Contents

1. Introduction

1.1 System Characteristics

This product comes pre-installed with the TI AM335x-based Linux operating system. The kernel version is 3.12.10-rt15, BusyBox v1.22.1. Applications use POSIX standards(UNIX-like platforms). For system-specific hardware devices, the kernel provides an easy-to-use driver interface that accelerates application development.

The system software system is divided into three parts, namely Bootloader (Uboot), Linux kernel (3.12.10-rt15) and rootfs (busybox). UBoot is mainly booting the kernel, supporting NFS mount and NAND Flash boot; linux kernel is the bottom layer of the whole operating system, responsible for the whole hardware driver, and providing the core functions required by various systems; rootfs is the collection of system files.

2. Linux System Introduction

2.1 Partition Description

The onboard TF (MicroSD) card is currently divided into 4 parts:

/dev/mmcblk0p1 on /media/mmcblk0p1	Used to store Uboot files and Linux Kernel files.
/dev/mmcblk0p2 on /	System root partition
/dev/mmcblk0p3 on /media/mmcblk0p3	Recovery partition. Used to store temporary upgrade files.
/dev/mmcblk0p4 on /home	System user partition

2.2 Directory Description

/dev	Device node directory
/media	Multimedia directory
/proc	System configuration directory
/sys	System configuration directory
/var	Temporary directory
/bin	Command directory for ordinary users
/etc	Configuration file directory
/lib	Dynamic library directory
/mnt	Same as /media directory
/sbin	Root user command
/tmp	Temporary directory

/www Web directory
/home User data directory
/lost+found Temporarily stored directory of deleted files
/opt Configuration directory
/srv cgi command directory
/usr Ordinary user directory

2.3 System Function Description

sshd
ftpd
httpd

3. Development Environment

3.1 Hardware Requirements

3.2 Essential Software

3.2.1 Software Download

a) Ubuntu 12.04 32-bit
Download link:
http://old-releases.ubuntu.com/releases/precise/ubuntu-12.04.4-desktop-i386.iso

b) 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

c) Ubuntu12.04 installed on 64-bit version
Since TISDK is a 32-bit program, if you install the software on a 64-bit system, please download and install the 32-bit compatibility package first.
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
```

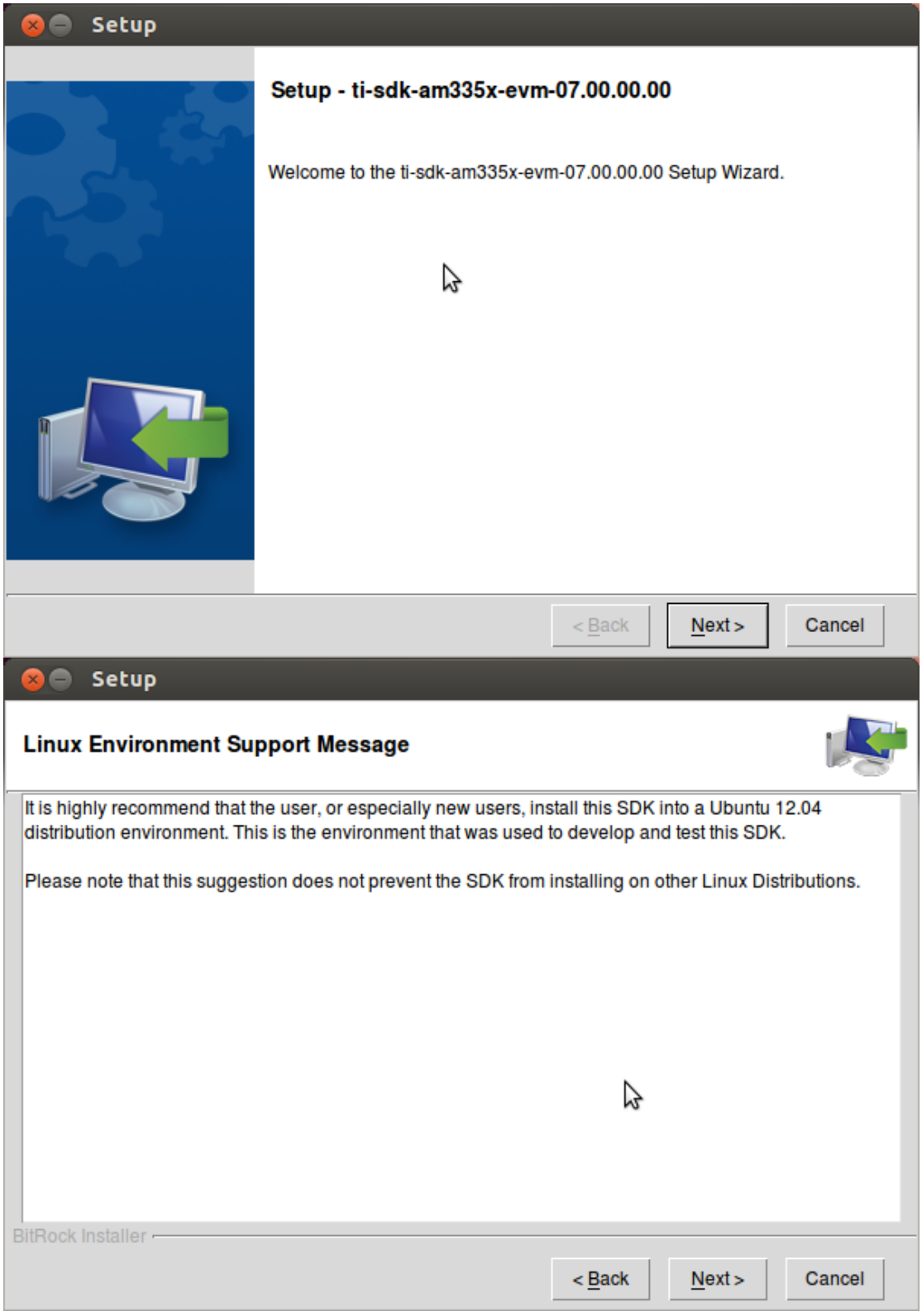
```
Compile the 32-bit program of the Uno platform
adv@adv-desktop:~ $apt install build-essential libc6:i386
libstdc++6:i386 gcc-multilib g++-multilib python doxygen
graphviz fp-utils-3.0.4 u-boot-tools zlib1g-dev:i386 cmake
zip  libssl-dev:i386  libcurl4-openssl-dev:i386
libxml2-dev:i386  libsqlite3-dev:i386
libmosquitto-dev:i386  unixodbc-dev:i386
libfcgi-dev:i386  libcap-dev:i386 uuid-dev:i386
```

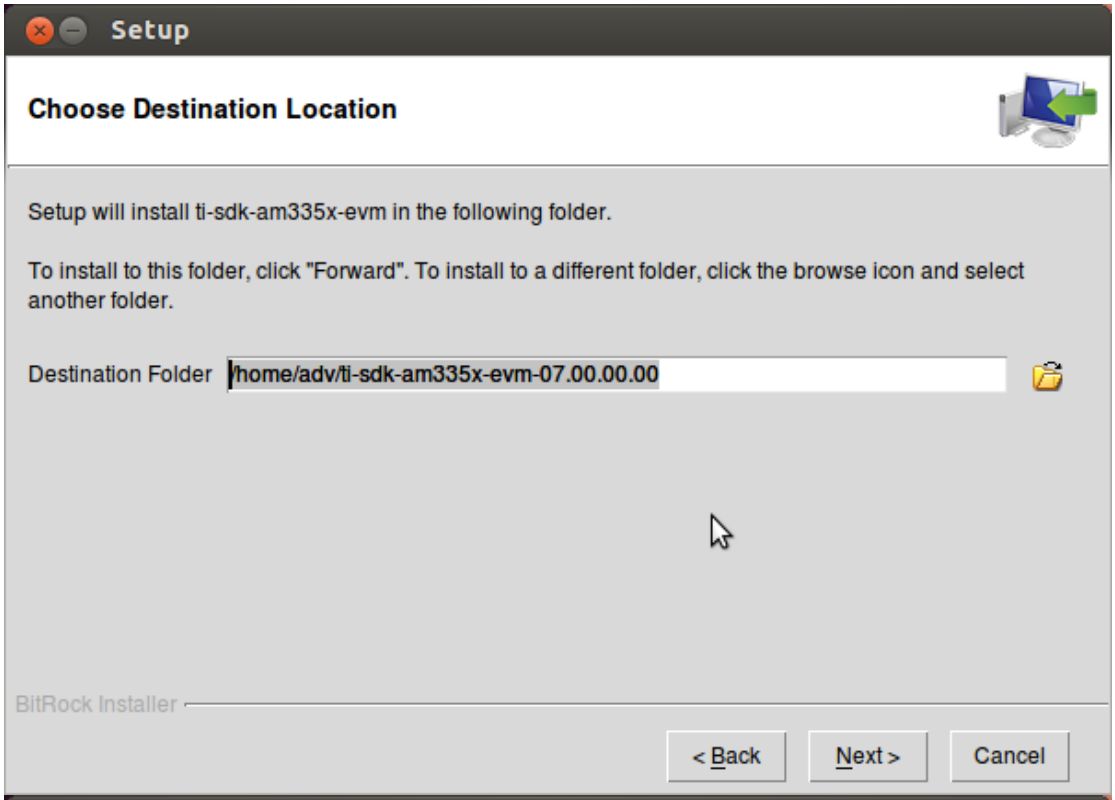
```
sudo apt-get install lib32ncurses5
lib32z1
```

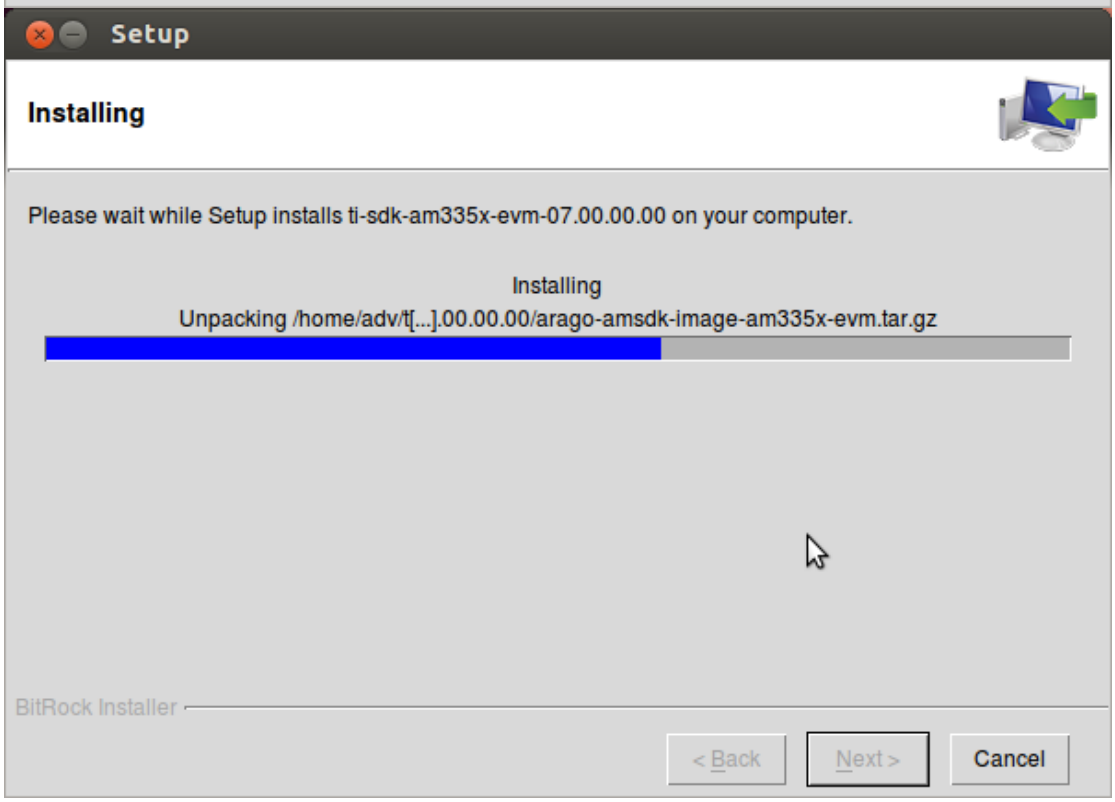
3.2.2 TISDK Software Installation

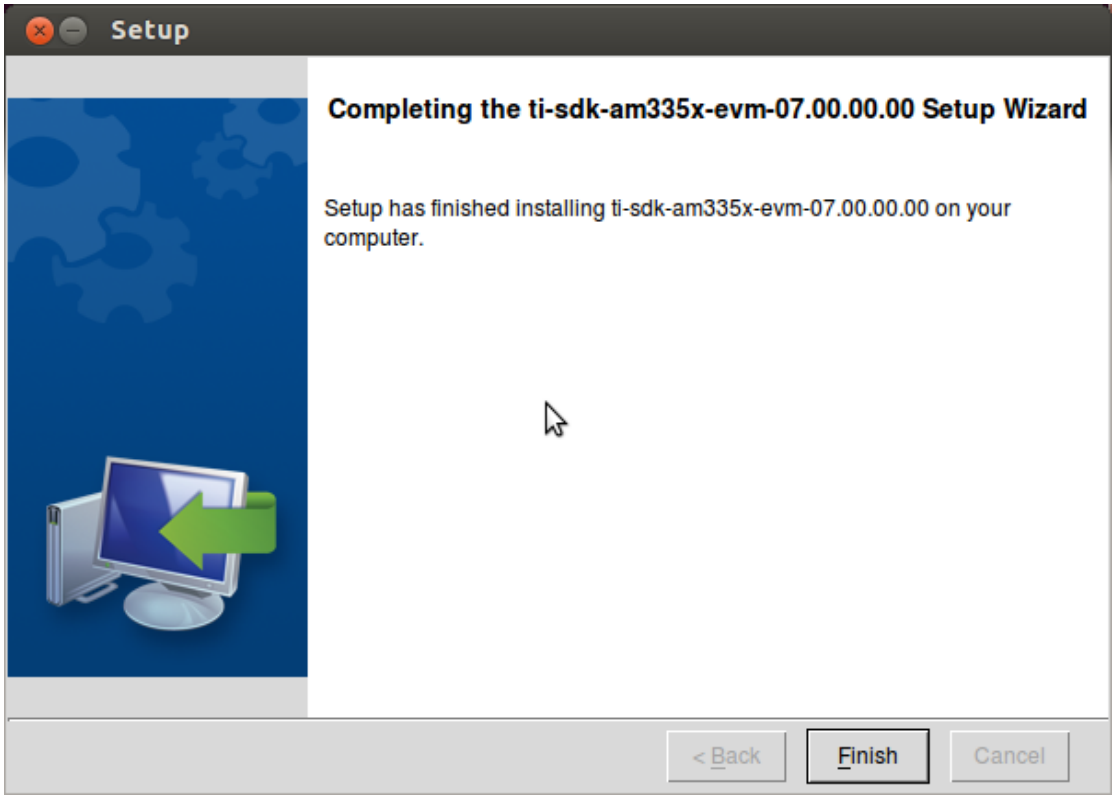
Add executable permissions to the file and execute the TISDK file.

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


Setup
Setup - ti-sdk-am335x-evm-07.00.00.00
Welcome to the ti-sdk-am335x-evm-07.00.00.00 Setup Wizard.
< Back
Next >
Cancel
Setup
Linux Environment Support Message
It is highly recommend that the user, or especially new users, install this SDK into a Ubuntu 12.04 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.
BitRock Installer
< Back
Next >
Cancel


Setup
Choose Destination Location
Setup will install ti-sdk-am335x-evm in the following folder.
To install to this folder, click "Forward". To install to a different folder, click the browse icon and select another folder.
Destination Folder
/home/adv/ti-sdk-am335x-evm-07.00.00.00
BitRock Installer
< Back
Next >
Cancel


Setup
Installing
Please wait while Setup installs ti-sdk-am335x-evm-07.00.00.00 on your computer.
Installing
Unpacking /home/adv/t[...].00.00.00/arago-amsdk-image-am335x-evm.tar.gz
BitRock Installer
< Back
Next >
Cancel



adv@adv-desktop:~ /Desktop $ cd
adv@adv-desktop:~$ cd ti-sdk-am335x-evm-07.00.00.00/
adv@adv-desktop:~/ti-sdk-am335x-evm-07.00.00.00$ ls
bin filesystem Makefile
board-support Graphics_SDK_setuplinux_hardfp_5_01_01_01.bin Rules.make
docs host-tools setup.sh
example-applications linux-devkit
adv@adv-desktop:~/ti-sdk-am335x-evm-07.00.00.00$sudo vi /etc/profile Add the PATH variable at the end of the file.
export PATH=/home/adv/ti-sdk-am335x-evm-07.00.00.00/linux-devkit/sysroots/i686-arago-linux/usr/bin:$PATH

adv@adv-desktop:~/ti-sdk-am335x-evm-07.00.00.00$source /etc/profile

3.2.3 TISDK Auxiliary Tool Installation

TISDK provides scripts that can automatically install and configure VSFTP, NFS and other auxiliary software, and run scripts for automatic installation and execution. This script is currently only available for Ubuntu 12.04. Execute the installation script and confirm that the system is connected to the network before installation. This process requires apt-get to install some packages.

adv@adv-desktop:~/ti-sdk-am335x-evm-07.00.00.00$ sudo ./setup.sh
Follow the prompts to complete the installation.

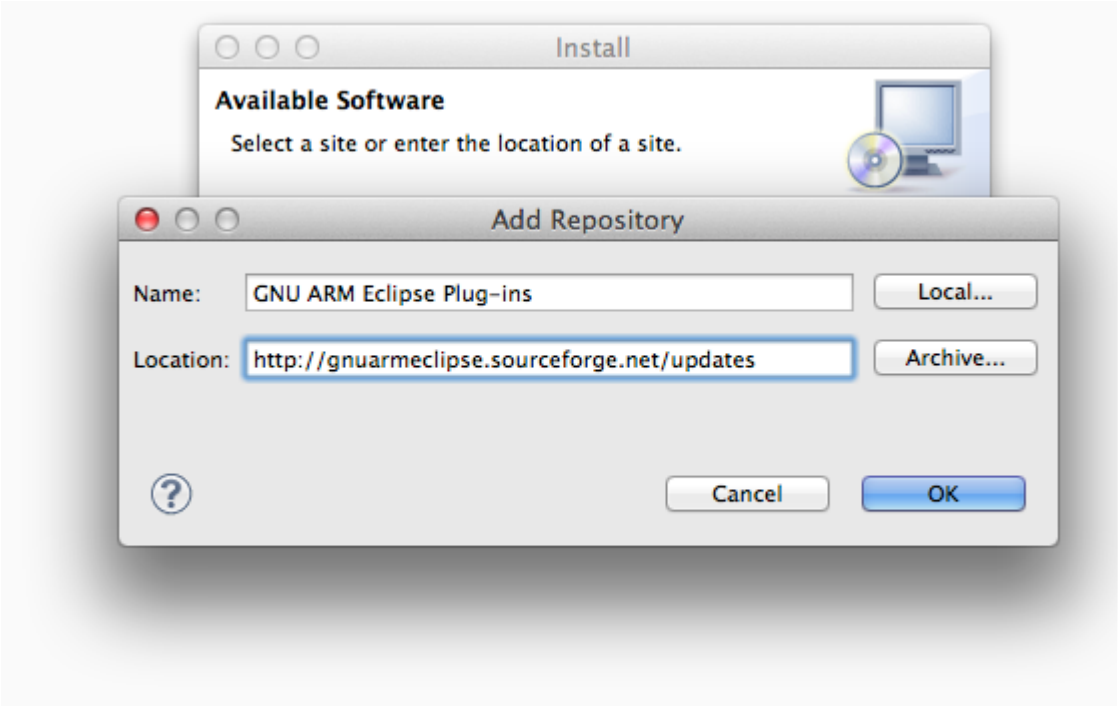
3.3 Available Software

a) Eclipse CDT + ARM GCC Plug-In
Download link：
https://eclipse.org/downloads/packages/eclipse-ide-cc-developers/neon3

b) Windows users need to install VirtualBox.
Download link：
https://www.virtualbox.org/

The Eclipse update site way

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

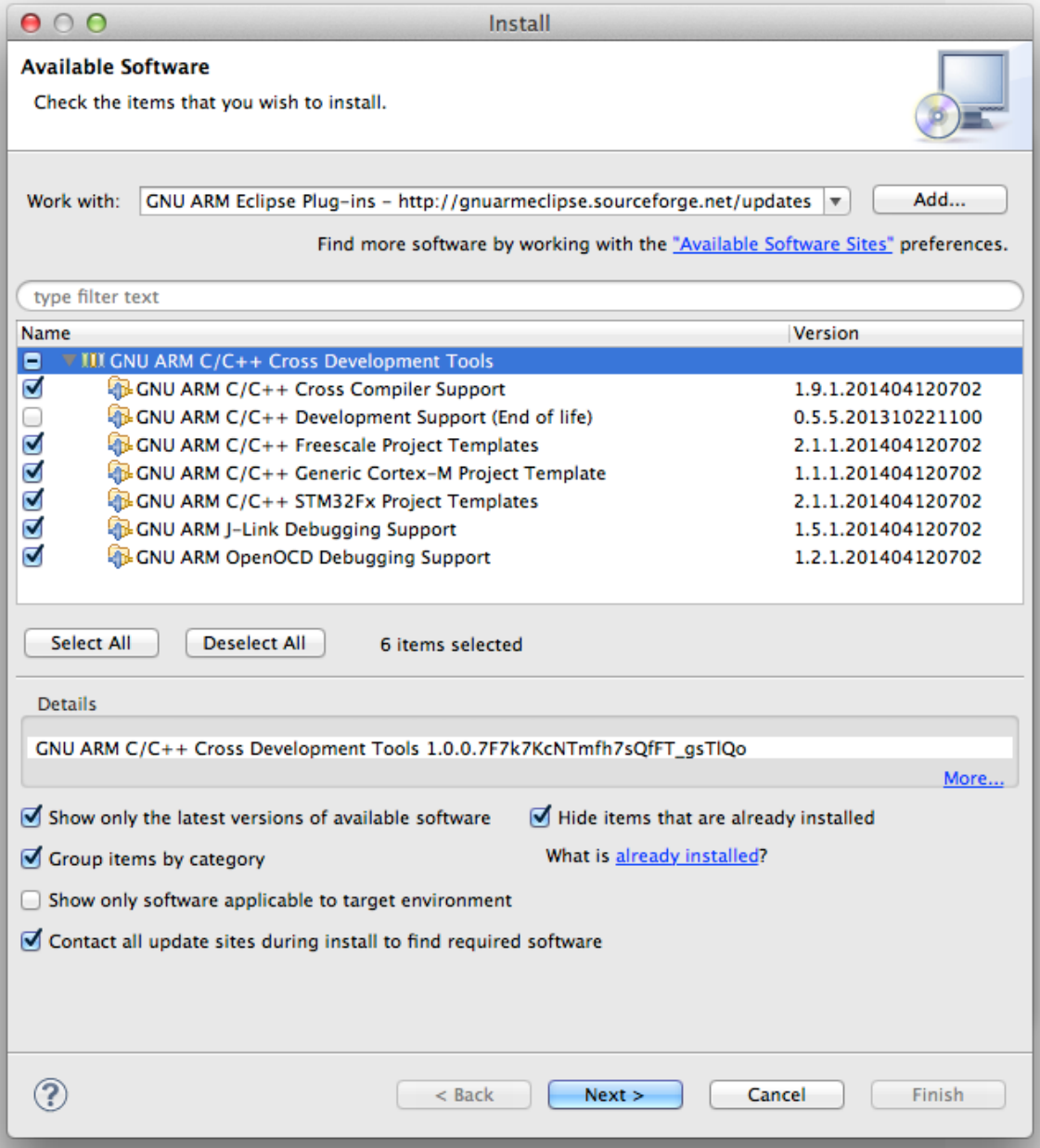


normally the main window should list a group named **CDT GNU Cross Development Tools**; expand it

(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 Settings

4.1 RTC Clock

4.1.1 RTC Clock Command

hwclock
Instructions：
hwclock -f /dev/rtc1 //Show current RTC time.
hwclock –s –f /dev/rtc1 //Synchronize the current RTC time to the Linux system time.
hwclock –w –f /dev/rtc1 //Synchronize the Linux system time to the RTC time.
hwclock -f /dev/rtc1 –localtime //RTC time is local time.
hwclock -f /dev/rtc1 –utc //RTC time is UTC time.
hwclock --hctosys -f /dev/rtc1 //Adjust the hardware clock to match the system clock.
hwclock --systohc -f /dev/rtc1 //Adjust the system clock to match the hardware clock.

Note: The rtc clocks of each platform have been uniformly soft-linked to /dev/rtc. Operating this node also provides direct access to clock information.

4.1.2 Time Zone Configuration

cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

4.2 NTP Timing

4.2.1 NTP Client

Synchronizing time with the ntpdate command can cause time hops and can affect some time-dependent programs and services, such as sleep, timer, and so on. The ntpd service can correct the cpu tick while correcting the time. The recommendation is to use ntpdate to force the synchronization time at boot time and then use the ntpd service to synchronize the time.

It should be noted that ntpd has a self-protection setting: If the difference between the local and the source time is too large, ntpd does not run. So the newly set time server must run

the ntpdate command firstly, update the initial value of the time, and then start the ntpd service. After the ntpd service runs, it synchronizes with the upstream server every 64 seconds. The time is adjusted according to the error value measured during each synchronization. As the error value decreases, the synchronization interval is gradually increased. This adjustment process is repeated for each beating.

General usage：

a) Use ntpdate to correct when booting.
#ntpdate -t 3 -u -s edu.ntp.org.cn
b) After booting up, use ntpd for micro-correction.
#/usr/sbin/ntpd

Ntpdate Timing

#ntpdate -t 3 -u -s edu.ntp.org.cn
-s The specified log is output to syslog instead of using standard output.
-t TimeOut Specifies the time to wait for a response. The value of the given TimeOut is rounded to a multiple of 0.2 seconds. The default is 1 second.
-u Specifies to send packets using an unprivileged port.

Ntpd Timing

/etc/ntp.conf configuration file, server address configuration see the red labeling section below.
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
```

4.2.2 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 nopee

# 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

4.2.3 Provide Time Service Only for Specified Network Segments

```
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
```

4.2.4 ntpd Related Commands

Query the time difference from the 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 is the time difference from the server. If the difference time is too large, please update it first by using the ntpdate command.

4.3 LAN setting

4.3.1 View Current Network Card Information

```
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)
```

4.3.2 Temporarily Configuration Network Card

```
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
```

4.3.3 Temporarily Use DHCP to Obtain 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)

4.3.4 Persistently Configure Fixed IP

Modify eth0 as a static IP address. Modify the relevant network card name file in the /etc/network/interfaces.d/ directory.
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

4.3.5 Persistently Configure Dynamic IP

Modify eth1 as a static IP address. Modify the related network card name file in the /etc/network/interfaces.d/ directory.
root@adam3600:~# **vi /etc/network/interfaces.d/eth1**

auto eth1
iface eth1 inet dhcp
allow-hotplug eth1

4.4 WIFI Configuration

4.4.1 Check If the Wireless Device Has Been Correctly Identified

The name of the interface of the wireless network card is generally 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)
```

4.4.2 Enable 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)
```

4.4.3 Scan for Searchable Wireless Networks

```
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:
2D1A6C181BFF000000000000000000000000000000000000000000
```

IE: Unknown:
3D160100130000000000000000000000000000000000000000
IE: Unknown: DD090010180201F0010000
IE: Unknown:
DD1E00904C336C181BFF000000000000000000000000000000000000000000000000
IE: Unknown:
DD1A00904C340100130000000000000000000000000000000000000000
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:
2D1AFE181BFFFF000001000000000000000000000000000000000000
IE: Unknown:
3D16020D000000000000000000000000000000000000000000000000
IE: Unknown: DD090010180205F02C0000
IE: WPA Version 1
Group Cipher : CCMP
Pairwise Ciphers (1) : CCMP
Authentication Suites (1) : PSK
IE: Unknown:
DD180050F2020101000003A4000027A4000042435E0062322F00

If you want to see if you have found the specified SSID, you can use the grep command to query the results.

root@adam3600:~# **iwlist wlan0 scan | grep WebAccess**

4.4.4 Modify the Configuration File (WPA2 Mode)

```
root@adam3600:~# vi /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1

# Only WPA-PSK is used. Any valid cipher combination is accepted.
network={
        ssid="WebAccess"
        scan_ssid=1
        proto=WPA2 WPA
        key_mgmt=WPA-PSK
        pairwise=CCMP TKIP
        group=CCMP TKIP WEP104 WEP40
        psk="password"
        priority=2
}
```

4.4.5 Modify the Configuration File (Open Mode)

```
root@adam3600:~# vi /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
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
}
```

4.4.6 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
```

4.4.7 Save the SSID Password Encrypted

The wpa_passphrase command parameter is the username and password.

```
root@adam3600:~#wpa_passphrase  max 1234567890
network={
        ssid="max"
        #psk="1234567890"
        psk=4b2bc7cbb3710e9ea43f09e8d57e8bdb3b2a2127af44960d73216c3612f6baba
}
```

Copy the psk= with encrypted password to 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
}
```

4.4.8 Configure 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

4.4.9 Configure Dynamic IP

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

auto wlan0
iface wlan0 inet dhcp
allow-hotplug wlan0

4.5 Configure WIFI as AP Mode

4.5.1 Configuration File

The Hostapd.conf configuration file is as follows. The 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

4.5.2 Start the Program

The command to start the program is:
#hostapd -B /home/root/hostap_wlan0.conf
ifconfig wlan0 192.168.1.34 netmask 255.255.255.0

4.5.3 Add Startup Item

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
```

4.6 DHCP Service

4.6.1 Configuration File

```
#vi /home/root/udhcpcd_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
```

4.6.2 Start Service

#/usr/sbin/udhcpd -S /home/root/udhcpcd_wlan0.conf

The 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)

4.7 Cellular Communication Configuration

The platform comes with a ppp program for dialing.

4.7.1 Check the Communication Module

Confirm that the module has been installed correctly and confirm the available serial port number according to the hardware manual.

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

```
[    0.000000] Kernel command line: console=ttyO0,115200n8 root=/dev/mmcblk0p2 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] userial_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

4.7.2 Dial Command

a) pppd dials directly

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

Pppd is the program name.

Call is the action parameter.

Default is the name of the calling script.

/dev/ttyUSB1 is the serial port parameter in the replacement script.

& means this program will run in the background.

b) Dial with the configured script.

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

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

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

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

A normal dialing prompt is as follows:

root@adam3600:~# wan.sh default

killall: pppd: no process killed

timeout set to 30 seconds

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
Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB1
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x79049dfc> <pcomp> <accomp>]
Warning - secret file /etc/ppp/pap-secrets has world and/or group access
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x28feffa7> <pcomp> <accomp>]
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x79049dfc> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 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

4.7.3 pppd Dial-up Script Configuration

root@adam3600:/etc/ppp/peers# ls default*
default default-chat-connect default-chat-disconnect

View the default script content.
root@adam3600:/etc/ppp/peers# cat default
debug
nodetach
/dev/ttyUSB1 #This parameter uses the serial port name by default. If the parameter of the command specifies a serial port, the serial port specified in the command parameter is used.
115200 #serial port baud rate
nocrtscts
lock
usepeerdns
noauth
noipdefault
novj
novjccomp
noccp
defaultroute
#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'

Check out the chat connect script. This script sends an AT command to the modem for configuration before dialing. If you have a command to query or set the configuration, you can add it in this script.

root@adam3600:/etc/ppp/peers# cat default-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 ""

ATZ factory reset
AT+CGDCONT setting environment
Value description:
<cid>： 1—4, PDP is the index value of setting environment. Other PDP-related commands can call the saved settings with this index value.
<PDP_type>：A string value indicating the type of exchange protocol.
Meaning of Value
IP IPv4 protocal
IPV6 IPv6 protocal
IPV4V6 IPv4/v6 protocal
PPP end-to-end protocol
<APN>：A string value indicating the access tag domain name to which the GGSN or external network is connected.

Operators	Access tag	Username& Password	Dial number	Remarks
China Mobile	cmnet cmwap	none	*99***1#	2.5G2.75G （GPRS）
China Mobile	cmnet cmwap	none	*98*1#	3G(TD-SCDMA)
China Unicom	3gnet	none	*99#	3G(WCDMA)
China Telecom	none	card card	#777	CDMA200

<PDP_addr>：A string value indicating the address of the MS.
<d_comp>：A numeric value that controls the compression of PDP data.

Value	Meaning
0	No compression
1	Compression

Note: Without <d_comp> means <d_comp> is 0.

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

Value	Meaning
0	No compression
1	Compression

Note: Without <h_comp> means <h_comp> is 0.

The chat-disconnect script is an 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 "\nGoodbay\n"

4.7.4 Common Operator Scripts

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 ""

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 ""
```

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 ""
```

4G LTE Dial

```
root@adam3600:/etc/ppp/peers# cat lte-chat-connect
TIMEOUT 30
ABORT "NO CARRIER"
ABORT "ERROR"
ABORT "NO DIALTONE"
ABORT "BUSY"
ABORT "NO ANSWER"
"" AT
OK AT+CFUN=1
OK AT+CGATT=1
OK AT+CGACT=1,1
OK AT+ZGACT=1,1
#CONNECT ""
```

4.7.5 ZTE ME3760 Module Configuration

The script can be called according to actual needs. Each module has its own available configuration serial port. Please modify the relevant parameters according to your own module.

```
root@adam3600:~# AutoDialup4G
netcard [eth2] not exist
Usage:
        AutoDialup4G com_port_name    netcard
        example: AutoDialup4G /dev/ttyUSB0    eth2
Configure the program parameter as needed.
root@adam3600:~# AutoDialup4G /dev/ttyUSB1 eth4
com port:/dev/ttyUSB1,netcard:eth4
[     0]AT
[     0]AT     OK
```

The commands sent automatically by this program are 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"
```

4.7.6 Debug Common AT Commands List.

Since each modem manufacturer has its own list of AT commands, only a few common commands are listed here for your reference. Generally, the minicom software is used to open the corresponding virtual serial port of the modem for operation. After opening the serial port, first send the AT command to see if there is an OK return to confirm whether the serial port is a configurable serial port.

```
~# minicom -D /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

Query the status of the SIM card

at+cpin?
+CPIN: READY

OK

Other reply parameters:
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
S**IM 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 passwordto 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

Query operator information

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 base station that is not connected to the operator returns the following information:

at+cops?

+COPS: 0

OK

Query phone number

at+cnum

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

OK

The normal return of this command is that the mobile phone number must be written in the sim card, otherwise the command will return ERROR.

[+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

Query signal strength

at+csq

+CSQ: 21,99

OK

+CSQ: <rssi>,<ber>

<rssi>

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

4.9 OpenVPN (To be client)

4.9.1 Using cert/key

a) Copy ca.crt，client.ovpn，client.crt，client.key，ta.key to /home/root/ovc

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          #Server IP and port
;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) Run the client

Note：The time of client should be the same as server's.

Running in foreground：#openvpn --config /home/root/ovc/client.ovpn

Running in background: #openvpn --daemon --config /home/root/ovc/client.ovpn

4.9.2 Using 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

4.9.3 Configure as auth-name

To modify the main configuration file of the openvpn service, add the following content: client-cert-not-required.
This parameter means that the client only uses the username and password to verify the login. If there is no such parameter, the client needs the certificate and the username and password to verify the login.
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

4.9.4 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

4.9.5 TAP Mode and TUN Mode

Users can modify the working mode by modifying the dev parameter directly in the configuration file.
Enable 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.

4.9.6 Add Execution Script

Add the following command to the client.ovpn script to use when the VPN is up or down.

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 the header #!/bin/sh, otherwise it will prompt that the execution failed.

4.10 Routing Table Configuration

4.10.1 View 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

4.10.2 Add 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

4.10.3 Delete 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:~#
```

4.10.4 Add the 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:~#
```

4.11 View 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:~#
```

4.12 Start 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

4.13 Start Telnet Service

由于 telnet
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

4.14 Start 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

4.15 View the Installed Driver

View the current installation module information.

```
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
```

Install module

```
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 the installed modules.

```
root@adam3600:~# rmmod boardio
```

4.16 Firewall Configuration

4.16.1 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
```

4.16.2 Set the White List

root@adam3600:~#**vi /home/sysuser/port_wihite.lst**

```
# port white list
443|tcp|all
345|tcp|eth0|172.0.0.0/8
```

The format is as follows:

Each line is a port setting parameter, separated by "|"

6001 is the port number

Tcp is the protocol, usually tcp or udp

All indicates all network cards, and you can also set a single network card, such as eth0, eth1.

172.0.0.0/8 is the network segment, which means that only the IP address of this network segment is accepted.

The black list format is the same as the white list, except that the relevant port is disabled to access the machine.

Since the iptables command needs to rely on multiple kernel modules when working properly, it is recommended to enable this function using the shell script we have already done.

The /usr/bin/firewall.sh script makes it easy to handle related dependencies.

4.16.3 Enable the Firewall

#**/usr/bin/firewall.sh /home/sysuser**

At this point the program will search for the configuration file in the /home/sysuser directory. Port_white.lst is a white list and port_black.lst is a black list.

4.16.4 Disable the Firewall

#**/usr/bin/firewall.sh stop** #Disable the firewall.

4.17 Web Server (lighttpd) Comfiguration

4.17.1 Configure the root directory

The Lighttpd configuration file is # /etc/lighttpd.conf. If necessary, you can modify this configuration file to configure the parameters you need. The default web directory is /home/sysuser/www.

4.17.2 Use Secure Link https

Currently, the system has its own https configuration file. If necessary, you can copy the /etc/lighttpd/lighttpd-https.conf file to the /home/sysuser/project/ directory.
mkdir /home/sysuser/project
cp /etc/lighttpd/lighttpd-https.conf /home/sysuser/project/

Generate a certificate file
openssl_gen_cert.sh
cat certificate.pem privatekey.pem > /home/sysuser/project/server.pem
cp certificate.pem /home/sysuser/project/ca.crt
After the configuration is complete, restart the system, and the https function is enabled.

4.18 Boot Self-start Configuration Method

Add the program that needs to be run at boot 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 & // “&” means working in background.
exit 0
~

Note: The executable permissions 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

After uploading the rc.local file to the system via FTP or other methods, the file may be missing executable permissions. In this case, you can add executable permissions through the chmod a+x /etc/rc.local command.

4.19 SNMP Configuration

4.19.1 SNMP support 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 every1,5,15 minutes .1.3.6.1.4.1.2021.10

4.19.2 ADVANTECH-IO-COMMON-MIB Introduction

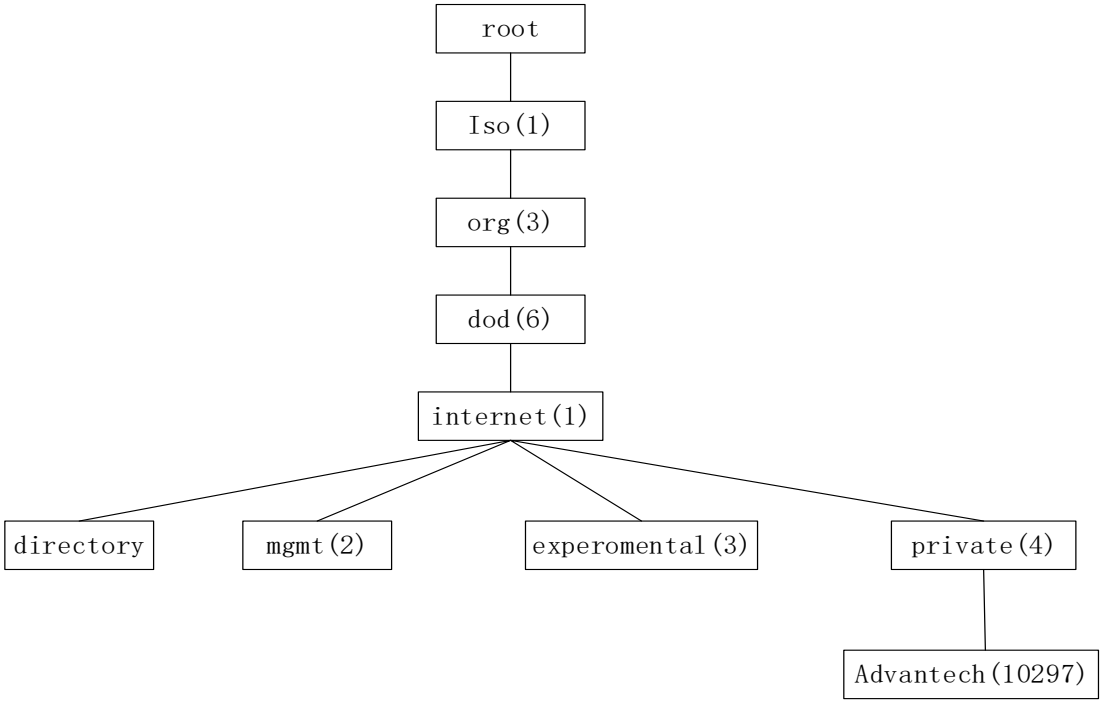


Figure1. OIDs tree

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

4.19.3 Create A snmpv3 User

Step1：

Execute the command on ADAM-5630:

#net-snmp-config --create-snmpv3-user -a "my_password" -X DES -A MD5 -x "my_password" myuser

Note: The SNMPv3 password is at least 8 characters long.

```
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 the /home/root/project/snmpd.conf file according to the prompt returned by the previous command:

```
root@adam5630:~# vi /home/root/project/snmpd.conf
```

Add the following:

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:
```

4.19.4 ADAM5630 Accesses Local SNMP Information

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::laErrMessage.1 = STRING:
UCD-SNMP-MIB::laErrMessage.2 = STRING:
UCD-SNMP-MIB::laErrMessage.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: 100000000
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, SNMPv2-MIB.

4.19.5 Obtain ADAM-5630 Local IO Module Information.

Support from ADVANTECH-IO-COMMON-MIB.mib is required. The placement path is:

```
root@adam5630:~# ls /home/root/project/
ADVANTECH-IO-COMMON-MIB.mib  ModbusDaemon.acr
```

Module information can be accessed via snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.10297.101. Support reading the following module information：

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 Thermpcouple input
	ADAM-5018P	7-ch Thermpcouple 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

4.19.6 Remote Access

Download the iReasoning MIB Browser for testing. Only SNMP v2 is supported in the personal version of the iReasoning MIB Browser.

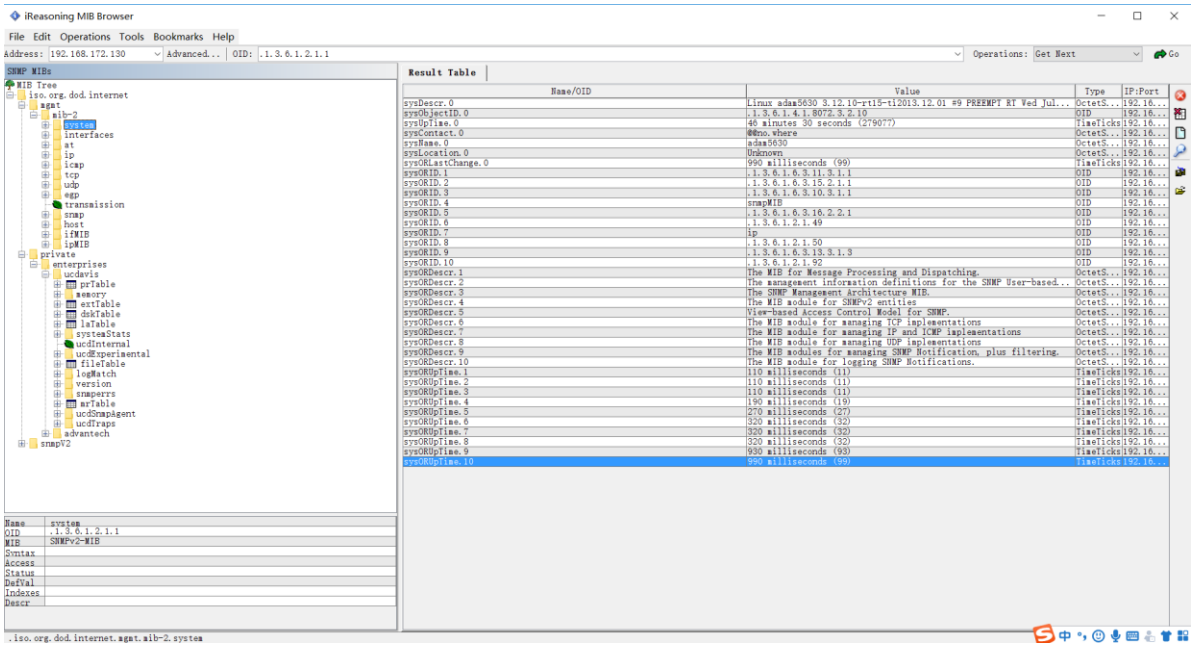


Figure2. iReasoning MIB Browser

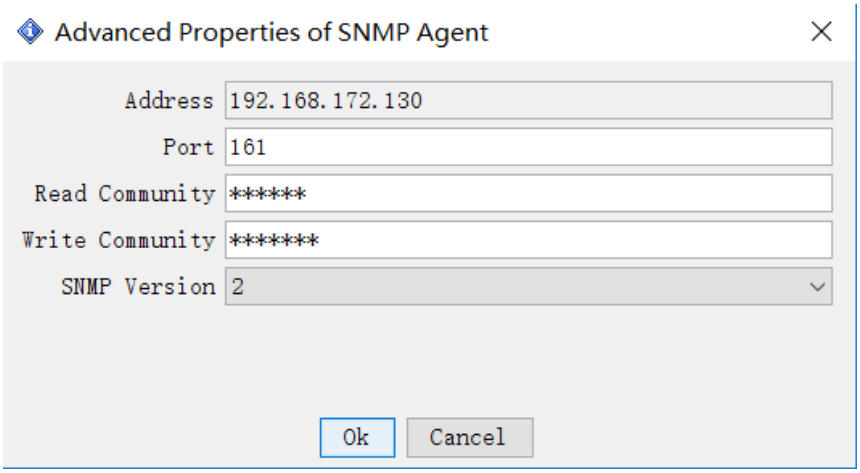


Figure3. Advanced.. Configuration

5. Program Development

5.1 Onboard Resource Programming (BoardResource SDK)

Refer to the BoardResource SDK.
http://support.advantech.com.cn/support/DownloadSRDetail_New.aspx?SR_ID=1-1V0VHE0&Doc_Source=Download

5.2 I/O programming (I/O SDK)

Refer to the ADAM5630_IO_SDK.
http://support.advantech.com.cn/support/DownloadSRDetail_New.aspx?SR_ID=1-1V0VHE0&Doc_Source=Download

5.3 Serial Port programming

5.3.1 Basic Steps of Programming

The serial port under Linux is mainly operated by setting struct termios. The steps are generally as follows:

a) Open a device node, such as /dev/ttyAP0.
b) Get the struct termios information of the node, set the serial port parameters such as baud rate, and then save the struct termios information.
c) Call the read and write functions to read and write data.
d) Close the device node.

5.3.2 Parameter Configuration Method

a) Open the device node

```
fd = open(Dev, O_RDWR | O_NOCTTY);
```

b) Set the baud rate

```
cfsetispeed(&opt, B9600);
cfsetospeed(&opt, B9600);
```

c) Set the data bit

```
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 the stop bit

```
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 the parity bit

```
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 Introduction of Other Parameters

In general, the most basic settings in the serial port include baud rate setting, parity bit and stop bit setting. The setting of the serial port is mainly to set the member values of the struct termios structure, as shown below:

```
struct termio
{
    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 */
};
```

The most important in this structure is c_cflag. By assigning it, users can set the baud rate, character size, data bit, stop bit, parity bit, hardware flow control, and so on. In addition, c_iflag and c_cc are also commonly used flags. The three members are mainly explained here.

Constant names supported by **c_cflag**:

CBAUD	bit mask of the baud rate
B0	0 baud rate (abandon DTR)
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	bit mask of data bits

CS5	5 data bits
CS6	6 data bits
CS7	7 data bits
CS8	8 data bits
CSTOPB	2 stop bits (not set to 1 stop bit)
CREAD	receive enable
PARENB	parity bit enable
PARODD	use odd parity instead of even parity
HUPCL	hang up the line when you close it (abandon DTR)
CLOCAL	local connection (do not change the port owner)
LOBLK	block job control output
CNET_CTSRTS	hardware flow control enable

Constant names supported by **c_iflag**

INPCK	parity check enable
IGNPAR	ignore parity errors
PARMRK	parity error mask
ISTRIP	remove parity
IXON	start export hardware flow control
IXOFF	start import software flow control
IXANY	allow characters to restart flow control
IGNBRK	ignore interrupts
BRKINT	send SIGINT signal when an interrupt occurs
INLCR	map NL to CR
IGNCR	ignore CR
ICRNL	map CP to NL
IUCLC	map high-level cases to low-level cases
IMAXBEL	reply to ECHO when the input is too long

c_cc Constant names supported by **c_cc**

VINTR	interrupt control, the corresponding key is CTRL+C
VQUIT	exit the operation, the corresponding key is CRTL+Z
VERASE	delete the operation, the corresponding key is Backspace (BS)
VKILL	Delete the line, the corresponding key is CTRL+U
VEOF	at the end of the file, the corresponding key is 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	specifies the minimum number of characters read
VTIME	specifies the wait time for reading each character

Serial port control function

tcgetattr	get property (termios structure)
tcsetattr	set property (termios structure)
cfgetispeed	get input speed

cfgetospeed	get output speed
cfsetispeed	set input speed
cfsetospeed	set output speed
tcdrain	wait for all outputs to be transmitted
tcflow	suspend transmission or receive
tcflush	clear pending input and/or output
tcsendbreak	send BREAK character

5.4 Network Programming

5.4.1 TCP Communication

TCP-based (connection oriented) socket programming is divided into client and server.
The client's process is as follows:

（1）Create a socket (socket)
（2）Send a connection request to the server (connect)
（3）Communicate with the server (send/recv)
（4）Close the socket

```
/*************************************************************************
 > File Name: tcpc.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("useage:./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 server-side process is as follows:

（1）Create a socket.(socket)

（2）Bind the socket to a local address and port. (bind)

（3）Set the socket to listen mode, ready to receive client request. (listen)

（4）Waiting for the customer request to arrive. When the request comes in, accept the connection request and return a new socket corresponding to the connection. (accept)

（5）Communicate with the client using the returned socket. (send/recv)

（6）Return and wait for another customer 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 specia in listening tcp connect
//bind , the ip and port information is aready in the sockaddr
	if (-1 == bind(sockfd, (struct sockaddr*) &serv_addr,
				sizeof(struct sockaddr)))
	{
		perror("bind error\n");
		exit(0);
	}
	printf("bind seccessful\n");

	if (-1 == listen(sockfd, BACKLOG))
	{
		perror("lisenning");
		exit(1);
	}
	printf("the server is listenning...\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 recvbytes; //the number of bytes receive from socket
                recvbytes = recv(sockfd_client, buf, MAXBUFSIZE, 0);
                if (-1 == recvbytes)
                {
                        perror("receive");
                        exit(1);
                }
                printf("%d bytes receive from connect:%s\n", recvbytes, buf);
                if(recvbytes > 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
 *
 * Main implementation: send a 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("useage:./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);

	/* creat 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：（receiver）

1) Create a socket. (socket)
2) Bind the socket to a local address and port. (bind)
3) Communicate with the client using the returned socket. (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 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 socket */
	int server_socket_fd = socket(AF_INET, SOCK_DGRAM, 0);
	if (server_socket_fd == -1)
	{
		perror("Create Socket Failed:");
		exit(1);
	}

	/* bind 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 the fastcgi feature by default, and can be extended directly by programming if needed.

5.5.1 Configure Lighttpd

Configure the fastcgi.server parameter at the end of the /etc/lighttpd.conf file. Modify as follows:

#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",
      )
   )
)
```

5.5.2 Code

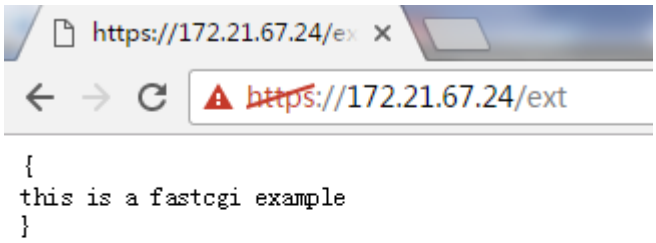
```
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include <locale.h>
#include <dirent.h>
#include <fcntl.h>
#include <shadow.h>
#include <dlfcn.h>
#include <unistd.h>
```

```
#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 );
		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;
}
```

5.5.3 Result Display：



6. To be Modbus Server

ADAM-5630 is running as Modbus server by default.

1. Configuration Steps

2. To confirm your image version is not lower than 2.3.0.
3. The configure file of Modbus server:
   /home/root/project/ModbusDaemon.acr

```
root@adam5630:~/project# cat ModbusDaemon.acr
<?xml version="1.0" encoding="utf-8"?>
<ModbusConfig>
        <ModbusServer  polling_cycle="1000" addr_type="0">
                <ModbusPortTCP id="1" timeout="3000" idle_time="120" ip="::" port="502" max_conn="4" gateway_port="" />
                <ModbusPortRTU id="2" slave_address="1" timeout="3000" port="4" baudrate="9600" byte_size="8" stop_bits="1" parity="0" />
                <AddressMappingList>
                        <AddressMappingItem slot_index="0" channel_index="0" modbus_address="00001" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="1" modbus_address="00002" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="2" modbus_address="00003" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="3" modbus_address="00004" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="4" modbus_address="00005" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="5" modbus_address="00006" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="6" modbus_address="00007" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="7" modbus_address="00008" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="8" modbus_address="00009" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="9" modbus_address="00010" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="10" modbus_address="00011" little_endian="false"/>
                        <AddressMappingItem slot_index="0" channel_index="11" modbus_address="00012" little_endian="false"/>
```

Modbus TCP:

<ModbusPortTCP id="1" timeout="3000" idle_time="120" ip="::" port="502" max_conn="4" gateway_port="" />

You can modify the port and the max connection of Modbus server.

Modbus RTU:

<ModbusPortRTU id="2" slave_address="1" timeout="3000" port="4" baudrate="9600" byte_size="8" stop_bits="1" parity="0" />

ModbusPortRTU id: The COM number of Modbus server in ADAM-5630. (2 means using COM2.)

Slave address: The device id of Modbus server.

The ”address mapping” is an extended function. It haven’t been supported so far. Stay default is ok.

4. The address of Modbus server tags.

   The offset address is beginning from 1 from slot1.

   For example, ADAM-5017 is in slot2. The channal0 of ADAM-5017 is 30017.

   Please refer to the form below

1、 Modbus 0X Coils Status：

	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
Bit0	00001	00033	00065	00097	00129	00161	00193	00225
Bit1	00002	00034	00066	00098	00130	00162	00194	00226
Bit2	00003	00035	00067	00099	00131	00163	00195	00227
Bit3	00004	00036	00068	00100	00132	00164	00196	00228
Bit4	00005	00037	00069	00101	00133	00165	00197	00229
Bit5	00006	00038	00070	00102	00134	00166	00198	00230
Bit6	00007	00039	00071	00103	00135	00167	00199	00231
Bit7	00008	00040	00072	00104	00136	00168	00200	00232
Bit8	00009	00041	00073	00105	00137	00169	00201	00233
Bit9	00010	00042	00074	00106	00138	00170	00202	00234
Bit10	00011	00043	00075	00107	00139	00171	00203	00235
Bit11	00012	00044	00076	00108	00140	00172	00204	00236
Bit12	00013	00045	00077	00109	00141	00173	00205	00237
Bit13	00014	00046	00078	00110	00142	00174	00206	00238
Bit14	00015	00047	00079	00111	00143	00175	00207	00239
Bit15	00016	00048	00080	00112	00144	00176	00208	00240
Bit16	00017	00049	00081	00113	00145	00177	00209	00241
Bit17	00018	00050	00082	00114	00146	00178	00210	00242
Bit18	00019	00051	00083	00115	00147	00179	00211	00243
Bit19	00020	00052	00084	00116	00148	00180	00212	00244
Bit20	00021	00053	00085	00117	00149	00181	00213	00245
Bit21	00022	00054	00086	00118	00150	00182	00214	00246

Bit22	00023	00055	00087	00119	00151	00183	00215	00247
Bit23	00024	00056	00088	00120	00152	00184	00216	00248
Bit24	00025	00057	00089	00121	00153	00185	00217	00249
Bit25	00026	00058	00090	00122	00154	00186	00218	00250
Bit26	00027	00059	00091	00123	00155	00187	00219	00251
Bit27	00028	00060	00092	00124	00156	00188	00220	00252
Bit28	00029	00061	00093	00125	00157	00189	00221	00253
Bit29	00030	00062	00094	00126	00158	00190	00222	00254
Bit30	00031	00063	00095	00127	00159	00191	00223	00255
Bit31	00032	00064	00096	00128	00160	00192	00224	00256

2、 Modbus 1X Input Status：

	Slot 0	**Slot 1**	**Slot 2**	**Slot 3**	**Slot 4**	**Slot 5**	**Slot 6**	**Slot 7**
Bit0	10001	10033	10065	10097	10129	10161	10193	10225
Bit1	10002	10034	10066	10098	10130	10162	10194	10226
Bit2	10003	10035	10067	10099	10131	10163	10195	10227
Bit3	10004	10036	10068	10100	10132	10164	10196	10228
Bit4	10005	10037	10069	10101	10133	10165	10197	10229
Bit5	10006	10038	10070	10102	10134	10166	10198	10230
Bit6	10007	10039	10071	10103	10135	10167	10199	10231
Bit7	10008	10040	10072	10104	10136	10168	10200	10232
Bit8	10009	10041	10073	10105	10137	10169	10201	10233
Bit9	10010	10042	10074	10106	10138	10170	10202	10234
Bit10	10011	10043	10075	10107	10139	10171	10203	10235
Bit11	10012	10044	10076	10108	10140	10172	10204	10236
Bit12	10013	10045	10077	10109	10141	10173	10205	10237
Bit13	10014	10046	10078	10110	10142	10174	10206	10238
Bit14	10015	10047	10079	10111	10143	10175	10207	10239
Bit15	10016	10048	10080	10112	10144	10176	10208	10240
Bit16	10017	10049	10081	10113	10145	10177	10209	10241
Bit17	10018	10050	10082	10114	10146	10178	10210	10242
Bit18	10019	10051	10083	10115	10147	10179	10211	10243
Bit19	10020	10052	10084	10116	10148	10180	10212	10244
Bit20	10021	10053	10085	10117	10149	10181	10213	10245
Bit21	10022	10054	10086	10118	10150	10182	10214	10246
Bit22	10023	10055	10087	10119	10151	10183	10215	10247
Bit23	10024	10056	10088	10120	10152	10184	10216	10248
Bit24	10025	10057	10089	10121	10153	10185	10217	10249
Bit25	10026	10058	10090	10122	10154	10186	10218	10250
Bit26	10027	10059	10091	10123	10155	10187	10219	10251
Bit27	10028	10060	10092	10124	10156	10188	10220	10252
Bit28	10029	10061	10093	10125	10157	10189	10221	10253

Bit29	10030	10062	10094	10126	10158	10190	10222	10254
Bit30	10031	10063	10095	10127	10159	10191	10223	10255
Bit31	10032	10064	10096	10128	10160	10192	10224	10256

3、 Modbus 3X Holding

	Slot 0	**Slot 1**	**Slot 2**	**Slot 3**	**Slot 4**	**Slot 5**	**Slot 6**	**Slot 7**
Word0	40001	40017	40033	40049	40065	40081	40097	40113
Word 1	40002	40018	40034	40050	40066	40082	40098	40114
Word 2	40003	40019	40035	40051	40067	40083	40099	40115
Word 3	40004	40020	40036	40052	40068	40084	40100	40116
Word 4	40005	40021	40037	40053	40069	40085	40101	40117
Word 5	40006	40022	40038	40054	40070	40086	40102	40118
Word 6	40007	40023	40039	40055	40071	40087	40103	40119
Word 7	40008	40024	40040	40056	40072	40088	40104	40120
Word 8	40009	40025	40041	40057	40073	40089	40105	40121
Word 9	40010	40026	40042	40058	40074	40090	40106	40122
Word 10	40011	40027	40043	40059	40075	40091	40107	40123
Word 11	40012	40028	40044	40060	40076	40092	40108	40124
Word 12	40013	40029	40045	40061	40077	40093	40109	40125
Word 13	40014	40030	40046	40062	40078	40094	40110	40126
Word 14	40015	40031	40047	40063	40079	40095	40111	40127
Word 15	40016	40032	40048	40064	40080	40096	40112	40128

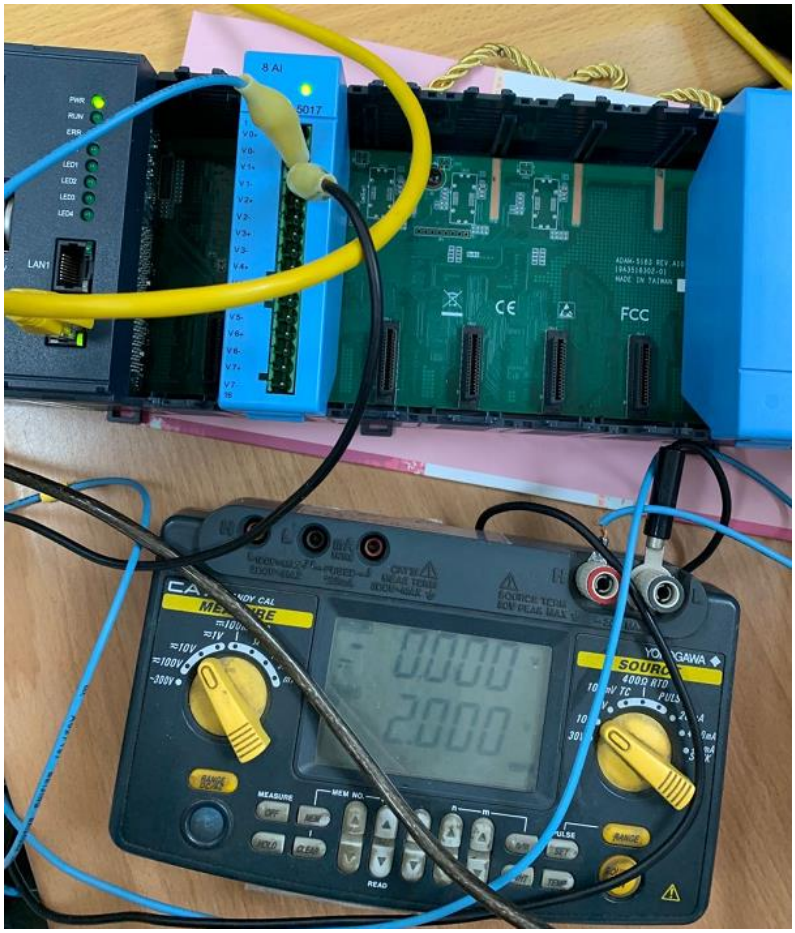
4、 Modbus 4X Input

	Slot 0	**Slot 1**	**Slot 2**	**Slot 3**	**Slot 4**	**Slot 5**	**Slot 6**	**Slot 7**
Word 0	30001	30017	30033	30049	30065	30081	30097	30113
Word 1	30002	30018	30034	30050	30066	30082	30098	30114
Word 2	30003	30019	30035	30051	30067	30083	30099	30115
Word 3	30004	30020	30036	30052	30068	30084	30100	30116
Word 4	30005	30021	30037	30053	30069	30085	30101	30117
Word 5	30006	30022	30038	30054	30070	30086	30102	30118
Word 6	30007	30023	30039	30055	30071	30087	30103	30119
Word 7	30008	30024	30040	30056	30072	30088	30104	30120
Word 8	30009	30025	30041	30057	30073	30089	30105	30121
Word 9	30010	30026	30042	30058	30074	30090	30106	30122
Word 10	30011	30027	30043	30059	30075	30091	30107	30123
Word 11	30012	30028	30044	30060	30076	30092	30108	30124
Word 12	30013	30029	30045	30061	30077	30093	30109	30125
Word 13	30014	30030	30046	30062	30078	30094	30110	30126
Word 14	30015	30031	30047	30063	30079	30095	30111	30127
Word 15	30016	30032	30048	30064	30080	30096	30112	30128

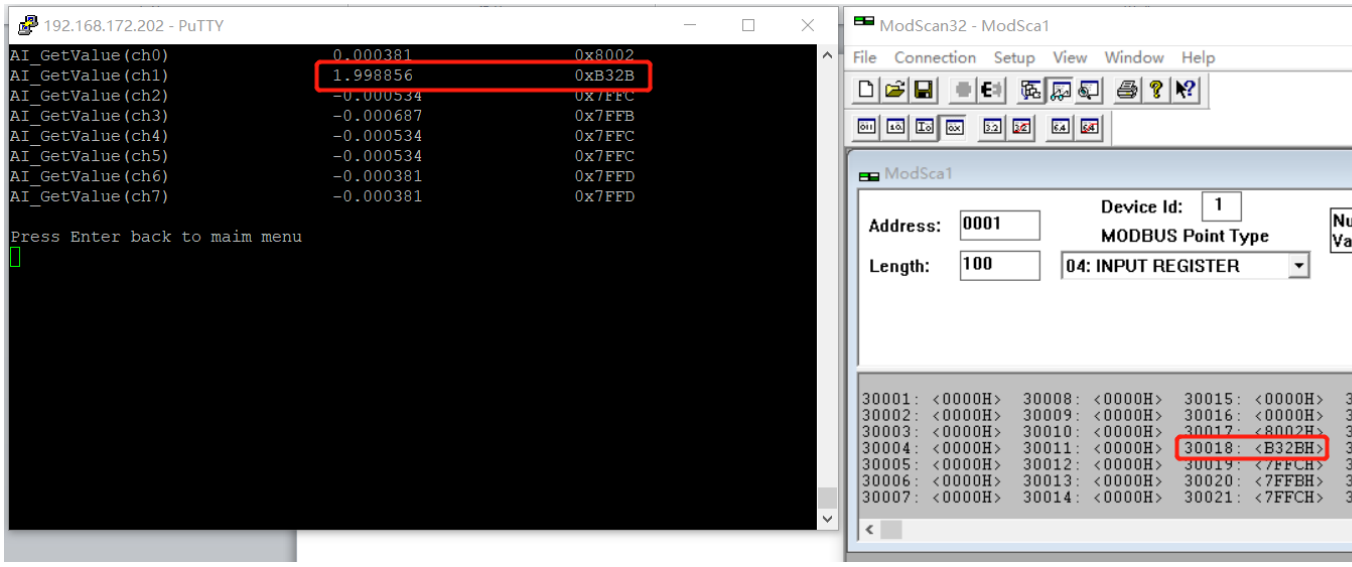
2. Example

In my example, ADAM-5017 is in slot1 and I have set the range -5V to 5V to channel 1 of ADAM-5017.

Input 2V to Channal1.



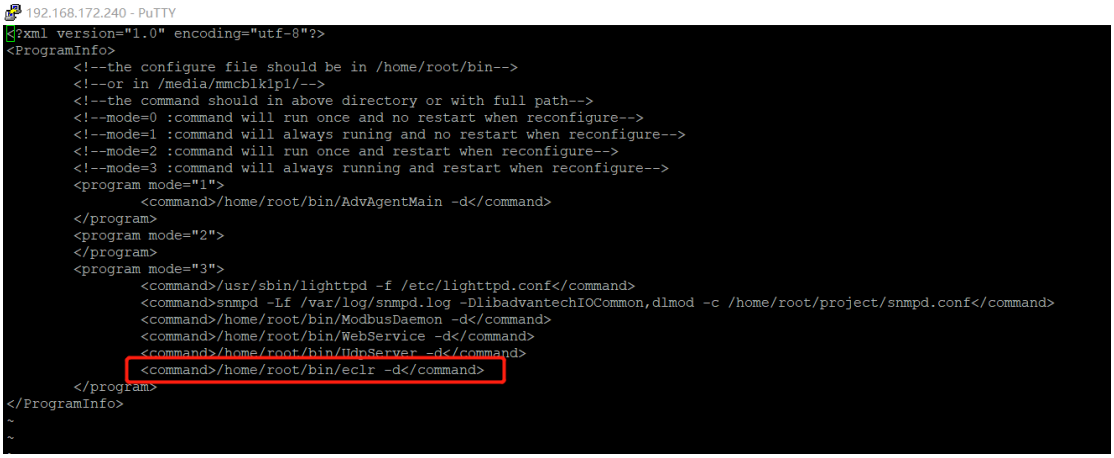
Use modscan to see the result.



7. KW-Software (MULTIPROG)

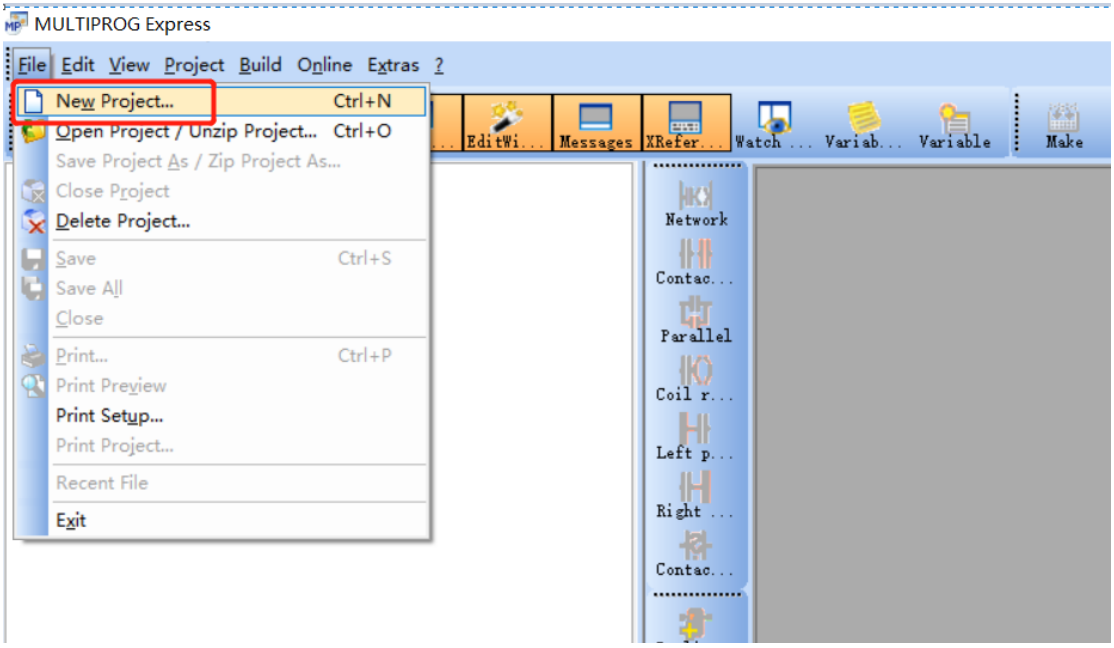
1. Start eclr in ADAM-5630

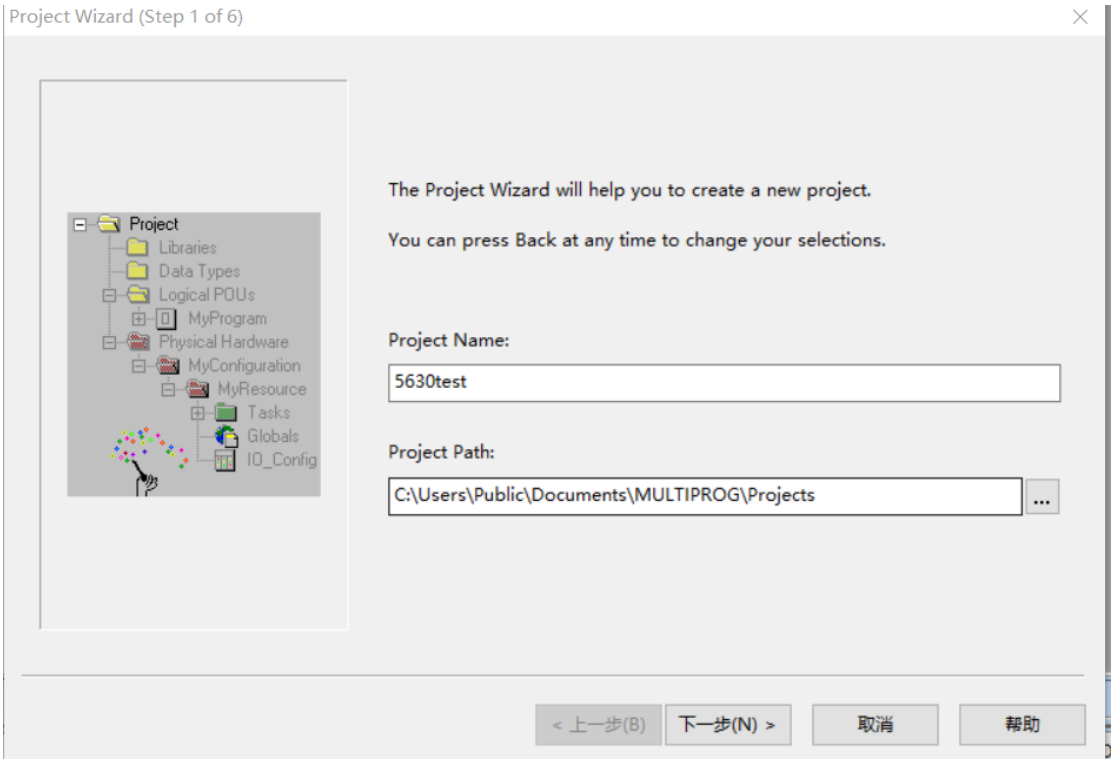
1. Execute /home/root/bin/eclr –d
2. Add <command>/home/root/bin/eclr -d</command> to /home/root/project/AdvProgramMgr.acr to make it to be as boot start program.

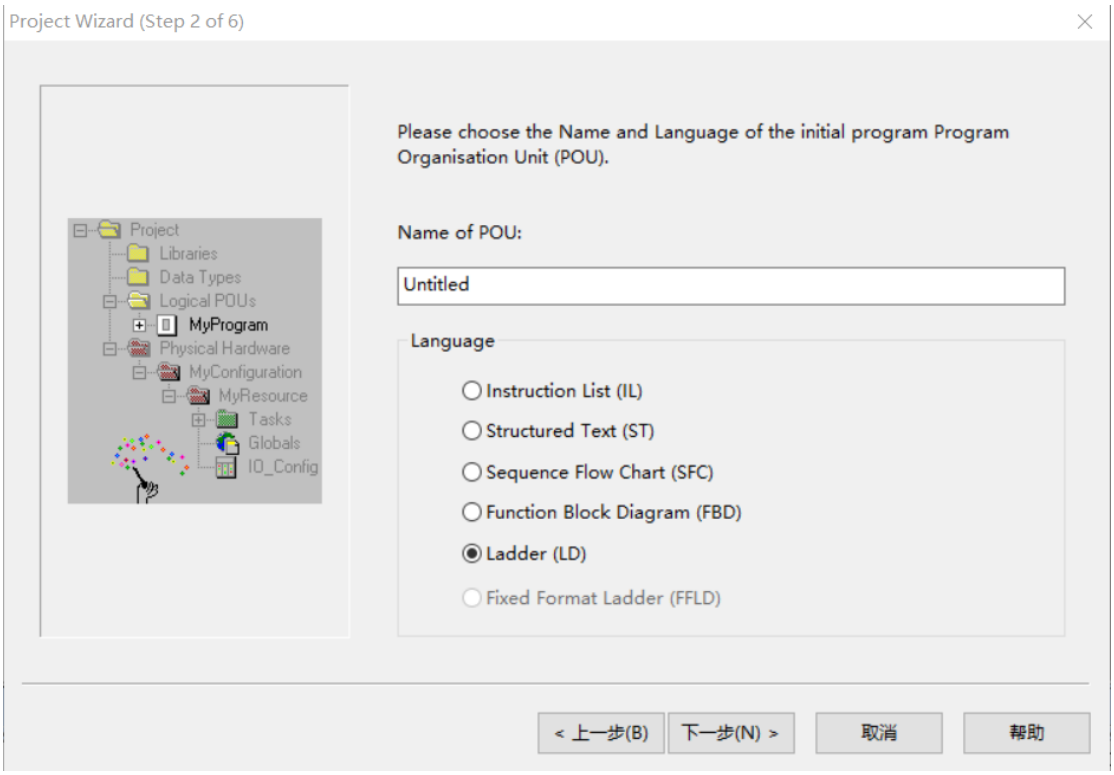


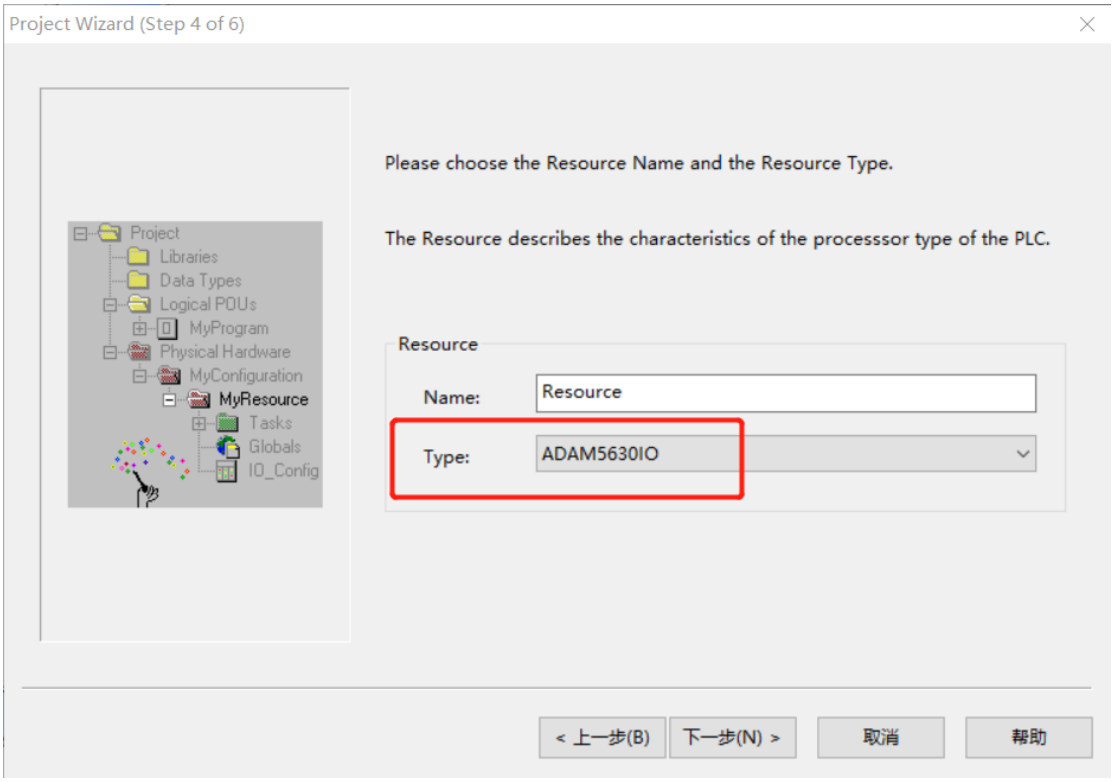
2. Create and download kw program

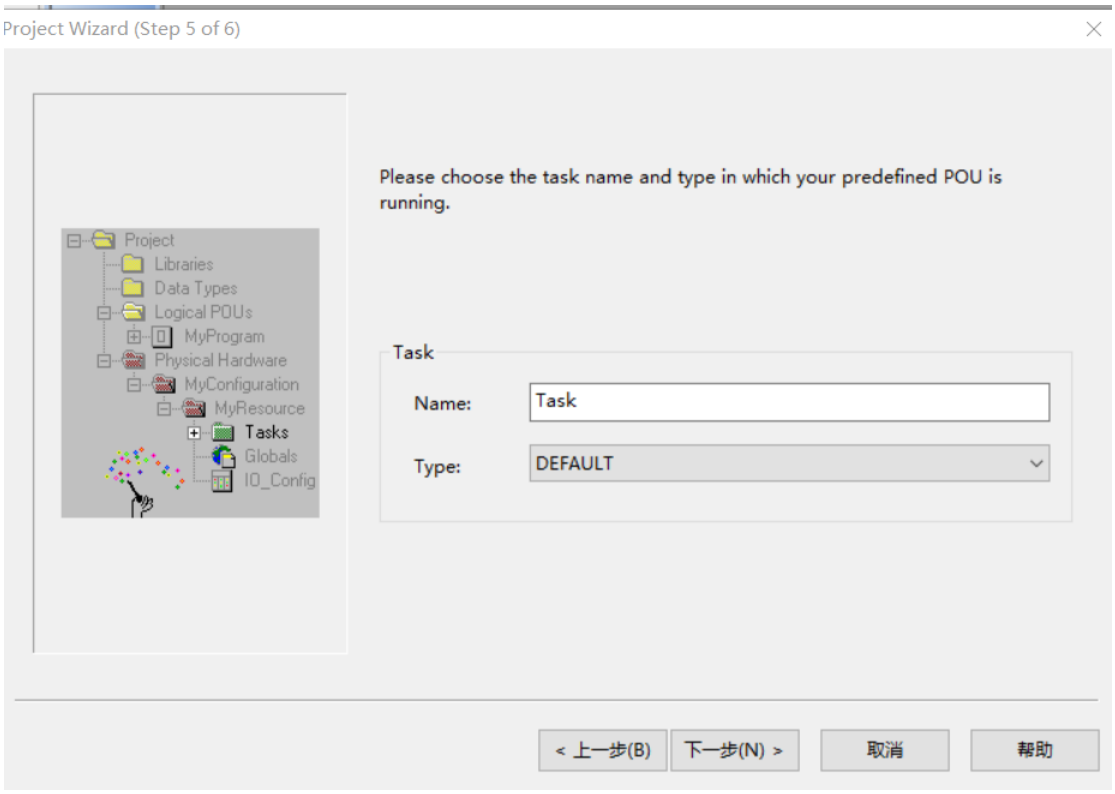
1）Create a new project



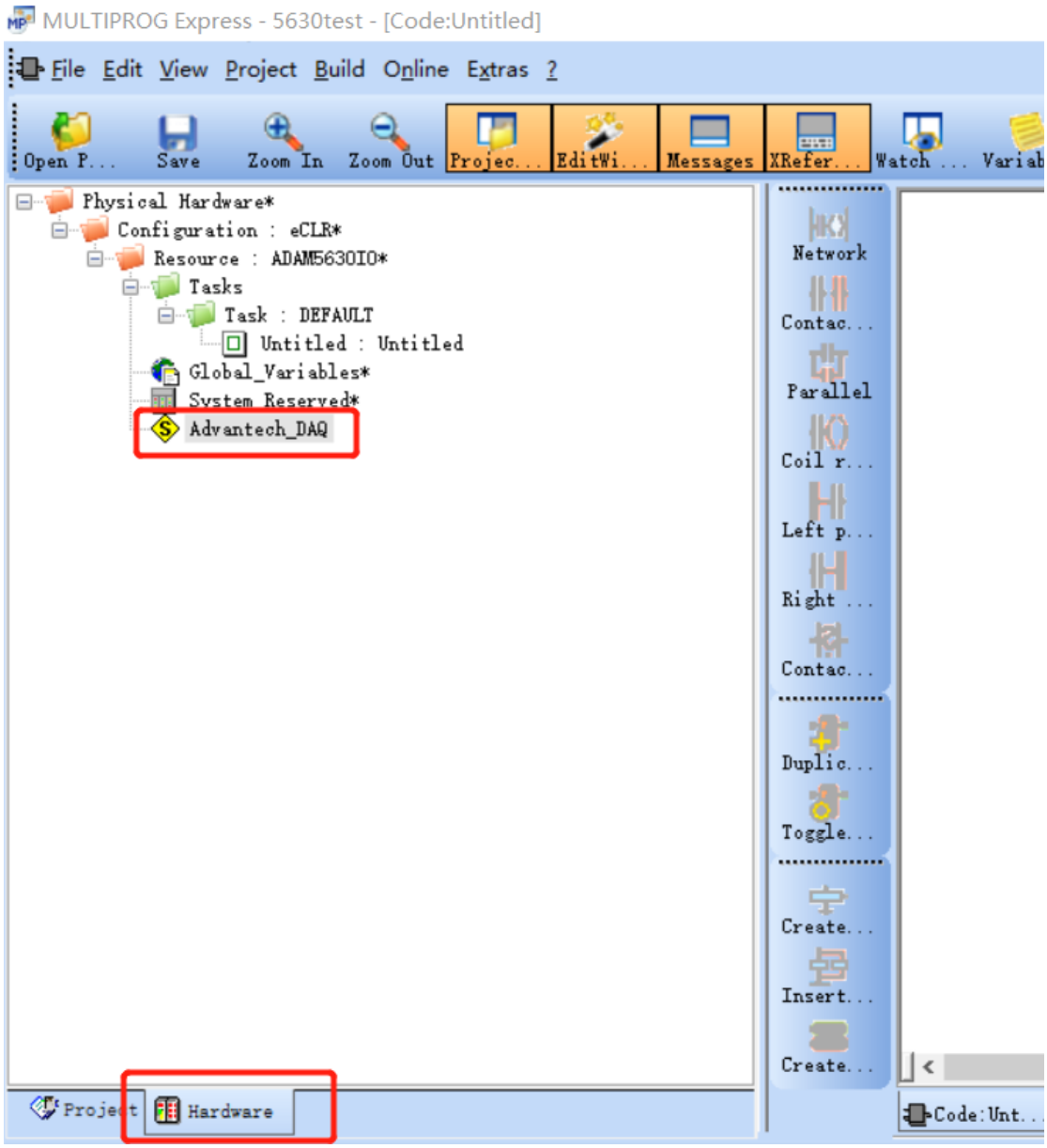



Project Wizard (Step 2 of 6)
Project
Libraries
Data Types
Logical POUs
MyProgram
Physical Hardware
MyConfiguration
MyResource
Tasks
Globals
IO_Config
Please choose the Name and Language of the initial program Program Organisation Unit (POU).
Name of POU:
Untitled
Language
Instruction List (IL)
Structured Text (ST)
Sequence Flow Chart (SFC)
Function Block Diagram (FBD)
Ladder (LD)
Fixed Format Ladder (FFLD)
< 上一步(B)
下一步(N) >
取消
帮助

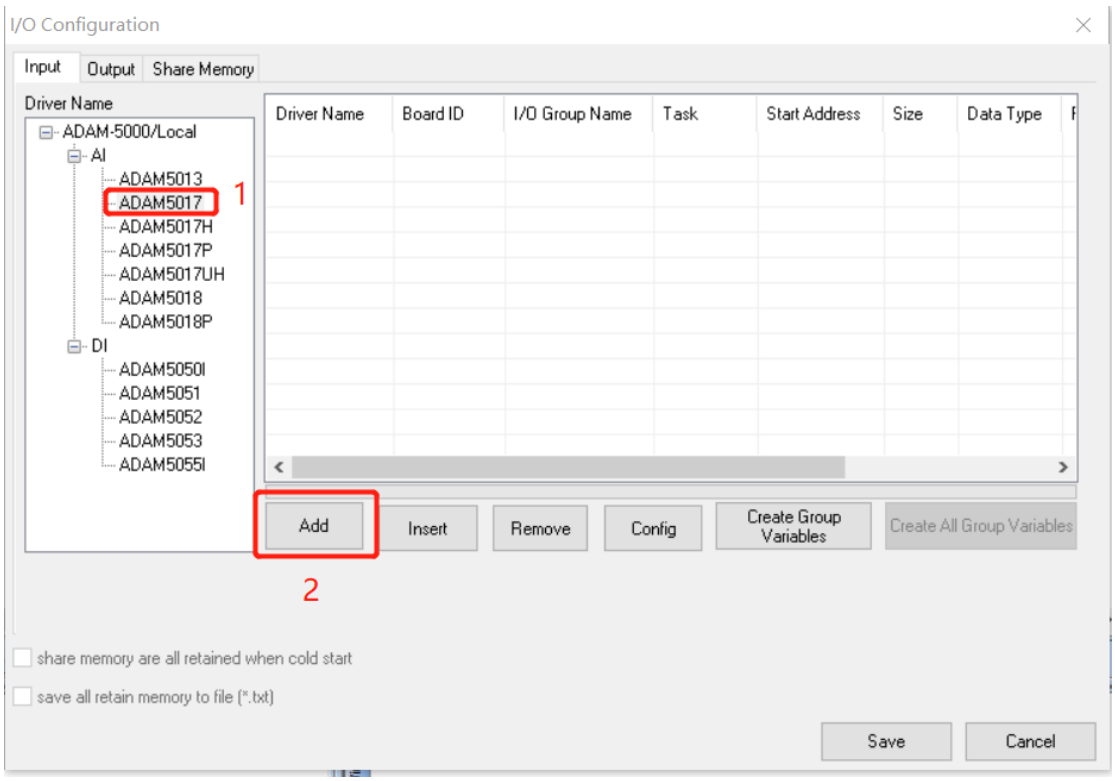

Project Wizard (Step 4 of 6)
Project
Libraries
Data Types
Logical POUs
MyProgram
Physical Hardware
MyConfiguration
MyResource
Tasks
Globals
IO_Config
Please choose the Resource Name and the Resource Type.
The Resource describes the characteristics of the processsor type of the PLC.
Resource
Name:
Resource
Type:
ADAM5630IO
< 上一步(B)
下一步(N) >
取消
帮助

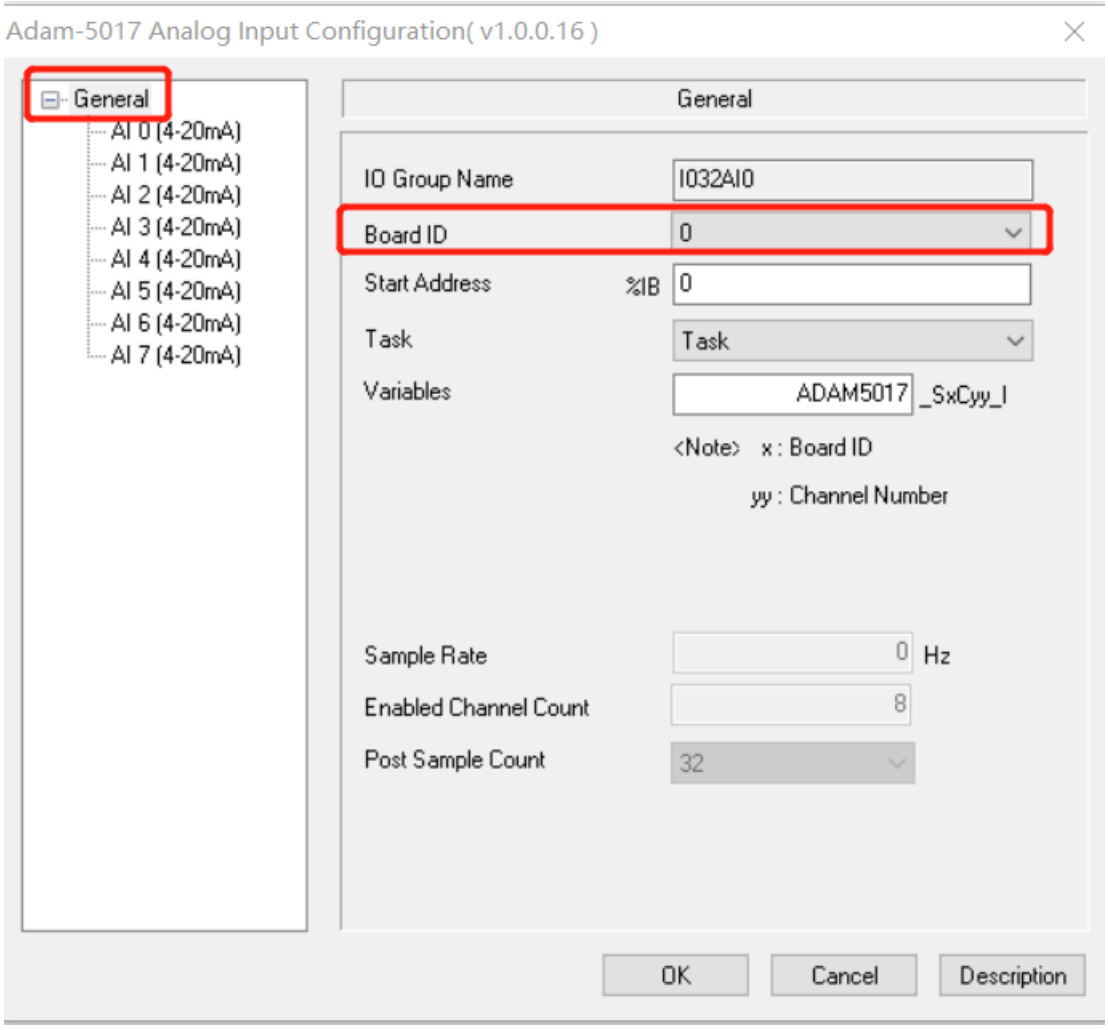

Project Wizard (Step 5 of 6)
Project
Libraries
Data Types
Logical POUs
MyProgram
Physical Hardware
MyConfiguration
MyResource
Tasks
Globals
IO_Config
Please choose the task name and type in which your predefined POU is running.
Task
Name:
Task
Type:
DEFAULT
< 上一步(B)
下一步(N) >
取消
帮助

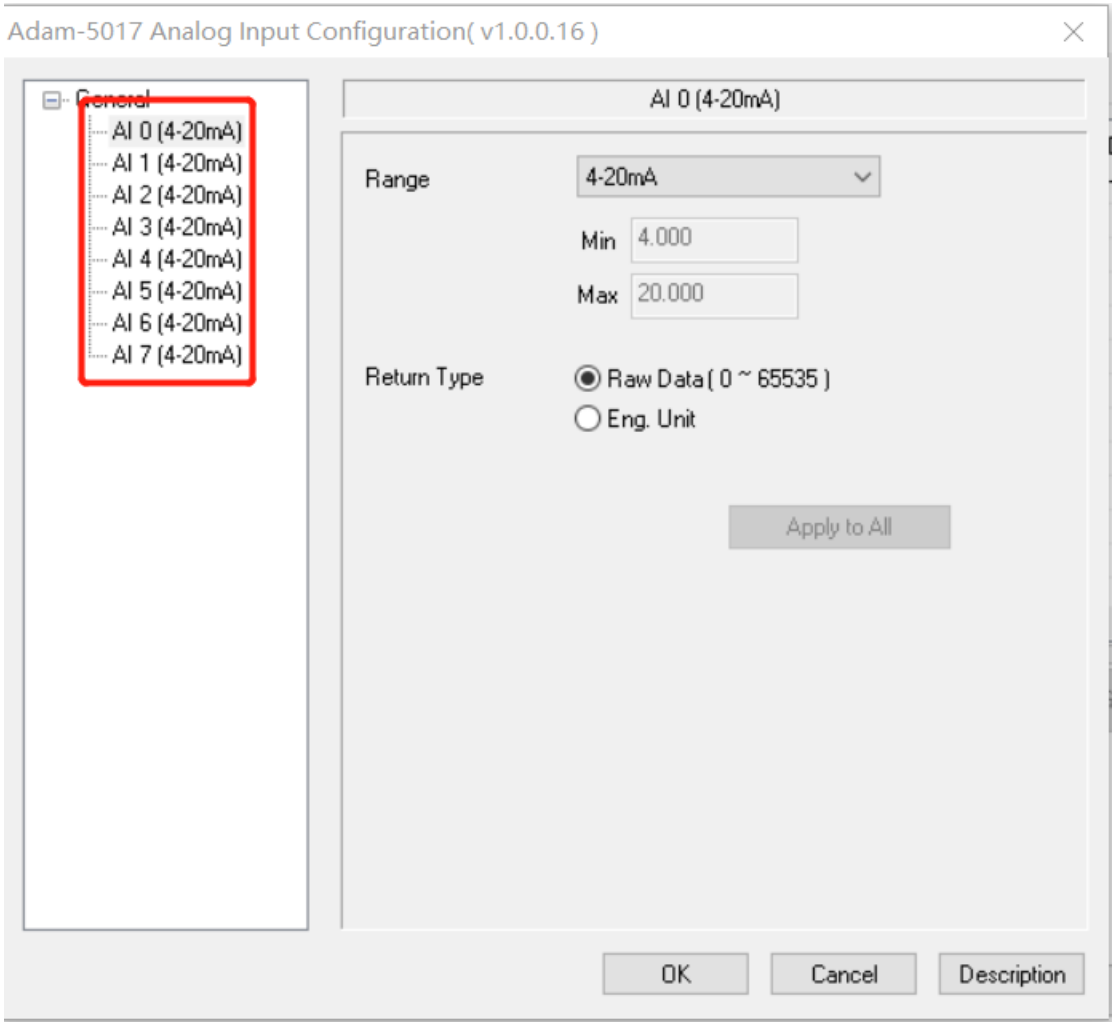
2) Create ADAM-5630 tags in kw

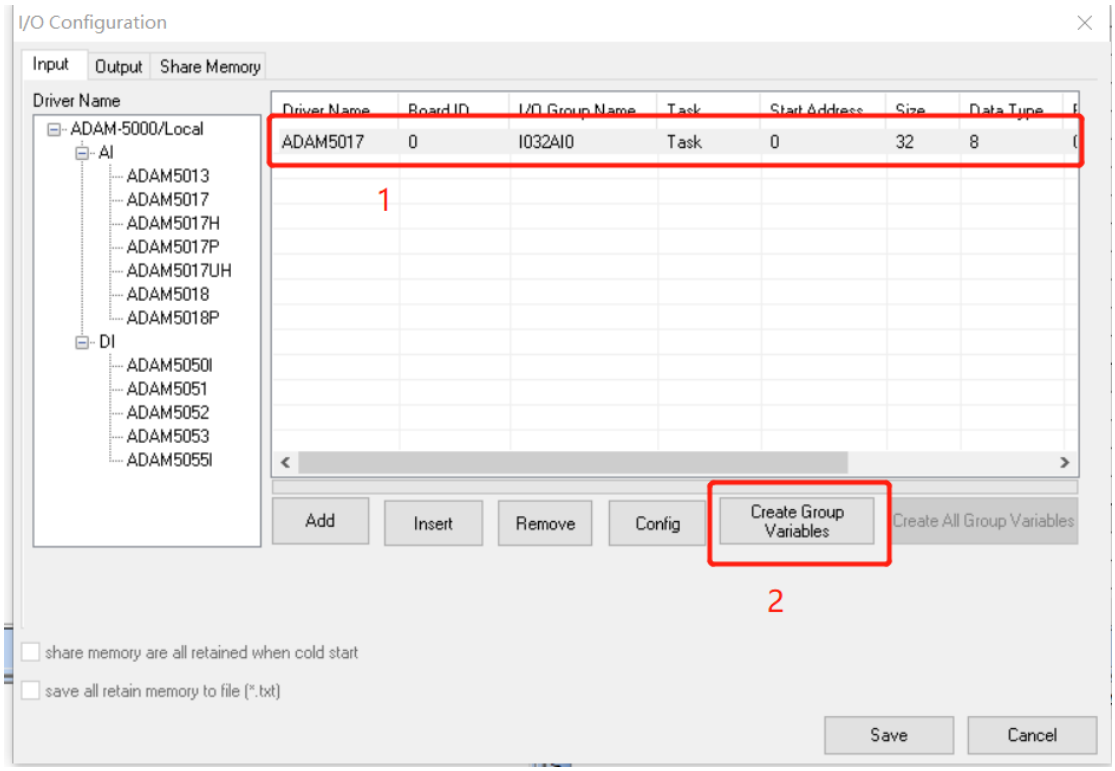


(1) IO tags


I/O Configuration
Input
Output
Share Memory
Driver Name
ADAM-5000/Local
AI
ADAM5013
ADAM5017
ADAM5017H
ADAM5017P
ADAM5017UH
ADAM5018
ADAM5018P
DI
ADAM5050I
ADAM5051
ADAM5052
ADAM5053
ADAM5055I
1
Driver Name
Board ID
I/O Group Name
Task
Start Address
Size
Data Type
Add
Insert
Remove
Config
Create Group Variables
Create All Group Variables
2
share memory are all retained when cold start
save all retain memory to file (*.txt)
Save
Cancel


Adam-5017 Analog Input Configuration(v1.0.0.16)
General
AI 0 (4-20mA)
AI 1 (4-20mA)
AI 2 (4-20mA)
AI 3 (4-20mA)
AI 4 (4-20mA)
AI 5 (4-20mA)
AI 6 (4-20mA)
AI 7 (4-20mA)
General
IO Group Name
I032AI0
Board ID
0
Start Address
%IB
0
Task
Task
Variables
ADAM5017
_SxCyy_I
<Note> x : Board ID
yy : Channel Number
Sample Rate
0
Hz
Enabled Channel Count
8
Post Sample Count
32
OK
Cancel
Description


Adam-5017 Analog Input Configuration(v1.0.0.16)
General
AI 0 (4-20mA)
AI 1 (4-20mA)
AI 2 (4-20mA)
AI 3 (4-20mA)
AI 4 (4-20mA)
AI 5 (4-20mA)
AI 6 (4-20mA)
AI 7 (4-20mA)
AI 0 (4-20mA)
Range
4-20mA
Min
4.000
Max
20.000
Return Type
Raw Data (0 ~ 65535)
Eng. Unit
Apply to All
OK
Cancel
Description

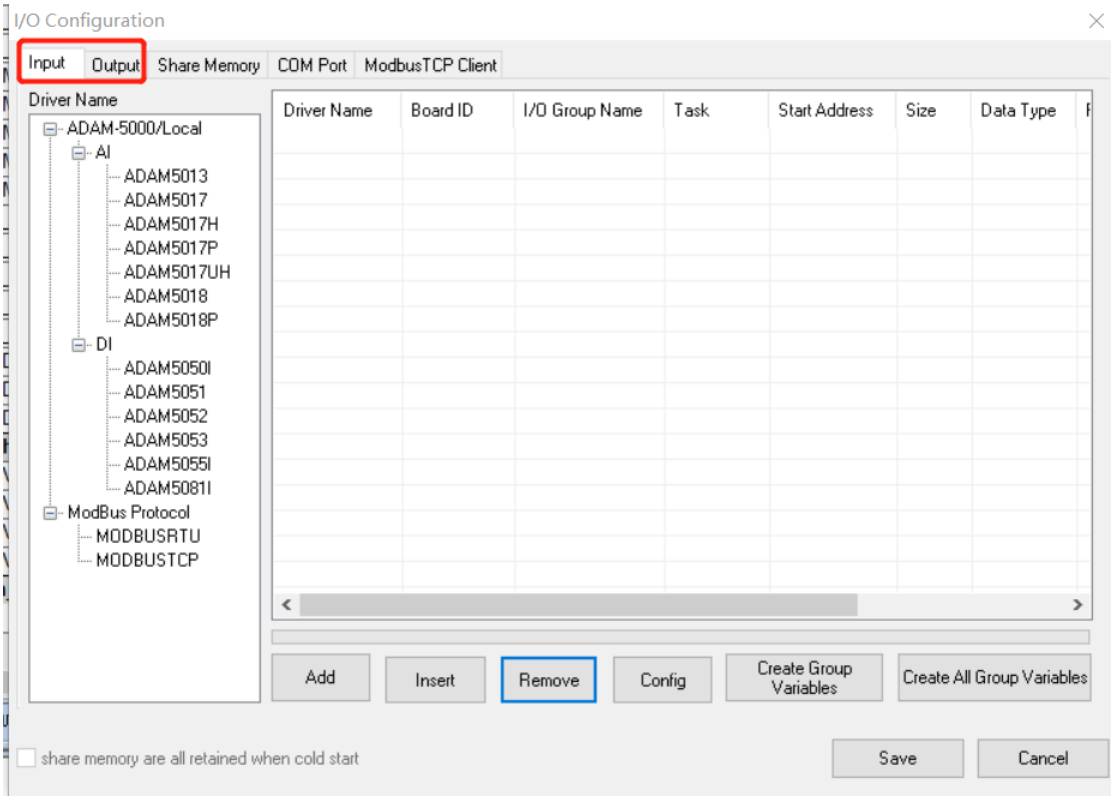

I/O Configuration
Input
Output
Share Memory
Driver Name
ADAM-5000/Local
AI
ADAM5013
ADAM5017
ADAM5017H
ADAM5017P
ADAM5017UH
ADAM5018
ADAM5018P
DI
ADAM5050I
ADAM5051
ADAM5052
ADAM5053
ADAM5055I
Driver Name
Board ID
I/O Group Name
Task
Start Address
Size
Data Type
ADAM5017
0
I032AI0
Task
0
32
8
1
Add
Insert
Remove
Config
Create Group Variables
Create All Group Variables
2
share memory are all retained when cold start
save all retain memory to file (*.txt)
Save
Cancel

(2) Collecting Modbus Tags

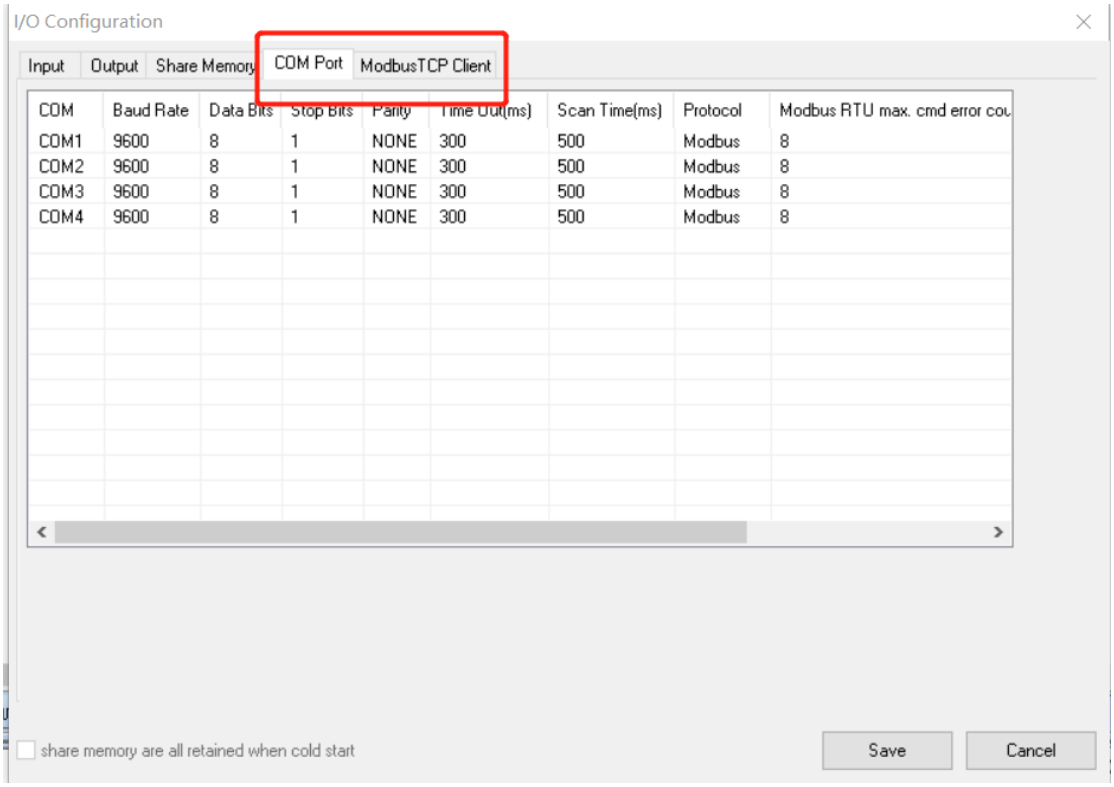
ADAM-5630 can be Modbus client to collect data from Modbus server.

Read tags in “Input”,write tags in “output”.

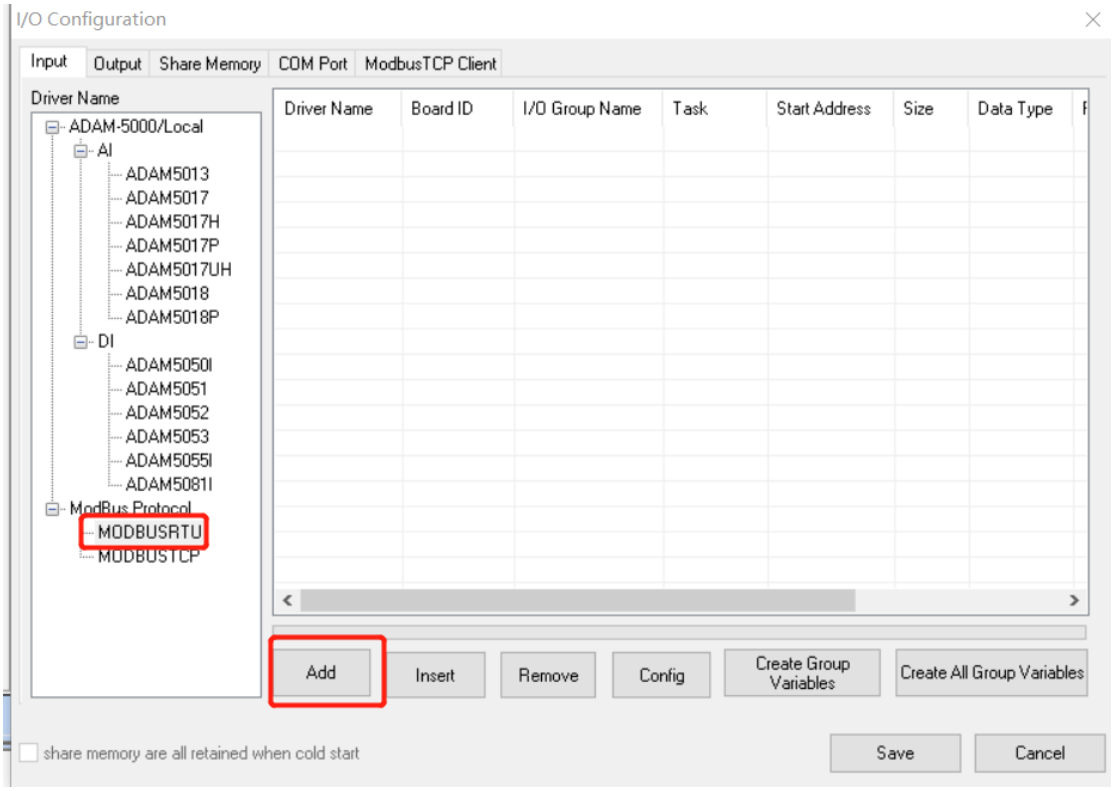
For example, 40001 you need to read and write, so you need to add it both in “input” and “output”.

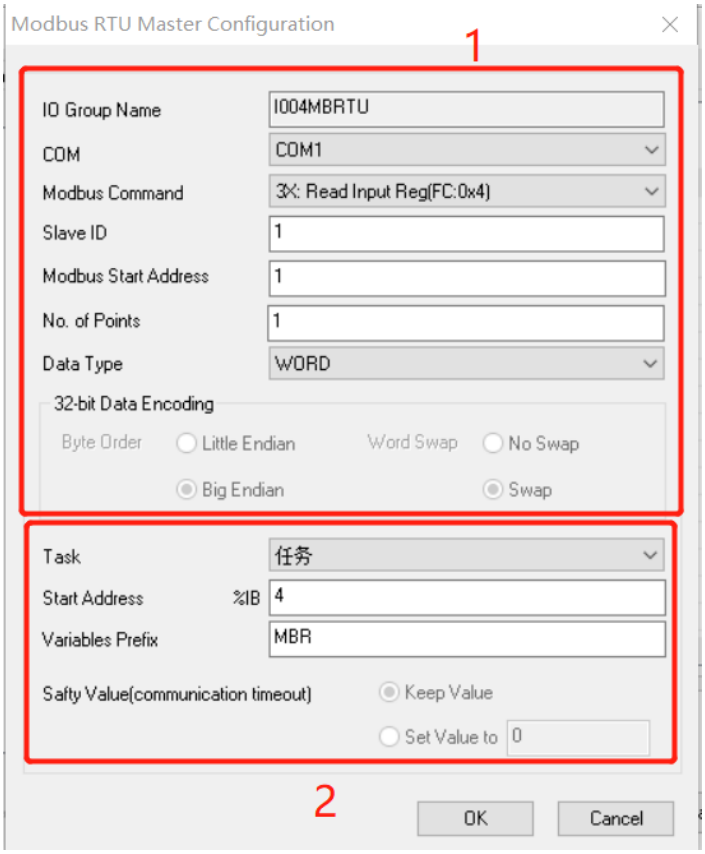


You can configure the COM and TCP paramers in the page below.


I/O Configuration
Input
Output
Share Memory
COM Port
ModbusTCP Client
COM
Baud Rate
Data Bits
Stop Bits
Parity
Time Out(ms)
Scan Time(ms)
Protocol
Modbus RTU max. cmd error cou
COM1 9600 8 1 NONE 300 500 Modbus 8
COM2 9600 8 1 NONE 300 500 Modbus 8
COM3 9600 8 1 NONE 300 500 Modbus 8
COM4 9600 8 1 NONE 300 500 Modbus 8
share memory are all retained when cold start
Save
Cancel

1. Modbus RTU


I/O Configuration
Input
Output
Share Memory
COM Port
ModbusTCP Client
Driver Name
ADAM-5000/Local
AI
ADAM5013
ADAM5017
ADAM5017H
ADAM5017P
ADAM5017UH
ADAM5018
ADAM5018P
DI
ADAM5050I
ADAM5051
ADAM5052
ADAM5053
ADAM5055I
ADAM5081I
ModBus Protocol
MODBUSRTU
MODBUSTCP
Driver Name
Board ID
I/O Group Name
Task
Start Address
Size
Data Type
Add
Insert
Remove
Config
Create Group Variables
Create All Group Variables
share memory are all retained when cold start
Save
Cancel



1 is the configuration for tags. 2 is the configuration for KW, stay default is OK.

COM: The serial to be used.

Modbus Command: The Modbus function command these tags use.

Slave ID: Device ID of the Modbus server.

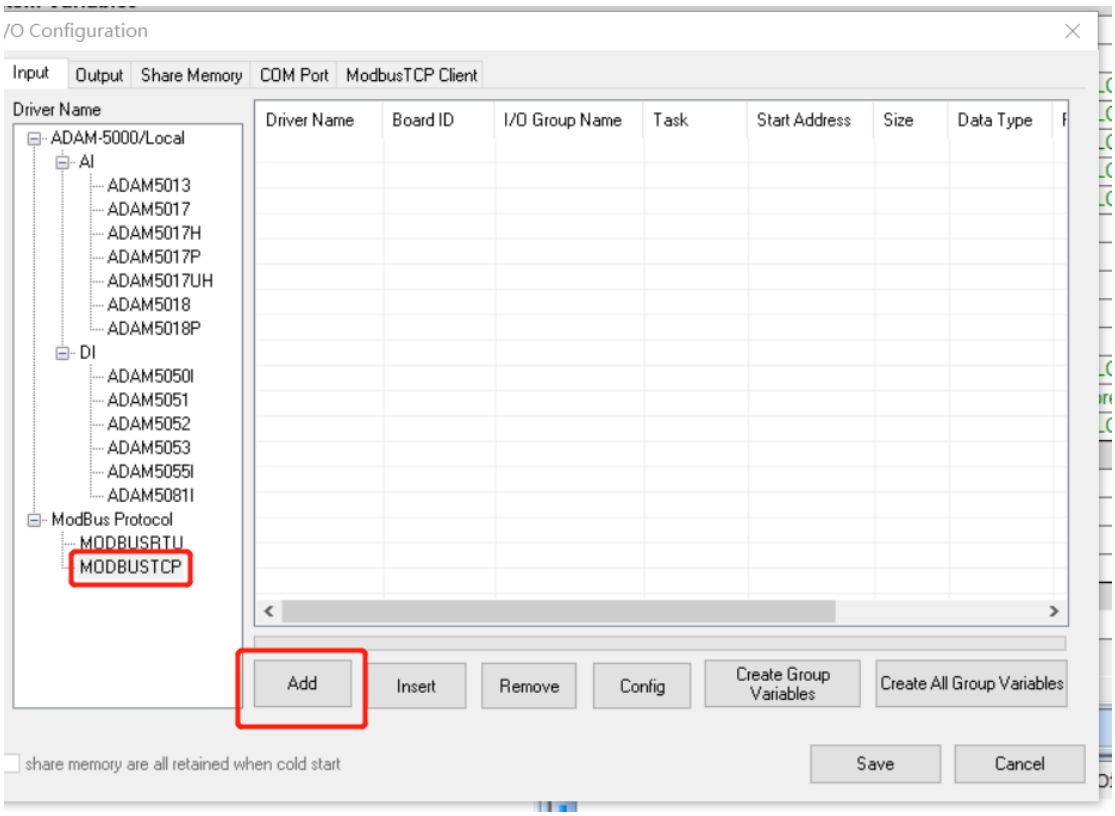
Modbus Start Address: The number these Modbus addresses start with.

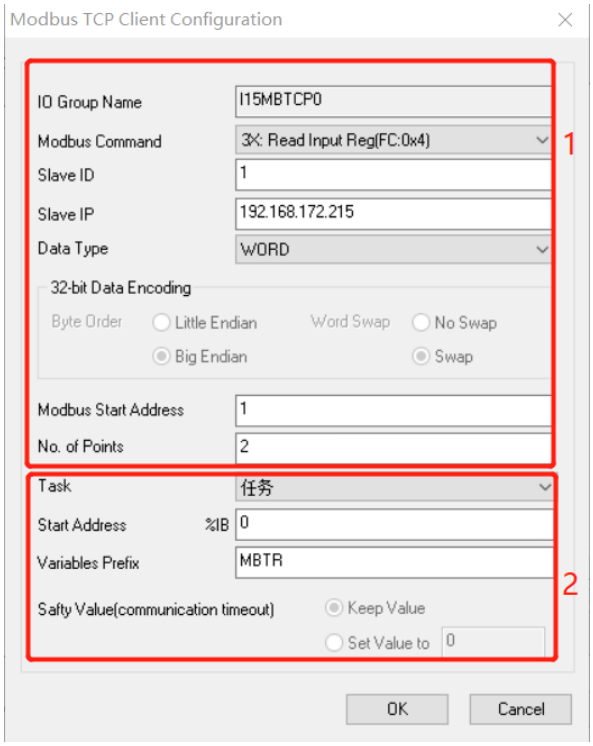
No. of points: The tag counts you want to create.

Data Type: Data Type of the tags.

In above picture's example, will use COM1 to collect Modbus address 30001 (Function Command 4, start address 1, 1 tag)

2. Modbus TCP





1 is the configuration for tags. 2 is the configuration for KW, stay default is OK.

Modbus Command: The Modbus function command these tags use.

Slave ID: Device ID of the Modbus server.

Slave IP: Slave IP of the Modbus server.

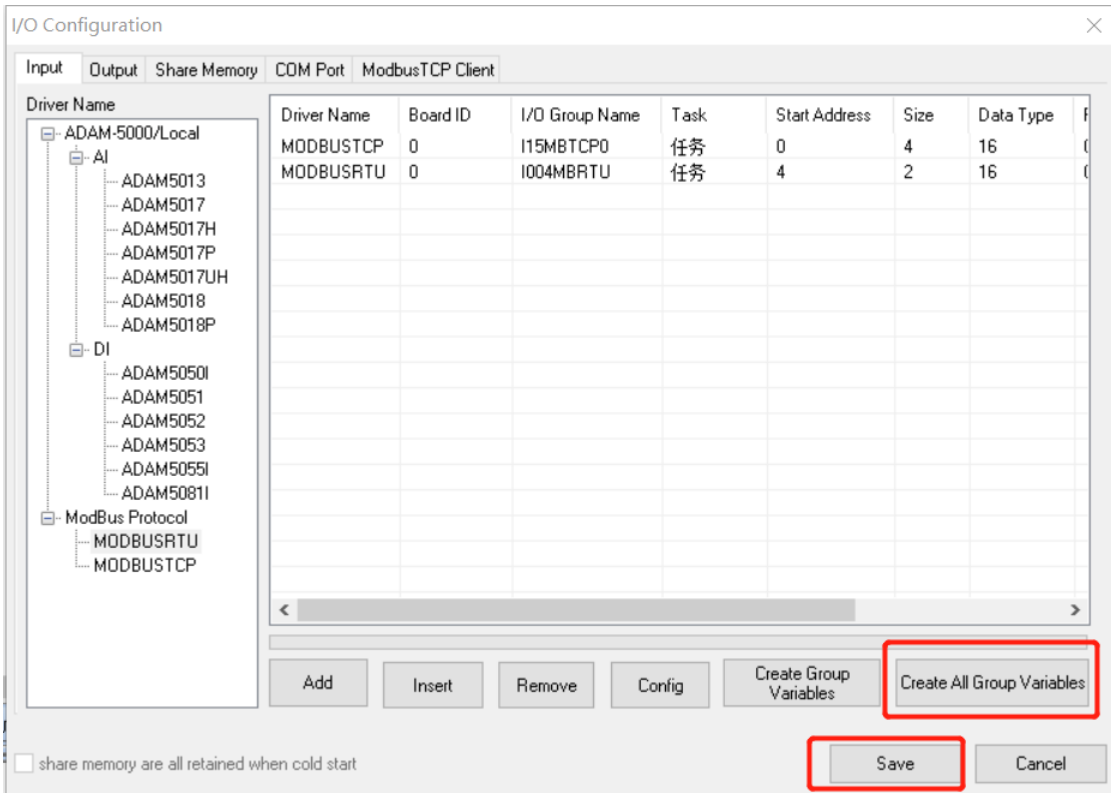
Data Type: Data Type of the tags.

Modbus Start Address: The number these Modbus addresses start with.

No. of points: The tag counts you want to create.

In above picture's example, the Modbus tags are 30001 and 30002 (Function Command 4, start address 1, 2 tags).

3. Create tags in KW

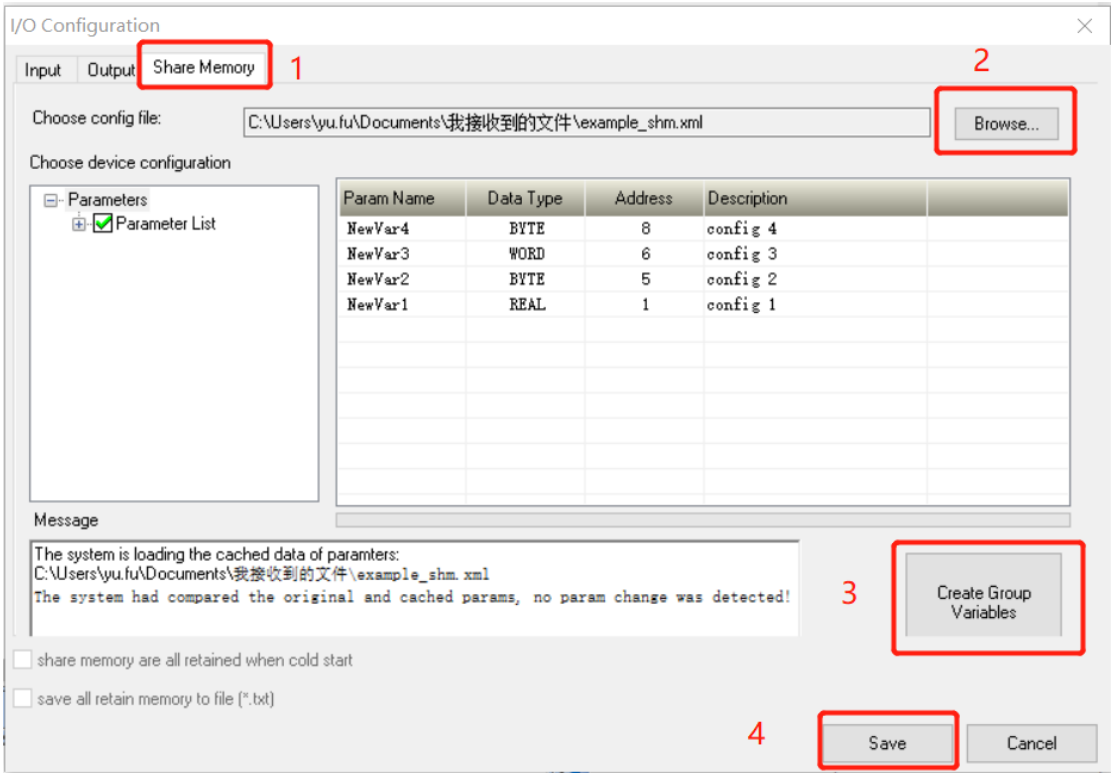


(3) Share Memory tags

Share Memory is used to mapping kw tags to ADAM-5630's Modbus Server.
Example_shm.xml is an example file to create share memory tags. You can modify the file to create share memory tags.
And you can also directly create share memory tags in kw.

example_shm.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Parameters>
<KWData name="NewVar1" address="1" dataType="REAL" description="config 1"/>
<KWData name="NewVar2" address="5" dataType ="BYTE" description="config 2"/>
<KWData name="NewVar3" address="6" dataType="WORD" description="config 3"/>
<KWData name="NewVar4" address="8" dataType="BYTE" description="config 4"/>
</Parameters>
```



ADAM-5630 Controller reserves 17K Bytes memory space for Modbus function. The memory block can store user's data and exchange the data through Modbus protocol.

The unit in Modbus 4X registers is Word so there are totally 8K Words available. The Modbus address is defined from 42001 to 49999. In order to exchange the data through Modbus, users need to move the data onto this memory block by setting the memory address in "I/O Address" field manually.

The memory address of the memory block is defined from %MB3.10000 to %MB3.26999. Address %MB3.10000 to %MB3.25999 is for Modbus holding registers, %MB3.26000 to %MB3.26999 is for Modbus coil registers.

Total memory size	17 KBytes
Address for 4X	%MB3.10000 ~ %MB3.25999
Address for 0X	%MB3.26000 ~ %MB3.26999

The mapping table of I/O Address and Modbus Address is as following table.

Holding registers (4X):

IO Address	Modbus Address
%MW3.10000	42001
%MW3.10002	42002
%MW3.10004	42003
……	
%MW3.25996	49999

Coil registers (0X):

For KW eclr address %MX3.x.1 ~ %MX3.x.7 is forbidden, the mapping data for coil registers can only use as below:

E.g. Define coildata_1 as BYTE type, the address of coildata_1 is %MB3.26000, define coildata_n as BYTE type, the address of coildata_n is %MB3.26999.

Name of variable	IO Address	Modbus Address
coildata_1.X0	——	02001
coildata_1.X1	——	02002
coildata_1.X2	——	02003
……		
coildata_n.X0	——	09993
coildata_n.X1	——	09994
……		
coildata_n.X6		09999

Since unit of MB3.10000 is Byte, users need to map the I/O Address and Modbus Address as following.

Bool data type

Note:

For KW eCLR %MW3.x.1 ~ %MW3.x.7 is forbidden, bool data type can only use as below:

e.g. Define coildata_1 as BYTE type, the address of coildata_1 is %MB3.26000, define coildata_n as BYTE type, the address of coildata_n is %MB3.26999

	Name of variable	Modbus Address	Length
Data1	coildata_1.X0	02001	1 Bit
Data2	coildata_1.X1	02002	1 Bit
Data3	coildata_1.X2	02003	1 Bit

BYTE and WORD data type

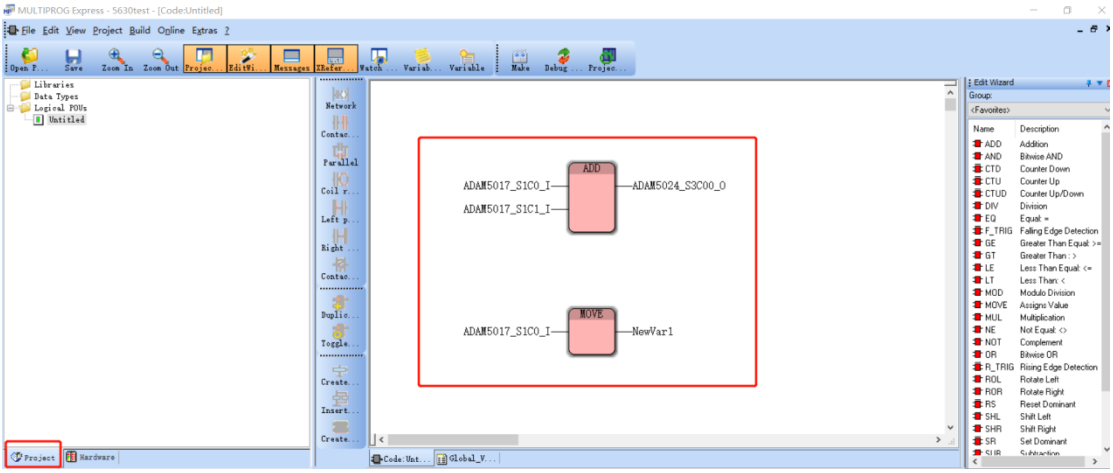
	IO Address	Modbus Address	Length
Data1	MW3.10000 + MW3.10001	42001	2 Bytes
Data2	MW3.10002 + MW3.10003	42002	2 Bytes
Data3	MW3.10004 + MW3.10005	42003	2 Bytes

DWORD and REAL data type

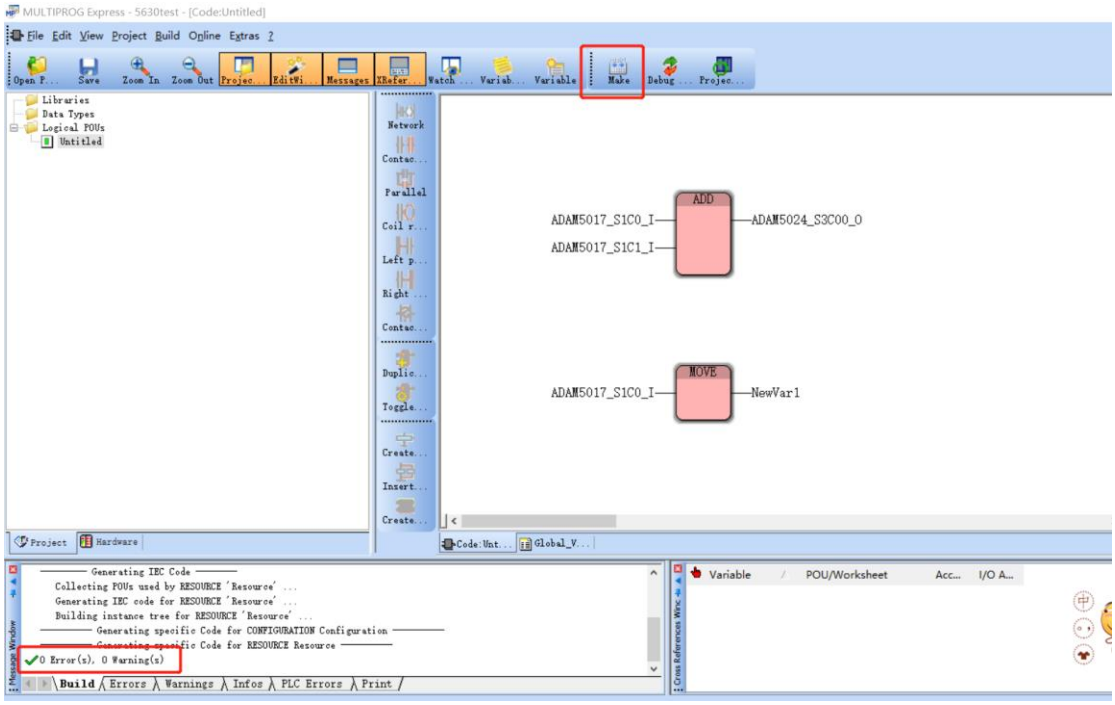
	IO Address	Modbus Address	Length
Data1	MW3.10000 + MW3.10001	42001+42002	4 Bytes

	+ MW3.10002 + MW3.10003		
Data2	MW3.10004 + MW3.10005 + MW3.10006 + MW3.10007	42003+42004	4 Bytes
Data3	MW3.10008 + MW3.10009 + MW3.100010 + MW3.10011	42005+42006	4 Bytes

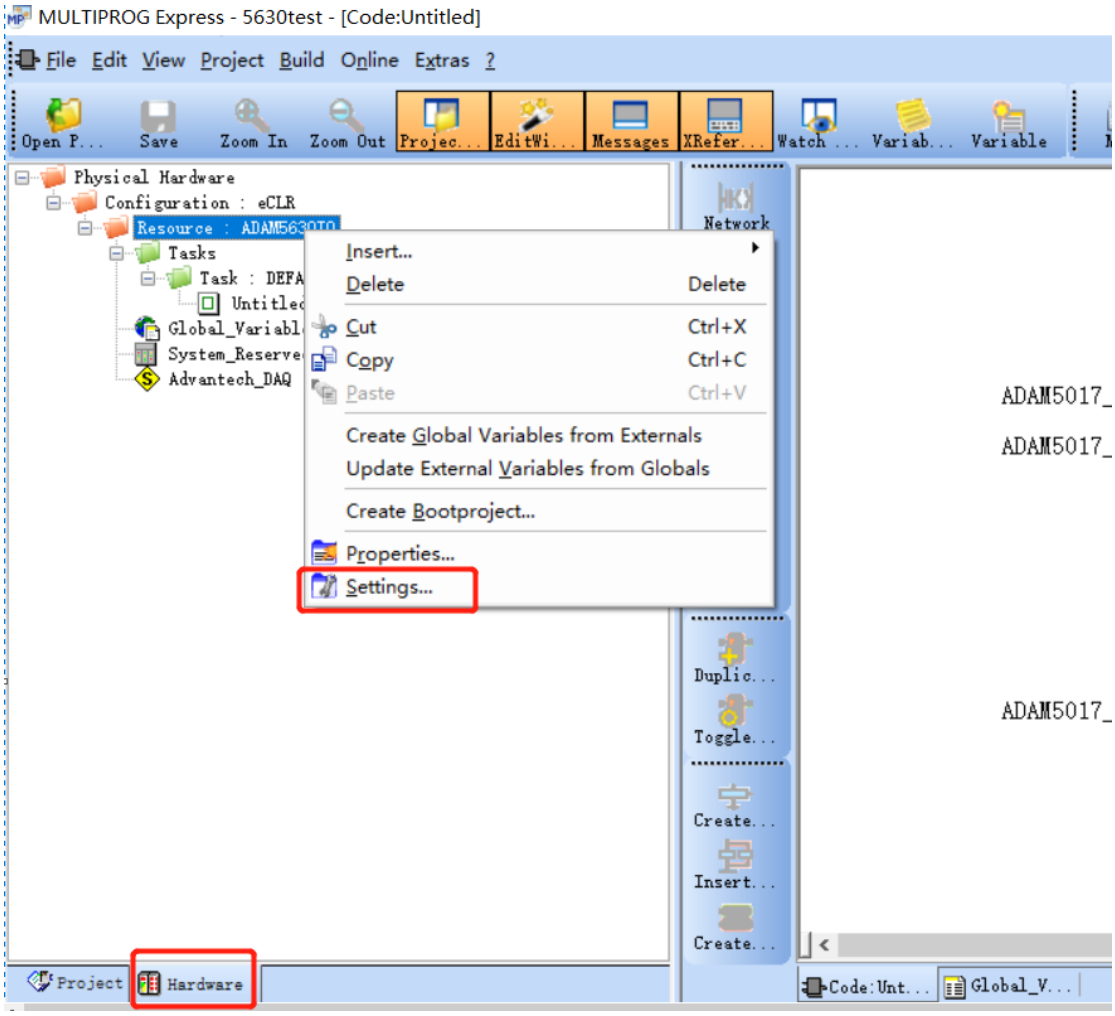
3) Write logic for tags.



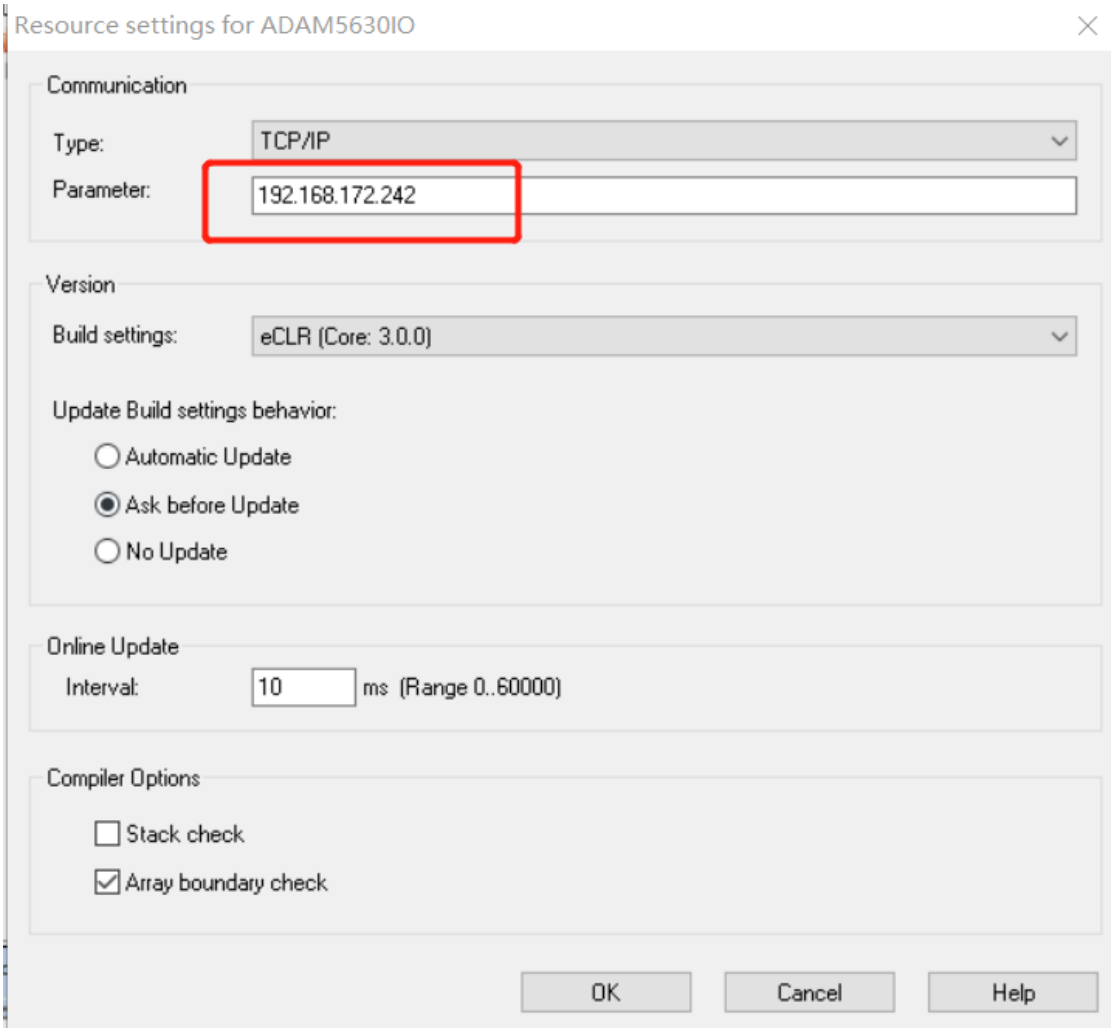
4) Make the logic.

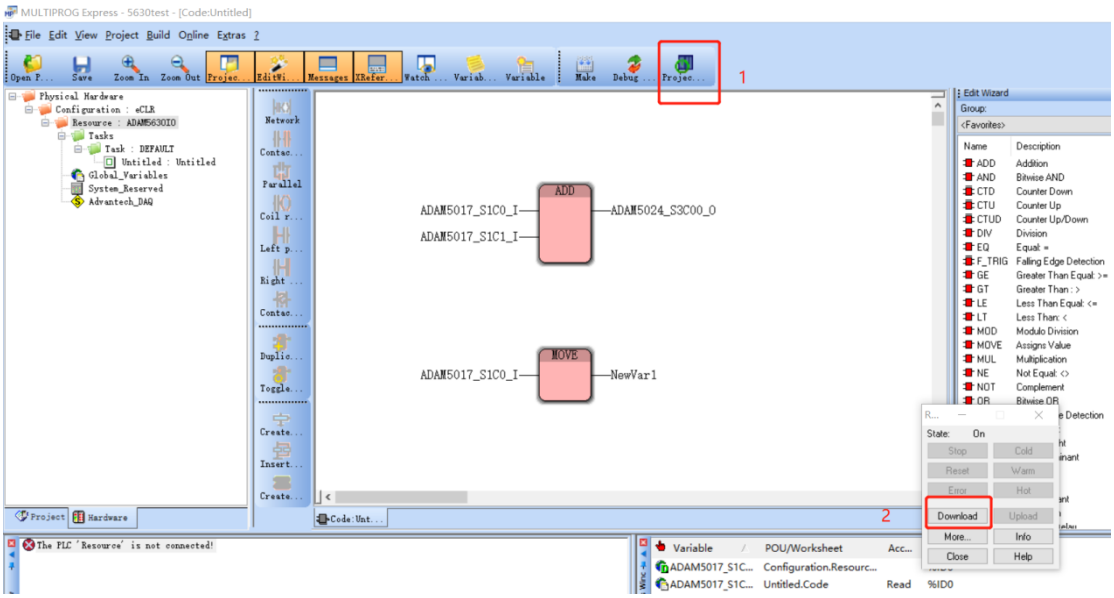


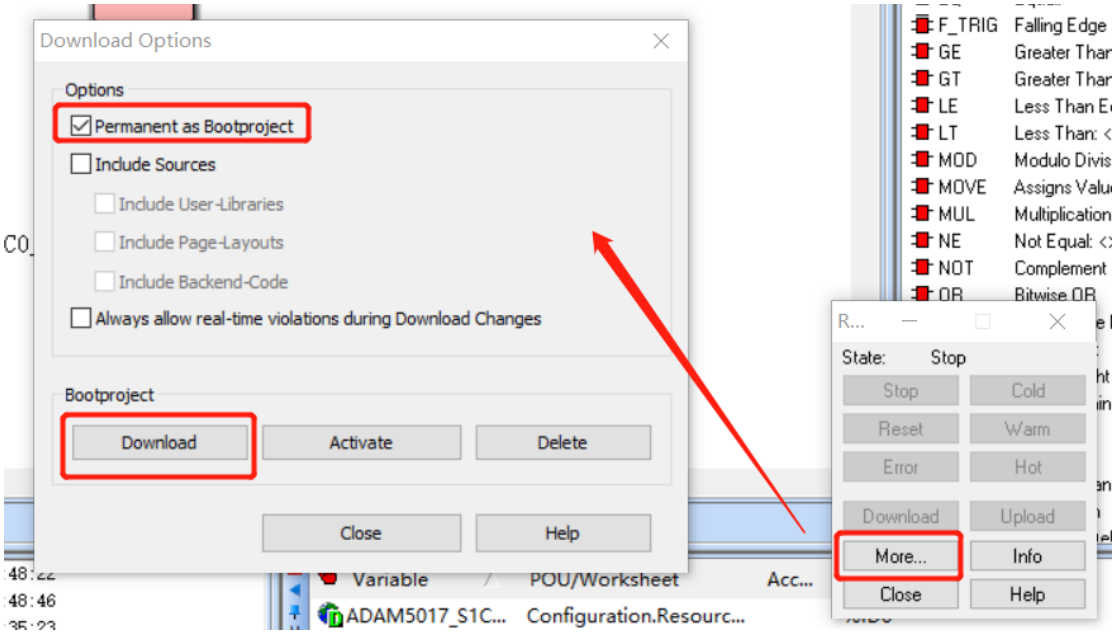
5) Download the kw logic to ADAM-5630.

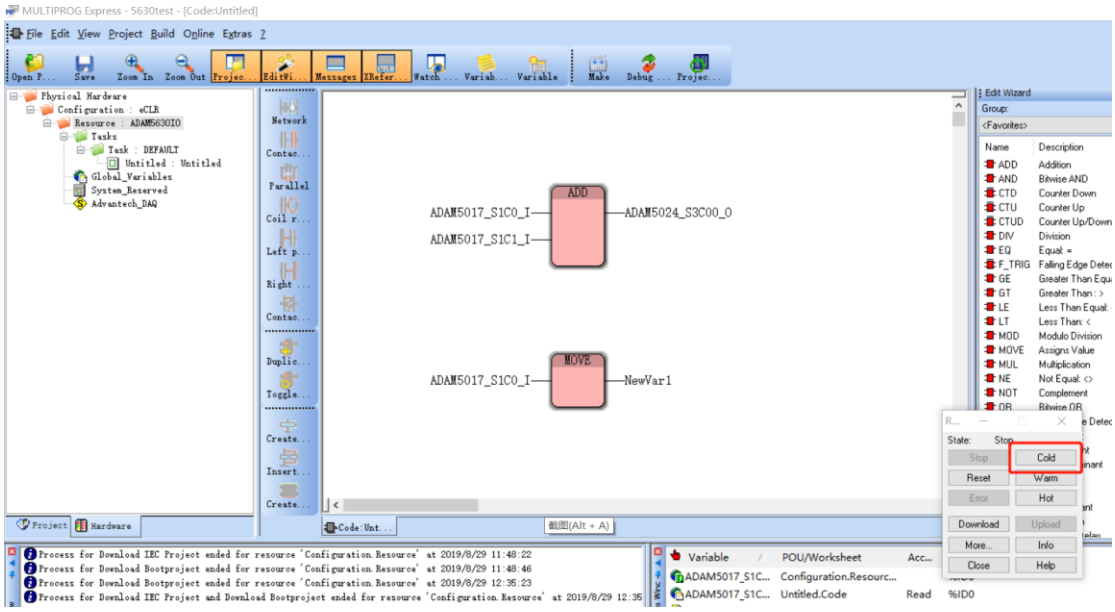


Fill in the IP of ADAM-5630.


Resource settings for ADAM5630IO
Communication
Type:
TCP/IP
Parameter:
192.168.172.242
Version
Build settings:
eCLR (Core: 3.0.0)
Update Build settings behavior:
Automatic Update
Ask before Update
No Update
Online Update
Interval:
10
ms (Range 0..60000)
Compiler Options
Stack check
Array boundary check
OK
Cancel
Help


ADAM5017_S1C0_I
ADAM5017_S1C1_I
ADD
ADAM5024_S3C00_0
ADAM5017_S1C0_I
MOVE
NewVar1
1
2
Download
The PLC 'Resource' is not connected!


Download Options
Options
Permanent as Bootproject
Include Sources
Include User-Libraries
Include Page-Layouts
Include Backend-Code
Always allow real-time violations during Download Changes
Bootproject
Download
Activate
Delete
Close
Help
F_TRIG Falling Edge
GE Greater Thar
GT Greater Thar
LE Less Than E
LT Less Than: <
MOD Modulo Divis
MOVE Assigns Valu
MUL Multiplication
NE Not Equal: <>
NOT Complement
R...
State: Stop
Stop
Cold
Reset
Warm
Error
Hot
Download
Upload
More...
Info
Close
Help
Variable
POU/Worksheet
Acc...
ADAM5017_S1C...
Configuration.Resourc...


MULTIPROG Express - 5630test - [Code:Untitled]
File Edit View Project Build Online Extras ?
Physical Hardware
Configuration : eCLR
Resource : ADAM5630IO
Tasks
Task : DEFAULT
Untitled : Untitled
Global_Variables
System_Reserved
Advantech_DAQ
ADD
ADAM5017_S1C0_I
ADAM5017_S1C1_I
ADAM5024_S3C00_0
MOVE
ADAM5017_S1C0_I
NewVar1
Edit Wizard
Process for Download IEC Project ended for resource 'Configuration.Resource' at 2019/8/29 11:48:22
Process for Download Bootproject ended for resource 'Configuration.Resource' at 2019/8/29 11:48:46
Process for Download Bootproject ended for resource 'Configuration.Resource' at 2019/8/29 12:35:23
Process for Download IEC Project and Download Bootproject ended for resource 'Configuration.Resource' at 2019/8/29 12:35
Variable
POU/Worksheet
ADAM5017_S1C...
Configuration.Resourc...
Untitled.Code
Read
%ID0