

ПРОТОКОЛ ПОСЛЕДОВАТЕЛЬНОЙ ЗАГРУЗКИ ПРЕЦИЗИОННОГО МИКРОКОНВЕРТОРА ADUC702X

Одна из многочисленных полезных особенностей семейства микроконверторов – это возможность внутрисхемной загрузки кода во внутреннюю флэш/ЭСРЗУ память (Flash/EEPROM) программ. Данные загружаются через универсальный последовательный асинхронный порт (UART) микросхемы, что позволяет разработчику программировать размещенный на плате микроконвертор без применения специального программатора. Чтобы обновить программу, достаточно подключиться к последовательному порту микроконвертора. А это означает, что программу можно обновлять оперативно "в полевых условиях" без демонтажа процессора. Рассмотрим протокол последовательной загрузки микроконвертора с тем, чтобы детально разобраться в нем и при необходимости реализовать его в своем устройстве (хост-процессор* + микроконвертор).

ПРОТОКОЛ ПОСЛЕДОВАТЕЛЬНОЙ ЗАГРУЗКИ

При определенной конфигурации выводов все микроконверторы переводятся в режим последовательной загрузки после сброса процессора от внешнего сигнала или после подачи питания. Если условие необходимого конфигурирования выводов выполнено, при включении питания или при аппаратном сбросе процессора микроконвертор переходит в режим последовательной загрузки. Запускается встроенная программа загрузки, которая конфигурирует порт UART процессора и с помощью специального протокола связывается с хост-машиной для приёма данных и последующей загрузки их в флэш/ЭСРЗУ память.

Необходимо иметь в виду, что режим последовательной загрузки подразумевает стандартное напряжение питания микроконвертора – 2,7–3,6 В. Так что для программирования не нужен специальный источник высокого напряжения – напряжение генерируется встроенной схемой на кристалле.

На рис. 1 показан перевод микроконвертора в режим последовательной загрузки на оценочной плате, входящей в набор QuickStart™ Development Tools компании. В набор также входит программа ARMWSD.exe под Windows (её можно скачать с сайта <ftp://ftp.analog.com/pub/MicroConverter/WSD/armwsd>), которая поз-



Од Ричард
Перевод А.Власенко
alexey.vlasenko@analog.spb.ru

воляет загружать в микроконвертор код из ПК через последовательный порт COM1, 2, 3 или 4. Однако следует подчеркнуть, что любая хост-машина (ПК, микроконтроллер или сигнальный процессор) сможет загружать код в микроконвертор в том случае, если она поддерживает рассматриваемый протокол последовательной загрузки.

Запуск программы-загрузчика в микроконверторах семейства ADuC702x компании Analog Devices. Для прогона программы-загрузчика вывод P0.0 должен быть "притянут" к земле (как правило, с помощью 1-кОм резистора) и выполнен сброс прибора (с помощью вывода RST или путем подачи внешнего импульса).

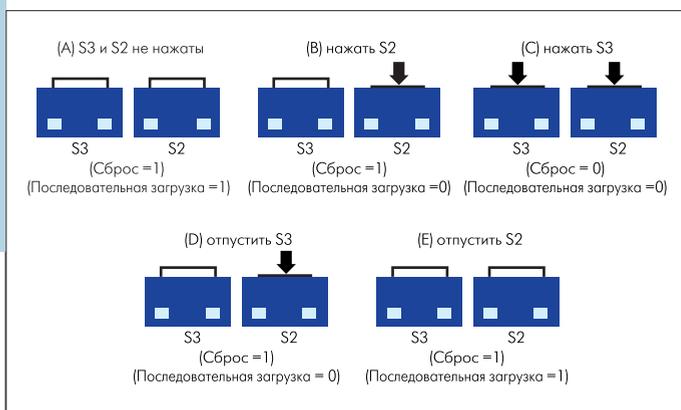


Рис. 1. Перевод микроконвертора в режим последовательной загрузки

Физический интерфейс. После запуска программа-загрузчик ожидает получения от хоста кода возврата (backspace – BS = 0x08) для установления синхронизации, после чего загрузчик конфигурирует UART-интерфейс на передачу/прием восьмиразрядных данных без бита четности в соответствии со скоростью передачи данных хоста. Скорость должна быть в пределах 600–19200 бит/с. После получения BS-кода загрузчик сразу посылает со скоростью 9600 бит/с 24-байтный пакет-идентификатор (ID) со следующей структурой (рис.2): 15 байт – идентификатор микросхемы, 3 бай-

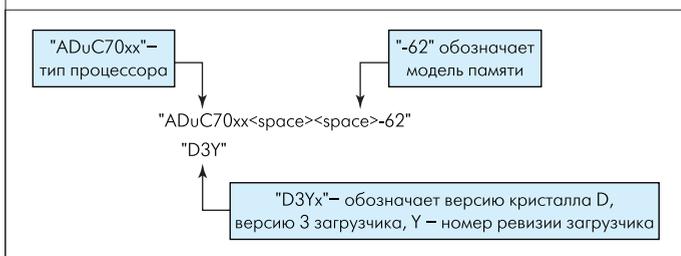


Рис. 2. Пакет-идентификатор ID

*Условимся, что под понятием "хост-процессор" подразумевается процессор (ПК, микроконтроллер или сигнальный процессор – DSP), который загружает данные в микроконвертор, а под понятием "загрузчик" – встроенная в микроконвертор заводская программа, реализующая последовательное программирование встроенной флэш/ЭСРЗУ памяти.



та – номер версии программно-аппаратного и аппаратного обеспечения, 4 байта – зарезервированные для использования в дальнейшем и 2 байта – коды перевода строки и обратного хода.

Задание формата пакета передаваемых данных. После того как интерфейс UART сконфигурирован, начинается передача данных. Пакет передаваемых данных имеет следующий формат:

Поле пакета	Байты
Стартовый ID0x07 и 0x0E
"Число байт данных"	..5–255
Команд (Data 1)E, W, V или R
Адреса (Data 2...5)h, u, m, l
Данных (Data 6...255)	...x
Контрольная сумма	...число байт данных + Data 1 + Data 2...5 + s Data x

(дополнительный код)

Байты стартового ID постоянны и используются загрузчиком с целью подтверждения достоверного пакета данных. Следующее поле пакета показывает общее число байт данных, включая поле Data 1 (команда). Минимальное число байт данных – 5, что соответствует полю команды и адресу. Максимально допустимое число – 255: команда, 4 байт – адрес и 250 байт – данные. Поле команды (Data 1) описывает назначение пакета данных. Допустима одна из четырех возможных команд в виде кода ASCII: E, W, V или R. Поле адреса (Data 2...5) содержит 32-разрядные адреса h, u, m и l. При этом формат данных – "младший в начале", т.е. младший значащий байт расположен в памяти по меньшему адресу (h), а старший значащий байт – по старшему адресу (l). Поле данных (Data 6–255) содержит до 250 байт пользовательского кода, который загружается/верифицируется побайтно. Данные извлекаются из 16-байт формата Intel Extended Hex, после чего хост-процессор ретранслирует их и включает в пакет для последующей загрузки. В поле контрольной суммы (КС) находится контрольная сумма пакета данных. Это сумма в дополнительном коде формируется за счет сложения шестнадцатеричных величин, содержащихся в поле "число байт данных" и в полях данных Data 1–Data 255 (если они имеются), т.е. КС = 0x00 – (число байт данных + ? – Data 1–Data 255).

Если по результату проверки контрольной суммы принятый пакет данных оказывается некорректным или загрузка производится по некорректному адресу, программа-загрузчик посылает отрицательный ответ – код BEL (0x07). Положительный ответ загрузчика на передачу пакета данных – код ACK (0x06). Если данные записываются в ячейки памяти, уже содержащие какие-либо не стертые данные, загрузчик не выдает предупреждения. Отработку подобной ситуации обеспечивает программа хоста. Полный набор функций пакета приведен в табл. 1.

Таблица 1. Команды пакета передаваемых данных

Команда	Код команды (поле Data 1)	Подтверждение загрузчика	Отрицательный ответ загрузчика
Стереть страницу	E (0x45)	ACK (0x06)	BEL (0x07)
Запись	W (0x57)	ACK (0x06)	BEL (0x07)
Верификация	V (0x56)	ACK (0x06)	BEL (0x07)
Запуск программы (переход на начало пользовательского кода)	R (0x52)	ACK (0x06)	BEL (0x07)

Команда стирания позволяет пользователю очищать память от одной и более (вплоть до стирания всех данных флэш/ЭСРПЗУ-памяти) страниц, начиная с адреса, указанного в поле Data 2...5. Пакет команды определяет и число стираемых страниц. Если адрес 0x00000000, а число страниц 0x00, загрузчик интерпретирует эту команду как команду стирания всего пользовательского кода. Пакет команды стирания имеет следующий формат:

Поле пакета	Байты
Стартовый ID0x07 и 0x0E
Число байт данных6
Команд (Data 1)E (0x45)
Адреса (Data 2...5)h, u, m, l
Данных (Data 6)x страниц
Контрольная суммачисло байт данных + ? Data x (дополнительный код)

Команда записи требует указания числа байт данных (Data 1 + Data 2 + Data 2...5 + Data x), самой команды, адреса первого байта загрузки и собственно данных, подлежащих записи. Пакет команды записи имеет следующий формат:

Поле команды	Байты
Стартовый ID0x07 и 0x0E
Число байт данных5 + число байт Data x (5...255)
Команд (Data 1)W (0x47)
Адреса (Data 2...5)h, u, m, l
Данных (Data 1...255)дополнительные байты данных
Контрольная суммачисло байт данных + s Data x – 515

Байты записываются в память в той последовательности, в которой они передаются. Если контрольная сумма не совпадает или переданное значение адреса лежит вне допустимого диапазона, загрузчик посылает код BEL. При приеме этого сигнала хостом процесс загрузки прерывается и она начинается сначала.

Таблица 2. Модификация байт при выполнении команды верификации

Исходный порядок бит	Передаваемые биты	Восстановленный порядок бит
7	4	7
6	3	6
5	2	5
4	1	4
3	0	3
2	7	2
1	6	1
0	5	0

Команда верификации флэш/ЭСРПЗУ-памяти почти идентична команде записи (табл. 2).

В поле команды передается код V (0x56), но для повышения вероятности обнаружения ошибки байты передаваемых данных модифицированы: младшие 5 бит перемещаются в старшую часть байта, а

старшие 3 бита – в младшую часть. Программа-загрузчик восстанавливает правильную последовательность бит и сравнивает ее с содержимым флэш-памяти. Если последовательность бит и контрольная сумма совпадают, то возвращается код ACK (0x06); в противном случае возвращается код BEL (0x07). Команда верификации данных флэш/ЭСРПЗУ-памяти имеет следующий формат:

Поле команды	Байты
Стартовый ID0x07 и 0x0E
Число байт данных5 + число байт (Data x) 5...255
Команд (Data 1)V (0x56)
Адреса (Data 2...5)h, u, m, l
Страниц (Data 6)дополнительные байты данных
Контрольная суммачисло байт данных + s Data x – 515

Команда запуска программы (удаленного запуска). После того как хост передал все данные загрузчику, он может послать команду перехода счетчика команд микроконвертора на заданный адрес и таким образом начать выполнение загруженного кода. Команда записи в память флэш/ЭСРПЗУ и удаленного запуска программы с адреса 0x00 имеет следующий формат:

Поле команды	Байты
Стартовый ID0x07 и 0x0E
Число байт данных0x05
Команд (Data 1)R (0x52)
Адреса (Data 2...5)h, u, m, l
Данных (Data 1...250)дополнительные байты данных
Контрольная сумма0xA9

Сейчас поддерживается только запуск с начала памяти Flash/EEPROM (h, u, m, l = 0x80000).