

Program Pg4uwMC podporuje nasledovné parametre príkazového riadku:

/prj:<názov súboru>

Načíta projektový súbor. Parameter <názov súboru> znamená celú alebo relatívnu cestu a názov. Z príkazového riadku môžete načítať projektový súbor zadáním mena súboru aj bez predpony /prj

Priklad:

Pg4uwMC.exe c:\projects\myproject.eprj

Načítanie projektu "c:\projects\myproject.eprj".

Programátory podporované Pg4uwMC

Zoznam aktuálne podporovaných programátorov sa zobrazí v Pg4uwMC pomocou príkazu menu Help / Supported programmers. Zjednodušene, programátory podporované Pg4uwMC sú 48 pinové programátory s USB rozhraním. Podporované sú tiež aj systémy násobného programovania pripojené cez USB sú. Pg4uwMC môže riadiť od 1 to 8 programovacích site. Jeden programovací site predstavuje jednu ZIF päticu.

Ak sa vyskytnú problémy

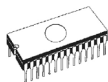
Komunikačné chyby počas vyhľadávania programátorov

Ak nastanú komunikačné chyby, prosím zatvorte všetky Pg4uw aplikácie a Pg4uwMC. Potom spustíte Pg4uwMC znova a kliknite na „Connect programmers“. Tým sa spustia Pg4uw pre každý programovací modul a opätovne sa pripoja programovacie moduly.

Všetky programátory sú správne pripojené, ale práca je nestabilná

Ak sa počas práce s obvody náhodne stráca komunikácia s programátormi, prosím ukončíte ostatné programy. Hlavne tie, ktoré sú náročné na systémové prostriedky (multimédia, CAD, grafické aplikácie atď.).

Poznámka: *Doporučujeme používať USB porty umiestnené na zadnej strane počítača, ktoré sú priamo pripojené ku matičnej doske. USB porty pripojené pomocou káblov môžu byť nespoľahlivé pri USB 2.0 vysokorychlostných prenosoch. Toto odporúčanie platí pre všetky zariadenia, nie len pre programátory.*



Spoločné poznámky

Starostlivosť o programátor

Na dosiahnutie bezproblémového používania programátora po dlhý čas odporúčame dodržiavať nasledovné pokyny a upozornenia.

Starostlivosť o programátor sa rozlišuje podľa charakteru a intenzity používania. Pre všetky prípady však platia aj všeobecné odporúčania:

- Nepoužívajte a nenechávajte programátor v prašnom a vlhkom prostredí.
- Vlhkosť zvyšuje rýchlosť zašpinenia pinov ZIF päťice programátora od pinov používaných obvodov a od iných vplyvov pracovného prostredia.
- Po skončení práce prikryte ZIF päťicu programátora priloženým protiprachovým krytom.
- Nepoužívajte a nenechávajte programátor v blízkosti zdrojov tepla a na priamom slnečnom svetle.

Každodenné intenzívne používanie (programovacie centrum, výroba)

Denná starostlivosť

Skontrolujte stav ZIF päťice programátora a používaných konvertorov, ich zašpinenie a opotrebovanie. Pomocou čistého, suchého stlačeného vzduchu z päťice odstráňte prach, špinu a pozostatky po vývodoch obvodov. Čistite päťicu v otvorenom aj zatvorenom stave.

Týždenná starostlivosť

Vykonajte test programátora+ pre každý programátor alebo programovací modul.

Štvrťročná starostlivosť

Mäkkou handričkou navlhčenou isopropyl alkoholom alebo technickým liehom jemne poutierajte povrch programátora. LCD display môže byť utretý iba handričkou navlhčenou vodou. Isopropyl alkohol môže poškodiť povrch LCD display-a.

Ak to programátor podporuje, vykonajte test kalibrácie.

Každodenné používanie (vývoj)

Denná starostlivosť

Po skončení práce prikryte ZIF päťicu programátora priloženým protiprachovým krytom. ZIF päťice konvertorov taktiež chráňte pred prachom a iným znečistením.

Mesačná starostlivosť

Skontrolujte stav ZIF päťice programátora a používaných konvertorov, ich zašpinenie a opotrebovanie. Pomocou čistého, suchého stlačeného vzduchu z päťice odstráňte prach, špinu a pozostatky po vývodoch obvodov. Čistite päťicu v otvorenom aj zatvorenom stave.

Štvrťročná starostlivosť

Vykonajte test programátora+ pre každý programátor alebo programovací modul.

Polročná starostlivosť

Mäkkou handričkou navlhčenou isopropyl alkoholom alebo technickým liehom jemne poutierajte povrch programátora. LCD display môže byť utretý iba handričkou navlhčenou vodou. Isopropyl alkohol môže poškodiť povrch LCD display-a.

Ak to programátor podporuje, vykonajte test kalibrácie.

Občasné používanie

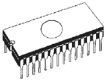
Denná starostlivosť

Po skončení práce prikryte ZIF päťicu programátora priloženým protiprachovým krytom. ZIF päťice konvertorov taktiež chráňte pred prachom a iným znečistením.

Štvrťročná starostlivosť

Skontrolujte stav ZIF päťice programátora a používaných konvertorov, ich zašpinenie a opotrebovanie. Pomocou čistého, suchého stlačeného vzduchu z päťice odstráňte prach, špinu a pozostatky po vývodoch obvodov. Čistite päťicu v otvorenom aj zatvorenom stave.

Polročná starostlivosť



Vykonajte test programátora+ pre každý programátor alebo programovací modul.

Ročná starostlivosť

Mäkkou handričkou navlhčenou isopropyl alkoholom alebo technickým liehom jemne poutierajte povrch programátora. LCD display môže byť utretý iba handričkou navlhčenou vodou. Isopropyl alkohol môže poškodiť povrch LCD display-a.

Ak to programátor podporuje, vykonajte test kalibrácie.

Upozornenie:

ZIF päťica programátora a konvertory púzdiar sú spotrebný materiál. ZIF päťica programátora má životnosť 25 000 mechanických cyklov (otvorenie a zatvorenie ZIF). ZIF päťice konvertorov majú väčšinou životnosť 10 000 cyklov, niektoré 5 000 a niektoré BGA ZIF 500000 mechanických cyklov. Na reálnu elektrickú životnosť ZIF päťice (kedy ešte ZIF päťica nespôsobuje chybovosť programovania) má však vplyv aj okolité prostredie, samotné obvody a starostlivosť o ZIF. Nedotýkajte sa kontaktov ZIF päťice prstami, pretože masnota a špina na prstoch môže spôsobiť zvýšenú chybovosť programovania. Ak Vám stúpne chybovosť programovania vymeňte ZIF päťicu programátora alebo konvertor.

Na zašpinené a opotrebované ZIF päťice, ktoré spôsobujú vysokú chybovosť programovania (alebo inej akcie na obvode) sa nevzťahujú záručné podmienky.

Software

Pg4uw je spoločný ovládací program pre všetky ELNEC programátory. Z toho dôvodu je možné pri práci s programom nájsť položky, ktoré sa netýkajú aktuálne vybratého programátora.

Niektoré špeciálne obvody (napr. rodina CoolRunner od Philips) vyžadujú externý DAT súbor, ktorý nie je štandardnou súčasťou programového vybavenia dodávaného na CD. Ak potrebujete programovať tieto obvody, môžete ich získať na internetovej adrese www.elnec.sk, časť Download.

Parametre príkazového riadku

Doporučujeme používať špeciálny program **Pg4uwcmd.exe** na riadenie Pg4uw parametrami z príkazového riadku. Pre zachovanie spätnej kompatibility je možné použiť niektoré parametre priamo s Pg4uw, lepšie je použiť **Pg4uwcmd.exe**, ktorý obsahuje viac parametrov a taktiež má schopnosť vrátiť ExitCode (alebo ErrorLevel) hodnotu indikujúcu úspech alebo chybu pri vykonávaní príkazu z príkazového riadku. Pre viac informácií o používaní programu **Pg4uwcmd.exe** pre **Pg4uw**, pozrite na kapitolu **Diaľkové ovládanie Pg4uw**.

Parametre príkazového riadku, ktoré môžu byť použité priamo s Pg4uw.

/Prj:<file_name> príkaz na vykonanie načítania projektu pri spustení programu alebo v prípade, že program už je spustený, <file_name> udáva plnú alebo relatívnu cestu k súboru projektu

/Loadfile:<file_name> príkaz na vykonanie načítania súboru pri spustení programu alebo v prípade, že program už je spustený, <file_name> udáva plnú alebo relatívnu cestu k súboru, formát súboru sa detekuje automaticky

Všimnite si, musia byť dodržané konvencie pre názvy súborov pod Windows. To tiež znamená, že keď názov súboru obsahuje medzery, parameter príkazového riadku musí mať názov súboru vo vnútri úvodzoviek.

Príklady:

```
/prj:c:\mojsubor.eprj
```

Načíta sa súbor c:\mojsubor.eprj

```
/loadfile:"c:\názov s medzerami.bin"
```

Načíta sa súbor "c:\názov s medzerami.bin" do buffera.

/Program[:switch] príkaz na spustenie akcie Programovať obvod pri spustení programu alebo v prípade, že program už je spustený, jeden z nasledujúcich nepovinných prepínačov môže byť použitý:

1. switch 'noquest' spôsobí spustenie akcie programovať obvod bez zobrazenia otázky o začatí programovania
2. switch 'noanyquest' spôsobí spustenie akcie programovať obvod bez zobrazenia otázky o začatí programovania, pričom po skončení operácie program nezobrazí otázku na opakovanie akcie, ale prejde priamo do hlavného okna programu

Príklady:

1. /Program

2. /Program:noquest

3. /Program:noanyquest

/Close možno použiť len spolu s príkazom /Program, pričom po úspešnom skončení akcie na obvode zatvorí program automaticky

/Close: always možno použiť len spolu s príkazom /Program, pričom po skončení akcie na obvode zatvorí program automaticky (bez ohľadu na skutočnosť, či akcia na obvode bola úspešná alebo neúspešná)

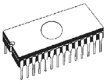
/Eprom_Flash_Autoselect[:xx] príkaz na vykonanie automatického výberu EPROM alebo FLASH podľa ID po spustení programu alebo v prípade, že program je už spustený. xx udáva počet pinov obvodu v ZIF (momentálne akceptované hodnoty sú 28 a 32) a je povinný len pre staršie programátory, ktoré nemajú možnosť "Insertion" testu a nedokážu zistiť počet pinov automaticky. Pre ostatné programátory sa tento údaj ignoruje.

Základné pravidlá používania výkonných parametrov príkazového riadku:

1. u parametrov príkazového riadku nie sú rozlišované malé a veľké písmena
2. parametre príkazového riadku môžu byť použité pri prvom spustení programu alebo aj počas behu programu
3. ak program už beží, tak príkazy z príkazového riadku sa vykonávajú len vtedy, keď program nie je zaneprázdnený vykonávaním nejakej akcie. Bežiaci program prijme parametre z príkazového riadku len v prípade, že je aktívne hlavné okno programu, žiadne dialógy a menu príkazy nesmú byť aktívne.
4. poradie vykonávania parametrov príkazového riadku v prípade súčasného použitia viacerých parametrov príkazového riadku je pevne definované nasledovne:
 1. Načítanie súboru (/Loadfile:...)
 2. Načítanie projektu (/Prj:...)
 3. EPROM/FLASH automatický výber
 4. Programovanie obvodu (/Program[:switch])
 5. Zatvorenie ovládacieho programu (/Close len spolu parametrom /Program)

Parametre príkazového riadku pre programátory pripojené prostredníctvom LPT portu:

/Axxx program pri spustení skontroluje prítomnosť programátora na porte so zadanou adresou xxx.



Príklad: /A3bc
/SPP nastaví jednosmernú komunikáciu medzi PC a programátorom

Parametre príkazového riadku pre štart programu Pg4uw v demo móde:

Demo mód je vhodný situáciách, keď nie je k dispozícii programátor. Sú dve možnosti ako program používať v demo móde. Prvá je stlačiť tlačítko Demo v dialógu "Hľadať programátor", druhá je použiť parameter /demo pri spustení programu.

Pg4uw.exe /demo /<meno programátora>

kde <meno programátora> má byť meno programátora tak, ako je zobrazované v programe Pg4uw.

Dial'kové ovládanie Pg4uw

Pg4uw môže akceptovať príkazy z príkazového riadka. Dial'kové ovládanie môže byť dosiahnuté pomocou týchto príkazov z príkazového riadku, ale oveľa efektívnejšie je používať špeciálny program **Pg4uwcmd.exe**, ktorý má mnohé výhody. Hlavnými výhodami sú vrátenie výsledkov a rýchlejšia odozva ako z príkazov z príkazového riadku.

Program Pg4uwcmd.exe môže byť použitý na nasledovné:

1. spustiť program Pg4uw so špecifikovanými parametrami
2. vnútiť parametre príkazového riadku bežiacemu programu Pg4uw

Veľmi dobrou vlastnosťou Pg4uwcmd.exe schopnosť vrátiť výsledok operácie s obvodom v Pg4uw.

Návratové hodnoty Pg4uwcmd.exe

Ak sú príkazy príkazového riadku v programe Pg4uw úspešne vykonané ExitCode (alebo ErrorLevel) Pg4uwcmd.exe je 0. V opačnom prípade je ExitCode 1 alebo viac.

Návratová hodnota programu Pg4uwcmd.exe môže byť testovaná v batch súboroch.

Nasledovné výkonné parametre príkazového riadku sú dostupné s Pg4uwcmd.exe

/Prj:<file_name> príkaz na vykonanie načítania projektu pri spustení programu alebo v prípade, že program už je spustený, <file_name> udáva plnú alebo relatívnu cestu k súboru projektu

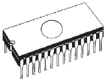
/Loadfile:<file_name> príkaz na vykonanie načítania súboru pri spustení programu alebo v prípade, že program už je spustený, <file_name> udáva plnú alebo relatívnu cestu k súboru, formát súboru sa detekuje automaticky

/Program[:switch] príkaz na spustenie akcie Programovať obvod pri spustení programu alebo v prípade, že program už je spustený, jeden z nasledujúcich nepovinných prepínačov môže byť použitý:

1. switch 'noquest' spôsobí spustenie akcie programovať obvod bez zobrazenia otázky o začatí programovania
2. switch 'noanyquest' spôsobí spustenie akcie programovať obvod bez zobrazenia otázky o začatí programovania, pričom po skončení operácie program nezobrazí otázku na opakovanie akcie, ale prejde priamo do hlavného okna programu

Príklady:

1. /Program
2. /Program:noquest
3. /Program:noanyquest



| Adresa Buffera | Data |
|----------------|------|
| FF000H | ABH |
| FF001H | CDH |
| FF002H | EFH |
| FF003H | 43H |
| FF004H | 21H |

`/writebufferex:INDEX:ADDR1:B11,B12,B13,B14,...,B1N[::ADDR2:B21,B22,B23,B24,...,B2M]`.
Príkaz `/writebuffer` sa používa na zapísanie bloku Bajtov do hlavného buffera Pg4uw na špecifikované adresy. Príkaz je veľmi podobný príkazu `/writebuffer`, až na jeden parameter – INDEX.

Parameter INDEX určuje poradie bufferov, do ktorých budú dáta poslané. Hlavný buffer má index '1'. Prvý sekundárny buffer má index '2', atď. Všimnite si, že sekundárny buffer (e) je dostupný iba pre niektoré obvody (napr. Microchip PIC16F628). Číslo buffera indexovaný parametrom buffindex závisí na poradí buffera v programe Pg4uw v dialógu Zobraziť/Editovať buffer. Napríklad obvod Microchip PIC16F628 má dodatočný buffer s názvom "Data EEPROM". Do tohto buffera sa dajú zapisovať dáta keď buffindex = 2

Príklad 1:

`/writebufferex:1:7FF800:12,AB,C5,D4,7E,80`

Príkaz je ekvivalentný príkazom

`/writebuffer:7FF800:12,AB,C5,D4,7E,80`

Opísaných v sekcii o príkaze `/writebuffer`.

Príklad 2:

`/writebufferex:2:2F:12,AB,C5,D4,7E,80`

Príkaz zapíše 6 Bytes 12H ABH C5H D4H 7EH 80H do sekundárneho buffera s indexom "2" na adresu 2FH.

Adresovanie vyzerá nasledovne:

Prvý Byte na najnižšiu adresu

| Adresa Buffera | Data |
|----------------|------|
| 00002FH | 12H |
| 000030H | ABH |
| 000031H | C5H |
| 000032H | D4H |
| 000033H | 7EH |
| 000034H | 80H |

Základné pravidlá používania výkonných parametrov príkazového riadku:

1. program Pg4uwcmd.exe musí byť volaný z rovnakého adresára ako program Pg4uw.
2. Ak program Pg4uw nebeží keď sa spustí program Pg4uwcmd.exe, ten sa spustí automaticky.
3. u parametrov príkazového riadku nie sú rozlišované malé a veľké písmena
4. parametre príkazového riadku môžu byť použité pri prvom spustení programu alebo aj počas behu programu
5. ak program už beží, tak príkazy z príkazového riadku sa vykonajú len vtedy, keď program nie je zaneprázdnený vykonávaním nejakej akcie. Bežiaci program prijme parametre z príkazového riadku len v prípade, že je aktívne hlavné okno programu, žiadne dialógy a menu príkazy nesmú byť aktívne.

6. poradie vykonávania parametrov príkazového riadku v prípade súčasného použitia viacerých parametrov príkazového riadku je pevne definované nasledovne:

1. Načítanie súboru (/Loadfile:...)
2. Načítanie projektu (/Prj:...)
3. EPROM/FLASH automatický výber
4. Programovanie obvodu (/Program[:switch])
5. Zatvorenie ovládacieho programu (/Close len spolu parametrom /Program)

Príklad 1:

```
Pg4uwcmd.exe /program:noanyquest /loadfile:c:\empfile.hex
```

Budú uskutočnené nasledovné operácie:

1. spustenie Pg4uw.exe (ak už nebeží)
2. načítanie súboru c:\empfile.hex
3. bez otázk sa začne programovanie obvodu
4. Pg4uwcmd.exe je spustený a periodicky kontroluje stav Pg4uw.exe
5. keď je dokončené programovanie obvodu, Pg4uwcmd.exe sa zatvorí a vráti ExitCode v závislosti na načítanom súbore výsledku programovania v Pg4uw.exe. Keď všetky operácie prebehli úspešne, Pg4uwcmd.exe vráti 0, v opačnom prípade vráti 1 alebo niečo iné.

Príklad 2:

```
Pg4uwcmd.exe /program:noanyquest /prj:c:\emproject.eprj
```

Operácie sú rovnaké ako v Príklade 1, iba operácia Load file bola nahradená operáciou Load project a načíta sa súbor c:\emproject.eprj.

Príklad 3:

Použitie Pg4uwcmd.exe v dávkovom súbore a testovanie návratovej hodnoty Pg4uwcmd.exe.

```
rem ----- začiatok dávkového súboru -----
@echo off
rem zavolanie programu s požadovanými parametrami
Pg4uwcmd.exe /program:noanyquest /prj:c:\emproject.eprj
rem Detekcia výsledku vykonania príkazu
rem Premenná ErrorLevel je testovaná, hodnota 1 alebo viac znamená, že nastala chyba
if ErrorLevel 1 goto FAILURE
echo Command line operation was successful
goto BATCHEND
:FAILURE
echo Command line operation error(s)
:BATCHEND
echo.
echo This is end of batch file (or continue)
pause
rem ----- koniec dávkového súboru -----
```

Príklad 4:

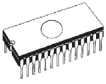
Predpokladajme, že program Pg4uw beží a je vybraný obvod. Potrebujeme načítať požadované dáta do buffera Pg4uw, uložiť nastavenia pre vybraný obvod a obsah buffera do projektového súboru. Požadované dáta sú uložené v súbore c:\15001-25001\file_10.bin. Projektový súbor bude uložený ako c:\projects\project_10.eprj.

Nasledovné príkazy príkazového riadka by mali byť vybrané na túto realizáciu:

```
Pg4uwcmd.exe /loadfile:c:\15001-25001\file_10.bin /saveproject:c:\projects\project_10.eprj
```

Kde Pg4uw prijíma príkazy, bude realizovaná nasledovná postupnosť:

1. načítanie súboru s dátami c:\15001-25001\file_10.bin



2. uloženie nastavení pre aktuálne vybraný obvod a obsahu buffera do projektového súboru c:\projects\project_10.eprj

Ak je výsledok vykonaných operácií OK, Pg4uwcmd vráti ExitCode (alebo ErrorLevel) hodnotu 0.

Ak sa vyskytnú nejaké chyby (nedá sa načítať súbor alebo uložiť projektový súbor), Pg4uwcmd vráti ExitCode hodnotu 1 alebo viac ako 1.

Poznámka: Keď sa používajú vyššie uvedené príkazy, užívateľ si musí byť istý, že Pg4uw nevykonáva žiadnu operáciu, napr. programovanie obvodu. Ak je Pg4uw zaneprázdnený, odmietne príkazy a vráti chybu (ExitCode je 1 alebo viac ako 1).

Hardware

Vzhľadom na ťažko vyčísliteľný počet druhov paralelných portov naozaj sa môže vyskytnúť prípad, keď sa naše zariadenie nevie s PC "dohodnúť". Ak na tento problém, prejavujúci sa neschopnosťou komunikácie medzi PC a programátorom, prípadne tiež nespoľahlivosťou komunikácie narazíte aj Vy, skúste, či sa rovnaký problém v spolupráci s programátorom prejavuje aj na iných PC, resp. na iných paralelných portoch vo vašom okolí.

Ak nie, zdokumentujte stav, teda presný popis konfigurácie Vášho PC (typ, výrobca, rýchlosť, operačný systém, rezidentné programy; výrobca, prípadne typ IO na Vašom paralelnom porte) a iné okolnosti, ktoré môžu mať súvis s daným problémom, a oboznámte s tým výrobcu. Prosím, použite formulár „**DEVICE PROBLEM REPORT**“

Upozornenie:

Poznámka pre triedu A ITE

Zariadenia opísané v tomto návode na použitie sú triedy A. V domácom prostredí môžu tieto zariadenia spôsobovať vysokofrekvenčnú interferenciu. V takom prípade sa môže požadovať, aby užívateľ vykonal adekvátne opatrenia.

Pretože BeeHive204 / BeeHive4+ a BeeProg2 / BeeProg+ majú zabudovaný napájací zdroj, dodržte nasledovné bezpečnostné opatrenia:

- Nadprúdová ochrana musí byť súčasťou inštalácie budovy.
- Ako odpojovací prostriedok sa používa vidlica napájacej šnúry. Zásuvka musí byť umiestnená blízko programátora a musí byť ľahko prístupná.

ISP (In-System Programming)

Definície

In-system programming umožňuje programovať a preprogramovávať obvod priamo v zariadení bez nutnosti vyberania zo zariadenia. Použitím jednoduchého rozhrania, ISP programátor sériovo komunikuje s obvodom, dokáže preprogramovať pamäť na čipe. ISP eliminuje fyzické vyberanie obvodu zo zariadenia, čo šetrí peniaze, čas... Je vhodné pre vývojové práce aj update software alebo parametrov zariadení v praxi.

Cieľový obvod je obvod (mikroprocesor, PLD...), ktorý je In-System programovateľný.

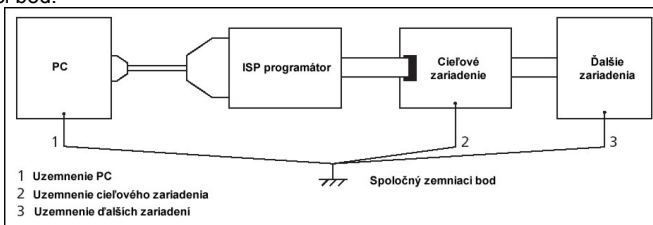
Cieľové zariadenie je fyzická doska plošných spojov, ktorá obsahuje obvod, ktorý bude In-System programovaný.

ISP programátor je programátor, ktorý je schopný ISP programovania (napr.: BeeProg2 / BeeProg+, SmartProg2...)

Všeobecné pravidlá pre In-System programming

Doporučujeme rešpektovať nasledujúce pravidlá, aby sa zabránilo poškodeniu PC, ISP programátora alebo cieľového systému.

- Zabezpečte spoločný zemniaci bod pre cieľové zariadenie, ISP programátor a PC.
- Pre laptop alebo iné typy PC, ktoré nie sú pripojené na spoločný zemniaci bod, urobte pomocou dostatočne hrubého vodiča prepojenie na spoločný zemniaci bod (využite napríklad LPT alebo COM konektor).
- Všetky zariadenia spojené s cieľovým zariadením musia byť tiež pripojené na spoločný zemniaci bod.



Postup pri pripájaní ELNEC ISP programátora k cieľovému zariadeniu

Počas In-System programovania sú spojené dve elektrické zariadenia – ISP programátor a cieľové zariadenie. Nekvalifikované prepojenie môže poškodiť až zničiť tieto zariadenia.

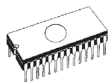
Poznámka: *Pokiaľ nedodržíte nasledujúce inštrukcie a poškodíte ISP programátor počas In-System programovania, toto poškodenie bude brané ako poškodenie nekvalifikovanou manipuláciou a nevzťahuje sa na to záruka.*

1. Vypnite obidve zariadenia – ISP programátor aj cieľové zariadenie.
2. Zabezpečte rovnaký GND potenciál pre všetky zariadenia, napr. prepojte GND všetkých zariadení.
3. Zasuňte jeden konektor ISP káblíka do ISP programátora.
4. V ovládacom programe vyberte cieľový obvod a nastavenia súvisiace s obvodom.
5. Spustíte akciu na cieľovom obvode (čítanie, programovanie...).
6. Postupujte podľa ovládacieho programu a na jeho pokyn zapojte druhý koniec ISP káblíka do cieľového zariadenia a zapnite ho.
7. Na pokyn ovládacieho programu odpojte ISP káblík z cieľového zariadenia a vypnite ho.
8. Ak chcete vykonať ďalšie operácie s cieľovým obvodom, pokračujte bodom 5.

Odporúčania na návrh cieľového zariadenia s ISP programovateľným obvodom

Cieľové zariadenie musí byť navrhnuté tak, aby signály potrebné na programovanie boli priamo privedené na ISP programátor cez ISP konektor. Ak cieľové zariadenie používa signály potrebné pre In-System programovanie aj pre iné účely, je nutné, aby tieto signály boli dostatočne oddelené, t.j. cieľové zariadenie nesmie ovplyvňovať tieto signály počas ISP programovania.

Pre In-System programovateľné obvody výrobcovia publikujú aplikačné poznámky. Programátory firmy ELNEC sú navrhnuté tak, aby sa pri dodržaní týchto aplikačných poznámok nevyskytovali žiadne problémy pri programovaní obvodov cez ISP konektor.

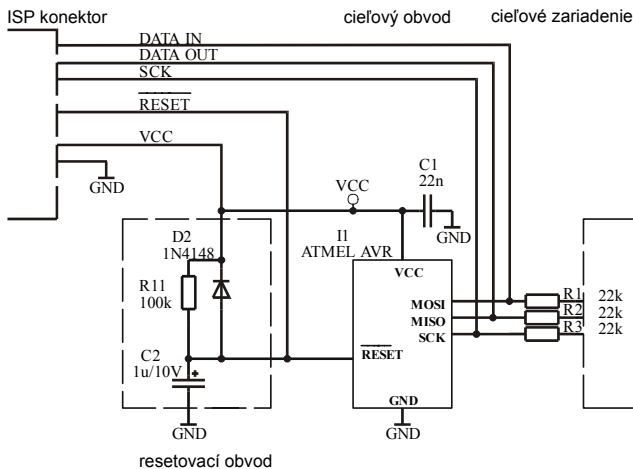


Podmienkou však je exaktné dodržiavanie týchto odporúčaní. Aplicačné poznámky, ktoré ELNEC používa pri ISP sú uverejnené na www.elnec.sk, časť Aplicačné poznámky.

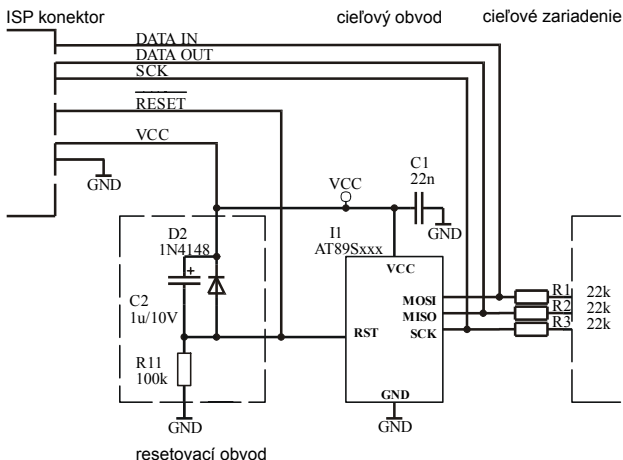
Príklady aplikačných poznámok

Procesory Atmel AVR a AT89Sxxx

Zapojenie korešponduje s aplikačnou poznámkou firmy Atmel AVR910: In-System Programming. Zapojenie pre ATMELE AVR odporúčané firmou ELNEC:



Zapojenie pre AT89Sxxx odporúčané firmou ELNEC:



PICmicro procesory

Zapojenie korešponduje s aplikačnými poznámkami firmy Microchip TB013, TB017, TB016: How to Implement ICSP™ Using PIC16CXXX OTP (PIC12C5XX OTP)(PIC16F8X Flash) MCUs. Tieto poznámky opisujú požiadavky na cieľové zariadenie s In-System programovateľným obvodom a na ISP programátor.

Na ISP programovaní PICmicro procesorov sa podieľajú nasledujúce signály:

MCLR\ / VPP reset / uvedenie do programovacieho módu

RB6 (GP1) hodiny

RB7 (GP0) vstup / výstup dát

VDD napájacie napätie

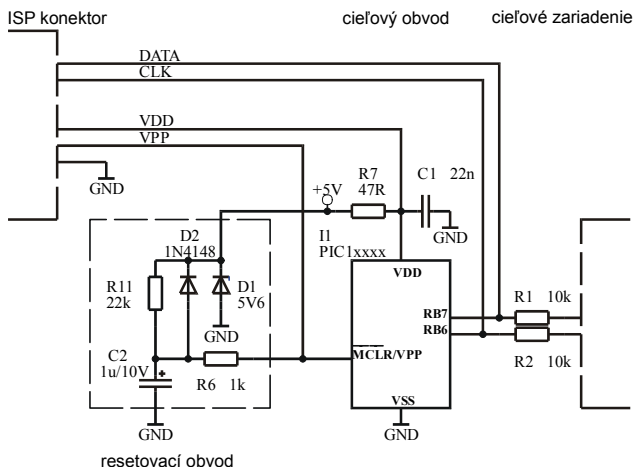
GND zem

Keď sa na pin MCLR\ / VPP privedie 12V obvod sa dostane do programovacieho módu. Tento signál je nutné oddeliť od zvyšku cieľového zariadenia, aby nedošlo k jeho zničeniu (alebo zničeniu programátora).

Signály RB6 a RB7 sa priamo podieľajú na programovaní a ich ovplyvňovanie cieľovým zariadením počas programovania by viedlo ku chybám pri programovaní.

Pri verifikácii dát naprogramovaných do PICmicro procesora sa používa marginálna verifikácia, t.j. verifikácia pri minimálnom a maximálnom povolenom napájacom napätí. Preto je potrebné mať aj VDD pin PICmicro procesora oddelený od zvyšku cieľového zariadenia.

Zapojenie pre PICmicro odporúčané firmou ELNEC:

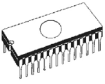


Poznámka: Externý resetrovací obvod je nutný iba, ak nábeh VDD je príliš pomalý.

Iné

Pre správnu činnosť ovládacieho programu a programátora je nutné, aby paralelný port, na ktorý je pripojený programátor bol rezervovaný iba pre neho. Inak povedané, iné programy bežiacie na PC nemú používať, prípadne modifikovať tento port.

Ovládací program Pg4uw podporuje všetky módy LPT portu, preto nie je nutné konfigurovať LPT port na správnu komunikáciu s programátorom.



Pri používaní Windows programu, prosím nepresúvajte **Info** okno, ak svieti LED BUSY. Kontrolné obvody komunikácie môžu zareagovať a vyhlási sa chyba komunikácie.

LPT port ovládač

Pre programátory pripájané k počítačom pomocou LPT portu, je potrebné mať správne nainštalovaný ovládač. Inštalovanie aj odinštalovanie ovládača sa vykonáva automaticky inštaláčnym programom. Za bežných okolností nie sú problémy s ovládačom. Niekedy sa však vyskytnú problémy. Často je to vtedy, keď na príslušnom počítači nie je v systéme Microsoft® Windows NT, Windows 2000, Windows XP (32-bit alebo 64-bit) nainštalovaný port LPT1. Ovládač vyžaduje pre správne fungovanie prítomnosť portu LPT1. Prítomnosť ďalších LPT portov je už voliteľná.

Ak je LPT port na PCMCIA alebo Express Card, karte, táto musí byť nainštalovaná v počítači pred štartom operačného systému.

LPT driver vyžaduje, aby port LPT1 bol prítomný v operačnom systéme. Skontrolujte si, či je prítomný. Krátky popis, ako skontrolovať prítomnosť portu LPT1 v systéme:

1. kliknite na menu štart
2. pravým tlačidlom myši kliknite na položku "Tento počítač" a zvolte "Vlastnosti"
3. v dialógu "Systém" vyberte záložku "Hardvér" a stlačte tlačidlo "Správca hardvéru"
4. v "Správcovi hardvéru" zvolte položku "Porty (Com & LPT)", dvojítm kliknutím a zobrazí sa zoznam prítomných portov LPT a portov COM

V zozname by sa mal nachádzať aj port LPT1.

Ak sa v zozname nachádzajú jeden alebo viac LPT portov, ale ani jeden nemá index LPT1, skúste použiť nasledujúci postup:

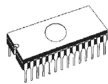
(kroky 1. až 4. použite z predošlého postupu)

5. kliknite dvojklikom na LPT port, ktorého index chcete nastaviť na LPT1, zobrazí sa dialóg s vlastnosťami portu
6. v dialógu "Vlastnosti LPT portu" vyberte položku "Nastavenia portu"
7. zmeňte číslo LPT portu v zozname voľných LPT indexov na LPT1
8. stlačte tlačidlo OK
9. reštartujte operačný systém (aj keď systém sám nevyzve na reštart, je to nutné, aby sa nové nastavenia uplatnili)

To je všetko. Náš program by mal fungovať správne.

Poznámka: *Pokiaľ používate programátor pripájaný k počítaču pomocou USB, tak nie je potrebný ovládač pre LPT porty.*

Ak sa vyskytnú problémy Záručné podmienky



Ak sa vyskytnú problémy s programátorom

Naozaj sa snažíme, aby ste boli s našimi výrobkami spokojní. Môže sa však stať, že sa vyskytnú problémy, ktoré nevieme predvídať, prípadne ovplyvniť. Prosíme, postupujte podľa nižšie uvedených inštrukcií predtým, ako sa obrátíte na svojho distribútora alebo priamo na nás s otázkami ohľadne inštalácie, prípadne skôr, ako pošlete na opravu výrobok, javiaci sa Vám ako chybný.

- Skúste nájsť odpoveď na problémy najjednoduchším spôsobom:
 - Dôkladne preštudujte dokumentáciu, dodávanú v písanej alebo elektronickej forme k programátoru. Možno nájdete odpoveď, ako riešiť problém, ktorý sa vyskytol.
 - Skúste inštalovať programátor na iný počítač. Ak sa inštalácia podarí, porovnaním rozdielov medzi počítačmi sa pokúste problém odstrániť.
 - Spýtajte sa na názor počítačového odborníka, ktorý sa stará o Váš PC.
 - Konzultujte s kýmkoľvek, kto už inštaloval, prípadne používa rovnaký programátor.
- Ak nevíete nájsť návod na riešenie problému v dokumentácii, prípadne iným z hore uvedených spôsobov, skontaktujte sa s firmou ELNEC. Veľa problémov je možné vyriešiť telefonicky, faxom, alebo e-mailom. Ak sa chcete s nami skontaktovať
 - **poštou/faxom** - Skopírujte formulár "**DEVICE PROBLEM REPORT**" a dôkladne ho vyplňte podľa inštrukcií, uvedených na konci tohto formulára. Uveďte všetky aj zdanlivo nepodstatné abnormality správania sa programátora a/alebo ovládacieho programu. Vyplnený formulár zašlite faxom na faxové číslo, ktoré nájdete v ovládacom programe PG4U sekcia **Info**, Pg4uw sekcia **Pomocník / O programe**.
 - **e-mailom** - Skopírujte formulár "**DEVICE PROBLEM REPORT**", nachádzajúci sa na sprievodnom CD alebo na našej web stránke a v ľubovoľnom editore ho dôkladne vyplňte podľa inštrukcií, uvedených na konci tohto formulára. Uveďte všetky aj zdanlivo nepatrné abnormality správania sa programátora a/alebo ovládacieho programu. Vyplnený formulár pošlite e-mailom na adresu (nospam verzia) **el nec at el nec dot sk**.
 - **telefonicky** - Skopírujte formulár "**DEVICE PROBLEM REPORT**" a dôkladne ho vyplňte podľa inštrukcií, uvedených na konci tohto formulára. Uveďte všetky aj zdanlivo nepatrné abnormality správania sa programátora a/alebo ovládacieho programu. Vyplnený formulár zašlite faxom na faxové číslo, ktoré nájdete v ovládacom programe Pg4uw sekcia **Pomocník/O programe**. Potom zatelefonujte do oddelenia podpory zákazníkov firmy ELNEC. Použite telefónne číslo, ktoré nájdete na rovnakom mieste v ovládacom programe. Prosím, majte k dispozícii manuál, programátor a vyplnený formulár "**DEVICE PROBLEM REPORT**", aby ste mohli operatívne reagovať na otázky kontaktnej osoby. Ak nenájdeme spolu riešenie problému, dohodneme spôsob pre Vás najlepšieho riešenia situácie.
- Ak je potvrdený predpoklad, že programátor je potrebné opraviť, potom pošlite programátor na opravu do servisného strediska firmy ELNEC pre danú krajinu, prípadne priamo firme ELNEC. Prosíme dôkladne zabaľte nasledovné:
 - chybný výrobok s príslušenstvom ,
 - vyplnený formulár "**DEVICE PROBLEM REPORT**"
 - kópiu dokladu o nákupe s dátumom predaja.

Bez týchto náležitostí nebude výrobok prijatý na opravu.

Poznámky:

Formulár "**DEVICE PROBLEM REPORT**" nájdete:

- Na našej internetovej stránke (www.elnec.com), v časti Podpora / Hlásenie problému (Support / Problem report).

Ak sa vyskytnú nepodporované obvody

Iste vo svojej praxi nájdete obvody, ktoré Vaša verzia ovládacieho programu k programátoru momentálne nepodporuje. Taktiež je možné, že sa vyskytnú problémy pri práci s obvody, ktoré ovládaci program podporuje a chyba nie je v programátore. V obidvoch prípadoch postupujte prosím nasledujúcim spôsobom:

- zabezpečte si najnovšiu verziu ovládacieho programu (najrýchlejšie cez Internet) a presvedčte sa, či už nie je podpora pre tento obvod v tejto verzii programu implementovaná. Ak nie je, potom
- Vyplňte formulár "**DEVICE PROBLEM REPORT**" podľa pokynov, uvedených na konci tohto formulára. Uveďte všetky aj zdanlivo nepodstatné detaily prípadu. Tento formulár zašlite spolu so vzorkami obvodov (ak sú k dispozícii) k nám. Po implementovaní podpory tohto obvodu do ovládacieho programu Vám vzorky samozrejme vrátíme.

Záručné podmienky

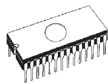
Firma ELNEC s.r.o. Prešov, Slovensko poskytuje záruku na bezchybnú funkciu programátorov aj na to, že programátor i všetky jeho súčasti sú bez chýb materiálu alebo spracovania a to **tri roky** odo dňa nákupu (pre BeeHive204, BeeHive4+, BeeProg2, BeeProg+, SmartProg2), alebo **dva roky** (pre MEMprog2, T51prog2, PIKprog2 a SEEprog) s obmedzením 25 000 cyklov na DIL ZIF päťicu alebo 10 000 cyklov na iný typ ZIF päťíc. Ak bude programátor uznaný ako chybný, firma ELNEC alebo autorizované servisné stredisko bezplatne opraví, alebo vymení chybné časti. Na vymenené časti a celý výrobok poskytujeme záruku len na zvyšok pôvodnej záručnej lehoty.

Pre poskytnutie záručnej opravy je nutné preukázanie dátumu predaja dokladom o zaplatení alebo dodacím listom. Rozšírená záruka sa týka len tých majiteľov (koncových zákazníkov) programátorov, ktorí majú potvrdenie o obdržaní rozšírenej záručnej lehoty (pozrite, prosím, poznámku na konci kapitoly "Záručné podmienky"). Pre ostatných zákazníkov platí 2 ročná záručná lehota.

Tieto záručné podmienky platia pre užívateľov, ktorý si kúpili programátor priamo od ELNEC s.r.o. Záručné podmienky distribútorov môžu byť iné a závislé od zákonov cieľovej krajiny alebo záručnej politiky distribútora.

Záruka sa nevzťahuje na bežné opotrebovanie, ani na akékoľvek mechanické poškodenie. Rovnako sa záruka nevzťahuje na výrobky, ktoré boli používané mimo pôvodného určenia, poškodené neodbornou manipuláciou, chybnou inštaláciou, alebo živelnou udalosťou. Záruka nie je poskytovaná ani na výrobky, do ktorých bol vykonaný akýkoľvek zásah neautorizovanou osobou.

Platba na pozáručné opravy závisí od nákladov na opravené/vymenené dielce, potrebného času opravy a dopravných nákladov. Firma ELNEC alebo distribútor má právo určiť, či bude výrobok opravovaný alebo vymenený a rozhodnúť o poskytnutí záručnej alebo pozáručnej opravy.

**Poznámky:**

- Potvrdenie **"Rozšírená záruka"** obdrží každý zákazník, ktorý sa zaregistruje u firmy ELNEC zaslaním úplne vyplnenej **registračnej karty** poštou, faxom alebo e-mailom najneskôr do 4 týždňov od zakúpenia programátora. Obratom od nás obdržíte potvrdenie **"Rozšírená záruka"** s Vaším registračným číslom. Riziko za nedoručenie registračnej karty k nám nesie zákazník. Registračná karta je súčasťou dodávky programátora. Ak by z akéhokoľvek dôvodu v zásielke chýbala, kópiu karty nájdete v prílohe tohto návodu.
- *Pozrite prosím tiež sekciu "Ak sa vyskytnú problémy s programátorom".*

Kontaktné údaje výrobcu:

Výrobca:

✉: **ELNEC s. r. o.**, Pošta 5, P. O. Box 22, SK - 08005 Prešov

☎: +42151/77 34 328, 77 31 007, fax 77 32 797

www.elnec.sk, (nosspam verzia) elnec at elnec dot sk

ELNEC sa snaží vyvíjať hardware a software, ktorý dlhodobo spoľahlivo pracuje a plní svoju funkciu.

ELNEC nezaručuje, že hardware a software je a bude bez chýb a porúch.

ELNEC zodpovedá za škody len do sumy, ktorú zaplatil kupujúci.

ELNEC nezodpovedá za :

Škody spôsobené nevhodným používaním a zachádzaním s výrobkom.

Škody spôsobené užívateľom alebo treťou stranou pri pokuse o zmenu alebo zmene výrobku.

Akéhokoľvek ďalšie alebo následné škody spôsobené hardwareovými alebo softwareovými chybami.

Príklad: *strata zisku, strata úspor, škody vzniknuté z požiadaviek tretích strán voči zákazníkovi, poškodenie alebo strata dát alebo súborov, poškodenie dobrého mena, škody vzniknuté z nemožnosti používať výrobok atď.*