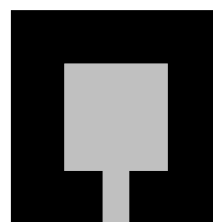


Семейство процессоров ADSP-2100

Справочное руководство по EZ-KIT Lite



Первое издание [оригинал] (1995)
Предварительное издание [перевод] (2001)

АВТОРЫ ПЕРЕВОДА :
ЗАКАЗ И ОРГАНИЗАЦИЯ ПЕРЕВОДА

ПОКРОВСКИЙ В.Н., СИЛАНТЬЕВ В.И.
ЦУКАНОВ Ю.В.

©2001, ЗАО АВТЭКС Санкт-Петербург

Литература

Руководства пользователя ADSP-2100

ADSP-2100 Family User's Manual (Prentice Hall)
ADSP-2171/81 User's Manual
ADSP-2100 Family Assembler Tools & Simulator Manual
ADSP-2100 Family C Tools Manual
ADSP-2100 Family C Runtime Library Manual

Руководства по применению (Prentice Hall)

Digital Signal Processing Applications Using the ADSP-2100 Family, Volume 1

Включены темы арифметики, фильтров, БПФ, предикативного линейного кодирования, модемов, алгоритмов, графики, кодово-импульсной модуляции, многоступенчатых фильтров, двухтонального многочастотного набора, многопроцессорной обработки, хост-интерфейса и гидролокации.

Digital Signal Processing Applications Using the ADSP-2100 Family, Volume 2

Включены темы модемов, предикативного линейного кодирования, GSM кодеков, АДИКМ кодирования, распознавания речи, дискретного косинусного преобразования, цифрового тонального детектирования, разработки цифровых систем управления, биквадратных БИХ-фильтров, организации программного и аппаратного UART интерфейса.

Информация по спецификациям

ADSP-21xx Data Sheet
ADSP-2181 Data Sheet
AD1847 Data Sheet

© 1995 Analog Devices, Inc.
© 2001 AUTEX Ltd.

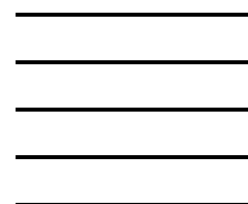
Все права защищены.

ЗАМЕЧАНИЯ ПО ИЗДЕЛИЯМ И ДОКУМЕНТАЦИИ ANALOG DEVICES

Analog Devices оставляет за собой право изменить данное изделие и его документацию без заблаговременного уведомления. Информация предоставляемая Analog Devices проверена и может считаться достоверной. Однако, Analog Devices не несет ответственности за использование изделия, также как за нарушение прав или других прав третьих лиц, которые могут быть затронуты в результате использования изделия. Не предоставляются никакие лицензии при включении или использовании патентных прав Analog Devices.

ПЕРВОЕ ОРИГИНАЛЬНОЕ ИЗДАНИЕ, 1995
ПЕРВОЕ ПЕРЕВЕДЕННОЕ ИЗДАНИЕ, 2001

Введение

**1**

Введение

EZ-KIT Lite является одной из самых эффективных по стоимости, мощной системой разработки доступной на рынке. В EZ-KIT Lite используется ADSP-2181 – 16-разрядный DSP процессор высокой интеграции с фиксированной точкой и встроенной памятью 32К слов ОЗУ, временем выполнения командного цикла 30 нс, портами прямого доступа к памяти и режимами пониженного энергопотребления.

Комплект поставки

EZ-KIT Lite поставляется со следующей комплектацией:

- ◆ Макетная плата ADSP-2181
- ◆ 3.5" дискета с программным обеспечением
- ◆ Руководство по EZ-KIT Lite
- ◆ Кабель RS-232 с двумя разъемами DB9
- ◆ Источник питания

EZ-KIT Lite

EZ-KIT Lite предоставляет простой путь исследования возможностей процессоров семейства ADSP-2100 и разработки собственных приложений для этих высокопроизводительных DSP. EZ-KIT Lite является законченной системой разработки, идеальной для начала работы с DSP, которая позволит пройти через все фазы процесса разработки. С EZ-KIT Lite можно:

- ◆ Оценить DSP фирмы Analog Devices
- ◆ Изучить применение DSP
- ◆ Разрабатывать DSP приложения
- ◆ Моделировать и проводить отладку приложений
- ◆ Создавать опытные образцы

EZ-KIT Lite состоит из небольшой базовой макетной/демонстрационной платы ADSP-2181 с полностью реализованной возможностью 16-разрядного стерео звукового ввода/вывода. В данном руководстве эта плата может быть обозначена EZ-LAB® или EZ-KIT Lite. Используемые термины могут меняться по тексту. Возможности платы:

- ◆ ADSP-2181 33 MIPS DSP
- ◆ AD1847 стерео звуковой порт
- ◆ Интерфейс RS-232
- ◆ Установленное EPROM (СППЗУ)
- ◆ Кнопки пользователя
- ◆ Регулирование энергопотребления
- ◆ Коннекторы расширений
- ◆ Настраиваемые пользователем соединительные перемычки

Плата может работать автономно или подключенная к ПК через порт RS-232. Программа монитор, запущенная на DSP, совместно с управляющей программой, запущенной на ПК, позволяет в интерактивном режиме загружать программы и опрашивать внутренние состояния ADSP-2181. Плата поставляется с установленной EPROM; поэтому можно как запускать программу монитор и демонстрационные программы, так и подключить EPROM с собственной программой.

EZ-KIT Lite поставляется с полным набором программного обеспечения, которое требуется для разработки собственных DSP приложений. Этот набор средств разработки включает: ассемблер (*assembler*), редактор связей или компоновщик (*linker*), разделитель программ для записи в ППЗУ (*PROM splitter*) и программу моделирования (*simulator*).

Содержание этого руководства

Это руководство содержит всю необходимую информацию, которая потребуется для:

- ◆ Установки программного обеспечения на IBM совместимый ПК
- ◆ Подключения EZ-KIT Lite к ПК
- ◆ Подключения EZ-KIT Lite к источнику питания
- ◆ Подключения входного источника (напр., микрофона или CD проигрывателя) к EZ-KIT Lite
- ◆ Подключение выходного устройства (напр., усилителя или динамика)
- ◆ Запуск платы EZ-KIT Lite
- ◆ Использование программного обеспечения
- ◆ Создания собственных ADSP-2181 программ для запуска на плате EZ-KIT Lite

Глава 2 описывает основные возможности программного обеспечения разработчика EZ-KIT Lite и рассказывает о расширениях, доступных при обновлении программного обеспечения.

Глава 3 содержит основную информацию о начале работы с платой. Рекомендуется прочитать эти инструкции полностью, особенно, если вы новичок в программировании DSP.

Глава 4 дает детальное описание для установки различных модулей программного обеспечения на IBM совместимые ПК. Это следующие программные модули:

- ♦ Программное обеспечение разработчика ADSP-2181 (Development Software)
- ♦ Базовое программное обеспечение EZ-KIT Lite (Host Software)
- ♦ Демонстрационные программы
- ♦ Примеры программ

Глава 5 содержит обзор процесса разработки. Она предоставляет большую часть информации, требующейся для создания собственных программ для ADSP-2181. Приведен и описан законченный пример программы. Можно использовать эту программу как основу для написания собственной программы. Кроме того, приведены примеры командных строк запуска программ ассемблера, редактора связей и разделителя программ для записи в ППЗУ.

Глава 6 описывает действие хост программы. Хост программа запускается в среде Windows™ на IBM совместимых ПК и используется для интерфейса с ADSP-2181 EZ-LAB.

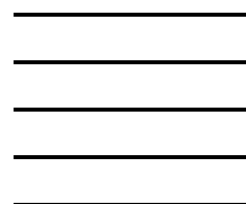
Глава 7 описывает программу монитор, которая загружена на плату EZ-KIT Lite. Приведены также руководящие инструкции по созданию собственных DSP программ для использования с программой монитором.

Глава 8 содержит детальное описание аппаратной части EZ-KIT Lite. Приведена информация, относящаяся к расширению аппаратной части EZ-KIT Lite.

Следующие главы не включены в перевод и могут быть найдены в оригинале руководства:

- ♦ **Краткий справочник по программированию**
- ♦ **Схемы**

Информация об обновлении

**2**

Обзор

EZ-KIT Lite поставляется с полностью функциональной версией программного обеспечения для основных операций разработки DSP приложений, таких как ассемблирование, компоновка, моделирование, отладка и разделение для программаторов ПЗУ. Это программное обеспечение позволяет создавать высокопроизводительные системы любой сложности, базирующиеся на ADSP-2181.

Предоставленные возможности удовлетворяют требованиям большинства пользователей, но тем, кто хочет воспользоваться гибкостью С компилятора, С отладчика, набором библиотек, управляющей программой для работы с библиотечными программами и программой моделирования, предоставляется возможность улучшения стандартного программного обеспечения для семейства ADSP-2100.

Описание дополнений

В дополнение к возможностям Lite версии, программное обеспечение разработки для семейства ADSP-2100 может содержать:

- ◆ **Системный конфигурактор**

Определяет конечную аппаратную часть системы в файле описания архитектуры. Редактор связей и программа моделирования используют эту информацию, чтобы определить размер памяти системы, тип памяти – ОЗУ или ПЗУ, какая часть памяти является внутренней и внешней, и какая периферия отображена в памяти.

- ◆ **Программы моделирования**

Предоставляют моделирование на уровне инструкций процессоров семейства ADSP-2100. Все программы моделирования являются интерактивными, показывая операции от такта к такту, процессор и аппаратную часть системы с помощью интерфейса, базирующемся на графическом оконном представлении.

- ◆ **Программа составления библиотек**

Объединяет часто используемые подпрограммы в одном библиотечном файле, облегчая процесс компоновки и рационализируя использование программной части системы.

- ◆ **С компилятор**

Позволяет составлять программы на языке С по стандарту ANSI.

♦ **Библиотека исполняемых программ С**

Использует вызываемую библиотеку стандартных ANSI и специфичных DSP функций. Библиотека исполняемых программ С включает в себя функции, позволяющие упростить обработку прерываний, автоматически сохраняя и восстанавливая необходимые регистры.

♦ **С отладчик**

Облегчает процесс отладки, действуя на уровне исходного кода на языке С. Интегрируется в среду окружения эмулятора и программы моделирования.

Ниже приведена таблица, сравнивающая программное обеспечение, поставляемое с EZ-KIT Lite и возможности программного обеспечения для процессоров семейства ADSP-2100.

Сравнение программного обеспечения

| Возможности | EZ-KIT Lite | ADSP-2100 |
|--|--------------------|------------------|
| Системный конфигурактор | * | ✓ |
| Ассемблер | ✓ | ✓ |
| Редактор связей | ✓ | ✓ |
| Программа моделирования для ADSP-2101 | | ✓ |
| Программа моделирования для ADSP-2111 | | ✓ |
| Программа моделирования для ADSP-2171 | | ✓ |
| Программа моделирования для ADSP-2181 | ✓ | ✓ |
| Разделитель программ для записи в ППЗУ | ✓ | ✓ |
| С компилятор | | ✓ |
| Библиотека исполняемых программ С | | ✓ |
| Программа составления библиотек | | ✓ |
| С отладчик | | ✓ |
| Документация | | ✓ |
| Техническая поддержка | | ✓ |

* EZ-KIT Lite поставляется с файлом описания архитектуры для ADSP-2181

Начало работы

**3**

Обзор

EZ-KIT Lite включает в себя макетную плату ADSP-2181 EZ-LAB, программное обеспечение разработки ADSP-2181, базовое программное обеспечение (*host software*), примеры программ и документацию. Может быть получена или куплена следующая литература:

- ◆ ADSP-2100 Family User's Manual
- ◆ ADSP-2100 Family Assembler Tools & Simulator Manual
- ◆ ADSP-2181 User's Manual

Чтобы полностью использовать все функции EZ-KIT Lite, необходимо установить сопровождающее программное обеспечение и подключить плату. Подключите плату к источнику питания, выберите источник аналогового входного сигнала и присоедините громкоговоритель.

Установка программного обеспечения

Установка программного обеспечения осуществляется в операционной системе Windows™:

1. Проверьте, что операционная система Windows™ запущена.
2. Вставьте в дисковод дискету с программным обеспечением EZ-KIT Lite
3. Перейдите к окну менеджера программ (или проводника)
4. Выберите из меню *File* (или меню *Пуск*) пункт *Run (Выполнить...)*.
5. В командной строке наберите `A:\SETUP.EXE` (или `B:\SETUP.EXE`)
6. Нажмите кнопку `OK` или `ENTER`
7. Следуйте инструкциям на экране

Примечание: необходимо внести следующие изменения в файл `AUTOEXEC.BAT`:

1. Заведите переменную среды окружения `SET ADI_DSP=C:\ADI_DSP`
2. В переменной `PATH` среды окружения допишите `C:\ADI_DSP\21XX\BIN`

Необходимые условия

Чтобы запустить программное обеспечение EZ-KIT Lite требуется:

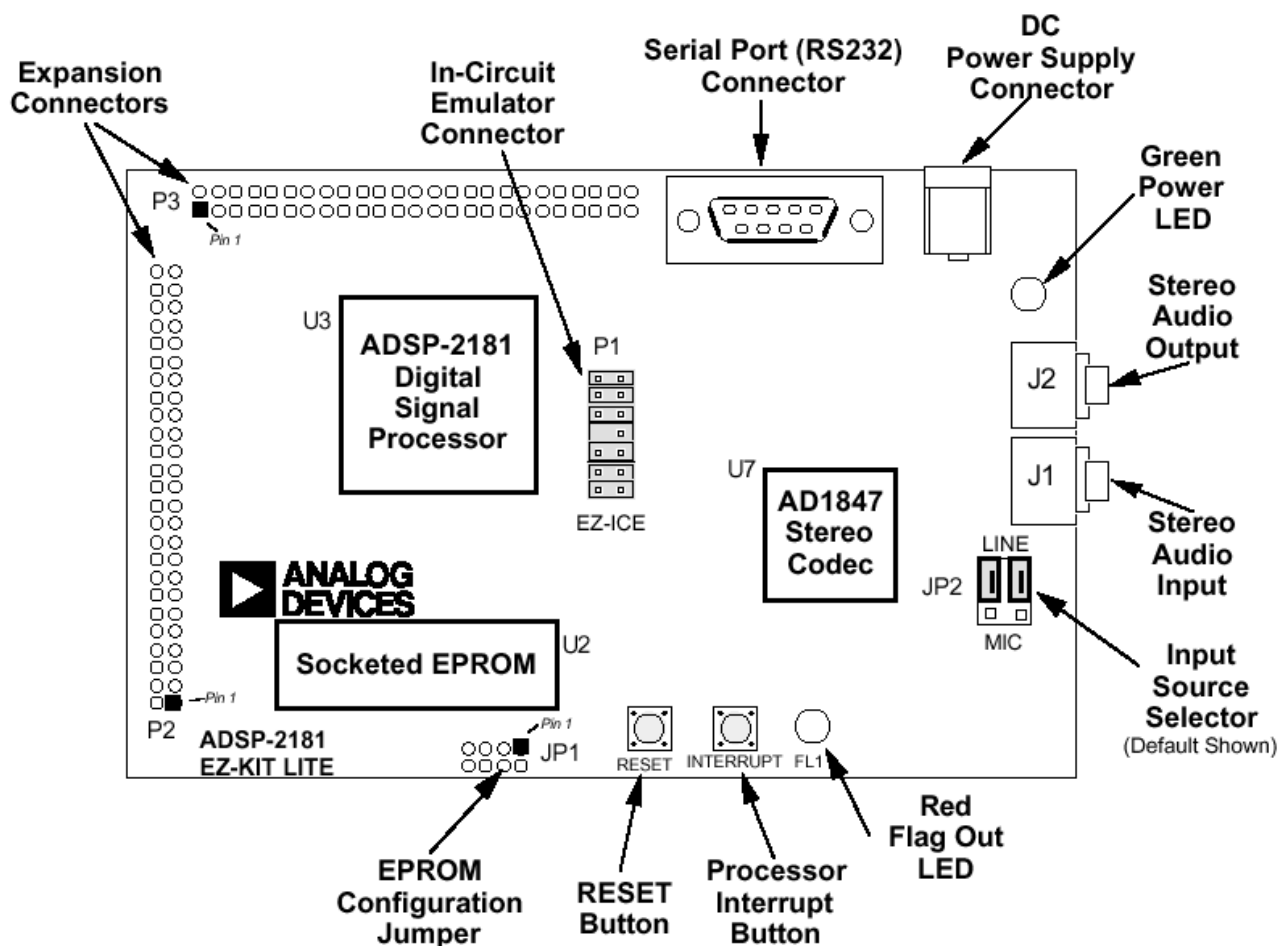
- ◆ ПК с процессором 386 (или выше), жестким диском, 3.5" дюймовый дисковод, цветная видеокарта с VGA монитором, и минимум 2 MB расширенной памяти.
- ◆ 4MB свободного дискового пространства для установки программного обеспечения
- ◆ DOS 3.1 или выше.
- ◆ Microsoft Windows 3.1 или выше.

Программа установки создает следующую структуру каталогов:

| | |
|------------------------------|---|
| 21XX\BIN | Ассемблер, редактор связей, программа моделирования, разделитель программ для записи в ППЗУ |
| 21XX\ETC | Сообщения об ошибках и справочная информация |
| 21XX\INCLUDE | Заголовочный файл для ADSP-2181 |
| 21XX\LIB | Файл описания архитектуры |
| 21XX\EZKITL | Базовая Windows программа и демонстрационные файлы |
| 21XX\EZKITL\2181\MONITOR | Исходный код программы монитора. |
| 21XX\EZKITL\2181\DSP | Исходный код примеров программ |
| 21XX\EZKITL\2181\DSP\ADPCM | |
| 21XX\EZKITL\2181\DSP\DTMF | |
| 21XX\EZKITL\2181\DSP\ECHO | |
| 21XX\EZKITL\2181\DSP\FIRDEMO | |
| 21XX\EZKITL\2181\DSP\LPC2K4 | |
| 21XX\EZKITL\2181\DSP\LPC7K8 | |

Установка аппаратного обеспечения

На приведенной диаграмме показано подключение источника питания, порта RS-232, звукового вход/выхода. После выполнения подключения, включите питание и нажмите на плате кнопку RESET.



Плата ADSP-2181 EZ-LAB

Процедура установки



4

Установка программного обеспечения

В состав EZ-KIT Lite включена дискета для установки программного обеспечения. Запуск программы установки должен производиться в среде Windows™. Программа инсталляции создаст требуемые директории и поддиректории и установит соответствующие файлы.

Процедура установки программного обеспечения

Чтобы запустить программное обеспечение EZ-KIT Lite требуется:

- ◆ ПК с процессором 386 (или выше), жестким диском, 3.5" дюймовым дисководом, цветовой видеокартой с VGA монитором, и минимум 2 MB расширенной памяти.
- ◆ 4MB свободного дискового пространства для установки программного обеспечения
- ◆ DOS 3.1 или выше.
- ◆ Microsoft Windows 3.1 или выше.

Шаг 1: Рабочая копия дискеты

Перед установкой программного обеспечения нужно скопировать оригинал дискеты на рабочий диск. Это может быть выполнено с помощью команды XCOPY или COPY (не пользуйтесь командой DISKCOPY). Оригинальные дискеты должны быть сохранены в надежном месте и использованы только для создания копии на рабочем диске.

Шаг 2: Изменения файла CONFIG.SYS

Для гарантирования правильной работы программного обеспечения, добавьте (или измените) директиву FILES=25 в файле CONFIG.SYS (переменная среды окружения FILES может быть установлена и числом большим, чем 25). Необходимо также добавить (или изменить) директиву BUFFERS=30. После изменения файла CONFIG.SYS, компьютер должен быть перезапущен для вступления в действие внесенных изменений.

Шаг 3: Установка программного обеспечения на жесткий диск

При установке программного обеспечения следуйте следующим инструкциям:

1. Проверьте, что операционная система Windows™ запущена.
2. Вставьте в дисковод дискету с программным обеспечением EZ-KIT Lite
3. Перейдите к окну менеджера программ (или проводника)
4. Выберите из меню *File* (или меню *Пуск*) пункт *Run* (*Выполнить...*).
5. В командной строке наберите `A:\SETUP.EXE` (или `B:\SETUP.EXE`)
6. Нажмите кнопку `OK` или `ENTER`
7. Следуйте инструкциям на экране

Примечание: необходимо внести следующие изменения в файл `AUTOEXEC.BAT`:

1. Заведите переменную среды окружения `SET ADI_DSP=C:\ADI_DSP`
2. В переменной `PATH` среды окружения допишите `C:\ADI_DSP\21XX\BIN`

Программа установки создает следующую структуру каталогов:

| | |
|------------------------------|---|
| 21XX\BIN | Ассемблер, редактор связей, программа моделирования, разделитель программ для записи в ППЗУ |
| 21XX\ETC | Сообщения об ошибках и справочная информация |
| 21XX\INCLUDE | Заголовочный файл для ADSP-2181 |
| 21XX\LIB | Файл описания архитектуры |
| 21XX\EZKITL | Базовая Windows программа и демонстрационные файлы |
| 21XX\EZKITL\2181\MONITOR | Исходный код программы монитора. |
| 21XX\EZKITL\2181\DSP | Исходный код примеров программ |
| 21XX\EZKITL\2181\DSP\ADPCM | |
| 21XX\EZKITL\2181\DSP\DTMF | |
| 21XX\EZKITL\2181\DSP\ECHO | |
| 21XX\EZKITL\2181\DSP\FIRDEMO | |
| 21XX\EZKITL\2181\DSP\LPC2K4 | |
| 21XX\EZKITL\2181\DSP\LPC7K8 | |

Переменные среды окружения

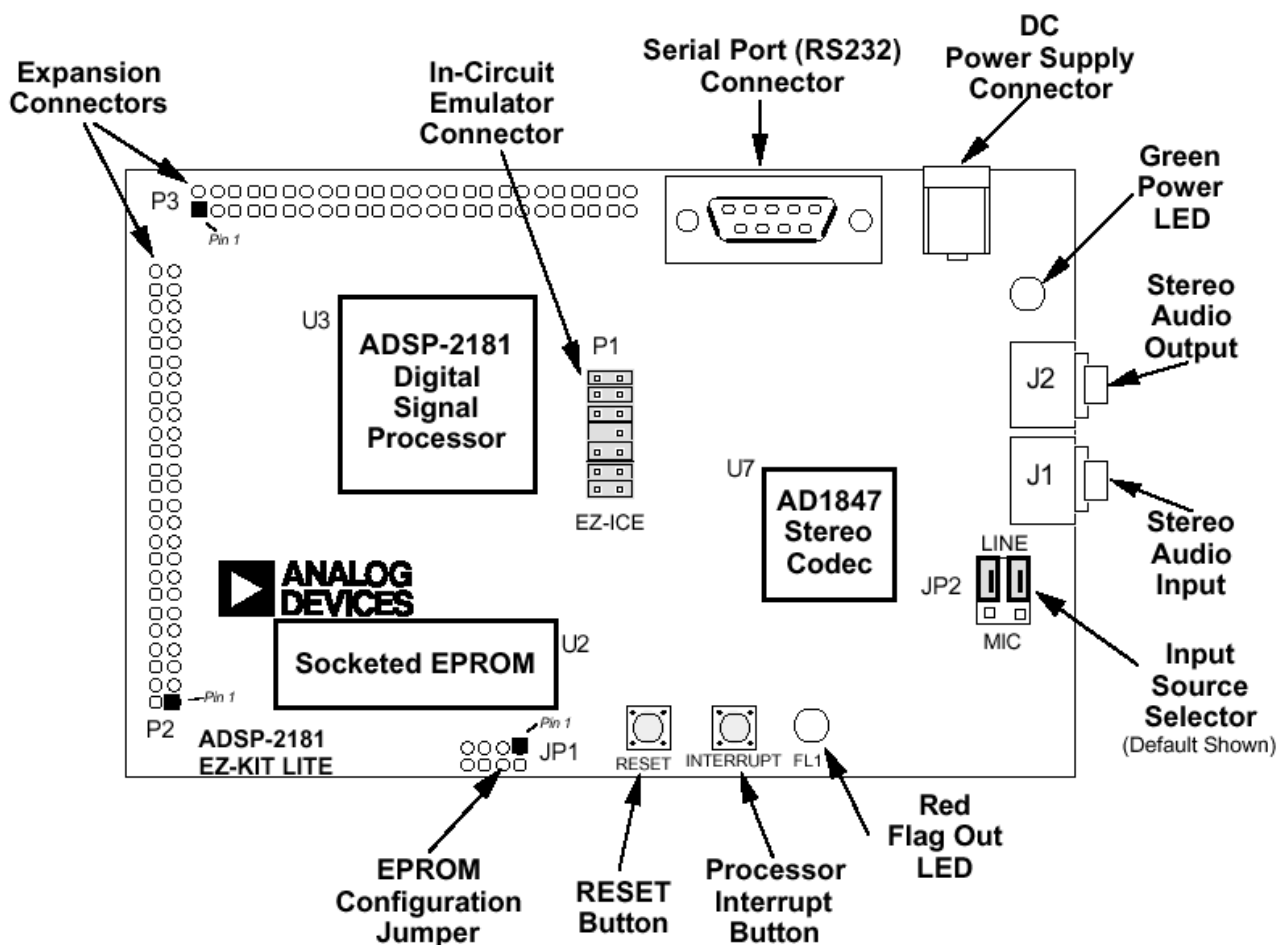
Следующие переменные среды окружения должны быть созданы и обозначены правильными параметрами (они должны быть скопированы в ваш файл AUTOEXEC.BAT, чтобы быть запущенными автоматически при запуске ПК):

ADI__DSP Путь к каталогу, содержащему установленные файлы
(по умолчанию ADI__DSP=C:\ADI__DSP\)

Процедура установки аппаратного обеспечения

Положите плату на ровной поверхности. Подсоедините J3 к последовательному порту (COM). Если последовательный порт ПК является IBM совместимым, то кабель соединит входящий разъем DB9 с выходящим. Однако может оказаться, что ПК имеет 25-штырьковый разъем вместо 9-штырькового. В этом случае нужно использовать специальный переходник.

Подсоедините усиливающие динамики к J2. Например, это могут быть динамики фирмы Labtec CS-550. Проверьте, что у динамика присутствует усилитель. К J1 также могут быть подключен микрофон или линия источника звука.



Плата ADSP-2181 EZ-LAB

Разъем JP2 должен быть установлен в соответствии с типом входного подключения к J1. Например, это могут быть микрофоны фирмы Radio Shack 33-1060 или Labtec AM-22.

Подключите к J4 источник с напряжением 9 В постоянного тока, способный обеспечить не менее 300 мА. Подключение должно быть осуществлено через разъем P5 с наружным диаметром 5.5 мм и внутренним диаметром 2.1 мм. Внешний рукав разъема должен быть положительной полярности. При подаче питания не включается зеленый светодиод, обозначенный как D2. При загрузке DSP из EPROM через порт RS-232 посылается соответствующее сообщение, через динамики воспроизводится звуковой сигнал, и начинает мигать красный светодиод D1.

Если не горит зеленый светодиод, проверьте соединение с источником питания. Убедитесь, что источник питания имеет необходимый соединитель и правильную полярность. Напряжение источника питания, измеренное в месте подключения к плате, должно быть в пределах от 8 В до 10 В постоянного тока. Проверьте, что сверху и снизу платы нет предметов, которые могут быть причиной короткого замыкания цепи.

Если соединение источника питания хорошее, и горит зеленый светодиод, но красный светодиод не мигает и отсутствует звуковой сигнал, убедитесь, что микросхема EPROM правильно размещена в панельке.

Если проявляется неправильная работа платы, нажмите кнопку RESET.

Разработка DSP системы

**5**

Обзор

Для разработки собственных программ можно воспользоваться программным обеспечением разработчика поставляемым с EZ-KIT Lite. При отсутствии опыта в разработке программ для систем, основанных на DSP, нужно следовать приведенным в этой главе действиям. Представленные шаги разработки служат руководящими указаниями для создания собственных программ. Принимается во внимание, что процесс разработки изменяется в зависимости от стиля конкретного разработчика. Приведенные действия следует рассматривать как отправную точку, и поэтому не стесняйтесь, если потребуется изменить их согласно собственному стилю работы.

Шаг 1: Системные требования

Первый шаг при создании DSP системы состоит в том, чтобы определить какие возможности системы потребуются. Эти возможности будут зависеть от типов применяемых алгоритмов, типов используемых сигналов и типов входных/выходных приборов, которые необходимо подключить к DSP процессору. Расчет размера требуемой памяти данных и памяти программ основан на количестве данных, требуемых системам ввода/вывода, и процессов обработки, которые должны быть выполнены. Предполагаемый размер программ, созданных в соответствии с используемыми алгоритмами, определяет требования к памяти программ.

Например, при создании системы обработки речевого сигнала выбирается подходящий алгоритм для сжатия речевого сигнала, допустим, LPC. Этот алгоритм требует сохранения некоторого количества данных и хранения некоторого числа программных инструкций. Конечно, пока этот алгоритм не выполняется, эти требования могут быть только приблизительными. Например, установим, что требуется 4К слов памяти данных и 1К слов памяти программ. Может потребоваться А/Ц или Ц/А преобразование. В этом случае к последовательному порту DSP подключается звуковой кодек. Необходимо, чтобы DSP процессор мог поддерживать скорость выборок данных. Частота 33MHz ADSP-2181 для этого более чем достаточна.

Шаг 2: Проектирование системы

После того, когда системные требования определены, необходимо спроектировать оборудование системы. EZ-KIT Lite использует для А/Ц и Ц/А преобразований звуковой кодек AD1847. AD1847 подключен к последовательному порту 0 (SPORT0). Используется только внутренняя память ADSP-2181 (16К слов памяти программ и 16К слов памяти данных), поэтому внешняя память не подключена. Для связи через интерфейс RS-232 используются сигналы последовательного порта 1 (SPORT1).

Шаг 3: Файл описания архитектуры

При использовании средств разработки для процессоров семейства ADSP-2100, оборудование системы нужно перечислить в файле описания архитектуры. Поскольку система ADSP-2181 уже определена, этот шаг уже сделан. Описание содержится в файле `ADSP2181.ACH` (файл созданный системным конфигуратором), который включен в программное обеспечение для EZ-KIT Lite.

```
.system demo;
.adsp2181;
.mmap0;

.seg/pm/ram/abs=0/code/data      int_pm_lo[8192];
.seg/pm/ram/abs=8192/code/data   int_pm_hi[8192];
.seg/dm/ram/abs=0/data           int_dm_lo[8192];
.seg/dm/ram/abs=8192/data        int_dm_hi[8160];

.endsys;
```

Первые три строчки файла определяют имя системы, использующей ADSP-2181 с возможностью начальной загрузки из EPROM (на вывод MMAP подается низкий уровень).

16K слов внутренней памяти программ описаны как два сегмента по 8K слов, которые могут содержать как коды, так и данные. Первая секция 8K слов будет всегда внутри ADSP-2181. Вторая секция может быть запрограммирована с помощью инструкций ADSP-2181 (установка режимов `PMOVLAY`) как внутренний или внешний оверлей. Архитектура с двумя секциями упрощает управление памятью для приложений, которые могут использовать дополнительную внешнюю память. Память может быть описана и как единая 16K секция.

Память данных описана подобно памяти программ. 32 слова резервируют для регистров управления отображенных в памяти ADSP-2181 .

Этот текстовый файл (`.SYS`) используется как входные данные системного конфигулятора для создания `.ACH` файла. Файл `.ACH` используется редактором связей и программой моделирования. Файл `ADSP2181.ACH` поставляется с программным обеспечением для ADSP-2181 EZ-KIT Lite. Для определения отличной архитектуры системы, потребуется обратиться к программе системного конфигулятора.

Шаг 4: Разработка программного кода

Когда оборудование определено (или приближено к требованиям), можно начинать разработку программного кода. Во-первых, определите требования к размеру памяти, необходимого для хранения переменных и массивов и все используемые прерывания. Предусмотрите любое оборудование или регистры, которые нуждаются в инициализации.

Написание программы осуществляется вводом инструкций в текстовый файл, который впоследствии обрабатывается программой ассемблера. Ассемблер транслирует набор алгебраических инструкций процессора в бинарный объектный файл. Существует несколько вещей, которые следует учесть, в частности, о языке ассемблера ADSP-2100/

Алгебраический синтаксис использует символ «=» для представления переноса данных:

`AX0 = MX0 ;`

В приведенном примере данные из регистра `MX0` переносятся в регистр `AX0`. Символы «+», «-», «*» используются для обозначения арифметических операций. Например, инструкция

`AR = AX0 + AY0 ;`

добавляет содержимое регистра `AX0` к содержимому регистра `AY0` и размещает результат (сумму) в регистре `AR`.

Все инструкции должны заканчиваться символом «;». Многофункциональные инструкции отделяются символом «,». Следующий пример показывает многофункциональные инструкции процессоров ADSP-21xx.

`MR = MR + MX1 * MY(SU), MX1 = DM(I0,M3), MY1 = PM(I4,M5) ;`

Первая инструкция (до первой запятой) это операция умножения/накопления (MAC). Содержимое входных регистров `MX1` и `MY1` перемножается и добавляется к содержимому регистра результата умножителя. Вторая инструкция загружает регистр `MX1` из памяти данных (DM), а третья загружает регистр `MY1` из памяти программ (PM). Регистры I используются для хранения адресов, а регистры M хранят величину, используемую для модификации адреса. Все эти инструкции могут быть выполнены за один процессорный цикл.

Объявление любых переменных, массивов или констант, также как и указание включаемых файлов и портов ввода/вывода, делается в начале программы, используя директивы ассемблера.

После перезапуска системы, ADSP-2181 начнет выполнение кода из памяти программ со стартового адреса 0. Из приведенного ниже листинга видно, что там расположена инструкция `jump start`. Она используется для обхода таблицы векторов прерываний.

При прерывании исполнение программы останавливается и переходит к адресу, указанному в таблице векторов прерываний. Кодовый сегмент, показанный ниже, включает таблицу векторов прерываний и размещается в первых 48 адресах памяти программ.

Для удобства в поле комментария показаны шестнадцатеричные адреса памяти. Необходимо заполнить неиспользованные адреса в памяти таблицы векторов инструкцией `rti`. Это делается в целях безопасности. Инструкция `rti` обеспечит перенос исполнения обратно в основную программу.

Следующий пример показывает процесс инициализации аппаратной части EZ-KIT Lite, чтение значений с аналогового входа и их вывод.

```
. module/RAM/ABS=0    my_program;

(**** Объявление констант ****)
(управляющие регистры ADSP-2181 отображенные в памяти)
.const IDMA =                0x3fe0;
.const BDMA_BIAD =           0x3fe1;
.const BDMA_BEAD =           0x3fe2;
.const BDMA_BDMA_Ctrl =      0x3fe3;
.const BDMA_BWCOUNT =        0x3fe4;
.const PFDATA =              0x3fe5;
.const PFTYPE =              0x3fe6;
.const SPORT1_Autobuf =      0x3fef;
.const SPORT1_RFSDIV =       0x3ff0;
.const SPORT1_SCLKDIV =      0x3ff1;
.const SPORT1_Control_Reg =  0x3ff2;
.const SPORT0_Autobuf =      0x3ff3;
.const SPORT0_RESDIV =       0x3ff4;
.const SPORT0_SCLKDIV =      0x3ff5;
.const SPORT0_Control_Reg =  0x3ff6;
.const SPORT0_TX_Channels0 = 0x3ff7;
.const SPORT0_TX_Channels1 = 0x3ff8;
.const SPORT0_RX_Channels0 = 0x3ff9;
.const SPORT0_RX_Channels1 = 0x3ffa;
.const TSCALE =              0x3ffh;
.const TCOUNT =              0x3ffc;
.const TRERIOD =             0x3ffd;
.const DM_Wait_Reg =         0x3ffe;
.const System_Control_Reg =  0x3fff;

(**** Объявления переменных и буферов ****)
.var/dm/ram/circ  rx_buf[3];          /* приемный буфер AD1847 */
.var/dm/ram/circ  tx_buf[3];          /* передающий буфер AD1847 */
.var/dm/ram/circ  init_cmds[13];
.var/dm           stat_flag;

(**** Инициализация переменных и буферов ****)
.init tx_buf:  0xc000, 0x0000, 0x0000;
.init init_cmds:
    0xc003,      (регистр управления левого входа AD1847)
    0xc103,      (регистр управления правого входа AD1847)
    0xc288,      (регистр управления левого 1 внешнего входа AD1847)
    0xc388,      (регистр управления левого 1 внешнего входа AD1847)
    0xc488,      (регистр управления левого 2 внешнего входа AD1847)
    0xc588,      (регистр управления левого 2 внешнего входа AD1847)
    0xc680,      (регистр управления левого ЦАП AD1847)
    0xc780,      (регистр управления правого ЦАП AD1847)
    0xc85b,      (регистр формата данных AD1847)
    0xc909,      (регистр конфигурации интерфейса AD1847)
    0xca00,      (регистр управления выводом AD1847)
    0xcc40,      (регистр различной информации AD1847)
    0xcd00;      (регистр управления цифрового миксера AD1847)

(**** Таблица векторов прерываний ****)
jump start;          (Адрес 0000: перезапуск)
rti;
rti;
rti;

rti;                  (Адрес 0004: IRQ2)
rti;
rti;
rti;
```

```

rti;                (Адрес 0008: IRQL1)
rti;
rti;
rti;

rti;                (Адрес 000C: IRQL0)
rti;
rti;
rti;

ar = dm(stat_flag); {Адрес 0010: SPORT0 tx}
ar = pass ar;
if eq rti;
jump next_cmd;

jump input_samples; {Адрес 0014: SPORT0 rx}
rti;
rti;
rti;

rti;                (Адрес 0018: IRQE)
rti;
rti;
rti;

rti;                (Адрес 001C: BDMA)
rti;
rti;
rti;

rti;                (Адрес 0020: SPORT1 tx или IRQ1)
rti;
rti;
rti;

rti;                (Адрес 0024: SPORT1 rx или IRQ0)
rti;
rti;
rti;

rti;                (Адрес 0028: таймер)
rti;
rti;
rti;

rti;                (Адрес 002C: power down)
rti;
rti;
rti;

{***** Инициализация ADSP-2181 *****)
start:
    i0 = ^rx_buf;    {установка указателя на начало буфера}
    l0 = %rx_buf;    {установка длины буфера}
    i1 = ^tx_buf;
    l1 = %tx_buf;
    i3 = ^init_cmds;
    l3 = %init_cmds;
    m1 = 1;

```

```
{**** Последовательный порт 0 (SPORT0) ****}
ax0 = b#0000001010000111;
dm(SPORT0_Autobuf) = ax0;

ax0 = 0;
dm(SPORT0_RFSDIV ) = ax0;
dm(SPORT0_SCLKDIV) = ax0;
ax0 = b#1000011000001111;
dm(SPORT0_Control_Reg) = ax0;

ax0 = b#00000000000000111;
dm(SPORT0_TX_Channels0) = ax0;

ax0 = b#00000000000000111;
dm(SPORT0_TX_Channels1) = ax0;

ax0 = b#00000000000000111;
dm(SPORT0_RX_Channels0) = ax0;

ax0 = b#00000000000000111;
dm(SPORT0_RX_Channels1) = ax0;

{**** Последовательный порт 1 (SPORT1) ****}
ax0 = 0;
dm(SPORT1_Autobuf) = ax0;      {автобуферирование недоступно}
dm(SPORT1_RFSDIV ) = ax0;      {RFSDIV не используется}
dm(SPORT1_SCLKDIV) = ax0;      {SCLKDIV не используется}
dm(SPORT1_Control_Reg) = ax0; {функции ctrl не установлены}

{**** Установка таймера ****}
ax0 = 0;
dm(TSCALE) = ax0;              {таймер не будет использоваться}
dm(TCOUNT) = ax0;
dm(TPERIOD) = ax0;

{**** Установка системы и памяти ****}
ax0 = b#0000000000000000;
dm(DM_Wait_Reg) = ax0;
ax0 = b#0001000000000000;      {доступен SPORT0 }
dm(System_Control_Reg) = ax0;

ifc = b#0000001111111111;      {очистка очереди прерываний}
nop;

icntl = b#00000;
mstat = b#1000000;

{**** Инициализация кодека AD1847 ****}
ax0 = 1;
dm(stat_flag) = ax0;            {очистка флага}
imask = b#0001000000;          {доступны прерывания}

ax0 = dm(i1, m1);
tx0 = ax0;

check_init:
ax0 = dm(stat_flag);            {ожидание до целой выборки}
af = pass ax0;                  {буфер переслать к кодеку}
if ne jump check_init;

ay0 = 2;
```

```

check_acih:
    ax0 = dm(rx_buf);           {ожидание инициализации пока}
    ar = ax0 and ay0;           { кодек выйдет из автокалибровки}
    if eq jump check_acih;

check_acil:
    ax0 = dm(rx_buf);           {ожидание инициализации пока}
    ar = ax0 and ay0;           {кодек выйдет из автокалибровки}
    if ne jump check_acil;
    idle;

    ay0 = 0xbf3f;               {включение левого DAC}
    ax0 = dm(init_cmds + 6);
    ar = ax0 AND ay0;
    dm(tx_buf) = ar;
    idle;

    ax0 = dm(init_cmds + 7);     {включение правого DAC}
    ar = ax0 AND ay0;
    dm(tx_buf) = ar;
    idle;

    ax0 = 0xc901;               {очистка требования автокалибровки}
    dm(tx_buf) = ax0;
    idle;

    ax1 = 0x8000; {управляющее слово очистки}
    dm(tx_buf) = ax1;

    ifc = b#0000001111111111;   {очистка очереди прерываний}
    nop;

    imask = b#0000100000;       {разрешение прерывания rx0}

{**** Ожидание прерывания ****}
talkthru: idle;
    jump talkthru;

{**** Подпрограммы обслуживания прерываний ****}
{**** прерывание по приему ****}
input_samples:
    ena sec_reg;                {использование теневого банка регистров}
    ax1 = dm(rx_buf + 1);       {получение данных от кодека}
    mx = dm(rx_buf + 2);
    dm(tx_buf + 1) = ax1;       {пересылка данных в кодек}
    dm(tx_buf + 2) = mx1;
    rti;

{**** прерывание по передаче для инициализации кодека ****}
next_cmd:
    ena sec_reg;
    ax0 = dm(i3, m1);           {получение следующего управляющего слова и}
    dm(tx_buf) = ax0;           { размещение в передающий слот 0}
    ax0 = i3;
    ay0 = ^init_cmds;
    ar = ax0 - ay0;
    if gt rti;                 {rti если больше управляющих слов}
    ax0 = 0x8000;               {еще установить флаг и }
    dm(tx_buf) = ax0;           {убрать MSE если завершена}
    ax0 = 0;
    dm(stat_flag) = ax0;        {переустановить флаг состояния }
    rti;
.endmod;

```

Шаг 5: Запуск ассемблера

После завершения написания текстового файла, который содержит программу на языке ассемблера можно запустить ассемблер следующей командой:

```
asm21 my_prog -2181
```

В этом примере ассемблируется текстовый файл `my_prog.dsp`. Ключ `-2181` указывает ассемблеру на необходимость присоединить специальные инструкции процессора ADSP-2181. Ассемблер создает соответствующий объектный файл или файлы.

Существует некоторое число дополнительных ключей ассемблера. Например, для создания файла листинга `my_prog.lst` используется следующий ключ:

```
asm21 my_prog -2181 -l
```

Ассемблер создает объектный файл с расширением `.obj`.

Шаг 6: Запуск редактора связей

Редактор связей создает исполняемый файл из объектного модуля созданного ассемблером. Следующий пример создает исполняемый файл с именем `demo.exe`:

```
ld21 my_prog -a adsp2181 -e demo
```

Файл описания архитектуры `adsp2181.ach` определяется ключом `-a`, а имя исполняемого файла выбирается ключом `-e`. Существует дополнительные ключи, которые позволяют создавать файл карты распределения памяти карты, файл с таблицей символов и др.

Шаг 7: Запуск программы моделирования

Программа моделирования позволяет запускать код в моделируемом окружении, чтобы проверить ваше программное обеспечение без использования реального оборудования системы. Шаг моделирования программы используют, чтобы удостовериться в работоспособности программы.

В большинстве случаев он позволяет избежать ситуации, когда вы загружаете ваше программное обеспечение в аппарат, и оно не работает. Если вы проверили работу вашего программного обеспечения на программе моделирования, но программа в аппарате не запускается, это может быть вызвано дефектом аппаратной части. Программа моделирования вызывается следующим образом:

```
sim2181 -a adsp2181 -e demo
```

Эта команда запускает программу моделирования ADSP-2181 для исполняемого файла `demo.exe` и системы, описанной в файле архитектуры `adsp2181.ach`. Если программа моделирования не может найти какой-либо из этих файлов, необходимо проверить установки путей в `autoexec.bat` или определить полный путь:

```
sim2181 -a c:\adi_dsp\21xx\lib\adsp2181 -e demo
```

Шаг 8: Программирование EPROM

После проверки, что программное обеспечение работает, можно разделить программу для программирования EPROM. Микросхема EPROM может быть вставлена в кровать на плате для запуска вашей программы. Преобразователь кодов для программирования ППЗУ может быть вызван следующей командой:

```
spl21 demo demoprom -loader -2181
```

Эта команда создает из исполняемого файла `demo.exe` PROM файл с именем `demoprom.bnm`. По умолчанию формат этого файла является записью Motorola S. При указании ключа `-i` можно задать запись формата Intel Hex.

После включения питания или когда вы нажали кнопку перезапуска на плате, содержимое EPROM автоматически загрузится во внутреннюю память программ и память данных процессора и начнется выполнение команд.

Шаг 9: Запуск платы ADSP-2181 EZ_KIT Lite

Плата EZ-KIT Lite имеет микросхему EPROM, которая содержит программу ADSP-2181. Когда на плату подают питание (или ее перезагружают), коды автоматически передаются из EPROM во внутреннюю память ADSP-2181. Коды загружаются вместе с включением программы монитора, которая позволяет ADSP-2181 связаться с ПК по интерфейсу RS-232.

Коды выполняют самотестирование, а затем посылают звуковой сигнал на выход кодека. Вам потребуется установить громкоговоритель, подключенный к разъему звукового выхода, чтобы услышать производимый звук. Сигнал на этом разъеме низкого уровня, таким образом, необходимо установить динамики, которые имеют встроенные усилители.

После установки программного обеспечения, можно запустить хост программу EZ-KIT Lite. Вы сможете загрузить исполняемый файл (.exe файл созданный редактором связей).

Шаг 10: Отладка

Обычно, если ваш EZ-KIT Lite не работает должным образом с поставленными программами, проблема чаще всего заключается в подключении питания, или подключении RS-232, или с компонентами на плате. Убедитесь, что все соединения сделаны плотно, и напряжение питания лежит в пределах от 8 В до 10 В выпрямленного тока. Убедитесь также, что EPROM должным образом установлена в кровать, и нет предметов оставленных на или под платой, вызывающих короткое замыкание цепи.

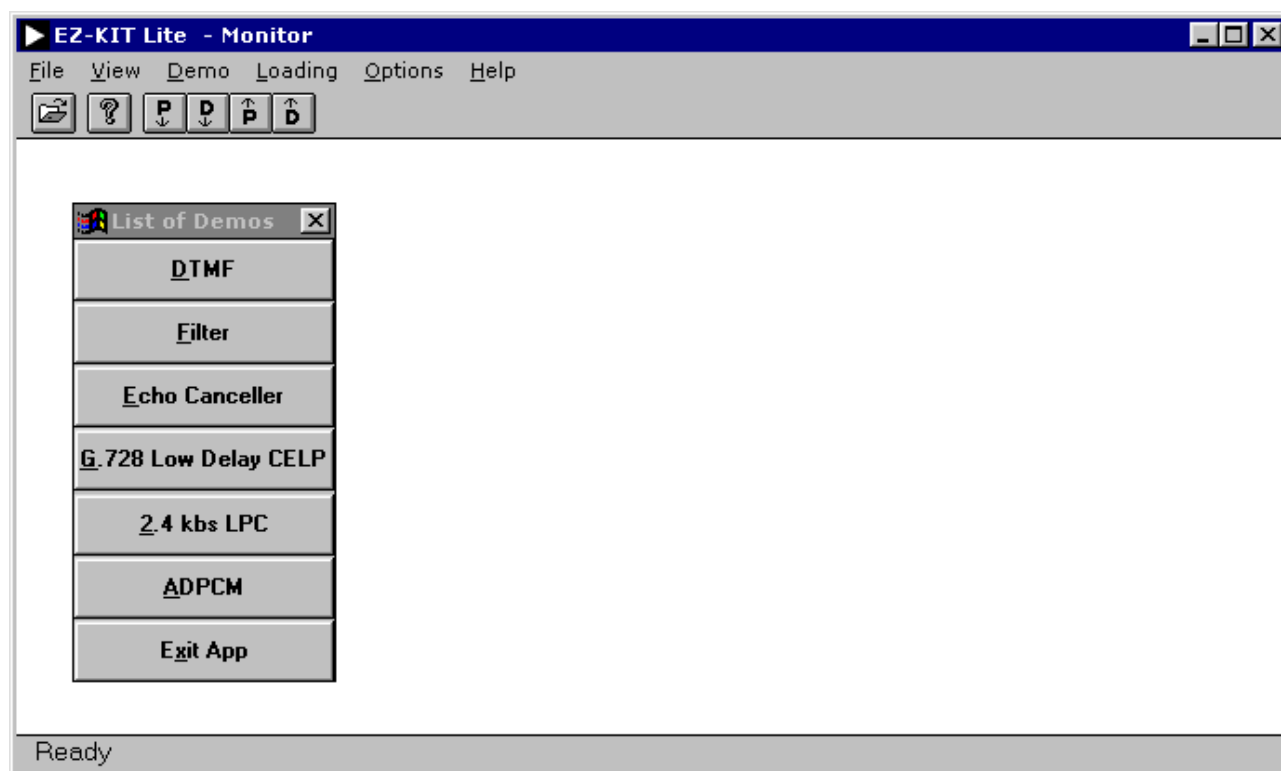
Если сомнения позади, нажмите кнопку перезапуска.

Хост программа EZ-KIT Lite

**6**

Обзор программы

Базовая (хост) программа EZ-KIT Lite является Windows приложением, следующим стандарту соглашения GUI. Эта программа используется для связи с макетной платой EZ-KIT Lite. С помощью нее можно запустить демонстрационные программы EZ-KIT Lite, выгрузить/загрузить содержимое памяти программ и памяти данных, загрузить и выполнить собственные DSP программы. Следующий экран демонстрирует внешний вид хост программы EZ-KIT Lite.



Сводка команд

Этот раздел кратко описывает команды, которые доступны из главного меню:

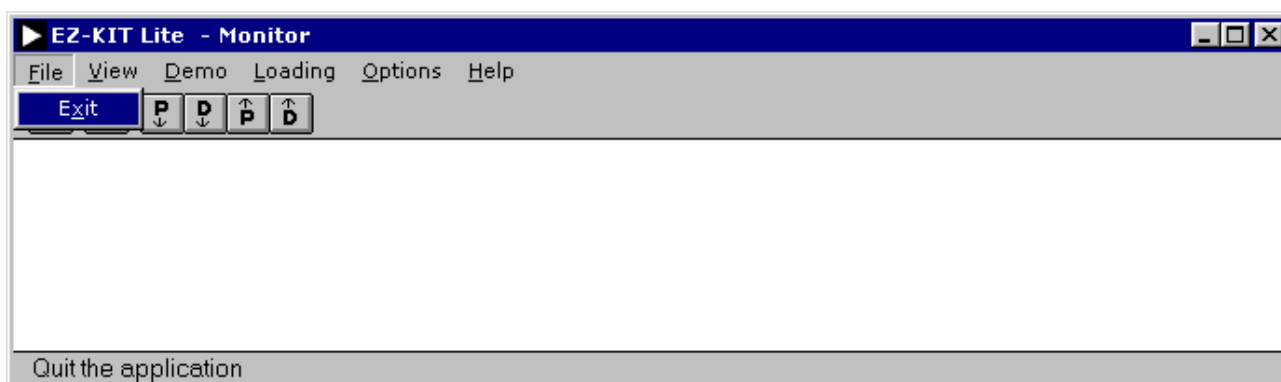
- **DTMF**
запускает демонстрационные программы DTMF
- **Filtering**
запускает демонстрационную программу фильтрации
- **Echo Cancellation**
запускает демонстрационную программу эхоподавления
- **Speech Compression**
запускает демонстрационные программы сжатия речевого сигнала LPC и ADPCM
- **Download DSP program & Go**
загружает пользовательскую программу и начинает выполнение
- **Download DSP program**
загружает пользовательскую программу, но не приступает к выполнению,
загружает файлы образа памяти
- **Go**
начинает выполнение с указанного адреса
- **Upload data memory**
выгружает блок памяти данных в файл образа памяти
- **Upload program memory**
выгружает блок памяти программ в файл образа памяти
- **Download data memory**
загружает файл образа памяти данных с указанного адреса
- **Download program memory**
загружает файл образа памяти программ с указанного адреса
- **About**
последняя строка в вызываемом диалоговом окне, указывает на наличие
последовательной связи с макетной платой.

Детализация команд

Все команды доступны через стандартную панель меню. Можно заметить, что вдобавок к стандартным пунктам меню Windows, например, «File» и «Help», присутствуют несколько подменю, относящихся к EZ-KIT Lite. Все пункты меню детально описаны ниже.

Меню «File»

Так как в памяти изначально нет информации, стандартные команды сохранения и открытия файлов не поддерживаются. Существует только одна команда – выход.

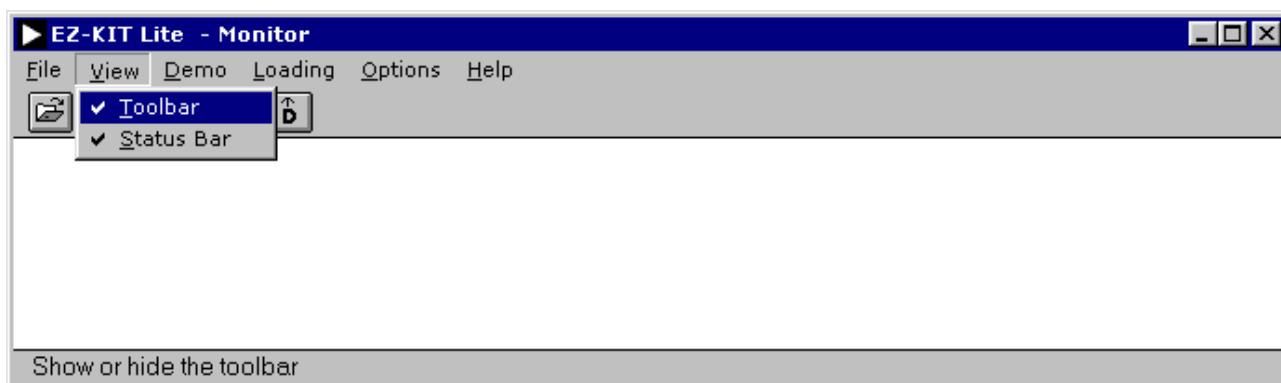


Exit

Выберите этот пункт, если хотите выйти из программы.

Меню «View»

Внешний вид меню представлен ниже.



Toolbar

Выбор этого пункта позволяет вызвать/скрыть панель инструментария, находящуюся под главным меню. Панель инструментария содержит кнопки, которые позволяют быстро вызвать необходимую команду из:

– *Download user program & Go (Загрузка пользовательской программы & Запуск)*

Это первая кнопка слева на панели инструментария с символом открытой папки. Она соответствует команде из меню «Loading», загружая пользовательскую программу и начиная ее выполнение. Выберите исполняемый файл ADSP-2181 с расширением .exe, который создан редактором связей ADSP-2181.

– *About EzkitApp (О программе Ezkit)*

Это следующая кнопка с символом вопроса. Она соответствует команде из меню «Help», вызывая диалоговое окно с информацией о программе. Может быть полезным прочитать информацию на последней строчке. Если хост программа соединилась с программой монитора, то там будет строчка «EZ-KIT monitor is alive and well». В противоположном случае, строчка будет содержать «EZ-KIT monitor is not running; try reset». Можно нажать кнопку сброса на плате, чтобы перезапустить программу монитор.

– *Download program memory (Загрузка памяти программ)*

Это кнопка с буквой «P» и стрелкой, направленной вниз. Она соответствует команде из меню «Loading», загружая файл с образом памяти программ в DSP, начиная с указанного адреса. Следует выбрать команду «Go», чтобы запустить программу.

– *Download data memory (Загрузка памяти данных)*

Это кнопка с буквой «D» и стрелкой, направленной вниз. Она соответствует команде из меню «Loading», загружая файл с образом памяти данных в DSP, начиная с указанного адреса.

– *Upload program memory (Выгрузка памяти программ)*

Это кнопка с буквой «P» и стрелкой, направленной вверх. Она соответствует команде из меню «Loading», выгружая копию памяти программ DSP в файл. Указывается начальный адрес и количество ячеек памяти программ.

– *Upload data memory (Выгрузка памяти данных)*

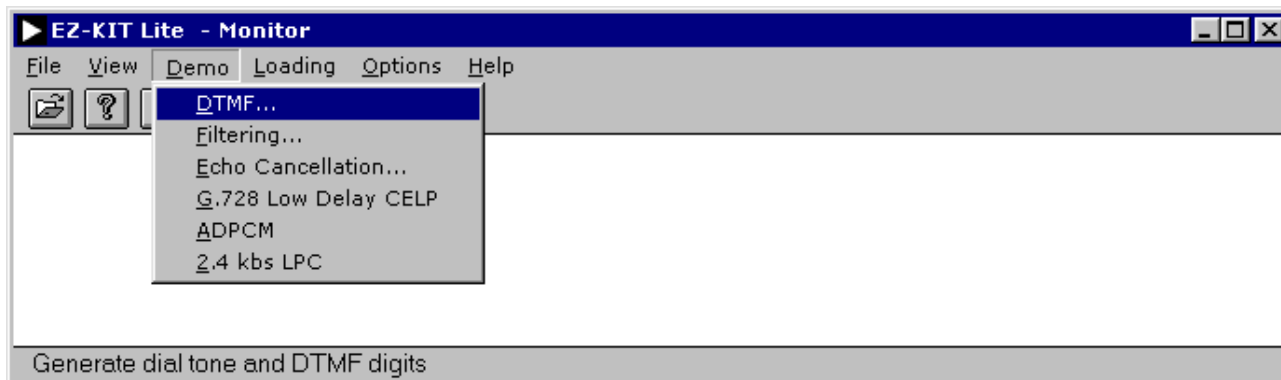
Это кнопка с буквой «D» и стрелкой, направленной вверх. Она соответствует команде из меню «Loading», выгружая копию памяти данных DSP в файл. Указывается начальный адрес и количество ячеек памяти данных.

Status Bar

Выбор этого пункта позволяет вызвать/скрыть строку текущего состояния, находящуюся внизу экрана. Строка состояния показывает детализированное описание команд меню.

Меню «Demo»

В этом меню можно выбрать одну из доступных демонстрационных программ для запуска на EZ-KIT Lite. Кроме того, эти программы могут быть выбраны из всплывающего окна списка демонстрационных программ.



DTMF

Эта демонстрационная программа генерирует DTMF сигналы - двухтональный многочастотный набор, которые используются в телефонных сетях для сигнализации нажатой кнопки. Тон DTMF составлен из двух различных одночастотных тонов - один из четырех тонов ряда складывается с одним из четырех тонов колонки. Таким образом, в полном соответствии стандарту тонов DTMF, генератор может выдавать 16 различных тонов (обычно используются только 12).

Когда выбран DTMF тон, EZ-KIT Lite генерирует две требуемых синусоидальные волны, масштабирует их, складывает и затем выводит результат на кодек для преобразования в аналоговый сигнал. При запуске программы, EZ-KIT Lite генерирует тон номеронабирателя в выходной динамик. Возможен следующий выбор:

– *Generate dial tone (Генерировать тональный вызов)*

Генерируется стандартный тон номеронабирателя, являющийся суммой незатухающих синусоидальных волн.

– *Generate DTMF digits (Генерировать DTMF номер)*

Генерируется запрограммированный в демонстрационной программе сигнал телефонного номера инженера по применению DSP Analog Devices. Если установить на выходе динамика трубку в состоянии обнаружения токового состояния линии, DTMF тоны укажут телефонной сети вызывать соединение телефоном инженера по применению.

– *Close (Заккрыть)*

Выход из демонстрационной программы DTMF и повторный вызов резидентной программы монитора EZ-KIT Lite. Позволяет выбрать новую демонстрационную программу.

Filtering

Эта программа демонстрирует действие четырех полосовых фильтров на сигнале, поступающем с микрофонного входа, или сигнале, генерируемом источником шума.

Когда выбран интерфейс Windows приложений, будет загружена необходимая DSP программа в EZ-KIT Lite и показано следующее диалоговое окно.

Демонстрация начинается со сквозной передачи через кодек неотфильтрованного речевого сигнала. AD1847 преобразует сигнал с аналогового входа в цифровое представление и передает данные по последовательному порту DSP. DSP читает данные из последовательного порта и передает данные обратно к кодеку. Кодировщик преобразует цифровые данные в аналоговый сигнал, который подается на громкоговоритель. Не применяется никакой цифровой обработки.

Верхний ряд управляющих элементов выбирает фильтр для речевого сигнала. Фильтры имеют эквивалентную полосу пропускания и равномерно расположены на логарифмической оси частот. Все КИХ-фильтры состоят из 256 точек и имеют неравномерность 0.1.

| КИХ-фильтр | Нижняя полоса подавления | Полоса пропускания | Верхняя полоса подавления |
|-------------------|---------------------------------|---------------------------|----------------------------------|
| FIR1 | 0 – 269 Гц | 328 – 448 Гц | 547 – 4000 Гц |
| FIR2 | 0 – 426 Гц | 521 – 710 Гц | 866 – 4000 Гц |
| FIR3 | 0 – 675 Гц | 825 – 1125 Гц | 1375 – 4000 Гц |
| FIR4 | 0 – 1070 Гц | 1308 – 1783 Гц | 2179 – 4000 Гц |

Нижний ряд управляющих элементов выбирает фильтр для случайного шумового сигнала.

– Close (Заккрыть)

Выход из демонстрационной программы фильтрации и повторный вызов резидентной программы монитора EZ-KIT Lite. Позволяет выбрать новую демонстрационную программу.

Echo cancellation

Эта программа демонстрирует эхоподавление в моделируемом эхоканале.

Кодек AD1847 преобразует в аналоговый сигнал с микрофонного входа в цифровое представление и пересылает данные в последовательный порт DSP. DSP генерирует эхосигнал и суммирует последовательный вход с выходом моделируемого эхоканала. Эхоканал состоит из линейной задержки, выполненной в виде длинного КИХ-фильтра с нулевыми значениями весовых коэффициентов и дисперсионного фильтра 16-го порядка. Выход сигнала с подавленным эхом передается к кодеку для реконструкции.

При запуске программы посылает на кодек отраженный сигнал без эхоподавления. Возможен следующий выбор:

– *Next (Следующий)*

При нажатии этой кнопки программа входит в режим эхоподавления. При этом будет ощутимое ослабление в эхосигнале. Дальнейшее нажатие кнопки будет изменять позицию дисперсионного фильтра и заставит фильтр адаптироваться к новому эхопути.

– *Close (Заккрыть)*

Выход из демонстрационной программы эхоподавления и повторный вызов резидентной программы монитора EZ-KIT Lite. Позволяет выбрать новую демонстрационную программу.

ADPCM

Эта программа демонстрирует возможности адаптивной дифференциальной импульсно-кодовой модуляции (ADPCM). ADPCM состоит из некоторого числа алгоритмов речевой компрессии реального времени. Для каждого периода выборки применяется линейный предсказывающий фильтр, чтобы генерировать предсказанный выход. Разница между предсказанным выходом и реальной величиной выборки посылается через коммуникационный канал. Так как динамический диапазон дифференциальной ошибки (разницы) существенно ниже, чем динамический диапазон речевого сигнала, получается меньшая результирующая скорость передачи разрядов.

Вначале преобразованный кодеком в цифровое представление микрофонный сигнал пересылается на выход кодека для восстановления. Возможен следующий выбор:

– *ADPCM coding (АДИКМ кодирование)*

Каждая оцифрованная выборка микрофонного сигнала кодируется и затем декодируется с использованием алгоритма ADPCM. После декодирования она посылается обратно на кодек для восстановления. Красный светодиод горит при выполнении кодирования.

– *Straight through (Прямое прохождение)*

Оцифрованные выборки микрофонного сигнала посылаются на выход кодека для восстановления. Красный светодиод в этом режиме будет выключен.

– *Close (Заккрыть)*

Выход из демонстрационной программы ADPCM и повторный вызов резидентной программы монитора EZ-KIT Lite. Позволяет выбрать новую демонстрационную программу.

7.8k LPC

Программа демонстрирует 7.8к кодирование методом линейного предсказания.

– *Interrupt button on EZ-KIT Lite (Кнопка вызова прерывания на EZ-KIT Lite)*

Нажмите кнопку вызова прерывания на EZ-KIT Lite, чтобы переключить режим работы между прохождением речи и кодированием 7.8k LPC. Красный светодиод будет гореть, когда действует кодирование LPC.

– Close (Заккрыть)

Выход из демонстрационной программы и повторный вызов резидентной программы монитора EZ-KIT Lite. Позволяет выбрать новую демонстрационную программу.

2.4k LPC

Программа демонстрирует 2.4к кодирование методом линейного предсказания.

– Interrupt button on EZ-KIT Lite (Кнопка вызова прерывания на EZ-KIT Lite)

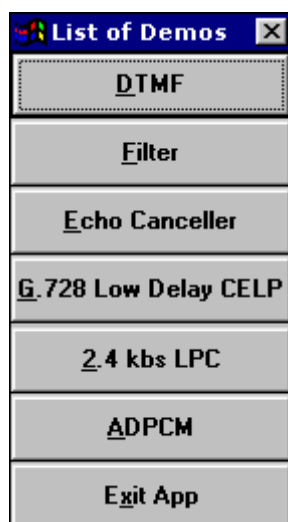
Нажмите кнопку вызова прерывания на EZ-KIT Lite, чтобы переключить режим работы между прохождением речи и кодированием 2.4k LPC. Красный светодиод будет гореть, когда действует кодирование LPC.

– Close (Заккрыть)

Выход из демонстрационной программы и повторный вызов резидентной программы монитора EZ-KIT Lite. Позволяет выбрать новую демонстрационную программу.

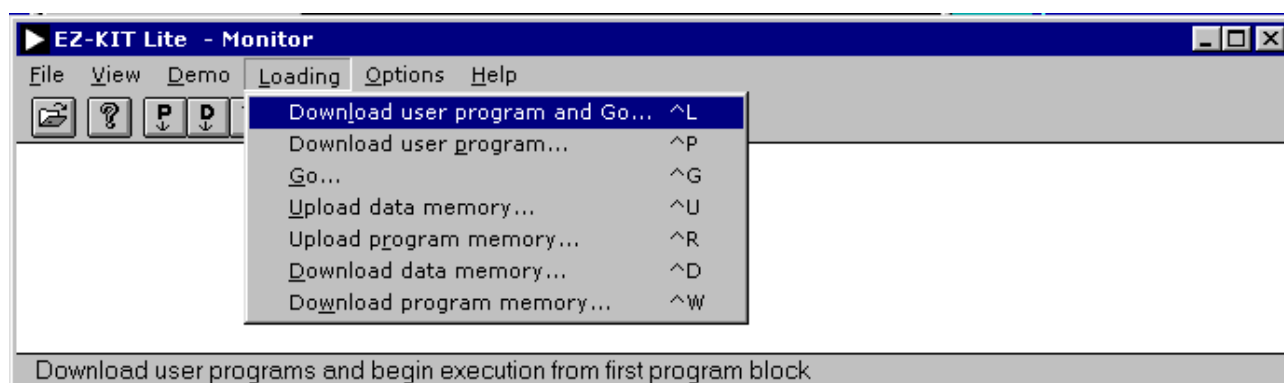
Всплывающее меню

Другим способом вызова всех демонстрационных программ является всплывающее меню «List of Demos». Это диалоговое окно может быть вызвано/спрятано через меню.



Меню «Loading»

Это меню выбирает варианты выгрузки и загрузки содержимого памяти, включая пользовательские программы.



Download User Program and Go (Загрузка пользовательской программы и запуск)

Выберите файл образа памяти (*.exe) для загрузки. При этом в область памяти, указанную в ядре файла образа памяти, будут загружены значения памяти программ и памяти данных, начиная с указанного адреса. После этого программа монитор передаст выполнение DSP заново загруженной программе.

Все пользовательские вектора прерываний будут записаны в специальный временный буфер, чтобы обеспечить функционирование программы монитора. Когда монитор приступит к выполнению команд пользовательской программы, сохраненные пользовательские вектора прерываний будут восстановлены из временного буфера в нужные ячейки памяти.

Прерывания недоступны перед началом исполнения пользовательской программы. Все пользовательские вектора маскируются с использованием инструкции DIS INTS. Значение маски дополнительно устанавливается в 0, поэтому таймер также и таймер недоступен.

Первый адрес памяти программ, вычисленный в файле образа, используется как входная точка в пользовательскую программу. Монитор выполняет вызов подпрограммы по этому адресу. Пользовательская программа должна выполнить инструкцию RTS инструкцию, чтобы вернуться к программе монитору.

С момента возвращения все вектора прерываний восстанавливаются для продолжения функций монитора. Можно также установить регистр BDMA, таким образом, чтобы смоделировать сброс по включению питания и вызвать перезагрузку программы монитор из микросхемы EPROM.

Программа монитор резервирует несколько адресов памяти. Пользовательские программы не могут использовать эту зарезервированную память.

Download User Program (Загрузка пользовательской программы)

Выберите файл образа памяти (*.exe) для загрузки. Разница между этой опцией и предыдущей состоит в том, что после загрузки не происходит передачи выполнения пользовательской программе.

После загрузки программы, монитор будет продолжать свою работу. Точка входа в пользовательскую программу сообщается в подтверждающем диалоговом окне. Выполнение пользовательской программы может быть начато с использованием команды «GO», описанной ниже.

Для выяснения других деталей, обратитесь к предыдущему разделу «Загрузка пользовательской программы и запуск».

GO (Запуск)

Введите шестнадцатеричный адрес точки входа в пользовательскую программу. Разрешенный диапазон адресов лежит в пределах от 0x0000 до 0x37FF включительно. Адреса памяти программ от 0x3800 до 0x3FFF используются для хранения инструкций программы монитора. Загрузка по этим адресам испортит код монитора.

Upload Data Memory (Выгрузка памяти данных)

Эта команда выгружает блок памяти данных DSP в файл образа памяти (*.exe), сохраняя его на управляющем ПК. При этом указываются шестнадцатеричный стартовый адрес и количество слов для выгрузки. Результирующий файл может быть обратно загружен в память данных по некоторому незапрещенному адресу. Файл может быть просмотрен посредством использования какого-либо текстового редактора.

При вводе неправильного шестнадцатеричного адреса или диапазона памяти, появится сообщение об ошибке. Проверьте, что количество выгружаемых слов, прибавленное к начальному адресу, не выходит за допустимые пределы.

Upload Program Memory (Выгрузка памяти программ)

Эта команда выгружает блок памяти программ DSP в файл образа памяти (*.exe), сохраняя его на управляющем ПК. При этом указываются шестнадцатеричный стартовый адрес и количество слов для выгрузки. Результирующий файл может быть обратно загружен в память программ по некоторому незапрещенному адресу. Файл может быть просмотрен посредством использования какого-либо текстового редактора.

При вводе неправильного шестнадцатеричного адреса или диапазона памяти, появится сообщение об ошибке. Проверьте, что количество выгружаемых слов, прибавленное к начальному адресу, не выходит за допустимые пределы.

Download Data Memory (Загрузка памяти данных)

Эта команда загружает файл образа памяти (*.exe) из управляющего компьютера в память данных DSP, начиная с указанного стартового адреса.

Загружаемый файл может быть файлом, созданным хост программой во время выгрузки памяти данных, или сгенерированным редактором связей ADSP-21xx. Так как файл, созданный хост программой, содержит только одну часть памяти (памяти данных), то будет загружено все содержимое файла. Редактор связей создает исполняемый файл, который может иметь несколько частей памяти программ и памяти данных. При этом будет загружена только первая часть памяти данных, а остальные будут игнорироваться.

Так как формат файла образа памяти – текстовый ASCII без проверки контрольной суммы, то можно использовать текстовый ASCII редактор для внесения изменений. Обратитесь к руководству пользователя ассемблера семейства ADSP-2100 и программе моделирования для уточнения формата файла. Однако так как формат файла довольно простой, путем экспериментов и проверок, вы можете модифицировать его и без обращения к руководству.

При вводе неправильного шестнадцатеричного адреса или диапазона памяти, появится сообщение об ошибке. Проверьте, что количество выгружаемых слов, прибавленное к начальному адресу, не выходит за допустимые пределы.

Не пытайтесь производить загрузку по адресам памяти, зарезервированные для программы монитора.

Download Program Memory (Загрузка памяти программ)

Эта команда загружает файл образа памяти (*.exe) из управляющего компьютера в память программ DSP, начиная с указанного стартового адреса.

Загружаемый файл может быть файлом, созданным хост программой во время выгрузки памяти программ, или сгенерированным редактором связей ADSP-21xx. Так как файл, созданный хост программой, содержит только одну часть памяти (памяти программ), то будет загружено все содержимое файла. Редактор связей создает исполняемый файл, который может иметь несколько частей памяти программ и памяти данных. При этом будет загружена только первая часть памяти программ, а остальные будут игнорироваться.

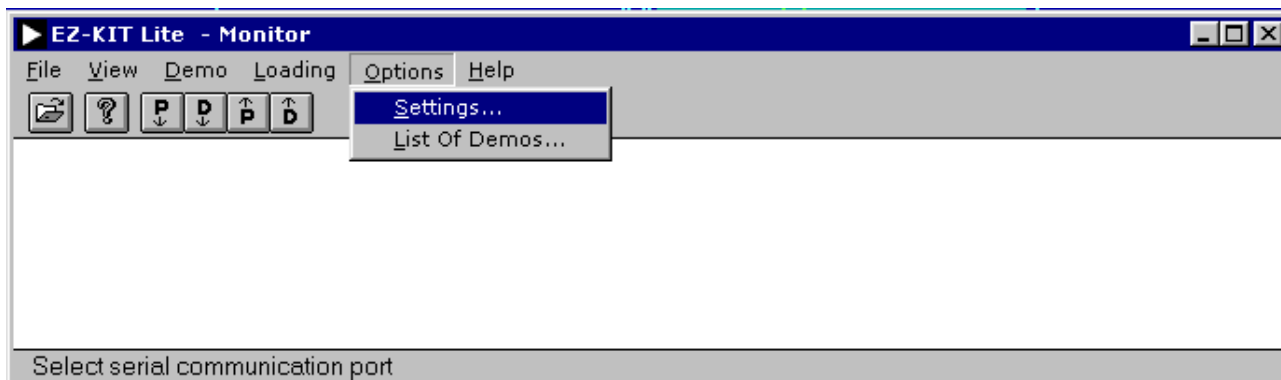
Так как формат файла образа памяти – текстовый ASCII без проверки контрольной суммы, то можно использовать текстовый ASCII редактор для внесения изменений. Обратитесь к руководству пользователя ассемблера семейства ADSP-2100 и программе моделирования для уточнения формата файла. Однако так как формат файла довольно простой, путем экспериментов и проверок, вы можете модифицировать его и без обращения к руководству.

При вводе неправильного шестнадцатеричного адреса или диапазона памяти, появится сообщение об ошибке. Проверьте, что количество выгружаемых слов, прибавленное к начальному адресу, не выходит за допустимые пределы.

Не пытайтесь производить загрузку по адресам памяти, зарезервированные для программы монитора.

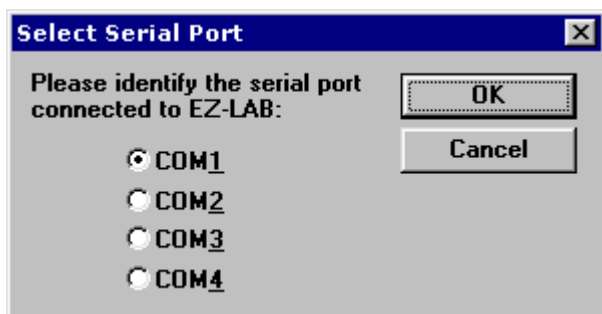
Меню «Options»

Этот меню определяет работу хост программы.



Settings (Установки)

При выборе этого пункта меню появляется следующее диалоговое окно:



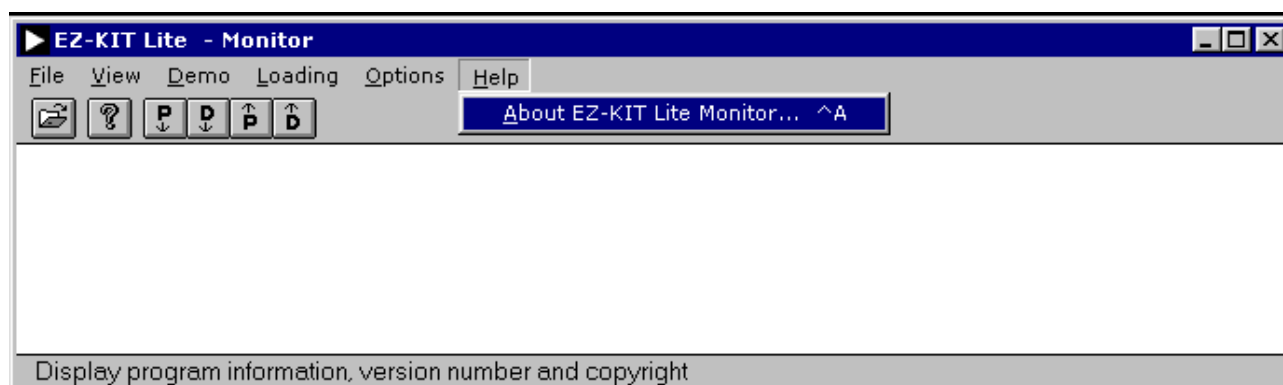
Выберите последовательный порт связи, к которому подключена плата EZ-KIT Lite. Конфигурация сохраняется в файле «ezkitapp.ini».

List of Demos (Список демонстрационных программ)

При выборе пункта появляется всплывающее меню для вызова демонстрационных программ.

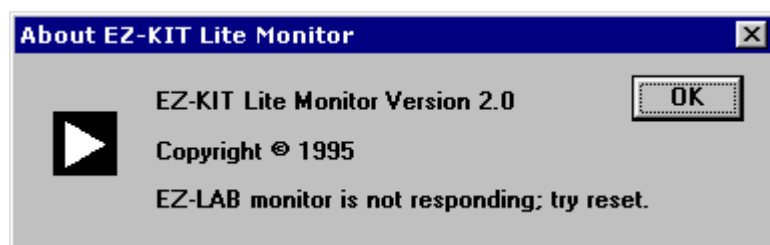


Меню «Help»



About EZ-KIT

При выборе этого пункта меню появляется следующее диалоговое окно:



В третьей строке содержится информация о состоянии соединения между ПК и платой. Если хост программа соединилась с программой монитора, то появится строка «EZ-KIT monitor is alive and well». В противном случае, будет выведено «EZ-KIT monitor is not running; try reset». Нажмите «OK», чтобы закрыть диалоговое окно.

Установки, настраиваемые пользователем

Некоторые аспекты работы хост программы могут быть изменены в файле «ezkitapp.ini». Некоторые автоматически устанавливаются хост программой. Если файл не найден, хост программа использует настройки по умолчанию и напомним пользователю о необходимости ввода номера последовательного порта связи.

- ◆ Port No = 2
Номер последовательного коммуникационного порта. Устанавливается хост программой и не должен изменяться вручную.
- ◆ Message TimeOut = 5
Продолжительность ожидания вывода диалогового окна (по умолчанию пять секунд). При установке в ноль, автоматический вывод отключается. Это единственная установка, требующая прямого вмешательства пользователя.
- ◆ ShowFloatingDialogBox = 1
При установке в «1» делает всплывающее меню видимым, когда хост программа начинает свою работу. Устанавливается хост программой и не должна изменяться.

Ошибки и неисправности

Существует несколько сообщений, говорящих об ошибках или неисправностях, выявленных при связи с программой монитором EZ-KIT Lite. Эти сообщения описаны ниже.

«Detected the following error(s):

**Memory mapped registers reset state.
DSP registers test.
Do loop test.
On-chip memory test.
Codec initialization.»**

Это сообщение говорит о том, что резидентный монитор обнаружил указанные ошибки во время процесса самотестирования после включения. Нажмите кнопку сброса на макетной плате EZ-KIT Lite и попробуйте повторить операцию снова.

«Time out waiting for EZ-LAB to complete operation»

Это сообщение говорит о том, что хост программа безуспешно прождала ответа от программы монитора. Возможная причина состоит в том, что память, зарезервированная под монитор, испорчена. Нажмите кнопку сброса на плате EZ-KIT Lite и попробуйте повторить операцию снова.

«Windows reported a communication error»

Это сообщение говорит о том, что модуль связи Windows сообщает об ошибке хост программы. Возможная причина - конфликт между Windows приложениями. Закройте все другие программы и попытайтесь повторить операцию снова. Следует также проверить, что выбранный последовательный порт связи не конфликтует с драйвером мыши.

«Checksum error»

Это сообщение говорит о том, что была обнаружена ошибка проверки контрольной суммы. При операции передачи данных контрольная сумма считается для каждого содержимого блока памяти, как монитором, так и базовой программой. Возможные причины – перегрузка линии или большая шумовая помеха. Если проблема не устраняется, постарайтесь укоротить длину кабеля связывающего ПК с платой. Проведите эксперимент с расположением платы EZ-KIT Lite и вашего компьютера.

«EZ-LAB's EPROM resident monitor not running. Try reset»

Это сообщение говорит о том, что хост программа не в состоянии установить правильное функционирование резидентной программы монитора EZ-KIT Lite. Возможная причина состоит в том, что хост программа наталкивается на проблемы с модулем связи Windows. Перезагрузите Windows и попытайтесь снова.

«Unknown error»

Это сообщение появляется при неопознанных ошибках. Возможная причина может состоять в потере исполняемых файлов демонстрационной программы, искажении файла или внутренней программной ошибке.

«EZ-LAB did not response to serial communication. Try reset»

Это сообщение говорит о том, что хост программа не может получить ответ от программы монитора. Возможная причина – плата EZ-KIT Lite не связана с выбранным последовательным портом, или другое Windows приложение использует выбранный последовательный порт. Закройте все другие Windows приложения и попробуйте снова.

«Attempted to write to reserved memories. Download not completed»

Это сообщение говорит о том, что пользовательский файл был загружен в содержимое памяти, инициализированные под резидентный монитор. Загрузка была прекращена, чтобы защитить зарезервированную память. Восстановите память с помощью инициализации и попробуйте повторить операцию снова.

Запуск демонстрационных программ (Running Demos)

Чтобы запустить демонстрационные программы, запустите хост программа под Windows. Обратитесь к разделу «Меню Демо» для получения детальной информации.

- DTMF
- Filtering
- Echo Cancellation
- ADPCM
- 7.8k LPC
- 2.4k LPC

Вы можете выбрать демонстрационную программу, нажав на одну из кнопок всплывающего меню или выбрав пункт из меню «Демо». После того, как демонстрационная программа загружена на плату EZ-KIT Lite, она начинает выполняться.

Создание ваших собственных программ

Управляющая программа, запущенная на ПК, связывается через последовательный порт с платой EZ-KIT Lite. Команды, посылаемые хост программой на плату EZ-KIT Lite, и ответы, принимаемые базовой программой, описаны в этой главе. EZ-KIT Lite понимает следующие последовательные команды:

0. Веер

принимает: \$\$\$

посылает: ok [led count][alive called count][selftest hi][selftest lo]

[led count] младшие 8 разрядов счетчика, увеличивающегося со скоростью 28800 в секунду. Задержка счета показывает, что прерывание таймера возможно не работает.

[alive called count] увеличивается на единицу каждый раз, когда эта вызывается эта команда или команда *alive*.

[selftest hi][selftest lo] результат самотестирования.

На аудио выход будет послан короткий сигнал.

1. Alive inquiry

принимает: \$OK

посылает: ok [led count][alive called count][selftest hi][selftest lo]

[led count] младшие 8 разрядов счетчика, увеличивающегося со скоростью 28800 в секунду. Задержка счета показывает, что прерывание таймера возможно не работает.

[alive called count] увеличивается на единицу каждый раз, когда эта вызывается эта команда или команда *alive*.

[selftest hi][selftest lo] результат самотестирования.

Подобна команде Веер, но сигнал на аудио выходе не генерируется.

2. upload data memory content (выгрузить содержимое памяти данных)

принимает: \$UD [hi start][lo start][hi len][lo len]

посылает: [hi][lo] [hi sum][lo sum]\$!

[hi start][lo start] стартовый адрес в памяти.

[hi len][lo len] количество слов.

[hi][lo] содержимое памяти, начиная со стартового адреса.

[hi sum][lo sum] 16-разрядное значение контрольной суммы данных.

3. upload program memory content (выгрузить содержимое памяти программ)

принимает: \$UP [*hi start*][*lo start*][*hi len*][*lo len*]

посылает: [*hi*][*mi*][*lo*] [*hi sum*][*mi sum*][*lo sum*]\$!

[*hi start*][*lo start*] стартовый адрес в памяти.

[*hi len*][*lo len*] количество слов.

[*hi*][*mi*][*lo*] содержимое памяти, начиная со стартового адреса.

[*hi sum*][*mi sum*][*lo sum*] 24-разрядное значение контрольной суммы данных.

4. download data memory content (загрузить содержимое памяти данных)

принимает: \$DD [*hi start*][*lo start*][*hi len*][*lo len*][*hi*][*lo*]

посылает: [*hi sum*][*lo sum*]

[*hi start*][*lo start*] стартовый адрес в памяти.

[*hi len*][*lo len*] количество слов.

[*hi*][*lo*] содержимое памяти, начиная со стартового адреса.

[*hi sum*][*lo sum*] 16-разрядное значение контрольной суммы данных.

5. download program memory content (загрузить содержимое памяти программ)

принимает: \$DP [*hi start*][*lo start*][*hi len*][*lo len*][*hi*][*mi*][*lo*]

посылает: [*hi sum*][*mi sum*][*lo sum*]

[*hi start*][*lo start*] стартовый адрес в памяти.

[*hi len*][*lo len*] количество слов.

[*hi*][*lo*] содержимое памяти, начиная со стартового адреса.

[*hi sum*][*mi sum*][*lo sum*] 24-разрядное значение контрольной суммы данных.

6. Begin user program execution (начать выполнение программы пользователя)

принимает: \$GO [*hi address*][*lo address*]

посылает: [*hi address*][*lo address*]

[*hi address*][*lo address*] точка входа в программу, посланная хост программой программе монитору. Для подтверждения приема адрес посылается программой монитора обратно хост программе.

Программа монитор EZ-KIT Lite



7



Обзор

Программа монитор находится в памяти микросхемы EPROM. Она автоматически загружается при перезапуске во встроенную память программ и память данных. После загрузки программа монитор начинает выполняться, производя самотестирование регистров DSP во встроенной памяти, и перезапускает и инициализирует кодек AD1847. После этого программа ждет команд через последовательный порт связи. Подпрограмма «общения» кодека работает и во время исполнения подпрограммы прерывания.

Возможности программы монитора

Программа монитор обладает следующими возможностями:

- ◆ Самотестирование при включении
- ◆ Инициализация кодека и связь с ним
- ◆ Эмуляция UART со скоростью 9600 бод
- ◆ Выгрузка и загрузка содержимого памяти через последовательный порт

Ограничения

Чтобы свободного выполнения программы монитора следует учесть некоторые ограничения на все пользовательские программы. Среди них есть строгие ограничения, которые могут привести к нарушению процесса загрузки пользовательских программ:

- ◆ Зарезервированная область памяти от 0x3800 до 0x3FFF.
Здесь располагается программа монитор. Любое нарушение приведет к разрушению программы монитора в результате незаконченной загрузки программы.
- ◆ Зарезервированная область памяти от 0x3E00 до 0x3FDF.
Здесь располагаются переменные программы монитора. Пользовательские программы, использующие эти места памяти будут накладываться на переменные монитора; в результате программа пользователя не загрузится полностью.
- ◆ Возврат в монитор
После завершения загрузки пользовательской программы, программное обеспечение монитора вызовет ее как подпрограмму. Пользовательская программа должна заканчиваться инструкцией RTS, чтобы должным образом вернуться в монитор.

Для пользовательских программ, которые не возвращаются в программу монитор, не существует никаких ограничений после завершения процесса загрузки. Другими словами, в то время как файл образа памяти не должен пытаться инициализировать зарезервированную память, пользовательская программа, тем не менее, свободна использовать эту память после начала своего выполнения.

Создание программ, используемых вместе с монитором

Для создания таких программ можно изменить один из примеров, включенных в комплект программного обеспечения EZ-KIT Lite, или написать собственную программу, используя включенный инструмент разработки. Для получения дополнительной информации обратитесь к главе 5. Существует несколько моментов, которые следует иметь в виду при разработке программ, используемых с монитором.

Пользовательские программы могут использовать всю незарезервированную память программ и незарезервированную память данных. Это адреса с 0x0000 до 0x37FF для памяти программ и адреса с 0x0000 до 0x3DFF для памяти данных.

- ◆ Точка входа в программу
Пользовательские программы, загружаемые хост программой, должны иметь свою точку входа в начале ядра памяти программ. Это достигается присваиванием точке входа программы первой инструкции в модуле и указанием этого модуля первым в процессе компоновки. В большинстве случаев программа будет стартовать с адреса 0x0000. Для этого можно применить директиву ассемблера `.MODULE/RAM/ABS=0`.
- ◆ Разрешение прерываний
Перед передачей процесса выполнения пользовательской программе, все прерывания маскируются (`IMASK=0`). Это делается для того, чтобы пользовательская программа могла безопасно выйти из процесса инициализации перед собственным разрешением прерываний.
- ◆ Векторы прерываний
Пользовательская программа должна инициализировать все используемые вектора прерываний. При этом в программе не может использоваться прерывание `IRQ1`, так как оно предназначено для управления связью через `RS-232`.
- ◆ Команды монитора
Исходный код программы монитора содержится в каталоге `21XX\EZKITL\MONITOR`. Разрешается изменять этот код в соответствии с вашими потребностями. Если вы приступите к редактированию программы монитора, то вам потребуется доступ к его рабочим командам.

Монитор может отвечать через последовательный порт приведенным ниже командам. Они могут использоваться при сопряжении EZ-KIT Lite с другой, нежели базовой программой.

0. Имя команды:

принимает: обозначение кода и данных, принятых от хост программы. Символы, не заключенные в скобки принимаются непосредственно, заключенные в скобки – в порядке от старшего байта к младшему.

посылает: обозначение кода и данных, отправленных хост программе. Символы, не заключенные в скобки принимаются непосредственно, заключенные в скобки – в порядке от старшего байта к младшему.

1. Веер

принимает: \$\$\$

посылает: ok [*led count*][*alive called count*][*selftest hi*][*selftest lo*]

[*led count*] младшие 8 разрядов счетчика, увеличивающегося со скоростью 28800 в секунду. Задержка счета показывает, что прерывание таймера возможно не работает.

[*alive called count*] увеличивается на единицу каждый раз, когда эта вызывается эта команда или команда *alive*.

[*selftest hi*][*selftest lo*] результат самотестирования.

На аудио выход будет послан короткий сигнал.

2. Alive inquiry

принимает: \$OK

посылает: ok [*led count*][*alive called count*][*selftest hi*][*selftest lo*]

[*led count*] младшие 8 разрядов счетчика, увеличивающегося со скоростью 28800 в секунду. Задержка счета показывает, что прерывание таймера возможно не работает.

[*alive called count*] увеличивается на единицу каждый раз, когда эта вызывается эта команда или команда *alive*.

[*selftest hi*][*selftest lo*] результат самотестирования.

Подобна команде Веер, но сигнал на аудио выходе не генерируется.

3. upload data memory content (выгрузить содержимое памяти данных)

принимает: \$UD [*hi start*][*lo start*][*hi len*][*lo len*]

посылает: [*hi*][*lo*] [*hi sum*][*lo sum*]\$!

[*hi start*][*lo start*] стартовый адрес в памяти.

[*hi len*][*lo len*] количество слов.

[*hi*][*lo*] содержимое памяти, начиная со стартового адреса.

[*hi sum*][*lo sum*] 16-разрядное значение контрольной суммы данных.

4. **upload program memory content (выгрузить содержимое памяти программ)**

принимает: \$UP [*hi start*][*lo start*][*hi len*][*lo len*]

посылает: [*hi*][*mi*][*lo*] [*hi sum*][*mi sum*][*lo sum*]\$!

[*hi start*][*lo start*] стартовый адрес в памяти.

[*hi len*][*lo len*] количество слов.

[*hi*][*mi*][*lo*] содержимое памяти, начиная со стартового адреса.

[*hi sum*][*mi sum*][*lo sum*] 24-разрядное значение контрольной суммы данных.

5. **download data memory content (загрузить содержимое памяти данных)**

принимает: \$DD [*hi start*][*lo start*][*hi len*][*lo len*][*hi*][*lo*]

посылает: [*hi sum*][*lo sum*]

[*hi start*][*lo start*] стартовый адрес в памяти.

[*hi len*][*lo len*] количество слов.

[*hi*][*lo*] содержимое памяти, начиная со стартового адреса.

[*hi sum*][*lo sum*] 16-разрядное значение контрольной суммы данных.

6. **download program memory content (загрузить содержимое памяти программ)**

принимает: \$DP [*hi start*][*lo start*][*hi len*][*lo len*][*hi*][*mi*][*lo*]

посылает: [*hi sum*][*mi sum*][*lo sum*]

[*hi start*][*lo start*] стартовый адрес в памяти.

[*hi len*][*lo len*] количество слов.

[*hi*][*lo*] содержимое памяти, начиная со стартового адреса.

[*hi sum*][*mi sum*][*lo sum*] 24-разрядное значение контрольной суммы данных.

7. **Begin user program execution (начать выполнение программы пользователя)**

принимает: \$GO [*hi address*][*lo address*]

посылает: [*hi address*][*lo address*]

[*hi address*][*lo address*] точка входа в программу, посланная хост программой программе монитору. Для подтверждения приема адрес посылается программой монитора обратно хост программе.

Отладка

В процессе отладки пользовательских программ могут быть полезны следующие инструкции:

- ◆ SET FL1
- ◆ RESET FL1
- ◆ TOGGLE FL1

Состояние светодиода FL1 непосредственно управляется флагом DSP FL1. Приведенные инструкции могут быть размещены в любом месте пользовательской программы, и использоваться для визуального отображения состояния некоторых переменных, или отображения выполнения какой-либо секции.

Для некритичных к времени программ, возможно получать некоторую информацию, программируя FL1 на мерцание с различной скоростью. Например, следующая программа включает мерцание светодиода с некоторой скоростью.

```
        cntr = 80;
        do lp1 until ce;
            cntr = 50;
            do lp2 until ce;
                cntr = 12000;
                do lp3 until ce;
lp3:                    nop;
lp2:                    nop;
lp1:                    toggle fl1;
```

Память DSP

Пользовательские программы могут хранить некоторые данные в памяти программ или в памяти данных, которые могут быть восстановлены после того, как пользовательские программы вернут управление обратно программе монитору (посредством инструкции RTS).

Для этого необходимо использовать команды выгрузки памяти программ или памяти данных.

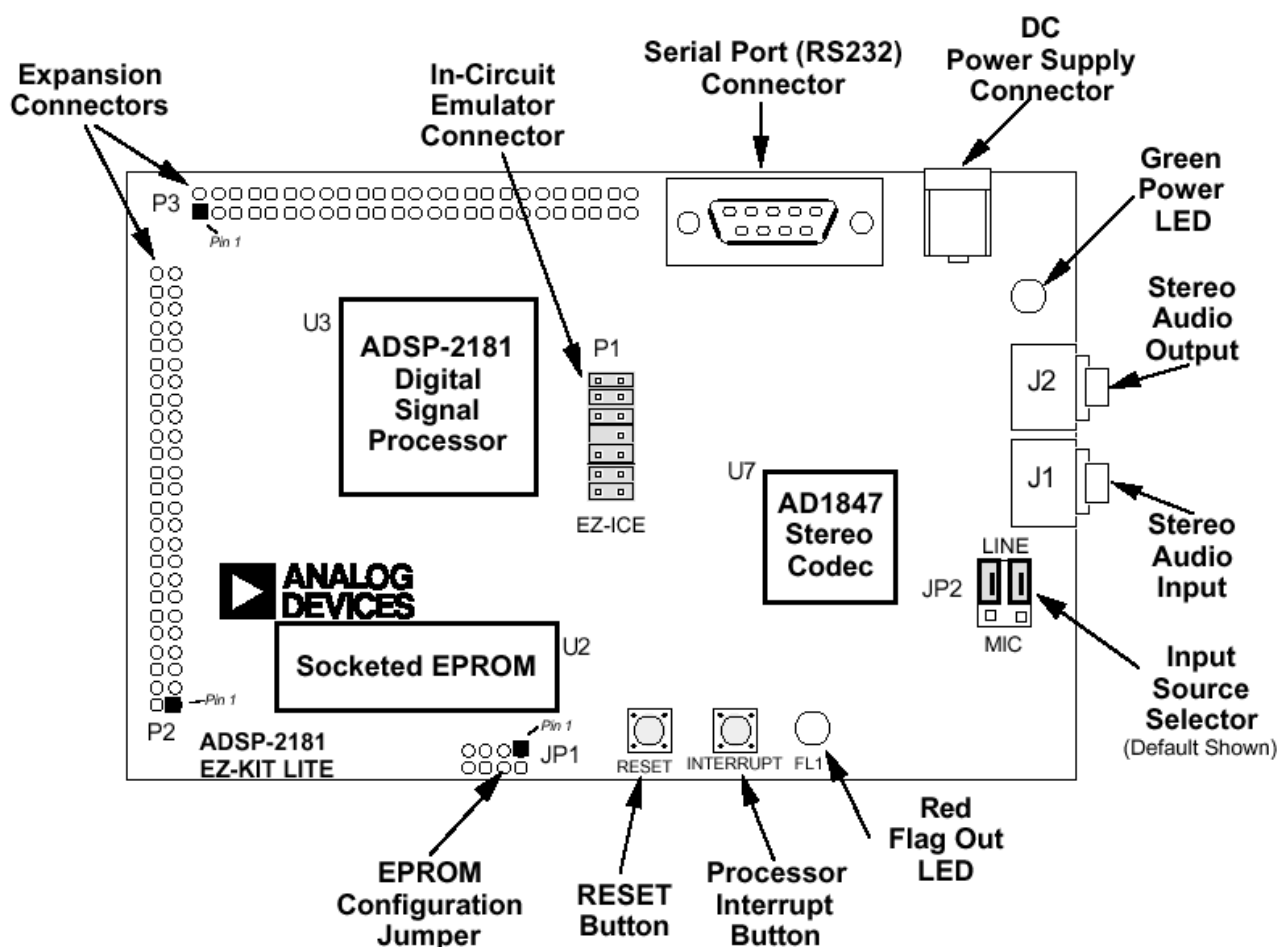
Описание аппаратной части EZ-KIT Lite



8

Обзор проекта

Аппаратная часть состоит из смонтированной печатной платы с размерами 8.9 x 14 см. На печатной плате расположены цифровой сигнальный процессор ADSP-2181, микросхема ППЗУ, кодек AD1847, а также различные схемы поддержки и соединители. Плата представляет собой завершённую систему обработки сигналов, спроектированную для демонстрации возможностей цифрового сигнального процессора ADSP-2181. Кроме того, можно использовать её в качестве платформы для разработки новых приложений для ADSP-2181.



EZ-KIT Lite является примером минимального оборудования для процессора ADSP-2181. Микросхема EPROM (ППЗУ) соединена с процессором через байтовый DMA порт (ПДП – канал прямого доступа к памяти).

Эта связь использует для переноса данных только 8 из 24 линий (D8 – D15). Восемь других линий (D16 – D23) используются для предоставления дополнительных разрядов шины адреса. Это позволяет ADSP-2181 обращаться к адресному пространству размером 32 Мбит (4 Мб). Встроенная на плате кроватка предназначена для присоединения микросхем EPROM с объемом памяти от 256 Кбит до 8 Мбит (наибольшая из доступных). JP1 обеспечивает настройку функций выводов 3 и 30 кроватки по требованию различных микросхем EPROM. DSP настраивается для запуска из EPROM при перезапуске.

Кодек AD1847 подключен к DSP через SPORT0. Этот высокоскоростной синхронный последовательный порт передает все данные, управляющие сигналы и информацию о состоянии между DSP и кодеком. Существует возможность отключить кодек, если последовательный порт используется для других целей. Сигнал CODECDIS/ на соединителе P3 позволяет отключить кодек. При подаче на этот вывод сигнал низкого уровня, кодек отключается, и его сигналы переводятся в высокоимпедансное состояние.

Выводы SPORT1 используются для связи с управляющим ПК через интерфейс RS-232 (U5). Выводы «Flag In» и «Flag Out» используются для обозначения сигнала переноса по приему и передаче данных. Прием данных вызывает прерывание IRQ1, таким образом, DSP может обнаруживать активность без запросов на выводе «Flag In». Программный запуск DSP эмулирует протокол UART для обеспечения правильной асинхронной последовательной связи на скорости 9600 бит/сек.

U1 является логическим устройством, состоящим из шести инверторов. Два из них используются для реализации сброса по включению и противодребезговой защиты кнопки сброса (RESET). Следующие два инвертора используются противодребезговой защиты кнопки прерывания (INTERRUPT). Пятый инвертор используют для управления красным светодиодом (D1), поскольку вывод FL1 не может обеспечивать достаточную нагрузку по току напрямую.

Порт IDMA не используется. Все IDMA сигналы на плате EZ-KIT Lite доступны на соединителе P3.

Спецификации

| | |
|-----------------------|---|
| Процессор: | ADSP-2181KS-133 работающий с частотой 33 МГц (16.667 МГц внешней синхронизации) |
| Аналоговый интерфейс: | AD1847 стерео кодек |
| Аналоговые входы: | Одна стереопара с уровнем сигнала 2 В эффективного значения переменного тока, связанная с линейными входами Одна стереопара с уровнем сигнала 20 мВ эффективного значения переменного тока, связанная с микрофонными входами |
| Аналоговые выходы: | Одна стереопара с уровнем сигнала 1 В эффективного значения переменного тока, связанная с линейными выходами |
| Мощность: | от 8 до 10 В постоянного тока @ 300 мА |
| Окружающая среда: | от 0 до 70 °C, от 10% до 90% относительной влажности |

Соединители

J1 – стерео гнездо с диаметром 1/8 дюйма (3.5 мм). Этот разъем используется для ввода линейных аудио сигналов или микрофонных звуковых сигналов.

J2 – стерео гнездо с диаметром 1/8 дюйма (3.5 мм). Этот разъем используют для вывода линейных аудио сигналов.

J3 – это 9-выводной D-SUB разъем штепсельного типа. Он используется для связи с управляющим ПК, используя сигналы RS-232 и асинхронный последовательный протокол.

J4 – это 5.5 мм цилиндрический штекер. Он используется для подачи питания на плату. Центральный вывод штекера имеет диаметр 2 мм и должен соединять отрицательный вывод источника питания. Внешняя оплетка изолированного штекера должна быть положительной.

JP1 – площадка многоконтактного соединителя на восьми выводов. Используется для конфигурирования платы под объем ППЗУ, отличный от 1 Мбит (128 Кб) микросхемы EPROM (27C010), поставляемой с макетной платой. Большинству пользователей эта возможность не потребуется.

JP2 – площадка многоконтактного соединителя на шесть выводов. Используется для конфигурирования входного гнезда (J1), переключая режим работы с линейного на микрофонный вход. Центральный вывод в каждой из трех групп соединен с одним из выводов «Line 1 Input» кодека AD1847. Навесные проводники (еще известные как шунты или переключки) используются для связи этих выводов или с выходом микрофонного усилителя, или с выходом фильтра линейного входа.

P1 – является 14-выводным соединителем, используемым для включения внутрисхемного эмулятора ADSP-2181 EZ-ICE . Для правильного соединения вывод 7 должен быть отключен.

P2 и P3 – являются площадками для 50-выводных соединителей. Могут использоваться для доступа к сигналам ADSP-2181 в целях расширения или тестирования.

U2 – кроватка для микросхемы EPROM в DIP исполнении. Плата позволяет подключить 27C512 (64 Кб) или 27C010 (128 Кб) EPROM. Изменение подключений JP1 позволяет плате подключить 27C256 (32 Кб), 27C020 (256 Кб), 27C040 (512 Кб) или 27C080 (1 Мб) EPROM. Эта кроватка связана с ADSP-2181 по интерфейсу памяти с байтовой организацией.

R28 – позиция для резистора с нулевым сопротивлением. Если этот резистор установлен, ADSP-2181 может перезапускать плату под программным управлением. Программа вызывает перезапуск установкой PF0 на выход и перевода его на низкий уровень.

R29 – другая позиция для резистора с нулевым сопротивлением. Если этот резистор установлен, и отключены выводы X3 и C37, кодек может работать без сигнала процессора CLKOUT, используя собственную частоту синхронизации 24.576 МГц. Необходимо также изменить значение X1 на меньшую частоту, чтобы установить ее в пределах скорости кодека.

Кнопки

S1 – кнопка перезапуска (RESET). Нажатие этой кнопки заставляет процессор ADSP-2181 и кодек AD1847 перейти в состояние аппаратного сброса и оставаться в нем до того, пока кнопка не будет отпущена. Выход кнопки имеет противодребезговую электронную защиту, предотвращающую множественные переходы из-за дребезга механических контактов.

S2 - кнопка прерывания (INTERRUPT). Нажатие этой кнопки заставляет процессор вызвать прерывание IRQE. Это может заставить процессор выполнить обработчику прерывания, если оно разрешено и указан вектор обработчика прерывания IRQE. Выход кнопки имеет противодребезговую электронную защиту, предотвращающую множественные переходы из-за дребезга механических контактов.

Индикаторы

D1 – красный светодиод, который управляется выводом FL1 процессора ADSP-2181. Программное обеспечение может управлять состоянием этого индикатора записью в соответствующий внутренний регистр.

D2 – зеленый светодиод, который горит, если плата подключена к источнику питания.

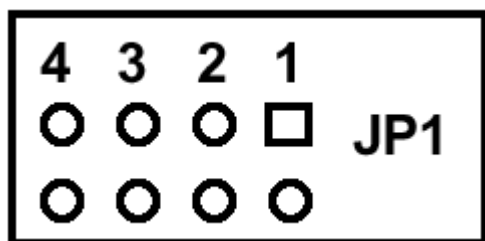
Работа аппаратной части

Когда к плате подключено питание, цепь сброса держит процессор в течение 30 мс. После этого процессор начинает операцию загрузки. Выводы BMODE и MMAP на ADSP-2181 заземлены, поэтому процессор загружается по интерфейсу памяти с байтовой организации, связанным с криваткой микросхемы EPROM. Если установлена микросхема EPROM, поставляемая с макетной платой, то работа оборудования будет происходить, как описано в главе о программном обеспечении.

Расширение аппаратной части

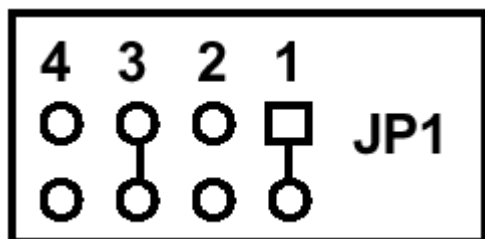
Конфигурирование платы для различных микросхем EPROM

JP1 позволяет конфигурировать плату ADSP-2181 EZ-KIT Lite для одного из шести различных объемов EPROM. Поставляемая плата позволяет подключать микросхемы 27C512 или 27C010. Если устанавливаются EPROM другого объема, потребуется изменить соединения на J1.

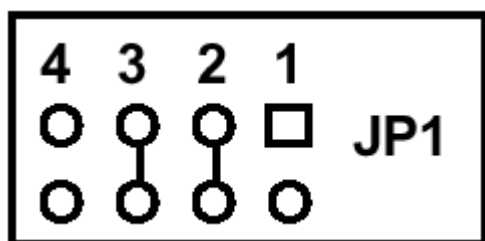


Соединения делаются вертикально между контактными площадками. Пара контактных площадок под номером представляет позицию переключки под этим номером. Соединения могут быть выполнены несколькими способами. Если предполагаемое изменение объема не происходит, это может быть вызвано дополнительной связью через припой между контактными площадками.

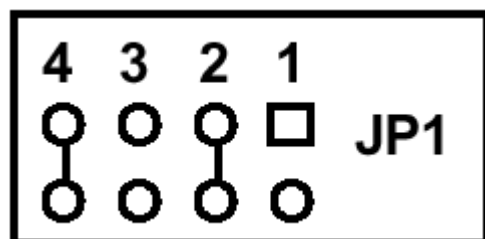
Для 27C256 EPROM соединения должны быть следующими.



Для 27C512 или 27C010 EPROM соединения должны быть следующими.

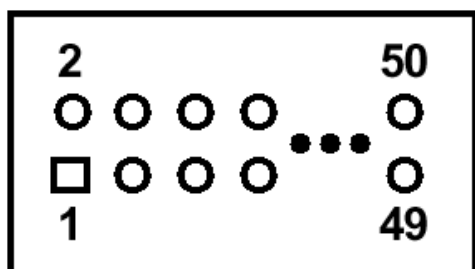


Для 27C020, 27C040 или 27C080 EPROM соединения должны быть следующими.



Соединители расширения

50-выводные позиции P2 и P3 обеспечивают доступ к сигналам ADSP-2181 в целях расширения или тестирования. Номера выводов на этом соединителе расположены как:



Сигналы доступные на этих выводах следующие.

| Номер вывода P2 | Сигнал | Номер вывода P2 | Сигнал |
|-----------------|--------|-----------------|--------|
| 1 | A0 | 2 | A1 |
| 3 | A2 | 4 | A3 |
| 5 | A4 | 6 | A5 |
| 7 | A6 | 8 | A7 |
| 9 | A8 | 10 | A9 |
| 11 | A10 | 12 | A11 |
| 13 | A12 | 14 | A13 |
| 15 | D0 | 16 | D1 |
| 17 | D2 | 18 | D3 |
| 19 | D4 | 20 | D5 |
| 21 | D6 | 22 | D7 |
| 23 | D8 | 24 | D9 |
| 25 | D10 | 26 | D11 |
| 27 | D12 | 28 | D13 |
| 29 | D14 | 30 | D15 |
| 31 | D16 | 32 | D17 |
| 33 | D18 | 34 | D19 |
| 35 | D20 | 36 | D21 |
| 37 | D22 | 38 | D23 |
| 39 | WR/ | 40 | RD/ |
| 41 | IOMS/ | 42 | BMS/ |
| 43 | DMS/ | 44 | CMS/ |
| 45 | PMS/ | 46 | BR/ |
| 47 | BGH/ | 48 | BG/ |
| 49 | VCC | 50 | GND |

| Номер вывода P3 | Сигнал | Номер вывода P3 | Сигнал |
|-----------------|-----------|-----------------|--------|
| 1 | GND | 2 | IAD0 |
| 3 | IAD1 | 4 | IAD2 |
| 5 | IAD3 | 6 | IAD4 |
| 7 | IAD5 | 8 | IAD6 |
| 9 | IAD7 | 10 | IAD8 |
| 11 | IAD9 | 12 | IAD10 |
| 13 | IAD11 | 14 | IAD12 |
| 15 | IAD13 | 16 | IAD14 |
| 17 | IAD15 | 18 | GND |
| 19 | IACK/ | 20 | IAL |
| 21 | IS/ | 22 | IWR/ |
| 23 | IRD/ | 24 | GND |
| 25 | PF0 | 26 | PF1 |
| 27 | PF2 | 28 | PF3 |
| 29 | PF4 | 30 | PF5 |
| 31 | PF6 | 32 | PF7 |
| 33 | FL0 | 34 | FL1 |
| 35 | FL2 | 36 | CLKOUT |
| 37 | RESET/ | 38 | IRQL0/ |
| 39 | IRQL1 | 40 | IRQ2/ |
| 41 | PWD/ | 42 | PWDACK |
| 43 | CODECDIS/ | 44 | TXD0 |
| 45 | TFS0 | 46 | RFS0 |
| 47 | RXD0 | 48 | SCK0 |
| 49 | VCC | 50 | GND |

Функции этих сигналов приведены в документации на ADSP-2181 за исключением сигнала CODECDIS/. Этот вывод может быть заземлен для того, чтобы выключить кодек AD1847, чтобы использовать последовательный порт в других целях.

Отладка аппаратной части

Если зеленый светодиод не горит, проверьте подключение источника питания. Убедитесь в том, что источник питания имеет необходимый размер разъема и правильную полярность. Результат измерения напряжения на контактах платы должен быть в пределах от 8 В до 10 В выпрямленного напряжения. Проверьте, что нет объектов около платы, которые могут быть причиной короткого замыкания цепи.

Если контакт источника питания хороший и зеленый светодиод включен, но красный светодиод не горит, и отсутствует звуковой сигнал, убедитесь, что микросхема EPROM хорошо вставлена в разъем.

Нажмите кнопку сброса, если плата работает неправильно.

