

# ROMSERVICE



**Универсальный программатор ПЗУ**

**модель AutoProg**

**Инструкция по эксплуатации**

Москва 2002



## Содержание.

1. Назначение прибора.....	7
2. Устройство прибора.....	7
3. Обновления.....	10
4. Работа с автономным питанием.....	11
5. Основы работы с энергонезависимой памятью. Особенности микросхем различных семейств.....	12
5.1. Общие сведения.....	12
5.2. EPROM.....	14
5.3. FLASH.....	15
5.4. EEPROM.....	17
5.5. Serial EEPROM (SEEPROM).....	17
5.6. Serial FLASH.....	19
5.7. Firmware HUB и LPC FLASH.....	20
5.8. Микроконтроллеры семейств MCS-48, UPI-42.....	21
5.9. Микроконтроллеры (ОМЭВМ) семейства MCS-51.....	21
5.10. PIC-контроллеры.....	23
5.11. AVR-контроллеры.....	25
6. Подготовка к работе.....	27
6.1. Общие указания.....	27
6.2. Выбор типа и установка ПЗУ в программатор.....	28
7. Порядок работы в автономном режиме.....	29
7.1. Общие указания.....	29
7.2. Описание команд.....	31
7.2.1. Меню <b>Main commands</b> ( <b>Основные команды</b> ).....	31
7.2.1.1. <b>Write Chip</b> ( <b>Записать м/схему</b> ).....	31
7.2.1.2. <b>Chip Check Sum</b> ( <b>К. Сумма м/схемы</b> ).....	32
7.2.1.3. <b>Buffer Check Sum</b> ( <b>К. Сумма буфера</b> ).....	32
7.2.1.4. <b>Read Chip</b> ( <b>Считать м/схему</b> ).....	32
7.2.1.5. <b>Erase Chip</b> ( <b>Стереть м/схему</b> ).....	33
7.2.1.6. <b>Test Chip</b> ( <b>Проверить м/сх.</b> ).....	33
7.2.2. Меню <b>Addit. commands</b> ( <b>Дополнит.команды</b> ).....	33
7.2.2.1. <b>Compare</b> ( <b>Сравнить</b> ).....	33
7.2.2.2. <b>Verify</b> ( <b>Проверить</b> ).....	33
7.2.2.3. <b>Blank test</b> ( <b>Провер. чистоту</b> ).....	34
7.2.2.4. <b>Erase Block</b> ( <b>Стереть блок</b> ).....	34
7.2.2.5. <b>Write &amp; SDP off</b> ( <b>Записать без SDP</b> ).....	34
7.2.2.6. <b>Set Lock Bit</b> ( <b>Устан.бит защиты</b> ).....	34

7.2.2.7.	Set Lock Bit 1	( Устан.бит защ. 1 )	.....	34
7.2.2.8.	Set Lock Bit 2	( Устан.бит защ. 2 )	.....	34
7.2.2.9.	Set Lock Bit 3	( Устан.бит защ. 3 )	.....	35
7.2.2.10.	write Code Table	( Запис.шифр.табл. )	.....	35
7.2.2.11.	Read Prot. Reg.	( Считать рег.защ. )	.....	35
7.2.2.12.	write Prot. Reg.	( Запис. рег.защ. )	.....	35
7.2.2.13.	Edit Prot. Reg.	( Измен. рег.защ. )	.....	35
7.2.2.14.	Clear Prot. Reg.	( Очист. рег.защ. )	.....	35
7.2.2.15.	Lock Prot. Reg.	( Блокир. рег.защ. )	.....	35
7.2.2.16.	Read Status Reg.	( Счит.рег.статуса )	.....	36
7.2.2.17.	write Status Reg	( Зап. рег.статуса )	.....	36
7.2.2.18.	Edit Status Bits	( Изм.биты статуса )	.....	36
7.2.2.19.	Read Option Reg.	( Счит. рег.опций )	.....	36
7.2.2.20.	write Option Reg	( Запис. рег.опций )	.....	36
7.2.2.21.	Edit Option Bits	( Измен.биты опций )	.....	36
7.2.2.22.	Read Configurat.	( Считать конфиг. )	.....	36
7.2.2.23.	write NEB posit.	( Устан.полож. НЕВ )	.....	37
7.2.2.24.	write Prot. Reg.	( Установ. защиту )	.....	37
7.2.2.25.	Edit NEB & Prot.	( Измен.защ. и НЕВ )	.....	37
7.2.2.26.	Protect Chip	( Защитить м/схему )	.....	37
7.2.2.27.	Read Lock Bits	( Счит. биты защ. )	.....	37
7.2.2.28.	Read Lock/SPIEN	( Счит. защ./SPIEN )	.....	38
7.2.2.29.	Read Lock/Fuses	( Счит. защ./конф. )	.....	38
7.2.2.30.	write Lock Bits	( Запис. биты защ. )	.....	38
7.2.2.31.	write SPIEN Fuse	( Запис. бит SPIEN )	.....	38
7.2.2.32.	write Fuse Bits	( Запис.биты конф. )	.....	38
7.2.2.33.	View Lock Bits	( Просм. биты защ. )	.....	38
7.2.2.34.	Edit Lock Bits	( Измен. биты защ. )	.....	39
7.2.2.35.	Edit SPIEN Fuse	( Измен. бит SPIEN )	.....	39
7.2.2.36.	Edit Fuse Bits	( Измен.биты конф. )	.....	39
7.2.2.37.	Read CFG & ID	( Считать CFG&ID )	.....	39
7.2.2.38.	write CFG & ID	( Записать CFG&ID )	.....	39
7.2.2.39.	Edit ID-code	( Изменить ID-код )	.....	39
7.2.2.40.	Edit Config.word	( Измен.слово конф )	.....	40

7.2.2.41.	Read Calibr.Byte	( Счит. калибр. б. )	40
7.2.2.42.	Read Reset Pol.	( Счит. полярность )	40
7.2.2.43.	Write Reset Pol.	( Запис.полярность )	40
7.2.2.44.	Edit Reset Pol.	( Измен.полярность )	40
7.2.2.45.	Enable DCLK	( Разрешить DCLK )	41
7.2.2.46.	Disable DCLK	( Запретить DCLK )	41
7.2.2.47.	ISP mode	( Прогр. в системе )	41
7.2.2.48.	Socket Mode	( Прогр. в панели )	41
7.2.2.49.	Se1. Data Memory	( Выбр. пам.данных )	41
7.2.2.50.	Se1.Progr.Memory	( Выбр. пам.прогр. )	41
7.2.2.51.	Switch To LDROM	( Перекл. в LDROM )	41
7.2.2.52.	Switch To APROM	( Перекл. в APROM )	42
7.2.2.53.	Select Add. Mem.	( Выбр. доп.память )	42
7.2.2.54.	Select Main Mem.	( Выбр. осн.память )	42
7.2.2.55.	Set Memory Part	( Выбр. часть пам. )	42
7.2.3. Меню	buffer commands	( Операции буфера )	43
7.2.3.1.	Save Up	( Сохран. в конец )	43
7.2.3.2.	Download	( Загруз.в начало )	43
7.2.3.3.	Edit Buffer	( Редактир. буфер )	43
7.2.4. Меню	Macro commands	( Макросы )	44
7.2.4.1.	New Macro	( Начать запись )	44
7.2.4.2.	End Macro	( Окончить запись )	44
7.2.4.3.	Run Macro	( Запустить макрос )	44
7.2.4.4.	Stop Macro	( Прервать макрос )	45
7.2.5. Меню	Settings&Service	( Настройки/сервис )	45
7.2.5.1.	Change Language	( Изменить язык )	45
7.2.5.2.	Turn Key Sound	( Озвучив. клавиш )	45
7.2.5.3.	Turn Event Sound	( Озвучив. событий )	45
7.2.5.4.	Model Info	( Информация )	45
7.2.5.5.	Check Database	( Пров.базу данных )	45
7.2.6. Меню	Set voltages	( Установ. напряж. )	46
7.2.6.1.	Write/Read	( Запись/чтение )	46
7.2.6.2.	Verification	( Напряж. проверки )	46
7.3.	Работа с ПЗУ различных типов.		46

---

8. Порядок работы под управлением персонального компьютера .....	46
9. Базовый комплект поставки .....	48
10. Гарантийные обязательства .....	48
<b>Приложение 1.</b> Контрольные суммы "чистых" микросхем .....	49
<b>Приложение 2.</b> Поблочно стираемые FLASH .....	51
<b>Приложение 3.</b> Отображение значений битов конфигурации AVR-контроллеров .....	53
<b>Приложение 4.</b> Распайка интерфейсного кабеля .....	54
<b>Приложение 5.</b> Возможные проблемы при работе с программатором и способы их решения .....	55

## 1. Назначение прибора.

Универсальный программатор AutoProg предназначен для работы с широким спектром микросхем энергонезависимой памяти семейств EPROM, Parallel & Serial EEPROM, Parallel & Serial FLASH, NVRAM, FRAM и др., а также с микроконтроллерами, содержащими такую память.

Программатор может работать как в автономном режиме, так и под управлением IBM-совместимого персонального компьютера (далее ПК).

Функциональные возможности и технические характеристики программатора AutoProg делают его пригодным для применения в условиях лабораторий и мастерских при разработке и ремонте электронной техники, а также при мелко- и среднесерийном промышленном производстве.

## 2. Устройство прибора.

На верхней (лицевой) стороне корпуса программатора расположены: контактная панель для установки микросхем, разъем расширения и внутрисхемного программирования, жидкокристаллический дисплей, 5-кнопочная клавиатура и два двухцветных светодиодных индикатора. В правой стенке корпуса размещены разъемы для подачи питания и подключения к персональному компьютеру. В нижней крышке корпуса имеется миниатюрное отверстие, за которым расположена скрытая кнопка принудительного запуска загрузчика.

Универсальная контактная панель позволяет устанавливать микросхемы в широких (600mil) и узких (300mil) корпусах типа "DIP", с количеством выводов от 8 до 40. Все микросхемы устанавливаются выводом 1 (ключом) в сторону прижимного рычага панели, в соответствии с рисунком, вытесненным на поверхности корпуса. Установка микросхем осуществляется при вертикальном положении рычага, для фиксирования рычаг переводится в горизонтальное положение.

Микросхемы в корпусе "DIP", имеющие более 40 выводов, а также выполненные в других видах корпусов (PLCC, SOIC, TSOP, QFP, SOJ, PGA, BGA и др.) устанавливаются в панель через

соответствующие переходники (адаптеры), которые могут быть приобретены отдельно или изготовлены самостоятельно. При этом, распайка переходников должна определяться соответствием распределения сигналов на микросхеме, с которой предполагается работать, и ее модификации в корпусе "DIP". Для микросхем, не выпускаемых в корпусе "DIP", сигналы на панели программатора и разъеме расширения расположены, в большинстве случаев, с учетом расположения выводов этих микросхем, что облегчает изготовление переходников, а во многих случаях, делает возможным использование стандартных широко распространенных переходников.

5-контактный разъем расширения используется при работе с микросхемами, требующими при программировании более 40 сигналов, а также при внутрисхемном программировании некоторых типов микроконтроллеров.

Жидкокристаллический дисплей, позволяющий отображать алфавитно-цифровую и псевдографическую информацию в двух строках по 16 символов, служит для вывода различных сообщений и другой информации, необходимой при работе с программатором, преимущественно в автономном режиме. Сообщения на дисплее выводятся на английском или русском языках. Переключение языка осуществляется командой в меню настроек. При редактировании различных параметров, изменяемый параметр на дисплее мигает.

Клавиатура программатора содержит четыре основных и одну дополнительную клавиши. Основные клавиши ("вверх", "вниз", "влево", "вправо") расположены в виде ромба, дополнительная клавиша ("режим") расположена слева.

При выборе пунктов меню, основные клавиши имеют следующие функции: "вверх", "вниз" – перебор пунктов меню, "вправо" – ввод выбранного пункта. При редактировании различных параметров и данных, клавиши "влево"/"вправо" используются для выбора параметра, который требуется изменить, а клавиши "вверх"/"вниз" – для его изменения. Длительное удержание клавиши "вправо" осуществляет ввод сделанных изменений, а длительное удержание клавиши "влево" – отмену изменений. При отображении сообщений о результатах выполнения операций, возникших ошибках, срабатывании защиты по току и др., для возврата программатора в текущий режим



работы, после прочтения сообщения, следует нажать любую клавишу. В различных режимах функции клавиш могут быть и иными. Эти функции, а также порядок использования клавиши "режим", описаны ниже, в соответствующих разделах инструкции.

Клавиша "режим" служит также выключателем питания программатора. Для включения/выключения питания, ее следует удерживать в нажатом состоянии более 3 секунд. Кроме того, включение программатора происходит автоматически при подаче внешнего питания.

Светодиод, помеченный надписью "CHIP", светится зеленым цветом в процессе выполнения операций над микросхемой, установленной в панели, когда на нее подано питание. В это время, установка/удаление микросхемы в панели не допускается. Этот светодиод светится красным цветом при срабатывании защиты по току, вызванном неисправностью программируемой микросхемы, ее неверным выбором в меню или неправильной установкой в панель.

Светодиод, помеченный надписью "BAT.", индицирует состояние аккумуляторной батареи, если она установлена в программатор. Подробнее см. раздел "Автономное питание".

В качестве внешнего питания программатора может быть применен любой источник постоянного напряжения 12...16 В при токе до 500 мА. Настоятельно рекомендуется использовать сетевой адаптер, входящий в комплект поставки. При работе в автомобиле, можно использовать напряжение 12 В бортовой сети. Внутренний контакт разъема питания – "+", внешний – "-".

Для подключения к последовательному порту компьютера, используется соединительный кабель, входящий в комплект поставки. При необходимости изготовления дополнительного кабеля (например, большей длины) следует пользоваться таблицей распайки, приведенной в приложении 4.

Использование режима принудительного запуска загрузчика при помощи скрытой кнопки, может потребоваться в случае сбоя в процессе обновления микропрограммы, например, из-за неожиданного выключения электроэнергии. Подробнее см. раздел "Обновление".

Программатор оснащен буферной памятью (далее – буфер), используемой для хранения информации при копировании и

сравнении микросхем в автономном режиме. Буфер программатора – энергонезависимый, т.е. способен хранить информацию в отсутствие электропитания. Буфер доступен для чтения и записи (загрузки) при работе с компьютером. Для этого необходимо выбрать в меню тип микросхемы "Buffer".

### 3. Обновления.

Конструкция программатора "AutoProg" предусматривает три вида обновлений, с целью расширения его функциональных возможностей, в том числе добавления поддержки работы с новыми типами микросхем.

**Обновление базы данных (Update)** позволяет добавить поддержку микросхем, алгоритмы считывания/записи/стирания которых совпадают с алгоритмами других микросхем, уже поддерживаемых программатором, но отличающихся маркировкой, объемом памяти, расположением сигналов на выводах, напряжениями питания при записи/считывании/проверке, напряжением программирования, набором доступных операций. Update осуществляется программным путем под управлением персонального компьютера, при запуске новой версии управляющей программы. Новые версии программы размещаются на web-сайте ROMSERVICE, распространяются БЕСПЛАТНО.

**Обновление микропрограммы контроллера программатора (Upgrade)** позволяет добавить новые алгоритмы для включения в список поддерживаемых, практически любых микросхем. Для программаторов с установленным модулем самообновления, Upgrade осуществляется программным путем, аналогично Update. Для программаторов без модуля самообновления, Upgrade заключается в перепрограммировании или замене микроконтроллера специалистами ROMSERVICE. Модуль самообновления может быть приобретен отдельно и установлен в программатор самостоятельно.

**Дополнительные и сменные модули**, которые могут быть самостоятельно приобретены и установлены в программатор, благодаря его модульной конструкции, позволяют расширить общие потребительские свойства программатора и улучшить его

технические характеристики. К выпускаемым в настоящее время и планируемым к выпуску дополнительным модулям, относятся:

- аккумуляторная батарея для обеспечения полностью автономной эксплуатации программатора в условиях недоступности электросети;
- модуль самообновления (самоперепрограммируемый микроконтроллер с программой-загрузчиком, позволяющий осуществлять Upgrade программным путем (см. выше);
- модули памяти объемом 16 и 32 Мбит для установки в программатор в качестве буфера;

В дальнейшем, ассортимент дополнительных модулей будет расширен. В перспективном плане - электронный диск, контроллер порта USB и др.

#### **4. Работа с автономным питанием.**

Конструкция программатора "AutoProg" предусматривает возможность полностью автономной работы с питанием от аккумуляторной батареи (аккумулятора). Для этого внутри корпуса программатора имеется место для установки такого аккумулятора, а контроллер питания программатора обеспечивает поддержку управления режимами заряда/разряда Ni-MH аккумуляторов номинальным напряжением 7,2 В, емкостью не менее 500 мАч.

Аккумулятор не входит в базовый комплект поставки, его следует приобретать отдельно. Инструкция по установке прилагается к аккумулятору.

Заряд аккумулятора осуществляется автоматически при подключении программатора к источнику внешнего электропитания. Режим заряда включается при остаточном заряде батареи менее 60...70% и выключается автоматически при достижении полного заряда. Заряд может производиться в процессе работы с программатором (обычный режим) или при выключенном программаторе (ускоренный режим).

Состояние аккумулятора отображается светодиодным индикатором, помеченном надписью "BAT.". Непрерывное зеленое свечение этого индикатора соответствует режиму заряда аккумулятора, мигающее - состоянию окончания заряда. Красное свечение предупреждает о скором разряде аккумулятора, при этом

мигание соответствует остаточному заряду менее 8...14%, непрерывное свечение - заряду менее 2...5%. В последнем случае, во избежание повреждения микросхемы, установленной в панель проگرامматора, или искажения информации в ней, рекомендуется как можно быстрее прекратить работу с проگرامматором, при необходимости, прервав выполнение текущей операции, удалить микросхему из панели, затем оставить проگرامматор включенным до автоматического выключения.

Время полного заряда батареи в обычном режиме составляет 15...25 часов, в ускоренном - около 2 часов. Время непрерывной работы проگرامматора от полностью заряженного аккумулятора составляет примерно 3...5,5 часов, в зависимости от типа аккумулятора, степени его износа, интенсивности работы с проگرامматором и типов проگرامмируемых микросхем.

Регулярно повторяющийся неполный разряд аккумулятора может привести к снижению его емкости (т.н. "эффект памяти"). Несмотря на то, что у Ni-MH аккумуляторов этот эффект выражен слабо, рекомендуется, время от времени, осуществлять "тренировку" батареи, полностью разряжая ее (оставляя проگرامматор включенным до автоматического выключения), затем полностью заряжая ее.

## **5. Основы работы с энергонезависимой памятью. Особенности микросхем различных семейств.**

### **5.1. Общие сведения.**

Энергонезависимой называется память, способная хранить информацию в отсутствие электропитания (в отличие от оперативной памяти, информация в которой при выключении питания теряется). Довольно часто используемая аббревиатура "ПЗУ" – Постоянное Запоминающее Устройство (по англ. ROM – Read Only Memory – память только для чтения), строго говоря, не совсем корректна. Первые, наиболее старые представители энергонезависимой памяти, действительно использовались в аппаратуре только в режиме чтения, а их запись (проگرامмирование) осуществлялась, либо в процессе изготовления кристалла, либо перед установкой в аппаратуру с помощью проگرامматора. В дальнейшем, по мере совершенствования

технологии производства и упрощения методов и алгоритмов записи, их современные модификации все чаще стали использовать в приборах и устройствах в режимах записи, стирания и перезаписи. Например, в модулях фискальной памяти кассовых аппаратов, в них заносится итоговая информация о дневной выручке и количестве покупок. В телевизорах ПЗУ используют для хранения различных настроек, а в телефонных аппаратах – для хранения и быстрого набора часто используемых телефонных номеров (записная книжка). Все эти применения противоречат самому смыслу понятия "память только для чтения". Попытки устранить это противоречие привели к обростанию аббревиатуры "ПЗУ" уточняющими приставками: ППЗУ – программируемые ПЗУ, СППЗУ – стираемые ППЗУ, РПЗУ – репрограммируемые ПЗУ (PROM – Programmable ROM, EPROM – Erasable PROM, EEPROM – Electrically Erasable PROM) и т.д. Однако, наиболее точным обобщающим названием этого класса приборов является "энергонезависимая память". Этого понятия, применяя сокращение ЭП, мы и будем придерживаться в настоящей инструкции.

Энергонезависимая память выпускается, как в виде самостоятельных устройств – собственно микросхем памяти, так и в составе других устройств, например, микроконтроллеров, где она используется для хранения управляющей микропрограммы, а иногда и данных.

По способу стирания (удаления старой информации перед записью новой), ЭП делится на нестираемую (программируемую в процессе изготовления кристалла либо однократно программируемую потребителем), стираемую ультрафиолетовым излучением, электрически стираемую (стирание осуществляется электрическими сигналами программатора), автостираемую (удаление старой информации выполняется автоматически, в процессе записи новой).

Наиболее простые представители микросхем ЭП содержат один массив ячеек памяти, информация в котором может быть считана, записана, а для электрически стираемых микросхем и стерта, при помощи программатора. Более сложные приборы содержат несколько (обычно – 2) массивов памяти, например, микроконтроллеры, содержащие память программ и память

данных. Во многих микросхемах памяти и микроконтроллерах имеются также одна или несколько дополнительных ячеек, запись определенных данных в которые, управляет отдельными специфическими функциями микросхемы. К таким дополнительным ячейкам относятся, например, биты и регистры защиты, конфигурации, состояния, идентификации и др.

Краткие сведения о функциональном составе различных микросхем памяти и микроконтроллерах приведены ниже, отдельно по семействам микросхем. Более подробную информацию по конкретным микросхемам можно найти в специальной литературе и в фирменных описаниях микросхем (datasheets), распространяемых производителями через web-сайты, а также на компакт-дисках и в виде бумажных каталогов.

Для первичной идентификации информации в микросхеме памяти или в файле, часто используют "контрольную сумму" – число, упрощенно говоря, представляющее собой сумму всех данных в блоке памяти. Обычно контрольной суммой пользуются для быстрой проверки микросхем после записи (она должна совпасть с контрольной суммой микросхемы-образца или файла, из которого производилась запись). Следует иметь в виду, что различные программаторы и программы используют при подсчете контрольной суммы разные алгоритмы (формулы), поэтому значения контрольной суммы одной и той же информации, вычисленные разными программаторами, могут различаться.

## 5.2. EPROM.

Микросхемы EPROM (наиболее распространены серии 27xxx и 27Cxxx производства различных фирм и их отечественные аналоги) выпускается в однократно программируемом, УФ-стираемом и электрически стираемом исполнении.

Корпус УФ-стираемых EPROM снабжен окном из кварцевого стекла, прозрачного для ультрафиолетовых лучей. Стирание осуществляется источником ультрафиолетового излучения, воздействующим на кристалл через это окно. Для надежного стирания, время облучения рекомендуется увеличить на 50% по отношению к минимально необходимому, после которого микросхема считывается программатором как чистая, либо

осуществлять контрольное считывание при пониженном напряжении питания микросхемы.

Для записи EPROM используется высокое напряжение (т.н. "напряжение программирования"). Требуемая величина напряжения программирования устанавливается программатором автоматически, но может быть, при необходимости, изменена пользователем. Следует иметь ввиду, что завышенное напряжение программирования, в большинстве случаев, приводит к выходу микросхемы из строя.

По организации памяти, EPROM делятся на 8- и 16-разрядные. При работе с последними, данные в буфере программатора или файле размещаются младшим байтом вперед. При выборе части ПЗУ начальный адрес и длина блока должны быть четными.

### **5.3. FLASH.**

Программатор AutoProg поддерживает работу с широким ассортиментом микросхем FLASH-памяти серий 28, 29, 39, 49 и др.

Ниже приводится краткое описание общих свойств и отличительных особенностей различных микросхем FLASH, знание которых может быть необходимым для работы с программатором.

Стирание всех микросхем FLASH-памяти осуществляется электрически, командами программатора. Стирание FLASH первого поколения осуществляется в два этапа. На первом производится предпрограммирование – запись по всем адресам кода 00. Второй этап – собственно стирание и специальная верификация с достиранием. Многие микросхемы FLASH, помимо общего стирания, допускают стирание отдельных частей памяти, называемых блоками или секторами, причем разбивка на блоки может быть равномерной (блоки равного объема) или неравномерной. При выполнении команды стирания блока, программатором будет выполнено стирание того блока, в котором находится начальный адрес выбранной для работы части микросхемы. Некоторые FLASH допускают только поблочное стирание, однако, для удобства пользователей, программатор AutoProg предлагает, при работе с такими микросхемами, команду общего стирания, которая выполняет поочередное стирание всех

блоков. Кроме того, отдельные фирмы выпускают автостираемые FLASH (подробнее см. ниже "Страничные FLASH").

Напряжение программирования FLASH первого-второго поколения составляет 12 В. Для большинства современных FLASH подача напряжения программирования не требуется при выполнении основных операций, однако оно может использоваться в особых случаях (например, при перезаписи защищенных блоков), а для некоторых микросхем – для более быстрой записи и стирания.

Многие микросхемы FLASH имеют возможность защиты отдельных блоков от случайного перепрограммирования. После установки защиты, информация в защищенных блоках не может быть изменена (стирание и запись этих блоков не будут выполнены, данные сохраняют старые значения). Часто, такая защита бывает однократно устанавливаемой, т.е. ее отключение невозможно. Однако, в большинстве микросхем, однократно устанавливаемая защита может временно отключаться при перезаписи программатором, путем подачи напряжения программирования, которое при нормальных операциях не требуется.

Страничные FLASH (например, серия 29 фирм Atmel, Silicon Storage Technology (SST) и Winbond) имеют функцию SDP (Software Data Protection – Программная защита данных) для обеспечения сохранности информации в памяти при сбоях в устройстве, где они применяются. При выключенной SDP программирование осуществляется простыми командами записи, при включенной – специальными усложненными последовательностями команд. Фирмы-изготовители этих микросхем настоятельно рекомендуют всегда включать этот режим. Программатор AutoProg предоставляет возможность программирования в обоих режимах. Стирание старой информации в этих микросхемах происходит автоматически, по мере записи новой, хотя имеется возможность быстрого стирания всей микросхемы. Запись данных производится программатором постранично. (Внешне это проявляется в заметном увеличении скорости записи.) При неполной загрузке страницы ее остаток просто стирается. Следовательно, при выборе для работы части ПЗУ, значения начального и конечного рабочих



адресов следует устанавливать кратными размеру страницы (64, 128 или 256 байт, в зависимости от типа микросхемы).

По организации памяти, FLASH делятся на 8-разрядные, 16-разрядные и переключаемые, способные работать как в 8-, так и в 16-разрядном режиме. При работе с 16-разрядными микросхемами, данные в буфере программатора или файле размещаются младшим байтом вперед. При выборе части ПЗУ начальный адрес и длина блока должны быть четными. Работа с переключаемыми FLASH осуществляется программатором AutoProg в 8-разрядном режиме.

#### **5.4. EEPROM.**

В отличие от FLASH, микросхемы EEPROM (серия 28Cxxx и др.) не требуют предварительного стирания старой информации перед записью новой, т.е. любую ячейку можно переписать, не затрагивая остальные. Все EEPROM не требуют подачи напряжения программирования. По алгоритмам записи EEPROM, объемом до 8 Кбайт включительно, делятся на два семейства – старые (EEPROM первого поколения), побайтно или постранично перезаписываемые, и современные – постранично перезаписываемые, с функцией программной защиты данных, аналогичной функции SDP страничных FLASH (см. выше). Все микросхемы объемом 16 Кбайт и более относятся к семейству современных.

#### **5.5. Serial EEPROM (SEEPROM).**

Микросхемы SEEPROM обладают свойствами обычных параллельных EEPROM (см. выше). Разница заключается в том, что доступ к хранящейся в них информации осуществляется по последовательным шинам I<sup>2</sup>C (серии 24, 85 и др.), Microwire (серия 93 и др.), 4-Wire (серия 59 и др.), SPI (серии 25, 95 и др.), MPS (серия 84), I<sup>2</sup>C/Bitstream (серия AT17), Mitsubishi 3-wire (серия M6M80), Rohm 3-wire (серия BR90) и INTERMETAL (микросхема NVM3060) и др. Большинство SEEPROM выполнены в 8-выводных корпусах.

Для всех микросхем SEEPROM стирание старой информации происходит автоматически в процессе записи новой.

Большинство микросхем SEEPROM серий 93, 59 M6M80, BR90 имеют 16-разрядную внутреннюю организацию. Многие

экземпляры имеют возможность переключения для работы в 8-разрядном режиме. При работе с ними программатор AutoProg использует 16-разрядный режим, данные в буфере программатора или файле размещаются старшим байтом вперед, что соответствует перераспределению данных в этих микросхемах при переключении разрядности. Каждая ячейке памяти микросхемы соответствует два байта в буфере программатора или файле, сначала младшие 8 бит, затем старшие. При выборе части ПЗУ начальный адрес и длина блока должны быть четными. Некоторые фирмы выпускают микросхемы серии 93, работающие только в 8-разрядном режиме.

Многие EEPROM имеют возможность защиты отдельных участков памяти от случайного перепрограммирования. В различных сериях микросхем такая защита организована по-разному.

EEPROM с шиной Microwire, имеющие регистр защиты, обычно содержат в своей маркировке букву "S" (например, 93Sxxx, 93CSxxx, 93LCSxxx). В этот регистр может быть записан адрес, начиная с которого информация в ПЗУ защищена от перезаписи. Содержимое этого регистра может быть считано, очищено (защита снята), записано (защита установлена) и затем заблокировано (защита установлена навсегда).

EEPROM с шиной I<sup>2</sup>C (34C02, 34WC02, 24LCS52, 24CS128/256 и некоторые другие) имеют возможность однократной установки защиты на часть памяти (обычно расположенную в конце массива). После установки защиты, информация в этом блоке не может быть изменена. Отключение (снятие) такой защиты невозможно.

В микросхемах 24xx65 фирмы Microchip также предусмотрена функция установки защиты информации от последующего изменения на любое количество расположенных подряд блоков размером 0.5 КБайт. Защита может быть установлена только 1 раз и не может быть снята. Кроме того, один из блоков (0.5 КБайт) имеет увеличенный в 10 раз ресурс по количеству циклов перезаписи (HEB – High Endurance Block), положение которого в адресном пространстве микросхемы может быть запрограммировано пользователем.

Выбор различных уровней защиты информации от случайной перезаписи в SPI SEEPROM серий 25xxx, 25Cxxx, 95xxx и др. осуществляется установкой отдельных битов имеющегося в микросхеме регистра статуса в соответствии с описанием на конкретную микросхему. В большинстве случаев, для отключения защиты и обеспечения возможности перезаписи всей памяти, в статус-регистр необходимо записать все нули.

SEEPROM серии AT17Cxxx содержат конфигурационный байт выбора полярности сигнала RESET/OE. Для выбора низкого активного уровня сигнала RESET, в этот байт необходимо записать значение FFh, для выбора высокого активного уровня RESET – 00h. Запись в этот байт других значений не допускается. Содержимое этого байта может быть также считано программатором.

Микросхемы AT17C512A/010A имеют дополнительный конфигурационный бит блокировки сигнала DCLK. Контрольное считывание состояния этого бита в данных микросхемах не предусмотрено.

SEEPROM серий 93, BR90 и некоторых других, выполненные в корпусах "SOIC", выпускается с двумя видами расположения сигналов на выводах – основным и альтернативным (кристалл внутри микросхемы повернут на 90°). При выборе такой микросхемы, программатор AutoProg, ориентируясь на буквенно-цифровые суффиксы в ее маркировке, автоматически устанавливает на контактах панели альтернативное расположение сигналов, что делает возможным применение для микросхем со стандартной и альтернативной разводкой одних и тех же переходников DIP8/SOIC8.

## 5.6. Serial FLASH.

В отличие от обычных параллельных FLASH, доступ к информации в Serial FLASH осуществляется по последовательной шине. Наиболее популярными представителями этого семейства ЭП являются серии M25P фирмы SGS-Thomson (ST-microelectronics), AT25F, AT45D и AT45DB фирмы Atmel, использующими шину SPI.

Serial FLASH серий M25P и AT25F обладают свойствами SEEPROM с шиной SPI серии 25, в т.ч. в части защиты от

случайной перезаписи, но требуют стирания старой информации перед записью новой.

Микросхемы серий AT45D и AT45DB не требуют стирания старой информации перед записью. Они имеют страничную организацию с размером страниц 256+8, 512+16 или 1024+32 байт. Помимо основной памяти (256, 512 или 1024 байт), каждая страница содержит дополнительную память (8, 16 или 32 байта), что делает эти микросхемы удобными для применения при изготовлении FLASH-карт и других устройств, действующих по принципу электронного диска. В этом случае, в дополнительной памяти размещается служебная информация о структуре диска.

При работе с сериями AT45D и AT45DB программатор AutoProg, как и большинство других программаторов, осуществляет раздельную работу с основной и дополнительной памятью. При этом информация из всех страниц выбранного типа памяти размещается в буфере программатора или файле слитно.

Serial FLASH серий AT45D и AT45DB не выпускаются в корпусах DIP. При выборе в меню этих микросхем с суффиксом "-R", распределение сигналов на панели программатора соответствует микросхемам в корпусе SOIC-28. Это делает возможным, при работе с ними, использование стандартных переходников DIP-28/SOIC-28 с разводкой "один-в-один". При выборе в меню этих микросхем без суффикса "-R", распределение сигналов соответствует корпусу SOIC-8. Для работы с другими корпусами, следует использовать переходники с соответствующей распайкой.

## 5.7. Firmware HUB и LPC FLASH.

Микросхемы семейств Firmware HUB и LPC FLASH предназначены для применения в современных материнских платах персональных компьютеров для хранения базовой системы ввода-вывода (BIOS). Они отличаются от микросхем FLASH-памяти, тем что доступ к хранящейся в них информации осуществляется по специальным 5-проводным шинам "Firmware HUB bus" и "LPC FLASH bus".

Firmware HUB и LPC FLASH не выпускаются в корпусе "DIP". При работе с ними, распределение сигналов на панели программатора соответствует разводке выводов этих микросхем,

выполненных в корпусе PLCC-32, что делает возможным применение стандартных широко распространенных переходников DIP32/PLCC32, предназначенных для работы с FLASH-памятью в 32-выводном корпусе PLCC.

### **5.8. Микроконтроллеры семейств MCS-48, UPI-42.**

В состав семейств MCS-48 и UPI-42 входят микросхемы с масочным ПЗУ (Mask ROM), информация в которое записывается в процессе изготовления микросхемы и не может быть изменена (серия 80xx), с однократно программируемым ПЗУ (OTP ROM) и УФ-стираемым ППЗУ (UVEPROM) (серия 87xx) и микросхемы без внутренней памяти программ. Для работы с последними программатор не нужен, а ПЗУ ОМЭВМ серии 80xx может быть только считано.

### **5.9. Микроконтроллеры (ОМЭВМ) семейства MCS-51.**

В состав семейства MCS-51 входят однократно программируемые и УФ-стираемые (например, серия 87), а также электрически стираемые (например, серии W77, W78, W89) микросхемы.

Корпус УФ-стираемых микроконтроллеров снабжен окном из кварцевого стекла, прозрачного для ультрафиолетовых лучей. Стирание осуществляется источником ультрафиолетового излучения, воздействующим на кристалл через это окно. Для надежного стирания, время облучения рекомендуется увеличить на 50% по отношению к минимально необходимому, после которого микросхема считывается программатором как чистая, либо осуществлять контрольное считывание при пониженном напряжении питания микросхемы.

Для записи большинства микроконтроллеров используется высокое напряжение (т.н. "напряжение программирования"). Требуемая величина напряжения программирования устанавливается программатором автоматически, но может быть, при необходимости, изменена пользователем. Следует иметь в виду, что завышенное напряжение программирования, в большинстве случаев, приводит к выходу микросхемы из строя.

Некоторые контроллеры семейства MCS-51, помимо основной памяти, используемой для хранения выполняемой микроконтроллером программы, содержат дополнительную энергонезависимую память для данных.

Некоторые контроллеры серии W78 фирмы Winbond содержат два ПЗУ программ – Application memory (APROM) и Loader memory (LDROM), что дает возможность осуществлять обновление программ непосредственно в процессе работы устройства, в котором установлен микроконтроллер. Программатор AutoProg позволяет работать с обоими ПЗУ. Стирание LDROM и APROM происходит одновременно.

Все контроллеры семейства MCS-51 имеют возможность защиты информации от считывания и копирования. Установку защиты следует производить после записи и контрольного считывания записанных данных. После установки защиты, считывание информации из контроллера становится невозможным или она считывается в зашифрованном виде.

Защита устанавливается путем программирования т.н. "битов защиты", количество которых может быть от 1 до 3. В некоторых современных микроконтроллерах предусмотрена функция считывания состояния битов защиты. В старых, такая возможность отсутствует.

Дополнительным средством защиты некоторых микроконтроллеров служит шифровальная таблица размером 32 или 64 байт, после записи которой, вся информация из микроконтроллера будет считываться в зашифрованном виде.

В микросхемах серий W77 и W78, биты защиты расположены в 8-разрядном регистре опций (Option Register). Бит 0 запрещает чтение информации из ПЗУ, бит 1 запрещает взятие данных из внутреннего ПЗУ командами, расположенными во внешнем ПЗУ, установка бита 2 (он имеется не у всех микросхем серии) включает функцию шифрования считываемых из ПЗУ данных. Шифрование осуществляется по случайному закону. Бит 7 регистра опций управляет коэффициентом усиления усилителя тактового генератора микроконтроллера. Установка бита 7 снижает коэффициент усиления вдвое, что позволяет уменьшить помехи по шинам питания. Остальные биты Option Register зарезервированы для использования в будущем, их установка не допускается.

Регистр опций доступен для чтения и записи, если бит 0 не установлен.

Биты защиты, шифровальная таблица и регистр опций стираются одновременно с остальной информацией в памяти микросхемы.

Серия AT89S/LS предусматривает дополнительный режим программирования по последовательному интерфейсу SPI (в т.ч. в составе различных устройств). Этот режим может быть разрешен или запрещен путем записи специального конфигурационного бита SPIEN. Для разрешения последовательного программирования в этот бит следует записать 0, для запрета – 1. Бит SPIEN доступен для записи и чтения в параллельном режиме при неустановленных битах защиты. При использовании последовательного режима, предварительное стирание информации не требуется (разумеется, если защита не была установлена). Старая информация стирается автоматически побайтно в процессе записи новой. Считывание состояния битов защиты не предусмотрено. При работе в этом режиме используется разъем расширения и внутрисхемного программирования. Сигналы на этом разъеме расположены в следующем порядке (слева направо): MOSI, SCK, GND, MISO, RST. Питание и тактирование микросхемы осуществляться устройством, в котором она установлена. Выводы микросхемы, используемые для программирования должны быть отключены от других цепей.

***При использовании режима внутрисхемного программирования, следует установить величину напряжения питания микросхемы в настройках программатора, равным напряжению питания микросхемы в схеме, где осуществляется ее программирование. Это необходимо для согласования уровней вырабатываемых программатором сигналов с напряжением питания схемы.***

Работа в панели программатора производится в параллельном режиме.

## 5.10. PIC-контроллеры.

Микросхемы PIC представляют собой высокопроизводительные однокристальные микроконтроллеры на базе RISC-процессора фирмы Microchip. В состав семейства входят однократно

программируемые, УФ-стираемые и электрически стираемые микросхемы.

Энергонезависимая память PIC-контроллеров имеет 12- или 14-разрядную организацию и содержит память программ (0,5...8К слов), ID-область (4 слова) и слово конфигурации. Каждое слово хранится в буфере программатора или файле в виде 2-х байтов, сначала младшие 8 бит, затем старшие.

Некоторые электрически стираемые PIC-контроллеры дополнительно имеют память данных размером 64, 128 или 256 слов, значащими в которых являются только 8 младших бит (младший байт), старшие биты содержат '0'.

В соответствии с рекомендациями фирмы Microchip, информация в файле или буфере программатора располагается следующим образом:

- для контроллеров с 12-разрядной памятью – сначала программа, затем ID-область, слово конфигурации по адресу 1FFEh-1FFFh;
- для контроллеров с 12-разрядной памятью – сначала программа, ID-область по адресам 4000h-4007h, слово конфигурации по адресу 400Eh-400Fh;
- память данных – по адресам 4200h-4xxxh.

4xxxh = 427Fh, 42FFh или 43FFh для памяти данных размером 64, 128 или 256 байт соответственно.

Последние 64 слова (128 байт) программной памяти микросхемы PIC14000 отведены для хранения калибровочной информации, предварительно записанной фирмой-изготовителем. Поскольку она не должна изменяться при программировании, следует выбирать для работы усеченный объем памяти PICa. При стирании микросхемы (стираемые экземпляры имеют в своем обозначении буквы JW), содержимое калибровочной области должно быть предварительно считано, сохранено в файле с программой по соответствующим адресам и в последствии восстановлено при записи путем выбора полного объема памяти PICa.

Память программ PIC12C5xx и PIC12C67x также содержит в конце калибровочную область размером 1 слово (2 байта) и имеет те же особенности в работе, что и PIC14000. Работа с полным объемом PIC12C5xx в автономном режиме не требуется и в программаторе не предусмотрена.



PIC-контроллеры имеют функцию защиты информации от несанкционированного доступа, которая включается путём записи определенных битов слова конфигурации.

Программирование подавляющего большинства PIC-контроллеров осуществляется в последовательном режиме, что делает возможным их программирование в составе различных устройств. Для использования этой возможности, программатор предусматривает для этих контроллеров режим внутрисхемного программирования (ISP – In-System Programming). Работа с микросхемой в этом режиме производится через разъем расширения и внутрисхемного программирования. Этот разъем может также использоваться для программирования PIC-контроллеров в корпусах SOIC, QFP и др., с целью упрощения конструкции переходников. Сигналы на разъеме расположены в следующем порядке (слева направо): Vpp (MCLR'), CLOCK (RB6), GND (Vss), DATA (RB7), Vdd. Сигнал Vdd (питание) используется только при работе с отдельными микросхемами, при внутрисхемном программировании он не подключается, а питание микросхемы осуществляется устройством, в котором она установлена. Выводы микросхемы, используемые для программирования должны быть отключены от других цепей. ***При использовании режима внутрисхемного программирования, следует установить величину напряжения питания микросхемы в настройках программатора, равным напряжению питания микросхемы в схеме, где осуществляется ее программирование. Это необходимо для согласования уровней вырабатываемых программатором сигналов с напряжением питания схемы.***

### 5.11. AVR-контроллеры

Микросхемы AVR представляют собой высокопроизводительные однокристальные микроконтроллеры на базе RISC-процессора фирмы ATMEL – серии AT90, ATmega, ATTiny.

Память AVR-контроллеров состоит из 16-разрядной FLASH-памяти программ (1...128 Кбайт), 8-разрядной EEPROM-памяти данных (64...4096 байт), двух битов защиты информации (Lock bits 1 & 2) и нескольких битов конфигурации (Fuse bits). ATmega161/163 имеют 4 дополнительных бита защиты и

увеличенное количество битов конфигурации. ATtiny10/11/28 не имеют EEPROM-памяти данных.

Lock bit 1 запрещает дальнейшую запись микросхемы, Lock bit 2 – считывание информации. В вышеупомянутых AVR-контроллерах присутствуют от одного до одиннадцати конфигурационных битов. Fuse bit 5 (SPIEN) разрешает режим последовательного программирования (в новой микросхеме разрешен). Назначение остальных в разных микросхемах различно и подробно описано в документации на них (они могут вообще отсутствовать). Некоторые микросхемы AVR, имеющие возможность тактирования от внутреннего RC-генератора, содержат служебную ячейку, в которой храниться записанное производителем число – калибровочный байт. Значение этого байта может быть считано и использовано записанной в микроконтроллер программой для коррекции частоты RC-генератора.

Большинство микросхем AVR допускают программирование как в параллельном, так и в последовательном режиме (по шине SPI). Параллельный режим является основным. Для некоторых микросхем в последовательном режиме отдельные функции недоступны (программирование и считывание битов конфигурации, считывание битов защиты). ATtiny28 программируется только в параллельном режиме.

Количество выводов 8-выводных AVR-контроллеров не позволяет реализовать режим параллельного программирования. Вместо него имеется режим "High voltage serial programming", называемый также "Special programming mode" (SPM).

Программирование в панели программатора осуществляется в параллельном режиме или режиме SPM. Для последовательного программирования используется разъем расширения и внутрисхемного программирования. Сигналы на этом разъеме расположены в следующем порядке (слева направо): MOSI, SCK, GND, MISO, RST. Питание и тактирование микросхемы осуществляется устройством, в котором она установлена. Выводы микросхемы, используемые для программирования должны быть отключены от других цепей. **При использовании режима внутрисхемного программирования, следует установить величину напряжения питания микросхемы в настройках**

**программатора, равным напряжению питания микросхемы в схеме, где осуществляется ее программирование. Это необходимо для согласования уровней вырабатываемых программатором сигналов с напряжением питания схемы.**

## 6. Подготовка к работе.

### 6.1. Общие указания.

Перед началом работы внимательно изучите настоящую инструкцию. Подключите источник питания.

Для работы программатора под управлением персонального компьютера (ПК), произведите установку управляющей программы, находящейся на компакт-диске, входящем в комплект поставки. Для установки Windows-версии программы, запустите файл setup.exe, далее следуйте указаниям инсталлятора. Установка DOS-версии программы осуществляется простым копированием файлов из архива на жесткий диск ПК в директорию \ROMSERV или в любую другую. Подключение программатора к ПК осуществляется через последовательный интерфейс RS-232 (порты COM1, COM2, COM3 или COM4). Для этого используется кабель, распайка которого приведена в Приложении 4. Такой кабель входит в комплект поставки программатора. Скорость обмена между программатором и ПК – 9600...57600 бит/с. **ВНИМАНИЕ!** Подключение и отключение интерфейсного кабеля между ПК и программатором рекомендуется производить только **при выключенном питании программатора** во избежание выхода из строя контроллера последовательного порта ПК.

Скорость обмена и номер последовательного порта выбираются в окне настроек управляющей программы, вызываемом одновременным нажатием комбинации клавиш "Alt+F9" на клавиатуре ПК. При первом запуске DOS-программы это окно появляется автоматически, введенные в окне параметры можно сохранить (Save). При этом будет создан файл конфигурации, который будет использоваться автоматически при последующих запусках.

После включения программатора, на дисплее отображается приглашающее сообщение, и программатор ожидает выбора режима работы. Для выбора автономного режима, следует нажать

любую клавишу клавиатуры программатора. Выбор режима работы под управлением персонального компьютера происходит автоматически после установки связи с компьютером по запросу управляющей программы. В этом режиме все управление осуществляется программой. Вся необходимая информация отображается на мониторе ПК, а ввод команд осуществляется с его клавиатуры или "мыши". На дисплее отображается сообщение

**PC-slave mode** ( **Управление от ПК** ).

## 6.2. Выбор типа и установка ПЗУ в программатор.

Выбор типа микросхемы для работы осуществляется при помощи меню-списка, отображаемом на дисплее программатора (при работе в автономном режиме) или в специальном окне управляющей программы (при работе под управлением персонального компьютера). Названия микросхем в меню соответствуют их обозначениям в фирменной технической документации производителей. На практике, маркировка на корпусе микросхем может несколько отличаться, например, быть сокращенной (особенно для миниатюрных корпусов). Микросхемы в меню расположены в алфавитном порядке, без учета буквенно-цифрового префикса - нескольких (до трех) символов, в большинстве случаев, обозначающих фирму-производителя. У некоторых фирм префикс обозначает исполнение микросхемы - тип корпуса и допустимый температурный диапазон. Префикс может вообще отсутствовать. Для сокращения списка микросхем и облегчения их выбора, буквенно-цифровые суффиксы в конце обозначения микросхем, не имеющие значения для программирования, во многих случаях опущены. Символ " \* " в префиксах или суффиксах маркировки заменяет любой символ, не имеющий значения для работы.

Для облегчения и ускорения поиска типа микросхемы в списке, предусмотрена возможность использования шаблона - ввода нескольких начальных символов названия микросхемы. При правке шаблона, для ввода предлагаются лишь те символы, которые действительно присутствуют в данной позиции хотя бы у одной микросхемы, удовлетворяющей уже составленной части шаблона.

Установку микросхемы в панель программатора следует производить после выбора ее типа в меню и подтверждения ввода (во избежание повреждения микросхемы или искажения информации в ней при смене конфигурации сигналов на панели программатора в момент выбора). После этого момента, питание и управляющие сигналы подаются на микросхему только во время выполнения операций над ней и сопровождаются зеленым свечением светодиода "CHIP". В паузах между операциями, на всех контактах панели, задействованных при установке микросхемы выбранного типа, поддерживается низкий потенциал, что обеспечивает безопасную установку и извлечение микросхемы.

## **7. Порядок работы в автономном режиме.**

### **7.1. Общие указания.**

Сеанс работы в автономном режиме начинается с выбора типа микросхемы. Названия микросхем отображаются в нижней строке дисплея, а названия фирм-производителей – в верхней. Клавиши "вверх"/"вниз" используются для перехода к предыдущей/следующей микросхеме, клавиша "вправо" – для ввода выбранной микросхемы, клавиша "режим" – для перехода в режим правки шаблона, а клавиша "влево" – для отображения списка последних 10 использованных микросхем (и возврата обратно к полному списку).

В режиме правки шаблона, для добавления символа в шаблон следует нажать клавишу "вправо", для выбора символа используются клавиши "вверх"/"вниз", для удаления символа – "влево". Возврат к просмотру списка микросхем (удовлетворяющих введенному шаблону) осуществляется клавишей "режим".

После выбора типа микросхемы для работы в меню, программатор перейдет в режим ожидания команд. Команды программатора размещены в нескольких меню. Для выбора меню служит клавиша "режим", для выбора команды – клавиши "вверх"/"вниз", для запуска команды – "вправо". Для завершения сеанса работы в автономном режиме, выбора другого типа микросхемы, перехода в режим работы с ПК, следует удерживать нажатой клавишу "влево".

Меню **main commands** (**Основные команды**) содержит набор часто используемых команд для операций с выбранной микросхемой (запись, считывание, стирание, подсчет контрольной суммы и др.)

Меню **Addit. commands** (**Дополнит. команды**) содержит реже используемы команды для операций с выбранной микросхемой.

В меню **Buffer commands** (**Операции буфера**) размещены команды перемещения и редактирования данных в буфере.

В меню **Macro commands** (**Макросы**) расположены команды для работы с макросами. Макросы позволяют запоминать последовательность операций, выполненных над одной микросхемой, с целью удобства ее повторения над партией микросхем.

В меню **Settings&Service** (**Настройки/сервис**) собраны команды управления функциями программатора, его проверки и т.д.

Просмотр и изменение величины напряжений питания микросхемы при записи, чтении и проверке, а также напряжения программирования, осуществляются командами, расположенными в меню **Set voltages** (**Установ. напряж.**).

При работе в автономном режиме, в начальной части буфера программатора (с нулевого адреса) выделяется рабочая область, размер которой, в большинстве случаев совпадает с объемом памяти выбранной микросхемы или выбранного типа памяти (для микросхем с несколькими массивами памяти). Исключения составляют микросхемы, объем памяти которых выражается значением, не являющимся двоично кратным числом. В этом случае, размер рабочий области округляется вверх до ближайшего кратного значения (например, для микросхемы объемом 20 Кбайт, рабочая область будет иметь размер 32 Кбайта). Другим исключением являются PIC-контроллеры. Для 12-разрядных PIC-контроллеров рабочая область составляет 8 Кбайт, для 14-разрядных – 32 Кбайта. Это необходимо для размещения вспомогательной информации (слова конфигурации и т.д.) по стандартным адресам, рекомендованным фирмой Microchip, что облегчает загрузку в буфер программатора данных из файла в

компьютере для ее дальнейшего тиражирования в PIC-контроллеры в автономном режиме.

При работе с микросхемами большого объема, превышающими размер буфера, копирование и сравнение данных следует производить по частям. Наоборот, при работе с микросхемами небольшого объема, свободную часть буфера можно использовать для хранения информации, считанной из нескольких микросхем-образцов. Для этого служат команды перемещения данных в буфере.

Перед началом выполнения команд записи и стирания, установки защиты и некоторых других, ошибочный ввод которых может привести к потере ценной информации, программатор требует подтверждения. На дисплее при этом появляется запрос:

**Are you sure?** ( **Вы уверены?** ). Для начала выполнения команды необходимо нажать клавишу "вправо", для отказа – любую другую "SELECT". Для команд, производящих однократно выполняемые операции с микросхемой, этот запрос сопровождается дополнительным предупреждением: **OneTimeOperation** ( **Однокр. операция** ).

В процессе выполнения большинства операций на дисплее высвечивается графическая шкала и число, показывающее степень ее выполнения в процентах. Для некоторых команд, в т.ч. тех, степень выполнения которых определить невозможно, отображается сообщение **Please wait...** ( **ждите...** ). Исключение составляют команды, выполняемые мгновенно или использующие дисплей для вывода другой информации.

## 7.2. Описание команд.

### 7.2.1. Меню **Main commands** ( **Основные команды** )

Это меню содержит основные, наиболее часто используемые команды для операций с микросхемами выбранного типа. Состав этого меню мало отличается для различных микросхем.

#### 7.2.1.1. **Write Chip** ( **Записать м/схему** )

Эта команда обеспечивает запись (программирование) в микросхему информации из буфера. Скорость записи

определяется объемом информации, типом и качеством прошиваемой микросхемы. В случае невозможности программирования вследствие неисправности микросхемы или неправильных действий пользователя, выполнение команды прекращается с выдачей сообщения: **Error** ( **Ошибка** ).

При работе со страничными FLASH и EEPROM, запись осуществляется в режиме SDP, функция SDP остается включенной.

При нормальном ходе записи процесс может быть прерван одновременным нажатием клавиш "влево" и "вправо".

#### 7.2.1.2. **Chip check sum** ( **К. Сумма м/схемы** )

Эта команда предназначена для подсчета контрольной суммы информации в микросхеме.

По окончании вычисления на дисплей выводится значение контрольной суммы. Нажатие любой клавиши обеспечивает выход в режим ожидания команд.

Буфер программатора при вычислениях не используется, его содержимое не изменяется. Таким образом, возможен подсчет контрольной суммы микросхем любого размера, в т.ч. превышающих размер буфера.

Можно использовать эту команду и для оперативного контроля чистоты микросхем. Контрольные суммы "чистых" микросхем приведены в Приложении 1.

#### 7.2.1.3. **buffer Check Sum** ( **К. Сумма буфера** )

Команда используется для подсчета контрольной суммы информации в буфере программатора и может применяться для проверки правильности чтения в буфер. Вывод информации аналогичен предыдущей команде.

#### 7.2.1.4. **Read Chip** ( **Считать м/схему** )

Выполнение этой команды обеспечивает чтение информации из микросхемы в буфер программатора для последующей записи в другую микросхему или для при сравнения двух микросхем между собой.



### 7.2.1.5. Erase Chip (Стереть м/схему) .

Команда обеспечивает стирание электрически стираемых микросхем.

В случае невозможности стирания вследствие неисправности микросхемы или неправильных действий пользователя, выполнение команды прекращается с выдачей сообщения:

**Error****Ошибка**

### 7.2.1.6. Test Chip (Проверить м/сх.) .

Команда служит для тестирования статических ОЗУ (SRAM). Это единственная команда в меню для этих микросхем.

По окончании тестирования на дисплей выводится сообщение

**Good chip****Микросхема исправна**

или

**Bad chip****Микросхема неисправна** .

### 7.2.2. Меню Addit. commands (Дополнит.команды) .

Меню содержит дополнительные, реже используемые команды, набор которых сильно различается для различных типов микросхем.

#### 7.2.2.1. Compare (Сравнить) .

Команда производит сравнение содержимого буфера с содержимым микросхемы. При нахождении отличий, на дисплей будут выведены адрес ячейки, содержащей отличия, значения несовпавших данных из буфера и ПЗУ, и результат операции "исключающее ИЛИ" над ними (например: **00057C:FFx4D=B2** ). Для продолжения сравнения следует нажать клавишу "вправо". Клавиша "влево" отменяет дальнейшее сравнение и переводит программатор в режим ожидания команд.

#### 7.2.2.2. Verify (Проверить) .

Команда дважды производит сравнение содержимого буфера с содержимым микросхемы, при различных значениях напряжения питания. Формат вывода сообщений при нахождении отличий, такой же, как у предыдущей команды.

**7.2.2.3. Blank test** ( **Провер. чистоту** )

Команда предназначена для проверки чистоты микросхемы. Формат вывода и управления аналогичны команде сравнения.

**7.2.2.4. Erase Block** ( **Стереть блок** )

Стирание блока для микросхем, допускающих поблочное стирание. Однократное выполнение команды обеспечивает стирание блока, внутри которого находится начальный адрес части памяти, выбранной командой **Set Memory Part** ( **Выбр. часть пам.** ).

В случае невозможности стирания вследствие неисправности ПЗУ или неправильных действий пользователя, выполнение команды прекращается с выдачей сообщения: **Error** ( **Ошибка** ).

**7.2.2.5. Write & SDP off** ( **Записать без SDP** )

Эта команда выключает функцию SDP для страничных FLASH и EEPROM, в которых такая имеется, и производит запись в обычном режиме. Функция SDP остается выключенной.

В остальном действие команды аналогично **Write Chip** ( **Записать м/схему** ).

**7.2.2.6. Set Lock Bit** ( **Устан.бит защиты** )

Эта команда обеспечивают запись (установку) бита защиты информации микроконтроллеров MCS-51, содержащих один такой бит.

**7.2.2.7. Set Lock Bit 1** ( **Устан.бит защ. 1** )

Эта команда обеспечивают запись (установку) 1-го бита защиты информации микроконтроллеров MCS-51, содержащих несколько таких битов.

**7.2.2.8. Set Lock Bit 2** ( **Устан.бит защ. 2** )

Эта команда обеспечивают запись (установку) 2-го бита защиты информации микроконтроллеров MCS-51, содержащих несколько таких битов.

**7.2.2.9. Set Lock Bit 3 (Устан.бит защ. 3) )**

Эта команда обеспечивают запись (установку) 3-го бита защиты информации микроконтроллеров MCS-51, содержащих несколько таких битов.

**7.2.2.10. Write Code Table (Запис.шифр.табл.) )**

Команда служит для записи шифровальной таблицы микроконтроллеров MCS-51. Данные для записи считываются из начала буфера программатора.

**7.2.2.11. Read Prot. Reg. (Считать рег.заш.) )**

Команда производит считывание адреса из регистра защиты EEPROM Microwire и его запоминание для последующего просмотра или установки защиты другой микросхемы.

**7.2.2.12. Write Prot. Reg. (Запис. рег.заш.) )**

Команда производит запись в регистр защиты EEPROM Microwire адреса, ранее считанного из другой микросхемы, или подготовленного командой **Edit Prot. Reg. (Измен. рег.заш.)**

**7.2.2.13. Edit Prot. Reg. (Измен. рег.заш.) )**

Команда используется для просмотра значения адреса (в 16-ричной форме), считанного из регистра защиты EEPROM Microwire, а также для ввода адреса для записи в регистр защиты микросхемы.

**7.2.2.14. Clear Prot. Reg. (Очист. рег.заш.) )**

Команда осуществляет очистку регистра защиты и снятие защиты от записи с микросхемы EEPROM Microwire.

**7.2.2.15. Lock Prot. Reg. (Блокир. рег.заш.) )**

Команда предназначена для блокировки регистра защиты EEPROM Microwire от последующих изменений. После выполнения этой команды микросхема защищается навсегда, снять защиту невозможно.

**7.2.2.16. Read Status Reg. (Счит. рег. статуса) )**

Команда производит считывание значения из регистра статуса SEEPROM SPI и его запоминание для последующего просмотра или записи в другую микросхему.

**7.2.2.17. Write Status Reg (Зап. рег. статуса) )**

Команда производит запись в регистр статуса SEEPROM SPI значения, ранее считанного из другой микросхемы, или подготовленного командой **Edit Status Bits (Изм. биты статуса)** ).

**7.2.2.18. Edit Status Bits (Изм. биты статуса) )**

Команда используется для просмотра значения, считанного из регистра статуса SEEPROM SPI (в двоичной форме), а также для ввода значения для записи в регистр статуса микросхемы.

**7.2.2.19. Read Option Reg. (Счит. рег. опций) )**

Команда производит считывание состояния битов регистра опций микроконтроллеров серий W77/W78 и его запоминание для последующего просмотра или записи в другую микросхему.

**7.2.2.20. Write Option Reg (Запис. рег. опций) )**

Команда производит запись в регистр опций микроконтроллеров серий W77/W78 значений битов, ранее считанных из другой микросхемы, или подготовленных командой **Edit Option Bits (Измен. биты опций)** ).

**7.2.2.21. Edit Option Bits (Измен. биты опций) )**

Команда используется для просмотра значений битов, считанных из регистра опций микроконтроллеров серий W77/W78, а также для ввода значений битов для записи в регистр опций микросхемы.

**7.2.2.22. Read Configurat. (Считать конфиг.) )**

Чтение содержимого ячеек конфигурации SEEPROM 24xx65 фирмы Microchip и их запоминание для последующего просмотра или записи в другую микросхему. Ячейки конфигурации включают

номер первого из защищенных блоков и их количество, номер блока с повышенным ресурсом (HEB – High Endurance Block).

#### 7.2.2.23. **Write HEB posit.** ( **Устан.полож. HEB** ) .

Команда осуществляет установку номера блока с повышенным ресурсом для SEEPROM 24xx65 фирмы Microchip в соответствии со значением, ранее считанным из другой микросхемы, или подготовленным ком командой **Edit HEB & Prot.** ( **Измен.защ. и HEB** ) .

#### 7.2.2.24. **Write Prot. Reg.** ( **Установ. защиту** )

Команда осуществляет запись в регистр защиты SEEPROM 24xx65 фирмы Microchip значений номера первого из защищаемых блоков и их количества, считанных ранее из другой микросхемы, или подготовленных командой **Edit HEB & Prot.** ( **Измен.защ. и HEB** ) .

#### 7.2.2.25. **Edit HEB & Prot.** ( **Измен.защ. и HEB** ) .

Команда используется для просмотра значений, считанных из ячеек конфигурации SEEPROM 24xx65 фирмы Microchip, а также для ввода значений для записи в ячейки конфигурации микросхемы.

Значения отображаются в виде трех 16-ричных цифр, в следующем порядке (слева направо): номер блока с повышенным ресурсом (HEB – High Endurance Block), количество защищенных блоков, номер первого защищенного блока.

#### 7.2.2.26. **Protect Chip** ( **Защитить м/схему** ) .

Команда осуществляет установку защиты на микросхемы SEEPROM I<sup>2</sup>C, оснащенные такой защитой. После выполнения команды, часть памяти в микросхеме не сможет быть перезаписана. Защита устанавливается однократно, отключение (снятие) защиты невозможно.

#### 7.2.2.27. **Read Lock Bits** ( **Счит. биты защ.** ) .

Чтение состояния битов защиты информации микроконтроллеров MCS-51, допускающих такую возможность.

**7.2.2.28. Read Lock/SPIEN (Счит. защ./SPIEN )**

Чтение состояния битов защиты и бита конфигурации SPIEN микроконтроллеров серий AT89S/AT89LS и их запоминание для последующего просмотра или записи в другую микросхему.

**7.2.2.29. Read Lock/Fuses (Счит. защ./конф.)**

Чтение состояния битов защиты и конфигурации (Fuse & Lock bits) AVR-контроллеров, и их запоминание для последующего просмотра или записи в другую микросхему.

**7.2.2.30. Write Lock Bits (Запис. биты защ.)**

Команда осуществляет запись битов защиты микроконтроллеров серий AT89S/AT89LS и AVR-контроллеров в соответствии со значениями, считанными ранее из другой микросхемы, или подготовленными командой **Edit Lock Bits (Измен. биты защ.)**.

**7.2.2.31. Write SPIEN Fuse (Запис. бит SPIEN )**

Команда осуществляет запись конфигурационного бита SPIEN, управляющего разрешением режима последовательного программирования, значением, считанным ранее из другой микросхемы, или подготовленным командой **Edit SPIEN Fuse (Измен. бит SPIEN )**.

**7.2.2.32. Write Fuse Bits (Запис. биты конф.)**

Команда осуществляет запись конфигурационных битов AVR-контроллеров в соответствии со значениями, считанными ранее из другой микросхемы, или подготовленными командой **Edit Fuse Bits (Измен. биты конф.)**.

**7.2.2.33. View Lock Bits (Просм. биты защ.)**

Команда служит для просмотра значений битов защиты информации, считанных из микроконтроллеров MCS-51.

**7.2.2.34. Edit Lock Bits** ( **Измен. биты защ.** )

Команда используется для просмотра значений битов защиты информации, считанных из микроконтроллеров MCS-51 или AVR-контроллеров, а также для ввода значений этих битов для записи в микросхему. Бит защиты, имеющий старший номер, отображается слева, младший – справа.

**7.2.2.35. Edit SPIEN Fuse** ( **Измен. бит SPIEN** )

Команда используется для просмотра значения конфигурационного бита разрешения режима последовательного программирования микроконтроллеров серий AT89S/AT89LS (SPIEN) а также для ввода значения этого бита для записи в микросхему. "0" – последовательное программирование разрешено, "1" – запрещено.

**7.2.2.36. Edit Fuse Bits** ( **Измен. биты конф.** )

Команда используется для просмотра значений конфигурационных битов AVR-контроллеров, а также для ввода значений этих битов для записи в микросхему. Отображение осуществляется в двоичном виде, в порядке, приведенном в таблице Приложения 3.

**7.2.2.37. Read CFG & ID** ( **Считать CFG&ID** )

Чтение содержимого ячеек идентификации (ID-кодов) и слова конфигурации PIC-контроллеров, и их запоминание в буфере программатора для последующего просмотра или записи в другую микросхему.

**7.2.2.38. Write CFG & ID** ( **Записать CFG&ID** )

Команда осуществляет запись ID-кодов и конфигурационного слова PIC-контроллера информацией, считанной ранее из другой микросхемы, или подготовленной командами **Edit ID-code** ( **Изменить ID-код** ) и **Edit Config.Word** ( **Измен. слово конф.** ).

**7.2.2.39. Edit ID-code** ( **Изменить ID-код** )

Команда используется для просмотра значений идентификационных кодов, считанных из PIC-контроллера (в 16-

ричной форме), а также для ввода значений этих кодов для записи в PIC-контроллер.

#### 7.2.2.40. **Edit Config.word** ( **Измен.слово конф** )

Команда используется для просмотра значений битов слова конфигурации, считанных из PIC-контроллера (в двоичной форме), а также для ввода значений этих битов для записи в PIC-контроллер.

#### 7.2.2.41. **Read Calibr.Byte** ( **Счит. калибр. б.** )

Команда осуществляет считывание калибровочного байта из служебной ячейки AVR-контроллера и выбор адреса для его размещения в памяти программ микроконтроллера. На дисплей в 16-ричном виде отображается считанное значение калибровочного байта и адрес размещения. Начальное значение адреса равно 0000. Для использования функции калибровки, его следует изменить на требуемое значение.

При последующем программировании AVR-контроллера, данные для записи в указанную ячейку памяти программ будут автоматически подменены на значение калибровочного байта.

#### 7.2.2.42. **Read Reset Pol.** ( **Счит. полярность** )

Чтение состояния конфигурационного байта выбора полярности сигнала RESET/OE микросхемы серии AT17 и его запоминание для просмотра или последующей записи в другую микросхему.

#### 7.2.2.43. **Write Reset Pol.** ( **Запис.полярность** )

Команда осуществляет запись конфигурационного байта выбора полярности сигнала RESET/OE в микросхему серии AT17, значением, предварительно считанным из другой микросхемы или подготовленным командой **Edit Reset Pol.** ( **Измен.полярность** ).

#### 7.2.2.44. **Edit Reset Pol.** ( **Измен.полярность** )

Команда используется для просмотра значения конфигурационного байта выбора полярности сигнала RESET/OE



микросхемы серии AT17, а также для ввода значения этого байта для записи в микросхему.

Конфигурационный байт отображается в 16-ричной форме и может принимать значения 00 или FF. Считывание другого числа свидетельствует об ошибке, а запись другого числа не допускается.

#### 7.2.2.45. **Enable DCLK** ( **Разрешить DCLK** )

Команда осуществляет установку конфигурационного бита блокировки сигнала DCLK.

#### 7.2.2.46. **Disable DCLK** ( **Запретить DCLK** )

Команда осуществляет сброс конфигурационного бита блокировки сигнала DCLK.

#### 7.2.2.47. **ISP mode** ( **прогр. в системе** )

Команда переключает программатор в режим внутрисхемного программирования памяти микроконтроллеров, имеющих такую возможность.

#### 7.2.2.48. **Socket Mode** ( **прогр. в панели** )

Команда выключает режим внутрисхемного программирования микроконтроллеров и возвращает режим программирования микросхем в панели программатора.

#### 7.2.2.49. **Se1. Data Memory** ( **Выбр. пам. данных** )

Команда переключает программатор в режим работы с памятью данных микроконтроллеров PIC, AVR и других, содержащих такую память.

#### 7.2.2.50. **Se1. Progr. Memory** ( **Выбр. пам. прогр.** )

Команда возвращает программатор из режима работы с памятью данных микроконтроллеров к работе с памятью программ.

#### 7.2.2.51. **Switch To LDROM** ( **Перекл. в LDROM** )

Команда переключает программатор в режим работы с памятью LDROM микроконтроллеров серии W78.

**7.2.2.52. Switch To APROM (Перекл. в APROM) .**

Команда переключает программатор в режим работы с памятью APROM микроконтроллеров серии W78.

**7.2.2.53. Select Add. Mem. (Выбр. доп.память) .**

Команда переключает программатор в режим работы с дополнительной памятью микросхем серий AT45D/AT45DB.

**7.2.2.54. Select Main Mem. (Выбр. осн.память) .**

Команда переключает программатор в режим работы с основной памятью микросхем серий AT45D/AT45DB.

**7.2.2.55. Set Memory Part (Выбр. часть пам.) .**

Эта команда предназначена для выбора (установки) части объема памяти в микросхеме для работы.

Задаваемые пользователем три числа, определяют номера начального и конечного выбранных блоков памяти и их общее количество. Блоки нумеруются с нуля. Например, **From 0 To 2 of 4** ( **с 0 по 2 из 4** ) для микросхемы объемом 8 Кбайт, означает выбор младших 6 Кбайт (первые три четверти всего объема памяти). **From 1 To 1 of 2** ( **с 1 по 1 из 2** ) соответствует выбору старшей половины микросхемы, а **From 0 To 0 of 2** ( **с 0 по 0 из 2** ) – младшей.

Как обычно, клавиши "влево"/"вправо" используются для выбора параметра, который требуется изменить, клавиши "вверх"/"вниз" – для его изменения. Длительное удержание клавиши "вправо" осуществляет ввод сделанных изменений, а длительное удержание клавиши "влево" – отмену изменений.

После выполнения этой команды, все действия программатора (запись, чтение, сравнение, подсчет контрольных сумм и др.) будут относиться только к выбранной части микросхемы и ее отображению в буфере.

Установка части отменяется после задания нового типа микросхемы в меню.

### 7.2.3. Меню **Buffer commands** ( **Операции буфера** )

Меню команд операций с буфером содержит команду редактирования информации в буфере, а также команды перемещения данных в буфере программатора, что позволяет хранить в нем информацию, считанную из нескольких различных микросхем-образцов. Кроме того, использование этих команд делает возможным блочное редактирование.

#### 7.2.3.1. **Save up** ( **Сохран. в конец** )

Эта команды осуществляет копирование рабочей области буфера в свободную область в его конце.

Перед началом перемещения задаются размер и номер блока, в который производится копирование. Размер блока может составлять не менее 1/128 и не более 1/2 части буфера.

Блок, используемый командами записи, чтения и сравнения, располагается в начальных адресах буфера и считается нулевым.

#### 7.2.3.2. **Download** ( **Загруз. в начало** )

Эта команды осуществляет копирование информации из какой-либо части буфера в его начало (в рабочую область), т.е. выполняет действия, обратные предыдущей команде.

#### 7.2.3.3. **Edit buffer** ( **Редактир. буфер** )

Команда служит для просмотра и редактирования (правки) данных в буфере программатора в десятичной, 16-ричной или символической форме.

В верхней строке дисплея отображаются адрес текущей ячейки и режим редактирования. В нижней строке выводятся данные из буфера, в т.ч. текущая (редактируемая ячейка), несколько ячеек, расположенных до и после текущей. Клавиши "вверх", "вниз", "влево", "вправо" используются как обычно при редактировании различных параметров. Клавиша "режим" осуществляет переключение между верхней и нижней строками дисплея. Для завершения правки следует удерживать в нажатом состоянии клавишу "влево" или "вправо".

#### 7.2.4. Меню **Macro commands** ( **Макросы** )

Меню команд работы с макросами, позволяющими существенно облегчить серийную работу, т.е. выполнение одной и той же последовательности операций с большим количеством микросхем. При создании макроса программатор запоминает последовательность команд (не более 15), выполняемых для одной микросхемы, а затем, при выполнении макроса, автоматически предлагает эти команды для выполнения в том же порядке, исключая необходимость их ручного выбора в меню. Программатор AutoProg запоминает до 8 макросов. Макросы хранятся в энергонезависимой памяти, т.е. не пропадают при выключении питания.

Описание команд для работы с макросами приводится ниже.

##### 7.2.4.1. **New Macro** ( **Начать запись** )

Эта команда служит для начала записи макроса, т.е. запоминания последовательности команд, выполняемых между **New Macro** ( **Начать запись** ) и **End Macro** ( **Окончить запись** ). Перед началом записи, будет предложено выбрать номер макроса.

##### 7.2.4.2. **End Macro** ( **Окончить запись** )

Эта команда служит для окончания записи макроса. Запомненная программатором последовательность команд сохраняется в энергонезависимой памяти.

##### 7.2.4.3. **Run Macro** ( **Запустить макрос** )

Команда осуществляют запуск макроса. В этом режиме, программатор автоматически предлагает для выполнения команды, записанные в макрос. Для выполнения предложенных команд следует нажимать "вправо". Нажатие других клавиш позволяет выбрать другую, вводную команду, после выполнения которой работа макроса будет продолжена.

Перед запуском макроса, будет предложено выбрать его номер. При попытке запустить макрос, созданный для микросхемы, набор команд которой отличается от набора команд текущей

выбранной микросхемы, запуск выполнен не будет, а на дисплей будет выведено сообщение **Unsuitable macro** ( **Макрос непригоден** ).

#### 7.2.4.4. **Stop Macro** ( **Прервать макрос** ) .

Команда используется для прекращения работы макроса.

#### 7.2.5. Меню **Settings&Service** ( **Настройки/сервис** ) .

Вход в подменю команд изменения настроек программатора др. общих операций.

##### 7.2.5.1. **Change Language** ( **Изменить язык** ) .

Изменение языка сообщений, выводимых на дисплей программатора (русский/английский).

##### 7.2.5.2. **Turn Key Sound** ( **Озвучив. клавиш** ) .

Включение/выключение озвучивания нажатия клавиш клавиатуры программатора.

##### 7.2.5.3. **Turn Event Sound** ( **Озвучив. событий** ) .

Включение/выключение звукового сопровождения завершения выполнения команд, вывода сообщений об ошибках и др. звуков, сопровождающих работу программатора (преимущественно в автономном режиме).

##### 7.2.5.4. **Model Info** ( **Информация** ) .

Отображение информации о текущей конфигурации программатора – версиях микропрограммы и базы данных, объеме буфера и др.

##### 7.2.5.5. **Check Database** ( **Пров.базу данных** ) .

Проверка целостности информации в базе данных программатора. При отсутствии ошибок выводится сообщение **No errors in database** ( **База данных без искажений** ). В случае появления сообщения **Database contains errors** ( **База данных искажена** ), следует произвести обновление базы

данных при помощи соответствующей функции, имеющейся в управляющей компьютерной программе.

### 7.2.6. Меню **Set voltages** ( **Установ. напряж.** )

Меню установки напряжений, используемых при работе с микросхемой. Начальные значения напряжений устанавливаются в соответствии с технической документацией на выбранную микросхему. Изменять эти значения следует лишь опытным пользователям, поскольку их неправильный выбор приводит, в большинстве случаев, к ненадежной записи информации или к выходу микросхемы из строя.

#### 7.2.6.1. **write/Read** ( **Запись/чтение** )

Установка напряжения питания микросхемы при выполнении команд записи и считывания информации, а также, напряжения программирования.

#### 7.2.6.2. **Verification** ( **Напряж. проверки** )

Установка значений напряжения питания при проверке микросхемы командой **Verify** ( **проверить** ).

## 7.3. Работа с ПЗУ различных типов.

Программатор AutoProg позволяет осуществлять перезапись (копирование) информации из микросхемы одного типа в микросхему другого типа. Для этого следует задать тип микросхемы образца, считать информацию в буфер, длительным нажатием клавиши "влево" перевести программатор в исходное состояние, задать тип микросхемы для записи и осуществить запись. Естественно, если объем микросхемы образца превышает объем прошиваемой ПЗУ, часть информации будет потеряна. Если, наоборот, образец меньше, он будет дополнен лишней информацией, находящейся в буфере.

## 8. Порядок работы под управлением персонального компьютера.

Программатор AutoProg комплектуется отдельным программным обеспечением для DOS и для Windows.

Удобный и доступный интерфейс управляющих программ и достаточно подробная справочная система Help позволяют не описывать процесс выполнения команд при работе под управлением компьютера в настоящей инструкции.

Для нормальной работы, версия используемого программного обеспечения должна соответствовать версиям микропрограммы контроллера и базы данных программатора. При обнаружении несоответствия версий, управляющая программа предложит осуществить замену (перепрограммирование) базы данных и микропрограммы. Этим можно воспользоваться для обновления программатора. Следует быть внимательным при замене версий базы данных и микропрограммы на более старые. Такая замена сузит возможности программатора. В этом случае, рекомендуется отказаться от замены и использовать подходящую версию управляющей программы.

При несоответствии версии управляющей программы версии микропрограммы контроллера программатора, перепрограммирование контроллера возможно только для программаторов с установленным модулем самообновления. В противном случае, перепрограммирование осуществляется специалистами ROMSERVICE.

## **9. Базовый комплект поставки.**

- 1) Программатор AutoProg.
- 2) Блок питания (сетевой адаптер).
- 3) Инструкция по эксплуатации.
- 4) Компакт-диск с программным обеспечением.
- 5) Кабель для подключения к ПК.
- 6) Кабель для внутрисхемного программирования.
- 7) Коробка упаковочная.

## **10. Гарантийные обязательства.**

Изготовитель гарантирует нормальную работу прибора в течение 12 месяцев с момента продажи потребителю, а также бесплатный ремонт в течении всего гарантийного срока при условии соблюдения правил пользования, установленных настоящей инструкцией.

Настоящая гарантия не распространяется на аккумуляторную батарею.

Заявки на гарантийный и послегарантийный ремонт принимаются по месту покупки прибора, по телефону или электронной почте.



## Приложение 1. Контрольные суммы "чистых" микросхем

**Контрольные суммы "чистых" микросхем EPROM,  
EEPROM, FLASH, Firmware HUB, LPC FLASH и их частей  
различного объема**

Объем, байт	Контр. Сумма	Объем, байт	Контр. Сумма
16	FFF0	32K	8000
32	FFE0	64K	0000
64	FFC0	128K	0000
128	FF80	256K	0000
256	FF00	512K	0000
512	FE00	1M	0000
1K	FC00	1,5M	0000
2K	F800	2M	0000
4K	F000	3M	0000
8K	E000	4M	0000
16K	C000	8M	0000

### Контрольные суммы "чистых" PIC-контроллеров

Тип микросхемы	Объем, слов	Контр. сумма
16C54, 16C55, 12C508, 12C518	512	0C00
16C56, 12C509, 12C519	1K	2800
16C57, 16C58	2K	6000
12C508(518)-Calibration	512-1	FDF2
12C509(519)-Calibration	1K-1	19F2
16C554, 16C620, 16C710, 16CR83, 16F83	512	3C00
16C556, 16C61/621, 16C71/711, 16C84/CR84, 16F84, 12C671/673, 16F627	1K	B800
16C558, 16C62/622/64, 16C72, 12C672/674, 16F628, 16F870/871/872	2K	B000
16C63/65, 16C73/74, 16C923/924, 14000, 16F73/74, 16F873/874	4K	A000
16C66/67, 16F76/77, 16F876/877	8K	8000
12C671/673-Calibration	1K-1	79C2
12C672/674-Calibration	2K-1	71C2
14000-Calibration	4K-64	D180

## Контрольные суммы "чистых" микроконтроллеров MCS-48, MCS-51, UPI-42 и AVR

Тип микросхемы	Память программ		Память данных	
	Объем, байт	Контр. сумма	Объем, байт	Контр. сумма
8048, 8748, 8741, K1816BE48	1K	0000		
8049, 8749, 8742, K1816BE49	2K	0000		
8050	4K	0000		
89C1051	1K	FC00		
89C2051	2K	F800		
8051, 87(C)51, 89C51, 89C4051 K1816/1830BE751	4K	F000		
87(C)52, 87(C)53, 87C51FA 89C52, 1830BE753	8K	E000		
89S8252	8K	E000	2K	F800
89S8252 целиком	8K	+	2K	D800
89S53	12K	D000		
87C54, 87C51FB	16K	C000		
89C55, 89C55WD	20K	B000		
87C58, 87C51FC, 89C51RC	32K	8000		
Tiny10/11	1K	FC00		
90S1200, Tiny12/15	1K	FC00	64	FFC0
Tiny28	2K	F800		
90S2313, 90S2323/43, 90S2333, Tiny22	2K	F800	128	FF80
90S4414, 90S4433, 90S4434	4K	F000	256	FF00
90S8515, 90S8535	8K	E000	512	FE00
Mega161/163	16K	C000	512	FE00
Mega103	128K	0000	4K	F000

Тип микросхемы	APROM		LDROM	
	Объем, байт	Контр. сумма	Объем, байт	Контр. сумма
W78E51, W78LE51	4K	F000		
W78E52, W78LE52, W78LE812	8K	E000		
W78E54, W78LE54	16K	C000		
W78E58	32K	8000		
W78LE58	32K	8000	4K	F000
W78E516, W78LE516	64K	0000	4K	F000

## Приложение 2. Поблочно стираемые FLASH.

Микросхема	Блоки, Кбайт (b – байт)
Am29F010	8x16
Am29F040	8x64
Am29F080	16x64
Am29F002B	16+8+8+32+64+64+64
Am29F002T	64+64+64+32+8+8+16
Am29F002NB	16+8+8+32+64+64+64
Am29F002NT	64+64+64+32+8+8+16
Am29F004B	16+8+8+32+7x64
Am29F004T	7x64+32+8+8+16
Am29F008BB	16+8+8+32+15x64
Am29F008BT	15x64+32+8+8+16
Am29F200BB	16+8+8+32+64+64+64
Am29F200BT	64+64+64+32+8+8+16
Am29F400BB	16+8+8+32+7x64
Am29F400BT	7x64+32+8+8+16
Am29F800BB	16+8+8+32+15x64
Am29F800BT	15x64+32+8+8+16
AT49F001	16+8+8+32+64
AT49F001N	16+8+8+32+64
AT49F001T	64+32+8+8+16
AT49F001NT	64+32+8+8+16
AT49F002	16+8+8+96+128
AT49F002N	16+8+8+96+128
AT49F002T	128+96+8+8+16
AT49F002NT	128+96+8+8+16
AT49F004	16+8+8+480
AT49F004T	480+8+8+16
AT49F008A	16+8+8+992
AT49F008AT	992+8+8+16
AT49F8011	16+32+4x8+32+16+14x64
AT49F8011T	14x64+16+32+4x8+32+16
SST39SF512	16x4
SST39SF010	32x4
SST39SF020	64x4
SST28SF040/040A	4096x256b
MX28F1000P	8x16 / 7x16+4x4
MX28F2000P	16x16 / 4x4+14x16+4x4
MX28F2000T	16x16 / 14x16+8x4
MX28F2100B	16+8+8+96+128
MX28F2100T	128+96+8+8+16
MX28F002B	16+8+8+96+128
MX28F002T	128+96+8+8+16
MX29F040	8x64
MX29F080	16x64
MX29F022B	16+8+8+32+64+64+64
MX29F022T	64+64+64+32+8+8+16
MX29F022NB	16+8+8+32+64+64+64
MX29F022NT	64+64+64+32+8+8+16
MX29F001B	8+4+4+8+8+32+64

Микросхема	Блоки, Кбайт (b – байт)
MX29F001T	64+32+8+8+4+4+8
MX29F002B	16+8+8+32+64+64+64
MX29F002T	64+64+64+32+8+8+16
MX29F002NB	16+8+8+32+64+64+64
MX29F002NT	64+64+64+32+8+8+16
MX29F004B	16+8+8+32+7x64
MX29F004T	7x64+32+8+8+16
MX29F100B	16+8+8+32+64
MX29F100T	64+32+8+8+16
MX29F200B	16+8+8+32+64+64+64
MX29F200T	64+64+64+32+8+8+16
MX29F400B	16+8+8+32+7x64
MX29F400T	7x64+32+8+8+16
MX29F800B	16+8+8+32+15x64
MX29F800T	15x64+32+8+8+16
V29C51000B	128x512b
V29C51000T	128x512b
V29LC51000	128x512b
V29C51001B	256x512b
V29C51001T	256x512b
V29LC51001	256x512b
V29C51002B	512x512b
V29C51002T	512x512b
V29LC51002	512x512b
V29C51004B	512x1
V29C51004T	512x1
V29C51400B	512x1
V29C51400T	512x1
28F001BX—B	8+4+4+112
28F001BX—T	112+4+4+8
28F002BX—B	16+8+8+96+128
28F002BX—T	128+96+8+8+16
28F002BC—T	128+96+8+8+16
28F200BX—B	16+8+8+96+128
28F200BX—T	128+96+8+8+16
28F004BX—B	16+8+8+96+128+128+128
28F004BX—T	128+128+128+96+8+8+16
28F400BX—B	16+8+8+96+128+128+128
28F400BX—T	128+128+128+96+8+8+16
M29F010B	8x16
M29F040	8x64
M29F040B	8x64
M29F080A	16x64
M29F002B	16+8+8+32+64+64+64
M29F002T	64+64+64+32+8+8+16
M29F002NT	64+64+64+32+8+8+16
M29F002BB	16+8+8+32+64+64+64
M29F002BT	64+64+64+32+8+8+16
M29F002BNT	64+64+64+32+8+8+16

Микросхема	Блоки, Кбайт (b – байт)
M29F008AB	16+8+8+32+15x64
M29F008AT	15x64+32+8+8+16
M29F100B	16+8+8+32+64
M29F100T	64+32+8+8+16
M29F100BB	16+8+8+32+64
M29F100BT	64+32+8+8+16
M29F200B	16+8+8+32+64+64+64
M29F200T	64+64+64+32+8+8+16
M29F200BB	16+8+8+32+64+64+64
M29F200BT	64+64+64+32+8+8+16
M29F400B	16+8+8+32+7x64
M29F400T	7x64+32+8+8+16
M29F400BB	16+8+8+32+7x64
M29F400BT	7x64+32+8+8+16
M29F800AB	16+8+8+32+15x64
M29F800AT	15x64+32+8+8+16
TMS28F002AxB	16+8+8+96+128
TMS28F002AxT	128+96+8+8+16
TMS28F200AxB	16+8+8+96+128
TMS28F200AxT	128+96+8+8+16
TMS28F004AxB	16+8+8+96+128+128+128
TMS28F004AxT	128+128+128+96+8+8+16
TMS28F400AxB	16+8+8+96+128+128+128
TMS28F400AxT	128+128+128+96+8+8+16
TMS29F010	8x16
TMS29F040	8x64
TMS29F002B	16+8+8+32+64+64+64
TMS29F002T	64+64+64+32+8+8+16
TMS29F002RB	16+8+8+32+64+64+64
TMS29F002RT	64+64+64+32+8+8+16
TMS29F008B	16+8+8+32+15x64
TMS29F008T	15x64+32+8+8+16
TMS29F400B	16+8+8+32+7x64
TMS29F400T	7x64+32+8+8+16
TMS29F800B	16+8+8+32+15x64
TMS29F800T	15x64+32+8+8+16
MBM29F040A	8x64
MBM29F080	16x64
MBM29F002B	16+8+8+32+64+64+64
MBM29F002T	64+64+64+32+8+8+16
MBM29F200BC	16+8+8+32+64+64+64
MBM29F200TC	64+64+64+32+8+8+16
MBM29F400BC	16+8+8+32+7x64
MBM29F400TC	7x64+32+8+8+16
MBM29F800BA	16+8+8+32+15x64
MBM29F800TA	15x64+32+8+8+16
M5M29KB800A	16+8+8+32+15x64
M5M29KT800A	15x64+32+8+8+16
W29D040C	8x64
W49F002	(16)+8+8+96+128
W49F002B	(16)+8+8+96+128
W49F002U	128+96+8+8+(16)
W49F002N	128+96+8+8+(16)
BM29F040	8x64

Микросхема	Блоки, Кбайт (b – байт)
BM29F400B	16+8+8+32+7x64
BM29F400T	7x64+32+8+8+16
CAT28F001BX—B	8+4+4+112
CAT28F001BX—T	112+4+4+8
CAT28F002BX—B	16+8+8+96+128
CAT28F002BX—T	128+96+8+8+16

### Приложение 3. Отображение значений битов конфигурации AVR-контроллеров.

Микросхема	Биты конфигурации (FUSE-биты)									
ATTiny22L	-	-	-	-	-	SPIEN	-	-	-	-
AT90S1200	-	-	-	-	-	SPIEN	-	-	-	-
AT90S2343	-	-	-	-	-	SPIEN	-	-	-	RCEN
ATTiny22	-	-	-	-	-	-	-	-	-	-
AT90S2313	-	-	-	-	-	-	-	-	-	-
AT90S4414	-	-	-	-	-	SPIEN	-	-	-	-
AT90S8515	-	-	-	-	-	SPIEN	-	-	-	-
AT90S4434	-	-	-	-	-	-	-	-	-	-
AT90S8535	-	-	-	-	-	-	-	-	-	FSTR
AT90S2343	-	-	-	-	-	-	-	-	-	-
AT90S2333	-	-	-	-	-	SPIEN	BODLEVEL	-	-	-
AT90S4433	-	-	-	-	-	SPIEN	BODLEVEL	BODEN	CKSEL2	CKSEL1
ATTiny10	-	-	-	-	-	SPIEN	BODLEVEL	BODEN	CKSEL2	CKSEL1
ATTiny11	-	-	-	-	-	SPIEN	BODLEVEL	BODEN	CKSEL2	CKSEL1
ATTiny12	-	-	-	-	BODLEVEL	BODEN	SPIEN	RSTDISBL	CKSEL3	CKSEL2
ATTiny15	-	-	-	-	BODLEVEL	BODEN	SPIEN	RSTDISBL	-	CKSEL2
ATTiny15	-	-	-	-	BODLEVEL	BODEN	SPIEN	RSTDISBL	-	CKSEL2
ATTiny15	-	-	-	-	BODLEVEL	BODEN	SPIEN	RSTDISBL	-	CKSEL2
ATTiny28	-	-	-	-	-	-	INTCAP	CKSEL3	CKSEL2	CKSEL1
ATTiny28	-	-	-	-	-	-	INTCAP	CKSEL3	CKSEL2	CKSEL1
ATTiny28	-	-	-	-	-	-	INTCAP	CKSEL3	CKSEL2	CKSEL1
ATTiny28	-	-	-	-	-	-	INTCAP	CKSEL3	CKSEL2	CKSEL1
ATMega103	-	-	-	-	-	SPIEN	-	EESAVE	-	SUT1
ATMega103	-	-	-	-	-	SPIEN	-	EESAVE	-	SUT1
ATMega103	-	-	-	-	-	SPIEN	-	EESAVE	-	SUT1
ATMega103	-	-	-	-	-	SPIEN	-	EESAVE	-	SUT1
ATMega161	-	-	-	-	BOOTRST	SPIEN	BODLEVEL	BODEN	CKSEL2	CKSEL1
ATMega161	-	-	-	-	BOOTRST	SPIEN	BODLEVEL	BODEN	CKSEL2	CKSEL1
ATMega161	-	-	-	-	BOOTRST	SPIEN	BODLEVEL	BODEN	CKSEL2	CKSEL1
ATMega161	-	-	-	-	BOOTRST	SPIEN	BODLEVEL	BODEN	CKSEL2	CKSEL1
ATMega163	BOOTSZ1	BOOTSZ0	BOOTRST	BODLEVEL	BODEN	SPIEN	-	CKSEL3	CKSEL2	CKSEL1
ATMega163	BOOTSZ1	BOOTSZ0	BOOTRST	BODLEVEL	BODEN	SPIEN	-	CKSEL3	CKSEL2	CKSEL1
ATMega163	BOOTSZ1	BOOTSZ0	BOOTRST	BODLEVEL	BODEN	SPIEN	-	CKSEL3	CKSEL2	CKSEL1
ATMega163	BOOTSZ1	BOOTSZ0	BOOTRST	BODLEVEL	BODEN	SPIEN	-	CKSEL3	CKSEL2	CKSEL1

**Приложение 4.**  
**Распайка интерфейсного кабеля.**

<b>Программатор 9 pin</b>		<b>COM1/2/3/4 (9 pin)</b>	<b>COM1/2/3/4 (25 pin)</b>
2	↔	3	2
3	↔	2	3
5	↔	5	7

## **Приложение 5.**

### **Возможные проблемы при работе с программатором и способы их решения.**

*1. Не удается выбрать определенный тип микросхемы. Выводится сообщение "Ошибка конфигурации".*

Возможная причина – искажение информации в базе данных программатора. Выполните проверку базы данных соответствующей командой управляющей программы или командой автономного режима. При подтверждении искажений, произведите обновление (замену) базы данных из окна обновлений управляющей программы.

*1. Неустойчивое считывание информации из ПЗУ.*

Если повторные подсчеты контрольной суммы дают различные результаты при заведомо исправной ПЗУ, то причиной может являться плохой контакт микросхемы с панелью программатора вследствие окисления или загрязнения выводов.

К улучшению контакта приводит сдвиг микросхемы в закрытой панели.

*2. Потеря связи компьютера с программатором (1).*

К подобной проблеме при использовании DOS-программы, обычно приводит недостаточное быстроедействие компьютера при высокой скорости обмена.

Уменьшить вероятность сбоев можно исключив перемещение "мышки" и нажатие кнопок клавиатуры компьютера во время выполнения команд или уменьшив скорость обмена.

*3. Потеря связи компьютера с программатором (2).*

При работе в Windows с использованием DOS-программы, для исключения сбоев может потребоваться установка движка "Приоритет при ожидании" (в свойствах программы, на закладке "разное") в положение "Высокий".

