

Министерство образования и науки Российской Федерации
Государственное бюджетное образовательное учреждение высшего профессионального образования
«Томский государственный университет систем управления и радиоэлектроники»

УТВЕРЖДАЮ

Зав.кафедрой ЭС

_____ Н.Е.Родионов
" ____ " _____ 2012 г.

Вводится в действие с " ____ " _____ 20 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

по дисциплине

Микропроцессорные устройства и системы

Составлена кафедрой

Электронных систем

Для студентов, обучающихся
по направлению подготовки 220600 «Инноватика»
по специальности 220601.65 «Управление инновациями»

Форма обучения

очная

Составитель доцент кафедры
Электронных систем, к.ф.-м.н.

Антипин М.Е.

" 04 " июля 2012 г

Томск 2012 г.

Введение

Лабораторные работы обеспечивают возможность закрепить знания, полученные в лекционной части курса, работая с реальным оборудованием и программным обеспечением. В процессе выполнения данных работ студенты получают основы программирования микроконтроллеров SuperH2A фирмы Renesas.

Общие требования

Лабораторные работы выполняются студентами очной формы обучения индивидуально под контролем со стороны преподавателя. Все консультации осуществляются преподавателем. Число студентов, одновременно присутствующих на занятии не должно превышать 8 человек. Если в списочном составе группы студентов больше 8, то группа должна быть разделена на подгруппы численностью от 4 до 8 человек в каждой.

Для выполнения лабораторных работ целесообразно в учебном расписании выделять 4 академических часа подряд, без больших перерывов. Расписание также должно предусматривать отдельное проведение занятий у подгрупп, если группа была разделена.

Перед началом занятий студенты должны изучить инструкцию по охране труда, действующую в лаборатории. Преподаватель должен убедиться в знании инструкции, задавая студенту вопросы по ее содержанию, после чего сделать соответствующую запись в журнале охраны труда.

Во время проведения лабораторных занятий в аудитории (лаборатории) студентам запрещается передавать друг другу файлы и другие материалы, являющиеся результатом выполнения заданий.

Студент имеет право:

- Выходить из аудитории (лаборатории) не спрашивая разрешения у преподавателя.
- Самостоятельно распределять аудиторное время, определяя необходимость перерыва или непрерывной работы.
- Просить консультации у преподавателя, если он в текущий момент не распределяет задания, не принимает выполненные работы и не консультирует другого студента.

Преподаватель, давая консультацию студенту, указывает раздел технической документации или методической литературы, в которой имеется ответ на вопрос студента. Если необходимые сведения в документации и литературе отсутствуют, то преподаватель должен дать устные пояснения или продемонстрировать практические действия, приводящие к требуемому результату, с последующей отменой для повторения студентом.

Самостоятельная работа студентов над лабораторными заданиями осуществляется в той же аудитории (лаборатории), где проводятся

лабораторные занятия. Преподаватель должен согласовать со студентами расписание самостоятельной работы - не менее 2 астрономических часов в неделю. В указанное время по учебному расписанию студентов и в аудитории (лаборатории) не должны проводиться другие занятия. Преподаватель должен обеспечить доступ студентов в аудиторию (лабораторию) в указанные часы. Необходимость самостоятельной работы определяет студент.

Консультации, выдача лабораторных заданий и прием результатов выполнения осуществляется только во время аудиторных занятий. Задания выполняются последовательно. Правильное выполнение некоторых заданий возможно только, если студент корректно выполнил предыдущие задания. Поэтому приступать к следующему заданию студент может, только сдав преподавателю результат выполнения предыдущего.

Техническое обеспечение практических работ

Для выполнения лабораторных работ студенту предоставляется индивидуальное рабочее место, в состав которого входят:

- стул;
- стол;
- персональный компьютер с операционной системой Windows XP;
- программное обеспечение фирмы Renesas;
- отладочная плата разработки фирмы Renesas на основе микроконтроллера SuperH2A.

Размещение и освещенность рабочих мест в учебной аудитории (лаборатории) должно удовлетворять действующим требованиям СанПиН.

Прием результатов выполнения лабораторных работ

Результаты выполнения лабораторных работ демонстрируются преподавателю. Во время приема выполненной работы преподаватель вправе:

- Требовать у студента демонстрации выполнения программы, предусмотренной заданием.
- Самостоятельно производить манипуляции с отладочной платой и средой программирования, не изменяя программы, составленной студентом.
- Требовать у студента пояснений, относящихся к исходному коду и способам реализации программы.

Задание считается выполненным и принимается преподавателем только в том случае, если реализованы все работы, предусмотренные заданием. Если какие то работы не выполнены, или выполнены неверно, то результат выполнения подлежит доработке. Студент должен работать над

кодом программы максимально самостоятельно, использовать отладочные средства, предоставляемые средой программирования.

Результаты выполнения заданий сохраняются преподавателем в электронном виде и хранятся в течение двух лет.

До конца семестра студент должен сдать результаты выполнения всех лабораторных работ, предусмотренным настоящими указаниями. В противном случае студент к сдаче экзамена не допускается.

Темы лабораторных работ

1. Возможности отладчика среды HEW. Описание работы в приложении А.
2. Ассемблерные инструкции микроконтроллера SuperH2A. Описание работы в приложении Б.

Приложение А

Возможности отладчика среды HEW

Цель : Получить практические навыки работы с отладчиком.

ВНИМАНИЕ

После выполнения каждого задания выполнить следующие 2 шага для того, чтобы удостовериться в правильности выполняемых действий:

А. Построить проект (см. Рис. 1).

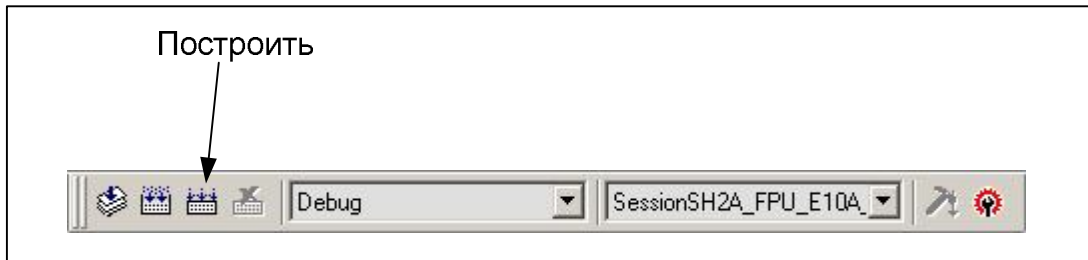


Рис. 1. Меню Standart

В. Прошить проект «Записать программу в кит» (см. Рис. 2).

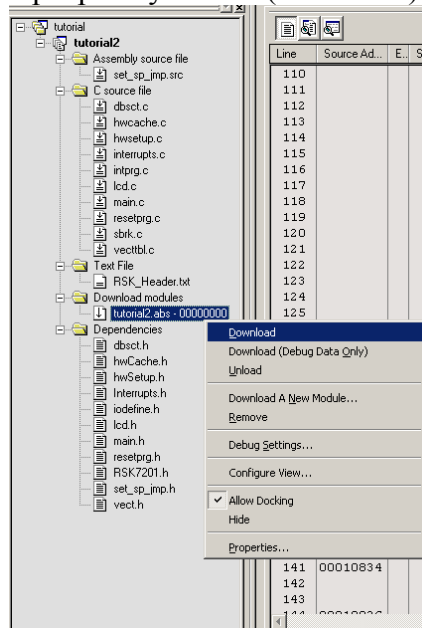


Рис. 2. «Дерево проекта»

С. Выполнить команду GO (см. рис. 6).

Задание:

1. Открыть проект Tutorial2. Установить соединение.
2. Внутри фоновой задачи написать программу, которая:
 1. Включает зеленый светодиод, делает секундную задержку,
 2. Включает оранжевый светодиод, делает секундную задержку,
 3. Включает красный светодиод, делает секундную задержку.
 4. Затем светодиоды должны потухнуть в том же порядке, с секундной задержкой.

Определения:

Светодиоды

RSK_LED0

RSK_LED1

зеленый светодиод;

оранжевый светодиод;

RSK_LED2 красный светодиод;
 Состояние светодиодов
 RSK_LED_ON - светодиод включен;
 RSK_LED_OFF - светодиод выключен;

Примечание1:

Программная задержка должна быть в виде холостого цикла (например,for()).

Примечание2:

Пример работы со светодиодом:

```
RSK_LED0 = RSK_LED_ON                   // включить зеленый светодиод;  

RSK_LED0 = RSK_LED_OFF                 // выключить зеленый светодиод;
```

3. Написать на LCD экране: «Hi» «Имя» (в две строчки)

Функции:

```
DisplayString(LCD_LINE1, "8 симв."); //для вывода в 1-ю строчку  

DisplayString(LCD_LINE2, "8 симв."); //для вывода во 2-ю строчку
```

4. Определить 3 глобальных переменных

```
unsigned int var_a;  

unsigned short var_b;  

unsigned char var_c;
```

Посмотреть их значения (по умолчанию). Для этого как показано на Рис. 3 выберете окно просмотра переменных Watch

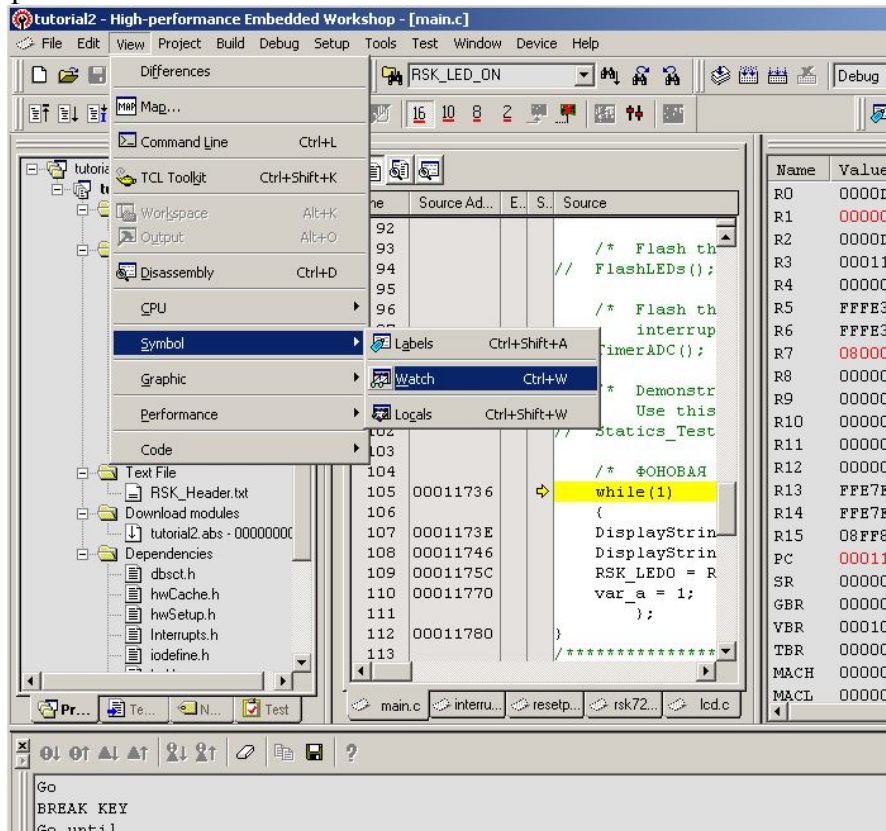


Рис. 3. Элементы Меню HEW

Нажать на кнопку Add Watch (смотри рис. 4)

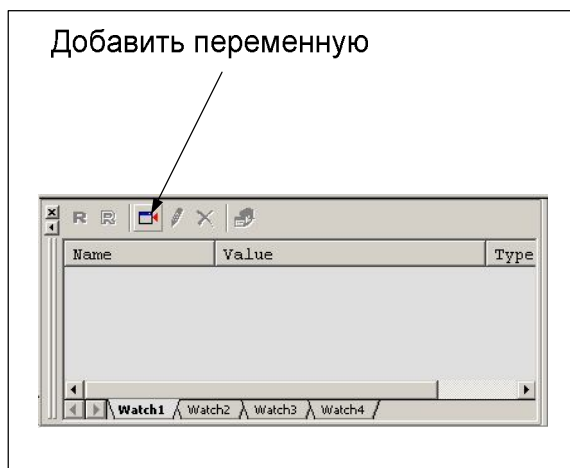


Рис. 4. Окно просмотра глобальных переменных
В окне Add Watch записать имя переменной и нажать ОК.

5. Определить 3 локальных переменных (переменные внутри функции main();)
`unsigned int loc_a;`
`unsigned short loc_b;`
`unsigned char loc_c;`

Посмотреть их значения (по умолчанию). Для этого как показано на рис. 3. выберите окно просмотра переменных Locals.

6. Присвоить локальным и глобальным переменным произвольные значения внутри фоновой задачи.

Прошить и Запустить программу. Посмотреть как изменилось значение всех 6-ти переменных.

7. Посмотреть регистры (см. Рис. 5).

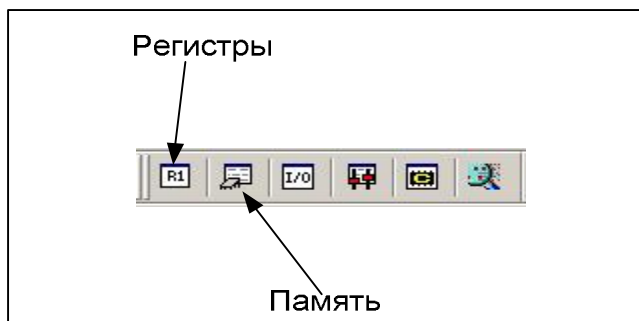


Рис. 5. Меню CPU

8. Посмотреть и найти адрес всех переменных в памяти (см. Рис. 5).

9. Выполнить все команды Меню Debug Run (см. Рис. 6).

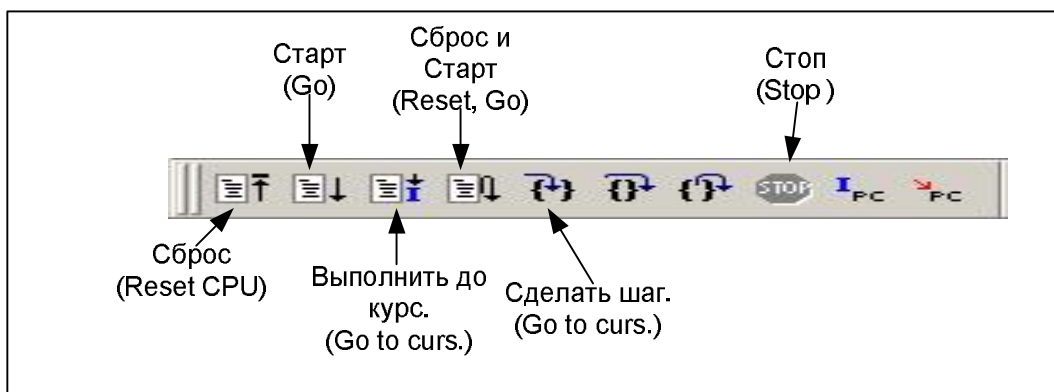


Рис. 6. Меню Debug Run

Контрольный вопрос: чем отличается «Step In» от «Reset CPU»

Приложение Б Ассемблерные инструкции

Цель : Получить практические навыки по использованию Ассемблерных инструкций.

Ход работы:

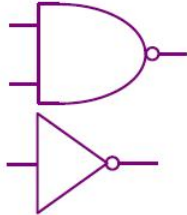
3. Открыть Проводник, сделать копию проекта Tutorial3. Ассемблерный код будем записывать в файле с именем tutorial_asm.src. Внести изменения в заголовок файла - tutorial_asm.src.
4. Арифметические (логические) инструкции.

Теория:

Arithmetic/Logic Instructions

- **ADD, SUB**
- **Compare: CMP**
- **Divide support: DIV**
- **Multiply $Rn * Rm \Rightarrow$ MUL**
- **Multiply $@Rn * @Rm \Rightarrow$ MAC**
- **Negate: NEG**
- **AND, OR, TST, XOR**
 - Addressing modes for AND, OR, TST, XOR:
 - AND Rm,Rn, AND #dd:8,Rn, AND #dd:8,@(R0,GBR)
- **TAS:**
 - TAS.B @Rn ;"test and set" operates on byte and T-bit
- **NOT:**
 - NOT Rm,Rn ; ~Rm \rightarrow Rn

Select a button to learn more about the instructions for the various types of operations.



Задание:

4.1. Инструкция ADD:

Записать нижеприведенный код:

```
;Arithmetic/Logic instructions
MOV.L    #1,R8          ;
MOV.L    #2,R7          ;
ADD      R8,R7          ;ADD Rm, Rn
ADD      #1,R7          ;ADD #imm, Rn
```

Сохранить Workspase. Построить (Build All). Прошить КИТ. Оценить результат.

4.2. Инструкция SUB:

Записать нижеприведенный код:

```
SUB      R8,R7          ;SUB Rm,Rn
```

Сохранить Workspase. Построить (Build All). Прошить КИТ. Оценить результат.

4.3. Инструкция CMP:

Записать нижеприведенный код:

```
CMP/EQ   R8,R7          ;CMP/EQ Rm,Rn
```

Сохранить Workspase. Построить (Build All). Прошить КИТ. Оценить результат.

Инструкции из следующего списка применить самостоятельно:

CMP/HS
CMP/GE
CMP/HI
CMP/GT
CMP/PL
CMP/PZ

Сохранить Workspace. Построить (Build All). Прошить КИТ. Оценить результат.

4.4. Инструкция DIV:

Записать нижеприведенный код:

```
MOV.L    #2, R0      ;  
MOV.L    #6, R7      ;  
DIVU     R0, R7      ;DIVU     R0, Rn (Unsigned)  
  
MOV.L    #-2, R0     ;  
MOV.L    #6, R7      ;  
DIVS     R0, R7      ;DIVS     R0, Rn (Unsigned)
```

Сохранить Workspace. Построить (Build All). Прошить КИТ. Оценить результат.

4.5. Инструкция MUL:

Записать нижеприведенный код:

```
MOV.L    #2, R0      ;  
MOV.L    #8, R7      ;  
MULR     R0, R8      ;MULR     R0, Rn
```

Сохранить Workspace. Построить (Build All). Прошить КИТ. Оценить результат.

4.6. Инструкция MUL: (умножение с использованием спец. регистров - MACCL)

Записать нижеприведенный код:

```
MOV.L    #2, R8      ;  
MOV.L    #6, R7      ;  
MUL.L    R7, R8      ;MUL.L    Rm, Rn
```

Сохранить Workspace. Построить (Build All). Прошить КИТ. Оценить результат.

4.7. Инструкция NEG:

Записать нижеприведенный код:

```
NEG      R0, R7      ;NEG      Rn, Rm
```

Сохранить Workspace. Построить (Build All). Прошить КИТ. Оценить результат.

4.8. Инструкции OR, XOR NOT, AND, .

Придумать примеры и выполнить самостоятельно.

Instruction	Code	Operation	Cycles	T Bit	Compatibility		
					SH2E	SH4	New SH-2A/SH2A-FPU
AND Rm, Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—	Yes	Yes	
AND #imm, R0	1100100111111111	R0 & imm → R0	1	—	Yes	Yes	
NOT Rm, Rn	0110nnnnmmmm0111	~ Rm → Rn	1	—	Yes	Yes	
OR Rm, Rn	0010nnnnmmmm1011	Rn Rm → Rn	1	—	Yes	Yes	
OR #imm, R0	1100101111111111	R0 imm → R0	1	—	Yes	Yes	
XOR Rm, Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—	Yes	Yes	
XOR #imm, R0	1100101011111111	R0 ^ imm → R0	1	—	Yes	Yes	
XOR.B #imm, @(R0, GBR)	1100111011111111	(R0+GBR) ^ imm → (R0+GBR)	3	—	Yes	Yes	

Сохранить Workspace. Построить (Build All). Прошить КИТ. Оценить результат.

5. Инструкции передачи данных

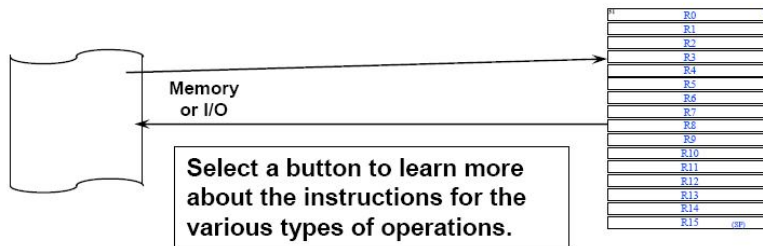
Теория:

Data Transfer Instruction: Move

- **MOV:** General move to / from registers; byte, word and longword operand
 - No modification of status register (*in contrast to H8 series*)
- **Immediate 8-bit:** MOV.B #value:8,Rn
- **Immediate 20-bit:** MOV.I20 #value:20,Rn
- **Immediate 16-bit:** MOV.W @(displ:8,PC),Rn
 - address: displ*2 + PC
- **Immediate 32-bit:** MOV.L @(displ:8,PC),Rn
 - address: displ*4 + PC

← In Instruction

← In Literal Pools



Задание:

Записать нижеприведенный код:

```

;Transferring Data instructions
MOV.B    #D'127, R7    ;
MOV.I20  #123456, R8   ;
MOV.W    @(0,PC), R7   ;
MOV.L    @(0,PC), R8   ;

```

Сохранить Workspace. Построить (Build All). Прошить КИТ. Оценить результат.

6. Инструкции перехода, вызова подпрограммы

Теория:

Program Flow Instructions

- **Conditional & unconditional delayed branches** (same as SH-2 CPU)
- **Conditional branch coding and handling sequence:**
 - T-bit handling with COMPARE instruction
 - Result of condition is tested in T-bit
 - Then branch conditional
- **Unconditional branch with no delay slot** (added instruction)
- **A subroutine call instruction causes the hardware to:**
 - Copy PC contents in PR
 - Put new value into PC
 - Go to next instruction
- **A return from subroutine instruction causes the hardware to:**
 - Copy PR contents in PC
 - Go to next instruction
- **Exception handling**

Задание:

6.1. Условный переход (delay) BT/S, BF/S

Самостоятельно

Сохранить Workspase. Построить (Build All). Прошить КИТ. Оценить результат.

6.2. Безусловный переход (delay) BRA

Самостоятельно

Сохранить Workspase. Построить (Build All). Прошить КИТ. Оценить результат.

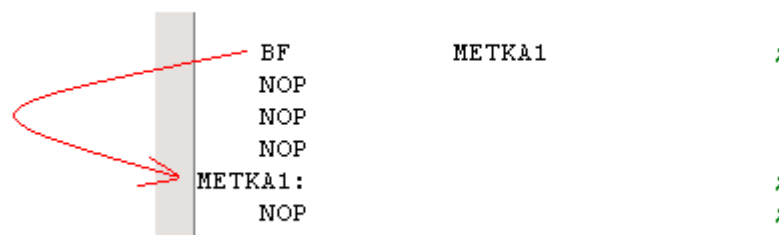
6.3. Условный переход BT

Самостоятельно

Сохранить Workspase. Построить (Build All). Прошить КИТ. Оценить результат.

6.4. Безусловный переход BF:

Безусловный переход



Сохранить Workspase. Построить (Build All). Прошить КИТ. Оценить результат.

6.5. Вызов подпрограммы JSR, Завершение подпрограммы RTS

Написать подпрограмму, например:

```

; **
; Function name      : _tutor2
; In (Вход)         :
; Out (Выход)       :
; Call (Вызов)      :
; Description (описание) :
_tutor2:
    NOP
    RTS                ; exit from _tutor2
    NOP

```

В функции _tutor1 оформить вызов подпрограммы:

```

MOV.L    Af_tutor2, R1
JSR      @R1

```

Код подпрограмм "завершен"

```

.ALIGN 4
Af_tutor2 .DATA.L _tutor2

```

Код данных

Метка подпрограммы

Сохранить Workspace. Построить (Build All). Прошить КИТ. Оценить результат.

Контрольный вопрос: Какие регистры и в каком порядке сохраняются в стек при выполнении инструкции JSR?