



## Ръководство за употреба на Prog-Express

## Съдържание

<b>СЪДЪРЖАНИЕ</b>	<b>2</b>
<b>СИСТЕМНИ ИЗИСКВАНИЯ</b>	<b>5</b>
32-БИТОВИ ОПЕРАЦИОННИ СИСТЕМИ	5
64-БИТОВИ ОПЕРАЦИОННИ СИСТЕМИ	5
<b>ЛИЦЕНЗИОННА ИНФОРМАЦИЯ ЗА PROG-EXPRESS</b>	<b>6</b>
УСЛОВИЯ ЗА БЕЗПЛАТЕН ЛИЦЕНЗ	6
<b>ИНСТАЛАЦИЯ</b>	<b>7</b>
СОФТУЕРНА ИНСТАЛАЦИЯ	7
ИНСТАЛАЦИЯ НА ДРАЙВЪРИТЕ НА УСТРОЙСТВОТО	8
АВТОМАТИЧНА ИНСТАЛАЦИЯ	8
РЪЧНА ИНСТАЛАЦИЯ	9
<b>РЕЖИМИ НА СОФТУЕРА</b>	<b>12</b>
КРАТКО ИЗЛОЖЕНИЕ	12
ПРОГРАМИРАНЕ НА ЧИП	13
КОПИРАНЕ НА ЧИП	14
ЧЕТЕНЕ НА ЧИП	15
ПРОИЗВОДСТВЕН РЕЖИМ	16
ПОЛЕТА ЗА ВЪВЕЖДАНЕ НА ДАННИ	19
ПРОГРАМАТОР	19
ЧИП-ИЗТОЧНИК И ЧИП-ПРИЕМНИК	19
ФАЙЛ-ИЗТОЧНИК	19
ЗАПАЗВАНЕ НА ФАЙЛ	21
ОПЦИИ НА ЧИПА	21
СЕРИЙНИ НОМЕРА	21
УПРАВЛЕНИЕ НА ПРОЦЕС	22
СТЪПКИ НА ПРОЦЕС	23
<b>HEX-РЕДАКТОР</b>	<b>25</b>
МЕНЮ И ИНСТРУМЕНТАЛНА ЛЕНТА НА HEX-РЕДАКТОРА	26

МЕНЮ И ИНСТРУМЕНТАЛНА ЛЕНТА „ФАЙЛ“ .....	26
МЕНЮ И ИНСТРУМЕНТАЛНА ЛЕНТА „РЕДАКТИРАНЕ“ .....	27
МЕНЮ И ИНСТРУМЕНТАЛНА ЛЕНТА „ИЗГЛЕД“ .....	28
МЕНЮ И ИНСТРУМЕНТАЛНА ЛЕНТА „ПРОГРАМАТОР“ .....	29
КОНТЕКСТНО МЕНЮ НА HEX-РЕДАКТОРА .....	31
РАБОТА СЪС СЕЛЕКЦИЯ .....	32
ИЗБОР НА ОБЛАСТИ .....	32
РЕДАКТИРАНЕ НА ОБЛАСТИ .....	32
ПРОМЯНА НА ИЗГЛЕДА ЧРЕЗ МИШКАТА .....	33
КЛАВИШИ ЗА БЪРЗ ДОСТЪП В HEX-РЕДАКТОРА .....	34
<b>СОФТУЕРНИ ОПЦИИ .....</b>	<b>36</b>
ОСНОВНИ ОПЦИИ .....	36
ПРОЕКТ .....	38
ЗАЩИТА .....	40
ФАЙЛОВИ ВРЪЗКИ .....	42
ОПЦИИ ЗА ЕЗИК .....	43
ДОБАВКИ .....	44
ЛОГ НА ДЕЙСТВИЯТА .....	45
<b>ЧИП БРАУЗЪР .....</b>	<b>46</b>
<b>АВТОМАТИЧНО ИДЕНТИФИЦИРАНЕ НА ЧИП .....</b>	<b>47</b>
<b>ОПЦИИ НА ЧИПА .....</b>	<b>48</b>
СПЕЦИФИЧНИ ОПЦИИ НА ЧИПА .....	48
ОПЦИИ ЗА РАЗДЕЛЯНЕ .....	49
<b>СЕРИЙНИ НОМЕРА .....</b>	<b>50</b>
ФАЙЛ ЗА СЕРИЙНИ НОМЕРА .....	51
ГЕНЕРАТОР ЗА СЕРИЙНИ НОМЕРА .....	52
<b>ПРОЕКТИ .....</b>	<b>53</b>
ЗАПАЗВАНЕ .....	53
ЗАРЕЖДАНЕ .....	53
<b>ОЩЕ .....</b>	<b>54</b>

ОПЕРАТИВЕН РЕЖИМ .....	54
ОБНОВЯВАНЕ НА ФЪРМУЕРА .....	54
КАЛИБРИРАНЕ .....	54
ИНФОРМАЦИЯ ЗА PROG-EXPRESS .....	54
ПОМОЩНИК НА PROG-EXPRESS .....	54
<b>ДИСТАНЦИОННО УПРАВЛЕНИЕ НА PROG-EXPRESS .....</b>	<b>55</b>
УПРАВЛЕНИЕ ЧРЕЗ КОМАНДЕН ИНТЕРПРЕТАТОР .....	55
ДИСТАНЦИОННО УПРАВЛЕНИЕ ЧРЕЗ СКРИПТОВИ ФАЙЛОВЕ .....	55
КОМАНДА REMOTEFILE .....	56
КОМАНДИ POLL ON И POLL OFF .....	56
КОМАНДА OPEN .....	56
КОМАНДА MODE .....	56
КОМАНДА SELECTFILE .....	57
КОМАНДА PROCESSSTEP .....	57
КОМАНДА SELECTCHIPVARIANT .....	57
КОМАНДА AUTOIDENTIFY .....	57
КОМАНДА RUN .....	58
КОМАНДА SAVELOG .....	58
КОМАНДА SAVEDEVICEINFO .....	58
КОМАНДА CLEARLOG .....	58
КОМАНДА SAVECHIPLIST .....	59
ДОПЪЛНИТЕЛНИ КОМАНДИ .....	59
ПРИМЕРНИ ПРИЛОЖЕНИЯ: ПРОГРАМИРАНЕ НА СПЕЦИФИЧНИ ДАННИ .....	59

## Системни изисквания

Тази глава съдържа системните изисквания за Prog-Express и Microsoft® .NET Framework 2.0.

Prog-Express изисква Microsoft® .NET Framework 2.0. Той е включен в компакт диска с ръководството и ще бъде инсталиран автоматично при нужда.

### 32-Битови операционни системи

- Поддържани операционни системи: Windows 7, Windows Vista, Windows XP Service Pack 2 или по-нов, Windows Server 2003, Windows 2000 Service Pack 3, Windows ME, Windows 98 Second Edition.
- Изисквания: Windows Installer 3.0 (с изключение на Windows 98/ME, които изискват Windows Installer 2.0 или по-нов). Препоръчва се Windows Installer 3.1 или по-нов. IE 5.01 или по-нов: За инсталация на .NET Framework Microsoft се изисква Internet Explorer 5.01 или по-нов.
- Минимално пространство на хард диска: 300 MB (x86)

### 64-Битови операционни системи

- Поддържани операционни системи: Windows 7 - 64 Bit, Windows Vista 64 Bit, Windows XP 64-bit, Windows Server 2003 x64 Edition.
- Изисквания: Windows Installer 3.0 (с изключение на Windows 98/ME, които изискват Windows Installer 2.0 или по-нов). Препоръчва се Windows Installer 3.1. IE 5.01 или по-нов: За инсталация на .NET Framework Microsoft се изисква Internet Explorer 5.01 или по-нов.
- 64-битова поддръжка: За поддържане на 64-битови процесори се изисква актуалната 64-битова версия на Windows XP Professional или Windows Server 2003.
- Минимално пространство на хард диска: 630 MB (64-бита)

## Лицензионна информация за Prog-Express

### Условия за безплатен лиценз

#### Ограничена гаранция

Софтуерът и документацията се предоставят за ваша употреба. Тъй като вероятността от възникване на неизправности никога не може да бъде изключена поради множеството компютърни конфигурации, дори с изцяло тестван софтуер, авторът не носи каквато и да е отговорност за щети, които могат да възникнат от пряката или непряката употреба на софтуера и документацията. При никакви обстоятелства авторът не може да бъде държан отговорен за щети като пропуснати ползи, прекъсване на бизнес операции, загуба на информация или данни и щети върху друг софтуер дори ако основният проблем е известен на автора. Потребителят носи пълна отговорност за всички последствия, произтичащи от използването на този софтуер.

#### Възпроизвеждане

Разпространението на програмите, публикуването на компакт диска в колекции за споделяне или в печатни медии, както публикуване чрез буукуер са допустими само с изричното писмено съгласие на автора.

Подправяне, декомпиляция или разделянето на части на софтуера и свързаните с него файлове ще доведе до максимум 5 години лишаване от свобода или глоба според § 263a от Наказателния кодекс на Германия. Авторът ще подаде жалба и ще търси компенсация за вредите.

#### Други

Ако тези условия не са изпълнени, авторът има право да анулира лиценза на потребителя за софтуера.

Всички използвани тук наименования на продукти и търговски марки се признават за принадлежност на техните собственици, независимо дали те са идентифицирани като такива.

Прилагат се разпоредбите на закона на Федерална Република Германия.

## Инсталация

Първо, моля инсталирайте софтуера Prog-Express. Можете да откриете софтуера в предоставения ви компакт диск или на [www.batronix.com](http://www.batronix.com) в раздела за сваляне.

След инсталацията можете да свържете USB програматора за първи път.

### Софтуерна инсталация

Тази глава съдържа необходимата информация за инсталиране на Prog-Express.

За да започнете инсталиране, кликнете върху Setup файла или използвайте функцията за автоматично стартиране на компакт диска.

Моля, следвайте инструкциите на екрана.

Инсталирането на Microsoft® .NET Framework 2.0 може да бъде поискано, тъй като той е едно от изискванията на софтуера. Ако е налице интернет връзка, вашият компютър ще свали необходимите файлове. В противен случай може да ги откриете и в компакт диска.

#### .NET Версия X64, x86 или IA64

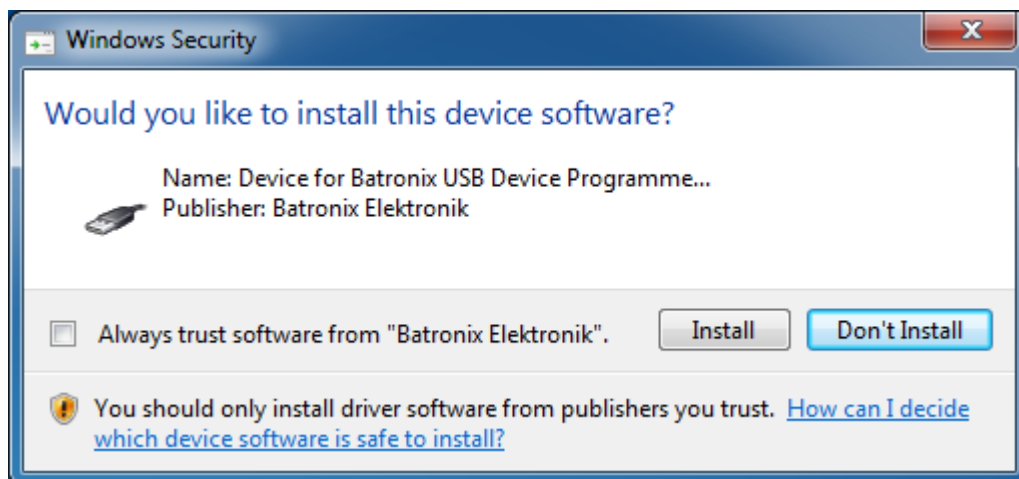
Ако използвате 32-битова операционна система, моля свалете версия x86. Версия x64 е съвместима единствено с 64-битови операционни системи. IA64 се изисква само за 64-битови Intel процесори с 64-битови операционни системи.

## Инсталация на драйвърите на устройството

### Автоматична инсталация

По време на инсталирането на Prog-Express драйвърите за USB програматорите на Batronix се инсталират автоматично, както можете да видите на изображенията. Windows ще започне USB инсталация и ще покаже следната информация на монитора.

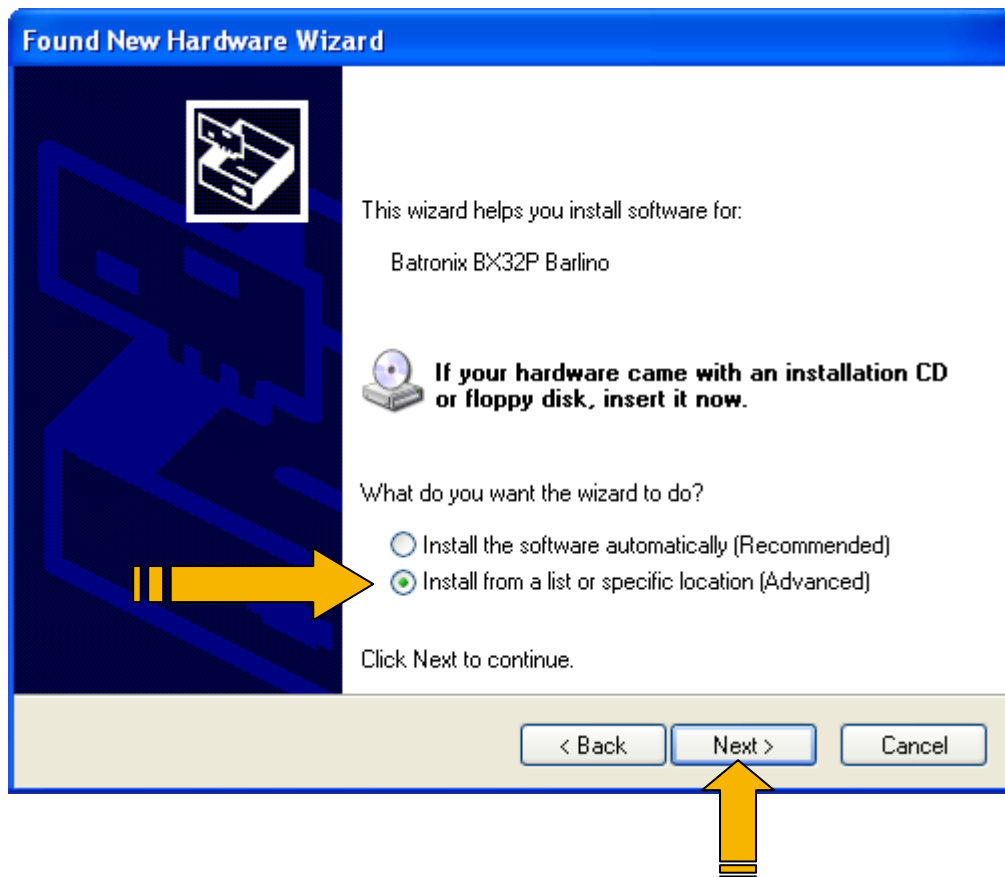
Ако програматорът няма да бъде показван в софтуера Prog-Express, може да се наложи рестартиране на компютъра.



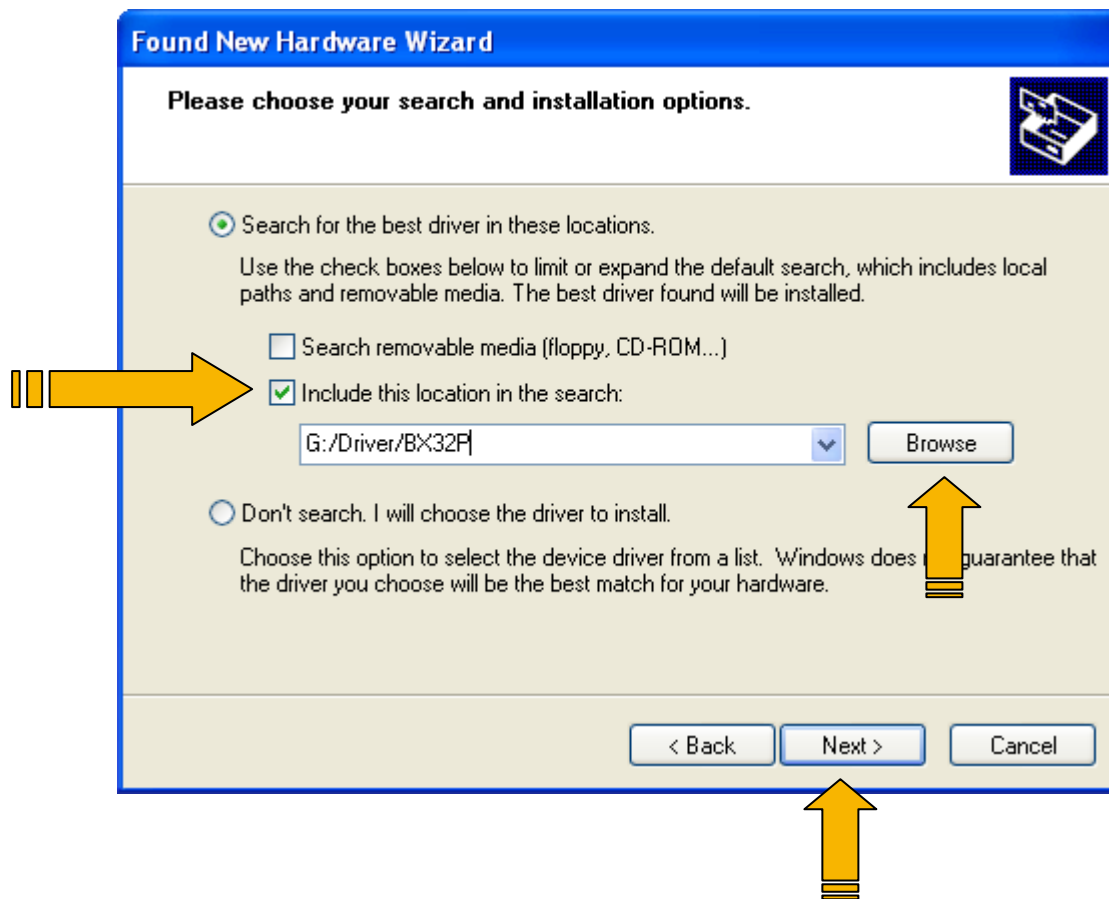
## Ръчна инсталация

Инсталацията на Prog-Express ще инсталира всички драйвъри за USB програматорите на Batronix автоматично. Но ако се нуждаете от ръчното им инсталиране, можете да следвате следните стъпки.

След инсталирането на софтуера можете да свържете USB програматор на Batronix за първи път. Windows незабавно ще засече устройството като ново USB устройство и ще покаже следното съобщение на екрана:



Моля, изберете втората опция „install from a list or specific location“ („Инсталиране от списък или конкретно място“). Потвърдете избора с „Next“ („Напред“).

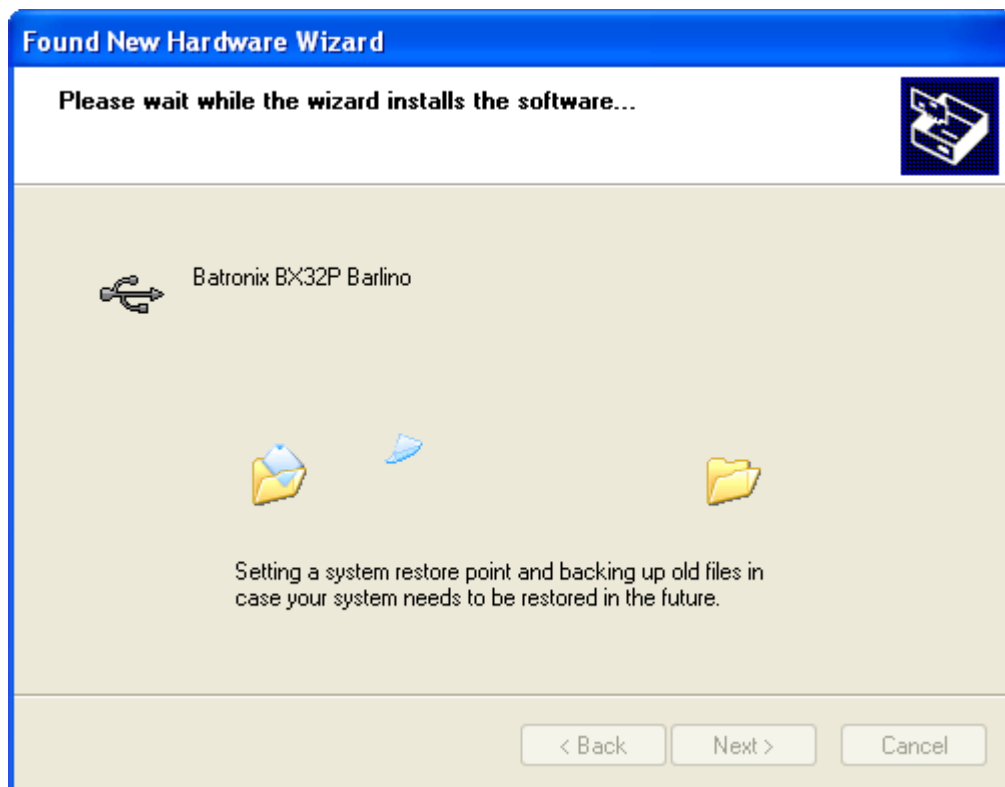


Активирайте „Include this location in the search“ („Включване на това местонахождение в търсенето“) и посочете директория, принадлежащата на програматора ви, в директория „driver“ от инсталацията на Prog-Studio или директория „driver“ от компакт диска. Например: C:\Program Files\Batronix\Prog-Express\driver\BX32P

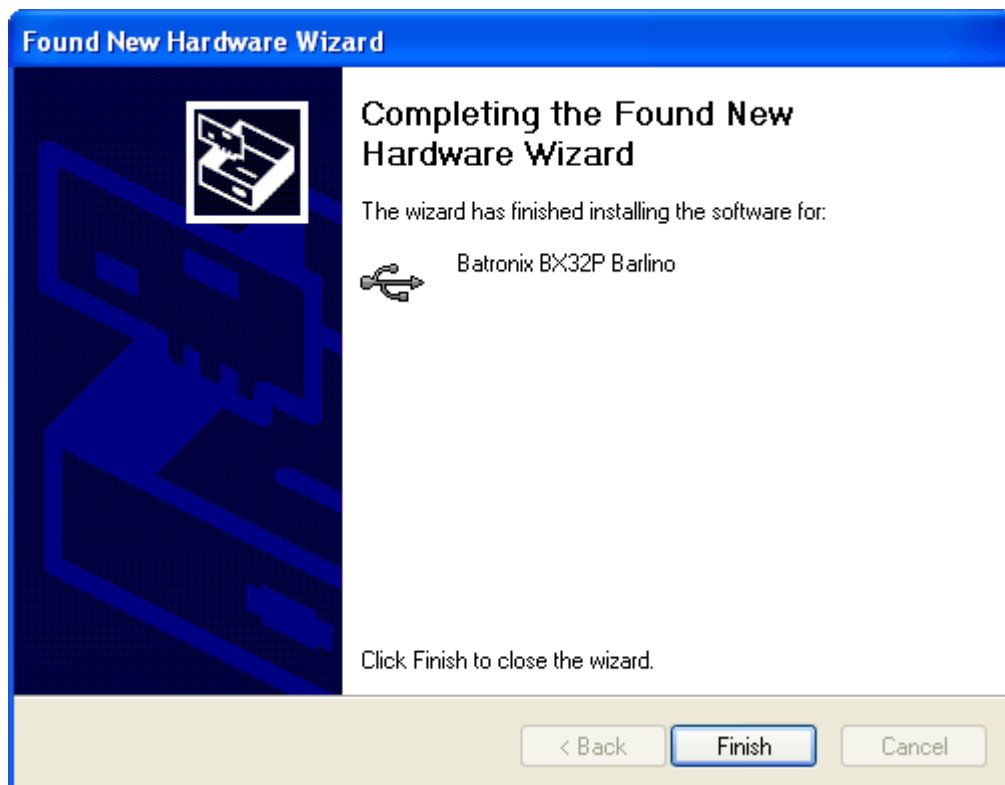
Потвърдете избора с „Next“ („Напред“).



Ако получите съобщение „Device has not passed Windows Logo testing“ просто кликнете върху „Continue Anyway“ („Продължи въпреки това“).



В този момент компютърът ви търси посочените източници и ще открие програматора. Драйвърът ще бъде инсталиран.



Съветникът завършва със съобщение „The wizard has finished installing the software for...” („Съветникът завърши инсталирането на софтуера за“). Потвърдете съобщението като натиснете бутона „Finish“ („Приключи“).

Драйвърът е инсталиран успешно и вече можете да използвате устройството.

## Режими на софтуера

Тази глава съдържа информация за различните режими на софтуера Prog-Express.

### Кратко изложение

Prog-Express притежава пет различни основни режима и диалог с опции, които могат да се посочват от менюто за избор. С тези режими можете да избирате оптималния интерфейс за вас.

На разположение са следните режими:

- Програмиране на чип: Програмиране на един или повече чипа
- Копиране на чип: Копиране на един или повече чипа
- Четене на чип: Четене на данни от чип
- Производствен режим: Програмиране на чипове с няколко устройства
- Буфер Нех-редактор: Редактиране на данни чрез hex-редактор
- Софтуерни опции: Редактиране на софтуерните опции








## Програмиране на чип

Режимът „Програмиране на чип“ предлага оптимизиран интерфейс за програмиране на данни от източник като хард диск или компакт диск върху един или повече чипа.



**Програмиране на чип**

	<b>BX32 Batupo II</b> Сериен номер: BL2507791 Версия на фърмуер: 02.30	<input type="button" value="Опресняване"/>
	<b>Winbond W29EE011P-90</b> Адаптер: PLCC32-DIP32 128 KByte ( 1 MBit )	<input type="button" value="Автоматично идентифициране на чип"/>
	<b>ProgramData1MBit.bin</b> Път: Y:\Files\ Последна промяна: 05 Май 2011 г. 10:21:51	<input type="button" value="Търсене"/>
	<b>Опции на чипа (по избор)</b> Не са използвани специални опции	<input type="button" value="Опции за настройка"/>
	<b>Сериен номер (по избор)</b> Няма да бъдат вмъквани серийни номера	<input type="button" value="Опции за настройка"/>

Копия

В горната част от интерфейса са разположени полета за въвеждане на данни, а в долната част – функциите за управление на процеса.

Ако сте свързали няколко програматора, изберете устройството, което искате да използвате в първото поле. След това във второто поле изберете чипа, който искате да използвате. В третото поле изберете файла, който искате да програмирате. Настройките за опции на чипа и серийни номера се изискват единствено при специални случаи и обикновено не е необходимо да се активират. Кликнете върху синия бутон за начало, за да стартирате програмния процес.


Полетата за въвеждане на данни в горната част са описани подробно в глава „Полета за въвеждане на данни“. Основни указания за управление на процеса са посочени в глава „Управление на процес“.


## Копиране на чип


Режимът „Копиране на чип“ предлага оптимизиран интерфейс за програмиране на данни от един чип върху един или повече чипа.



**Копиране на чип**

 **BX32 Batupo II**  
Сериен номер: BL2507791  
Версия на фърмуер: 02.30


 **Чип-източник: Winbond W29EE011P-90**  
Адаптер: PLCC32-DIP32  
128 KByte ( 1 MBit )

 **Чип-приемник: Winbond W29EE011P-90**  
Адаптер: PLCC32-DIP32  
128 KByte ( 1 MBit )

Опресняване

Автоматично идентифициране на чип

Автоматично идентифициране на чип

 **Стартиране на процес копиране**

Копия

В горната част от интерфейса са разположени полета за въвеждане на данни, а в долната част – функциите за управление на процеса.

Ако сте свързали няколко програматора, изберете устройството, което искате да използвате в първото поле. След това във второто поле изберете чипа, който искате да копирате, а в третото поле – чипа-приемник. Възможно е чипът-източник да се различава от чиповете-приемници. За да може копието да работи точно като оригиналния чип, двата чипа трябва да имат един и същ размер памет, еднакво предназначение на крачетата и да използват еднакви алгоритми за управление. Кликнете върху синия бутон за начало, за да стартирате процеса на копиране.

Преди стъпката „Проверка сигнатурата на чипа-приемник“ от потребителя се изисква да постави чипа-приемник (без значение дали функцията „Проверка сигнатурата на чипа-приемник“ е активирана или не).

Полетата за въвеждане на данни в горната част са описани подробно в глава „Полета за въвеждане на данни“. Основни указания за управление на процеса са посочени в глава „Управление на процес“.

## Четене на чип

Режимът „Четене на чип“ предлага оптимизиран интерфейс за четене на данни от чип и запамятаването им във файл или прегледа им в hex-редактор.



В горната част от интерфейса са разположени полета за въвеждане на данни, а в долната част – функциите за управление на процеса.

Ако сте свързали няколко програматора, изберете устройството, което искате да използвате в първото поле. След това във второто поле изберете чипа, който искате да прочетете.

Ако е активирана опцията за запазване на данните от буфера, данните ще бъдат записани във файл веднага след като бъдат прочетени. Ако тази опция е активирана, ще имате на разположение трето поле, в което да определите пътя до файла и неговото име.

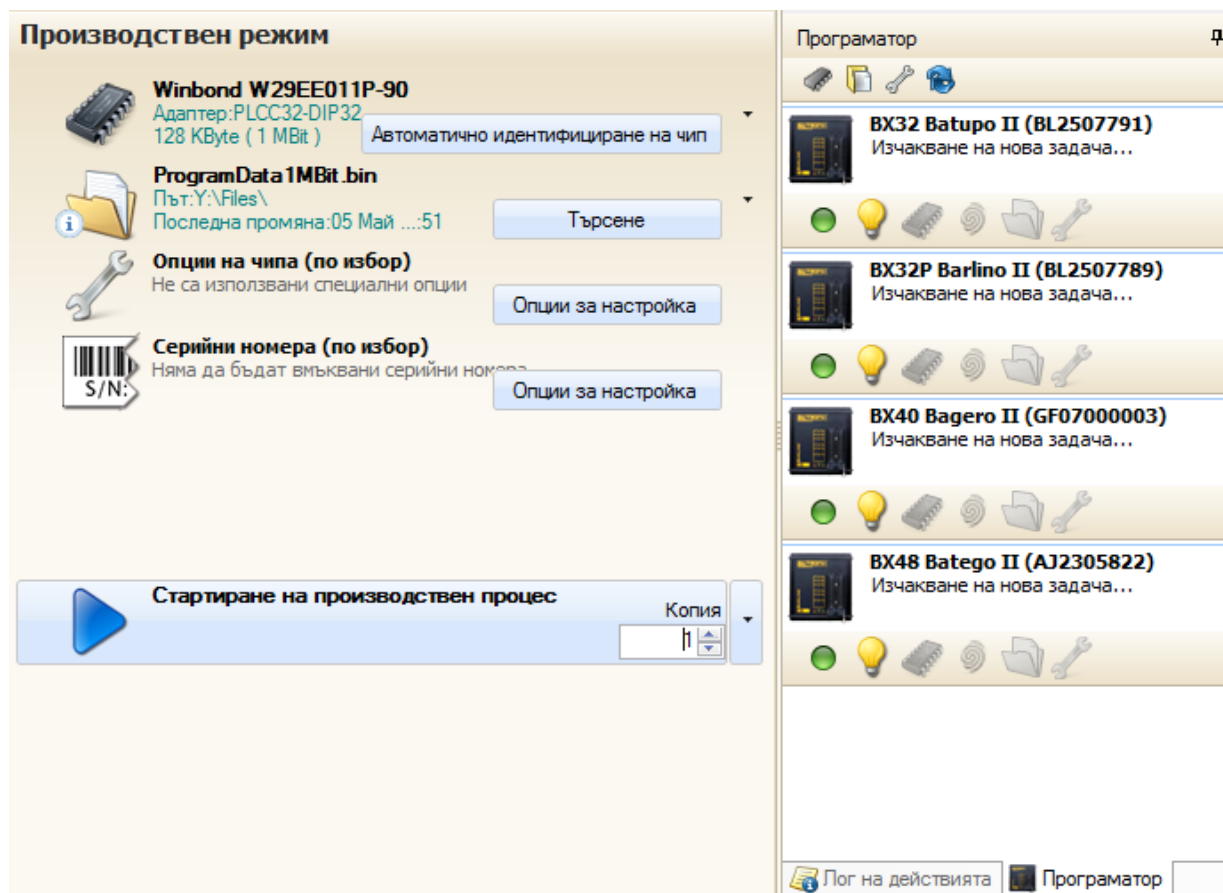
Ако е активирана опцията за показване в hex-редактора, данните ще бъдат показани в hex-редактора веднага след като бъдат прочетени.

Кликнете върху синия бутон за начало, за да стартирате процеса на четене.

Полетата за въвеждане на данни в горната част са описани подробно в глава „Полета за въвеждане на данни“. Основни указания за управление на процеса са посочени в глава „Управление на процес“.

## Производствен режим

Производственият режим предлага оптимизиран интерфейс за едновременно програмиране на чипове на няколко програматора.



В този режим могат да бъдат управлявани до осем USB програматора и осем професионални програматора (BX-програмиращи устройства). Всички разпознати устройства се показват в списъка за избор на устройства (дясната страна от изображението).

Всяко програмиращо устройство се управлява независимо, за да се постигне високо ниво на продуктивност. Скоростта на програмиране на всеки програматор е приблизително толкова бърза, колкото при използване на едно единствено устройство.

Ако използвате хъб USB връзката между компютъра и хъба трябва да поддържа режим на висока скорост на USB 2.0, така че трансферът на данни да не бъде ограничаван излишно.

Полетата за въвеждане на данни в горната част са описани подробно в глава „Полета за въвеждане на данни“. Основни указания за управление на процеса са посочени в глава „Управление на процес“.

В производствен режим от дясната страна на екрана са показани програмиращите устройства. Най-отгоре на екрана е разположена лента с инструменти. Това ви позволява да активирате настройки на чиповете, файловете и опциите индивидуално за всяко програмиращо устройство. Ако е активирана настройка, специфична за определен програматор, съответният бутон ще бъде оцветен в оранжево.



Използвайте бутона за опресняване, за да опресните списъка със свързаните в момента програмиращи устройства.

Показани са информация за устройството и допълнителни пет или шест бутона за всеки свързан програматор.



Кликнете върху този бутон, за да прибавите или премахнете програмиращо устройство от производствения процес. Ако бутонът е оцветен в оранжево, устройството е включено в производствения процес.

За да откриете определено програмиращо устройство на работната маса от няколко



програматора, е необходимо единствено да натиснете този бутон. Зеленият диод на съответното устройство ще премигне за кратко.

Използвайте този бутон, за да изберете отделен чип за всяко програмиращо устройство. По този начин можете например да използвате четири програматора, за да програмирате четири различни чипа в един производствен процес.



Използвайте този бутон, за да изберете отделен файл за всяко програмиращо устройство. По този начин можете например да използвате четири програматора, за да програмирате четири различни файла в един производствен процес.



С този бутон можете да избирате различни опции (офсет, разделяне и т.н.) за всяко програмиращо устройство. По този начин можете например да използвате два програматора с различни опции за разделяне (четни, нечетни) в един производствен процес.



Ако желаният чип не съдържа сигнатура, поставянето и отстраняването на тези чипове не може да бъде засечено автоматично. Стъпката от процеса за изчакване на чип не може да бъде активирана. Следователно последователността на процеса за съответното устройство трябва да бъде започната със стартовия бутон на устройството, след като чипът бъде поставен.



The screenshot shows a software interface with a toolbar at the top containing icons for a chip, a file, a wrench, and a blue triangle. Below the toolbar, there is a list of four programming devices, each with a small image of the device, its name, chip information, and file path. At the bottom of each device entry is a row of five icons: a green circle, a lightbulb, an IC chip, a folder, and a wrench, followed by the text 'Успешни:0/1'.

Device Name	Chip	File	Status
<b>BX32 Batupo II (BL2507791)</b>	Чип:Winbond W29EE011P-90	Файл:ProgramData 1MBit.bin	Успешни:0/1
<b>BX32P Barlino II (BL2507789)</b>	Чип:Winbond W29EE011P-90	Файл:ProgramData 1MBit.bin	Успешни:0/1
<b>BX40 Bagero II (GF07000003)</b>	Чип:Winbond W29EE011P-90	Файл:ProgramData 1MBit.bin	Успешни:0/1
<b>BX48 Batego II (AJ2305822)</b>	Чип:Winbond W29EE011P-90	Файл:ProgramData 1MBit.bin	Успешни:0/1

В горния десен ъгъл е изобразен символ, посочващ съответния статус на всяко програмиращо устройство.



Зареждане на файл в буфера.



Изчакване за поставяне на чип.



Проверка сигнатурата на чипа.



Премахване на защитните битове.



Изтриване на поставения чип.



Проверка дали чипът е празен.



Програмиране на чипа.



Проверка на програмираните данни.



Задаване на защита срещу запис.



Сравняване на опциите на чипа.

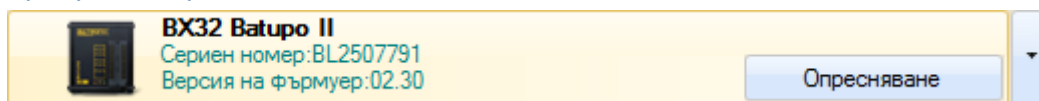


Изчакване за отстраняване на чип.

## Полета за въвеждане на данни

В режимите „Програмиране на чип“, „Копиране на чип“, „Четене на чип“ и „Производствен режим“ са налице няколко полета за въвеждане на данни в горната част на екрана.

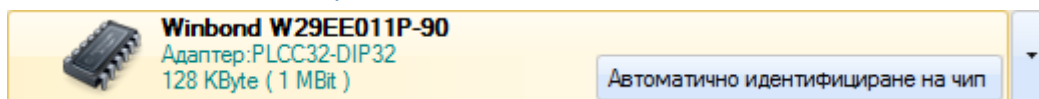
### Програматор



Сериен номер и фърмуер версията на избраното програмиращо устройство са показани под името на устройството в това поле. Кликането в лявата област от този бутон предизвиква няколкократното премигване на зеления диод на избрания програматор. Тази функция е полезна, когато са свързани няколко програматора и е необходимо да идентифицирате определено устройство.

Кликането върху стрелката отдясно отваря списък с всички свързани програмиращи устройства. Когато отворите списъка свързаните в момента устройства се засичат и списъкът се опреснява. Желаните устройства се избират чрез кликане върху тях в списъка.

### Чип-източник и чип-приемник

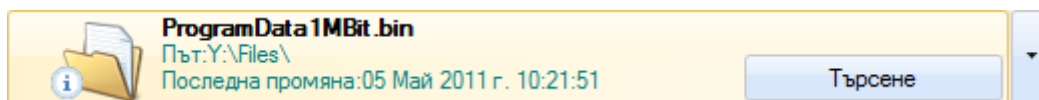


В това поле са посочени името на чипа и размера на паметта му, както и името на подходящия адаптер, ако се изисква такъв. Кликането в лявата област от този бутон отваря чип браузър, който позволява избор на чип.

Използването на бутона „Автоматично идентифициране на чип“ позволява автоматично избиране чрез сигнатурата на чипа (виж глава „Автоматично идентифициране на чип“).

Кликането върху стрелката отдясно отваря списък с последните 10 използвани чипа. Желаният чип се избира чрез кликане върху него в списъка.

### Файл-източник



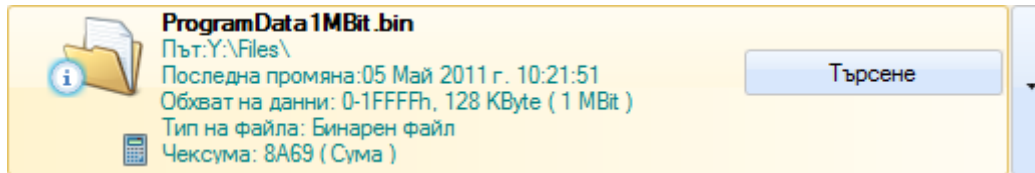
Използвайте полето за отваряне на файл, за да изберете файла, който да бъде зареден. Пътят до файла и датата и часа на последната му промяна са посочени под името му. Кликането в лявата област от бутона отваря браузър, който позволява да изберете файл.

Кликането върху стрелката отдясно отваря списък с последните 10 използвани файла. Желаният файл се избира чрез кликане върху него в списъка.

### Списък на последните използвани файлове

С кликане върху стрелката отдясно се отваря списък с последните 10 използвани файла. С едно кликане желаният файл може да бъде отворен.

## Допълнителна информация за файла

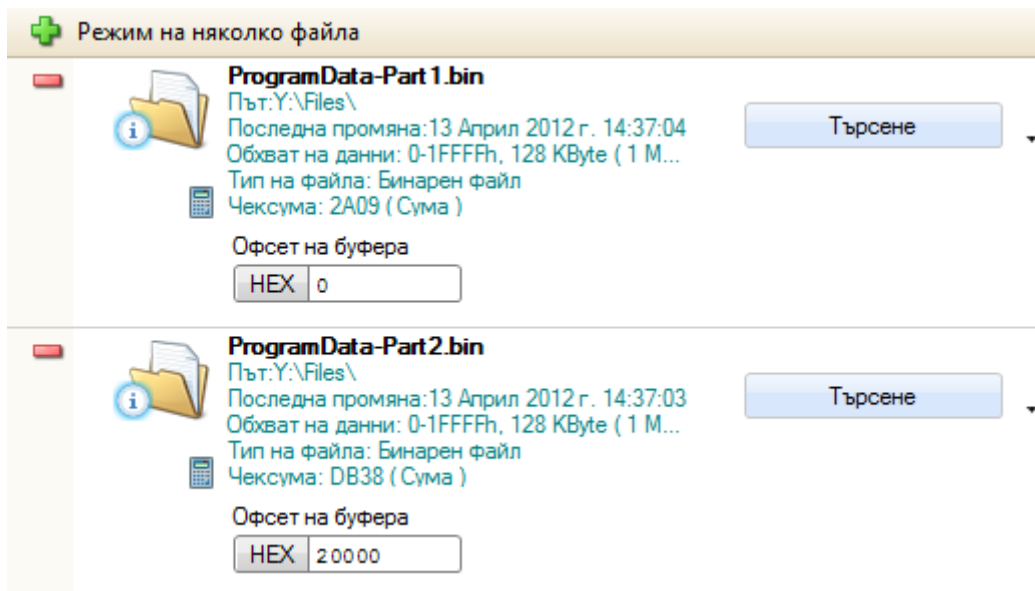


Натискайки малкия i-символ, вляво от иконата на папката, се показва допълнителна информация. Алгоритмът за изчисляване на чексума може да бъде избран с кликане върху символа калкулатор.

## Контекстно меню

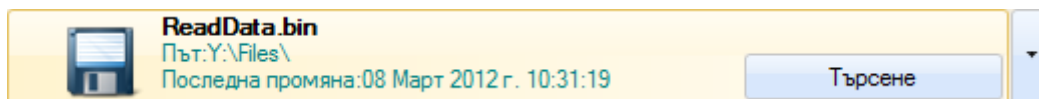
Може да отворите контекстно меню с натискане на десния бутон на мишката върху полето за отваряне на файл. В това контекстно меню ще намерите допълнителни функции като отваряне на папки в Explorer, активиране/деактивиране на режим на няколко файла и показване на различна информация за избрания файл.

## Режим на няколко файла



Режимът на няколко файла може да бъде включван и изключван от контекстното меню. В този режим могат да бъдат избирани няколко файла, които да се програмират един след друг. Позицията на всеки файл може да бъде определяна от полето „Офсет на буфера“.

## Запазване на файл



В полето „Запазване на файл“ можете да определите къде и под какво име данните от чипа да бъдат запазени след прочитането им. Текущата информация от буфера се запазва чрез процесната стъпка за запазване на данните от буфера.

Кликането върху стрелката отдясно отваря списък с последните 10 използвани файла. Желаният файл се избира чрез кликане върху него в списъка.

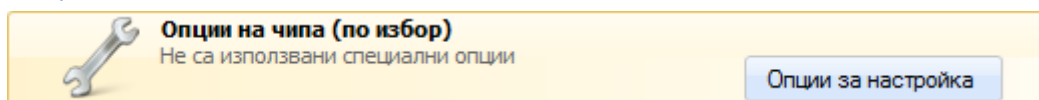
В това поле можете да използвате специални символи, които ще бъдат автоматично заменени:

! Удивителният знак ще бъде заменен с името на избрания чип.

# Диез ще бъде заменен с 1 или по-голямо число. Ако вече съществува файл с такова име, числото ще бъде увеличавано с единица до получаване на уникално име на файл.

Пример: Четете данните от AT27C010 и въвеждате име на файл "File-!-#.bin" в диалога за запазване на файл. Данните ще бъдат запазени под името „File-AT27C010-1.bin“. Ако прочетете и запазите същия чип отново, данните ще бъдат запазени под името "File-AT27C010-2.bin".

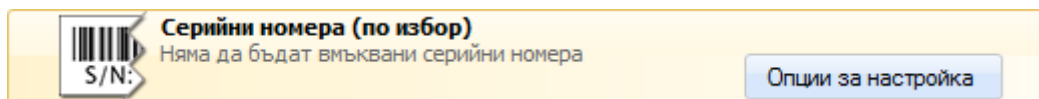
## Опции на чипа



Можете да използвате специални опции на чипа, за да промените данните за записване или да използвате специални функции на чиповете. Ако не сте сигурни за какво се използват тези настройки или ако само искате да запишете данни върху чип, не активирайте специалните опции.

Кликането върху бутона отваря прозорец с настройки за опциите на чипа. Повече информация можете да откриете в главата „Опции на чип“.

## Серийни номера

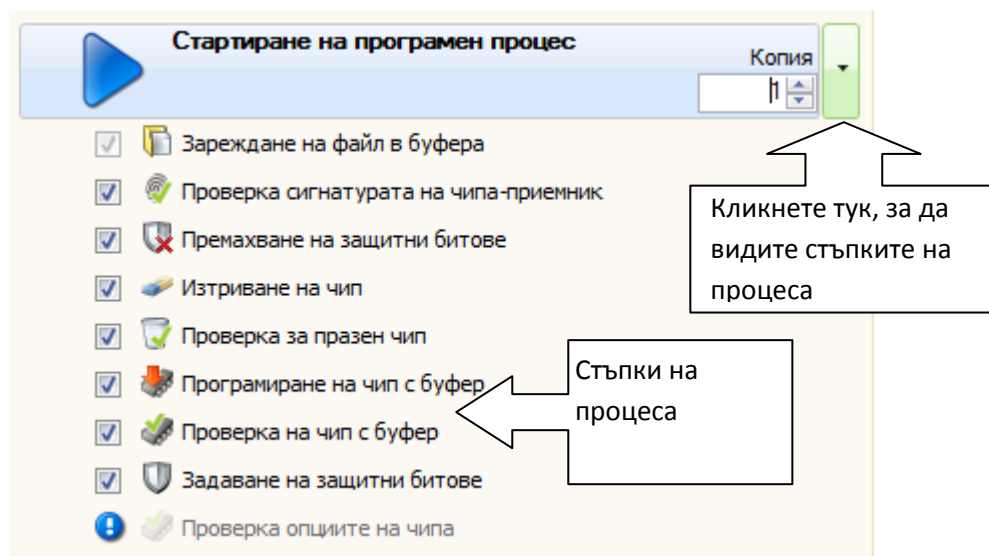


За производствени цели към данните могат да се прибавят серийни номера преди да бъдат записани.

Кликането върху бутона отваря прозорец с настройки за серийни номера. Повече информация можете да откриете в главата „Серийни номера“.

## Управление на процес

Управлението на процеса е сходно за софтуерните режими „Програмиране на чип“, „Копиране на чип“, „Четене на чип“ и „Производствен режим“ и е описано в тази глава. Всеки от тези режими има синя лента с символ за стартиране (син триъгълник).





Кликането върху стрелката вдясно отваря списък, който показва всички стъпки на процеса. Когато се стартира процес този списък се отваря автоматично.



Всеки процес се състои от отделни стъпки. Тези стъпки могат да бъдат активирани или деактивирани чрез отметките пред тях. Всяка стъпка може да бъде изпълнявана самостоятелно и независимо от цялостния процес с кликане върху бутона на желаната стъпка.

Полето „Копия“ може да се използва за определяне на броя пъти, за които процесът да бъде изпълнен. В режим „Програмиране“ числото определя броя на чиповете, които ще бъдат програмирани, а в режим „Копиране“ – броя на копията на чипа-източник, които ще бъдат направени.

### Контроли

-  Избраните стъпки се изпълняват отгоре надолу след стартиране на процеса с този бутон.
-  Кликането върху този символ (видим само по време на изпълнение на процес) спира процеса и предотвратява програмирането на още чипове.

### Специални контроли (Само в „Производствен режим“)

-  Кликането върху този символ (видим само по време на изпълнение на процес) спира процеса и предотвратява програмирането на още чипове. Въпреки това активната стъпка от процеса ще бъде завършена.
-  Този бутон се появява само след натискане на стоп бутона и единствено в производствен режим. Той показва, че нормалната последователност на процеса е спряна и само активната стъпка ще се завърши. Натискането на този бутон ще прекрати незабавно всички протичащи процеси.

## Стъпки на процес



### Зареждане на файл в буфера

Файлът, посочен в полето за отваряне на файл, се зарежда в буфера по време на тази стъпка.



### Запазване на данните от буфер във файл

Данните от буфера се запаметяват във файл. Несъществуващите файлове се създават автоматично, а съществуващите – се презаписват.



### Проверка на сигнатурата на чипа-източник/приемник

Сигнатурата на поставения чип се сравнява с настройката, определена за „Чип-източник“/„Чип-приемник“. Ако бъде открито несъответствие, се появява прозорец с опции за отмяна на процеса, повторна проверка на сигнатурата и игнориране на несъответствието и продължаване.



### Изтриване на чип

Този процес изтрива чипа, поставен в програмиращото устройство. По принцип изтриване на чип означава, че на всички битове ще бъде зададена стойност 1. Следователно всички байтове в изтрят чип са FFh (Байт FFh = 11111111).

Не всички чипове могат да бъдат изтрити чрез програмиращото устройство. Например EPROM 27с със стъкло прозорче могат да бъдат изтрити единствено чрез силна UV-C светлина в специална апаратура за изтриване на EPROM. EPROM 27с без стъкло прозорче не могат да бъдат изтрити и следователно не могат да бъдат програмирани с нови данни. Тези чипове обикновено се маркират като OTP (One Time Programmable, програмируеми само веднъж).



### Проверка за празен чип

По време на тази стъпка се проверява дали чипът е изтрят, т.е. дали всички битове имат стойност 1. Цялото съдържание на чипа се прочита и всички битове се проверяват.



### Четене на данни от чипа от буфера

Чипът-източник се прочита и данните му се зареждат в буфера. Данните от буфера могат да бъдат преглеждани и редактирани чрез Нех-редактора (виж глава „Нех-редактор“).



### Програмиране на чип с буфер

Данните от буфера се записват върху чипа като се използват всички настройки, зададени в „Опции на програмиране“ и „Серийни номера“.



### Проверка на чип с буфер

Цялото съдържание на чипа се прочита и всички байтове се сравняват с тези в буфера.

 **Задаване на защитни битове**

С тази стъпка се задават така наречените защитни битове, които предотвратяват възможността чипът да бъде случайно презаписан по-късно. Тази функция не се поддържа от всички чипове.

 **Показване на буфера на Hex-редактора**

Софтуерът превключва в режим на Hex-редактор.

 **Автоматично изчакване за поставяне на чип**


Съответното програмиращо устройство изчаква докато бъде поставен нов чип. Когато това се случи се проверява сигнатурата на чипа. Ако желаните чипове не съдържат сигнатура, отстраняването и поставянето на тези чипове не може да бъде засичано автоматично. За тези чипове функцията „Автоматично изчакване за поставяне на чип“ не може да се активира. В тези случаи процесът се стартира от стартовия бутон на съответното устройство.

 **Автоматично изчакване за отстраняване на чип**

Съответното програмиращо устройство изчаква докато програмираният чип бъде отстранен.

Когато процесът е завършен светлината, индикираща, че програматорът работи, започва да премигва. Това показва, че чиповете могат да бъдат премахнати. Съответното програмиращо устройство изчаква докато чипът бъде премахнат. Ако след отстраняване на чипа зеленият диод на програматора премигва, това показва, че устройството изчаква за следващия чип за програмиране. Ако зеленият диод не премигва, не са зададени други чипове за програмиране от устройството.

Ако желаните чипове не съдържат сигнатура, отстраняването и поставянето на тези чипове не може да бъде засичано автоматично. За тези чипове функцията „Автоматично изчакване за отстраняване на чип“ не може да се активира.

 Този символ се изобразява до стъпката на процеса, ако тази стъпка не се поддържа от избрания чип или програмиращо устройство.

## Hex-редактор

Hex-редакторът се използва за преглед и редактиране на двоични данни. Той предоставя разнообразни функции за екрана, както и много лесни за използване функции за редактиране на данни.



**Буфер Hex-редактор**

Файл Редактиране Изглед Програматор

Чексума: 8A69

Intel P28F001BX-T

ProgramData1MBit.bin

H	Адрес	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Windows-1252
00000000	45 B2 C9 C6 27 F0 8C 12 FC B3 A2 53 E8 8F A9 15																	É'É.ü'cSè.©.
00000010	EB E2 34 32 1F CE F0 D6 D3 42 30 F3 A0 22 07 34																	ëâ42.Î8ÖÓB0ó ".4
00000020	AF 23 42 34 B2 E7 5D 23 45 30 8B CD 4E 58 FD FD																	~#B4'ç] #EO<INXýý
00000030	42 34 28 81 59 33 D7 85 F2 83 67 3E 7F 85 59 E0																	B4 (.Y3×...ðfg>...Yà
00000040	AC 1C 60 62 AF B5 5D 1E 94 8E 92 3D 5C 35 66 08																	~. 'b~u] ."Z'=\5f.
00000050	82 45 D6 7A 48 B8 E7 8E 0A F9 E2 EB C7 83 41 F2																	,EÖZH,çŽ.ùâæÇfAð
00000060	83 DC 4D 00 63 89 3E 8C C4 6A EB C9 A0 9A 28 BD																	fÜM.c%>EÄjæÉ š (%
00000070	1C 5F BA 3A AE DD 97 F2 33 D0 57 B5 D2 70 56 10																	. °:©Ý-ð3ÐWµOpV.
00000080	4C 9E B8 7E E6 66 A4 F8 06 49 B5 DD CC 1D ED 6F																	Lž,~æfæø.IpÝÏ.ío
00000090	D0 25 CF 68 C8 07 44 A5 45 59 CB 91 31 88 B9 54																	Ð\$ÎhÈ.D\$EYË'1^'T
000000A0	14 DA 7C 01 E0 A9 D5 B5 B9 C0 33 64 28 B9 26 16																	.Ú .à©Öµ'À3d(²&.
000000B0	E1 E6 E3 AC A0 EC 47 45 5A 12 38 8D 9A 12 C6 7D																	áæä~ iGEZ.8.š.É}
000000C0	8F 60 C9 F0 9F EC 8D 27 4F 19 AD 08 D2 DE 7D 8B																	. 'Éšÿi.'O. .ÖB}<
000000D0	32 B4 AE 4A A3 0C B3 26 93 37 A5 CE 70 47 1B A7																	2'©Jf.²&"7\$ÏpG.\$
000000E0	F9 B5 D4 C6 EE 35 75 F4 58 BB 5D 86 3F 3E 71 7B																	ùµÖEi5uðX» †?>q{
000000F0	93 AC 5A E4 62 7C D8 82 AC A2 5C 68 46 BE B6 07																	"~Zäb Ø,~c\hF%¶.
00000100	FD 06 0E 1C 09 B6 9B FC 3F B9 F0 E7 AE 2C E6 54																	ý....¶>ü?'ðçð,æT
00000110	C4 2A 3A C2 EC 35 45 7C DC 51 19 4C D7 0D AE 98																	Ä*:Äi5E ÜQ.I×.©~
00000120	05 55 87 39 30 54 71 DC 11 5D A5 A8 8B F0 D3 80																	.U+90TqÜ. ]\$~<ðÖ€
00000130	BB 08 A2 1B 84 F2 1A C7 C0 90 3F B7 9C 4F 24 31																	».c.„ð.ÇÀ.?·æO\$1
00000140	0A 57 E2 58 A4 AC 23 6F B8 5C B1 8E 8F 9B F0 EB																	.WâX~#o,\±Ž. >ðë
00000150	7D 98 8F CF 6A 92 14 11 3B BE 78 9F 15 B8 3F C1																	}~.Ïj'...;xÿ. ?Ä
00000160	38 74 9D 33 0D 7A DA 0E E3 32 62 27 41 8A 38 24																	8t.3.zÚ.ä2b'AS8\$
00000170	9E 53 39 04 BE 54 C5 76 43 D0 1C E4 34 83 5A FE																	žS9.%TÄvCÐ.ä4fZp

0-1FFFFh, 128 KByte ( 1 MBit ), Бинарен файл Текуща селекция: HEX 0 до HEX 0

Функциите за редактиране могат да бъдат избрани чрез бутоните от инструменталната лента или от контекстното меню. Контекстното меню е достъпно чрез натискане на десния бутон на мишката.







Текущият маркиран адрес се показва в статус лентата. Кликането в полето за адрес позволява въвеждане на точен адрес, който да бъде показан.

Когато се използват функции на програмиращото устройство, текущата стъпка от процеса се изписва в статус лентата, заедно с развитието на процеса (отдолу вдясно).

## Меню и инструментална лента на Нех-редактора

Инструменталната лента съдържа функции за управление и работа с данни в Нех-редактора.










### Меню и инструментална лента „Файл“

	Нов	Отваря нов празен прозорец в Нех-редактора.
	Отваряне на файл	Отваря съществуващ файл. Списък с последните 10 използвани файла може да бъде отворен чрез кликане върху стрелката отлясно на този бутон.
	Затваряне	Затваря текущия документ.
	Затваряне на всички документи	Затваря всички отворени документи.
	Затваряне на всички останали	Затваря всички отворени документи без текущия.
	Запазване като файл	Запазва текущото съдържание на буфера.
	Запазване като...	Запазва данните под ново име на файл.  Файлов формат: Файловият формат може да бъде засечен автоматично (по подразбиране) или определен ръчно. Настройката се използва за всички функции за отваряне и запазване в Нех-редактора.
	Разпечатване	Разпечатва текущия документ.
	Вмъкване на файл	Вмъква съществуващ файл с офсет и/или допълнително към данните в Нех-редактора.
	Сравняване на данните от Нех-редактора с тези от файла	Сравнява данните от Нех-редактора с тези от съществуващ файл.
	Файлов формат	Предоставя възможност за избор между автоматично засичане на файловия формат или определяне на точен формат за отваряните файлове.
	Текущи файлове	Последните използвани файлове могат да бъдат отворени чрез този диалог.

## Меню и инструментална лента „Редактиране“

	Назад	Връща назад последната операция.
	Напред	Изпълнява отново последната операция, която е била върната назад.
	Изрязване	Изрязва избраните данни.
	Копиране	Копира избраните данни.
	Копиране със запазване на форматирането	Копира избраните данни, заедно с форматирането им.
	Поставяне	Поставя копираните данни на мястото на курсора.
	Изтриване	Изтрива избраните данни.
	Избиране на всички	Избира всички данни в редактора.
	Ръчен избор	Избира определена част от данните.
	Търсене и заместване	Отваря диалог за търсене и заместване.
	Показване на първия байт...	Преминава на първия байт от данните, който не е FFh.
	Показване на следващия байт...	Преминава на следващия байт от данните, който не е FFh.
	Показване на последния байт...	Преминава на последния байт от данните, който не е FFh.
	Калкулиране на чексума	Калкулира чексума за избраната област.
	Вмъкване в	Вмъква определен брой данни със зададена стойност.
	Запълване на селекцията	Избраната област се запълва със зададени стойности, произволни числа или поредица от числа.
	Увеличаване/Намаляване на стойности	Стойностите от избраната област се увеличават/намаляват с определена стойност или проценти.
	Копиране на избраното	Избраното се копира на зададен адрес.
	Размяна на данни	Думите, които са една до друга в избраната област, се разменят.
	Размяна на блокове данни	Избраната област се разменя с друга.

## Меню и инструментална лента „Изглед“

Изглед за редактиране	Показва данните в Нех-редактора. Тази опция е видима само за активиран модул за чип-тунинг!
2D изглед	Показва данните в 2D изглед. Тази опция е видима само за активиран модул за чип-тунинг!
3D изглед	Показва данните в 3D изглед. Тази опция е видима само за активиран модул за чип-тунинг!
Думи	Настройва дължината на думите
	Показва данните като байтове (8 bit).
	Показва данните като думи (16 bit).
	Показва данните като думи (24 bit).
	Показва данните като двойни думи (32 bit).
	Показва данните като четворни думи (64 bit).
Байтове	Настройва реда на байтовете. В зависимост от целевата система първият байт може да бъде най-ниският (LSB) или най-високият (MSB).
Широчина на екрана	Широчината на редактора може да бъде увеличавана или намалявана.
	Намаля броя на байтовете, показвани на един ред от редактора.
	Увеличава броя на байтовете, показвани на един ред от редактора.
Офсет	
Увеличаване	Премества позицията на първия байт на екрана надясно.
Намаляване	Премества позицията на първия байт на екрана наляво.
	Дава възможност за избор на изгледа на лявата част от екрана на редактора.
	Дава възможност за избор на изгледа на дясната част от екрана на редактора.
 Синхронизиране на превъртането...	Превъртането на всички видими редактори ще бъде синхронизирано. По този начин данните могат да бъдат сравнявани ръчно.

## Меню и инструментална лента „Програматор“

Всички основни функции, необходими за програмирането на чип, са достъпни и в Нех-редактора. Използвайте падащото меню отляво, за да изберете чип, с който да работите. От тук имате достъп до чип браузъра за улеснение на избора.



Свързани  
програматори

Избор на програматор

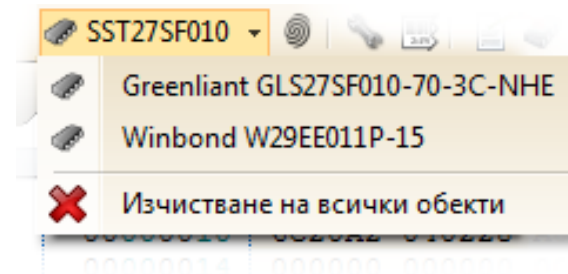


Показване на  
програматор

Зеленият диод на избрано устройство започва да мига, за да бъде идентифициран.



(Инструментална лента) Показва изборът от чип браузъра. С натискане на стрелката от дясната страна се отваря списък с последните използвани чипове.



Чип браузър

Отваря чип браузъра.



Автоматично  
идентифициране

Автоматично идентифициране на чип (виж глава „Автоматично идентифициране на чип“).

Последни 10  
използвани чипа

Показва последните 10 използвани чипа за директно избиране.



Опции на чипа

Тук можете да настроите специалните опции на чипа.



Серийни номера

Тук можете да зададете серийни номера на чипове.



Стартира цял програмен процес. Настройките по подразбиране на процеса определят стъпките „Проверка сигнатурата на чипа“, „Премахване на защитни битове“, „Изтриване на чип“, „Проверка за празен чип“, „Програмиране на чип с буфер“, „Проверка на чип с буфер“ и „Задаване на защитни битове“. Стъпките, които не се поддържат от чипа, автоматично се деактивират. С натискане на черната стрелка можете да отворите списъка със стъпките на процеса и да активирате/деактивирате стъпки.



Стартира пълен процес на четене. Настройките по подразбиране на процеса определят стъпките „Проверка сигнатурата на чипа“, „Четене на данни от чипа“ и „Проверка на чип с буфер“. С натискане на черната стрелка можете да отворите списъка със

стъпките на процеса и да активирате/деактивирате стъпки.



Проверка сигнатурата на чипа

Проверява сигнатурата на чипа.



Изтриване на чип

Изтрива чипа.



Проверка за празен чип

Проверява дали чипът е изтрит.



Четене на данни от чипа...

Чете данните от чипа в буфера.



Програмиране на чип...

Програмира данните от буфера върху чипа.



Проверка на чип...

Сравнява данните от буфера с тези от чипа.



Задаване на защитни битове

Задава защитни битове.



Премахване на защитни битове

Премахва защитни битове.



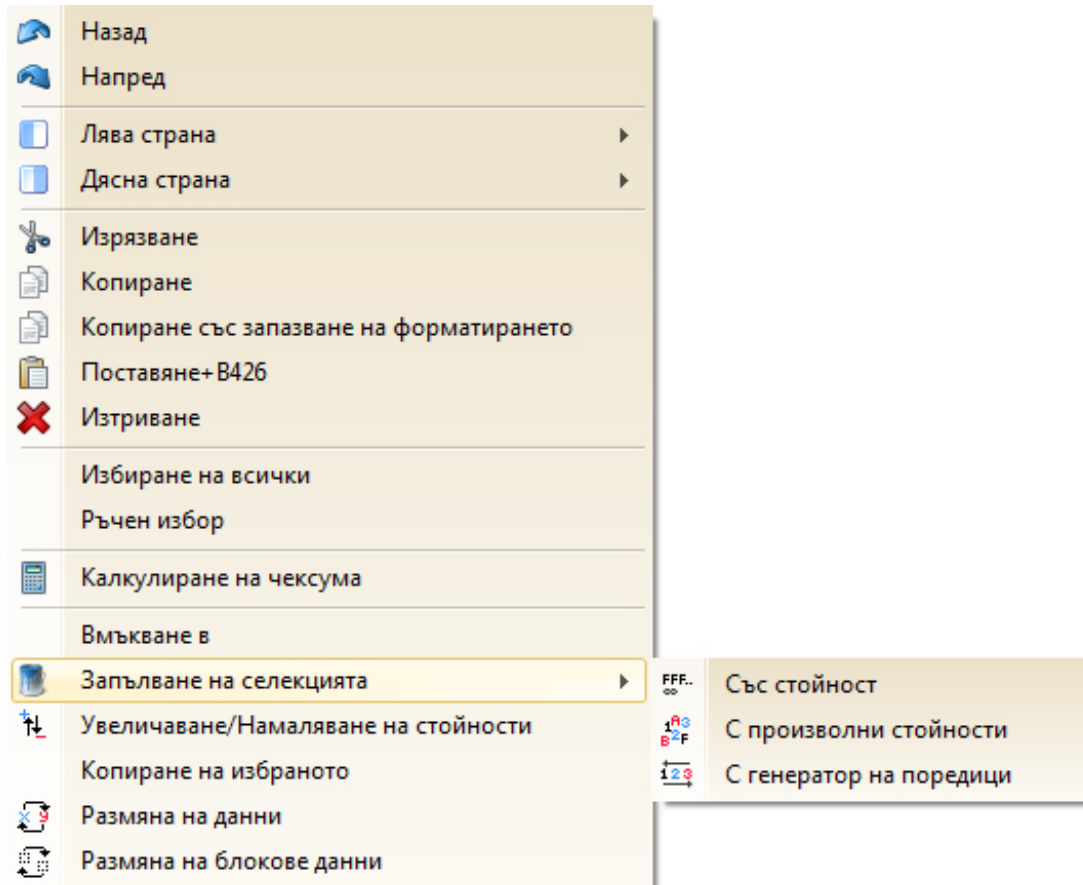
Спиране на текущия процес

Спира текущия програмен процес (не се отнася за действащи процеси в други модули!)

## Контекстно меню на Hex-редактора

Можете да отворите контекстното меню с натискане на десния бутон на мишката в Hex-редактора.

Достъпните функции бяха описани в главата „Меню и инструментална лента на Hex-редактора“.

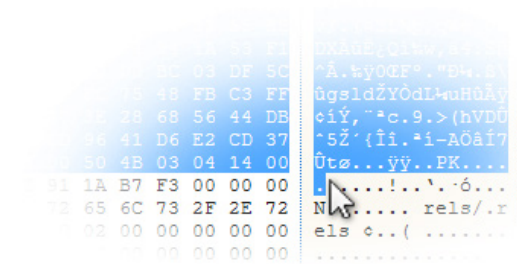


## Работа със селекция

Тези опции могат да бъдат избрани и чрез инструменталната лента, и чрез контекстното меню.

### Избор на области

#### Избор с мишката



В Нех-редактора определена област от шестнайсетичен код може да бъде избран и редактиран. Избраната част се оцветява в син фон.

#### Точен избор чрез въвеждане



Област може да бъде избрана чрез въвеждане на началната и крайната позиция. С тази функция лесно могат да се управляват големи области. Полетата за въвеждане на позициите се намират в долния десен ъгъл.

### Редактиране на области

Тази опция е достъпна от менюто „Редактиране“, от инструменталната лента и от контекстното меню.

#### Ръчен избор

Тук можете да определите точно кои данни трябва да бъдат избрани.

#### Калкулиране на чексумата на селекция

Калкулира чексумата на избраната област по определен алгоритъм (сума, MD5, SHA-1, EPT1 CRC16 или CRC32).

#### Запълване на селекцията – със стойност

Запълва избраната област с определена стойност

#### Запълване на селекцията – с произволни стойности

Запълва избраната област с произволни стойности

#### Запълване на селекцията – с генератор на поредици

Запълва избраната област с определена поредица от стойности

#### Увеличаване/Намаляване на стойности

Увеличава или намалява всички стойности в избраната област с определена стойност или процент.

## Копиране на избраното

Копира избраните данни на определен адрес. Можете да избирате дали целевата област да бъде разширена или презаписана.

## Размяна на данни

В зависимост от настройките разменя стойностите на първия и втория байт/дума/двойна дума или четворна дума. Ако са избрани повече от една двойки размяната продължава по същия начин (например байт 1 се разменя с байт 2, байт 3 с байт 4 и т.н.).

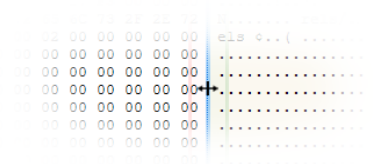
## Размяна на блокове данни

Разменя избраните данни с данни, започващи от зададен адрес.

## Промяна на изгледа чрез мишката

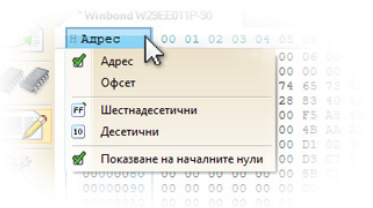
В допълнение на опциите за настройка от меню „Изглед“ някои настройки в Hex-редактора могат да бъдат направени директно с мишката.

### Регулиране широчината на екрана



С мишката можете да промените широчината на екрана. Натиснете разделителната лента между лявата и дясната област в Hex-редактора. Същата функция се поддържа и от най-дясната вертикална линия.

### Регулиране изгледа на адресите



С натискане върху заглавието „Адрес“ на най-лявата колона се отваря меню. С избора на „Адрес“ или „Офсет“ се превключва между адрес дума и адрес байт. В случай на дума с дължина един байт (8 Bit) изгледът е идентичен. Възможно е също така адресите, показвани с шестнайсетични или десетични числа, да се изобразяват с или без началните нули.

## Клавиши за бърз достъп в Нех-редактора

Нех-редакторът поддържа следните клавиши за бърз достъп:

- <CTRL+Z>: Назад
- <CTRL+Y>: Напред
- <Page Up>: Премества се до първия ред от текущата страница. Ако курсорът вече е на първия ред, екранът се предвижва нагоре с цяла страница.
- <Page Down>: Премества се до последния ред от текущата страница. Ако курсорът вече е на последния ред, екранът се предвижва надолу с цяла страница.
- <Home>: Премества курсора на първия байт от текущия ред.
- <CTRL+Home>: Премества курсора на първия байт в Нех-редактора.
- <End>: Премества курсора на последния байт от текущия ред.
- <CTRL+End>: Премества курсора на последния байт в Нех-редактора.
- <Insert>: Превключва между режими на презаписване и вмъкване. В режим презаписване (настройка по подразбиране) данните на текущата позиция на курсора се презаписват при въвеждане на нови данни. В режим вмъкване новите данни се вмъкват без да се презаписват съществуващите.
- <CTRL+A>: Избира всички данни.
- <Shift+Стрелка>: Разширява селекция.
- <CTRL+X>: Изрязва избраните данни и ги копира в клипборда.
- <CTRL+C>: Копира избраните данни в клипборда.
- <CTRL+V>: Поставя данни от клипборда.
- <Delete>: Изтрива избраните данни.
- <Tab>: Превключва между HEX и ANSI полетата.
- <CTRL+H>: Изглед – шестнайсетични числа
- <CTRL+D>: Изглед – десетични числа
- <CTRL+T>: Изглед - текст
- <CTRL+Стрелка надясно/наляво>: Разширява/намалява широчината на екрана
- <CTRL+Alt+ Стрелка надясно/наляво >: Увеличава/намалява офсета
- <CTRL+1>: Дължина на думата 1 байт (8 Bit)

- <CTRL+2>: Дължина на думата 2 байта (16 Bit)
- <CTRL+3>: Дължина на думата 4 байта (32 Bit)
- <CTRL+4>: Дължина на думата 8 байта (64 Bit)

## Софтуерни опции

Софтуерните опции могат да се използват за конфигуриране настройките при стартиране и изход от Prog-Express, настройките на съобщенията, използваните аудио файлове, специалните опции на програмиращите устройства, оперативния режим и езика.



Софтуерните опции са разделени в няколко отделни страници „Основни“, „Проект“, „Защита“, „Файлови връзки“, „Език“ и „Добавки“.

### Основни опции

**Софтуерни опции**

Основни | Проект | Защита | Файлови връзки | Език | Добавки

Опции при стартиране на Prog-Express

☒ Автоматично зареждане на последните настройки  
☐ Автоматично зареждане на последния използван проект  
☐ Автоматично зареждане на следния проект:

☐ Промяна на изгледа на прозорците на Batronix

Опции при изход на Prog-Express

☒ Автоматично запазване на файл на проект

Работна директория

☐ Отваряне на файловите прозорци в работна директория

Автоматично обновяване на софтуера

☒ Търсене на актуализации за софтуера онлайн при стартиране

Съобщения

☒ Показване на предупреждение, ако данните от буфера са по-големи от размера на чипа по време  
☐ Показване на предупреждение, ако чипът трябва да бъде поставен на необичайна позиция.

Звук

☒ Възпроизвеждане на аудио файл, когато даден процес е завършен успешно:

☒ Възпроизвеждане на аудио файл, когато даден процес е неуспешен:

### Опции при стартиране на Prog-Express

Софтуерът може да зарежда последния използван проект или определен проект при стартиране. Всеки проект съдържа избрания режим, избраните чипове, опциите на програмиране, опциите за серийни номера и активираните стъпки на процеса.

### Изглед на прозорците на Batronix

Потребителят може да променя изгледа на стандартния прозорец на Windows със специалния дизайн на програмата.

### Опции при изход от Prog-Express

При изход от Prog-Express софтуерът може автоматично да запазва точните настройки в последния зареден проектен файл (по подразбиране).

### Работна директория

Софтуерът може да зарежда търсачката за проекти винаги в определена директория. Ако тази опция не се използва, търсачката зарежда последната използвана директория.

### Автоматично обновяване на софтуера

Софтуерът може да проверява за актуализации онлайн, когато го стартирате. Ако е на разположение актуализация, тя може да бъде свалена и инсталирана автоматично.

### Съобщения

Показването на определени съобщения може да бъде включвано и изключвано от тук.

### Звук

След завършване на процес или след грешка по време на програмиране софтуерът може да възпроизведе звук. Някои файлове идват заедно със софтуера Prog-Express в подпапката „Sounds“. Можете да избирате и собствени .wav аудио файлове от компютъра си.

## Проект

**Софтуерни опции**

Основни

Проект

Защита

Файлови връзки

Език

Добавки

Управление на данни

Запълване на неизползваните байтове с:  

FF

hex

Дублиране на буферните данни върху свободните области от по-големите адреси на чипа  
☐ Активирайте тази опция, ако искате да програмирате данните от чип на по-голям заместващ чип.  
Ред на байтове при 16-битови чипове:  

☒ Програмиране на младшите байтове преди старшите (LSB-MSB, стандартно)  
☐ Програмиране на старшите байтове преди младшите (MSB-LSB)

Multi-Pass проверка (само за BX32-II, BX32P-II, BX40-I/II, BX48-I/II, BX448 & BX848)  

☐ Проверка при минимално напрежение  
☒ Проверка при номинално напрежение  
☐ Проверка при максимално напрежение

Функции за сигурност и тестване  

☒ Тестване на връзките на крачетата (само за BX48, BX448 & BX848)  
☒ Активиране на мониторинг за ток на претоварване (само за BX48, BX448 & BX848)

Ръководство за употреба на Prog-Express © 1998 – 2012 от Batronix Elektronik, [www.batronix.com](http://www.batronix.com)

Този документ е защитен от немските и международни закони. Непълна или частична употреба, продажба или възпроизвеждане, публикуване – включително откъси – от какъвто и да е род изискват писмено разрешение от автора.

38

### Управление на данни

Няколко специални настройки могат да бъдат определени тук.

Неизползваните байтове са байтове, за които не са определени никакви данни. Това може да се случи, когато например по-малък файл се програмира върху по-голям чип.

Опцията за ред на байтовете е приложима само при програмиране на 16-битови чипове. Тук можете да задавате в какъв ред да се използват байтовете от файла.

### Multi-Pass проверка

VX40 поддържа Multi-Pass проверка. По време на този процес данните от чипа се прочитат и сравняват няколко пъти при различни захранващи напрежения. Ако процесът открие разлики в данните проверката се прекратява и се показва грешка.

Използваните напрежения зависят от позволения диапазон на оперативни напрежения за чипа. Например много чипове позволяват диапазон от  $\pm 10\%$ . В този случай например ако номиналното напрежение е 5V, проверката може да бъде извършена при 4.5V, 5.0V и 5.5V. Повтарящата се проверка на програмираните чипове предоставя по-голяма сигурност за програмираните данни. Това елиминира чипове, които не са програмирани 100% и по този начин могат да се развалят в крайното устройство под условия на силно променливо напрежение.

### Функции за сигурност и тестване

Някои програматори могат да тестват контакта на крачетата преди да започнат процес и да следят тока през чипа по време на процеса. Тези функции повишават сигурността и често трябва да бъдат активирани. Въпреки това в някои специални случаи може да е от полза те да бъдат деактивирани. Например ако входният ток на потенциално развален чип е много по-висок от определения от производителя, мониторингът за ток на претоварване може да бъде изключен. Така чипът може да бъде използван, въпреки че изисква повече ток от посочения от производителя.

## Защита

**Софтуерни опции**

Основни

Проект

Защита


Файлови връзки


Език

Добавки

Оперативен режим

Настройка	Софтуер	Проект
Активиране на Оперативен режим при стартиране на приложение	<input type="checkbox"/>	<input type="checkbox"/>
Заклучване избора на режим с парола	<input type="checkbox"/>	<input type="checkbox"/>
Заклучване на диалога за опции и Оперативния режим с парола	<input type="checkbox"/>	<input type="checkbox"/>
Парола	<input type="text"/>	<input type="text"/>

 Натиснете символа за заключване от заглавната лента на Prog-Express, за да активирате Оперативен режим.

 Ако опция от Оперативния режим е достъпна в софтуера или в контекста на проект, Prog-Express винаги използва най-рестриктивната.

### Оперативен режим

С активиране на оперативния режим софтуерът е защитен срещу промени, било то желани или допуснати по невнимание. Този оперативен режим се препоръчва при производства, където софтуерният инженер наглася всички настройки, които операторът използва за ръководене на серийното производство.

Докато е активиран оперативен режим файлът, избраните чипове, опциите на програмиране, настройките за серийни номера и активираните стъпки на процеса са защитени срещу промени. В софтуерните опции промяната на режима също може да бъде деактивирана.

Оперативният режим може да бъде активиран и деактивиран с натискане на символа за заключване (катуна) от заглавната лента на Prog-Express. Ако е активирана защита с парола, софтуерът ще поиска парола преди да активира или деактивира оперативния режим.

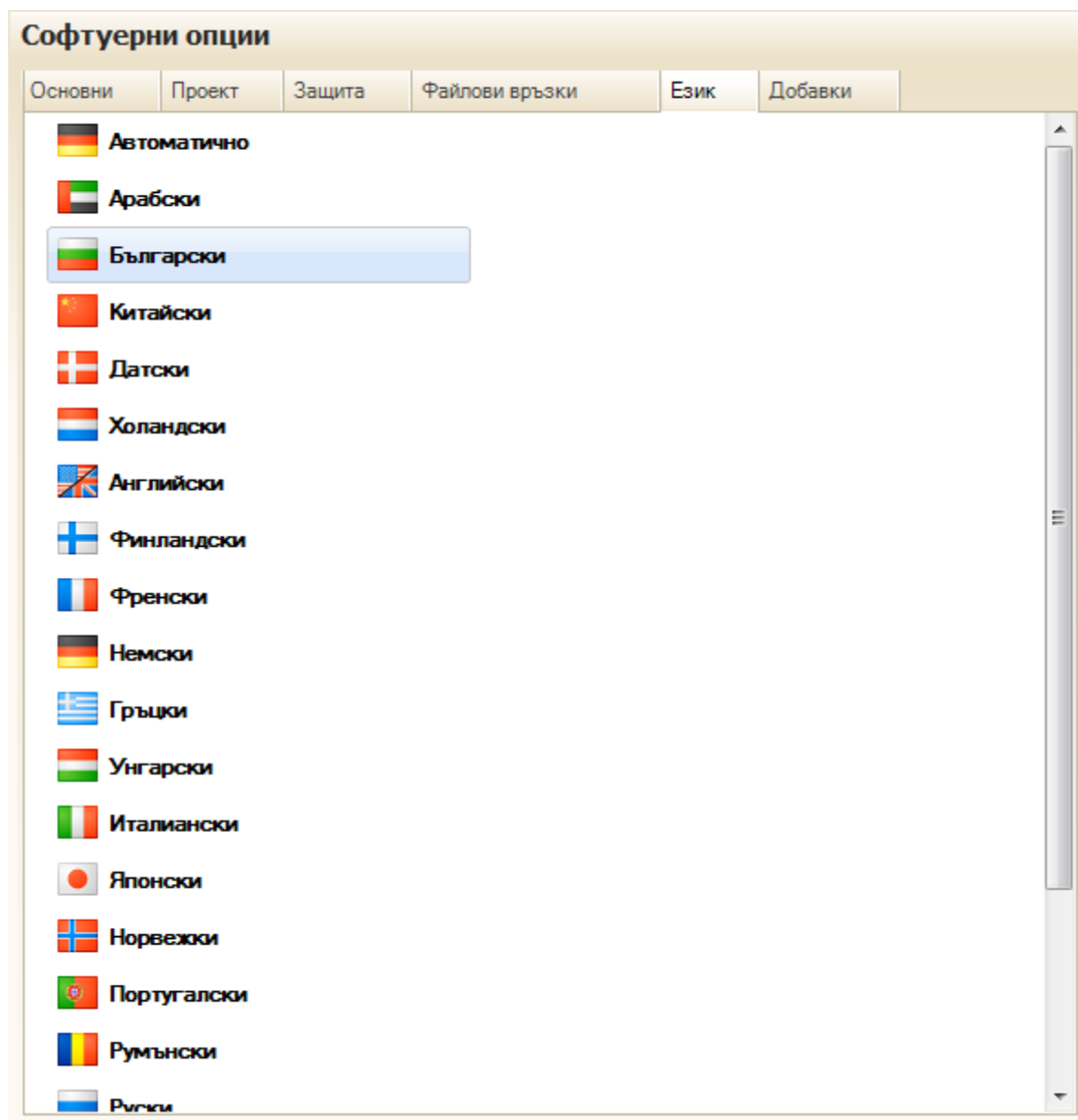
Настройките могат да се използват за проект или за софтуера като цяло. Когато съществува опция за заключване на оперативния режим за проект или за софтуера, Prog-Express винаги използва най-рестриктивната.

## Файлови връзки



В тази секция различни файлове могат да бъдат свързани с Prog-Express. Двойно кликане върху такъв свързан файл ще стартира Prog-Express автоматично. За да се правят промени в тази секция се изискват администраторски права.

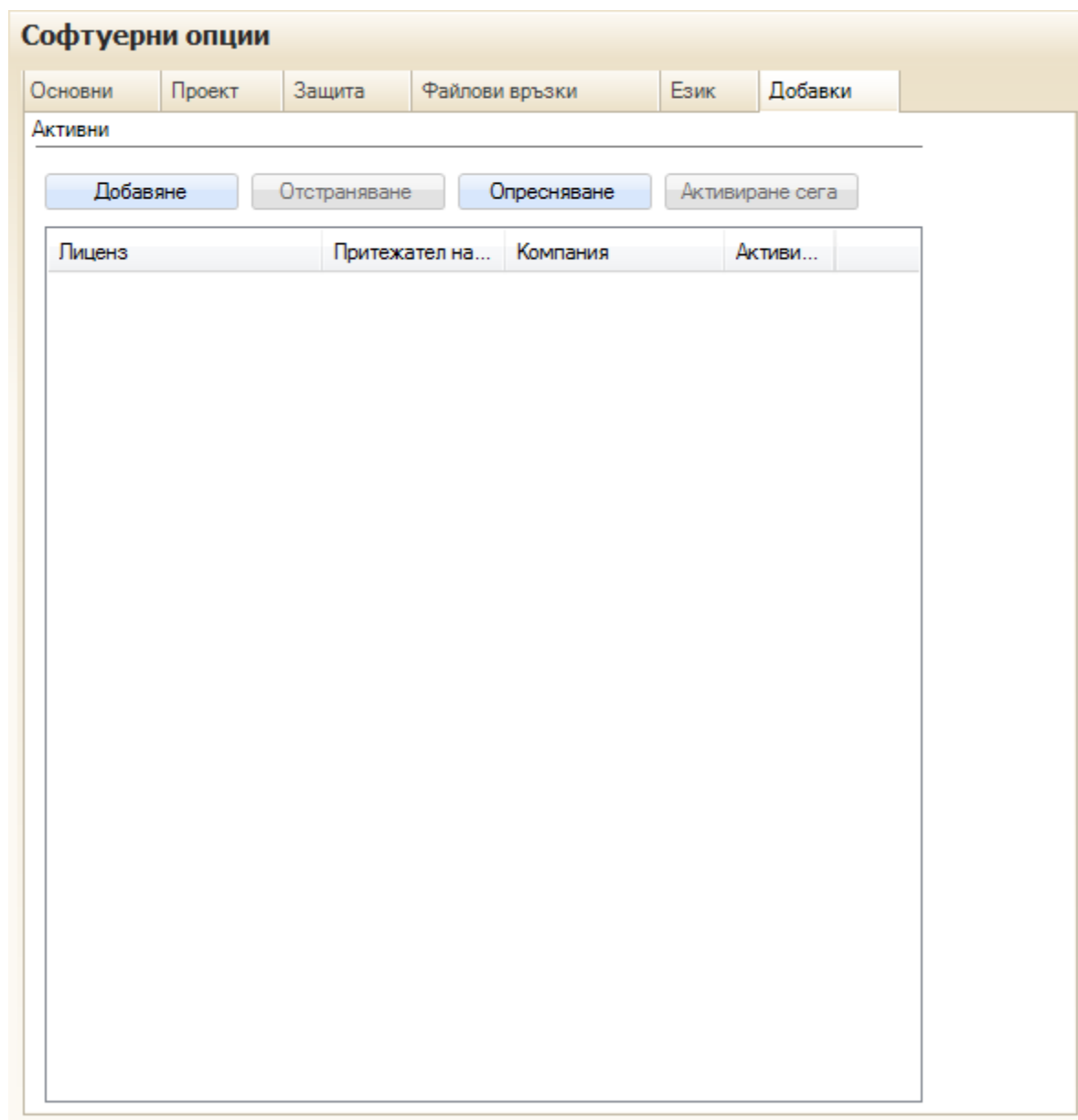
## Опции за език



Желаният език може лесно да бъде избран чрез кликане върху него. Ако е възможно, свалете най-новата версия на Prog-Express от нашия сайт [www.batronix.com](http://www.batronix.com), за да избирате от всички поддържани за момента езици.

Първата опция „Автоматично“ използва езиковите настройки на операционната ви система. Останалите езици са подредени по азбучен ред в съответствие с изписването им на английски език.

## Добавки



От секцията „Добавки“ е възможно добавяне или активиране на допълнителни модули за повече функции на софтуера.

## Лог на действията

В полето за лог на действията всички подробности за вече завършени или за действащи процеси ще бъдат записвани.

С натискане върху символа плюс или минус разклоненията могат да бъдат показвани или скривани. След започването на нов процес, софтуерът ще скрие разклоненията на последния процес.

За всеки процес има едно информационно разклонение и по едно разклонение за всяка стъпка от процеса. Информационното разклонение съдържа основна информация за процеса като използван програматор, чип, файл и чексуми.

Чексумата на файла може да бъде различна от чексумата на чипа поради няколко причини. Файлът може да е по-голям и да съдържа повече данни от чипа, ако изберете да програмирате само част от чипа. Може да е различна и ако използвате специални опции за програмиране или серийни номера. Функцията „Проверка на чип с буфер“ гарантира правилно програмиране. Като цяло можем да кажем, че функцията „Проверка на чип с буфер“ на Prog-Express е много по-сигурна от сравняването на чексумата на файла с тази на чипа, тъй като функцията за проверка сравнява всички байтове точно, а не само чексумите на данните.

Информационното разклонение съдържа времето на процеса, с включено време за потвърждаване на съобщенията от потребителя и т.н.... Времето за всяка стъпка от процеса се отбелязва в съответното разклонение за стъпката.

Отхвърлените процеси се отбелязват с червен знак стоп, неуспешните – с червен X, а процесите, които са завършени успешно – с зелена отметка.

С натискане на червения X целия лог на действията може да бъде изтрит.

Успешни: 1

Неуспешни: 0

✖

📄

✓

Стартиране на процес от 09:18:04

📁

Информация

📁

BX32 Baturo II

📄

Серийн номер BL2507791

📄

Версия на фърмуер 02.30

📁

Чип-приемник Winbond W29EE...

📄

Производител Winbond

📁

Технология Flash

📄

Проверена чексума

📄

Сума: 8A69

📄

EPT-1 CRC-16: 80E2

📄

CRC-32: 9F00202D

📁

Зареждане на файл в буфера

📄

Проверка на сигнатурата на чипа-...

✓

OK

🕒

0,087 с.

✖

Отстраняване на защитните битове

✓

OK

🕒

0,353 с.

📁

Изтриване на чипа

✓

OK

🕒

0,319 с.

📄

Проверка за празен чип

✓

OK

🕒

0,269 с.

✖

Програмиране на чип с буфер

✓

OK

🕒

5,235 с.

📄

Проверка на чипа с буфер

⬆️

Проверка при номинално напр...

✓

OK

🕒

0,241 с.

📄

Настройване на защитни битове

✓

OK

🕒

0,272 с.

✓

Процесът е завършен успешно!

## Чип браузър

Чип браузърът може да бъде активиран от много места в програмата, например при избора на чип-приемник в режим „Програмиране на чип“.

Чип браузърът има различни функции, които ви помагат да откриете и изберете правилния чип от всички възможни.

Чиповете са разпределени в дървовидна структура, като е възможно да определяте дълбочината ѝ чрез следните опции:

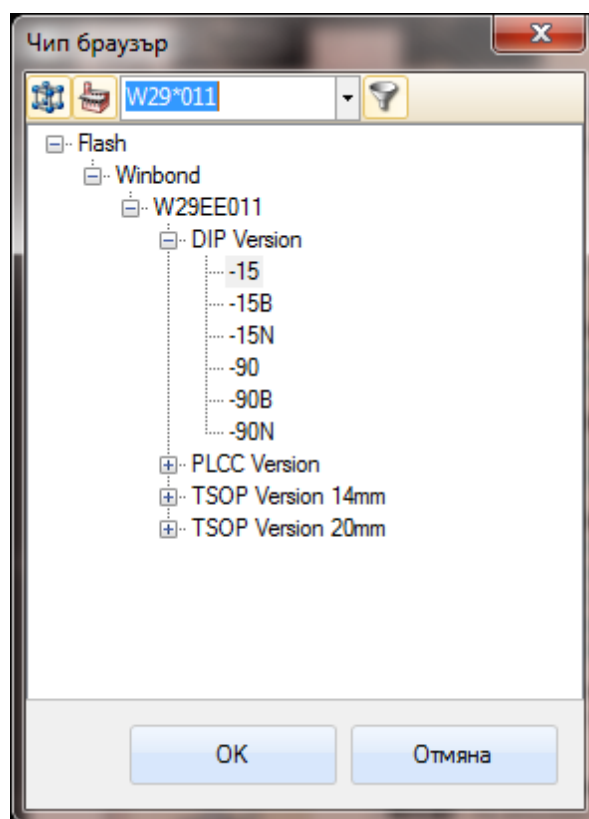


Групиране на чипове според технология.

Ако тази опция бъде активирана всички чипове се групират според технологията и е лесно да бъдат различени, например флаш чипове и EPROM.



Групиране на чипове според производител.



Ако е активирана тази опция всички чипове се групират в зависимост от производителя.

И двете опции могат да бъдат активирани едновременно. Тогава чиповете се групират първо според технология, а след това според производител.



### Опция за филтриране

В текстовото поле може да бъде въведен критерий за търсене. С активиране на опцията за филтриране ще бъдат показани само чиповете, съдържащи зададения критерий в името си.

Четири различни заместващи символа могат да бъдат използвани при филтрирането:

% Този символ може да представлява какъвто и да е вид и брой символи. Например филтриране „27%512“ ще върне следните резултати „27512“/„27c512“/„27SF512“.

\* Звездата притежава същата функционалност като %.

\_ Долната черта може да замества един символ. Например филтриране „27\_512“ връща следните резултати „27C512“ и „27E512“, но не и резултати като „27512“ или „27SF512“.

? Въпросителната притежава същата функционалност като долната черта.

## Автоматично идентифициране на чип

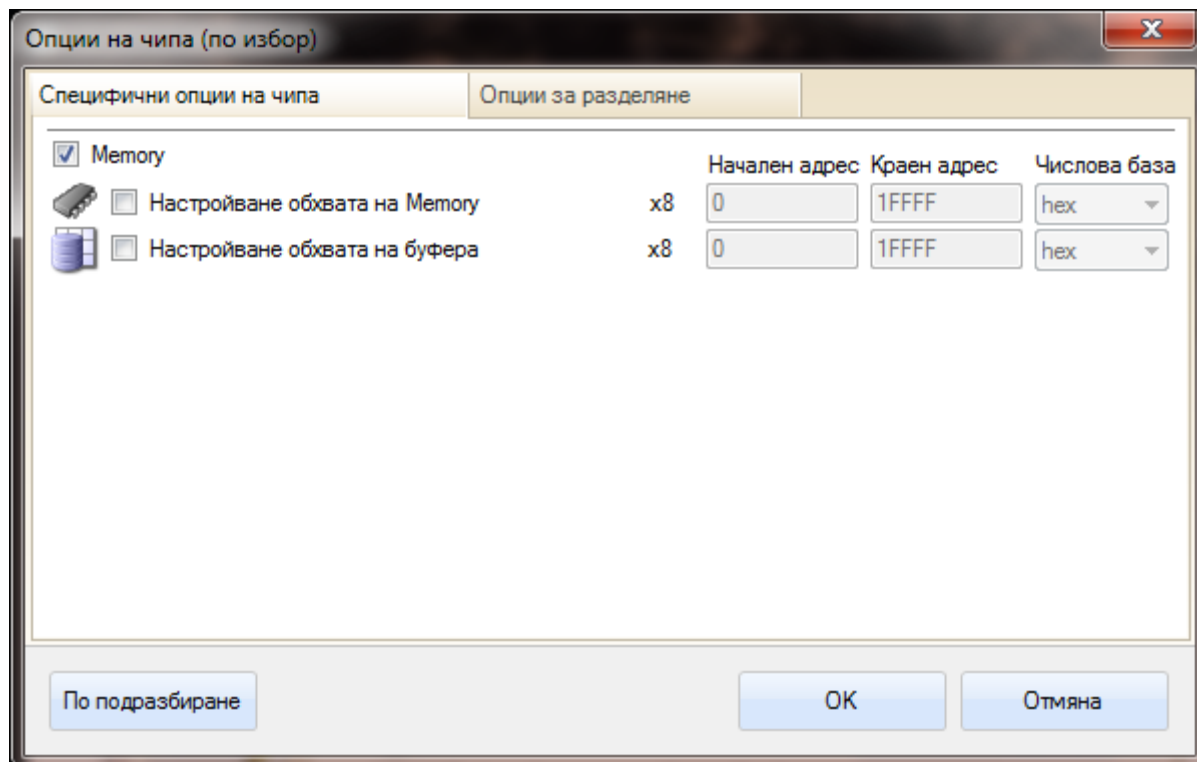
Повечето чипове съдържат сигнатура, която може да бъде прочетена чрез специални функции. Сигнатурата обикновено се състои от идентификатори за производител и чип, с което вида на чипа може да бъде определен точно. Има и чипове, които не съдържат сигнатура, и следователно не могат да бъдат автоматично засечени от софтуера.

Моля, забележете, че много чипове изискват сравнително високо напрежение (12.5V) на адресен ред A9 за прочитане на сигнатурата. Това напрежение е достатъчно, за да повреди други чипове, които не издържат такова напрежение на съответните крачета.

Prog-Express използва функцията за автоматично идентифициране автоматично в зависимост от настройките на процеса.

## Опции на чипа

### Специфични опции на чипа



### Памет

Някои чипове имат повече от една област памет. Например някои микроконтролери имат отделна програмна памет, памет за данни и конфигурационна памет. За всеки тип памет ще откриете отделни контроли за настройка.

### Настройване обхвата на паметта

По подразбиране ще бъде използван целият обхват на паметта. Ако искате да определите частичен обхват, моля, изберете „Настройване обхвата на паметта“. Обхватът може да бъде зададен от „Начален адрес“ и „Краен адрес“. Само адресите от паметта в този обхват ще бъдат програмирани, прочетени и проверени.

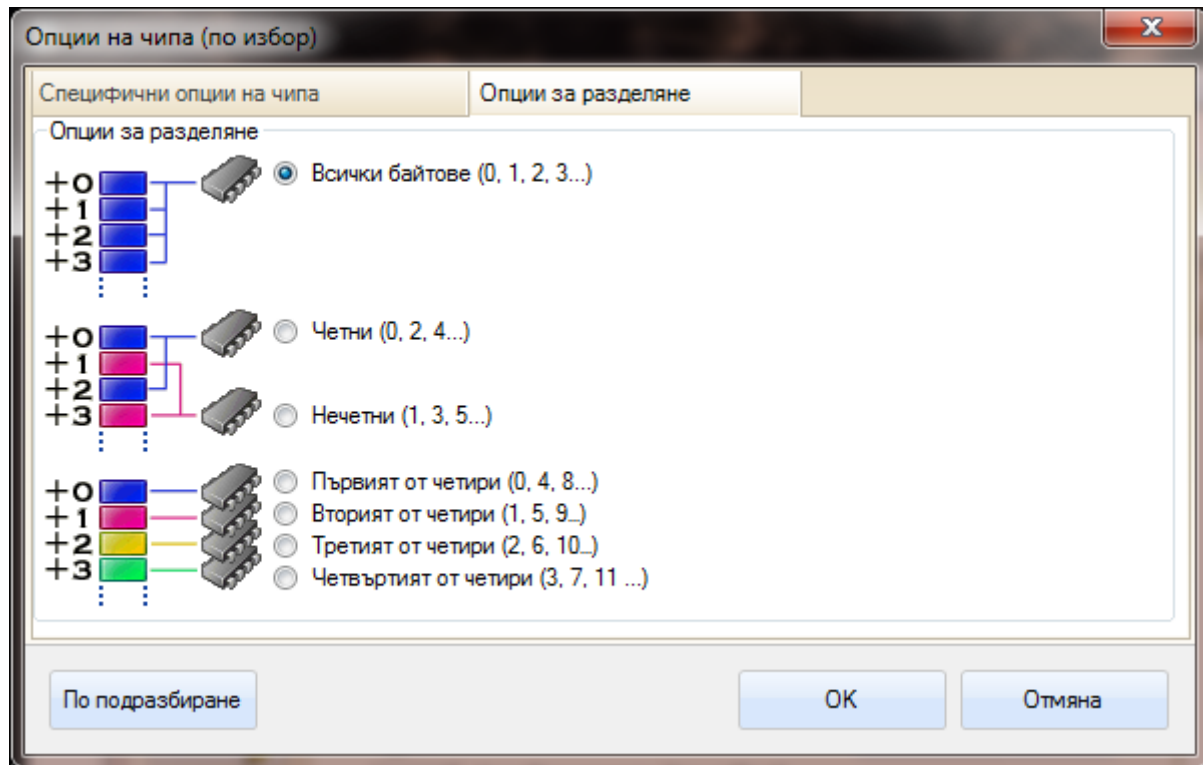
### Настройване обхвата на буфера

Обикновено данните се използват от първия адрес на файла (буфера). Ако е необходимо, можете да зададете „Начален адрес“ като офсет тук. Информацията за офсета се основава на широчината на данните на чипа. За 16-битов чип (2 байта данни на адрес) първите 4 байта ще бъдат пропуснати, ако използвате офсет 2.

### Специални опции на чип

Ако чипът притежава специални опции като сериен адрес EEPROM или настройка за конфигурация, те също ще бъдат показани тук.

## Опции за разделяне



Възможни са три различни функции за разделяне:

1. Без разделяне: Всички данни се програмират в чипа по обичайния начин.
2. Разделяне според четен/нечетен адрес: При избиране на настройката „Четни“ всички буферни данни с четни адреси се програмират върху чипа, а при избиране на „Нечетни“ се програмират всички данни с нечетни адреси.
3. Разделяне в четири адресни секции: От тук можете да изберете коя позиция от всеки четворен блок да бъде програмирана. Всеки първи, втори, трети или четвърти адрес от четири последователни адреса може да бъде избран.

## Серийни номера

За производството на чип често е необходимо определянето на сериен номер или адрес. За тази цел са на разположение следните настройки.

Серийни номера

☐ Няма да бъдат въвеждани серийни номера

☐ Зареждане на серийни номера от файл

Настояща позиция  
DEC 1

☒ Използване на генератор за серийни номера

Адрес за първия байт от серийния  
HEX 0

Брой байтове на серийния номер  
DEC 8

Следващ сериен номер  
DEC 1

Увеличаване на всяка стъпка с :  
HEX 1

Байтове  
Старшият байт - пръв (H-L)

Числова база  
Десетични

Числов формат  
Текст

Шестнадесетичен код  
30 30 30 30 30 30 30 31

ASCII  
00000001

OK Отхвърляне

## Файл за серийни номера

С избора на „Зареждане на серийни номера от файл“ може да бъде определен файл за серийни номера, от който желаните серийни номера да бъдат прочетени. След всеки успешен процес на записване следващият ред от файла се прочита и използва за следващия процес на записване.

Файлът за серийни номера трябва да има следния формат:

Редове, започващи с „#“ са коментари и се пропускат. Не се допускат коментари в редовете със серийни номера.

Между отделните стойности са позволени разстояния.

Редовете със серийни номера трябва да бъдат форматираны както следва: Редът трябва да започва с шестнайсетичен адрес, следван от двоеточие и списък от разделени със запетаи шестнайсетични стойности.

### Пример:

Шестнайсетичните стойности могат да бъдат написани в различни формати.

```
#Тест....
#Този коментар е позволен
1A0h:1, 2, 3, 4, 5, 6, 7, 8, 9
1A0h:11,12,13,14,15,16,17,18,19 #Този коментар не е позволен
1A0h:21h, 0x22, 23, &H24, 25, 26, 27, 28, 29
1A0h:*Ред с грешки*...
```

4E,10,F2 : обикновен шестнайсетичен без допълнения

4Eh,10h,F2h : обикновен шестнайсетичен с „h“ като наставка

0x4E,0x10,0xF2 : шестнайсетичен с „0x“ като представка

&h4E, &h10, &hF2 : шестнайсетичен с „&h“ като представка

Възможно е да се въвеждат и ASCII данни, но те трябва да се превърнат в шестнайсетичните им еквиваленти. Например ако е включен списък от MAC адреси (хардуерни адреси на компютърни мрежови карти) те ще изглеждат по следния начин:

Адресът е: 7F-3C-01-47-FF-04 и трябва да бъде записан в чипа на 05A2h:

```
7 F - 3 C - 0 1 - 4 7 - F F - 0 4
05A2h: 37,46,2D,33,43,2D,30,31,2D,34,37,2D,46,46,2D,30,34
```

Всеки сериен номер се записва в чипа от посочения адрес с дължината на стойността му.

## Генератор за серийни номера

Генераторът произвежда серийни номера според зададените настройки. Възможни са следните настройки:

Адрес за първия байт от серийния номер: Първият байт от серийния номер се записва на този адрес. Следващите байтове от серийния номер се записват на следващите позиции от адреса.

Брой байтове на серийния номер: Дължината на серийния номер в байтове. Настройката 8 дава като резултат 8-цифрени серийни номера.

Следващ сериен номер: Сериеният номер, който ще бъде използван за следващия производствен цикъл. След всеки успешен производствен цикъл серийният номер се увеличава със стойността, посочена в „Увеличаване на всяка стъпка с“.

Увеличаване на всяка стъпка с: След всеки успешен производствен цикъл серийният номер се увеличава с тази стойност.

Байтове: Тук определяте дали на първата позиция да бъде записан младшият или старшият байт от серийния номер.

Числова база: Генерираният сериен номер може да бъде в десетична или шестнайсетична бройна система.

Числов формат: Генерираният сериен номер може да бъде генериран като ASCII-текст или като двоични стойности.

Шестнадесетичен код/ ASCII: Тези две полета показват следващия сериен номер, който ще бъде генериран според зададените настройки.

## Проекти

Проектите позволяват всички настройки, свързани с режимите на програмиране, копиране, четене и серийно производство да бъдат запазени и задавани отново.

Изборът на чип, изборът на файл, опциите на чипа, настройките на серийните номера, броят на копията и активираните/деактивирани стъпки на процесите се запазват за всеки софтуерен режим. Освен това, чипът, файлът и опциите на чипа, характерни за всяко свързано програмиращо устройство, също се запазват за производствения режим.

Проектните файлове използват разширението „per“. То е съкращение от „Prog-Express Project“. Файловете се записват като стандартни текстови файлове, които могат да бъдат отворени и с обикновен текстов редактор. За целите на автоматизацията и дистанционното управление проектните файлове могат да бъдат създавани в други приложения и зареждани в Prog-Express.

## Запазване



Бутонът за запазване отваря файлов браузър. В него могат да бъдат въведени и запазени местонахождението и наименованието на проектния файл. Ако файлът съществува, той може да бъде презаписан или процесът да бъде отхвърлен.

Всички настройки от всички диалози и софтуерни режими се запаметяват.

## Зареждане



При зареждане всички запаметени настройки се възстановяват след избиране на желанния проект от файловия браузър.

Внимание! Всички предходни настройки се изтриват при зареждане на проект, запазват се само данните в буфера.

## Още

В горния десен ъгъл на софтуера се намират още няколко допълнителни функции: Те са „Оперативен режим“, „Обновяване на фърмуера“, „Калибриране“, „Информация за Prog-Express“ и „Помощник на Prog-Express“.



### Оперативен режим



Тук можете да запазите настройки срещу погрешно използване или пренастройване.

### Обновяване на фърмуера



С тази функция текущата версия фърмуер може да бъде прехвърлена на програматора.

### Калибриране



С помощта на инструмента за калибриране и тестване програматорите могат да бъдат проверени, а регулирането на напрежението им – калибрирано. Тази функция не се поддържа от всички програмиращи устройства.

### Информация за Prog-Express



Показва информационен прозорец за софтуера.

### Помощник на Prog-Express



Отваря ръководството за софтуера.

## Дистанционно управление на Prog-Express

За специални приложения можете да управлявате Prog-Express дистанционно от външна програма. Това може да се използва например за изпълнението на напълно автоматизирано програмиране на характерни за устройството настройки или измервателни стойности.

Команди могат да бъдат изпращани до Prog-Express, когато софтуерът е зареден чрез контролен файл или чрез параметри, зададени в команден интерпретатор.

### Управление чрез команден интерпретатор

Една или повече команди могат да бъдат подадени чрез команден интерпретатор. Всяка команда започва с наклонена черта. Някои команди изискват стойности на параметри като името на файла, който да бъде зареден. Параметрите се отделят с разстояния, а пътя до файла се поставя в кавички.

Примери:

```
Prog-Express.exe /open „C:\Directory\FileName.bin”
```

```
Prog-Express.exe /mode program /run 2 /exit
```

Prog-Express е приложение, до което можете да изпращате нови команди чрез командния ред, докато софтуерът работи, без да е необходимо всеки път да го стартирате отново.

### Дистанционно управление чрез скриптові файлове

Ако Prog-Express е стартиран чрез команден ред `“/remotefile filename”`, тогава всички команди във файла ще бъдат изпълнени. С използването на допълнителната команда `„roll”` файлът може непрекъснато да бъде проверяван за промени. Веднага след като файлът е променен, всички команди ще бъдат изпълнени.

Скриптовите файлове също могат да съдържат коментари, които се маркират с точка и запетая.

Пример:

```
; примерен скрипт  
open "C:\Directory\FileName.bin"  
mode program  
run 2  
exit
```

## Команда Remotefile

Командата „remotefile“ може да бъде използвана за четене и изпълнение на команди от файл.

Пример:

- `remotefile „C:\Directory\Remote.txt“` Зарежда командите от посочения файл и ги изпълнява.

## Команди Poll On и Poll Off

След изпълнение на командата „poll on“ файлът, посочен с командата „remotefile“, е непрекъснато проверяван за промени. При промяна на файла от външна програма командите от файла се изпълняват от Prog-Express.

Наблюдението се изключва чрез командата „poll off“. Командата „poll on“ има допълнителен параметър, чрез който може да бъде определен интервалът за проверка за промени на файла в милисекунди.

Пример:

- `Poll on` Активира наблюдение на контролния файл.
- `Poll off` Изключва наблюдението на контролния файл.
- `Poll on 2000` Активира наблюдение на контролния файл. Файлът се проверява за промени на всеки 2 секунди (2000 милисекунди).

## Команда Open

С командата „open“ можете да отворите проектен файл (.per) или файл с данни (.bin, .hex, .mhx,...). Всички файлове, различни от .per се зареждат в Hex-редактора. Всички .per файлове се разпознават като проектни и се отварят като такива.

Примери:

- `open „C:\Directory\FileName.bin“` Зарежда посочения файл в Hex-редактора.
- `open „C:\Directory\ProjectSettings.per“` Зарежда файла с запааметените настройки на Prog-Express.

## Команда Mode

Режимът на софтуера може да бъде зададен чрез команда „mode“. Валидни параметри са „program“ (програмиране), „copy“ (копиране), „read“ (четене), „production“ (производство), „hexeditor“ (hex-редактор) и „options“ (опции). Например „Prog-Express.exe /mode program“ настройва софтуера в режим на програмиране.

Примери:

- `mode program` Настройва софтуера в режим на програмиране.

- `mode production` Настройва софтуера в режим на серийно производство.

## Команда `Selectfile`

Командата „selectfile“ определя файла-источник на данни за режимите програмиране и производство и файла с данни за режим четене. Пример:

- `selectfile "C:\Directory\FileName.bin"` Задава посочения файл като източник на данни/целеви файл за избрания режим програмиране, четене или производство.

## Команда `Processstep`

Отделна стъпка от процес може да бъде активирана или деактивирана от командата „processstep“, номера на стъпката и параметъра „on“ или „off“.

Примери:

- `processstep 3 on` Активира стъпка номер 3 от текущия софтуерен режим.
- `processstep 5 off` Деактивира стъпка номер 5 от текущия софтуерен режим.

## Команда `Selectchipvariant`

С командата „selectchipvariant“ може да бъде зададен чипа за текущия режим.

Като параметър трябва да бъде определен уникалният идентификатор за вариант на чипа. Можете да откриете правилния идентификатор за вариант на чипа в списъка чипове, който може да бъде създаден с команда `savechiplist` (вижте по-долу).

Пример:

- `selectchipvariant 697` Задава чип с идентификатор за вариант на чипа 697 в текущия режим.

## Команда `Autoidentify`

Чипът може да бъде автоматично идентифициран чрез командата „autoidentify“.

Като допълнителен параметър може да бъде зададено име на файл. В този случай Prog-Express запазва идентификатора за вариант на чипа и името на чипа в посочения файл.

Примери:

- `autoidentify` Изпълнява автоматично идентифициране на чипа в активния режим на софтуера.
- `autoidentify "C:\Directory\chip.txt"` Запазва идентификатора за вариант на чипа и името му в посочения файл.

## Команда Run

Процесът на активния софтуерен режим може да бъде стартиран чрез командата „run“. Броя на циклите на процеса може да бъде определен с допълнителен параметър по избор.

Ако след командата „run“ е изпратена друга команда, тя се изпълнява едва след приключване на процеса. По този начин няколко процеса могат да бъдат изпълнявани един след друг.

Примери:

- `run` Стартира процес.
- `run 10` Стартира процес за 10 копия.

## Команда Savelog

Съдържанието от лога на действия може да бъде запазено във файл чрез командата „savelog“.

Като допълнителен параметър може да бъде зададено името на файла. Ако то не е определено, Prog-Express запазва данните от лога на действията в „log.txt“ на десктопа.

Примери:

- `savelog` Запазва данните от лога във файл log.txt.
- `savelog "C:\Directory\mylog.txt"` Запазва данните от лога в посочения файл.

## Команда Savedeviceinfo

Информацията за всички свързани Batronix USB устройства могат да бъдат запазени във файл чрез командата „savedeviceinfo“.

Командата запазва номера на устройството, наименованието, серийния номер, фърмуер версията, текущата задача, процентите на напредък и друга информация за свързаните устройства.

Името на файла може да бъде определено като допълнителен параметър. Ако то не е посочено, Prog-Express запазва данните за устройствата в „deviceinfo.txt“ на десктопа.

Примери:

- `savedeviceinfo` Запазва данните за устройствата в deviceinfo.txt.
- `savedeviceinfo "C:\Directory\devices.txt"` Запазва данните за устройствата в посочения файл.

## Команда Clearlog

С команда „clearlog“ можете да изтриете съдържанието на лога на действията.

## Команда Savechiplist

Командата „savechiplist“ създава текстови файл с всички поддържани варианти на чипове от програматора. Като първи параметър трябва да бъде определено съкратеното наименование на програматора (BX32, BX32P, BX40 or BX48). Като втори параметър се посочва хардуер версията на програматора (1, 2).

Като трети параметър може да се посочи име на файл. Ако то не е зададено, Prog-Express запазва списъка с чипове във файл „chiplist.csv“ на десктопа.

Примери:

- `savechiplist BX48 2` Запазва списък с чипове за „BX48 Batego II“ като chiplist.csv на десктопа.
- `savechiplist BX48 2 „C:\Directory\chip.csv“` Запазва списък с чипове под посоченото име и директория.

## Допълнителни команди

- `hide` Скрива Prog-Express.
- `show` Показва Prog-Express след изпълнена команда „hide“.
- `exit` Затваря Prog-Express.

## Примерни приложения: Програмиране на специфични данни

Казус: В производствена среда специален софтуер улавя измервателни данни от определени устройства, а отговарящите стойности за настройка трябва да бъдат програмирани върху чип.

Първоначално всички желани специални настройки като опции на чипа и серийни номера, както и чипът и файлът, които ще бъдат програмирани, се задават при нормални условия от Prog-Express и запазват в проектен файл project.per.

След това специалният софтуер стартира Prog-Express с извикване от командния ред:

```
Prog-Express.exe /remotefile "C:\Directory\Remote.txt" /poll on
```

Това стартира Prog-Express, който продължава да следи файла remote.txt за промени. Специалният софтуер първо тества устройството, след това запазва данните за настройка във файл adjustments.bin, а накрая запазва remote.txt със следното съдържание:

;Примерно приложение	
open "C:\Directory\Project.per"	;Зарежда програмните настройки
mode program	;Превключва в режим програмиране
run	;Стартира процеса
savelog	;Запазва данните от лога на действията в log.txt

След това специалният софтуер следи файла log.txt, оценява го и продължава със следващото устройство.