

посты [q&a](#) [блоги](#) [события](#) [компании](#) .тостер

индекс

Программирование микроконтроллеров* 182,08

[Захабренные](#)[Новые \(1\)](#)[Отхабренные](#)

MSP430, учимся программировать и отлаживать железо



Сегодня, уважаемый хабрапользователь, я постараюсь заполнить некоторый пробел, образовавшийся в статьях об MSP430, а именно азы и подход к программированию устройств на данном микроконтроллере.

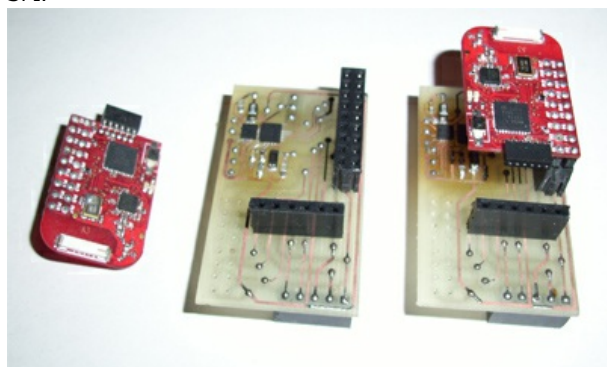
Эта статья прежде всего направлена на новичков, поскольку я буду рассматривать ряд достаточно простых задач, таких как работа с SPI, мигание

лампочкой и отладка в proteus.

Введение

В данной статье будет рассмотрено устройство, в основу которого легла отладочная плата [eZ430-RF2500](#). На плате находится микроконтроллер MSP430F2274 и беспроводной модуль CC2500, который, надо заметить, не будет рассмотрен далее.

Моим коллегой, Соколовым С. А., была изготовлена небольшая надстройка для этого отладочного комплекта, она присоединена ко всем выводам. На надстройке расположен акселерометр LIS331DLH, с которым мы и будем взаимодействовать по SPI.



Должен заметить, что устройства STMicroelectronics, работающие по SPI, очень похожи и, соответственно, работа с ними будет выглядеть примерно также.

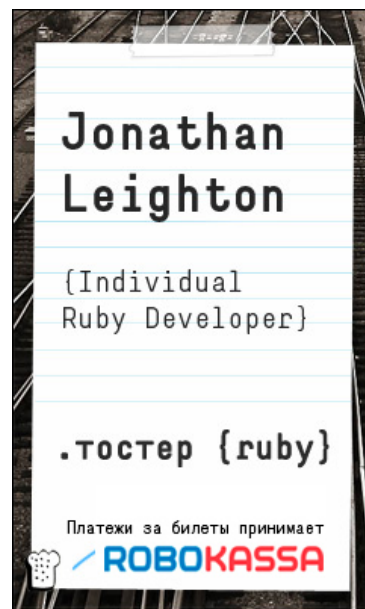
Что нам потребуется

Среда разработки

Для начала нужно скачать и установить среду разработки и компилятор. На сегодняшний день существует три варианта — [Code Composer Studio](#), [IAR Embedded Workbench for TI MSP430](#) и [mspgcc](#).

Я буду использовать Workbench KickStart Edition. [KickStart](#) бесплатный, он имеет

Итого

13036 читателей
67 постов

Лич

адап



.тостер

[Каскадные Таблицы](#)[Стилей](#) → [Bootstrap 2.0](#)— [HTML+CSS+JS UI фреймворк](#)[Open source](#) → [Игры id Software](#) — на [Github](#)[PHP](#) → [Деобфускация PHP кода](#)[Dura Lex](#) → [Как избежать уголовной ответственности за распространение нелицензионного ПО и что делать, если вас поймали](#)[Виртуализация](#) → [Проброс видеокарты в виртуальную машину](#)[jQuery](#) → [Facebook Timeline](#)[Nginx](#) → [MySQL в NGINX: использование блокирующих библиотек в неблокирующем сервере](#)[Электронное правительство](#) → [Платформа gov.uk будет полностью открытой. Исходные коды опубликованы на GitHub](#)

ограничение по количеству кода, но для изучения этого более чем достаточно.

Средство отладки

Если у вас нет под рукой осциллографа или логического анализатора, то часто возникают сложности, связанные с непониманием того, что же на самом деле происходит в вашем устройстве. Понять причины того, почему же устройство отказывается работать часто помогает Proteus.

В нём можно найти очень многие микроконтроллеры MSP430. К сожалению MSP430F2274 в Proteus не оказалось, но имеется аналог — MSP430F2272, его и будем использовать.

Приступим к написанию кода

Создание проекта

На данном этапе сложностей возникнуть не должно. В качестве языка программирования выбираем C++.

- Заменяем стандартный `#include <io430.h>` на `#include «msp430f2274.h»` (заголовочный файл для нашего микроконтроллера);
- В свойствах проекта на вкладке Debugger выбираем Driver: FET Debugger;
- На вкладке General Options в окне Device выбираем MSP430F2274;
- Для того чтобы убедиться, что всё сделано правильно нажимаем Ctrl+D (Download and Debug). Как только контроллер прошивка загрузится нажимаем F5 для её выполнения.

В случае если нужно скомпилировать файл для Proteus, на вкладке Debugger в окне Driver выбираем Simulator, а на вкладке Linker Output в окне Format ставим Other и Output format выбираем intel-standart, а в окне Output file меняем расширение на hex.

Работа с портами

Первое, что стоит освоить в микроконтроллере — это работа с портами. Давайте рассмотрим небольшой пример.

```
#include "msp430f2274.h"

void main( void )
{
    WDTCTL = WDTPW + WDTOLD;

    P1DIR &= ~BIT2;
    P1REN |= BIT2;

    P1DIR |= BIT1 + BIT0;

    while(true)
    {
        if(P1IN & BIT2)
        {
            P1OUT |= BIT1;
            P1OUT &= ~BIT0;
        }
        else
        {
            P1OUT |= BIT0;
            P1OUT &= ~BIT1;
        }
    }
}
```

PxDIR отвечает за направление порта 1. Когда конкретный бит данного регистра установлен в 0, соответствующий пин работает на вход. И наоборот, если

[Разработка под Android → Существующие приложения и будущие Android x86 устройства](#)

[Информационная безопасность → В Украине продолжаются атаки на сайты органов власти](#)

[« все лучшие »](#)



12.01.2012 → Программирование микроконтроллеров → [Digital Metawatch WDS112 оригинальный отладочный комплект от Texas Instruments](#)

21.02.2011 → Программирование микроконтроллеров → [Необычный отладочный комплект от Texas Instruments](#)

03.01.2012 → Программирование микроконтроллеров → [Комплект разработчика на базе MSP430 от Texas Instruments](#)

08.02.2011 → Железо → [Texas Instruments готовит систему-на-кристалле OMAP 5 с поддержкой 3d](#)

14.02.2007 → Гаджеты. Устройства для гиков → [Texas Instruments демонстрирует новый мобильный процессор OMAP3430](#)

02.02.2012 → Программирование микроконтроллеров → [MSP430, учимся программировать и отлаживать железо \(часть 2\)](#)

11.09.2007 → Железо → [Процессор за \\$10 от Texas Instruments для мобильных устройств](#)

11.01.2007 → Персональные блоги → [Texas Instruments сделает мобильник ценой \\$20](#)

24.01.2012 → DIY или Сделай Сам → [Обзор ЛанчПада MSP430 и тестовый проект](#)

18.01.2012 → JAVA → [Выпуск №72 — The Art Of Programming \[Drinking \] Переезд в TEXAS](#)



[mihaidd](#) → [Как избежать уголовной ответственности за распространение нелегального ПО и что делать, если вас поймали 145](#)

[konsolytyper](#) → [Swing. Десять полезных простых вещей 52](#)

[zilia](#) → [Проброс видеокарты в виртуальную машину 59](#)

[Ajex](#) → [В Украине продолжаются атаки на сайты органов власти 53](#)

[b3atb0x](#) → [To-do лист Леонардо да Винчи 45](#)

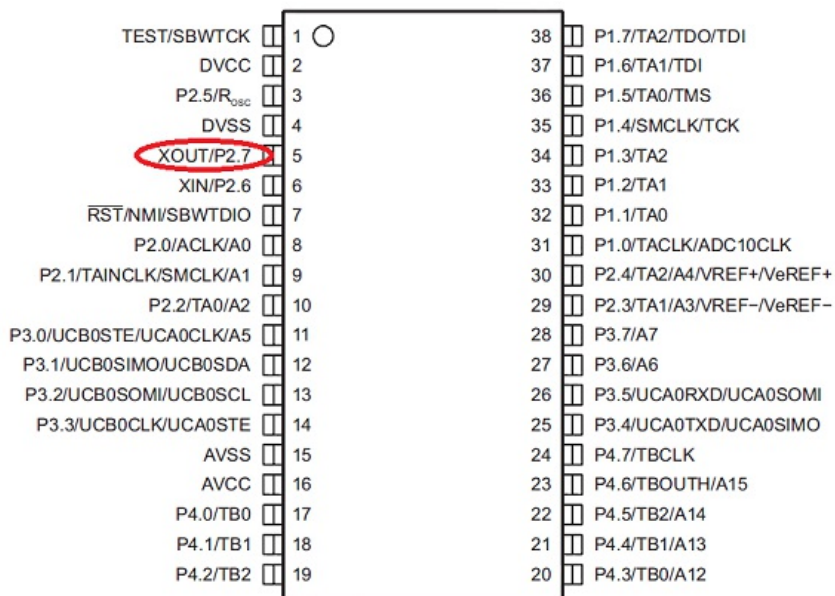
[9mm](#) → [Google запустила сайт для разработчиков: JavaScript vs. Dart 243](#)

соответствующий бит установлена в 1, то пин работает на выход. В примере фигурируют 3 пина: P1.2 — кнопка, P1.0 — красный светодиод, P1.1 — зеленый светодиод.

PxREN включает внутренний резистор подтяжки. Кнопка замыкает пин на землю, и, соответственно, переводит его в состояние нуля. Когда кнопка не нажата пин ни к чему не подключен и для обеспечения логической единицы на нём требуется подключить его через резистор к питанию, что и делает регистр P1REN.

PxIN и **PxOUT** содержат в себе состояние пинов порта. Устанавливая ноль или единицу в регистр PxOUT мы меняем напряжение на лапке микроконтроллера, тем самым включая и выключая светодиод. Читая конкретный бит из регистра PxIN мы получаем логический сигнал, который сейчас подан на пин.

PxSEL выбирает функцию пина. В datasheet на изображении микроконтроллера функции обычно указывают через знак «/».



Например на рисунке P2.7 работает как обычный пин в случае, если P2SEL имеет 0 в соответствующем разряде. По умолчанию, в данном случае, там установлена единица, что означает, что эта лапка предназначена для подключения внешнего часового кварцевого резонатора.

Константы **BIT0..BITF** содержатся в файле msp430f2274.h и представляют собой 16-ти разрядные слова в заданном разряде которых содержится 1, все остальные разряды содержат 0.

Надо заметить, что файл msp430f2274.h содержит много полезной информации. Там находятся все константы контроллера с комментариями на английском.

В примере используются побитовые операции Си, «|=» установит соответствующий значению справа бит в регистре слева в единицу, а «&= ~» напротив установит его в 0.

Что получилось:

[xeLL](#) → Когда нужен RVM, а когда достаточно rbeny 27

[Iamer](#) → Firefox: сочетания клавиш в стиле Emacs, а также загрузка содержимого clipboard по нажатию средней кнопки мыши 6

[openkazan](#) → Бюджетный роутер Buffalo WCR-HP-GN 10

[Incole](#) → Обновление рейтинга лучших приложений для Windows Phone 7 35



Уникальный сервис «МультиФон» позволяет совершать и принимать звонки по привлекательным тарифам!

[Попробовать мультифон](#)

Совершай и принимай **видеовызовы**, общайся в чате и пользуйся другими возможностями.



[Indigo](#) → В какой стране должен стоять сервер с полулегальным контентом, чтобы его не беспокоили? 8

[hudson](#) → Как заставить разработчиков сдать протестированный продукт? 4

[Evengard](#) → Расширения для Chrome Portable 2

[Indigo](#) → Посоветуйте JSON библиотеку под Java 1

[ili](#) → Как вызвать "синий экран" на локальном компьютере? 8

[Indigo](#) → Наушники apple и обычные капельки 1

[SLIDERWEB](#) → Что случилось с Rutracker.org? 5

[ili](#) → Как сделать сенсорный терминал (интернет-киоск) 2

Работа с SPI

Сразу начну с примера

```
#include "msp430f2274.h"

unsigned char spi(unsigned char data, unsigned char dataEx = 0x00);

void main( void )
{
    WDTCTL = WDTPW + WDT HOLD;

    P1DIR |= BIT0 + BIT1;
    P1OUT &= ~BIT0;
    P1OUT &= ~BIT1;

    P3SEL = BIT1 + BIT2 + BIT3;
    P3DIR |= BIT0;
    P3OUT |= BIT0; // Отключаем CC2500 (устанавливаем 1 на CS)

    P2SEL &= ~BIT6;
    P2SEL &= ~BIT7;
    P2DIR |= BIT6 + BIT7;
    P2OUT |= BIT6; // Отключаем датчик температуры (тоже подключен к SPI)
    P2OUT |= BIT7; // Отключаем акселерометр

    // Конфигурируем SPI
    UCB0CTL0 |= UCMSB + UCMST + UCSYNC;
    UCB0CTL1 |= UCSSEL_2;
    UCB0BR0 = 0x02;
    UCB0BR1 = 0;
    UCB0CTL1 &= ~UCSWRST;

    if(spi(0x8F) == 0x32)
    {
        P1OUT |= BIT1; // Красный светодиод
    }
    P1OUT |= BIT0; // Зеленый светодиод
}

unsigned char spi(unsigned char data, unsigned char dataEx)
{
    unsigned char RX;

    P2OUT &= ~BIT7; // Включаем акселерометр

    while (!(IFG2 & UCB0TXIFG)); // Ожидаем готовность буфера отправки
    UCB0TXBUF = data;
    while (!(IFG2 & UCB0RXIFG)); // Ожидаем готовность буфера приёма
```

[Zloy1](#) → [Git windows+linux Russian filename problem](#) 5

[asm0dev](#) → [Какой торрент клиент выбрать?](#) 3

[« все вопросы](#)



[Python-разработчик](#)

[Presale менеджер](#)

[Python разработчик](#)

[UI-дизайнер](#)

[Разработчик PHP ASP .NET MVC](#)

[Программист ASP.NET](#)

[iOS-разработчик 75-го уровня](#)

[Lead game developer](#)

[Junior PHP-программист \(мало опыта, стажер\)](#)

[Разработчик Flex](#)

[« все вакансии](#)



Алгоритмы

Ненормальное программирование

Регулярные выражения

Git

Веб-разработка

Sphinx

Системное программирование

Разработка

Game Development

Совершенный код

[« все блоги](#)



03 февраля [Лекция Санберга: интерактив в телевидении](#)

04 февраля [Мастер-класс по разработке приложений для Android](#)

06 февраля [QML тренинг в Петрозаводске](#)

07 февраля [Начальный курс по разработке сайтов на TPL-шаблонизаторе](#)

07 февраля [Лекция Джона Перри Барлоу, который против цензуры в Интернете](#)

[« все события](#)

```

RX = UCB0RXBUF;

while (!(IFG2 & UCB0TXIFG));
UCB0TXBUF = dataEx;
while (!(IFG2 & UCB0RXIFG));
RX = UCB0RXBUF;

P2OUT |= BIT7;

return RX;
}

```

В примере запрашивается значение регистра по адресу 0x8F, там содержится код, который идентифицирует устройство. Этот код указан в datasheet. Это позволяет убедиться в том, что обмен данными произошёл. В случае успеха включаем красный светодиод.

Соответственно все остальные устройства подключенные к SPI необходимо отключить от интерфейса. Для этого CS на них устанавливается в единицу.

Заключение

В следующий раз постараюсь рассказать подробнее про работу с LIS331DLH, добраться до прерываний, поработать со встроенным в программатор мостом USB-UART и рассказать немного про watchdog.

Я надеюсь, что эта статья оказалась полезна тебе, читатель.

[msp430](#), [texas instruments](#), [proteus](#), [workbench](#)

↑ +33 ↓ 29 января 2012, 02:47 ☆ 109 [B](#) [f](#) [kirill89](#)

💬 комментарии (9)

 **Linol** 29 января 2012, 03:37 # 0  

Хотелось бы более подробно узнать про работу с SPI.

 **kirill89** 29 января 2012, 03:40 # ↑ 0  

Обязательно расскажу подробнее на следующей неделе.

 **ShouldNotSeeMe** 29 января 2012, 17:40 # 0  

ИМХО, MSP430 ждёт участь AVR, т.к. за ту же цену есть более мощный и менее прожорливый на той же частоте ARM. А вот работать с периферией на MSP430 проще и удобнее, спору нет.

 **Derailed** 29 января 2012, 20:06 # ↑ +1  

Участь AVR? Очень хорошая участь, знаете ли...

 **ShouldNotSeeMe** 29 января 2012, 20:46 # ↑ 0  

В более-менее серийных изделиях их теперь ставить нецелесообразно.

 **Derailed** 30 января 2012, 00:46 # ↑ +1  

Популярность платформы зависит не только от её цены. Важны в совокупности, кроме цены — мощность, количество примеров кода, доступность чипов, наличие хорошей документации и апноутов.

 **kirill89** 30 января 2012, 00:55 # ↑ 0  

Полностью согласен.
К этому стоит добавить наличие доступных отладочных наборов.

 **ibnteo** 30 января 2012, 23:02 # ↑ 0  

Ну уж доступнее MSP430 LaunchPad за символические \$4.30 с доставкой на дом ничего не найти.



Klaus 29 января 2012, 17:49 <#>

+1

Эх, у меня есть Chronos ez430. думал буду его постоянно мучать-перепрошивать, да очень привык к часам и боюсь остаться без них :)

Пишите еще, очень полезно и толково.

Только зарегистрированные пользователи могут оставлять комментарии.
[Войдите](#), пожалуйста.

[Войти](#)
[Регистрация](#)

Разделы
[Q&A](#)
[События](#)
[Работа](#)
[Блоги](#)
[Компании](#)
[Люди](#)
[Лучшие](#)

Блоги
[Все](#)
[Тематические](#)
[Корпоративные](#)
[Песочница](#)

Инфо
[О сайте](#)
[Правила](#)
[Помощь](#)
[Соглашение](#)
[Статистика](#)

Услуги
[Реклама](#)
[Корпоративные пакеты](#)
[Семинары](#)



© 2006–2012
«Тематические Медиа»

Служба поддержки:
support@habrahabr.ru

[Мобильная версия](#)