Bitfire Software Development

ARMSchool Seminars 2005

2005-10-25

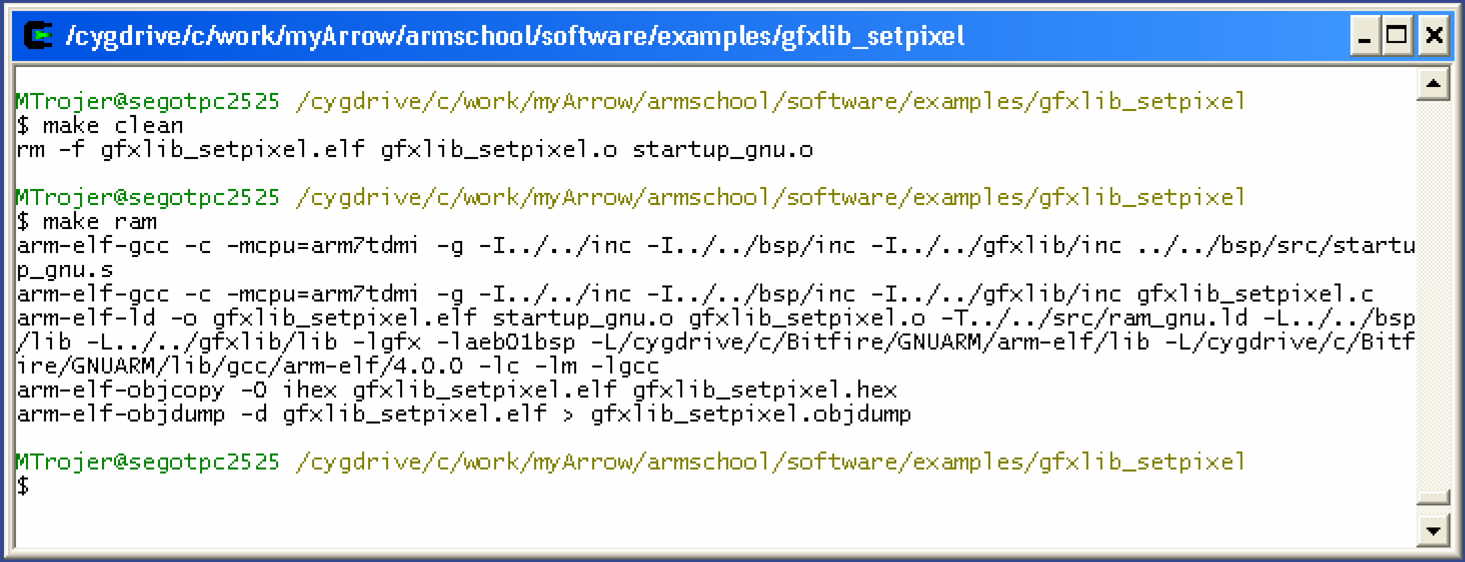
Powering the Supply Chain.SM

Overview

- Compiling C/ASM and linking executable (elf)
  - Makefiles (GCC/LD)
- Running/Debugging
  - OCDRemote + GDB (Insight)
- Project Management + Debug
  - Eclipse
- Examples/RTOS/BSP/GFXLIB

Building with Cygwin

- Makefiles

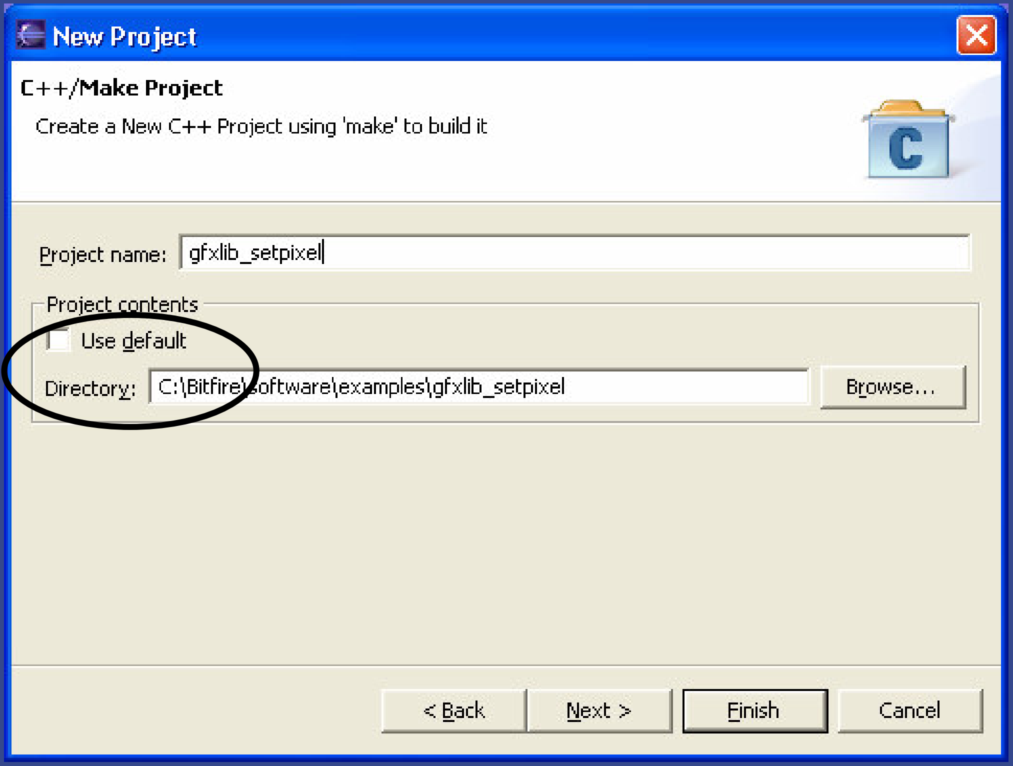


```
MTrojer@segotpc2525 /cygdrive/c/work/myArrow/armschool/software/examples/gfxlib_setpixel
$ make clean
rm -f gfxlib_setpixel.elf gfxlib_setpixel.o startup_gnu.o

MTrojer@segotpc2525 /cygdrive/c/work/myArrow/armschool/software/examples/gfxlib_setpixel
$ make ram
arm-elf-gcc -c -mcpu=arm7tdmi -g -I../../inc -I../../bsp/inc -I../../gfxlib/inc ../../bsp/src/startu
p_gnu.s
arm-elf-gcc -c -mcpu=arm7tdmi -g -I../../inc -I../../bsp/inc -I../../gfxlib/inc gfxlib_setpixel.c
arm-elf-ld -o gfxlib_setpixel.elf startup_gnu.o gfxlib_setpixel.o -T../../src/ram_gnu.ld -L../../bsp
/lib -L../../gfxlib/lib -lgfx -laeb01bsp -L/cygdrive/c/Bitfire/GNUARM/arm-elf/lib -L/cygdrive/c/Bitf
ire/GNUARM/lib/gcc/arm-elf/4.0.0 -lc -lm -lgcc
arm-elf-objcopy -O ihex gfxlib_setpixel.elf gfxlib_setpixel.hex
arm-elf-objdump -d gfxlib_setpixel.elf > gfxlib_setpixel.objdump

MTrojer@segotpc2525 /cygdrive/c/work/myArrow/armschool/software/examples/gfxlib_setpixel
$
```

- Output .elf, .hex, .objdump

Building with Eclipse

- **Menu File->New->Project** and chose **C++/Standard Make C++ project** in the list.



C++/Make Project

Create a New C++ Project using 'make' to build it

Project name: gfxlib_setpixel

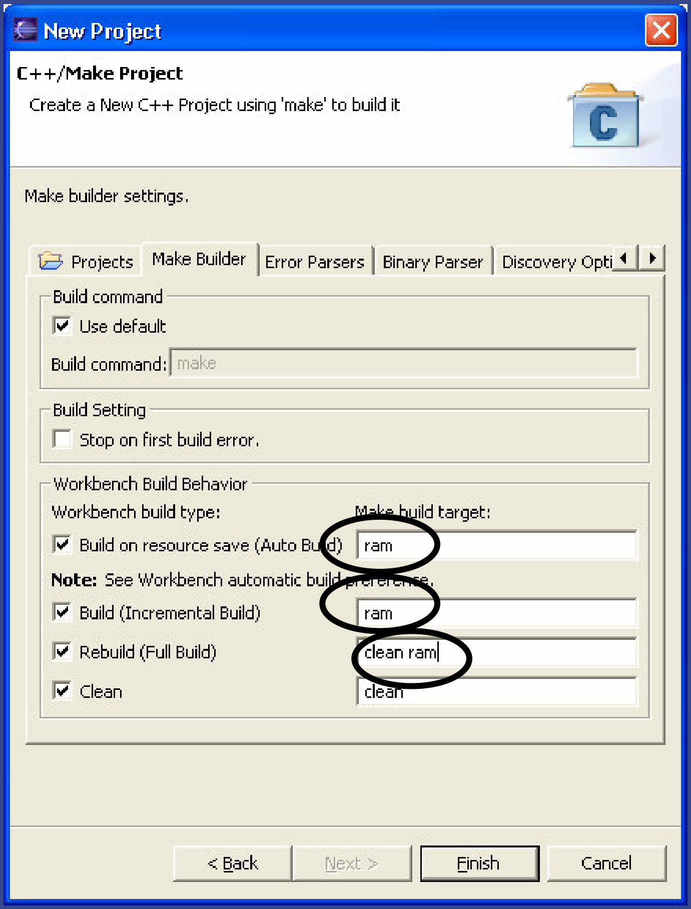
Project contents

□ Use default

Directory: C:\Bitfire\software\examples\gfxlib_setpixel

Building with Eclipse (2)

- Make the following changes in **Make Builder** and **Discovery Options**



New Project

C++/Make Project

Create a New C++ Project using 'make' to build it

Make builder settings.

Build command

☑ Use default

Build command: make

Build Setting

☐ Stop on first build error.

Workbench Build Behavior

Workbench build type: Make build target:

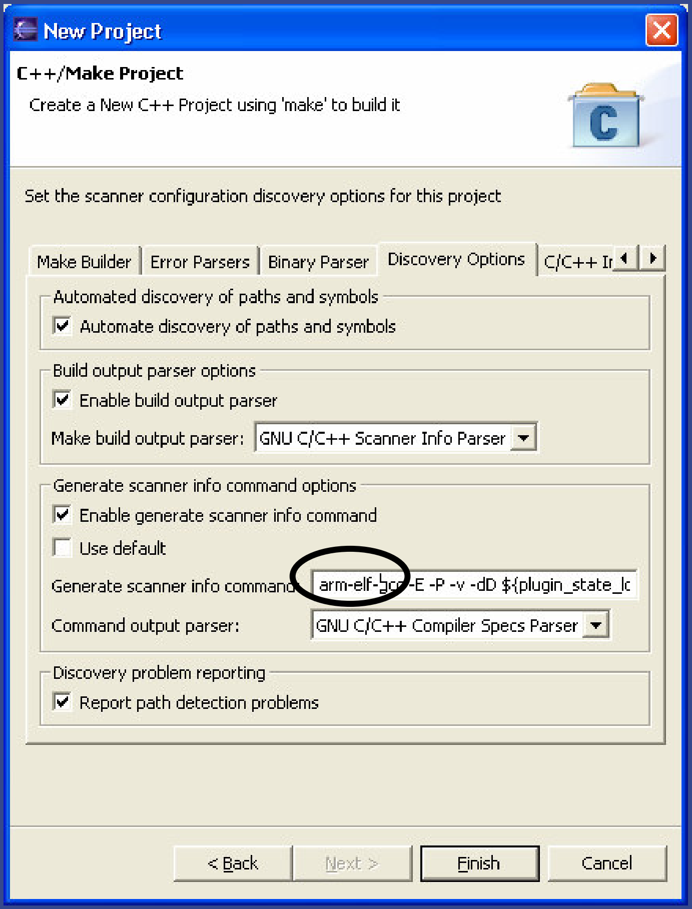
☑ Build on resource save (Auto Build) ram

Note: See Workbench automatic build preference.

☑ Build (Incremental Build) ram

☑ Rebuild (Full Build) clean ram

☑ Clean clean



New Project

C++/Make Project

Create a New C++ Project using 'make' to build it

Set the scanner configuration discovery options for this project

Automated discovery of paths and symbols

☑ Automate discovery of paths and symbols

Build output parser options

☑ Enable build output parser

Make build output parser: GNU C/C++ Scanner Info Parser

Generate scanner info command options

☑ Enable generate scanner info command

☐ Use default

Generate scanner info command: arm-elf-gcc -E -P -v -dD ${plugin_state_lc

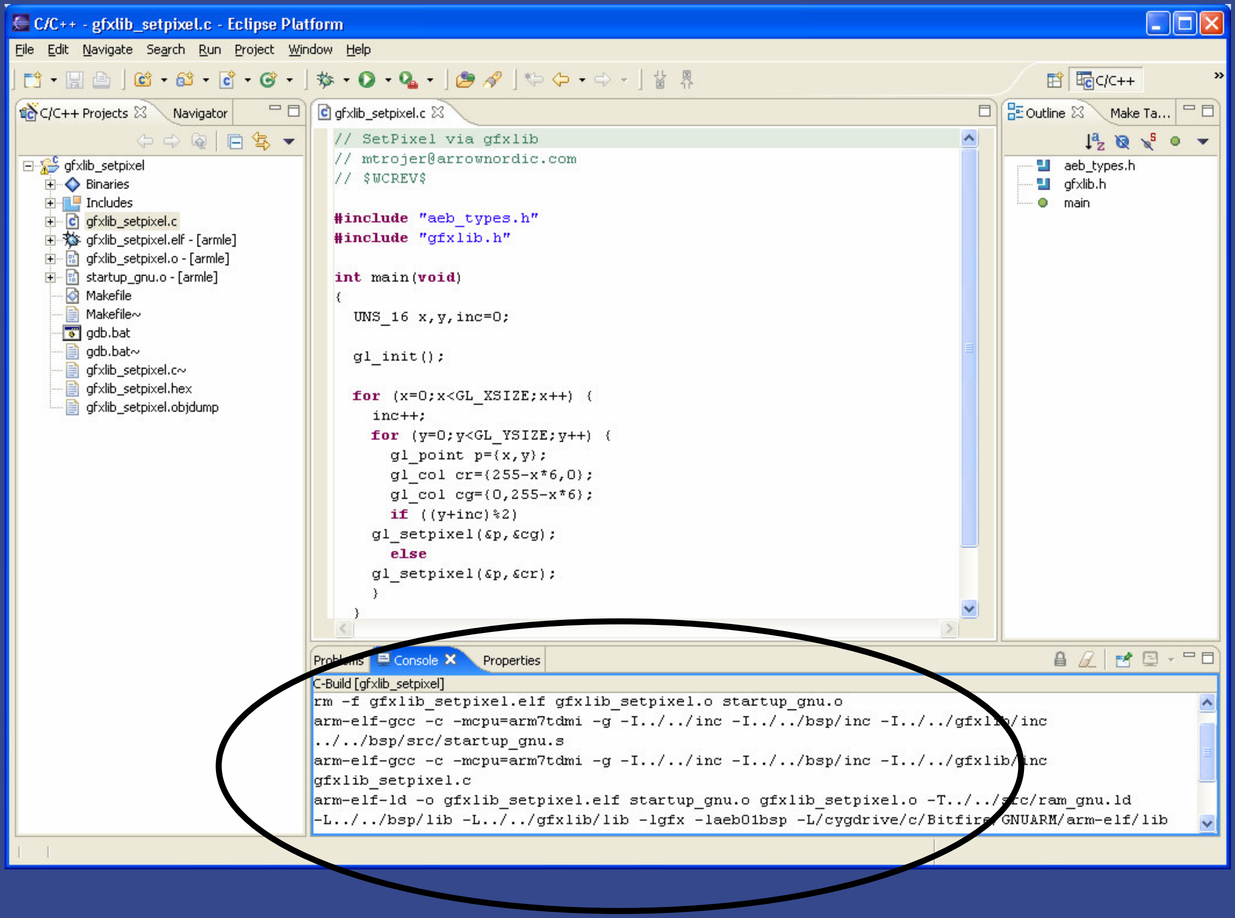
Command output parser: GNU C/C++ Compiler Specs Parser

Discovery problem reporting

☑ Report path detection problems

Building with Eclipse (3)

- Done! Eclipse uses the exact same Makefiles



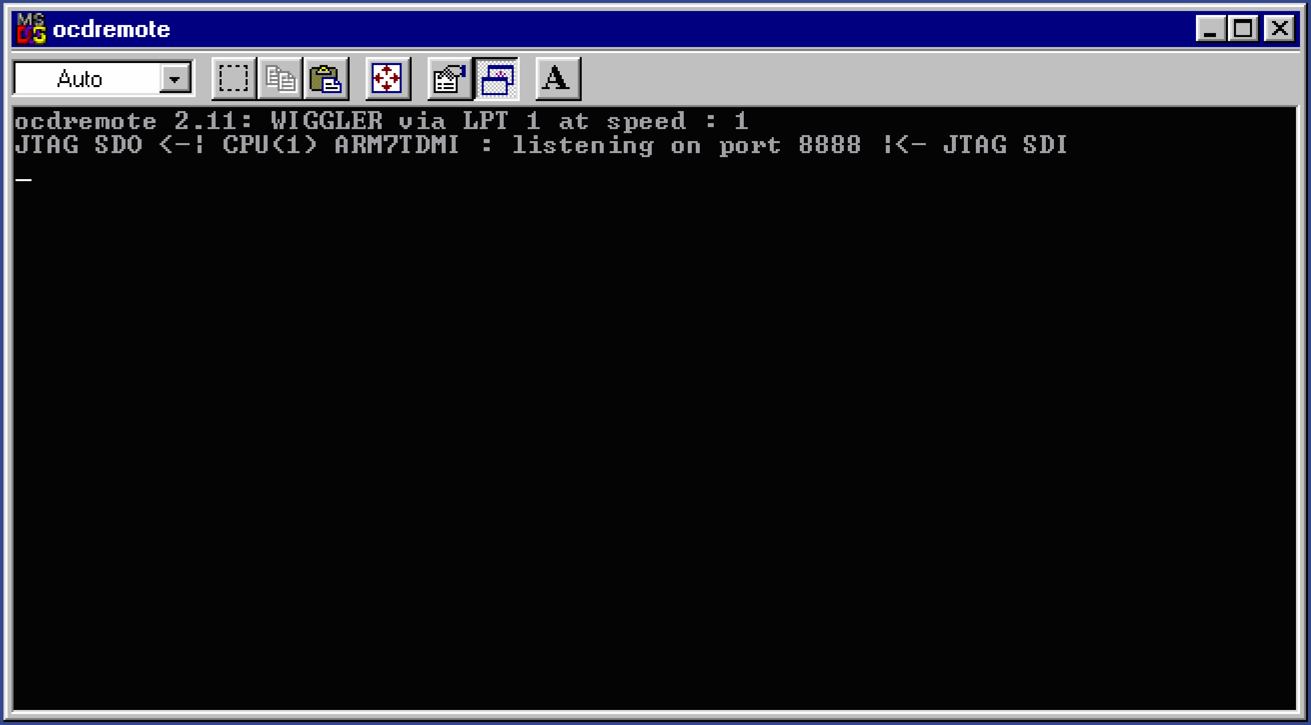
```
// SetPixel via gfxlib
// mtrojer@arrownordic.com
// $WCREV$

#include "aeb_types.h"
#include "gfxlib.h"

int main(void)
{
  UNS_16 x,y,inc=0;

  gl_init();

  for (x=0;x<GL_XSIZE;x++) {
    inc++;
    for (y=0;y<GL_YSIZE;y++) {
      gl_point p={x,y};
      gl_col cr={255-x*6,0};
      gl_col cg={0,255-x*6};
      if ((y+inc)%2)
    gl_setpixel(&p,&cg);
      else
    gl_setpixel(&p,&cr);
    }
  }
```

C-Build [gfxlib_setpixel]

```
rm -f gfxlib_setpixel.elf gfxlib_setpixel.o startup_gnu.o
arm-elf-gcc -c -mcpu=arm7tdmi -g -I../../inc -I../../bsp/inc -I../../gfxlib/inc
../../bsp/src/startup_gnu.s
arm-elf-gcc -c -mcpu=arm7tdmi -g -I../../inc -I../../bsp/inc -I../../gfxlib/inc
gfxlib_setpixel.c
arm-elf-ld -o gfxlib_setpixel.elf startup_gnu.o gfxlib_setpixel.o -T../../src/ram_gnu.ld
-L../../bsp/lib -L../../gfxlib/lib -lgfx -laeb01bsp -L/cygdrive/c/Bitfire/GNUARM/arm-elf/lib
```

Running/Debugging

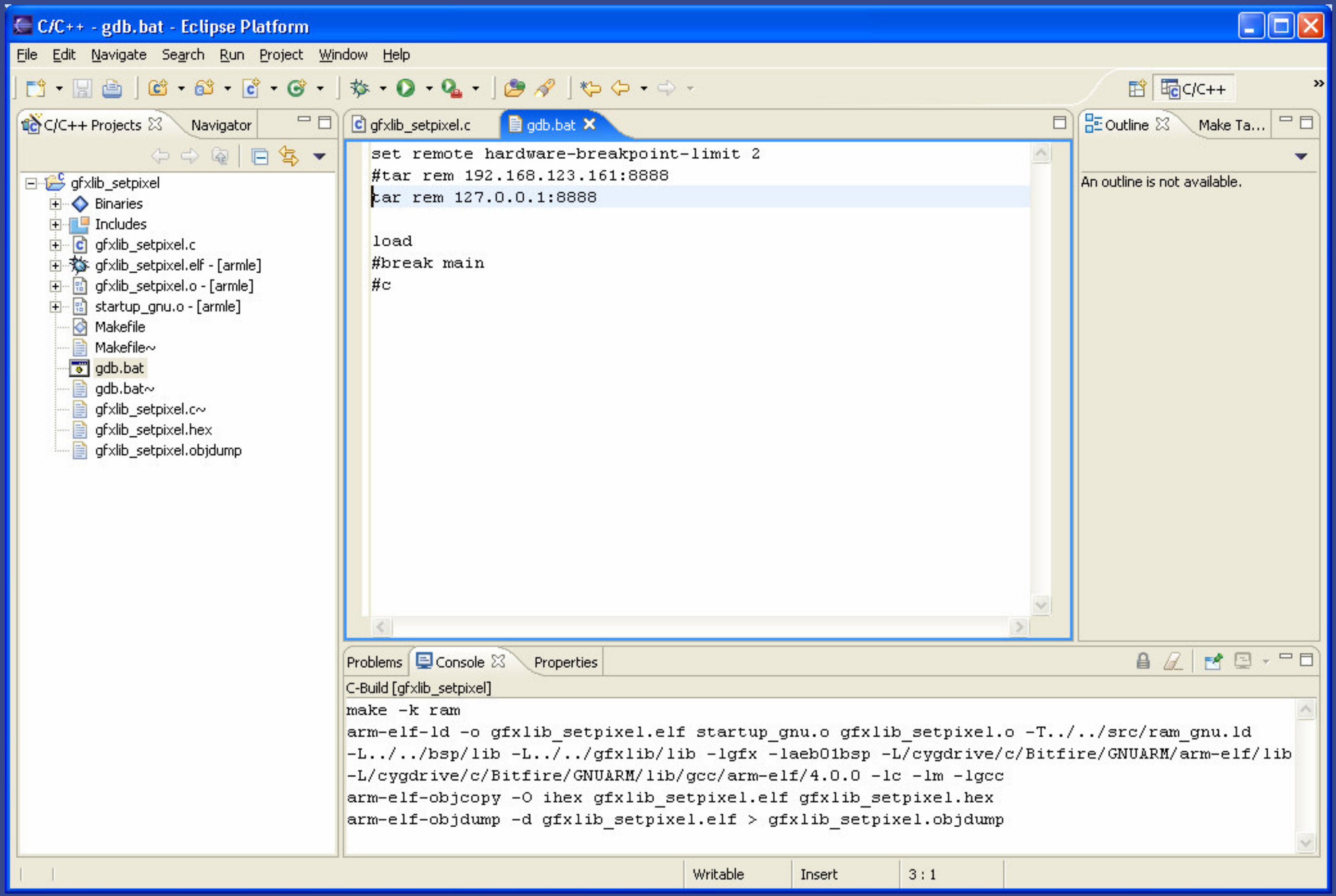
1. Parallel port in EPP (or ECP) mode in Bios.
2. Bitfire in “Wiggler mode”
3. Start OCDRemote `c:\Bitfire\ocdremote\ordremote.exe -c arm7tdmi -d wiggler`



```
ocdremote 2.11: WIGGLER via LPT 1 at speed : 1
JTAG SDO <-¦ CPU(1) ARM7TDMI  : listening on port 8888 ¦<- JTAG SDI
_
```

Running/Debugging with Eclipse

- Edit the file gdb.bat (only once)

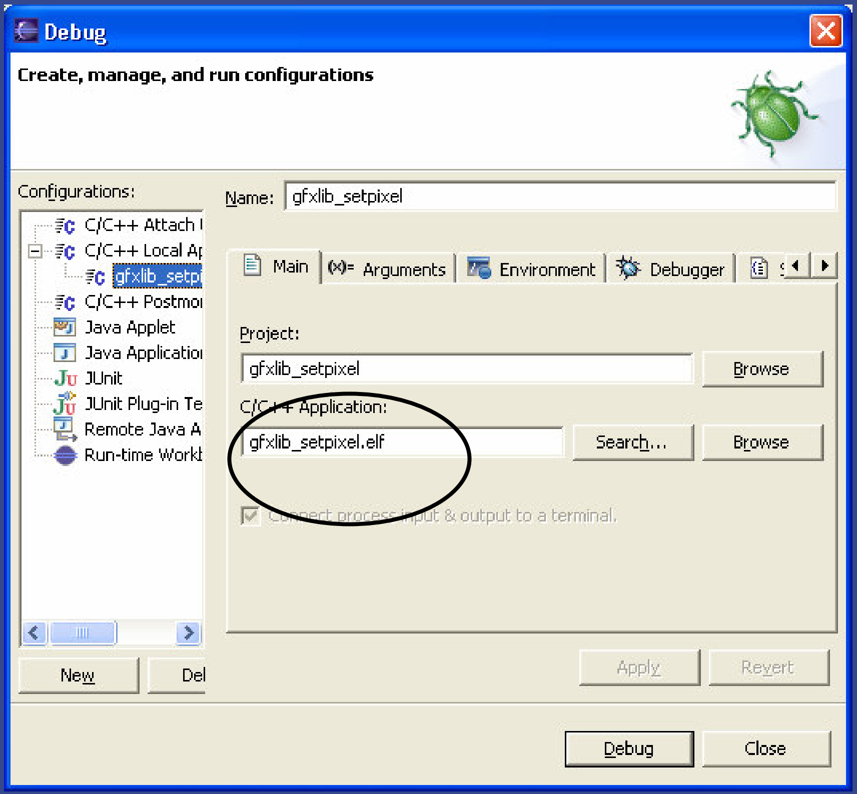


```
set remote hardware-breakpoint-limit 2
#tar rem 192.168.123.161:8888
tar rem 127.0.0.1:8888

load
#break main
#c
```

```
make -k ram
arm-elf-ld -o gfxlib_setpixel.elf startup_gnu.o gfxlib_setpixel.o -T../../src/ram_gnu.ld
-L../../bsp/lib -L../../gfxlib/lib -lgfx -laeb01bsp -L/cygdrive/c/Bitfire/GNUARM/arm-elf/lib
-L/cygdrive/c/Bitfire/GNUARM/lib/gcc/arm-elf/4.0.0 -lc -lm -lgcc
arm-elf-objcopy -O ihex gfxlib_setpixel.elf gfxlib_setpixel.hex
arm-elf-objdump -d gfxlib_setpixel.elf > gfxlib_setpixel.objdump
```

Running/Debugging with Eclipse (2)

- **Run->Debug…**
- C/C++ Local Application + New



Debug

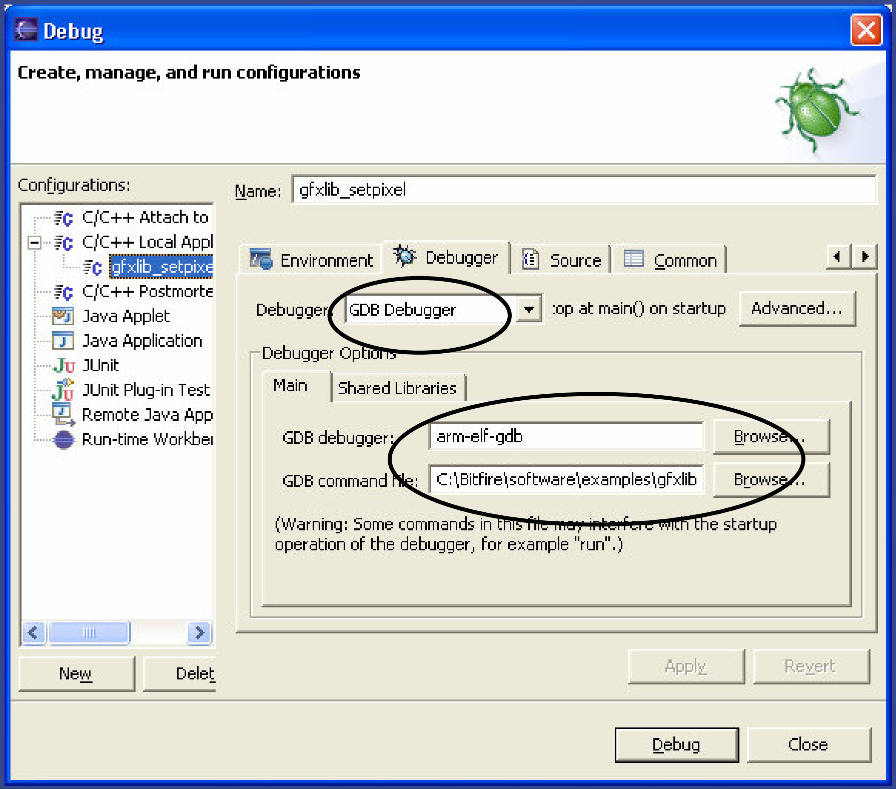
Create, manage, and run configurations

Name: gfxlib_setpixel

Project: gfxlib_setpixel

C/C++ Application: gfxlib_setpixel.elf

☑ Connect process input & output to a terminal.



Debug

Create, manage, and run configurations

Name: gfxlib_setpixel

Debugger: GDB Debugger

:op at main() on startup

Debugger Options

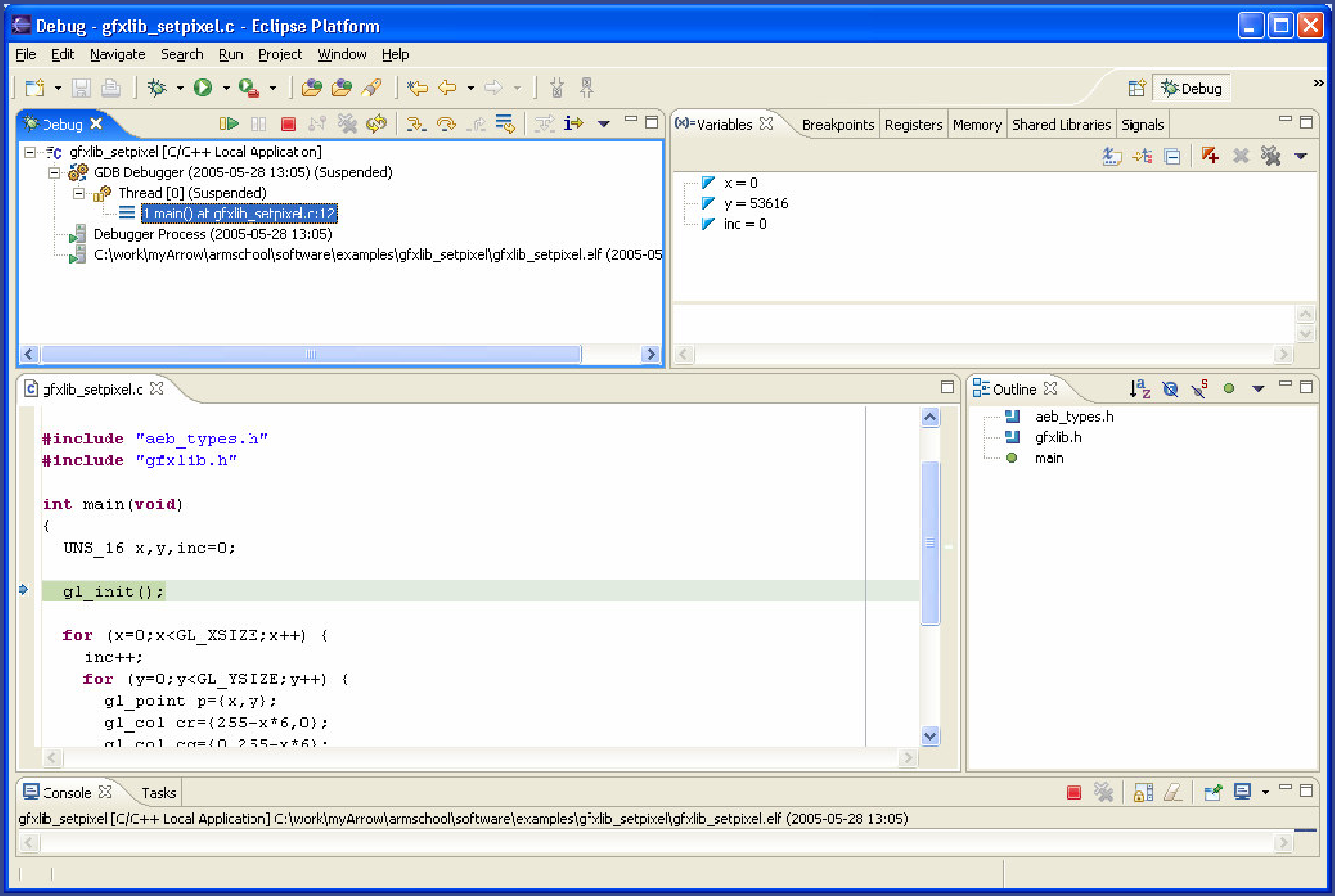
GDB debugger: arm-elf-gdb

GDB command file: C:\Bitfire\software\examples\gfxlib

(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)

Running/Debugging with Eclipse (3)

- Done!


Debug - gfxlib_setpixel.c - Eclipse Platform
File Edit Navigate Search Run Project Window Help
Debug
Debug
Variables
Breakpoints
Registers
Memory
Shared Libraries
Signals
Outline
Console
Tasks

```
gfxlib_setpixel [C/C++ Local Application]
  GDB Debugger (2005-05-28 13:05) (Suspended)
    Thread [0] (Suspended)
      1 main() at gfxlib_setpixel.c:12
  Debugger Process (2005-05-28 13:05)
  C:\work\myArrow\armschool\software\examples\gfxlib_setpixel\gfxlib_setpixel.elf (2005-05
```

```
x = 0
y = 53616
inc = 0
```

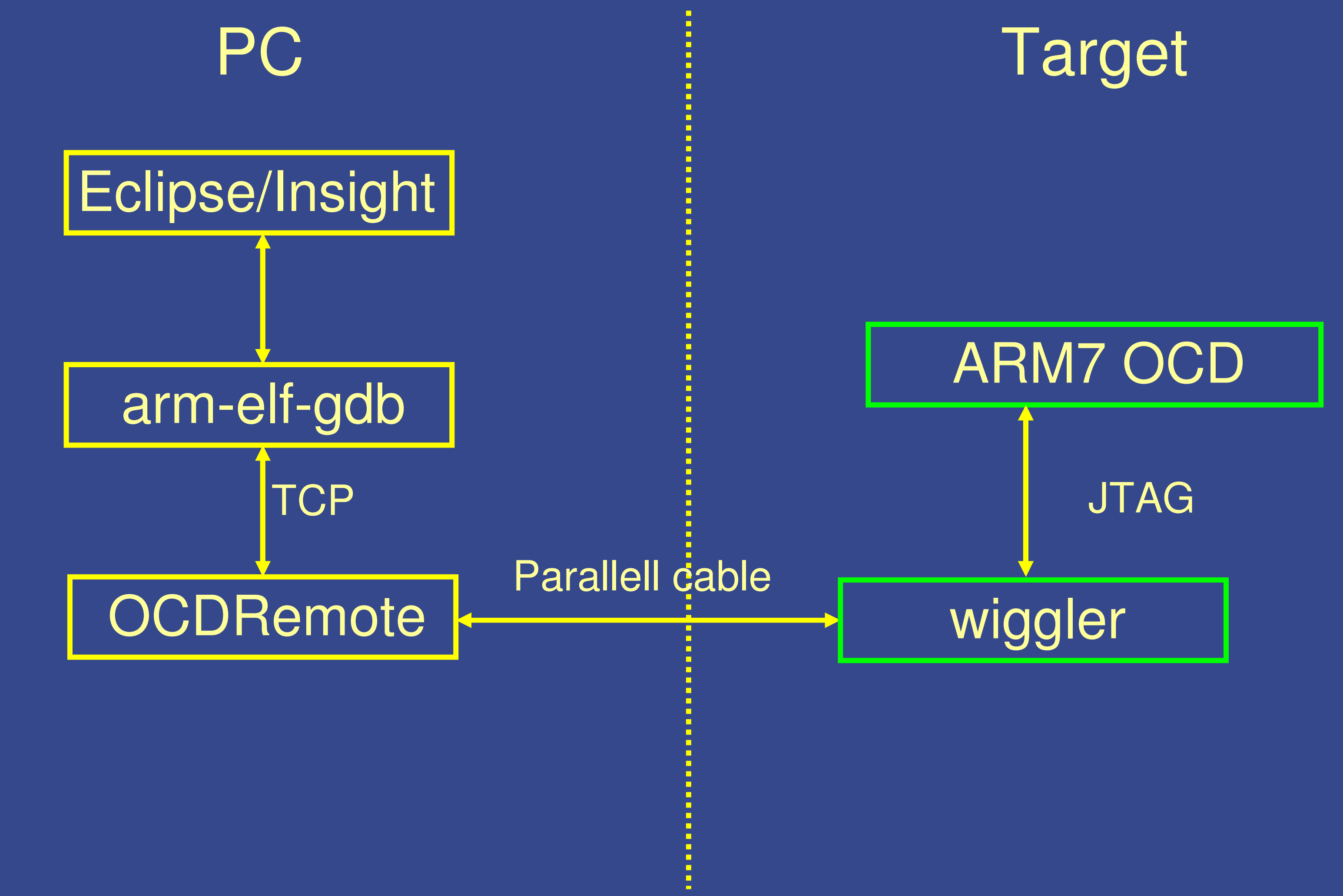
gfxlib_setpixel.c

```
#include "aeb_types.h"
#include "gfxlib.h"

int main(void)
{
  UNS_16 x,y,inc=0;

  gl_init();

  for (x=0;x<GL_XSIZE;x++) {
    inc++;
    for (y=0;y<GL_YSIZE;y++) {
      gl_point p={x,y};
      gl_col cr={255-x*6,0};
      gl_col cg={0,255-x*6};
```

```
aeb_types.h
gfxlib.h
main
```

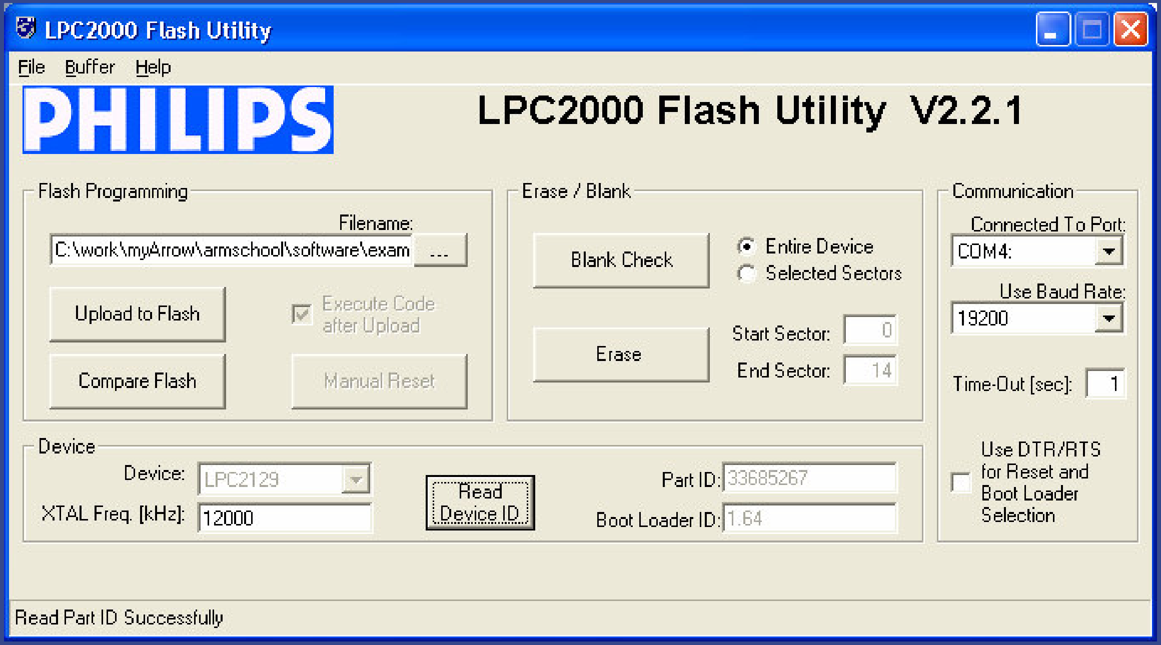
```
gfxlib_setpixel [C/C++ Local Application] C:\work\myArrow\armschool\software\examples\gfxlib_setpixel\gfxlib_setpixel.elf (2005-05-28 13:05)
```

Debug session



Running from Flash

- make (not make ram)
- Bitfire in Bootloader mode
- Use LPC2000 to burn .hex file into flash.



LPC2000 Flash Utility V2.2.1

Filename: C:\work\myArrow\armschool\software\exam

☑ Execute Code after Upload

● Entire Device
○ Selected Sectors

Start Sector: 0
End Sector: 14

Connected To Port: COM4:
Use Baud Rate: 19200
Time-Out [sec]: 1
□ Use DTR/RTS for Reset and Boot Loader Selection

Device: LPC2129
XTAL Freq. [kHz]: 12000

Part ID: 33685267
Boot Loader ID: 1.64

Read Part ID Successfully

Debugging from Flash

- OCDRemote is not good at hardware breakpoints and stepping.
- We need to control this ourselves with GDB
- Thus command-driven debugging, yeah!

```
arm-elf-gdb gfxlib_setpixel.elf (--command gdb.bat)
(gdb) target remote 127.0.0.1:8888
(gdb) set $pc=0
(gdb) hbreak main
(gbd) c
```

- No stepping! (only stepi available)
- Use hbreak + ‘c’ debugging

GDB commands

• i br (info breakpoints)	- Lists your breakpoints.
• del x (delete x)	- Delete breakpoint X
• i reg (info registers)	- Show registers
• i s (info stack)	- Show call stack
• i loc (infor local)	- Show local variables
• x 0x123456789 (examine)	- Look a memory at location 0x12345678
• p i (print expression)	- Print result of generic expression, can be anything from function calls to variables etc.
• c (continue)	- continue execution
• b <file><line><symbol>	- software breakpoint
• hb <file><line><symbol>	- hardware breakpoint

The Examples

- `software/examples`
- `make` in each example to build
    - can_polled
      An example of the polled can driver.
    - Demoreel
      A demo reel with effects and scrolling text.
    - gfxlib_effects
      An example of how to setup and run the different effects.
    - gfxlib_ghs
      A GreenHills MULTI project with the graphics lib and BSP.
    - gfxlib_setpixel
      An example of how to set a pixel with the graphics library.
    - gfxlib_text
      An example of how to write text to the screen with the graphics library.
    - Gpio
      An example of how to control the LPC2129 gpio pins.
    - spi_pixeltest
      An example of how to set pixels without using the graphics library.
    - spi_polled
      An example of how to use the spi peripheral.
    - Testsuite
      A complete testsuite for all the different peripherals on the Bitfire board, including the CAN interface.
    - uart_int
      An example of the interrupt-driven uart driver.
    - uart_polled
      An example of the polled uart driver.

The BSP

- `software/bsp`
- `make` in bsp/lib to build
  - aeb01.h
    Defines the speed of the CPU and VPB clock.
  - aeb01_fpga.h
    Functions for sending commands and data to the Bitfire FPGA. This is controlled by GPIO P0.12 and P0.13.
    Defines FGPACMD_* constants.
  - lpc2129_can.h
    Common include file for CAN drivers. Defines CAN_* constants. Defines struct can_control
  - lpc2129_can_polled.h
    Polled UART driver
  - lpc2129_spi.h
    Common include file for SPI drivers. Defines SPCR_* and SPI_* constants. Defines struct spi_control
  - lpc2129_spi_polled.h
    Polled SPI driver
  - lpc2129_uart.h
    Common include file for UART drivers. Defines LCR_*, UART_* constants. Defines struct uart_control
  - lpc2129_uart_int.h
    Interrupt-driven UART driver
  - lpc2129_uart_polled.h
    Polled UART driver
  - lpc2129_vic.h
    Vectored Interrupt Controller (VIC) Driver. Defines VIC_* constants. Defines struct vic_control
  - lpc21xx.h
    All register addresses for LPC2129

The Graphics Library

- `software/gfxlib`
- `make` in gfxlib/lib to build
    - gfxfx.h

      The Graphics Library effects part.
    - gfxlib.h

      The Graphics Library main functions and structs.
    - gfxtxt.h

      The Graphics Library text part.
    - gfxvec.h

      The Graphics Library 3D vector part.

RTOS

- `software/rtos`
- FreeRTOS
  Example in FreeRTOS/Demo/ARM7_LPC2129_GCC
  Use make to build
- uC/OS-II
  Example in MICRIUM/SOFTWARE/EvalBoards/Philips/Bitfire/GCC/app
  Use make to build

Lab 1

- Compile and run a program using Eclipse and GreenHills.
- The program should set a pixel on the display.

- A directory has been setup for lab 1 (with makefile and lab1.c).
- `software/workshop/lab1`
- `software/workshop/lab1_ghs`

- `See the Software Developers Guide, BUT use workshop/lab1 instead of examples/gfxlib_setpixel`