1 Eclipse projects for KNJN boards

By Victor Suarez.

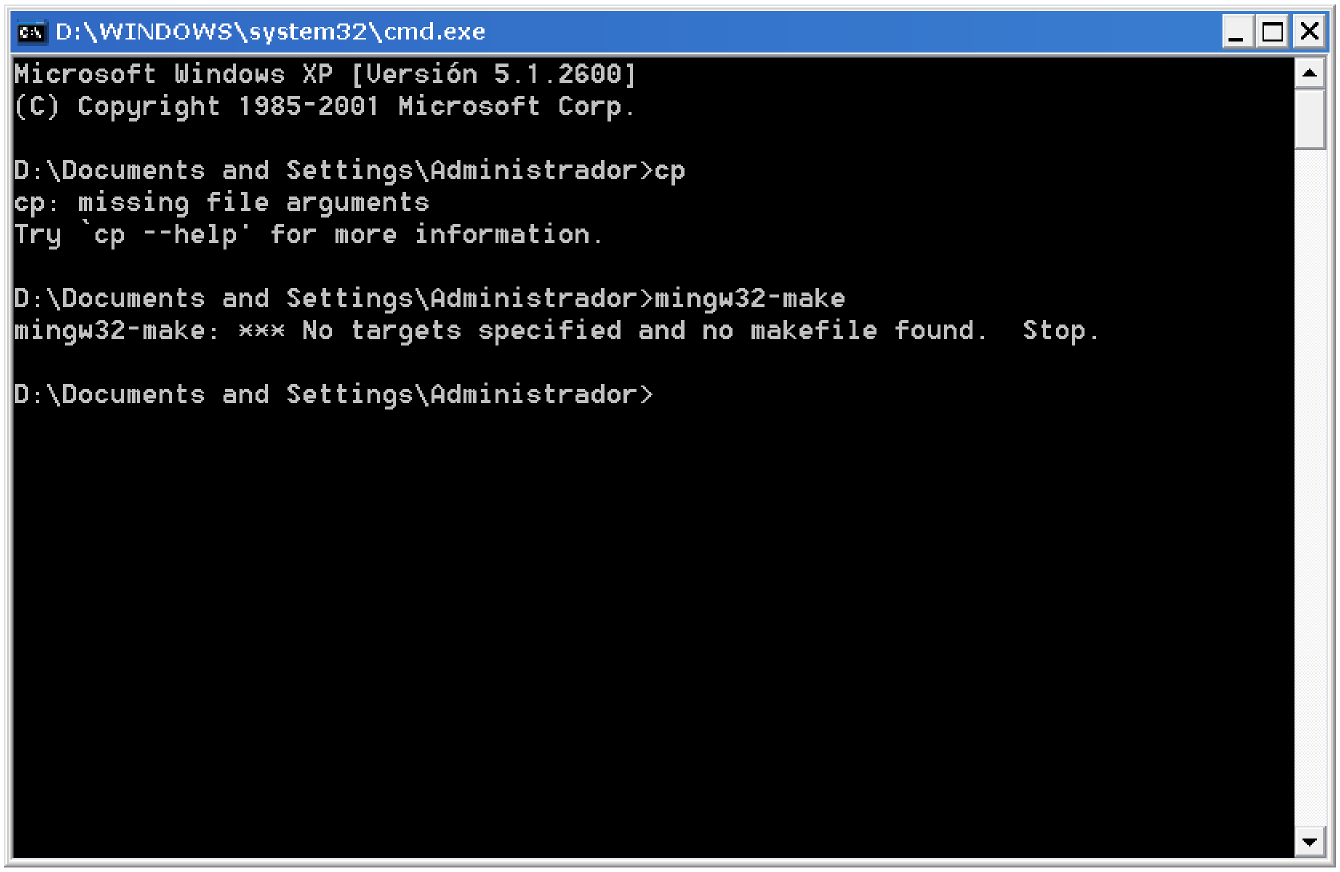
Last modified February 2, 2008

This documentation describes how to use Eclipse with KNJN boards. We use code from one of the KNJN board startup-kit example. We don't use makefiles (as KNJN examples do) because they are automatically generated by CDT, including new files you add to the project. Other features are the automatic generation of listings of generated assembly code intermixed with C/C++ code for the ease of debugging, compiler switches like in your project, automatic compilation of startup or other assembly files, use of linker script, etc.

2 Installation

If you don't have gnu make or msys, download and install them. Make sure they are in your path.
Download make from http://ufpr.dl.sourceforge.net/sourceforge/mingw/mingw32-make-3.80.0-3.exe
Download Msys from http://ufpr.dl.sourceforge.net/sourceforge/mingw/MSYS-1.0.10.exe

Test mingw32-make and msys tools are working well, opening a console and running some commands, like cp and mingw32-make.

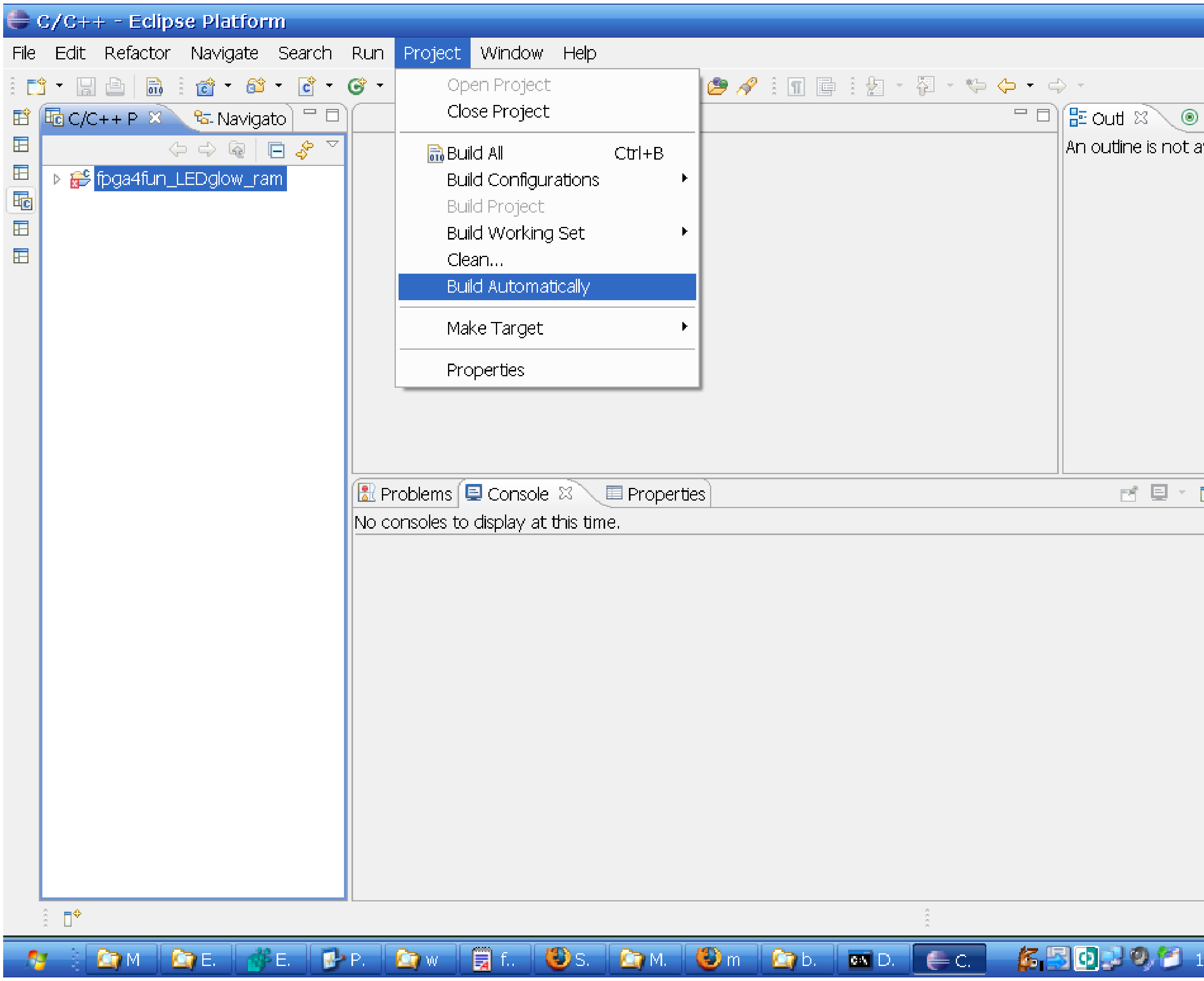


```
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\Administrador>cp
cp: missing file arguments
Try `cp --help' for more information.

D:\Documents and Settings\Administrador>mingw32-make
mingw32-make: *** No targets specified and no makefile found.  Stop.

D:\Documents and Settings\Administrador>
```

Download Yagarto IDE and toolchain.
Install them, default options works ok.

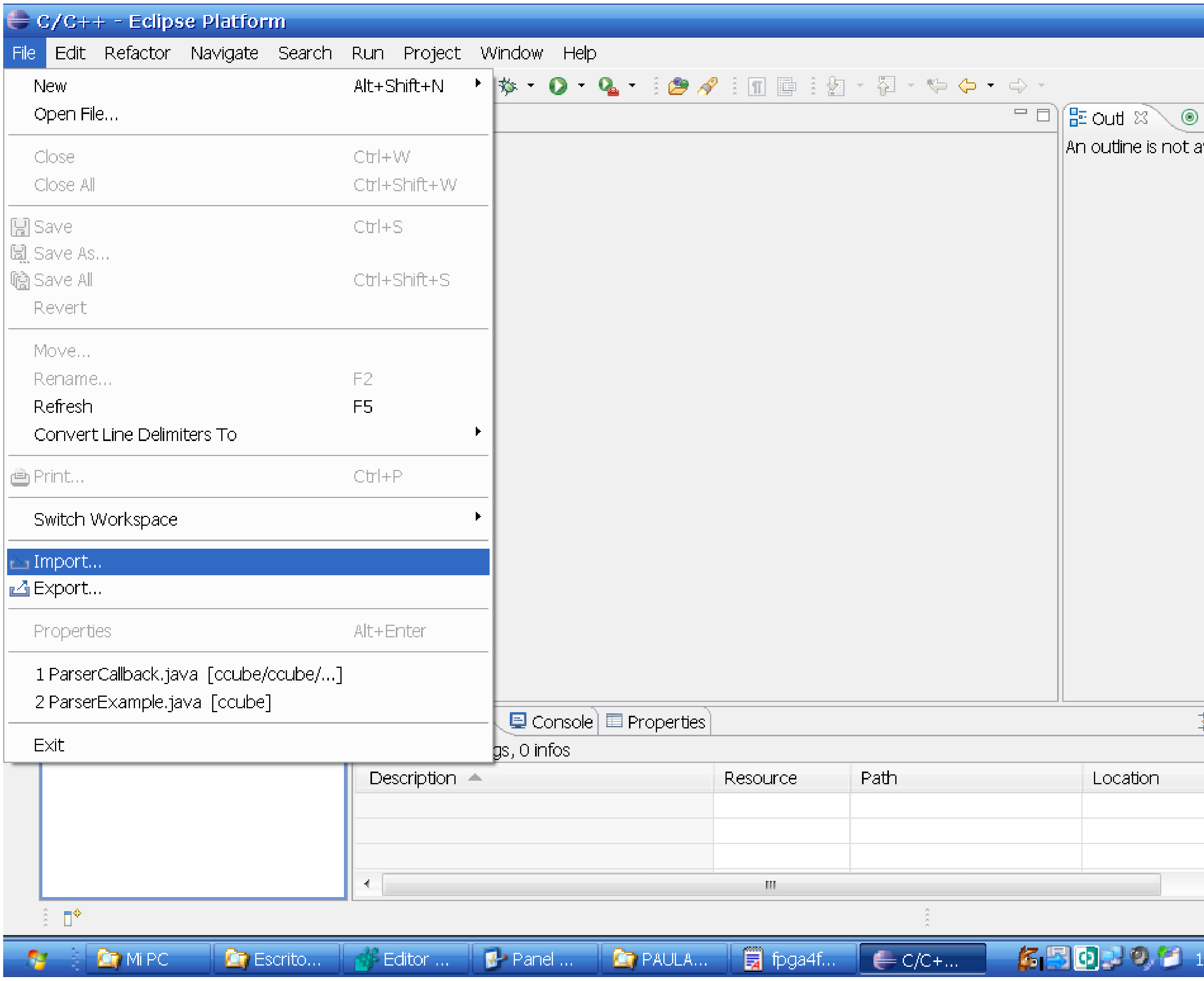
Run eclipse

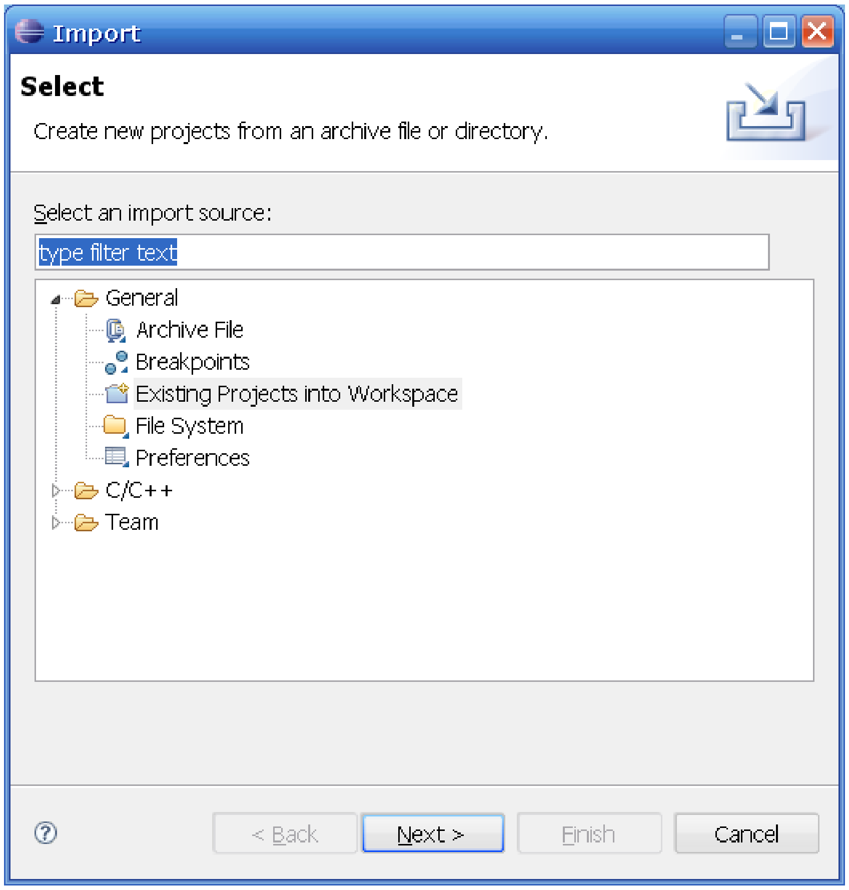
3 Disable automatic build



No consoles to display at this time.

4 Menu File/Import




Import

Select

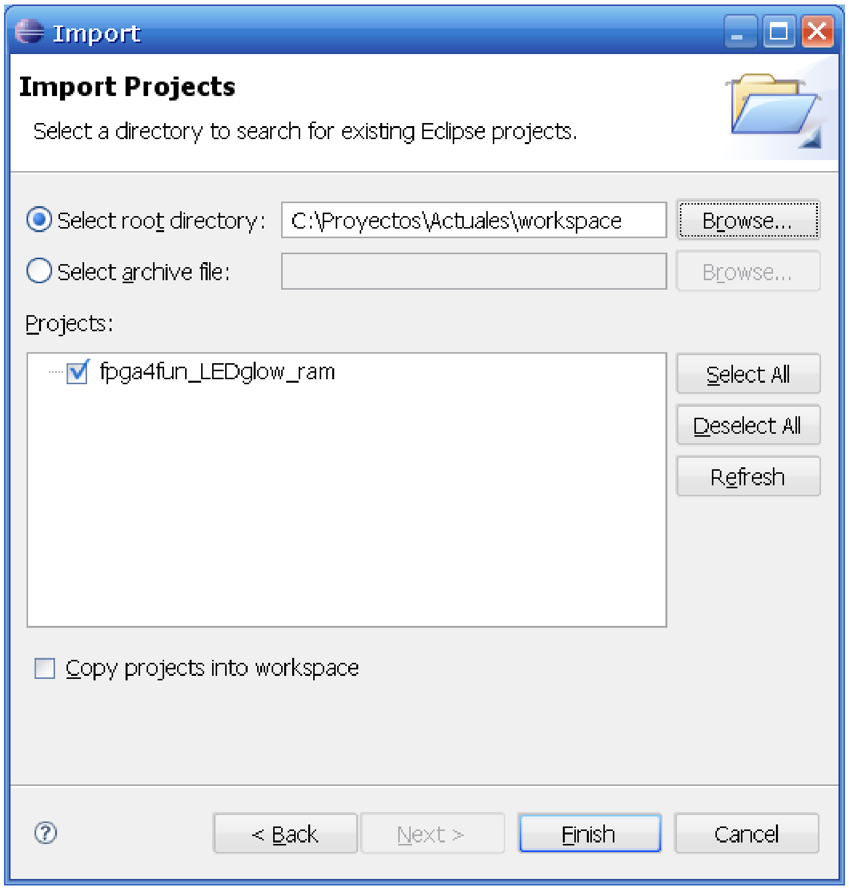
Create new projects from an archive file or directory.

Select an import source:

type filter text

- General
  - Archive File
  - Breakpoints
  - Existing Projects into Workspace
  - File System
  - Preferences
- C/C++
- Team

5 Select root directory and select project



Import

Import Projects

Select a directory to search for existing Eclipse projects.

[✓ Select root directory:] C:\Proyectos\Actuales\workspace

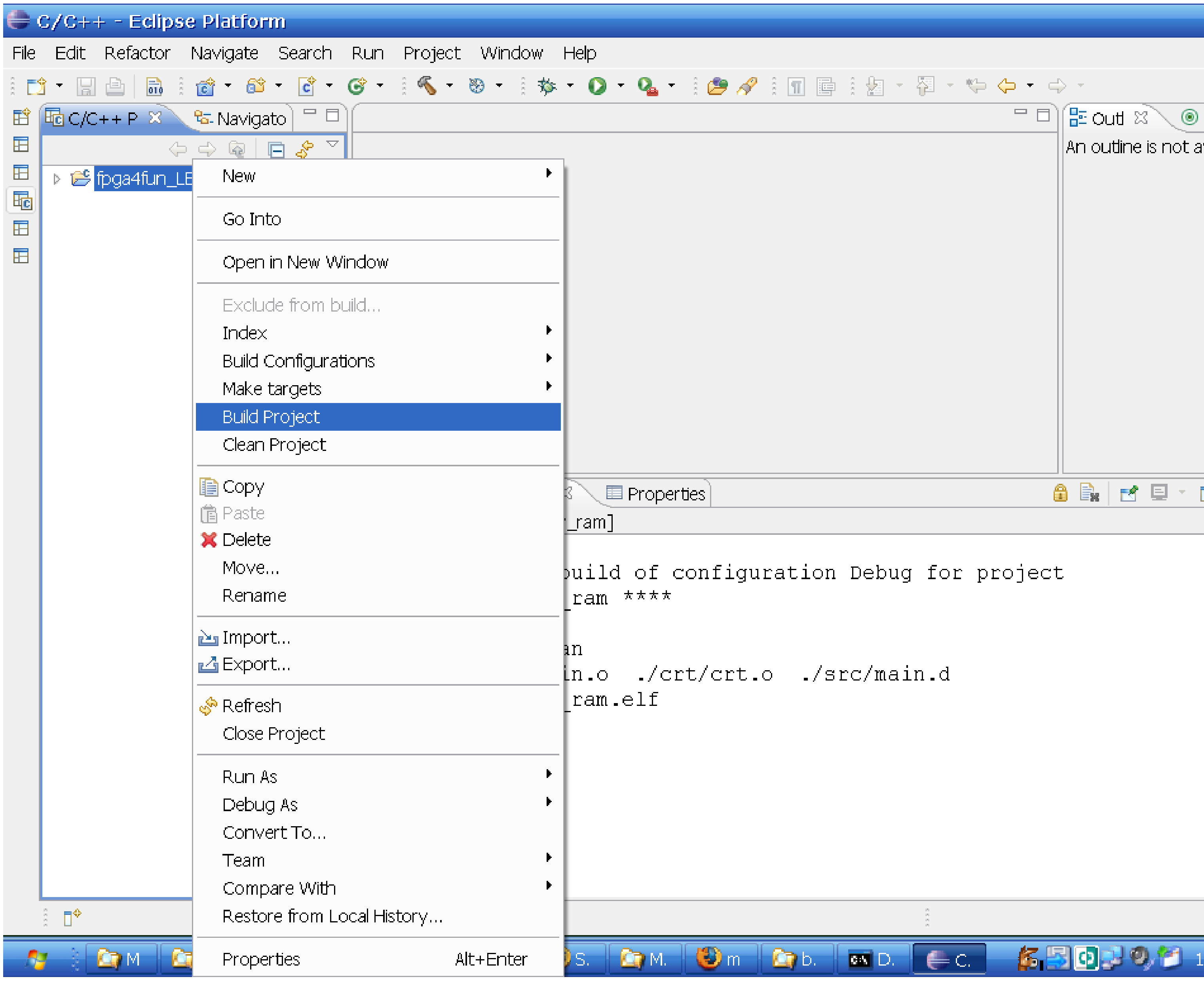
Select archive file:

Projects:

✓ fpga4fun_LEDglow_ram

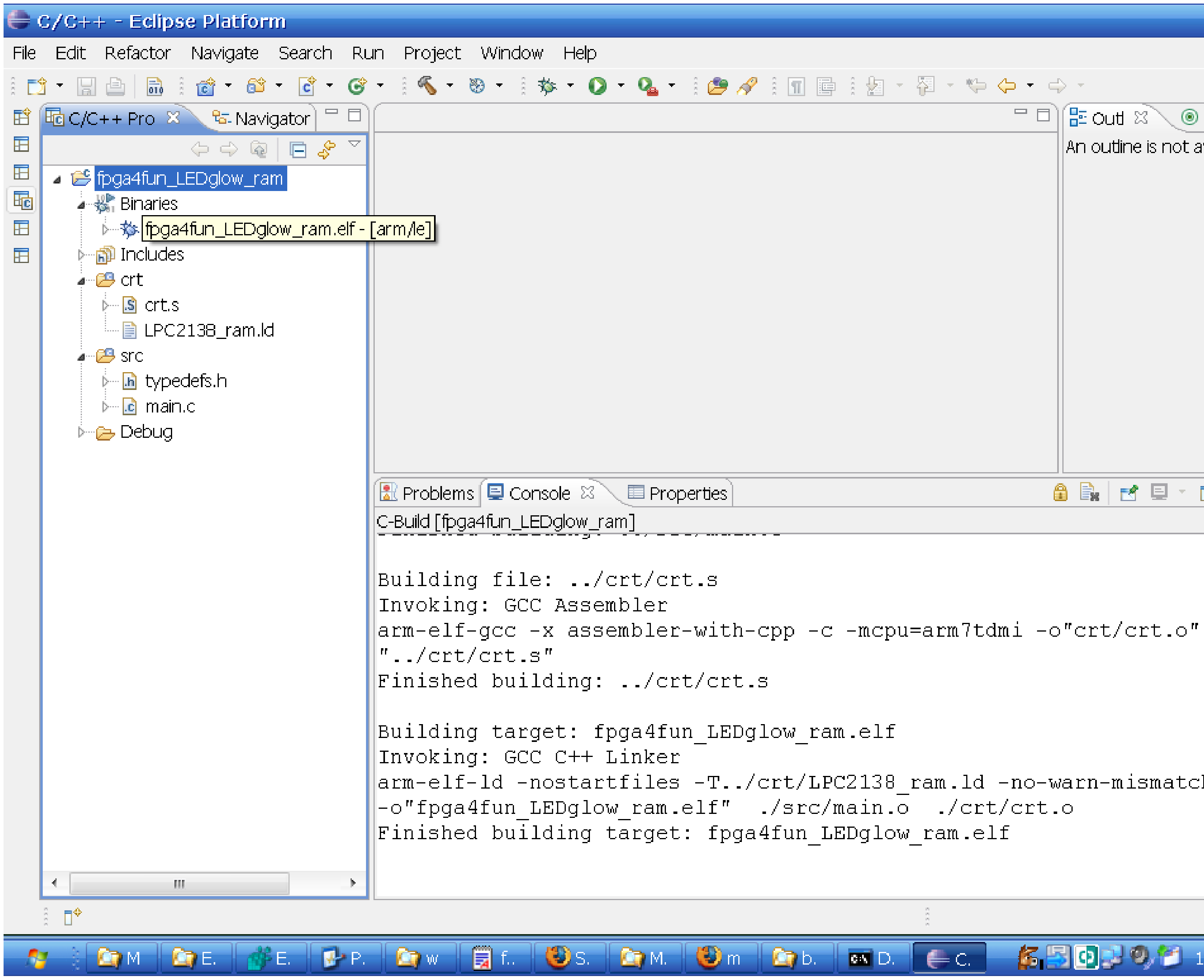
☐ Copy projects into workspace

6 Build project



```
_ram]

build of configuration Debug for project
_ram ****

an
in.o  ./crt/crt.o  ./src/main.d
_ram.elf
```

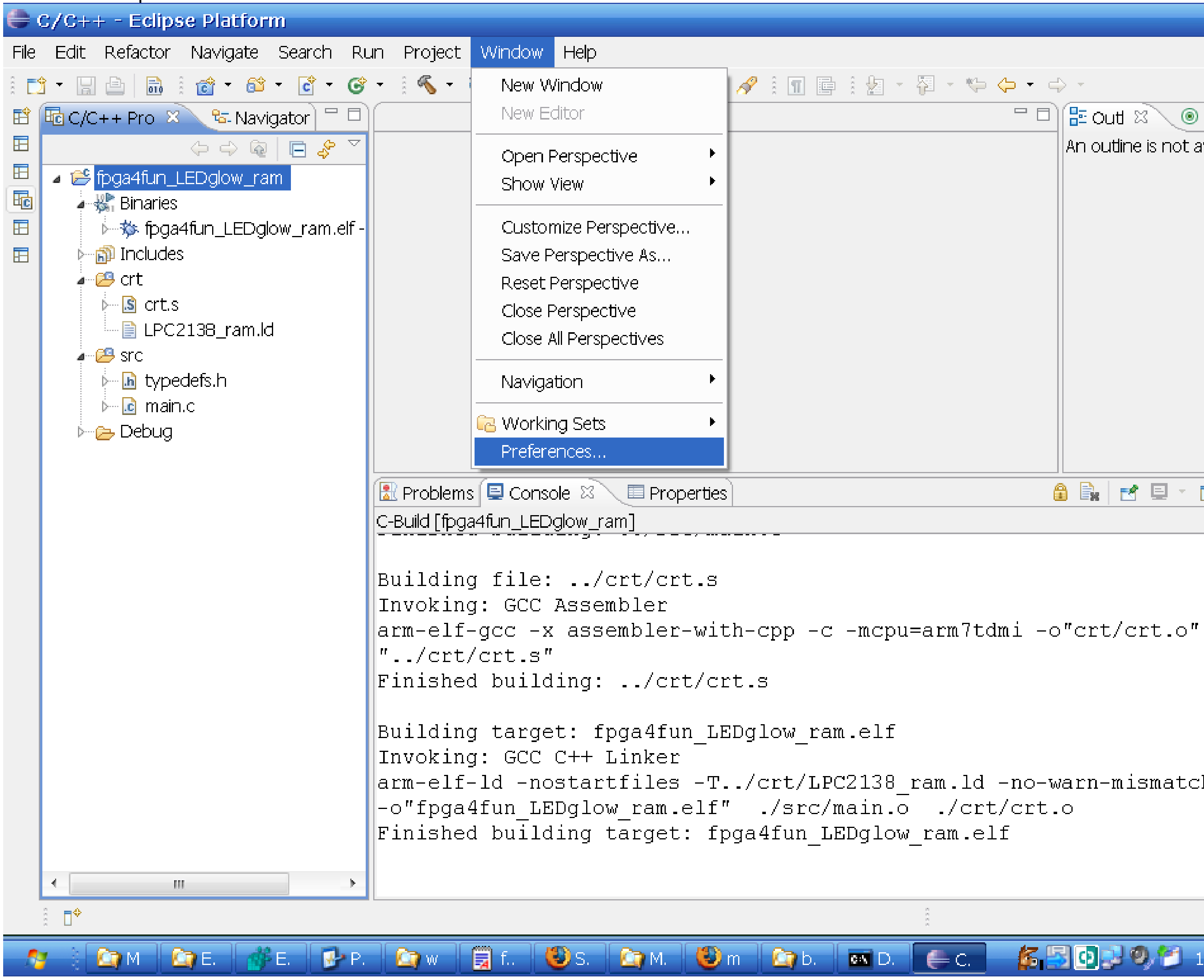
7 Build done (elf generated)



```
Building file: ../crt/crt.s
Invoking: GCC Assembler
arm-elf-gcc -x assembler-with-cpp -c -mcpu=arm7tdmi -o"crt/crt.o"
"../crt/crt.s"
Finished building: ../crt/crt.s

Building target: fpga4fun_LEDglow_ram.elf
Invoking: GCC C++ Linker
arm-elf-ld -nostartfiles -T../crt/LPC2138_ram.ld -no-warn-mismatc
-o"fpga4fun_LEDglow_ram.elf"  ./src/main.o  ./crt/crt.o
Finished building target: fpga4fun_LEDglow_ram.elf
```

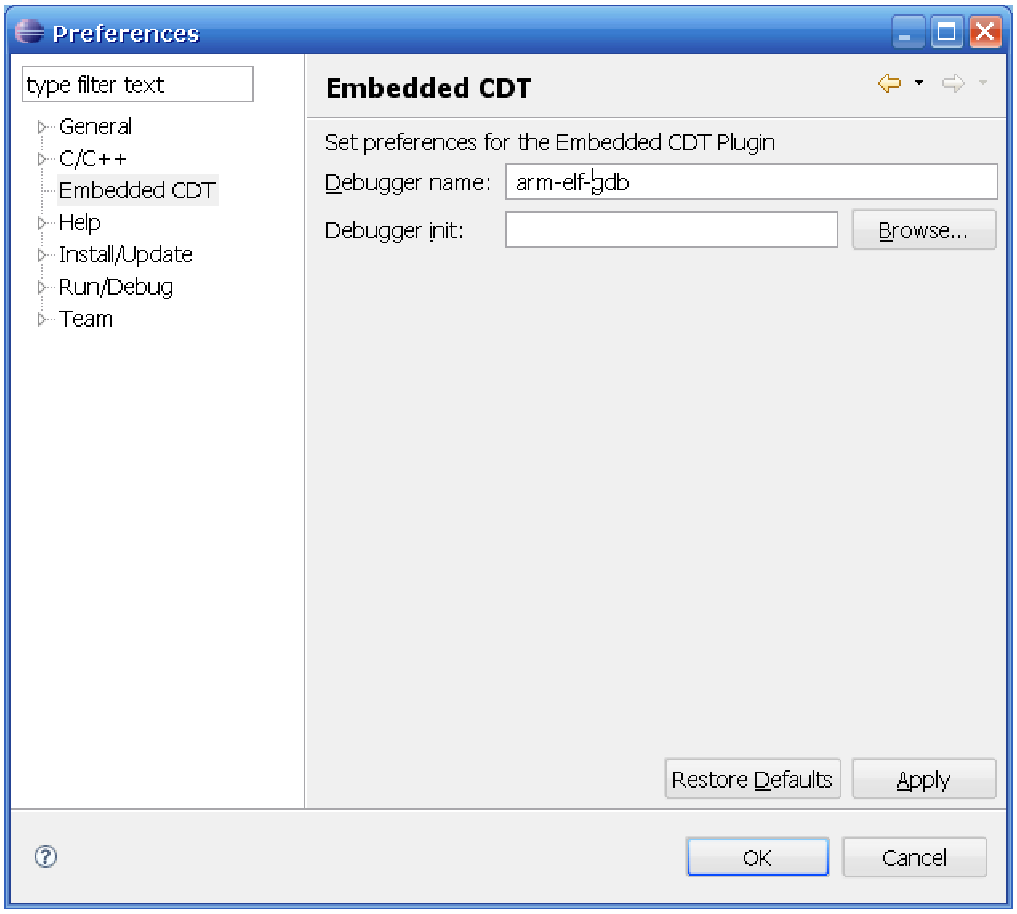
8 Setup selected debugger if not set

Menu window/preferences



```
C-Build [fpga4fun_LEDglow_ram]

Building file: ../crt/crt.s
Invoking: GCC Assembler
arm-elf-gcc -x assembler-with-cpp -c -mcpu=arm7tdmi -o"crt/crt.o"
"../crt/crt.s"
Finished building: ../crt/crt.s

Building target: fpga4fun_LEDglow_ram.elf
Invoking: GCC C++ Linker
arm-elf-ld -nostartfiles -T../crt/LPC2138_ram.ld -no-warn-mismatc
-o"fpga4fun_LEDglow_ram.elf"  ./src/main.o  ./crt/crt.o
Finished building target: fpga4fun_LEDglow_ram.elf
```

9 Type arm-elf-gdb



Preferences

type filter text

- General
- C/C++
- Embedded CDT
- Help
- Install/Update
- Run/Debug
- Team

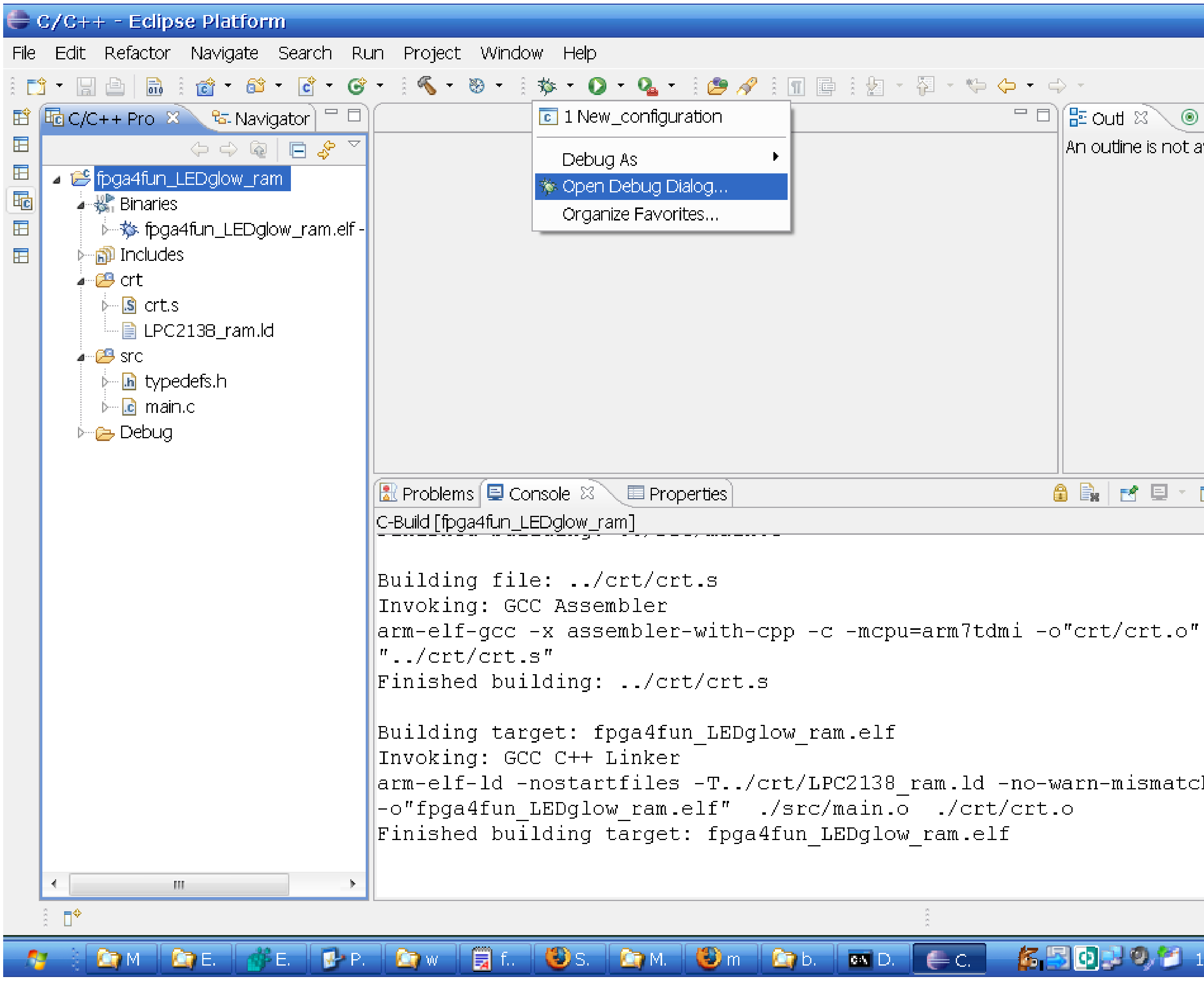
Embedded CDT

Set preferences for the Embedded CDT Plugin

Debugger name: arm-elf-gdb

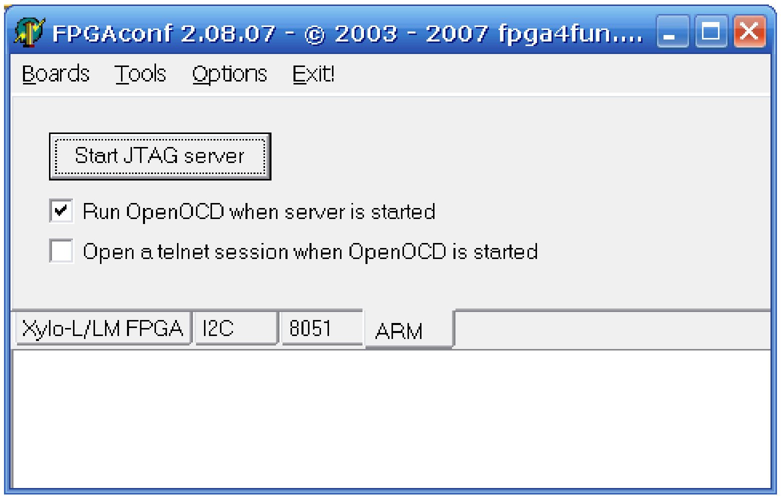
Debugger init:

10 Open debug dialog



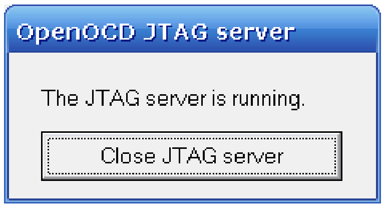
```
C-Build [fpga4fun_LEDglow_ram]

Building file: ../crt/crt.s
Invoking: GCC Assembler
arm-elf-gcc -x assembler-with-cpp -c -mcpu=arm7tdmi -o"crt/crt.o"
"../crt/crt.s"
Finished building: ../crt/crt.s

Building target: fpga4fun_LEDglow_ram.elf
Invoking: GCC C++ Linker
arm-elf-ld -nostartfiles -T../crt/LPC2138_ram.ld -no-warn-mismatc
-o"fpga4fun_LEDglow_ram.elf"  ./src/main.o  ./crt/crt.o
Finished building target: fpga4fun_LEDglow_ram.elf
```

11 Make sure openocd and jtag server are running



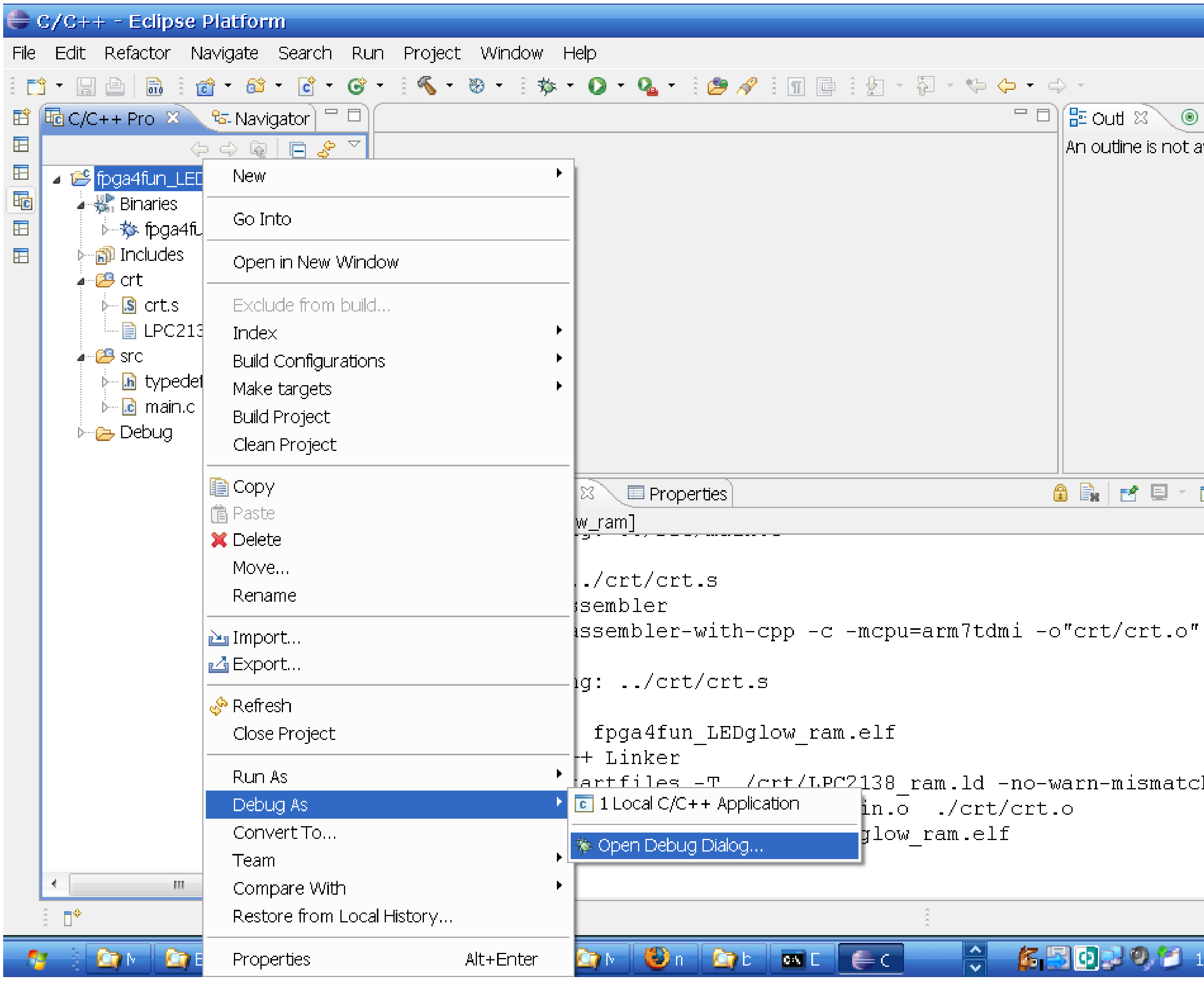
[✓] Run OpenOCD when server is started

[ ] Open a telnet session when OpenOCD is started



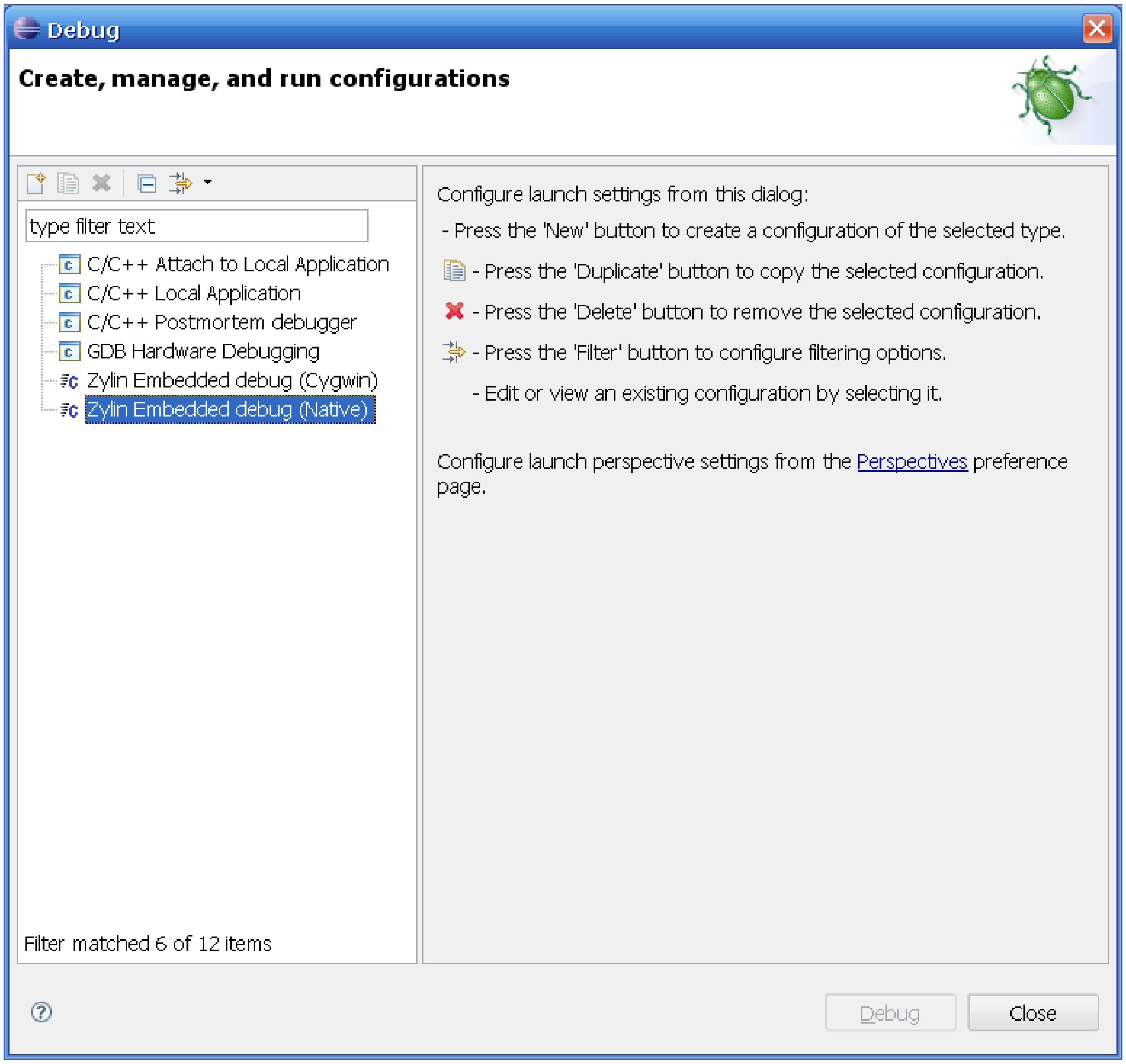
The JTAG server is running.

12 Open debug dialog



```
w_ram]

../crt/crt.s
ssembler
assembler-with-cpp -c -mcpu=arm7tdmi -o"crt/crt.o"

ng: ../crt/crt.s

 fpga4fun_LEDglow_ram.elf
++ Linker
artfiles -T ../crt/LPC2138_ram.ld -no-warn-mismatc
                                   in.o  ./crt/crt.o
                                   glow_ram.elf
```

13 Select Zylin embedded CDT



Create, manage, and run configurations

type filter text

- C/C++ Attach to Local Application
- C/C++ Local Application
- C/C++ Postmortem debugger
- GDB Hardware Debugging
- Zylin Embedded debug (Cygwin)
- Zylin Embedded debug (Native)

Filter matched 6 of 12 items

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.

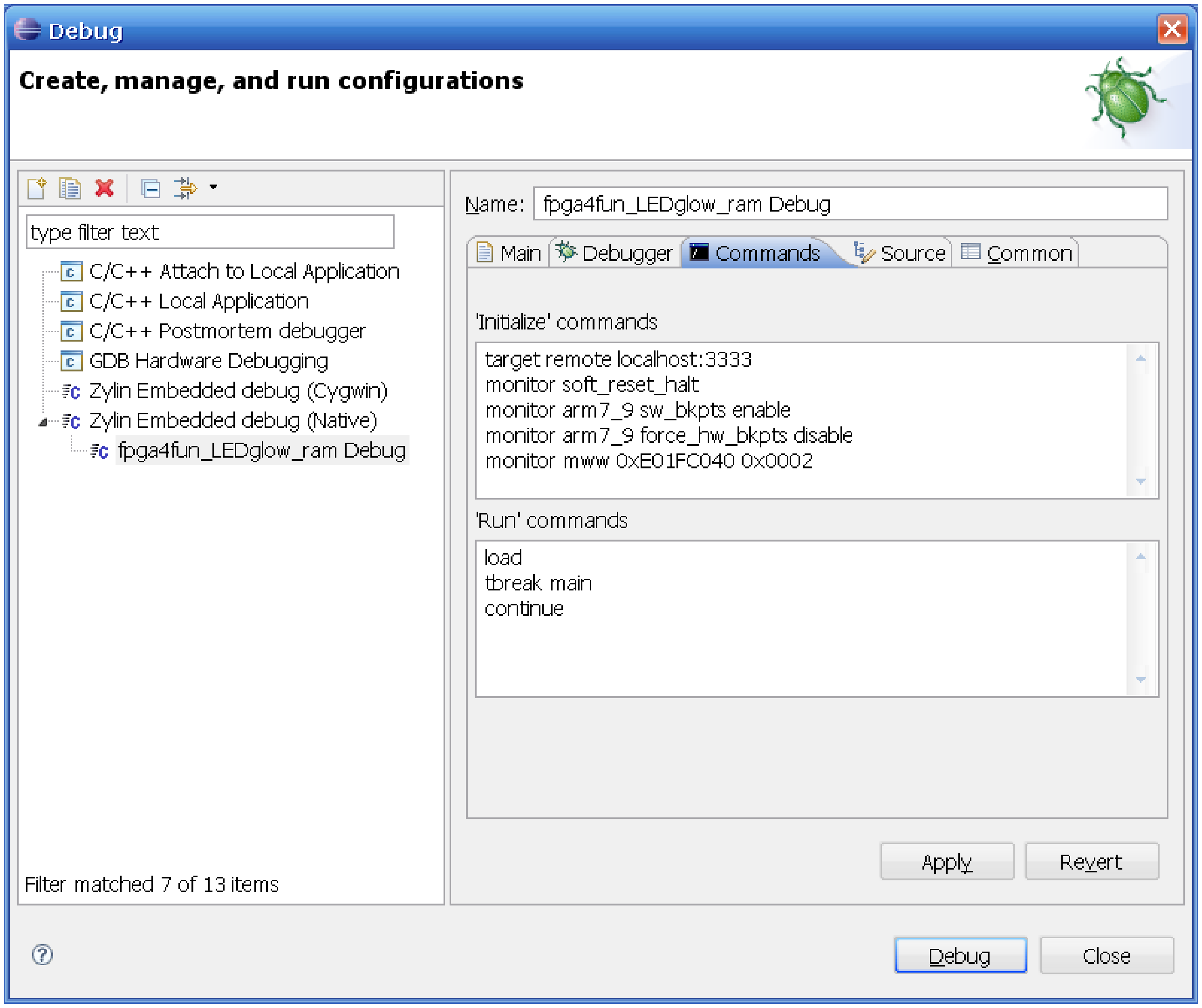
14 Set initalize commands

Set initialize commands to:

```
target remote localhost:3333
monitor soft_reset_halt
monitor arm7_9 sw_bkpts enable
monitor arm7_9 force_hw_bkpts disable
monitor mww 0xE01FC040 0x0002
```

Also set run commands to:

```
load
tbreak main
continue
```


Debug
Create, manage, and run configurations
type filter text
C/C++ Attach to Local Application
C/C++ Local Application
C/C++ Postmortem debugger
GDB Hardware Debugging
Zylin Embedded debug (Cygwin)
Zylin Embedded debug (Native)
fpga4fun_LEDglow_ram Debug
Filter matched 7 of 13 items
Main
Debugger
Commands
Source
Common
Apply
Revert
Debug
Close

Name: fpga4fun_LEDglow_ram Debug

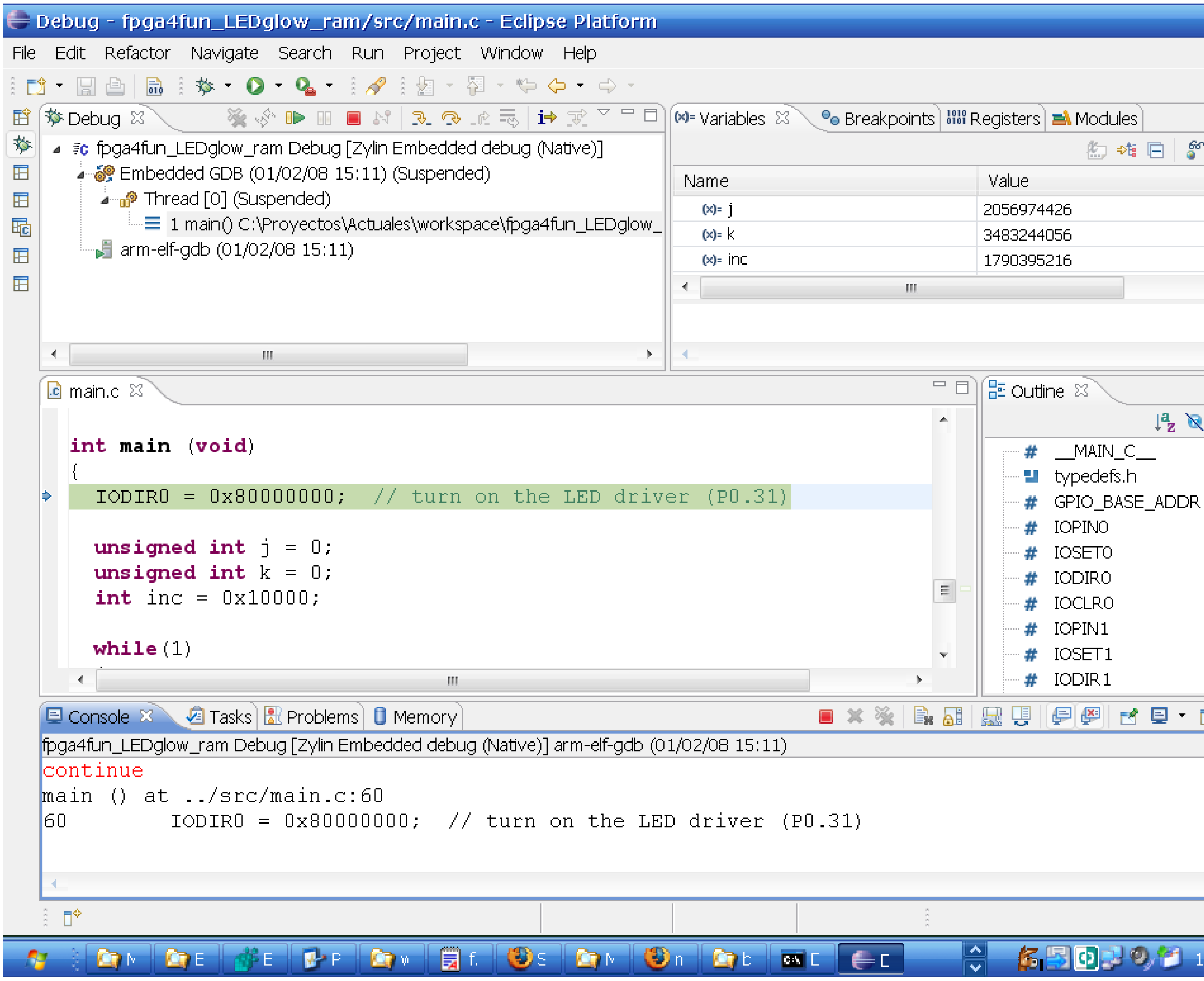
'Initialize' commands

```
target remote localhost:3333
monitor soft_reset_halt
monitor arm7_9 sw_bkpts enable
monitor arm7_9 force_hw_bkpts disable
monitor mww 0xE01FC040 0x0002
```

'Run' commands

```
load
tbreak main
continue
```

15 Press 'Debug'


Debug - fpga4fun_LEDglow_ram/src/main.c - Eclipse Platform
File Edit Refactor Navigate Search Run Project Window Help
Debug
Variables
Breakpoints
Registers
Modules
main.c
Outline
Console
Tasks
Problems
Memory

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)]
  Embedded GDB (01/02/08 15:11) (Suspended)
    Thread [0] (Suspended)
      1 main() C:\Proyectos\Actuales\workspace\fpga4fun_LEDglow_
  arm-elf-gdb (01/02/08 15:11)
```

Name	Value
j	2056974426
k	3483244056
inc	1790395216

```
int main (void)
{
  IODIR0 = 0x80000000;   // turn on the LED driver (P0.31)

  unsigned int j = 0;
  unsigned int k = 0;
  int inc = 0x10000;

  while(1)
```

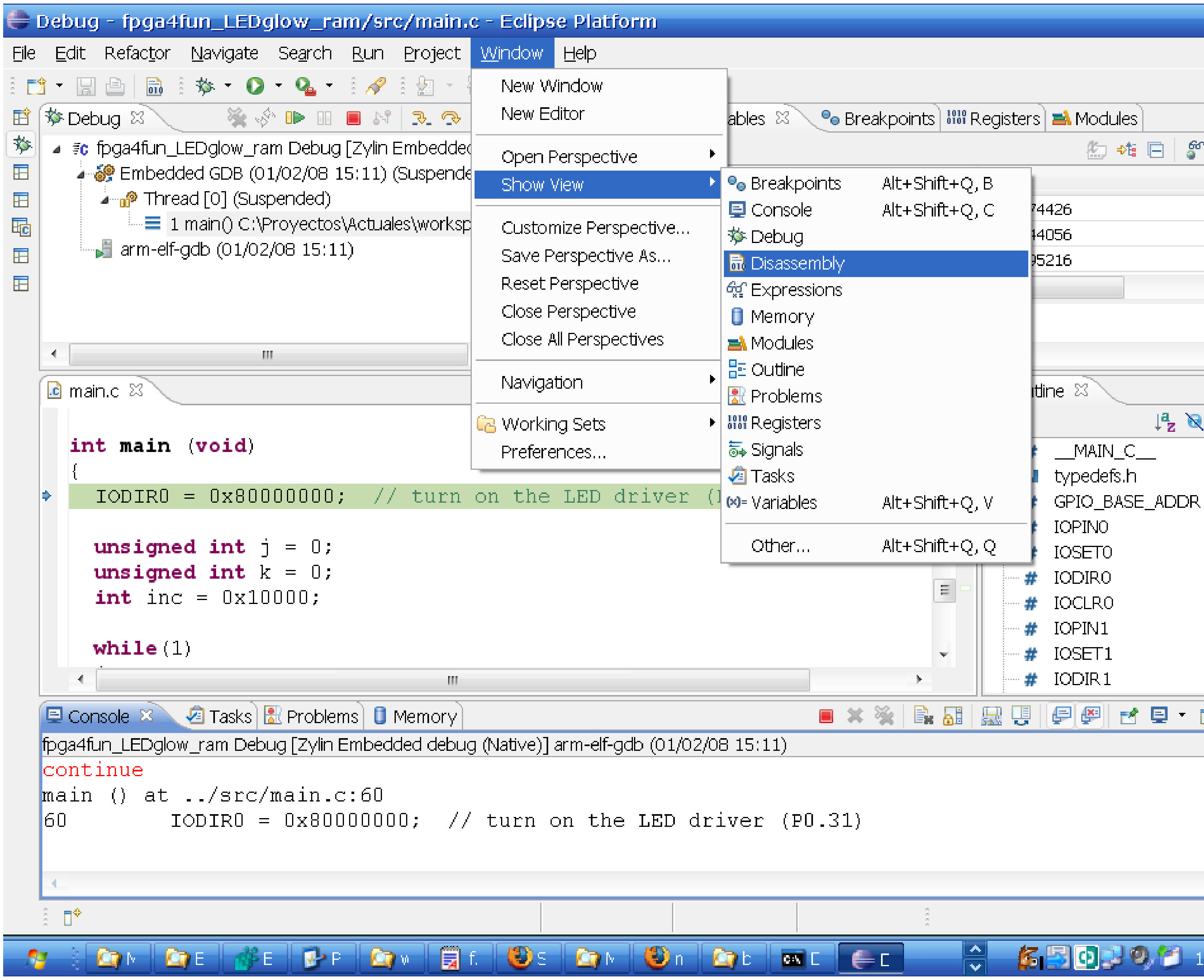
```
__MAIN_C__
typedefs.h
GPIO_BASE_ADDR
IOPIN0
IOSET0
IODIR0
IOCLR0
IOPIN1
IOSET1
IODIR1
```

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)] arm-elf-gdb (01/02/08 15:11)
continue
main () at ../src/main.c:60
60          IODIR0 = 0x80000000;   // turn on the LED driver (P0.31)
```

there is a temporary breakpoint at main, then program stops there.

16 Debug program

For example, Dissasembling shows assemby code intermixed wit C code

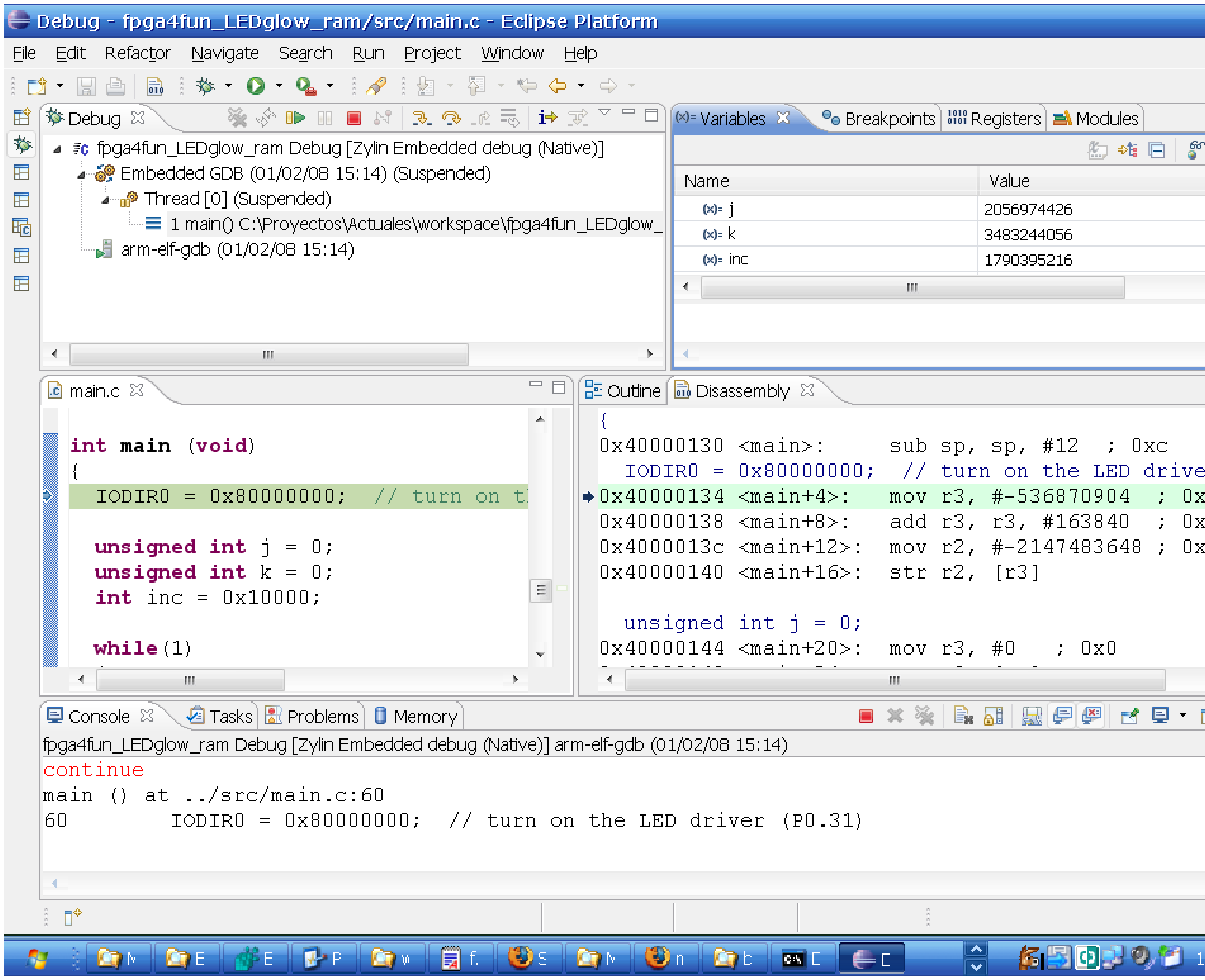


```
int main (void)
{
  IODIR0 = 0x80000000;  // turn on the LED driver (

  unsigned int j = 0;
  unsigned int k = 0;
  int inc = 0x10000;

  while(1)
```

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)] arm-elf-gdb (01/02/08 15:11)
continue
main () at ../src/main.c:60
60          IODIR0 = 0x80000000;  // turn on the LED driver (P0.31)
```

17 Press F6, F5

Press F6, F5 and other eclipse debugger keys in assembly or source code, set breakpoints, show variable values, etc.



Debug - fpga4fun_LEDglow_ram/src/main.c - Eclipse Platform

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)]
  Embedded GDB (01/02/08 15:14) (Suspended)
    Thread [0] (Suspended)
      1 main() C:\Proyectos\Actuales\workspace\fpga4fun_LEDglow_
  arm-elf-gdb (01/02/08 15:14)
```

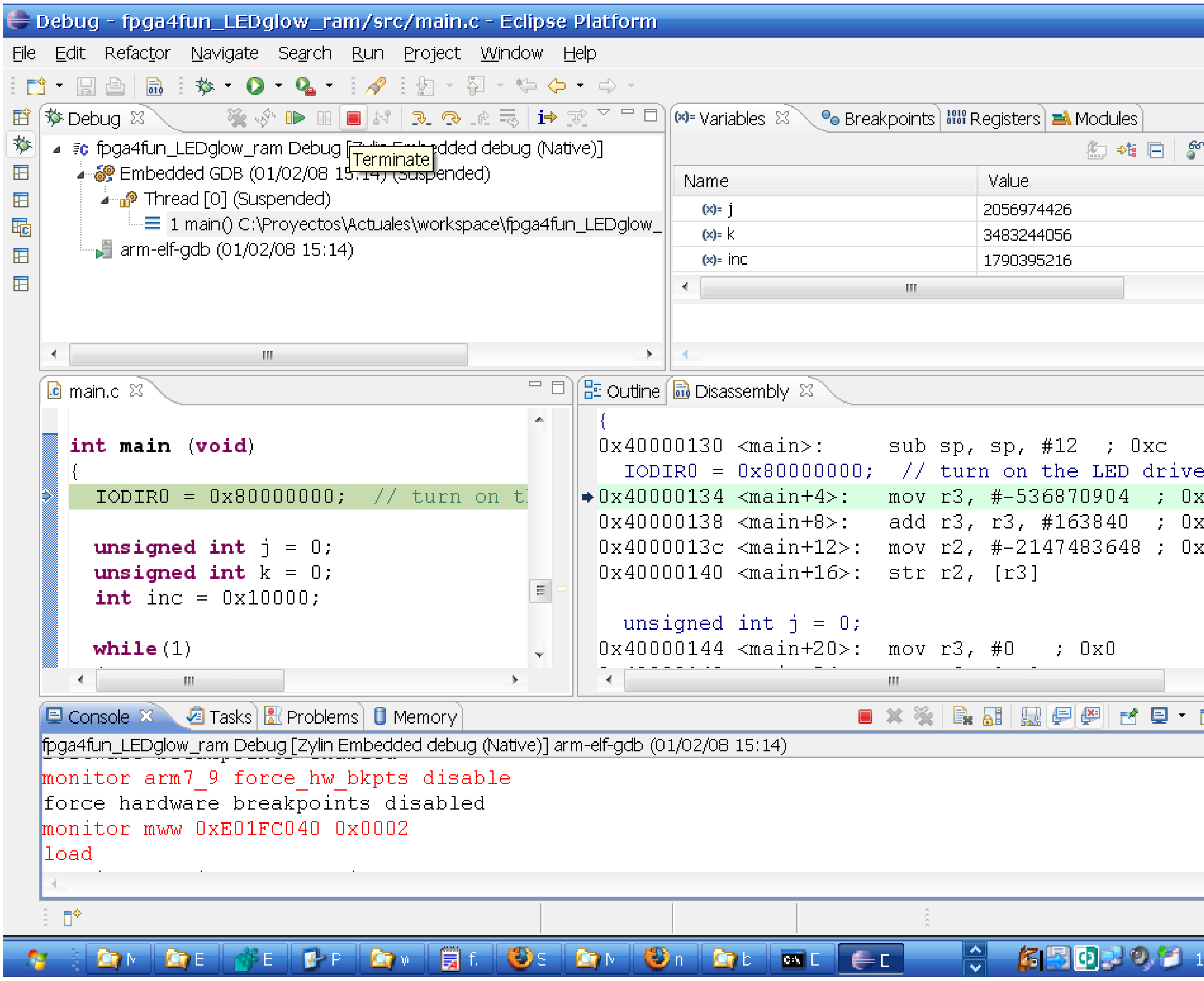
Name	Value
j	2056974426
k	3483244056
inc	1790395216

```
int main (void)
{
  IODIR0 = 0x80000000;  // turn on t

  unsigned int j = 0;
  unsigned int k = 0;
  int inc = 0x10000;

  while(1)
```

```
{
0x40000130 <main>:     sub sp, sp, #12   ; 0xc
  IODIR0 = 0x80000000;  // turn on the LED drive
0x40000134 <main+4>:   mov r3, #-536870904  ; 0x
0x40000138 <main+8>:   add r3, r3, #163840  ; 0x
0x4000013c <main+12>:  mov r2, #-2147483648 ; 0x
0x40000140 <main+16>:  str r2, [r3]

  unsigned int j = 0;
0x40000144 <main+20>:  mov r3, #0    ; 0x0
```

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)] arm-elf-gdb (01/02/08 15:14)
continue
main () at ../src/main.c:60
60        IODIR0 = 0x80000000;  // turn on the LED driver (P0.31)
```

18 Stop running program



Debug - fpga4fun_LEDglow_ram/src/main.c - Eclipse Platform

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)]
  Embedded GDB (01/02/08 15:14) (Suspended)
    Thread [0] (Suspended)
      1 main() C:\Proyectos\Actuales\workspace\fpga4fun_LEDglow_
  arm-elf-gdb (01/02/08 15:14)
```

Name	Value
j	2056974426
k	3483244056
inc	1790395216

```
int main (void)
{
  IODIR0 = 0x80000000;   // turn on t

  unsigned int j = 0;
  unsigned int k = 0;
  int inc = 0x10000;

  while(1)
```

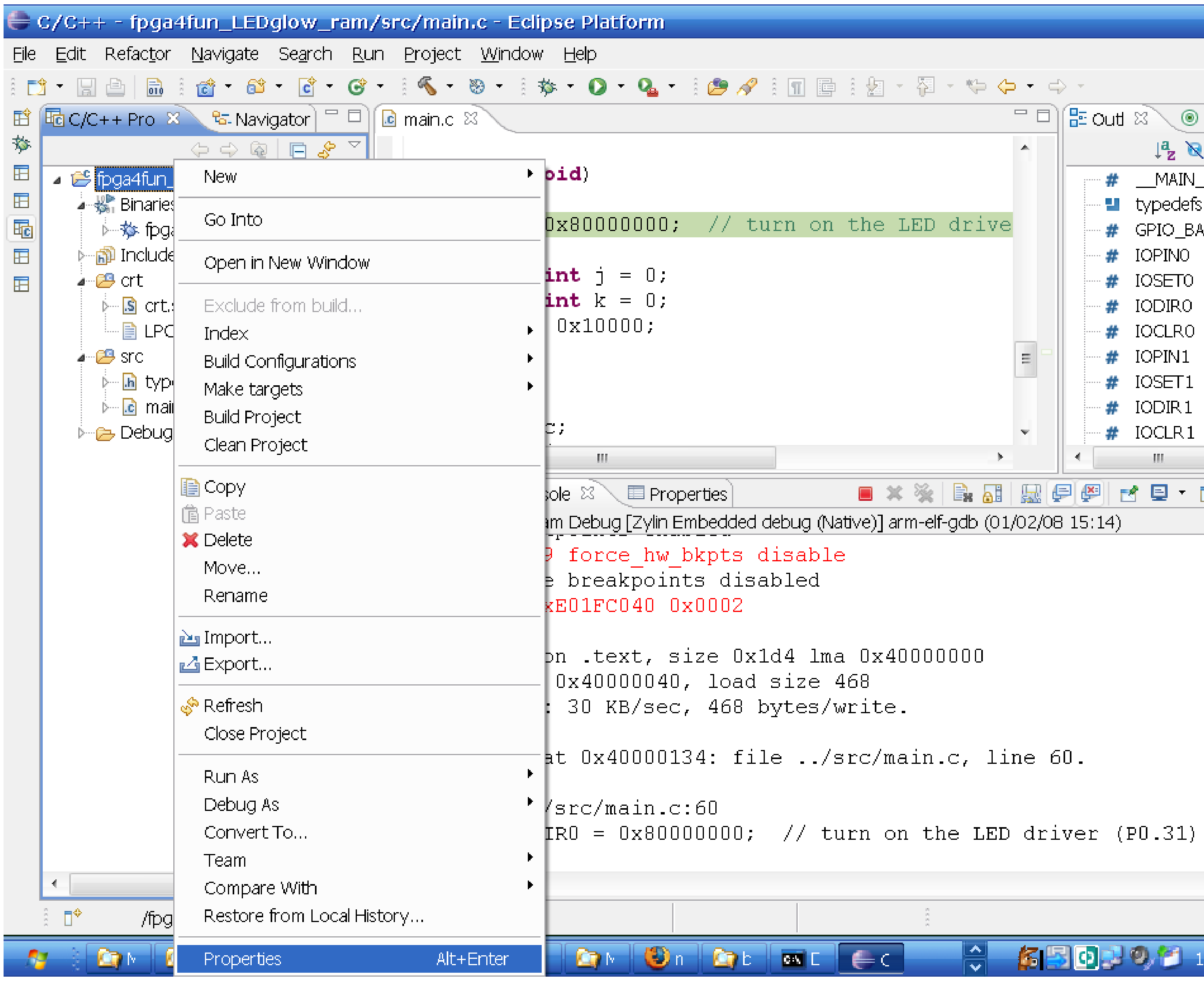
```
{
0x40000130 <main>:     sub sp, sp, #12  ; 0xc
  IODIR0 = 0x80000000;  // turn on the LED drive
0x40000134 <main+4>:   mov r3, #-536870904  ; 0x
0x40000138 <main+8>:   add r3, r3, #163840  ; 0x
0x4000013c <main+12>:  mov r2, #-2147483648 ; 0x
0x40000140 <main+16>:  str r2, [r3]

  unsigned int j = 0;
0x40000144 <main+20>:  mov r3, #0   ; 0x0
```

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)] arm-elf-gdb (01/02/08 15:14)
monitor arm7_9 force_hw_bkpts disable
force hardware breakpoints disabled
monitor mww 0xE01FC040 0x0002
load
```

19 Modify your project to suit your needs

For example if you add a C file to the project's src folder, it will be included in the compilation automatically (thanks CDT managed build project, that generates makefiles).

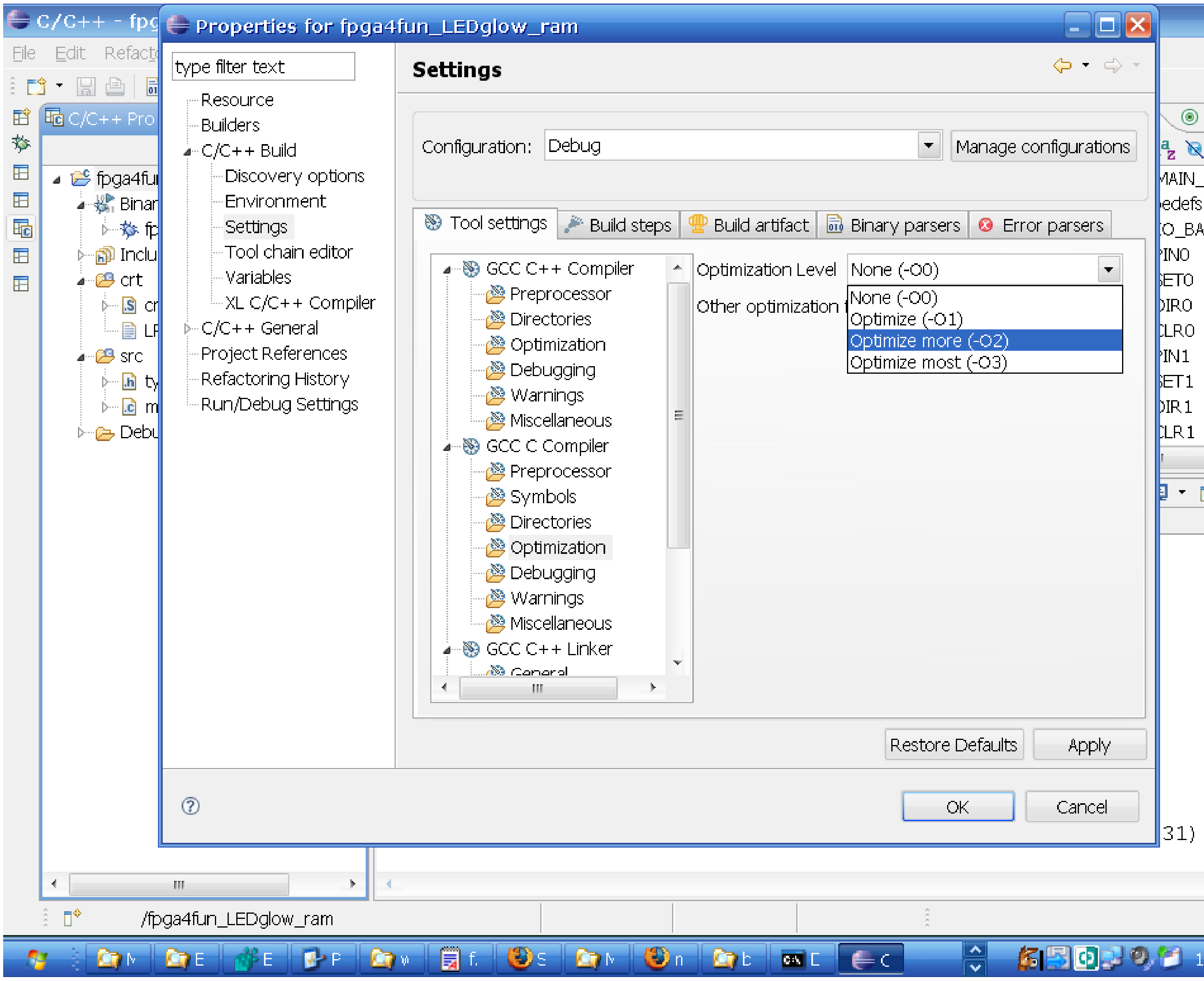
Compiler options are set in the example project to generate detailed assembly listings, debug information, etc. change to suit your needs: right click in your project and set properties:



```
oid)

0x80000000;   // turn on the LED drive

int j = 0;
int k = 0;
 0x10000;



c;
```

```
am Debug [Zylin Embedded debug (Native)] arm-elf-gdb (01/02/08 15:14)
9 force_hw_bkpts disable
e breakpoints disabled
xE01FC040 0x0002

on .text, size 0x1d4 lma 0x40000000
 0x40000040, load size 468
: 30 KB/sec, 468 bytes/write.

at 0x40000134: file ../src/main.c, line 60.

/src/main.c:60
IR0 = 0x80000000;  // turn on the LED driver (P0.31)
```

For example, set optimization level 2


Properties for fpga4fun_LEDglow_ram

Settings

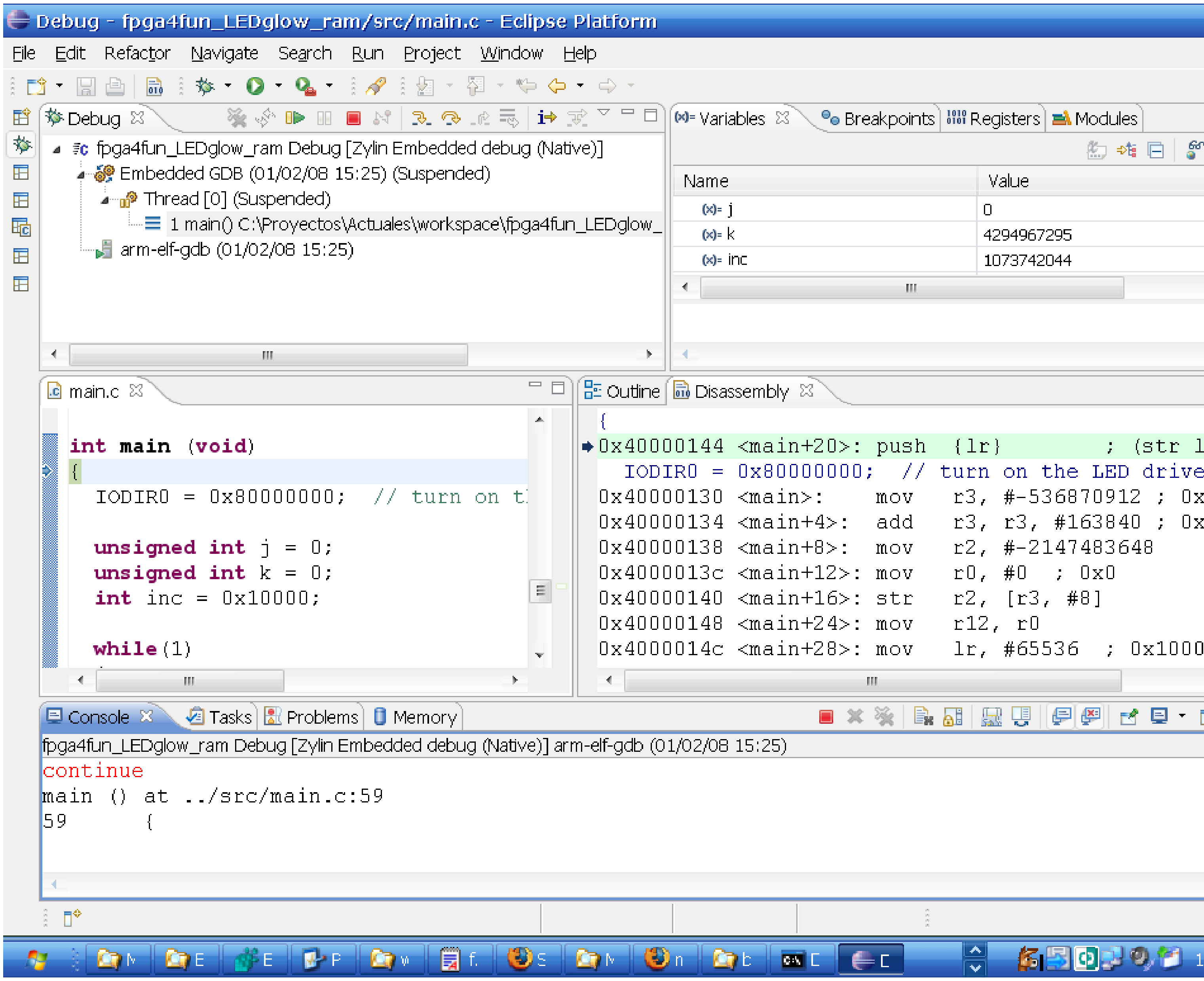
Configuration: Debug

Optimization Level: None (-O0)

None (-O0)
Optimize (-O1)
Optimize more (-O2)
Optimize most (-O3)

20 Press OK and then CTRL-B (or project/build project) to rebuild the project.

Debug it. Assembly listing changed.


Debug - fpga4fun_LEDglow_ram/src/main.c - Eclipse Platform
File Edit Refactor Navigate Search Run Project Window Help
Debug
Variables
Breakpoints
Registers
Modules
main.c
Outline
Disassembly
Console
Tasks
Problems
Memory

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)]
  Embedded GDB (01/02/08 15:25) (Suspended)
    Thread [0] (Suspended)
      1 main() C:\Proyectos\Actuales\workspace\fpga4fun_LEDglow_
  arm-elf-gdb (01/02/08 15:25)
```

Name	Value
j	0
k	4294967295
inc	1073742044

```
int main (void)
{
  IODIR0 = 0x80000000;   // turn on t

  unsigned int j = 0;
  unsigned int k = 0;
  int inc = 0x10000;

  while(1)
```

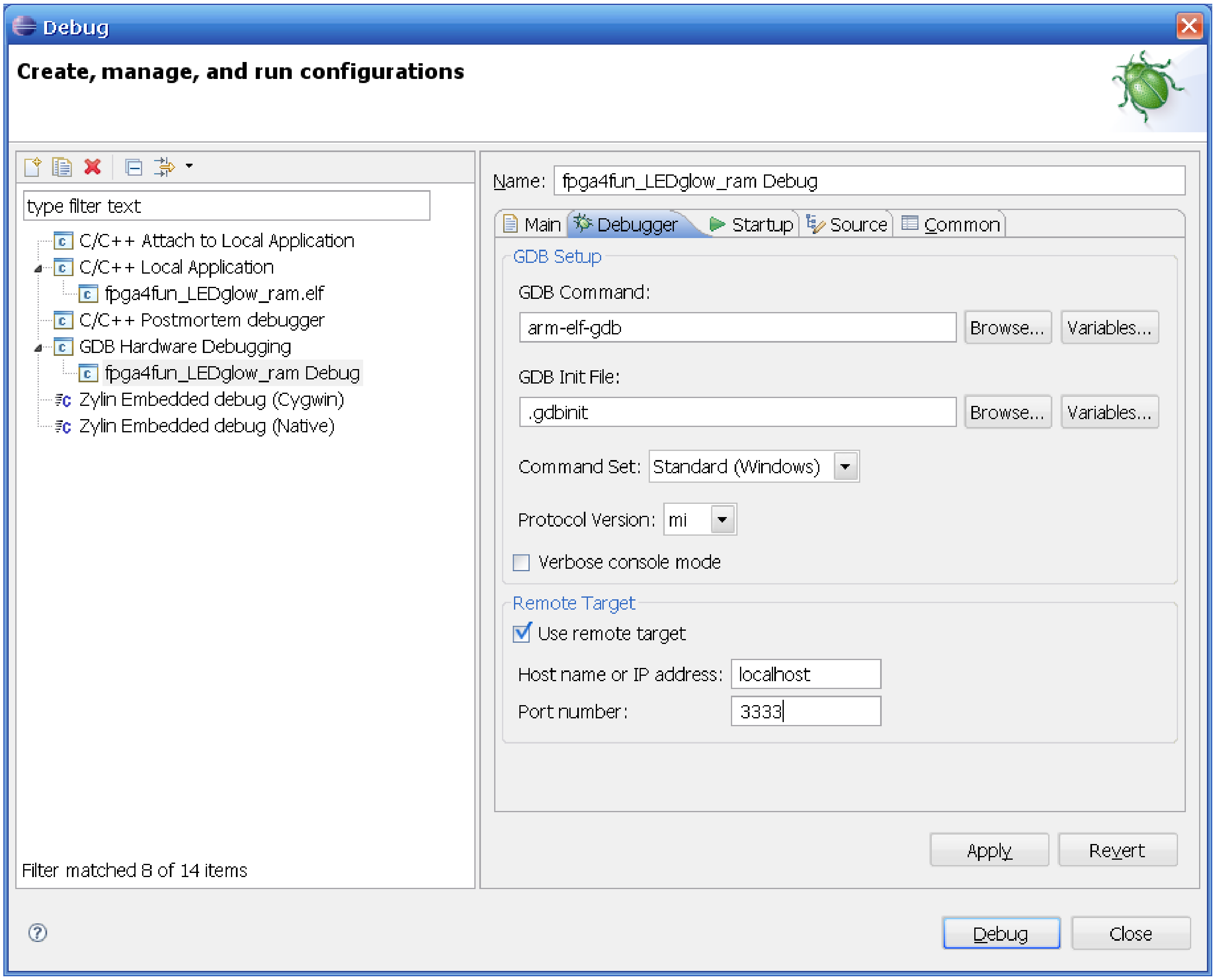
```
{
0x40000144 <main+20>: push  {lr}          ; (str l
  IODIR0 = 0x80000000;  // turn on the LED drive
0x40000130 <main>:    mov   r3, #-536870912 ; 0x
0x40000134 <main+4>:  add   r3, r3, #163840 ; 0x
0x40000138 <main+8>:  mov   r2, #-2147483648
0x4000013c <main+12>: mov   r0, #0   ; 0x0
0x40000140 <main+16>: str   r2, [r3, #8]
0x40000148 <main+24>: mov   r12, r0
0x4000014c <main+28>: mov   lr, #65536   ; 0x1000
```

```
fpga4fun_LEDglow_ram Debug [Zylin Embedded debug (Native)] arm-elf-gdb (01/02/08 15:25)
continue
main () at ../src/main.c:59
59      {
```

21 CDT Hardware debugging

New CDT comes with support for hardware debugging, it could be used instead of Zylin's.

Create a new debuug configuration, set debugger to **arm-elf-gdb** and Port number to **3333**



Debug

Create, manage, and run configurations

type filter text

- C/C++ Attach to Local Application
- C/C++ Local Application
  - fpga4fun_LEDglow_ram.elf
- C/C++ Postmortem debugger
- GDB Hardware Debugging
  - fpga4fun_LEDglow_ram Debug
- Zylin Embedded debug (Cygwin)
- Zylin Embedded debug (Native)

Filter matched 8 of 14 items

Name: fpga4fun_LEDglow_ram Debug

GDB Setup

GDB Command: arm-elf-gdb

GDB Init File: .gdbinit

Command Set: Standard (Windows)

Protocol Version: mi

☐ Verbose console mode

Remote Target

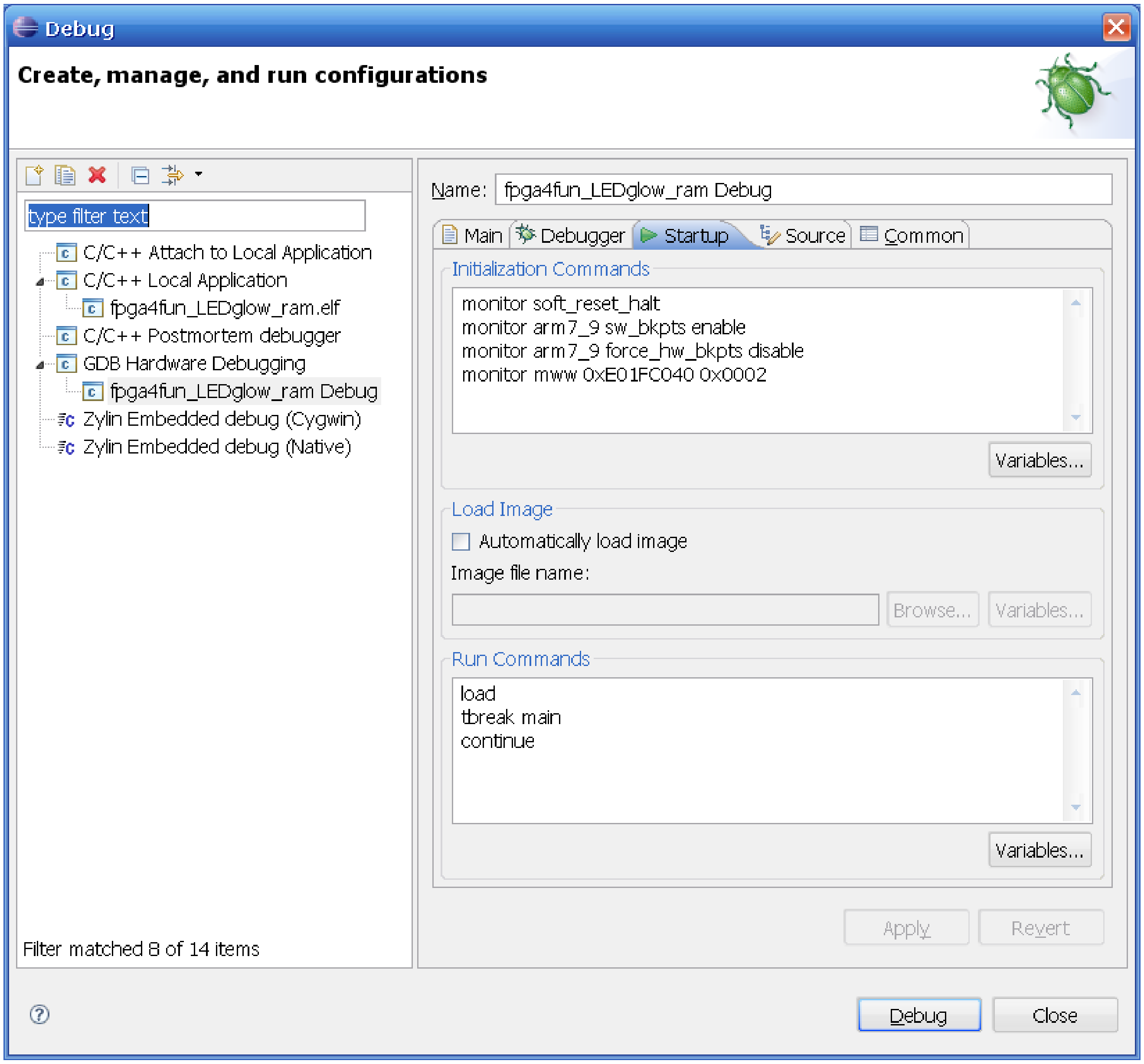
☑ Use remote target

Host name or IP address: localhost

Port number: 3333

22 Startup code

Startup code is basically the same as with Zylin, but you haven’t to specify port number again


Debug
Create, manage, and run configurations
type filter text
C/C++ Attach to Local Application
C/C++ Local Application
fpga4fun_LEDglow_ram.elf
C/C++ Postmortem debugger
GDB Hardware Debugging
fpga4fun_LEDglow_ram Debug
Zylin Embedded debug (Cygwin)
Zylin Embedded debug (Native)
Filter matched 8 of 14 items
Main
Debugger
Startup
Source
Common
Variables...
Browse...
Variables...
Apply
Revert
Debug
Close

Name: fpga4fun_LEDglow_ram Debug

Initialization Commands

```
monitor soft_reset_halt
monitor arm7_9 sw_bkpts enable
monitor arm7_9 force_hw_bkpts disable
monitor mww 0xE01FC040 0x0002
```

Load Image

☐ Automatically load image

Image file name:

Run Commands

```
load
tbreak main
continue
```