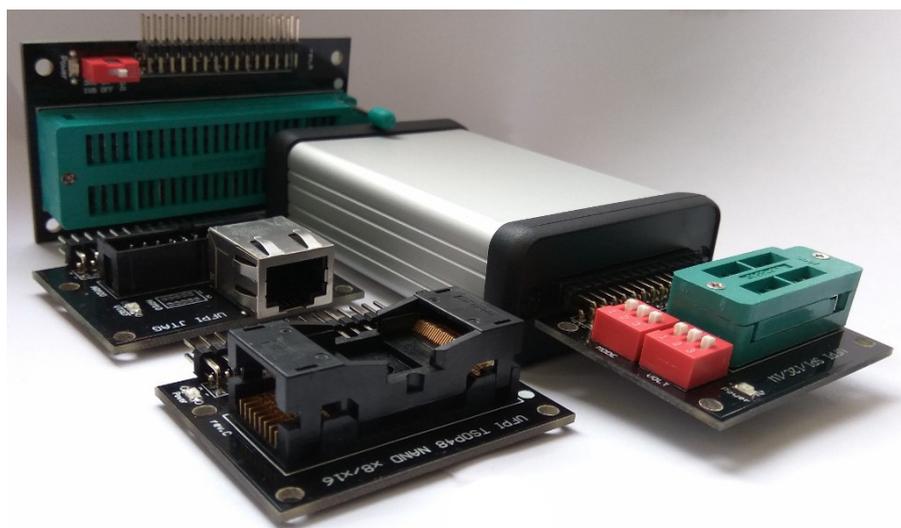


UFPI Программатор

Руководство Пользователя



Редакция от 18.06.21

Оглавление

Введение.....	5
1 Загрузка и установка.....	6
1.1 Загрузка программного обеспечения.....	6
1.2 Установка драйвера.....	7
2 Описание вкладок и меню.....	8
2.1 Главная вкладка.....	8
2.2 Панель инструментов.....	12
2.3 Главное меню.....	16
2.4 Лог.....	19
3 Настройки программы.....	20
3.1 Вкладка UFPI.....	21
3.2 Вкладка GUI.....	24
3.3 Медиа.....	25
3.4 Клавиши.....	26
3.5 Модули.....	28
3.5.1 NAND.....	29
3.5.2 OneNAND.....	32
3.5.3 Serial NAND.....	34
3.5.4 NOR.....	35
3.5.5 SDMMC.....	36
3.5.6 SPI.....	38
3.5.7 EEPROM.....	39
4 Модули.....	40
4.1 LOGGER.....	41
4.2 NAND.....	43
4.2.1 Опции. Выбор CE (чипа).....	43
4.2.2 Цикл доступа.....	44
4.2.3 «Чтение в» и «Запись из».....	44
4.2.4 Модель.....	45
4.2.5 Режим (доступа).....	45
4.2.6 Плохие блоки.....	45
4.2.7 ECC.....	47
4.2.8 Кнопки управления списком плохих блоков.....	47
4.2.9 Доп. функции. (некоторые пункты появляются после чтения ID).....	47
4.2.10 Приложение. Плохие блоки (BB).....	50
4.2.11 Приложение. Таблица плохих блоков (BBT).....	56
4.2.12 Приложение. ECC.....	59
4.2.13 Приложение. Анализ дампа. Подготовка дампа. Получение «чистого» дампа.....	61
4.2.14 Приложение. Анализ дампа при чтении ID и запись «чистого» дампа с управлением «Использовать резервную зону».....	63
4.2.15 Описание пунктов в логе при работе.....	65
4.3 SD/eMMC.....	66
4.3.1 Опции.....	67
4.3.2 «Чтение в» и «Запись из».....	67
4.3.3 Режим.....	67
4.3.4 Модель.....	67
4.3.5 Раздел.....	67
4.3.6 4-Бит режим / 1-Бит режим.....	67

4.3.7	Отчет Smart.....	67
4.3.8	Конфиг загрузки.....	67
4.3.9	Полный бэкап.....	68
4.3.10	Бэкап платформы.....	68
4.3.11	Доп. функции. (некоторые пункты появляются после чтения ID).....	69
4.3.12	Приложение. Регистры CID, CSD, eCSD, OCR.....	74
4.3.13	Приложение. Разделы USER, BOOT1, BOOT2, RPMB, GPP1, GPP2, GPP3, GPP4.....	76
4.3.14	Приложение. ISP подключение.....	77
4.3.15	Описание пунктов в логе при работе.....	79
4.4	NOR.....	80
4.5	1-Wire.....	82
4.6	EEPROM I2C.....	83
4.7	EEPROM SPI.....	84
4.8	EEPROM Microwire (3-Wire).....	85
4.9	SPI Flash.....	86
4.10	UART.....	90
4.11	Serial NAND.....	92
4.12	OneNAND.....	95
4.13	BDM.....	97
4.14	JTAG.....	99
4.15	Mount File.....	100
4.16	Mount Chip.....	101
5	Режим (доступа).....	102
6	Управление питанием.....	106
7	Файлы устройств UDEV.....	108
8	GZIP.....	111
9	Работа с буфером.....	112
10	Тестовый файл.....	114
11	Файловый менеджер и монтирование ФС.....	115
12	Контейнеры.....	118
13	Самодиагностика программатора.....	120
14	Работа со скриптами.....	121
14.1	Скрипты.....	122
15	FAQ.....	124
15.1	По работе с форумом.....	125
15.2	Общие вопросы.....	127
15.3	Принято называть.....	130
15.4	Питание при работе.....	131
15.5	NAND.....	131
15.6	EMMC.....	132
15.7	SPI.....	132
15.8	UART.....	132
16	Приложение.....	134
16.1	Программаторы, сокет, подбор и совместимость.....	134
16.1.1	Комплекты NAND.....	135
16.1.2	Комплекты SD/eMMC.....	136
16.1.3	Комплекты NOR.....	136
16.1.4	Остальные комплекты.....	139
16.1.5	Советы и полезная информация для выбора вашего комплекта адаптеров к программатору.	139
16.2	Примеры.....	142

16.2.1 Приложение. Составление конфигурационного файла NAND Flash.....	142
16.2.2 Приложение. Составление конфигурационного файла SPI Flash.....	144
16.2.3 Приложение. Составление конфигурационного файла NOR Flash.....	145
16.2.4 Приложение. Файловый менеджер и монтирование ФС.....	147
16.2.5 Приложение. Вычитывание и запись разделов / регионов.....	149
16.2.6 Приложение. Полный бэкап с помощью режима доступа и .UDEV.....	150
16.2.7 Приложение. Работа с разделами. Разделение, замена, запись.....	151
16.2.8 Приложение. Работа с «Бэкап платформ».....	153
16.2.9 Приложение. Проверка хэшей, ключей, восстановление и извлечение ключей и BOOT на примере платформы QV14/15.....	154
16.2.10 Приложение. Исправление дампа и запись дампа NAND на примере Samsung D5500....	157
16.2.11 Приложение. Контейнеры. Сборка и запись.....	164
16.2.12 Приложение. Использование LOGGER и анализ протокола.....	166
16.3 Распиновки и схемы.....	167
16.3.1 eMMC EASY JTAG 8-Bit Socket schematics.....	168
16.3.2 eMMC E-Mate Pro 8-Bit Socket schematics.....	169
16.3.3 eMMC E-Mate X 8-Bit Socket schematics.....	170
16.3.4 eMMC ICFRIEND 8-Bit Socket schematics.....	171
16.3.5 SD/eMMC Socket schematics.....	172
16.3.6 SPI/I2C/mWire/1Wire DIP24 Socket schematics.....	173
16.3.7 LOGGER Socket schematics.....	174
16.3.8 NAND Socket schematics.....	175
16.3.9 NOR TSOP48 Socket schematics.....	176
16.3.10 NOR TSOP56 Socket schematics.....	177
16.3.11 OneNAND Socket schematics.....	178
16.3.12 UART/GPIO Socket schematics.....	179
16.3.13 JTAG/BDM Socket schematics.....	180

Введение.

Данное руководство предназначено для ознакомления с основами и демонстрации работы в программно-аппаратном комплексе UFPI. Для опытных пользователей может быть так же полезно ознакомиться с [FAQ](#) разделом руководства.

UFPI (Universal Flash Programming Interface) - это устройство с питанием от USB на базе современного защищенного двухъядерного процессора 200 МГц с High-Speed USB Phy (480 Мбит / с). Основное назначение этого интерфейса и программного обеспечения - программирование памяти Flash или EEPROM и работа с основными последовательными протоколами, такими как JTAG, SPI, I2C, NAND, eMMC, SD, 1W и так далее. Основные цели - максимальная скорость, простота интерфейса и максимальное количество возможностей для пользователя. Скриптовый язык с синтаксисом, подобным C, и прямым доступом к программному обеспечению, ОС и аппаратным функциям блока UFPI позволяют пользователю создавать (при необходимости) свои собственные полнофункциональные апплеты сценариев без использования IDE, компиляторов и другого оборудования для программирования. Возможность использования внешних файлов конфигурации для ИС позволяет использовать любые необходимые параметры, команды и частоту, а также создавать любую необходимую конфигурацию даже для «неподдерживаемых» микросхем.

Ключевые особенности

- Очень быстрая скорость. Например работает с NAND быстрее, чем «самый быстрый» ChipProg-481.
- Драйвера для Windows не нужны. Plug-and-Play для Win10 и Win8 (устройство WCID).
- Питание от USB с разрешенными USB VID и PID.
- Двунаправленный ввод-вывод с уровнями напряжения от 1 В до 5 В
- Сокеты Plug-and-Play. Программное обеспечение автоматически выбирает подходящий режим и напряжение при подключении.
- Безопасные и надежные обновления Вох. Зеркальное обновление с AES256.
- Поддержка параметров пользовательского чипа - геометрия, напряжение, время доступа и т. д.
- Гибкая система лицензирования. Вы можете использовать те модули, которые вам нужны.
- Мощный C-подобный скриптовый язык.
- Многоязычный пользовательский интерфейс.
- Использование контейнеров данных из нескольких частей с определяемым пользователем сжатием и шифрованием.
- Функции монтирования разделов ФС.
- Поддержка распространенных таблиц разделов и алгоритмов ECC.
- Встроенный шестнадцатеричный редактор.
- Фиксированные режимы для подключенных сокетов.

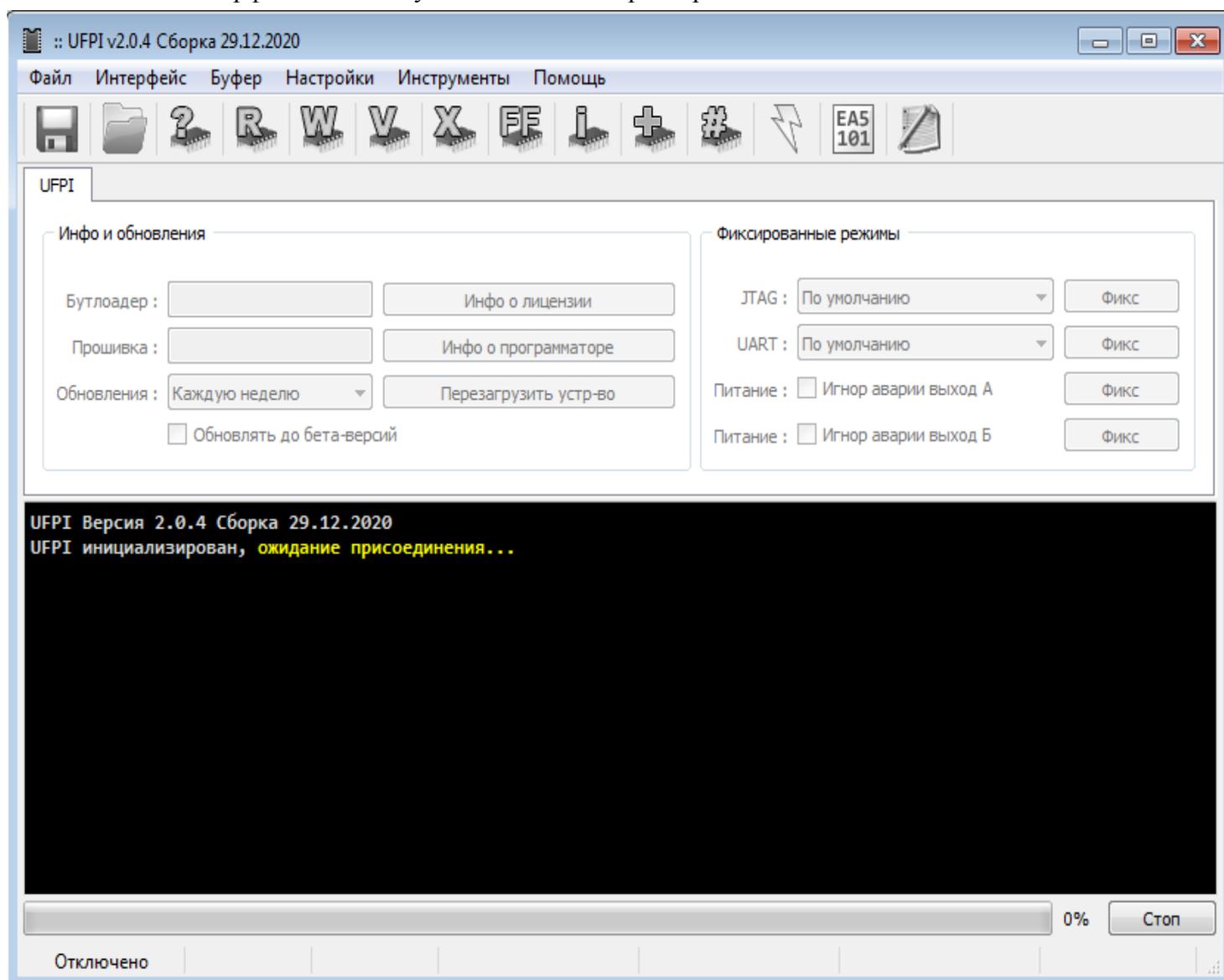
1 Загрузка и установка.

Системные требования.

- Любой настольный или портативный компьютер с процессором x86 / x64 1 ГБ ОЗУ (рекомендуется 2 ГБ и более)
- Операционная система Windows XP/Vista/7/8/10 (а так же Linux_x64 и MacOS_x64)
- Высокоскоростной порт USB 2.0 (480 Мбит / с)

1.1 Загрузка программного обеспечения.

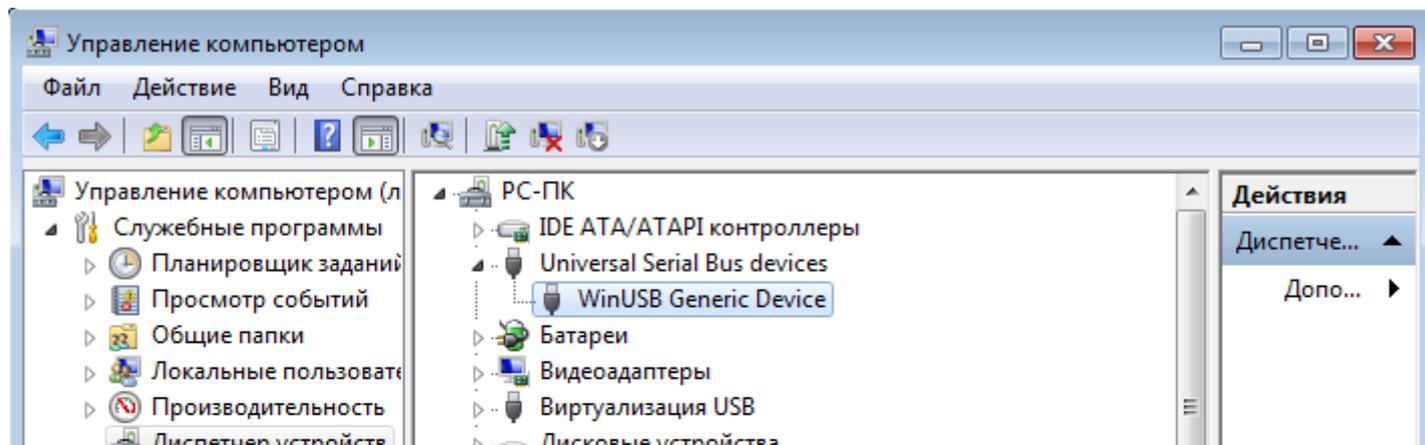
Загрузить программное обеспечение UFPI по [ссылке](#), распаковать и запустить ufpi.exe. Нет необходимости в установке. На снимке экрана ниже показано программное обеспечение UFPI с отключенным интерфейсом или неустановленными драйверами.



1.2 Установка драйвера.

Программатор UFPI - это USB устройство WCID (Windows Compatible ID).

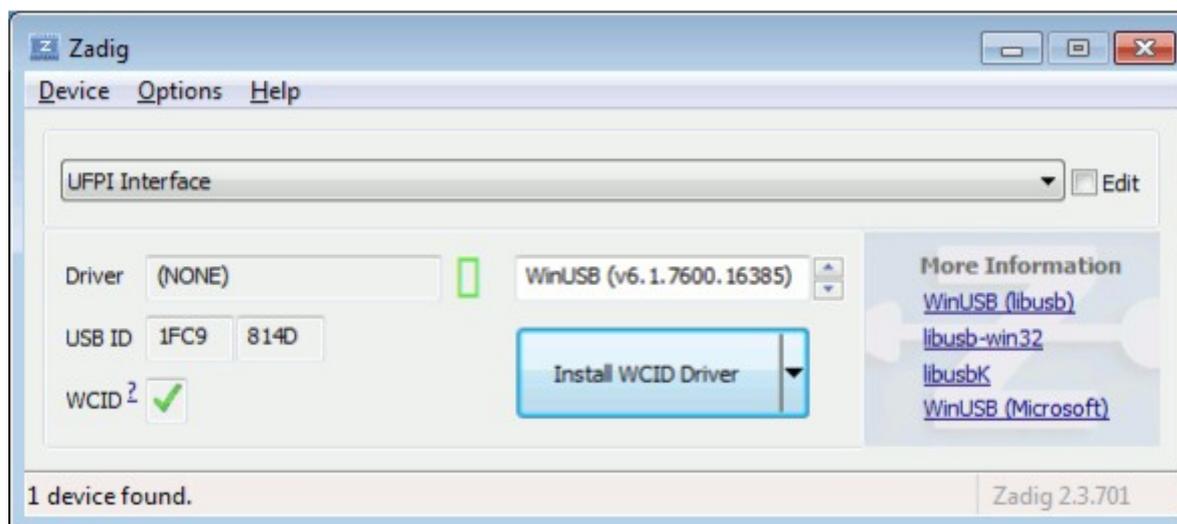
В системе Windows 8 или 10 драйвер будет установлен автоматически, даже без подключения к Интернету. С подключенным интерфейсом UFPI в диспетчере устройств должно появиться новое USB-устройство. На снимке экрана ниже показан диспетчер устройств с подключенным программатором в Windows 10.



В Windows XP / Vista / 7 без установленной поддержки WCID.

- Загрузить [Zadig](#).
- Подключить интерфейс UFPI.
- Нажать «Установить драйвер WCID».

Ниже показана установка драйвера с помощью Zadig.

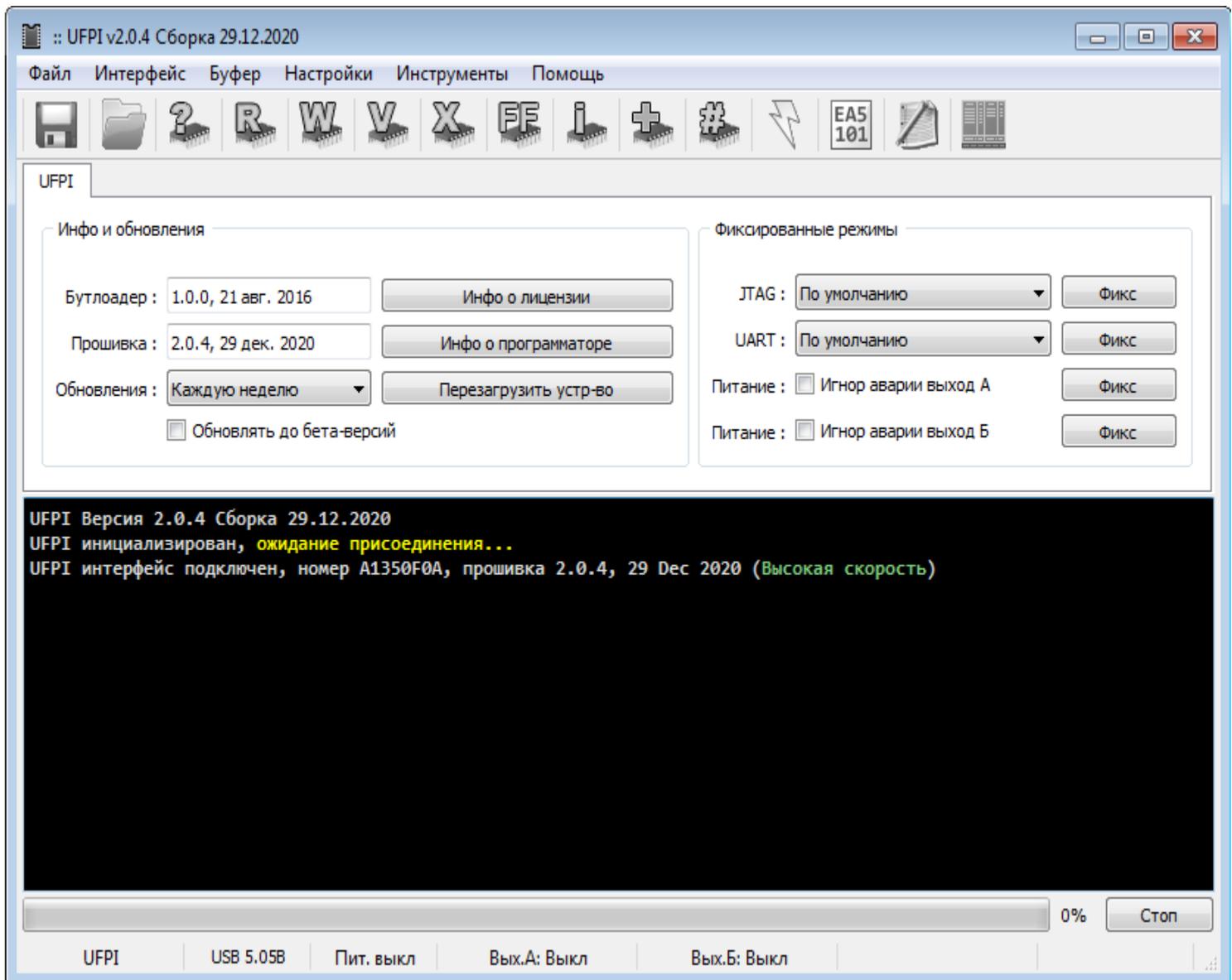


Linux.

Для установки в Linux необходимо запустить в терминале командой `sudo ./ufpi.run` и запустить в меню «Интерфейс — Инсталлировать».

2 Описание вкладок и меню.

2.1 Главная вкладка.



Напряжение USB

USB 5.05В

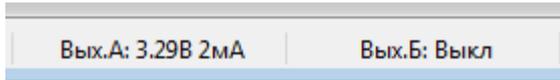
Напряжение питания программатора.

Выставленное напряжение

Пит. выкл

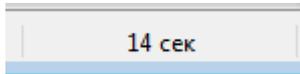
Показатель выставленного на сокет напряжения.

Напряжения на выходах питания сокета



Показания напряжения и тока на соответствующих каналах питания бокса.

Время до окончания текущей операции



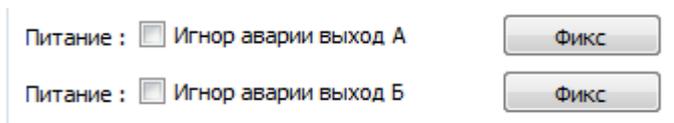
Показатель расчетного времени до окончания операции.

Скорость текущей операции



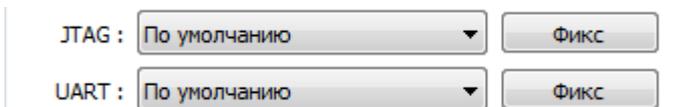
Показатель скорости текущей операции.

Отключение защиты каналов питания сокета



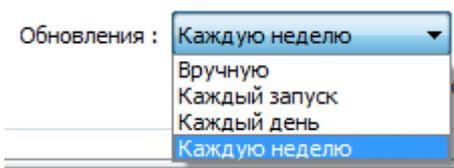
Игнор аварии выхода А и Б. Для установки режима, требуется установить соответствующий каналу флаг и нажать кнопку «Фикс».

Режимы эмуляции



Есть возможность эмуляции CMSIS DAP. Для этого нужно выбрать в меню JTAG соответствующий пункт и нажать кнопку "Фикс". При вставленном сокете JTAG в диспетчере устройств появится устройство соответствующее CMSIS DAP. При этом программа перестанет определять программатор. Чтобы отключить эмуляцию в таком случае, достаточно подключить программатор без сокета.

Расписание обновления



Выбор регулярности проверки обновлений.

Режим обновления

Обновлять до бета-версий

Крайне не рекомендуется для неподготовленных пользователей. Всем, кому требуется работать, а не экспериментировать, не рекомендуется устанавливать этот флаг. По умолчанию этот флаг снят.

Инфо о боксе и лицензии.

Список активированных лицензий и других параметров.

Инфо о лицензии

Инфо о программаторе

Дополнительная информация о боксе включает в себя:

UFPI информация о владельце: ***

UFPI активированные модули: ***

UFPI ревизия платы ***

UFPI микросхема USB питания ***

UFPI трансляторы уровней ***

UFPI 3V3 калибровочное значение *** мВ

2.2 Панель инструментов.



Кнопка "Открыть файл для чтения".



Кнопка "Открыть файл для записи".



Кнопка "Прочитать ID".



Кнопка чтения ID и проведение инициализации микросхемы. Обычно с этой кнопки всегда начинается работа с микросхемой, получение информации о ней, а так же установка параметров программы и программатора.

Кнопка "Чтение микросхемы".



Кнопка чтения памяти микросхемы. Чтение производится в файл для чтения или буфер, в зависимости от [настройки](#).

Кнопка "Запись микросхемы".



Запись в память микросхемы из файла для записи или из буфера, в зависимости от [настройки](#).

Кнопка "Верификация микросхемы".



Сверка содержимого памяти микросхемы с файлом для записи или буфером, в зависимости от [настройки](#).

Кнопка "Стереть микросхему".



Кнопка "Проверка чистоты".



Кнопка "Информация о микросхеме".



Кнопка "Добавить/редактировать микросхему".

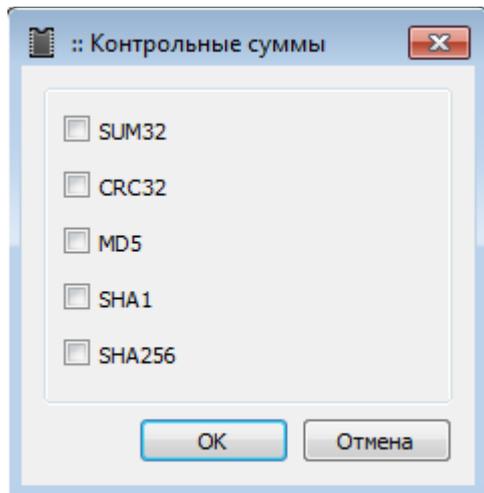


Конфигуратор параметров микросхемы. Описано в [приложении](#).

Кнопка "Контрольные суммы".



Опции подсчета контрольной суммы при чтении/записи/верификации. Контрольная сумма выводится в лог.



Кнопка «Питание».



Включить/выключить питание сокета, а так же настройка параметров питания. Описание в соответствующем [разделе](#).

Кнопка «Редактировать Буфер».



Встроенный HEX редактор буфера. Описание в соответствующем [разделе](#).

Кнопка «Выполнить Скрипт».



Запуск/остановка скрипта. Описание в соответствующем [разделе](#).

Кнопка «Файловый Менеджер»



Файловый менеджер. Монтирование разделов, файловых систем и работа с их содержимым. Описание в соответствующем [разделе](#).

2.3 Главное меню.

Файл Интерфейс Буфер Настройки Инструменты Помощь

По порядку:

Файл.

Открыть файл для чтения	Ctrl+S
Открыть файл для записи	Ctrl+O
Редактировать файл для чтения	Alt+R
Редактировать файл для записи	Alt+W
Открыть устройство	Ctrl+D
Заккрыть устройство	Alt+D
Открыть контейнер	Ctrl+K
Заккрыть контейнер	Alt+K
Выход	Ctrl+Q

Открыть файл для чтения/записи — Дублирует кнопки на панели инструментов.

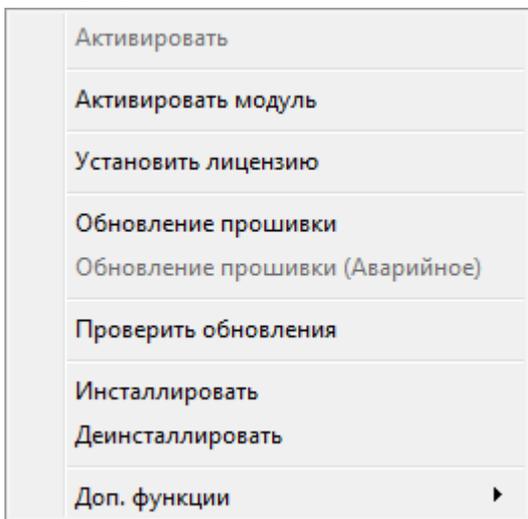
Редактировать файл для чтения/записи — Открывает во встроенном HEX редакторе выбранный файл, не загружая его в буфер.

Открыть/заккрыть устройство — Работа с файлами .UDEV. Описание в соответствующем [разделе](#).

Открыть/заккрыть контейнер — Работа с файлами контейнеров. Описание в соответствующем [разделе](#).

Выход — Заккрыть программу.

Интерфейс.

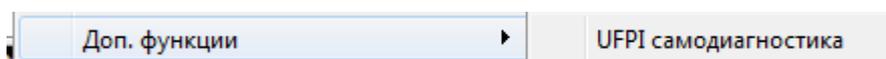


Установить лицензию — При покупке лицензии модуля, для его активации, через это меню нужно выбрать файл, полученный по почте.

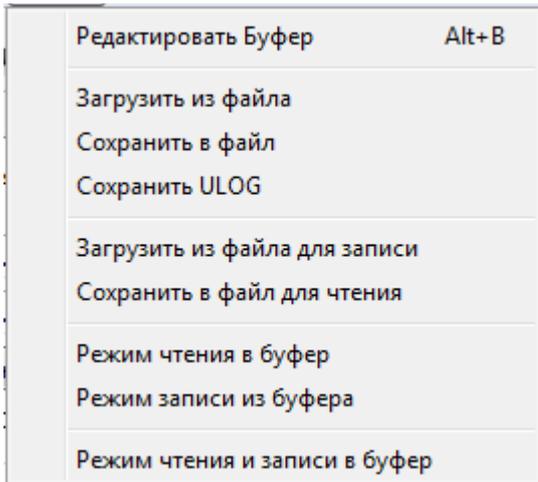
Обновление прошивки (Аварийное) — Особо частый вопрос, [описание процедуры](#).

Инсталлировать/Деинсталлировать — Подсказка или установка/удаление ПО в ОС.

Доп. функции — Содержат [инструмент для самотеста](#).



Буфер.



Редактировать Буфер — Дублирует кнопку на панели инструментов. Описание в соответствующем [разделе](#).

Загрузить/сохранить в файл и из файла — Соответственно открывает диалог выбора файла и загружает из файла в буфер, или наоборот сохраняет содержимое буфера в файл.

Сохранить ULOG — Сохранение результата записи [LOGGER](#) в буфер в файл формате ULOG.

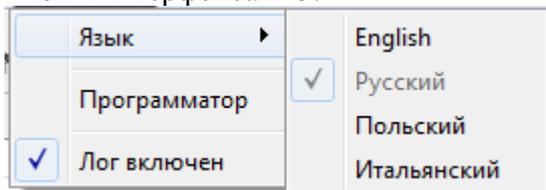
Загрузить/сохранить в/из файл для чтения/записи — Соответствует названию пункта в применении файлов указанных в полях "чтение в" и "запись из" на вкладке модуля.

Режим чтения/записи в/из буфера — Читать/писать флеш в буфер. При этом поля для соответствующего пункта "чтение в" и "запись из" становятся не активны.

Режим чтения и записи в буфер — Активирует/деактивирует сразу оба предыдущих пункта.

Настройки.

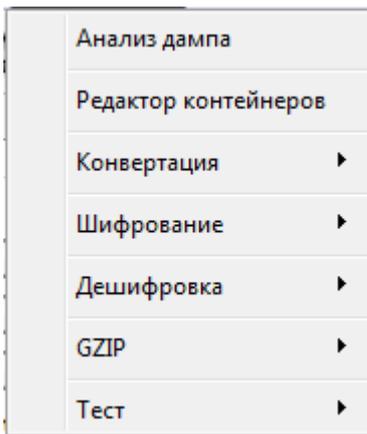
Язык интерфейса ПО.



[Настройки программы](#) — Описание настроек.

[Лог](#) включен — Флаг сохранения логов в текстовом виде в папке Logs в папке с программой.

Инструменты.



Анализ дампа — Инструмент подготовки дампа NAND к записи. Содержит настройки корректировки [ЕСС](#), [таблицы ББ](#), разделов и прочее. В [приложении](#) имеется пример использования.

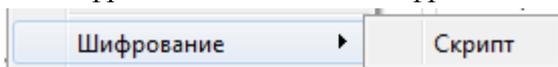
Редактор контейнеров — Открыть форму редактора контейнеров. Описание в соответствующем [разделе](#).

Конвертация:

S-Records — формат применяемый в MCU при работе через BDM

ULOG — (внутренний формат LOGGER UFPI) в Sigrok .SR (формат PulseView) конвертация записанного логером потока в формат для просмотра в Pulseview соответственно из файла или буфера, куда был произведен захват.

Шифрование — Опции шифрование скрипта. Описание в соответствующем [разделе](#).

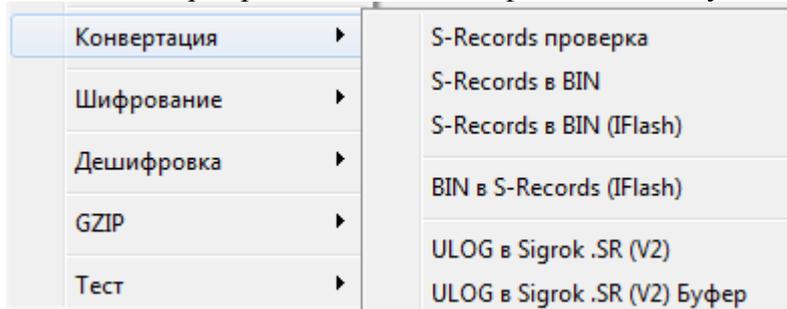


Дешифровка — Расшифровка дампов .nttc, .userc.



[GZIP](#) — Упаковка/распаковка файла gzip.

Тест — Формирование тестового файла индивидуально под подключенную микросхему.

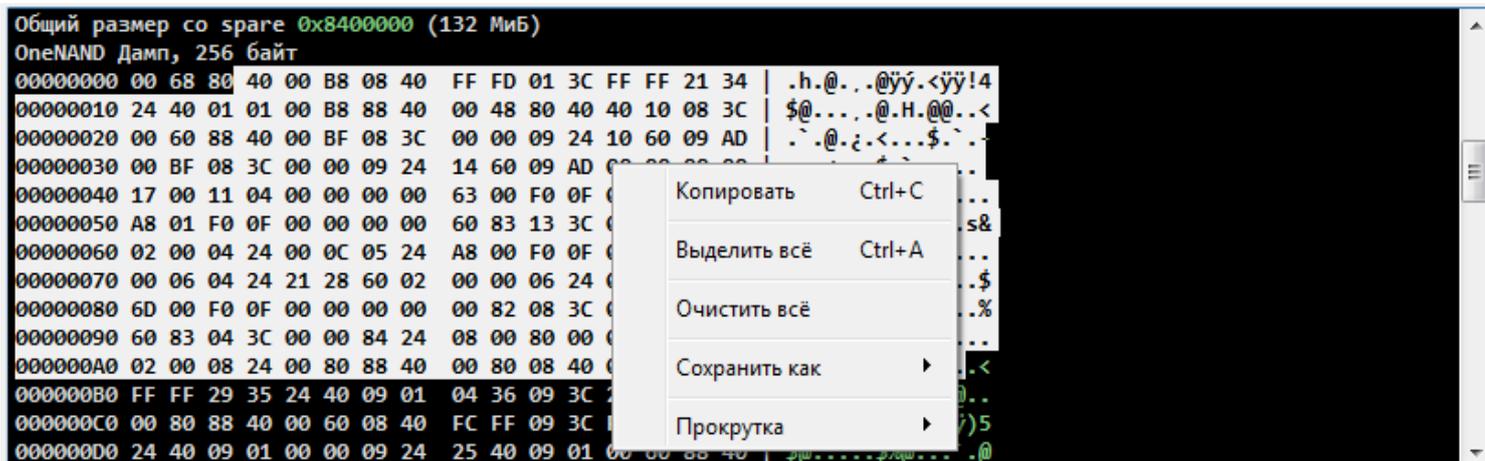


Помощь.

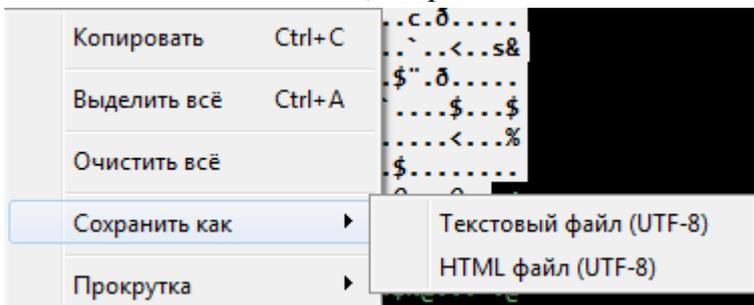
Информация о программе и файл справки.

2.4 Лог.

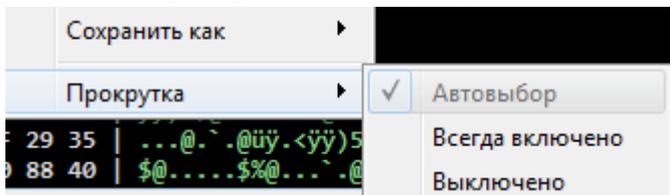
В логе выводится информация обо всех происходящих операциях. Текст из лога можно выделять и копировать в буфер обмена.



Так же можно очистить, сохранить в текстовый или HTML файл



Либо выбрать режим автопрокрутки



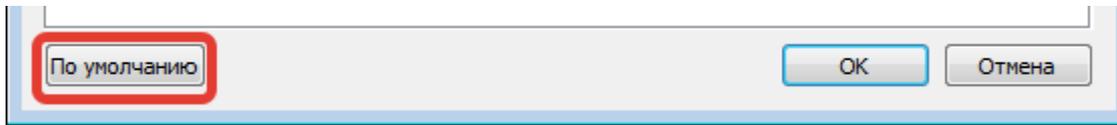
По умолчанию стоит «Автовыбор». В этом режиме работает автопрокрутка, если полоса прокрутки опущена в самое нижнее положение и нет других выделенных фрагментов в лог.

Режимы «Всегда включено» и «Выключено» соответствуют названиям.

3 Настройки программы.

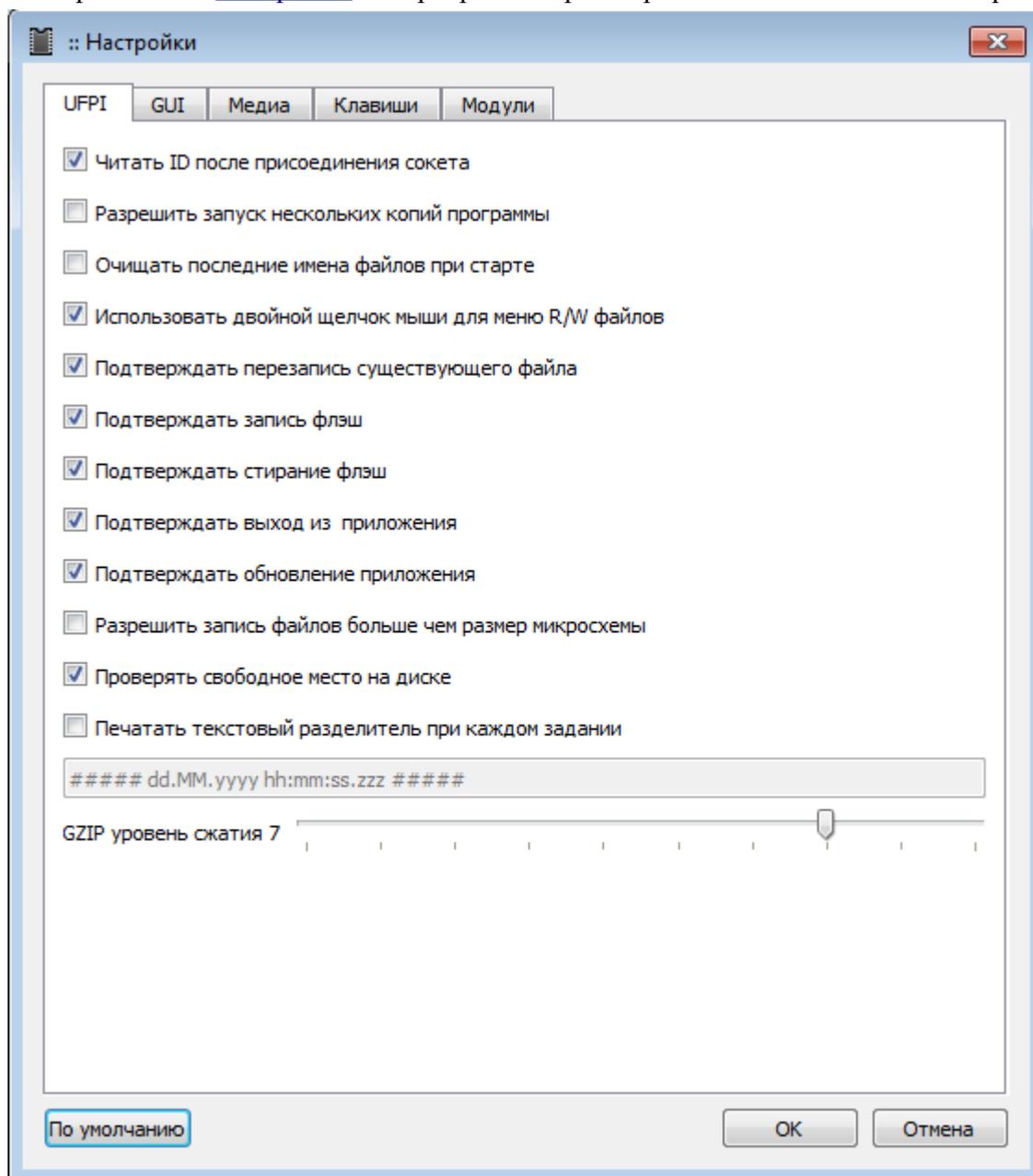
Все настройки сохраняются папке рядом с программой в файле ufr1.ini

Для сброса настроек можно либо удалить файл и тогда при следующем запуске он создастся автоматически с настройками по умолчанию, либо внизу слева в окне «Настройки» есть кнопка «По умолчанию», которая выставит флаги по умолчанию на текущей вкладке.



3.1 Вкладка UFPI.

Через меню «[Настройки](#) — Программатор» открывается главное окно настроек на вкладке UFPI.



По пунктам:

Читать ID после присоединения сокета.

При установленном флаге, сразу при подключении сокета, или бокса с сокетом, выполняется команда чтения ID микросхемы. Если не установлен, то читать ID будет только по кнопке «[Прочитать ID](#)». (По умолчанию установлен)

Разрешить запуск нескольких копий программы.

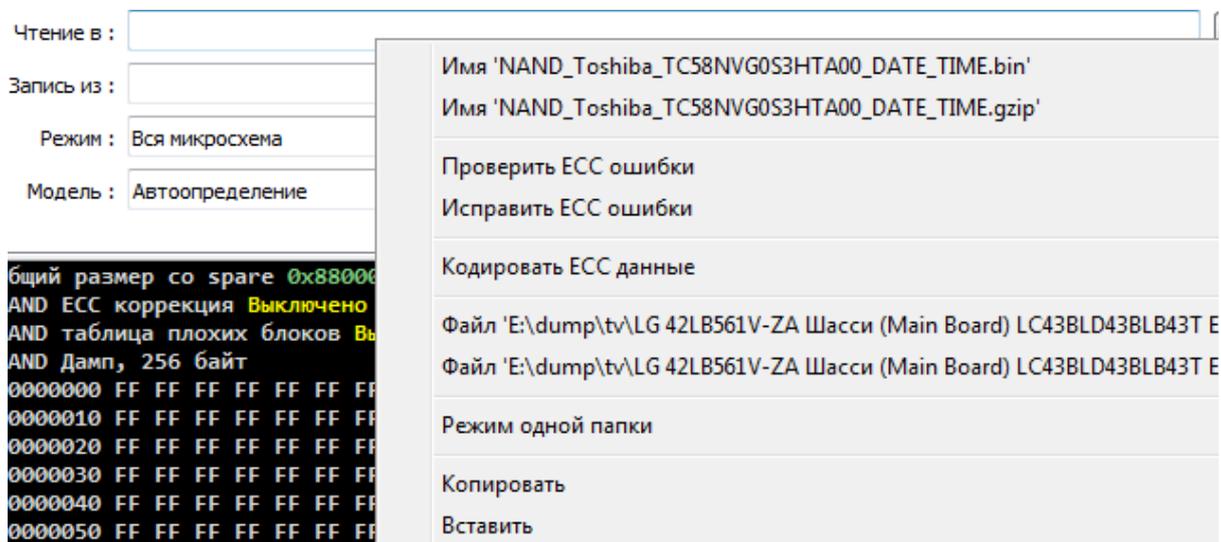
Может быть полезна, если потребуется параллельно запускать скрипт, не требующих подключения бокса, а программа уже запущена. Но это не дает возможности работать боксом со второй копии запущенной программы, даже если боксов физически подключено два. (По умолчанию не установлен)

Очищать последние имена файлов при запуске.

При запуске программы очищает поля «Чтение в» и «Запись из». Тогда при выборе файла для чтения/записи, будет открываться папка расположения программы UFPI. (По умолчанию не установлен)

Использовать двойной щелчек мыши для меню R/W файлов.

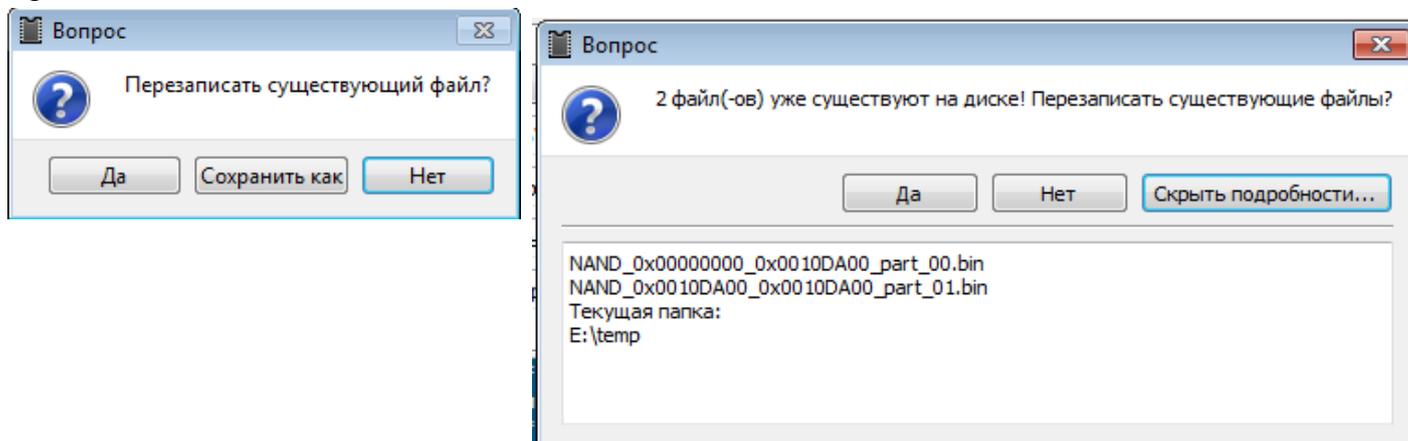
По двойному клику в поле «Чтение в» и «Запись из» меню позволяет задавать автоимена, проверять и корректировать ECC, выбирать файлы из истории и другое... (По умолчанию установлен)



Подтверждать перезапись существующего файла.

Снятие флажка позволяет избежать запросов на перезапись существующего файла, например при операции чтения в файл. (По умолчанию установлен)

При активации выводит диалог.



Подтверждать запись флеш.

При снятом флажке программа не запрашивает подтверждения на запись во флеш. (По умолчанию установлен)

Подтверждать стирание флеш.

При снятом флажке программа не запрашивает подтверждения на стирание флеш. (По умолчанию установлен)

Подтверждать выход из приложения.

При снятом флажке программа не запрашивает подтверждения на выход из приложения. (По умолчанию установлен)

Подтверждать обновление приложения.

При снятом флажке программа будет обновляться без подтверждения. (По умолчанию установлен)

Разрешить запись файлов больше чем микросхемы.

Со снятым флажком при попытке записать дамп больше размера флеш, операция будет прервана с описанием причины в логе. Если флажок установлен, то запись пройдет, но в логе размер записываемый размер будет выделен [цветом предупреждения](#). (По умолчанию не установлен)

Проверять свободное место на диске.

При установленном флажке, перед операцией чтения будет оценено свободное место на целевом диске и если его недостаточно, то выдано предупреждение. (По умолчанию установлен)

Печатать текстовый разделитель при каждом задании.

При установленном флаге печатает заданный ниже в окошке разделитель между операциями. Пример с датой и временем представлен (По умолчанию не установлен)

Печатать текстовый разделитель при каждом задании

dd.MM.yyyy hh:mm:ss.zzz

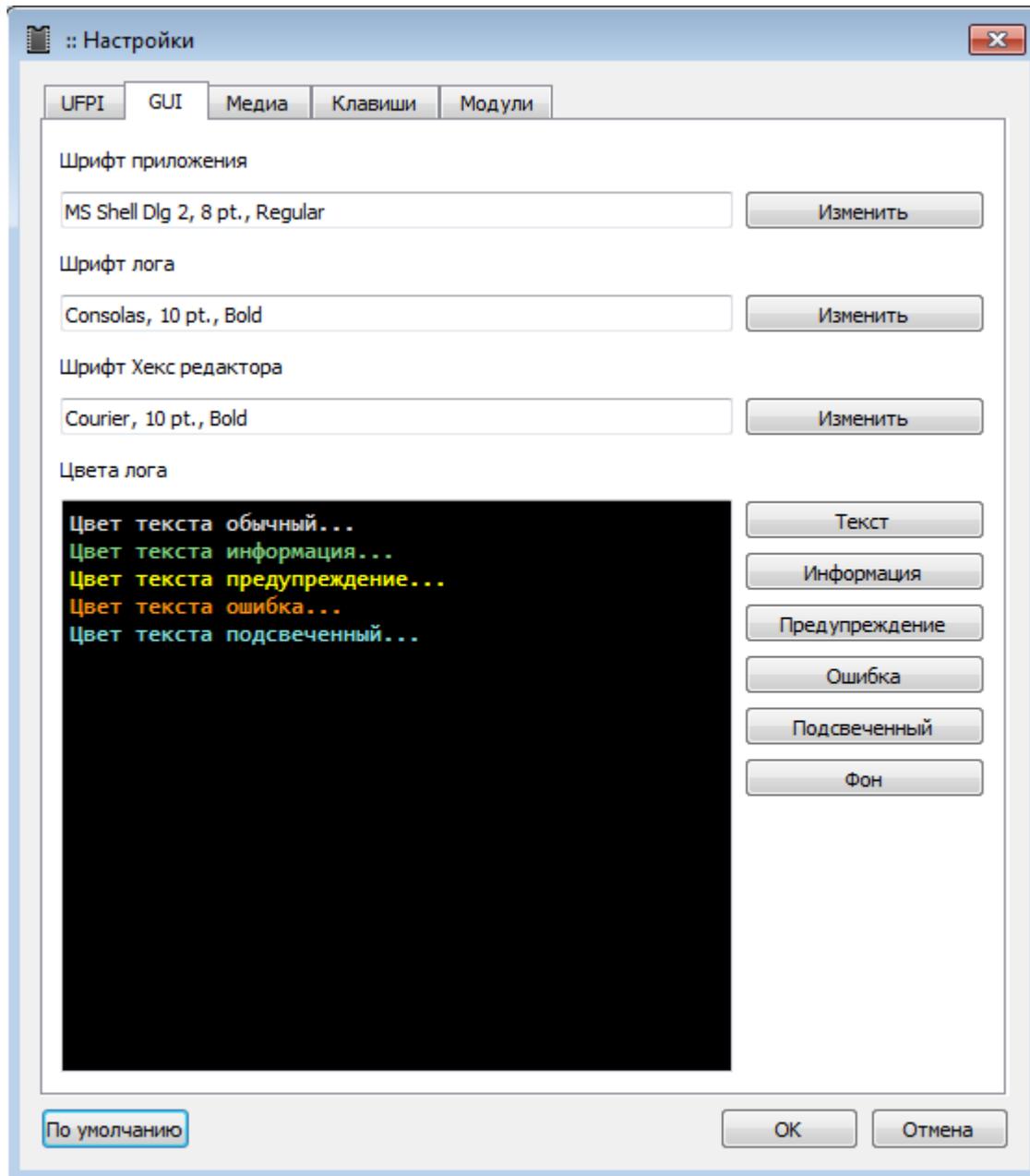
GZIP уровень сжатия.

Задать уровень сжатия для создаваемого архива при операции чтения в [gzip](#). (По умолчанию 7)

3.2 Вкладка GUI.

Настройки позволяют задать произвольные [цвета и шрифты интерфейса](#).

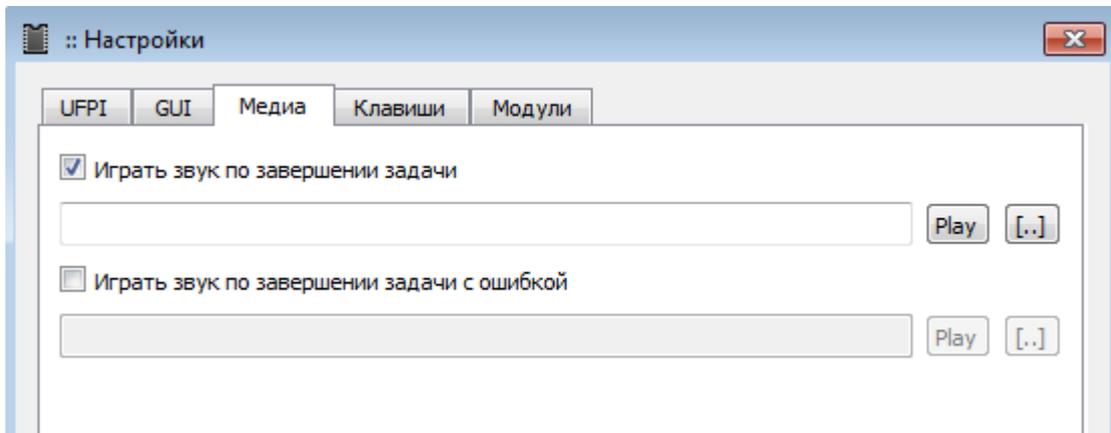
Напротив каждого пункта есть соответствующая кнопка, которая вызывает диалог выбора шрифта, или цвета. На снимке представлено окно с настройками по умолчанию.



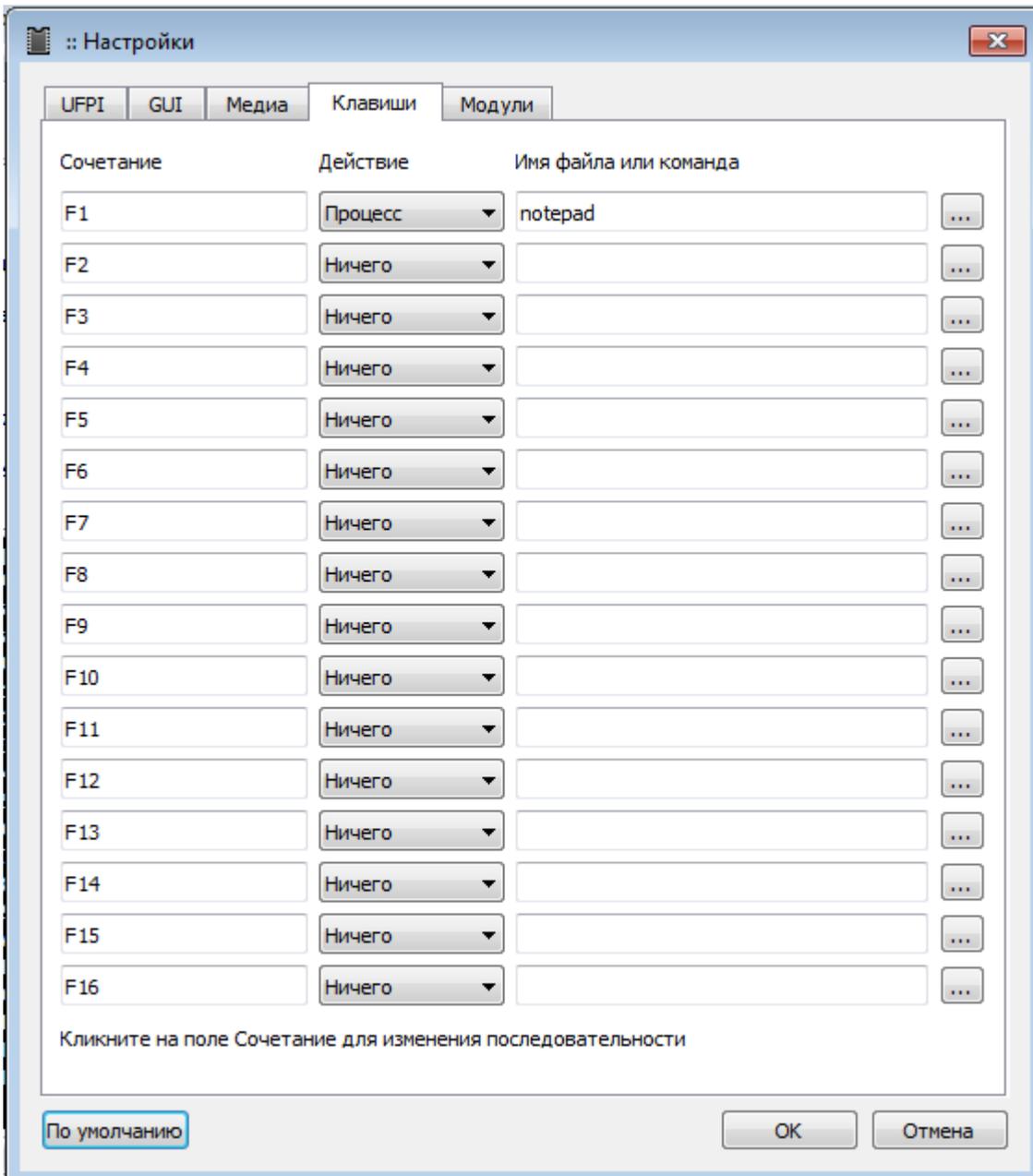
3.3 Медиа.

На этой вкладке можно задать и проверить выбранные звуковые эффекты на завершение задачи.

Например для Windows 7 можно выбрать из стандартных звуков C:\Windows\Media\Savanna\Windows Ding.wav.



3.4 Клавиши.

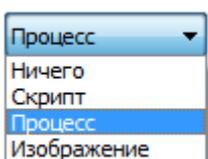


В этой вкладке можно задать до 16 горячих клавиш.

Клик по окошку в столбце «Сочетание».

Можно выбрать произвольное сочетание клавиш для действия.

В столбце «Действие».



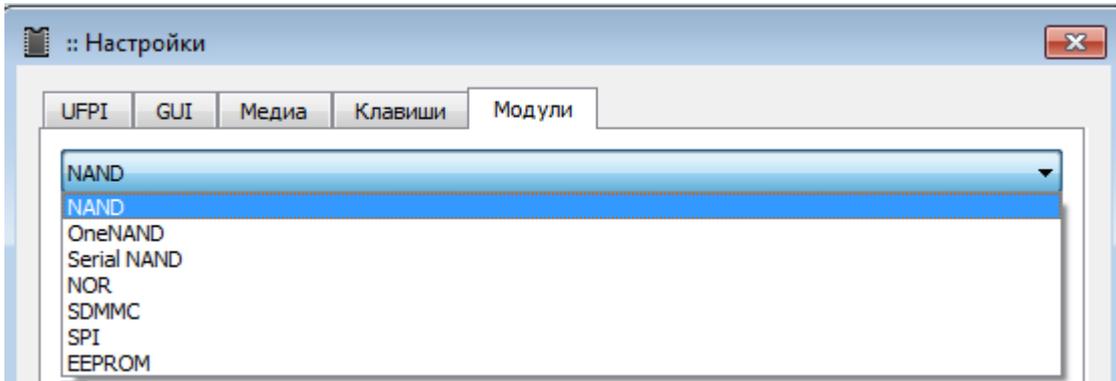
Выбирается тип исполнения: скрипт, процесс или изображение (открывается во встроенном вьювере, допустимы форматы .jpg, .png, .gif, .bmp).

В столбце «Имя файла или команда».

Выбирается объект запуска (скрипт, программу, или изображение), либо можно вписать команду с параметрами.

3.5 Модули.

Вкладка имеет настройки под каждый модуль, которые перебираются через комбобокс.



3.5.1 NAND.

NAND

Показать дамп первой страницы при чтении ID. Разме байт (0=размер страницы)

Показать полную информацию о микросхеме при чтении ID

Показывать ошибки для целых блоков

Показывать исправленные ECC ошибки

Не останавливать верификацию при ошибках

Не использовать сигнал RB. Задержка при чтении мкс

Показывать дамп первой страницы при чтении ID.

При установленном флаге, при успешном чтении ID флеш, будет отображено содержимое первой страницы. Пример выделен на картинке ниже. (По умолчанию установлен)

```

NAND сокет x8/x16, 4xCE, 3.30В
NAND ID 98F18015721608 (98F1801572160800), CE1 (*)
NAND Инфо Toshiba TC58NVG0S3HTA00, 128 Миб, стр. 2048+128, x8, SLC ECC-1/512, 3.30В, 25нс, 1xCE
NAND ECC коррекция Выключено
NAND таблица плохих блоков Выключено (Не используется) (Блок #0)
NAND Дамп, 256 байт
00000000 FF | yyyyyyyyyyyyyyy
00000010 FF | yyyyyyyyyyyyyyy
00000020 FF | yyyyyyyyyyyyyyy
00000030 FF | yyyyyyyyyyyyyyy
00000040 FF | yyyyyyyyyyyyyyy
00000050 FF | yyyyyyyyyyyyyyy
00000060 FF | yyyyyyyyyyyyyyy
00000070 FF | yyyyyyyyyyyyyyy
00000080 FF | yyyyyyyyyyyyyyy
00000090 FF | yyyyyyyyyyyyyyy
000000A0 FF | yyyyyyyyyyyyyyy
000000B0 FF | yyyyyyyyyyyyyyy
000000C0 FF | yyyyyyyyyyyyyyy
000000D0 FF | yyyyyyyyyyyyyyy
000000E0 FF | yyyyyyyyyyyyyyy
000000F0 FF | yyyyyyyyyyyyyyy

```

Размер.

Задать размер выводимой при чтении ID первой страницы. По умолчанию стоит значение 0x0 и размер страницы берется исходя из реального размера страницы флеш. Страницы NAND бывают достаточно большими и чаще хватает размера 0x100 - 0x200 для оценки.

Показать полную информацию о микросхеме при чтении ID.

Выводить в лог более детальную информацию. Пример на картинке ниже в сравнении с предыдущей.
(По умолчанию не установлен)

```

NAND сокет x8/x16, 4xCE, 3.30В
NAND ID 98F18015721608 (98F1801572160800), CE1 (*)
Производитель Toshiba
Название модели TC58NVG0S3HTA00
Шина и напряжение x8 3.30В
Время доступа 25нс
Адресные циклы 2/2 (Col/Row)
Размер страницы 0x800 (2048)
Размер spare 0x80 (128)
Страниц в блоке 0x40 (64)
Размер данных блока 0x20000 (131072) 128 КиБ
Размер блока со spare 0x22000 (139264) 136 КиБ
Количество блоков 0x400 (1024)
Бит в ячейке 1 (SLC)
Требования к ECC 1/512 (Bits/CW)
Кристаллов в чипе 1
Количество чипов (CE) 1
Общий размер данных 0x8000000 128 МиБ
Общий размер со spare 0x8800000 136 МиБ
NAND ECC коррекция Выключено
NAND таблица плохих блоков Выключено (Не используется) (Блок #0)
NAND Дамп, 256 байт
00000000 FF | yyyyyyyyyyyyyyy
00000010 FF | yyyyyyyyyyyyyyy
00000020 FF | yyyyyyyyyyyyyyy
00000030 FF | yyyyyyyyyyyyyyy
00000040 FF | yyyyyyyyyyyyyyy
00000050 FF | yyyyyyyyyyyyyyy
00000060 FF | yyyyyyyyyyyyyyy
00000070 FF | yyyyyyyyyyyyyyy
00000080 FF | yyyyyyyyyyyyyyy
00000090 FF | yyyyyyyyyyyyyyy
000000A0 FF | yyyyyyyyyyyyyyy
000000B0 FF | yyyyyyyyyyyyyyy
000000C0 FF | yyyyyyyyyyyyyyy
000000D0 FF | yyyyyyyyyyyyyyy
000000E0 FF | yyyyyyyyyyyyyyy
000000F0 FF | yyyyyyyyyyyyyyy

```

Показывать ошибки для целых блоков.

При снятом флажке в лог будет выводиться информация о количестве ошибок, (верификации или ошибок при корректировке [ECC](#)), в случае их наличия на каждой странице.

Показывать исправленные ECC ошибки.

При установленном флажке в лог будут выводиться список страниц или блоков с количеством исправленных ошибок. Если флажок снят, то только общее количество ошибок на флеш в конце операции.

Не останавливать верификацию при ошибках

Установленный флажок позволяет провести верификацию до конца при наличии ошибок и посчитать их количество.

Не использовать сигнал RB. Задержка при чтении.

Этот флажок позволяет не контролировать готовность NAND по сигналу RB и выставить задержку вручную. Эта опция не нужна для повседневных работ с нормальными флеш в колодках. Она может пригодиться для подключения на проводках, когда расположение этого сигнала неизвестно.

3.5.2 OneNAND.

OneNAND

Показать дамп первой страницы при чтении ID

Показать полную информацию о микросхеме при чтении ID

Показывать ошибки для целых блоков

Не останавливать верификацию при ошибках

Показывать дамп первой страницы при чтении ID.

При установленном флажке, при успешном чтении ID флеш, будет отображено содержимое первой страницы. Пример выделен на картинке ниже. (По умолчанию не установлен)

```

OneNAND ID 00EC0035 (1xCE) OTP
OneNAND Инфо Samsung KFG1G16U2M, 128 МиБ, страница 2048+64
OneNAND дамп, 256 байт
00000000 00 68 80 40 00 B8 08 40 FF FD 01 3C FF FF 21 34 | .h.@. .@ÿÿ.<ÿÿ!4
00000010 24 40 01 01 00 B8 88 40 00 48 80 40 40 10 08 3C | $@. . .@.H.@@. .<
00000020 00 60 88 40 00 BF 08 3C 00 00 09 24 10 60 09 AD | `.@.¿.<...$.`.-
00000030 00 BF 08 3C 00 00 09 24 14 60 09 AD 00 00 00 00 | ¿.<...$.`.....
00000040 17 00 11 04 00 00 00 00 63 00 F0 0F 00 00 00 00 | .....c.ð.....
00000050 A8 01 F0 0F 00 00 00 00 60 83 13 3C 00 00 73 26 | ".ð.....`..<...s&
00000060 02 00 04 24 00 0C 05 24 A8 00 F0 0F 00 00 00 00 | ...$.$.`".ð.....
00000070 00 06 04 24 21 28 60 02 00 00 06 24 00 08 07 24 | ...$!(`.....$...$
00000080 6D 00 F0 0F 00 00 00 00 00 82 08 3C 00 00 1D 25 | m.ð.....<...%
00000090 60 83 04 3C 00 00 84 24 08 00 80 00 00 00 00 00 | `.<...$.`.....
000000A0 02 00 08 24 00 80 88 40 00 80 08 40 00 80 09 3C | ...$.@...@...<
000000B0 FF FF 29 35 24 40 09 01 04 36 09 3C 25 40 09 01 | ÿÿ)5$@...6.<%@..
000000C0 00 80 88 40 00 60 08 40 FC FF 09 3C FF FF 29 35 | ...@.`.@ÿÿ.<ÿÿ)5
000000D0 24 40 09 01 00 00 09 24 25 40 09 01 00 60 88 40 | $@.....$%@...`.@
000000E0 08 00 E0 03 00 00 00 00 00 00 00 00 00 00 00 00 | ..à.....
000000F0 22 BF 0B 3C 18 2C 6B 35 00 00 69 85 01 00 29 31 | "¿.<.,k5...i...)1

```

Показать полную информацию о микросхеме при чтении ID.

Выводить в лог более детальную информацию. Пример на картинке ниже в сравнении с предыдущей. (По умолчанию не установлен).

```

OneNAND ID 00EC0035 (1xCE) OTP
Производитель Samsung
Наименование модели KFG1G16U2M (Single)
Шина и напряжение x16 (3.30В)
Тест RAM Норма (0xAAAA5555)
Статус контроллера 0x0020 (OTP)
Размер страницы 0x800 (2048)
Размер page 0x40 (64)
Страниц в блоке 0x40 (64)
Размер данных блока 0x20000 (131072) 128 КиБ
Размер блока со page 0x21000 (135168) 132 КиБ
Количество блоков 0x400 (1024)
Бит в ячейке 1 (SLC)
Требования к ECC 1/512 (Bits/CW)
Кристаллов в чипе 1
Количество чипов (CE) 1
Общий размер данных 0x8000000 (128 МиБ)
Общий размер со page 0x8400000 (132 МиБ)

```

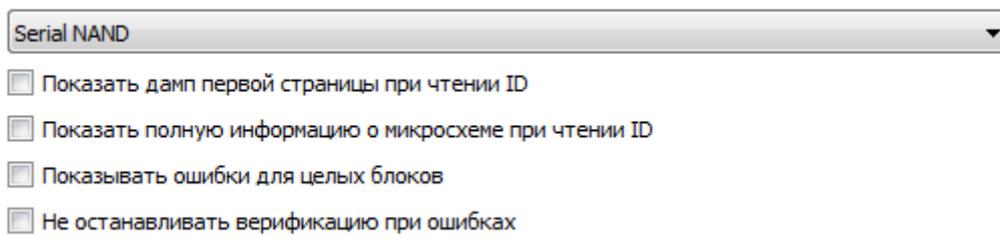
Показывать ошибки для целых блоков.

При снятом флажке в лог будет выводиться информация о количестве ошибок, (верификации или ошибок при корректировке [ЕСС](#)) в случае их наличия на каждой странице.

Не останавливать верификацию при ошибках.

Установленный флажок позволяет провести верификацию до конца при наличии ошибок и подсчитать их количество.

3.5.3 Serial NAND.



Serial NAND

- Показать дамп первой страницы при чтении ID
- Показать полную информацию о микросхеме при чтении ID
- Показывать ошибки для целых блоков
- Не останавливать верификацию при ошибках

Показывать дамп первой страницы при чтении ID.

При установленном флажке, при успешном детекте флеш, будет отображено содержимое первой страницы. (По умолчанию не установлен)

Показать полную информацию о микросхеме при чтении ID.

Выводить в лог более детальную информацию.

Показывать ошибки для целых блоков.

При снятом флажке в лог будет выводиться информация о количестве ошибок, (верификации или ошибок при корректировке [ECC](#)) в случае их наличия на каждой странице.

Не останавливать верификацию при ошибках.

Установленный флажок позволяет провести верификацию до конца при наличии ошибок и посчитать их количество.

3.5.4 NOR.

NOR

Показать дамп первого сектора при чтении ID

Показать полную информацию о микросхеме при чтении ID

Показывать дамп первого сектора при чтении ID.

При установленном флажке при успешном детекте флеш, будет отображено содержимое первого сектора. (По умолчанию не установлен)

Показать полную информацию о микросхеме при чтении ID.

Выводить в лог более детальную информацию. (По умолчанию не установлен)

3.5.5 SDMMC.

SDMMC

- Показать дамп первой страницы при чтении ID
- Показать полную информацию о микросхеме при чтении ID
- Выбирать доступ к USER разделу при чтении ID
- Не останавливать верификацию при ошибках
- Исключить RPMB из полного бэкапа

Повторы чтения при ошибке

Показывать дамп первой страницы при чтении ID.

При установленном флажке при успешном чтении ID, будет отображено содержимое первой страницы. (По умолчанию не установлен)

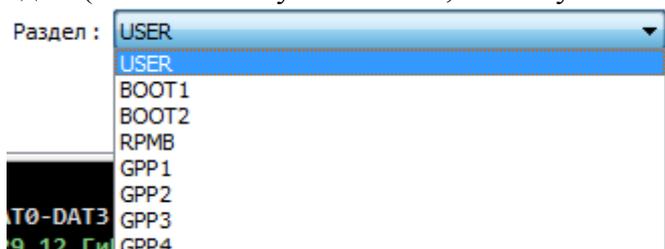
Показать полную информацию о микросхеме при чтении ID.

Вывод в лог детальной информации о флеш. (По умолчанию не установлен)

```
eMMC ID 13014E51334A3937561007C5CC5B94AD, DAT0-DAT3
eMMC Инфо Manufacturer 0x13 'Q3J97V', size 29.12 ГиБ, SN 07C5CC5B, Sep 2017
eMMC режим 4-Бит, Transfer режим, TI 2, Дрв. 0, частота 95МГц
eMMC детальная информация:
OCR 0xC0FF8080 (3V3/1V8)
CID 13014E51334A3937561007C5CC5B94AD (Manufacturer 0x13 Q3J97V, CN 0x07C5CC5B (130403419), Se
CSD D07F01328F5913FFFFFFFFFEF9240009F
Ревизия устройства [192] 0x08 (MMC v5.1)
Тип устройства [196] 0x57 (SDR HS52, HS200, HS400)
Классы команд 0x8F5 (Class 0, 2, 4, 5, 6, 7, 11)
Скорость обмена 0x32 (26МГц)
Информация о конце жизни 0x01 (Норма)
Время жизни памяти типа А 0x02 (10-20% использовано)
Время жизни памяти типа В 0x01 (0-10% использовано)
Функция Reset [162] 0x01 (Постоянно включен)
Конфигурация загрузки [179] 0x78 (Загрузка с USER, access to USER, ACK)
Шина загрузки [177] 0x00 (1-Bit SDR, Reset to 1-Bit)
Поддержка разделов [160] 0x07 (Yes, ENH, EXT)
Разделы завершены [155] 0x00 (NO)
USER раздел, размер 0x0747C00000 (29.12 ГиБ)
BOOT1 раздел, размер 0x0000400000 (4 МиБ)
BOOT2 раздел, размер 0x0000400000 (4 МиБ)
RPMB раздел, размер 0x0000400000 (4 МиБ)
```

Выбирать доступ к USER разделу при чтении ID.

При снятом флажке, при чтении ID в комбобоксе «Раздел:» будет сохраняться ранее выбранный раздел (USER/BOOT1/BOOT2/GPP*). Так же при установленном флажке, в «Показывать дамп первой страницы при чтении ID», будет выведен в лог дамп первой страницы соответствующе выбранного раздела(если галочка установлена, как по умолчанию, то всегда будет USER).



Не останавливать верификацию при ошибках.

При снятом флажке, в случае возникновения ошибок процесс верификации проходит до конца и выводит в лог количество ошибок. eMMC при нормальных условиях не должны читаться с ошибками.

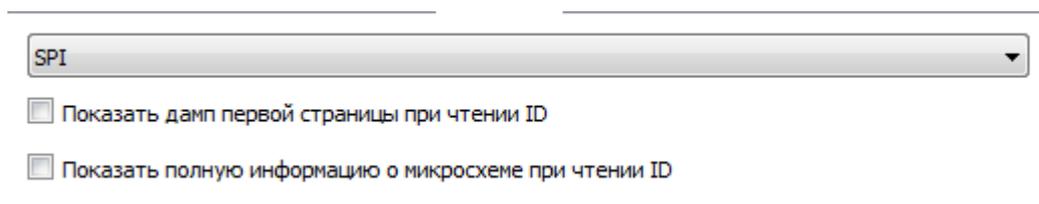
Повторы чтения при ошибке.

Количество повторов чтения при сбоях eMMC. Может потребоваться при [вычитывании неисправных флеш](#). Для этого дополнительно требуется отключить в «Доп. функции» опцию «Асинхронный» и в «Режим чтения» выставить «Игнорировать ошибки чтения» и «Режим чтения : Автовыбор»

Исключить RPMB из полного бэкапа.

Исключает из операции «Полный бэкап» раздел RPMB. Этот раздел обычно не содержит свободные для клонирования данные и его наличие в полном бэкапе может ввести в заблуждение, а попытка записи может привести к порче флеш.

3.5.6 SPI.



The screenshot shows a settings window for the SPI module. At the top, there is a dropdown menu with 'SPI' selected. Below it, there are two checkboxes, both of which are unchecked:

- Показать дампы первой страницы при чтении ID
- Показать полную информацию о микросхеме при чтении ID

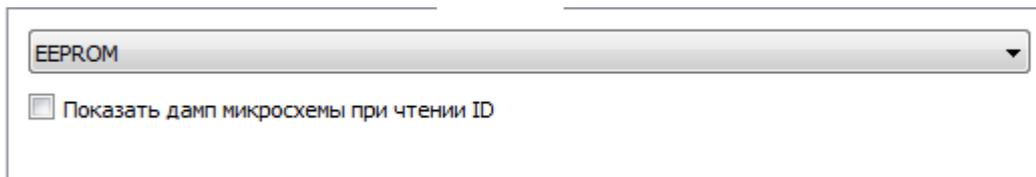
Показывать дампы первой страницы при чтении ID.

При установленном флажке при успешном чтении ID, будет отображено содержимое первой страницы. (По умолчанию не установлен)

Показать полную информацию о микросхеме при чтении ID.

Выводить в лог более детальную информацию. (По умолчанию не установлен)

3.5.7 EEPROM.



EEPROM

Показать дамп микросхемы при чтении ID

Показывать дамп первой страницы при чтении ID.

При установленном флажке при успешном чтении ID, будет отображено содержимое первой страницы. (По умолчанию не установлен)

4 Модули.

Приведен список модулей:

[1-Wire](#)

[EEPROM I2C](#)

[EEPROM SPI](#)

[EEPROM Microwire \(3-Wire\)](#)

[SPI Flash](#)

[UART](#)

[SD/eMMC](#)

[NOR](#)

[NAND](#)

[Serial NAND](#)

[OneNAND](#)

[BDM](#)

[JTAG](#)

[LOGGER](#)

[Mount File](#)

[Mount Chip](#)

4.1 LOGGER.

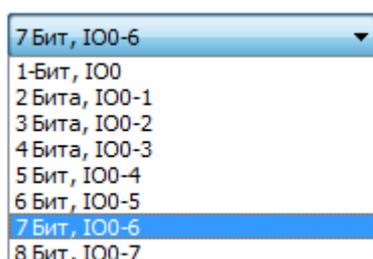
Сокет.

Работает с сокетом [2045](#).

Особенности.

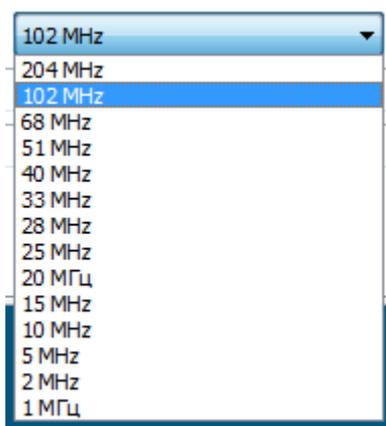
- Сокет имеет светодиоды индикации подачи питания и записи. Так же может иметь переключатель напряжения логических уровней Vref/VCC.
- Logger производит запись состояния подключенных линий (до 8 линий) в ОЗУ, или файл (объем ограничен лишь свободным местом на носителе, куда указан путь сохранения) для анализа в программе ([PulseView от Sigrok](#)). Можно подключиться к шине обмена i2c и изучить протокол обмена.

Количество линий.



Выбор количества линий для записи. Количество линий влияет на скорость записи и объем итогового файла.

Частота семплирования.



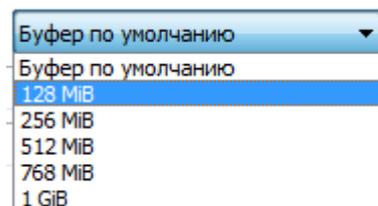
Частота выставляется из расчета не менее $n \cdot F \cdot 5$, где

n – количество выставленных каналов.

F – частота блока, или соответственно длительность наименьших импульсов в исследуемом сигнале.

Например для I2C (линии SDA+SCL) с частотой блока 200кГц потребуется выставить не менее 2МГц.

Величина буфера.



Выбор лимита буфера. После заполнения отведенного объема запись останавливается автоматически. При записи на диск опция не учитывается и не ограничивает файл.

Стабильнее всего (при работе с высокими частотами) захват производить в ОЗУ компьютера. Это позволит избежать возможных задержек операционной системы и жесткого диска.

Триггер.

Триггер (на любую активность)

Устанавливает триггер (старт записи) на активность по любому из записываемых каналов.

4.2 NAND.

UFPI NAND

Опции : CE #1 Доступ по умолчанию (мин) Проверить бэды

Чтение в : асси (Main Board) LC43BLD43BLB43T EAX65388005(1.0)\NAND_HYNIX_H27U1G8F2CTR-BC_2021-01-07_11-42.gzip Инфо о бэдах

Запись из : E:\dump\tv\LG 42LB561V (LC43B_LD43B_LB_43T EAX65388006(1.0))_2021-01-06_17-44.bin Доп. функции

Режим : Вся микросхема Плохие блоки : ключено (Плохие блоки игнорируются)

Модель : Автоопределение ECC : Выключено

```

NAND сокет x8/x16, 4xCE, 3.30В
NAND ID 98F18015721608 (98F1801572160800), CE1 (*)
Производитель Toshiba
Название модели TC58NVG053HTA00
Шина и напряжение x8 3.30В
Время доступа 25нс
Адресные циклы 2/2 (Col/Row)
Размер страницы 0x800 (2048)
Размер spage 0x80 (128)
Страниц в блоке 0x40 (64)
Размер данных блока 0x20000 (131072) 128 КиБ
Размер блока со spage 0x22000 (139264) 136 КиБ
Количество блоков 0x400 (1024)
Бит в ячейке 1 (SLC)
Требования к ECC 1/512 (Bits/CW)
Кристаллов в чипе 1
  
```

0% Стоп

NAND USB 4.99В Пит. 3V3 Вых.А: Выкл Вых.Б: Выкл

Сокеты.

Работает с сокетами различных комбинаций. Точнее описано в [разделе](#).

Особенности.

- Модуль NAND позволяет работать с флеш x8/x16 до 4xCE с напряжением от 1 до 5 вольт. UFPI позволяет работать с [BB](#), [ECC](#), ONFI, JESD, Partition table и прочее.
- Возможность работы скриптами.
- [Возможность добавления ручных конфигураций флеш.](#)

По элементам на главном окне:

4.2.1 Опции. Выбор CE (чипа).

UFPI NAND

Опции : CE #1

Чтение в : CE #2 538i

Запись из : CE #3 В_4

CE #4

Режим : Вся микросхема

Выбор чипа, если во флеш их присутствует несколько. *В общих случаях ее трогать не нужно.*

ПО автоматически определяет сколько чипов при чтении ID

```

NAND сокет x8/x16, 4xCE, 3.30В
NAND ID ECD3552558 (ECD3552558ECD355), CE1 (*)
NAND ID ECD3552558 (ECD3552558ECD355), CE2
Производитель Samsung
Название модели K9HAG08U1M
Шина и напряжение x8 3.30В
Время доступа 30нс
Адресные циклы 2/3 (Col/Row)
Размер страницы 0x800 (2048)
Размер spare 0x40 (64)
Страниц в блоке 0x80 (128)
Размер данных блока 0x40000 (262144) 256 КиБ
Размер блока со spare 0x42000 (270336) 264 КиБ
Количество блоков 0x1000 (4096)
Бит в ячейке 2 (MLC)
Требования к ECC 4/512 (Bits/CW)
Кристаллов в чипе 1
Количество чипов (CE) 2
Общий размер данных 0x80000000 2 ГиБ
Общий размер со spare 0x84000000 2.06 ГиБ

```

Звездочка (*) напротив CE1 означает выбранный чип.

Из лога видно, что определилась двухчиповая флеш. Каждый чип по 1Гб. Общий размер данных как указано 2Гб. Так и будет читаться в один файл.

Нужно отметить, что если флеш определена автоматом, есть в базе программатора и в логе отображается суммарным размером (т.е. в логе так и написано «Количество чипов (CE) 2»), то читаться будет одним файлом. Если чип неизвестен и конфигурация добавляется вручную, то каждый CE нужно будет вычитывать по отдельности выбрав необходимый CE в меню. Т.е. в конфигурации нельзя выставить количество CE!

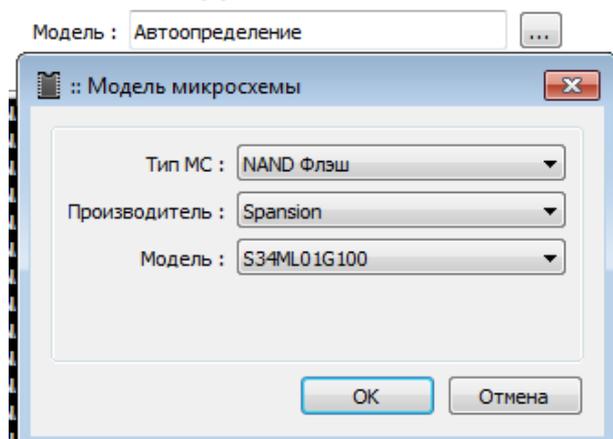
4.2.2 Цикл доступа.

Позволяет выставить длительность сигнала RE, если требуется увеличить его при использовании нестандартных подключений, проводами. По умолчанию он используется согласно внутренней базе, составленной по документации флеш.

4.2.3 «Чтение в» и «Запись из».

Стандартные практически для каждого модуля. [Двойной щелчек выводит меню](#) автоимен, операций с [ECC](#), истории файлов и другое.

4.2.4 Модель.

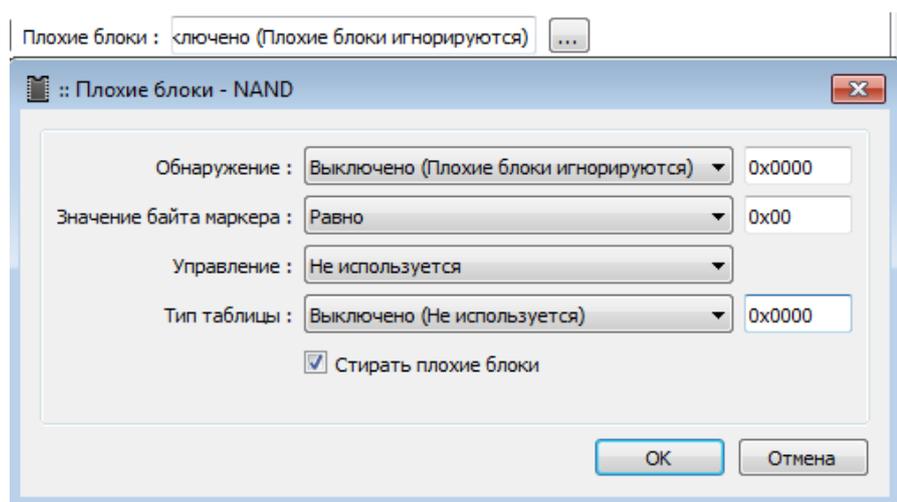


В UFPI не требуется вручную выбирать модель NAND, поскольку по ID флеш определяется автоматически или применяются параметры из базы. Но имеется возможность выбора и вручную.

4.2.5 Режим (доступа).

Описан в соответствующем общем [разделе](#). Стоит учитывать особенности [адресации для работы с NAND флеш](#).

4.2.6 Плохие блоки.



Меню, в котором задаются параметры определения и алгоритм обхода [ВВ](#). Эти опции можно загружать файлом [UDEV](#), либо выставить вручную. После изменения настроек следует перечитать ID микросхемы для применения.

Обнаружение.

Параметр по которому обнаруживается ББ.

Можно либо выключить обнаружение, либо определять их по маркеру, который можно указать произвольно (*Обычно первый, или шестой байт*), либо по записям в [таблице плохих блоков](#) (на данный момент поддерживаются некоторые Samsung и LG), либо внутренняя BBM LUT, которая встречается например у Winbond SNAND.

Значение байта маркера.

Значение маркера. *Обычно принято такой маркер считать 0x00, однако это может быть любой не равный 0xFF.*

Управление.

Выбор способа обработки ББ.

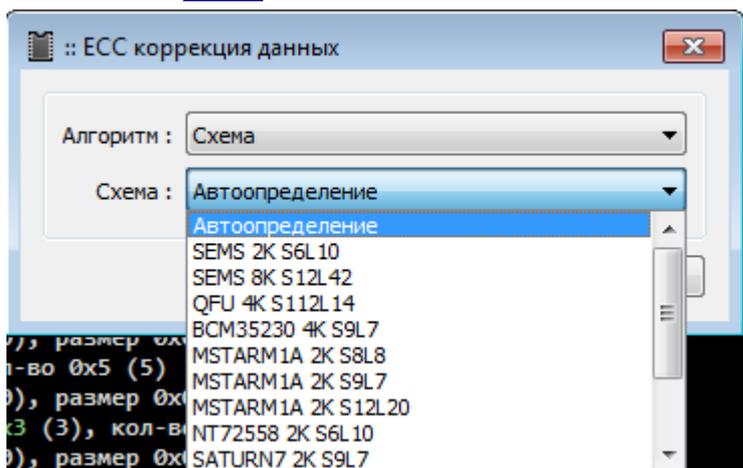
Либо не используется, либо пропуск с продолжением записи в следующий(тут не важен тип таблицы), либо, наиболее правильный способ, использовать резервную зону. *Использовать резервную зону возможно в том случае, если правильно определилась [таблица плохих блоков](#) при анализе дампа. Для обычного пропуска или записи «как есть» должен быть выставлен в «Не используется».*

Тип таблицы.

Выбор типа таблицы плохих блоков.

Для обычного пропуска или записи «как есть» должна быть выключена. Тип таблицы указывается при «Использовании резервной зоны». Если работаем с дампами Samsung или LG, то выставляем «Автоопределение» или соответствующий пункт.

4.2.7 ECC.



Выбор используемого ECC.

Выбор схемы (применяемая в дампе) или включение внутреннего алгоритма коррекции самой флеш (так называемый HWЕСС, это коррекция, код которой генерируется и обрабатывается самой флеш).

Различных схем очень много, поэтому сделано автоопределение. Выставленная ECC коррекция требуется для чтения, записи при формировании новой ВВТ, верификации, анализе и подготовке дампов.

4.2.8 Кнопки управления списком плохих блоков.

Проверить бэды.

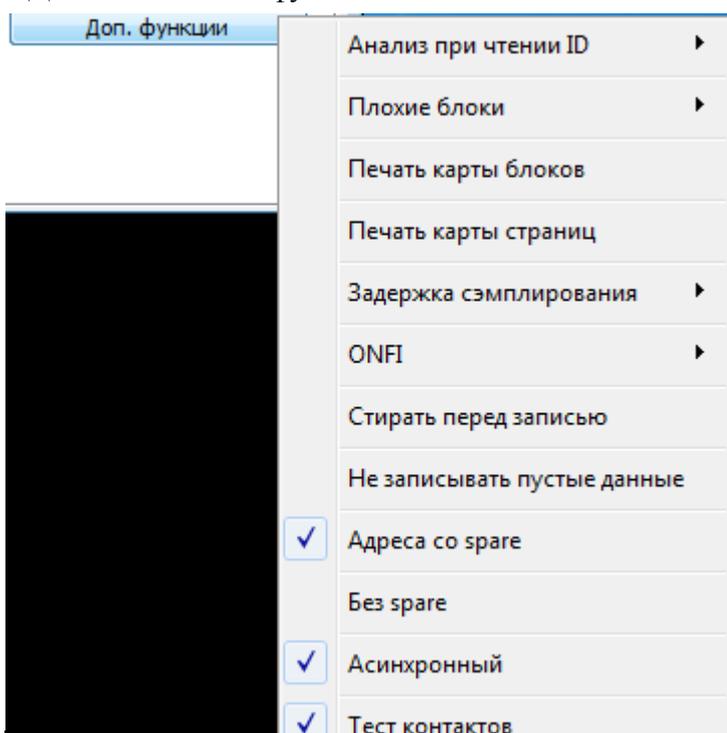
Сканирование флеш на наличие ББ в зависимости от выбранного параметра «Обнаружение».

Инфо о бэдах.

Вывод информации в лог о ББ в текущей сессии.

4.2.9 Доп. функции. (некоторые пункты появляются после чтения ID)

Дополнительные функции:



Анализ при чтении ID.

Выбор объекта анализа. Можно выбрать анализ по дампу (файлу выбранному для записи) и флеш (содержимое во флеш). *Проведя анализ, программа проверит геометрию дампа, ECC, наличие BB, VBT, таблицы разделов и сконфигурирует программатор для записи соответствующими параметрами.*
[Подробнее.](#)

Плохие блоки.

Для работы со [списком BB](#). Можно сохранить (в файл .ubad), восстановить (из файла .ubad), либо пометить плохие блоки, добавляя нужные блоки, через запятую .

Печать карты блоков.

Вывод в лог списка блоков с их рабочей адресацией. *В скобках указан адрес с учетом spare.*

Пример:

```
Блок 0x0000 (0) адрес 0x00000000 (0x00000000)
Блок 0x0001 (1) адрес 0x00100000 (0x0010DA00)
Блок 0x0002 (2) адрес 0x00200000 (0x0021B400)
Блок 0x0003 (3) адрес 0x00300000 (0x00328E00)
Блок 0x0004 (4) адрес 0x00400000 (0x00436800)
Блок 0x0005 (5) адрес 0x00500000 (0x00544200)
Блок 0x0006 (6) адрес 0x00600000 (0x00651C00)
Блок 0x0007 (7) адрес 0x00700000 (0x0075F600)
Блок 0x0008 (8) адрес 0x00800000 (0x0086D000)
Блок 0x0009 (9) адрес 0x00900000 (0x0097AA00)
Блок 0x000A (10) адрес 0x00A00000 (0x00A88400)
Блок 0x000B (11) адрес 0x00B00000 (0x00B95E00)
Блок 0x000C (12) адрес 0x00C00000 (0x00CA3800) ...
```

Печать карты страниц.

Вывод в лог списка страниц с их рабочей адресацией. *В скобках указан адрес с учетом spare.*

Пример:

```
Страница 0x0000 (0), блок 0x0000 (0) адрес 0x00000000 (0x00000000)
Страница 0x0001 (1), блок 0x0000 (0) адрес 0x00002000 (0x000021B4)
Страница 0x0002 (2), блок 0x0000 (0) адрес 0x00004000 (0x00004368)
Страница 0x0003 (3), блок 0x0000 (0) адрес 0x00006000 (0x0000651C)
Страница 0x0004 (4), блок 0x0000 (0) адрес 0x00008000 (0x000086D0)
Страница 0x0005 (5), блок 0x0000 (0) адрес 0x0000A000 (0x0000A884)
Страница 0x0006 (6), блок 0x0000 (0) адрес 0x0000C000 (0x0000CA38)
Страница 0x0007 (7), блок 0x0000 (0) адрес 0x0000E000 (0x0000EBEC)
Страница 0x0008 (8), блок 0x0000 (0) адрес 0x00010000 (0x00010DA0)
Страница 0x0009 (9), блок 0x0000 (0) адрес 0x00012000 (0x00012F54) ...
```

Задержка сэмпирования.

Установка задержек может применяться при подключении внутрисхемно, проводках или самодельном адаптере в целях компенсировать длинные линии.

ONFI.

Включено/Выключено (По умолчанию включено). [Некоторые NAND имеют специальный регион памяти](#) ONFI, в котором хранится ее конфигурация. UFPI умеет анализировать эти параметры и даже, если флеш нет в базе программатора, применять, или дополнять параметры из этих заводских конфигураций.

Проверка CRC (По умолчанию включено). Некоторые флеш содержат ONFI, заполненный не по стандарту, или имеющий ошибки в нем, или даже некорректный CRC по различным причинам. В таких случаях на усмотрение пользователя можно отключить проверку CRC или вообще отключить обращение к ONFI.

Стирать перед записью.

При установленном флажке, автоматически запускается команда стирания перед записью.

Не записывать пустые данные.

При установке флажка, во время записи пропускаются блоки, заполненные 0xFF. В таком случае можно существенно сэкономить время. Однако стоит учитывать, что при работе с NAND имеющих HWECC, запись может быть неверной.

Адреса со Spare (По умолчанию выключено).

Вывод в лог адресов с учетом Spare. В таком случае в логе адреса с учетом Spare будут подсвечены голубым шрифтом. И флаг задачи будет иметь SADDR.

Без Spare. (По умолчанию выключен)

Флажок устанавливается для работы с дампом, в котором отсутствует Spare (размер Spare=0 байт). Такие дампы имеют специфическое назначение, например для работы через jtag. Для работы с программаторами такие дампы не годятся.

Асинхронный (По умолчанию включен).

Режим работы программатора, позволяющий при нормальных условиях добиться приличного увеличения скорости работы чтения/записи за счет управления алгоритмами самим процессором программатора.

Тест контактов (По умолчанию включен).

При чтении ID флеш, предварительно проверяется корректность установки и отсутствие замыкания линий между собой или на VCC/GND. При определении замыкания в лог выводится сообщение с указанием конкретных линий:

NAND ID Ошибка при тесте контактов!

Ноги на VCC: CLE WE

Составление файлов конфигураций.

Описано в [приложении](#).

4.2.10 Приложение. Плохие блоки (BB).

ББ в логе.

Работа начинается с чтения ID и вывода списка ББ NAND.

```

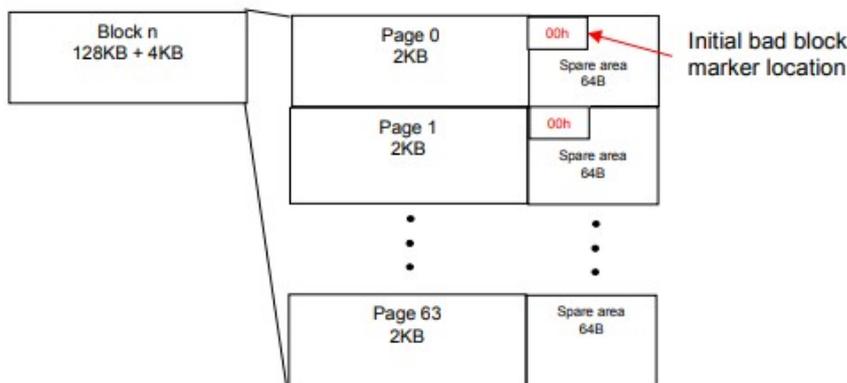
NAND сокет x8/x16, 4xCE, 3.30В
NAND ID ECD584725042 (ECD584725042ECD5), CE1 (*)
Производитель Samsung
Название модели K9GAG08U0E
Шина и напряжение x8 3.30В
Время доступа 30нс
Адресные циклы 2/3 (Col/Row)
Размер страницы 0x2000 (8192)
Размер spare 0x1B4 (436)
Страниц в блоке 0x80 (128)
Размер данных блока 0x100000 (1048576) 1 МиБ
Размер блока со spare 0x10DA00 (1104384) 1.05 МиБ
Количество блоков 0x81C (2076)
Бит в ячейке 2 (MLC)
Требования к ECC 24/1024 (Bits/CW)
Кристаллов в чипе 1
Количество чипов (CE) 1
Общий размер данных 0x81C00000 2.03 ГиБ
Общий размер со spare 0x88A7D800 2.13 ГиБ
NAND ECC коррекция MSTAR_P8K_SP436_CW8_S12L42 (нет ECC ошибок на странице #0)
NAND таблица плохих блоков Samsung RFS (Блок #0)
Обнаружение плохих блоков ВКЛ (Spare байт #0 == 0x00) //способ обнаружения «Первый байт маркер в spare»
равно 0x00
Управление плохими блоками Использовать резервную зону
Запрос плохих блоков...Норма
Количество плохих блоков: 9 //далее перечислены по этому критерию блоки
Блок 0x0001 (1) адрес 0x00100000 (0x0010DA00) Плохой блок
Блок 0x0003 (3) адрес 0x00300000 (0x00328E00) Плохой блок
Блок 0x0005 (5) адрес 0x00500000 (0x00544200) Плохой блок
Блок 0x0007 (7) адрес 0x00700000 (0x0075F600) Плохой блок
Блок 0x0009 (9) адрес 0x00900000 (0x0097AA00) Плохой блок
Блок 0x000B (11) адрес 0x00B00000 (0x00B95E00) Плохой блок
Блок 0x025E (606) адрес 0x25E00000 (0x27E40C00) Плохой блок
Блок 0x07E4 (2020) адрес 0x7E400000 (0x84F82800) Плохой блок
Блок 0x0819 (2073) адрес 0x81900000 (0x88754A00) Плохой блок

```

В логе при чтении ID описан список только тех ББ, которые определились согласно выставленным [опциям и маркерам](#). Никаких универсальных способов верно определять маркер нет, хотя стандарты существуют, но производители не всегда их придерживаются. Чаще всего маркер плохого блока расположен в первом байте в spare, первой страницы каждого блока и равен 0x00.

Стоит знать, что блок 0 обычно гарантирован производителем флеш, как надежный и не может иметь маркера ББ. Если по каким то причинам блок 0 отмечен, как «Плохой блок» в логе, значит маркер выбран не верно.

Figure 2-1: Macronix NAND flash Memory Array Structure



Если количество обнаруженных ББ в логе достигает 256 штук, значит маркер выбран неверно.

Если количество обнаруженных ББ в логе превышает 20-30 штук, то такая флеш скорее всего не подходит для надежной работы. Поскольку резервная область для замещения ББ ограничена, хотя все зависит от флеш. Например в K9GAG08U0E у Samsung D5500 составляет:

NAND RBA 0x07FB-0x0785 (2043-1925), 0x77 (119) блоков

Физические ББ.

При производстве NAND на заводе блокируются для записи блоки, которые не могут обеспечить необходимой надежности. Так же блокируются блоки специального назначения, такие, как в SAMSUNG K9GAG08U0E 2073 блок заблокирован на всех флеш. Обычно заблокированный блок просто заполнен 00 и установлен запрет на запись, но может встречаться другое заполнение.

Логические ББ.

В процессе работы в устройстве, ошибки в NAND практически всегда присутствуют и накапливаются. Для их исправления применяется [код коррекции](#). Когда количество ошибок в странице подходит к пределу возможностей коррекции ECC, содержимое блока переносится в зависимости от алгоритма на новое место. Об этом заносится запись в [таблицу ББ](#), а в блок с ошибками вносится маркер о его состоянии. Такие блоки можно стирать ([если установлена галка в опциях](#)), записывать, но стоит внимательно проверить их состояние перед использованием.

Часто на вид новые микросхемы (например K9GAG08U0E) с Aliexpress, имеющие всего 1-3 ББ, после внимательной проверки имеют десятки изношенных блоков, которые стоит так же метить как «[Плохие блоки](#)». Эти флеш не новые, просто заботливо стертые продавцом вместе с маркерами ББ. Вот пример верификации такой флеш через неделю, после записи:

NAND сокет x8/x16, 4xCE, 3.30V

NAND ID ECD584725042 (ECD584725042ECD5), CE1 (*)

Производитель Samsung

Название модели K9GAG08U0E

Шина и напряжение x8 3.30V

Время доступа 30нс

```

Адресные циклы 2/3 (Col/Row)
Размер страницы 0x2000 (8192)
Размер spare 0x1B4 (436)
Страниц в блоке 0x80 (128)
Размер данных блока 0x100000 (1048576) 1 МиБ
Размер блока со spare 0x10DA00 (1104384) 1.05 МиБ
Количество блоков 0x81C (2076)
Бит в ячейке 2 (MLC)
Требования к ECC 24/1024 (Bits/CW)
Кристаллов в чипе 1
Количество чипов (CE) 1
Общий размер данных 0x81C00000 2.03 ГиБ
Общий размер со spare 0x88A7D800 2.13 ГиБ
NAND ECC коррекция MSTAR_P8K_SP436_CW8_S12L42 (нет ECC ошибок на странице #0)
NAND таблица плохих блоков Samsung RFS (Блок #0)
Обнаружение плохих блоков ВКЛ (Spare байт #0 == 0x00)
Управление плохими блоками Использовать резервную зону
Запрос плохих блоков...Норма
Количество плохих блоков: 2 //всего два ББ
Блок 0x0714 (1812) адрес 0x71400000 (0x77470800) Плохой блок
Блок 0x0819 (2073) адрес 0x81900000 (0x88754A00) Плохой блок
NAND Дамп, 256 байт
00000000 46 53 52 5F 53 54 4C 00 01 01 02 01 FF FF FF FF | FSR_STL....ÿÿÿÿ
00000010 FE FF FF FF 00 08 00 12 00 00 01 00 01 00 02 00 | þÿÿÿ.....
00000020 01 00 10 00 00 00 04 00 01 00 00 00 01 00 00 00 | .....
00000030 02 00 D4 00 04 00 01 01 C9 00 0A 00 07 00 C9 00 | ..ô.....É.....É.
00000040 00 48 06 00 80 64 00 00 01 00 00 02 09 00 00 00 | .H...d.....
00000050 02 00 03 00 04 00 05 00 FF FF FF FF FF FF FF FF | .....ÿÿÿÿÿÿÿÿ
00000060 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000070 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000080 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000090 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000A0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000B0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000C0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000D0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000E0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000F0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
NAND Dump 'E:\temp\damp\Новая папка\SAMSUNG\D55xx\dados flash tv samsung un32d5500 k9gag08u0e - ic1302\
CLEAR\dados flash tv samsung un32d5500 k9gag08u0e - ic1302_fix_nobads.bin' Analysis, size 0x88A7D800...
NAND Страница 8192+436 (0x21B4), страниц в блоке 128, блок 0x10DA00 (1104384), 0x081C (2076) блоков, IC
геометрия
NAND ECC коррекция MSTAR_P8K_SP436_CW8_S12L42
NAND таблица плохих блоков Samsung RFS, Блок 0x07FD (2045)
NAND информация о разделах Samsung RFS, Блок 0x07FE (2046)
NAND LPCH блок 0x07FE (2046) по адресу 0x86AE4C00
NAND LPCH TPCB 0x07FC (2044), годы 1-1
NAND UPCH блок 0x07FD (2045) по адресу 0x869D7200, 1 запись(-и) (Активен)
NAND UPCH TPCB 0x07FC (2044), годы 1-2, инфо 0x0000
NAND RBA 0x07FB-0x0785 (2043-1925), 0x77 (119) блоков
NAND раздел #1, 'part-00-ID1B', блок 0x0 (0), кол-во 0xD6 (214)
NAND раздел #1, адрес 0x00000000 (0x00000000), размер 0x0D600000 (0x0E163C00)
NAND раздел #2, 'part-01-ID1C', блок 0xD6 (214), кол-во 0x100 (256)

```

```

NAND раздел #2, адрес 0x0D600000 (0x0E163C00), размер 0x10000000 (0x10DA0000)
NAND раздел #3, 'part-02-ID1D', блок 0x1D6 (470), кол-во 0xD6 (214)
NAND раздел #3, адрес 0x1D600000 (0x1EF03C00), размер 0x0D600000 (0x0E163C00)
NAND раздел #4, 'part-03-ID1E', блок 0x2AC (684), кол-во 0x410 (1040)
NAND раздел #4, адрес 0x2AC00000 (0x2D067800), размер 0x41000000 (0x4475A000)
NAND раздел #5, 'part-04-ID1F', блок 0x6BC (1724), кол-во 0xC8 (200)
NAND раздел #5, адрес 0x6BC00000 (0x717C1800), размер 0x0C800000 (0x0D2A5000)
NAND раздел #6, 'UPCH', блок 0x7FD (2045), кол-во 0x1 (1)
NAND раздел #6, адрес 0x7FD00000 (0x869D7200), размер 0x00100000 (0x0010DA00)
NAND раздел #7, 'LPCH', блок 0x7FE (2046), кол-во 0x1 (1)
NAND раздел #7, адрес 0x7FE00000 (0x86AE4C00), размер 0x00100000 (0x0010DA00)
Загружаем список плохих блоков микросхемы...2 плохих блока(-ов)
Записываем новый список плохих блоков...
NAND плохой блок 0x0714 (1812) заменен на 0x07FB (2043)
NAND плохой блок 0x0819 (2073) игнорирован
Кодируем ECC в измененных страницах...MSTAR_P8K_SP436_CW8_S12L42
Управление плохими блоками Использовать резервную зону
*****2021.05.16 22:11:34:1134*****
Источник данных Файл 'E:\temp\damp\Новая папка\SAMSUNG\D55xx\dados flash tv samsung un32d5500 k9gag08u0e -
ic1302\CLEAR\dados flash tv samsung un32d5500 k9gag08u0e - ic1302_fix_nobads.bin'...Норма
Пользовательские флаги задачи Асинхронный, ECC
Верификация NAND с адреса 0x00000000, размер 0x88A7D800...
Блок 0x0002 (2) адрес 0x00200000 (0x0021B400) 2 битовых ошибок в 1 страницах!
Блок 0x0003 (3) адрес 0x00300000 (0x00328E00) 2 битовых ошибок в 1 страницах!
Блок 0x0004 (4) адрес 0x00400000 (0x00436800) 3 битовых ошибок в 1 страницах!
Блок 0x0005 (5) адрес 0x00500000 (0x00544200) 5 битовых ошибок в 4 страницах!
Блок 0x0006 (6) адрес 0x00600000 (0x00651C00) ECC, 5 бит(-а) исправлено
Блок 0x0007 (7) адрес 0x00700000 (0x0075F600) 1 битовых ошибок в 1 страницах!
Страница 0x6C02 (27650), блок 0x00D8 (216) адрес 0x0D804000 (0x0E383368) Некорректируемая ECC
ошибка!//ненадежный блок
Блок 0x00D8 (216) адрес 0x0D800000 (0x0E37F000) 4 битовых ошибок в 2 страницах!
Блок 0x00D9 (217) адрес 0x0D900000 (0x0E48CA00) 4 битовых ошибок в 1 страницах!
Блок 0x00DA (218) адрес 0x0DA00000 (0x0E59A400) 2 битовых ошибок в 1 страницах!
Блок 0x00DB (219) адрес 0x0DB00000 (0x0E6A7E00) ECC, 1 бит(-а) исправлено
Блок 0x00DB (219) адрес 0x0DB00000 (0x0E6A7E00) 8 битовых ошибок в 5 страницах!
Блок 0x00DC (220) адрес 0x0DC00000 (0x0E7B5800) 1 битовых ошибок в 1 страницах!
Блок 0x00DD (221) адрес 0x0DD00000 (0x0E8C3200) 3 битовых ошибок в 1 страницах!
Блок 0x00DE (222) адрес 0x0DE00000 (0x0E9D0C00) ECC, 3 бит(-а) исправлено
Страница 0x6F90 (28560), блок 0x00DF (223) адрес 0x0DF20000 (0x0EB00140) Некорректируемая ECC
ошибка!//ненадежный блок
Блок 0x00DF (223) адрес 0x0DF00000 (0x0EAE6000) 1 битовых ошибок в 1 страницах!
Блок 0x00E0 (224) адрес 0x0E000000 (0x0EBEC000) 1 битовых ошибок в 1 страницах!
Блок 0x01D6 (470) адрес 0x1D600000 (0x1EF03C00) 1 битовых ошибок в 1 страницах!
Блок 0x01D8 (472) адрес 0x1D800000 (0x1F11F000) 16 битовых ошибок в 8 страницах!
Блок 0x01D9 (473) адрес 0x1D900000 (0x1F22CA00) 10 битовых ошибок в 7 страницах!
Страница 0xED0C (60684), блок 0x01DA (474) адрес 0x1DA18000 (0x1F353870) Некорректируемая ECC
ошибка!//ненадежный блок
Блок 0x01DA (474) адрес 0x1DA00000 (0x1F33A400) 16 битовых ошибок в 11 страницах!
Страница 0xED92 (60818), блок 0x01DB (475) адрес 0x1DB24000 (0x1F46DCA8) Некорректируемая ECC
ошибка!//ненадежный блок
Страница 0xEDAC (60844), блок 0x01DB (475) адрес 0x1DB58000 (0x1F4A48F0) Некорректируемая ECC
ошибка!//ненадежный блок
Блок 0x01DB (475) адрес 0x1DB00000 (0x1F447E00) ECC, 1 бит(-а) исправлено
...

```

Блок 0x0214 (532) адрес 0x21400000 (0x23050800) ECC, 4 бит(-а) исправлено

Блок 0x0215 (533) адрес 0x21500000 (0x2315E200) ECC, 1 бит(-а) исправлено

Блок 0x0215 (533) адрес 0x21500000 (0x2315E200) 3 битовых ошибок в 1 страницах!

Блок 0x0216 (534) адрес 0x21600000 (0x2326BC00) ECC, 2 бит(-а) исправлено

Блок 0x0216 (534) адрес 0x21600000 (0x2326BC00) 2 битовых ошибок в 1 страницах!

Блок 0x0217 (535) адрес 0x21700000 (0x23379600) ECC, 4 бит(-а) исправлено

Блок 0x0218 (536) адрес 0x21800000 (0x23487000) ECC, 9 бит(-а) исправлено

Блок 0x02AE (686) адрес 0x2AE00000 (0x2D282C00) 3 битовых ошибок в 2 страницах!

Блок 0x02AF (687) адрес 0x2AF00000 (0x2D390600) 2 битовых ошибок в 1 страницах!

Блок 0x02B0 (688) адрес 0x2B000000 (0x2D49E000) 3 битовых ошибок в 1 страницах!

Страница 0x15884 (88196), блок 0x02B1 (689) адрес 0x2B108000 (0x2D5B40D0) Некорректируемая ECC ошибка!//ненадежный блок

Страница 0x1588E (88206), блок 0x02B1 (689) адрес 0x2B11C000 (0x2D5C91D8) Некорректируемая ECC ошибка!//ненадежный блок

Блок 0x02B1 (689) адрес 0x2B100000 (0x2D5ABA00) 11 битовых ошибок в 7 страницах!

Блок 0x02B2 (690) адрес 0x2B200000 (0x2D6B9400) 4 битовых ошибок в 2 страницах!

Блок 0x02B4 (692) адрес 0x2B400000 (0x2D8D4800) 3 битовых ошибок в 1 страницах!

Блок 0x02B5 (693) адрес 0x2B500000 (0x2D9E2200) 2 битовых ошибок в 1 страницах!

Блок 0x02B6 (694) адрес 0x2B600000 (0x2DAEFC00) 2 битовых ошибок в 1 страницах!

Страница 0x15B88 (88968), блок 0x02B7 (695) адрес 0x2B710000 (0x2DC0E3A0) Некорректируемая ECC ошибка!//ненадежный блок

Страница 0x15B94 (88980), блок 0x02B7 (695) адрес 0x2B728000 (0x2DC27810) Некорректируемая ECC ошибка!//ненадежный блок

Страница 0x15BB0 (89008), блок 0x02B7 (695) адрес 0x2B760000 (0x2DC627C0) Некорректируемая ECC ошибка!//ненадежный блок

Страница 0x15BF4 (89076), блок 0x02B7 (695) адрес 0x2B7E8000 (0x2DCF1B90) Некорректируемая ECC ошибка!//ненадежный блок

Блок 0x02B7 (695) адрес 0x2B700000 (0x2DBFD600) 37 битовых ошибок в 30 страницах!

Блок 0x02B8 (696) адрес 0x2B800000 (0x2DD0B000) 4 битовых ошибок в 1 страницах!

Блок 0x02B9 (697) адрес 0x2B900000 (0x2DE18A00) 5 битовых ошибок в 1 страницах!

Страница 0x15D0C (89356), блок 0x02BA (698) адрес 0x2BA18000 (0x2DF3F870) Некорректируемая ECC ошибка!//ненадежный блок

Страница 0x15D4C (89420), блок 0x02BA (698) адрес 0x2BA98000 (0x2DFC6570) Некорректируемая ECC ошибка!//ненадежный блок

Блок 0x02BA (698) адрес 0x2BA00000 (0x2DF26400) ECC, 1 бит(-а) исправлено

Блок 0x02BA (698) адрес 0x2BA00000 (0x2DF26400) 43 битовых ошибок в 29 страницах!

Блок 0x02BB (699) адрес 0x2BB00000 (0x2E033E00) 2 битовых ошибок в 1 страницах!

Страница 0x15E14 (89620), блок 0x02BC (700) адрес 0x2BC28000 (0x2E16BA10) Некорректируемая ECC ошибка!//ненадежный блок

Страница 0x15E26 (89638), блок 0x02BC (700) адрес 0x2BC4C000 (0x2E1918B8) Некорректируемая ECC ошибка!//ненадежный блок

Страница 0x15E2E (89646), блок 0x02BC (700) адрес 0x2BC5C000 (0x2E1A2658) Некорректируемая ECC ошибка!//ненадежный блок

Страница 0x15E70 (89712), блок 0x02BC (700) адрес 0x2BCE0000 (0x2E22D6C0) Некорректируемая ECC ошибка!//ненадежный блок

Блок 0x02BC (700) адрес 0x2BC00000 (0x2E141800) 42 битовых ошибок в 32 страницах!

Страница 0x15EE2 (89826), блок 0x02BD (701) адрес 0x2BDC4000 (0x2E31D8E8) Некорректируемая ECC ошибка!//ненадежный блок

Блок 0x02BD (701) адрес 0x2BD00000 (0x2E24F200) ECC, 1 бит(-а) исправлено

Блок 0x02BD (701) адрес 0x2BD00000 (0x2E24F200) 38 битовых ошибок в 29 страницах!

Блок 0x02BE (702) адрес 0x2BE00000 (0x2E35CC00) 1 битовых ошибок в 1 страницах!

Блок 0x02BF (703) адрес 0x2BF00000 (0x2E46A600) 50 битовых ошибок в 33 страницах!

Блок 0x02C0 (704) адрес 0x2C000000 (0x2E578000) 4 битовых ошибок в 3 страницах!

Страница 0x16086 (90246), блок 0x02C1 (705) адрес 0x2C10C000 (0x2E692438) Некорректируемая ECC ошибка!//ненадежный блок

Блок 0x02C1 (705) адрес 0x2C100000 (0x2E685A00) ECC, 2 бит(-а) исправлено

Блок 0x02C1 (705) адрес 0x2C100000 (0x2E685A00) 26 битовых ошибок в 17 страницах!
 ...
 Блок 0x038F (911) адрес 0x38F00000 (0x3BF7C600) ECC, 19 бит(-а) исправлено
 Блок 0x0390 (912) адрес 0x39000000 (0x3C08A000) ECC, 24 бит(-а) исправлено
 Блок 0x0391 (913) адрес 0x39100000 (0x3C197A00) 4 битовых ошибок в 2 страницах!
 Блок 0x0392 (914) адрес 0x39200000 (0x3C2A5400) ECC, 12 бит(-а) исправлено
 Блок 0x0393 (915) адрес 0x39300000 (0x3C3B2E00) ECC, 17 бит(-а) исправлено
 Страница 0x35F06 (220934), блок 0x06BE (1726) адрес 0x6BE0C000 (0x719E9638) Некорректируемая ECC ошибка!//ненадежный блок
 Блок 0x06BE (1726) адрес 0x6BE00000 (0x719DCC00) 6 битовых ошибок в 2 страницах!
 Блок 0x06BF (1727) адрес 0x6BF00000 (0x71AE6000) 1 битовых ошибок в 1 страницах!
 Страница 0x36074 (221300), блок 0x06C0 (1728) адрес 0x6C0E8000 (0x71CEC590) Некорректируемая ECC ошибка!//ненадежный блок
 Блок 0x06C0 (1728) адрес 0x6C000000 (0x71BF8000) 13 битовых ошибок в 12 страницах!
 Страница 0x36084 (221316), блок 0x06C1 (1729) адрес 0x6C108000 (0x71D0E0D0) Некорректируемая ECC ошибка!//ненадежный блок
 Блок 0x06C1 (1729) адрес 0x6C100000 (0x71D05A00) 5 битовых ошибок в 2 страницах!
 ...
 Блок 0x06EE (1774) адрес 0x6EE00000 (0x74C6AC00) ECC, 2 бит(-а) исправлено
 Блок 0x06EE (1774) адрес 0x6EE00000 (0x74C6AC00) 2 битовых ошибок в 1 страницах!
 Блок 0x07FB (2043) адрес 0x7FB00000 (0x867BBE00) Плохой блок
 Блок 0x07FD (2045) адрес 0x7FD00000 (0x869D7200) ECC, 1 бит(-а) исправлено
 Блок 0x07FD (2045) адрес 0x7FD00000 (0x869D7200) 19 битовых ошибок в 2 страницах!
 Блок 0x07FE (2046) адрес 0x7FE00000 (0x86AE4C00) 1 битовых ошибок в 1 страницах!
 Блок 0x0819 (2073) адрес 0x81900000 (0x88754A00) Плохой блок
 ECC проверено в 36793 страницах, 228357 пустых страниц пропущено, 174 пустых кодовых слов, 2607 кодовых слов с пустым ECC
 Исправлено 4349 ECC битовых ошибок, 269 в данных ECC
 Обнаружено 27 некорректируемых ECC ошибок в 27 страницах //основной показатель
 Общее количество битовых ошибок во время верификации: 567
 Завершено за 3 мин. 46 сек. Скорость загрузки кода 9.64 МиБ/сек.

В итоге метить нужно как минимум блоки 216, 223, 474, 475, 689, 695, 698, 700, 701, 705, 829, 875, 881, 889, 893, 1726, 1728, 1729, 1812 и получается не все так красиво, хотя верификация после записи показала минимальное количество корректируемых ошибок и все выглядело хорошо. Можно заметить кучность плохих блоков, они соответствуют наиболее активным в работе разделам во флеш. Видно, что за недолгий промежуток времени получили явно нерабочую флеш, с которой устройство не сможет правильно работать. *Это особенно касается людей, которые покупают или продают готовые, записанные NAND флеш для ремонта.* Только записав и проверив лично, можно оценить флеш и спрогнозировать надежность.

После добавления ББ, перед следующими операциями, следует применить новый список ББ путем повторного чтения ID.

После добавления и повторной проверки через неделю, можно увидеть в логе некорректируемые ошибки в блоках, которые уже были помечены. Это означает, что блок, на который переназначен ББ тоже изношен и требует внесения в список. Для этого исходя из списка нужно пометить именно блок, на который он ссылается:

NAND плохой блок 0x0714 (1812) заменен на 0x07FB (2043)

4.2.11 Приложение. Таблица плохих блоков (BBT).

Назначение.

Стирать и работать с неисправными блоками нельзя. Определять состояние, сканируя первую страницу каждого блока при обращении занимает много времени, поэтому используется таблица плохих блоков. Таблица ББ, это место в дампе, где записан список плохих блоков, места их нового размещения и прочие служебные данные. Никаких универсальных способов работать с таблицами ББ не существует. Хотя определенные спецификации существуют, но производителей не обязывают их соблюдать. Поэтому обычно у производителей оборудования имеется свое представление таблицы и UFPI умеет некоторые из них определять и работать с ними.

Расположение BBT.

Располагаться таблица плохих блоков может в начале, в конце флеш или по другим фиксированным смещениям. В настройках определения «Плохих блоков» можно указать точный адрес расположения выбранной таблицы.

Изменение BBT.

Изменение BBT как и любых других данных в NAND требует дальнейшего пересчета ECC этой страницы. Применение обхода, пропуска, переноса плохих блоков без занесения в BBT не сможет правильно работать в устройстве. Исходя из назначения и содержания таблицы Samsung D5500 видно, что каждый плохой блок имеет переназначение. В логе прогнатора:

```
flash tv samsung un32d5500 k9gag08u0e - ic1302_fix_nobads.bin' Analysis, size 0x88A7D800...
NAND Страница 8192+436 (0x21B4), страниц в блоке 128, блок 0x10DA00 (1104384), 0x081C (2076) блоков, IC
геометрия
NAND ECC коррекция MSTAR_P8K_SP436_CW8_S12L42
NAND таблица плохих блоков Samsung RFS, Блок 0x07FD (2045)
NAND информация о разделах Samsung RFS, Блок 0x07FE (2046)
NAND LPCH блок 0x07FE (2046) по адресу 0x86AE4C00
NAND LPCH TPCB 0x07FC (2044), годы 1-1
NAND UPCH блок 0x07FD (2045) по адресу 0x869D7200, 1 запись(-и) (Активен)
NAND UPCH TPCB 0x07FC (2044), годы 1-2, инфо 0x0000
NAND RBA 0x07FB-0x0785 (2043-1925), 0x77 (119) блоков
NAND раздел #1, 'part-00-ID1B', блок 0x0 (0), кол-во 0xD6 (214)
NAND раздел #1, адрес 0x00000000 (0x00000000), размер 0x0D600000 (0x0E163C00)
NAND раздел #2, 'part-01-ID1C', блок 0xD6 (214), кол-во 0x100 (256)
NAND раздел #2, адрес 0x0D600000 (0x0E163C00), размер 0x10000000 (0x10DA0000)
NAND раздел #3, 'part-02-ID1D', блок 0x1D6 (470), кол-во 0xD6 (214)
NAND раздел #3, адрес 0x1D600000 (0x1EF03C00), размер 0x0D600000 (0x0E163C00)
NAND раздел #4, 'part-03-ID1E', блок 0x2AC (684), кол-во 0x410 (1040)
NAND раздел #4, адрес 0x2AC00000 (0x2D067800), размер 0x41000000 (0x4475A000)
NAND раздел #5, 'part-04-ID1F', блок 0x6BC (1724), кол-во 0xC8 (200)
NAND раздел #5, адрес 0x6BC00000 (0x717C1800), размер 0x0C800000 (0x0D2A5000)
NAND раздел #6, 'UPCH', блок 0x7FD (2045), кол-во 0x1 (1)
NAND раздел #6, адрес 0x7FD00000 (0x869D7200), размер 0x00100000 (0x0010DA00)
NAND раздел #7, 'LPCH', блок 0x7FE (2046), кол-во 0x1 (1)
NAND раздел #7, адрес 0x7FE00000 (0x86AE4C00), размер 0x00100000 (0x0010DA00)
Загружаем список плохих блоков микросхемы...20 плохих блока(-ов)
Записываем новый список плохих блоков...
NAND плохой блок 0x00D8 (216) заменен на 0x07FB (2043)
NAND плохой блок 0x00DF (223) заменен на 0x07FA (2042)
```

NAND плохой блок 0x01DA (474) заменен на 0x07F9 (2041)
NAND плохой блок 0x01DB (475) заменен на 0x07F8 (2040)
NAND плохой блок 0x02B1 (689) заменен на 0x07F7 (2039)
NAND плохой блок 0x02B7 (695) заменен на 0x07F6 (2038)
NAND плохой блок 0x02BA (698) заменен на 0x07F5 (2037)
NAND плохой блок 0x02BC (700) заменен на 0x07F4 (2036)
NAND плохой блок 0x02BD (701) заменен на 0x07F3 (2035)
NAND плохой блок 0x02C1 (705) заменен на 0x07F2 (2034)
NAND плохой блок 0x033D (829) заменен на 0x07F1 (2033)
NAND плохой блок 0x036B (875) заменен на 0x07F0 (2032)
NAND плохой блок 0x0371 (881) заменен на 0x07EF (2031)
NAND плохой блок 0x0379 (889) заменен на 0x07EE (2030)
NAND плохой блок 0x037D (893) заменен на 0x07ED (2029)
NAND плохой блок 0x06BE (1726) заменен на 0x07EC (2028)
NAND плохой блок 0x06C0 (1728) заменен на 0x07EB (2027)
NAND плохой блок 0x06C1 (1729) заменен на 0x07EA (2026)
NAND плохой блок 0x0714 (1812) заменен на 0x07E9 (2025)
NAND плохой блок 0x0819 (2073) игнорирован

В дампе:

869D93A0	D5 C4 EF 3C 88 5D DA 14 E0 10 CC 0B A9 AF 29 18	ХДп<Є]ъ.а.М.©I) .
869D93B0	FF FF FF FF 00 00 FF FF D8 00 FB 07 DF 00 FA 07	яяяя. .яяШ.ы.Я.ъ.
869D93C0	DA 01 F9 07 DB 01 F8 07 B1 02 F7 07 B7 02 F6 07	ъ.щ.Ы.ш.±.ч. .ц.
869D93D0	BA 02 F5 07 BC 02 F4 07 BD 02 F3 07 C1 02 F2 07	е.х.ј.ф.С.у.Б.т.
869D93E0	3D 03 F1 07 6B 03 F0 07 71 03 EF 07 79 03 EE 07	=.с.к.р.қ.п.у.о.
869D93F0	7D 03 ED 07 BE 06 EC 07 C0 06 EB 07 C1 06 EA 07	}.н.э.м.А.л.Б.к.
869D9400	14 07 E9 07 FF	. .й. яяяяяяяяяяяяяя
869D9410	FF	яяяяяяяяяяяяяяяяяя
869D9420	FF	яяяяяяяяяяяяяяяяяя

Поэтому никакие «Универсальные» обходы и пропуски реализуемые на сторонних программаторах не решат проблему. Программатор должен уметь работать с каждой таблицей лично и на основе нее применять алгоритм обхода плохих блоков. *Это стоит знать всем, кто поднимает темы про «волишебный» обход ББ на других программаторах и предлагают варианты с пропуском плохого блока и записью данных в следующий блок.*

В качестве примера обхода ББ с применением RBA (*Reserved Block Area - резервной зоны*) можно ознакомиться с [приложением](#).

4.2.12 Приложение. ECC.

Назначение.

ECC - (error-correcting code) код коррекции ошибок. Не вдаваясь в физические тонкости образования ошибок, стоит лишь понимать, что количество ошибок увеличивается с течением времени, а чем больше циклов стирания блок флеш, тем быстрее эти ошибки могут в нем накапливаться. Флеш NAND изначально предполагает применение кода коррекции. Для поддержания работы NAND применяются коды коррекции **Hamming, BCH, Reed-Solomon** с различными опциями и производные от них с применением шифрований и прочих стандартных и нестандартных приемов. Полный размер страницы NAND состоит из page и spare. Обычно в page содержатся данные, а в spare код коррекции.

Никаких изменений, патчей, пересчётов BBT без пересчета кода ECC сделать невозможно.

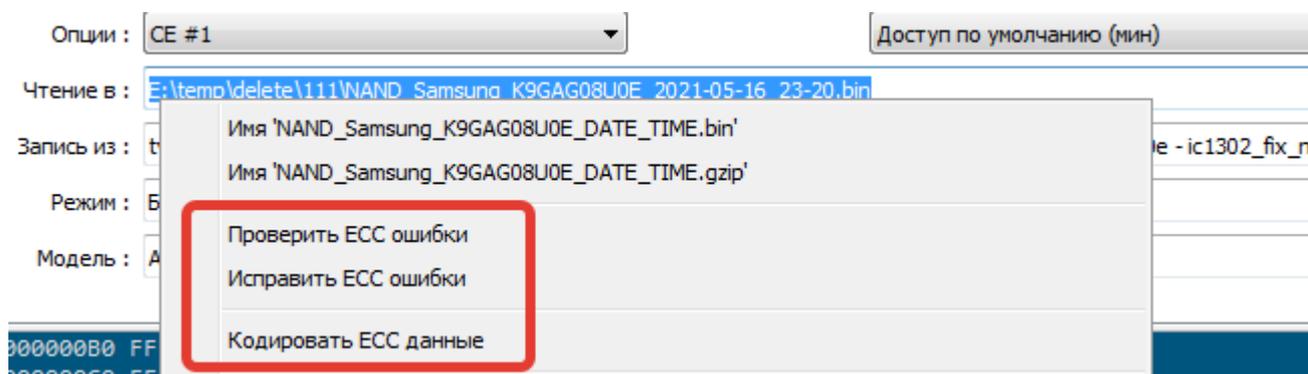
Никаких гарантий на дамп и его запись в NAND (даже если в ней нет ни одного ББ) без проверки ECC давать невозможно.

Выбор ECC.

Обычному пользователю сложно или даже невозможно определить параметры и алгоритм ECC, поэтому в UFPI упростили задачу, введя автоопределение алгоритма. Для этого достаточно при выставленной схеме «Автоопределение» вставить флеш (имеющую нужный дамп) и прочитать ID. В логе будет название алгоритма:

NAND ECC коррекция Автоопределение MSTAR_P8K_SP436_CW8_S12L42 (нет ECC ошибок на странице #0)

Либо при выбранном нужном файле дампа **двойным кликом** по полю «Чтение в» или «Запись из» выбрать в контекстном меню проверить или исправить ECC.



Таким образом можно проверить файл на корректность еще до прошивки и установки флеш в аппарат.

NAND ECC коррекция MSTAR_P8K_SP436_CW8_S12L42 (Автоопределение)

Проверка ECC в 'E:\temp\delete\111\NAND_Samsung_K9GAG08U0E_2021-05-16_23-20.bin'...

ECC проверено в 36767 страницах, 228940 пустых страниц пропущено, 7 пустых кодовых слов, 168 кодовых слов с пустым ECC

Обнаружено 1 ECC битовых ошибок, 1 в данных ECC

Не редкость файлы, которые в файлообменниках, с пометкой «проверенный» или ими делятся на форумах, имеют некорректируемые ошибки. Файлы с некорректируемыми ошибками не могут дать гарантии на корректную работу.

Обозначение.

Для облегчения и визуального восприятия для схем ECC введены названия, по которым можно ориентироваться не вникая в сложности алгоритма. Для примера приведен дамп от Samsung D5500.

При чтении ID определяется алгоритм автоматически:

NAND ECC коррекция Автоопределение MSTAR_P8K_SP436_CW8_S12L42 (нет ECC ошибок на странице #0)

MSTAR — производитель процессора «Mstar». Можно узнать по надписи на процессоре или описании в интернете.

P8K — размер page 8192 байта. Можно увидеть из параметров NAND или лога при чтении ID в программаторе.

SP436 — размер spare 436 байт. Можно увидеть из параметров NAND или лога при чтении ID в программаторе.

S12 — количество байт от начала spare до начала кода коррекции.

```

00001ff0 ff | .....
00002000 ff ff ff ff 01 00 fe ff ff ff ff ff ff 6a b9 e2 bd | .....j...
00002010 e6 c2 f4 af 7a ea 54 a4 13 2e 8f f3 56 e1 84 c0 | ....z.T....V...
00002020 e2 b2 fc 2e 59 fb 69 fc fc 82 b6 d6 16 7d 5f 39 | ....Y.i.....} 9
00002030 89 a2 35 7a f1 ce 40 01 bf fe ff ff ff ff ff ff | ..5z..@.....
00002040 ff ff bc ea 73 1f e0 77 12 c2 cc 63 cb 40 17 cc | ....s..w...c.@..
00002050 61 b8 5b 05 5b 49 7a 26 8d e3 e9 c8 5f 4e f8 af | a.[.[Iz&.... N..
00002060 c9 5f c5 a3 29 11 62 77 e4 be f7 26 00 00 31 09 | . ..) .bw...&..1.

```

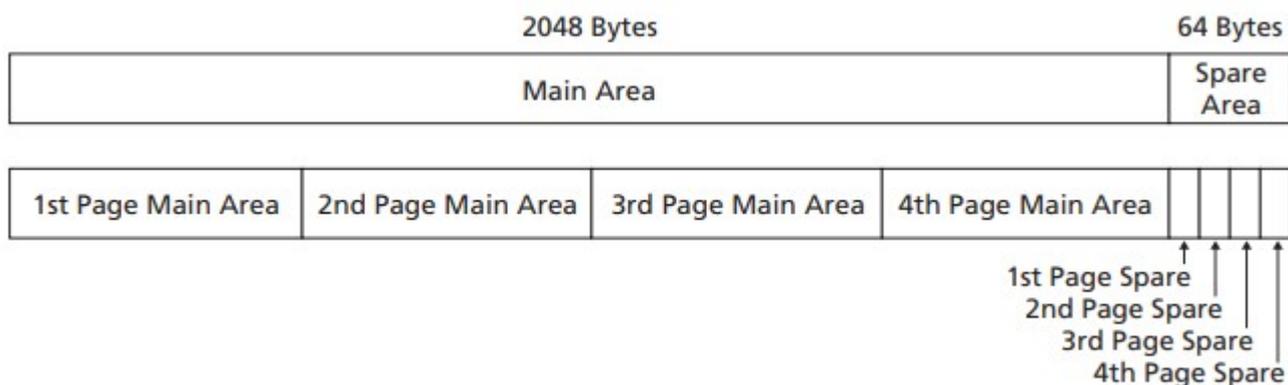
L42 — количество байт самого кода коррекции.

```

00001ff0 ff | .....
00002000 ff ff ff ff 01 00 fe ff ff ff ff ff ff 6a b9 e2 bd | .....j...
00002010 e6 c2 f4 af 7a ea 54 a4 13 2e 8f f3 56 e1 84 c0 | ....z.T....V...
00002020 e2 b2 fc 2e 59 fb 69 fc fc 82 b6 d6 16 7d 5f 39 | ....Y.i.....} 9
00002030 89 a2 35 7a f1 ce 40 01 bf fe ff ff ff ff ff ff | ..5z..@.....
00002040 ff ff bc ea 73 1f e0 77 12 c2 cc 63 cb 40 17 cc | ....s..w...c.@..
00002050 61 b8 5b 05 5b 49 7a 26 8d e3 e9 c8 5f 4e f8 af | a.[.[Iz&.... N..

```

CW8 — количество частей на которые разбита страница. Примерная иллюстрация в документации Micron выглядит так:



Только в дампе от D5500 страница 8192 байт, spare 436 и разделена страница на 8 частей. Это наглядно можно увидеть, посчитав сколько частей кода коррекции для полной страницы в spare, разделенных друг от друга, байтами 0xFF.

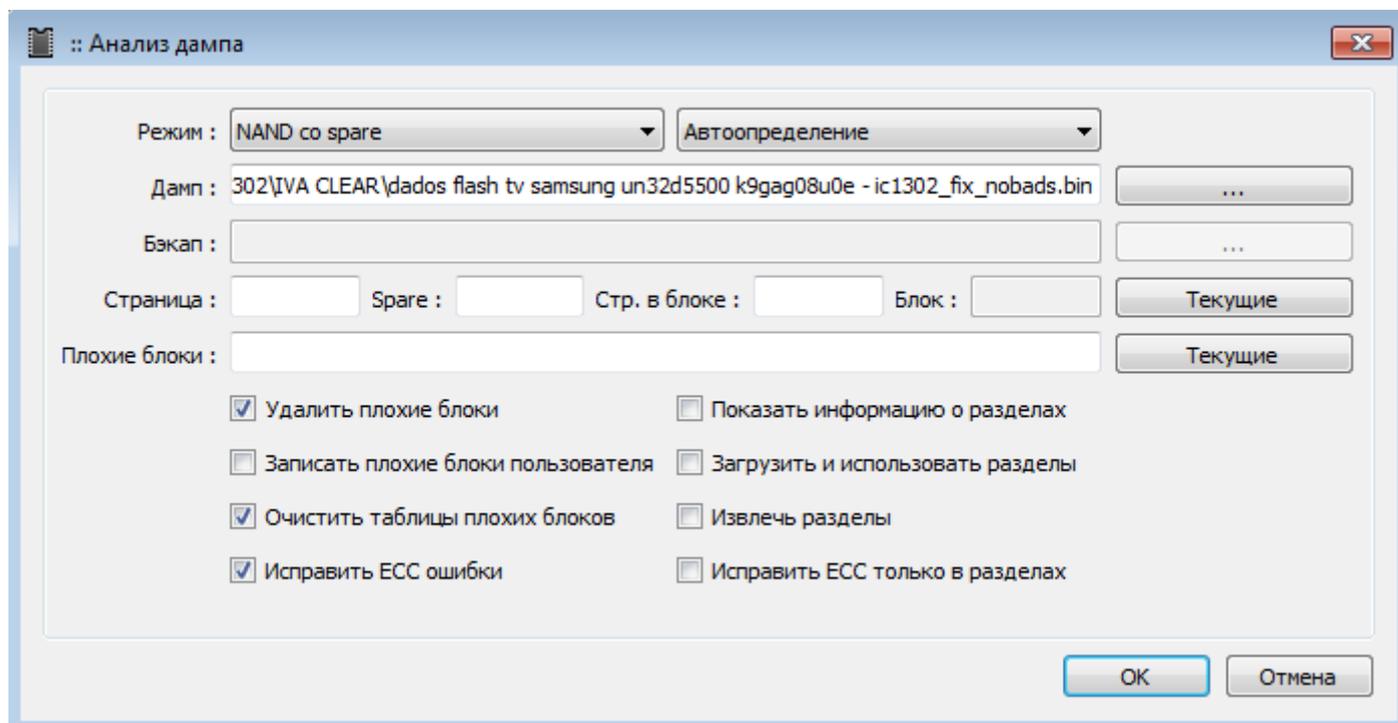
Остальные добавки к названию алгоритма для пользователя не несут полезной информации.

4.2.13 Приложение. Анализ дампа. Подготовка дампа. Получение «чистого» дампа.

Подключить программатор с микросхемой.

В настройках [ЕСС](#) выбрать нужный [алгоритм](#). Чаще если алгоритм известен, то автоопределение справляется.

В меню «Инструменты» выбрать «Анализ дампа».



В поле режим выбрать «NAND со spare» и «Автоопределение». В некоторых случаях, когда анализ автоматически не может распознать, потребуется явно указать размеры, выбрав «Геометрия пользователя» или установить нужную флеш, прочитать ID и выбрать «Геометрия микросхемы».

В поле «Дамп» выбрать исправляемый дамп.

Поля «Страница», «Spare», «Стр. в блоке» заполнить, если как сказано ранее, автоопределение не распознало геометрию NAND по дампу. Тут нужно вписать размер страницы, spare и количество страниц в блоке. В поле «Блок» автоматически посчитается размер блока. Если в программаторе установлена нужная флеш и предварительно считано ID, то можно нажать кнопку «Текущие» и поля заполнятся текущими (из этой микросхемы) размерами страницы, spare и их количеством.

Поле «Плохие блоки» предназначено для формирования в подготавливаемом дампе пользовательских [ББ](#) и используется при установленной галке «Записать плохие блоки пользователя». Может быть полезной при подготовке индивидуального дампа под конкретный экземпляр флеш для записи без функций обхода ББ. По кнопке «Текущие» автоматически заполняется [текущим списком ББ, который был получен при чтении ID](#). Для формирования чистого дампа для дальнейшей записи с управлением ББ в UFPI эти параметры не нужны.

Установить галки «Удалить плохие блоки», «Очистить таблицы плохих блоков» и «Исправить ошибки ECC».

Флажки «Показать информацию о разделах» и «Загрузить и использовать разделы» для анализа разметки разделов и перевода в режим доступа [«Разделы»](#).

При установленном флажке «Извлечь разделы», при анализе в папке с дампом будут созданы отдельные файлы разделов этого дампа.

Флажок «Исправить ECC только в разделах» бывает необходима при использовании разных схем ECC

в разных разделах одного дампа. При этом флажке проводится автоопределение ЕСС на каждом разделе индивидуально.

С демонстрацией применения можно ознакомиться в дополнительном [приложении](#).

4.2.14 Приложение. Анализ дампа при чтении ID и запись «чистого» дампа с управлением «Использовать резервную зону».

Доп. функции.

Включить «[Анализ дампа при чтении ID](#)» в «Доп. функции».

Для обычных условий и флеш выставить «Асинхронный» и «Не записывать пустые данные».

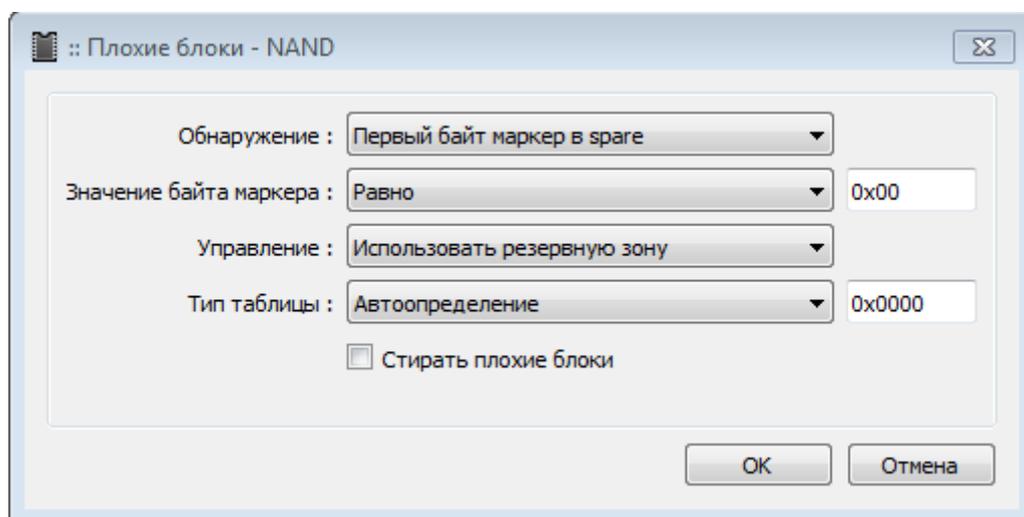
Настройки работы с плохими блоками:

Включить обнаружение ББ по маркеру. Обычно это первый байт со значением 0x00.

Для поддерживаемых таблиц ББ управление выставить «Использовать резервную зону».

Тип таблицы обычно определяется автоматически в случае с использованием анализа при чтении ID. Поэтому выбрать «Автоопределение».

Стирать плохие блоки не рекомендуется без необходимости. Если только маркер ББ не совпадает и при чтении ID по нему определилось слишком много ББ.



Настройка ECC.

Для обычных NAND нужно выбрать схему с «Автоопределение».

Файл.

Выбрать [«чистый» дамп](#). При выборе дампа содержащего ББ в логе при анализе будет получена соответствующая ошибка.

Процесс конфигурирования программатора для записи.

Прочитать ID, анализ дампа сконфигурирует программатор для дальнейшей работы с флеш и дампом. Трансляция замененных блоков будет производится при чтении/записи/стирании автоматически прошивкой программатора.

Стереть флешь.

При необходимости ввести список плохих блоков через «Доп. функции» - «Плохие блоки» - «Пометить бэды». [Учитывать блоки, которые уже имеют пенал](#). Обычно кандидаты на бэд блоки проявляются спустя непродолжительное время на старых флеш. Поэтому перезаписывать бывает приходится по несколько раз, пометчая плохие блоки.

Прочитать ID повторно, анализ дампа сконфигурирует программатор для дальнейшей работы с флеш и дампом. Трансляция замененных блоков будет производится при чтении/записи/стирании

автоматически прошивкой программатора.

Записать, верифицировать.

Если в процессе верификации будут найдены некорректируемые ошибки, то следует повторить процесс конфигурации программатора с занесением дополнительных плохих блоков и произвести запись, верификацию повторно.

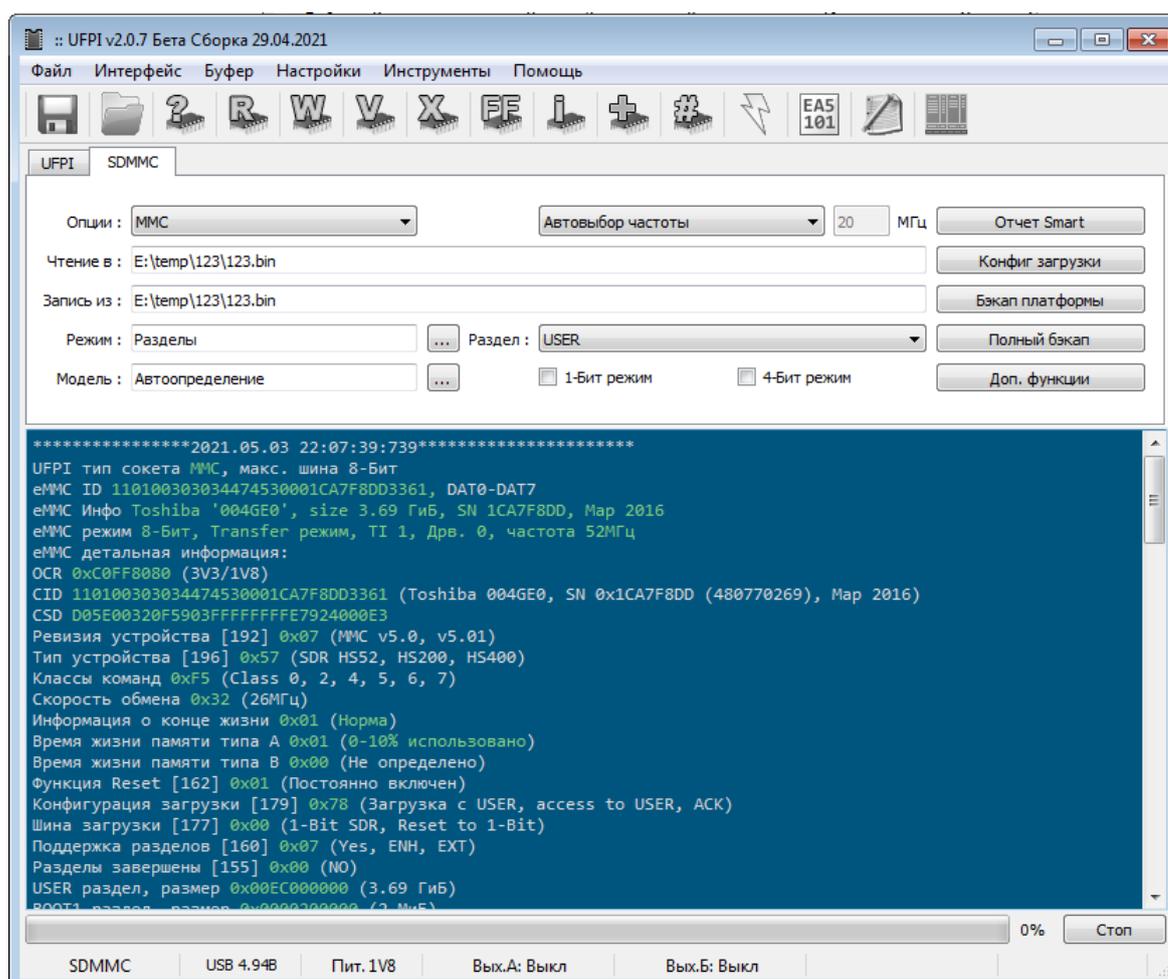
Верификацию можно проводить спустя неограниченное время. Для этого выполнить пункты, кроме стирания и записи флеш.

С демонстрацией применения можно ознакомиться в дополнительном [приложении](#).

4.2.15 Описание пунктов в логе при работе.

В разработке...

4.3 SD/eMMC.



Сокеты.

Большой выбор сокетов для eMMC перечислен в [приложении](#).

Особенности.

- Самый быстрый среди известных конкурентов. Скорость ограничена лишь свойствами самой флеш, подключением и USB портом компьютера.
- Позволяет в большинстве случаев удачно работать внутрисхемно даже на высокой скорости.
- Не требует выбора модели микросхемы. Модель и параметры определяются автоматически.
- Продвинутое функциями работы с разделами и опциями загрузки, а так же возможность использования конфигураций для бэкапов платформ и разделов и многое другое. Желательно ознакомиться с основными положениями в спецификации [JEDEC](#).
- Расширение функционала с помощью скриптов.
- Работа 1-4-8 битном режиме, а так же синхронно, блоками с контролем целостности и повтором перечитывания при ошибках, позволяет вычитывать проблемные флеш.
- eMMC не требуют [стирания](#).

4.3.1 Опции.



Автоопределение/MMC/SDC - установка типа устройства.

Автовыбор частоты/Частота пользователя. Обычно для работы с исправной микросхемой в колодке, можно оставлять «автовыбор». Понижать частоту может понадобиться при нестабильной работе, но ниже 5-10МГц опускаться нет смысла. Повышать можно в целях эксперимента даже до 100МГц, однако на таких частотах eMMC даже в колодках стабильно работать не могут. На практике повышать при HS52 выше 52МГц нет смысла, а при HS200 более 80-85МГц часто начинается нестабильная работа. После ввода в поле частоты, для применения параметра следует нажать «Enter» и затем заново прочитать ID.

4.3.2 «Чтение в» и «Запись из».

Стандартные практически для каждого модуля. [Двойной щелчок выводит меню](#) автоимен, истории и другое.

4.3.3 Режим.

Описано в другом [разделе](#) и имеются [примеры](#).

4.3.4 Модель.

Не требуется.

4.3.5 Раздел.

Выбор раздела [USER/RPMB/BOOT1/BOOT1/GPP1-4](#).

4.3.6 4-Бит режим / 1-Бит режим.

Программатор автоматически определяет в каком режиме подключена флеш. Т.е. если из 8 (8-Бит режим доступен только в [Pro](#) версии программатора) линий будут подключены в правильном порядке 4, то автоматически применится конфигурация для работы в 4-битном режиме.

4.3.7 Отчет Smart.

Вывести Smart отчет для eMMC ревизий обычно до v5.0. В нем описано состояние «здоровья» флеш.

Стоит обратить особое внимание на пункты:

«Режим ошибки» - должно быть 0xD2D2D2D2 Норма.

«количество резервных блоков оставшееся» - блоков должно оставаться >10 штук, или достаточно для продолжения работы.

«Количество некорректируемых ошибок ECC» - этих ошибок не должно быть у хорошей флеш.

Важно знать, что в eMMC ревизии выше v5.0 уже присутствует «[Health life time](#)», он при наличии выводится в лог автоматически и отображает текущий уровень износа:

Информация о конце жизни 0x01 (Норма)

Время жизни памяти типа A 0x01 (0-10% использовано)

Время жизни памяти типа B 0x01 (0-10% использовано)

4.3.8 Конфиг загрузки.

В этом окне собраны основные параметры, которые стоит выставить при замене флеш. Параметры,

которые нужно выставить, можно посмотреть при чтении ID родной флеш. По сути представлен калькулятор трех байт (162, 177, 179) из регистра [eCSD](#). В верхней части выбираем в выпадающих меню параметры, а внизу исходя из конфигурации бит, сразу составляется байт. Так же работает и наоборот, вписав байт и нажав «Применить» можно увидеть расшифровку по пунктам сверху. Нажав «ОК», выставленные значения запишутся в соответствующие регистры.

Отдельно вынесено по RST_N [162] байт. В качестве предохранителя снята галка и подписано предупреждение, что параметр «ОТР» (т.е. обычно это означает однократно записываемое без возможности изменить обратно).

Все эти байты по отдельности можно записать в меню «[Доп. функции](#)»

4.3.9 Полный бэкап.

Сохранить одним нажатием полный набор разделов USER, BOOT1/2, RPMB (если не установлена галка в настройках модуля «Исключить RPMB из полного бэкапа») и eCSD с полным логом, в одну папку.

4.3.10 Бэкап платформы.

Служит для упрощения рутинных задач по переносу регионов и разделов имеющих привязки и ключи в дампах. На примере можно ознакомиться в [приложении](#).

PBI это обычный INI файл, кодировка UTF-8.

;Перечисление производителей.

```
[groups]
group1=LG
group2=Philips
group3=Samsung
```

;Перечисление моделей.

```
[LG]
model1=LC_LD_LE42B_42G
```

```
[LC_LD_LE42B_42G]
```

```
Name=LC_LD_LE42B_42G
act1=0x3F00000:0x80000:USER:sestore
act2=0x3F80000:0x80000:USER:sedata
```

```
[Philips]
```

```
model1=QV14
model2=QV15
model3=TPM14
model4=TPM15
model5=QM16
```

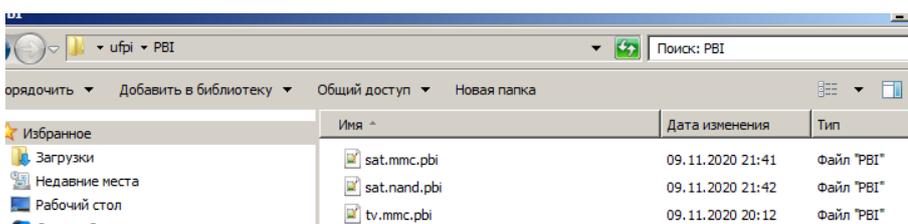
```
[QV14]
```

```
Name=QV14
```

```
...
```

Для каждого модуля будут использованы .pbi с определенными именами типа tv.mmc.pbi, sat.nand.pbi и т.д.

Может быть несколько .pbi файлов с одним расширением для одного модуля типа tv.mmc.pbi, sat.mmc.pbi.



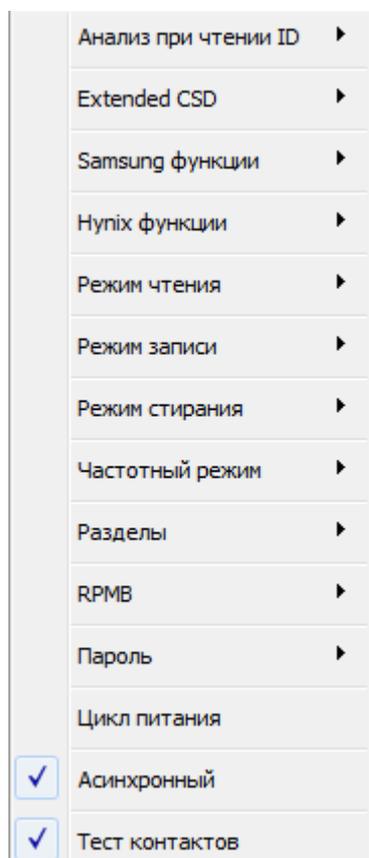
После нажатия кнопки "?" и чтения ID, становится доступна кнопка меню "Бэкап платформы".

Для eMMC используются .mmc.pbi файлы.

Для SD карт используются .sdc.pbi файлы.

Файлы PBI можно составлять самим, обмениваться, либо искать на форуме. Например вот [вариант](#) и [тут](#).

4.3.11 Доп. функции. (некоторые пункты появляются после чтения ID)



Анализ при чтении ID.

Выбор объекта для поиска таблицы разделов. Можно назначить отдельно «eMMC» или «Дамп», либо комбинацию, выбрав приоритет «Дамп, eMMC». При этом если открыт .UDEV файл с описанием разделов, то будет загружен список разделов из UDEV. Иначе программа попытается проанализировать разметку собственными алгоритмами.

Extended CSD.

Записать отдельный байт в регистр, или целиком файл [eCSD](#).

При чтении ID в логе описаны и подписаны отдельные байты eCSD. В квадратных скобках указан номер байта в регистре. Далее его значение и расшифровка. При записи байта достаточно так же указать его номер и значение. Пример:

Конфигурация загрузки [179] 0x00 (Нет загрузки, access to USER)

При каждом чтении ID в папке «ufpi\Logs\eMMC» создаются файлы, в том числе полного eCSD. Они подписаны номером флеш и дата/время создания. При записи полного файла, достаточно выбрать его, при необходимости отредактировать в hexeditor.

Важно помнить, что не все регистры [eCSD](#) доступны для записи и не все регистры доступны для восстановления. У разных производителей разные зарезервированные регистры, которые нельзя менять без последствий. Стоит внимательно ознакомиться со спецификациями JEDEC.

Основные байты можно сконфигурировать в калькуляторе [Конфиг загрузки](#).

Samsung функции.

Большой выбор спец. функций доступен преимущественно для eMMC Samsung ревизии до v5.0. Собраны в одном месте.

«Запись [CID](#)» и «Запись [CID #2](#)» — записывает пользовательское значение CID, даже от eMMC другого производителя.

«Запись [CSD](#)» — записывает пользовательское значение CSD.

«Снять защиту от записи» — снятие защиты от записи, статус которой установлен в CSD и eCSD и отображен в логе при чтении ID:

Permanent Write Protect

или

Boot Write Protection 0x0A (BOOT1_PERM BOOT2_PERM)

или

User Write Protection 0x10 (US_PERM_WP_DIS)

«Форматирование» — очищает содержимое разделов и регистров до заводского состояния.

«Чтение прошивки» и «Запись прошивки» - специальные команды чтения и записи внутреннего FW eMMC. Поля «Чтение в» и «Запись из» должны быть заполнены соответствующими файлами для чтения и записи. Файл прошивок считанные посторонними программаторами могут быть несовместимы, или требуют корректировки. Операция прошивки внутреннего FW у eMMC не гарантирует корректной дальнейшей работы флеш.

Hynix функции.

«Очистить Lifetimer (v5.1)» - Экспериментальная функция сброса показателя счетчика наработки [«LifeTimer»](#). Сделана только для того, чтобы продлить срок работы в девайсах, для которых он критичен. Работает на eMMC Hynix ревизии v5.1. Помимо сброса LifeTimer, записывает в [155] байт [eCSD](#) значение «Разделы завершены».

Режим чтения.

«Автовыбор» — Оптимальный режим для чтения исправных флеш в колодке на высокой скорости. Стоит по умолчанию.

«Одиночными блоками» — вычитывание происходит последовательно по одному блоку с контролем целостности принятого блока CMD17 (READ_SINGLE_BLOCK) . Размер блока по умолчанию указан в регистре CSD. Подходит для работы с неисправными флеш, где требуется контроль и повторное перечитывание проблемных блоков. Скорость этого режима самая низкая.

«Несколькими блоками (Открытый)» — вычитывание происходит по несколько блоков командой CMD18 (READ_MULTIPLE_BLOCK) и прерывается командой остановки.

«Несколькими блоками (Счетчик)» — вычитывание происходит по несколько блоков командой CMD18 (READ_MULTIPLE_BLOCK) и останавливается. Счетчик задается командой CMD23 (SET_BLOCK_COUNT).

«Несколькими блоками (Один запрос)» — вычитывание происходит командой CMD18 (READ_MULTIPLE_BLOCK) без остановок до конца заданного региона.

«Игнорировать ошибки чтения» — не останавливаться при ошибках чтения, читать со следующего блока. Опция может пригодиться при вычитывании неисправных флеш.

Режим записи.

«Автовыбор» — Оптимальный режим для записи исправных флеш в колодке на высокой скорости. Стоит по умолчанию.

«Одиночными блоками» — аналогично чтению.

«Несколькими блоками (Открытый)» — аналогично чтению.

«Несколькими блоками (Счетчик)» — аналогично чтению.

«Несколькими блоками (Один запрос)» — аналогично чтению.

«4-Бит режим при VCCQ=1V8» — при записи переводит режим в 4 бит, если VCCQ 1,8V.

Рекомендуется использовать эту опцию для стабильного и предсказуемого результата.

Режим стирания.

«Автовыбор» — по умолчанию. [По стиранию eMMC.](#)

«Одиночными блоками» — Стирание командами по одному блоку.

«Несколькими блоками» — Стирание командами по несколько блоков.

«Запись нулей» — запись 0x00.

Частотный режим.

«Автовыбор» — Оптимальный режим. Установлен по умолчанию.

«Устаревший 26 МГц» — режим совместимости с устаревшими MMC картами.

«Высокая скорость 26 МГц» — высокоскоростной режим (HS52), с выставленной частотой 26МГц.

«Высокая скорость 52 МГц» — высокоскоростной режим (HS52) с выставленной частотой 52МГц.

Этот режим установлен по умолчанию как наиболее стабильный и подходящий для большинства случаев.

«Высокая скорость 200МГц» — Высокоскоростной режим (HS200) до 200МГц. На практике выше 90МГц стабильной работы в распространенных колодках BGA добиться сложно.

«Задержка семплирования» и «Задержка выхода» — параметры могут пригодиться для согласования линий при работе на высокой частоте. Для использования достаточно выставить максимальное значение, если ошибки пропали, понижать до меньшего стабильного значения.

Разделы.

BOOT/ RPMB / GPP. — некоторые eMMC, в частности Samsung до ревизии 5.0 поддерживают функцию разметки разделов BOOT, RPMB, GPP. Для изменения, ввести размерность (кБ/МБ) и целочисленное значение.

RPMB.

«Аутентификация» — ввод ключа для дальнейшей работы и записи в RPMB.

«Записать ключ аутентификации» — установка ключа аутентификации производится только в том случае, если он ранее не был записан. Это согласно спецификации 32 байтный ключ в HEX виде (т.е. 0x...). *Перед использованием следует понимать необходимость действий. Операция для большинства флеш необратима.*

«Выключено» — отключение работы и опроса статуса RPMB при чтении ID. По наблюдениям пользователей, встречались неисправные eMMC Toshiba, которые после опроса статуса RPMB переставали читаться до следующей перезагрузки.

Пароль.

«Разблокировать» — требует ввод пароля (в формате ASCII) для запуска сессии работы с заблокированной флеш. Например в логе такой флеш будет:

eMMC ID 1501003030585056300304051227C0CF, DAT0 Заблокировано!

«Заблокировать» — завершение сессии с заблокированной флеш.

«Очистить» — Позволяет снять известный пароль.

«Установить» — установить пароль для блокировки флеш.

«Стереть» - стирает пароль и очищает содержимое флеш. Для этой операции не требуется ввод пароля.

«Брутфорс» — автоподбор пароля. Можно задавать длину, символы, а так же останавливать и продолжать перебор.

Цикл питания.

Полный цикл отключения питания, задержка и заново инициализация и чтение ID.

Асинхронный.

[Аналогично остальным модулям.](#) *Важно не забывать, что вычитывание и пропуск ошибок невозможен в асинхронном режиме.*

Тест контактов.

[Аналогично остальным модулям.](#)

4.3.12 Приложение. Регистры CID, CSD, eCSD, OCR.

CID (Card identification data) .

Регистр MMC, который содержит данные, по которым можно идентифицировать карту памяти (серийный номер, ID производителя, дату изготовления имя микросхемы и т.д.), длина 16 байт. Встречаются аппараты где в загрузчике прописано имя микросхемы, при замене микросхемы на не аналогичную, иногда приходится прописывать CID от такой же микросхемы или от родной, если его удаётся вычитать. Такие операции применимы только на eMMC Samsung ревизии до v5.0. Ниже приведена таблица из документации, описывающая значение этого регистра.

CSD (Card-specific data).

CID Fields Name	Field	Width	CID slice	Value
Manufacturer ID	MID	8	[127:120]	70h
Reserved		6	[119:114]	0h
Device/BGA	CBX	2	[113:112]	1h
OEM/Application ID	OID	8	[111:104]	0h
Product name	PNM	48	[103:56]	(4D3532353136h"M52516")
Product revision	PRV	8	[55:48]	51h*
Product serial number	PSN	32	[47:16]	Random by Production
Manufacturing date	MDT	8	[15:8]	month, year
CRC7 checksum	CRC	7	[7:1]	-(Note 1)
not used, always "1"	-	1	[0]	1h

Содержит различную информацию о карте памяти (от размера сектора карты памяти до потребления в режиме чтения/записи). Длинна 16 байт.

Важными областями в CSD являются 12 и 13 биты:

PERM_WRITE_PROTECT - Постоянная защита от записи / стирания.

TMP_WRITE_PROTECT - Защита от записи / стирания до перезагрузки карты.

Если выставлен флаг постоянной защиты у рабочего аппарата - он будет включаться и работать но все изменения будут обнуляться после следующей перезагрузки. Некоторые такие флеш можно вылечить форматирование, или прошивкой внутреннего FW, но надежность их уже все равно останется под сомнением.

extCSD (Extended Card-specific data).

Содержит всевозможную дополнительную информацию и режимы карты памяти. Длинна 512 байт. Достаточно длинный регистр - но основная его часть ReadOnly - никогда не меняются и прописаны еще на заводе. Основная часть регистра описана [JEDEC](#) спецификациями, но отдельные байты зарезервированы и имеют индивидуальное у каждого производителя назначение, значения которых отсутствуют в общедоступной документации.

OCR (Operation conditions register).

Содержит данные о напряжении питания карты памяти, тип питания карты памяти, статус процесса инициализации карты.

Так нормальный ответ ходовой современной eMMC - OCR 0xC0FF8080 (3V3/1V8), т.е. последний [31] бит устанавливается в 1 если процедура включения карты завершена успешно.

OCR Register Definitions OCR bit	VDD voltage window	High Voltage MultimediaCard	Dual voltage MultimediaCard and e•MMC™
[6:0]	Reserved	00 00000b	00 00000b
[7]	1.70 - 1.95V	0b	1b
[14:8]	2.0-2.6V	000 0000b	000 0000b
[23:15]	2.7-3.6V	1 1111 1111b	1 1111 1111b
[28:24]	Reserved	0 0000b	0 0000b
[30:29]	Access Mode	00b (byte mode) 10b (sector mode)	00b (byte mode) 10b (sector mode)
[31]	(Device power up status bit (busy)) ₁		
Note1 : This bit is set to LOW if the Device has not finished the power up routine.			

4.3.13 Приложение. Разделы USER, BOOT1, BOOT2, RPMB, GPP1, GPP2, GPP3, GPP4.

RPMB (Replay Protected Memory Block) .

Защищенный раздел, который поддерживает механизм проверки секретным ключом. Может использоваться производителем техники для защиты от клонирования и саботажа защищенных функций ПО. UFPI поддерживает работу с этим разделом в полном объеме, описанном в спецификации. Позволяет писать, читать данные, устанавливать секретный ключ и размер (если это предусмотрено производителем). Этот раздел всегда создается на заводе (обычно размер 4МБ) и присутствует на флеш, даже если не используется и пустой.

BOOT1, BOOT2.

Эти разделы так же всегда присутствуют на флеш, даже если не используются и пусты. Обычно размер составляет по 4МБ. Предназначены обычно для размещения загрузочной области. UFPI поддерживает работу с этими разделами в полном объеме, описанном в спецификации. Позволяет писать, читать данные и устанавливать размер (если это предусмотрено производителем).

GPP1 – GPP4.

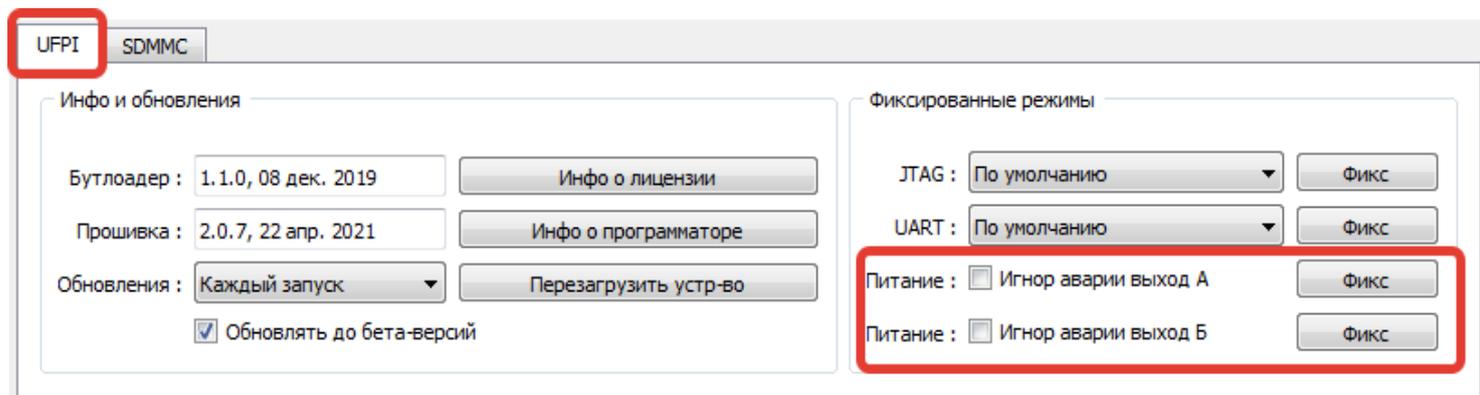
Разделы общего назначения. По умолчанию они не включены на заводе изготовителе, но пользователь может их активировать. Их количество и размер устанавливается пользователем и может быть от 1 до 4. UFPI поддерживает работу с этими разделами в полном объеме, описанном в спецификации. Позволяет писать, читать данные и устанавливать количество и размер (если это предусмотрено производителем).

USER.

Основной пользовательский раздел, в котором может располагаться загрузчик, таблица разделов, разделы и прочие данные. Его емкость это общий объем минус емкость занятая предыдущими разделами.

4.3.14 Приложение. ISP подключение.

Для работы в этом режиме, в некоторых случаях, может потребоваться отключение защиты по току.



Для подключения используются следующие шины.

VSS (GND) — земля

VCC(VDDF) — питание ядра, нанд, микроконтроллера = 2.7v-3.6v.

VCCQ(VDD) — питание интерфейса = 1.7v-1.95v или 2.7v-3.6v.

CMD — двусторонняя шина передачи и приёма команд.

CLK — тактовая.

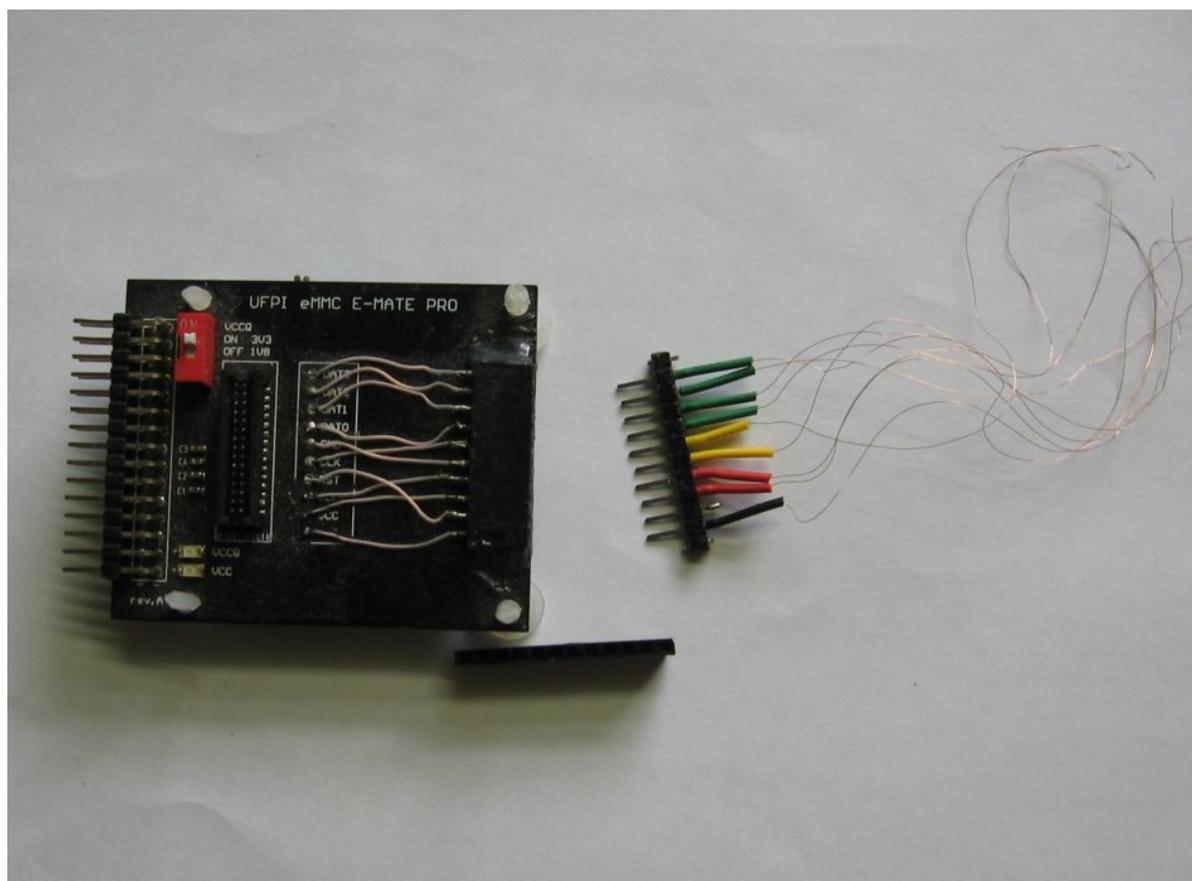
D0-D3 — шины передачи и приёма данных.

RST — линия сброса. Может быть программно установлена зависимость микросхемы от этой шины - 162 байт в extCSD. Если там прописана 1, то на этой шине обязательно должна быть 1. В большинстве случаев эта шина подтянута резистором к питанию, но встречаются аппараты, где этой шиной управляет процессор, в этом случае, для подключения в isр режиме нужно подавать штатное питание и останавливать процессор или переводить его в третье состояние, например через reset.

На некоторых майнах достаточно подать в одну точку 3.3V. Там либо объединены питание ядра и интерфейса, либо стабилизатор выдающий 1.8v для питания интерфейса питается от шины 3.3V.

На некоторых майнах приходится удалять дроссель, стоящий по питанию микросхемы, чтобы не запитывать остальную часть платы. Иногда приходится резать дорожки или удалять проходные резисторы, чтобы устранить влияние процессора. В этих случаях нужно удостовериться, что на резисторы подтяжки шин обмена данными и на резистор сброса, подаётся питание.

Провод лучше всего использовать мягкий, толщиной 0.1 мм и длиной не более 10-15 см. Нужно быть очень внимательным при пайке, так как можно повредить торцы SMD резисторов и получить не включающийся майн.



При попытке запитать от программатора майн токи составляют сотни мА, следует учесть, что [usb кабель](#), и [порт должны быть хорошего качества](#).

Если плата запитывается от внешнего источника, а не от программатора, то на адаптере следует установить переключатель питания VCCQ в то положение, которое соответствует напряжению подаваемому на майн. Объединять VCCQ программатора и майн при этом нельзя.

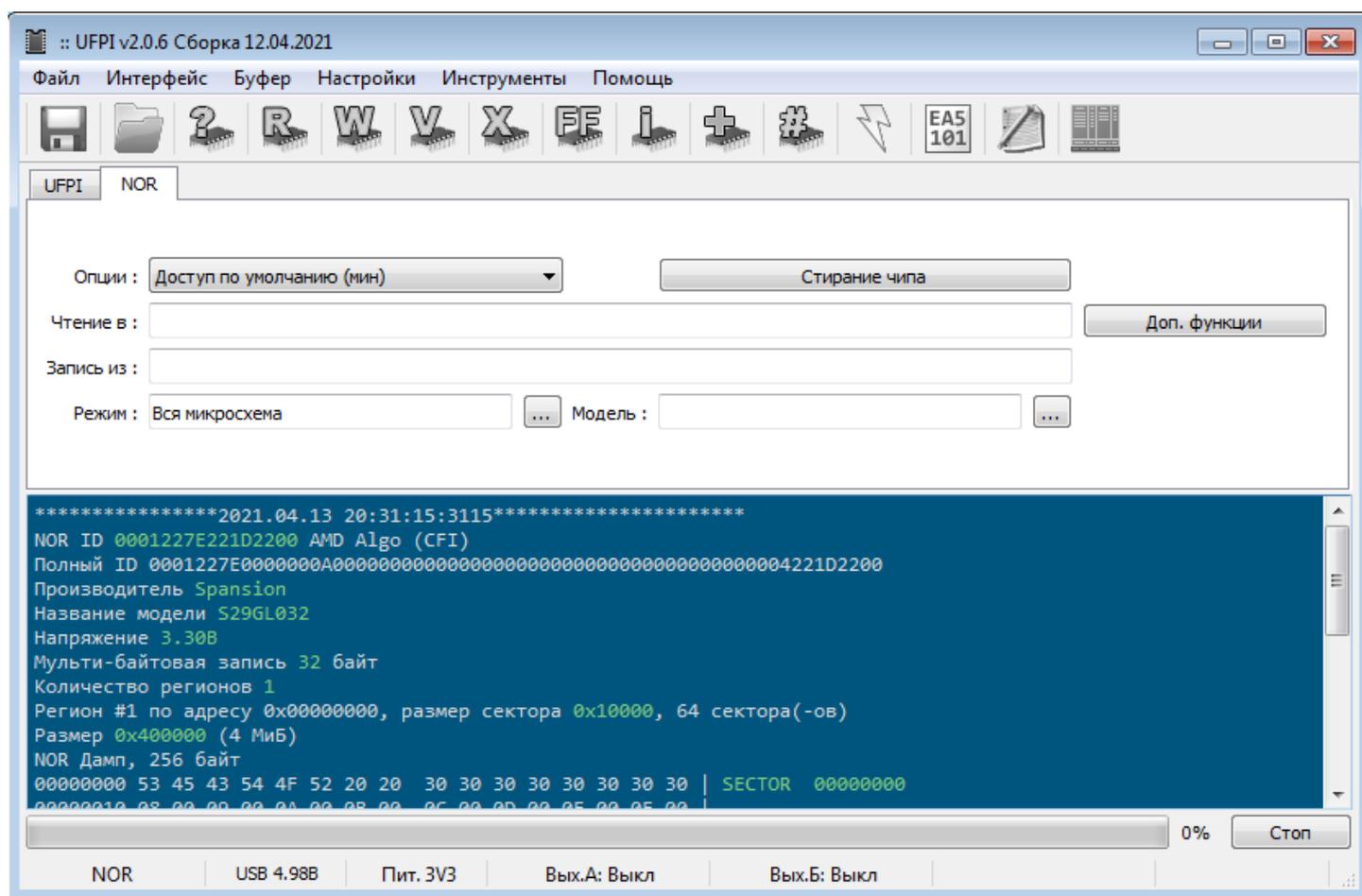
Для подобных работ удобнее отключить в настройках «Чтение ID после присоединения сокета» и управлять подачей питания через [кнопку питания](#).

При отсутствии описанного сокета или при использовании макетной платы, нужно ознакомиться со [схемой](#) и учесть, что VCC программатора для работы с eMMC подключается к VCCQ.

4.3.15 Описание пунктов в логе при работе.

В разработке...

4.4 NOR.



Сокеты.

Работает с сокетом [2046](#) или [2060](#).

Особенности.

- Определение параметров флеш происходит автоматически, при условии, что правильно подключена. Выбор флеш в этом модуле даже не активен.
- Для наиболее корректной работы, UFPI представляет адресное пространство регионами, посекторно с индивидуальными размерами.
- Поддерживается [CFI](#) и [ручная конфигурация](#).
- Возможность работы скриптами.

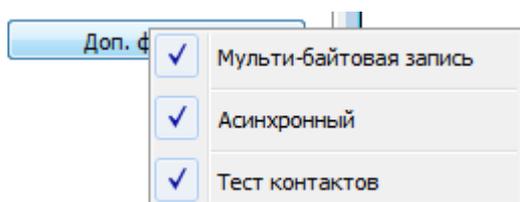
Стирание.

Производится либо кнопкой <X> либо «Стирание чипа». В первом случае стирается посекторно, необходимо при работе с частями/секторами/разделами. Во втором стирается вся микросхема одной командой.

Режим.

Аналогично как и в других микросхемах есть [возможность работать с отдельными регионами флеш](#).

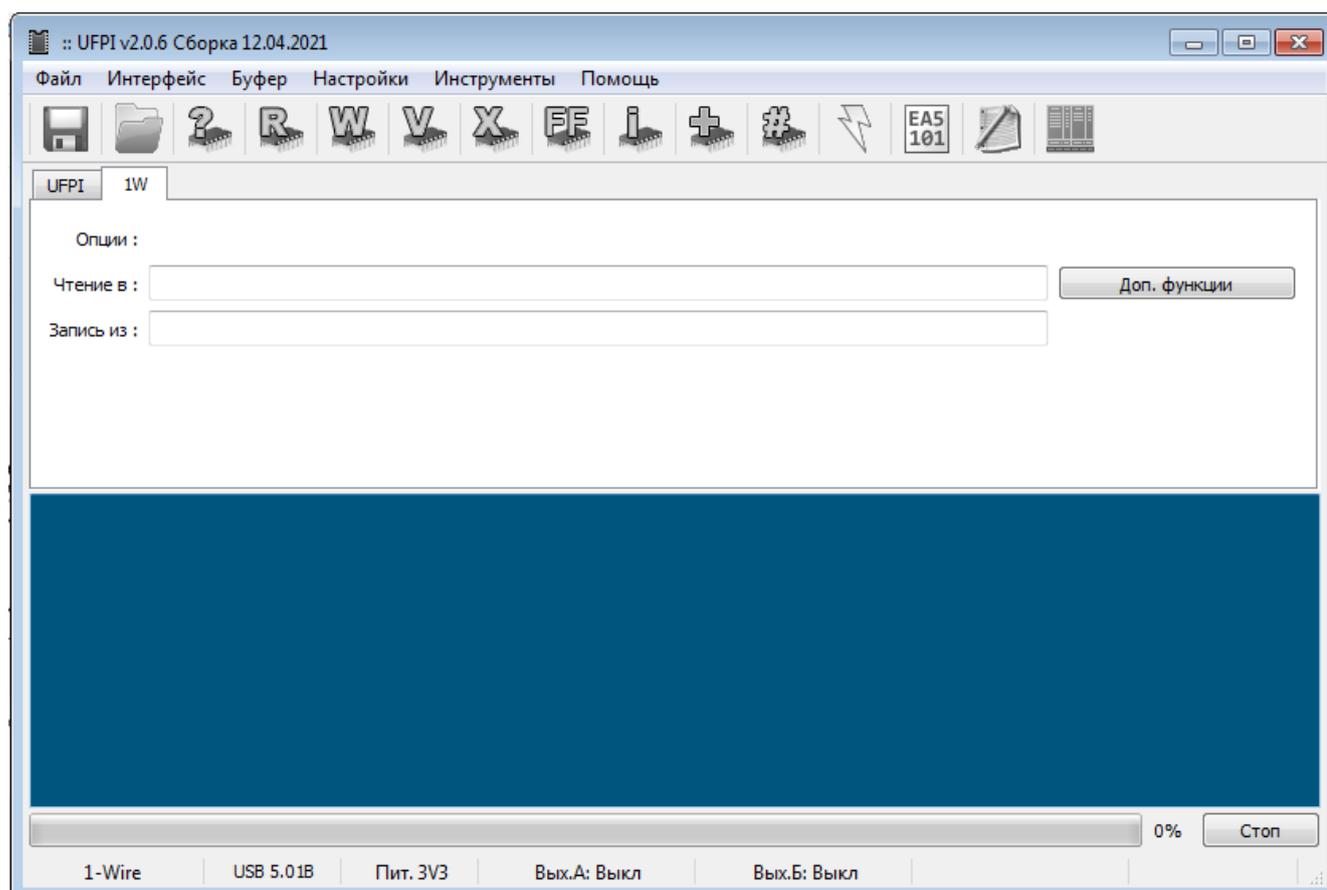
Доп. функции.



Можно выбрать «[Асинхронный режим](#)», «[Тест контактов](#)» и «Мульти-байтовая запись».

Мульти-байтовая запись, если поддерживается в текущей флеш, позволяет многократно сократить время записи при больших объемах.

4.5 1-Wire.



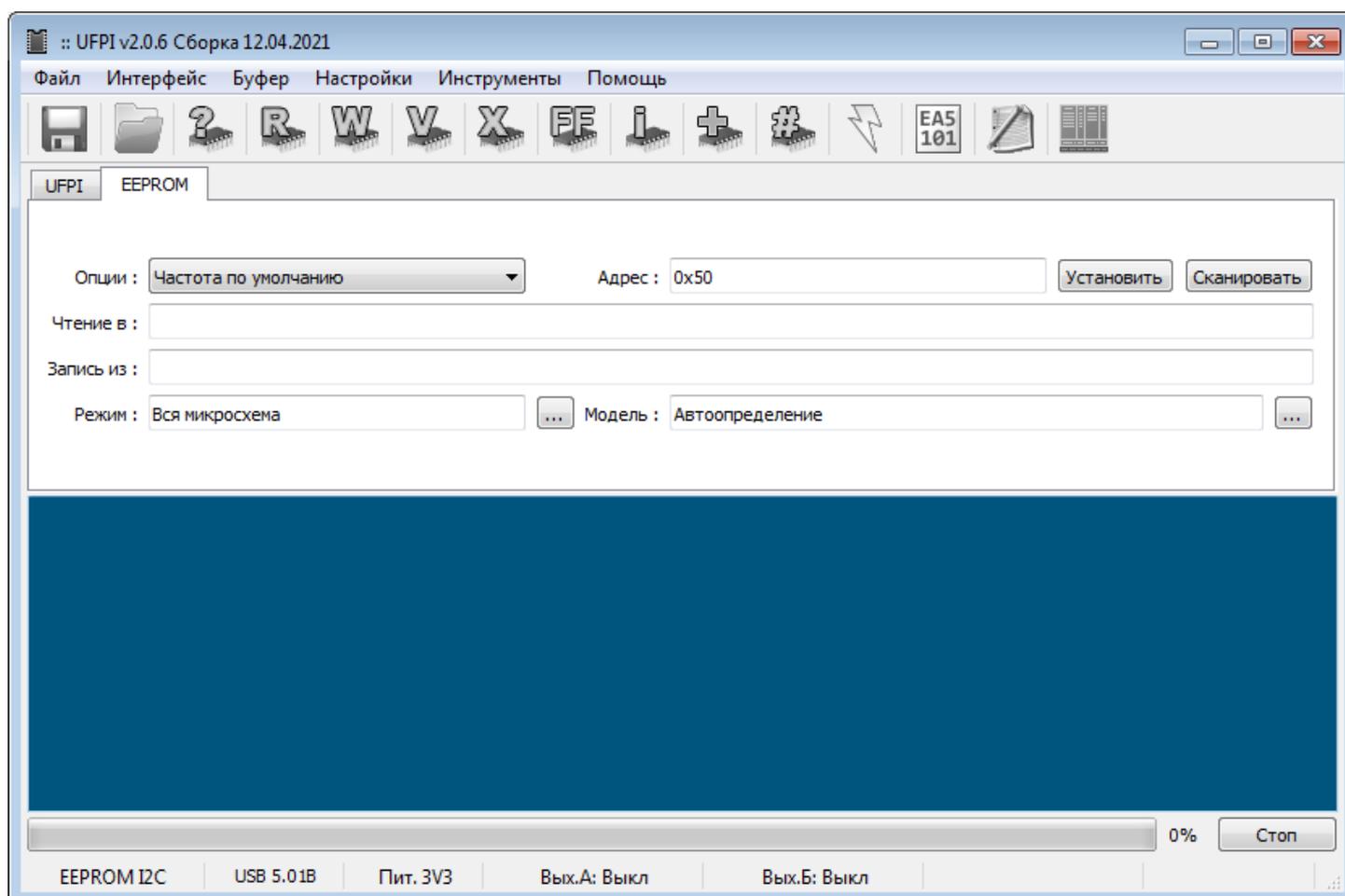
Сокеты.

Работает с сокетом [2046](#) или [2060](#).

Особенности.

- Этот модуль не имеет стандартных функций чтения/записи и прочего. В нем работа происходит через скрипты. На форуме имеются готовые скрипты, описания и обсуждения к ним.
- Подключается на GND и ногу SDA (1W IO). Распиновка по [схеме](#).

4.6 EEPROM I2C.



Сокеты.

Работает с сокетами [2046](#) или [2060](#).

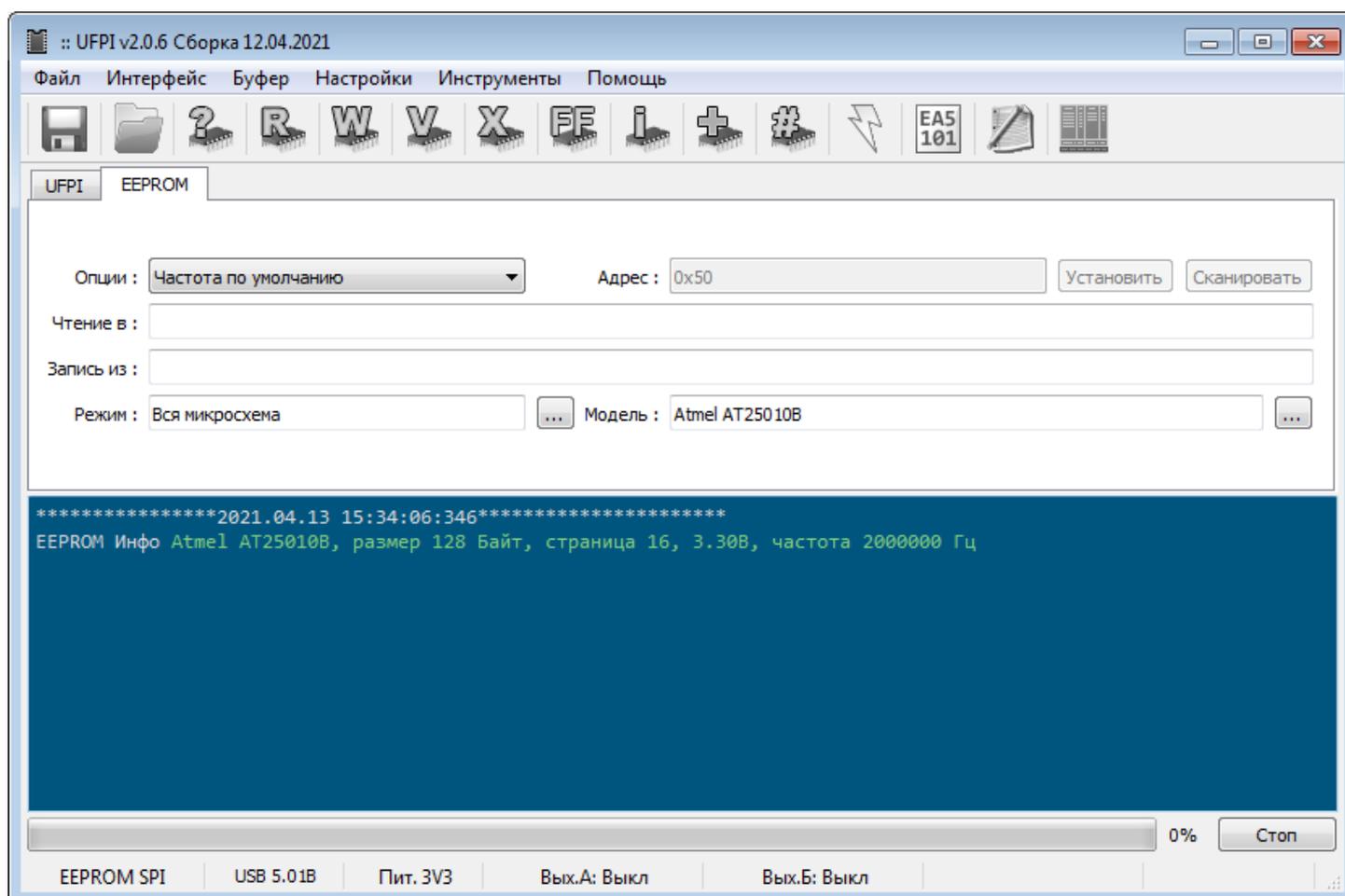
Особенности.

- Помимо обычных функций чтения/записи имеет уникальную методику автоопределения размера EEPROM 24сXX по нажатию на кнопку чтения ID <?>.
- Микросхемы 24сXX не требуют стирания перед записью, кнопка стирания в этом модуле производит запись 0xFF на весь объем микросхемы.
- Частота и адрес для обычных 24сXX определяются автоматически и не требуют изменений.
- Возможность работы с отдельными регионами микросхемы через [режим доступа](#).
- Возможность работы скриптами.

Адрес.

В модуле есть возможность установить произвольный адрес шины микросхемы. Для этого в поле «Адрес:» нужно ввести известный или полученный в логе при сканировании и нажать кнопку «Установить». Чтобы просканировать адреса, нужно нажать на кнопку «Сканировать», в логе будут выведены адреса, по которым был получен ответ.

4.7 EEPROM SPI.



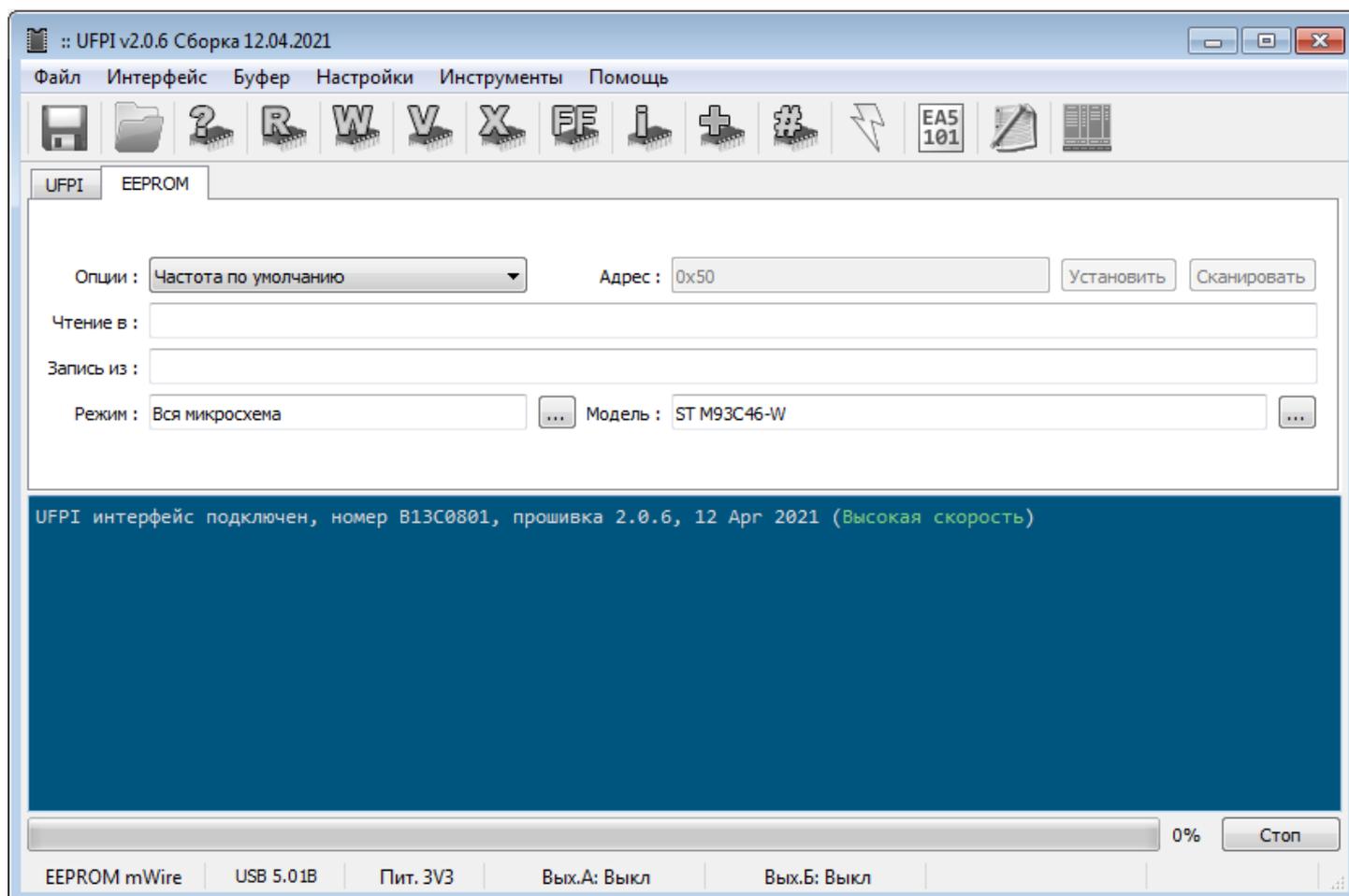
Сокеты.

Работает с сокетами [2046](#) или [2060](#).

Особенности.

- Предназначен для работы со старыми 25XXX/95XXX микросхемами. Эти микросхемы не имеют функции автоопределения и перед работой с ними требуется выбрать модель из списка.
- Работа аналогична обычным [24сXX EEPROM](#).

4.8 EEPROM Microwire (3-Wire).



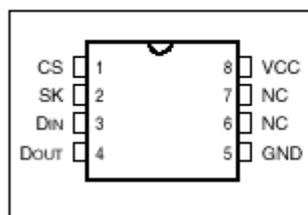
Сокеты.

Работает с сокетами [2046](#) или [2060](#).

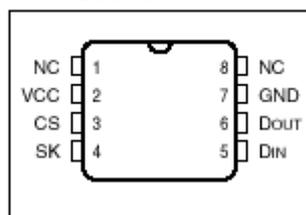
Особенности.

- Работа аналогична обычным [24сXX EEPROM](#)
- 93сXX помимо напряжения и разрядности может отличаться распиновкой. На это стоит обращать особое внимание, [сверяя при подключении](#).

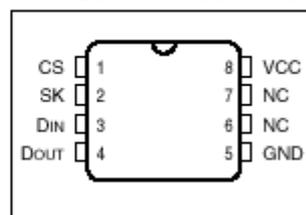
PIN CONFIGURATION
8-Pin DIP



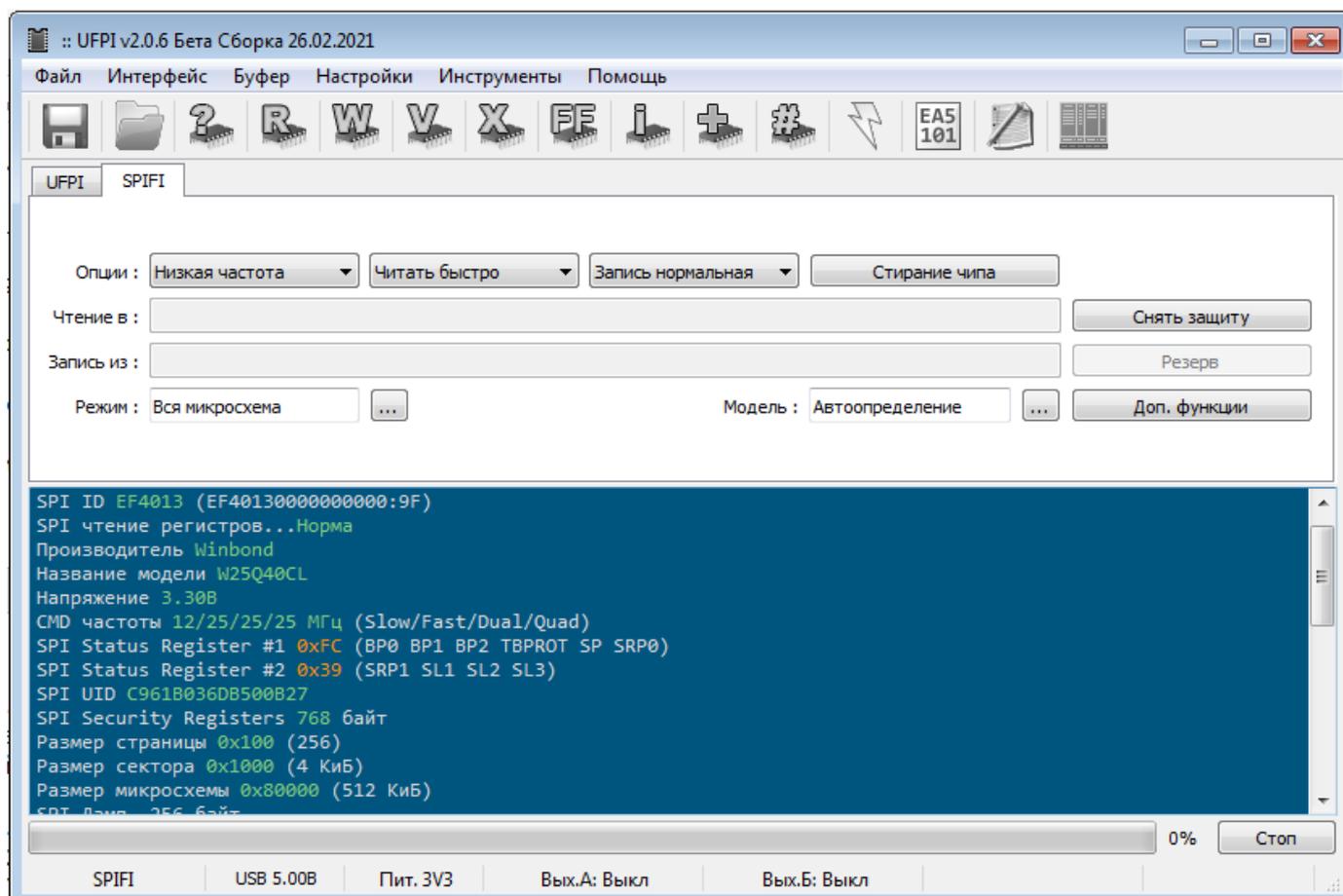
PIN CONFIGURATION
8-Pin JEDEC Small Outline "G"



PIN CONFIGURATION
8-Pin JEDEC Small Outline "GR"



4.9 SPI Flash.



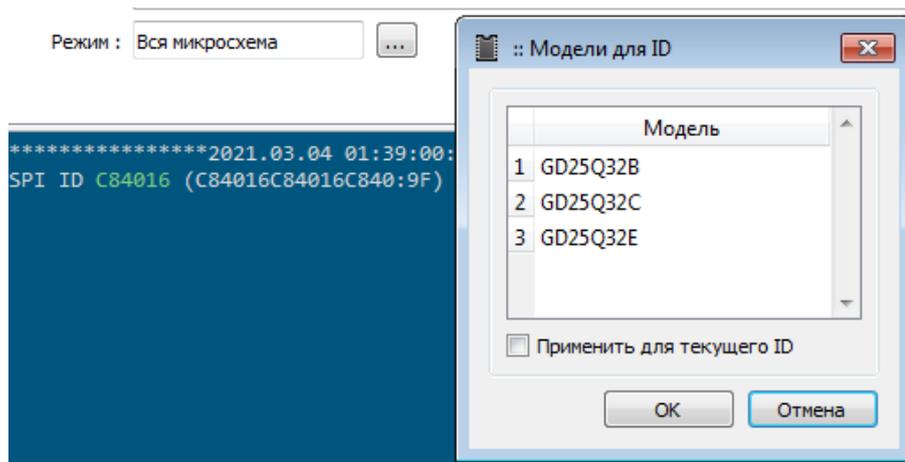
Сокеты.

Работает с сокетами [2046](#) или [2060](#).

Особенности.

- Модуль универсален и даже не требует присутствия модели микросхемы в базе. Для неизвестных микросхем как правило автоматически успешно применяются параметры из ID/SFDP/CFI. Этого обычно всегда хватает для чтения/разблокировки/стирания и записи.
- Имеется возможность [составлять конфигурации вручную по кнопке «+»](#).
- Возможность работы скриптами.

Чтение ID.



При совпадении прочитанного ID с несколькими различными микросхемами, выдается окно выбора для уточнения необходимой модели. Следует выбрать нужную и нажать кнопку «ОК». Если во время работы необходимо многократно читать ID одинаковых или одной и той же флеш, то можно поставить галочку «Применить для текущего ID». Тогда при каждом чтении ID будет применяться выбранная модель и запросов выводиться не будет до смены ID (например при установке другой микросхемы), перезапуска программы, либо выбора в «Доп. функциях» пункта «Очистить модель пользователя»

При определении флеш по приоритету используются параметры «файла конфигурации» - «внутренняя база» - «SFDP» - «размер по ID исходя из стандартов JEDEC». Если флеш по ID отсутствует в конфигурационных файлах или базе, то в логе ID подсвечивается желтым цветом.

Опции:



Набор частот — Значения выведены в логе при чтении ID «CMD частоты 50/100/100/100 МГц (Slow/Fast/Dual/Quad)». Варианты Низкая/Средняя/Максимальная/ISP.

Режим чтения — Медленно/Быстро/Dual/Quad. Соответствует режимам чтения Normal/Fast и Dual/Quad (некоторые режимы могут не поддерживаться флеш) соответственно спецификации микросхемы.

Стирание чипа — Стирание чипа одной командой, если такая поддерживается. Если стирать кнопкой «X», это займет больше времени для микросхемы целиком. Но разделено из расчета совместимости работы с отдельными блоками флеш.

Снять защиту.

По кнопке отправляется команда разблокировки, либо установки статусрегистров запрещающих запись в 0. Т.е. снимается запрет на запись.

Стоит понимать, что в некоторых флеш эти статусрегистры защищены от обнуления и записанная блокировка уже не снимается.

Нужно знать, что некоторые старые SPI флеш имеют регистры блокировки, которые восстанавливаются, если снять питание с флеш. В таком случае нужно включить питание в режим [«Всегда включено»](#).

Режим (Доступа).

Работа в модуле аналогична как и в [других](#).

Модель.

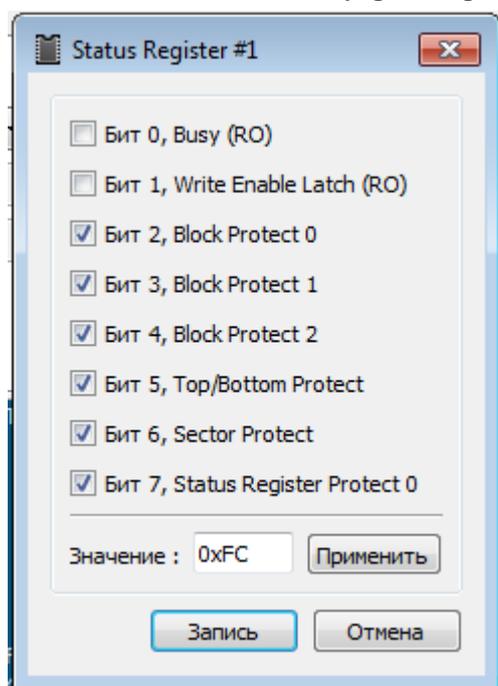
Ручной выбор. По умолчанию работает автоопределение.

Доп. функции. (некоторые пункты появляются после чтения ID)

Меню может отображать дополнительные пункты в зависимости от текущей модели флеш.

Status Register #1/#2/#3...

В зависимости от модели флеш будут отображены доступные статусрегистры. Вся работа с ручным выставлением значений статусрегистров SPI проходит через это меню.



Значения устанавливаются либо выставляя флажки, либо вводя шестнадцатеричное значение (не забывать при этом нажимать кнопку «Применить»).

Все регистры и в логе (при чтении ID, если включен подробный вывод информации) и в окне установки подписаны согласно документации на микросхему.

После нажатия кнопки «Запись», происходит запись и сверка записанного статуса. При несовпадении в логе выделяется предупреждение желтым цветом.

Security Registers (OTP).

Дополнительные регистры, могут содержать пользовательские данные или регистры защиты. При наличии таких пользовательских регистров во флеш, в логе (при чтении ID, если включен подробный вывод информации) будет описано его наличие и размер (например «SPI Security Registers 768 байт» или «SPI OTP 64 байт»). В данном меню можно выбрать пункт читать/писать в файл/буфер или стирать, если поддерживается микросхемой.

Очистить модель пользователя.

Подробнее описывалось выше в [пункте](#) про чтение ID.

SFDP режим.

Режим по умолчанию. Работать определение будет, как описано [выше](#).

Выключено. Пункт проверки SFDP будет пропущен.

«SPI режим», «Обратная частота» и «Чтение данных»

Эти параметры относятся к стандартам и режимам протокола. Все SPI флеш будут работать в состоянии «По умолчанию». Эти параметры могут пригодиться при необходимости изменения режима SPI в работе со скриптами.

Не записывать пустые данные.

Пропуск пустых страниц во время записи. Включен по умолчанию, как наиболее распространенный и правильный.

Верификация данных при записи.

Автоматически, параллельно, в процессе записи проводит сверку записанного. Т.е. разница с отдельной верификацией после записи в том, что если произошла ошибка сверки записанного блока, то процесс остановится, не дожидаясь окончания записи.

Проверка чистоты при стирании.

Аналогично предыдущему пункту, но происходит проверка на чистоту в процессе стирания.

Асинхронный.

Увеличение скорости [аналогично](#) как и в других модулях.

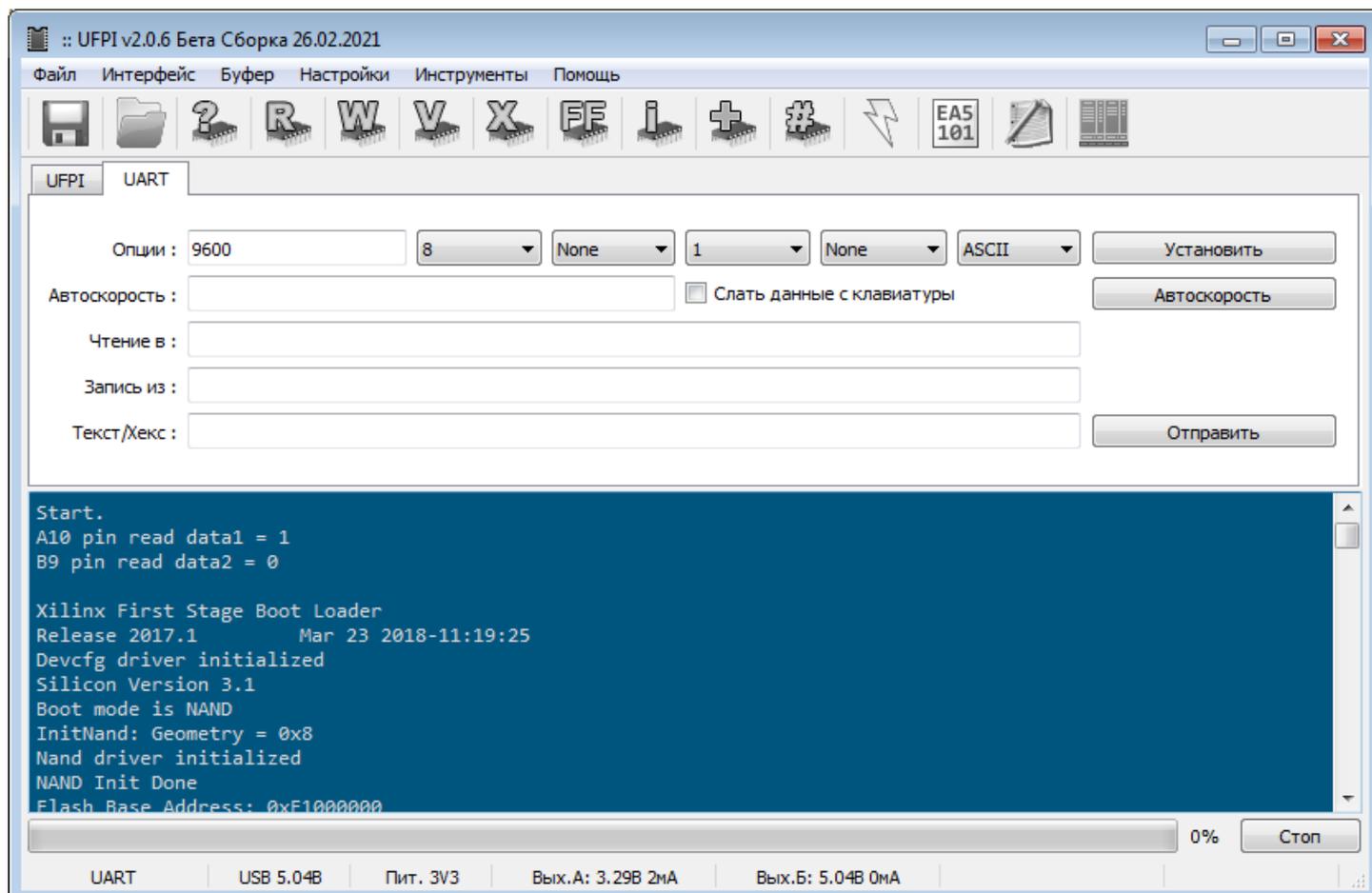
Тест контактов.

Работает аналогично как и в других [модулях](#).

Составление файлов конфигураций.

Описано в [приложении](#).

4.10 UART.



Сокет.

Работает на сокете [2044 и 2060](#).

Особенности.

- Автоопределение скорости передачи (baud rate).
- Поддержка работы скриптами.

Основные настройки.

Интуитивно понятные, как и во всех терминальных для UART программах параметры:

Скорость UART. Существует общепринятый ряд стандартных скоростей: 300; 600; 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200; 230400; 460800; 921600 бод.

Количество бит данных (обычно 8).

Наличие бита четности.

И 1 стоповый бит.

Управление потоком.

Тип выводимой информации в лог (HEX/ASCII).

На картинке приведена конфигурация UART 9600 8-N-1, поток Off (ASCII). Выбрав нужную конфигурацию нужно нажать кнопку «Установить» для применения параметров. В логе пропишется примененная конфигурация.

Автоскорость.

Автоопределение скорости baudrate. Запускать желательно до начала обмена для правильной синхронизации со стартовым битом. При удачном обнаружении в поле «Автоскорость» будет подписана определившаяся скорость.

Слать данные с клавиатуры.

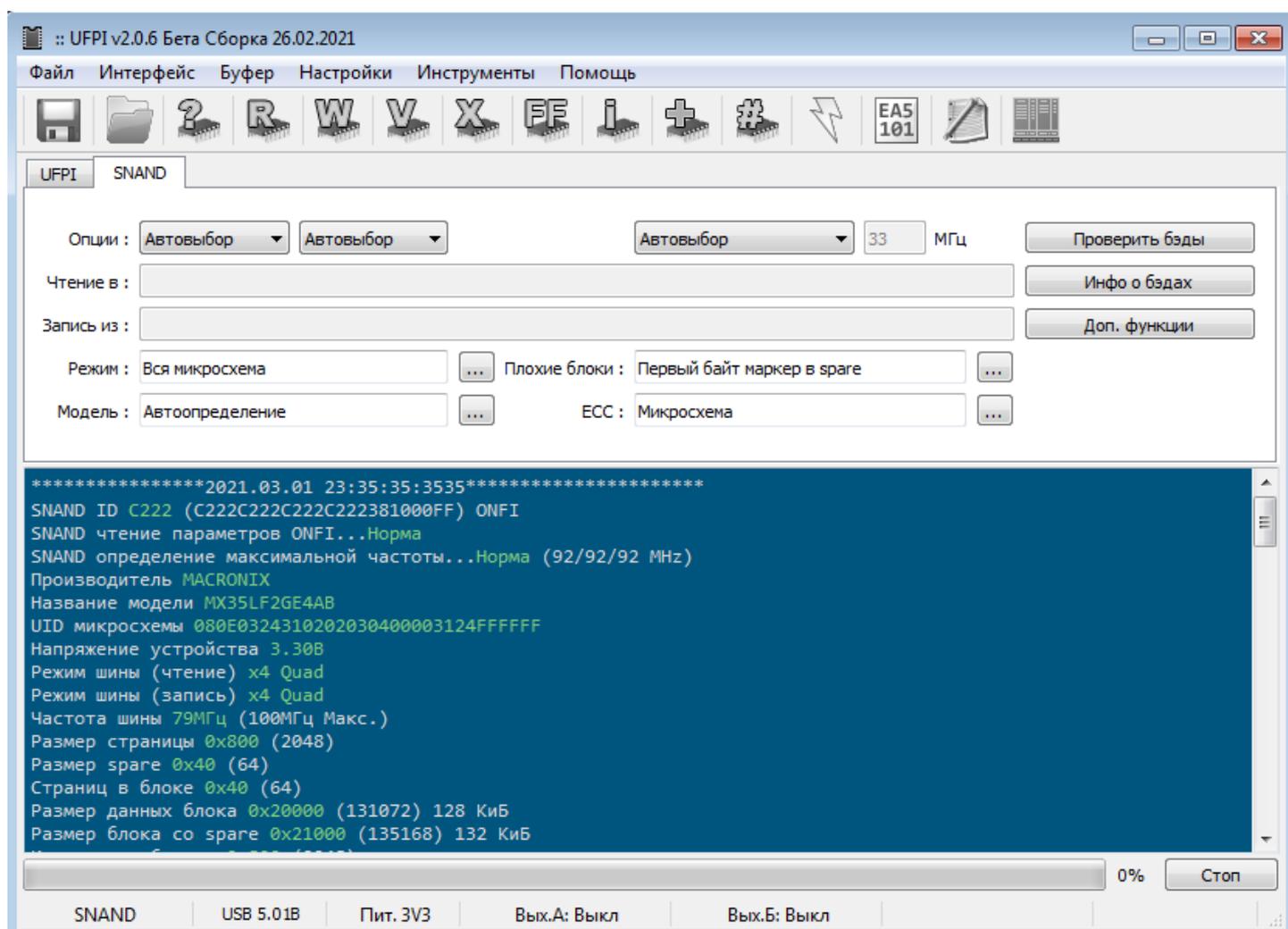
Включается перехват нажатия клавиш и отправки их кода по каналу TX UART.

Поле «Текст/Хекс» и кнопка «Отправить» предназначены для ввода и отправки целой строки. Формат в зависимости от выбранного в конфигурации (HEX/ASCII).

Прокрутка лога.

[Автопрокрутка](#) лога настраиваемая.

4.11 Serial NAND.



Сокет.

Работает на сокетах [2046](#) или [2060](#). Размещается в положении SPI.

Опции скорости и частоты.

Опции : 33 МГц

По умолчанию работает чтение/запись в режиме одной линии и частотой по умолчанию в пределах 85МГц.

Можно выставлять скорость увеличив режимы чтения записи до x2/x4 и даже увеличив частоту до 100МГц, если позволяет флеш и текущее подключение.

Режим (доступа).

Описан в соответствующем [разделе](#).

Модель.

В данном модуле нет необходимости выбирать модель вручную, поскольку все параметры содержатся

в ONFI самой флеш и применяются автоматически.

Плохие блоки.

Описано подробнее с [разделе](#) про ББ.

В отличие от обычных NAND, в SNAND WINBOND может встречаться BBM LUT. Остальное все аналогично обычным.

Кнопки «Проверить бэды» и «Инфо о бэдах», а так же в Доп. функциях меню «Плохие блоки» работают так же как в обычных [NAND](#).

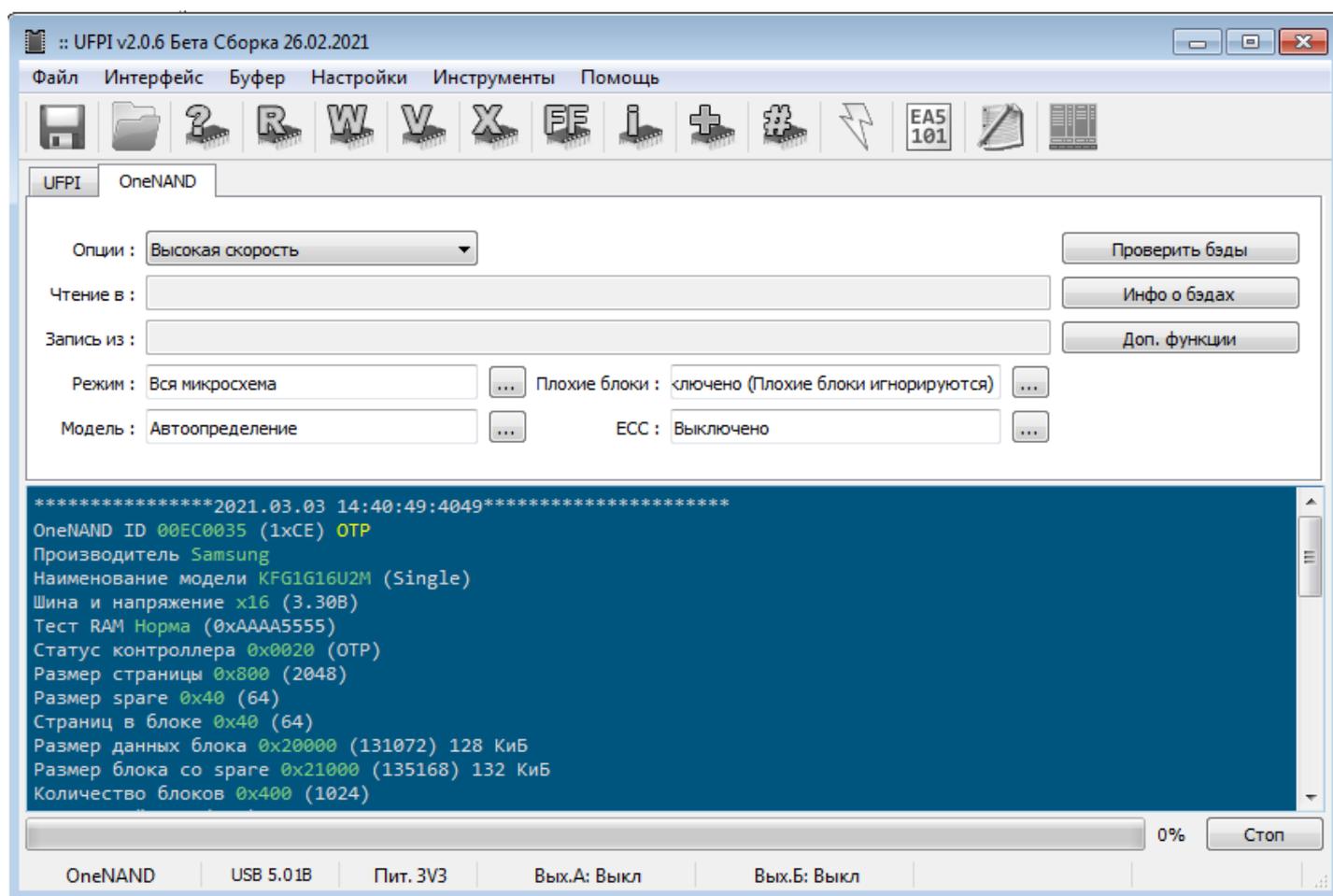
ЕСС.

Аналогично обычной NAND, за исключением того, что обычно используется внутренний контроль [ЕСС](#), который выставляется как «Микросхема»(HWECC).

Доп. функции.

Наполнение функций соответствует обычному NAND, за исключением пункта BBM LUT. Этот пункт позволяет заносить блок во внутреннюю (в контроллере флеш) [таблицу плохих блоков](#) и обрабатывать. *Операция над BBM LUT блоком однократная и не может быть отменена. Занесенный блок или таблицу очистить нельзя.*

4.12 OneNAND.



Сокет.

Работает на сокетах [2031](#) или [2032](#) и базой под BGA колодку.

Опции.

Выбор скорости: Низкая/Нормальная/Высокая.

Режим (доступа).

Описан в соответствующем [разделе](#).

Модель.

В данном модуле нет необходимости выбирать модель вручную, поскольку все флеш содержатся в базе и определяются автоматически по ID.

Плохие блоки.

Описано подробнее с [разделе](#) про ББ.

Кнопки «Проверить бэды» и «Инфо о бэдах», а так же в Доп. функциях меню «Плохие блоки» работают так же как в обычных [NAND](#).

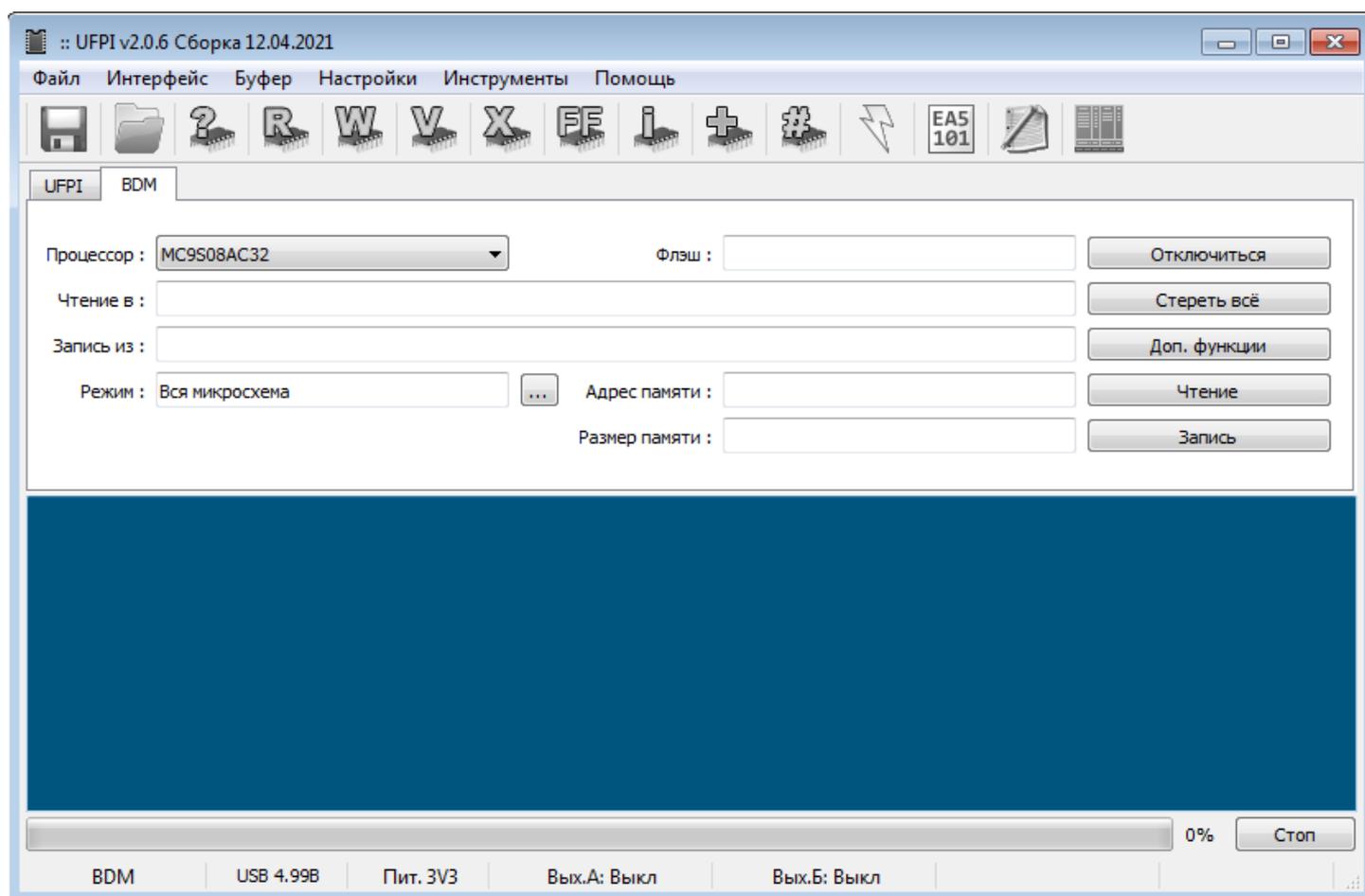
ECC.

Аналогично обычной NAND.

Доп. функции.

Наполнение функций соответствует обычному NAND, за исключением пункта Flex OneNAND. Этот пункт для переключения между чипами в корпусе микросхемы.

4.13 BDM.



Сокет.

Работает на сокете [2043](#).

Особенности.

- Возможность работы скриптами.
- Поддерживаются процессоры:
 - MC9S08AC60
 - MC9S08AC48
 - MC9S08AC32
 - MC9S08AW60
 - MC9S08AW48
 - MC9S08AW32
 - MC9S08AW16
 - MC9S08GB60
 - MC9S08GB32
 - MC9S08GT60
 - MC9S08GT32
 - MC9S08GT16
 - MC9S12XF384
 - и др.
- Прошиваются .bin файлы, которые можно конвертировать из s-records в меню «Инструменты —

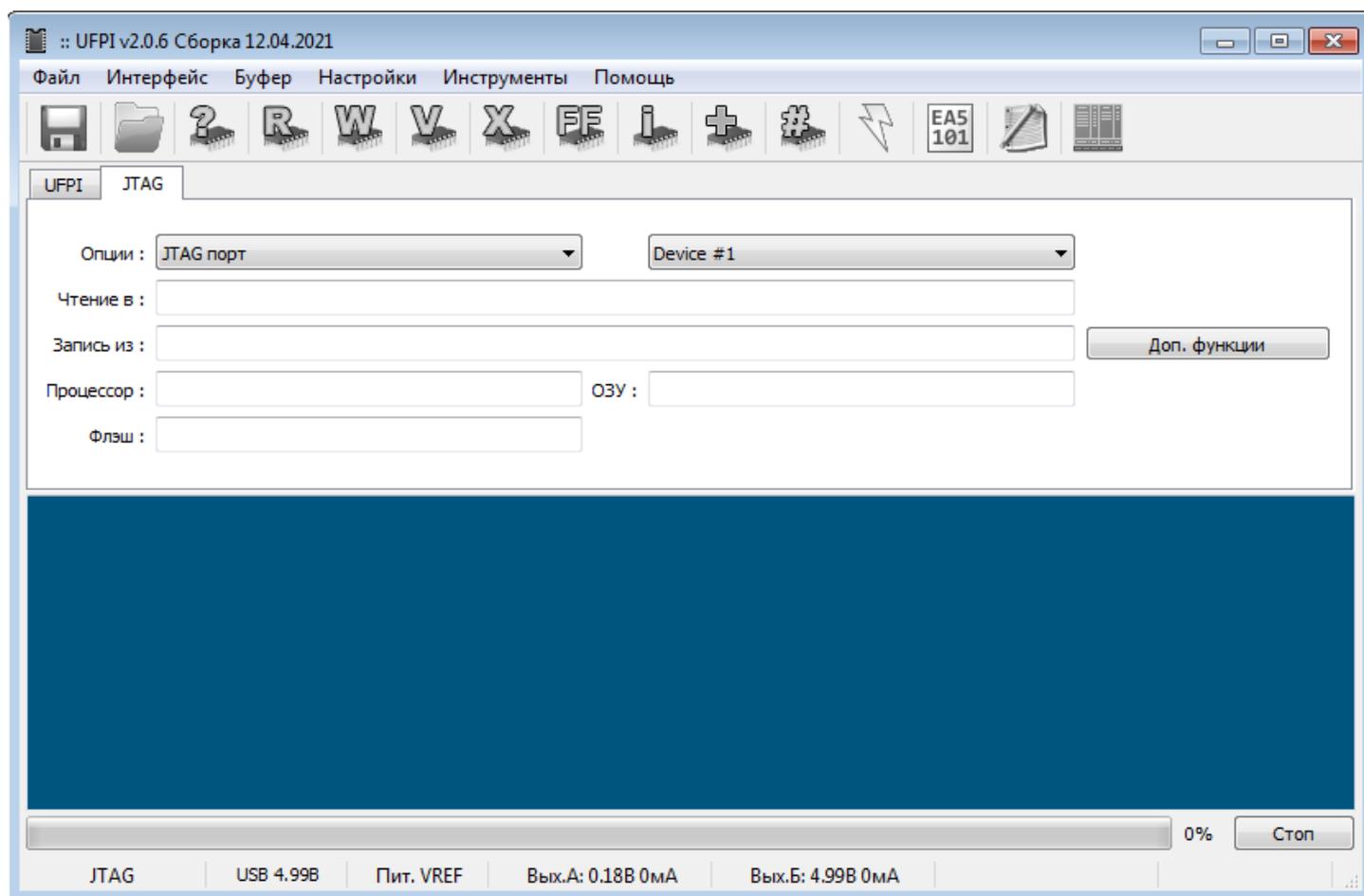
Конвертация».

- Больше информации в обсуждениях на [форуме](#).

Последовательность действий.

- Выбрать тип процессора.
- Прочитать ID.
- Работать как с обычной флешкой.
- Стереть и снять защиту кнопкой "Стереть всё".

4.14 JTAG.



Сокет.

Работает на сокете [2043](#).

Особенности.

- Для работы через этот интерфейс создается .udev файл с параметрами инициализации и конфигурацией флеш.
- Уровень VREF подается с внешнего источника соответственно уровням сигнала (обычно 3,3V).
- После подключения и чтения ID открываем вкладку определившейся флеш и работаем как обычно соответственно используемому модулю.

Поддерживаемые девайсы.

На данный момент имеются готовые UDEV файлы только процессоров S4LJ162X01 и JUPITER4E и их внутренних флеш. Для работы с этими процессорами приведены примеры на форуме:

[Режимы темы Восстановление принтера на базе процессора JUPITER4E](#)

[Восстановление принтера на базе процессора S4LJ162X01](#)

Так же имеется возможность запуска [эмулятора CMSIS DAP](#).

4.15 Mount File.

Применение.

Позволяет [подключать файловую систему](#) как виртуальный диск для работы с файлами.

С примерами можно ознакомиться в [приложении](#).

Особенности.

На данный момент отличает от «Mount Chip» лишь тем, что объектом монтирования должен быть файл. Оба модуля в основном практически равноценны, разница скорее в удобстве. Этот вариант может быть более удобным, если разделы вычитаны и требуется работать с разделами на диске перед записью во флеш.

4.16 Mount Chip.

Применение.

Позволяет [подключать файловую систему](#) как виртуальный диск для работы с файлами.

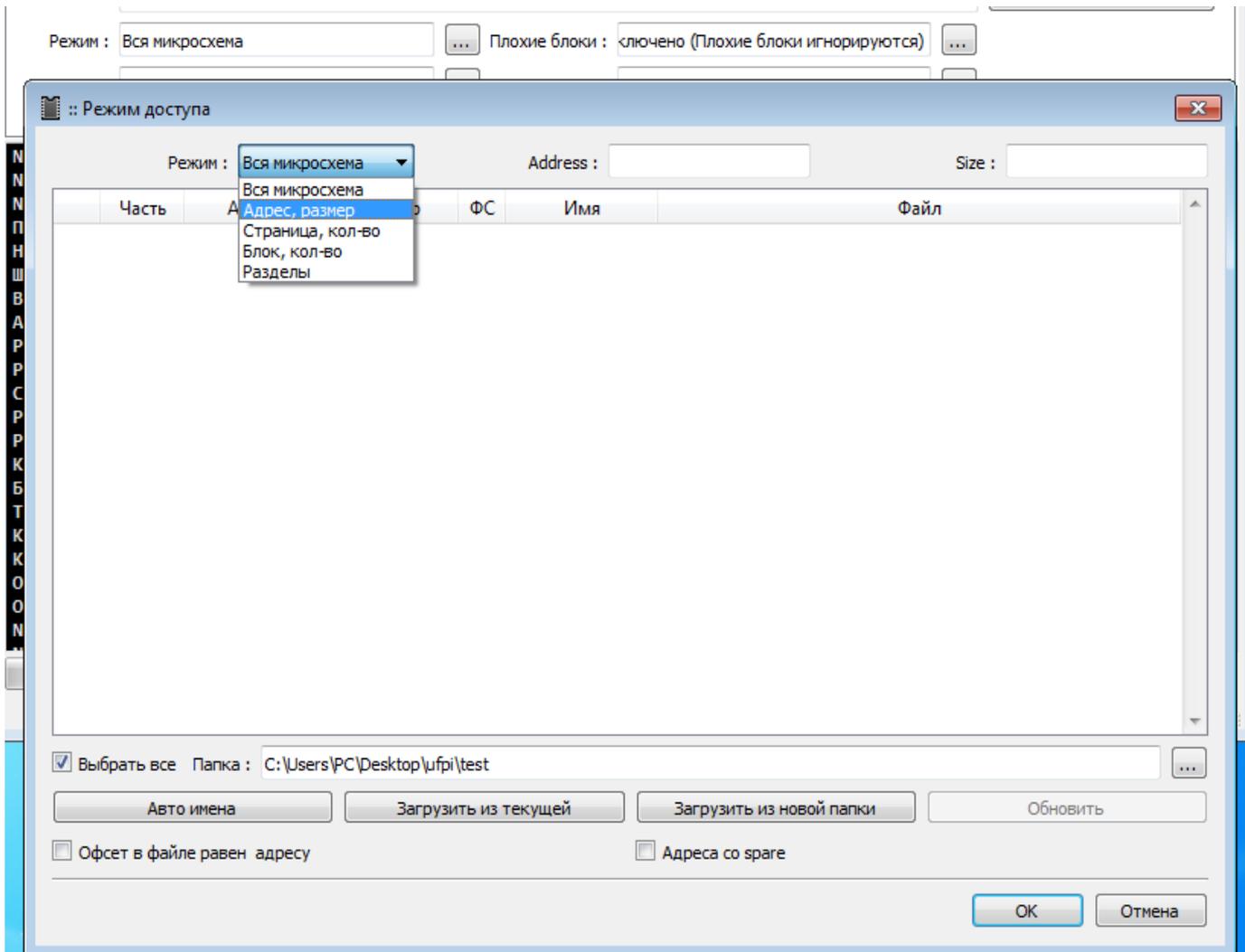
С примерами можно ознакомиться в [приложении](#).

Особенности.

На данный момент отличает от «Mount File» лишь тем, что объектом монтирования должна быть флеш. Оба модуля практически равноценны, разница скорее в удобстве. Этот вариант может быть более удобным, если требуется работать с разделами не тратя время на вычитывание и обратную запись дампа на диск.

5 Режим (доступа).

Позволяет работать как с флеш целиком, так и задавать отдельно адреса, номера страниц, блоков, разделов. Работает и присутствует практически в каждом модуле, но с учетом особенностей каждого. Ниже рассматривается на примере NAND.



Режим: Адрес, размер.

Размер нужно вводить кратно страницам. Причем если в меню «Доп.функции» установлена галочка «Без spare», то размер страницы нужно вводить без spare. Если иначе, то со spare.

Адрес необходимо вводить без учета spare, если не установлена галочка в «Доп. функции» «Адреса со spare». Вводить можно в десятичном и шестнадцатеричном виде.

Режим: Страница, кол-во.

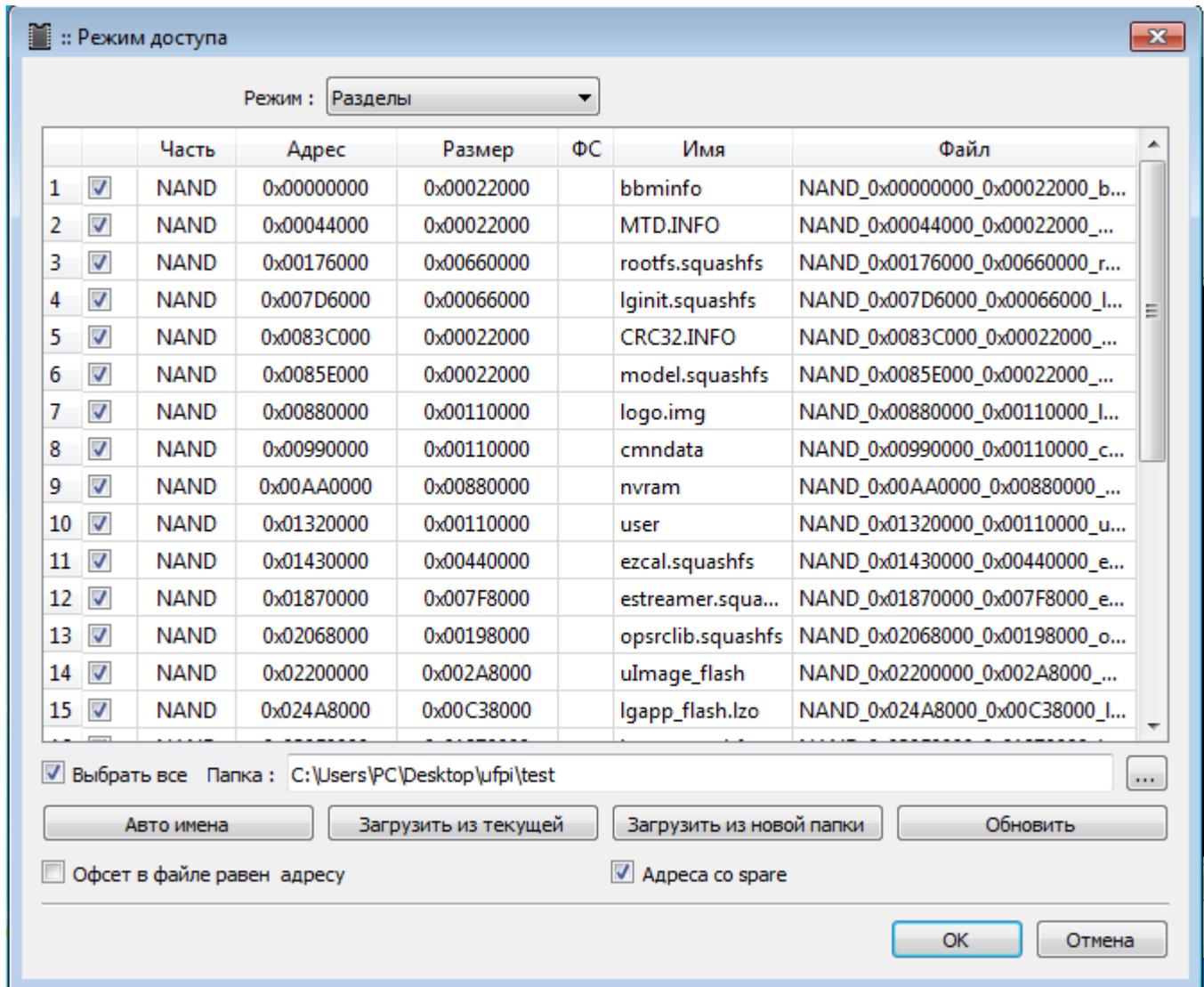
Предварительно нужно прочитать ID и определить геометрию флеш. Вводить номер страницы начала требуемого региона и количество страниц. *Стоит учесть, что минимальный размер стирания у NAND это размер блока!*

Режим: Блок, кол-во.

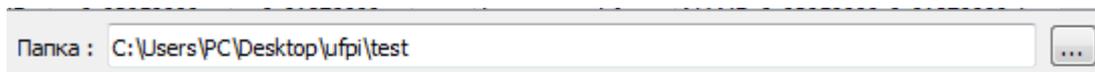
То же самое, что и со страницами, только размерности в блоках.

Режим: Разделы

В этом режиме можно работать с разделами флеш. Разделы можно загрузить из UDEV файла, заполнить вручную, либо получить с помощью анализа дампа.



Все поля внутри таблицы можно редактировать по двойному щелчку. *Файлы разделов пишутся и сохраняются в указанную папку. Файлы в полях «Чтение в» и «Запись из» не используются в этом режиме! При дописывании пути папки клавиатурой в строке, требуется нажать Enter для применения.*



Для выбора необходимых файлов/регионов (имена присваиваются автоматически из имени и размещения раздела, если не указано другого), которые требуется читать/писать нужно выставить флажки в соответствующих строках.

Меню по нажатию правой кнопкой

18	<input checked="" type="checkbox"/>	NAND	0x05588000	0x00022000		tplist.squashfs	NAND_0x05588000_0x000
19	<input checked="" type="checkbox"/>	NAND	0x055AA000	0x002A8000		uimage flash	NAND_0x055AA000_0x000
20	<input checked="" type="checkbox"/>	NAND	0x05852000	0x00000000			NAND_0x05852000_0x000
21	<input checked="" type="checkbox"/>	NAND	0x0648A000	0x00000000			NAND_0x0648A000_0x000
22	<input checked="" type="checkbox"/>	NAND	0x07CFA000	0x00000000			NAND_0x07CFA000_0x000
23	<input checked="" type="checkbox"/>	NAND	0x07D1C000	0x00000000			NAND_0x07D1C000_0x000
24	<input checked="" type="checkbox"/>	NAND	0x00088000	0x00000000			NAND_0x00088000_0x000
25	<input checked="" type="checkbox"/>	NAND	0x000AA000	0x00000000			NAND_0x000AA000_0x000
26	<input checked="" type="checkbox"/>	NAND	0x00066000	0x00000000			NAND_0x00066000_0x000
27	<input checked="" type="checkbox"/>	NAND	0x00154000	0x00000000			NAND_0x00154000_0x000

Добавить раздел

Удалить раздел(-ы)

Открыть файл

Выбрать имя файла

Создать пустые файлы

Сохранить в UDEV файл (UTF-8)

Добавить раздел.

При составлении можно добавить новый раздел в конец таблицы.

Удалить раздел(-ы).

Удалить выделенный раздел. Можно выделить несколько через Ctrl или Shift.

Открыть файл.

Открывает диалог выбора существующего файла региона *для записи* в выбранной строке.

Выбрать имя файла.

Открывает диалог для задания расположения и имени файла, который будет *создан при чтении, если не существует*, либо перезаписан.

Создать пустые файлы.

Создает в рабочей папке по списку пустые файлы с заданными именами и нулевым размером. Если файлы существуют, то переписываются соответственно нулевыми размерами.

Сохранить в UDEV файл (UTF-8).

Создает [UDEV](#) файл согласно составленным разделам в выбранную папку.

Адреса со spare (флажок).

Ввод размеров и адресов при установленной галочке следует производить с учетом размера spare (как в дампе). И вывод в логе адресов в этом случае так же будет с учетом spare и подсвечен.

Офсет в файле равен адресу (флажок).

При установленном флажке из полного дампа будет записан выделенный регион. Если нужно использовать в операции чтения, то файл должен быть соответствующего размера.

Загрузить из новой/текущей папки.

Позволяет загрузить из папки заранее считанные регионы имеющие соответствующие имена. Для работы с разделами и сразу с файлами, необходимо называть файлы разделов именами "по соглашению" типа boot1_0x0_0x1000_zboot.bin и нажимать в режиме доступа разделы "Загрузить из папки".

	Часть	Адрес	Размер	ФС	Имя	Файл
1	<input checked="" type="checkbox"/>	NAND	0x00000000	0x00022000	bbminfo	NAND_0x00000000_0x00022000_b...
2	<input checked="" type="checkbox"/>	NAND	0x00044000	0x00022000	MTD.INFO	NAND_0x00044000_0x00022000_...
3	<input checked="" type="checkbox"/>	NAND	0x00066000	0x00022000	mac address st...	NAND_0x00066000_0x00022000_...
4	<input checked="" type="checkbox"/>	NAND	0x00088000	0x00022000	cert.squashfs	NAND_0x00088000_0x00022000_c...
5	<input checked="" type="checkbox"/>	NAND	0x000AA000	0x000AA000	authcxt	NAND_0x000AA000_0x000AA000_...
6	<input checked="" type="checkbox"/>	NAND	0x00154000	0x00022000	keyrsrv	NAND_0x00154000_0x00022000_k...
7	<input checked="" type="checkbox"/>	NAND	0x00176000	0x00660000	rootfs.squashfs	NAND_0x00176000_0x00660000_r...

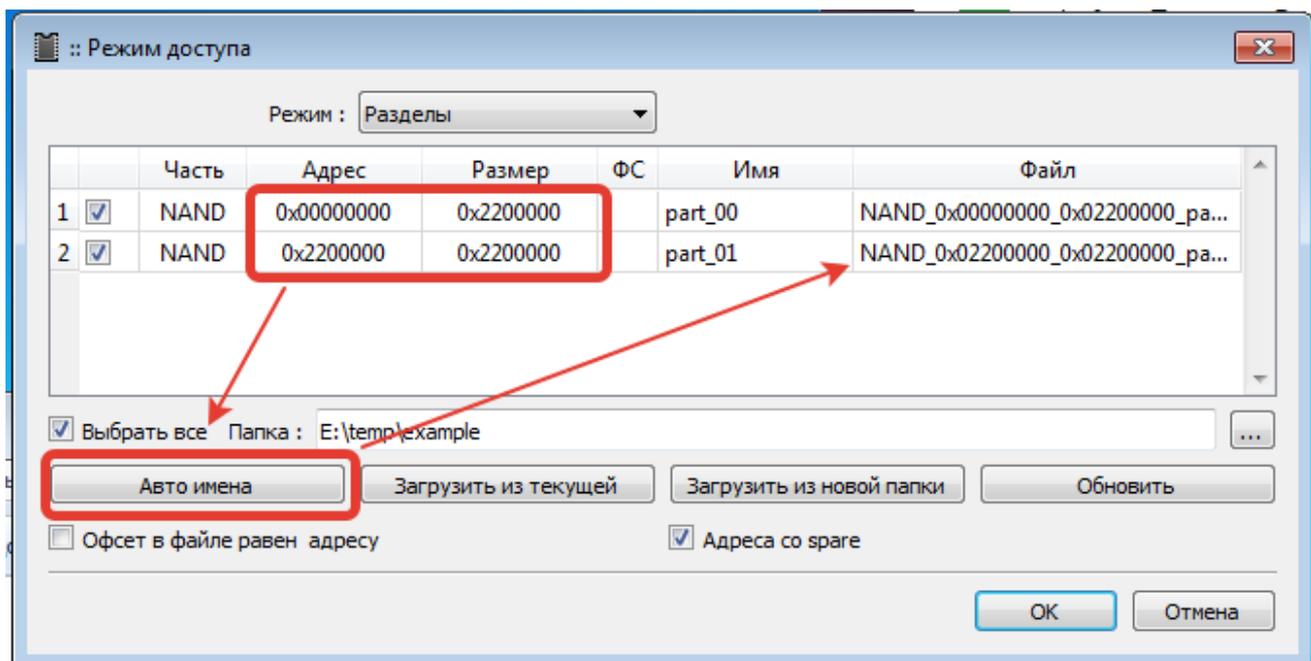
Компьютер > work (E:) > temp > example

Добавить в библиотеку | Общий доступ | Новая папка

Имя

- NAND_0x00000000_0x00022000_bbminfo.bin
- NAND_0x000AA000_0x000AA000_authcxt.bin
- NAND_0x00AA0000_0x00880000_nvram.bin
- NAND_0x007D6000_0x00066000_lginit.squashfs.bin
- NAND_0x07CF4000_0x00022000_tnlist.squashfs.bin

Авто имена.



После заполнения таблицы и выставки размеров и адресов, нажав на кнопку, автоматически заполнится столбец «Файлы». С этими именами будут сохранены файлы регионов при чтении.

6 Управление питанием.



Левой кнопкой мыши включает/выключает заданное питание.

Правой кнопкой мыши показывает меню настроек.

NAND	USB 4.99В	Пит. 3V3	Вых.А: 3.29В 2мА
------	-----------	----------	------------------

При включенном питании в строке статуса отображено напряжение и ток потребляемый каналом питания. В SPI, NAND и подобных модулях для питания используется один канал, а для eMMC задействованы оба канала и второй канал показывает так же напряжение и ток второго канала. Основной канал, на котором можно регулировать напряжение при работе программатором является канал А.

По превышении тока 500мА на канале ([если не отключена](#)) то сработает защита по перегрузке. В логе появится "UFPI интерфейс отключен" и на программаторе начнет моргать светодиод соответствующий каналу, который ушел в аварию ([канал А - VCC/VREF соответствует LED1 или канал Б - VUSB соответственно LED2](#)).

Если на основной вкладке установлен "Игнор аварии", то бокс не отключится аварийно, продолжит работать, но в логе при превышении порога 500мА появится сообщение "UFPI авария по питанию! Выход А"

Включать "Игнор аварии" не рекомендуется при повседневной работе в обычных условиях! Лишь в некоторых случаях с пониманием последствий при работе внутрисхемно, когда требуется питать от бокса прошиваемое устройство.

Для внутрисхемной работы с большими потребителями тока можно воспользоваться режимом Vref. Для этого необходимо переключить на сокете в соответствующее положение выключателя и подключить внешний источник питания к выходу VCC прошиваемой флеш и соединить с VCC сокета. При этом при подаче напряжения с внешнего источника питания (например ЛБП), будут запитаны внутренние выходные буферы бокса и в строке статуса будет отображаться напряжение.

Остальные напряжения питания выставляемые выключателями на сокете описаны в разделе [распиновки](#).

По нажатию правой кнопкой мыши на [кнопку с молнией](#) появляется меню, в нем по умолчанию все флажки сняты (так же для отмены изменений можно нажать на пункт "Напряжение по умолчанию"). Напряжение при чтении ID используется согласно выставленному на сокете, затем после чтения ID определяется ее модель и/или выставляется соответствующее напряжение автоматически.

Например:

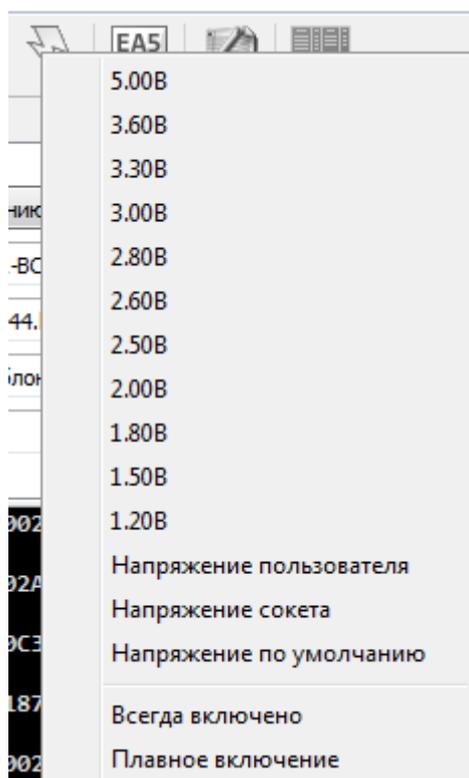
прочитав ID у NAND512R3A в сокете с выставленным напряжением 3.3В, программа автоматически определит ее как 1.8 вольтовую и работать с ней будет на 1.8В.

прочитав ID у W25Q64FW с выставленным на сокете 3.3В напряжением, в логе увидим, что определилась как W25Q64FV, поскольку они имеют абсолютно одинаковые ID!!! Если выставить напряжение на сокете 1.8В и прочитать еще раз ID, то в логе уже будет написано W25Q64FW. Все соответственно напряжению.

В каждом модуле настройка индивидуально запоминается!

Можно выставлять любое из перечисленных напряжений (состояние выключателей питания сокета при этом должно быть отличным от Vref), либо в пункте "Напряжение пользователя" можно ввести ручную цифрами в милливольтгах. При этом если при выставленном через меню напряжении изменить его переключателем на сокете, то оно изменится так же соответственно выставленному на сокете.

Т.е. напряжение выставленное в меню не мешает выставлять его на сокете, и наоборот.



Пункт "Напряжение сокета" переключает с текущего напряжения на напряжение выставленное на сокете.

Флаг "Всегда включено" может быть полезен при работе внутрисхемно, со скриптами, старыми SPI (которые не сохраняют статус после снятия питания). В модуле eMMC питание включено постоянно по умолчанию и не требуется активации этого флага.

Флаг "Плавное включение", позволяет плавно поднимать напряжение при включении, что часто помогает при внутрисхемном подключении, не давая срабатывать защите при начальном заряде конденсаторов и переходных процессах при включении питания от бокса.

7 Файлы устройств .UDEV.

Применение и особенности.

UFPI имеет возможность работы с файлами устройств (.UDEV). Файлы устройств это текстовые файлы со своим форматом (похожим на INI), которые содержат в себе информацию необходимую для работы с вашим устройством для каждого конкретного модуля. Это название, тип процессора, адрес памяти, строки инициализации, данные о разделах и т.п.

UDEV можно открывать используя горячие клавиши. В секции [INFO] можно писать любую информацию, которая будет выводиться при загрузке UDEV. Кодировка UTF-8, а так же выделять цветом, применяя тэги #C1, #C2...

UDEV можно составлять в обычном блокноте, либо скриптами ([например для поиска разделов](#)) или в [режиме доступа «Разделы»](#), через меню по правой кнопке мыши.

Структура.

На примере NAND для TV SAMSUNG D5500.

[DESC]

```
; Имя устройства
Name = D5500
```

```
; Типа памяти
; eMMC, SD, NAND, ONENAND
FlashType1 = NAND
```

```
; Базовый адрес, используется только с JTAG и т.п.
FlashBase1 = 0x0
```

```
; Алгоритм ECC коррекции
; 0 - выключено
; 1 - схема
; 2 - HW
ECCAlgo = 1
```

```
; Схема ECC коррекции
ECCScheme = MSTAR_P8K_SP436_CW8_S12L42
```

```
; Анализ после чтения ID
; 0 - выключено
; 1 - dump
; 2 - dump, ic
; 3 - ic
; 4 - ic, dump
AnalysisOnID = 1
```

[BBM]

```
; Стирать плохие блоки или нет
bbErase = false
```

```

; Обнаружение плохих блоков
; 0 - выключено
; 1 - во время стирания
; 2 - первый байт маркер spare
; 3 - шестой байт маркер spare
; 4 - произвольная позиция маркера в spare
; 5 - произвольная позиция маркера в странице
; 6 - таблица плохих блоков
; 7 - BVM LUT (Serial NAND)
bbDetect = 2

```

```

; Управление плохими блоками
; 0 - выключено
; 1 - пропустить плохой блок
; 2 - использовать резервную зону
bbMgmt = 2

```

```

; Значение маркера
bbMarkerVal = 0x00

```

```

; Оператор маркера
; 0 - равно
; 1 - неравно
bbMarkerOp = 0

```

```

; Пользовательская позиция маркера
bbMarkerPos = 0

```

```

; Тип таблицы плохих блоков
; OFF - не используется
; AUTODETECT - автодетект
; LG BVMINFO
; Samsung RFS
bbTableType = Samsung RFS

```

```

; Номер блока таблицы плохих блоков
bbTableBlock = 0

```

```

[NAND]
;офлайн геометрия NAND для монтирования и анализа ББ.
PageData = 8192
PageSpare = 436
PagesInBlock = 128
Blocks = 2076

```

```

[INFO]
; любая текстовая информация, которая выводится при загрузке

```

```

[PARTITIONS]
; информация для работы с разделами

```

Пример .UDEV разделов для eMMC

```

[DESC]

```

```
Name = Partitions
FlashType1 = eMMC
FlashBase1 = 0
```

```
[PARTITIONS]
```

```
;Активировать режим доступа «Разделы».
```

```
PartitionsMode = true
```

```
;Начальный адрес, Размер раздела, Имя раздела, Часть/раздел(user/boot1/boot2/RPMB), Имя файла, Смещение в файле, Файловая система
```

```
0x0000000000,0x00200000,Partitions_table,USER,
```

```
0x0000200000,0x00300000,MBOOT,USER,
```

```
...
```

```
0x0003080000,0x00040000,RTPM,USER,
```

```
0x00030C0000,0x2BC00000,system,USER,USER_0x00030C0000_0x2BC00000_system.bin,0,EXT4
```

```
0x002ECC0000,0x40400000,userdata,USER,,,EXT4
```

Для работы с разделами и сразу с файлами, необходимо называть файлы разделов именами "по соглашению" типа boot1_0x0_0x1000_zboot.bin и нажимать в режиме доступа разделы "Загрузить из папки".

Пример .UDEV разделов для NAND

```
[PARTITIONS]
```

```
PartitionsMode = true
```

```
; если адреса разделов указаны с учетом spare.
```

```
RawAddrMode = true
```

```
;если размеры разделов указаны без учета spare.
```

```
RawSizeMode = false
```

```
0x00000000,0x1950000,UBOOT
```

```
0x01950000,0x1734000,UBOOTENV
```

```
0x03084000,0x1B4A4000,SYSTEM.UBI
```

```
;0x00000000 - начальный адрес раздела, можно указывать как с учётом spare, так и без
```

```
;0x1950000 - размер раздела, указываем только с учётом spare.
```

```
;UBOOT - имя раздела
```

Примеры работы с разделами можно посмотреть в [приложении](#).

8 GZIP.

Одной из приятных особенностей UFPI, это работа с [GZIP](#) (открытый распространенный формат архива) форматом на лету. Это значит можно читать и писать напрямую из архива. Тем самым экономим место на диске и не тратим время на распаковку дампов.

Для чтения в архив, нужно выбрать файл для сохранения с расширением .gzip.

Для записи из архива .gzip, не требуется его распаковывать, просто выбрать архив и нажать «Запись». Программатор сам определит формат, и запишет его содержимое.

Упаковывать и распаковывать можно не только дампы, но и любые файлы.

В настройках можно найти степень сжатия. Может влиять на скорость работы на слабых компьютерах. По умолчанию установлен уровень 7, как наиболее оптимальный.

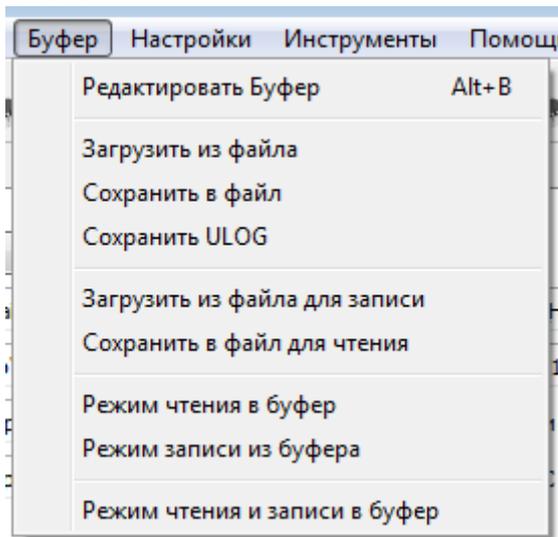
9 Работа с буфером.



Буфер позволяет быстро оценить содержимое, а так же не сохраняя дампы в файл можно считать, отредактировать и записать обратно во флеш.

Работа с буфером может оказаться стабильнее на высокой скорости, чем напрямую с файлом на HDD. Буфер это выделенное в оперативной памяти пространство, отсюда и ограничения в его размере. Для работы с [LOGGER](#) это самый быстрый вариант (после захвата сигнала можно либо сконвертировать в формат .sg, либо сохранить в родном бинарном формате UFPI .ulog).

Редактор буфера открывается по кнопке на панели инструментов, либо через меню "Редактировать буфер"



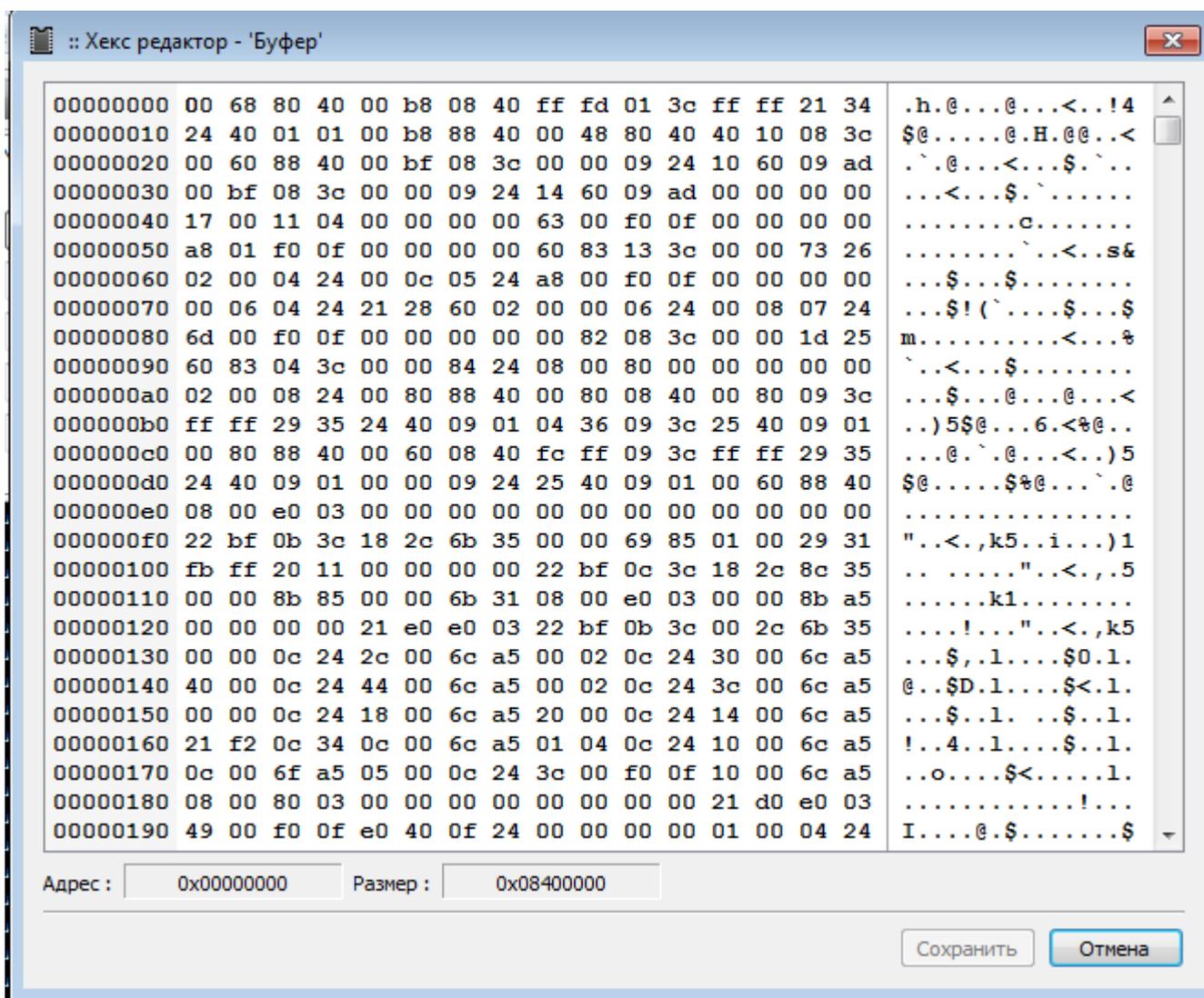
Флаги "Режим чтения в буфер" и "Режим записи из буфера" можно ставить и снимать либо по отдельности, либо одновременно пунктом "Режим чтения и записи в буфер"

При этом соответствующее активированному пункту поле "Чтение в" или "Запись из" становятся не активны.

Чтение в :

Запись из :

Сам редактор позволяет просмотреть, отредактировать и сохранить. Стандартные горячие клавиши Ctrl+C, Ctrl+V и другие так же работают.

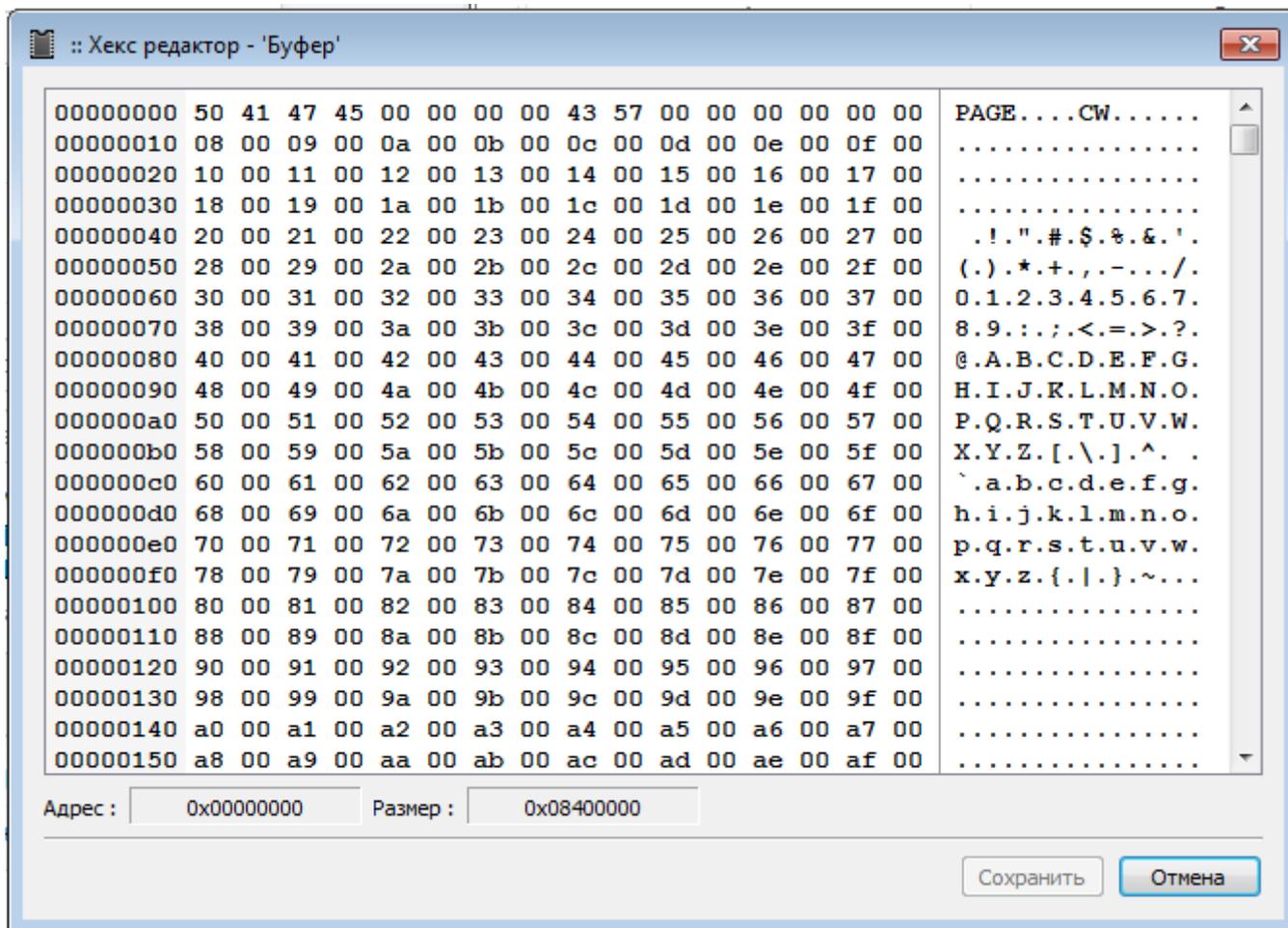


Размер буфера ограничен и не может быть более 1,5Гб.

10 Тестовый файл.

Тестовый файл позволит сгенерировать визуально понятную последовательность в дампе.

Для каждого модуля и каждой флеш, тестовый дамп генерируется индивидуально с учетом геометрии и размеров и прочих опций.



Визуально в редакторе видно разделение страниц, кодовых слов и даже sparse с заданной ECC.

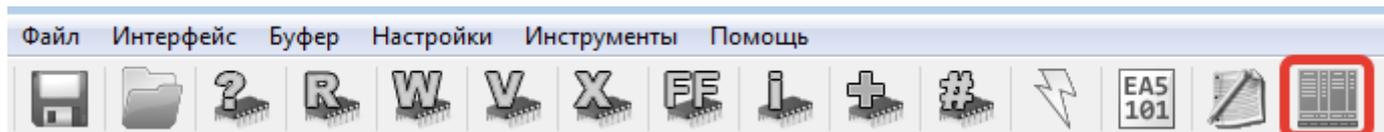
11 Файловый менеджер и монтирование ФС.

Особенности.

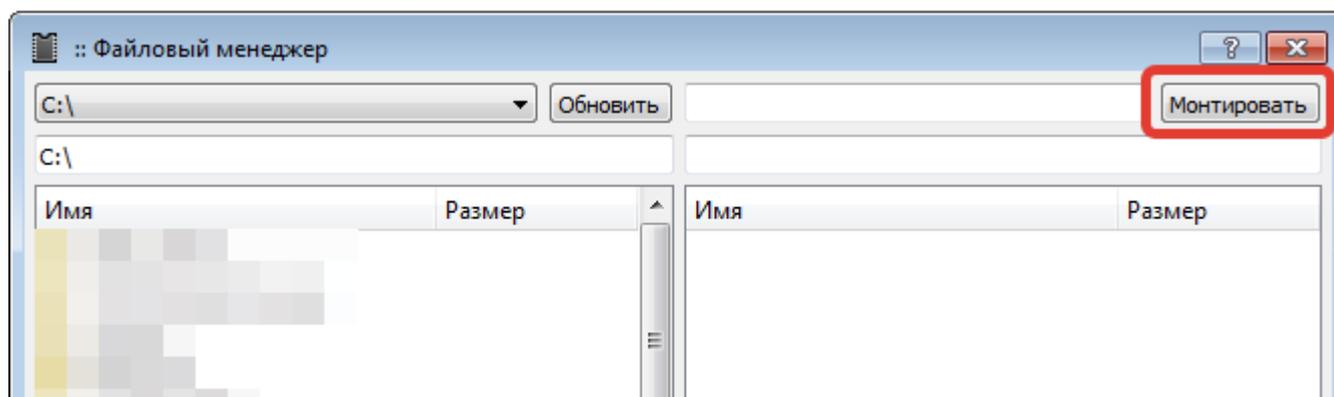
Файловый менеджер дает возможность монтировать и работать с файловой системой напрямую производя чтение и запись на лету самой файловой системой в дампе файла или непосредственно во флеш. Монтировать можно как раздел в дампе, так и отдельный файл раздела.

Запуск.

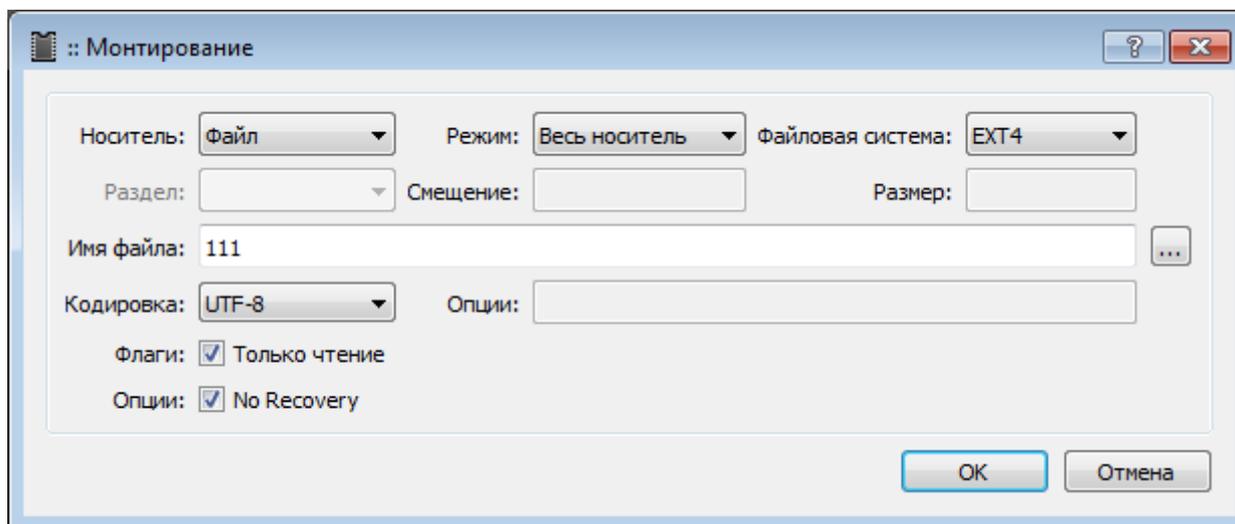
При подключенном программаторе в панели инструментов появляется кнопка запуска файлового менеджера.



При запуске открывается файловый менеджер.



Для монтирования раздела нужно нажать кнопку «Монтировать».



Опции монтирования:

Носитель — чип/файл;

Режим — весь носитель(если весь файл или носитель является одним разделом) / часть носителя (вписываем вручную смещение и размер раздела в дампе)/ раздел(выбрав раздел из списка);

Файловая система — выбор необходимой ФС;

Раздел — выбор раздела среди списка определившихся при анализе во время чтения ID флеш или

дампа;

Смещение, размер — для установки адресов вручную при режиме «часть носителя»;

Имя файла — необходимо указать, если работаем с файлом;

Работа в файловом менеджере.

Описана в примерах в [приложении](#).

Примеры. (рекомендуется ознакомиться)

[Монтирования](#).

[Работа в файловом менеджере](#).

Заметки.

Если не примонтировался раздел, то аналогично перепроверяем все по пунктам, перезапускаем софт, убеждаемся в корректности или совместимости дампа или берем другой.

В окне «Монтирование» после выставки всех полей, перед нажатием ОК перепроверяйте все поля заново. При неправильных последовательностях выставки поля могут сами меняться.

Поле Режим: «Часть носителя» нужно для ручного указания раздела. Если разделы не найдены, то можно в ручную ввести, например <https://mslw.com/bb/showthread.php?tid=2...2#pid35582>

Поле Режим: «Весь носитель» нужно для монтирования одного извлеченного раздела (например вычитывали один раздел через "режим: разделы" , его можно примонтировать), путь к которому нужно будет указать в поле ниже.

Флаг «Только чтение» думаю понятна без объяснений. Это предохранитель от необдуманных действий. Некоторые ФС сами по себе являются «Только для чтения»

Флаг «No Recovery» отключение встроенного механизма восстановления структуры ФС.

Копировать можно только файлы. Никакого копирования папок нет. Папки можно создавать или удалять, если не пустые.

Переходить по другим папкам в файловом менеджере нельзя во время копирования.

Если требуется переподключать бокс, то и программу нужно перезапустить. Иначе не будет монтироваться. Т.е. если просто бокс реконнект сделал после сокета и пр. - все будет работать, если с него пропало питание - то перезапускаем софт.

В Linux и MacOS нет монтирования. Только для Windows.

Добавлены возможности проверки файлов по хеш (например ключей, а так же их экспорт/импорт).

Добавлены возможность восстановления испорченных ключей в платформах QV.

12 Контейнеры.

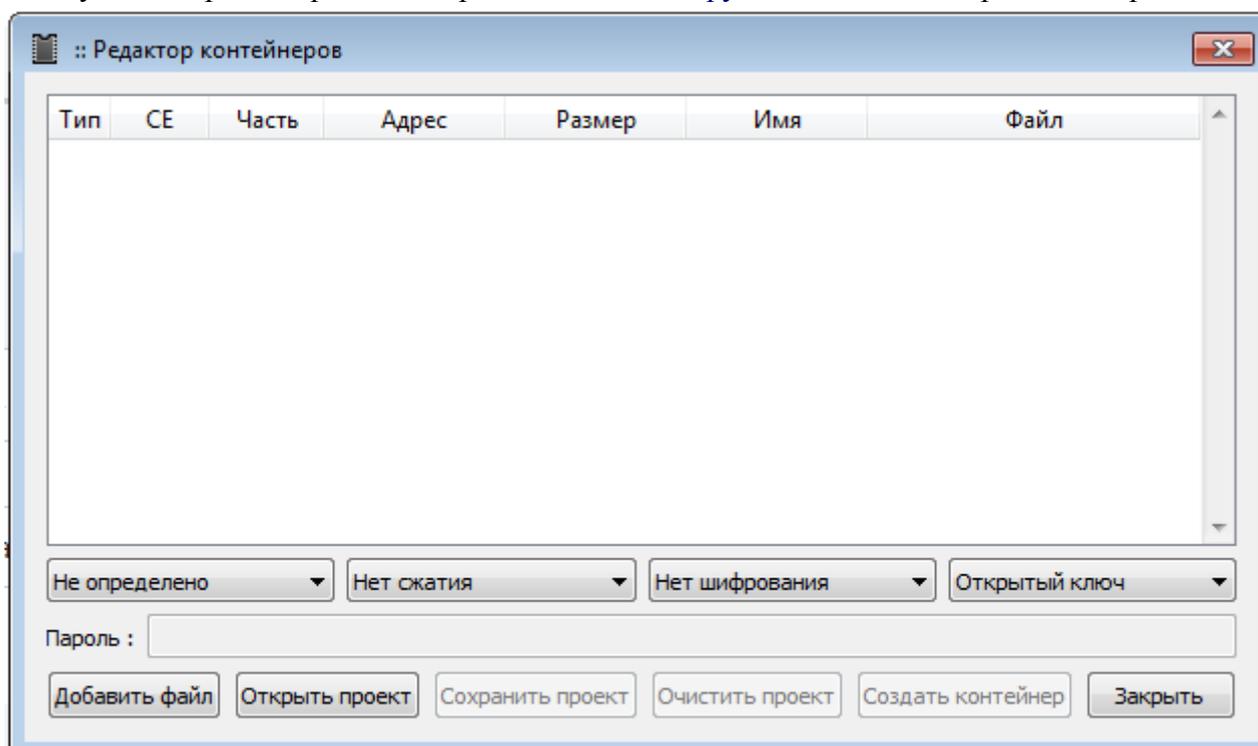
Особенности.

Основное назначение контейнеров – это хранение нескольких частей прошивки в одном файле и возможность прошить все эти куски по разным адресам из одного файла. Так же есть возможность добавить изображение, которое будет содержать необходимую информацию или напоминание и показываться при открытии контейнера. При желании данные в контейнере могут быть сжаты и зашифрованы. Если в качестве файла для записи выбран контейнер, то все операции записи/стирания/верификации будут производиться по адресам и размерам бинарных частей, которые содержатся в контейнере. По смыслу это почти противоположная операция от «[Бэкап платформ](#)».

При создании контейнера для NAND, адресация указывается без учёта spare.

Редактор контейнеров.

Запускается редактор контейнеров в меню «[Инструменты](#) — Редактор контейнеров».



В выпадающих меню устанавливается:

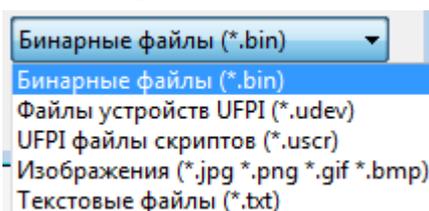
Тип микросхемы. Если NAND или eMMC, то нужно выбрать соответствующий пункт.

[Устанавливается уровень сжатия.](#)

Шифрование и пароль ([по аналогии со скриптами](#)).

Кнопки:

Добавить файл. Это может быть любая часть дампа, либо картинка и прочие.



Открыть/Сохранить/Очистить проект. Кнопки управления текущим проектом.

Создать контейнер. Формирует файл контейнера с выбранными опциями.

Открытие контейнера для записи.

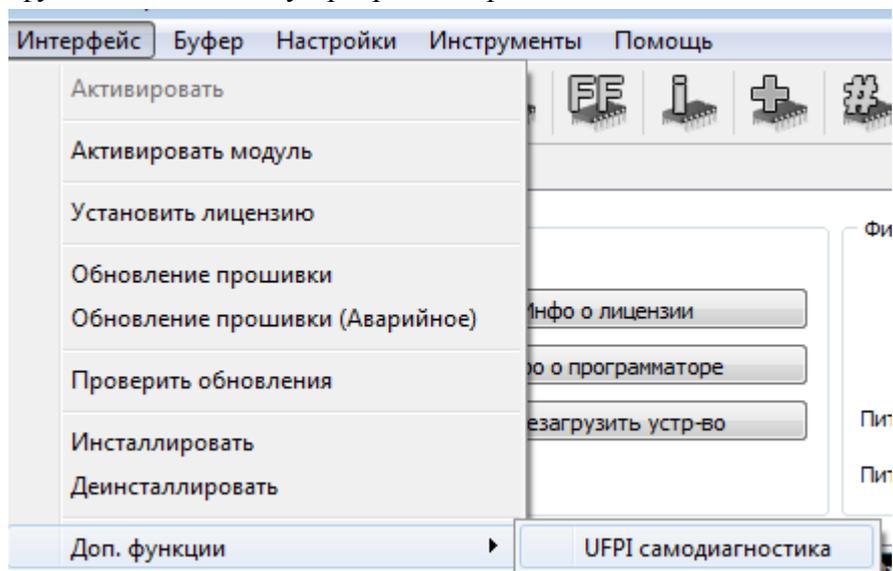
Открыть/закрыть контейнер можно горячими клавишами либо через меню «Файл — Открыть контейнер» и «Файл — Закрыть контейнер».

Пример работы.

[Сборка контейнера и пример работы описаны к приложению.](#)

13 Самодиагностика программатора.

Предназначена в основном для выявления дефектов сборки программатора - обрывов, коротких замыканий, некорректной работы датчиков и т.д. Была сделана для внутреннего пользования, но т.к. иногда есть необходимость в проверке не только при производстве, то была вынесена в отдельную функцию в оболочку программатора.



Как работает:

Для полного теста нужно сделать небольшой сокет с 4-мя резисторами - 910K на ID1, 1K на ID2, 75 Ом 0.5 Вт между землей в VCC, 120 Ом 0.5 Вт на VUSB. После включения такого сокета загорится желтый светодиод. Для запуска нажимаем "Интерфейс, доп ф-ции, UFPI самодиагностика". Если что-то не так, то в лог будет выведен список ошибок. Сначала тестируются сигнальные линии процессора на плате, генерация частот, датчики напряжений, потом тестируются линии буферов. Без сокета будут протестированы только внутренние линии.

14 Работа со скриптами.

По написанию скриптов есть отдельное [руководство](#), которое пополняется актуальными функциями.

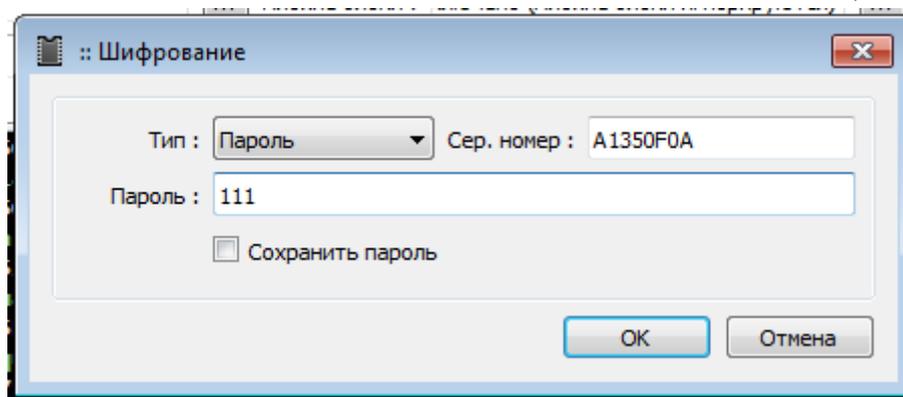
Скрипты бывают обычные, которые можно открыть и отредактировать как текстовый документ свободно, а так же без ограничений запускать, так и зашифрованные. Расширение файлов одинаковое .uscr.

Шифровать скрипты можно по

открытому ключу - просто чтобы нельзя было подглядеть содержимое, но использование не ограничивает.

паролю - для запуска требуется пароль.

серийному номеру бокса - идет привязка и запустить скрипт сможет лишь владелец указанного бокса (стоит обратить внимание, что серийный номер в логе указан в HEX виде, соответственно как принято в UFPI ввод HEX значения должен выглядеть как 0xA1350F0A).



Однако нужно знать, что зашифрованные скрипты нельзя подключать (`#include`) в другом скрипте.

Написание скриптов.

Скрипты пишутся в любом удобном текстовом редакторе (например с подсветкой синтаксиса Notepad++).

Для удобства отладки скрипт можно [привязать к горячей клавише в настройках](#).

Вся отладка при написании скриптов проходит путем запуска и проверки в окне лога. Например описывается место и тип ошибки:

```
Line 7, error 'No matching symbol 'setPowerMode''
```

```
Line 2, error 'Unexpected token '{''
```

Для отладки и контроля удобно пользоваться выводом значений функцией `print()`;

14.1 Скрипты.

Немало возможностей программатора осуществляется с помощью отдельных приложений, так называемых скриптов. Пишут их сами пользователи. Форум для обсуждения скриптов [здесь](#). В этом разделе приведены некоторые из них.

(Некоторые старые скрипты не обновлялись после изменений или добавления функций и могут не запускаться на последних версиях ПО, требуют соответствующих изменений).

[Горячие клавиши](#). Можно привязать вызов каждого скрипта к определённой кнопке.

OneNAND

[Скрипт для поиска разделов в дампе и создания файла .udev](#)

NAND

[Скрипт для быстрого определения попадает BAD блок установленной в программатор NAND на блок с информацией в дампе или нет](#)

[Скрипт для определения разделов в дампе NAND и создания UDEV файла](#). Что такое UDEV файл и для чего он нужен, читаем [здесь](#)

[Мини редактор работы с разделами \(чтение/вставка разделов из/в дампа с поддержкой UDEV файлов и размеров разделов исходя из названия + ручной режим ввода данных и легкий режим работы\)](#). Можно работать с одним или несколькими дампами, деля их на разделы и тасуя эти разделы между ними.

[Деление дампа NAND, на main и spare, а также обратное действие - собирание дампа из main и spare.](#)

[Скрипт создания файла дампа, в котором произведен классический пропуск BAD блока](#) Этот принцип обхода встречается в практике очень редко, не спешите его применять пока не поймёте - как именно работает обход бб для Вашего устройства.

[Скрипт определения и выставления оптимального параметра ReadRetry](#)

eMMC

[Скрипт для определения разделов в дампе eMMC](#). MBR/EBR, GPT, MSTAR ANDROID, MTK LG, ANDROID TV SONY

[Разделение дампа на разделы по данным из файла UDEV](#)

[Редактирование конфигурационных констант в дампах SMART TV ANDROID на базе CPU MSTAR](#). Нужен, например, для того, чтобы активировать uart. Так же, с помощью этого скрипта удалось открыть uart на шасси Vestel 17MB95

Для работы с разделами дампа eMMC, при наличии файла UDEV можно использовать скрипт, который мы использовали для работы с дампами NAND

[Мини редактор работы с разделами \(чтение/вставка разделов из/в дампа с поддержкой UDEV файлов и размеров разделов исходя из названия + ручной режим ввода данных и легкий режим работы\)](#)

SPI

[Скрипты по работе с OTP областью](#)

Super I/O

[Скрипт по работе с мультиконтроллерами KB9010, KB9012, KB9022](#)

Weltrend

[WT802, WT805, WT806, WT807](#)

Touch Sensor

[CT1C08x](#)

15 FAQ

Тут будут собраны и сгруппированы заметки, которые могут быть полезны для быстрого ознакомления работы с форумом, микросхемами и программатором. В [теме сопровождения руководства](#) можно озвучивать новые вопросы, которые стоит включить в руководство и FAQ.

15.1 По работе с форумом.

Как попасть в группу UFPI после покупки программатора.

Нужно просто прислать на email свой ник на форуме с просьбой добавить в группу UFPI. Письмо должно быть отправлено с того же самого почтового адреса, который был использован при покупке.

Общие советы и рекомендации.

Это технический форум. Вы должны понимать, что в случае если вы напишете сообщение в стиле "не могу прошить микросхему SO8" или "памагите прошить нанд из самсунга", нормального ответа на свой вопрос вы не получите. Получите, в лучшем случае, подколки и отношение к вам как к неграмотному новичку. Это русский форум, здесь почти нет толерантности.

Лично вам на этом форуме никто и ничего не должен. Вы должны понимать, что помогать вам здесь могут только на добровольной основе, тратя свое личное время. Не хамите людям которые хотят вам помочь, даже если вы считаете, что они просят что-то "не то". Старайтесь сами помогать новичкам, если знаете ответ на вопрос.

Не нужно давать советы или отвечать людям если вам "кажется", что вы знаете ответ на вопрос.

Не нужно давать советы когда их никто не просит. Это одна из худших черт русскоязычных форумов.

Запрос в теме на добавление микросхемы.

Первое, что нужно сделать, это воспользоваться поиском по форуму. Ввести название или ID. Если найдется, то все вопросы отпадут.

Второе. Вы должны разобраться в подключении и предоставить считанный ID и по возможности найти документацию на микросхему.

Если вы не можете понять что это за микросхема, куда подключать и тем более как читать ID, то не нужно спрашивать об этом в теме на запросы. Создайте отдельную тему, или спросите в теме [«Вопрос новичка»](#).

Сообщения в темах по запросу имеют вполне определенную структуру. Название — ID — Документация — Комментарий.

Тема. [Тестовые версии. Ошибки, обсуждение для всех модулей.](#)

В этой теме обсуждаются только текущие бэта версии. Если это ошибка, то формулируется сообщение в таком виде, чтобы можно было повторить с ваших слов и увидеть такой же результат. Описываются особенности подключения флеш, ОС и другие, которые могут иметь значение. Это не тема для обсуждения Вашего непонимания в работе с программатором. Если вы собираетесь писать в этой теме, значит Вы уверены в своих навыках, пользуетесь текущей бэта версией и готовы ответить на встречные вопросы.

Сообщение должно содержать следующую информацию:

Версия и разрядность ОС. Пример - "Windows 10 x64".

Версия ПО UFPI, это первая строчка при запуске программы. Пример - "UFPI Версия 2.0.4 Сборка 29.12.2020".

Модуль UFPI. Пример - "SPI Flash"

Тип сокета UFPI и подключение микросхемы. Пример - "SPI/I2C socket, панелька"

Производитель и название микросхемы. Пример - "Winbond W25Q80DLIG"

Краткое описание ошибки и как ее проверить/воспроизвести

Текстовый лог программы с ошибкой

Лог ОБЯЗАТЕЛЬНО должен быть убран под спойлер!

Лог должен содержать только минимальное количество текста с вашими действиями и ошибкой!

Тема. Тестовые версии программы.

В этой теме публикует только разработчик. В теме не нужно ничего писать или спрашивать. Тема для отслеживания развития функционала программатора.

15.2 Общие вопросы.

Где скачать ПО, инструкции и драйверы?

Можно найти на официальном сайте по [ссылке](#) выбрав соответствующую ОС версию. Также, если [требуется](#) по [ссылке](#) можно найти драйвера (не требуется для Windows 8 и выше).

Куда и как подключать?

Не обделяйте вниманием кабель USB и гнезда. Это должен быть хороший USB High-Speed кабель и использоваться разъемы на материнской плате (всякие USB хабы и передние панели ПК крайне не рекомендуются), желательно USB 3.0 в виду того, что он поддерживает большую нагрузку по току.

Как правильно подсоединять адаптеры во время работы?

Нет разницы, в какой последовательности подключать флеш-сокеты-программатор. Есть лишь явные меры предосторожности при работе при включенном питании Vcc, когда светится соответствующий светодиод на сокете. Напряжение Vcc подается только во время обмена с флеш (исключение составляет eMMC), все остальное время напряжение отключено и допускаются манипуляции.

Сколько микросхем в базе, поддерживается программатором?

Практически любые и практически все! Если рассматривать SPI, то их не всегда необходимо иметь в базе для работы, некоторые из них имеют специальную область с конфигурацией Serial Flash Discoverable Parameter (SFDP). У NAND имеется ONFI, у NOR имеется CFI, а у EMMC так же JESD. С помощью умения программатора и конфигураций, микросхемы флеш определяются автоматически. Так же можно всегда оперативно попросить на форуме помочь составить файл конфигурации (при наличии документации на флеш), или разработчика добавить в поддержку. При совсем тяжелых случаях и экзотических задачах, можно использовать скрипты.

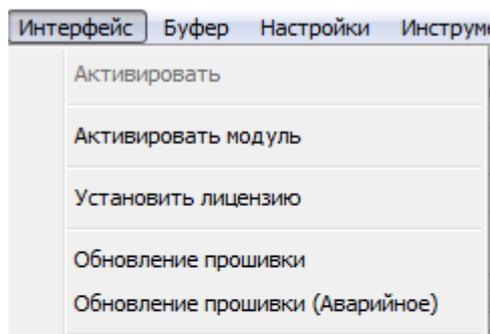
Почему нужно выбирать перед чтением файл для чтения и для записи отдельно файл для записи?

UFPI не создает никаких временных файлов при чтении или записи. Весь процесс происходит напрямую с выбранным файлом. Это позволяет прогнозировать объемы и свободное место на диске даже при больших объемах флеш, а так же не тратить время на работу с временными файлами.

Как "откатиться" на старую версию после обновления? Или чаще этот вопрос возникает после сообщения «UFPI Устаревшая версия для устр-ва с прошивкой vX.X.X. Пожалуйста используйте последнюю версию.»

Запустить нужную версию на которую хотите "откатиться". Далее нажать и удерживать кнопку на отключенном программаторе. Подключить программатор к USB. После того как программатор перейдет в режим обновления (начнут одновременно мигать все три светодиода желтым цветом), отпустить кнопку. [Далее в меню "Интерфейс" нажать "Обновление прошивки \(Аварийное\)"](#), после чего программатор будет прошит прошивкой из текущей версии и станет возможным его использования с этой версией. *Важно! Не рекомендуется злоупотреблять данной возможностью, т.к. любая флэш память имеет конечный ресурс для стирания и записи.*

Обычно откат версий работает в пределах бэта версий. Нельзя с версии 2.0.7 откатиться на версию 2.0.5.



Очередь задач.

UFPI умеет выстраивать очередь задач, выполняя потом их по очереди. Например хотим мы считать-стереть-записать-верифицировать флеш большого объема. UFPI хоть и самый быстрый, но ждать порой даже 10 минут окончания операции не нужно. Выбрав файл и запустив чтение сразу можно нажать кнопку стирания, далее проверку на чистоту, выбрать файл для записи и нажать "запись"...и программа сама по очереди выполнит все эти действия в указанном порядке без остановок.

Ввод HEX/DEC значений.

В программе UFPI нужно различать ввод шестнадцатеричных (0x100) и десятичных (100) чисел.

Меню для работы с файлами.

[Двойной щелчок](#) левой кнопки мыши по полям "Чтение в" и "Запись из" выдаст очень полезное меню. В котором помимо истории можно задавать автоимена файлам, работать с ECC у NAND и задать "Режим одной папки". Содержимое меню отличается у разных модулей.

Проверка на свободное место и наличие существующих файлов.

В [настройках, в главной вкладке](#) есть флаги перед чтением проверять свободное место на диске и наличие одноименного файла перед перезаписью с выводом оповещения.

Где расположены логи работы программы?

[Логи](#) сохраняются в папке Logs рядом с файлом самой программы UFPI.exe. В папке логи разбиты по файлам с датами и названиями модулей. Так же в папке Logs содержатся папки NAND и eMMC в которых соответственно модулю сохраняются ONFI, JESD, eCSD, CSD, CID и др. каждой подключенной флеш. Ничего не потеряется!

Встроенный архиватор.

В программе имеется [встроенный архиватор](#), который поможет сэкономить место на диске и умеет работать на лету без необходимости распаковывать отдельно. Так же можно распаковывать не только дампы, но другие файлы сжатые gzip, например обновления UFPI.

Как сбросить настройки? Как переустановить программу.

В Windows никаких настроек и изменений в реестре или иных системных папках не производится. Поэтому для переустановки достаточно удалить папку с программой и заново скачать, распаковать в нужную папку.

Для сброса настроек можно удалить файл ufpi.ini (либо аналогично имени исполняемого файла, если он был переименован). И тогда при следующем запуске он создастся автоматически с настройками по умолчанию.

Внизу слева в окне «Настройки» есть кнопка «По умолчанию», которая выставит флаги по умолчанию на текущей вкладке.

Тест контактов при подключении.

Практически в каждом модуле присутствует в «Доп. функции» пункт «[Тест контактов](#)». Позволяет выявлять замкнутые между собой, или на VCC/GND контакты с указанием адреса.

Иногда при работе с внутрисхемным подключением можно наблюдать ложные срабатывания. В таких случаях можно отключить его. В остальных случаях он полезен и по умолчанию включен.

[Асинхронный режим.](#)

Позволяет при нормальных условиях получить существенный прирост в скорости. Однако в некоторых случаях его приходится отключать, например если нужно вычитать полуживую eMMC или NAND с повторами при ошибках.

–[Выделения в логе цветом.](#) (По умолчанию)

Белый — основной текст. Например название операции, имя файла...

Зеленый — выделение информации в общем тексте. Флаги запускаемой задачи, значения...

Желтый — предупреждение. Это могут быть битовые ошибки NAND, или несоответствующий размер файла, или другие некритичные предупреждения, на которые стоит обратить внимание.

Красный — ошибка. Ошибки критической значимости и результат в таком случае неверный.

Голубой — подсвеченная выделенная информация. Специфическая информация, например в NAND если выставлено «Адреса со spare», в таком случае подсвечиваются адреса.

[Режимы эмуляции.](#)

Во время подключения сокета в режиме эмуляции вы не увидите открывшихся вкладок модуля, при этом в диспетчере устройств появится новое соответствующее устройство. Если при подключении например сокета JTAG вы столкнулись с подобным, то просто отключите режим эмуляции в основной вкладке ПО выбрав «По умолчанию» и нажав кнопку «Фикс».

15.3 Принято называть:

Сокет.

Это плата, которая вставляется в программатор, имеет переключатели и резисторы ID для выставления нужного напряжения и модуля. На нее может вставляться дополнительно база с необходимой колодкой или уже содержаться колодка для микросхемы или имеет разъемы и пины для подключения произвольных переходников.

База сокета.

Вставляется в соответствующий ей сокет и предназначена для установки bga или tsop колодки, другой базы или флеш непосредственно.

Бокс.

Сам программатор UFPI.

Модуль .

Это если проще для понимания вкладка в ПО, например NAND/SPI/eMMC...которая активна при вставленном соответствующем сокете и наличии лицензии.

ВВ или ББ .

[Бэд блок, плохой блок или bad block](#). Неисправный блок в NAND флеш.

Page, Block, Spare.

Page (страница), из которых состоит Block (блок). Из них состоят практически все флеш. Их количество и размер составляет размер флеш. В NAND еще имеется служебная область в конце страницы, называемая Spare.

Геометрия (параметры/конфигурация) флеш.

Принято описывать параметры флеш в размерах Page, Block, а так же Sector, Spare, LUN, CE и другие. Совокупность параметров и называется «Геометрия флеш»

CRC (Контрольная сумма).

Это алгоритм, или значение, которое предназначено для проверки целостности данных.

15.4 Питание при работе.

Управление питанием.

Щелчок правой кнопки мыши по [кнопке с молнией](#) (переключатель режимов питания сокета) покажет меню управления напряжением, где можно выставить любой необходимый режим.

Заниженное напряжение USB.

Важно! Напряжение питания с нормальным кабелем, разъемом и ПК будет не ниже 5 вольт. При критически заниженном напряжении показания в [статусбаре](#) будут выделены красным шрифтом! При таких просадках напряжения, не исключены (особенно при работе внутрисхемно) сбои в работе.

Отключение защиты питания по току.

UFPI имеет два канала питания с защитой отключающей нагрузку по достижении 500мА. Порой при внутрисхемной работе бывает приходится переваливать через этот порог. На свой страх и риск можно выставить флаг "Игнор аварии" на нужном канале и нажать в подтверждение кнопку "Фикс"

15.5 [NAND](#).

Как работать с NAND имеющих несколько CE.

Если флеш определилась автоматом, есть в базе программатора и в логе отображается правильным полным размером (т.е. в логе так и написано «Количество чипов (CE) 2»), то читаться будет одним файлом. Если чип неизвестен и конфигурация добавляется вручную, то каждый CE нужно будет вычитывать по отдельности выбрав необходимый CE в меню. Т.е. в конфигурации нельзя выставить количество CE!

Как читать/писать микросхемы частично, или разделами.

Для этого есть [«Режим доступа»](#). В каждом модуле он может иметь свои особенности. В NAND нужно иметь в виду, что минимальная величина стирания - «Блок», а минимальная величина записи - «Страница». При чтении ID их можно посмотреть в логе для конкретной микросхемы.

Адресация.

В устройстве, работа с NAND происходит постранично. Spare не рассматривается как область данных. В обычном считанном дампе можно видеть, что он состоит из Page и из Spare. Поэтому адресация в дампе будет отличаться от адресации при работе в [«Режим доступа»](#), или чтении терминальных логов, или программатора. Для того, чтобы в логе видеть адресацию с учетом Spare, есть в меню «Доп.функции» параметр «Адреса со Spare».

Запись дампа с помощью анализа.

[Анализ дампа при чтении ID](#) позволяет автоматически сконфигурировать программатор для работы с необходимыми (если они известны) ECC, BB, BBT.

Однако нужно помнить, что дампы должны быть очищены [предварительно от BB](#) и соответствовать геометрии установленной NAND.

[–Не записывать пустые данные.](#)

Позволяет не только хорошо сэкономить время записи и является режимом по умолчанию. Но в

некоторых, крайне редких случаях для NAND с HWECC запись может оказаться неверной.

При чтении «Слишком много ошибок. Прервано!».

Можно встретить при некорректно подобранной ЕСС схеме или испорченном дампе. Чтобы считать такой дамп, нужно просто отключить ЕСС. Либо отключить «Асинхронный» режим.

15.6 [EMMC.](#)

Вычитывание неисправных EMMC.

Повторы чтения при ошибке eMMC позволяет вычитывать при сбоях. Может потребоваться при вычитывании некоторых неисправных флеш. Для этого дополнительно требуется отключить в [«Доп. функции»](#) опцию «Асинхронный», в «Режим чтения» выставить «Игнорировать ошибки чтения» и «Режим чтения : Автовыбор» (или даже одиночными блоками). В [настройках модуля](#) выставить количество повторов. Так же желательно уменьшить частоту и количество линий до 1bit.

Анализ при чтении ID (EMMC).

В [«Доп. функциях»](#) позволяет выбрать объект для проверки и применения таблицы разделов внутренними алгоритмами UFPI без применения скриптов. Можно выбрать файл, или флеш.

Стирание eMMC.

Согласно спецификации, контроллер этих флеш сам автоматически стирает в процессе перед записью блоки во флеш. Поэтому нет необходимости проводить процедуру стирания отдельно, тем более, часто флеш вообще не поддерживает такие команды. Но возможность такая присутствует и в разных микросхемах может поддерживаться либо команда стирания, либо можно просто записать нули, либо 0xFF. По кнопке «X» алгоритм стирания соответствует установленному [«Доп. функции»](#) - «Режим стирания». В eMMC Samsung до ревизии v5.0 так же поддерживается функция форматирования (Отдельный пункт в «Доп. функциях»), который стирает как данные, так и конфигурацию загрузки и прочее до заводского вида.

15.7 [SPI.](#)

[Питание.](#)

Особое внимание перед началом работы следует уделить питанию целевой флеш. Поскольку не всегда при чтении ID/SFDP возможно корректно различить 1,8/3,3V. Точные значения можно получить из документации к микросхеме или изучив оборудование на котором установлена флеш.

Для внутрисхемной работы с 3,3 вольтовыми флеш (например «прищепкой») часто более удачным бывает попытка работы при уровнях и напряжении 2,8 или даже 2,6V. При этом полезно установить флаг питания «Всегда включено» и несколько раз перечитать ID контролируя, что все работает стабильно.

15.8 [UART.](#)

Автоскорость.

Очень редкой и удобной является возможность автоматически определять baudrate (скорость потока).

Введенные символы или сообщения дублируются в лог.

При вводе с клавиатуры (при установленной галке «Слать данные с клавиатуры») или отправке

сообщения через поле «Текст/Хекс» можно видеть повторы типа: "hheellpp". Стоит помнить, что это не «Терминал», а ЛОГ. В нем отображено то, что отправляется и принимаемое эхо. По этой же причине в логе могут некорректно отображаться vi и подобные программы.

16 Приложение.

16.1 Программаторы, сокет, подбор и совместимость.

В пределах этого руководства ориентироваться проще по артикулам:

UFPI Красный Вернуться к...

UFPI Red Box - комплектация Расширенная.

Артикул 2007



Цена без н
Разм

1 + -

Добав



- 2006 [UFPI Silver](#)
- 2007 [UFPI Red](#)
- 2008 [UFPI Black](#)
- 2037 [UFPI PCB](#)
- 2036 [UFPI Pro PCB \(снят с продажи\)](#)
- 2041 [ICFRIEND 8-Bit Socket](#)
- 2040 [E-MATE X 8-Bit Socket](#)
- 2039 [E-MATE Pro 8-Bit Socket](#)
- 2038 [EASY JTAG eMMC 8-bit Socket](#)
- 2063 [UFPI to Easy JTAG eMMC Socket](#)
- 2018 [NAND TSOP48 Wells-CTI](#)
- 2019 [NAND TSOP48 Pinboard](#)
- 2017 [NAND DIP48 ARIES](#)
- 2020 [NAND Base SMT](#)
- 2021 [NAND BGA63 KZT SMT](#)
- 2009 [NAND BGA63 KZT SMT Base](#)
- 2010 [NAND BGA63 JRS SMT Base](#)
- 2056 [TSOP48 Wells SMT](#)
- 2058 [TSOP48 Pinboard SMT](#)
- 2061 [TSOP56 Pinboard SMT](#)
- 2057 [ANDK TSOP48-0.5](#)
- 2062 [ANDK TSOP56-0.5](#)
- 2031 [OneNAND Base SMT](#)
- 2030 [OneNAND BGA63 KZT SMT](#)
- 2011 [OneNAND BGA63 KZT SMT Base](#)

- 2059 [OneNAND BGA63 JRS SMT Base](#)
- 2023 [NOR Base SMT](#)
- 2024 [NOR BGA64-1.0 KZT SMT](#)
- 2012 [NOR BGA64-1.0 JRS SMT Base](#)
- 2026 [TSOP56 ANDK SMT Universal](#)
- 2025 [TSOP56 ANDK SMT J3X Series \(NOR\)](#)
- 2027 [TSOP56 ANDK SMT P3X Series \(NOR\)](#)
- 2042 [SD Card Socket](#)
- 2046 [SPI/I2C/1W DIP24 Socket](#)
- 2060 [SPI/I2C/1W/UART DIP24W Aries Socket](#)
- 2044 [UART/GPIO Socket](#)
- 2043 [JTAG/BDM Socket](#)
- 2045 [LOGGER Socket](#)
- 2034 [UFPI Breadboard](#)
- 2013 [Frame 10.5×13.5 BGA63 KZT](#)
- 2014 [Frame 10×13 BGA63 KZT](#)
- 2015 [Frame 11×13 BGA63 KZT](#)
- 2016 [Frame 9×9 BGA64 KZT](#)
- 2035 [eMMC RT809H 8-Bit PCB](#)
- 2032 [OneNAND ENTT PCB](#)
- 2028 [OneNAND ENTT BGA JRS PCB](#)

Далее будем ориентироваться по артикулам и перечисляться варианты. Перечислю варианты некоторых комплектов, на которые стоит обратить внимание для ориентирования.

16.1.1 Комплекты NAND.

Варианты наиболее распространенного TSOP48.

Вариант наиболее качественного и долговечного адаптера — 2018.

Вариант наиболее бюджетный — 2019 + 2057.

Вариант более новый и универсальный — 2020 + 2058 + 2057.

Вариант предыдущий с колодкой Wells — 2020 + 2056.

Вариант максимально универсальный с качественной колодкой для использования переходников с распиновкой 1:1 с других программаторов (можно использовать не только для tsop48, если распиновка соответствует) — 2017.

Варианты NAND BGA63.

Вариант с наиболее качественной колодкой KZT – 2020 + 2021.

Вариант с установкой своей колодки (например если имеется 2031 + 2030) KZT – 2020 + 2009.

Вариант с установкой своей колодки JRS – 2020 + 2010.

Варианты OneNAND BGA63.

Вариант с наиболее качественной колодка KZT – 2031 + 2030.

Вариант с установкой своей колодки (например если имеется 2020 + 2021) KZT – 2031 + 2011.

Вариант с установкой своей колодки JRS – 2031 + 2059.

Вариант с использованием совмещенного с ENTTv2 и установкой своей колодки JRS – 2032 + 2028.

16.1.2 Комплекты SD/eMMC.

Сокет 2042 в первую очередь для работы с SD картами и в меньшей степени предназначен для eMMC и тем более внутрисхемного подключения (хотя не запрещает).

Сокеты 2038, 2039, 2040, 2041 готовы, проверены и предназначены как для работы с соответствующими колодками, так и внутрисхемному подключению. Стоит очень внимательно относиться к выбору соответствующего своей колодке сокета, поскольку все они имеют отличия в распиновке.

Сокет 2035 является пустой платой, для соответствующей из названия колодке. Элементы и разъемы не входят в комплект. Покупаются и устанавливаются пользователем согласно типовым схемам адаптеров eMMC.

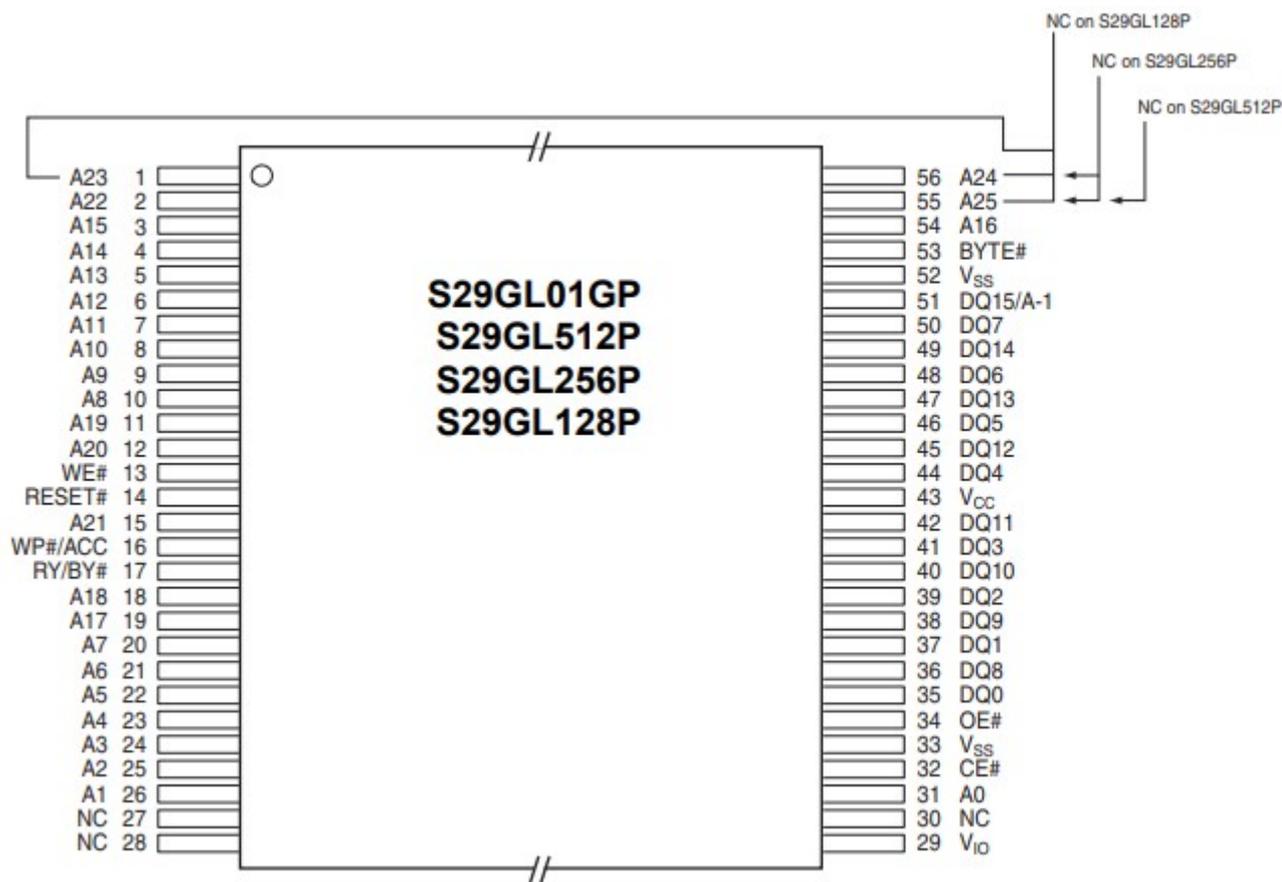
Сокет 2063 можно использовать в качестве универсального переходника UFPI - Easy JTAG. Подходит вероятно для любых вертолетов eMMC.

16.1.3 Комплекты NOR.

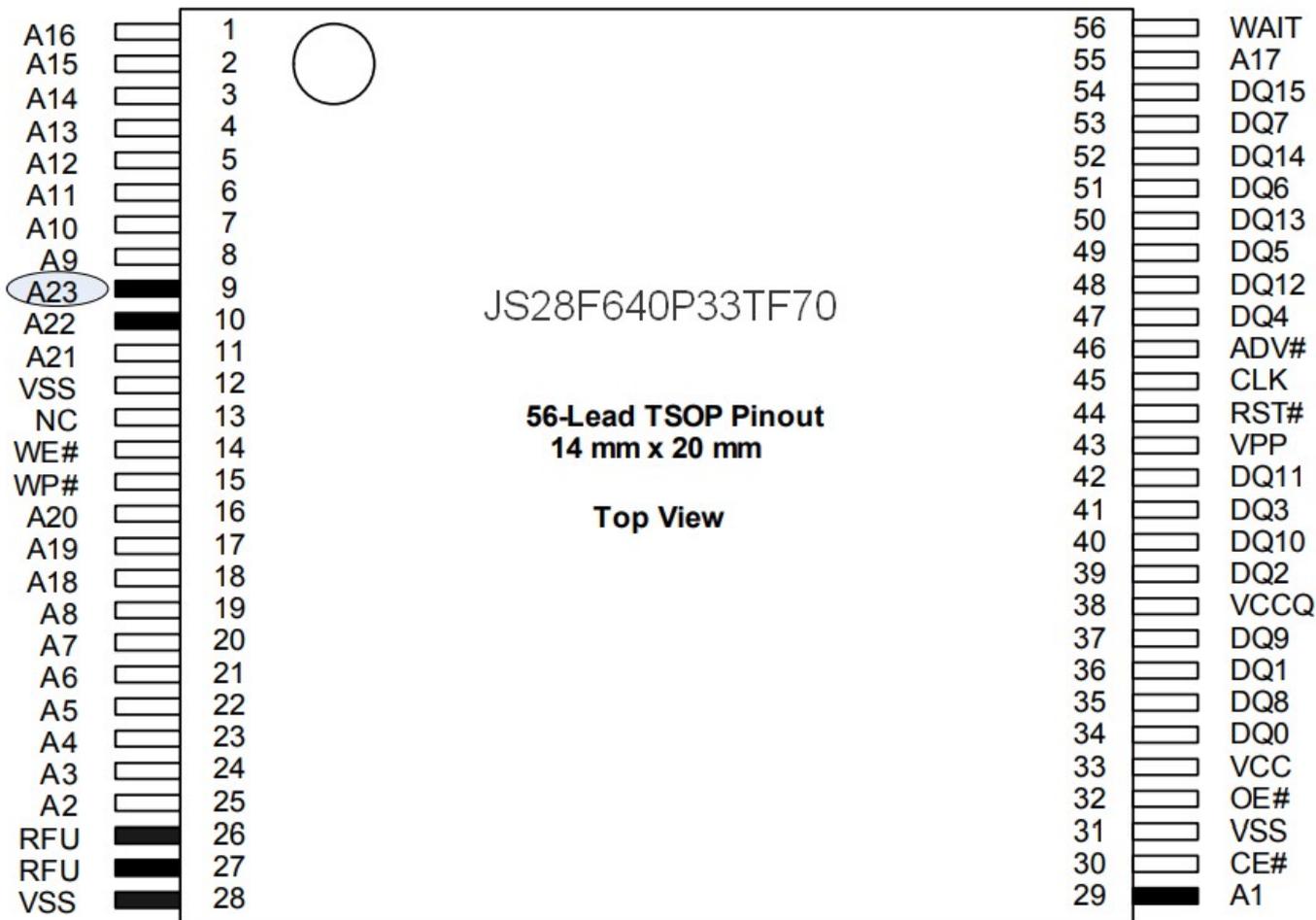
Вариант комплекта для TSOP48 – 2023 + 2058 + 2057.

Вариант комплекта для TSOP48 с колодкой WELLS - 2023 + 2056.

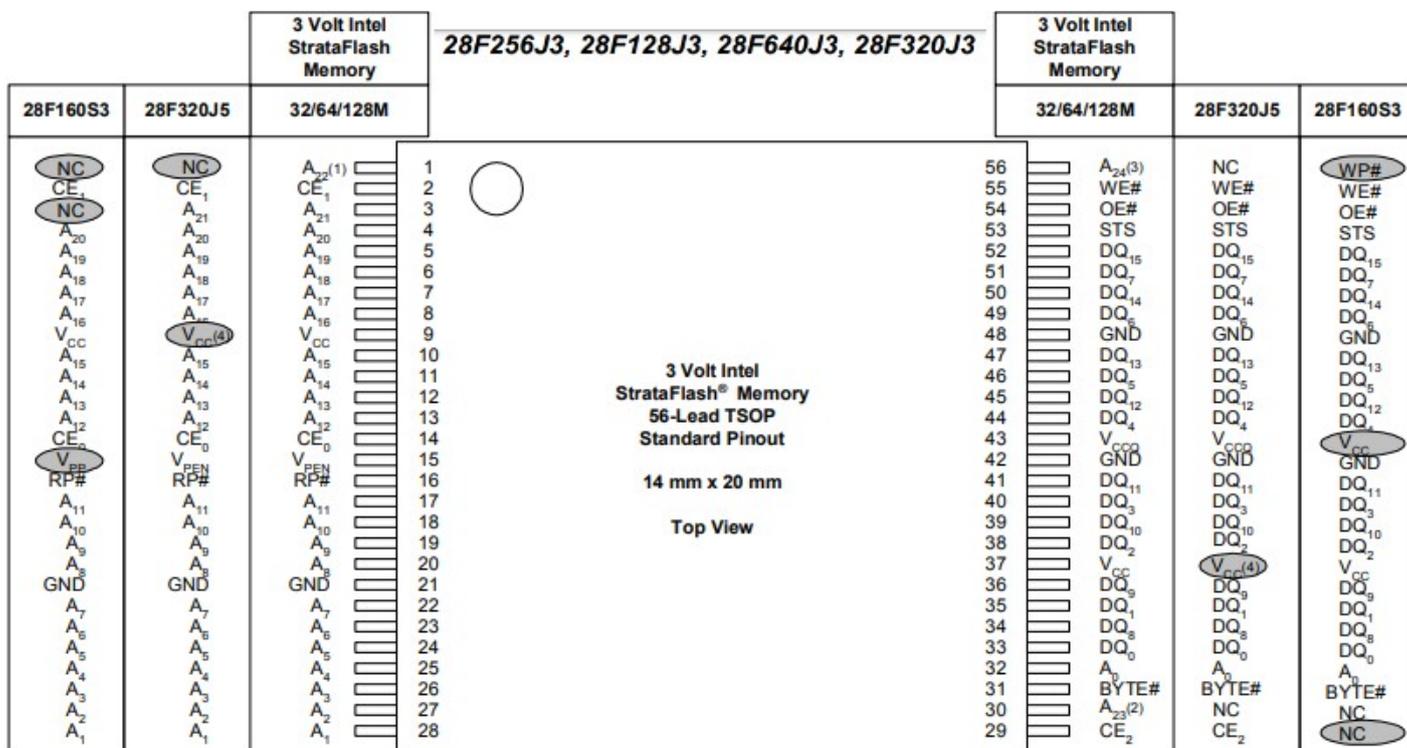
Вариант комплекта для TSOP56 (стандартная распиновка) – 2023 + 2026 (или 2061) + 2062.



Вариант комплекта для TSOP56 (P3X распиновка) – 2023 + 2027 + 2062.



Вариант комплекта для TSOP56 (J3X распиновка) – 2023 + 2025 + 2062.



Вариант комплекта для BGA64 с качественной колодкой KZT – 2023 + 2024.

Вариант комплекта для BGA64 с установкой своей колодки JRS– 2023 + 2012.

16.1.4 Остальные комплекты

SPI/I2C/1W/UART

Сокет 2046 наиболее распространен и привычен, однако на сегодня более новым уже будет 2060. В 2060 добавлена возможность включать UART и выведены для этого дополнительно площадки для подключения.

Положение переключателей на сокете соответствует режимам:

Режим	Vcc
ON ON ON – SPI Flash	ON ON ON – 3V3
OFF ON ON – I2C EEPROM	OFF ON ON – 1V8
ON OFF ON – SPI EEPROM	OFF OFF OFF – 5V0
OFF OFF ON – mWire EEPROM	ON OFF ON – 2V6
OFF OFF OFF – 1-Wire MODE	
ON OFF OFF – UART	
ON ON OFF – SNAND (SPI NAND)	

Примечание.

- Стоит отметить, что колодки вместо 2057 и 2062 можно использовать любые свои подходящие. Они универсальны.
- На таких сокетах, как 2043, 2044, присутствует разъем для подключения собственных шнуров, обжатых в RJ45. Очень удобно для быстрой замены и организации большого набора собственных переходников.
- Пустые платы сокетов (PCB) не комплектуются ни разъемами, ни элементами.
- Для мастера ТВ обычно в порядке убывания важны:
 - SPI/I2C/UART.
 - NAND TSOP48.
 - EMMC.
 - OneNAND BGA63.
- Остальное, такое как NAND BGA63, NOR уже крайне мало вероятно.

16.1.5 Советы и полезная информация для выбора вашего комплекта адаптеров к программатору.

Адаптеры универсальные.

Если вы уже решили купить программатор, то SPI/I2C/1W DIP24 (арт. 2060 или 2046) сокет это то, что принято называть "мастхэв" адаптер. При помощи данного универсального адаптера программатор может превращаться в инструмент для работы с различными типами памяти, протоколами и напряжениями. Выбрав нужный режим работы возможна работа с SPI NOR, SPI NAND, SPI EEPROM, I2C EEPROM, Microwire (3-Wire) EEPROM и 1-Wire устройствами.

Адаптеры для работы с eMMC.

При выборе нужно учитывать и понимать следующие моменты. В магазине не продаются BGA комплекты для работы с eMMC, они приобретаются отдельно на Алиэкспресс и подобных площадках.

E-MATE X EMMC BGA 13 IN 1 E-MATE PRO BOX EMMC TOOL



EASY JTAG Socket



ICFRIEND High Speed E-MATE



Какой из китайских BGA комплектов выбрать и купить зависит только от ваших предпочтений, финансов и того, что там продается в данный момент. В магазине UFPI продаются только адаптеры для этих "стандартных" китайских комплектов. Т.е. вопрос выбора адаптера для работы с BGA eMMC сводится лишь к тому какой набора BGA у вас будет или уже есть.

Для программатора RT809 есть печатная плата, которая позволит самостоятельно собрать eMMC сокет для UFPI и использовать имеющиеся BGA адаптеры.

Если имеется, или планируется купить, например, EASY JTAG EMMC BGA комплект - значит покупать UFPI EASY JTAG eMMC сокет (2038) и т.д. Если нужно работать только в ISP режиме, то нет разницы какой eMMC адаптер вам нужен, подойдут все. При желании можно собрать адаптер самостоятельно, схемы адаптеров в [приложении](#) и на [сайте](#).

Адаптеры для работы с NAND.

При выборе адаптеров для NAND нужно определиться для себя планируется ли работать с OneNAND, NOR в TSOP48.

Если нужно работать только с NAND в TSOP48, то выбор довольно простой. Есть деньги - купить NAND TSOP48 Wells-CTI (2018). Если нужно сэкономить - покупать NAND TSOP48 Pinboard (2019) с ANDK TSOP48-0.5 (2057) или NAND DIP48 ARIES (2017) если есть подходящие NAND адаптеры от других программаторов.

Если планируется работа с BGA и/или не только с NAND, но и с NOR, то лучшим выбором будет NAND Base SMT адаптер с необходимыми для вас панельками из SMT серии (TSOP48 Wells SMT (2056), TSOP48 Pinboard SMT (2058), NAND BGA63 JRS SMT Base (2059)). Преимуществом такого выбора является то, что верхние сменные адаптеры, например, TSOP48 Wells SMT (2056) можно использовать со всеми базами-адаптерами серии SMT. Т.е. TSOP48 Wells SMT (2056) может быть использована как с NAND, так и с NOR и OneNAND базами SMT.

Адаптеры для работы с OneNAND.

Важно понимать, что несмотря на то, что почти все микросхемы OneNAND в корпусах BGA63 схожих по размеру и виду с NAND, этот тип памяти имеет совершенно иную конфигурацию сигналов и питания, которые не совпадают с NAND. Поэтому для работы с OneNAND обязательно нужен базовый адаптер OneNAND Base SMT. Если уже имеется в наличии BGA63 сокет от JRS, то можно будет работать с использованием NAND BGA63 JRS SMT Base (2010). Есть деньги или хочется качества получше - для этого в магазин есть OneNAND BGA63 KZT SMT (2030).

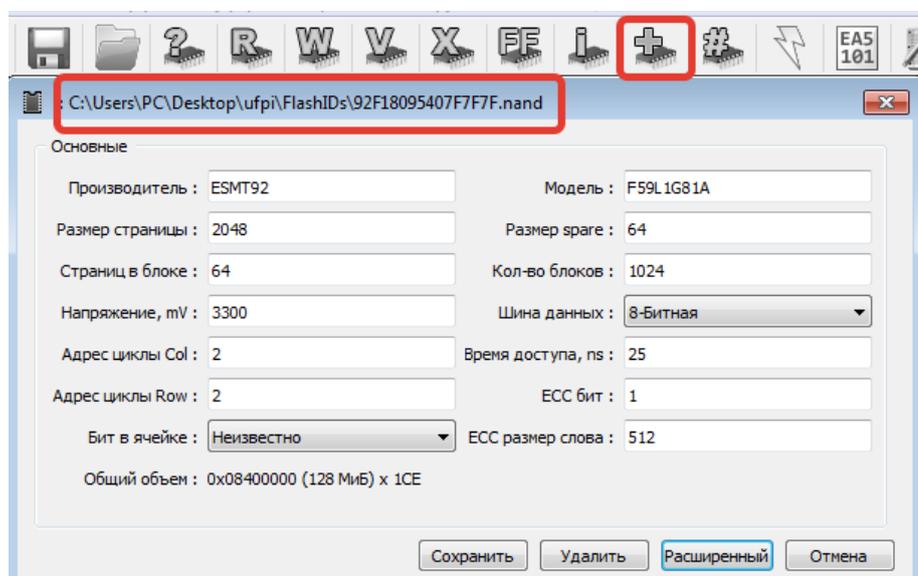
Адаптеры для работы с NOR.

Для работы с NOR обязательно нужен базовый адаптер NOR Base SMT. Для работы с TSOP48 подойдут TSOP48 Wells SMT (2056) или TSOP48 Pinboard SMT (2058). Для TSOP56 нужен TSOP56 Pinboard SMT (2061). Вышеуказанные адаптеры предназначены для работы с большинством NOR микросхем имеющих "стандартную" распиновку сигналов. Плата TSOP56 ANDK SMT J3X Series (2025) предназначена для работы с микросхемами типа Intel J3x. А плата TSOP56 ANDK SMT P3X (2027) Series предназначена для работы с микросхемами с синхронными NOR P3x серий. BGA адаптер NOR BGA64-1.0 KZT SMT (2024) предназначен для работы с микросхемами NOR в BGA исполнении с шагом 1mm. Если уже имеется в наличии BGA64 сокет от JRS, то с ним можно использовать NOR BGA64-1.0 JRS SMT Base (2012).

16.2 Примеры.

16.2.1 Приложение. Составление конфигурационного файла NAND Flash.

Если флеш не определяется, или определяется не верно, по кнопке **«+»** открывается форма конфигуратора, куда нужно вписать основные для работы параметры микросхемы.

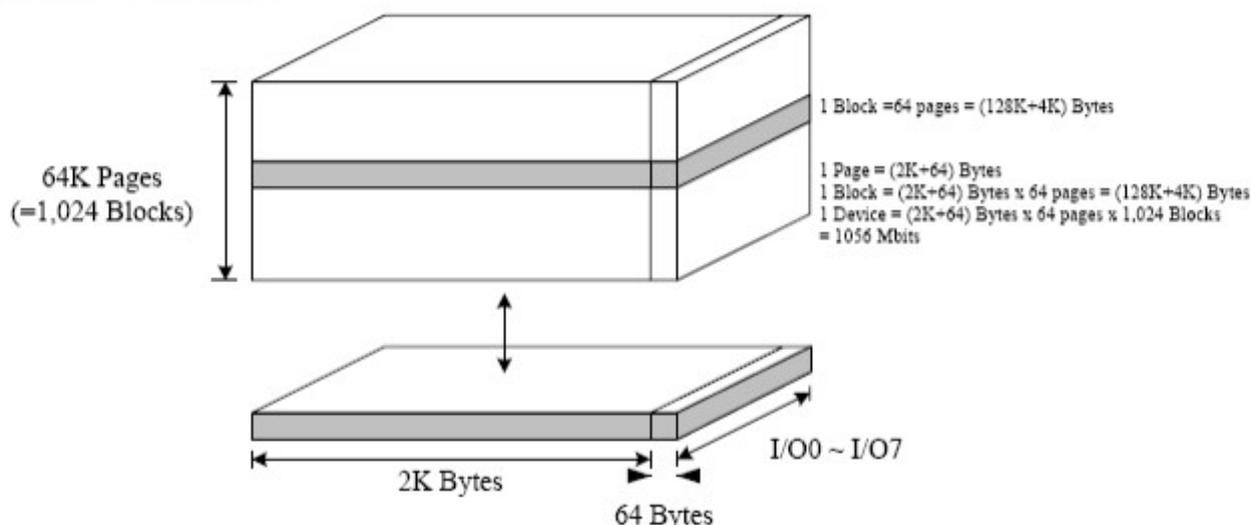


Основным достоверным источником официальная документация от производителя, которая обычно свободно распространяется в интернет.

Рассмотрим флеш ESMT F59L1G81A SPI ID 92F18095407F7F7F. В свободном доступе имеется [документация](#).

Обычно все необходимое написано на первых страницах. В шапке первой страницы написано 1 Gbit. Полный размер 1Gbit можно представить как 128Mbyte или как $128 * 2^{20} = 134217728$ byte или перевести в шестнадцатеричную систему 0x8000000 байт. Это размер без учета spare области. Напряжение 3,3V для записи в конфигуратор переводится в милливольты 3300.

Далее на первой странице перечислены основные параметры «геометрии» флеш. Так же обычно в документации к каждой флеш имеется наглядная картинка «ARRAY ORGANIZATION», аналогично приведенной в текущей документации на странице 4.



Размер страницы: «- Page Program: (2K + 64) bytes» (тут он складывается из размера page + spare).

Page = 2K = $2 * 2^{10} = 2048 = 0x800$ байт;

Spare = 64 = 0x40 байт.

Общий объем страницы в считываемом дампе получается 0x840 байт.

Размер блока: «- Block Erase: (128K + 4K) bytes» (так же указан как сумма размеров страниц и spare).

128K = $128 * 2^{10} = 131072 = 0x20000$ байт размер блока без spare;

4K = $4 * 2^{10} = 4096 = 0x1000$ байт;

Суммарный размер блока равен: $0x20000 + 0x1000 = 0x21000$ байт;

Чтобы посчитать количество страниц в блоке, нужно размер блока (учитывая или не учитывая spare) поделить на размер страницы: $0x20000 / 0x800 = 0x21000 / 0x840 = 0x40 = 64$ (страницы в блоке).

Количество блоков можно получить разделив полный размер флеш на размер блока. Выше отмечалось, что полный размер считался без учета spare. Значит для получения количества блоков его нужно дели на размер блока без spare: $0x8000000 / 0x20000 = 0x400 = 1024$ блоков.

Шина данных определяется по распиновке. Например для TSOP48 корпусов у NAND стандартно выводы 29-32 и 41-44 это распределены I/O0-I/O7 т.е. 8 бит шины данных. Если помимо них, рядом расположены еще I/O8-I/O15, то это уже будет x16 битная.

Время доступа. Это значение в данной документации названо как tREA и в таблице «AC Characteristics for Operation» на странице 8 имеет значение 20ns. На странице 13 проиллюстрировано положение tREA среди последовательности сигналов. Это то время, за которое флеш из внутреннего буфера выставляет байт на шину данных I/O и готов к считыванию.

Адресные циклы Col/Row. Самое простое это найти таблицу «Address Cycle Map» на странице 4, которая обычно всегда наглядно описывает необходимые значения. Из нее видно, что первые две строки (количество ограничено значением *L) это «Column Address» адресация внутри страницы. Вторые две это «Row Address» для адресации по страницам/блокам и др.

Легко посчитать, что если полный размер страницы флеш 0x840 в двоичном виде можно представить 12 байт (100001000000) это и есть A0-A11 обозначенные в первых строках. Так же если посчитать, то общее количество страниц 0x10000. Для их перечисления от 0x до 0xFFFF требуется два байта. Это и есть две вторые строки по 8 бит в таблице A12-A27.

ЕСС бит. Это значение указывает на количество допустимых ошибок в течении гарантированных циклов записи флеш. Исходя из этого, производителем оборудования применяется алгоритм коррекции [ЕСС](#). Так на первой странице документации указано: «- 100K Program/Erase Cycles (with 1 bit/528 bytes ЕСС)» и означает, что в течении 100000 циклов стирания/записи количество ошибок не превысит 1 бит на 528 байт. В случае конфигурирования округляется до 512. Это значение не является критическим для работы программатора и используется лишь при верификации для оценки износа флеш после записи.

Бит в ячейке. В случае с F59L1G81A это SLC. Среди прочих данных на первой странице указано: «Memory Cell: 1bit/Memory Cell».

Небольшая подсказка:

Single-Level Cell or SLC (1 bit per cell)

Multi-Level Cell or MLC (2 bits per cell)

Triple-Level Cell or TLC (3 bits per cell)

Quad-Level Cell or QLC (4 bits per cell)

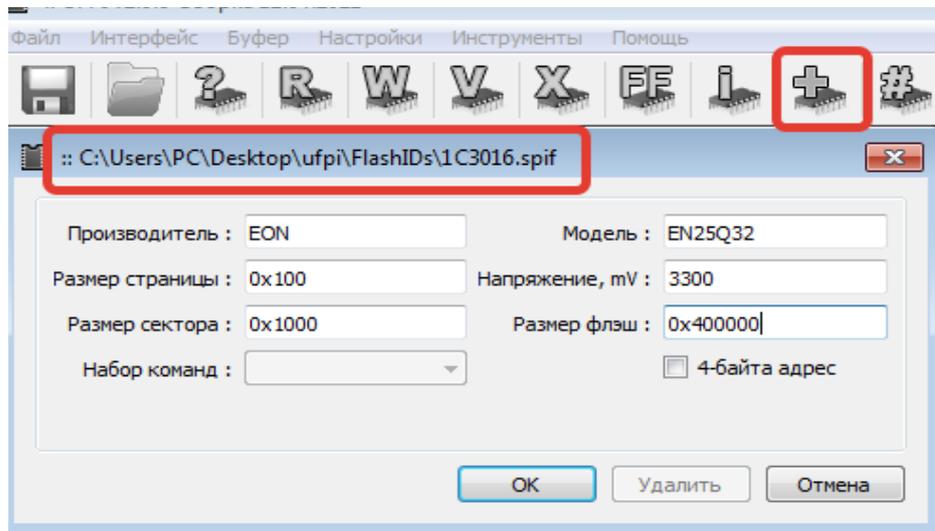
Penta-Level Cell or PLC (5 bits per cell)

После правильного заполнения всех полей, внизу будет посчитан полный размер, совпавший с расчетами выше.

Кнопка «Сохранить» сохраняет конфигурационный файл в папку FlashIDs и по умолчанию при чтении ID уже будет срабатывать эта конфигурация. Чтобы удалить, можно воспользоваться кнопкой «Удалить» или удалить сам файл, указанный в заголовке окна конфигурирования.

16.2.2 Приложение. Составление конфигурационного файла SPI Flash.

Если флеш не определяется, или определяется не верно, по кнопке «+» открывается форма конфигуратора, куда нужно вписать основные для работы параметры микросхемы.



Основным достоверным источником официальная документация от производителя, которая обычно свободно распространяется в интернет.

Рассмотрим флеш EON EN25Q32B SPI ID 1C3016 (1C30161C30161C30:9F). На официальном сайте имеется [документация](#).

Обычно все необходимое написано на первых страницах. В шапке первой страницы написано 32 Megabit. Полный размер 32Mbit можно представить как $32/8 = 4\text{Mbyte}$ или как $4 \cdot 2^{20} = 4194304\text{byte}$ или перевести в шестнадцатеричную систему $0x400000$ байт.

Напряжение далее указано как: «Full voltage range: 2.7-3.6 volt». В милливольтгах общепринято 3300mV вводить в конфигуратор.

Размер страницы дословно тут же на первой странице: «256 bytes per programmable page». Можно так и вписать в конфигуратор 256, либо перевести в шестнадцатеричный вид $0x100$.

Размер сектора чуть ниже: «1024 sectors of 4-Kbyte». Это перевести в байты $4 \cdot 2^{10} = 4096 = 0x1000$.

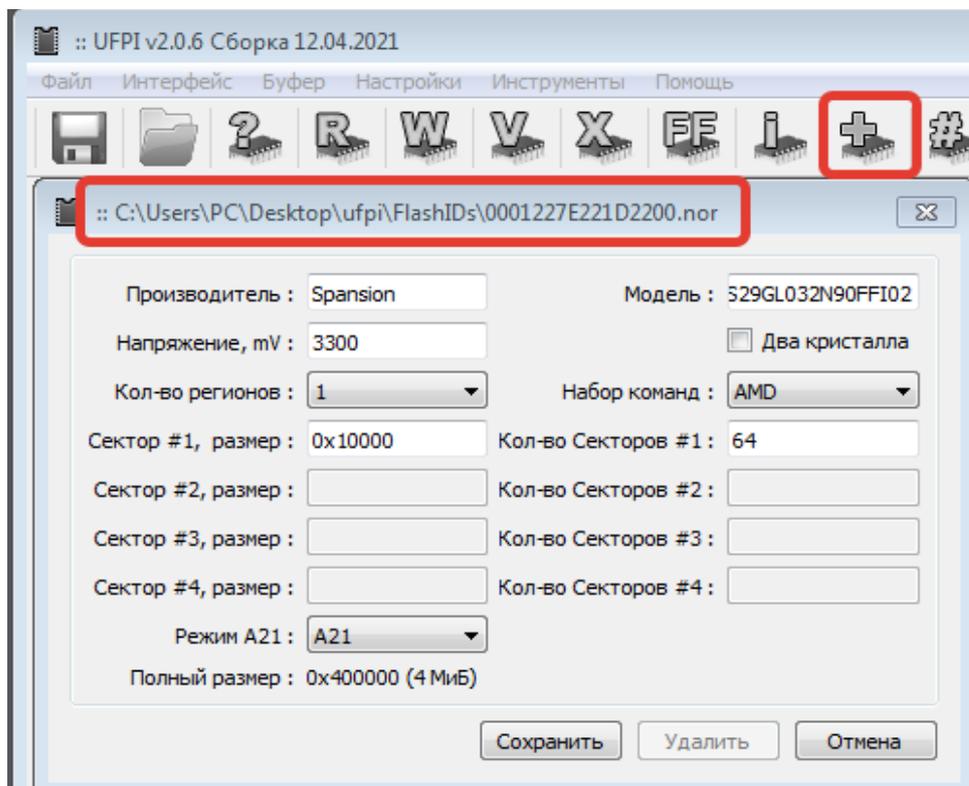
Но для убедительности обязательно свериться с определившимся ID на странице 15. ID в логе «SPI ID 1C3016».

4-байт адрес понадобится лишь для больших флеш.

Например на странице 22 документации присутствует описание команды чтения и строка: «The instruction code for the Read Data Bytes (READ) instruction is followed by a 3-byte address (A23-A0)». В иллюстрации тоже показаны эти 24 байта. Это предельный размер адресуемой флеш получится три байта от 0 до $0xFFFFFFFF = 0x1000000$ байт. При переводе в Мебибайты $16777216 / 2^{20} = 16\text{MiB}$ или в битах $16 \cdot 8 = 128\text{Mib}$. Т.е. для SPI флеш более 128Mib (например 25Q256) уже будет в документации описано о 4 байте адреса.

16.2.3 Приложение. Составление конфигурационного файла NOR Flash.

Если флеш не определяется, или определяется не верно, по кнопке «+» открывается форма конфигуратора, куда нужно вписать основные для работы параметры микросхемы.



Основным достоверным источником официальная документация от производителя, которая обычно свободно распространяется в интернет.

Рассмотрим флеш Spansion/Cypress S29GL032N90FFI02. NOR ID 0001227E221D2200 AMD Algo (CFI).

На официальном сайте имеется [документация](#).

Объем и рабочее напряжение можно увидеть в «шапке» на первой странице. Тут 3V записывается в окно конфигуратора как 3300(mV).

Полный размер 32Mbit можно представить как $32/8 = 4\text{Mbyte}$ или как $4 \cdot 2^{20} = 4194304\text{byte}$ или перевести в шестнадцатеричную систему $0x400000$ байт.

Обращаем внимание на полную маркировку. На 12 странице можно определить, что MODEL NUMBER = 02 = $x8/x16$, VCC = VIO = 2.7 – 3.6 V, Uniform sector, WP#/ACC = VIL protects lowest addressed sector.

Организация памяти. На 19 странице идет таблица с перечисленными секторами. Видно из таблицы, что все 64 сектора одинакового размера 64KB. В таком виде в конфигураторе выбирается как один регион и соответственно размер и количество секторов.

Аналогично переводим 64KB в байты, умножив на $2^{10} = 64 \cdot 1024 = 65536 = 0x10000$.

Если опуститься на 20 страницу для соседней модели памяти, то по таблице видно, что регионов два. Первый считая по списку 63 сектора по 64KB и второй 8 секторов по 8KB.

Стоит внимательно обращать внимание на нумерацию секторов, поскольку она может идти как от начала, так и с конца. Так называемые «Top Boot Sector» и «Bottom Boot Sector». От этого будет зависеть очередность регионов.

Размер флеш соответствует размеру от начала первого до конца последнего сектора и можно проверить в столбце «8-bit Address Range». Получается от 0 до 3FFFFFFh в сумме получается $0x400000$ байт. Это соответствует ранее рассчитанному.

Проверка ID. На странице 29 приведена таблица. Всегда стоит сверять ID с документацией. Поскольку неверный ID может свидетельствовать о плохом контакте, неисправности или несоответствии этой документации. Исходный ID 0001227E221D2200.

Первыми всегда идут Manufacturer ID: Cypress Products

DQ8 to DQ15 = 0x00;

DQ7 to DQ0 = 0x01;

Далее строки для S29GL032N

Cycle 1 = 0x227E;

Cycle 2 = 0x221D;

Cycle 2 = 0x2200;

Остальные столбцы можно сверить из полного ID в логге. Там бывает так же присутствует полезная информация.

Набор команд. Наиболее ходовыми являются AMD. На странице 47 можно увидеть в таблице команды. Первыми циклами в командах в случае с AMD всегда вставлены подобные 555/AAA наборы байт.

В случае с набором команд INTEL таких добавочных байт не будет и будут просто команды. Например можно посмотреть в [документации](#) на Intel 28F256J3 на 35 странице.

Не стоит думать, что если флеш Intel, то и система команд у нее Intel. Это не всегда так.

Режим A21 выбирается исходя из распиновки флеш. Для TSOP56 это 15 (либо 13 для TSOP48) нога микросхемы. В случае с S29GL032N она выполняет роль адресной линии A21.

Если открыть [документацию](#) на Intel 28F256J3 на 15 странице, то на 15 ноге будет Vpp/Vrep. В таблице 7 данной документации описано, что обычно он подключен к Vcc (2.7 V–3.6 V)

После правильного заполнения всех полей, внизу будет посчитан полный размер, совпавший с расчетами выше.

Кнопка «Сохранить» сохраняет и по умолчанию при чтении ID уже будет срабатывать конфигурация заполненная и сохраненная в файле в папке с программой. Чтобы удалить, можно воспользоваться кнопкой «Удалить» или удалить сам файл, указанный в заголовке окна конфигулятора.

16.2.4 Приложение. Файловый менеджер и монтирование ФС.

Разметка и таблица разделов.

После установки EMMC, и чтения ID(в «Доп. функции» можно выбрать для анализа дампа или EMMC), при удачном распознавании разметки получаем в лог что-то подобное:

```
eMMC проверка дампа...LG
```

```
swue SQUASHFS
```

```
swue SQUASHFS
```

```
cert SQUASHFS
```

```
authcxt EXT4
```

```
rootfs SQUASHFS
```

```
...
```

```
fonts SQUASHFS
```

```
smartkey SQUASHFS
```

```
db8 EXT4
```

```
data EXT4
```

```
apps EXT4
```

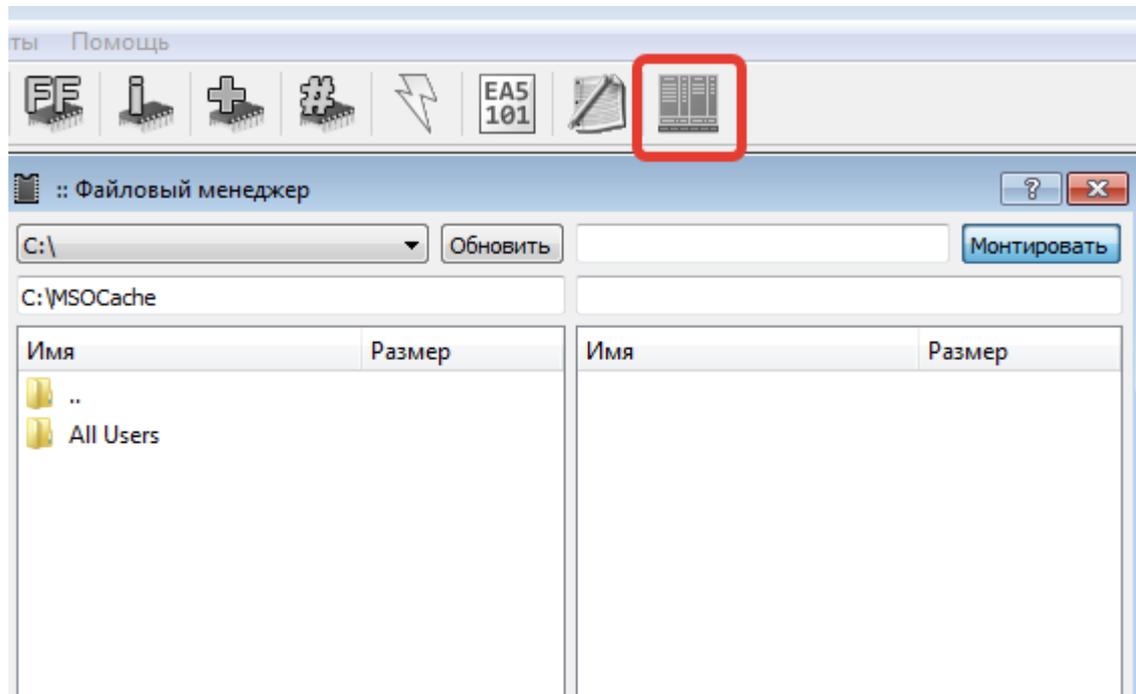
```
eMMC анализ дампа завершен. Извлечено 49 части(ей)
```

Если ничего не найдено:

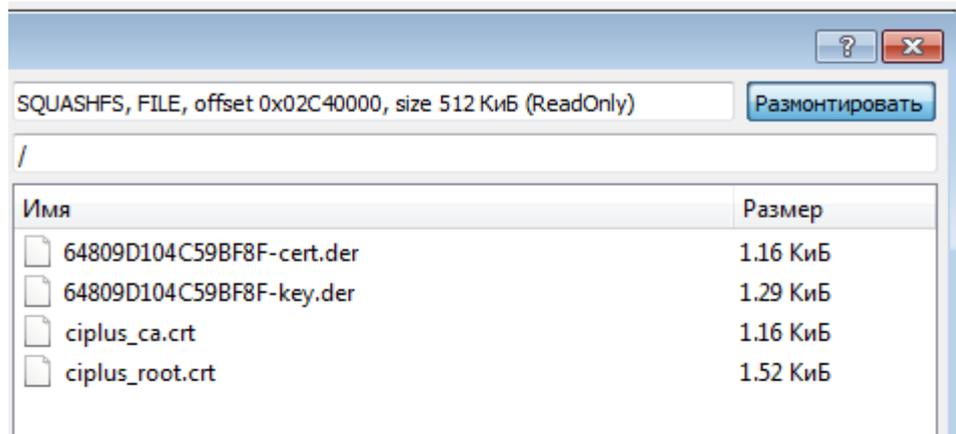
```
eMMC проверка дампа...Разделы не найдены!
```

Проверяем лог. Описаны найденные разделы, количество и тип ФС, с которыми можно работать на данный момент. Полный список разделов можно увидеть в окне «режим доступа» (Как вариант, всегда можно использовать udev или скрипт для его создания на ранее считанным дампе. После отработки скрипта не забываем выйти из работы скрипта)

Кнопкой на панели открываем файловый менеджер и нажимаем «[Монтировать](#)»



При удачном монтировании в файловом менеджере появится содержимое и описание раздела.



Файловый менеджер позволяет проходить по каталогам, копировать, заменять и многие другие операции.

Операции над выделенными файлами можно производить путем вызова контекстного меню по нажатию правой кнопки мыши.

16.2.5 Приложение. Вычитывание и запись разделов / регионов.

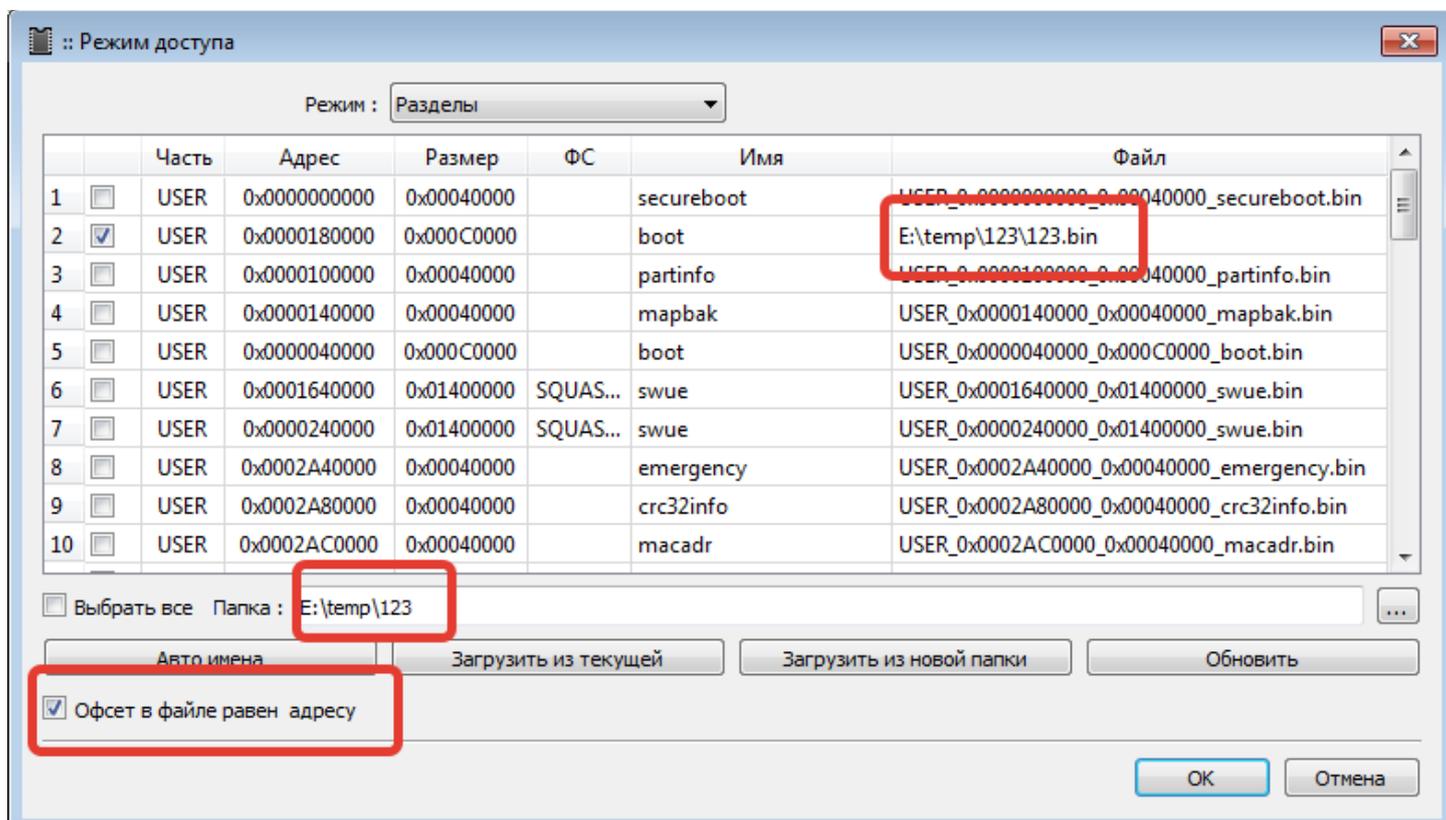
Вычитывание/запись разделов через режим доступа, режим «разделы».

Наиболее удобным и наглядным в работе является разделение на разделы с помощью анализа при чтении ID, или с применением скриптов, или .UDEV файлов.

Открываем окно режима доступа, выбираем режим «разделы», отмечаем галками необходимые, выбираем рабочую папку, обновить и «ОК». Далее читаем/пишем обычными кнопками чтения/записи. Все регионы будут отдельными файлами в выбранной папке.

Запись/восстановление одного раздела из дампа.

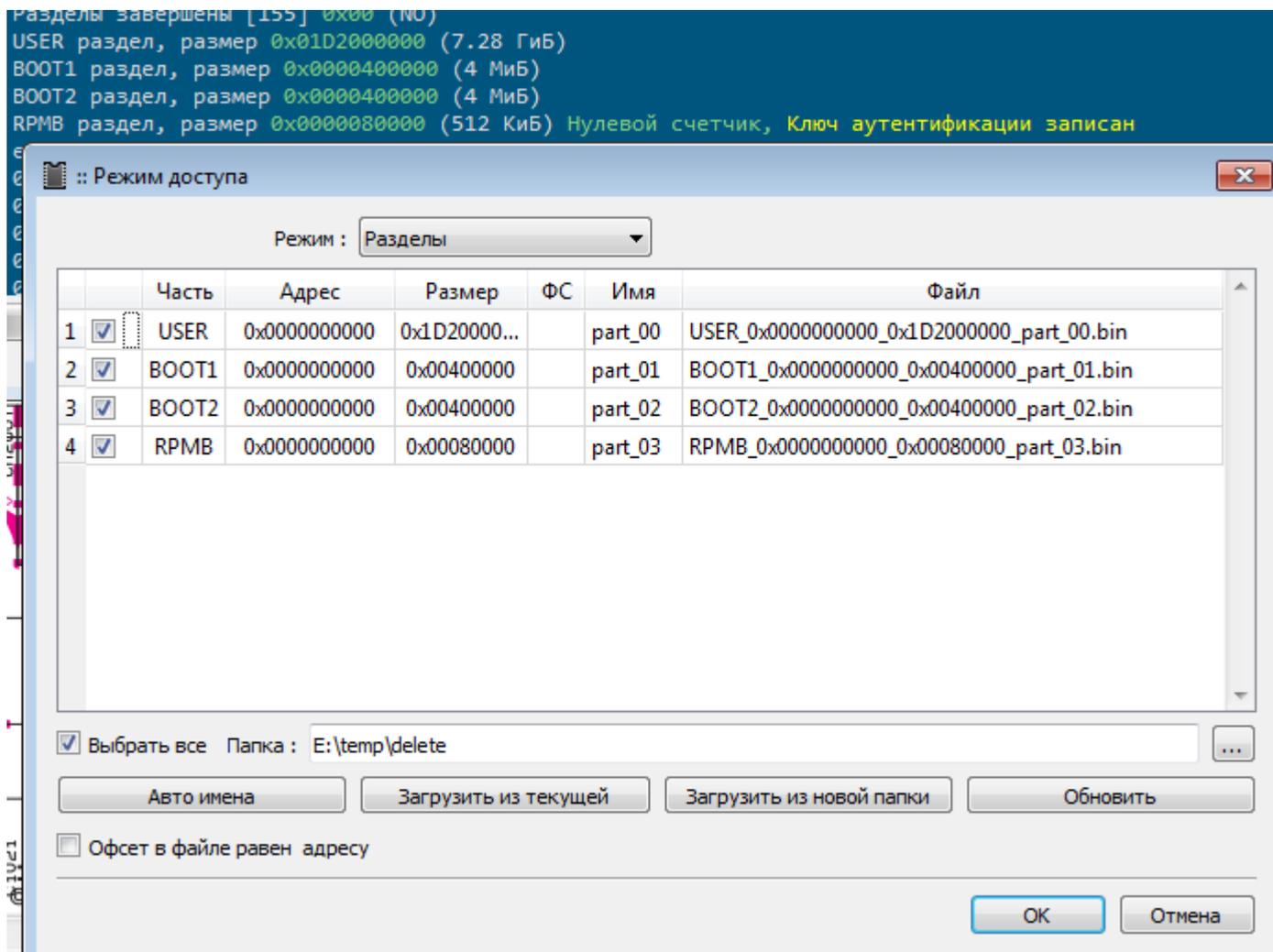
Помимо удобного «Бэкап платформы», можно записывать отдельно раздел из целого дампа донора. Для этого в режиме раздел выбираем в строке нужного раздела файл полного дампа. Папка рабочая соответственно. И самое главное устанавливаем галку «Офсет в файле равен адресу». Жмем «ОК» и запись. Из полного файла дампа запишется только нужный нам раздел.



16.2.6 Приложение. Полный бэкап с помощью режима доступа и .UDEV.

Режим доступа позволяет сохранять/записывать разделы не только из USER eMMC, но и комбинировать.

Прочсть ID флеш, и заполнить в окне режима доступа «Разделы» размеры разделов:



Так можно и читать и записывать. Можно выбрав файлы из папки, в которую читался «Полный бэкап» записать все нужные разделы разом.

Можно аналогично составив или сохранив .UDEV файл:

```
[DESC]
Name = Partitions
FlashType1 = eMMC
FlashBase1 = 0
```

```
[PARTITIONS]
PartitionsMode = true
0x0000000000,0x1D200000,part_00,USER,USER_0x0000000000_0x1D2000000_part_00.bin
0x0000000000,0x00400000,part_01,BOOT1,BOOT1_0x0000000000_0x00400000_part_01.bin
0x0000000000,0x00400000,part_02,BOOT2,BOOT2_0x0000000000_0x00400000_part_02.bin
0x0000000000,0x00080000,part_03,RPMB,RPMB_0x0000000000_0x00080000_part_03.bin
```

16.2.7 Приложение. Работа с разделами. Разделение, замена, запись.

Чтение разделов из флеш в папку.

Вставить флеш, в «Доп. функциях» установить «Анализ при чтении ID» - eMMC и нажать кнопку «?» (Прочитать ID).

Если разделы найдены, то открыть «[Режим доступа](#)» - «Разделы», выбирать рабочую папку, установить галки на нужных разделах, нажать «ОК» и чтение. Папка заполнится вычитанными разделами.

Если разделы автоматически не найдены, то можно либо подгрузить [.UDEV файл](#) с описанием разделов, либо вычитать полный дамп и запустить скрипт [Partition finder](#).

Разделение дампа на разделы.

Выбрать полный дамп и запустить скрипт [Partition finder](#). Далее следовать подсказкам:

```
Button 'Read' - analyze dump from path 'Read to'.
Button 'Write' - search for file systems in dump from path 'Read to'.
Button 'Verify' - enter list of parts to extract.
Button 'Erase' - extract parts.
Button 'Blank Check' - split NAND dump from 'Read to' into main and spare.
Button 'Info' - print parts for extract.
Buttons 'Stop' or 'Execute script'- exit from script.
```

Следуя подсказке, при выбранном в поле «Читать в» и вычитанном дампе нажимаем кнопку «R» и если все определилось, то получаем список разделов и .udev файл на основании анализа в папке с дампом.

```
Search for partitions in the dump.
Input File 'E:/temp/123/2/bbk40lex5026.User.bin'.
File size = 0xE9000000
Found MSTAR ANDROID marking.
Number of parts 18.
№ Offset Length Name
1 0x00000000 0x00200000 Partitions_table
2 0x00200000 0x00300000 MBOOT
3 0x00500000 0x00200000 MPOOL
4 0x00700000 0x00080000 misc
...
16 0xDB4C0000 0x04200000 tvcustomer
17 0xDF6C0000 0x06400000 factory
18 0xE5AC0000 0x03540000 gap_01
UDEV file created with the name
E:/temp/123/2/partitions_2021-05-05_14-03-55.udev
```

Для извлечения следуя подсказкам жмем «V» для выбора разделов или сразу «X» для извлечения полного списка.

Работа с разделами в проводнике.

Работу заменой между собой разделов в двух дампах можно проводить любым удобным способом как с файлами. Например обычном проводнике, или в TotalCommander и пр...

Запись разделов.

Открыть «[Режим доступа](#)» - «Разделы», нажать «Загрузить из новой папки» или выбрать рабочую папку и нажать «Загрузить из текущей». Нажать «ОК» и можно записывать. Все разделы

последовательно запишутся. Автозаполнение таблицы разделов происходит исходя из согласованного имени файла, в котором указаны размещение, адрес и размер.

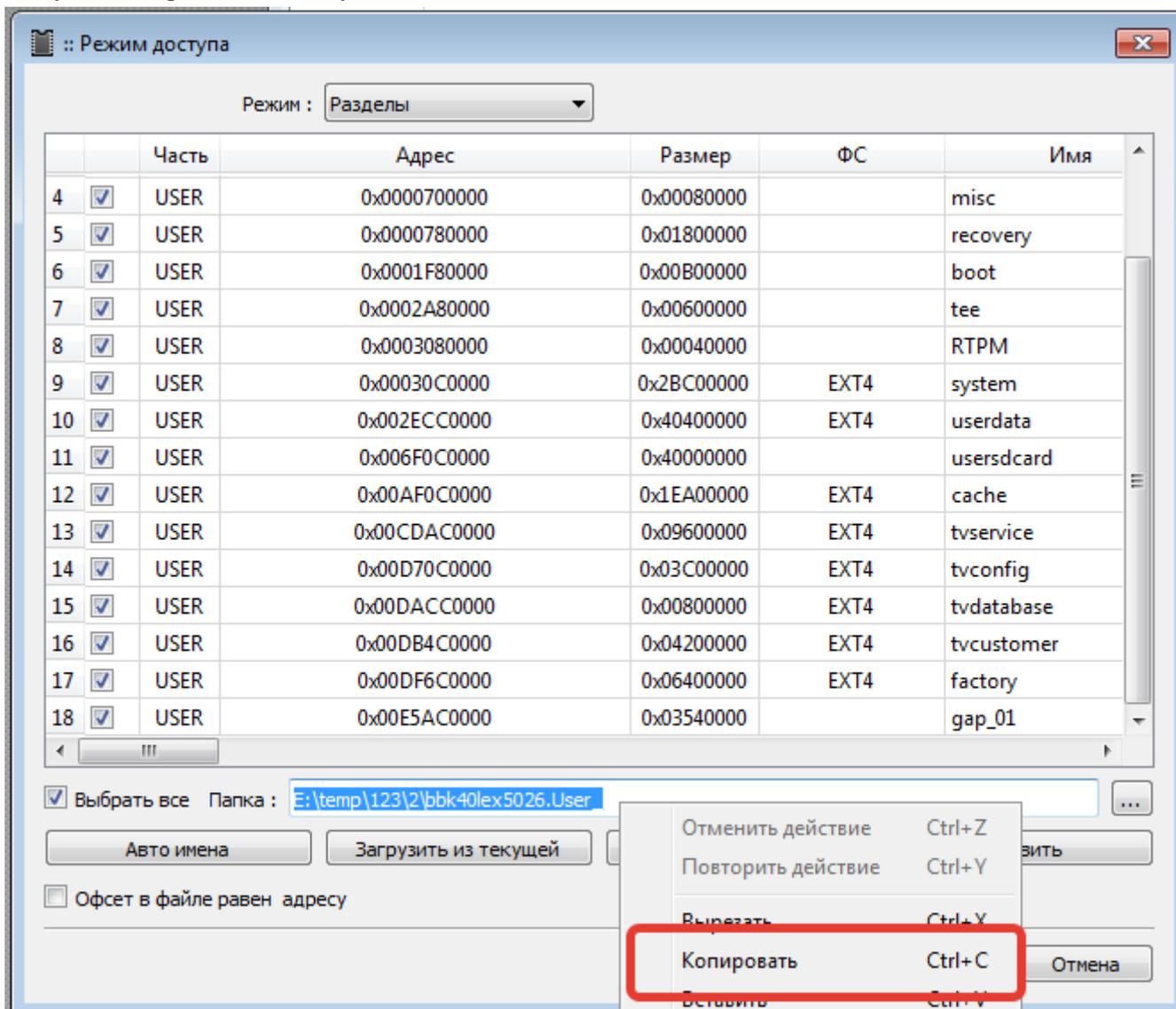
Сборка дампа из разделов.

Для объединения файлов можно воспользоваться [скриптом](#).

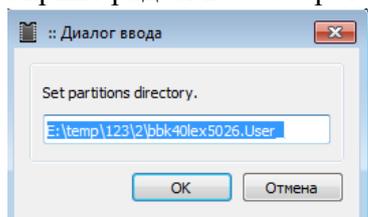
Для обратной сборки в единый дамп, при условии, что структура имен разделов соответствует, можно воспользоваться [скриптом](#) автоматической сборки.

Для этого потребуется путь рабочей папки, в котором лежат файлы разделов:

Запустить скрипт, ввести путь этой папки и нажать «ОК»



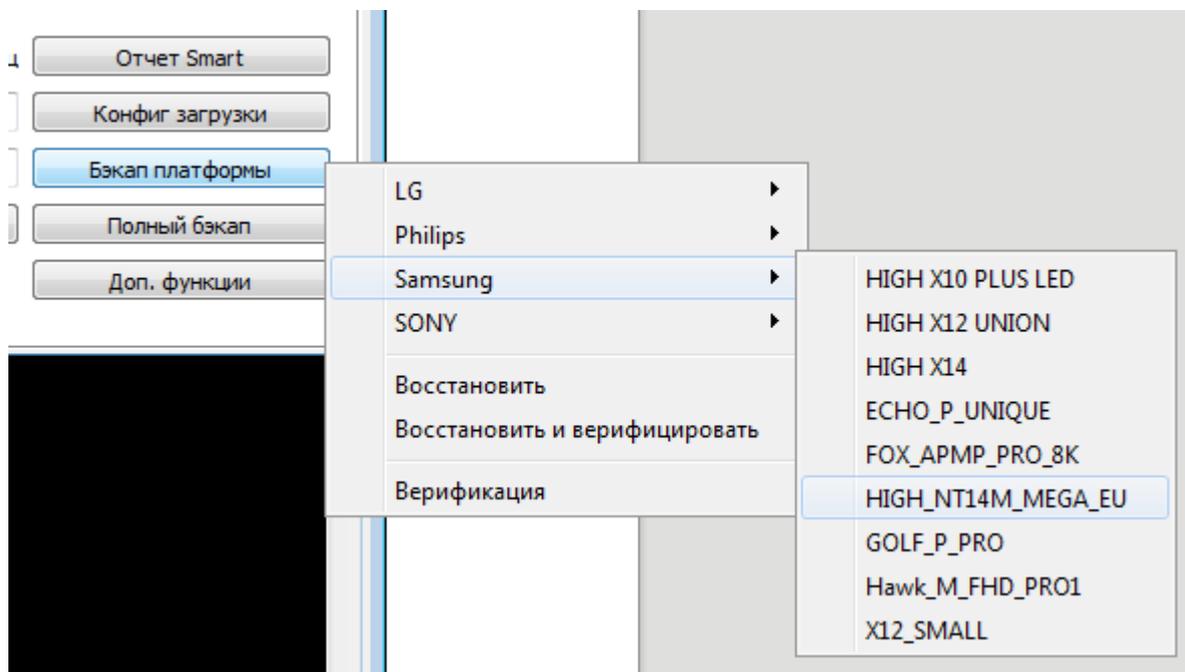
Скрипт рядом все соберет в один файл в папке FULL DUMP.



16.2.8 Приложение. Работа с «Бэкап платформ».

После нажатия кнопки «?» и чтения ID нам становится доступна кнопка меню "Бэкап платформ".

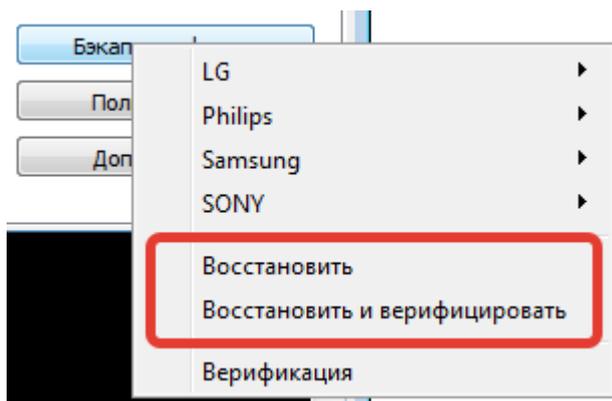
При нажатии на кнопку "Бэкап платформ" появляется меню, которое основано на содержании файлов .rbi из папки RBI в каталоге программы UFPI.



После выбора платформы указанные данные будут сохранены в UBAK файле.

```
eMMC бэкап платформы для 'LC_LD_LE42B_42G'...
Сохранение данных в Файл 'E:\temp\backup\LC_LD_LE42B_42G.ubak'...Норма
Чтение eMMC USER с адреса 0x03F00000, размер 0x00080000...Норма
Чтение eMMC USER с адреса 0x03F80000, размер 0x00080000...Норма
Завершено за 5.53 секунд(у). Скорость загрузки кода 185.07 КиБ/сек.
```

При восстановлении из UBAK файла все эти части будут записаны обратно



(Можно выбрать восстановление/восстановление+верификация и регионы привязки пропишутся во флеш, либо просто верификация региона привязки.)

```
eMMC восстановление платформы из 'E:\temp\backup\LC_LD_LE42B_42G.ubak'...MMC Рев.1
Имя бэкапа 'LC_LD_LE42B_42G', размер 1 МиБ, создан 9 Ноябрь 2020 21:52:22
Запись eMMC USER с адреса 0x03F00000, размер 0x00080000...Норма
Запись eMMC USER с адреса 0x03F80000, размер 0x00080000...Норма
Завершено за 1.99 секунд(у). Скорость загрузки кода 513.8 КиБ/сек.
eMMC верификация платформы с 'E:\temp\backup\LC_LD_LE42B_42G.ubak'...MMC Рев.1
Имя бэкапа 'LC_LD_LE42B_42G', размер 1 МиБ, создан 9 Ноябрь 2020 21:52:22
Верификация eMMC USER с адреса 0x03F00000, размер 0x00080000...Норма
```

Верификация eMMC USER с адреса 0x03F80000, размер 0x00080000...Норма
Завершено за 1.78 секунд(у). Скорость загрузки кода 576.58 КиБ/сек.

16.2.9 Приложение. Проверка хэшей, ключей, восстановление и извлечение ключей и BOOT на примере платформы QV14/15.

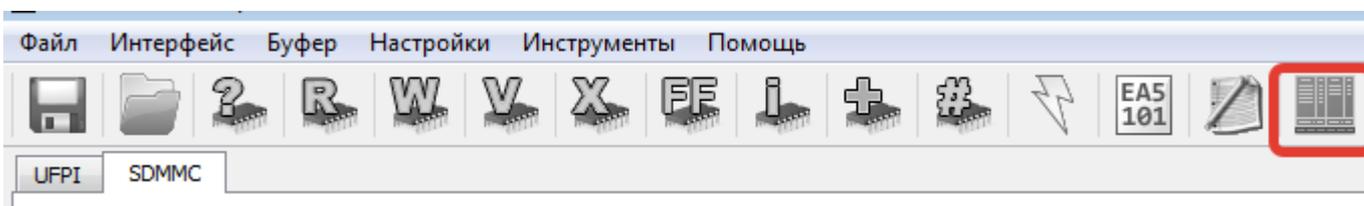
Функционал монтирования приобретает дополнительные полезные функции. Такие как проверка ключей, boot (рассмотрен будет пример QV14)

Ну например имеем EMMC от платформы QV14/15. Включаем анализ дампа микросхемы при чтении ID (Настройки - программатор - модули - SDMMC).

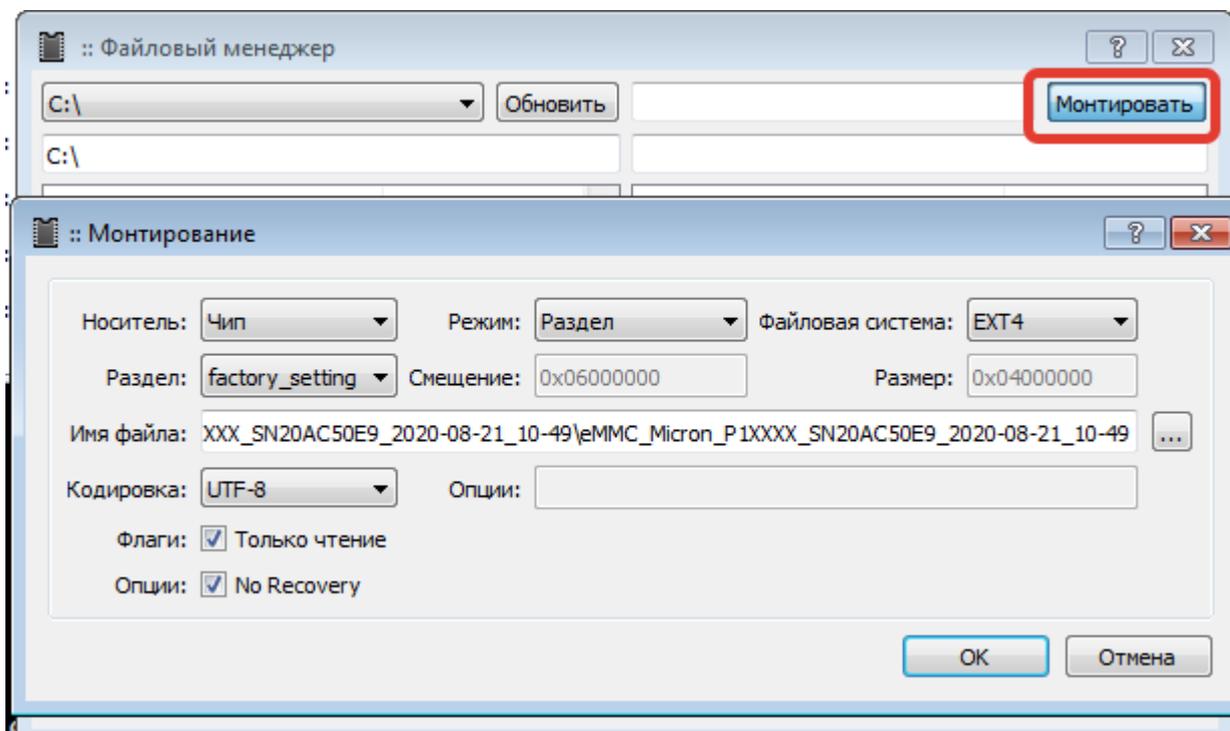
Определяем флеш и видим в логе разделы доступные для монтирования:

```
eMMC проверка дампа...GPT раздел, смещение 0x01
factory_setting EXT4
system EXT4
user_setting EXT4
cache EXT4
firmware EXT4
factory_data EXT4
userdata EXT4
eMMC анализ дампа завершен. Извлечено 13 части(ей)
```

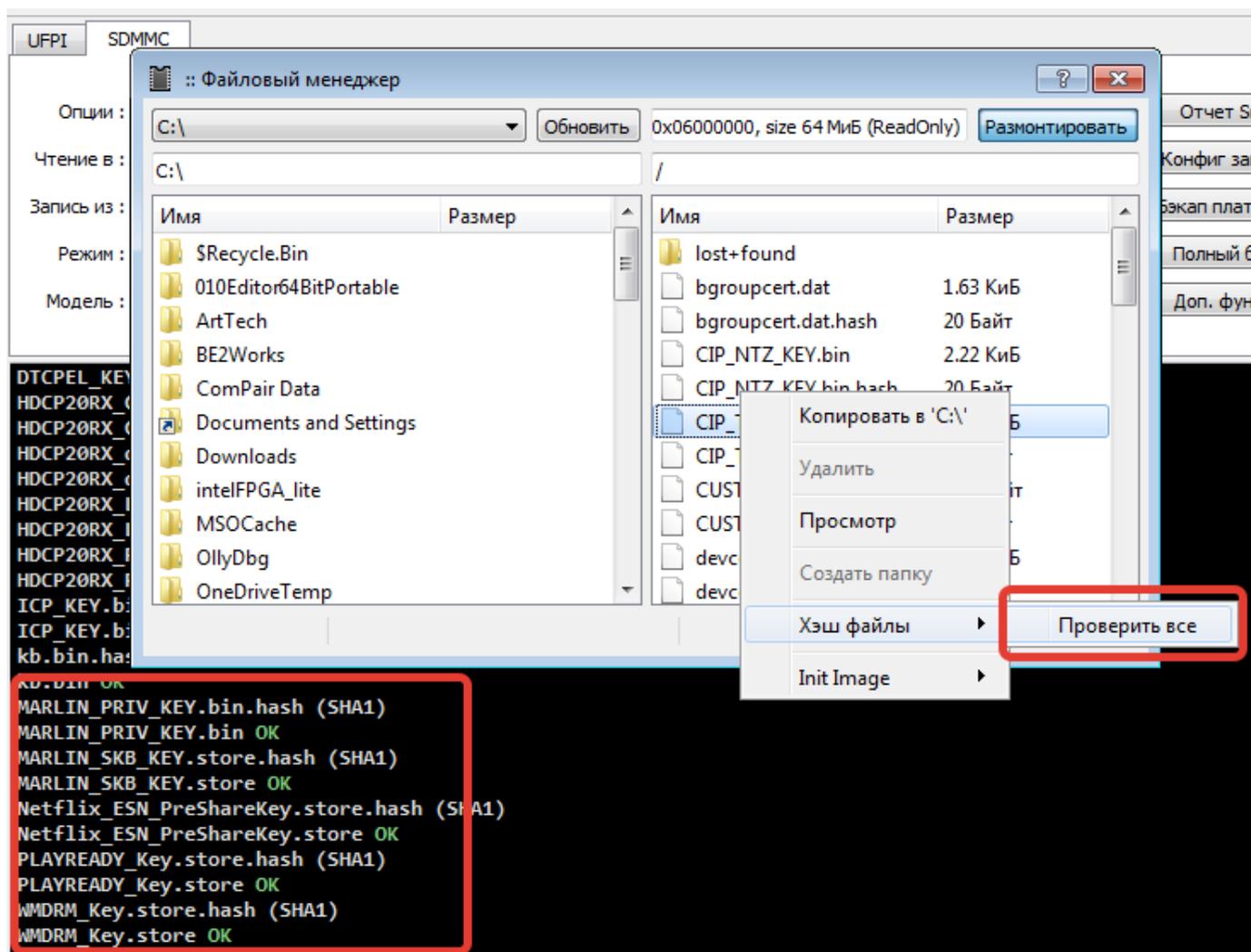
Нажать кнопку файлового менеджера



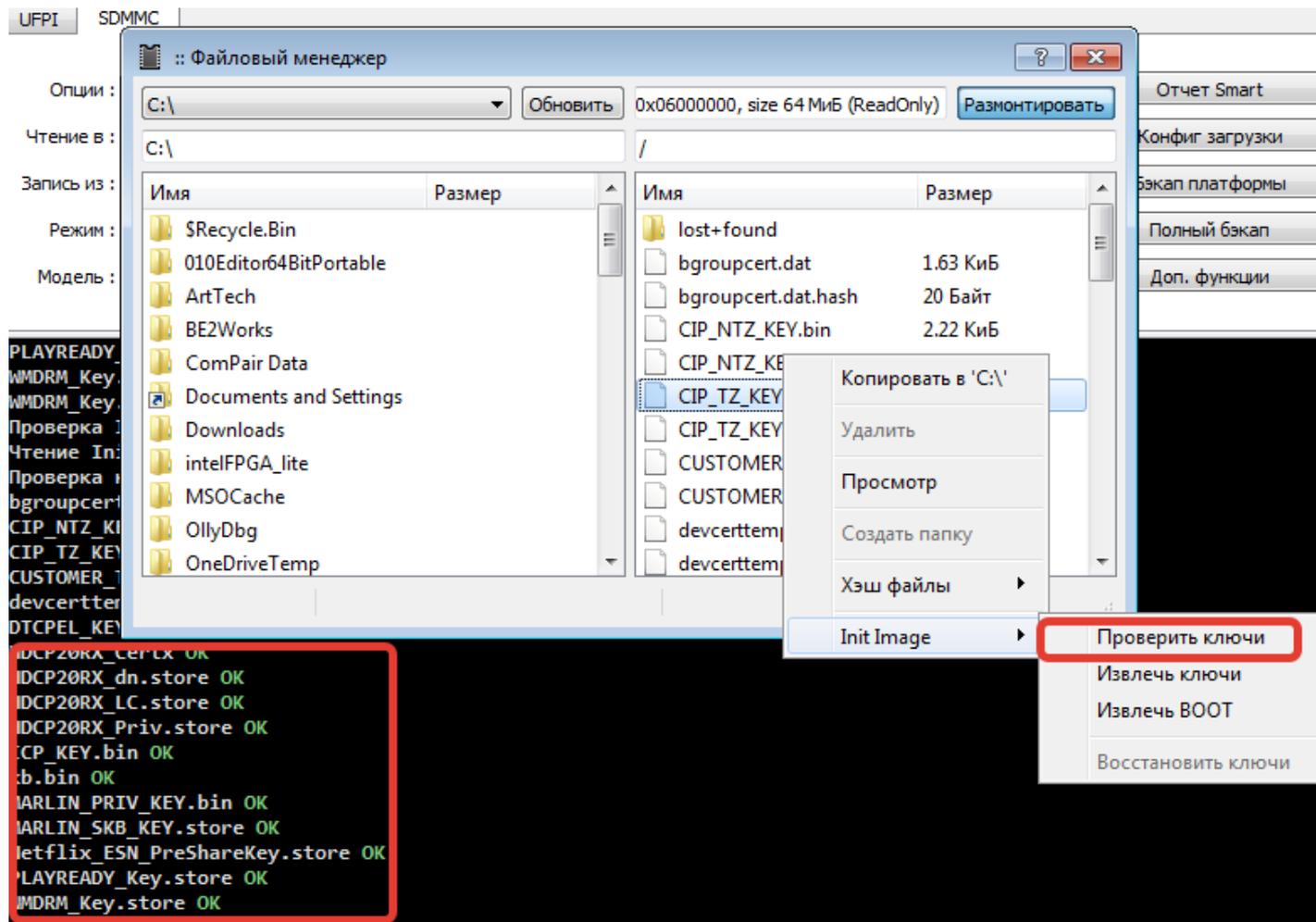
Нажать кнопку "монтировать", выбрать последовательно носитель, режим, файловую систему и раздел .



Жмем "ОК" и переходим в файловый менеджер с примонтированным разделом. Далее по меню ПКМ можно проверить хэши файлов в папке.



Можно проверить/извлечь ключи/boot.



И самое интересное, что если снять в настройках монтирования флаг "Только для чтения", то доступны остальные пункты и испорченные ключи можно волшебным образом автоматом восстановить.

16.2.10 Приложение. Исправление дампа и запись дампа NAND на примере Samsung D5500.

Коррекция от ошибок и очистка от бэд блоков в дампе.

Подключить программатор с микросхемой.

В настройках [ЕСС](#) выбрать «Автоопределение»

В меню «Инструменты» выбрать «[Анализ дампа](#)».

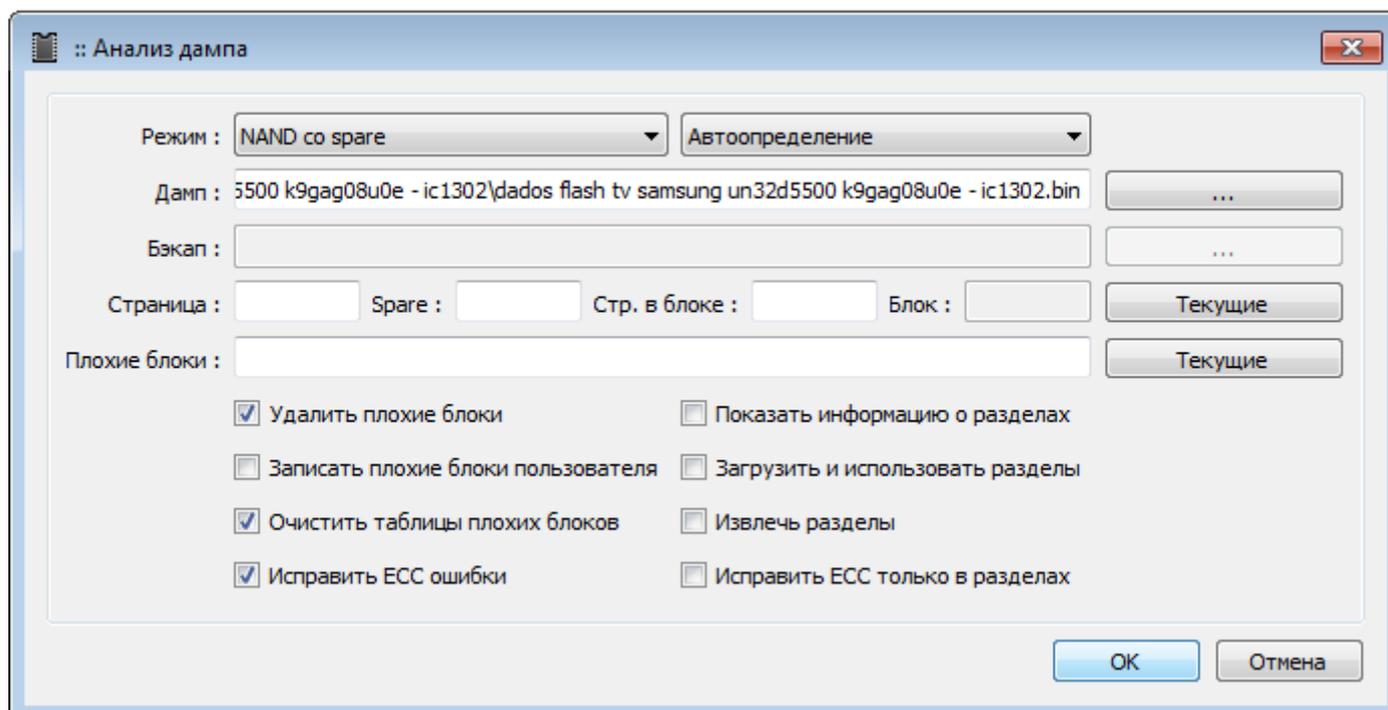
В поле режим выбрать «NAND со spare» и «Автоопределение».

В поле «Дамп» выбрать исправляемый дамп. Для демонстрации будет использован всем известный дамп K9GAG08U0E от Dados для UN32D5500. *Стоит учесть, что в сети часто встречается неполный дамп, считанный через ENTТ. Этот дамп не годится для обработки и использования! Нормальный дамп имеет размер «Общий размер со spare 0x88A7D800 2.13 ГиБ».*

Поля «Страница», «Spare», «Стр. в блоке» заполнять нет необходимости. Дампы этих аппаратов определяются автоматически.

Поля «Плохие блоки» и галку «Записать плохие блоки пользователя» устанавливать не нужно.

Установить галки «Удалить плохие блоки», «Очистить таблицы плохих блоков» и «Исправить ошибки ЕСС» и нажать «ОК».



Файл «чистого» дампа будет пересохранен с добавлением к имени «fixed_nobads_ecc_clean.bin». В логе будут описаны детали по имеющейся в дампе таблице ББ и количеству исправленных ошибок.

Запись «чистого» дампа с управлением «Использовать резервную зону».

Включить «[Анализ дампа при чтении ID](#)», «Асинхронный» и «Не записывать пустые данные» в «Доп. функции».

Включить обнаружение ББ по маркеру. «Первый байт маркер в spare» равно 0x00.

Управление ББ выставить «Использовать резервную зону», тип таблицы «Автоопределение».

ЕСС выбрать схему с «Автоопределение».

Все эти настройки можно выставить загрузив файл [.UDEV](#).

NAND LPCH блок 0x07FE (2046) по адресу 0x86AE4C00
 NAND LPCH TPCB 0x07FC (2044), годы 1-1
 NAND UPCH блок 0x07FD (2045) по адресу 0x869D7200, 1 запись(-и) (Активен)
 NAND UPCH TPCB 0x07FC (2044), годы 1-2, инфо 0x0000
 NAND RBA 0x07FB-0x0785 (2043-1925), 0x77 (119) блоков
 NAND раздел #1, 'part-00-ID1B', блок 0x0 (0), кол-во 0xD6 (214)
 NAND раздел #1, адрес 0x00000000 (0x00000000), размер 0x0D600000 (0x0E163C00)
 NAND раздел #2, 'part-01-ID1C', блок 0xD6 (214), кол-во 0x100 (256)
 NAND раздел #2, адрес 0x0D600000 (0x0E163C00), размер 0x10000000 (0x10DA0000)
 NAND раздел #3, 'part-02-ID1D', блок 0x1D6 (470), кол-во 0xD6 (214)
 NAND раздел #3, адрес 0x1D600000 (0x1EF03C00), размер 0x0D600000 (0x0E163C00)
 NAND раздел #4, 'part-03-ID1E', блок 0x2AC (684), кол-во 0x410 (1040)
 NAND раздел #4, адрес 0x2AC00000 (0x2D067800), размер 0x41000000 (0x4475A000)
 NAND раздел #5, 'part-04-ID1F', блок 0x6BC (1724), кол-во 0xC8 (200)
 NAND раздел #5, адрес 0x6BC00000 (0x717C1800), размер 0x0C800000 (0x0D2A5000)
 NAND раздел #6, 'UPCH', блок 0x7FD (2045), кол-во 0x1 (1)
 NAND раздел #6, адрес 0x7FD00000 (0x869D7200), размер 0x00100000 (0x0010DA00)
 NAND раздел #7, 'LPCH', блок 0x7FE (2046), кол-во 0x1 (1)
 NAND раздел #7, адрес 0x7FE00000 (0x86AE4C00), размер 0x00100000 (0x0010DA00)
 Загружаем список плохих блоков микросхемы...2 плохих блока(-ов)
 Записываем новый список плохих блоков...
 NAND плохой блок 0x0714 (1812) заменен на 0x07FB (2043)
 NAND плохой блок 0x0819 (2073) игнорирован
 Кодируем ECC в измененных страницах...MSTAR_P8K_SP436_CW8_S12L42
 Управление плохими блоками Использовать резервную зону

При необходимости ввести список дополнительных плохих блоков через «Доп. функции» - «Плохие блоки» - «Пометить бэды» и прочитать ID повторно.

Проверить список ББ. Анализ дампа сконфигурирует программатор для дальнейшей работы с флеш и дампом.

Стереть флеш, если не установлена галка «Стирать перед записью» в «Доп. функции»

Записать, верифицировать.

Источник данных Файл '***dados flash tv samsung un32d5500 k9gag08u0e - ic1302_fix_nobads.bin'...Норма

Пользовательские флаги задачи Асинхронный, DNWRB, PACKUSB

Стирание NAND с адреса 0x00000000, размер 0x88A7D800...

Блок 0x07FB (2043) адрес 0x7FB00000 (0x867BBE00) Пропуск!

Блок 0x0819 (2073) адрес 0x81900000 (0x88754A00) Пропуск!

Запись NAND с адреса 0x00000000, размер 0x88A7D800...

Завершено за 1 мин. 38 сек. Скорость загрузки кода 22.12 МИБ/сек.

Источник данных Файл '***dados flash tv samsung un32d5500 k9gag08u0e - ic1302_fix_nobads.bin'...Норма

Пользовательские флаги задачи Асинхронный, ECC

Верификация NAND с адреса 0x00000000, размер 0x88A7D800...

Блок 0x00DE (222) адрес 0x0DE00000 (0x0E9D0C00) ECC, 1 бит(-а) исправлено//это обычные битовые ошибки в страницах, они имеют кода коррекции и исправляются.

...

Страница 0xED0C (60684), блок 0x01DA (474) адрес 0x1DA18000 (0x1F353870) Некорректируемая ECC ошибка!//это ошибка в странице с данными, с которой алгоритм коррекции ECC не справился. Блоки с такими ошибками нужно пометить как ББ.

...

Блок 0x02BF (703) адрес 0x2BF00000 (0x2E46A600) 1 битовых ошибок в 1 страницах! //это ошибка в пустой странице, подобные не имеют кода коррекции и не исправляются.

...

Блок 0x06EA (1770) адрес 0x6EA00000 (0x74834400) ECC, 4 бит(-а) исправлено

Блок 0x07FB (2043) адрес 0x7FB00000 (0x867BBE00) Плохой блок
 Блок 0x0819 (2073) адрес 0x81900000 (0x88754A00) Плохой блок
 ECC проверено в 36766 страницах, 228706 пустых страниц пропущено
 Исправлено 193 ECC битовых ошибок, 5 в данных ECC
 Завершено за 3 мин. 52 сек. Скорость загрузки кода 9.42 МиБ/сек.

Если в процессе верификации будут найдены некорректируемые ошибки, то добавить дополнительные плохие блоки, заново прочитать ID и заново записать.

Если используется сомнительная или старая флеш, со стертыми старыми маркерами ББ, то стоит уделить дополнительное время на определение и пометку ее плохих блоков. Для этого после записи отложить на несколько дней флеш, чем дольше, тем более явно плохие блоки проявятся.

Дополнительная проверка.

Спустя несколько дней вставить в программатор, выставить все настройки ECC, ББ (или загрузить .UDEV) и дамп как описано выше. В [настройках модуля](#) выставить «Не останавливать верификацию при ошибках» и для наглядности можно выставить «Показывать ошибки для целых блоков».

Прочитать ID.

Запустить верификацию и проверить наличие блоков с некорректируемыми ошибками.

NAND сокет x8/x16, 4xCE, 3.30В
 NAND ID ECD584725042 (ECD584725042ECD5), CE1 (*)
 Производитель Samsung
 Название модели K9GAG08U0E
 Шина и напряжение x8 3.30В
 Время доступа 30нс
 Адресные циклы 2/3 (Col/Row)
 Размер страницы 0x2000 (8192)
 Размер spare 0x1B4 (436)
 Страниц в блоке 0x80 (128)
 Размер данных блока 0x100000 (1048576) 1 МиБ
 Размер блока со spare 0x10DA00 (1104384) 1.05 МиБ
 Количество блоков 0x81C (2076)
 Бит в ячейке 2 (MLC)
 Требования к ECC 24/1024 (Bits/CW)
 Кристаллов в чипе 1
 Количество чипов (CE) 1
 Общий размер данных 0x81C00000 2.03 ГиБ
 Общий размер со spare 0x88A7D800 2.13 ГиБ
 NAND ECC коррекция MSTAR_P8K_SP436_CW8_S12L42 (нет ECC ошибок на странице #0)
 NAND таблица плохих блоков Samsung RFS (Блок #0)
 Обнаружение плохих блоков ВКЛ (Spare байт #0 == 0x00)
 Управление плохими блоками Использовать резервную зону
 Запрос плохих блоков...Норма
 Количество плохих блоков: 20
 Блок 0x00D8 (216) адрес 0x0D800000 (0x0E37F000) Плохой блок
 Блок 0x00DF (223) адрес 0x0DF00000 (0x0EAE6000) Плохой блок
 Блок 0x01DA (474) адрес 0x1DA00000 (0x1F33A400) Плохой блок
 Блок 0x01DB (475) адрес 0x1DB00000 (0x1F447E00) Плохой блок
 Блок 0x02B1 (689) адрес 0x2B100000 (0x2D5ABA00) Плохой блок
 Блок 0x02B7 (695) адрес 0x2B700000 (0x2DBFD600) Плохой блок
 Блок 0x02BA (698) адрес 0x2BA00000 (0x2DF26400) Плохой блок
 Блок 0x02BC (700) адрес 0x2BC00000 (0x2E141800) Плохой блок

Блок 0x02BD (701) адрес 0x2BD00000 (0x2E24F200) Плохой блок
Блок 0x02C1 (705) адрес 0x2C100000 (0x2E685A00) Плохой блок
 Блок 0x033D (829) адрес 0x33D00000 (0x3691F200) Плохой блок
 Блок 0x036B (875) адрес 0x36B00000 (0x39991E00) Плохой блок
 Блок 0x0371 (881) адрес 0x37100000 (0x39FE3A00) Плохой блок
 Блок 0x0379 (889) адрес 0x37900000 (0x3A850A00) Плохой блок
 Блок 0x037D (893) адрес 0x37D00000 (0x3AC87200) Плохой блок
 Блок 0x06BE (1726) адрес 0x6BE00000 (0x719DCC00) Плохой блок
Блок 0x06C0 (1728) адрес 0x6C000000 (0x71BF8000) Плохой блок
 Блок 0x06C1 (1729) адрес 0x6C100000 (0x71D05A00) Плохой блок
 Блок 0x0714 (1812) адрес 0x71400000 (0x77470800) Плохой блок
 Блок 0x0819 (2073) адрес 0x81900000 (0x88754A00) Плохой блок

NAND Дамп, 256 байт

```
00000000 46 53 52 5F 53 54 4C 00 01 01 02 01 FF FF FF FF | FSR_STL....ÿÿÿÿ
00000010 FE FF FF FF 00 08 00 12 00 00 01 00 01 00 02 00 | рÿÿÿ.....
00000020 01 00 10 00 00 00 04 00 01 00 00 00 01 00 00 00 | .....
00000030 02 00 D4 00 04 00 01 01 C9 00 0A 00 07 00 C9 00 | ..ô.....é.....é.
00000040 00 48 06 00 80 64 00 00 01 00 00 02 09 00 00 00 | .H...d.....
00000050 02 00 03 00 04 00 05 00 FF FF FF FF FF FF FF FF | .....ÿÿÿÿÿÿÿÿ
00000060 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000070 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000080 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000090 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000A0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000B0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000C0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000D0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000E0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000F0 FF | ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

NAND Dump 'E:\temp\damp\Новая папка\SAMSUNG\D55xx\dados flash tv samsung un32d5500 k9gag08u0e - ic1302\CLEAR\dados

flash tv samsung un32d5500 k9gag08u0e - ic1302_fix_nobads.bin' Analysis, size 0x88A7D800...

NAND Страница 8192+436 (0x21B4), страниц в блоке 128, блок 0x10DA00 (1104384), 0x081C (2076) блоков, IC геометрия

NAND ECC коррекция MSTAR_P8K_SP436_CW8_S12L42

NAND таблица плохих блоков Samsung RFS, Блок 0x07FD (2045)

NAND информация о разделах Samsung RFS, Блок 0x07FE (2046)

NAND LPCH блок 0x07FE (2046) по адресу 0x86AE4C00

NAND LPCH TPCB 0x07FC (2044), годы 1-1

NAND UPCH блок 0x07FD (2045) по адресу 0x869D7200, 1 запись(-и) (Активен)

NAND UPCH TPCB 0x07FC (2044), годы 1-2, инфо 0x0000

NAND RBA 0x07FB-0x0785 (2043-1925), 0x77 (119) блоков

NAND раздел #1, 'part-00-ID1B', блок 0x0 (0), кол-во 0xD6 (214)

NAND раздел #1, адрес 0x00000000 (0x00000000), размер 0x0D600000 (0x0E163C00)

NAND раздел #2, 'part-01-ID1C', блок 0xD6 (214), кол-во 0x100 (256)

NAND раздел #2, адрес 0x0D600000 (0x0E163C00), размер 0x10000000 (0x10DA0000)

NAND раздел #3, 'part-02-ID1D', блок 0x1D6 (470), кол-во 0xD6 (214)

NAND раздел #3, адрес 0x1D600000 (0x1EF03C00), размер 0x0D600000 (0x0E163C00)

NAND раздел #4, 'part-03-ID1E', блок 0x2AC (684), кол-во 0x410 (1040)

NAND раздел #4, адрес 0x2AC00000 (0x2D067800), размер 0x41000000 (0x4475A000)

NAND раздел #5, 'part-04-ID1F', блок 0x6BC (1724), кол-во 0xC8 (200)

NAND раздел #5, адрес 0x6BC00000 (0x717C1800), размер 0x0C800000 (0x0D2A5000)

NAND раздел #6, 'UPCH', блок 0x7FD (2045), кол-во 0x1 (1)

NAND раздел #6, адрес 0x7FD00000 (0x869D7200), размер 0x00100000 (0x0010DA00)

```

NAND раздел #7, 'LPCH', блок 0x7FE (2046), кол-во 0x1 (1)
NAND раздел #7, адрес 0x7FE00000 (0x86AE4C00), размер 0x00100000 (0x0010DA00)
Загружаем список плохих блоков микросхемы...20 плохих блока(-ов)
Записываем новый список плохих блоков...
NAND плохой блок 0x00D8 (216) заменен на 0x07FB (2043)
NAND плохой блок 0x00DF (223) заменен на 0x07FA (2042)
NAND плохой блок 0x01DA (474) заменен на 0x07F9 (2041)
NAND плохой блок 0x01DB (475) заменен на 0x07F8 (2040)
NAND плохой блок 0x02B1 (689) заменен на 0x07F7 (2039)
NAND плохой блок 0x02B7 (695) заменен на 0x07F6 (2038)
NAND плохой блок 0x02BA (698) заменен на 0x07F5 (2037)
NAND плохой блок 0x02BC (700) заменен на 0x07F4 (2036)
NAND плохой блок 0x02BD (701) заменен на 0x07F3 (2035)
NAND плохой блок 0x02C1 (705) заменен на 0x07F2 (2034)
NAND плохой блок 0x033D (829) заменен на 0x07F1 (2033)
NAND плохой блок 0x036B (875) заменен на 0x07F0 (2032)
NAND плохой блок 0x0371 (881) заменен на 0x07EF (2031)
NAND плохой блок 0x0379 (889) заменен на 0x07EE (2030)
NAND плохой блок 0x037D (893) заменен на 0x07ED (2029)
NAND плохой блок 0x06BE (1726) заменен на 0x07EC (2028)
NAND плохой блок 0x06C0 (1728) заменен на 0x07EB (2027)
NAND плохой блок 0x06C1 (1729) заменен на 0x07EA (2026)
NAND плохой блок 0x0714 (1812) заменен на 0x07E9 (2025)
NAND плохой блок 0x0819 (2073) игнорирован
Кодируем ECC в измененных страницах...MSTAR_P8K_SP436_CW8_S12L42
Управление плохими блоками Использовать резервную зону
*****2021.05.18 01:17:56:1756*****
Источник данных Файл 'E:\temp\damp\Новая папка\SAMSUNG\D55xx\dados flash tv samsung un32d5500 k9gag08u0e -
ic1302\CLEAR\dados flash tv samsung un32d5500 k9gag08u0e - ic1302_fix_nobads.bin'...Норма
Пользовательские флаги задачи Асинхронный, ECC, SADDR
Верификация NAND с адреса 0x00000000, размер 0x88A7D800...
Блок 0x0006 (6) адрес 0x00600000 (0x00651C00) ECC, 2 бит(-а) исправлено
Блок 0x00DB (219) адрес 0x00DB0000 (0x0E6A7E00) 2 битовых ошибок в 1 страницах!
Блок 0x00DE (222) адрес 0x00DE0000 (0x0E9D0C00) ECC, 2 бит(-а) исправлено
Блок 0x00E0 (224) адрес 0x00E00000 (0x0EBEC000) ECC, 1 бит(-а) исправлено
Блок 0x01D9 (473) адрес 0x01D90000 (0x1F22CA00) 2 битовых ошибок в 2 страницах!
Блок 0x01DB (475) адрес 0x01DB0000 (0x1F447E00) 1 битовых ошибок в 1 страницах!
Блок 0x01DC (476) адрес 0x01DC0000 (0x1F555800) ECC, 2 бит(-а) исправлено
...
Блок 0x0216 (534) адрес 0x02160000 (0x2326BC00) ECC, 1 бит(-а) исправлено
Блок 0x0218 (536) адрес 0x02180000 (0x23487000) ECC, 2 бит(-а) исправлено
Блок 0x02B1 (689) адрес 0x02B10000 (0x2D5ABA00) 1 битовых ошибок в 1 страницах!
Блок 0x02B7 (695) адрес 0x02B70000 (0x2DBFD600) 3 битовых ошибок в 3 страницах!
Блок 0x02BA (698) адрес 0x02BA0000 (0x2DF26400) 3 битовых ошибок в 3 страницах!
Блок 0x02BC (700) адрес 0x02BC0000 (0x2E141800) 3 битовых ошибок в 3 страницах!
Блок 0x02BD (701) адрес 0x02BD0000 (0x2E24F200) 3 битовых ошибок в 3 страницах!
Блок 0x02BF (703) адрес 0x02BF0000 (0x2E46A600) 6 битовых ошибок в 6 страницах!
Страница 0x1608C (90252), блок 0x02C1 (705) адрес 0x2C118000 (0x2E69EE70) Некорректируемая ECC ошибка!
Блок 0x02C2 (706) адрес 0x02C20000 (0x2E793400) ECC, 1 бит(-а) исправлено
...
Блок 0x0390 (912) адрес 0x03900000 (0x3C08A000) ECC, 3 бит(-а) исправлено
Блок 0x0392 (914) адрес 0x03920000 (0x3C2A5400) ECC, 3 бит(-а) исправлено
Блок 0x0393 (915) адрес 0x03930000 (0x3C3B2E00) ECC, 6 бит(-а) исправлено

```

Страница 0x3600C (221196), блок 0x06C0 (1728) адрес 0x6C018000 (0x71C11470) Некорректируемая ECC ошибка!

Блок 0x06C0 (1728) адрес 0x6C000000 (0x71BF8000) 1 битовых ошибок в 1 страницах!

...

Блок 0x06ED (1773) адрес 0x6ED00000 (0x74B5D200) ECC, 4 бит(-а) исправлено

Блок 0x07E9 (2025) адрес 0x7E900000 (0x854C6A00) Плохой блок

Блок 0x07EA (2026) адрес 0x7EA00000 (0x855D4400) Плохой блок

Блок 0x07EB (2027) адрес 0x7EB00000 (0x856E1E00) Плохой блок

Блок 0x07EC (2028) адрес 0x7EC00000 (0x857EF800) Плохой блок

Блок 0x07ED (2029) адрес 0x7ED00000 (0x858FD200) Плохой блок

Блок 0x07EE (2030) адрес 0x7EE00000 (0x85A0AC00) Плохой блок

Блок 0x07EF (2031) адрес 0x7EF00000 (0x85B18600) Плохой блок

Блок 0x07F0 (2032) адрес 0x7F000000 (0x85C26000) Плохой блок

Блок 0x07F1 (2033) адрес 0x7F100000 (0x85D33A00) Плохой блок

Блок 0x07F2 (2034) адрес 0x7F200000 (0x85E41400) Плохой блок

Блок 0x07F3 (2035) адрес 0x7F300000 (0x85F4EE00) Плохой блок

Блок 0x07F4 (2036) адрес 0x7F400000 (0x8605C800) Плохой блок

Блок 0x07F5 (2037) адрес 0x7F500000 (0x8616A200) Плохой блок

Блок 0x07F6 (2038) адрес 0x7F600000 (0x86277C00) Плохой блок

Блок 0x07F7 (2039) адрес 0x7F700000 (0x86385600) Плохой блок

Блок 0x07F8 (2040) адрес 0x7F800000 (0x86493000) Плохой блок

Блок 0x07F9 (2041) адрес 0x7F900000 (0x865A0A00) Плохой блок

Блок 0x07FA (2042) адрес 0x7FA00000 (0x866AE400) Плохой блок

Блок 0x07FB (2043) адрес 0x7FB00000 (0x867BBE00) Плохой блок

Блок 0x0819 (2073) адрес 0x81900000 (0x88754A00) Плохой блок

ECC проверено в 36768 страницах, 226376 пустых страниц пропущено, 14 пустых кодовых слов, 192 кодовых слов с пустым ECC

Исправлено 992 ECC битовых ошибок, 56 в данных ECC

Обнаружено 2 некорректируемых ECC ошибок в 2 страницах

Общее количество битовых ошибок во время верификации: 25

Завершено за 3 мин. 22 сек. Скорость загрузки кода 10.78 МИБ/сек

В логе присутствуют блоки с некорректируемыми ошибками. Такие блоки далее должны быть помечены как «Плохие блоки» в «Доп. функции». Если их больше одного, то можно перечислить их номера через запятую.

При верификации включен обход ББ с использованием резервной зоны и стоит учесть, что в логе верификации блок 705, в котором присутствуют некорректируемые ошибки уже помечен как ББ. В этом случае за 705 блок у нас выступает 2034 блок. Это выше описано с списке трансляции плохих блоков. Именно его и нужно дополнительно метить как ББ.

Так же можно в [настройках модуля](#) выставить опцию «Показывать исправленные ECC ошибки» и увидеть в логе блоки с особо большим количеством скорректированных ошибок. Это тоже кандидаты на добавление в список плохих блоков, если ошибок более 20-30.

Заново прочитать ID, стереть и записать.

16.2.11 Приложение. Контейнеры. Сборка и запись.

Цель.

Для примера взят dump BIOS от ноутбука и требуется сделать прошивку, чтобы прошивать в SPI flash не затирая ключ Windows.

Подготовка.

Найдя смещение ключа по, dump делится на три региона с размерами кратными сектору. Сектор в SPI является минимальной стираемой частью. Например если это была бы NAND flash, то это будут блоки.

Из дампа, который нужно записать, вырезаются два региона - до Winkey и после, а затем сохраняются отдельными файлами с расширением .bin. Все регионы должны быть кратны минимальному стираемому региону — блоку.

Например:

0x000000-0x41F000 – регион до ключа. Сохранить в отдельный файл.

0x41F000-0x423000 – регион с ключом.

0x423000-0x800000 – регион после ключ. Сохранить в отдельный файл.

Создание проекта.

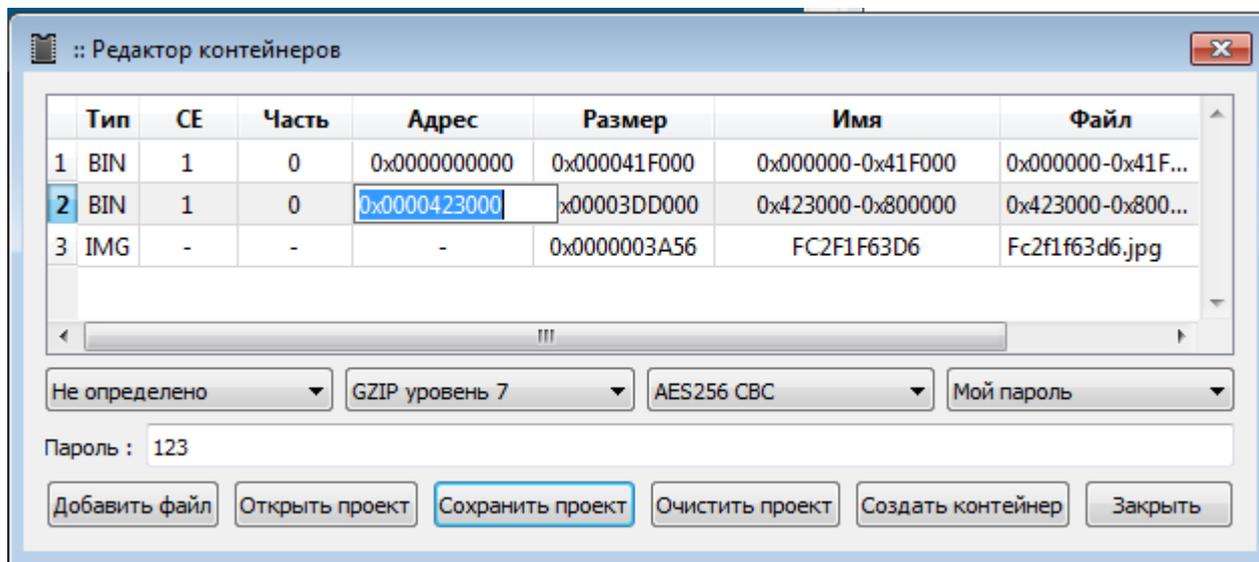
Открыть «Инструменты — Редактор контейнеров».

Нажать «Добавить файл» и выбрать по порядку первый файл региона 0x000000-0x41F000. Значения адреса и расположения в таблице оставить, поскольку совпадает с фактическими.

Нажать еще раз «Добавить файл» и выбрать второй файл региона 0x423000-0x800000.

Два раз щелкнуть по значению «Адреса» второго файла и установить 0x423000.

Далее можно добавить картинку, .udev файл, или текстовый документ, они будут открываться при открывании контейнера.



Тип флеш оставляем для SPI как «Не определено».

Сразу можно выбрать уровень сжатия для архиватора, чтобы файл занимал меньше места. На картинке выбрано «GZIP уровень 7».

Для защиты можно зашифровать и установить пароль.

Как все настройки выполнены и части добавлены, нужно сохранить проект .urpj нажав соответствующую кнопку.

После сохранения проекта можно его изменить, или уже создать контейнер соответствующей кнопкой.

Работа с контейнером.

Запустить контейнер можно либо через «Файл — Открыть контейнер», либо по кнопке «Открыть файл для записи», выбрав тип файла .urkg.

Для записи регионов из контейнера, нужно прочитать ID флеш, и обычными кнопками нажать стереть и записать. Все операции будут проведены только с регионами установленных в контейнере файлов, а регион 0x41F000-0x423000 не затронется.

После окончания работы нужно закрыть контейнер через в меню файл. Либо контейнер сам закроется, если выбрать другой файл для записи.

Важно знать.

Чтобы избежать ошибок, все части файлов и проект нужно размещать в одной папке.

При открытии контейнера с добавленным .udev, автоматически применяются его параметры. Например устанавливается режим доступа «Разделы» с включенными в него списком разделов. Рабочая папка с разделами должна находиться рядом с файлом контейнера .urkg.

При открытии контейнера содержащего .txt файл, его содержимое будет выведено в лог. Это может быть описание.

Если в контейнере была добавлена картинка, то она откроется параллельно окну программы. Это может быть иллюстрация или схема.

16.2.12 Приложение. Использование LOGGER и анализ протокола.

В разработке...

16.3 Распиновки и схемы.

Актуальная информация находится по [адресу](#)

UFPI IO распиновка разъема:

PIN01 - IO00

PIN02 - IO01

...

PIN24 - IO23

PIN25 - ID1 (Модуль. Резистор между PIN25 и GND)

PIN26 - ID2 (Напряжение. Резистор между PIN26 и GND)

PIN27 - VCC/VREF

PIN28 - VUSB (+5V)

PIN29 - GND

PIN30 - GND

ID1 (Модуль) значения резисторов:

JTAG - 0R

NAND - 1K

SPI Flash - 2K

EEPROM I2C - 3K

EEPROM SPI - 4K

EEPROM mWire - 5K

Serial NAND - 7K1

1-Wire - 9K1

OneNAND - 20K

NOR - 24K

SD/eMMC - 30K

E-Mate - 33K

UART - 51K

LOGGER - 68K

BDM - 75K

ID2 (Напряжение) значения резисторов:

VREF - 0R или 8K2 (Внешнее питание)

3V3 - 1K

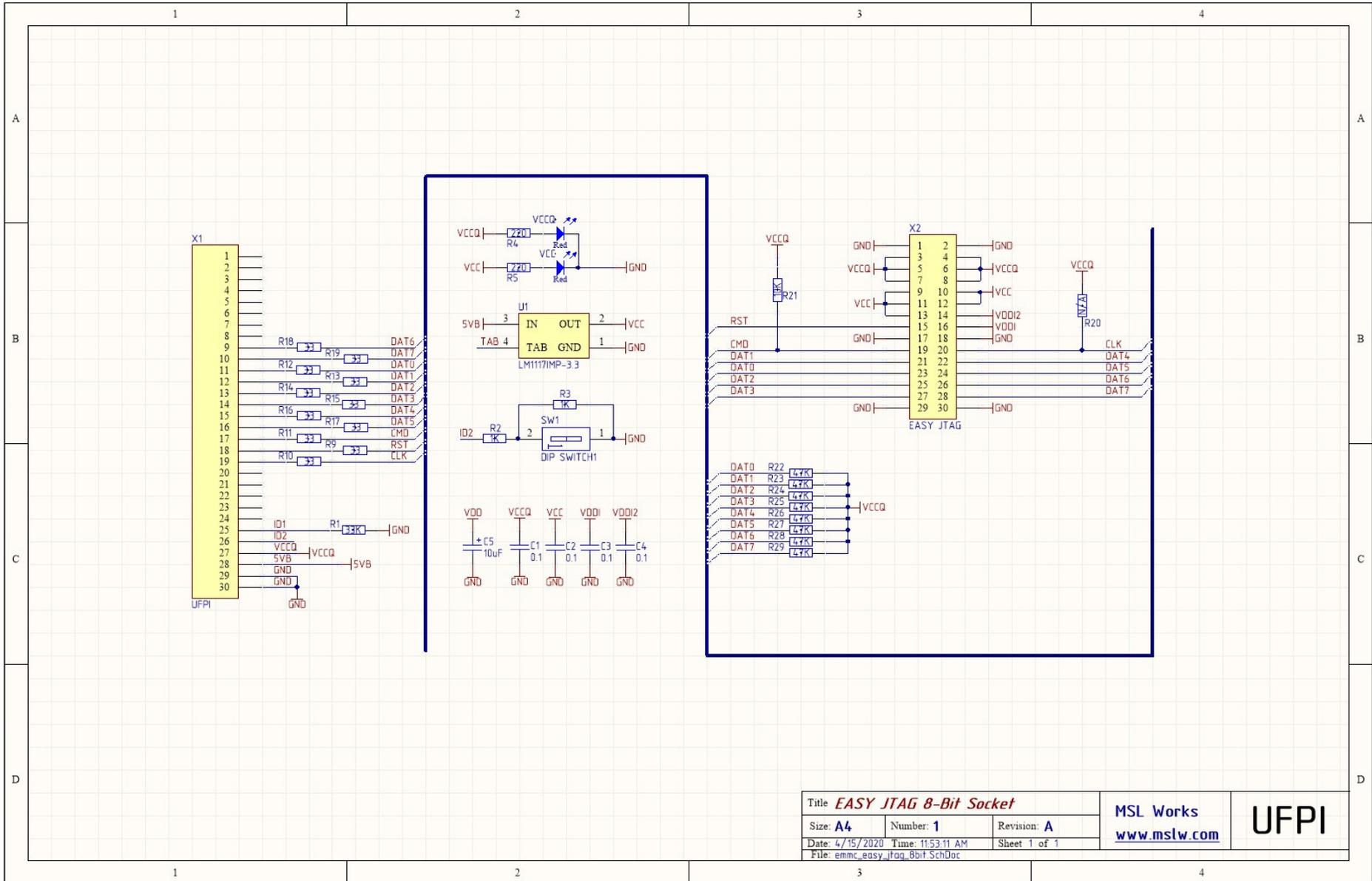
1V8 - 2K

2V6 - 3K

5V0 - 9K1



16.3.1 eMMC EASY JTAG 8-Bit Socket schematics

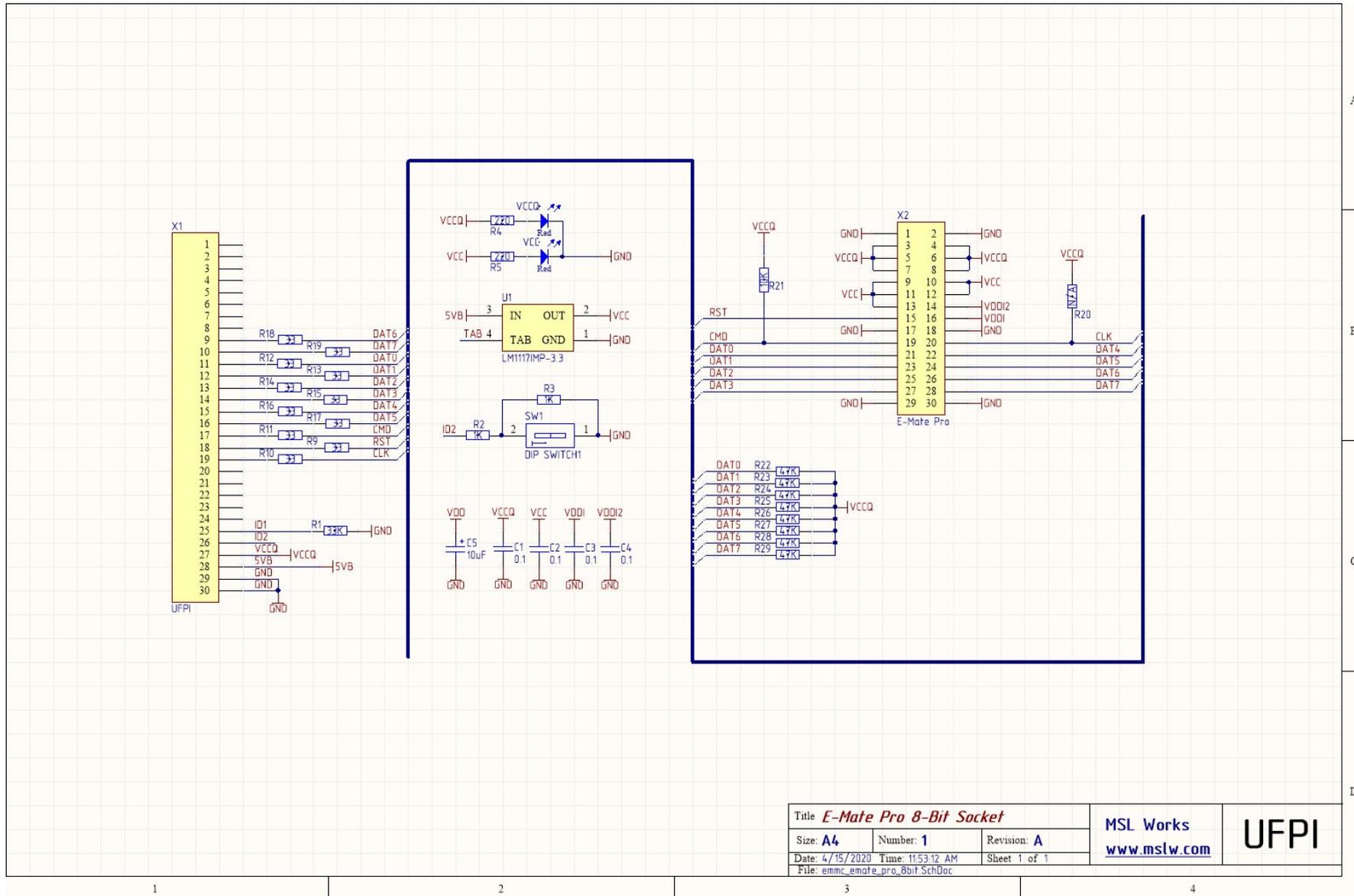


Title <i>EASY JTAG 8-Bit Socket</i>		
Size: A4	Number: 1	Revision: A
Date: 4/15/2020	Time: 11:53:11 AM	Sheet 1 of 1
File: emmc_easy_jtag_8bit_SchDoc		

MSL Works
www.mslw.com

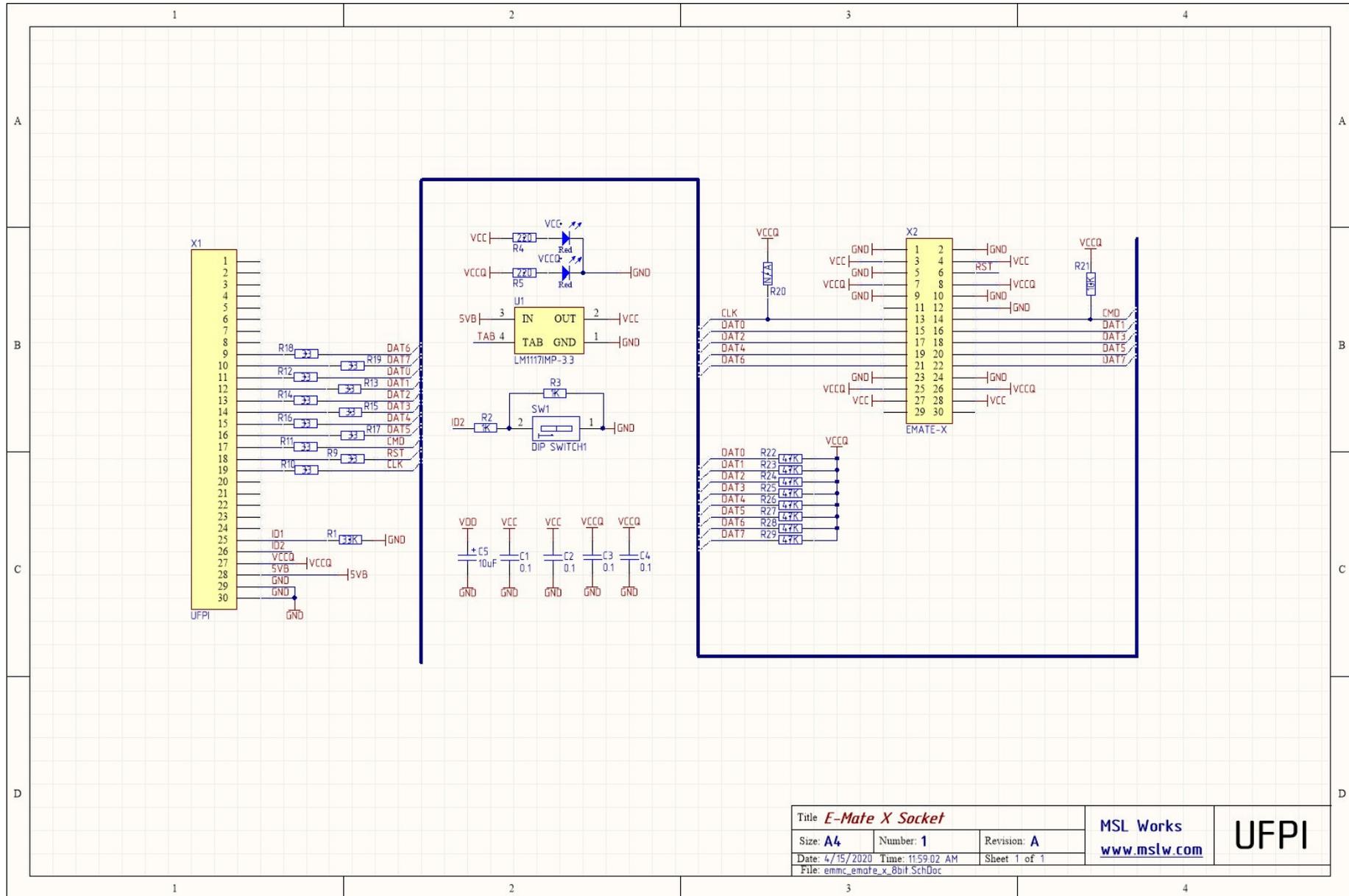
UFPI

16.3.2 eMMC E-Mate Pro 8-Bit Socket schematics



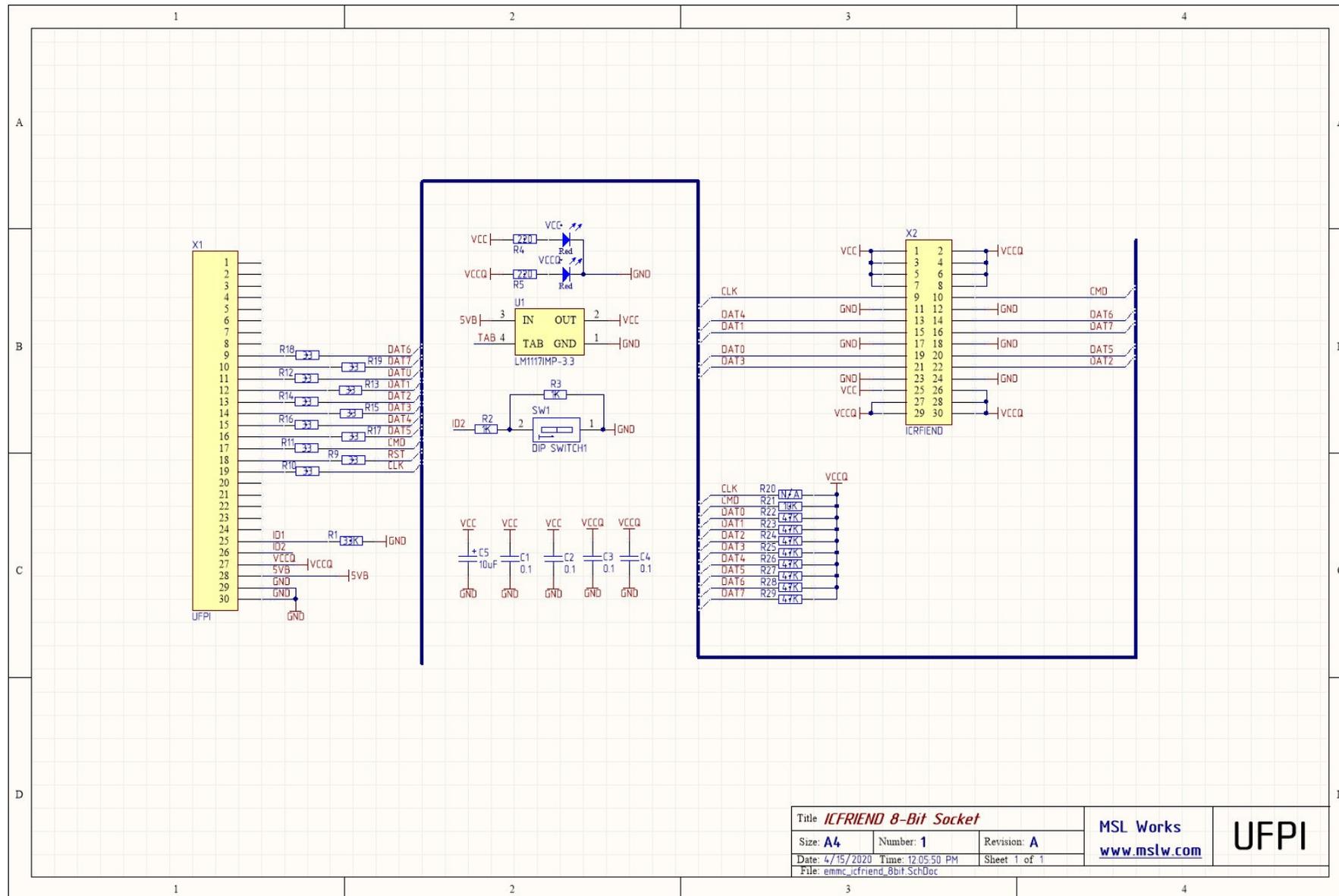
Title <i>E-Mate Pro 8-Bit Socket</i>			MSL Works www.mslw.com	UFPI
Size: A4	Number: 1	Revision: A		
Date: 4/15/2020	Time: 11:53:12 AM	Sheet 1 of 1		
File: emmc_emate_pro_8bit SchDoc				

16.3.3 eMMC E-Mate X 8-Bit Socket schematics

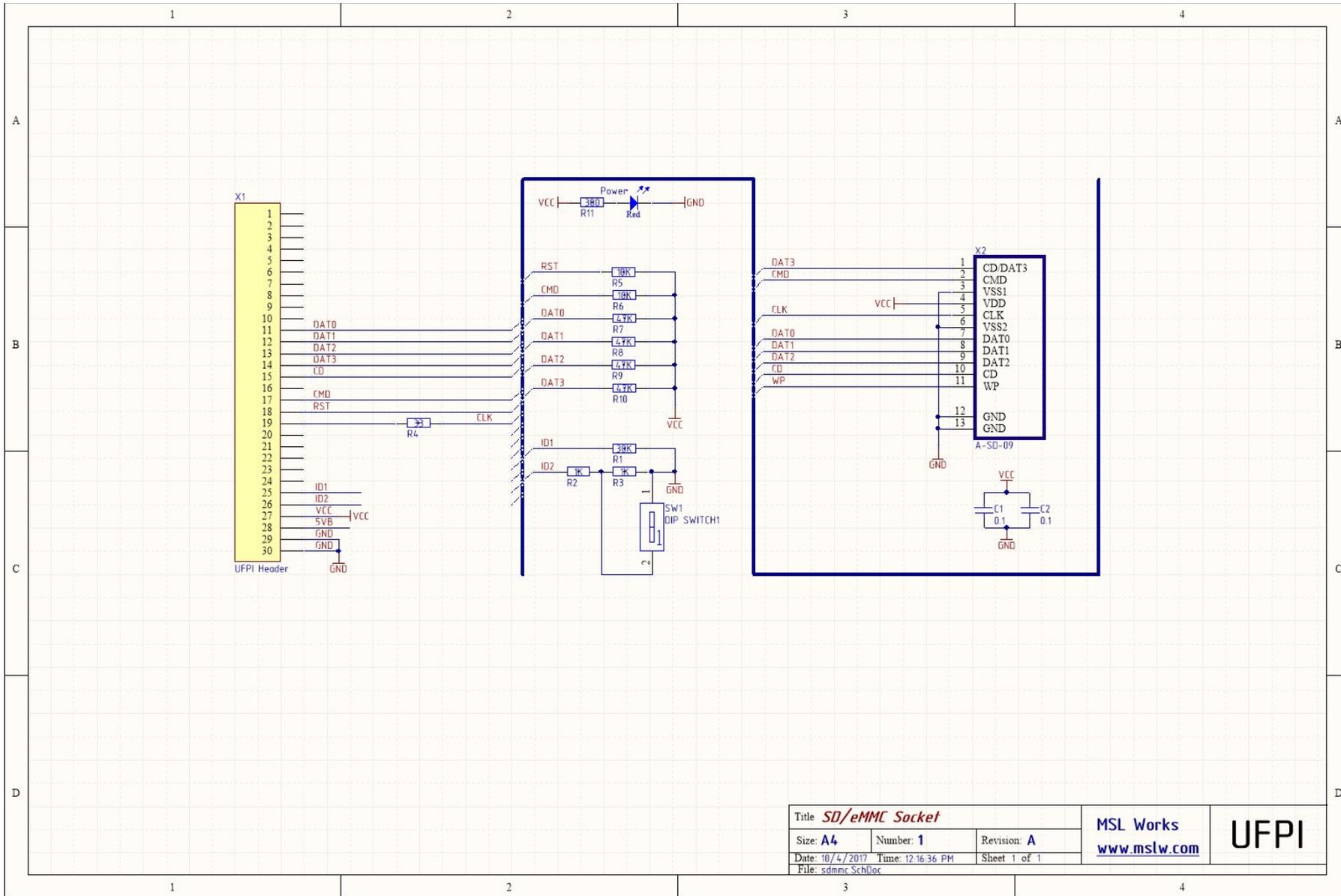


Title <i>E-Mate X Socket</i>			MSL Works www.mslw.com	UFPI
Size: A4	Number: 1	Revision: A		
Date: 4/15/2020	Time: 11:59:02 AM	Sheet 1 of 1		
File: emmc_emate_x_8bit_SchDoc				

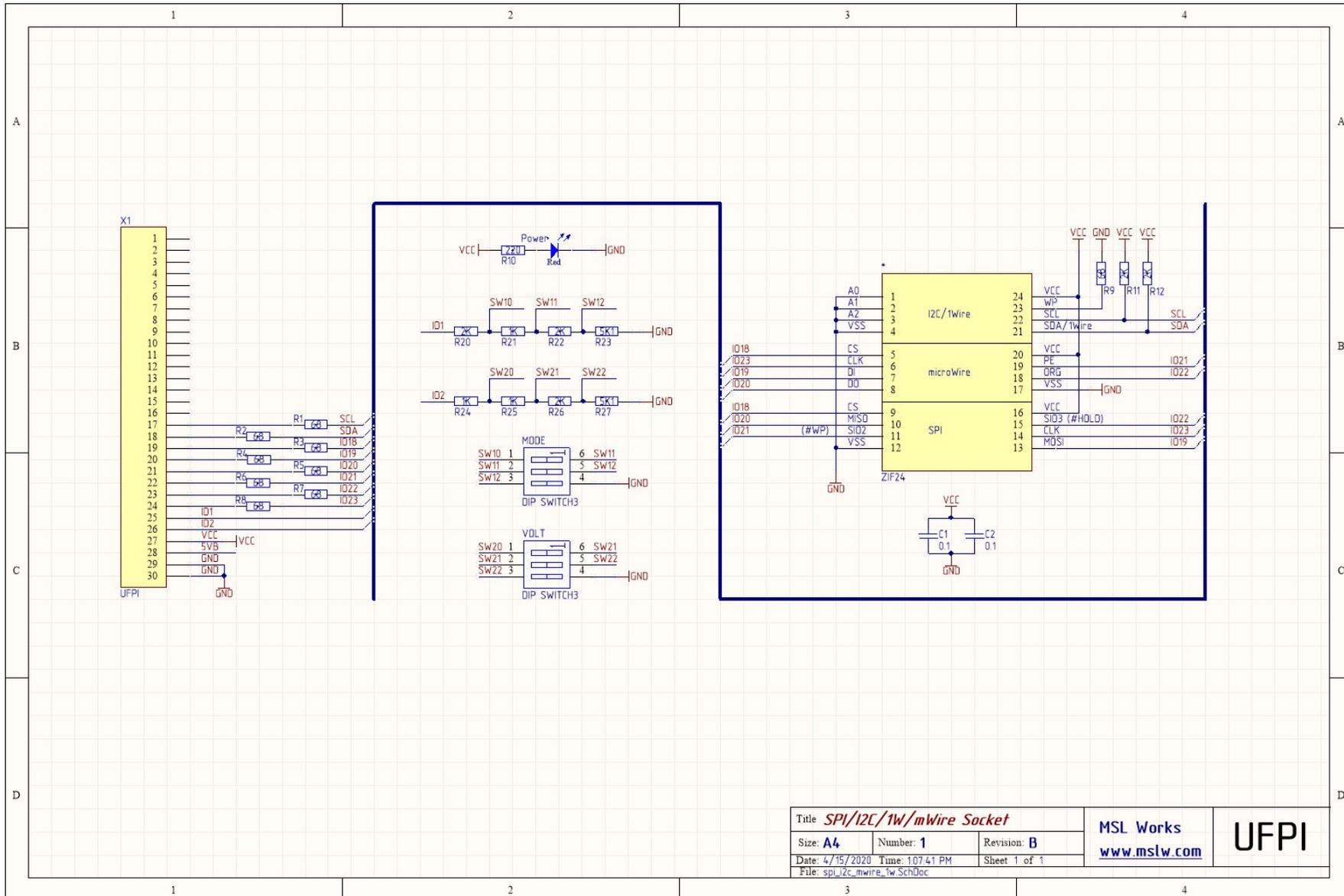
16.3.4 eMMC ICFRIEND 8-Bit Socket schematics



16.3.5 SD/eMMC Socket schematics

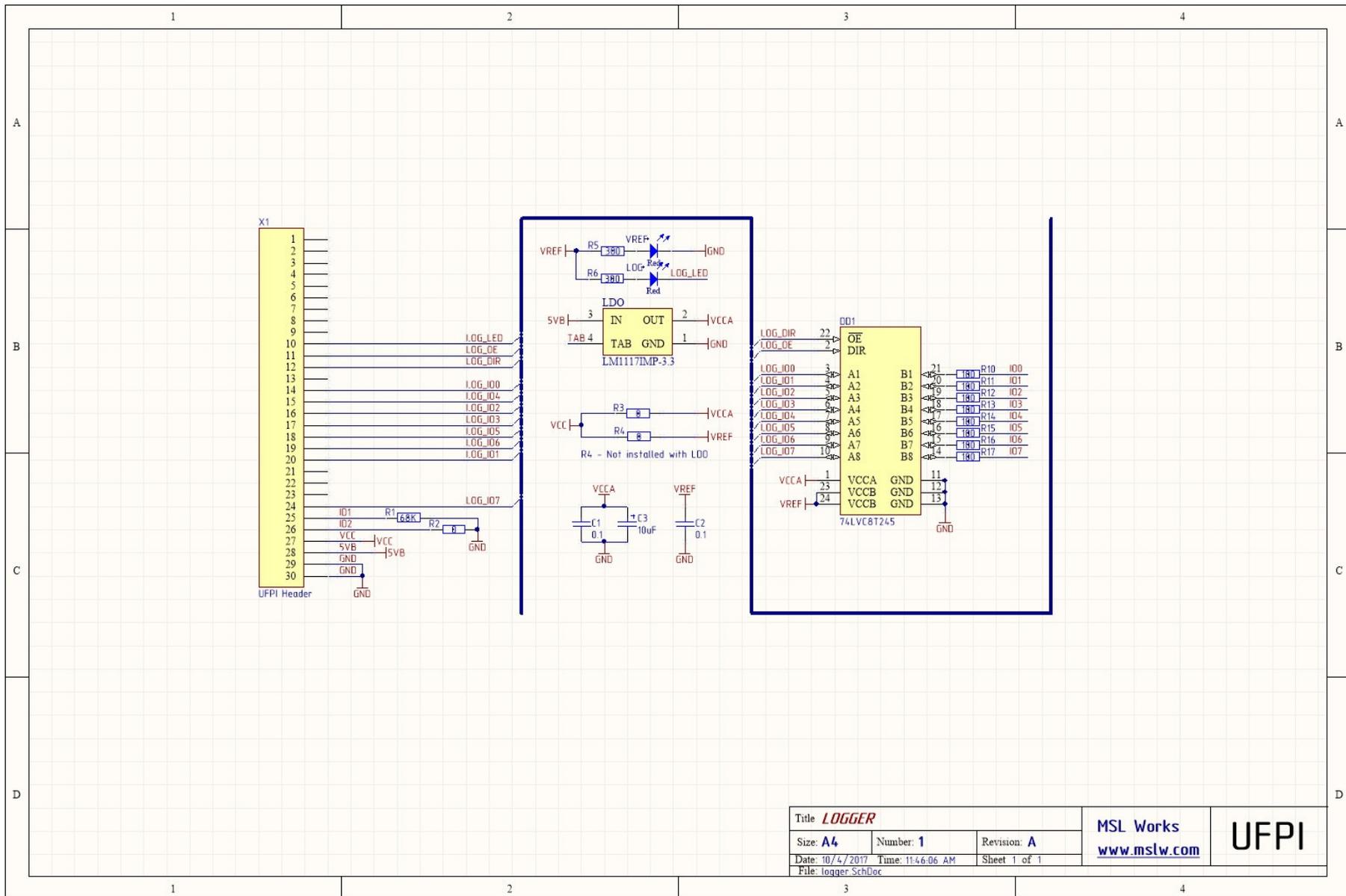


16.3.6 SPI/I2C/mWire/1Wire DIP24 Socket schematics



Title <i>SPI/I2C/1W/mWire Socket</i>			MSL Works www.mslw.com	UFPI
Size: A4	Number: 1	Revision: B		
Date: 4/15/2020	Time: 107:41 PM	Sheet 1 of 1		
File: spi_i2c_mwire_1wSchDoc				

16.3.7 LOGGER Socket schematics

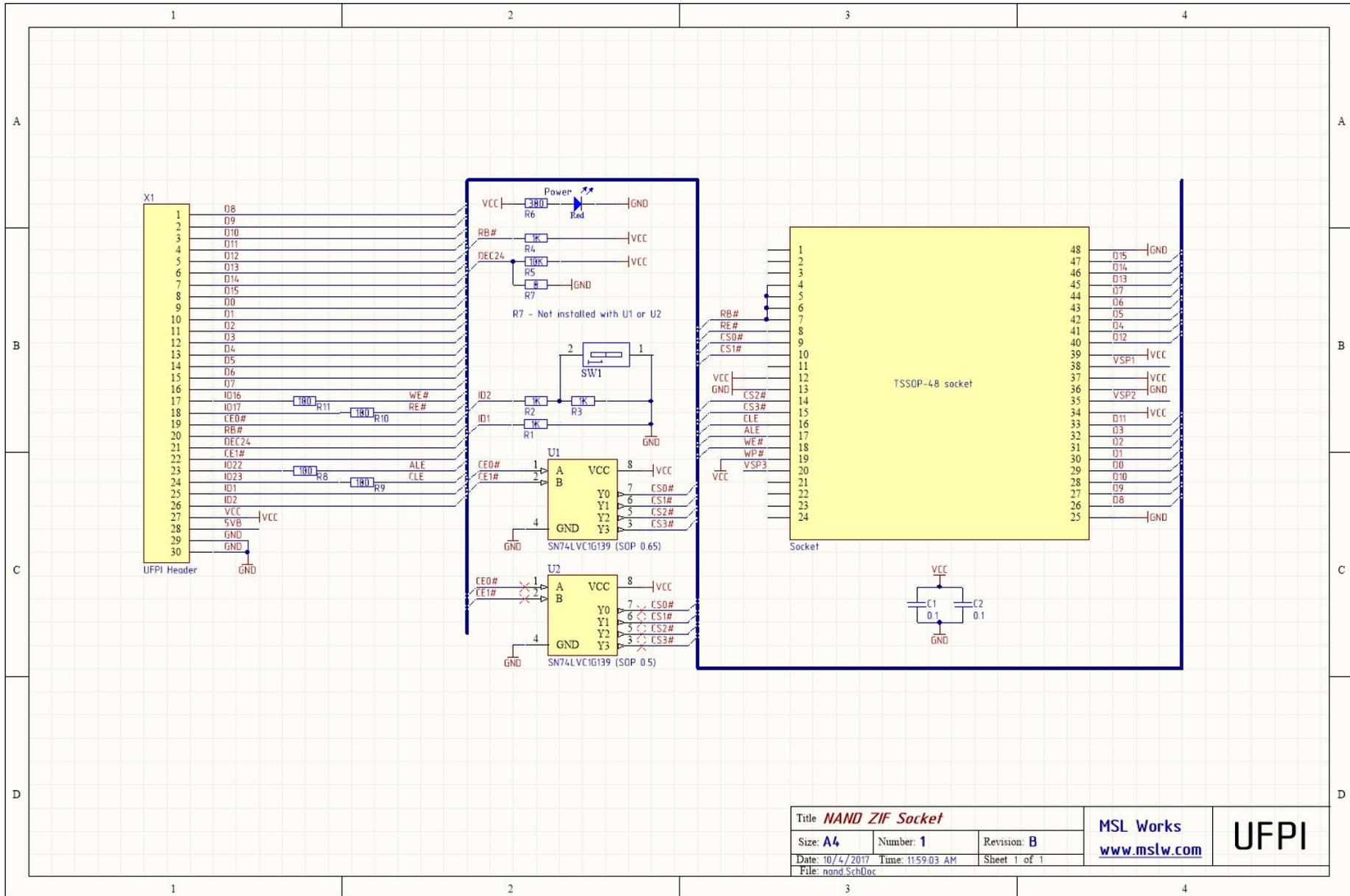


Title LOGGER		
Size: A4	Number: 1	Revision: A
Date: 10/4/2017	Time: 11:46:06 AM	Sheet 1 of 1
File: logger_SchDoc		

MSL Works
www.mslw.com

UFPI

16.3.8 NAND Socket schematics

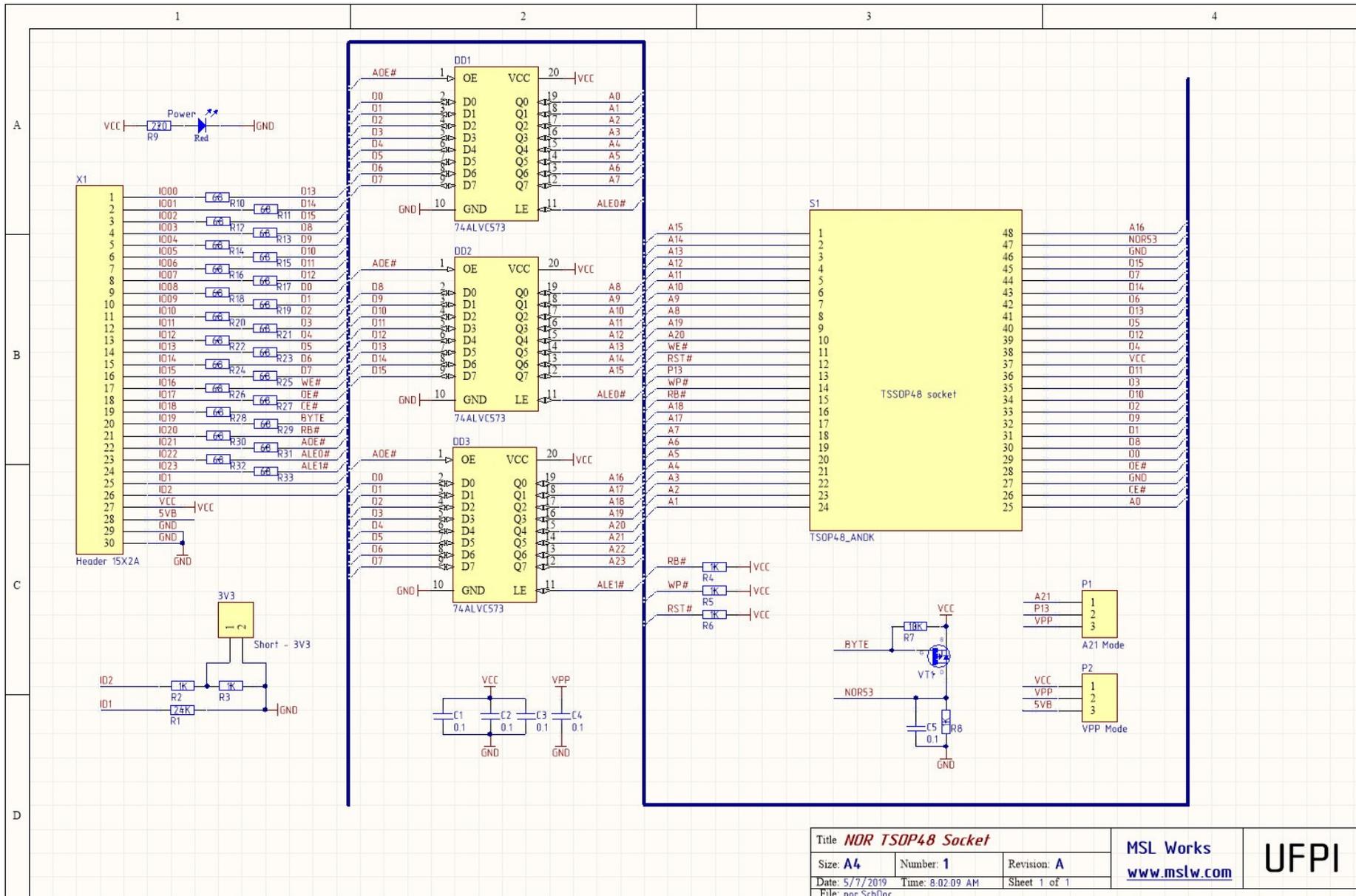


Title NAND ZIF Socket		
Size: A4	Number: 1	Revision: B
Date: 10/4/2017	Time: 11:59:03 AM	Sheet 1 of 1
File: nand_SchDoc		

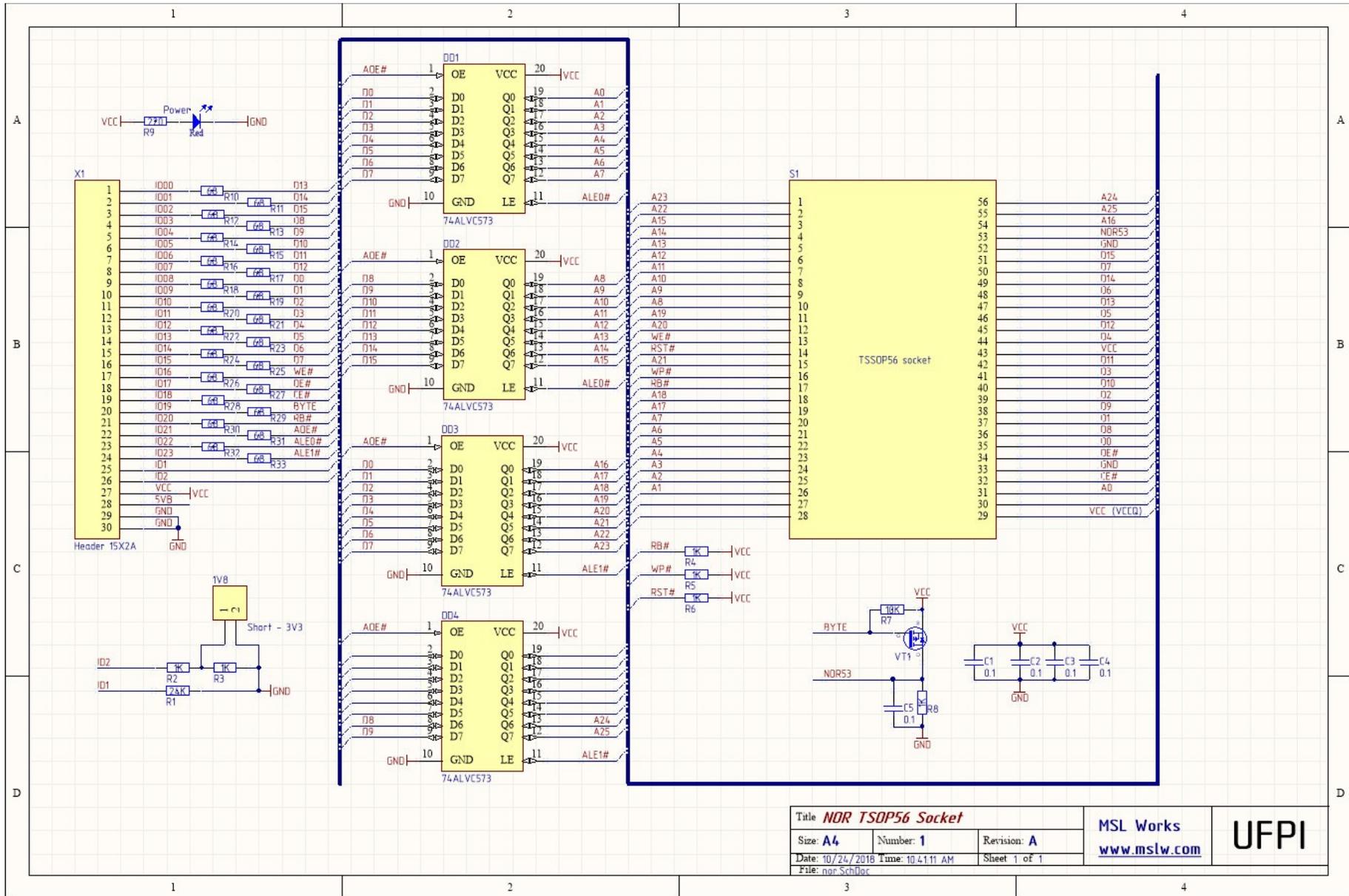
MSL Works
www.mslw.com

UFPI

16.3.9 NOR TSOP48 Socket schematics



16.3.10 NOR TSOP56 Socket schematics

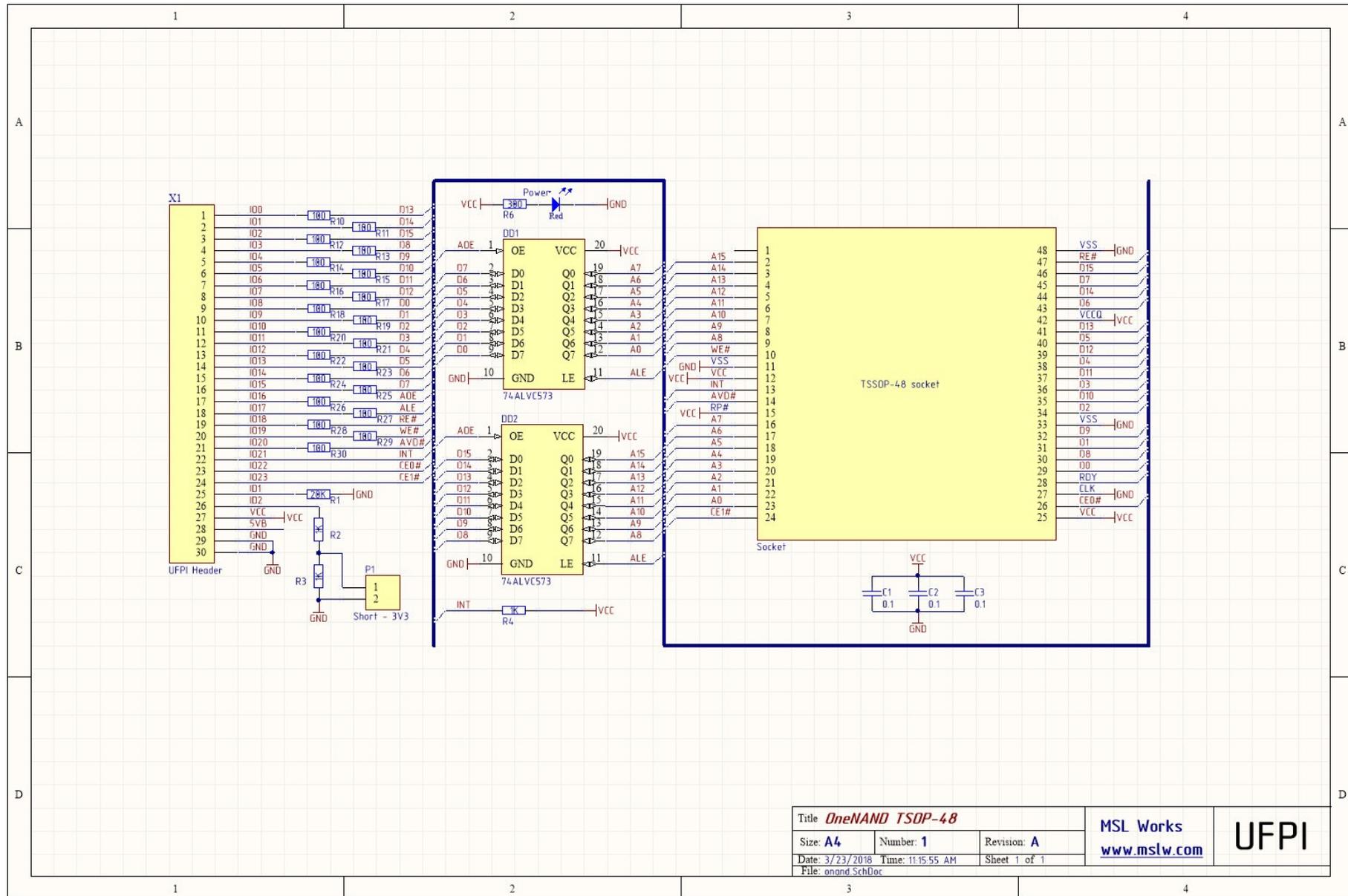


Title NOR TSOP56 Socket		
Size: A4	Number: 1	Revision: A
Date: 10/24/2018	Time: 10:41:11 AM	Sheet 1 of 1
File: nor_SchDoc		

MSL Works
www.mslw.com

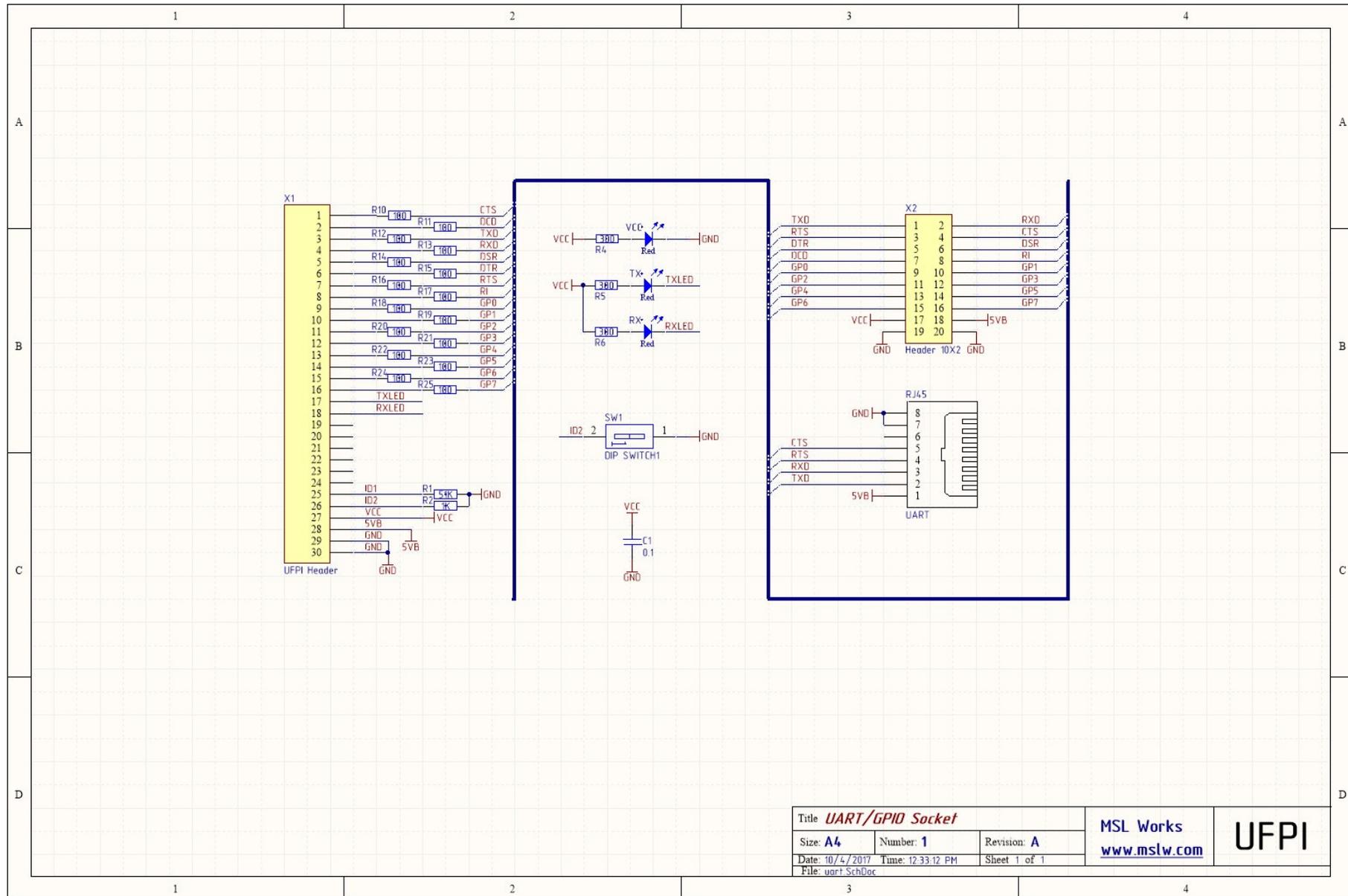
UFPI

16.3.11 OneNAND Socket schematics



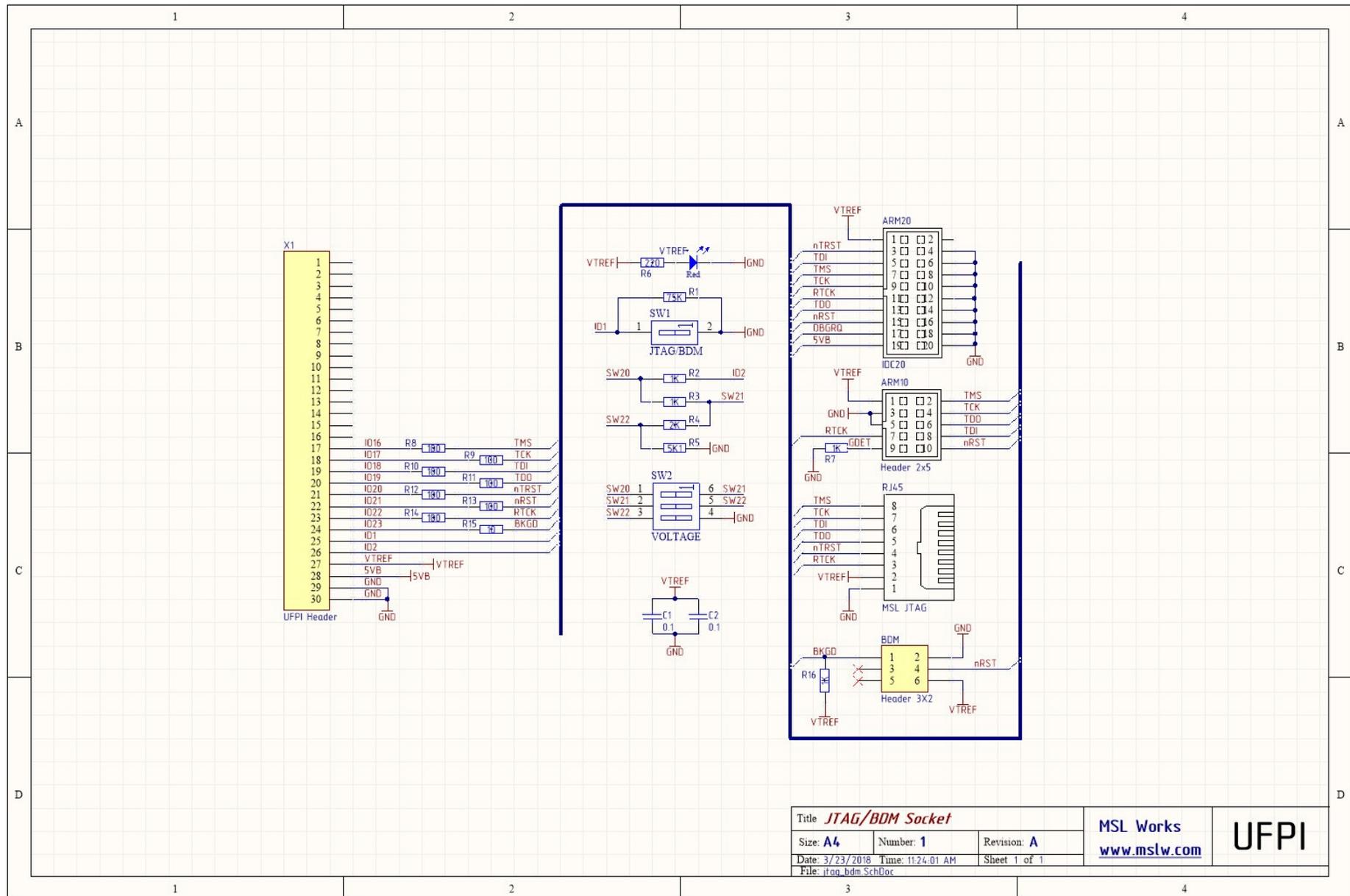
Title OneNAND TSSOP-48			MSL Works www.mslw.com	UFPI
Size: A4	Number: 1	Revision: A		
Date: 3/23/2018	Time: 11:15:55 AM	Sheet 1 of 1		
File: onand.SchDoc				

16.3.12 UART/GPIO Socket schematics



Title UART/GPIO Socket			MSL Works www.mslw.com	UFPI
Size: A4	Number: 1	Revision: A		
Date: 10/4/2017	Time: 12:33:12 PM	Sheet 1 of 1		
File: uart.SchDoc				

16.3.13 JTAG/BDM Socket schematics



Title <i>JTAG/BDM Socket</i>			MSL Works www.mslw.com	UFPI
Size: A4	Number: 1	Revision: A		
Date: 3/23/2018	Time: 11:24:01 AM	Sheet 1 of 1		
File: jtag_bdm SchDoc				