

Степанов А.Б., Суворов К.А.

**ЦИФРОВАЯ ОБРАБОТКА
СИГНАЛОВ В
РАДИОТЕХНИЧЕСКИХ
СИСТЕМАХ**

***РЕАЛИЗАЦИЯ АЛГОРИТМОВ НА ЭЛЕ-
МЕНТНОЙ БАЗЕ СО СВЕРХНИЗКИМ
ЭНЕРГОПОТРЕБЛЕНИЕМ***

Лабораторный практикум по дисциплине

СПбГУТ

Лабораторная работа №1

Знакомство с интегрированной средой разработки Code Composer Studio

Цель работы: получить навыки создания проекта в интегрированной среде разработки Code Composer Studio.

1.1. Краткая теоретическая справка

Интегрированная среда разработки Code Composer Studio (CCS) предназначена для написания программного кода для всех цифровых сигнальных процессоров и микроконтроллеров, выпускаемым компанией Texas Instruments (TI) [1]. Она включает такие инструменты, как:

- редактор исходных кодов;
- компилятор;
- компоновщик;
- отладчик;
- средства для построения графиков;
- другие инструменты.

Code Composer Studio – это единый графический пользовательский интерфейс (рис. 1), позволяющий производить пошаговую разработку и отладку программного кода [2]. Code Composer Studio базируется на кроссплатформе Eclipse с открытым исходным кодом. Для микроконтроллеров MSP430G2xx существует свободная для скачивания лицензия с ограничением объема кода – не более 16 Кбайт.

Реализация системы управления или обработки сигналов с использованием Code Composer Studio, как правило, включает следующие этапы:

1. Подключение отладочной платы с установленным микроконтроллером или цифровым сигнальным процессором к компьютеру.
2. Запуск CCS.
3. Создание нового проекта.
4. Создание нового исходного файла.
5. Набор текста программы в окне редактирования исходного кода.
6. Настройка подключения к отладчику.
7. Сборка проекта.
8. Отладка приложения.
9. Запуск исполняемой программы.

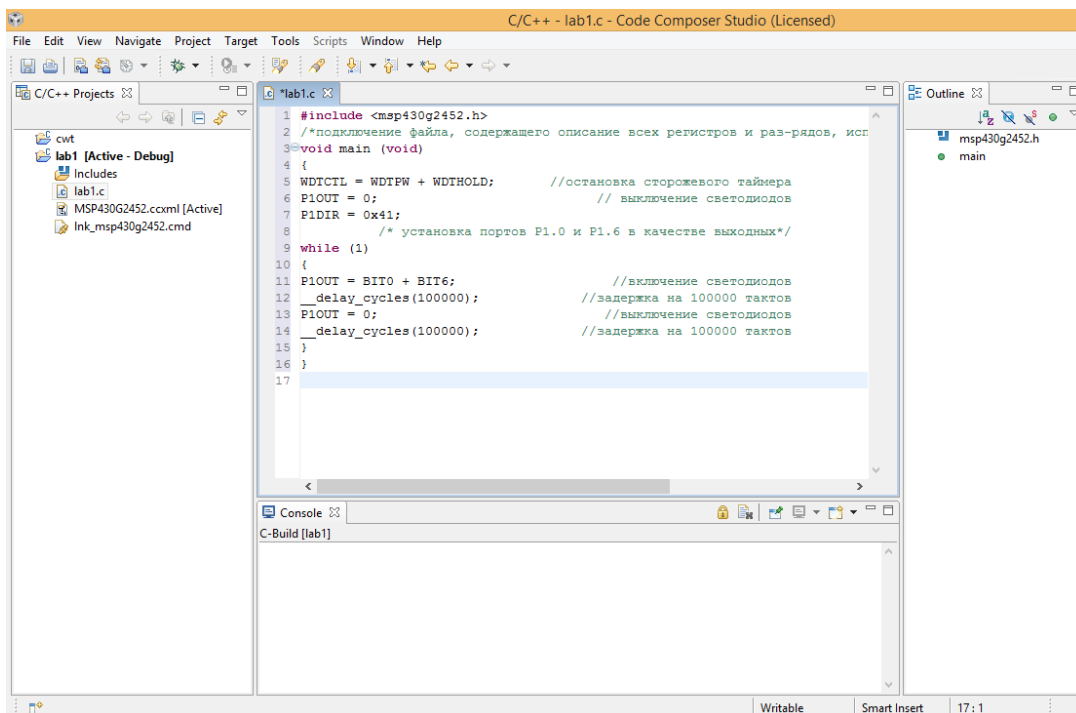


Рис. 1. Основное окно Code Composer Studio

10. Приостановка выполнения исполняемой программы.

11. Остановка исполняемой программы и закрытие отладчика.

12. Выход из CCS.

В большинстве случаев работа реализованной системы должна осуществляться независимо от персонального компьютера, а иногда полностью автономно. Это учитывается при написании программного кода и при распределении памяти микроконтроллера или цифрового сигнального процессора.

1.2. Задание на лабораторную работу

Выполнение лабораторной работы осуществляется в следующем порядке:

1. Подключите отладочную плату MSP-EXP430G2 к компьютеру.
2. Запустите Code Composer Studio: «Start → All Programs → Texas Instruments → Code Composer Studio».
3. Создайте новый проект: «File → New → CCS Project» (рис. 2).
4. В поле «Project name» введите имя проекта, нажмите «Next», установите тип проекта: «Project Type → MSP430». Нажмите «Next» два раза для перехода к странице настроек проекта – «Project Settings». В поле «Device Variant» выберите необходимое устройство из списка – семейство и модель микроконтроллера, например, MSP430G2452 (рис. 3). Модель микроконтроллера можно уточнить, обратившись к маркировке микросхемы. Далее нажмите «Finish».

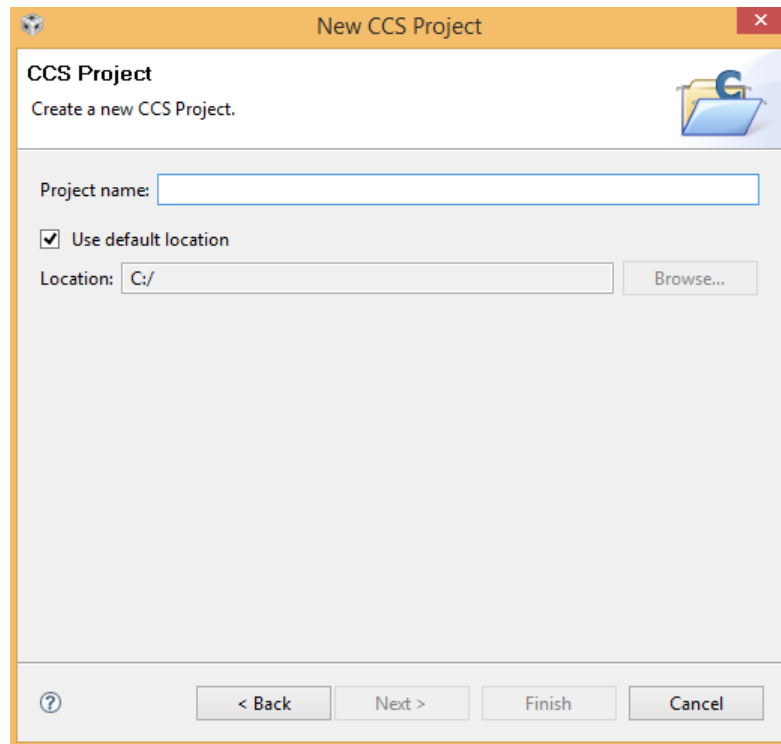


Рис. 2. Создание нового проекта

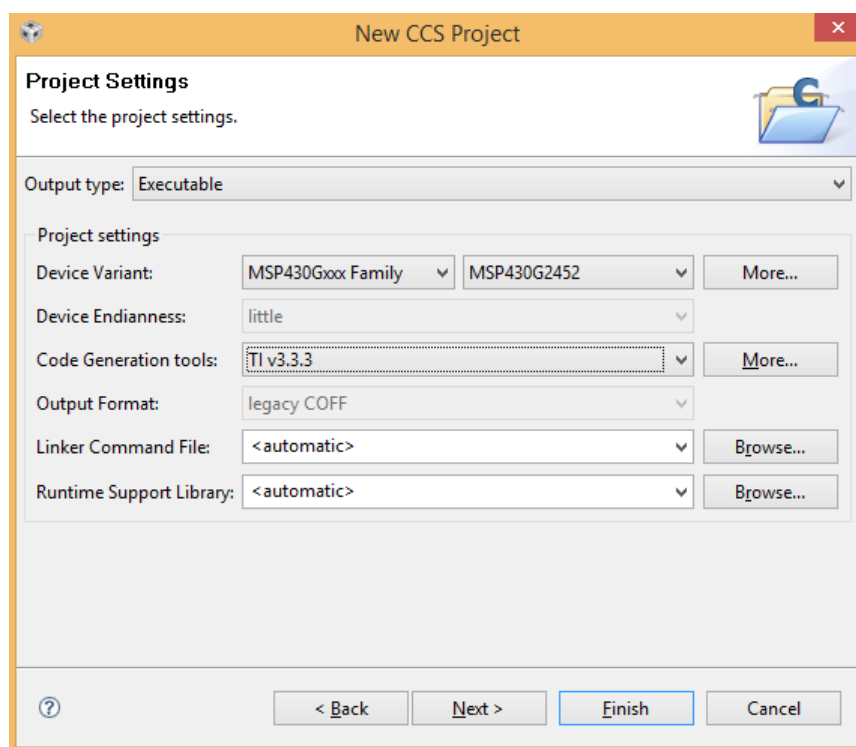


Рис. 3. Выбор целевого устройства

5. Создайте новый исходный файл: «File → New → Source File». Введите имя файла на английском языке и добавьте расширение *.c.
6. В окне редактирования исходного кода введите текст программы [1]:

```
#include <msp430g2452.h>
```

```

/*подключение файла, содержащего описание всех регистров и раз-
рядов, используемых в микроконтроллере G2452.*/
void main (void)
{
WDTCTL = WDTPW + WDTNHOLD;          //остановка сторожевого таймера
P1OUT = 0;                          // выключение светодиодов
P1DIR = 0x41;
        /* установка портов P1.0 и P1.6 в качестве выходных*/
while (1)
{
P1OUT = BIT0 + BIT6;                //включение светодиодов
__delay_cycles(100000);             //задержка на 100000 тактов
P1OUT = 0;                          //выключение светодиодов
__delay_cycles(100000);             //задержка на 100000 тактов
}
}

```

Данная программа позволяет управлять работой светодиодов – включая их (подавая на вход соответствующего порта «1») или выключая (при подаче «0»). Установленное значение «1» соответствует постоянной работе светодиода. Частота мигания светодиодов (чередование «0» и «1») устанавливается задержкой между выводами этих значений.





На отладочной плате MSP-EXP430G2 есть два встроенных светодиода, подключенных к портам P1.0 и P1.6. Им соответствуют стандартные биты BIT0 и BIT6. Команда:

```
P1OUT = BIT0 + BIT6;
```

позволяет включить данные светодиоды. Команда:

```
__delay_cycles(n);
```

выполняет задержку выполнения следующей команды на n тактов. Задержка задается на этапе написания программного кода и формируется во время его отладки.

7. Запустите отладчик: «Target → Debug Active Project» или нажмите кнопку . При обнаружении ошибок в программном коде их необходимо исправить. При их отсутствии отладчик автоматически произведет форматирование памяти микроконтроллера и запишет в нее новый исполняемый код.
8. Запустите исполняемую программу: «Target → Run» или во вкладке «Debug» нажмите кнопку . Убедитесь в работе отладочной платы. Встроенные светодиоды P1.0 и P1.6 должны мигать.
9. Приостановите выполнение исполняемой программы – нажмите: «Target → Halt» или кнопку .
10. Для остановки выполнения исполняемой программы нажмите: «Target → Terminate All» или кнопку .
11. Для задания режима поочередного включения светодиодов в программный код может быть интегрирован следующий фрагмент:

```
P1OUT = BIT0;
__delay_cycles(100000);
P1OUT = BIT6;
__delay_cycles(100000);
```

Внесите данные исправления в программный код и повторите этапы 7–10.

12. Для выхода из Code Composer Studio нажмите: «File → Exit».

1.3. Требования к отчету

Отчет по лабораторной работе оформляется согласно принятым требованиям и включает:

1. Титульный лист с указанием: названия университета и кафедры, номера и названия лабораторной работы, ФИО обучающихся, номера их учебной группы, ФИО и должности преподавателя. Внизу страницы указывается город и год.

2. Раздел «Выполнение лабораторной работы» содержит: цель работы, основные этапы создания проекта в интегрированной среде разработки Code Composer Studio с рисунками и комментариями. При технической возможности получения фотографий работающей отладочной платы, они также размещаются в отчете.

3. Выводы и ответы на контрольные вопросы.

4. Список литературы.

1.4. Контрольные вопросы

1. Назовите основную область применения микроконтроллеров.

2. Дайте определение симулятора.

3. Где применяются цифровые сигнальные процессоры?

4. Дайте определение интегрированной среды разработки.

5. Что называется эмулятором? Приведите основные типы эмуляторов.

6. Какая архитектура используется в микроконтроллерах семейства MSP430G2xxx?

7. Какие языки могут применяться для программирования микроконтроллеров и цифровых сигнальных процессоров?

8. Приведите основные этапы создания проекта в интегрированной среде разработки Code Composer Studio.

1.5. Список литературы

1. Официальный сайт компании Texas Instruments [Электронный ресурс]. URL: <http://www.ti.com> (дата обращения 01.09.2019).

2. Солонина, А. И. Алгоритмы и процессоры цифровой обработки сигналов / А. И. Солонина, Д. А. Улахович, Л. А. Яковлев. – СПб. : БХВ–Петербург, 2002. – 464 с.