

Лабораторная работа №1

Среда разработки Code Composer Studio. Плата MSP-EXP430F5529. Цифровой ввод-вывод

Цель: Ознакомиться с интегрированной средой разработки Code Composer Studio. Ознакомиться с основными функциональными возможностями платы MSP-EXP430F5529. Изучить приемы работы с цифровыми выводами.

Задача: Написать программу по управлению цифровым вводом-выводом в соответствии с заданием варианта.

1.1 Интегрированная среда разработки Code Composer Studio

Разработка проектов для лабораторного макета MSP-EXP430F5529 проводится в интегрированной среде разработки Code Composer Studio.

Подробную информацию и руководства можно найти на сайте компании-разработчика Texas Instruments: <http://www.ti.com/tool/ccstudio>. Здесь будут рассмотрены только основные приемы работы.

Чтобы открыть проект примера Code Composer Studio требуется в среде разработки в меню выбрать Project → CCS Example Projects (либо Project → Import Existing CCS Eclipse Project):

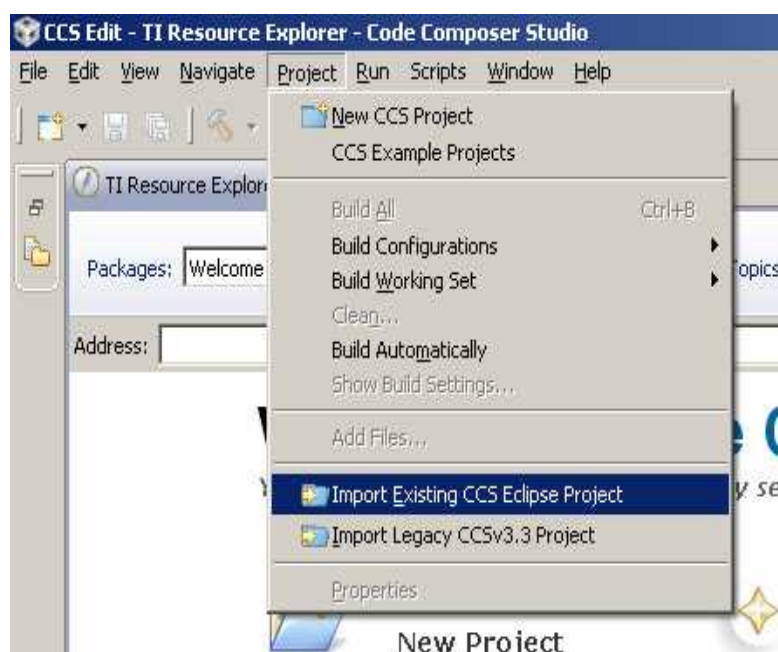


Рис. 1.1 Выбор проекта с примером

В открывшейся вкладке в дереве поиска необходимо выбрать серию устройств (MSP430ware → Devices → MSP430F55xx/6xx → Code Examples → MSP430FF552x), выбрать требуемый пример, и в выпавшем списке выбрать

устройство MSP430F5529:

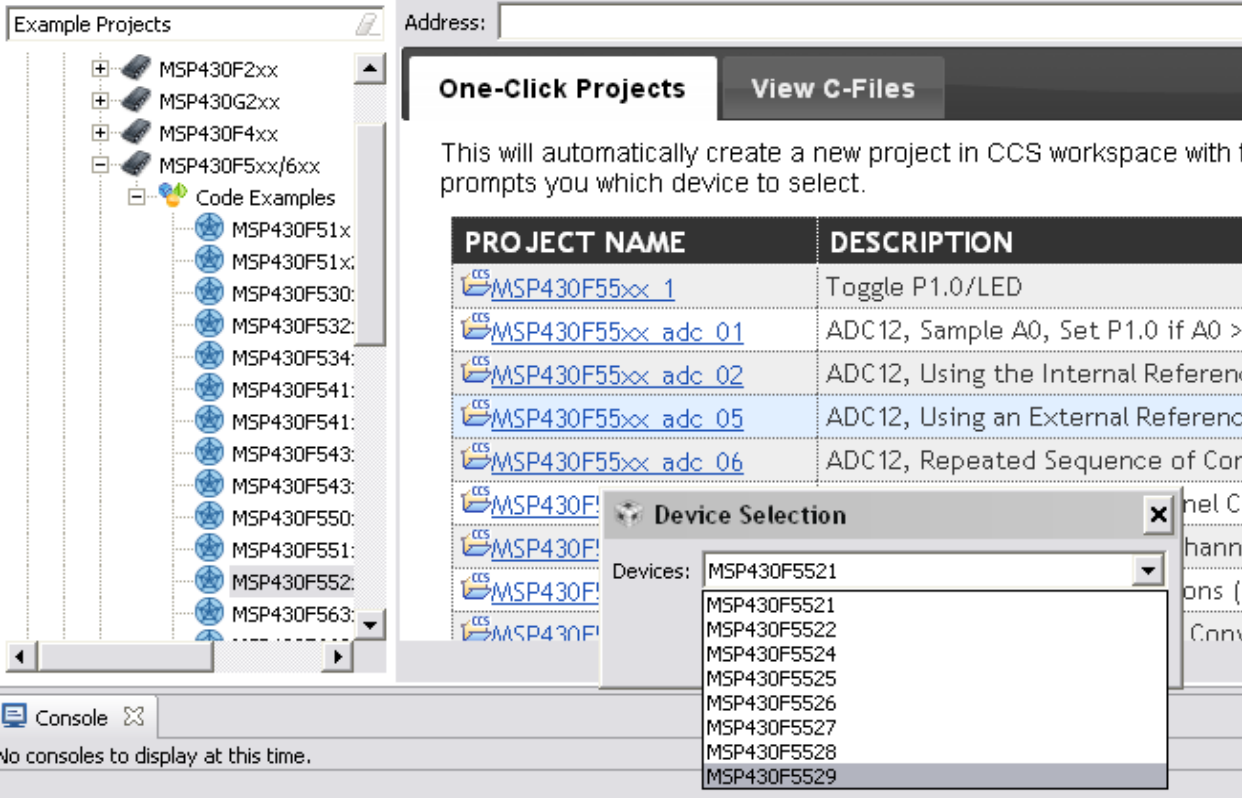


Рис. 1.2 Выбор примера и устройства

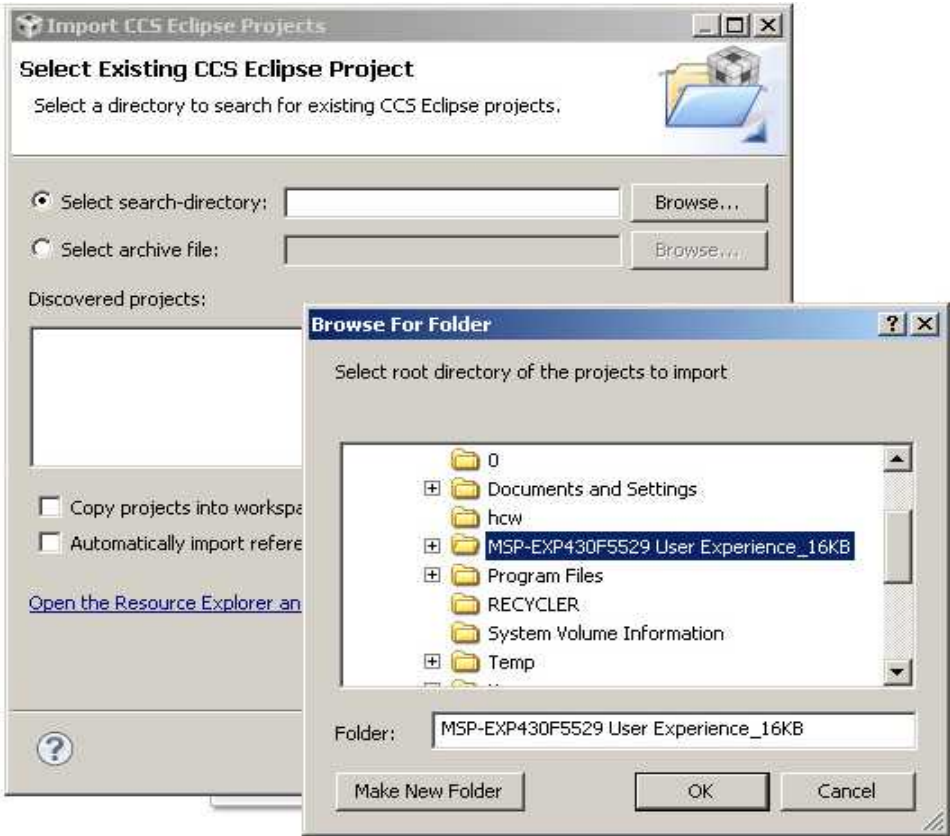


Рис. 1.3 Выбор Eclipse проекта

При выборе Eclipse проекта, в открывшемся окне следует нажать кнопку Browse... и выбрать папку, содержащую проект (MSP-EXP430F5529 User Experience_16KB_Cut), как показано на рис. 1.3.

После выбора проекта он отображается слева в поле Project Explorer. Двойной щелчок по нужному файлу открывает его в редакторе. В редакторе исходного кода при удержании клавиши Ctrl и при помощи щелчка мыши на вызове функции или использовании переменной можно перейти к ее объявлению или определению.

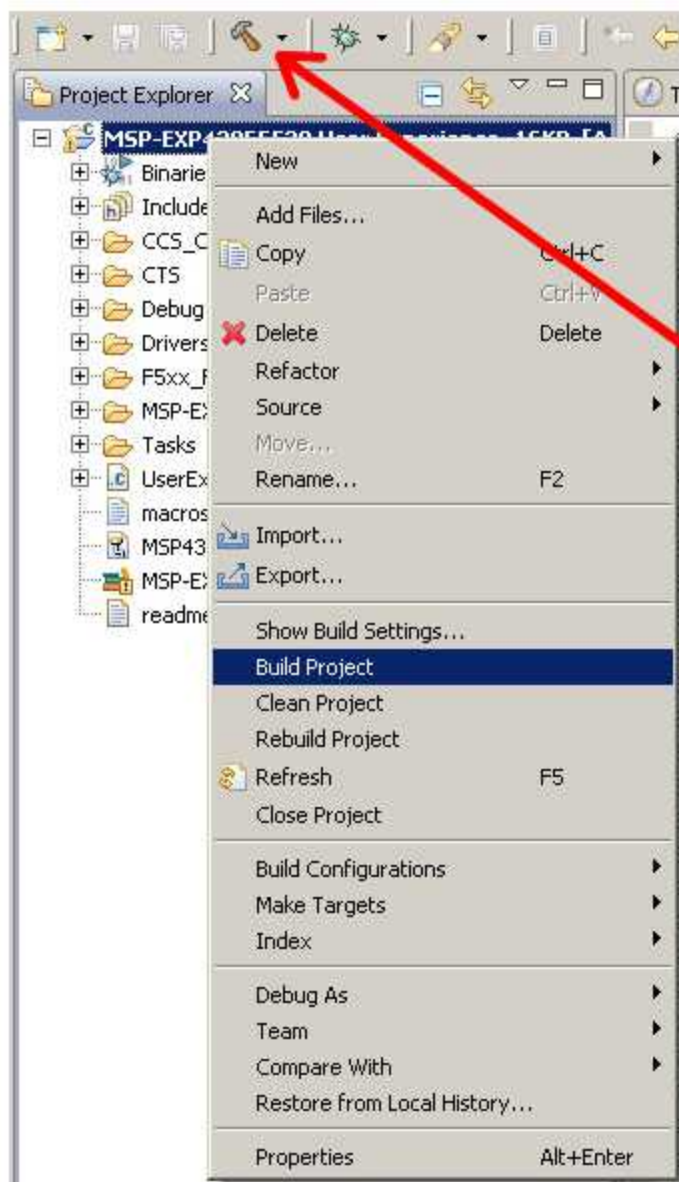


Рис. 1.4 Два варианта сборки проекта

Сборка проекта может быть запущена двумя способами:

- используя контекстное меню по щелчку правой кнопкой мыши на папке проекта в поле Project Explorer (рис. 1.4);

- щелчок мышью на панели инструментов по кнопке Build (на рис. 1.4 показан стрелкой).

Запись собранного проекта на контроллер и запуск на выполнение в режиме отладки выполняется вызовом из основного меню среды разработки Run → Debug или на панели инструментов кнопкой Debug (рис. 1.5).

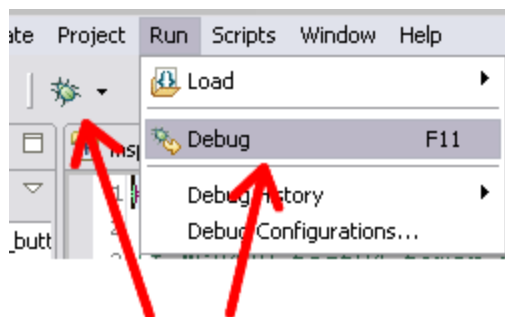


Рис. 1.5 Запуск загрузки программы в микроконтроллер

После прошивки и запуска CCS переходит в режим отладки, открывая соответствующее окно. Продолжить выполнение, войти внутрь вызываемой функции, выйти на уровень выше или остановить выполнение программы можно используя кнопки отладки на панели приложения. Точки останова ставятся в окне отладки слева от кода двойным щелчком. После остановки выполнения среда возвращается в режим редактирования.

1.2 Плата MSP-EXP430F5529

Подробную информацию и руководства можно найти на сайте компании-разработчика Texas Instruments: <http://www.ti.com/tool/msp-exp430f5529>. Здесь будут рассмотрены только основные возможности.

Экспериментальная плата MSP-EXP430F5529 разработана на основе микроконтроллера MSP430F5529 компании Texas Instruments. Это серия процессоров для обработки смешанных сигналов со сверхнизким энергопотреблением.

Основные особенности архитектуры:

- 16-разрядная ортогональная RISC архитектура;
- Фон-Неймановская адресная шина общей памяти и шина данных памяти;
- 27 (51) команд + 37 расширенных инструкций (20-бит адрес) + 11 адресных инструкций (20-бит операнды, но ограничения в режимах адресации);
- 7 согласованных способов адресации;
- полный программный доступ к регистрам, включая счетчик команд (PC), регистр состояния (SR), указатель стека (SP);
- одноктактные регистровые операции;

- большой размер регистрового файла, уменьшающий количество обращений к памяти;
- 20-битная шина адреса, 16-битная шина данных;
- генератор констант (6);
- пересылки память-память без промежуточного сохранения в регистре;
- гибкая система тактирования;
- несколько режимов пониженного энергопотребления;
- моментальный переход в активный режим (порядка 6 мкс).

Микроконтроллер обладает следующими характеристиками:

- производительность до 25 MIPS;
- напряжение питания 1,8-3,6 В;
- ток утечки вывода 50 нА;
- потребление в режиме хранения данных 0,1 мкА;
- потребление в режиме часов реального времени 2,5 мкА.

Микроконтроллер включает в свой состав:

- флеш-память 128 Кб, SRAM 8 Кб;
- 80 выводов, 63 линии входа/выхода;
- 4 асинхронных 16-разрядных таймера/счетчика (7,5,3,3 регистров захвата соответственно);
- сторожевой таймер (WDT) и таймер часов реального времени (RTC);
- модуль управления питанием PMM с блоками защиты от падений напряжения (BOR) и контроля напряжения питания (SVS);
- универсальный последовательный коммуникационный интерфейс USCI 2 x UART/LIN/IrDA/SPI + 2 x I2C/SPI;
- 3 канала DMA;
- умножитель-накопитель MPY 32 x 32 бита;
- компаратор;
- 12 разрядный АЦП (ADC 12A), 16 каналов;
- полноскоростной USB 2.0 (12Мб/с), до 8 линий в/в со встроенным 3,3 В стабилизатором (питание от 5 В шины, обеспечивает ток 12 мА);
- интерфейс для измерения линейных и угловых перемещений (SIF);
- LCD контроллер до 128 сегментов;
- внутренний генератор частоты с цифровым управлением.

Обобщенная архитектура микроконтроллера представлена на рис. 1.6. Элементы архитектуры микроконтроллера будут описаны по мере выполнения лабораторных работ. Более подробную информацию можно найти на <http://www.ti.com/product/msp430f5529>. Внешний вид экспериментальной платы представлен на рис. 1.7, а назначение основных элементов - на рис. 1.8.

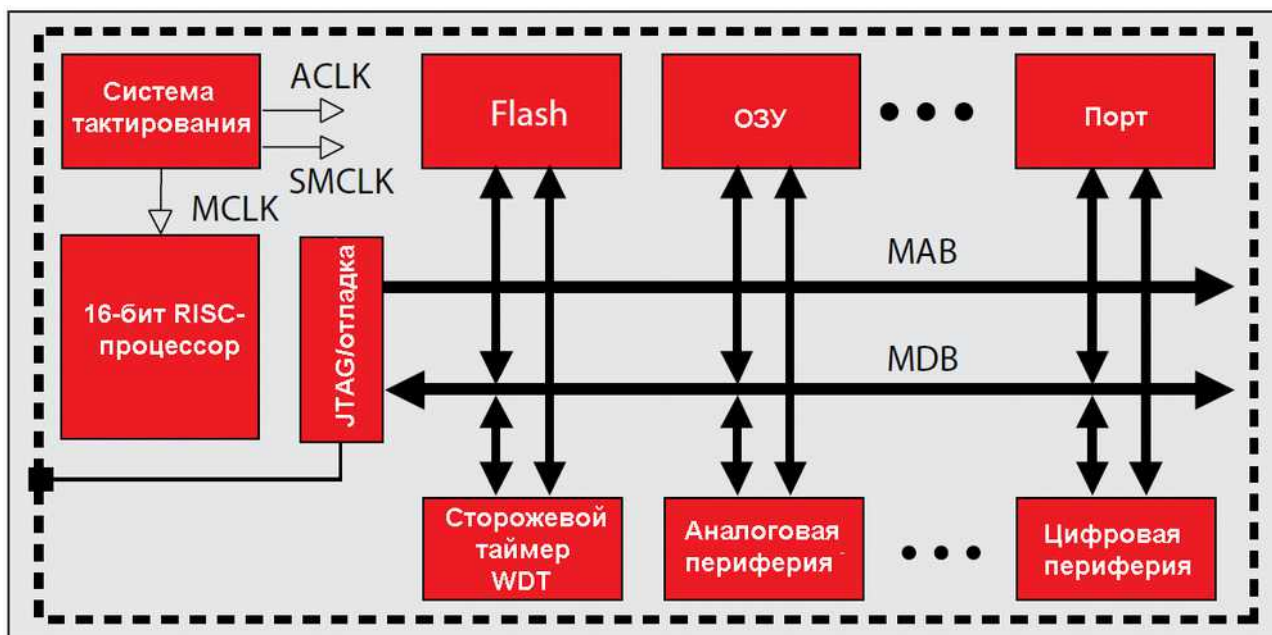


Рис. 1.6 Архитектура микроконтроллера MSP430

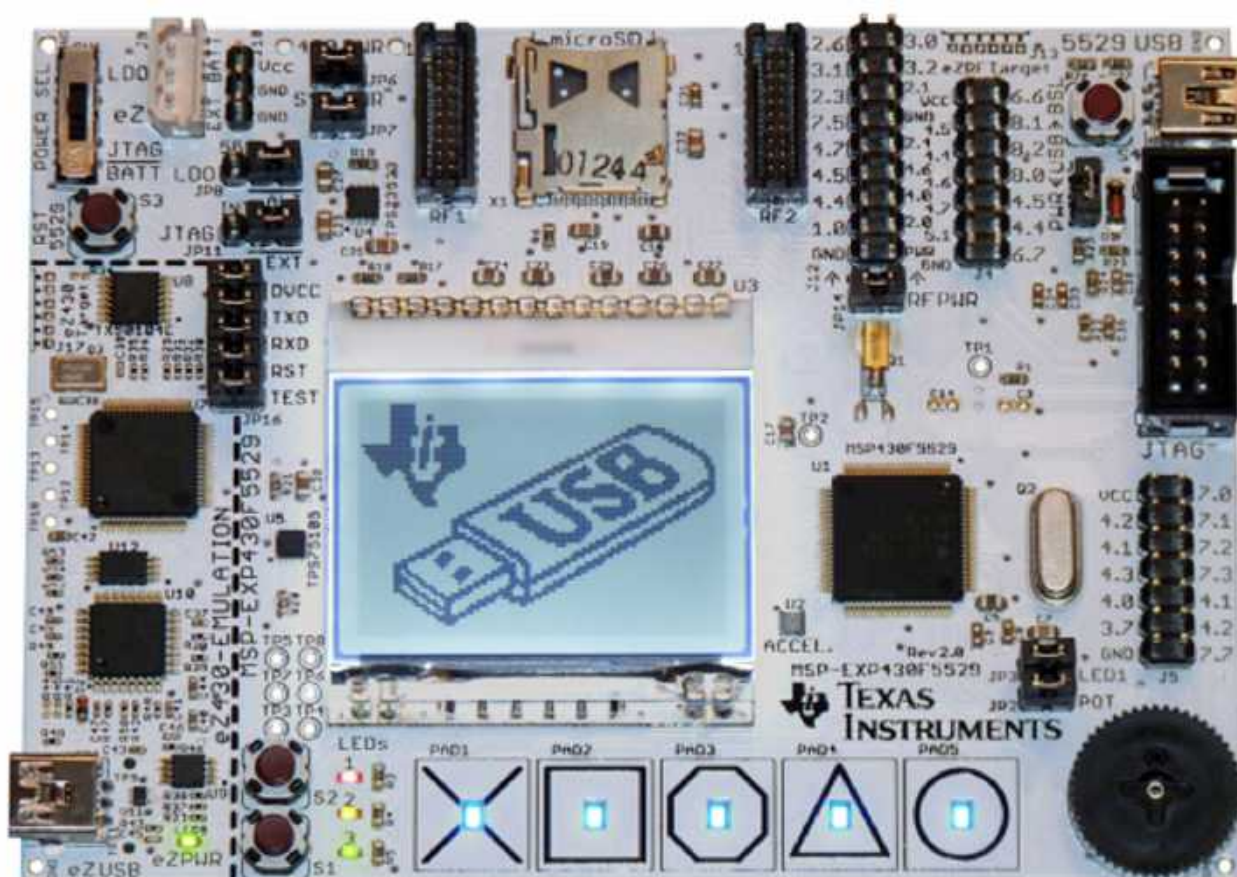


Рис. 1.7 Внешний вид экспериментальной платы MSP-EXP430F5529

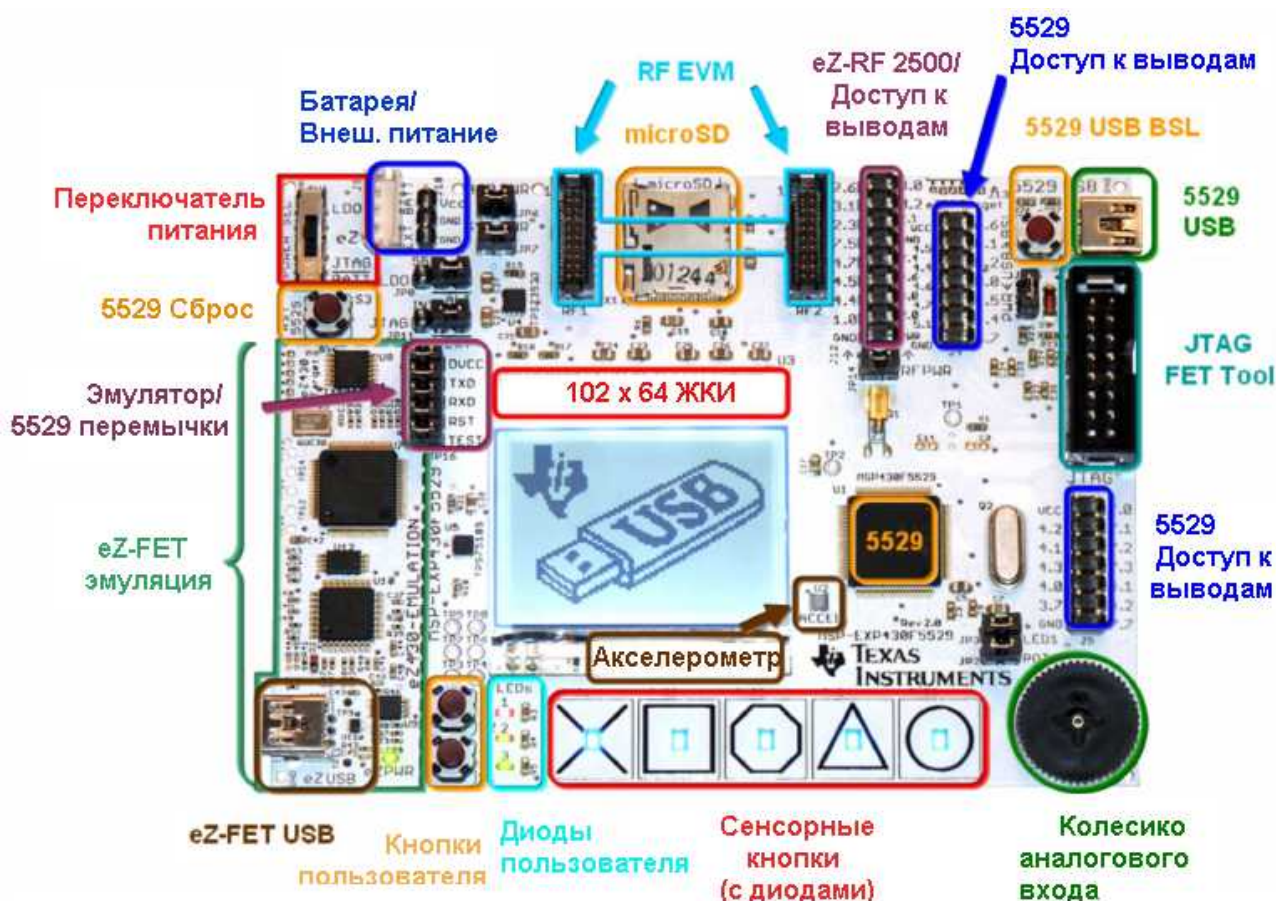


Рис. 1.8 Назначение элементов экспериментальной платы MSP-EXP430F5529

Плата MSP-EXP430F5529 подключается к USB-порту ПК через разъем ezUSB платы. При исследовании возможностей экспериментальной платы для управления меню будут использоваться пользовательские кнопки и колесико.

1.3 Цифровой ввод-вывод

8-разрядные порты P1, P2, P3, ..., P8, PJ управляют выводами контроллера. Выводы программируются либо как I/O, либо как вход/выход периферии. Порты могут объединяться в пары: P1 и P2 = PA, P3 и P4 = PB, P5 и P6 = PC, P7 и P8 = PD. При работе с прерываниями порты в пары не объединяются. Для порта могут быть доступны регистры:

- PxIN – чтение данных с вывода;
- PxOUT – установка значения выхода;
- PxDIR – выбор направления: 0 – вход, 1 – выход;
- PxREN – разрешение подтягивающего резистора;
- PxDS – выбор допустимой силы вывода;
- PxSEL – выбор функции вывода: 0 – I/O, 1 – периферия;
- PxIV – генерирует значение для изменения счетчика команд, соответствующее прерыванию с максимальным приоритетом;

PxIES – выбор направления перепада для генерации запроса на прерывание: 0 – по фронту, 1 – по спаду;

PxIE – разрешение прерывания;

PxIFG – флаг прерывания.

Адреса соответствующих портов представлены в таблице:

Таблица 1.1 — Адреса портов ввода-вывода

№ порта	1	2	3	4	5	6	7	8	J
База	200h		0220h		0240h		0260h		0320h
PxIN		1	0	1	0	1	0	1	0
PxOUT	2	3	2	3	2	3	2	3	2
PxDIR	4	5	4	5	4	5	4	5	4
PxREN	6	7	6	7	6	7	6	7	6
PxDS	8	9	8	9	8	9	8	9	8
PxSEL	A	B	A	B	A	B	A	B	-
PxIV	E	1E	-	-	-	-	-	-	-
PxIES	18	19	-	-	-	-	-	-	-
PxIE	1A	1B	-	-	-	-	-	-	-
PxIFG	1C	1D	-	-	-	-	-	-	-

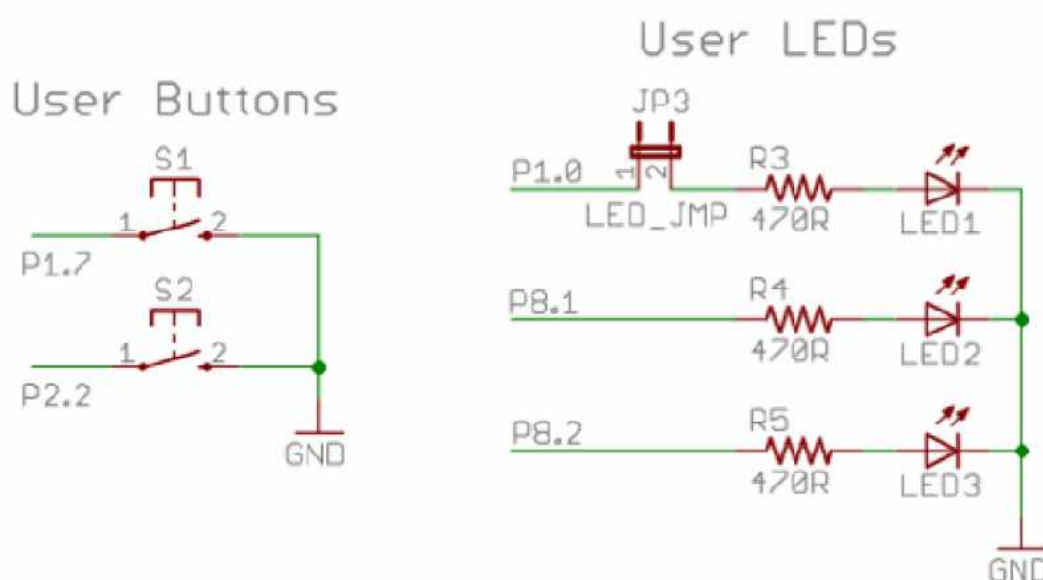


Рис. 1.9 Подключение пользовательских кнопок и светодиодов

Пользователю программно доступны две кнопки S1 и S2, подключенные соответственно к выводу 7 порта 1 и выводу 2 порта 2 (см. рис. 1.9). В

дальнейшем такое подключение будем обозначать как P1.7 и P2.2 соответственно. Также программно доступны 8 светодиодов, три из которых (LED1 – LED3, см. рис. 1.9) размещены рядом с кнопками и подключены соответственно к выводам P1.0, P8.1, P8.2. Еще 5 светодиодов (LED4 – LED8) размещаются в блоке сенсорных кнопок и подключены к выводам P1.1 – P1.5 соответственно.

Логика управления выводом на примере порта 1 представлена на рисунке ниже. Для других портов схемотехника может несколько отличаться, в зависимости от особенностей подключаемой к выводу периферии микроконтроллера.

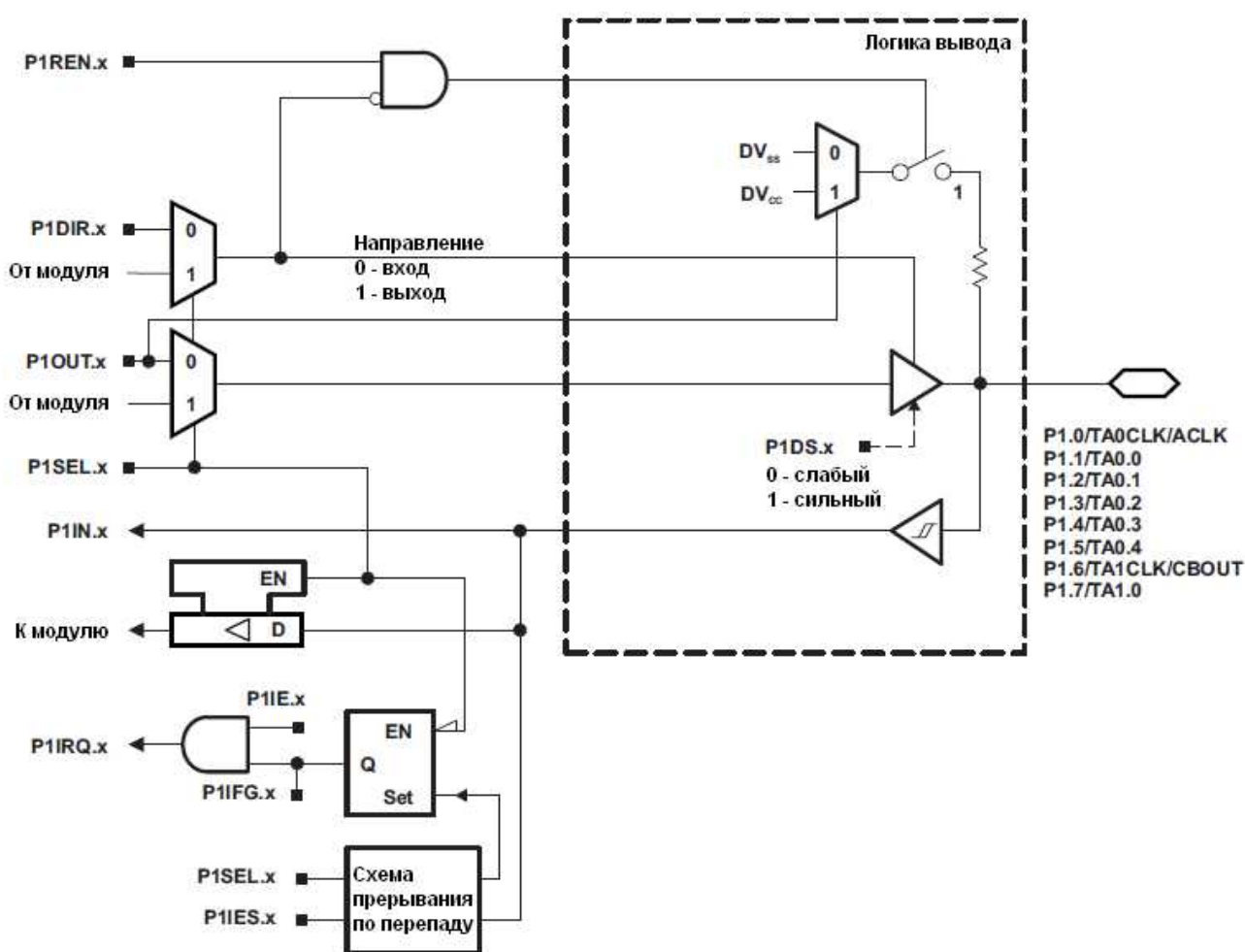


Рис. 1.10 Организация входа-выхода с триггером Шмидта на примере порта 1

Следует обратить внимание, что регистр RxOUT управляет подключением подтягивающего резистора, если вывод сконфигурирован как цифрой I/O, направление — выход, и разрешен подтягивающий резистор. В случае, если вывод сконфигурирован как вывод периферии микроконтроллера, прерывания не генерируются. Отметим также, что после сброса цифровые выходы конфигурируются на вход, кроме того запускается сторожевой таймер в сторожевом режиме.

При написании кода следует учесть несколько моментов. Вначале следует подключить заголовочный файл `msp430.h`, который в свою очередь подключает файл `msp430f5529.h`, содержащий необходимые константы в соответствии с архитектурой контроллера. Далее, поскольку после сброса запускается сторожевой таймер, его следует отключить (иначе через какое-то время сработает сброс).

Константы и определения заданы как для портов, так и для отдельных полей и их значений. Поэтому работа с портами становится максимально удобной для программиста. Так, например, запись `P8DIR |= BIT2`; означает, что в порт `P1DIR`, отвечающий за выбор направления выводов порта 1, заносится новое значение, которое получено логическим ИЛИ его текущего состояния и бита 2. Фактически, это устанавливает бит 2 в заданном порту.

Следует обратить внимание, что при наименовании констант использовались следующие принципы:

- константа, соответствующая биту поля-флага именуется по имени поля, например, полю `CPUOFF` регистра состояния процессора `SR` (бит 4) соответствует константа `CPUOFF`;
- константа соответствующая биту `n` в поле `NNN` именуется `NNNn`;
- константа, соответствующая номеру `x` выбранного варианта для поля `NNN` именуется `NNN_x`;
- константа, соответствующая выбранному режиму `zz` для поля `NNN` именуется `NNN__zz`.

Так, например, для 3-битного поля `SELA`, константа, соответствующая 0 биту поля, именована `SELA0`, вариант выбора 0 (`SELA = 000`) именован `SELA_0`, а режим, соответствующий данному варианту именован `SELA__XT1CLK`. В некоторых случаях поля задают делители либо множители, соответствующие степени двойки. Тут надо быть особо внимательным и не спутать похожие мнемоники, например, `NN4` (четвертый бит, т.е. 10000), `NN_4` (четвертый вариант, т.е. 00100), `NN__4` (режим деления на 4, т.е. 00011).

Задание

1. Загрузить Code Composer Studio. Изучить интерфейс пользователя и основные возможности. В соответствии с вариантом (табл. 1.2) загрузить и скомпилировать демонстрационный пример, загрузить его в микроконтроллер.

2. Загрузить прошивку `MSP-EXP430F5529 User Experience`. Изучить возможности экспериментальной платы.

3. В соответствии с вариантом (табл. 1.3), не используя прерываний и таймеров, запрограммировать кнопки и светодиоды. Комментарии в тексте программы обязательны, они должны пояснять что именно делает данный фрагмент. Не допускается использовать иные заголовочные файлы, кроме `msp430`, не допускается также использовать высокоуровневые библиотеки.

Табл. 1.2. Наименования демонстрационных примеров к п.1 задания

№ варианта	Наименование демонстрационного примера
1	MSP430F55xx_adc_01
2	MSP430F55xx_adc_02
3	MSP430F55xx_adc_05
4	MSP430F55xx_adc_06
5	MSP430F55xx_adc_07
6	MSP430F55xx_compB_01
7	MSP430F55xx_compB_02
8	MSP430F55xx_compB_03
9	MSP430F55xx_compB_04
10	MSP430F55xx_compB_05
11	MSP430F55xx_ta1_01
12	MSP430F55xx_ta1_02
13	MSP430F55xx_ta1_03
14	MSP430F55xx_ta1_04
15	MSP430F55xx_ta1_05

Табл. 1.3. Варианты к п.3 задания

№ варианта	Светодиоды и кнопки	Режим	Переключение
1	LED1, S1, S2	Непрерывный	Включение при отпускании одной кнопки, выключение при нажатии другой
2	LED2, S1, S2	Мигание	Включение и выключение режима при нажатии любой из кнопок
3	LED3, S1	Импульсный	Короткий импульс при нажатии и при отпускании кнопки
4	LED1, LED2, S2	Непрерывный	Когда один включен — второй выключен. Переключение светодиодов по нажатию кнопки
5	LED1, LED3, S1, S2	Импульсный	Короткий импульс на одном диоде при нажатии любой из кнопок, на другом — при отпускании
6	LED2, LED3, S2	Мигание	Когда один включен — второй выключен. Включение и выключение режима при нажатии кнопки
7	LED1 – LED3, S1	Зажигание	При нажатии на кнопку диоды включаются друг за другом с небольшой задержкой, при повторном нажатии — одновременно гаснут
8	LED1 – LED3, S2	Потухание	При нажатии на кнопку все диоды включаются, при отпускании - гаснут друг за другом с небольшой задержкой
9	LED4 – LED8, S1	Перемещение	Включение и выключение режима нажатием кнопки. Диод на некоторое время включается, потом гаснет, и включается следующий.
10	LED4 – LED8, S2	Движение	Включение и выключение режима нажатием кнопки. Диоды включаются друг за другом с некоторой задержкой, потом гаснут друг за другом с некоторой задержкой.
11	LED4 – LED8, S1, S2	Движение	При нажатии на одну кнопку, диоды включаются друг за другом с некоторой задержкой. При нажатии на вторую - гаснут друг за другом с некоторой задержкой.
12	LED4 – LED8, S1	Движение	Режим включен, только когда кнопка нажата. Диоды включаются друг за другом с некоторой задержкой, потом гаснут друг за другом с некоторой задержкой.
13	LED1 – LED3, S1	Зажигание	При нажатии на кнопку включается очередной диод. Если все горят — то при нажатии на кнопку они гаснут.
14	LED1 – LED3, S2	Потухание	При нажатии на кнопку диоды включаются, при каждом следующем нажатии гаснет один диод.
15	LED4 – LED8, S1	Перемещение	И при нажатии, и при отпускании кнопки на некоторое время загорается очередной диод

Литература

1. Гук И. Краткий обзор отладочной среды Code Composer Studio 4. – Компоненты и технологии, №12, 2009. – с. 90-96
2. Code Composer Studio. Getting Started Guide. – Texas Instruments, Lit. Num. SPRU509C, 2001. – 131 p.
3. Code Composer Studio v5.4. User's Guide for MSP430. – Texas Instruments, Lit. Num. SLAU157Y, 2005. – Rev. 2013. – 52 p.
4. Code Composer Studio v6.0 for MSP430. User's Guide. – Texas Instruments, Lit. Num. SLAU157AE, 2005. – Rev. 2014. – 69 p.
5. Grace for Code Composer Studio IDE. Getting Started Guide. – Texas Instruments, Lit. Num. SLAU476, 2012. – 17 p.
6. MSP430 Optimizing C/C++ Compiler v4.1. User's Guide. - Texas Instruments, Lit. Num. SLAU 132G, 2012. - 155 p.
7. MSP430 Assembly Language Tools v4.1. User's Guide. - Texas Instruments, Lit. Num. SLAU131G, 2012. - 284 p.
8. S. Schauer. Mixing C and Assembler With the MSP430. - Texas Instruments, Lit. Num. SLAA140, 2002. - 21 p.
9. K. Quiring. MSP430 Software Coding Techniques. - Texas Instruments, Lit. Num. SLAA294A, 2006. - 7 p.
10. MSP-EXP430F5529 Experimenter Board. User Guide. - Texas Instruments, Lit. Num. SLAU330A, 2011. - 36 p.
11. MSP430x5xx and MSP430x6xx Family. User's Guide.- Texas Instrument, Lit. Num. SLAU208O, 2008. - Rev. 2015. - 1191 p.
12. Гук И. Краткий обзор микроконтроллеров семейства MSP430 компании Texas Instruments. - Компоненты и технологии, №6, 2006.
13. Семейство микроконтроллеров MSP430x2xx. Архитектура. Программирование. Разработка приложений. - М.: ИД «Додэка-XXI», 2010. - 544 с.
14. Семейство микроконтроллеров MSP430x4xx. Руководство пользователя. - М.: ЗАО «Компэл», 2005. - 416 с.
15. Семейство микроконтроллеров MSP430. Рекомендации по применению. - М.: ЗАО «Компэл», 2005. - 544 с.
16. MSP430F5529, MSP430F5528, MSP430F5527, MSP430F5526, MSP430F5525, MSP430F5524, MSP430F5522, MSP430F5521, MSP430F5519, MSP430F5517, MSP430F5515, MSP430F5514, MSP430F5513. Mixed Signal Microcontroller. - Texas Instruments, Lit. Num. SLAS590L, 2009. - Rev. 2013. - 122 p.
17. MSP430F5529 Device Erratasheet. - Texas Instruments, Lit. Num. SLAZ314I, 2012. - Rev. 2013. - 35 p.
18. Digital I/O Module. Chapter Excerpt from SLAU 208. - Texas Instruments, Lit. Num. SLAU396B, 2011. - Rev. 2013. - 20 p.