Ethernut 3 Source Code Debugging

Requirements

This is a short listing only. For Details please refer to the related manuals.

Required Hardware

- Ethernut 3 Board
- Turtelizer 2 JTAG Dongle
- PC with USB and RS-232 port
- 12V Power Supply
- RS-232 DB9 Null Modem Cable
- USB A/B Cable

The first 2 items are available at http://www.egnite.de/en/egnite-shop.html.

Required Software

The following software needs to be installed on the Windows PC:

- Windows XP or later
- Java Runtime Environment 1.6 or later
- Eclipse Platform Runtime Binary 3.6.2 (Helios)
- Eclipse CDT 7.0.2
- TeraTerm or similar Terminal Emulator
- Nut/OS Beta 4.9.10
- Turtelizer OpenOCD Package
- Turtelizer FTDI Driver

You can download the Java Runtime at http://www.oracle.com/technetwork/java/javase/downloads/.

Eclipse is available at http://download.eclipse.org/eclipse/downloads/ and the CDT can be found at http://www.eclipse.org/cdt/. Remember to install the Platform Runtime Binary. You may try Eclipse IDE for C/C++ Developers, which includes the CDT, but this version is not covered here.

TeraTerm can be downloaded at http://ttssh2.sourceforge.jp/.

The last 3 items are available at http://www.ethernut.de/en/download/.

Installation details can be found in the documents that come with the specific product.

Configuring Nut/OS

When using the latest YAGARTO distribution, you must be aware of a significant change. While earlier releases used arm-elf-gcc, YAGARTO is now using arm-none-eabi-gcc. Nut/OS 4.9.10 has been prepared to make this switch easy. Open the file *nut/Makevars.arm-gcc* with your text editor. Notepad will be fine. Locate the line

```
#TRGT = arm-none-eabi-
```

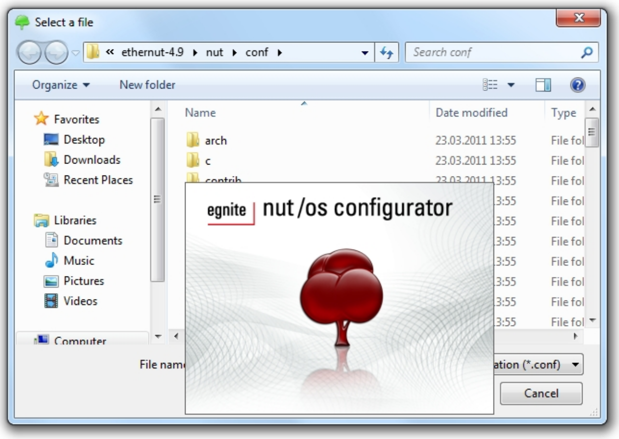
and remove the leading numero sign:

```
TRGT = arm-none-eabi-
```

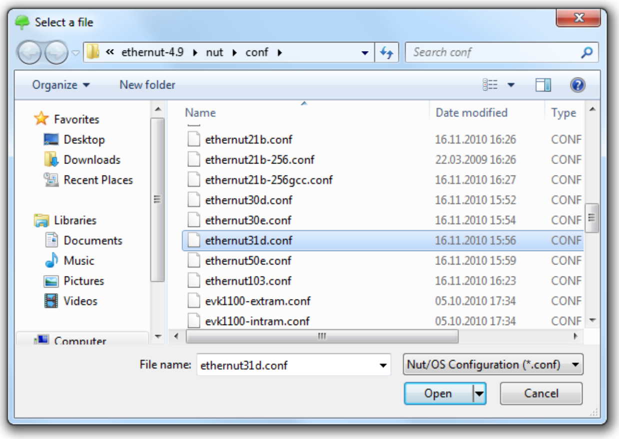
Save your changes.

In order to include debugging information in the compiled binaries you need to switch from arm-gcc to arm-gccdbg. When done, rebuild Nut/OS and recreate the application directory. The following steps will give detailed instructions.

When started for the first time, the Nut/OS Configurator will display a file selection dialog. If this is hidden by the splash window, simply click on that window to close it.



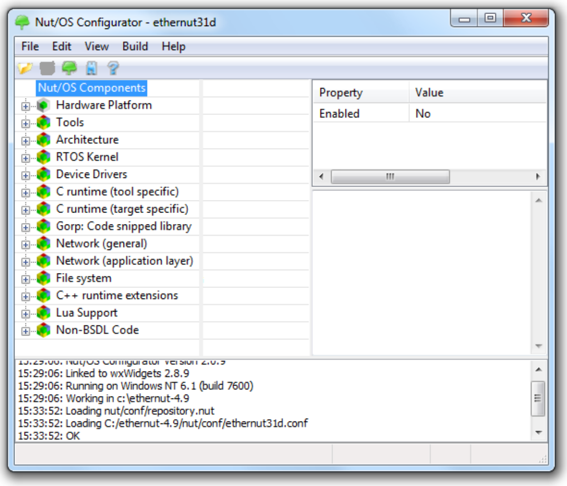
Select the file *ethernut30e.cfg* for Ethernut 3.0 boards or *ethernut31c.cfg* for Ethernut 3.1 boards and click open



Name	Date modified	Type
ethernut21b.conf	16.11.2010 16:26	CONF
ethernut21b-256.conf	22.03.2009 16:26	CONF
ethernut21b-256gcc.conf	16.11.2010 16:27	CONF
ethernut30d.conf	16.11.2010 15:52	CONF
ethernut30e.conf	16.11.2010 15:54	CONF
ethernut31d.conf	16.11.2010 15:56	CONF
ethernut50e.conf	16.11.2010 15:59	CONF
ethernut103.conf	16.11.2010 16:23	CONF
evk1100-extram.conf	05.10.2010 17:34	CONF
evk1100-intram.conf	05.10.2010 17:34	CONF

File name: ethernut31d.conf
Nut/OS Configuration (*.conf)

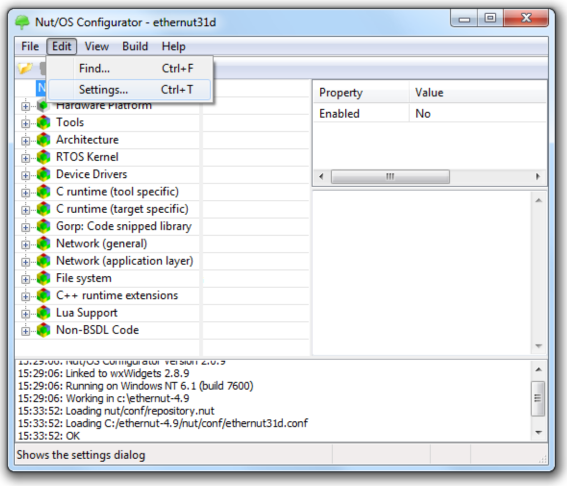
The Configurators main window will show the Nut/OS Components in a tree structure on the right side.


Nut/OS Configurator - ethernut31d
File Edit View Build Help
Nut/OS Components
Hardware Platform
Tools
Architecture
RTOS Kernel
Device Drivers
C runtime (tool specific)
C runtime (target specific)
Gorp: Code snipped library
Network (general)
Network (application layer)
File system
C++ runtime extensions
Lua Support
Non-BSDL Code
Property
Value

Property	Value
Enabled	No

```
15:29:06: Linked to wxWidgets 2.8.9
15:29:06: Running on Windows NT 6.1 (build 7600)
15:29:06: Working in c:\ethernut-4.9
15:33:52: Loading nut/conf/repository.nut
15:33:52: Loading C:/ethernut-4.9/nut/conf/ethernut31d.conf
15:33:52: OK
```

From the *Edit* menu select *Settings...*.

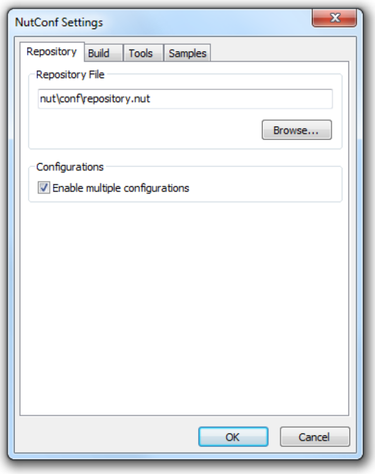


Property	Value
Enabled	No

```
15:29:06: Linked to wxWidgets 2.8.9
15:29:06: Running on Windows NT 6.1 (build 7600)
15:29:06: Working in c:\ethernut-4.9
15:33:52: Loading nut/conf/repository.nut
15:33:52: Loading C:/ethernut-4.9/nut/conf/ethernut31d.conf
15:33:52: OK
```

Shows the settings dialog

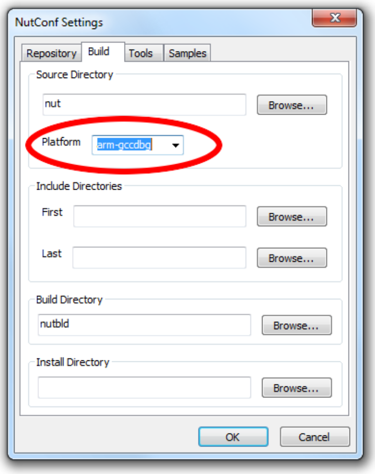
Nothing needs to be changed on the *Repository* page of the settings dialog. You may check *Enable multiple configurations* if you are concurrently working with different target boards.



Repository File: nut\conf\repository.nut

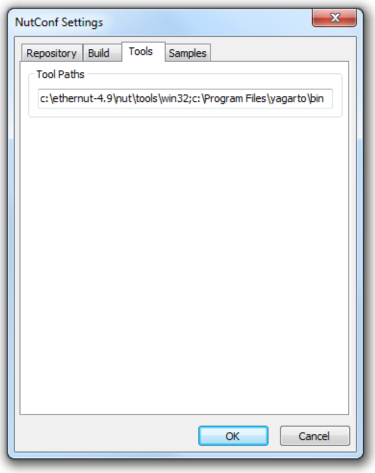
Configurations: [✓ Enable multiple configurations]

Select *arm-gccdbg* as the platform on the *Build* page.



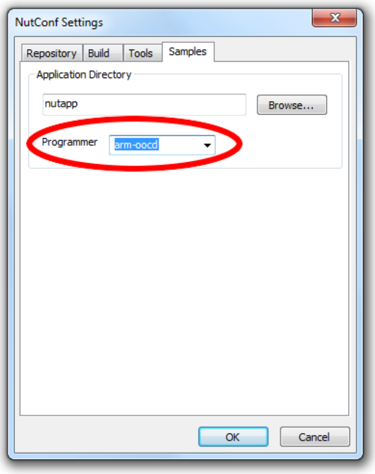
Source Directory: nut
Platform: arm-gccdbg
Include Directories
First:
Last:
Build Directory: nutbld
Install Directory:

On the *Tools* page you need to enter the path to all required tools, separated by semicolons. For the Ethernut 3 board this should point to the Nut/OS win32 tools directory and the YAGARTO bin directory. While all other settings use normal slashes in paths, here we must use backslashes.



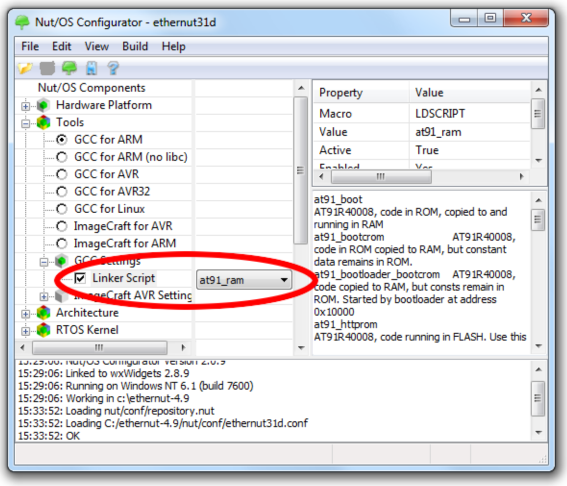
Tool Paths: c:\ethernut-4.9\nut\tools\win32;c:\Program Files\yagarto\bin

On the last page titled *Samples* you may select *arm-oocd* as the programmer of choice, but this is only required for programming the flash memory, not for debugging in RAM.



Application Directory: nutapp
Programmer: arm-oocd

Before building the system, change the *Linker Script* in the components tree under *Tools/GCC Settings* to *at91_ram*.

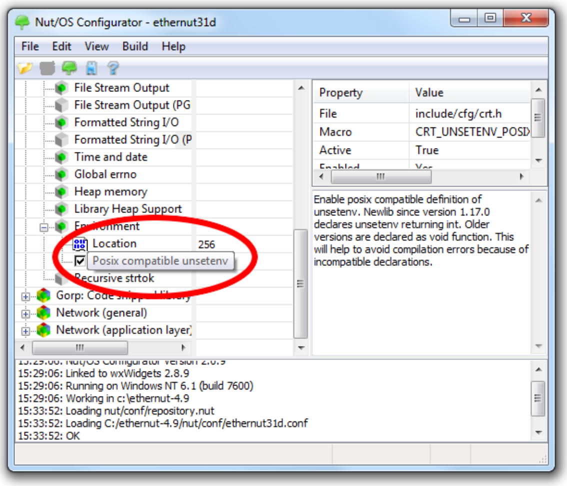


Property	Value
Macro	LDSCRIPT
Value	at91_ram
Active	True

at91_boot
AT91R40008, code in ROM, copied to and running in RAM
at91_bootcrom AT91R40008, code in ROM copied to RAM, but constant data remains in ROM.
at91_bootloader_bootcrom AT91R40008, code copied to RAM, but consts remain in ROM. Started by bootloader at address 0x10000
at91_httprom
AT91R40008, code running in FLASH. Use this

```
15:29:06: Linked to wxWidgets 2.8.9
15:29:06: Running on Windows NT 6.1 (build 7600)
15:29:06: Working in c:\ethernut-4.9
15:33:52: Loading nut/conf/repository.nut
15:33:52: Loading C:/ethernut-4.9/nut/conf/ethernut31d.conf
15:33:52: OK
```

We are using YAGARTO with Newlib 1.19. Make sure that *Posix compatible unsetenv* is checked under *C runtime (target specific)/Environment*.

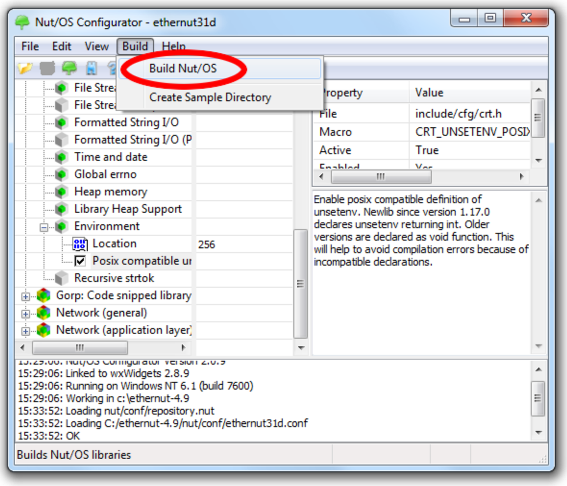


Property	Value
File	include/cfg/crt.h
Macro	CRT_UNSETENV_POSIX
Active	True

Enable posix compatible definition of unsetenv. Newlib since version 1.17.0 declares unsetenv returning int. Older versions are declared as void function. This will help to avoid compilation errors because of incompatible declarations.

```
15:29:06: Linked to wxWidgets 2.8.9
15:29:06: Running on Windows NT 6.1 (build 7600)
15:29:06: Working in c:\ethernut-4.9
15:33:52: Loading nut/conf/repository.nut
15:33:52: Loading C:/ethernut-4.9/nut/conf/ethernut31d.conf
15:33:52: OK
```

Now we are ready to build Nut/OS for debugging. Select *Build Nut/OS* in the *Build* menu.



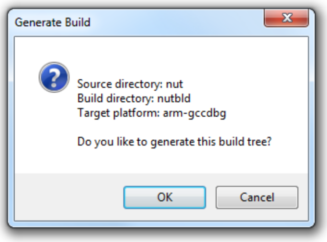
Property	Value
File	include/cfg/crt.h
Macro	CRT_UNSETENV_POSIX
Active	True

Enable posix compatible definition of unsetenv. Newlib since version 1.17.0 declares unsetenv returning int. Older versions are declared as void function. This will help to avoid compilation errors because of incompatible declarations.

```
15:29:06: Linked to wxWidgets 2.8.9
15:29:06: Running on Windows NT 6.1 (build 7600)
15:29:06: Working in c:\ethernut-4.9
15:33:52: Loading nut/conf/repository.nut
15:33:52: Loading C:/ethernut-4.9/nut/conf/ethernut31d.conf
15:33:52: OK
```

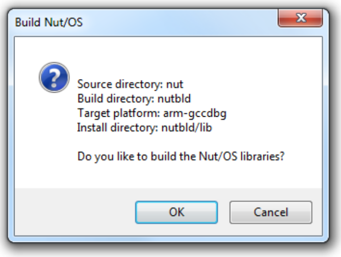
Builds Nut/OS libraries

A message box shows the settings for creating the build directory. Verify them and click OK to build this directory.



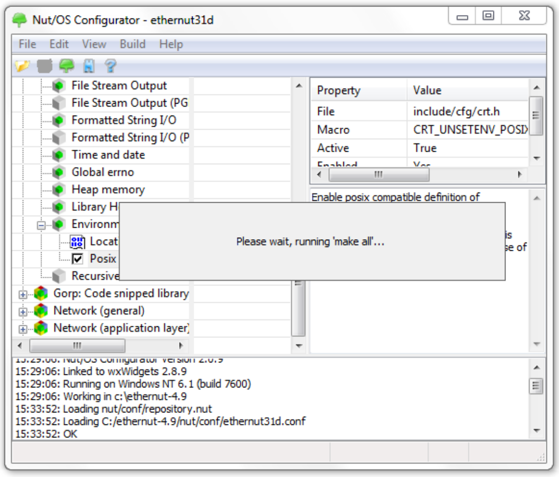
```
Source directory: nut
Build directory: nutbld
Target platform: arm-gccdbg

Do you like to generate this build tree?
```

A second message box shows the settings for building the system. Again verify them and click OK.



```
Source directory: nut
Build directory: nutbld
Target platform: arm-gccdbg
Install directory: nutbld/lib

Do you like to build the Nut/OS libraries?
```

The build may take a few minutes.



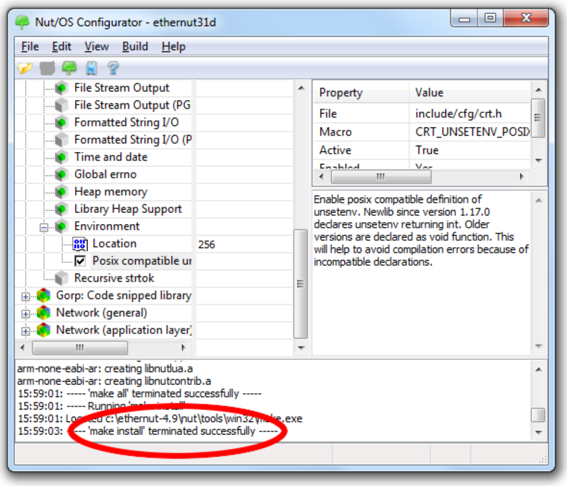
Property	Value
File	include/cfg/crt.h
Macro	CRT_UNSETENV_POSIX
Active	True

Enable posix compatible definition of

Please wait, running 'make all'...

```
15:29:06: Linked to wxWidgets 2.8.9
15:29:06: Running on Windows NT 6.1 (build 7600)
15:29:06: Working in c:\ethernut-4.9
15:33:52: Loading nut/conf/repository.nut
15:33:52: Loading C:/ethernut-4.9/nut/conf/ethernut31d.conf
15:33:52: OK
```

When done, make sure that it has been terminated successfully.

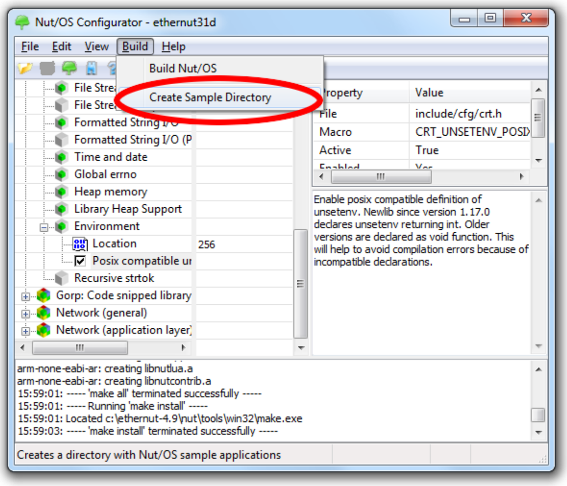


Property	Value
File	include/cfg/crt.h
Macro	CRT_UNSETENV_POSIX
Active	True

Enable posix compatible definition of unsetenv. Newlib since version 1.17.0 declares unsetenv returning int. Older versions are declared as void function. This will help to avoid compilation errors because of incompatible declarations.

```
arm-none-eabi-ar: creating libnutlua.a
arm-none-eabi-ar: creating libnutcontrib.a
15:59:01: ----- 'make all' terminated successfully -----
15:59:01: ----- Running 'make install'
15:59:01: Located c:\ethernut-4.9\nut\tools\win32\make.exe
15:59:03: ----- 'make install' terminated successfully -----
```

Nut/OS uses a separate directory for the application code. A template can be created by selecting *Create Sample Directory*, containing all Nut/OS samples that are included in the distribution. Remember, that we will use the events sample.



Property	Value
File	include/cfg/crt.h
Macro	CRT_UNSETENV_POSIX
Active	True
Enabled	Yes

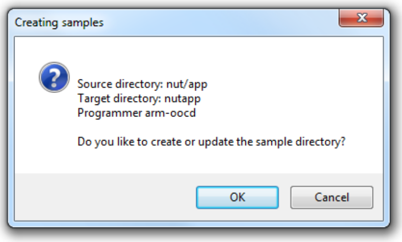
Location: 256

Enable posix compatible definition of unsetenv. Newlib since version 1.17.0 declares unsetenv returning int. Older versions are declared as void function. This will help to avoid compilation errors because of incompatible declarations.

```
arm-none-eabi-ar: creating libnutlua.a
arm-none-eabi-ar: creating libnutcontrib.a
15:59:01: ----- 'make all' terminated successfully -----
15:59:01: ----- Running 'make install' -----
15:59:01: Located c:\ethernut-4.9\nut\tools\win32\make.exe
15:59:03: ----- 'make install' terminated successfully -----
```

Creates a directory with Nut/OS sample applications

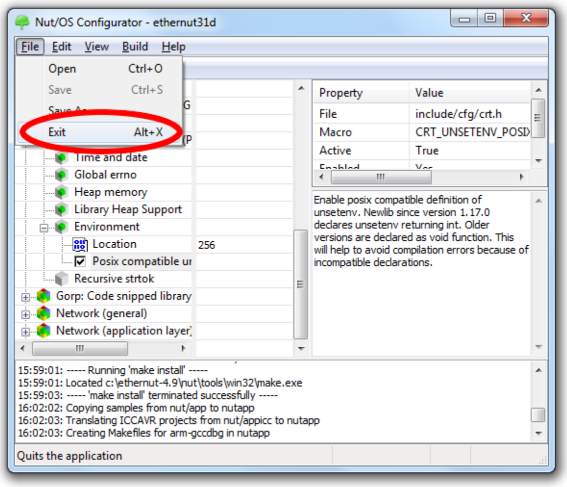
A message box will show the current settings. Click OK, if everything looks as expected.



Source directory: nut/app
Target directory: nutapp
Programmer arm-oocd

Do you like to create or update the sample directory?

When done, you may exit the Configurator by selecting *Exit* in the *File* menu.



Property	Value
File	include/cfg/crt.h
Macro	CRT_UNSETENV_POSIX
Active	True
Enabled	Yes

Location: 256

Enable posix compatible definition of unsetenv. Newlib since version 1.17.0 declares unsetenv returning int. Older versions are declared as void function. This will help to avoid compilation errors because of incompatible declarations.

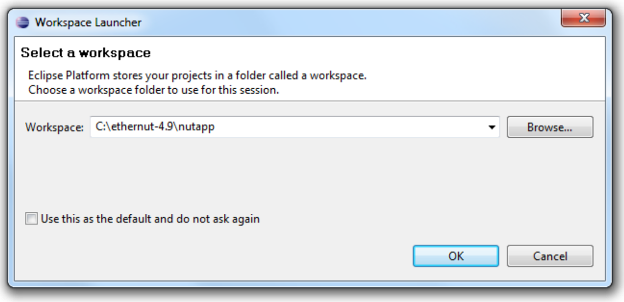
```
15:59:01: ----- Running 'make install' -----
15:59:01: Located c:\ethernut-4.9\nut\tools\win32\make.exe
15:59:03: ----- 'make install' terminated successfully -----
16:02:02: Copying samples from nut/app to nutapp
16:02:03: Translating ICCAVR projects from nut/appicc to nutapp
16:02:03: Creating Makefiles for arm-gccdbg in nutapp
```

Quits the application

Configuring Eclipse

It is assumed that you successfully installed Eclipse and the CDT plug-in. For a first demonstration we will use the events sample that is included in the Nut/OS distribution.

When started, Eclipse will ask for the Workspace Path. Choose the Nut/OS sample directory, e.g. C:\Ethernut-4.9\nutapp.



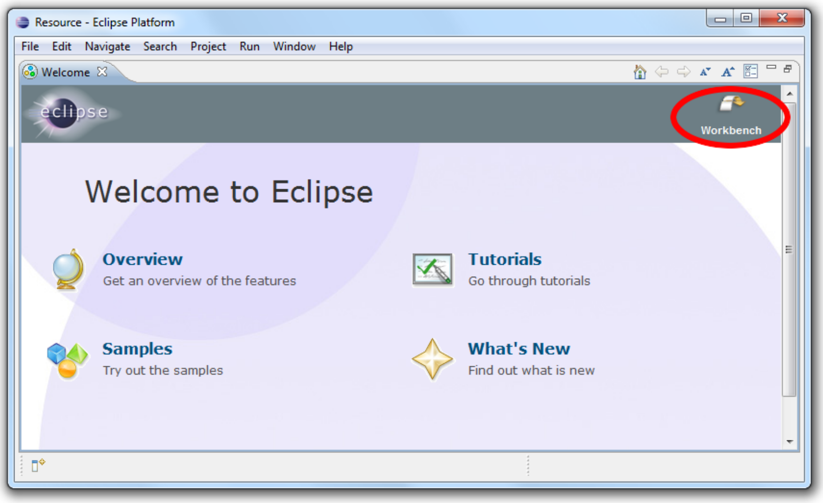
Workspace Launcher

Select a workspace

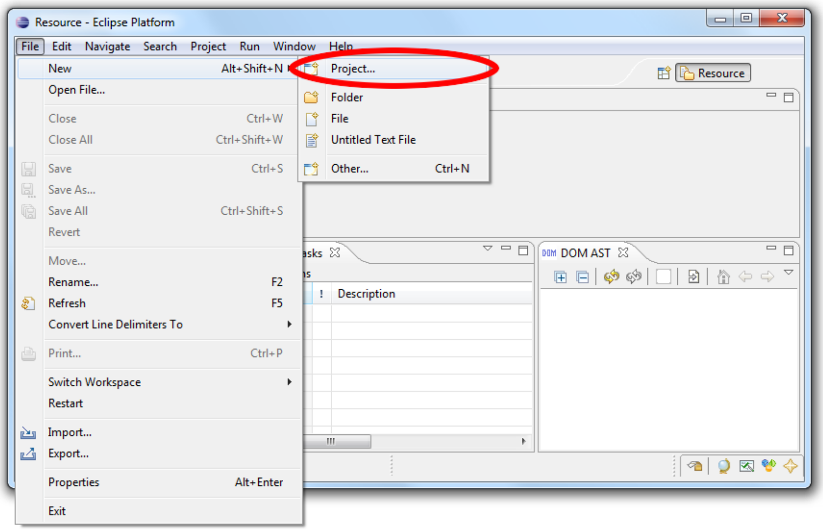
Eclipse Platform stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\ethernut-4.9\nutapp

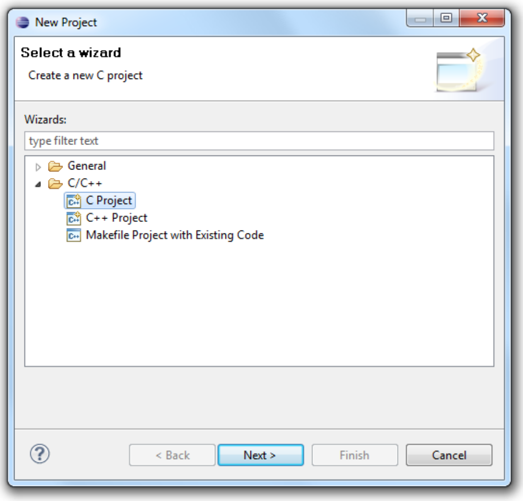
The welcome page appears. Click on *Workbench* in the upper right corner.



The so called *Resource Perspective* or, depending on what else had been installed on your computer, the *Java Perspective* appears. We will later change to the C Perspective.



In the New Project dialog choose C Project and click on Next.


New Project

Select a wizard

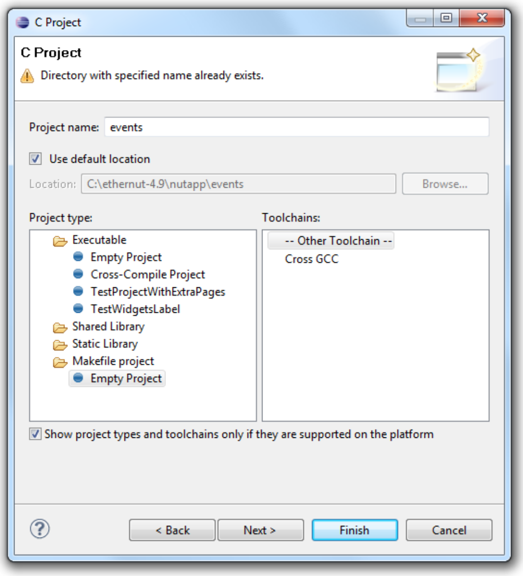
Create a new C project

Wizards:

type filter text

- General
- C/C++
  - C Project
  - C++ Project
  - Makefile Project with Existing Code

In the C Project dialog enter events for the project name. Then choose Empty Project and -- Other Toolchain --. Again click on Next.


C Project

C Project

Directory with specified name already exists.

Project name: events

☑ Use default location

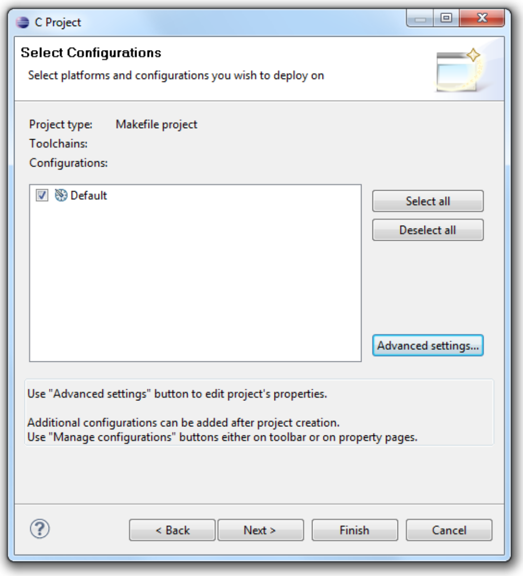
Location: C:\ethernut-4.9\nutapp\events

Project type:
- Executable
  - Empty Project
  - Cross-Compile Project
  - TestProjectWithExtraPages
  - TestWidgetsLabel
- Shared Library
- Static Library
- Makefile project
  - Empty Project

Toolchains:
- -- Other Toolchain --
- Cross GCC

☑ Show project types and toolchains only if they are supported on the platform

On the next page click on Advanced settings....


C Project

Select Configurations

Select platforms and configurations you wish to deploy on

Project type: Makefile project

Toolchains:

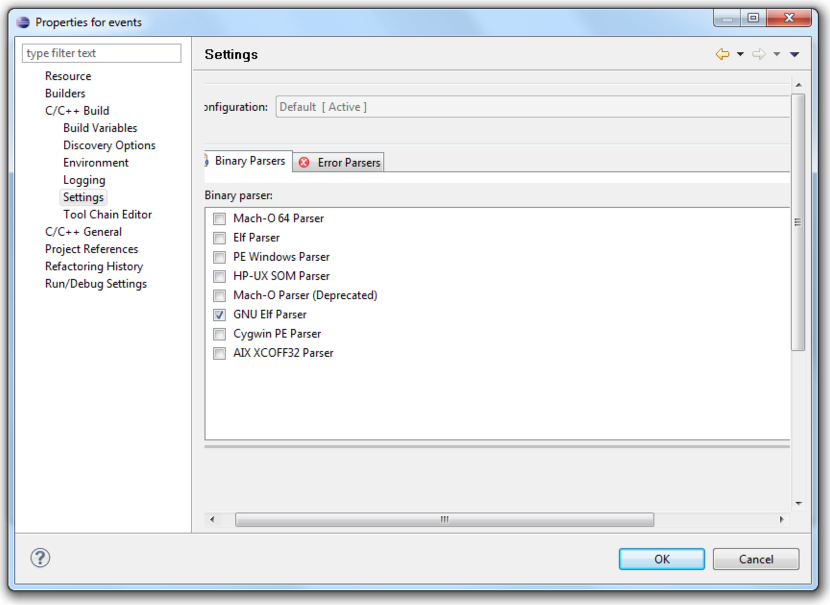
Configurations:

☑ Default

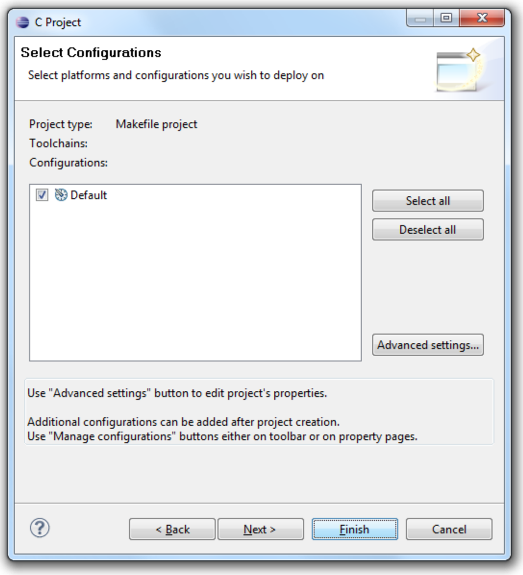
Use "Advanced settings" button to edit project's properties.

Additional configurations can be added after project creation.
Use "Manage configurations" buttons either on toolbar or on property pages.

In the Project Properties page select Settings and check GNU Elf Parser. Then click on OK.



Back in the C Project dialog click Finish.



Select Configurations

Select platforms and configurations you wish to deploy on

Project type: Makefile project

Toolchains:

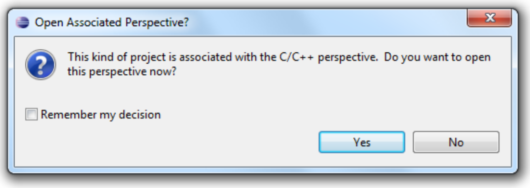
Configurations:

☑ Default

Use "Advanced settings" button to edit project's properties.

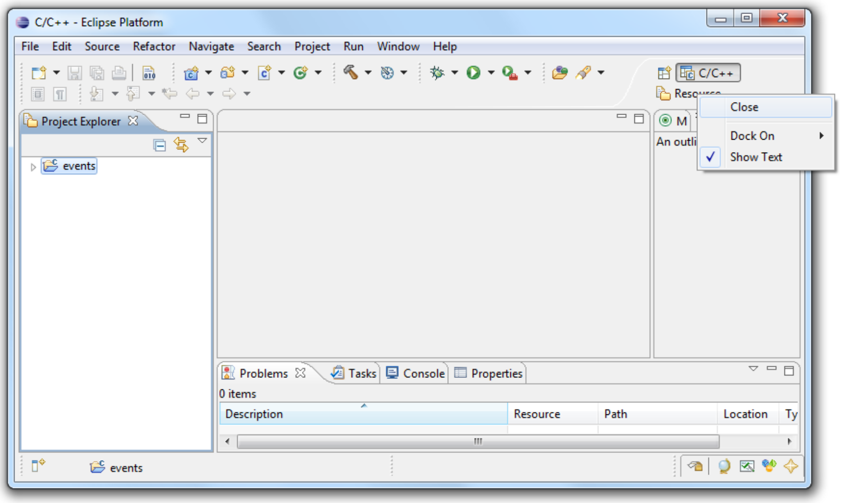
Additional configurations can be added after project creation.
Use "Manage configurations" buttons either on toolbar or on property pages.

Click Yes to open the C/C++ perspective.

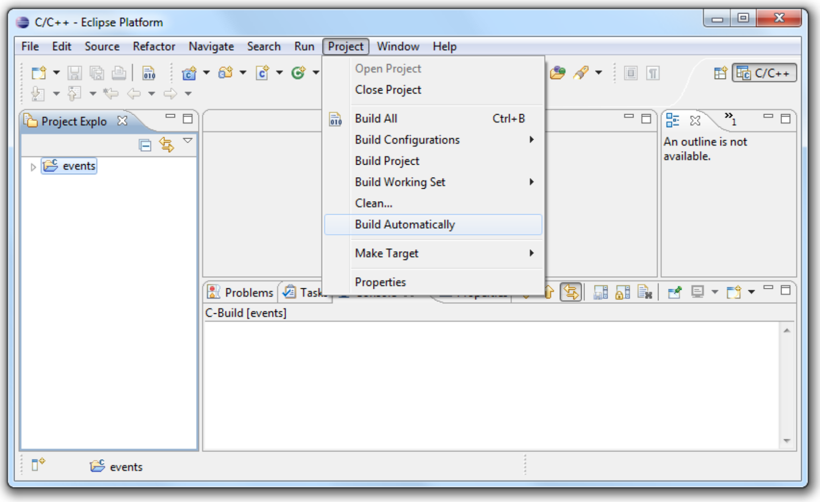


This kind of project is associated with the C/C++ perspective. Do you want to open this perspective now?

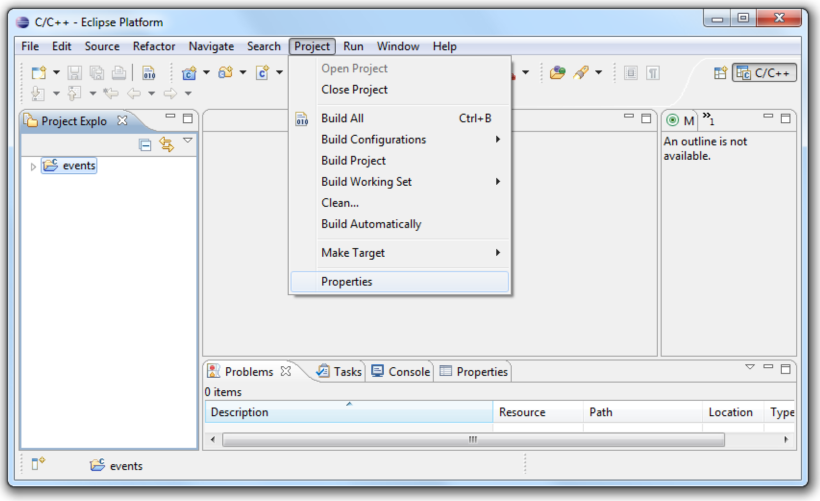
We do not need the *Resource* or *Java* Perspective, therefore you should close it.



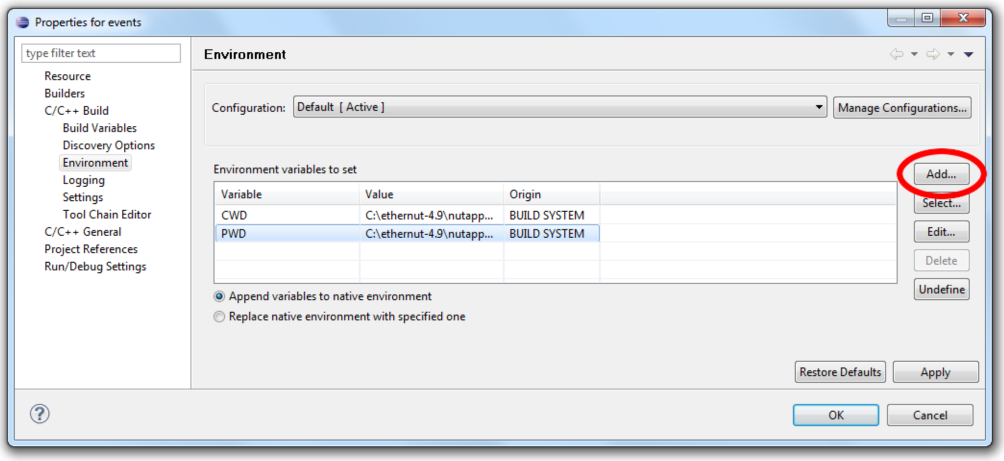
Uncheck *Build Automatically* in the *Project* menu.



In the same menu select *Properties*.



In the Properties dialog select Environment in the tree on the right. Then click Add.


Properties for events

type filter text

- Resource
- Builders
- C/C++ Build
  - Build Variables
  - Discovery Options
  - Environment
  - Logging
  - Settings
  - Tool Chain Editor
- C/C++ General
- Project References
- Run/Debug Settings

Environment

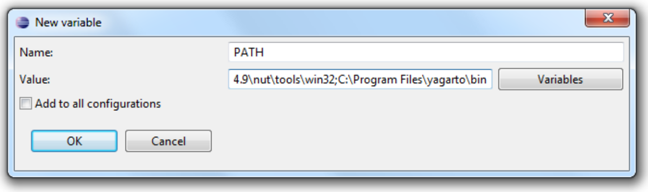
Configuration: Default [Active]

Environment variables to set

Variable	Value	Origin
CWD	C:\ethernut-4.9\nutapp...	BUILD SYSTEM
PWD	C:\ethernut-4.9\nutapp...	BUILD SYSTEM

[✓ Append variables to native environment]
Replace native environment with specified one

The variable *Name* must be `PATH`. You need to enter the path to all required tools as the *Value*, separated by semicolons. The first path should point to the Nut/OS win32 tools directory and the second one to the YAGARTO bin directory.

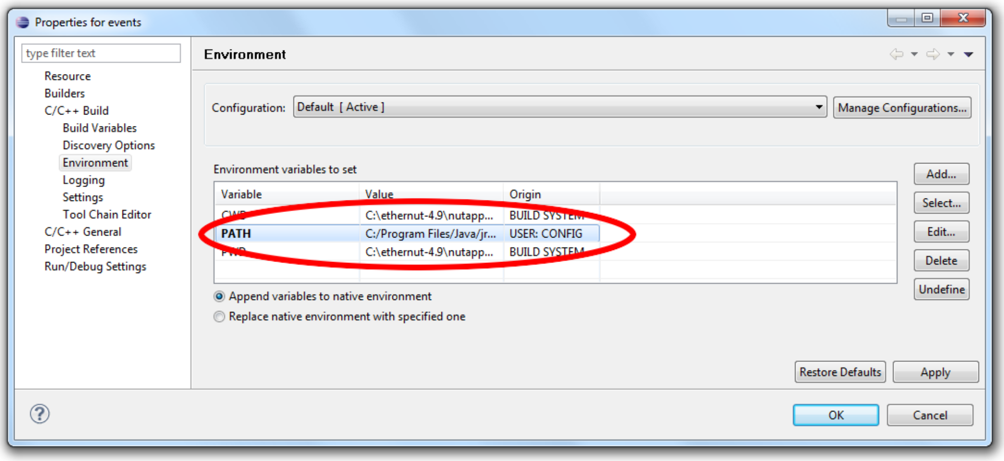

New variable

Name: PATH

Value: 4.9\nut\tools\win32;C:\Program Files\yagarto\bin

Add to all configurations

Click OK to store the updated Environment.


Properties for events

type filter text

- Resource
- Builders
- C/C++ Build
  - Build Variables
  - Discovery Options
  - Environment
  - Logging
  - Settings
  - Tool Chain Editor
- C/C++ General
- Project References
- Run/Debug Settings

Environment

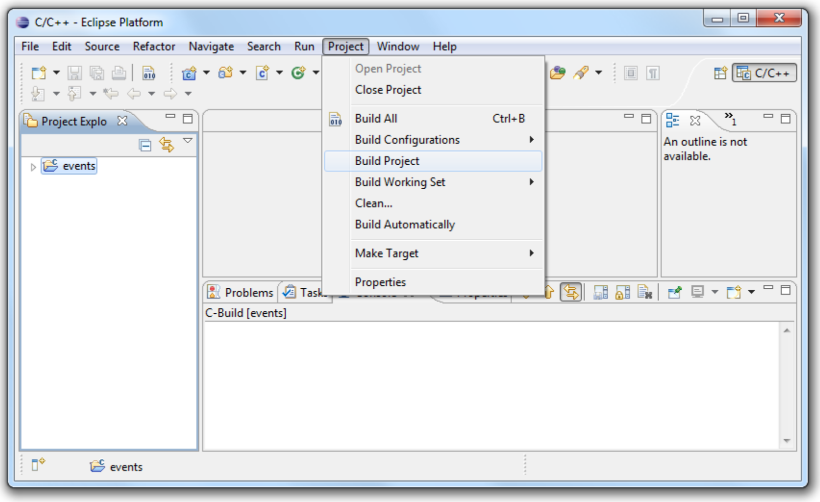
Configuration: Default [Active]

Environment variables to set

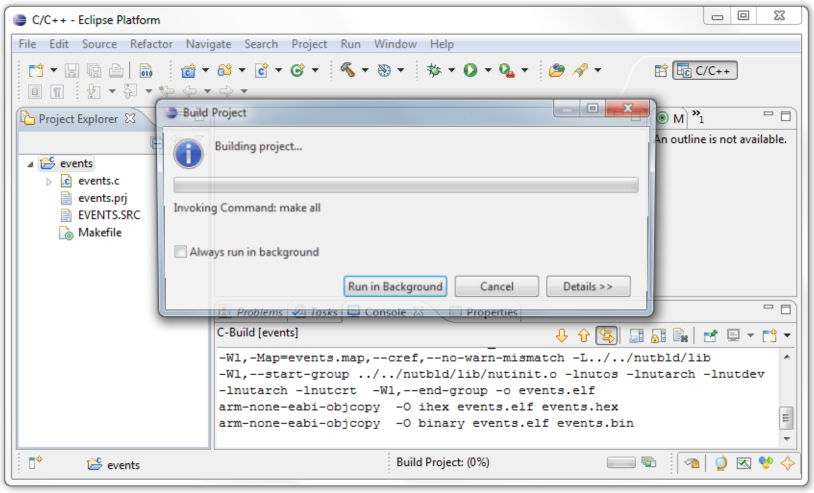
Variable	Value	Origin
CWD	C:\ethernut-4.9\nutapp...	BUILD SYSTEM
PATH	C:/Program Files/Java/jr...	USER: CONFIG
PWD	C:\ethernut-4.9\nutapp...	BUILD SYSTEM

[✓ Append variables to native environment]
Replace native environment with specified one

Then, in the same menu, select *Build Project*.

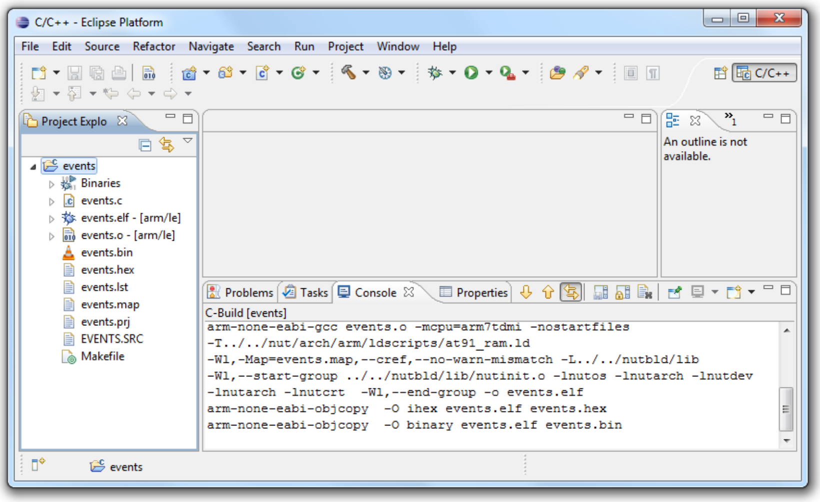


Building this simple application requires a few seconds only.



```
-Wl,-Map=events.map,--cref,--no-warn-mismatch -L../../nutbld/lib
-Wl,--start-group ../../nutbld/lib/nutinit.o -lnutos -lnutarch -lnutdev
-lnutarch -lnutcrt  -Wl,--end-group -o events.elf
arm-none-eabi-objcopy  -O ihex events.elf events.hex
arm-none-eabi-objcopy  -O binary events.elf events.bin
```

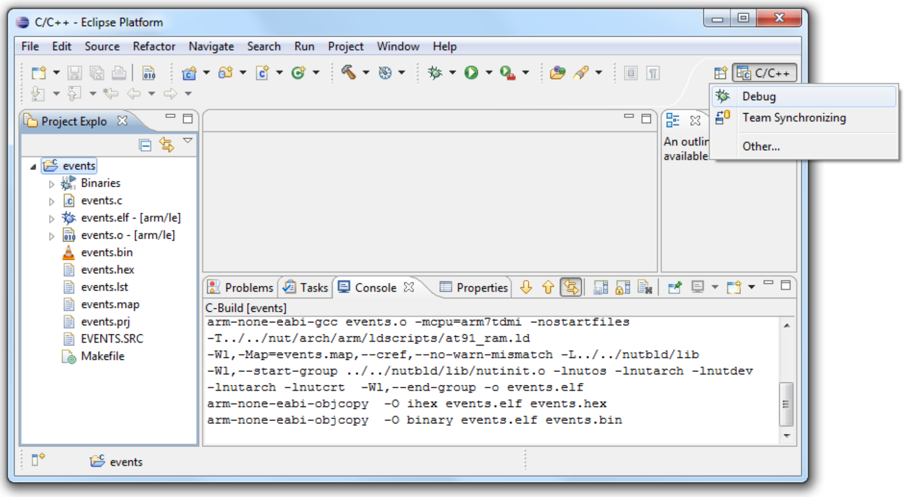
When done, the newly created files appear in the *Project Explorer*. The file *events.elf* contains the executable binary including all debug information.



```
arm-none-eabi-gcc events.o -mcpu=arm7tdmi -nostartfiles
-T../../nut/arch/arm/ldscripts/at91_ram.ld
-Wl,-Map=events.map,--cref,--no-warn-mismatch -L../../nutbld/lib
-Wl,--start-group ../../nutbld/lib/nutinit.o -lnutos -lnutarch -lnutdev
-lnutarch -lnutcrt  -Wl,--end-group -o events.elf
arm-none-eabi-objcopy  -O ihex events.elf events.hex
arm-none-eabi-objcopy  -O binary events.elf events.bin
```

Debugging the Application

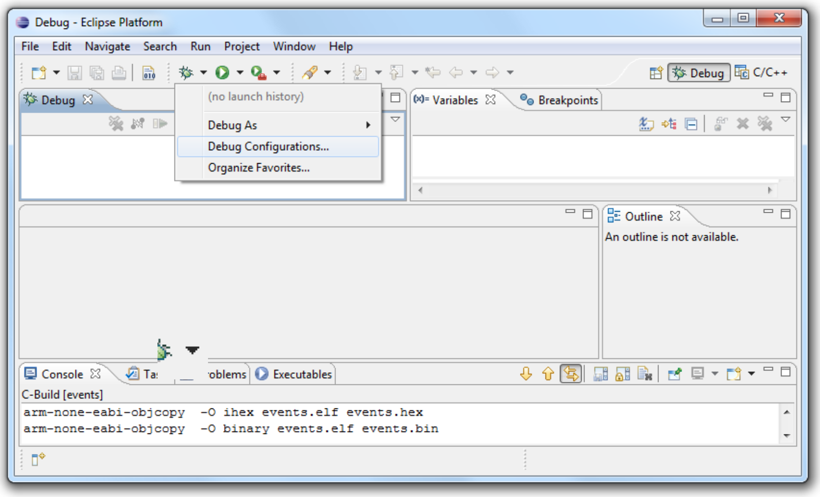
Click on the Open Perspective button and selectDebug. This will open the Debug Perspective.



C-Build [events]

```
arm-none-eabi-gcc events.o -mcpu=arm7tdmi -nostartfiles
-T../../nut/arch/arm/ldscripts/at91_ram.ld
-Wl,-Map=events.map,--cref,--no-warn-mismatch -L../../nutbld/lib
-Wl,--start-group ../../nutbld/lib/nutinit.o -lnutos -lnutarch -lnutdev
-lnutarch -lnutcrt  -Wl,--end-group -o events.elf
arm-none-eabi-objcopy  -O ihex events.elf events.hex
arm-none-eabi-objcopy  -O binary events.elf events.bin
```

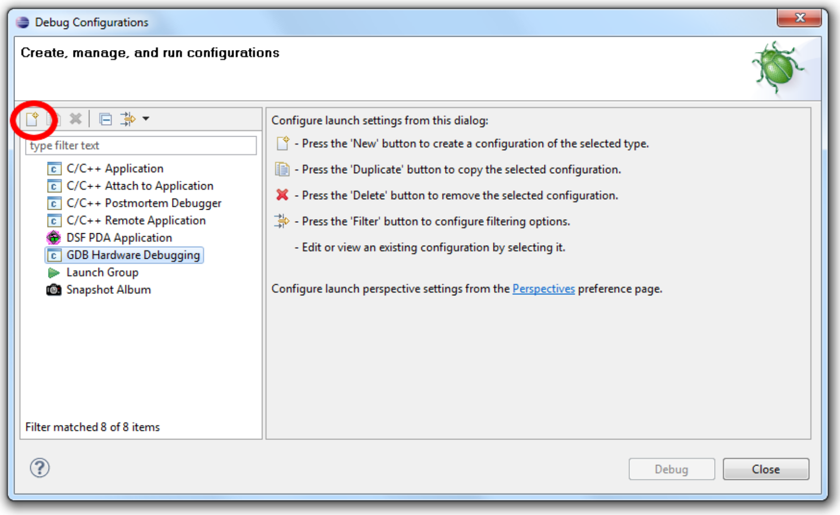
Click on the tiny arrow near the bug in the toolbar to open the related menu. Then select Debug Configuration....



C-Build [events]

```
arm-none-eabi-objcopy  -O ihex events.elf events.hex
arm-none-eabi-objcopy  -O binary events.elf events.bin
```

In the *Debug Configurations* dialog select *GDB Hardware Debugging* and click on the *New* button.

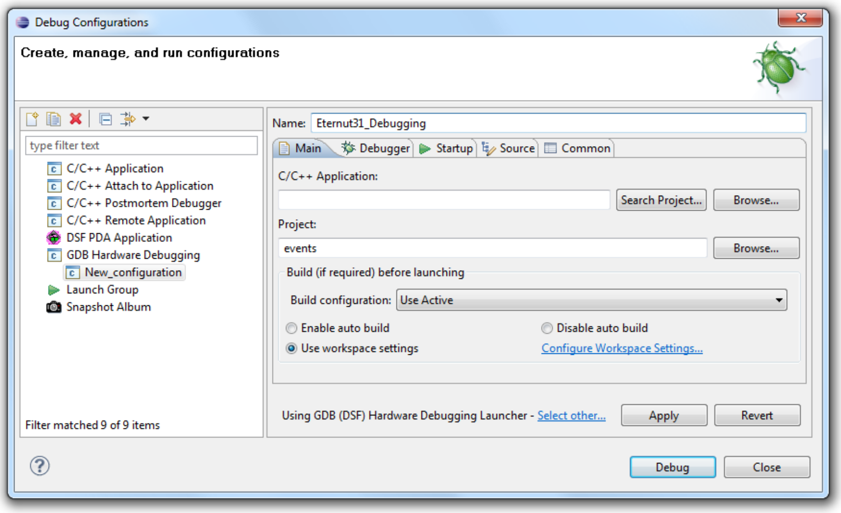


Create, manage, and run configurations

Configure launch settings from this dialog:
- Press the 'New' button to create a configuration of the selected type.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.

Configure launch perspective settings from the Perspectives preference page.

Enter events.elf as the configuration name and events as the project name. Next click on Search Project....


Name: Eternut31_Debugging

C/C++ Application:

Project: events

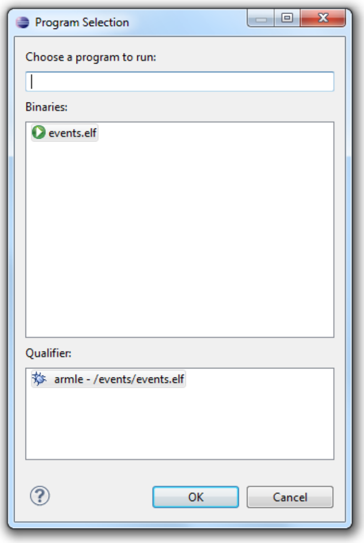
Build configuration: Use Active

Use workspace settings

Using GDB (DSF) Hardware Debugging Launcher - Select other...

Filter matched 9 of 9 items

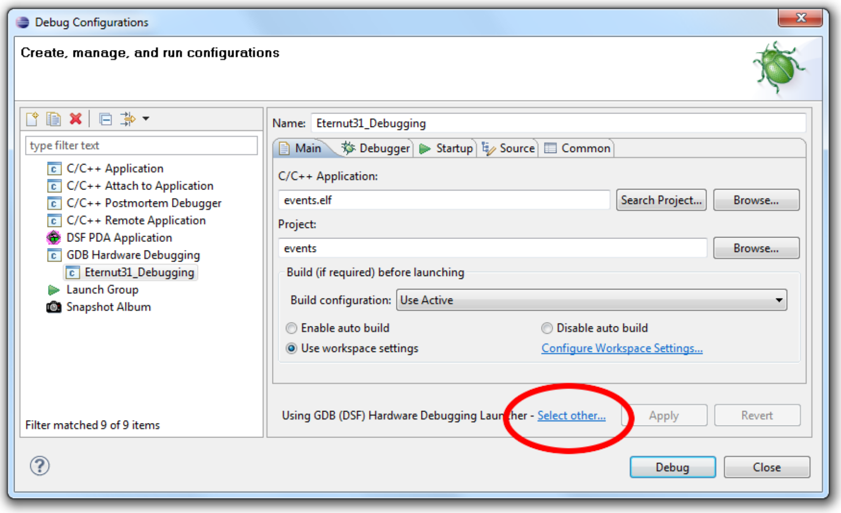
Select events.elf and click OK.


Choose a program to run:

Binaries: events.elf

Qualifier: armle - /events/events.elf

Back in the *Debug Configurations* dialog click on *Select other*.


Name: Eternut31_Debugging

C/C++ Application: events.elf

Project: events

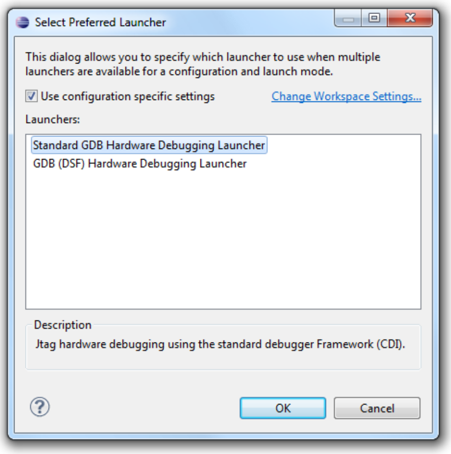
Build configuration: Use Active

Use workspace settings

Using GDB (DSF) Hardware Debugging Launcher - Select other...

Filter matched 9 of 9 items

Choose Standard GDB Hardware Debugging Launcher as the preferred launcher and click OK.



Select Preferred Launcher

This dialog allows you to specify which launcher to use when multiple launchers are available for a configuration and launch mode.

☑ Use configuration specific settings Change Workspace Settings...

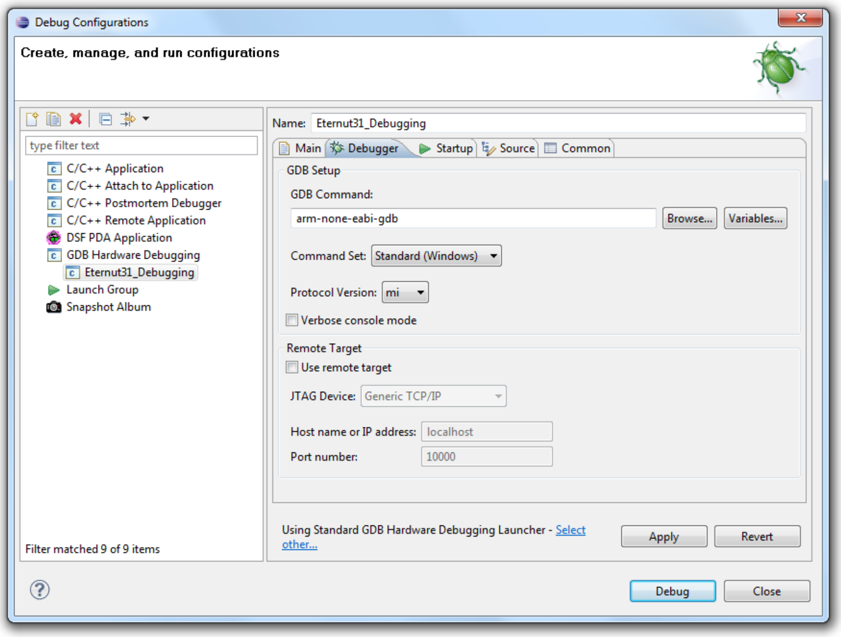
Launchers:

Standard GDB Hardware Debugging Launcher
GDB (DSF) Hardware Debugging Launcher

Description

Jtag hardware debugging using the standard debugger Framework (CDI).

Back in the *Debug Configurations* dialog click on the *Debugger* tab and enter `arm-none-eabi-gdb` in *GDB Command*. Although we will use remote target debugging, *Use remote target* must be unchecked here.



Debug Configurations

Create, manage, and run configurations

Name: Eternut31_Debugging

GDB Setup

GDB Command: arm-none-eabi-gdb

Command Set: Standard (Windows)

Protocol Version: mi

☐ Verbose console mode

Remote Target

☐ Use remote target

JTAG Device: Generic TCP/IP

Host name or IP address: localhost

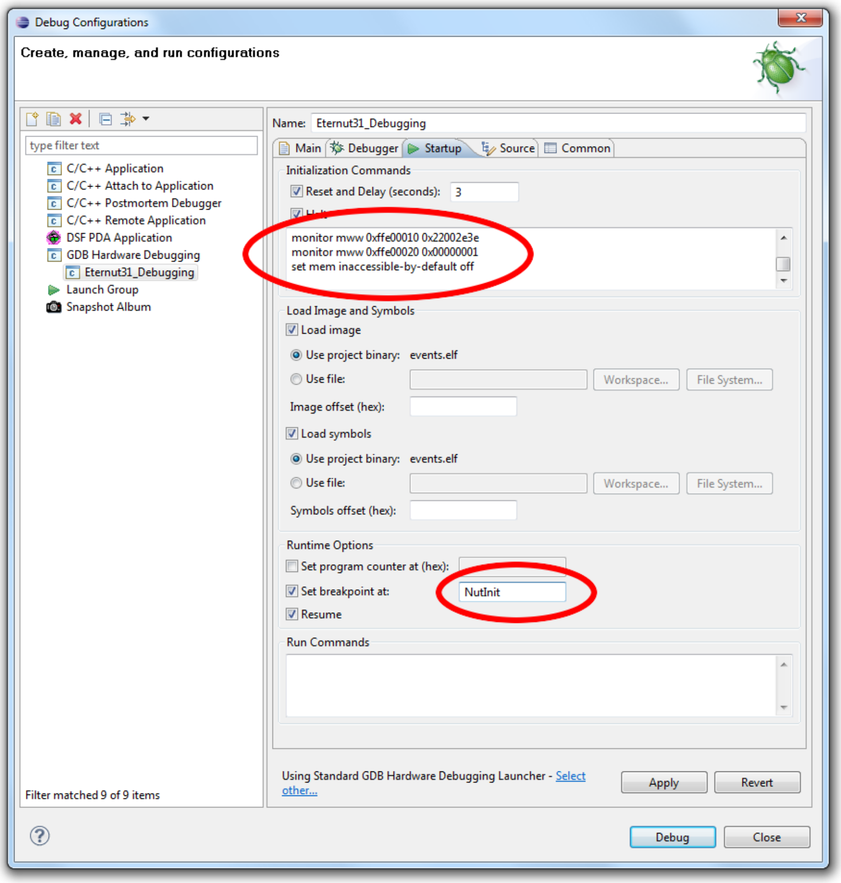
Port number: 10000

Using Standard GDB Hardware Debugging Launcher - Select other...

Now click on the *Startup* tab. *Set breakpoint at* and *Resume* must be checked. Enter `NutInit` as the breakpoint target.

Then copy and paste the following commands into the field *Initialization Commands*.

```
target remote localhost:3333
monitor reset
monitor sleep 500
monitor poll
monitor soft_reset_halt
monitor gdb_breakpoint_override soft
monitor mww 0xffe00000 0x1000212d
monitor mww 0xffe00004 0x20003025

monitor mww 0xffe00008 0x21002026
monitor mww 0xffe00010 0x22002e3e
monitor mww 0xffe00020 0x00000001
set mem inaccessible-by-default off
```



Debug Configurations

Create, manage, and run configurations

type filter text

- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- DSF PDA Application
- GDB Hardware Debugging
  - Eternut31_Debugging
- Launch Group
- Snapshot Album

Filter matched 9 of 9 items

Name: Eternut31_Debugging

Main | Debugger | Startup | Source | Common

Initialization Commands

[✓] Reset and Delay (seconds): 3

[✓] Halt

```
monitor mww 0xffe00010 0x22002e3e
monitor mww 0xffe00020 0x00000001
set mem inaccessible-by-default off
```

Load Image and Symbols

[✓] Load image

(●) Use project binary: events.elf

( ) Use file: Workspace... File System...

Image offset (hex):

[✓] Load symbols

(●) Use project binary: events.elf

( ) Use file: Workspace... File System...

Symbols offset (hex):

Runtime Options

[ ] Set program counter at (hex):

[✓] Set breakpoint at: NutInit

[✓] Resume

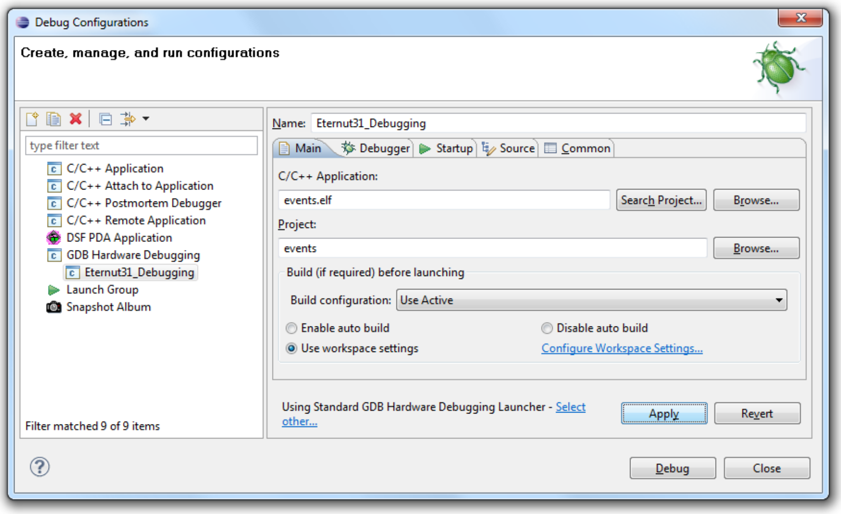
Run Commands

Using Standard GDB Hardware Debugging Launcher - Select other...

Apply Revert

Debug Close

Finally click Apply to store the settings and Close to close the dialog.



Debug Configurations

Create, manage, and run configurations

type filter text

- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- DSF PDA Application
- GDB Hardware Debugging
  - Eternut31_Debugging
- Launch Group
- Snapshot Album

Filter matched 9 of 9 items

Name: Eternut31_Debugging

Main | Debugger | Startup | Source | Common

C/C++ Application:

events.elf Search Project... Browse...

Project:

events Browse...

Build (if required) before launching

Build configuration: Use Active

( ) Enable auto build ( ) Disable auto build

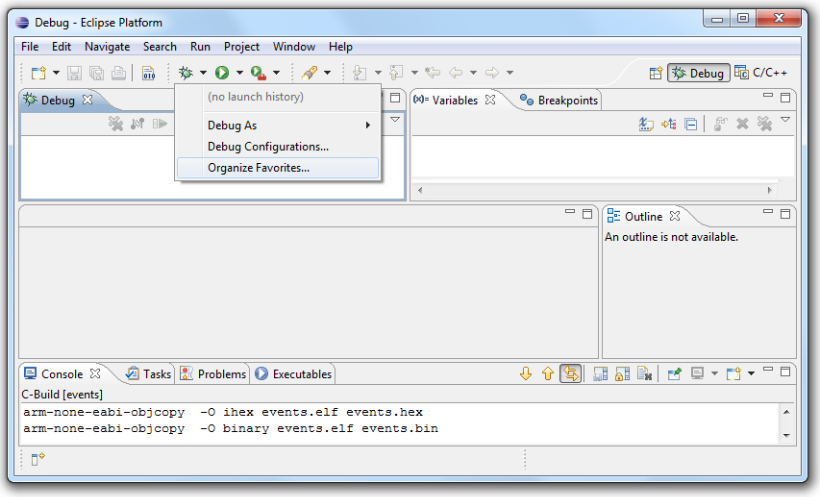
(●) Use workspace settings Configure Workspace Settings...

Using Standard GDB Hardware Debugging Launcher - Select other...

Apply Revert

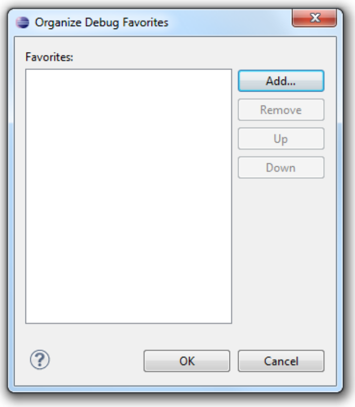
Debug Close

Once again click on the tiny arrow near the bug to open the menu and select *Organize Favorites...*.

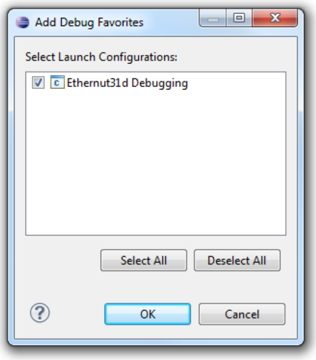


```
arm-none-eabi-objcopy  -O ihex events.elf events.hex
arm-none-eabi-objcopy  -O binary events.elf events.bin
```

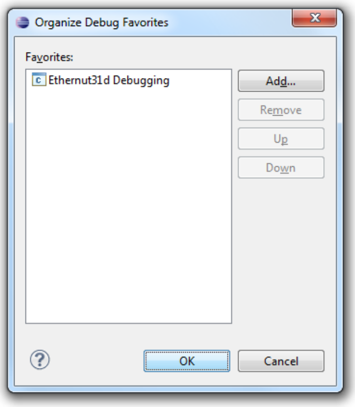
Click Add



Check events.elf and click OK.

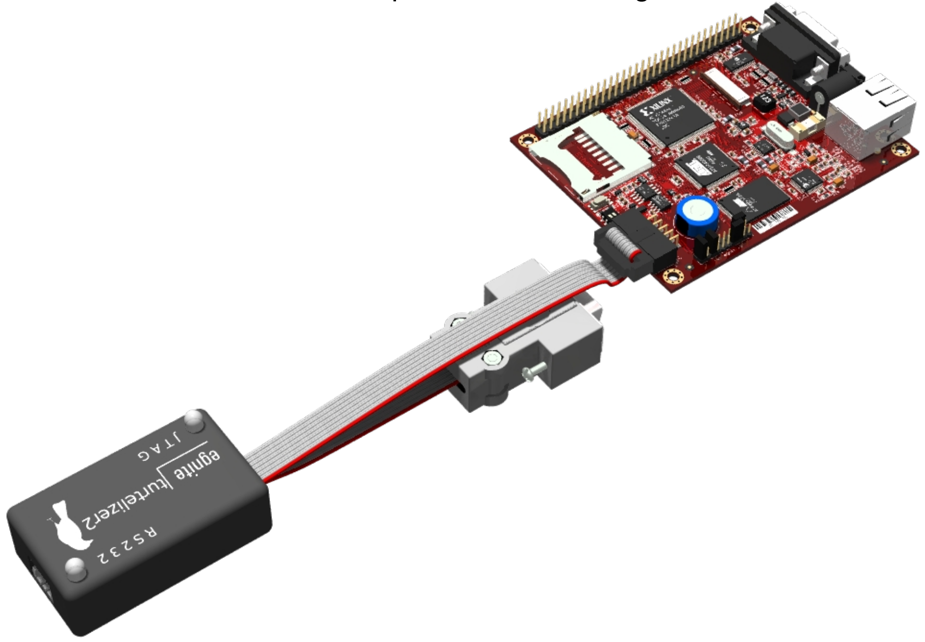


In the previous dialog click OK again.



We will now setup the hardware, using the following steps:

- Connect the Turtelizer's JTAG port to the Ethernut board's 10-pin connector.
- Attach the power supply to the Ethernut Board and switch it on. The red LED should go on.
- Connect the Turtelizer's USB port to the PC using the USB A/B cable.



If the Turtelizer is used for the first time, it will be required to install the FTDI driver. For more details please refer to the Turtelizer's documentation.

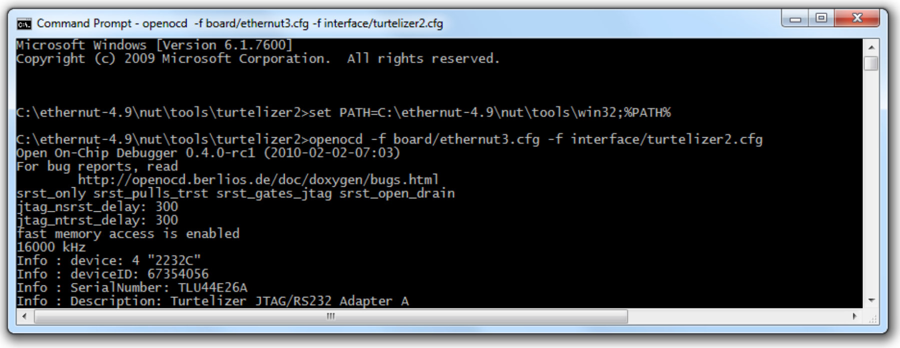
Next we will start OpenOCD. We could do this within Eclipse, but for the first time it makes sense to start it manually to check the output in case of problems.

Unfortunately, the standard configuration for OpenOCD will not work out of the box. You can use a simple text editor like *Notepad* to fix this. Open the file *C:\Ethernut-4.9\nut\tools\turtelizer2\board\ethernut3.cfg* and put a number sign in front of the following line:

```
#arm7_9 dcc_downloads enable
```

Save your change

Now open a command line and change to directory C:\Ethernut\4.9\nut\tools\turtelizer2. Add the Nut/OS tools directory for Windows to your path, using the following command


Command Prompt - openocd -f board/ethernut3.cfg -f interface/turtelizer2.cfg

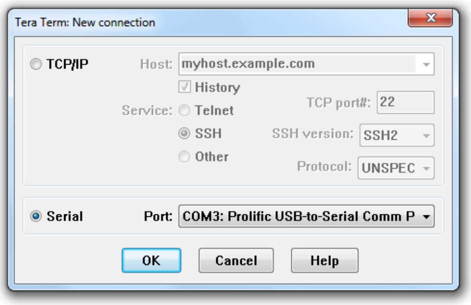
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.


C:\ethernut-4.9\nut\tools\turtelizer2>set PATH=C:\ethernut-4.9\nut\tools\win32;%PATH%

C:\ethernut-4.9\nut\tools\turtelizer2>openocd -f board/ethernut3.cfg -f interface/turtelizer2.cfg
Open On-Chip Debugger 0.4.0-rc1 (2010-02-02-07:03)
For bug reports, read
        http://openocd.berlios.de/doc/doxygen/bugs.html
srst_only srst_pulls_trst srst_gates_jtag srst_open_drain
jtag_nsrst_delay: 300
jtag_ntrst_delay: 300
fast memory access is enabled
16000 kHz
Info : device: 4 "2232C"
Info : deviceID: 67354056
Info : SerialNumber: TLU44E26A
Info : Description: Turtelizer JTAG/RS232 Adapter A
```

```
set PATH=C:\ethernut-4.9\nut\tools\win32;%PATH%
```

Then start OpenOCD using

```
openocd -f board/ethernut3.cfg -f interface/turtelizer2.cfg
```

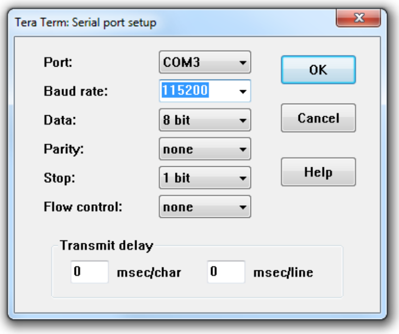
The *events* sample will create some output on the RS-232. To make it visible, we start a *terminal emulator*. On Windows PCs we could use *HyperTerminal*, but *TeraTerm* is the preferred one. After *TeraTerm* started, select *Serial port...* from the *Setup* menu.


Tera Term: New connection

TCP/IP — Host: myhost.example.com; History; Service: Telnet, SSH (selected), Other; TCP port#: 22; SSH version: SSH2; Protocol: UNSPEC

Serial (selected) — Port: COM3: Prolific USB-to-Serial Comm P

Select the COM port to which the Ethernut board is connected. Configure the port to 8 data bits, no parity, 1 stop bit and no flow control and click on OK.


Tera Term: Serial port setup

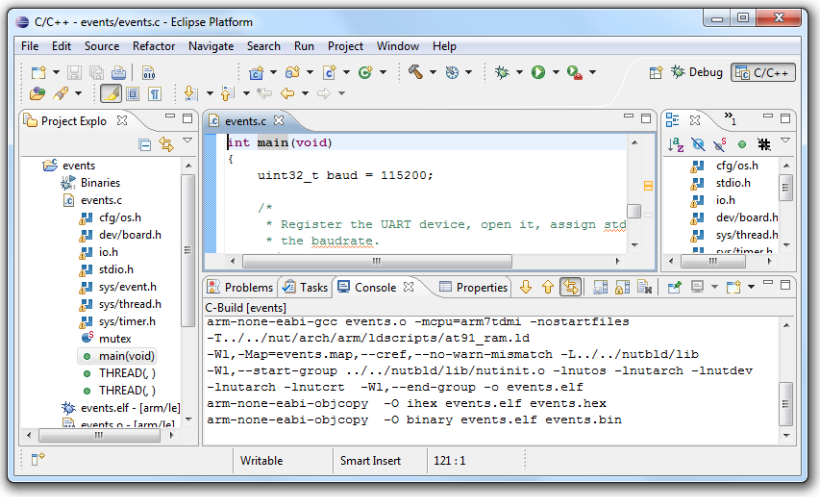
Port:	COM3
Baud rate:	115200
Data:	8 bit
Parity:	none
Stop:	1 bit
Flow control:	none

Transmit delay: 0 msec/char, 0 msec/line

Let's get back to *Eclipse*. Remember that we specified a breakpoint at `NutInit` in the debugger configuration. For some unknown reason the debugger fails, if this is not defined. However, the function NutInit is located in the initialization of Nut/OS, while you will be mostly interested in debugging your application code.

Lets specify a second breakpoint at the beginning of the application code, which is the function main. Click on the tab *C/C++* in the upper right corner to change to the C Perspective.

In the *Project Explorer* double click on the events folder to expand it. Then click on the tiny arrow in front of *events.c* to expand this part of the tree. Note, that this arrow will appear only when moving the mouse over the *Project Explorer* window.

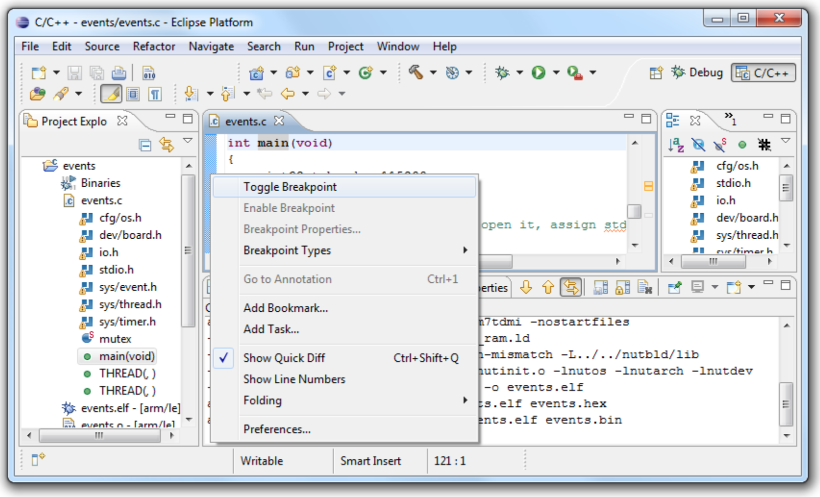

C/C++ - events/events.c - Eclipse Platform

```
int main(void)
{
    uint32_t baud = 115200;

    /*
     * Register the UART device, open it, assign std
     * the baudrate.
```

```
arm-none-eabi-gcc events.o -mcpu=arm7tdmi -nostartfiles
-T../../nut/arch/arm/ldscripts/at91_ram.ld
-Wl,-Map=events.map,--cref,--no-warn-mismatch -L../../nutbld/lib
-Wl,--start-group ../../nutbld/lib/nutinit.o -lnutos -lnutarch -lnutdev
-lnutarch -lnutcrt  -Wl,--end-group -o events.elf
arm-none-eabi-objcopy  -O ihex events.elf events.hex
arm-none-eabi-objcopy  -O binary events.elf events.bin
```

Now double click on *main(void)* to open the source file at this location.

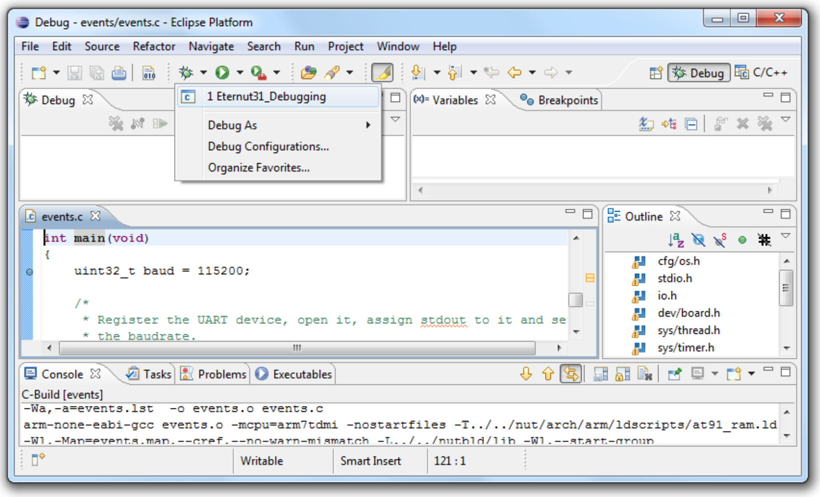
At the first line click with the right mouse button on blue border to open the context menu. Select *Toggle Breakpoint*.



```
int main(void)
{
                                 open it, assign std
```

```
m7tdmi -nostartfiles
_ram.ld
n-mismatch -L../../nutbld/lib
nutinit.o -lnutos -lnutarch -lnutdev
 -o events.elf
ts.elf events.hex
ents.elf events.bin
```

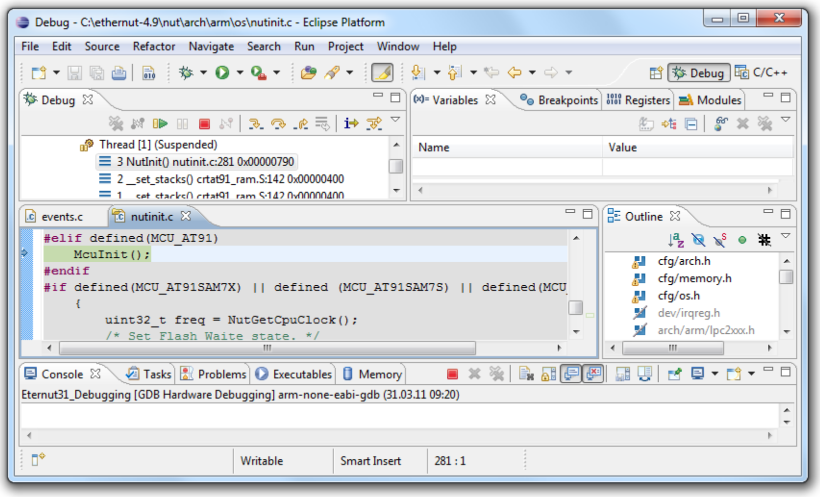
Switch back to the debug perspective by clicking on the tab *Debug* in the upper right corner. Click on the bug button in the toolbar to start debugging.



```
int main(void)
{
    uint32_t baud = 115200;

    /*
     * Register the UART device, open it, assign stdout to it and se
     * the baudrate.
```

```
C-Build [events]
-Wa,-a=events.lst  -o events.o events.c
arm-none-eabi-gcc events.o -mcpu=arm7tdmi -nostartfiles -T../../nut/arch/arm/ldscripts/at91_ram.ld
-Wl,-Map=events.map,--cref,--no-warn-mismatch -L../../nutbld/lib -Wl,--start-group
```

The code will be downloaded to the target and started. Then it stops at the first breakpoint at the beginning of NutInit. Press F8 or click the *Resume* button

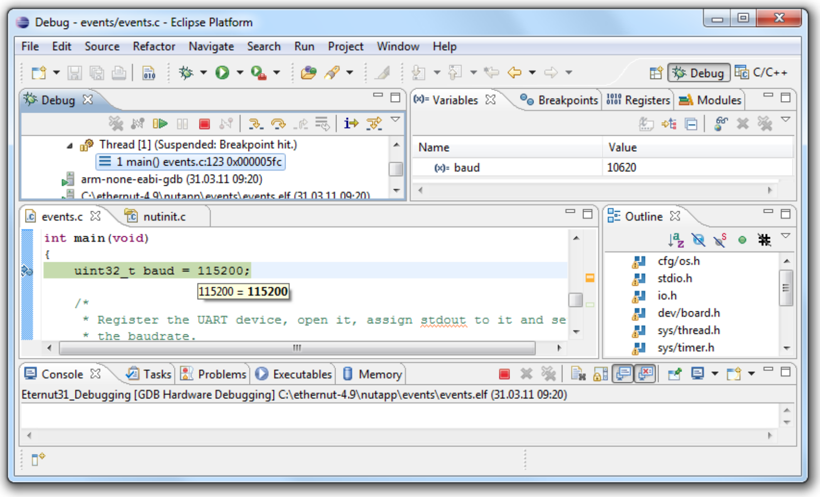


```
Thread [1] (Suspended)
    3 NutInit() nutinit.c:281 0x00000790
    2 __set_stacks() crtat91_ram.S:142 0x00000400
    1 __set_stacks() crtat91_ram.S:142 0x00000400
```

```
#elif defined(MCU_AT91)
    McuInit();
#endif
#if defined(MCU_AT91SAM7X) || defined (MCU_AT91SAM7S) || defined(MCU_
    {
        uint32_t freq = NutGetCpuClock();
        /* Set Flash Waite state. */
```

```
Eternut31_Debugging [GDB Hardware Debugging] arm-none-eabi-gdb (31.03.11 09:20)
```

The target code will run up to the next breakpoint in main. You can now press F5 or click the utton to step execute a single line, F6 or click the button tep over a function call.



Debug - events/events.c - Eclipse Platform

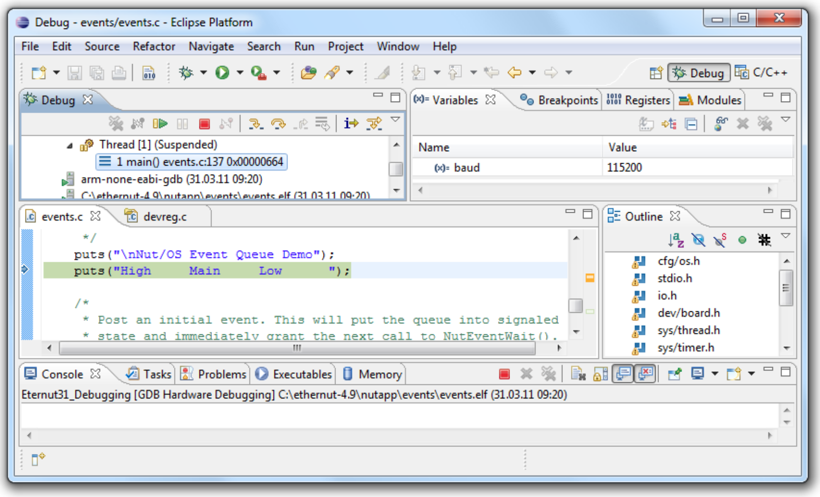
```
Thread [1] (Suspended: Breakpoint hit.)
    1 main() events.c:123 0x000005fc
arm-none-eabi-gdb (31.03.11 09:20)
C:\ethernut-4.9\nutapp\events\events.elf (31.03.11 09:20)
```

Name	Value
baud	10620

```
int main(void)
{
    uint32_t baud = 115200;
                       115200 = 115200
    /*
     * Register the UART device, open it, assign stdout to it and se
     * the baudrate.
```

```
Eternut31_Debugging [GDB Hardware Debugging] C:\ethernut-4.9\nutapp\events\events.elf (31.03.11 09:20)
```

Now continue pressing F6 until we stepped over the first puts() function call.



Debug - events/events.c - Eclipse Platform

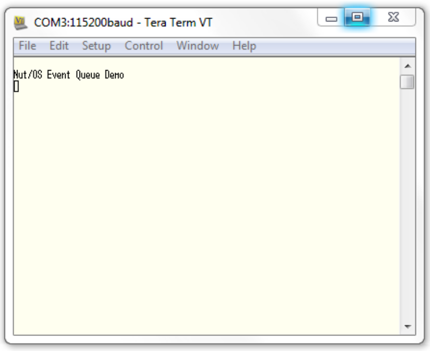
```
Thread [1] (Suspended)
    1 main() events.c:137 0x00000664
arm-none-eabi-gdb (31.03.11 09:20)
C:\ethernut-4.9\nutapp\events\events.elf (31.03.11 09:20)
```

Name	Value
baud	115200

```
     */
    puts("\nNut/OS Event Queue Demo");
    puts("High     Main     Low      ");

    /*
     * Post an initial event. This will put the queue into signaled
     * state and immediately grant the next call to NutEventWait().
```

```
Eternut31_Debugging [GDB Hardware Debugging] C:\ethernut-4.9\nutapp\events\events.elf (31.03.11 09:20)
```

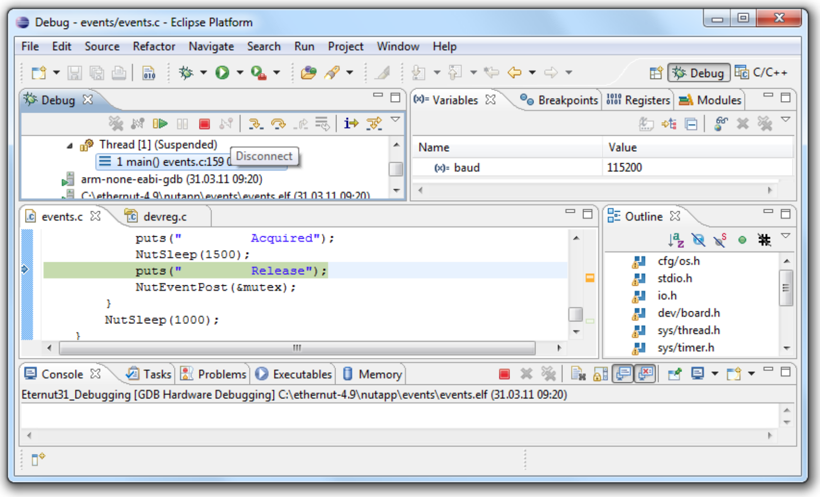
When this statement has been executed, the text `Nut/OS Event Queue Demo` should appear in the terminal emulator window.



COM3:115200baud - Tera Term VT

```
Nut/OS Event Queue Demo
```

As you continue stepping through the main loop...


Debug - events/events.c - Eclipse Platform
File Edit Source Refactor Navigate Search Run Project Window Help
Debug C/C++
Debug
Variables Breakpoints Registers Modules
Disconnect
events.c devreg.c
Outline
Console Tasks Problems Executables Memory

Thread [1] (Suspended)
1 main() events.c:159
arm-none-eabi-gdb (31.03.11 09:20)
C:\ethernut-4.9\nutapp\events\events.elf (31.03.11 09:20)

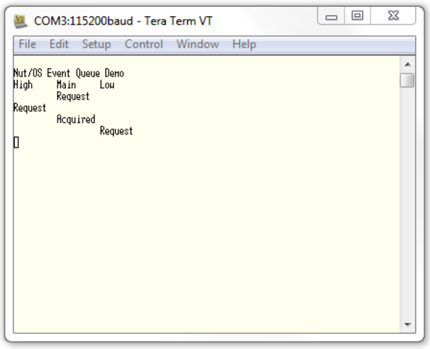
Name	Value
baud	115200

```
            puts("          Acquired");
            NutSleep(1500);
            puts("          Release");
            NutEventPost(&mutex);
        }
        NutSleep(1000);
    }
```

cfg/os.h
stdio.h
io.h
dev/board.h
sys/thread.h
sys/timer.h

Eternut31_Debugging [GDB Hardware Debugging] C:\ethernut-4.9\nutapp\events\events.elf (31.03.11 09:20)

...additional output appears on the serial port.


COM3:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Nut/OS Event Queue Demo
High    Main    Low
        Request
Request
        Acquired
                Request
```

You may set additional breakpoints, evaluate variables or memory areas and do many other useful things during debugging. Please refer to the Eclipse documentation for further details.

Links

- http://www.yagarto.de/howto/yagarto2/ Michael Fischer provided the technical details to make it work.

egnite GmbH
Erinstr. 9
44575 Castrop-Rauxel
Germany

Phone +49 (0)23 05-44 12 56
Fax +49 (0)23 05-44 14 87

Email info@egnite.de
http://www.egnite.de
http://www.ethernut.de