

Роман Попов (КОМПЭЛ)

МИКРОКОНТРОЛЛЕРЫ STM8 «С НУЛЯ»



В статье подробно рассказывается о том, как начать работу с микроконтроллерами STM8 компании STMicroelectronics. Мы рассмотрим среды разработки, разберемся, как настроить проект для работы, коснемся некоторых особенностей линковки библиотек к проекту, и, в конце концов, совместно разберем небольшой рабочий проект для закрепления полученных знаний.

Линейка микроконтроллеров STM8 заменила STR7 в 2008 году. На сегодня она состоит из трех семейств: для автомобильного, низкопотребляющего и общего применений. Это, соответственно, STM8A, STM8L и STM8S. Общая же номенклатура составляет более 160 позиций, и компания постоянно работает над расширением каждой линейки. Хотя на рынке наметилась тенденция перехода на 16/32-битные микроконтроллеры, 8-битные все еще составляют основную часть рынка, и в ближайшее десятилетие их доля останется значительной.

Инструментарий для работы

Для работы с любым микроконтроллером, и STM8 — в частности, нам понадобятся отладочный комплект и среда разработки с Си-инструментарием. В качестве отладочного комплекта можно использовать любой из предоставляемых комплектов как от ST, так и от сторонних производителей. Рассмотрим STM8S-Discovery, его внешний вид представлен на рис. 1.

Данный выбор обусловлен, в первую очередь, очень низкой стоимостью комплекта (меньше \$15), его доступностью и дальнейшей пригодностью. Комплект состоит из двух частей. Одна — это контроллер STM8S105C6 со светодиодом, сенсорной кнопкой и разъемами, на которые выведены все свободные порты ввода/вывода. Вторая — программатор-отладчик ST-Link с USB-интерфейсом. Если внимательно посмотреть на изображение отладочного комплекта, то можно увидеть, что с помощью небольших усилий его можно разделить на две части, и в дальнейшем использовать отдельный ST-Link для программирования и отладки своих собственных разработок. Всю необходимую документацию вы можете посмотреть и скачать по ссылке [http://](http://www.st.com/mcu/contentid-130-113-STM8S_DISCOVERY.html)

www.st.com/mcu/contentid-130-113-STM8S_DISCOVERY.html.

Выбор программного инструментария для разработки

На данный момент для разработки и отладки программного обеспечения для STM8 существует четыре среды: ST Toolset от STMicroelectronics (www.st.com), Ride7 от Raisonance (www.raisonance.com), CXSTM8 от Cosmic software (www.cosmicsoftware.com), IAR Embedded Workbench от IAR Systems (www.iar.com). Сравнительный анализ средств разработки программного обеспечения представлен в таблице 1.

Пакет ST Toolset включает в себя среду разработки ST Visual Develop и отдельную программу для более функционального внутрисхемного программирования flash-памяти микроконтроллеров ST Visual Programmer. Среда разработки ST Visual Develop имеет встроенный инструментарий для раз-

работки программного обеспечения на языке assembler, но у нее также имеется возможность подключения и использования Си-инструментария от Raisonance и Cosmic software. Стоит заметить, что все четыре среды с Си-инструментарием предоставляют возможность использования с некоторыми ограничениями, а именно — по загружаемому коду во flash-память. У Raisonance оно составляет 16 Кбайт, у Cosmic software — 32 Кбайт, а у IAR — 8 Кбайт или полную версию с 30-дневным ограничением. Самый дешевый и оптимальный вариант — это использование ST Visual Develop в качестве среды разработки и Си-инструментария либо от Raisonance, либо от Cosmic software. Вы, конечно же, можете заметить, что наиболее известным и популярным является инструментарий от IAR System, и тут с вами трудно не согласиться. Он представляет собой более серьезный продукт с лучшей технической поддержкой, но и является самым дорогим. Итак, мы останавливаем свой выбор на ST Visual Developer, плюс Си-инструментарий от Cosmic software и Raisonance. Большой разницы в использовании обоих инструментариев нет, и далее в своих разработках вы можете остановиться на любом из них. Если вы — начинающий

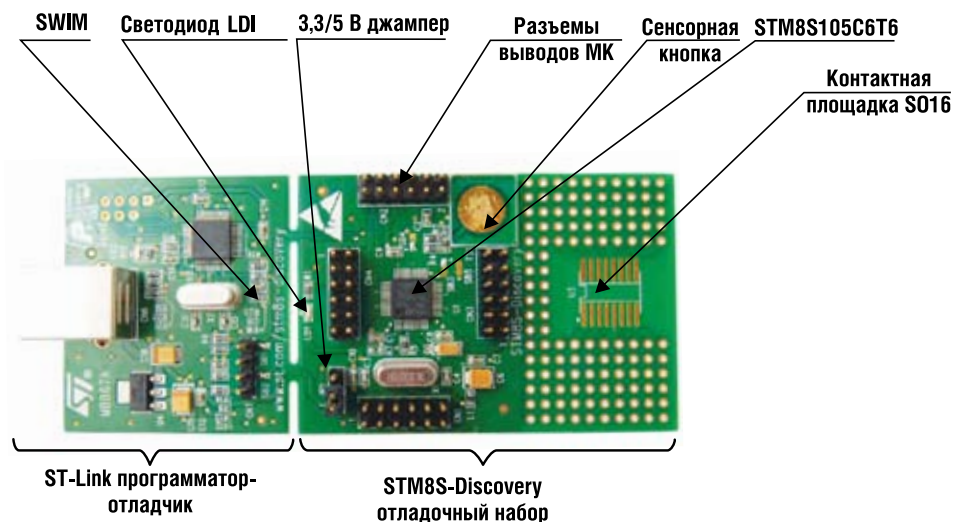


Рис. 1. STM8S-Discovery

Таблица 1. Сравнительный анализ бесплатных средств разработки программного обеспечения

Инструментарий	Среда разработки	Си-инструментарий	Си-инструментарий других производителей	Ограничение Си-инструментария, Кбайт	Программатор-отладчик
STMicroelectronics	ST Visual Develop	Нет	Cosmic software, Raisonance	Нет	ST-Link R-Link STICE
Raisonance	Ride 7	Есть	Нет	16	R-Link
Cosmic software	CXSTM8	Есть	Нет	32	Нет
IAR Systems	IAR Workbench	Есть	Нет	8 или полная версия на 30 дней	ST-Link STICE

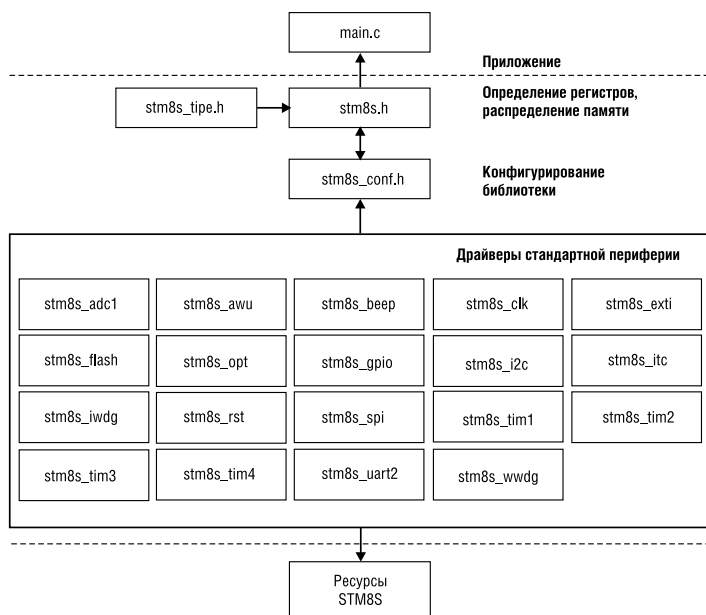


Рис. 2. Структура библиотеки стандартной периферии

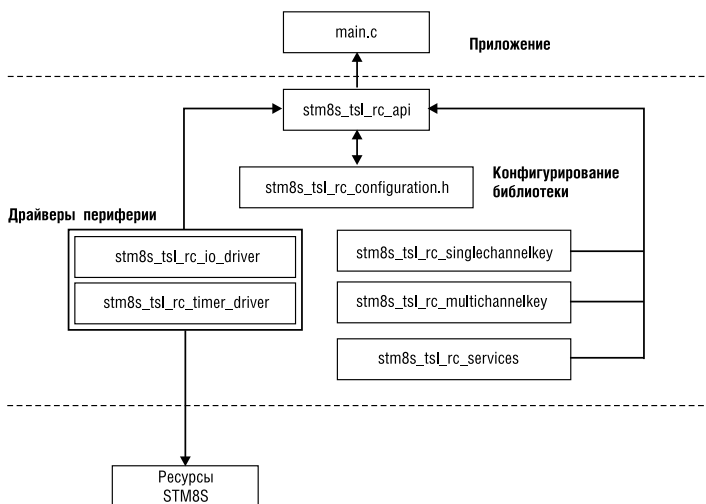


Рис. 3. Структура сенсорной библиотеки

разработчик, и у вас недостаточный опыт, рекомендую устанавливать все программное обеспечение по предполагаемым установщиком местам расположения, т.е. по умолчанию. Это необходимо для меньшей путаницы при дальнейшей настройке проектов, расположении файлов, библиотек и синхронизации с информацией приведенных материалов.

Установка программного инструментария для разработки

Итак, скачиваем и устанавливаем ST Toolset по ссылке <http://www.st.com/stonline/products/support/micro/files/sttoolset.exe>. Далее нам необходимо скачать Си-инструментарий от Raisonance и от Cosmic software.

Открываем браузер, входим по ссылке <http://www.mcu-raisonance.com/microcontrollers-link-free-download.html>, скачиваем два установочных файла – это среда разработки Ride7 и инструментарий RKit-STM8. Устанавливаем в той же последовательности. После установки, необходимо получить бесплатную лицензию на использование Си-инструментария, заполнив регистрационную форму http://www.mcu-raisonance.com/stm8_st7_registration.html. В течение короткого срока вы получите лицензионный ключ по указанной в регистрации электронной почте. Запускаем среду «Пуск → Программы → Raisonance Tools → Ride7 → Ride7» и переходим на «Help → License». Вводим общую информацию, выбираем «Manual Activation», копируем ключ из поля «This computer Serial Key» в поле «Serial Key», вводим ключ, присланный по электронной почте в поле «Paste the Activation code and Select Next» и, если все сделано верно, Си-инструментарий будет активирован и выдаст соответствующее сообщение.

Для установки Си-инструментария от Cosmic Software переходим по ссылке <http://www.cosmicsoftware.com/download.php> и скачиваем инструментарий «STM8 32k free tools 2010 special edition», предварительно заполнив регистрационные данные. Далее устанавливаем инструментарий, отправляем информацию, выданную при его установке, на электронную почту «stm8_32k@cosmic.fr». Спустя некоторое время вам придет ответное письмо с файлом лицензии. Теперь вы можете активировать Си-инструментарий, запустив среду разработки «Пуск → Программы → Cosmic Tools → STM8 32K Compiler 4.3.4 → CXSTM8», которая предложит вам активировать продукт двумя методами. Мы выбираем «Specify the License File» и указываем файл с лицензией.

На сайте STMicroelectronics вы можете найти всю необходимую информацию для работы с STM8S (<http://www.st.com/mcu/familiesdocs-113.html>). На основе одного из примеров мы рассмотрим проект, более подробно останавливаясь на ключевых моментах. Также мы остановимся на двух библиотеках – сенсорной и библиотеке стандартной периферии микроконтроллера – написанных инженерами STMicroelectronics для быстрого освоения всех линеек МК и вывода продукции на рынок. Вы можете отказаться от использования библиотек и работать напрямую с именами регистров или написать собственное программное обеспечение для работы с периферией микроконтроллера.

Обзор библиотек

Библиотека стандартной периферии содержит набор функций, структур данных и макросов, охватывающих свойства периферии микроконтроллеров STM8S. Использование библиотеки в значительной степени облегчает процесс разработки собственного программного обеспечения, т.к. устраняется необходимость изучения документации с именами регистров и их функционального назначения. Последняя версия библиотеки со всей необходимой информацией доступна по ссылке <http://www.st.com/mcu/familiesdocs-113.html>, раздел «firmware». Структура библиотеки представлена на рис. 2.

Заголовочный файл «stm8s.h» содержит определения констант и структур регистров для всей периферии. Для ис-

пользования библиотеки данный файл необходимо включить («#include "stm8s.h"») в основной листинг программы «main.c», и раскомментировать строку («#define USE_STDPERIPH_DRIVER»). Функционал каждого периферийного модуля состоит из заголовочного и исполняемого файла, например, для GPIO – это «stm8s_gpio.h» и «sym8s_gpio.c». В заголовочном файле описаны все переменные, константы, структуры данных и функции, обеспечивающие полный функционал конкретного периферийного модуля. В исполняемом файле осуществляется реализация функционала, описанного в заголовочном файле.

В заголовочном файле «stm8s_type.h» стандартные типы переменных стандарта ANSI C переопределены в более короткие и понятные имена типов. В конечном счете вы можете скорректировать данный файл под наиболее понятные и удобные для вас имена типов.

Заголовочный файл «stm8s_conf.h» используется для конфигурирования библиотеки необходимых для работы периферийных модулей, задания определенных констант (например, тактовой частоты работы ядра). Это осуществляется через раскомментирование макроопределений, соответствующих определенному периферийному модулю, например, в приведенном ниже листинге раскомментированы вторая и четвертая строки:

```
1. /***** CLK *****/
2. #define _CLK (1)
3. /***** EXTI *****/
4. #define _EXTI (1)
5. /**** FLASH/DATA EEPROM ****/
6. /**** #define _FLASH (1) ****/
7. /**** OPTION BYTES *****/
8. /**** #define _OPT (1) ****/.
```

В связи с наличием разной периферии в файле «stm8s_conf.h» учтены особенности периферийных модулей для определенного микроконтроллера. Например, в нижеприведенном листинге таймер 3 определен только в микроконтроллерах STM8S208, STM8S207 и STM8S105:

```
1. /***** TIM3 *****/;
2. #if defined(STM8S208) || defined(STM8S207) ||
   defined(STM8S105);
3. /**** #define _TIM3 (1) ****/;
4. #endif /**** (STM8S208) || (STM8S207) ||
   (STM8S105) ****/.
```

Работа с библиотекой и ее использование на конкретном примере будут рассмотрены ниже в разделе «Создание нового проекта, конфигурирование библиотек». Более подробную информацию, файл справки «stm8s_fwlib_um.chm» вы можете посмотреть либо по ссылке <http://www.st.com/mcu/familiesdocs-113.html>, либо в директории «STM8S_StdPeriph_Driver» загруженного пакета программного обеспечения.

При использовании прерываний необходимо быть внимательным, так как обработка источников прерывания различна для обоих инструментариев. Для инструментария от Raisonance используется функция без параметров (void) с префиксом «interrupt» и номера прерывания, например:

```
void timerA_handler(void)
interrupt 8
{
// Код обработки прерывания
}
void top_level_interrupt_handler(void)
interrupt 0
{
```

```
// Код обработки прерывания
}.
```

Для инструментария от Cosmic все «пустые» обработчики прерывания описаны в файле «stm8s_it.c», и весь код по обработке прерывания следует размещать здесь. Таблица векторов прерывания располагается в «stm8s_interrupt_vector.c», который связан с документом «stm8s_it.c» через заголовочный файл. Для более подробного ознакомления с обработкой прерываний и особенностями компилятора необходимо посмотреть документацию соответствующего компилятора.

Сенсорная библиотека разработана инженерами STMicroelectronics для применения в микроконтроллерах STM8, так как в настоящее время популярность и актуальность использования решений на основе сенсоров очень велика. Библиотека представляет собой набор совместимых C-файлов, включающих API, который образует интерфейс для работы с другими уровнями программного обеспечения. У компании STMicroelectronics существуют разнообразные решения в данной области, более подробную информацию вы можете получить на сайте компании. Структура библиотеки представлена на рис. 3.

Для использования библиотеки в приложении необходимо добавить все заголовочные и исполняемые файлы в проект. Исключение составляют два файла – «STM8_TSL_RC_Configuration_TOADAPT.h» и «STM8_TSL_RC_routines.asm». Первый копируется в директорию проекта и переименовывается в «stm8_TSL_RC_Configuration.h». Второй добавляется в проект только при использовании инструментария от Raisonance. Файлы «stm8_tsl_rc_api.h» и «stm8_tsl_rc_api.c» определяют функции API, переменные, структуры данных, константы для связи между библиотекой и кодом пользователя. Для использования библиотеки заголовочный файл «stm8_tsl_rc_api.h» должен быть включен в основной исполняемый файл «main.c». Документ «stm8_TSL_RC_Configuration.h» содержит статические конфигурационные параметры, которые должны быть сконфигурированы в соответствии с аппаратной частью проекта. Необходимо проверить все параметры с префиксом «#define» в соответствии с правильными значениями. Более подробно работа с библиотекой будет рассмотрена в разделе «Проект discover».

Среда разработки ST Visual Developer

Для начала работы нам необходимо ознакомиться со средой разработки ST Visual Developer. В среде существуют два основных понятия – проект и рабочая область.

Одна рабочая область может содержать несколько проектов. Это может быть удобно при разработке ПО. Рабочая область создается или открывается через «File → New Workspace...» или «File → Open Workspace...». В уже созданной рабочей области можно создавать, удалять и добавлять новый проект через «Project → Add New Project to WorkSpace...», «Project → Remove Project from WorkSpace» или «Project → Insert Project to WorkSpace...». Для работы с определенным проектом необходимо его сделать активным: «Project → Set Active Project». Все вышеописанные операции также можно выполнить, кликнув правой кнопкой мыши на проекте или рабочей области. Добавление папок и файлов в проект осуществляется кликом правой кнопки мыши на проекте или файле; в выпадающем меню вам будут доступны эти операции. Операции по компилированию и сборке проекта доступны во вкладке основного меню «Build → Compile filename», «Build → Build», «Build → Rebuild All». Выбор программатора-отладчика доступен во вкладке «Debug Instrument → Target Settings...», а все операции по отладке находятся во вкладке основного меню «Debug» и на дополнительной панели инструментов. Для отдельной настройки каждого проекта в среде, по умолчанию, доступны два режима конфигурации проекта –

«debug» и «release». Режим «debug» предназначен для использования в режиме отладки. Режим «release» предназначен для автономной работы ПО в микроконтроллере без избыточной функциональности режима «debug». Каждый из режимов вы можете настроить персонально или создать свой собственный режим со специфичными для вас настройками. Более подробную информацию для работы со средой вы можете прочитать в соответствующем руководстве.

Структура пакета программного обеспечения микроконтроллера STM8S-Discovery

Для дальнейшей работы нам необходимо скачать пакет программного обеспечения (примеры, библиотеки, файлы справки) для оценочного набора STM8S-Discovery (http://compeljournal.ru/images/articles/discover_pack.zip (8,5 Мбайт)). Пакет имеет следующую структуру:

- «Libraries» – директория состоит из двух частей:
 - STM8_TouchSensing_Driver – сенсорная библиотека с заголовочными, исполняемыми и справочными файлами;
 - STM8S_StdPeriph_Driver – библиотека периферии с заголовочными, исполняемыми и справочными файлами;
 - «Project» – директория с примерами готовых проектов:
 - discover – проект демонстрирующий работу сенсорной библиотеки и библиотеки периферии;
 - program_template – шаблон для быстрого создания проекта;
 - Справочная информация.
- Каждый проект состоит из трех поддиректорий:
- inc – в ней хранятся заголовочные файлы проекта, включая конфигурационные файлы для сенсорной библиотеки (STM8_TSL_RC_Configuration.h) и библиотеки периферийных устройств (stm8s_conf.h);
 - src – в ней хранятся все файлы исходных текстов, включенные и используемые в проекте (main.c, stm8_interrupt_vector.c и др.);
 - STVD\(\Raisonance, Cosmic) – в ней хранятся рабочие файлы проектов, областей, настроек (discover.stw).

Итак, копируем папки «Libraries» и «Project» в рабочую область, где мы предполагаем их использовать, например, в следующее местоположение: «C:\Program Files\STMicroelectronics\st_toolset». Теперь запускаем среду разработки «Пуск → Программы → ST Toolset → Development Tools – ST Visual Develop», кликаем «File → Open Workspace...» и открываем проект «discover» из папки «C:\Program Files\STMicroelectronics\st_toolset\Project\discover\STVD\Raisonance» или «C:\Program Files\STMicroelectronics\st_

toolset\Project\discover\STVD\Cosmic». Теперь мы можем скомпилировать, собрать, прошить микроконтроллер и запустить программу на исполнение. Сборка проекта должна пройти без каких-либо проблем, единственное, что понадобится сделать – это выбрать программатор-отладчик ST-Link через «Debug instrument → Target Settings → Swim ST-Link» (рис. 4). Если по каким-то причинам сборка проекта произошла с ошибками, ниже будут приведены все настройки проекта.

Самостоятельная настройка проекта

Если вы решили разместить проект, библиотеки, исходные и заголовочные файлы по вашему собственному усмотрению или создать проект «с нуля», то у вас могут возникнуть проблемы с линковкой файлов библиотек. Все пути к этим файлам можно прописать вручную, но это слишком загромождает проект и делает ваш код менее наглядным. Ниже будут приведены основные настройки проекта для каждого из Си-инструментария в отдельности, и начнем мы с настройки инструментария от Raisonance. Заметим, что в качестве примера будет использоваться проект «discover», скачанный по вышеприведенной ссылке. Переходим на «Projects → Settings...» (рис. 4).

В поле «Toolset» выбирается Си-инструментарий с необходимыми настройками, в нашем случае выбран инструментарий от Raisonance. В поле «Root path» указывается путь к Си-инструментариям, а в «Output directory» – режим конфигурации проекта, для которого и будут применяться все настройки; на рисунке 4 выбран режим «debug».

Следующая вкладка «Projects → Settings... → Debug». В данной вкладке нас интересует поле «Source Directories:», где прописываются пути к используемым файлам проекта с исходным кодом. Для проекта «discover» – это «C:\Program files\stmicroelectronics\st_toolset\project\discover\src», «C:\Program files\stmicroelectronics\st_toolset\libraries\stm8s_stdperiph_driver\src» и «C:\Program files\stmicroelectronics\st_toolset\libraries\stm8_touchsensing_driver\src».

Следующая вкладка «Projects → Settings... → MCU Selection». Здесь выбирается тип микроконтроллера – **STM8S105C6** для выбранного нами оценочного комплекта.

Следующая вкладка – «Projects → Settings... → C Compiler» (рис. 5). В поле «Category» необходимо выбрать «Preprocessor» и далее в поле «Additional include directories» прописать пути к заголовочным файлам, используемым в проекте. Для проекта «discover» – это «C:\Program Files\STMicroelectronics\st_toolset\Project\discover\inc», «C:\Program Files\STMicroelectronics\st_toolset\Libraries\STM8_TouchSensing_Driver\inc» и «C:\Program Files\STMicroelectronics\st_toolset\Libraries\STM8S_StdPeriph_Driver\inc».

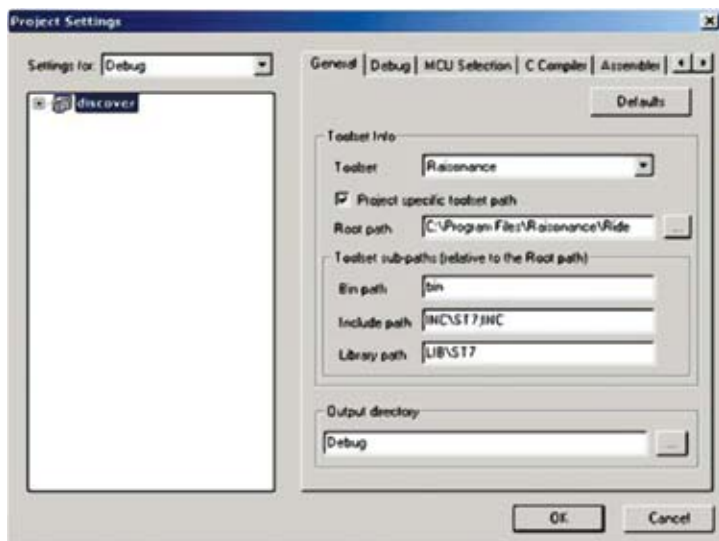


Рис. 4. Свойства проекта

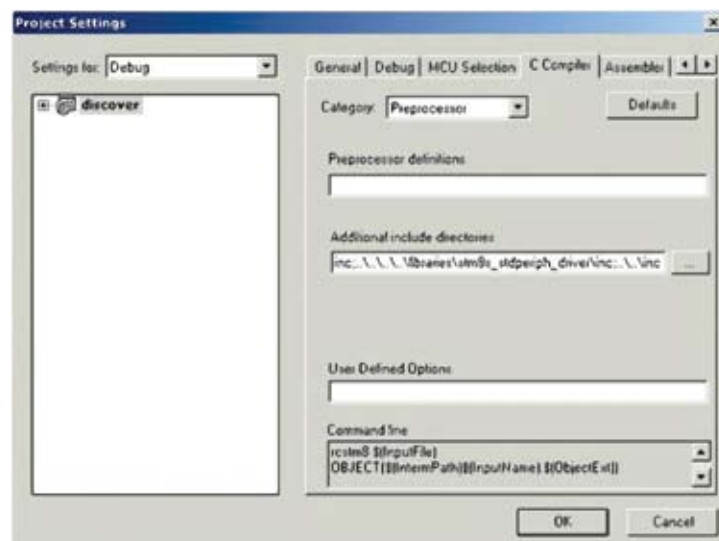


Рис. 5. Свойства проекта

И последняя вкладка — «Projects → Settings... → Linker». Если вы используете ограниченно-бесплатный Си-инструментарий, то необходимо убрать флажок «Run Code Compressor», иначе при сборке проекта будет сгенерирована ошибка.

Что касается использования Си-инструментария от Cosmic Software, необходимо всего лишь произвести некоторые изменения в поле «General» (рис. 6), остальные настройки идентичны приведенным выше для Raisonance. В частности в поле «Toolset» выбирается Си-инструментарий от Cosmic software и соответствующие настройки.

В разделе, посвященном настройке проекта, были затронуты лишь основные моменты. К сожалению, описать все настройки и тонкости не хватит места, да и в этом, собственного говоря, нет необходимости. Для более детальной информации вам следует обратиться к руководству описания среды ST Visual Developer. Что касается приведенных выше настроек проекта — путей к файлам, вы можете пересобрать и перекомпоновать проект по вашему усмотрению, нужно лишь прописать правильные пути к исходным и заголовочным файлам. По мере работы со средой и приобретения опыта вы сами разберетесь, какие настройки для чего необходимы.

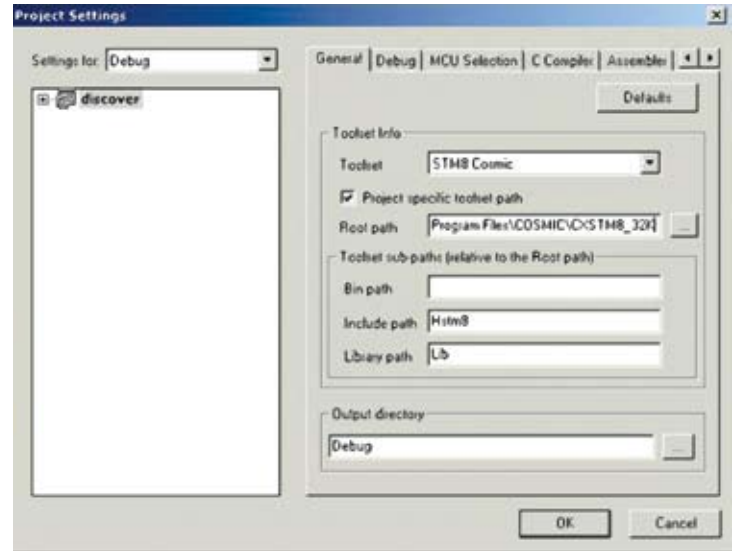


Рис. 6. Свойства проекта

Создание нового проекта, конфигурирование библиотек

Возможно, по каким-либо причинам у вас не возникнет желания использовать предоставляемые производителем библиотеки. В таком случае вам понадобится следующая информация. Все файлы с описанием регистров находятся в папке «C:\Program Files\STMicroelectronics\st_toolset\include». Пути для Си-компилятора прописываются в свойствах проекта «Project → Settings → C compiler → Category Preprocessor». Свойства линковщика находятся в «Project → Settings → Linker → Category Input».

Для создания собственного проекта с использованием библиотек наиболее простым и быстрым методом является использование шаблона проекта «Project_template», расположенного в скачанном пакете в папке «Project». При создании проекта вы вручную создаете папку с названием вашего проекта и копируете туда содержимое проекта «Project_template». Затем в среде ST Visual Developer открываем проект из вновь созданной папки (рис. 7). Структура проекта состоит из следующих директорий:

- Source Files — содержит исходные файлы проекта;
- Source Files\FWLib — содержит исходные файлы библиотеки периферии;
- Source Files\TSLib — содержит исходные файлы сенсорной библиотеки;
- Include Files — содержит заголовочные файлы проекта;
- Include Files\FWLib — содержит заголовочные файлы библиотеки периферии;
- Include Files\STLib — содержит заголовочные файлы сенсорной библиотеки.

Далее нам необходимо выбрать в свойствах проекта программатор-отладчик, Си-инструментарий, тип микроконтроллера, прописать пути к файлам, если они не указаны. Эти операции описывались выше.

Для использования библиотеки стандартной периферии необходимо убедиться, что в основном файле проекта «main.c» был подключен файл «stm8s.h» (#include <stm8s.h>), который содержится в директории «inc» библиотеки периферии микроконтроллера. Также файл «stm8s.h» необходимо добавить в поддиректорию «Include Files\FWLib», кликнув правой кнопкой мыши на папке, и выбрать в выпадающем меню «Add Files to Folder...» (рис. 7).

Шаблон проекта подразумевает, что разработчик будет использовать библиотеку стандартной периферии микроконтроллера. В другом случае вам нужно закомментировать строку

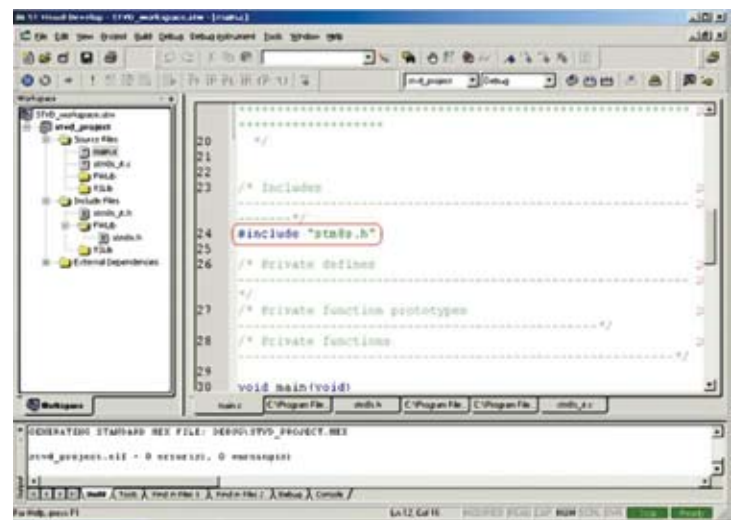


Рис. 7. Создание нового проекта

«#define USE_STDPERIPH_DRIVER» в файле «stm8s.h». Если вы все же будете использовать библиотеку стандартной периферии, то вам необходимо выполнить следующие шаги:

- 1) Убедиться, что строка «#define USE_STDPERIPH_DRIVER» в файле «stm8.h» раскомментирована;
- 2) Добавить файл «stm8s_conf.h» из директории вашего проекта «My Project\inc» в папку корневого каталога «Include Files» проекта с помощью операции «правая кнопка мыши → Add Files to Folder...». Для использования какого-либо периферийного модуля необходимо раскомментировать его название в файле «stm8s_conf.h» (рис. 8);
- 3) Вся периферия и функции работы с ней описаны в двух файлах: заголовочном (расширение .h) и исходном (расширение .c) с именем «stm8s_PeriphType». Данные файлы расположены в директориях «inc» и «src» по следующему пути «C:\Program Files\STMicroelectronics\st_toolset\Libraries\STM8S_StdPeriph_Driver». При использовании любого периферийного модуля в вашем проекте необходимо добавить исходный файл в поддиректорию «Source Files\FWLib», а заголовочный — в «Include Files\FWLib» (рис. 8). Данные операции выполняются кликом правой кнопкой мыши на соответствующей папке: «Правая кнопка мыши → Add Files to Folder...»

После вышеперечисленных действий вы можете использовать функционал библиотеки периферии. В примере «discover» используется сенсорная кнопка, работа с которой реализована в специальной библиотеке. Для использования библиотеки ее необходимо сконфигурировать следующим образом:

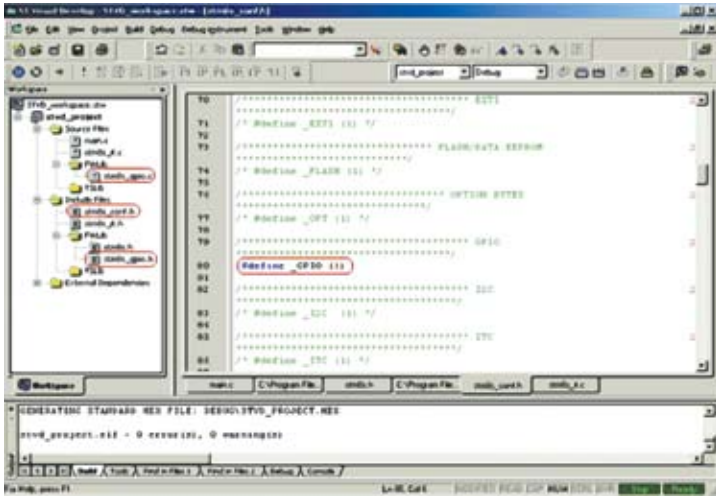


Рис. 8. Конфигурирование библиотеки периферии

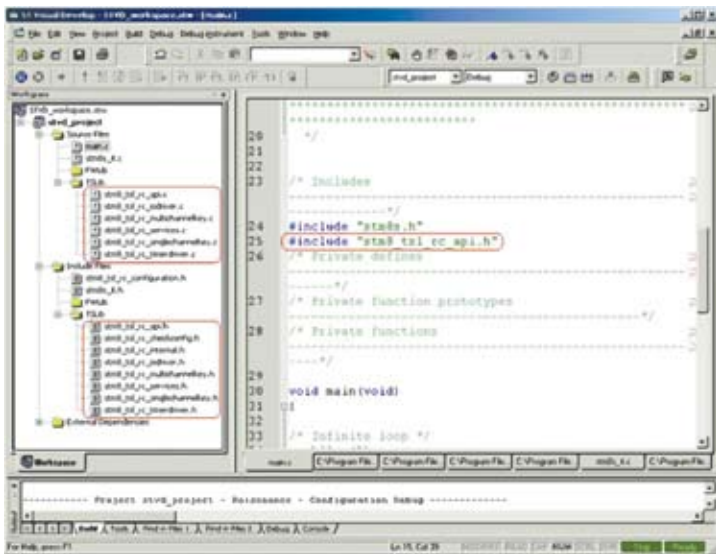


Рис. 9. Конфигурирование сенсорной библиотеки

1) Сконфигурировать библиотеку в соответствии с количеством сенсорных кнопок (до 24), ползунков (до двух) и колец (до двух):

- Скопировать файл «STM8_TSL_RC_Configuration_TOADAPT.h» из директории «C:\Program Files\STMicroelectronics\st_toolset\Libraries\STM8_TouchSensing_Driver\inc» в директорию проекта «C:\Program Files\STMicroelectronics\st_toolset\Project\new_project\inc» и переименовать данный файл в «STM8_TSL_RC_Configuration.h»;

- Сконфигурировать файл «STM8_TSL_RC_Configuration.h» в соответствии с аппаратными особенностями проекта;

- Добавить файл «STM8_TSL_RC_Configuration.h» в директорию «Include Files» проекта: «правая кнопка мыши→Add Files to Folder...»;

- При использовании инструментария от Raisonance добавить файлы с расширением «.h» и «.c», исключая файл «STM8_TSL_RC_Configuration_TOADAPT.h» из сенсорной библиотеки C:\Program Files\STMicroelectronics\st_toolset\Libraries\STM8_TouchSensing_Driver», соответственно, в директории «Include Files\TSLib» и «Source Files\TSLib» (рис. 9). При использовании инструментария от Cosmic Software файл «STM8_TSL_RC_routines.asm» также не добавляется, вместо него необходимо сконфигурировать «Project → Settings → Linker», что будет описано ниже;

- Добавить строку «#include <stm8_tsl_rc_api.h>» в основной файл проекта «main.c»;

- Некоторые функции библиотеки должны быть расположены в памяти по специальным адресам при использовании инструментария Cosmic software (рис. 10):

- Выбираем «Project → Settings → Linker»;
- На вкладке «Linker» в поле «Category» выбираем «Input»;

- В поле «Vector File name» прописываем путь к файлу «stm8_interrupt_vector.c»;

- В окне «Segment/Section name» добавляем «TSL_IO_ALCODE» в секции «Code,Constant» с опцией «-r2»;

- При использовании инструментария Raisonance в операциях с «Project → Settings → Linker» нет необходимости. Необходимо всего лишь добавить файл «STM8_TSL_RC_routines.asm» в директорию проекта «Source Files\FWLib».

Общую настройку проекта с обеими библиотеками вы можете посмотреть на примере «discover» в загруженном пакете программного обеспечения.

Проект «discover»

Алгоритм работы программного обеспечения проекта следующий: программа ожидает срабатывания нажатия сенсора TS1 и меняет частоту мерцания светодиода LD1. Рассмотрим проект более подробно. Открываем среду разработки ST Visual Developer, затем «File → Open Workspace...» и выбираем проект по следующему пути C:\Program Files\STMicroelectronics\st_toolset\Project\discover\STVD\Raisonance». Как вы можете заметить, у данного проекта структура фактически идентична шаблону проекта «project_template» и используются обе библиотеки – сенсорная библиотека и библиотека периферии. Рассмотрим более подробно основные моменты основного файла проекта «main.c».

```
#include <stm8s.h>
// Заголовочный файл библиотеки периферии
#include <stm8_tsl_rc_api.h>
// Заголовочный файл сенсорной библиотеки
// Макроопределения
.....
// Объявление функций
.....
// Объявление глобальных переменных
.....
// Главная функция
void main(void)
{
//Конфигурация тактового сигнала
CLK_Configuration();
// Конфигурация портов ввода/вывода
GPIO_Configuration();
// Инициализация сенсорной библиотеки
TSL_Init();
// Инициализация сенсорных кнопок
ExtraCode_Init();
// Старт таймера для управления частотой
мерцания светодиода LD1 с периодом 100 мс
TSL_Tick_Flags.b.User_Start_100ms = 1;
// Основной цикл программы
for (;;)
{
// Код разработчика
ExtraCode_StateMachine();
// Главная функция сенсорной библиотеки
TSL_Action();
}
}
// Реализация объявленных функций
.....
```

Программы для микроконтроллеров имеют похожие структуры, точкой входа программы является функция «main()». Для

начала работы с микроконтроллером, в зависимости от задачи, его необходимо сконфигурировать — настроить тактовые частоты, периферийные модули, разрешить прерывания, если необходимо. После этого уже запускается т.н. диспетчер, обычно — это бесконечный цикл «while(1){...}» или «for(;;){...}», в контексте которого уже и выполняются функции основной программы — реакция на события, выполнение вычислений и т.д.

В проекте из библиотеки стандартной периферии используются функции для настройки тактового сигнала, настройка работы с портами ввода/вывода. С помощью функции «CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1)» в качестве источника тактового сигнала выбирается внутренний генератор HSI, работающий на частоте 16 МГц, коэффициент деления равен 1, соответственно рабочая частота микроконтроллера равна 16 МГц. Для работы со светодиодом LD1 необходимо настроить соответствующую ему ножку ввода/вывода. Данная операция выполняется с помощью функции библиотеки «GPIO_Init(GPIOD, GPIO_PIN_0, GPIO_MODE_OUT_PP_LOW_FAST)». По передаваемым параметрам функции видно, что используется ножка «0» порта «D» микроконтроллера, режим push — pull, режим быстрого переключения (до 10 МГц), и в данный момент на ее выходе — сигнал низкого уровня. Переключение состояния ножки осуществляется с помощью функции «GPIO_WriteReverse(GPIOD, GPIO_PIN_0)».

Процесс работы с сенсорной библиотекой начинается с ее инициализации: «TSL_Init()» — инициализация памяти, функций и структур данных. Далее необходимо настроить библиотеку согласно аппаратным требованиям. В нашем случае библиотека настраивается под сенсорную кнопку TS1 в функции ExtraCode_Init(). После того, как все настроено, можно начинать работать с библиотекой, которая предоставляет API для работы с ней, нам в частности необходимо определять событие нажатия TS1. Данная операция осуществляется в функции «ExtraCode_StateMachine()», также здесь задается режим мерцания светодиода LD1. Для работоспособности сенсорной библиотеки необходимо включать в основной цикл програм-

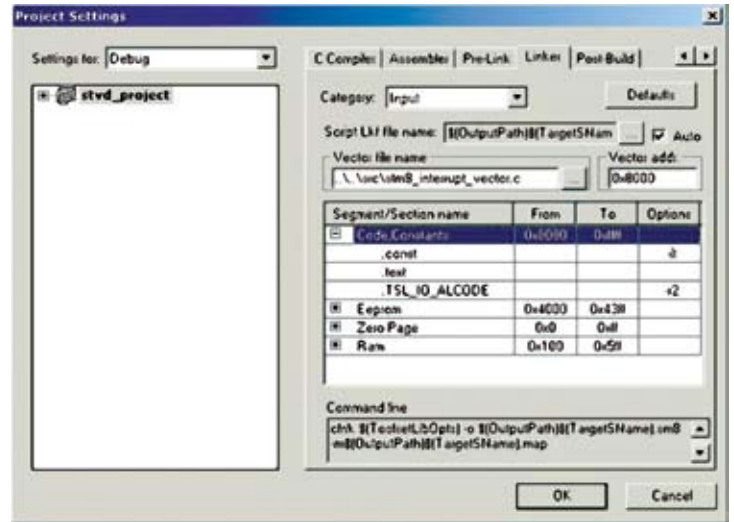


Рис. 10. Конфигурирование сенсорной библиотеки для Cosmic software

мы (диспетчер) функцию «TSL_Action()», которая выполняет необходимые операции для обработки и определения событий сенсорных кнопок и других сенсорных объектов.

Заключение

В статье мы представили необходимую информацию для начала работы с микроконтроллерами STM8. В материале были рассмотрены лишь основные моменты, для более подробного и глубокого ознакомления читателю следует обращаться к справочным материалам, доступным на сайте <http://www.st.com/mcu/>.

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



НОВЫЕ 8-БИТНЫЕ МК СО СВЕРХНИЗКИМ ЭНЕРГОПОТРЕБЛЕНИЕМ



- Ядро STM8, 16 МГц
- От 4 до 32 Кбайт встроенной Flash, до 2 Кбайт SRAM
- $U_{пит}$ от 1,8 до 3,6 В (1,65 В в режиме «power down»)
- $I_{потр}$ при сохранении данных ОЗУ 350 нА
- Энергопотребление в активном режиме 150 мкА/МГц
- 12-битные АЦП и ЦАП
- Диапазон рабочих температур от -40...85°C или до 125°C

Москва
Тел.: (495) 995-0901
Факс: (495) 995-0902

Санкт-Петербург
Тел.: (812) 327-9404
Факс: (812) 327-9403

Компэл
www.compel.ru