

Микроконтроллер BE-U1000

Руководство пользователя

АО «БАЙКАЛ ЭЛЕКТРОНИКС», ИНН: 7707767484

АО «БАЙКАЛ ЭЛЕКТРОНИКС» оставляет за собой право вносить любые изменения в настоящий документ без дополнительного уведомления

Содержание

1 ВВЕДЕНИЕ	3
2 УСЛОВНЫЕ ОБОЗНАЧЕНИЯ, ТЕРМИНЫ И СОКРАЩЕНИЯ	3
3 НАЧАЛЬНАЯ ЗАГРУЗКА	5
3.1 Общее описание.....	5
3.2 Режимы загрузки.....	5
3.2.1 Режим загрузки «eFlash»	6
3.2.2 Режим загрузки «UART»	6
3.2.3 Режим загрузки «QSPI»	7
3.2.4 Режим загрузки «USB»	7
3.2.5 Режим загрузки «JTAG Ext».....	9
3.2.6 Режимы загрузки «UART MP», «USB MP»	9
3.2.7 Режим загрузки «Multi»	10
3.2.8 Режим загрузки «JTAG Int».....	10
3.2.9 Режим загрузки «USB CDC»	11
3.2.10 Режим загрузки «Factory reset»	11
3.2.11 Режим загрузки «Multi USB»	11
3.1 Интерфейс командной строки загрузчика (BootROM CLI).....	11
3.1.1 Параметры.....	12
3.1.2 Функции	12
3.1.3 Расширенные функции (m)	13
3.2 Состояние МК после начальной загрузки	15
4 SDK	17
4.1 Описание SDK.....	17
4.2 Подготовка SDK	18
4.2.1 Загрузка SDK	18

4.2.2 Распаковка SDK	18
4.2.2.1 OC Windows.....	18
4.2.2.2 OC Linux.....	18
5 ПОДГОТОВКА ПРОЕКТА ПРОГРАММЫ	19
6 СБОРКА ОБРАЗА ПРОГРАММЫ	20
6.1 СБОРКА ОБРАЗА ДЛЯ ИСПОЛНЕНИЯ ИЗ eFLASH	20
6.2 СБОРКА ОБРАЗА ДЛЯ ИСПОЛНЕНИЯ ИЗ TCM	21
7 ЗАПИСЬ И ИСПОЛНЕНИЕ ОБРАЗА ПРОГРАММЫ В МК.....	22
7.1 OC WINDOWS	22
7.1.1 Работа с TCM в режиме DFU	22
7.1.2 Работа с eFlash в режиме DFU	23
7.1.3 Работа с QSPI Flash в режиме DFU	24
7.1.4 Работа с TCM в режиме UART	25
7.1.5 Работа с eFlash в режиме UART	27
7.1.6 Работа с QSPI Flash в режиме UART.....	30
7.1.7 Работа с TCM в режиме JTAG	32
7.1.8 Работа с eFlash в режиме JTAG.....	34
7.2 OC LINUX	36
7.2.1 Работа с TCM в режиме DFU	36
7.2.2 Работа с eFlash в режиме DFU	37
7.2.3 Работа с QSPI Flash в режиме DFU	38
7.2.4 Работа с TCM в режиме UART	39
7.2.5 Работа с eFlash в режиме UART	40
7.2.6 Работа с QSPI Flash в режиме UART.....	42
7.2.7 Работа с TCM в режиме JTAG	43
7.2.8 Работа с eFlash в режиме JTAG.....	45
ИСТОРИЯ ИЗМЕНЕНИЙ.....	47
КОНТАКТНЫЕ ДАННЫЕ	48

1 Введение

Данный документ содержит следующую информацию о работе с микроконтроллером BE-U1000:

- [Начальная загрузка](#): общее описание, описание режимов загрузки, описание интерфейса командной строки загрузчика, состояние МК после начальной загрузки.
- [SDK](#): описание, подготовка к работе.
- [Подготовка проекта программы](#).
- [Сборка образа программы](#).
- [Запись и исполнение образа программы в МК](#) образа в память МК для ОС Linux и Windows.

2 Условные обозначения, термины и сокращения

В документе используются условные обозначения, сокращения и термины, указанные в таблицах ниже.

Табл. 2-1 – Условные обозначения

Обозначение	Расшифровка
[Текст]	Переменные данные в командах. Примеры: [путь к файлу], [имя образа]. Данные вводятся без символов [].
<Текст>	Клавиша на клавиатуре или сочетание клавиш. Примеры: <D>, <Ctrl+V>.
Текст	Элемент графического интерфейса.
Текст → Текст	Последовательность переходов между элементами графического интерфейса.
[SDK_Path]	Путь до директории с SDK.

Табл. 2-2 – Сокращения и термины

Обозначение	Расшифровка
МК	Микроконтроллер BE-U1000.
ОС	Операционная система, установленная на управляемом ПК.
ПК	Управляющий персональный компьютер.
Страповый вывод	Выход МК, предназначенный для выбора режима работы при начальной загрузке.
BootROM	Программа начального загрузчика, расположенная в ROM0.
BootROM CLI	Интерфейс командной строки BootROM
CDC	Communications Device Class (класс коммуникационного устройства) – технология передачи информации по виртуальному COM-порту поверх USB.
Core 0/1/2	Ядро 0/1/2 микроконтроллера BE-U1000.
CSR	Служебные регистры ядер.
DFU	Device Firmware Update (обновление прошивки устройства) – протокол работы с памятью МК через USB, реализованный в BootROM в соответствии со стандартом DFU v1.1.
eFlash	Встроенная флэш-память МК. Содержит две области – Main и NVR.

Обозначение	Расшифровка
eFlash Main	Основная область eFlash, предназначенная для хранения программ или данных пользователя.
GDB	GNU Debugger – кроссплатформенный программный отладчик.
JTAG	Joint Test Action Group – стандартизованный аппаратный интерфейс (IEEE 1149.1), предназначенный для тестирования и отладки электронных устройств.
NRESET	Вывод (а также сигнал), предназначенный для перезагрузки МК.
NVR / eFlash NVR	Область eFlash MK, предназначенная для хранения конфигурационных и пользовательских параметров, параметров MK и BootROM.
NVR1	Служебный сектор eFlash для хранения заводских настроек. Пользователь ограничен в возможности записи и стирания данного сектора.
NVR2	Служебный сектор eFlash для хранения параметров BootROM. Пользователь может ограничить запись и стирание данного сектора.
NVR3, NVR4	Служебный сектор eFlash для хранения параметров BootROM.
OpenOCD	Open On-Chip Debugger – программный отладчик, предназначенный для взаимодействия с MK.
PD	PullDown – состояние вывода, при котором он подключен к земле через встроенный подтягивающий резистор.
PU	PullUp – состояние вывода, при котором он подключен к питанию через встроенный подтягивающий резистор.
QSPI	Quad SPI – расширение классического SPI, позволяющее передавать данные по четырём линиям данных параллельно вместо одной, что значительно увеличивает пропускную способность последовательного интерфейса.
QSPI Flash	Тип флэш-памяти, которая использует интерфейс QSPI для передачи данных.
REPL	Read, Evaluate, Print, Loop (чтение, вычисление, вывод, цикл) – интерактивная среда разработки, позволяющая поочередно выполнять команды или фрагменты кода.
ROM	Read-Only Memory (постоянная память, ПЗУ) – тип памяти, которая хранит неизменяемые данные и инструкции.
ROM0	ROM ядра Core 0 – область памяти с адресом 0x4000_0000, из которой происходит начальная загрузка MK. Для получения более подробной информации см. документ BE-U1000. Техническая спецификация .
SPI	Serial Peripheral Interface (последовательный периферийный интерфейс) – стандарт последовательной синхронной передачи данных в режиме полного дуплекса, предназначенный для обеспечения простого и недорогого высокоскоростного сопряжения микроконтроллеров и периферии.
TCM	Tightly Coupled Memory (тесно связанная память) – память типа SRAM на MK BE-U1000, расположенная рядом с ядром для более быстрого доступа.
UART	Universal Asynchronous Receiver-Transmitter (Универсальный Асинхронный Приемопередатчик) – это протокол последовательной связи, используемый для обмена данными между электронными устройствами без синхронизирующего тактового сигнала, с настройкой общей битовой скорости и добавлением служебных битов (стартовый, стоповый) для кадрирования данных.
XMODEM	Стандартный протокол передачи данных, реализованный в BootROM.

3 Начальная загрузка

3.1 Общее описание

Загрузка МК начинается после подачи питания и снятия сигнала NRESET. После этого Core 0 начинает исполнять код начального загрузчика (BootROM) из ROM0.

В процессе исполнения BootROM считывается конфигурация страповых выводов и параметров конфигурации из eFlash NVR, после чего выбирается соответствующий режим загрузки.

При чтении параметров конфигурации из eFlash NVR, BootROM проверяет наличие специальных кодовых слов. В случае их отсутствия выполняется полная очистка eFlash (за исключением NVR1) и производится начальная разметка eFlash NVR. Очистка и начальная разметка eFlash NVR занимает около 20 мс.

Ниже приведено краткое описание режимов загрузки.

3.2 Режимы загрузки

В данном разделе описаны алгоритмы работы МК с заводскими настройками в различных режимах.

Табл. 3-1 – Режимы загрузки МК

Режим	Строповые выводы		Описание
	DBG_SEL PC[8]	MODE[2:0] PC[4],PB[10],PA[10]	
eFlash	0	000	Исполнение программы из eFlash.
UART	0	001	Запись образа программы по UART0 в TCM по протоколу XMODEM с последующим ее исполнением из TCM или вход в BootROM CLI .
QSPI	0	010	Копирование образа программы из QSPI Flash по порту QSPI1 в TCM с последующим ее исполнением из TCM.
USB	0	011	Запись образа программы в TCM, eFlash или QSPI Flash по USB с помощью протоколов DFU или CDC с возможностью исполнения записанной программы или вход в BootROM CLI .
JTAG Ext	0	100	Режим работы с внешним JTAG-отладчиком.
UART MP	0	101	Режим работы со встроенным интерпретатором MicroPython через UART0.
Multi	0	110	Поочередные попытки запуска в режимах: 1. eFlash 2. QSPI 3. UART
Factory reset	0	111	Сброс параметров eFlash NVR до заводских и переход в режим USB
USB CDC	1	001	Циклические попытки загрузки: - BootROM CLI через USB по протоколу CDC - UART (BootROM CLI или XMODEM)
JTAG Int	1	100	Режим работы со встроенным USB JTAG-отладчиком на ядре Core 0.

Режим	Страповые выводы		Описание
	DBG_SEL PC[8]	MODE[2:0] PC[4],PB[10],PA[10]	
USB MP	1	101	Режим работы со встроенным интерпретатором MicroPython через USB CDC.
MULTI USB	1	110	Попытки запуска в режимах: 1. eFlash 2. QSPI 3. Циклические попытки загрузки: - USB CDC - UART



Для комбинаций DBG_SEL и MODE, не указанных в данной таблице, режим работы не определен.

3.2.1 Режим загрузки «eFlash»

Данный режим предназначен для исполнения программ, записанных в eFlash, непосредственно из eFlash.

BootROM проверяет наличие кодового слова 0x0B0F5262 по адресу 0xa0000020 (eFlash). Если кодовое слово присутствует – BootROM передает управление программе, записанной в eFlash, по адресу 0xa0000030.

Если кодовое слово не найдено – переход в режим «USB».

3.2.2 Режим загрузки «UART»

Данный режим предназначен для записи программ в память МК по протоколу XMODEM.

Алгоритм работы в режиме «UART»:

1. BootROM инициализирует UART0 со следующими параметрами:

Скорость передачи (Baud Rate)	115200
Управление потоком (Flow Control)	Нет
Биты четности (Parity)	Нет
Биты данных (Data Bits)	8
Стоп-биты (Stop Bits)	1

2. BootROM ожидает поступления данных по протоколу XMODEM, либо получения символа `d` (код 0x64). В случае получения символа `d` BootROM переходит в [BootROM CLI](#).
3. В случае поступления данных по XMODEM, BootROM записывает эти данные в TCM по адресу 0x40010000. После успешного принятия данных осуществляется проверка наличия кодового слова 0x0B10D522 по адресу 0x40010020. Если слово присутствует – BootROM передает управление принятой программе в TCM по адресу 0x40010030.

Если произошла ошибка при передаче, или кодовое слово не найдено – BootROM продолжает ожидание поступления данных или получения символа `d` (возврат к п. 2).

3.2.3 Режим загрузки «QSPI»

Данный режим предназначен для загрузки программ из QSPI Flash в TCM и их исполнения из TCM.

Алгоритм работы в режиме «QSPI»:

1. Проверка наличия кодового слова `0x0B10D522` в QSPI Flash (на порту QSPI1) со смещением `0x20`. В QSPI Flash образ программы располагается по нулевому смещению.
2. Если кодовое слово присутствует, то выполняется копирование образа программы из QSPI Flash в TCM (`0x40010000`) и передача управления в TCM по адресу `0x40010030`.
Максимальный размер образа указывается по смещению `0x2C` и не должен превышать `0x20000` (128 кБ), т.к. старшие 32 кБ TCM используются BootROM. После получения управления программа может использовать все 160 кБ TCM.
3. Если кодовое слово отсутствует или произошла ошибка чтения – переход в режим UART.



BootROM взаимодействует с QSPI Flash в режиме *Standard SPI Frame Format + National Microwire Frame Format*.

3.2.4 Режим загрузки «USB»

Данный режим предназначен для записи образа программы в TCM, eFlash или QSPI Flash по USB с помощью протокола DFU с возможностью исполнения записанной программы. Возможно использование BootROM CLI по протоколу CDC.

В данном режиме MK по USB определяется как составное USB-устройство с функциями DFU и CDC.

CDC используется для входа в BootROM CLI при передаче четыре раза подряд символа `\d`.

DFU используется для записи программ или изменения настроек MK. Доступны альтернативы (alt), перечисленные ниже.

Alt=0: Запись в TCM с исполнением

1. Выполняется запись образа программы в TCM (`0x40010000`)
2. Проверяется наличие кодового слова `0x0B10D522` по адресу `0x40010020`.
3. Если кодовое слово присутствует, то выполняется передача управления в TCM по адресу `0x40010030`.

Максимальный размер образа не должен превышать `0x20000` (128 кБ), т.к. старшие 32 кБ TCMB используются BootROM. После получения управления программа может использовать все 160 кБ TCM.

Alt=1: Перезапуск с выбором режима загрузки

1. Принимается 1 байт
2. Перезапуск МК с выбором режима загрузки в соответствии со значением принятого байта:
Биты [2] – [0] – эквивалентно установке выводов PC[4], PB[10], PA[10] (см. [Табл. 3-1](#))
Бит [3]:
0x0 – загрузка в строгом соответствии с установленными битами [4], [2]–[0] принятого байта.
0x1 – при загрузке в режиме USB CDC не будет происходить переход в режим UART.
Бит [4] – эквивалентно установке вывода PC8 (см. [Табл. 3-1](#)).

Выбранный режим действует на один перезапуск, после следующей перезагрузки МК режим загрузки определяется страповыми выводами.

Alt=2: Запись в eFlash Main с последующим исполнением из eFlash

1. Запись образа программы (не более 256 кБ) в eFlash (0xa0000000)
2. Перезапуск МК в режиме загрузки «EFLASH».

Выбранный режим действует на один перезапуск, после следующей перезагрузки МК режим загрузки определяется страповыми выводами.

Alt=3: Запись в eFlash Main

1. Выполняется запись образа программы (не более 256 кБ) в eFlash (0xa0000000)
2. После записи МК продолжает работу в текущем режиме.

Alt=4: Запись в eFlash NVR2

1. Выполняется запись данных (не более 1 кБ) в NVR2
2. После успешного принятия данных МК продолжает работу в текущем режиме.

Alt=5: Запись в eFlash NVR3, NVR4

1. Выполняется запись данных (не более 2 кБ) в NVR3 и NVR4
2. После принятия данных МК продолжает работу в текущем режиме.

Alt=6: Запись в QSPI Flash

1. Выполняется запись данных в QSPI1 Flash по смещению 0x0. В случае записи образа программы доступно не более 16 МБ.
2. После записи МК продолжает работу в текущем режиме.

Alt=7: Запись в QSPI Flash со смещением

1. Выполняется запись образа программы (не более 14 МБ) в QSPI1 Flash по смещению 0x200000
2. После записи МК продолжает работу в текущем режиме.

3.2.5 Режим загрузки «JTAG Ext»

Данный режим предназначен для отладки программ, запускаемых на ядрах Core 0, Core 1 и Core 2, с использованием внешнего JTAG-отладчика.

BootROM выполняет настройку выводов PA0 – PA4 для работы в режиме JTAG и ожидает подключения внешнего отладчика.

3.2.6 Режимы загрузки «UART MP», «USB MP»

Старт в режиме «MicroPython» начинается с поиска на QSPI1 по смещению 0x200000 образа файловой системы в формате FAT, содержащего дополнительные модули Python. В образе также может присутствовать автоматически исполняемый при старте MicroPython файл `_boot.py`. Если образ файловой системы или файл `_boot.py` отсутствуют, запускается REPL MicroPython.

Встроенный MicroPython совместим с MicroPython Remote Shell.

Строка приглашения REPL MicroPython:

```
MicroPython 92e3b1df4 on 2025-04-04; bupy with rv32-cpu
Type "help()" for more information.
>>>
```

Вывод подсказки:

```
>>> help()
Welcome to MicroPython!

For online docs please visit http://docs.micropython.org/

Control commands:
CTRL-A -- on a blank line, enter raw REPL mode
CTRL-B -- on a blank line, enter normal REPL mode
CTRL-C -- interrupt a running program
CTRL-D -- on a blank line, exit or do a soft reset
CTRL-E -- on a blank line, enter paste mode

For further help on a specific object, type help(obj)
>>>
```

Вывод списка встроенных модулей:

```
>>> help('modules')
__main__ builtins gc struct
_thread bupy io sys
array collections micropython uctypes
binascii errno os vfs
Plus any modules on the filesystem
>>>
```

Вывод содержания модуля на примере `bupy`:

```
>>> import bupy
>>> help(bupy)
object <module 'bupy'> is of type module
    __name__ -- bupy
    Flash -- <class 'Flash'>
    Param -- <class 'Param'>
    Pin -- <class 'Pin'>
    freq -- <function>
    repl -- <function>
    quit -- <function>
    cfg -- <>
    mem8 -- <8-bit memory>
    mem16 -- <16-bit memory>
    mem32 -- <32-bit memory>
>>>
```

Чтение каталога в смонтированном FAT-разделе:

```
>>> import os
>>> os.listdir()
['_boot.py']
>>>
```

3.2.7 Режим загрузки «Multi»

В данном режиме осуществляются попытки поочередной загрузки в различных режимах.

1. Загрузка в режиме «EFLASH». Если кодовое слово не найдено – попытка загрузки в режиме «QSPI».
2. Загрузка в режиме «QSPI». Если кодовое слово не найдено – попытка загрузки в режиме «UART».
3. Загрузка в режиме «UART». Ожидание поступления данных по протоколу XMODEM, либо получения символа `d`.

3.2.8 Режим загрузки «JTAG Int»

Данный режим предназначен для отладки программ, запускаемых на ядрах Core 0 и Core 1. При этом Core 2 выполняет функции встроенного адаптера USB-JTAG.

В данном режиме MK по USB определяется как составное USB-устройство с функциями JTAG и CDC.

Для функции JTAG используется протокол, совместимый с ESP JTAG.

Функция CDC позволяет работать в BootROM CLI. Вход выполняется при передаче четыре раза подряд символа `d`. Возможен перевод функции CDC в режим выгрузки трассы с ядра Core 0 или Core 1.

3.2.9 Режим загрузки «USB CDC»

Данный режим предназначен для доступа к BootROM CLI через USB по протоколу CDC.

Алгоритм работы в режиме «USB CDC»:

- Проверка наличия подключения по USB-кабелю. Если USB-кабель подключен – ожидание ввода символа `d` в течение 5 секунд.

Если символ `d` получен – подключение к [BootROM CLI](#) по протоколу CDC.

Если символ `d` не получен в течение 5 секунд – ожидание получения данных по XMODEM через USB CDC.

- Если USB-кабель не подключен, или не началась передача данных по XMODEM через USB CDC в течение 32 секунд, или во время передачи по XMODEM через USB CDC возникли ошибки – ожидание получения данных по UART: ожидается либо ввод символа `d`, либо передача данных по XMODEM через UART0.

Если символ `d` получен – подключение к [BootROM CLI](#) по UART.

Если символ `d` не получен в течение 5 секунд и не началась передача по XMODEM – переход к п.1.

3.2.10 Режим загрузки «Factory reset»

В данном режиме выполняется очистка eFlash Main, NVR3, NVR4.

3.2.11 Режим загрузки «Multi USB»

В данном режиме осуществляются попытки поочередной загрузки в различных режимах.

- Загрузка в режиме «EFLASH». Если кодовое слово не найдено – переход в режим «QSPI».
- Загрузка в режиме «QSPI». Если кодовое слово не найдено – переход режим «USB CDC».
- Загрузка в режиме «USB CDC». Если USB-кабель не подключен, или не началась передача данных по XMODEM через USB CDC в течение 32 секунд, или во время передачи по XMODEM через USB CDC возникли ошибки – переход в режим «UART».
- Загрузка в режиме «UART». Если символ `d` не получен в течение 5 секунд и не началась передача по XMODEM – переход к п.3.

3.1 Интерфейс командной строки загрузчика (BootROM CLI)

В режимах загрузки UART или USB можно перейти в интерфейс командной строки загрузчика BootROM (далее по тексту – BootROM CLI). Для входа в BootROM CLI необходимо в консоль передать символ `d` (для режима UART или USB CDC – 1 раз, для режима USB – 4 раза).

Типичный вид UART-консоли после загрузки МК во время ожидания начала передачи данных по XMODEM (символы `C` добавляются один раз в секунду):

CC

Сообщение о входе в BootROM CLI:

```
BU ROM[v1.1.0 (2025-04-04)_2c05769d]  
@
```

BootROM CLI работает по следующему принципу: задаются значения параметров, после чего они могут быть использованы при выполнении функции.

Ниже описаны команды BootROM CLI.

3.1.1 Параметры

p0, p1, p2, p3 - Изменяемые параметры.

Могут быть использованы функциями, указанными ниже. Значения параметров вводятся через разделитель : в шестнадцатеричном формате.

Пример установки параметров **p0, p1, p2, p3**: в значения **40010000, 100, 1, 2** соответственно:
40010000:100:1:2

Пример установки параметров **p0, p1** в значения **40010000** и **100** соответственно:
40010000:100

Пример установки параметра **p3** в значение **100** без изменения значений параметров **p0, p1** и **p2**:

:::100

Пример ввода параметров и вызова функции «**m**» одной командой:

a0000000:22800:0:100m

3.1.2 Функции

Функции предназначены для совершения различных операций в BootROM CLI. Некоторые функции для выполнения требуют наличия определенных данных, записанных в параметры **p0, p1, p2, p3**.

Ниже представлено краткое описание функций BootROM CLI.

g – Выполнение функции по адресу

p0 – Адрес выполнения функции

h – Справка

i – Вывод в консоль значений параметров p0, p1, p2, p3

l – Загрузка образа памяти с медианосителя

p0 – Адрес

p1 – Размер

p2 – Смещение на медианосителе

p3 – Выбор медианосителя: 0 – QSPI Flash, 3 – XMODEM, 101 – eFlash, 102 – eFlash NVR,
7 – /dev/null

o – Выход из BootROM CLI без ошибки

p – Вывод в консоль дампа из памяти с использованием значений, записанных в p0, p1, p2

p0 – Адрес памяти

p1 – Размер дампа в байтах

p2 – Способ представления данных: 1 – 32-х битное слово, 0 – 8-ми битное слово

p3 – Не используется

q – Выход из BootROM CLI с ошибкой EIO

Вывод ошибки активируется, если значение p0 не равно нулю. По умолчанию отключено для USB.

p0 = 0 – эквивалентно функции o

r – Чтение и вывод в консоль 32-х битного слова из адреса, записанного в p0

Функция может использоваться для чтения информации из памяти или регистров, которые имеют отображение в адресное пространство.

v – Сохранение образа памяти на медианоситель

p0 – Адрес памяти

p1 – Размер (диапазон от адреса до указываемого числа)

p2 – Смещение на медианосителе

p3 – Выбор медианосителя: 0 – QSPI Flash, 3 – XMODEM, 101 – eFlash, 102 – eFlash NVR, 7 – /dev/null

w – Запись 32-х битного слова из p1 в адрес, записанный в p0

Функция может использоваться для записи информации в память или регистры, которые имеют отображение в адресное пространство.

x – Сохранение X-Modem-потока на медианоситель

p2 – Смещение на медианосителе

p3 – Выбор медианосителя: 0 – QSPI Flash, 3 – XMODEM, 101 – eFlash, 102 – eFlash NVR, 7 – /dev/null.

3.1.3 Расширенные функции (m)

Расширенные функции вызываются при вводе m.

Данная функция использует p0, p1, p2 в качестве вспомогательных параметров, в которых хранятся значения.

Все действия определяются параметром p3 и описаны ниже.

0 – Справка

1 – Информация о параметрах

2 – Дамп параметров. Зависит от значения p2.

0 – Загруженных

1 – Из eFlash (кроме NVR)

2 – Из NVR

3 – Справка

4 – Вывод версии загрузчика

5 – Выбор режима загрузки. Зависит от значения p2. Выбранный режим будет активирован после выхода из BootROM CLI с помощью команды o или q.

Биты [2] – [0] – эквивалентно установке выводов PC[4], PB[10], PA[10] (см. [Табл. 3-1](#))

Бит [3]:

0x0 – загрузка в строгом соответствии с установленными битами [4], [2]–[0] принятого байта

0x1 – при загрузке в режиме USB CDC не будет происходить переход в режим UART

Бит [4] – эквивалентно установке вывода PC8 (см. [Табл. 3-1](#))

10 – Вывод параметра. Зависит от значения p0.

p0: 0x20000 + ID параметра.

11 – Установка параметра. Зависит от значений p0 и p1.

p0: 0x20000 + ID параметра.

p1: значение параметра.

22 – Выполнение инструкции fence.i (ассемблерная инструкция очистки кэша инструкций).

100 – Настройка таймингов eFlash для частоты, задаваемой в p2 (в Гц):

0 – применяется значение из данных BootROM (только если не было изменено при помощи регистров CRU).

[n] – применяется введенное шестнадцатеричное значение. Пример: для частоты 25 МГц необходимо ввести 17D7840

101 – Очистка eFlash Main.

102 – Выборочная очистка eFlash. Зависит от значений p0, p1, p2.

p0: адрес первого сектора размером 1 кБ (начиная с нуля).

p1: адрес последнего сектора размером 1 кБ (p0 и p1 образуют диапазон очистки).

p2: 0 – eFlash Main, 1 – eFlash NVR.

3.2 Состояние МК после начальной загрузки

Состояния МК в момент передачи управления программе указаны в таблицах ниже.

Табл. 3-2 – Состояния МК после начальной загрузки в режимах eFlash, UART, QSPI, USB, JTAG Ext

Элемент	Режим загрузки				
	EFLASH	UART	QSPI	USB	JTAG EXT
PLL, МГц	Float	Float	Float	100	Float
CCLK, МГц	25	25	25	100	25
PCLK0, МГц	25	25	25	25	25
PCLK1, МГц	25	25	25	25	25
PCLK2, МГц	25	25	25	25	25
HCLK, МГц	25	25	25	100	25
UART0	OFF/ON	ON	OFF	OFF	OFF
QSPI1, МГц	OFF	1,5625	1,5625	1,5625	OFF
USB	OFF/HS	OFF	OFF	HS	OFF
Тайминги eFlash, МГц	50	50	50	50	50
PA0	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down
PA1	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down
PA2	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down
PA3	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down
PA4	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down
PA5	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down
PA6	AF 1	AF 1	IN Pull Down	IN Pull Down	IN Pull Down
PA7	AF 1	AF 1	IN Pull Down	IN Pull Down	IN Pull Down
PB0	IN Pull Down	AF 2	AF 2	AF 2	IN Pull Down
PB1	IN Pull Down	AF 2	AF 2	AF 2	IN Pull Down
PB2	IN Pull Down	AF 2	AF 2	AF 2	IN Pull Down
PB3	IN Pull Down	AF 2	AF 2	AF 2	IN Pull Down
PB4	IN Pull Down	AF 2	AF 2	AF 2	IN Pull Down
PB5	IN Pull Down	AF 2	AF 2	AF 2	IN Pull Down
PB6	IN Pull Down	IN_NOPULL	IN_NOPULL	IN_NOPULL	IN Pull Down
PB7	IN Pull Down	IN_NOPULL	IN_NOPULL	IN_NOPULL	IN Pull Down

Табл. 3-3 – Состояния МК после начальной загрузки в режимах UART MP, Multi, USB CDC, JTAG Int, USB MP

Элемент	Режим загрузки				
	UART MP	MULTI	USB CDC	JTAG INT	USB MP
PLL, МГц	Float	Float/100	100	100	100
CCLK, МГц	25	25/100	100	100	100
PCLK0, МГц	25	25	25	25	25
PCLK1, МГц	25	25	25	25	25
PCLK2, МГц	25	25	25	25	25
HCLK, МГц	25	25/100	100	100	100
UART0	ON	ON	OFF/ON	OFF	OFF
QSPI1, МГц	1,5625	1,5625	OFF	OFF	1,5625
USB	OFF	OFF/HS	HS	HS	HS

Элемент	Режим загрузки				
	UART MP	MULTI	USB CDC	JTAG INT	USB MP
Тайминги eFlash, МГц	50	50	50	50	50
PA0	IN Pull Down	AF 4	IN Pull Down	IN Pull Down	IN Pull Down
PA1	IN Pull Down	AF 4	IN Pull Down	IN Pull Down	IN Pull Down
PA2	IN Pull Down	AF 4	IN Pull Down	IN Pull Down	IN Pull Down
PA3	IN Pull Down	AF 4	IN Pull Down	IN Pull Down	IN Pull Down
PA4	IN Pull Down	AF 4	IN Pull Down	IN Pull Down	IN Pull Down
PA5	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down
PA6	AF 1	AF 1	AF 1	IN Pull Down	IN Pull Down
PA7	AF 1	AF 1	AF 1	IN Pull Down	IN Pull Down
PB0	AF 2	AF 2	IN Pull Down	IN Pull Down	AF 2
PB1	AF 2	AF 2	IN Pull Down	IN Pull Down	AF 2
PB2	AF 2	AF 2	IN Pull Down	IN Pull Down	AF 2
PB3	AF 2	AF 2	IN Pull Down	IN Pull Down	AF 2
PB4	AF 2	AF 2	IN Pull Down	IN Pull Down	AF 2
PB5	AF 2	AF 2	IN Pull Down	IN Pull Down	AF 2
PB6	IN_NOPULL	AF 1	IN Pull Down	IN Pull Down	IN_NOPULL
PB7	IN_NOPULL	AF 1	IN Pull Down	IN Pull Down	IN_NOPULL



1. В режимах «eFlash» и «USB CDC» UART0 инициализируется, если не подключён кабель USB.
2. Если ранее была выполнена запись программы в QSPI Flash, то при загрузке МК в режиме «UART» или «USB», будет инициализирован QSPI1.
3. В режиме «EFLASH» клоки и USB инициализируются, если не удалось загрузить образ с eFlash Main.
4. В режиме «MULTI» клоки и USB инициализируются, если не удалось загрузить образ с eFlash Main и QSPI1.

4 SDK

4.1 Описание SDK

SDK (Software Development Kit) – набор программных модулей и утилит, предназначенный для разработки программы МК.

SDK может использоваться в ОС Windows 10/11 или Linux – выбор сборки доступен при загрузке.

В состав SDK входят:

- BASIS – общая библиотека поддержки архитектуры RISC-V микроконтроллера BE-U1000.
- HAL – библиотека драйверов периферийных устройств.
- BSP – библиотеки поддержки отладочных плат.
- Шаблон проекта программы.
- Примеры программ.
- Система сборки образов программ.
- Средства кросс-компиляции (toolchain) и отладки программ.
- ОС FreeRTOS, портированная для архитектуры RISC-V микроконтроллеров Baikal-U.
- Библиотека tinyUSB.

Файловая структура SDK:

└── BSP	– Директория с библиотеками поддержки отладочных плат.
└── Drivers	– Директория с драйверами для МК.
└── BASIS	– Библиотека поддержки архитектуры RISC-V.
└── HAL	– Библиотека драйверов периферийных устройств.
└── Middlewares	– Директория с ОС FreeRTOS, а также сторонними драйверами.
└── Projects	– Директория с шаблонами и примерами ПО.
└── _template	– Шаблон проекта программы.
└── HAL_examples	– Примеры программ.
└── Tools	– Директория вспомогательных утилит.
└── build	– Система сборки образов программ.
└── OpenOCD	– Отладчик OpenOCD.
└── scripts	– Вспомогательные скрипты
└── svd	– Директория с файлами описания регистров и периферийных устройств МК
└── toolchain	– Средства кросс-компиляции (GCC) для архитектуры RISC-V.
└── riscv32...	– Утилита Make
└── xpack...	

4.2 Подготовка SDK

4.2.1 Загрузка SDK

SDK может использоваться на компьютере с операционной системой Linux или Windows.

Для загрузки SDK необходимо выполнить следующие действия:

1. Открыть страницу help.baikalelectronics.ru.
2. Войти в систему с использованием логина и пароля, полученных в службе поддержки.
3. В верхней панели навигации выбрать раздел База знаний.
4. На открывшейся странице перейти в SDK → SDK MCU.
5. В окне со списком доступных для скачивания версий SDK выбрать необходимую версию.
6. На открывшейся странице выбрать необходимые файлы и загрузить их нажатием правой кнопки мыши, или нажать Скачать все файлы.

4.2.2 Распаковка SDK

4.2.2.1 ОС Windows



Для установки SDK в ОС Windows необходимы права на запуск файлов .bat.

1. Распаковать архив в рабочую директорию (далее – **[SDK_Path]**).
2. Перейти в директорию **[SDK_Path]\Tools\toolchain\xpack-windows-build-tools** и выполнить файл **add_to_path.bat** для установки переменных окружения и работы с утилитой **Make**.

4.2.2.2 ОС Linux

1. Перейти в директорию с архивом:

```
cd [путь к директории]
```

2. Распаковать архив в рабочую директорию (далее - **[SDK_Path]**):

```
tar -xf [имя архива].tar.gz -C [SDK_Path]
```

5 Подготовка проекта программы

В данном разделе приведены действия, которые необходимо выполнить перед сборкой образа из собственного проекта программы для МК.

Шаг 1: Создать директорию для хранения проекта

Для упрощения установки новых версий SDK, директорию рекомендуется создавать вне директории с SDK.

Шаг 2: Поместить проект в директорию

В созданную директорию поместить файлы с исходными кодами проекта `[project].c` / `[project.h]` / `[project].s`.

Шаг 3: В директории с проектом создать файл `makefile`

Для сборки образа необходимо в директории с проектом создать сборочный скрипт `makefile`, включающий файл общих правил сборки (`$(SDK_DIR)/Tools/build/common.mk`), и добавить в него требуемые параметры.

Пример содержимого файла `makefile`:

```
SDK_DIR = /home/user_name/mcu-sdk
BOARD = EVU_BA_2_0
include $(SDK_DIR)/Tools/build/common.mk
```

Список возможных параметров и их описание приведены в файле

`[SDK_Path]/Tools/build/Makefile_template`.

Также для создания проекта можно использовать шаблон проекта из состава SDK. Шаблон расположен в `Projects/_template`.

6 Сборка образа программы

Образ – это бинарный файл, результат сборки примера или собственного проекта. Исполнение программы после [начальной загрузки](#) МК возможно из TCM или eFlash. Образ программы может быть записан в TCM, eFlash или QSPI Flash. Подробнее о типах памяти см. в документе **BE-U1000. Reference Manual**.

В данном разделе представлена инструкция по подготовке собственного проекта к сборке, а также примеры сборки образов для типов памяти TCM и eFlash. Для записи в QSPI Flash с последующим запуском из TCM можно использовать сборку для TCM. Сборка образов показана на примере GPIO_IOToggle из состава SDK.

Для сборки образов необходимо [загрузить](#) и [распаковать](#) SDK.

При сборке образов используются файлы `makefile` из состава SDK. Данные файлы расположены в директориях с примерами программ для МК.

Параметры сборки могут быть записаны в `makefile` проекта или указаны после команды `make`, используемой для сборки. Полный перечень параметров указан в файле `Tools/build/makefile_template`.

6.1 Сборка образа для исполнения из eFlash

Шаг 1: Запустить терминал командной строки

Рекомендуется использовать `cmd` в Windows или `bash` в Linux.

Шаг 2: Перейти в директорию с проектом

Для Linux:

```
cd [SDK_Path]/Projects/HAL_examples/GPIO(GPIO_IOToggle
```

Для Windows:

```
cd [SDK_Path]\Projects\HAL_examples\GPIO\GPIO_IOToggle
```

Шаг 3: Собрать образ

Удалить результаты предыдущих сборок (если сборка производилась ранее):

```
make clean
```

Запустить сборку:

```
make MEM_REG_ROM=EFLASH
```

Бинарные файлы (`GPIO_IOToggle.elf`, `GPIO_IOToggle.bin`) по умолчанию будут генерированы в директории `build/debug`.

6.2 Сборка образа для исполнения из TCM

Шаг 1: Запустить терминал командной строки

Рекомендуется использовать cmd в Windows или bash в Linux.

Шаг 2: Перейти в директорию с проектом

Для Linux:

```
cd [SDK_Path]/Projects/HAL_examples/GPIO/GPIO_IOToggle
```

Для Windows:

```
cd [SDK_Path]\Projects\HAL_examples\GPIO\GPIO_IOToggle
```

Шаг 3: Собрать образ

Удалить результаты предыдущих сборок (если сборка производилась ранее):

```
make clean
```

Запустить сборку:

```
make MEM_REG_ROM=TCMA
```

Файлы (GPIO_IOToggle.elf, GPIO_IOToggle.bin) по умолчанию будут сгенерированы в директории build/debug.

7 Запись и исполнение образа программы в МК

В данном разделе содержатся инструкции по способам записи образов в различные виды памяти с использованием ОС Linux или Windows.

В таблице ниже находятся ссылки для быстрого перехода к нужному разделу.

Табл. 7-1 – Быстрый переход

ОС	Режим записи	Тип памяти		
		TCM	eFlash	QSPI ¹
Windows	DFU	7.1.1	7.1.2	7.1.3
	UART	7.1.4	7.1.5	7.1.6
	JTAG	7.1.7	7.1.8	–
Linux	DFU	7.2.1	7.2.2	7.2.3
	UART	7.2.4	7.2.5	7.2.6
	JTAG	7.2.7	7.2.8	–

¹ Описывается способ записи программы в QSPI Flash с ее исполнением из TCM.

7.1 ОС Windows

7.1.1 Работа с TCM в режиме DFU



Память TCM очищается после снятия питания.

Перед началом работы необходимо установить на ПК утилиту dfu-util, и драйвер WinUSB.

Шаг 1: Выполнить подготовку

- Выполнить [сборку образа](#) для TCM.
- Подключиться к МК в соответствии с документацией на используемую плату.
- Запустить МК в режиме «DFU» (см. главу [Начальная загрузка](#)).

Шаг 2: Запустить терминал командной строки

Рекомендуется использовать cmd.

Шаг 3: При необходимости вывести список DFU-устройств

Вывод списка позволяет проверить корректность подключения по DFU, а также получить список устройств, режимов записи и параметров для подключения.

```
dfu-util -l
.
.
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=0,
name="TCM and run", serial="0123456789ABCDEF0123456789ABCDEF"
```

В случае получения сообщения об ошибке проверить корректность подключения и режима работы МК, после чего повторить попытку.

Шаг 4: Записать образ

Записать образ в память МК с указанием режима записи `--alt 0` (запись в TCM с исполнением):

```
dfu-util --alt 0 --download [путь к директории с образом] \main.bin
```

При необходимости указать устройство (например `--path 1-2`).

После загрузки образа управление сразу передается ТСМ.

7.1.2 Работа с eFlash в режиме DFU

Перед началом работы необходимо установить на ПК утилиту `dfu-util`, и драйвер `WinUSB`.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для eFlash.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «DFU» (см. главу [Начальная загрузка](#)).

Шаг 2: Запустить терминал командной строки

Рекомендуется использовать `cmd`.

Шаг 3: При необходимости вывести список DFU-устройств

Вывод списка позволяет проверить корректность подключения по DFU, а также получить список устройств, режимов записи и параметров для подключения.

```
dfu-util -l
.
.
.
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=3,
name="eFlash", serial="0123456789ABCDEF0123456789ABCDEF"
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=2,
name="eFlash and run", serial="0123456789ABCDEF0123456789ABCDEF"
.
.
```

В случае получения сообщения об ошибке проверить корректность подключения и режима работы МК, после чего повторить попытку.

Шаг 4: Записать образ

Записать образ в память МК с указанием режима записи `alt 2` (запись в eFlash с исполнением):

```
dfu-util --alt 2 --download [путь к директории с образом] \main.bin
```

Или `--alt 3` (Запись в eFlash):

```
dfu-util --alt 3 --download [путь к директории с образом] \main.bin
```

При необходимости указать устройство (например `--path 1-2`).

При вводе `-alt 2` после загрузки образа происходит перезапуск МК в режиме «EFLASH».

При вводе `--alt 3` после загрузки образа МК продолжает работу в текущем режиме. Для исполнения программы необходимо установить режим загрузки «EFLASH» и перезагрузить МК.

7.1.3 Работа с QSPI Flash в режиме DFU

В данном разделе описывается способ записи программы в QSPI Flash с ее исполнением из TCM.

Перед началом работы необходимо установить на ПК утилиту `dfu-util` и драйвер WinUSB.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для ТСМ.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «DFU» (см. главу [Начальная загрузка](#)).

Шаг 2: Запустить терминал командной строки

Рекомендуется использовать `cmd`.

Шаг 3: При необходимости вывести список DFU-устройств

Вывод списка позволяет проверить корректность подключения по DFU, а также получить список устройств, режимов записи и параметров для подключения.

```
dfu-util -l
.
.
.
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=7,
name="SPI part", serial="0123456789ABCDEF0123456789ABCDEF"
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=6,
name="SPI", serial="0123456789ABCDEF0123456789ABCDEF"
.
.
```

В случае получения сообщения об ошибке проверить корректность подключения и режима работы МК, после чего повторить попытку.

Шаг 4: Записать образ

Записать образ в память МК с указанием режима записи `--alt 6`:

```
dfu-util --alt 6 --download [путь к директории с образом]\main.bin
```

При необходимости указать устройство (например `--path 1-2`).

Максимальный размер образа – 16 МБ.

Для исполнения программы необходимо установить режим загрузки «QSPI» и перезагрузить МК.

7.1.4 Работа с TCM в режиме UART



Память TCM очищается после снятия питания.

Перед началом работы необходимо установить на ПК программу удаленного терминала с поддержкой протокола XMODEM (например, ExtraPuTTY).

В инструкции ниже в качестве примера используется ExtraPuTTY.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для TCM.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «UART» (см. главу [Начальная загрузка](#)).

Шаг 2: Подключиться к интерфейсу МК «UART0»

1. Подключиться к МК через виртуальный COM-порт.
2. Запустить ExtraPuTTY и подключиться к интерфейсу МК «UART0».

Установить следующие параметры подключения к интерфейсу:

Скорость передачи (Baud Rate)	115200
Управление потоком (Flow Control)	Нет
Биты четности (Parity)	Нет
Биты данных (Data Bits)	8
Стоп-биты (Stop Bits)	1

Установить параметр для корректного отображения конца строки:

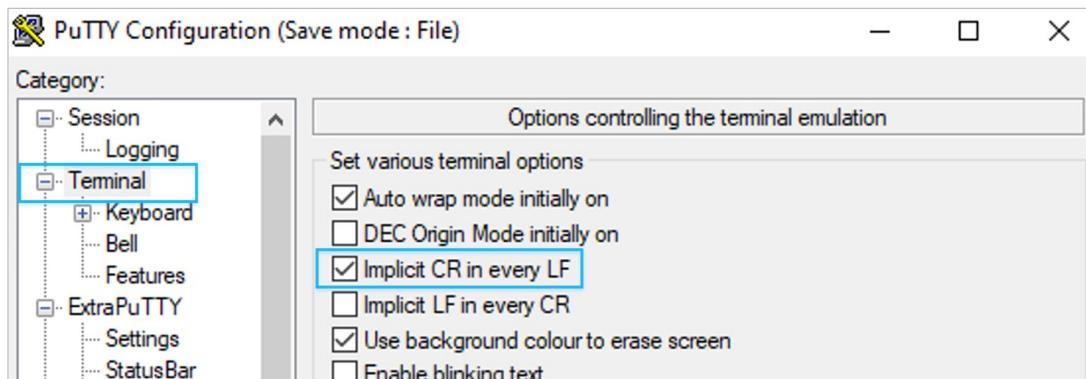


Рис. 7-1 – Настройка ExtraPutty

После настройки подключения нажать Open:

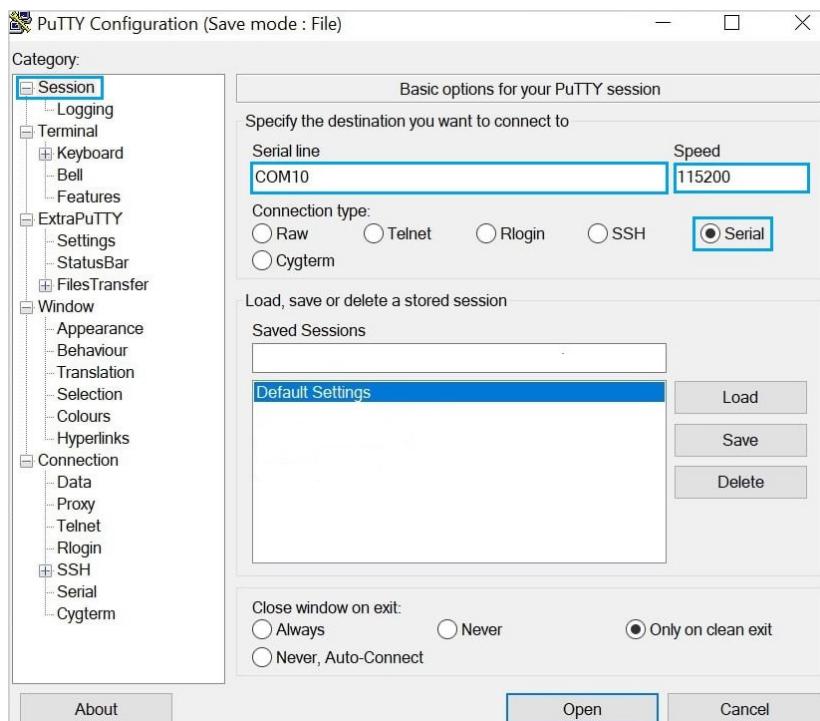


Рис. 7-2 – Подключение к МК

Откроется терминал и отобразятся следующие сообщения:

CC

Шаг 3: Войти в BOOTROM CLI

1. На плате нажать кнопку *RESET*.
2. Нажать и удерживать клавишу <D> до появления сообщения:

```
BU ROM[v1.1.0 (2025-04-04)_2c05769d]  
@
```

Лишние символы `d` необходимо удалить.

Шаг 4: Записать образ в память МК

1. Нажать клавишу <I>. На экране отобразится сообщение о готовности:

```
Loading to 0x40010000 sz 0x22800 from 0x0 media 0x3  
C
```



Запись `sz 0x24000` обозначает максимальный размер файла. Для его изменения после входа в BootROM CLI необходимо ввести команду «`: [размер]`» и нажать <Enter>. Значение вводится в шестнадцатеричном формате.

2. В интерфейсе ExtraPutty Нажать Files Transfer → XMODEM → Send:

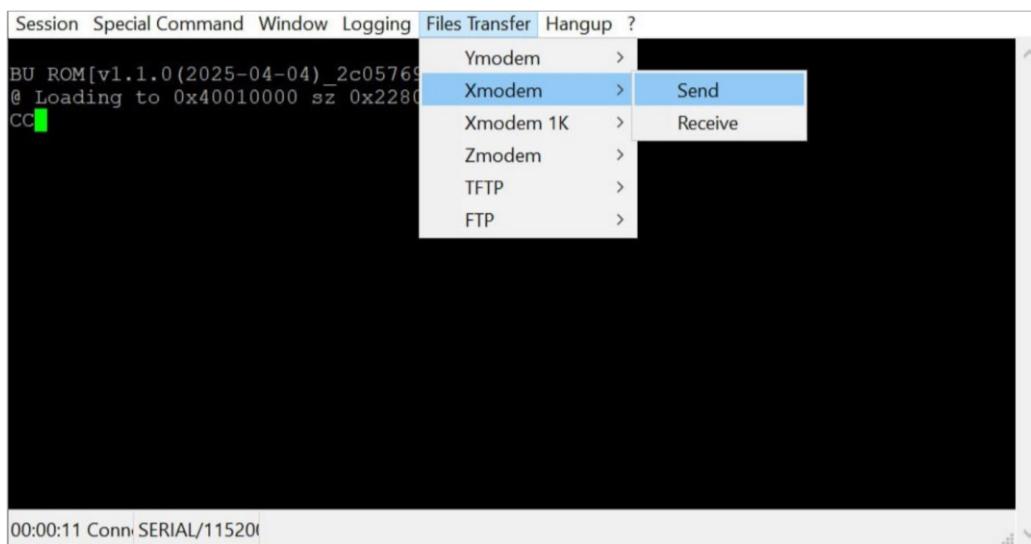


Рис. 7-3 – Передача по XMODEM

3. В открывшемся окне выбрать файл образа с расширением .bin.

4. Дождаться окончания передачи файла.

В UART-консоли появится символ @.

 В случае появления сообщений об ошибке передачи повторить запись, начиная с входа в BootROM CLI.

Шаг 5: Запустить исполнение программы

Нажать клавишу <G>. Начнется исполнение кода.

7.1.5 Работа с eFlash в режиме UART

Перед началом работы необходимо установить на ПК программу удаленного терминала с поддержкой протокола XMODEM (например, ExtraPuTTY).

В инструкции ниже в качестве примера используется ExtraPuTTY.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для eFlash.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «UART» (см. главу [Начальная загрузка](#)).

Шаг 2: Подключиться к интерфейсу MK «UART0»

1. Подключиться к MK через виртуальный COM-порт.
2. Запустить ExtraPutty
3. Подключиться к интерфейсу MK «UART0».

Установить следующие параметры подключения к UART-интерфейсу:

Скорость передачи (Baud Rate)	115200
Управление потоком (Flow Control)	Нет
Биты четности (Parity)	Нет
Биты данных (Data Bits)	8
Стоп-биты (Stop Bits)	1

Установить параметр для корректного отображения конца строки:

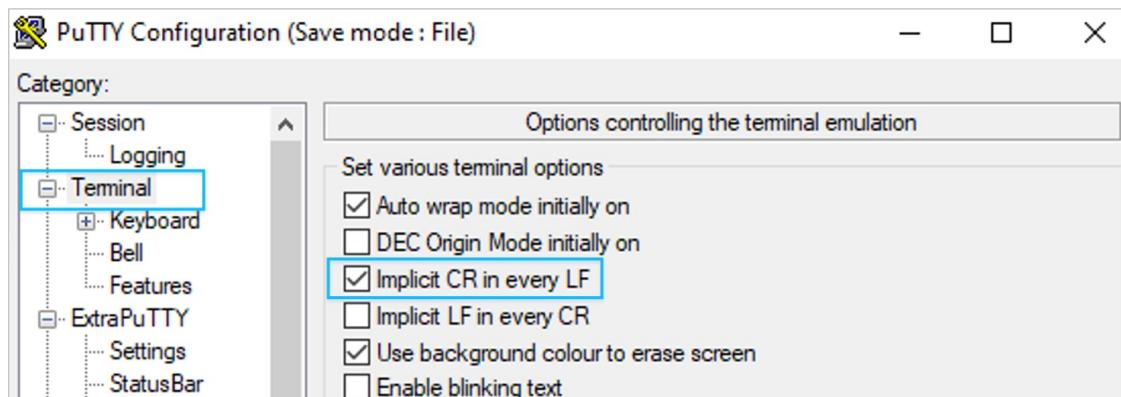


Рис. 7-4 – Настройка ExtraPutty

После настройки подключения нажать Open:

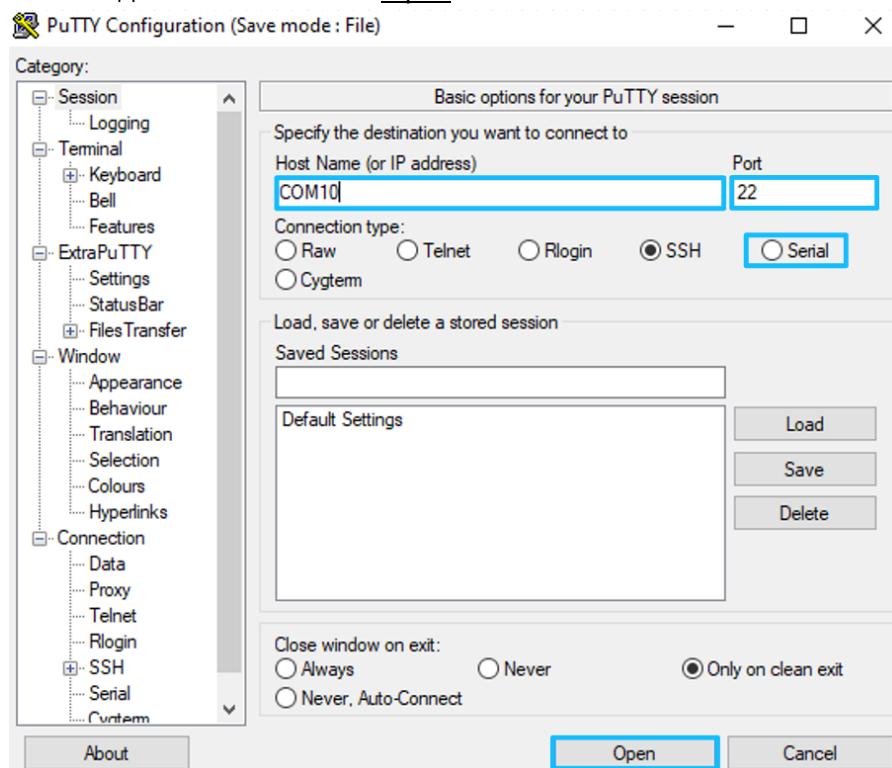


Рис. 7-5 – Подключение к MK

Откроется терминал и отобразятся следующие сообщения:

```
CC
```

Шаг 3: Войти в BootROM CLI

1. На плате нажать кнопку *RESET*.
2. Нажать и удерживать клавишу <D> до появления сообщения:

```
BU ROM[v1.1.0(2025-04-04)_2c05769d]  
@
```

Лишние символы @ необходимо удалить.

Шаг 4: Снять защиту с eFlash

```
14000018:1w
```

Шаг 5: Выставить настройки

Установить адрес eFlash (@a0000000), максимальный размер файла (в данном примере – 22800), частоту тактирования 25 МГц (100):

```
a0000000:22800:0:100m
```

Шаг 6: Очистить eFlash

```
:::101m
```

Шаг 7: Дать команду на ожидание файла образа по протоколу XMODEM

```
:::3L
```

Шаг 8: Загрузить программу в память МК

1. В интерфейсе ExtraPutty нажать Files Transfer → XMODEM → Send:

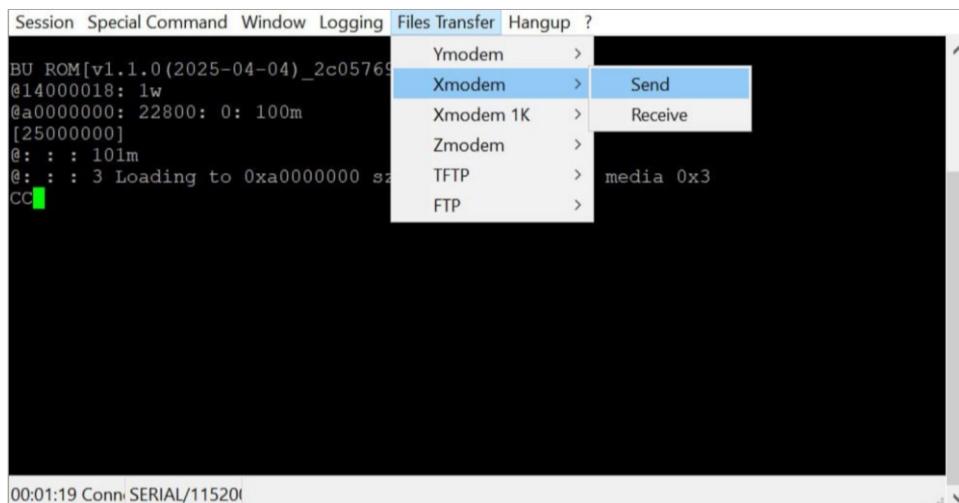


Рис. 7-6 – Передача по XMODEM

2. В открывшемся окне выбрать файл образа с расширением .bin.
3. Дождаться окончания передачи файла.

В UART-консоли появится символ @.



В случае появления сообщений об ошибке передачи повторить запись, начиная с входа в BootROM CLI ([Шаг 3](#)).

Шаг 9: Запустить исполнение программы

Способ 1 - Из BootROM CLI:

В BootROM CLI указать адрес eFlash-памяти:

```
a0000000g
```

Способ 2 - Установкой страповых выводов:

1. Снять питание с платы.
2. Установить на используемой плате режим загрузки из eFlash.
3. Подать питание на плату.

7.1.6 Работа с *QSPI Flash* в режиме *UART*

В данном разделе описывается способ записи программы в QSPI Flash с ее исполнением из TCM.

Перед началом работы необходимо установить на ПК программу удаленного терминала с поддержкой протокола XMODEM (например, ExtraPuTTY).

В инструкции ниже в качестве примера используется ExtraPuTTY.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для ТСМ.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «UART» (см. главу [Начальная загрузка](#)).

Шаг 2: Подключиться к интерфейсу МК «UART0»

1. Подключиться к МК через виртуальный COM-порт.
2. Открыть утилиту ExtraPutty и подключиться к интерфейсу МК «UART0».

Установить следующие параметры подключения к интерфейсу:

Скорость передачи (Baud Rate)	115200
Управление потоком (Flow Control)	Нет
Биты четности (Parity)	Нет
Биты данных (Data Bits)	8
Стоп-биты (Stop Bits)	1

Установить параметр для корректного отображения конца строки:

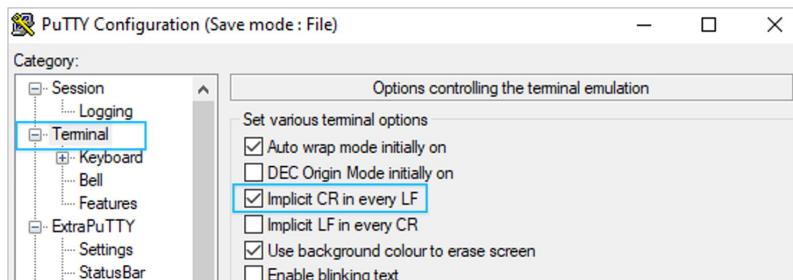


Рис. 7-7 – Настройка ExtraPutty

После настройки подключения нажать Open:

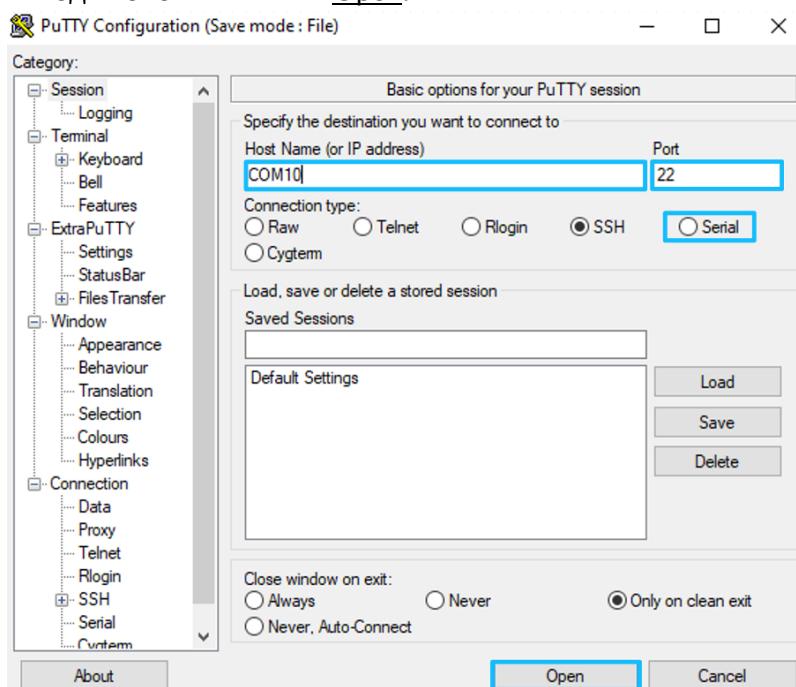


Рис. 7-8 – Подключение к МК

Откроется терминал и отобразятся следующие сообщения:

CC

Шаг 3: Войти в BootROM CLI

- На плате нажать кнопку *RESET*.
- Нажать и удерживать клавишу <D> до появления сообщения:

```
BU ROM[v1.1.0 (2025-04-04)_2c05769d]
```

@

Лишние символы @ необходимо удалить.

Шаг 4: Записать образ в память МК

- Ввести команду на отключение смещения, прием данных по XMODEM и сохранение их в QSPI Flash:

```
::0:0x
```

Будет выведено сообщение о готовности к принятию файла:

```
@: : 0: 0 Xmodem saving to 0x0 media 0x0  
C
```

2. В интерфейсе ExtraPutty нажать Files Transfer → XMODEM → Send:

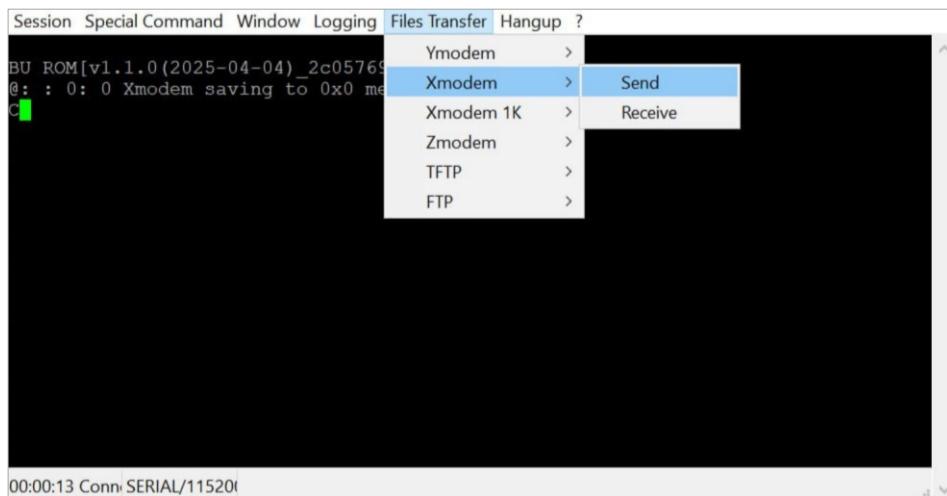


Рис. 7-9 – Передача по XMODEM

3. В открывшемся окне выбрать файл образа с расширением **.bin**.

4. Дождаться окончания передачи файла.

В UART-консоли появится символ **@**.



В случае появления сообщений об ошибке передачи повторить запись, начиная с входа в BootROM CLI.

Шаг 5: Запустить исполнение программы

1. Снять питание с платы.
2. Установить на используемой плате режим загрузки «QSPI».
3. Подать питание на плату.

7.1.7 Работа с TCM в режиме JTAG



Память TCM очищается после снятия питания.

Перед началом работы необходимо установить на ПК драйвер WinUSB, [загрузить](#) и [распаковать](#) SDK.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для TCM.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «JTAG» (см. главу [Начальная загрузка](#)).

Шаг 2: Запустить терминал командной строки

Рекомендуется использовать cmd.

Шаг 3: В терминале командной строки открыть директорию с OpenOCD

```
cd [SDK_Path]\Tools\OpenOCD\bin
```

Шаг 4: Запустить OpenOCD

Запустить бинарный файл с указанием файла конфигурации для используемой платы:

```
.\openocd.exe -f ..\share\openocd\scripts\board\baikal\evu-ba_2_0.cfg
```

При успешном подключении будут выведены сообщения вида:

```
Info : [BMCU_Core2] datacount=2 proobufsize=16
Info : [BMCU_Core2] Disabling abstract command reads from CSRs.
Info : [BMCU_Core2] Disabling abstract command writes to CSRs.
Info : [BMCU_Core2] Examined RISC-V core
Info : [BMCU_Core2] XLEN=32, misa=0x40100106
[BMCU_Core2] Target successfully examined.
Info : [BMCU_Core2] Examination succeed
```

Шаг 5: Открыть второе окно терминала командной строки

Все дальнейшие действия выполнять в данном окне.

Шаг 6: Открыть директорию с кросс-компилятором

Во втором окне консоли перейти к директории:

```
cd [SDK_Path]\tools\toolchain\riscv32-picolibc-elf\bin
```

Шаг 7: Запустить отладчик GDB

Во втором окне консоли ввести команду:

```
[SDK_path]\Tools\toolchain\riscv32-picolibc-elf\bin\riscv32-picolibc-elf-gdb -x
[SDK_path]\Tools\scripts\bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 8: Записать образ через отладчик GDB

Записать бинарный образ в память МК:

```
restore [путь к директории с образом]\[имя образа].bin binary 0x40010000
```

После успешной записи будут выведены сообщения вида:

```
Loading section .text, size 0x364 lma 0x40010000
Loading section .note.gnu.build-id, size 0x24 lma 0xa0000364
Start address 0x40010000, load size 904
Transfer rate: 8 KB/sec, 452 bytes/write.
```

Шаг 9: Запустить исполнение программы

- Перейти к адресу TCM:

```
bmcu jump 0x40010000
```

- Дать команду на исполнение:

```
continue
```



Перед повторной записью программы в одной и той же сессии, для сброса состояния МК рекомендуется в GDB перезапустить МК командой `bmcu reset`.



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.1.8 Работа с eFlash в режиме JTAG

Перед началом работы необходимо установить на ПК драйвер WinUSB, [загрузить](#) и [распаковать](#) SDK.

Шаг 1: Выполнить подготовку

- Выполнить [сборку образа](#) для ТСМ.
- Подключиться к МК в соответствии с документацией на используемую плату.
- Запустить МК в режиме «JTAG» (см. главу [Начальная загрузка](#)).

Шаг 2: Запустить терминал командной строки

Рекомендуется использовать `cmd`.

Шаг 3: В терминале командной строки открыть директорию с OpenOCD

```
cd [SDK_Path]\Tools\OpenOCD
```

Шаг 4: Запустить OpenOCD

Запустить бинарный файл с указанием файла конфигурации:

```
.\openocd.exe -f ..\share\openocd\scripts\board\baikal\evu-ba_2_0.cfg
```

При успешном подключении будут выведены сообщения вида:

```
Info : [BMCU_Core2] datacount=2 progbufsize=16
Info : [BMCU_Core2] Disabling abstract command reads from CSRs.
Info : [BMCU_Core2] Disabling abstract command writes to CSRs.
Info : [BMCU_Core2] Examined RISC-V core
Info : [BMCU_Core2] XLEN=32, misa=0x40100106
[BMCU_Core2] Target successfully examined.
Info : [BMCU_Core2] Examination succeed
```

Шаг 5: Открыть второе окно терминала командной строки

Все дальнейшие действия выполнять в данном окне.

Шаг 6: Открыть директорию с кросс-компилятором

Во втором окне консоли перейти к директории:

```
cd [SDK_Path]\tools\toolchain\riscv32-picolibc-elf\bin
```

Шаг 7: Запустить отладчик GDB

Во втором окне консоли ввести команду:

```
[SDK_path]\Tools\toolchain\riscv32-picolibc-elf\bin\riscv32-picolibc-elf-gdb -x  
[SDK_path]\Tools\scripts\bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 8: Записать образ через отладчик GDB

1. Очистить память eFlash:

```
bmcu eFlash erase
```

2. Перезагрузить МК:

```
bmcu reset
```

3. Записать бинарный образ в память МК:

```
restore [путь к директории с образом] \ [имя образа].bin binary 0xa0000000
```

После успешной записи будут выведены сообщения вида:

```
Loading section .text, size 0x364 lma 0xa0000000  
Loading section .note.gnu.build-id, size 0x24 lma 0xa0000364  
Start address 0xa0000000, load size 904  
Transfer rate: 8 KB/sec, 452 bytes/write.
```

Шаг 9: Запустить исполнение программы

1. Перейти к адресу eFlash:

```
bmcu jump 0xa0000000
```

2. Дать команду на запуск:

```
continue
```



Перед повторной записью программы в одной и той же сессии, для сброса состояния МК рекомендуется в GDB перезапустить МК командой `bmcu_reset`.



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.2 ОС Linux

7.2.1 Работа с TCM в режиме DFU



Память TCM очищается после снятия питания.

Перед началом работы необходимо установить на ПК утилиту dfu-util.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для TCM.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «DFU» (см. главу [Начальная загрузка](#)).

Шаг 2: При необходимости вывести список DFU-устройств

Вывод списка позволяет проверить корректность подключения по DFU, а также получить список устройств, режимов записи и параметров для подключения.

```
sudo dfu-util -l
...
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=0,
name="TCM and run", serial="0123456789ABCDEF0123456789ABCDEF"
```

В случае получения сообщения об ошибке проверить корректность подключения и режима работы МК, после чего повторить попытку.

Шаг 3: Записать образ

Записать образ в память МК с указанием режима записи alt=0 (запись в TCM с исполнением):

```
sudo dfu-util --alt 0 --download / [путь к директории с образом] /main.bin
```

При необходимости указать устройство (например --path 1-2).

При вводе --alt 2 после загрузки образа происходит перезапуск МК в режиме «EFLASH».

При вводе --alt 3 после загрузки образа МК продолжает работу в текущем режиме. Для исполнения программы необходимо установить режим загрузки «EFLASH» и перезагрузить МК.

7.2.2 Работа с eFlash в режиме DFU

Перед началом работы необходимо установить на ПК утилиту dfu-util.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для eFlash.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «DFU» (см. главу [Начальная загрузка](#)).

Шаг 2: При необходимости вывести список DFU-устройств

Вывод списка позволяет проверить корректность подключения по DFU, а также получить список устройств, режимов записи и параметров для подключения.

```
sudo dfu-util -l
.
.
.
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=3,
name="eFlash", serial="0123456789ABCDEF0123456789ABCDEF"
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=2,
name="eFlash and run", serial="0123456789ABCDEF0123456789ABCDEF"
.
.
```

В случае получения сообщения об ошибке проверить корректность подключения и режима работы МК, после чего повторить попытку.

Шаг 3: Записать образ

Записать образ в память МК с указанием режима записи `alt 2` (запись в eFlash с исполнением):

```
dfu-util --alt 2 --download [путь к директории с образом]\main.bin
```

Или `--alt 3` (Запись в eFlash):

```
dfu-util --alt 3 --download [путь к директории с образом]\main.bin
```

При необходимости указать устройство (например `--path 1-2`).

При вводе `-alt 2` после загрузки образа происходит перезапуск МК в режиме «EFLASH».

При вводе `-alt 3` после загрузки образа МК продолжает работу в текущем режиме. Для исполнения программы необходимо установить режим загрузки «EFLASH» и перезагрузить МК.

7.2.3 Работа с QSPI Flash в режиме DFU

В данном разделе описывается способ записи программы в QSPI Flash с ее исполнением из TCM.

Перед началом работы необходимо установить на ПК утилиту dfu-util.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для TCM.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «DFU» (см. главу [Начальная загрузка](#)).

Шаг 2: При необходимости вывести список DFU-устройств

Вывод списка позволяет проверить корректность подключения по DFU, а также получить список устройств, режимов записи и параметров для подключения.

```
sudo dfu-util -l
.
.
.
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=7,
name="SPI part", serial="0123456789ABCDEF0123456789ABCDEF"
Found DFU: [cafe:2303] ver=0100, devnum=6, cfg=1, intf=2, path="1-2", alt=6,
name="SPI", serial="0123456789ABCDEF0123456789ABCDEF"
.
.
```

В случае получения сообщения об ошибке проверить правильность подключения и выбора режима работы МК, после чего повторить попытку.

Шаг 3: Записать образ

Записать образ в память МК с указанием режима записи alt 6:

```
sudo dfu-util --alt 6 --download / [путь к директории с образом] /main.bin
```

При необходимости указать устройство (например --path 1-2).

Максимальный размер образа – 16 МБ.

Для исполнения программы необходимо установить режим загрузки «QSPI» и перезагрузить МК.

7.2.4 Работа с TCM в режиме UART



Память TCM очищается после снятия питания.

Перед началом работы необходимо установить на ПК утилиту Picocom и пакет Lrzsz.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для TCM.
2. Подключиться к MK в соответствии с документацией на используемую плату.
3. Запустить MK в режиме «UART» (см. главу [Начальная загрузка](#)).

Шаг 2: Подключиться к интерфейсу MK «UART0»

Подключиться к интерфейсу MK «UART0» через COM-порт со следующими параметрами:

Скорость передачи (Baud Rate)	115200
Управление потоком (Flow Control)	Нет
Биты четности (Parity)	Нет
Биты данных (Data Bits)	8
Стоп-биты (Stop Bits)	1

Команда на подключение:

```
sudo picocom /dev/ttyUSB0 -b 115200 --send-cmd sx --imap lfcrlf
```

Откроется терминал и отобразятся следующие сообщения:

```
CC
```

Шаг 3: Войти в BootROM CLI

1. На плате нажать кнопку *RESET*.
2. Нажать и удерживать клавишу <D> до появления сообщения:

```
BU ROM[v1.1.0(2025-04-04)_2c05769d]  
@
```

Лишние символы @ необходимо удалить.

Шаг 4: Записать программу в память MK

1. Нажать клавишу <I>. На экране отобразится сообщение о готовности:

```
Loading to 0x40010000 sz 0x24000 from 0x0 media 0x3  
C
```



Запись sz 0x24000 обозначает максимальный размер файла. Для его изменения после входа в BootROM CLI необходимо ввести команду «: [размер]» и нажать <Enter>. Значение вводится в шестнадцатеричном формате.

2. Открыть второе окно терминала и выполнить в нем команду:

```
sudo sx '/[путь к директории с образом]/[имя образа].bin' \  
</dev/[имя устройства> /dev/[имя устройства]
```

Будет выведено сообщение об отправке файла:

```
Sending / [путь к директории с образом] / [имя образа].bin, 7 blocks: Give your  
local XMODEM receive command now.
```

После успешной передачи файла во втором окне терминала появится сообщение:

```
Bytes Sent: 1024 BPS:2911  
Transfer complete
```

В первом окне терминала появится символ Ⓛ.



Появление промежуточных сообщений `Retry 0: NAK on sector` сигнализирует об ошибке в процессе записи. В случае их появления повторить запись, начиная с входа в BootROM CLI ([Шаг 3](#)).

Шаг 5: Запустить исполнение программы

В первом окне терминала нажать клавишу `<G>`. В случае успешной установки начнется исполнение кода.

7.2.5 Работа с eFlash в режиме UART

Перед началом работы необходимо установить на ПК утилиту Picocom и пакет Lrzsz.

Шаг 1: Выполнить подготовку

- Выполнить [сборку образа](#) для eFlash.
- Подключиться к МК в соответствии с документацией на используемую плату.
- Запустить МК в режиме «UART» (см. главу [Начальная загрузка](#)).

Шаг 2: Подключиться к интерфейсу МК «UART0»

Подключиться к интерфейсу МК «UART0» через COM-порт со следующими параметрами:

Скорость передачи (Baud Rate)	115200
Управление потоком (Flow Control)	Нет
Биты четности (Parity)	Нет
Биты данных (Data Bits)	8
Стоп-биты (Stop Bits)	1

Команда на подключение:

```
sudo picocom /dev/ttyUSB0 -b 115200 --send-cmd sx --imap lfcrlf
```

Откроется терминал и отобразятся следующие сообщения:

```
CC
```

Шаг 3: Войти в BootROM CLI

1. На плате нажать кнопку *RESET*.
2. Нажать и удерживать клавишу <D> до появления сообщения:

```
BU ROM[v1.1.0 (2025-04-04)_2c05769d]  
@
```

Лишние символы @ необходимо удалить.

Шаг 4: Снять защиту с eFlash

```
14000018:1w
```

Шаг 5: Выставить необходимые настройки

Установить адрес eFlash (`a0000000`), максимальный размер файла (в данном примере – `22800`), частоту тактирования `25 МГц (100)`:

```
a0000000:22800:0:100m
```

Шаг 6: Очистить eFlash

```
:::101m
```

Шаг 7: Дать команду на ожидание файла образа по протоколу XMODEM

```
:::3L
```

Шаг 8: Записать программу в память МК

Открыть второе окно терминала и выполнить в нем команду:

```
sudo sx '/[путь к директории с образом]/[имя образа].bin' \  
</dev/[имя устройства> /dev/[имя устройства]
```

Будет выведено сообщение об отправке файла:

```
Sending /[путь к директории с образом]/[имя образа].bin, 7 blocks: Give your  
local XMODEM receive command now.
```

После успешной передачи файла во втором окне терминала появится сообщение:

```
Bytes Sent: 1024 BPS:2911  
Transfer complete
```

В первом окне терминала появится символ @.



Появление промежуточных сообщений `Retry 0: NAK on sector` сигнализирует об ошибке в процессе записи. В случае их появления повторить запись, начиная с входа в BootROM CLI ([Шаг 3](#)).

Шаг 9: Запустить исполнение программы**Способ 1 - Из BootROM CLI**

В BootROM CLI указать адрес eFlash-памяти:

```
a0000000g
```

Способ 2 - Установкой страповых выводов.

1. Снять питание с платы.
2. Установить на используемой плате режим загрузки «eFlash».
3. Подать питание на плату.

7.2.6 Работа с QSPI Flash в режиме UART

В данном разделе описывается способ записи программы в QSPI Flash с ее исполнением из TCM.

Перед началом работы необходимо установить на ПК утилиту Picocom и пакет Lrzsz.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для ТСМ.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «UART» (см. главу [Начальная загрузка](#)).

Шаг 2: Подключиться к интерфейсу МК «UART0»

Подключиться к интерфейсу МК «UART0» через СОМ-порт со следующими параметрами:

Скорость передачи (Baud Rate)	115200
Управление потоком (Flow Control)	Нет
Биты четности (Parity)	Нет
Биты данных (Data Bits)	8
Стоп-биты (Stop Bits)	1

Команда на подключение:

```
sudo picocom /dev/ttyUSB0 -b 115200 --send-cmd sx --imap lfcrlf
```

Откроется терминал и отобразятся следующие сообщения:

```
CC
```

Шаг 3: Войти в BootROM CLI

1. На плате нажать кнопку *RESET*.
2. Нажать и удерживать клавишу <D> до появления сообщения:

```
BU ROM[v1.1.0(2025-04-04)_2c05769d]  
@
```

Лишние символы `d` необходимо удалить.

Шаг 4: Записать образ в память МК

1. Ввести команду на отключение смещения, прием данных по XMODEM и сохранение их в QSPI Flash:

```
:::0:0x
```

Будет выведено сообщение о готовности к принятию файла:

```
@: : 0: 0 Xmodem saving to 0x0 media 0x0  
C
```

2. Нажать <Ctrl+A> и <Ctrl+S>, после чего ввести путь к файлу [имя образа].bin.
3. Начнется загрузка, после ее окончания будет выведено сообщение.

Шаг 5: Запустить исполнение программы

1. Снять питание с платы.
2. Установить на используемой плате режим загрузки «QSPI».
3. Подать питание на плату.

7.2.7 Работа с TCM в режиме JTAG



Память TCM очищается после снятия питания.

Перед началом работы необходимо [загрузить](#) и [распаковать](#) SDK.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для TCM.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «JTAG» (см. главу [Начальная загрузка](#)).

Шаг 2: Открыть в терминале директорию с OpenOCD

```
cd / [SDK_Path]/Tools/OpenOCD/bin
```

Шаг 3: Запустить OpenOCD

Запустить бинарный файл с указанием файла конфигурации:

```
sudo ./openocd -f ../../share/openocd/scripts/board/baikal/evu-ba_2_0.cfg
```

При успешном подключении будут выведены сообщения вида:

```
Info : [BMCU_Core2] datacount=2 progbufsize=16
Info : [BMCU_Core2] Disabling abstract command reads from CSRs.
Info : [BMCU_Core2] Disabling abstract command writes to CSRs.
Info : [BMCU_Core2] Examined RISC-V core
Info : [BMCU_Core2] XLEN=32, misa=0x40100106
[BMCU_Core2] Target successfully examined.
Info : [BMCU_Core2] Examination succeed
```

Шаг 4: Открыть второе окно терминала

Все дальнейшие действия выполнять в данном окне.

Шаг 5: Открыть директорию с кросс-компилятором

Во втором окне терминала перейти к директории:

```
cd / [SDK_Path]/Tools/toolchain/riscv32-picolibc-elf/bin/
```

Шаг 6: Запустить отладчик GDB

В новом окне терминала ввести команду:

```
[SDK_path]/Tools/toolchain/riscv32-picolibc-elf/bin/riscv32-picolibc-elf-gdb \
-x [SDK_path]/Tools/scripts/bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 7: Записать образ через отладчик GDB

Записать бинарный образ в память МК:

```
restore [путь к директории с образом] / [имя образа].bin binary 0x40010000
```

После успешной записи будут выведены сообщения вида:

```
Loading section .text, size 0x364 lma 0x40010000
Loading section .note.gnu.build-id, size 0x24 lma 0xa0000364
Start address 0x40010000, load size 904
Transfer rate: 8 KB/sec, 452 bytes/write.
```

Шаг 8: Запустить исполнение программы

1. Перейти к адресу TCM:

```
bmcu jump 0x40010000
```

2. Дать команду на исполнение:

```
continue
```



Перед повторной записью образа в одной и той же сессии, для сброса состояния МК рекомендуется в GDB перезапустить МК командой `bmcu_reset`.



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.2.8 Работа с eFlash в режиме JTAG

Перед началом работы необходимо [загрузить](#) и [распаковать](#) SDK.

Шаг 1: Выполнить подготовку

1. Выполнить [сборку образа](#) для TCM.
2. Подключиться к МК в соответствии с документацией на используемую плату.
3. Запустить МК в режиме «JTAG» (см. главу [Начальная загрузка](#)).

Шаг 2: Открыть в терминале директорию с OpenOCD

```
cd /[SDK_Path]/Tools/OpenOCD
```

Шаг 3: Запустить OpenOCD

Запустить бинарный файл с указанием файла конфигурации:

```
sudo ./openocd -f ../../share/openocd/scripts/board/baikal/evu-ba_2_0.cfg
```

При успешном подключении будут выведены сообщения вида:

```
Info : [BMCU_Core2] datacount=2 proobufsize=16
Info : [BMCU_Core2] Disabling abstract command reads from CSRs.
Info : [BMCU_Core2] Disabling abstract command writes to CSRs.
Info : [BMCU_Core2] Examined RISC-V core
Info : [BMCU_Core2] XLEN=32, misa=0x40100106
[BMCU_Core2] Target successfully examined.
Info : [BMCU_Core2] Examination succeed
```

Шаг 4: Открыть второе окно терминала

Все дальнейшие действия выполнять в данном окне.

Шаг 5: Открыть директорию с кросс-компилятором

Во втором окне терминала перейти к директории:

```
cd /[SDK_Path]/Tools/toolchain/riscv32-picolibc-elf/bin/
```

Шаг 6: Запустить отладчик GDB

В новом окне терминала ввести команду:

```
[SDK_path]/Tools/toolchain/riscv32-picolibc-elf/bin/riscv32-picolibc-elf-gdb \
-x [SDK_path]/Tools/scripts/bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 7: Записать образ через отладчик GDB

1. Очистить память eFlash:

```
bmcu eFlash erase
```

2. Перезагрузить МК:

```
bmcu reset
```

3. Записать бинарный образ в память МК:

```
restore [путь к директории с образом] / [имя образа].bin binary 0xa0000000
```

После успешной записи будут выведены сообщения вида:

```
Loading section .text, size 0x364 lma 0xa0000000
Loading section .note.gnu.build-id, size 0x24 lma 0xa0000364
Start address 0xa0000000, load size 904
Transfer rate: 8 KB/sec, 452 bytes/write.
```

Шаг 8: Запустить исполнение программы**1. Перейти к адресу eFlash:**

```
bmcu jump 0xa0000000
```

2. Дать команду на исполнение:

```
continue
```



Перед повторной записью образа в одной и той же сессии, для сброса состояния МК рекомендуется в GDB перезапустить МК командой `bmcu_reset`.



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

История изменений

Версия	Дата	Описание
1.0.0	12.12.2025	Начальная версия

Контактные данные



Официальный сайт
www.baikalelectronics.ru



Главный офис
www.baikalelectronics.ru/contacts



Электронная почта
info@baikalelectronics.ru



Телефон
+7 495 221-39-47