

Deployment Instructions

EdgeLink User Development Manual

Revision 1.0 Date: February 7, 2023

ADVANTECH

Enabling an Intelligent Planet

Revision History

Date	Version	Author	Reviewer	Description
Feb. 7,2023	1.0	Lili.Zheng		Initial Release

TABLE OF CONTENTS

1	INTRODUCTION.....	9
1.1	System Features.....	9
2	INTRODUCTION TO LINUX SYSTEM.....	9
2.1	Partition Introduction.....	9
2.2	Defintions of Directory	9
2.3	Supported System Services	10
3	DEVELOPMENT ENVIRONMENT SETUP	10
3.1	Cross-development Environment Version Selection	10
3.2	Predecessors of EdgeLink 2.6.0	11
3.2.1	Software Downloading	11
3.2.2	TISDK Installation.....	12
3.2.3	Installing TISDK’s Auxiliary Software.....	13
3.2.4	Update libssl	13
3.3	EdgeLink 2.6.0 and Its Successors.....	14
3.3.1	Software Downloading	14
3.3.2	TISDK Installation.....	14
3.3.3	Installing TISDK’s Auxiliary Software.....	14
3.3.4	Update libssl	15
3.4	EdgeLink 2.6.0 and Its Successors (WIN10 WSL).....	15
3.4.2	Install wsl and ubuntu 18.04 in WIN10 System	15
3.4.3	TISDK Installation.....	20
3.4.4	Update libssl	21
3.4.5	Notes on WSL.....	22
3.5	Install tisd on Non-ubuntu System.....	23
3.6	Other Common Software	23
3.6.1	MobaXterm	23

3.6.2	Putty	24
3.6.3	WinSCP	24
3.6.4	VirtualBox	24
3.6.5	Eclipse CDT.....	24
3.6.6	Eclipse ARM GCC Plug-In	24
4	SYSTEM SETUP	27
4.1	Make the System Partition Read Only [2.6.1 and Its Successors].....	27
4.1.1	Make the Partition Readable and Writable Temporarily	27
4.1.2	Make the Partition Readable and Writable Permanently.....	28
4.2	Serial Port	30
4.2.1	Test Whether the Serial Ports are Normal	30
4.2.2	minicom Command.....	30
4.2.3	microcosm Command.....	32
4.3	sramutil	34
4.3.1	Help.....	35
4.3.2	Execute Commands Using File Reading and Writing Method.....	35
4.4	RTC Clock	35
4.4.1	RTC Clock Command	35
4.4.2	Time Zone Configuration	36
4.5	Calibrate Time with ntp	36
4.5.1	ntp Client.....	36
4.5.2	ntp Server.....	38
4.5.3	Provide Time Services to Specified Network Segments Only	40
4.5.4	ntpd's Relevant Commands	41
4.6	Configure a Fixed Network Card IP	41
4.6.1	View the Information of the Current Network Card.....	41
4.6.2	Configure Temporary Network Card IP	42
4.6.3	Use DHCP to Obtain Temporary Network Card IP	42
4.6.4	Runtime and Permanent Configuration of a Fixed IP.....	43
4.6.5	Runtime and Permanent Configuration of a Dynamic IP	43
4.7	WIFI Configuration.....	44
4.7.1	Check Whether the Wireless Network Card has been Correctly Identified	44
4.7.2	Enable the Wireless Network Card.....	45

4.7.3	Scan Searchable Wireless Network	45
4.7.4	Modify Configuration Files (WPA2 Method)	47
4.7.5	Modify Configuration Files (Open Method)	47
4.7.6	Connect AP	48
4.7.7	Save SSID Password by Encryption	49
4.7.8	Configure a Fixed IP	49
4.7.9	Configure a Static IP	49
4.8	Configure WIFI as AP Mode	50
4.8.1	Configuration File	50
4.8.2	Enable the Program	50
4.8.3	Add Startup Programs	50
4.9	DHCP Service	51
4.9.1	Configuration File	51
4.9.2	Enable Services	51
4.10	Cellular Information Query [Supported by 2.6.1 and Its Successors]	52
4.11	Celluar Communication Configuration	54
4.11.1	pppd Dialing Process	54
4.11.2	Check the Communication Module	55
4.11.3	Dialing Command	56
4.11.4	Configuration Parameter Parsing	59
4.11.5	Parameter Parsing for connect Script	59
4.11.6	Parameter Parsing for Disconnect Script	61
4.11.7	Common Operators' connect Scripts	61
4.11.8	ZTE ME3760 Module Configuration	63
4.11.9	List of Common AT Commands for Debugging	64
4.12	4G Model Resetting	67
4.13	Dual sim Card Switching	68
4.14	AdvWrielessCheckd Instructions [Supported by OS 2.6.0]	68
4.14.1	Workflow	69
4.14.2	Usage	70
4.14.3	Description of Configuration File CellularDeviceInfo.acr [Supported by Successors of 2.7.0]	74
4.14.4	Description of Configuration File [SystemSetting.acr]	75
4.15	OpenVPN (Gateway as Client)	83

1) Use cert/key.....	83
2) Use Username/Password.....	84
3) Configure as auth-name	84
4) TLS-auth Communication.....	86
5) TAP and TUN Modes	86
6) Add Execution Script.....	86
4.16 OpenVPN (Gateway as the Server)	87
4.16.1 Intstall OpveVPN Server in Windows.....	87
4.16.2 Generate a Root Certificate	87
4.16.3 Generate cert/key for Server.....	88
4.16.4 Generate cert/key for Client	88
4.16.5 Use Cert/Key to Configure Server.....	88
4.16.6 Configure Client with Cert/Key	89
4.16.7 Configure as auth-name.....	90
4.16.8 TLS-auth Communication	92
4.16.9 TAP and TUN Modes.....	92
4.16.10 Add Executable Script.....	93
4.17 Routing Table Configuration.....	94
4.17.1 View the Current Touting Table.....	94
4.17.2 Add a Touting Table.....	94
4.17.3 Delete a Touting Table	94
4.17.4 Add a Gateway	94
4.18 Check USB Device.....	95
4.19 Enable the FTP Service	95
4.19.1 Modify Listening Ports.....	96
4.20 Enable telnet Service.....	96
4.20.1 Modify Listening Ports.....	97
4.21 Enable SSH Service.....	97
4.21.1 Modify Listening Ports.....	98
4.22 DNS Configuration	99
4.22.1 /etc/rc.local	99
4.22.2 Recreate a Soft Link	100
4.22.3 Remove a Soft Link.....	100

4.23	Firewall Configuration	100
4.23.1	View the Current Status of the Firewall	100
4.23.2	Set up a Whitelist.....	101
4.23.3	Enable the Firewall.....	101
4.23.4	Disable the Firewall.....	101
4.24	Web Server (lighttpd) Configuration.....	101
4.24.1	Configure Root Directory.....	101
4.24.2	Use https Secure Link.....	102
4.25	Ways to Configure Startup Programs	102
4.26	SNMP Configuration	103
4.26.1	SNMP Functions.....	103
4.26.2	Introduction to ADVANTECH-IO-COMMON-MIB	104
4.26.3	Create snmpv3 User.....	108
4.26.4	Accesses Local SNMP on ADAM5630	109
4.26.5	Obtain the Information of ADAM-5630's Local IO Module.....	112
4.26.6	Remote Access	113
4.27	Install a Driver to View lsmod	113
5	PROGRAM DEVELOPMENT	114
5.1	Onboard Resource Programming (BoardResource SDK).....	114
5.1.1	Watchdog.....	114
5.1.2	PLED	118
5.1.3	NODE_ID.....	120
5.1.4	DIO (Non-ADAM3600 Serials).....	122
5.2	IO Resource (ADAM3600 Serials).....	123
5.3	Serial Port Programming	127
5.3.1	Basic Steps.....	127
5.3.2	Parameter Configuration.....	127
5.3.3	Other Parameter Instructions	129
5.4	Network Programming.....	131
5.4.1	TCP Communication	131
5.4.2	UDP Communication.....	136

5.5 WebService Programming141

- 5.5.1 Lighttpd Configuration141
- 5.5.2 Code142
- 5.5.3 Results.....145
- 5.5.4 Precautions for Using the EDGELINK Platform145

5.6 SRAM Programming.....145

- 5.6.1 Basic Instructions of SRAM.....145

1 Introduction

1.1 System Features

EdgeLink is preinstalled with TI AM335x-based Linux operating system, with Kernel 4.9.69 deployed for successors of EdgeLink 2.6 and Kernel 3.12.10-RT15 for predecessors of EdgeLink 2.6. BusyBox v1.22.1. The kernel provides a simple and easy-to-use driver interface for the system's unique hardware to accelerate the development of application programs.

The operating system consists of three parts: Bootloader (Uboot), Linux kernel, and rootfs (busybox). UBoot is mainly used to start the kernel. It supports NFS mounting and can start NAND Flash. Linux kernel is the bottom layer of the operating system and it is responsible for actuating the hardware as well as providing core functions needed by the system. rootfs is a collection of system files.

You can run command `uname -r` to check the kernel version in the current device. Different cross-compilation environments are deployed based on the device versions. See *Section 3 Development Environment Setup*.

```
# uname -r  
4.9.69-g9ce43c71ae
```

2 Introduction to Linux System

2.1 Partition Introduction

Onboard TF (MicroSD) card has 4 partitions.

`/dev/mmcblk0p1` on `/media/mmcblk0p1` Store Uboot files and Linux Kernel files.

`/dev/mmcblk0p2` on/ System's root partition, which is read-only on successors of EDEGELINK 2.6.

`/dev/mmcblk0p3` on `/media/mmcblk0p3` A recovery partition, used for storing temporary upgraded files.

`/dev/mmcblk0p4` on `/home` System user partition.

2.2 Defintions of Directory

`/dev` Directory of device nodes

/media	Multimedia directory
/proc	System configuration directory
/sys	System configuration directory
/var	Temporary directory
/bin	Common user's command directory
/etc	Directory of configuration files
/lib	Directory of dynamic link library
/MNT	Same as the /media directory
/sbin	Root user command
/TMP	Temporary directory
/WWW	Web directory
/home	User data directory
/lost+found	Directory for temporarily storing deleted files
/opt	Configuration directory
/SRV	cgi command directory
/usr	Common user directory

2.3 Supported System Services

sshd
ftpd
httpd

3 Development Environment Setup

3.1 Cross-development Environment Version Selection

Before installing the device, please make sure that different versions of EdgeLink adopt different cross-compilation environments.

```
# uname -r
4.9.69-g9ce43c71ae
```

Kernel 4.9.69 is deployed for successors of EdgeLink 2.6. Please refer to the installation methods in *Section 3.3 EdgeLink 2.6.0 and Its Successors* or *Section 3.4 EdgeLink 2.6.0 and its Successors (WIN10 WSL)*.

```
#uname -r
```

```
3.12.10-rt15
```

Kernel 3.12.10 is deployed for predecessors of EdgeLink 2.6. Please install EdgeLink 2.6.0 based on the installation method *in Section 3.2 Predecessors of EdgeLink 2.6.0*. As the cross-compilation environment for predecessors of EdgeLink 2.6.0 is a 32-bit cross-compilation toolchain, please install the 32-bit application compatibility package to install them on X86_64-bit system.

If Ubuntu or a VM is installed, please refer to *Sections 3.2* and *Section 3.3*.

If Windows 10 is usually used, and the user does not want to install a VMS, see *Section 3.4 Installation Method*.

3.2 Predecessors of EdgeLink 2.6.0

3.2.1 Software Downloading

3.2.1.1 Ubuntu 12.04 32-bit

Download link:

<http://old-releases.ubuntu.com/releases/precise/ubuntu-12.04.4-desktop-i386.iso>

3.2.1.2 TISDK

Download link:

http://software-dl.ti.com/sitara_linux/esd/AM335xSDK/07_00_00_00/exports/ti-sdk-am335x-evm-07.00.00.00-Linux-x86-Install.bin

3.2.1.3 Install TISDK on 64-bit Ubuntu12.04

As TISDK is a 32-bit program, please download and install 32-bit compatibility package before installing TISDK on a 64-bit system.

http://processors.wiki.ti.com/index.php/Sitara_Linux_SDK_64_Bit_Ubuntu_Support

```
adv@adv-desktop:~ $sudo apt-get install libc6:i386
```

```
adv@adv-desktop:~ $sudo apt-get install libx11-6:i386 libasound2:i386 libatk1.0-0:i386  
libcairo2:i386 libcups2:i386 libdbus-glib-1-2:i386 libgconf-2-4:i386 libgdk-pixbuf2.0-0:i386 libgtk-3-  
0:i386 libice6:i386 libncurses5:i386 libsm6:i386 liborbit2:i386 libudev1:i386 libusb-0.1-4:i386  
libstdc++6:i386 libxt6:i386 libxtst6:i386 libgnomeui-0:i386 libusb-1.0-0-dev:i386 libcanberra-gtk-
```

module:i386 gtk2-engines-murrine:i386

Compile 32-bit programs on Uno platform

```
adv@adv-desktop:~$ apt install build-essential libc6:i386 libstdc++6:i386 gcc-multilib g++-multilib python doxygen graphviz fp-utils-3.0.4 u-boot-tools zlib1g-dev:i386 cmake zip libssl-dev:i386 libcurl4-openssl-dev:i386 libxml2-dev:i386 libsqlite3-dev:i386 libmosquitto-dev:i386 unixodbc-dev:i386 libfcgi-dev:i386 libcap-dev:i386 uuid-dev:i386
```

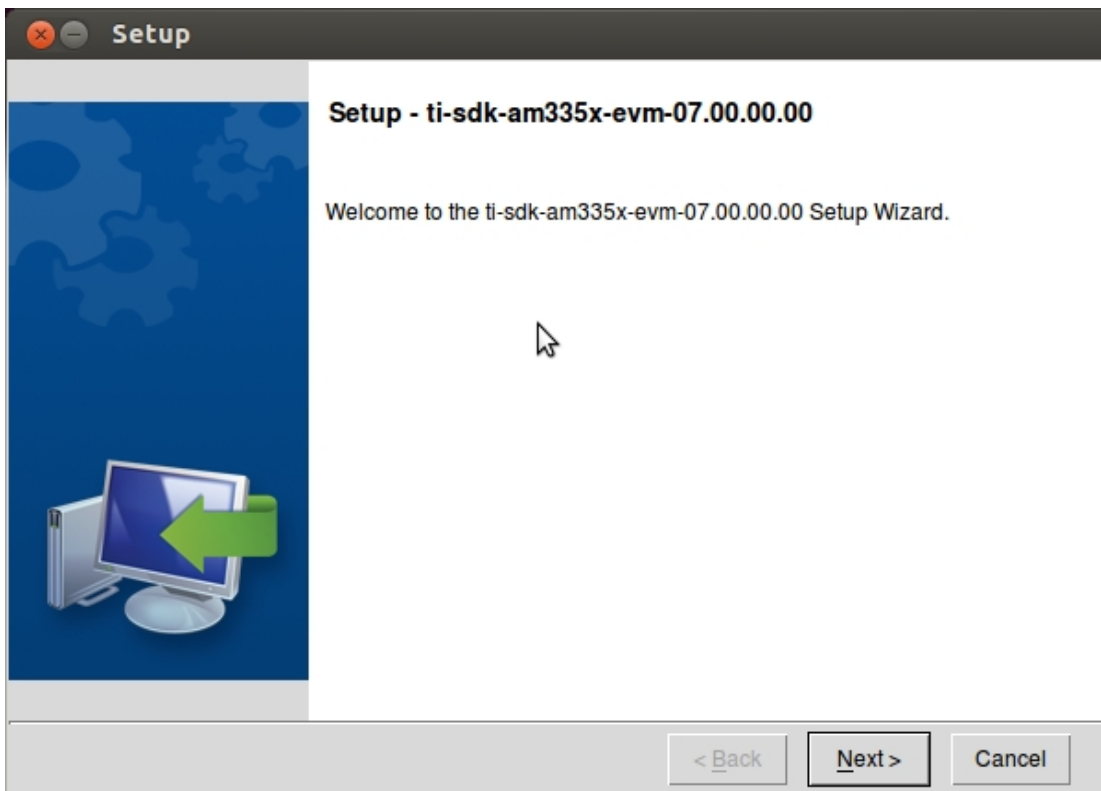
```
sudo apt-get install lib32ncurses5 lib32z1
```

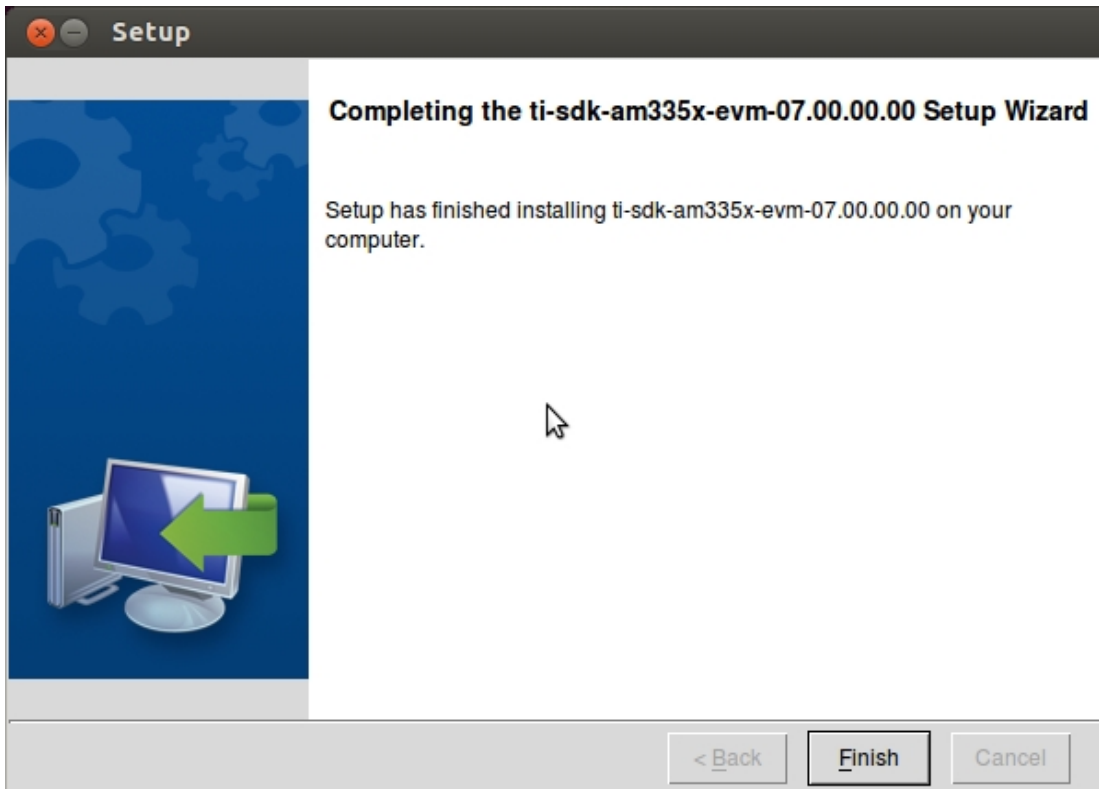
3.2.2 TISDK Installation

Add executable permissions to TISDK file and execute it.

```
adv@adv-desktop:~/Desktop$ chmod a+x ti-sdk-am335x-evm-07.00.00.00-Linux-x86-Install.bin
```

```
adv@adv-desktop:~/Desktop$ sudo ./ti-sdk-am335x-evm-07.00.00.00-Linux-x86-Install.bin
```





```
adv@adv-desktop:~/ $sudo vi /etc/profile Add PATH variable at the end of the file
export PATH=$PATH:/opt/ti-sdk-am335x-evm-07.00.00.00/linux-devkit/sysroots/i686-arago-linux/usr/bin
```

```
adv@adv-desktop:~/ $source /etc/profile
```

3.2.3 Installing TISDK's Auxiliary Software

TISDK provides script for automatic installation, configuration and operation of auxiliary software such as VSFTP and NFS. The script currently only works on Ubuntu 12.04. To execute the script, please make sure that the system is connected to the network before the installation. The process requires the installation of relevant software packages using apt-get.

```
adv@adv-desktop:~/ $ sudo /opt/ti-sdk-am335x-evm-07.00.00.00/setup.sh
Please complete the installation as instructed.
```

3.2.4 Update libssl

```
$tar -zxvf libssl_path_20181205.tar.gz
$cd libssl/
```

```
$sudo ./install_libssl.sh 1.0.0 # Install the predecessors of libssl 2.1.1 (including this
version), which is the default cross-compilation environment and does not need upgrading.
$sudo ./install_libssl.sh 1.1 # Install the successors of libssl 2.1.1
```

3.3 EdgeLink 2.6.0 and Its Successors

3.3.1 Software Downloading

3.3.1.1 64-Bit Ubuntu 18.04

Download link:

<http://old-releases.ubuntu.com/releases/bionic/ubuntu-18.04.2-desktop-amd64.iso>

3.3.1.2 TISDK

Link of download page:

http://software-dl.ti.com/processor-sdk-linux-rt/esd/AM335X/04_03_00_05/index_FDS.html

Download link:

http://software-dl.ti.com/processor-sdk-linux-rt/esd/AM335X/04_03_00_05/exports/ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05-Linux-x86-Install.bin

3.3.2 TISDK Installation

Add executable permissions to TISDK file and execute it.

```
adv@adv-desktop:~/$ chmod a+x ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05-Linux-x86-Install.bin
adv@adv-desktop:~/$ sudo ./ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05-Linux-x86-Install.bin
adv@adv-desktop:~/$sudo vi /etc/profile Add variable PATH at the end of the file
export PATH=$PATH:/opt/ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/
```

```
adv@adv-desktop:~/$source /etc/profile
```

3.3.3 Installing TISDK's Auxiliary Software

TISDK provides script for automatic installation, configuration and operation of auxiliary software such as VSFTP and NFS. The script currently only works on Ubuntu 12.04. To execute the script, please make sure that the system is connected to the network before the installation. The process requires the installation of relevant software packages using apt-

get.

```
adv@adv-desktop:~/ $ sudo /opt/ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05/setup.sh
```

Please complete the installation as instructed.

3.3.4 Update libssl

```
$tar -zxvf libssl1.1.0g-20200419.tar.gz
```

```
$cd libssl1.1.0g-20200419
```

```
$sudo ./install_libssl_1.1.sh
```

3.4 EdgeLink 2.6.0 and Its Successors (WIN10 WSL)

Windows Subsystem for Linux (WSL) is a compatibility layer capable of executing native Linux binary executables (ELF format) on Windows 10. It was co-developed by Microsoft and Canonical, with a view to downloading pure Ubuntu images to users' local computers and decompress them in situ, and enabling the tools and utilities within the images to run natively on the subsystem.

3.4.1.1 Software Downloading

3.4.1.2 TISDK

Link of the download page:

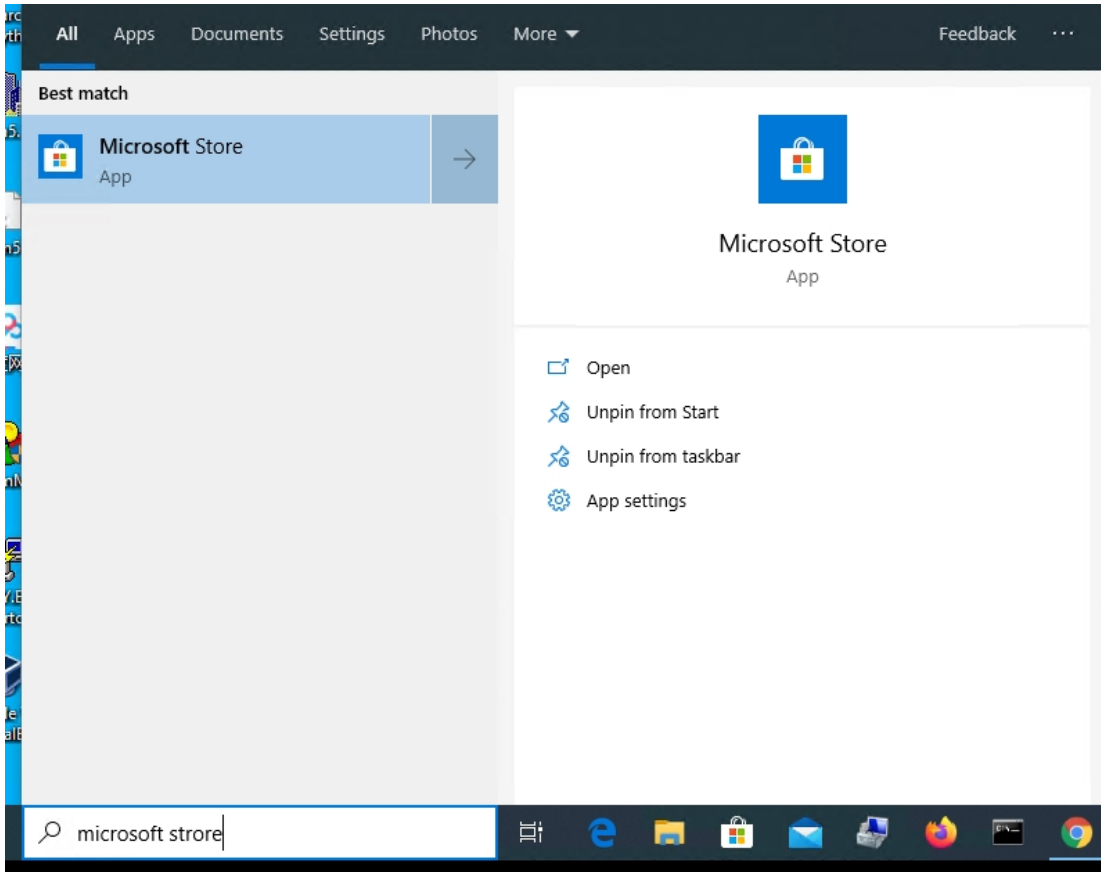
http://software-dl.ti.com/processor-sdk-linux-rt/esd/AM335X/04_03_00_05/index_FDS.html

Download link:

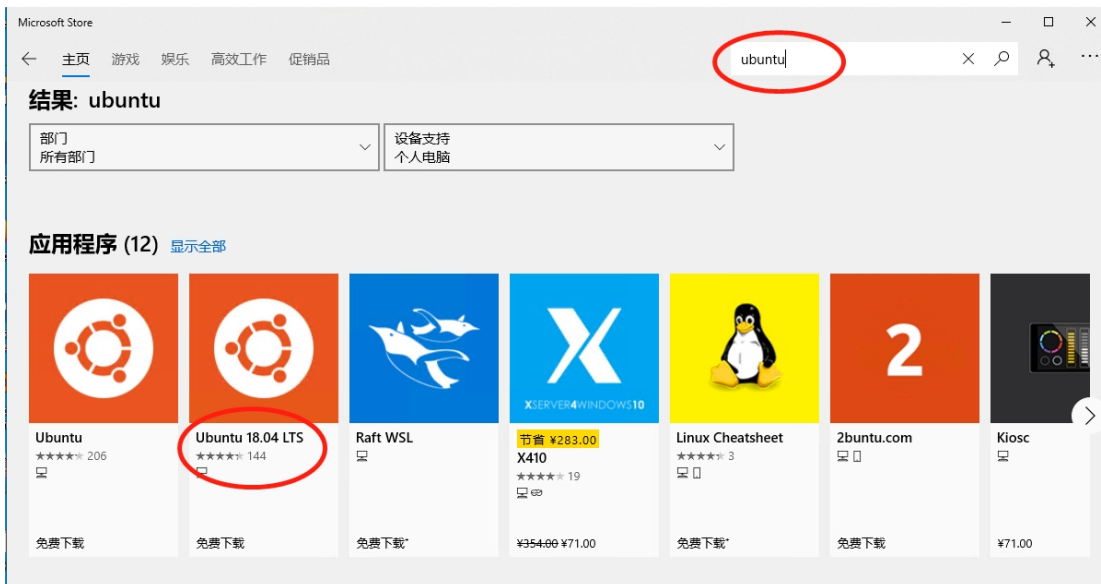
http://software-dl.ti.com/processor-sdk-linux-rt/esd/AM335X/04_03_00_05/exports/ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05-Linux-x86-Install.bin

3.4.2 Install wsl and ubuntu 18.04 in WIN10 System

1. Enter "Microsoft store" in the command line to open the Microsoft Store.

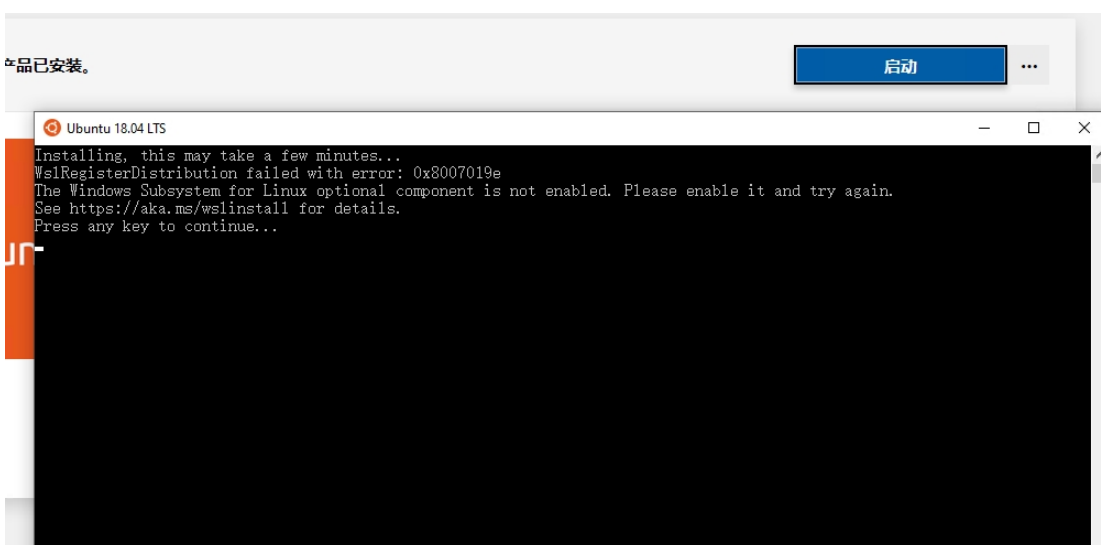


2. Search for "Ubuntu", and select "Ubuntu 18.04 LTS" from the drop-down list to install it.





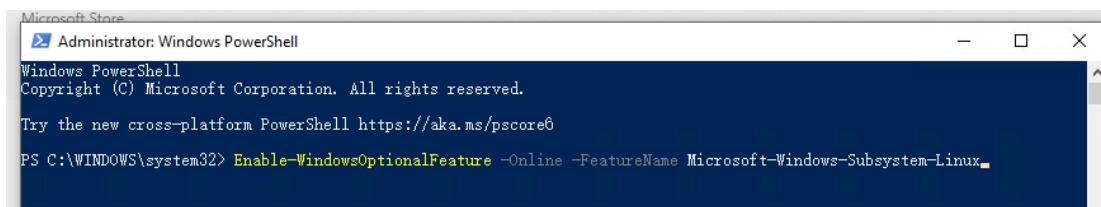
3. Start ubuntu1804.



4. Please enable wsl when you install it for the first time.

Run powershell with administrator privileges and execute the following commands:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

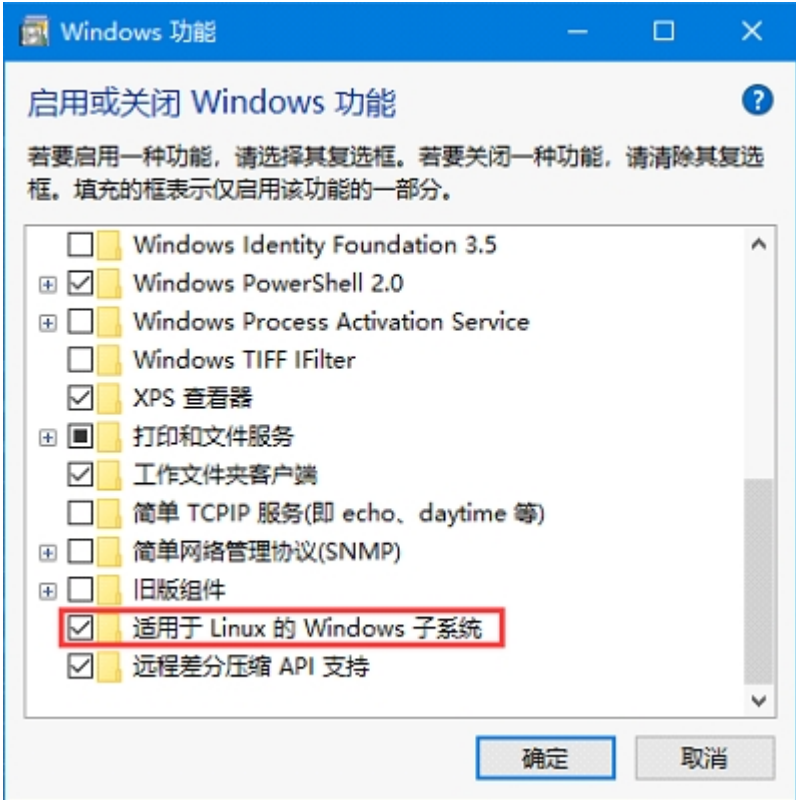
Enable-WindowsOptionalFeature: Microsoft-Windows-Subsystem-Linux
Running
[ooooooooooooo]
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

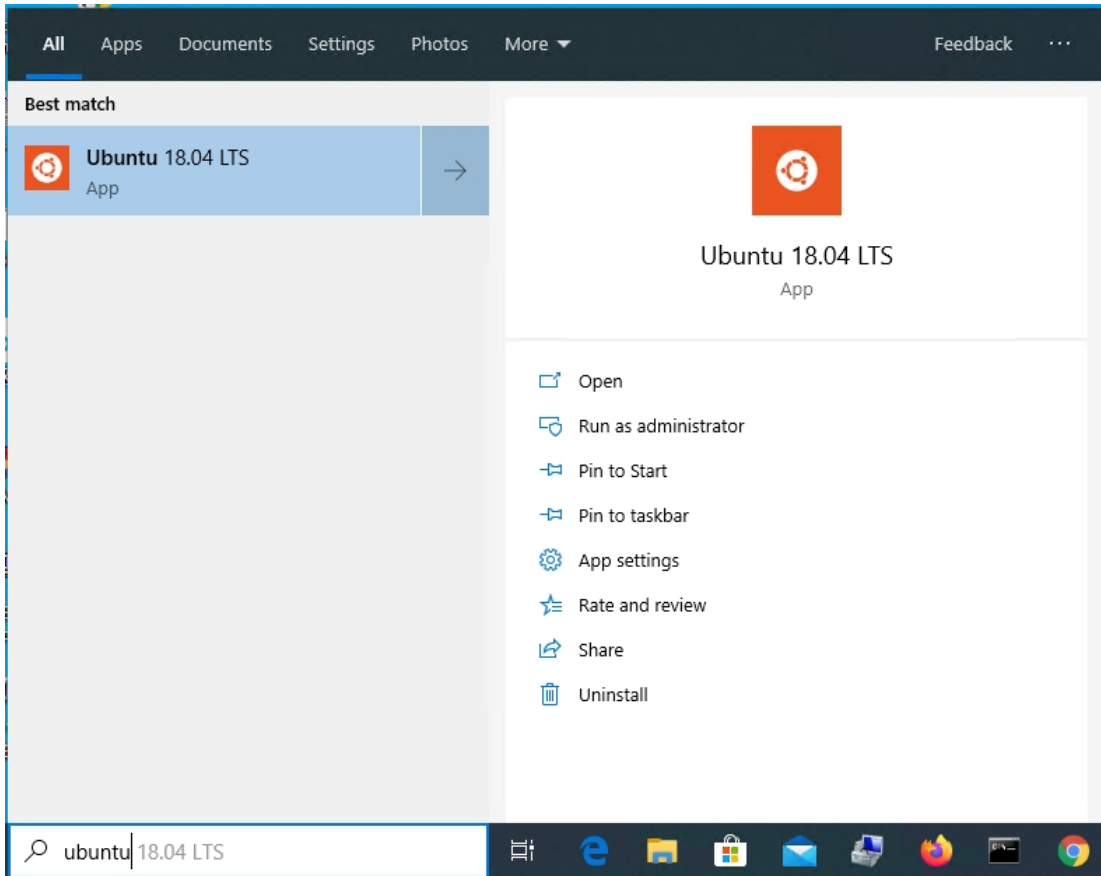
PS C:\WINDOWS\system32> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
Do you want to restart the computer to complete this operation now?
[Y] Yes [N] No [?] Help (default is 'Y'):
```

Or: Control Panel -> Programs and Features -> Turn on or off Windows features -> tick Windows Subsystem applicable for Linux.



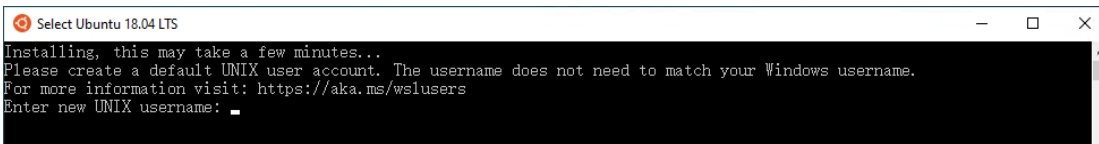
After the configuration, please restart Windows for it to take effect.

- 5. After you enter "Ubuntu" in the command line, the system will prompt you to enter the command to connect.





6. Set the user name and password at your first use.



3.4.3 TISDK Installation

Install TISDK in the cross toolchain directory in /mnt/d/ in disk D in the wsl system.

```
adv@DESKTOP-QGUL382:~$ cd /mnt/d/vm/
```

```
adv@DESKTOP-QGUL382:/mnt/d/vm$ sudo ./ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05-Linux-x86-Install.bin
```

Welcome to the ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05 Setup Wizard.

Linux Environment Support Message

It is highly recommend that the user, or especially new users, install this SDK into a Ubuntu 14.04 or Ubuntu 16.04 distribution environment. This is the environment that was used to develop and test this SDK.

Please note that this suggestion does not prevent the SDK from installing on other Linux Distributions.

Press [Enter] to continue:

....
.....

Please wait while Setup installs

ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05 on your computer.

Installing

0% _____ 50% _____ 100%

#####

As the installation will take a long time, you need to wait for about 10-30 minutes based on the machine performance.

\$ cd

\$ vi .bashrc

Add the below information at the end of the file:

export PATH=\$PATH:/opt/ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/

\$source .bashrc

3.4.4 Update libssl

\$tar -zxvf llibssl1.1.0g-20200419.tar.gz

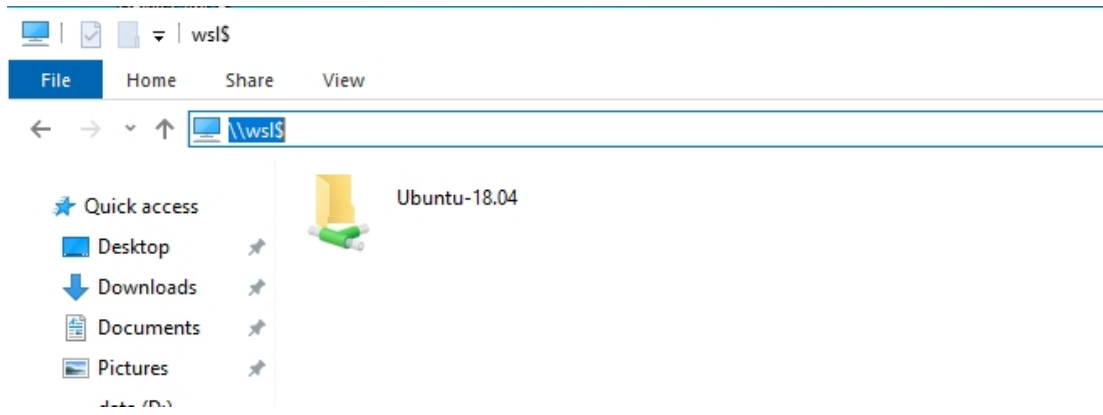
\$cd llibssl1.1.0g-20200419

\$sudo ./install_libssl_1.1.sh

3.4.5 Notes on WSL

3.4.5.1 Access the WSL System Directory

Enter `\\wsl$` in the file directory to access its file system.



3.4.5.2 Access the Hard Disk of the Windows System

The access address of wsl is `/mnt/d/`.

3.4.5.3 Modify the Background Color of WSL Directory

Change the background color to dark green as below:

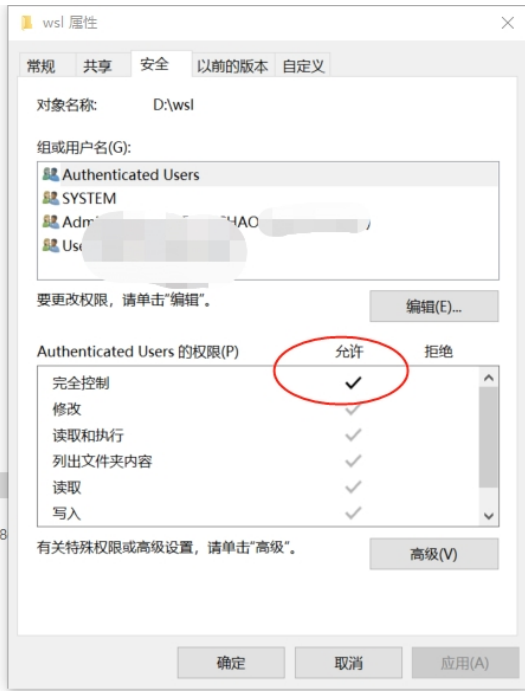
```
$dircolors -p > ~/.dircolors && sed -i "s/34;42/34;49/g" ~/.dircolors
```

After the configuration, restart WSL.

3.4.5.4 Change the Install Directory to D Drive

Create a new directory such as WSL on the D drive.

Right click -> Properties -> Security -> Edit -> Set Full Control



Create a directory and connect it to WSL's installation location.

```
mklink /j C:\Users\xxxxxx\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu18.04onWindows_79rhkp1fndgsc D:\wsl\
```

Wherein, C:\Users\xxxxxx is the actual username, e.g. C:\Users\UserName. Then install WSL.

3.5 Install tisdck on Non-ubuntu System

As the system may prompt that the header file cannot be found after tisdck is installed in systems such as centos, please make sure that python3 has already been installed in the system.

```
#yum install python3
```

If an error occurs in the cross-compilation environment, you can view the installation log to find the cause.

```
#vi /tm/bitrock_installer.log
```

3.6 Other Common Software

3.6.1 MobaXterm

MobaXterm supports SSH, FTP, serial port, VNC, X server, etc. It also supports tabs and the tab switching is convenient.

Free version download link:

<https://mobaxterm.mobatek.net/download-home-edition.html>

3.6.2 Putty

PuTTY is a Telnet, SSH, rlogin and pure TCP and serial interface connection software.

Download link:

<http://www.putty.be/latest.html>

3.6.3 WinSCP

WinSCP is an open source graphical SFTP client that uses SSH in Windows. It supports SCP protocol and its main function is to copy files between local and remote computers.

<https://winscp.net/eng/download.php>

3.6.4 VirtualBox

Windows users install virtual machine software.

Download link:

<https://www.virtualbox.org/>

3.6.5 Eclipse CDT

Download link:

<https://eclipse.org/downloads/packages/eclipse-ide-cc-developers/neon3>

3.6.6 Eclipse ARM GCC Plug-In

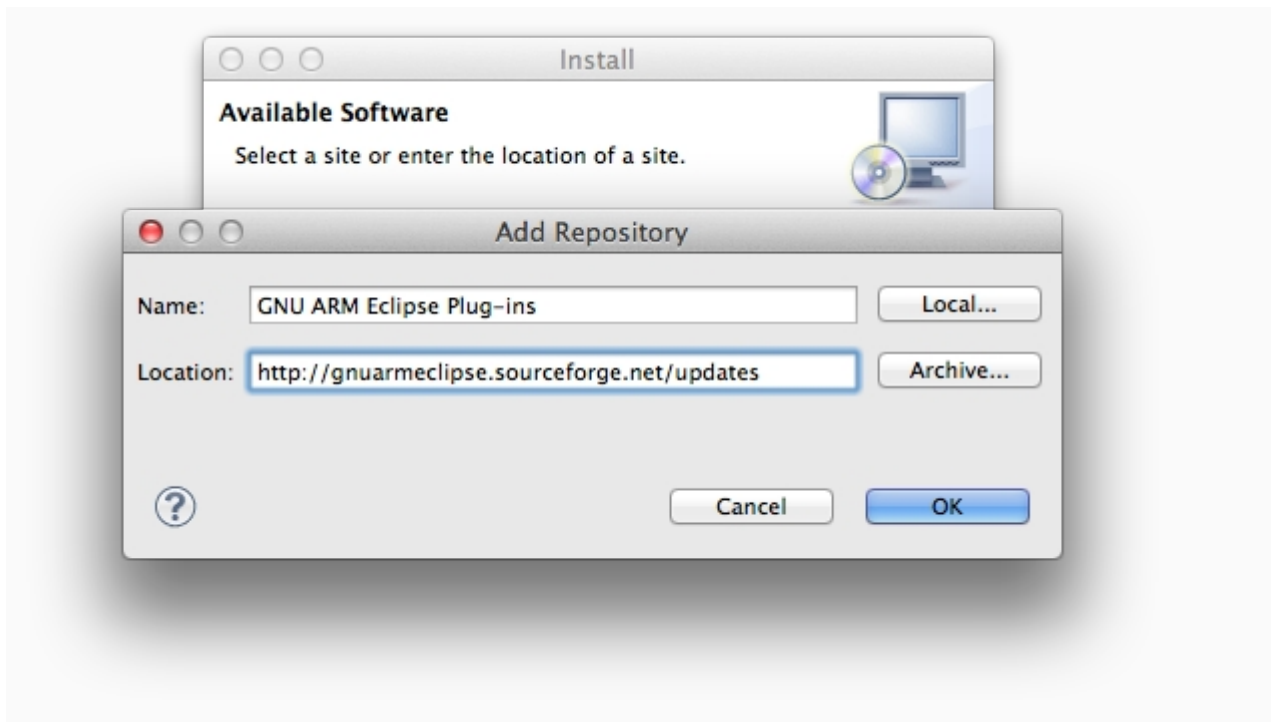
The classical way to install the GNU ARM Eclipse plug-ins is to use the Eclipse standard install/update mechanism: In the *Eclipse* menu → **Help** → **Install New Software...**

in the *Install* window, click the **Add...** button (on future updates, select the URL in the **Work with:** combo)

fill in *Name:* with **GNU ARM Eclipse Plug-ins**

fill in *Location:* with **<http://gnuarmeclipse.sourceforge.net/updates>**

click the **OK** button

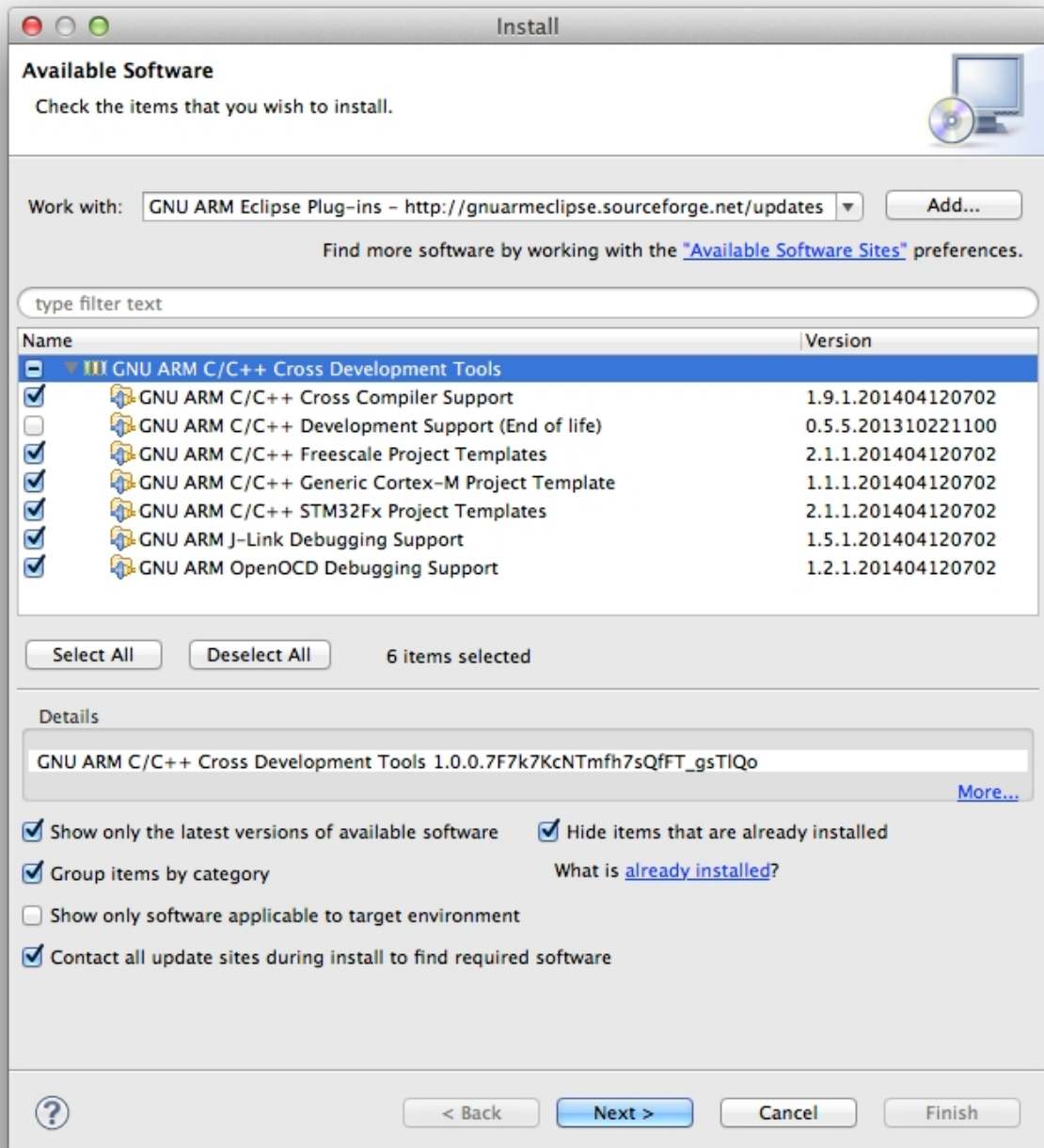


normally the main window should list a group named **CDT GNU Cross Development Tools**;
expand it

(in case the main window will list *There are no categorized items*, you are probably using a very
old version; disable the Group items by category option)

select all the plug-ins (the one marked *End of life* is needed only for compatibility with previous
version, normally can be safely skipped)

click the **Next** button and follow the usual installation procedure



Once you define the update site URL, further updates are greatly simplified (**Help** → **Check For Updates**)

4 System Setup

4.1 Make the System Partition Read Only [2.6.1 and Its Successors]

In order to prevent system files from being damaged due to illegal power failure, change the current system partition to read only. To modify a file, please change the read/write permissions of the partition.

The directory can be read and modified directly.

/home/

/opt/

/etc/network/

/etc/ppp/

/etc/openssh/

As other directories cannot be modified directly, please make them readable and writable first. The command is as follows:

4.1.1 Make the Partition Readable and Writable Temporarily

View the current status

```
root@adam3600:~# cat /proc/mounts
```

```
/dev/root / ext3 ro,relatime,data=ordered 0 0
```

```
devtmpfs /dev devtmpfs rw,relatime,size=99192k,nr_inodes=24798,mode=755 0 0
```

```
proc /proc proc rw,relatime 0 0
```

```
sysfs /sys sysfs rw,relatime 0 0
```

```
debugfs /sys/kernel/debug debugfs rw,relatime 0 0
```

```
tmpfs /var/volatile tmpfs rw,relatime,size=16384k 0 0
```

```
tmpfs /media tmpfs rw,relatime,size=16384k 0 0
```

```
/dev/mmcblk0p4 /home ext4 rw,relatime,data=ordered 0 0
```

```
/dev/mmcblk0p1 /media/mmcblk0p1 vfat rw,relatime,fmask=0022,dmask=0022,codepage=437,ioccharset=iso8859-
```

```
1,shortname=mixed,errors=remount-ro 0 0
```

```
/dev/mmcblk0p3 /media/mmcblk0p3 ext4 rw,relatime,data=ordered 0 0
```

```
devpts /dev/pts devpts rw,relatime,gid=5,mode=620,ptmxmode=000 0 0
```

Make it readable and writable

```
# mount -o remount,rw /
```

View the changed status

```
# cat /proc/mounts
```

```
/dev/root / ext3 rw,relatime,data=ordered 0 0
devtmpfs /dev devtmpfs rw,relatime,size=99192k,nr_inodes=24798,mode=755 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,relatime 0 0
debugfs /sys/kernel/debug debugfs rw,relatime 0 0
tmpfs /var/volatile tmpfs rw,relatime,size=16384k 0 0
tmpfs /media tmpfs rw,relatime,size=16384k 0 0
/dev/mmcblk0p4 /home ext4 rw,relatime,data=ordered 0 0
/dev/mmcblk0p1 /media/mmcblk0p1 vfat
rw,relatime,fmask=0022,dmask=0022,codepage=437,ioccharset=iso8859-
1,shortname=mixed,errors=remount-ro 0 0
/dev/mmcblk0p3 /media/mmcblk0p3 ext4 rw,relatime,data=ordered 0 0
devpts /dev/pts devpts rw,relatime,gid=5,mode=620,ptmxmode=000 0 0
```

4.1.2 Make the Partition Readable and Writable Permanently

```
# mount -o remount,rw /
```

```
# vi /etc/fstab
```

stock fstab - you probably want to override this with a machine specific one

rootfs	/	auto	ro	1	1
proc	/proc	proc	defaults		0 0
devpts	/dev/pts	devpts	mode=0620,gid=5		0 0
usbfs	/proc/bus/usb	usbfs	defaults		0 0
tmpfs	/var/volatile	tmpfs	defaults,size=16M		0 0
tmpfs	/dev/shm	tmpfs	mode=0777		0 0
tmpfs	/media/	tmpfs	defaults,size=16M		0 0
tmpfs	/media/ram	tmpfs	defaults,size=16M		0 0

uncomment this if your device has a SD/MMC/Transflash slot

```
#/dev/mmcblk0p1 /media/card auto defaults,sync,noauto 0 0
```

```
/dev/mmcblk0p4    /home                auto                defaults            0 2
```

Modify ro marked red above to defaults, and the modified file is as follows:

stock fstab - you probably want to override this with a machine specific one

```
rootfs            /                    auto                defaults            1 1
proc              /proc               proc                defaults            0 0
devpts            /dev/pts            devpts              mode=0620,gid=5    0 0
usbfs             /proc/bus/usb       usbfs               defaults            0 0
tmpfs             /var/volatile       tmpfs               defaults,size=16M  0 0
tmpfs             /dev/shm            tmpfs               mode=0777          0 0
tmpfs             /media/              tmpfs               defaults,size=16M  0 0
tmpfs             /media/ram          tmpfs               defaults,size=16M  0 0
```

uncomment this if your device has a SD/MMC/Transflash slot

```
#/dev/mmcblk0p1    /media/card         auto                defaults,sync,noauto 0 0
```

```
/dev/mmcblk0p4    /home                auto                defaults            0 2
```

After the modification, reboot the partition to make it take effect.

#reboot

Note: Please add # before command mount -o remount,ro to modify /etc/init.d/utilcheck.sh in Edgelink 2.6.1.

```
#!/bin/sh
#### BEGIN INIT INFO
# Provides: banner
# Required-Start:
# Required-Stop:
# Default-Start:    S
# Default-Stop:
#### END INIT INFO
```

```
#check whether is first run
```

```
TAGLINK_PATH=/home/sysuser
export TAGLINK_PATH
```

```
#set device config
mount -o remount,rw /
```

```
/usr/sbin/update-modules
```

```
sync
sync
# mount -o remount,ro /
exit 0
```

4.2 Serial Port

The hardware serial ports of the product are named ttyAP0, ttyAP1, and so on. The number of serial ports varies for different device types. Use `ls /dev/ttyAP*` to view the serial port on the device. Since some serial ports support RS232/RS485 and other modes, please place the jumper to the corresponding position based on the hardware manual before use.

```
# ls /dev/ttyAP*
/dev/ttyAP0 /dev/ttyAP1 /dev/ttyAP2 /dev/ttyAP3 /dev/ttyAP4
```

4.2.1 Test Whether the Serial Ports are Normal

Wire hardware com1 to com2. Please ensure that both com1 and com2 adopt RS232 mode or RS485 mode at the same time.

Enable the first ssh. Type in the below command in the command line.

```
#cat /dev/ttyAP0
```

Enable the second ssh. Type in the below command in the command line.

```
#echo "abcd" > /dev/ttyAP1
```

You will see "abcd" in the first window.

4.2.2 minicom Command

```
#minicom -s
```

Use the up and down arrow keys to select “Serial port setup” to go to the configuration page.

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup          |
| Modem and dialing           |
| Screen and keyboard         |
| Save setup as dfl           |
| Save setup as..            |
| Exit                         |
| Exit from Minicom          |
+-----+
```

```
+-----+
| A - Serial Device           : /dev/ttyAP0
| B - Lockfile Location       : /var/lock
| C - Callin Program         :
| D - Callout Program        :
| E - Bps/Par/Bits           : 9600 8N1
| F - Hardware Flow Control  : No
| G - Software Flow Control  : No
|
| Change which setting? █
+-----+
| Screen and keyboard        |
| Save setup as dfl         |
| Save setup as..          |
| Exit                      |
| Exit from Minicom        |
+-----+
```

Enter the followings on the configuration page:

Enter **A** to go to the configuration option of serial port name. Press **Enter** to finish editing.

Enter **E** to configure the baud rate.

Enter **F** to enable or disable hardware flow control.

Enter **G** to enable or disable software flow control.

Enter **ESC** to exit the configuration page.

```

+-----[configuration]-----+
| Filenames and paths         |
| File transfer protocols    |
| Serial port setup          |
| Modem and dialing          |
| Screen and keyboard        |
| Save setup as dfl          |
| Save setup as..            |
| Exit                       |
| Exit from Minicom          |
+-----+

```

After the configuration, select **Exit** to enter the communication mode.

```

root@localhost/etc/yum.repos.d
Welcome to minicom 2.3
OPTIONS: I18n
Compiled on Dec 4 2009, 12:16:46.
Port /dev/ttyAP0

Press CTRL-A Z for help on special keys

CTRL-A Z for help | 9600 8N1 | NOR | Minicom 2.3 | VT102 | Online 00:00

```

4.2.3 microcosm Command

```
#microcom -t 5000 -s 115200 /dev/ttyUSB3
```

at

OK

4.2.3.1 Run microcom Command to Read Modem

Please edit the document in /home/root directory. For other directories, please modify the relevant file location as needed.

4.2.3.2 Create an At Command File

Store commands to be executed

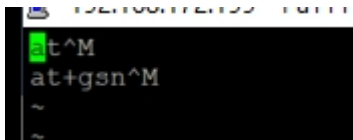
```
#vi at.txt  
at  
at+gsn
```

Convert the document to dos format,

```
#unix2dos at.txt
```

Confirm that the file format is correct

```
#vi at.txt
```



If the file is in DOS, each line will end with ^M.

4.2.3.3 Test

Execute the below command:

```
#microcom -t 500 -s 115200 /dev/ttyUSB2 < /home/root/at.txt > /home/root/gsn.txt
```

Copy bytes for stdin to TTY and from TTY to stdout

- d Wait up to DELAY ms for TTY output before sending every next byte to it
- t Exit if both stdin and TTY are silent for TIMEOUT ms
- s Set serial line to SPEED
- X Disable special meaning of NUL and Ctrl-X from stdin

-t 500 If there is no input or output operation for 500ms, you will exit the program

-s 115200 Set the baud rate to 115200

/dev/ttyUSB2 Name of virtual serial port, modified as the name of the actual serial port
< /home/root/at.txt Enter the information of /home/root/at.txt file. As the file must use \r as the newline character, please use unix2dos to convert it to newline character \r\n to execute the command.

> /home/root/gsn.txt Execution the result file.

View the execution results.

```
# cat gsn.txt
at
OK
at+gsn
862815030700775

OK
```

4.2.3.4 Add the Test Script to Startup Programs

```
#mount -o remount,rw /
#vi /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
microcom -t 500 -s 115200 /dev/ttyUSB2 < /home/root/at.txt > /home/root/gsn.txt
exit 0
```

After restarting the device, check whether /home/root/gsn.txt generates imei.

4.3 sramutil

sramutil is supported by successors of EdgeLink 2.8.0.

4.3.1 Help

```
# sramutil
```

```
fram dev name : /dev/sram
```

```
fram size: 32 k[ 32768 bytes]
```

```
Usage:
```

```
./sramutil r ADDRESS [WIDTH] ;read value with file
./sramutil w ADDRESS VALUE [WIDTH] ;write value with file
./sramutil mr ADDRESS [WIDTH] ;read value with mmap
./sramutil mw ADDRESS VALUE [WIDTH] ;write value with mmap
./sramutil hexshow ADDRESS length ;show value with file
./sramutil dump dump.bin ;dump sram to a file
WIDTH 8/16/32... default is 32
```

4.3.2 Execute Commands Using File Reading and Writing Method

Currently, users can only execute the command using file reading and writing method, and the address must be 4 byte-aligned.

```
./sramutil r ADDRESS [WIDTH] ;read value with file
```

```
./sramutil w ADDRESS VALUE [WIDTH] ;write value with file
```

```
WIDTH 8/16/32... default is 8
```

Read the content of position 0x0. 4 bytes.

```
# sramutil r 0x0
```

```
fram dev name : /dev/sram
```

```
fram size: 32 k[ 32768 bytes]
```

```
Read address = 0x0,value = 0x3020100[50462976],width = 32,return = 4
```

4.4 RTC Clock

4.4.1 RTC Clock Command

```
hwclock
```

```
Usage:
```

```
hwclock -f /dev/rtc1 //Display the current RTC time
```

```
hwclock -s -f /dev/rtc1 //Sync current RTC time to Linux time
hwclock -w -f /dev/rtc1 //Sync Linux time to RTC time
hwclock -f /dev/rtc1 -localtime //RTC time is the local time
hwclock -f /dev/rtc1 -utc //RTC time is the UTC time
hwclock --hctosys -f /dev/rtc1 //Sync the hardware clock to system clock
hwclock --systohc -f /dev/rtc1 //Sync the system clock to hardware clock.
```

Note: At present, a soft link /dev/rtc is created for rtc clocks of all platforms. You can directly access the clock information through the node.

4.4.2 Time Zone Configuration

Successors of OS2.6.1 are read only. As its time configuration file is a soft link, dd command, instead of cp command, must be used to update the time zone file.

```
dd if=/usr/share/zoneinfo/Asia/Shanghai of=/etc/localtime
```

4.5 Calibrate Time with ntp

4.5.1 ntp Client

Synchronizing time with ntpdate will cause time leap and affect time-dependent programs and services such as sleep and timer. Moreover, the ntpd service can calibrate time while correcting cpu tick. Ideally, ntpdate must be used to synchronize time at startup, and ntpd service is used to synchronize time at the rest of the time.

It should be noted that ntpd has a self-protection setting: if the time difference between the local computer and the remote server is too large, ntpd will not run. Therefore, you need to update time with ntpdate command for the newly configured time server before enabling the ntpd service. After the ntpd service runs, it will synchronize with the remote server every 64 seconds, gradually adjust its time through complex calculations based on the error values measured during every synchronization, and increase the synchronization interval as the error reduces. The adjustment will be repeated for every time leap.

General usage:

a) Use ntpdate to calibrate time at startup.

```
#ntpdate -t 3 -u -s edu.ntp.org.cn
```

b) Use ntpd for micro-calibration

```
#!/usr/sbin/ntpd
```

4.5.1.1 Calibrate Time with ntpdate

```
#ntpdate -t 3 -u -s edu.ntp.org.cn
```

-s Divert logging output from the standard output (default) to the system syslog facility.

-t TimeOut specifies the time to wait for a response. The value of the TimeOut is rounded to a multiple of 0.2 seconds and its default value is 1 second.

-u Direct ntpdate to use an unprivileged port to send data package.

4.5.1.2 Calibrate Time with ntpd

See the information marked red below for /etc/ntp.conf configuration file and server address configuration.

```
root@adam3600:~# cat /etc/ntp.conf
```

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help
```

```
driftfile /var/lib/ntp/ntp.drift
```

```
# Enable this if you want statistics to be logged.
```

```
statsdir /var/log/ntpstats/
```

```
statistics loopstats peerstats clockstats
```

```
filegen loopstats file loopstats type day enable
```

```
filegen peerstats file peerstats type day enable
```

```
filegen clockstats file clockstats type day enable
```

```
# You do need to talk to an NTP server or two (or three).
```

```
server time.windows.com
```

```
server 127.127.1.0
```

```
# Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for
```

```
# details. The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
```

```
# might also be helpful.
```

```
#
```

```
# Note that "restrict" applies to both servers and clients, so a configuration
```

```
# that might be intended to block requests from certain clients could also end
```

```
# up blocking replies from your own upstream servers.
```

```
# By default, exchange time with everybody, but don't allow configuration.
```

```
restrict -4 default kod notrap nomodify nopeer noquery
```

```
restrict -6 default kod notrap nomodify nopeer noquery
```

```
# Local users may interrogate the ntp server more closely.
```

```
restrict 127.0.0.1
```

```
# restrict ::1
```

```
# Clients from this (example!) subnet have unlimited access, but only if
```

```
# cryptographically authenticated.
```

```
# restrict 172.21.67.0 mask 255.255.255.0 nomodify
```

```
# If you want to provide time to your local subnet, change the next line.
```

```
# (Again, the address is an example only.)
```

```
#broadcast 192.168.123.255
```

```
# If you want to listen to time broadcasts on your local subnet, de-comment the
```

```
# next lines. Please do this only if you trust everybody on the network!
```

```
#disable auth
```

```
#broadcastclient
```

4.5.2 ntp Server

```
root@adam3600:~# cat /etc/ntp.conf
```

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help
```

```
driftfile /var/lib/ntp/ntp.drift
```

```
# Enable this if you want statistics to be logged.
```

```
statsdir /var/log/ntpstats/
```

```
statistics loopstats peerstats clockstats
```

```
filegen loopstats file loopstats type day enable
```

```
filegen peerstats file peerstats type day enable
```

```
filegen clockstats file clockstats type day enable
```

You do need to talk to an NTP server or two (or three).

```
server 127.127.1.0
```

Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for
details. The web page <<http://support.ntp.org/bin/view/Support/AccessRestrictions>>
might also be helpful.

```
#
```

Note that "restrict" applies to both servers and clients, so a configuration
that might be intended to block requests from certain clients could also end
up blocking replies from your own upstream servers.

By default, exchange time with everybody, but don't allow configuration.

```
restrict -4 default kod notrap nomodify nopeer noquery
```

```
restrict -6 default kod notrap nomodify nopeer noquery
```

Local users may interrogate the ntp server more closely.

```
restrict 127.0.0.1
```

```
# restrict ::1
```

Clients from this (example!) subnet have unlimited access, but only if
cryptographically authenticated.

```
# restrict 172.21.67.0 mask 255.255.255.0 nomodify
```

If you want to provide time to your local subnet, change the next line.

(Again, the address is an example only.)

```
#broadcast 192.168.123.255
```

If you want to listen to time broadcasts on your local subnet, de-comment the
next lines. Please do this only if you trust everybody on the network!

```
#disable auth
```

```
#broadcastclient
```

Startup method:

```
#!/usr/sbin/ntpd
```

4.5.3 Provide Time Services to Specified Network Segments Only

```
root@adam3600:~# cat /etc/ntp.conf
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help
```

```
driftfile /var/lib/ntp/ntp.drift
```

```
# Enable this if you want statistics to be logged.
statsdir /var/log/ntpstats/
```

```
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
```

```
# You do need to talk to an NTP server or two (or three).
server time.windows.com
server 127.127.1.0
```

```
# Access control configuration; see /usr/share/doc/ntp-doc/html/acopt.html for
# details. The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
```

```
#
```

```
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.
```

```
# By default, exchange time with everybody, but don't allow configuration.
```

```
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
```

```
# Local users may interrogate the ntp server more closely.
```

```
restrict 127.0.0.1
```

```
# restrict ::1
```



```
# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
```

```
restrict 172.21.67.0 mask 255.255.255.0 nomodify
```

```
# If you want to provide time to your local subnet, change the next line.
```

```
# (Again, the address is an example only.)
```

```
#broadcast 192.168.123.255
```

```
# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines. Please do this only if you trust everybody on the network!
```

```
#disable auth
```

```
#broadcastclient
```

4.5.4 ntpd's Relevant Commands

Check the time difference with server.

```
# ntpq -p
```

```
root@adam3600:~# ntpq -p
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
192.168.1.1	LOCAL(0)	6	u	55	64	37	0.518	-0.021	149923.
*LOCAL(0)	.LOCL.	5	l	33	64	77	0.000	0.000	0.004

```
=====
```

Note: The offset column lists the time difference with the server. If the time difference is too huge, please use the ntpdate command to update it first.

4.6 Configure a Fixed Network Card IP

4.6.1 View the Information of the Current Network Card

```
root@adam3600:~# ifconfig -a
```

```
eth0      Link encap:Ethernet  HWaddr 54:4A:16:8F:71:98
          inet addr:192.168.0.253  Bcast:0.0.0.0  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)
Interrupt:56

eth1 Link encap:Ethernet HWaddr 54:4A:16:8F:71:9A
inet addr:172.21.67.37 Bcast:0.0.0.0 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:102657 errors:0 dropped:3992 overruns:0 frame:0
TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:15166631 (14.4 MiB) TX bytes:5614 (5.4 KiB)

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:65536 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:0
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

4.6.2 Configure Temporary Network Card IP

```
root@adam3600:~# ifconfig eth0 192.168.1.252 netmask 255.255.255.0
```

```
root@adam3600:~# ifconfig eth0
```

```
eth0 Link encap:Ethernet HWaddr 54:4A:16:8F:71:98  
inet addr:192.168.1.252 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)  
Interrupt:56
```

4.6.3 Use DHCP to Obtain Temporary Network Card IP

```
root@adam3600:~# udhcpc -i eth1
```

```
udhcpc (v1.22.1) started
```

```
Sending discover...
Sending select for 172.21.67.37...
Lease of 172.21.67.37 obtained, lease time 1800
/etc/udhcpc.d/50default: Adding DNS 172.21.66.40
/etc/udhcpc.d/50default: Adding DNS 172.21.66.83
root@adam3600:~# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 54:4A:16:8F:71:9A
          inet addr:172.21.67.37  Bcast:0.0.0.0  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:113053 errors:0 dropped:4449 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16752245 (15.9 MiB)  TX bytes:6700 (6.5 KiB)
```

4.6.4 Runtime and Permanent Configuration of a Fixed IP

Change eth0 to a static IP address by modifying the relevant network card's name file in directory `/etc/network/interfaces.d/`

```
root@adam3600:~# vi /etc/network/interfaces.d/eth0
```

```
auto eth0
iface eth0 inet static
allow-hotplug eth0
address 192.168.0.253
netmask 255.255.255.0
gateway 192.168.0.1
```

4.6.5 Runtime and Permanent Configuration of a Dynamic IP

Change eth1 to a static IP address by modifying the relevant network card's name file in the directory `/etc/network/interfaces.d/`

```
root@adam3600:~# vi /etc/network/interfaces.d/eth1
```

```
auto eth1
iface eth1 inet dhcp
allow-hotplug eth1
```

4.7 WIFI Configuration

4.7.1 Check Whether the Wireless Network Card has been Correctly Identified

The interface name of the wireless network card is usually wlan0.

```
root@adam3600:~# ifconfig -a
```

```
eth0      Link encap:Ethernet  HWaddr 54:4A:16:8F:71:98
          inet addr:192.168.0.253  Bcast:0.0.0.0  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)
          Interrupt:56

eth1      Link encap:Ethernet  HWaddr 54:4A:16:8F:71:9A
          inet addr:172.21.67.37  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::564a:16ff:fe8f:719a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:207 errors:0 dropped:11 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:32497 (31.7 KiB)  TX bytes:1332 (1.3 KiB)

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:65536  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:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 00:0E:8E:6C:16:B3
          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)
```

4.7.2 Enable the Wireless Network Card

```
root@adam3600:~# ifconfig wlan0
```

```
wlan0      Link encap:Ethernet  HWaddr 00:0E:8E:6C:16:B3
           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)
```

```
root@adam3600:~# ifconfig wlan0 up
```

```
root@adam3600:~# ifconfig wlan0
```

```
wlan0      Link encap:Ethernet  HWaddr 00:0E:8E:6C:16:B3
           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)
```

4.7.3 Scan Searchable Wireless Network

```
root@adam3600:~# iwlist wlan0 scan
```

```
wlan0      Scan completed :
           Cell 01 - Address: 1C:AF:F7:C0:3D:E1
                   Channel:1
                   Frequency:2.412 GHz (Channel 1)
                   Quality=43/70  Signal level=-67 dBm
                   Encryption key:off
                   ESSID:"MOTT"
                   Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                               24 Mb/s; 36 Mb/s; 54 Mb/s
                   Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
                   Mode:Master
                   Extra:tsf=000000049fe1d8eb
                   Extra: Last beacon: 60ms ago
```

IE: Unknown: 00044D4F5454
IE: Unknown: 010882848B962430486C
IE: Unknown: 030101
IE: Unknown: 2A0100
IE: Unknown: 2F0100
IE: Unknown: 32040C121860
IE: Unknown:

2D1A6C181BFF00

IE: Unknown:

3D1601001300

IE: Unknown: DD090010180201F0010000

IE: Unknown:

DD1E00904C336C181BFF00

IE: Unknown:

DD1A00904C3401001300

Cell 02 - Address: C8:3A:35:05:3E:80

Channel:2

Frequency:2.417 GHz (Channel 2)

Quality=63/70 Signal level=-47 dBm

Encryption key:on

ESSID:"WebAccess"

Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
24 Mb/s; 36 Mb/s; 54 Mb/s

Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s

Mode:Master

Extra:tsf=0000000000a986e3

Extra: Last beacon: 60ms ago

IE: Unknown: 0009576562416363657373

IE: Unknown: 010882840B162430486C

IE: Unknown: 030102

IE: Unknown: 2A0100

IE: Unknown: 2F0100

IE: IEEE 802.11i/WPA2 Version 1

Group Cipher : CCMP

Pairwise Ciphers (1) : CCMP

Authentication Suites (1) : PSK

IE: Unknown: 32040C121860

IE: Unknown:


```
update_config=1
```

```
# Only WPA-PSK is used. Any valid cipher combination is accepted.
```

```
network={  
    ssid="Advantech"  
    scan_ssid=1  
    key_mgmt=NONE  
    priority=1  
}
```

4.7.6 Connect AP

```
root@adam3600:~# wlan.sh up
```

```
Successfully initialized wpa_supplicant
```

```
OK
```

```
udhcpc (v1.22.1) started
```

```
Sending discover...
```

```
Sending discover...
```

```
Sending discover...
```

```
Sending discover...
```

```
Sending select for 192.168.10.36...
```

```
Lease of 192.168.10.36 obtained, lease time 86400
```

```
RTNETLINK answers: File exists
```

```
/etc/udhcpc.d/50default: Adding DNS 192.168.10.1
```

```
/etc/udhcpc.d/50default: Adding DNS 0.0.0.0
```

```
OK
```

```
root@adam3600:~# iwconfig wlan0
```

```
wlan0 IEEE 802.11bgn ESSID:"WebAccess"
```

```
Mode:Managed Frequency:2.417 GHz Access Point: C8:3A:35:05:3E:80
```

```
Bit Rate=1 Mb/s Tx-Power=20 dBm
```

```
Retry long limit:7 RTS thr:off Fragment thr:off
```

```
Encryption key:off
```

```
Power Management:off
```

```
Link Quality=55/70 Signal level=-55 dBm
```

```
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
```

```
Tx excessive retries:1 Invalid misc:8 Missed beacon:0
```


4.7.7 Save SSID Password by Encryption

The parameters of command `wpa_passphrase` is username and password.

```
root@adam3600:~# wpa_passphrase max 1234567890
```

```
network={
    ssid="max"
    #psk="1234567890"
    psk=4b2bc7cbb3710e9ea43f09e8d57e8bdb3b2a2127af44960d73216c3612f6baba
}
```

Copy encrypted password `psk=` into `wpa_supplicant.conf`.

The final file is as follows:

```
network={
    ssid="max" //Fill in the username of the wireless network
    key_mgmt=WPA-PSK
    proto=WPA
    pairwise=TKIP
    group=TKIP
    psk=4b2bc7cbb3710e9ea43f09e8d57e8bdb3b2a2127af44960d73216c3612f6baba
}
```

4.7.8 Configure a Fixed IP

```
root@adam3600:~# vi /etc/network/interfaces.d/wlan0
```

```
auto wlan0
iface wlan0 inet static
allow-hotplug wlan0
address 192.168.0.253
netmask 255.255.255.0
gateway 192.168.0.1
```

4.7.9 Configure a Static IP

```
root@adam3600:~# vi /etc/network/interfaces.d/wlan0
```

```
auto wlan0
iface wlan0 inet dhcp
```

```
allow-hotplug wlan0
```

4.8 Configure WIFI as AP Mode

4.8.1 Configuration File

The Hostapd.conf configuration file is as below, wherein, SSID is “abc” and the password is “12345678”.

```
root@adam3600:~# vi /home/root/hostap_wlan0.conf
```

```
interface=wlan0
driver=nl80211
ssid=abc
channel=6
hw_mode=g
ignore_broadcast_ssid=0
auth_algs=1
wpa=3
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

4.8.2 Enable the Program

The command to enable the program is:

```
#hostapd -B /home/root/hostap_wlan0.conf
# ifconfig wlan0 192.168.1.34 netmask 255.255.255.0
```

4.8.3 Add Startup Programs

```
# vi /etc/rc.local
```

```
#!/bin/sh -e
#
# rc.local
```

```
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
```

```
hostapd -B /home/root/hostap_wlan0.conf
ifconfig wlan0 192.168.1.34 netmask 255.255.255.0
```

```
exit 0
```

4.9 DHCP Service

4.9.1 Configuration File

```
#vi /home/root/udhcpd_wlan0.conf
# The start and end of the IP lease block
start    192.168.0.20
end      192.168.0.254

# The interface that udhcpd will use
#interface eth0
interface wlan0
opt      dns      8.8.8.8 8.8.4.4 #public google dns servers
option  subnet  255.255.255.0
opt      router  192.168.10.1
#opt     wins    192.168.10.10
#option  dns     129.219.13.81 # appended to above DNS servers for a total of 3
#option  domain  local
option  lease    864000          # default: 10 days
```

4.9.2 Enable Services

```
#!/usr/sbin/udhcpd -S /home/root/udhcpd_wlan0.conf
```

Basic usage of each parameter

Usage: udhcpd [-fS] [-I ADDR] [CONFFILE]

DHCP server

- f Run in foreground
- S Log to syslog too
- I ADDR Local address
- a MSEC Timeout for ARP ping (default 2000)

4.10 Cellular Information Query [Supported by 2.6.1 and Its Successors]

When using the program, please ensure that the serial port of the module is not in use, so as to correctly identify whether each serial port can return an AT command.

modemscan

Current con tty: /dev/pts/0

Current cmd tty: ttyO0

=====tty used info=====

1347 /bin/tinylogin /dev/ttyO0

1347 /bin/tinylogin /dev/ttyO0

1347 /bin/tinylogin /dev/ttyO0

1348 /bin/tinylogin /dev/tty1

1348 /bin/tinylogin /dev/tty1

1348 /bin/tinylogin /dev/tty1

1347 root 0:00 /sbin/getty 115200 ttyO0

1348 root 0:00 /sbin/getty 38400 tty1

1412 root 0:00 sh -c ps aux | grep tty

1414 root 0:00 grep tty

-----tty used info-----

scan_interface:[/sys/bus/usb/devices//2-1:1.0/]driver:option

scan_interface:[/sys/bus/usb/devices//2-1:1.1/]driver:option

scan_interface:[/sys/bus/usb/devices//2-1:1.2/]driver:option

scan_interface:[/sys/bus/usb/devices//2-1:1.3/]driver:option

scan_interface:[/sys/bus/usb/devices//2-1:1.4/]driver:qmi_wwan

set_operation_quectel

quectel_pre_init,vendor=2c7c,procut=0125

```
scan_interface:[/sys/bus/usb/devices//2-1:1.0/]driver:option
scan_interface:[/sys/bus/usb/devices//2-1:1.1/]driver:option
scan_interface:[/sys/bus/usb/devices//2-1:1.2/]driver:option
scan_interface:[/sys/bus/usb/devices//2-1:1.3/]driver:option
scan_interface:[/sys/bus/usb/devices//2-1:1.4/]driver:qmi_wwan
set_operation_quectel
quectel_pre_init,vendor=2c7c,proecut=0125
```

```
=====USB Driver Info=====
```

```
[0]type=1, driver=option, node=/dev/ttyUSB0
[1]type=1, driver=option, node=/dev/ttyUSB1
[2]type=4, driver=option, node=/dev/ttyUSB2
[3]type=4, driver=option, node=/dev/ttyUSB3
[4]type=2, driver=qmi_wwan, node=wwan0
```

```
=====USB Interface device Info=====
```

```
QMI count=1,interface=4
```

```
QMI interface=wwan0
```

```
PPP count=2,
```

```
ppp interface[0]=/dev/ttyUSB2
```

```
ppp interface[1]=/dev/ttyUSB3
```

```
    [gps status:0]
```

```
    [gps status:1]
```

```
[0]type=1, driver=option, node=/dev/ttyUSB0
[1]type=1, driver=option, node=/dev/ttyUSB1
[2]type=4, driver=option, node=/dev/ttyUSB2
[3]type=4, driver=option, node=/dev/ttyUSB3
[4]type=2, driver=qmi_wwan, node=wwan0
```

```
=====Cellular device Info=====
```

```
find modem usbid [2c7c:0125]
```

```
[0] [/dev/ttyUSB0] tty port
```

```
[1] [/dev/ttyUSB1] tty port
```

```
[2] [/dev/ttyUSB2] at port
```

```
[3] [/dev/ttyUSB3] at port
```

```
[4] [wwan0] qmi net
```

```
Modem info:
```

```
    [version:   Quectel  EC20F  Revision: EC20CEFAR02A04M4G   ]
```

```
[modem imei=862815030700775]
[sim status: READY]
[imsi :460115864165295]
[csq= 30,ber=99]
[operator: mode= 0,format=2,oper="46011",act=7]
[operator: mode= 0,format=0,oper="CHN-CT",act=7]
```

=====
=====XML device Info=====

```
get_modem_version:/dev/ttyUSB2,0x7d728
```

Please copy the xml data to </home/sysuser/project/CellularDeviceInfo.acr> or </home/root/project/CellularDeviceInfo.acr>

```
<Device deviceID="2c7c:0125" GPSType="embedded" GPSInterface="1" ATPortCount="2" ATPortInterface="2,3"
DialType="ppp" deviceName="  Quectel  EC20F  Revision: EC20CEFAR02A04M4G  QMI(wwan0) " />
```

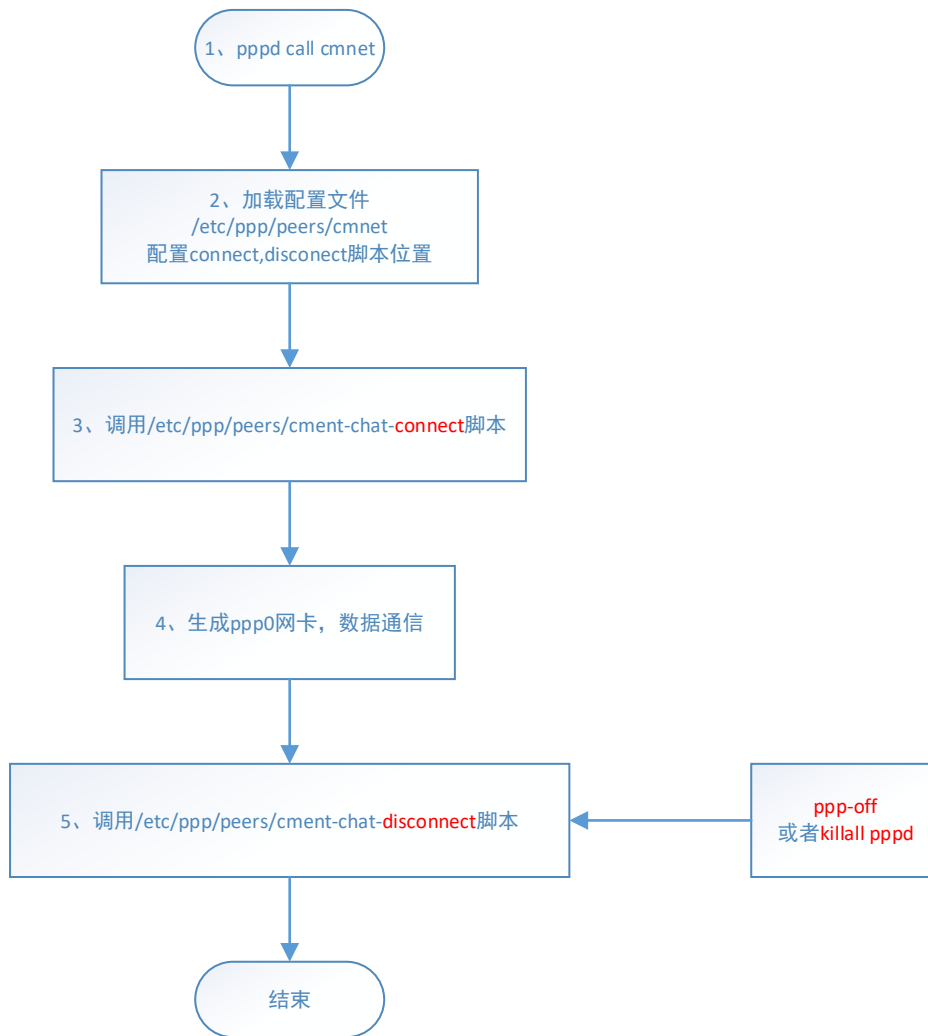
See 4.14.3 Description of Configuration File Cellulardeviceinfo.Acr [Supported by Successors of 2.7.0] for the content analysis of xml file format.

4.11 Cellular Communication Configuration

Use pppd program provided by the platform for dialing.

4.11.1 pppd Dialing Process

The pppd dialing process is as below:



1. **pppd call cmnet** in the command line.
2. pppd will load configuration parameters **/etc/ppp/peers/cmnet** mainly to initialize the name of serial port **/dev/ttyXXX**, and configure locations for **connect** and **disconnect** scripts.
3. Call script **/etc/ppp/peers/cmnet-chat-connect** for dialing.
4. Generate **ppp0** network card for communication.
5. Call script **/etc/ppp/peers/cmnet-chat-disconnect** to disconnect.

We can modify the connect script as needed and set commands or parameters based on the previous process.

4.11.2 Check the Communication Module

Confirm that the module has been installed correctly and the number of available serial ports according to the hardware manual. You can also use the modemsan command [supported by successors of version 2.6.1] to search for the information of available modems.

root@adam3600:~# **dmesg | grep tty**

```
[ 0.000000] Kernel command line: console=ttyO0,115200n8 root=/dev/mmcbk0p2 ro rootfstype=ext3 rootwait ip=None
[ 1.553842] serial8250.0: ttyS0 at MMIO 0x1000000 (irq = 161, base_baud = 921600) is a XR16850
[ 1.554793] serial8250.0: ttyS1 at MMIO 0x1000801 (irq = 160, base_baud = 921600) is a XR16850
[ 1.555659] serial8250.0: ttyS2 at MMIO 0x1001201 (irq = 250, base_baud = 921600) is a XR16850
[ 1.556892] 44e09000.serial: ttyO0 at MMIO 0x44e09000 (irq = 88, base_baud = 3000000) is a OMAP UART0
[ 2.233388] console [ttyO0] enabled
[ 2.238325] 48022000.serial: ttyO1 at MMIO 0x48022000 (irq = 89, base_baud = 3000000) is a OMAP UART1
[ 2.249277] 481a6000.serial: ttyO3 at MMIO 0x481a6000 (irq = 60, base_baud = 3000000) is a OMAP UART3
[ 2.536298] serial_init: registered 4 ttyGS* devices
[ 25.741860] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB0
[ 25.757201] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB1
[ 25.772418] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB2
[ 25.787615] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB3
[ 25.802911] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB4
[ 25.818109] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB5
[ 25.833328] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB6
[ 25.848539] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB7
[ 25.863781] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB8
[ 25.879082] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB9
[ 25.894403] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB10
[ 25.909913] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB11
[ 25.925279] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB12
[ 25.940661] usb 2-1: GSM modem (1-port) converter now attached to ttyUSB13
```

4.11.3 Dialing Command

a) pppd dials the number directly.

root@adam3600:~# **pppd call default /dev/ttyUSB1 &**

pppd is the program name.

call is the action parameter.

default is the name of the script called.

/dev/ttyUSB1 is the parameter for replacing serial port in the script.

& means the program will run in the background.

b) Use a configured script for dialing.

root@adam3600:~# **wan.sh**

Usage: wan.sh unicom|cmnet|telecom|other [devicename]

```
root@adam3600:~# wan.sh default /dev/ttyUSB1
```

Usage: wan.sh unicom|cmnet|telecom|other [devicename]

Below is a common dial prompt:

```
root@adam3600:~# wan.sh default /dev/ttyUSB1
```

```
killall: pppd: no process killed
```

```
timeout set to 30 seconds
```

```
#The execution content of the chat-connect script
```

```
will be displayed below
```

```
abort on (NO CARRIER)
```

```
abort on (ERROR)
```

```
abort on (NO DIALTONE)
```

```
abort on (BUSY)
```

```
abort on (NO ANSWER)
```

```
send (AT^M)
```

```
expect (OK)
```

```
AT^M^M
```

```
OK
```

```
-- got it
```

```
send (ATZ^M)
```

```
expect (OK)
```

```
^M
```

```
ATZ^M^M
```

```
OK
```

```
-- got it
```

```
send (AT+CGDCONT=1,"IP","3GNET",,0,0^M)
```

```
expect (OK)
```

```
^M
```

```
AT+CGDCONT=1,"IP","3GNET",,0,0^M^M
```

```
OK
```

```
-- got it
```

```
send (ATDT*99#^M)
```

```
expect (CONNECT)
```

```
^M
```

```
ATDT*99#^M^M
```

CONNECT

-- got it

send (^M)

Script /usr/sbin/chat -s -v -f /etc/ppp/peers/default-chat-connect finished (pid 2441), status = 0x0

Prompt for successful execution of script #chat-

connect

Serial connection established.

using channel 1

Using interface ppp0

Connect: ppp0 <--> /dev/ttyUSB1 #Generate ppp0 network card and start protocol negotiation with operators

Connect: ppp0 <--> /dev/ttyUSB1

rcvd [LCP ConfReq id=0x1 <asynctest 0x0> <magic 0x79049dfc> <pcomp> <accomp>]

Warning - secret file /etc/ppp/pap-secrets has world and/or group access

sent [LCP ConfReq id=0x1 <asynctest 0x0> <magic 0x28feffa7> <pcomp> <accomp>]

sent [LCP ConfAck id=0x1 <asynctest 0x0> <magic 0x79049dfc> <pcomp> <accomp>]

rcvd [LCP ConfAck id=0x1 <asynctest 0x0> <magic 0x28feffa7> <pcomp> <accomp>]

sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]

sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]

sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]

rcvd [IPCP ConfReq id=0x1]

sent [IPCP ConfNak id=0x1 <addr 0.0.0.0>]

rcvd [IPCP ConfNak id=0x1 <addr 10.53.206.231> <ms-dns1 123.123.123.123> <ms-dns2 123.123.123.124>]

sent [IPCP ConfReq id=0x2 <addr 10.53.206.231> <ms-dns1 123.123.123.123> <ms-dns2 123.123.123.124>]

rcvd [IPCP ConfReq id=0x2 <addr 10.53.206.231>]

sent [IPCP ConfAck id=0x2 <addr 10.53.206.231>]

rcvd [IPCP ConfAck id=0x2 <addr 10.53.206.231> <ms-dns1 123.123.123.123> <ms-dns2 123.123.123.124>]

not replacing existing default route via 172.21.67.1

local IP address 10.53.206.231

remote IP address 10.53.206.231

primary DNS address 123.123.123.123

secondary DNS address 123.123.123.124

Script /etc/ppp/ip-up started (pid 2476)

Script /etc/ppp/ip-up finished (pid 2476), status = 0x0

4.11.4 Configuration Parameter Parsing

```
root@adam3600:/etc/ppp/peers# ls default*
```

```
default          default-chat-connect  default-chat-disconnect
```

View the content of the default script

View default script content

```
root@adam3600:/etc/ppp/peers# cat default
```

```
debug          # Syslog will output relevant information when the parameter is enabled.
#nodetach      # pppd is not running in the background, ctrl+c will interrupt pppd.
/dev/ttyUSB1   # The parameter is the default serial port name used. If it is in the
command line, execute it.
115200        # Baud rate of serial port
noctrlscts
lock
usepeerdns    #The parameter uses the information of DNS server
noauth        # The parameter must be enabled to disable the use of auth
authorization.
noipdefault
novj
novjccomp
noccip
defaultroute  #Use the default route provided by the server. If there is a default
route locally, the route will not be added.
#lcp-echo-failure 5
#lcp-echo-interval 30
persist
#ipcp-accept-local
#ipcp-accept-remote
connect '/usr/sbin/chat -s -v -f /etc/ppp/peers/default-chat-connect'
disconnect '/usr/sbin/chat -s -v -f/etc/ppp/peers/default-chat-disconnect'
```

4.11.5 Parameter Parsing for connect Script

Check chat connect script which is used to send AT command before the modem dialing to deploy relevant configurations. For example, APN is also configured in the script. You can also modify the script to query or set configuration commands.

```
root@adam3600:/etc/ppp/peers# cat default-chat-connect
```

TIMEOUT 30 Set the timeout (default value is 45s) for responding to AT command.
If the timeout has expired, quit dialing.

ABORT "NO CARRIER" #Quit dialing if "NO CARRIER" is returned

ABORT "ERROR"

ABORT "NO DIALTONE"

ABORT "BUSY"

ABORT "NO ANSWER"

"" AT #Send AT command to confirm that the modem is working, "" means the execution result of the previous line does not need to be placed at the beginning of the next line.

OK ATZ #If OK is returned for the previous command, then ATZ command will be sent.

OK AT+CGDCONT=1,"IP","3GNET",,0,0 #The parameter is used to set APN.

OK ATDT*99# # Formally start dialing the number of the call center.

CONNECT ""

Meanings of common commands:

ATZ Restore to factory settings

AT+CGDCONT Set APN

Value definition

□ <cid>: 1-4 is the index value of PDP's setup environment. Other PDP-related commands can use the index value to call the saved settings.

□ <PDP_type>: String value, indicating the type of packet switching protocol.

Value Definition

IP IPv4 protocol

IPV6 IPv6 protocol

IPV4V6 IPv4/v6 protocol

PPP End-to-end protocol

□ <APN>: String value, indicating the domain name of the access point connecting to GGSN or extranet.

Operator	Access point	Username and password	Dialing number	Note
China Mobile	cmnet cmwap	Nil	*99***1#	2.5G/2.75G (GPRS)
China Mobile	cmnet cmwap	Nil	*98*1#	3G(TD-SCDMA)
China Unicom	3gnet	Nil	*99#	3G(WCDMA)

China Telecom	Nil	card	#777	CDMA200
---------------	-----	------	------	---------

<PDP_addr>: String value, indicating MS address.

<d_comp>: Numeric value that controls the compression of PDP data.

Value	Definition
0	Uncompressed
1	Compressed

Note: When there is no <d_comp>, it means the value of <d_comp> is 0.

<h_comp>: Numeric value that controls the compression of PDP header.

Value	Definition
0	Uncompressed
1	Compressed

Note: when there is no < h_comp >, it means the value of < h_comp > is 0.

4.11.6 Parameter Parsing for Disconnect Script

chat-disconnect script is the AT command sent when the connection is terminated.

```
root@adam3600:/etc/ppp/peers# cat default-chat-disconnect
```

```
ABORT "ERROR"
```

```
ABORT "NO DIALTONE"
```

```
SAY "\nSending break to the modem\n"
```

```
" \K"
```

```
" +++ATH"
```

```
SAY "\nGoodbye\n"
```

4.11.7 Common Operators' connect Scripts

4.11.7.1 China Mobile

```
root@adam3600:/etc/ppp/peers# cat cmnet-chat-connect
```

```
TIMEOUT 30
```

```
ABORT "NO CARRIER"
```

```
ABORT "ERROR"
```

```
ABORT "NO DIALTONE"
```

```
ABORT "BUSY"
```

```
ABORT "NO ANSWER"
```

```
"" AT
```

```
#OK AT+COPS=2
#OK AT+URAT=1,2
#OK AT+COPS=0
OK ATZ
OK AT+CGDCONT=1,"IP","CMNET"
OK ATDT*99**1#
CONNECT ""
```

4.11.7.2 China Telecom

```
root@adam3600:/etc/ppp/peers# cat telecom-chat-connect
TIMEOUT 60
ABORT "NO CARRIER"
ABORT "ERROR"
ABORT "NO DIALTONE"
ABORT "BUSY"
ABORT "NO ANSWER"
"" AT
OK ATZ
OK ATDT#777
CONNECT ""
```

4.11.7.3 China Unicom

```
root@adam3600:/etc/ppp/peers# cat unicom-chat-connect
TIMEOUT 30
ABORT "NO CARRIER"
ABORT "ERROR"
ABORT "NO DIALTONE"
ABORT "BUSY"
ABORT "NO ANSWER"
"" AT
OK ATZ
OK AT+CGDCONT=1,"IP","3GNET",,0,0
OK ATDT*99#
CONNECT ""
```

4.11.7.4 Custom connect Script

The below use case takes the unicom-chat-connect script as an example.

```
root@adam3600:/etc/ppp/peers# cat unicom-chat-connect
TIMEOUT 30
ABORT "NO CARRIER"
ABORT "ERROR"
ABORT "NO DIALTONE"
ABORT "BUSY"
ABORT "NO ANSWER"
"" AT
OK ATZ
OK AT+COPS?           #Add a custom command to query registered operators.
OK AT+CNUM;+CSQ #Add multiple commands, separated by ";", in one line. The second
command does not need AT at the beginning.
OK AT+CGDCONT=1,"IP","3GNET",,0,0
OK ATDT*99#
CONNECT ""
```

4.11.8 ZTE ME3760 Module Configuration

The ZTE ME3460 module needs to use special commands for dialing rather than ppp.

```
root@adam3600:~# AutoDialup4G
```

```
netcard [eth2] not exist
```

Usage:

```
AutoDialup4G com_port_name netcard
```

```
example: AutoDialup4G /dev/ttyUSB0 eth2
```

Configure the program parameters according to actual needs.

```
root@adam3600:~# AutoDialup4G /dev/ttyUSB1 eth4
```

```
com port:/dev/ttyUSB1,netcard:eth4
```

```
[ 0]AT
```

```
[ 0]AT OK
```

The program automatically sends commands as follows:

```
"AT"
```

```
"AT+ZGACT?"
```

```
"AT^SYSCONFIG=17,0,1,1"
```

```
"AT+CFUN=1"
```

```
"AT^SYSINFO"  
"AT+CGACT=1,1"  
"AT+ZGACT?"  
"AT+ZGACT=1,1"
```

4.11.9 List of Common AT Commands for Debugging

Since each modem manufacturer has its own AT command list, only a few common commands are listed for your reference. Use minicom to open the virtual serial port corresponding to the modem for inquiry. After opening the serial port, send an AT command to see if OK is returned to confirm whether the serial port is configurable.

Use minicom to read module information

```
# minicom -D /dev/ttyUSB0
```

Use microcom to read module information

```
#microcom -t 15000 -s 115200 /dev/ttyUSB0
```

```
Welcome to minicom 2.7
```

```
OPTIONS: I18n
```

```
Compiled on Jun 20 2014, 20:17:16.
```

```
Port /dev/ttyUSB0, 09:45:28
```

```
Press CTRL-A Z for help on special keys
```

```
at
```

```
OK
```

4.11.9.1 Check SIM Card's Status

```
at+cpin?
```

```
+CPIN: READY
```

```
OK
```

Other parameters returned:

```
ERROR: MT is not found sim card
```

```
READY: MT is not pending for any password
```

```
SIM PIN: MT is waiting for SIM PIN to be given
```


SIM PUK: MT is waiting for SIM PUK to be given

SIM PIN2: MT is waiting for SIM PIN2 to be given

SIM PUK2: MT is waiting for SIM PUK2 to be given

PH-NET PIN: MT is waiting for network personalization password to be given

PH-NET PUK: MT is waiting for network personalization unblocking password to be given

PH-NETSUB PIN: MT is waiting for network subset personalization password to be given

PH-NETSUB PUK: MT is waiting for network subset personalization unblocking password to be given

PH-SP PIN: MT is waiting for service provider personalization password to be given

PH-SP PUK: MT is waiting for service provider personalization unblocking password to be given

PH-CORP PIN: MT is waiting for corporate personalization password to be given

PH-CORP PUK: MT is waiting for corporate personalization unblocking password to be given

4.11.9.2 Search for Information of Connected Operators

at+cops?

+COPS: 0,0,"CHINA-UNICOM",7

OK

+COPS: <mode>[,<format>[,<oper>][,<Act>]]

<mode>

0 Automatic mode. <oper> field is ignored

1 Manual operator selection. <oper> field shall be present and <Act> optionally

2 Manually deregister from network

3 Set only <format> (for **AT+COPS?** Read Command), and do not attempt registration/deregistration (<oper> and <Act> fields are ignored). This value is invalid in the response of Read Command.

4 Manual/automatic selection. <oper> field shall be presented. If manual selection fails, automatic mode (<mode>=0) is entered

<format>

0 Long format alphanumeric <oper> which can be up to 16 characters long

1 Short format alphanumeric <oper>

2 Numeric <oper>. GSM location area identification number

<Act>

Access technology selected. Values 3, 4, 5 and 6 occur only in the response of Read Command while MS is in data service state and is not intended for the **AT+COPS** Write Command.

- 0 GSM
- 2 UTRAN
- 3 GSM W/EGPRS
- 4 UTRAN W/HSDPA
- 5 UTRAN W/HSUPA
- 6 UTRAN W/HSDPA and HSUPA
- 7 E-UTRAN
- 100 CDMA

The following information will be returned if operator's base station is not connected:

at+cops?

+COPS: 0

OK

4.11.9.3 Check the Phone Number

at+cnum

+CNUM: "", "+8618600100000", 145

OK

Please write cell phone number in the sim card to return a normal result for the command. Otherwise, ERROR will be returned.

[+CNUM: [<alpha>],<number>,<type>]

<alpha>

Optional alphanumeric string associated with **<number>**.

<number>

String type phone number of format specified by **<type>**

<type>

Type of address of octet in integer format (Refer to *3GPP TS 24.008 subclause 10.5.4.7* for details). Usually, it has three kinds of values:

- 129 Unknown type
- 145 International type (contains the character "+")
- 161 National type

4.11.9.4 Check Signal Strength

```
at+csq
```

```
+CSQ: 21,99
```

```
OK
```

```
+CSQ: <rss>,<ber>
```

```
<rss>
```

```
0 -113dBm or less
```

```
1 -111dBm
```

```
2...30 -109dBm... -53dBm
```

```
31 -51dBm or greater
```

```
99 Not known or not detectable
```

```
100 -116dBm or less
```

```
101 -115dBm
```

```
102...190 -114dBm...-26dBm
```

```
191 -25dBm or greater
```

```
199 Not known or not detectable
```

```
100~199 Extended to be used in TD-SCDMA indicating received signal code
```

```
power (RSCP)
```

```
<ber>
```

```
Channel bit error rate (in percent)
```

```
0...7 As RXQUAL values in the table in 3GPP TS 45.008 subclause 8.2.4
```

```
99 Not known or not detectable
```

4.124G Model Resetting

Confirm BUS number:

```
# lsub
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

```
Bus 002 Device 002: ID 2c7c:0125
```

```
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Use lsub command to confirm that the USB bus number of device 2c7c:0125 is 2.

View the parameters of reset script.

```
# /usr/bin/minipcie_reset.sh
```

usage : minipcie_reset.sh PowerReset/ModuleReset BusNum

Reset module

```
#/usr/bin/minipcie_reset.sh PowerReset 2
```

```
crw-rw----  1 root    dialout  188,  0 Jul 22 14:43 /dev/ttyUSB0
crw-rw----  1 root    dialout  188,  1 Jul 22 14:43 /dev/ttyUSB1
crw-rw----  1 root    dialout  188,  2 Jul 22 14:43 /dev/ttyUSB2
crw-rw----  1 root    dialout  188,  3 Jul 22 14:43 /dev/ttyUSB3
```

Modem initial Success....

4.13 Dual sim Card Switching

The method is only applicable for dual SIM card devices such as ADAM3600DS and ECU1051.

```
# /usr/bin/sim_switch.sh
```

usage: /usr/bin/sim_switch.sh [1/2]

parameters and options:

[1 -> SMI1]

[2 -> SIM2]

```
#/usr/bin/sim_switch.sh 1
```

board name : ecu1051

switch to SIM1

```
crw-rw----  1 root    dialout  188,  0 Jul 22 15:01 /dev/ttyUSB0
crw-rw----  1 root    dialout  188,  1 Jul 22 15:01 /dev/ttyUSB1
crw-rw----  1 root    dialout  188,  2 Jul 22 15:01 /dev/ttyUSB2
crw-rw----  1 root    dialout  188,  3 Jul 22 15:01 /dev/ttyUSB3
```

Modem initial Success....

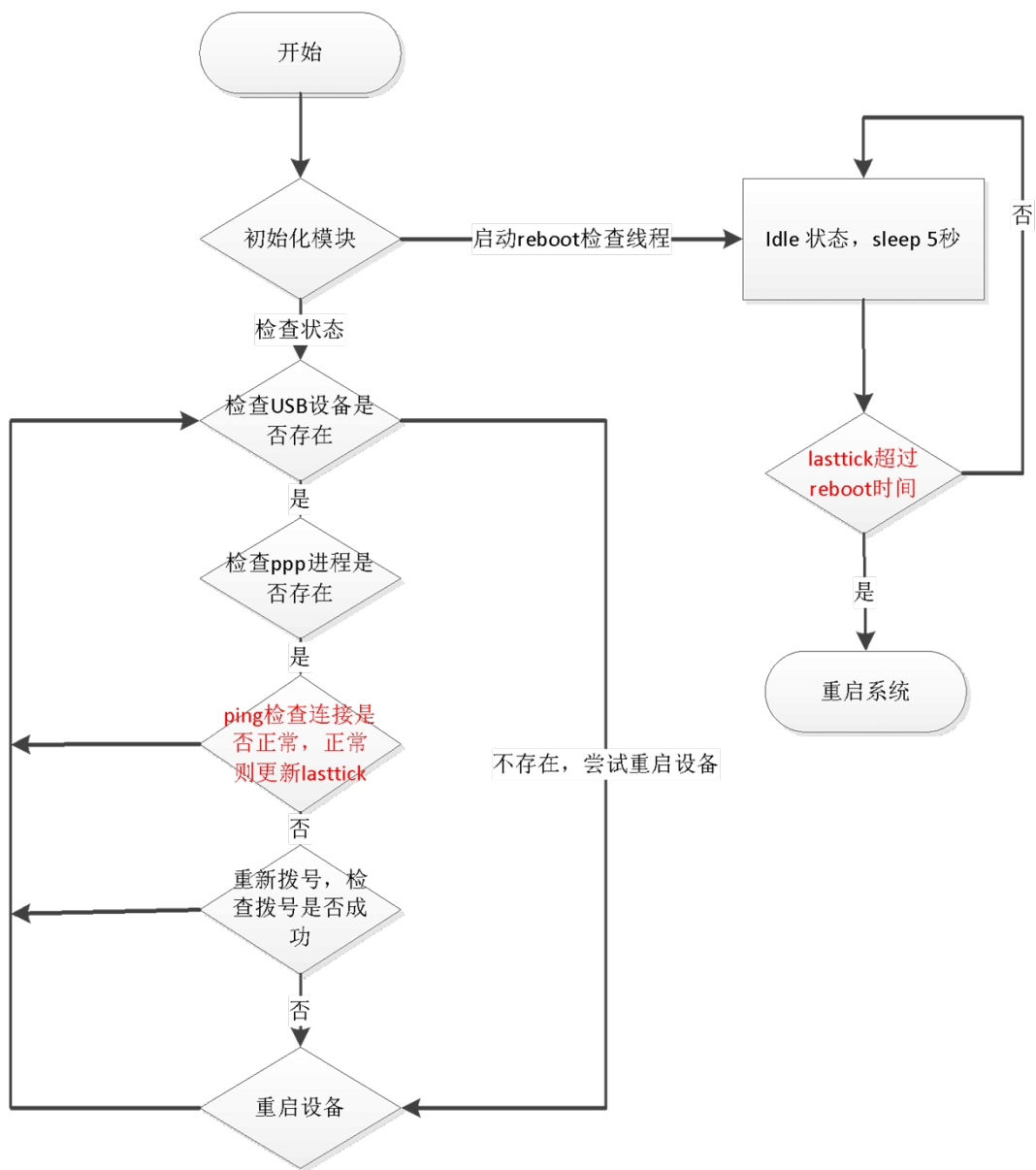
4.14 AdvWrielessCheckd Instructions [Supported by OS 2.6.0]

Note:

For predecessors of OS 2.6.0, please decompress the file to the system before using AdvWrielessCheckd. Get the program from AE and decompress it to /home/root/ directory.

```
# tar -C /home/root/ -zxvf advwirlesscheckd.tar.gz
```

4.14.1 Workflow



图表内容：

开始 Start

初始化模块 Initialize the module

检查状态 Check status

检查 USB 设备是否存在 Check if the USB device exists

是 Yes

检查 PPP 进程是否存在 Check if the PPP process exists

Ping 检查连接是否正常，正常则更新 lasttick Use ping command to check whether the connection is normal. If it is normal, update lasttick

否 No

重新拨号，检查拨号是否成功 Re-dial the number to check if the dialing is successful

否 No

重启设备 Restart the device

启动 Reboot 检查线程 Reboot to check thread

不存在，尝试重启设备 No USB device, try restarting the device

Idle 状态，sleep 5 秒 Idle state, sleep for 5 seconds

lasttick 超过 reboot 时间 lasttick exceeds the reboot time

是 Yes

重启系统 Restart the system

否 No

Currently ppp0 and fixed LAN network card can be checked.

4.14.2 Usage

4.14.2.1 Select the Correct Configuration Files

In predecessors of EdgeLink 2.7.0:

Since different cellular modules adopt different virtual serial ports, configuration files need to be selected according to the modules used. Copy SystemSetting.acr configuration file to the

/home/root/project directory based on the cellular modules used.

The default configuration parameters of each module are placed in the /home/root/project/cellular/ directory based on the module name, which can be directly copied to the system configuration file for use.

```
# cp /home/root/project/cellular/EC20CEFA-512-STD\{Quectel\}/SystemSetting.acr  
/home/root/project/
```

EdgeLink2.7.0 and its susccesors

Device information are stored in /home/root/project/CellularDeviceInfo.acr. No additional configuration is required.

4.14.2.2 Start the Testing Program

```
root@ecu1251:~# AdvWirelessCheckd
```

```
AdvWirelessCheck Aug  2 2019 build 15:21:28
```

```
open libDCTag.so failed
```

```
ERROR:libDCTag.so: cannot open shared object file: No such file or directory
```

```
open libwatchprocess.so failed
```

```
ERROR:libwatchprocess.so: cannot open shared object file: No such file or directory
```

```
TAGLINK_PATH:/home/root
```

```
Config file: /home/root/project/SystemSetting.acr
```

```
ADAM3600DS,ECU1251
```

```
ECU1051,ECU1251
```

```
Dual sim:0
```

```
open libDCTag.so failed
```

```
ERROR:libDCTag.so: cannot open shared object file: No such file or directory
```

```
load DCTag failed
```

```
connProcessThread,/home/root/bin/awc_3g.so++++
```

```
Load module /home/root/bin/awc_3g.so, netcard ppp0
```

```
ERROR: PPP link is not active on ppp0
```

```
killall: pppd: no process killed
```

```
switchTag=(null),switchType=None
```

```
smsOnly=0
```

```
ppp0,cb=80
```

```
ppp0,restartSystemSeconds=0
ppp0,dialUpCommand=wan.sh
ppp0,dialDownCommand=/etc/ppp/ppp-off;/usr/bin/killall pppd
ppp0,processName=pppd
ppp0,tyCommPort=Not Init
ppp0,tyConfigPort=Not Init
ppp0,tyCommPortNo=3
ppp0,tyConfigPortNo=2
ppp0,usbName=
ppp0,usbLable=Android_Android
ppp0,usbID=2c7c:0125
ppp0,usbBus=0
ppp0,operator=auto
ppp0,netmode=1
ppp0,checkmode=0
ppp0,sim1,operator=
ppp0,sim1,netmode=0
ppp0,sim2,operator=
ppp0,sim2,netmode=0
return:1
```

```
rebootCheckThread:last connect tick:15324
ERROR: PPP link is not active on ppp0
killall: pppd: no process killed
return:1,dual sim:0
```

```
setMode+++
setMode:file[/home/root/project/urat] not found
ppp0,cb=80
ppp0,restartSystemSeconds=0
ppp0,dialUpCommand=wan.sh
ppp0,dialDownCommand=/etc/ppp/ppp-off;/usr/bin/killall pppd
ppp0,processName=pppd
ppp0,tyCommPort=/dev/ttyUSB3           #The serial port is used for dialing and must
be identified
ppp0,tyConfigPort=/dev/ttyUSB2
ppp0,tyCommPortNo=3
ppp0,tyConfigPortNo=2
```


ppp0,usbName=
ppp0,usbLable=Android_Android
ppp0,usbID=2c7c:0125
ppp0,usbBus=2
ppp0,operator=auto
ppp0,netmode=1
ppp0,checkmode=0
ppp0,sim1,operator=
ppp0,sim1,netmode=0
ppp0,sim2,operator=
ppp0,sim2,netmode=0
n3g_init return:1

killall: GPSManager: no process killed
ERROR: PPP link is not active on ppp0
killall: pppd: no process killed
check_carrier:sim card found!
get_mno_info: scan operator name failed! buf = +COPS: 0,0,"JD Mobile",7

OK

#The information marked red below is the normal information read before dialing.

set_cops_mode: tty = /dev/ttyUSB3
get_mobile_mno: csq = 27
get_mno_info: operator code = 46001
get_mno_info: tty = /dev/ttyUSB3,mcc = 460,mnc = 01
find_provider_apn: found apn = 3gnet for mcc = 460, mnc = 01.
checkProvider: find apn = 3gnet, netmode = 1
checkProvider:mcc = 460, mnc = 01,ret = 0, wan.sh default
check_carrier:checkProvider return = 1
get_mno_info: operator code = 46001
get_mno_info: tty = /dev/ttyUSB3,mcc = 460,mnc = 01
switch to 3
connProcessThread,awc_3g.so:checkProcess failed
switch to 5
switch to 6
connProcessThread,awc_3g.so:redialUp[1/4]
ERROR: PPP link is not active on ppp0

```
killall: pppd: no process killed
rebootCheckThread:last connect tick:15324
killall: pppd: no process killed
rebootCheckThread:last connect tick:15324
rebootCheckThread:last connect tick:15324
switch to 0
switch to 0
rebootCheckThread:last connect tick:15324
rebootCheckThread:last connect tick:15324
```

4.14.2.3 Start the Tesing Program by Default at Startup

```
~# vi /etc/rc.local
```

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
export LD_LIBRARY_PATH=$ LD_LIBRARY_PATH :/home/root/lib/
/home/root/bin/AdvWirelessCheckd -d
exit 0
```

4.14.3 Description of Configuration File CellularDeviceInfo.acr [Supported by Successors of 2.7.0]

```
~# cat /home/root/project/CellularDeviceInfo.acr
```

```
<?xml version="1.0" encoding="utf-8"?>
<CellularDeviceInfo name="" description="">
  <Device deviceId="2cb7:0001" GPSType="none" GPSInterface="0" ATPortCount="2" ATPortInterface="2,4" DialType="ppp" deviceName="CU101-GL(UNICOM)" />
  <Device deviceId="12d1:15c1" GPSType="none" GPSInterface="0" ATPortCount="3" ATPortInterface="4,2,5" DialType="ppp" deviceName="ME909S(Huawei)" />
```

```

<Device deviceId="12d1:1c25" GPSType="none" GPSInterface="0" ATPortCount="3" ATPortInterface="4,2,5" DialType="ppp" deviceName="MU709S(Huawei)" />
<Device deviceId="12d1:1573" GPSType="none" GPSInterface="0" ATPortCount="3" ATPortInterface="4,2,6" DialType="ppp" deviceName="MU609(Huawei)" />
<Device deviceId="19d2:0199" GPSType="none" GPSInterface="0" ATPortCount="2" ATPortInterface="0,2" DialType="none" deviceName="ME3760(ZTE)" />
<Device deviceId="19d2:1476" GPSType="none" GPSInterface="0" ATPortCount="2" ATPortInterface="1,2" DialType="ppp" deviceName="ME3630(ZTE)" />
<Device deviceId="05c6:90b3" GPSType="embedded" GPSInterface="2" ATPortCount="1" ATPortInterface="3" DialType="none" deviceName="MDG100 RNDIS(usb0)" />
<Device deviceId="1546:01a7" GPSType="independ" GPSInterface="1" ATPortCount="0" ATPortInterface="0" DialType="none" deviceName="EWM-G108" />
<Device deviceId="2c7c:0296" GPSType="embedded" GPSInterface="1" ATPortCount="2" ATPortInterface="2,3" DialType="ppp" deviceName="BG96(Quectel)" />
</CellularDeviceInfo>

```

This file lists the information of all currently supported devices.

deviceId="2c7c:0296" Modem module's usbId.

GPSType="embedded" The parameter indicates whether gps is supported. GPSType has 3 values: none, embedded, independ.

GPSInterface="1" usb device's interface number

ATPortCount="2" The parameter indicates the number of virtual serial ports that can send AT command. At least 1 virtual serial port is required.

ATPortInterface="2,3" The parameter indicates the usb interface number that can send AT command. The first is used to query the module information, the second is used for ppp dialing, like Huawei series. Please put the dedicated ppp dialing in the second position.

DialType="ppp" Dial type. Currently DialType has two values, ppp and none.

deviceName="BG96(Quectel)" Module name for display only.

The information can be automatically generated by modescan, see *4.10 Cellular Information Query [Supported by 2.6.1 and its successors]*

4.14.4 Description of Configuration File [SystemSetting.acr]

For new modules, please refer to the demo configuration file (/home/root/project/cellular/demo/SystemSetting.acr) in the system.

4.14.4.1 Cellular Parameter <GPRS> [Predessors of 2.7.0]

```

<tDeviceConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

```

```

  <LAN>

```

```

    <GPRS othername2="telecom" isMutliSim="false" is4G="false" othername="unicom"
connection="true" othername1="cmnet">

```

```

      <ModuleName>EC20CEFA-512-STD(Quectel)</ModuleName>

```

```

<APN />
<UserName />
<Password />
<PINNumber />
<PhoneNumber />
<PingInterval>60</PingInterval>
<RetryCount>0</RetryCount>
<NetworkMode>4G</NetworkMode>
<Operator>auto</Operator>
<deviceName>EC20CEFA-512-STD(Quectel)</deviceName>
<Lable>Android_Android</Lable>
<CommunicationPort>3</CommunicationPort>
<ConfigPort>2</ConfigPort>
<USBDeviceID>2c7c:0125</USBDeviceID>
<SMSPort>2</SMSPort>
<USBDeviceName />
<ConnectionCheckType>0</ConnectionCheckType>
<MaxSilenceTime>1</MaxSilenceTime>
<RebootOnFailureTime>0</RebootOnFailureTime>
<NetworkInterface>ppp0</NetworkInterface>
<DNSList_ipv4 isAutomatically="true" />
<DNSList_ipv6 isAutomatically="true" />
<EnableSim>>false</EnableSim>
<MutliSim switchType="None" masterSIM="1">^M
  <Sims name="sim1" enable="true">^M
    <Operator>auto</Operator>^M
    <NetworkMode />^M
    <APN />^M
    <UserName />^M
    <Password />^M
    <PhoneNumber />^M
    <AuthMethod />^M
    <isAuthentication>>false</isAuthentication>^M
  </Sims>^M
  <Sims name="sim2" enable="true">^M
    <Operator>auto</Operator>^M
    <NetworkMode />^M
    <APN />^M

```

```

        <UserName />^M
        <Password />^M
        <PhoneNumber />^M
        <AuthMethod />^M
        <isAuthentication>>false</isAuthentication>^M
    </Sims>^M
</MutliSim>
<AuthMethod />
<isAuthentication>>false</isAuthentication>
</GPRS>
</LAN>
</tDeviceConfig>

```

For the definition of configuration parameters of Cellular, see 4.14.4.5.2-4.14.4.5.5 *Parameter Explanation*.

4.14.4.2 Configuration Parameters of Cellular <GPRS> [2.7.0 and Its Successors]

```

<tDeviceConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <LAN>
        <GPRS othername2="telecom" isMutliSim="false" is4G="false" othername="unicom"
connection="true" othername1="cmnet">
            <ModuleName>Auto</ModuleName>
            <APN />
            <UserName />
            <Password />
            <PINNumber />
            <PhoneNumber />
            <PingInterval>60</PingInterval>
            <RetryCount>0</RetryCount>
            <NetworkMode>4G</NetworkMode>
            <Operator>auto</Operator>
            <deviceName>EC20CEFA-512-STD(Quectel)</deviceName>
            <Lable>Android_Android</Lable>
            <CommunicationPort>3</CommunicationPort>
            <ConfigPort>2</ConfigPort>
            <USBDeviceID>2c7c:0125</USBDeviceID>

```

```

<SMSPort>2</SMSPort>
<USBDeviceName />
<ConnectionCheckType>0</ConnectionCheckType>
<MaxSilenceTime>1</MaxSilenceTime>
<RebootOnFailureTime>0</RebootOnFailureTime>
<NetworkInterface>ppp0</NetworkInterface>
<DNSList_ipv4 isAutomatically="true" />
<DNSList_ipv6 isAutomatically="true" />
<EnableSim>>false</EnableSim>
<MutliSim switchType="None" masterSIM="1">^M
  <Sims name="sim1" enable="true">^M
    <Operator>auto</Operator>^M
    <NetworkMode />^M
    <APN />^M
    <UserName />^M
    <Password />^M
    <PhoneNumber />^M
    <AuthMethod />^M
    <isAuthentication>>false</isAuthentication>^M
  </Sims>^M
  <Sims name="sim2" enable="true">^M
    <Operator>auto</Operator>^M
    <NetworkMode />^M
    <APN />^M
    <UserName />^M
    <Password />^M
    <PhoneNumber />^M
    <AuthMethod />^M
    <isAuthentication>>false</isAuthentication>^M
  </Sims>^M
</MutliSim>
<AuthMethod />
<isAuthentication>>false</isAuthentication>
</GPRS>
</LAN>
</tDeviceConfig>

```

For the definition of configuration parameters of Cellular, see 4.14.4.5.2-4.14.4.5.5

Parameter Explanation.

4.14.4.3 Configuration Parameter < ChildLAN > of Fixed Network Card

```
<tDeviceConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <LAN>
    <ChildLAN name="eth0" isDHCPv6="true" isDHCP="true">^M
      <DNSList_ipv4 isAutomatically="true" />^M
      <DNSList_ipv6 isAutomatically="true" />^M
      <ConnectionCheckType>0</ConnectionCheckType>^M
      <PingInterval>60</PingInterval>^M
      <PingURL>www.badiu.com</PingURL>^M
      <MaxSilenceTime>1</MaxSilenceTime>^M
      <RebootOnFailureTime>0</RebootOnFailureTime>^M
    </ChildLAN>
  </LAN>
</tDeviceConfig>
```

Configure network card name name="ethx" in the ChildLAN node based on *4.6 Configure a Fixed Network Card IP*. Enable the feature in the configuration file after making sure that the network card can be enabled using #ifup ethX command.

4.14.4.4 WIFI Configuration < WiFi >

```
<tDeviceConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <LAN>
    <WiFi isDHCPv6="true" isDHCP="true" BSSID="" enableBSSID="false"
NetworkCard="wlan0" Enable="true" Security="Open"
PassWord="eXqGgy55tYmGuv1Jx8ZRGg==">
      <DNSList_ipv4 isAutomatically="true" />
      <DNSList_ipv6 isAutomatically="true" />
      <ConnectionCheckType>0</ConnectionCheckType>
      <WiFiAPMode isDHCPv6="false" isDHCP="true" enable="false" max_num_sta="0">
        <DNSList_ipv4 isAutomatically="false" />
        <DNSList_ipv6 isAutomatically="false" />
        <ConnectionCheckType>0</ConnectionCheckType>
      </WiFiAPMode>
    </LAN>
</tDeviceConfig>
```

```
</WiFi>
</LAN>
</tDeviceConfig>
```

Configure WiFi in the ChildLAN node based on *4.7 WiFi Configuration* to enable the check. Enable the feature in the configuration file after making sure that the number can be dialed using `#wlan.sh up` command.

4.14.4.5 Description of Configuration Parameter

4.14.4.5.1 ConnectionCheckType

```
<ConnectionCheckType>0</ConnectionCheckType>
```

The parameter is used to check the connection type.

0: disable, which means the connection type is not checked.

1: ping, ping to confirm whether the connection address is normal.

2: traffic, check the connection by examining whether the data sent and received by the network card has changed. This method is not applicable for WIFI.

4.14.4.5.2 PingInterval

```
<PingInterval>60</PingInterval>
```

PingInterval refers to the time interval between two pings, its unit is second, and the default interval is 60 seconds.

PingInterval is valid when `<ConnectionCheckType>` is 1.

4.14.4.5.3 PingURL

```
<PingURL>www.baidu.com</PingURL>
```

PingURL is used to ping host address, which can be a domain name or an IP address. Up to three IP addresses can be pinged. When there is more than one IP address, if one address can be pinged, this means the network is connected.

PingURL is valid when the value of ConnectionCheckType is 1.

4.14.4.5.4 Label [Discarded by Successors of 2.7.0]

```
<Label>Android_Android</Label>
```

The parameter is used to automatically find the parameters used by USB's serial port.

```
root@adam3600:~# ls -l /dev/serial/by-id/
```

```
total 0
```

```
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if00-port0 -> ../ttyUSB0
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if01-port0 -> ../ttyUSB1
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if02-port0 -> ../ttyUSB2
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if03-port0 -> ../ttyUSB3
```



```
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if04-port0 -> ../../ttyUSB4
```

<Label> is the Label name of the serial port device. For example, for EC20's full name "usb-Android_Android-if00-port0", please enter the shared constant part "Android_Android" after usb- as the label name.

4.14.4.5.5 RebootOnFailureTime

<RebootOnFailureTime>0</RebootOnFailureTime>

The function is disabled when the value is 0 by default and enabled when the value is greater than 0 and less than 24. The parameter's unit is an hour, and decimals can be used.

When the value is less than 0.2, the time is counted as the minimum value 0.2 and the maximum value is 24.

For example, if the value is 0.2, when the ping fails twice with a time interval of 12 minutes, the system will reboot.

4.14.4.5.6 CommunicationPort [Discarded by Successors of 2.7.0]

<CommunicationPort>3</CommunicationPort>

```
root@adam3600:~# ls -l /dev/serial/by-id/
```

```
total 0
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if00-port0 -> ../../ttyUSB0
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if01-port0 -> ../../ttyUSB1
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if02-port0 -> ../../ttyUSB2
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if03-port0 -> ../../ttyUSB3
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if04-port0 -> ../../ttyUSB4
```

CommunicationPort is the sequence number of serial port used by PPP for dialing. The sequence number is counted from 0. When its value is 3, which means that the serial port corresponding to usb-Android_Android-if03-port0 is used.

4.14.4.5.7 ConfigPort [Discarded by Successors of 2.7.0]

<ConfigPort>2</ConfigPort>

```
root@adam3600:~# ls -l /dev/serial/by-id/
```

```
total 0
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if00-port0 -> ../../ttyUSB0
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if01-port0 -> ../../ttyUSB1
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if02-port0 -> ../../ttyUSB2
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if03-port0 -> ../../ttyUSB3
lrwxrwxrwx 1 root root 13 Jan 30 11:38 usb-Android_Android-if04-port0 -> ../../ttyUSB4
```

ConfigPort is the serial port for reading module status by the AdvWirlessCheckd program. The ConfigPort is counted from 0. When its value is 2, which means that the serial port corresponding to usb-Android_Android-if02-port0 is used.

4.14.4.5.8 USBDeviceID [Discarded by Successors of 2.7.0]

```
<USBDeviceID>2c7c:0125</USBDeviceID>
```

```
root@adam3600:~# lsusb
```

```
Bus 001 Device 002: ID 0424:2512 Standard Microsystems Corp. USB 2.0 Hub
```

```
Bus 002 Device 002: ID 2c7c:0125
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

```
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Enter device's USB Device ID in USBDeviceID.

4.14.4.6 DUAL SIM< MutliSim >

The parameter is only valid on dual SIM card devices such as ADAM3600DS and ECU1051.

```
<MutliSim switchType="None" masterSIM="1">^M
```

```
<Sims name="sim1" enable="true">^M
```

```
<Operator>auto</Operator>^M
```

```
<NetworkMode />^M
```

```
<APN />^M
```

```
<UserName />^M
```

```
<Password />^M
```

```
<PhoneNumber />^M
```

```
<AuthMethod />^M
```

```
<isAuthentication>>false</isAuthentication>^M
```

```
</Sims>^M
```

```
<Sims name="sim2" enable="true">^M
```

```
<Operator>auto</Operator>^M
```

```
<NetworkMode />^M
```

```
<APN />^M
```

```
<UserName />^M
```

```
<Password />^M
```

```
<PhoneNumber />^M
```

```
<AuthMethod />^M
```

```
<isAuthentication>>false</isAuthentication>^M
```

```
</Sims>^M
</MutliSim>
```

4.14.4.6.1 Use SIM Card First

```
masterSIM="1"
```

1: When there are SIM cards in both SIM card slots, SIM1 shall be used first.

2: When there are SIM cards in both SIM card slots, SIM2 shall be used first.

If there is only one SIM card in the two SIM card slots, the parameter is invalid.

4.14.4.6.2 Ways to Switch SIM Cards

```
switchType="None"
```

None: Automatic mode. Traversal search will be conducted for the two slots at startup.

ConnectCheck: use ConnectCheck with a value of 1 or 2 to confirm whether to switch the SIM card.

When the connection check fails, the number will be redialed. If the dialing is successful, SIM card will not be switched. Otherwise, it will be switched.

4.15 OpenVPN (Gateway as Client)

1) Use cert/key

- a) Copy ca.crt, client.ovpn, client.crt, client.key, ta.key to /home/root/ovc directory.
- b) Modify the configuration file.

```
# vi /home/root/ovc/client.ovpn
```

```
client
```

```
;dev tap
```

```
dev tun
```

```
;dev-node MyTap
```

```
proto tcp
```

```
;proto udp
```

```
remote 172.21.67.33 1194 #The server's IP address (or domain name address)
and port number
```

```
;remote my-server-2 1194
```

```
;remote-random
```

```
resolv-retry infinite
```

```
nobind
```

```
;user nobody
```

```
;group nobody
```

```
persist-key
persist-tun
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]
;mute-replay-warnings
ca /home/root/ovc/ca.crt
cert /home/root/ovc/client.crt
key /home/root/ovc/client.key
;auth-user-pass /home/root/ovc/pass.txt
;remote-cert-tls server
tls-auth /home/root/ovc/ta.key 1
cipher BF-CBC
comp-lzo
verb 3
;mute 20
```

c) Enable the client.

Note: The time needs to be consistent with the server.

Run in the foreground #openvpn --config /home/root/ovc/client.ovpn

Run in the background #openvpn --daemon --config /home/root/ovc/client.ovpn # Run in the background with --daemon added

2) Use Username/Password

```
#vi /home/root/ovc/client.ovpn
```

```
ca /home/root/ovc/ca.crt
;cert /home/root/ovc/client.crt
;key /home/root/ovc/client.key
auth-user-pass /home/root/ovc/pass.txt
```

```
# vi /home/root/ovc/pass.txt
```

```
user1
12345678
```

3) Configure as auth-name

Add the following content to the main configuration file of open*** service. If **client-cert-not-required** is added, it means that the username and password are needed to verify the login. Otherwise, a certificate is also needed!

```
# vi /home/root/ovs/server.ovpn
auth-user-pass-verify /home/root/checkpsw.sh via-env
client-cert-not-required
username-as-common-name
script-security 3
```

```
#vi /home/root/ovs/checkpsw.sh
```

```
#!/bin/sh
```

```
#####
```

```
# checkpsw.sh (C) 2004 Mathias Sundman <mathias@openvpn.se>
```

```
#
```

```
# This script will authenticate OpenVPN users against
```

```
# a plain text file. The passfile should simply contain
```

```
# one row per user with the username first followed by
```

```
# one or more space(s) or tab(s) and then the password.
```

```
PASSFILE="/home/root/ovs/psw"
```

```
LOG_FILE="/var/log/openvpn-password.log"
```

```
TIME_STAMP=`date "+%Y-%m-%d %T"`
```

```
#####
```

```
if [ ! -r "${PASSFILE}" ]; then
```

```
    echo "${TIME_STAMP}: Could not open password file \"${PASSFILE}\" for reading." >> ${LOG_FILE}
```

```
    exit 1
```

```
fi
```

```
CORRECT_PASSWORD=`awk '!/^;/&&!/^#&&$1=="${username}"{print $2;exit}' ${PASSFILE}`
```

```
if [ "${CORRECT_PASSWORD}" = "" ]; then
```

```
    echo "${TIME_STAMP}: User does not exist: username=\"${username}\", password=\"${password}\"." >> ${LOG_FILE}
```

```
    exit 1
```

```
fi
```

```

if [ "${password}" = "${CORRECT_PASSWORD}" ]; then
    echo "${TIME_STAMP}: Successful authentication: username=\"${username}\"." >> ${LOG_FILE}
    exit 0
fi

echo "${TIME_STAMP}: Incorrect password: username=\"${username}\", password=\"${password}\"." >> ${LOG_FILE}
exit 1

```

```
#vi /home/root/ovs/psw
```

```
#add username password in one line and separate with space blank
```

```
user1 12345678
```

```
#vi /home/root/ovc/client.ovpn
```

```
ca /home/root/ovc/ca.crt
```

```
;cert /home/root/ovc/client.crt
```

```
;key /home/root/ovc/client.key
```

```
auth-user-pass /home/root/ovc/pass.txt
```

```
# vi /home/root/ovc/pass.txt
```

```
user1
```

```
12345678
```

4) TLS-auth Communication

Building ta.key

Building ta.key, this file is secret and nonessential.

Generate with: #**openvpn --genkey --secret ta.key**

The server and each client must have a copy of this key. The second parameter should be '0' on the server and '1' on the clients.

```
#vi /home/root/ovs/server.ovpn
```

```
tls-auth ta.key 0
```

```
#vi /home/root/ovc/client.ovpn
```

tls-auth ta.key 1

5) TAP and TUN Modes

Modify dev in the configuration file to change the working mode.

Enable method.

TUN mode

```
;dev tap
```

```
dev tun
```

TAP mode

```
dev tap
```

```
;dev tun
```

6) Add Execution Script

Add the following commands in client.ovpn script, which will be used when the VPN is enabled or disabled.

script-security 2

up /home/root/ovc/vpn-start

down /home/root/ovc/vpn-stop

```
#vi /home/root/ovc/vpn-start
```

```
#!/bin/sh
```

```
echo $dev
```

```
#vi /home/root/ovc/vpn-stop
```

```
#!/bin/sh
```

```
echo $dev
```

Add executable permission

```
#chmod a+x /home/root/ovc/vpn-stop
```

```
#chmod a+x /home/root/ovc/vpn-start
```

Note:

1. script-security 2 must be specified to enable script security level.
2. The script must have header `#!/bin/sh`, otherwise it will prompt that the execution fails.

4.16 OpenVPN (Gateway as the Server)

4.16.1 Install OpenVPN Server in Windows

See attached document "ADAM-3600-C2GA1E OpenVPN Configuration SOP_V1.0_20160407.docx"

4.16.2 Generate a Root Certificate

As the root certificate will be used when all openvpn nodes generate relevant certificates, please keep it properly.

To generate the root certificate ca, Fully Qualified Domain Name must be filled in Common Name during the generation process (the IP address can be used in the test as long as the certificate name is unique. If the full domain name is not used, errors may be reported in some applications.)

```
/home/root/ovs/# openssl genrsa -out ca.key 2048
```

```
/home/root/ovs/# openssl req -new -key ca.key -out ca.csr -subj /CN=OpenVPN-CA/
```

```
/home/root/ovs/# openssl x509 -req -in ca.csr -out ca.crt -signkey ca.key -days 365
```

Check the generated file

```
root@adam3600:~/ovs# ls
```

```
ca.crt      ca.key      dh1024.pem
```

Check the information of the two files

```
# openssl rsa -noout -text -in ca.key
```

```
#openssl x509 -noout -text -in ca.crt
```

4.16.3 Generate cert/key for Server

- a) This step needs ca.crt,ca.key to generate keys and certificates.

Generate keys

```
/home/root/ovs #openssl genrsa -out server.key 2048
```

Certification and registration

```
/home/root/ovs #openssl req -out server.csr -key server.key -new -subj /CN=OpenVPN/
```

Generate certificates

```
/home/root/ovs #openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial  
-out server.crt -days 365
```



```
/home/root/ovs #openssl dhparam -out dh1024.pem 1024
```

4.16.4 Generate cert/key for Client

- b) This step needs ca.crt,ca.key to generate keys and certificates.

Generate password

```
/home/root/ovs #openssl genrsa -out client.key 2048
```

Certification and registration

```
/home/root/ovs #openssl req -out client.csr -key client.key -new -subj /CN=OpenVPN-Client/
```

Generate certificates. ca.crt and ca.key need to be the same as the one used by the Server.

```
/home/root/ovs #openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out  
client.crt -days 365
```

4.16.5 Use Cert/Key to Configure Server

- a) The server uses cert/key for verification.

```
#vi /home/root/ovs/server.ovpn
```

```
port 1194
```

```
proto tcp
```

```
dev tun
```

```
topology subnet
```

```
ca /home/root/ovs/ca.crt
```

```
cert /home/root/ovs/server.crt
```

```
key /home/root/ovs/server.key
```

```
dh /home/root/ovs/dh1024.pem
```

```
server 10.8.0.0 255.255.255.0
```

```
ifconfig-pool-persist /home/root/ovs/ipp.txt
```

```
keepalive 10 120
```

```
comp-lzo
```

```
persist-key
```

```
persist-tun
```

```
status /home/root/ovs/openvpn-status.log
```

```
verb 3
```

```
duplicate-cn
```

```
client-to-client
```

- b) Start the server.

```
# openvpn --config /home/root/ovs/server.ovpn
# openvpn --daemon --config /home/root/ovs/server.ovpn #The command will run in the
background when --daemon is added.
```

4.16.6 Configure Client with Cert/Key

- c) Copy the certificate from the server to the client machine.

```
/home/root/ovc #ls
```

```
ca.crt client.crt client.key
```

- d) Modify the configuration file.

```
# vi /home/root/ovc/client.ovpn
```

```
client
```

```
;dev tap
```

```
dev tun
```

```
;dev-node MyTap
```

```
proto tcp
```

```
;proto udp
```

```
remote 172.21.67.33 1194 #The server IP address (or domain name address) and
port number
```

```
;remote my-server-2 1194
```

```
;remote-random
```

```
resolv-retry infinite
```

```
nobind
```

```
;user nobody
```

```
;group nobody
```

```
persist-key
```

```
persist-tun
```

```
;http-proxy-retry # retry on connection failures
```

```
;http-proxy [proxy server] [proxy port #]
```

```
;mute-replay-warnings
```

```
ca /home/root/ovc/ca.crt
```

```
cert /home/root/ovc/client.crt
```

```
key /home/root/ovc/client.key
```

```
;auth-user-pass /home/root/ovc/pass.txt
```

```
;remote-cert-tls server
```

```
;tls-auth /home/root/ovc/ta.key 1
```

```
;cipher BF-CBC
```

```
comp-lzo
```

```
verb 3
;mute 20
```

e) Enable the client.

Check whether the local time is the same as the server time.

```
#date
#openvpn --config /home/root/ovc/client.ovpn
#openvpn --daemon --config /home/root/ovc/client.ovpn #The command will run in the
background after --daemon is added
```

4.16.7 Configure as auth-name

Add the following content to the main configuration file of open*** service. If **client-cert-not-required** is added, which means that the username and password are needed to verify the login. Otherwise, a certificate is also needed!

```
# vi /home/root/ovs/server.ovpn
auth-user-pass-verify /home/root/checkpsw.sh via-env
client-cert-not-required
username-as-common-name
script-security 3
```

```
#vi /home/root/ovs/checkpsw.sh
```

```
#!/bin/sh
```

```
#####
```

```
# checkpsw.sh (C) 2004 Mathias Sundman <mathias@openvpn.se>
```

```
#
```

```
# This script will authenticate OpenVPN users against
```

```
# a plain text file. The passfile should simply contain
```

```
# one row per user with the username first followed by
```

```
# one or more space(s) or tab(s) and then the password.
```

```
PASSFILE="/home/root/ovs/psw"
```

```
LOG_FILE="/var/log/openvpn-password.log"
```

```
TIME_STAMP=`date "+%Y-%m-%d %T"`
```

```
#####
```

```
if [ ! -r "${PASSFILE}" ]; then
```

```
    echo "${TIME_STAMP}: Could not open password file \"${PASSFILE}\" for reading." >> ${LOG_FILE}
```

```

    exit 1
fi

CORRECT_PASSWORD=`awk '!/^:/{&&#1=="${username}"{print $2;exit}' ${PASSFILE}`

if [ "${CORRECT_PASSWORD}" = "" ]; then
    echo "${TIME_STAMP}: User does not exist: username=\"${username}\", password=\"${password}\" >> ${LOG_FILE}
    exit 1
fi

if [ "${password}" = "${CORRECT_PASSWORD}" ]; then
    echo "${TIME_STAMP}: Successful authentication: username=\"${username}\", password=\"${password}\" >> ${LOG_FILE}
    exit 0
fi

echo "${TIME_STAMP}: Incorrect password: username=\"${username}\", password=\"${password}\" >> ${LOG_FILE}
exit 1

```

#vi /home/root/ovs/psw

#add username password in one line and separate with space blank

user1 12345678

#vi /home/root/ovc/client.ovpn

ca /home/root/ovc/ca.crt

;cert /home/root/ovc/client.crt

;key /home/root/ovc/client.key

auth-user-pass /home/root/ovc/pass.txt

vi /home/root/ovc/pass.txt

user1

12345678

4.16.8 TLS-auth Communication

Building ta.key

Building ta.key, this file is secret and nonessential.

Generate with: `#openvpn --genkey --secret ta.key`

The server and each client must have a copy of this key. The second parameter should be '0' on the server and '1' on the clients.

```
#vi /home/root/ovs/server.ovpn
```

```
tls-auth ta.key 0
```

```
#vi /home/root/ovc/client.ovpn
```

```
tls-auth ta.key 1
```

4.16.9 TAP and TUN Modes

Modify dev in the configuration file to change the working mode.

Enable the method

TUN mode

```
;dev tap
```

```
dev tun
```

TAP mode

```
dev tap
```

```
;dev tun
```

A TAP device is a virtual ethernet adapter, while a TUN device is a virtual point-to-point IP link.

You cannot mix `--dev tun` and `--dev tap` on different ends of the connection. Use one or the other consistently.

4.16.10 Add Executable Script

Add the following commands in client.ovpn script. The commands can be used when the VPN is enabled or disabled.

```
script-security 2
```

```
up /home/root/ovc/vpn-start
```

```
down /home/root/ovc/vpn-stop
```

```
#vi /home/root/ovc/vpn-start
```

```
#!/bin/sh
```

```
echo $dev
```

```
#vi /home/root/ovc/vpn-stop
```

```
#!/bin/sh
```

```
echo $dev
```

Add executable permissions

```
#chmod a+x /home/root/ovc/vpn-stop
```

```
#chmod a+x /home/root/ovc/vpn-start
```

Note:

1. script-security 2 must be specified to enable script security level.
2. The script must have header `#!/bin/sh`, otherwise it will prompt that the execution fails.

4.17 Routing Table Configuration

4.17.1 View the Current Routing Table

```
root@adam3600:~# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.21.67.1	0.0.0.0	UG	1	0	0	eth1
172.21.67.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1

4.17.2 Add a Routing Table

```
root@adam3600:~# route add default eth0
```

```
root@adam3600:~# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	0.0.0.0	0.0.0.0	U	0	0	0	eth0
0.0.0.0	172.21.67.1	0.0.0.0	UG	1	0	0	eth1
172.21.67.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1

4.17.3 Delete a Routing Table

```
root@adam3600:~# route del default eth0
```

```
root@adam3600:~# route -n
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.21.67.1	0.0.0.0	UG	1	0	0	eth1
172.21.67.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1

```
root@adam3600:~#
```

4.17.4 Add a Gateway

```
root@adam3600:~# route -n
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.21.67.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1

```
root@adam3600:~# route add default gw 172.21.67.1 dev eth1
```

```
root@adam3600:~# route -n
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.21.67.1	0.0.0.0	UG	0	0	0	eth1
172.21.67.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1

```
root@adam3600:~#
```

4.18 Check USB Device

```
root@adam3600:~# lsusb
```

```
Bus 001 Device 002: ID 0424:2512 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
```

```
root@adam3600:~# lsusb -t
```

```
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=musb-hdrc/1p, 480M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=musb-hdrc/1p, 480M
   |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/2p, 480M
      |__ Port 2: Dev 3, If 0, Class=Vendor Specific Class, Driver=rt2800usb, 480M
```

```
root@adam3600:~#
```

4.19 Enable the FTP Service

```
root@adam3600:~# vsftpd /etc/vsftpd.conf &
```

```
root@adam3600:~# netstat -atn
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:41100	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:21	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:443	0.0.0.0:*	LISTEN
tcp	0	0	:::6001	:::*	LISTEN
tcp	0	0	:::22	:::*	LISTEN
tcp	0	0	:::23	:::*	LISTEN
tcp	0	0	:::504	:::*	LISTEN
tcp	0	0	:::7001	:::*	LISTEN

4.19.1 Modify Listening Ports

Add `listen_port` at the end of the configuration file to modify listening ports.

```
#vi /etc/vsftpd.conf
```

```
# When "listen" directive is enabled, vsftpd runs in standalone mode and
```

```
# listens on IPv4 sockets. This directive cannot be used in conjunction
```

```
# with the listen_ipv6 directive.
```

```
listen=YES
```

```
listen_port=1080
```

Enable the service after modifying the parameter.

```
# netstat -atnp
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN	1564/lighttpd
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	740/dropbear
tcp	0	0	0.0.0.0:1080	0.0.0.0:*	LISTEN	7591/vsftpd
tcp	0	0	0.0.0.0:443	0.0.0.0:*	LISTEN	1564/lighttpd

4.20 Enable telnet Service

The service is no longer used since telnet's communication data is not encrypted. You can use ssh service to replace it for remote login maintenance.

```
root@adam3600:~# /usr/sbin/telnetd
```



```
root@adam3600:~# netstat -atn
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:21	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	64	172.21.67.25:22	172.21.67.89:59892	ESTABLISHED
tcp	0	0	:::6001	:::*	LISTEN
tcp	0	0	:::22	:::*	LISTEN
tcp	0	0	:::23	:::*	LISTEN
tcp	0	0	:::504	:::*	LISTEN
tcp	0	0	:::7001	:::*	LISTEN

4.20.1 Modify Listening Ports

```
# telnetd -h
```

```
telnetd: invalid option -- 'h'
```

```
BusyBox v1.28.4 (2018-09-18 09:10:21 CST) multi-call binary.
```

```
Usage: telnetd [OPTIONS]
```

Handle incoming telnet connections

-l LOGIN	Exec LOGIN on connect
-f ISSUE_FILE	Display ISSUE_FILE instead of /etc/issue
-K	Close connection as soon as login exits (normally wait until all programs close slave pty)
-p PORT	Port to listen on
-b ADDR[:PORT]	Address to bind to
-F	Run in foreground
-i	Inetd mode
-w SEC	Inetd 'wait' mode, linger time SEC
-S	Log to syslog (implied by -i or without -F and -w)

```
#!/usr/sbin/telnetd -p 1090
```

```
# netstat -atnp
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
-------	--------	--------	---------------	-----------------	-------	------------------

```

tcp      0      0 0.0.0.0:80          0.0.0.0:*          LISTEN   1564/lighttpd
tcp      0      0 0.0.0.0:22          0.0.0.0:*          LISTEN   740/dropbear
tcp      0      0 0.0.0.0:1080        0.0.0.0:*          LISTEN   7591/vsftpd
tcp      0      0 0.0.0.0:443         0.0.0.0:*          LISTEN   1564/lighttpd
tcp      0      0 0 :::1090           :::*                LISTEN   8016/telnetd

```

4.21 Enable SSH Service

```
root@adam3600:~# /etc/init.d/dropbear start
```

```
root@adam3600:~# netstat -atn
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:41100	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:21	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:443	0.0.0.0:*	LISTEN
tcp	0	0	:::6001	:::*	LISTEN
tcp	0	0	:::22	:::*	LISTEN
tcp	0	0	:::23	:::*	LISTEN
tcp	0	0	:::504	:::*	LISTEN
tcp	0	0	:::7001	:::*	LISTEN

4.21.1 Modify Listening Ports

```
#vi /etc/init.d/dropbear
```

```
#!/bin/sh
```

```
### BEGIN INIT INFO
```

```
# Provides:          sshd
```

```
# Required-Start:    $remote_fs $syslog $networking
```

```
# Required-Stop:     $remote_fs $syslog
```

```
# Default-Start:     2 3 4 5
```

```
# Default-Stop:      1
```

```
# Short-Description: Dropbear Secure Shell server
```

```
### END INIT INFO
```

```
#
```

```
# Do not configure this file. Edit /etc/default/dropbear instead!
```

```
#  
  
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin  
DAEMON=/usr/sbin/dropbear  
NAME=dropbear  
DESC="Dropbear SSH server"  
  
DROPBEAR_PORT=22  
DROPBEAR_EXTRA_ARGS=  
NO_START=0
```

```
set -e
```

Change DROPBEAR_PORT to the new port number you need, and then restart the service.

4.22 DNS Configuration

The DNS configuration file is /etc/resolv.conf and its content is as below:

```
# cat /etc/resolv.conf  
nameserver 8.8.8.8
```

resolv.conf has four main keywords:

```
nameserver # The IP address of the DNS server  
domain     # The local domain name  
search     #The search list of domain names  
sortlist   # Sort out the domain names returned
```

Generally, you only need to set nameserver.

Since the file needs frequent reading and writing operations when obtaining IP address from the network card, the default soft link in the system is /var/run/resolv.conf.

```
# ll /etc/resolv.conf  
lrwxrwxrwx    1 root root 20 Jun 23 23:54 /etc/resolv.conf -> /var/run/resolv.conf
```

If the default DNS needs to be used at startup, you can choose one of the following three methods as needed.

4.22.1 /etc/rc.local

```
#vi /etc/resolv.conf
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
```

```
echo "nameserver 114.114.114.114" >> /etc/resolv.conf
exit 0
```

4.22.2 Recreate a Soft Link

```
# cp /etc/resolv.conf /home/etc/
# ln -sf /home/etc/resolv.conf /etc/resolv.conf
# ll /etc/resolv.conf
lrwxrwxrwx    1 root    root 21 Jun 29 15:56 /etc/resolv.conf -> /home/etc/resolv.conf
```

4.22.3 Remove a Soft Link

```
# cp /etc/resolv.conf /etc/resolv.conf.bak
#cp /etc/resolv.conf.bak /etc/resolv.conf
```

Since the system is read-only by default, after the soft link is removed, the new DNS may not be written in after cellular and wifi dialing or dialing through a fixed network card.

4.23 Firewall Configuration

4.23.1 View the Current Status of the Firewall

```
root@adam3600:~# iptables -L -n -v
```

```
Chain INPUT (policy DROP 4848 packets, 402K bytes)
```

pkts	bytes	target	prot	opt	in	out	source	destination
0	0	ACCEPT	tcp	--	eth0	*	172.0.0.0/8	0.0.0.0/0

```
tcp dpt:345
```

```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
```

pkts	bytes	target	prot	opt	in	out	source	destination
------	-------	--------	------	-----	----	-----	--------	-------------

```
Chain OUTPUT (policy DROP 0 packets, 0 bytes)
```

pkts	bytes	target	prot	opt	in	out	source	destination
0	0	DROP	icmp	--	*	*	0.0.0.0/0	0.0.0.0/0

```
state INVALID
```

4.23.2 Set up a Whitelist

```
root@adam3600:~# vi /home/sysuser/port_wihite.lst
```

```
# port white list
```

```
443|tcp|all
```

```
345|tcp|eth0|172.0.0.0/8
```

tcp is a protocol. Usually tcp or udp is adopted.

The format is as below:

Set up parameters, separated by "|", for each port in each line

6001 is a port number

tcp is a protocol, usually tcp or udp

all means all network cards. You can also set a single network card, such as eth0, eth1.

172.0.0.0/8 is the network segment. You can only access the IP address of the network segment.

The format of the blacklist is the same as that of the whitelist, except that relevant ports are disabled from accessing the local machine.

Since the iptables command needs several kernel modules to work properly, it is recommended that the shell script we have prepared shall be used to enable the function.

The /usr/bin/firewall.sh script is handy for dealing with dependency problems.

4.23.3 Enable the Firewall

```
#!/usr/bin/firewall.sh /home/sysuser
```

The program will search for the configuration files in the /home/sysuser directory. port_white.lst is the whitelist and port_black.lst is the blacklist.

4.23.4 Disable the Firewall

```
#!/usr/bin/firewall.sh stop    Disable the firewall
```

4.24 Web Server (lighttpd) Configuration

4.24.1 Configure Root Directory

Lighttpd's configuration file is /etc/lighttpd.conf. If necessary, modify its parameters as needed. The default directory for storing web pages is /home/sysuser/www, and the pure platform directory is /home/root/www/.

4.24.2 Use https Secure Link

At present, the system has its own https configuration file. If necessary, please copy the /etc/lighttpd/lighttpd-https.conf file to the /home/root/project/ directory.

```
# mkdir /home/root/project  
# cp /etc/lighttpd/lighttpd-https.conf /home/root/project/
```

Generate certificate files

```
# openssl_gen_cert.sh  
# cat certificate.pem privatekey.pem > /home/root/project/server.pem  
# cp certificate.pem /home/root/project/ca.crt
```

After the configuration, restart the system even when the https function is already enabled.

4.25 Ways to Configure Startup Programs

Add startup programs to /etc/rc.local.

```
root@adam3600:~# vi /etc/rc.local
```

```
#!/bin/sh -e
```

```

#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
/home/sysuser/start.sh
exit 0
~

```

Note: The executable permission of rc.local must be guaranteed.

```
root@adam3600:~# ll /etc/rc.local
```

```
-rwxr-xr-x  1 root  root          306 Nov  2 13:37 /etc/rc.local
```

When the saved rc.local file is uploaded to the system through FTP or other methods for convenience, the executable permission may be lost. You can use the `chmod a+x /etc/rc.local` command to add the executable permission.

4.26 SNMP Configuration

4.26.1 SNMP Functions

Function	MIB file	OIDs
System Info	SNMPv2-MIB	.1.3.6.1.2.1.1
	UCD-SNMP-MIB	.1.3.6.1.4.1.2021.11
Net Info	IF-MIB	.1.3.6.1.2.1.2
Memory Info	UCD-SNMP-MIB	.1.3.6.1.4.1.2021.4
Disk Info	UCD-SNMP-MIB	.1.3.6.1.4.1.2021.9
Loading Info	UCD-SNMP-MIB	.1.3.6.1.4.1.2021.10
Advantech IO Module Info	ADVANTECH-IO-COMMON-MIB	.1.3.6.1.4.1.10297.101

Table1. MIB List

Device Command on OS:

```

#uname -a          #check kernel verison  .1.3.6.1.2.1.1
#hostname          #check hostname      .1.3.6.1.2.1.1

```

#cat /proc/stat	#check softirq etc	.1.3.6.1.4.1.2021.11
#ifconfig -a	#check interface list	.1.3.6.1.2.1.2
#ethtool eth0	#check interface link speed	.1.3.6.1.2.1.2
#cat /proc/meminfo	#check memory info	.1.3.6.1.4.1.2021.4
#df	#check disk info	.1.3.6.1.4.1.2021.9
#htop	#check cpu load every 1,5,15 minutes	.1.3.6.1.4.1.2021.10

4.26.2 Introduction to ADVANTECH-IO-COMMON-MIB

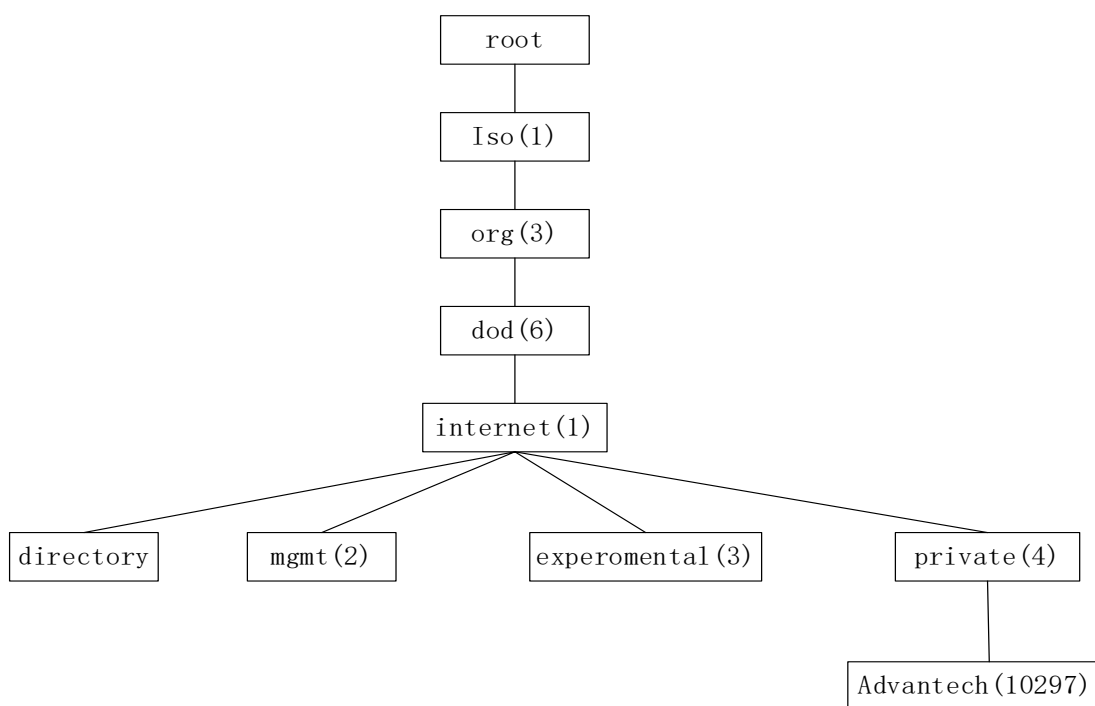


Figure1. OIDs tree

atBasicIO (1.3.6.1.4.1.10297.101.1)

+---ioModuleObj (1.3.6.1.4.1.10297.101.1.1): SEQUENCE of ioModuleEntry

+--- ioModuleTable (1.3.6.1.4.1.10297.101.1.1.1)

+--- ioModuleEntry(1.3.6.1.4.1.10297.101.1.1.1.1): Each entry contains IO module information

+---analogInputObj(1.3.6.1.4.1.10297.101.1.2)

+--- aiTable (1.3.6.1.4.1.10297.101.1.2.1)

+--- aiEntry(1.3.6.1.4.1.10297.101.1.2.1.1):

+---analogOutputObj(1.3.6.1.4.1.10297.101.1.3)

- +--- aoTable (1.3.6.1.4.1.10297.101.1.3.1)
 - +--- aoEntry(1.3.6.1.4.1.10297.101.1.3.1.1):
- +---digitalInputObj(1.3.6.1.4.1.10297.101.1.4)
 - +--- diTable (1.3.6.1.4.1.10297.101.1.4.1)
 - +--- diEntry(1.3.6.1.4.1.10297.101.1.4.1.1):
- +---digitalOutputObj(1.3.6.1.4.1.10297.101.1.5)
 - +--- doTable (1.3.6.1.4.1.10297.101.1.5.1)
 - +--- doEntry(1.3.6.1.4.1.10297.101.1.5.1.1):
- +---counterObj(1.3.6.1.4.1.10297.101.1.6)
 - +--- counterTable (1.3.6.1.4.1.10297.101.1.6.1)
 - +--- counterEntry(1.3.6.1.4.1.10297.101.1.6.1.1):

Indexes	Syntax	Access	Status	Descr.
ioModuleIndex	Integer32 (1..32)	RO	current	The index of the module entry
ioModuleSlotIndex	Integer32 (0..31)	RO	current	The module (slot) index
ioModuleIdentify	OCTET STRING (SIZE(1..16))	RO	current	The module ID
ioModuleDescr	OCTET STRING (SIZE(1..256))	RO	current	The module description

Table2. ioModuleEntry Table

Indexes	Syntax	Access	Status	Descr.
aiIndex	Integer32 (1..32)	RO	current	A unique value for each analog input contained by the IO module.
aiChannelIndex	Integer32 (0..31)	RO	current	The AI channel index.
aiModbusAddress	Integer32	RO	current	The MODBUS address of the AI channel
aiIntergrationTime	DisplayString	RO	current	The integration time of all AI channel. The value could be '50 Hz', '60 Hz', 'Auto' .If the module has no AI, this field will be empty
aiEnabled	INTEGER { true(1), false(2) }	RW	current	The enabled status of the AI channel
aiRangeName	OCTET STRING (SIZE(1..32))	RO	current	The name of the range.
aiRangeCode	OCTET STRING (SIZE(4))	RW	current	The setting code of the

				range in HEX.
aiRangeHigh	OCTET STRING (SIZE(1..8))	RO	current	The maximum boundary of the AI range.
aiRangeLow	OCTET STRING (SIZE(1..8))	RO	current	The minimum boundary of the AI range.
aiRangeUnit	OCTET STRING (SIZE(1..8))	RO	current	The unit name of the AI range..
aiRawValue	OCTET STRING (SIZE(0..4))	RO	current	The MODBUS data value in HEX. The value is from '0' to 'FFFF'.
aiEngValue	OCTET STRING (SIZE(0..8))	RO	current	The engineering unit value. For example, '5.232'.

Table3. aiEntry Table

Indexes	Syntax	Access	Status	Descr.
aoIndex	Integer32 (1..32)	RO	current	A unique value for each analog output contained by the IO module.
aoChannelIndex	Integer32 (0..31)	RO	current	The AO channel index.
aoModbusAddress	Integer32	RO	current	The MODBUS address of the AO channel
aoStartupValue	OCTET STRING (SIZE(1..8))	RO	current	The power on startup value of the AO channel in engineering unit.
aoRangeName	OCTET STRING (SIZE(1..32))	RO	current	The name of the range.
aoRangeCode	OCTET STRING (SIZE(4))	RW	current	The setting code of the range in HEX.
aoRangeHigh	OCTET STRING (SIZE(1..8))	RO	current	The maximum boundary of the AO range.
aoRangeLow	OCTET STRING (SIZE(1..8))	RO	current	The minimum boundary of the AO range.
aoRangeUnit	OCTET STRING (SIZE(1..8))	RO	current	The unit name of the AO range.
aoRawValue	OCTET STRING (SIZE(1..4))	RW	current	The MODBUS data value in HEX. The value

				is from '0' to '0FFF' for normal 12 bits AO.
aoEngValue	OCTET STRING (SIZE(1..8))	RW	current	The engineering unit value. For example, '5.232'.

Table4. aoEntry Table

Indexes	Syntax	Access	Status	Descr.
diIndex	Integer32 (1..64)	RO	current	A unique value for each digital input contained by the IO module.
diChannelIndex	Integer32 (0..63)	RO	current	The DI channel index.
diModbusAddress	Integer32	RO	current	The MODBUS address of the DI channel
diInverted	INTEGER { true(1), false(0) }	RW	current	The DI signal invert function enabled status.
diValue	OCTET STRING (SIZE(1..8))	RO	current	The DI value, the value will be '0' or '1'.

Table5. diEntry Table

Indexes	Syntax	Access	Status	Descr.
doIndex	Integer32 (1..64)	RO	current	A unique value for each digital output contained by the IO module.
doChannelIndex	Integer32 (0..63)	RO	current	The DO channel index
doModbusAddress	Integer32	RO	current	The MODBUS address of the DO channel
doValue	OCTET STRING (SIZE(1..8))	RW	current	The value will be '0' or '1'.

Table6. doEntry Table

Indexes	Syntax	Access	Status	Descr.
counterIndex	Integer32 (1..16)	RO	current	A unique value for each counter

				contained by the IO module.
counterChannelIndex	Integer32 (0..15)	RO	current	The Counter channel index
counterModbusAddress	Integer32	RO	current	The MODBUS address of the Counter channel
counterFilterValue	Integer32	RW	current	The signal filter value
counterState	INTEGER { Start(1), Stop(0) }	RW	current	The counter enabled status
counterStartup	Integer32	RW	current	The startup value of the counter
counterOverflow	INTEGER { true(1), false(0) }	RO	current	The counter value is overflow or not
counterModeName	OCTET STRING (SIZE(1..32))	RO	current	The counter mode name.
counterModeCode	INTEGER	RW	current	The setting code of the mode in HEX.
counterValue	OCTET STRING (SIZE(1..8))	RO	current	The value will be from '00000000' to '4294967295' in Decimal.
counterFreqAcqTime	Integer32	RW		Frequency acquire time for frequency mode.

Table7. counterEntry Table

4.26.3 Create snmpv3 User

Step1 :

Execute commands on ADAM-5630:

```
#net-snmp-config --create-snmpv3-user -a "my_password" -X DES -A MD5 -x "my_password" myuser
```

Note: SNMPv3 password must have 8 characters at least.

```
root@adam5630:~# net-snmp-config --create-snmpv3-user -a "my_password" -X DES -A MD5 -x "my_password" myuser
adding the following line to /var/net-snmp/snmpd.conf:
    createUser myuser MD5 "my_password" DES my_password
adding the following line to /usr/local/net-snmp/share/snmp/snmpd.conf:
    rwuser myuser
root@adam5630:~#
```

Step2: Modify /home/root/project/snmpd.conf according to the prompt returned by the previous command:

```
root@adam5630:~# vi /home/root/project/snmpd.conf
```

Add the following content:

```

createUser myuser MD5 "my_password" DES my_password
rouser myuser
rouser myuser AuthPriv
group groupv3 usm myuser
access groupv3 "" any auth exact all all all

```

```

# First, map the community name "public" into a "security name"
#   sec.name source community
com2sec notConfigUser default public
com2sec advantechsnmp default private
rocommunity public default

createUser username MD5 "SNMP_PWD" DES SNMP_PWD
rouser username
rouser username AuthPriv
group groupv3 usm username
access groupv3 "" any auth exact all all all

# Second, map the security name into a group name:

```

4.26.4 Accesses Local SNMP on ADAM5630

1) Disk info :

```

#snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.9
#snmpwalk -v 3 -u myuser -l authPriv -a MD5 -A my_password -x DES -X my_password
localhost .1.3.6.1.4.1.2021.9

```

```

root@adam5630:~# snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.9
UCD-SNMP-MIB::dskIndex.1 = INTEGER: 1
UCD-SNMP-MIB::dskIndex.2 = INTEGER: 2
UCD-SNMP-MIB::dskIndex.3 = INTEGER: 3
UCD-SNMP-MIB::dskIndex.4 = INTEGER: 4
UCD-SNMP-MIB::dskIndex.5 = INTEGER: 5
UCD-SNMP-MIB::dskIndex.6 = INTEGER: 6
UCD-SNMP-MIB::dskPath.1 = STRING: /
UCD-SNMP-MIB::dskPath.2 = STRING: /var
UCD-SNMP-MIB::dskPath.3 = STRING: /media/mmcblk0p1
UCD-SNMP-MIB::dskPath.4 = STRING: /var/volatile
UCD-SNMP-MIB::dskPath.5 = STRING: /dev/shm
UCD-SNMP-MIB::dskPath.6 = STRING: /media/ram
UCD-SNMP-MIB::dskDevice.1 = STRING: ubi0:rootfs
UCD-SNMP-MIB::dskDevice.2 = STRING:
UCD-SNMP-MIB::dskDevice.3 = STRING: /dev/mmcblk0p1
UCD-SNMP-MIB::dskDevice.4 = STRING: tmpfs
UCD-SNMP-MIB::dskDevice.5 = STRING: tmpfs
UCD-SNMP-MIB::dskDevice.6 = STRING: tmpfs
UCD-SNMP-MIB::dskMinimum.1 = INTEGER: 10000
UCD-SNMP-MIB::dskMinimum.2 = INTEGER: -1
UCD-SNMP-MIB::dskMinimum.3 = INTEGER: -1
UCD-SNMP-MIB::dskMinimum.4 = INTEGER: -1
UCD-SNMP-MIB::dskMinimum.5 = INTEGER: -1
UCD-SNMP-MIB::dskMinimum.6 = INTEGER: -1
UCD-SNMP-MIB::dskMinPercent.1 = INTEGER: -1
UCD-SNMP-MIB::dskMinPercent.2 = INTEGER: 5
UCD-SNMP-MIB::dskMinPercent.3 = INTEGER: 10
UCD-SNMP-MIB::dskMinPercent.4 = INTEGER: 10
UCD-SNMP-MIB::dskMinPercent.5 = INTEGER: 10
UCD-SNMP-MIB::dskMinPercent.6 = INTEGER: 10
UCD-SNMP-MIB::dskTotal.1 = INTEGER: 209284

```

2) Memory info :

```

#snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.4

```

```
#snmpwalk -v 3 -u myuser -l authPriv -a MD5 -A my_password -x DES -X  
my_password localhost .1.3.6.1.4.1.2021.4
```

```
root@adam5630:~# snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.4  
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0  
UCD-SNMP-MIB::memErrorName.0 = STRING: swap  
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 0 kB  
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 0 kB  
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 507844 kB  
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 436220 kB  
UCD-SNMP-MIB::memTotalFree.0 = INTEGER: 436220 kB  
UCD-SNMP-MIB::memMinimumSwap.0 = INTEGER: 16000 kB  
UCD-SNMP-MIB::memShared.0 = INTEGER: 528 kB  
UCD-SNMP-MIB::memBuffer.0 = INTEGER: 476 kB  
UCD-SNMP-MIB::memCached.0 = INTEGER: 39220 kB  
UCD-SNMP-MIB::memSwapError.0 = INTEGER: error(1)  
UCD-SNMP-MIB::memSwapErrorMsg.0 = STRING: Running out of swap space (0)  
root@adam5630:~#
```

3) CPU loading :

```
#snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.10  
#snmpwalk -v 3 -u myuser -l authPriv -a MD5 -A my_password -x DES -X my_password  
localhost .1.3.6.1.4.1.2021.10
```

```
root@adam5630:~# snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.10  
UCD-SNMP-MIB::laIndex.1 = INTEGER: 1  
UCD-SNMP-MIB::laIndex.2 = INTEGER: 2  
UCD-SNMP-MIB::laIndex.3 = INTEGER: 3  
UCD-SNMP-MIB::laNames.1 = STRING: Load-1  
UCD-SNMP-MIB::laNames.2 = STRING: Load-5  
UCD-SNMP-MIB::laNames.3 = STRING: Load-15  
UCD-SNMP-MIB::laLoad.1 = STRING: 2.01  
UCD-SNMP-MIB::laLoad.2 = STRING: 2.03  
UCD-SNMP-MIB::laLoad.3 = STRING: 2.05  
UCD-SNMP-MIB::laConfig.1 = STRING: 12.00  
UCD-SNMP-MIB::laConfig.2 = STRING: 10.00  
UCD-SNMP-MIB::laConfig.3 = STRING: 5.00  
UCD-SNMP-MIB::laLoadInt.1 = INTEGER: 200  
UCD-SNMP-MIB::laLoadInt.2 = INTEGER: 202  
UCD-SNMP-MIB::laLoadInt.3 = INTEGER: 204  
UCD-SNMP-MIB::laLoadFloat.1 = Opaque: Float: 2.010000  
UCD-SNMP-MIB::laLoadFloat.2 = Opaque: Float: 2.030000  
UCD-SNMP-MIB::laLoadFloat.3 = Opaque: Float: 2.050000  
UCD-SNMP-MIB::laErrorFlag.1 = INTEGER: noError(0)  
UCD-SNMP-MIB::laErrorFlag.2 = INTEGER: noError(0)  
UCD-SNMP-MIB::laErrorFlag.3 = INTEGER: noError(0)  
UCD-SNMP-MIB::laErrorMessage.1 = STRING:  
UCD-SNMP-MIB::laErrorMessage.2 = STRING:  
UCD-SNMP-MIB::laErrorMessage.3 = STRING:  
root@adam5630:~#
```

4) SystemInfo :

```
#snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.11  
#snmpwalk -v 3 -u myuser -l authPriv -a MD5 -A my_password -x DES -X my_password  
localhost .1.3.6.1.4.1.2021.11
```

```

root@adam5630:~# snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.11
UCD-SNMP-MIB::ssIndex.0 = INTEGER: 1
UCD-SNMP-MIB::ssErrorName.0 = STRING: systemStats
UCD-SNMP-MIB::ssSwapIn.0 = INTEGER: 0 kB
UCD-SNMP-MIB::ssSwapOut.0 = INTEGER: 0 kB
UCD-SNMP-MIB::ssIOSent.0 = INTEGER: 0 blocks/s
UCD-SNMP-MIB::ssIOReceive.0 = INTEGER: 0 blocks/s
UCD-SNMP-MIB::ssSysInterrupts.0 = INTEGER: 971 interrupts/s
UCD-SNMP-MIB::ssSysContext.0 = INTEGER: 1921 switches/s
UCD-SNMP-MIB::ssCpuUser.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuSystem.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuIdle.0 = INTEGER: 98
UCD-SNMP-MIB::ssCpuRawUser.0 = Counter32: 24518
UCD-SNMP-MIB::ssCpuRawNice.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawSystem.0 = Counter32: 13975
UCD-SNMP-MIB::ssCpuRawIdle.0 = Counter32: 2146027
UCD-SNMP-MIB::ssCpuRawWait.0 = Counter32: 11
UCD-SNMP-MIB::ssCpuRawKernel.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawInterrupt.0 = Counter32: 0
UCD-SNMP-MIB::ssIORawSent.0 = Counter32: 4
UCD-SNMP-MIB::ssIORawReceived.0 = Counter32: 10892
UCD-SNMP-MIB::ssRawInterrupts.0 = Counter32: 22951535
UCD-SNMP-MIB::ssRawContexts.0 = Counter32: 45369843
UCD-SNMP-MIB::ssCpuRawSoftIRQ.0 = Counter32: 1033
UCD-SNMP-MIB::ssRawSwapIn.0 = Counter32: 0
UCD-SNMP-MIB::ssRawSwapOut.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawSteal.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawGuest.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawGuestNice.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuNumCpus.0 = INTEGER: 1
root@adam5630:~#

```

5) Net info :

```
#snmpwalk -v 2c -c public localhost .1.3.6.1.2.1.2
```

```
#snmpwalk -v 3 -u myuser -l authPriv -a MD5 -A my_password -x DES -X my_password
```

localhost .1.3.6.1.2.1.2

```

root@adam5630:~# snmpwalk -v 2c -c public localhost .1.3.6.1.2.1.2
IF-MIB::ifNumber.0 = INTEGER: 3
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: eth0
IF-MIB::ifDescr.3 = STRING: eth1
IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)
IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.3 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifMtu.1 = INTEGER: 65536
IF-MIB::ifMtu.2 = INTEGER: 1500
IF-MIB::ifMtu.3 = INTEGER: 1500
IF-MIB::ifSpeed.1 = Gauge32: 10000000
IF-MIB::ifSpeed.2 = Gauge32: 10000000
IF-MIB::ifSpeed.3 = Gauge32: 10000000
IF-MIB::ifPhysAddress.1 = STRING:
IF-MIB::ifPhysAddress.2 = STRING: 98:5d:ad:6e:dc:64
IF-MIB::ifPhysAddress.3 = STRING: 98:5d:ad:6e:dc:66
IF-MIB::ifAdminStatus.1 = INTEGER: up(1)
IF-MIB::ifAdminStatus.2 = INTEGER: up(1)
IF-MIB::ifAdminStatus.3 = INTEGER: up(1)
IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.2 = INTEGER: down(2)
IF-MIB::ifOperStatus.3 = INTEGER: up(1)
IF-MIB::ifLastChange.1 = Timeticks: (0) 0:00:00.00
IF-MIB::ifLastChange.2 = Timeticks: (0) 0:00:00.00
IF-MIB::ifLastChange.3 = Timeticks: (0) 0:00:00.00
IF-MIB::ifInOctets.1 = Counter32: 54555
IF-MIB::ifInOctets.2 = Counter32: 2265937
IF-MIB::ifInOctets.3 = Counter32: 10481657
IF-MIB::ifInUcastPkts.1 = Counter32: 644
IF-MIB::ifInUcastPkts.2 = Counter32: 27372
IF-MIB::ifInUcastPkts.3 = Counter32: 119544
IF-MIB::ifInNUcastPkts.1 = Counter32: 0
IF-MIB::ifInNUcastPkts.2 = Counter32: 0

```

The above information needs to be supported by the following MIB files: UCD-SNMP-MIB, IF-MIB and SNMPv2-MIB.

4.26.5 Obtain the Information of ADAM-5630's Local IO Module

ADVANTECH-IO-COMMON-MIB.mib is needed and it is placed at the following location:

```
root@adam5630:~# ls /home/root/project/
ADVANTECH-IO-COMMON-MIB.mib  ModbusDaemon.acr
```

Below module information can be read via `snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.10297.101`.

Module Type	Module Name	Specification
Analog I/O	ADAM-5013	3-ch RTD input
	ADAM-5017	8-ch AI
	ADAM-5017P	8-ch AI with independent input
	ADAM-5017H	8-ch High-speed AI
	ADAM-5017UH	8-ch Ultra High-speed AI
	ADAM-5018	7-ch Thermcouple input
	ADAM-5018P	7-ch Thermcouple input with independent input
	ADAM-5024	4-ch AO
Digital I/O	ADAM-5050	16-ch DI/O
	ADAM-5051	16-ch DI
	ADAM-5051D	16-ch DI w/LED
	ADAM-5051S	16-ch Isolated DI w/LED
	ADAM-5052	8-ch DI
	ADAM-5053S	32-ch Isolated DI
	ADAM-5055S	16-ch Isolated DI/O w/LED
	ADAM-5056	16-ch DO
	ADAM-5056D	16-ch DO w/LED
	ADAM-5056S	16-ch Isolated DO w/LED
	ADAM-5056SO	16-ch Isolated DO w/LED(source)
	ADAM-5057S	32-ch Isolated DO
Relay Output	ADAM-5060	6-ch Relay output
	ADAM-5068	8-ch Relay Output
	ADAM-5069	8-ch Relay Output
Counter	ADAM-5080	4-ch Counter/Frequency
	ADAM-5081	4-ch High speed Counter/Frequency

Table8. Adam IO Module List

4.26.6 Remote Access

Download iReasoning MIB Browser for testing. iReasoning MIB Browser personal edition only supports SNMP v2.

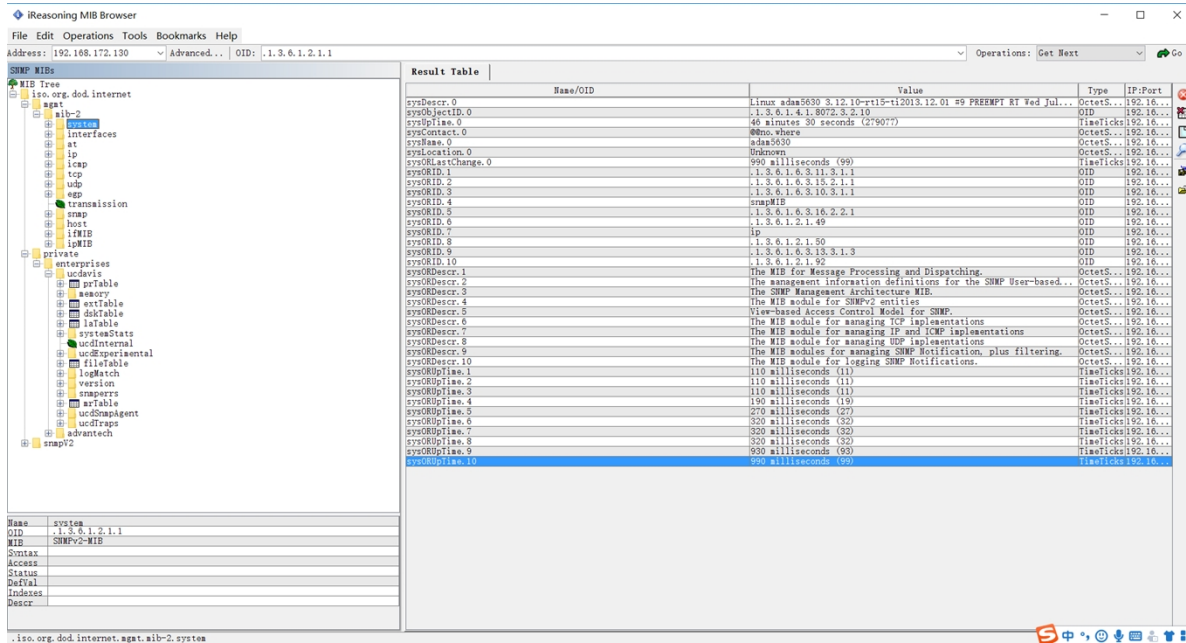


Figure2. iReasoning MIB Browser

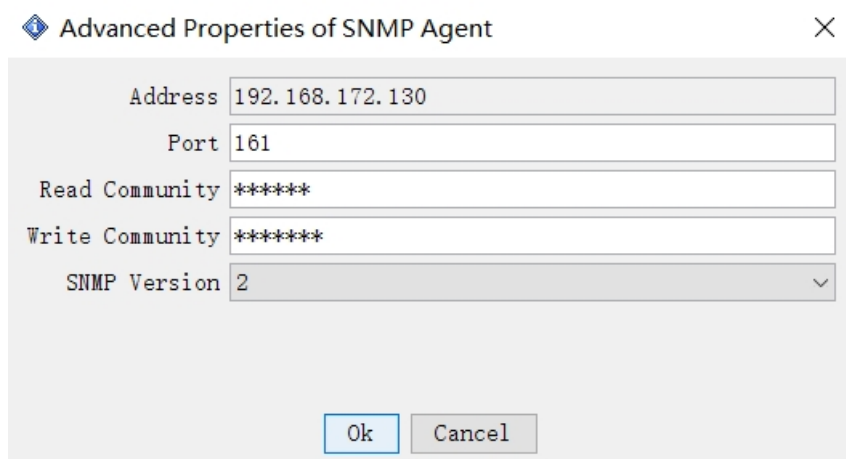


Figure3. Advanced.. configuration

4.27 Install a Driver to View Ismod

View the information of installed modules

```
root@adam3600:~# lsmod
```

Module	Size	Used by
boardio	27695	0
biokernbase	5963	1 boardio

```

gpioinfo          5514  1
ipv6              268782 12
option           26392  0
usb_wwan         5240  1 option
ext4             331096  0
jbd2             55796  1 ext4

```

Installed modules

```
root@adam3600:~# insmod /home/sysuser/driver/boardio.ko
```

```
root@adam3600:~# lsmod
```

```

Module          Size  Used by
boardio         27695  0
biokernbase     5963  1 boardio
gpioinfo        5514  1
ipv6            268782 12
option          26392  0
usb_wwan        5240  1 option
ext4            331096  0
jbd2            55796  1 ext4

```

```
root@adam3600:~#
```

Uninstall modules

```
root@adam3600:~# rmmod boardio
```

5 Program Development

5.1 Onboard Resource Programming (BoardResource SDK)

5.1.1 Watchdog

a) Steps to use the watchdog:

- Initialize the watchdog (WDT_Init)
- Enable the watchdog (WDT_Enable)
- Feed the watchdog (WDT_Strobe)
- Disable the watchdog (WDT_Disable)
- Release watchdog resources (WDT_DeInit)

b) Introduction to API

- **BR_RESULT WDT_Init** (BR_HANDLE * handle)

Initialize the Watchdog.

This function must be called before any other watchdog functions.

Parameters

[out] handle Handle of the Watchdog.

Returns

result, BR_SUCCESS if successful.

- **BR_RESULT WDT_Enable** (BR_HANDLE handle,
unsigned int spanSeconds
)

Enable the Watchdog.

Parameters

[in] handle Handle of the Watchdog.

[in] spanSeconds time span of the Watchdog.range from 1 to 3600 seconds

Returns

result, BR_SUCCESS if successful.

- **BR_RESULT WDT_Strobe** (BR_HANDLE handle)

Strobe the Watchdog.

after enabling the Watchdog using WDT_Enable, the application must continuously call WDT_Strobe to trigger the Watchdog.

Parameters

[in] handle Handle of the Watchdog.

Returns

result, BR_SUCCESS if successful.

- **BR_RESULT WDT_Disable** (BR_HANDLE handle)

Disable the Watchdog.

Parameters

[in] handle Handle of the Watchdog.

Returns

result, BR_SUCCESS if successful.

➤ BR_RESULT **WDT_DeInit** (BR_HANDLE handle)

De-initialize the Watchdog.

Parameters

[in] handle Handle of the Watchdog.

Returns

result, BR_SUCCESS if successful.

c). Code instance

```
/******
```

```
> File Name: wdtttest.c
```

```
*****/
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
#include <signal.h>
```

```
#include <syslog.h>
```

```
#include <string.h>
```

```
#include "board_resource.h"
```

```
#define TIMEOUT 10
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    BR_HANDLE wdt_fd = 0;
```

```
    int timeout = TIMEOUT;
```

```
    BR_RESULT ret = BR_SUCCESS;
```

```
    //init handle
```

```
    printf("init handle\n");
```

```

ret = WDT_Init(&wdt_fd);
if (ret != BR_SUCCESS)
{
    printf("open device fail[%d]\n", ret);
    return 0;
}
//enable watch dog
ret = WDT_Enable(wdt_fd,timeout);
if (ret != BR_SUCCESS)
{
    printf("enable wdt fail[%d]\n", ret);
    return 0;
}

printf("press Ctrl+C in %d second,the wdt will reboot system\n",timeout);
while(timeout--){
    //strobe dog
    ret = WDT_Strobe(wdt_fd);
    if (ret == BR_SUCCESS)
    {
        printf("strobe wdt success[%d]\n",timeout);
    }
    sleep(1);
}

sleep(5);

//disable dog
printf("disable wdt\n");
ret = WDT_Disable(wdt_fd);
if (ret != BR_SUCCESS)
{
    printf("disable wdt fail[%d]\n", ret);
    return 0;
}
//uninit handle
WDT_DeInit(wdt_fd);
printf("test over\n");

```

```
    return 0;
}
```

5.1.2 PLED

a). Steps to use programmable LED

- Initialize LED (LED_Init)
- Turn on the led (LED_On)
- Turn off the LED (LED_Off)
- Release resources (LED_DeInit)

b). Introduction to API

Available LED types are marked on the panel.

LED_TYPE_RUN RUN LED

LED_TYPE_ERROR ERROR LED

LED_TYPE_PROGRAM PROG LED

LED_TYPE_P1 LED 1

LED_TYPE_P2 LED 2

LED_TYPE_P3 LED 3

LED_TYPE_P4 LED 4

➤ BR_RESULT **LED_Init** (LEDType type,
BR_HANDLE * handle
)

Initialize the LED device.

This function must be called before any other LED functions.

Parameters

[in] type type of the LED device.

[out] handle Handle of the LED device.

Returns

result, BR_SUCCESS if successful.

➤ BR_RESULT LED_On (BR_HANDLE handle)

Light the LED.

Parameters

[in] handle Handle of the LED device.

Returns

result, BR_SUCCESS if successful.

➤ BR_RESULT LED_Off (BR_HANDLE handle)

Turn off the LED.

Parameters

[in] handle Handle of the LED device.

Returns

result, BR_SUCCESS if successful.

➤ BR_RESULT LED_DeInit (BR_HANDLE handle)

De-initialize the LED device.

Parameters

[in] handle Handle of the LED device.

Returns

result, BR_SUCCESS if successful.

c) Code instances

```
/******  
> File Name: ledtest.c  
*****/
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <unistd.h>  
#include "board_resource.h"
```

```

int main(int argc, char *argv[])
{
    int fd =0;
    BR_RESULT ret = BR_SUCCESS;
    ret = LED_Init(LED_TYPE_RUN, &fd);
    if (ret != BR_SUCCESS)
    {
        printf("open LED_TYPE_RUN[P1] fail[%d]\n", ret);
        return -1;
    }
    LED_On(fd);
    sleep(1);
    LED_Off(fd);
    LED_DeInit(fd);

    return 0;
}

```

5.1.3 NODE_ID

a) Steps to query Node ID:

- Initialize resources (Board_Init)
- Read node ID number (Board_GetNodeID)
- Release resources (Board_DeInit)

□

b) Introduction to API

➤ BR_RESULT **Board_Init** (BR_HANDLE * handle)

Initialize the Board device.

This function must be called before any other board functions.

Parameters

[out] handle Handle of the Board device.

Returns

result, BR_SUCCESS if successful.

➤ BR_RESULT **Board_GetNodeID** (BR_HANDLE handle,
unsigned int * nodeID
)

get node id of the Board device.

Parameters

[in] handle Handle of the node id.

[out] nodeID node id.

Returns

node id.

➤ BR_RESULT **Board_DeInit** (BR_HANDLE handle)

De-initialize the Board device.

Parameters

[in] handle Handle of the Board device.

Returns

result, BR_SUCCESS if successful.

c). Code instance

```
/******  
> File Name: nodeidtest.c  
*****/  
  
#include <stdio.h>  
#include <unistd.h>  
#include "board_resource.h"  
  
int main(int argc, char *argv[])  
{
```

```

int fd;
unsigned int nodeId;
BR_RESULT ret = BR_SUCCESS;
ret = Board_Init(&fd);
if (ret != BR_SUCCESS)
    printf("open device fail[%d]\n", ret);
Board_GetNodeID( fd, &nodeId );
Board_DelInit(fd);
printf("nodeid:%d(%xh)\n",nodeId,nodeId);
return 0;
}

```

5.1.4 DIO (Non-ADAM3600 Serials)

a) Steps to use DIO nodes:

- Initialize resources (DIO_Init)
- Get DO count (Get_DOCount)
- Read DO value (Do_Read)
- Set DO value (Do_Write)
- Get DI count (Get_DICount)
- Read DI value (Di_Read)
- Release resources (DIO_DelInit)

b) Code instance

```

/*****
> File Name: diotest.c
*****/

```

```

#include <stdio.h>
#include <unistd.h>
#include "board_resource.h"

```

```

int main(int argc, char *argv[])
{
    int fd;
    BR_RESULT ret = BR_SUCCESS;
    ret = DIO_Init(&fd);

```

```

if (ret != BR_SUCCESS)
    printf("open device fail[%d]\n", ret);

int DI = Get_DICount();
int DO = Get_DOCount();
printf("DI:%d,DO:%d\n",DI,DO);
unsigned int value;

int i = 0;
while(1)
{
    if(DI)
    {
        Di_Read(fd,0,DI,&value);
        printf("DI:%08x\n",value);
    }
    if(DO)
    {
        for(i=0;i< DO;i++)
        {
            Do_Read(fd,i,1,&value);
            value = ~value;
            Do_Write(fd,i,1,value);
            Do_Read(fd,i,1,&value);
            printf("DO[%d]:%04x\n",i,value);
            sleep(1);
        }
    }

    sleep(3);
}

return 0;
}

```

5.2 IO Resource (ADAM3600 Serials)

1. View module information:

```
root@adam3600:~# mdlsrch
```

```
IO Module: 0, ADAM-3600
```

```
IO Module: 0, ADAM-3600,ver:01010183,ai: 8,ao: 0,di: 8,do: 4
```

2. Programming code:

diread.cpp

```
/******  
*  
* Linux Example:  
*   diread.cpp  
*  
* Example Category:  
*   DI  
*  
* Description:  
*   This example demonstrates how to use DI function.  
*  
* Instructions for Running:  
*   1. Set the 'deviceNumber' for opening the device.  
*   2. Set the 'startChannel' as the first channel for scan analog samples  
*   3. Set the 'channelCount' to decide how many sequential channels to scan analog  
samples.  
*  
* I/O Connections Overview:  
*   Please refer to your hardware reference manual.  
*  
*****/  
  
#include <stdlib.h>  
#include <stdio.h>  
#include "compatibility.h"  
#include "bdaqcl.h"  
using namespace Automation::BDaq;  
//-----  
// Configure the following three parameters before running the example  
//-----  
#define      deviceNumber      0  
#define      channelCountMax 8
```

```
#define    MAX_MODULE_COUNT  5
```

```
inline void waitAnyKey()
```

```
{  
    do{SLEEP(1);} while(!kbhit());  
}
```

```
int main(int argc, char* argv[])
```

```
{  
    ErrorCode    ret = Success;
```

```
    BDaqDevice *device = NULL;
```

```
    BDaqDio    *dio = NULL;
```

```
    long    moduleNumber = 0;
```

```
    long    startChannel = 0;
```

```
    long    channelCount = 8;
```

```
    //Open device
```

```
    ret = BDaqDevice::Open(deviceNumber, ModeWrite, device);
```

```
    do
```

```
    {
```

```
        //Get dio module
```

```
        ret = device->GetModule(0, dio);
```

```
        CHK_RESULT(ret);
```

```
        long rngCode[channelCountMax] = { DI_NORMAL_MODE };
```

```
        ret = BDaqDevice::Open(deviceNumber, ModeWrite, device);
```

```
        CHK_RESULT(ret);
```

```
        ret = dio->DiSetFuncCode(moduleNumber, startChannel, channelCount,  
&rngCode[startChannel]);
```

```
        CHK_RESULT(ret);
```

```
        memset(rngCode, 0, channelCountMax*sizeof(long));
```

```
        SLEEP(1);
```

```
        ret = dio->DiGetFuncCode(moduleNumber, startChannel, channelCount,  
&rngCode[startChannel]);
```

```

CHK_RESULT(ret);
for (long i = startChannel; i < startChannel + channelCount; ++i)
{
    printf("Channel %ld function code: %lx\n", i, rmgCode[i - startChannel]);
}

printf("Acquisition is in progress, any key to quit!\n\n");
BYTE      dioData[ 1 ] = {0xff};

do
{
    //Read di value
    ret = dio->DiRead(moduleNumber, startChannel, channelCount, dioData);
    CHK_RESULT(ret);
    printf("dio value: %2x\n", dioData[0]);

    SLEEP(1);
} while(!kbhit());
}while(false);

//Close device
if(device != NULL)
{
    device->Close();
}
// If something wrong in this execution, print the error code on screen for tracking.
if(BioFailed(ret))
{
    printf("Some error occurred. And the last error code is Ox%X.\n", ret);
    waitAnyKey();// wait any key to quit!
}
return 0;
}

```

Makefile:

```
arm-linux-gnueabi-g++ -I../inc -L ../lib/ diread.cpp -o diread
```

5.3 Serial Port Programming

5.3.1 Basic Steps

Set struct termios to operate serial port under Linux as below:

- a) Open the device nodes such as /dev/ttyAP0.
- b) Obtain the information of node struct termios, configure serial port parameters such as baud rate, and save the information of struct termios.
- c) Directly call the read and write functions to read and write data.
- d) Close the device node.

5.3.2 Parameter Configuration

- a) Open device node

```
fd = open(Dev, O_RDWR | O_NOCTTY);
```
- b) Set baud rate

```
cfsetispeed(&opt, B9600);  
cfsetospeed(&opt, B9600);
```
- c) Set data bits

```
opt.c_cflag &= ~CSIZE;  
switch ( databits )  
{  
case 5:  
    opt.c_cflag |= CS5;  
    break;  
case 6:  
    opt.c_cflag |= CS6;  
    break;  
case 7:  
    opt.c_cflag |= CS7;  
    break;  
case 8:  
    opt.c_cflag |= CS8;  
    break;  
default:  
    printf( "Unsupported data size\n" );  
    return -1;
```

```
}
```

d) Set stop bits

```
switch (stopbits)
```

```
{
```

```
case 1:
```

```
    opt.c_cflag &= ~CSTOPB;
```

```
    break;
```

```
case 2:
```

```
    opt.c_cflag |= CSTOPB;
```

```
    break;
```

```
default:
```

```
    printf("Unsupported stop bits\n");
```

```
    return -1;
```

```
}
```

e) Set parity bits

```
switch (parity)
```

```
{
```

```
case 'n':
```

```
case 'N':
```

```
    opt.c_cflag &= ~PARENB;
```

```
    opt.c_iflag &= ~INPCK;
```

```
    break;
```

```
case 'o':
```

```
case 'O':
```

```
    opt.c_cflag |= (PARODD | PARENB);
```

```
    opt.c_iflag |= INPCK;
```

```
    break;
```

```
case 'e':
```

```
case 'E':
```

```
    opt.c_cflag |= PARENB;
```

```
    opt.c_cflag &= ~PARODD;
```

```
    opt.c_iflag |= INPCK;
```

```
    break;
```



```

default:
    printf("Unsupported parity\n");
    return -1;
}

```

5.3.3 Other Parameter Instructions

Generally speaking, to set the serial port is to set the baud rate, parity bit and stop bit, namely, the value of each member of the struct termios, as shown below:

struct termio

```

{
    unsigned char c_line; /* line discipline */
    unsigned short c_iflag; /* input mode flag */
    unsigned short c_oflag; /* output mode flag */
    unsigned short c_cflag; /* control mode flag */
    unsigned short c_lflag; /*local mode flag */
    unsigned char c_line; /* line discipline */
    unsigned char c_cc[NCC]; /* control characters */
};

```

c_cflag is the most important parameter in the structure. By assigning a value to it, the user can set the baud rate, character size, data bit, stop bit, parity bit and hardware flow control. In addition, c_iflag and c_cc are common flags. c_cflag, c_iflag and c_cc will be explained in detail below.

Constants supported by **c_cflag**

CBAUD	Bitmask of CBAUD baud rate
B0	0 baud rate (DTR discarded)
B1800	1800 baud rate
B2400	2400 baud rate
B4800	4800 baud rate
B9600	9600 baud rate
B19200	19200 baud rate
B38400	38400 baud rate
B57600	57600 baud rate
B115200	115200 baud rate
EXTA	External clock rate
EXTB	External clock rate
CSIZE	Bitmask of data bit

CS5	5 data bits
CS6	6 data bits
CS7	7 data bits
CS8	8 data bits
CSTOPB	2 stop bits (If CSTOPB does not set stop bit, then there is one stop bit)
CREAD	Enable receiver
PARENB	Enable parity bit
PARODD	Use odd parity instead of even parity
HUPCL	Hang up on last close (DTR discarded)
CLOCAL	Local connection (port owner not changed)
LOBLK	block job control output
CNET_CTSRTS	Enable hardware flow control
Constants supported by c_iflag	
INPCK	Enable parity check
IGNPAR	Ignore parity check errors
PARMRK	Mask of parity check errors
ISTRIP	Remove parity check bit
IXON	Enable output hardware flow control
IXOFF	Enable input software flow control
IXANY	Allow characters to re-enable flow control
IGNBRK	Ignore interruption
BRKINT	Send SIGINT signal when an interruption occurs
INLCR	Maps NL to CR
IGNCR	Ignore CR
IUCLC	Map upper-case to lower-case
ICRNL	Maps CR to NL
IUCLC	Map upper-case to lower-case
IMAXBEL	Reply with ECHO when the input is too long
Constants supported by c_cc	
VINTR	Interrupt control. The corresponding keys are CTRL+C.
VQUIT	Quit the operation. The corresponding keys are CTRL+Z.
VERASE	Delete the operation. The corresponding keys is Backspace (BS).
VKILL	Delete lines. The corresponding keys are CTRL+U.
VEOF	At the end of the file, and the corresponding keys are CTRL+D.
VEOL	At the end of the line, and the corresponding key is Carriage return (CR).

VEOL2 At the end of the second line, and the corresponding key is Line feed (LF).
 VMIN Specify the minimum number of characters to be read.
 VTIME Specify the wait time for reading each character.

Serial control function

tcgetattr Get attributes (termios structure)
 tcsetattr Set attributes (termios structure)
 cfgetispeed Get input speed
 cfgetospeed Get output speed
 cfsetispeed Set input speed
 cfsetospeed Set output speed
 tcdrain Wait until all output is transmitted.
 tcflow Suspend transmission or reception
 tcflush Flush pending input and/or output
 tcsendbreak Send a BREAK character

5.4 Network Programming

5.4.1 TCP Communication

Socket programming based on TCP (connection-oriented) is done at client and server.

The procedures on the client are as follows:

- (1) Create a socket (socket)
- (2) Send a connection request to the server (connect)
- (3) Communicate with the server (send/rcv)
- (4) Close the socket

```

/*****
> File Name: tcp.c
*****/

#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <fcntl.h>
#include <netinet/in.h>
#include <stdlib.h>

```

```

#include <arpa/inet.h>

int main(int argc, char **argv)
{
    int index = 0;
    char buf[1024];
    int sockfd;
    struct sockaddr_in dest_addr;
    if (argc != 3)
    {
        printf("usage: ./tcpc ipaddress port\n ");
        printf("\teg: ./tcpc 127.0.0.1 5555\n");
        return -1;
    }
    int destport = atoi(argv[2]);
    if (-1 == (sockfd = socket(AF_INET, SOCK_STREAM, 0)))
    {
        perror("error in create socket\n");
        exit(0);
    }
    memset(&dest_addr, 0, sizeof(dest_addr));
    dest_addr.sin_family = AF_INET;
    dest_addr.sin_port = htons(destport);
    dest_addr.sin_addr.s_addr = inet_addr(argv[1]);
    //connect
    if (-1 == connect(sockfd, (struct sockaddr*) &dest_addr,
        sizeof(struct sockaddr)))
    {
        perror("connect error\n");
        exit(0);
    }

    while (1)
    {
        sprintf(buf, "%s %d", "tcp send data", index++);
        int n_send_len;
        n_send_len = send(sockfd, buf, strlen(buf), MSG_NOSIGNAL);
    }
}

```

```

    if(n_send_len < 0)
    {
        perror("socket send");
        break;
    }
    printf("send:[%d]s\n", n_send_len, buf);

    int nread = recv(sockfd, buf, sizeof(buf), 0);
    if (nread > 0)
    {
        printf("receive:[%d]s\n", nread, buf);
    }
    if (nread < 0)
    {
        break;
    }
    sleep(1);
}
printf("exit program\n");
shutdown(sockfd, 0);
close(sockfd);
return 0;
}

```

The procedures on the server are as follows:

- (1) Create a socket (socket).
- (2) Bind the socket to a local address and a port (bind).
- (3) Set the socket to listening mode to receive client requests (listen).
- (4) Wait for the client request. When the request arrives, accept the connection request and return a new socket corresponding to the connection (accept).
- (5) Use the returned socket to communicate with the client (send/recv).
- (6) Return and wait for another client request.
- (7) Close the socket.

```

/*****
> File Name: tcps.c
*****/
#include <sys/socket.h>

```

```

#include <unistd.h> // for close function
#include <string.h> // for bzero function
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <arpa/inet.h>

#define SERV_PORT 5555
#define BACKLOG 10 //the counts of connect can keep in wait queen
#define MAXBUFSIZE 200

int main(int argc, char **argv)
{
    char buf[MAXBUFSIZE]; //receive buf
    int sockfd, sockfd_client = 0;
    socklen_t sin_size;
    struct sockaddr_in serv_addr, client_sockaddr; //server ip info
    int serverport;
    if (argc == 2)
    {
        serverport = atoi(argv[1]);
    }
    else
    {
        serverport = SERV_PORT;
    }
    if (-1 == (sockfd = socket(AF_INET, SOCK_STREAM, 0)))
    {
        perror("error in create socket\n");
        exit(0);
    }
    //set the sockaddr_in struct
    memset(&serv_addr, 0, sizeof(serv_addr));

```

```

serv_addr.sin_family = AF_INET;
serv_addr.sin_port = htons(serverport); //server listening port
serv_addr.sin_addr.s_addr = INADDR_ANY; //here is the special in listening tcp connect
//bind , the ip and port information is already in the sockaddr
if (-1 == bind(sockfd, (struct sockaddr*) &serv_addr,
               sizeof(struct sockaddr)))
{
    perror("bind error\n");
    exit(0);
}
printf("bind successful\n");

if (-1 == listen(sockfd, BACKLOG))
{
    perror("listening");
    exit(1);
}
printf("the server is listening...\n");
//accept
if (-1 == (sockfd_client = accept(sockfd,
                                 (struct sockaddr*) &client_sockaddr, &sin_size)))
{
    perror("accept");
    exit(1);
}
printf("accept connect from ip:%s port:%d\n",inet_ntoa(client_sockaddr.sin_addr),ntohs(client_sockaddr.sin_port));
while (1)
{
    memset(buf,0,sizeof(buf));
    int rcvbytes; //the number of bytes receive from socket
    rcvbytes = recv(sockfd_client, buf, MAXBUFSIZE, 0);
    if (-1 == rcvbytes)
    {
        perror("receive");
        exit(1);
    }
    printf("%d bytes receive from connect:%s\n", rcvbytes, buf);
    if(rcvbytes > 0) {

```

```

        recvbytes = send(sockfd_client, buf, recvbytes, MSG_NOSIGNAL);
    }else{
        recvbytes = send(sockfd_client, "heartbeat", strlen("heartbeat"), MSG_NOSIGNAL);
    }
    if(recvbytes < 0)
        break;
}
printf("eixt program\n");
shutdown(sockfd_client,0);
close(sockfd_client);
shutdown(sockfd,0);
close(sockfd);
return 0;
}

```

5.4.2 UDP Communication

Client: (Sender)

- 1) Create a socket (socket).
- 2) Send data to the server (sendto).
- 3) Close the socket.

```

/*
 * File:   udpc.c
 * UDP client
 *
 * Purpose: send a piece of text message every second
 */

```

```

#include<sys/types.h>
#include<sys/socket.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<stdio.h>
#include<stdlib.h>
#include<errno.h>
#include<netdb.h>
#include<stdarg.h>

```



```

#include<string.h>

#define SERVER_PORT 5555
#define BUFFER_SIZE 1024

int main(int argc, char **argv)
{
    /* Server address */
    struct sockaddr_in server_addr;

    if (argc != 3)
    {
        printf("usage: ./udpc ipaddress port\n ");
        printf("\teg: ./udpc 127.0.0.1 5555\n");
        return -1;
    }

    int destport = atoi(argv[2]);

    bzero(&server_addr, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr(argv[1]);
    server_addr.sin_port = htons(destport);

    /* create a socket */
    int client_socket_fd = socket(AF_INET, SOCK_DGRAM, 0);
    if (client_socket_fd < 0)
    {
        perror("Create Socket Failed:");
        exit(1);
    }

    int index=0;
    char buffer[BUFFER_SIZE];
    bzero(buffer, BUFFER_SIZE);

    while (1)
    {

```

```

sprintf(buffer, "%s %d", "udp send data",index++);
int n_send_len;
n_send_len = sendto(client_socket_fd, buffer, strlen(buffer), 0,
                    (struct sockaddr*) &server_addr, sizeof(server_addr));

if(n_send_len < 0)
{
    perror("socket send");
    break;
}
printf("send:[%d]%s\n", n_send_len, buffer);

struct sockaddr_in client_addr;
size_t client_addr_length = 0;
int nread = recvfrom(client_socket_fd, buffer, BUFFER_SIZE, 0,
                    (struct sockaddr*) &client_addr, &client_addr_length);
if ( nread == -1)
{
    perror("Receive Data Failed:");
    exit(1);
}
printf("recv:[%d]%s\n", nread, buffer);
sleep(1);
}

close(client_socket_fd);
return 0;
}

```

Server: (receiving end)

- 1) Create a socket (socket).
- 2) Bind the socket to a local address and a port (bind).
- 3) Use the returned socket to communicate with the client (recvfrom).
- 4) Close the socket.

```

/*****

```

```

> File Name: server.c

```

```

*****/

#include<sys/types.h>
#include<sys/socket.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<stdio.h>
#include<stdlib.h>
#include<errno.h>
#include<netdb.h>
#include<stdarg.h>
#include<string.h>

#define SERVER_PORT 5555
#define BUFFER_SIZE 1024

int main(int argc, char **argv)
{
    /* Create a UDP socket */
    struct sockaddr_in server_addr;
    int serverport;
    if (argc == 2)
    {
        serverport = atoi(argv[1]);
    }
    else
    {
        serverport = SERVER_PORT;
    }

    bzero(&server_addr, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    server_addr.sin_port = htons(serverport);

    /* create a socket */
    int server_socket_fd = socket(AF_INET, SOCK_DGRAM, 0);
    if (server_socket_fd == -1)

```

```

{
    perror("Create Socket Failed:");
    exit(1);
}

/* bind the socket */
if (-1 == (bind(server_socket_fd, (struct sockaddr*) &server_addr,
                sizeof(server_addr))))
{
    perror("Server Bind Failed:");
    exit(1);
}
printf("bind port %u success\n", ntohs(server_addr.sin_port = htons(serverport)));

char buffer[BUFFER_SIZE];
/* data transmission */
while (1)
{
    /* Define an address to capture the client address */
    struct sockaddr_in client_addr;
    socklen_t client_addr_length = sizeof(client_addr);

    /* Receive data */
    bzero(buffer, BUFFER_SIZE);
    int nread = recvfrom(server_socket_fd, buffer, BUFFER_SIZE, 0,
                        (struct sockaddr*) &client_addr, &client_addr_length);
    if ( nread == -1)
    {
        perror("Receive Data Failed:");
        exit(1);
    }
    printf("from ip:%s port:%d,[%d]%s\n", inet_ntoa(client_addr.sin_addr), ntohs(client_addr.sin_port), nread, buffer);
    if (sendto(server_socket_fd, buffer, strlen(buffer), 0,
              (struct sockaddr*) &client_addr, sizeof(client_addr)) < 0)
    {
        perror("Send Failed:");
        exit(1);
    }
}

```

```

    }
    close(server_socket_fd);
    return 0;
}

```

5.5 WebService Programming

Lighttpd uses fastcgi function by default, and it can be extended by programming if necessary.

Please refer to *4.24 Web Server (lighttpd) Configuration* to configure web to https.

5.5.1 Lighttpd Configuration

Configure fastcgi.server at the end of the /etc/lighttpd.conf file as below:

#vi /etc/lighttpd.conf

```

fastcgi.server += (
    "/data" => (
        "test.fastcgi.handler" => (
            "socket" => "/tmp/WebService.fastcgi.socket",
            "check-local" => "disable",
        )
    ),
    "/sys" => (
        "test.fastcgi.handler" => (
            "socket" => "/tmp/WebService.fastcgi.socket",
            #"allow-x-sendfile" => "enable",
            "check-local" => "disable",
        )
    ),
    "/ext" => (
        "test.fastcgi.handler" => (
            "socket" => "/tmp/ext.fastcgi.socket",
            #"allow-x-sendfile" => "enable",
            "check-local" => "disable",
        )
    )
)

```

)

Enable lighttd

```
#lighttpd -f /etc/lighttpd.conf
```

5.5.2 Code

```
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include <locale.h>
#include <dirent.h>
#include <fcntl.h>
#include <shadow.h>
#include <dlfcn.h>
#include <unistd.h>
```

```
#include "fastcgi.h"
#include "fcgios.h"
#include "fcgi_stdio.h"
```

```
#define STANDALONE_MODE 1
```

```
#ifndef FCGI_SOCKET
#define FCGI_SOCKET "/tmp/ext.fastcgi.socket"
#endif
```

```
#if STANDALONE_MODE
static int stdinFds[3];
#endif
```

```
static void handle_request (
    char * request_uri,
    char * query_string,
    char * request_method,
```

```

char * http_accept,
char * http_user_agent,
char * http_accept_encoding,
char * http_accept_language )
{

if ( strcmp( request_method, "GET" ) != 0 )
{
    printf( "Status: 403 Forbidden\r\nContent-Type: text/plain\r\n\r\nInvalid Request" );
    printf( "{\"HTTP\":\"403\"}" );
    return;
}

if ( strstr( request_uri, "/ext" ) == NULL )
{
    printf( "Status: 403 Forbidden\r\nContent-Type: text/plain\r\n\r\nIncorrect URI" );
    printf( "{\"HTTP\":\"403\"}" );
    return;
}

printf( "Status: 200 OK\r\nContent-Type: application/json\r\n\r\n" );
printf( "{\r\n" );
printf( "this is a fastcgi example\r\n" );
printf( "}\r\n" );
}

int main ( int argc, char * argv[] )
{
    int rc;
    int listen_fd;

    daemon(0,0);

#if STANDALONE_MODE
    rc = OS_LibInit( stdinFds );
    if ( rc != 0 )
    {
        printf( "Error initializing OS library: %d\n", rc );
    }
#endif
}

```

```

    return -1;
}

if ( ( listen_fd = OS_CreateLocalIpcFd( FCGI_SOCKET, 5 ) ) == -1 )
{
    printf( "OS_CreateLocalIpcFd failed\n" );
    return -2;
}

chmod( FCGI_SOCKET, ACCESSPERMS );

close( STDIN_FILENO );
if ( listen_fd != FCGI_LISTENSOCK_FILENO )
{
    dup2( listen_fd, FCGI_LISTENSOCK_FILENO );
    close( listen_fd );
}

close( STDOUT_FILENO );
close( STDERR_FILENO );
#endif

while ( FCGI_Accept() >= 0 )
{
    handle_request(
        getenv( "REQUEST_URI" ),
        getenv( "QUERY_STRING" ),
        getenv( "REQUEST_METHOD" ),
        getenv( "HTTP_ACCEPT" ),
        getenv( "HTTP_USER_AGENT" ),
        getenv( "HTTP_ACCEPT_ENCODING" ),
        getenv( "HTTP_ACCEPT_LANGUAGE" ) );
}

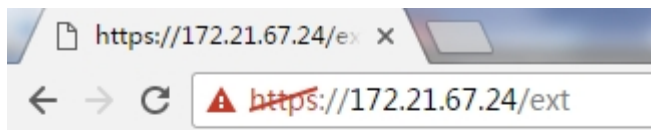
return 0;
}

```

Run the compiled program in the command line.


```
#make
#cd output/
#./example
```

5.5.3 Results



```
{
this is a fastcgi example
}
```

5.5.4 Precautions for Using the EDGELINK Platform

As the EdgeLink platform is encrypted, it cannot display the use case normally when used as above. Please use the pure platform version to test the use case.

5.6 SRAM Programming

5.6.1 Basic Instructions of SRAM

ADAM3600 serials use device node /dev/sram. The node size is 32K, wherein, 0-16K is used by EdgeLink, and 16-32K is reserved for customers.

ECU1051 serial uses device node /dev/fram. The node size is 128K, wherein, 0-64K is used by EdgeLink, and 64-128K is reserved for customers.

5.6.1.1 Write Operation of sram

The method to operate SRAM is similar to that of file operation. Use lseek to locate the required operation position, and then read and write sram.

The code of write operation is as below:

```
int write_with_file(int address, char buff[], int len)
```

```

{
int fd = open(devname, O_RDWR);
int nret = 0;
if (fd == -1)
{
perror("open");
return -1;
}

nret = lseek(fd, address, SEEK_SET);
if (nret < 0)
{
close(fd);
return 0;
}
nret = write(fd, buff, len);
close(fd);
return nret;
}

```

5.6.1.2 Read Operation of sram

Code of read operation is as below:

```

int read_with_file(int address, char buff[], int len)
{
int fd = open(devname, O_RDWR);
int nret = 0;
if (fd == -1)
{
perror("open");
return -1;
}

nret = lseek(fd, address, SEEK_SET);
if (nret < 0)

```

```
{
    close(fd);
    return 0;
}
nret = read(fd, buff, len);
close(fd);
return nret;
}
```