

**PICPROG**  
**Руководство пользователя**  
**Версия 1.06**

# Руководство пользователя PICPROG

---

## История модернизации

ВЕРСИЯ	ДАТА ВЫПУСКА	КОММЕНТАРИИ
1.00	17 Февраля 1999	DOS- V5.04, Win32- Beta 2, Прошивка- V3.1
1.01	11 Мая 1999	DOS- V5.05, Win32- Beta 3, Прошивка- V3.2
1.02	13 Июня 1999	DOS- V5.05m, Win32- Beta 3m, Прошивка- V3.3
1.03	3 Окт 1999	DOS- V5.05r, Win32- Beta 3r, Прошивка- V3.3
1.04	20 Дек 1999	DOS- V5.06, Win32- Beta 4, Прошивка- V3.4
1.04a	23 Янв 2000	DOS- V5.06a, Win32- Beta 4a, Прошивка- V3.4
1.05	31 Июля 2000	DOS- V5.07, Win32- Beta 5, Прошивка- V3.5
1.06	31 Мар 2001	DOS- V5.08, Win32- Beta 6, Прошивка- V3.6

©1999 Телесистемы. Все права защищены. Телесистемы не предоставляют никаких гарантий и не несут никакой ответственности за точность данной документации, а также за возможность использования изделия для какого-либо конкретного приложения. Телесистемы не передают никаких прав ни на свою интеллектуальную собственность, ни на собственность других лиц и организаций. Информация в данном документе может быть изменена без предварительного уведомления.

Все торговые марки, упомянутые в данном документе, являются собственностью их владельцев.

Телесистемы, а/я 232, Зеленоград, Москва, 103575, Россия

Тел/факс: (095) 530-1001, 531-4840

Веб-сайт: <http://www.telesys.ru>

# Руководство пользователя PICPROG

## Содержание

<b>ИСТОРИЯ МОДЕРНИЗАЦИИ.....</b>	<b>1</b>
<b>СОДЕРЖАНИЕ.....</b>	<b>2</b>
<b>ГЛАВА 1. ИНФОРМАЦИЯ О PICPROG.....</b>	<b>4</b>
Что такое PicPROG.....	4
СПИСОК ПОДДЕРЖИВАЕМЫХ МИКРОСХЕМ.....	5
ТРЕБОВАНИЯ К КОМПЬЮТЕРУ.....	8
<b>ГЛАВА 2. УСТАНОВКА PICPROG.....</b>	<b>9</b>
ПОДКЛЮЧЕНИЕ АППАРАТУРЫ PICPROG (LPT-ПОРТ).....	9
ПОДКЛЮЧЕНИЕ АППАРАТУРЫ PICPROG (COM-ПОРТ).....	9
ГДЕ ИСКАТЬ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ PICPROG.....	9
УСТАНОВКА PICPROG ДЛЯ WINDOWS.....	9
УДАЛЕНИЕ PICPROG ДЛЯ WINDOWS.....	10
УСТАНОВКА PICPROG ДЛЯ DOS.....	10
ПРОВЕРКА АППАРАТУРЫ PICPROG.....	10
<b>ГЛАВА 3. РАБОТА С PICPROG ДЛЯ WINDOWS.....</b>	<b>11</b>
Начальные сведения.....	11
Загрузка файлов.....	12
Слежение за файлами.....	14
Интеграция в MPLAB.....	14
Выбор типа устройства.....	15
Выбор областей для программирования.....	16
Установка битов конфигурации, Идентификаторов и Калибровочных значений.....	16
Комментарии по поводу битов конфигурации.....	17
Выбор пределов напряжения.....	18
Выбор алгоритма программирования и режима Внутрисхемного Программирования (ISP).....	18
Редактирование области программ и EEPROM данных.....	20
Установка микросхемы.....	20
Проверка устройства на чистоту.....	21
Программирование устройства.....	22
Статистика программирования и LOG-файл.....	22
Верификация устройства.....	23
Создание отчета программирования/верификации.....	23
Чтение устройства.....	24
Сохранение файла.....	24
Стирание устройства на основе EEPROM.....	25
Стирание устройства на основе EPROM.....	25
Снятие защиты с PIC16C(F)8х.....	26
Снятие защиты с AT89Cх051/AT89C5х.....	27
<b>ГЛАВА 4. АВТОНОМНЫЙ РЕЖИМ.....</b>	<b>28</b>
Загрузка файла для автономного программирования.....	28
Автономное программирование.....	29
Чтение информации автономного программирования.....	29
Стирание загруженного файла.....	30
<b>ГЛАВА 5. КОНФИГУРАЦИЯ PICPROG ДЛЯ WINDOWS.....</b>	<b>31</b>
<b>ГЛАВА 6. РАБОТА С PICPROG ДЛЯ DOS.....</b>	<b>34</b>
Запуск PicPROG для DOS.....	34
Список команд.....	34
Список параметров.....	35

# Руководство пользователя PICPROG

---

УСТАНОВКА БИТОВ КОНФИГУРАЦИИ .....	36
<b>ГЛАВА 7. ОБНОВЛЕНИЕ PICPROG.....</b>	<b>40</b>
ПОЛИТИКА ОБНОВЛЕНИЙ И ПОДДЕРЖКИ PICPROG .....	40
НОМЕРА ВЕРСИЙ.....	40
ПРОВЕРКА ВЕРСИЙ НА СОВМЕСТИМОСТЬ.....	40
ОБНОВЛЕНИЕ ПРОШИВОК.....	41
ВНУТРИСХЕМНОЕ ОБНОВЛЕНИЕ ПРОШИВОК.....	41
<b>ПРИЛОЖЕНИЕ А. АДАПТЕРЫ УСТРОЙСТВ PICPROG .....</b>	<b>42</b>
<b>ПРИЛОЖЕНИЕ В. ВНУТРИСХЕМНОЕ ПРОГРАММИРОВАНИЕ С ПОМОЩЬЮ PICPROG.....</b>	<b>45</b>
<b>ПРИЛОЖЕНИЕ С. СХЕМА АДАПТЕРА ДЛЯ ПОДКЛЮЧЕНИЯ PICPROG К СОМ- ПОРТУ КОМПЬЮТЕРА .....</b>	<b>47</b>
<b>ПРИЛОЖЕНИЕ D. ИСТОРИЯ МОДЕРНИЗАЦИИ PICPROG ДЛЯ WINDOWS.....</b>	<b>48</b>
<b>ПРИЛОЖЕНИЕ Е. ИСТОРИЯ МОДЕРНИЗАЦИИ PICPROG ДЛЯ DOS .....</b>	<b>49</b>
<b>ПРИЛОЖЕНИЕ F. ИСТОРИЯ МОДЕРНИЗАЦИИ ПРОШИВОК PICPROG .....</b>	<b>55</b>
<b>ПРИЛОЖЕНИЕ G. ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ (FAQ) ПО ПРОГРАММАТОРУ PICPROG.....</b>	<b>58</b>

## Глава 1. Информация о PicProg

---

### Что такое PicProg

PicProg – это промышленный программатор микроконтроллеров и E(E)PROM, идеально подходящий для применения как в условиях разработки, так и в условиях мелко- и средне - серийного производства. К основным возможностям программатора можно отнести следующие:

- Поддержка более 700 типов устройств, включая Microchip PIC12/14/16/17/18, ATMEL AVR и 89Cх051/89C5х/89Sхххх, микроконтроллеры Scenix SX, Ангстрем An15Exx, Intel I87C5х, Philips P87C5х и Fairchild ACE, кодеры/декодеры KeeLoq, а также различные параллельные и последовательные микросхемы E(E)PROM и Флэш-памяти
- Постоянные обновления программного обеспечения для поддержки новых кристаллов и семейств
- Возможность внутрисхемного обновления прошивки программатора в случае установки дополнительного флэш-процессора
- Включает 40-контактную ZIF панельку + 7-контактный разъем для внутрисхемного программирования (ISP)
- Два перестраиваемых источника напряжения: Vdd (2.0В-6.3В) и Vpp (4.0В-21В) обеспечивают надежную работу кристаллов при предельных напряжениях
- Безопасность установки и удаления микросхем обеспечивается заземлением всех выводов колодки (включая Vdd и Vpp) в режиме ожидания
- Постоянное отображение состояния Vdd и Vpp с помощью светодиодов
- Встроенная система снятия защиты для PIC16C8х/16F8х, а также для ряда старых образцов AT89Cх051/89C5х
- Работа под управлением PC или в автономном режиме (встроенный буфер 14Кб)
- Подключение к IBM PC через параллельный порт (прозрачный для принтера режим)
- Возможно подключение к компьютеру через COM-порт (требуется дополнительный адаптер) (в настоящее время не работает под Windows NT/2000)
- Автономное программирование одной кнопкой с сохранением статистики (количество программируемых микросхем можно ограничивать)
- Качественный пластиковый корпус
- Напряжение питания: 220V
- Размеры: 110мм x 75мм x 35мм
- В комплект поставки входит кабель связи, источник питания (может поставляться без блока питания – обращайтесь к дистрибьюторам) и CD-ROM с программным обеспечением и документацией
- Гарантия - 1 год
- Бесплатная поддержка пользователя по E-mail
- Имеется программное обеспечение как для DOS, так и для Win32 (Windows 9х/NT/2000)
- Бесплатные обновления программного обеспечения и прошивок всегда доступны в Интернет
- Поддерживает входные файлы форматов Binary, Intel HEX, Motorola-S и Tektronix HEX
- Поддерживает расширения HEX файлов, генерируемые многими популярными пакетами (Microchip MPASM/MPLAB, Parallax PASM/PASMХ/SPASM, TechTools CVASM, ATMEL AVR Studio, Scenix SASM, Parallax SKKey и т.д.)
- Функции проверки на чистоту, чтения, программирования, верификации и стирания
- Программирует и читает биты конфигурации (fuses), слова ID и EEPROM данных
- Поддерживает стандартные алгоритмы программирования
- Наличие дополнительных алгоритмов программирования обеспечивает существенное увеличение ресурса кристаллов PIC-JW (могут настраиваться пользователем)

# Руководство пользователя PICPROG

- Возможность установки стираемой защиты для всех типов кристаллов PIC-JW (включая кристаллы с улучшенной системой защиты)
- Ведение подробной статистики по каждой сессии программирования (ведение Log файла и дополнительного файла отчета)
- 32-разрядное графическое программное обеспечение для Windows 95/98/NT/2000
- Интеграция в Проводник Windows (загрузка, программирование и верификация любых файлов)
- Слежение за изменениями входных файлов
- Интеграция в пакет Microchip MPLAB (включает ускоритель MPSIM)
- Доступ ко всем функциям, включая чтение, программирование, проверку на чистоту, стирание и верификацию посредством клавиатуры
- Встроенный редактор памяти программ и EEPROM данных (в форматах Hex, Decimal и Octal)
- Программное обеспечение для DOS, полностью конфигурируемое из командной строки (отлично работает в окне DOS под Windows 95/98)
- Пакетное программирование серии кристаллов (под DOS)

## Список поддерживаемых микросхем

При условии использования последних версий программного обеспечения и прошивки PicProg поддерживает следующие типы микросхем:

### Microchip Technology Inc.

#### **PicMicro(TM):**

- 16(L)C52/54/54A/54B/55/55A/56/56A/57/57C/58A/58B/505, 16LV54A/58A
- 16(L)C61/62/62A/62B/63/63A/64/64A/65/65A/65B/66/67, 16C432
- 16(L)C71/710/711/712/715/716/717
- 16(L)C72/72A/73/73A/73B/74/74A/74B/76/77/745/765/773/774
- 16(L)C923/924
- 16(L)C84, 16(L)F83/84/84A, 16(L)F870/871/872/873/874/876/877
- 16(L)F627/628, 16(L)F73/74/76/77
- 16(L)C620/620A/621/621A/622/622A/641/642/661/662, 16CE623/624/625
- 16(L)C554/554A/556/556A/558/558A
- 12(L)C508/508A/509/509A/671/672, 12CE518/519/673/674
- 17(L)C42/42A/43/44, 17(L)C752/756/756A/762/766
- 18(L)C242/252/442/452
- 14000

#### **КeeLoq(R) кодеры/декодеры:**

- HCS101/200/201/300/301/320/360/361, HCS410/412, HCS512

#### **Последовательные I2C EEPROM:**

- 24C(LC,AA)00/01/21/21A/02/024/025/04/08/16/164/174/32/32A/64/65/128/256
- 24(L)CS21/21A/52, 85C72/82/92

#### **Последовательные Microwire(R) EEPROM:**

- 93C06, 93(L)C46/46A/46B/56/56A/56B/66/66A/66B/76/86
- 93(L)CS56/66

#### **Последовательные SPI EEPROM:**

- 25(L)C010/020/040/080/160/320/640

#### **Параллельные EEPROM:**

- 28C04A/16A/17A/64A, 28LV64A

### ATMEL Corp.

#### **AVR & MegaAVR:**

- AT90(L)S1200/2313/2323/2333/2343/4414/4433/4434/8515/8535, AT90C8534
- ATmega603/103/163/161(L)
- ATtiny10/11/12/15/22/28(L,V)

#### **MCS51 MCU:**

- AT89C1051/1051U/2051/4051

# **Руководство пользователя PICPROG**

---

- AT89C51/51-5/52/52-5/55/55-5, AT89LV51/52/55
- AT87F51/52/55/55WD/51RC, AT87LV51/52/55
- AT89(L)S8252/53

## **Последовательные I2C EEPROM:**

- AT24C01/01A/21/02/02A/04/04A/08/08A/16/16A/32/64/128/256/512
- AT34C02

## **EEPROM конфигурации FPGA:**

- AT17C(LV)65/65A/128/128A/256/256A/512/512A/010/010A/020/020A

## **Последовательные Microwire(R) EEPROM:**

- AT93C46/46A/46B/46C/56/57/66

## **Последовательные SPI EEPROM:**

- AT25010/020/040/080/160/320/640/128/256, AT25HP256/512

## **Параллельные EEPROM:**

- AT28C16/16E/17/17E/64/64B/64E/64X/256/010/040, AT28BV16/64
- AT28HC64B/256, AT28LV64B/256/010

## **Параллельные Flash:**

- AT29C256/512/010A/020/040/040A
- AT49F512/020/040/040T
- AT49F010/HF010
- AT49F001(T/N/NT)/002(T/N/NT), AT49LV002(N)T, AT49BV002(N)T

## **Ангстрем (Россия)**

### **Микроконтроллеры Тесей:**

- An15E03/KP1878BE1

## **Fairchild Semiconductor**

### **Микроконтроллеры ACE:**

- ACE1101(B,L)/1202(B,L)

## **INTEL Corp.**

### **Микроконтроллеры MCS51:**

- 87C51/52/54/58, 87C51RA/RB/RC/FA/FB/FC

### **Параллельные Flash – 12В:**

- 28F256A/512/010/020/001BXT/001BXB

## **Philips Semiconductors**

### **Микроконтроллеры MCS51:**

- P87C51/52/54/58, P87C51RA+/RB+/RC+/FA/FB/FC

## **SCENIX Semiconductor Inc.**

### **Микроконтроллеры SX:**

- SX18AC/20AC/28AC (старые и новые версии)
- SX48BD/52BD (ES и окончательные версии)

## **AMD**

### **Параллельные Flash – 12V:**

- Am28F256/512/010/020, Am28F256A/512A/010A/020A

### **Параллельные Flash:**

- Am29F010/010A/002(T/B/NT/NB/BT/BB/NBT/NBB)/004BT/004BB/040B

## **Mosel Vitelec**

### **Параллельные Flash:**

- V29C51000/002/004(T,B), V29LC51000/001/002(T,B), V29C31004(T,B)

## **Ramtron International Co.**

### **Последовательные I2C FRAM:**

# Руководство пользователя PICPROG

---

- FM24C04/16/64/256, FMCL16/64
- Последовательные SPI FRAM:**
- FM25040/160/640, FM25C(L,CL)160
- Параллельные FRAM:**
- FM1608/1808, FM18L08

## Silicon Storage Technology Inc.

**Параллельные Flash:**

- SST29EE512/010/020
- SST29LE(VE)512/010/020
- SST39SF(LF,VF)512/010/020/040, SST39SF(LF,VF)020P/040P

## ST Microelectronics (SGS-Thomson)

**Последовательные I2C EEPROM:**

- ST24C(W,E)01/02/04/08/16/164(R), ST25C(W,E)01/02/04/08/16/164(R)
- M24C01/02/04/08/16/32/64/128/256(W,R,A), M34C02(W,R), M34S32(W)

**Последовательные Microwire(R) EEPROM:**

- ST93C06/46/47/56/57/66/67(A,C), ST93CS46/47/56/57
- M93C06/46/56/66/76/86(W,R), M93S46/56/66(W,R)

**Последовательные SPI EEPROM:**

- ST95010/020/040(W), ST95P02/04/08, M95080/160/320/640/128/256(W,R)

**Параллельные EEPROM:**

- M28C16/16A/17/17A/64/64A/64C/64X(W), M28256/010(W,R)
- M28LV64/64X

**Параллельные Flash – 12V:**

- M28F256/512/010

**Параллельные Flash:**

- M29F010B/002T/002B/002NT

## Winbond Electronics Corp.

**Параллельные Flash:**

- W29EE512/011, W29C010M/020/040

## Xicor

**Последовательные SPI EEPROM:**

- X25C02
- X25020/040/160/170/320/330/642/650/128/138/057/097 (5V,3V,2.7V,2.5V,1.8V)

## Другие производители

**Параллельные CMOS EPROM:**

- 27C16/32 (только чтение)
- 27C64/128/256/512/010/020/040, 27LV64/256

**Параллельные Flash – 12V (Macronix):**

- MX28F1000P/2000P

**Параллельные Flash (NexFlash):**

- NX29F010

**Последовательные NAND Flash (Toshiba):**

- TC58A040F

**Параллельные NAND Flash (Samsung):**

- KM29N040T

## Требования к компьютеру

Для работы программного обеспечения Win32 необходимо:

- IBM PC/AT совместимый компьютер 386 или старше
- Windows 95/98 или Windows NT4/2000 с VGA- или лучшим монитором и мышью
- 8 MB ОЗУ
- 2 MB на жестком диске
- Привод CD-ROM
- 1 LPT- или COM-порт

Для работы программного обеспечения DOS необходимо:

- IBM PC/AT совместимый компьютер 286 или старше
- MS-DOS 5.0 или старше
- Монохромный (или лучше) монитор
- 640 KB ОЗУ
- 1MB на жестком диске
- Привод CD-ROM
- 1 LPT- или COM-порт

## Глава 2. Установка PicProg

### Подключение аппаратуры PicProg (LPT-порт)

В стандартной конфигурации PicProg осуществляет связь с PC через LPT-порт. Связной кабель данных/питания входит в комплект поставки PicProg.

- Подключите разъем RJ11 кабеля к программатору, а разъем DB-25 к любому из параллельных портов вашего компьютера. Учтите, что PicProg использует режим, прозрачный для принтера, который позволяет одновременно использовать PicProg и принтер на одном LPT-порту. Поэтому, если на Вашем компьютере установлен только один параллельный порт и к нему подключен принтер, то сначала отключите кабель принтера от порта, затем подключите разъем PicProg к порту, после чего подключите кабель принтера к ответной части разъема PicProg.
- Воткните вилку источника питания PicProg в сеть 220В x 50-60Гц. Должен загореться зеленый светодиод **"POWER"** в левом верхнем углу программатора. Кроме того, должен на несколько секунд загореться, а затем погаснуть красный светодиод **"ERR/AUTO"**. После этого Ваш программатор готов к работе.

### Подключение аппаратуры PicProg (COM-порт)

PicProg также может подключаться к PC через стандартный COM-порт. Для этого необходим дополнительный адаптер (не входит в стандартную поставку программатора).

- Подключите разъем RJ11 связного кабеля к программатору, а к разъему DB-25 того же кабеля дополнительный адаптер COM-порта. После этого разъем DB-9 адаптера COM-порта подключите к любому из последовательных портов вашего компьютера.
- Воткните вилку источника питания PicProg в сеть 220В x 50-60Гц. Должен загореться зеленый светодиод **"POWER"** в левом верхнем углу программатора. Кроме того, должен на несколько секунд загореться, а затем погаснуть красный светодиод **"ERR/AUTO"**. После этого Ваш программатор готов к работе.

**Примечание:** Подключение PicProg через COM-порт при работе под Windows NT/2000 в данной версии программного обеспечения **не поддерживается**.

**Примечание:** Адаптер COM-порта для PicProg поставляется отдельно. Схему адаптера для самостоятельного изготовления см. в данном руководстве: [Приложение С. Схема адаптера для подключения PicProg к COM-порту компьютера](#).

### Где искать программное обеспечение PicProg

CD-ROM «Современные микроконтроллеры», входящий в комплект поставки PicProg, содержит все необходимые программы и документацию, необходимые для успешного функционирования PicProg. Программное обеспечение находится в каталоге \Telesys\PicProg-Russian.

Если со времени последнего выпуска CD-ROM произошли изменения в программном обеспечении или документации, то последние версии находятся на дополнительной дискете, включенной в комплект поставки.

Кроме того, последние версии программного обеспечения, прошивок и документации всегда можно бесплатно получить с Веб-сайта фирмы **Телесистемы**:

<http://www.telesys.ru/picprog.shtml>

### Установка PicProg для Windows

PicProg для Windows – 32-разрядное графическое программное обеспечение, предназначенное для использования в среде Windows 95/98 или Windows NT4/2000.

# Руководство пользователя PICPROG

---

**Примечание:** PicProg для Windows не работает под Windows 3.1 или Windows NT3.51.

Программа установки PicProg для Windows находится в подкаталоге \Windows на CD-ROM или дискете. Называется она **ppwXXX.exe**, где **XXX** означает номер версии. Запустите эту программу и следуйте инструкциям, которые она выдает на экран.

**Примечание:** Для установки и работы PicProg для Windows под Windows NT4 следует входить в систему с административными полномочиями. В противном случае программа установки сразу же закончит работу с предупреждающим сообщением.

## Удаление PicProg для Windows

Удалить PicProg для Windows можно, открыв стандартный диалог Windows **Панель управления>Установка и удаление...** . Выберите пункт **PicProg for Windows** и нажмите кнопку **Добавить/Удалить**.

**Примечание:** Для корректного удаления PicProg для Windows под Windows NT4 следует запускать процедуру удаления с административными привилегиями. В противном случае NT драйвер для PicProg не будет удален из системы.

## Установка PicProg для DOS

Программное обеспечение PicProg для DOS не требует специальной установки. Просто копируйте все файлы из подкаталога \Dos с CD-ROM или дискеты на жесткий диск.

## Проверка аппаратуры PicProg

Вы можете также скопировать программу тестирования аппаратуры PicProg с CD-ROM или дискеты из подкаталога \Test. Эта программа: **pictest3.exe** (или **pictest2.exe** для программаторов версии 2.x) предназначена в основном для предпродажной проверки и настройки программатора на фирме-изготовителе, но может быть также полезна для проверки или поиска неисправностей конечным пользователем. Программа PicTest работает под DOS (а также в окне DOS под Windows9x) и имеет простой и понятный интерфейс пользователя.

**Примечание:** Программа PicTest не работает под Windows NT.

## Глава 3. Работа с PicProg для Windows

### Начальные сведения

После того, как PicProg подключен к PC и включен в сеть, а программное обеспечение PicProg для Windows успешно установлено в соответствии с предыдущей главой, можно начинать работу с программой.

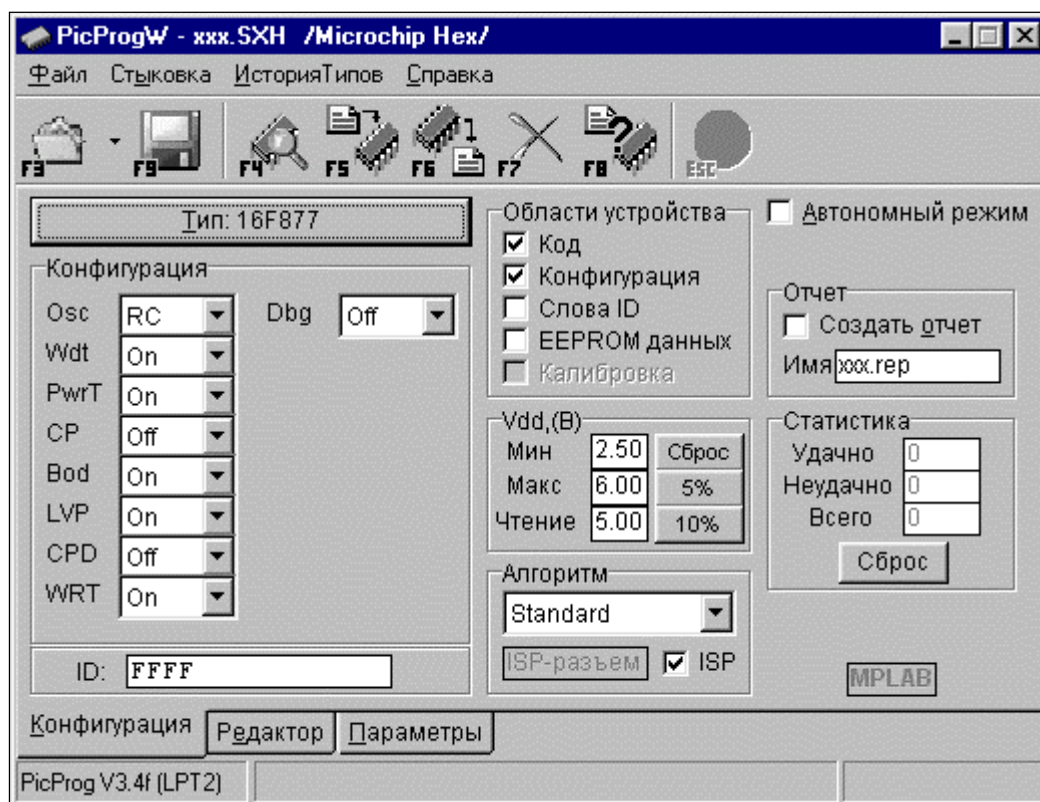
Программа установки PicProg создает пиктограмму с именем «**PicProg for Windows**» в меню **Пуск** Windows. Щелчок мышью на этой пиктограмме запускает программу, после чего на экране появляются два окошка:



Маленькую кнопку в верхнем правом углу экрана, обозначенную как **F12**, можно использовать для быстрого доступа к главному окну, быстрой загрузки файла из среды MPLAB (см. [Интеграция в MPLAB](#)) и в качестве панели для перетаскивания файлов из Проводника Windows (см. [Загрузка файлов](#)).

Эта кнопка всегда находится поверх всех остальных окон в системе. Ее можно передвинуть в любое место экрана мышью. Новое положение сохранится и при следующем запуске программы. Можно вообще удалить кнопку **F12** с экрана (см. [Глава 5. Конфигурация PicProg для Windows](#)).

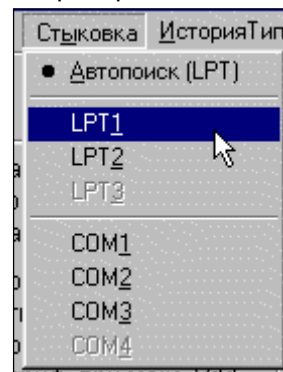
В центре экрана Вы увидите главное окно программы:



В нижней части окна находится строка состояния. На ней отображается текущее состояние выполняемых операций, а также состояние аппаратуры PicProg. При первом запуске программа автоматически ищет PicProg на всех доступных LPT-портах. Если Вы подключили PicProg к одному из LPT-портов компьютера, то должны увидеть номер LPT-порта и номера версий аппаратуры и прошивки PicProg в нижнем левом углу окна (например, **PicProg V3.4f (LPT2)**). Буква **'f'** после номера версии означает, что в Вашем программаторе установлен флэш-процессор, что позволяет осуществлять внутрисхемное обновление прошивки (см. [Внутрисхемное обновление прошивок](#)). Если же вместо этого Вы видите сообщение **Нет PicProg (Демо)**, значит, PicProg не был обнаружен ни на одном из LPT-портов Вашего компьютера (возможно,

# Руководство пользователя PICPROG

программатор подключен к одному из COM-портов). В этом случае проверьте надежность подключения кабелей и наличие питания PicProg (зеленый светодиод **“POWER”** должен гореть), после чего попробуйте повторить соединение с аппаратурой PicProg, выбрав один из пунктов меню **Стыковка**:



При работе под Windows 95/98 названия некоторых портов будут затенены. Это означает, что соответствующий порт не зарегистрирован в BIOS Вашего компьютера. Кроме того, если номера портов у Вас идут не по порядку (например, установлены только порты COM1 и COM3), BIOS перетранслирует номера портов (для вышеприведенного примера Вы увидите разрешенными порты COM1 и COM2). При работе под Windows NT/2000 не существует простого способа определения наличия портов. Поэтому Вы увидите, что все LPT-порты разрешены, а все COM-порты запрещены (поскольку текущая версия программы не поддерживает работу PicProg через COM-порты под Windows NT/2000).

При выборе пункта **Автопоиск** программа снова будет искать подключенный PicProg на доступных портах. Вы можете выбрать диапазон портов, используемых при автопоиске из: LPT, COM и All (LPT+COM) (см. [Глава 5. Конфигурация PicProg для Windows](#)). В случае, если PicProg снова не найден (и Вы уверены в отсутствии аппаратных проблем), можно явно указать номер порта, к которому подключен PicProg.

**Примечание:** Несмотря на то, что **Телесистемы** прилагают все усилия к обеспечению прозрачности работы PicProg для любых типов принтеров, возможны конфликты отдельных драйверов принтера с PicProg. В этом случае Вам, возможно, придется переключить PicProg на отдельный LPT-порт или запретить работу драйвера принтера во время работы с PicProg.

Если Вы не хотите видеть кнопку **F12** на своем экране, но хотите иметь быстрый доступ к главному окну программы, то можете воспользоваться **системной иконкой** (см. [Глава 5. Конфигурация PicProg для Windows](#)). При ее разрешении в правом нижнем углу экрана рядом с системными часами появляется значок. При однократном щелчке мышкой на этом значке главное окно программы раскрывается и помещается поверх остальных окон в системе. При двойном щелчке мышкой происходит загрузка файла из среды MPLAB (см. [Интеграция в MPLAB](#)).



**Примечание:** Если **системная иконка** разрешена, то при попытке минимизировать главное окно программы оно будет полностью удалено с экрана. Для его восстановления щелкните на **системной иконке** мышкой.

## Загрузка файлов

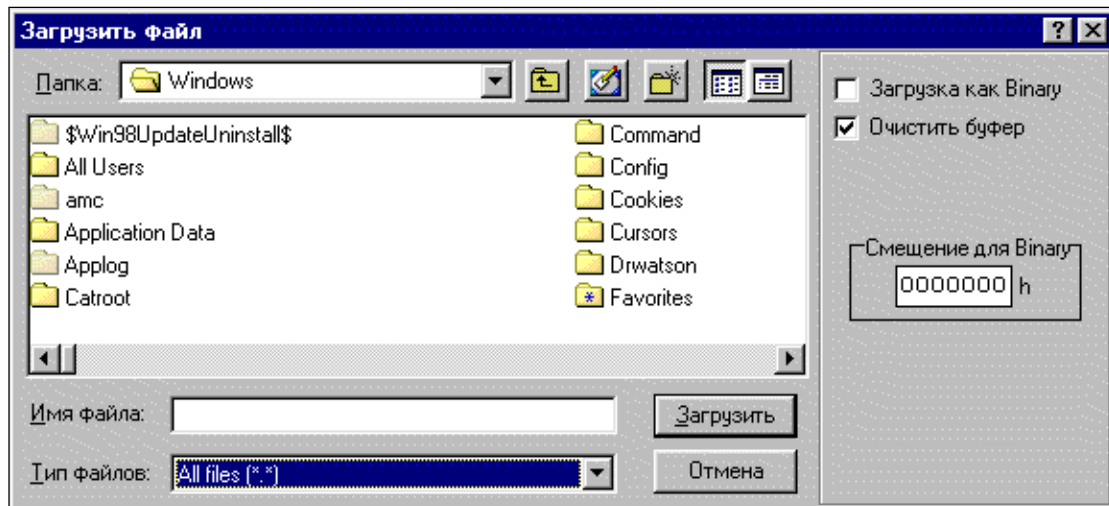
После установления соединения можно загружать файл с программой. Для этого существует несколько способов. Самый простой из них – щелкнуть мышью на кнопке



**Загрузить файл**, нажать клавишу **F3** или выбрать пункт меню **Файл>Открыть**. В результате на экране появится диалог **Открыть файл**, в котором можно выбрать имя файла для загрузки. При желании в диалоге загрузки можно выбрать один из нескольких фильтров файлов по расширениям, но фактически PicProg может загрузить любой файл, поскольку определяет формат входного файла по его содержанию, а **не** по расширению. Перед загрузкой файла в диалоге загрузки можно разрешить или запретить очистку буфера программатора, устанавливая состояние пункта **Очистить буфер**. Начальное состояние данного пункта при **каждом** вызове диалога определяется пунктом **Очищать при загрузке** в закладке Параметры (см. [Глава 5. Конфигурация PicProg для Windows](#)).

# Руководство пользователя PICPROG

При загрузке файла через диалог загрузки можно отменить автоопределение формата файла по его внутреннему содержанию, и загрузить любой файл как бинарный. Для этого нужно отметить пункт **Загрузка как Binary** диалога (данный пункт сбрасывается каждый раз при вызове диалога загрузки). Если загружается бинарный файл (или он принудительно объявлен бинарным), то можно определить его смещение в буфере программатора с помощью соответствующего пункта диалога. Величина смещения (т.е. начальный адрес загрузки файла в буфере программатора) может задаваться в форматах Hex, Decimal или Octal в зависимости от того, какой формат выбран по умолчанию для адресов в редакторе программы (см. [Редактирование области программ и EEPROM данных](#)). Формат смещения обозначается соответствующей буквой после величины смещения (например, **h** – Hex формат). Величина смещения сбрасывается в ноль при каждом вызове диалога загрузки.



PicProg может загружать файлы следующих форматов: **Intel HEX** (стандартный и расширенный), **Motorola S** и **Tektronix HEX**. Если оказывается, что файл не относится ни к одному из этих форматов, он загружается как **Двоичный**. Кроме того, программа распознает расширения HEX файлов, содержащие информацию о конфигурации, идентификаторе и данных EEPROM, а также типе устройства, генерируемые следующими системами разработки: Microchip MPASM/MPLAB, Parallax PASM/PASM/SPASM, TechTools CVASM, ATMEL AVR Studio, Scenix SASM и Parallax SXKey. Для HEX файлов информация о типе устройства может быть также автоматически загружена из соответствующего .cod файла (который генерируется рядом ассемблеров и компиляторов, включая Microchip MPASM и Hi-Tech PIC C).

Имеется несколько параметров конфигурации, определяющих способ установки или определения типа устройства для загружаемых файлов. Программа может автоматически извлекать такую информацию из самого файла, брать ранее установленный тип устройства или просить пользователя выбирать тип каждый раз при загрузке файла (см. [Глава 5. Конфигурация PicProg для Windows](#)).

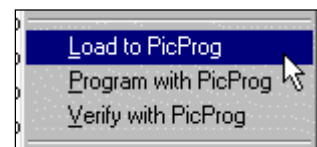
Файл можно загружать непосредственно из **Проводника Windows**. Во время установки PicProg для Windows добавляет в контекстное меню всех файлов несколько новых пунктов. Поэтому при щелчке правой кнопкой мышки на любом файле в появившемся меню можно увидеть следующие пункты:

При выборе команды **Load to PicProg** будет автоматически запущен PicProg для Windows (если он уже не запущен), а указанный файл в него загружен.

Команда **Program with PicProg** одновременно с загрузкой файла автоматически выполнит команду

**Программирование**, упрощая таким образом процесс программирования.

Команда **Verify with PicProg** одновременно с загрузкой файла автоматически выполнит команду **Верификация**.



# Руководство пользователя PICPROG

Если PicProg уже запущен, то в него можно загрузить любой файл из **Проводника Windows**. Для этого нужно просто перетащить пиктограмму файла из **Проводника** на кнопку **F12** программы. Файл будет загружен, а главное окно программы появится поверх всех остальных окон в системе. Файл в программу можно загрузить и из командной строки (либо командой **Выполнить** из меню **Windows Пуск**, либо из командной строки DOS в окне DOS). Нужно всего лишь набрать строку: PicProgW.exe <имя файла> и указанный файл будет загружен в буфер программы (программа будет запущена при необходимости).

**Примечание:** Независимо от способа загрузки файла одновременно в системе может быть запущена только одна копия программного обеспечения для PicProg. Поэтому попытка загрузить файл при уже запущенной программе не открывает новое окно, а записывает его **вместо** ранее открытого файла.

**Примечание:** Пока программатор выполняет любую операцию с микросхемой (программирование, проверка, верификация, стирание, чтение) все попытки загрузить новый файл игнорируются.

Файлы можно загружать непосредственно из системы Microchip MPLAB (см. [Интеграция в MPLAB](#)).

Независимо от способа загрузки файла его имя будет добавлено в **Список Ранее Открытых Файлов**, доступный либо в меню **Файл**, либо в списке рядом с кнопкой **Загрузить файл** (содержимое данного списка сохраняется от сессии к сессии).

Одновременно с именем файла в этом списке сохраняется и тип устройства, для которого был открыт данный файл, что облегчает последующее его открытие. Можно настроить программу так, что при ее запуске последний открытый файл будет автоматически загружаться (см. [Глава 5. Конфигурация PicProg для Windows](#)).

Если в буфер программы загружен файл, его имя и формат отображаются в титульной строке главного окна программы.

**Примечание:** При любом способе загрузки файла, кроме диалога загрузки, необходимость стирания буфера программатора перед загрузкой определяется пунктом **Очищать при загрузке закладки Параметры**. Возможность принудительной загрузки бинарного файла и его смещение в буфере недоступны.

## Слежение за файлами

После загрузки файла в буфер программы его содержимое постоянно проверяется на изменения. При обнаружении таких изменений программа может сообщить об этом пользователю либо звуковым сигналом, либо перезагрузкой файла (автоматически либо после получения подтверждения от пользователя) (см. [Глава 5. Конфигурация PicProg для Windows](#)). Если выбран режим автоматической перезагрузки, то после нее главное окно программы появляется перед всеми остальными окнами системы, значительно упрощая таким образом процесс программирования во время отладки устройств (когда требуется перепрограммировать кристаллы много раз в течение одного сеанса). При включенном слежении за файлами после каждой компиляции проекта требуется нажимать лишь одну кнопку **Программирование**.

Следует иметь в виду, что PicProg отслеживает содержимое оттранслированного файла. Поэтому результат не зависит от того, какими средствами программирования Вы пользуетесь. Возможно применение любых компиляторов как для Windows, так и для DOS (в окне DOS).

## Интеграция в MPLAB

PicProg для Windows обладает рядом дополнительных свойств, позволяющих интегрировать его в популярную среду разработки Microchip's MPLAB. Хотя выбранный вариант интеграции и не является столь же тесным, как для собственных

# Руководство пользователя PICPROG

программаторов Microchip: PROMATE и PICSTART+ , он весьма прост и к тому же позволяет выполнять некоторые функции, недоступные в исходном пакете.

Если пакет MPLAB запущен и в него загружен какой-либо проект, в нижнем левом углу главного окна PicProg появляется значок **MPLAB**. В этом случае при щелчке на кнопке **F12** или нажатии на клавишу **F12** (только если она разрешена – см. [Глава 5. Конфигурация PicProg для Windows](#)) конечный HEX файл проекта MPLAB будет загружен в буфер программы, а ее главное окно появится поверх остальных окон системы. Если вместо кнопки **F12** используется **системная иконка**, то двойной щелчок мышкой на ней приведет к аналогичному результату.

**Примечание:** Создавая новый проект в MPLAB, следует сохранить его из меню **MPLAB Project>SaveProject** и оттранслировать проект хотя бы раз, прежде чем пытаться загрузить его в буфер PicProg. В противном случае Вы увидите сообщение: **Файл не найден**.

Следует помнить, что, начиная с версии MPLAB 3.40, проект может создаваться из нескольких исходных файлов. Нажатие **F12** в PicProg загружает конечный HEX файл, имя которого не обязательно совпадает с именами исходных файлов. Если Вы хотите загрузить HEX файл, являющийся результатом компиляции любого из открытых в MPLAB IDE файлов, выберите окно с этим файлом и щелкните на кнопке PicProg **F12**, одновременно удерживая нажатой клавишу **Alt** (или нажмите клавиши **Alt+F12**, если они разрешены).

**Примечание:** Если MPLAB не запущен, нажатие клавиш **F12/Alt+F12** или кнопки **F12** вызывает появление окна PicProg поверх остальных окон в системе.

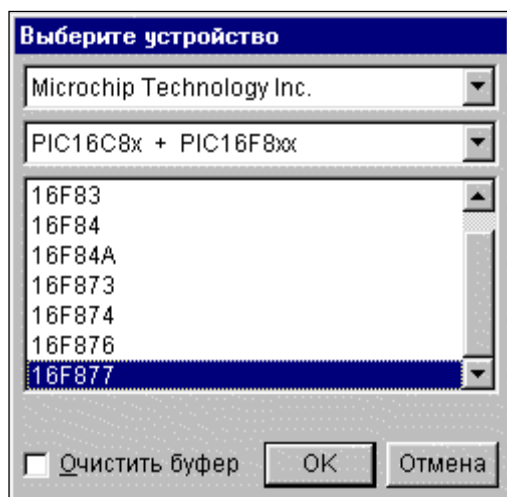
**Примечание:** Если MPLAB запущен, а Вы хотите просто увидеть окно PicProg, не перегружая файл, щелкните правой кнопкой мыши на кнопке **F12** и выберите пункт **Показать**.

PicProg для Windows имеет в своем составе также программный ускоритель симулятора MPLAB – MPSIM. Если ускорение разрешено, режим MPSIM **Run** можно ускорить примерно в 10 раз (см. [Глава 5. Конфигурация PicProg для Windows](#)).

## Выбор типа устройства

Чтобы изменить тип устройства после того, как файл загружен, щелкните мышью на кнопке **Тип** или нажмите клавиши **Alt+T**:

Тип: 16C73A



Откроется диалог **Выберите устройство**, перечисляющий все доступные типы устройств для выбора.

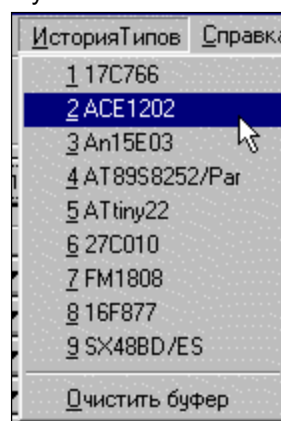
Выбрав пункт **Очистить буфер** в этом диалоге, можно полностью стереть содержимое программного буфера после нажатия кнопки **OK**.

Следует помнить, что изменение семейства микросхем может привести к изменениям в битах конфигурации и идентификации, поскольку адреса, по которым располагаются данные области, различны для разных семейств. Следует поэтому сначала выбирать тип устройства, а потом устанавливать биты конфигурации.

# Руководство пользователя PICPROG

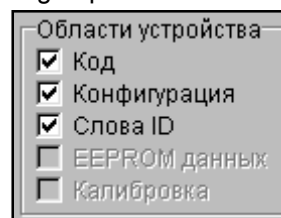
Программное обеспечение также сохраняет последние 10 используемых типов устройств в меню **ИсторияТипов**:

Пункт **Очистить буфер** в данном меню полностью аналогичен соответствующему пункту в диалоге **Выберите устройство** и может быть использован вместо него. Кроме того, это единственный способ определить необходимость стирания содержимого буфера при изменении типа устройства через список истории.



## Выбор областей для программирования

Большинство семейств микроконтроллеров (а также ряд микросхем памяти) имеют дополнительные области конфигурации и идентификации. PicProg обрабатывает информацию об этих областях, если ее вводят либо вручную, либо из загружаемого файла. Можно выбрать области для программирования или проверки, отметив нужные пункты на панели **Области устройства**. Области, недоступные в данном устройстве, будут запрещены.



При загрузке файла PicProg отмечает лишь области, содержащиеся в файле. Кроме того, если Вы меняете содержимое любой из областей вручную, PicProg отмечает соответствующий пункт на панели **Области устройства** автоматически.

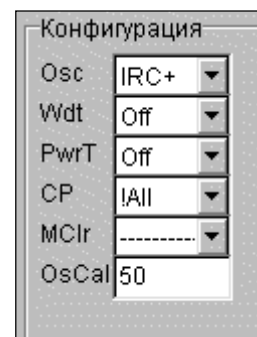
### Как PicProg пользуется информацией об Областих Устройства:

PicProg программирует/верифицирует/загружает **лишь** области, отмеченные на панели **Области**. PicProg проверяет на чистоту/стирает/читает **все** области, независимо от состояния панели **Области**.

## Установка битов конфигурации, Идентификаторов и Калибровочных значений

Если выбранное устройство имеет область конфигурации, ее содержимое можно ввести вручную в панели **Конфигурация**:

Большинство пунктов имеет список возможных значений битов конфигурации. В каждом таком списке имеется один дополнительный пункт в виде горизонтальной линии (как для бита **MCir** в примере). Выбор этого пункта означает, что состояние данного бита не определено пользователем и во время программирования останется неизменным. Список битов и их значений может сильно отличаться для разных типов устройств. В основном названия битов не требуют пояснений. Однако имеется ряд случаев, которые мы хотели бы прокомментировать (см. следующий раздел).



В некоторых системах разработки имеется возможность явно определить состояние битов конфигурации в исходном тексте. Например, в MPASM можно воспользоваться директивой `__CONFIG`. Там же можно установить слова идентификации директивой `__IDLOCS`. Каждый раз при компиляции проекта или загрузке HEX файла указанные

# Руководство пользователя PICPROG

биты и идентификатор будут установлены в соответствии с содержанием данных директив.

Панель **Конфигурация** может содержать также калибровочные значения для некоторых типов устройств (например, калибровку Генератора для PIC12Cxxx, как показано на примере). Если Вы хотите запрограммировать эти значения, следует ввести их **шестнадцатеричные** значения в соответствующей строке редактирования и отметить пункт **Калибровка** в панели **Области устройства**. Если Вы не хотите, чтобы данные значения были запрограммированы в устройстве, просто снимите эту отметку перед программированием. В результате чтения устройства его калибровочные значения отображаются в панели **Конфигурация**.

Если устройство имеет область идентификатора, Вы увидите на экране панель **ID**: В случае Microchip PIC, в данной панели следует вводить 4-хразрядное шестнадцатеричное значение ID.



При работе с микроконтроллерами Scenix SX в данное поле вводится 8-символьная ASCII-строка (в соответствии с форматом, принятом в пакетах Parallax SXKey и Scenix SASM).



В ряде случаев ID может быть определен в исходном тексте программы.

## Комментарии по поводу битов конфигурации

Данный раздел содержит разъяснения по поводу некоторых битов конфигурации, которые, на наш взгляд, не являются интуитивно понятными.

- **Osc** для PIC12C6xx включает значения **IRC** и **ERC**, а также **IRC+** и **ERC+**. **IRC** и **ERC** – это Внутренний и Внешний RC генератор соответственно, а значения **IRC+** и **ERC+** означают, что при тех же типах генератора на выходе OSC2 микросхемы будет присутствовать частота  $F_{osc}/4$ .
- Биты Защиты (**CP**) для некоторых PICов могут иметь значок **!** перед основным значением (например, **!AI1**). Это означает, что данный тип устройства может иметь Нестираемую Защиту. Хотя Microchip и не говорит об этом открыто, но мы обнаружили, что в большинстве случаев (мы, конечно, говорим только о **/JW** устройствах) в этих устройствах может быть установлена и Стираемая Защита. Поэтому для таких устройств мы определяем два набора битов защиты: Нестираемая (со значком **!**) и Стираемая (без значка **!**). Какой защитой пользоваться – решать Вам, хотя мы и не понимаем, зачем нужна нестираемая защита в устройствах **/JW**. Программа PicProg может уведомить Вас, если Вы пытаетесь установить Нестираемую Защиту, или автоматически преобразовывать такую защиту в Стираемую (см. [Глава 5. Конфигурация PicProg для Windows](#)).

**Примечание:** Ассемблер Microchip MPASM выбирает Нестираемую Защиту при определении битов защиты директивой `__CONFIG`.

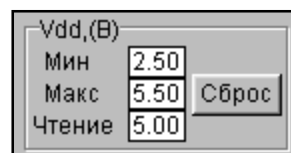
- Некоторые биты конфигурации могут содержать значок **?** в начале значения (например **?RC** в PIC12C6xx). Это означает, что данное значение является запрещенным и не рекомендуется к использованию производителем микросхемы. При программировании устанавливать такое значение не стоит, хотя Вы можете увидеть его после чтения устройства, если оно запрограммировано неверно.
- Значения Калибровки Генератора для PIC12Cxxx записываются в виде: **MOVLW XX** (для 12C5xx) или **RETLW XX** (для 12C6xx), где **XX** - 8-битное шестнадцатеричное число. При вводе значения в строку **OsCal** следует указывать только значение **XX** (PicProg автоматически дополняет это значение до нужной формы).

# Руководство пользователя PICPROG

## Выбор пределов напряжения

PicProg является промышленным программатором. Это означает, что он проверяет устройство на чистоту и верифицирует его при нижнем и верхнем пределах напряжения питания, чтобы обеспечить надежную работу устройства в диапазоне напряжений. Установка пределов позволяет изменить Vdd при программировании, верификации и чтении устройства. Напряжения регулируются шагами по 0.05В:

**Мин и Макс** – Напряжения проверки и верификации  
**Чтение** – Напряжение чтения  
**Сброс** – установка значений по умолчанию для всех напряжений



**Примечание:** Все напряжения должны лежать в пределах: 2.0В – 6.3В. Ошибочные напряжения будут автоматически заменяться на соответствующие предельные.

Напряжения по умолчанию могут сильно отличаться для разных типов устройств. Поэтому при изменении типа устройства все напряжения сбрасываются в исходные значения для нового типа устройства.

### Как PicProg пользуется Пределами Напряжений:

PicProg осуществляет операции при Vdd, установленном следующим образом:

- Проверяет на чистоту при мин. Vdd
- Читает при Vdd чтения
- Программирует при номинальном напряжении (не меняется пользователем)
- Верифицирует при мин. и макс. Vdd

Причины этого объясняются следующим:

Частично стертые ячейки EPROM будут считаться стертыми при повышенном напряжении, но нестертыми при низких напряжениях. В связи с этим PicProg осуществляет проверку на чистоту при мин. Vdd. Ячейка EPROM, запрограммированная не полностью, будет правильно читаться при низких напряжениях, но неправильно на высоких. В связи с этим PicProg верифицирует устройства при макс. и мин. Vdd. PicProg читает устройства при Vdd чтения (обычно 5V или 3.3V) - номинальном напряжении, при котором устройство должно работать.

## Выбор алгоритма программирования и режима Внутрисхемного Программирования (ISP)

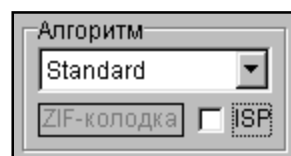
Хотя по умолчанию PicProg программирует устройства стандартными алгоритмами, определенными производителем, существует возможность изменить алгоритм программирования для определенного класса устройств (а именно, для микроконтроллеров Microchip PicMicro).

**Телесистемы** проделали большую работу за последние 5 лет с тем, чтобы найти наиболее оптимальные алгоритмы программирования устройств PIC /JW. Целью данной работы было обеспечение, с одной стороны, высокой надежности работы устройств, а с другой стороны, высокой скорости программирования и стирания устройств, а также увеличение ресурса программирования кристаллов. В результате появился список дополнительных алгоритмов, которые можно выбрать на панели

### Алгоритм.

Для большинства устройств доступен только один алгоритм - **Standard**, но для устройств PicMicro имеются еще 2 алгоритма. Мы

рекомендуем пользоваться алгоритмом **Standard** при промышленном программировании устройств, и алгоритмом **Light** при разработке с использованием устройств /JW (мы сами пользуемся таким алгоритмом в процессе разработки в течение трех последних лет без каких-либо проблем).



# Руководство пользователя PICPROG

**Внимание:** Хотя **Телесистемы** и считают свои оптимизированные алгоритмы весьма надежными для отладки устройств, мы не несем никакой ответственности за какие-либо отрицательные последствия при их применении, поскольку эти алгоритмы не сертифицированы производителями микросхем.

Панель **Алгоритм** содержит также пункт **ISP** (Внутрисхемное Программирование). Если данное поле разрешено (зависит от типа устройства и версии прошивки PicProg), то выбранное устройство можно программировать посредством 7-выводного ISP-разъема, расположенного в нижнем левом углу блока PicProg. Распайка кабеля ISP и специфические требования к программированию различных устройств изложены в главе **Приложение В. Внутрисхемное программирование с помощью PicProg**

Поле слева от пункта **ISP** сообщает о том, каким образом подключать устройство к программатору. В данном поле возможно появление следующих сообщений:

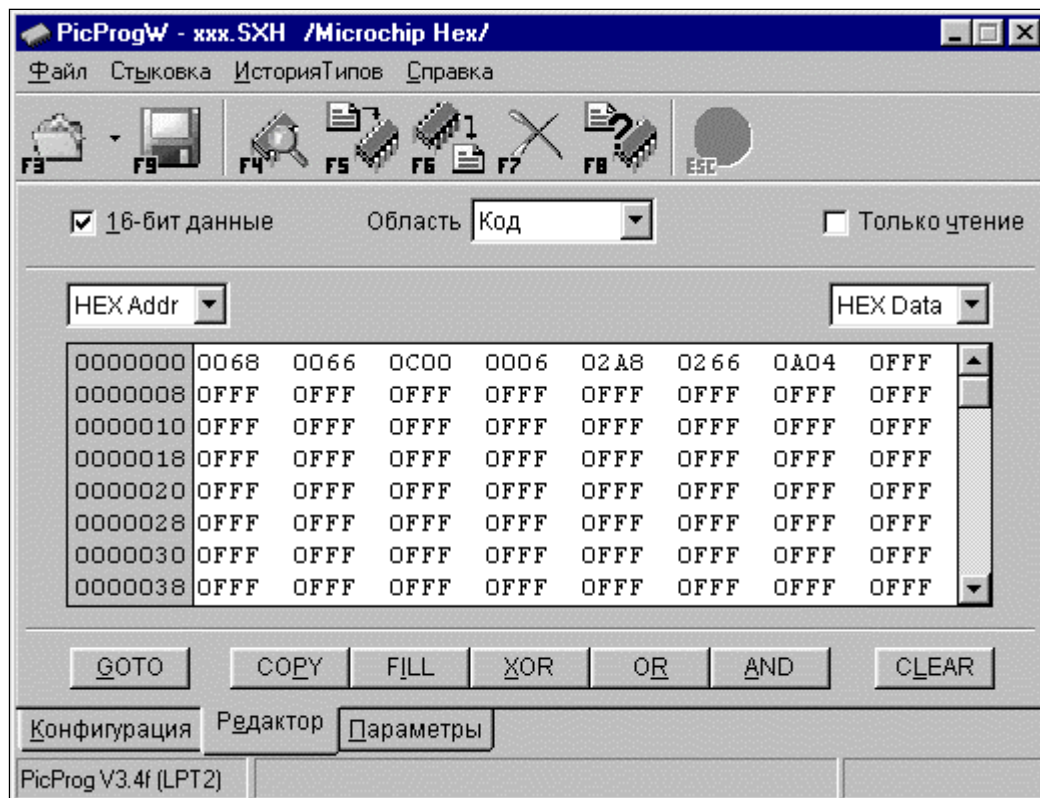
- **ZIF-колодка** – 40-выводная ZIF колодка, расположенная на блоке PicProg (используется для большинства устройств в корпусах DIP)
- **Адаптер XX** – требуется дополнительный адаптер устройства (с номером **XX**) (в основном для устройств в корпусах, отличных от DIP). Все доступные адаптеры перечислены в главе **Приложение А. Адаптеры устройств PicProg**.
- **ISP-разъем** – 7-выводной ISP-разъем Внутрисхемного Программирования (только если отмечен пункт **ISP**). Более подробно смотри **Приложение В. Внутрисхемное программирование с помощью PicProg**

**Примечание:** Некоторые типы устройств могут программироваться **только** через ISP-разъем. К ним относятся Toshiba TC58A040, Microchip HCS512 и Scenix SX48/52.

# Руководство пользователя PICPROG

## Редактирование области программ и EEPROM данных

PicProg для Windows имеет встроенный редактор. Доступ к нему открывается после выбора закладки **Редактор** в главном окне PicProg. Окно редактора отображает либо данные для программирования, либо данные, считанные из устройства.



Можно просмотреть или отредактировать область либо программ, либо данных EEPROM (при наличии). Для этого следует выбрать один из пунктов в списке **Область**. Данные в редакторе могут быть представлены в 16- или 8-разрядном виде. Их можно редактировать либо вручную, либо воспользовавшись рядом блочных функций, перечисленных в виде программных кнопок под окном редактора. Формат адресов и данных может независимо устанавливаться с помощью списков выбора, расположенных над окном редактора как Hexadecimal, Decimal или Octal. Следует помнить, что выбор определенного формата адресов или данных в редакторе приводит также к соответствующему изменению формата значений, используемых в блочных операциях, а также диалогах загрузки и сохранения файлов. В указанных случаях на формат соответствующих значений указывает буква, стоящая после самого значения (h – Hexadecimal, d – Decimal, o – Octal).

**Примечание:** При вводе значений для блочных функций операцию всегда можно отменить, нажав клавишу **Esc**.

## Установка микросхемы

После ввода всей необходимой информации следует установить микросхему в PicProg. Здесь возможны несколько вариантов (см. [Выбор алгоритма программирования и режима Внутрисхемного Программирования \(ISP\)](#)). При установке микросхемы в ZIF-колотку на блоке PicProg следует ориентироваться на условные обозначения, приведенные на верхней крышке программатора.

# Руководство пользователя PICPROG

**Примечание:** Существует общее правило для соблюдения правильной ориентации микросхемы при установке. Кристалл должен располагаться таким образом, чтобы его ключ был направлен к рычагу ZIF-колодки, а сам кристалл прижат к **нижней** части PicProg.

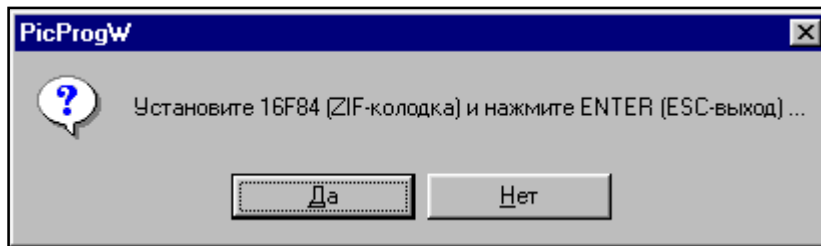
Правила подключения микросхем к дополнительным адаптерам и ISP-кабелю описаны в главах [Приложение А. Адаптеры устройств PicProg](#) и [Приложение В. Внутрисхемное программирование с помощью PicProg](#)

соответственно.

Процесс установки и извлечения микросхем в/из PicProg абсолютно безопасен, поскольку все выводы ZIF-колодки и ISP-разъема заземляются в ждущем режиме. Таким образом, исключаются проблемы статического электричества. Более того, если в процессе программирования происходит потеря связи программатора с компьютером, PicProg автоматически переходит в ждущий режим через 2 сек, позволяя безопасно извлечь кристалл.

**Примечание:** Не устанавливайте микросхему в PicProg, пока он находится в активном режиме (горят зеленый **“VPP”** и/или красный **“ERR/AUTO”** светодиоды), поскольку это может привести к полной неработоспособности кристалла.

Для обеспечения еще большей безопасности Вы можете отметить пункт **Вопрос “Вставьте устройство”** на закладке **Параметры** (см. [Глава 5. Конфигурация PicProg для Windows](#)). В результате каждый раз в начале процесса программирования или верификации Вы будете видеть следующее сообщение, напоминающее о необходимости правильной установки кристалла:



Кроме того, для устройств, имеющих Идентификатор Типа Устройства (например, для большинства устройств ATMEL) программатор может проверять этот идентификатор на соответствие заданному значению. Данную операцию можно и запретить (см. [Глава 5. Конфигурация PicProg для Windows](#)).

## Проверка устройства на чистоту



PicProg может проверить содержимое микроконтроллера или микросхемы памяти, чтобы проверить, все ли ячейки устройства находятся в исходном состоянии (состояние единицы для большинства устройств). Можно это делать автоматически перед программированием устройства или вручную, щелкнув мышью на кнопке **Проверка на чистоту** или нажав клавишу **F4**. PicProg проверяет устройство на чистоту при мин. Vdd (см. [Выбор пределов напряжения](#)) и сообщает результаты проверки в строке состояния главного окна.

Возможны следующие варианты результатов проверки:

- **Устройство чистое;**
- **Устройство не чистое;**
- **Устройство содержит только калибровочную информацию** – возможно для устройств PicMicro с предварительно записанными калибровочными коэффициентами (PIC12Cxxx, PIC16C50x, PIC14Cxxx); нормальное состояние для новых OTP и JW устройств;

# Руководство пользователя PICPROG

- **Устройство содержит только информацию о типе генератора** – возможно для устройств PicMicro с предварительно установленным типом генератора (PIC16C5х-XT/LP/HS) – нормальное состояние для новых OTP устройств.

## Программирование устройства



При щелчке мышью на кнопке **Программирование** или нажатии клавиши **F5** запускается режим программирования, в процессе которого происходит запись кода в микросхему. Данный режим определяется следующей последовательностью:

**Примечание:** Программирование осуществляется только для выбранных областей устройства – см. [Выбор областей для программирования](#).

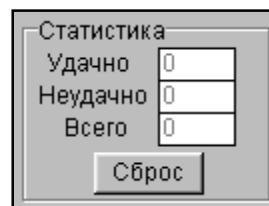
- PicProg автоматически стирает устройство, если оно основано на EEPROM (необязательная процедура, запрещенная по умолчанию – см. [Глава 5. Конфигурация PicProg для Windows](#)). Может быть полезно при программировании устройств, не чистых в исходном состоянии – например, PIC16F8х;
- PicProg проверяет выбранные **Области устройства** на чистоту (можно отключить – см. [Глава 5. Конфигурация PicProg для Windows](#)). Если устройство не чистое, PicProg попросит разрешения продолжить;
- PicProg записывает выбранные **Области устройства** (кроме **Конфигурации**) в кристалл;
- PicProg проверяет выбранные **Области устройства** (кроме **Конфигурации**) при макс. и мин. Vdd (проверка на одном или обоих пределах может быть отключена – см. [Глава 5. Конфигурация PicProg для Windows](#)) на соответствие заданной информации и сообщает результаты проверки в строке состояния главного окна;
- Если верификация прошла успешно, PicProg программирует биты конфигурации, если это необходимо (в случае ошибки верификации будет сделан запрос о продолжении);
- PicProg проверяет правильность записи конфигурации на одном или двух пределах Vdd и выводит результаты в строку состояния.

## Статистика программирования и Log-файл

PicProg учитывает результаты программирования всех микросхем в панели **Статистика** и Log-файле (необязательно).

Панель **Статистика** содержит следующую информацию:

- **Удачно** – количество удачно запрограммированных микросхем;
- **Неудачно** – количество неудачно запрограммированных микросхем;
- **Всего** – общее количество запрограммированных микросхем.



Счетчики обновляются после программирования каждого устройства и могут быть сброшены путем нажатия на кнопку **Сброс**. Если операция программирования была прервана еще до того, как начался реальный процесс программирования или стирания (например, во время проверки на чистоту), счетчики статистики **не** изменяются.

PicProg добавляет результаты программирования в конец Log-файла **PicProgW.log**, расположенного в том же каталоге, что и установленная программа (обычно C:\Program Files\PicProgW). Сохраняется информация о дате и времени программирования, имени запрограммированного файла, типе устройства, запрограммированном идентификаторе и количестве запрограммированных микросхем.

# Руководство пользователя PICPROG

## Пример Log-файла PicProg

```
.....
01/12/99 15:42:11 Most.hex          16C73A          ID:NO OK:1 Bad:0
01/13/99 13:42:33 Vddprog2.obj          12CE673         ID:1234 OK:0 Bad:1
02/02/99 17:45:47 TELEGRAF.hex          16C77           ID:0000 OK:0 Bad:1
02/02/99 17:47:04 TELEGRAF.hex          16C77           ID:0000 OK:0 Bad:1
.....
```

Вы можете заметить, что закладка **Параметры** главного окна (см. [Глава 5. Конфигурация PicProg для Windows](#)) не позволяет изменять пункт **Вести Log-файл**. Это сделано из соображений безопасности (позволяет избежать неконтролируемого несанкционированного программирования микросхем). Однако ведение Log-файла можно запретить вручную. Для этого следует закрыть PicProg для Windows, открыть файл **PicProgW.ini** (расположенный в том же каталоге, что и установленная программа) в любом текстовом редакторе (например, **Блокнот**) и изменить в нем строку: **LogFileEnable=1** на: **LogFileEnable=0**. При следующем запуске программы ведение Log-файла будет запрещено.

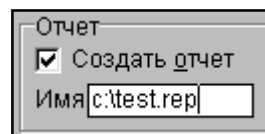
## Верификация устройства



Вы можете убедиться, что данные в текущем кристалле соответствуют содержимому буфера программы. Для этого следует щелкнуть мышью на кнопке **Верификация** или нажать клавишу **F8**. PicProg проверит выбранные области устройства (см. [Выбор областей для программирования](#)) и сообщит о результатах проверки в строке состояния. Верификация проводится при тех же значениях напряжения питания, что и программирование (см. [Выбор пределов напряжения](#)).

## Создание отчета программирования/верификации

Если Вы хотите получить более детальную информацию о результатах последней операции программирования или верификации, PicProg может создать для Вас файл отчета. Чтобы разрешить создание отчета, следует отметить пункт **Создать Отчет** на панели **Отчет**.



При загрузке файла в буфер программы PicProg устанавливает имя файла отчета, заменяя расширение входного файла на **.rep** (по умолчанию отчет будет создан в том же каталоге, что и загруженный файл). Однако имя и каталог создания файла всегда можно изменить, введя новую информацию в поле **Файл** панели **Отчет**. После программирования или верификации устройства Вы можете просмотреть содержание отчета, открыв его в любом текстовом редакторе.

## Пример файла отчета PicProg

Сравнение 16C84 с файлом COPIER.HEX при Vdd=6.00В:

```
000000: 0000 018A
000001: 0000 30CF
.....
002007: 0000 0017
.....
```

Как видно из примера, файл отчета содержит информацию обо всех ячейках, которые были неудачно запрограммированы или верифицированы.

**Примечание:** В результате повторения операции программирования или верификации с одним и тем же именем отчета содержимое нового отчета **заменяет** старый файл отчета.

## Чтение устройства



При щелчке мышью на кнопке **Чтение** или нажатии клавиши **F6** PicProg считывает содержимое микросхемы в программный буфер. В процессе чтения устройство также проверяется на чистоту, и результаты проверки выводятся в строку состояния (см. [Проверка устройства на чистоту](#)).

Независимо от состояния панели **Области устройства** PicProg читает (и проверяет) все устройство целиком, так что по окончании чтения все области устройства оказываются отмеченными.

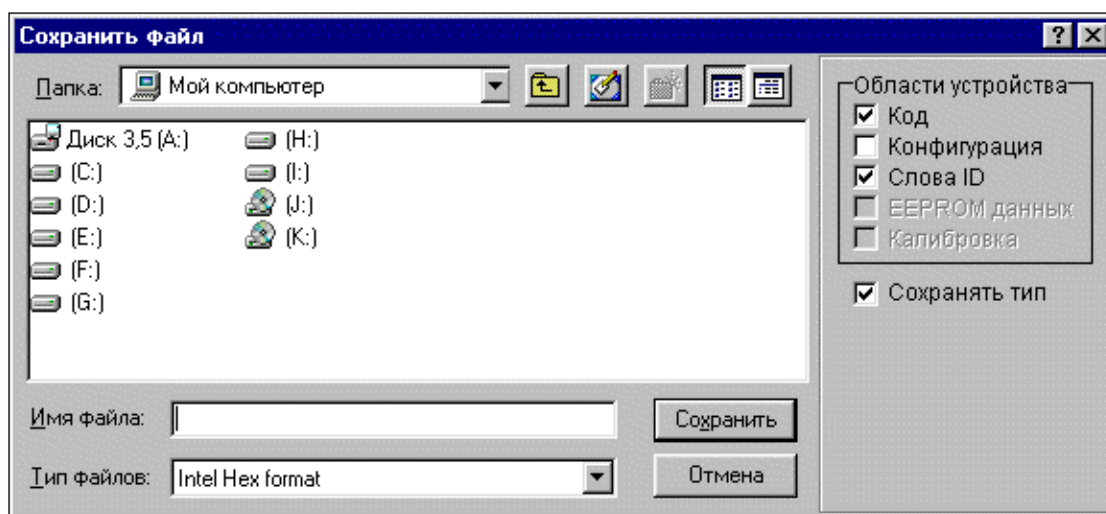
После чтения устройства имя загруженного файла меняется на тип устройства и запрещается слежение за файлом.

**Примечание:** Буфер программы полностью стирается перед каждой операцией чтения.

## Сохранение файла



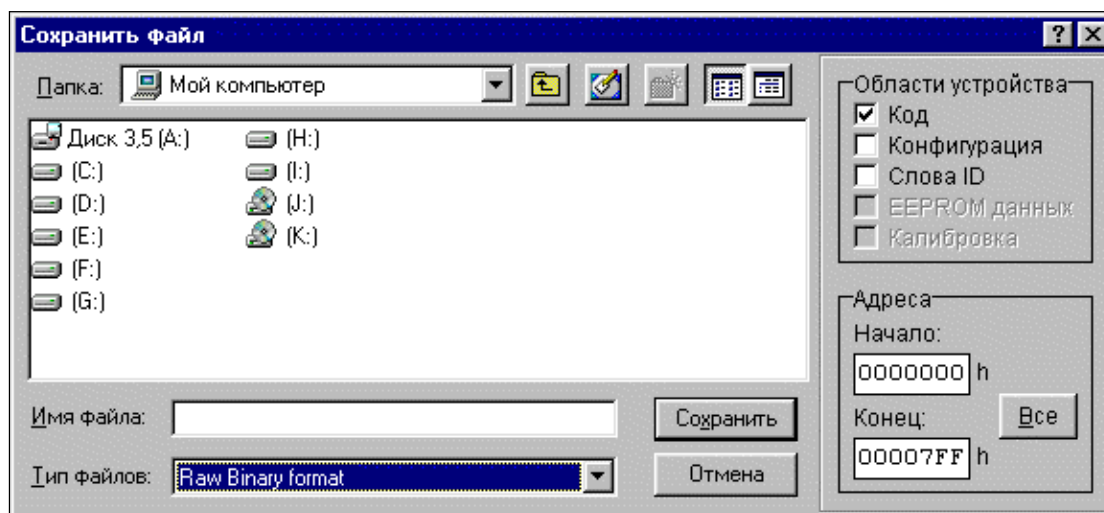
Сохранить содержимое буфера программы в файле можно, щелкнув мышью на кнопке **Сохранить в файле** или нажав клавишу **F9**. Появится диалог, в котором можно выбрать имя файла для сохранения информации и каталог, где этот файл должен располагаться. Можно также выбрать формат сохраняемого файла из списка **Тип файлов**. Содержимое буфера можно сохранить в формате Intel Hex или двоичного файла (**binary file**).



При сохранении информации в формате **Intel HEX** можно указать любой набор областей устройства, которые должны содержаться в файле, пользуясь панелью **Области устройства** диалога сохранения. Данная панель повторяет установки главного окна программы. Однако изменения данной панели в диалоге не сохраняются после закрытия диалога.

При выборе пункта **Сохранять тип** в конец файла будет добавлена запись о типе устройства и пределах напряжения верификации. Формат этой записи разработан фирмой **Телесистемы** и не поддерживается какими-либо другими средствами разработки. Если Вы сохранили данную запись в файле и пытаетесь загрузить такой файл в другой системе, то, возможно, получите сообщение об ошибке. В этом случае следует вручную удалить из файла вторую строку от конца (эта строка должна начинаться следующей последовательностью символов: **:040000FE ....** ).

При необходимости сохранения файла в бинарном формате вид диалога сохранения несколько меняется:



В этом случае можно выбрать только одну область устройства для сохранения в файле. При этом можно сохранить либо часть этой области, либо ее всю. Адреса начала и конца блока данных для сохранения в файле указываются в панели **Адреса**. Формат ввода адресов зависит от формата, выбранного для адресов в редакторе программы (см. [Редактирование области программ и EEPROM данных](#)). По умолчанию устанавливаются адреса всей выбранной области. Того же эффекта можно добиться, нажав кнопку **Все**.

**Примечание:** Следует всегда указывать как имя файла, так и его расширение независимо от выбранного формата.

## Стирание устройства на основе EEPROM



Большинство устройств на основе EEPROM- или Flash- памяти могут быть стерты электрически (под стиранием мы подразумеваем установку всех ячеек памяти в исходное состояние). Устройство можно стереть, щелкнув мышью на кнопке **Стирание** или нажав клавишу **F7**. Следует иметь в виду, что перед программированием устройства проводить его отдельное стирание необязательно, поскольку это входит в процедуру записи.

**Примечание:** Операция стирания не включает в себя последующую проверку устройства на чистоту, которая должна проводиться отдельно (см. [Проверка устройства на чистоту](#)).

## Стирание устройства на основе EPROM

Определенные типы устройств на основе EPROM (например, кристаллы /JW PicMicro или EPROM 27Cxxx в корпусе с окном) могут стираться ультрафиолетовой лампой. Обычная процедура при этом состоит в том, что микросхема помещается под лампу и через некоторое время, обычно определяемое экспериментальным путем, извлекается оттуда и проверяется на чистоту в программаторе. Если кристалл оказывается недостаточно стертым, процедура повторяется с самого начала. Можно, конечно, оставлять кристалл под лампой с большим запасом по времени. Но кому хочется ждать в 2-3 раза дольше, чем это действительно необходимо? Кроме того, длительные операции стирания неблагоприятно сказываются на ресурсе перезаписи микросхемы. PicProg может помочь Вам осуществлять процедуру ультрафиолетового стирания кристалла наиболее эффективным способом.

Вы просто устанавливаете микросхему в ZIF-панель программатора и устанавливаете над ней лампу для стирания (естественно, конструкция лампы должна это позволять). Затем щелкаете мышью на кнопке **Стирание** или нажимаете клавишу **F7**. PicProg быстро проверяет устройство на чистоту при минимальном напряжении питания (см.

# Руководство пользователя PICPROG

**Выбор пределов напряжения).** Если оказывается, что устройство не чистое, то программатор замыкает все выводы колодки на землю (что помогает уберечься от импульсов статического электричества) и переходит в режим ожидания на определенное время (по умолчанию 60 сек – можно изменить через [Глава 5. Конфигурация PicProg для Windows](#)).

Во время ожидания PicProg отображает время, прошедшее с начала операции в строке состояния. Как только интервал ожидания заканчивается (или Вы нажимаете клавишу **Enter**), PicProg снова проверяет устройство на чистоту – и повторяет эту процедуру до тех пор, пока устройство не оказывается чистым. Тогда программатор прерывает процедуру, дает звуковой сигнал и показывает в строке состояния время, затраченное на стирание устройства. Процедуру можно также прервать в любой момент, нажав клавишу **Esc**.

Как видно из приведенного описания, процесс стирания полностью автоматизируется (можно даже выполнять его в фоновом режиме), причем время ожидания точно соответствует необходимой степени стирания кристалла.

**Примечание:** Не следует выбирать мин. уровень Vdd намного меньше нижнего предела работы устройства в системе, поскольку в этом случае понапрасну расходуется ресурс кристалла.

## Снятие защиты с PIC16C(F)8x

Если Вы запрограммировали свою программу в кристалл семейства PIC16C(F)8x и установили защиту программы, а потом потеряли исходный текст программы, то в общем случае считать содержимое программы из микросхемы невозможно. PicProg может помочь Вам стереть биты защиты кода без стирания самого кода, позволяя таким образом восстановить утерянный текст программы. Для этого следует выполнить следующую последовательность операций:

- Установите защищенный PIC16C(F)8x в ZIF-колодку PicProg;
- Нажмите комбинацию клавиш **Control+F7**;
- PicProg выполнит процедуру стирания битов защиты (она может занять достаточно долгое время в зависимости от экземпляра микросхемы) и сообщит о результатах в строке состояния;
- В случае удаи PicProg содержимое микросхемы можно считать в буфер программы (см. [Чтение устройства](#));

**Примечание:** При чтении устройства со снятой защитой следует установить напряжение чтения равным 2.0V (см. [Выбор пределов напряжения](#)).

- В случае неудачи снятия защиты процедуру можно повторить еще два-три раза.

**Внимание:** Вы пользуетесь указанной процедурой на свой страх и риск. **Телесистемы** не гарантируют удаи при снятии защиты и не несут ответственности за возможную потерю кристалла в результате выполнения данной процедуры.

**Внимание:** Помните, что следует выполнять процедуру снятия защиты, только если это не нарушает ничьих авторских прав. В любом случае **Телесистемы** не несут ответственности за возможные отрицательные последствия, возникающие на этой почве.

## Снятие защиты с AT89Cх051/AT89C5х

С некоторых экземпляров микросхем серий AT89Cх051 и AT89C5х старых выпусков (примерно до начала 1998 года) биты защиты также могут быть сняты без стирания содержимого устройства. Для этого следует выполнить следующую последовательность операций:

- Установите защищенный кристалл AT89хххх в ZIF-колодку PicProg;
- Нажмите комбинацию клавиш **Control+F7**;
- PicProg выполнит процедуру стирания битов защиты (в данном случае она выполняется очень быстро);
- Теперь можно попытаться считать содержимое микросхемы в буфер программы (см. [Чтение устройства](#));

**Внимание:** Вы пользуетесь указанной процедурой на свой страх и риск. **Телесистемы** не гарантируют удачи при снятии защиты и не несут ответственности за возможную потерю кристалла в результате выполнения данной процедуры.

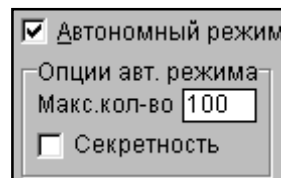
**Внимание:** Помните, что следует выполнять процедуру снятия защиты, только если это не нарушает ничьих авторских прав. В любом случае **Телесистемы** не несут ответственности за возможные отрицательные последствия, возникающие на этой почве.

## Глава 4. Автономный режим

Автономный режим позволяет запрограммировать серию микросхем без участия компьютера. Автономный режим может быть полезен в случае, когда компьютер недоступен или его использование является неразумным, например, в полевых условиях или в процессе производства. Чтобы запрограммировать устройства в автономном режиме, следует сначала загрузить в PicProg нужную информацию.

### Загрузка файла для автономного программирования

Чтобы получить доступ к командам и параметрам автономного программирования PicProg для windows следует отметить пункт **Автономный режим** в правом верхнем углу главного окна программы. На экране появится панель **Опции авт. Режима**, а кнопки на рабочей панели программы поменяют свои функции на команды, связанные с автономным режимом.



Следует загрузить файл с программой и настроить все параметры программирования так же, как и для обычного режима программирования (см. [Глава 3. Работа с PicProg для Windows](#)). Здесь возможно применение **всех** параметров программирования, включая: установку конфигурации и ID (см. [Установка битов конфигурации, Идентификаторов и Калибровочных значений](#)), правку программы и данных EEPROM (см. [Редактирование области программ и EEPROM данных](#)), выбор областей для программирования (см. [Выбор областей для программирования](#)), установку пределов напряжения (см. [Выбор пределов напряжения](#)), выбор алгоритма программирования и режима ISP (см. [Выбор алгоритма программирования и режима Внутрисхемного Программирования \(ISP\)](#)). Да, и в автономном режиме Вы можете пользоваться внутрисхемным программированием. И конечно, можно пользоваться всеми настройками параметров программирования (см. [Глава 5. Конфигурация PicProg для Windows](#)).

Кроме того, имеются два специфических для автономного режима параметра:

- **Макс. кол-во** микросхем, программируемых в автономном режиме – позволяет установить верхний предел (от 1 до 65535) количества микросхем, которые разрешается запрограммировать. Когда этот предел будет превышен, программатор заблокирует дальнейшее использование автономного программирования. Если **Макс. кол-во** установлено равным 0, предел программирования не устанавливается;
- **Секретность** – отметив этот пункт, Вы повышаете уровень секретности при использовании автономного программирования микроконтроллеров, имеющих защиту кода программы. В этом режиме при возникновении ошибок в процессе программирования PicProg сначала записывает биты конфигурации и только после этого прекращает выполнение операции и сообщает об ошибке. Таким образом, если в **Конфигурации** были установлены биты защиты (см. [Установка битов конфигурации, Идентификаторов и Калибровочных значений](#)), то невозможно будет считать информацию даже из частично записанных устройств.

После того, как все параметры автономного программирования заданы, щелкните мышью на кнопке **Загрузка в PicProg** или нажмите клавишу **F5**. Вся необходимая информация будет загружена во внутренний энергонезависимый буфер программатора.



**Примечание:** При каждой загрузке информации в PicProg счетчики статистики сбрасываются в ноль.

## Автономное программирование

Чтобы начать процедуру автономного программирования, установите кристалл в PicProg (см. [Установка микросхемы](#)) и нажмите кнопку **“START”** на блоке PicProg. Должен загореться красный светодиод **“ERR/AUTO”**, показывая, что процесс программирования начался. Если этого не случается, значит, либо в программатор ничего не загружено, либо превышен максимальный предел программирования (см. [Загрузка файла для автономного программирования](#)). Чтобы узнать причину, следует прочитать загруженную информацию из PicProg (см. [Чтение информации автономного программирования](#)).

PicProg записывает устройство, после чего гасит красный светодиод **“ERR/AUTO”**, показывая, что операция завершена успешно. Кроме того, программатор увеличивает на единицу энергонезависимый счетчик успешно запрограммированных устройств. После этого Вы можете заменить кристалл в программаторе и снова нажать кнопку **“START”**.

Если для автономного режима была разрешена проверка на чистоту (см. [Глава 5. Конфигурация PicProg для Windows](#)) и в процессе ее выполнения оказывается, что устройство нельзя дописать, PicProg прерывает процедуру, а красный светодиод **“ERR/AUTO”** начинает очень быстро мерцать. Чтобы погасить его, следует повторно нажать кнопку **“START”**, после чего микросхему можно извлечь из PicProg. В этом случае счетчики статистики не меняются, поскольку содержимое микросхемы не менялось.

В случае возникновения ошибок в процессе программирования или последующей верификации (если она была разрешена – см. [Глава 5. Конфигурация PicProg для Windows](#)) PicProg прерывает выполнение операции (возможно запрограммировав предварительно биты конфигурации – см. [Загрузка файла для автономного программирования](#)), а красный светодиод **“ERR/AUTO”** начинает медленно мигать. Увеличивается также внутренний счетчик испорченных кристаллов. Чтобы погасить светодиод, следует нажать кнопку **“START”**, после чего можно извлекать устройство из программатора.

**Примечание:** Не следует извлекать микросхему, пока горит красный светодиод **“ERR/AUTO”**. Это может испортить ее или повредить программатор.

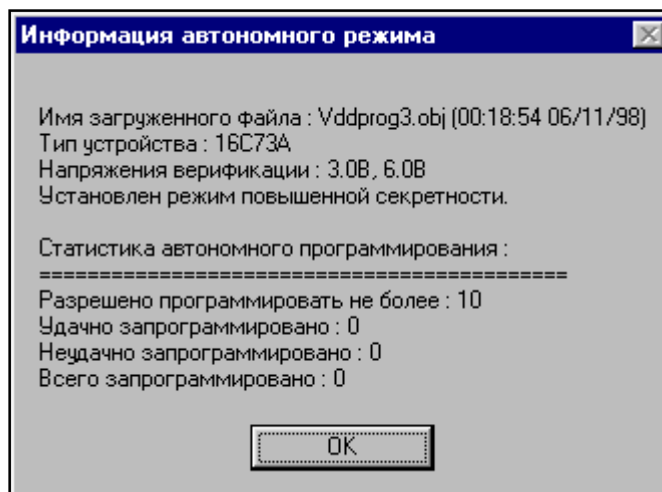
## Чтение информации автономного программирования



Информацию о файле, загруженном в PicProg для автономного программирования можно прочитать, щелкнув мышью на кнопке **Чтение информации...** или нажав клавишу **F6**, когда отмечен пункт **Автономное программирование**.

Если файл не был загружен, Вы увидите сообщение: **В PicProg ничего не загружено!** В строке состояния. В противном случае появится окно **Информация автономного программирования**, в котором показываются: имя и время создания загруженного файла, тип устройства, пределы напряжения верификации и параметры автономного программирования (см. [Загрузка файла для автономного программирования](#)).

Кроме того, отображается значение счетчиков статистики, включая общее количество



# Руководство пользователя PICPROG

---

запрограммированных устройств и отдельно количество удачно и неудачно запрограммированных кристаллов (см. [Автономное программирование](#)).

**Примечание:** Из соображений секретности содержимое загруженного файла прочитать невозможно.

## Стирание загруженного файла

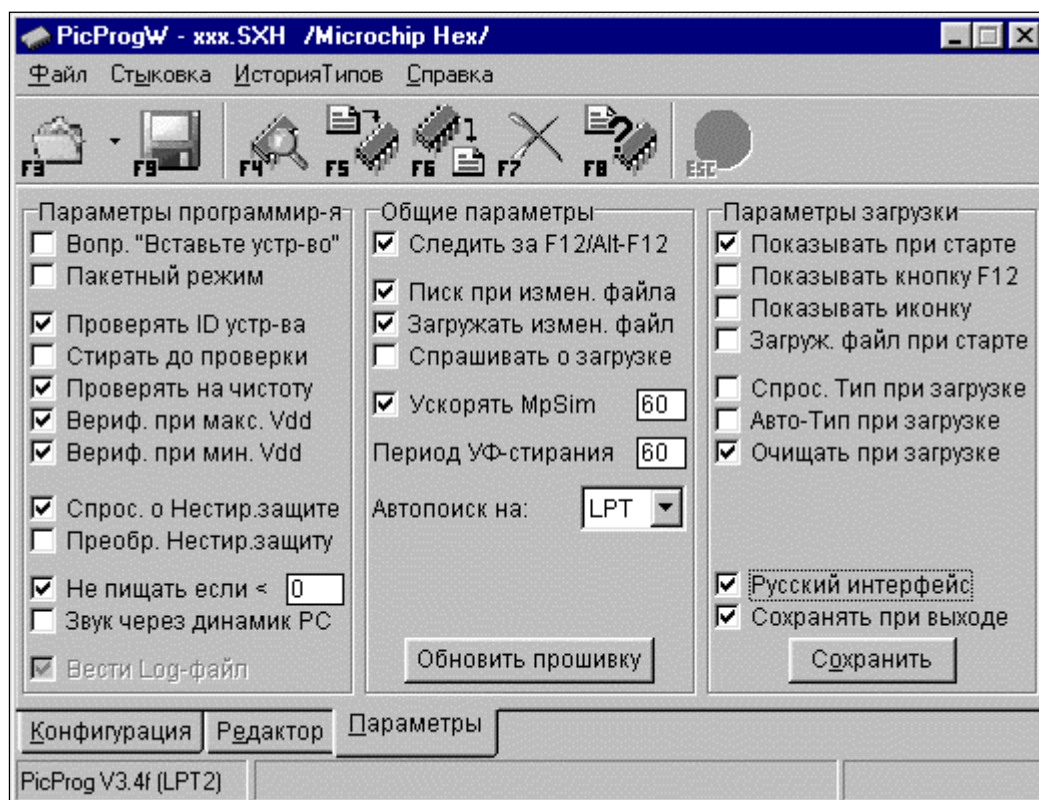


Чтобы стереть информацию о файле, загруженном в PicProg для автономного программирования, щелкните мышью на кнопке **Стирание загруженного файла** или нажмите клавишу **F7**, когда отмечен пункт **Автономный режим**.

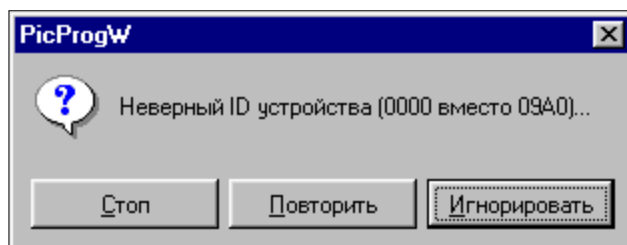
Вся информация стирается из внутреннего буфера PicProg и при следующей попытке считать ее (см. [Чтение информации автономного программирования](#)) Вы увидите сообщение: **В PicProg ничего не загружено!**

## Глава 5. Конфигурация PicProg для Windows

В данной главе описаны параметры конфигурации PicProg для Windows. Эти параметры расположены на закладке **Параметры** главного окна программы и автоматически сохраняются (если отмечен пункт **Сохранять при выходе**). Независимо от этого текущие параметры можно сохранить, нажав кнопку **Сохранить**.



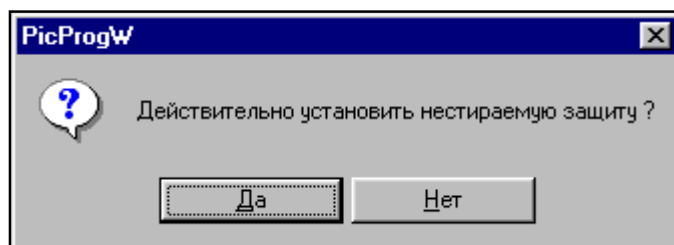
- **Вопрос “Вставьте устройство”** – разрешает выдачу предупреждения о необходимости установки микросхемы в PicProg (см. [Установка микросхемы](#));
- **Пакетный режим** – установка приводит к тому, что каждый раз после программирования устройства автоматически вызывается окно “Вставьте устройство”, упрощая таким образом процедуру программирования серии устройств (автоматически отмечает предыдущий пункт).
- **Проверять ID устройства** – разрешает проверку ID типа устройства на правильность (при несовпадении показывает следующее предупреждение);



- **Стирать до проверки** – разрешает автоматическое стирание устройства в самом начале процесса программирования (см. [Программирование устройства](#));
- **Проверять на чистоту** – разрешает проверку устройства на чистоту перед программированием (см. [Программирование устройства](#));

# Руководство пользователя PICPROG

- **Верификация при макс. Vdd** – разрешает верификацию устройства при верхнем пределе напряжения после программирования (см. [Программирование устройства](#)) или во время **Верификации** (см. [Верификация устройства](#));
- **Верификация при мин. Vdd** – разрешает верификацию устройства при верхнем пределе напряжения после программирования (см. [Программирование устройства](#)) или во время **Верификации** (см. [Верификация устройства](#));
- **Вопрос о Нестираемой защите** – разрешает выдачу сообщения в случае попытки установки нестираемой защиты (при ответе **Нет** защита будет преобразована в стираемую форму) – см. [Комментарии по поводу битов конфигурации](#);



- **Преобразовывать Нестираемую Защиту** – автоматически преобразовывать нестираемую форму защиты в стираемую перед программированием (см. [Комментарии по поводу битов конфигурации](#));
- **Не пищать если короче** – если пункт не отмечен, звуковой сигнал будет выдаваться при каждом завершении процесса программирования. Если пункт отмечен, а соседнее поле равно 0, сигнал выдаваться не будет. При установке поля в другое значение сигнал будет выдаваться, только если время программирования превысило указанное время в секундах. Таким образом, можно получать звуковое подтверждение только при долгом программировании;
- **Звук через динамик PC** – если пункт не отмечен, звуковые сигналы будут выдаваться через звуковую карту Вашего компьютера. Если пункт отмечен, то все звуковые сигналы будут выдаваться через системный динамик компьютера;
- **Вести Log файл** – отмечен, если статистика программирования сохраняется в Log файле (см. [Статистика программирования и Log-файл](#));
- **Следить за F12/Alt-F12** – делает функцию загрузки из MPLAB доступной на уровне системы (см. [Интеграция в MPLAB](#));
- **Писк при изменении файла** – разрешает звуковой сигнал при каждом изменении входного файла (см. [Слежение за файлами](#));
- **Загружать измененный файл** – перегружает входной файл каждый раз при его изменении (см. [Слежение за файлами](#));
- **Спрашивать о загрузке** – при отмеченных предыдущем и данном пунктах спрашивает пользователя о необходимости загрузки измененного файла (иначе загружает автоматически) – см. [Слежение за файлами](#);
- **Ускорять MPSIM** – разрешает работу ускорителя режима Run MPSIM (см. [Интеграция в MPLAB](#)). Величина ускорения сильно зависит от компьютера и регулируется соседним полем;
- **Период УФ-стирания** – определяет время (в секундах), после которого устройство на основе EPROM будет проверяться на чистоту при стирании его в (см. [Стирание устройства на основе EPROM](#));
- **Автопоиск на:** – определяет диапазон портов, на которых будет производиться автопоиск программатора. Выбирается из набора: **LPT** – все доступные параллельные порты, **COM** – все доступные последовательные порты или **ALL** – все доступные параллельные и последовательные порты (см. [Начальные сведения](#));

- **Обновить прошивку** – если в программатор установлен флэш-процессор, позволяет обновить прошивку программатора внутрисхемно (см. [Внутрисхемное обновление прошивок](#));
- **Показывать при старте** – разрешает показывать на экране главное окно в развернутом виде при каждом запуске PicProg для Windows. В противном случае это окно будет минимизировано на панели задач (если запрещен показ **системной иконки**) или вообще не видно (если показ **системной иконки** разрешен) и будет открываться только после загрузки файла (см. [Загрузка файлов](#)). Может быть полезно, если запускать PicProg для Windows одновременно с запуском самой Windows;
- **Показывать кнопку F12** – разрешает показывать на экране кнопку **F12** поверх всех остальных окон в системе (автоматически запрещает показ **системной иконки**);
- **Показывать иконку** – разрешает показывать на экране **системную иконку** (автоматически запрещает показ кнопки **F12**);
- **Загружать файл при старте** – разрешает автоматическую загрузку последнего открытого файла при запуске PicProg для Windows;
- **Спрашивать Тип при загрузке** – разрешает запрос о выборе типа устройства каждый раз при загрузке файла. В противном случае вопрос задается, только если информации о типе не содержится в самом файле (или соответствующем .cod файле) – см. [Загрузка файлов](#);
- **Авто-Тип при загрузке** – разрешает автоматическую подстановку предыдущего типа при загрузке файлов, не содержащих информацию о типе. Если отмечен предыдущий пункт, указывает на предыдущее значение в диалоге выбора типа устройства (см. [Выбор типа устройства](#));
- **Очищать при загрузке** – разрешает очищать содержимое буфера перед загрузкой очередного файла. При загрузке файла с помощью диалога загрузки данное значение может быть изменено непосредственно в диалоге (см. [Загрузка файлов](#));
- **Русский интерфейс** – переключает язык интерфейса с русского на английский;

## Глава 6. Работа с PicProg для DOS

PicProg для DOS – это программа, предназначенная в основном для пакетного программирования микросхем на компьютерах с ограниченными ресурсами (PC AT 286+). Все параметры работы программы полностью задаются из командной строки.

### Запуск PicProg для DOS

Запуск PicProg для DOS производится командной строкой следующего формата:

```
Picprog.exe <ком> [-<пар> [-<пар>]] [[<имя_файла1>][+<имя_файла2>]...]
```

Первым параметром командной строки всегда является команда (см. [Список команд](#)), определяющая операцию, которую необходимо выполнить над заданным устройством. Далее могут следовать один или несколько параметров (см. [Список параметров](#)) в произвольном порядке (каждая опция должна начинаться со знака '-' (минус)). Если выполняемая операция, определяемая командой, требует указания файла, то имя этого файла может быть указано до или после параметров командной строки. Для операций программирования и верификации могут быть указаны и несколько файлов (несколько файлов можно задать и с помощью знаков '\*' и '?').

**Примечание:** Для микросхем, имеющих область данных EEPROM (PIC, AVR, ...), можно задать одновременное программирование файлов с кодом программы и с данными EEPROM. Для этого следует их указать в следующем виде:

```
<имя_файла_кода>:<имя_файла_данных>
```

Все команды и параметры могут задаваться в любом регистре.

При запуске программы без команды выводится справка по командам и параметрам.

### Список команд

**p** – запрограммировать из файла(ов) (если файл не указан, записать только конфигурацию и/или ID)  
**pr** – аналогично p, но с созданием файла отчета (имя файла отчета совпадает с именем первого файла, но с расширением **.rep**)  
**v** – сравнить содержимое устройства с файлом(ами)  
**vr** – аналогично v, но с созданием файла отчета (имя файла отчета совпадает с именем первого файла, но с расширением **.rep**)  
**b** – проверить устройство на чистоту (проводится при мин. Vdd)  
**e** – для устройств на основе EEPROM/Flash – стереть устройство  
**e** – для УФ-стираемых устройств – периодически проверять устройство на чистоту (при мин. Vdd) до тех пор, пока устройство не окажется чистым (при этом будет выдан звуковой сигнал)  
**ex** – стереть защиту в PIC16C(F)8x  
**r** – считать содержимое устройства в файл (HEX файл, содержащий все области устройства и информацию о типе устройства)  
**rm** – считать содержимое устройства в файл (HEX файл, содержащий все области устройства)  
**rh** – считать содержимое устройства в файл (HEX файл, содержащий только область программы)  
**rb** – считать содержимое устройства в файл (двоичный файл, содержащий только область программы)  
**c** – преобразовать файл в формат HEX (содержащий все области устройства и информацию о типе устройства); выходной файл будет иметь расширение **.obj**  
**cm** – преобразовать файл в формат HEX (содержащий все области устройства); выходной файл будет иметь расширение **.hex**

# Руководство пользователя PICPROG

**ch** – преобразовать файл в формат HEX (содержащий только область программы); выходной файл будет иметь расширение .hex  
**cb** – преобразовать файл в двоичный формат (содержащий только область программы); выходной файл будет иметь расширение .bin  
**d** – загрузить в PicProg файл для автономного программирования  
**dr** – вывести на экран информацию о файле, загруженном в PicProg для автономного программирования и статистику автономного программирования  
**de** – стереть из PicProg информацию автономного программирования  
**u** – обновить прошивку PicProg внутрисхемно (только если установлен флэш-процессор)

## Список параметров

**-pl<порт>** - явно указать номер LPT-порта, к которому подключен PicProg (от 1 до 3)  
**-pl** – искать PicProg на всех доступных LPT-портах (значение по умолчанию)  
**-pc<порт>** - явно указать номер COM-порта, к которому подключен PicProg (от 1 до 4)  
**-pc** – искать PicProg на всех доступных COM-портах  
**-pa** – искать PicProg на всех доступных LPT- и COM-портах  
**-t<тип>** - явно указать тип устройства (н-р, -t16C74 или -t24LC01)  
**-vl<vмин>** - задать мин. Vdd (Вольт) для проверки на чистоту и верификации (н-р, -vl3.5)  
**-vh<vмакс>** - задать макс. Vdd (Вольт) для верификации (н-р, -vh5.75)  
**-vr<vчтен>** - задать Vdd (Вольт) для чтения (н-р, -vr4.5)  
**-i<id>** - задать ID для программирования в устройство (для Microchip PIC - до 4 HEX цифр (н-р -iF32D); для Scenix SX – до 8 ASCII символов (н-р -iSX\_Demo1)  
**-is<id>** - задать ряд ID – при программировании серии устройств, ID каждого следующего устройства будет на единицу больше, чем у предыдущего (начиная со значения <id>)  
**-a[<алг>]** – выбрать алгоритм программирования <алг>, где <алг> - буква из набора [s,q,l] (по умолчанию выбран 's' – стандартный)  
**-a+** - выбрать режим Внутрисхемного Программирования (через ISP-разъем)  
**-a+<алг>** - выбрать алгоритм внутрисхемного программирования  
**-cq** – запретить появление предупреждения о записи нестираемой защиты PicMicro  
**-ct** – всегда преобразовывать нестираемую форму защиты в стираемую  
**-s[b][l][h][r][k][e][i][c][s[<ts>]][t<tc>]** – установить следующие параметры  
i – запретить проверку идентификатора устройства  
e – всегда стирать устройство перед программированием  
b – запретить проверку на чистоту перед программированием  
l – запретить верификацию при мин. Vdd  
h – запретить верификацию при макс. Vdd  
s – запретить звуковые сигналы в конце процесса программирования  
**s<ts>** - разрешить звуковые сигналы, если время программирования превышает <ts> (секунд)  
**s0** – разрешить звуковые сигналы по окончании процесса программирования  
**c** – разрешить запуск процесса программирования и по кнопке "START"  
**r** – запретить ведение лог-файла  
**t<tc>** - установить период проверки устройства на чистоту во время стирания УФ-стираемых устройств <tc> (секунд); по умолчанию – 60сек  
**k** – безклавишный режим – в этом режиме в процессе программирования вообще не требуется нажимать клавиши на клавиатуре (полезно для программирования с помощью пакетных файлов DOS). Программа возвращает следующие коды:  
0 – программирование прервано пользователем  
1 – во время программирования возникли ошибки  
2 – программирование прошло успешно  
>2 – различные ошибки связи и командной строки

Имеются также два параметра, специфических для автономного режима:

**-dm<сmax>** - установить верхний предел программируемых кристаллов равным <сmax> (от 1 до 65535; по умолчанию предел не устанавливается)  
**-ds** – установить режим повышенной секретности автономного программирования

## Установка битов конфигурации

Биты конфигурации задаются параметрами следующего вида (в одном параметре можно определять любое количество битов, а командной строке может быть любое количество таких параметров):

**-f[<конф>]...[<конф>]**

где **<конф>** может быть записан в следующем виде:

**<конф>** - прожечь соответствующий бит

**<fuse>+** - разрешить соответствующий бит

**<fuse>-** - запретить соответствующий бит

**<fuse>!** – для битов защиты установить Нестираемую Защиту

Если какой-либо бит конфигурации не определен ни в командной строке, ни в самом файле, его состояние в процессе программирования не будет изменено.

Ниже перечислены возможные значения битов конфигурации для различных семейств микросхем:

### Конфигурация для PIC12/14/16Cxxx:

**R** – выбрать тип тактового генератора RC

**H** - выбрать тип тактового генератора HS

**X** - выбрать тип тактового генератора XT

**L** - выбрать тип тактового генератора LP

**I** - выбрать тип тактового генератора Встроенный RC (для PIC12/14Cxxx)

**E** - выбрать тип тактового генератора Внешний RC (для PIC12Cxxx)

**I+**, **E+** - то же, что и **I** и **E** , но на выходе OSC2 будет присутствовать частота Fosc/4 (для PIC12C6xx/505)

**W[+/-]** – разрешить/запретить сторожевой таймер

**P[+/-]** – разрешить/запретить таймер Power-on

**B[+/-]** – разрешить/запретить Brown-Out Detector

**C[!|+|-5|7]** – установить Защиту Кода (нестираемую/стираемую/выключить/50%/75%)

**D[!|+/-]** установить защиту данных EEPROM (для PIC14Cxxx и PIC16Fxxx)

**Y[+/-]** – разрешить/запретить аппаратный контроль по четности (для PIC16C62x/64x/66x/715)

**M[+/-]** – разрешить/запретить внешний вывод MCLR для PIC12Cxxx/16C50x

**V[0|1|2|3]** – установить напряжение Brown-Out 4.5B/4.2B/2.7B/2.5B (для PIC16C77x)

**F[+/-]** – разрешить/запретить бит WRT (для PIC16F87x)

**G[+/-]** – разрешить/запретить бит Debug (для PIC16F87x)

**T[+/-]** – разрешить/запретить бит LVP (для PIC16Fxxx)

### Конфигурация генератора для PIC16C745/765:

**H** – выбрать тип тактового генератора HS

**E** – выбрать тип тактового генератора EC

**H+** – выбрать тип тактового генератора HS с PLL

**E+** – выбрать тип тактового генератора EC с PLL

### Конфигурация генератора для PIC16F62x и PIC16C717/770/771:

**L** - выбрать тип тактового генератора LP

**X** - выбрать тип тактового генератора XT

**H** - выбрать тип тактового генератора HS

**E** – выбрать тип тактового генератора EC

**I** - выбрать тип тактового генератора Встроенный RC

**R** - выбрать тип тактового генератора Внешний R

**I+**, **R+** - то же, что и **I** и **R** , но на выходе OSC2 будет присутствовать частота Fosc/4

# Руководство пользователя PICPROG

## Конфигурация для PIC17Cxxx

**E** – выбрать тип тактового генератора EC  
**R** - выбрать тип тактового генератора RC  
**X** - выбрать тип тактового генератора XT  
**L** - выбрать тип тактового генератора LF  
**W1** – установить режим сторожевого таймера 1:1  
**W2** – установить режим сторожевого таймера 1:256  
**W6** – установить режим сторожевого таймера 1:64  
**W0** – запретить сторожевой таймер  
**ZM** – выбрать режим микропроцессора  
**ZC** – выбрать режим микроконтроллера  
**ZX** – выбрать режим расширенного микроконтроллера  
**ZP** – выбрать режим защищенного микроконтроллера

## Конфигурация для PIC18Cxxx

**E** – выбрать тип тактового генератора EC  
**E+** – выбрать тип тактового генератора EC с  $F_{osc}/4$  на OSC2  
**R** - выбрать тип тактового генератора Внешний RC  
**R+** - выбрать тип тактового генератора Внешний RC с  $F_{osc}/4$  на OSC2  
**X** - выбрать тип тактового генератора XT  
**L** - выбрать тип тактового генератора LP  
**H** - выбрать тип тактового генератора HS  
**H+** - выбрать тип тактового генератора HS с PLL  
**W-** – запретить сторожевой таймер  
**W0[1|2|3|4|5|6|7|8]** – установить режим сторожевого таймера  
1:1/1:2/1:4/1:8/1:16/1:32/1:64/1:128  
**N[+|-]** – разрешить/запретить OscEn  
**P[+|-]** – разрешить/запретить таймер Power-on  
**B[+|-]** – разрешить/запретить Brown-Out Detector  
**V[0|1|2|3]** – установить напряжение Brown-Out 4.5V/4.2V/2.7V/2.5V  
**F[+|-]** – разрешить/запретить Мультиплексирование CCP2  
**S[+|-]** – разрешить/запретить Сброс по Стеку  
**T[+|-]** – разрешить/запретить Детектор Низкого Напряжения  
**C[!|+|-]** – установить Защиту Кода (нестираемую/стираемую/выключить)

## Конфигурация для SX (серийные кристаллы)

**H1** – выбрать тип тактового генератора HS1  
**H2** – выбрать тип тактового генератора HS2  
**H3** – выбрать тип тактового генератора HS3  
**L1** – выбрать тип тактового генератора LP1  
**L2** – выбрать тип тактового генератора LP2  
**R** – выбрать тип тактового генератора Внешний RC  
**X1** – выбрать тип тактового генератора XT1  
**X2** – выбрать тип тактового генератора XT2  
**I[0|1|2|3]** – выбрать тип тактового генератора встроенный RC 4MHz/1MHz/128kHz/32kHz  
**W[+|-]** – разрешить/запретить сторожевой таймер  
**C[+|-]** – разрешить/запретить Защиту Кода  
**Y[+|-]** – разрешить/запретить режим Sync  
**T[+|-]** – разрешить/запретить режим Turbo (SX18/20/28)  
**N[+|-]** – разрешить/запретить XTAL (SX48/52)  
**D[+|-]** – разрешить/запретить IFBD  
**S[+|-]** – разрешить/запретить расширенный Option/Stack (SX18/20/28)  
**K[+|-]** – разрешить/запретить Sleep clock (SX48/52)  
**F[+|-]** – разрешить/запретить CF  
**P[0|1|2|3]** – установить время WDR 60ms/960ms/60us/18ms (SX48/52)  
**B[0|1|2|3]** – установить напряжение Brown-Out 4.2V/2.6V/2.2V/Off  
**A[0|1|2|3|4|5|6|7]** – выбрать калибровочный коэффициент встроенного RC генератора

# Руководство пользователя PICPROG

---

V[0|1|2|3] – выбрать калибровочный коэффициент BOR

## Конфигурация для SX (бета/инженерные кристаллы)

H – выбрать тип тактового генератора HS

L – выбрать тип тактового генератора LP

R – выбрать тип тактового генератора Внешний RC

X – выбрать тип тактового генератора XT

I[0|1|2|3|4|5|6|7] – выбрать IntRC 4MHz/2MHz/1MHz/500kHz/250kHz/125kHz/62kHz/31kHz

W[+|-] – разрешить/запретить сторожевой таймер

C[+|-] – разрешить/запретить Защиту Кода

Y[+|-] – разрешить/запретить режим Sync

T[+|-] – разрешить/запретить режим Turbo (SX18/20/28)

S[+|-] – разрешить/запретить расширенный Stack

O[+|-] – разрешить/запретить расширенный Option

K[+|-] – разрешить/запретить Sleep clock (SX48/52)

F[+|-] – разрешить/запретить CF

B[+|-] – разрешить/запретить Brown-Out Detector

P[0|1|2|3|5|6|7] – выбрать WDR 60ms/480ms/960ms/1.92s/60us/7.7ms/18ms (SX48/52)

A[0|1|2|3|4|5|6|7] – выбрать калибровочный коэффициент встроенного RC генератора

## Конфигурация для AVR

I – выбрать тип тактового генератора Встроенный RC

E – выбрать тип тактового генератора Внешнее тактирование или кварц

X – выбрать тип тактового генератора XT (для ATtiny10/11)

L – выбрать тип тактового генератора LP (для ATtiny10/11)

R – выбрать тип тактового генератора Внешний RC (для ATtiny10/11)

S[+|-] – разрешить/запретить режим последовательного программирования

C[1|2|3] – установить режим защиты кода L1/L2/L3

P[+|-] – установить время power-on таймера Длинное/Короткое

P[0|1|2|3] – установить режим Power-on таймера Выкл/0.5мс/4мс/16мс (для ATmega03)

P[000|001|...|110|111] – выбрать параметры Power-on (для 2333/4433/mega161)

P[0000|0001|...|1110|1111] – выбрать параметры Power-on и Osc (ATtiny12/28/mega163)

F[+|-] – разрешить/запретить бит EESAVE (для ATmega03)

T[+|-] – разрешить/запретить режим BOOTRST (для ATmega16x)

Z[0|1|2|3] – установить размер Boot 1024/512/256/128 (для ATmega163)

A[0|1|2|3] – установить режим BLB0 BL3/BL4/BL2/BL1 (для ATmega16x)

W[0|1|2|3] – установить режим BLB1 BL3/BL4/BL2/BL1 (для ATmega16x)

M[+|-] – разрешить/запретить внешний RESET (для ATtiny1x)

M[+|-] – разрешить/запретить IntCAP (для ATtiny28)

B[+|-] – разрешить/запретить Brown-Out Detector

V[0|1] – установить напряжение Brown-Out 4.0V/2.7V (для 2333/4433/mega16x)

V[0|1] – установить напряжение Brown-Out 2.7V/1.8V (для ATtiny12)

## Конфигурация для AT89Cх051

C[1|2|3] – установить режим защиты кода программы L1/L2/L3

## Конфигурация для AT89C5х/AT87F5х

C[1|2|3|4] – установить режим защиты кода программы L1/L2/L3/L4

## Конфигурация для AT89Sxxx

C[1|2|3|4] – установить режим защиты кода программы L1/L2/L3/L4

## Конфигурация для AT89Sxxx (параллельный режим)

S[+|-] – разрешить/запретить режим последовательного программирования

**C[1|2|3|4]** - установить режим защиты кода программы L1/L2/L3/L4

## Конфигурация для ACExxx

**R** – выбрать тип тактового генератора Внешний RC  
**X** - выбрать тип тактового генератора XT  
**E** - выбрать режим внешнего тактового генератора  
**I** - выбрать тип тактового генератора Встроенный RC  
**F[+|-]** – разрешить/запретить UBD  
**T[+|-]** – разрешить/запретить LBD  
**B[+|-]** – разрешить/запретить Brown-Out Detector  
**W[+|-]** – разрешить/запретить сторожевой таймер  
**C[0|1|2|3]** – установить режим защиты Off/Rdis/Wdis/All

## Конфигурация для An15Exx:

**R** – выбрать тип тактового генератора Внешний RC  
**X** - выбрать тип тактового генератора XT  
**L** - выбрать тип тактового генератора LP  
**I** - выбрать тип тактового генератора Встроенный RC  
**P[+|-]** – разрешить/запретить таймер Power-on

## Конфигурация для 24LC65

**D<heb>** - установить высокопроизводительный блок в положение **<heb>** (0-15)  
**D<b0>/<bn>** - установить защиту записи на **<bn>** блоков, начиная с блока **<b0>**  
**D<heb>/<b0>/<bn>** - установить положение высокопроизводительного блока и защиту записи одной операцией

## Конфигурация для 25Cxxx

**W[+|-]** – разрешить/запретить вывод WPEN  
**C[0|1|2|3]** – установить режим защиты от записи 0/1/2/3

## Конфигурация для X250x7

**C[0|1|2|3|4|5|6|7]** – установить режим защиты Off/Q1/Q2/Q3/Q4/H1/P0/Pn

## Конфигурация для AT29Cxxx

**B[+|-]** – разрешить/запретить защиту Загрузочного блока

## Конфигурация для AT49Fxxx

**B[+|-]** – разрешить/запретить защиту обоих Загрузочных блоков  
**B[|h]** – разрешить/запретить защиту только нижнего/верхнего Загрузочного блока

## Глава 7. Обновление PicProg

### Политика обновлений и поддержки PicProg

**Телесистемы** постоянно работают над улучшением и расширением свойств программатора PicProg, а также добавляют поддержку новых устройств и семейств. Все обновления (программ и прошивок) являются бесплатными и всегда доступны со страницы поддержки программатора PicProg на Веб-сайте фирмы **Телесистемы**:

<http://www.telesys.ru/picprog/picpupgr.htm>

Вы можете также получать последнюю информацию об обновлениях PicProg по E-mail, подписавшись на лист рассылки **Телесистем**:

<http://www.telesys.ru>

При возникновении вопросов или предложений по программатору или необходимости сообщить об ошибке в программе или аппаратуре PicProg просьба связаться с **Телесистемами** по E-mail: [support@telesys.ru](mailto:support@telesys.ru)  
Поддержка пользователей по E-mail является бесплатной.

**Примечание:** Обращаясь по поводу технической поддержки, не забудьте указать версии программ и прошивки PicProg (см. [Номера версий](#)).

### Номера версий

#### Версия программного обеспечения PicProg для Windows

Чтобы увидеть версию программы, выберите пункт меню **Справка>О программе**.

#### Версия программного обеспечения PicProg для DOS

Запустите программу picprog.exe без параметров. Номер текущей версии программы отображается в правом верхнем углу экрана.

#### Версии аппаратуры и прошивки PicProg

Чтобы увидеть номер версии прошивки, установленной в программаторе, запустите PicProg для Windows при включенном в сеть программаторе. Вы увидите номера версий аппаратуры и прошивки в нижнем левом углу главного окна программы (см. [Начальные сведения](#)).

Номера версий представляются в следующем виде: **PicProg VX.Y[f]** (н-р, V3.0 или V3.4f), где **X** означает версию аппаратуры PicProg, а **Y** – версию прошивки программатора. Буква 'f' после номера версии означает наличие в программаторе флэш-процессора, который позволяет осуществлять обновление прошивки программатора внутрисхемно (см. [Внутрисхемное обновление прошивок](#)).

### Проверка версий на совместимость

Каждый раз при запуске программного обеспечения PicProg происходит проверка на совместимость с текущими версиями аппаратуры и прошивки. При выборе типа устройства в диалоге **Выберите устройство** иногда можно увидеть, что некоторые функциональные кнопки (такие как **Программирование** или **Чтение**) запрещены. Таким образом программа сообщает, что текущая версия прошивки не поддерживает данный тип устройства. В этом случае следует загрузить и установить обновленную версию прошивки для программатора (см. [Обновление прошивок](#)).

## Обновление прошивок

Если Вы обнаружили обновленную версию прошивки на странице поддержки PicProg (см. [Политика обновлений и поддержки PicProg](#)), следует загрузить файл **Ppfr\_XY.zip** (н-р, Ppfr\_34.zip), где **X** означает версию аппаратуры PicProg, к которой подходит данная прошивка, а **Y** - номер версии новой прошивки. Распакуйте содержимое файла на своем жестком диске и следуйте инструкциям, содержащимся внутри.

**Примечание:** Для новой прошивки Вам понадобится один чистый кристалл PIC16C67 или PIC16C77 в 40-выводном DIP-корпусе. Вы можете также использовать '67 или '77 JW-кристалл, чтобы иметь возможность обновлять прошивку несколько раз. В этом случае не выбрасывайте кристалл со старой прошивкой, так как он понадобится Вам для последующих обновлений.

**Примечание:** В качестве новой прошивки Вы можете также использовать кристалл PIC16F877 в 40-выводном DIP-корпусе. Этот вариант является предпочтительным, поскольку, записав в такой кристалл прошивку версии 3.4+ (или 2.4+) один раз, Вы получаете возможность в дальнейшем осуществлять обновление прошивки внутрисхемно. Но и в этом случае не выбрасывайте кристалл со старой прошивкой, так как всегда существует вероятность, хотя и небольшая, того, что прошивка Вашего флэш-процессора по каким-либо причинам окажется испорченной и Ваш старый кристалл понадобится Вам для ее восстановления.

## Внутрисхемное обновление прошивок

Если в Вашем программаторе установлен флэш-процессор (см. [Номера версий](#)), то его прошивка может обновляться внутрисхемно. Получив файл с новой прошивкой (см. [Обновление прошивок](#)) поместите его в каталог, в который было установлено ПО. Затем запустите PicProg для Windows, выберите закладку **Параметры** и нажмите кнопку **Обновить прошивку**. В открывшемся диалоге выберите файл с новой прошивкой и нажмите кнопку **Загрузить**. Программа автоматически определит возможность внутрисхемного обновления прошивки программатора и после запроса подтверждения от Вас выполнит обновление прошивки. В случае успешного обновления номер версии PicProg в строке состояния программы изменится на новое значение.

**Внимание:** Пожалуйста, не пытайтесь прервать процесс обновления прошивки! В противном случае прошивка может быть испорчена до такой степени, что для ее восстановления понадобится ее перепрограммирование с помощью другого программатора!

Если по какой-либо причине во время обновления прошивки произошел сбой, то при следующей попытке запуска программного обеспечения Вы увидите сообщение о возможных проблемах с прошивкой. В этом случае постарайтесь немедленно повторить процедуру обновления прошивки. В случае устойчивых проблем может потребоваться перепрограммирование прошивки внешним программатором (или Вашим же PicProg со старой OTP прошивкой).

## Приложение А. Адаптеры устройств PicProg

В данном приложении описана распейка дополнительных адаптеров устройств для всех версий PicProg.

В настоящее время разработаны следующие адаптеры:

**Адаптер 1** - PIC16C62/63/66/72/73/76/773 – **DIP28** (только для PicProg V1.x)

**Адаптер 2** - PIC14Cxxx – **DIP28** (только для V1.x)

**Адаптер 3** - PIC12Cxxx – **DIP8** (только для V1.x)

**Адаптер 4** - PIC16C92x – **PLCC68** (для всех версий PicProg)

**Адаптер 5** - PIC17C75x – **PLCC68** (только для V2.x и V3.x)

**Адаптер 6** - PIC16C50x – **DIP14** (только для V2.x)

**Адаптер 7** – MegaAVRx03 – **TQFP64** (только для V3.4+)

**Адаптер 8** - PIC17C76x – **PLCC84** (только для V2.x и V3.x)

**Адаптер 10** - ACExxxx – **SO8/14** (только для V3.4+)

**Адаптер 11** – AT90C8534 – **QFP48** (только для V3.2+)

**Адаптер 12** – KM29N040T – **TSOP40** (только для V3.2+ и V2.4+)

**Адаптер 13** – SST29LE(VE)xxx – **PLCC32** (только для V2.x и V3.x)

**Адаптер 14/DIP** – MCS51 – **DIP40** (только для Vx.4+)

**Адаптер 14/PLCC** – MCS51 – **PLCC44** (только для Vx.4+)

**Адаптер 15** – AT17Cxxx – **PLCC20** (только для Vx.6+)

**Примечание: NC** – Нет Соединения – выводы **должны** висеть в воздухе.

**Адаптер 1** - PIC16C62/63/66/72/73/76/773 – **DIP28** (только для V1.x)

Имя	RB6	RB7	Vdd	MCLR	NC	Vss
ZIF-колодка	22	23	25	1		31
Устройство	27	28	20	1	10	Все остальные

**Адаптер 2** - PIC14Cxxx – **DIP28** (только для V1.x)

Имя	RC6	RC7	Vdd	MCLR	NC	Vss
ZIF-колодка	22	23	25	1		31
Устройство	12	11	9	14	7,10,21,22	Все остальные

**Адаптер 3** - PIC12Cxxx – **DIP8** (только для V1.x)

Имя	GP1	GP0	Vdd	MCLR	NC	Vss
ZIF-колодка	22	23	25	1		31
Устройство	6	7	1	4	3	Все остальные

**Адаптер 4** - PIC16C92x – **PLCC68** (для всех версий PicProg)

Имя	RB6	RB7	Vdd	MCLR	NC	Vss
ZIF-колодка	22	23	32 *	1		31
Устройство	65	66	21,22,64	2	25,63	7,23,24

\* – к выводу **25** ZIF-колодки для V1.x

# Руководство пользователя PICPROG

## Адаптер 5 - PIC17C75x – PLCC68 (только для V2.x и V3.x)

Имя	NC	Vdd	Vss	Test	ra0	ra1	ra2	ra3	ra4	Osc1	MCLR
ZIF-колотка		1	31	27	26	25	24	23	22	19	32
Устр-во	12,13, 14,51	2,20,29, 37,49	19,30,36, 53,68	17	60	44	45	46	43	50	16

## Адаптер 5 - продолжение

Имя	rc0	rc1	rc2	rc3	rc4	rc5	rc6	rc7	rb0	rb1	rb2	rb3	rb4	rb5	rb6	rb7
ZIF-колот.	2	3	4	5	6	7	8	9	11	12	13	14	15	16	17	18
Уст-во	3	67	66	65	64	63	62	61	59	58	54	57	56	55	47	48

## Адаптер 6 - PIC16C50x – DIP14 (только для V2.x)

Имя	RB1	RB0	Vdd	MCLR	NC	Vss
ZIF-колотка	22	23	32	1		31
Устройство	12	13	1	4	3	Все остальные

## Адаптер 7 – MegaAVRx03 – TQFP64 (только для V3.4+)

Имя	NC	VCC	GND	Reset	Xtal1	Pd1	Pd2	Pd3	Pd4	Pd5	Pd6
ZIF-колотка		30	20	11	15	13	16	17	18	19	21
Устр-во	23,33, 34,48	21,52, 62,64	22,53,63	20	24	26	27	28	29	30	31

## Адаптер 7 – продолжение

Имя	Pd7	Pa0	Pb0	Pb1	Pb2	Pb3	Pb4	Pb5	Pb6	Pb7
ZIF-колотка	20	12	22	23	24	25	26	27	28	29
Устр-во	32	51	10	11	12	13	14	15	16	17

## Адаптер 8 - PIC17C76x – PLCC84 (только для V2.x и V3.x)

Имя	NC	Vdd	Vss	Test	ra0	ra1	ra2	ra3	ra4	Osc1	MCLR
ZIF-колотка		1	31	27	26	25	24	23	22	19	32
Устр-во	16,17,18, 63	2,24,37, 45,61	23,38,44, 65,84	21	72	56	57	58	51	62	20

## Адаптер 8 - продолжение

Имя	rc0	rc1	rc2	rc3	rc4	rc5	rc6	rc7	Rb0	rb1	rb2	rb3	rb4	rb5	rb6	rb7
ZIF-колот.	2	3	4	5	6	7	8	9	11	12	13	14	15	16	17	18
Уст-во	3	83	82	81	80	79	78	77	71	70	66	69	68	67	59	60

## Адаптер 10 - ACExxxx – SO8/14 (только для V3.4+)

Имя	LOAD	Vdd	Vss	CKI	SFT_I	SFT_O	G5
ZIF-колотка	17	24	23	21	18	22	19
8-ног	1	8	7	5	2	6	3
14-ног	1	14	13	8	2	11	6

## Адаптер 11 – AT90C8534 – QFP48 (только для V3.2+)

Имя	Osc	Vcc	Gnd	Rdy	Oe	Wr	Bs	Xa0	Xa1	Res	d0	d1	d2	d3	d4	d5	d6	d7
ZIF-колот.	15	30	20	13	16	17	18	19	21	11	22	23	24	25	26	27	28	29
Уст-во	24	22, 18	31, 11,1	34	13	14	15	16	17	20	47	46	45	44	39	38	33	35

# Руководство пользователя PICPROG

## Адаптер 12 – KM29N040T – TSOP40 (только для V3.2+ и V2.4+)

Имя	CE	Vcc	GND	Cle	Ale	We	Wp	Re	Sp	d0	d1	d2	d3	d4	d5	d6	d7
ZIF-колод.	26	36	20	33	34	35	6	28	32	17	18	19	21	22	23	24	25
Уст-во	39	21, 40	1,20	2	3	4	5	38	36	16	17	18	19	22	23	24	25

## Адаптер 13 – SST29LE(VE)xxx – PLCC32 (только для V2.x & V3.x)

Выводы микросхем в корпусе **PLCC32** полностью соответствуют выводам аналогичных микросхем в корпусе **DIP32**.

## Адаптер 14/DIP – MCS51 – DIP40 (только для Vx.4+)

Имя	P10	P11	P12	P13	P14	P15	P16	P17	P20	P21	P22	P23	P24	P25	P26	P27
ZIF-колод.	16	15	14	13	12	11	10	9	31	30	27	29	8	32	6	34
Уст-во	1	2	3	4	5	6	7	8	21	22	23	24	25	26	27	28

## Адаптер 14/DIP – продолжение

Имя	P00	P01	P02	P03	P04	P05	P06	P07	P30	P31	P32	P33	P34	P35	P36	P37
ZIF-колод.	17	18	3 *	21	22	23	24	25	7	37	37	7	4	37	35	36
Уст-во	39	38	37	36	35	34	33	32	10	11	12	13	14	15	16	17

## Адаптер 14/DIP – продолжение

Имя	RST	XTAL1	PSEN	ALE	EA	VCC	GND	NC
ZIF-колод.	26	19	37	28	20	26	37	
Уст-во	9	19	29	30	31	40	20	18

\* – к выводу 1 ZIF-колотки для V2.4

## Адаптер 14/PLCC – MCS51 – PLCC44 (только для Vx.4+)

Имя	P10	P11	P12	P13	P14	P15	P16	P17	P20	P21	P22	P23	P24	P25	P26	P27
ZIF-колод.	16	15	14	13	12	11	10	9	31	30	27	29	8	32	6	34
Уст-во	2	3	4	5	6	7	8	9	24	25	26	27	28	29	30	31

## Адаптер 14/PLCC – продолжение

Имя	P00	P01	P02	P03	P04	P05	P06	P07	P30	P31	P32	P33	P34	P35	P36	P37
ZIF-колод.	17	18	3 *	21	22	23	24	25	7	37	37	7	4	37	35	36
Уст-во	43	42	41	40	39	38	37	36	11	13	14	15	16	17	18	19

## Адаптер 14/PLCC – продолжение

Имя	RST	XTAL1	PSEN	ALE	EA	VCC	GND	NC
ZIF-колод.	26	19	37	28	20	26	37	
Уст-во	10	21	32	33	35	44	22	20

\* – к выводу 1 ZIF-колотки для V2.4

## Адаптер 15 – AT17Cxxx – PLCC20 (только для Vx.6+)

Имя	Dat	Clk	Reset	CE	A2	SerEn	Vcc	Gnd	NC
ZIF-колотка	17	18	19	20	22	23	24	21	
Устройство	2	4	6,9	8	12,14	17,18	20	5,7,10	15

## Приложение В. Внутрисхемное программирование с помощью PicProg

В данном приложении описана распылка кабелей Внутрисхемного Программирования через 7-выводной ISP-разъем на PicProg V3.

Внутрисхемное Программирование возможно для следующих типов устройств:

**Примечание: NC** – Нет Соединения – при программировании устройств через ISP-разъем, но не в схеме, данные выводы **должны** висеть в воздухе.

### PIC12Cxxx – DIP8

Имя	MCLR	Vdd	Vss	GP1	GP0	NC
ISP-разъем	MCLR	VDD	GND	SCK	SDO	
Устройство	4	1	2,8	6	7	3

### PIC14Cxxx – DIP28

Имя	MCLR	Vdd	Vss	RC6	RC7	NC
ISP-разъем	MCLR	VDD	GND	SCK	SDO	
Устройство	14	9	8,20	12	11	7,10,21,22

### PIC16C50x – DIP14

Имя	MCLR	Vdd	Vss	RB1	RB0	NC
ISP-разъем	MCLR	VDD	GND	SCK	SDO	
Устройство	4	1	2,14	12	13	3

### PIC16Cxxx, PIC16Fxxx, PIC18Cxxx – DIP

Имя	MCLR	Vdd	Vss	RB6	RB7	NC
ISP-Разъем	MCLR	VDD	GND	SCK	SDO	
18-ног	4	14	5,16	12	13	15
28-ног	1	20	8,9	27	28	10
40-ног	1	11,32	12,13,31	39	40	14

### PIC16C92x – PLCC68

Имя	MCLR	Vdd	Vss	RB6	RB7	NC
ISP-разъем	MCLR	VDD	GND	SCK	SDO	
Устройство	2	21,22,64	7,23,24	65	66	25,63

### PIC17C7xx – PLCC

Имя	MCLR	Vdd	Vss	RA5	RA4	RA1	TEST	NC
ISP-разъем	MCLR	VDD	GND	SCK	SDO	SDI	OSC	
68-ног	16	2,20,29, 37,49	19,30,36, 53,68	42	43	44	17	51
84-ног	20	2,24,37, 45,61	23,38,44, 65,84	50	51	56	21	63

### HCS512 – DIP18

Имя	MCLR	VDD	GND	CLK	DATA	OSCIN	OSCOUT
ISP-разъем	MCLR	VDD	GND	SCK	SDO	SDI	OSC
Устройство	4	14	5	12	13	16	15

# Руководство пользователя PICPROG

## AVR – DIP

Имя	RESET	VCC	GND	SCK	MOSI	MISO	XTAL1	NC
ISP-разъем	MCLR	VDD	GND	SCK	SDO	SDI	OSC	
8-ног	1	8	4	7	5	6	2	3
8-ног (tiny15)	1	8	4	7	5	6	3	2
20-ног	1	20	10	19	17	18	5	4
28-ног	1	7,20,21	8,22	19	17	18	9	10
40-ног (8515/m161)	9	40	20	8	6	7	19	16,17, 19,29,30
40-ног (АЦП) (8535/m163)	9	10, 30,32	11,31	8	6	7	13	12

## МегаAVRx03 – TQFP64

Имя	RESET	VCC	GND	PB1	PE0	PE1	XTAL1	NC
ISP-разъем	MCLR	VDD	GND	SCK	SDO	SDI	OSC	
Устройство	20	21, 52, 64	22, 53, 63	11	2	3	24	23

## AT89S8252/S53 – DIP40

Имя	RST	VCC	GND	SCK	MOSI	MISO	XTAL1	NC
ISP-разъем	MCLR	VDD	GND	SCK	SDO	SDI	OSC	
Устройство	9	40	20	8	6	7	19	16,17,18,30

## TC58A040F – SOP28

Имя	CS	Vcc	GND	SO	SI	SCK
ISP-разъем	OSC	VDD	GND	SDO	SDI	SCK
Устройство	4	28	1	7	6	5

## ACE – SO

Имя	LOAD	Vdd	Vss	CKI	SFT_I	SFT_O	NC
ISP-Разъем	MCLR	VDD	GND	SCK	SDI	SDO	
8-ног	1	8	7	5	2	6	3
14-ног	1	14	13	8	2	11	6

## SX18/20/28 – DIP/SOIC/SSOP

Имя	OSC1	Vdd	Vss	OSC2
ISP-Разъем	MCLR	VDD	GND	SCK
18-ног(DIP/SOIC)	16	14,4	5	15
20-ног(SSOP)	18	15,16,4	5,6	17
28-ног(DIP/SOIC)	27	2,28	4	26
28-ног(SSOP)	27	3,4,28	1,14	26

**Примечание:** Scepix настоятельно рекомендует включать дополнительный резистор 100 Ом между выводами MCLR (ISP-разъем) и OSC1 (SX) при программировании кристаллов старых выпусков.

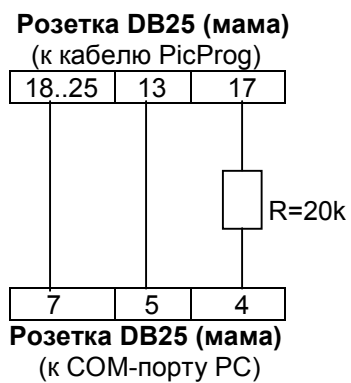
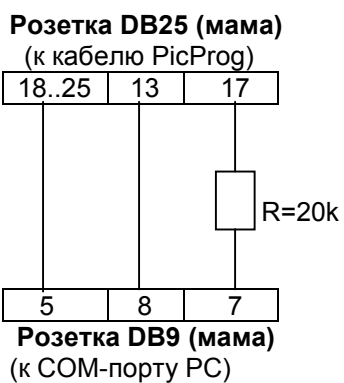
## SX48/52 – QFP

Имя	OSC1	Vdd	Vss	OSC2
ISP-Разъем	MCLR	VDD	GND	SCK
48-ног	2	4,18,32,46,1	5,19,33,47	3
52-ног	4	6,20,34,48,3	7,21,35,49	5

**Примечание:** Scepix настоятельно рекомендует включать дополнительный резистор 100 Ом между выводами MCLR (ISP-разъем) и OSC1 (SX) при программировании кристаллов старых выпусков.

## Приложение С. Схема адаптера для подключения PicProg к COM-порту компьютера

---



## **Приложение D. История модернизации PicProg для Windows**

---

Поскольку PicProg для Windows находится на стадии Бета-тестирования, история модернизации для него не приводится.

## Приложение Е. История модернизации PicProg для DOS

---

### V4.16 (23 Июня 97)

- доработана программа верификации (теперь операции проверки на чистоту и верификации прерываются после обнаружения первой же ошибки)
- появилась возможность информацию о записываемых микросхемах накапливать в файле picprog.log (это накопление включено по умолчанию и может быть отключено ключом -sr)
- в ответ на запрос программы о необходимости установки нестираемой защиты появилась возможность ответов 'a' (т.е. "да для всех") и 's' (т.е. "нет для всех")

### V4.17 (15 Авг 97)

- исправлена ошибка, в результате которой было невозможно осуществлять внутрисхемное программирование PIC16C/F8x

### V4.18 (25 Авг 97)

- исправлена ошибка программирования PIC16C64x/66x/715; к сожалению, программирование этих устройств требует замены внутренней прошивки программатора на версию не менее V2.2
- исправлена ошибка, в результате которой при стирании PIC16C/F8x (если в них не была установлена защита) не стиралась область данных
- в списке устройств появилась категория "на основании предварительной информации" (т.е. алгоритмы работы с этими устройствами недостаточно проверены на реальных кристаллах)
- введена возможность программирования битов защиты и положения high-endurance блока в EEPROM 24xx65 (с помощью ключа -fd)
- on-line help вынесен в отдельные файлы picprog.hle(англ.) и picprog.hlr(рус.)

### V4.19 (2 Сен 97)

мелкие доработки для PICPROG V3.x

- переработан формат ini-файла на Windows-подобный;
- теперь выбор языка интерфейса осуществляется в ini-файле, в связи с чем исчезла команда "l"

### V4.20 (3 Окт 97)

- доработаны алгоритмы работы с PIC16C64x/66x/715
- введен режим программирования, не требующий подтверждения установки микросхемы и окончания программирования с клавиатуры (с помощью ключа -sk)

### V4.21 (16 Окт 97)

- изменен порядок программирования положения high-endurance блока в 24xx65 (перед программированием основных данных)

### V4.22 (21 Окт 97)

- исправлена ошибка, в результате которой в некоторых случаях устройства, не полностью прошедшие верификацию при программировании, в log-файле считались записанными правильно

### V4.23 (11 Ноя 97)

- добавлена возможность устанавливать стираемую защиту в PIC16C710/711
- добавлена поддержка PIC12CE518/519
- при проверке на чистоту OTP кристаллов PIC16C5x учитывается возможность наличия в них заводской установки типа генератора
- добавлена возможность вывода информации об ошибках программирования в файл (командой pr)

### V4.24 (14 Янв 98)

# Руководство пользователя PICPROG

---

- в log-файле теперь запоминается и время окончания программирования
- изменено сообщение о возможности дописать информацию при программировании: вариант ответа "Ignore" заменен на "Continue"
- исправлена ошибка, возникающая при сравнении некоторых типов файлов с правильно записанными кристаллами PIC16C64x/66x/715
- несколько изменен формат ini-файла

## V4.24a (22 Янв 98)

- несколько изменен порядок стирания битов защиты в PIC16F8x (полностью работает только с внутренней прошивкой программатора V2.3)

## V4.24b (20 Фев 98)

- исправлена ошибка внутрисхемного программирования PIC12/16 (режим не включался при операциях чтения/верификации/проверки на чистоту)
- добавлена поддержка PIC12CE673/674

## V4.25 (31 Мар 98)

- изменена обработка битов контроля по четности в словах конфигурации PIC16C64x/66x/715
- добавлена поддержка PIC16C554A/556A/558A/620A/621A/622A, PIC16CE623/624/625, PIC16C54B/56A/58B
- on-line help содержит теперь список поддерживаемых устройств

## V4.26 (5 Июня 98)

- опять изменена обработка битов контроля по четности в словах конфигурации PIC16C64x/66x/715
- исправлена процедура записи 27Cxxx (особенно 27C512)
- изменена процедура стирания 28Cxx

## V4.27 (5 Июля 98)

- в соответствии с последними рекомендациями Microchip изменены алгоритмы программирования PIC12C5xx, PIC16C5x (также изменился формат ini-файла)
- добавлена возможность стирания партии устройств за один сеанс

## V5.00 (29 Окт 98)

- новая версия приурочена к выпуску PICPROG V3.x и V2.4

*Добавлена поддержка следующих устройств:*

- PIC16C55A/57C/62B/63A/65B/72A/73B/773/74B/774/84A/508A/509A
- 24LC025/32A, 93LC46B/56B/66B
- PIC16C505, PIC17C756A/762/766 (V2.0+)
- 24LC00/21, 24LCS21, 93LC46A/56A/66A, 93LCS56/66 (V2.4+)
- 25LC01/020/040/080/160/320/640 (V2.4+)
- HCS512 (V3.0+)
- AT24C01/01A/21/02/02A/04/04A/08/08A/16/16A/32/64/128/256/512, AT34C02
- AT93C46/46A/46B/46C/56/57/66, AT28C16/17/64/64X
- AT25010/020/040/080/160/320/640/128/256, AT25HP256/512 (V2.4+)
- AT28C256/010/040, AT28BV16/64, AT28HC64B/256, AT28LV64B/256/010 (V2.4+)
- AT29C256/512/010A/020/040/040A, AT49F512/010/020/040/040T, AT49HF010 (V2.4+)
- AVR 1200/2313/4414/8515/2323/2343/4434/8535, Mega603/103 (V3.0+)
- AT89C1051/1051U/2051/4051, AT89(L)S8252/53 (V3.0+)
- SX18AC/28AC (V2.4+)
- добавлена поддержка режима внутрисхемного программирования PIC17C7xx (V3.0+)
- время программирования 24LCxxx заметно сократилось при наличии V2.4+
- добавлена проверка идентификатора устройства (для V2.4+), можно отключать опцией -si
- введен режим принудительного стирания устройств на основе EEPROM перед программированием опцией -se (верно и в автономном режиме для V2.4+), может

# Руководство пользователя PICPROG

---

быть полезно для устройств, не стертых в исходном состоянии (например, PIC16F8х)

- звуковой сигнал по окончании программирования может теперь выдаваться при условии, что процесс программирования длился дольше заданного времени, определяемого опцией `-s<ts>`
- кнопка START на PICPROG может быть задействована для инициации программирования в обычном режиме (для V2.4+, опция `-sc`)
- появился новый режим: команда `e` (стирание) для УФ-стираемых устройств инициирует процесс периодической проверки устройства на чистоту при минимальном Vdd, т.е. теперь можно установить кристалл в PICPROG, разместить над ним УФ-лампу и дожидаться достаточной степени стирания кристалла; проверка на чистоту осуществляется периодически с интервалом, задаваемым опцией `-st<tc>` (по умолчанию 60сек), по окончании процесса выдается звуковой сигнал
- исправлена ошибка чтения 27C010 в бинарный файл
- разрешено задание частичной защиты кода программы из командной строки для PIC-контроллеров (`-fc5/7`)
- добавлен ряд битов конфигурации для новых типов устройств
- применен более сложный алгоритм определения необходимости стирания устройства перед программированием
- добавлена поддержка расширенных форматов Intel hex как с линейной, так и с сегментной адресацией
- при загрузке hex-файлов, созданных MPASM, PicProg проверяет наличие соответствующего .cod файла и загружает из него тип устройства
- добавлена возможность одновременной загрузки двух файлов (одного с программным кодом, а второго - с данными EEPROM) для AVR и AT89S; вызывается строкой вида `<файл1>+<файл2>` или `+<файл2>`
- изменена форма ряда сообщений с целью совместимости с версией Win32

## V5.01 (26 Ноя 98)

- добавлена поддержка форматов входных файлов, генерируемых ассемблерами SPASM фирмы Parallax и CVASM фирмы TechTools
- добавлена поддержка форматов входных файлов Motorola S и Tektronix Hex
- изменена форма ряда сообщений и внешний вид режима УФ-стирания для совместимости с версией Win32
- исправлен вывод информации о конфигурации для ATmega103L и AT90S1200A
- исправлен ввод из командной строки битов защиты кода для PIC16C5xB

## V5.02 (30 Ноя 98)

*Исправлен ряд ошибок, появившихся в версиях 5.00 и 5.01*

--- **ЗАМЕНИТЕ ЭТИ ВЕРСИИ НА ТЕКУЩУЮ, ПОЖАЛУЙСТА** ---

- исправлена ошибка записи микросхем параллельной памяти xxx020 и xxx040
- исправлена ошибка неверной проверки на чистоту ID в PIC16C5x
- уменьшены требования к объему свободной памяти DOS до 490кбайт

## V5.03 (15 Дек 98)

- доработаны алгоритмы записи/стирания AT29Cxxx/AT49Fxxx
- в связи с обнаруженной ошибкой записи HCSxxx в версии 3.0 программа предложит обновить эту версию до V3.1

## V5.03a (17 Дек 98)

*Исправлены ошибки, появившиеся в версиях 5.02 и 5.03*

- исправлена ошибка записи/чтения AVR и AT89S8252
- исправлена ошибка, приводящая к неверному определению типа устройства при загрузке файла для EEPROM AVR

## V5.04 (12 Янв 99)

- исправлена ошибка верификации HCS512
- различные исправления для совместимости с программой Win32 Beta2

# Руководство пользователя PICPROG

---

- правильно отражается тот факт, что для HCS512 и ATmega возможна работа только через ISP-разъем
- разрешена возможность ISP-программирования 17C7xx, запрещенная по ошибке в V5.00-5.03a
- список поддерживаемых устройств вынесен из on-line help в отдельный файл devicer.txt

## V5.05 (10 Мая 99)

- исправлена ошибка программирования 24C02
- исправлена ошибка программирования AT29C020
- исправлена ошибка программирования 27C64/128
- исправлены ошибочные ID устройств для AT90S8535/4434
- алгоритмы программирования AVR настроены согласно последней спецификации ATMEL
- исправлена ошибка, когда при неверной верификации после программирования только на макс. Vdd устройство считалось правильно запрограммированным
- исправлена ошибка, когда во время верификации или проверки на чистоту значения fuses при определенных условиях отображались неверно
- явно запрещены операции чтения, проверки и верификации для HCSxxx
- добавлена поддержка PIC16F87x
- добавлена поддержка Winbond W29Cxxx и W29Exxx
- добавлена поддержка SST29Exxx
- добавлена поддержка ST24/25xxxx, M24xxxx, ST93xxxx, M93xxxx, ST95xxxx, M95xxxx от ST Microelectronics
- добавлена поддержка AT90S4433/2333, ATtiny10/11/12/22, AT90C8534 (требуется обновления прошивки до V3.2)
- добавлена поддержка микроконтроллеров An15Exx от Ангстрема (требуется обновления прошивки до V3.2)
- добавлена поддержка Toshiba TC58A040F (требуется обновления прошивки до V3.2)
- добавлена поддержка Samsung KM29N040T (требуется обновления прошивки до V3.2)
- с этой версии все подтипы (C,LC,AA,FC) последовательных EEPROM Microchip будут присутствовать в списке устройств для корректной верификации при предельных значениях напряжений питания

## V5.05a (30 Мая 99)

- существенно улучшена процедура снятия защиты кода с PIC16x8x
- подробные описания битов конфигурации удалены из on-line help-a

## V5.05m (13 Июня 99)

- добавлена поддержка AT28C16E/17E/64E
- добавлена поддержка PIC16C712/716
- добавлена полная поддержка Intel 28F001BX (требуется обновления прошивки до V3.3)
- добавлена поддержка Am29Fxxx и Am28Fxxx(A) (требуется обновления прошивки до V3.3)
- добавлена поддержка Macronix MX28Fxxx (требуется обновления прошивки до V3.3)
- добавлена поддержка STMicro M29Fxxx и M28Fxxx (требуется обновления прошивки до V3.3)
- добавлена поддержка NexFlash NX29Fxxx
- добавлена поддержка AT49F00x
- изменена процедура стирания AT28BV16/64
- улучшены процедуры стирания и записи AT49Fxxx
- расширены возможности процедуры проверки ID устройства

## V5.05r (3 Окт 99)

- исправлена ошибка записи 1878BE1 (ошибка появилась только в V5.05m)
- звуковые сигналы всегда идут через динамик компьютера

# Руководство пользователя PICPROG

---

## V5.06 (12 Дек 99)

- добавлена полная поддержка SX18/20/28/48/52, включая старые и новые версии кристаллов и внутрисхемное программирование (требует прошивки 3.4)
- добавлена возможность внутрисхемного обновления прошивки программатора при условии установки в нем флэш-процессора (требует прошивки 3.4 или 2.4)
- добавлена поддержка AT89C5x (требует прошивки 3.4 или 2.4)
- добавлена возможность снятия битов защиты для некоторых старых версий кристаллов AT89Cх051 и AT89C5x (требует прошивки 3.4)
- добавлена поддержка микроконтроллеров ACExxxx (требует прошивки 3.4)
- добавлена поддержка AT24C01, X24C01P и аналогичных (требует прошивки 3.4 или 2.4)
- добавлена поддержка X25Cxx и X25xxx (X25C требует прошивки 3.4 или 2.4)
- добавлена возможность определения и модификации алгоритмов программирования семейства 27Cxxx через ini-файл; внесены два предустановленных алгоритма – QuickPulse и Intelligent
- теперь все сообщения программы выводятся не в stdout, а на console
- ответы с клавиатуры обрабатываются верно, даже если включена русская раскладка клавиатуры
- добавлена поддержка FRAM фирмы Ramtron, включая FM24Cxxx, FM1x08 и FM25xxxx (1x08 и 25160 требуют прошивки 3.4 или 2.4)
- добавлена возможность работы PicProg еще и через COM-порты компьютера (требует дополнительного адаптера); в связи с этим опция командной строки – `<port>` заменена на `-pl<port>`, `-pc<port>`, `-pa`; кроме того, добавлена возможность устанавливать номер порта для связи или автопоиска непосредственно в ini-файле
- добавлена поддержка SST29LE(VE)xxx
- исправлена ошибка, приводящая к верификации при неверных напряжениях питания некоторых параллельных микросхем Flash-памяти, имеющих биты защиты загрузочного блока
- добавлена возможность программирования битов защиты загрузочного блока для AT49F001(N)T
- проверены размеры страниц и биты конфигурации для всех SPI EEPROM
- при загрузке файлов в форматах IntelHex и MotorolaS наличие пустых строк в начале файла больше не считается ошибкой
- с целью полной совместимости с 2000 годом год в log-файле и информации автономного режима теперь обозначается 4мя цифрами
- добавлена поддержка AT89Sxxxx в параллельном режиме (требует прошивки 3.4 или 2.4)
- добавлена возможность программирования ATmega103/603 в параллельном режиме (требует прошивки 3.4)
- добавлены типы ATmega103/Old(603/Old) для корректной поддержки первых выпусков данных устройств
- доработаны процедуры стирания AVR в соответствии с последними требованиями спецификации программирования от ATMEL
- отдельная процедура верификации сделана более ясно – теперь она всегда проверяет только выбранные области устройства и не пытается проверять устройство на чистоту (используйте для этого процедуру Проверки на чистоту)
- исправлена ошибка чтения/записи калибровочного значения для PIC16C505, появившаяся в версии 5.05
- при попытке задать в командной строке алгоритм программирования, не определенный для заданного типа устройства (например, `-al` для PIC16F84) будет выдано сообщение об ошибке

## V5.06a (23 Янв 00)

- исправлена ошибка, приводящая к инверсии старшего разряда адреса при чтении/записи AT28C256 и FM18(L)08
- добавлен режим автоматического накопления служебной информации в случае аварийной потери связи с программатором в файле `picprog.rpt` для последующего анализа у производителя

# Руководство пользователя PICPROG

---

## V5.07 (31 Июля 00)

- добавлена поддержка PIC18Cxxx (требуется прошивка 3.5 или 2.5)
- добавлена поддержка PIC16F870/871/872
- добавлена поддержка PIC16C745/765
- добавлена поддержка Intel 87C5x/51Rx/51Fx
- добавлена поддержка Philips P87C5x/51Fx/51Rx+
- добавлена поддержка Xicor X25642/128/160/170
- добавлена поддержка SST39xFxxx(P)
- добавлена поддержка Mosel Vitelic V(F)29(L)C5100x/3100x(T,B)
- добавлена поддержка STMicro M28Cxxx
- добавлена поддержка AT28C64B
- добавлена поддержка HCS101/201 (требуется прошивка 3.5 или 2.5)
- добавлена поддержка HCS 410/412/320
- исправлена ошибка, приводившая к неработоспособности ATmega103 при попытке ее стирания
- исправлена ошибка программирования HCS360/361 (требуется прошивка 3.5 или 2.5)
- исправлена ошибка стирания AT49F00x
- реализован способ обхода ошибки в некоторых партиях AT90S4433/2333, приводящей к невозможности одновременного программирования битов конфигурации BOR и CP
- сделаны некоторые изменения в отображении состояния битов защиты области данных в PIC14C000/16F87x
- названия и значения битов конфигурации SX48/52 (промышленная версия) приведены в соответствие с последней версией спецификации Scenix
- исправлена ошибка загрузки HEX-файлов размером 512 кБ

## V5.08 (31 Мар 01)

- добавлена поддержка PIC16F73/74/76/77 (требуется прошивка x.6)
- добавлена поддержка PIC16F627/628
- добавлена поддержка PIC16C717
- добавлена поддержка PIC16C432
- добавлена поддержка AT17C(LV)65/128/256/512/010/020(A) (требуется прошивка x.6)
- добавлена поддержка AT87F(LV)55/55WD/51RC (требуется прошивка x.6)
- добавлена поддержка ATtiny28L/28V
- добавлена поддержка ATtiny15L (требуется прошивка 3.6)
- добавлена поддержка ATmega163(L)/161(L) (требуется прошивка 3.6)
- добавлена поддержка чтения байта калибровки ATtiny&ATmega
- скорректирован DeviceID для PIC16F84A
- скорректирован DeviceID для PIC18C442
- исправлена ошибка, возникающая при одновременном задании битов конфигурации в командной строке и входном файле
- исправлена ошибка, приводящая к появлению неверного сообщения об ошибке записи битов конфигурации PIC18 при задании конфигурации в файле
- исправлена ошибка программирования битов защиты ATtiny12 в высоковольтном режиме (требуется прошивка 3.6)
- скорректированы биты конфигурации PIC18 в соответствии с последней спецификацией производителя (удален бит LVEp, инвертирован бит CCP2MX)
- теперь ПО автоматически заканчивает работу, если не находит PicProg в "бескнопочном" режиме программирования
- реализована возможность загружать из командной строки одновременно файлы программы и данных EEPROM для PIC, AVR, ACE, указывая имена файлов через знак ':', а именно: **<имя\_файла\_программы>:<имя\_файла\_данных\_EEPROM>**. Реализация данной функции через знак '+' также оставлена для совместимости с предыдущими версиями.

## Приложение F. История модернизации прошивок PicProg

---

### V1 – первая аппаратная версия PicProg

Выпущены следующие версии прошивок:

#### V1.0

- начальная версия

#### V1.1

- исправлена ошибка верификации HCS3xxx
- несколько оптимизирован алгоритм программирования I2C EEPROM
- добавлена функция определения версии прошивки (требует замены программы)

#### V1.2

- исправлена ошибка программирования конфигурации PIC12Cxxx (при программировании IN и MCLR одновременно PicProg не мог больше читать устройство)

### V2 – вторая аппаратная версия PicProg

Выпущены следующие версии прошивок (**НЕ СОВМЕСТИМЫ** с PicProg V1)

#### V2.0

- начальная версия (почти все кристаллы программируются теперь без адаптеров; добавлена поддержка ряда параллельных EPROM и EEPROM; диапазон и точность перестройки Vdd и Vpp заметно улучшены; добавлен автономный режим; добавлен разъем внутрисхемного программирования)

#### V2.1

- исправлены две ошибки в аппаратуре (требуется доработка платы)
- исправлена ошибка программирования конфигурации PIC12Cxxx (при программировании IN и MCLR одновременно PicProg не мог больше читать устройство)

#### V2.2

- исправлена ошибка программирования PIC с контролем по четности (типа PIC16C64x/66x и PIC16C715) (требует обновления программы)
- сделано исправление, позволяющее спасти кристаллы PIC16CxxA/JW, если они по ошибке программируются как PIC16Cxx

#### V2.3

- доработан алгоритм снятия защиты PIC16F8x

#### V2.4

упрощенная версия V3.4, требует PIC16C67/77 или PIC16F877

- добавлена поддержка внутрисхемного обновления прошивки программатора при условии установки в PicProg кристалла 16F877 вместо 16C77/67
- доработан автономный режим (появились возможности стирания устройств и проверки их идентификатора)
- появилась возможность проверять состояние кнопки START
- ускорена процедура программирования побайтных микросхем Flash памяти
- увеличено время ожидания при связи с PC для максимально надежной работы под Win32 на медленных компьютерах
- доработана процедура снятия защиты с PIC16x8x
- исправлены ошибки в алгоритмах для HCS36x, 24LC21x

# **Руководство пользователя PICPROG**

---

- добавлена поддержка 8-битовых SEEPROM (24LC00, 93LCxxA)
- поддержка записи программной защиты для SEEPROM (24LC65, 24LCSxx, 93LCSxx)
- режим страничного программирования для 24LCxxx
- поддержка AT28Cxxx, AT29Cxxx, AT49Fxxx, 25Cxxx
- добавлена полная поддержка Intel 28F001BX
- добавлена поддержка Macronix MX28xxxx
- добавлена поддержка AMD Am28Fxxx(A)
- добавлена поддержка NAND-flash KM29xxxx
- добавлена поддержка контроллеров An15Exx (Ангстрем)
- добавлена поддержка семейства AT89C5x (требует дополнительного адаптера)
- добавлена поддержка программирования семейства AT89Sxxxx в параллельном режиме
- добавлена поддержка микросхем AT24C01, X24C01P и аналогичных
- добавлена поддержка SPI EEPROM без регистра состояния (типа X25C02)
- добавлена возможность корректного чтения параллельных кристаллов памяти с защелкой адреса (типа FM1608/1808)
- добавлена поддержка микросхем FM25160/25L160

## V2.5

- добавлена поддержка PIC18Cxxx
- исправлена ошибка программирования HCS360/361
- добавлена поддержка HCS101/201
- сделаны некоторые изменения в алгоритмах программирования ATMEL AVR для лучшей совместимости с рядом будущих кристаллов

## V2.6

- добавлена поддержка PIC16F7x
- добавлена поддержка AT17Cxxx
- добавлена поддержка DevID новых AT89C(87F)5x
- добавлена поддержка FM24Cxxx в 20-выводном корпусе
- модифицирован алгоритм записи AT89C5x для лучшей совместимости с определенными партиями микросхем

## **V3 – третья аппаратная версия PicProg**

*Выпущены следующие версии прошивок (**НЕ СОВМЕСТИМЫ** с PicProg V1 и V2)*

## V3.0

- начальная версия (несколько увеличен объем встроенного буфера Flash)
- улучшен автономный режим (теперь может стирать и проверять ID устройства)
- добавлена возможность постоянного опроса кнопки START
- исправлены ошибка программирования HCS36x, 24LC21x
- добавлена поддержка 8-разрядных SEEPROM (24LC00, 93LCxxA)
- добавлена поддержка программной защиты в SEEPROM (24LC65, 24LCSxx, 93LCSxx) и режим страничной записи 24LCxxx
- добавлена поддержка AVR, AT89Cх051, AT89Sxxxx, HCS512, SX, AT28Cxxx, AT29Cxxx, AT49Fxxx, 25Cxxx, 16C50x
- добавлена поддержка режима последовательного программирования PIC17C7xx

## V3.1

- исправлена ошибка программирования HCSxxx (только для V3.0)

## V3.2

- добавлена поддержка NAND-flash TC58Axxx и KM29xxxx
- добавлена поддержка 28-выводных AVR, TinyAVR и AT90C8534
- добавлена поддержка микроконтроллеров An15Exx (Angstrom)

## V3.3

# Руководство пользователя PICPROG

---

- добавлена полная поддержка Intel 28F001BX
- добавлена поддержка Macronix MX28xxxx
- добавлена поддержка AMD Am28Fxxx(A) и STMicro M28Fxxx
- ускорена процедура программирования побайтных микросхем Flash памяти
- увеличено время ожидания при связи с PC для максимально надежной работы под Win32 на медленных компьютерах
- доработана процедура снятия защиты с PIC16x8x

## V3.4

- исправлена ошибка верификации AVR в режиме внутрисхемного программирования (появилась в версии 3.3)
- добавлена поддержка внутрисхемного обновления прошивки программатора при условии установки в PicProg кристалла 16F877 вместо 16C77/67
- добавлена полная поддержка Scenix SX18/20/28/48/52 старых и новых модификаций, а также режим внутрисхемного программирования этих кристаллов
- добавлена поддержка семейства AT89C5x (требуется дополнительный адаптер)
- добавлена поддержка программирования семейства AT89Sxxxx в параллельном режиме
- добавлена поддержка микроконтроллеров ACE
- добавлена поддержка микросхем AT24C01, X24C01P и аналогичных
- добавлена поддержка SPI EEPROM без регистра состояния (типа X25C02)
- добавлена возможность корректного чтения параллельных кристаллов памяти с защелкой адреса (типа FM1608/1808)
- добавлена поддержка микросхем FM25160/25L160
- добавлены процедуры снятия битов защиты для некоторых старых версий кристаллов серий AT89Cx051 и AT89C5x
- доработан алгоритм записи микросхем AT49F010 с целью повышения надежности записи
- добавлена возможность программирования ATmega103/603 в параллельном режиме
- доработаны процедуры стирания AVR с учетом последних изменений в спецификации программирования ATMEL

## V3.5

- добавлена поддержка PIC18Cxxx
- исправлена ошибка программирования HCS360/361
- добавлена поддержка HCS101/201
- сделаны некоторые изменения в алгоритмах программирования ATMEL AVR для лучшей совместимости с рядом будущих кристаллов

## V3.6

- добавлена поддержка PIC16F7x
- добавлена поддержка AT17Cxxx
- добавлена поддержка DevID новых AT89C(87F)5x
- добавлена поддержка ATtiny15
- добавлена поддержка ATmega16x
- добавлена поддержка FM24Cxxx в 20-выводном корпусе
- исправлена ошибка программирования битов защиты ATtiny12 в высоковольтном режиме
- модифицирован алгоритм записи AT89C5x для лучшей совместимости с определенными партиями микросхем

## Приложение G. Часто Задаваемые Вопросы (FAQ) по программатору PicProg

---

- У меня возникла проблема с программатором PicProg. Что мне делать?
- У меня программатор версии 1.0. Стоит ли мне загружать обновленную версию программного обеспечения с вашего сайта или мой программатор уже не поддерживается?
- В Вашей программе PicProg для Windows находится вирус-троян. Зачем вы за мной следите?
- При запуске программы PicProg для Windows на кнопках нет никаких рисунков и обозначений, а закладки программы не подписаны. В чем дело?
- Почему мой принтер не работает с PicProg в прозрачном режиме?
- Можно ли запрограммировать с помощью PicProg микросхемы 573РФх?
- Я хочу запрограммировать микросхему 2764. Поскольку такой микросхемы в списке поддерживаемых нет, то я выбираю тип 27С64. Однако программатор говорит, что микросхема не чистая, а при попытке записи выдает множество ошибок. Разве эти микросхемы не идентичны?
- Раньше микросхемы АТ49F010 вались нормально, а сейчас при программировании возникает довольно много ошибок. Что случилось?
- Почему при попытке снять защиту кода с PIC16C84 в PicProg для Windows нажатием клавиш Ctrl-F7 защита не снимается, а вместо этого кристалл стирается целиком?
- PicProg для Windows сообщил об успешном снятии защиты кода с PIC16F84. Однако считанная после этого из кристалла информация содержит одни нули. Что делать?
- Почему при попытке программирования микросхемы АТ89С51 на PicProg загорается красная лампочка, а ПО сообщает о потере связи с программатором?

### **У меня возникла проблема с программатором PicProg. Что мне делать?**

Во-первых, поискать в данном списке Часто задаваемых вопросов: возможно, данная проблема возникает не только у Вас.

Затем внимательно прочитайте "Руководство пользователя PicProg": там содержится на удивление много полезной для работы информации...

Далее попробуйте загрузить с нашего сайта <http://www.telesys.ru/picprog/picpupgr.htm> самую последнюю версию программного обеспечения: может быть, Ваша проблема уже решена.

Возможно, после обновления ПО Вас попросят обновить и прошивку Вашего программатора. В этом случае новую прошивку также можно найти на нашем сайте. Ну и, наконец, если все эти рекомендации не помогли, обращайтесь к нам по E-mail [support@telesys.ru](mailto:support@telesys.ru) (предпочтительно) или по телефону: мы обязательно постараемся Вам помочь.

### **У меня программатор версии 1.0. Стоит ли мне загружать обновленную версию программного обеспечения с вашего сайта или мой программатор уже не поддерживается?**

Программное обеспечение, разрабатываемое для программаторов PicProg (как DOS-версия, так и Windows-версия), совместимо с любыми версиями аппаратуры PicProg, начиная с версии 1.0. При этом расширение ассортимента программируемых кристаллов касается также всех версий PicProg, если аппаратура его позволяет. Поэтому загрузить обновленную версию программного обеспечения всегда полезно. В конце концов, если Ваша версия программатора не позволяет работать с определенным типом кристаллов, то ПО сообщит Вам об этом и порекомендует пути возможного решения проблемы.

## **В Вашей программе PicProg для Windows находится вирус-троян. Зачем вы за мной следите?**

Многие антивирусные программы в начале 2000 года стали сообщать о наличии в файле fngkhlb.dll, распространяемом в составе PicProg для Windows, вируса Backdoor.Buttman. На самом деле автор указанного вируса просто использовал библиотеку общего пользования fngkhlb.dll в своем творении, что не сразу поняли создатели антивирусных программ. К настоящему моменту в большинстве антивирусов эта ошибка уже исправлена, поэтому Вам стоит обновить базы данных своего антивирусного пакета.

## **При запуске программы PicProg для Windows на кнопках нет никаких рисунков и обозначений, а закладки программы не подписаны. В чем дело?**

По-видимому, у Вас установлена ОС Windows95 или WindowsNT. Дело в том, что в состав указанных ОС входит устаревшая версия системной библиотеки Common Controls, которая поддерживает не все элементы пользовательского интерфейса, необходимые для работы PicProg для Windows.

Обновленную версию этой библиотеки можно загрузить с сайта фирмы Microsoft: <http://www.microsoft.com/msdownload/ieplatform/ie/comctrlx86.asp>

## **Почему мой принтер не работает с PicProg в прозрачном режиме?**

Работа принтера, подключенного к тому же LPT-порту, что и PicProg, обеспечивается только в однонаправленном режиме LPT-порта. Поэтому в случае возникновения проблем с прозрачным подключением попробуйте установить SPP-режим работы LPT-порта в BIOS-е Вашего компьютера.

Следует учесть, однако, что некоторые так называемые Windows-принтеры (например, OKIPAGE 4W) вообще не работают в SPP-режиме. В этом случае PicProg и принтер следует подключать к разным LPT-портам. Кроме того, PicProg можно подключить и к COM-порту компьютера, воспользовавшись дополнительным переходником, схема которого приведена в "Руководстве пользователя PicProg".

## **Можно ли программировать с помощью PicProg микросхемы 573РФх?**

К сожалению, указанные микросхемы потребляют слишком большой ток даже в режиме чтения, который PicProg не может надежно обеспечить. В рамках существующей архитектуры PicProg данное ограничение обойти невозможно даже с помощью дополнительных адаптеров, поэтому такие микросхемы запрограммировать или считать не получится.

## **Я хочу запрограммировать микросхему 2764. Поскольку такой микросхемы в списке поддерживаемых нет, то я выбираю тип 27С64. Однако программатор говорит, что микросхема не чистая, а при попытке записи выдает множество ошибок. Разве эти микросхемы не идентичны?**

Не случайно в списке поддерживаемых нет микросхем 27xxx. Данные микросхемы выполнены по технологии NMOS и имеют поэтому слишком большой для PicProg ток потребления (см. ответ на предыдущий вопрос). Микросхемы же серии 27Сxxx выполнены по CMOS-технологии и ток их потребления гораздо меньше.

## **Раньше микросхемы AT49F010 вались нормально, а сейчас при программировании возникает довольно много ошибок. Что случилось?**

Начиная с определенного момента, фирма ATMEL модернизировала технологию производства указанных микросхем, в результате чего они стали потреблять заметно больший ток в режиме программирования. К счастью, данную проблему удалось решить программно. Вам лишь необходимо обновить прошивку программатора до версии 3.4(2.4) или выше.

# **Руководство пользователя PICPROG**

---

**Почему при попытке снять защиту кода с PIC16C84 в PicProg для Windows нажатием клавиш Ctrl-F7 защита не снимается, а вместо этого кристалл стирается целиком?**

Вероятно, у Вас установлена европейская версия ОС Windows (немецкая, итальянская, ...). По неустановленной причине нажатие клавиши Ctrl в этих версиях ОС обрабатывается неправильно. В последней версии ПО, доступной для загрузки с нашего сайта, данная проблема решена.

**PicProg для Windows сообщил об успешном снятии защиты кода с PIC16F84. Однако считанная после этого из кристалла информация содержит одни нули. Что делать?**

Данные из PIC16F8x со снятой защитой кода необходимо считывать при напряжении питания (поле Vdd чтения программы PicProg для Windows), равном 2.0В. Между прочим, это написано в Руководстве пользователя...

**Почему при попытке программирования микросхемы AT89C51 на PicProg загорается красная лампочка, а ПО сообщает о потере связи с программатором?**

Для программирования с помощью PicProg микросхем семейства MCS51 (в том числе и AT89C5x, кроме AT89Sxxxx) требуется дополнительный адаптер (Адаптер 14), о чем PicProg для Windows сообщает в поле рядом с флагом ISP-программирования, а также в сообщении о необходимости установки микросхемы перед программированием. Микросхемы данного типа НЕ программируются в основной ZIF-колодке PicProg. Адаптер 14 можно приобрести у производителя или изготовить самостоятельно по схеме, приведенной в "Руководстве пользователя PicProg".