

Составил, проверил и отдал в добрые руки д. Фёдор - "fsem"

Программатор Postal AVR – инструкция по работе с микросхемами памяти EEPROM 24,25.93 серий, SPIflash 25 серии, Data flash 45 серии.

Подключаем программатор к USB – порту и запускаем программу.

Последняя версия программы лежит здесь:

<http://monitor.net.ru/forum/viewtopic.php?t=436716&postdays=0&postorder=asc&start=0>

Работа с микросхемами памяти I2C EEPROM.

Для примера: **24C32**. Напряжение питания = **5V0 (джемпер в положении 5V0)**. Микросхему вставляем в панельку «**24Схх**»!

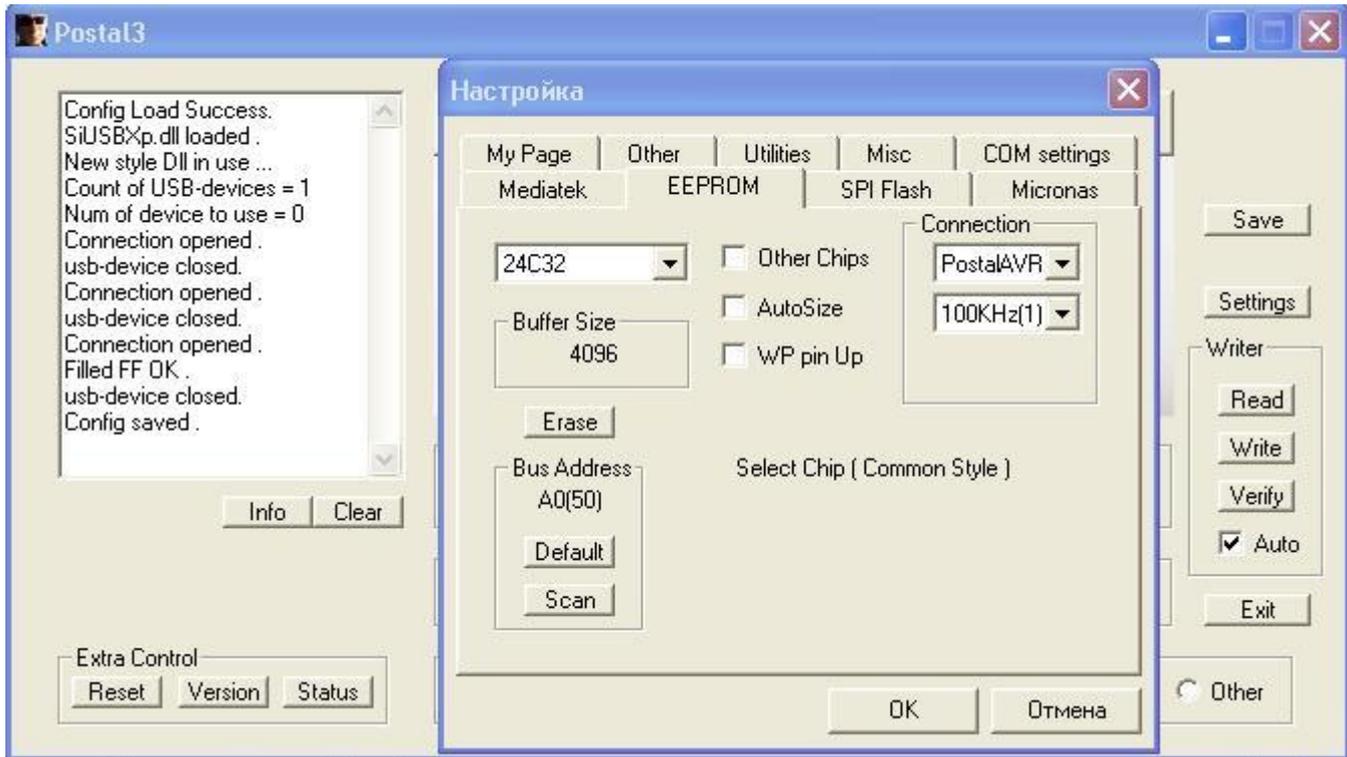
Кратко по основным функциям:

- **Info** - окно информации (логов)
- **Clear** - обнуление (стирание) информации в окне логов
- **Reset** - сброс
- **Version** - версия
- **Status** - статус
- **Setting** - вкладка настроек
- **Read** - чтение
- **Write** - запись
- **Verify** - сравнение повторным считыванием (верификация, проверка)
- **Auto** - если стоит галочка, то после чтения, либо записи, верификация пройдёт автоматически
- **Open** - путь выбора файла для записи
- **Exit** - выход
- **Select Mode** - выбор режима работы программы
- **Buffer Size** - размер буфера

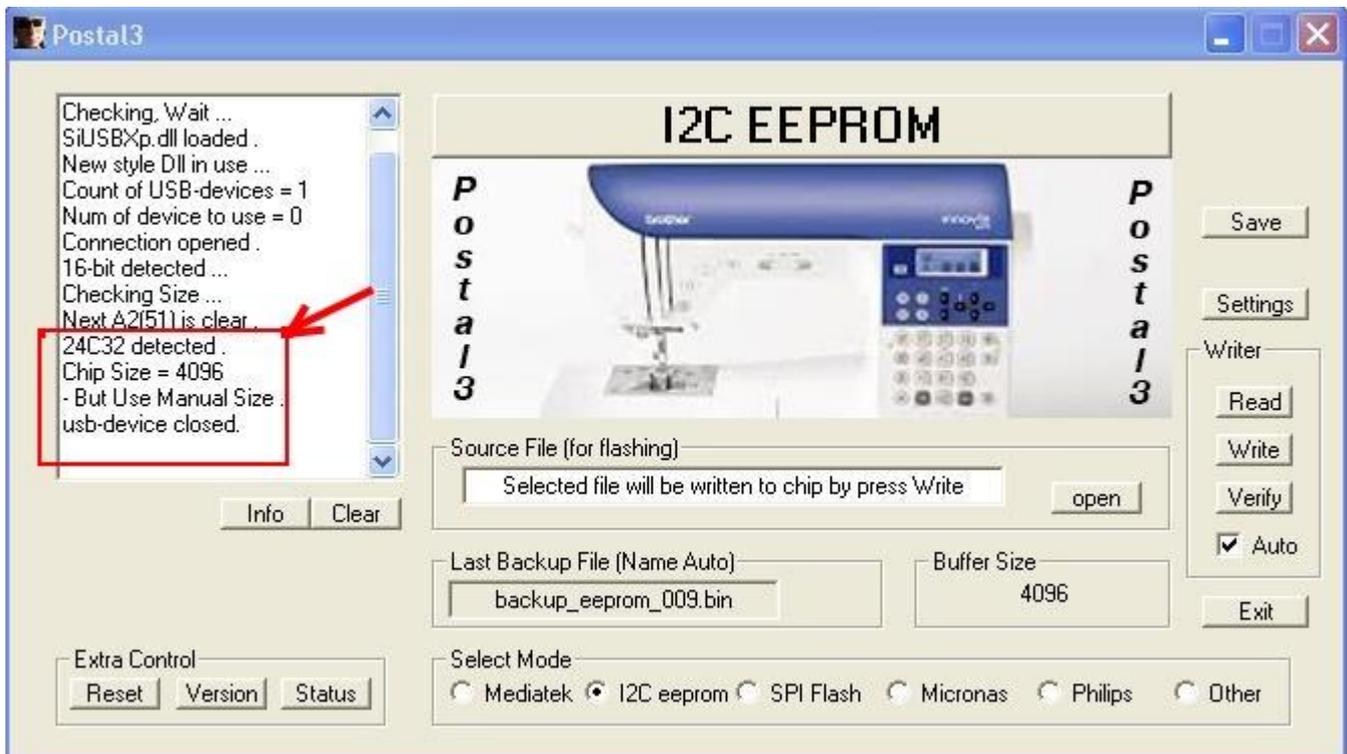
Выбираем режим **I2C EEPROM**, автоматическую верификацию «**Auto**» оставляем и жмём **Setting**:



Режим **Auto Size** очень удобен, так как программа сама определит размер EEPROM, с которой будет работать, но можно выбрать и ручную. На картинке вы видите ручной выбор микросхемы памяти, если нужно автоматом, ставим галку возле **Auto Size**.



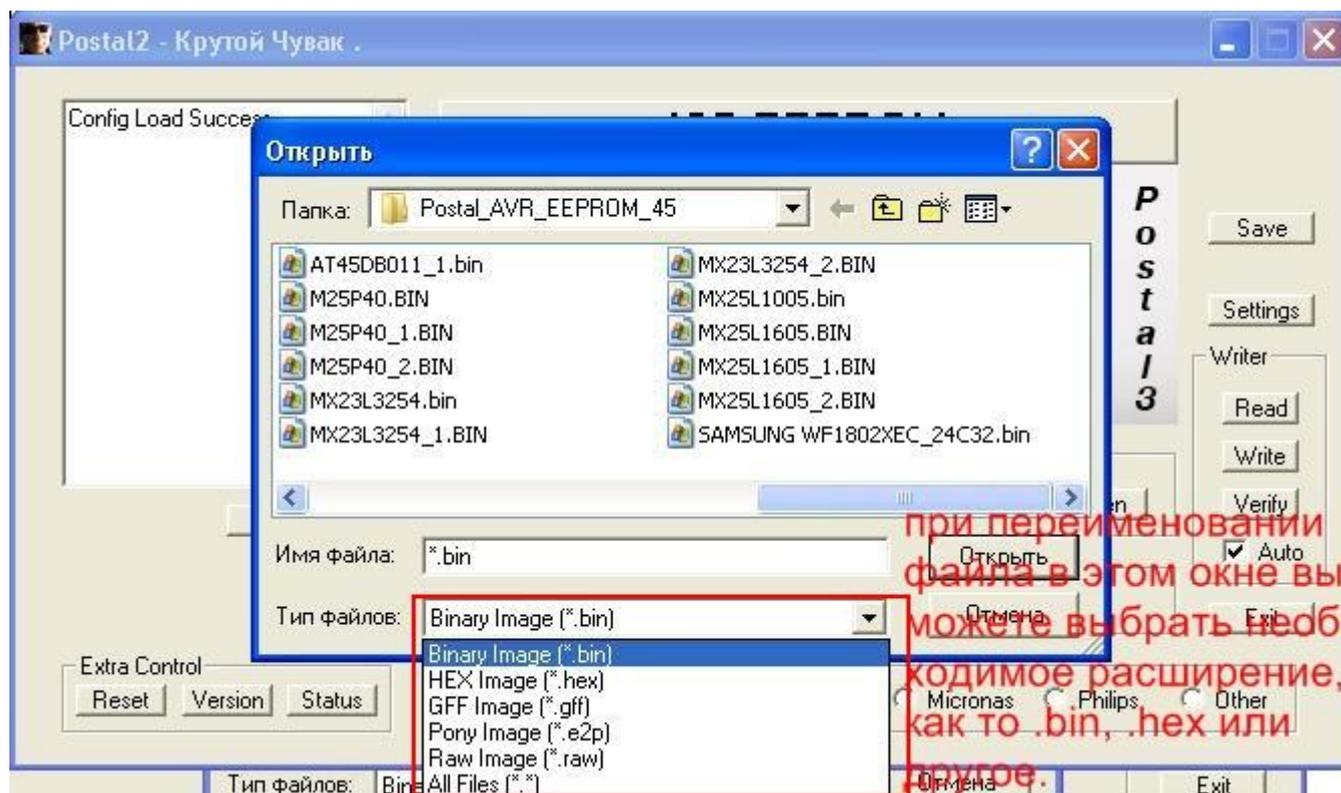
Жмём ОК и переходим к следующему шагу. Если хотим увидеть, что нашла программа, то жмём **Status** и видим тип микросхемы и адрес её отклика. Жмём **Version** и вся информация в лог-окне. Программа готова к работе с EEPROM



Жмём **Read**.

Так как мы оставили галочку в **Авто**, верификация происходит автоматически.

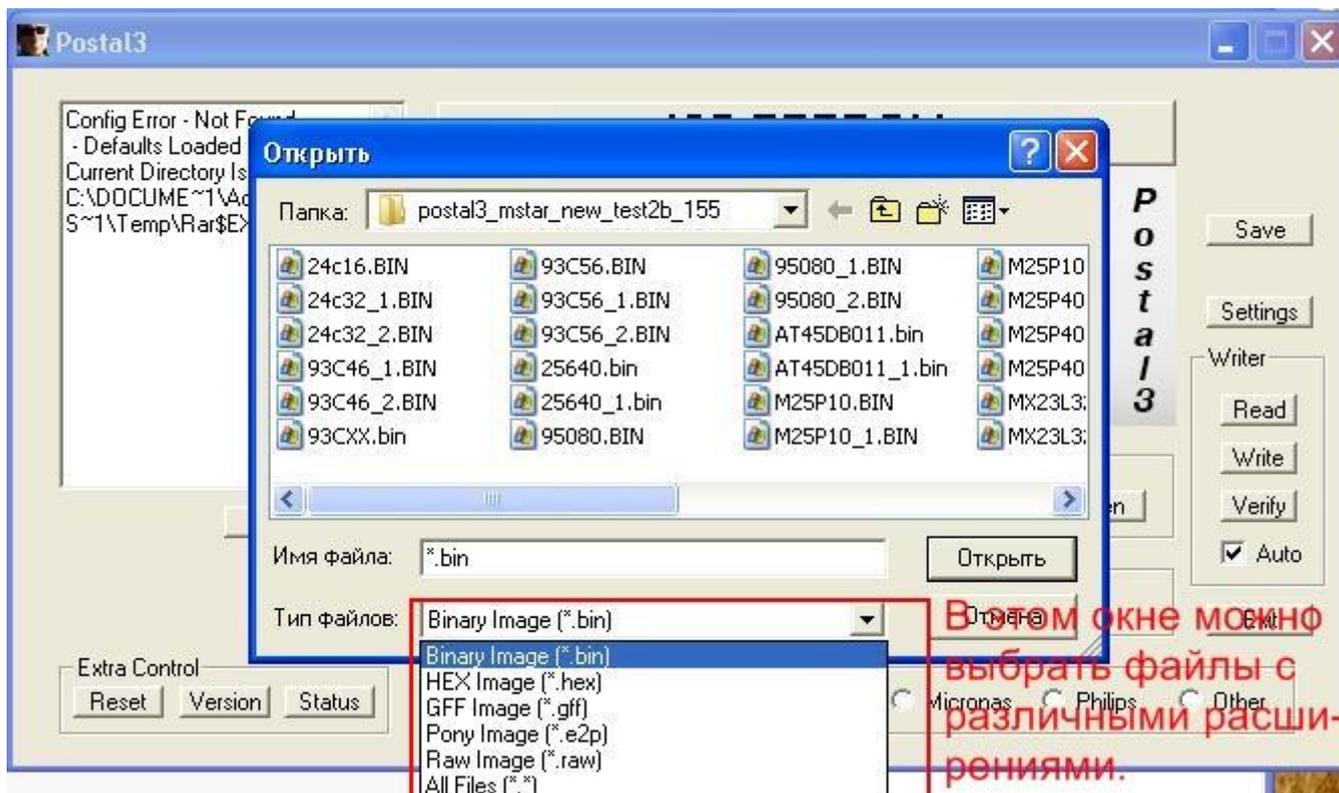
Считанный дамп (**backup_eeeprom_001**) находится в том же каталоге, что и сама программа. Очень удобно, что не нужно каждый раз самому менять название файла, с каждым новым чтением программа сама увеличит номер backup на единицу (**то есть backup_eeeprom_002, затем 003 и т.д.**) Если же мы хотим обнулить счётчик, то жмём **Reset** рядом с окошком. Кроме того, даже после сброса, если файл **backup_eeeprom_001** лежит на месте, программа запишет новый **backup_eeeprom_002**, а не сотрёт предыдущий. Но по завершении работы желательно переименовать файлы, например, считался файл под именем "**backup_eeeprom_001**", что бы было ясно, откуда файл переименовываем его, например: **SAMSUNG WF1802XEC_24C32**. Теперь мы не забудем, что данный файл прошивки считан со стиральной машины **SAMSUNG WF1802XEC_24C32** и м/с памяти стоит там **24C32**.



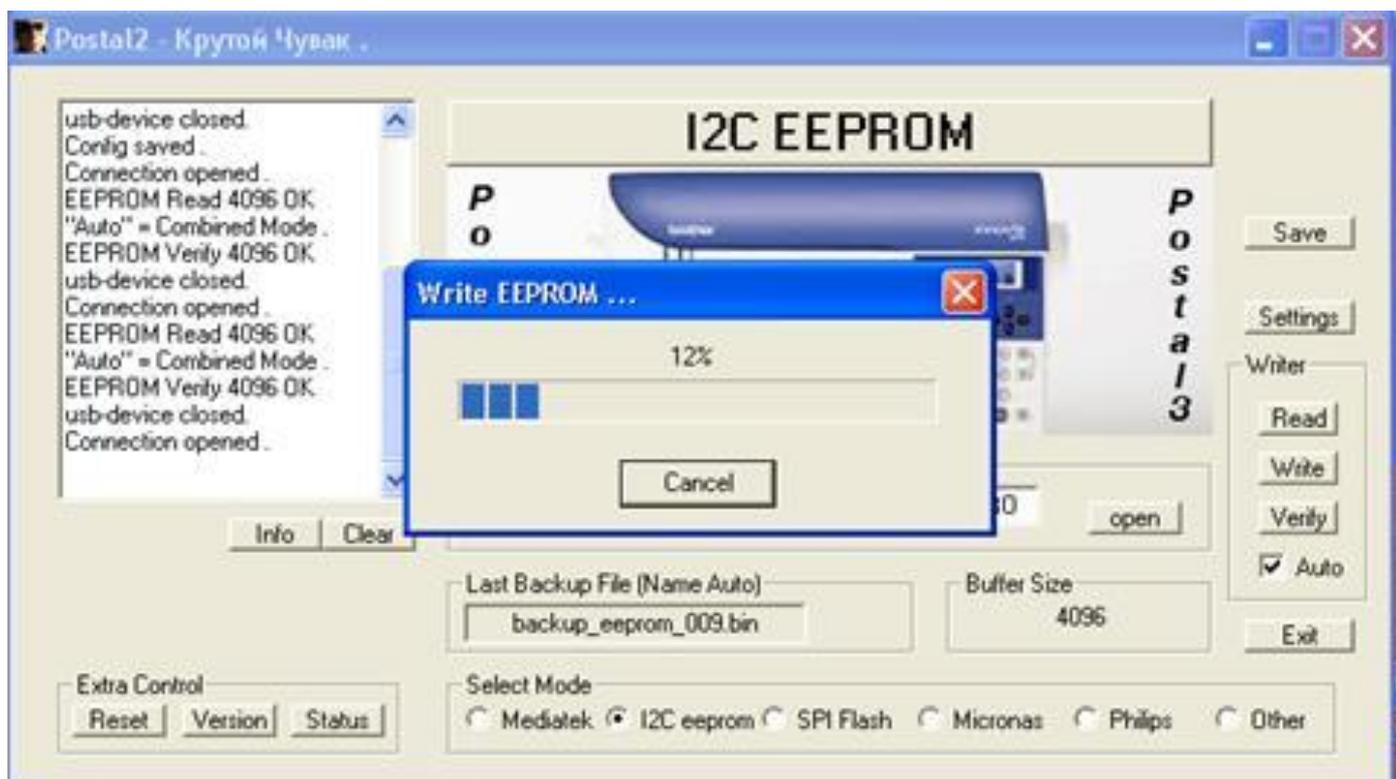
При переименовании файла, в нашем случае на **SAMSUNG WF1802XEC_24C32.bin** мы можем изменять расширение, см. картинку, но мы выбрали **.bin**

Составил, проверил и отдал в добрые руки д. Фёдор - "fsem"

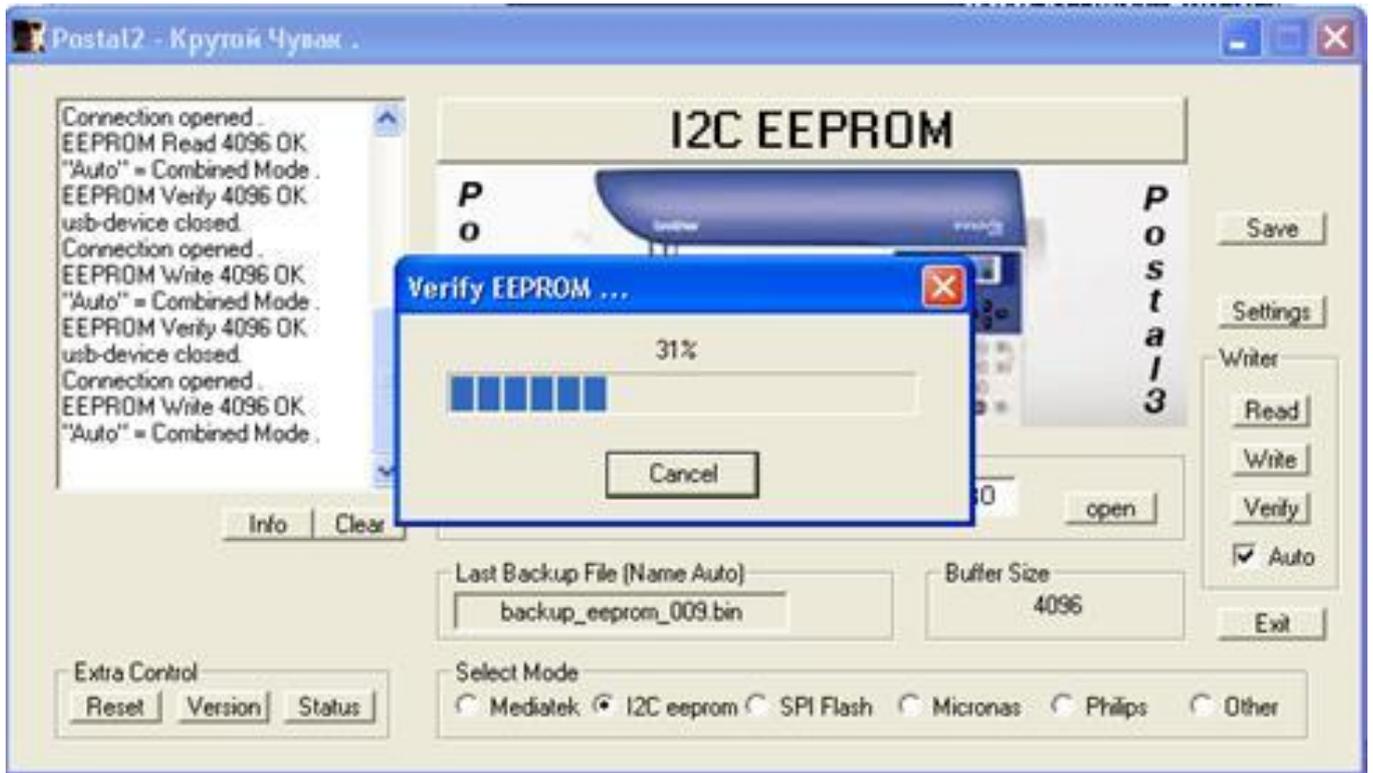
Запись I2C EEPROM.



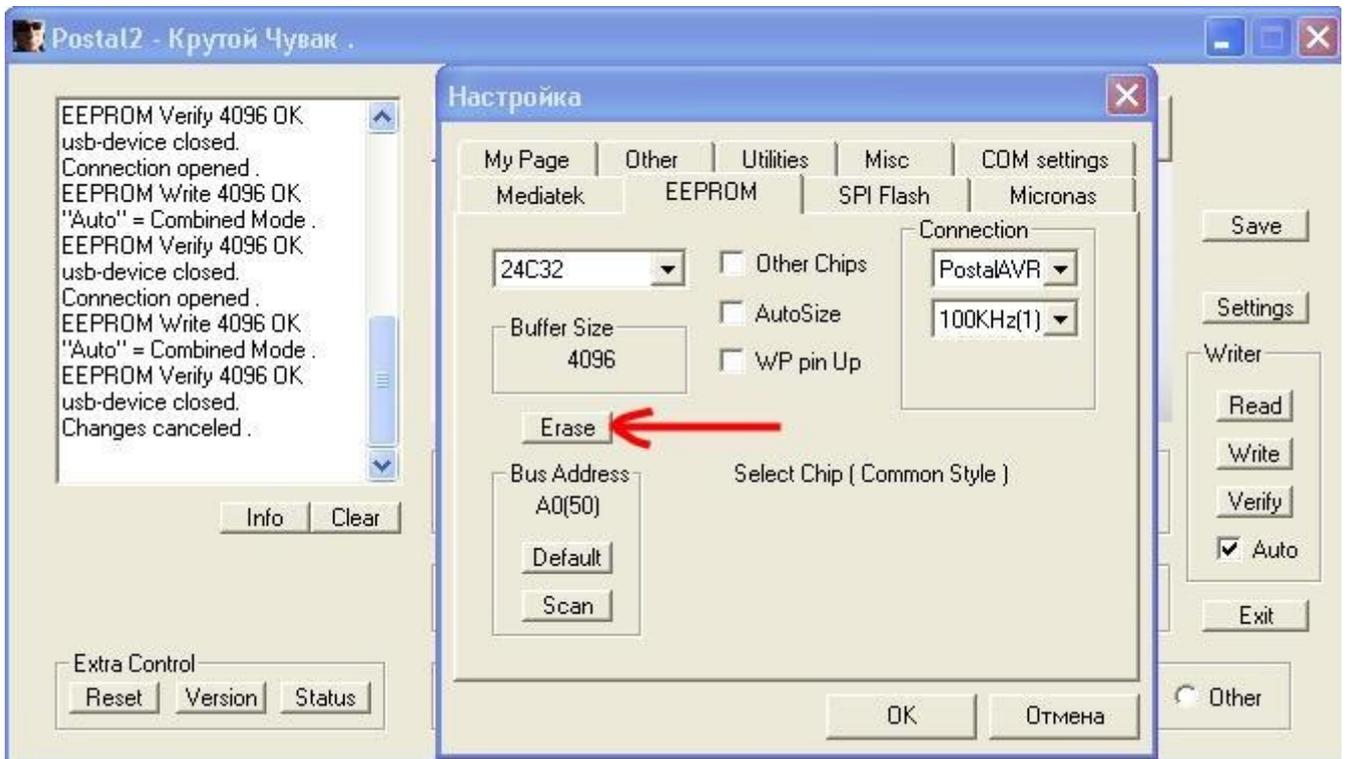
Для записи жмём **Open** и выбираем нужный файл (необязательно, чтобы он находился в той же папке) и с нужным расширением, как **.bin**, **.hex**, **.gff**, **.e2p**, **.raw**, а нажав **All Files** вы можете отрывать любые форматы в цифровому коде , в том числе и **.doc**, **.bmp**, **.jpg**.



Жмём **Write** и ждём окончания процесса записи.



После чего происходит проверка записанного дампа с данными в буфере.



Если же необходимо просто стереть, то во вкладке Настройки жмём Erase. При этом в микросхему запишется **FF**, согласно с выбранным размером буфера. А вот это мы увидим, если считываемая память была "чистая" (FF).



Микросхемы **I2C EEPROM** основаны на технологии '**EEPROM**'. При перепрограммировании микросхемы, предварительное стирание не требуется. Операция стирания, т.е. запись во все ячейки **FF** потребуется вам в том случае, когда нужно вставить в аппарат чистую микросхему.

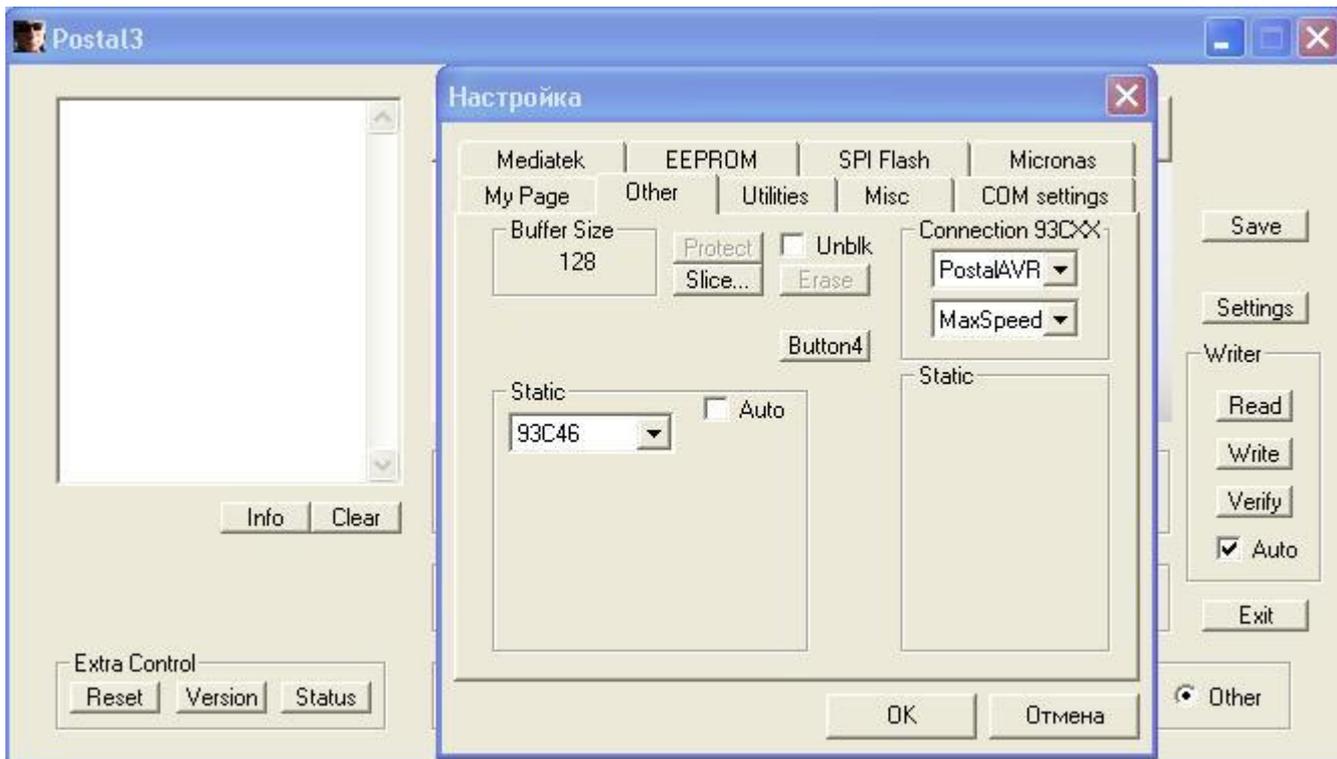
Работа с микросхемами памяти MW EEPROM, 93 серия и её производные.

Напряжение питания = **5V0 (переключатель в положении "5V0")**.

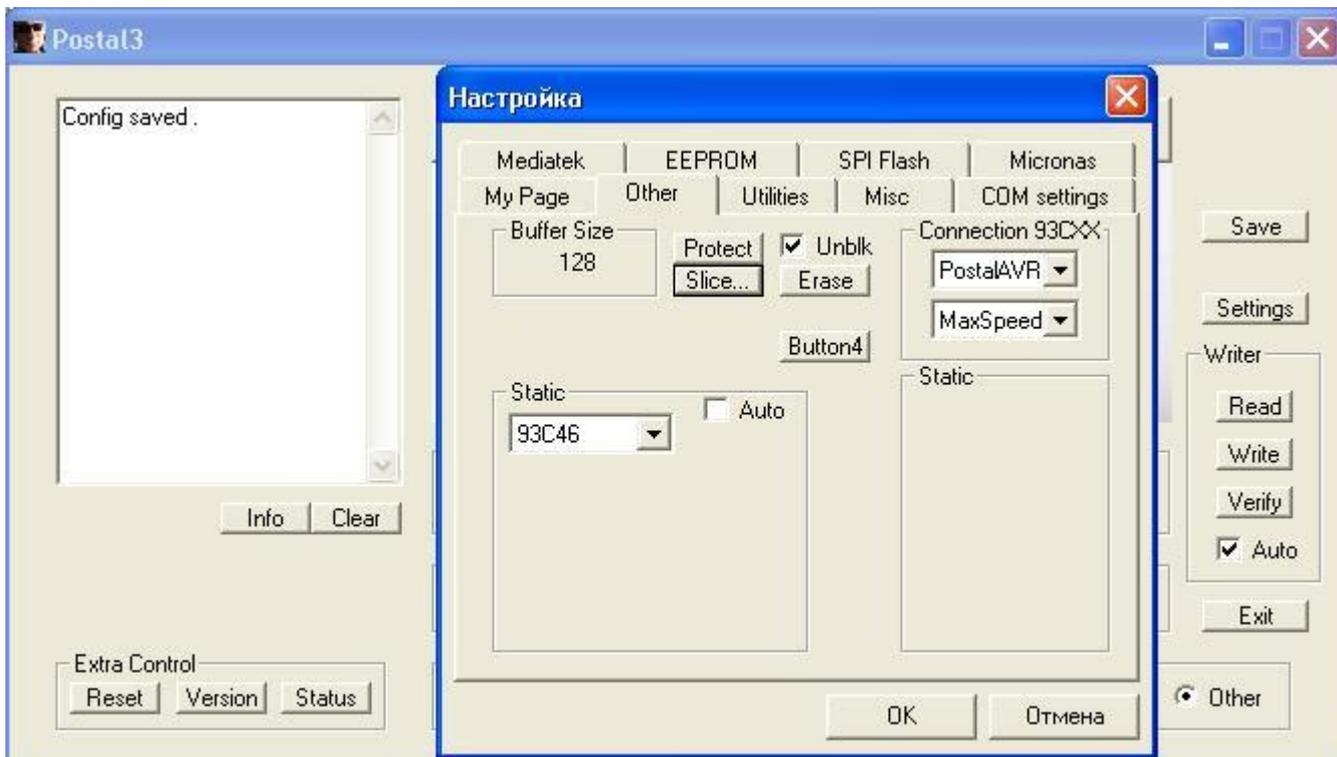
Для примера возьмём микросхему **93C46**, работа с которой ничем не отличается от работы с **I2C EEPROM** описанной выше.



Для начала поставьте «галку» в окошке **Other**.



далее жмём **Setting** и в ниспадающем меню выбираем **93C46**, можно работать и в режиме авто определения м/с, тогда должна стоять «галка» в окошке **Auto**



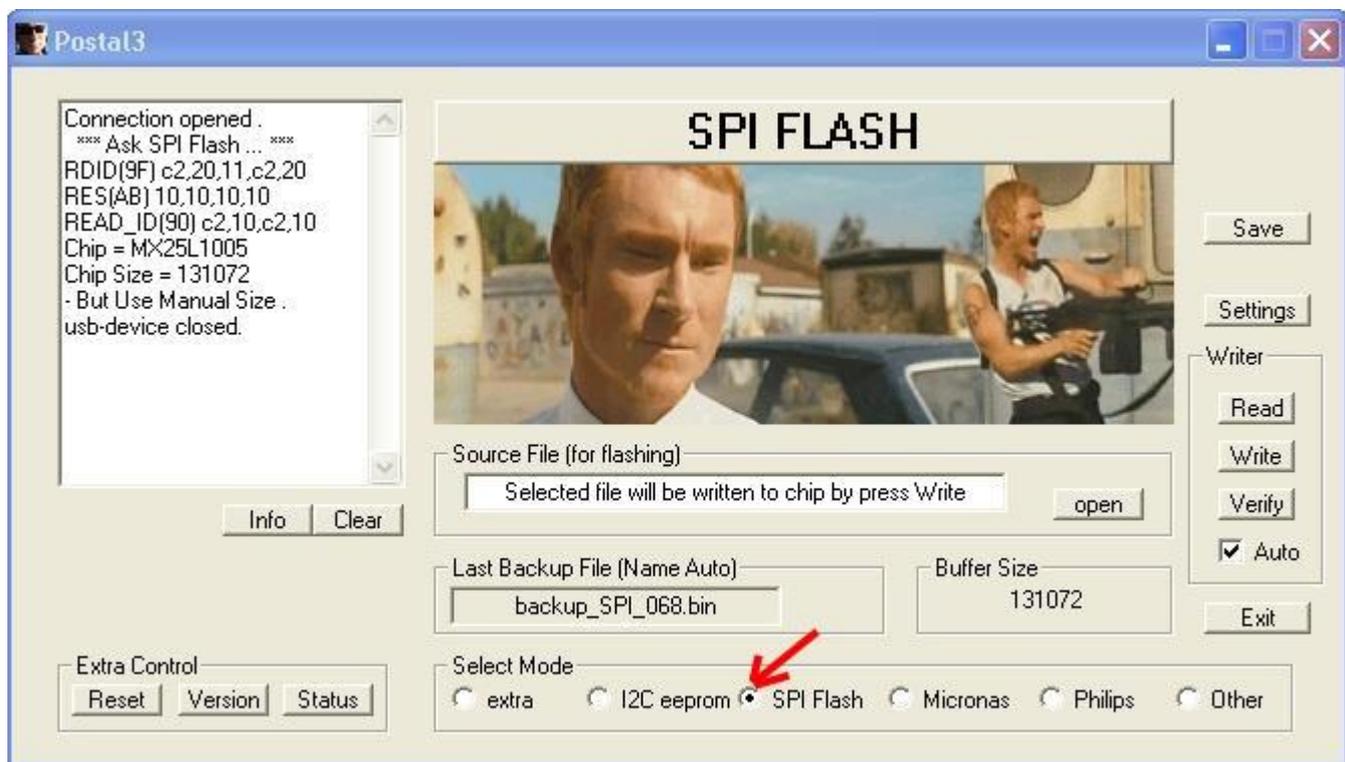
Для входа в режим стирания нужно поставить «галку» в окошке **Unblk** и тогда кнопка **Erase** станет активной.

Остальные операции (**Чтение/Стирание/Запись**) ничем не отличаются от аналогичных операций с **I2C EEPROM**.

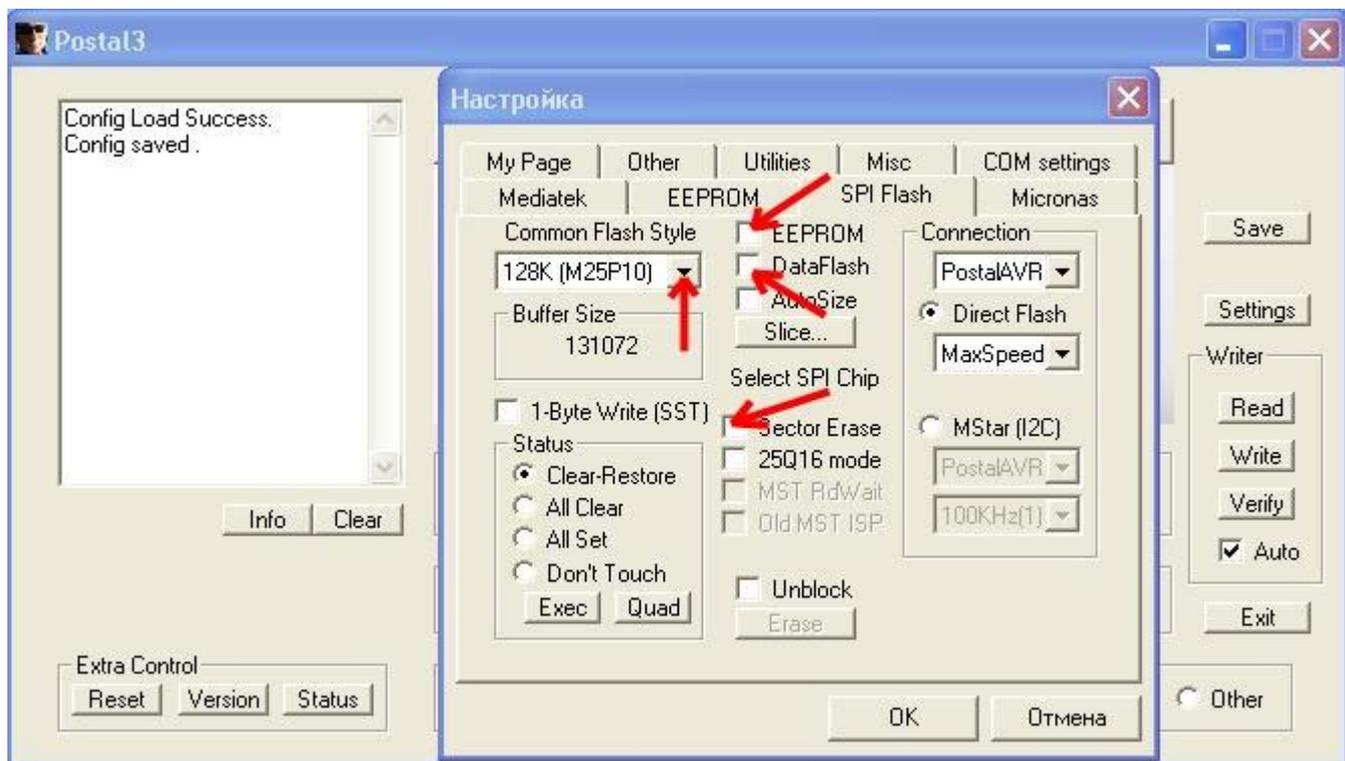
Микросхемы **MW EEPROM** основаны на технологии '**EEPROM**'. При перепрограммировании микросхемы, предварительное стирание не требуется. Операция стирания, т.е. запись во все ячейки **FF** потребуется вам в том случае, когда нужно вставить в аппарат чистую микросхему.

Инструкция по работе в режиме "SPI FLASH"

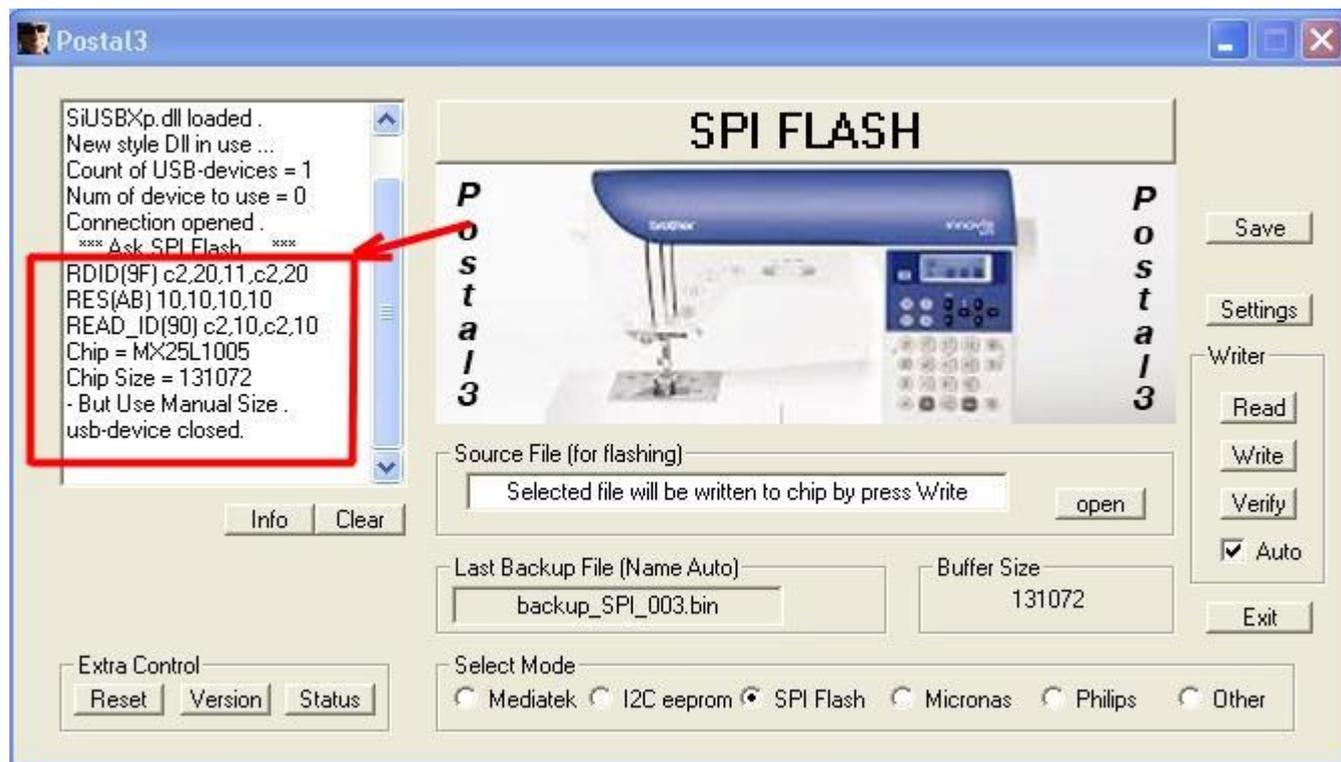
Подключаем программатор Postal_AVR, в панельку SPI-Flash вставляем микросхему памяти, например: **MX25L1005**, напряжение питания **3V3** (переключатель в положении **3V3**).



На иконке ставим галку в окошке **SPI Flash** и переходим к вкладке **Setting** (Настройки)



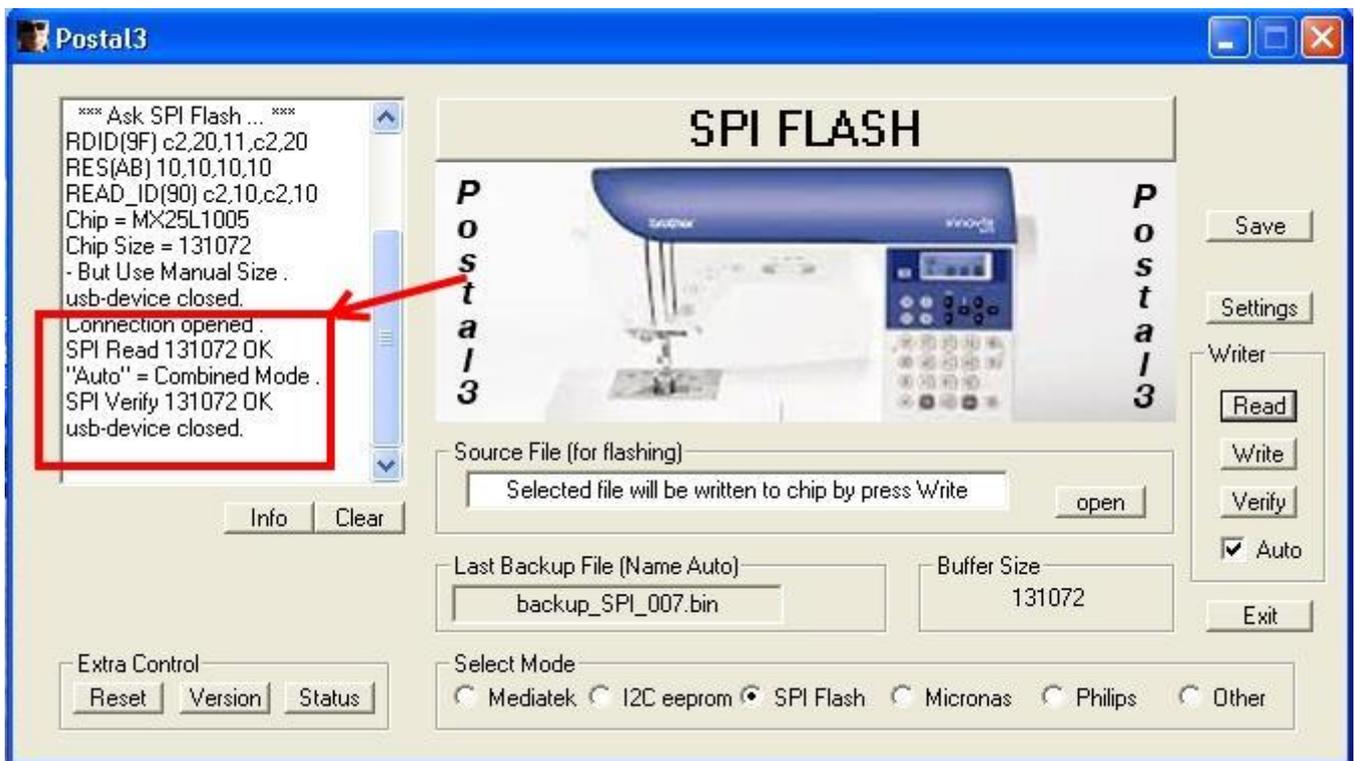
Снимаем галки EEPROM и DataFlash, выбираем **MX25L1005**, в нашем примере **28K (M25P10)**. Снятые галки в окошках позволяют работать с микросхемами памяти **SPI flash** с объемами от **64KB** до **64MB**. Галка, поставленная в окошке **EEPROM**, разрешает работу с микросхемами памяти **Big SPI** с объемами от **128B** до **128KB**. **Не путайте Big SPI EEPROM и SPI Flash – это разные типы микросхем, хотя и разводка выводов одинаковая.** Галка, поставленная в окошке **DataFlash**, открывает нам возможность работы с микросхемами памяти 45 серии. Микросхемы памяти **DataFlash 45 серии** вставляются в слот **SPI Flash** через переходник-адаптер, так как разводка выводов отличается от разводки микросхем памяти **SPI Flash 25 серии**. Работа с микросхемами памяти **DataFlash 45 серии** ничем не отличается от работы с микросхемами памяти **SPI Flash 25 серии**.



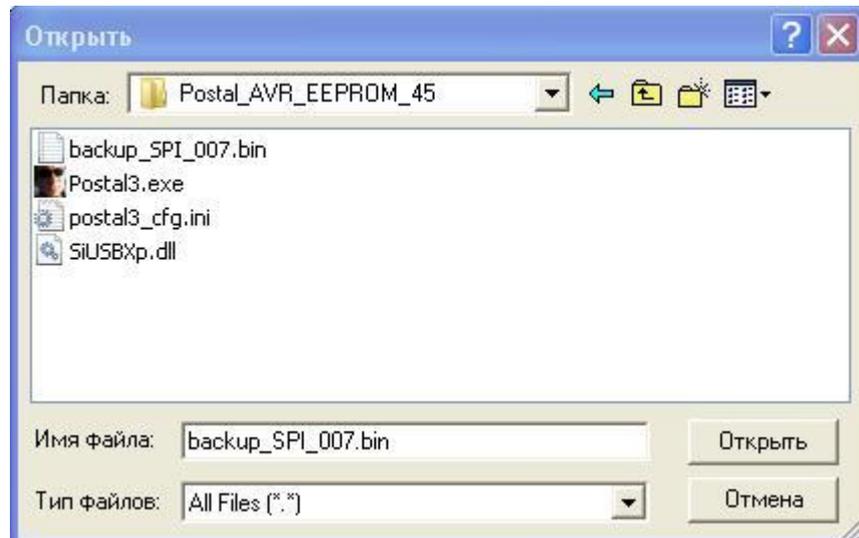
Жмём **Version**, в лог-окне видим, что всё в порядке, далее – жмём **Read**. Идёт чтение, внизу видно, под каким именем будет сохранённый считанный дамп: backup_SPI_006. каждое следующее считывание увеличивает номер "**backup_SPI**" на "1", пока не будет нажата кнопка **Reset**



После чтения происходит верификация (повторное считывание и сравнение), если галочка в окошке **Auto** не убрана.



Результат проверки, в нашем случае для **MX25L1005** - «SPI Flash Verify 13107 OK», успешный.

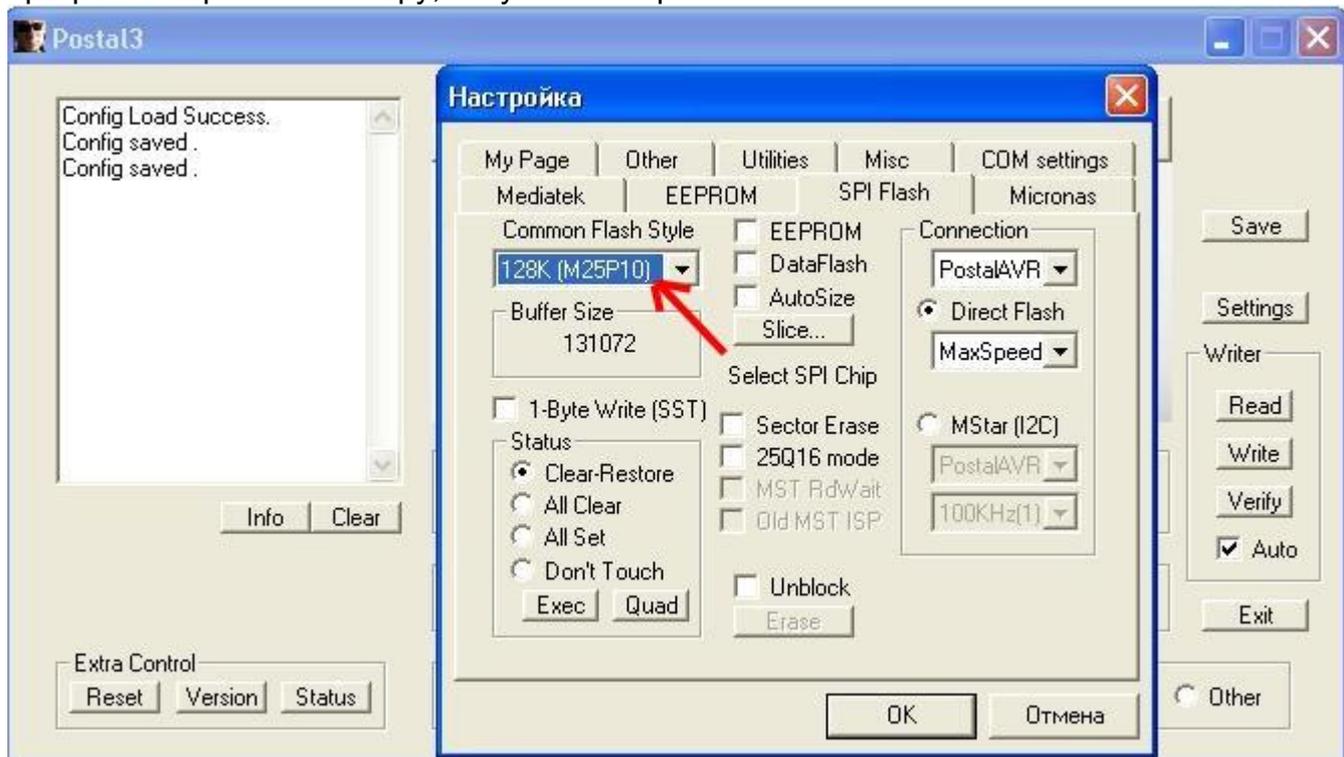


Считанный дамп находится в папке с софтом и в данном случае имеет название **backup_SPI_007**, вы можете его переименовать, например: **G41M-D20_MX25L1005**, т.е. вы указали название материнки и м/с, которая там стоит.

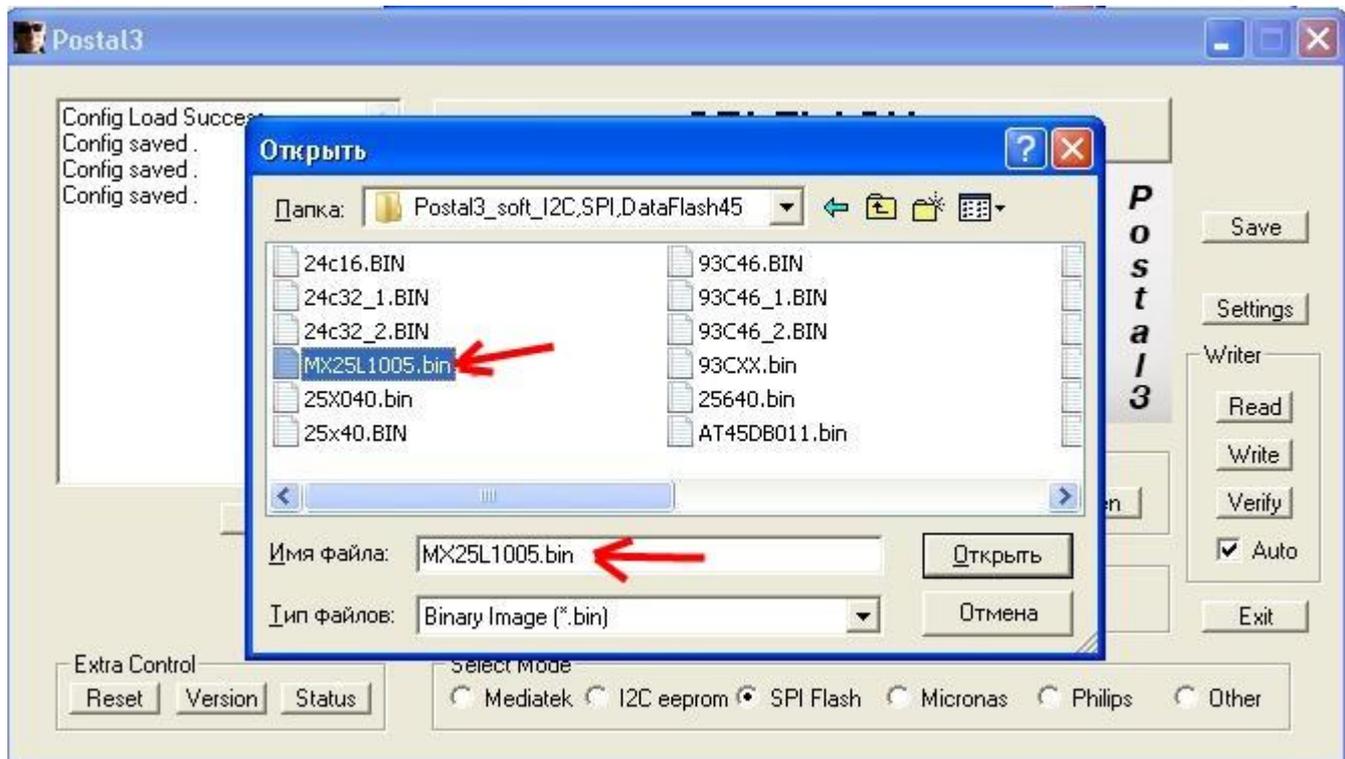
Запись микросхем памяти SPI Flash 25 серии.

Напряжение питания **3v3 – джемпер в положении 3v3.**

Вставляем микросхему памяти, например **MX25L1005** в слот **SPI Flash**, подключаем программатор к компьютеру, запускаем софт.



Выбираем для **MX25L1005 – 128K (M25P10)** и жмём **OK**.



Жмём **Open** и выбираем прошивку под данную микросхему, в нашем случае **MX25L1005.bin**. Далее нажимаем **Write**, если стоит галка в окошке **Auto**, сначала произойдёт стирание чипа, программа сообщит об успешном стирании: **Chip is Empty. Erase Not Need. May be DATA-Line disconnected?**, жмём **OK**. Дальше произойдёт запись чипа выбранным дампом.



По окончании записи и проверки мы увидим **SPI Verify 131072 OK** – это значит, что данные операции прошли успешно.

Микросхемы **SPI Flash** перед записью обязательно требуют стирания. Эту операцию можно сделать вручную, или в автоматическом режиме поставив галку в окошке **Auto**.

