

Синтез музыки

MIDI (англ. *Musical Instrument Digital Interface* — цифровой интерфейс музыкальных инструментов) — стандарт цифровой звукозаписи на формат обмена данными между электронными музыкальными инструментами.

Интерфейс позволяет единообразно кодировать в цифровой форме такие данные как нажатие клавиш, настройку громкости и других акустических параметров, выбор тембра, темпа, тональности и др., с точной привязкой во времени. В системе кодировок присутствует множество свободных команд, которые производители, программисты и пользователи могут использовать по своему усмотрению. Поэтому интерфейс MIDI позволяет, помимо исполнения музыки, синхронизировать управление другим оборудованием, например, осветительным, пиротехническим и т. п.

Последовательность MIDI-команд может быть записана на любой цифровой носитель в виде файла, передана по любым каналам связи. Воспроизводящее устройство или программа называется *синтезатором (секвенсором) MIDI* и фактически является автоматическим музыкальным инструментом.

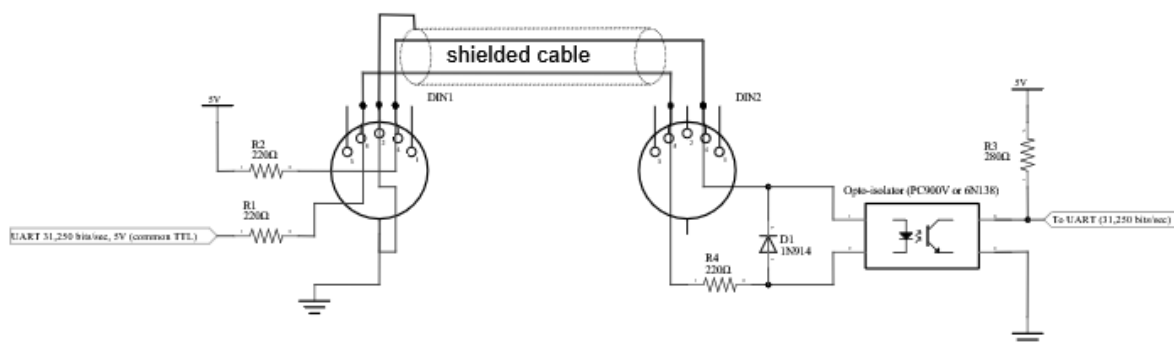
К концу 70-х годов XX века получили распространение музыкальные синтезаторы. Они представляли собой наборы генераторов звуковых частот, управляемых напряжением. Нажатие клавиши на клавиатуре синтезатора включало генератор, частота которого задавалась напряжением от регулятора тональности данной клавиши.

Синтезаторы в то время были полностью аналоговые, все внутренние их блоки (звуковые генераторы, генераторы огибающей, фильтры) управлялись напряжением. Например, звуковой генератор инструмента при подаче напряжения в 1 В мог давать высоту тона 100 Гц, 2 В — 200 Гц, 3 В — 400 Гц и так далее. Очевидно, что для внешнего управления таким прибором мог использоваться только аналоговый интерфейс. Он имел название CV/Gate. На вход CV подавалось управляющее напряжение (Control Voltage), пропорциональное высоте ноты, на вход Gate — импульс (trigger), от которого стартовала и выключалась нота.

Количество отдельных генераторов определяло количество тонов синтезатора, которые могут звучать одновременно. Конкретные модели синтезаторов могли иметь особенности звучания и характерные специальные эффекты. Реализовать управление всей гаммой потенциальных возможностей синтеза звука в рамках одного аналогового устройства было невозможно. Рабочее место музыканта того времени могло состоять из нескольких разнородных синтезаторов; работать с таким зоопарком было сложно. В начале 80-х годов необходимость увеличения гибкости управления разнородными синтезаторами стала очевидной, а развитие электроники

подказало путь решения проблемы: цифровое программное управление. Компании-производители синтезаторов удачно смогли договориться о разработке и поддержке единого стандарта на интерфейс управления синтезаторами, который и появился в 1982 году. Идеология стандарта подразумевала разделение органов управления и аппаратуры синтеза звука. Интерфейс MIDI фактически стал средством передачи положения клавиш и регуляторов от музыкальной клавиатуры к аппаратуре синтеза звука. Теперь музыкант мог с одной клавиатуры управлять несколькими синтезаторами разных типов и фирм-изготовителей. Более того, цифровой формат передачи позволил дополнить аппаратуру секвенсорами — устройствами, запоминающими наигранную мелодию. Теперь музыкант мог записывать композиции или их части, микшировать без потери качества звука и использования сложных студийных магнитофонов того времени. Стандарт быстро завоевал популярность. Постепенно дополняясь новыми возможностями, он не потерял актуальности и в наше время.

Физический уровень интерфейса представляет собой токовую петлю (что обеспечивает гальваническую развязку и безопасность при соединении устройств между собой). Передатчик активный, 0..5 мА, наличие тока обозначает 0, нет тока — 1. Разъем 5-штырьковый DIN 41524. Приёмопередатчик асинхронный, скорость 31,25 кбит/с, формат 8-N-1.



MIDI имеет шинную топологию. Источник данных (например, MIDI клавиатура) имеет разъем выхода данных (MIDI OUT), который соединяется кабелем с входом (MIDI IN) первого синтезатора. Синтезатор имеет второй разъем (MIDI THRU), на котором дублируется поток данных со входа. Это позволяет соединить в цепочку произвольное количество синтезаторов. Синтезатор с клавиатурой имеет все три разъема - IN, OUT, THRU.

Интерфейс однонаправленный (симплексный), то есть источник сообщений (например, музыкальная клавиатура) только передает. Никакой обратной связи не предусмотрено.

Стандарт описывает аппаратный интерфейс, который позволяет соединять электронные музыкальные инструменты и компьютеры различных производителей, описывает протоколы связи для передачи данных от одного устройства к другому. MIDI-устройства могут взаимодействовать с программными приложениями, используя коммуникационный протокол MIDI. Используя соответствующий программный MIDI-секвенсор, внешние MIDI-устройства могут посылать информацию на синтезатор звуковой карты. MIDI базируется на пакетах данных, каждый из которых соответствует MIDI-событию (англ. *MIDI-events*), от нажатия клавиши до простой паузы, эти события разделяются по каналам. Сложная среда MIDI может включать различную аппаратуру, причём каждая часть системы будет отвечать за события на соответствующем канале. Альтернативным вариантом может быть одиночный синтезатор, который сам может управлять всеми каналами.

MIDI — это протокол связи между устройством управления, генерирующим команды, и подчиненным устройством, выполняющим эти команды. Если очень сильно сузить это определение, то можно привести типичный пример: MIDI позволяет исполнителю нажать клавишу на одном инструменте, а получить при этом звук другого или даже нескольких. Любые воздействия исполнителя на органы управления (нажатие клавиш, педалей, изменение положений регуляторов и т. п.) могут быть преобразованы в команды, которые можно передать по MIDI-кабелю на другие инструменты. Эти инструменты, получая команды, обрабатывают их так же, как и при воздействии на их собственные органы управления.

На самом деле протокол MIDI не конкретизирует состав взаимодействующих устройств и не требует наличия живого исполнителя. Суть протокола в том, что в некой системе, состоящей из нескольких устройств, одно устройство (мастер) генерирует команды управления, а все другие устройства (подчиненные) выполняют эти команды. Если подчиненные устройства являются источниками звука (синтезаторы, звуковые модули, семплеры, драм-машины, одним словом, тон-генераторы), то они управляются командами, связанными со звукообразованием: например, "взять ноту До первой октавы" или "переключить тембр на номер 5". Если подчиненные устройства выполняют другие функции, например, обработку аудиосигнала, то и команды для них будут несколько иными. Как бы там ни было, прибор, получает команды управления через свой MIDI-вход (MIDI In).

В качестве мастер-устройства может выступать любой прибор, имеющий MIDI-выход (MIDI Out) и способный посылать на этот выход команды управления. Мастер-устройства можно разделить на два типа: устройства, на которые непосредственно воздействует исполнитель (например, синтезатор) и устройства, которые генерируют управляющие команды автоматически (без участия

исполнителя), на основе ранее введенных данных. Типичным примером устройства последнего типа является секвенсор.

Секвенсор напоминает магнитофон, только записывает он не звук, а команды управления, и не на ленту, а в память компьютера (в широком смысле слова, это может быть и встроенный компьютер синтезатора). Секвенсор позволяет записать действия исполнителя (включая динамику исполнения, стиль, штрихи и т. п.), а затем воспроизвести их в первоизданном виде, точно так же, как если бы исполнитель снова сел за инструмент и сыграл то же самое. Кроме того, в секвенсоре можно редактировать записанную информацию способами, невыполнимыми на магнитофоне: транспонировать партии или отдельные ноты, изменять ритмическую позицию событий или тембр, которым синтезатор будет воспроизводить партию.

Протокол MIDI разрабатывался для управления синтезаторами, а в них, как известно, самый главный орган управления — клавиатура. Неудивительно поэтому, что разработчики MIDI для описания действий исполнителя выбрали принцип клавишного инструмента.

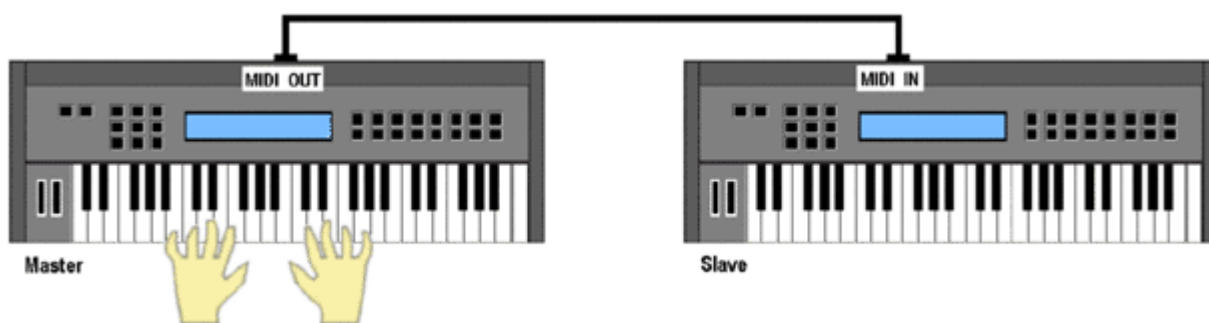
MIDI является выраженным клавишно-ориентированным протоколом.

Это не означает, что управлять тон-генератором можно только с клавиатуры — существуют множество других способов ввода, например, электронные пэды и целые ударные установки, гитарные или духовые контроллеры (о них мы поговорим отдельно и более подробно). Однако, какое бы средство ввода не использовалось, сообщения от него преобразуются в клавишно-ориентированные.

Приемы звукоизвлечения, нехарактерные для клавишного инструмента, могут быть лишь симитированы средствами MIDI с той или иной степенью достоверности.

Коммутация

Как же соединяются устройства в MIDI? Представим себя на месте разработчиков. У нас есть два синтезатора, и мы хотим, чтобы при нажатии клавиши на одном из них второй синтезатор сыграл ту же ноту, но своим звуком. Очевидно, для этого нужно сделать на первом синтезаторе выходной MIDI-разъем, а на втором — входной MIDI-разъем и соединить инструменты MIDI-кабелем. Первый синтезатор при нажатии клавиши должен генерировать сообщение о взятии ноты и посылать его на свой выход, а второй синтезатор — получать это сообщение через вход и воспроизводить звук (рис. 3).



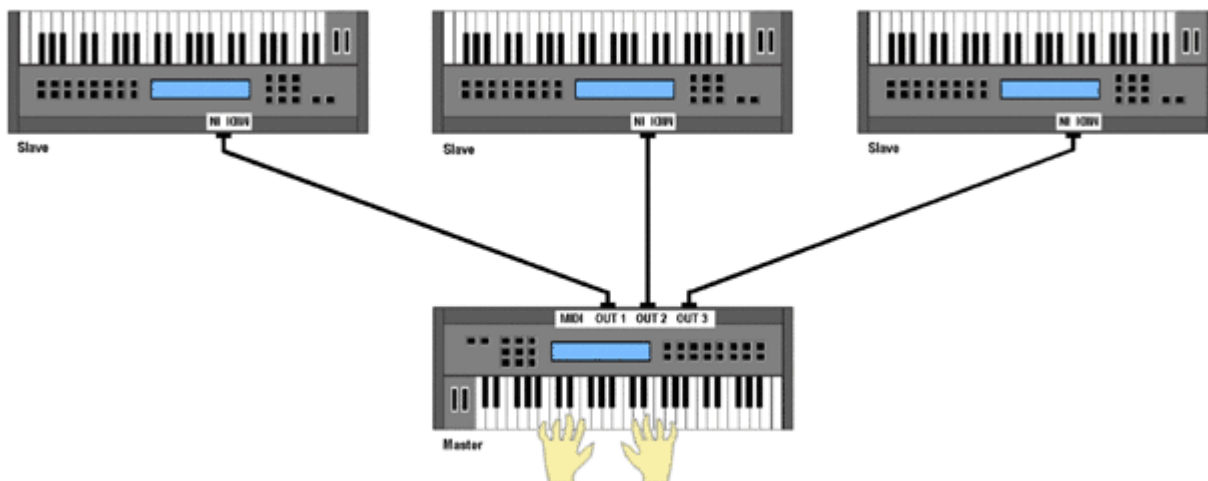
А что будет, если играть аккордами? В прошлом для одновременного извлечения нескольких звуков нужно было бы встраивать в инструмент несколько CV/Gate входов, по одному на каждый голос полифонии. Но MIDI — протокол цифровой, и высота нот кодируется цифрами, а не напряжением. Цифры передаются по MIDI кабелю в виде очень коротких импульсов. Разработчики могли сделать параллельный интерфейс, то есть передавать эти импульсы по нескольким проводам одновременно. Тогда MIDI-кабель мог бы состоять, например, из восьми или шестнадцати проводов, а инструмент получал бы все ноты аккорда одновременно. Но в силу разных причин (о которых поговорим далее подробно) было решено сделать интерфейс последовательным. В этом случае для MIDI-кабеля нужно всего два провода в экранированной оплетке.

Последовательный интерфейс означает то, что импульсы по MIDI-кабелю передаются один за другим. Так что в каждый момент времени приемника достигает только один импульс. Значит, передать одновременно несколько сообщений через один MIDI-разъем невозможно. При игре аккордов ноты каждого аккорда будут передаваться последовательно. Фактически, вместо аккордов синтезатор-приемник будет исполнять только очень-очень плотное арпеджио.

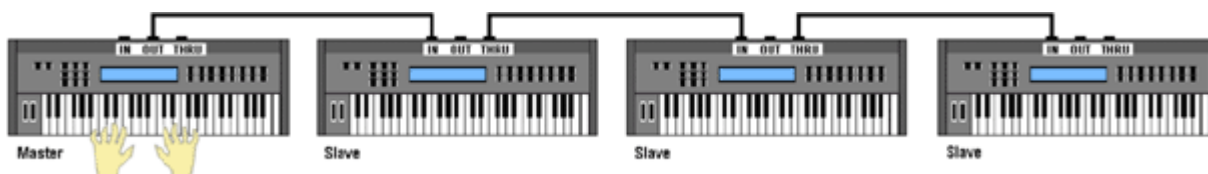
По одному MIDI-кабелю одновременная передача нескольких сообщений невозможна.

Этот факт служит одним из главных аргументов в устах людей, критикующих протокол. На слух запаздывание каждой последующей ноты аккорда неуловимо (так как составляет менее половины миллисекунды), да и к тому же в реальной жизни ни один исполнитель не сыграет аккорд, абсолютно одновременно взяв все его ноты. Однако это не многих успокаивает — все-таки гораздо лучше иметь точную во времени систему.

Итак, мы соединили два синтезатора MIDI-кабелем, по которому передаются сообщения о взятии нот (в том числе и нескольких одновременно, с поправкой на предыдущее замечание). Теперь мы хотим подключить еще два синтезатора, но управлять ими по-прежнему с первого. Для этого можно сделать в первом синтезаторе несколько MIDI-выходов и соединить их тремя MIDI-кабелями с MIDI-входами остальных, то есть, говоря компьютерным языком, организовать сеть с топологией "звезда" (рис. 4).



Это вполне возможно, но не все инструменты (хотя бы по экономическим соображениям) будут оборудоваться несколькими MIDI-выходами. А если потребуется управлять еще двумя или тремя синтезаторами, то выходов точно не напасешься. Было найдено решение: сделать дополнительный разъем MIDI Thru (сквозной). Задача синтезатора, оборудованного таким разъемом — дублировать на него все сообщения, приходящие на вход MIDI In. Тогда нашу систему из четырех синтезаторов можно соединить цепочкой (рис. 5).



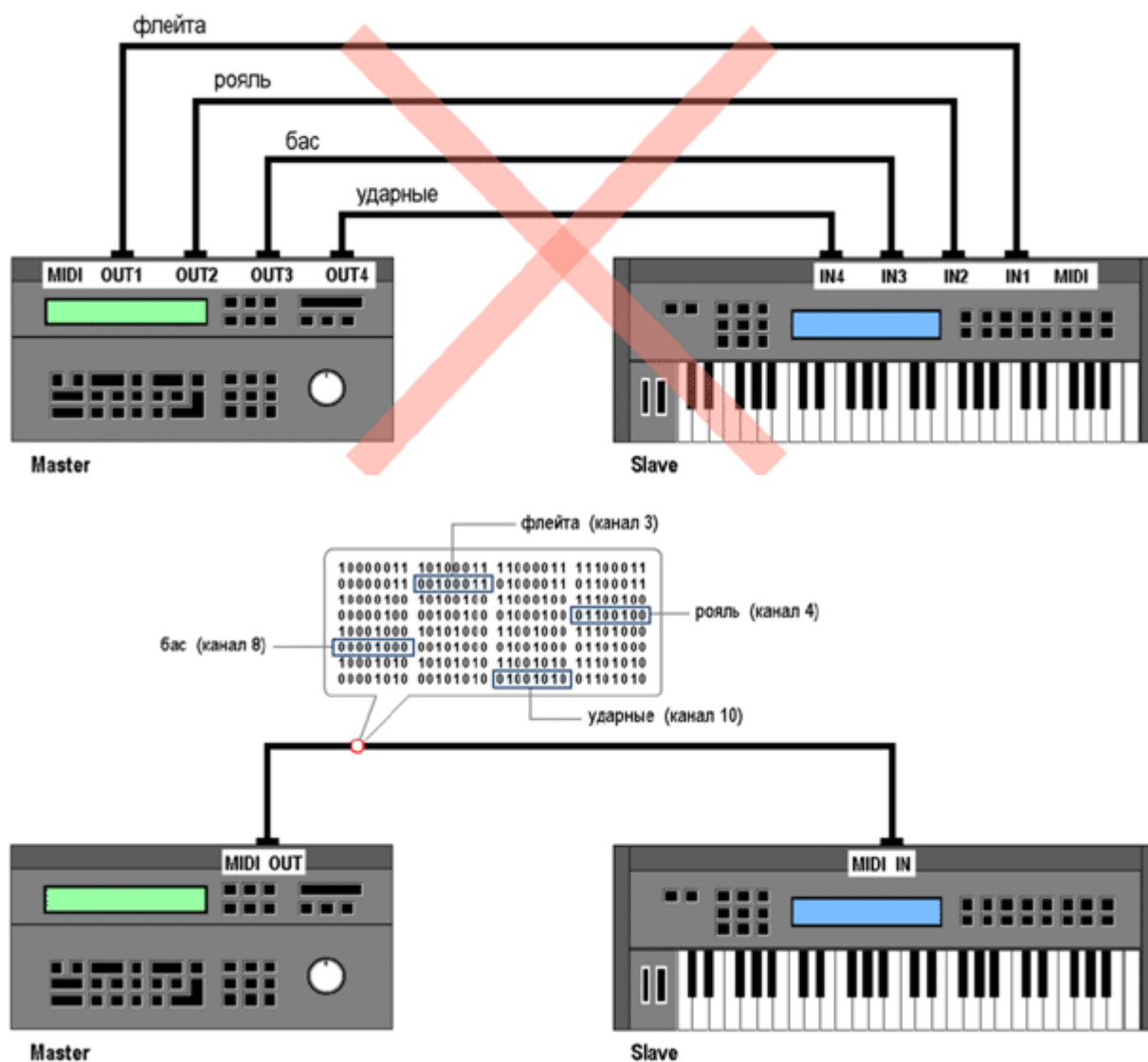
При этом на первом синтезаторе нужен всего один MIDI-выход. Вся информация с этого выхода поступает на MIDI-вход второго синтезатора, дублируется через его сквозной разъем MIDI Thru и далее, по второму MIDI-кабелю, попадает на MIDI-вход третьего инструмента, дублируется, и так до тех пор, пока не достигнет последнего инструмента в цепочке.

Все это хорошо, но как быть, если мы хотим, чтобы каждый инструмент играл по очереди, а не со всеми сразу? Предположим, на первом синтезаторе выбран тембр флейты, на втором — рояля, на третьем — бас-гитары, а на последнем — ударных. Мы нажимаем клавиши на первом инструменте и хотим для начала только рояль, затем только ударные. Конечно, можно бегать и на ходу подключать и отключать нужный синтезатор (что полезно для собственного физического развития), но это не выход, особенно, если приборы соединены цепочкой.

А если мы хотим играть левой рукой партию баса, а правой — рояля? В этом случае клавиатура первого синтезатора должна быть разделена на две зоны. Клавиши из каждой зоны должны посылать сообщения о взятии ноты своему инструменту. Как это сделать, если инструмент оборудован только одним MIDI-выходом?

И, наконец, если наш синтезатор мультитембральный (то есть способен воспроизводить несколько партий одновременно разным тембром), как осуществить его коммутацию с секвенсором, в котором записаны эти партии?

На поставленные вопросы, в сущности, ответа два: либо использовать столько MIDI-входов, выходов и кабелей, сколько партий звучит одновременно (рис. 6, а), либо... Сделать так, чтобы все партии передавались по одному MIDI-кабелю (рис. 6, б). Нужно ли пояснять, почему был выбран второй вариант?!

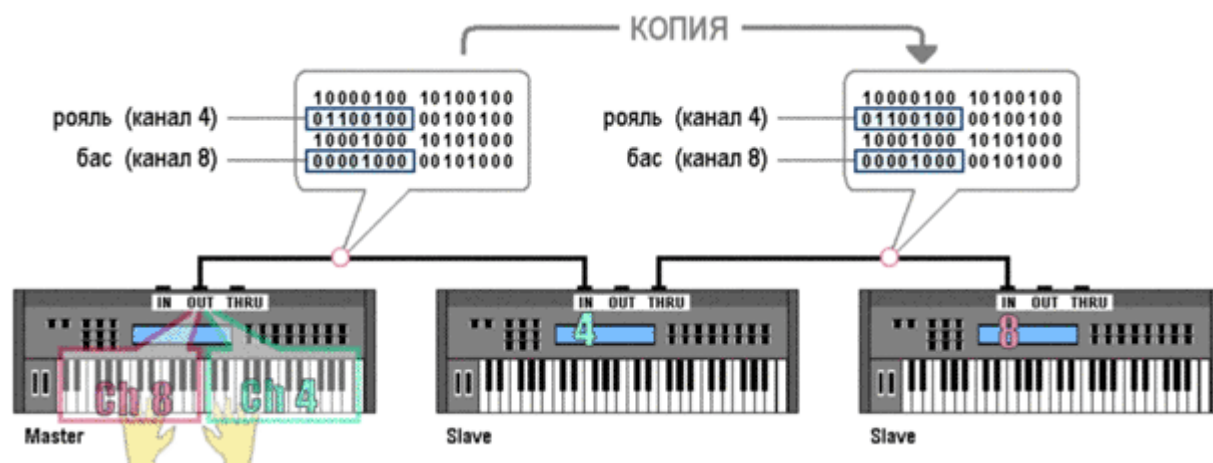


Разработчики предложили передавать данные по шестнадцати логическим каналам. Слово "логический" означает то, что все каналы существуют в виде абстракции и передаются по одному MIDI-кабелю. Просто каждое сообщение, например, о взятии ноты, снабжается дополнительным числом — номером канала, на котором звучит нота. В этом коренное отличие MIDI от аналоговой звуковой коммутации, где, например, партия рояля идет с магнитофона на микшер по отдельному кабелю, который можно потрогать руками. В MIDI по одному кабелю могут передаваться одновременно партии всего симфонического оркестра (естественно, в виде управляющих команд, а не звука).

Понятно, что при последовательной передаче в каждый момент времени приемника будет достигать сообщение только для одного канала. Так что абсолютной ритмической согласованности в игре, например, бочки и бас-гитары при передаче данных по одному MIDI-кабелю добиться нельзя. Но опять же, покажите басиста, который идеально "попадает" в бочку барабанщика. Многих критиков протокола подобные фразы не утешают, хотя разница в приходе сообщений практически неощутима — все в том же порядке долей миллисекунды.

Каждый тон-генератор может быть настроен на прием сообщений по одному или нескольким MIDI-каналам. Данные тех каналов, по которым прием не идет, просто игнорируются. В нашем примере пусть первый синтезатор играет на MIDI-канале 3, второй настроен на прием по каналу 4, третий — по каналу 8 и последний — по каналу 10. Тогда для переключения тембров нужно переключать каналы, по которым передает информацию первый синтезатор: включили на канал 3 — звучит его же флейта, на канал 8 — бас с третьего синтезатора, на канал 10 — барабаны с последнего. Но тогда клавиатуры на втором, третьем и четвертом синтезаторах вообще не нужны. Это соображение привело к широкому распространению звуковых модулей — синтезаторов без клавиатуры. С другой стороны, первый синтезатор может вообще не производить звук, а только служить центром управления для всех остальных. Это соображение привело к появлению MIDI-клавиатур — устройств, которые звуков не содержат, а служат для управления другими синтезаторами или звуковыми модулями.

Если первый синтезатор позволяет разделить клавиатуру на две зоны, то можно назначить зону для левой руки на передачу по каналу 8, а для правой — по каналу 4. Тогда мы сможем одновременно исполнять партии баса и рояля. При этом на первом синтезаторе потребуется только один MIDI-выход (рис. 7).



Возможности живого исполнения нескольких партий одновременно ограничены, поэтому чаще всего партии записываются в секвенсор по очереди. Причем записать партии можно на простом и неказистом по звучанию инструменте — ведь записывается не звук, а только действия исполнителя (можно даже настучать

партию на MIDI-клавиатуре без подзвучки). А для сведения одолжить пару качественных синтезаторов или звуковых модулей.

Каждый синтезатор при наличии выхода MIDI Thru обязан дублировать на него всю приходящую информацию, в том числе и по тем каналам, на прием которых он сам настроен. Это позволяет передавать партию дальше, на другие синтезаторы, настроенные на тот же канал, для создания, например, жирного звучания в унисон.

MIDI — протокол реального времени Это означает, что вся система работает по принципу "получил — выполнил". Исполнитель нажимает клавишу, клавиатура генерирует сообщение "взять ноту" и передает его на вход тон-генератора. Тон-генератор немедленно воспроизводит ноту. Таким образом, никакие параметры, связанные с моментами выполнения команд, в сообщениях не передаются. Моментом выполнения считается момент получения команды. Поэтому сообщения вроде "сыграть ноту через две секунды" в MIDI отсутствуют. Как это ни странно, сообщение "сыграть ноту длительностью две секунды" в системе реального времени также невозможно.

Представьте: как только исполнитель нажимает клавишу, тон-генератору посылается команда "взять ноту", и он должен немедленно начать воспроизведение. Какой длительности будет нота, тон-генератор не знает и не может знать. Здесь все зависит от исполнителя — нота будет звучать до тех пор, пока тот не отпустит клавишу, и в тон-генератор не поступит команда "снять ноту". Вот теперь можно сказать — нота звучала полторы секунды. В MIDI все параметры, связанные со временем протекания того или иного процесса, ведут себя аналогично. Например, темп задается не каким-нибудь сообщением "120 ударов в минуту", а самими сообщениями о взятии ноты, приходящими от устройства управления в соответствующем темпе.

Таким образом, задача тон-генератора в MIDI существенно упрощается: не нужно отслеживать время чего-либо, нужно только сразу реагировать на команды. Отсюда — одна из самых распространенных проблем MIDI, так называемые зависшие ноты. Если в тон-генератор пришло сообщение "взять ноту", а сообщение "снять ноту" по каким-либо причинам не пришло (например, из-за обрыва кабеля), нота будет звучать вечно. Ну, или до тех пор, пока не перезапустит тон-генератор.

В общем, получается, что временем звучания нот, темпом, и другими временными моментами управляет сам исполнитель. Если мы хотим записать действия исполнителя, нужно каждое MIDI-сообщение, поступившее в секвенсор, снабдить меткой времени. Примерно так: сообщение "взять ноту" поступило на первый тик. Записываем его в память вместе с меткой 1. Сообщение "снять ноту" поступило на двадцать первый тик, записываем его с меткой 21. Если секвенсор отсчитывал двадцать тиков в секунду, то, очевидно, длина записанной ноты — одна секунда. При воспроизведении секвенсор начинает снова отсчитывать тики. Подошел первый тик, в тон-генератор отправилось сообщение о взятии ноты, подошел 21-й тик — о

снятии ноты. Таким образом, мы записали, а затем воспроизвели действия исполнителя.

Кроме того, секвенсор позволяет те же действия *запрограммировать*, то есть записать в память без необходимости живого исполнения. Такое программирование в свое время дало жизнь множеству стилей электронной музыки и в корне изменило творческий процесс.

Компоненты MIDI Понятия, затронутые выше, также требуют подробного обсуждения, но уже сейчас ясно, из чего должен состоять протокол. Очевидно, для того, чтобы одно устройство реагировало на управляющие воздействия с другого, нужно стандартизировать две вещи: язык общения устройств и физический способ их соединения. Для того, чтобы сообщения можно было сохранять, а впоследствии воспроизводить, нужен также стандартный формат их хранения.

Протокол MIDI состоит из трех частей: спецификации формата данных, аппаратной спецификации интерфейса и спецификации формата хранения данных.

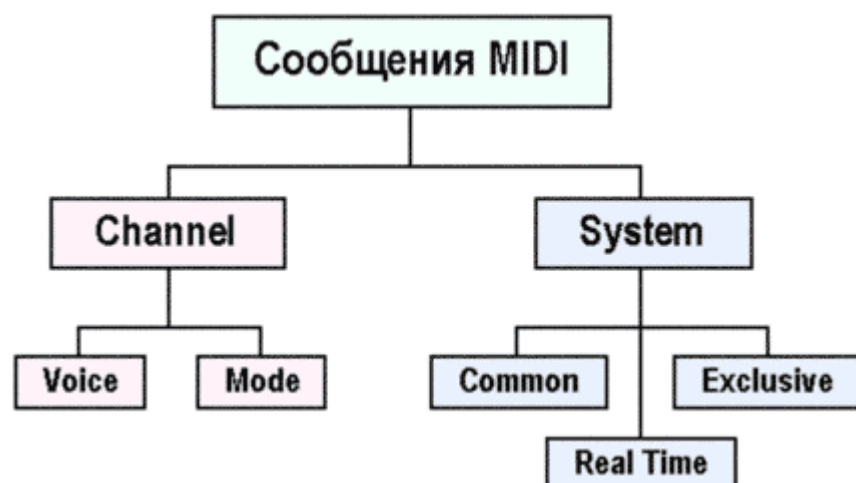
Начнем разговор о MIDI с первой части, то есть с языка, на котором общаются устройства.

Язык MIDI Для управления одним устройством с другого нужно придумать язык команд, понятный им обоим. То есть составить список команд, обозначить каждую команду своим кодом и договориться: если устройство понимает команду, оно должно на нее реагировать так-то и так-то, если не понимает — игнорировать.

Язык MIDI состоит только из команд управления и параметров этих команд. Ничего другого по MIDI-кабелю не передается. Любые, даже такие непростые музыкальные моменты, как, например, игра баса легато с "подъездом" к каждой ноте на сильной доле такта, — это всего лишь результат обработки команд тон-генератором. Команды в языке MIDI называются сообщениями.

Сообщения логично разделить на два основных типа: одни управляют звукообразованием, то есть говорят, например, какую ноту и как громко играть, вторые выполняют служебные функции, вроде изменения настроек тон-генератора и синхронизации. Поскольку извлечение звуков происходит в MIDI на определенном канале, сообщения первого типа называются сообщениями канала (Channel Messages). Сообщения второго типа называются системными (System Messages). Сообщения канала всегда относятся к какому-либо MIDI-каналу и обрабатываются тон-генератором только в том случае, если он настроен на прием данного канала. Выше говорилось, что соединение нескольких тон-генераторов через разъем MIDI Thru, при соответствующих настройках инструментов, позволяет передавать "каждому свое". Системные сообщения предназначены всем

устройствам в MIDI-системе, независимо от того, на прием каких каналов они настроены.



Сообщения канала делятся, в свою очередь, на голосовые (Channel Voice Messages) и сообщения режима канала (Channel Mode Messages). Системные сообщения делятся на общесистемные (System Common Messages), сообщения реального времени (System Real Time Messages) и эксклюзивные (System Exclusive Messages).

Голосовые сообщения канала посылают в тон-генератор информацию об управлении звуком. Они "докладывают" тон-генератору о том, что сейчас делает исполнитель — нажимает клавишу, крутит колесо модуляции, двигает фейдер или отпускает педаль. То есть голосовые сообщения описывают действия исполнителя в цифровой форме. Тон-генератор, получая эти сообщения, анализирует их и реагирует на них примерно так: "Ага, это нажатие клавиши До первой октавы — нужно воспроизвести ноту; это поворот звуковысотного колеса — нужно сместить у ноты высоту тона; а это — неизвестное мне сообщение, ничего делать не буду".

Тон-генератор может работать в нескольких режимах (о них поговорим далее), и в каждом режиме он реагирует на голосовые сообщения по-разному. Сообщения режима канала как раз и нужны для переключения этих режимов, но они выполняют и еще несколько полезных действий.

Общесистемные сообщения выполняют несколько разнородных задач. В их числе — синхронизация MIDI- и аудиоустройств (например, магнитофонов) посредством протокола MIDI Time Code (MTC), передача позиции песни, выбор песни и даже запрос на подстройку осцилляторов синтезатора.

Системные сообщения реального времени предназначены для синхронизации MIDI-устройств, например, секвенсоров и драм-машин, по протоколу MIDI Clock. Сюда же входят сообщения начальной инициализации ("сброса") устройства и предотвращения некорректной работы (выключения зависших нот). Сообщения реального времени отличаются от всех других тем, что имеют наивысший приоритет

передачи, то есть, например, легко могут вклиниться между частями другого сообщения. Это и понятно — ритмическая точность дороже всего.

Системные эксклюзивные сообщения (обозначаемые для краткости SysEx) — это своеобразная палочка-выручалочка для производителей устройств. Сегодня на рынке существует множество типов оборудования, и каждый конкретный прибор имеет свои специфические возможности. Организации MMA и JMSC могли пойти по пути постоянных обновлений спецификации MIDI: появилась какая-то новая функция, скажем, в дымовой машине — выделить под нее новое сообщение и занести в стандарт. Понятно, что это тупиковый путь. За всем не уследишь, а абсолютной совместимости между устройствами все равно не добиться (да и какая совместимость может быть у дымовой машины и синтезатора?). Поэтому было решено дать возможность каждому производителю определять свои собственные (эксклюзивные, исключительные) сообщения, даже под конкретное устройство.

Несколько сообщений SysEx являются универсальными и поддерживаются устройствами разных производителей. Эти сообщения применяются для настройки таких параметров тон-генератора, как общая громкость и панорама, а также для реализации дополнительных протоколов в рамках MIDI, например, протокола передачи семплов (MIDI Sample Dump Standard), управления сценическим светом и пиротехническими устройствами (MIDI Show Control), управления транспортными функциями оборудования (MIDI Machine Control).

Принцип кодирования сообщений. Очевидно, сообщение должно состоять из двух компонентов: первый описывает тип сообщения, то есть отвечает на вопрос "что надо делать", второй — уточняет действие и отвечает на вопрос "как надо делать". Например, при нажатии клавиши первый компонент сообщает тон-генератору "взять ноту", а второй уточняет: "Эта нота — До малой октавы". В некоторых случаях уточняющий компонент не нужен, например, в системном сообщении реального времени, останавливающем секвенсор, достаточно одного компонента, который говорит "стоп".

Компоненты сообщений в протоколе MIDI представлены байтами. Компонент, описывающий тип сообщения, называется статус-байтом, компонент, уточняющий сообщение — байтом данных. Если уточняющей информации много, она может быть представлена несколькими байтами данных. Таким образом, каждое MIDI-сообщение состоит из одного статус-байта и, если необходимо, одного или нескольких байтов данных. Передаются эти сообщения по MIDI-кабелю в том же порядке — сначала статус-байт, затем байты данных. Количество байтов данных жестко закреплено за каждым сообщением. Так что, если, например, тон-генератор получил статус-байт "нажата клавиша", он ждет за ним два байта данных, первый из которых содержит номер нажатой клавиши, второй — скорость нажатия.

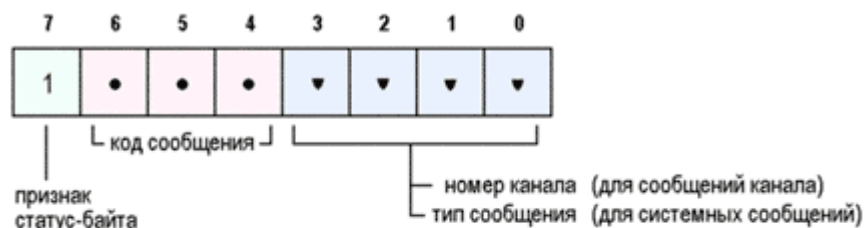
Для системных эксклюзивных сообщений сделано исключение — их длина жестко не задается. Она определяется специальным статус-байтом, который помещается в конец сообщения.

MIDI-сообщения — это поток данных в реальном времени. При передаче данных иногда могут происходить их потери и прочие неприятности. В компьютерных сетевых протоколах, в случае прихода испорченных данных (что проверяется по контрольной сумме) происходит повторный запрос к серверу до тех пор, пока данные не придут в целости и сохранности. В протоколе MIDI такая возможность отсутствует (по крайней мере, при передаче голосовых сообщений), и приемник всегда находится в пассивном состоянии по отношению к передатчику. Что принял, то принял.

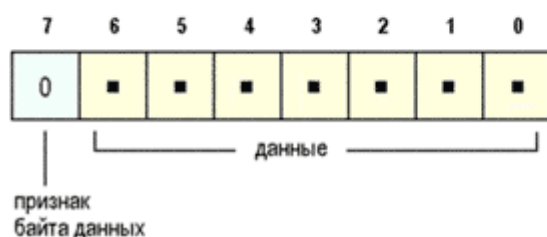
Отсюда вывод: необходимо сделать так, чтобы приемник в любой момент мог отличить статус-байт от байта данных, без всяких подсчетов и необходимости синхронизации с передатчиком. Для этого в протоколе MIDI каждый статус байт содержит в старшем значащем разряде единицу, а каждый байт данных — ноль. Так что один бит из байта расходуется по служебной необходимости. Для кодирования сообщений и передачи данных остаются только семь бит. Отсюда — фундаментальное число 128, пронизывающее насквозь весь протокол. Семью битами можно представить только 128 различных значений (от 0 до 127, 2^7-1). Именно поэтому в MIDI 128 нот, 128 уровней динамики и так далее.

Стало быть, и в статус-байте можно закодировать 128 сообщений. Разработчики протокола могли так и сделать, но поступили иначе, и этим во многом объясняется гибкость и живучесть MIDI. Во-первых, 128 сообщений рано или поздно окажется недостаточно, во-вторых, в каждом сообщении канала нужно передавать номер канала, которому адресовано сообщение. Под номер канала придется отводить один байт данных, что не рационально — сообщения канала используются в MIDI гораздо активнее системных, а если к каждому такому сообщению будет "пришит" дополнительный байт, это намного снизит пропускную способность MIDI. К тому же каналов всего 16, и для их представления достаточно четырех бит. Поэтому было решено кодировать номер канала в младшей половине статус-байта (рис. 12)

Статус-байт



Байт данных



Получается такая картина: старший бит статус-байта занят, так как должен всегда содержать единицу, четыре младших бита — тоже, так как содержат номер MIDI-канала. На кодирование типа сообщения остается всего три бита, которыми можно закодировать восемь сообщений. Семь из них отвели под сообщения канала, а последнее, все биты которого единичные (111), сделали признаком системного. Системные сообщения не требуют номера канала, поэтому младшие четыре бита статус-байта могут быть использованы для кодирования конкретного типа сообщения (всего их получается 16).

Тон-генератор анализирует полученный байт примерно по следующей схеме. Если его старший бит равен нулю — это байт данных и младшие семь бит нужно рассматривать как числовой параметр. Если старший бит равен единице — это статус-байт, и тогда три бита после него отражают тип сообщения. Если эти три бита единичные, сообщение системное, какое именно — задается младшими четырьмя битами. В противном случае — это сообщение канала, и тогда младшие четыре бита задают номер канала. Все это может показаться сложным для человека, но компьютер работает с битами "на ура" и вышеуказанную процедуру выполняет за доли микросекунды.

Однако, что же получается: вместо 128 возможных сообщений мы можем закодировать только семь голосовых и шестнадцать системных? Да, именно так. Системных сообщений, в принципе, достаточно, а вот все голосовые в семь штук явно не уложишь. Поэтому разработчики MIDI применили многоуровневый метод кодирования. Но об этом поговорим в следующий раз.

Мы выяснили, что структура статус-байта MIDI позволяет закодировать семь голосовых и шестнадцать системных сообщений. Возможные варианты наглядно демонстрирует рис. 1.

Сообщения канала									
номер	код								
					номер канала				
1	1	0	0	0	n	n	n	n	
2	1	0	0	1	n	n	n	n	
3	1	0	1	0	n	n	n	n	
4	1	0	1	1	n	n	n	n	
5	1	1	0	0	n	n	n	n	
6	1	1	0	1	n	n	n	n	
7	1	1	1	0	n	n	n	n	

Системные сообщения									
номер	код								
1	1	1	1	1	0	0	0	0	
2	1	1	1	1	0	0	0	1	
3	1	1	1	1	0	0	1	0	
4	1	1	1	1	0	0	1	1	
5	1	1	1	1	0	1	0	0	
6	1	1	1	1	0	1	0	1	
7	1	1	1	1	0	1	1	0	
8	1	1	1	1	0	1	1	1	
9	1	1	1	1	1	0	0	0	
10	1	1	1	1	1	0	0	1	
11	1	1	1	1	1	0	1	0	
12	1	1	1	1	1	0	1	1	
13	1	1	1	1	1	1	0	0	
14	1	1	1	1	1	1	0	1	
15	1	1	1	1	1	1	1	0	
16	1	1	1	1	1	1	1	1	

Системные сообщения отложим до лучших времен, а сегодня поговорим о сообщениях канала — основном строительном материале языка MIDI.

Напомню, что сообщения канала делятся на два типа: голосовые и сообщения режима канала. Первые связаны со звукообразованием, вторые к этому прямого отношения не имеют, но играют в деле управления MIDI-устройством не последнюю роль.

Голосовое сообщение заставляет тон-генератор произвести какое-либо изменение в звуке (или, собственно, сам звук). Например, если исполнитель нажимает клавишу, в тон-генератор посылается сообщение Note On, на которое тот реагирует воспроизведением ноты. Если исполнитель поворачивает колесо модуляции, тон-генератор изменяет глубину модуляции звучащей ноты.

Во многих случаях то, каким органом управления создается сообщение, принципиального значения не имеет. Тогда спецификация привязывает сообщение непосредственно к параметру синтеза. Так, для изменения панорамы в MIDI-канале используется сообщение Pan, но спецификация не уточняет, как оно генерируется. То же самое справедливо и в отношении многих других сообщений, например, задающих длительность участков огибающей.

Особое внимание следует обратить на принципиальный момент: в MIDI все параметры синтеза задаются безразмерными числами. Вы поворачиваете колесо модуляции — в тон-генератор приходит сообщение типа "колесо модуляции: 652". Вы нажимаете клавишу — сообщение имеет вид "нота номер 60, динамика взятия 120". Тон-генератор сам решает, как преобразовать эти числа в реальный коэффициент модуляции, в высоту тона в герцах, и в выходной уровень в децибелах.

В некоторых случаях спецификация *однозначно задает* такое преобразование. Например, нота с номером 60 должна быть всегда нотой До первой октавы и иметь частоту 261 Гц. Тон-генератор подчиняется. В других случаях спецификация *рекомендует*, чтобы, например, при динамике 120 уровень на выходе осциллятора снижался на 0,98 дБ от максимального. Наконец, для параметров вроде модуляции никаких рекомендаций не дается — коэффициент модуляции зависит от особенностей внутренней архитектуры инструмента.

Уровни кодирования Голосовых сообщений довольно много: это и нажатие/отпускание клавиши, и давление на клавишу после нажатия (послекасание), вращение колес высоты тона и модуляции, нажатие педалей, кнопок, передвижение слайдеров; наконец, сообщения, напрямую задающие параметры синтеза. В общем, почти сотня наберется. А на все про все только семь статус-байтов...

Проблема была решена так: самым часто употребляемым сообщениям выделили собственный статус-байт, а для всех остальных сделали статус-байт-посредник. Ситуацию можно сравнить с телефонными номерами. Для звонков внутри города (предположительно, наиболее частых) вы набираете прямой номер, а для междугородных звонков (предположительно, более редких) вы должны набрать префикс (код города), и только потом номер. Междугородный номер приходится набирать дольше, чем внутригородской. Так и в MIDI — сообщения с прямым статус-байтом проходят быстрее, чем сообщения с байтом-посредником.

Итак, можно выделить два множества сообщений или, иначе, два уровня кодирования. Доступ к первому — прямой, ко второму — только через посредника.

Следующая таблица показывает сообщения первого уровня (рис. 4). Статус-байты даны в двоичном, десятичном и шестнадцатеричном виде. Младшая половина статус-байта (4 бита) содержит номер канала, которому адресовано сообщение. Так, например статус-байты в диапазоне от 10010000 (0x90) до 10011111 (0x9F) означают одно и тоже сообщение Note On, в диапазоне от 11000000 (0xC0) до 11001111 (0xCF) — сообщение Program Change и т. д. То есть, статус-байт, соответствующий одному типу сообщения, может быть представлен в шестнадцати вариантах, в зависимости от номера канала. В MIDI, как и во всем компьютерном мире, отсчет ведется с нуля. При передаче сообщений первый канал кодируется как 0000, второй как 0001, а шестнадцатый — как 1111 (десятичное 15, шестнадцатеричное 0xF).

Статус-байт			Сообщение	Байт данных 1	Байт данных 2
Hex	Bin	Dec			
8n	1000nnnn	128..143	Note Off	номер ноты	динамика
9n	1001nnnn	144..159	Note On	номер ноты	динамика
An	1010nnnn	160..175	Polyphonic Key Pressure	номер ноты	давление
Bn	1011nnnn	176..191	Control Change	номер контроллера	значение контроллера
Cn	1100nnnn	192..207	Program Change	номер программы	—
Dn	1101nnnn	208..223	Channel Pressure	давление	—
En	1110nnnn	224..239	Pitch Wheel Change	LSB	MSB

Основной "целевой группой" MIDI являются инструменты клавишного типа (синтезаторы), поэтому набор наиболее часто употребляемых сообщений вполне обоснован. Это, прежде всего, сообщения

- о нажатии и отпускании клавиши (Note On / Note Off),
- Послекасание (сила нажатия на клавишу) (Pressure),
- вращения звуковысотного колеса (Pitch Wheel Change)
- смены программы (Program Change).

Извлечение звуков MIDI является выраженным клавишно-ориентированным протоколом, поэтому процесс извлечения звуков кодируется двумя простыми сообщениями — взять ноту (Note On) и снять ноту (Note Off). Исполнитель при нажатии клавиши задает сразу три параметра: *момент начала звучания, динамику и высоту тона*. Длительность звука определяется по моменту отпускания клавиши.

Такая модель хорошо подходит и для ударных инструментов. Здесь, правда, основным выразительным средством является динамика, так как высотой тона ударных в процессе исполнения чаще всего управлять нельзя, как и длительностью звука.

К инструментам с другим принципом звукоизвлечения (струнным, духовым) такая модель не подходит. Например, на скрипке музыкант может управлять тремя

указанными параметрами по отдельности: левой рукой он задает высоту звука (путем прижатия струны в нужной позиции), правой — момент начала и окончания звучания, а также динамику (ведением смычка по струне). Эти параметры музыкант может изменять независимо друг от друга, причем в процессе звучания. Например, момент появления звука никак не укладывается в схему "нажать клавишу с нужной громкостью". Здесь это очень тонкий процесс: звук может постепенно возникать из тишины, набирая громкость, и так же уходить в тишину, то есть музыкант способен изменять динамику в процессе звукоизвлечения.

MIDI все упрощает и требует однозначного решения — в какой момент и с какой динамикой должна звучать нота. После получения сообщения Note On динамика передается в генератор огибающей синтезатора, и ее изменение в процессе звучания происходит не по воле музыканта, а автоматически на основе параметров огибающей. Конечно, в MIDI есть способы имитации "неклавишной" техники звукоизвлечения, но это требует дополнительных усилий и хорошего знания возможностей конкретного тон-генератора.

В MIDI используется равномерно-темперированный строй и 128 нот различной высоты (с номерами от 0 до 127). Частота нот задается с помощью номера. Нота с номером 60 — всегда До первой октавы (частота 261 Гц), рис. 25.



Полный частотный диапазон, покрываемый таким образом, превышает 10 октав (от 8 Гц до 12543 Гц), что более чем достаточно. Для перевода номера ноты в частоту можно использовать формулу на рис. 26.

Зависимость частоты ноты от ее MIDI-номера

$$f_N = 440 \cdot 2^{\frac{N-69}{12}},$$

где N — номер ноты;

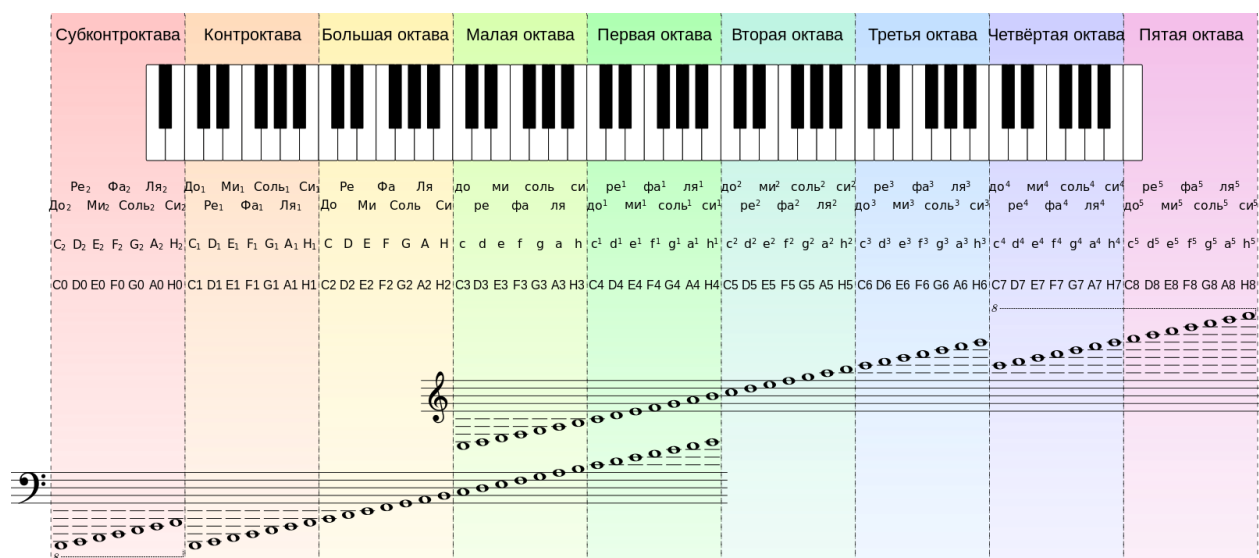
f_N — частота ноты в Гц.

$$\sqrt[12]{2} = 1.0594631$$

Частоты в герцах (интервал от До первой октавы в полутонах)											
Октава → Нота ↓	Суб-контр	Контр	Большая	Малая	1	2	3	4	5	6	7
C	16,352 (-48)	32,703 (-36)	65,406 (-24)	130,81 (-12)	261,63 (±0)	523,25 (+12)	1046,5 (+24)	2093,0 (+36)	4186,0 (+48)	8372,0 (+60)	16744,0 (+72)
C# / D$\flat$	17,324 (-47)	34,648 (-35)	69,296 (-23)	138,59 (-11)	277,18 (+1)	554,37 (+13)	1108,7 (+25)	2217,5 (+37)	4434,9 (+49)	8869,8 (+61)	17739,7 (+73)
D	18,354 (-46)	36,708 (-34)	73,416 (-22)	146,83 (-10)	293,66 (+2)	587,33 (+14)	1174,7 (+26)	2349,3 (+38)	4698,6 (+50)	9397,3 (+62)	18794,5 (+74)
D# / E$\flat$	19,445 (-45)	38,891 (-33)	77,782 (-21)	155,56 (-9)	311,13 (+3)	622,25 (+15)	1244,5 (+27)	2489,0 (+39)	4978,0 (+51)	9956,1 (+63)	19912,1 (+75)
E	20,602 (-44)	41,203 (-32)	82,407 (-20)	164,81 (-8)	329,63 (+4)	659,26 (+16)	1318,5 (+28)	2637,0 (+40)	5274,0 (+52)	10548 (+64)	21096,2 (+76)
F	21,827 (-43)	43,654 (-31)	87,307 (-19)	174,61 (-7)	349,23 (+5)	698,46 (+17)	1396,9 (+29)	2793,8 (+41)	5587,7 (+53)	11175 (+65)	22350,6 (+77)
F# / G$\flat$	23,125 (-42)	46,249 (-30)	92,499 (-18)	185,00 (-6)	369,99 (+6)	739,99 (+18)	1480,0 (+30)	2960,0 (+42)	5919,9 (+54)	11840 (+66)	23679,6 (+78)
G	24,500 (-41)	48,999 (-29)	97,999 (-17)	196,00 (-5)	392,00 (+7)	783,99 (+19)	1568,0 (+31)	3136,0 (+43)	6271,9 (+55)	12544 (+67)	25087,7 (+79)
G# / A$\flat$	25,957 (-40)	51,913 (-28)	103,83 (-16)	207,65 (-4)	415,30 (+8)	830,61 (+20)	1661,2 (+32)	3322,4 (+44)	6644,9 (+56)	13290 (+68)	26579,5 (+80)
A	27,500 (-39)	55,000 (-27)	110,00 (-15)	220,00 (-3)	440,00 (+9)	880,00 (+21)	1760,0 (+33)	3520,0 (+45)	7040,0 (+57)	14080 (+69)	28160,0 (+81)
A# / B$\flat$	29,135 (-38)	58,270 (-26)	116,54 (-14)	233,08 (-2)	466,16 (+10)	932,33 (+22)	1864,7 (+34)	3729,3 (+46)	7458,6 (+58)	14917 (+70)	29834,5 (+82)
B	30,868 (-37)	61,735 (-25)	123,47 (-13)	246,94 (-1)	493,88 (+11)	987,77 (+23)	1975,5 (+35)	3951,1 (+47)	7902,1 (+59)	15804 (+71)	31608,5 (+83)

Примечание: Иногда нота Си обозначается как «Н» вместо «В».

С номерами нот работать неудобно, но и музыкальный способ наименования (вроде *d3* — Ре третьей октавы или *Cis1* — До диез контроктавы) не подходит; еще и потому, что в разных странах применяются разные обозначения. Спецификация MIDI по этому поводу молчит, а большинство производителей используют такую систему: октава указывается после буквенного обозначения ноты, но первая октава (с музыкальной точки зрения) считается третьей.



Например, До первой октавы называется C3. Производные ступени всегда обозначаются диэзами. Так, Ре бемоль второй октавы будет называться C#4 (До диэз в темперированном строе равно Ре бемоль), Соль контроктавы — G1. Нота Си обозначается по англо-американской нотации, то есть как B, а не как H. Соответственно, Си бемоль будет называться A#. Замечу, что в некоторых инструментах и секвенсорах (в основном, старых) До первой октавы обозначается как C4, что может привести к путанице. Некоторые устройства в этой связи предлагают выбрать, что считать нотой До первой октавы: C3 или C4.

Для наборов ударных частота ноты и строй не имеют смысла: в зависимости от номера ноты тон-генератор просто запускает сопоставленный с этим номером звук. Как правило, звук ударного инструмента обозначается в секвенсоре осмысленным именем, а не просто MIDI-номером ноты. Имя берется из так называемой карты ударных (Drum Map).

Динамика

Динамику извлечения звука называют по-разному: просто динамика, скорость нажатия, громкость. Наиболее корректно в случае клавишного инструмента говорить о *скорости*, поскольку именно скорость нажатия клавиши определяет громкость звучания ноты. Сила нажатия не всегда однозначно соответствует громкости — так, можно довольно сильно давить на клавишу фортепиано, но при этом инструмент будет звучать тихо. Это связано с особенностями механики клавишных.

В спецификации MIDI также используется термин "скорость нажатия", Velocity (от англ. "velocity" — "скорость"). Но для удобства (в ущерб корректности) будем называть "скорость нажатия" одним словом — "динамика".

Разработчики MIDI решили кодировать динамику одним байтом данных, и поэтому в MIDI есть 128 уровней Velocity. Можно задать динамику как целой партии (или громкости канала), так и для отдельно взятой ноты.

Использование значения динамики (за исключением нулевого) спецификацией жестко не определено и зависит от конкретного тон-генератора. Как правило, динамика управляет блоком VCA инструмента — в зависимости от динамики изменяется время или уровень атаки, а, следовательно, и общая громкость звука. Часто динамика управляет параметрами фильтра, делая звук ярче при громкой игре. В семплерах может применяться раскладка по динамике, когда на одну и ту же ноту вешаются несколько семплов, полученных при записи с разными нюансами исполнения. В зависимости от динамики воспроизводится нужный семпл, либо результат микширования двух смежных семплов (для маскирования их разного спектрального состава). Это позволяет довольно успешно имитировать акустические инструменты, звук которых может принципиально отличаться по тембру при разной динамике исполнения.

Возникает вопрос: как должна зависеть громкость сигнала на выходе осциллятора от значения Velocity? Спецификация MIDI рекомендует тон-генераторам использовать логарифмическую зависимость в соответствии с формулой

Зависимость уровня сигнала от Velocity

$$L_{dB} = 40 \lg \left(\frac{Velocity}{127} \right)$$

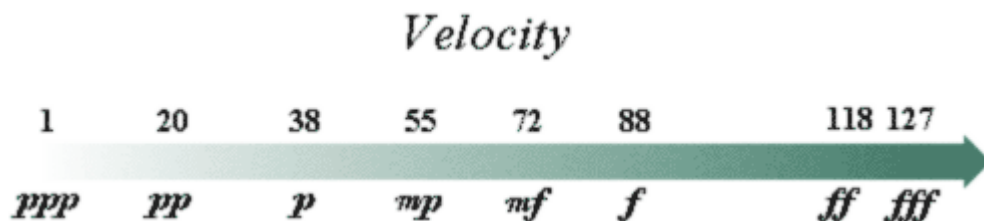
Где L_{dB} — максимальный уровень на выходе осциллятора при заданной огибающей сигнала; обычно точка максимального уровня совпадает с моментом завершения стадии атаки. Зависимость амплитуды сигнала (в процентах) от значения динамики можно найти по формуле:

Зависимость амплитуды сигнала от Velocity

$$A_{\%} = 10^{2 \lg \left(\frac{Velocity}{127} \right)} \cdot 100\%$$

При максимальной динамике ($Velocity = 127$) этот уровень равен 0 дБ (100%), при динамике 1 — -84,15 дБ (0,000062%). Значение динамики 0 используется для служебных целей (см. далее) и в формуле неприменимо. Таким образом, MIDI рекомендует использовать для отдельного звука динамический диапазон в 84 дБ.

Многие тон-генераторы не следуют этой рекомендации буквально, ведь для большинства звуков такой огромный диапазон просто не нужен. Но все же зависимость, близкая к логарифмической, обычно выдерживается на каком-либо отрезке динамики. MIDI также рекомендует шкалу соответствия музыкальных динамических оттенков и Velocity, показанную на рис.



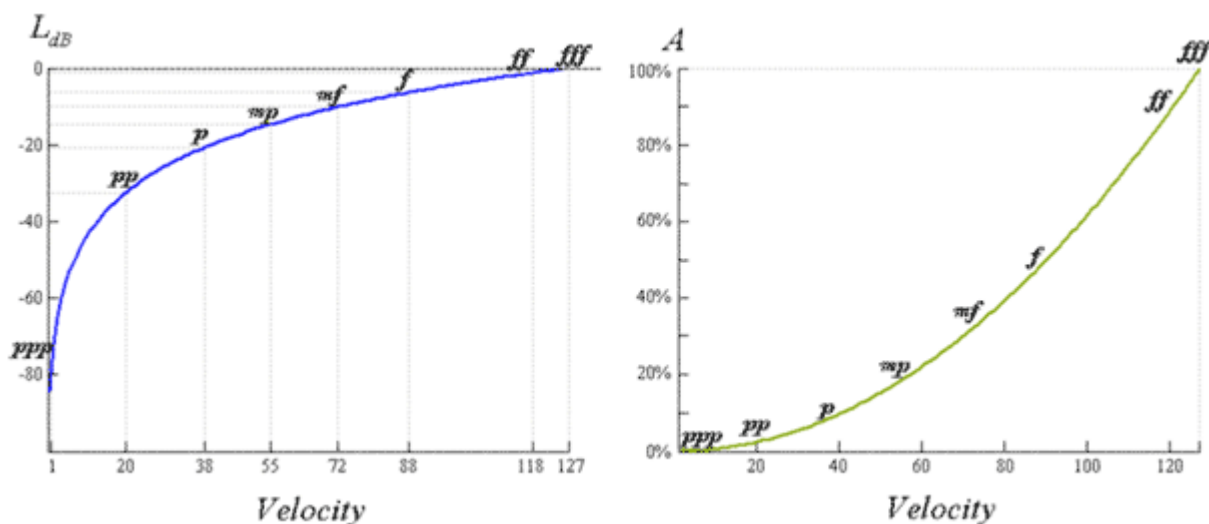
Dynamics' Note Velocity

Dynamic	Velocity*	Voice
<i>ppp</i>	16	Whispering
<i>pp</i>	33	Almost at a whisper
<i>p</i>	49	Softer than speaking voice
<i>mp</i>	64	] Speaking voice
<i>mf</i>	80	
<i>f</i>	96	Louder than speaking
<i>ff</i>	112	Speaking loud
<i>fff</i>	127	Yelling

		
Decrescendo (diminuendo)	Crescendo	Accent

*Note velocity adopted from Logic Pro

Обращает на себя внимание большой промежуток между forte (*f*, Velocity = 88) и fortissimo (*ff*, Velocity = 118). На следующих графиках (рис. 31) показана рекомендуемая логарифмическая зависимость: уровень в децибелах от динамики (левый график) и относительный уровень (в процентах) от динамики (правый график). Для удобства на графиках отмечены музыкальные динамические оттенки.



Отпускание клавиши

Сообщение **Note Off** дает команду снять ноту заданной высоты. Первый байт данных содержит MIDI-номер ноты, второй — динамику ее снятия. Снятие ноты заставляет тон-генератор перейти к стадии затухания (Release) огибающей. Таким образом, после снятия пальца с клавиши нота звучит еще ровно столько, сколько длится этот участок. Правда, если отпустить клавишу при нажатой педали Sustain (контроллер CC#64, Hold Pedal, см. далее), то звуковой генератор, хотя и получит сообщение Note Off, но участок Release не включит. Нота продолжит звучание на участке поддержки (Sustain) до тех пор, пока педаль не будет отпущена. При отпускании педали генератор, помня о том, что сообщение Note Off уже приходило, включит участок Release автоматически, и нота скоро прекратит звучание.

Послекасание

Послекасание (Aftertouch) — это давление, прикладываемое к клавише после ее нажатия. Послекасание позволяет имитировать приемы звукоизвлечения, несвойственные клавишному инструменту, например, вибрато или тремоло, причем довольно естественным и удобным способом — нажав клавишу, можно непрерывно изменять давление на нее.

Это давление измеряют сенсоры клавиатуры. Делать сенсор под каждую клавишу — дорогое удовольствие, поэтому большинство клавиатур имеют один сенсор, который измеряет общее давление. Исходя из этого, спецификация определяет два варианта послекасания: монофоническое (или послекасание канала, Channel Pressure) — это сообщение генерируют клавиатуры с одним-единственным сенсором, и полифоническое (Polyphonic Key Pressure) — для клавиатур с сенсором на каждую клавишу. Бывают клавиатуры, которые не генерируют послекасания вообще.

Понятно, что для изменений в звуке генератор должен понимать сообщения послекасания и как-то на них реагировать. Обычно изменяются уровень поддержки (Sustain) на огибающей громкости или фильтра, или глубина/скорость модуляции, производимой генератором низкой частоты. Спецификация не определяет конкретной реакции устройства на сообщения послекасания, и многие синтезаторы их игнорируют.

Послекасание канала менее информативно и не имеет такой гибкости и выразительности, как полифоническое. Так, если вы дожмете одну клавишу больше другой на устройстве, передающем послекасание канала, оба давления будут усреднены, словно вы дожали обе клавиши с одинаковой силой. Если в аккорде вы хотите выделить басовую ноту, ничего не выйдет — эффект будет применен ко всем нотам, звучащим на канале.

Обычно клавиатура посылает один вид послекасания. Но бывают клавиатуры (с сенсором на каждую клавишу), которые могут посылать оба вида сообщений сразу: полифоническое послекасание формируется как обычно, а послекасание канала вычисляется как среднее от полифонического по каждой клавише.

Послекасание — типичный непрерывный контроллер низкого разрешения, с характерной особенностью: оба типа сообщений довольно сильно загружают канал передачи данных. Давить на клавиши с постоянным усилием получается редко, а при любом изменении давления передается новая порция байт. Послекасание канала загружает канал передачи значительно меньше, чем полифоническое, так как производит меньше байт в единицу времени.

Звуковысотное колесо Звуковысотное колесо (Pitch Wheel) используется для плавного изменения высоты тона звучащей ноты. Процесс изменения высоты называется питч-бендом (Pitch Bend), а соответствующее MIDI-сообщение — Pitch Wheel Change (его часто и называют Pitch Bend). Сообщение об изменении высоты относится сразу ко всему MIDI-каналу, поэтому нельзя, например, сыграть на одном канале аккорд, а сдвинуть высоту тона только у его басовой ноты.

Pitch Wheel Change — сообщение высокого разрешения, то есть позволяет использовать 16384 значений изменения высоты, и относится к контроллерам непрерывного типа. При движении колеса высота ноты скачкообразно изменяется во времени, но благодаря маленьким шагам (равным 1/16384 диапазона изменения) эти скачки обычно незаметны, и создается ощущение плавного изменения высоты.

Каков диапазон изменения высоты тона? Спецификация оставляет этот вопрос на совести производителя тон-генератора. MIDI-клавиатура (или другое средство ввода) всегда передает значения от 0 до 16383 (0x3FFF), получаемые с колеса, а тон-генератор интерпретирует их так, как считает нужным. Так что нет никакой гарантии, что, соединив два синтезатора по MIDI и вращая звуковысотное колесо на

одном из них, мы получим на втором синтезаторе ту же самую величину изменения высоты нот. Обычно диапазон составляет ± 2 полутона от центрального значения, но его можно изменять с помощью PRN-параметра Pitch Bend Sensitivity, о чем поговорим далее. Большинство тон-генераторов позволяют устанавливать независимый диапазон для каждого канала, но есть и такие, в которых диапазон задается глобально, на все каналы, или фиксирован на значении ± 2 полутона. Попадаются инструменты, которые вообще не реагируют на сообщение Pitch Wheel Change.

Программы и банки

Сообщение о смене программы (Program Change) назначает программу определенному MIDI-каналу в тон-генераторе. Это сообщение может поступить в любой момент, в процессе исполнения или воспроизведения, и, таким образом, позволяет динамически менять одну звучащую программу на другую. Вообще говоря, разные типы MIDI-устройств могут реагировать на это сообщение по-разному. Ритм-машинка может посредством Program Change менять текущий паттерн, процессоры эффектов — переключать алгоритмы или пресеты.

В мультитембральном синтезаторе реакция на сообщение Program Change может отличаться, в зависимости от того, в каком режиме находится инструмент, и на каком канале сообщение пришло. В режиме перфоманса обычно меняется не программа, а целая комбинация (перфоманс).

Один из главных подводных камней, на который можно натолкнуться с этим сообщением, — нумерация программ в конкретном устройстве. В компьютерном мире все начинается с нуля, поэтому номера программ приходят в диапазоне от 0 до 127. И действительно, *внутри устройства* программы нумеруются с нуля. Но многие производители считают, что музыканты не в состоянии нормально воспринимать программу номер 0 (как, впрочем, и MIDI-канал номер 0), и поэтому нумеруют программы *для пользователя*, начиная с 1. Такие "адаптированные" номера могут указываться на передней панели или в руководстве к инструменту. Другие фирмы нуля не боятся, и первая программа у них, как и положено, имеет номер 0. Отсюда — гарантированная путаница при работе с разными инструментами. Хорошо, когда программы называются по имени, а если по номеру? В вышеприведенном примере программа номер 36 на "адаптированных" инструментах будет показана пользователю как 37.

Второй подводный камень — совместимость инструментов. На одном синтезаторе программе номер 5 может соответствовать гитара, на другом — флейта, на третьем — вообще набор ударных. Так что композиция, подготовленная на одном инструменте, на другом может звучать, мягко говоря, некорректно, что неприемлемо для обмена и распространения. С появлением в 1991 году спецификации General

MIDI эта проблема была отчасти решена — каждому номеру программы сопоставили свой характерный тембр. (см. стандарты MIDI)

Некоторые старые инструменты имеют меньше 128 программ. Что произойдет, если в инструменте, скажем, только 64 программы, а в сообщении Program Change пришел номер 110? Скорее всего, сообщение будет проигнорировано. В некоторых случаях такой номер может быть преобразован в один из доступных.

Большинство же современных тон-генераторов содержат более 128 программ (в некоторых устройствах число программ превышает 1000). Понятно, что выбрать программу, скажем, 512 с помощью одного сообщения Program Change нельзя. Поэтому сложилась практика организовывать программы в банки. Один банк может содержать, например, 128 программ, и тогда программа с номером 512 будет первой программой в пятом банке ($128 \times 4 = 512$).

Как ее выбрать? Очень просто — с передней панели инструмента. Сначала кнопкой выбирается нужный банк, а затем уже программа. А если то же самое нужно сделать посредством MIDI-сообщения? До 1990 года такого сообщения не было. Какие только способы не выдумывали производители! Все они больше напоминали ритуальные пляски.

Например, использовались карты переадресации. Синтезатор настраивался так, чтобы, скажем, при получении сообщения Program Change с номером 12 выбиралась внутренняя программа номер 344. Эта процедура выполнялась для всех тембров, используемых в композиции. Другой способ — скопировать нужные программы в пользовательский банк, выбрать его вручную с панели инструмента, после чего сообщение Program Change позволит использовать программы из этого банка. Некоторые производители применяли для переключения банков дополнительное MIDI-сообщение, какое именно — зависело от конкретного инструмента. Часто использовались сообщения SysEx.

Наконец в 1990 году организация MMA решила навести порядок и стандартизировать процесс выбора банка. Появилось сообщение высокого разрешения под названием Bank Select. Его поместили во второй уровень кодирования (первый был уже весь занят), к контроллерам, и разбили на два контроллера. Старшие 7 бит задаются контроллером CC#0, младшие — контроллером CC#32.

С помощью Bank Select можно получить доступ к одному из 16384 банков, а конкретную программу в этом банке выбрать как обычно — с помощью сообщения Program Change. Таким образом, если в каждом банке будет по 128 программ, то "сладкая парочка" Bank Select / Program Change позволит адресовать одну из 2 097 152 программ (16384×128). Но не стоит впадать в эйфорию от этого числа. Во-первых, ни один производитель пока не выпустил инструмента с таким количеством звуков (да и зачем?) Во-вторых, есть замечательные инструменты, выпущенные до

августа 1990 года, которые на сообщение Bank Select не реагируют. И выбор нужной программы для них по MIDI возможен только с помощью одного из вышеупомянутых "древних" способов.

Наконец, и это самое неприятное, спецификация MIDI не определяет точного механизма выбора звуков с помощью Bank Select. А именно: не определено количество доступных банков, число программ в каждом банке и принцип нумерации программ. Так что каждый производитель применяет собственные схемы, часто на основе сложившейся традиции.

Например, в некоторых инструментах банк содержит от 8 до 10 программ, в других — 16, 32, 64 или 128. Банки могут нумероваться буквами (А, В, С) или цифрами, арабскими или римскими. Нумерация программ в каждом банке может начинаться с 0 или с 1. В одном из старых вариантов номер программы состоял из двух цифр. Первая определяла банк, вторая — номер программы в банке. При этом в каждом банке было по восемь программ. Таким образом, первая программа первого банка имела номер 11, потом нумерация шла до 18, затем перепрыгивала на 21 (первая программа второго банка) и т. д. Буквенно-цифровая нумерация может выглядеть как B054, что означает программу номер 54 из банка В.

Некоторые производители предложили собственные стандарты, определяющие минимальные характеристики совместимых устройств. Частью этих стандартов является и схема организации программ в банках. Наиболее известные — GS (Roland) и XG (Yamaha), (см. стандарты MIDI)

При получении сообщения Bank Select тон-генератор не должен выполнять каких-либо действий. Полученный номер банка должен сохраняться на будущее. А затем, как только поступит сообщение Program Change, устройство выберет программу из банка с сохраненным номером. Некоторые устройства все же реагируют на Bank Select дополнительными действиями. Так, в семплерах это сообщение может служить командой для загрузки набора семплов с диска в оперативную память. В некоторых инструментах посредством сообщений Bank Select переключаются наборы ударных, что, вообще говоря, — нарушение спецификации.

Если нужно переключиться на другую программу внутри текущего банка на том же MIDI-канале, то достаточно одного сообщения Program Change, посылать Bank Select не надо. Если нужно выбрать программу из другого банка, то сначала посылается Bank Select, затем Program Change.

Control Change

Статус-байт-посредник получил название Control Change ("изменение управления"). В спецификации все сообщения, вызываемые через этот байт, называются сообщениями типа Control Change, и предполагается, что они "генерируются манипулятором, отличным от клавиатуры". Но, например, колесо питч-бенда — тоже манипулятор, отличный от клавиатуры, однако он к группе Control Change не относится. Музыканты настолько часто им пользуются, что его просто невыгодно помещать во второй уровень. А вот колесо модуляции используется пореже, и там ему самое место. Хотя, повторю, это все в среднем: иногда вы можете руку не снимать с колеса модуляции, а к колесу высоты тона даже не прикоснуться.

Таким образом, "сообщения об изменении управления" — название условное, а функция статус-байта Control Change служебная. Он просто действует как префикс, открывающий путь к другим сообщениям.

Получается такая картина. Если пришел статус байт вида 1100nnnn, то инструмент сразу по нему определяет тип сообщения, — в данном случае это Program Change. Если пришел статус-байт 1011nnnn, то инструмент видит, что это байт-посредник, который о типе сообщения ничего не говорит, а просто уведомляет: сейчас пойдет сообщение второго уровня. Какое именно — смотри байт данных, следующий за посредником. Синтезатор читает этот байт и определяет, что это сообщение колеса модуляции. Положение колеса будет передано в третьем, завершающем байте (см. рис. 7). Таким образом, с помощью статус-байта-посредника и следующего за ним байта данных можно закодировать 128 сообщений.

Эти сообщения показаны в таблице на рис. 5. Их принято называть сообщениями контроллеров, но правильнее было бы называть сообщениями второго уровня. Слово контроллер не очень удачно еще и потому, что под контроллером часто подразумевается физический орган управления — колесо, движок или педаль. Поэтому догадаться о том, что имеется в виду — сообщение типа Control Change или физическое устройство, иногда бывает очень сложно, даже по контексту. Но традиция сложилась, поэтому под словом "контроллер" я буду иметь в виду сообщение второго уровня. А физическое устройство буду называть органом управления.

Номер		Контроллер
Hex	Dec	
Сообщения канала		
00	0	Bank Select MSB
01	1	Modulation Wheel MSB
02	2	Breath Controller MSB
03	3	Undefined
04	4	Foot Controller MSB
05	5	Portamento Time MSB
06	6	Data Entry MSB
07	7	Channel Volume MSB
08	8	Balance MSB
09	9	Undefined
0A	10	Pan MSB
0B	11	Expression Controller MSB
0C..0D	12..13	Effect Control 1, 2 MSB
0E..0F	14..15	Undefined
10..13	16..19	General Purpose Controller 1, 2, 3, 4 MSB
14..1F	20..31	Undefined
20	32	Bank Select LSB
21	33	Modulation Wheel LSB
22	34	Breath Controller LSB
23	35	Undefined
24	36	Foot Controller LSB
25	37	Portamento Time LSB
26	38	Data Entry LSB
27	39	Channel Volume LSB
28	40	Balance LSB
29	41	Undefined
2A	42	Pan LSB
2B	43	Expression Controller LSB
2C..2D	44..45	Effect Control 1, 2 LSB
2E..2F	46..47	Undefined
30..33	48..51	General Purpose Controller 1, 2, 3, 4 LSB
34..3F	52..63	Undefined
40	64	Damper Pedal On/Off
41	65	Portamento On/Off
42	66	Sustenuto On/Off
43	67	Soft Pedal On/Off
44	68	Legato Footswitch
45	69	Hold 2
46..4F	70..79	Sound Controller 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
50..53	80..83	General Purpose Controller 5, 6, 7, 8
54	84	Portamento Control
55..5A	85..90	Undefined
5B..5E	91..95	Effect 1, 2, 3, 4, 5 Depth
60	96	Data Increment
61	97	Data Decrement
62	98	NRPN LSB
63	99	NRPN MSB
64	100	RPN LSB
65	101	PRN MSB
66..77	102..119	Undefined
Сообщения режима канала		
78	120	All Sound Off
79	121	Reset All Controllers
7A	122	Local Control On/Off
7B	123	All Notes Off
7C	124	Omni Mode Off
7D	125	Omni Mode On
7E	126	Poly Mode Off
7F	127	Poly Mode On

Идея такая: выделить специальное сообщение контроллера, в байте данных которого будет передаваться номер сообщения третьего уровня. Было решено задавать такой номер двумя байтами данных. На деле это означает, что для адресации должна использоваться пара MSB:LSB. Плюс ко всему, сообщения третьего уровня разбили на два типа: *зарегистрированные параметры* (RPN, Registered Parameter Numbers) и *незарегистрированные параметры* (NRPN, Non Registered Parameter Numbers). Первые кодируются с помощью пары CC#101:CC#100, вторые — CC#99:CC#98. Таким образом, третий уровень может содержать 16384 зарегистрированных параметров и 16384 незарегистрированных, то есть всего 32768 сообщений. Что ж, на первое время хватит...

Спецификация определяет назначение только для зарегистрированных параметров. В настоящий момент определены пять параметров, которые перечислены в таблице на рис. 6.

Номер RPN		Сообщение	CC#101 (MSB)		CC#100 (LSB)	
Hex	Dec		Hex	Dec	Hex	Dec
0000	0	Pitch Bend Sensitivity	00	0	00	0
0001	1	Channel Fine Tuning	00	0	01	1
0002	2	Channel Coarse Tuning	00	0	02	2
0003	3	Tuning Program Change	00	0	03	3
0004	4	Tuning Bank Select	00	0	04	4
3FFF	16383	RPN Reset	7F	127	7F	127

Незарегистрированные параметры предназначены для свободного использования. Каждый производитель может придумывать свои номера NRPN и использовать их так, как считает нужным. Фирмы Roland и Yamaha первыми приложили к ним руку, что отражено в стандартах GS и XG соответственно.

На рис. 7 показана схема передачи сообщений разного уровня, а на рис. 8 — требуемое количество байт.

Уровень кодирования	Статус-байт 1	Байт данных 1 2	Байт данных 2 3	Статус-байт 4	Байт данных 1 5	Байт данных 2 6	Статус-байт 7	Байт данных 1 8	Байт данных 2 9
1 Program Change	1100nnnn	00000001 номер программы							
2 Control Change	1011nnnn	00000001 номер контроллера	01001101 значение						
3 Control Change	1011nnnn	01100101 RPN MSB (CC#101)	00000000 MSB	1011nnnn Control Change	01100100 RPN LSB (CC#100)	00000001 LSB	1011nnnn Control Change	00000110 Data Entry (CC#6)	00101101 значение

Уровень	Число байт для описания типа сообщения	Общее число байт в одном сообщении*
1	1	2 .. 3
2	2	3 .. 6
3	6	9 .. 12

* - для сообщений 1-го уровня зависит от типа сообщения, для остальных сообщений - от того, какая настройка (грубая или тонкая) используется

Классификация сообщений

Сообщения можно разделить на три основные группы: непрерывные, контроллеры-переключатели и простые команды.

Непрерывные контроллеры (Continuous Controllers) — это такие контроллеры, которые обычно посылаются целой серией и представляют собой поток плавно изменяющихся значений параметра. Например, при перемещении колеса модуляции из положения 100 в положение 200 генерируется множество сообщений Modulation Wheel, каждое сообщение будет передавать текущее положение колеса (в данном случае 100, 101, 102 и т. д.). Поскольку отличие двух соседних значений небольшое, будет складываться ощущение действительно непрерывного изменения глубины модуляции.

Непрерывные контроллеры можно дополнительно разделить на контроллеры высокого и контроллеры низкого разрешения. К первому типу спецификация относит сообщения из диапазона CC#0-CC#63 и CC#98-CC#101 (то есть пары MSB:LSB), ко второму — из диапазона CC#70-CC#95, а также сообщение режима канала Poly Mode Off (CC#126).

На самом деле, часть этих "непрерывных" контроллеров используется для статичной настройки. Например, сообщение Bank Select (выбор банка), в отличие от Modulation Wheel, нет смысла посылать целой серией с последовательно увеличивающимися номерами банков. Обычно банк выбирается один раз, ну иногда меняется по ходу дела. Подобные сообщения правильнее было бы назвать "статичными".

В первом уровне кодирования к сообщениям непрерывного типа относятся два вида послекасания (низкое разрешение) и Pitch Wheel Change (высокое разрешение). Сообщения Note Off, Note On и Program Change можно отнести к статичным, а сообщение Control Change само по себе ни в какую группу не попадает, так как является посредником.

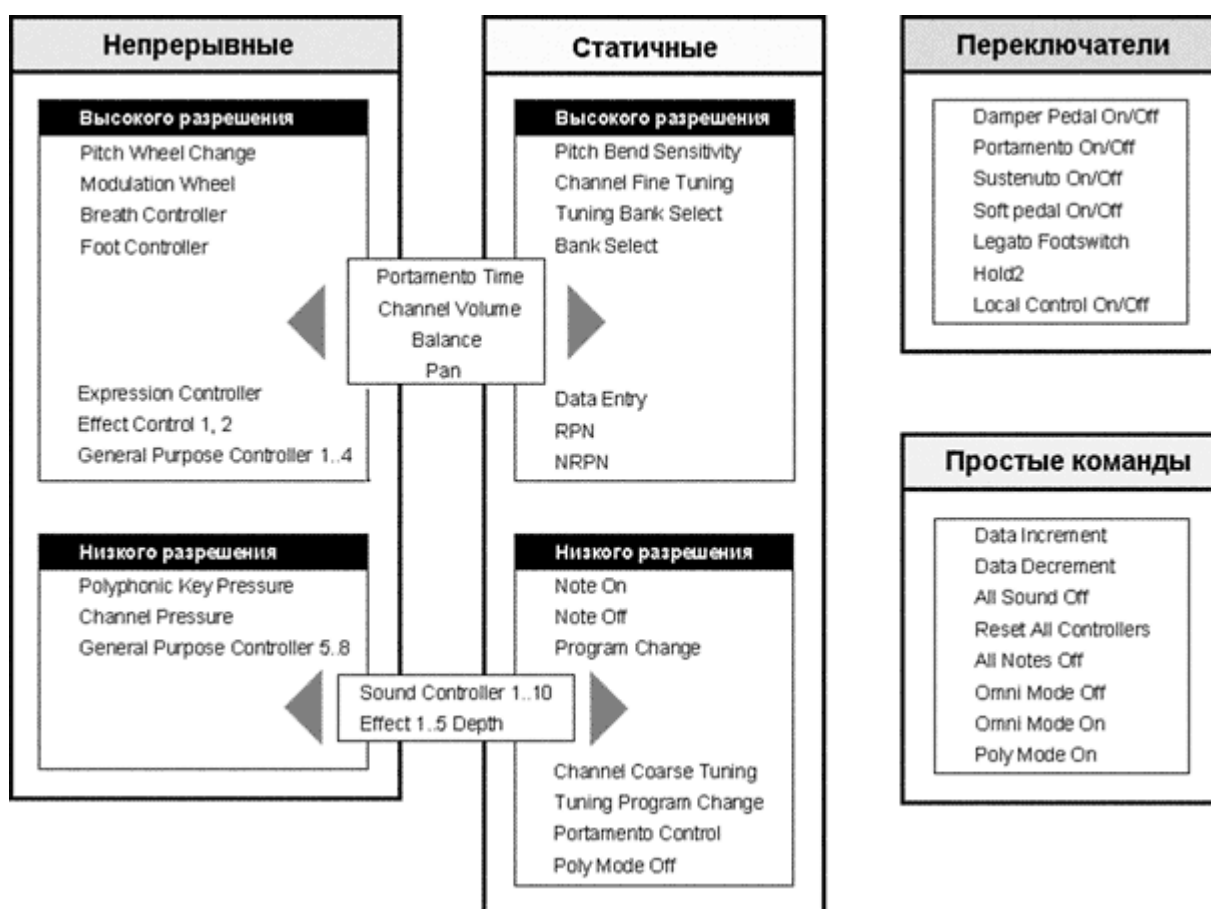
Все зарегистрированные параметры третьего уровня также относятся к статичным.

Контроллеры-переключатели (СС#64-СС#69) могут передавать только два значения: включено или выключено (On/Off). Протокол связывает их с педалями всевозможных типов. Сюда же относится сообщение режима канала Local Control On/Off (СС#122).

Простые команды вообще не передают никакой информации, кроме самой команды. Например, сообщение режима канала All Notes Off (СС#123) просит тон-генератор выключить звучащие ноты. И только.

К простым командам относятся контроллеры СС#96, СС#97, СС#120, СС#121, СС#123, СС#124, СС#125, СС#127.

Рис. 9 подытоживает вышесказанное. Некоторые сообщения (вроде панорамы) однозначно отнести к непрерывным или статичным сложно. Например, панорама (Pan) может выставляться один раз в начале работы, а может меняться непрерывно, для создания эффекта перемещения по стереобазе.



Генерирование сообщений

В названии многих сообщений отражен физический орган управления. Но это не значит, что, например, сообщение Pitch Wheel Change можно генерировать только с помощью звуковысотного колеса. Блоку синтеза внутри тон-генератора абсолютно все равно, какой орган управления используется. Важно то, какой смысл несет в себе сообщение и на какой параметр синтеза влияет. В данном случае изменяется высота тона.

Если вы управляете тон-генератором из секвенсора, то вопрос о генерировании сообщений вообще становится второстепенным. Главное — они записаны и передаются на выход секвенсора при нажатии кнопки Play. Для записи сообщения Pitch Wheel Change вы могли крутить реальное колесо, могли нарисовать график с помощью мыши, а могли вводить в память устройства "голые" цифры.

Современные синтезаторы позволяют управлять массой параметров, но количество органов управления на передней панели, как правило, ограничено. Так что, например, один и тот же слайдер может использоваться для управления разными функциями. Если в данный момент нужно управлять частотой среза фильтра, вы назначаете движок на передачу соответствующего сообщения. В другой ситуации вы можете использовать тот же движок для управления громкостью или панорамой. Одним словом, синтезаторы позволяют назначать органы управления на передачу любых сообщений. Если вам удобнее менять глубину модуляции с помощью фейдера, — пожалуйста, назначьте на фейдер сообщение CC#1 (Modulation Wheel).

Естественно, сообщения должны быть *совместимы по типу*. Так, если педаль представляет собой контроллер-переключатель, то есть имеет только два состояния "нажата/отпущена", то чаще всего нет смысла назначать ее для управления непрерывным сообщением вроде модуляции.

Кроме того, существует возможность *переназначения контроллеров*. Например, секвенсор можно настроить так, чтобы при приеме сообщений Modulation Wheel они преобразовывались в сообщения Pan. Тогда при вращении колеса модуляции будут записаны сообщения об изменении панорамы. Подобным образом можно настроить и многие тон-генераторы. Например, сообщения послекасаания, поступающие на MIDI-вход устройства, могут преобразовываться в сообщения модуляции. Переназначение можно выполнять и на стороне передатчика, когда генерируется сообщение одного типа, но прежде чем попасть на MIDI-выход, оно преобразуется в сообщение другого типа. Одним словом, переназначение дает дополнительную гибкость в управлении MIDI-системой.

Обработка сообщений

Сообщение может поступить напрямую с органа управления, из секвенсора, образоваться как результат переназначения — это не так существенно. После того, как оно поступило в звукообразующую часть тон-генератора, оно должно использоваться по прямому назначению. Так, например, сообщение контроллера CC#10 должно приводить к изменению панорамы, а не высоты тона. В случаях, когда действие контроллера четко не определено, производителю дается свобода выбора. Например, сообщение Foot Pedal может управлять темпом ритм-машинки. Для реализации функций, которых нет среди стандартных сообщений, могут использоваться сообщения типа General Purpose, NRPN или SysEx.

Естественно, не каждое устройство реагирует на все определенные в протоколе сообщения. Обычно в руководстве к устройству содержится так называемая карта MIDI-реализации (MIDI Implementation Chart), которая наглядно показывает, на какие сообщения устройство реагирует, какие может генерировать и т. д. Более подробно о картах реализации поговорим в статье, посвященной устройствам.

Большинство контроллеров действуют на уже звучащие и последующие ноты, то есть эффект от их использования слышен немедленно. Сообщения контроллеров для одного MIDI-канала не затрагивают другой, поэтому в мультитембральном устройстве каждая программа, звучащая на своем канале, может иметь собственные настройки контроллеров.

Кроме того, инструмент запоминает однажды выставленное значение контроллера и использует его до тех пор, пока оно явно не будет изменено или не произойдет принудительный сброс посредством специальных команд. Так, если сдвинуть колесо модуляции из нулевого положения, то модуляция будет применяться к звукам до тех пор, пока не повернуть колесо обратно. Со звуковысотным колесом проще — при отпускании оно обычно возвращается в исходное состояние автоматически.

Тот факт, что устройство "помнит" ранее выставленный контроллер, позволяет избежать многократной передачи пары MSB:LSB. Если происходит постоянная тонкая подстройка в узком диапазоне, то можно один раз передать MSB, а затем — серию из нескольких LSB. Это и удобнее, и время передачи уменьшается.

Многие устройства не реагируют на тонкую подстройку для большинства контроллеров. Это вполне нормально, если производитель считает, что повышенная точность в данном сообщении не нужна. Например, в контроллерах громкости и экспрессии обычно хватает 128 уровней.

Состояние контроллеров-переключателей передается в одном байте, а, следовательно, допускает 128 значений. Возникает вопрос — что считать за On, а

что за Off? Спецификация требует, чтобы значения от 0 до 63 рассматривались как Off (выключено), а значения от 64 до 127 — как On (включено). Для совместимости со старыми инструментами рекомендуется всегда посылать Off как 0, а On как 127.

Контроллеры, относящиеся к простым командам, не передают дополнительной информации, и третий байт в сообщении должен быть всегда равен нулю. Приемное устройство должно игнорировать этот байт, даже если по какой-то причине он содержит значение, отличное от нуля.

Сообщения режима канала

При разработке протокола MIDI упор делался на монотембральные устройства. Такие устройства в каждый момент времени воспроизводят только одну партию, а, следовательно, принимают и передают данные только по одному MIDI-каналу. Для получения рабочего ансамбля, например, баса, рояля и скрипки, нужно иметь три монотембральных синтезатора, первый из которых будет производить партию баса, второй — рояля, а третий, соответственно, скрипки.

Большинство современных инструментов — мультитембральные. Напомню, что мультитембральность измеряется в "частях". Вообще-то с английского языка слово "part" можно перевести и как "партия" (музыкальная), но в русском языке сочетание "однопартийный синтезатор" будет вызывать ненужные ассоциации. Когда про синтезатор говорят, что он имеет мультитембральность 16 частей, то это значит, что он может принимать данные по 16 MIDI-каналам, и на каждом канале партия может звучать своим тембром.

Однако начнем разговор о режимах применительно к монотембральным устройствам, как того и требует спецификация.

В монотембральном тон-генераторе существует так называемый *основной канал* (Basic Channel). Это тот канал, по которому ведется прием или передача сообщений. Так, если основной канал имеет номер 3, то сообщения для пятого канала, поступающие на вход устройства, будут проигнорированы. Обычно основной канал можно выбирать произвольно, но в некоторых устройствах он может быть жестко определен — "зашит". Такое устройство довольно сложно интегрировать в MIDI-систему. Например, если на первом инструменте зашитый основной канал имеет номер 1, а на втором — 2, то устройства просто бесполезно соединять MIDI-кабелем. Первое будет всегда передавать по каналу 1, а второе — ожидать сообщений по каналу 2. Разговор слепого с глухим...

Если все же соединить инструменты необходимо, можно включить между ними так называемый MIDI-процессор. Он представляет собой небольшую коробочку, которая занимается преобразованием MIDI-сообщений в соответствии с заданным

алгоритмом. Можно настроить процессор таким образом, чтобы все сообщения первого канала преобразовывались в сообщения второго.

Если основной канал можно задавать произвольно, то обычно это делается с передней панели инструмента. Некоторые устройства позволяют выбирать основной канал посредством сообщений SysEx.

Спецификация определяет восемь сообщений режима канала (CC#120-CC#127). Однако непосредственно MIDI-режимами управляют только четыре последних. Остальные заведуют функциями сброса и локальным управлением. С него и начнем.

Режимы MIDI

Многие музыканты, использующие протокол MIDI, считают его режимы кошмаром или, по крайней мере, бесполезным усложнением. Многие технические специалисты считают эти режимы нерациональными, а многие производители устройств реализуют какой-либо один режим, игнорируя остальные. Трудно сказать, почему так сложилось: может быть, из-за не очень ясных намерений разработчиков и недостаточно доходчивого изложения режимов в самой спецификации, или потому, что большинству музыкантов они просто не нужны. Во всяком случае, "негатив" укрепился, особенно после того, как возросло число мультитембральных устройств.

Однако при всем при этом некоторые MIDI-режимы могут использоваться для интересных творческих решений, поэтому полезно иметь о них хотя бы общее представление.

Так что же такое MIDI-режим? Если кратко, то это схема распределения голосов тон-генератора между MIDI-каналами и нотами. Под голосом имеется в виду простейшая единица синтеза, минимальная часть ресурсов звукообразующей части, которая требуется для воспроизведения одной ноты. Производители называют эту единицу по-разному: тон, голос, элемент, модуль и т. п. В аналоговом синтезаторе это может быть отдельный электронный блок, в цифровом — тайм-слот DSP.

Напомню, что тон-генераторы могут быть монофоническими, то есть способными в каждый момент производить только одну ноту, и полифоническими, то есть способными производить одновременно две ноты и более. Полифония современных инструментов находится в широких пределах, от 2 до 128 и даже более голосов. Теоретически, такое количество нот устройство может производить одновременно. Однако зачастую в пэтчах со сложной структурой для генерирования одной ноты используются два, три и даже больше голосов. В таких случаях реальная полифония гораздо меньше.

Начнем рассмотрение режимов применительно к монотембральному устройству.

Сообщения контроллеров из диапазона CC#124-CC#127 задают определенные правила работы устройства при приеме и передаче сообщений. Устройство должно реагировать на эти команды только в том случае, если они приходят по основному каналу. При получении любой из команд все звучащие ноты, поступившие ранее на MIDI-вход устройства, должны быть выключены. Фактически, при этом выполняется неявный вызов команды All Notes Off.

Слово **Omni** происходит от латинского "Omnis", означающего "все". Команда **Omni Mode On** (CC#125) заставляет тон-генератор принимать голосовые сообщения по всем MIDI-каналам, независимо от того, на какой основной канал он настроен. Команда **Omni Mode Off** (CC#124) возвращает к приему по основному каналу. Команда **Poly Mode Off** (CC#126) означает, что тон-генератор должен воспроизводить поступающие ноты монофонически. То есть, в каждый момент времени должна воспроизводиться только одна нота — последующая нота снимает предыдущую. В этой команде передается дополнительный параметр, который имеет смысл только в режиме 4 (см. далее). Команда **Poly Mode On** (CC#127) означает, что тон-генератор должен воспроизводить поступающие ноты полифонически — количество одновременно звучащих нот ограничивается только доступной полифонией.

MIDI-режимы образованы сочетанием этих команд (рис. 14a). Всего есть четыре режима, а наиболее распространенные из них имеют собственное имя (рис. 14b).

Номер режима	Имя	Команды
1	OMNI	Omni On / Poly On
2	—	Omni On / Poly Off
3	POLY	Omni Off / Poly On
4	MONO	Omni Off / Poly Off

Текущий режим может быть выбран с помощью органов управления на передней панели устройства, либо посредством комбинации команд, приведенных ранее. Так, например, устройство переходит в режим 1 после получения комбинации команд Omni On / Poly On.

MIDI-режимы могут быть как у принимающего, так и у передающего устройства. Скоммутированные друг с другом приемник и передатчик, как правило, должны находиться в одном и том же режиме, в противном случае возможна непредсказуемая реакция на голосовые сообщения.

Многие устройства поддерживают не все режимы. При получении команд, требующих перейти в режим, который не поддерживается, устройство может либо проигнорировать их, либо перейти в режим, наиболее близкий по смыслу.

Режим 1 (Omni)

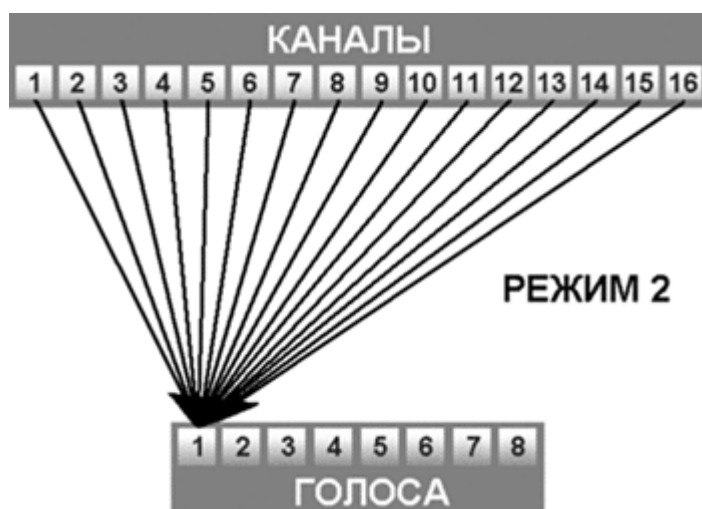
В этом режиме устройство реагирует на каждое поступившее голосовое сообщение, независимо от того, по какому каналу оно пришло. По сути, это отказ от системы MIDI-каналов как таковой — младшая половина статус-байта, содержащая номер канала, в режиме Omni просто игнорируется. При этом инструмент отвечает на сообщения полифонически (рис. 15).



Если в этом режиме находится передающее устройство, то оно должно посылать все голосовые сообщения по основному каналу, при этом количество одновременно посылаемых нот (полифония) не ограничивается. Для монотембрального устройства такое поведение естественно.

Режим 2

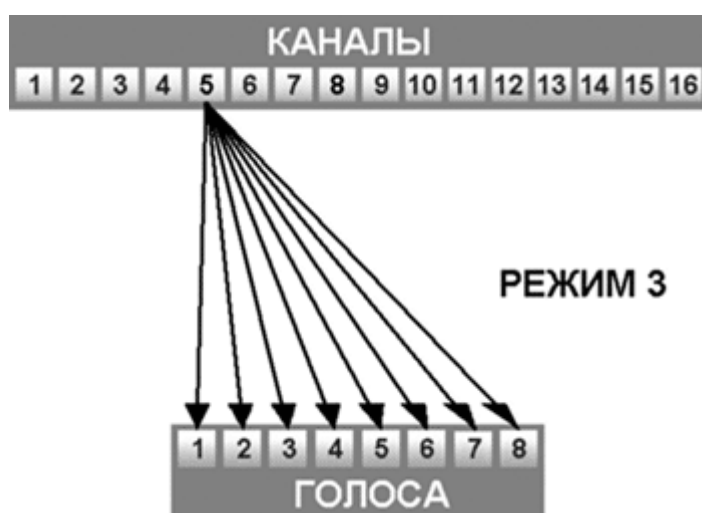
Этот режим настолько бесполезен, что даже не имеет собственного имени. Устройство переходит в него после получения комбинации сообщений Omni On / Poly Off. MIDI-спецификация описывает режим 2 так: "Голосовые сообщения принимаются по всем каналам, но в каждый момент времени звучит только одна нота". Вы можете представить, зачем это нужно? Я тоже с трудом. Единственный вариант, который приходит на ум, — это соединение в цепь нескольких монофонических синтезаторов. При этом не нужно заморачиваться с настройкой основного канала на каждом — они будут реагировать на сообщение по любому каналу. Рис. 17 отражает работу приемного устройства.



Если в режиме 2 находится передающее устройство, то оно должно посылать все голосовые сообщения по основному каналу, при этом ноты могут посылаться по одной в каждый момент времени, то есть монофонически, а могут и одновременно (полифонически).

Режим 3 (Poly)

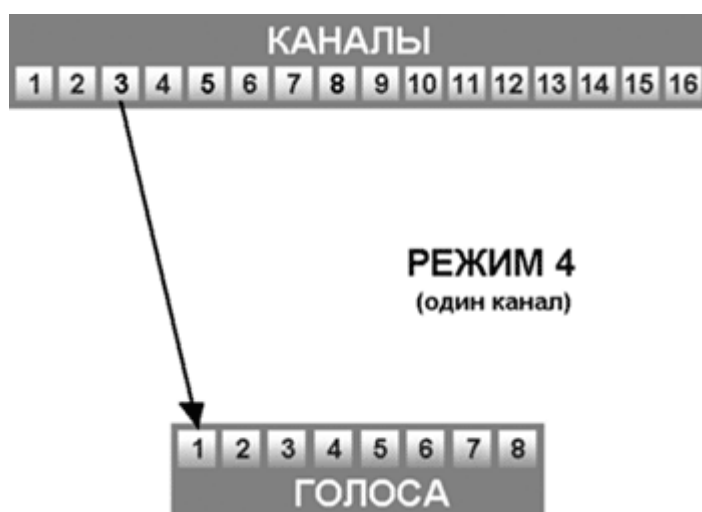
Этот режим является самым приемлемым для полифонического монотембрального устройства. Прием ведется только по одному каналу — тому, который является основным. Поступающие сообщения распределяются между доступными голосами, то есть инструмент отвечает на сообщения полифонически (рис. 18). Передающее устройство работает здесь так же, как и в режиме Omni.



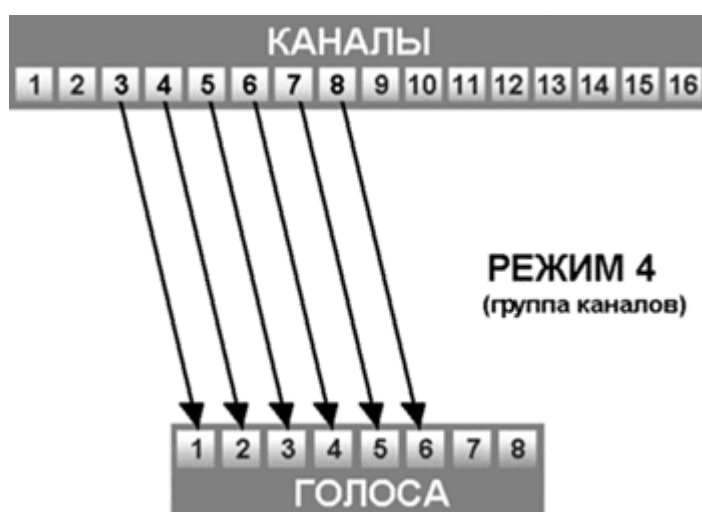
В режиме Poly, и только в нем, система MIDI-каналов используется по своему прямому назначению. Устройства можно соединять в цепь, при этом каждое будет реагировать только на сообщение для одного канала и игнорировать другие. Несколько монотембральных тон-генераторов в этом режиме могут исполнять мультитембральную аранжировку, передаваемую секвенсором.

Режим 4 (Mono)

Устройство переходит в этот режим после получения комбинации сообщений Omni Off / Poly Off. Поведение устройства здесь зависит от параметра, переданного в сообщении Poly Off. Если параметр в сообщении имеет значение от 1 до 16, то он задает число каналов, на которые будет реагировать инструмент, начиная с основного канала. Если параметр равен 1, то это, по сути, означает монотембральный монофонический инструмент. Так, если основной канал у нас третий, то прием будет вестись только по нему; следовательно, доступен только один тембр, с одним голосом. Рисунок 19 иллюстрирует это.



Однако самое интересное начинается, когда параметр сообщения Poly Off больше 1. Если, например, он равен 6, а основной канал — третий, то инструмент будет принимать сообщения на каналах 3, 4, 5, 6, 7 и 8 мультитембрально и монофонически (рис. 20).



Теперь, если назначить на каждый из этих каналов один и тот же тембр, то мы получим систему, идеально подходящую для имитации гитары и других струнных инструментов. Каждая струна гитары будет звучать на своем MIDI-канале! Это

позволит обойти две проблемы. Во-первых, гитара позволяет делать подтяжку на одной струне, не смещая высоты тона на других струнах. Если поместить всю партию гитары на один MIDI-канал, то мы не сможем делать подтяжку выборочно, на каких-либо нотах, — сообщение звуковысотного колеса, как и все сообщения контроллеров, действует сразу на все ноты канала. Это касается и других приемов игры, таких как глиссандо и легато.

Во-вторых, гитара имеет возможность играть две одинаковых ноты в один и тот же момент на разных струнах. При назначении каждой струне собственного канала можно в любой момент времени получить унисон независимо от того, насколько гибко тон-генератор обращается с двойными нотами на одном канале.

В-третьих, струны гитары ведут себя монофонически, то есть при защипывании струны новый звук немедленно гасит старый. Режим 4 позволяет об этом не беспокоиться — выключение нот произойдет автоматически, так как для каждого канала разрешен одновременно только один голос. Ну и, наконец, для каждой струны можно задать свой тембр — получится гитара, не имеющая аналогов в реальной жизни.

Модуляция

Сообщение Modulation (CC#1/CC#33) задает глубину частотной модуляции (вибрато) и управляет соответствующим LFO-блоком тон-генератора. В синтезаторах управление модуляцией происходит обычно с помощью колеса, либо рычага на передней панели. Диапазон, в котором изменяется глубина модуляции, спецификация оставляет на усмотрение тон-генератора.

Громкость и экспрессия

Итак, параметра Velocity для выстраивания баланса нескольких партий недостаточно. На помощь приходит сообщение MIDI Volume (контроллер CC#7/CC#39), которое задает громкость звучания на MIDI-канале. Его действие можно сравнить с фейдером канала в микшерском пульте. С какой бы динамикой не извлекал звуки исполнитель, уровень партии в миксе будет зависеть от положения фейдера. Говоря языком MIDI — какая бы скорость нажатия не использовалась, выходная громкость партии будет зависеть от значения MIDI Volume.

В современных тон-генераторах тембр при увеличении динамики, как правило, изменяется (становится ярче). Применяя совместно Velocity и MIDI Volume, можно получить любые сочетания "яркость/громкость". Например, максимальное значение Velocity при небольшой громкости канала создает впечатление очень яркого по тембру, но тихого инструмента.

В мультитембральном тон-генераторе после отстройки относительного баланса партий (с помощью сообщения MIDI Volume) может потребоваться регулировка общей громкости устройства (уровня мастер-выхода). Для этого чаще всего используется сообщение SysEx Master Volume (поговорим в следующей статье), иногда — значение громкости одного из каналов или контроллера типа General Purpose.

Большинство устройств игнорируют LSB в сообщении MIDI Volume и позволяют производить только грубую настройку (128 уровней), посредством MSB (контроллер CC#7). Некоторые старые синтезаторы на сообщение MIDI Volume не реагируют.

Бок о бок с сообщением громкости канала идет сообщение экспрессии (Expression, CC#11 / CC#43). Его название может сбить с толку, поскольку прямой перевод с английского (expression — выразительность) — довольно расплывчатое понятие. Многие в этой связи склонны ожидать от контроллера экспрессии какую-либо тонкую настройку пэтча (например, посредством фильтров) или других чудес. На самом деле все гораздо проще.

Слово Expression как музыкальный термин соответствует понятию "нюанс" русской музыкальной терминологии, которое означает, прежде всего, динамический нюанс исполнения — форте, пиано и т. д. Таким образом, контроллер экспрессии предназначен для дополнительного управления громкостью MIDI-канала. Как и в случае с сообщением MIDI Volume, большинство тон-генераторов допускают только грубую настройку экспрессии (128 уровней) посредством MSB (контроллер CC#11).

При максимальном значении экспрессии громкость канала равна текущему значению MIDI Volume. При уменьшении экспрессии начинается уменьшение результирующей громкости. Когда экспрессия равна нулю, громкость канала также равна нулю; когда экспрессии "половина", то и громкость канала равна половине от текущего значения MIDI Volume.

Гораздо проще управлять динамикой внутри партии с помощью сообщения экспрессии, а с помощью MIDI Volume балансировать партию со всеми остальными. Тогда в нашем случае задача упрощается: нужно только увеличить общую громкость канала, а кривая нарастания останется неизменной.

Исходя из этого, протокол MIDI рекомендует использовать сообщение MIDI Volume только для общей балансировки партий, один раз в самом начале аранжировки, а динамические изменения в процессе звучания производить с помощью сообщения экспрессии.

Панорама

Сообщение Pan (контроллеры CC#10/CC#42) устанавливает панораму на MIDI-

канале, то есть определяет положение партии в стереобазе. Сообщение имеет смысл только для тех тон-генераторов, которые имеют стереовыход или несколько выходов, работающих как стереопары.

В зависимости от конкретной архитектуры тон-генератора это сообщение может быть регулятором панорамы или баланса. Например, если структура пэтча такова, что в блок VCA инструмента приходит моносигнал, то под воздействием сообщения Pan синтезатор просто делит моносигнал на левый и правый канал в нужной пропорции, и с выхода блока снимается псевдостереосигнал. Но во многих тон-генераторах используются стереосемплы (или задействованы два осциллятора, или несколько слоев с установленным положением слоя в стереобазе на уровне пэтча). Так как в этом случае сигнал приходит на вход VCA уже стереофоническим, то сообщение будет работать как регулятор баланса, то есть задавать относительный уровень левого и правого каналов.

Зависимость уровней левого и правого каналов от значения панорамы

$$L_{dBLeft} = 20 \lg \sqrt{1 - \frac{P}{127}}$$

$$L_{dBRight} = 20 \lg \sqrt{\frac{P}{127}} ,$$

где P — значение Pan (CC#10).

Синтез

В каждом синтезаторе применяется своя схема синтеза, своя конструкция компонентов (осцилляторов, фильтров, усилителей и т. д.), свои варианты их использования. Причем в продукции одной и той же фирмы эти варианты часто отличаются. Поэтому спецификация определяет только самые общие, наиболее характерные для большинства тон-генераторов параметры. Например, управление длительностью фаз в генераторе огибающей. Если предложенные сообщения производителя не удовлетворяют, он всегда может использовать SysEx или NRPN.

Системные сообщения

Системные сообщения немногочисленны, но берут на себя едва ли не основную нагрузку, когда речь идет о тонком управлении устройством, сохранении и восстановлении настроек, передачи специализированной информации или объединении нескольких устройств в MIDI-систему для совместной работы.

Название сообщений, их статус-байты и формат передачи показаны в таблице на рис. 1. Как видно, большая часть сообщений предназначена для синхронизации и управления воспроизведением. Но, как показывает практика, наиболее активно используются системные эксклюзивные сообщения (SysEx), которые скрываются за одним-единственным статус-байтом 0xF0. Они имеют сложную структуру и произвольную длину.

Статус-байт			Сообщение	Байт данных 1	Байт данных 2
Hex	Bin	Dec			
System Exclusive (SysEx)					
F0	11110000	240	System Exclusive	ID	...
System Common					
F1	11110001	241	MTC Quarter Frame	тайм-код	—
F2	11110010	242	Song Position Pointer	LSB	MSB
F3	11110011	243	Song Select	номер песни	—
F4	11110100	244	Undefined		
F5	11110101	245	Undefined		
F6	11110110	246	Tune Request	—	—
F7	11110111	247	End Of Exclusive (EOX)	—	—
System Real Time					
F8	11111000	248	Timing Clock	—	—
F9	11111001	249	Timing Tick	—	—
FA	11111010	250	Start	—	—
FB	11111011	251	Continue	—	—
FC	11111100	252	Stop	—	—
FD	11111101	253	Undefined		
FE	11111110	254	Active Sensing	—	—
FF	11111111	255	System Reset	—	—

К общесистемной категории (System Common) относятся пять разнородных сообщений. Первое из них, MTC Quarter Frame, появилось в спецификации в 1987 году вместе с первым расширением протокола (MIDI Time Code). Последующие расширения MIDI стали кодироваться в группе SysEx. О другом общесистемном сообщении, **Tune Request** (статус-байт 0xF6), расскажу сразу же. Оно дает команду синтезатору произвести подстройку звуковых генераторов и, очевидно, имеет смысл только для старых аналоговых инструментов, у которых через некоторое время после включения начинает "съезжать" строй. Естественно, на это сообщение реагируют только те инструменты, у которых есть схема автоподстройки.

Сообщения категории Real Time имеют в MIDI наивысший приоритет, могут вклиниваться между частями других сообщений и состоят из одного статус-байта, без байтов данных. Они применяются для синхронизации и других задач, критичных ко времени.

Все три типа системных сообщений имеют общую черту — они относятся не к отдельному каналу, а ко всем устройствам MIDI-системы (впрочем, сообщения SysEx можно передавать избирательно, но об этом позже).

Синхронизация MIDI-устройств

До тех пор, пока MIDI-система состоит из синтезаторов, звуковых модулей и одного центра управления — секвенсора (программного или аппаратного), вопрос о синхронизации не стоит. И действительно, что тут и с чем синхронизировать? Секвенсор сам себе хозяин, работает от внутренних часов и при нажатии кнопки Play передает устройствам команды управления.

Другое дело, когда в системе появляется источник звука, содержащий встроенный секвенсор (на языке MIDI-спецификации — Rhythm Unit). Чаще всего это ритм-машинка, иногда — рабочая станция, то есть синтезатор с секвенсором в одном корпусе. Эти устройства можно использовать просто как источники звука, записывая всю управляющую информацию в основной секвенсор. Но иногда их собственный секвенсор и записывающие/ воспроизводящие возможности приходятся очень кстати.

Например, можно "набросать" паттерн в ритм-машинке, запустить его на воспроизведение, а на клавиатуре подбирать партию баса, записывая ее в основной секвенсор. Тонкостей этого процесса мы коснемся позже, в статье про организацию MIDI-систем, но уже ясно: в системе одновременно работают два секвенсора, то есть два устройства, имеющие внутренние часы. Нужно сделать так, чтобы устройства стартовали одновременно и работали в одном темпе, то есть синхронизировать их. Очевидно, одни из часов придется выключить, а вторые сделать общими. Иначе говоря, одно из устройств будет мастером (ведущим), генерирующим сигналы "точного времени", а другое — подчиненным (ведомым), слушающим эти сигналы и работающим с мастером "в такт".

Задолго до появления MIDI применялся сигнал, который представлял собой последовательность обычных импульсов. Они передавались определенное число раз за четвертную ноту, например, 48 или 96. Так как музыкальные длительности относительны, то эти импульсы давали не абсолютное (в минутах и секундах), а относительное время. При повышении темпа частота следования импульсов повышалась, при понижении — понижалась. Этого достаточно, чтобы секвенсоры и ритм-машинки играли в одном темпе, абсолютное время им и не важно. Все бы

хорошо, но в разных устройствах применялись разные частоты следования импульсов, поэтому иногда требовались дополнительные преобразователи.

Проблема иного рода состоит в синхронизации секвенсоров и ритм-машинок с магнитофонами. Последние представляют собой устройства другой природы, где скорость движения ленты от темпа произведения не зависит. Непосредственно записать импульсы синхронизации на ленту нельзя, поэтому применяется один из двух методов: FSK-преобразование или таймкод SMPTE. Использование FSK будет рассмотрено в статье про построение MIDI-систем. Что же касается SMPTE, то на его базе был разработан протокол MIDI Time Code (MTC), который сегодня является основным при синхронизации секвенсоров с магнитофонами и другим звуковым оборудованием. Но MIDI Time Code — это уже расширение спецификации, поэтому о нем поговорим чуть позже. А сначала остановимся на "родном" методе синхронизации MIDI-устройств, который называется MIDI Clock и является неотъемлемой частью протокола.

MIDI Clock

MIDI Clock предназначен для синхронизации тех устройств, внутренняя временная сетка которых выражена в относительных единицах, зависящих от темпа. Типичные варианты: секвенсор и ритм-машинка, два секвенсора, две ритм-машинки. Для синхронизации одно из устройств делается мастером (у него включается функция передачи сообщений MIDI Clock), а другое — подчиненным (переводится в режим внешней синхронизации, который может иметь название "Sync: External Clock", "Sync: MIDI", "Clock: External" и т. п.). В принципе, ведомых устройств может быть несколько.

При этом не требуются специальные разъемы и кабели — сообщения MIDI Clock передаются по обычному MIDI-кабелю и разъему. Причем их вполне можно передавать и одновременно с аранжировкой. Здесь прослеживается аналогия с цифровой звуковой коммутацией, когда синхросигнал внедряется в звуковые данные (self-clock). С инженерной точки зрения это нехорошо, поскольку пропускная способность MIDI и без того ограничена. Но для пользователя удобно — не нужно лишней коммутации и обязательного наличия свободного MIDI-входа (рис. 2).

Для синхронизации по MIDI Clock применяются пять сообщений: позиционное (Song Position Pointer), информация о скорости (Timing Clock) и три команды (Start, Stop, Continue). Позиционное сообщение относится к категории System Common, информация о скорости и команды — к категории Real Time.

Основное сообщение здесь — **Timing Clock**. Оно состоит только из статус-байта (0xF8) и посылается мастер-устройством 24 раза за четверть. Если, например, текущий темп равен 120 bpm (ударами в минуту), то генерируются 48 сообщений

Timing Clock в секунду ($120 / 60 * 24$), если 180 bpm — 72 сообщения. Поскольку Timing Clock никакой числовой информации не несет, то для определения темпа приемное устройство должно получить как минимум два "клока" и измерить временной интервал между ними.

Известно, что разрешение секвенсоров гораздо больше, чем 24 импульса в четверти (у современных программных секвенсоров оно доходит до 15360 PPQN — Pulse Per Quarter Note). При передаче клона ведомому устройству секвенсор вынужден отбрасывать часть своих внутренних импульсов высокого разрешения. Приемное устройство также может иметь более высокое разрешение, и должно будет добавлять внутренние импульсы между приходящими клоками. Возможна ситуация, когда в мастер-устройстве произойдет плавное изменение темпа в промежутке между MIDI-клоками. Точно его передать на ведомое устройство не получится — в ведомом произойдет скачок темпа (рис. 3). Обычно это не представляет особой проблемы, но лучше такое ограничение иметь в виду.

Спецификация предлагает еще один способ передачи информации о скорости — так называемые "тики" (**Timing Tick**). Один тик — это статус-байт 0xF9, который передается от мастера к ведомому устройству каждые 10 мс (то есть 100 раз в секунду). Скорость этого потока не зависит от темпа, но, с другой стороны, и не содержит абсолютного времени, как, например, в МТС. Ведомое устройство должно самостоятельно определить, сколько тиков приходится на четверть, базирясь на своих внутренних часах.

Стандарты MIDI

В начале существования интерфейса MIDI у каждого изготовителя были собственные методики организации звуков и каналов. Некоторые устройства использовали один банк данных из 128 сэмплов, другие делили их на кусочки поменьше. Звуки пианино и клавишных могли храниться рядом или быть случайным образом разбросаны по всей памяти устройства. Это вызывало постоянную путаницу, а композиторам, сочинявшим музыку для игр или других приложений, нужно было писать варианты для каждой системы.

General MIDI

Спецификация General MIDI (сокращенно GM) упорядочила этот хаос, создав систему нумерации сэмплов. Теперь, независимо от того, кто сделал ваш синтезатор или звуковую карту, сэмпл номер 1 — это акустический рояль, а номер 128 — звук выстрела. Благодаря GM вам не надо беспокоиться о том, что ваше творение будет (или не будет) воспроизводиться правильно. Почти любые существующие синтезаторы, звуковые карты или другие устройства воспроизведения сэмплов (и аппаратные, и программные) поддерживают GM, а многие и расширяют его. Это же касается операционных систем Mac OS и Windows.

GM использует каналы 1—9 и 11—16 для таких инструментов, как клавишные, гитары и духовые (канал 10 закреплен за ударными). Для этих инструментов настройка стандартизирована таким образом, чтобы воспроизведение клавиши под номером 60 звучало как до первой октавы.

Звуки размещены в более или менее логичном порядке: например, звуки пианино сгруппированы в банках с 1 по 9, за ними следуют звуки хроматических ударных, такие как маримба, в каналах с 9 по 16. Звуки басовых можно найти в каналах с 33 по 40. Хотя спецификация GM определяет, какой звук будет вызываться, она ничего не говорит о том, как он должен воспроизводиться.

Соответственно качество и характер отдельных звуков различаются у разных производителей или даже у различных устройств одного изготовителя. Некоторые звуки GM, например акустического пианино, известны своей сложностью качественного исполнения. Другие же, например звук блок-флейты (№ 75), очень просто воссоздать даже на самой дешевой звуковой карте

Канал 10 зарезервирован за звуками ударных, таких как наборы барабанов или ручные ударные инструменты. В отличие от хроматических звуков других каналов, каждая нота канала № 10 используется для выбора конкретного ударного инструмента. Это позволяет играть партию ударных прямо на клавишном пульте. Например, используйте ноты № 60 и 61 (среднее до и ре на октаву выше) для ритма с высоким и низким бонго.

Некоторые GM-пульты и модули расширяют идею единого набора барабанов. Используя сообщения о выборе банка данных, вы сможете отдавать предпочтение тому или иному комплекту, разработанному для джаза, тяжелого рока, электронной музыки и другого.

General MIDI — это нечто больше, чем просто список звуковых банков. Спецификации требуют, чтобы совместимые звуковые модули (и аппаратные, и программные) распознавали все 16 MIDI-каналов и использовали любой из 24 динамически распределяемых голосов для одновременного воспроизведения звуков ударных и мелодии, иногда разделяемых на 8 голосов для ударных и 16 — для звуков мелодии. GM также определяет, как модули будут отвечать на конкретные сообщения контроллера, и некоторые другие полезные мелочи.

Главное, что можно создать песню на своей системе и знать, что она будет правильно воспроизводиться на любой GM. Так как файлы MIDI очень маленькие по сравнению с файлами цифрового аудио, MIDI идеально подходит для использования в Сети.

Roland GS

GS начал развиваться в 1991 году сразу за GM и неоднократно расширялся в связи с выпуском новых продвинутых моделей. XG от Yamaha активно продвигался с 1996 года, и при его разработке учитывался опыт фирмы Roland.

Стандарт GS имеет обратную совместимость с General MIDI, то есть любая MIDI-композиция, созданная для GM-устройств, может быть успешно воспроизведена и на GS-устройствах. Все GS должны соответствовать требованиям к GM-устройствам. Кроме того, для них предусмотрены некоторые дополнительные условия:

- минимальный набор из 226 инструментов (вместо 128 в GM), причем для выбора инструментов, не входящих в базовый набор, используется переключение банков с помощью контроллера № 0;

- несколько (до 128) различных 61-звучных наборов ударных;

- поддержка как минимум 20 контроллеров;

- возможность динамической редакции по MIDI восьми звуковых параметров (время атаки и затухания, частота среза фильтра, скорость вибрато и пр.);

- возможность динамической редакции по MIDI пяти параметров для ударных тембров (высота, пространственная локализация, а также уровень громкости, реверберации и хоруса);

- широкие возможности редактирования звучания с помощью исключительных системных MIDI-сообщений.

Стандарт GS не получил такого широкого распространения, как General MIDI, однако некоторые музыканты все же отдают ему предпочтение. Использовать у себя Roland GS вы можете в том случае, если у вас есть соответствующий синтезатор или звуковой модуль этой фирмы

Yamaha XG

А в сентябре 1994 года появился формат XG (Extended General MIDI), разработанный фирмой Yamaha, чтобы расширить и усложнить стандарт GM (люди вообще любят все усложнять). XG полностью обратно совместим с GM и в то же время обеспечивает большой набор дополнительных элементов: новые инструменты и редактирование их, несколько сотен эффектов и множество других возможностей, которые используются мультимедийными приложениями. Вот почему многие компании — производители музыкальной аппаратуры приняли данный стандарт и выпускают технику, которая позволяет его применять.

Прежде всего спецификация XG поддерживает гораздо большее количество инструментов, чем GM. Минимальный набор, определенный в спецификации,

должен содержать 480 инструментов. Вместе со всеми эффектами цифра получается еще более внушительной — 676 инструментов.

Спецификация XG предусматривает и большие возможности редакции исполнительских параметров по MIDI. Предусмотрено также много эффектов — реверберация (эффект «эха», когда звуковой сигнал смешивается со своими копиями, задержанными на различные временные интервалы), хорус (эффект исполнения одного звука несколькими инструментами) и другие эффекты, коих несколько десятков. Минимальное количество типов реверберации и хоруса — 8, а остальных — 35. Последние включают в себя как различные варианты и комбинации реверберации и хоруса, так и дополнительные эффекты: вращение, тремоло, «вау-вау», искажения, шумы и т.п. Любой эффект может быть использован как на отдельном MIDI-канале, так и на общем выходе.

Все группы эффектов можно применять одновременно, причем их параметры легко изменяются с помощью MIDI-сообщений. Некоторые XG-устройства имеют управляемые по MIDI графические эквалайзеры. Эти эквалайзеры могут быть либо пресетные (то есть с заранее записанными установками: в этом случае по MIDI можно изменять тип установки эквалайзера), либо настраиваемые. В последнем случае пользователь получает возможность очень гибкой работы со звуковым спектром в реальном времени, причем совершенно не загружая ресурсы процессора, ведь компьютер работает с обычным потоком MIDI-информации. Кроме того, в XG-устройствах можно применять эффекты к внешнему звуковому каналу. Практически это означает, что пользователь имеет возможность подключить к устройству микрофон или, к примеру, электрогитару и применять к входящему звуковому сигналу все те же самые эффекты, которые доступны на MIDI-дорожках. Сами эффекты при этом контролируются по MIDI обычным образом

Инструменты MIDI

Электронные музыкальные инструменты выпускаются в двух основных вариантах: это синтезаторы, создающие звуки, и сэмплеры, которые воспроизводят записанные музыкальные данные, называемые сэмплами. Конечно, многие инструменты имеют особенности обоих вариантов. Аппаратные синтезаторы и сэмплеры могут быть изготовлены в виде клавишного пульта, что позволяет им выполнять функцию MIDI-контроллеров. Все больше и больше классических синтезаторов и сэмплеров, а также инструментов широкого применения появляются в виде программных инструментов

Сейчас большинство электронных инструментов могут хранить гораздо больше информации, чем изначальные 128 звуков, определенные стандартами GM.

Синтезаторы

В своей основе синтезатор — это электронное устройство для производства и преобразования звуков. Программные воссоздания совсем старых аналоговых синтезаторов имеют огромное преимущество перед оригиналами — они понимают MIDI.

Сэмплеры

Сэмплеры отличаются от синтезаторов тем, что основным фундаментом, на котором строится выходной звук, является короткая цифровая запись, создаваемая или выбираемая оператором.

Конечно, не всегда получается записать сотни сэмплов, которые могут потребоваться, поэтому существуют библиотеки, охватывающие все: от оркестровых инструментов до коротких, ритмичных ударных и шаблонов басовых.

Виртуальные музыкальные инструменты

Теперь перейдем к виртуальным инструментам, которые тоже являются MIDI-творениями, но немного другого класса — это программное воссоздание настоящего музыкального инструмента. Виртуальные инструменты могут работать как самостоятельные программы или храниться как надстройки в других приложениях. Благодаря продвинутым технологиям моделирования многие старые аналоговые синтезаторы и электронные органы были воссозданы как виртуальные инструменты.

Классификации виртуальных инструментов

Для начала давайте с вами попробуем немного упорядочить виртуальные музыкальные инструменты, то есть классифицировать, взяв за основу тот или иной признак.

По способу запуска можно выделить две категории. Программы, запускаемые сами по себе (standalone): для работы с ними не требуется дополнительных программ или утилит. Вторая категория — это подключаемые модули (плагины). Они могут появиться только в окне другой программы, называемой хостом.

По типу технологии plug-in. Двумя словами смысл плагинов можно выразить так: на компьютере устанавливается мощная хост-программа и программный модуль-плагин. При запуске хоста она находит этот плагин и подключает его к себе как компонент. Таким образом программа-хост дополняется новой функцией, которая первоначально в ней не была заложена. Плагины бывают следующих типов: VSTi (PC и Macintosh), DirectX (Windows), DSSI/LADSPA (Linux), MESS (Linux), Audio Units (Macintosh), RTAS (Pro Tools), TDM (Time-division multiplexing) и MAS (MOTU).

По привязке к аппаратной части. Отдельные виртуальные синтезаторы могут работать только в комплексе с определенной звуковой картой — например, компании Digidesign. Плюсом здесь является то, что в этой ситуации все операции (запись и воспроизведение аудио и MIDI, работа программного синтезатора, обработка и микширование аудиопотока и т.д.) проводятся силами специального процессора на звуковой карте и не трогают центральный процессор компьютера. Минус очевиден — на чужом «железе» программа работать не будет. Но большинство виртуальных синтезаторов, к счастью, передоверяют заботу о совместимости с аудиоинтерфейсом операционной системе и соответствующим драйверам.

По целевому типу. То есть — синтезатор, сэмплер, звуковой модуль, драм-машина, имитатор акустического инструмента и др.

По сложности архитектуры. Программный музыкальный инструмент может иметь очень простую архитектуру и дизайн, а может представлять собой самый сложный комплекс совместно работающих программ (а иногда и устройств). Например, комбайн по имени Reason от компании Propellerhead — это и аудио/MIDI-секвенсор, и микшер, и синтезатор, и библиотека звуков. А Korg Legacy Collection — это набор синтезаторов, имитирующих знаменитые модели M1 и Workstation.

По технологии создания тембра. Синтезаторы создают музыкальный тембр на основе вложенных в них алгоритмов, а сэмплеры — тембр на основе загружаемых в них шаблонов — сэмплов, и каждый новый сэмпл расширяет тембровые возможности инструмента.

По наличию или отсутствию прототипа. Как вы уже знаете, существует довольно много виртуальных музыкальных инструментов, имитирующих реальные прототипы (особенно те, которые давно сняты с производства). В качестве прототипов могут быть использованы как традиционные инструменты: гитара, рояль, саксофон, струнная группа и т.д., так и электромузыкальные инструменты: орган Hammond, синтезаторы Moog или KORG.

По алгоритму синтеза звука. Все виртуальные синтезаторы цифровые по своей сути, но их внутренняя архитектура может опираться на различные идеи. К примеру, программный аналоговый синтезатор с помощью математических моделей имитирует электронные схемы, существующие в реальном прототипе. Программные имитации «реальных» инструментов могут быть действительно убедительными. Некоторые из них, вроде уже упоминаемой программы Giga/Studio, поселились в профессиональных звукозаписывающих студиях по всему миру.

Имейте также в виду, что чем мощнее программный инструмент, тем больше ресурсов от потребует у вашего компьютера. Несмотря на то что файлы MIDI занимают мало места, убедитесь, что ваш CPU справится с этой задачей. Помните,

что для комфортной работы вам нужно не меньше 2 Гбайт оперативной памяти и процессор с частотой не менее 2,8 ГГц. Существуют буквально тысячи вариантов для всех стандартных форматов плагинов VSTi, DX, MAS, TDM, RTAS и Audio Units. Вы можете выбрать один или два определенных инструмента, подготовить обширную библиотеку звуков для увеличения возможностей программ GM, поставляемых с вашей звуковой картой, или углубиться в мир аналогового синтеза.

Некоторые программные инструменты, наподобие уже упоминаемого Reason от Propellerhead, работают как standalone (надеюсь, вы помните, что значит это слово: если нет, вернитесь на несколько страниц назад). У многих из таких программ есть встроенные секвенсоры, и они связываются с другими программами с помощью дополнительных программных протоколов, например ReWire, поддержка которого есть и в других не менее популярных продуктах.

«Железные» музыкальные инструменты имеют одно неоспоримое преимущество по сравнению с их программными собратьями: они никак не влияют на ресурсы вашего компьютера. Если у вас уже есть клавишный пульт или звуковой модуль, используйте их. Благодаря сообщениям Sysx программы-библиотеки сделали хранение ваших звуков совсем простым: многие секвенсоры выпускаются в комплекте со списками сэмплов всех распространенных клавишных пультов и звуковых модулей. Поэтому вам не придется запоминать, была ли программа № 64 частью группы басовых, клавишных или флейтой-пикколо.

MIDI-файлы

Одной из трех составляющих протокола MIDI является спецификация формата хранения данных (напомню, что две другие составляющие — это формат сообщений и аппаратная спецификация интерфейса). Он был предложен организацией MMA в конце 1987 года и получил название "Стандартные MIDI-файлы" (Standard MIDI Files, SMF).

Цель MIDI-файлов — обеспечить обмен событиями (то есть MIDI-сообщениями, имеющими метку времени) между различными устройствами и программами. До появления стандартных MIDI-файлов аранжировку, подготовленную в одном секвенсоре, невозможно было загрузить в другой из-за несовместимости форматов.

Поскольку хранение информации непосредственно связано с устройством секвенсоров, остановимся на этом деле подробнее, но лишь в той мере, какая необходима для понимания SMF.

События

Сообщение MIDI, снабженное меткой времени, называется *событием*. Для указания времени могут использоваться разные единицы — тики, внутренние импульсы, время в формате SMPTE и т. п. Важно понять принципиальное отличие события от сообщения. Сообщение "живет" долю миллисекунды реального времени — от момента, когда генерируется источником, до момента, когда поступает к исполнению в приемник. Его можно отловить в процессе передачи по MIDI-кабелю в виде набора импульсов. Событие — это несколько байт информации, записанной в память устройства, на основании которой в будущем, в назначенное время, будет сгенерировано сообщение.

Например, в памяти секвенсора может храниться событие "взять ноту" с меткой времени 100 мс от начала воспроизведения. Вы можете редактировать это событие в двух измерениях: во-первых, изменять параметры самого MIDI-сообщения (в данном случае, высоту или динамику ноты), а во-вторых — перемещать ноту по дорожке, то есть изменять время исполнения сообщения.

События появляются в памяти секвенсора в процессе записи MIDI-сообщений. При нажатии кнопки Record секвенсор включает аппаратный генератор импульсов (тиков) и начинает "слушать" заданный MIDI-вход. Например, при нажатии клавиши на вход поступает сообщение "взять ноту". Секвенсор смотрит — ага, сообщение пришло на 20-й тик, и записывает его в память с меткой 20. Через пару секунд отпустили клавишу — пришло сообщение "снять ноту", внутренний генератор в этот момент радостно отмахал секвенсору 64 тика. Секвенсор сохраняет сообщение с меткой 64. Вот теперь мы имеем дело с двумя событиями — Note On и Note Off. При воспроизведении снова включается генератор импульсов. Когда подходит 20-й тик, на MIDI-выход секвенсора отправляется сообщение Note On, 64-й тик — Note Off. Мы записали, а затем воспроизвели действия исполнителя!

Очевидно, то же самое можно сделать и в "офлайне", то есть без необходимости живого исполнения. Щелкнув мышью в нужном месте дорожки (и заранее выбрав длительность ноты), мы построим в памяти секвенсора точь-в-точь предыдущую картину.

Измерение времени

На самом деле, внутри секвенсора никаких "тиков" нет. Есть аппаратный таймер, который генерирует импульсы со строго постоянной частотой (например, каждую микросекунду). Заставлять музыкантов измерять время в микросекундах было бы чудовищным издевательством, как, впрочем, и в других единицах реального времени (секундах, минутах). Музыканты привыкли мыслить в тактах и долях, а время выражать в относительных единицах (длительностях нот), зависящих от текущего темпа.

"Нет проблем, — ответили производители, — пусть меряют так, как им удобнее". Только минимальной единицей измерения будет не какая-нибудь 32-я длительность, а условно взятый тик (единица еще мельче, так, что, например, в одной тридцать второй может содержаться 48 тиков). Поскольку еще с классической эпохи темп измеряется количеством четвертей в минуту (BPM, beats per minute), то за основную длительность решили взять четверть и указывать количество тиков, приходящихся на четверть — PPQN (Pulse Per Quarter Note). Чем больше PPQN, тем лучше разрешение секвенсора и тем точнее по времени он может фиксировать сообщения при записи и выдавать их на MIDI-выход при воспроизведении. Большинство секвенсоров позволяют задавать PPQN произвольно — например, от 32 до 1536 тиков на четверть (современные секвенсоры — до 15360 PPQN).

Тик — темпозависимая единица: чем быстрее темп, тем короче интервал между тиками в единицах реального времени. Этот интервал можно найти по формуле на рис. 1.

$$\Delta t_{[\text{мс}]} = \frac{60\,000}{\text{BPM} \cdot \text{PPQN}}$$

программным секвенсором и синтезатором на звуковой плате, либо виртуальным синтезатором), то внутреннее разрешение такой системы может быть сколь угодно большим (число 15360 PPQN впечатляет). Это позволяет осуществить синхронизацию MIDI- и аудиоданных с точностью до семпла. Но как только мы соединяем секвенсор и тон-генератор MIDI-кабелем посредством MIDI-интерфейса, высокое разрешение утрачивает актуальность.

Почему? Напомню, что один байт передается по MIDI-интерфейсу в течение 320 микросекунд. А это значит, что, например, сообщение о взятии ноты (состоящее из трех байт) будет передаваться 960 мкс или почти целую миллисекунду. Теперь представим, что в секвенсоре при темпе 120 BPM и разрешении 2048 PPQN запрограммированы две ноты с интервалом два тика между собой. В единицах реального времени это составляет 488 микросекунд. Так вот: тон-генератор не сможет получить вторую ноту через 488 микросекунд после первой, а реально — только через 960 мкс. Так что исполнит он ее спустя не два тика, а почти четыре.

Отсюда вывод: при работе через MIDI-интерфейс (когда секвенсор и тон-генератор разнесены) разрешение секвенсора более одного тика на 960 микросекунд не имеет смысла. Чтобы узнать, сколько это будет в PPQN, можно воспользоваться формулой на рис. 2.

$$PPQN_{max} = \frac{60\,000\,000}{BPM \cdot 960}$$

В таблице на рис. 3 для разного темпа показаны значения PPQN, превышать которые бессмысленно.

BPM	PPQN _{max}
40	1563
60	1042
80	782
100	625
120	521
140	447
180	348
220	285
240	261

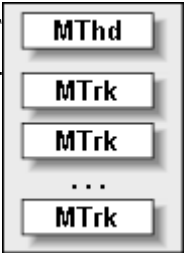
Структура SMF

Стандартные MIDI-файлы состоят из блоков (Chunks). Всего есть два типа блоков: блок заголовка (Header Chunk) и блок трека (Track Chunk). В файле SMF может быть только один блок заголовка и один или более блоков трека. Блок имеет структуру: первые четыре байта — идентификатор, следующие четыре байта — длина блока в байтах, исключая восемь байт тип/длина. Идентификатор блока заголовка — это четыре символа "MThd", блока трека — четыре символа "MTrk". Такая структура позволит в будущем определять новые типы блоков, а незнакомый блок может быть легко проигнорирован на основе его длины. Спецификация SMF предупреждает: "Программы должны быть готовы, встретив блоки неизвестных им типов, игнорировать их".

Блок заголовка содержит общую информацию о файле, блок трека — поток MIDI-сообщений с метками времени. Кроме того, в MIDI-файле сохраняется

дополнительная информация, необходимая секвенсорам: темп, размер, тональность, настройки метронома и тому подобное. Эта информация по MIDI-интерфейсу не передается, а образующие ее события называются мета-событиями.

MIDI-файл всегда начинается с блока заголовка, за которым следуют один или более блоков трека (рис. 9). То есть, любой стандартный MIDI-файл начинается с четырех букв "M", "T", "h", "d".



В 1991 году Microsoft предложила "заворачивать" стандартные MIDI-файлы в блок с именем "data" внутри файла RIFF. В результате родился гибрид под названием RMID (расширение RMI). По сути, это все тот же стандартный MIDI-файл, только с приписанной к началу файла шапкой из двадцати байт (рис. 10).

RIFF ID	размер RMID	тип файла	ID блока	размер SMF	SMF
'RIFF'	nn nn nn nn	'RMID'	'data'	nn nn nn nn	'MThd'

20 байт