

Введение.

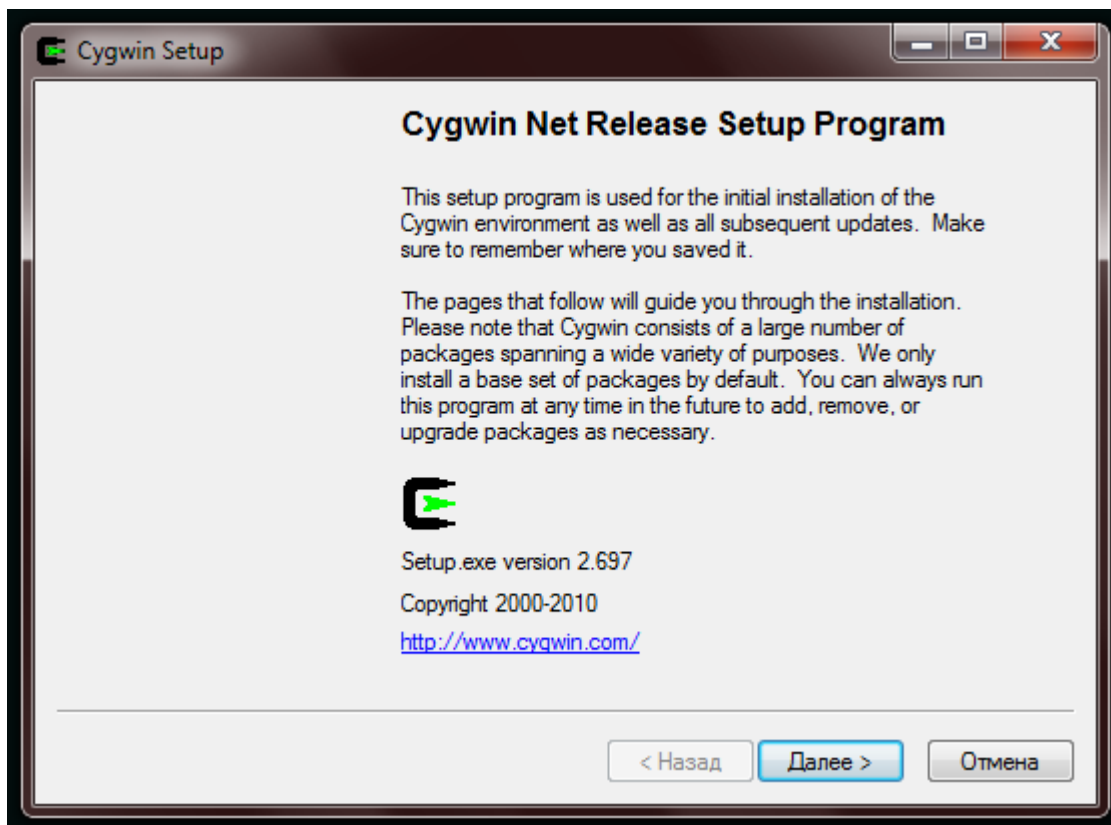
В данном руководстве кратко описывается процесс сборки внутрисхемного отладчика OpenOCD (программа доступна под лицензией GNU GPL, автор — Dominic Rath, официальный сайт <http://openocd.berlios.de>).

1. Установка Cygwin.

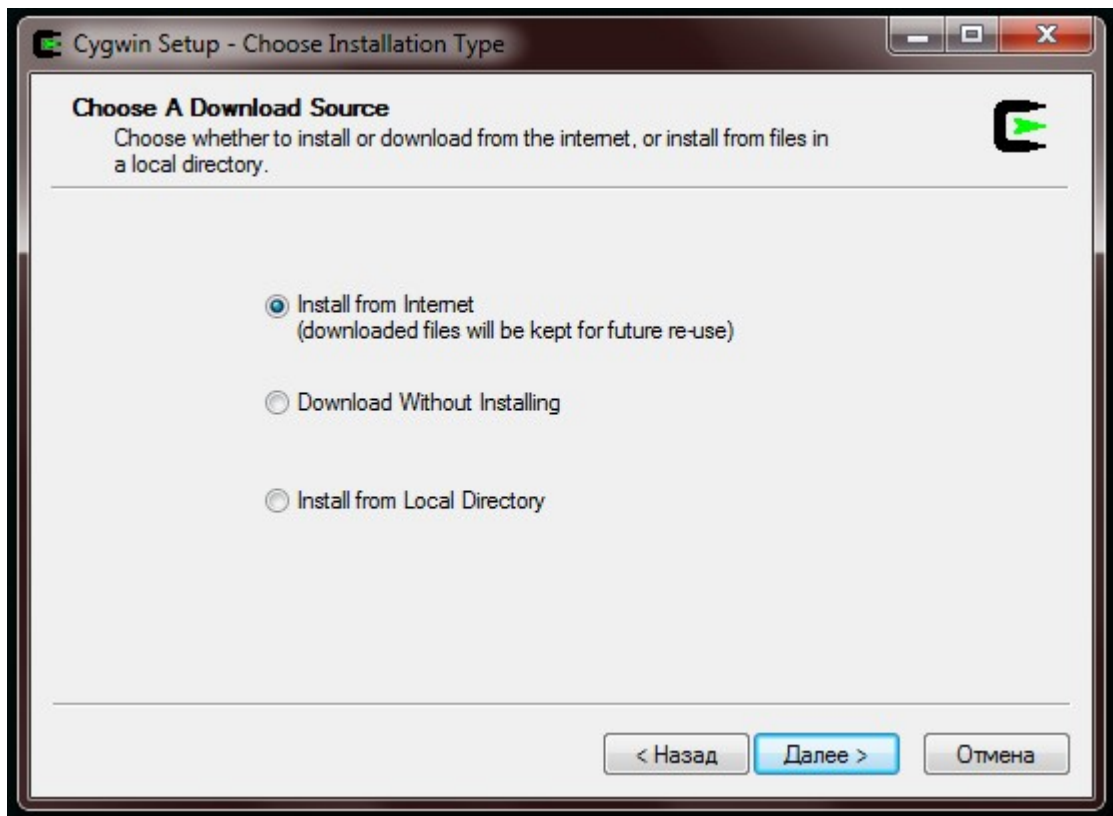
Для сборки будет использован набор свободных программных инструментов Cygwin (пакет доступен под лицензией GNU GPL, официальный сайт <http://www.cygwin.com>).

Для установки пакета необходимо скачать с сайта <http://www.cygwin.com> программу установки ([setup.exe](#)), размер программы несколько мегабайт, основная задача установка (или обновление, уже установленных) программных инструментов Cygwin.

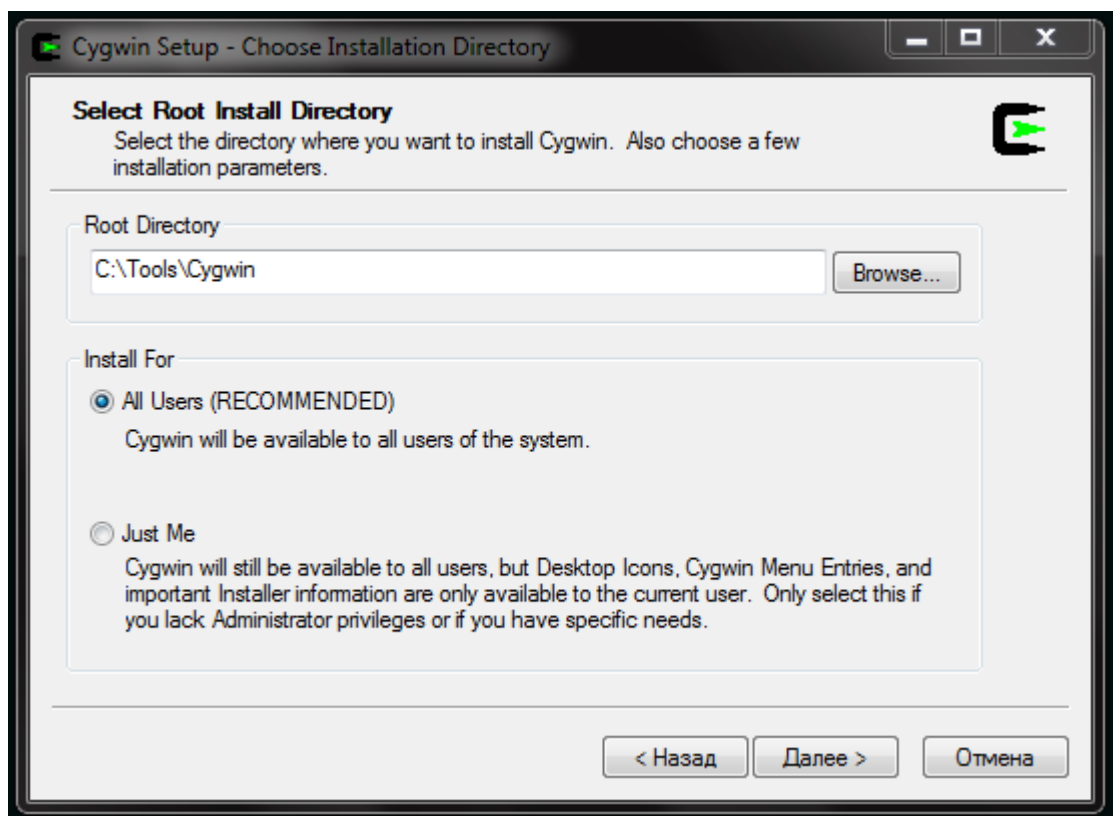
После загрузки программы установки запустите её.



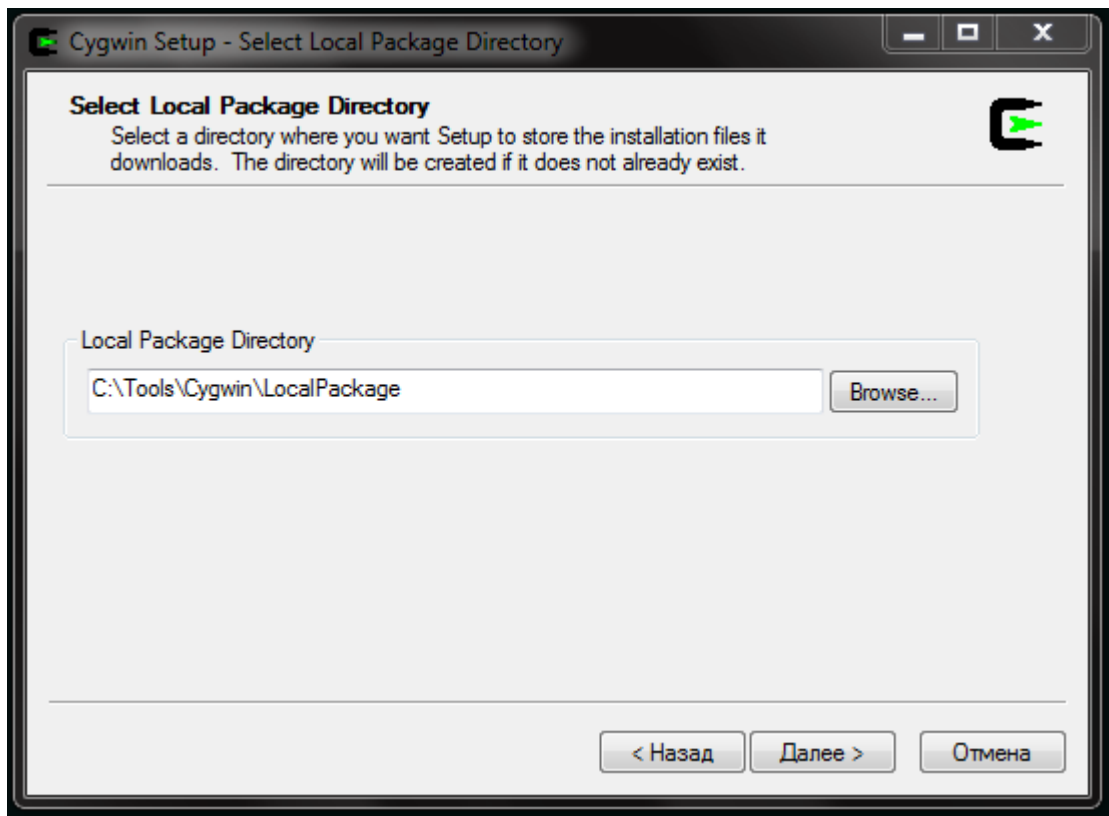
Программа установки позволяет выбрать способ получения необходимых программных инструментов, оптимально устанавливая напрямую из интернета. В случае повторной установки или установки на другом ПК, целесообразно воспользоваться локальными пакетами.



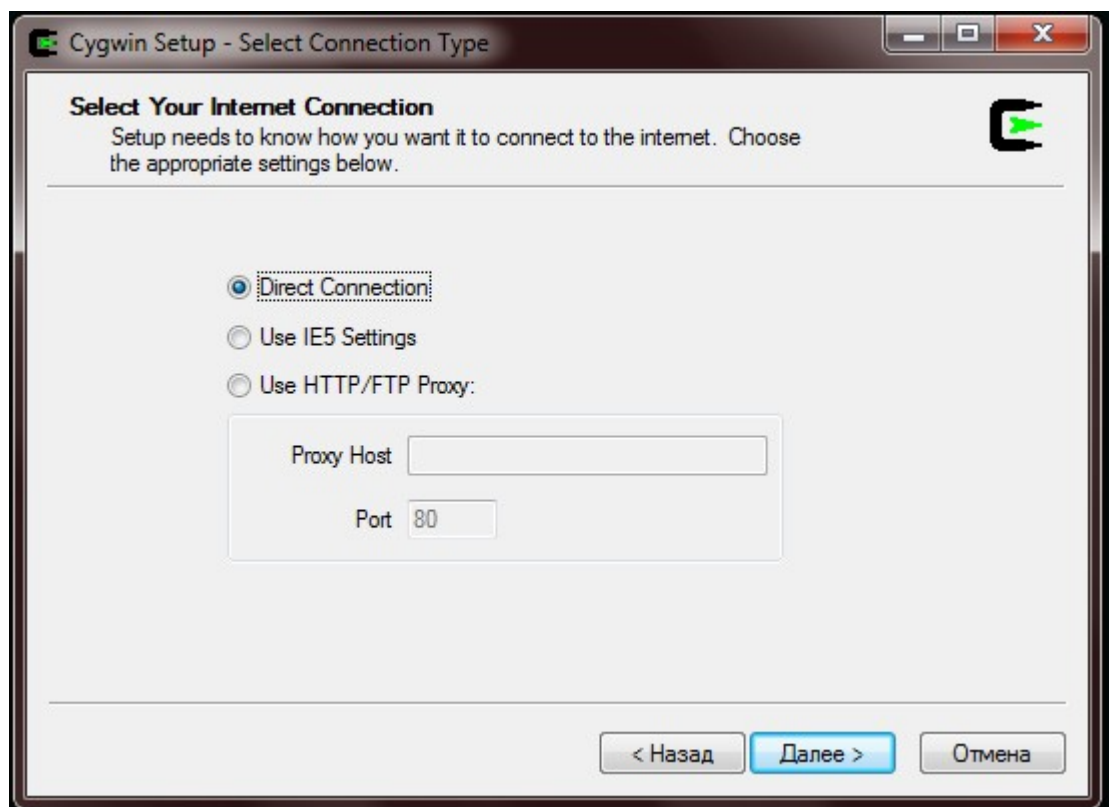
На следующем шаге необходимо выбрать место для установки. Изменим место установки по умолчанию, в указании пути нужно избегать пробелов и кириллических символов.



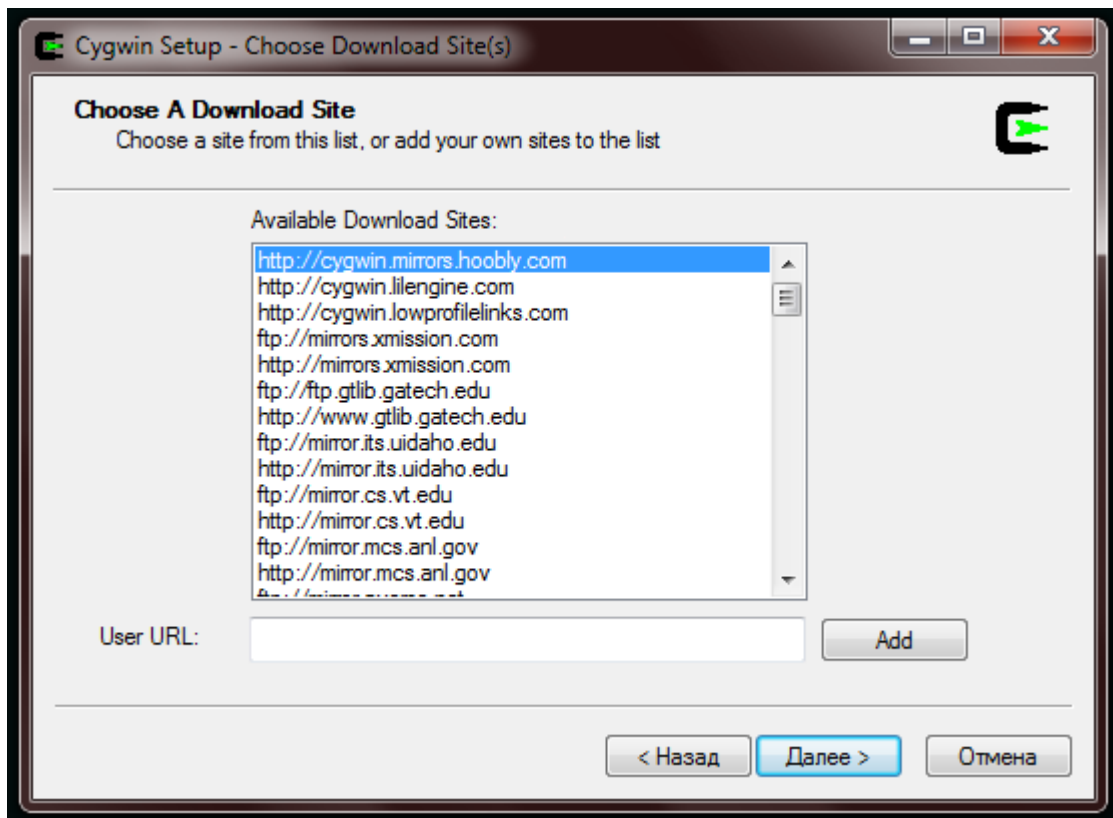
Изменим так же место хранения локальных пакетов. Локальные пакеты потребуются при повторной установке или установке на другом ПК аналогичных программных инструментов Cygwin.



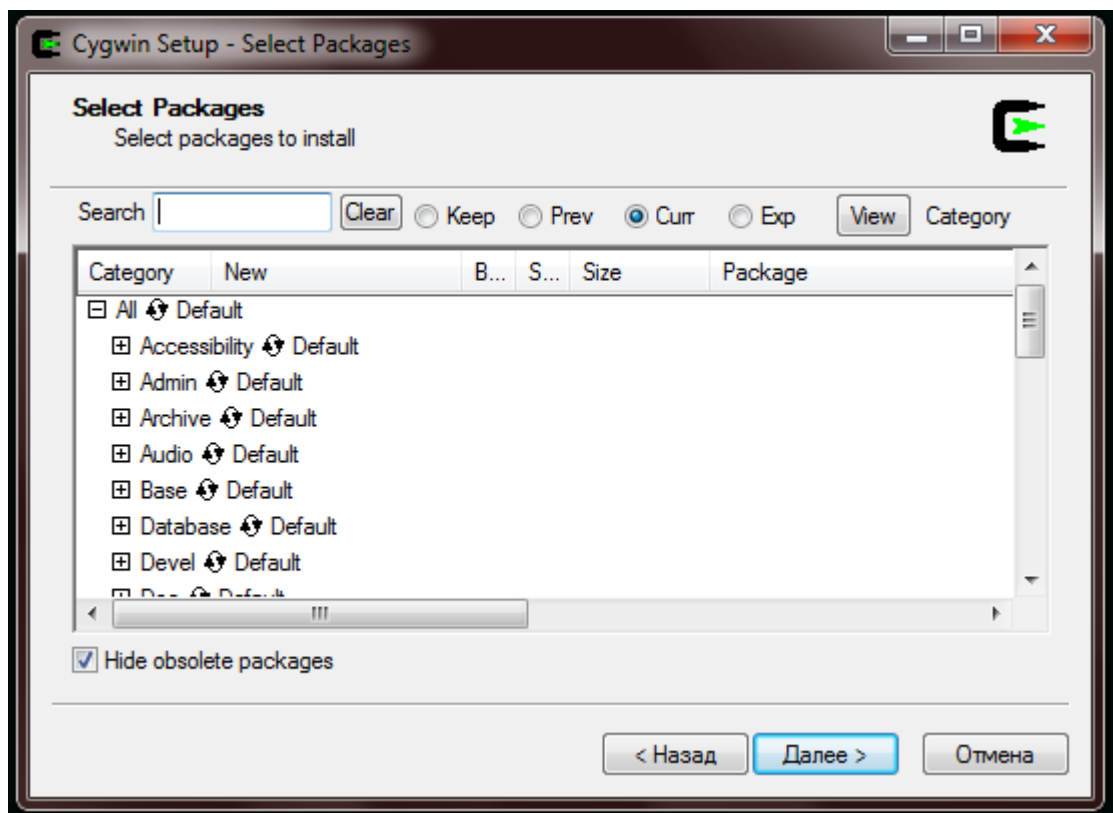
В случае загрузки пакетов посредством интернета работающего не напрямую, а через прокси сервер, необходимо указать соответствующие параметры соединения. Если прокси сервер требует авторизации диалоговое окно для ввода имени и пароля появится на следующем этапе установки.



Так же необходимо указать сайт с которого будут загружены соответствующие пакеты.

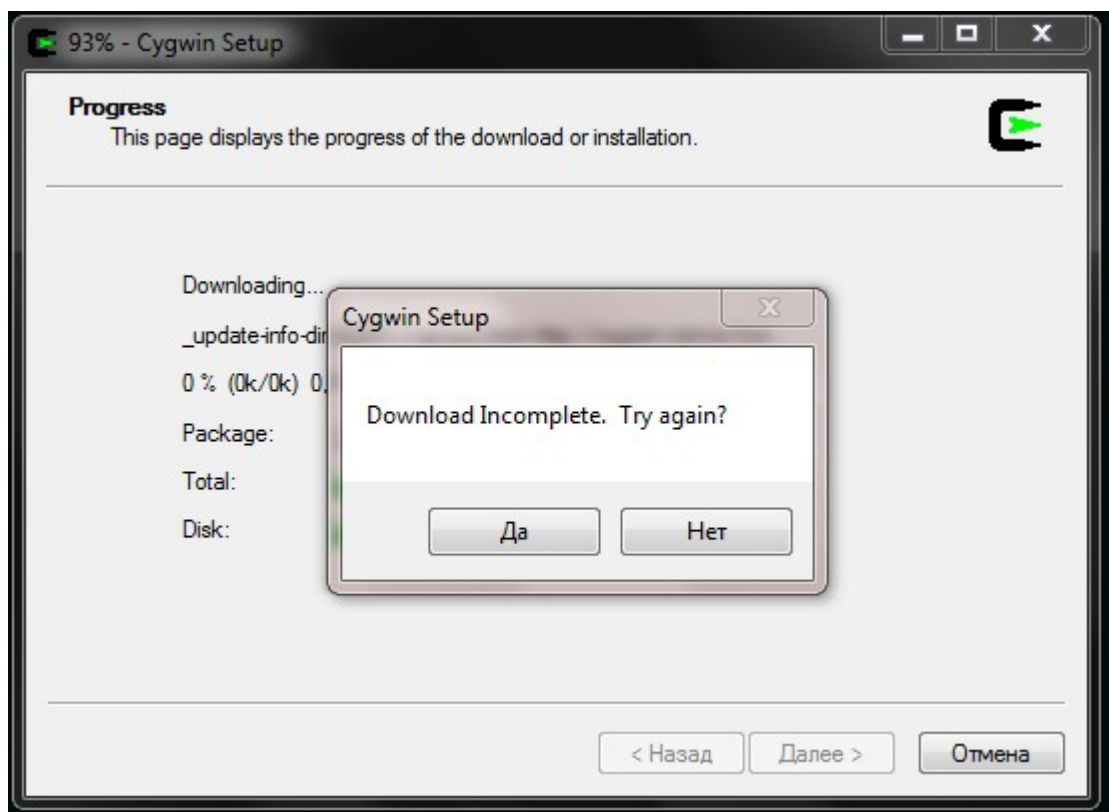
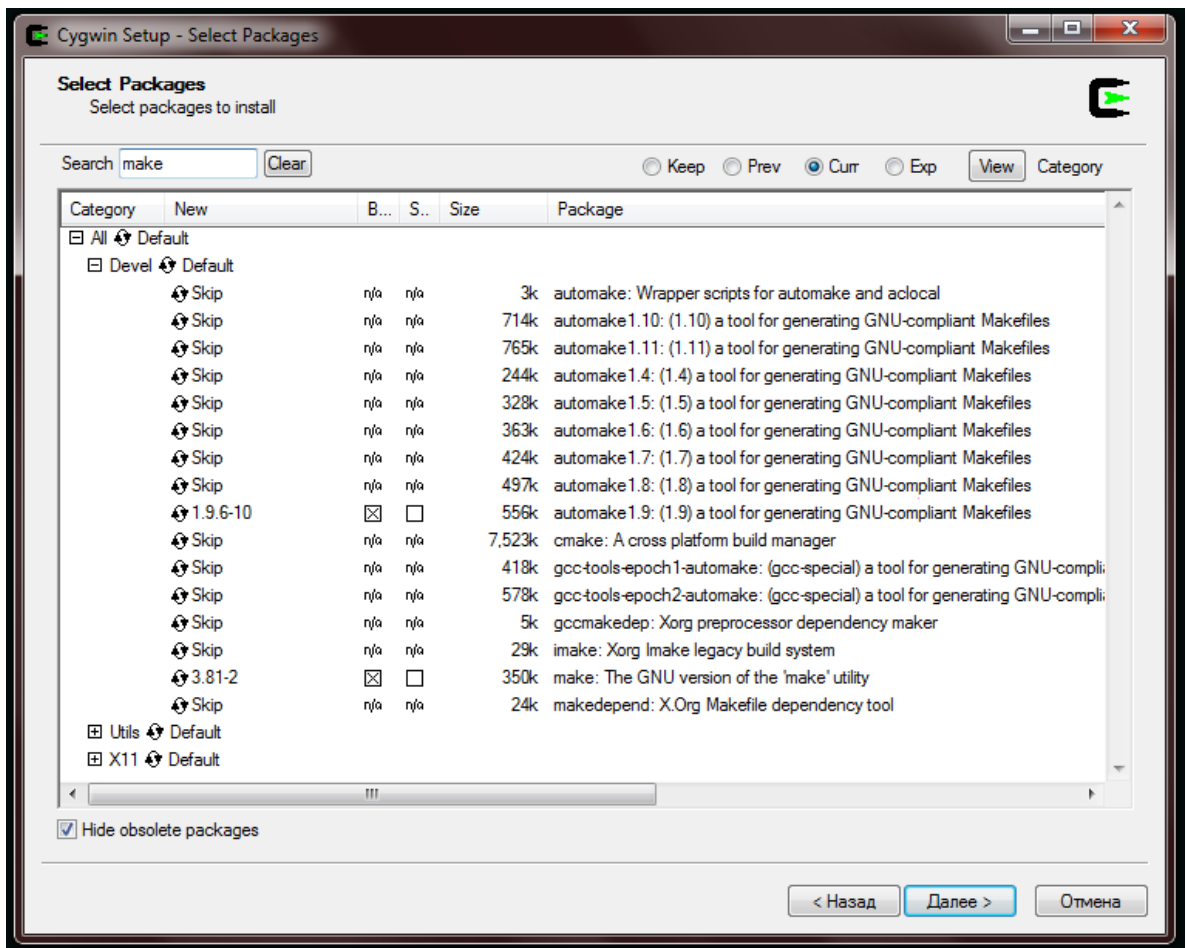


Выбор пакетов установки производить посредством поиска. Для этого необходимо в текстовом поле указать название требуемого пакета.

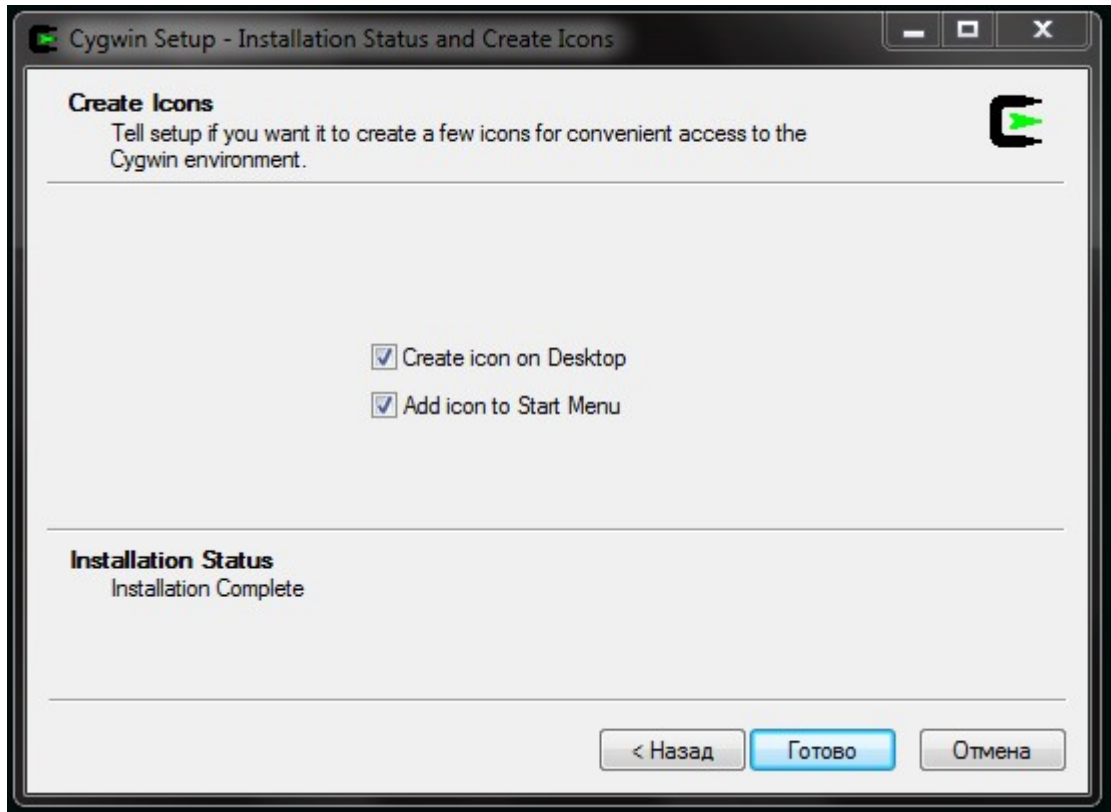


Для успешной сборки потребуются ниже перечисленные пакеты (все пакеты должны быть выбраны из категории «Devel»):

- make; automake; autoconf;
- libtool;
- gcc.



По окончании установки будет предложено создать соответствующие ярлыки.



В дальнейшем при обновлении или установке дополнительных пакетов повторную загрузку программы установки можно не производить, для этого разместим файл программы установки в директории с файлами Cygwin и создадим в меню «Пуск» соответствующий ярлык.

По окончании установки воспользуемся ярлыком «Cygwin Bash Shell» для запуска консольной оболочки.



Выполним проверку необходимых программ. Для этого в консольной оболочке выполним поочередно следующие команды:

- `make -v`
- `gcc -v`
- `automake --version`
- `autoconf --version`

Реакцией на каждую команду будет вывод версии соответствующей программы.

На момент сборки были установлены следующие версии программ:

- make - 3.81
- gcc - 4.3.4
- automake - 1.11.1
- autoconf - 2.65

На этом установку можно считать законченной.

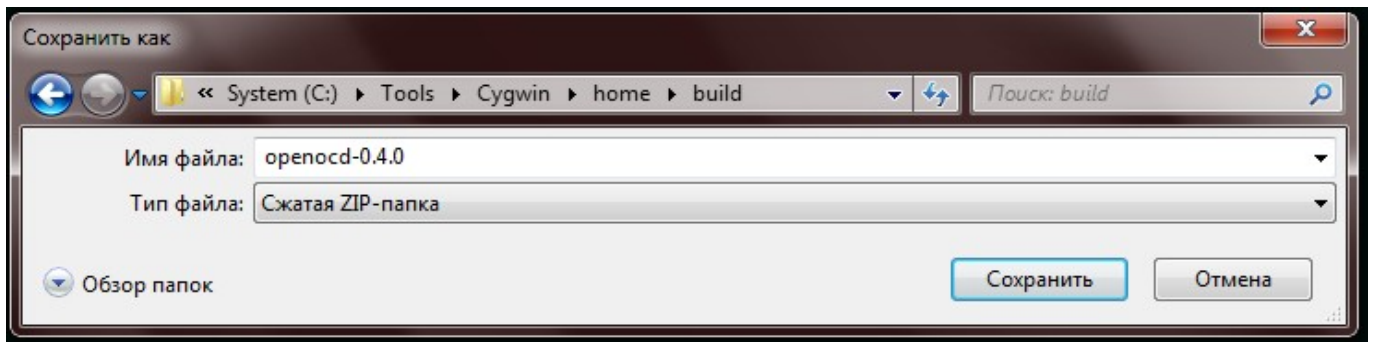
Размер загруженных пакетов на момент установки составил 66,7 Мб.

2. Подготовка к сборке.

Для сборки OpenOCD необходимо скачать с официального сайта <http://openocd.berlios.de> архив с исходным кодом программы.

На момент установки был использован архив с программой OpenOCD версии 0.4.0 <http://downloads.sourceforge.net/project/openocd/openocd/0.4.0/openocd-0.4.0.zip>.

Сохраним архив в директорию «C:\Tools\Cygwin\home\build», предварительно создав папку «build».



После загрузки и сохранения распакуем архив с программой в директорию «C:\Tools\Cygwin\home\build\openocd».

На этом подготовку можно считать законченной.

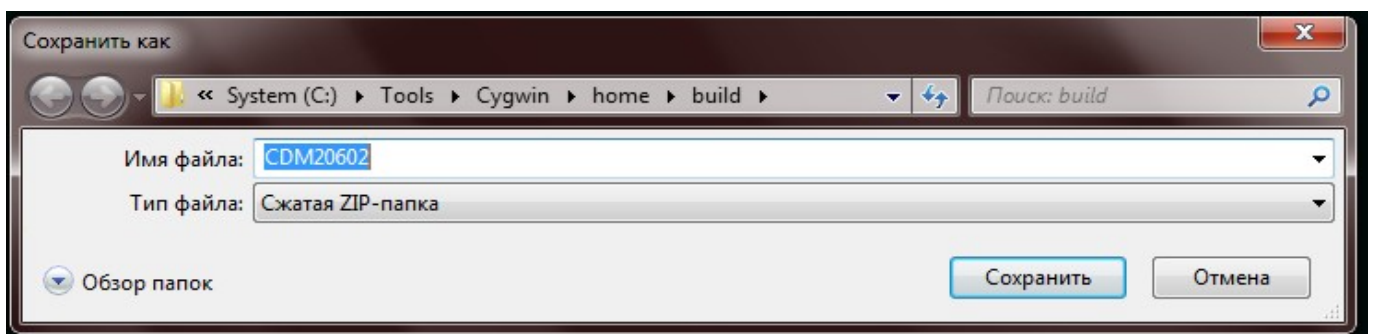
Размер загруженного архива на момент установки составил 1,97 Мб.

2.1. Сборка OpenOCD с модулем для работы JTAG адаптерами на чипе FTDI.

Для сборки необходимо дополнительно скачать драйвера с официального сайта <http://www.ftdichip.com/Drivers/D2XX.htm>.

На момент установки был использован архив с драйверами версии 2.06.02 <http://downloads.sourceforge.net/project/openocd/openocd/0.4.0/openocd-0.4.0.zip>.

Сохраним архив в директорию «C:\Tools\Cygwin\home\build».



После загрузки и сохранения распакуем архив с программой в директорию «C:\Tools\Cygwin\home\build\ftdi», предварительно создав папку «ftdi».

Размер загруженного архива на момент установки составил 1,14 Мб.

Воспользуемся ярлыком «Cygwin Bash Shell» для запуска консольной оболочки.

Для сборки необходимо перейти в директорию с исходными файлами OpenOCD, для этого после запуска консольной оболочки выполним поочередно следующие команды:

- cd ..
- cd build/openocd

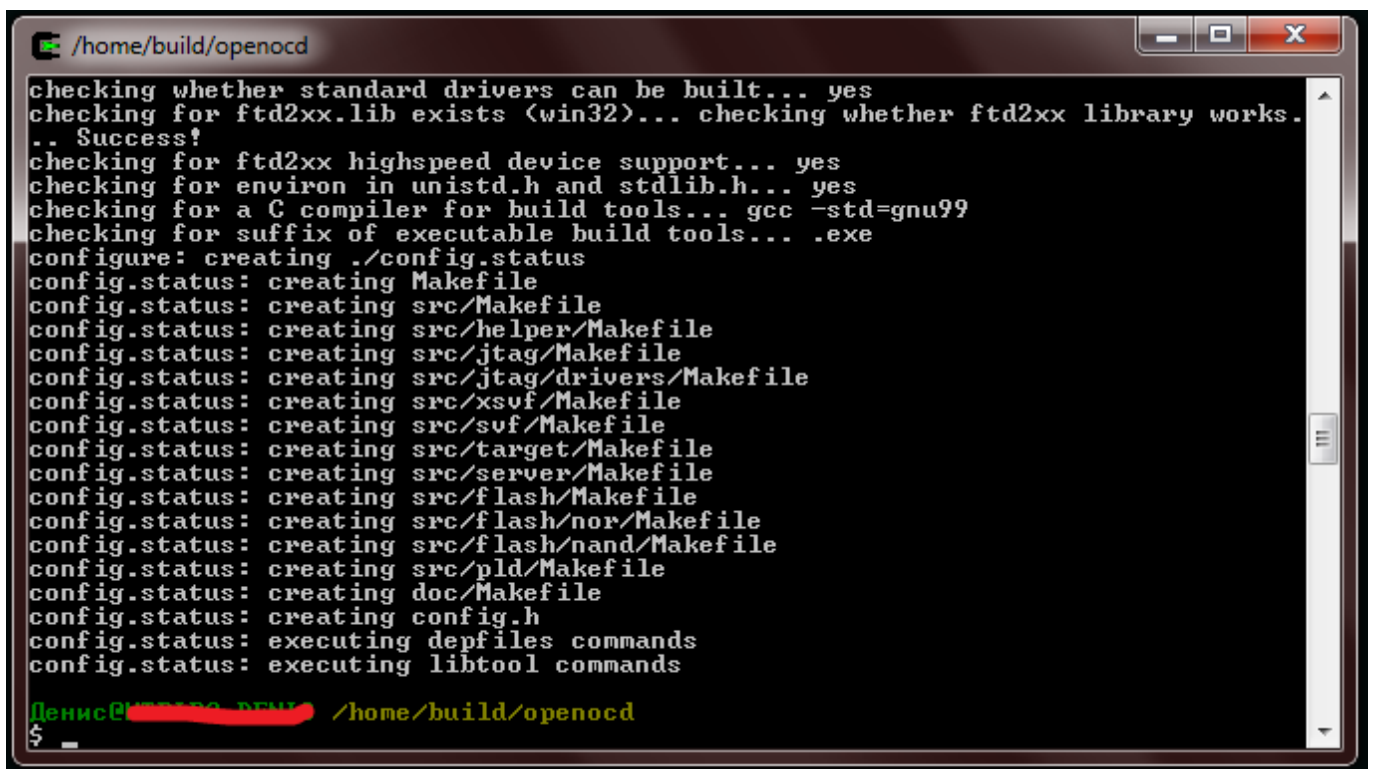
Произведем конфигурирование с ключами соответствующие сборке OpenOCD, выполнив команду:

- ./configure --enable-maintainer-mode --disable-werror --disable-shared --enable-ft2232-ftd2xx --with-ftd2xx-win32-zipdir=/home/build/ftdi



```
 /home/build/openocd
Денис@... ~
$ cd ..
Денис@... /home
$ cd build/openocd
Денис@... /home/build/openocd
$ ./configure --enable-maintainer-mode --disable-werror --disable-shared --enable-ft2232-ftd2xx --with-ftd2xx-win32-zipdir=/home/build/ftdi_
```

При отсутствии ошибок можно переходить к сборке, в случае появления ошибок необходимо анализировать лог файл «configure.log».

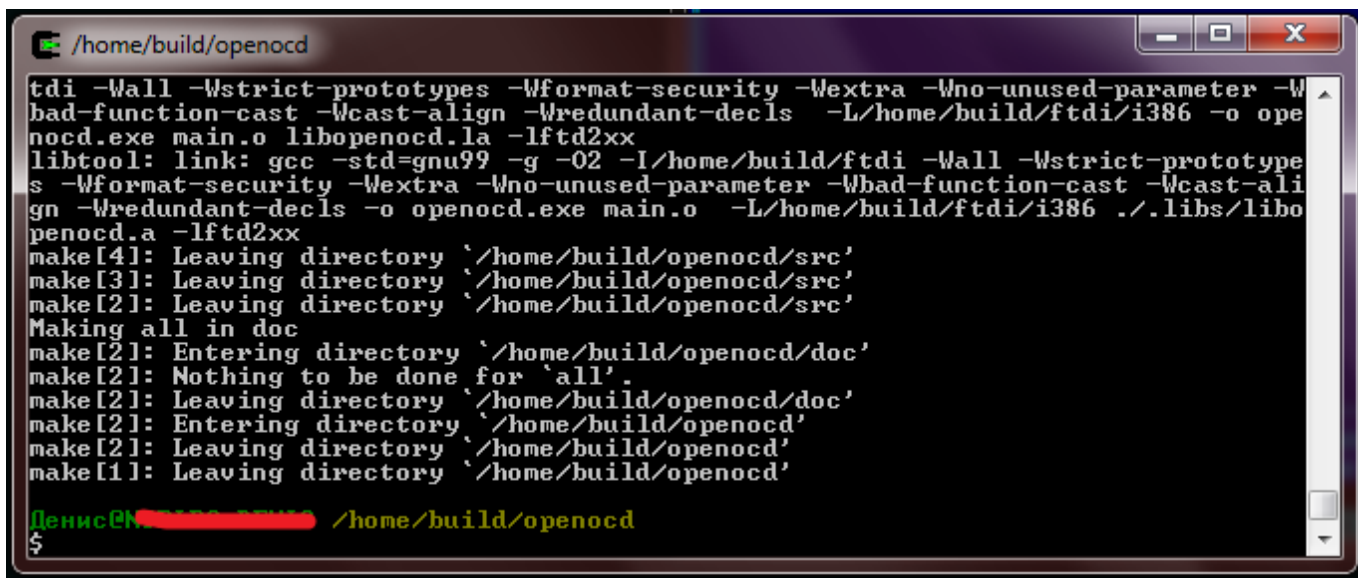


```
 /home/build/openocd
checking whether standard drivers can be built... yes
checking for ftd2xx.lib exists (win32)... checking whether ftd2xx library works.
.. Success!
checking for ftd2xx highspeed device support... yes
checking for environ in unistd.h and stdlib.h... yes
checking for a C compiler for build tools... gcc -std=gnu99
checking for suffix of executable build tools... .exe
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating src/helper/Makefile
config.status: creating src/jtag/Makefile
config.status: creating src/jtag/drivers/Makefile
config.status: creating src/xsvf/Makefile
config.status: creating src/svf/Makefile
config.status: creating src/target/Makefile
config.status: creating src/server/Makefile
config.status: creating src/flash/Makefile
config.status: creating src/flash/nor/Makefile
config.status: creating src/flash/nand/Makefile
config.status: creating src/pld/Makefile
config.status: creating doc/Makefile
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
Денис@... /home/build/openocd
$ _
```


Произведем сборку OpenOCD, выполнив команду:

- make

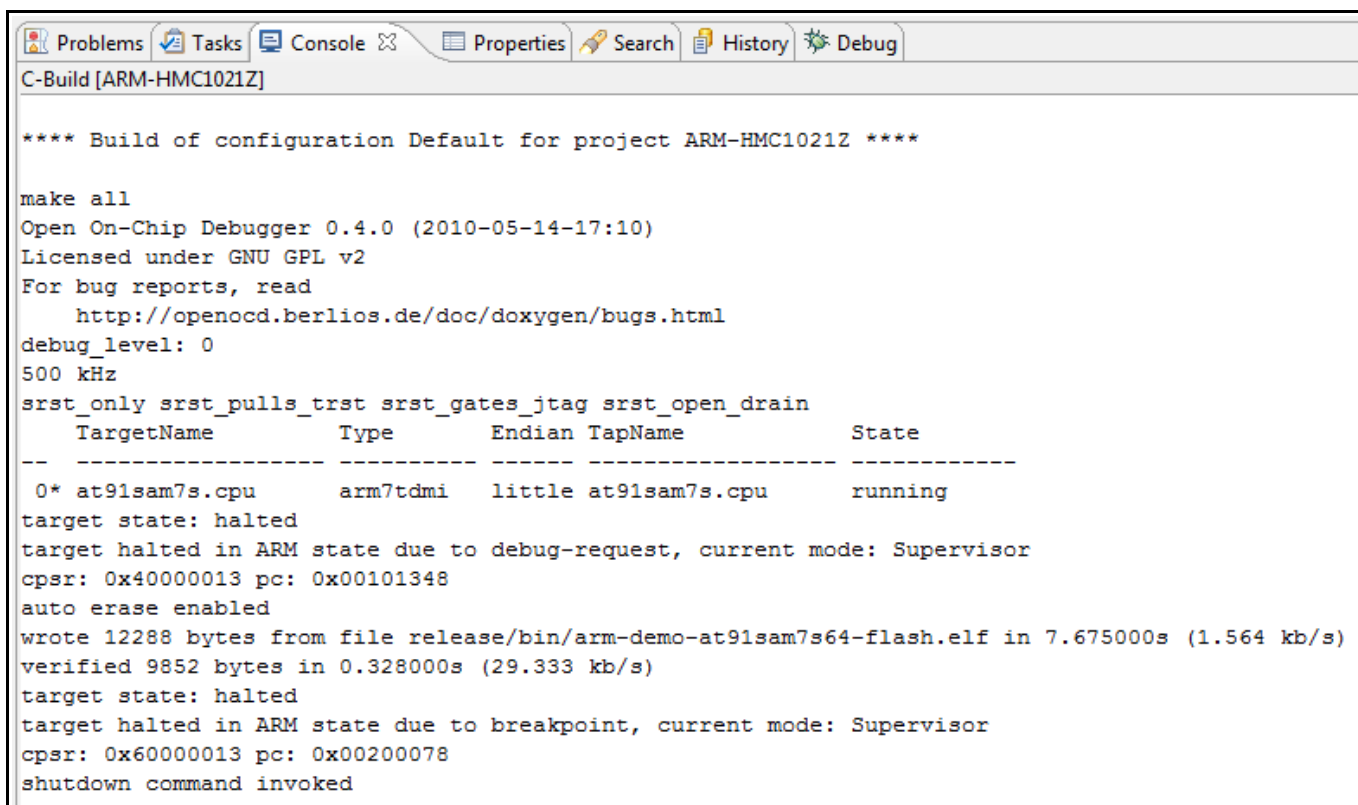
После успешной сборки в директории «C:\Tools\Cygwin\home\build\openocd\src» будет доступен исполняемый файл «openocd.exe», и файлы конфигурации в соответствующих директориях.



```
/home/build/openocd
tdi -Wall -Wstrict-prototypes -Wformat-security -Wextra -Wno-unused-parameter -W
bad-function-cast -Wcast-align -Wredundant-decls -L/home/build/ftdi/i386 -o ope
nocd.exe main.o libopenocd.la -lftd2xx
libtool: link: gcc -std=gnu99 -g -O2 -I/home/build/ftdi -Wall -Wstrict-prototype
s -Wformat-security -Wextra -Wno-unused-parameter -Wbad-function-cast -Wcast-ali
gn -Wredundant-decls -o openocd.exe main.o -L/home/build/ftdi/i386 ./libs/libo
penocd.a -lftd2xx
make[4]: Leaving directory `/home/build/openocd/src'
make[3]: Leaving directory `/home/build/openocd/src'
make[2]: Leaving directory `/home/build/openocd/src'
Making all in doc
make[2]: Entering directory `/home/build/openocd/doc'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/build/openocd/doc'
make[2]: Entering directory `/home/build/openocd'
make[2]: Leaving directory `/home/build/openocd'
make[1]: Leaving directory `/home/build/openocd'

Ленис@... /home/build/openocd
$
```

3. Проверка OpenOCD.



```
Problems Tasks Console Properties Search History Debug
C-Build [ARM-HMC1021Z]

**** Build of configuration Default for project ARM-HMC1021Z ****

make all
Open On-Chip Debugger 0.4.0 (2010-05-14-17:10)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.berlios.de/doc/doxygen/bugs.html
debug_level: 0
500 kHz
srst_only srst_pulls_trst srst_gates_jtag srst_open_drain
  TargetName      Type      Endian TapName      State
-----
0* at91sam7s.cpu  arm7tdmi  little at91sam7s.cpu  running
target state: halted
target halted in ARM state due to debug-request, current mode: Supervisor
cpsr: 0x40000013 pc: 0x00101348
auto erase enabled
wrote 12288 bytes from file release/bin/arm-demo-at91sam7s64-flash.elf in 7.675000s (1.564 kb/s)
verified 9852 bytes in 0.328000s (29.333 kb/s)
target state: halted
target halted in ARM state due to breakpoint, current mode: Supervisor
cpsr: 0x60000013 pc: 0x00200078
shutdown command invoked
```