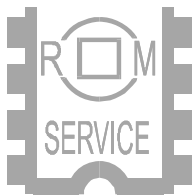


ROMSERVICE

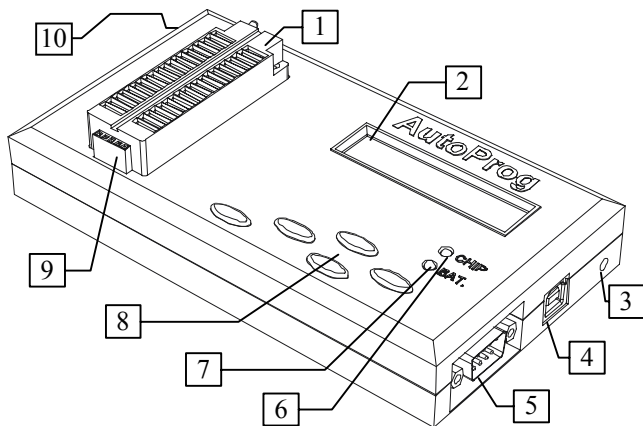


Универсальный программатор

AutoProg SE

Инструкция по эксплуатации

Москва 2008



1. ZIF-панель для установки микросхем
2. ЖК-дисплей
3. Разъем питания
4. Разъем интерфейсный USB
5. Разъем интерфейсный RS-232 (COM)
6. Светодиод "CHIP"
7. Светодиод "BAT"
8. Клавиатура 5-кнопочная
9. Разъем расширения и внутрисхемного программирования
10. Скрытая кнопка принудительного запуска загрузчика

Содержание.

1. Назначение прибора.....	8
2. Устройство прибора.....	8
3. Обновления.....	11
4. Работа с автономным питанием.....	12
5. Основы работы с энергонезависимой памятью. Особенности микросхем различных семейств.....	13
5.1. Общие сведения.....	13
5.2. EPROM.....	15
5.3. FLASH.....	16
5.4. EEPROM.....	18
5.5. Serial EEPROM (SEEPROM).....	18
5.6. Serial FLASH.....	21
5.7. Firmware HUB и LPC FLASH.....	22
5.8. NAND FLASH.....	22
5.9. Микроконтроллеры семейств MCS-48, UPI-42.....	23
5.10. Микроконтроллеры (ОМЭВМ) семейства MCS-51.....	23
5.11. PIC-контроллеры.....	26
5.12. AVR-контроллеры.....	28
6. Подготовка к работе.....	30
6.1. Общие указания.....	30
6.2. Выбор типа и установка ПЗУ в программатор.....	31
7. Порядок работы в автономном режиме.....	32
7.1. Общие указания.....	32
7.2. Описание команд.....	35
7.2.1. Меню Main commands (Основные команды).....	35
7.2.1.1. Write Chip (Записать м/схему).....	35
7.2.1.2. Chip Check Sum (К. Сумма м/схемы).....	35
7.2.1.3. Buffer Check Sum (К. Сумма буфера).....	35
7.2.1.4. Read Chip (Считать м/схему).....	36
7.2.1.5. Erase Chip (Стереть м/схему).....	36
7.2.1.6. Test Chip (Проверить м/сх.).....	36
7.2.2. Меню Addit. commands (Дополнит.команды).....	36
7.2.2.1. Compare (Сравнить).....	36
7.2.2.2. Verify (Проверить).....	37
7.2.2.3. Blank test (Провер. чистоту).....	37
7.2.2.4. Erase block (Стереть блок).....	37
7.2.2.5. Erase mini block (Стереть миниблок).....	37

7.2.2.6.	Erase Part	(Стереть часть)37
7.2.2.7.	Erase MainMemory	(Стер. осн.память)38
7.2.2.8.	Erase Main Block	(Стереть глав.бл.)38
7.2.2.9.	Erase Add.Memory	(Стер. доп.память)38
7.2.2.10.	Write & SDP off	(Записать без SDP)39
7.2.2.11.	Fast write Chip	(Быстрая запись)39
7.2.2.12.	Set Lock Bit	(Устан.бит защиты)39
7.2.2.13.	Set Lock Bit 1	(Устан.бит защ. 1)39
7.2.2.14.	Set Lock Bit 2	(Устан.бит защ. 2)39
7.2.2.15.	Set Lock Bit 3	(Устан.бит защ. 3)39
7.2.2.16.	Write Code Table	(Запис.шифр.табл.)40
7.2.2.17.	Read Prot. Reg.	(Считать рег.защ.)40
7.2.2.18.	Write Prot. Reg.	(Запис. рег.защ.)40
7.2.2.19.	Edit Prot. Reg.	(Измен. рег.защ.)40
7.2.2.20.	Clear Prot. Reg.	(Очист. рег.защ.)40
7.2.2.21.	Lock Prot. Reg.	(Блокир. рег.защ.)40
7.2.2.22.	Read Status Reg.	(Счит.рег.статуса)40
7.2.2.23.	Write Status Reg	(Зап. рег.статуса)41
7.2.2.24.	Edit Status Bits	(Изм.биты статуса)41
7.2.2.25.	Read Option Reg.	(Счит. рег.опций)41
7.2.2.26.	Write Option Reg	(Запис. рег.опций)41
7.2.2.27.	Edit Option Bits	(Измен.биты опций)41
7.2.2.28.	Read Configurat.	(Считать конфиг.)41
7.2.2.29.	Write NEB posit.	(Устан.полож. НЕВ)42
7.2.2.30.	Write Prot. Reg.	(Установ. защиту)42
7.2.2.31.	Edit NEB & Prot.	(Измен.защ. и НЕВ)42
7.2.2.32.	Protect chip	(Защитить м/схему)42
7.2.2.33.	Protect Block	(Защитить блок)42
7.2.2.34.	Unprotect All	(Снять защ.блоков)43
7.2.2.35.	Read Lock Bits	(Счит. биты защ.)43
7.2.2.36.	Read Lock/SPIEN	(Счит. защ./SPIEN)43
7.2.2.37.	Read Lock/X2mode	(Счит.защ./X2mode)43
7.2.2.38.	Read Lock/Fuses	(Счит. защ./конф.)43
7.2.2.39.	Write Lock Bits	(Запис. Биты защ.)43

7.2.2.40.	Write SPIEN Fuse	(Запис. бит SPIEN)	43
7.2.2.41.	Write Fuse Bits	(Запис. биты конф.)	44
7.2.2.42.	Write Fuse&Lock	(Зап. конф. и защ.)	44
7.2.2.43.	Special Erase	(Спец. стирание)	44
7.2.2.44.	View Lock Bits	(Просм. биты защ.)	44
7.2.2.45.	Edit Lock Bits	(Измен. биты защ.)	44
7.2.2.46.	Edit SPIEN Fuse	(Измен. бит SPIEN)	45
7.2.2.47.	Edit Fuse Bits	(Измен. биты конф.)	45
7.2.2.48.	Edit Ext. Fuse B.	(Изм. доп. б. конф.)	45
7.2.2.49.	Read SEED	(Читать SEED)	45
7.2.2.50.	Write SEED	(Записать SEED)	45
7.2.2.51.	Edit SEED	(Изменить SEED)	46
7.2.2.52.	Read Sbyte&BVect	(Чит. Sbyte&BVect)	46
7.2.2.53.	Erase Sbyte&BVec	(Стер. Sbyte&BVect)	46
7.2.2.54.	Write StatusByte	(Запис. StatusByte)	46
7.2.2.55.	Write BootVector	(Запис. BootVector)	46
7.2.2.56.	Edit Status Byte	(Измен. StatusByte)	46
7.2.2.57.	Edit Boot Vector	(Измен. BootVector)	46
7.2.2.58.	View X2 Status	(Просм. сост. X2)	47
7.2.2.59.	Set X2 Mode Off	(Выкл. Режим X2)	47
7.2.2.60.	Set X2 Mode On	(Вкл. Режим X2)	47
7.2.2.61.	Read CFG & ID	(Читать CFG&ID)	47
7.2.2.62.	Write CFG & ID	(Записать CFG&ID)	47
7.2.2.63.	Edit ID-code	(Изменить ID-код)	47
7.2.2.64.	Edit Config.word	(Измен. слово конф)	47
7.2.2.65.	Read Cfg words	(Читать конфиг.)	48
7.2.2.66.	Read ID-code	(Читать ID-код)	48
7.2.2.67.	Write Cfg words	(Записать конфиг.)	48
7.2.2.68.	Write ID-code	(Записать ID-код)	48
7.2.2.69.	Edit ID-code	(Изменить ID-код)	48
7.2.2.70.	Edit Config.word	(Измен. слово конф)	48
7.2.2.71.	Edit Conf.word 2	(Изм. слово конф. 2)	49
7.2.2.72.	Edit Cfg words	(Изменить конфиг.)	49
7.2.2.73.	Set Calibration	(Устан. калибровку)	49

7.2.2.74.	Read Reset Pol.	(Счит. полярность)	49
7.2.2.75.	Write Reset Pol.	(Запис.полярность)	49
7.2.2.76.	Edit Reset Pol.	(Измен.полярность)	50
7.2.2.77.	Enable DCLK	(Разрешить DCLK)	50
7.2.2.78.	Disable DCLK	(Запретить DCLK)	50
7.2.2.79.	Enable AutoStore	(Разреш.автосохр.)	50
7.2.2.80.	DisableAutoStore	(Запрет.автосохр.)	50
7.2.2.81.	ISP mode	(Progr. в системе)	50
7.2.2.82.	Socket Mode	(Progr. в панели)	50
7.2.2.83.	Sel. Data Memory	(Выбр. пам. данных)	51
7.2.2.84.	Sel. Progr. Memory	(Выбр. пам. прогр.)	51
7.2.2.85.	Switch To LDROM	(Перекл. в LDROM)	51
7.2.2.86.	Switch To APROM	(Перекл. в APROM)	51
7.2.2.87.	Select Add. Mem.	(Выбр. доп.память)	51
7.2.2.88.	Select Main Mem.	(Выбр. осн.память)	51
7.2.2.89.	Select Prot.Area	(Выбр. обл.защиты)	51
7.2.2.90.	Set slave addr.	(Уст.схемн.адрес)	51
7.2.2.91.	Set Memory Part	(Выбр. часть пам.)	52
7.2.3. Меню	Buffer commands	(Операции буфера)	52
7.2.3.1.	Save Up	(Сохран. в конец)	52
7.2.3.2.	Download	(Загруз.в начало)	53
7.2.3.3.	Edit Buffer	(Редактир. буфер)	53
7.2.4. Меню	Macro commands	(Макросы)	53
7.2.4.1.	New Macro	(Начать запись)	54
7.2.4.2.	End Macro	(Окончить запись)	54
7.2.4.3.	Run Macro	(Запустить макрос)	54
7.2.4.4.	Stop Macro	(Прервать макрос)	54
7.2.5. Меню	Settings&Service	(Настройки/сервис)	54
7.2.5.1.	Change Language	(Изменить язык)	54
7.2.5.2.	Turn Key Sound	(Озвучив. Клавиш)	55
7.2.5.3.	Turn Event Sound	(Озвучив. событий)	55
7.2.5.4.	LCD Light Off	(Отключ.подсветки)	55
7.2.5.5.	Auto_Power_Off	(Автовывключение)	55
7.2.6. Меню	Set voltages	(Установ. напряж.)	55

7.2.6.1.	Write/Read	(Запись/чтение)	56
7.2.6.2.	Verification	(Напряж. проверки)	56
7.2.7.	Новые команды.				56
7.3.	Работа с ПЗУ различных типов.				56
8.	Порядок работы под управлением персонального компьютера.				56
9.	Базовый комплект поставки.				58
10.	Гарантийные обязательства.				58
11.	Возможные проблемы при работе с программатором и способы их решения.				59

1. Назначение прибора.

Универсальный программатор AutoProg SE 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 персонального компьютера (далее ПК).

Функциональные возможности и технические характеристики программатора AutoProg SE 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 (600 mil) и узких (300 mil) 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 фиксации рычаг переводится в горизонтальное положение.

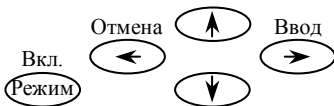
Микросхемы в корпусе DIP, имеющие более 40 выводов, а также выполненные в других видах корпусов (PLCC, SOIC, TSOP, QFP, PSOP, SSOP, BGA и др.) устанавливаются в панель через соответствующие адаптеры (переходники), которые могут быть

приобретены отдельно или изготовлены самостоятельно. При этом, распайка адаптеров 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 адаптеров, а во многих случаях, делает возможным использование стандартных широко распространенных адаптеров.

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

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. Для увеличения контрастности и более комфортной работы при недостаточном освещении, дисплей снабжен подсветкой. Для экономии электроэнергии, что особенно важно при работе с автономным питанием, предусмотрена функция автоматического отключения подсветки при паузах в работе (отсутствии нажатия клавиш). Сообщения на дисплее выводятся на английском или русском языках. Переключение языка и управление продолжительностью включения подсветки осуществляется командами 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.

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 "Работа с автономным питанием".

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 бортовой сети. Внутренний контакт разъема питания – "+", внешний – "-".

Для подключения к порту USB компьютера, используется стандартный соединительный кабель "USB A-B".

Для подключения к порту COM компьютера, используется соединительный кабель, включаемый в комплект поставки по требованию. При необходимости изготовления дополнительного

кабеля (например, большей длины) следует пользоваться таблицей распайки, приведенной ниже.

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

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

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 микросхем управляющей программы "Встроенный буфер программатора".

3. Обновления.

Конструкция программатора AutoProg SE предусматривает возможность обновления его внутреннего программного обеспечения (ПО) с целью расширения его функциональных возможностей, в том числе добавления поддержки работы с новыми типами микросхем, а также установку аккумуляторной батареи и изменение объема памяти буфера путем замены модуля памяти (замене подлежит правый модуль, левый – служебный).

К обновляемым модулям внутреннего ПО относятся микропрограмма главного контроллера и база данных. Обновление осуществляется под управлением управляющей компьютерной программы и предлагается автоматически при несоответствии версий управляющей программы и внутреннего ПО программатора. "Обратное" обновление (замена более новой

версии на более старую) также допускается, но не рекомендуется, т.к. уменьшает функциональные возможности программатора. Новые версии управляющей программы размещаются, по мере их выпуска, на интернет-сайте производителя, в разделе "Техподдержка".

В случае возникновения сбоев в процессе обновления, например, при разрыве соединения или отключении электропитания, может потребоваться специальная восстановительная процедура, включающая следующие действия:

- Закрыть управляющую программу.
- Включить питание программатора, удерживая нажатой скрытую кнопку, расположенную за миниатюрным отверстием на нижней стороне корпуса программатора, при помощи длинного и тонкого (но не острого!) предмета, например, разогнутой канцелярской скрепки.
- Запустить восстановительную утилиту, которая осуществит запись в программатор реанимационной версии ПО.
- Запустить управляющую программу и выполнить обычное обновление программатора до текущей рабочей версии.

4. Работа с автономным питанием.

Конструкция программатора "AutoProg SE" 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.

Зарядка аккумулятора осуществляется автоматически при подключении программатора к источнику внешнего электропитания. Режим зарядки включается при остаточном заряде батареи менее 60...70% и выключается автоматически при достижении полного заряда. Зарядка может производиться в

процессе работы с программатором (обычный режим) или при выключенном программаторе (ускоренный режим).

Состояние аккумулятора отображается светодиодным индикатором, помеченным надписью "BAT.". Непрерывное зеленое свечение этого индикатора соответствует режиму зарядки аккумулятора, мигающее – состоянию окончания зарядки. 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 программатора, или искажения информации в ней, рекомендуется как можно быстрее прекратить работу с программатором, при необходимости, прервав выполнение текущей операции, удалить микросхему из панели, затем оставить программатор включенным до автоматического выключения.

Время полной зарядки батареи в обычном режиме составляет 15...25 часов, в ускоренном – около 2...3 часа. Время непрерывной работы программатора от полностью заряженного аккумулятора составляет примерно 3...7 часов, в зависимости от номинальной емкости аккумулятора, степени его износа, интенсивности работы с программатором и типов программируемых 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 программатор 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 (программирование) осуществлялась, либо в процессе изготовления кристалла, либо перед установкой в аппаратуру с помощью программатора. 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 обрастанию 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 (УФ), 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 их производителями через интернет-сайты, а также на компакт-дисках и в виде печатных 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 ввиду, 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 SE 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 SE 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 SE 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 режимах. 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 SE 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 I²C (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), I²C/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 SE использует 16-разрядный режим, данные в буфере программатора или файле размещаются старшим байтом вперед, что соответствует внутреннему автоматическому перераспределению данных в этих микросхемах при переключении разрядности. Каждой 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.

SEEPROM с шиной I²C допускают возможность внутрисхемного программирования. Такой режим доступен в программаторе AutoProg SE для микросхем, у которых в качестве производителя указано "Standard SEEPROM". При включении этого режима, сигналы вырабатываются не на панели программатора, а на 5-контактном разъеме внутрисхемного программирования. Сигналы на этом разъеме расположены в следующем порядке (слева направо): -, SCL, GND, SDA, -. При установке в схему эти микросхемы получают схемный адрес (Slave address), определяемый комбинацией сигналов на ее адресных входах, что дает возможность подключать несколько микросхем к одной общей шине I²C. В режиме внутрисхемного программирования программатор AutoProg SE позволяет осуществлять выбор этого адреса для обращения к конкретной микросхеме. **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.

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

SEEPROM 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).

SEEPROM с шиной I²C (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 SEEPROМ 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 SE, 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 микросхемы серий 25 и 45, использующие шину SPI.

Многие Serial FLASH обладают свойствами SEEPROM с шиной SPI серии 25, в т.ч. в части защиты от случайной перезаписи (см. выше), 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 или контрольная информация.

При работе с сериями AT45D и AT45DB программатор AutoProg SE, как и большинство других программаторов,

осуществляет раздельную работу с основной и дополнительной памятью. При этом информация из всех станиц 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 SOIC28. Это делает возможным, при работе с ними, использование стандартных переходников DIP28>>SOIC28 с таблицей соединений (распайкой) "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 PLCC32, что делает возможным применение стандартных широко распространенных переходников DIP32>>PLCC32, предназначенных для работы с FLASH-памятью в 32-выводном корпусе PLCC.

5.8. NAND FLASH.

Микросхемы NAND Flash, строго говоря, не рассчитаны на запись и считывание программаторами. Это Flash-память большого и очень большого объема, изготовленная по удешевленной технологии, предназначенная для применения в устройствах массового хранения данных – Flash-дисках, картах памяти и т.п. Однако, возможность считывания этих микросхем и перезаписи их отдельных блоков может быть полезна в некоторых

случаях, например, при восстановлении данных из неисправных Flash-дисков. Поэтому, по просьбам пользователей, они были включены в список микросхем, поддерживаемых программатором AutoProg SE. NAND Flash обладают специфическими особенностями, к которым относятся большой объем памяти и наличие неисправных ячеек (следствие упрощенной технологии изготовления), для чего память в этих микросхемах разбита на объединенные в блоки страницы двоично некруглого размера, т.к. каждая страница содержит дополнительные ячейки, в которых может размещаться корректирующая информация, позволяющая обнаруживать и исправлять ошибки данных в неисправных ячейках. Вследствие этих особенностей, работа с NAND Flash отличается от работы с другими микросхемами памяти – их считывание и запись возможны только под управлением компьютера, целым количеством блоков, напрямую из файла / в файл, минуя редактор.

5.9. 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.

5.10. 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, AT89, P89) 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 SE 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.

Современные микроконтроллеры семейства MCS-51 содержат ряд дополнительных программируемых ячеек памяти – биты конфигурации, статус-байт, бут-вектор и др.

Серия AT89S/LS предусматривает дополнительный режим программирования по последовательному интерфейсу SPI (в т.ч. внутрисхемно 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 осуществляются 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.

5.11. 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-, 14- или 16-разрядную организацию и содержит память программ, ID-область и одно или несколько слов конфигурации. Каждое слово, соответствующее ячейке памяти микросхемы, 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, в которой значащими являются только 8 младших бит (младший байт) слова, 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 14-разрядной памятью – сначала программа, ID-область по адресам 4000h-4007h, слово конфигурации по адресу 400Eh-400Fh;
- память данных размещается, начиная с адреса 4200h;

- для контроллеров серии PIC18 с 16-разрядной памятью – программа, начиная с адреса 000000h, ID-область с адреса 200000h, слова конфигурации с адреса 300000h, память данных с адреса F00000h.

Распределение адресов для блоков информации, предназначенных для записи в различные области памяти в HEX-файлах, получаемых при компиляции программ для микроконтроллеров PIC18, делает неудобным прямое преобразование таких файлов в простой двоичный формат – размер файла получается слишком большим. Загрузку HEX-файла при работе с этими микросхемами следует осуществлять при помощи команды "Загрузить PIC18 HEX-файл", а не стандартной команды «Импорт» меню «Файл». Данная команда открывает окно проводника для выбора HEX-файла, создает в окне редактора файлов новый файл и помещает в него информацию из загружаемого HEX-файла. При выполнении команды выводится информационное окно, содержащее сведения о наличии и объеме данных, предназначенных для тех или иных областей памяти микроконтроллера, и адреса размещения этих данных в созданном двоичном файле. Размер создаваемого двоичного файла и адреса размещения информации в нем зависят от типа микроконтроллера, выбранного в меню микросхем. Поэтому загрузку HEX-файла следует выполнять после выбора типа микросхемы в меню.

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 своей маркировке 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.

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

Программирование подавляющего большинства PIС-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 PIС-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.***

5.12. AVR-контроллеры

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

Память AVR-контроллеров состоит из 16-разрядной FLASH-памяти программ (1...128 Кбайт), 8-разрядной EEPROM-памяти данных (64...4096 байт), нескольких битов защиты информации и нескольких битов конфигурации (Fuse bits).

Некоторые микросхемы, например ATTiny10/11/28, не имеют EEPROM-памяти данных.

Назначение отдельных битов защиты и конфигурации в разных микросхемах различно и подробно описано в документации на них. Некоторые микросхемы AVR, имеющие возможность тактирования от внутреннего RC-генератора, содержат до четырех служебных ячеек, в которых хранятся записанные производителем числа – калибровочные байты. Значения этих байтов 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", ранее называемый "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 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.**

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 прибора. Для установки программы, запустите файл setup.exe, далее следуйте указаниям инсталлятора. Подключение программатора к ПК осуществляется через последовательный интерфейс USB или RS-232 (COM-порт). При использовании для подключения порта USB, потребуется установка драйвера с прилагаемого компакт-диска, которая будет предложена операционной системой компьютера при первом подключении программатора.

Копии файлов ПО и драйверов, а также их новые версии, можно скачать с интернет-сайта ROMSERVICE. Они размещаются в разделе "Техподдержка".

Выбор порта подключения и параметров соединения осуществляются в окне настроек управляющей программы, вызываемом одновременным нажатием клавиш "Alt+F9" на клавиатуре ПК или из меню.

После включения программатора (длительным нажатием клавиши «Режим»), 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**).

В исходном состоянии после включения питания продолжительное (более 3 секунд) удержание в нажатом состоянии любой клавиши, кроме клавиши "Режим", вызывает вход

в сервисный режим, в котором осуществляется самодиагностика программатора и вывод на индикатор информации об объемах внутренней памяти и версиях компонентов внутреннего ПО.

6.2. Выбор типа и установка ПЗУ в программатор.

Выбор типа микросхемы для работы осуществляется при помощи меню-списка, отображаемого 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 и могут отличаться от маркировки на корпусе микросхемы, которая может быть сокращенной (особенно для миниатюрных корпусов) или, напротив, содержать в конце маркировки дополнительные символы, обозначающие тип корпуса, быстродействие и допустимый температурный диапазон. У некоторых микросхем дополнительные символы располагаются не в конце маркировки, а в начале или внутри названия. В этом случае, они заменяются в меню символом " * ". Микросхемы в меню расположены в алфавитном порядке, без учета буквенно-цифрового префикса – нескольких (до трех) символов, в большинстве случаев, обозначающих фирму-производителя. Префикс может вообще отсутствовать.

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.

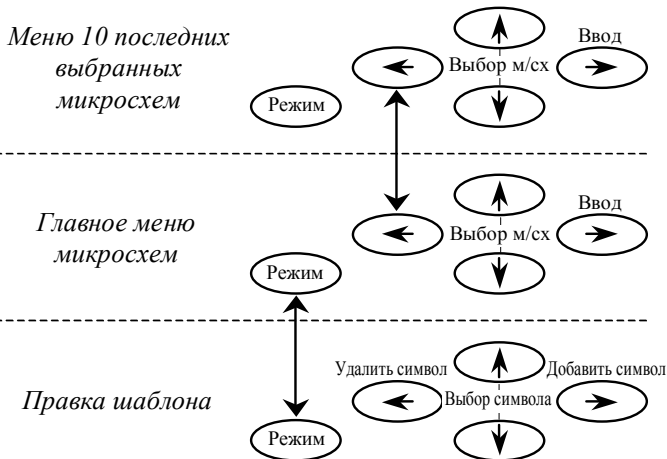
Установку микросхемы в панель программатора следует производить после выбора ее типа в меню и подтверждения ввода (во избежание ее повреждения или искажения информации в ней при смене конфигурации сигналов на панели программатора в момент выбора). После этого момента, питание и управляющие сигналы подаются 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 использовавшихся 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 используемые 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 и сервисные операции.

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 рабочей 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 Microsip, 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. 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.

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 считывание информации из микросхемы в буфер программатора для последующей записи в другую микросхему или для 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, 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. **Erase Mini Block** (**Стереть миниблок**).

Стирание маленького блока для микросхем, допускающих поблочное стирание блоками меньшего и большего объема. 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**).

7.2.2.6. **Erase Part** (**Стереть часть**).

Стирание части микросхемы – идущих последовательно блоков или страниц для микросхем с поблочным или постраничным стиранием, соответственно. Однократное выполнение команды обеспечивает стирание блоков (страниц),

входящих в часть памяти, выбранную 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**).

7.2.2.7. **Erase MainMemory** (**Стер. осн.память**).

Стирание основного блока памяти в микросхемах Flash, имеющих функцию однократной защиты boot-блока, после установки которой команда общего стирания микросхемы не может быть выполнена.

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

7.2.2.8. **Erase Main Block** (**Стереть глав.бл.**).

Стирание основного блока памяти в микросхемах Serial Flash, имеющих основной и дополнительный блоки памяти.

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

7.2.2.9. **Erase Add.Memory** (**Стер. доп.память**).

Стирание дополнительного блока памяти в микросхемах Serial Flash, имеющих основной и дополнительный блоки памяти.

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

7.2.2.10. write & SDP off (**Записать без SDP**).

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

В остальном, действие команды аналогично **write chip** (**Записать м/схему**).

7.2.2.11. Fast write Chip (**Быстрая запись**).

Эта команда применяется для микросхем, имеющих возможность записи с автоматическим стиранием и без такового. Команда осуществляет запись без автоматического стирания, что позволяет значительно сократить время, требуемое для записи чистых (новых или предварительно стертых) микросхем.

В остальном действие команды аналогично **write chip** (**Записать м/схему**).

7.2.2.12. Set Lock bit (**Устан.бит защиты**).

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

7.2.2.13. 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.

7.2.2.14. 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.

7.2.2.15. 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.

7.2.2.16. Write Code Table (Запис.шифр.табл.)

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

7.2.2.17. 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.

7.2.2.18. 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.)**

7.2.2.19. 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.

7.2.2.20. Clear Prot. Reg. (Очист. рег.заш.)

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

7.2.2.21. 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.

7.2.2.22. Read Status Reg. (Счит.рег.статуса)

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

7.2.2.23. 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)**.

7.2.2.24. 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.

7.2.2.25. 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.

7.2.2.26. 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)**.

7.2.2.27. 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.

7.2.2.28. 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).

7.2.2.29. **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**).

7.2.2.30. **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**).

7.2.2.31. **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.

7.2.2.32. **Protect Chip** (**Защитить м/схему**).

Команда осуществляет установку защиты на микросхемы SEEPROM I²C, оснащенные такой защитой. После выполнения команды, часть памяти в микросхеме не сможет быть перезаписана. Защита устанавливается однократно, отключение (снятие) защиты невозможно.

7.2.2.33. **Protect Block** (**Защитить блок**).

Команда осуществляет установку защиты бока микросхемы Flash-памяти. После выполнения команды, данные в защищенном блоке не могут быть стерты и перезаписаны.

7.2.2.34. Unprotect All (Снять защ.блоков).

Команда осуществляет снятие защиты со всех защищенных блоков микросхемы Flash-памяти.

7.2.2.35. Read Lock Bits (Счит. биты защ.).

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

7.2.2.36. 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.

7.2.2.37. Read Lock/x2mode (Счит. защ./x2mode).

Чтение состояния битов защиты и бита конфигурации X2 микроконтроллеров серии P89 и их запоминание для последующего просмотра или записи в другую микросхему.

7.2.2.38. 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.

7.2.2.39. 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.**).

7.2.2.40. 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**).

7.2.2.41. **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.**).

7.2.2.42. **Write Fuse&Lock** (**Зап.конф.и защ.**).

Команда осуществляет запись конфигурационных битов AVR-контроллеров, а затем битов защиты, в соответствии со значениями, считанными ранее из другой микросхемы, или подготовленными командами **Edit Fuse Bits** (**Измен.биты конф.**) и **Edit Lock Bits** (**Измен. биты защ.**), без выключения питания микросхемы. Используется, когда выбранная конфигурация меняет функции выводов микросхемы, используемых при программировании и делает последующее программирование невозможным.

7.2.2.43. **Special Erase** (**Спец. стирание**).

Команда служит для восстановления возможности программирования AVR-контроллеров после записи в них конфигурационных битов, когда выбранная конфигурация меняет функции выводов микросхемы, используемых при программировании и делает последующее программирование невозможным.

7.2.2.44. **View Lock Bits** (**Просм. Биты защ.**).

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

7.2.2.45. **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.

7.2.2.46. 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.

7.2.2.47. 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 их физического расположения в микросхеме.

7.2.2.48. Edit Ext.Fuse B. (**Изм.доп.б.конф.**).

Команда используется для просмотра значений конфигурационных битов, составляющих дополнительный конфигурационный байт (Extended Fuse Byte) AVR-контроллеров, а также для ввода значений этих битов для записи в микросхему. Отображение осуществляется в двоичном виде, в порядке их физического расположения в микросхеме.

7.2.2.49. Read SEED (**Считать SEED**).

Команда производит считывание информации из ячеек шифрования SEED микроконтроллеров серии W77 и ее запоминание для последующего просмотра или записи в другую микросхему.

7.2.2.50. Write SEED (**Записать SEED**).

Команда производит запись ячеек шифрования SEED микроконтроллеров серии W77, значениями, ранее считанными из другой микросхемы, или подготовленными командой

Edit SEED (**Изменить SEED**).

7.2.2.51. **Edit SEED** (**Изменить SEED**).

Команда используется для просмотра значений, считанных из ячеек шифрования SEED микроконтроллеров серии W77, а также для изменения этих значений для их последующей записи в микросхему.

7.2.2.52. **Read Sbyte&BVec** (**Счит. Sbyte&BVec**).

Команда производит считывание значений статус-байта и бут-вектора микроконтроллеров и их запоминание для последующего просмотра или записи в другую микросхему.

7.2.2.53. **Erase Sbyte&BVec** (**Стер. Sbyte&BVec**).

Команда производит стирание статус-байта и бут-вектора.

7.2.2.54. **Write StatusByte** (**Запис. StatusByte**).

Команда производит запись статус-байта, значением, ранее считанным из другой микросхемы, или подготовленным командой **Edit Status Byte** (**Измен. StatusByte**).

7.2.2.55. **Write BootVector** (**Запис. BootVector**).

Команда производит запись бут-вектора, значением, ранее считанным из другой микросхемы, или подготовленным командой **Edit Boot Vector** (**Измен. BootVector**).

7.2.2.56. **Edit Status Byte** (**Измен. StatusByte**).

Команда используется для просмотра значения статус-байта, считанного из микроконтроллера, а также для изменения этого значения для его последующей записи в микросхему.

7.2.2.57. **Edit Boot Vector** (**Измен. BootVector**).

Команда используется для просмотра значения бут-вектора, считанного из микроконтроллера, а также для изменения этого значения для его последующей записи в микросхему.

7.2.2.58. `view X2 Status` (`просм. сост. X2`).

Команда служит для просмотра ранее считанного состояния бита конфигурации, управляющего режимом X2 микроконтроллера.

7.2.2.59. `Set X2 Mode Off` (`Выкл. Режим X2`).

Команда выключает режим X2 микроконтроллера.

7.2.2.60. `Set X2 Mode On` (`Вкл. Режим X2`).

Команда включает режим X2 микроконтроллера.

7.2.2.61. `Read CFG & ID` (`Считать CFG&ID`).

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

7.2.2.62. `Write CFG & ID` (`Записать CFG&ID`).

Команда осуществляет запись ID-кодов и конфигурационного слова (или двух слов) PIC-контроллера информацией, считанной ранее из другой микросхемы, или подготовленной командами

`Edit ID-code` (`Изменить ID-код`) и `Edit Config.Word` (`Измен.слово конф`).

7.2.2.63. `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.

7.2.2.64. `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.

7.2.2.65. **Read Cfg Words** (**Считать конфиг.**).

Чтение содержимого ячеек слов конфигурации микроконтроллеров PIC18, и их запоминание в буфере программатора для последующего просмотра или записи в другую микросхему.

7.2.2.66. **Read ID-code** (**Считать ID-код**).

Чтение содержимого ячеек идентификации (ID) микроконтроллеров PIC18, и их запоминание в буфере программатора для последующего просмотра или записи в другую микросхему.

7.2.2.67. **Write Cfg Words** (**Записать конфиг.**).

Команда осуществляет запись ячеек слов конфигурации микроконтроллеров PIC18 информацией, считанной ранее из другой микросхемы, или подготовленной командой **Edit Cfg words** (**Изменить конфиг.**).

7.2.2.68. **Write ID-code** (**Записать ID-код**).

Команда осуществляет запись ячеек идентификации (ID) микроконтроллеров PIC18 информацией, считанной ранее из другой микросхемы, или подготовленной командой **Edit ID-code** (**Изменить ID-код**).

7.2.2.69. **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.

7.2.2.70. **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.

7.2.2.71. Edit Conf.word 2 (Изм.слово конф.2).

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

7.2.2.72. Edit Cfg Words (Изменить конфиг.).

Команда используется для просмотра значений слов конфигурации, считанных из микроконтроллера PIC18, а также для ввода значений этих битов для записи в микросхему.

7.2.2.73. Set Calibration (Устан.калибровку).

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

Для AVR-контроллеров, содержащих несколько калибровочных байтов, предлагается также выбрать номер байта.

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

7.2.2.74. 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.

7.2.2.75. 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)**.

7.2.2.76. 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.

7.2.2.77. Enable DCLK (Разрешить DCLK).

Команда осуществляет установку конфигурационного бита блокировки сигнала DCLK.

7.2.2.78. Disable DCLK (Запретить DCLK).

Команда осуществляет сброс конфигурационного бита блокировки сигнала DCLK.

7.2.2.79. Enable AutoStore (Разреш.автосохр.).

Команда разрешает функцию автосохранения информации при выключении питания в микросхеме NVRAM.

7.2.2.80. DisableAutoStore (Запрет.автосохр.).

Команда запрещает функцию автосохранения информации при выключении питания в микросхеме NVRAM.

7.2.2.81. ISP mode (прогр. в системе).

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

7.2.2.82. Socket Mode (прогр. в панели).

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

7.2.2.83. Sel. Data Memory (Выбр. пам.данных).

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

7.2.2.84. Sel.Progr.Memory (Выбр. пам.прогр.).

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

7.2.2.85. Switch To LDR0M (Перекл. в LDR0M).

Команда переключает программатор в режим работы с памятью LDR0M микроконтроллеров серии W78.

7.2.2.86. Switch To APROM (Перекл. в APROM).

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

7.2.2.87. Select Add. Mem. (Выбр. доп.память).

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

7.2.2.88. Select Main Mem. (Выбр. осн.память).

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

7.2.2.89. Select Prot.Area (Выбр. обл.защиты).

Команда переключает программатор в режим работы с дополнительной областью памяти микросхем AT24RF08 и PCA24S08, содержащей биты защиты блоков основной памяти от считывания и перезаписи.

7.2.2.90. Set slave addr. (Уст.схемн.адрес).

Команда служит для выбора схемного адреса микросхемы на шине I²C при внутрисхемном программировании EEPROM.

7.2.2.91. **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 конце**).

Эта команда 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 команда 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 символьной 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 SE запоминает до 8 макросов. Макросы хранятся в энергонезависимой памяти, т.е. сохраняются при выключенном питании.

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**).

Меню команд изменения настроек программатора и сервисных 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. LCD Light Off (**Отключ. подсветки**).

Настройка функции автоматического выключения подсветки дисплея программатора при паузах в работе (отсутствии нажатия клавиш) – выбор длительности таких пауз или отключение этой функции.

После автоматического выключения подсветка вновь включается при продолжении работы с программатором – при выводе нового сообщения после нажатия любой клавиши.

7.2.5.5. Auto Power Off (**Автовключение**).

Настройка функции автоматического выключения программатора при паузах в работе (отсутствии нажатия клавиш) – выбор длительности таких пауз или отключение этой функции.

За 1 минуту до автоматического выключения программатор выдает предупреждающий звуковой сигнал.

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**).

7.2.7. *Новые команды.*

По мере выпуска обновлений, расширяющих список поддерживаемых программатором микросхем, возможно появление новых команд, предназначенных для работы со специфическими функциями новых микросхем. Название команд и их интерфейс обычно интуитивно понятны пользователю. В случае необходимости, следует обращаться за консультацией в службу техподдержки производителя программатора по телефону или электронной почте.

7.3. Работа с ПЗУ различных типов.

Программатор AutoProg SE позволяет осуществлять перезапись (копирование) информации из микросхемы одного типа в микросхему другого типа. Для этого следует задать тип микросхемы образца, считать информацию в буфер, извлечь микросхему из панели, 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 SE комплектуется программным обеспечением (ПО) для работы в операционной системе Windows.

Удобный и доступный интерфейс управляющей программы 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 ПО должна соответствовать версии внутреннего ПО программатора. При обнаружении несоответствия версий, управляющая программа предложит осуществить замену (перезапись) внутреннего ПО – базы данных и микропрограммы главного контроллера. Эта операция используется для обновления программатора. Не рекомендуется замена 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.

Для запуска управляющей программы используйте ярлык в разделе "Все программы" главного меню Windows, вызываемого кнопкой "Пуск". При необходимости, этот ярлык можно скопировать на "Рабочий стол" или на панель быстрого запуска, перетащив его правой кнопкой мыши с последующем выбором операции "Копировать".

Первый запуск программы лучше всего выполнить без автоматической инициализации программатора при запуске, используя ярлык "AutoProg SE without initialization". Это позволит выбрать порт для подключения программатора и параметры соединения.

Удаление программы производится при помощи "Установка/Удаление программ" в "Панели управления" Windows.

Перед установкой новых версий программы, ее следует закрыть – Windows не позволит осуществить замену компонентов работающей программы. Удаление старой версии не требуется.

9. Базовый комплект поставки.

- 1) Программатор AutoProg SE.
- 2) Блок питания (сетевой адаптер).
- 3) Инструкция по эксплуатации.
- 4) Компакт-диск с программным обеспечением.
- 5) Кабель интерфейсный USB A-B.
- 6) Кабель интерфейсный COM 9F-9F (по требованию).
- 7) Кабель для внутрисхемного программирования.
- 8) Коробка упаковочная.

10. Гарантийные обязательства.

Изготовитель гарантирует нормальную работу прибора в течение 24 месяцев с момента продажи потребителю, а также бесплатный ремонт в течение 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.

11. Возможные проблемы при работе с программатором и способы их решения.

1. Вид окна управляющей программы отличается от описанного в Help. Отсутствует меню микросхем, команда установки связи с программатором недоступна.

Неправильно осуществлен запуск управляющей программы. Для правильного запуска используйте ярлык в разделе "Все программы" главного меню Windows, вызываемого кнопкой "Пуск".

2. Не удастся выбрать определенный тип микросхемы. Выводится сообщение "Ошибка конфигурации" ("Configuration Error").

3. Не удастся начать сеанс работы в автономном режиме. Выводится сообщение "Ошибка базы данных" ("Database Error").

Возможная причина – искажение информации в базе данных программатора. Выполните проверку базы данных соответствующей командой управляющей программы или, используя функцию самодиагностики в автономном режиме, которая запускается длительным удержанием любой клавиши программатора, кроме клавиши "Режим", в исходном состоянии, после включения питания. При подтверждении искажений, произведите обновление (замену) внутреннего ПО программатора из окна "Обслуживание программатора" управляющей программы.

4. 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.

Гарантийный талон.

Универсальный программатор AutoProg SE.

Серийный номер _____

Дата выпуска программатора _____

Дата продажи _____

Продавец _____

Штамп или стикер

Отметки о ремонтах:
