

Quick Start Guide

ZY1000 JTAG Debugger



Version 2.00 – March 1, 2012

Index

1 ZY1000.....	2
1.1 Warranty.....	2
1.2 Compliance.....	2
2 Basic ZY1000 setup.....	3
2.1 Set TCP/IP address.....	3
2.2 Connect the ZY1000 to your LAN.....	5
2.3 Network discovery.....	5
2.4 Access ZY1000 Web-server.....	6
2.5 Check for firmware updates.....	7
3 Using the ZY1000 Web Interface.....	8
3.1 Connecting the ZY1000 to a Target.....	8
3.2 Target Configuration.....	9
3.3 Edit Target Configuration.....	11
3.4 Programming Flash.....	12
3.5 Browsing and editing memory.....	13
4 Telnet server.....	14
4.1 Connecting to the Telnet Server.....	14
5 GDB Server.....	16
5.1 Using GDB interface with text frontend.....	16
5.2 Using GDB interface with GUI (Eclipse).....	17

Ultimate Solutions, Inc.

<http://www.ultsol.com>

info@ultsol.com

10 Clever Drive

Tewksbury, MA 01876

USA

1 ZY1000

1.1 Warranty

The ZY1000 comes with a 1 year warranty for hardware failures due to material or production defects, under normal use from date of original retail purchase of the product.

1.2 Compliance

RoHS

The ZY1000 is RoHS compliant.

CE/FCC

The ZY1000 is not CE nor FCC certified, and is intended for lab use only. Care should be taken when the unit is used outside of a lab environment.

2 Basic ZY1000 setup

2.1 Set TCP/IP address



Illustration 1: ZY1000 Serial Connection

You can connect the ZY1000 to your computer using your serial cable and set up a static IP address. If you do nothing, the network configuration is taken from a DHCP server. Once the IP address is configured, disconnect the serial cable.

Connect the serial cable and the power supply cable to the ZY1000 as shown in the illustration above. The other end of the serial cable should be connected to your computer.

The power supply can be used for both 110V 60 Hz and 220V 50 Hz.

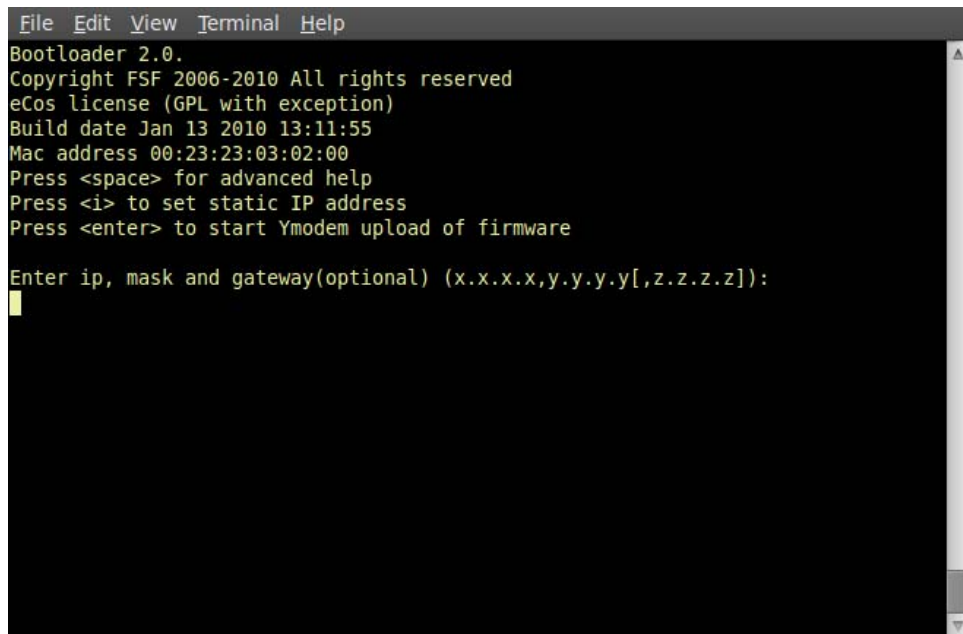
When you have connected the ZY1000 to your computer you will need to start a serial terminal program, like HyperTerm for Windows or minicom for Linux.

Set up the terminal program to use the correct serial port, and setup the serial port with these setting:

- **Baudrate: 38400**
- **Databit: 8**
- **Parity: None**
- **Stopbit: 1**
- **Flow-control: None**

When you power on your ZY1000 you should see something similar to the text below in your terminal program.

Minicom output shown here:



```
File Edit View Terminal Help
Bootloader 2.0.
Copyright FSF 2006-2010 All rights reserved
eCos license (GPL with exception)
Build date Jan 13 2010 13:11:55
Mac address 00:23:23:03:02:00
Press <space> for advanced help
Press <i> to set static IP address
Press <enter> to start Ymodem upload of firmware

Enter ip, mask and gateway(optional) (x.x.x.x,y.y.y.y[,z.z.z.z]):
|
```

Illustration 2: Setting IP address.

Press 'I' (capital 'i') when the ZY1000 is booting to set the IP address in the format:

```
IP           Subnet Mask      Gateway
192.168.0.100,255.255.255.0 [,192.168.0.1]
(This is an example IP address)
```

2.2 Connect the ZY1000 to your LAN



Illustration 3: ZY1000 Ethernet Connection.

Connect the ZY1000 to your local network (LAN) as shown above.

The ZY1000 must have the right type of patch cable. If you connect to a switch you should use a normal patch cable, but if you connect the ZY1000 directly to your computer you will need a crossed patch cable. The Green LED on the ZY1000 Ethernet connector should light up when a valid Ethernet link is established.

Note! ZY1000 *only supports 100Mbit*, make sure that the ZY1000 is connected to a 100Mbit capable switch!

2.3 Network discovery

The ZY1000 1.47 or later will broadcast its presence every two seconds for the first minute after boot (UDP port 4950). This can be used to find the IP address of the ZY1000.

Download discover.jar from <http://www.ultsol.com/ZY1000/discover.jar> and run it from the command prompt as you can see in Figure 4 and then power cycle the ZY1000. You can then see that the ZY1000 is on IP address 10.0.0.69. Note that the ZY1000 stops broadcasting after 60 seconds. The discover.jar works on all operating systems with Java installed.

```

C:\WINDOWS\system32\cmd.exe - java -jar discover.jar
C:\temp>
C:\temp>
C:\temp>
C:\temp>
C:\temp>
C:\temp>
C:\temp>
C:\temp>
C:\temp>
C:\temp>java -jar discover.jar
10.0.0.69: ZY1000 Zylin JTAG debugger
10.0.0.69: ZY1000 Zylin JTAG debugger
10.0.0.69: ZY1000 Zylin JTAG debugger
10.0.0.69: ZY1000 Zylin JTAG debugger
10.0.0.69: ZY1000 Zylin JTAG debugger
10.0.0.69: ZY1000 Zylin JTAG debugger
10.0.0.69: ZY1000 Zylin JTAG debugger

```

Illustration 4: Network discovery of ZY1000

You can compile the discover application from <http://www.ultsol.com/ZY1000/DiscoverCos.java> with GCJ and build it into a native application, so a Java virtual machine will not be required.

2.4 Access ZY1000 Web-server

Once you have set or discovered the assigned IP address and power cycled the ZY1000 unit, you can connect to the ZY1000 web server with any web browser.

Connect to the ZY1000 web server by using a URL like this in your web browser:
http://192.168.0.100
 (Example IP address)

Your web browser should take you to the ZY1000 web server start page as shown below:

The screenshot shows the ZY1000 JTAG Debugger web interface. The main content area is titled "ZY1000 JTAG Debugger" and includes a table of configuration details, a small image of the hardware unit, and a "Target Status" section with control buttons. A sidebar on the left contains navigation links, and a documentation sidebar on the right provides additional information.

ZY1000 JTAG Debugger	
Current target	board/atmel_sam3s_ek.cfg
Startup	OK
JTAG speed	1000 kHz
Version	1.86-2-gc0ad915
OpenOCD ver.	v0.4.0-875-g6349a47
Build date	May 16 2011
Build time	08:20:50
Current IP	192.168.0.125
GDB Port	3333
HW rev	c

Target Status
 Target power: no

Buttons: **Reset and run** **Halt** **Resume** **Relay on** **Relay off** **Reboot ZY1000**

Hide details
 background polling: off
 TAP: sam3.cpu (enabled)

Documentation sidebar:
 Quick Start Manual
 OpenOCD Manual
 Contact Ultimate Solutions
 ZY1000 Manual

Target status shows that status of the connected target.
 Current target - selected target configuration.
 Startup - whether or not the target script ran to completion. Note that even if the target is disconnected, powered down or unresponsive, the startup script will still run to completion. Startup - OK does not mean that the target is fully operational, simply that the configuration script did not contain syntax errors for instance. See log for details.
 Target power - Detects power on target. If the JTAG cable is not connected, or the target has no power, then no target power will be detected.
 Relay on - Turn on ZY1000 target power relay.
 Relay off - Turn off ZY1000 target power relay.
 Type "help power" in telnet for command to control power relay.
 Reboot ZY1000 - Reboots ZY1000 unit. Type "help reboot" in telnet.

Illustration 5: ZY1000 web server startup/status page

2.5 Check for firmware updates

Visit the ZY1000 page at Ultimate Solutions, Inc. to download the latest firmware for your ZY1000.

Visit the ZY1000 homepage at this address:

<http://www.ultsol.com/index.php/component/content/article/17/217-zv1000-firmware>

3 Using the ZY1000 Web Interface

3.1 Connecting the ZY1000 to a Target

The ZY1000 must be connected to the target through the JTAG interface connector. See the figure below on how to connect the ZY1000 to your target.



Illustration 6: Connecting the ZY1000 to your target.

3.2 Target Configuration

The ZY1000 JTAG Debugger comes with a list of target configuration files. To select a configuration file for your target, go to the **Config Target -> Select Target Config** menu web page.

Select the configuration that matches your target hardware, and press the **select** button, as shown below.

Edit target configuration will bring up the current target configuration on a web page where it can be edited.

The “board” and “target” prefix refers to the PCB with a CPU vs. just the CPU essentially. A “board” script typically uses some “target” script but has e.g. setup of DRAM or similar in the reset init sequence.

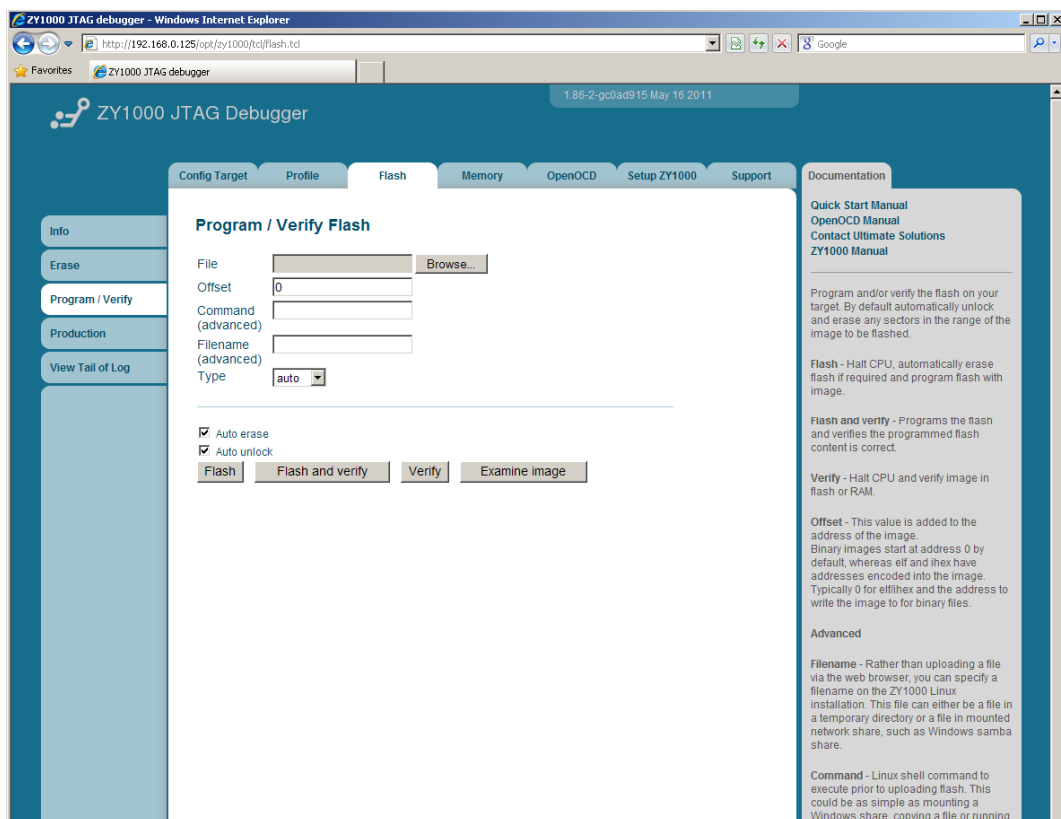


Illustration 7: Selecting target configuration

After loading the new target configuration for the first time, check in the OpenOCD log and verify that everything was loaded correctly.

Go to the **OpenOCD -> View tail of log** page. See example below.

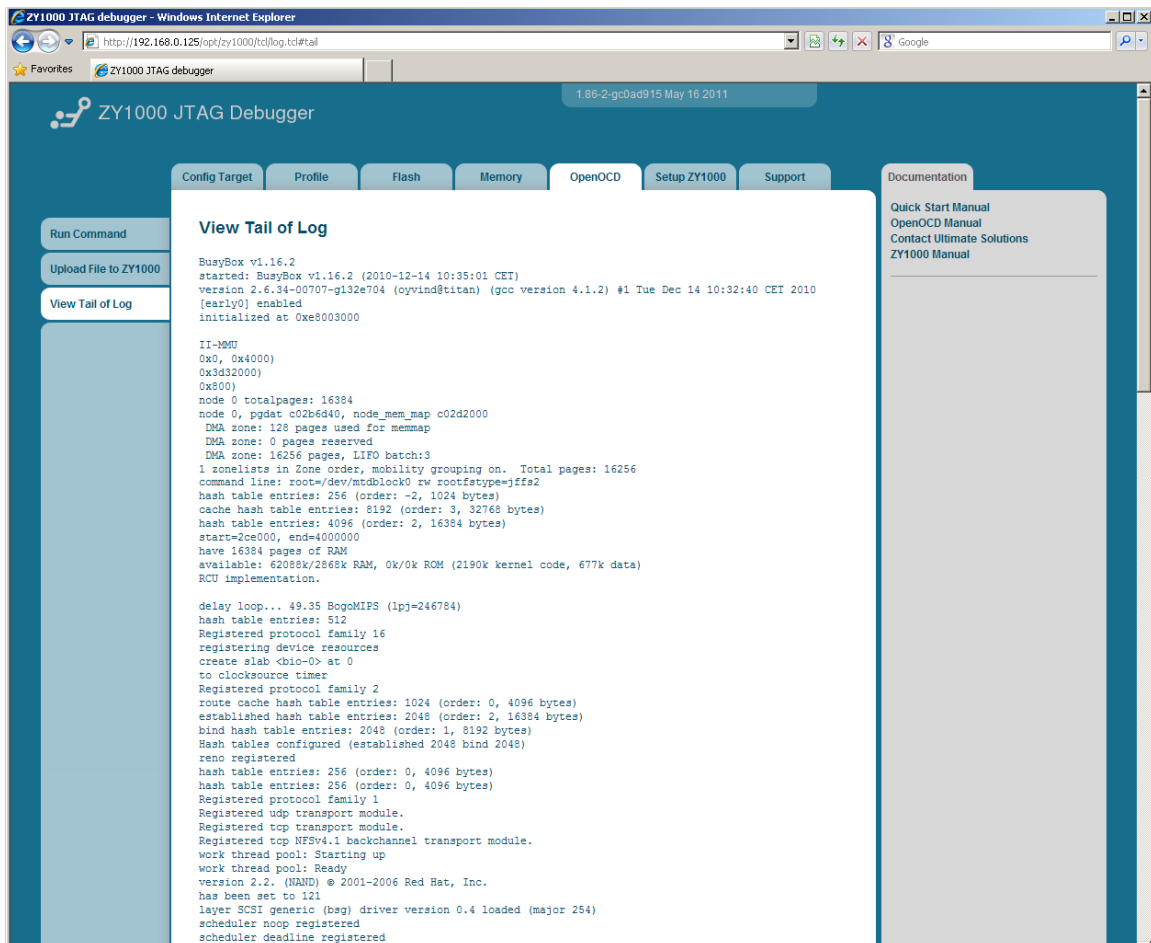


Illustration 8: Verifying that the target configuration is working

3.3 Edit Target Configuration

You can edit existing target configurations, or create your own target configurations. The screen shot below shows the web page for creating/editing your own configuration script.

A modified target configuration can be saved with the same or a new target name. If you save a target definition with “target/myboard.cfg” it will be available in the dropdown list on the Select Target Config page.

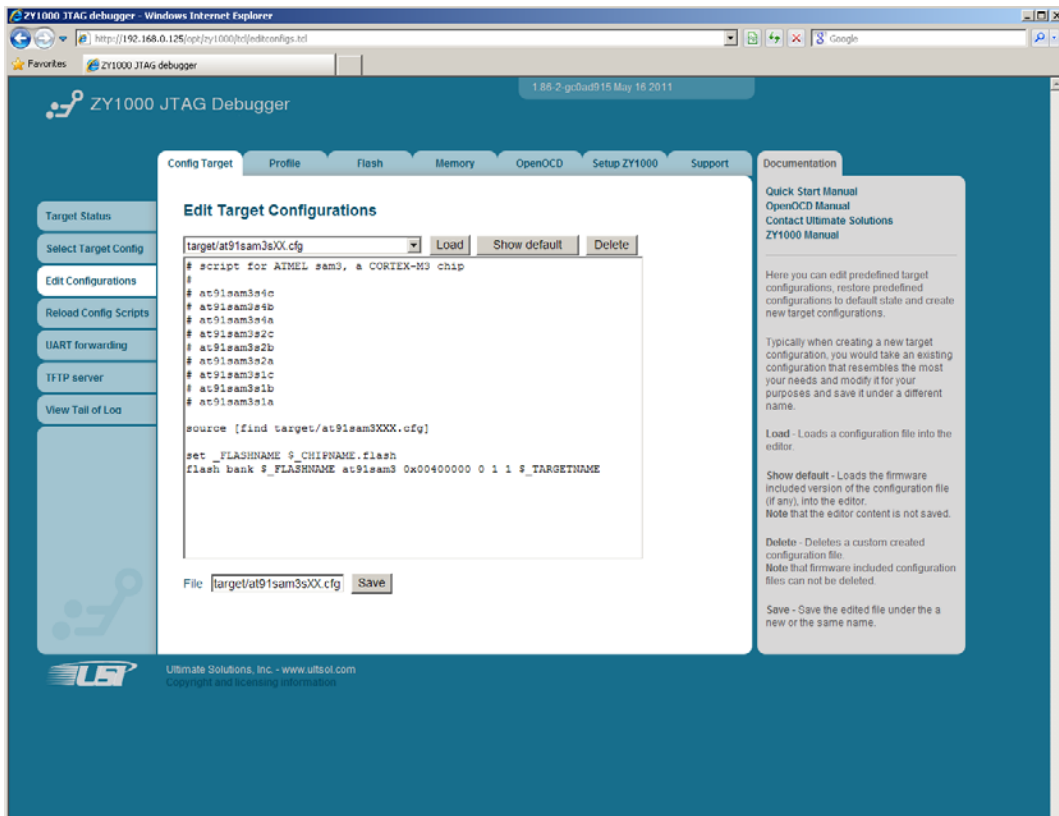


Illustration 9: Edit target configuration script.

3.4 Programming Flash

To program the flash go to the **Flash -> Program/Verify** page. Select the file (image) that you want to program to the flash, and select the flash type. If you have a raw binary, use the Offset field to specify the start address where the image should be placed in the flash, otherwise the flash address will normally be encoded in the .elf/.srec file.

Pressing **Flash**, will erase the flash if required, and then program the flash with the file selected.

The “Auto erase” will erase all sectors in the memory range of the image to be flashed. Note that the flash sectors generally will cover a wider memory range than the actual image.

“Auto unlock” runs an unlock of CFI flash (typically). Note that this only covers the straightforward accidental overwrite flash locking mechanisms and not more exotic mechanisms that are employed for many micro-controllers to avoid tampering and IP theft.

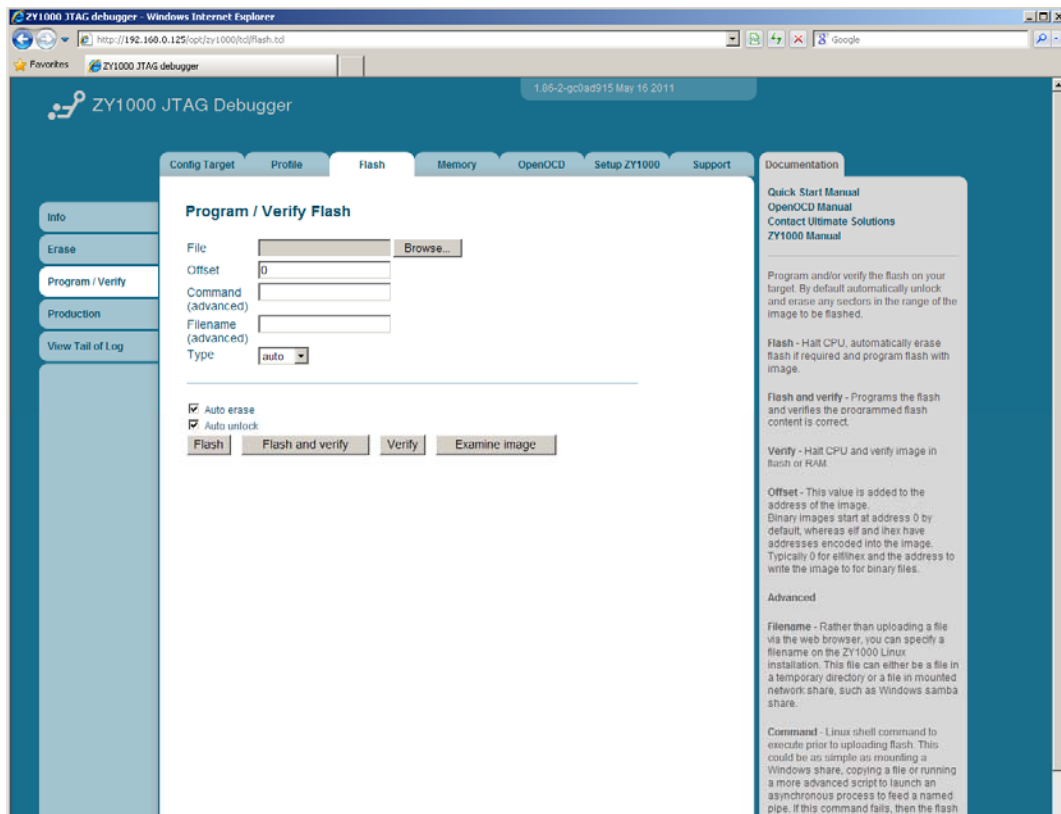


Illustration 10: Programming the flash.

3.5 Browsing and editing memory

You can use the ZY1000 to browse the memory on your target by selecting the **Memory -> Browse/Edit** page.

Select the start address, the length of the memory region, and what type of memory accesses that should be used.

Press the **Refresh** button to show you the content of the specified memory region. See example below.

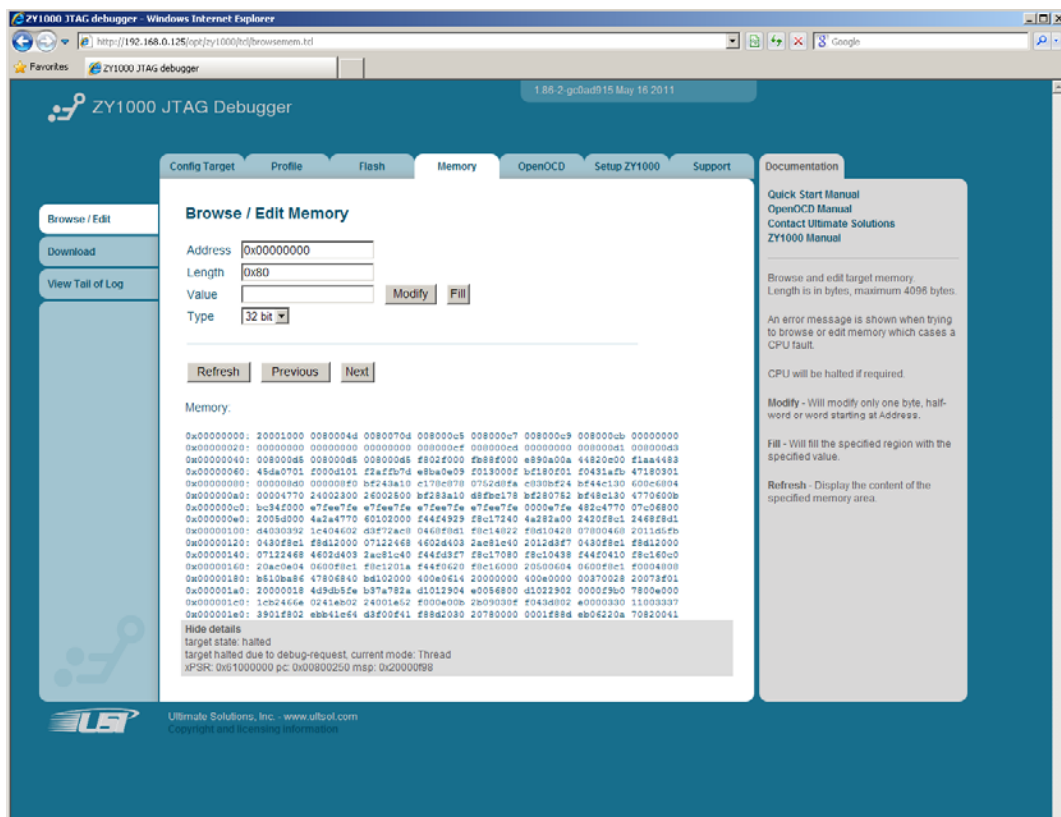


Illustration 11: Browse/edit target memory content

4 Telnet server

The ZY1000 includes a telnet server. This server is active after you have successfully connected to your target.

The telnet server offers an OpenOCD command line.

4.1 Connecting to the Telnet Server

```
Connect to the ZY1000 telnet server:  
telnet 192.168.1.100  
(example)
```

You can use any telnet client you want, but the example below will use the Windows telnet application started from the command line.

The screen-shot below shows how to connect to the Telnet server.
(ZY1000 at 10.0.0.80 in this example)

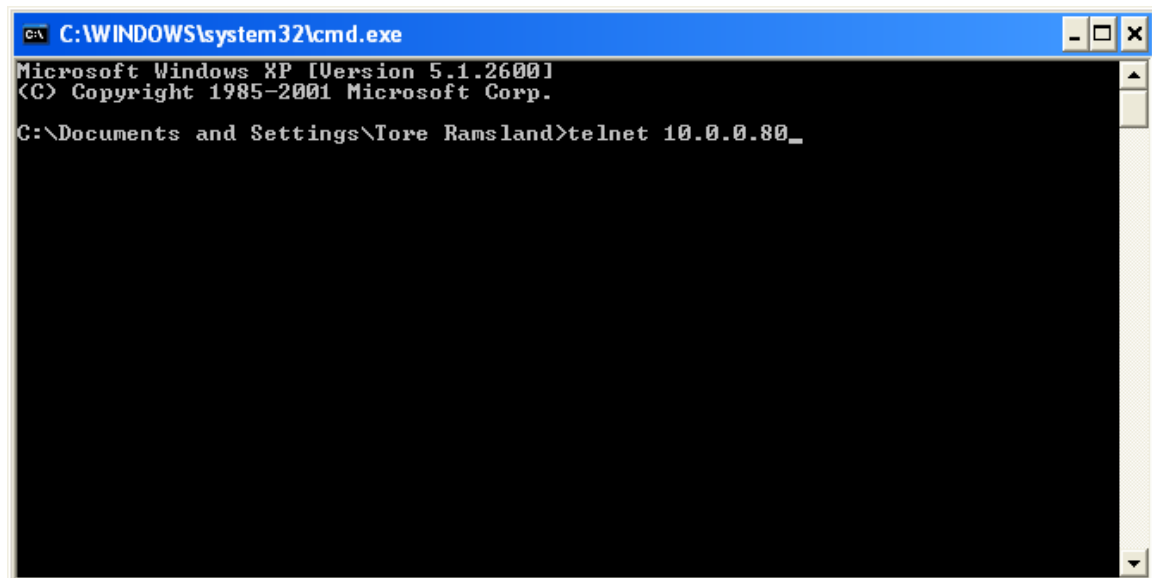
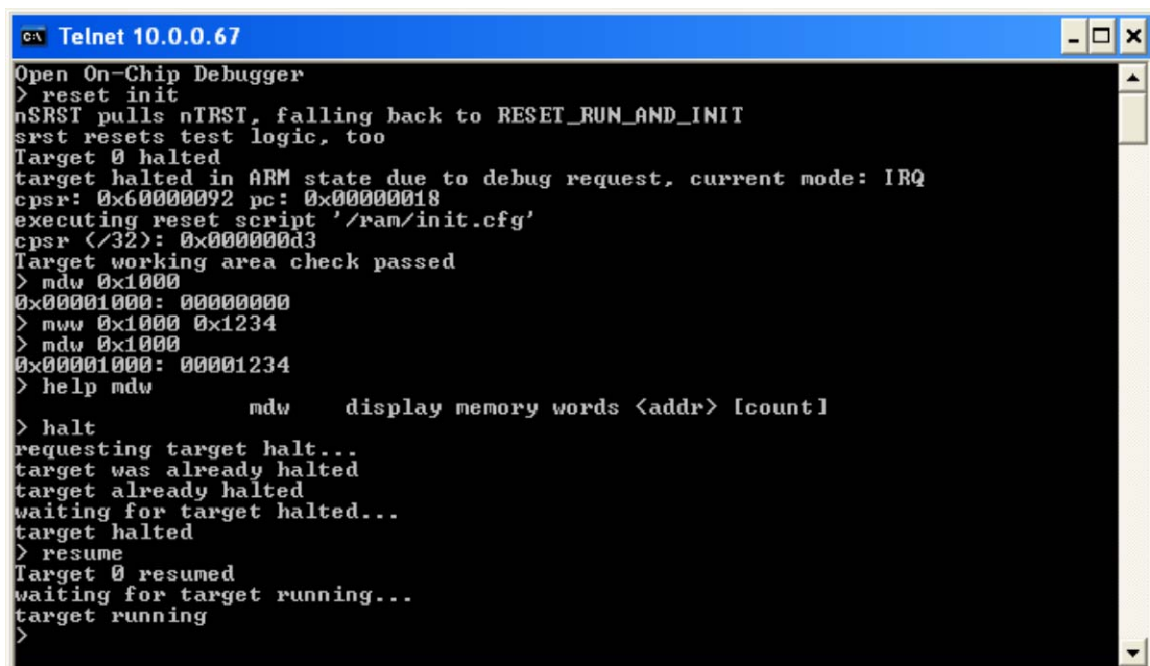


Illustration 12: Connect to ZY1000 Telnet server.

Information about the different OpenOCD commands is outside the scope of this document. Look in the 'OpenOCD Manual' available at: <http://www.ultsol.com/pdfs/openocd.pdf>

The screen-shot gives you an example of Telnet server usage.

A screenshot of a Telnet window titled "Telnet 10.0.0.67". The window contains a text-based interface for an on-chip debugger. The text shows the following sequence of events: a reset is initiated, the target halts in ARM state, registers (CPSR and PC) are displayed, a reset script is executed, memory words are displayed, the target is halted and then resumed, and the target is running again.

```
ca Telnet 10.0.0.67
Open On-Chip Debugger
> reset init
nSRST pulls nTRST, falling back to RESET_RUN_AND_INIT
srst resets test logic, too
Target 0 halted
target halted in ARM state due to debug request, current mode: IRQ
cpsr: 0x60000092 pc: 0x00000018
executing reset script '/ram/init.cfg'
cpsr (</32>): 0x000000d3
Target working area check passed
> mdw 0x1000
0x00001000: 00000000
> mww 0x1000 0x1234
> mdw 0x1000
0x00001000: 00001234
> help mdw
mdw display memory words <addr> [count]
> halt
requesting target halt...
target was already halted
target already halted
waiting for target halted...
target halted
> resume
Target 0 resumed
waiting for target running...
target running
>
```

Illustration 13: Example of Telnet Server usage

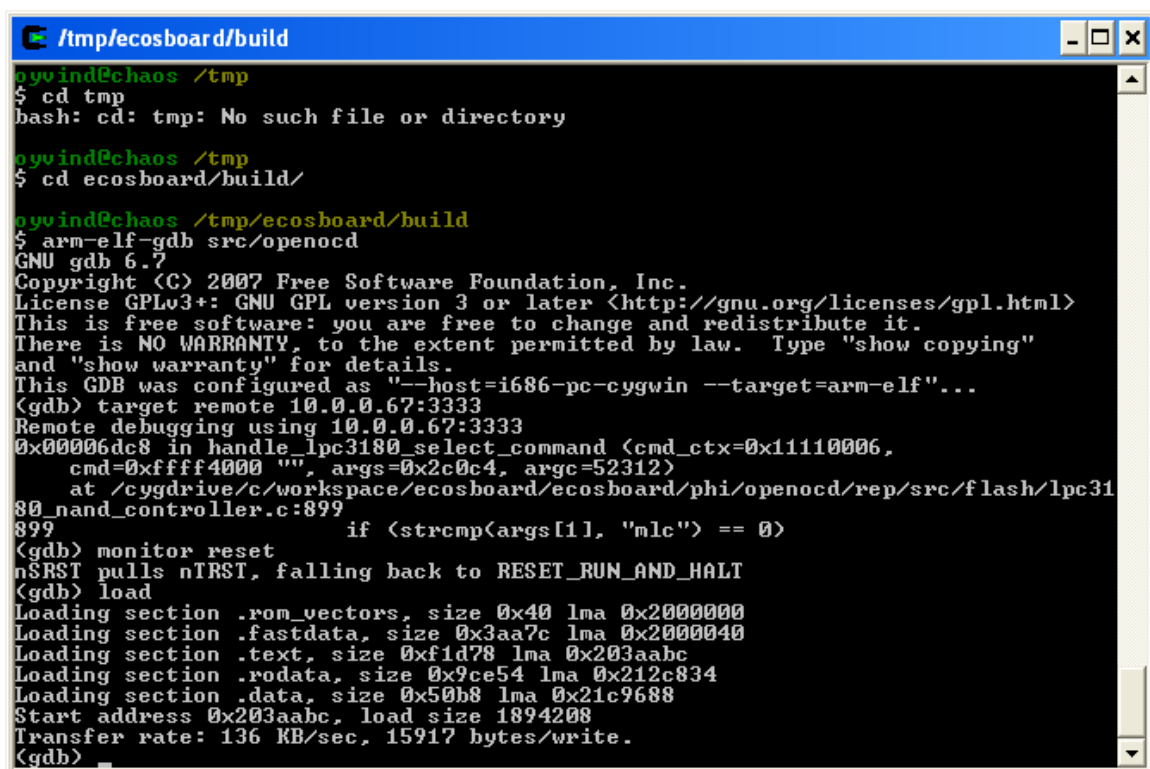
5 GDB Server

The ZY1000 has a built in GDB TCP/IP server and therefore supports all GDB GUI frontends.

5.1 Using GDB interface with text frontend

Below is an example of connecting to ZY1000 and loading an application using the text frontend to GDB.

To send OpenOCD commands to ZY1000 via GDB, add the “monitor” prefix.



```

/tmp/ecosboard/build
ayvind@chaos /tmp
$ cd tmp
bash: cd: tmp: No such file or directory

ayvind@chaos /tmp
$ cd ecosboard/build/

ayvind@chaos /tmp/ecosboard/build
$ arm-elf-gdb src/openocd
GNU gdb 6.7
Copyright (C) 2007 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
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-cygwin --target=arm-elf"...
(gdb) target remote 10.0.0.67:3333
Remote debugging using 10.0.0.67:3333
0x000006dc8 in handle_lpc3180_select_command (cmd_ctx=0x11110006,
  cmd=0xffff4000 "", args=0x2c0c4, argc=52312)
  at /cygdrive/c/workspace/ecosboard/ecosboard/phi/openocd/rep/src/flash/lpc31
80_nand_controller.c:899
899          if (strcmp(args[1], "mlc") == 0)
(gdb) monitor reset
nSRST pulls nTRST, falling back to RESET_RUN_AND_HALT
(gdb) load
Loading section .rom_vectors, size 0x40 lma 0x20000000
Loading section .fastdata, size 0x3aa7c lma 0x20000040
Loading section .text, size 0xf1d78 lma 0x203aabc
Loading section .rodata, size 0x9ce54 lma 0x212c834
Loading section .data, size 0x50b8 lma 0x21c9688
Start address 0x203aabc, load size 1894208
Transfer rate: 136 KB/sec, 15917 bytes/write.
(gdb)

```

Illustration 14: Example of using GDB interface with text GUI

5.2 Using GDB interface with GUI (Eclipse)

There are many graphical frontends to GDB. LinuxScope-JTD from Ultimate Solutions is a debugger for the Eclipse IDE. For more information please visit:

<http://www.ultsol.com/index.php/products/cross-compiler-debuggers/36-ultimate-solutions-linuxscope-jtd-jtag-target-debugger->