

Руководство по установке, настройке и использованию GNU Tools/Eclipse IDE в среде Windows для разработки программного обеспечения микроконтроллеров ARM

Содержание

1. Требования к составу средств.....	1
2. Установка средств разработки.....	1
3. Создание/импорт проекта.....	4
4. Настойка проекта.....	7
5. Программирование микроконтроллера и отладка проекта.....	13

Приложение. Web-ссылки

1. Требования к составу средств

Для работы требуются компьютер с ОС Windows и следующие пакеты:

- Пакет инструментов командной строки KGP , включающий GNU Tools для ARM (Binutils, GCC, GDB, Newlib, make) JTAG сервер OpenOCD
- Схемный эмулятор Olimex ARM-USB-OCD или совместимый с ним (с драйверами)
- Среда выполнения приложений Java - Java Runtime Environment (JRE)
- Интегрированная среда разработки Eclipse IDE for C/C++ Developers.

Все компоненты актуальных версий доступны с ресурсов, указанных в заключительном разделе документа.

2. Установка средств разработки

Существует много альтернативных пакетов GNU Tools для ARM процессоров и микроконтроллеров - Sourcery G++, KGP и т.д. Здесь рассматривается пакет KGP (Klen's GNU Package). Он удобен тем, что в него включены не только компилятор(GCC), утилиты работы с бинарными файлами(Binutils) и libc-библиотека, но и отладчик(GDB), утилиты rm, sh, make из среды MinGW, а также GDB сервер OpenOCD. Таким образом, пакет KGP позволяет быстро начать процес разработки.

Пакет KGP хранится в виде архива формата 7z. Для установки пакета требуется распаковать архив, скопировать его удобную директорию и добавить к переменной **PATH** путь к директории **root/bin** пакета, где **root** – имя директории установки пакета. После этого инструменты командной строки должны стать доступными для использования. Проверить это можно командами

```
arm-kgp-eabi-ld -v
arm-kgp-eabi-gcc -v
arm-kgp-eabi-gdb -v
openocd -v
make -v
```

Если все в порядке, то команды должны вернуть свои версии.

```
C:\WINXP\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\klen.DEVELOPER>arm-kyp-eabi-ld -v
GNU ld (GNU Binutils) 2.20.51.20091127

C:\Documents and Settings\klen.DEVELOPER>arm-kyp-eabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-kyp-eabi-gcc
COLLECT_LTO_WRAPPER=d:/kyp/kyp_arm_eabi/bin/./libexec/gcc/arm-kyp-eabi/4.5
o-wrapper.exe
Target: arm-kyp-eabi
Configured with: ../../../../src/gcc/configure --prefix=/opt/home/cross/arm-kg
i/distr --target=arm-kyp-eabi --host=i686-pc-mingw32 --enable-multilib --en
interwork --with-dwarf2 --disable-shared --disable-threads --disable-libstd
ebug --with-float=soft --with-newlib --with-headers=../../src/newlib/newlib
/include --disable-libssp --disable-libmudflap --disable-libgomp --enable-l
with-ppl --with-cloog --with-pkgversion='Klen's GCC package (KGP) for AR
platform' --with-bugurl=klen_s@mail.ru 'HOST_LIBGCC2_CFLAGS=-g0 -Os -fomit
e-pointer -funroll-loops -ffunction-sections -fgraphite' FCFLAGS='-g0 -Os -
-frame-pointer -funroll-loops -ffunction-sections -fgraphite' CFLAGS='-g0 -
omit-frame-pointer -funroll-loops -ffunction-sections -fgraphite' CXXFLAGS=
-Os -fomit-frame-pointer -funroll-loops -ffunction-sections -fgraphite' --e
-languages=c,c++,fortran
Thread model: single
gcc version 4.5.0 20091127 (experimental) (Klen's GCC package (KGP) for ARM
platform)

C:\Documents and Settings\klen.DEVELOPER>arm-kyp-eabi-gdb -v
GNU gdb (GDB) 7.0.0.20091127-cvs
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later (http://gnu.org/licenses/gpl.htm
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-mingw32 --target=arm-kyp-eabi".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.

C:\Documents and Settings\klen.DEVELOPER>openocd -v
Open On-Chip Debugger 0.4.0-dev (2009-11-16-14:52)
$URL$
For bug reports, read
http://openocd.berlios.de/doc/doxygen/bugs.html

C:\Documents and Settings\klen.DEVELOPER>make -v
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for i686-pc-msys

C:\Documents and Settings\klen.DEVELOPER>
```

Установка драйверов для эмулятора Olimex ARM-USB-OCD ничем не отличается от установки драйверов для большинства USB устройств — при первом подключении устройства операционная система сообщит о попытке добавления нового устройства и попросит указать расположение драйверов.

JRE для Windows поставляется как пакет установки, который после скачивания необходимо запустить. Для проверки доступности JRE среды необходимо в командной строке ввести команду

```
java -version
```

Если среда JRE правильно установлена, команда вернет версию JRE.

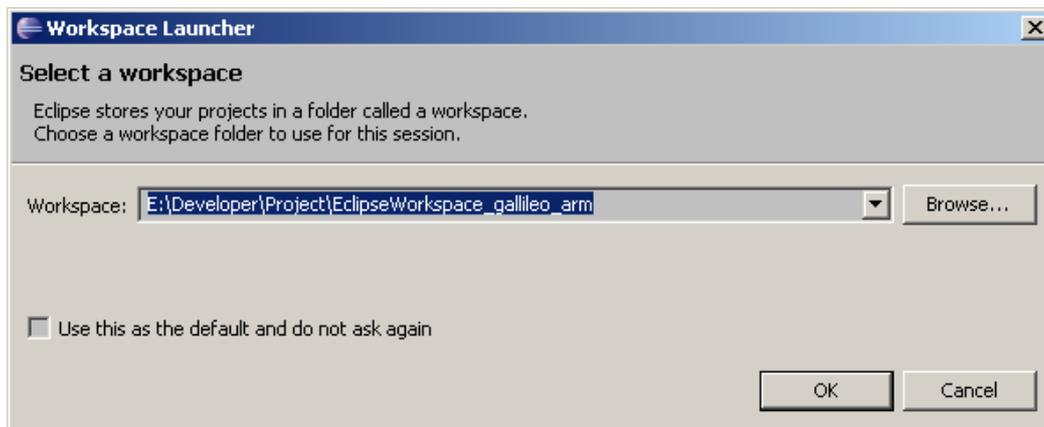
```
C:\WINXP\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\klen.DEVELOPER>java -version
java version "1.6.0_17"
Java(TM) SE Runtime Environment (build 1.6.0_17-b04)
Java HotSpot(TM) Client VM (build 14.3-b01, mixed mode, sharing)

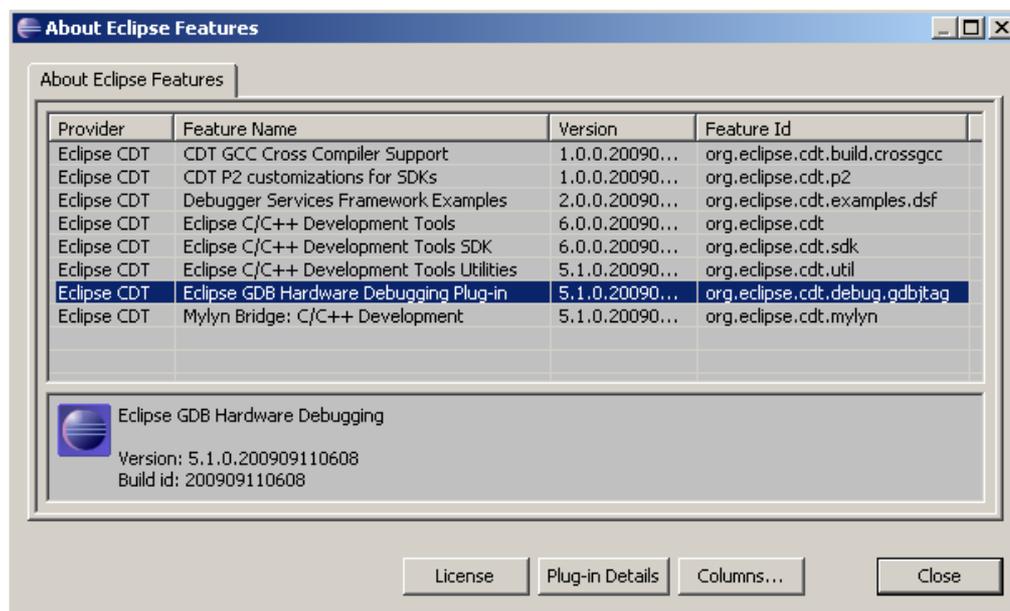
C:\Documents and Settings\klen.DEVELOPER>
```

Среда Eclipse IDE for C/C++ Developers (далее IDE) хранится в виде архива. Установка

состоит в распаковке архива и записи его содержимого в удобную директорию. После этого при успешной установке JRE-среда готова к запуску из директории установки командой *eclipse.exe*. При запуске IDE выдается запрос на указание расположения Workspace – директории, в которой будут храниться проекты.

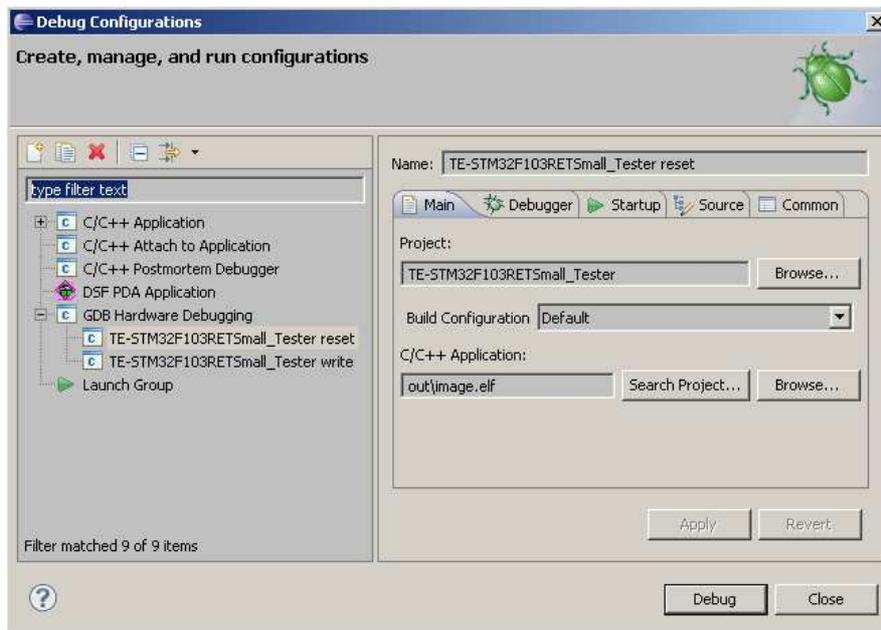


Следующим шагом является настройка среды Eclipse для разработки приложений для микроконтроллеров. Для этого необходимо поставить плагины, позволяющие выполнять отладку микроконтроллеров через JTAG. Вызвав меню *Help->Install new software* необходимо поставить плагин Eclipse GDB Hardware Debugging Plug-in, который обеспечивает возможность отладки через JTAG.

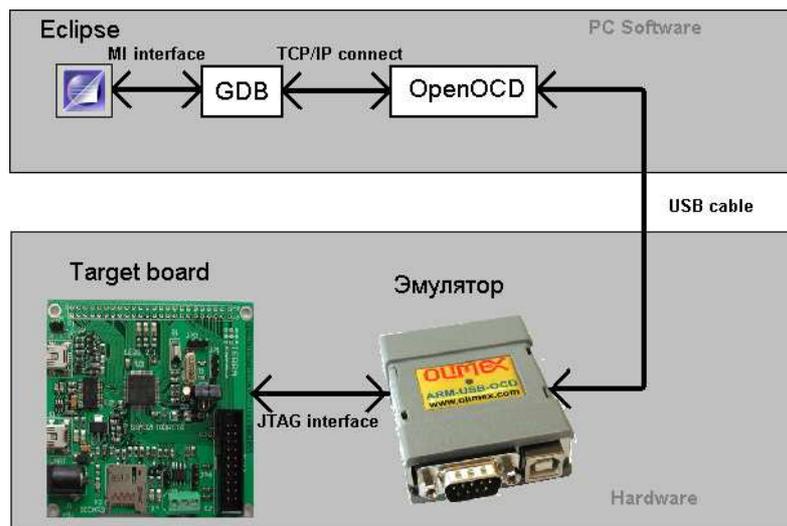


В версии IDE из архива это плагин уже установлен.

Если плагин установлен правильно, то IDE позволяет добавить конфигурацию отладки через JTAG - “GDB Hardware Debugging” которую можно настроить, вызвав меню *Debug Configuration*:

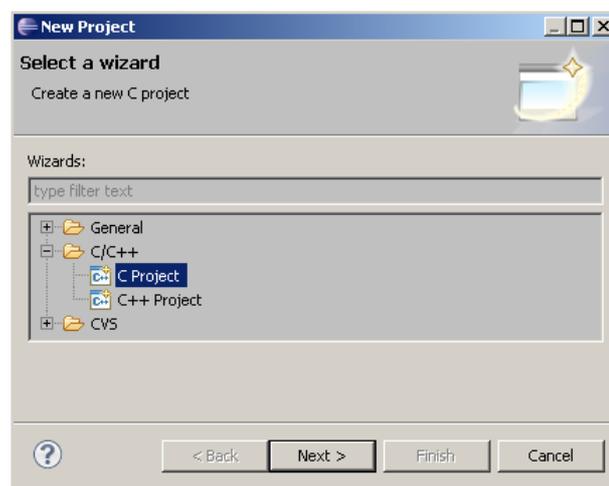


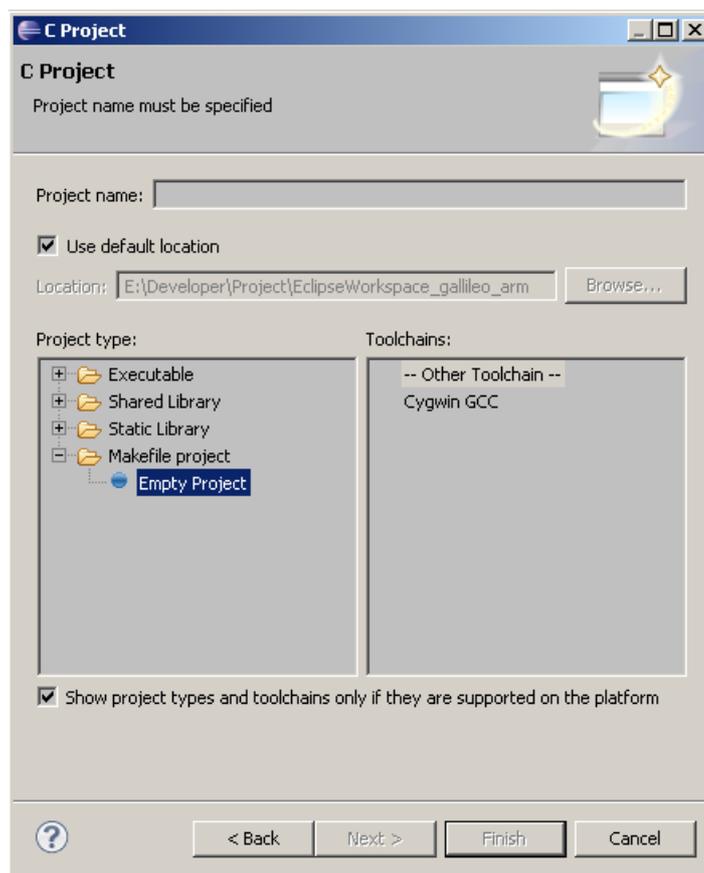
Для разработчика важно понимать, как работает IDE в связке с отладочными средствами. Не вдаваясь в подробности реализации, схема взаимодействия компонентов имеет следующий вид:



3. Создание/импорт проекта

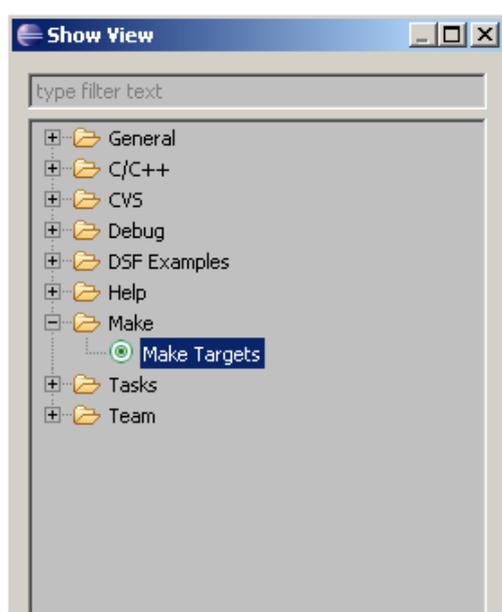
Создание проекта выполняется по меню *File->New->Project*,





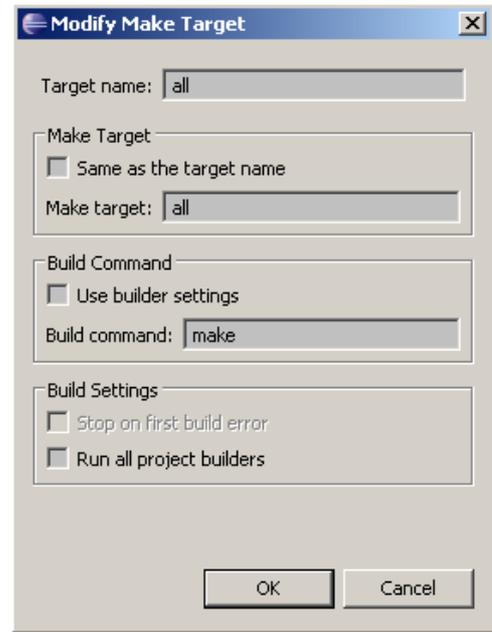
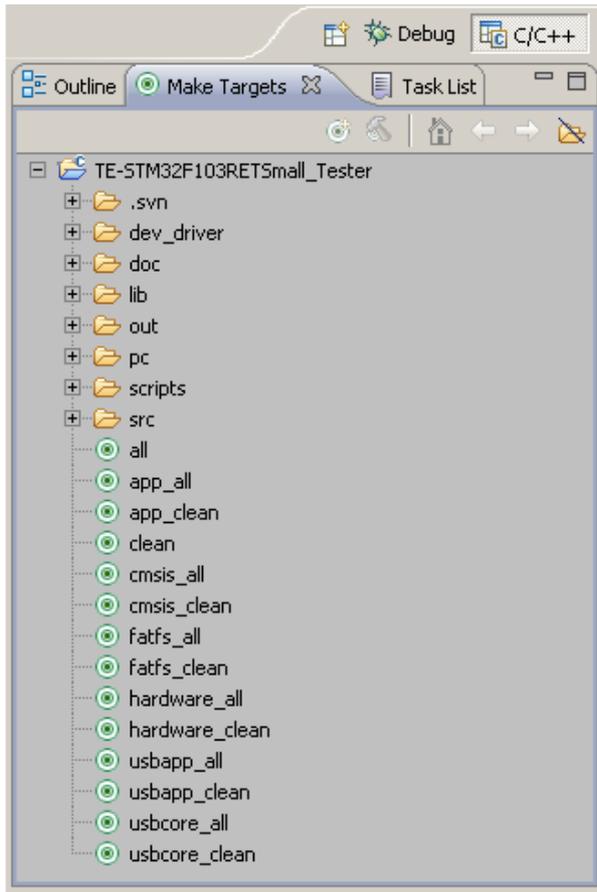
где необходимо выбрать **C Project/Makefile project/Empty Project**

После указания имени проекта и нажатия кнопки **Finish**, IDE создаст пустой проект. Добавив дерево исходников и Makefile в директорию проекта, Вы сможете работать с проектом. Все действия с проектом, которые требуются при разработке, должны быть отражены в виде make целей и зависимостей. IDE, со своей стороны, позволяет выполнить любую команду по выполнению той или иной цели. Данная функциональность отражена в окне Make Targets. Если это окно не отражено в текущей «перспективе», то его можно отобразить с помощью команды меню **Windows->Show View->Other**, необходимо выбрать **Make Targets**.

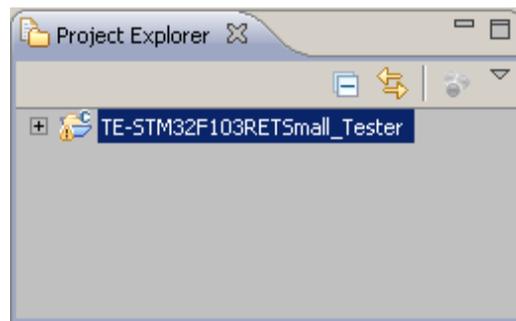


Теперь, имея доступ к этому окну, можно добавить команды для выполнения целей

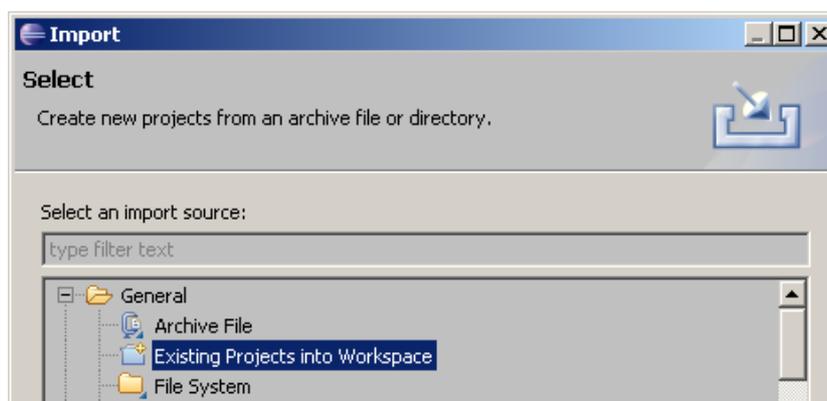
проекта.



IDE поддерживает импорт существующего проекта. Можно выполнить импорт проекта тестирования платы TE-STM32F103, имеющегося на компакт диске. В окне Project Explorer нажатием правой кнопки мыши вызвать меню **Import**



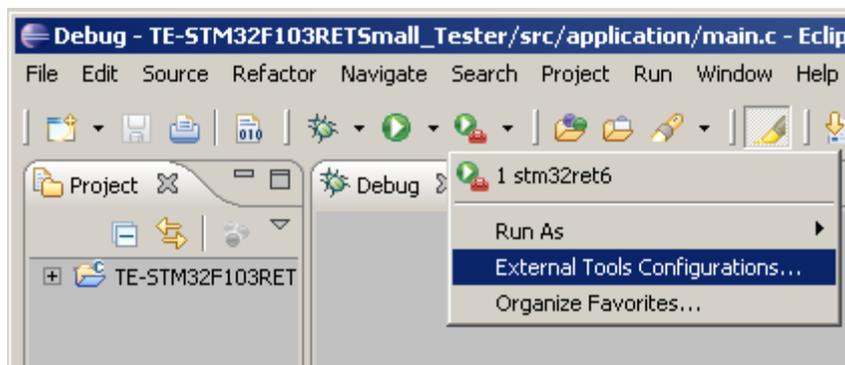
Выбрать **Exist Projects into Workspace**



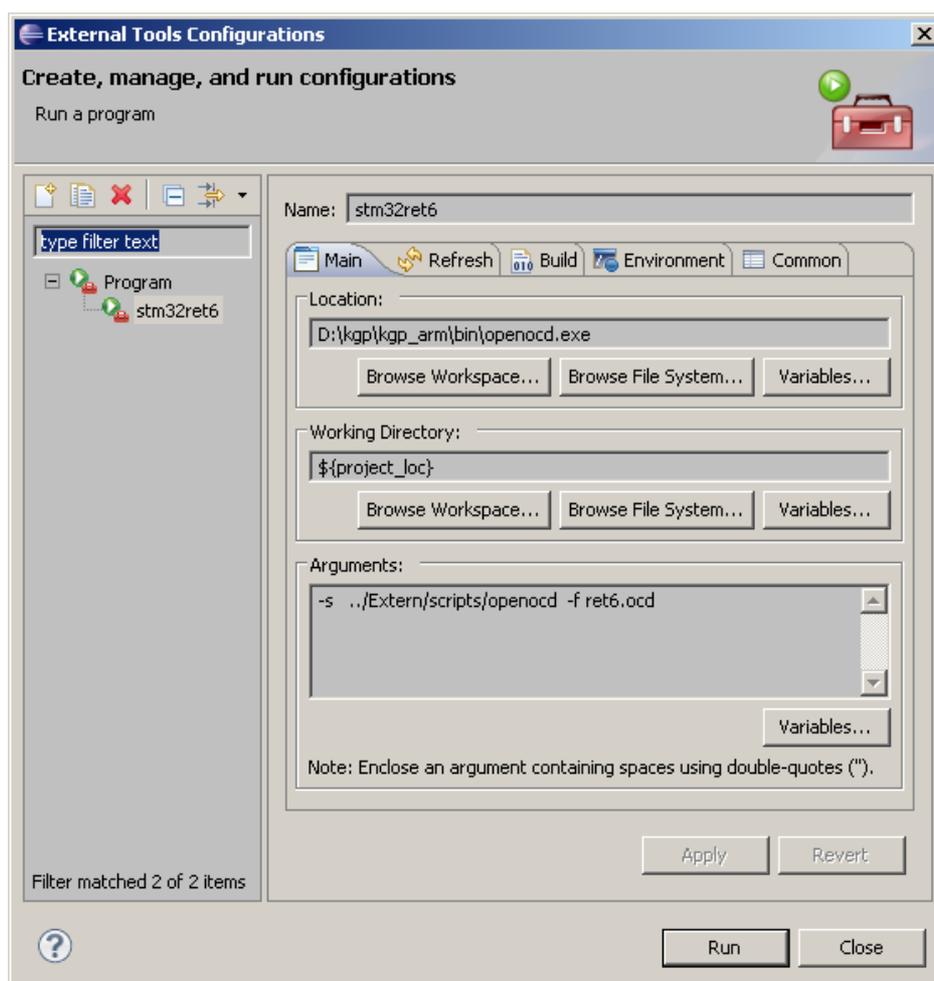
После нажатия *Next* и указания пути к импортируемому проекту IDE скопирует проект в «Workspace». В окне *Project Explore* должно появиться дерево файлов проекта.

4. Настойка проекта

Настройка проекта связана в первую очередь с установкой параметров отладки. Вначале нужно настроить IDE для работы с OpenOCD. После выбора *External Tools Configuration*

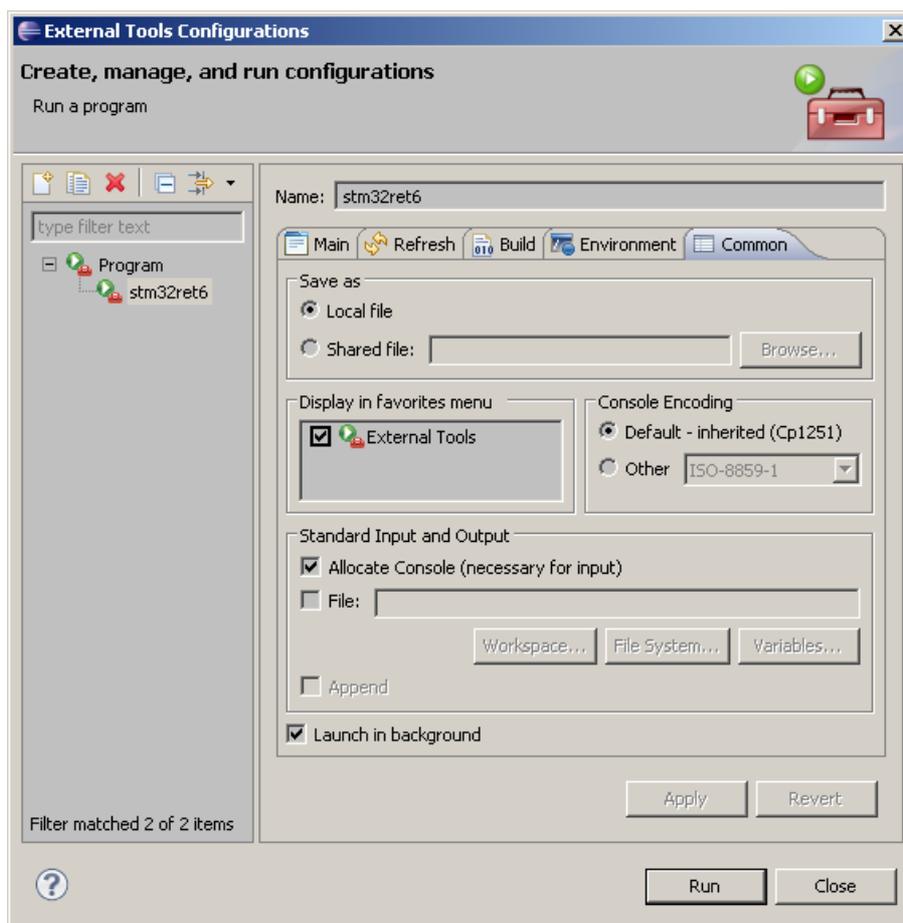


откроется диалоговое окно настройки внешних программ.

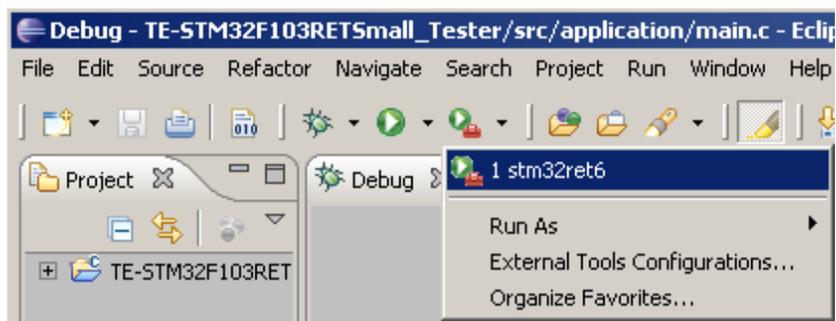


Используя его, нужно добавить новую внешнюю программу и заполнить вкладки. Основной вкладкой в данном случае является *Main*, она позволяет определить путь к файлу внешней программы, указать рабочую директорию и аргументы командной строки. В данном случае в качестве аргумента указывается внешняя по отношению к проекту

директория с файлами конфигурации OpenOCD при работе с Stm32 и сам файл конфигурации OpenOCD.



Вкладка **Common** обеспечивает настройку вывода лога OpenOCD в одно из окон IDE, а также быстрый вызов OpenOCD из меню.



После настройки взаимодействия IDE с OpenOCD, необходимо проверить работоспособность конфигурации. Подключите к компьютеру целевую плату (TARGET) через эмулятор, соединив их кабелем JTAG. Сделайте проект активным и вызовите из меню OpenOCD (на рисунке выше меню «1 stm32ret6»). В случае успеха лог будет примерно следующим:

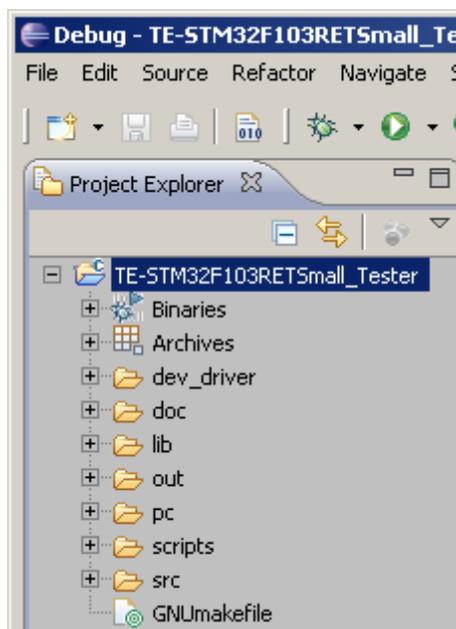
```
stm32ret6 [Program] D:\kqp\kqp_arm\bin\openocd.exe
Open On-Chip Debugger 0.4.0-dev (2009-11-16-14:52)
$URL$
For bug reports, read
    http://openocd.berlios.de/doc/doxygen/bugs.html
750 kHz
jtag_nsrst_delay: 100
jtag_ntrst_delay: 100
trst_and_srst separate srst_gates_jtag trst_push_pull srst_open_drain
Warn : stm32.bs: nonstandard IR mask
Warn : use 'stm32.cpu' as target identifier, not '0'
Info : device: 4 "2232C"
Info : deviceID: 364511235
Info : SerialNumber: FTP7LM76A
Info : Description: Olimex OpenOCD JTAG A
Info : clock speed 750 kHz
Info : JTAG tap: stm32.cpu tap/device found: 0x3ba00477 (mfg: 0x23b, part:
Info : JTAG tap: stm32.bs tap/device found: 0x06414041 (mfg: 0x020, part:
Writable Smart Insert 39 : 17
```

Также, IDE добавляет отображение запущенного процесса в окно Debug.

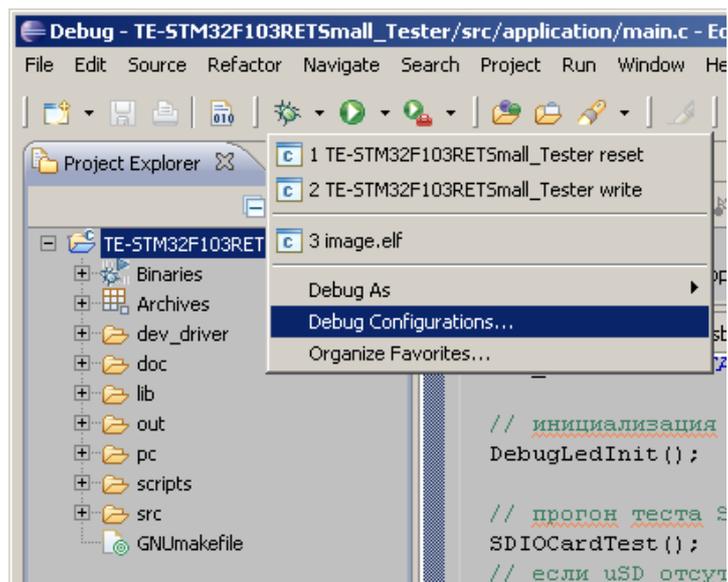


OpenOCD подключился к эмулятору, произвел его инициализацию и TAP-контролер процессора целевой платы и готов к работе.

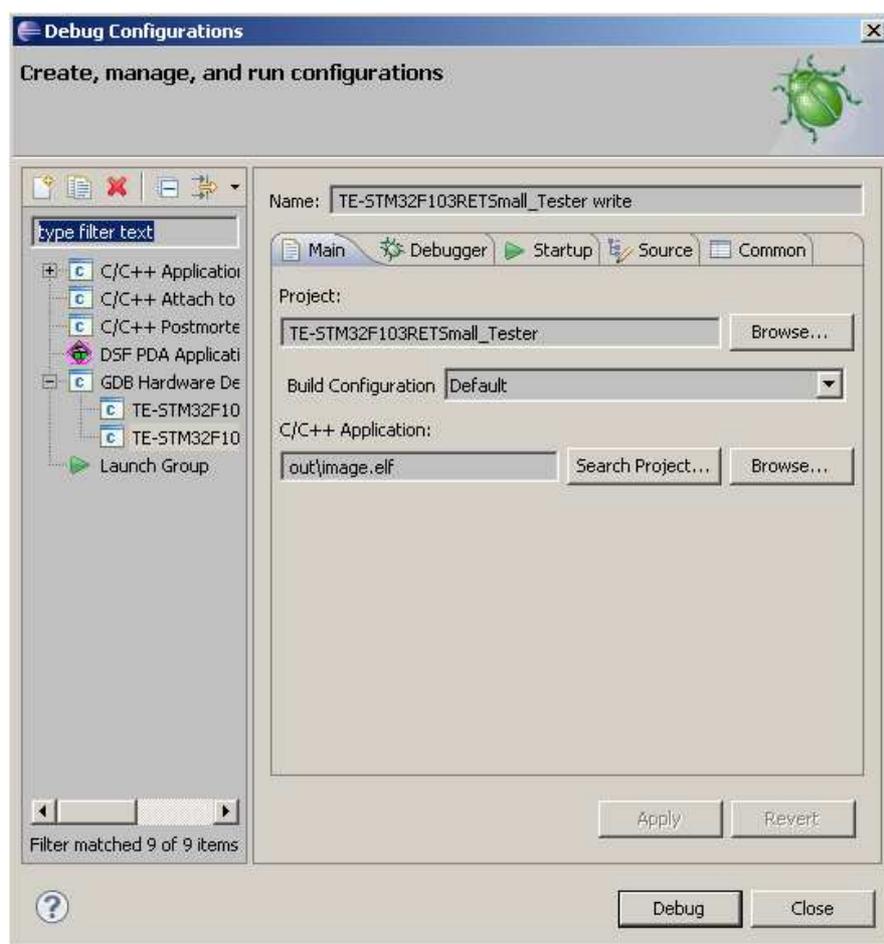
Теперь нужно настроить **GDB**. Для этого необходимо открыть текущий проект и настроить конфигурацию отладки (все действия по конфигурации всегда выполняются по отношению к открытому и выбранному проекту).



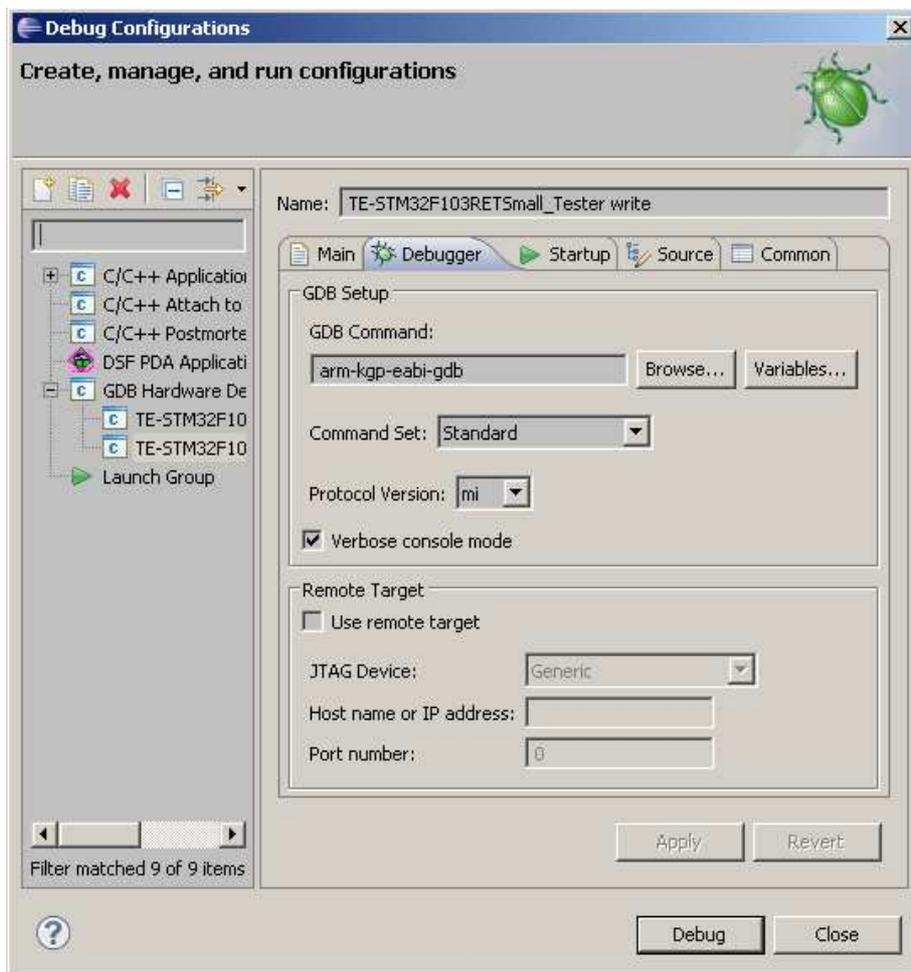
После выбора *Debug Configuration*



откроется диалог конфигурации отладки



Вначале необходимо добавить конфигурацию в раздел дерева меню *GDB Hardware Debugging*, затем дать ему имя и настроить вкладки.

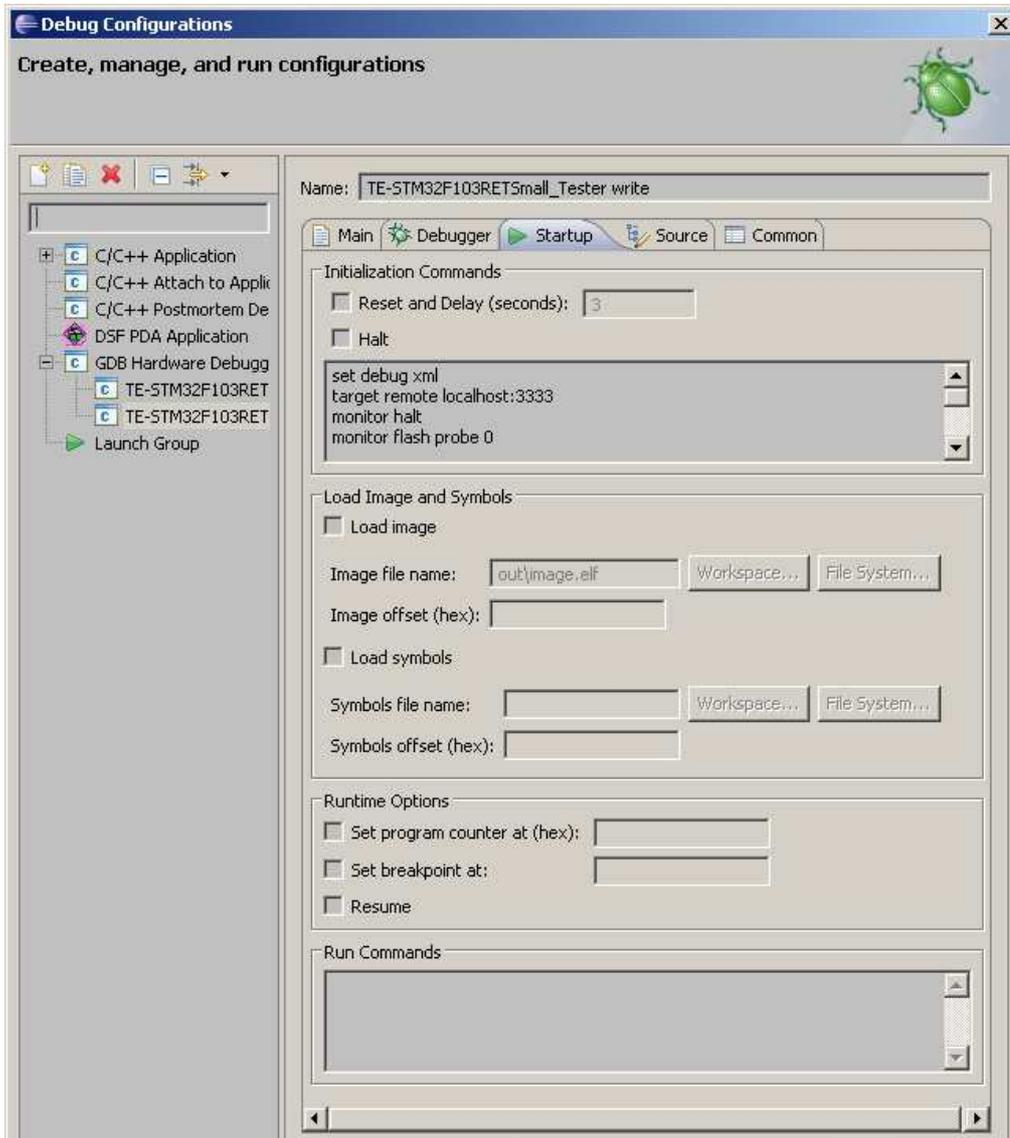


В поле GDB Command необходимо указать используемый отладчик. Остальные поля рекомендуется оставить, как на рисунках.

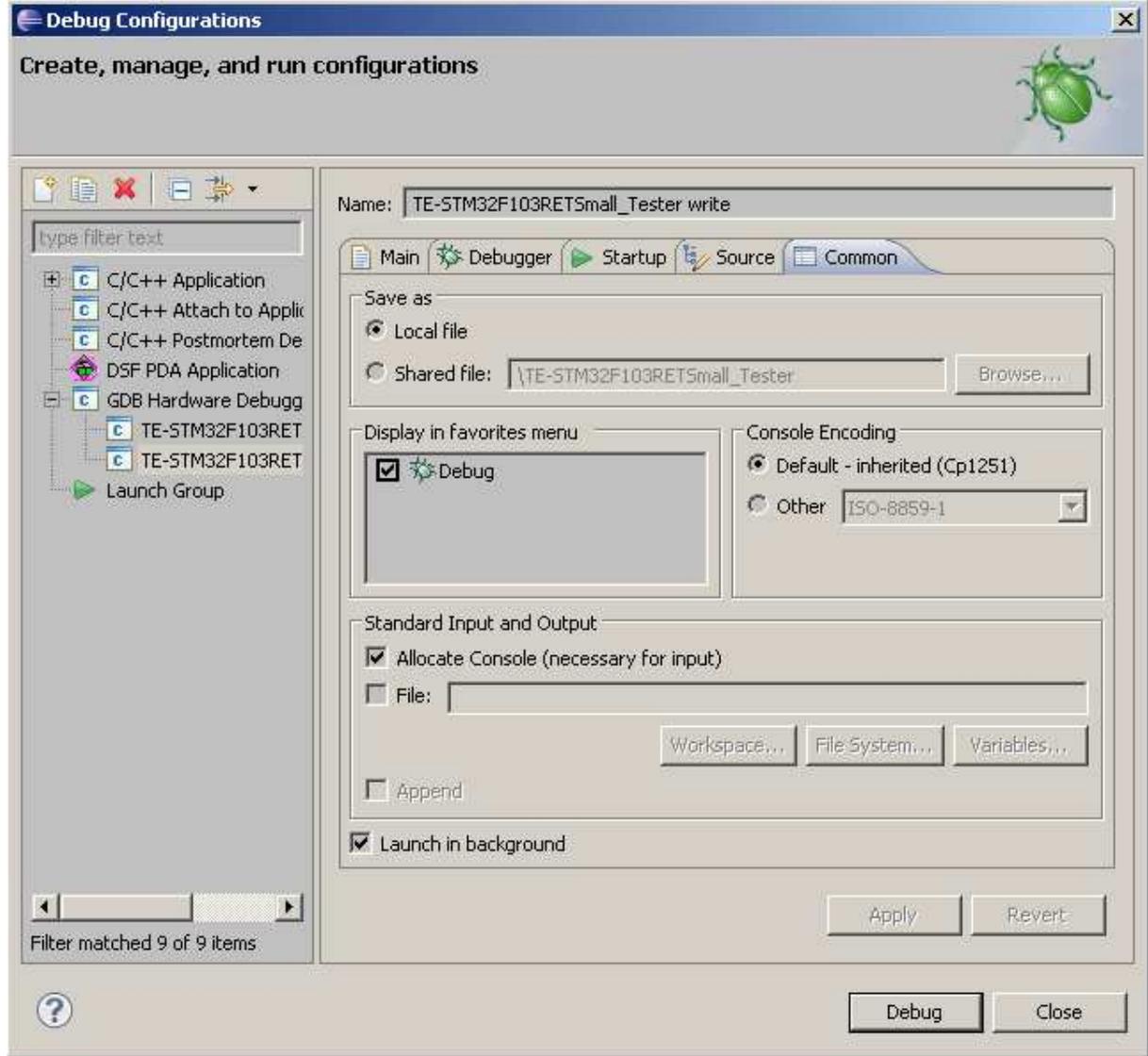
Ниже приведена вкладка **Startup**. Самым важным полем является **Initialization Command**. Необходимо отметить, что инициализация связи GDB с эмулятором происходит исключительно под управлением команд скрипта. Любые другие элементы вкладки должны быть неактивны, если Вы сами целенаправленно не желаете воспользоваться встроенными автоматическими возможностями IDE по взаимодействию с GDB.

Скрипт инициализации(для конфигурации запись+отладка) имеет вид	
set debug xml	поддержка описаний и карт памяти на XML
target remote localhost:3333	соединение с сервером OpenOCD по TCP/IP
monitor halt	останов таргета
monitor flash probe 0	проверка флеша
monitor stm32x mass_erase 0	стирание флеша
monitor flash write_image out/image.bin	запись во флеш программы
monitor reset halt	сброс и останов таргета
monitor soft_reset_halt	программный сброс и останов
thbreak main	установка точки прерывания
continue	запуск таргета на исполнение программы

Разумно также иметь конфигурацию типа сброс+отладка, которая позволит выполнить рестарт платы и начать отладку без записи во флеш микроконтроллера. В этом случае в скрипте отсутствуют команды стирания флеш и записи в нее прошивки.



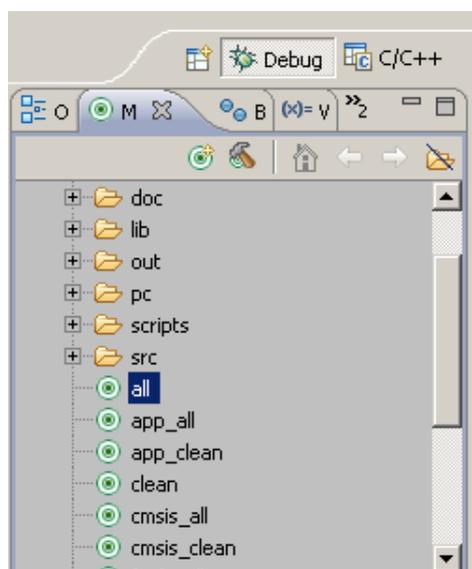
Вкладка *Common* позволяет добавить конфигурацию в быстрое меню.



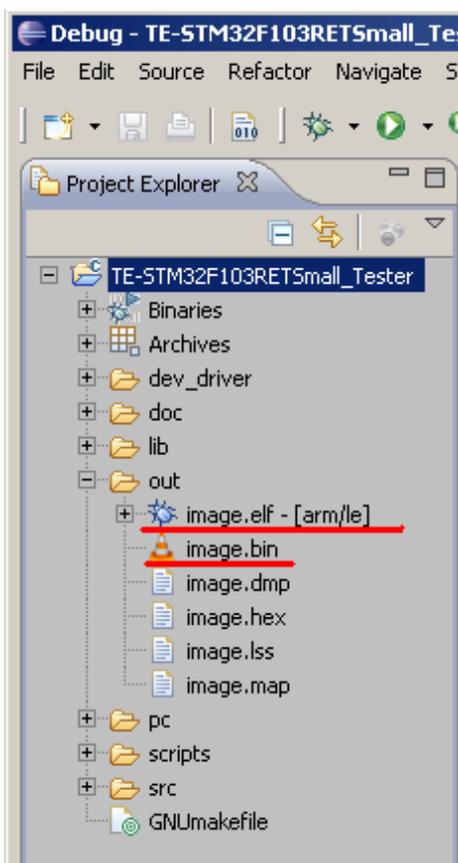
После проведенных манипуляций IDE можно считать готовой к созданию кода проекта, выполнению записи кода во флэш-память микроконтроллера и отладке.

5. Программирование микроконтроллера и отладка проекта

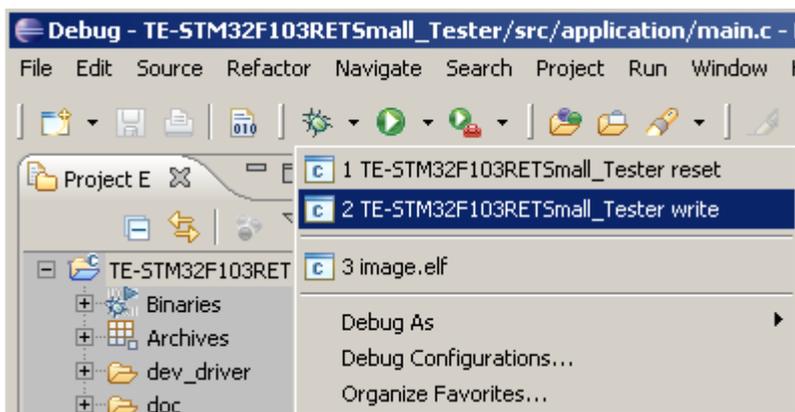
На диске имеется проект TE-STM32RET6Small тестирования модуля TE-STM32F103, который предлагается как пример. Описываемые далее операции будут проводится на примере этого проекта. Вначале соберите проект, выполнив цель clean (очистка проекта), затем all (сборка проекта).



Если сборка успешно завершилась, то в ее результате будут в числе других сгенерированы файлы `image.elf` и бинарный файл прошивки `image.bin`. Первый содержит код программы и отладочную информацию для GDB, второй — образ прошивки.

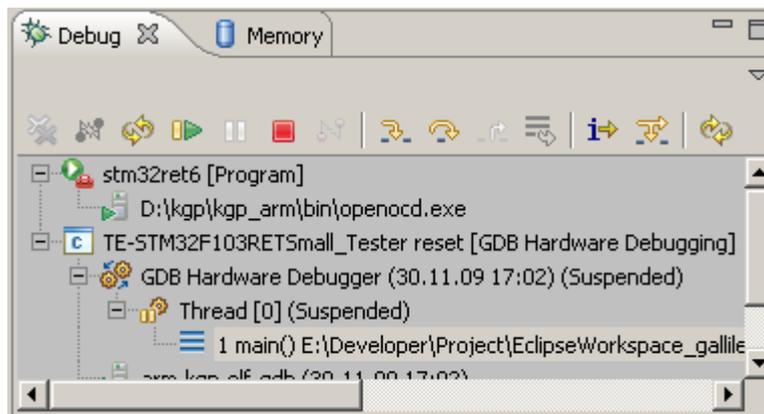


Теперь можно переходить к отладке. Сделайте проект активным и в меню отладки вызовите «TE-STM32F103RETSmall-Tester write».

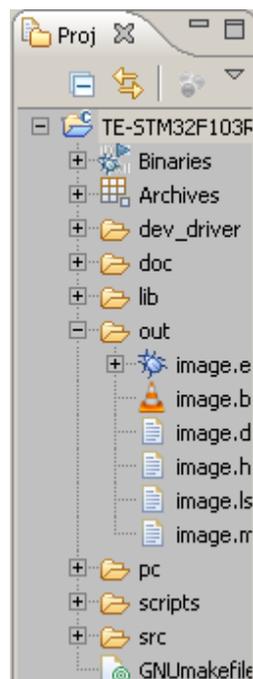


В результате IDE запускает GDB, который через интерфейс MI взаимодействует с IDE, а по TCP/IP с сервером OpenOCD, который обеспечивает связь с платой по JTAG. После исполнения скрипта инициализации (обеспечивает запись прошивки) отладчик останавливает устройство на точке прерывания *main*. На рисунках ниже представлены окна IDE после запуска отладки.

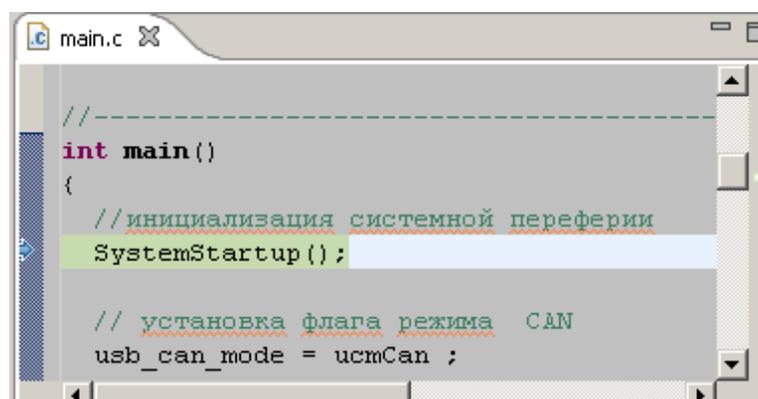
Окно управления отладкой. Элементы управления окна (кнопки) позволяют управлять отладкой и ее режимами. Также окно индуцирует развернутый стек вызовов и список потоков в многопоточных приложениях.



Окно проводника проектов.

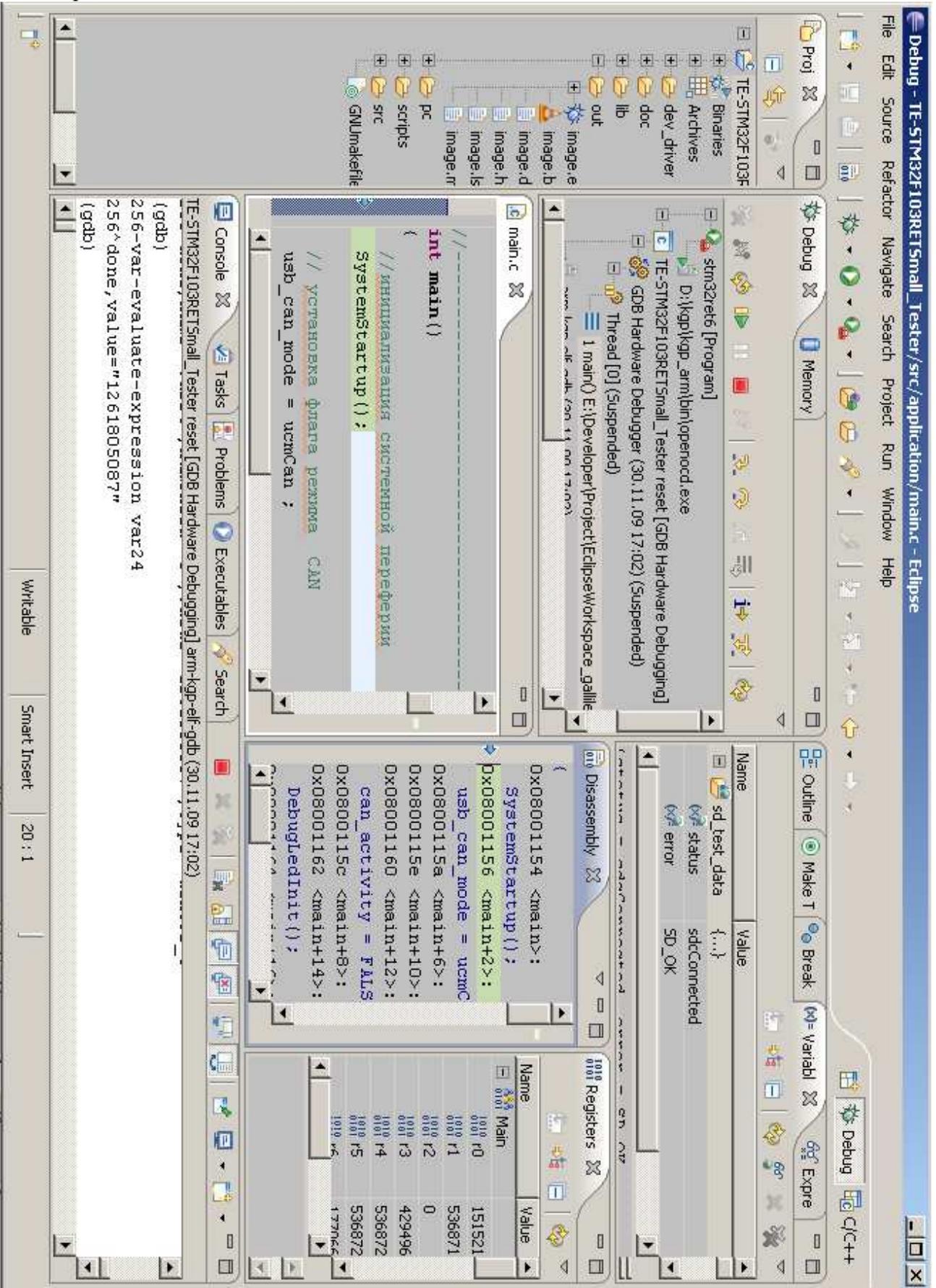


Выполняет навигацию по деревьям файлов открытых проектов.
Окно редактор кода.



Окно дизассемблера.

Ниже представлен полный вид IDE.



Более детальные и тонкие вопросы, связанные с использованием Eclipse IDE предлагается обсудить на форуме www.electronix.ru (см. раздел Web-ссылки)

Приложение Web-ссылки

GNU GCC, <http://gcc.gnu.org>

GNU Binutils, <http://www.gnu.org/software/binutils>

GNU Make, <http://www.gnu.org/software/make>

JRE, <http://java.com/ru/download/index.jsp>

Eclipse IDE for C/C++ Developers, <http://www.eclipse.org/downloads>

KGP for ARM, http://www.klen.org/Files/DevTools/kgp_arm_full_20091116.7z

Sourcery G++ for ARM EABI, <http://www.codesourcery.com>

Olimex ARM-USB-OCD, <http://www.olimex.com/dev/arm-usb-ocd.html>

OpenOCD, <http://openocd.berlios.de/web>

Полезно ознакомиться с содержанием форума <http://electronix.ru> .

В разделе [GNU/OpenSource средства разработки для avr/arm/mips](#) постоянно ведется обсуждение вопросов, связанных с использованием средств, описанных в документе.