MiraBox User Guide   **-Sep18, 2012**

Contents

A. Package contents

1 · Standard package contents
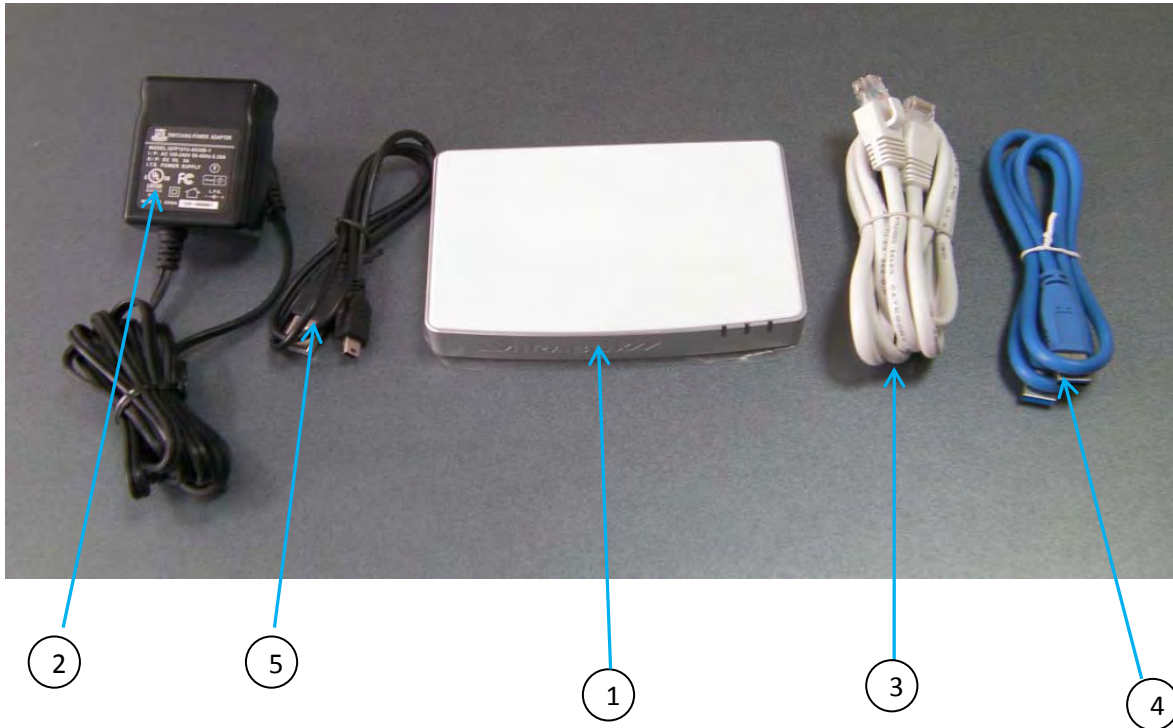
	MiraBoxstandard content List		Remark
1	MiraBox	1 unit	Mirabox main unit
2	AC-DC Power Adapter	1 pc	Input 90-240VAC / output 5V,3A DC
3	Ethernet Cable	1 pc	Cat 5e
4	USB3.0 Cable	1 pc	
5	Mini-USB Cable	1 pc	For debug console use
6	Quick reference card	1 pc	
7	Warranty card	1 pc	

GLOBALSCALE
TECHNOLOGIES, INC.

2. Optional package contents
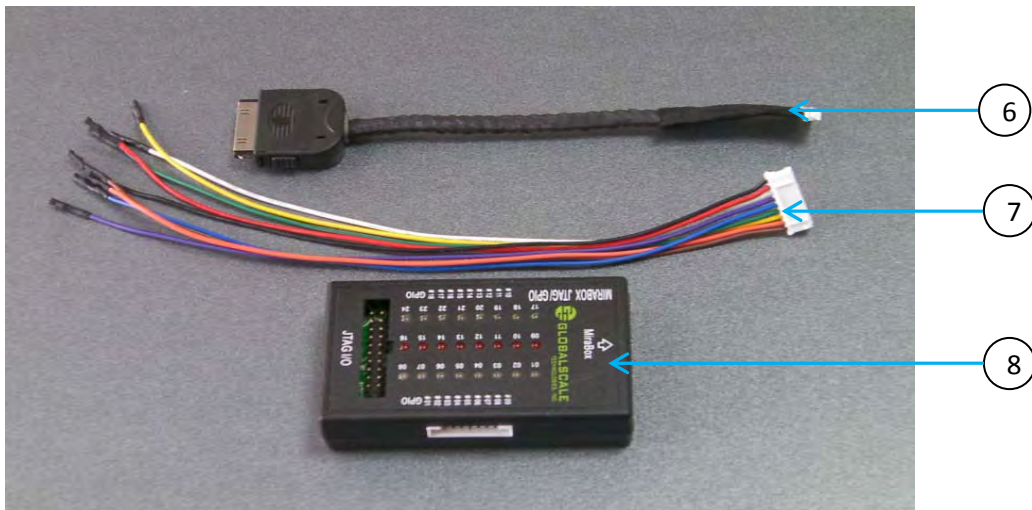
	MiraBoxOptional Content List		
6	Multi I/O Cable	1 pc	Connect from MiraBox to JTAG/GPIO box
7	GPIO Cable	1 pc	For GPIO port connection
8	MiraBox JTAG/GPIO box	1 pc	External JTAG/GPIO box for debugging

B. MiraBox appearance and connecting ports
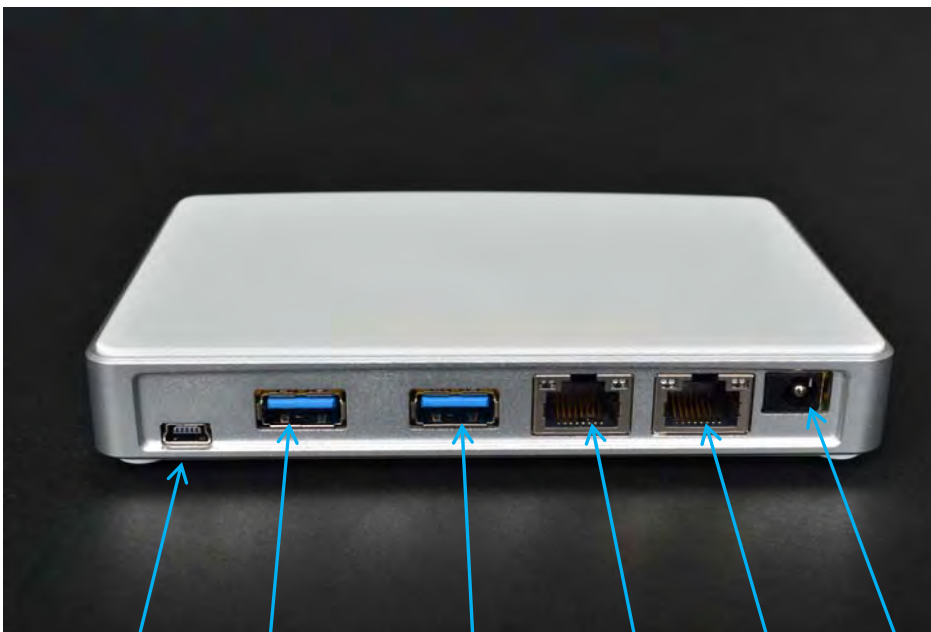
Mirabox ports description

	Connection port	Description	Remark
1	Power Port	DC 5V/3A port	
2	RJ45 #1	Gigabit Ethernet port1	
3	RJ45 #2	Gigabit Ethernet port2	
4	USB 3.0 port#1	USB 3.0 high speed host	
5	USB 3.0 port#2	USB 3.0 high speed host	
6	Mini USB console port	Debug console	Connect to PC USB port
7	Multi-I/O port	JTAG and GPIO port	Connect to external JTAG/GPIO box for system development.
8	Micro SD slot	External Micro-SD slot	
9	Reset button hole	System reset button	Reset through GPIO

C. LED indication
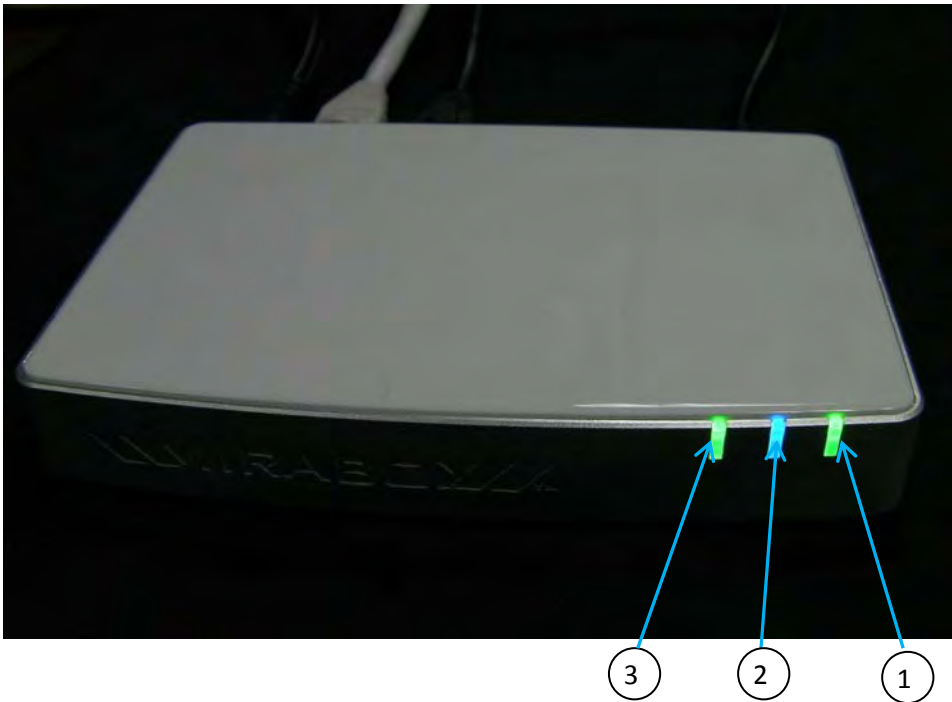
LED indication table

	LED	Color/ Pattern	Description
1	Power on LED	Solid green	Upon power on, this LED lights up
2	WiFi AP	Blinking blue	Indicate WiFi AP mode is activated as default after boot up
		Off	WiFi AP mode is not activated
3	WiFi client	Blinking green	WiFi client mode is activated
		Off	WiFi client mode is not activated

GLOBALSCALE
TECHNOLOGIES, INC.

D. System console and debugging

1. Driver and tool installation.

   (1) Prepare one Windows PC
   (2) Download the serial communication tool "putty.exe".
   (3) Downloadthe driver "2KXPVDock.exe"forProlific-USB-to-Serial-Comm-Portat our Website.
       http://www.globalscaletechnologies.com/t-downloads.aspx
   (4) Install driver on your PC (only for the first time)

   (4-1) Connect Mirabox to your Windows PC

Connect Mini USB cable here.The other end connects to computer's USB port.

   (4-2) The first time when you connect Mirabox to the Windows computer you will be asked to install the driver
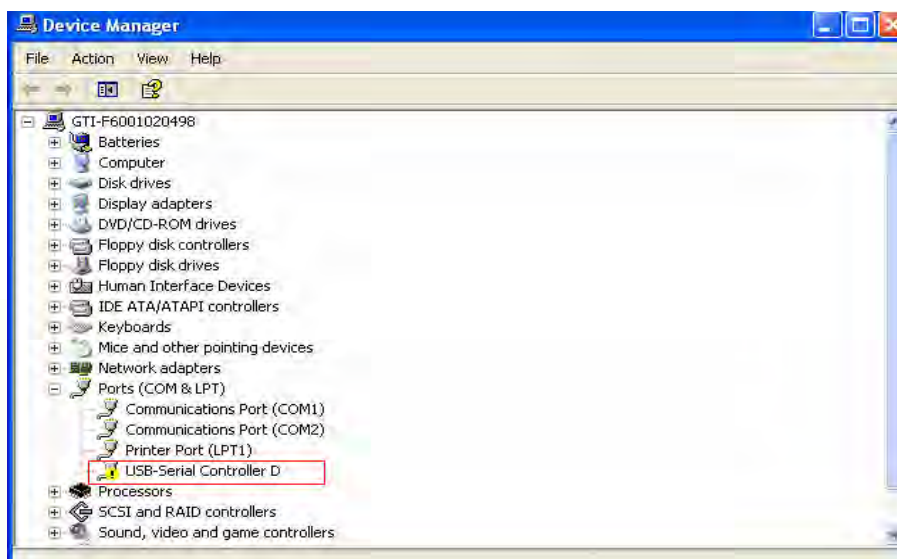
(4-3) Run file "2KXPVDock.exe"and the driver will be installed to the PATH C:\cabs\2KXPVDock".
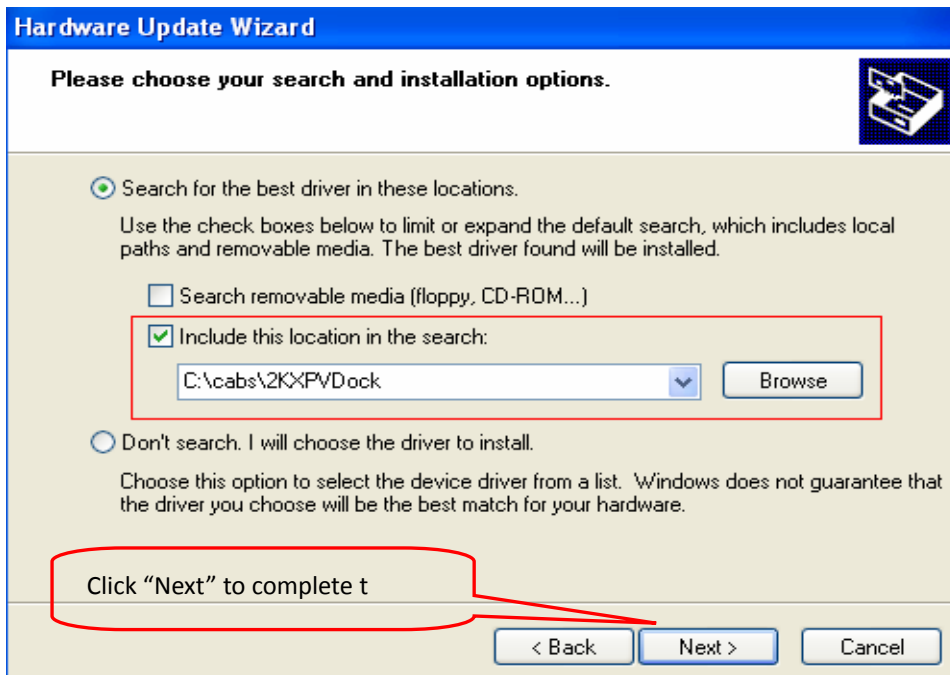
(5) Find out the com port of your debugger on your windows PC.

(5-1) Right click [My Computer] → [Properties] → [Hardware] → [Device Manager]

GLOBALSCALE
TECHNOLOGIES, INC.
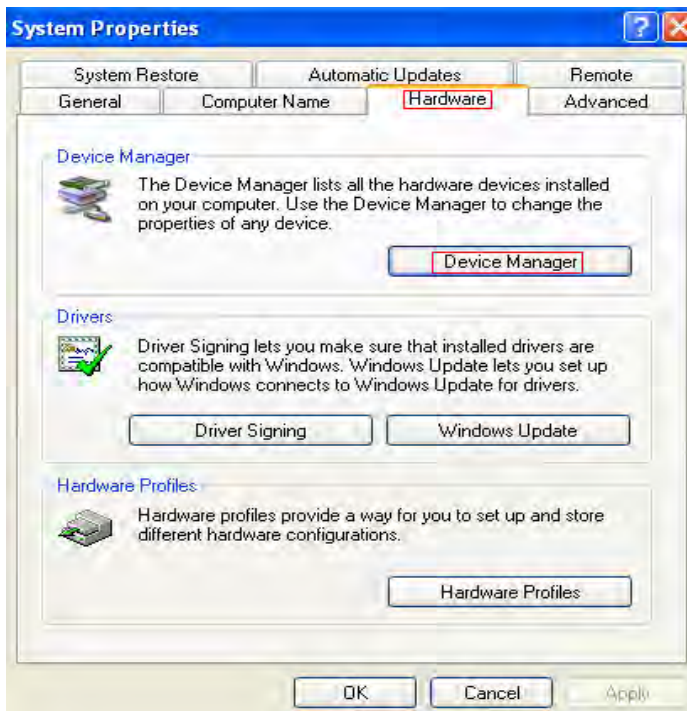
(5-2) Find out your com port as below (here is com3 for example)
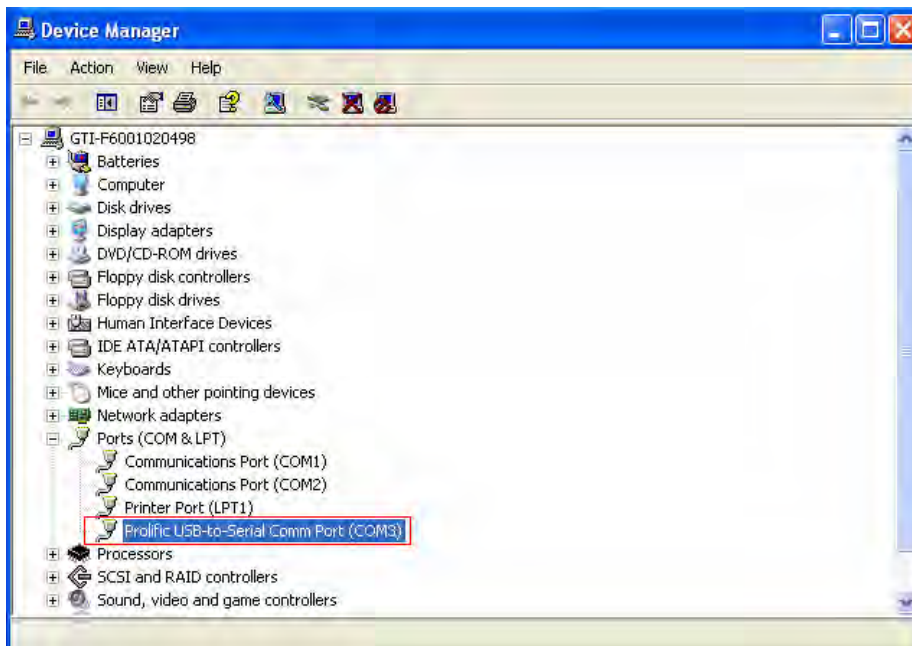
(6) Run "putty.exe"and configure as below on yourWindows PC.

The "Serial line" should be configured the same as the COM port in Device Manager

Give it a name and save for easy execution next time

Click open
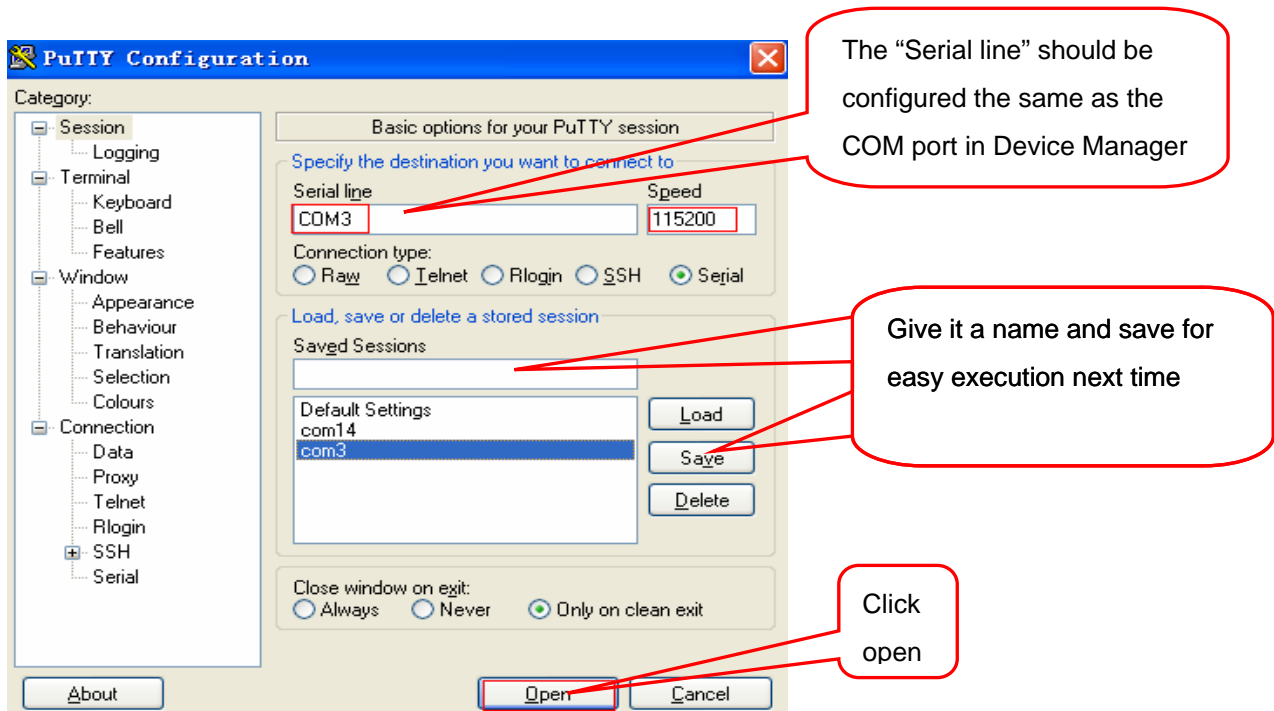
2. Go into debugging console.

(1) Power on the MiraBoxand you will see messages on screen as below

```
U-Boot 2009.08 (Sep 16 2012 - 22:50:06)Marvell version: 1.1.2 NQ
U-Boot Addressing:
        Code:              00600000:006AFFF0
        BSS:               006F8E40
        Stack:             0x5fff70
        PageTable:         0x8e0000
        Heap address:      0x900000:0xe00000
Board: DB-88F6710-BP
SoC:   MV6710 A1
CPU:   Marvell PJ4B v7 UP (Rev 1) LE
       CPU @ 1200Mhz, L2 @ 600Mhz
       DDR @ 600Mhz, TClock @ 200Mhz
       DDR 16Bit Width, FastPath Memory Access
PEX 0: Detected No Link.
PEX 1: Root Complex Interface, Detected Link X1
DRAM:   1 GB
        CS 0: base 0x00000000 size 512 MB
        CS 1: base 0x20000000 size 512 MB
        Addresses 14M - 0M are saved for the U-Boot usage.
NAND:  1024 MiB
Bad block table found at page 262016, version 0x01
Bad block table found at page 261888, version 0x01
FPU not initialized
USB 0: Host Mode
USB 1: Host Mode
Modules/Interfaces Detected:
        RGMII0 Phy
        RGMII1 Phy
        PEX0 (Lane 0)
        PEX1 (Lane 1)
phy16= 72
phy16= 72
MMC:   MRVL_MMC: 0
Net:   egiga0 [PRIME], egiga1
Hit any key to stop autoboot:  0
Marvell>>
```

(2) You can press any key to stop auto-boot when you see the bootdelay timer is counting down.
After entering the uboot prompt, you can also change the uboot environment variables such asbootdelaytime, Ipaddr,serverip and so on.

(3) If no key was pressed to interrupt the uboot, it will continue running to the login screen where urges you to input the login name and password, here is the default login information.

Login :root

Password: nosoup4u

```
Bluetooth: BNEP (Ethernet Emulation) ver 1.3
===========================================================================
BT does not work by default,please execute the following command to enable
    turnon_bt.sh----on, turnoff_bt.sh-----off
===========================================================================
Bluetooth: SCO (Voice Link) ver 0.6
Bluetooth: SCO socket layer initialized

Debian GNU/Linux 6.0 mirabox-debian ttyS0

mirabox-debian login:
```

(4) Now you are the root user and have the full control of the mirabox

3. WiFi AP mode testing

MiraBoxServer has a built-in WiFi module which is in compliance with 802.11 b/g/n standard. The WiFican work as client or AP modebut only one at a time. The default mode is AP modeevery time when it powers on, and the indication light D6 is blinking blue.

Here are steps for testing:

(1) Enter command and you will see message for uap0 device.

#ifconfig

```
root@mirabox-debian:~# ifconfig
eth0      Link encap:Ethernet  HWaddr f0:ad:4e:01:a3:ea
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:8

eth1      Link encap:Ethernet  HWaddr f0:ad:4e:01:a3:eb
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:10

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:480 (480.0 B)  TX bytes:480 (480.0 B)

uap0      Link encap:Ethernet  HWaddr 00:90:a2:4a:63:44
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```
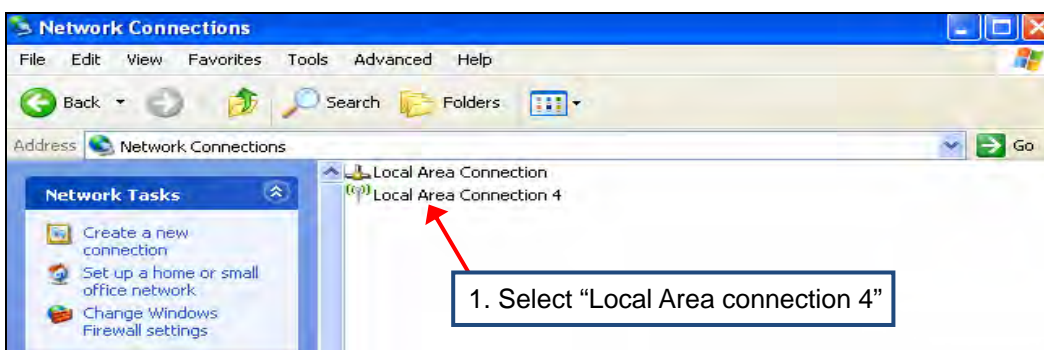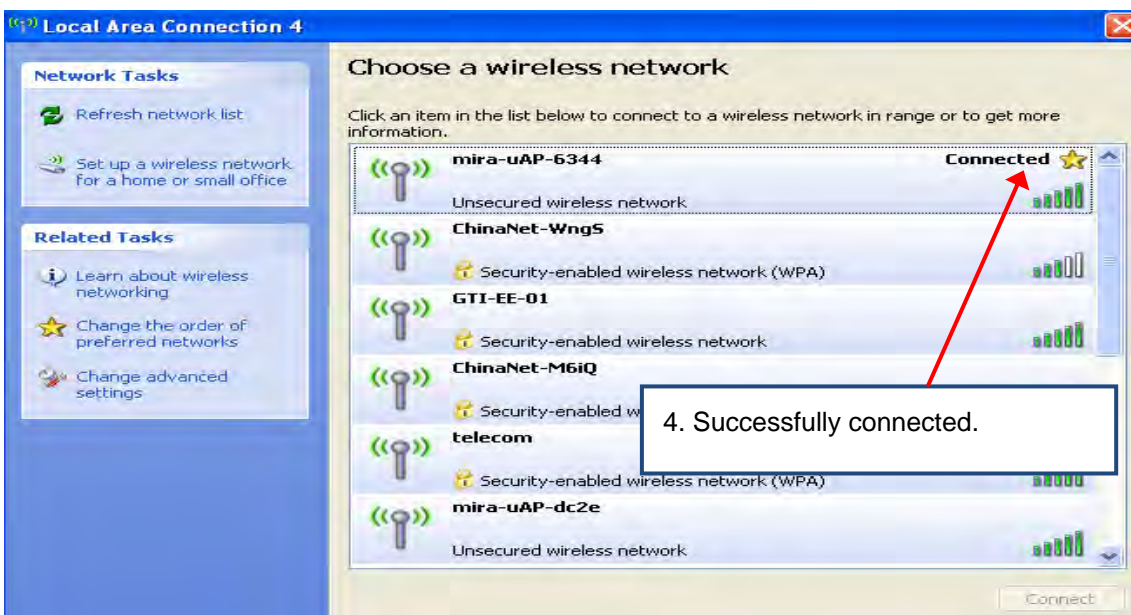
(2) Prepare one computer installed with Wi-Fi Lan card, here we use computer with Windows XP operating system for example.

(3) Go to "Network Connections"as shown below..

1. Select "Local Area connection 4"

GLOBALSCALE
TECHNOLOGIES, INC.

2. Select "View Available Wireless Networks"

3. Choose name with "mira-uAP-6344", here digits"6344" is the last four digits of MAC address for the MiraBoxWiFi

Then click "Connect".

4. Successfully connected.

GLOBALSCALE
TECHNOLOGIES, INC.

4. WIFI Client mode testing

(1) Switchto WiFiclient mode by giving command as following.

wlan.sh

```
root@mirabox-debian:~# wlan.sh
Stopping very small Busybox based DHCP server: Stopped /usr/sbin/udhcpd (pid 1911).
udhcpd.
Stopping DNS forwarder and DHCP server: dnsmasq.
WLAN FW already running! Skip FW download
WLAN FW is active
ADDRCONF(NETDEV_UP): mlan0: link is not ready
```

When done successfully, the LED(D7) is blinking green and mlan0 is activated.

(2) To check mlan0 with ifconfig command

ifconfig

```
root@mirabox-debian:~# ifconfig
eth0      Link encap:Ethernet   HWaddr f0:ad:4e:01:a3:ea
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:8

eth1      Link encap:Ethernet   HWaddr f0:ad:4e:01:a3:eb
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:10

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:480 (480.0 B)  TX bytes:480 (480.0 B)

mlan0     Link encap:Ethernet   HWaddr 00:90:a2:4a:63:44
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```
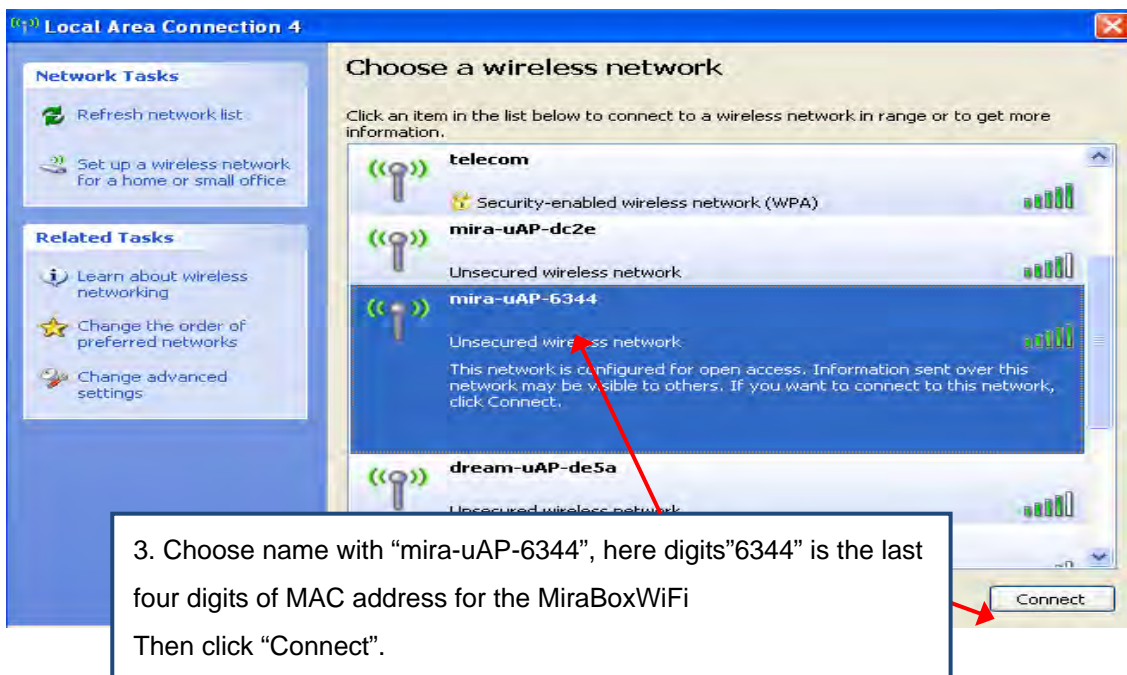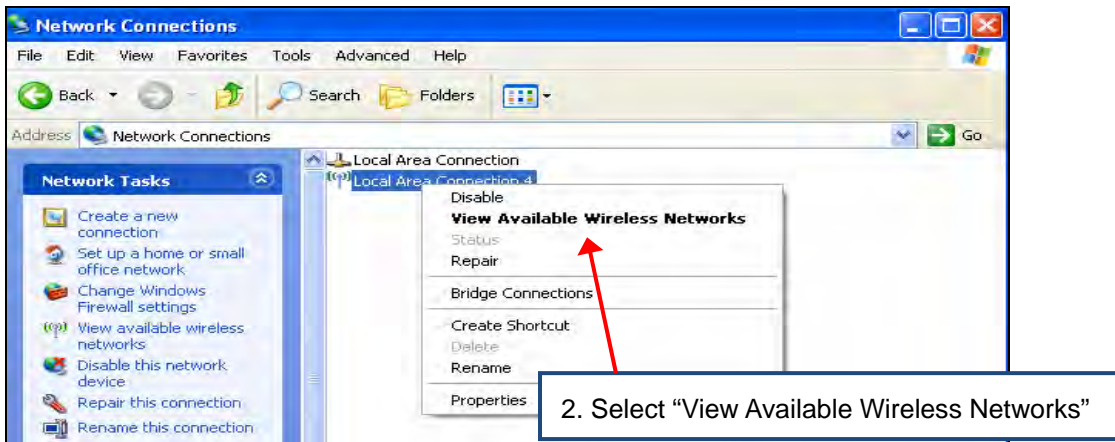
GLOBALSCALE
TECHNOLOGIES, INC.

(3) Scan for WiFi AP near-by

iwlist mlan0 scan

```
root@mirabox-debian:~# iwlist mlan0 scan
mlan0     Scan completed :
          Cell 01 - Address: 08:18:1A:90:97:E9
                    ESSID:"ChinaNet-WngS"
                    Mode:Master
                    Frequency:2.412 GHz (Channel 1)
                    Quality:3/5  Signal level:-68 dBm  Noise level:-96 dBm
                    Encryption key:on
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                              9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
                              48 Mb/s; 54 Mb/s
                    Extra:Beacon interval=100
                    IE: WPA Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
                    IE: Unknown: DD180050F2020101000003A4000027A4000042435E0062322F00
                    IE: Unknown: DD1E00904C330C181EFF0000000000000000000000000000000000000000
                    IE: Unknown: DD1A00904C3401000000000000000000000000000000000000000000
                    IE: Unknown: DD0600E04C020160
                    Extra:band=bg
          Cell 02 - Address: 40:16:9F:44:CB:2C
                    ESSID:"GTI-EE-01" [2]
                    Mode:Master
                    Frequency:2.412 GHz (Channel 1)
                    Quality:5/5  Signal level:-42 dBm  Noise level:-96 dBm
                    Encryption key:on
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                              9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
                              48 Mb/s; 54 Mb/s
                    Extra:Beacon interval=100
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (2) : CCMP TKIP
                        Authentication Suites (1) : PSK
                    IE: Unknown: DD180050F2020101020003A4000027A4000042435E0062322F00
                    IE: Unknown: DD0900037F01010000FF7F
                    IE: Unknown: DD810050F204104A0001101044000102103B0001031047001000000000000000
00000040169F44CB2C1021000754502D4C494E4B10230009544C2D57523734304E10240003312E3010420003312E301
4000800060050F204000110110019576972656C65737320526F7574657220544C2D57523734304E1008000200861030
0101
                    Extra:band=bg
          Cell 03 - Address: C8:3A:35:48:A2:28
                    ESSID:"Tenda_48A228" [3]
                    Mode:Master
```

GLOBALSCALE
TECHNOLOGIES, INC.

(4) Connect to WiFi AP

　　# iwconfig mlan0 essid<AP's ID>

　　# dhclient mlan0

　　# ifconfig mlan0

　　#ping 192.168.1.1

The below screenshot is a example for how to connect a mlan device. If you can see IP address (for example:
192.168.1.102) means you have already connected and got an IP.

(5) Switch back to WiFi AP mode

　　There are two useful script files:

　　Client_ap.sh – switch from wifi client mode to wifi AP mode

　　Wlan.sh – switch from wifi AP mode to wifi client mode

5. Gigabit ethernet ports

(1) Connect the two Gigabit Ethernet ports to Gigabit switch by Network cables. Normally it will get an IP address assigned by DHCP.

Enter command as below to check:

#ifconfig

```
Mon Sep 17 06:20:41 2012  -0.252600 seconds
root@mirabox-debian:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:43:9d:2b:2d
          inet addr:10.10.10.152  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::250:43ff:fe9d:2b2d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1309 errors:0 dropped:0 overruns:0 frame:0
          TX packets:138 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:136008 (132.8 KiB)  TX bytes:15562 (15.1 KiB)
          Interrupt:8

eth1      Link encap:Ethernet  HWaddr 00:50:43:81:2b:2d
          inet addr:10.10.10.189  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::250:43ff:fe81:2b2d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1359 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:141431 (138.1 KiB)  TX bytes:1278 (1.2 KiB)
          Interrupt:10

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:480 (480.0 B)  TX bytes:480 (480.0 B)

uap0      Link encap:Ethernet  HWaddr 94:db:c9:d2:ae:ec
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

(2) Testing the speed of Gigabit Ethernet ports

#ethtool eth0

Normally you will see the information as below:

```
root@mirabox-debian:~# ethtool eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Half 1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: No
        Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                             100baseT/Half 100baseT/Full
                                             1000baseT/Half 1000baseT/Full
        Link partner advertised pause frame use: No
        Link partner advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 0
        Transceiver: internal
        Auto-negotiation: on
        Link detected: yes
```

6. USB 3.0 port

(1) Plug in the usb3.0 hard disk or flash disk to the USB port then you can see some driver messages as below of this device

```
root@mirabox-debian:~# hub 3-0:1.0: unable to enumerate USB device on port 2
usb 3-4: new SuperSpeed USB device using xhci_hcd and address 3
xhci_hcd 0000:01:01.0: WARN: short transfer on control ep
xhci_hcd 0000:01:01.0: WARN: short transfer on control ep
xhci_hcd 0000:01:01.0: WARN: short transfer on control ep
xhci_hcd 0000:01:01.0: WARN: short transfer on control ep
scsi4 : usb-storage 3-4:1.0
scsi 4:0:0:0: Direct-Access     Seagate  Portable         SF06 PQ: 0 ANSI: 4
sd 4:0:0:0: [sdc] 976773164 512-byte logical blocks: (500 GB/465 GiB)
sd 4:0:0:0: [sdc] Write Protect is off
sd 4:0:0:0: [sdc] Assuming drive cache: write through
sd 4:0:0:0: [sdc] Assuming drive cache: write through
 sdc: sdc1
sd 4:0:0:0: [sdc] Assuming drive cache: write through
sd 4:0:0:0: [sdc] Attached SCSI disk
```

(2) View the usb3.0 hard disk

#fdisk-l

you will see below messages of the usb3.0 devicewhich is usually shown as /dev/sdb*、/dev/sdc*.

```
Disk /dev/sdc: 500.1 GB, 500107859968 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x165060a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1       60802   488384032+   7  HPFS/NTFS
```

(3) Testing the access speed of the usb3.0 ports

#hdparm-t   /dev/sdc1

```
root@mirabox-debian:~# hdparm -t /dev/sdc1

/dev/sdc1:
 Timing buffered disk reads: 118 MB in  3.02 seconds =  39.10 MB/sec
```
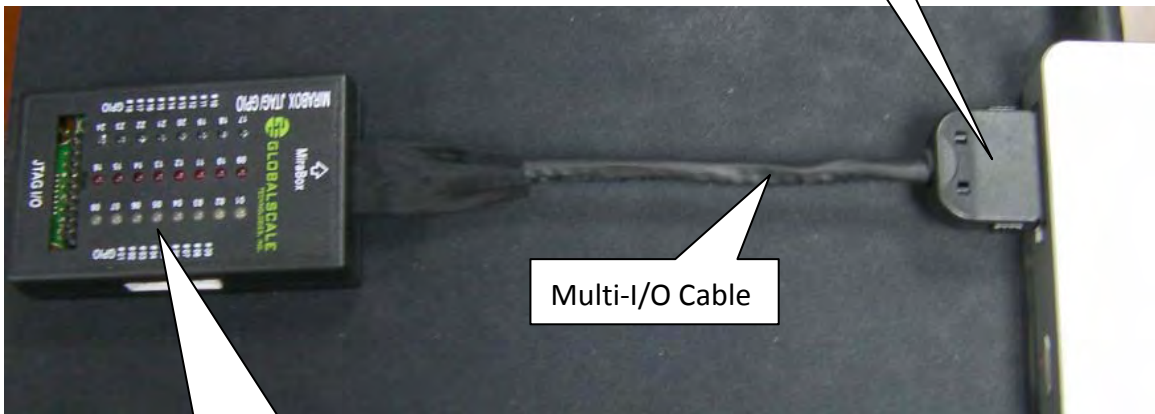
7. Multi-IO port

Connect the MIRABOX JTAG/GPIO box to the Multi-IO port via Multi-IO cable.
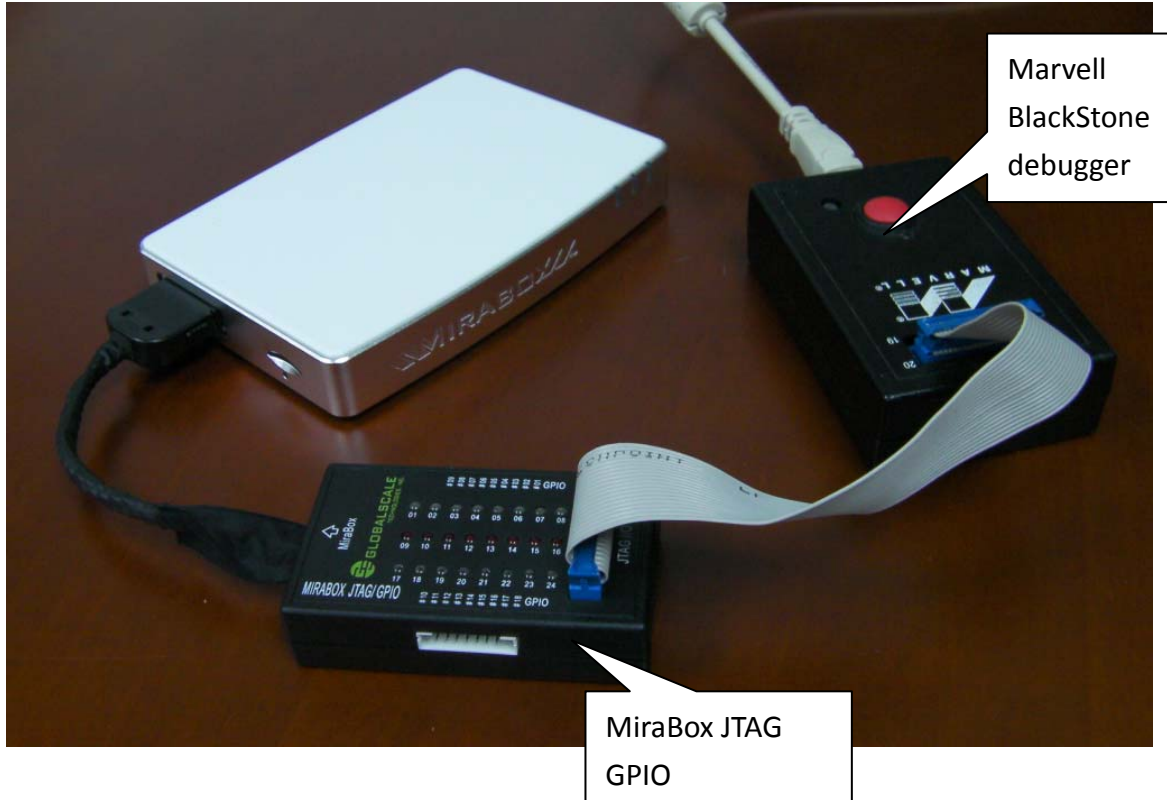
Multi-I/O port

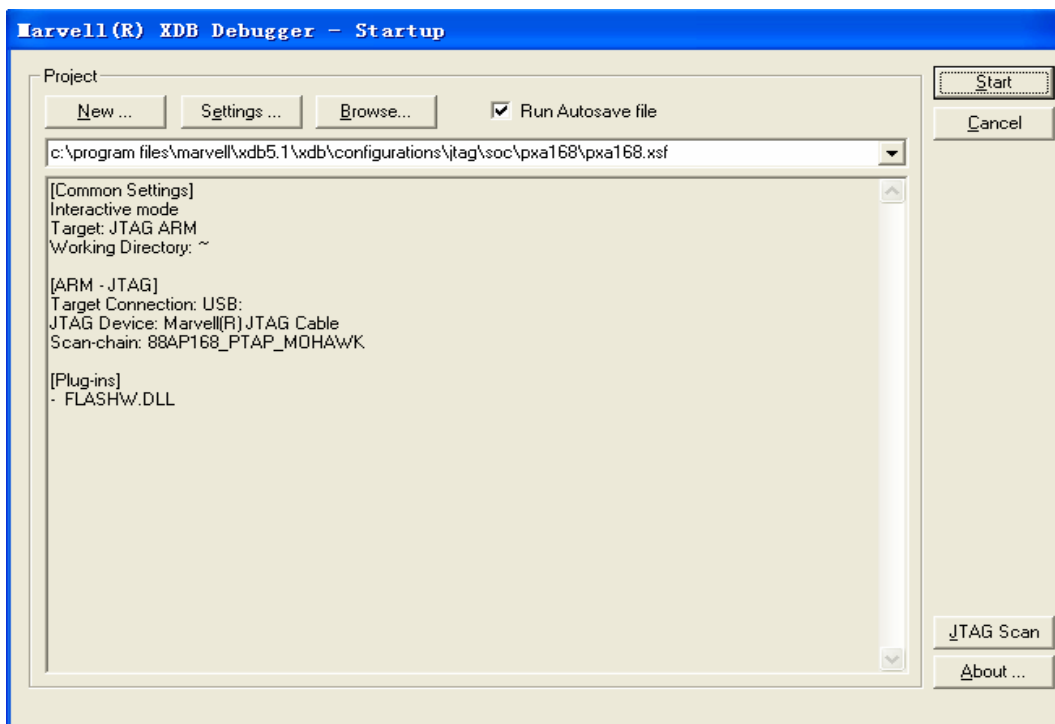Multi-I/O Cable

MiraBox JTAG/GPIO box

(1) JTAG interface
Please connect the debugger to the 20-pin JTAG slot as shown below, here we use the Marvell
BlackStone debugger and XDB(Marvell eXtreme Debugger 5.1) software on Windows PC for
example.

Debugger connection

Marvell
BlackStone
debugger

MiraBox JTAG
GPIO

Run XDB on Windows PC

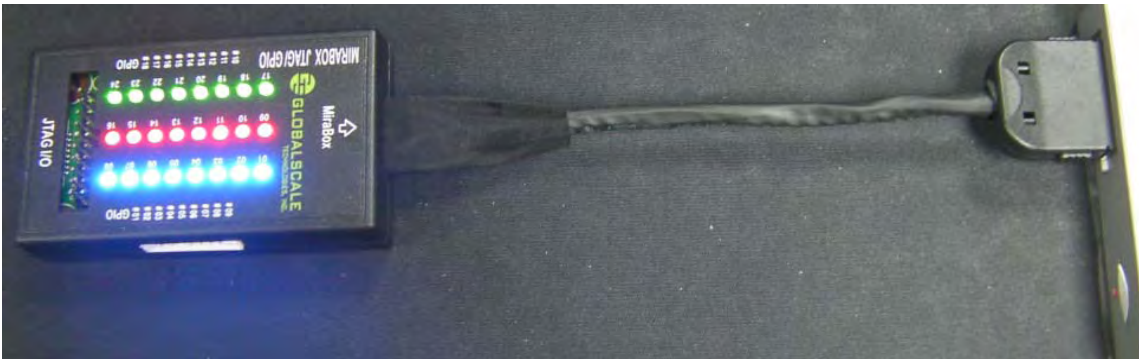GLOBALSCALE
TECHNOLOGIES, INC.

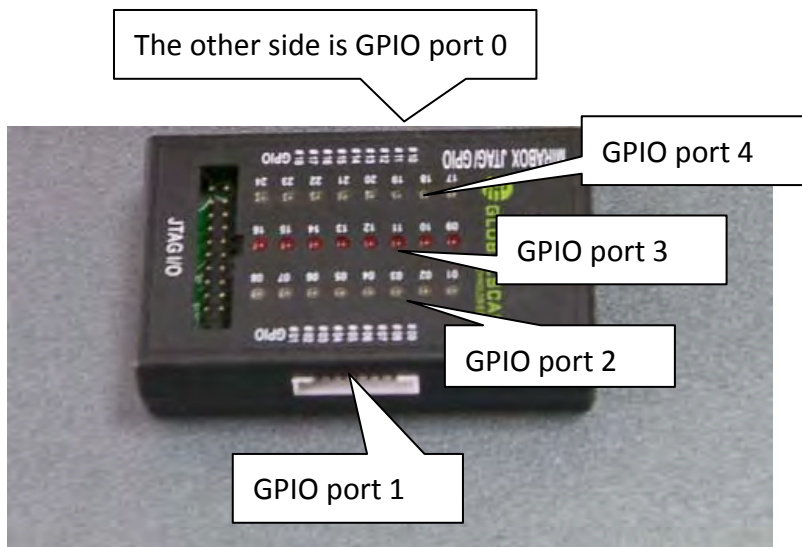(2) GPIO interface and control

(2-1) There is a demo program to show the GPIO LEDs by entering command as below:

#ledtest1

This will light up the LEDs for 3 seconds then light off.

（2-2）Control the LED by yourself

The other side is GPIO port 0

GPIO port 4

GPIO port 3

GPIO port 2

GPIO port 1

\

a. preparation

1. Prepare a host pc with linux (as:fedora 14) installed, and download the cross tool chain

   "arm-marvell-linux-gnueabi-vfp.tar.bz2" from our

   website http://www.plugcomputer.org/downloads/d2plug/ or

   http://www.globalscaletechnologies.com   or

2. Install and configure the tool chain

Copy "arm-marvell-linux-gnueabi-vfp.tar.bz2" to directory /opt

   #cd /opt

   #tar -jxfarm-marvell-linux-gnueabi-vfp.tar.bz2

   #ls

   #vim /root/.bashrc

   Add "export PATH=$PATH:/opt/arm-marvell-linux-gnueabi-vfp/bin" to last line as below, then save and exit.

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

export PATH=$PATH:/opt/arm-marvell-linux-gnueabi-vfp/bin
```

   Enable the PATH

#source /root/.bash_profile

b. compile the program

Here has an example "i2c_led.c" for you to debugging

```
/*i2c_led.c*GTI--Globalscaletechnologies.,INC**/
        #include <stdio.h>
        #include <linux/types.h>
        #include <stdlib.h>
        #include <fcntl.h>
        #include <unistd.h>
        #include <sys/types.h>
        #include <sys/ioctl.h>
        #include <errno.h>
        #define I2C_RETRIES 0x0701
        #define I2C_TIMEOUT 0x0702
        #define I2C_RDWR 0x0707
        #define I2C_M_RD 0x0001
        #define I2C_M_NOSTART          0x4000      /* if I2C_FUNC_PROTOCOL_MANGLING */
     #define I2C_M_REV_DIR_ADDR  0x2000      /* if I2C_FUNC_PROTOCOL_MANGLING */
        #define I2C_M_IGNORE_NAK 0x1000      /* if I2C_FUNC_PROTOCOL_MANGLING */
     #define I2C_M_NO_RD_ACK        0x0800      /* if I2C_FUNC_PROTOCOL_MANGLING */
        #define I2C_M_RECV_LEN         0x0400      /* length will be first received byte */
/*********define struct i2c_rdwr_ioctl_data and struct i2c_msg，they must be consistent to kernel*******/
struct i2c_msg
        {
unsigned short addr;
unsigned short flags;
        #define I2C_M_TEN 0x0010
unsigned short len;
unsigned char *buf;
        };
struct i2c_rdwr_ioctl_data
        {
struct i2c_msg *msgs;
intnmsgs;/* the nmsgs decide the num of start signal*/
        };
/***********the main program***********/
int main()
        {
intfd,ret;
```

```
                         struct i2c_rdwr_ioctl_data e2prom_data;
fd=open("/dev/i2c-0",O_RDWR);/*     /dev/i2c-0 is registered to the system */
if(fd<0)
                         {
perror("open error");
                         }
                         e2prom_data.nmsgs=2;
                         e2prom_data.msgs=(struct i2c_msg*)malloc(e2prom_data.nmsgs*sizeof(struct i2c_msg));
if(!e2prom_data.msgs)
                         {
perror("malloc error");
exit(1);
                         }
ioctl(fd,I2C_TIMEOUT,1);/*timeout */
ioctl(fd,I2C_RETRIES,2);/*retries times*/
/***write data to e2prom**/
                         e2prom_data.nmsgs=1;
                         (e2prom_data.msgs[0]).len=2;
                         (e2prom_data.msgs[0]).addr=0x25;//e2prom device address
                         (e2prom_data.msgs[0]).flags=0;    //write
                         (e2prom_data.msgs[0]).buf=(unsigned char*)malloc(2);
  /**********    control the GPIO OP-0 *********************************/
                         (e2prom_data.msgs[0]).buf[0]=0x18;// e2prom write address
                         (e2prom_data.msgs[0]).buf[1]=0x0;//the data to write
                              ret=ioctl(fd,I2C_RDWR,(unsigned long)&e2prom_data);
if(ret<0)
                         {
perror("ioctl error2");
                         }
  /**********    control the GPIO OP-1 *********************************/
(e2prom_data.msgs[0]).buf[0]=0x19;// e2prom write address
                         (e2prom_data.msgs[0]).buf[1]=0x0;//the data to write
                              ret=ioctl(fd,I2C_RDWR,(unsigned long)&e2prom_data);
if(ret<0)
                         {
perror("ioctl error2");
                         }
```

```
/**********    control the GPIO OP-2 ***********************************/
    //turn on LED
(e2prom_data.msgs[0]).buf[0]=0x1a;// e2prom write address
                (e2prom_data.msgs[0]).buf[1]=0x0;//the data to write
                    ret=ioctl(fd,I2C_RDWR,(unsigned long)&e2prom_data);
if(ret<0)
                {
perror("ioctl error3");
                }
  //turn off LED
sleep(1);                            //delay 1 second
                (e2prom_data.msgs[0]).buf[0]=0x1a;// e2prom write address
                (e2prom_data.msgs[0]).buf[1]=0xFF;//the data to write
                    ret=ioctl(fd,I2C_RDWR,(unsigned long)&e2prom_data);
if(ret<0)
                {
perror("ioctl error3");
                }
  /**********    control the GPIO OP-3 ***********************************/
  //turn on LED
sleep(1);
                (e2prom_data.msgs[0]).buf[0]=0x1b;// e2prom write address
                (e2prom_data.msgs[0]).buf[1]=0x0;//the data to write
                    ret=ioctl(fd,I2C_RDWR,(unsigned long)&e2prom_data);
if(ret<0)
                {
perror("ioctl error4");
                }
  //turn off LED
sleep(1);
(e2prom_data.msgs[0]).buf[0]=0x1b;// e2prom write address
                (e2prom_data.msgs[0]).buf[1]=0xFF;//the data to write
                    ret=ioctl(fd,I2C_RDWR,(unsigned long)&e2prom_data);
if(ret<0)
                {
perror("ioctl error4");
                }
```

```
    /**********    control the GPIO OP-4 *********************************/
    //turn on LED
sleep(1);
                (e2prom_data.msgs[0]).buf[0]=0x1c;// e2prom write address
                (e2prom_data.msgs[0]).buf[1]=0x0;//the data to write
                    ret=ioctl(fd,I2C_RDWR,(unsigned long)&e2prom_data);
if(ret<0)
                {
perror("ioctl error5");
                }
    //turn off LED
sleep(1);
                (e2prom_data.msgs[0]).buf[0]=0x1c;// e2prom write address
                (e2prom_data.msgs[0]).buf[1]=0xFF;// the data to write
                    ret=ioctl(fd,I2C_RDWR,(unsigned long)&e2prom_data);
if(ret<0)
                {
perror("ioctl error5");
                }
        printf("testing ok\n");
close(fd);
return 0;
            }
```

Copy this program to file "i2c_led.c" and save to "/home" directory or other else, and compile it use cross-compile tool just have been installed.

#cd /home

#arm-marvell-linux-gnueabi-gcc -o led i2c_led.c

Then copy the executable file "led" to Mirabox and run it. You can see the LED is controlled by youself.

8. Reset

6、GPIO reset button hole.

(1) GPIO reset button

The reset button is connected to one GPIO which means controlled by software.

When you press down this button by using a sharp pin, it arouses attention of CPU then CPU asserts the master reset low signal to start the system reset session.

Below is the reboot screen after reset

Note: Since this reset function is controlled by software so it will not be handled during boot-up session.

(2) More push button functions

System developer can program this button to have more functions by different press-then-release delay time.

9.   Download sites

To download the files for MiraBox server, please visit:

http://www.globalscaletechnologies.com/t-downloads.aspx

Other useful resource links are:

http://www.plugcomputer.org/

http://plugcomputer.org/plugwiki/index.php/GuruPlug

10. FCC STATEMENT

1. This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

   (1) This device may not cause harmful interference.

   (2) This device must accept any interference received, including interference that may cause undesired operation.

2. Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation.

This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

   --- Reorient or relocate the receiving antenna.

   --- Increase the separation between the equipment and receiver.

   --- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

   --- Consult the dealer or an experienced radio/TV technician for help.

11. RF warning statement:

   This equipment complies with FCC radiation exposure limits setforth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between theradiator & your body