

Пауль Бекманн, Драгос Давидеску, Джон Кнаб (STMicroelectronics)

РАЗРАБОТКА ВЫСОКОКАЧЕСТВЕННЫХ АУДИОПРИЛОЖЕНИЙ ДЛЯ ПОТРЕБИТЕЛЬСКОЙ ЭЛЕКТРОНИКИ

Обеспечить высокое качество звука в портативных аудиосистемах способен микроконтроллер из линейки **STM32F4**. Помимо этого, микроконтроллер поможет добиться ряда специальных звуковых эффектов и организовать параллельные потоки высокоскоростного обмена аудиоданными. А специальный мощный программный инструмент **Audio Weaver** поможет разработчику сократить время на создание сложных алгоритмов обработки аудиосигнала.

Наряду со снижением цен на высокопроизводительные микроконтроллеры во многих типах мобильных устройств производители стараются добавить возможности цифрового аудио. Одной из их основных целей является поддержка как можно более широкого спектра источников медиаданных — мобильных устройств типа коммуникаторов, плееров, внешних USB-устройств, SD-карты, Интернет источники.

При всей кажущейся легкости задачи обеспечения высококачественного звука, для портативных компактных устройств она является отнюдь не тривиальной. Качество звука зависит от множества факторов, в частности, от конфигурации конечной системы, что затрудняет разработку, т.к. отличия в конструкции макета или прототипа от финальной версии могут сыграть в сторону снижения качества звучания. Дополнительной проблемой для разработчиков аудиосистем на основе микроконтроллеров является реализация алгоритмов цифровой обработки сигналов в реальном времени, что включает в себя:

- реализацию специализированных фильтров и алгоритмов обработки звука;
- отслеживание корректности выполнения операций арифметики с плавающей точкой;
- использование инструкций, подобных инструкциям цифровой обработки сигналов (ЦОС);
- оптимизация алгоритмов с точки зрения производительности, времени выполнения, использования памяти, процессорного времени и потребляемой мощности.

В данной статье будет рассмотрено, каким образом разработчики могут ис-

пользовать возможности ЦОС и блока арифметики с плавающей точкой для качественного воспроизведения звука, реализации аудио-алгоритмов, преобразований данных с различными частотами дискретизации, осуществления высокоскоростного обмена данными без потери производительности, оптимизации процесса разработки с точки зрения качества и стоимости разработки. А также выполнения других задач, таких как обслуживание пользовательского интерфейса, в том числе графического, силами одного контроллера.

Прикладные задачи при воспроизведении звука

Обычно дополнительные возможности воспроизведения и записи звука во встраиваемых приложениях требуют цифровой обработки сигналов в совокупности с возможностями управляющего контроллера общего назначения. Даже такое устройство, как переносные портативные колонки для цифровых плееров типа iPod, для качественного воспроизведения звука требует реализации достаточно большого числа алгоритмов ЦОС.

Пространственное разнесение: В переносных колонках расстояние между динамиками правого и левого каналов небольшое — обычно 30...40 сантиметров. Для обеспечения пространственных эффектов требуется алгоритмическая компенсация близкого физического расположения динамиков.

Многоканальные аудиосистемы: Для систем, поддерживающих более двух динамиков, требуется формировать сигналы для дополнительных аудиоканалов из входного стереосигнала.

Эквалайзер: Для улучшения качества звука на конкретных динамиках

или для учета персональных предпочтений не лишней будет функция эквалайзера. Более того, при смене динамиков должна обеспечиваться подстройка заданных частотных характеристик. В конечном устройстве разработчик может заложить несколько методов подстройки, включая графическую подстройку, параметрические настройки. Для более сложных устройств разработчик может реализовать собственные методы коррекции, используя различные инструменты для их разработки (например, MATLAB).

Ограничение пиков: Как правило, АЧХ динамиков нелинейна, и на некоторых частотах возможно увеличение громкости звука при одном и том же уровне входного сигнала. Используя зависящее от времени усиление и контроль уровня сигнала, возможно воспроизвести звук на более высокой громкости при минимальных искажениях.

Подъем уровня: При прослушивании музыки на низком уровне звука многие детали звучания, например, глубина, могут быть потеряны. Подъем уровня низких частот (басов) и некоторых выбранных частот с использованием компенсации громкости или управления звуком в зависимости от восприятия могут существенно улучшить субъективное качество звука.

Поддержание определенного уровня звука: Поддержание определенного уровня звука особенно важно для пользователей при непрерывном прослушивании отдельных песен или альбомов для устранения необходимости постоянного регулирования звука для поддержания комфортного уровня.

Производительность DSP или универсальность?

В основном цифровая обработка звука в пользовательских встраиваемых устройствах реализуется при помощи выделенного специализированного процессора. Однако, в свете высокой конкуренции на данном секторе рынка, разработчики вынуждены стремиться к снижению стоимости устройств за счет снижения количества элементов, уменьшения времени разработки программно-

го обеспечения, расширения набора дополнительных функций.

В общем случае реализовать алгоритмы обработки аудиосигналов на контроллере общего назначения несколько проще, чем реализовывать функции управления реальным временем, интерфейсные и коммуникационные функции на цифровом сигнальном процессоре (DSP). DSP, при всей своей производительности, не имеют столь богатой периферии и гибкой системы обслуживания прерываний, требуемых для систем реального времени. Архитектура DSP адаптирована для мощных высокопроизводительных систем с высоким уровнем параллелизма данных, что превышает требования типичных пользовательских устройств (высокая избыточность вычислительных ресурсов, повышенное энергопотребление, высокая цена). В дополнение к этому, DSP чаще всего не поддерживают простые коммуникационные интерфейсы типа USB, интерфейса к SD-картам, WiFi и т.п. Таким образом, для портативных аудиоустройств с DSP в качестве центрального процессора потребуется дополнительный коммуникационный процессор, обеспечивающий интерфейсные функции.

Справедливости ради следует отметить появление на рынке гибридных систем-на-кристалле, включающих в себя микроконтроллер общего назначения (чаще всего RISC- или ARM-архитектуры) и цифровой сигнальный процессор, но это устройства несколько иной ценовой категории и уровня потребления энергии, рассчитанные, как правило, на аппаратуру более высокого класса.

Расширение набора инструкций микроконтроллеров за счет DSP-инструкций позволяет реализовывать

на контроллерах не только базовые алгоритмы обработки сигналов, но и более сложные и комплексные алгоритмы, позволяющие улучшить качество звука или реализовать дополнительные эффекты. Типичной ситуацией при разработке программного обеспечения для DSP является значительный объем низкоуровневого кода (ассемблерного кода), особенно при оптимизации вычислений. Микроконтроллеры общего назначения находятся в более выигрышной позиции, так как при прочих равных условиях имеется широкий выбор компиляторов языков высокого уровня, например, языка C/C++, что, конечно же упрощает освоение нового типа контроллеров, ускоряет процесс разработки программного обеспечения, и, как следствие, способствует более быстрому выходу продукта на рынок. Кроме того, разработчикам доступен широкий выбор программных библиотек, распространяемых и сообществами программистов, и фирмами-разработчиками программного обеспечения.

Также контроллеры общего назначения обладают более развитой периферией, архитектурно специализированы на возможность обработки прерываний от множества источников, как внешних, так и внутренних.

Архитектура микроконтроллеров STM32 для аудиоприложений

Архитектура контроллеров 32-рядных микроконтроллеров STM32F4 от STMicroelectronics позволяет разработчикам использовать микроконтроллер общего назначения в таких устройствах как мобильные портативные колонки, мультимедийные проигрыватели, наушники, гарнитуры и тому подобные устройства для воспроизведения звука. В основе STM32F4 лежит процес-

сорное ядро ARM Cortex-M4F, способное работать на частотах до 168 МГц, и объединяющее в себе преимущества богатой системы команд контроллера, расширенной DSP-инструкциями с однократным исполнением таких операций как умножение, умножение с накоплением, SIMD-команды; поддержки целочисленной арифметики с насыщением и модуля с плавающей точкой (FPU). Все это в совокупности с широким выбором периферийных устройств и интерфейсов с высокопроизводительной шинной архитектурой микроконтроллеров (рисунок 1) позволяет реализовывать мощные алгоритмы обработки аудиосигнала, гарантируя высокое качество записи или воспроизведения звука при низкой цене самого контроллера.

Микроконтроллеры линейки STM32F4 демонстрируют отличную производительность на аудиоприложениях, вполне сравнимую с производительностью специализированных аудио-процессоров или DSP. Так на рисунке 2 представлены типичные требования к производительности процессоров различных типов для декодирования и воспроизведения MP3-аудиопотока.

Богатый набор периферийных устройств позволяет реализовывать высокопроизводительные встраиваемые аудиоприложения с набором коммуникационных и интерфейсных возможностей — подключение различных источников звука, прием-передача цифровых аудиопотоков, возможность легкого подключения внешних устройств.

Ключевыми возможностями микроконтроллеров линейки STM32F4 для увеличения производительности аудио-приложений и снижения стоимости конечной системы являются следующие:

Поддержка набора DSP-инструкций: Дополнительно к набору базовых инструкций, ядро ARM-Cortex-M4 поддерживает 105 специализированных инструкций цифровой обработки сигналов — так называемые DSP-инструкции. Они включают в себя:

- однократные MAC-инструкции (умножение с накоплением — умножение двух операндов с последующим сложением с предыдущим результатом);
- инструкции арифметики с насыщением — предотвращение выхода результата за границы диапазона допустимых значений;
- SIMD-инструкции над 8- и 16-битными данными.

Это позволяет разработчикам эффективно использовать возможности цифровой обработки сигналов во встраиваемых системах при лучшем соотношении цена-производительность, чем для DSP.

Блок вычислений с плавающей точкой: Все контроллеры линейки STM32F4 содержат модуль вычислений в форма-

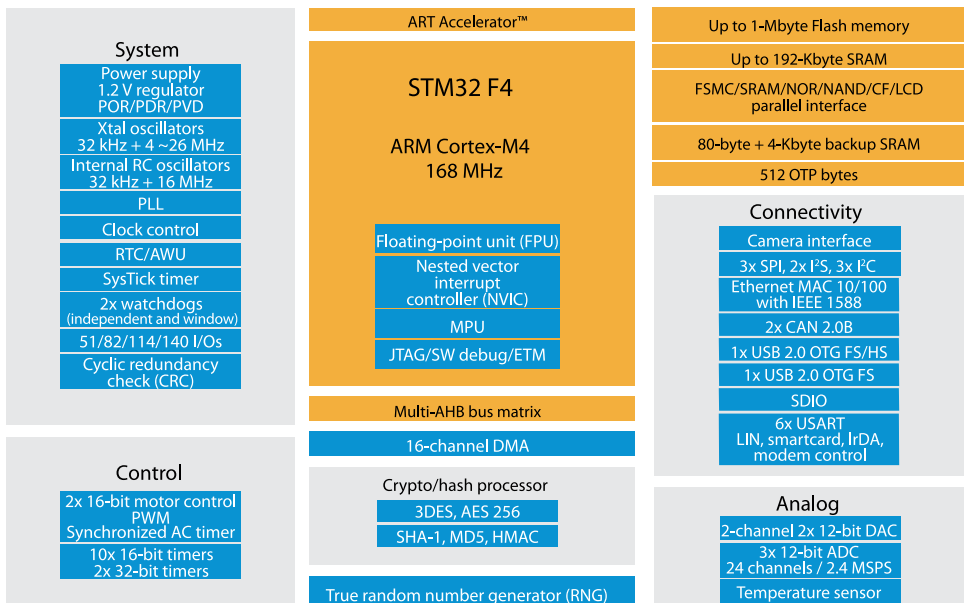


Рис. 1. Структура микроконтроллеров семейства STM32F4

Таблица 1. Сравнение времени выполнения некоторых базовых операций ЦОС ядрами Cortex-M3 и Cortex-M4

Операция	Время выполнения, такты	
	Cortex-M3	Cortex-M4
32-битное умножение (32 x 32 = 32)	1	1
32-битный MAC – 32-битное умножение с последующим сложением или вычитанием (32 ± (32 x 32) = 32)	2	1
32-битное умножение с 64-битным результатом (32 x 32 = 64)	5...7	1
32-битное умножение с последующим сложением с 64-битным результатом (32 x 32) + 64 = 64	5...7	1

те с плавающей точкой (FPU). Наличие аппаратного модуля вещественных вычислений существенно ускоряет вычисления в формате с плавающей точкой (в среднем на порядок) что позволяет практически без потери производительности реализовывать алгоритмы ЦОС с вещественными данными. В свою очередь, работа с данными в формате с плавающей точкой дает ряд преимуществ, основным из которых является большой динамический диапазон значений – удобная работа с очень маленькими или очень большими значениями без потери точности. Широкий динамический диапазон значений позволяет практически без ограничений применять улучшенные алгоритмы DSP (например, фильтры более высокого порядка), а также алгоритмы обработки звука, доступные ранее только на ПК или профессиональном оборудовании. В ряде случаев варианты алгоритмов, работающие с данными в формате с плавающей точкой, имеют преимущество в размере кода, потребляемой памяти и времени выполнения, чем в варианте фиксированной точки.

Эффективность 32-битной архитектуры: Разрядность шин данных процессора имеет огромное влияние на его производительность и энергоэффективность. Даже если поток отсчетов аудиосигнала 16-битный, обрабатывающей его системе все равно требуется 32 бита для хранения промежуточных результатов. 16-битному микроконтроллеру или сигнальному процессору требуется порядка семи операций (четыре операции умножения и три – сложения) для выполнения умножения 32-битных чисел. Ядро Cortex-M4 позволяет STM32F4 выполнять MAC-инструкции за один цикл (таблица 1).

Многоуровневая высокопроизводительная системная шина: Один из ключевых моментов обработки сигналов в реальном времени – эффективное управление потоками данных. В аудиоустройствах контроллер должен не только обрабатывать поток данных, управлять памятью, коммуникационными портами и другими системными задачами. Специфика аудиоалгоритмов заключается в том, что они должны

быть тесно связаны с пользовательским приложением и работой системных задач – даже незначительная задержка или неравномерность звука при воспроизведении будет заметна. Вместе с тем, в аудиоустройствах обычно присутствуют несколько параллельных потоков данных, часть из которых требует обработки, часть является вспомогательными или данными параллельно выполняющихся задач.

Архитектура STM32 разработана для минимизации подобных проблем, существенно сокращая время разработчика на разрешение потенциальных конфликтов доступа к данным. Достигается это при помощи развитой системы обработки прерываний в совокупности с многослойной системной шиной, позволяющей осуществление нескольких параллельных транзакций прямого доступа в память без участия процессорного ядра. На рисунке 3 показан высокий уровень параллелизма, который может обеспечить системная шина.

Программный код, работающий с данными в оперативной памяти, исполняется из flash-памяти (красная линия).

Поток сжатых аудиоданных поступает через USB-интерфейс и сохраняется в оперативной памяти. Процессорное ядро получает доступ к аудиопотоку

для декомпрессии и обработки сигнала (зеленые линии).

Поток декодированных MP3-данных поступает от процессора в оперативную память (желтая линия).

Аудиоданные выдаются на I2S (оранжевые линии), графические данные передаются из флеш-памяти на дисплей (синие линии) посредством DMA-контроллера.

Коммуникационные интерфейсы: Пользователю желательно иметь возможность воспроизводить аудио, полученные из различных источников через различные коммуникационные интерфейсы. С набором наиболее популярных и часто используемых интерфейсов, включая USB (с режимами хоста и конечного устройства), Ethernet, SDIO (подключение SD карт) и интерфейсами к внешним носителям данных разработчики могут создавать многофункциональные устройства со многими вариантами использования.

В дополнение к возможности получения данных без загрузки процессора разработчикам желательно также отслеживать поток аудиоданных для обработки ситуации потери пакетов данных и осуществления управления скоростью потока.

К примеру, возможности обратной связи для предотвращения исчерпания

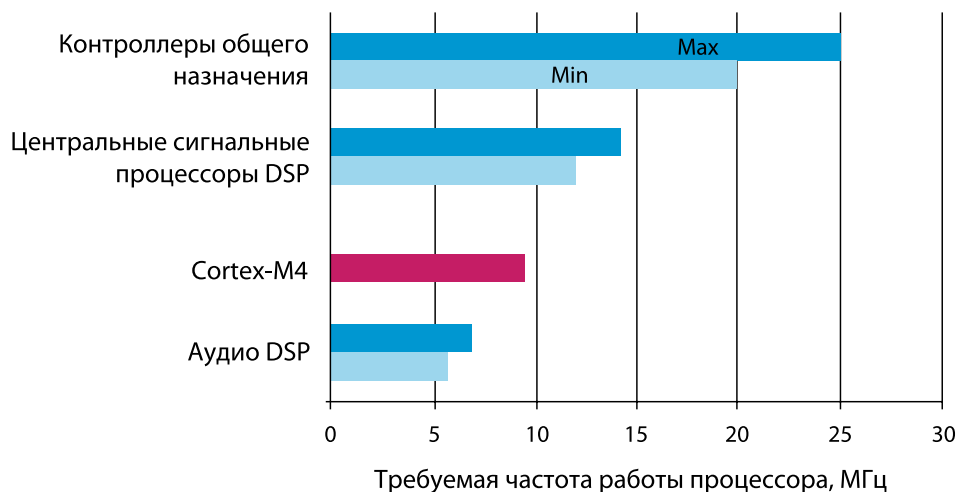


Рис. 2. Средние требования к производительности процессоров различных типов для декодирования воспроизведения MP3-аудио

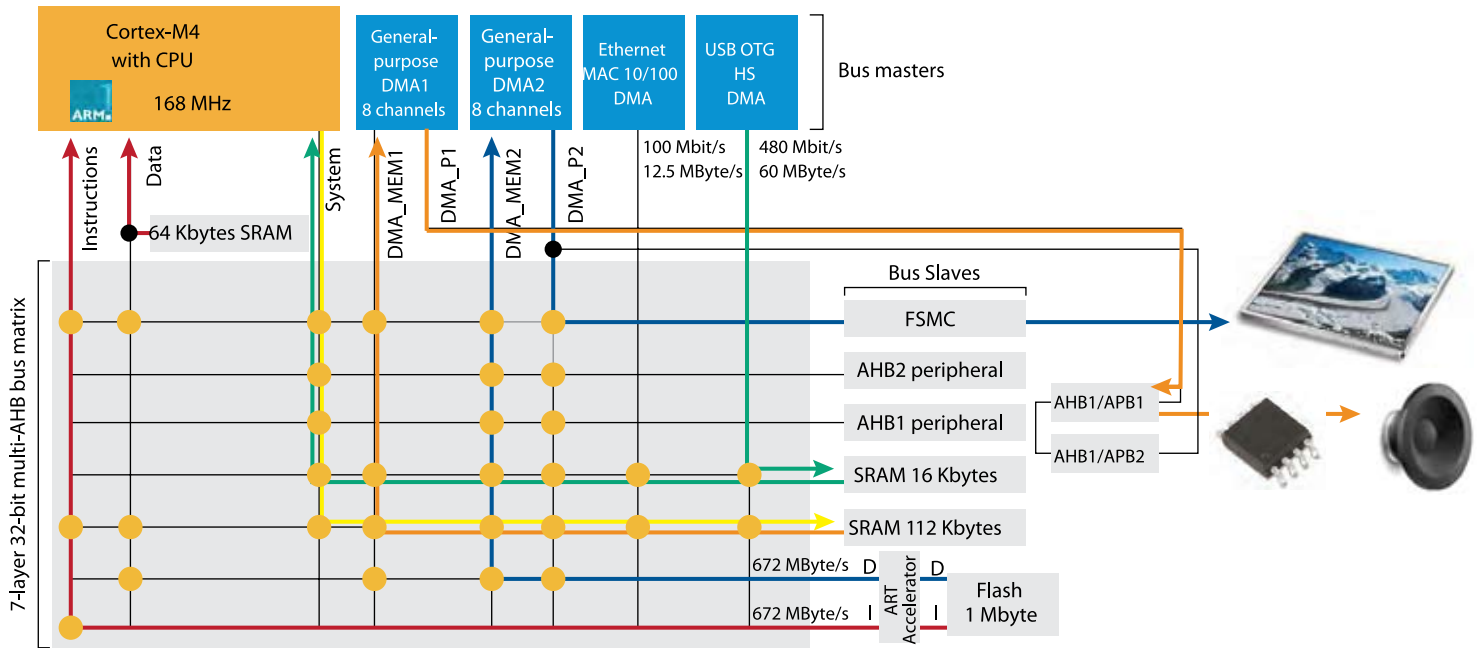


Рис. 3. Демонстрация возможности поддержки нескольких параллельных потоков данных многослойной шиной STM32F4

или переполнения приемного буфера, заложенные в USB, не всегда используются или реализованы не в полном размере в ряде устройств. Это может привести к потере или отбрасыванию пакетов аудио данных, что, в свою очередь, может негативно сказаться на качестве звука.

Для решения данных проблем разработчики могут использовать механизм преобразования частоты отсчетов (sample-rate conversion – SRC). SRC также часто используется при микшировании аудиопотоков, поступающих от разных источников, для преобразования между различными скоростями аудиопотоков (например в случае разности частот дискретизации записанного аудиопотока и рабочих частот воспроизводящей аппаратуры) или для компенсации небольших разностей частот. Реализация SRC на STM32F4 дает загрузку процессора примерно процентов на 10%, оставляя ресурсы для других алгоритмов.

Несколько источников тактирования: Цифровому аудио устройству даже в самом простом варианте необходимо уметь поддерживать и генерировать несколько тактовых частот: кроме, собственно, системной тактовой частоты, чаще всего необходимо тактирование

интерфейсов типа USB или I²S, имеющих определенные стандартизованные частоты с регламентированными отклонениями от номинальных значений. Попытка использовать один источник для формирования трех и более частот, скорее всего, приведет к потере качества и исходного сигнала, и формируемых тактовых последовательностей; кроме того, это возможно для кратных частот. Так, вполне реально на базе одного тактового генератора напрямую получить тактовую частоту 168 МГц для процессорного ядра, 44,1 кГц для I²S интерфейса или 48 МГц для USB, но не все три частоты сразу.

Для расширения возможностей микроконтроллера при формировании тактовых последовательностей для интегрированных периферийных устройств и тактирования внешних ведомых устройств в состав STM32F4 входят два множителя частоты (PLL).

Основной PLL используется для формирования системной тактовой частоты, второй – для формирования тактовых частот, использующихся в аудиоприложениях. Это дает возможность одновременной работы и аудиоинтерфейсов, и коммуникационных интерфейсов (USB), и, как следствие – возможность

одновременного воспроизведения звука и обмена данными.

Встроенные аудиоинтерфейсы: STM32F4 имеет два встроенных полнодуплексных стереоинтерфейса I²S (точность поддержания частоты отсчетов лучше 0,5%). Есть также возможность внешнего тактирования интерфейсов, если по каким-либо соображениям это является более предпочтительным. Наличие встроенных I²S-интерфейсов уменьшает число и общую стоимость компонентов, экономит площадь печатной платы.

Периферийные устройства: Контроллеры STM32 включают все необходимые периферийные устройства, требующиеся для основных классов задач, решаемых при помощи микроконтроллеров.

Совокупность перечисленных возможностей делает контроллеры линейки STM32F4 прекрасным выбором для портативных аудиоустройств. Это касается и работы с периферийными устройствами, и вычислительной мощности. В таблице 2 представлено сравнение производительности контроллеров линеек STM32F2 и STM32F4 при выполнении распространенных аудиоалгоритмов. Применение DSP-инструкций дает

Таблица 2. Сравнение производительности контроллеров линеек STM32F2 и STM32F4

Алгоритм	Загрузка процессора, %		Объем флеш-памяти	Объем требуемой оперативной памяти
	STM32F2	STM32F4		
Декодирование MP3	17	6	23K	12344
MP3-кодирование	22,5	9	25K	16060
Декодирование WMA	17,5	6	45K	36076
Декодер AAC+ v2	25	11	54K	87000
Микширование каналов	2,5	2	0,6K	16
Параметрический эквалайзер	16	12	2K	300
Контроль уровня громкости	4,5	3,5	3,25K	632
SRC	22,5	10	17,5K	1880

STM32F4 выиграш в среднем от 25% до 70%.

Дополнительным эффектом от снижения загрузки процессора является возможность перехода системы в состояние низкого энергопотребления после менее продолжительных периодов активной работы. Результат — снижение общего энергопотребления и более длительное время автономной работы без перезарядки или смены источников питания.

С STM32F4 разработчикам становятся доступны преимущества библиотеки CMSIS DSP, включающей в себя большое число алгоритмов DSP в вариантах вычислений с фиксированной и с плавающей точкой. Библиотека CMSIS DSP поддерживается компанией ARM и является независимой от конкретного производителя библиотекой алгоритмов для процессоров серий Cortex-M.

Разработка алгоритмов обработки аудиоданных — инструментарий Audio Weaver

Автором библиотеки CMSIS DSP является компания DSP Concept. Опыт разработки данной библиотеки был использован компанией при разработке мощного инструментария для разработки алгоритмов обработки сигналов Audio Weaver.

Audio Weaver позволяет разработчикам существенно сократить время на разработку и отладку алгоритмов обработки звука, начиная от получения отсчетов аудио сигнала до его выхода (воспроизведения).

Audio Weaver является средой визуального программирования, позволяющей разработчику планировать обработку потока сигналов путем выбора и соединения необходимых функциональных блоков (рисунок 4).

За каждым функциональным блоком стоит оптимизированный блок кода, реализующий данный алгоритм или этап алгоритма. Необходимые для взаимодействия блоков структуры данных генерируются автоматически. В результате на выходе получается эффективный, высокопроизводительный код, практически не имеющий потерь производительности на всех своих этапах. Это дает ощутимый выигрыш во времени и трудоемкости разработки программного обеспечения по сравнению с построением программного кода «с нуля», так как в последнем случае неизменно потребуются время на проверку и оптимизацию каждого из этапов основного алгоритма, плюс проектирование структур данных и проверка корректности передачи данных и параметров между отдельными этапами. Кроме этого, необходимо отслеживать качество звука, получающееся на выходе алгоритма.

С Audio Weaver цикл разработки протекает гораздо быстрее, что дает воз-

можность разработчикам за то же время протестировать и оценить большее число вариантов, таким образом повышая качество звука при снижении затрат на разработку.

Код функциональных модулей оптимизирован по скорости и использованию памяти, поддерживаются процессоры с блоком вычислений с плавающей точкой (такие как STM32F4). Для программ, созданных при помощи Audio Weaver, не требуется наличие RTOS при выполнении на микроконтроллере — соответственно, снижаются требования к ресурсам контроллера, возможно снижение энергопотребления устройства в целом. В состав библиотеки функциональных модулей входят более 150 блоков, включая IP-блоки сторонних производителей.

С помощью инструментов, подобных Audio Weaver, благодаря технологии drag-and-drop и визуальному представлению, к процессу проектирования алгоритмов обработки звука могут быть привлечены специалисты из других предметных областей, не являющиеся программистами. Для начинающих разработчиков в комплекте поставки доступны примеры алгоритмов, применимых для различных ситуаций. В случае необходимости можно также воспользоваться технической поддержкой DSP Concepts. Все это может помочь начинающим компаниям при выходе на рынок аудиоустройств.

Опытные разработчики могут напрямую работать с функциональными блоками, объединяя их в собственные, более комплексные алгоритмы. Audio Weaver не только предоставляет среду исполнения с компонентами, позволяющими ускорить построение алгоритмов, но и предлагает среду разработки с возможностью быстрого прототипирования и настройки. С этой точки зрения для опытных компаний-разработчиков (и для отдельных разработчиков) использование Audio Weaver — один из путей сокращения времени выхода продукта на рынок.

Ускорение оптимизации алгоритмов с Audio Weaver

Для ускорения процесса Audio Weaver поддерживает возможности кроссплатформенной разработки. Среда позволяет выполнять спроектированные алгоритмы для STM32 на персональном компьютере, что дает инженерам прекрасную возможность разрабатывать и оптимизировать программное обеспечение параллельно с разработкой аппаратного решения для конечного устройства. Данная возможность позволяет сократить общее время на разработку продукта (разработка аппаратной и программной частей могут вестись параллельно), а также выявить и исправить ошибки проектирования на ранних этапах развития проекта.

По мере готовности целевого устройства разработанный алгоритм компилируется и оптимизируется уже для целевой платформы. При финальной оптимизации кода разработчику предоставляется возможность настройки каждого функционального блока или этапа алгоритма с целью оптимизации по скорости выполнения, размеру занимаемой памяти. В итоге это дает возможность разработчикам оценить во что, в терминах загрузки процессора, памяти и периферийных устройств, может вылиться требуемое качество звука, и допустимо ли это для выбранной конфигурации системы.

Самым простым примером, наверное, может стать выбор порядка выходного фильтра сигнала для подачи его на динамики. Фильтр низкого порядка может иметь отклонение от идеальной желаемой характеристики на величину порядка 3 дБ, в то время как при помощи фильтра более высокого порядка можно добиться отклонения 1 дБ и менее.

Знание разницы в загрузке процессора при реализации данных фильтров может быть учтено при разработке самого алгоритма обработки звука, например, при настройке и оптимизации других этапов.

Качество звука в конечном итоге определяется не формальными характе-

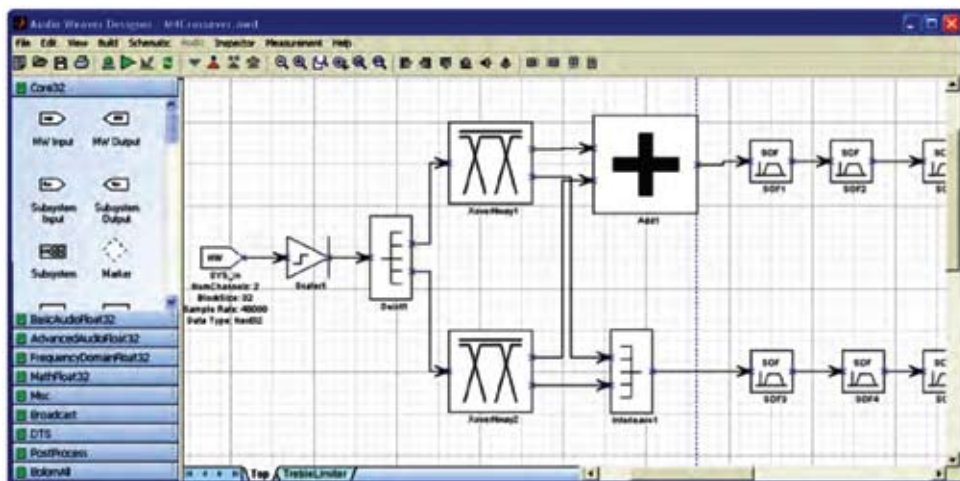


Рис. 4. Внешний вид рабочей области Audio Weaver при создании алгоритма

ристикami типа АЧХ, ФЧХ, импульсных характеристик, а восприятием этого звука человеком — не всегда идеальные характеристики соответствуют идеальному звуку с точки зрения слушателя. В этой связи на этапах разработки алгоритмов желательно иметь возможность услышать результат воздействия на звук того или иного этапа его обработки, не говоря уже о конечном результате.

Во многих средах разработки для этого необходимо после внесения изменений в проект его перекомпилировать, загрузить в конечное устройство. На это в любом случае требуется время, а для оценки влияния на звук фильтров низкого и высокого порядков человеку желательно прослушать оба варианта один за другим, возможно даже несколько раз. Audio Weaver решает данную проблему поддерживая интерфейс настройки, позволяющего в режиме реального времени изменить параметры фильтра (при желании — не только фильтра). Возможность задания конфигурации и переключения между вариантами буквально нажатием одной кнопки поможет разработчику сравнить оба варианта. При этом интерфейс настройки прост в использовании и прозрачен для программиста, что позволяет сосредоточиться на решении основной задачи.

Возможность подстройки параметров «на лету», без перекомпилирования всего проекта, также дает сокращение времени разработки и делает её более эффективной, что справедливо даже для начинающих разработчиков.

Не стоит также забывать о том, что аудиоприложение — это не только совокупность алгоритмов обработки аудиопотоков. Неотъемлемой его частью является взаимодействие с пользователем, планирование вычислительной работы, взаимодействие с периферийным оборудованием. Помимо аудиоалгоритмов, DSP Concepts предлагает широкий набор функциональных блоков с дополнительными возможностями:

- ядро реального времени;
- управление аудиовводом-выводом;
- управление интерфейсами взаимодействия с HOST-системой или ПК;
- начальный загрузчик;
- менеджер обновлений программного обеспечения;
- поддержка файловой системы во flash-памяти.

Системный подход к разработке аудиоприложений от STMicroelectronics

Одной из проблем при разработке встраиваемых аудиоприложений является то, что многие производители программного и аппаратного обеспечения, предоставляя примеры типовых систем на базе их продукции, часто обходят вниманием аудиоприложения.

STMicroelectronics приложила существенные усилия для устранения этого недостатка и создала ряд ресурсов с опорными решениями, оптимизированных для архитектуры STM32, и инструментарием для задач обработки звука. К примеру, ST и компании-партнеры предлагают множество оценочных плат с возможностями ввода-вывода звука. STMicroelectronics также предлагает несколько примеров реализации переносных колонок, способные служить основой для новых разработок.

Под лицензией Apple “Made for iPod” (MFI) STMicroelectronics предлагает ряд решений Apple iAP, основанных на контроллерах линеек STM32F2, STM32F4.

Apple iAP представляют собой аудиоаксессуары для устройств iPod, iPhone, iPad. Предлагаемые компоненты включают:

- оценочные платы STM32xG-EVAL или STM324xG-EVAL, к которым пользователь подключает средства авторизации Apple Authentication Coprocessor (ACP);
- протокол «iPod Accessory Protocol» с Lingoes для аутентификации, управления и получения информации о данных;
- библиотеку USB Host Library с классом HID-устройств.

Для устройств, работающих с потоками аудиоданных Apple iAP, поддерживает:

- классы USB устройств USB Host, USB Audio;
- удаленное управление iPod/iPhone/iPAD;
- цифровые аудиопотоки;
- извлечение тэгов;
- работу с флеш-картами.

Заключение

Современные бытовые аудиоустройства являются сложными комплексными системами, требующими высокой производительности для поддержания высокого качества звука и гибкости для противостояния быстро изменяющимся требованиям рынка. Имея высокопроизводительное процессорное ядро, эффективную шинную систему, допускающую существование нескольких параллельных потоков, богатый набор периферийных устройств, STM32F4 является прекрасным решением для многих типов встраиваемых и бытовых аудиоприложений. STM32F4 позволяет на базе одного микроконтроллера строить системы, обеспечивающие высокое качество звука при одновременной поддержке многочисленных пользовательских задач и удобной конфигурации коммуникационных интерфейсов.

Получение технической информации, заказ образцов, поставка — e-mail: mcu.vesti@compel.ru



МЭМС-микрофоны



- ✓ MP34DB01
- ✓ MP34DT01
- ✓ MP45DT02

- Один источник напряжения (1,64...3,6 В)
- Точка акустического насыщения 120 дБ (SPL)
- Чувствительность -26 дБFS (децибел полной шкалы)
- SNR @ 1 кГц / 1Па = 62 дБ

- Низкое потребление 0,65 мА (активный режим)
- Выход данных в формате PMD
- Диапазон рабочей температуры: -30(-40)...85°C
- Корпуса RHLGA, HLGA, HCLGA

Поддержка разработчиков:
 E-mail: st@compel.ru
www.compel.ru/projects-support



Компэл
www.compel.ru