[посты](#) [q&a](#) [блоги](#) [события](#) [компании](#)**Прокачаем**  
ваш Интернет в 2 раза

индекс

# Программирование микроконтроллеров \* 180,74

[Захабренные](#)[Новые](#)[Отхабренные](#)**Индикатор**13397 читателей  
71 пост

## MSP430, учимся программировать и отлаживать железо (часть 3)



Возможно Вы уже не первый раз садитесь за программирование Вашего MSP430. Если это не так, то рекомендую ознакомиться с предыдущими статьями по данной тематике: [часть 1](#), [часть 2](#). В этой статье мы рассмотрим инструмент для графической настройки периферии нашего микроконтроллера — Grace, познакомимся с принципом работы watchdog и поработаем с

виртуальным COM-портом (через программатор). Уже традиционно будет рассмотрен небольшой пример кода, и предоставлены все необходимые для понимания ссылки.

### Введение

В первой статье я упоминал [Code Composer Studio](#), от рассмотрения которого отказался, но недавно моё внимание, благодаря Соколову А.В., привлёк один плагин для неё — [Graphical Peripheral Configuration Tool \(Grace\)](#).

Каждый раз, когда приходится настраивать очередной периферийный модуль MSP430, мы сталкиваемся с рядом сложностей. К таким сложностям можно отнести незнание списка регистров конкретного модуля, отличительные особенности конкретного микроконтроллера или плохое понимание англоязычных документов. В любом из перечисленных случаев Grace может стать полезным.

### Grace

Скачать и установить этот инструмент можно вместе с Code Composer Studio v5 по следующей [ссылке](#). Для этого потребуются зарегистрироваться на сайте Texas Instruments.

После установки CCS выбираем *Project* → *New CCS Project*, в области *Project templates and examples* выбираем *Empty Project* → *Empty Grace (MSP430) Project*, в области *Device* не забудьте указать микроконтроллер из списка.

Как только появится экран «Grace — Welcome», нажимаем клавишу *Device Overview*. Перед Вами появится следующая картинка.

Время переходить  
на **мегаплан**

Четверть лучших веб-разработчиков  
доверяет нам совместную работу  
и управление проектами.

Совместная  
работа



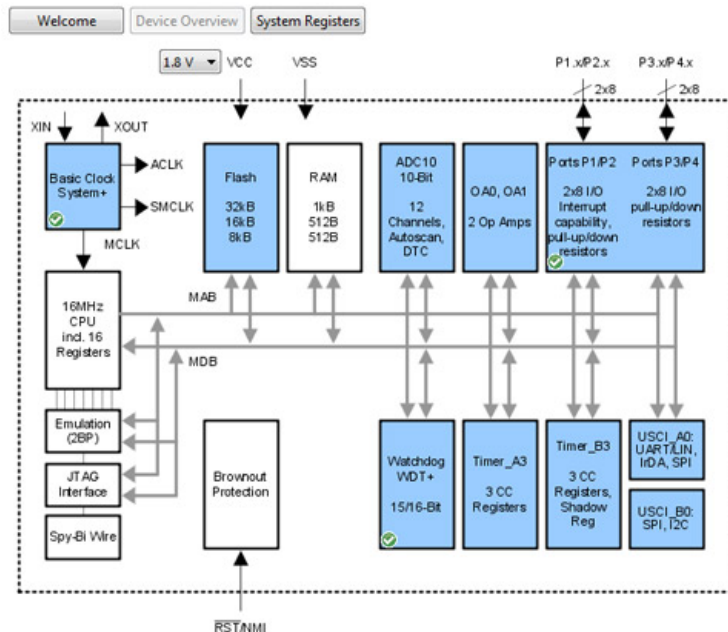
CRM:  
Клиенты  
и продажи

**Лун**

аан 2 а

**.тостер** [Информационная  
безопасность](#) →[Trustwave признает выдачу корневого  
сертификата сторонней компании](#)[Arduino](#) → [Arduino LCD Informer](#)[Веб-разработка](#) → [Изучить jQuery за 30  
дней](#)[Хостинг](#) → [+2Гб для вашего DropBox  
аккаунта. Промо-акция Samsung Бразилия](#)[Веб-стандарты](#) → [WebKit и веб-стандарты](#)[Копирайт](#) → [Американский суд решит,  
можно ли продавать МРЗ-файлы б/у](#)[C++](#) → [Thunk: без ассемблера и](#)

## Grace - MSP430F2274



Кликнув по конкретному устройству на ней, Вы перейдете к его настройке, которую можно производить в нескольких режимах. Режимы могут отличаться для разных устройств, но обычно это *Basic User*, *Power User* и *Registers*. *Basic User* и *Power User* предназначены для быстрой настройки периферийного устройства интуитивно. *Registers* отображает список всех регистров устройства и позволяет менять их значения.

*Совет:* в *datasheet* к микроконтроллеру не всегда содержится полная информация об интересующем устройстве, про все регистры и параметры обычно можно прочитать в файле *MSP430xxxx Family User's Guide*, который можно скачать на сайте [Texas Instruments](http://www.ti.com).

Поскольку для работы я уже привык использовать *Workbench*, после завершения настройки, все конфигурационные данные требуется перенести туда. Для этого нажимаем *Project* → *Build All*, находим в папке проекта директорию */src/csl/*, в ней и находятся все \*.c файлы с конфигурацией каждого устройства.

### Пример кода

Следующий код работает с USB-UART мостом встроенным в программатор, так же в код включено использование *watchdog*, но обо всём по порядку.

```
1.
2. #include "msp430f2274.h"
3. #include <string>
4.
5. void UARTWriteString(string str);
6.
7. bool watchdogReset = true;
8.
9. void main(void)
10. {
11.     WDTCTL = WDT_MRST_32; // Watchdog автоматически
12.                          // перезапустит систему через 32ms.
13.
14.     BCSCTL1 = CALBC1_1MHZ; // Устанавливаем тактовую частоту Basic Clock Sy
15.     DCOCTL = CALDCO_1MHZ; // Устанавливаем тактовую
16.                          // частоту Digital Controlled Oscillator.
17.
18.     P3SEL = BIT4 + BIT5; // Выбираем функцию P1.4/P1.5 как TXD/RXD для UART.
19.
20.     UCA0CTL1 |= UCSWRST; // Этот бит блокирует работу прерываний от UART и
21.
22.                          // сдвигового регистра чтобы не мешать
23.                          // настройке (грубо говоря отключает UART).
24.     UCA0CTL1 |= UCSSEL_2; // Наш UART будет работать от
```

машинного кода

Сетевые технологии → [Тог выпустил обфусцированный прокси для маскировки трафика в Иране](#)

Разработка под Apple iOS → [Настройка внешнего вида UIPopoverController](#)

Opera → [Opera 12 \(Wahoo\)](#), обновление ядра с исправлениями функции Do Not Track, почты и темизации

« [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 \(часть 2\)](#)

29.01.2012 → [Программирование микроконтроллеров](#) → [MSP430, учимся программировать и отлаживать железо](#)

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

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

10.02.2012 → [.NET](#) → [Реализация Code Action с помощью Roslyn](#)



[catstail1954](#) → [HomeLisp](#) два года спустя 70

[youROCK](#) → [Нарушения эвристики Нильсена в Mac OS 3](#)

[alemiks](#) → [Как закрыть сайт, наплевать на соцсети, запретить копираст и стать ведущей медиа-компанией 11](#)

[Ualde](#) → [Немного истории: кто такой Anonymous 7](#)

[mejedi](#) → [Thunk: без ассемблера и машинного кода 11](#)

```

24. // SMCLK (Sub-main clock), то есть от 1MHz.
25. UCA0BR0 = 0x68; // Делитель частоты для SMCLK (1000000 / 9600).
26. UCA0BR1 = 0x00;
27. UCA0MCTL = 0x04; // Определяет маску модуляции.
28. // Это помогает минимизировать ошибки.
29. UCA0CTL1 &= ~UCSWRST; // Включаем UART обратно.
30.
31. P1DIR |= BIT0; // Настройка светодиодов.
32. P1DIR |= BIT1;
33. P1OUT &= ~BIT0;
34. P1OUT &= ~BIT1;
35.
36. UARTWriteString("---Привет, Хабр!---");
37.
38. unsigned char data;
39. while(true)
40. {
41.     while (!(IFG2&UCA0RXIFG)) // Проверка готовности буфера приёма.
42.         if(watchdogReset)
43.             WDTCTL = WDTPW + WDTCTL; // Сброс таймера watchdog в ноль.
44.
45.     data = UCA0RXBUF;
46.     if(data == 0x01)
47.     {
48.         UARTWriteString("---Погасить зеленый светодиод.---");
49.         P1OUT &= ~BIT1;
50.     }
51.     else if(data == 0x02)
52.     {
53.         UARTWriteString("---Зажечь зеленый светодиод.---");
54.         P1OUT |= BIT1;
55.     }
56.     else if(data == 0x03)
57.     {
58.         UARTWriteString("---Переключить состояние красного светодиода.---");
59.         P1OUT ^= BIT0;
60.     }
61.     else
62.     {
63.         UARTWriteString("---Принятые данные не соответствуют.---");
64.         watchdogReset = false;
65.     }
66. }
67.
68. void UARTWriteString(string str)
69. {
70.     int strSize = str.length();
71.     for(int i = 0; i < strSize; i++)
72.     {
73.         WDTCTL = WDTPW + WDTCTL; // Сброс таймера watchdog в ноль.
74.         while (!(IFG2&UCA0TXIFG)); // Проверка готовности буфера отправки.
75.         UCA0TXBUF = str[i];
76.     }
77. }
78.

```

Для работы с COM-портом со стороны компьютера нам потребуется программное обеспечение, мой выбор пал на [COM Port Toolkit](#). Что именно выберите Вы — не имеет значения.

Небольшое видео, которое позволит понять что именно делает пример.

[iOrange](#) → +2Гб для вашего DropBox аккаунта. Промо-акция Samsung Бразилия 84

[utrack](#) → В комнате с белым потолком 127

[Shpuntik](#) → Изучить jQuery за 30 дней 32

[disinvis](#) → Opera 12 (Wahoo), обновление ядра с исправлениями функции Do Not Track, почты и темизации 31

[toTheSky](#) → Общаемся с SIM-картой на низком уровне 24



Хочешь бесплатный и удобный инструмент для разработки приложений под Windows Phone?

Скачать и начать

Участуй в конкурсе на лучшее приложение и получай призы.



Яндекс

Последний пост: [Поисковые запросы в холода](#)

4584 поклонника



[sainnr](#) → Точка в email 2

[etc](#) → Напоминка о перерывах для mac 2

[trilodi](#) → javascript и упаковка изображения в base64 3

[Norraxx](#) → Брейн штормы 1

[Genome\\_X](#) → Сколько стоит ждать ответа при сообщении о наличии уязвимости на сайте 6

[barker](#) → В какой город переехать? 36

[eucariot](#) → Мнение о работе в Huawei 3

[Sytrus](#) → Каким должен быть военный сайт? 2

[ArturSitnikoff](#) → Куда пойти лечить голову? 9

В отличие от предыдущих статей, я постарался дать внятные комментарии прямо в коде программы. Не вижу смысла в этот раз пояснять каждый использованный в коде регистр. Хочу лишь обратить внимание на некоторые аспекты.






1. Для расчёта значений **UCA0BR0**, **UCA0BR1** и **UCA0MCTL** существует неплохой [онлайн калькулятор](#).
2. **SMCLK** — сигнал, который поступает из внешнего резонатора (если установлен) или из DCO с применением делителей 1, 2, 4 или 8. Используется как тактовый сигнал для периферии.
3. В примере, watchdog используется не совсем по назначению, данный код лишь объясняет принцип его работы. Смысл заключается в том, что если Ваша программа в течении 32ms (**WDT\_MRST\_32**) не установит бит **WDTCNTCL** в регистре **WDTCTL** в единицу, то система будет перезагружена. Watchdog требуется для предотвращения зависания Вашего программного обеспечения.
4. Регистр **WDTCTL** имеет 16 бит, первые 8 необходимо устанавливать в **WDTPW** каждый раз когда производится запись в него. Это механизм защиты регистра от случайной записи в случае программных сбоев.
5. Описание и настройка Basic Clock System это повод для целой статьи, пока следует понять, что MSP430 имеет очень гибкую систему тактовых генераторов, которую, в упрощенном виде, можно настроить с помощью Grace.
6. Использование такого метода отладки сильно замедляет работу программы в целом, однако это позволяет достоверно определить порядок выполнения кода, в том числе в обработчиках прерываний.

## Заключение


Чем дольше я пишу, тем сложнее и больше становится материал. Эта статья станет предпоследней в серии для новичков. В следующий раз я затрону все вопросы, на которые не успел ответить ранее.

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

 [msp430](#), [texas instruments](#), [watchdog](#), [uart](#), [Code Composer Studio](#)

 +15  7 февраля 2012, 00:57  75     [kirill89](#)

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

 alexxxst → Падение производительности SSD с TrueCrypt 3

« [все вопросы](#)



[Программист PHP / JS / MySQL](#)

[Ведущий программист PHP](#)

[Программист PHP](#)

[Программист со знанием 1С-Битрикс](#)

[Разработка сайта на Django](#)

[Системный администратор \(можно удаленно\) крупного веб-сервиса](#)

[Архитектор Cognos BI](#)

[PHP-разработчик на высокие нагрузки и интересные задачи](#)

[Graphics Hardware Engineer](#)

[Research Intern \(DSP\)](#)

« [все вакансии](#)



[Алгоритмы](#)

[Ненормальное программирование](#)

[Регулярные выражения](#)

[Git](#)

[Веб-разработка](#)

[Sphinx](#)

[Системное программирование](#)

[Разработка](#)

[Game Development](#)

[Совершенный код](#)

« [все блоги](#)



№ 13 февраля [Business Intelligence. Бизнес анализ от 1С:БИТ и QlikView Business Discovery](#)

№ 13 февраля [О некоторых аспектах бизнес-анализа на базе платформы QlikView Business Discovery](#)

№ 13 февраля [Auto ConfeT&QA: онлайн-конференция для специалистов по автоматизации тестирования](#)

№ 13 февраля [Лекция Скотта Козна про то, как стать знаменитым без спама](#)

№ 14 февраля [Круглый стол «Технологии объединенных коммуникаций Microsoft»](#)

« [все события](#)

---

[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](mailto:support@habrahabr.ru)

[Мобильная версия](#)