RSC-W910
Linux BSP User's Manual

Avalue Technology Inc.
7F, 228, Lian-cheng Road, Chung Ho City, Taipei, Taiwan

www.avalue.com.tw
sales@avalue.com.tw
Tel: +886-2-8226-2345
Fax: +886-2-8226-2777

Avalue USA
sales@avalue-usa.com
Tel: +1-732-578-0200
Fax: +1-732-578-0250

BCM Advanced Research
BCMsales@bcmcom.com
Tel: +1-949-470-1888
Fax: +1-949-470-0971

Avalue China
sales.china@avalue.com.cn
Tel: +86-21-5169-3609
Fax: +86-21-5445-3266

1. GENERAL DESCRIPTION

This is a Linux BSP for RSC-W910 with Nuvoton W910P910 MCU. Nuvoton W910P910 MCU is built around an outstanding CPU core : the 16/32 ARM926EJS RISC processor designed by Advanced RISC Machines, Ltd. The ARM926EJS core, offers 8K-byte I-cache and 8K-byte D-cache with MMU, is a low power, general-purpose integrated circuits. This series of micro-controllers are suitable for a high end, high performance and low cost related products as well as general purpose applications.

This Linux BSP contains following items:

_ Linux 2.6.17.14 kernel source code including the drivers for W910P910 MCU.

_ GCC 4.2.1 with EABI support

_ uClibc-0.9.29

_ Binutils-2.17

_ Source code of sample applications for different interfaces, busybox, bash, microwindows, mtdutil., and other open source applications

_ Nuvoton NAND flash bootloader binary file

_ U-Boot source code (for NOR Flash only)

_ Windows side driver and tools

_ Relative documents

2. DEVELOPMENT ENVIRONMENT SETUP

This BSP only provides cross development tool chain in Linux environment. So Linux platform is a must to build Linux kernel/ applications using the cross compiling tool chain in BSP. This platform could be a dedicate Linux server or running on virtual machine.

PC can communicate with NUC900 EV board via different communication interfaces, such as UART, USB, and MAC. Above interfaces could be used to load binary file to RSC-W910 for execution. USB interface is the interface used by Turbo Writer to write NOR/ NAND flash. Fig 1 Development Environment is an example of development environment.

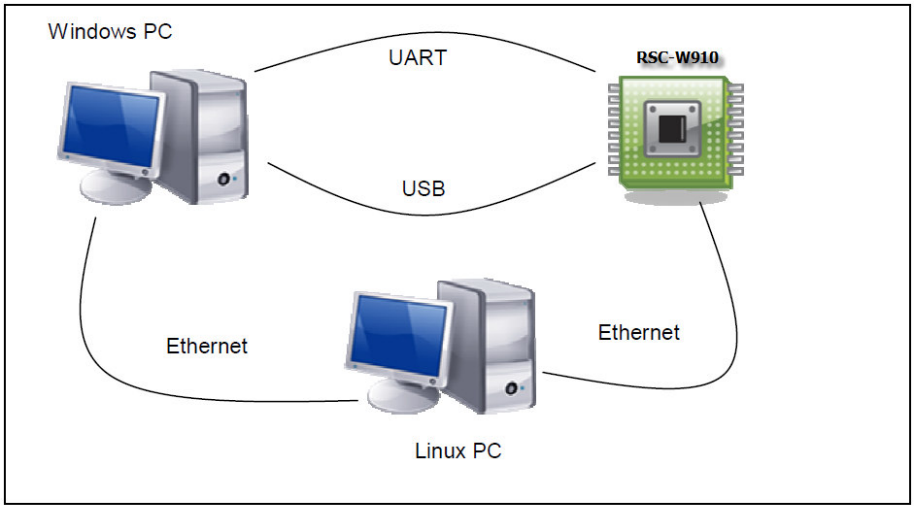


Fig 1 Development Environment

2.1 RSC-W910 Jumper setting

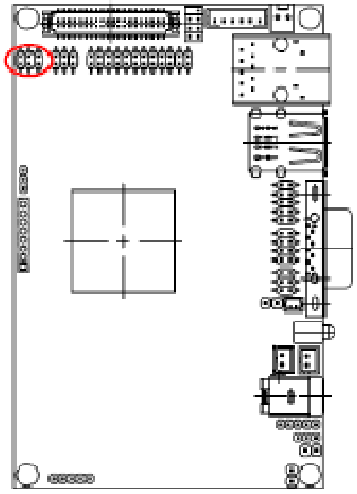
Set RSC-W910 jumper “JSET1” in INSTALL mode (Turbo Writer Mode).

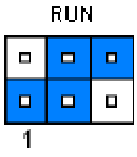
JSET1 =>1-3,4-6: NOR Boot Mode

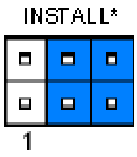
JSET1 =>1-3,2-4: NAND Boot Mode

JSET1 =>3-5,4-6: Turbo Writer Mode

2.3.4 Boot ROM/Flash data bus width (JSET1)







* Default

3. BSP INSTALLATION

The BSP contains four directories. The contents of each directories describes in following table:

Directory Name	Contents
Bootloader	Bootloader and U-boot source code
BSP	Tar ball of Linux related stuffs. Including kernel source code, sample applications, sample rootfs.
Documents	BSP related documents
LCD Support list	List of LCD supported by BSP
Linux Image	Pre-compiled kernel & RAMDISK for 5.7" & 8" RISC Panel PC
NFS	HOW-TO for setup NFS and sample NFS based on Debain 5.0 ARM
Utility	Windows tools and driver

Please copy the tar ball in BSP directory into a Linux machine, untar it and enter the newly created directory with following command:

```
$ tar zxvf W910BSP.tar.gz
$ cd W910BSP
```

The installation script must be executed by root. You can ether su to root, than execute the script:

```
$ su
Password: (Enter password of root)
# ./install.sh
```

Or execute the script with sudo. (Users of Ubuntu or other distribution without root account can use this method to execute the installation script)

```
# sudo ./install.sh
```

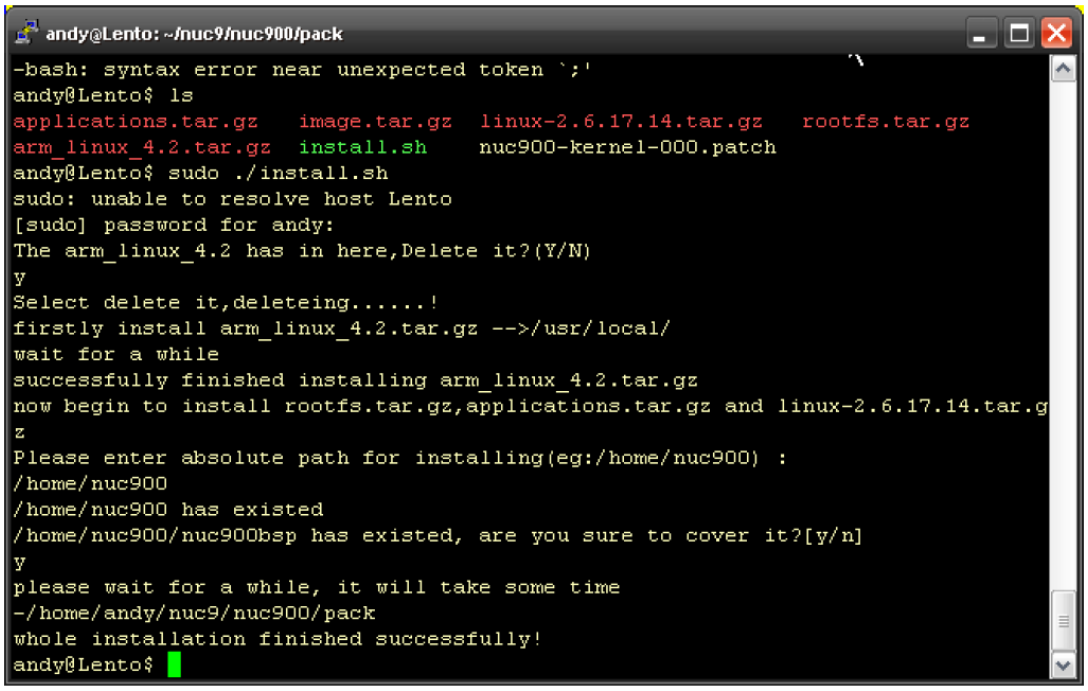


Fig 2 BSP installation

If your Linux server has already installed the arm_linux_4.2 tools, the installation script will ask whether or not to remove the existing tool chain. Otherwise the script will install the tool chain into /usr/local/arm_linux_4.2 without asking. For the first case, if you want to update the tool chain, you can select Y(or yes、y、YES), then hit <enter>.

After install the tool chain, the installation script will ask for the absolute path for install kernel and applications. The table below listed the item will be installed in the specified location.

Directory Name	Description
applications	Source code of sample applications, bash, busybox.
image	Pre-build images and rootfs
Linux-2.6.17.14	Kernel source code
rootfs	Root file system

The installation script will try to configured the installed directory with correct owner and group, and add the path of compiler into $PATH. However, this doesn't work correctly in every Linux distribution. User might need to set the owner/group of installed directory with correct user's name, and add /usr/local/arm_linux_4.2/bin to $PATH manually.

Please logout and re-login after installation complete to make the changes take effect.

4. KERNEL CONFIGURATION AND COMPILATION

4.1 Default configuration

Avalue provides default kernel configuration for RSC-W910. You can follow the way below to load the setting before you compile kernel source code.

```
#cd linux-2.6.17.14
#cp RSC-W910.config .config
#make oldconfig
#make menuconfig
```

After loading the default configuration, type “make” to compile the kernel. The newly created kernel will be copied into image/ directory and the file name is “910kernel”.

4.2 Detailed Kernel configuration

4.2.1 System configuration

Please type “make menuconfig” command in linux-2.6.17.14/ directory to enter kernel configuration menu.

4.2.1.1 Code maturity level selection

Code maturity level options --->
[*] Prompt for development and/or incomplete code/drivers

4.2.1.2 Loadable module support

During boot up, Linux kernel assumes itself could fits into 4MB region. Other kernel modules could be installed into kernel later. Or drivers under debug could link into kernel as a module to avoid update kernel frequently.

Loadable module support --->
[*] Enable loadable module support
[*] Module unloading

4.2.1.3 System Type

In the System Type sub menu, users need to make sure the CPU type is the same with the MCU on RSC-W910.

```
System Type --->
ARM system type (Nuvoton ARM9 NUC900) --->
Nuvoton arm9 cpu type (Nuvoton NCU950) --->
--- NUC900 Boot
[ ] NUC900 Reboot on decompression error
set NUC900 cpu Frequency (200MHZ) --->
--- NUC900 Setup
[ ] NUC900 DMA support
(0) NUC900 UART to use for low-level messages
--- NUC900 Device Interface conflict pre-set
--->select NAND or LCD or KPI support? (LCD driver) --->
--->select Nuvoton MS or SD ? (SD Card ) --->
--->select I2C or USI(SPI) support? (I2C driver) --->
```

4.2.1.4 Boot options

Boot options allows system features, such as rootfs type, initrd size and location, system ram size, to be specified during compilation. Default configuration will set initrd at 0xA00000, size no larger than 4MB as rootfs, ttyS0 running at 115200 8-N-1 as console port, and 64MB system memory. So default boot option looks like:

```
Boot options --->
(root=/dev/ram0 console=ttyS0,115200n8 console=tty1 initrd=0xa00000,4000000 mem=64M)
```

If using NFS as root file system during development is required, boot option could be modified to:

```
console=ttyS0,115200n8 root=/dev/nfs rw nfsroot=<NFS server IP>:<directory
name> ip=<EV Board IP>:<NFS server IP>:<Gateway IP>:<Net mask> mem=64M
```

Besides above change, some kernel options also need to be enabled as well. They are:

```
File systems --->
Network File Systems --->
[*] NFS file system support
[*] Root file system on NFS
Networking --->
Networking options --->
```

[*] IP: kernel level auto configuration

4.2.1.5 Networking

To enable TCP/IP network, please following the setting below:

Networking --->
[*] Networking support
Networking options --->
<*> Packet socket
<*> Unix domain sockets
[*] TCP/IP networking

4.2.2 Device driver configuration

This section describes how to enable driver for each interfaces. Some options only appear if the interface is enabled in system type configuration.

4.2.2.1 MAC interface support

Device Drivers --->
Network device support --->
[*] Network device support
<*> Dummy net driver support
Ethernet (10 or 100Mbit) --->
[*] Ethernet (10 or 100Mbit)
[*]Nuvoton NUC900 Ethernet driver

4.2.2.2 NAND Flash support

NAND driver source code is included in BSP. However, physical to logical mapping scheme and wear leveling algorithm (gnand.ko) is not GPL licensed, and only released as kernel. So NANND driver also needs to be built as a kernel module. To use onboard NAND flash, load gnand.ko (You can find it in /usr/gnand.ko in rootfs.gz), and then load nanddrv.ko

This driver supports both SLC and MLC NAND flash.

```
System Type --->
NUC900 Device Interface conflict pre-set
--->Select NAND or LCD or KPI support?
⊙NAND driver
◯LCD driver
◯KPI driver
```

```
Device Drivers --->
SCSI device support --->
[*] SCSI disk support
[M] Nuvoton NUC900 NAND driver
```

4.2.2.3 USB host support

Here list the configuration for USBH interface to support mass storage device.

```
Device Drivers --->
USB support --->
[*] Support for Host-side USB
[*] EHCI HCD (USB 2.0) support
[*] OHCI HCD support
[*] USB Mass Storage support
SCSI device support --->
<*> SCSI disk support
```

4.2.2.4 USB device support

Here list the configuration for USBD mass storage support

```
Device Drivers --->
Character devices --->
[*] Nuvoton NUC900 USB Device Driver
```

4.2.2.5 PS/2 keyboard & mouse support

```
Device Drivers --->
Input device support --->
[*] Keyboards --->
[*] AT keyboard support
[*] Mouse --->
[*] PS/2 mouse
[*] Nuvoton NUC900 PS2 Mouse Support
Hardware I/O ports --->
[*] Nuvoton NUC900 PS2 Keyboard Support
```

4.2.2.6 Audio support

```
Device Drivers --->
Sound --->
[*] Sound card support
Open Sound System --->
[*] Open Sound System (DEPRECATED)
[*] Nuvoton NUC900 Audio Support
```

4.2.2.7 Video support

```
Device Drivers --->
Graphics support --->
<*> Support for frame buffer devices
<*> NUC900 LCD frame buffer support
[*] AUO A035QN02 320X240 LCD support
Console display driver support --->
[*] Framebuffer Console support
Logo configuration --->
[*] Bootup logo
[*] Standard 224-color Linux logo
```

4.2.2.8 RTC support

```
Device Drivers --->
[*] RTC class
--- RTC interfaces
[*] sysfs
[*] proc
[*] dev
--- RTC drivers
[*] Nuvoton NUC900
```

4.2.2.9 SD card support

```
Device Drivers --->
SCSI device support --->
[*] legacy /proc/scsi/ support
[*] SCSI disk support
[*] Nuvoton NUC900 SD Card support
[*] Enable SD0 port
```

4.2.2.10 Touch Screen support

The touch screen resolution should be the same with panel resolution.

```
Device Drivers --->
Input device support --->
[*] Touchscreen interface
(640) Horizontal screen resolution
(480) Vertical screen resolution
[*] Event interface
[*] Touchscreens --->
[ ] NUC910 touchscreens
[*] TSC2007 touchscreens
```

4.2.2.11 Memory Stick support

SD and MS interface shares the same pins. If SD0 and MS0 or SD1 and MS1 are enabled at the same time, kernel compilation will failed.

```
Device Drivers --->
SCSI device support --->
[*] legacy /proc/scsi/ support
[*] SCSI disk support
[*] Nuvoton NUC900 SD Card support
[*] Enable SD0 port
```

4.2.3 File system support

4.2.3.1 EXT3 support

```
File systems --->
[*] Ext3 journaling file system support
```

4.2.3.2 EXT2 support

```
File systems --->
[*] Second extended fs support
```

4.2.3.3 FAT-based file systems(MS-DOS, VFAT) support

```
File systems --->
DOS/FAT/NT Filesystems --->
[*] MSDOS fs support
[*] VFAT (Windows-95) fs support
(437) Default codepage for FAT
(iso8859-1) Default iocharset for FAT
Partition Types --->
[*] Advanced partition selection
[*] PC BIOS (MSDOS partition tables) support
Native Language Support --->
(iso8859-1) Default NLS Option
[*] Codepage 437 (United States, Canada)
[*] NLS ISO 8859-1 (Latin 1; Western European Languages)
```

4.2.3.4 /proc file system support

File systems --->
Pseudo filesystems --->
[*] /proc file system support

4.2.3.5 NFS support

File systems --->
Network File Systems --->
[*] NFS file system support
[*] Provide NFSv3 client support

5. SAMPLE APPLICATIONS

There are some sample applications in the applications/ directory. Content of each directory listed in the following table.

Directory	Description
bash-3.2/	Bash shell source code. The build sequence listed below: 1. ./configure --enable-static-link --host=arm-linux 2. make 3. arm-linux-strip bash
benchmark/bonnie	A benchmark which measures the performance of file system operations.
benchmark/dhrystone	A processor performance benchmark.
benchmark/netperf-2.4.4	A network performance benchmark.
boa-0.94-13	An open source browser. To build boa, follow the steps: 1. ./configure –host=arm-linux 2. make LDFLAGS=”-static” Execute with following steps: 1.Copy examples/etc/boa to NUC900/rootfs/etc 2.Copy src/boa to NUC900/rootfs/bin/ 3.Launch it with "boa -c /etc/boa"
busybox-1.9.1/	Busybox source code. The build sequence listed below: 1. make menuconfig 2. Load NUC900 as configure file 3. make
demos/audio/	Audio sample application. *
demos/cpp/	C++ sample application. *
demos/hotplug/	Storage device auto mount sample code
demos/i2c/	I2C sample application. *
demos/keypad/	Keypad sample application. *
demos/lcm/	LCD sample application. *
demos/mass/	Sample mass storage application. The usage is: mass <device_name>. Please do not mount <device_name> while executing this application, otherwise the result is unpredictable. *
demos/rtc/	RTC sample application.

Directory	Description
demos/smartcard/	Smartcard sample application. *
demos/thread/	Thread sample applications. *
demos/touchscreen	Touch screen sample application *
demos/uart	UART sample code. Usage: uart_demo <port num> *
demos/usi/	USI sample application. *
gdb-6.4	arm-linux-gdb and gdbserver source code To build arm-linux-gdb, enter gdb-6.4, type "./configure --host=i686-pc-linux-gnu --target=arm-linux;make;make install". To build gdbserver, enter gdb-6.4/gdb/gdbserver, type "make".
microwindows-0.91/	Microwindows source code. The build sequence listed below: 1. cd src 2. Edit config to choose mouse device. Set TSLIBMOUSE to "Y" to enable touch screen, NUC900MOUSE to "Y" to enable mouse. 3. Edit engine/devmouse.c, comment out "#define TSLIBMOUSE" if not using touch screen. 4. make The binary codes could be found in bin/ To use microwindows support touch screen, some environment variable most be set: export TSLIB_CONFFILE="/usr/gui/tslib/ts.conf" export TSLIB_PLUGINDIR="/usr/gui/tslib" export TSLIB_TSDEVICE="/dev/input/event0"
mtd/	MTD related utilities. The build sequence listed below: 1. cd util 2. make

Directory	Description
proftpd-1.3.1	GPL licensed FTP server. To build the application, enter proftpd-1.3.1/ directory and type "./configure –host=arm-linux; make LDFLAGS="-all-static -L./lib"", the generated binary file is named proftpd in the same directory. To execute proftpd, copy proftpd-1.3.1/nuvoton-ftpd/etc/ftpd to rootfs/var and proftpd into rootfs as well. Modify following files/directories according to you environment: rootfs/etc/hosts rootfs/etc/hostname rootfs/etc/ftpd/etc/proftpd.conf rootfsproftpd-1.3.1/nuvoton-ftpd/etc/ftpd Launch FTP server with command: ./proftpd &
qtopia-coreopensource-src-4.3.3	Open source Qtopia. Please note, you need handle license issue your self. The build steps list below: 1. ./configure -xplatform qws/linux-arm-g++ -prefix /usr/gui -static -embedded arm -little-endian -qt-mouse-tslib -L $PWD/lib -I $PWD/include/tslib -qt-libjpeg -qt-libpng -qt-freetype –no-openssl –L $PWD/plugins/imageformats 2. Input "yes" and "16" during configuration. 3. make
wireless_tools.29	Wireless tools, including iwconfig, iwlist, iwpriv… etc.
wpa_supplicant-0.5.8	WPA Supplicant with support for WPA and WPA2.

*. The execution result will be incorrect if the driver is not enabled in kernel configuration and/or jumper/ switch setting on RSC-W910 setting is inconsistent with kernel configuration.

5.1 Generate root file system image

Kernel will search the initrd as root file system locates at 0xA00000 in SDRAM according to the default boot command. The Makefiles under demos/ directory will copy the generated executable files into root file system and generate the binary file system file to image/rootfs.gz. But for other applications, user has to copy the executable files into root file system and generate the binary file system with command:

```
$ genromfs –d <source directory> -f <output file name>
$ gzip <output file name>
```

Linux kernel will decompress this image and mount it as root file system. The default shell (bash) only search for

/bin to find execution files. To execute the application outside bin/ directory in root file system, use absolute path to execute it, or enter its directory and type ./<file name> to execute it, or create a soft link in /bin directory.

If the size of newly created image is large then the size specified in boot command, it is necessary to enlarge the initrd size in boot command and rebuild the kernel.

User can also use bootloader to pass TAG list of initrd location and size to kernel. To enable kernel parsing the TAG list, please set the boot_param value in arch/arm/mach-NUC900/mach-NUC900.c to 0x100 and rebuild Linux kernel. Please refer to Bootloader user's manual for how to set the TAG list via ft, fu, fx, and tag command.

Linux also supports using initramfs as root file system, some modification are necessary for doing this.(gawk package is required, please install it if it doesn't exist on host PC)

1. Modify the filename of first execution binary from "linuxrc" to "init". Or modify boot command, remove initrd setting and add: "rdinit=/sbin/init"
2. Specify the file system directory file at menuconfig General Setup -> initramfs source files
3. Rebuild the kernel
4. If the kernel cannot boot up, please modify the default kernel size at create_page_table() function in /arch/arm/kernel/head.S.

5.2 Cross compilation makefile

Sometimes a project requires porting an application to ARM platform. If the application's Makefile doesn't support cross compilation options, the modification of Makefile is necessary. The Makefile used for cross compiling could be alike with the original one, only part of it needs to be modified.

- The prefix of tool chain must be set. For example, the original Makefile use gcc for compiling, the new Makefile use arm-linux-gcc for cross compiling.
- The path of library and include files need to be set. The cross compiler doesn't use the glibc or other library using in x86 system.
- -static option must be given. Current tool chain doesn't support dynamic linking

Here is a sample Makefile

```
.SUFFIXES : .x .o .c .s
ROOT = /usr/local/arm_linux_4.2/
LIB = $(ROOT)/lib/gcc/arm-linux/4.2.1
LIB1 =$(ROOT)/arm-linux/lib
INCSYS:=$(ROOT)/arm-linux/sys-include
```


```
INC :=$(ROOT)/arm-linux/include
CC=arm-linux-gcc -O2 -I$(INC) -I$(INCSYS) –static
WEC_LDFLAGS=-L$(LIB) -L$(LIB1)
STRIP=arm-linux-strip
TARGET = hello
SRCS := hello.c
LIBS= -lc -lgcc -lc
all:
$(CC) $(WEC_LDFLAGS) $(SRCS) -o $(TARGET) $(LIBS)
$(STRIP) $(TARGET)
clean:
rm -f *.o
rm -f $(TARGET)
rm -f *.gdb
```

Avalue Technology Inc. 7F, 228, Lian-cheng Road, Chung Ho City, Taipei, Taiwan

www.avalue.com.tw
sales@avalue.com.tw
Tel: +886-2-8226-2345
Fax: +886-2-8226-2777

Avalue USA
sales@avalue-usa.com
Tel: +1-732-578-0200
Fax: +1-732-578-0250

BCM Advanced Research
BCMsales@bcmcom.com
Tel: +1-949-470-1888
Fax: +1-949-470-0971

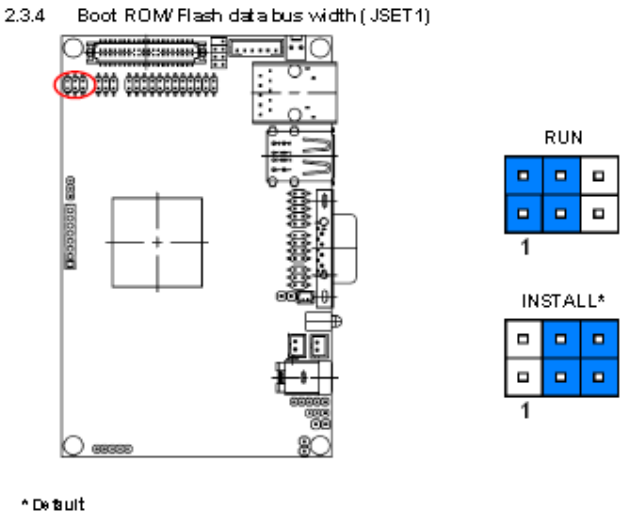
Avalue China
sales.china@avalue.com.cn
Tel: +86-21-5169-3609
Fax: +86-21-5445-3266

6. FLASH BOOTLOADER, KERNEL AND ROOTFS

To program RSC-W910 in Linux, you will need three files: NandBoot.img, 910kernel & rootfs.gz. Please follow steps below to program NAND flash of RSC-W910

6.1 Install Nuvoton W90P910 driver in your Host PC. (WinXP). You can find the driver in the path of "\RSC-W910\BSP\Linux\Utility\TurboWriter\driver" in driver CDROM.

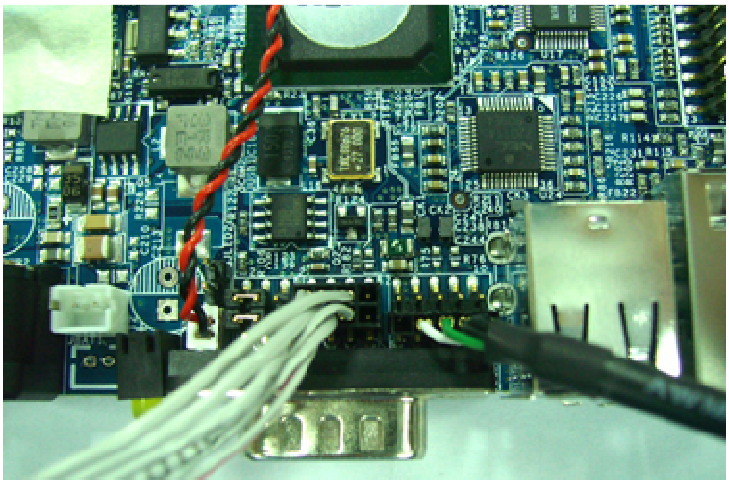
6.2 Set RSC-W910 jumper "JSET1" in INSTALL mode (Turbo Writer Mode).
JSET1 =>1-3,2-4: NAND Boot Mode
JSET1 =>3-5,4-6: Turbo Writer Mode



6.3 Connect USB Client cable & RS232 debug port cable on RSC-W910.

E1700100152R	USB Cable Dupont 5P/2mm-Mini B Type USB
E1706122000R	COM port cable for RSC-W910 JCOM2

Please connect E1700100152R on JUSB1 and E1706122000R on JCOM2.

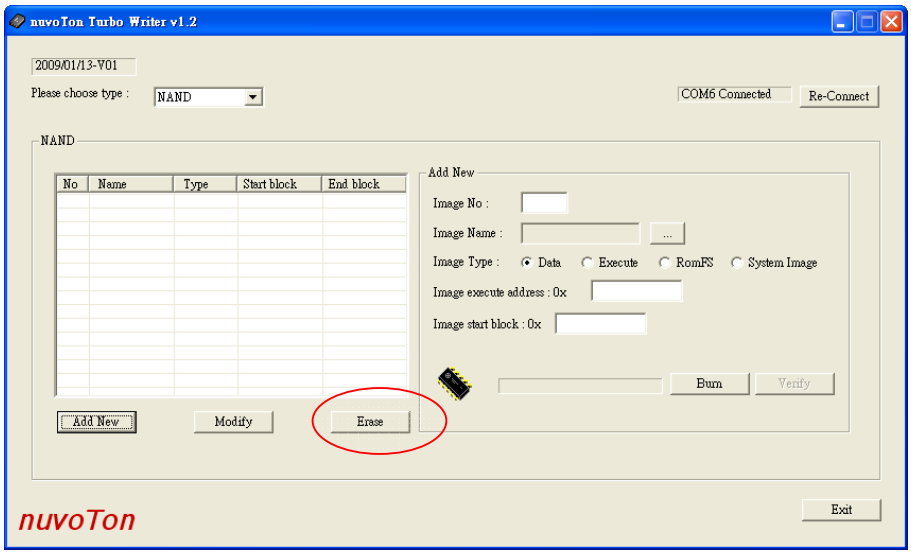


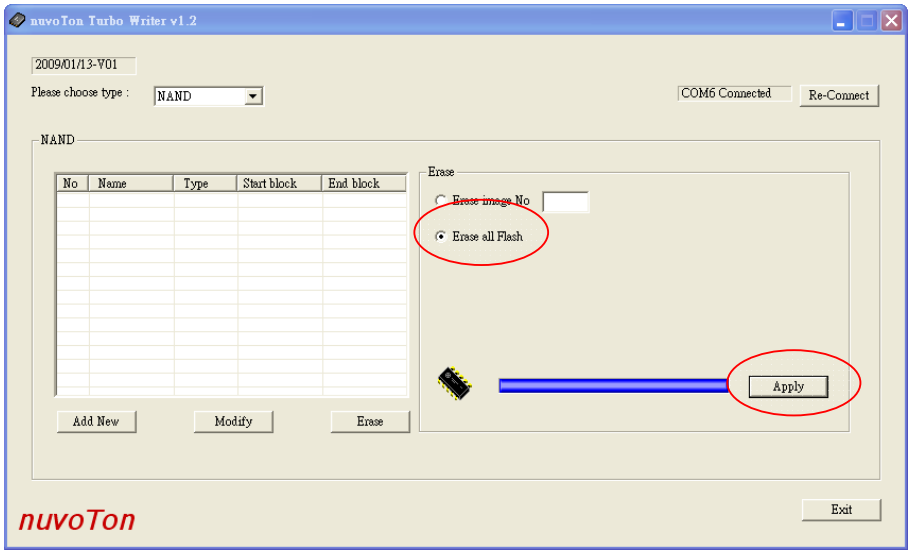
6.4 Power on RSC-W910 and connect it to USB interface of your Host PC.

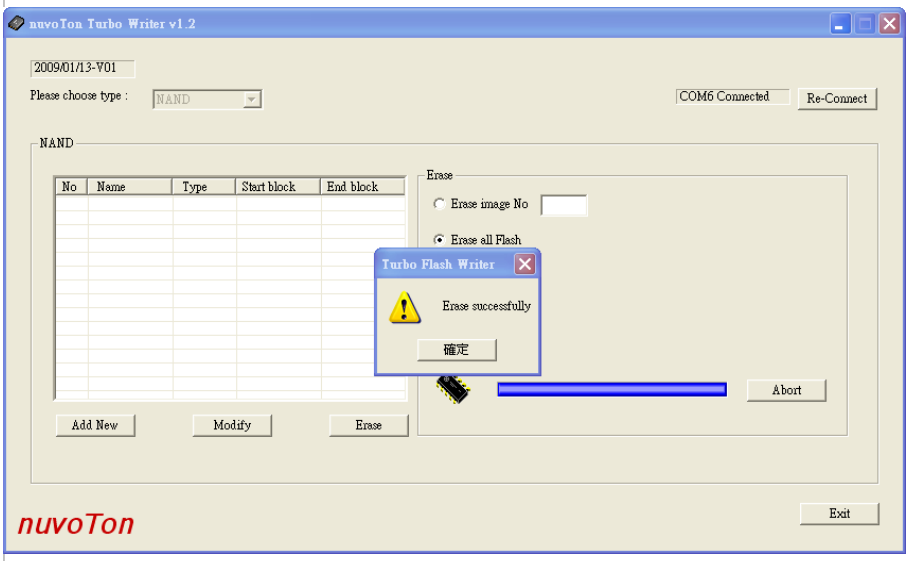
On device manager, it will recognize a new device as below.

6.5 Execute turbo writer. You can find it in the path "\RSC-W910\BSP\Linux\Utility\TurboWriter\utility" in driver CDROM.

6.6 Erase all the content in NAND flash.

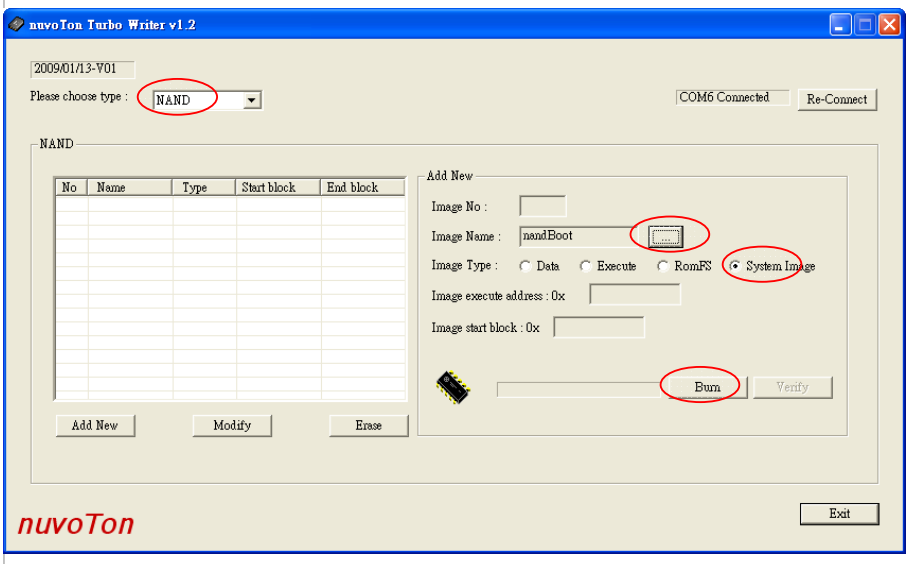






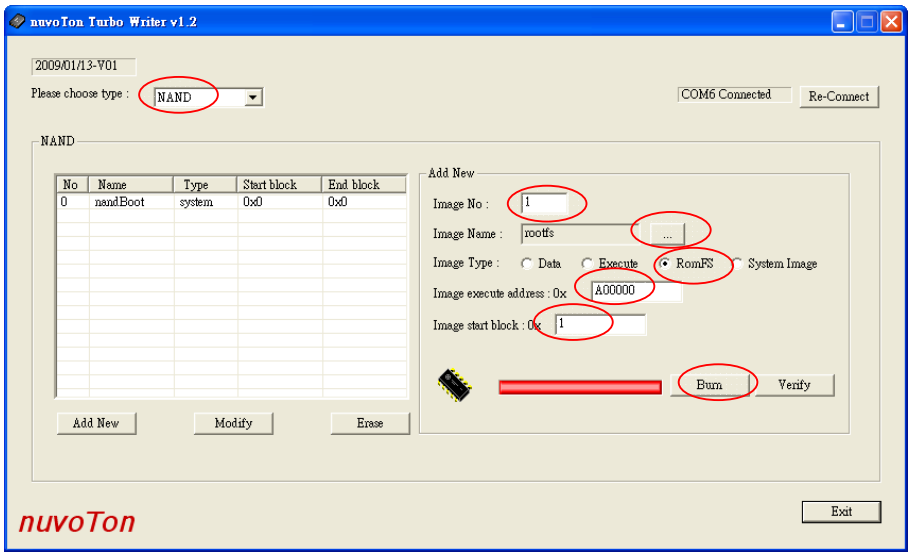
6.7 Flash nandboot.img

- Please choose type "NAND"
- Browse the file "nandboot.img"
- Set image type "System Image"
- Press "Burn" Button



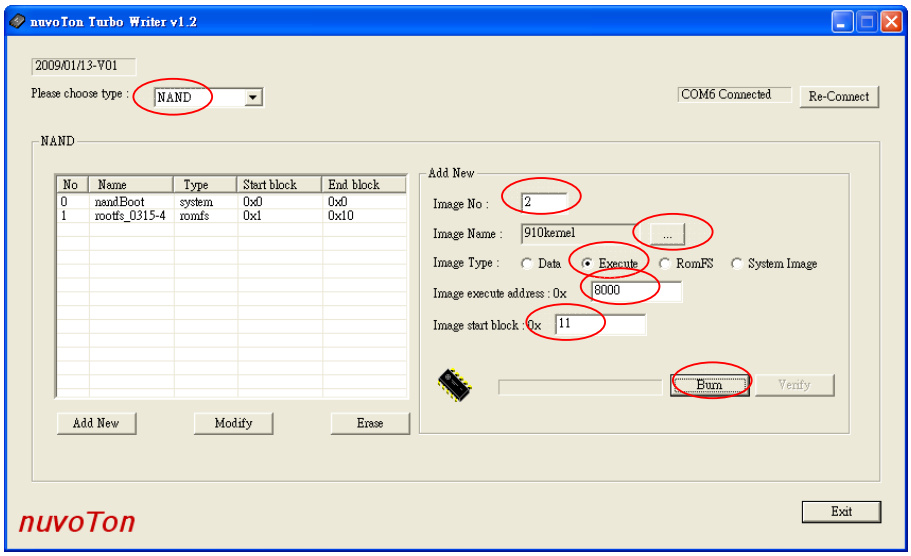
6.8 Flash rootfs.gz

- Please choose tye “NAND”
- Image No => 1
- Browse the file “rootfs.gz”
- Image type “RomFS”
- Image execute address :0xA00000
- Image start block : 0x1
- Press “Burn” Button



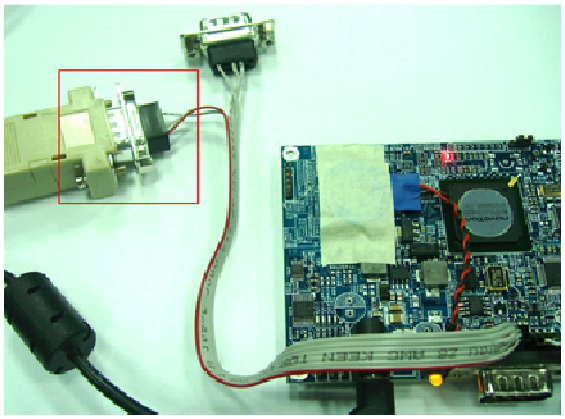
6.9 Flash 910kernel

- Please choose type “NAND”
- Image No => 2
- Browse the file “910kernel”
- Image Type : Execute
- Image execute address : 0x8000
- Image start block : 0x11
- Press “Burn” Button

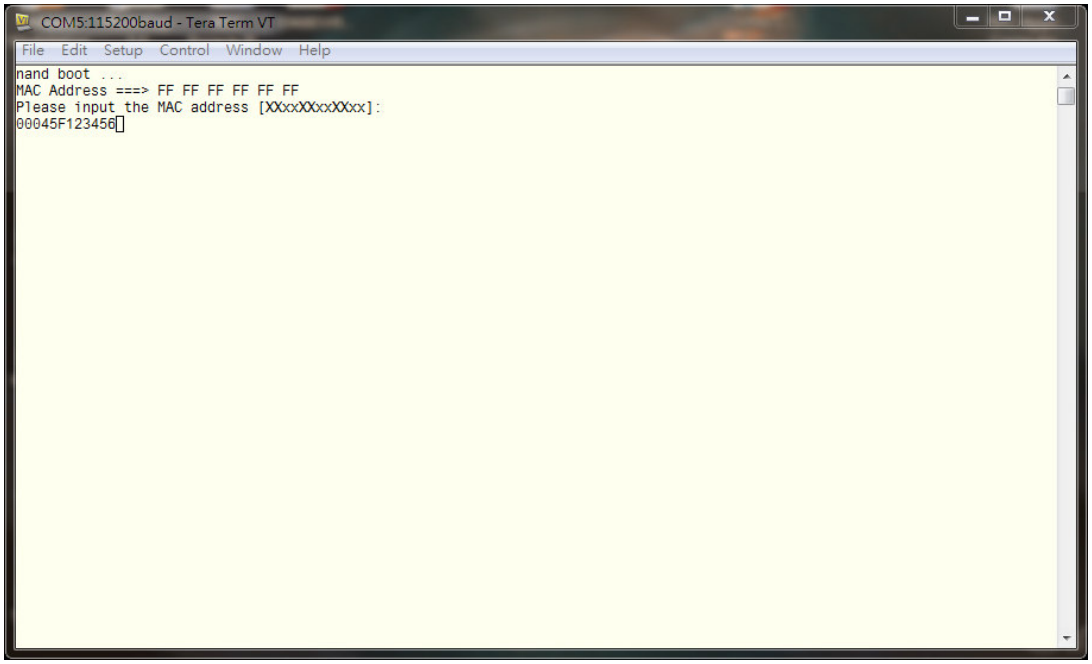


Then Power off RSC-W910 and set JSET1 in “NAND Boot Mode”

6.10 Connect JCOM2 cable (E1706122000R) with your Host PC by a RS232 Cross over cable (Avalue Part number : E1706181500R)



6.11 Run Windows Hyper Terminal and power on RSC-W910, press Space key and key-in MAC address.



Then you can boot into root file system w/o problem.