



ключить кварцевый резонатор, цепи сброса и питания. Оператор может воздействовать на микроконтроллер при помощи органов управления. Это могут быть обыкновенные кнопки или порт связи с компьютером. Для программирования ARM-микроконтроллера необходимо организовать канал связи с компьютером через последовательный порт. Этот порт будет посылать в контроллер коды клавиш, нажимаемых на клавиатуре компьютера. Через этот же порт можно принимать информацию от микроконтроллера и отображать её на экране монитора. Все указанные компоненты присутствуют на приведённой схеме устройства.

### СРЕДА РАЗРАБОТКИ

Необходимо создать тестовую программу для нашего устройства и запрограммировать микроконтроллер. После этого он будет воспроизводить звуки и фразы под воздействием команд.

Для создания программы воспользуемся пакетом *Keil uVision 3* для ARM. Эта среда разработки свободно распространяется производителем. Бесплатные версии позволяют создавать программы с 30-дневным ограничением и без ограничения по сроку использования, но с ограничением размера кода несколькими десятками килобайт. Оба варианта подходят для решения поставленной задачи. Загрузить пакет можно с интернет-страницы разработчиков [1].

После установки пакета, в каталоге, где он разместится, будет создана папка с примерами готовых программ. Запустите среду разработки и откройте в ней проект с самой простой программой, которая позволяет выводить на экран монитора сообщение «Hello world». Окно с внешним видом среды разработки и кодом указанной программы показано на рисунке 2.

Пример написан на языке программирования Си и начинается со строк, описывающих подключения стандартной библиотеки ввода-вывода *stdio.h* и заголовочного файла описания внутренних регистров ARM-микроконтроллера серии LCP21xx. Далее следует главный модуль программы *main*, который включает в себя настройку порта

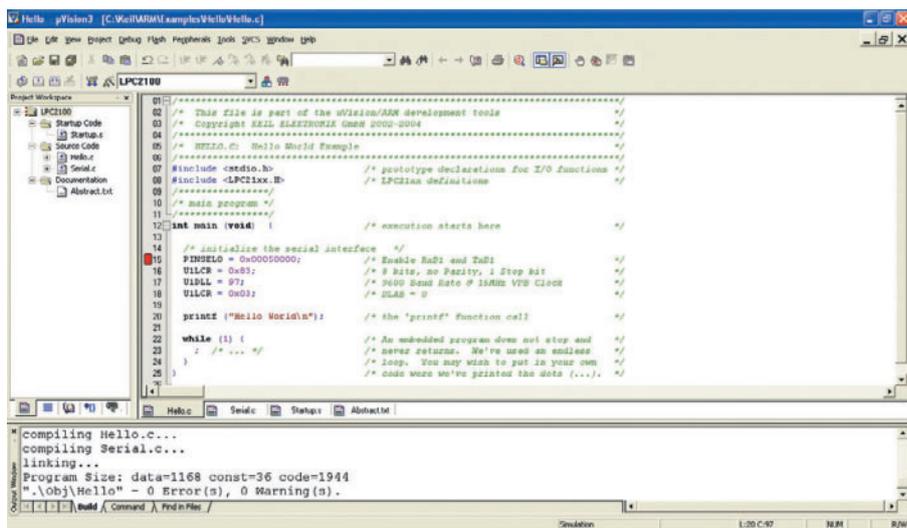


Рис. 2. Окно среды разработки с кодом программы

ввода-вывода Port0 и инициализацию встроенного приёмопередатчика UART1 с установкой его скорости и формата данных.

Команда вывода строки *printf* обеспечивает вывод информации в порт UART1 благодаря наличию программного модуля *Serial.c*. Содержимое этого модуля можно просмотреть в среде разработки, щёлкнув дважды левой кнопкой мыши по названию модуля в левом окне среды. Данный модуль содержит функции стандартного ввода-вывода *getchar* и *putchar*, используемые поточными функциями ввода-вывода *scanf* и *printf* для переназначения файлового ввода-вывода на устройство UART1. Кроме этих файлов проект программы содержит файл *Startup.s*, предназначенный для инициализации МК, который присутствует в среде разработки в готовом виде и не требует редактирования.

Компиляция программы выполняется любым из трёх способов: через главное меню *Project → Build Target*, с помощью кнопки  на панели инструментов или нажатием клавиши F7.

Запуск программы в отладочном режиме выполняется также любым из трёх способов: через главное меню *Debug-Start/Stop Debug Session → Build Target*, с помощью кнопки  на панели инструментов или нажатием клавиш Ctrl+F5.

В этом режиме можно выполнять программу полностью или по шагам, а также просматривать содержимое переменных программы, памяти и регистров МК. Кроме того, в отладочном режиме можно контро-

лировать состояние всех портов ввода-вывода, таймеров, АЦП и ЦАП и пр. через закладку *Peripherals* в главном меню.

Среда отладки позволяет эмулировать последовательный порт компьютера. Открыть окно с эмулятором можно через главное меню *View → Serial Window #1* или с помощью программной кнопки  на панели инструментов. В окне эмулятора последовательного порта компьютера будет отображаться информация, выводимая в программе через порт с номером, меньшим на единицу, т.е. через UART0. Поскольку в рассматриваемой программе функции ввода-вывода *getchar* и *putchar* перенаправлены в программном модуле *Serial.c* на порт UART1 микроконтроллера, необходимо открыть окно для симулятора этого порта с помощью главного меню *View → Serial Window #2*. В этом окне будет выводиться информация с помощью оператора *printf*, а вводимые с клавиатуры символы будут передаваться в программу через порт UART1. В частности, сообщение программы «Hello world» будет отображено именно в этом окне. Таким образом, можно отлаживать программу для МК без наличия самого устройства.

Ещё одним полезным инструментом среды разработки *Keil* является логический анализатор сигналов. Его окно открывается в отладочном режиме через главное меню *View → Logic Analyzer Window* или с помощью кнопки  на панели инструментов. Анализатор позволяет наблюдать состояния любых вы-

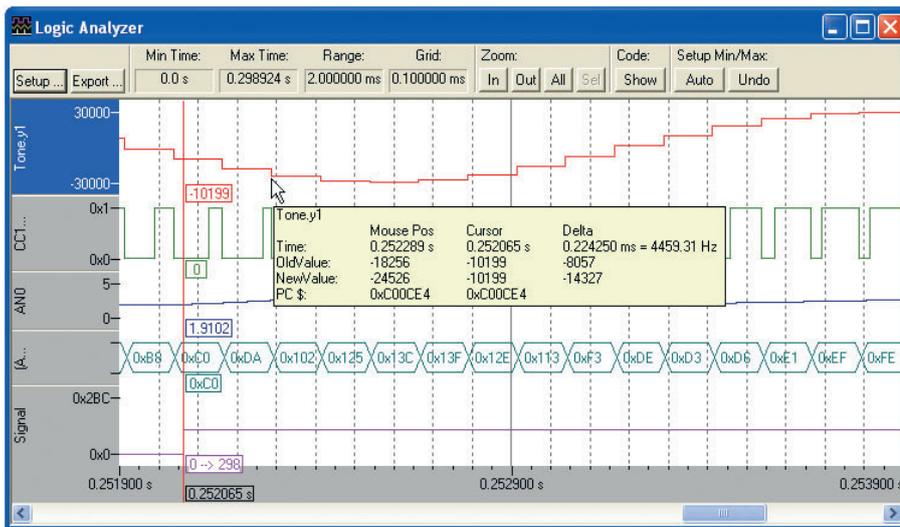


Рис. 3. Окно программы логического анализатора

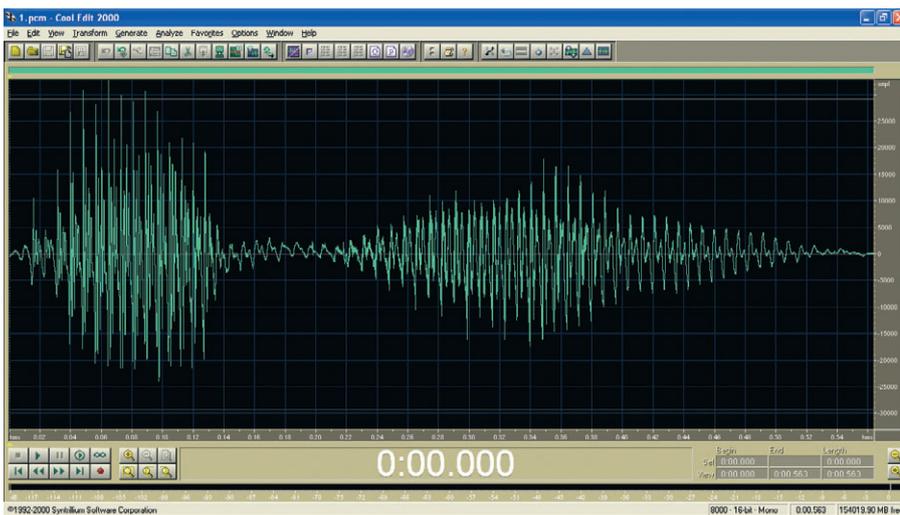


Рис. 4. Осциллограмма сигнала звуковой фразы в окне звукового редактора

водов МК в процессе выполнения программы. При этом сигналы будут отображаться в графическом виде, как на дисплее запоминающего осциллографа (см. рис. 3).

В программе можно устанавливать и убирать точки останова с помощью кнопки F9. Пошаговое выполнение программы можно осуществлять с помощью кнопок F10 и F11.

Среда разработки Keil предоставляет и другие возможности для отладки программного обеспечения, в числе которых – программная эмуляция внешних устройств и воздействий на входы микроконтроллера. Среда разработки имеет развитую справочную систему, которая содержит описания самой среды и примеров программ.

**ПРОГРАММА**

Проект программы состоит из нескольких файлов, которые со-

держат главный модуль, функции для последовательного порта UART0 и ЦАП, а также обработчик прерывания для таймера. Файлы проекта можно скачать с интернет-страницы журнала «Современная электроника» ([www.soel.ru](http://www.soel.ru)). Алгоритм программы представляет собой программный автомат, который циклически опрашивает последовательный порт UART0 и, в зависимости от полученного кода символа, выполняет процедуру его озвучивания.

В программе предусмотрена функция генерации звука произвольной высоты, с заданной длительностью и числом повторов. Эта функция вызывается при получении из порта UART1 кода символа «s». С её помощью можно формировать звуки различных музыкальных нот и создать устройство, выполняющее функции программного генератора низкой частоты. Текст кода

главного модуля *main.c* программы приведён в листинге 1 в дополнительных материалах к статье на интернет-странице журнала «Современная электроника».

При формировании звуковых сообщений используются данные из программного модуля *sound.c*, который содержит массивы слов с цифровыми выборками заранее подготовленных звуковых фраз. Для создания таких массивов данных необходимо воспользоваться любой подходящей программой для записи звука через микрофон и последующего его редактирования, например звуковым редактором Cool Edit 2000 (Adobe Audition). Окно этой программы с осциллограммой звуковой фразы показано на рисунке 4.

Звуковой редактор позволяет открыть любой звуковой файл в формате wav, mp3 или др. и преобразовать его в необходимый формат. Каждая из фраз нормируется по амплитуде и сохраняется в отдельном файле в формате PCM 16 бит/8 кГц. После этого файлы объединяются в один программный модуль *sound.c*. Во время компиляции этот файл вместе с другими программными модулями будет обработан, и данные, находящиеся в нём, будут перенесены в HEX-файл для занесения в память программ микроконтроллера.

Обработка файлов с массивом звуковых данных заключается в приведении значений этих данных к формату 10-разрядного ЦАП микроконтроллера, который преобразует входной код в диапазоне от 0 до 0x3FF в выходной аналоговый сигнал 0...3,3 В. Для автоматизации этой операции автором была написана специальная программа *pcmtoc.c*, которая преобразует 16-разрядные слова, имеющие положительные и отрицательные значения, в 10-разрядные однополярные. Эта программа представляет собой консольное приложение и выполняется из командной строки с одним входным параметром, в качестве которого используется имя файла со звуковыми данными.

В результате работы программа *pcmtoc.c* формирует выходной файл *sound.c*. В этом файле все данные, подготовленные для ЦАП, записаны в виде массива, представленного в формате языка Си. Если для работы

устройства требуется несколько отдельных фраз, то для каждой фразы выполняется процедура, описанная выше, после чего все записи из файлов *sound.c* переносятся в один файл *sound.c* и упорядочиваются индексы массивов данных для каждой фразы. Затем отредактированный файл *sound.c* подключается к проекту программы.

### ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРА

После сборки всего проекта, его компиляции и отладки, можно осуществить программирование микроконтроллера. Загрузка флэш-памяти программ АРМ-микроконтроллера также осуществляется из среды разработки *Keil* с помощью свободно распространяемой программы *LPC210x\_ISP.exe* от компании NXP. Для запуска этой программы необходимо нажать программную кнопку . Внешний вид окна программы для прошивки HEX-файла во флэш-память программ МК показан на рисунке 5.

После прошивки микроконтроллера устройство может быть подключено непосредственно к СОМ-порту компьютера, и можно проконтролировать его работу в режиме реального времени. В качестве отладочной или управляющей программы компьютера, которая будет передавать коды символов клавиатуры через СОМ-порт, можно использовать любую подходящую терминальную программу, например *terminal* [2]. Внешний вид её рабочего окна показан на рисунке 6.

Данная программа позволяет выбрать для работы любой свободный СОМ-порт компьютера и установить скорость и формат данных для этого порта. Для контроля передаваемых и принимаемых данных программа имеет два окна. Формат представления отображаемых данных может задаваться пользователем. Программа *terminal* имеет множество других дополнительных возможностей, которые могут пригодиться в дальнейшем.

При выполнении компиляции проекта среда разработки *Keil* генерирует вспомогательные файлы, которые используются во время отладки программы. После завершения отладки необходимость в

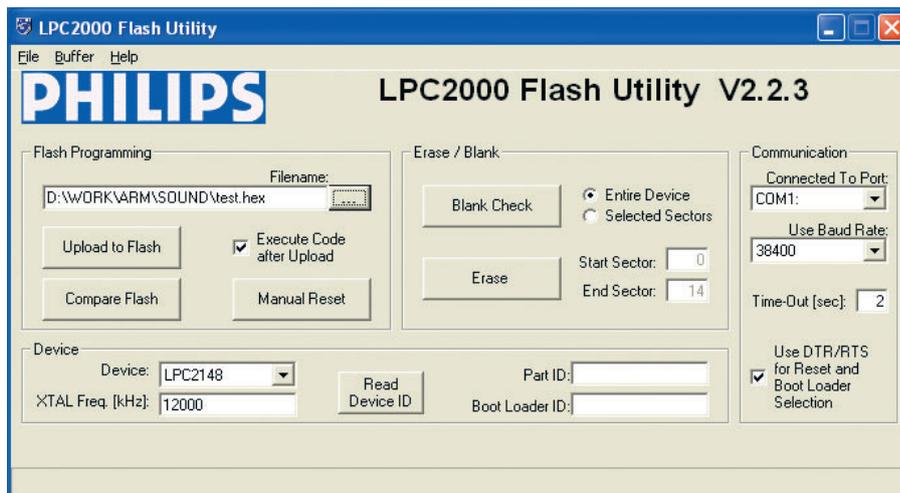


Рис. 5. Окно программы для прошивки HEX-файла во флэш-память программ МК

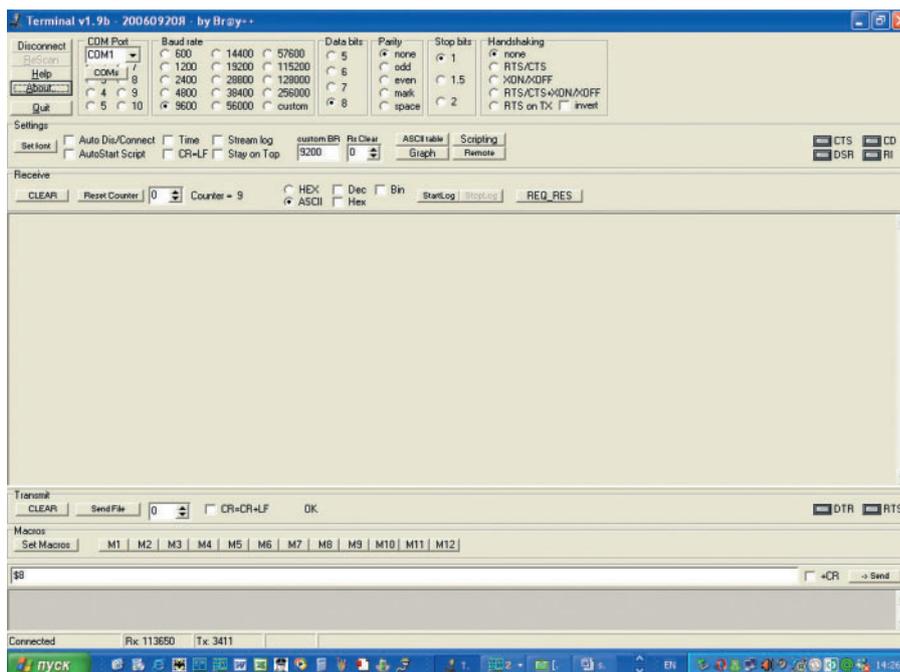


Рис. 6. Окно программы связи через последовательный порт

этих файлах отпадает, и они могут быть удалены из проекта. Для корректного удаления вспомогательных файлов автором был создан специальный командный файл с названием *clear.bat*, который удаляет все файлы с определёнными расширениями.

### ЗАКЛЮЧЕНИЕ

Представленный в статье проект программы и устройство можно развить и дополнить компонентами, необходимыми для выполнения конкретной задачи.

Следует помнить, что для хранения звуковых фраз требуется довольно большой объём памяти, составляющий 16 Кб для 1 с звучания фразы (при 10-разрядном представлении, при 8-разрядном потребует-

ся вдвое меньше памяти). Поэтому при подготовке звуковых сообщений необходимо, зная объём памяти микроконтроллера, подсчитать их общую длительность. Кроме того, следует зарезервировать место в памяти МК и для управляющей программы.

Для создания программ можно воспользоваться любой другой средой разработки для АРМ-микроконтроллеров, например IAR [3], которая обладает не менее развитыми средствами редактирования и отладки.

### ЛИТЕРАТУРА

1. <http://www.keil.com>.
2. <http://eldigi.ru/site/programs/?download=terminal.rar>.
3. <http://www.iar.com>.

