

# Программные средства разработки для микроконтроллеров Precision32 фирмы Silicon Labs

С недавнего времени компания Silicon Labs начала выпуск 32-разрядных микроконтроллеров Precision32 на базе ядра Cortex-M3, а также средств поддержки разработки для них (рис. 1). Аппаратные средства включают в себя программаторы и оценочные платы с возможностью подключения модулей, расширяющих их функционал, а в набор программных средств, получивший название Precision32 Development Suite, входят мастер создания кода начальной конфигурации, интегрированная среда разработки и библиотека для более быстрой разработки кода под эти 32-разрядные микроконтроллеры. Программные продукты предоставляются на бесплатной основе, но для полноценного использования некоторых из них необходимо пройти бесплатную активацию на сайте [www.silabs.com](http://www.silabs.com). В статье рассмотрены программные средства, в состав которых входят все необходимые компоненты для разработки проекта под 32-разрядные микроконтроллеры компании Silicon Labs.

Сергей СОЛОДКОВ  
ssa@efo.ru

## Общая информация

Precision32 Development Suite («Набор разработчика Precision32») — это готовая система разработки для 32-разрядных микроконтроллеров Silicon Labs, включающая в себя интегрированную среду разработки (IDE), мастер создания кода AppBuilder и программный набор разработки (Software Development Kit).

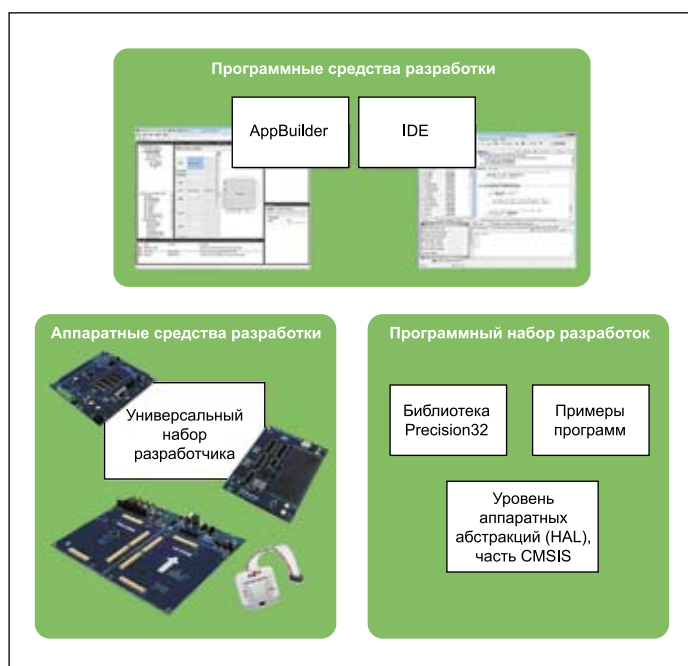


Рис. 1. Набор разработчика

Набор разработчика Precision32 доступен для скачивания на сайте производителя [1].

Отладка проекта осуществляется в интегрированной среде с помощью аппаратных средств поддержки разработки.

На рис. 2 изображены средства поддержки разработки, которые могут быть использованы для микроконтроллеров Precision32. Программный продукт AppBuilder создает начальный код для микроконтроллеров в простом и удобном графическом интерфейсе. Сгенерированный код в типовой реализации импортируется в Precision32 IDE, где его можно отредактировать. Готовый проект загружается во Flash-память микроконтроллера с помощью внутреннего эмулятора USB Debug Adapter.

Существуют и сторонние среды, такие как Keil uVision и IAR Embedded Workbench, которые предоставляют альтернативные пути разработки (рис. 2).

## Precision32 AppBuilder

Precision32 AppBuilder — программное обеспечение, генерирующее код начальной конфигурации микроконтроллера и упрощающее дальнейшую реконфигурацию его периферийных блоков.

Precision32 AppBuilder позволяет сконфигурировать необходимую периферию в графическом интерфейсе, например установить режим тактирования и настроить порты ввода/вывода. После задания параметров программа генерирует код на языке C, который упростит дальнейшую разработку программного кода для микроконтроллера.

Precision32 AppBuilder обладает следующими возможностями:

- Создание шаблонов конфигурации для различных режимов работы микроконтроллера.
- Оповещение о необходимости дополнительной настройки периферии, с которой взаимосвязан конфигурируемый блок.
- Просмотр сгенерированного исходного кода в режиме чтения с подсветкой синтаксиса.

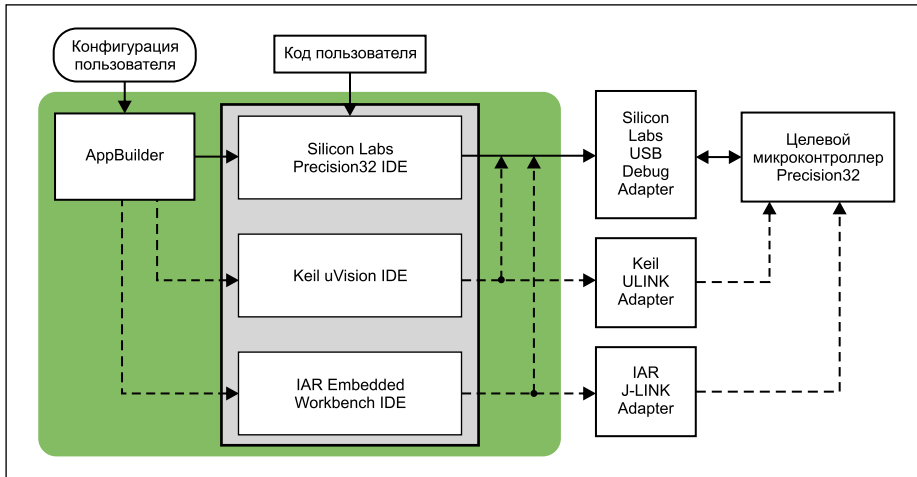


Рис. 2. Схема разработки программ для микроконтроллеров Precision32

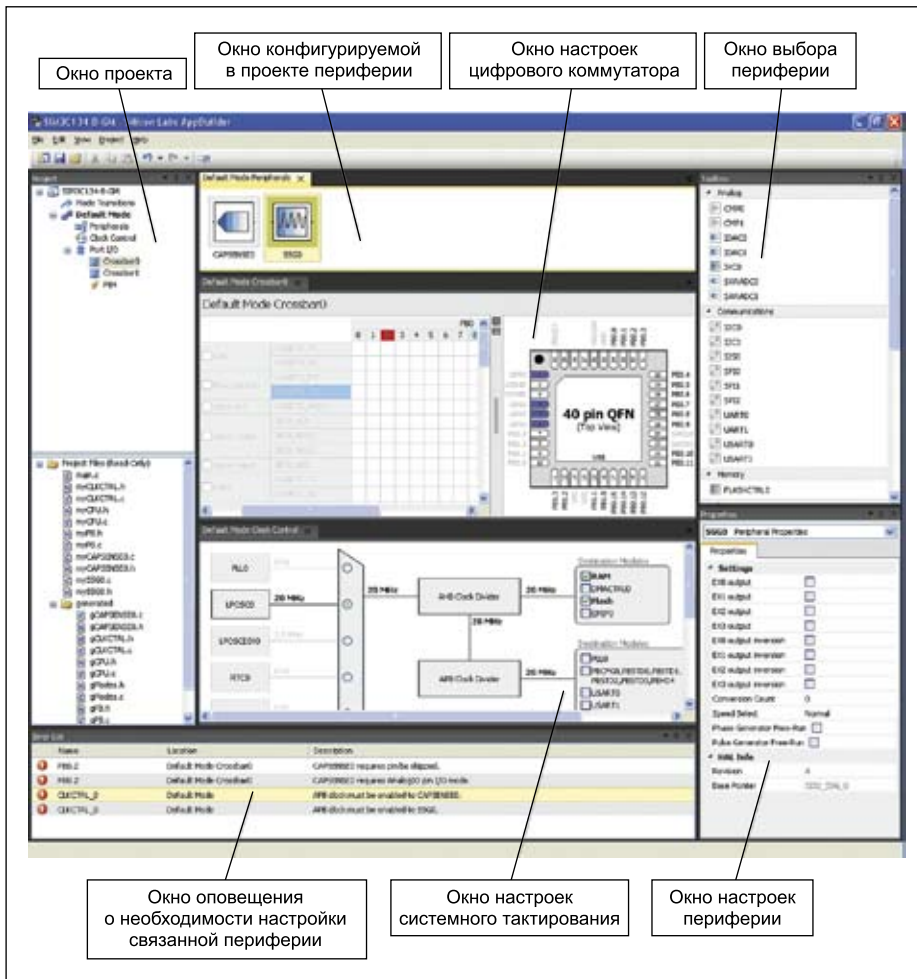


Рис. 3. Пользовательский интерфейс AppBuilder

Precision32 AppBuilder предназначен для создания кода только для 32-разрядных микроконтроллеров Silicon Labs. Графический интерфейс включает в себя несколько окон с отображением структуры проекта, списка доступной периферии и ее свойств (рис. 3).

Каждое действие, производимое с периферией и ее настройками в графическом интерфейсе, отображается в сгенерированном коде

в режиме реального времени. При добавлении периферийного блока в проект программа анализирует необходимость изменения каких-либо настроек периферии, связанной с добавленным блоком, и при наличии такой необходимости уведомляет об этом разработчика. Например, система настройки источника тактирования предлагает выбрать нужный источник тактирования и значение

делителя частоты, а также выбрать периферию, на которую следует подать сигнал с источника тактирования.

В отдельное окно графического интерфейса вынесена настройка коммутатора (crossbar).

AppBuilder позволяет описать несколько режимов работы микроконтроллера конфигурациями, в каждой из которых задаются свои настройки периферии. Существующие конфигурации отображаются в окне структуры проекта, добавить которые можно в этом же окне. Выбор необходимого режима работы в определенный момент времени осуществляется в коде вызовом функции с интуитивно понятным названием, указывающим на нужный режим, например, при названии режима Default mode соответствующая этому режиму функция будет иметь название `enter_default_mode_from_reset`. В данном случае часть названия `_from_reset` указывает на режим, из которого микроконтроллер переходит в эту конфигурацию.

Полученные заготовки проектов AppBuilder далее экспортируются в интегрированную среду разработки Precision32.

### Интегрированная среда разработки Precision32

Интегрированная среда разработки Precision32 построена на базе платформы Eclipse и включает в себя необходимые средства для компиляции проекта и его отладки с использованием фирменных средств поддержки разработки. В среду входит библиотека `si32 Library`, которая необходима для написания программ для 32-разрядных микроконтроллеров. Все средства, включенные в среду разработки, предназначены для разработки на языке C стандарта C99.

Графический интерфейс программы является типовым интерфейсом платформы Eclipse (рис. 4). Окно программы поделено на несколько частей: окно структуры проекта, окно вспомогательных инструментов и основное окно с программным кодом.

Редактор кода поддерживает подсветку синтаксиса.

В Precision32 IDE присутствует панель (Dashboard) со ссылками на документацию для аппаратных и программных средств, исходные коды примеров программ и видеоролики. Precision32 IDE может как создавать собственные проекты с нуля, так и импортировать заготовки из мастера создания кода AppBuilder. После написания кода и сборки проекта программу можно загрузить и отладить на микроконтроллере непосредственно из среды Precision32.

Все описанные средства предоставляются бесплатно, но для полноценного использования среды разработки Precision32 IDE необходима ее бесплатная активация.

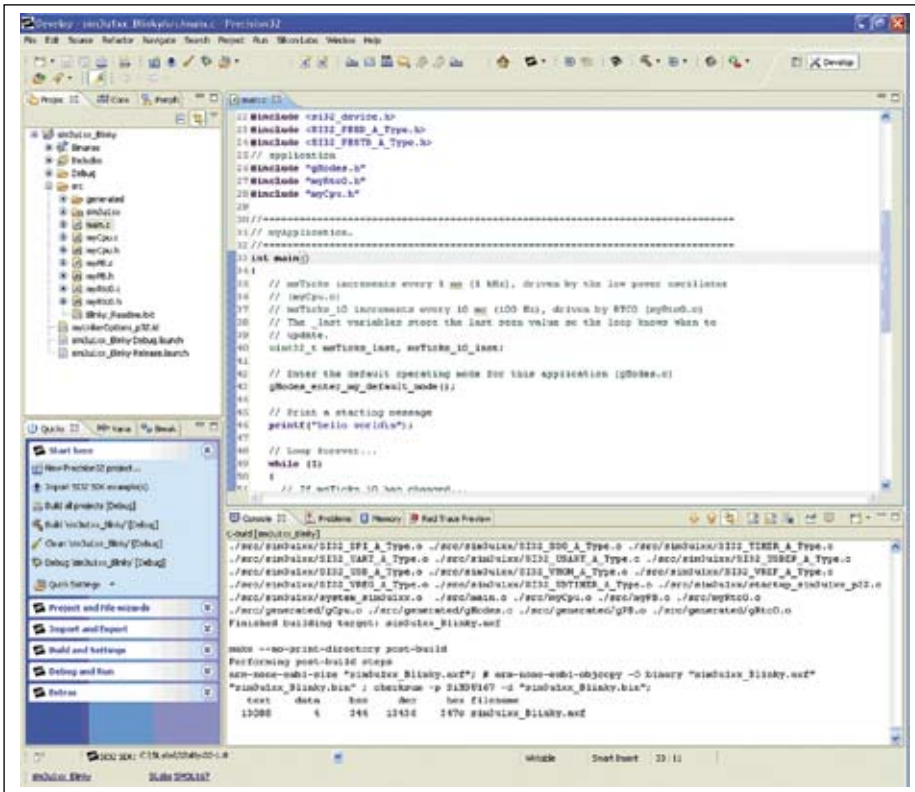


Рис. 4. Пользовательский интерфейс IDE Precision32

**Программный набор разработки (SDK) Precision32**

Программный набор разработки (SDK) — это несколько средств для упрощения разработки проекта для 32-разрядных микроконтроллеров Silicon Labs. SDK состоит из трех частей: HAL (Hardware Access Layer, уровень аппаратных абстракций), библиотеки si32Library и примеров программ. SDK устанавливается вместе с пакетом программного обеспечения Precision32 и располагается в папке si32-x.y инсталляционной директории, где x и y указывают на версию SDK. Это позволяет устанавливать новые версии SDK в отдельные папки, чтобы не переписывать исходные коды существующих проектов при изменении форматов вызовов процедур и функций.

На рис. 5 изображена структурная схема SDK.

**CMSIS и HAL**

CMSIS — это стандарт программного интерфейса для микроконтроллеров Cortex, упрощающий перенос кода с одной линейки контроллеров на базе ядра Cortex на другую. HAL является частью CMSIS и представляет собой библиотеку процедур управления аппаратными средствами. HAL дает доступ к регистрам периферии SIM3xxxx. Такой стандарт разработан для замены битов регистров специальных функций периферийных блоков на функциональный элемент программы, имя которого объясняет назначение бита. Функции и макросы HAL не являются пото-

кобезопасными и не запрещают прерывания при динамической модификации регистров.

HAL является программным уровнем над аппаратным средством. Процедуры CMSIS и HAL используются в примерах программ, а также в библиотеке Precision32.

На рис. 6 представлена блок-схема программных уровней.

HAL организован в виде набора общих модулей для однотипных периферийных блоков SIM3xxxx. В файлах модуля описаны общие регистры однотипных блоков и реализованы процедуры для работы с ними.

Регистры BP (Base point) и векторы прерываний конкретных периферийных блоков содержатся в заголовочных файлах с именем микроконтроллера. Например, в файлах модуля USART\_A содержатся описание регистров и реализация процедур общих для всех блоков USART, а адрес каждого регистра описан смещением относительно базового адреса, указывающего на начало описания соответствующего блока USART.

Первичная инициализация процессора в Si32Hal не соответствует рекомендациям CMSIS в связи с поддержкой разных программных средств разработки. Каждая поддерживаемая среда разработки имеет возможность настройки стека, кучи, ОЗУ, retention RAM, а также интерфейса внешней памяти. Для указания модели и размера выделяемой памяти HAL использует файлы линкера, которые предоставляются вместе с примерами программ.

Для многих процедур HAL реализует как функции, так и макросы. Это позволяет в программе, которая вызывает HAL, выбирать между быстрым макросом и компактной функцией. Для некоторых процедур макросы реализованы в виде вызова функции.

Версия HAL может быть выбрана опционально в проекте AppBuilder.

**Библиотека Precision32**

Набор библиотек Precision32 включает в себя готовые программные модули в исходных кодах на языке C стандарта C99, обеспечивающие функционал для сбора и передачи данных, оповещения о событиях, организации многозадачности, коммуникаций по интерфейсу USB, обработки ошибок, записи в журнал значений переменных в течение работы программы и т.д.

Библиотека используется через стандартные вызовы функций и упрощает портирование проектов между различными сериями микросхем.

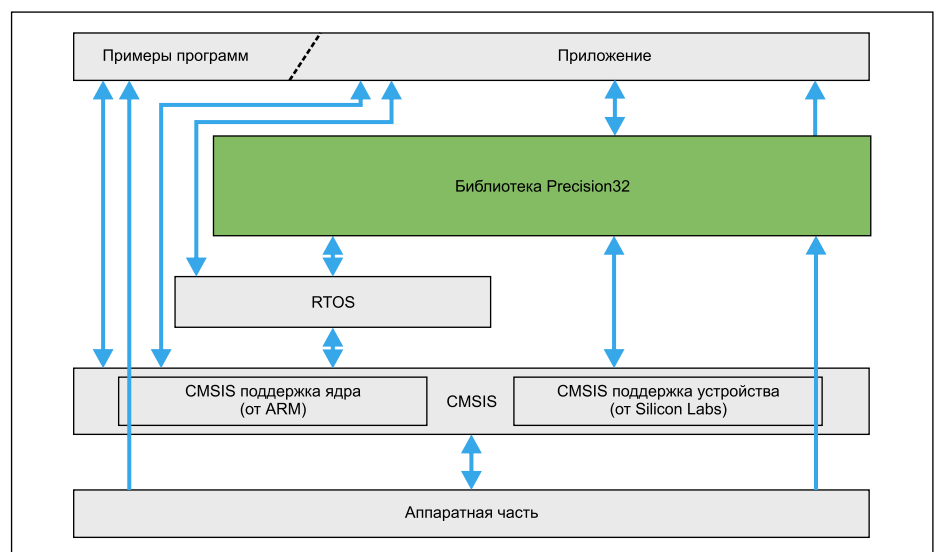


Рис. 5. Структурная схема программного набора разработки (SDK)

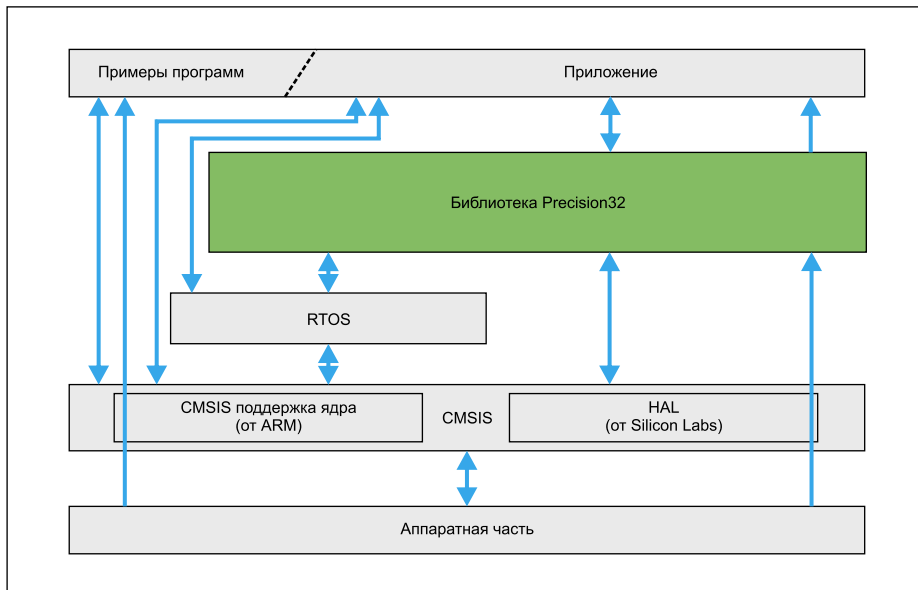


Рис. 6. Блок-схема программных уровней

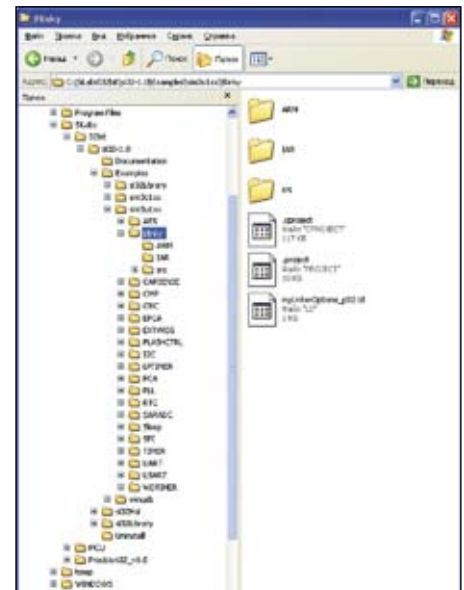


Рис. 7. Структура каталогов набора кодов программ

Si32Library построена как набор независимых и взаимодействующих компонентов.

Компоненты реализованы в виде наборов файлов со стандартными для языка C расширениями. Компоненты могут быть разделены на субкомпоненты. Субкомпоненты служат для объединения компонентов в логические группы.

Библиотека включает в себя файлы параметров сборки приложения. Параметры устанавливаются для каждого приложения индивидуально. Например, с помощью этих файлов можно установить параметры стека и кучи. Наличие таких файлов позволяет конфигурировать библиотеку для каждого приложения индивидуально, без необходимости изменять какой-либо ее код.

### Примеры кодов программ

Примеры кодов программ являются частью SDK, установленного вместе с пакетом программного обеспечения Precision32. Примеры кодов реализуют типовые задачи, демонстри-

рующие режимы работы и возможности периферии. Они предоставляются в виде модулей, поэтому код можно скопировать в пользовательский проект, что облегчит написание кода типовых функций. В директориях примеров содержатся папки, в которых расположены проекты для uVision и IAR (рис. 7).

### Заключение

Подводя итог всему изложенному, нужно сказать, что компания Silicon Labs полностью обеспечила разработчиков программным обеспечением для создания проектов под 32-разрядные микроконтроллеры. Благодаря программному инструменту AppBuilder разработка будет проходить с меньшими затратами сил и времени, а библиотека Precision32 упростит перенос кода проекта между различными сериями микросхем.

Кроме того, набор разработки включает достаточное количество примеров кода для ознакомления со всеми возможностями микроконтроллеров Precision32.

Эти программные компоненты привычны для многих реализаций микроконтроллеров с различными версиями ядра ARM от разных производителей, что сделано для того, чтобы разработчик, осваивающий эту серию микроконтроллеров фирмы Silicon Labs, сфокусировался на освоении новых возможностей самой линейки кристаллов и не тратил время на изучение особенностей программного обеспечения для них. ■

### Литература

1. [www.silabs.com/32bit-software](http://www.silabs.com/32bit-software)
2. AN673: Precision32 Software Development Kit Overview
3. AN664: Precision32 CMSIS and HAL User's Guide
4. AN672: Precision32 si32Library Overview
5. AN668: Precision32 Software Development Kit Code Examples Overview
6. AN670: Getting Started with the Silicon Labs Precision32 AppBuilder
7. AN667: Getting Started with the Silicon Labs Precision32 IDE