Интегральная среда разработки «CodeMaster-Leon»

Руководство системного программиста

Аннотация

Программное обеспечение (ПО) «CodeMaster-Leon» предназначено для разработки и отладки программ, предназначенных для микропроцессора 1906ВМ016 (на основе ядра LEON4 с архитектурой SPARC v8), с использованием отладочной платы.

ПО предназначено для работы в среде операционной системы Windows.

Взаимодействие персонального компьютера с установленным ПО «CodeMaster-Leon» и микропроцессора осуществляется через интерфейс JTAG.

Содержание

1 Инсталляция ИСР CodeMaster-Leon

В результате инсталляции дистрибутива устанавливаются компоненты ИСР для LEON4, разработанные open source community (Eclipse Mars, набор программных средств, используемых для компиляции, GNU Sparc – Bare-C Cross Compilation, и т.д.), а также плагины к Eclipse, разработанные компанией Cobham Gaisler и компанией Альмакод, которые распространяются свободно.

Для корректной работы ПО необходима установка пакета Visual C++ Redistributable, который можно скачать по ссылке: https://www.microsoft.com/ru-ru/download/details.aspx?id=53840.

Полное руководство по инсталляции ИСР CodeMaster-Leon приведено в «Приложение А. Подробное руководство по инсталляции ИСР».

Краткие рекомендации по первичной инсталляции:

1. В диалоге выбора компонентов выберите «Compact installation»
2. Если на вашем компьютере не установлен MinGW, то к компактной версии инсталляции рекомендуется добавить MSYS 1.0.11 (MSYS – Minimal SYStem – это набор утилит GNU, таких как bash, make, gawk и grep, позволяющих создавать приложения и программы, которые зависят от традиционных инструментов UNIX), являющийся компонентом MinGW. В процессе установки MSYS потребуется разрешить ему выполнить post-install действия (ответить "y"), но после указать, что других версий MinGW или MSYS не установлено (ответить "n").
3. Если планируется производить отладку работы библиотечных функций, то можно добавить набор исходных кодов Newlib-1.13.0, который будет распакован в «C:\opt\src\newlib-1.13.0».

2 Создание проекта в ИСР CodeMaster-Leon

2.1 Выбор рабочей области

После установки, в процессе запуска ИСР, нужно выбрать место, где будет располагаться рабочая область (workspace). Eclipse поддерживает возможность переключения с одной рабочей области на другую (см. рисунок 2.1).

Важно! Для корректной сборки проекта набором программных средств GNU Compiler Collection (GCC) путь к рабочей области не должен содержать пробелов.

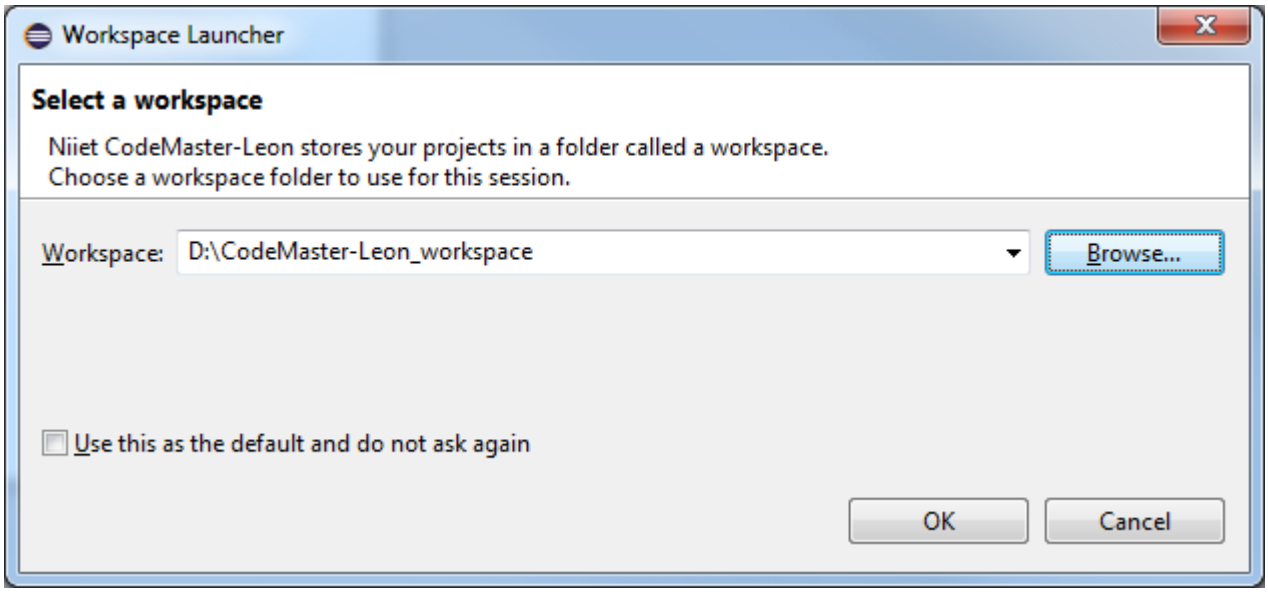


Рисунок 2.1 – Выбор места для расположения рабочей области

2.2 Конфигурирование аппаратной части

После создания рабочей области нужно хотя бы один раз зайти в панель настроек параметров микроконтроллера (Window → Preferences → Chip Selection) и нажать "Apply" или "OK", даже если Вы ничего в настройках не изменяли.

В окне "Chip Preferences" можно указать виды используемых памятей, включить или отключить верификацию, настроить параметры регистров (на высоких частотах для корректной работы необходимо менять настройки Wait State), а также автоматическое определение типа процессора (см. рисунок 2.2). Если значение какого-либо регистра управления контроллерами памяти не должно быть изменено в процессе запуска сессии отладки, соответствующее поле на панели настроек необходимо оставить пустым.

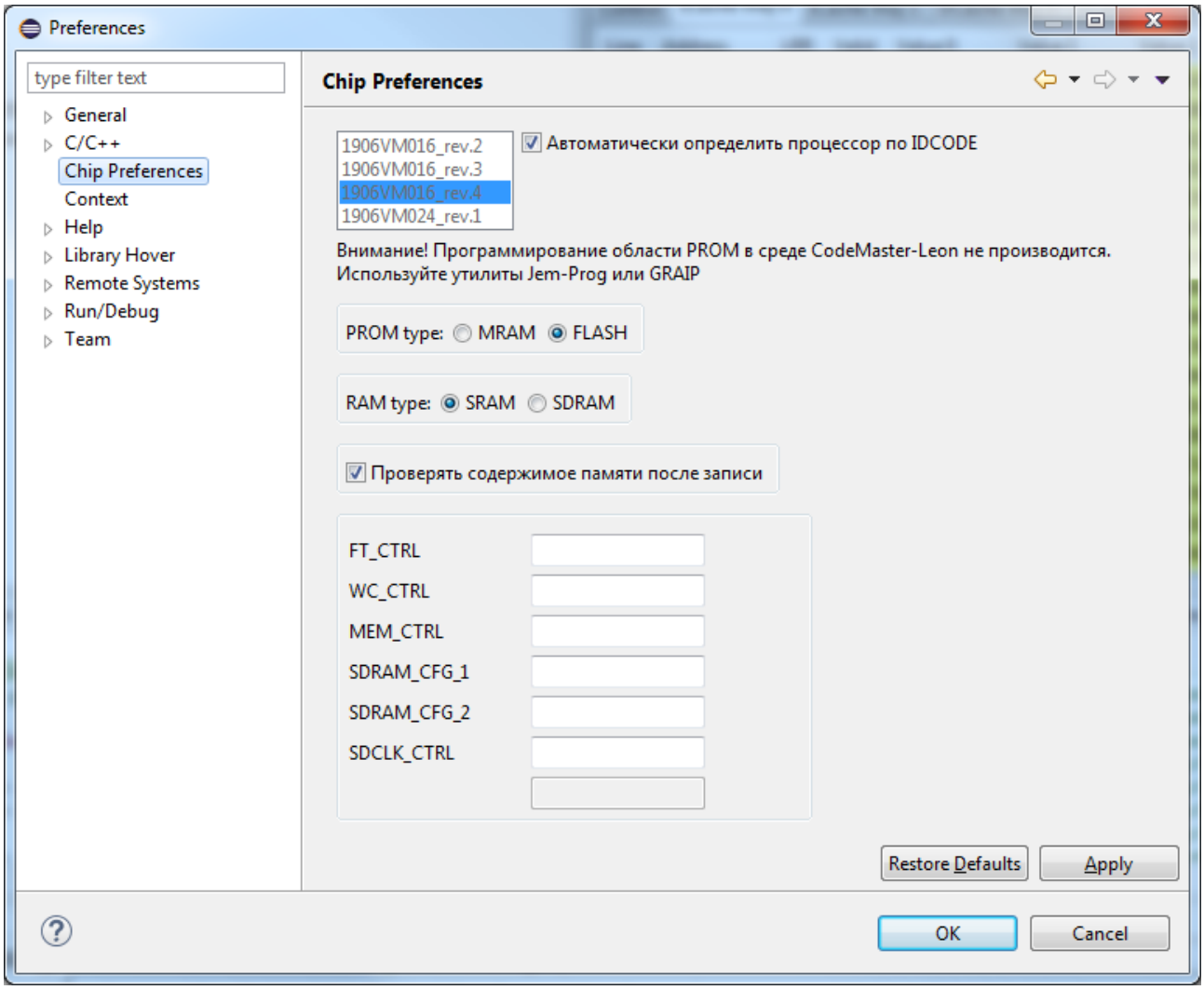


Рисунок 2.2 – Окно "Chip preferences"

2.3 Создание проекта

После того, как ИСР запустится и загрузит рабочую область, можно создавать в ней проекты (File → New → C Project или File → New → C++ Project). На первой закладке мастера создания проекта нужно в качестве набора программных средств (toolchain) выбрать "Cross GCC" (см. рисунок 2.3). На третьей закладке указать в качестве префикса набора программных средств "sparc-elf-" и выбрать путь к папке bin одной из двух версий Bare-C Cross Compilation System (BCC), предпочтительнее более свежую версию 4.4.2 (см. рисунок 2.4). Наборы по умолчанию устанавливаются в каталог "C:\opt", таким образом, путь к BCC версии 4.4.2 будет "C:\opt\sparc-elf-4.4.2-mingw\bin".

Также рекомендуется отключить опцию автоматической пересборки проекта (меню "Project → Build Automatically").

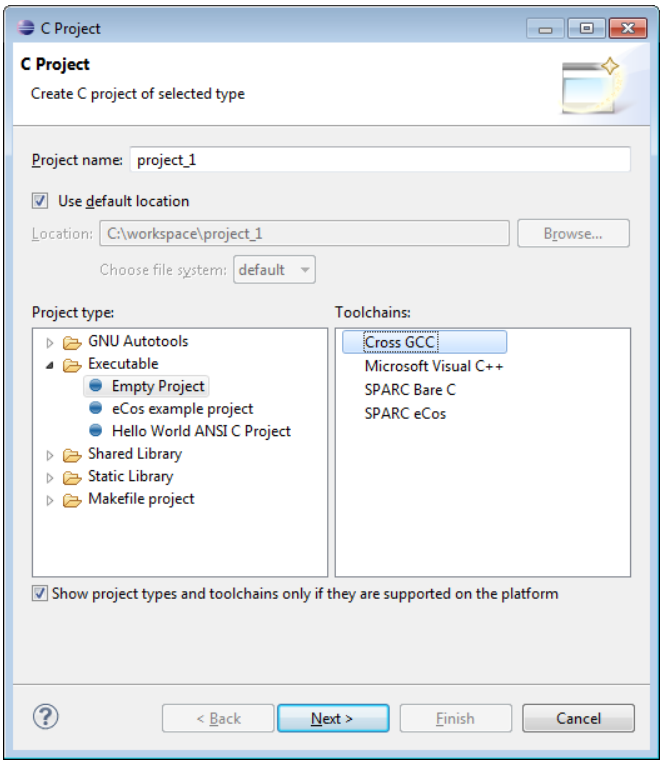


Рисунок 2.3 – Выбор набора программных средств "Cross GCC"

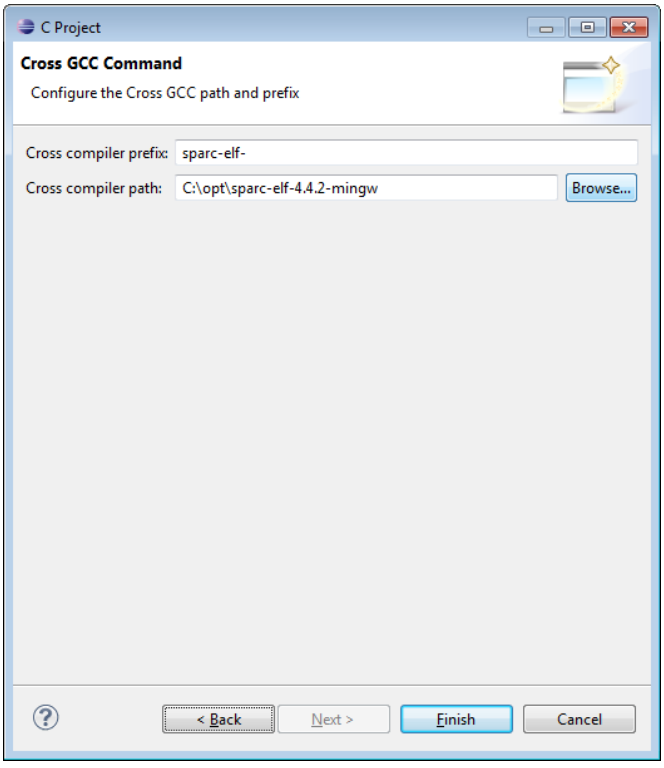


Рисунок 2.4 – Указание пути до компилятора и префикса его комманд

2.4 Низкоуровневая пользовательская инициализация

Для корректного запуска кода проекта необходимо либо скопировать из папки "C:opt\src\CM-Leon\startups\" файл "a_user_initialization.S" низкоуровневой пользовательской инициализации (он требует параметризации и модификации под нужды проекта) в каталог с исходными кодами проекта (обычно это <workspace>/<project_name> или <workspace>/<project_name>/src), либо написать и встроить собственный код инициализации, который бы выполнял сходные функции.

Если не выполнить данный шаг, то будет некорректно установлен указатель вершины стека (вероятно в область, где нет физической памяти), а также возможна некорректная работа кэша процессора 1906ВМ016. Для ускорения выполнения инициализации кэша желательно сконфигурировать контроллер памяти.

Для корректного установления указателя стека (%sp) и указателя кадра (%fp) в ходе инициализации требуется скопировать файл "a_user_initialization.S" (из папки "C:\opt\src\CM-Leon\startups") в папку с исходными кодами проекта (обычно это <workspace>/<project_name> или <workspace>/<project_name>/src). В противном случае значение _FRAMEPOINTER определяется константой _RAM_END из файла линковки (leon4.x). На рисунке 2.5 .

Назначение пользовательского кода инициализации:

– Конфигурация контроллеров памяти;

– Инициализация и активация кэшей;

– Установление указателя стека (%sp) и указателя кадра (%fp);

– Конфигурирование сторожевого таймера

– [при использовании помехоустойчивого кодирования] Инициализация областей памяти, к которым возможны обращения, выполняемые через чтение-модификацию-запись (например, к секции "**.bss**", либо к коду или данным, размещаемым в SDRAM). В общем случае, через заполнение нулевыми значениями всего участка памяти с помощью 64-битных обращений записи.

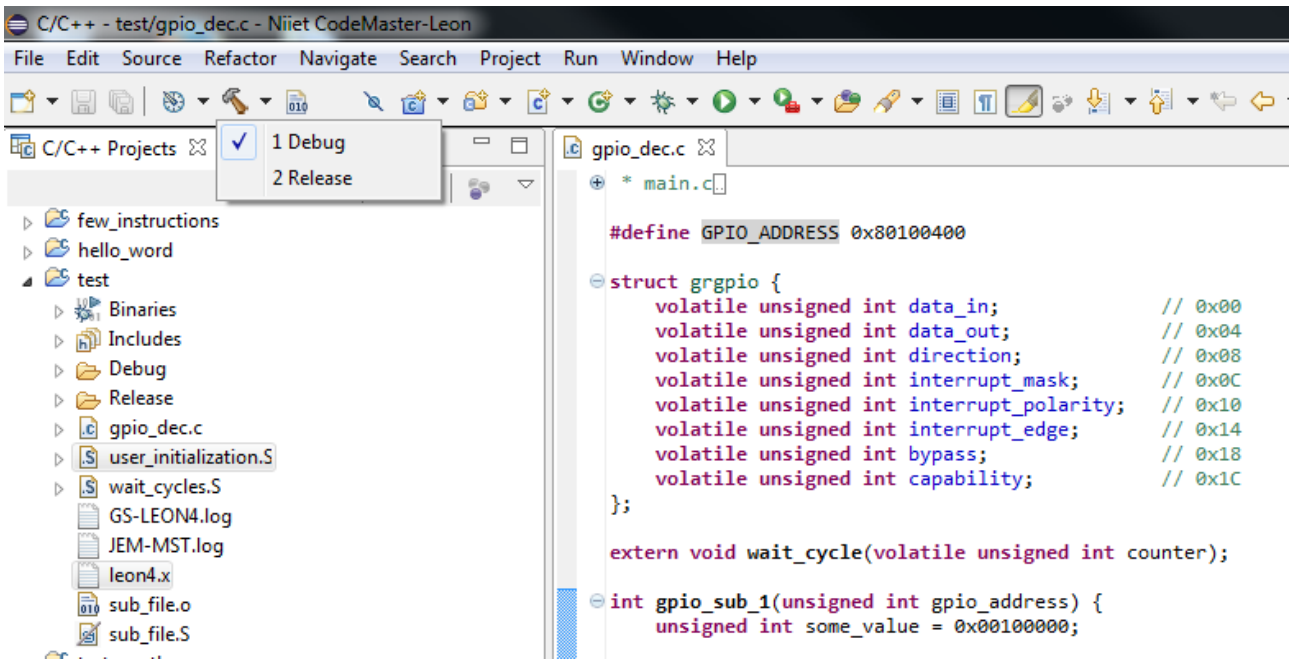


Рисунок 2.5 – Размещение файла низкоуровневой пользовательской инициализации

Размещение и использование кода без указания ключа "-nostartfiles":

Приведенный пример низкоуровневой инициализации использует переопределение слабой функции "_cleanregs_custom_weak" для внедрения пользовательской инициализации на раннем этапе выполнения кода (см. "sparc-elf-4.4.2-mingw\src\libgloss\sparc_leon\" файлы "locore_mvt.S" и "locore.S")

Обычная последовательность выполнения кода (имена меток обрамлены угловыми скобками) начинается с метки <start> (совпадает с <_trap_table> адресом метки). Далее следует прыжок к метке <_hardreset_mvt>, код которой сбрасывает бит SVT регистра %asr17 (не обязательный этап, так как это соответствует значению по умолчанию). Осуществляется прыжок к метке <_hardreset>, где располагаются инструкции:

– вызова функции с именем <_hardreset_custom_weak> (в которой выполняется перезапись констант _leon_version, _nwindows _nwindows_min1 и _nwindows_min2 на основе значений полей регистров %psr и %asr17. Данная функция может использоваться для размещения кода инициализации mkprom);

– установки _trap_table в качестве (TBA);

– вызова функции с именем <_hardreset_custom_svt_weak> (дает mkprom шанс переопределить значение регистра %tbr);

– вызова функции с именем <_cleanregs_custom_weak> / <_cleanregs_libgloss>;

– ...

Размещение и использование кода с указанием ключа "-nostartfiles":

В качестве точки входа будет использоваться метка start из файла "a_user_initialization.S".

2.5 Сборка проекта для активной конфигурации (Debug или Release)

Сборка проекта для активной конфигурации (Debug или Release) производится командой "Project → Build Project", или из контекстного меню командой "Build Project" (стоя на имени проекта в дереве проектов). Настройка параметров проекта производится через меню "Project → Properties" (см. рисунок 2.6), на закладке "C/C++ Build → Settings" (см. рисунок 2.7).

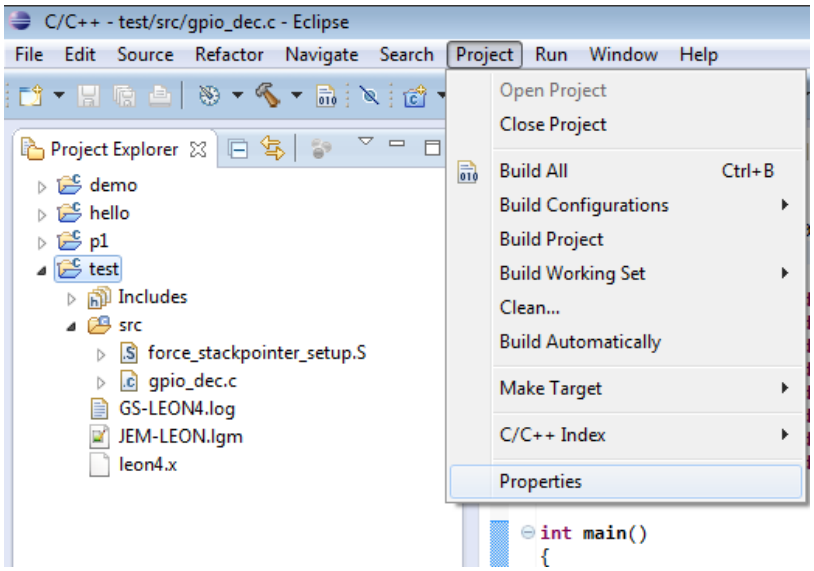


Рисунок 2.6 – Вызов окна настроек текущего проекта

Примечание: Меню "Project → Properties" можно вызвать из контекстного меню, нажав правую кнопку мыши, стоя на имени проекта в дереве проектов, выбрав пункт "Properties".

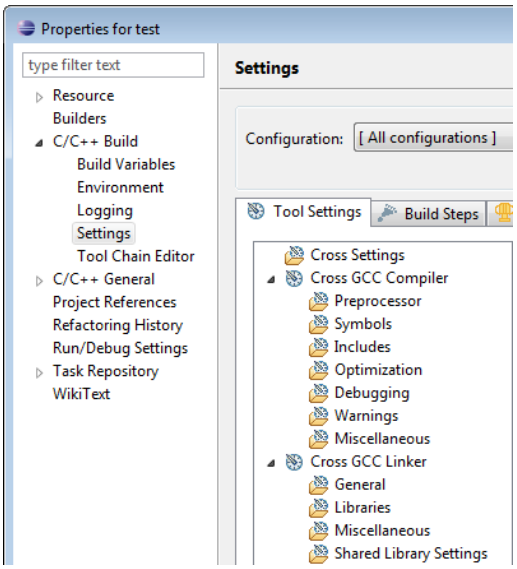


Рисунок 2.7 – Настройка параметров проекта

2.6 Настройка параметров компилятора

В настройках параметров проектов желательно добавить ключ "-mcpu=v8", который позволит компилятору использовать инструкции аппаратного умножения и деления umul/umulcc/smul/smulcc и udiv/udivcc/sdiv/sdivcc (вместо программной реализации), использующие блок аппаратного умножения.

Последовательность действий:

– вызвать окно свойств текущего проекта ("Project → Properties"), стоя на имени проекта в дереве проектов;

– на закладке "C/C++ Build → Settings" в поле " Configuration:" выбрать "All configuration" (см. рисунок 2.8);

– на вкладке "Tool Settings" выбрать закладку "Cross GCC Compiler → Miscellaneous" и в поле "Other options" дописать "-mcpu=v8" (без кавычек), данная опция разрешает компилятору использовать инструкции аппаратного умножения и деления;

– нажать кнопку "Apply", чтобы применить выбранную конфигурацию.

В настройках параметров проектов следует изменить компилятор ассемблерных файлов на gcc (sparc-elf-gcc) с дополнительным ключом "-c".

Последовательность действий:

– вызвать окно свойств текущего проекта ("Project → Properties"), стоя на имени проекта в дереве проектов;

– на закладке "C/C++ Build → Settings" в поле "Configuration:" выбрать "All configuration" (см. рисунок 2.8);

– в закладке "Cross GCC Assembler" в поле "Command:" вместо 'as' ввести 'gcc' (см. рисунок 2.9). Замена компилятор ассемблерных файлов требуется, поскольку 'sparc-elf-as', используемый по умолчанию, не поддерживает C-подобный препроцессинг (использование директив #include, #define и т.д.), без которого будут наблюдаться ошибки при компиляции низкоуровневой пользовательской инициализации, приведенной в примерах;

– на вкладке "Tool Settings" выбрать закладку "Cross GCC Assembler → General" и в поле "Assembler flags:" записать "-c" (без кавычек), данная опция указывает gcc не линковать скомпилированный файл в исполняемый;

– нажать кнопку "Apply", чтобы применить выбранную конфигурацию.

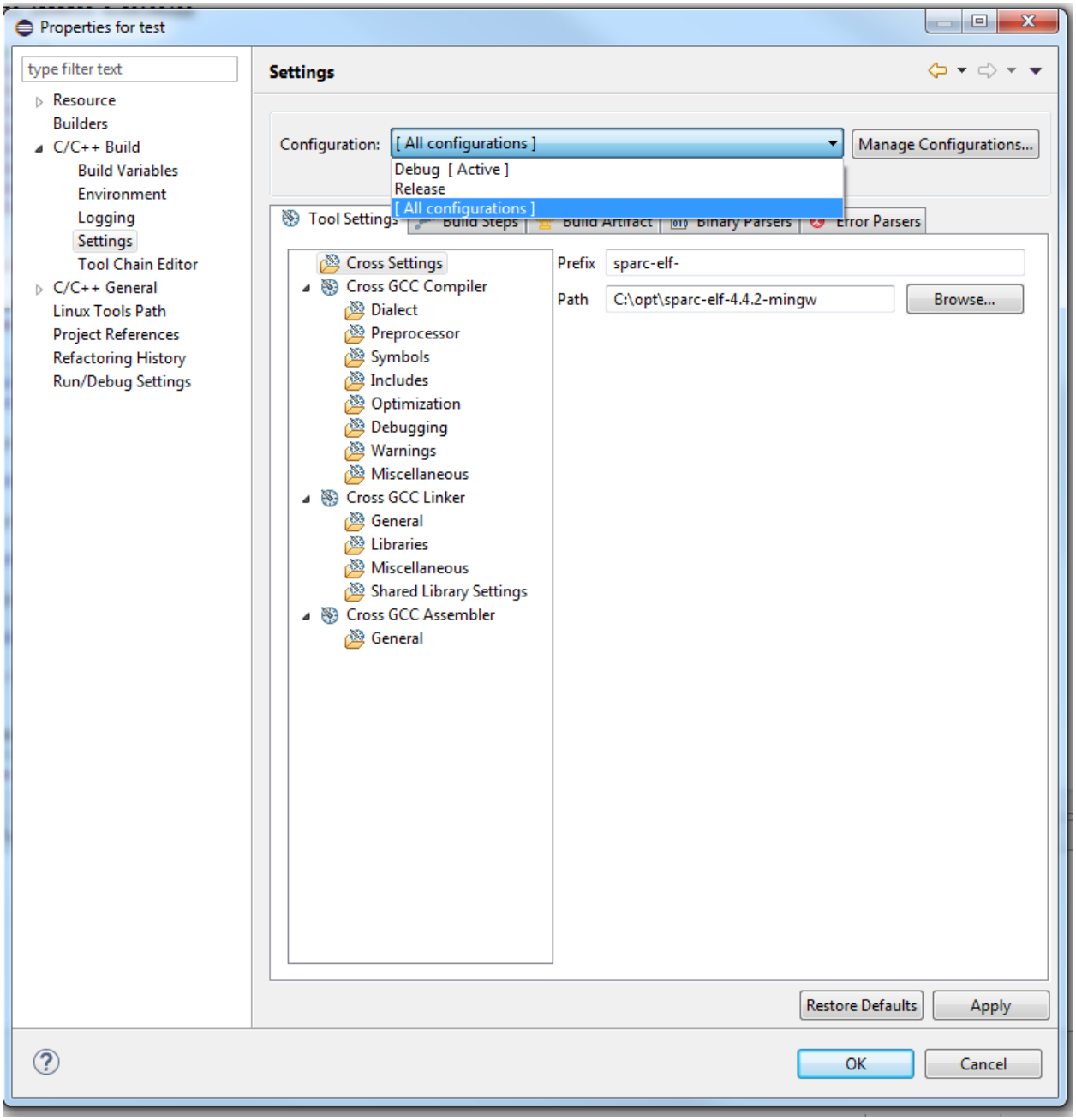


Рисунок 2.8 – Выбор "All configuration"

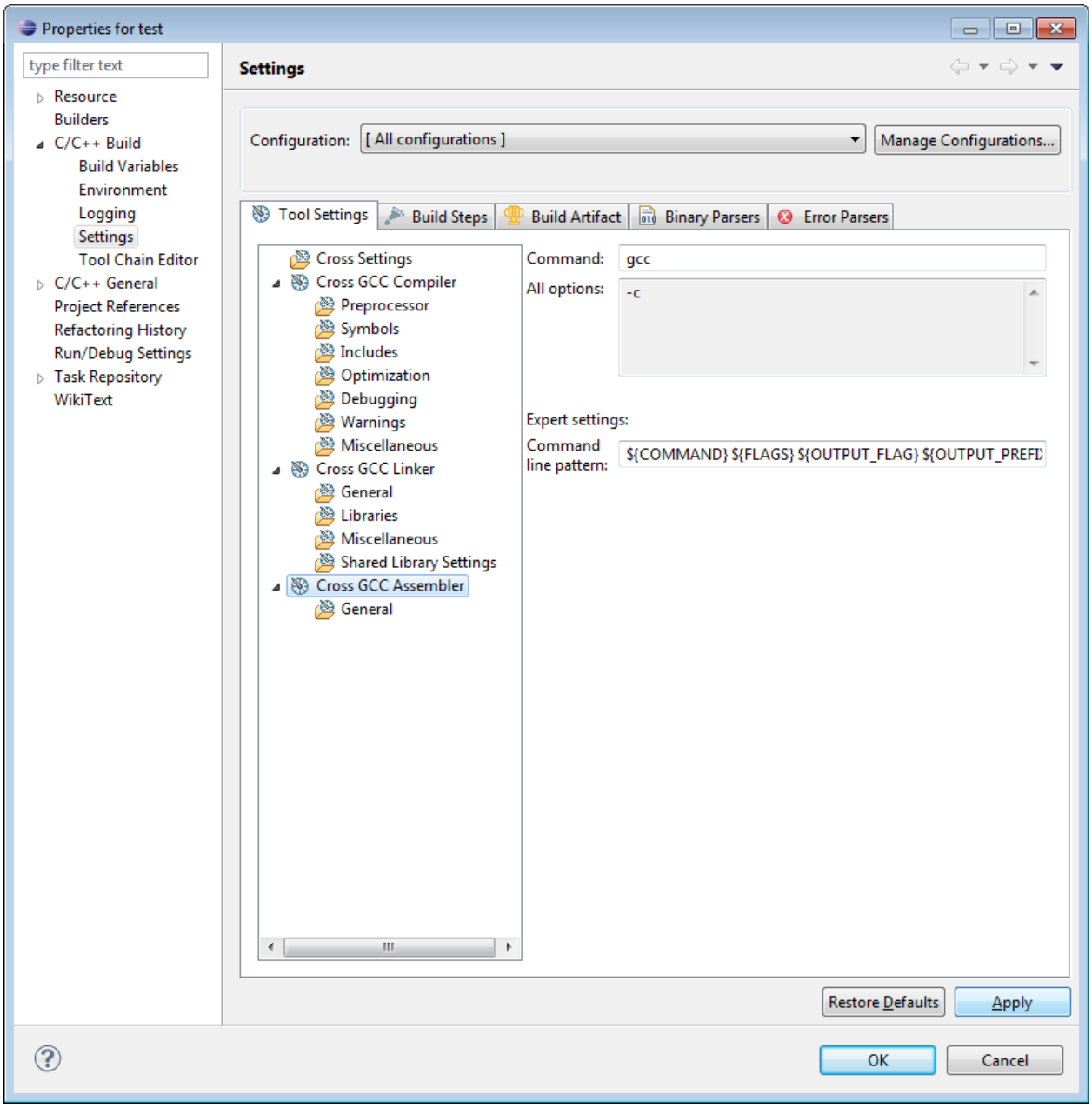


Рисунок 2.9 – Замена компилятор ассемблерных файлов

2.7 Управление размещением кода и данных (опционально)

Если предполагается размещение кода и данных, отличное от размещения по умолчанию (например, в область PROM), нужно скопировать в каталог проекта файл, содержащий заготовку скрипта линкера (один из файлов из каталога "sparc-elf\lib\ldscripts" выбранного тулчейна; настройки по умолчанию содержит файл "sparcleon.x") и подключить его, задав опцию "-T ../<имя файла>"в качестве ключа линкера ("C/C++ Build → Settings → Cross GCC Linker → Miscellaneous → Other option (-Xlinker [option])"). После этого можно вносить в файл скрипта линкера необходимые правки.

Если требуется размещение исполняемого кода или данных по нестандартному адресу (по умолчанию используется 0x40000000), то линковщику следует указать подходящий файл с разметкой. Для этого следует:

– скопировать в каталог проекта (<workspace>/<project_name>) файл, содержащий заготовку скрипта линкера (один из файлов из каталога "sparc-elf\lib\ldscripts" выбранного компилятора; настройки по умолчанию содержит файл "sparcleon.x"). Например, для GCC версии 4.4.2, установленного в директорию по умолчанию, полный путь до файла будет "C:\opt\sparc-elf-4.4.2-mingw\sparc-elf\lib\ldscripts\sparcleon.x";

– внести в файл скрипта линкера необходимые правки (указать действительный размер областей памяти и указать желаемое размещение сегментов кода по областям) В скопированном файле, далее именуемом leon4.x, необходимо изменить параметр SEARCH_DIR ("C:\opt\sparc-elf-4.4.2-mingw\sparc-elf\lib");

– подключить скорректированный файл, задав опцию "-T ../<имя файла>"в качестве ключа линкера ("C/C++ Build → Settings → Cross GCC Linker → Miscellaneous → Other option (-Xlinker [option])"). Для этого следует открыть окно опций проекта (меню "Project → Properties", стоя на имени проекта в дереве проектов). На закладке "C/C++ Build → Settings" в поле "Configuration" следует выбрать "All configuration". На вкладке "Tool Settings" выбрать "Cross GCC Linker → Miscellaneous" и нажать кнопку "Add.." (см. рисунок 2.10).В появившемся диалоговом окне следует ввести "-T ../leon4.x" без кавычек (указывается относительный путь от папки активной конфигурации проекта, часто <workspace>/<project_name>/Debug или <workspace>/<project_name>/Release, до файла разметки <workspace>/<project_name>/leon4.x) (см. рисунок 2.11), после чего нажать кнопку "Apply" (см. рисунок 2.12). После этого можно вносить в файл скрипта линкера необходимые правки.

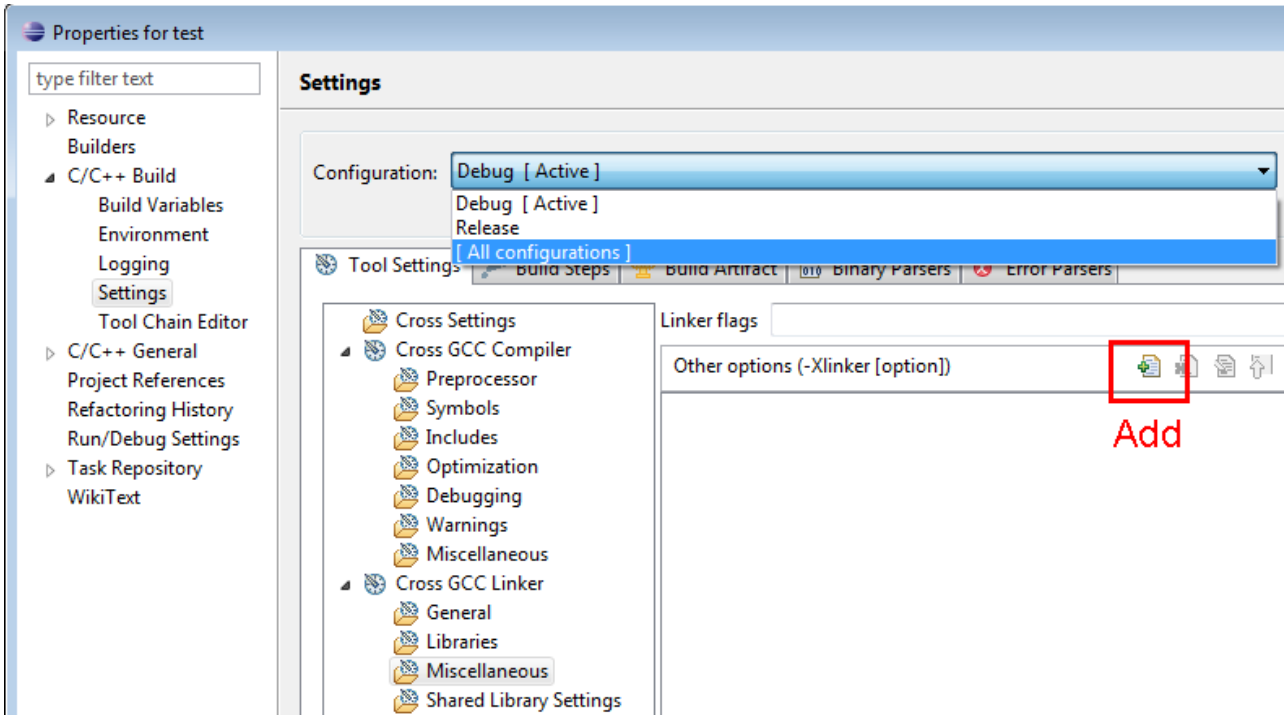


Рисунок 2.10– Добавление ключа для линкера

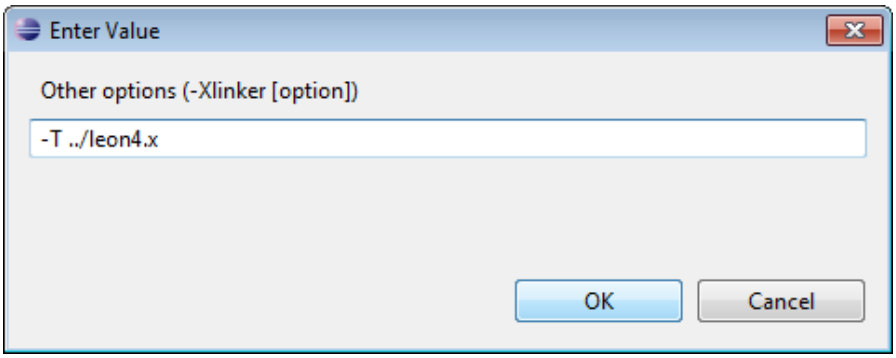


Рисунок 2.11 – Указание пути до скрипта линковки

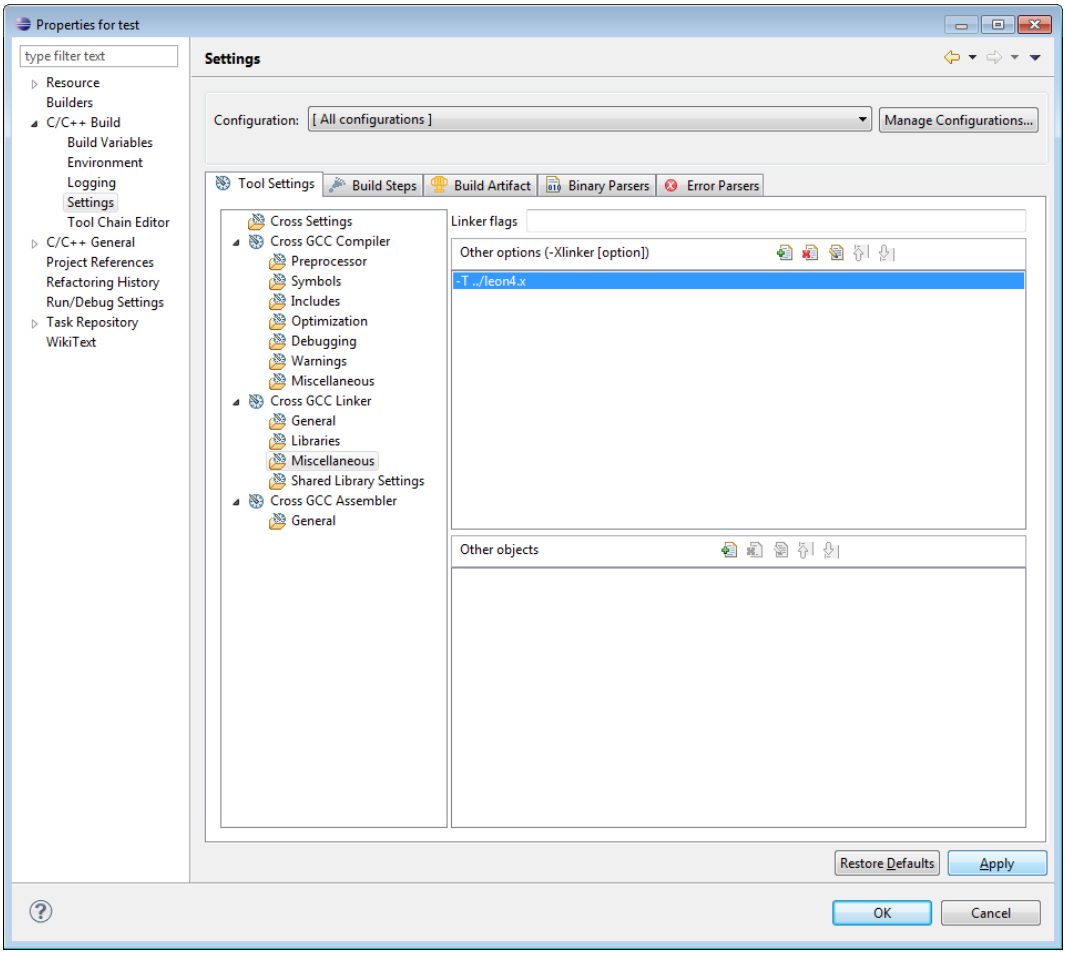


Рисунок 2.12 – Итоговый вид окна конфигурации линкера

2.8 Генерация вспомогательных файлов (опционально)

Файл в формате SREC может потребоваться при использовании ПО «GRAIP» для программирования и отладки (в том числе и для программирования flash).

Последовательность действий для автоматического создания SREC файлов:

– вызвать окно свойств текущего проекта ("Project → Properties"), стоя на имени проекта в дереве проектов;

– на закладке "C/C++ Build → Settings" в поле " Configuration:" выбрать "All configuration" (см. рисунок 2.13);

– перейти на вкладку "Build Steps" в поле "Post-build steps Commands:" добавить следующую строку (без кавычек):

"sparc-elf-objcopy -O srec ${ProjName} ${ProjName}.srec".

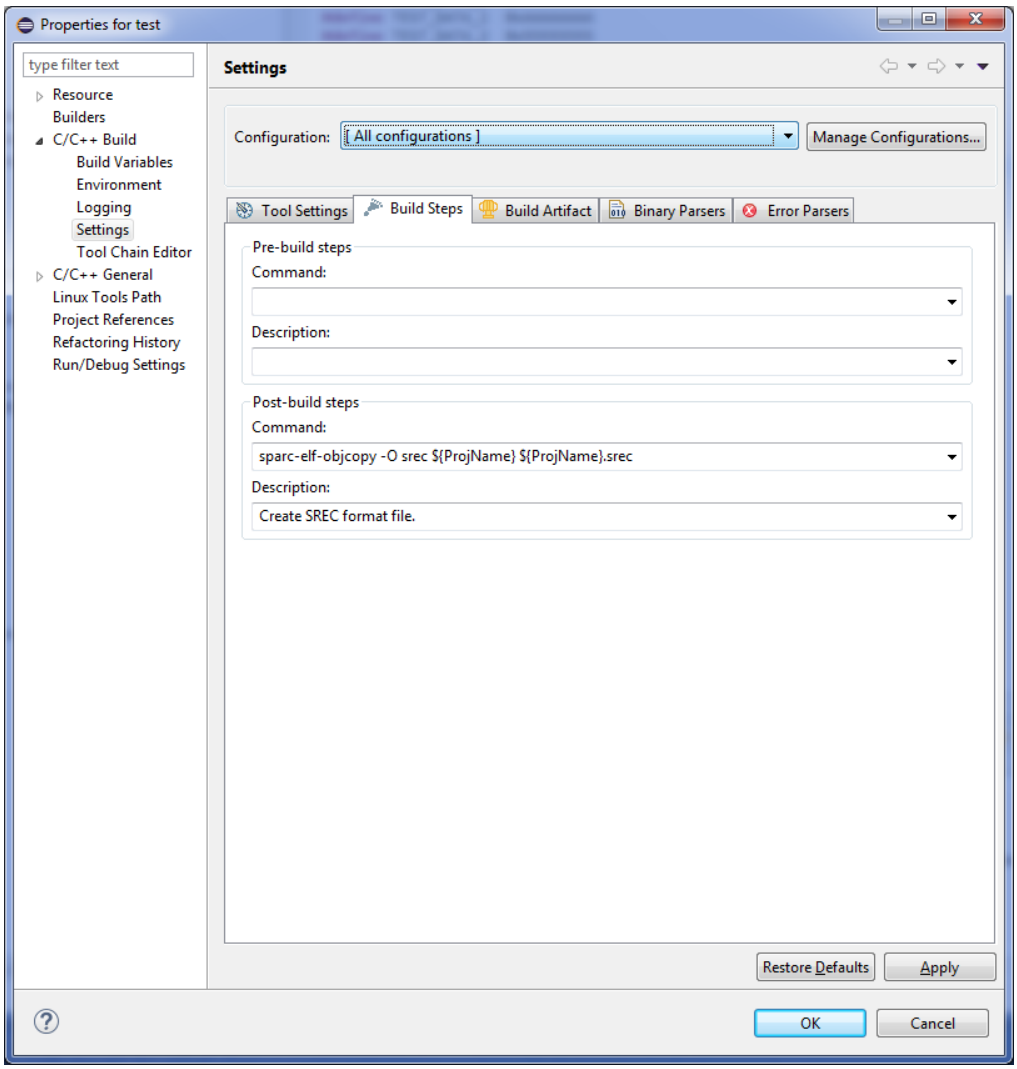


Рисунок 2.13 – Окно Registers

Если помимо файла в формате SREC нужно получить и дамп ассемблерного кода исполняемых секций elf файла, то в поле "Post-build steps Commands:" следует поместить следующую строку (без ковычек):

"sparc-elf-objcopy -O srec ${ProjName} ${ProjName}.srec; sparc-elf-objdump -d ${ProjName} > ${ProjName}.dump".

Замена ключа "-d" на "-D" приведет к выводу всех секций.

Добавление ключа "-S" к вызову "sparc-elf-objdump" сформирует дамп в формате перемежения строк исходного кода с итоговым ассемблерным дампом, если elf-файл включает достаточно отладочной информации (был скомпилирован с ключом "-d" или "-d3").

Описание остальных ключей доступно при вызове "sparc-elf-objdump --help".

3 Отладка проекта в ИСР CodeMaster-Leon

3.1 Настройка отладочной конфигурации

После сборки проекта для его отладки нужно создать отладочную конфигурацию. Конфигурация создается в диалоге "Debug Configurations" (вызываемом через меню "Run → Debug Configurations…"), двойным кликом по шаблону "LEON C/C++ Application" (см. рисунок 3.1). Можно откорректировать название конфигурации; кроме того, на закладке "Debugger", в поле "GDB command file" необходимо выбрать файл "jem.gdbinit" из корневого каталога ИСР (с полным путем). Данный файл содержит команду автоматического запуска утилиты gdbserver (обеспечивающей взаимодействие с аппаратным отладчиком) с необходимыми параметрами. Чтобы изменения вступили в силу, требуется нажать кнопку "Apply" (см. рисунок 3.2). На уровне брандмауэра Windows нужно разрешить программе gdbserver работу по указанному порту (2222). Можно выбрать другой порт в настройках конфигурации отладки; в этом случае нужно также корректировать файл "jem.gdbinit".

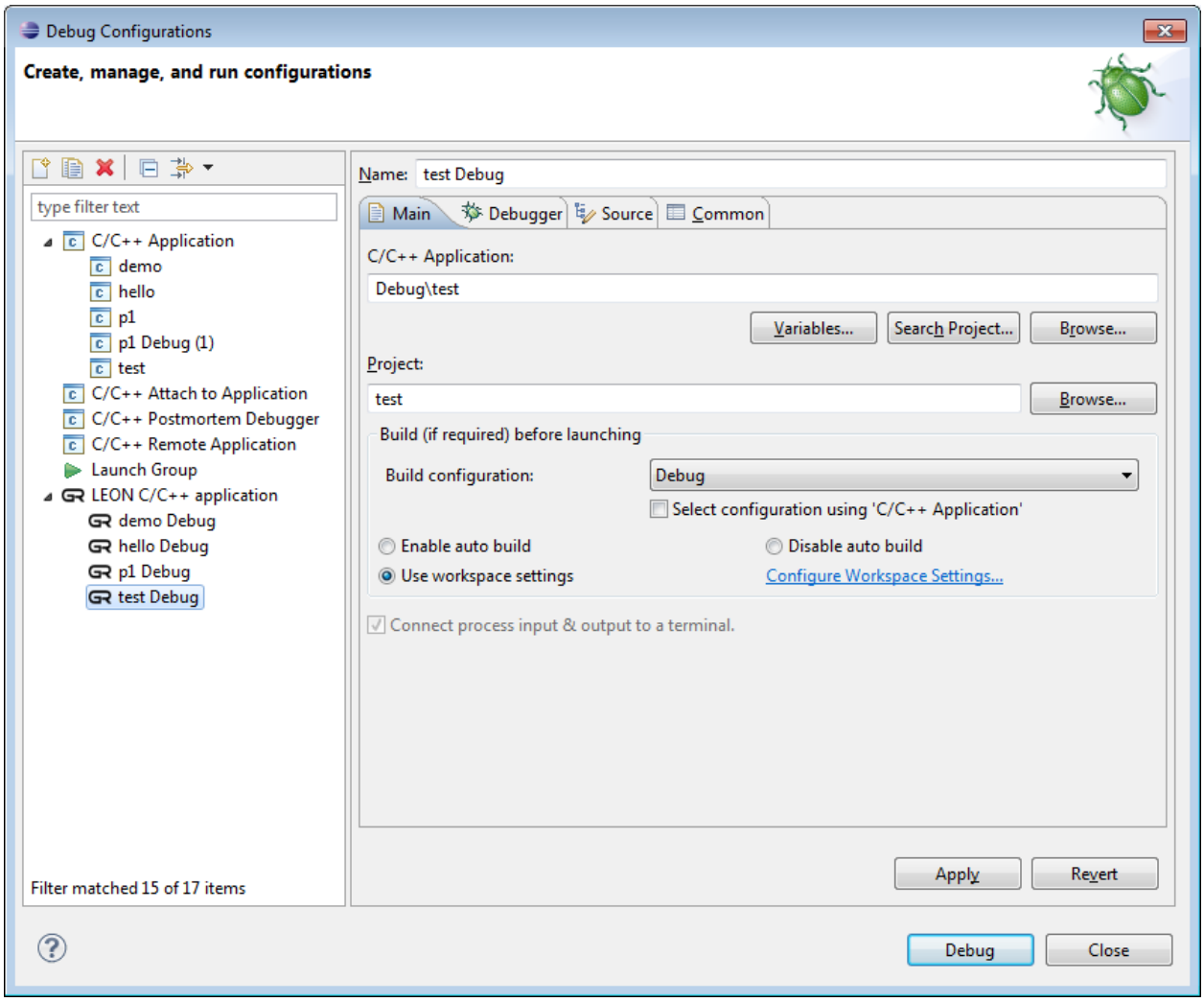


Рисунок 3.1 – Отладочная конфигурация

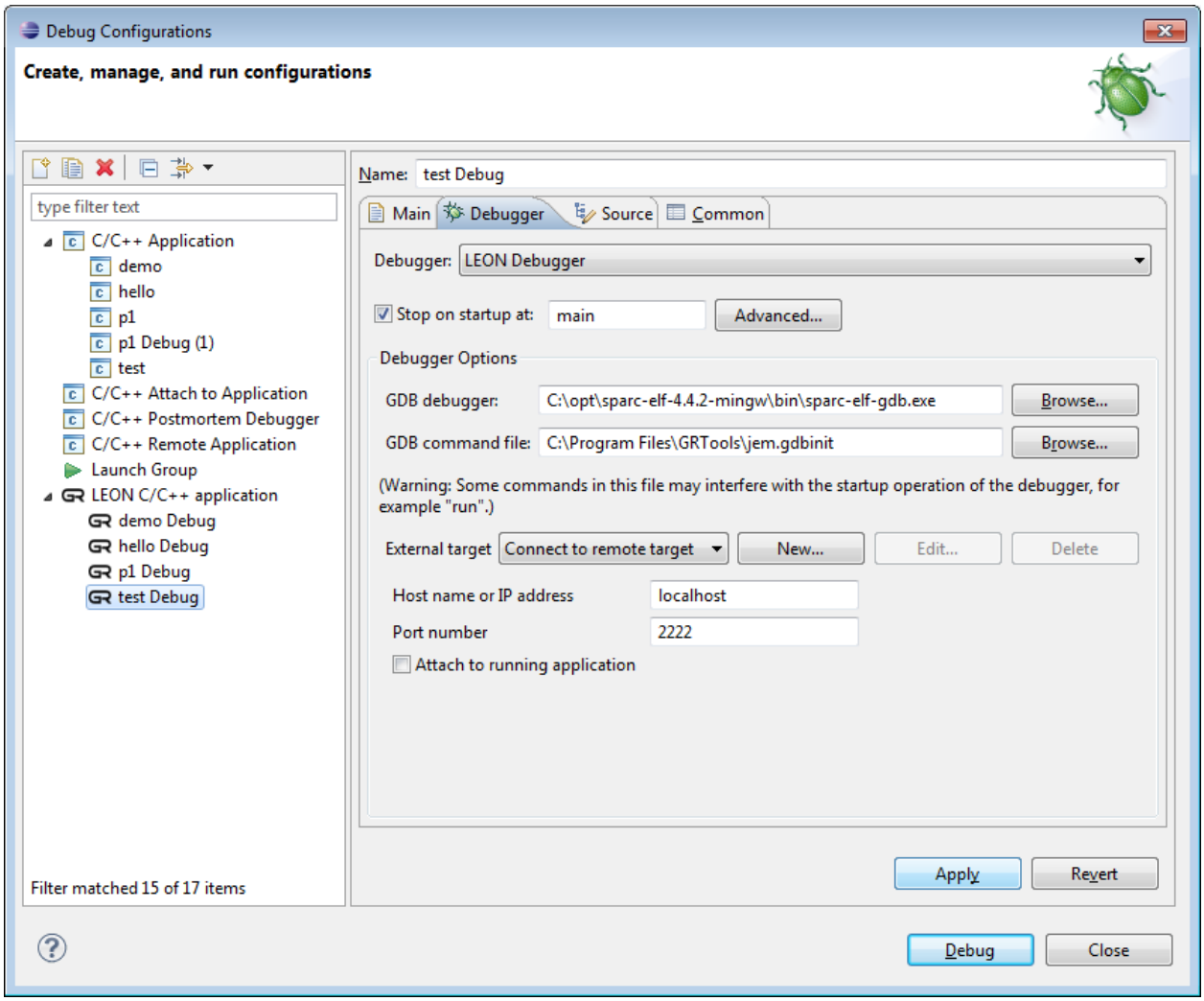


Рисунок 3.2 – Настройки режима отладочной конфигурации

3.2 Указание пути до исходных кодов библиотеки (опционально)

Если есть необходимость отладить работу исходного кода инициализации из стандартной библиотеки (например, чтобы проверить корректность последовательности выполнения низкоуровневой пользовательской инициализации) необходимо указать путь до каталога с исходными файлами (C:\opt\src\newlib-1.13.0). Для этого в закладке "Source" окна "Debug Configuration" необходимо указать путь до подкаталога "src\libgloss\sparc_leon" выбранного набора программных средств. Например, для GCC версии 4.4.2, установленного в директорию по умолчанию, этот путь будет "C:\opt\sparc-elf-4.4.2-mingw\src\libgloss\sparc_leon".

3.3 Режим отладки

Переход в режим отладки (Debug) производится либо из диалога "Debug Configurations" (кнопка "Debug"), упомянутого в подразделе «3.1 Настройка отладочной конфигурации», либо по кнопке Debug основной панели инструментов или из ее выпадающего меню. Второй вариант становится доступен только после первого запуска через окно "Debug Configurations". На вопрос "переключаться ли в

перспективу отладки" рекомендуется ответить положительно (см. рисунок 3.3), также можно установить признак "больше не спрашивать".

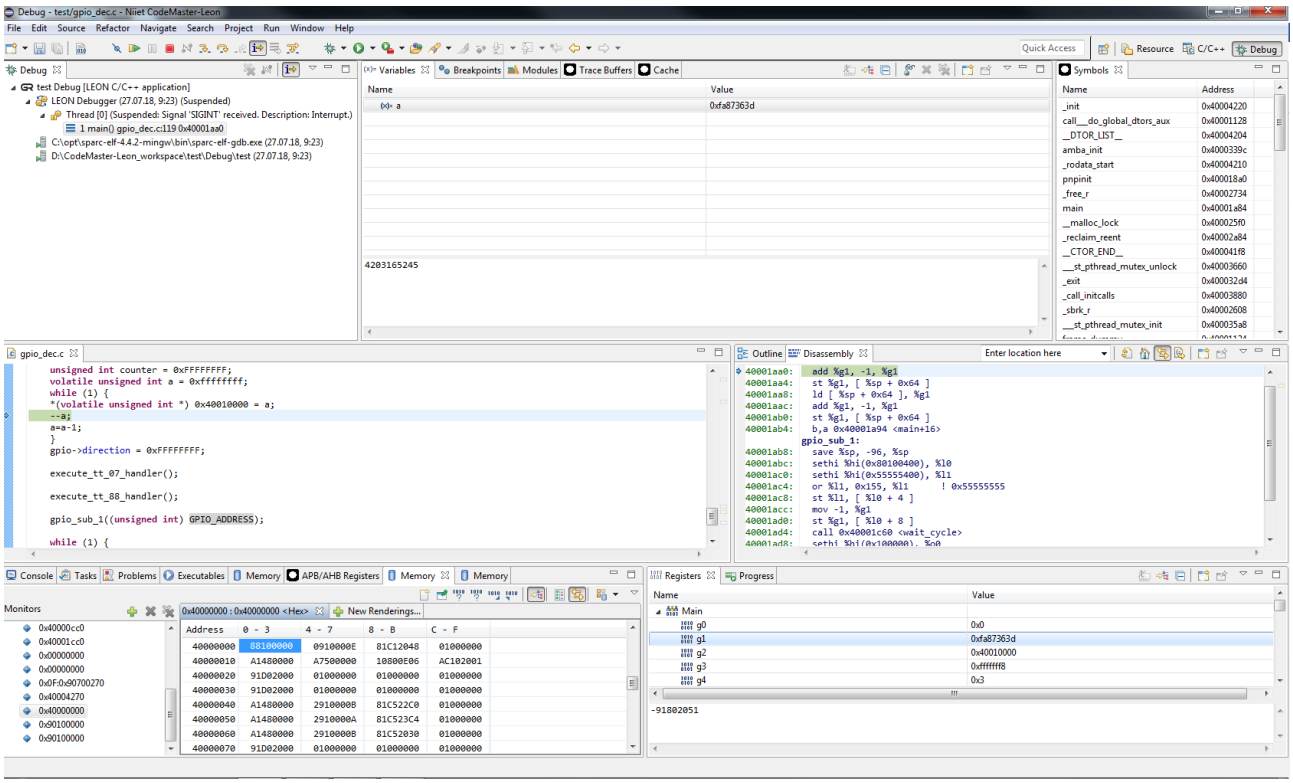


Рисунок 3.3 – Окно, иллюстрирующее работу в режиме отладки

3.4 Окно APB/AHB Registers

В окне APB/AHB Registers можно просмотреть все регистры процессора и их содержимое в реальном времени (см. рисунок 3.4).

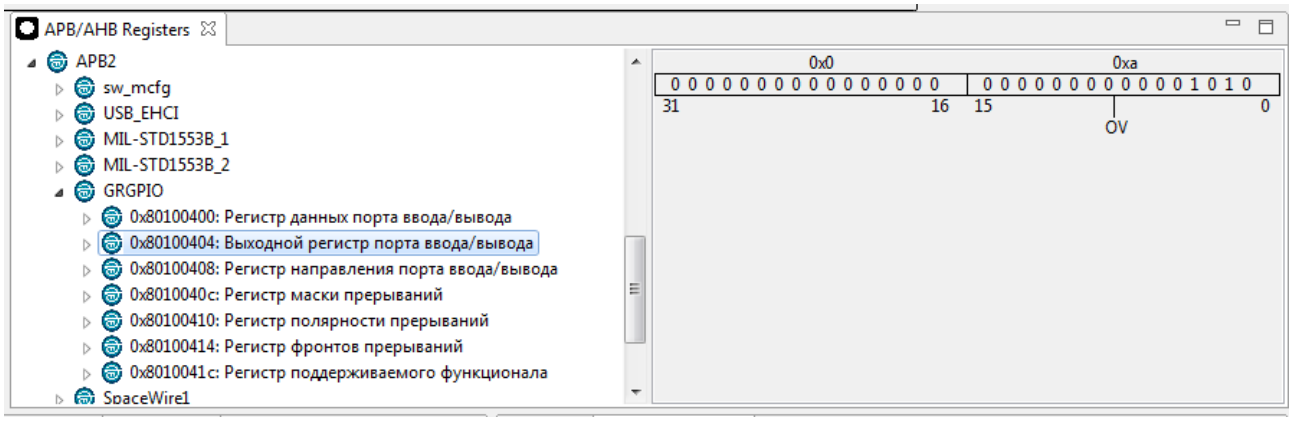


Рисунок 3.4 – Окно APB/AHB Registers

3.5 Окно Trace Buffers

В окне Trace Buffers можно просматривать содержимое буферов трассировки инструкций и AHB в режиме реального времени (см. рисунки 3.5 – 3.7).

Trace Buffers

AHB | Инструкции | Смешанный

Время	Адрес	Инструкция	Результат	Собственный адрес
132657264	0x401FFF18	AHB READ, MST=0, SIZE=64 BIT	[80000100 80100400]	0x90200BE0
132657266	0x40001AA0	AHB READ, MST=0, SIZE=64 BIT	[91D02001 84103FFF]	0x90200E20
132657271	0x80000108	AHB WRITE, MST=0, SIZE=32 BIT	[80000003 80000003]	0x90200C00
132657271	0x40001AA8	AHB READ, MST=0, SIZE=64 BIT	[C4206008 C207BFFC]	0x90200E40
132657276	0x8000010C	AHB WRITE, MST=0, SIZE=32 BIT	[00000035 00000035]	0x90200C20
132657276	0x40001AB0	AHB READ, MST=0, SIZE=64 BIT	[91D02001 C4206004]	0x90200E60
132657409	0x40001AB8	AHB READ, MST=0, SIZE=64 BIT	[1080001A 01000000]	0x90200E80
132657423	0x40001B20	AHB READ, MST=0, SIZE=64 BIT	[C407BFF4 82102001]	0x90200EA0
132657433	0x40001AA0	AHB READ, MST=0, SIZE=64 BIT	[91D02001 84103FFF]	0x90200EC0
132657438	0x40001AA8	AHB READ, MST=0, SIZE=64 BIT	[C4206008 C207BFFC]	0x90200EE0
132657443	0x40001AB0	AHB READ, MST=0, SIZE=64 BIT	[8410200A 91D02001]	0x90200F00
132657448	0x40001AB8	AHB READ, MST=0, SIZE=64 BIT	[1080001A 01000000]	0x90200F20
132657465	0x40001AA0	AHB READ, MST=0, SIZE=64 BIT	[91D02001 84103FFF]	0x90200F40
132657470	0x40001AA8	AHB READ, MST=0, SIZE=64 BIT	[C4206008 C207BFFC]	0x90200F60
132657475	0x40001AB0	AHB READ, MST=0, SIZE=64 BIT	[8410200A C4206004]	0x90200F80
132657480	0x40001AB8	AHB READ, MST=0, SIZE=64 BIT	[91D02001 01000000]	0x90200FA0
132657492	0x40001AC0	AHB READ, MST=0, SIZE=64 BIT	[C407BFF4 C207BFF8]	0x90200FC0
132657497	0x40001AC8	AHB READ, MST=0, SIZE=64 BIT	[C4204000 C407BFF4]	0x90200FE0

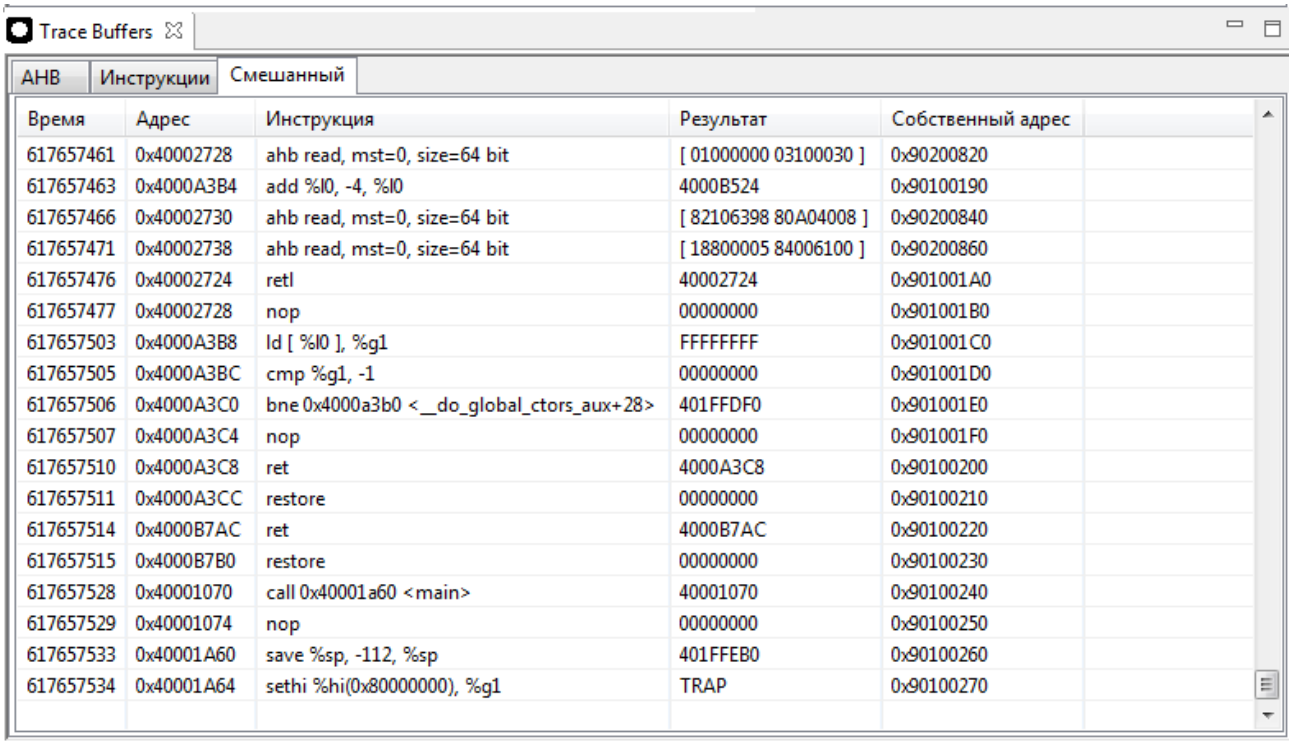
Рисунок 3.5 – Окно Trace Buffers (только обращения по AHB)

Trace Buffers

AHB | Инструкции | Смешанный

Время	Адрес	Инструкция	Результат	Собственный адрес
617657158	0x40001140	cmp %g1, 0	40006318	0x901002E0
617657159	0x40001144	be,a 0x40001164 <frame_dummy+48>	00000000	0x901002F0
617657163	0x4000114C	sethi %hi(0x4000a000), %o0	4000A000	0x90100300
617657186	0x40001150	sethi %hi(0x4000c000), %o1	4000C000	0x90100310
617657188	0x40001154	or %o0, 0x3e0, %o0	4000A3E0	0x90100320
617657213	0x40001158	call 0x40006318 <__register_frame_info>	40001158	0x90100330
617657214	0x4000115C	or %o1, 0x378, %o1	4000C378	0x90100340
617657215	0x40006318	cmp %o0, 0	4000A3E0	0x90100350
617657216	0x4000631C	be 0x4000635c <__register_frame_info+68>	00000000	0x90100360
617657217	0x40006320	nop	00000000	0x90100370
617657231	0x40006324	ld [%o0], %g1	00000010	0x90100380
617657233	0x40006328	cmp %g1, 0	00000010	0x90100390
617657234	0x4000632C	be 0x4000635c <__register_frame_info+68>	00000000	0x901003A0
617657235	0x40006330	sethi %hi(0x1fe00000), %g1	1FE00000	0x901003B0
617657245	0x40006334	st %o0, [%o1 + 0xc]	4000C384	0x901003C0
617657263	0x40006338	st %g1, [%o1 + 0x10]	4000C388	0x901003D0
617657264	0x4000633C	sethi %hi(0x4000cc00), %g1	4000CC00	0x901003E0
617657268	0x40006340	clr [%o1 + 4]	4000C37C	0x901003F0

Рисунок 3.6 – Окно Trace Buffers (только инструкции)



Время	Адрес	Инструкция	Результат	Собственный адрес
617657461	0x40002728	ahb read, mst=0, size=64 bit	[01000000 03100030]	0x90200820
617657463	0x4000A3B4	add %l0, -4, %l0	4000B524	0x90100190
617657466	0x40002730	ahb read, mst=0, size=64 bit	[82106398 80A04008]	0x90200840
617657471	0x40002738	ahb read, mst=0, size=64 bit	[18800005 84006100]	0x90200860
617657476	0x40002724	retl	40002724	0x901001A0
617657477	0x40002728	nop	00000000	0x901001B0
617657503	0x4000A3B8	ld [%l0], %g1	FFFFFFFF	0x901001C0
617657505	0x4000A3BC	cmp %g1, -1	00000000	0x901001D0
617657506	0x4000A3C0	bne 0x4000a3b0 <__do_global_ctors_aux+28>	401FFDF0	0x901001E0
617657507	0x4000A3C4	nop	00000000	0x901001F0
617657510	0x4000A3C8	ret	4000A3C8	0x90100200
617657511	0x4000A3CC	restore	00000000	0x90100210
617657514	0x4000B7AC	ret	4000B7AC	0x90100220
617657515	0x4000B7B0	restore	00000000	0x90100230
617657528	0x40001070	call 0x40001a60 <main>	40001070	0x90100240
617657529	0x40001074	nop	00000000	0x90100250
617657533	0x40001A60	save %sp, -112, %sp	401FFEB0	0x90100260
617657534	0x40001A64	sethi %hi(0x80000000), %g1	TRAP	0x90100270

Рисунок 3.7 – Окно Trace Buffers (смешанный вид)

3.6 Окно Cache

В окне Cache можно просматривать, копировать и изменять содержимое кэша в режиме реального времени (см. рисунки 3.9 – 3.12), а также включать и отключать, верифицировать, записывать значения в регистры кэша инструкций и данных (см. рисунок 3.8).

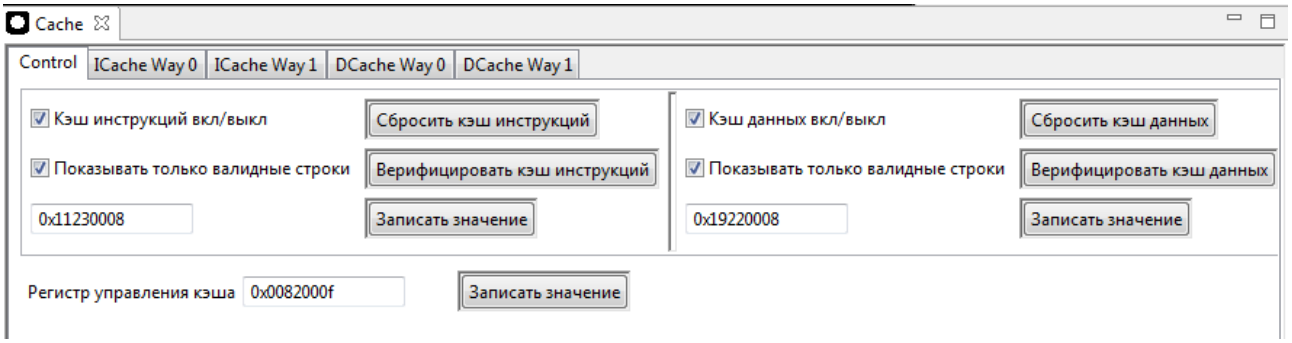
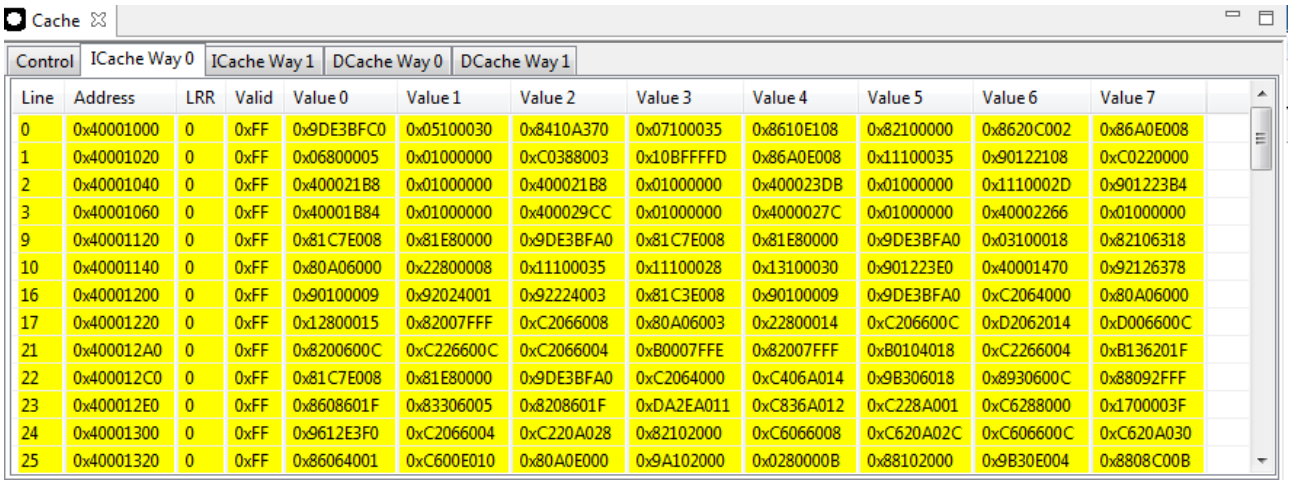
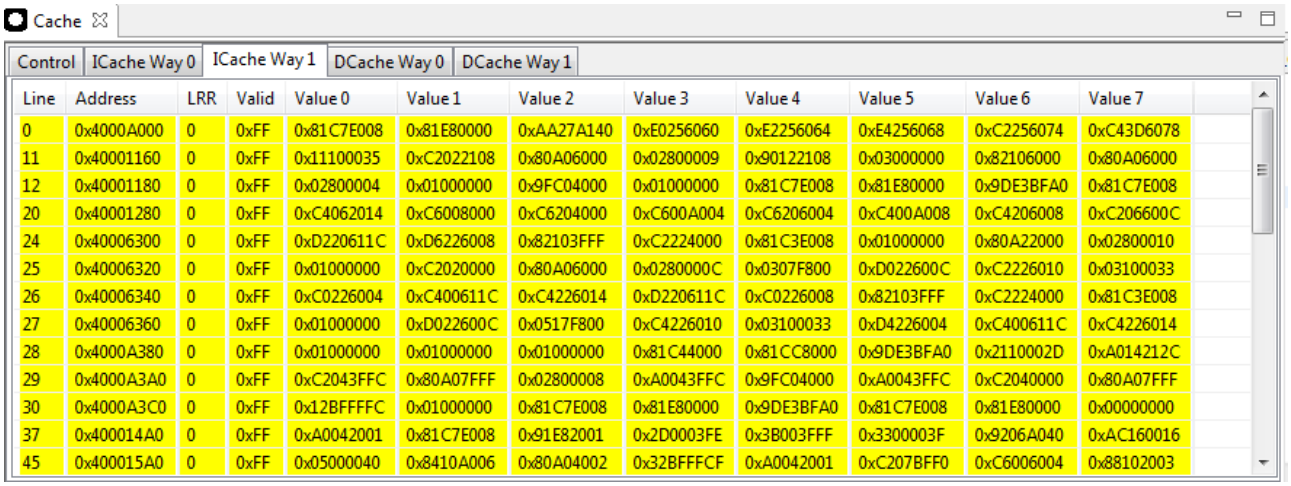


Рисунок 3.8 – Окно управления кэшем



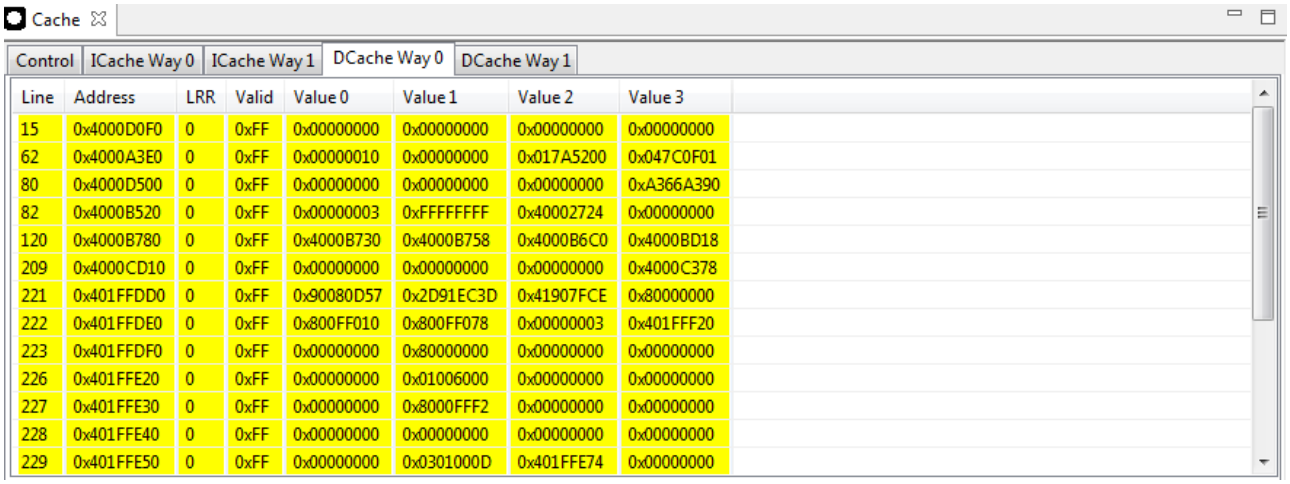
Line	Address	LRR	Valid	Value 0	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7
0	0x40001000	0	0xFF	0x9DE3BFC0	0x05100030	0x8410A370	0x07100035	0x8610E108	0x82100000	0x8620C002	0x86A0E008
1	0x40001020	0	0xFF	0x06800005	0x01000000	0xC0388003	0x10BFFFFD	0x86A0E008	0x11100035	0x90122108	0xC0220000
2	0x40001040	0	0xFF	0x400021B8	0x01000000	0x400021B8	0x01000000	0x400023DB	0x01000000	0x1110002D	0x901223B4
3	0x40001060	0	0xFF	0x40001B84	0x01000000	0x400029CC	0x01000000	0x4000027C	0x01000000	0x40002266	0x01000000
9	0x40001120	0	0xFF	0x81C7E008	0x81E80000	0x9DE3BFA0	0x81C7E008	0x81E80000	0x9DE3BFA0	0x03100018	0x82106318
10	0x40001140	0	0xFF	0x80A06000	0x22800008	0x11100035	0x11100028	0x13100030	0x901223E0	0x40001470	0x92126378
16	0x40001200	0	0xFF	0x90100009	0x92024001	0x92224003	0x81C3E008	0x90100009	0x9DE3BFA0	0xC2064000	0x80A06000
17	0x40001220	0	0xFF	0x12800015	0x82007FFF	0xC2066008	0x80A06003	0x22800014	0xC206600C	0xD2062014	0xD006600C
21	0x400012A0	0	0xFF	0x8200600C	0xC226600C	0xC2066004	0xB0007FFE	0x82007FFF	0xB0104018	0xC2266004	0xB136201F
22	0x400012C0	0	0xFF	0x81C7E008	0x81E80000	0x9DE3BFA0	0xC2064000	0xC406A014	0x9B306018	0x8930600C	0x88092FFF
23	0x400012E0	0	0xFF	0x8608601F	0x83306005	0x8208601F	0xDA2EA011	0xC836A012	0xC228A001	0xC6288000	0x1700003F
24	0x40001300	0	0xFF	0x9612E3F0	0xC2066004	0xC220A028	0x82102000	0xC6066008	0xC620A02C	0xC606600C	0xC620A030
25	0x40001320	0	0xFF	0x86064001	0xC600E010	0x80A0E000	0x9A102000	0x0280000B	0x88102000	0x9B30E004	0x8808C00B

Рисунок 3.9 – Окно содержимого кэша инструкций (канал 0)



Line	Address	LRR	Valid	Value 0	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7
0	0x4000A000	0	0xFF	0x81C7E008	0x81E80000	0xAA27A140	0xE0256060	0xE2256064	0xE4256068	0xC2256074	0xC43D6078
11	0x40001160	0	0xFF	0x11100035	0xC2022108	0x80A06000	0x02800009	0x90122108	0x03000000	0x82106000	0x80A06000
12	0x40001180	0	0xFF	0x02800004	0x01000000	0x9FC04000	0x01000000	0x81C7E008	0x81E80000	0x9DE3BFA0	0x81C7E008
20	0x40001280	0	0xFF	0xC4062014	0xC6008000	0xC6204000	0xC600A004	0xC6206004	0xC400A008	0xC4206008	0xC206600C
24	0x40006300	0	0xFF	0xD220611C	0xD6226008	0x82103FFF	0xC2224000	0x81C3E008	0x01000000	0x80A22000	0x02800010
25	0x40006320	0	0xFF	0x01000000	0xC2020000	0x80A06000	0x0280000C	0x0307F800	0xD022600C	0xC2226010	0x03100033
26	0x40006340	0	0xFF	0xC0226004	0xC400611C	0xC4226014	0xD220611C	0xC0226008	0x82103FFF	0xC2224000	0x81C3E008
27	0x40006360	0	0xFF	0x01000000	0xD022600C	0x0517F800	0xC4226010	0x03100033	0xD4226004	0xC400611C	0xC4226014
28	0x4000A380	0	0xFF	0x01000000	0x01000000	0x01000000	0x81C44000	0x81CC8000	0x9DE3BFA0	0x2110002D	0xA014212C
29	0x4000A3A0	0	0xFF	0xC2043FFC	0x80A07FFF	0x02800008	0xA0043FFC	0x9FC04000	0xA0043FFC	0xC2040000	0x80A07FFF
30	0x4000A3C0	0	0xFF	0x12BFFFFC	0x01000000	0x81C7E008	0x81E80000	0x9DE3BFA0	0x81C7E008	0x81E80000	0x00000000
37	0x400014A0	0	0xFF	0xA0042001	0x81C7E008	0x91E82001	0x2D0003FE	0x3B003FFF	0x3300003F	0x9206A040	0xAC160016
45	0x400015A0	0	0xFF	0x05000040	0x8410A006	0x80A04002	0x32BFFFCF	0xA0042001	0xC207BFF0	0xC6006004	0x88102003

Рисунок 3.10 – Окно содержимого кэша инструкций (канал 1)



Line	Address	LRR	Valid	Value 0	Value 1	Value 2	Value 3
15	0x4000D0F0	0	0xFF	0x00000000	0x00000000	0x00000000	0x00000000
62	0x4000A3E0	0	0xFF	0x00000010	0x00000000	0x017A5200	0x047C0F01
80	0x4000D500	0	0xFF	0x00000000	0x00000000	0x00000000	0xA366A390
82	0x4000B520	0	0xFF	0x00000003	0xFFFFFFFF	0x40002724	0x00000000
120	0x4000B780	0	0xFF	0x4000B730	0x4000B758	0x4000B6C0	0x4000BD18
209	0x4000CD10	0	0xFF	0x00000000	0x00000000	0x00000000	0x4000C378
221	0x401FFDD0	0	0xFF	0x90080D57	0x2D91EC3D	0x41907FCE	0x80000000
222	0x401FFDE0	0	0xFF	0x800FF010	0x800FF078	0x00000003	0x401FFF20
223	0x401FFDF0	0	0xFF	0x00000000	0x80000000	0x00000000	0x00000000
226	0x401FFE20	0	0xFF	0x00000000	0x01006000	0x00000000	0x00000000
227	0x401FFE30	0	0xFF	0x00000000	0x8000FFF2	0x00000000	0x00000000
228	0x401FFE40	0	0xFF	0x00000000	0x00000000	0x00000000	0x00000000
229	0x401FFE50	0	0xFF	0x00000000	0x0301000D	0x401FFE74	0x00000000

Рисунок 3.11 – Окно содержимого кэша данных (канал 0)

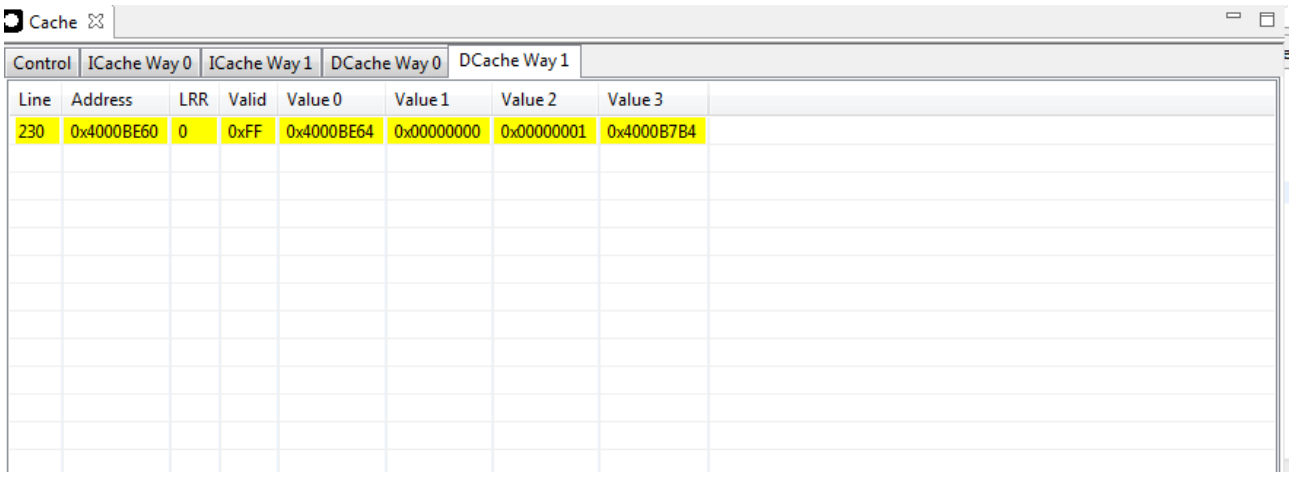


Рисунок 3.12 – Окно содержимого кэша данных (канал 1)

3.7 Окно Symbols

В окне Symbols можно просматривать таблицу меток исполняемого файла (см. рисунок 3.13).

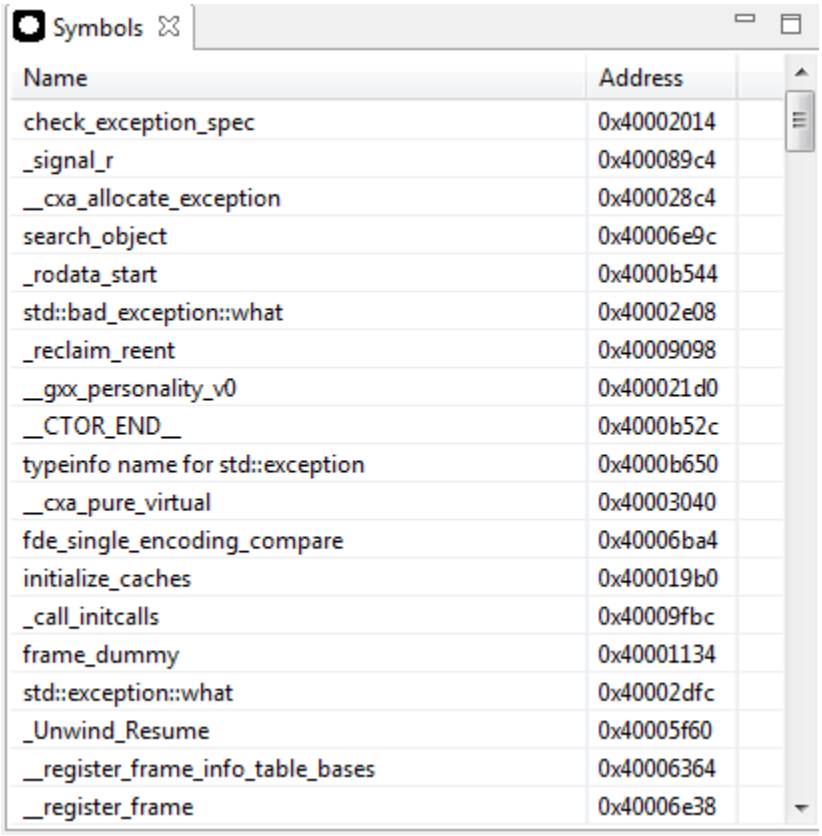


Рисунок 3.13 – Окно Symbols

3.8 Окно Memory и Memory Browser

В окне Memory можно просматривать содержимое любых областей процессора в различных форматах (см. рисунок 3.14), а также копировать и изменять данное содержимое (см. рисунок 3.15).

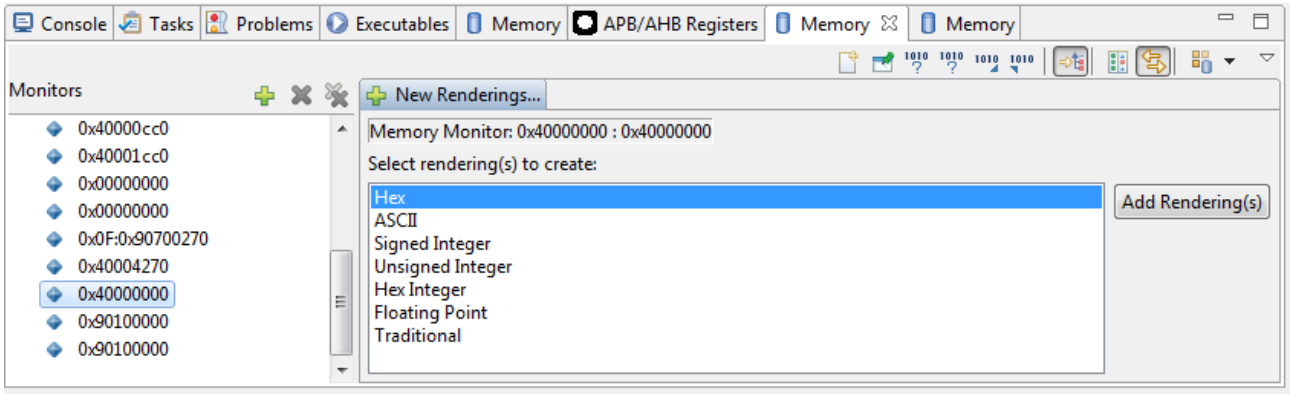


Рисунок 3.14 – Выбор формата отображения содержимого памяти

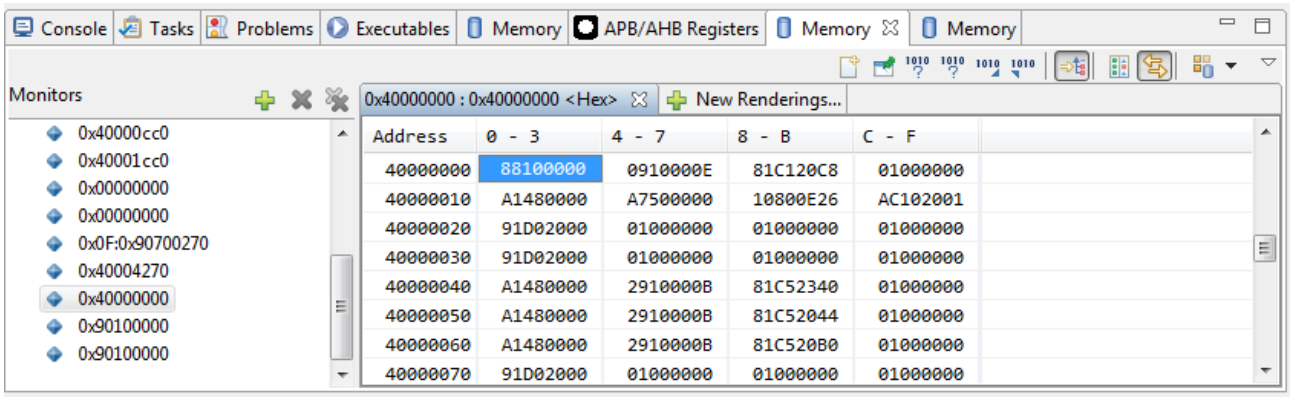


Рисунок 3.15 – Содержимое памяти по адресу 0x40000000

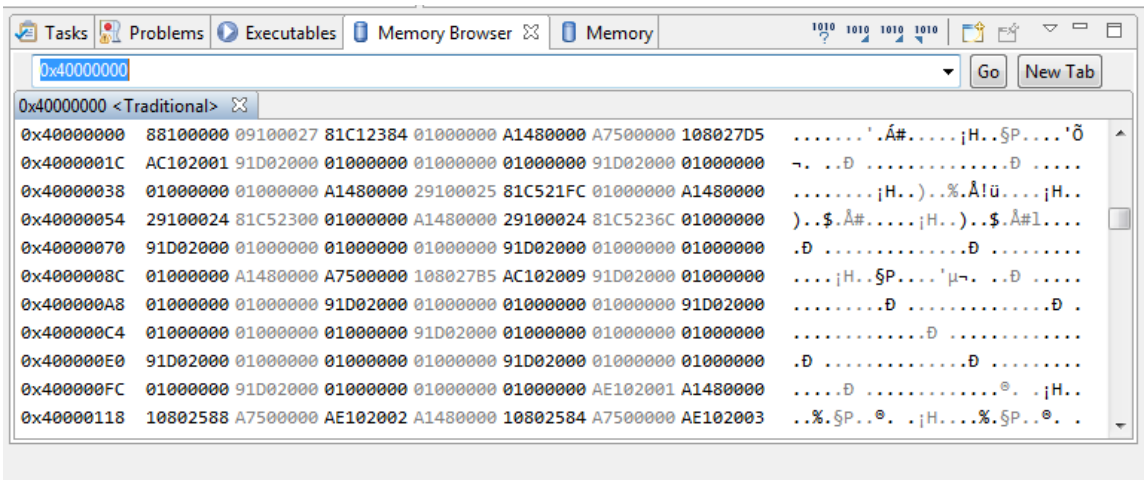


Рисунок 3.16 – Содержимое памяти по адресу 0x40000000 в окне Memory Browser.

3.9 Окно Breakpoints

В окне Breakpoints можно просматривать все установленные точки останова (см. рисунок 3.17).

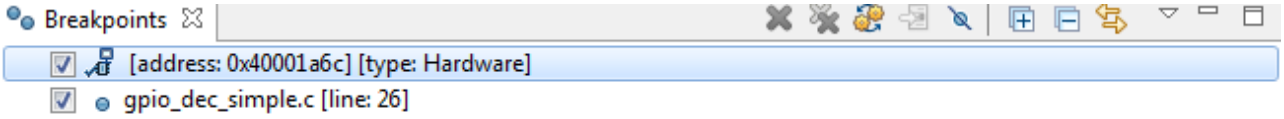


Рисунок 3.17 – Окно Breakpoints

3.10 Окно Variables

В окне Variables можно просматривать содержимое переменной в режиме реального времени (см. рисунок 3.18).

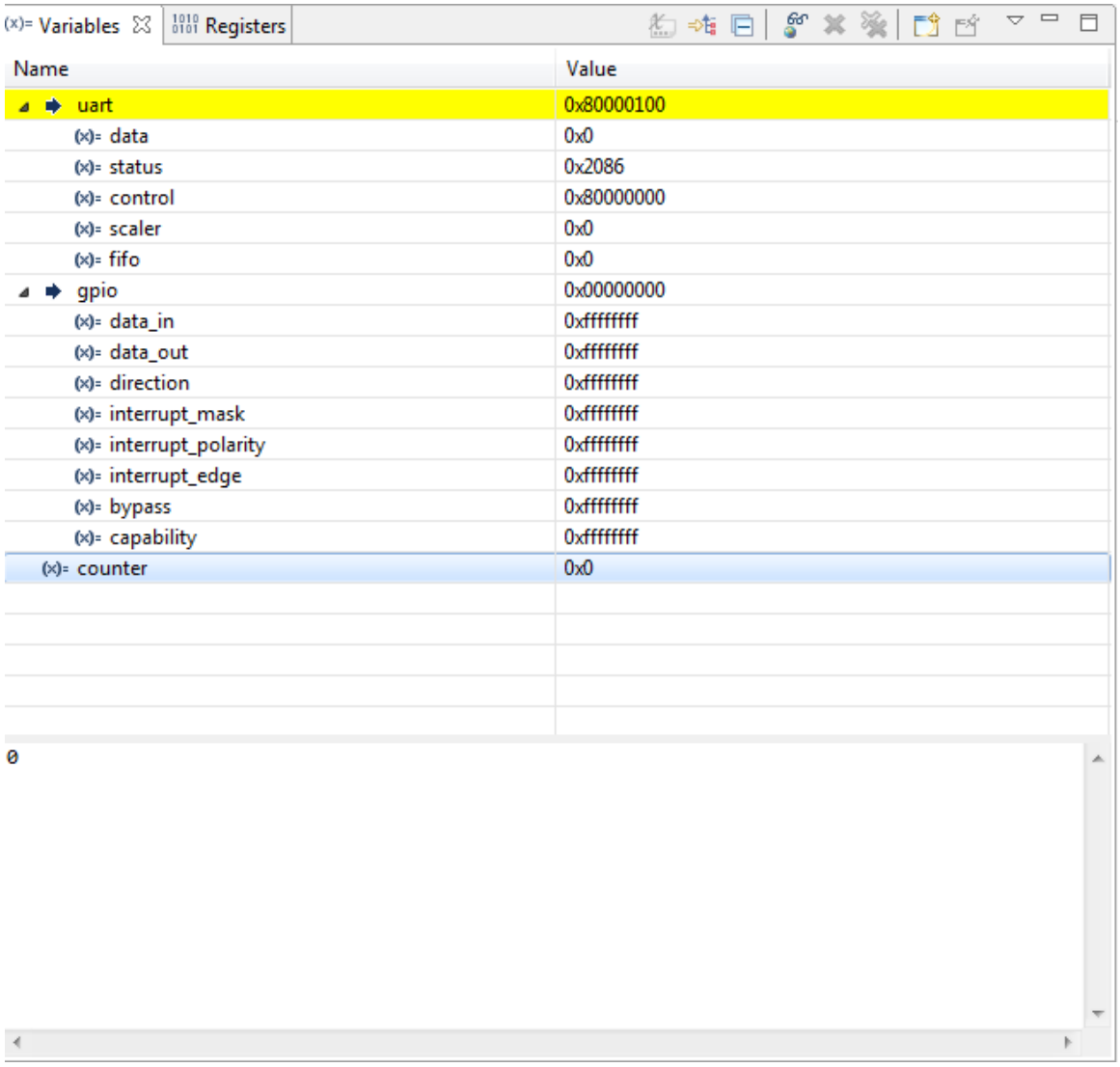


Рисунок 3.18 – Окно Variables

3.11 Окно Registers

В окне Registers можно просматривать содержимое регистров в режиме реального времени (см. рисунок 3.18).

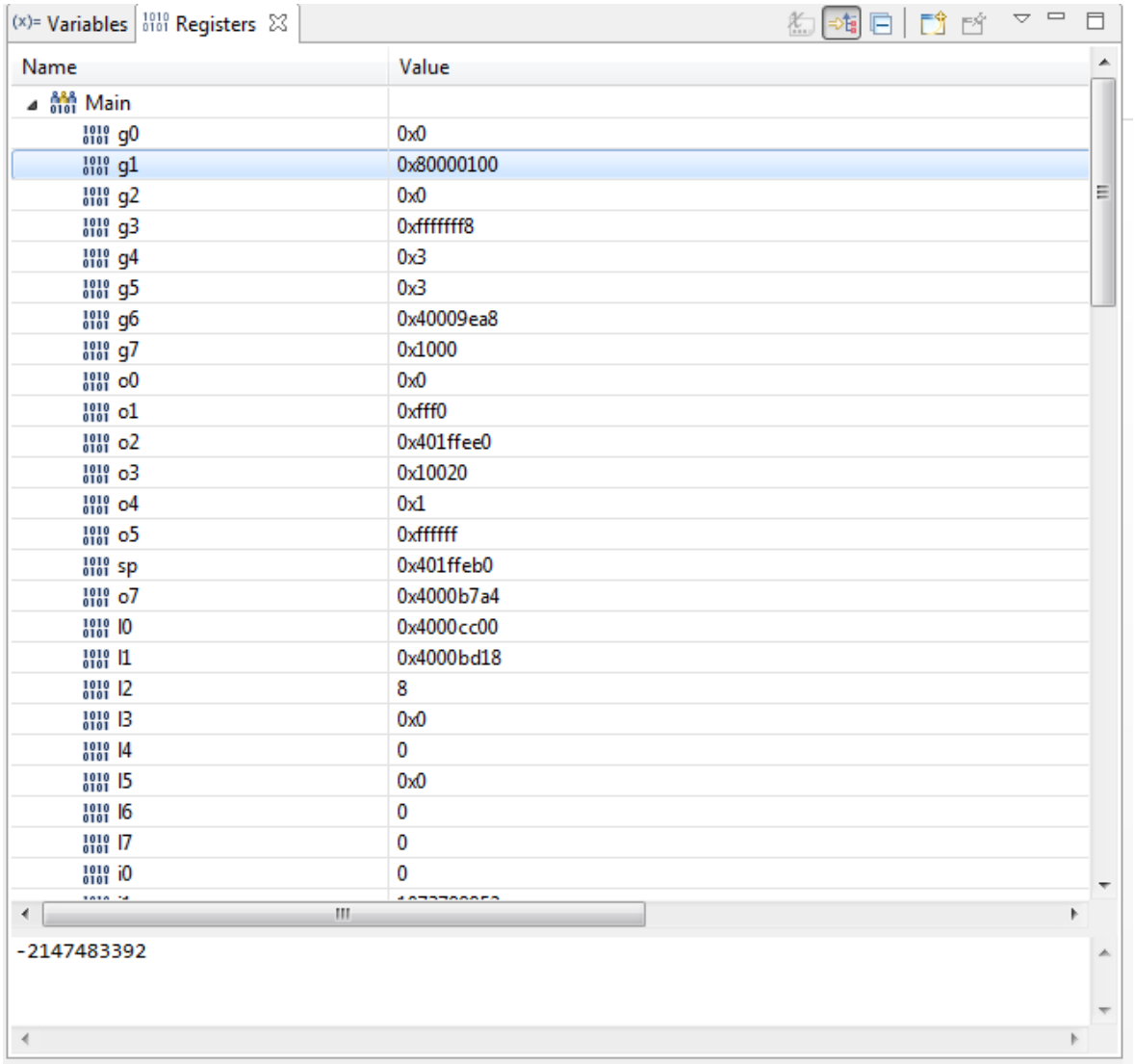


Рисунок 3.19 – Окно Registers

4 Возможные проблемы и пути их устранения

4.1 Программ "g++" и "gcc" не обнаружено в переменной PATH

После создания Cross GCC проекта при компиляции выводится две ошибки: «Program "g++" not found in PATH» и «Program "gcc" not found in PATH».

Это происходит, поскольку Eclipse по умолчанию для индексации использует gcc, а на вашем компьютере не установлена нативная версия набора программных средств GCC. Насколько нам известно, данные ошибки безвредны, но могут приводить к ложному детектированию проблем.

Следующие шаги должны помочь устранить данные ошибки:

1. Откройте меню "Project → Properties";

2. Выберите "C/C++ General → Preprocessor Include Paths, Macros etc";
3. Откройте вкладку "Providers";
4. Снимите отметку напротив "CDT Cross GCC Built-in Compiler settings";
5. Нажмите кнопку "Apply";
6. Поставьте отметку напротив "CDT Cross GCC Built-in Compiler settings";
7. Нажмите кнопку "Apply" и закройте окно.

4.2 Программа "make" не обнаружена в переменной PATH

На этапе сборки проекта выводится ошибка: «Program "make" not found in PATH».

Плагины Cross GCC plugins используют компоновщик GNU Make для сборки исполняемого файла. Компоновщик GNU Make требует наличия на компьютере программы "make". Её можно получить из проекта GnuWin32 по ссылке http://gnuwin32.sourceforge.net/packages/make.htm.

Альтернативным вариантом решения проблемы будет переключение на использование внутреннего компоновщика CDT (CDT Internal builder).

Следующие шаги изменят систему сборки:

1. Откройте меню "Project → Properties";
2. Выберите "C/C++ Build → Tool Chain Editor"
3. В качестве текущего компановщика выберите из выпадающего списка "CDT Internal Builder";
4. Повторите предыдущий шаг для всех конфигураций проекта;
5. Нажмите кнопку "Apply" и закройте окно.

4.3 Нет связи с целевым устройством

Вопрос: «I get the error message: "error creating session localhost:2222: Bad file descriptor". What am I doing wrong?»

Ответ: «This error message is usually caused by GRMON/TSIM not being able to launch correctly. Make sure that your launch configuration settings are correct and that GRMON/TSIM is launched with the right settings.»

Рисунок 4.1 – Завершение установки CodeMaster-Leon

Следует удостовериться, что TCP порт, который используется GDB для организации соединения, открыт (для исключения влияния брандмауэра можно осуществить его временное отключить и повторить попытку запуска режима отладки).

Зайдите в папку с проектом и изучите файлы логов, это может дать представление о этапе, на котором возникла проблема.

Если в файле «GS-LEON4.log» присутствует следующий текст «Session opening failed. Failed to find JEM-LEON4.dll or JEM-MST.dll», то стоит проверить установлен ли пакет Visual C++ Redistributable (см. раздел «1 Инсталляция ИСР CodeMaster-Leon»).

4.4 Превышено время ожидания запуска отладки

Для того чтобы предотвратить аварийное завершение запуска отладки из-за превышения установленного времени ожидания (timeout) рекомендуется через меню "Window → Preferences" (см. рисунок 4.2), на закладке "C/C++ → Debug → GDB MI" установить значения "Launch timeout (ms)" – 60000 или более и "Debugger timeout (ms)" – 50000 или более. Данные рекомендации приводятся, поскольку для подключения используется последовательный интерфейс, для загрузки бинарного кода в память (в особенности больших приложений) может потребоваться значительное время.

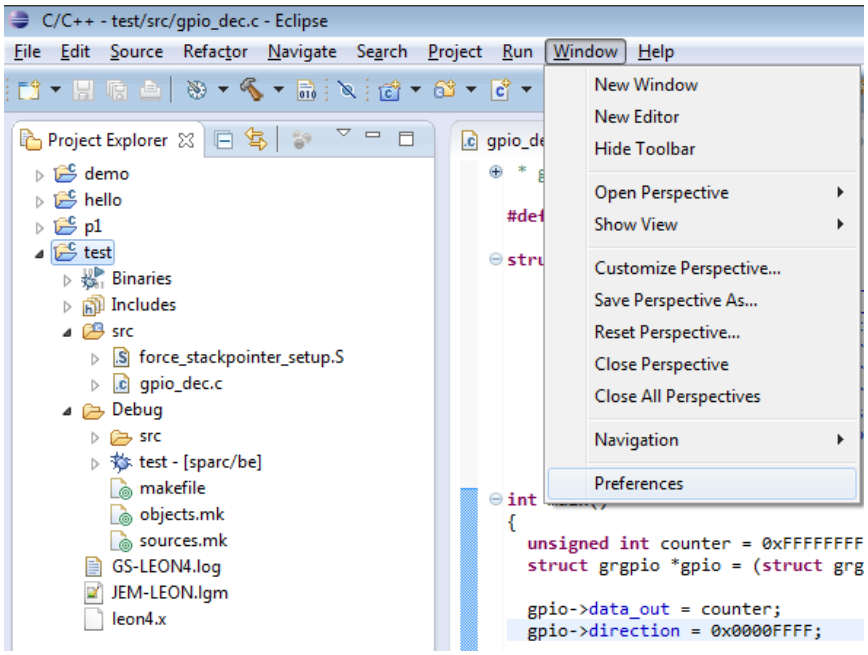


Рисунок 4.2 – Window → Preferences

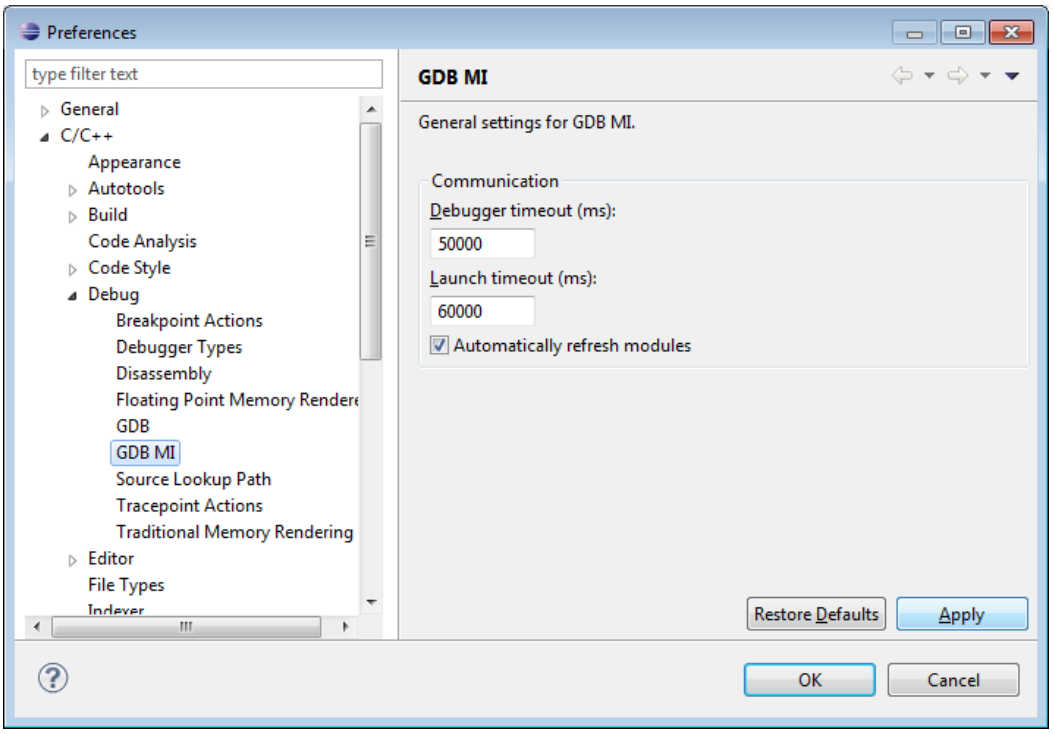


Рисунок 4.3 – Настройка времени ожидания

4.5 Сообщения при запуске отладочной сессии

При запуске отладочной сессии в консоли появляются сообщения вида:

«No symbol "auto" in current context. monitor symbol C:\workspace\rtems-tasks\Debug\rtems-tasks Previous frame identical to this frame (corrupt stack?)».

The target endianness is set automatically (currently big endian)

Это просто информационные сообщения от GDB, их можно проигнорировать.

4.6 Некорректный запуск отладки из оперативной памяти

По умолчанию (если не используется пользовательский скрипт линкера, размещающий секцию .text в область PROM) исполняемый код размещается в область RAM, которая начинается с адреса 0x40000000. За область RAM конкурируют контроллеры SRAM (активен по умолчанию) и SDRAM (не активен по умолчанию, требуется инициализации). Конфигурация по сбросу контроллера SRAM может не позволить

Для процессора 1906ВМ016 rev.4 (связано с изменением временной диаграммы доступа относительно 1906ВМ016 rev.3) рекомендуется выставить не менее одного дополнительного такта к доступам чтения SRAM, даже если время доступа к памяти не превышает периода тактового сигнала (см. рекомендации из последнего абзаца подраздела «2.7 Основные характеристики микросхемы» технического описания процессора).

Для решения данной проблемы есть несколько вариантов решения:

а) выставить необходимое значение в регистре WC_CTRL в окне Chip Selection (Window → Preferences → Chip Selection). Данное значение будет записано непосредственно перед загрузкой кода в память.

б) разместить в области PROM небольшой загрузчик, который будет устанавливать нужное значение WC_CTRL. Например, содержимое такого загрузчика для значения 0x000004e7 (WC_CTRL.PRWC = WC_CTRL.PWWC = 7, WC_CTRL.SRWC = 1, WC_CTRL.SWWC = 0):

```
0x00000000: 03 20 04 00   sethi   %hi(0x80100000), %g1
0x00000004: 84 10 24 e7   mov     0x4e7, %g2
0x00000008: c4 20 60 08   st      %g2, [%g1 + 8]
0x0000000c: 01 00 00 00   nop
```

Содержимое файла в формате SREC:

```
S00F000077635F6374726C2E7372656327
S113000003200400841024E7C420600801000000D9
S9030000FC
```

в) запускать исполняемый код из области PROM (для нее установлено заведомо большее количество дополнительных тактов доступа к памяти, чем может потребовать реальным микросхемам).

4.7 Получение информации из приложения в консоли внутри Eclipse

Вопрос:

«How can I get the output from my application in a console inside Eclipse?»

Ответ:

«Open a Local Terminal, or a Command Shell console, in Eclipse and start GRMON/TSIM from there.

To open a Local Terminal:

1. Select the Terminal view
2. Click on the Open a Terminal icon on the Terminal view toolbar
3. In the Choose terminal drop-down list, select Local Terminal
4. Press OK»

Приложение А. Подробное руководство по инсталляции ИСР

Запуск программы инсталляции ИСР CodeMaster-Leon показан на рисунке А.1.

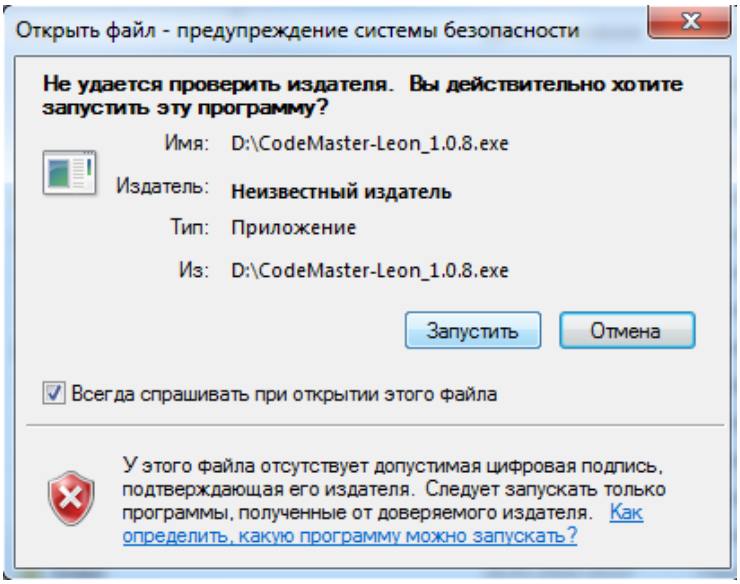


Рисунок А.1 – Запуск CodeMaster-Leon

Выбираем путь инсталляции CodeMaster-Leon стандартный (см. рисунок А.2), либо пользовательский (см. рисунок А.3).

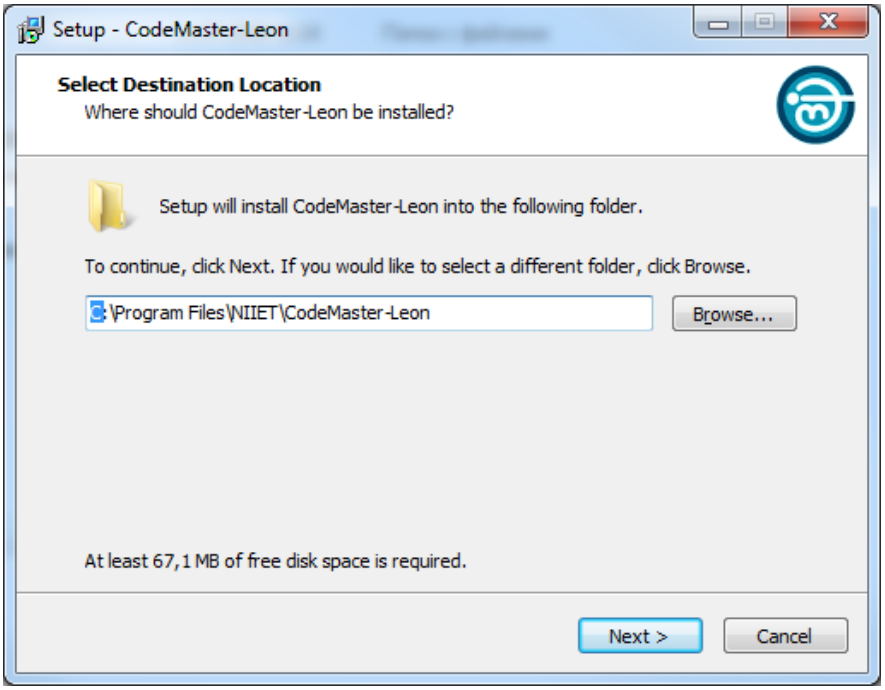


Рисунок А.2 – Стандартный путь инсталляции CodeMaster-Leon.

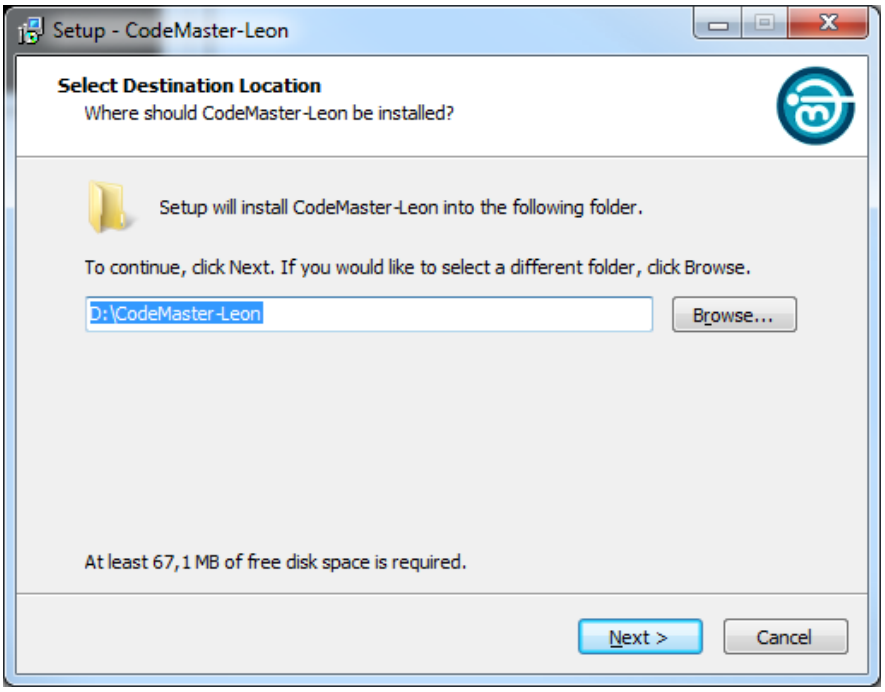


Рисунок А.3 – Пользовательский путь инсталляции CodeMaster-Leon.

Выбираем компоненты для инсталляции CodeMaster-Leon full_installation (см. рисунок А.4, compact_installation (см. рисунок А.5), custom_installation (см. рисунок А.6).

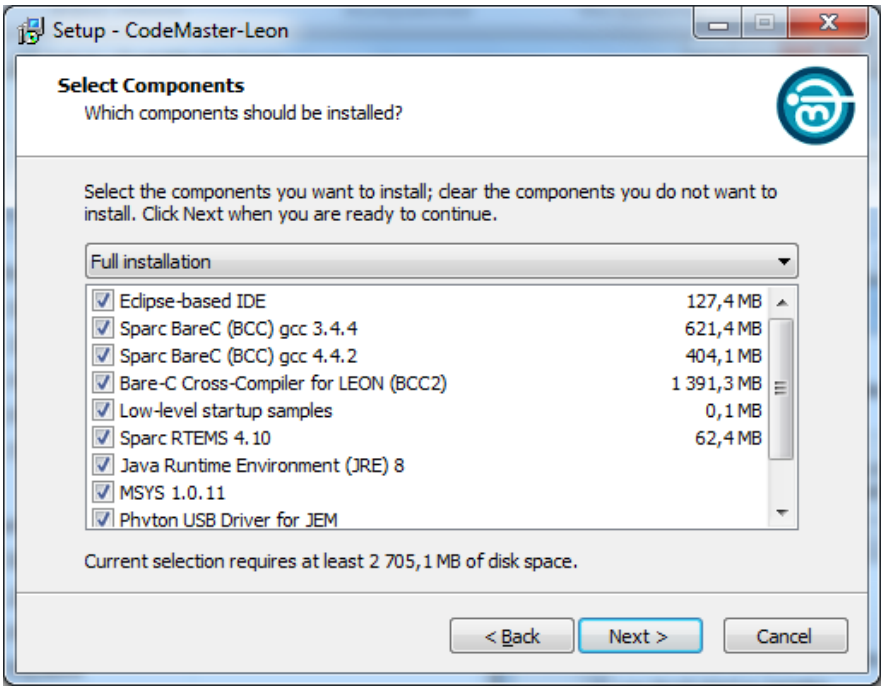


Рисунок А.4 – Выбор компонентов инсталляции full installation

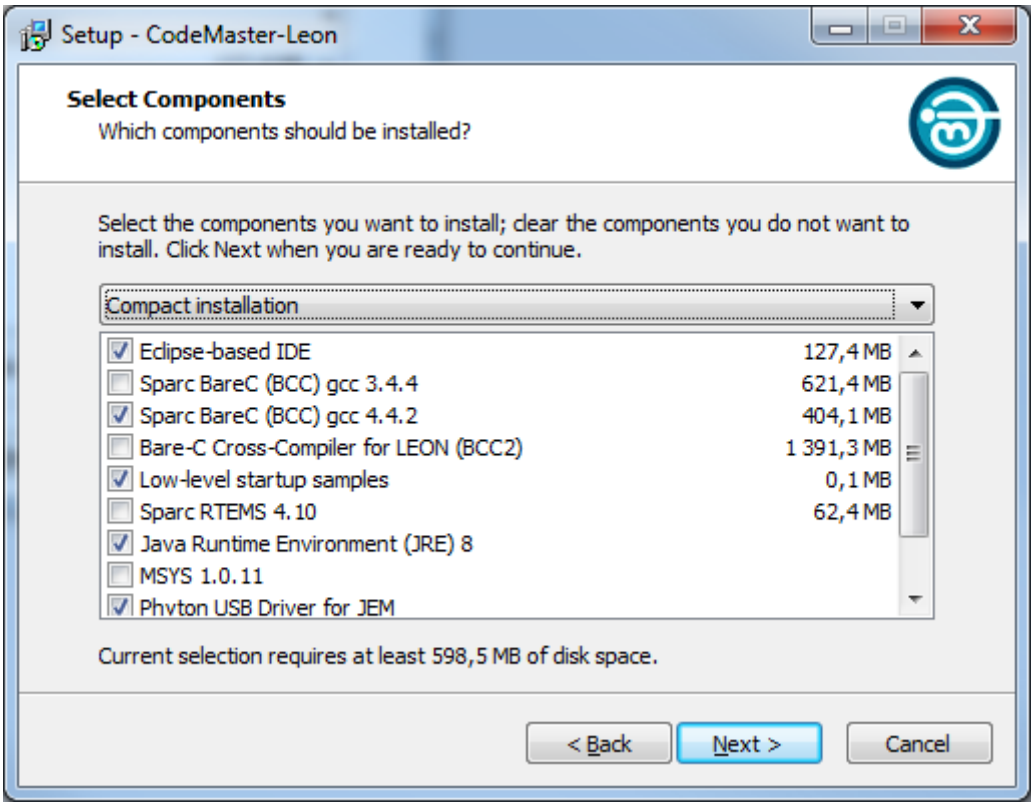


Рисунок А.5 – Выбор компонентов инсталляции compact_installation

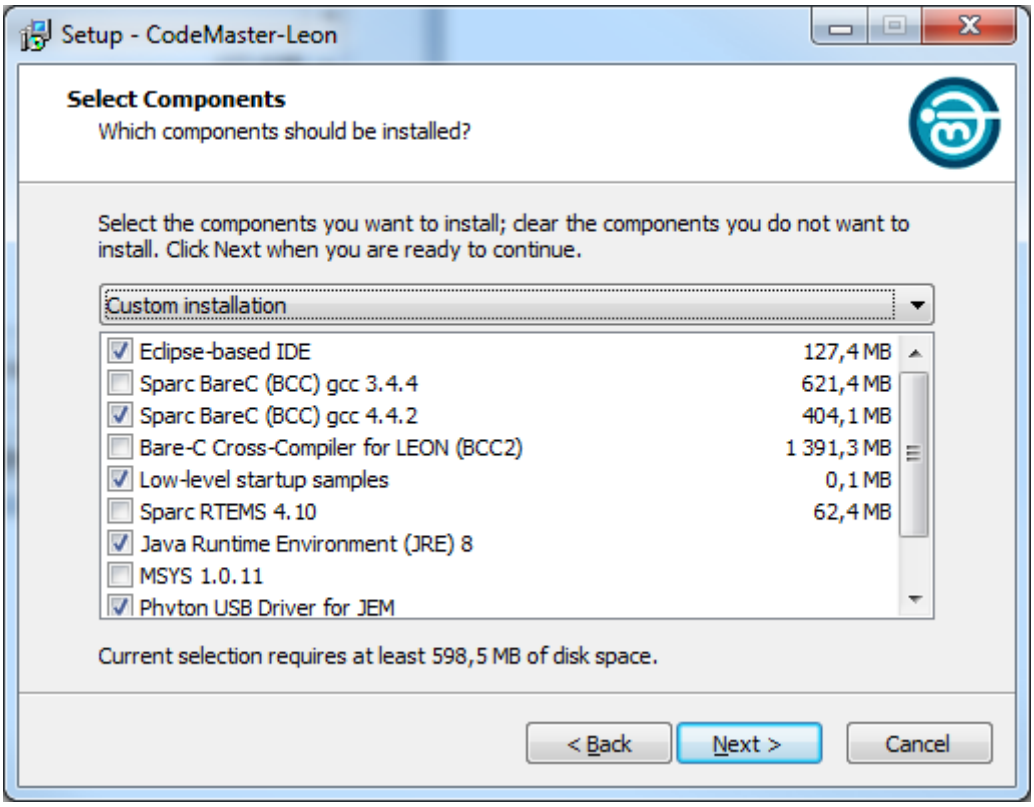


Рисунок А.6 – Выбор компонентов инсталляции custom_installation

Выбираем название рабочей папки в меню Windows для CodeMaster-Leon (см. рисунок А.7).

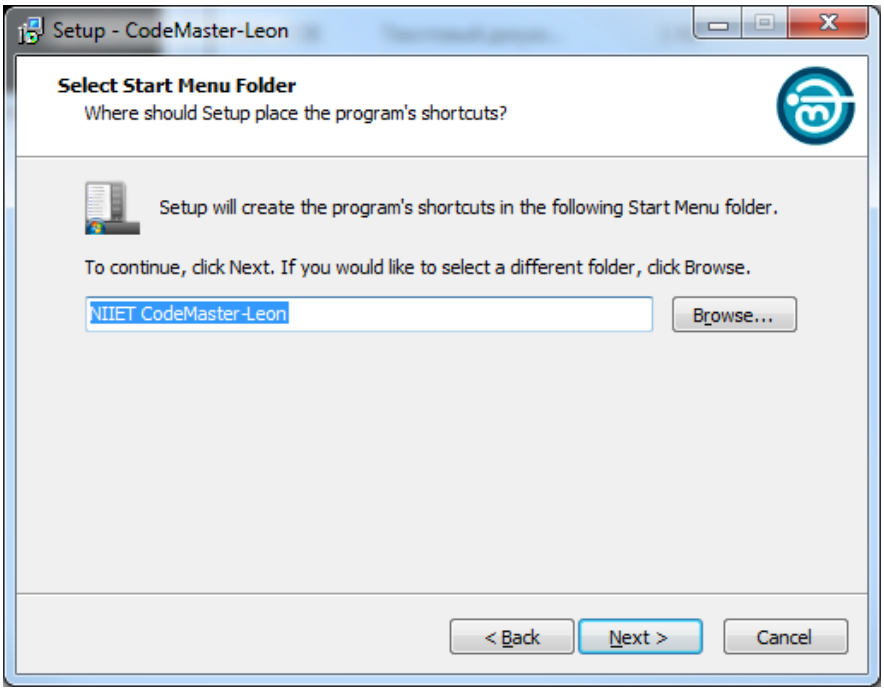


Рисунок А.7 – Выбор названия рабочей папки в меню

Просмотр всех настроек для инсталляции CodeMaster-Leon (см. рисунок А.8).

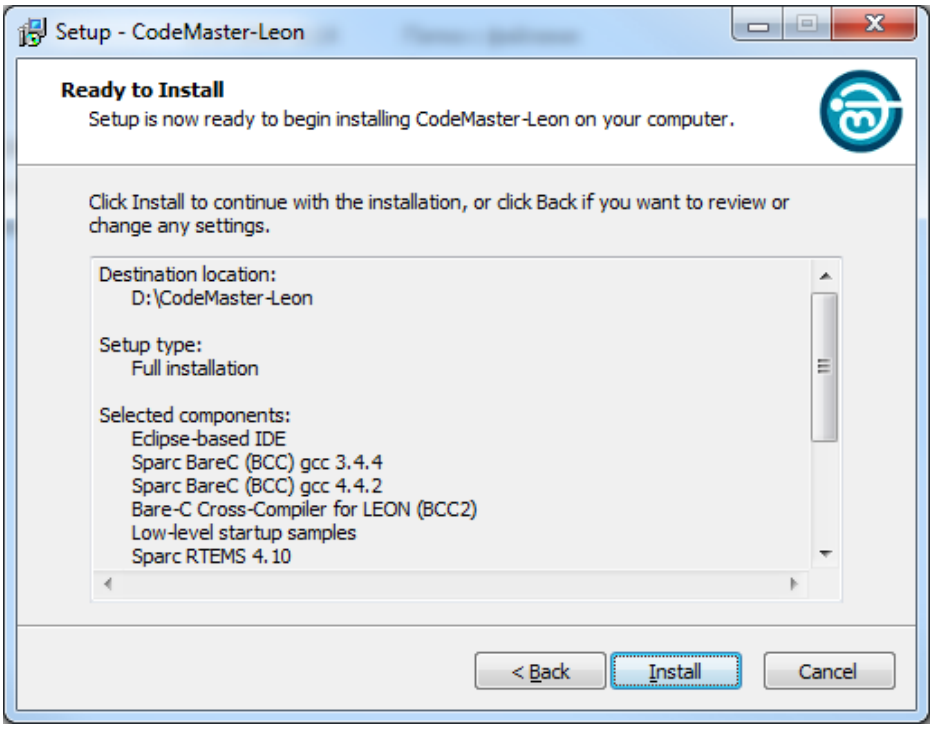


Рисунок А.8 – Просмотр всех настроек для инсталляции

После просмотра настроек инсталляции запускается процесс установки CodeMaster-Leon (см. рисунок А.9).

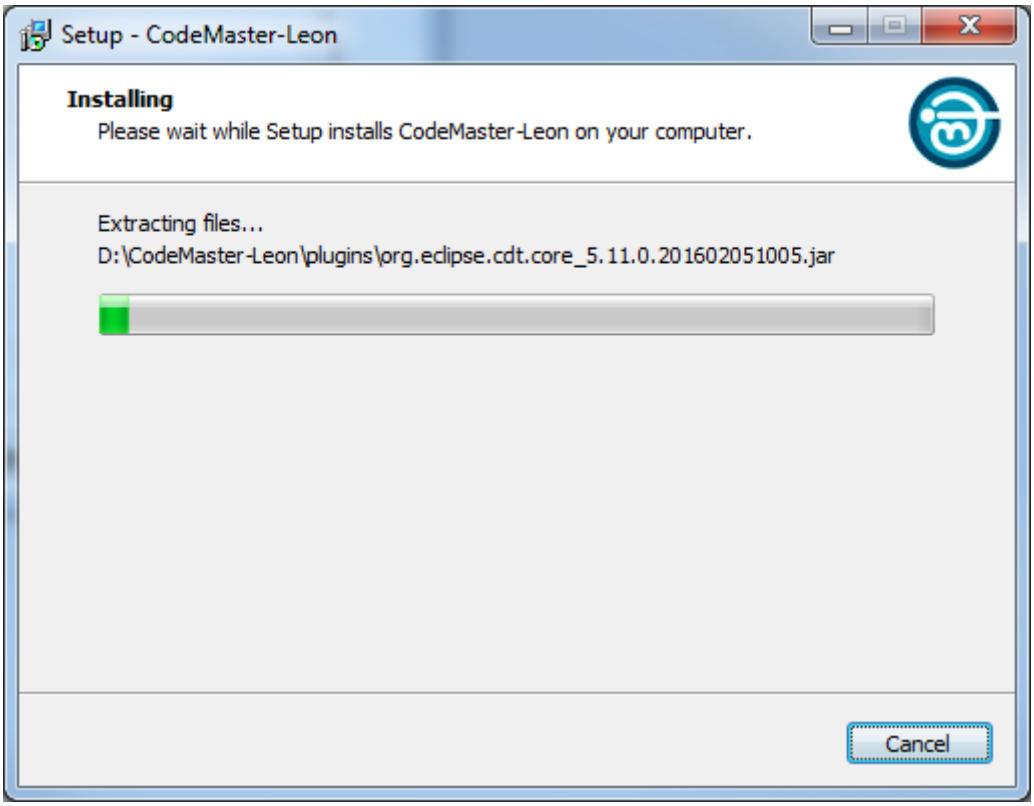


Рисунок А.9 – Установка CodeMaster-Leon_1.0.8

Появляется просьба об установке Minimal_SYStem (MSYS), соглашаемся (см. рисунок А.10.

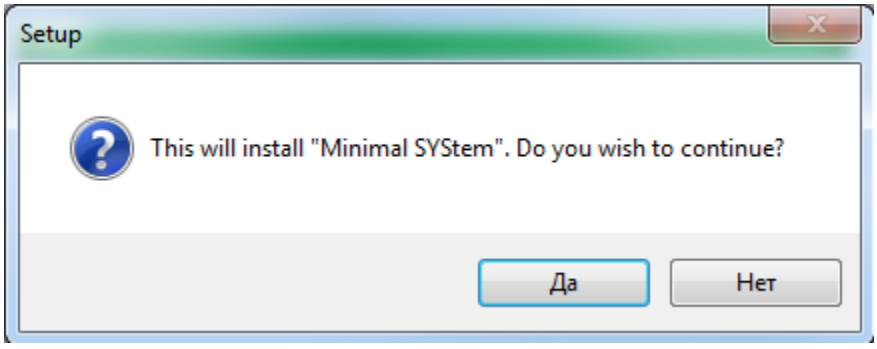


Рисунок А.10 – Просьба об установки Minimal SYStem

Появляется окно приветствия Minimal_SYStem, в нем нажимаем кнопку «Next» (см. рисунок А.11).

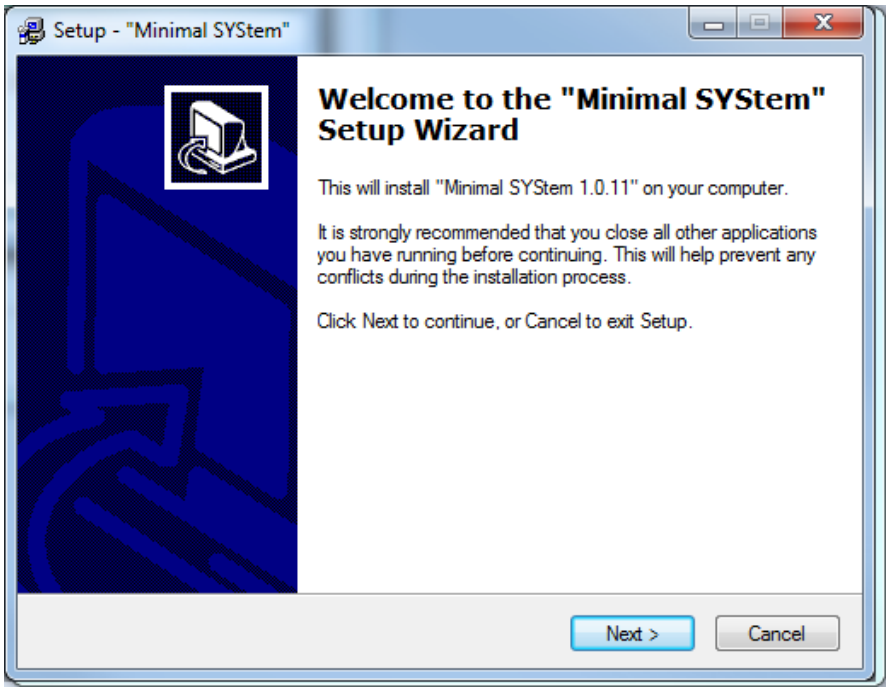


Рисунок А.11 – Окно приветствия

Соглашаемся с лицензионное соглашение Minimal_SYStem (см. рисунок А.12).

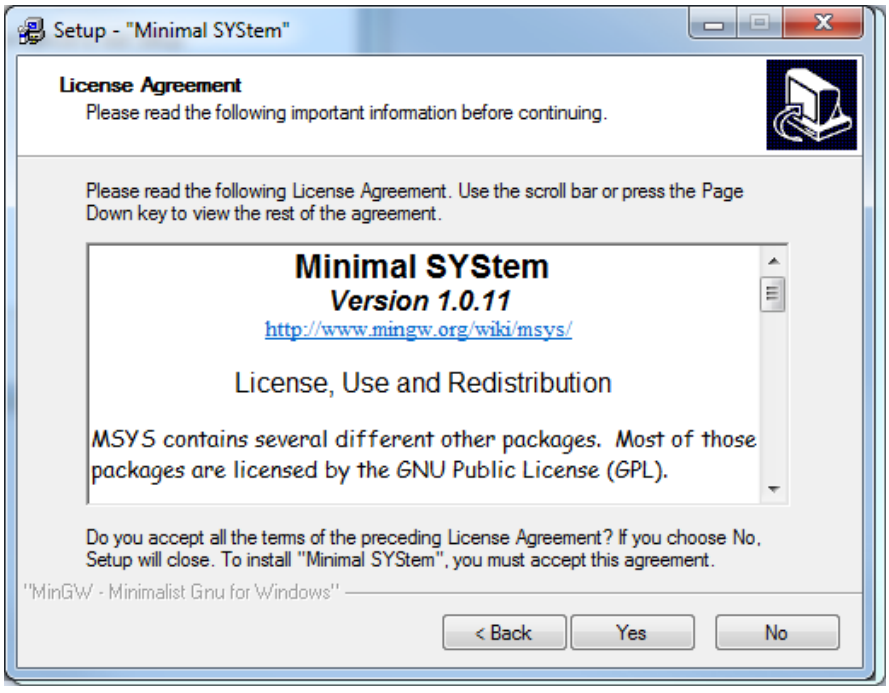


Рисунок А.12 – Лицензионное соглашение

В окне информации Minimal SYStem, следует нажать кнопку «Next» (см. рисунок А.13).

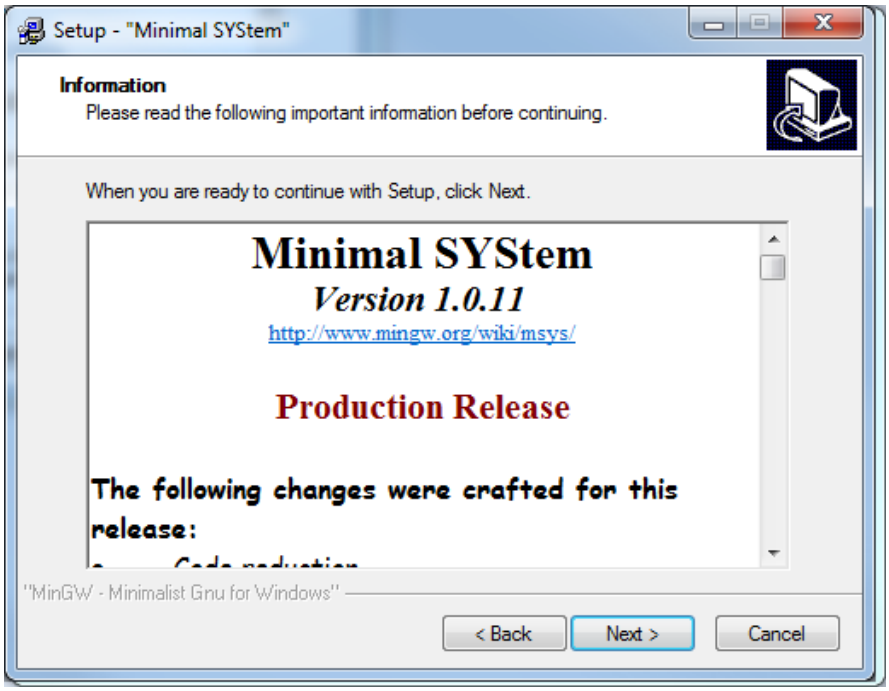


Рисунок А.13 – Окно информации

Выбор пути по которому будет установлена Minimal SYStem (см. рисунок А.14).

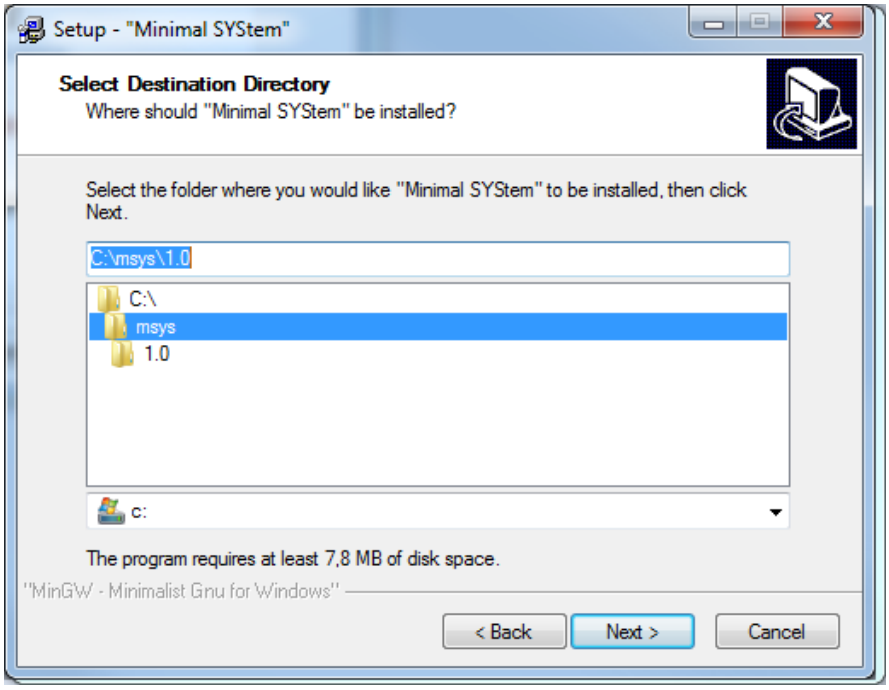


Рисунок А.14 – Выбор пути установки

Выбор имени папки Minimal SYStem в стартовом меню (см. рисунок А.15).

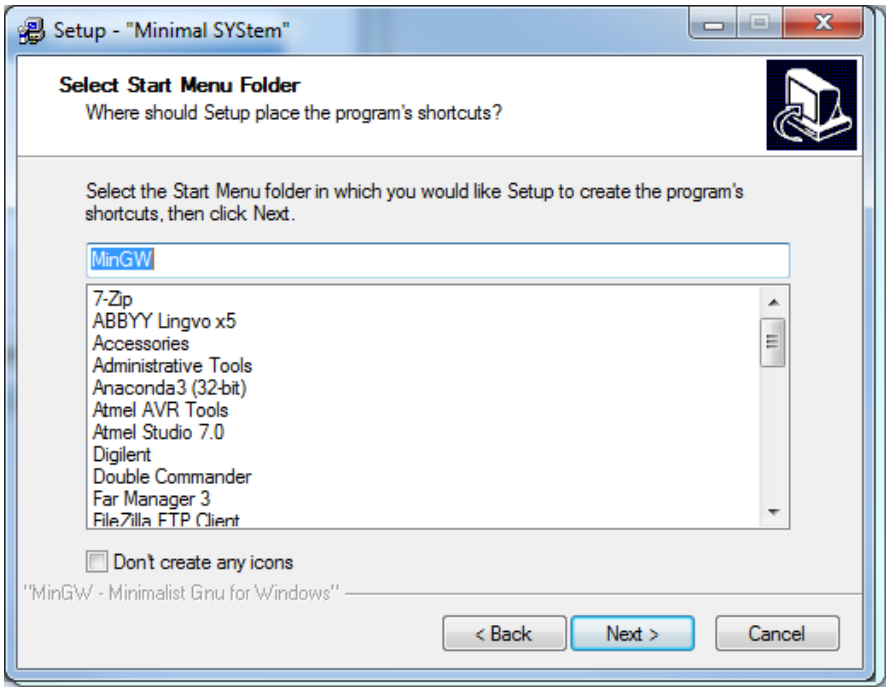


Рисунок А.15 – Выбор названия

Просмотр всех настроек для инсталляции Minimal SYStem (см. рисунок А.16).

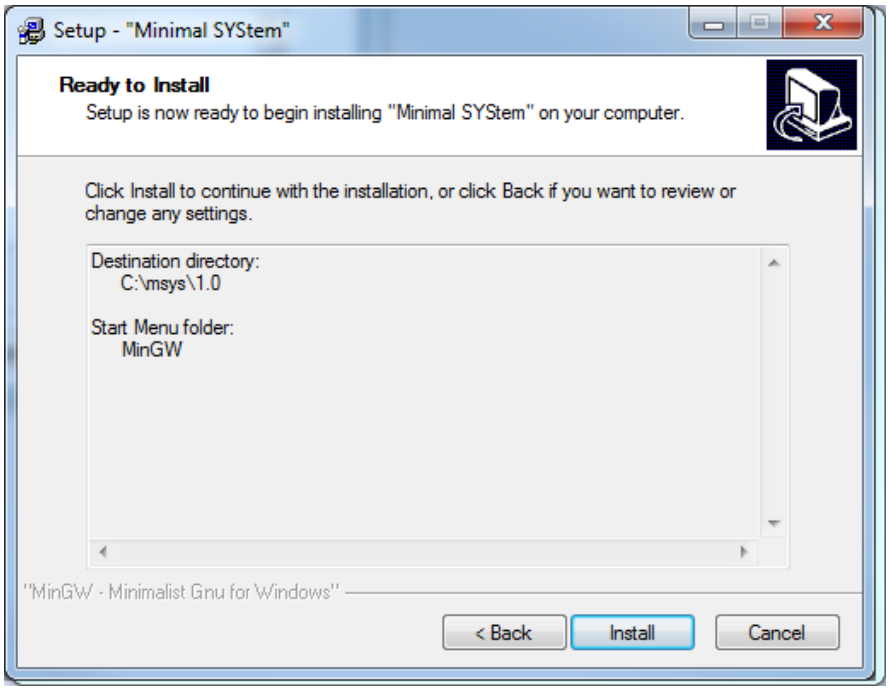


Рисунок А.16 – Просмотр всех настроек для инсталляции

Процесс инсталляции и запуск командной строки Minimal SYStem (см. рисунок А.17).

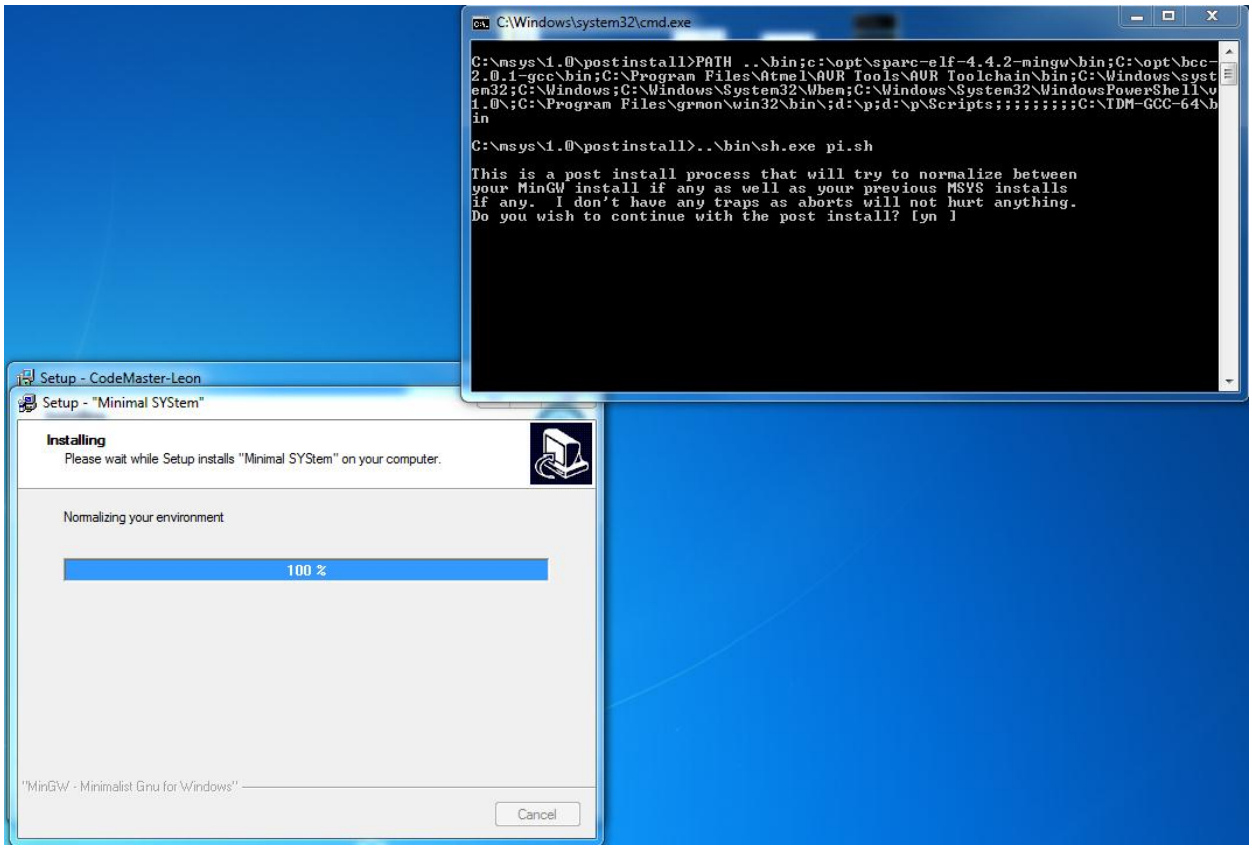


Рисунок А.17 – Процесс инсталляции и запуск командной строки

В процессе установки MSYS разрешить ему выполнить post-install действия (ответить "y"), но после указать, что других версий MinGW или MSYS не установлено (см. рисунок А.18).

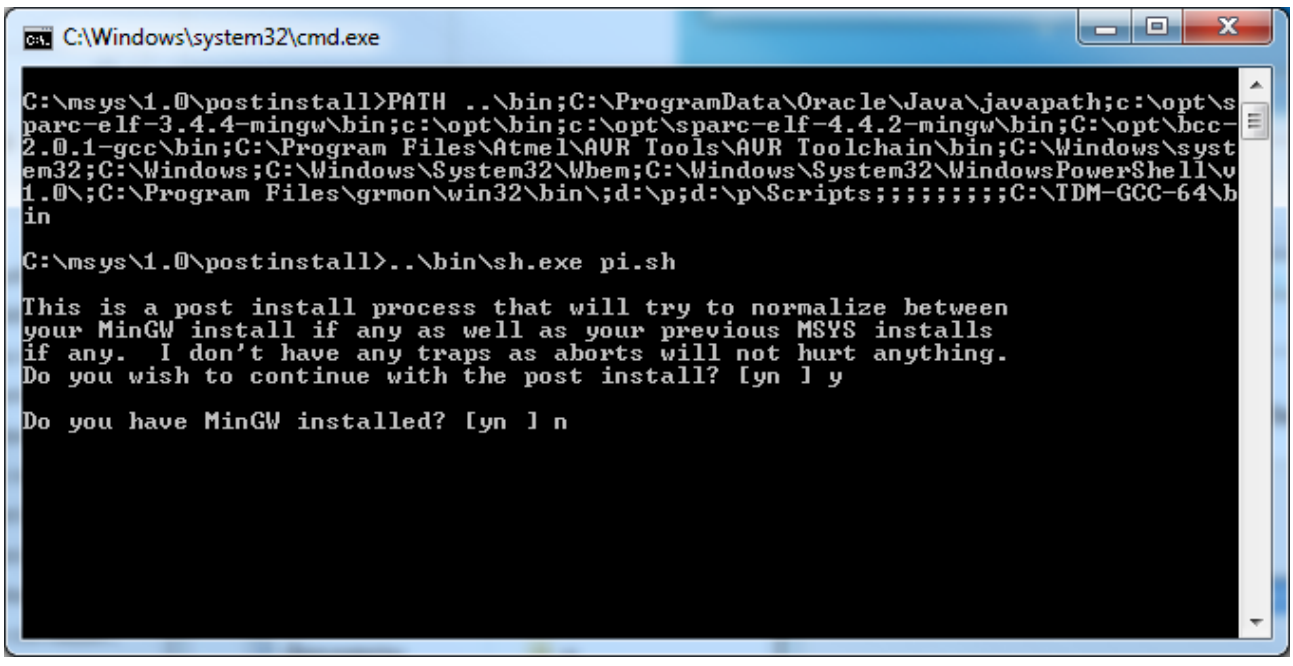


Рисунок А.18 – post-install действия (ответить "y")

Информация об успешной установке Minimal SYStem показана на (см. рисунок А.19).

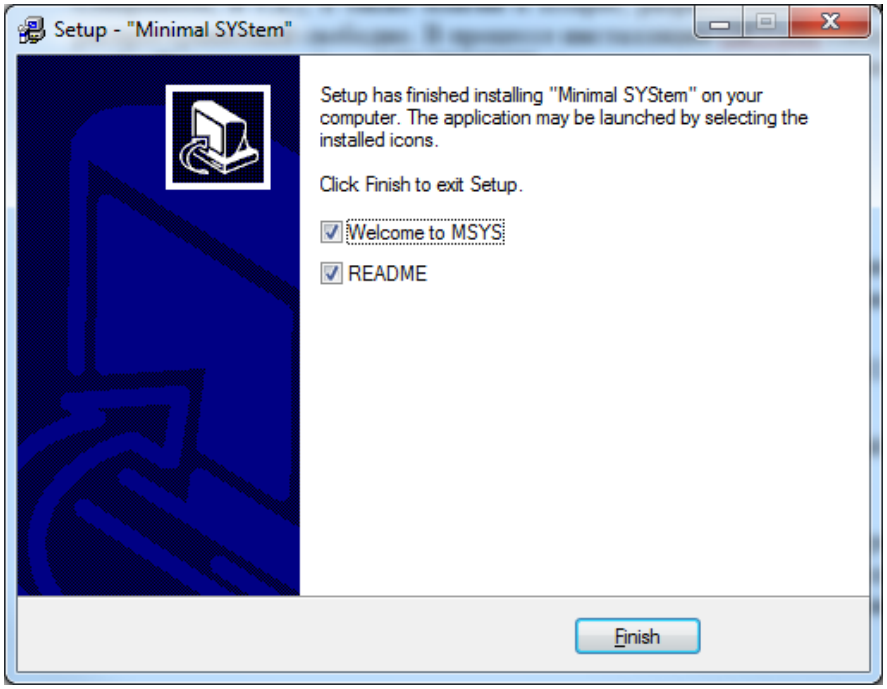


Рисунок А.19 – Успешная установка

Предложение установить Java показано на (см. рисунок А.20), в данном окне следует нажать кнопку «Next».

Рисунок А.20 – Устанавливаем Java

Установка Java показана на (см. рисунок А.21).

Рисунок А.21 – Устанавливаем Java

В случае успешной установки Java будет показано окно (см. рисунок А.22).

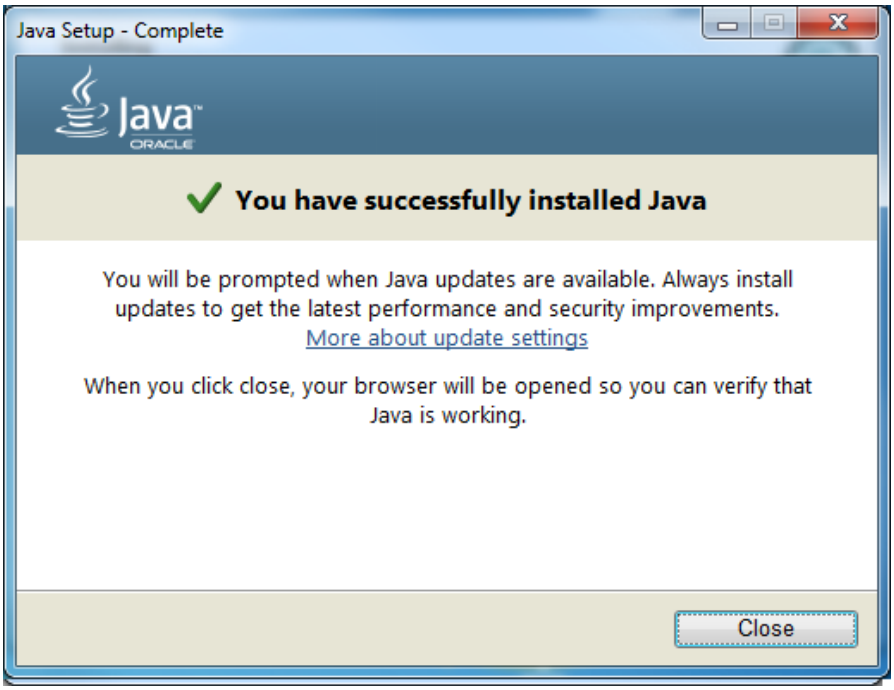


Рисунок А.22 – Успешная установка Java

Выбор языка инсталлятора Phyton USB Driver (см. рисунок А.23).

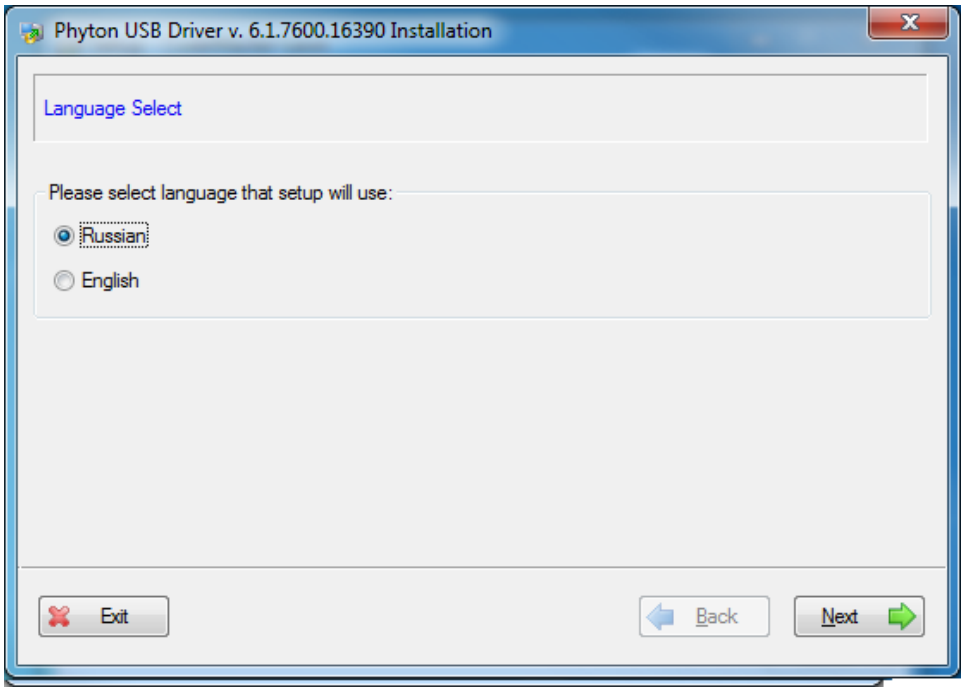


Рисунок А.23 – Установка Python

Выбор папки установки Phyton USB Driver (см. рисунок А.24).

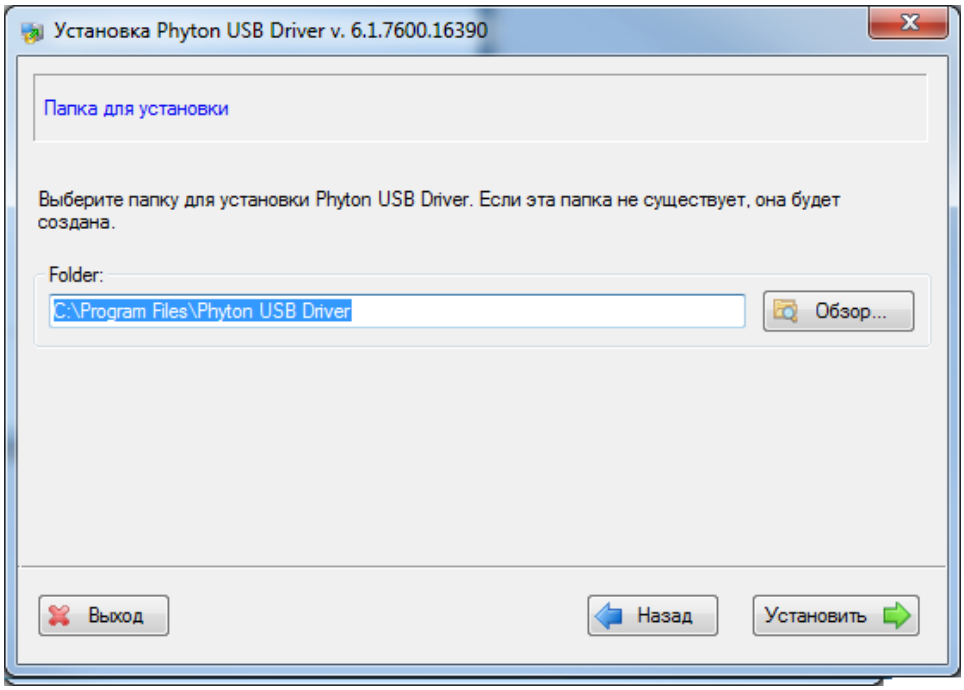


Рисунок А.24 – Выбор папки для установки Phyton

Процесс установки и её завершения для Phyton USB Driver (см. рисунок А.25).

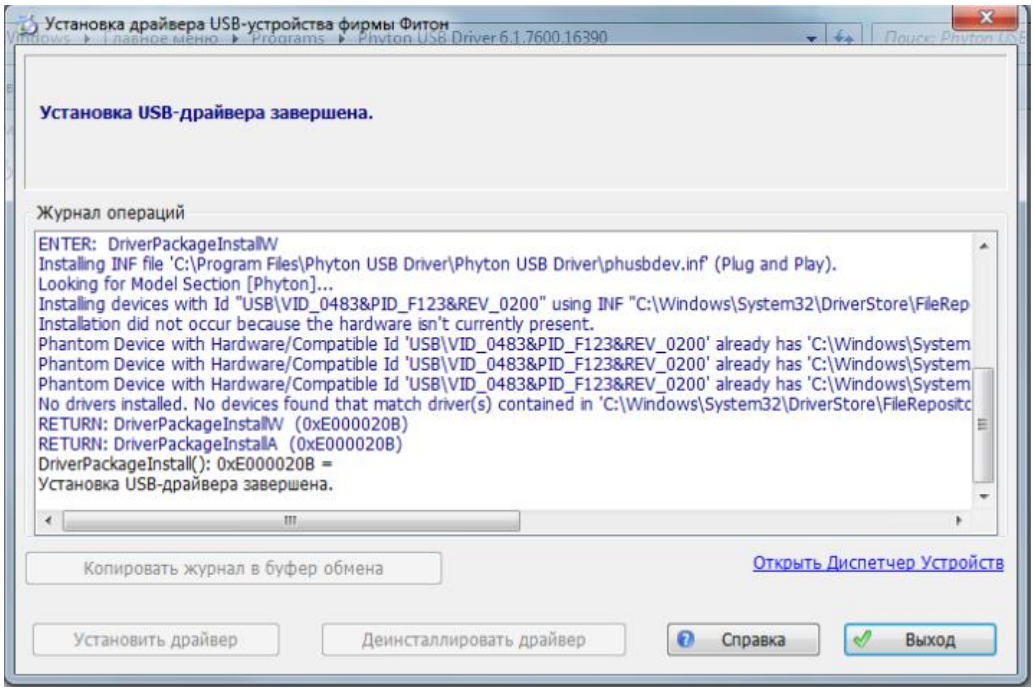


Рисунок А.25 – установка Phyton и её завершение

Завершение установки CodeMaster-Leon приведено на рисунке А.26.

Рисунок А.26 – Завершение установки CodeMaster-Leon

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (стра-ниц) в доку-менте	Номер документа	Подпись	Дата
	измене нных	замененн ых	новых	аннулир ованных				