

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Нижегородский государственный университет им. Н.И. Лобачевского
Национальный исследовательский университет

ПЕРВЫЕ ШАГИ В ПРОГРАММИРОВАНИИ
МИКРОКОНТРОЛЛЕРА СЕРИИ MSP430

Практикум

Рекомендовано методической комиссией радиофизического факультета
для студентов ННГУ, обучающихся по направлениям подготовки
03.03.03 «Радиофизика», 02.03.02 «Фундаментальная информатика и
информационные технологии» и специальности 10.05.02
«Информационная безопасность телекоммуникационных систем»

Нижний Новгород
2015

УДК 681.3

ББК 32.973.2

ПЕРВЫЕ ШАГИ В ПРОГРАММИРОВАНИИ МИКРОКОНТРОЛЛЕРА СЕРИИ MSP430 / Составители: Шкелёв Е.И., Иванов А.В., Калинин В.А., Пархачёв В.В. – Практикум. – Нижний Новгород: Нижегородский госуниверситет, 2015. – 15с.

Рецензент: доцент кафедры общей физики, к.ф.-м.н. С.Н. Жуков

Методические указания содержат экскурс в среду разработки программного обеспечения IAR Embedded Workbench, представлены примеры программ, управляющих различным периферийным оборудованием контроллера MSP430, даны необходимые пояснения. Приводятся задания для самостоятельного выполнения. Имеется перечень контрольных вопросов.

Методические указания к лабораторной работе разработаны для преподавания на радиофизическом факультете дисциплины «Аппаратные средства вычислительной техники» по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии», дисциплины «Микропроцессорные системы» по направлению подготовки 03.03.03 «Радиофизика» и дисциплины «Аппаратные средства телекоммуникационных систем» по специальности 10.05.02 «Информационная безопасность телекоммуникационных систем».

Ответственные за выпуск:

председатель методической комиссии радиофизического факультета ННГУ,
к.ф.-м.н., доцент **Н.Д. Миловский**,
зам. председателя методической комиссии радиофизического факультета
ННГУ, д.ф.-м.н., профессор **Е.З. Грибова**

УДК 681.3
ББК 32.973.2

© Нижегородский государственный
университет им. Н.И. Лобачевского, 2015

1. Введение

Цель лабораторной работы состоит в изучении структуры, архитектуры и программирования микроконтроллеров (МК) фирмы Texas Instruments серии MSP430 [1-6], а также в освоении способов тестирования и средств отладки разрабатываемого программного обеспечения.

Лабораторная работа является продолжением работы "Знакомство с микроконтроллером серии MSP430", которая предполагается уже выполненной.

2. Описание установки

Для выполнения данной лабораторной работы используется отладочная плата OLIMEX MSP430-P2618 [7] с установленным на ней микроконтроллером MSP430F2618 (рис. 1) и программатор.

ВНИМАНИЕ! При работе с микроконтроллером во избежание поломок необходимо соблюдать осторожность — запрещается трогать руками штырьковые контакты отладочной платы, поскольку контроллер может быть повреждён статическим электричеством.

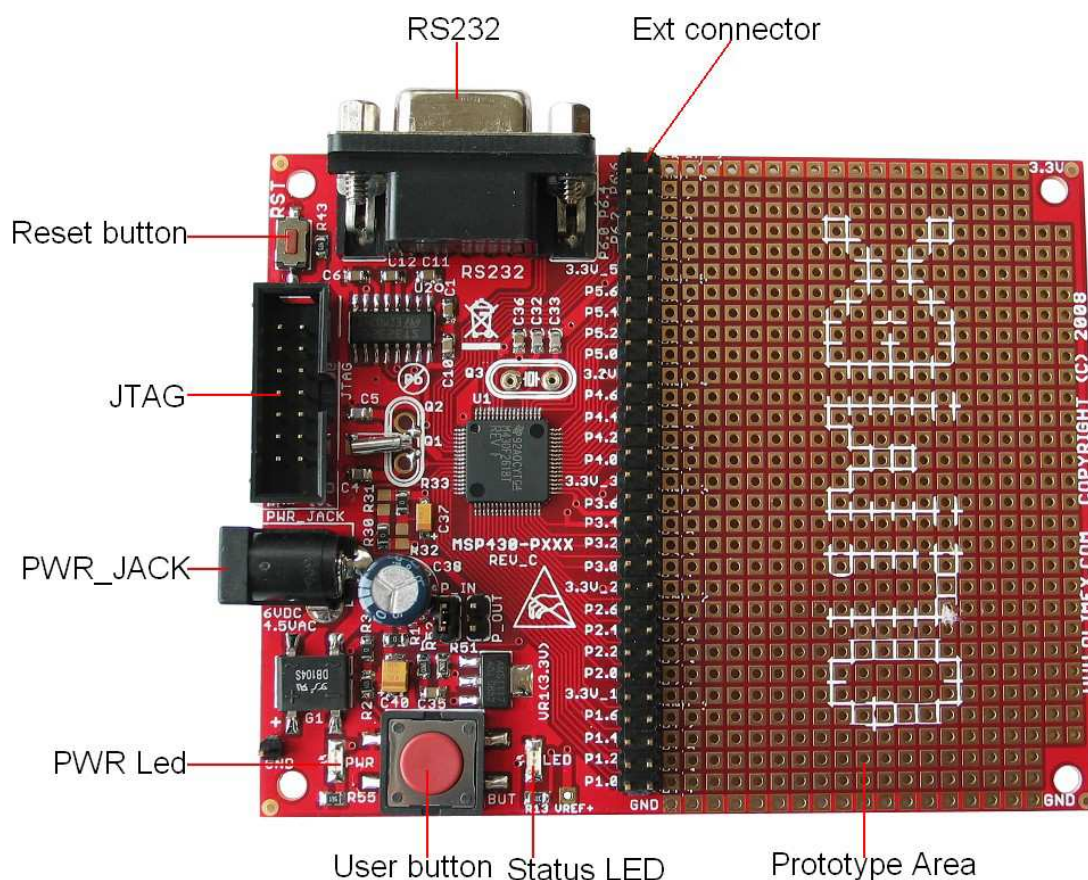


Рис. 1. Отладочная плата

На плате имеется 60-штырьковых разъёмов (pin connectors). Обратите внимание на подписи около каждого из коннекторов. Для удобства эти подписи продублированы на рис. 2. Легко видеть, что кроме портов P1 – P6 на разъём несколько раз выведены контакты для напряжения питания (VCC = +3.3 В) и "земля" (GND).

Pin #	Signal Name	Pin #	Signal Name
1	VCC	2	GND
3	P6.6	4	P6.7
5	P6.4	6	P6.5
7	P6.2	8	P6.3
9	P6.0	10	P6.1
11	VCC	12	GND
13	P5.6	14	P5.7
15	P5.4	16	P5.5
17	P5.2	18	P5.3
19	P5.0	20	P5.1
21	VCC	22	GND
23	P4.6	24	P4.7
25	P4.4	26	P4.5
27	P4.2	28	P4.3
29	P4.0	30	P4.1
31	VCC	32	GND
33	P3.6	34	P3.7
35	P3.4/TXD0	36	P3.5/RXD0
37	P3.2	38	P3.3
39	P3.0	40	P3.1
41	VCC	42	GND
43	P2.6	44	P2.7
45	P2.4	46	P2.5
47	P2.2	48	P2.3
49	P2.0	50	P2.1
51	VCC	52	GND
53	P1.6	54	P1.7
55	P1.4	56	P1.5
57	P1.2	58	P1.3
59	P1.0	60	P1.1

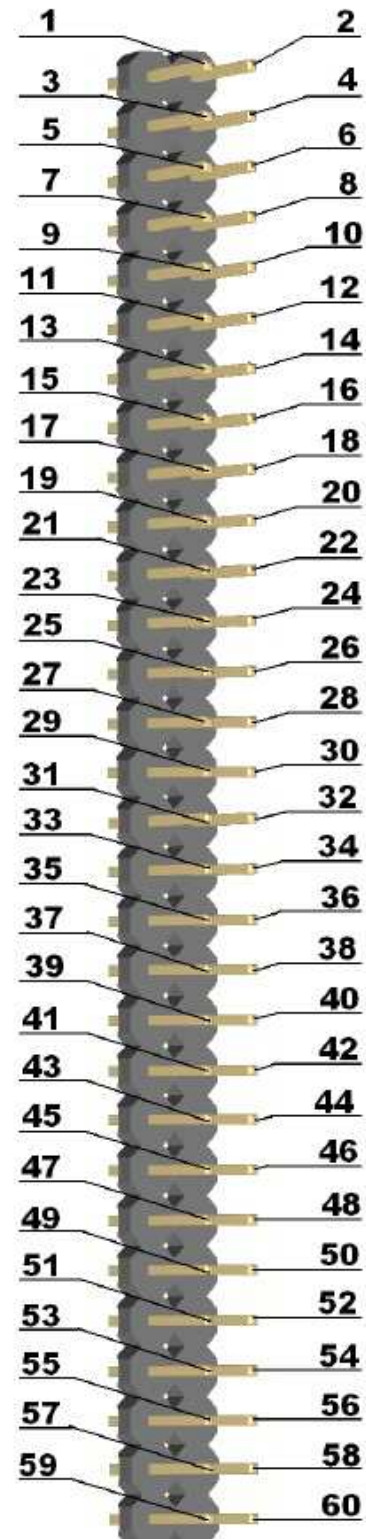


Рис. 2. Разъёмы на отладочной плате

3. Задания для лабораторной работы

Задание 1. Знакомство со средой разработки

Целью данного задания является общее знакомство с средой разработки IAR Embedded Workbench IDE, процессом отладки программы и прошивки флеш-памяти микроконтроллера.

Этапы выполнения задания

1. Запуск проекта

1.1. Запустите среду разработки IAR Embedded Workbench IDE, путём нажатия на ярлык программы на рабочем столе или путём выбора программы из меню Пуск.

1.2. Откройте проект кнопка_1, пройдя по пунктам меню File → Open → Workspace и выбрав проект кнопка_1.eww (рис. 3).

1.3. Изучите текст программы (листинг 1). С помощью преподавателя разберитесь, как работает и что делает приведённая в листинге 1 программа.

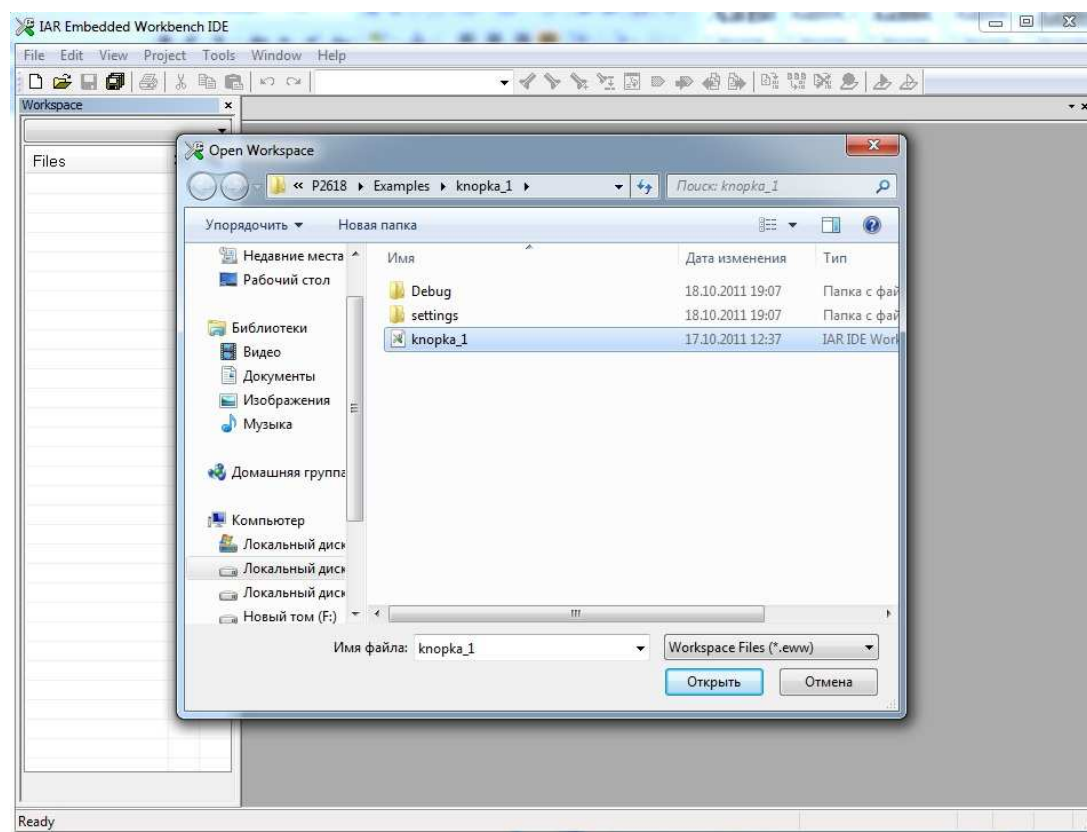


Рис. 3. Открытие проекта в IAR Embedded Workbench IDE

Листинг 1

```

#include "msp430x26x.h"

        NAME    main
        PUBLIC  main
        RSEG    CSTACK
        RSEG    CODE
RESET    MOV     #SFE(CSTACK), SP
        CALL    #Setup           ;Вызов процедуры настройки портов

main
Mainloop    MOV.b    &P6IN,R15
;поместим значение регистра входа порта P6 в регистр R15

        AND.b    #BIT1,R15
;логическое перемножение 0000 0010 и значения в регистре R15
;если получился 0, то кнопка нажата, а если 1, то не нажата

        JZ      inflame
;если флаг нуля установлен (кнопка нажата), то переход по метке inflame

        MOV.b    #BIT0,&P6OUT
;если флаг нуля сброшен (кнопка не нажата)
;подадим высокий уровень на P6.0 (погасим диод)

        JMP     Mainloop        ;зацикливание

inflame    MOV.b    #00h,&P6OUT
;подадим на ножку P6.0 низкий уровень (зажём светодиода)

        JMP     Mainloop        ;зацикливание

Setup     MOV     #WDTPW+WDTHOLD,&WDTCTL
;отключим сторожевой таймер, чтобы он не инициировал программный сброс

        MOV.b    #BIT0,&P6DIR
;установим работу порта P6.0 на выход

        MOV.b    #BIT0,&P6OUT
;выставим на ножку P6.0 высокий уровень напряжения
;(логическую единицу), чтобы диод погас

        EINT     ; Устанавливаем флаг разрешения прерываний

        RET
;Выходим из процедуры "setup" в программу.
;Точнее, возвращаемся в точку программы, откуда был вызван CALL

COMMON   INTVEC           ; Определяем вектора прерываний

ORG      RESET_VECTOR
DW       RESET
END

```

2. Выполнение программы в режиме эмуляции

Режим эмуляции применяется для отладки программы без использования самого микроконтроллера.

2.1. В левом верхнем окне, отображающем иерархию проекта, наведите курсор на самый верхний файл кнопка_1-Debug, и нажмите правую клавишу мыши. В появившемся всплывающем списке выберите пункт Options (рис. 4).

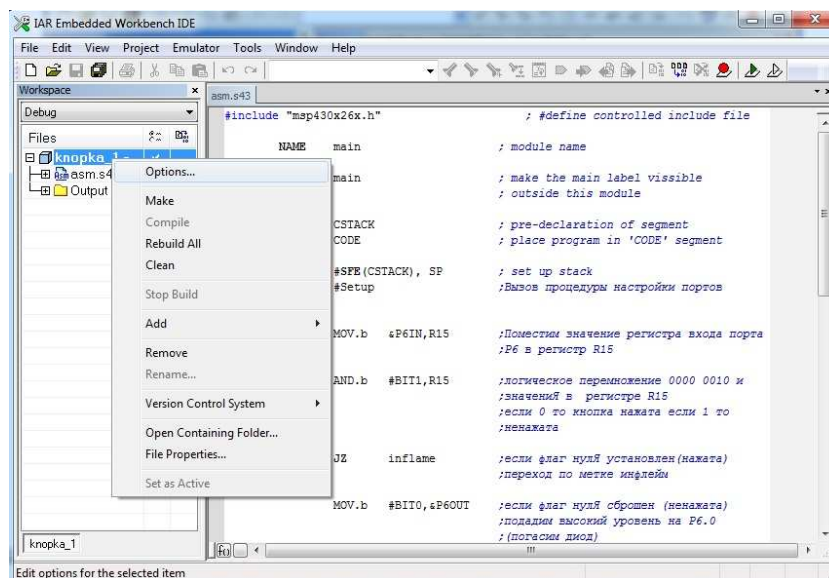


Рис. 4. Открытие опций проекта

2.2. В появившемся окне настроек проекта установите параметры так, как показано на рис. 5: Device – MSP430F2618, Driver – Simulator.

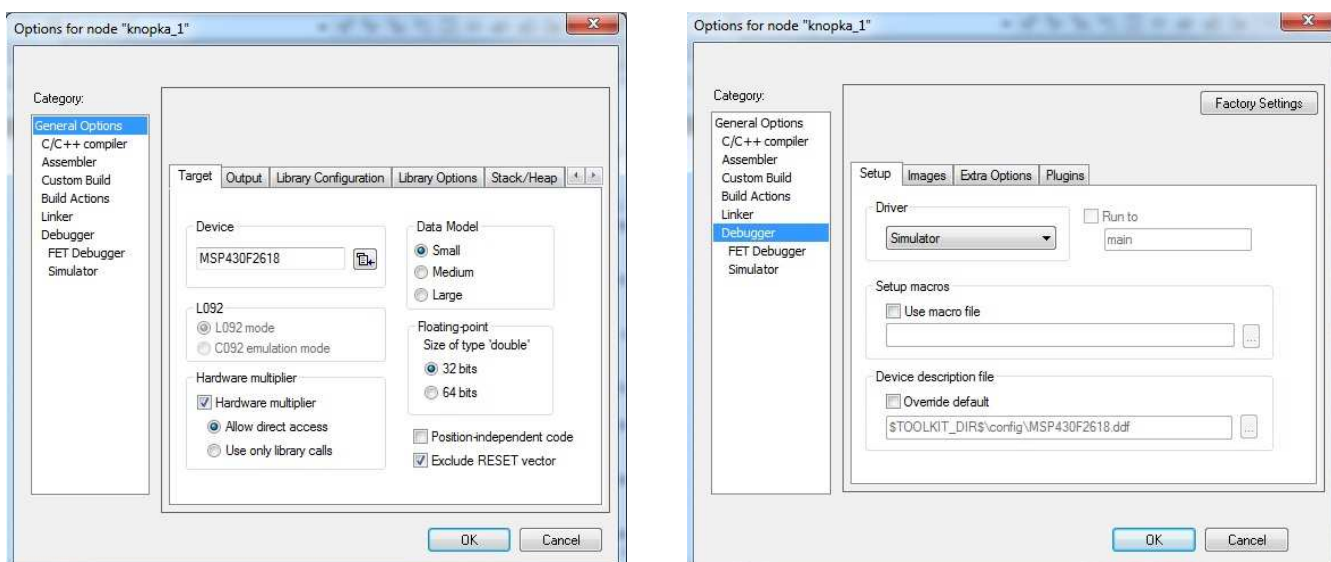


Рис. 5. Настройка опций проекта для эмуляции

2.3. Закройте окно Options, нажав кнопку ОК внизу окна. Запустите выполнение программы в режиме эмуляции, нажав на пиктограмму в виде зеленого треугольника, расположенную в верхней части окна программы, справа от ряда кнопок New, Open, Save all, или выбрав в меню Project пункт Download and debug (рис. 6).

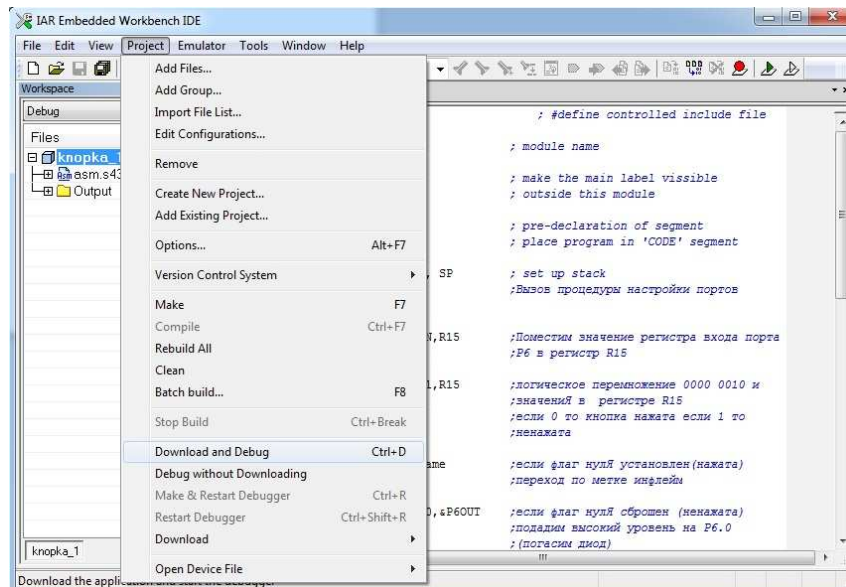


Рис. 6. Запуск проекта

2.4. Откройте окно Register и Watch 1, воспользовавшись меню View (рис. 7). Выведите в окно Register регистры микроконтроллера (CPU Registers), воспользовавшись выпадающим окном в верхней части рабочего окна.

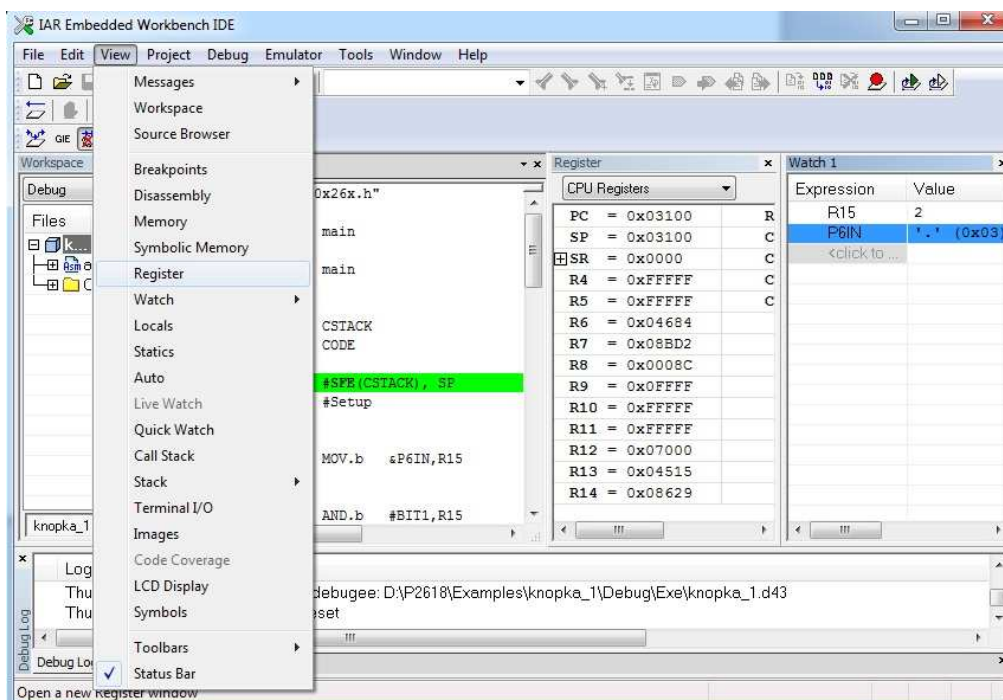


Рис. 7. Использование окон при отладке проекта

2.5. Нажимая на клавишу пошагового выполнения программы Step Into (рис. 8) или кнопку F11, выполните по шагам один цикл программы полностью. Ответьте на вопросы: 1) какие строчки кода программы выполняются? 2) как и почему меняются значения регистров микроконтроллера в процессе выполнения программы?

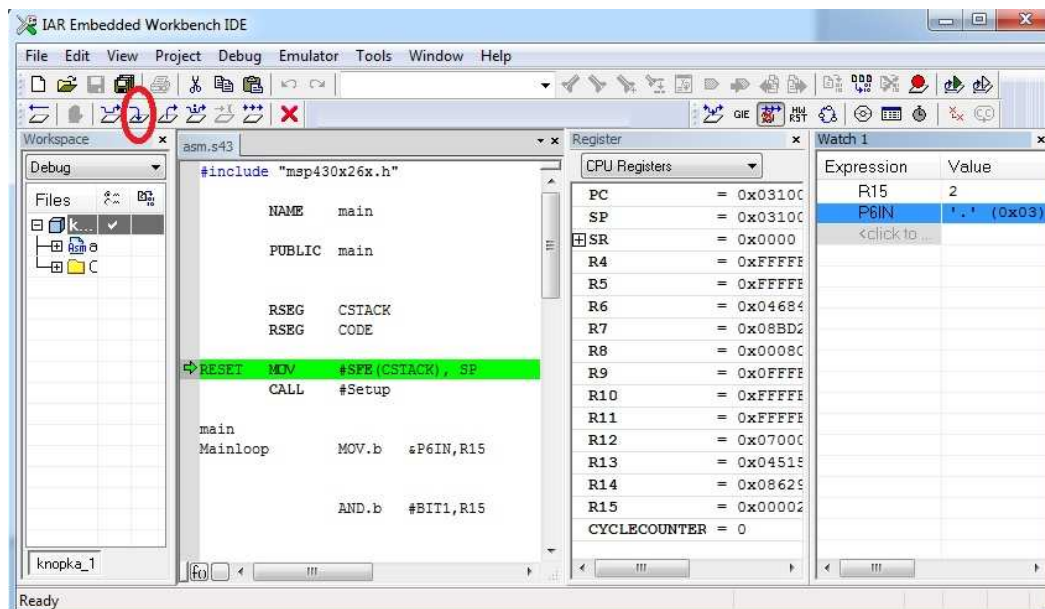


Рис. 8. Пошаговое выполнение проекта

3. Программирование микроконтроллера и выполнение предназначенной для него программы после её отладки

3.1. В окне настроек проекта установите параметры, соответствующие микроконтроллеру и типу JTAG - программатора: Device – MSP430F2618, Driver – FET Debugger, Connection – Olimex USB.

3.2. Подключите JTAG-программатор к компьютеру и микроконтроллеру.

3.3. Запустите процесс программирования микроконтроллера, выбрав пункт выпадающего меню Download and debug (рис. 6).

3.4. В окне Watch 1 нажмите левой клавишей на первую свободную строчку в поле Expression, введите с клавиатуры название входного регистра P6IN. Аналогично, введите название регистра процессора R15. При выполнении следующих заданий следите за изменением этих регистров.

3.5. Установите точку останова break point на строчку MOV.b #00h, &P6OUT (рис. 9), отметив её позицию левой клавишей мыши. После этого нажмите на кнопку Toggle Breakpoint в правой верхней части рабо-

чего окна. Обратите внимание на то, что помеченная строчка становится красной.

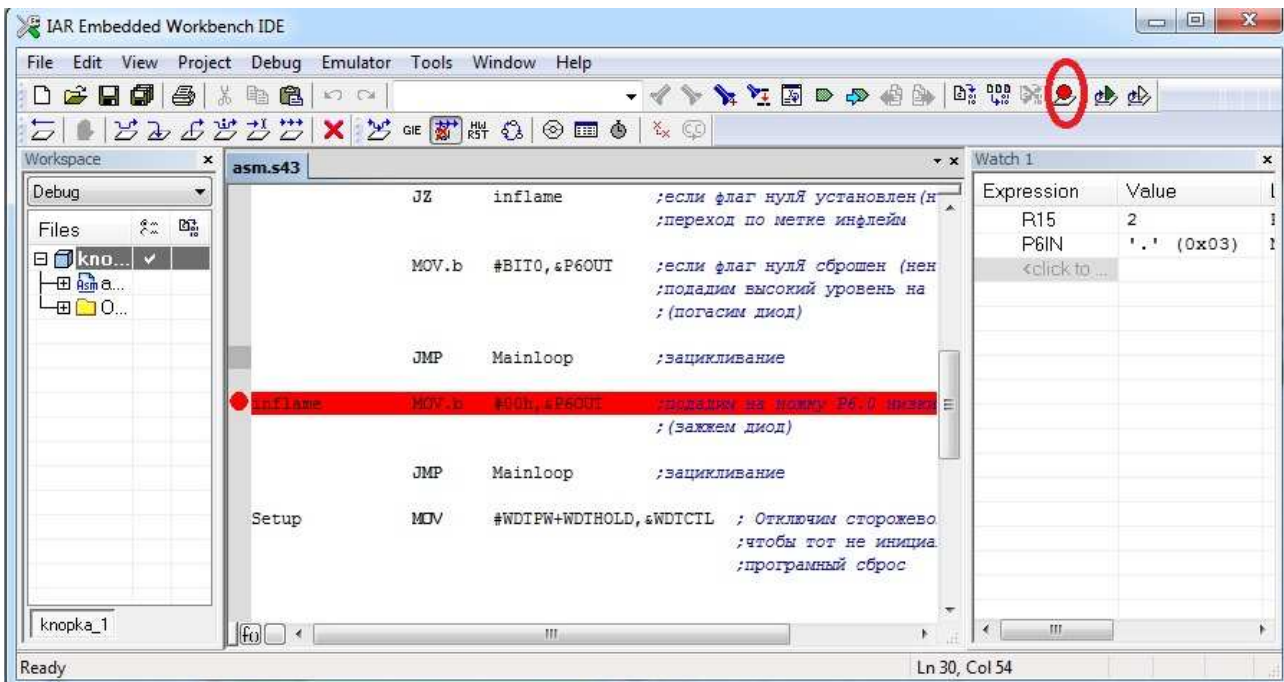


Рис. 9. Установка break point

3.6. Запустите программу на выполнение, нажав кнопку Go (рис. 10).

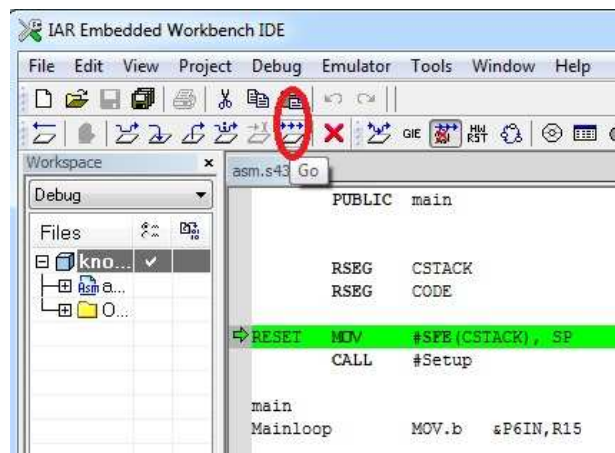


Рис. 10. Запуск программы

3.7. Обратите внимание на то, что на экране ничего не происходит и установленная в п. 3.5 точка останова (break point) не остановила выполнение программы. Почему?

3.8. Нажмите большую круглую кнопку на отладочной плате. Следите за тем, что происходит в окне программы.

3.9. Нажимая кнопку F11 (Step Into) продолжите выполнение программы в пошаговом режиме. По информации в окне Watch 1 следите за изменением переменных.

3.10. Пройдите в пошаговом режиме (F11) два полных цикла работы программы: один раз при отпущенной кнопке, другой раз при нажатой. Как меняется содержимое ячейки P6IN? Как ведёт себя светодиод на отладочной плате?

3.11. В окне "Watch 1" измените содержимое регистра P6OUT. Выполните ещё один цикл работы программы по шагам, нажимая клавишу Step Into. Каково теперь состояние светодиода?

3.12. Снимите метку break point, установленную в п. 3.5. Нажмите кнопку Go (F5). Как реагирует светодиод на нажатие кнопки?

3.13. Снимите все точки останова, после чего остановите процесс отладки программы, нажав клавишу Stop Debugging (рис. 11).

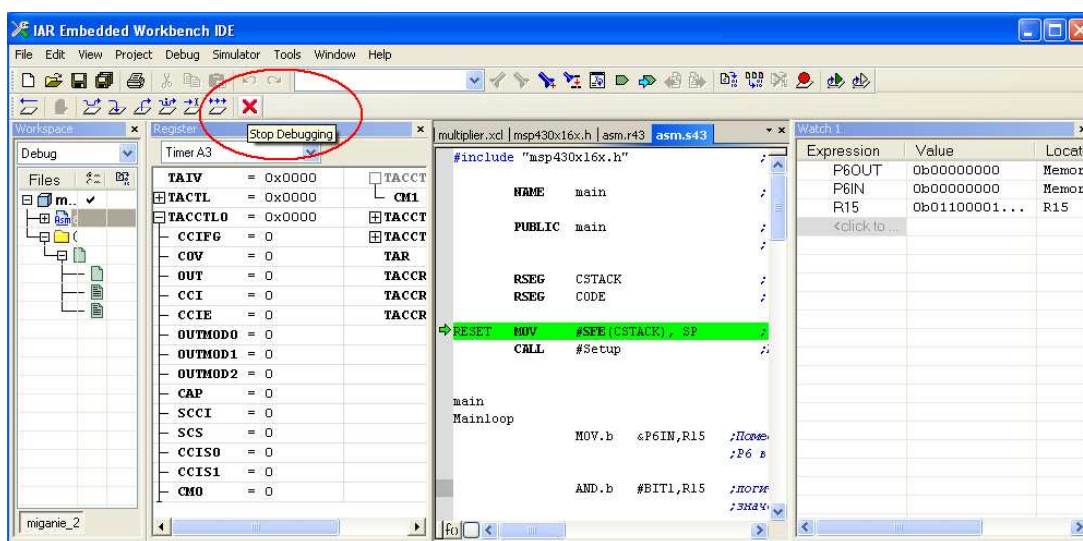


Рис. 11. Остановка отладки программы

3.14. Закройте программу IAR. Продолжает ли диод реагировать на нажатие кнопки?

3.15. Завершив работу, отсоедините JTAG от компьютера и отладочной платы.

Задание 2. Работа с прерываниями

Целью данного задания является получение навыков работы с прерываниями микроконтроллера на примере прерывания от таймера.

Этапы выполнения задания

1. Запуск проекта

1.1. Запустите проект `miganie_2` по той же схеме, что и в задании 1.

1.2. Изучите текст программы (листинг 1). С помощью преподавателя разберитесь, как работает и что делает программа, приведённая в листинге 1.

Листинг 2

```

#include "msp430x26x.h"           ; подключаем библиотеку
    NAME    main                 ; даём модулю имя main
    PUBLIC  main                 ; делаем его глобальным
    RSEG    CSTACK              ; резервируем место под сегмент стека
    RSEG    CODE                ; резервируем место под сегмент кода
RESET      MOV    #SFE(CSTACK), SP ; инициализируем регистр
                                     ; SP (указатель стека)
                                     ; Вызов процедуры настройки
                                     ; портов по метке #Setup
main
    MOV.b   &P6IN,R15           ;Поместим значение регистра
                                     ;входа порта P6 в регистр R15
    AND.b   #BIT1,R15           ;логическое перемножение
                                     ;0000 0010 и
                                     ;значения в регистре R15
                                     ;если 0, то кнопка нажата
                                     ;если 1, то не нажата
    JZ      Quater_Sec         ;если флаг нуля установлен
                                     ;(нажата), то
                                     ;переход по метке Quater_Sec
    MOV     #8000h,&CCR0        ;Период прерывания 1с
    JMP     main                ;снова main
Quater_Sec      MOV     #2000h,&CCR0 ;Установить период прерывания
                                     ;1/4 секунды
    JMP     main                ;снова main
Setup          MOV     #WDTPW+WDTHOLD,&WDTCTL ;Отключим сторожевой таймер
setupTA        MOV     #TASSEL0+TACLR, &TACTL ;Настройка режима таймера A
    BIS     #CCIE,&CCTL0        ;Разрешение прерывания от таймера A
    MOV     #8000h,&CCR0        ;Период прерываний установить 1сек
    BIS     #MC_1, &TACTL      ;Запуск таймера
    MOV.b   #BIT0,&P6DIR ;Установим работу порта P6.0 на выход
    EINT                                         ; Разрешение прерываний
    RET                                         ; Настройка закончена,
                                               ; возврат
CCR0INT        ;процедура обработки прерывания таймера A
    XOR.b   #BIT0,&P6OUT        ;изменим значение регистра P6.0
                                     ;на противоположный
    RETI                                         ;возврат из процедуры обработки прерывания

COMMON        INTVEC           ; Вектора прерываний
ORG           TIMERA0_VECTOR    ; Timer_A0 Вектор
DW           CCR0INT            ;Переход на процедуру обработки
                                     ; прерывания таймера A
ORG           RESET_VECTOR      ;POR, ext. Reset (низкое напряжение
DW           RESET              ;питания, внешний сброс)
END

```

2. Элементарная модернизация проекта

Измените проект так, чтобы

- 1) период мигания диода при отпущенной кнопке стал равен 2-м секундам (можно ли сделать мигание ещё более медленным?),
- 2) при отпущенной кнопке светодиод мигал чаще, чем при нажатой,
- 3) при нажатой кнопке светодиод вообще не загорался.

3. Разработка собственного проекта

Получите у преподавателя индивидуальное задание и, взяв за основу код программы из листинга 2, выполните его.

4. Контрольные вопросы

1. Что такое *.h файл и для чего он нужен?
2. Зачем нужны строчки «NAME main», «PUBLIC main» и просто «main»? Что они делают?
3. В какой области памяти в микропроцессоре размещается написанный и скомпилированный код? Что такое ОЗУ, ПЗУ, флеш-память?
4. Как в программе обрабатывается событие Reset? Что такое вектор прерывания «RESET_VECTOR»? Какое значение туда записывается?
5. Что обозначает .b в конце команд AND и MOV в тексте программ?
6. Каковы режимы адресации операндов. Что обозначает #, & перед операндами в тексте программы?
7. Что такое флаги АЛУ? Что является условием для операции условного перехода JZ?
8. Что такое прерывания? Зачем они нужны? Как происходит их обработка и чем вызов процедуры обработки прерывания отличается от вызова функций?
9. Как происходит настройка таймера A? Как устанавливается направление счета, максимальное значение и источник тактирования таймера? Какие прерывания от таймера разрешаются? Как происходит запуск и остановка таймера?

Список литературы

1. Семенов Б.Ю. Микроконтроллеры MSP430. Первое знакомство, М.: Изд-во «Солон-пресс», 2006, 120 с.
2. Шкелёв Е.И. Электронные цифровые системы и микропроцессоры. Учебное пособие. //Н.Новгород: Изд.ННГУ, 2004, 152 с.
3. Цифровые процессоры обработки сигналов. Справочник. // Под редакцией А.Г. Остапенко. – М.: Радио и связь, 1994.
4. Электронная версия руководства пользователя (оригинал) // <http://www.ti.com/lit/ug/slau144j/slau144j.pdf>
5. Семейство микроконтроллеров MSP430x2xx: Архитектура. Программирование. Разработка приложений. М.: "Додека XXI". 2010.
6. Электронный учебный курс по MSP 430 // <http://we.easyelectronics.ru/blog/msp430>
7. MSP430-P2618 development board. Users manual

Содержание

1. Введение	3
2. Описание установки	3
3. Задания для лабораторной работы	5
Задание 1. Знакомство со средой разработки	5
Задание 2. Работа с прерываниями	11
4. Контрольные вопросы	13
Список литературы	14

Евгений Иванович Шкелёв
Алексей Викторович Иванов
Владимир Андреевич Калинин
Владимир Владимирович Пархачёв

ПЕРВЫЕ ШАГИ В ПРОГРАММИРОВАНИИ МИКРОКОНТРОЛЛЕРА СЕРИИ MSP430

Практикум

Федеральное государственное автономное образовательное учреждение
высшего профессионального образования «Нижегородский государственный
университет им. Н.И. Лобачевского».
603950, Нижний Новгород, пр. Гагарина, 23.