

# Современные микроконтроллеры и речь

Олег Вальпа (Челябинская обл.)

Описывается простое устройство на базе АРМ-микроконтроллера и методика его программирования, которая позволяет озвучивать режимы работы и события.

## ВВЕДЕНИЕ

Современные микроконтроллеры (МК) имеют в своём составе, кроме основных функциональных блоков (порты, таймеры, АЦП, ЦАП и пр.), достаточно большой объём постоянной памяти программ, составляющий десятки и даже сотни килобайт. Причём размер этой памяти постоянно увеличивается с созданием новых моделей микроконтроллеров, что позволяет создавать для них программы нового качества, обогащённые дополнительными сервисами. Например, озвучивать человеческим голосом режим работы, событие, аварийные ситуации и действия оператора.

Конечно, подобные программы можно было создавать и раньше, но для этого требовались дополнительные компоненты и соединения. Это приводило к усложнению и удо-

рожанию устройства. Современные микроконтроллеры уже содержат всё необходимое для решения подобной задачи и не требуют дополнительных компонентов. Поэтому внедрение дополнительного сервиса не приводит к повышению стоимости устройства.

## Постановка задачи

Требуется разработать устройство, способное озвучивать режимы работы и некоторые события, связанные с действиями оператора. Например, при включении сообщать о готовности к работе «Устройство готово!»; при нажатии цифровых кнопок озвучивать их названия: «Один», «Два» и т.д.

Предполагается использовать популярный АРМ-микроконтроллер LPC2148 фирмы NXP (Philips). Данный микроконтроллер содержит

всё необходимое для решения данной задачи: встроенный 10-разрядный ЦАП, достаточный объём памяти программ (512 Кб), два порта UART, порты ввода-вывода, таймеры с широтно-импульсными модуляторами, два интерфейса I<sup>2</sup>C, интерфейс SPI и SSP, 14-канальный 10-разрядный АЦП и порт USB. Такой богатый арсенал позволяет создавать на основе МК множество самых разнообразных устройств при весьма скромной стоимости.

## РЕАЛИЗАЦИЯ

Поскольку для реализации устройства требуется лишь записать программу в микроконтроллер и вывести звуковой сигнал на внешний усилитель, схема устройства получается простой (см. рис. 1). На ней показано подключение только тех элементов, которые обеспечивают функционирование микроконтроллера и решение поставленной задачи. Остальные выводы микроконтроллера могут использоваться для других целей.

Чтобы микроконтроллер начал работать, к нему необходимо под-

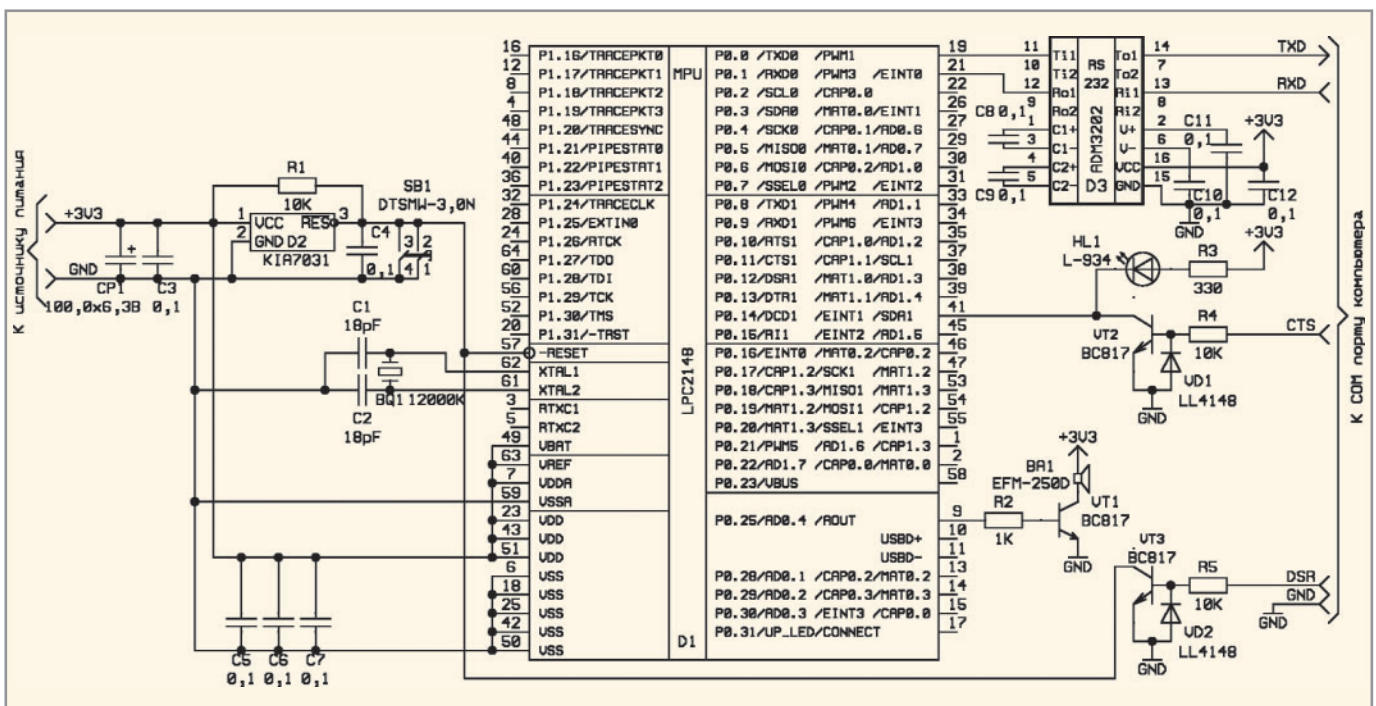


Рис. 1. Принципиальная электрическая схема



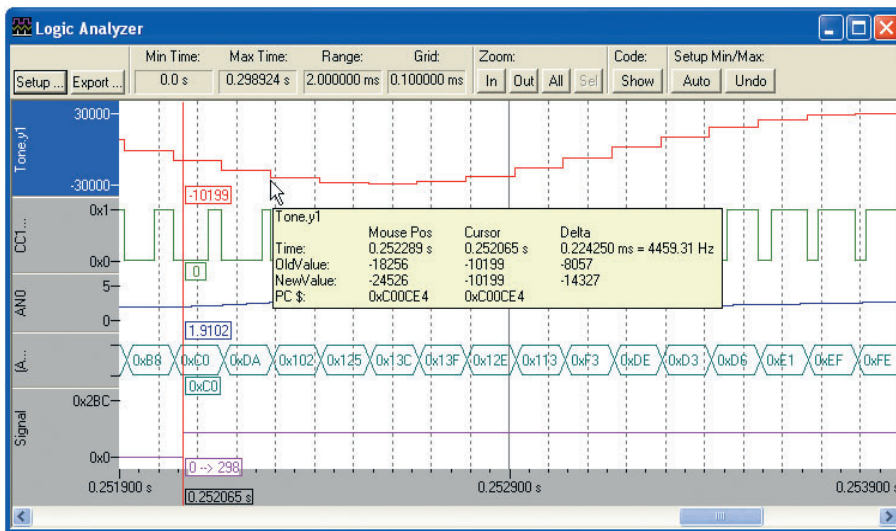


Рис. 3. Окно программы логического анализатора

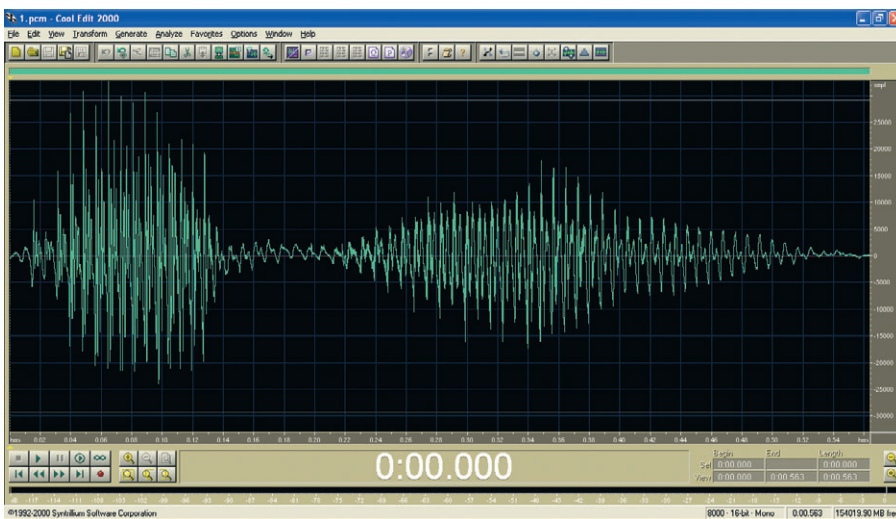


Рис. 4. Осциллограмма сигнала звуковой фразы в окне звукового редактора

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

