



Описание загрузочной программы для K1986BK025



Оглавление

1	Лист регистрации изменений загрузочной программы	3
2	Введение	4
3	UART-загрузчик	6
3.1	Параметры связи по UART	6
3.2	Протокол обмена по UART	6
3.3	Синхронизация с внешним устройством	6
3.4	Команды UART загрузчика	7
3.4.1	Команда CMD_SYNC	7
3.4.2	Команда CMD_CR	7
3.4.3	Команда CMD_BAUD	8
3.4.4	Команда CMD_LOAD	8
3.4.5	Команда CMD_VFY	9
3.4.6	Команда CMD_RUN	9
3.4.7	Прием параметров команды	10
3.4.8	Сообщения об ошибках	10
4	Лист регистрации изменений	11

[illegible]



2 Введение

После включения питания и снятия внутренних (POR) и внешних (nRESET) сигналов сброса микросхема начинает выполнять программу из загрузочной области BOOT OTP. В загрузочной программе микроконтроллер определяет, в каком из режимов он будет функционировать, и переходит в этот режим. Перед определением режима функционирования загрузочная программа анализирует бит `fprog` в регистре `BKP_LDO`:

- `fprog` = 0: идёт выполнение программы после сброса по питанию, подано основное (U_{CC} , U_{CCA}) и батарейное питание (U_{CCB}). Режим функционирования определяется внешними выводами `MODE[1:0]` (`PC[0]`, `PC[1]`), при этом перед опросом состояния этих выводов для них включается внутренняя подтяжка к шине «Общий» (встроенные резисторы подтяжки к шине «Общий» имеют сопротивление ~ 85 кОм). После определения режима функционирования состояние выводов `MODE[1:0]` записывается в регистр `BKP_LDO`, поле `mode[1:0]`, а бит `fprog` устанавливается в 1. Данное состояние бита `fprog` сохраняется до отключения основного и батарейного питания, при условии, что бит `fprog` не будет изменен в пользовательской программе.
- `fprog` = 1: идёт выполнение программы после системного сброса, батарейное питание (U_{CCB}) не отключалось. После перезапуска микроконтроллера уровни на выводах `MODE[1:0]` не влияют на режим функционирования микроконтроллера, так как в этом случае режим работы будет прочитан из бит `mode[1:0]` регистра `BKP_LDO`.

В пользовательской программе выводам `PC[0]`, `PC[1]` пользователем могут присваиваться функции самостоятельно.

Таблица 1 – Режимы работы микроконтроллера

MODE[1:0]	Режим	Стартовый адрес/ таблица векторов прерываний	Описание
00	Режим отладки	0x10000000	Процессор начинает выполнять программу из внутренней OTP, а затем передаёт управление во FLASH-память программ. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в OTP-памяти
01 или 10	UART загрузчик	Определяется пользователем	Микроконтроллер через интерфейс UART1 на выводах <code>PB[1]</code> , <code>PB[0]</code> получает код программы в ОЗУ для исполнения. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в OTP-памяти
11	Режим отладки	0x80000000	Процессор начинает выполнять программу из внутренней OTP, а затем передаёт управление в ОЗУ память программ. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в OTP-памяти

При работе в режиме отладки разрешается работа отладочного интерфейса JTAG. При этом к микроконтроллеру может быть подключен JTAG-адаптер, с помощью которого программные средства разработки позволяют работать с микроконтроллером в отладочном режиме.



В отладочном режиме можно:

- стирать, записывать, считывать внутреннюю FLASH-память программ;
- считывать и записывать содержимое ОЗУ, периферии;
- выполнять программу в пошаговом режиме;
- запускать программу в нормальном режиме;
- останавливать программу по точкам остановки;
- просматривать переменные выполняемой программы;
- проводить трассировку хода выполнения программного обеспечения.

Ячейка памяти OTP размером 1 байт по адресу 0x0002_3FDF/0x7000_3FDF используется для хранения номера версии загрузочной программы.



3 UART-загрузчик

Режим UART-загрузчика предоставляет достаточный набор операций, необходимых для записи в ОЗУ какой-либо программы (в частности программатора Flash-памяти), верификации ее и запуска на выполнение. Кроме того, существует возможность задания внешним устройством скорости обмена. Помимо доступа к ОЗУ может быть осуществлен доступ и к другим адресным диапазонам (FLASH, OTP, Периферия).

В качестве источника тактовой частоты UART1 используется внутренний RC-генератор HSI с частотой 8 МГц. Так как имеется разброс значений частоты HSI, то требуется этап подбора значения делителя частоты UART1 для синхронизации с внешним устройством.

При загрузке программы в ОЗУ необходимо учитывать, что часть ОЗУ используется загрузочной программой для хранения следующих объектов:

- 0x8000_0000-0x8000_00FF - таблица векторов прерываний;
- 0x8000_FE00-0x8000_FFFF - стек;
- 0x8001_0000-0x8001_016D - код обработчиков прерываний.

3.1 Параметры связи по UART

Для связи по UART выбраны следующие параметры канала связи:

- начальная скорость – 9600 бод;
- количество бит данных – 8;
- четность – нет;
- количество Stop бит – 1;
- загрузчик не использует FIFO UART1;
- загрузчик всегда выступает в качестве Slave, а внешнее устройство, подающее команды – в качестве Master;
- данные передаются младшим битом вперед.

3.2 Протокол обмена по UART

После синхронизации с Master загрузчик переходит в диспетчер команд. Таким образом, Master-у доступны команды, приведенные в таблице 2.

3.3 Синхронизация с внешним устройством

Начальные условия.

На этапе синхронизации с внешним устройством (Master) вывод Rx используется как вход. Master постоянно посылает в канал синхросимвол – 0. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ошибки обмена. Как только Загрузчик настроил скорость, он переходит в диспетчер команд и выдает Master-у приглашение (3 байта – 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>')).

Master завершает выдачу синхросимволов и теперь может подавать команды, согласно протоколу обмена.



3.4 Команды UART загрузчика

Таблица 2 – Команды UART загрузчика

Команда	Код	ASCII Символ	Описание
CMD_SYNC	0x00		Пустая команда. Загрузчик ее принимает, но ничего по ней не делает
CMD_CR	0x0D		Выдача приглашения Master-y
CMD_BAUD	0x42	'B'	Установка скорости обмена
CMD_LOAD	0x4C	'L'	Загрузка массива байт
CMD_VFY	0x59	'Y'	Выдача массива байт
CMD_RUN	0x52	'R'	Запуск программы на выполнение

3.4.1 Команда CMD_SYNC

Пустая команда.

Загрузчик (Slave) ее принимает, но ничего не делает. Код команды соответствует символу синхронизации.

Таблица 3 – Команда CMD_SYNC

Код команды	CMD_SYNC = 0x00
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master: Выдает код команды CMD_SYNC	Slave: если команда принята с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды

3.4.2 Команда CMD_CR

Выдача приглашения Master-y.

Таблица 4 – Команда CMD_CR

Код команды	CMD_CR = 0x0D
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master: Выдает код команды CMD_CR	Slave: если команда принята с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды. Иначе выдаёт 3 байта: - код команды CMD_CR; - код 0x0A; - код 0x3E (ASCII символ '>').



3.4.3 Команда CMD_BAUD

Установка скорости обмена.

Таблица 5 – Команда CMD_BAUD

Код команды	CMD_BAUD = 0x42
ASCII символ, соответствующий коду команды	'B'
Количество параметров команды	1
Параметр	Новое значение скорости обмена [бод]
Формат команды:	
Master: Выдает код команды CMD_BAUD	Slave: если команда принята с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр	Slave: Если параметр принят с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_BAUD и завершает обработку текущей команды. Иначе: - выдает код команды CMD_BAUD; - после передачи кода команды CMD_BAUD устанавливает новое значение скорости обмена.

3.4.4 Команда CMD_LOAD

Загрузка массива байт в память микроконтроллера.

Таблица 6 – Команда CMD_LOAD

Код команды	CMD_LOAD = 0x4C
ASCII символ, соответствующий коду команды	'L'
Количество параметров команды	2
Параметр 1	Адрес памяти приемника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master: Выдает код команды CMD_LOAD	Slave: если команда принята с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master: Выдает параметр 1	Slave: ожидает получения всех параметров.
Master: Выдает параметр 2	Slave: если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке с типом ERR_CHN и завершает обработку текущей команды. Иначе выдает код команды CMD_LOAD.
Master: Выдает массив байт младшим байтом вперед	Slave: принимает массив байт. Если хотя бы один байт принят с ошибками, то выдает сообщение об ошибке с типом ERR_CHN и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончании принятия массива выдает код ответа REPLY_OK = 0x4B ('K').



3.4.5 Команда CMD_VFY

Выдача массива байт из памяти микроконтроллера.

Таблица 7 – Команда CMD_VFY

Код команды	CMD_VFY = 0x59
ASCII символ, соответствующий коду команды	'Y'
Количество параметров команды	2
Параметр 1	Адрес памяти источника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master: Выдает код команды CMD_VFY	Slave: если команда принята с ошибками, выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master: Выдает параметр 1	Slave: ожидает получения всех параметров.
Master: Выдает параметр 2	Slave: если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке с типом ERR_CHN и завершает обработку текущей команды. Иначе: <ul style="list-style-type: none">- выдает код команды CMD_VFY;- выдает массив байт младшим байтом вперед;- по окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K').

3.4.6 Команда CMD_RUN

Запуск программы на выполнение.

Таблица 8 – Команда CMD_RUN

Код команды	CMD_RUN = 0x52
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр	Адрес первой команды загруженной программы
Формат команды:	
Master: Выдает код команды CMD_RUN	Slave: если команда принята с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр	Slave: если параметр принят с ошибками, то выдает сообщение об ошибке с типом ERR_CHN и завершает обработку текущей команды. Иначе: <ul style="list-style-type: none">- выдает код команды CMD_RUN;- устанавливает значение PC согласно принятому адресу (SP и PLIC не перепрограммируются) и, таким образом, Slave завершает свое выполнение. Передача управления загруженной программе происходит после окончания отправки кода команды CMD_RUN.



3.4.7 Прием параметров команды

Параметры команд – это 4-байтные числа.

Параметры передаются младшим байтом вперед.

В качестве значения параметра запрещено использовать число 0xFFFFFFFF.

Если при приеме параметра обнаружена аппаратная ошибка (UART установил в '1' какой-либо из флагов ошибки), то прием параметров не прекращается.

Анализ всех видов ошибок, связанных с передачей параметров, загрузчик производит только после принятия всех параметров команды.

3.4.8 Сообщения об ошибках

Сообщения об ошибках – это 2-х байтные последовательности символов. Первый символ всегда 0x45 ('E'). Второй символ определяет тип ошибки.

После выдачи сообщения об ошибке загрузчик переходит в режим ожидания следующей команды, поэтому Master после получения такого сообщения должен прекратить передачу байт, относящихся к текущей команде.

После принятия сообщения об ошибке Master должен подавать команду CMD_CR до тех пор, пока не получит корректный ответ, соответствующий этой команде.

Возможны следующие сообщения об ошибках: ERR_CHN, ERR_CMD, ERR_BAUD.

Ошибка ERR_CHN

Аппаратная ошибка UART.

Код ошибки 0x69 ('i').

Выдается, если UART установил в '1' один из аппаратных флагов ошибки при приеме очередного байта.

Ошибка ERR_CMD

Принята неизвестная команда.

Код ошибки 0x63 ('c').

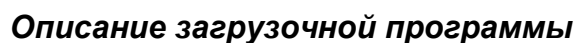
Выдается диспетчером команд, если принят неизвестный код команды.

Ошибка ERR_BAUD

Принята неизвестная команда.

Код ошибки 0x62 ('b').

Выдается диспетчером команд, если по принятому от Master-а значению скорости обмена невозможно вычислить корректное значение делителя частоты UART.

[illegible]