

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ПРОГРАММНЫХ СИСТЕМ

Отчет по лабораторной работе
Средства отладки программного обеспечения

Выполнил:

Кислюк И. В.

студент группы К4120

Проверил: Осипов Н. А.

Санкт-Петербург
2017

ЦЕЛЬ РАБОТЫ:

- ознакомление с аппаратными и программными средствами отладки ПО;
- изучение команд отладчика среды AVR Studio;
- приобретение навыков отладки программ под управлением отладчика.

ХОД РАБОТЫ:

Пошаговое выполнение программы задаётся командами Step Into, Step Over меню Debug.

Команда Step Into позволяет выполнить одну команду программы (в том числе команду вызова подпрограммы). Для завершения выполнения подпрограммы может использоваться команда Step Out. Команда Step Over также выполняет одну команду программы, но если это команда вызова подпрограммы, последняя полностью выполняется за один шаг. Следующая выполняемая команда (команда, адрес которой содержится в программном счётчике) обозначается символом стрелки в окне исходного текста программы. Сброс выполнения программы осуществляется с помощью команды Reset.

Прогон (запуск или продолжение выполнения) программы осуществляется командой Run. Для остановки выполнения программы служит команда Break.

Контрольные точки представляют собой специальные маркеры для программы-отладчика и могут быть трёх типов: точки останова, точки трассировки и точки наблюдения.

Точки останова задаются командой Toggle Breakpoint меню Debug или контекстного меню редактора исходного текста программы. Пример настройки Breakpoint приведен на рисунке 1

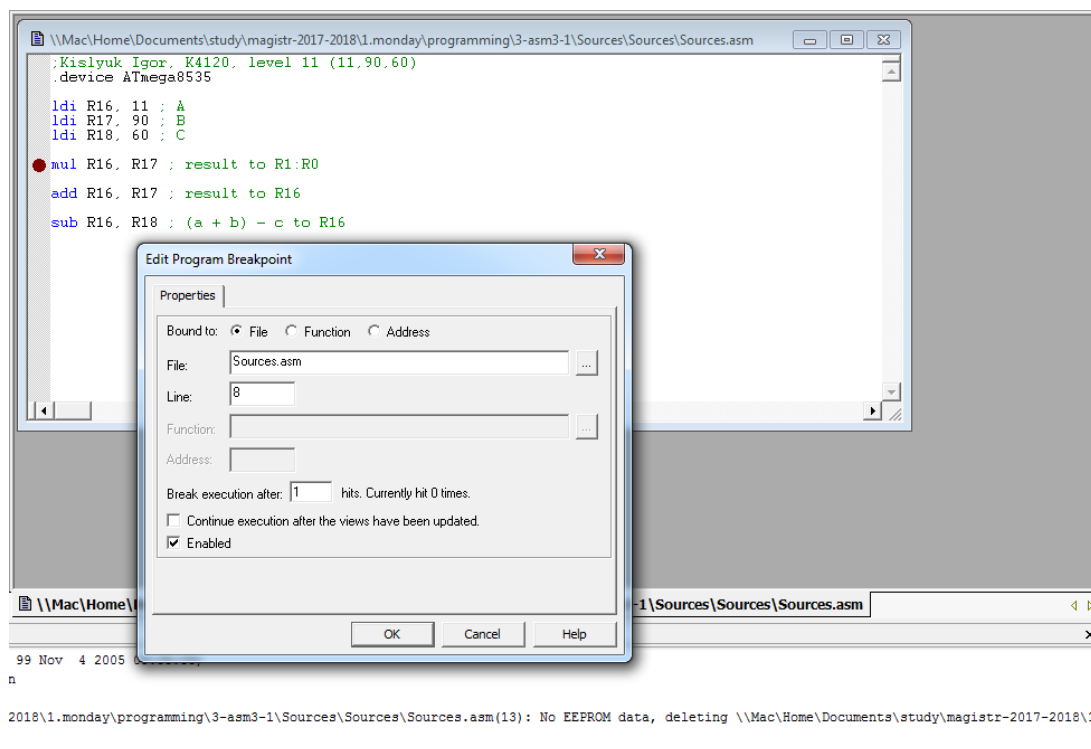


Рисунок 1 – Пример настройки breakpoint

Точка останова обозначается в редакторе исходного текста специальным символом слева от помечаемой строки. Просмотреть заданные точки останова можно на закладке

Breakpoints окна Output; там же точки останова могут быть запрещены (путём сброса флажка напротив точки останова) и разрешены (путём установки флажка). При достижении точки останова во время прогона программы её выполнение приостанавливается. Повторный вызов команды установки точки останова на той же строке программы приводит к удалению точки останова. Удалить все заданные точки останова позволяет команда Remove Breakpoints меню Debug или команда Remove all Breakpoints контекстного меню закладки Breakpoints окна Output.

Параметры точки останова задаются в диалоговом окне Breakpoint Condition, вызов которого осуществляется командой Breakpoints Properties контекстного меню редактора исходного текста программы. Установка флажка Iterations позволяет задать количество итераций (повторных выполнений) команды до останова прогона программы. При установке флажка Watchpoint по достижению точки останова производится только обновление значений регистров и ячеек памяти в окнах просмотра.

Флажки Iterations и Watchpoint не должны устанавливаться одновременно. Установка флажка Show message обеспечивает отображение сообщений о достижении точки останова на закладке Breakpoints окна Output. Вызов диалогового окна задания свойств и удаление точки останова могут быть произведены из контекстного меню закладки Breakpoints окна Output.

Точки трассировки предназначены для контроля выполнения программы в режиме реального времени. Трассировка позволяет отслеживать так называемую трассу программы – изменение содержимого регистров и ячеек памяти при выполнении определённых команд (команд, по адресам которых заданы точки трассировки). В среде AVR Studio функция трассировки может использоваться только при отладке программы с применением внутрисхемного эмулятора; при работе в режиме симулятора функция трассировки недоступна.

Точки наблюдения задаются командой Add to Watch контекстного меню редактора исходного текста программы. Точки наблюдения представляют собой символические имена регистров или ячеек памяти, содержимое которых необходимо отслеживать. При выполнении команды Add Watch на экране появляется окно Watches, разделённое на четыре столбца: Name (символическое имя точки наблюдения), Value (значение), Type (тип), Location (местонахождение).

Новая точка наблюдения может быть также задана в выделенной ячейке столбца Name окна Watches или командой Quickwatch в окне редактора исходного текста программы (при этом курсор должен находиться на имени регистра или ячейки памяти). Значения, отображаемые в столбце Value, обновляются при изменении содержимого соответствующего регистра или ячейки памяти. Удалить заданные точки наблюдения можно из окна Watches.

Отладчик среды AVR Studio также обеспечивает следующие функции: выполнение до курсора (команда Run to Cursor меню Debug, пример приведен на рисунке 2) и последовательное выполнение команд с паузами между ними (команда Auto Step меню Debug).

Для удобства использования в процессе отладки ряд команд отладчика доступен с клавиатуры (табл. 1)

Таблица 1

Команда отладчика	Клавиша	Команда отладчика	Клавиша
Run	F5	Step into	F11
Break	Ctrl+F5	Step out	Shift+F11
Reset	Shift+F5	Step over	F10
Run to cursor	Ctrl+F10	Toggle breakpoint	F9

Для просмотра и изменения содержимого регистров и ячеек памяти служат команды Registers, Memory, Memory 1, Memory 2, Memory 3 меню View.

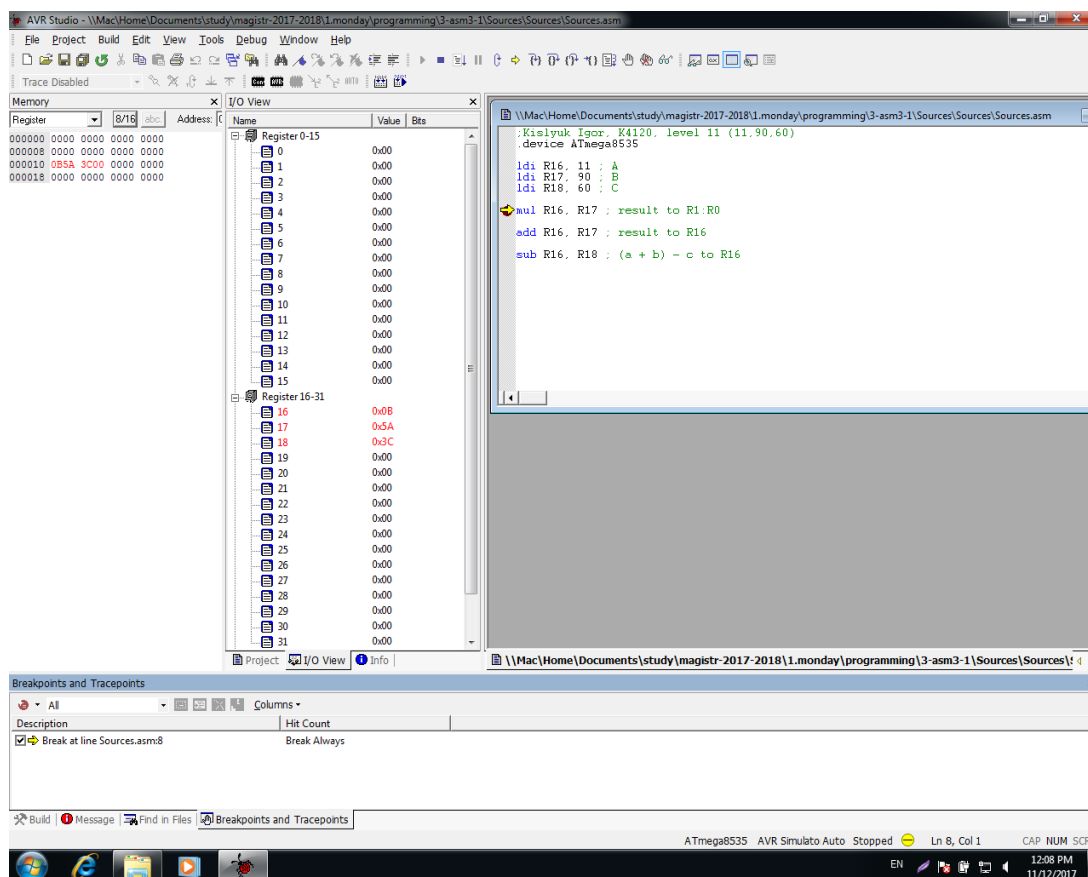


Рисунок 2 – Пример работы команды Run to Cursor

По команде Registers на экране отображается окно Registers, в котором приводятся шестнадцатеричные представления содержимого РОН. Изменение (модификация) содержимого регистров производится путём двойного щелчка мышью. Наблюдение за содержимым РОН может быть также произведено с помощью дерева устройств микроконтроллера, находящегося на закладке I/O окна Workspace. Для этого необходимо раскрыть объекты Register 0-15 и Register 16-31 щелчком мыши по знаку «+».

Команды Memory, Memory 1, Memory 2, Memory 3 обеспечивают вызов окон Memory, служащих для отображения содержимого ячеек оперативной и энергонезависимой памяти данных, памяти программ, регистров ввода-вывода и РОН. Выбор типа памяти, отображаемой в окне Memory, производится с помощью списка, расположенного в панели управления окна (Data – оперативная память данных, Eeprom – энергонезависимая память данных, I/O – регистры ввода-вывода, Program – память программ, Register – РОН).

Для контроля содержимого регистров ввода-вывода необходимо раскрыть объект I/O * закладки I/O окна Workspace, где * – тип микроконтроллера. Регистры ввода-вывода, входящие в объект I/O, сгруппированы по типам периферийных устройств

ВЫВОД:

Основными программными средствами отладки являются симуляторы и отладчики. Симуляторы (simulator) или симуляторы системы команд представляют собой программы, имитирующие работу того или иного процессора на уровне его команд.

Симуляторы обычно используются для проверки программы или её отдельных частей перед испытанием на аппаратных средствах.

Отладчики (debugger) представляют собой программы, предназначенные для анализа работы созданного программного обеспечения.