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 4
3 НАЧАЛЬНАЯ ЗАГРУЗКА 6
3.1 Общее описание 6
3.2 Сценарии начальной загрузки 6
3.2.1 Сценарий загрузки «EFLASH» 7
3.2.2 Сценарий загрузки «UART» 7
3.2.3 Сценарий загрузки «QSPI» 8
3.2.4 Сценарий загрузки «USB» 8
3.2.5 Сценарий загрузки «JTAG EXT» 10
3.2.6 Сценарии загрузки «UART MP», «USB MP» 10
3.2.7 Сценарий загрузки «MULTI» 12
3.2.8 Сценарий загрузки «JTAG INT» 12
3.2.9 Сценарий загрузки «USB CDC» 12
3.2.10 Сценарий загрузки «Factory reset» 13
3.2.11 Сценарий загрузки «Multi USB» 13
3.3 Интерфейс командной строки начального загрузчика 13
3.3.1 Параметры 14
3.3.2 Функции 14
3.3.3 Расширенные функции (m) 15
3.4 Состояние МК после начальной загрузки 17
4 SDK 19
4.1 Описание SDK 19
4.2 Подготовка SDK 20
4.2.1 Загрузка SDK 20
4.2.2 Распаковка SDK 20
4.2.2.1 ОС Windows 20
4.2.2.2 ОС Linux 20

5 ПОДГОТОВКА ПРОЕКТА ПРОГРАММЫ 21
6 СБОРКА ОБРАЗА ПРОГРАММЫ 22
6.1 Сборка образа для исполнения из eFlash 22
6.2 Сборка образа для исполнения из TCM 23
6.3 Сборка образа для исполнения из QSPI Flash (XIP) 23
6.3.1 Ручная сборка образа для исполнения из QSPI Flash (XIP) 24
6.3.1.1 Сборка пользовательской программы 24
6.3.1.2 Сборка XIP-загрузчика 24
6.3.2 Автоматическая сборка образа для исполнения из QSPI Flash (XIP) 25
7 ЗАПИСЬ И ИСПОЛНЕНИЕ ОБРАЗА ПРОГРАММЫ В МК 26
7.1 ОС Windows 26
7.1.1 Работа с TCM через DFU 26
7.1.2 Работа с eFlash через DFU 27
7.1.3 Работа с QSPI Flash / TCM через DFU 28
7.1.4 Работа с QSPI Flash / XIP через DFU 29
7.1.5 Работа с TCM через UART 30
7.1.6 Работа с eFlash через UART 32
7.1.7 Работа с QSPI Flash / TCM через UART 35
7.1.8 Работа с QSPI Flash / XIP через UART 38
7.1.9 Работа с TCM через JTAG 41
7.1.10 Работа с eFlash через JTAG 43
7.1.11 Работа с QSPI Flash / TCM через JTAG 45
7.1.12 Работа с QSPI Flash / XIP через JTAG 47
7.2 ОС Linux 49
7.2.1 Работа с TCM через DFU 49
7.2.2 Работа с eFlash через DFU 50
7.2.3 Работа с QSPI Flash / TCM через DFU 51
7.2.4 Работа с QSPI Flash / XIP через DFU 52
7.2.5 Работа с TCM через UART 53
7.2.6 Работа с eFlash через UART 55
7.2.7 Работа с QSPI Flash / TCM через UART 57
7.2.8 Работа с QSPI Flash / XIP через UART 58
7.2.9 Работа с TCM через JTAG 60
7.2.10 Работа с eFlash через JTAG 61
7.2.11 Работа с QSPI Flash / TCM через JTAG 63
7.2.12 Работа с QSPI Flash / XIP через JTAG 65
ИСТОРИЯ ИЗМЕНЕНИЙ 67
КОНТАКТНЫЕ ДАННЫЕ 67

1 Введение

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

- Начальная загрузка: общее описание, описание сценариев 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 для ОС Linux и Windows.

2 Условные обозначения, термины и сокращения

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

Таблица 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.

Таблица 2-2 – Сокращения и термины

Обозначение	Расшифровка
МК	Микроконтроллер BE-U1000.
ОС	Операционная система, установленная на управляющем ПК.
ПК	Управляющий персональный компьютер.
AF 1/2/3/4	Альтернативные функции МК. Подробную информацию см. в документе *BE-U1000. Reference Manual*, раздел «Alternate Functions».
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 начальном загрузчике в соответствии со стандартом DFU v1.1.
eFlash	Встроенная флэш-память МК. Содержит две области – Main и NVR.
eFlash Main	Основная область eFlash, предназначенная для хранения программ или данных пользователя.
GDB	GNU Debugger – кроссплатформенный программный отладчик.
JTAG	Joint Test Action Group – стандартизированный 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 МК, предназначенная для хранения конфигурационных и пользовательских параметров, параметров МК и начального загрузчика.
NVR1	Служебный сектор eFlash для хранения заводских настроек. Пользователь ограничен в возможности записи и стирания данного сектора.
NVR2	Служебный сектор eFlash для хранения параметров начального загрузчика. Пользователь может ограничить запись и стирание данного сектора.
NVR3, NVR4	Служебный сектор eFlash для хранения параметров начального загрузчика.
OpenOCD	Open On-Chip Debugger – отладочный сервер, предназначенный для взаимодействия с JTAG-интерфейсом.
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.

Обозначение	Расшифровка
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 `0x40000000`, из которой происходит начальная загрузка МК. Для получения более подробной информации см. документ «*Микроконтроллер BE-U1000. Техническая спецификация*».
SPI	Serial Peripheral Interface (последовательный периферийный интерфейс) – это синхронный последовательный интерфейс для обмена данными между микроконтроллерами и периферией (датчики, дисплеи, SD-карты).
TCM	Tightly Coupled Memory (тесно связанная память) — память типа SRAM на МК 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.
XIP	Execute-In-Place – способ исполнения кода непосредственно из QSPI Flash, без предварительной загрузки в TCM.
XMODEM	Стандартный протокол передачи данных, реализованный в начальном загрузчике.

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 из ROM0.

В процессе исполнения начального загрузчика считывается состояние конфигурационных выводов и параметров конфигурации из eFlash NVR, после чего выбирается соответствующий сценарий загрузки.

При чтении параметров конфигурации из eFlash NVR, начальный загрузчик 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 сценариев загрузки.

3.2 Сценарии начальной загрузки

В данном разделе описаны алгоритмы работы МК с заводскими настройками при различных сценариях начальной загрузки.

Таблица 3-1 – Сценарии загрузки МК

Сценарий	Конфигурационные выводы		Описание
	DBG 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 интерфейс командной строки начального загрузчика.
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 интерфейс командной строки начального загрузчика.
JTAG EXT	0	100	Сценарий работы с внешним JTAG-отладчиком.
UART MP	0	101	Сценарий работы со встроенным интерпретатором MicroPython через UART0.
MULTI	0	110	Поочередные попытки запуска в сценариях: 1. «EFLASH» 2. «QSPI» 3. «UART»
FACTORY RESET	0	111	Сброс параметров eFlash NVR до заводских и переход в сценарий «USB»

Сценарий	Конфигурационные выводы		Описание
	DBG PC[8]	**MODE[2:0]** PC[4],PB[10],PA[10]	
USB CDC	1	001	Циклические попытки загрузки: - интерфейс командной строки начального загрузчика через USB по протоколу CDC - UART (интерфейс командной строки начального загрузчика или XMODEM)
JTAG INT	1	100	Работа со встроенным USB JTAG-отладчиком, запускаемом на ядре Core 2. Доступна отладка на ядрах Core 0 и Core 1.
USB MP	1	101	Сценарий работы со встроенным интерпретатором MicroPython через USB CDC.
MULTI USB	1	110	Попытки запуска в сценариях: 1. «EFLASH» 2. «QSPI» 3. Циклические попытки загрузки в сценариях: - «USB CDC» - «UART»
Для комбинаций DBG и MODE, не указанных в данной таблице, сценарий работы не определен.			

3.2.1 Сценарий загрузки «EFLASH»

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

Начальный загрузчик проверяет наличие кодового слова `0x0BEF5262` по адресу `0xa0000020` (eFlash). Если кодовое слово присутствует – начальный загрузчик передает управление программе, записанной в eFlash, по адресу `0xa0000030`.

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

3.2.2 Сценарий загрузки «UART»

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

Алгоритм сценария «UART»:

1. Начальный загрузчик 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. Начальный загрузчик ожидает поступления данных по протоколу XMODEM, либо получения символа `d` (код `0x64`). В случае получения символа `d` начальный загрузчик переходит в интерфейс командной строки.

3. В случае поступления данных по XMODEM, начальный загрузчик записывает эти данные в TCM по адресу `0x40010000`. После успешного принятия данных осуществляется проверка наличия кодового слова `0x0B10D522` по адресу `0x40010020`. Если слово присутствует – начальный загрузчик передает управление принятой программе в TCM по адресу `0x40010030`.

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

3.2.3 Сценарий загрузки «QSPI»

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

Алгоритм сценария «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 начальным загрузчиком. После получения управления программа может использовать все 160 кБ TCM.
3. Если кодовое слово отсутствует или произошла ошибка чтения – переход в сценарий «UART».



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

3.2.4 Сценарий загрузки «USB»

Данный сценарий предназначен для записи образа программы в TCM, eFlash или QSPI Flash по USB с помощью протокола DFU с возможностью исполнения записанной программы. Возможно использование интерфейса командной строки начального загрузчика по протоколу CDC.

В данном сценарии МК по USB определяется как составное USB-устройство с функциями DFU и CDC.

CDC используется для входа в интерфейс командной строки начального загрузчика при передаче четыре раза подряд символа `d`.

DFU используется для записи программ или изменения настроек МК. 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 начальным загрузчиком. После получения управления программа может использовать все 160 кБ TCM.

Alt=1: Перезапуск с выбором сценария загрузки

1. Принимается 1 байт
2. Перезапуск МК с выбором сценария загрузки в соответствии со значением принятого байта:

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

Бит `[3]`:

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

`0x1` – при загрузке со сценарием «USB CDC» не будет происходить переход в сценарий «UART».

Бит `[4]` – эквивалентно установке вывода PC[8] (см. таблицу 3-1).

Выбранный сценарий действует на один перезапуск, после следующей перезагрузки МК сценарий загрузки определяется конфигурационными 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 со сценарием загрузки «EFLASH».

Выбранный сценарий действует на один перезапуск, после следующей перезагрузки МК сценарий загрузки определяется конфигурационными 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 сценарии.

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 сценарии.

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 сценарии.

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 сценарии.

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 сценарии.

3.2.5 Сценарий загрузки «JTAG EXT»

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

Начальный загрузчик выполняет настройку выводов PA[0] – PA[4] для работы по сценарию «JTAG» и ожидает подключения внешнего отладчика.

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

Старт со сценарием «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.

`_boot.py` – это пользовательский python-скрипт, который будет выполнен (при его наличии) до перехода в REPL. Если исполнение пользовательского скрипта не требуется, удалите `_boot.py`, или оставьте его пустым.

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 Сценарий загрузки «MULTI»

В данном сценарии осуществляются попытки поочередной загрузки в сценариях «EFLASH», «QSPI», «UART».

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

3.2.8 Сценарий загрузки «JTAG INT»

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

В данном сценарии МК 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 интерфейсе командной строки начального загрузчика. Вход выполняется при передаче четыре раза подряд символа `d`. Через CDC возможен вывод трассы с ядер Core 0 и Core 1.

3.2.9 Сценарий загрузки «USB CDC»

Данный сценарий предназначен для доступа к интерфейсу командной строки начального загрузчика через USB по протоколу CDC.

Ниже приведен алгоритм работы в сценарии «USB CDC».

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

   Если символ `d` получен – подключение к интерфейсу командной строки начального загрузчика по протоколу CDC.

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

2. 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 интерфейсу командной строки начального загрузчика по UART.

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

3.2.10 Сценарий загрузки «Factory reset»

При данном сценарии выполняется очистка eFlash Main, NVR3, NVR4.

3.2.11 Сценарий загрузки «Multi USB»

При данном сценарии осуществляются попытки поочередной загрузки в различных сценариях.

1. Загрузка в сценарии «EFLASH». Если кодовое слово не найдено – переход в сценарий «QSPI».
2. Загрузка в сценарии «QSPI». Если кодовое слово не найдено – переход сценарий «USB CDC».
3. Загрузка в сценарии «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 сценарий «UART».
4. Загрузка в сценарии «UART». Если символ `d` не получен в течение 5 секунд и не началась передача по XMODEM – переход к п.3.

3.3 Интерфейс командной строки начального загрузчика

В сценариях загрузки «UART» или «USB» можно перейти в интерфейс командной строки начального загрузчика. Для входа в интерфейс необходимо в консоль передать символ `d` (для сценария загрузки «UART» или «USB CDC» – 1 раз, для сценария «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 интерфейс командной строки:

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

Интерфейс командной строки работает по следующему принципу: задаются значения параметров, после чего они могут быть использованы при выполнении функции.

Ниже приведено описание параметров и функций для ввода команд в интерфейс командной строки.

3.3.1 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
```

3.3.2 Функции

Функции предназначены для совершения различных операций в интерфейсе командной строки начального загрузчика. Некоторые функции для выполнения требуют наличия определенных данных, записанных в параметры `p0`, `p1`, `p2`, `p3`.

Ниже представлено краткое описание функций интерфейса командной строки.

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 интерфейса командной строки 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 интерфейса командной строки 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.

3.3.3 Расширенные функции (`m`)

Расширенные функции вызываются при вводе **`m`** и определяются параметром **`p3`**.
Данная функция использует `p0`, `p1`, `p2` в качестве вспомогательных параметров, в которых хранятся значения.



Пример команды:
`20048:13000000::11m`, где:

`11` – функция установки параметров в NVR
`M` – выполнение специальной функции
`20048` – маска записи в NVR+ID изменяемого параметра (`20000`+`48`)
`13000000` – новое значение параметра

`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 сценария загрузки. Зависит от значения `p2`. Выбранный сценарий будет активирован после выхода из интерфейса командной строки с помощью команды `o` или `q`.

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

Бит `[3]`:

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

`0x1` – при загрузке в сценарии «USB CDC» не будет происходить переход в сценарий «UART»

Бит `[4]` – эквивалентно установке вывода PC[8] (см. таблицу 3-1)



При настройках загрузчика по умолчанию переход в режим «USB MP» с помощью данной команды невозможен. Для перехода в этот режим можно воспользоваться функцией `g` с адресом `40006b10` (команда `40006b10g`).

`10` – Вывод параметров из NVR (PMIDs). Зависит от значения `p0`.
`p0`: `0x20000` + ID параметра.

`11` – Установка параметров в NVR (PMIDs). 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 временны́х параметров eFlash для частоты, задаваемой в `p2` (в Гц):
`0` – применяется значение из начального загрузчика (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.

3.4 Состояние МК после начальной загрузки

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

Таблица 3-2 – Состояния МК после начальной загрузки

Параметр	Сценарий загрузки									
	EFLASH	UART	QSPI	USB	JTAG EXT	UART MP	MULTI	USB CDC	JTAG INT	USB MP
Частота PLL, МГц	Float	Float	Float	100	Float	Float	Float/100	100	100	100
Частота CCLK, МГц	25	25	25	100	25	25	25/100	100	100	100
Частота PCLK0 – PCLK2, МГц	25	25	25	25	25	25	25	25	25	25
Частота HCLK, МГц	25	25	25	100	25	25	25/100	100	100	100
Состояние UART0	OFF/ON	ON	OFF	OFF	OFF	ON	ON	OFF/ON	OFF	OFF
Частота QSPI1, МГц	OFF	1,5625	1,5625	1,5625	OFF	1,5625	1,5625	OFF	OFF	1,5625
Состояние USB	OFF/HS	OFF	OFF	HS	OFF	OFF	OFF/HS	HS	HS	HS
Частота CCLK, для которой задаются временны́е параметры eFlash, МГц	50	50	50	100	50	50	50	100	100	100
Состояние выводов PA[0] – PA[4]	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	AF 4	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down

© 2025 АО «БАЙКАЛ ЭЛЕКТРОНИКС»

Параметр	Сценарий загрузки									
	EFLASH	UART	QSPI	USB	JTAG EXT	UART MP	MULTI	USB CDC	JTAG INT	USB MP
Состояние вывода PA[5]	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down	IN Pull Down
Состояние выводов PA[6] – PA[7]	AF 1	AF 1	IN Pull Down	IN Pull Down	IN Pull Down	AF 1	AF 1	AF 1	IN Pull Down	IN Pull Down
Состояние выводов PB[0] – PB[5]	IN Pull Down	AF 2	AF 2	AF 2	IN Pull Down	AF 2	AF 2	IN Pull Down	IN Pull Down	AF 2
Состояние выводов PB[6] – PB[7]	IN Pull Down	IN_NOPULL	IN_NOPULL	IN_NOPULL	IN Pull Down	IN_NOPULL	AF 1	IN Pull Down	IN Pull Down	IN_NOPULL

1. В сценариях «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 сценарии «UART» или «USB», будет инициализирован QSPI1.
3. В сценарии «EFLASH» тактирование и USB инициализируются, если не удалось загрузить образ с eFlash Main.
4. В сценарии «MULTI» клоки и USB инициализируются, если не удалось загрузить образ с eFlash Main и QSPI1.
5. Для временны́х параметров указана соответствующая им частота. Подробнее об их настройке см. в документе «*BE-U1000. Reference Manual*».

4 SDK

4.1 Описание SDK

SDK (Software Development Kit) – набор программных модулей и утилит, предназначенный для разработки программ 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
│  ├─ Components      : Драйвера компонентов отладочных плат
│  └─ <board_name>    : Файлы для конкретных плат
├─ Drivers            : Драйверы периферийных модулей МК
│  ├─ BASIS           : Макросы, определения, исходные и заголовочные файлы
│  └─ HAL             : Драйверы аппаратных абстракций
├─ Middlewares
│  └─ Third_Party     : Стороннее ПО: TinyUSB, FreeRTOS, и т.д.
├─ Projects
│  ├─ _template       : Шаблон для создания нового проекта программы
│  └─ HAL_examples    : Примеры программ с использованием HAL
└─ Tools
   ├─ build           : Системные файлы сборки и шаблон Makefile
   ├─ IDE             : Шаблоны для IDE
   ├─ OpenOCD         : Отладчик OpenOCD
   ├─ utils           : Вспомогательные скрипты
   ├─ svd             : Файлы System View Description
   └─ toolchain       : Средства компиляции и отладки
```

В составе SDK предоставлен набор инструментов кросс-компиляции и отладки для архитектуры RISC-V, расположенный в директории `Tools\toolchain\riscv32-none-elf\bin`. Основным средством отладки является утилита GDB. Подробное руководство по работе с отладчиком, включая пошаговое выполнение кода, приведено в официальной документации GDB.

4.2 Подготовка SDK

4.2.1 Загрузка SDK

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

Для загрузки SDK выполните следующие действия:

1. Откройте страницу mcu.baikalelectronics.ru, раздел Разработчикам.
2. Выберите необходимую версию из списка доступных для скачивания версий SDK.
3. На открывшейся странице выберите необходимые файлы и загрузите их нажатием правой кнопки мыши, или нажмите Скачать.

4.2.2 Распаковка SDK

4.2.2.1 ОС Windows



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

1. Распакуйте архив в рабочую директорию (далее – `[SDK_Path]`).
2. Перейдите в директорию `[SDK_Path]\Tools\toolchain\xpack-windows-build-tools` и выполните файл `add_to_path.bat` от имени администратора для установки переменных среды и работы с утилитой `Make`. В качестве альтернативы исполнения bat-файла можно добавить в системную переменную среды `Path` путь: `[SDK_Path]\Tools\toolchain\xpack-windows-build-tools\bin`.

4.2.2.2 ОС Linux

1. Перейдите в директорию с архивом:

```
cd [путь к директории]
```

2. Распакуйте 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: Создание директории для хранения проекта

Создайте директорию для хранения проекта. Для упрощения установки новых версий SDK, директорию рекомендуется создавать вне директории с SDK.

Шаг 2: Добавление файлов проекта в директорию

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

Шаг 3: Создание `makefile`

Для сборки образа в директории с проектом создайте сборочный скрипт `makefile`, включающий файл общих правил сборки (`$(SDK_DIR)/Tools/build/common.mk`), и добавьте 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 представлены примеры сборки образов для типов памяти TCM, eFlash и QSPI. Для записи в QSPI Flash с последующим запуском из TCM используйте сборку для TCM. Для записи в QSPI Flash с последующим запуском непосредственно из QSPI Flash используйте соответствующий раздел данной главы.

Сборка образов показана на примере `GPIO_LEDBlink` 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: Запуск терминала командной строки

Запустите `cmd.exe` в Windows или `bash` в Linux.

Шаг 2: Переход в директорию с проектом

Для Linux:

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

Для Windows:

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

Шаг 3: Сборка образа

Удалите результаты предыдущих сборок (если сборка производилась ранее):

```
make clean
```

Запустите сборку:

```
make MEM_REG_ROM=EFLASH
```

Бинарные файлы (`GPIO_LEDBlink.elf`, `GPIO_LEDBlink.bin`) по умолчанию будут сгенерированы в директории `output/debug`.

6.2 Сборка образа для исполнения из TCM

Шаг 1: Запуск терминала командной строки

Запустите `cmd.exe` в Windows или `bash` в Linux.

Шаг 2: Переход в директорию с проектом

Для Linux:

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

Для Windows:

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

Шаг 3: Сборка образа

Удалите результаты предыдущих сборок (если сборка производилась ранее):

```
make clean
```

Запустите сборку:

```
make MEM_REG_ROM=TCMA
```

Файлы (`GPIO_LEDBlink.elf`, `GPIO_LEDBlink.bin`) по умолчанию будут сгенерированы в директории `output/debug`.

6.3 Сборка образа для исполнения из QSPI Flash (XIP)



Описанные ниже способы сборки корректны для памяти *Micron MT25QL128ABA*. Корректная работа с другими чипами памяти не гарантируется!

Образ будет собран для XIP (execute-in-place) – способа исполнения кода непосредственно с QSPI Flash, без предварительной загрузки в TCM.

Для запуска пользовательской программы в режиме XIP необходима утилита `QSPI_XIPLoader` из состава SDK. `QSPI_XIPLoader` позволяет запускать в режиме XIP пользовательскую программу, записанную на QSPI Flash. `QSPI_XIPLoader` расположен в директории `[SDK_Path]\Projects\HAL_examples\QSPI\QSPI_XIPLoader`.

`QSPI_XIPLoader` должен быть записан в QSPI Flash с нулевым смещением, пользовательская программа должна быть записана в QSPI Flash с некоторым смещением (по умолчанию – `0x10000`). Величина смещения должна совпадать со значением параметра `QSPI_APP_OFFSET`, устанавливаемого в `Makefile` проекта `QSPI_XIPLoader`.

Алгоритм работы `QSPI_XIPLoader`:

1. Запуск на TCM после начальной загрузки МК в сценарии «QSPI» (см. раздел Сценарий загрузки «QSPI»).
2. Инициализация тактовых сигналов.
3. Инициализация QSPI Flash.
4. Переключение QSPI Flash в режим XIP.

5. Передача управления пользовательской программе по адресу, определенному параметром `QSPI_APP_OFFSET`.

6.3.1 Ручная сборка образа для исполнения из QSPI Flash (XIP)

Данный алгоритм сборки является альтернативным автоматической сборке.

Оба раздела ниже (6.3.1.1 и 6.3.1.2) являются обязательными к выполнению.

6.3.1.1 Сборка пользовательской программы

Шаг 1: Запуск терминала командной строки

Запустите `cmd.exe` в Windows или `bash` в Linux.

Шаг 2: Переход в директорию с проектом

Для Linux:

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

Для Windows:

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

Шаг 3: Сборка образа

Удалите результаты предыдущих сборок (если сборка производилась ранее):

```
make clean
```

Запустите сборку:

```
make MEM_REG_ROM=QSPI1
```

Бинарные файлы (`GPIO_LEDBlink.elf`, `GPIO_LEDBlink.bin`) по умолчанию будут сгенерированы в директории `/output/debug`.

6.3.1.2 Сборка XIP-загрузчика

Шаг 1: Запуск терминала командной строки

Запустите `cmd.exe` в Windows или `bash` в Linux.

Шаг 2: Переход в директорию с проектом

Для Linux:

```
cd [SDK_Path]/Projects/HAL_examples/QSPI/QSPI_XIPLoader
```

Для Windows:

```
cd [SDK_Path]\Projects\HAL_examples\QSPI\QSPI_XIPLoader
```

Шаг 3: Редактирование `Makefile` (необязательно)

В директории проекта откройте `Makefile` и установите для параметра `QSPI_APP_OFFSET` (адрес пользовательской программы) необходимое значение смещения.



При использовании DFU для загрузки рекомендуется использовать значение смещения, установленное по умолчанию для DFU «Alt=7» – `0x2000000`.

Шаг 4: Сборка образа

Удалите результаты предыдущих сборок (если сборка производилась ранее):

```
make clean
```

Запустите сборку:

```
make
```

Бинарные файлы (`QSPI_XIPLoader.elf`, `QSPI_XIPLoader.bin`) по умолчанию будут сгенерированы в директории `/output/debug`.

6.3.2 Автоматическая сборка образа для исполнения из QSPI Flash (XIP)

Данный алгоритм сборки является альтернативным ручной сборке.

При использовании Windows перед началом работы необходимо установить на ПК интерпретатор `bash`-скриптов (например, `Git Bash`).

Шаг 1: Запуск терминала командной строки

Запустите `cmd.exe` в Windows или `bash` в Linux.

Шаг 2: Переход в директорию с проектом

Для Linux:

```
cd [SDK_Path]/Tools/scripts
```

Для Windows:

```
cd [SDK_Path]\Tools\scripts
```

Шаг 3: Редактирование скрипта

Откройте файл `qspi_img.sh` и при необходимости отредактируйте параметры `SCRIPT_DIR`, `SDK_ROOT`, `XIP_LOADER_DIR`, `APP_PROJECT_DIR`, `QSPI_IMAGE_NAME`, `QSPI_APP_OFFSET`.

Шаг 4: Выполнение скрипта сборки

Введите команду:

```
qspi_img.sh -a $(realpath .) -o ./output/qspi_img.bin
```

При использовании Windows может возникнуть необходимость выбора программы по умолчанию, используемую для `bash`-скриптов.

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 7-1 находятся ссылки для быстрого перехода к нужному разделу.

Таблица 7-1 – Быстрый переход

ОС	Способ записи (сценарий)	Тип памяти			
		TCM	eFlash	QSPI Flash (TCM)[1]	QSPI Flash (XIP)[2]
Windows	DFU	7.1.1	7.1.2	7.1.3	7.1.4
	UART	7.1.5	7.1.6	7.1.7	7.1.8
	JTAG	7.1.9	7.1.10	7.1.11	7.1.12
Linux	DFU	7.2.1	7.2.2	7.2.3	7.2.4
	UART	7.2.5	7.2.6	7.2.7	7.2.8
	JTAG	7.2.9	7.2.10	7.2.11	7.2.12

[1] Способ записи программы в QSPI Flash с ее исполнением из TCM.
[2] Способ записи программы в QSPI Flash с ее исполнением непосредственно из QSPI Flash.

7.1 ОС Windows

7.1.1 Работа с TCM через DFU

⚠ Память TCM очищается после снятия питания.

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

Шаг 1: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «DFU» (см. главу Начальная загрузка).

Шаг 2: Запуск терминала командной строки

Запустите `cmd.exe`.

Шаг 3: Вывод списка DFU-устройств (необязательно)

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 проверьте корректность подключения и выбора сценария загрузки МК, после чего повторите попытку.

Шаг 4: Запись образа

Запишите образ в память МК с указанием режима записи `alt 0` (запись в ТСМ с исполнением):

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

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

После загрузки образа управление сразу передается ТСМ.

7.1.2 Работа с eFlash через DFU

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

Шаг 1: Подготовка

1. Выполните сборку образа для eFlash.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «DFU» (см. главу Начальная загрузка).

Шаг 2: Запуск терминала командной строки

Запустите `cmd.exe`.

Шаг 3: Вывод списка DFU-устройств (необязательно)

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 проверьте корректность подключения и выбора сценария загрузки МК, после чего повторите попытку.

Шаг 4: Запись образа

Запишите 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 укажите устройство (например `--path 1-2`).

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

При вводе `-alt 3` после загрузки образа МК продолжает работу в текущем сценарии. Для исполнения программы установите сценарий загрузки «EFLASH» и перезагрузите МК.

7.1.3 Работа с QSPI Flash / TCM через 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: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «DFU» (см. главу Начальная загрузка).

Шаг 2: Запуск терминала командной строки

Запустите `cmd.exe`.

Шаг 3: Вывод списка DFU-устройств (необязательно)

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 проверьте корректность подключения и выбора сценария загрузки МК, после чего повторите попытку.

Шаг 4: Запись образа

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

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

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

Максимальный размер образа – 16 МБ.

Для исполнения пользовательской программы установите сценарий загрузки «QSPI» и перезагрузите МК.

7.1.4 Работа с QSPI Flash / XIP через DFU

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

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

Шаг 1: Подготовка

1. Выполните сборку образов для XIP.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «DFU» (см. главу Начальная загрузка).

Шаг 2: Запуск терминала командной строки

Запустите `cmd.exe`.

Шаг 3: Вывод списка DFU-устройств (необязательно)

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 проверьте корректность подключения и выбора сценария загрузки МК, после чего повторите попытку.

Шаг 4: Запись образа пользовательской программы

Запишите образ в QSPI Flash с указанием режима записи:

- `alt 7` – запись со смещением (по умолчанию – `0x200000`) – используется в случае ручной сборки образа:

```
dfu-util --alt 7 --download [путь к директории с образом]\main.bin
```

- `alt 6` – запись без смещения – используется в случае автоматической сборки образа:

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

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

Максимальный размер образа – 14 МБ (при значении смещения по умолчанию).

Шаг 5: Запись образа XIP-загрузчика (в случае ручной сборки образа)

Запишите образ программы `QSPI_XIPLoader` в QSPI Flash с указанием режима записи `alt 6` – запись без смещения:

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

Для исполнения пользовательской программы установите сценарий загрузки «QSPI» и перезагрузите МК. `QSPI_XIPLoader` будет скопирован в ТСМ и запущен. После запуска он выполнит необходимые настройки и передаст управление пользовательской программе.

7.1.5 Работа с ТСМ через UART

⚠ Память ТСМ 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: Подготовка

1. Выполните сборку образа для ТСМ.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Подключение к интерфейсу МК «UART0»

1. Подключитесь к МК через виртуальный COM-порт.
2. Запустите `ExtraPuTTY` и подключитесь к интерфейсу МК «UART0».

Установите 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

Установите параметр для корректного отображения конца строки:

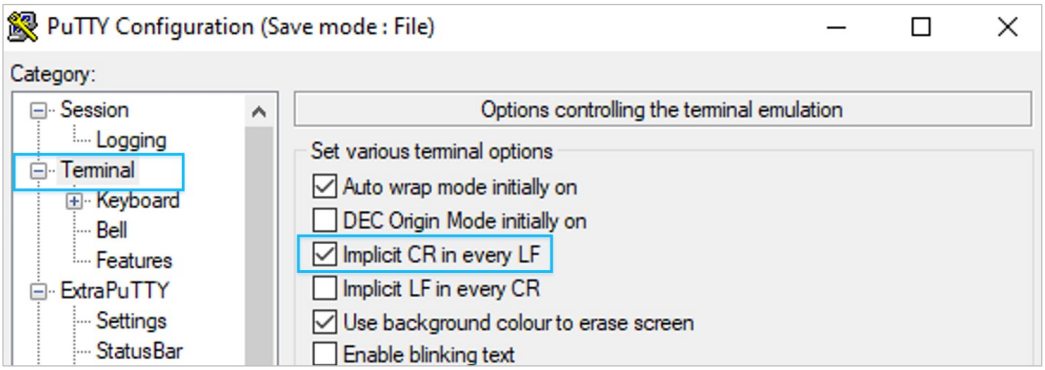


Рисунок 7-1 – Настройка `ExtraPutty`

После настройки подключения нажмите Open:

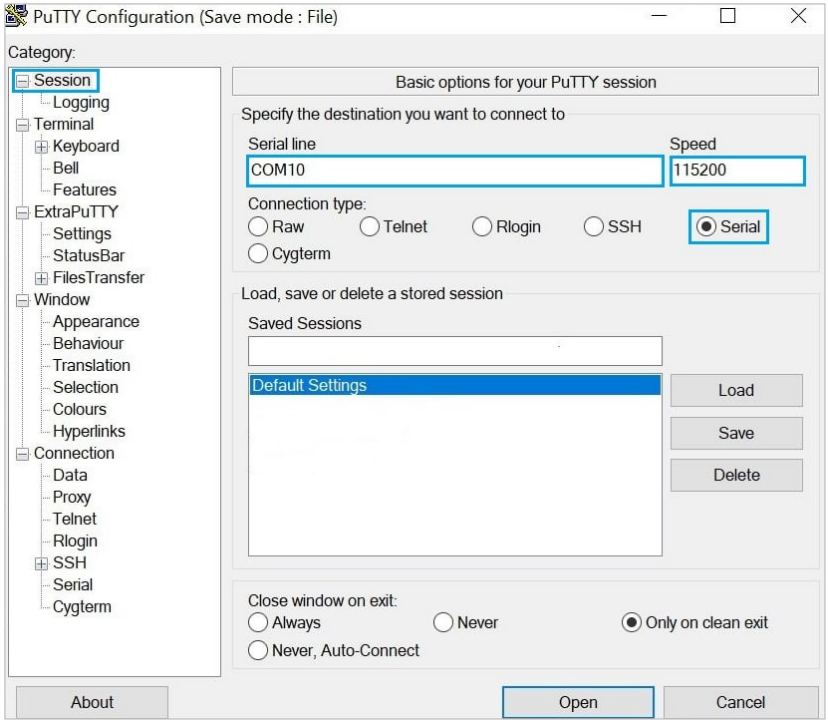


Рисунок 7-2 – Подключение к МК

Откроется терминал и отобразятся следующие сообщения:

```
CC
```

Шаг 3: Вход в интерфейс командной строки начального загрузчика

1. На плате нажмите кнопку *RESET*.
2. Нажмите и удерживайте клавишу <D> до появления сообщения:

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

3. Удалите лишние символы `d`.

Шаг 4: Запись образа в память МК

1. Нажмите клавишу <L>. 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 интерфейс командной строки начального загрузчика необходимо ввести команду «`:[размер]`» и нажмите <Enter>. Значение вводится в шестнадцатеричном формате.

2. В интерфейсе `ExtraPutty` Нажмите Files Transfer → XMODEM → Send:

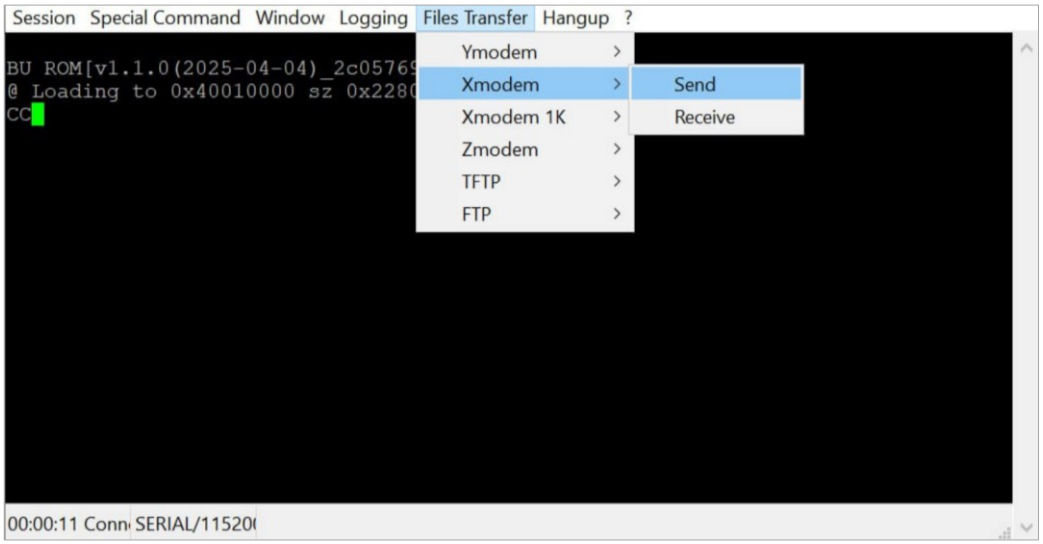


Рисунок 7-3 – Передача по XMODEM

3. В открывшемся окне выберите файл образа с расширением `.bin`.
4. Дождитесь окончания передачи файла.

В UART-консоли появится символ `@`.

> ⚠ В случае появления сообщений об ошибке передачи повторите запись, начиная с входа в интерфейс командной строки начального загрузчика.

Шаг 5: Запуск исполнения программы

Нажмите клавишу <G>. Начнется исполнение кода.

7.1.6 Работа с eFlash через 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: Подготовка

1. Выполните сборку образа для eFlash.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Подключение к интерфейсу МК «UART0»

1. Подключитесь к МК через виртуальный COM-порт.
2. Запустите `ExtraPutty`
3. Подключитесь к интерфейсу МК «UART0».

Установите 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

Установите параметр для корректного отображения конца строки:

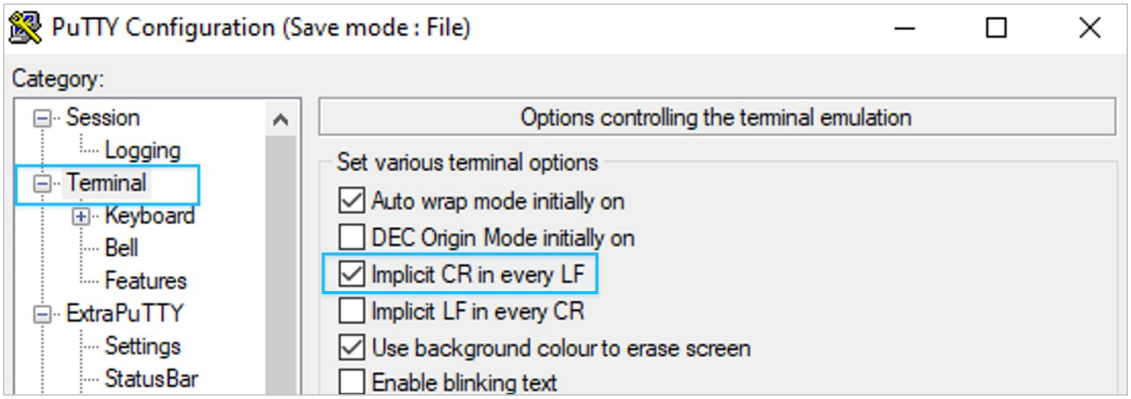


Рисунок 7-4 – Настройка `ExtraPutty`

После настройки подключения нажмите Open:

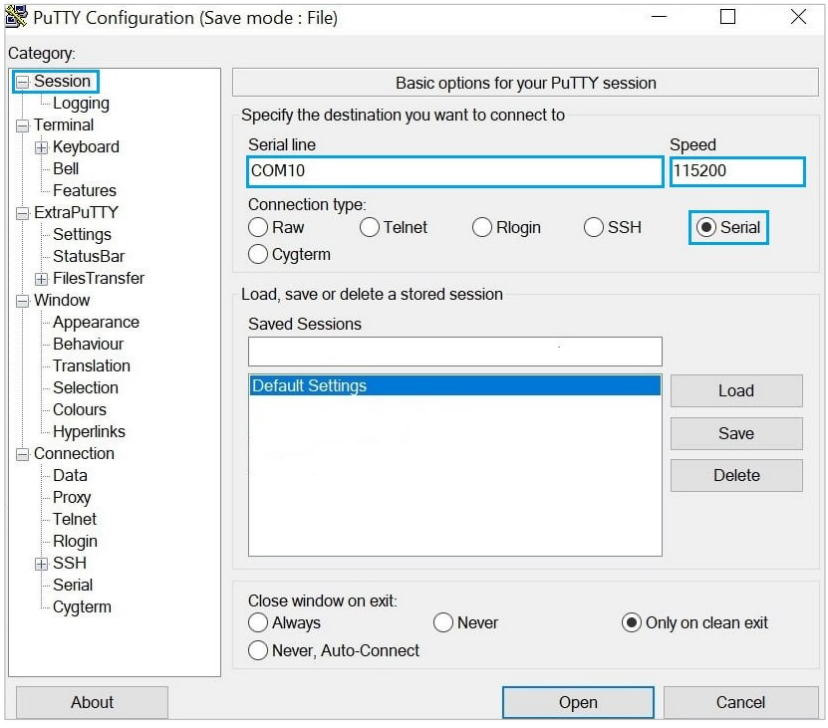


Рисунок 7-5 – Подключение к МК

Откроется терминал и отобразятся следующие сообщения:

```
CC
```

Шаг 3: Вход в интерфейс командной строки начального загрузчика

1. На плате нажмите кнопку *RESET*.
2. Нажмите и удерживайте клавишу <D> до появления сообщения:

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

3. Удалите лишние символы `d`.

Шаг 4: Снятие защиты с eFlash

Снимите защиту с eFlash командой:

```
14000018:1w
```

Шаг 5: Установка параметров

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

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

Шаг 6: Очистка eFlash

Очистите eFlash при помощи команды:

```
:::101m
```

Шаг 7: Включение ожидания отправки

Дайте команду на ожидание файла образа по протоколу XMODEM:

```
:::3L
```

Шаг 8: Загрузка программы в память МК

1. В интерфейсе `ExtraPutty` нажмите Files Transfer → XMODEM → Send:

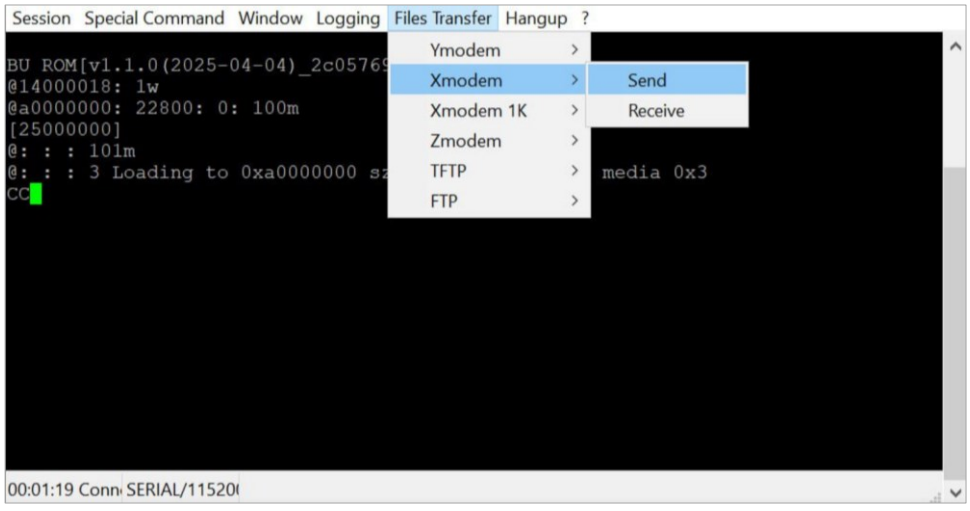


Рисунок 7-6 – Передача по XMODEM

2. В открывшемся окне выберите файл образа с расширением `.bin`.
3. Дождитесь окончания передачи файла.

В UART-консоли появится символ `@`.



В случае появления сообщений об ошибке передачи повторите запись, начиная с входа в интерфейс командной строки начального загрузчика (Шаг 3).

Шаг 9: Запуск исполнения программы

Способ 1 - Из интерфейса командной строки начального загрузчика:

В интерфейсе укажите адрес eFlash-памяти:

```
a0000000g
```

Способ 2 - Установкой конфигурационных выводов:

1. Снимите питание с платы.
2. Установите сценарий загрузки «EFLASH».
3. Подайте питание на плату.

7.1.7 Работа с QSPI Flash / TCM через 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: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Подключение к интерфейсу МК «UART0»

1. Подключитесь к МК через виртуальный COM-порт.
2. Откройте утилиту `ExtraPutty` и подключитесь к интерфейсу МК «UART0».

Установите 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

Установите параметр для корректного отображения конца строки:

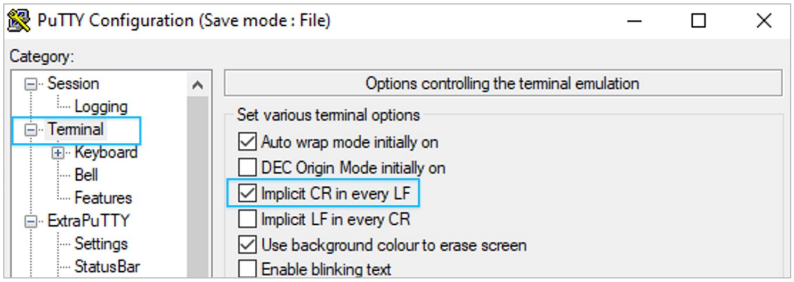


Рисунок 7-7 – Настройка `ExtraPutty`

После настройки подключения нажмите Open:

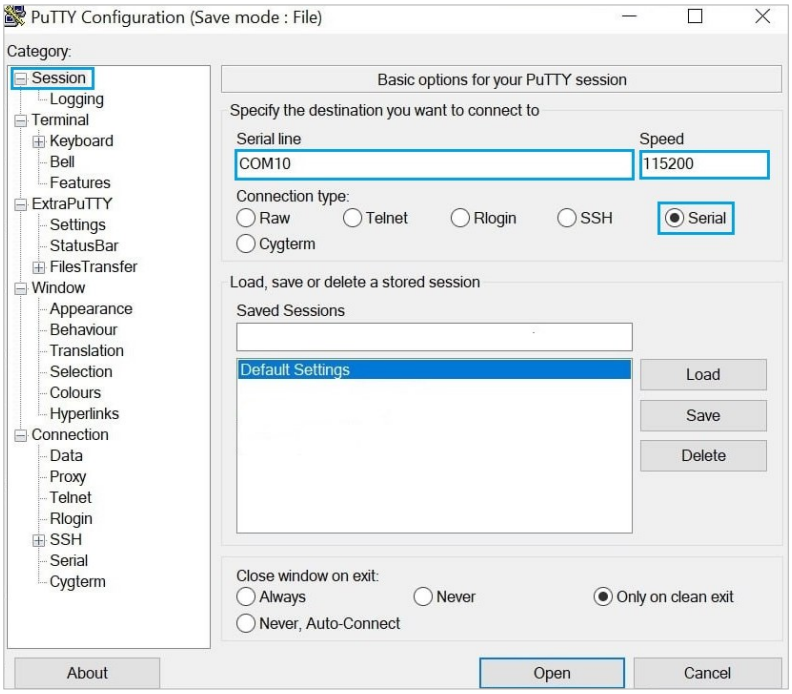


Рисунок 7-8 – Подключение к МК

Откроется терминал и отобразятся следующие сообщения:

```
CC
```

Шаг 3: Вход в интерфейс командной строки начального загрузчика

1. На плате нажмите кнопку *RESET*.
2. Нажмите и удерживайте клавишу <D> до появления сообщения:

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

3. Удалите лишние символы `d`.

Шаг 4: Запись образа в память МК

По умолчанию производится запись в QSPI1.

Для записи в QSPI0 необходимо ввести команду `20048:13000000::11m`.

Для записи в QSPI1 необходимо ввести команду `20048: 11000000::11m`.

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. В интерфейсе `ExtraPutty` нажмите Files Transfer → XMODEM → Send:

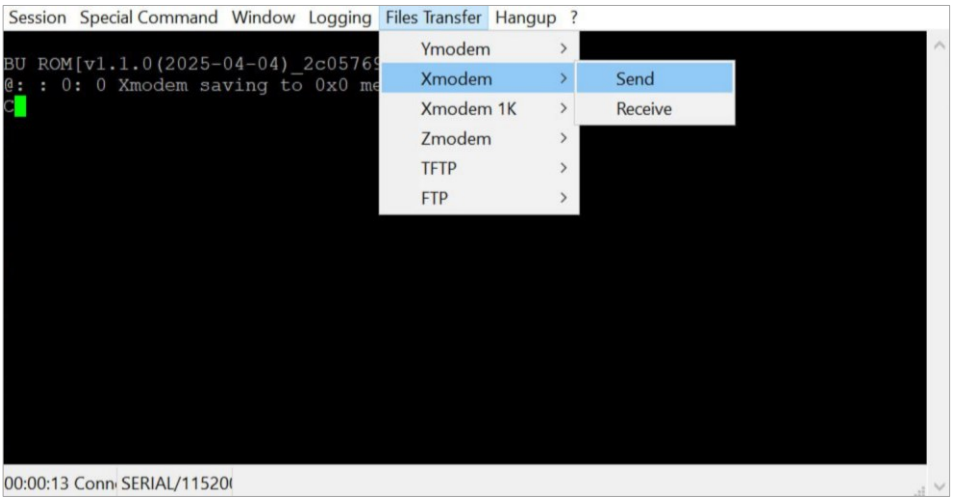


Рисунок 7-9 – Передача по XMODEM

3. В открывшемся окне выберите файл образа с расширением `.bin`.
4. Дождитесь окончания передачи файла.

В UART-консоли появится символ `@`.

В случае появления сообщений об ошибке передачи повторите запись, начиная с входа в интерфейс командной строки начального загрузчика.

Шаг 5: Запуск исполнения программы

1. Снимите питание с платы.
2. Установите на используемой плате сценарий загрузки «QSPI».
3. Подайте питание на плату.

7.1.8 Работа с QSPI Flash / XIP через UART

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

Перед началом работы необходимо установить на ПК программу удаленного терминала с поддержкой протокола XMODEM (например, `ExtraPuTTY`), а также загрузить и распаковать SDK.

В инструкции ниже в качестве примера используется `ExtraPuTTY`.

Шаг 1: Подготовка

1. Выполните сборку образов для XIP.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Подключение к интерфейсу МК «UART0»

1. Подключитесь к МК через виртуальный COM-порт.
2. Откройте утилиту `ExtraPutty` и подключитесь к интерфейсу МК «UART0».

Установите 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

Установите параметр для корректного отображения конца строки:

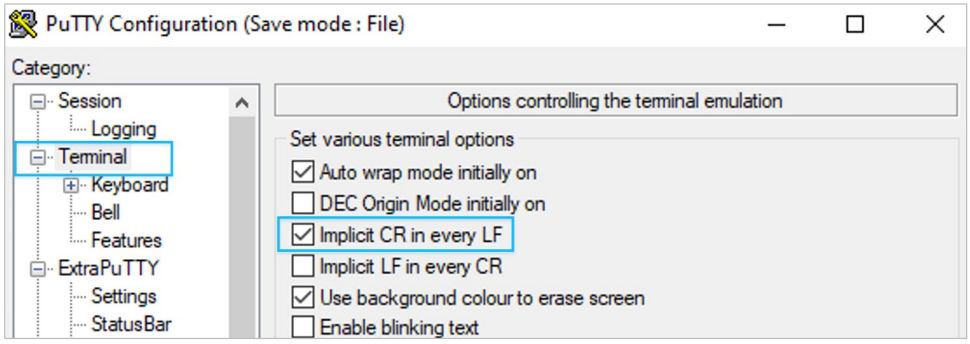


Рисунок 7-10 – Настройка `ExtraPutty`

После настройки подключения нажмите Open:

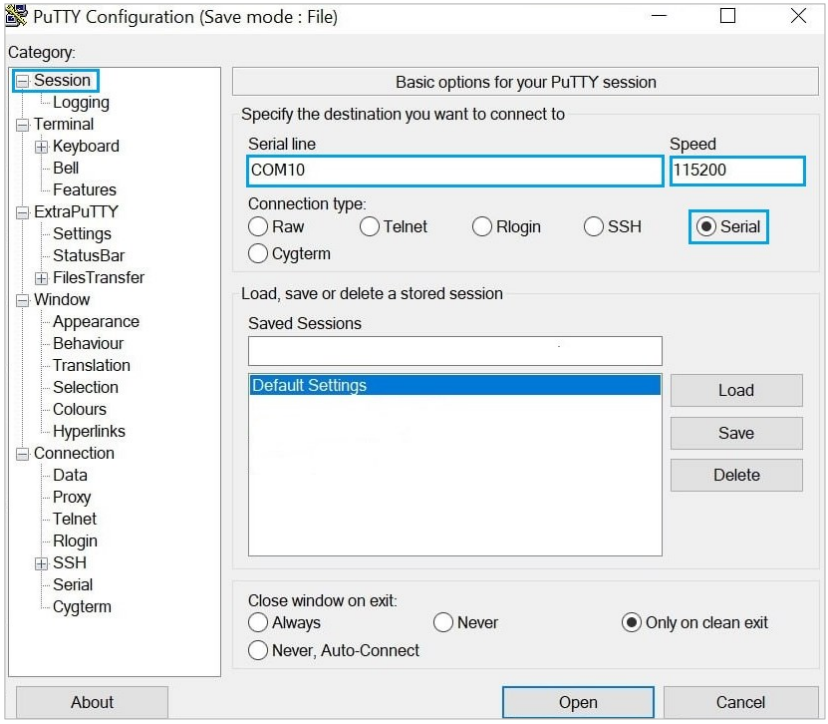


Рисунок 7-11 – Подключение к МК

Откроется терминал и отобразятся следующие сообщения:

```
CC
```

Шаг 3: Вход в интерфейс командной строки начального загрузчика

1. На плате нажмите кнопку *RESET*.
2. Нажмите и удерживайте клавишу <D> до появления сообщения:

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

3. Удалите лишние символы `d`.

Шаг 4: Запись образа пользовательской программы в память МК

1. Дайте команду на прием данных по XMODEM и сохранение их в QSPI Flash



По умолчанию производится запись в QSPI1.

Для записи в QSPI0 необходимо ввести команду `20048:13000000::11m`.

Для записи в QSPI1 необходимо ввести команду `20048: 11000000::11m`.

- В случае ручной сборки образа – со смещением (в данном примере – `0x10000`):

```
::10000:0x
```



Смещение должно совпадать со значением параметра `QSPI_APP_OFFSET`, устанавливаемого в `Makefile` проекта `QSPI_XIPLoader`.

Будет выведено сообщение о готовности к принятию файла:

```
@: : 10000: 0 Xmodem saving to 0x0 media 0x0
C
```

- В случае автоматической сборки образа – без смещения:

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

Будет выведено сообщение о готовности к принятию файла:

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

2. В интерфейсе `ExtraPutty` нажмите Files Transfer → XMODEM → Send:

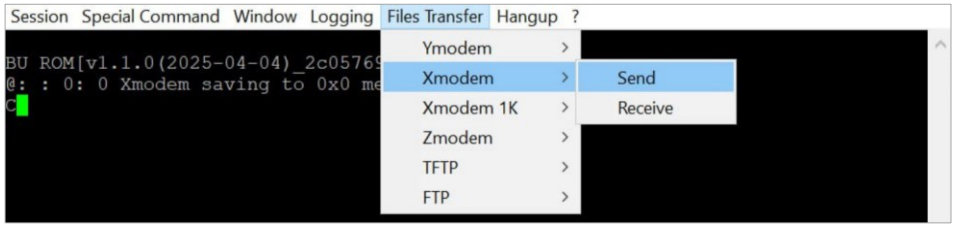


Рисунок 7-12 – Передача по XMODEM

3. В открывшемся окне выберите файл образа пользовательской программы с расширением `.bin`.
4. Дождитесь окончания передачи файла.

В UART-консоли появится символ `@`.

> ⚠ В случае появления сообщений об ошибке передачи повторите запись, начиная с вход в интерфейс командной строки начального загрузчика.

Шаг 5: Запись образа XIP-загрузчика в память МК (в случае ручной сборки образа)

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. В интерфейсе `ExtraPutty` нажмите Files Transfer → XMODEM → Send:

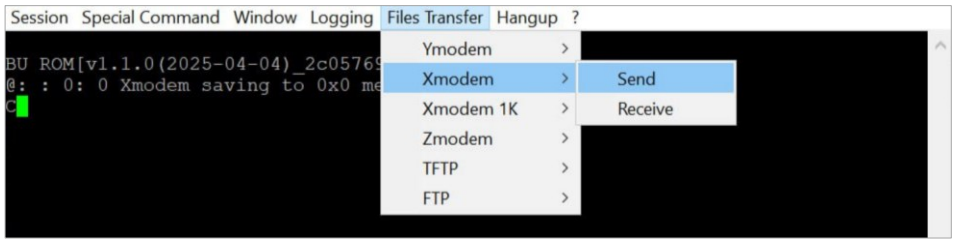


Рисунок 7-13 – Передача по XMODEM

3. В открывшемся окне выберите файл образа пользовательской программы с расширением `.bin`.
4. Дождитесь окончания передачи файла.

В UART-консоли появится символ `@`.



В случае появления сообщений об ошибке передачи повторите запись, начиная с входа в интерфейс командной строки начального загрузчика.

Шаг 6: Запуск исполнения программы

1. Снимите питание с платы.
2. Установите на используемой плате сценарий загрузки «QSPI».
3. Подайте питание на плату.

7.1.9 Работа с TCM через JTAG



В данном разделе представлена информация по записи программ в МК.
Подробное руководство по работе с отладчиком см. в документации на `GDB`.



Память TCM очищается после снятия питания.

Перед началом работы необходимо установить на ПК драйвер `WinUSB`, загрузить и распаковать SDK.

Шаг 1: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «JTAG» (см. главу Начальная загрузка).

Шаг 2: Запуск терминала командной строки

Запустите `cmd.exe`.

Шаг 3: Открытие директории с OpenOCD

Открытие директории с OpenOCD в терминале командной строки:

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

Шаг 4: Запуск OpenOCD

Запустите бинарный файл с указанием файла конфигурации для используемой платы:

```
.\openocd.exe -f ..\share\openocd\scripts\board\[путь к файлу конфигурации].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: Открытие второго окна терминала командной строки

Все дальнейшие действия выполняйте в данном окне.

Шаг 6: Откройте директорию с GDB

Во втором окне терминала перейдите к директории:

```
cd [SDK_Path]\tools\toolchain\riscv32-none-elf\bin
```

Шаг 7: Запуск отладчик GDB

Во втором окне терминала ввести команду:

```
[SDK_path]\Tools\toolchain\riscv32-none-elf\bin\riscv32-none-elf-gdb -x
[SDK_path]\Tools\scripts\bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 8: Запись образа через отладчик GDB

Запишите бинарный образ в память МК:

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

После успешной записи будут выведены сообщения вида:

```
Restoring binary file \[путь к директории с образом]\[имя образа].bin into memory
0x40010000 to 0xa00002f8)
```

Шаг 9: Запуск исполнения программы

1. Перейдите к адресу TCM:

```
bmcu jump 0x40010000
```

2. Дайте команду на исполнение:

```
continue
```



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



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.1.10 Работа с eFlash через JTAG



В данном разделе представлена информация по записи программ в МК.
Подробное руководство по работе с отладчиком см. в документации на GDB.

Перед началом работы необходимо установить на ПК драйвер WinUSB, загрузить и распаковать SDK.

Шаг 1: Подготовка

1. Выполните сборку образа для eFlash.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «JTAG» (см. главу Начальная загрузка).

Шаг 2: Запуск терминала командной строки

Запустите cmd.exe.

Шаг 3: Открытие директории с OpenOCD

В терминале командной строки откройте директорию с OpenOCD:

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

Шаг 4: Запуск OpenOCD

Запустите бинарный файл с указанием файла конфигурации:

```
.\openocd.exe -f ..\share\openocd\scripts\board\[путь к файлу конфигурации].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: Открытие второго окна терминала командной строки

Откройте cmd.exe в новом окне. Все дальнейшие действия выполняйте в данном окне.

Шаг 6: Открытие директории с GDB

Во втором окне терминала перейдите к директории:

```
cd [SDK_Path]\tools\toolchain\riscv32-none-elf\bin
```

Шаг 7: Запуск отладчика GDB

Во втором окне терминала ввести команду:

```
[SDK_path]\Tools\toolchain\riscv32-none-elf\bin\riscv32-none-elf-gdb -x
[SDK_path]\Tools\scripts\bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 8: Запись образа через отладчик GDB

1. Очистите память eFlash:

```
bmcu eFlash erase
```

2. Перезагрузите МК:

```
bmcu reset
```

3. Запишите бинарный образ в память МК:

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

После успешной записи будут выведены сообщения вида:

```
Restoring binary file \[путь к директории с образом]\[имя образа].bin into
memory (0xa0000000 to 0xa00002f8)
```

Шаг 9: Запуск исполнения программы

1. Перейдите к адресу eFlash:

```
bmcu jump 0xa0000000
```

2. Дайте 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.

7.1.11 Работа с QSPI Flash / TCM через JTAG



В данном разделе представлена информация по записи программ в МК.
Для получения информации по работе с отладчиком см. документацию на GDB.

Ниже описывается способ записи программы в QSPI Flash с ее исполнением из TCM.

Перед началом работы необходимо загрузить и распаковать SDK.

Шаг 1: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «JTAG» (см. главу Начальная загрузка).

Шаг 2: Запуск терминала командной строки

Запустите `cmd.exe`.

Шаг 3: Открытие директории с OpenOCD

В терминале командной строки откройте директорию с OpenOCD

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

Шаг 4: Запуск OpenOCD

Запустите бинарный файл с указанием файла конфигурации для используемой платы:

```
.\openocd.exe -f ..\share\openocd\scripts\board\[путь к файлу конфигурации].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: Открытие второго окна терминала командной строки

Откройте `cmd.exe` в новом окне. Все дальнейшие действия выполняйте в данном окне.

Шаг 6: Открытие директории с GDB

Во втором окне терминала перейдите к директории:

```
cd [SDK_Path]\tools\toolchain\riscv32-none-elf\bin
```

Шаг 7: Запуск отладчик GDB

Во втором окне терминала ввести команду:

```
[SDK_path]\Tools\toolchain\riscv32-none-elf\bin\riscv32-none-elf-gdb -x
[SDK_path]\Tools\scripts\bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 8: Запись образа через отладчик GDB

1. Инициализируйте QSPI Flash:

```
bmcu QSPI_clk_init
bmcu QSPI1 init
```

2. Очистите QSPI Flash (если требуется):

```
bmcu QSPI1 erase
```

3. Запишите бинарный образ в память МК:

```
bmcu flash write_bank 1 [путь к образу]\[имя образа].bin 0x90000000
bmcu reset
```

После успешной записи будут выведены сообщения вида:

```
Restoring binary file \[путь к директории с образом]\[имя образа].bin into
memory (0x90000000 to 0x900002f8)
```



Показана запись в QSPI1. Для записи в QSPI0 необходимо использовать адрес `0x90000000`.

Шаг 9: Запуск исполнения программы

1. Снимите питание с платы.
2. Установите на используемой плате сценарий загрузки «QSPI».
3. Подайте питание на плату.



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



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.1.12 Работа с QSPI Flash / XIP через JTAG



В данном разделе представлена информация по записи программ в МК.
Для получения информации по работе с отладчиком см. документацию на GDB.

Ниже описывается способ записи программы в QSPI Flash с ее исполнением непосредственно из QSPI Flash.

Перед началом работы необходимо загрузить и распаковать SDK.

Шаг 1: Подготовка

1. Выполните сборку образов для XIP.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Запуск терминала командной строки

Запустите `cmd.exe`.

Шаг 3: Открытие директории с OpenOCD

В терминале командной строки откройте директорию с OpenOCD:

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

Шаг 4: Запуск OpenOCD

Запустите бинарный файл с указанием файла конфигурации для используемой платы:

```
.\openocd.exe -f ..\share\openocd\scripts\board\[путь к файлу конфигурации].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: Открытие второго окна терминала командной строки

Откройте `cmd.exe` в новом окне. Все дальнейшие действия выполняйте в данном окне.

Шаг 6: Открытие директории с GDB

Во втором окне терминала перейдите к директории:

```
cd [SDK_Path]\tools\toolchain\riscv32-none-elf\bin
```

Шаг 7: Запуск отладчик GDB

Во втором окне терминала ввести команду:

```
[SDK_path]\Tools\toolchain\riscv32-none-elf\bin\riscv32-none-elf-gdb -x
[SDK_path]\Tools\scripts\bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 8: Запись образа через отладчик GDB

1. Инициализируйте QSPI Flash:

```
bmcu QSPI_clk_init
bmcu QSPI1 init
```

2. Очистите QSPI Flash (если требуется):

```
bmcu QSPI1 erase
```

3. Запишите бинарный образ в память МК

- В случае ручной сборки образа – со смещением (в данном примере – `0x10000`) от базового адреса QSPI Flash:

```
bmcu flash write_bank 1 [путь к образу]\[имя образа].bin 0x90001000
bmcu reset
```

- В случае автоматической сборки образа – без смещения:

```
bmcu flash write_bank 1 [путь к образу]\[имя образа].bin 0x90000000
bmcu reset
```

После успешной записи будут выведены сообщения вида:

```
Restoring binary file \[путь к директории с образом]\[имя образа].bin into
memory (0x90000000 to 0x900002f8)
```



Показана запись в QSPI1. Для записи в QSPI0 необходимо использовать адрес `0x90000000`.

Шаг 9: Запуск исполнения программы

1. Снимите питание с платы.
2. Установите на используемой плате сценарий загрузки «QSPI».
3. Подайте питание на плату.



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



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.2 ОС Linux

7.2.1 Работа с ТСМ через DFU

⚠ Память ТСМ очищается после снятия питания.

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

Шаг 1: Подготовка

1. Выполните сборку образа для ТСМ.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «DFU» (см. главу Начальная загрузка).

Шаг 2: Вывод списка DFU-устройств (необязательно)

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 проверьте корректность подключения и выбора сценария загрузки МК, после чего повторите попытку.

Шаг 3: Запись образа

Запишите образ в память МК с указанием режима записи `alt=0` (запись в ТСМ с исполнением):

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

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

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

При вводе `alt 3` после загрузки образа МК продолжает работу в текущем сценарии. Для исполнения программы установите сценарий загрузки «EFLASH» и перезагрузите МК.

7.2.2 Работа с eFlash через DFU

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

Шаг 1: Подготовка

1. Выполните сборку образа для eFlash.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «DFU» (см. главу Начальная загрузка).

Шаг 2: Вывод списка DFU-устройств (необязательно)

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 проверьте корректность подключения и выбора сценария загрузки МК, после чего повторите попытку.

Шаг 3: Запись образа

Запишите 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 укажите устройство (например `--path 1-2`).

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

При вводе `alt 3` после загрузки образа МК продолжает работу в текущем сценарии. Для исполнения программы установите сценарий загрузки «EFLASH» и перезагрузить МК.

7.2.3 Работа с QSPI Flash / TCM через 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: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «DFU» (см. главу Начальная загрузка).

Шаг 2: Вывод списка DFU-устройств (необязательно)

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 проверьте правильность подключения и выбора сценария загрузки МК, после чего повторите попытку.

Шаг 3: Запись образа

Запишите 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 укажите устройство (например `--path 1-2`).

Максимальный размер образа – 16 МБ.

Для исполнения программы установите сценарий загрузки «QSPI» и перезагрузите МК.

7.2.4 Работа с QSPI Flash / XIP через DFU

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

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

Шаг 1: Подготовка

1. Выполните сборку образа для XIP.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «DFU» (см. главу Начальная загрузка).

Шаг 2: Вывод списка DFU-устройств (необязательно)

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 проверьте правильность подключения и выбора сценария загрузки МК, после чего повторите попытку.

Шаг 3: Запись образа пользовательской программы

Запишите образ в QSPI Flash с указанием режима записи:

- `alt 7` – запись со смещением (по умолчанию – `0x200000`) – используется в случае ручной сборки образа:

  ```
  dfu-util --alt 7 --download [путь к директории с образом]/main.bin
  ```

- `alt 6` – запись без смещения – используется в случае автоматической сборки образа:

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

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

Максимальный размер образа – 14 МБ (при значении смещения по умолчанию).

Шаг 4: Запись образа XIP-загрузчика (в случае ручной сборки образа)

Запишите образ программы `QSPI_XIPLoader` в QSPI Flash с указанием режима записи `alt 6` – запись без смещения:

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

Для исполнения пользовательской программы установите сценарий загрузки «QSPI» и перезагрузите МК. `QSPI_XIPLoader` будет скопирован в ТСМ и запущен. После запуска он выполнит необходимые настройки и передаст управление пользовательской программе.

7.2.5 Работа с ТСМ через UART


Память ТСМ очищается после снятия питания.

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

Шаг 1: Подготовка

1. Выполните сборку образа для ТСМ.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Подключение к интерфейсу МК «UART0»

Подключитесь 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: Вход в интерфейс командной строки начального загрузчика

1. На плате нажмите кнопку *RESET*.
2. Нажмите и удерживайте клавишу <D> до появления сообщения:

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

3. Удалите лишние символы `d`.

Шаг 4: Запись программы в память МК

1. Нажмите клавишу <L>. 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 интерфейс командной строки начального загрузчика необходимо ввести команду «`:[размер]`» и нажмите <Enter>. Значение вводится в шестнадцатеричном формате.

2. Откройте второе окно терминала и выполните 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 повторите запись, начиная с входа в интерфейс командной строки начального загрузчика (Шаг 3).

Шаг 5: Запуск исполнения программы

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

7.2.6 Работа с eFlash через UART

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

Шаг 1: Подготовка

1. Выполните сборку образа для eFlash.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Подключение к интерфейсу МК «UART0»

Подключитесь 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: Вход в интерфейс командной строки начального загрузчика

1. На плате нажмите кнопку *RESET*.
2. Нажмите и удерживайте клавишу `<D>` до появления сообщения:

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

3. Удалите лишние символы `d`.

Шаг 4: Снятие защиты с eFlash

Снимите защиту с eFlash командой:

```
14000018:1w
```

Шаг 5: Установка параметров

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

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

Шаг 6: Очистка eFlash

Очистите eFlash при помощи команды:

```
:::101m
```

Шаг 7: Включение ожидания загрузки

Дайте команду на ожидание файла образа по протоколу XMODEM:

```
:::3L
```

Шаг 8: Запись программы в память МК

Откройте второе окно терминала и выполните 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 повторите запись, начиная с входа в интерфейс командной строки начального загрузчика (Шаг 3).

Шаг 9: Запуск исполнения программы

Способ 1 - Из интерфейса командной строки начального загрузчика

В интерфейсе укажите адрес eFlash-памяти:

```
a0000000g
```

Способ 2 - Установкой конфигурационных выводов.

1. питание с платы.
2. Установите сценарий загрузки «EFLASH».
3. Подайте питание на плату.

7.2.7 Работа с QSPI Flash / TCM через 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: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Подключение к интерфейсу МК «UART0»

Подключитесь 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: Вход в интерфейс командной строки начального загрузчика

1. На плате нажмите кнопку *RESET*.
2. Нажмите и удерживайте клавишу `<D>` до появления сообщения:

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

3. Удалите лишние символы `d`.

Шаг 4: Запись образа в память МК



По умолчанию производится запись в QSPI1.

Для записи в QSPI0 необходимо ввести команду `20048:13000000::11m`.

Для записи в QSPI1 необходимо ввести команду `20048: 11000000::11m`.

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. Нажмите <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: Запуск исполнения программы

1. Снимите питание с платы.
2. Установите на используемой плате сценарий загрузки «QSPI».
3. Подайте питание на плату.

7.2.8 Работа с QSPI Flash / XIP через UART

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

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

Шаг 1: Подготовка

1. Выполните сборку образа для XIP.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «UART» (см. главу Начальная загрузка).

Шаг 2: Подключение к интерфейсу МК «UART0»

Подключитесь 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: Вход в интерфейс командной строки начального загрузчика

1. На плате нажмите кнопку *RESET*.
2. Нажмите и удерживайте клавишу <D> до появления сообщения:

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

3. Удалите лишние символы d.

Шаг 4: Запись образа пользовательской программы в память МК

1. Дайте команду на прием данных по XMODEM и сохранение их в QSPI Flash



По умолчанию производится запись в QSPI1.

Для записи в QSPI0 введите команду `20048:13000000::11m`.

Для записи в QSPI1 введите команду `20048: 11000000::11m`.

- В случае ручной сборки образа – со смещением (в данном примере – `0x10000`):

```
::10000:0x
```



Смещение должно совпадать со значением параметра `QSPI_APP_OFFSET`, устанавливаемого в `Makefile` проекта `QSPI_XIPLoader`.

Будет выведено сообщение о готовности к принятию файла:

```
@: : 10000: 0 Xmodem saving to 0x0 media 0x0
C
```

- В случае автоматической сборки образа – без смещения:

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

Будет выведено сообщение о готовности к принятию файла:

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

2. Нажмите `<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: Запись образа XIP-загрузчика в память МК (в случае ручной сборки образа)

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

Шаг 6: Запуск исполнения программы

1. Снимите питание с платы.
2. Установите на используемой плате сценарий загрузки «QSPI».
3. Подайте питание на плату.

7.2.9 Работа с TCM через JTAG

В данном разделе представлена информация по записи программ в МК.
Подробное руководство по работе с отладчиком см. в документации на GDB.

Память TCM очищается после снятия питания.

Перед началом работы необходимо загрузить и распаковать SDK.

Шаг 1: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «JTAG» (см. главу Начальная загрузка).

Шаг 2: Открытие директории с OpenOCD

В терминале откройте директорию с OpenOCD:

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

Шаг 3: Запуск OpenOCD

Запустите бинарный файл с указанием файла конфигурации:

```
sudo ./openocd -f ../share/openocd/scripts/board/[путь к файлу конфигурации].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: Открытие второго окна терминала

Откройте терминал в новом окне. Все дальнейшие действия выполняйте в данном окне.

Шаг 5: Открытие директории с GDB

Во втором окне терминала перейдите к директории:

```
cd /[SDK_Path]/Tools/toolchain/riscv32-none-elf/bin/
```

Шаг 6: Запуск отладчик GDB

В новом окне терминала ввести команду:

```
[SDK_path]/Tools/toolchain/riscv32-none-elf/bin/riscv32-none-elf-gdb \
-x [SDK_path]/Tools/scripts/bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 7: Запись образа через отладчик GDB

Запишите бинарный образ в память МК:

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

После успешной записи будут выведены сообщения вида:

```
Restoring binary file /[путь к директории с образом]/[имя образа].bin into memory
0x40010000 to 0xa00002f8)
```

Шаг 8: Запуск исполнения программы

1. Перейдите к адресу ТСМ:

```
bmcu jump 0x40010000
```

2. Дайте команду на исполнение:

```
continue
```



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



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.2.10 Работа с eFlash через JTAG



В данном разделе представлена информация по записи программ в МК.
Подробное руководство по работе с отладчиком см. в документации на GDB.

Перед началом работы необходимо загрузить и распаковать SDK.

Шаг 1: Подготовка

1. Выполните сборку образа для eFlash.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «JTAG» (см. главу Начальная загрузка).

Шаг 2: Открытие директорию с OpenOCD

В терминале откройте директорию с OpenOCD

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

Шаг 3: Запуск OpenOCD

Запустите бинарный файл с указанием файла конфигурации:

```
sudo ./openocd -f ../share/openocd/scripts/board/[путь к файлу конфигурации].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: Открытие второго окна терминала

Откройте терминал в новом окне. Все дальнейшие действия выполняйте в данном окне.

Шаг 5: Открытие директории с GDB

Во втором окне терминала перейдите к директории:

```
cd /[SDK_Path]/Tools/toolchain/riscv32-none-elf/bin/
```

Шаг 6: Запуск отладчик GDB

В новом окне терминала ввести команду:

```
[SDK_path]/Tools/toolchain/riscv32-none-elf/bin/riscv32-none-elf-gdb \
-x [SDK_path]/Tools/scripts/bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 7: Запись образа через отладчик GDB

1. Очистите память eFlash:

```
bmcu eFlash erase
```

2. Перезагрузите МК:

```
bmcu reset
```

3. Запишите бинарный образ в память МК:

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

После успешной записи будут выведены сообщения вида:

```
Restoring binary file /[путь к директории с образом]/[имя образа].bin into
memory (0xa0000000 to 0xa00002f8)
```

Шаг 8: Запуск исполнения программы

1. Перейдите к адресу eFlash:

```
bmcu jump 0xa0000000
```

2. Дайте команду на исполнение:

```
continue
```

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

Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.2.11 Работа с QSPI Flash / TCM через JTAG

В данном разделе представлена информация по записи программ в МК.
Для получения информации по работе с отладчиком см. документацию на GDB.

Ниже описывается способ записи программы в QSPI Flash с ее исполнением из TCM.

Перед началом работы необходимо загрузить и распаковать SDK.

Шаг 1: Подготовка

1. Выполните сборку образа для TCM.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК в сценарии «JTAG» (см. главу Начальная загрузка).

Шаг 2: Открытие директории с OpenOCD

В терминале откройте директорию с OpenOCD:

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

Шаг 3: Запуск OpenOCD

Запустите бинарный файл с указанием файла конфигурации:

```
./openocd.exe -f ../share/openocd/scripts/board/[путь к файлу конфигурации].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: Открытие второго окна терминала

Откройте терминал в новом окне. Все дальнейшие действия выполняйте в данном окне.

Шаг 5: Открытие директории с GDB

Во втором окне терминала перейдите к директории:

```
cd [SDK_Path]/tools/toolchain/riscv32-none-elf/bin
```

Шаг 6: Запуск отладчик GDB

Во втором окне терминала ввести команду:

```
[SDK_path]/Tools/toolchain/riscv32-none-elf/bin/riscv32-none-elf-gdb -x
[SDK_path]/Tools/scripts/bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 7: Запись образа через отладчик GDB

1. Инициализируйте QSPI Flash:

```
bmcu QSPI_clk_init
bmcu QSPI1 init
```

2. Очистите QSPI Flash (если требуется):

```
bmcu QSPI1 erase
```

3. Запишите бинарный образ в память МК:

```
bmcu flash write_bank 1 [путь к образу]/[имя образа].bin 0x90000000
bmcu reset
```

После успешной записи будут выведены сообщения вида:

```
Restoring binary file /[путь к директории с образом]/[имя образа].bin into
memory (0x90000000 to 0x900002f8)
```



Показана запись в QSPI1. Для записи в QSPI0 необходимо использовать адрес `0x90000000`.

Шаг 8: Запуск исполнения программы

1. Снимите питание с платы.
2. Установите сценарий загрузки «QSPI».
3. Подайте питание на плату.



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

Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

7.2.12 Работа с QSPI Flash / XIP через JTAG

В данном разделе представлена информация по записи программ в МК.
Для получения информации по работе с отладчиком см. документацию на GDB.

Ниже описывается способ записи программы в QSPI Flash с ее исполнением непосредственно из QSPI Flash.

Перед началом работы необходимо загрузить и распаковать SDK.

Шаг 1: Подготовка

1. Выполните сборку образов для XIP.
2. Подключитесь к МК в соответствии с документацией на используемую плату.
3. Запустите МК со сценарием загрузки «UART» (см. главу Начальная загрузка).

Шаг 2: Открытие директорию с OpenOCD в терминале

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

Шаг 3: Запуск OpenOCD

Запустите бинарный файл с указанием файла конфигурации для используемой платы:

```
./openocd.exe -f ../share/openocd/scripts/board/[путь к файлу конфигурации].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: Открытие второго окна терминала

Откройте новое окно терминала. Все дальнейшие действия выполняйте в данном окне.

Шаг 5: Открытие директории с GDB

Во втором окне терминала перейдите к директории:

```
cd [SDK_Path]/tools/toolchain/riscv32-none-elf/bin
```

Шаг 6: Запуск отладчика GDB

Во втором окне терминала введите команду:

```
[SDK_path]/Tools/toolchain/riscv32-none-elf/bin/riscv32-none-elf-gdb -x
[SDK_path]/Tools/scripts/bmcu.gdb
```

При успешном подключении будет выведена строка приглашения:

```
(gdb)
```

Шаг 7: Запись образа через отладчик GDB

1. Инициализируйте QSPI Flash:

```
bmcu QSPI_clk_init
bmcu QSPI1 init
```

2. Очистите QSPI Flash (если требуется):

```
bmcu QSPI1 erase
```

3. Запишите бинарный образ в память МК

- В случае ручной сборки образа – со смещением (в данном примере – `0x10000`) от базового адреса QSPI Flash:

```
bmcu flash write_bank 1 [путь к образу]/[имя образа].bin 0x90001000
bmcu reset
```

- В случае автоматической сборки образа – без смещения:

```
bmcu flash write_bank 1 [путь к образу]/[имя образа].bin 0x90000000
bmcu reset
```

После успешной записи будут выведены сообщения вида:

```
Restoring binary file /[путь к директории с образом]/[имя образа].bin into
memory (0x90000000 to 0x900002f8)
```



Показана запись в QSPI1. Для записи в QSPI0 необходимо использовать адрес `0x90000000`.

Шаг 8: Запуск исполнения программы

1. Снимите питание с платы.
2. Установите на используемой плате сценарий загрузки «QSPI».
3. Подайте питание на плату.



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



Для удобства отладки рекомендуется использовать образ, включающий отладочную информацию.

История изменений

Версия	Дата	Описание
1.0.30	12.12.2025	Начальная версия
1.1.3	11.02.2026	Добавлена информация по XIP Незначительные дополнения и исправления
1.2.3	07.04.2026	Добавлена информация по загрузке в QSPI через JTAG Незначительные дополнения и исправления
1.3.0	27.04.2026	Правки в связи с выпуском SDK v2.2.0 Незначительные дополнения и исправления
5	26.05.2026	Таблицы 3-2 и 3-3 объединены в одну, в данную таблицу добавлены временны́е параметры eFlash
6	06.07.2026	Актуализация и исправления
6.1	01.09.2026	Актуализация и исправления
6.2	29.10.2026	Добавлена уточняющая информация по переходу в режим «USB MP» из интерфейса командной строки начального загрузчика Незначительные исправления

Контактные данные

Официальный сайт
www.baikalelectronics.ru

Главный офис
www.baikalelectronics.ru/contacts

Электронная почта
info@baikalelectronics.ru

Телефон
+7 495 221-39-47