BareBoard

BareBoard

HARDWARE MANUAL

WPJ543

Wireless Host CPU Board

RoHS Compliance

TABLE OF CONTENTS

REVISION HISTORY

Revision	Information / Changes
Rev 7A- 03	First release for WPJ543 BareBoard
Rev 7A-04	Added DC Jack (J2) and LED header pins (J3)

**** <u>Note</u>: -**

Except for the physical board size and rearrangement of the LEDs layout differences all software programming interfaces and controls are the same.

The software files for WP543 also apply to WPJ543 board. Thus the software filename that reference WP543 will be shown as in original filename.

USING DEVELOPMENT KIT ADVANTAGES

The Development Kit is especially useful for customers who are developing their firmware. Below are the reasons how we have made it more user-friendly for you.

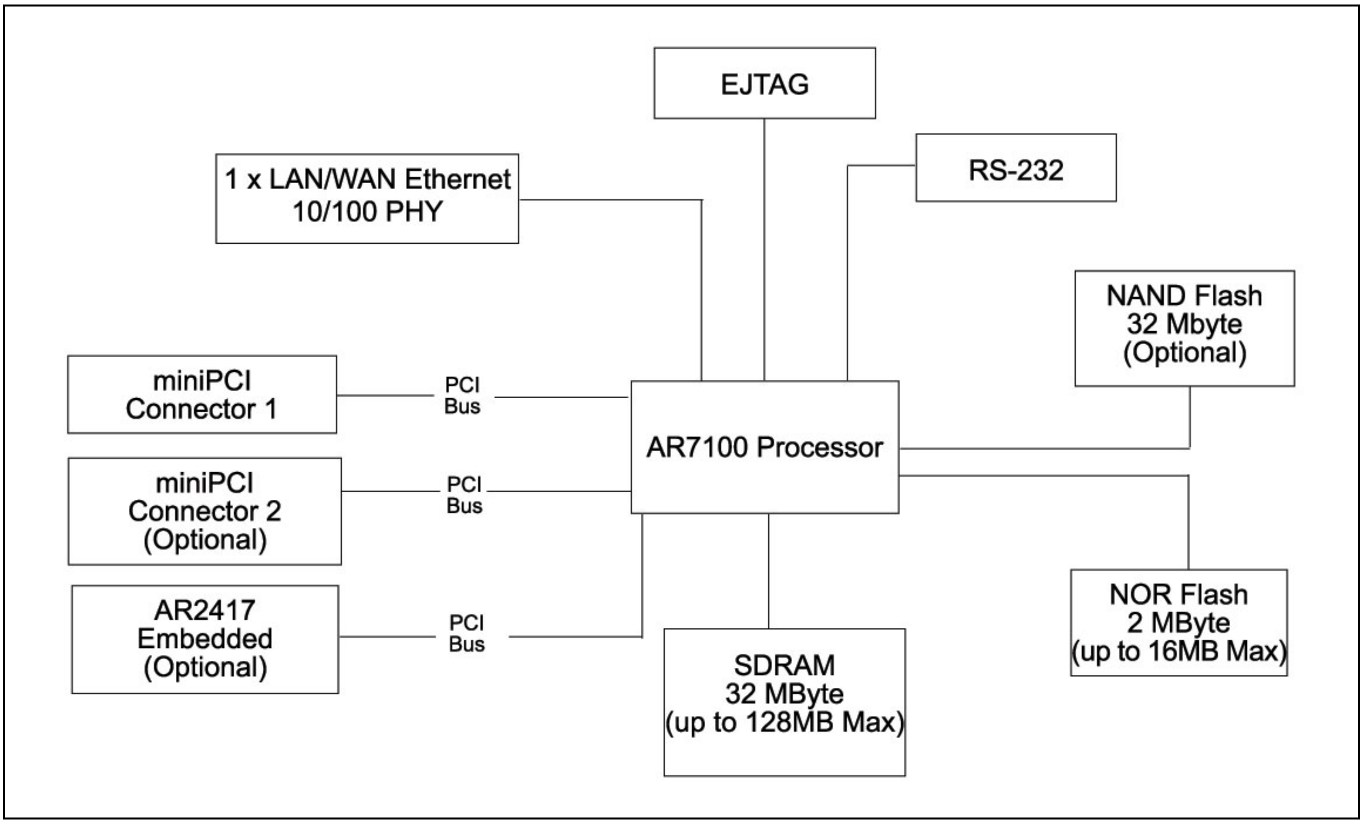
PURPOSE	WHY IS DEVELOPMENT KIT USEFUL?
• Develop Open-WRT firmware on WPJ543 board (using Compex Mylo loader v2.56) • Develop Redboot loader on WPJ543 board	Serial Converter can be used to debug the Open-WRT firmware on Compex Mylo loader. Also, it can be used for developing on redboot loader.
Port Own Firmware Over to WPJ543 board	Serial Converter can be used to debug the Serial Output messages.
Port Own Firmware and Loader Over to WPJ543 board	Serial Converter can be used to debug the Serial Output messages. JTAG Programmer can be used to load in your loader.

Ordering Options - Standard Configurations*

Please contact our sales team at sales@compex.com.sg

* Configurations are subjected to change without notice

BLOCK DIAGRAM


EJTAG
1 x LAN/WAN Ethernet 10/100 PHY
RS-232
NAND Flash 32 Mbyte (Optional)
miniPCI Connector 1
PCI Bus
AR7100 Processor
miniPCI Connector 2 (Optional)
PCI Bus
AR2417 Embedded (Optional)
PCI Bus
NOR Flash 2 MByte (up to 16MB Max)
SDRAM 32 MByte (up to 128MB Max)

KEY FEATURES

GENERAL INFORMATION

PROCESSOR	Atheros AR71XX
MEMORY	32MB or 64MB SDRAM (Up to 128MB max.)
NOR FLASH	Standard 4MB (Up to 16MB max.)
PHYSICAL PORTS	1 x Type III Mini-PCI Slot 1 X 10/100 Base-TX Ethernet Port (with Auto MDI/MDIX)
RADIO SUPPORTED	802.11a/b/g to 802.11n
DEBUG INTERFACE	Serial (TTL) / JTAG (ARM-standard 20 pin) Optional JTAG Programmer** available Optional Serial Converter*** available
OPERATING TEMPERATURE	-22°C to 70°C
LED INDICATORS	6 LEDs Power, Ethernet, Signal Strength (programmable)
OTHER FEATURES	4 x LED are programmable Push-Button Reset Surge Arrestors (Optional)
DIMENSIONS	105 x 95 x 18 (mm)
ENCLOSURE	Compex WPMN enclosure or any other third party enclosure with proper mounting plates.

INFORMATION ON POWER

OWER OVER ETHERNET	High-power passive PoE input voltage Range: 12V-24V DC)
TYPICAL OPERATING POWER	4 W (board only)
DC SUPPLY	12V ~ 24V DC Supply
MINIPCI SLOTS	Max. 12 W total power Support Type III-B and Type III-A radio cards

* Depend on Order Configuration
** JTAG Programmer available to reprogram the flash in case of loader corruption.
*** Serial Converter available to change the TTL signals on board to RS232 signals for debugging

GPIO Bit Mapping

GPIO Bit	Description
0	N/A
1	N/A
2	N/A
3	DE1 (LED)
4	N/A
5	DS3 (LED)
6	DS4 (LED)
7	DS5 (LED)
8	SW6 (Reset button)
9	UART_SIN
10	UART_SOUT
11	N/A
12	JTAG(TCK)
13	JTAG(TDO)
14	JTAG(TDI)
15	JTAG(TMS)
16	JTAG(TRST_L)

Interface Connectors

The board interface connector pin assignments and signal descriptions are included in the following sections. The connectors are listed in the section below and the connector locations are shown in the following diagrams.

Connector	Function
J2	DC Jack
P2	Ethernet Port
J11	JTAG
J30	MiniPCI Slot
J31	Serial (TTL) Port

Serial Port Header

The Serial Port (J31) Header signaling is shown in the following table.

Pin	Signal (TTL)
1	VCC – 3.3V
2	UART 0 Transmit Data
3	UART 0 Receive Data
4	GND

Note:

Our Serial port Implementation requires an external high-impedance serial port not usually available with the serial ports of the notebooks/computers. You will need a Serial Converter available in the market. For our customers' convenience, it is bundled together with the board Development Kit.

Serial Console Settings

The serial console settings used together with the serial port is given below.
This serial port uses TTL signals. Requires a serial converter to convert TTL to RS232 signal output to be able to connect to a standard PC COM port.
MAX-211 IC (or other IC in the market that convert TTL signals to RS232 signals) can be use.

Baud Rate	115200
Data	8 Bit
Parity	None
Stop	1 Bit
Flow Control	None

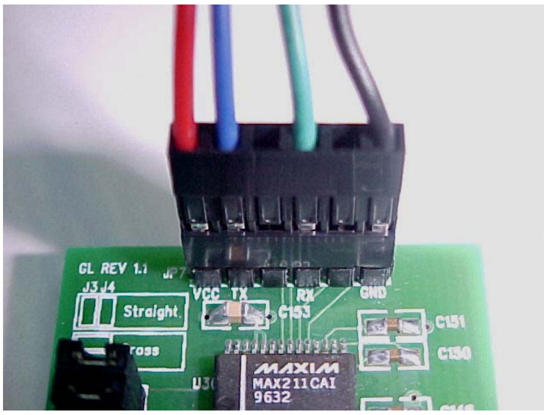
Precaution when using Serial Converter

Please attach the serial converter first on the board serial header, before attaching the power supply. This is to ensure that there is no surge of power to the serial converter, and prevent any damage the chipset on the serial converter.

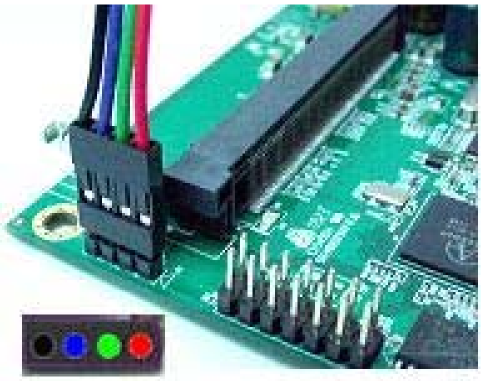
Serial Converter Pin Layouts

Cables on the serial converters are provided. You can use the 6 Pin (Fixed) to 4 Pin (Fixed) provided. The pin layouts of the serial converters for use with the board are as follows:

Pin Assignment (Serial Converters)	Signal (Serial Converters)	Connected to Pin on WPJ543	Signal (WPJ543)
Pin 1	VCC(3.3V) –Red	Pin 1	VCC (3.3V) -Red
Pin 2	TX – Blue	Pin 2	RX
Pin 4	RX - Green	Pin 3	TX
Pin 6	GND – Black	Pin 4	GND



Arrangement of Cables on Serial Converter to the board



Arrangement of Cables on the board itself

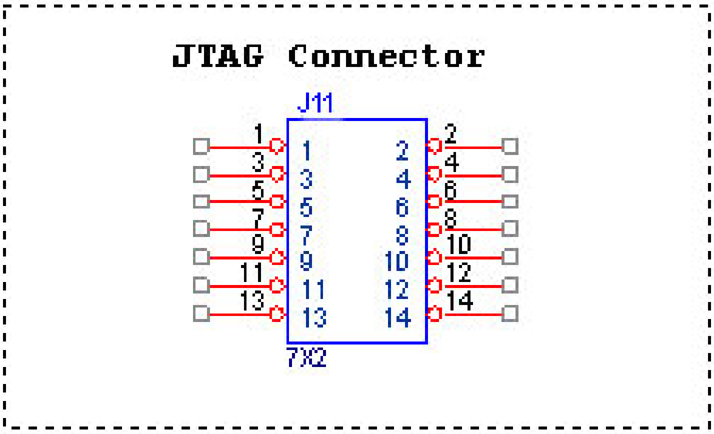
JTAG Port Header

The primary purpose of the board JTAG Port Header is to facilitate program download into Flash memory.

Pin	Signal	Pin	Signal
1	TRST_N	2	GND
3	TDI	4	GND
5	TDO	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	RESET	12	NC
13	DINT	14	3V3

Note:

Normally, it has a JTAG Programmer compatible with the board. It is bundled with the board Development Kit. This JTAG programmer is able to download file onto the Flash, and thus recover a corrupted loader.



Ethernet Connectors

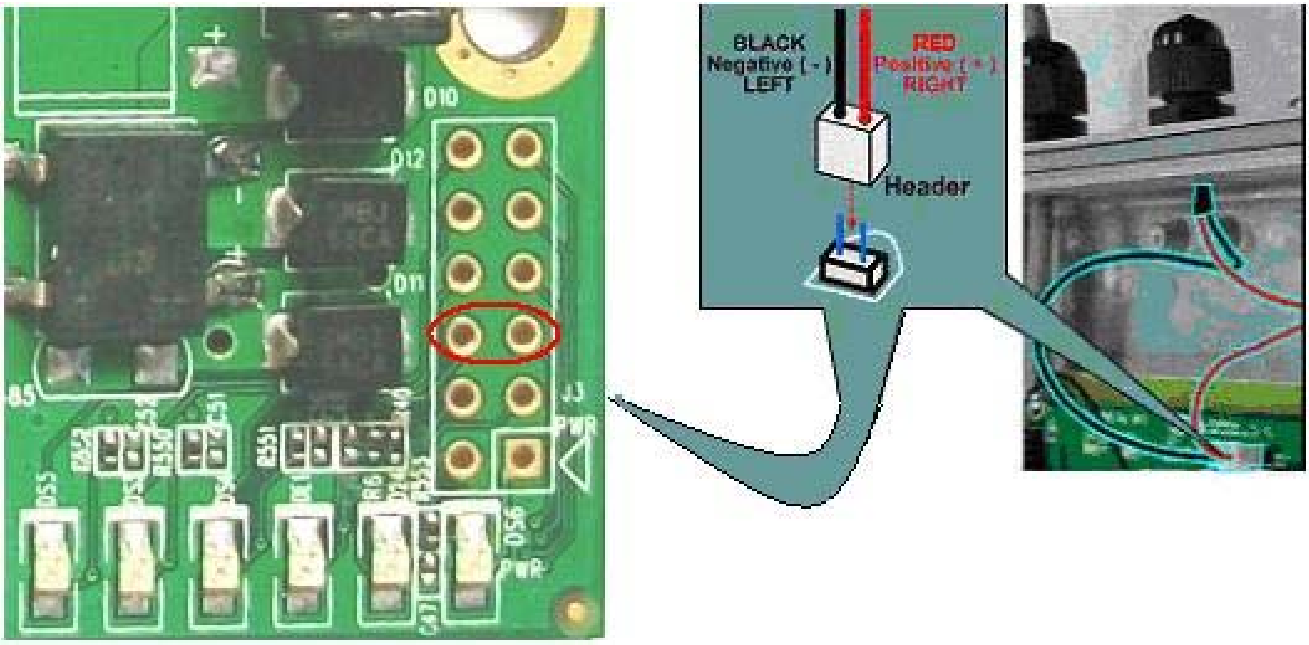
The board contains 1 X 10/100 Base-TX Ethernet Channels. The Ethernet Channels are available through standard 8-pin RJ45 connectors.

Ethernet Connectors(P2) signaling is shown below.

Pin	Signal
1	TX+
2	TX-
3	RX+
4	PoE+V
5	PoE+V
6	RX-
7	GND
8	GND

How to Extend LED on board to enclosure

If you want to extend the LED on board to the enclosure, you need prepare for each LED 2 jumper wire plug. Pin 1 is marked with an arrow.
Pin 1,3,5,7,9,11are connected to 3.3V DC source to LED (+ve)lead.
Pin 2,4,6,8,10,12 are to connect to the LED (–ve) lead.



.

Vcc	Pin		LED label
3.3v	**1**	**2**	**DS5** (POWER)
3.3v	**3**	**4**	**D24** (ETHERNET)
3.3v	**5**	**6**	**DE1** (programmable)
3.3v	**7**	**8**	**DS4** (programmable)
3.3v	**9**	**10**	**DS3** (programmable)
3.3v	**11**	**12**	**DS5** (programmable)

JTAG Process

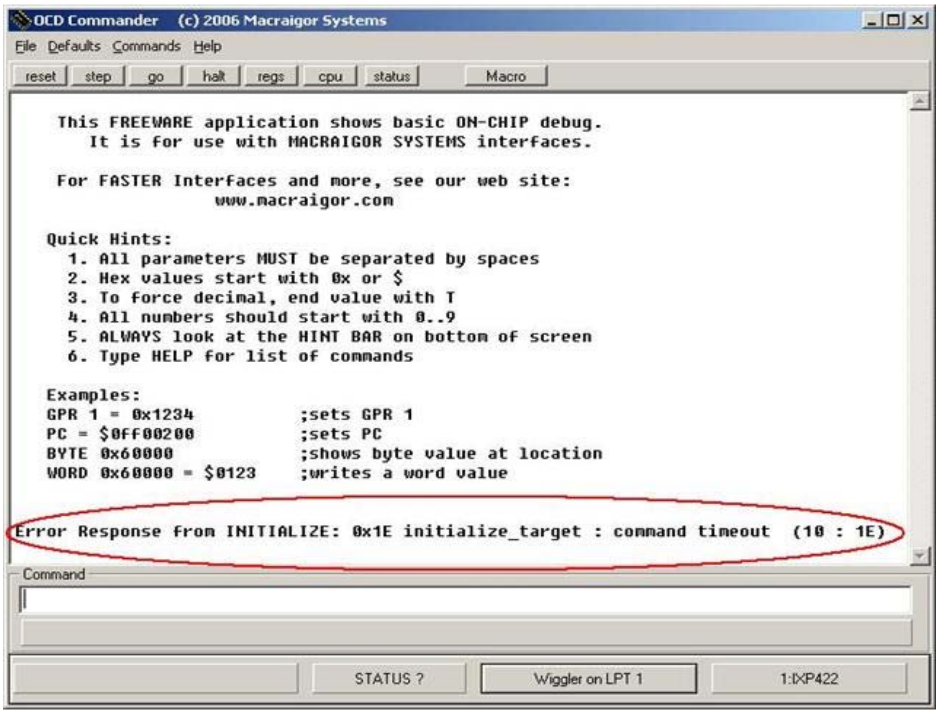
Minimum Requirement

1. OCD Commander ver2.5.4 or higher.
2. upbios.tst file (same for all Compex device)
3. zMylo.bin file(different device have different zMylo.bin)
4. JTAG cable

Steps

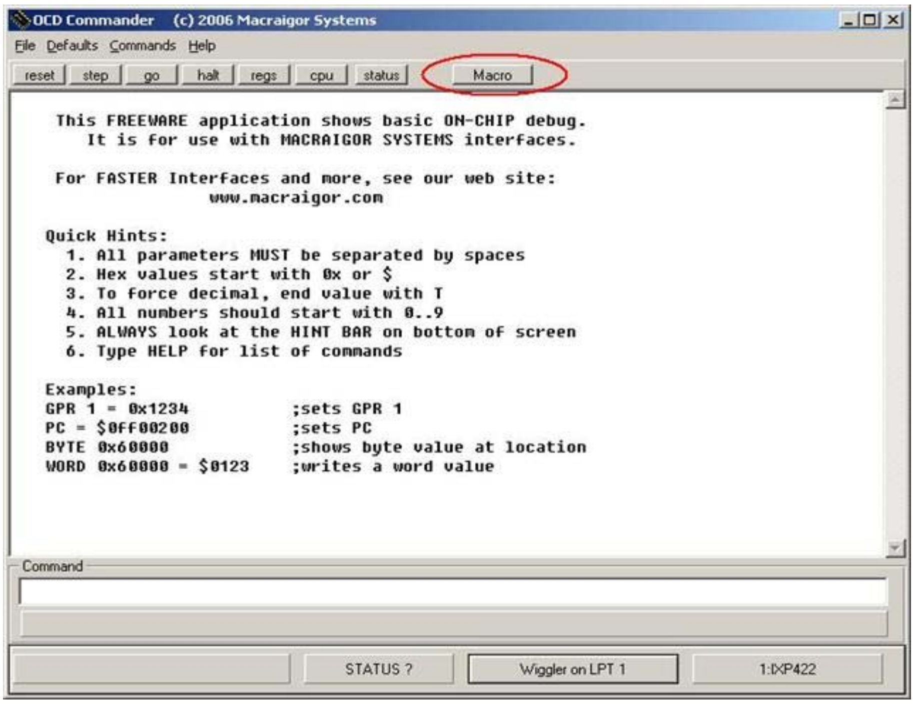
1. Install the OCD Commander to your PC
2. Plug the JTAG cable to the JTAG port of the device
3. Run OCD Commander Program
   Set "Target Processor" for the particular device
   eg. WP18 - INTEL, IXP422

   Click "OK"
4. If there is this error message “Error Response from INITIALIZE....”, please check the JTAG cable connection. Close the OCD Commander Program and go back to Step 3.



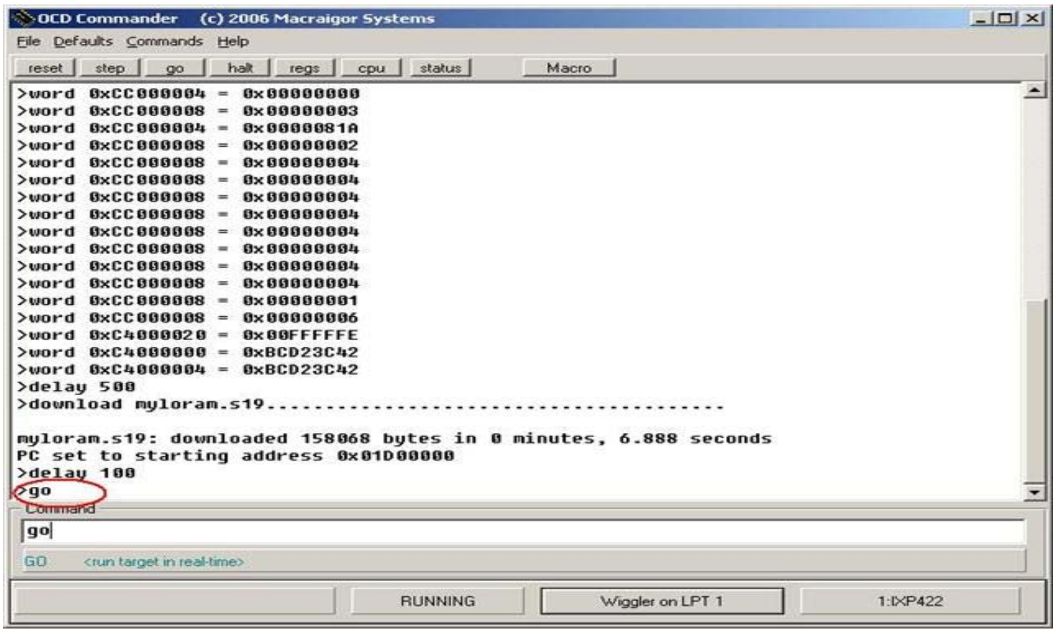
```
   This FREEWARE application shows basic ON-CHIP debug.
      It is for use with MACRAIGOR SYSTEMS interfaces.

   For FASTER Interfaces and more, see our web site:
                  www.macraigor.com

  Quick Hints:
    1. All parameters MUST be separated by spaces
    2. Hex values start with 0x or $
    3. To force decimal, end value with T
    4. All numbers should start with 0..9
    5. ALWAYS look at the HINT BAR on bottom of screen
    6. Type HELP for list of commands

  Examples:
  GPR 1 = 0x1234          ;sets GPR 1
  PC = $0ff00200          ;sets PC
  BYTE 0x60000            ;shows byte value at location
  WORD 0x60000 = $0123    ;writes a word value

Error Response from INITIALIZE: 0x1E initialize_target : command timeout  (10 : 1E)
```

5. Click on the “Macro” and choose a specific .mac file.


OCD Commander (c) 2006 Macraigor Systems
File Defaults Commands Help
reset step go halt regs cpu status Macro
Command
STATUS ?
Wiggler on LPT 1
1:IXP422

```
  This FREEWARE application shows basic ON-CHIP debug.
     It is for use with MACRAIGOR SYSTEMS interfaces.

  For FASTER Interfaces and more, see our web site:
                 www.macraigor.com

 Quick Hints:
   1. All parameters MUST be separated by spaces
   2. Hex values start with 0x or $
   3. To force decimal, end value with T
   4. All numbers should start with 0..9
   5. ALWAYS look at the HINT BAR on bottom of screen
   6. Type HELP for list of commands

 Examples:
 GPR 1 = 0x1234          ;sets GPR 1
 PC = $0ff00200          ;sets PC
 BYTE 0x60000            ;shows byte value at location
 WORD 0x60000 = $0123    ;writes a word value
```

6. Let it run until u see “go”


OCD Commander (c) 2006 Macraigor Systems
File Defaults Commands Help
reset step go halt regs cpu status Macro
Command
RUNNING
Wiggler on LPT 1
1:IXP422

```
>word 0xCC000004 = 0x00000000
>word 0xCC000008 = 0x00000003
>word 0xCC000004 = 0x0000081A
>word 0xCC000008 = 0x00000002
>word 0xCC000008 = 0x00000004
>word 0xCC000008 = 0x00000004
>word 0xCC000008 = 0x00000004
>word 0xCC000008 = 0x00000004
>word 0xCC000008 = 0x00000004
>word 0xCC000008 = 0x00000004
>word 0xCC000008 = 0x00000004
>word 0xCC000008 = 0x00000004
>word 0xCC000008 = 0x00000001
>word 0xCC000008 = 0x00000006
>word 0xC4000020 = 0x00FFFFFE
>word 0xC4000000 = 0xBCD23C42
>word 0xC4000004 = 0xBCD23C42
>delay 500
>download myloram.s19......................................

myloram.s19: downloaded 158068 bytes in 0 minutes, 6.888 seconds
PC set to starting address 0x01D00000
>delay 100
>go
```

```
go
GO      <run target in real-time>
```

7. Open command prompt

8. tftp upbios.tst

9. tftp zMylo.bin (please observe the DIAG LED is off)

10. If either step 9 or step 10 fail, please start from step 3 again.

11. Power off the device and unplug JTAG cable

12. Power on the device and tftp the firmware into the device.

13. Reboot when done.

Build and Install Process (For OpenWRT firmware on Compex Mylo Loader)

Minimun Requirement

1. Compex loader version 2.54 or above.
2. OpenWRT will only be supported on WPJ543 board with 4MB NOR flash and above.
3. Please ensure that the Ethernet connection is able to ping address = 192.168.168.1
4. If there is a USB NAND flash available on board, you would need to change the loader configuration. Please refer to "what to do if there is an onboard NAND Flash".

Compiling OpenWRT suitable for use on WPJ543 board

1. Getting source codes

   ```
   >svn co –r 12448 https://snv.openwrt.org/openwrt/trunk
   ```

2. Apply patches from the files mod-wp543.tgz

   ```
   >tar zxvf mod-wp543.tgz
   >cp –a mod-wp543/* trunk/
   ```

3. Compile

   ```
   >cd trunk
   >cp wp543.config .config
   >make
   ```

OpenWRT Firmware will be in bin/openwrt-ar71xx-wp543.bin

Uploading the OpenWRT firmware to WPJ543 board running MyLoader v2.54.0717

a. Via Compex Firmware

   - Put the AP in Firmware Upgrade mode and upload the file.
   - This file is for WPJ543 with 4MB or 8MB NOR flash.

b. Via TFTP

   - Go to the firmware upgrade mode (By pressing and hold the Reset button and plug-in the power adapter).
   - Upload the OpenWRT image to the device (tftp –i 192.168.168.1 put openwrt-ar71xx-wp543.bin)

First run of OpenWRT

- For customers with Serial console
  During first run after flashing the firmware, do not power off the AP until the following messages appear:
  jffs2_scan_eraseblock(): End of filesystem marker found at 0x0
  jffs2_build_filesystem(): unlocking the mtd device... done.
  jffs2_build_filesystem(): erasing all blocks after the end marker... done.

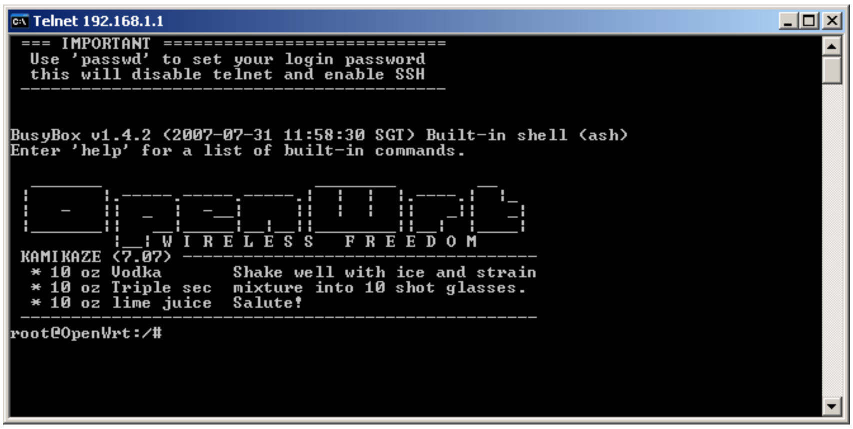
- For customers without Serial console
  If you do not have console, just wait for 5 minutes.

Completion

Reboot the AP.

Start using OpenWRT by telnet to default IP and you will see the picture shows below.

> Telnet 192.168.1.1
> The picture shows below will appear if you are successful.



Telnet 192.168.1.1

```
 === IMPORTANT ============================
  Use 'passwd' to set your login password
  this will disable telnet and enable SSH
 ------------------------------------------


BusyBox v1.4.2 (2007-07-31 11:58:30 SGT) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 KAMIKAZE (7.07) -----------------------------------
  * 10 oz Vodka       Shake well with ice and strain
  * 10 oz Triple sec  mixture into 10 shot glasses.
  * 10 oz lime juice  Salute!
 ---------------------------------------------------
root@OpenWrt:/#
```

What to do if there is a onboard USB Flash

If there is onboard USB flash, you would need to use the serial converter, enter the loader mode, by pressing,

please configure the Loader to use NOR-Flash, i.e.
"9 – USB Flash" → "3 – Boot Device" → "2 – Onboard NOR-Flash"

Default Configurations
LAN (bridge eth0+ath0):
IP Address: 192.168.1.1
Wireless (ath0):
Driver: madwifi
Mode: ap
ESSID: OpenWRT
IP Address: 192.168.2.1

Please refer to http://madwifi.org/ for more information.

Use of Compex Patches

1. MAC Address from loader
2. Can detect that it is a Compex board
3. Enable USB

Build and Install Process (For Redboot Loader)

Note: Wireless drivers NOT in SDK

Note that in Compex WP543 SDK, redboot loader binaries are provided as part of the SDK, which is only given to customers who have signed NDA/TLA with Compex

Standard Platform

Requirements

The main requirement for building the standard distribution is having a Linux based development platform with at least 1 GByte of free disk space, and a working GCC compiler tool chain. These procedures have been tested on a Fedora Core 8 machine, and on older machines with Fedora Core 4.

The distribution contains all tools required to build the bootloader, kernel, and jffs2 image to be loaded onto the flash.

Further, a tftp server is required, preferably on the development platform. This server is used by the reference platform to download all file images required.

Unpacking and Build Process

For board with NOR Flash only(without NAND flash), **WP543-sdk-NOR-rel.tar** is provided., where "***rel***" is the combination of board and release, such as "**WP543-sdk-NOR-V1.0.tar**". The SDK file contains all of the files associated with the BSP, Kernel distribution, applications provided, build tool source, and the build system. This file can be used if the user is building for another system, and is not interested in the BSP, build tools, etc.

Select a directory to open the build into, and use the following procedure. This procedure assumes that "athbuild" is the directory to unpack into, and that the SDK files are located in the directory immediately above. Modify the procedure according to your configuration:

```
#cd athbuild
#tar –xzvf ../ WP543-sdk-NOR-V1.0.tar
(large number of files unpack)
#tar –xzvf ../ WP543-sdk-NOR-V1.0.tar
(smaller number of files unpack).
```

When the source files are unpacked, the build process can begin. The build directory contains the make file for building all components required. The source distribution will contain the required files for the build of your choice.

The build system will create two new directories: images and rootfs.build. The images directory will receive the final output of the build process that is used to update the platform board. The redboot images, the Linux kernel image in compressed format, and the jffs2 filesystem are all copied into this directory, under the subdirectory for the specific reference platform. The rootfs.build directory is used as an install target to create the jffs2 filesystem. All components that are build are installed into this directory, and the final step is to run mkfs.jffs2 on this directory to create the filesystem image.

To do a full system build, use the following procedure:

```
#cd build
# make BOARD_TYPE=board
```

where ***board*** is the specific board type of your reference platform, **WPJ543**. This will generate a full build of all components, and the cross tools required to build for the reference platform. Note that on subsequent builds the tools will NOT be rebuilt (this is a long process, the first build can take well over an hour).

After a full build is performed, components can be rebuild as required. To build a specific component, you will use the command:

```
# make target BOARD_TYPE=board
```

where target is the specific target of choice. If you are unsure if the rootfs.build directory is properly populated, run the full build to erase and recreate the image. The following table outlines the main targets available.

Target	Builds
Fusion_build	Rebuilds all driver files, and regenerates the module .ko files. Copies new files into the rootfs.build directory. This will also rebuild the jffs2 filesystem.
busybox_build	Rebuilds the busybox component, and installs into the rootfs.build directory.
hostapd_fus	Rebuilds the hostapd application provided to support WPA encryption. Installs into rootfs.build directory.
toolchain_build	Rebuilds the gcc cross build tools provided. These are left in the build area, and are **not** put into the rootfs.build system
redboot_fusion	Rebuilds the redboot bootloader, and copies the image into the images directory.
enet_build	Rebuilds the Ethernet driver, and copies the ar7100.ko module into the rootfs.build directory.
Kernel_build	Rebuilds the Linux kernel. The result is copied to the images directory
fus_supplicant	Rebuilds the WPA supplicant application provided, and installs into the rootfs.build.

Installation

To perform a software update on the system, the following are required

- WPJ543 board
- A serial converter
- A server system with a tftp server
- A terminal system with terminal emulation software, such as hyperterm or Minicom
- An Ethernet cable between the server system and the WPJ543

Connect the Ethernet ports on the server system and the Ethernet port. Connect the serial cable between the Terminal system and the AP, using the serial converter. (Refer to “Serial Port Header” section, *pg 8*)

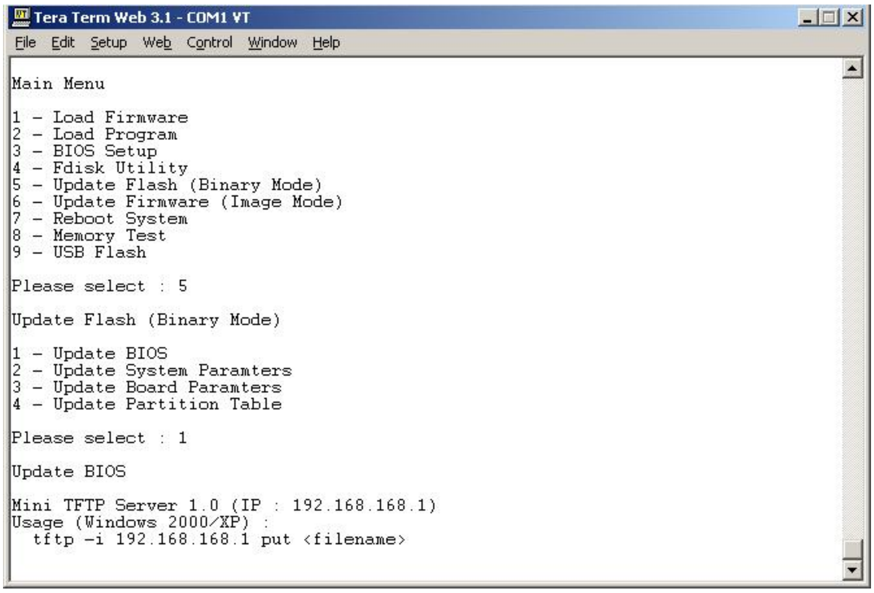
A) Boot Monitor Update (with Compex Bootloader)

This procedure is provided to update the Compex loader to redboot loader. The source code for redboot is included. Redboot.rom is also included in the SDK.

WARNING Incorrect implementation of this procedure can cause board failure due to erasing the boot monitor from the Flash. If this occurs, the board will have to be reloaded via download from an EJTAG emulator.

(Please refer to JTAG Port Header Section)

1. **Boot with existing Compex Bootloader and halt it from booting into firmware.**



Tera Term Web 3.1 - COM1 VT
File Edit Setup Web Control Window Help

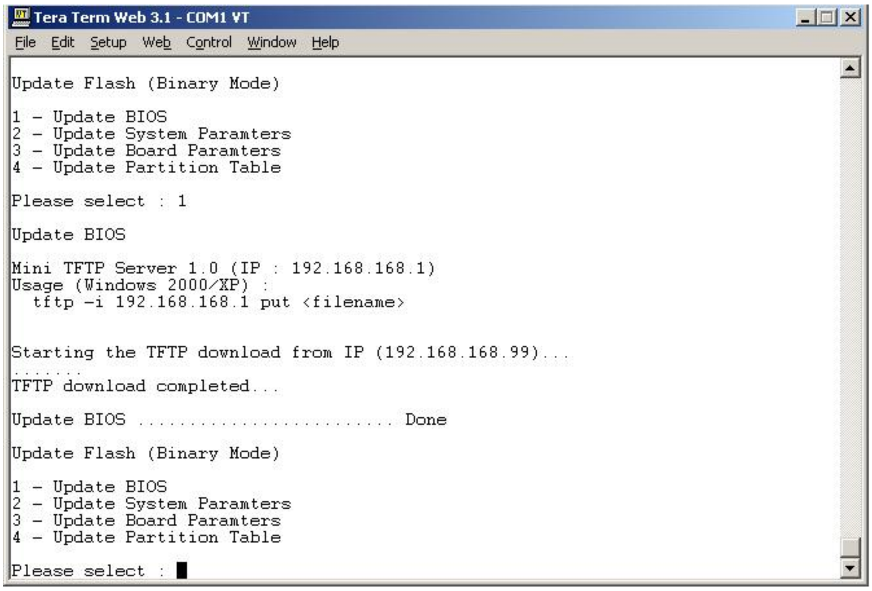
```
Main Menu

1 - Load Firmware
2 - Load Program
3 - BIOS Setup
4 - Fdisk Utility
5 - Update Flash (Binary Mode)
6 - Update Firmware (Image Mode)
7 - Reboot System
8 - Memory Test
9 - USB Flash

Please select : 5

Update Flash (Binary Mode)

1 - Update BIOS
2 - Update System Paramters
3 - Update Board Paramters
4 - Update Partition Table

Please select : 1

Update BIOS

Mini TFTP Server 1.0 (IP : 192.168.168.1)
Usage (Windows 2000/XP) :
  tftp -i 192.168.168.1 put <filename>
```

2. **Load new redboot.rom version into memory using Command Prompt.**

   tftp –i 192.168.168.1 put redboot.rom

   (redboot.rom is included in /bin)

3. **Restart the board after the loading is done.**



Tera Term Web 3.1 - COM1 VT
File Edit Setup Web Control Window Help

```
Update Flash (Binary Mode)

1 - Update BIOS
2 - Update System Paramters
3 - Update Board Paramters
4 - Update Partition Table

Please select : 1

Update BIOS

Mini TFTP Server 1.0 (IP : 192.168.168.1)
Usage (Windows 2000/XP) :
  tftp -i 192.168.168.1 put <filename>


Starting the TFTP download from IP (192.168.168.99)...
.......
TFTP download completed...

Update BIOS ......................... Done

Update Flash (Binary Mode)

1 - Update BIOS
2 - Update System Paramters
3 - Update Board Paramters
4 - Update Partition Table

Please select : █
```

4. **Initialize the configuration on the board using the following value (user entries underlined)**

```
Redboot> fis init
About to initialize [format] FLASH image system – continue (y/n)? y
*** Initialize FLASH Image System
... Erase from 0xbf7e0000-0xbf7f0000
... Program from 0x80ff0000-0x81000000 at 0xbf7e0000:.
```

B) i) Boot Monitor Update (with Existing Redboot)

This procedure is provided to update the boot monitor with a newer version that is generated. Note that the source code for the boot monitor is included in your distribution.

> **WARNING Incorrect implementation of this procedure can cause board failure due to erasing the boot monitor from the Flash. If this occurs, the board will have to be reloaded via download from an EJTAG emulator.**
>
> (Please refer to JTAG Port Header Section)

1. **Boot with existing Redboot and halt it from booting into Linux.**

2. **Load new redboot.rom version into memory.**

   ```
   RedBoot> load -r -v -b 0x80500000 redboot.rom -h <tftp server ip addr>
   ```

3. **Write redboot to flash**

   ```
   Redboot> fis write -b 0x80500000 -f 0xbf000000 -l 0x40000
   ```

4. **Reboot the board with new Redboot and break into the monitor with <ctrl-c> before booting Linux**

   ```
   Redboot> reset
   ```

5. **Now reformat the flash with new redboot running.**

   ```
   Redboot> fis init -f
   ```

6. **Initialize the configuration on the board using the following values (user entries underlined)**

   ```
   RedBoot> fconfig –i
   Initialize non-volatile configuration - continue (y/n)? y
   Run script at boot: true
   Boot script:

   Enter script, terminate with empty line
   >> fis load -d vmlinux
   >> exec
   >>
   Boot script timeout (1000ms resolution): 3
   Use BOOTP for network configuration: false
    Gateway IP address:
   Local IP address: <Your IP address>
   Local IP address mask: <Your required netmask>
   ```

```
Default server IP address: <TFTP server IP address>
Console baud rate: 115200
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Update RedBoot non-volatile configuration - continue (y/n)? y
... Erase from 0xbf7e0000-0xbf7f0000: .
... Program from 0x80ff0000-0x81000000 at 0xbf7e0000: .
```

Since this procedure erases the flash device, you will also have to perform the software update process in the following section. This is the normal process used to update the driver or kernel when changes are made.

ii) Kernel/Driver Update

The Kernel and Driver image update is performed using the Red Boot boot monitor, commands through the serial console, and tftp file transfers. This procedure must be performed after doing a redboot update as described in the preceding section.

1. **Power up the board, and hit <ctrl-c> to break into monitor**
2. **Load with new linux image.**

   ```
   Redboot> load -r -v -b 0x80500000 vmlinux.bin.gz -h <tftp server ip>
   Redboot> fis create -b 0x80500000 -e 0x80256000 -r 0x80060000 -l
   0x100000 vmlinux
   ```

3. **Load new jffs2 filesystem**

   ```
   Redboot> load -r -v -b 0x80500000 pb42fus-jffs2 -h <tftp server ip>
   Redboot> fis create -b 0x80500000 -e 0x0 -l 0x600000 filesystem
   ```

4. **Reset the board. Board will now boot up into Linux.**

iii) Board Startup

Once the redboot is loaded.

1. **Power on the board and wait for Linux to boot.**

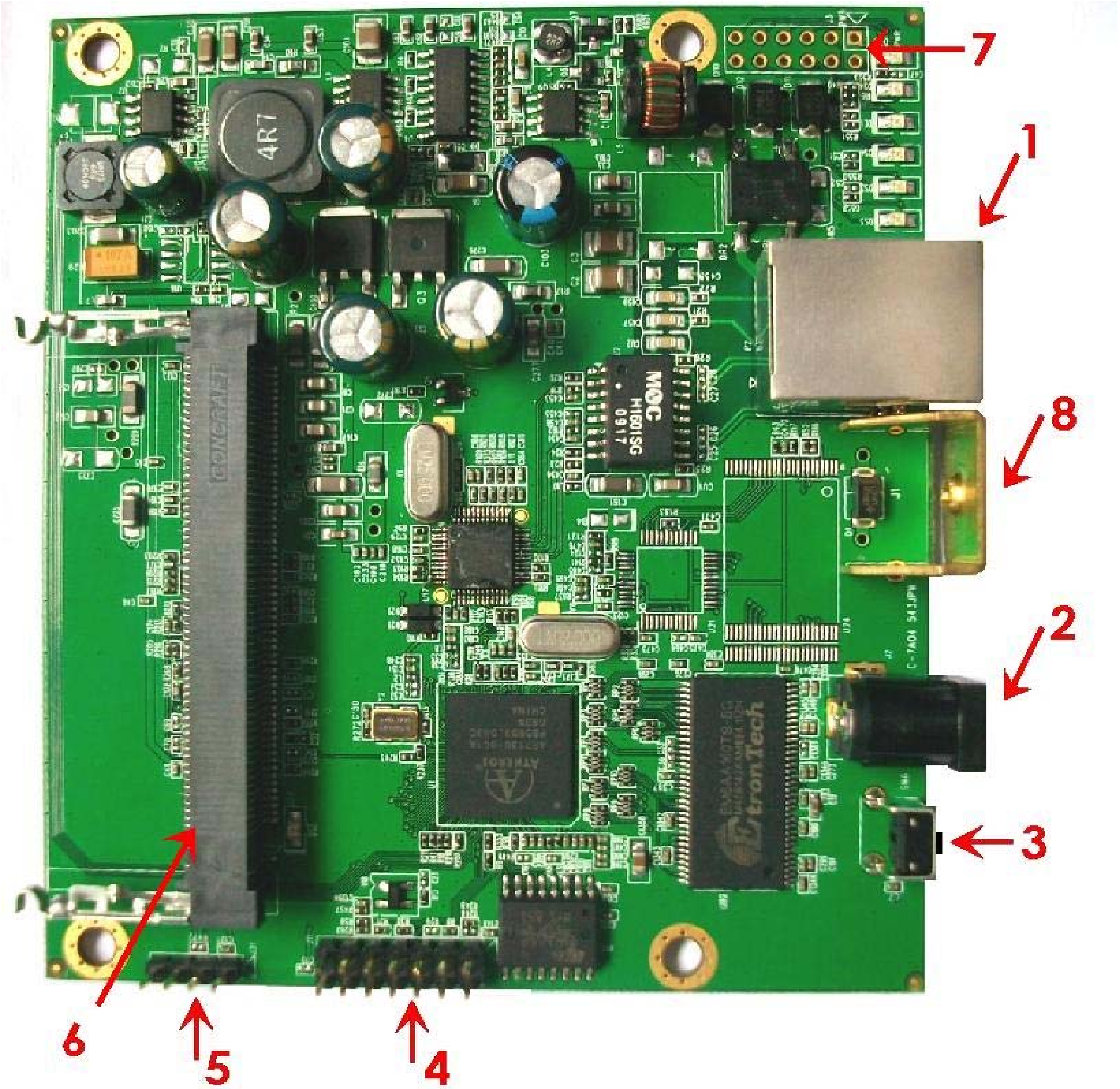
   Note that Redboot is configured to automatically boot into Linux.

2. **Log into the board**

   User: root Password: 5up

Appendix I

Board Features



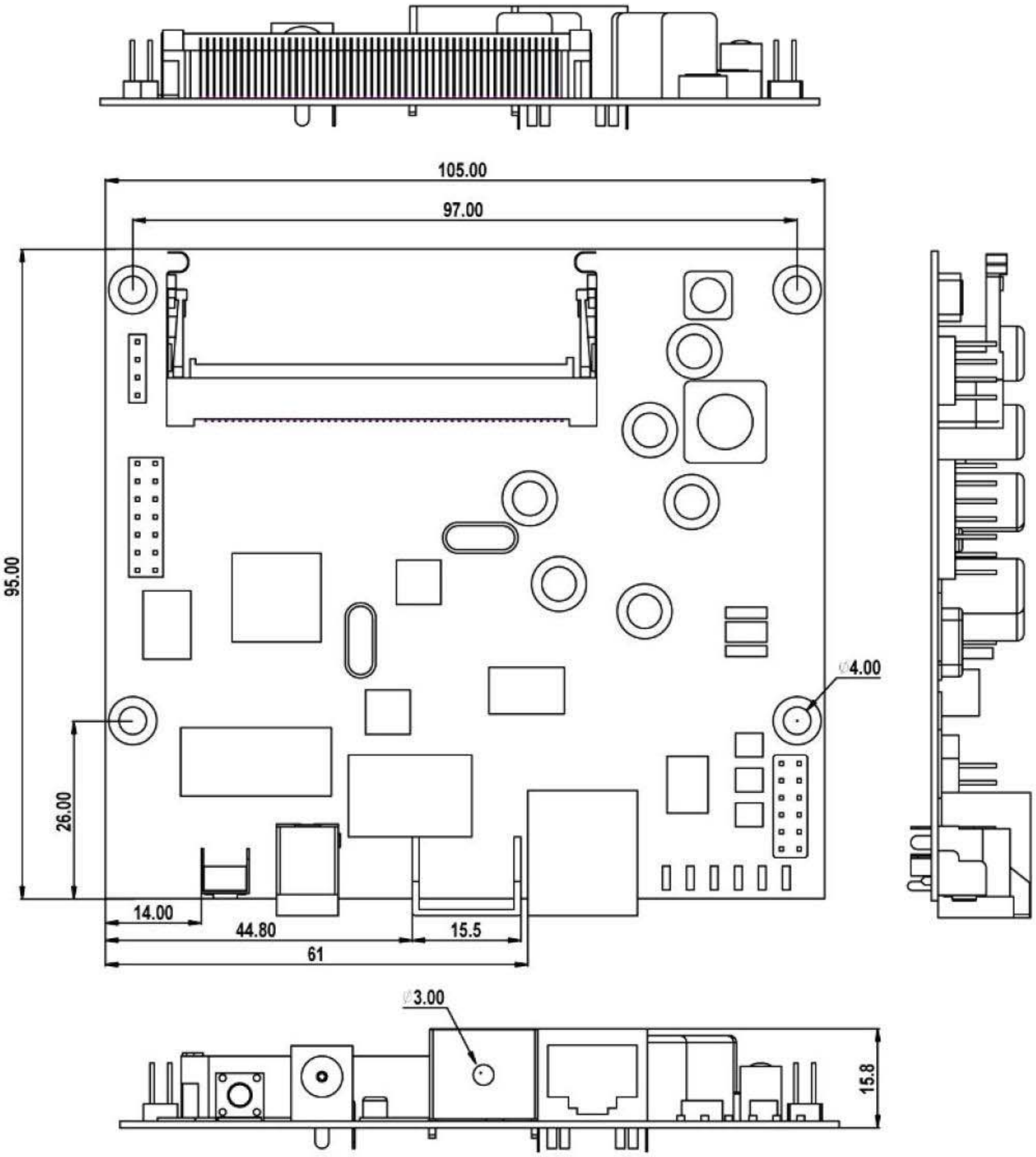
Top Side Of Board

No:	Feature	Descriptions
1	Ethernet cum PoE port	10/100Mbps Ethernet port
2	DC Jack	12V - 24V DC Supply
3	Reset button	For board reset and startup mode control
4	JTAG port	JTAG jumper header for programming
5	Serial port	Serial port connection header
6	mini-PCI slot	9.2mm high Type IIIB mini-PCI slots
7	LED Pin Header	Connect to enclosure LED
8	Ground Terminal	Connect to Earth ground point

Appendix II

Board Drawing and Dimensions

DIMENSIONS DRAWING



105.00
97.00
95.00
26.00
14.00
44.80
15.5
61
∅4.00
∅3.00
15.8