Программирование микроконтроллеров NXP (Philips) P89V51RD2 на программаторе ChipProg фирмы Phyton

В настоящей инструкции описывается процесс программирования микроконтроллеров NXP (Philips) P89V51RD2 при прожиге прошивок для ККМ Меркурий.

Настоящая инструкция достоверно подходит для программатора «ChipProg+» (это проверено). Также должна бы подходить и для остальных «ChipProg»-ов, похожих на «ChipProg+».

Содержание

1. NXP или PHILIPS? 2
2. Важное замечание при программировании микросхем PLCC! 2
3. Вариант программирования N1 (на новых версиях программной оболочки к программатору ChipProg) 4
   3.1. Настройка кнопки «Auto» (автоматическое программирование) 6
4. Вариант программирования N2 (на устаревших версиях программной оболочки к программатору ChipProg) 7
   4.1. Настройка буфера кода программ Windows-оболочки 7
   4.2. Процесс программирования 8
5. Программирование бита удвоения генератора микроконтроллера 10

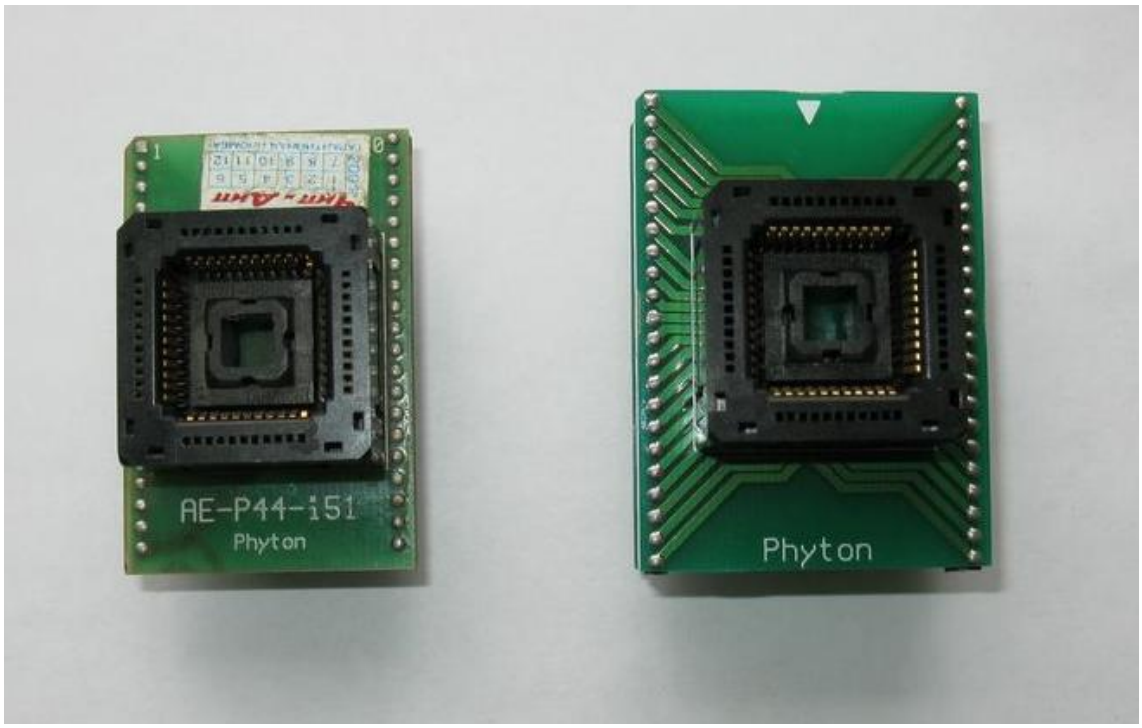
1. NXP или PHILIPS?

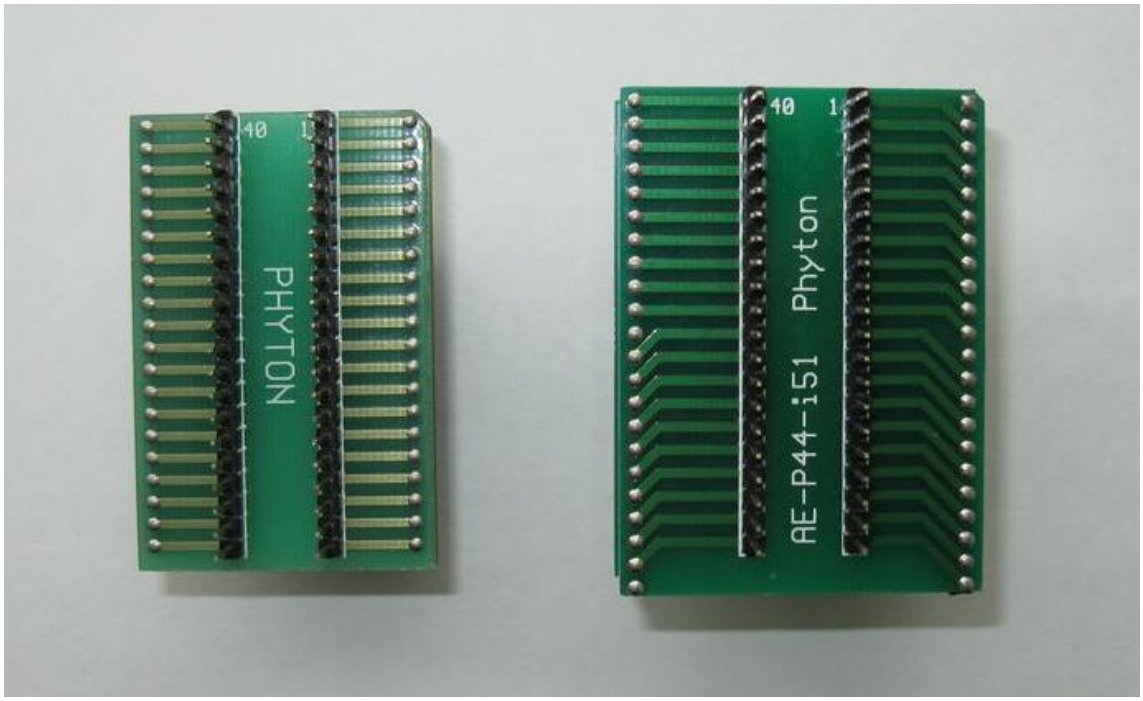
Часто встречается вопрос относительно выбора **или** названия используемых микроконтроллеров. В настоящей инструкции речь идет о МК «NXP89V51RD2». Это и есть наш старый добрый PHILIPS, только теперь все, что называлось «PHILIPS», теперь называется «NXP». Но это утверждение справедливо только для микроконтроллеров и прочих микросхем последних 2…3-х лет выпуска (сейчас 2009-й). В общем - отныне считать NXP равнозначным PHILIPS. Соответственно, и в программаторах, при выборе микросхем, искать по этим названиям, на случай, если в программе есть только один из них. По хорошему, при выборе PHILIPS в программе должна быть сноска «Смотрите NXP», или что-то в этом роде.

2. Важное замечание при программировании микросхем PLCC!

Важное замечание при программировании микросхем PLCC! При программировании микроконтроллеров (МК) **NXP (Philips) P89V51RD2** необходимо использовать адаптер «AE-P44-i51» Phyton, в котором задействованы все 44 вывода! Раньше выпускались одноименные адаптеры, где были задействованы только 40 выводов, что неприемлемо для МК NXP (Philips) P89V51RD2 последних ревизий (по крайней мере, 2008-го года). Микроконтроллеры более ранних ревизий (например, 2005-го года) программируются и на старом адаптере, но их использовать нельзя, т.к. в них существуют ошибки, которые отражаются на работоспособности некоторых режимов МК. Определить, сколько выводов задействовано в адаптере, можно путем прозванивания соответствующих контактов, либо визуально (но не гарантирую, что визуальный метод на 100% достоверен). Фото адаптеров см. ниже.

Слева на фото находится «старая» (задействовано 40 выводов) версия адаптера «AE-P44-i51», а справа - «новая» (задействовано 44 вывода) версия. Как видно по фотографиям, определить какой у вас адаптер, можно по маркировке (ее расположению) и по разводке платы.

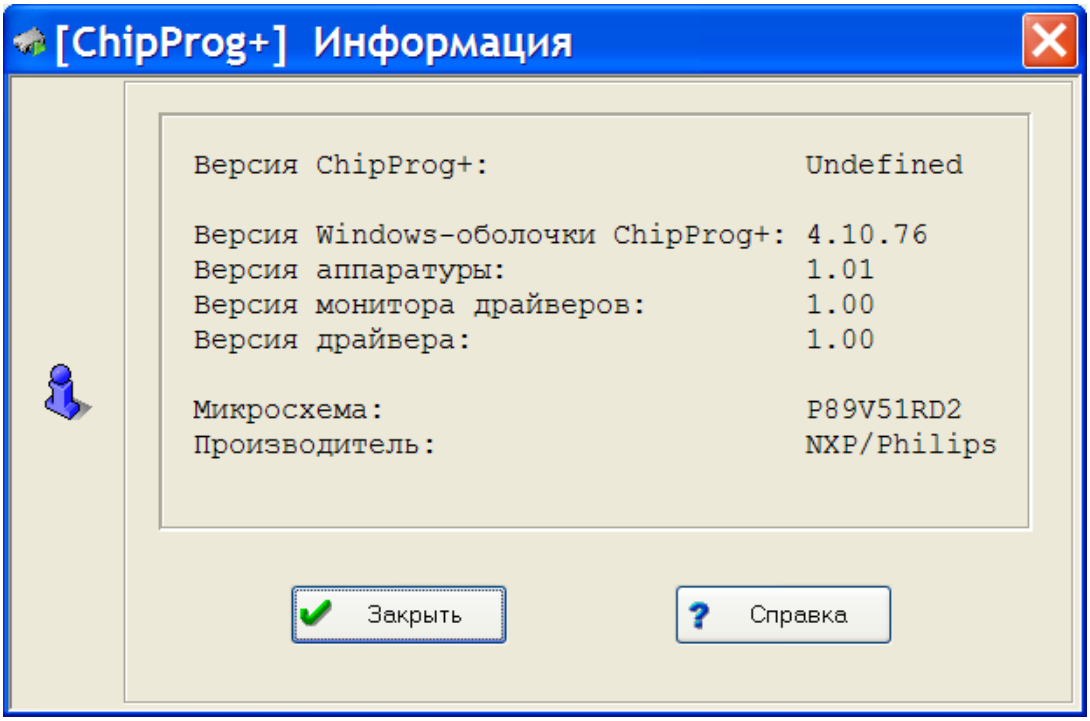



40
1
AE-P44-i51 Phyton
40
PHYTON

Phyton
AE-P44-i51
Phyton

3. Вариант программирования N1 (на новых версиях программной оболочки к программатору ChipProg)

Итак, представлено 2 варианта программирования микроконтроллеров. В первом варианте (который будет расписан далее в этом пункте) мы будем ориентироваться на более свежую версию программной оболочки к программатору. У меня, например, установлена вот эта версия (пункт меню «Справка» выбираем «О ChipProg…»). Т.е. версия 4.10.76 (версия Windows-оболочки).

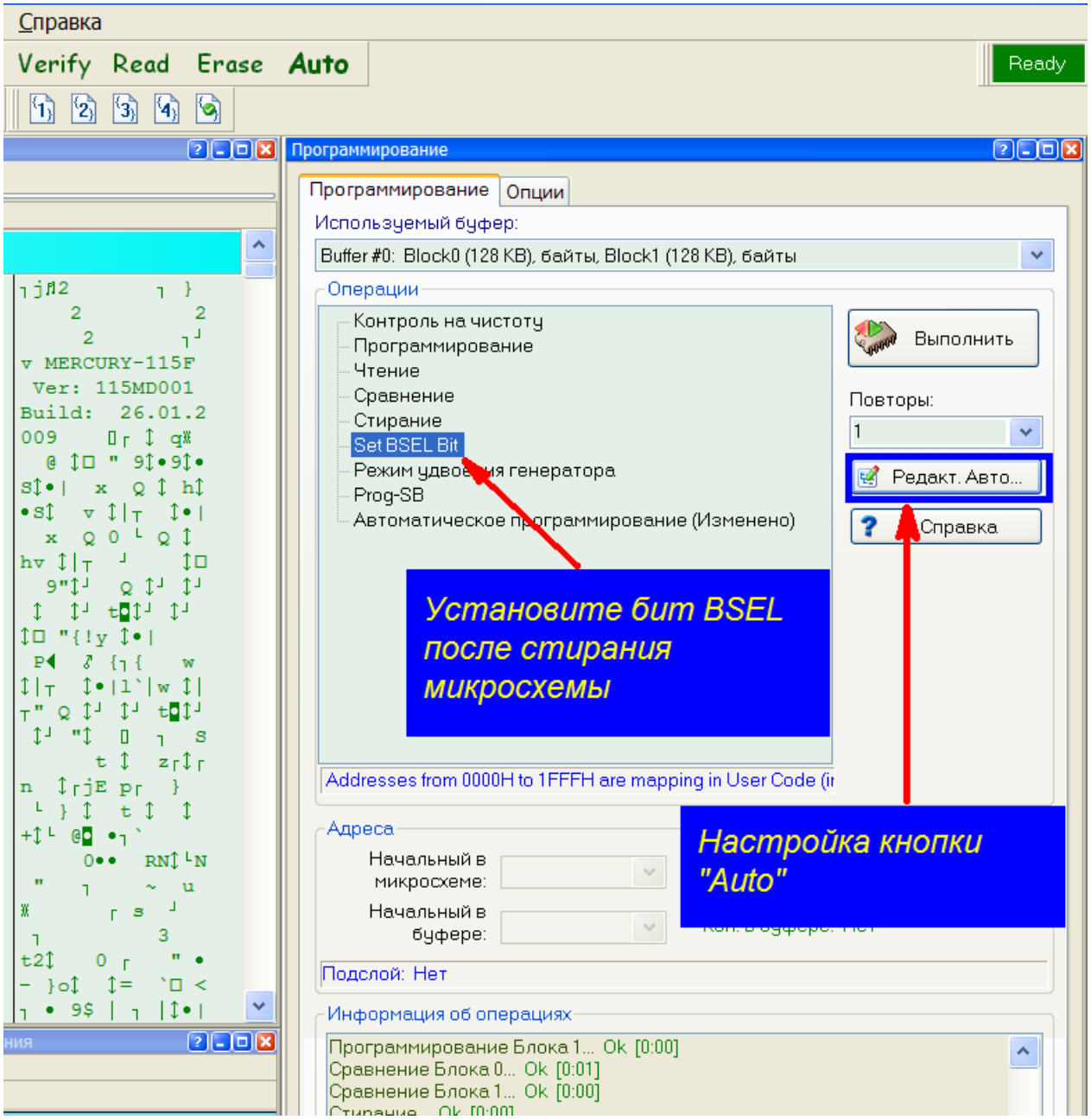


В обновленных версиях оболочки учтены аппаратные особенности микроконтроллера, поэтому процесс программирования сводится к стандартным действиям «Стирание (Erase)» → «Контроль на чистоту (Check)» → «Программирование (Program)» → «Сравнение (Verify)». Единственной особенностью программирования NXP P89V51RD2 в данном случае является то, что необходимо после цикла стирания МК запрограммировать бит BSEL «Set BSEL bit» (двойным щелчком мыши по соответствующему пункту), чтобы наша прошивка стартовала из нужного банка кода программ. Без программирования этого бита прошивка будет пытаться стартовать из банка «boot-загрузчика», а его мы не программировали, и соответственно, в итоге наш процессор просто не начнет выполнять код основной прошивки с нужного адреса).

Установка бита показана на рисунке ниже.

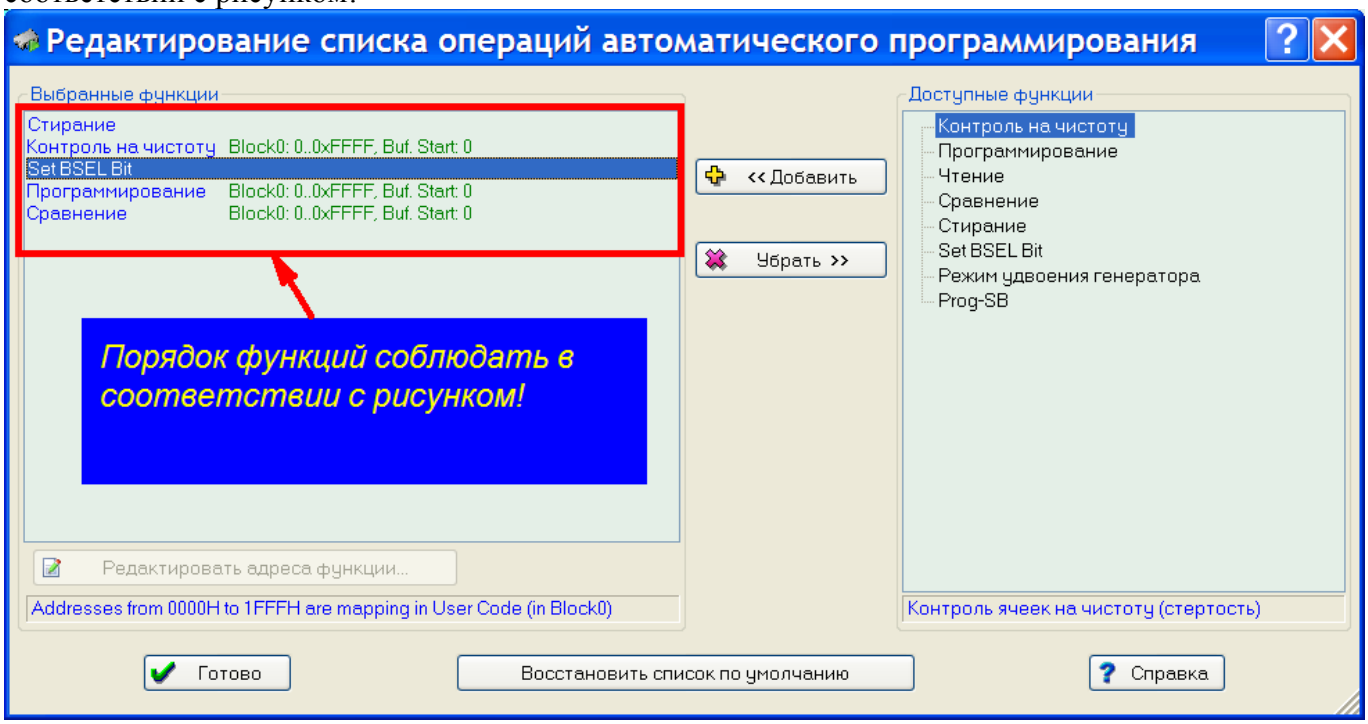
Таким образом, процесс программирования сводится к следующим действиям:

1) **Выбрать микросхему NXP P89V51RD2 (параллельный режим программирования)**
2) **Загрузить файл прошивки**
3) **Выполнить действия «Стирание (Erase)» → «Контроль на чистоту (Check)» → «Установить бит BSEL (Set BSEL bit)» → «Программирование (Program)» → «Сравнение (Verify)»**
4) **Конец**


Справка
Verify Read Erase Auto
Ready
Программирование
Программирование
Опции
Используемый буфер:
Buffer #0: Block0 (128 KB), байты, Block1 (128 KB), байты
Операции
Контроль на чистоту
Программирование
Чтение
Сравнение
Стирание
Set BSEL Bit
Режим удвоения генератора
Prog-SB
Автоматическое программирование (Изменено)
Выполнить
Повторы:
1
Редакт. Авто...
Справка
Установите бит BSEL после стирания микросхемы
Addresses from 0000H to 1FFFH are mapping in User Code
Адреса
Начальный в микросхеме:
Начальный в буфере:
Настройка кнопки "Auto"
Подслой: Нет
Информация об операциях
Программирование Блока 1... Ok [0:00]
Сравнение Блока 0... Ok [0:01]
Сравнение Блока 1... Ok [0:00]

3.1. Настройка кнопки «Auto» (автоматическое программирование)

Вы можете упростить процесс программирования, настроив, кнопку «Auto» (Автоматическое программирование). Для этого нажмите кнопку «Редакт. Авто…» (см. рисунок выше), и установите список функций в соответствии с рисунком ниже. Порядок функций соблюдать в соответствии с рисунком!



Теперь, процесс программирования сводится к следующим действиям:

1) **Выбрать микросхему NXP P89V51RD2 (параллельный режим программирования)**
2) **Загрузить файл прошивки**
3) **Выполнить действие «Автоматическое программирование» (кнопка «Auto»).**
4) **Конец**

4. Вариант программирования N2 (на устаревших версиях программной оболочки к программатору ChipProg)

В этом варианте программирования предполагается, что у вас установлена устаревшая версия программной Windows-оболочки к программатору (версии, увы, не помню, но кажется, те, что ниже 4.00.00). Причем, МК NXP (или PHILIPS) P89V51RD2 в списке выбора микросхем должен присутствовать, но в окне программирования оболочки нет пункта установки бита BSEL (Set BSEL bit) (показанного на рисунке в предыдущем пункте).

Идея этого варианта сводится к тому, что прошивка прожигается в оба банка кода программ микроконтроллера (МК) – в банк boot-загрузчика и в банк кода программ пользователя. Таким образом, будет совершенно безразлично, из какого банка будет стартовать прошивка.

Следует внимательно выполнять все нижеописанные действия. Порядок действий при загрузке файла прошивки и прожиге важен! Иначе, вы не получите требуемый результат.

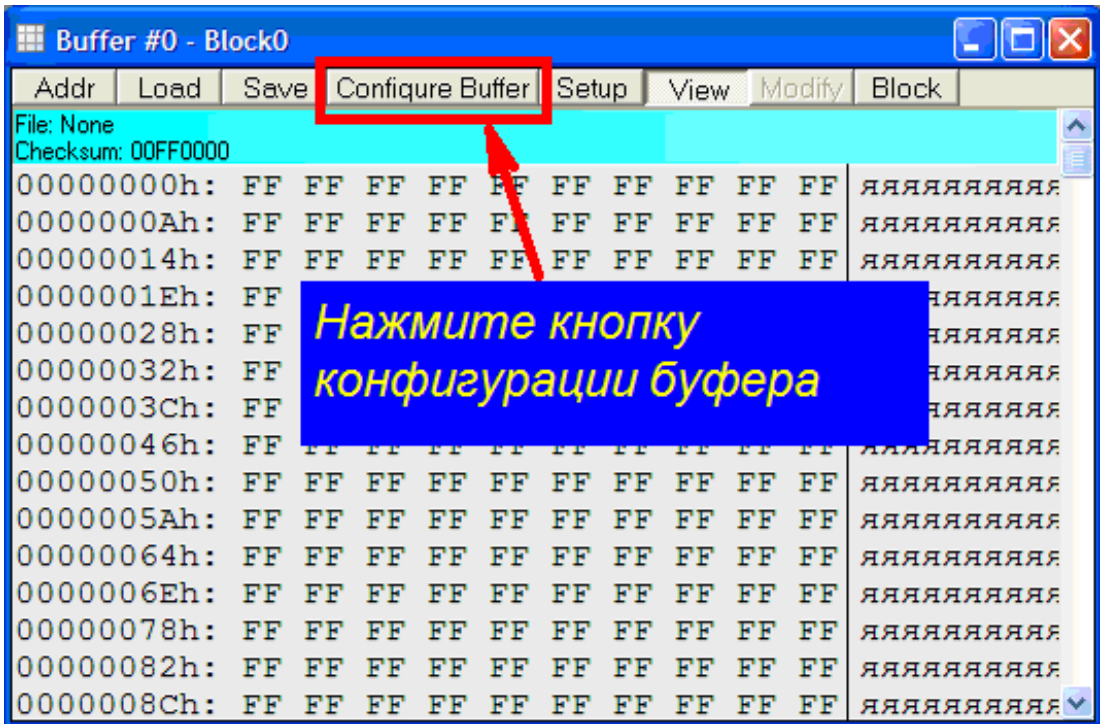
4.1. Настройка буфера кода программ Windows-оболочки

Первым делом, как известно, самолеты… Хотя с этим порой и не хочется соглашаться…

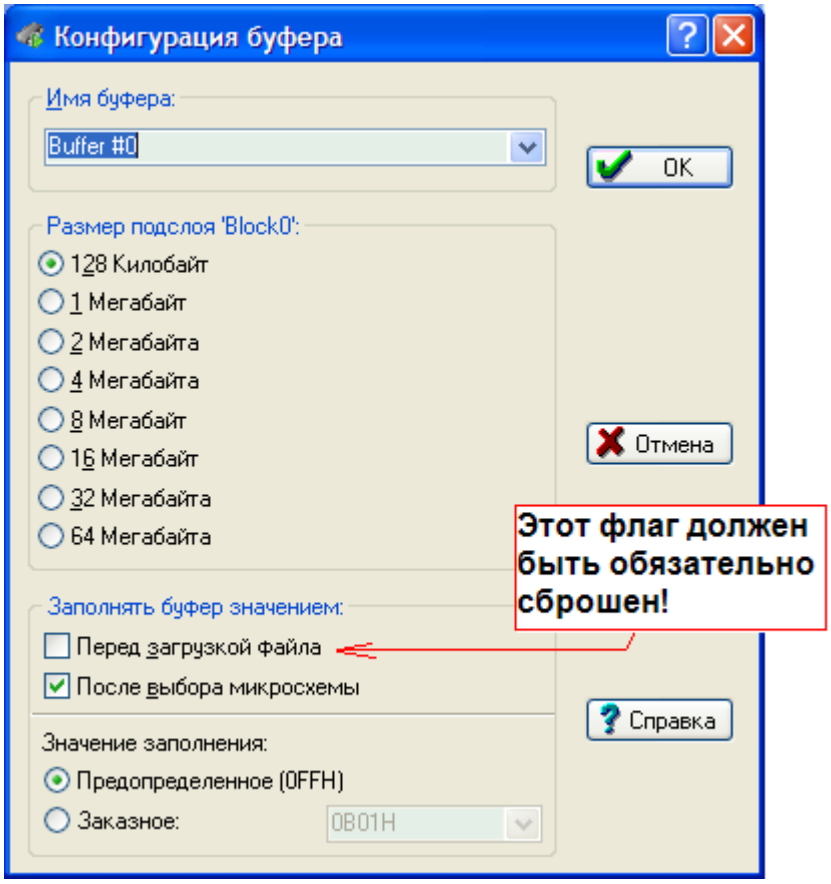
Итак, первым нашим делом будет настройка буфера(-ов) кода программ. Зачем? Ответ прост. Мы будем программировать два банка кода программ МК, соответственно прошивка из файла должна загружаться для этих двух банков в два независимых буфера (что реализовано в оболочке). По умолчанию в настройках оболочки зачастую выставлен флаг, благодаря которому каждый раз перед загрузкой файла буфера кода очищаются (устанавливаются в значение по умолчанию), а нам-то, как раз, это сейчас и не нужно.

Сразу замечу, что есть версии оболочек, в которых после выбора МК автоматически устанавливается и соответствующее количество буферов в оболочке для МК (для каждого банка кода программ).

Настроим буфер кода программ оболочки в соответствии с рисунком ниже.



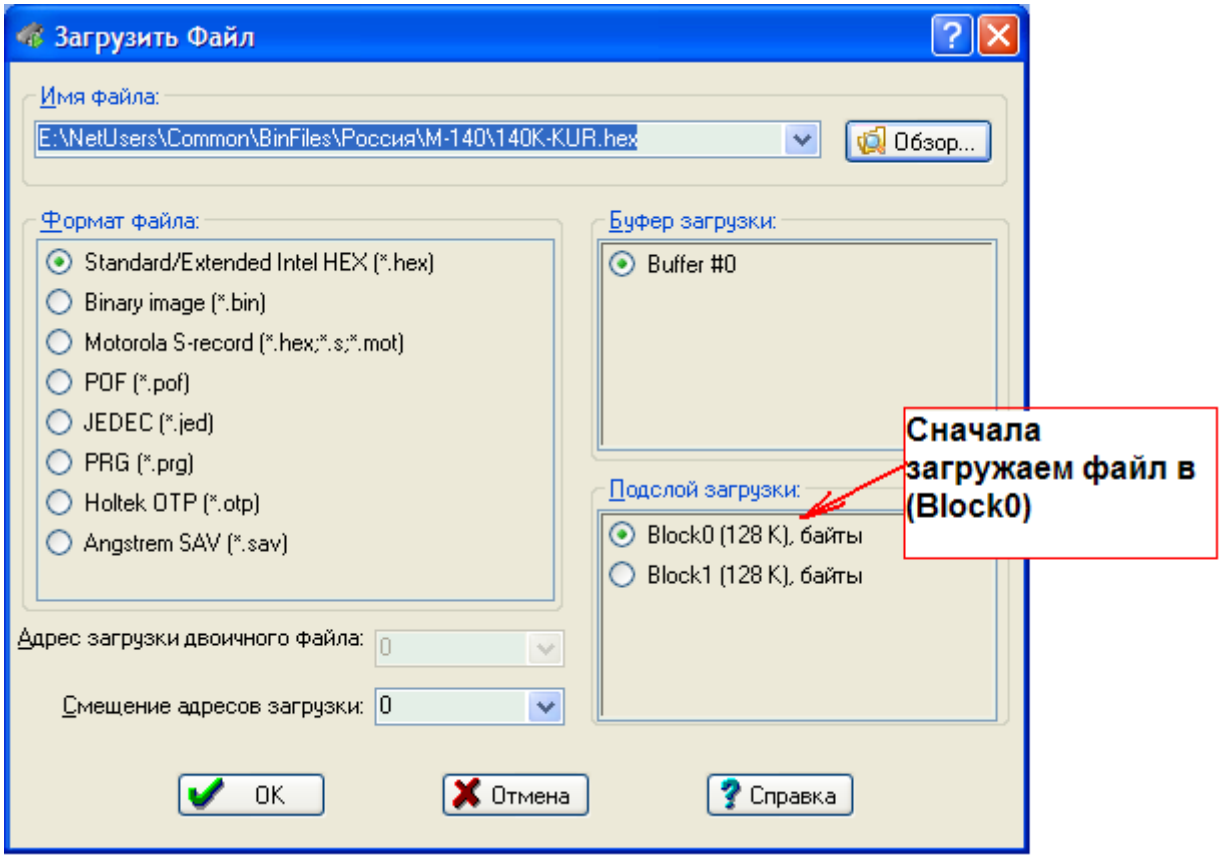
Установите параметры буфера в соответствии с рисунком ниже



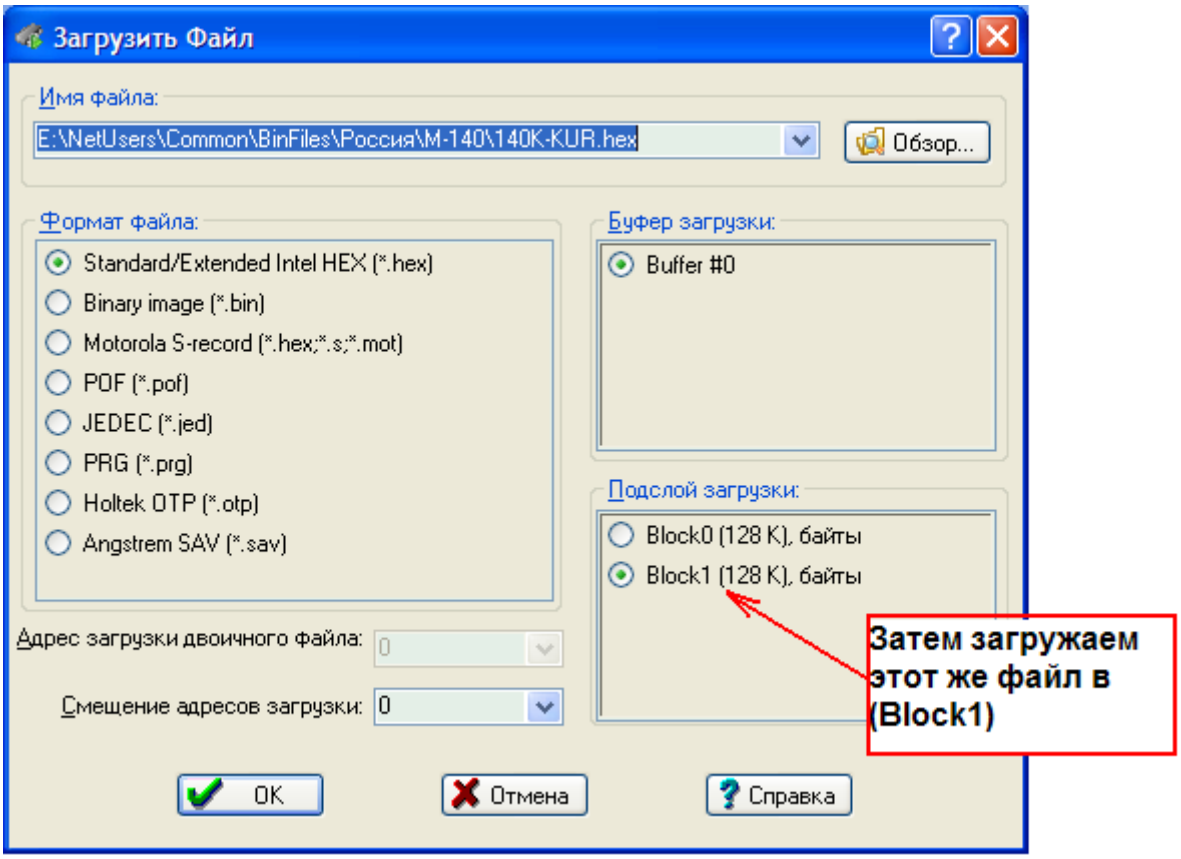
4.2. Процесс программирования

Предполагается, что вы уже выполнили настройку буфера программирования Настройка буфера кода программ Windows-оболочки. Сам же процесс программирования выполняется, как обычно, с одним отличием – теперь нам необходимо загружать прошивку из файла в два буфера кода оболочки.

1) **Выбрать микросхему NXP P89V51RD2 (параллельный режим программирования)**
2) **Сначала загружаем файл в блок 0 (Block0) буфера кода**



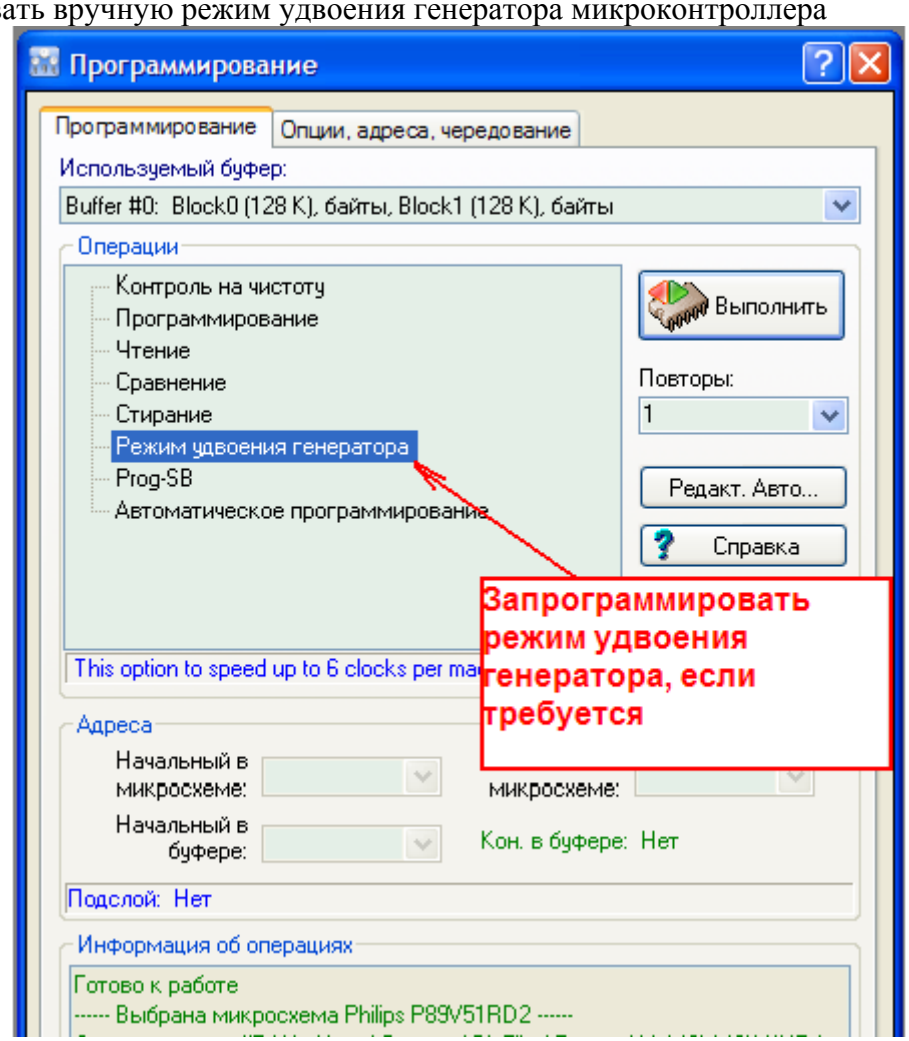
3) **Затем загружаем этот же файл в блок 1 (Block1) буфера кода**



4) **Выполнить действия «Стирание (Erase)» → «Контроль на чистоту (Check)» → «Программирование (Program)» → «Сравнение (Verify)»**
5) **Конец**

Если кнопка «Auto» настроена соответствующим образом, то выполните 4-й пункт одним нажатием кнопки «Auto»

5. Программирование бита удвоения генератора микроконтроллера

Бит удвоения генератора программируется, если этого требует сопроводительный файл к прошивке. Запрограммировать его можно либо вручную (см. рисунок ниже), либо вставить его программирование в функции кнопки «Auto»

Если требуется **(что указывается в сопроводительном файле к прошивке!!!)**, запрограммировать вручную режим удвоения генератора микроконтроллера