AX110xx Development Kit
User Guide

Revision 1.70
Aug. 01st, 2011

Revision History

Revision	Date	Description
1.0	2006/7/13	New release.
1.1	2006/7/21	1. Add Section 5 “I2C EEPROM Programming”, Appendix B. “80-pin Development Board I2C EEPROM Default Setting” and Appendix C. “128-pin Development Board I2C EEPROM Default Setting”. 2. Modify Figure 5 “AX11015 128-pin Development Board DIP Switches” to correct the labels of P0_0~P0_7, P1_0~P1_7, P2_0~P2_7 and Ethernet LEDs on AX11015 128-pin development board. 3. Update Appendix D “80-pin Development Board schematic” to AX1100x 80-pin Development Board schematics v1.01. 4. Update Appendix E “128-pin Development Board schematic” to AX11015 128-pin Development Board schematics v1.02.
1.2	2007/4/2	1. Add Section 7-1 “COM Port Flash Programming Method Under Windows”. 2. Add Section 8-3 “Upper Protocol Modules”. 3. Add Section 10 “AX110xx Mass Production Solutions”. 4. Add a note for RS-232 Null modem converter and AX11025 128-pin Development Kit information in Section 1. 5. Add Section 4 “How to compile AX110xx development board demo firmware”. 6. Update the pictures of the demo firmware web pages in Section 3. 7. Add the CAN interface configuration information in Section 5-2-7 and Appendix C for AX11025 128-pin development kit. 8. Modify Section 8-2 to add information about AX110xx uIP TCP/IP module. 9. Update the Software Availability table in Appendix A. 10. Change the default value of 80-pin EEPROM offset 0x12 to 0x87 in Appendix B. 11. Update 80-pin/128-pin development board reference schematics in Appendix D and E.
1.3	2007/12/31	1. Add Section 3-4 to introduce AX11025 demo firmware functions. 2. Add Figure 5 “AX110xx Demo Firmware Run-time Code Starting Address” in Section 4. 3. Add some more information in Section 2, 3-1-5, 3-2-1, 3-3-1, 6 and 7-3. 4. Add the “UARTH.EXE” utility information in Appendix A. 5. Update AX11025 Development Kit information in Section 1. 6. Update some information in Section 5-2. 7. Correct some typos in Section 3-1-4, 3-1-5. 8. Update the latest 128-pin development board reference schematic in Appendix E.

1.4	2008/8/25	1. Modified Section 2 to add Quick Start information. 2. Updated the revised Device Finder utility information in Section 3-1. 3. Added Section 3-1-3 to indicate the EEPROM setup function information of the Device Finder utility. 4. Modified the note messages in Section 5-2-9, 5-2-10, 5-2-11, 6-5. 5. Updated the revised Windows ISP tool information in Section 7-1. 6. Added Figure 17 "AX110xx Peripheral Software Modules" in Section 8-1. 7. Modified some descriptions in Section 8-2. 8. Added Appendix D to indicate the AX11025 128-pin Development Board I2C EEPROM Default Setting. 9. Updated Appendix E "80-pin Development Board schematic" to AX11001&AX11005 80-pin Development Board schematic v1.10. 10. Updated Appendix F "128-pin Development Board schematic" to AX11015&AX11025 128-pin De25lopment Board schematic v1.30.
1.5	2009/12/01	1. Remove Appendix E "AX11001&AX11005 80-pin Development Board schematic". 2. Remove Appendix F "AX11015&AX11025 128-pin Development Board schematic". 3. Correct some typos in Section 5-1-5, 5-2-13. 4. Added Section 8-1 AX110xx Ethernet Boot Loader. 5. Added Section 9-3, 9-4, 9-5 DoCD HAD2 Debugger description. 6. Modified some descriptions in Appendix A.
1.6	2010/04/13	1. Updated the revised "How to use DoCD HAD2 Debugger with Keil IDE" in Section 9-5. 2. Updated Figure 8 and 18 "TFTPD32 TFTP/DHCP Server Setting".
1.70	2011/08/01	1. Updated the information in Section 2-1 and 4. 2. Updated Figure 1, 2, 10, 11, 23. 3. Added ELNEC's Device Programmer information in Section 10-2. 4. Updated the Section 9-4 description. 5. Changed the revision number to 3-digit format.

Table of Contents

List of Figures

1. Introduction

This document provides the overviews of AX110xx Development Kit.

ASIX Electronics provides three kinds of AX110xx Development Kits for customers' reference. AX1100x 80-pin Development Kit is for users to evaluate AX11001/AX11005 products, AX11015 128-pin Development Kit is for users to evaluate AX11015 product and AX11025 128-pin Development Kit is for users to evaluate AX11025 product. If you need to purchase the AX110xx development boards or reference design boards, please contact ASIX's Sales (sales@asix.com.tw) for details.

AX1100x 80-pin development kit consists of five components:
- AX1100x 80-pin development board with a 1-Wire temperature sensor,
- AX110xx Development Kit CD,
- One RS-232 cable with a Null modem converter,
- One RJ-45 Ethernet cable,
- One 5V/3A AC/DC power adapter

AX11015 128-pin development kit consists of five components:
- AX11015 128-pin development board with a 1-Wire temperature sensor,
- AX110xx Development Kit CD,
- One RS-232 cable with a Null modem converter,
- One RJ-45 Ethernet cable,
- One 5V/3A AC/DC power adapter

AX11025 128-pin development kit consists of five components:
- AX11025 128-pin development board,
- AX110xx Development Kit CD,
- One RS-232 cable with a Null modem converter,
- One RJ-45 Ethernet cable,
- One 5V/3A AC/DC power adapter

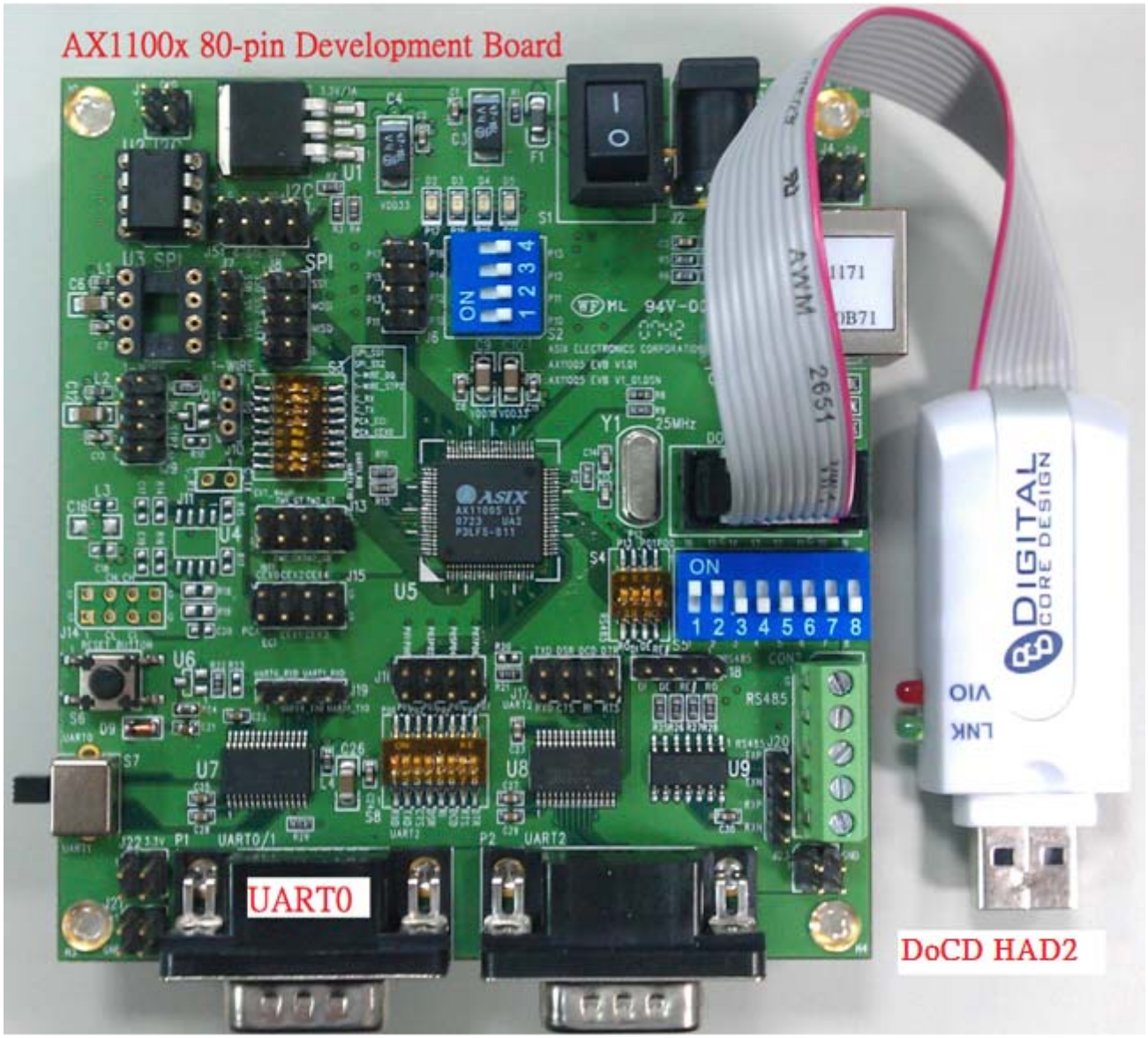


Figure 1. AX1100x 80-pin Development Board

Note: The DoCD HAD2 module is not included in the AX110xx Development Kit.

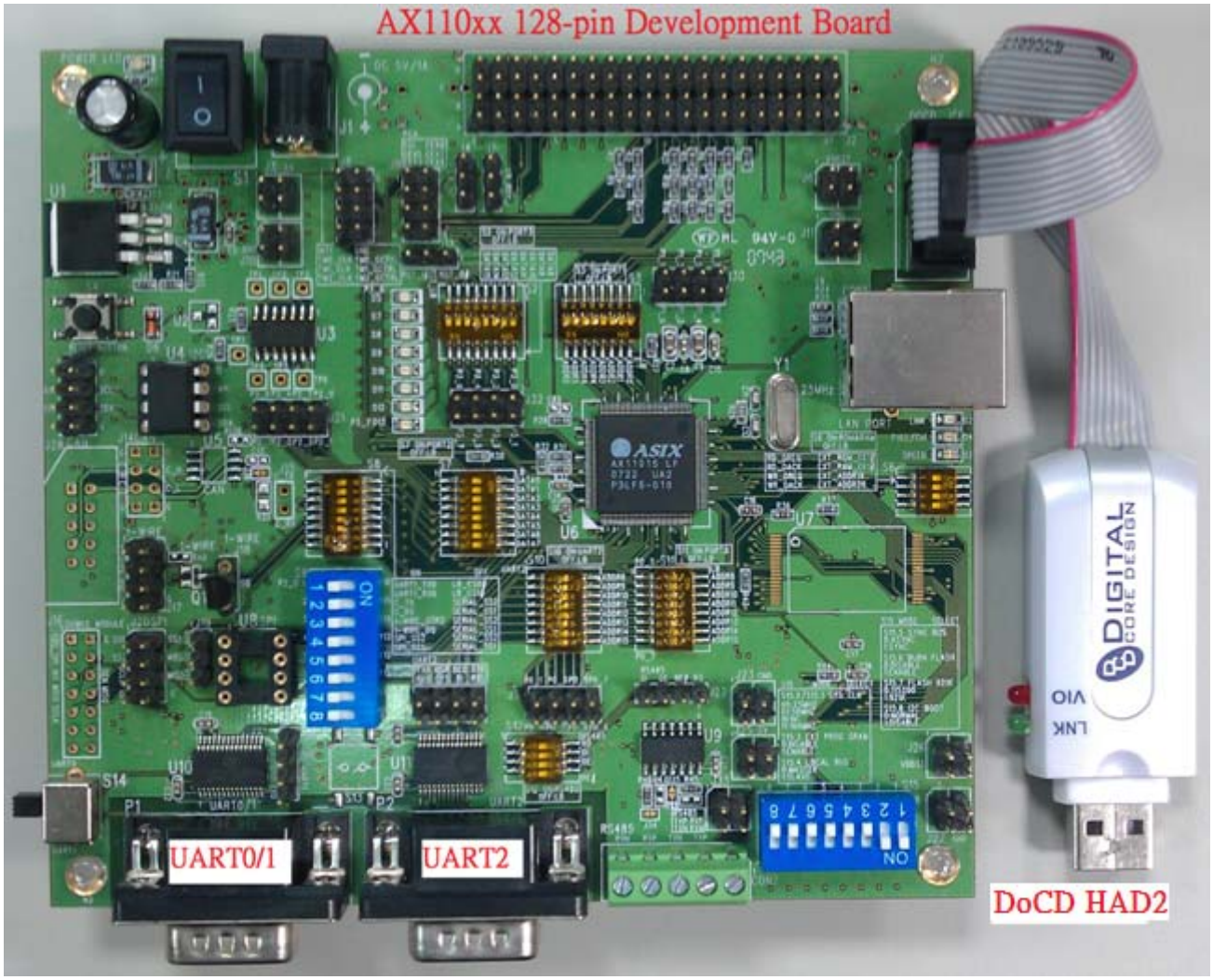


Figure 2. AX11015 128-pin Development Board

Note: The DoCD HAD2 module is not included in the AX110xx Development Kit.

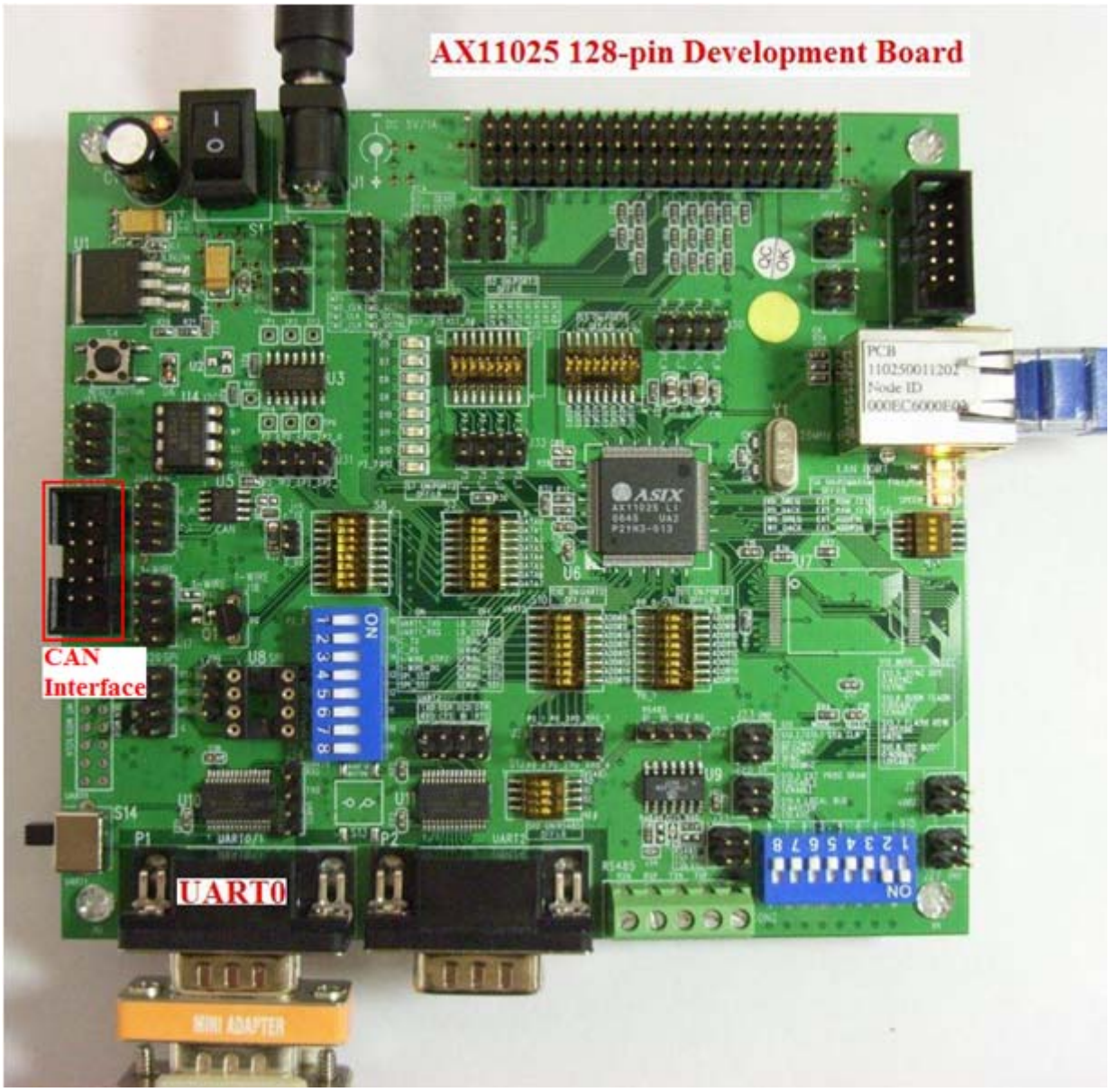


Figure 3. AX11025 128-pin Development Board

2. Quick Start

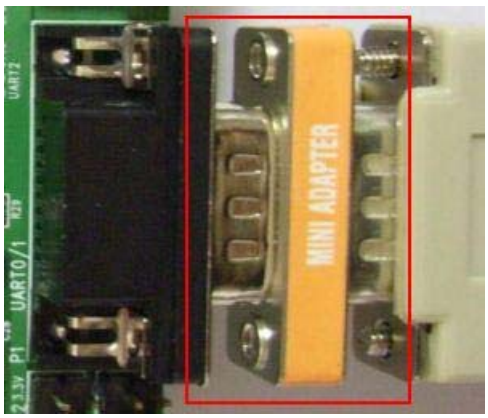
This section describes the information about how to start evaluating the AX110xx development board and implementing the firmware code on the AX110xx target applications.

The AX110xx Development Kit CD provides all the necessary AX110xx technical documents, reference schematics, demo firmware source codes and utilities. Users just need to prepare the Keil IDE development tool (refer to **Section 9-1: Software Compiler Tool** for details) before implementing AX110xx firmware on your target applications. The DCD's DoCD HAD2 debugger is optional for AX110xx (refer to **Section 9-2: Software Debugger Tool** for details) because all AX110xx demo firmware source codes already support the UART console debugging function.

Please prepare the following equipments before starting to evaluate the AX110xx development board:

- Windows machine
- Keil IDE development tool (http://www.keil.com/c51/selector.asp)
- AX110xx development board with 1 5V/3A AC/DC power adapter
- AX110xx Development Kit CD
  - AX110xx development board demo firmware source code
  - AX110xx boot loader binary code
  - AX110xx I2C EEPROM utility binary code
  - AX110xx Windows ISP
  - AX110xx Device Finder utility
  - TFTPD32 TFTP/DHCP Server utility
- RS-232 NULL modem cable (*Note)
- RJ-45 Ethernet cable

Note: The following RS-232 Null modem converter should be connected to the RS-232 cable included in AX110xx development board package; otherwise, the AX110xx couldn't establish the connection with PC or other RS-232 devices.



AX110xx demo firmware source code supports two project files (BUILD\AX110xx.Uv2 and BUILD_RT\AX110xx.Uv2) for different AX110xx applications. The BUILD\AX110xx.Uv2 project file is suitable to the AX110xx applications that need not use the AX110xx Ethernet boot loader code and the starting address of AX110xx firmware is allocated at 0. The BUILD_RT\AX110xx.Uv2 project file is suitable to the AX110xx applications that work with AX110xx Ethernet boot loader code and the starting address of AX110xx firmware is allocated at 24K. The AX110xx Flash range between offset 16K and 24K can be used to store some software variables of AX110xx firmware such as the configuration settings of Device Finder utility. Please refer to the following figure for more details.

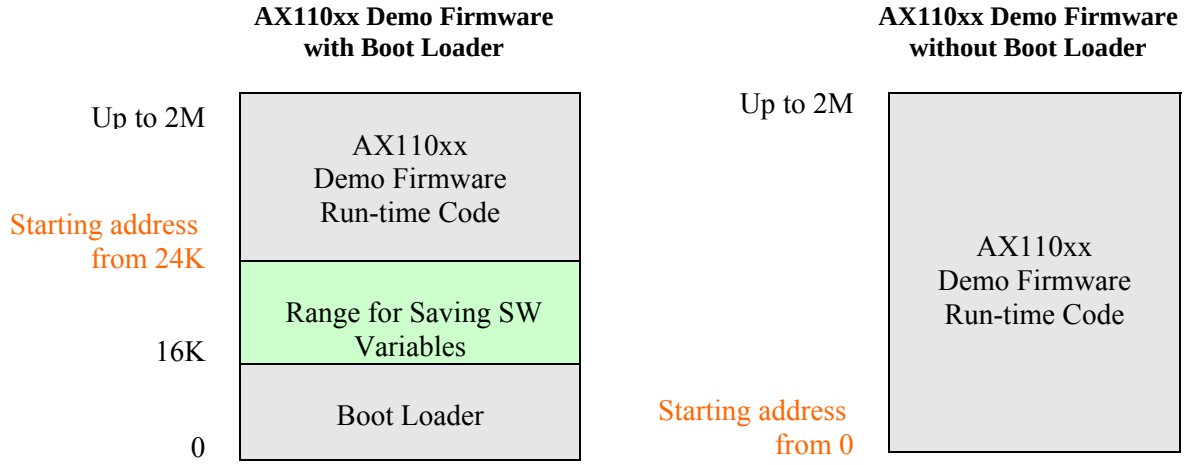
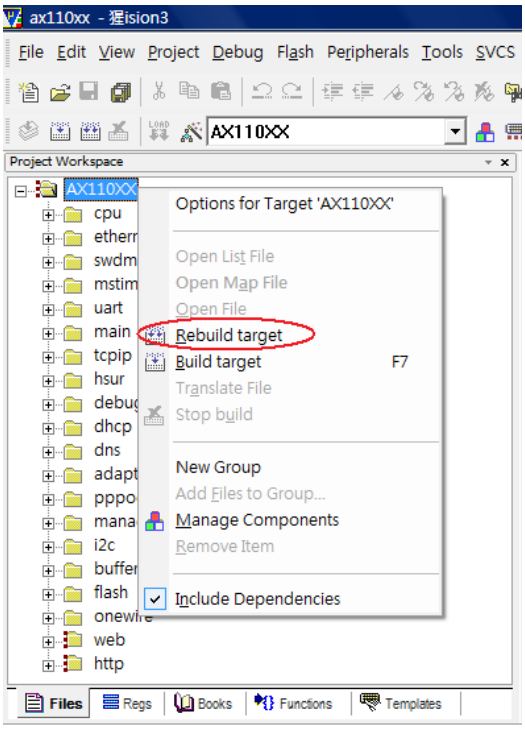


Figure 4. AX110xx Demo Firmware Run-time Code Starting Address

2-1. How to compile AX110xx Demo Firmware Source Code?

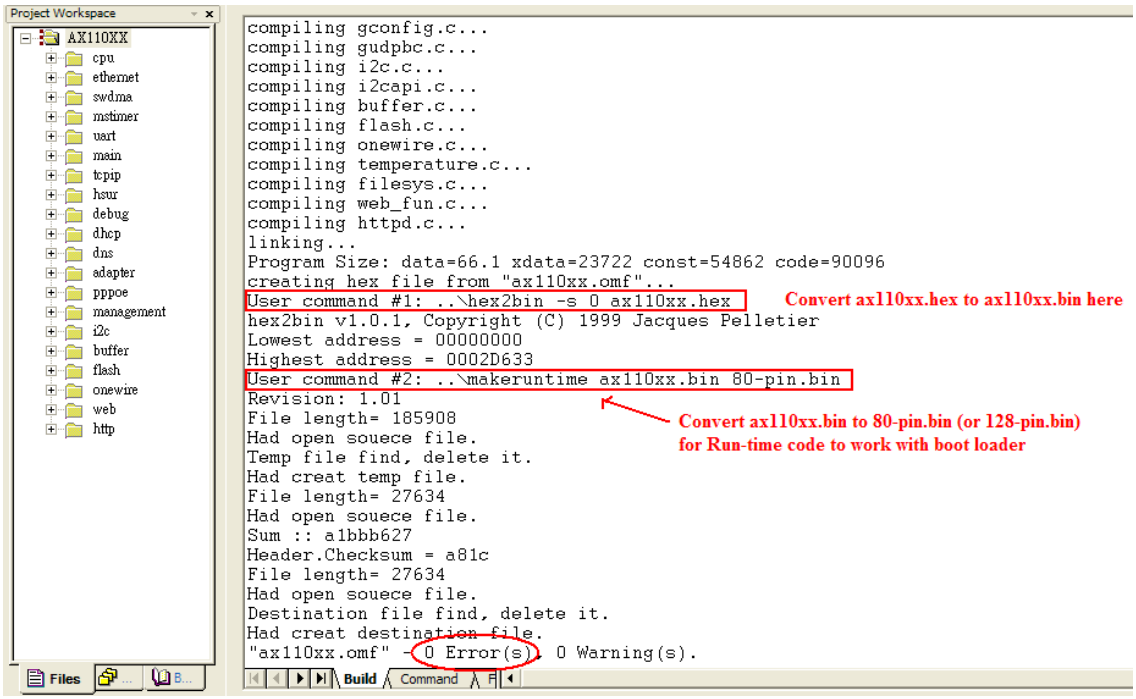
The following are the procedures to compile the AX110xx demo firmware source code. Please refer to **Section 4: How to compile AX110xx development board demo firmware** for more details.

a) Please copy a proper AX110xx demo firmware source code (e.g. the whole "AX1100x 80-pin Development Board" folder) from the "Demo Firmware" subdirectory of AX110xx Development Kit CD to the Windows machine (e.g. "D:\AX1100x 80-pin Development Board").
b) Change to the AX110xx demo firmware source folder (e.g. "D:\AX1100x 80-pin Development Board") on the Windows machine.
c) Open a proper project file (BUILD\AX110xx.Uv2 or BUILD_RT\AX110xx.Uv2) from the AX110xx demo firmware source code in the Keil IDE development tool.
d) Select **Rebuild target** on the Keil IDE to compile the AX110xx demo firmware source code.



e) After the source code was compiled successfully, you should see the following similar messages on the Keil IDE development tool. The following is the sample compiler messages for the BUILD_RT\AX110xx.Uv2 project file of AX1100x 80-pin demo firmware source code.

Note: The AX110xx.Uv2 project file is for Keil IDE uVision 2 & 3 while the AX110xx.uvproj project file is for Keil IDE uVision 4 only.



f) You can program the re-compiled AX110xx firmware code onto the AX110xx Flash through a proper AX110xx Flash Programming method. Please refer to **Section 2-2: How to program AX110xx Demo Firmware Code onto AX110xx Flash memory** for more details.

2-2. How to program AX110xx Demo Firmware Code onto AX110xx Flash memory?

There are two solutions to program the AX110xx firmware run-time code onto the AX110xx Flash memory of your target applications. One solution is to program the AX110xx firmware run-time code with starting address 0 onto AX110xx Flash memory by running the AX110xx Windows ISP tool directly; another one is to program the AX110xx firmware run-time code with starting address 24K onto AX110xx Flash memory through the AX110xx Ethernet boot loader. In this case, you should program the AX110xx Ethernet boot loader code onto AX110xx Flash memory by running the AX110xx Windows ISP tool first.

The AX110xx firmware run-time code with starting address 0 compiled by using the BUILD\AX110xx.Uv2 project file should be programmed onto the AX110xx Flash memory by using the Windows ISP UART0 Flash programming method. Please refer to **Section 7-1: COM Port Flash Programming Method Under Windows** for more details.

The AX110xx firmware code with starting address 24K compiled by using the BUILD_RT\AX110xx.Uv2 project file should be programmed onto the AX110xx Flash memory by using the AX110xx Ethernet Boot Loader Flash programming method. Please refer to **Section 7-3: Ethernet Boot Loader Flash Programming Method** for more details.

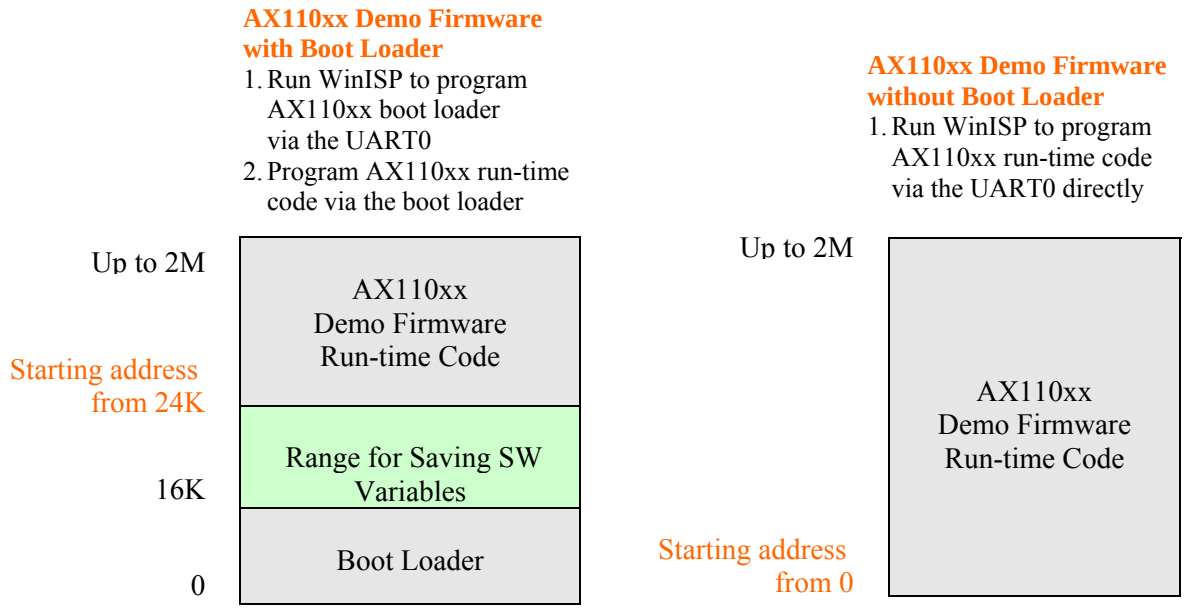


Figure 5. AX110xx Demo Firmware Code Flash Programming Methods

2-3. How to verify AX110xx Demo Firmware functionality?

The demo system illustration of AX110xx development board is shown in below figure.

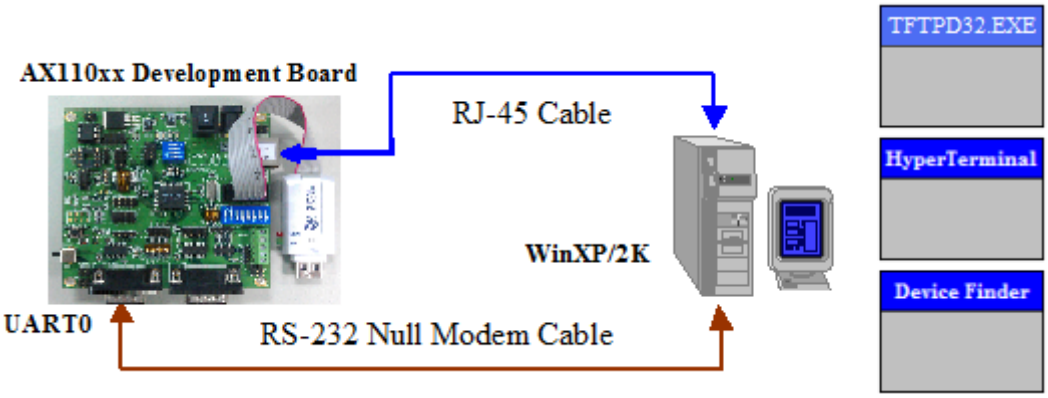


Figure 6. AX110xx Development Board Demo System

AX110xx development board demo firmware provides a basic Web Server to support the LED Control and Temperature Sensor functions and will request an IP address from DHCP server. If there is no DHCP server on the network, AX110xx development board firmware will assign a default IP address (e.g. **192.168.0.3)**.

The following are the detailed procedures about how to verify AX110xx demo firmware functionality,

1. Connect the Ethernet port of AX110xx development board and the Ethernet port of Windows PC with a RJ-45 cable.
2. Connect the UART0 interface of AX110xx development board and the COMx port of Windows machine with a RS-232 Null modem cable.

> **Note:** This step is optional and can be skipped if the user need not check the UART0 debugging messages of AX110xx demo firmware.

3. Connect the 5V/3A AC/DC power adapter to AX110xx development board.
4. Boot up the Windows machine with Ethernet interface that is configured to a proper IP address (e.g. 192.168.0.190).

5. Run Hyper Terminal application to create a connection between AX110xx board and the Windows machine, the COMx port should be set to 9600 baud rate, 8 data bit, NO parity check, 1 stop bit and NO flow control.

Note: Windows 7/Vista doesn't support the Hyper Terminal Tool; please find the third party utility (e.g. Tera Term) to see the UART debug messages on these Operation Systems.

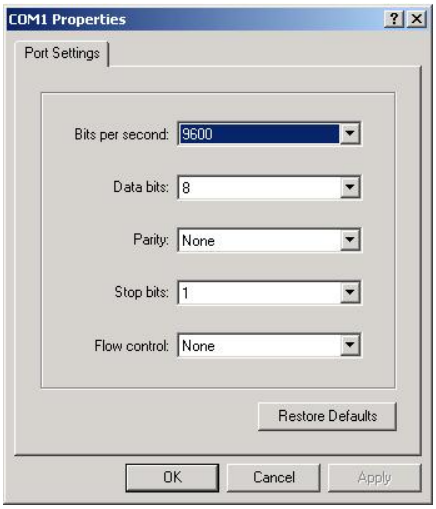


Figure 7. Windows machine COM Port Setting Example

6. Startup the TFTPD32 DHCP/TFTP Server utility (i.e. TFTPD32.EXE) on the Windows machine and follow the following procedures to configure the TFTPD32 DHCP Server. Please refer to below figure for more details.

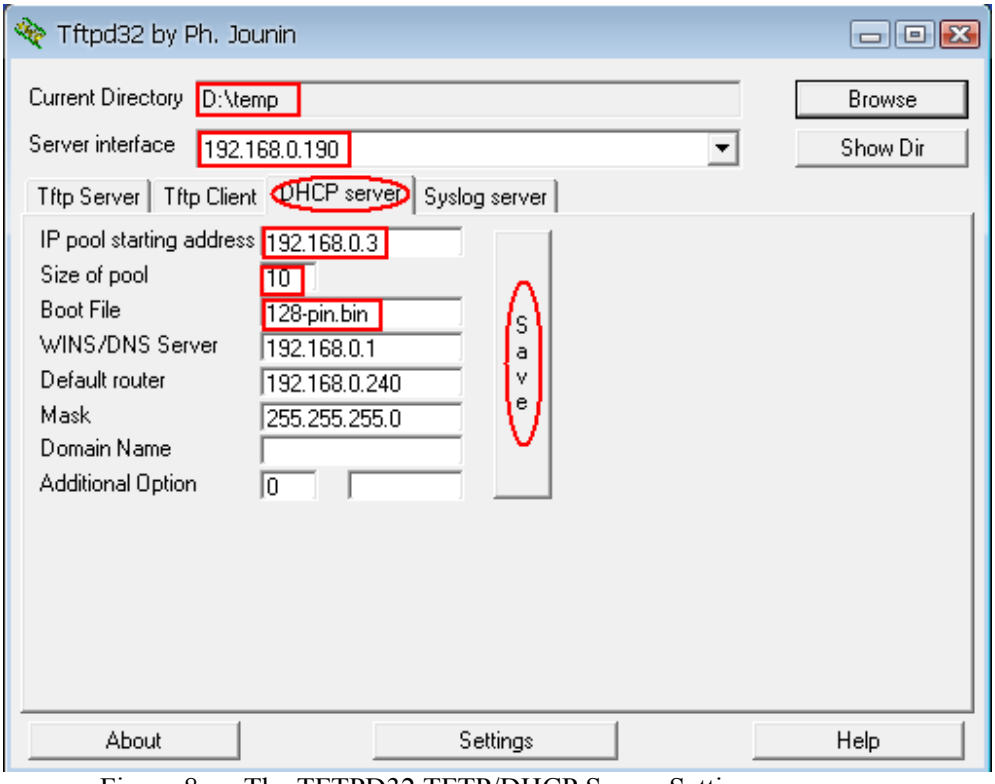


Figure 8. The TFTPD32 TFTP/DHCP Server Setting

a) Set the “IP pool starting address“ field (e.g. 192.168.0.3) to the same subnet mask with the “Server interfaces IP address” (e.g. 192.168.0.190).
b) Set the “Size of pool” field to a number larger than 0 (e.g. 10).
c) Press the [Save] button to take effect the new setting.

7. Power ON the AX110xx development board, and then you should see the following debug messages of AX110xx Ethernet boot loader and demo firmware run-time code on the HyperTerminal console.

```
AX110xx bootloader v3.1.0 version for DoCD HAD Debugger

1. Execute Runtime.
2. Download runtime via RS232.
3. Download runtime via Ethernet.
4. Download new bootloader via Ethernet.

Please input 1,2,3 or 4 to execute above :

Get new IP : c0a8000a
Get subnet mask : ffffff00
Get gateway : c0a80001
TFTP server IP : c0a80032
Download file total length = 0x2c0cf bytes.
download ok.

Wait runtime code.
UART-0 init ok.
UART-2 init ok.

ASIX AX110xx 128PIN Demo Firmware V3.0.0 (06/07/11 16:39:53)
DHCP init ok.
GCONFIG_Init()..
…
DHCP request... IP: 192.168.0.3
…
  12.EEPROM DATA: 21 3c 0f 80 30 00 54 0c 00 c6 0e 00 f2 05 10 e0
                  1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
                  ff 04
GUDPBC_Init()...
```

8. Now, you can run the AX110xx Device Finder utility to search the AX110xx development board and enter the AX110xx web server by pressing the “Web Browser” button on the Device Finder main window. Please refer to **Section 3-1: Device Finder Utility** for more details.

2-4. How to start implementing my own AX110xx firmware?

The AX110xx Development Kit CD provides the workable AX110xx demo firmware source codes. Customers can easily implement their own AX110xx firmware based on the AX110xx demo firmware source code.

Before start modifying the AX110xx demo firmware source code to reach the requirements of your AX110xx target applications, we strongly suggest you to study the **AX110xx Software User Guide** and **AX110xx Upper Protocol Developer Guide** first. These documents describe the detailed information of all AX110xx driver and upper protocol modules.

Please refer to **Section 8: Software Sample codes** for the details of all available AX110xx drivers and upper protocols. You can get all AX110xx driver modules source codes from the AX110xx Development Kit CD. If you need the available AX110xx upper protocol source codes, please contact ASIX's Sales (Sales@asix.com.tw) for more details.

3. Software Function Description

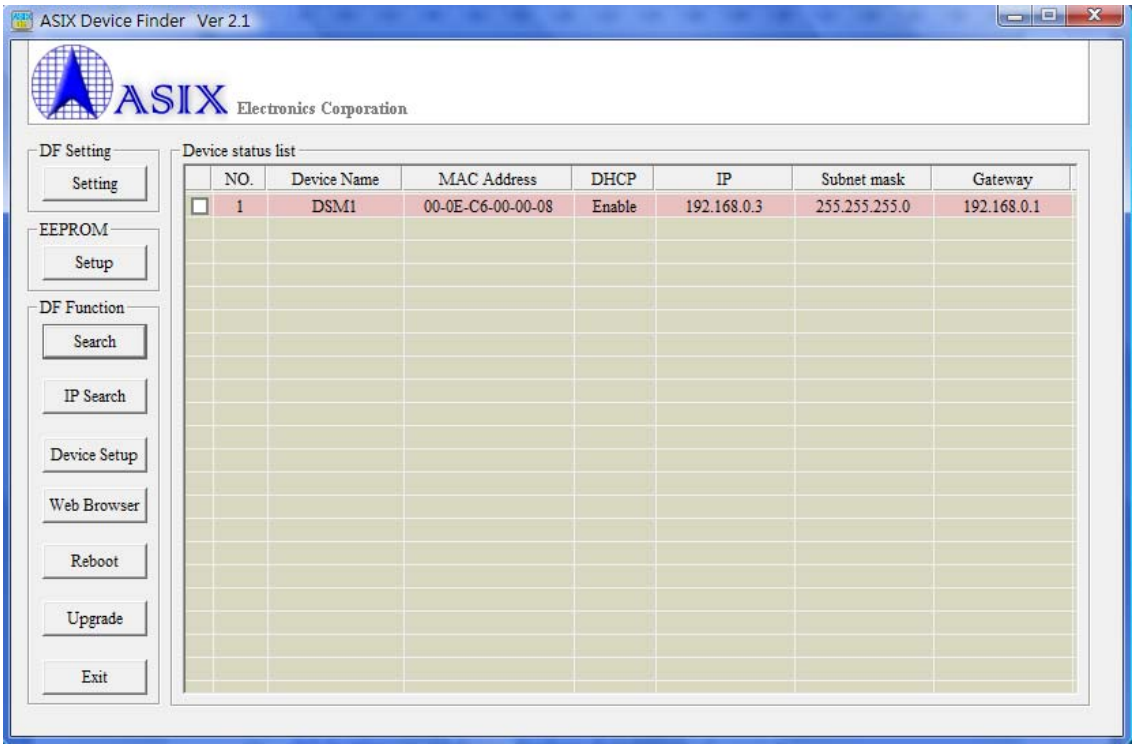
3-1. Device Finder Utility

This section describes the detailed functions of the Device Finder utility.

Note: The Device Finder utility can only work with AX110xx development board demo firmware. Please make sure if the AX110xx development board demo firmware was already programmed into the Flash of AX110xx development board before running this utility.

3-1-1. Main Window

When running Device Finder the main window will appear,



The main window mainly provides the following functions,

(1) **DF Setting**: to configure the Search and Upgrade period.
(2) **EEPROM**: to configure the EEPROM data of AX110xx board.
(3) **Search**: to search available device server(s) on the LAN.
(4) **IP Search**: to search the device server with specified IP address.
(5) **Device Setup**: to configure the settings of selected device server(s).
(6) **Web Browser**: Open remote configuration web server of selected device server(s).
(7) **Reboot**: to restart the selected device server(s).
(8) **Upgrade**: to upgrade the run-time firmware code of selected device server(s).
(9) **Exit**: Quit this application.

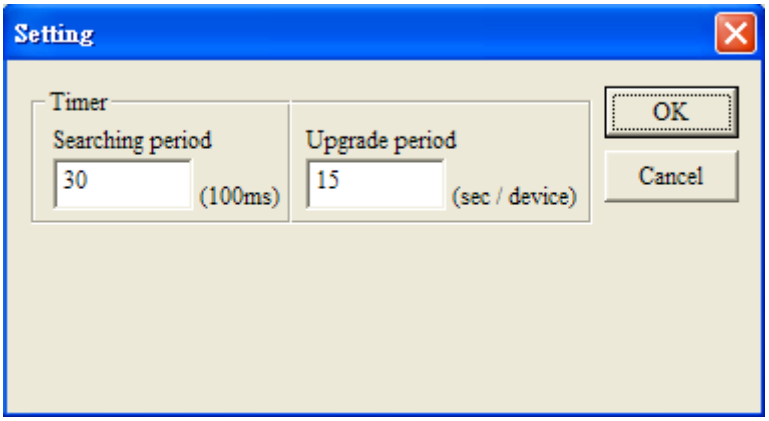
After executing [Search] function, if any device servers are found, they will be added in the Device Server List and the following information is displayed:

Field	Description
No	Device server index
Device Name	Device server name, 16 bytes maximum string
MAC Address	Device server MAC address
DHCP	Enable or disable
IP	- If DHCP is enabled, dynamic IP is acquired from the DHCP server, - Or, static IP is assigned as dynamic IP.
Subnet mask	Subnet mask IP address
Gateway	Gateway IP address

Users can check the selected device box in the "Device status list" window and click a proper button (i.e. the [Device Setup], [Web Browser], [Reboot] or [Upgrade] button) on the DeviceFinder main window to run the selected function.

3-1-2. DF Setting Dialog

When click [Setting] button of DF Setting on main window, the Setting dialog will appear,



The Setting dialog provides two functions,

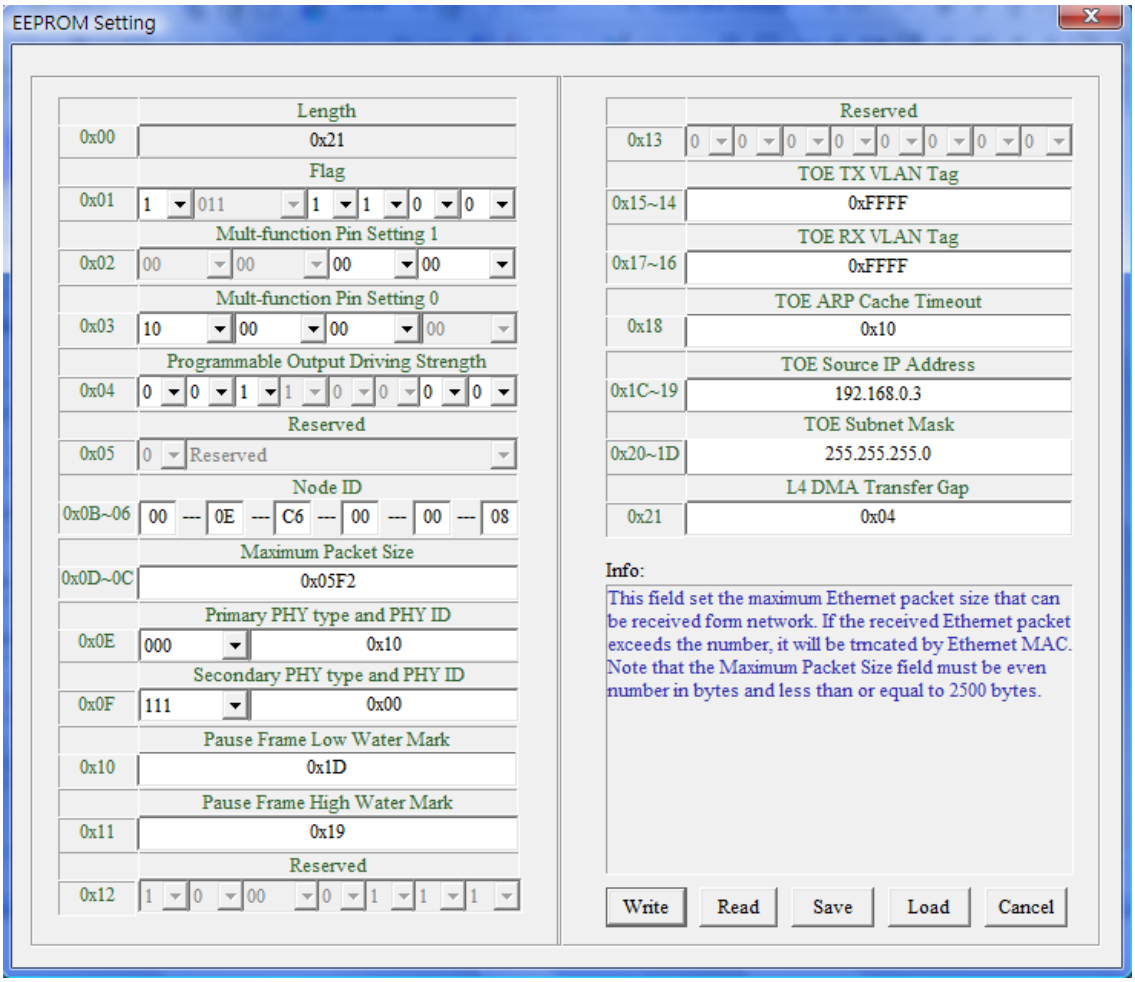
(1) **OK**: Enable the new period setting.
(2) **Cancel**: cancel the new period setting.

The Setting dialog provides following parameters,

Parameter	Description
Searching period (100ms)	Set the search timeout period
Upgrade period (sec /device)	Set the firmware upgrade timeout period

3-1-3. EEPROM Dialog

Users can check the selected device box in the "Device status list" window and click the [Setup] button of EEPROM dialog on main window, the following dialog will appear,



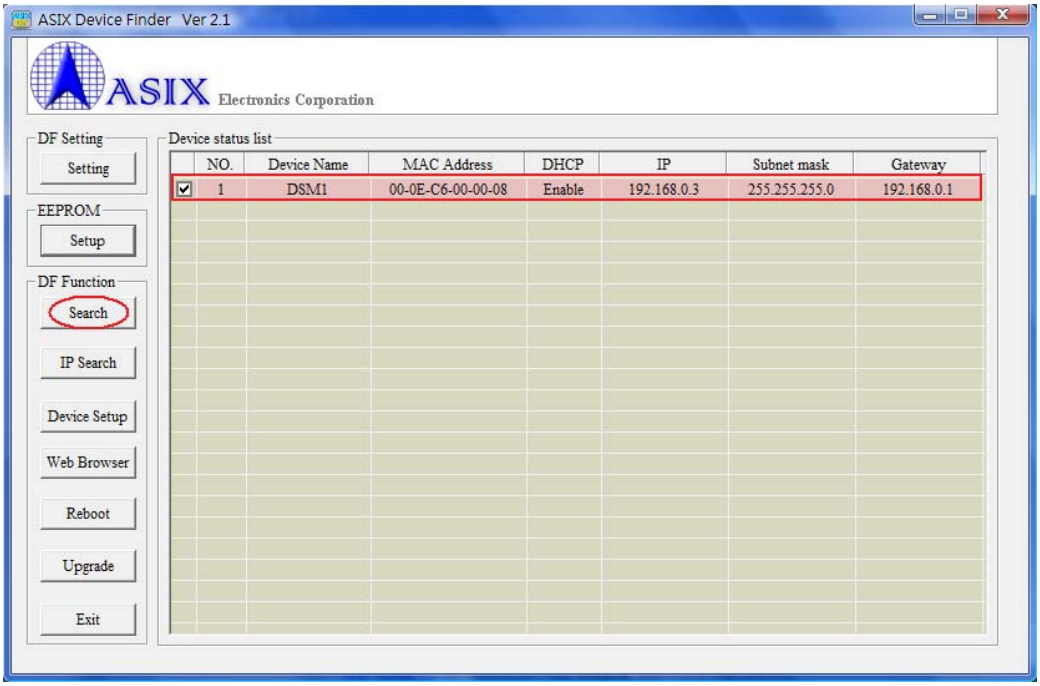
The Device Setup dialog provides four functions,

(1) **Write**: write data to the I2C EEPROM.
(2) **Read**: read data from the I2C EEPROM.
(3) **Save**: save the EEPROM settings to a file.
(4) **Load**: read a set of EEPROM settings from a file.
(5) **Cancel**: cancel/quit the EEPROM settings.

Note: Please refer to Section 3.1 of AX110xx datasheet for details of AX110xx EEPROM format.

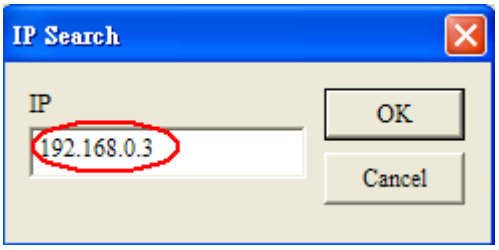
3-1-4. Search Dialog

When click [Search] button on main window, the found AX110xx device will be shown in the "Device status list".



3-1-5. IP Search Dialog

When click [IP Search] button on main window, the IP Search dialog will appear,



The IP Search dialog provides two functions,

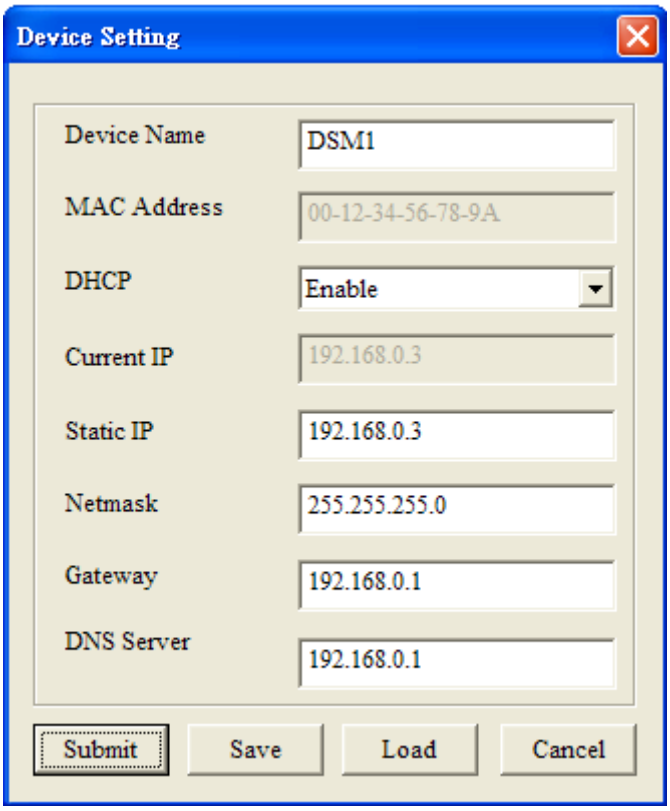
(1) **OK**: start the search operation
(2) **Cancel**: cancel the search operation.

The IP Search dialog provides following parameters,

Parameter	Description
IP	The device server's IP address

3-1-6. Device Setup Dialog

Users can check the selected device box in the "Device status list" window and click the [Device Setup] button on main window, and then the following Device Setup dialog will appear,



The Device Setup dialog provides four functions,

(1) **Submit**: submit new settings.
(2) **Save**: save the settings to a file.
(3) **Load**: read a set of settings from a file.
(4) **Cancel**: cancel new settings.

The Device Setting dialog provides following parameters,

Parameter	Description
Device Name	Search via UDP multicast packet
MAC Address	MAC address
DHCP	Enable / Disable
Current IP	Current IP address
Static IP	Static IP address saved in AX110xx Flash
Netmask	Subnet mask IP address
Gateway	Gateway IP address
DNS Server	DNS server IP address

3-2. AX1100x 80-pin Development Board Web Server

This section describes the detailed functions of AX1100x 80-pin development board web server.

3-2-1. Web Server Main Page

Users can check the selected device box in the "Device status list" window and click the [Web Browser] button on main window, and then the following AX110xx web server main page will appear,

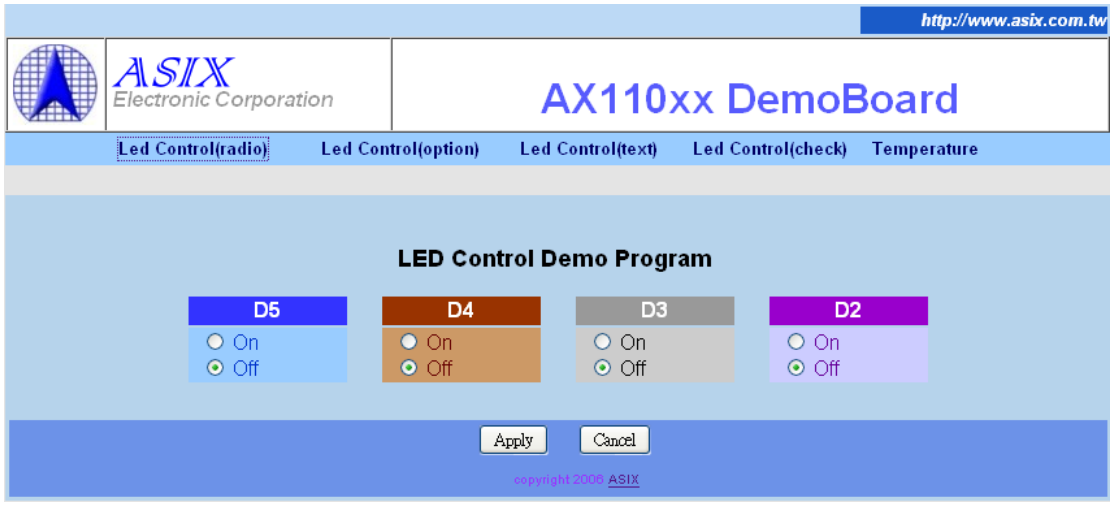
Alternately, users can also open the AX110xx web server main page by entering the AX110xx web server IP address on the web browser (e.g. http://192.168.0.3) and then the following web page will appear.

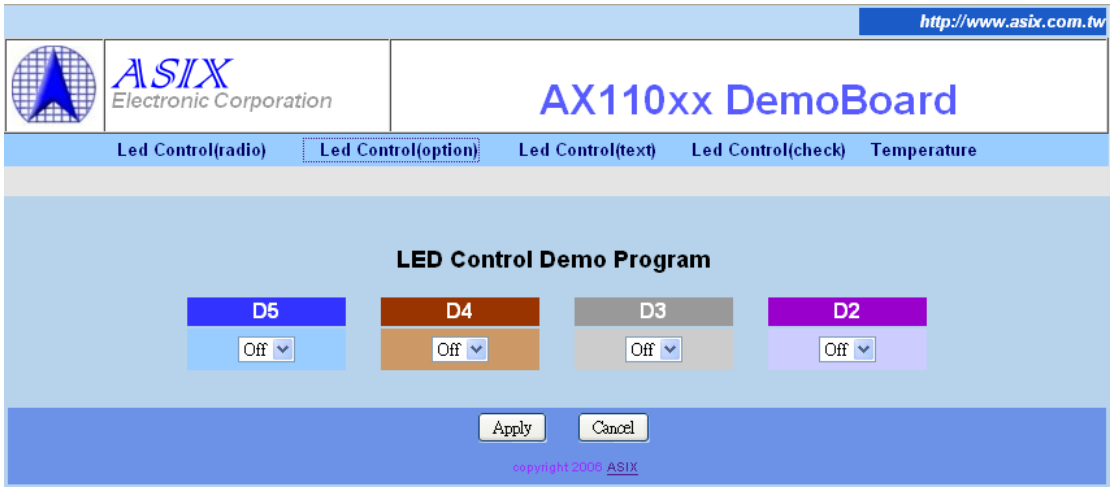
The main page mainly provides two functions,

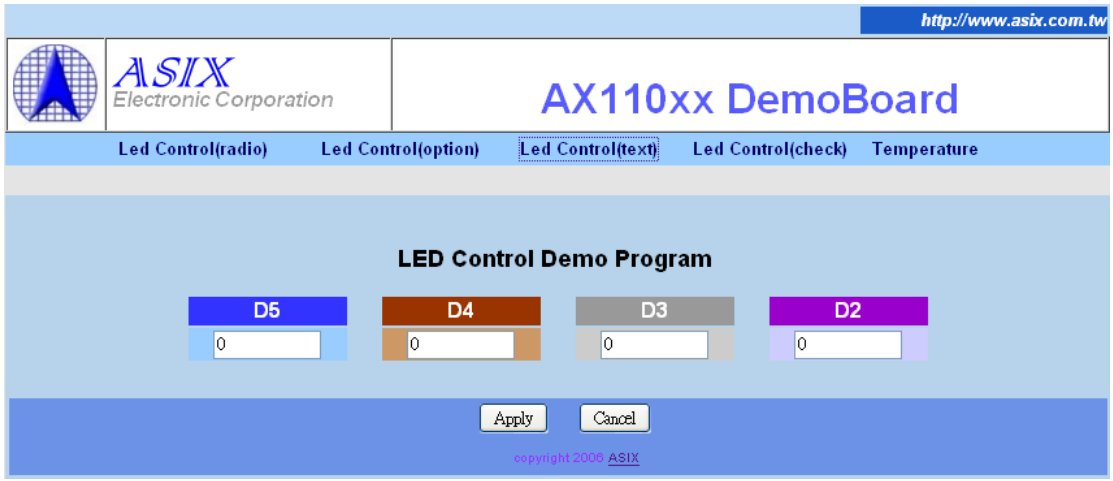
(1) **Led Control (radio button/option/text/check)**: to go to LED Control web pages.
(2) **Temperature**: to go to Temperature Sensor web page.

3-2-2. LED Control Web Page

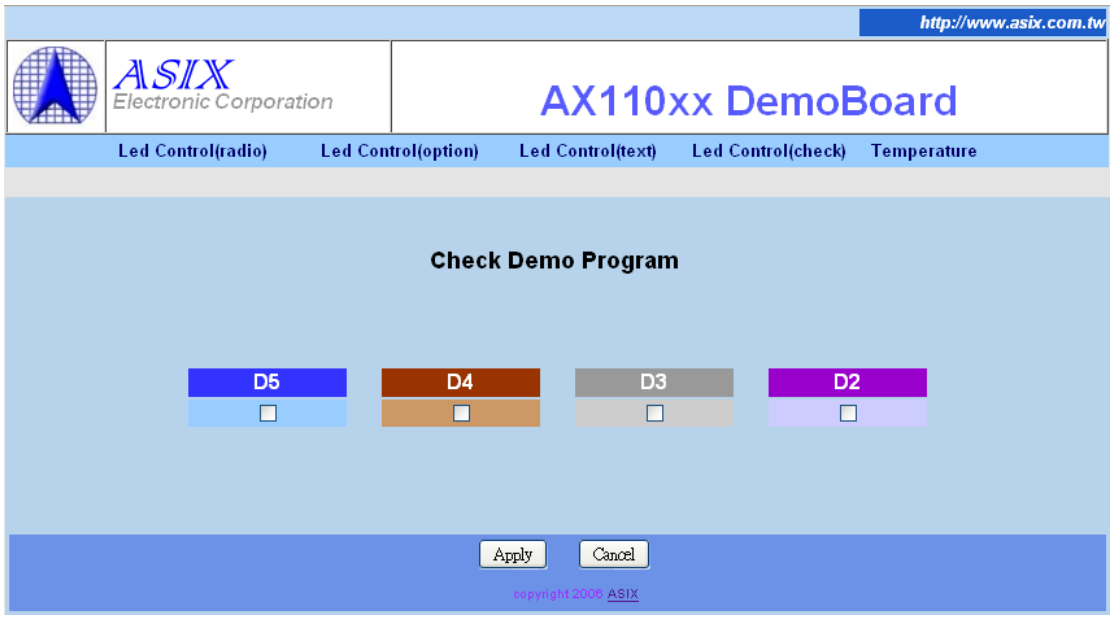
The user can turn ON/OFF the LEDs of AX1100x 80-pin development board on these web pages.





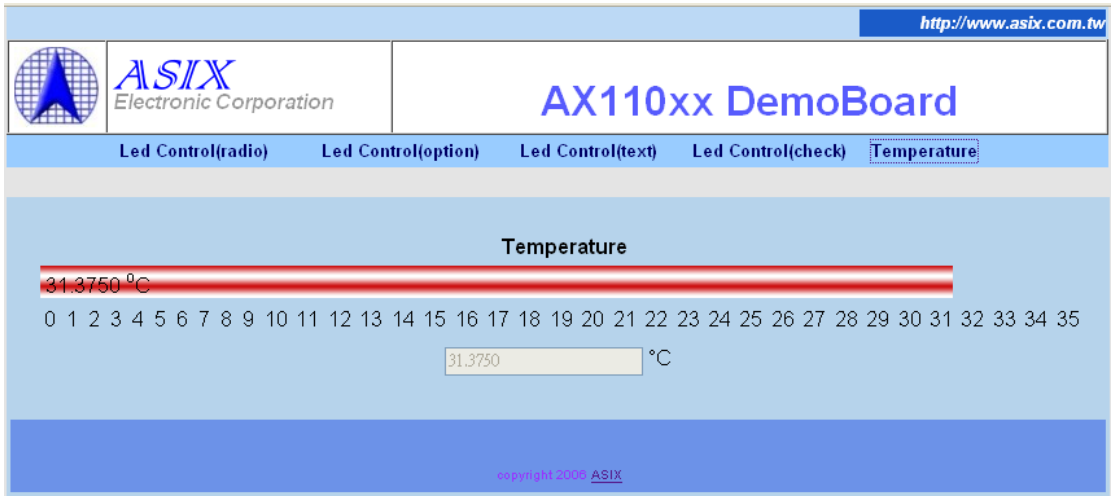


Note: Set to 0 will turn OFF LED and set to non-zero will turn ON LED.



3-2-3. Temperature Sensor Web Page

The user can monitor the temperature values reported from the 1-Wire Temperature Sensor of AX1100x 80-pin development board on this web page.



3-3. AX11015 128-pin Development Board Web Server

This section describes the detailed functions of AX11015 128-pin development board web server.

3-3-1. Web Server Main Page

Users can check the selected device box in the "Device status list" window and click the [Web Browser] button on main window, and then the following AX110xx web server main page will appear,

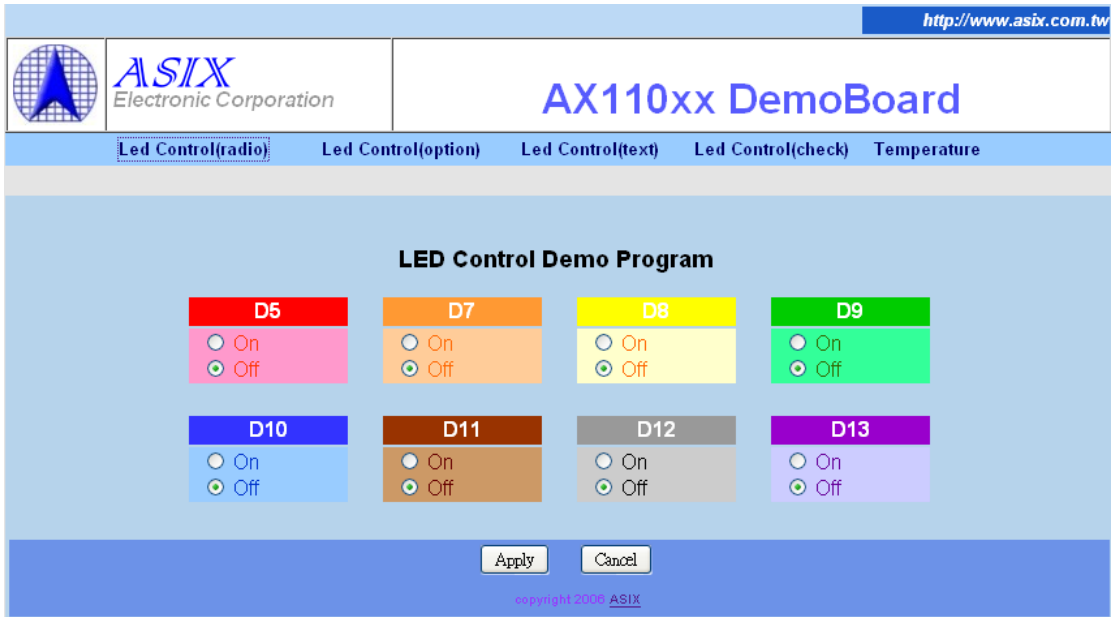
Alternately, users can also open the AX110xx web server main page by entering the AX110xx web server IP address on the web browser (e.g. http://192.168.0.3) and then the following web page will appear.

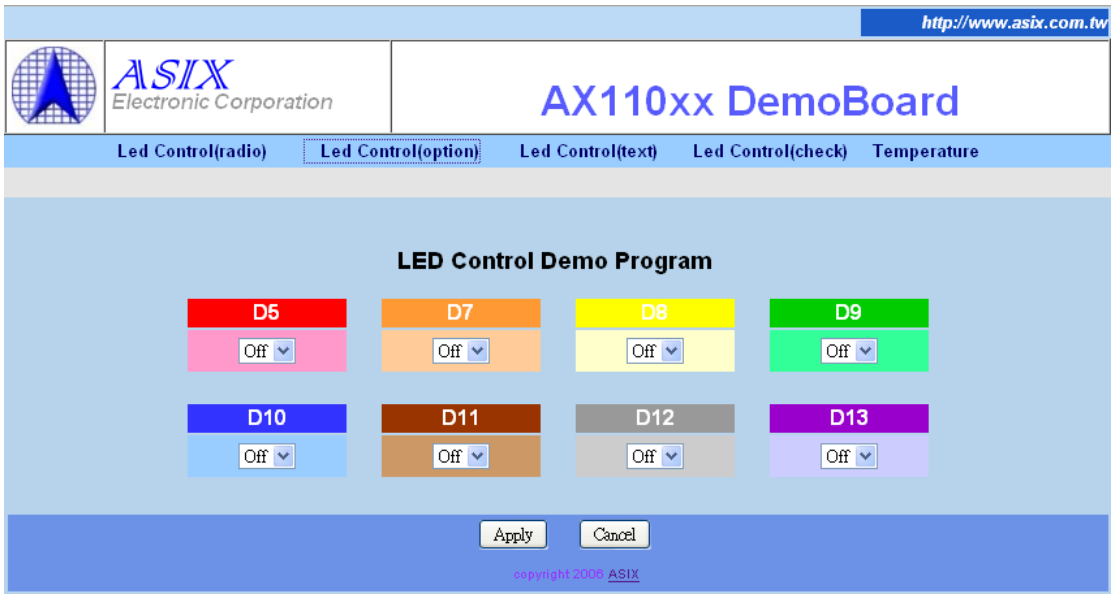
The main page mainly provides two functions,

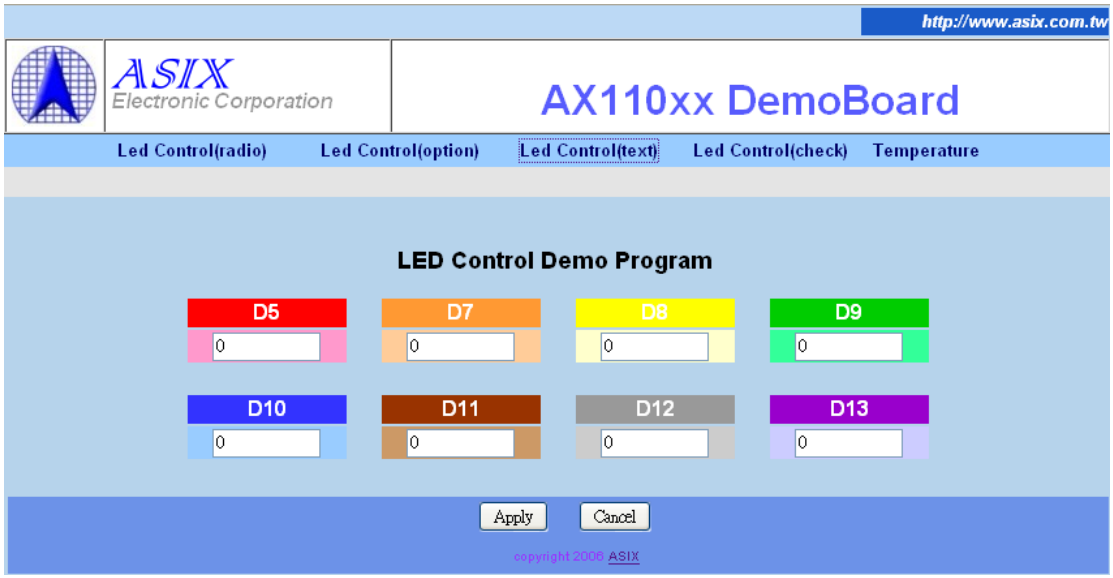
(1) **Led Control (radio button/option/text/check)**: to go to LED Control web pages.
(2) **Temperature**: to go to Temperature Sensor web page.

3-3-2. LED Control Web Page

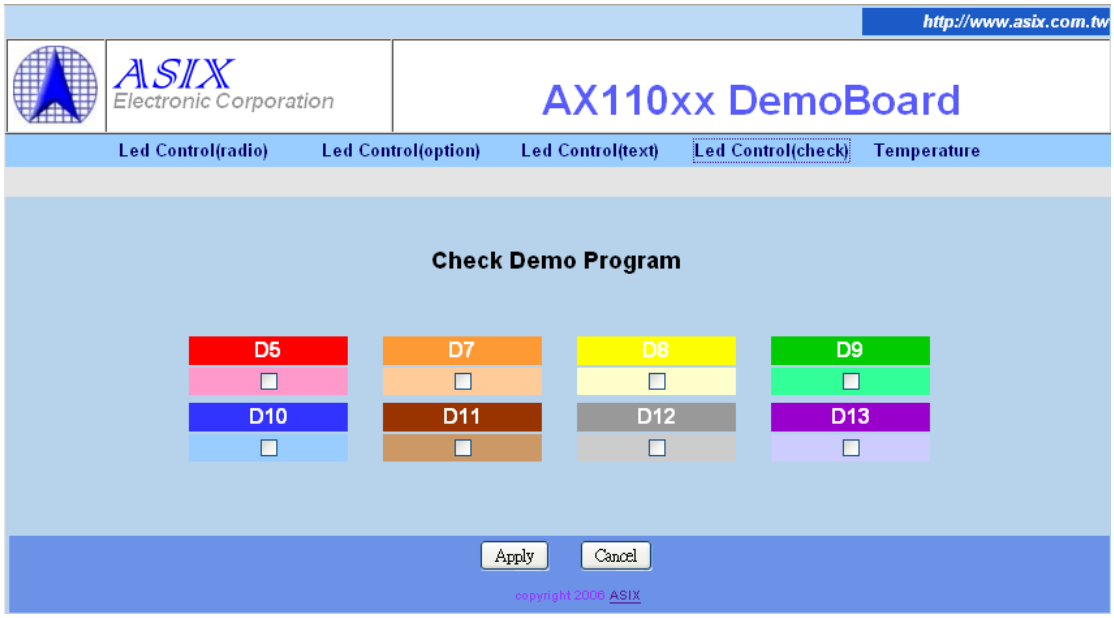
The user can turn ON/OFF the LEDs of AX11015 128-pin development board on these web pages.





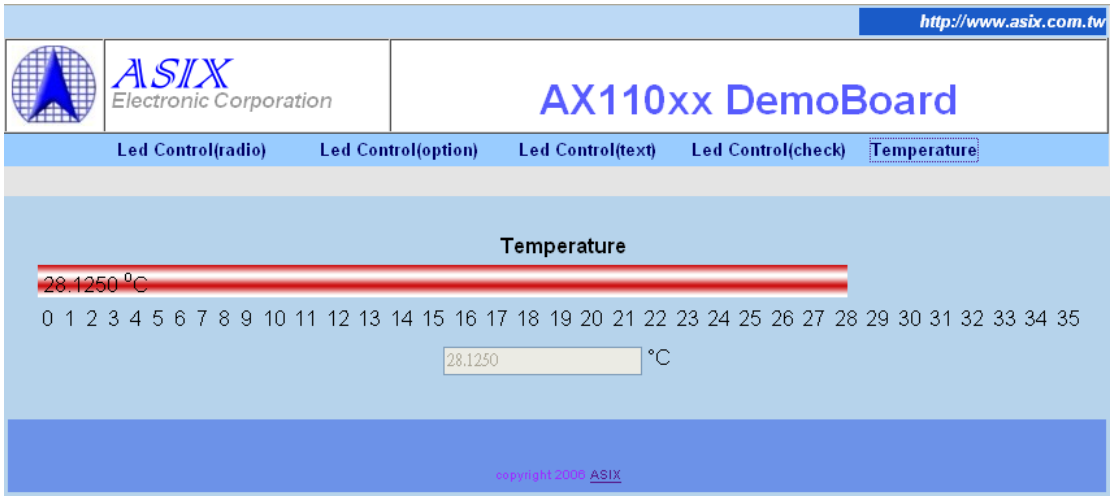


Note: Set to 0 will turn OFF LED and set to non-zero will turn ON LED.



3-3-3. Temperature Sensor Web Page

The user can monitor the temperature values reported from the 1-Wire Temperature Sensor of AX11015 128-pin development board on this web page.



3-4. AX11025 128-pin Development Board Web Server

This section describes the detailed functions of AX11025 128-pin development board web server.

3-4-1. Web Server Main Page

Users can check the selected device box in the "Device status list" window and click the [Web Browser] button on main window, and then the following AX110xx web server main page will appear,

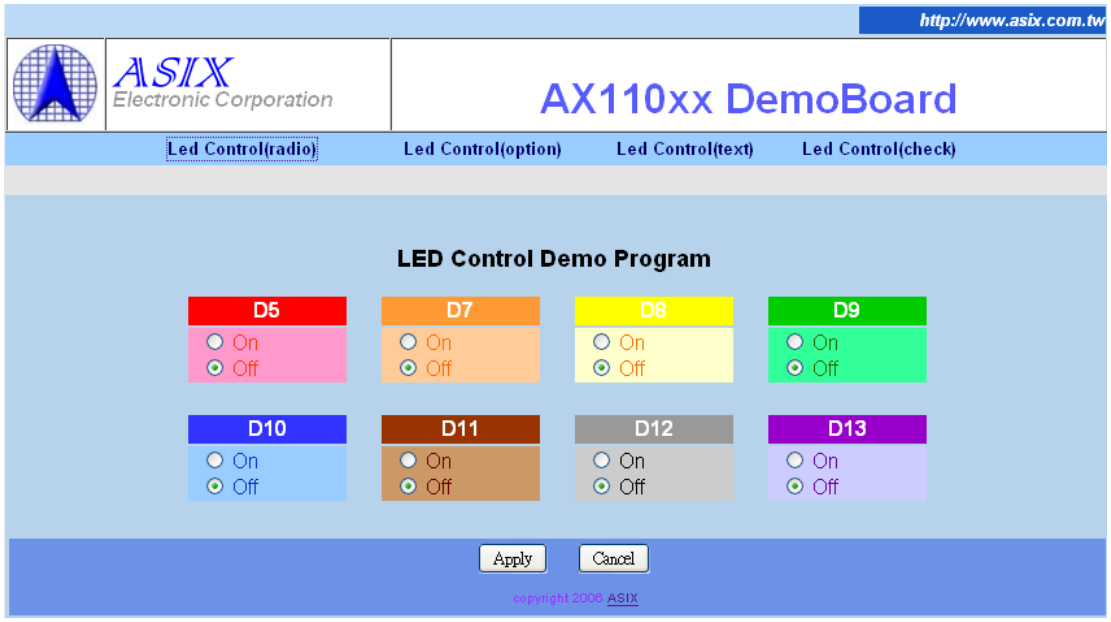
Alternately, users can also open the AX110xx web server main page by entering the AX110xx web server IP address on the web browser (e.g. http://192.168.0.3) and then the following web page will appear.

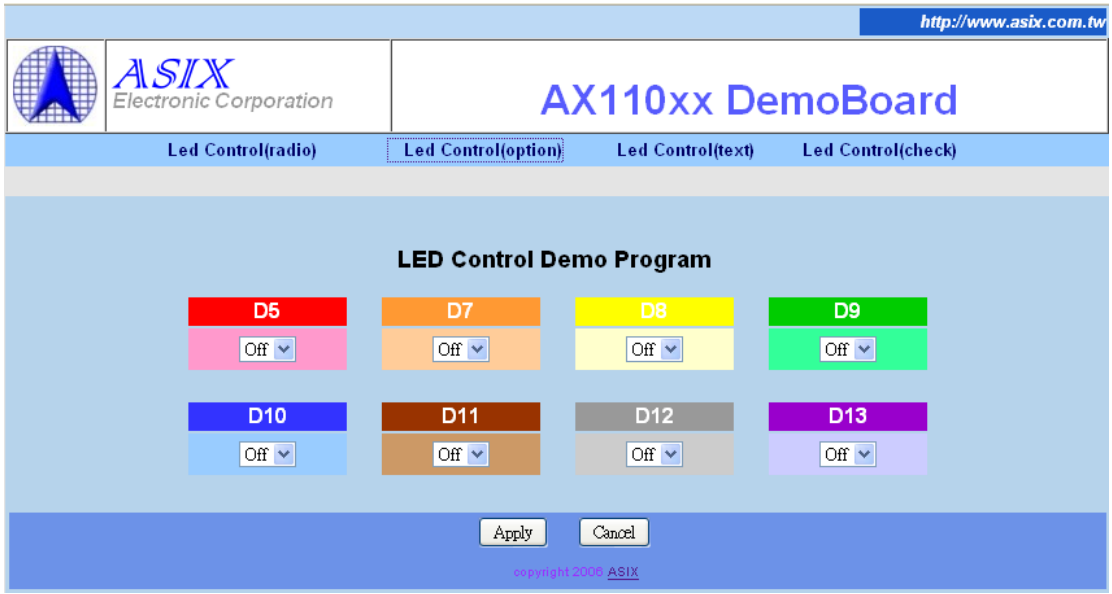
The main page mainly provides the LED control function,

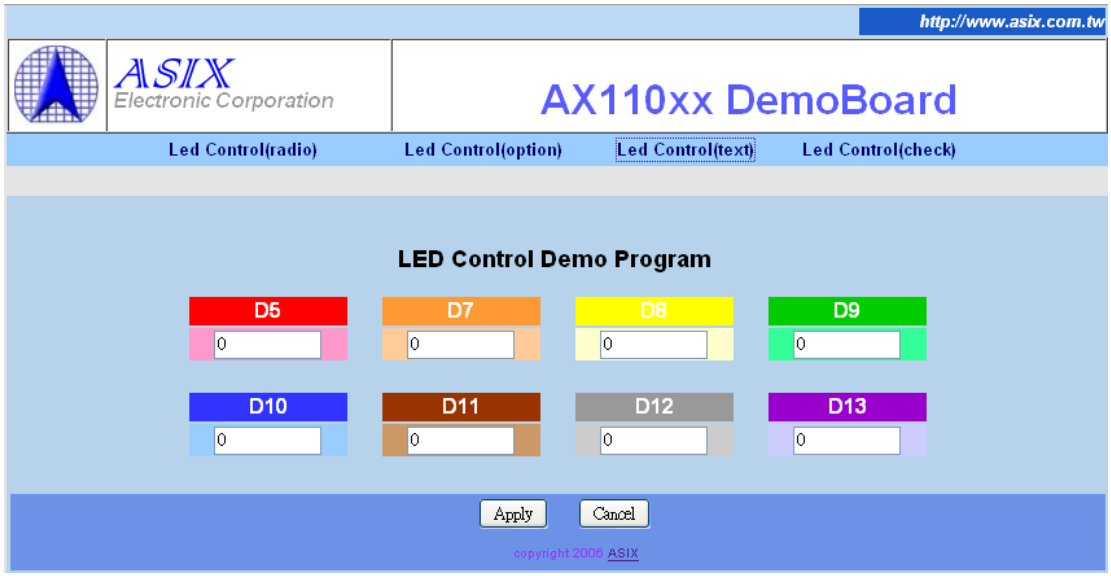
(1) **Led Control (radio button/option/text/check)**: to go to LED Control web pages.

3-4-2. LED Control Web Page

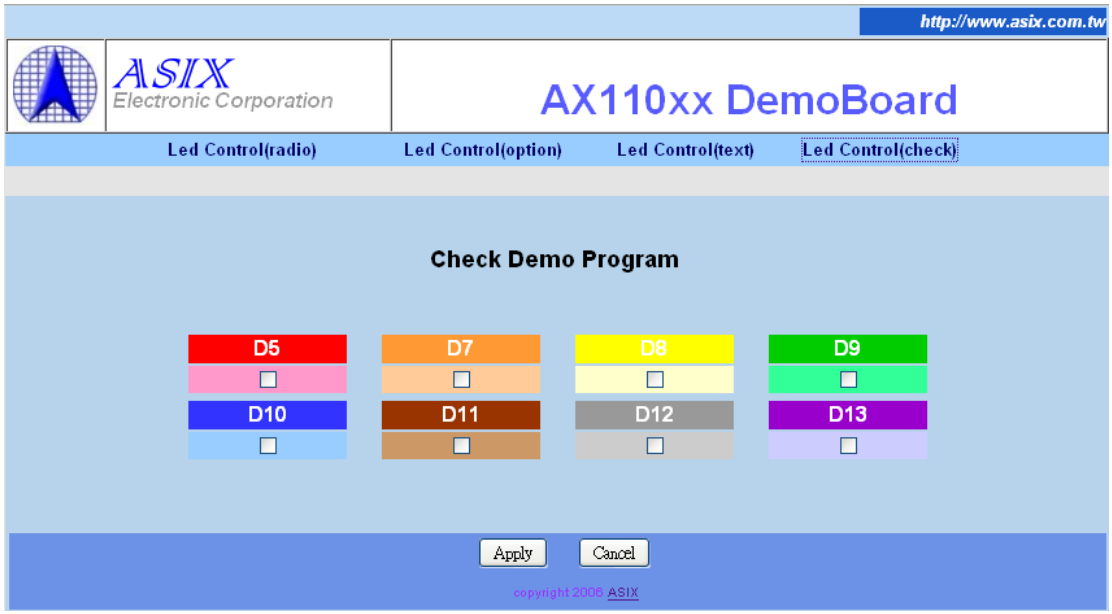
The user can turn ON/OFF the LEDs of AX11025 128-pin development board on these web pages.







Note: Set to 0 will turn OFF LED and set to non-zero will turn ON LED.



4. How to compile AX110xx development board demo firmware

AX110xx supports three demo firmware source codes, AX1100x 80-pin demo firmware source code, AX11015 128-pin demo firmware source code and AX11025 128-pin demo firmware source code. These AX110xx demo firmware source codes are developed in C language under Keil IDE development environment. Users can purchase the Keil IDE Development Tool from Keil's web site (http://www.keil.com/c51/selector.asp). In general, users need to purchase the PK51 development tool for C-language compiler, debugger and simulator. Those users only needing the compiler function can purchase the CA51 package.

1. Double click "build\ax110xx.uv2" or "build_rt\ax110xx.uv2" file to open AX110xx project by Keil C Development Tool. (Refer to below Note for details)

Note: AX110xx development board demo firmware source code includes two project files ("build\ax110xx.uv2" and "build_rt\ax110xx.uv2") to build two different binary codes,

a) If users want to program the demo firmware code to AX110xx Flash via AX110xx UART 0 interface. Please build the demo firmware code by running the "BUILD/ax110xx.uv2" project file.

b) If users want to program the demo firmware code to AX110xx Flash via AX110xx Ethernet Boot Loader interface. Please build the demo firmware code by running the "BUILD_RT/ax110xx.uv2" project file.

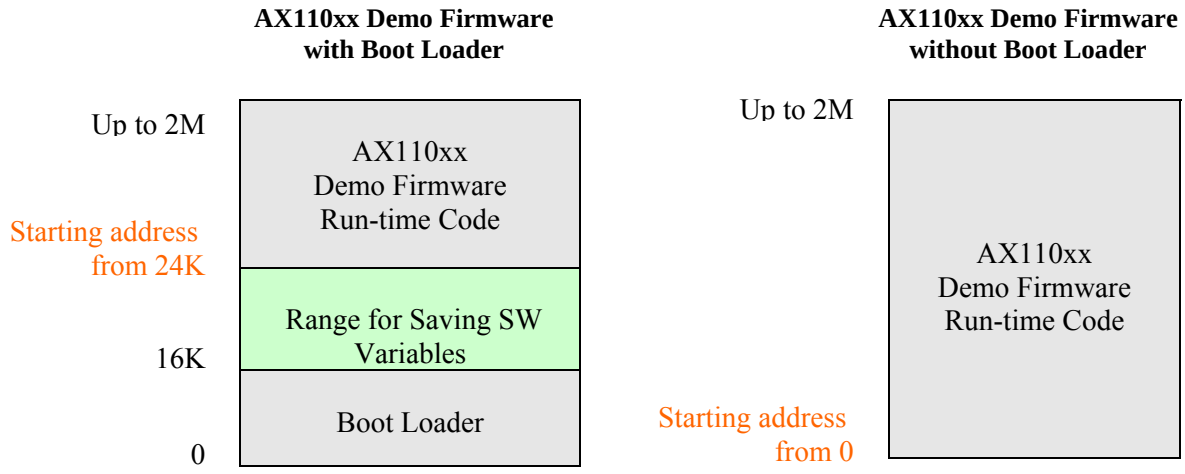


Figure 9. AX110xx Demo Firmware Run-time Code Starting Address

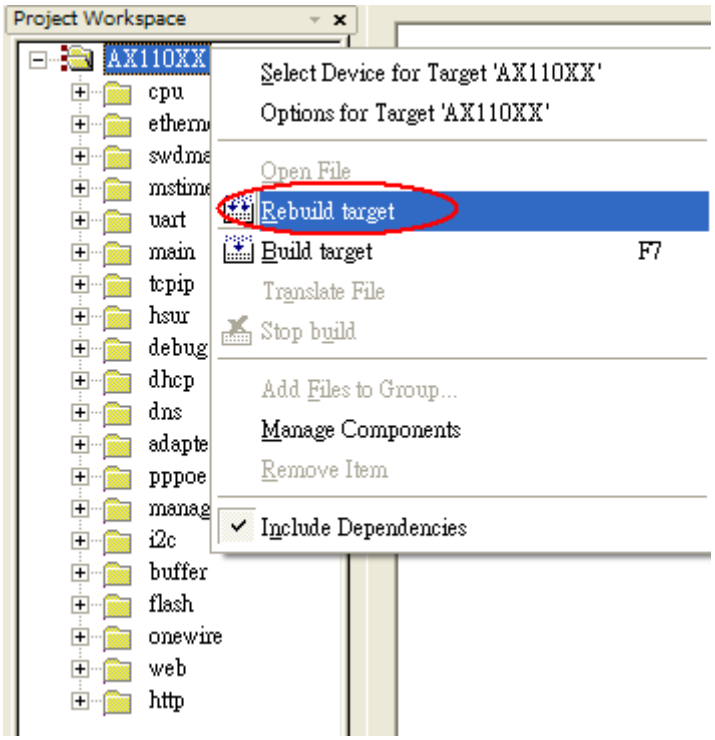
2. Select “Rebuild Target” item from the property menu of AX110xx project to rebuild AX110xx development board demo firmware code.

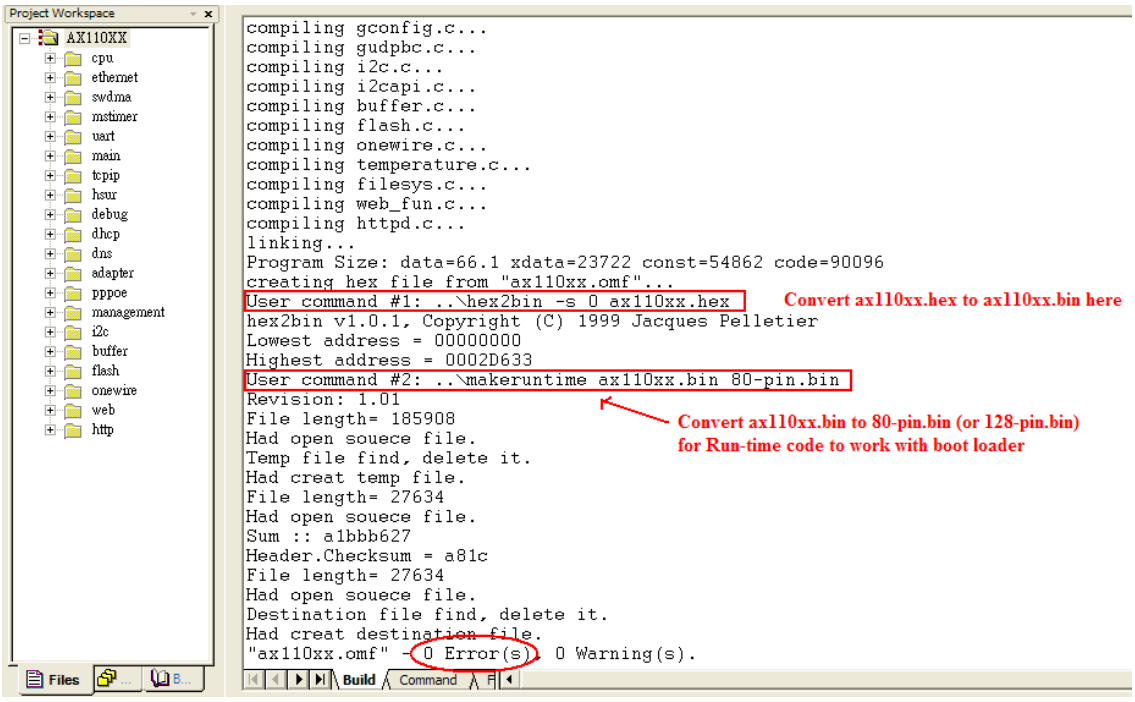
Note: Both AX110xx project files (build\ax110xx.uv2 and build_rt\ax110xx.uv2) will auto-run the “HEX2BIN.EXE” utility to convert the AX110xx.HEX into AX110xx.BIN file for AX110xx Flash Programming utility (UARTL.EXE).

User command #1: HEX2BIN –s 0 AX110xx.HEX

Note: The AX110xx project file (build_rt\ax110xx.uv2) will auto-run the “MakeRuntime.exe” utility to convert the header field of AX110xx firmware code to work with AX110xx Boot Loader code for AX110xx Ethernet Boot Loader TFTP download.

User command #2: MAKERUNTIME AX110xx.BIN 80-PIN.BIN (for 80-pin)
User command #2: MAKERUNTIME AX110xx.BIN 128-PIN.BIN (for 128-pin)





3. Start to program the AX110xx demo firmware code into AX110xx Flash. Please refer to the following AX110xx Flash programming section for details

5. Development Board DIP Switches Setup

5-1. AX1100x 80-pin Development Board DIP Switches Setting

AX1100x 80-pin development board provides a couple of switches (S1~S8) for different purpose configurations.

Note:
1. The S3, S4 and S8 DIP switches are used to configure multi-function pins setting, users MUST make sure these DIP switches setting are the same as the setting of the Multi-function Pin Setting 0, 1 (offset 02h, 03h) of AX11001/AX11005 I2C EEPROM.
2. Please make sure the poles of S4 and S8 DIP switches are switched together based on the multi-function pins setting of the Multi-function Pin Setting 0, 1 (offset 02h, 03h) of AX11001/AX11005 I2C EEPROM.

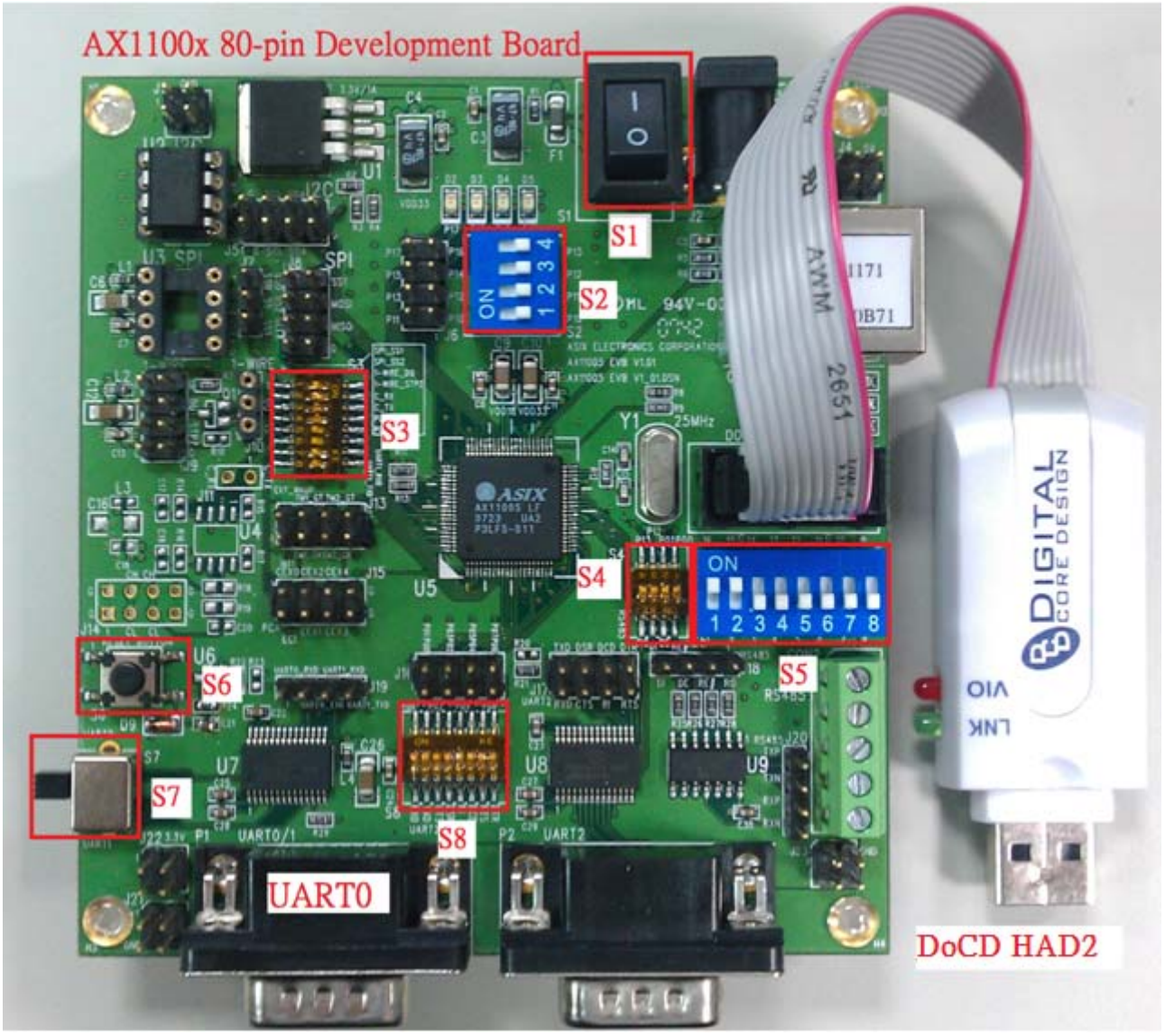


Figure 10. The DIP Switches of AX1100x 80-pin Development Board

5-1-2. S1 Switch Setting

The S1 switch is used to power ON/OFF the development board.

5-1-3. S2 Switch Setting

The S2 switch is used to pull up/down the signals of the Port 1 GPIO pins. Please refer to AX1100x 80-pin development board schematics for details.

S2 Pole no.	Pin Name	ON	OFF
1	P10	Pull Down	Pull Up
2	P11	Pull Down	Pull Up
3	P12	Pull Down	Pull Up
4	P13	Pull Down	Pull Up

5-1-4. S3 Switch Setting

The S3 switch is used to control the SPI, 1-Wire, PCA, and UART1 multi-function pins. The S3 switch H/W setting should match the Multi-function Pin Setting 1 (offset 03h) of AX11001/AX11005 I2C EEPROM. Users can set Pole #7 and #8 to ON to enable the PCA function or set to OFF to enable the UART 1 function. Please refer to AX1100x 80-pin development board schematics and Section 3.1.3 of AX11001/AX11005 datasheet for details.

S3 Pole no.	ON	OFF
1	SPI_SS1	SS1
2	SPI_SS2	SS2
3	1WIRE_DQ	SS1
4	1WIRE_STPZ	SS2
5	(Always OFF)	SS1
6	(Always OFF)	SS2
7	PCA_ECI	UART1_RXD
8	PCA_CEX0	UART1_TXD

The (Pole #1, #2) and (Pole #3, #4) could not be set to ON at the same time because the SPI_SS1 and SPI_SS2 pins are shared with 1WIRE_DQ and 1WIRE_STPZ pins respectively. For example,

(1) To enable SPI_SS1 and SPI_SS2 interfaces,

S3 Pole no.	Pin Name	Setting
1	SPI_SS1	ON
2	SPI_SS2	ON
3	1WIRE_DQ	OFF
4	1WIRE_STPZ	OFF
5	-	OFF
6	-	OFF

(2) To enable 1-Wire interface,

S3 Pole no.	Pin Name	Setting
1	SPI_SS1	OFF
2	SPI_SS2	OFF
3	1WIRE_DQ	ON
4	1WIRE_STPZ	ON
5	-	OFF
6	-	OFF

Note: If users want to enable the 1-Wire Temperature Sensor of AX1100x 80-pin development board, the Pole #3 and #4 of S3 switch should be set to ON to enable 1-Wire interface.

5-1-5. S4 Switch Setting

The S4 switch is used to control the Port 0,1 GPIO or RS485 multi-function pins. The S4 switch H/W setting should match the Multi-function Pin Setting 0 (offset 02h) of AX11001/AX11005 I2C EEPROM. Please refer to AX1100x 80-pin development board schematics and Section 3.1.3 of AX11001/AX11005 datasheet for details.

S4 Pole no.	ON	OFF
1	RS485_RXD2	P00
2	RS485_TXD2	P01
3	RS485_DE	P12
4	RS485_RE_N	P13

Note: The RXD and TXD pins of UART2 interface are shared with the RXD2 and TXD2 pins of RS-485 interface so when the RS-485 interface is enabled (i.e. all poles of S4 are set to ON), the UART2 interface should be disabled (i.e. all poles of S8 should be set to OFF).

5-1-6. S5 Switch Setting

The S5 switch is used to configure some general configurations for AX1100x 80-pin development board. Please refer to AX1100x 80-pin development board schematics for details.

The Pole #5 (Burn Flash Enable) of S5 switch should be set to ON for UART Flash programming and should be set to OFF for Ethernet Boot Loader Flash programming and normal operation. The Pole #6 (Burn Flash 921K) of S5 switch should be set to OFF for normal UART interface and can be set to ON if the user's machine supports 921KB high speed UART mode.

S5 Pole no.	Function	ON	OFF
2-1	System Clock Select	00: 25MHz; 01: 50MHz; 10: Reserved; 11: 100MHz	
3	Local Bus Mode	Slave	Master

4	Bus Mode	Sync. Bus Mode	Async. Bus Mode
5	Burn Flash Enable	Enable UART Flash Programming function	Disable UART Flash Programming function
6	Burn Flash 921K	High Speed UART mode (921Kbps)	UART mode (115200bps)
7	I2C Boot Disable	Disable I2C Boot	Enable I2C Boot
8	Test SpeedUp	Enable	Disable

5-1-7. S6 Switch Setting

The S6 switch is used to reset the development board.

5-1-8. S7 Switch Setting

The S7 switch is used to set the UART0/1 interface to UART 0 or UART 1. Users can set S7 switch to UART 0 for UART console debugging or UART Flash programming purposes or set to UART 1 for customer specific purpose.

S7 UART Interface	Comments
UART 0	For debugging console or Flash programming
UART 1	For customer specific purpose

5-1-9. S8 Switch Setting

The S8 switch is used to control the Port 0 GPIO and UART2 multi-function pins. The S8 switch H/W setting should match the Multi-function Pin Setting 0 (offset 02h) of AX11001/AX11005 I2C EEPROM. Users can set all poles of S8 switch to ON to enable the UART2 interface. Please refer to AX1100x 80-pin development board schematics and Section 3.1.3 of AX11001/AX11005 datasheet for details.

S8 Pole no.	ON	OFF
1	UART2_RXD	P00
2	UART2_TXD	P01
3	UART2_CTS	P02
4	UART2_DSR	P03
5	UART2_RI	P04
6	UART2_DCD	P05
7	UART2_RTS	P06
8	UART2_DTR	P07

Note: The RXD and TXD pins of UART2 interface are shared with the RXD2 and TXD2 pins of RS-485 interface so when the UART2 interface is enabled (i.e. all poles of S8 are set to ON), the RS-485 interface should be disabled (i.e. all poles of S4 should be set to OFF).

5-2. AX11015/AX11025 128-pin Development Board DIP Switches Setting

AX11015/AX11025 128-pin development boards provide a couple of switches for different purpose configurations.

Note:
1. The S2, S3, S6, S7, S8, S10, S11 and S12 DIP switches are used to configure multi-function pins setting, users MUST make sure these DIP switches setting are the same as the setting of the Multi-function Pin Setting 0, 1 (offset 02h, 03h) of AX11015/AX11025 I2C EEPROM.
2. Please make sure the poles of S2, S3, S6, S7, S10, S11 and S12 DIP switches are switched together based on the multi-function pins setting of the Multi-function Pin Setting 0, 1 (offset 02h, 03h) of AX11015/AX11025 I2C EEPROM.

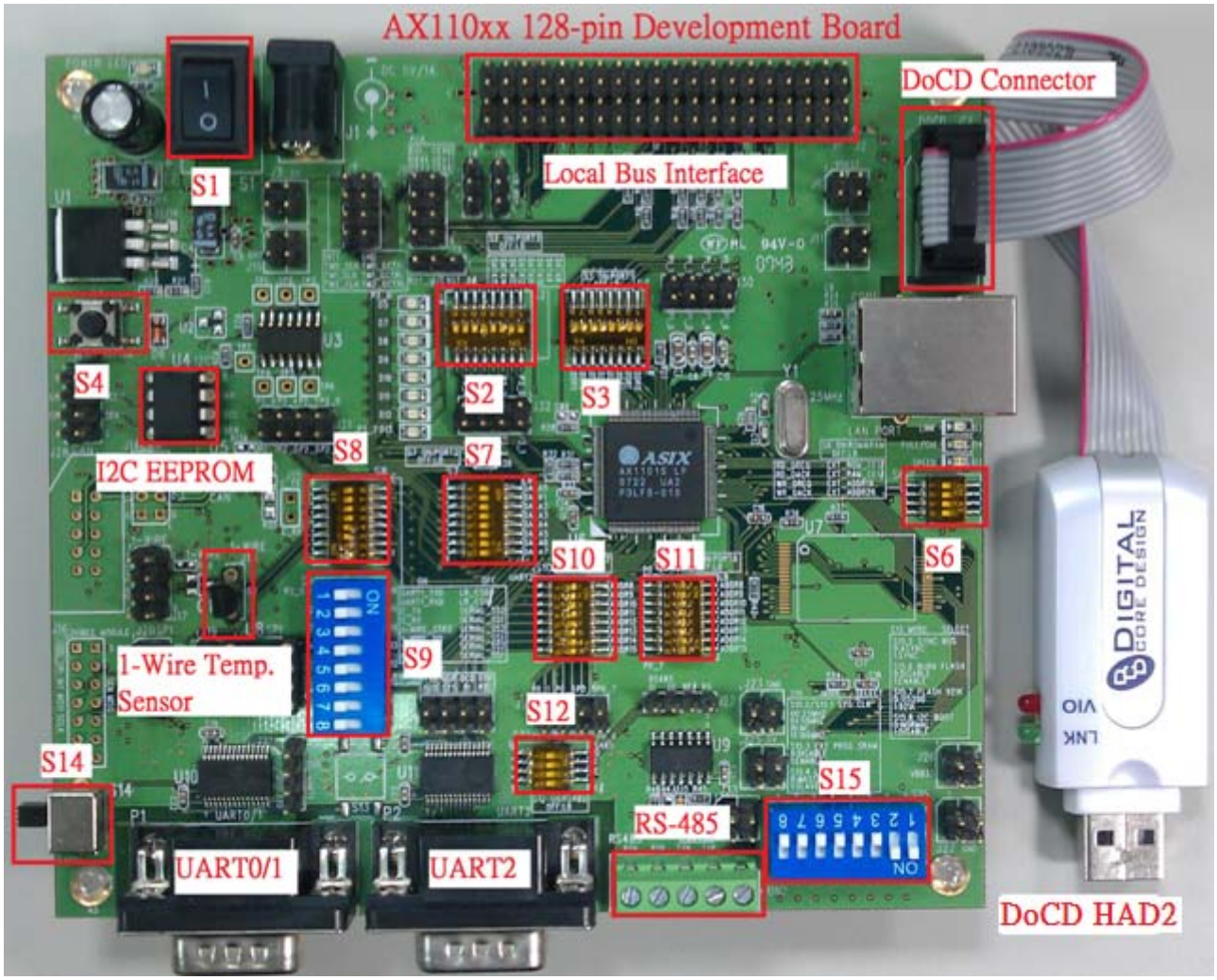


Figure 11. The DIP Switches of AX11015/AX11025 128-pin Development Boards

5-2-1. S1 Switch Setting

The S1 switch is used to power ON/OFF the development board.

5-2-2. S2 Switch Setting

The S2 switch is used to control the Port 3 GPIO, Local Bus and MII multi-function pins. The S2 switch H/W setting should match the Multi-function Pin Setting 0 (offset 02h) of AX11015 I2C EEPROM. Please refer to AX11015/AX11025 128-pin development board schematics and Section 3.1.3 of AX11015/AX11025 datasheets for details.

S2 Pole no.	ON	OFF
1	P30	LB_LDA8/MII_TX_CLK
2	P31	LB_LDA9/MII_MTXD0
3	P32	LB_LDA10/MII_MTXD1
4	P33	LB_LDA11/MII_MTXD2
5	P34	LB_LDA12/MII_MTXD3
6	P35	LB_LDA13/MII_TX_EN
7	P36	LB_LDA14/MII_TX_ER
8	P37	LB_LDA15/MII_COL

5-2-3. S3 Switch Setting

The S3 switch is used to control the Port 1 GPIO, Local Bus and MII multi-function pins. The S3 switch H/W setting should match the Multi-function Pin Setting 0 (offset 02h) of AX11015 I2C EEPROM. Please refer to AX11015/AX11025 128-pin development board schematics and Section 3.1.3 of AX11015/AX11025 datasheets for details.

S3 Pole no.	ON	OFF
1	P10	LB_LA0/MII_MDC
2	P11	LB_LA1/MII_MDIO
3	P12	LB_LA2
4	P13	LB_LA3
5	P14	LB_LA4
6	P15	LB_LA5
7	P16	LB_LA6
8	P17	LB_LA7

5-2-4. S4 Switch Setting

The S4 switch is used to reset the development board.

5-2-5. S6 Switch Setting

The S6 switch is used to control the external memory interface or Local Bus DMA signals multi-function pins. The S6 switch H/W setting should match the Multi-function Pin Setting 1 (offset 03h) of AX11015 I2C EEPROM. Please refer to AX11015/AX11025 128-pin development board schematics and Section 3.1.3 of AX11015/AX11025 datasheets for details.

S6 Pole no.	ON	OFF
1	EXT_ROM_CE1_N	LB_RD_DREQ
2	EXT_RAM_CE1_N	LB_RD_DACK
3	EXT_ADDR19	LB_WR_DREQ
4	EXT_ADDR20	LB_WR_DACK

5-2-6. S7 Switch Setting

The S7 switch is used to control the Port 2 GPIO, Local Bus and MII multi-function pins. The S7 switch H/W setting should match the Multi-function Pin Setting 0 (offset 02h) of AX11015 I2C EEPROM. Please refer to AX11015/AX11025 128-pin development board schematics and Section 3.1.3 of AX11015/AX11025 datasheets for details.

S7 Pole no.	ON	OFF
1	P20	LB_LDA0/MII_RX_CLK
2	P21	LB_LDA1/MII_MRXD0
3	P22	LB_LDA2/MII_MRXD1
4	P23	LB_LDA3/MII_MRXD2
5	P24	LB_LDA4/MII_MRXD3
6	P25	LB_LDA5/MII_RX_DV
7	P26	LB_LDA6/MII_CRS
8	P27	LB_LDA7/MII_RX_ER

5-2-7. S8 Switch Setting

The S8 switch is used to control the SPI, 1-Wire, CAN, UART 1 or Local Bus control signals multi-function pins. The S8 switch H/W setting should match the Multi-function Pin Setting 1 (offset 03h) of AX11015 I2C EEPROM. Please refer to AX11015/AX11025 128-pin development board schematics and Section 3.1.3 of AX11015/AX11025 datasheets for details.

S8 Pole no.	ON	OFF
1	SPI_SS1	SS1
2	SPI_SS2	SS2
3	1WIRE_DQ	SS1
4	1WIRE_STPZ	SS2
5	CAN_RX	SS1
6	CAN_TX	SS2
7	UART1_RXD	LB_CS0_N
8	UART1_TXD	LB_CS1_N

Note: The (Pole #1, #2), (Pole #3, #4) and (Pole #5, #6) could not be set to ON at the same time because the SPI_SS1 and SPI_SS2 pins are shared with 1WIRE_DQ/CAN_RX and 1WIRE_STPZ/CAN_TX pins respectively.

The following are some examples,

(1) To enable SPI_SS1 and SPI_SS2 interfaces,

S8 Pole no.	Pin Name	Setting
1	SPI_SS1	ON
2	SPI_SS2	ON
3	1WIRE_DQ	OFF
4	1WIRE_STPZ	OFF
5	CAN_RX	OFF
6	CAN_TX	OFF

(2) To enable 1-Wire interface,

S8 Pole no.	Pin Name	Setting
1	SPI_SS1	OFF
2	SPI_SS2	OFF
3	1WIRE_DQ	ON
4	1WIRE_STPZ	ON
5	CAN_RX	OFF
6	CAN_TX	OFF

(3) To enable CAN interface (for AX11025 128-pin development board only),

S8 Pole no.	Pin Name	Setting
1	SPI_SS1	OFF
2	SPI_SS2	OFF
3	1WIRE_DQ	OFF
4	1WIRE_STPZ	OFF
5	CAN_RX	ON
6	CAN_TX	ON

Note: If users want to enable the CAN interface of AX11025 128-pin development board, the S8 switch should be set as above configuration to enable the CAN interface and the Multi-function Pin Setting 1 (offset 0x03) of AX11025 EEPROM should be set to 0xC0.

5-2-8. S9 Switch Setting

The S9 switch is used to pull up/down the signals of the Port 2 GPIO pins. Please refer to AX11015/AX11025 128-pin development board schematics for details.

S9 Pole no.	Pin Name	ON	OFF
1	P20	Pull Down	Pull Up
2	P21	Pull Down	Pull Up
3	P22	Pull Down	Pull Up
4	P23	Pull Down	Pull Up
5	P24	Pull Down	Pull Up
6	P25	Pull Down	Pull Up
7	P26	Pull Down	Pull Up
8	P27	Pull Down	Pull Up

5-2-9. S10 Switch Setting

The S10 switch is used to control the UART2 and Local Bus multi-function pins. The S10 switch H/W setting should match the Multi-function Pin Setting 0 (offset 02h) of AX11015 I2C EEPROM. Please refer to AX11015/AX11025 128-pin development board schematics and Section 3.1.3 of AX11015/AX11025 datasheets for details.

S10 Pole no.	ON	OFF
1	UART2_RXD2	LB_LA8
2	UART2_TXD2	LB_LA9
3	UART2_CTS	LB_LA10
4	UART2_DSR	LB_LA11
5	UART2_RI	LB_LA12
6	UART2_DCD	LB_LA13
7	UART2_RTS	LB_LA14
8	UART2_DTR	LB_LA15

Note: The pins of AX110xx UART2/RS-485/GPIO Port 0 interfaces are shared each other so the S10/S11/S12 DIP switches cannot be turned ON at the same time. For example, you should turn ON the S10 DIP switch and turn OFF both S11 and S12 DIP switches to enable the UART2 interface or turn ON the S12 DIP switch and turn OFF both S10 and S11 DIP switches to enable the RS-485 interface.

5-2-10. S11 Switch Setting

The S11 switch is used to control the Port 0 GPIO and Local Bus multi-function pins. The S11 switch H/W setting should match the Multi-function Pin Setting 0 (offset 02h) of AX11015 I2C EEPROM. Please refer to AX11015/AX11025 128-pin development board schematics and Section 3.1.3 of AX11015/AX11025 datasheets for details.

S11 Pole no.	ON	OFF
1	P00	LB_LA8
2	P01	LB_LA9
3	P02	LB_LA10
4	P03	LB_LA11
5	P04	LB_LA12
6	P05	LB_LA13
7	P06	LB_LA14
8	P07	LB_LA15

Note: See the note message in Section 5-2-9

5-2-11. S12 Switch Setting

The S12 switch is used to control the RS485 and Local Bus multi-function pins. The S12 switch H/W setting should match the Multi-function Pin Setting 0 (offset 02h) of AX11015 I2C EEPROM. Please refer to AX11015/AX11025 128-pin development board schematics and Section 3.1.3 of AX11015/AX11025 datasheets for details.

S12 Pole no.	ON	OFF
1	RS485_RXD2	LB_LA8
2	RS485_TXD2	LB_LA9
3	RS485_DE	LB_LA2
4	RS485_RE_N	LB_LA3

Note: See the note message in Section 5-2-9

5-2-12. S14 Switch Setting

The S14 switch is used to set the UART0/1 interface to UART 0 or UART 1. Users can set S14 switch to UART 0 for UART console debugging or UART Flash programming purposes or set to UART 1 for customer specific purpose.

S14 UART Interface	Comments
UART 0	For debugging console or Flash programming
UART 1	For customer specific purpose

5-2-13. S15 Switch Setting

The S15 switch is used to configure some general configurations for AX11015/AX11025 128-pin Development Board. Please refer to AX11015/AX11025 128-pin development board schematics for details.

The Pole #6 (Burn Flash Enable) of S15 switch should be set to ON for UART Flash programming and should be set to OFF for Ethernet Boot Loader Flash programming and normal operation. The Pole #7 (Burn Flash 921K) of S15 switch should be set to OFF for normal UART interface and can be set to ON if the user's machine supports 921KB high speed UART mode.

S15 Pole no.	Function	ON	OFF
2-1	System Clock Select	00: 25MHz; 01: 50MHz; 10: Reserved; 11: 100MHz	
3	Ext. SRAM Enable	Enable	Disable
4	Local Bus Mode	Slave	Master
5	Bus Mode	Sync. Bus Mode	Async. Bus Mode
6	Burn Flash Enable	Enable UART Flash Programming function	Disable UART Flash Programming function
7	Burn Flash 921K	High Speed UART mode	UART mode (115200bps)
8	I2C Boot Disable	Disable I2C Boot	Enable I2C Boot

6. I2C EEPROM Programming

AX110xx development boards provide two I2C EEPROM programming firmware codes for users to modify the EEPROM content. One is the ax110xx.bin file in the "I2C EEPROM\Build" subdirectory of AX110xx Development Kit CD for the UART Flash programming method; another is the eeprom.bin file in the "I2C EEPROM\Build_RT" subdirectory of AX110xx Development Kit CD for the Ethernet Boot Loader Flash programming method. Please refer to Section 7 "Flash Programming" for more details about how to program the I2C EEPROM programming utility into the AX110xx Flash memory.

Note: The AX110xx I2C EEPROM Programming firmware is an engineering development tool and will override the existent firmware code in the Flash of AX110xx board. So please make sure you can restore the original firmware before you start uploading the I2C EEPROM Programming firmware into the Flash of AX110xx board.

6-1. Power ON the AX110xx development board after the I2C EEPROM Programming firmware was programmed successfully.

6-2. If the AX110xx I2C EEPROM Programming firmware is programmed into the Flash of AX110xx development board successfully. The following I2C EEPROM Programming console screen will be appeared in the HyperTerminal window.

```
ASIX AX110xx I2C EEPROM Utility V1.01 (07/14/06 16:06:33)

AX110xx Station: general commands:
  24c16bwr   - write a byte data to an 24c16 EEPROM memory address
  24c16brd   - read a byte data from an 24c16 EEPROM memory address
  24c16pwr   - page write 24c16 EEPROM memory
  24c16dump  - Dump 24c16 EEPROM memory
  fwupdate   - Firmware Update
  help       - dispaly help messages for menus

ASIX>
```

6-3. Run "24c16dump 0 0 30" command in the I2C EEPROM Programming console to display the current AX110xx EEPROM content.

Note: The default I2C address (i.e. i2caddr) of the I2C EEPROM is 0.

```
ASIX>24c16dump 0 0 30
Dump Address : 00
Dump Data :
21 bc 0f 00 30 00 01 00 00 00 00 00 f2 05 10 e0
1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
ff 04 ff ff ff ff ff ff ff ff ff ff ff ff ff ff

ASIX>
```

6-4. Run below "24c16pwr" commands in the I2C EEPROM Programming console to program the whole AX110xx EEPROM from offset 00h to offset 21h.

```
24c16pwr 0 0   21 BC 0F 00 30 00 01 00   ==> Write EEPROM offset address 0x00~0x07
24c16pwr 0 8   00 00 00 00 F2 05 10 E0   ==> Write EEPROM offset address 0x08~0x0F
24c16pwr 0 10 1D 19 87 00 FF FF FF FF  ==> Write EEPROM offset address 0x10~0x17
24c16pwr 0 18 10 03 00 A8 C0 00 FF FF  ==> Write EEPROM offset address 0x18~0x1F
24c16pwr 0 20 FF 04 FF FF FF FF FF FF  ==> Write EEPROM offset address 0x20~0x27
```

```
ASIX>24c16pwr 0 0   21 BC 0F 00 30 00 01 00
Write Address : 0
Write Data : 21 bc 0f 00 30 00 01 00

ASIX>24c16pwr 0 8   00 00 00 00 F2 05 10 E0
Write Address : 8
Write Data : 00 00 00 00 f2 05 10 e0

ASIX>24c16pwr 0 10 1D 19 87 00 FF FF FF FF
Write Address : 10
Write Data : 1d 19 87 00 ff ff ff ff

ASIX>24c16pwr 0 18 10 03 00 A8 C0 00 FF FF
Write Address : 18
Write Data : 10 03 00 a8 c0 00 ff ff

ASIX>24c16pwr 0 20 FF 04 FF FF FF FF FF FF
Write Address : 20
Write Data : ff 04 ff ff ff ff ff ff

ASIX>24c16dump 0 0 30
Dump Address : 00
Dump Data :
21 bc 0f 00 30 00 01 00 00 00 00 00 f2 05 10 e0
1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
ff 04 ff ff ff ff ff ff ff ff ff ff ff ff ff ff

ASIX>
```

6-5. Run the "24c16bwr" command in the I2C EEPROM Programming console to modify AX110xx EEPROM content.

> **Note:** Please refer to Section 3.1 of AX110xx datasheet for the detailed EEPROM data format and refer to Appendix B/C/D of this document for the I2C EEPROM default settings of AX110xx development boards.

```
ASIX>24c16dump 0 0 30
Dump Address : 00
Dump Data :
21 bc 0f 00 30 00 01 00 00 00 00 00 f2 05 10 e0
1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
ff 04 ff ff ff ff ff ff ff ff ff ff ff ff ff ff

ASIX>24c16bwr 0 1 3c              ==> Modify EEPROM to enable debug mode for DoCD HAD
Write Address : 01 ; Data : 3c

ASIX>24c16pwr 0 6 9a 78           ==> Modify the MAC address 02 12 34 56 78 9A (Node ID 0~1)
Write Address : 6
Write Data : 9a 78

ASIX>24c16pwr 0 8 56 34 12 02     ==> Modify the MAC address 02 12 34 56 78 9A (Node ID 2~5)
Write Address : 8
Write Data : 56 34 12 02

ASIX>24c16dump 0 0 30
Dump Address : 00
Dump Data :
21 3c 0f 00 30 00 9a 78 56 34 12 02 f2 05 10 e0
1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
ff 04 ff ff ff ff ff ff ff ff ff ff ff ff ff ff

ASIX>
```

6-6. Run "help" or "?" command in the I2C EEPROM Programming console to display the help messages and run the I2C EEPROM command without parameter to display the command syntax.

```
ASIX>help

ASIX AX110xx I2C EEPROM Utility V1.01 (07/14/06 16:06:33)

AX110xx Station: general commands:
  24c16bwr   - write a byte data to an 24c16 EEPROM memory address
  24c16brd   - read a byte data from an 24c16 EEPROM memory address
  24c16pwr   - page write 24c16 EEPROM memory
  24c16dump  - Dump 24c16 EEPROM memory
```

```
  fwupdate   - Firmware Update
  help       - dispaly help messages for menus

Also try 'help [general]'

ASIX>24c16bwr
Usage: 24c16bwr i2caddr addr data
  i2caddr : device address
  addr    : memory address
  data    : memory data

ASIX>24c16brd
Usage: 24c16brd i2caddr addr
  i2caddr : device address
  addr    : memory address

ASIX>24c16pwr
Usage: 24c16pwr i2caddr addr data
  i2caddr : device address
  addr    : memory address
  data    : memory data

ASIX>24c16dump
Usage: 24c16dump i2caddr addr len
  i2caddr : device address
  addr    : memory address
  len     : memory length

ASIX>
```

6-7. After the AX110xx I2C EEPROM was modified successfully, users can install the TFTP/DHCP server on the network and then run the "fwupdate" command in the I2C EEPROM Programming console to restore the original AX110xx development board firmware code via the Ethernet Boot Loader Flash Programming method.

6-8. Power ON the AX110xx development board to take effect the new EEPROM settings.

7. Flash Programming

AX110xx development board provides three Flash programming solutions, two of them are via the COM Port interface; the third one is via the Ethernet Boot Loader.

7-1. COM Port Flash Programming Method Under Windows

ASIX Electronics provides a Windows In-System Programming (ISP) tool for customers to program the Flash under Windows environment. The Windows ISP tool is a Windows dialog-based software program that can be run on Windows 7/Vista/XP/2000 machine.

7-1-1. Environment Setup

Before using the Windows ISP tool, following tasks have to be completed.

1. Copy the Windows ISP tool onto a PC running Windows 32-bit or 64-bit system.
2. Set AX110xx development board to UART 0 interface. (Refer to Section 5-1-7, 5-2-12 and 7-2 for details)
3. Set AX110xx development board to enable Flash Programming mode. (Refer to Section 5-1-5, 5-2-13 and 7-2 for details)
4. Connect the UART 0 interface of AX110xx development board to the COMx port of Windows PC via a RS-232 NULL modem cable.

7-1-2. Starting Windows ISP Tool

Click the executable file "AX110xx_ISP.exe" on Windows 32-bit system or "AX110xx_ISP_64.exe" on Windows 64-bit system to start the Windows ISP tool; the main window appears as shown below.

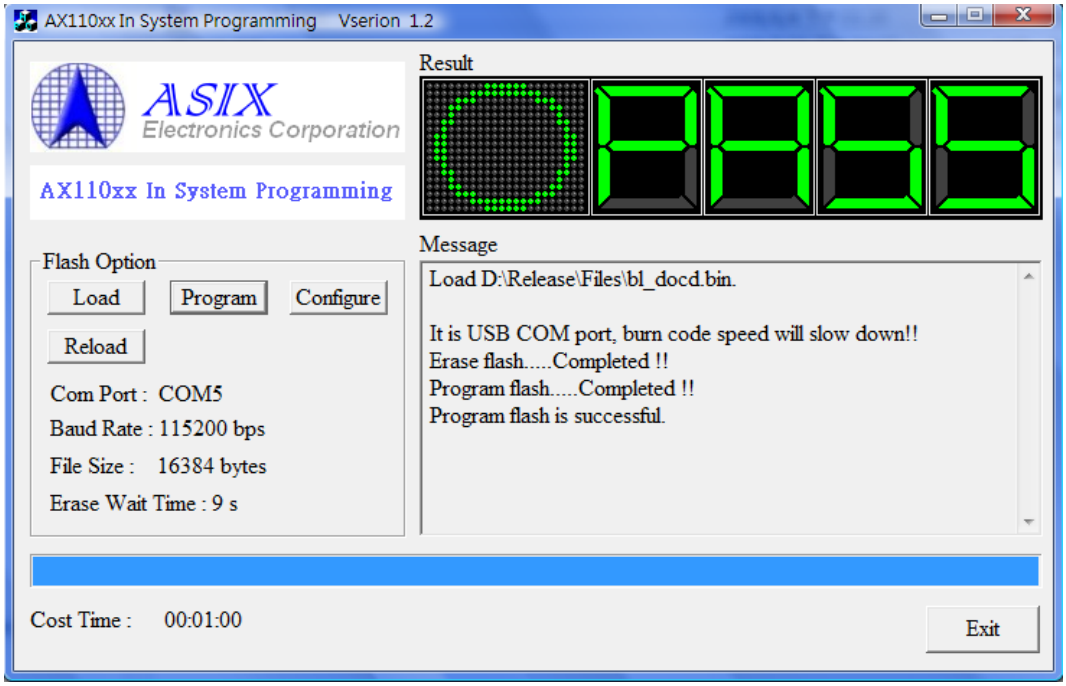


Figure 12. The Main Window of Windows ISP Tool

There are four buttons on the main window,

1. **Load:** click this button to select the binary file for programming.

2. **Reload:** click this button to load the binary file again for programming.

3. **Program:** click this button to program the Flash memory. The ISP program will write the content of binary file into the Flash memory. This button is disabled if no binary file is selected.

4. **Configure:** click this button to select the COM port, baud rate and erase waiting time. The ISP program supports baud rates 921600 bps and 115200 bps.

5. **Exit:** click this button to close the ISP program.

The result of programming operation will be displayed in the *Result* area. Helpful messages will be displayed accordingly in the *Message* area.

The Flash programming speed of AX110xx Windows ISP tool is about 30 seconds for 16K bytes code size via the on-board COM port and about 1 minute for 16K bytes code size via the USB-to-Serial COM port.

7-1-3. Example Procedure

An example for using the Windows ISP tool is provided. Here the example shows how to use the Windows ISP tool to download a bootloader into the Flash of an AX110xx development board.

1. Refer to **Section 7-1-1: Environment Setup** to setup the environment.
2. Run the Windows ISP tool.
3. Use the default settings or click the [Configure] button to change settings.
4. Click the [Load] button to select the binary file of bootloader (e.g. bl_docd.bin).
5. Click the [Program] button to write the bootloader into the Flash memory.
6. After completing the programming operation, click the [Exit] button to close the program.

Note: AX110xx_ISP.EXE for Windows 32-bit systems and AX110xx_ISP_64.EXE for Windows 64-bit systems.

7-2. COM Port Flash Programming Method Under MS-DOS

ASIX Electronics provides two Flash programming utilities "UARTH.EXE" for High Speed UART interface (921Kbps) and "UARTL.EXE" for UART interface (115200bps) to program the Flash of AX110xx development board via the UART 0 interface.

Note: The High Speed UART Flash programming utility (UARTH.EXE) can only be run with the PC machine that supports the high speed UART mode (i.e. the baud rate up to 921Kbps). If your machine only supports the standard UART mode (i.e. the baud rate up to 115200bps), you should run the "UARTL.EXE" utility to program the AX110xx Flash.

Figure 13. AX1100x 80-pin Development Board UART 0 Interface

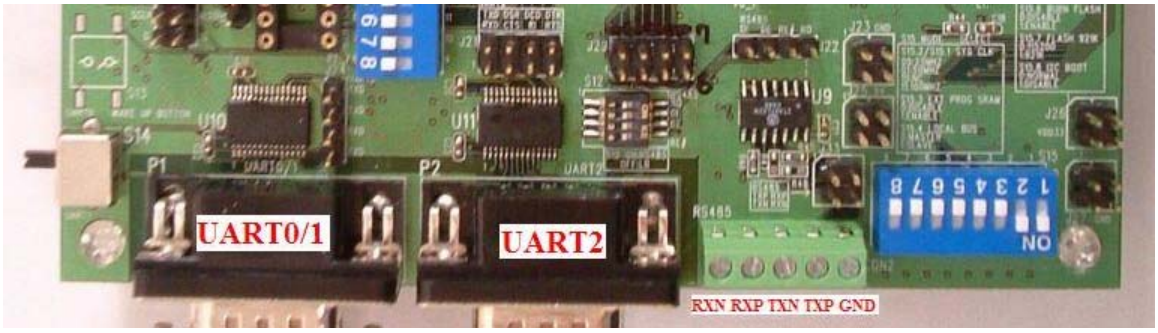


Figure 14. AX11015/AX11025 128-pin Development Board UART 0 Interface

1. Set AX110xx development board to UART 0 interface. (Refer to Section 5-1-7 and 5-2-12 for details)
2. Set AX110xx development board to enable Flash Programming mode. (Refer to Section 5-1-5 and 5-2-13 for details)
3. Connect the UART 0 interface of AX110xx development board to the COMx port of a MS-DOS or Windows PC via a RS-232 NULL modem cable.
4. Boot up the MS-DOS or Windows PC and copy the "UARTH.EXE" or "UARTL.EXE" and the AX110xx development board firmware code (e.g. filename.bin) files into the same directory.
5. Power ON the AX110xx development board.

Note: Don't disconnect the RS-232 NULL modem cable after AX110xx development board was powered ON; otherwise, the UART Flash programming operation might fail. If the RS-232 NULL modem cable is disconnected after AX110xx development board

was powered ON, you must reset the AX110xx development board again before running the UART Flash programming utility.

6. Open a DOS Prompt window from the Windows PC. (Only for Windows PC)
7. Change the current directory into the subdirectory where the "UARTH.EXE" or "UARTL.EXE" and the firmware code are copied.
8. Run the "UARTH" or "UARTL" command in the DOS Prompt console.

```
Address assigned to COM1 is 3F8h
ASIX Flash Progrom Utility V3.1 (2006-05-03) 115200bps
 scp x            Set a COM port to use <x:COM port number 1-4>
 rat              Read Access Time
 rit              Read Interval Time
 wat xx           Write Access Time <xx:Access Time Value>
 wit xx           Write Interval Time <xx:Interval Time Value>
 ce1              Chip Erase for Embedded Flash
 fp1 xxx.bin      File Programming for Embedded Flash <xxx.bin:filename>
 cea              Chip Erase with an Expanding Flash
 fpa xxx.bin      File Programming with an Expanding Flash
 quit             Quit program
 help             Help message
COM Port Address assigned to 3F8h
UART>
```

Figure 15. Flash Programming Utility Command Lines

9. Run "scp x" command to set a proper COM port (COMx) in the Flash programming console.
10. Run "rat" command to make sure if the COM port connection is established properly. If the connection is established, the return code will be a number; otherwise, an error message will be displayed in the Flash programming console.
11. If the COM port connection is established successfully, run "fp1 filename.bin" command to start AX110xx firmware code programming operation. The Flash programming utility might take a long time to complete the whole Flash programming operation.

UART Mode	Baud Rate	Command file	Operating System	Flash Programming Time
HS UART	921Kbps	UARTH.EXE	MS-DOS 6.22	About 30 secs/100Kbytes
HS UART	921Kbps	UARTH.EXE	Windows XP/2K	Longer than 5 mins/100Kbytes
UART	115200bps	UARTL.EXE	MS-DOS 6.22	About 4 mins/100Kbytes
UART	115200bps	UARTL.EXE	Windows XP/2K	Longer than 20 mins/100Kbytes

Figure 16. Flash Programming Time

Note: ASIX Electronics suggests running the Flash programming utility on MS-DOS platform for a better performance. To run the Flash programming utility on Windows platform might take a much longer time than the expected time.

12. After the Flash programming is completed, power OFF AX110xx development board and set AX110xx development board to disable Flash Programming mode. (Refer to Section 5-1-5 and 5-2-13 for details)
13. Power ON AX110xx development board to take effect the new firmware code.

7-3. Ethernet Boot Loader Flash Programming Method

In addition to the COM Port Flash programming method, AX110xx also supports a faster way to program the Flash memory of AX110xx development board by AX110xx Ethernet Boot Loader. To do so, the Ethernet Boot Loader code should be first programmed into the Flash memory of AX110xx development board via the COM Port interface before using the Ethernet Boot Loader Flash programming method. Users need to setup the DHCP and TFTP servers to provide the firmware code download function and select the "3 Download new runtime code via Ethernet" or "4 Download new bootloader code via Ethernet" function from HyperTerminal console to start programming AX110xx runtime code or Ethernet Boot Loader code to the Flash memory of AX110xx board.

AX110xx Ethernet Boot Loader will send out a BOOTP packet to request an IP address from the DHCP server. After getting an IP address from DHCP server, the Ethernet Boot Loader will start to download the firmware code from the TFTP server and will auto-restart AX110xx to load the new firmware code.

The following are the detailed procedures about how to program a new AX110xx runtime code to AX110xx Flash via an existent Ethernet Boot Loader.

1. Connect the UART 0 interface of AX110xx development board to the COMx port of Windows machine via a RS-232 NULL modem cable.
2. Connect the RJ-45 port of AX110xx development board and the Ethernet port of the Windows machine with a RJ-45 Ethernet cable.
3. Boot up the Windows machine.
4. Configure the IP address of Windows machine to **xxx.xxx.xxx.yyy** (e.g. 192.168.0.190).

Note: The pool IP starting address of DHCP server should be set to the same segment as the IP address of Windows machine like **xxx.xxx.xxx.zzz** (e.g. 192.168.0.3).

5. Run Hyper Terminal application to create a connection between AX110xx board and the Windows machine, the COMx port should be set to 9600 baud rate, 8 data bit, NO parity check, 1 stop bit and NO flow control.

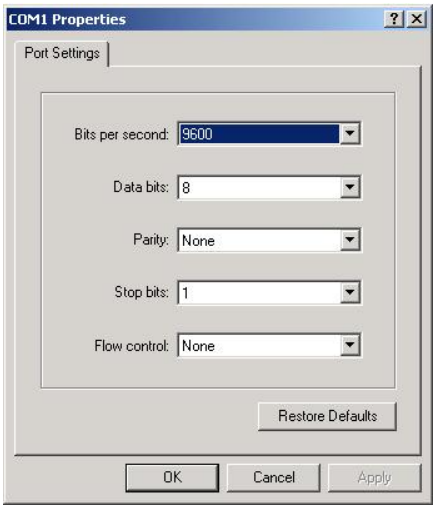


Figure 17. The COM Port Setting for AX110xx UART0 Debug Console

6. Startup the TFTP Server by running “tftpd32.exe” (a free TFTP Server application) on the Windows machine.
7. Configure a proper IP address and Boot File download path on TFTP Server as below pictures.

Note: The following are the TFTP/DHCP server (TFTPD32.EXE) setting procedures for your reference:
1. Set the “IP pool starting address“ field (e.g. 192.168.0.3) to the same subnet mask with the “Server interfaces IP address” (e.g. 192.168.0.190).
2. Set the “Size of pool” field to a number larger than 0 (e.g. 10).
3. Select a proper “Current Directory” (e.g. D:\temp) by pressing the [Browser] button.
4. Set the “Boot File” field (e.g.128-pin.bin) to the correct AX110xx firmware code allocated at the “Current Directory”.
5. Press the [Save] button to take effect the new setting.

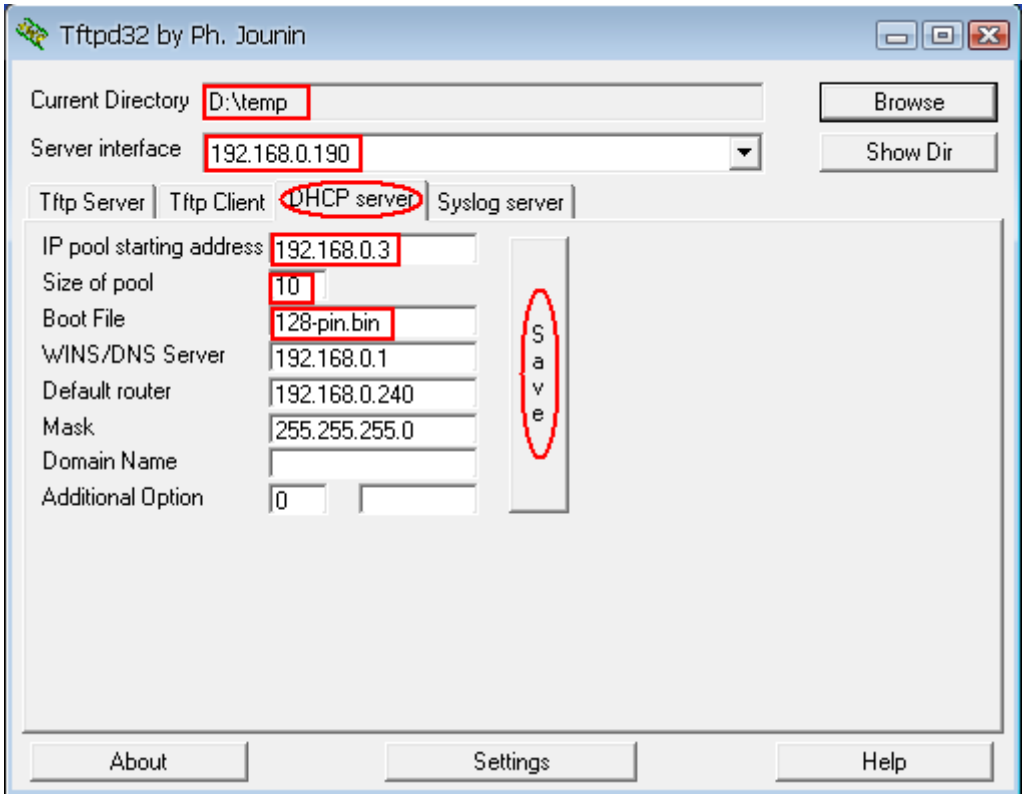


Figure 18. The TFTPD32 TFTP/DHCP Server Setting

Power ON AX110xx development board. The following AX110xx Ethernet Boot Loader messages should be displayed on the HyperTerminal console.

```
AX110xx bootloader v3.1.0 version for DoCD HAD Debugger

1. Execute Runtime.
2. Download runtime via RS232.
3. Download runtime via Ethernet.
4. Download new bootloader via Ethernet.

Please input 1,2,3 or 4 to execute above :
```

Note: The AX110xx Ethernet boot loader for DoCD HAD2 debugger (with 4 options in the boot loader menu) supports the Ethernet boot loader upgrade function and the source tracking function through DoCD HAD2 debugger; the AX110xx Ethernet boot loader for Flash Read Protection (with 3 options in the boot loader menu) disallows the Ethernet boot loader upgrade function and the source tracking function through DoCD HAD2 debugger to protect the customized AX110xx firmware from an illegal firmware copy by users.

8. Enter "3" and press ENTER key to start downloading a new AX110xx runtime code from TFTP Server via the Ethernet Boot Loader. After the new AX110xx runtime code was programmed successfully, AX110xx development board will be auto-rebooted by running the new AX110xx runtime code.

   For example, the following are the TFTPD32.EXE logo messages and the AX11015 128-pin demo firmware (i.e. 128-pin.bin) boot up messages displayed on the HyperTerminal console for your reference.

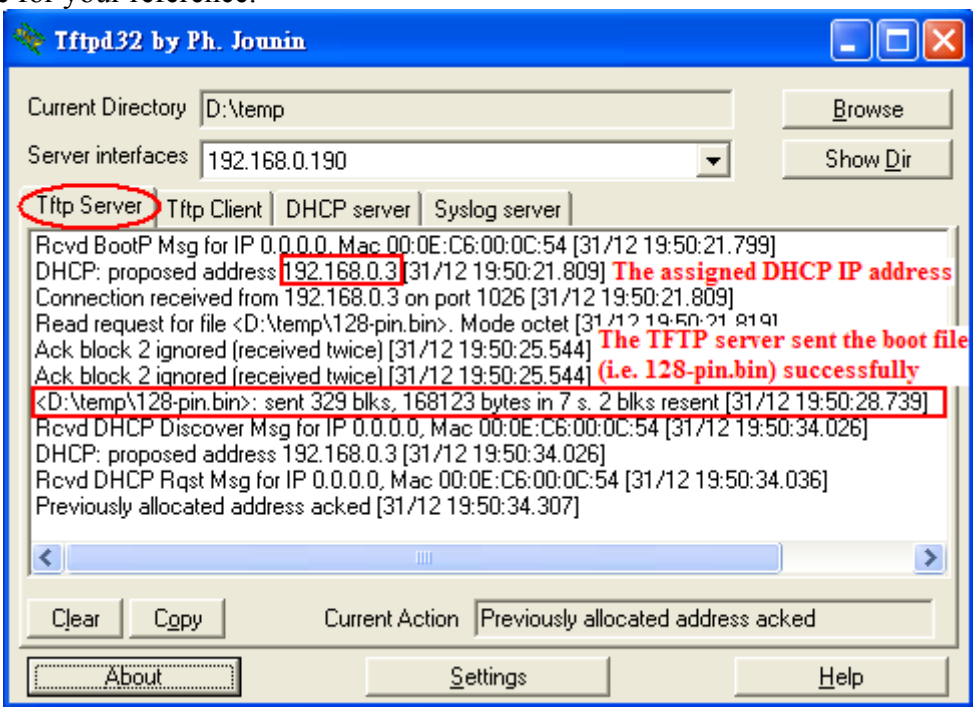


Figure 19. The TFTPD32 TFTP/DHCP Server Log Messages

Note: Please check the TFTPD32 log messages to make sure if the TFTPD32 utility assigned a DHCP IP address (e.g. 192.168.0.3) and sent the boot file (e.g. 128-pin.bin) to AX110xx board successfully.

```
AX110xx bootloader v3.1.0 version for DoCD HAD Debugger

1. Execute Runtime.
2. Download runtime via RS232.
3. Download runtime via Ethernet.
4. Download new bootloader via Ethernet.

Please input 1,2,3 or 4 to execute above :

Get new IP : c0a8000a
Get subnet mask : ffffff00
Get gateway : c0a80001
TFTP server IP : c0a80032
Download file total length = 0x2c0cf bytes.
download ok.

Wait runtime code.
UART-0 init ok.
UART-2 init ok.

ASIX AX110xx 128PIN Demo Firmware V3.0.0 (06/07/11 16:39:53)
DHCP init ok.
GCONFIG_Init()...
gconfig_ReStoreParameter()...
  1.Header: ff ff ff ff ff ff ff ff
  2.GID: ff ff ff ff ff ff ff ff
  3.Opcode | Reserved: ff ff
  4.Device Name: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
  5.MAC Addr: ff ff ff ff ff ff
  6.Network Setting: ff ff SW:Cli DHCP:On UDP:On TCP:On MC:On BC:On
  7.D-IP | S-IP | U-M-B Port: 255 255 255 255 255 255 255 255 4294967295
4294967295 4294967295
  8.Dest IP | Dest Port: 255 255 255 255
  9.Netmask | Gateway | DNS: 255 255 255 255 255 255 255 255 255 255 255 255
  10.Serial Port Setting:
  11.Status: ff ff
  12.EEPROM DATA: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
                  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
                  ff ff
gconfig_ReStoreParameter() failed! (Wrong checksum)
Read EEProm data...
GCONFIG_ReadConfigData()...
```

```
  1.Header: ff ff ff ff ff ff ff ff
  2.GID: ff ff ff ff ff ff ff ff
  3.Opcode | Reserved: ff ff
  4.Device Name: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
  5.MAC Addr: ff ff ff ff ff ff
  6.Network Setting: ff ff SW:Cli DHCP:On UDP:On TCP:On MC:On BC:On
  7.D-IP | S-IP | U-M-B Port: 255 255 255 255 255 255 255 255 4294967295
4294967295 4294967295
  8.Dest IP | Dest Port: 255 255 255 255
  9.Netmask | Gateway | DNS: 255 255 255 255 255 255 255 255 255 255 255 255
  10.Serial Port Setting:
  11.Status: ff ff
  12.EEPROM DATA: 21 bc 00 80 30 00 cc 0c 00 c6 0e 00 f2 05 10 e0
                  1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
                  ff 04
ConfigData error
default setting 840GCONFIG_ReadDefaultConfigData()...
  1.Header: 00 00 00 00 00 00 00 00
  2.GID: 41 53 49 58 58 49 53 41
  3.Opcode | Reserved: ff 00
  4.Device Name: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  5.MAC Addr: ff ff ff ff ff ff
  6.Network Setting: 4d 00 SW:Ser DHCP:On UDP:On TCP:On MC:Off BC:On
  7.D-IP | S-IP | U-M-B Port: 255 255 255 255 192 168 0 3 25000 25100 25122
  8.Dest IP | Dest Port: 192 168 0 2
  9.Netmask | Gateway | DNS: 255 255 255 0 192 168 0 1 192 168 0 1
  10.Serial Port Setting: BR:115200 DB:8 P:None FC:None SB:1
  11.Status: 00 00
  12.EEPROM DATA: 21 bc 00 80 30 00 cc 0c 00 c6 0e 00 f2 05 10 e0
                  1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
                  ff 04
GCONFIG_WriteConfigData()...
gconfig_StoreParameter()...
  1.Header: 00 00 00 00 00 00 00 00
  2.GID: 41 53 49 58 58 49 53 41
  3.Opcode | Reserved: ff 00
  4.Device Name: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  5.MAC Addr: ff ff ff ff ff ff
  6.Network Setting: 4d 00 SW:Ser DHCP:On UDP:On TCP:On MC:Off BC:On
  7.D-IP | S-IP | U-M-B Port: 255 255 255 255 192 168 0 3 25000 25100 25122
  8.Dest IP | Dest Port: 192 168 0 2
  9.Netmask | Gateway | DNS: 255 255 255 0 192 168 0 1 192 168 0 1
  10.Serial Port Setting: BR:115200 DB:8 P:None FC:None SB:1
  11.Status: 00 00
  12.EEPROM DATA: 21 bc 00 80 30 00 cc 0c 00 c6 0e 00 f2 05 10 e0
                  1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
                  ff 04
ServerBroadcastListenPort = 25122
```

```
GUDPBC_Init()...
DHCP request... IP: 192.168.0.10
GCONFIG_WriteConfigData()...
gconfig_StoreParameter()...
  1.Header: 1f 00 00 00 00 00 00 00
  2.GID: 41 53 49 58 58 49 53 41
  3.Opcode | Reserved: ff 00
  4.Device Name: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  5.MAC Addr: 00 0e c6 00 0c cc
  6.Network Setting: 4d 00 SW:Ser DHCP:On UDP:On TCP:On MC:Off BC:On
  7.D-IP | S-IP | U-M-B Port: 192 168 0 10 192 168 0 3 25000 25100 25122
  8.Dest IP | Dest Port: 192 168 0 2
  9.Netmask | Gateway | DNS: 255 255 255 0 192 168 0 1 192 168 0 1
  10.Serial Port Setting: BR:115200 DB:8 P:None FC:None SB:1
  11.Status: 00 00
  12.EEPROM DATA: 21 bc 00 80 30 00 cc 0c 00 c6 0e 00 f2 05 10 e0
                  1d 19 87 00 ff ff ff ff 10 03 00 a8 c0 00 ff ff
                  ff 04
GUDPBC_Init()...
```

9. If you don't see any error messages on the HyperTerminal console, the new AX110xx runtime code is programmed successfully now.

8. Software Sample codes

8-1. AX110xx Ethernet Boot Loader

8-1-1. AX110xx Ethernet Boot Loader Description

ASIX Electronics provides two kinds of Flash programming solution, one is via AX110xx UART 0 interface; another one is via AX110xx Ethernet Boot Loader interface. The Ethernet Boot Loader Flash Programming method is far faster than the UART Flash Programming method. The Boot Loader code is programmed in the Flash of AX110xx development board by default. Users need to setup DHCP and TFTP servers to provide the firmware code download function and select the "3 Download new runtime code via Ethernet" or "4 Download new bootloader code via Ethernet" function from AX110xx UART 0 console to start programming the Flash of AX110xx development board. Please refer to Section 7-3 **"Ethernet Boot Loader Flash Programming Method"** for details.

AX110xx Ethernet Boot Loader will send out a BOOTP packet to request an IP address from the DHCP server. After getting an IP address from DHCP server, the Ethernet Boot Loader will start to download the firmware code from the TFTP server and will auto-restart AX110xx to load the new firmware code.

8-1-2. How to program AX110xx Ethernet Boot Loader code?

If users want to program the new Ethernet Boot Loader code via an existent Ethernet Boot Loader interface, you need to setup DHCP and TFTP servers to provide the firmware code download function and select the "4 Download new bootloader code via Ethernet" function from AX110xx UART 0 console to start programming the Flash of AX110xx development board.

AX110xx Ethernet Boot Loader will send out a BOOTP packet to request an IP address from the DHCP server. After getting an IP address from DHCP server, the Ethernet Boot Loader will start to download the firmware code from the TFTP server and will auto-restart AX110xx to load the new firmware code. Please refer to Section 7-3 **"Ethernet Boot Loader Flash Programming Method"** for details.

8-1-3. How to verify AX110xx Ethernet Boot Loader code?

Please repeat the procedures of the "Ethernet Boot Loader Flash Programming Method" section to verify the new Ethernet Boot Loader code. If you can see the AX110xx boot loader menu and don't see any error messages, the new Ethernet Boot Loader code works fine now.

8-2. Peripheral Software Modules

AX110xx Development Kit provides some software module sample codes like CPU, Ethernet, S/W DMA, MS Timer, Local Bus, I2C, SPI, 1-Wire, PCA, UART2, and 8051 standard modules (like UART, Timer). Please refer to the software module sections of **"AX110xx Software User Guide"** for details.

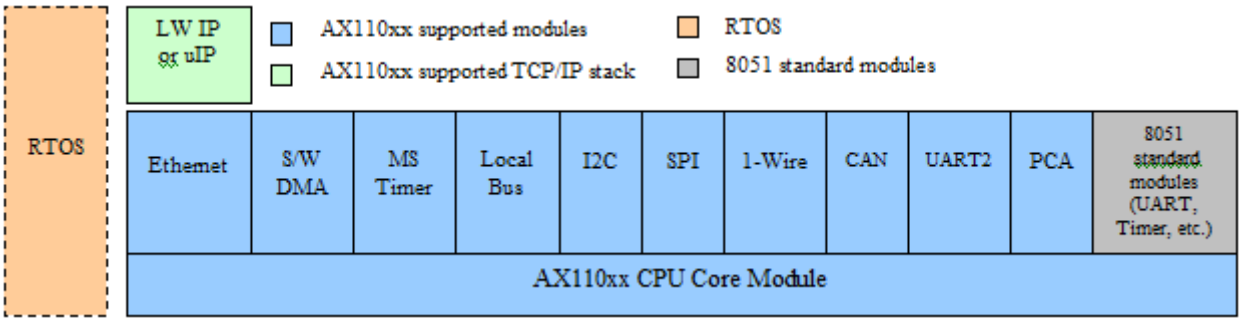


Figure 20. AX110xx Peripheral Software Modules

8-3. TCP/IP Stacks

AX110xx Development Kit provides two TCP/IP Stacks sample codes, one is the uIP TCP/IP Stack without OS; the other one is the Lightweight IP (lwIP) TCP/IP Stack without OS. The uIP is a TCP/IP Stack for 8-bit and 16-bit microcontrollers with very small code footprint and RAM requirements. The lwIP is a TCP/IP Stack with full-scale TCP functions supported. The uIP and lwIP are open source software that are originally developed by Adam Dunkels and are ported for Keil C51 by Murray R. Van Luyn.

The AX110xx uIP TCP/IP module has been modified slightly by ASIX Electronics for supporting functions of UDP packets, keep alive, don't fragment, PPPoE packets, and so on. In addition, although uIP itself supports the feature of IP fragment reassembly, it is not supported in the AX110xx uIP TCP/IP module.

We strongly suggest users to implement AX110xx firmware based on AX110xx development board demo firmware source codes with AX110xx TCP/IP module source code since the AX110xx TCP/IP module source code has been optimized and much more stable than the original uIP TCP/IP Stack source code.

Below table shows the features supported in lwIP, original uIP and AX110xx uIP TCP/IP module.

Feature	lwIP	Original uIP	AX110xx uIP TCP/IP Module
IP and TCP checksums	YES	YES	Support with hardware accelerator
IP fragment reassembly	YES	YES	NO
IP options	NO	NO	NO
Multiple interfaces	YES	NO	YES
UDP	YES	NO	YES
Multiple TCP connections	YES	YES	YES
TCP options	YES	YES	YES
Variable TCP MSS	YES	YES	YES
RTT estimation	YES	YES	YES
TCP flow control	YES	YES	YES
Sliding TCP window	YES	NO	NO
TCP congestion control	YES	No needed	No needed
Out-of-sequence TCP data	YES	NO	NO
TCP urgent data	YES	YES	YES
Data buffered for re-transmission	YES	NO	YES
TCP keep alive timer	YES	NO	YES

Figure 21. Features Supported in lwIP, Original uIP and AX110xx uIP TCP/IP Modules

8-4. Upper Protocol Modules

There are some protocol modules provided for AX110xx software developers to develop applications. These protocol modules are developed mainly for but not limited to applications of IP camera and RS232-toEthernet. The AX110xx protocol modules consist of eight application protocol modules and three network protocol modules. All modules are developed based on the AX110xx software driver modules. Please refer to "**AX110xx Software User Guide**" and "**AX110xx Upper Protocol Developer's Guide**" for details. Below diagram shows the architecture of AX110xx software protocol modules.

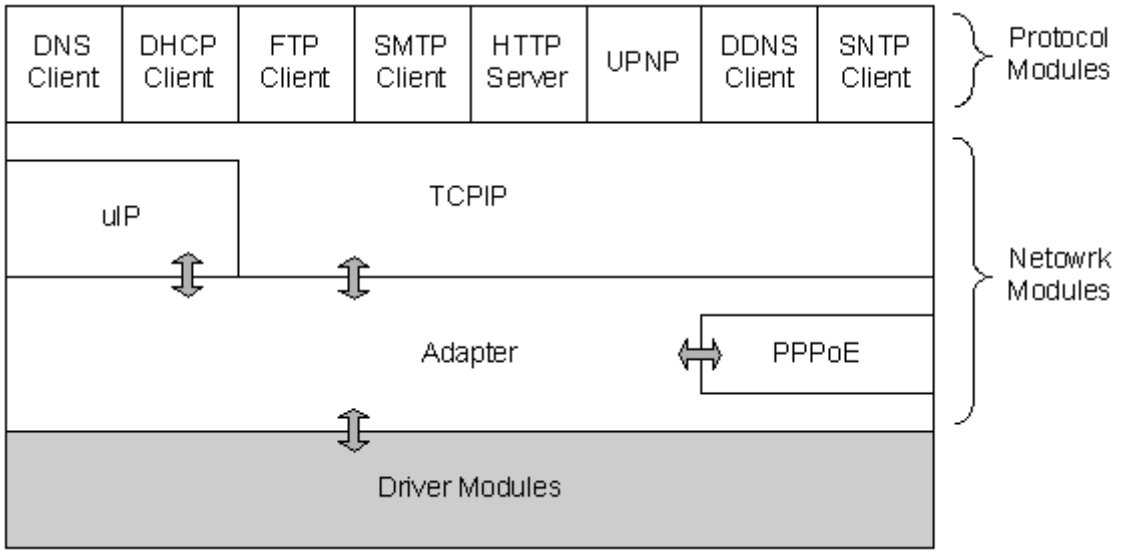


Figure 22. The Architecture of AX110xx Upper Protocol Modules

8-5. More sample codes

ASIX Electronics will provide more sample codes for customers' reference. Please contact ASIX sales (sales@asix.com.tw) for latest updated information.

9. Software Development Tools

9-1. Software Compiler Tool

All software modules for AX110xx family are developed in C language under Keil IDE development environment. Users can purchase the Keil IDE Development Tool from Keil's web site (http://www.keil.com/c51/selector.asp). In general, users need to purchase the PK51 development tool for C-language compiler, debugger and simulator. Those users only needing the compiler function and program code size less than 64K bytes, you can purchase the CA51 package. Users can also download the Keil C51 evaluation software from Keil's web site for free, but the evaluation software can only compile the sample codes with less than 2K bytes binary code. Please refer to Section 4 “How to compile AX110xx development board demo firmware” for detailed procedures about how to compile AX110xx demo firmware source code.

9-2. Software Debugger Tool

AX110xx provides two software debugging solutions, one is the UART console debugging; another one is the DCD's DoCD HAD2 debugger. All AX110xx software modules support the basic UART console debugging function by default. If the user needs more powerful debugging tool like source level debugging, AX110xx development board supports the Digital Core Design’s DoCD Hardware Debugger – the HAD2 module. Through the HAD2 module, the software running on AX110xx development board can be real-time debugged. The user may consider purchasing the HAD2 module from Digital Core Design and download the debugger software from Digital Core Design’s web site (http://www.dcd.pl/).

Figure 23. DoCD HAD2 Hardware Debugger Module

The DoCD HAD2 is a small hardware adapter that manages communication between the Debugger inside AX110xx and an USB port of the host PC running Debug Software. The USB communication interface to target host PC is at USB Full speed and its power supply comes directly from the USB port.

The DoCD HAD2 Debug Software is a Windows based application. It is fully compatible with all existing 8051/80390 C compilers and Assemblers. The Debug Software allows user to work in two major modes: software simulator mode and hardware debugger mode. Those two modes assure software validation in simulation mode and then real-time debugging of developed software inside AX110xx using debugger mode. Once loaded, the program may be observed in Source Window, run at full-speed, single stepped by machine or C level instructions, or stopped at any of the breakpoints. The main features of Debug Software are listed below,

9-3. DoCD HAD2 Debugger Key Features

- In-System-Programming (ISP) of on-chip/off-chip Flash memory
- Two working modes
  - ○ Hardware debugger
  - ○ Software simulator
- Source Level Debugging:
  - ○ C level hardware/software breakpoints
  - ○ C code execution
    - ○ Line by line
    - ○ Over line
    - ○ Out of function
    - ○ Skip line
  - ○ ASM code execution
    - ○ Instruction by instruction
    - ○ Over instruction
    - ○ Out of function
    - ○ Skip instruction
  - ○ ASM, C source view of code
- Symbol Explorer provides hierarchical tree view of all symbols:
  - ○ Modules
  - ○ Functions
  - ○ Blocks
  - ○ Variables and more
- Contents sensitive Watch window
- Symbolic debug including:

- Code
- Variables
- Variable types

- One real-time hardware breakpoints for Program Memory (CODE):
- Two real-time hardware watch-points for each space:
  - Internal (direct) Data Memory (IDM)
  - Special Function Registers (SFR)
  - eXternal Data Memory (XDM)
- Unlimited number of software breakpoints
  - Program Memory
  - Internal (direct) Data Memory (IDM)
  - Special Function Registers (SFR)
  - eXternal Data Memory (XDM)
- Set/clear software or hardware breakpoints, watch-points in Disassembled and C Source Code windows
- 1024 steps deep Software Trace
- Load Program Memory content from:
  - OMF-51, extended OMF-51 files
  - OMF-251 file
  - Intel HEX-51, HEX-386 files
  - BIN file
- Auto refresh of all windows during execution of program
  - Registers' panel including ACC, B, PSW, PC, SP, DPTR, DPP and four banks of general purpose registers R0-R7
  - Internal (direct) Data Memory (IDM)
  - Special Function Registers (SFR)
  - eXternal Data Memory (XDM)

- Timers/Counters
- UARTs
- I/O Ports

- Dedicated windows for peripherals
- Configurable auto refresh time period with 1s step resolution
- Status bar containing number of actually executed instructions, number of clock periods and real processor speed rate
- Hardware Assisted Debugger interface- DTAG interface
- The system runs on a Windows® 98/Me/ 2000/XP PC
- Supports software tools from Keil, Archimedes, IAR, Tasking, Franklin, SDCC and the others

9-4. How to setup DoCD HAD2 Debugger Environment?

The following are the procedures to setup the DoCD HAD2 debugger environment,

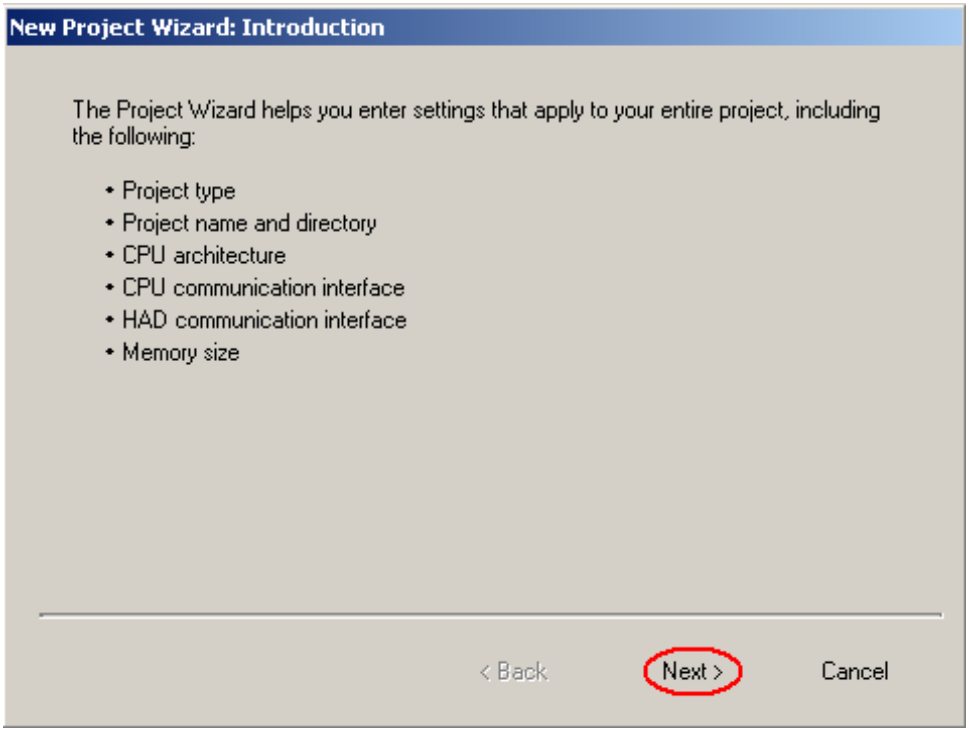
Note: Before setting up the DoCD HAD2 debugger environment, the user must make sure the bit 7 (DBG_PSEL) of AX110xx I2C Configuration EEPROM offset 01h is set to 0 to enable the debug mode. The default value of bit 7 (DBG_PSEL) of AX110xx I2C Configuration EEPROM offset 01h is 1 for normal Ethernet LEDs function.

1. Run the "SOFTWARE\DoCD-setup.exe" setup program to install DoCD Windows Based Debug Software.
2. Connect AX110xx board and DoCD HAD2 board with the DTAG cable.

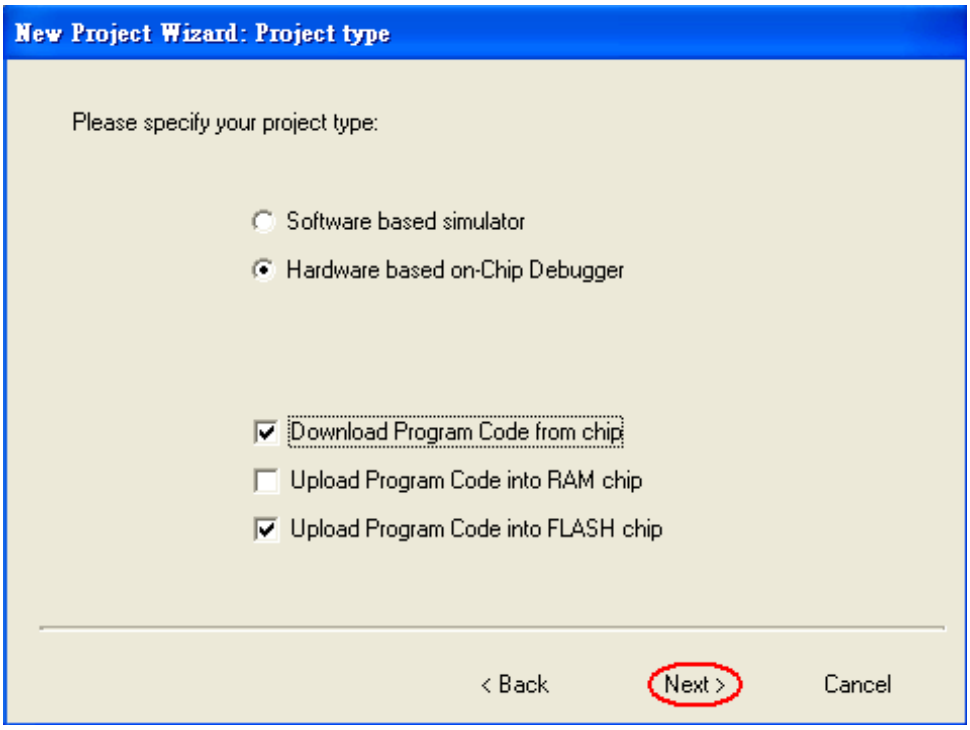
   Note: The user must connect the DTAG cable between AX110xx board and DoCD HAD2 board correctly; otherwise, the DoCD Debugger Software couldn't recognize the AX110xx board.

3. Power up AX110xx board.
4. Connect DoCD HAD2 board and a Windows XP/2K machine with the USB cable.
5. The DoCD HAD2 board will be detected by WinXP/2K system and then the WinXP/2K system will pop up a Dialog window to install the HAD2 driver. Following the screen instructions to complete the driver installation.
6. Run the DoCD Windows Based Debug Software on WinXP/2K machine.
7. Before starting debugging AX110xx Software, you should create a DoCD project file for AX110xx board on DoCD Debug Software.

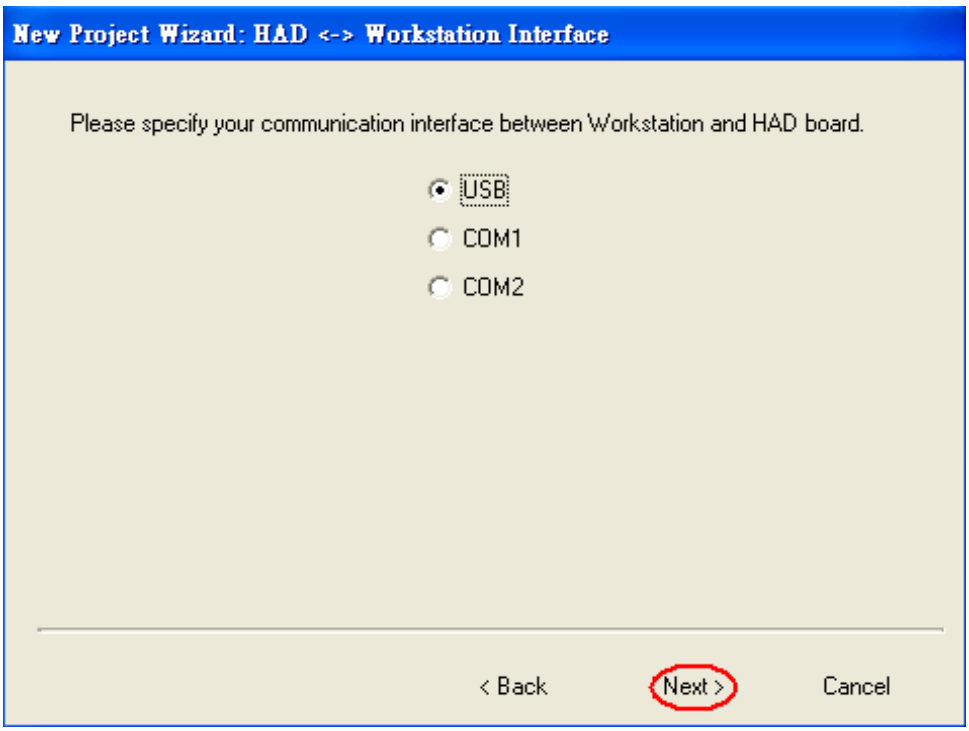
8. To create new project click on the **Create New Project** button or select "Create New" item from debugger's Project menu. Project Creation Wizard will appear to guide you through project creation process:



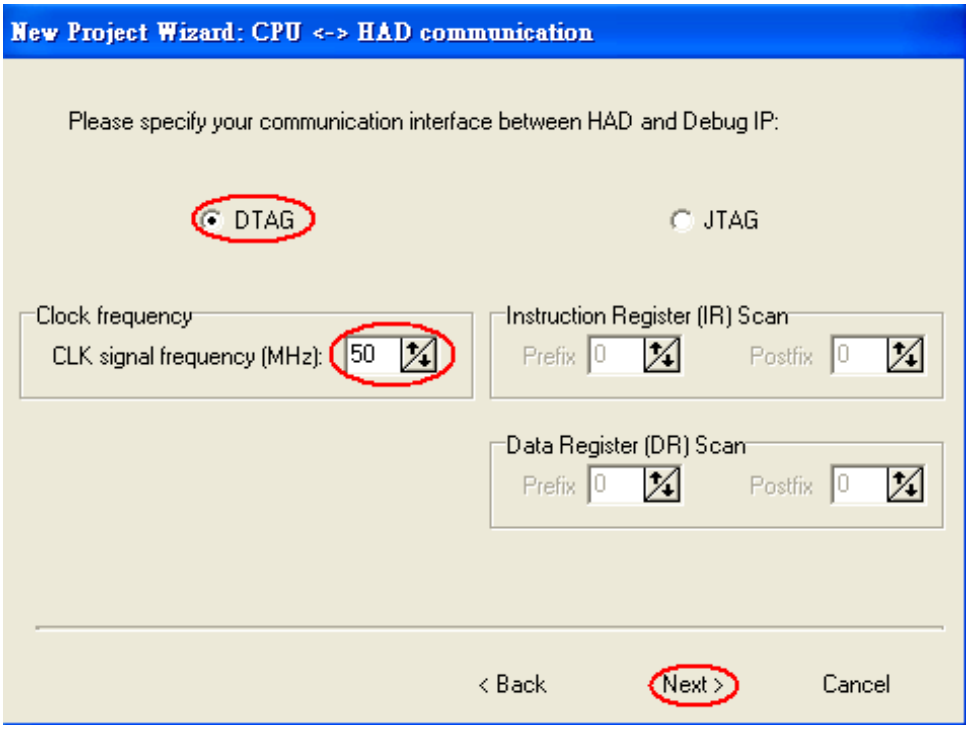
9. Select "Hardware based on-Chip debugger" item and uncheck the "Upload Program Code into RAM chip" item, and then press "Next" button.



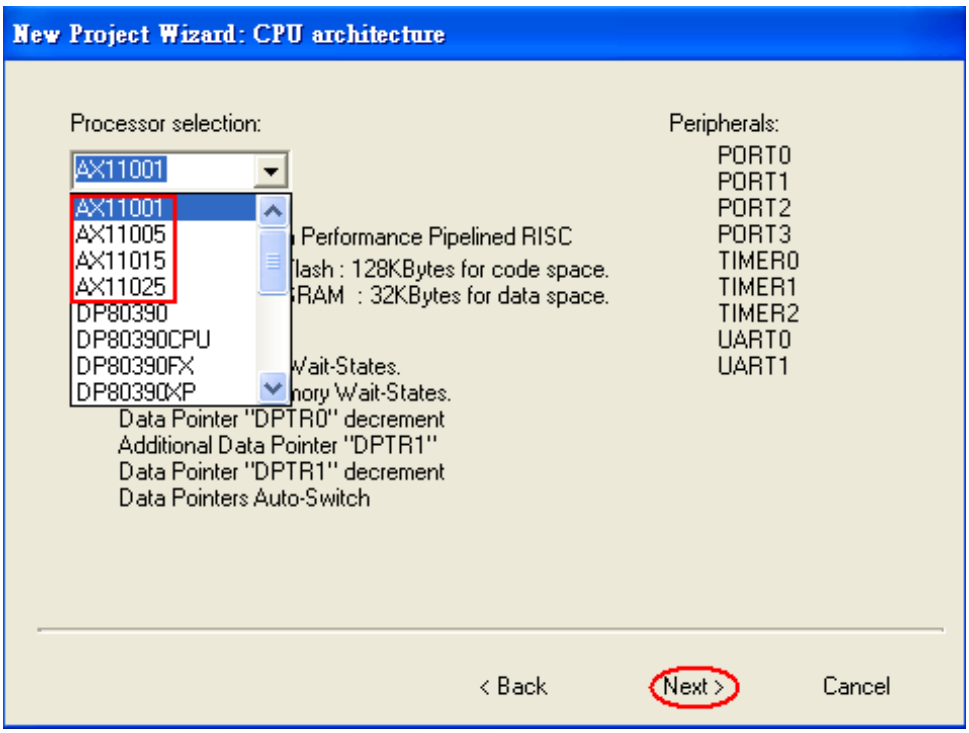
10. Select “USB” communication interface and then press “Next” button.



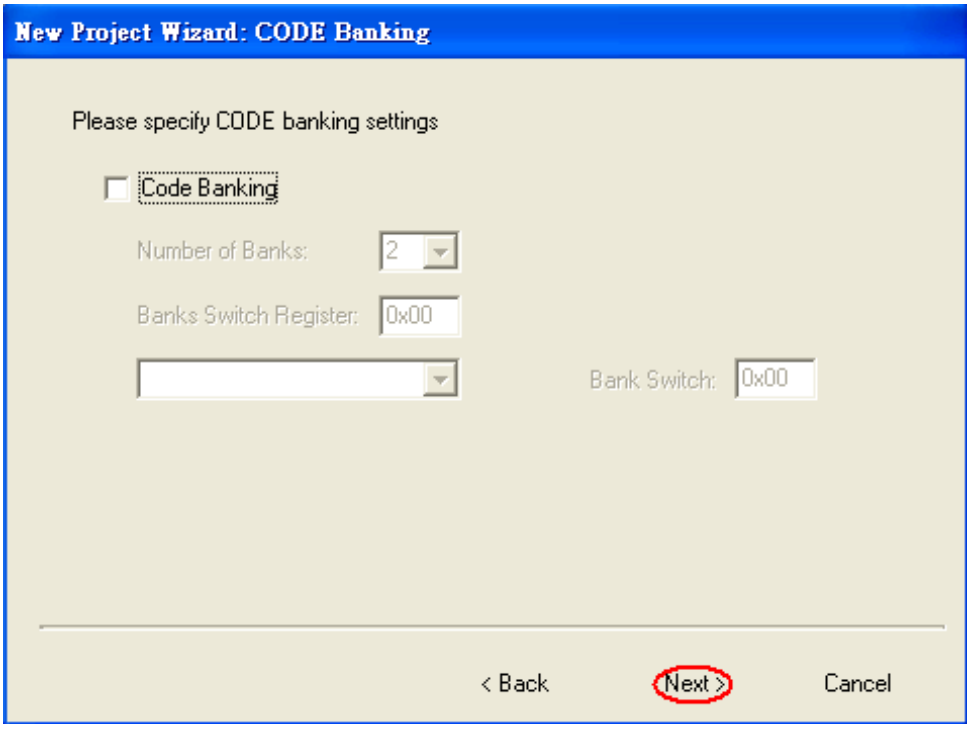
11. Select “DTAG” connecter type and enter the Clock frequency to 25, 50 or 100MHz dependent on the AX110xx board H/W setting, and then press “Next” button.



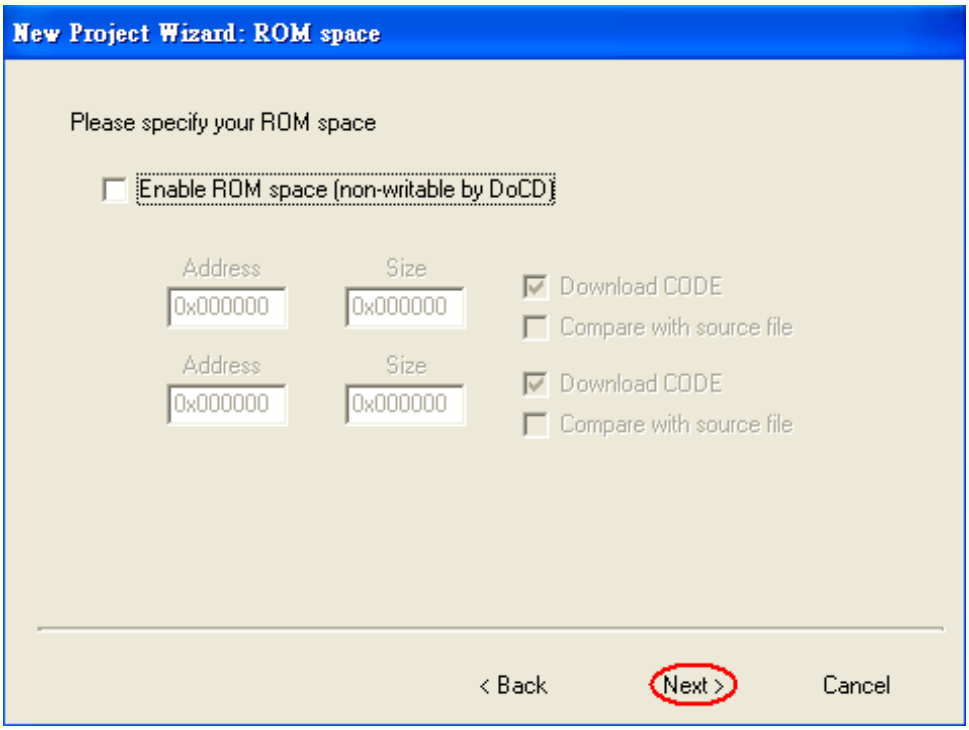
12. Select a proper processor type from the menu and then press “Next” button.



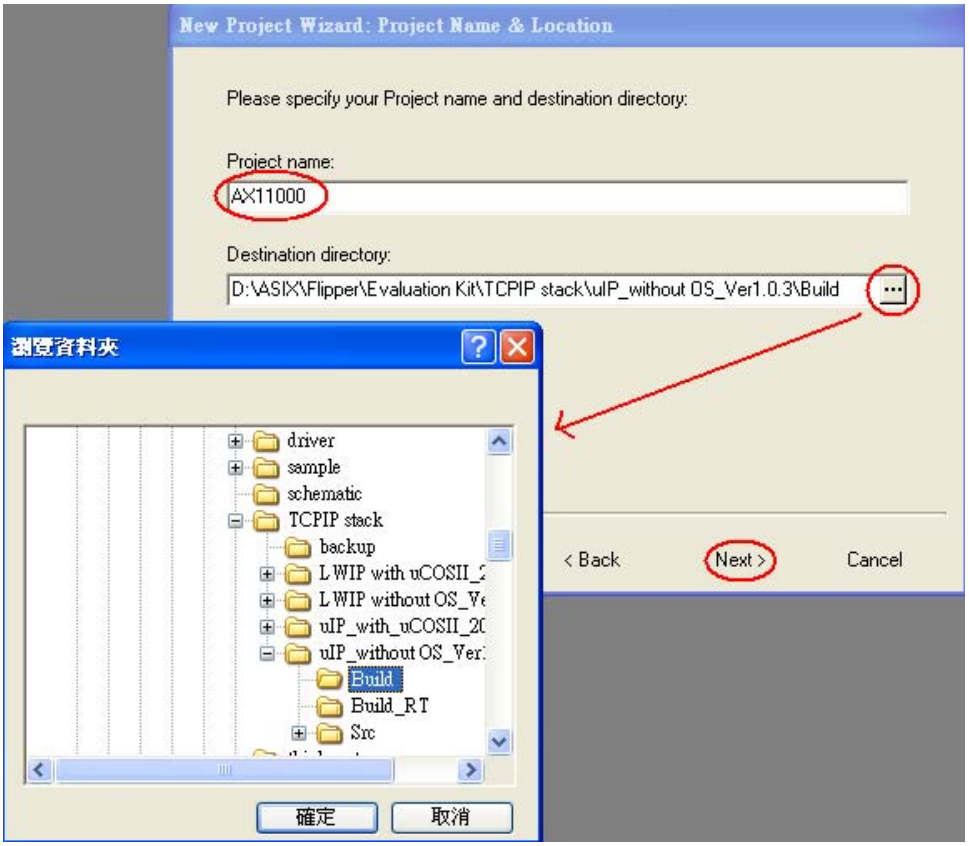
13. Don’t need to set anything and just press “Next” button.



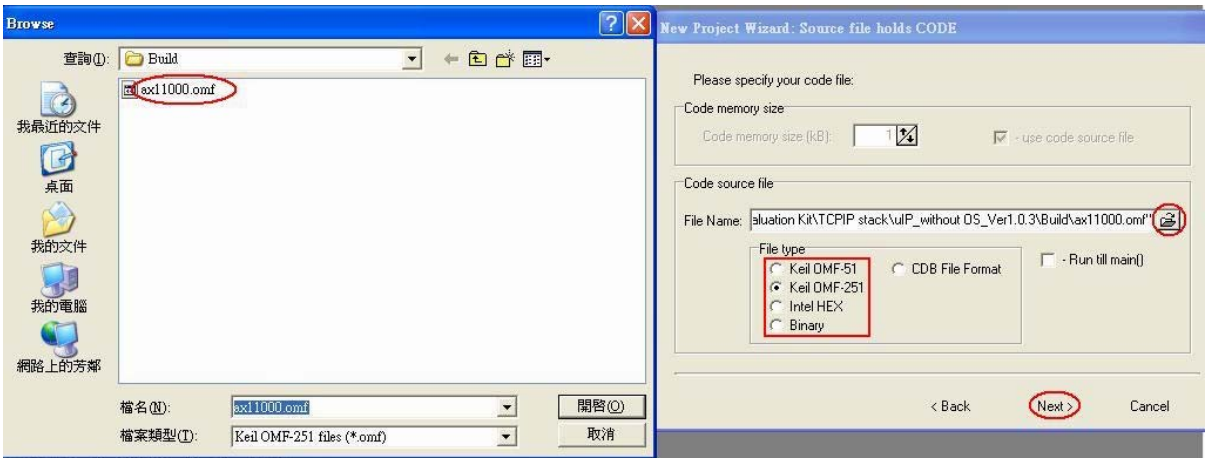
14. Don't need to set anything and just press "Next" button.



15. Enter your project name (e.g. AX11000) and press "…" button to browser the destination directory that AX110xx firmware code is allocated, and then press "Next" button.

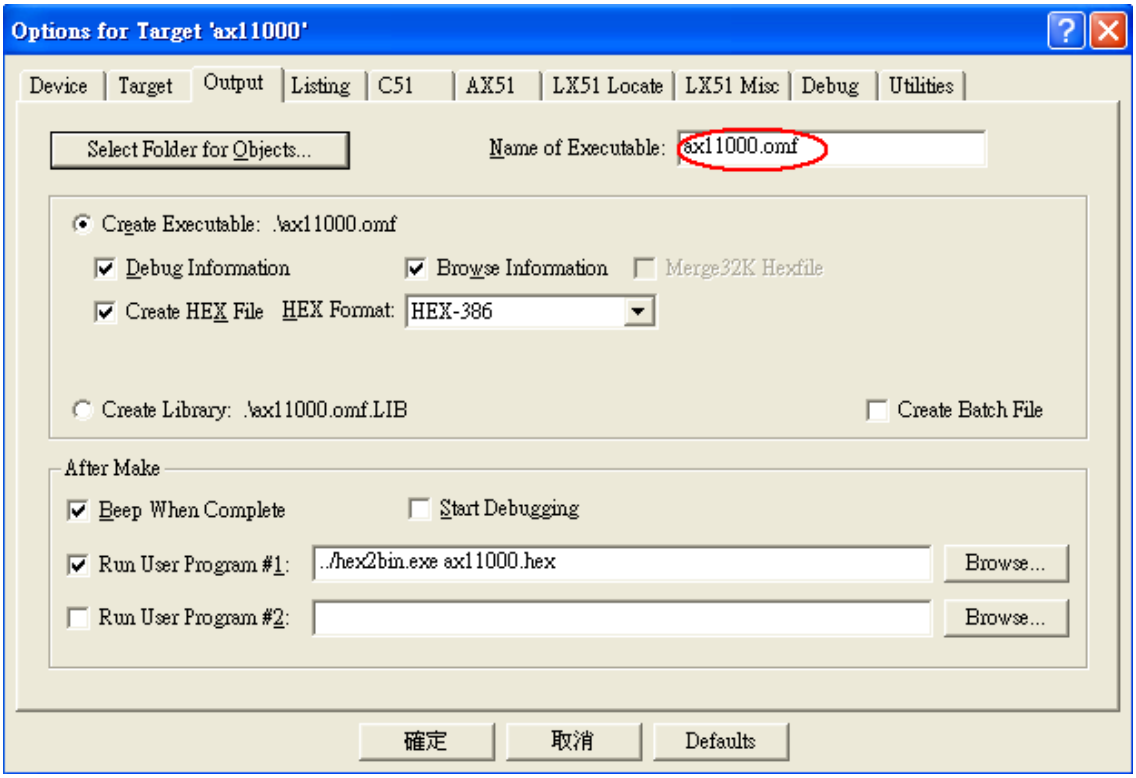


16. Select a proper File type and press button to browse the AX110xx firmware code, and then press “Next” button.

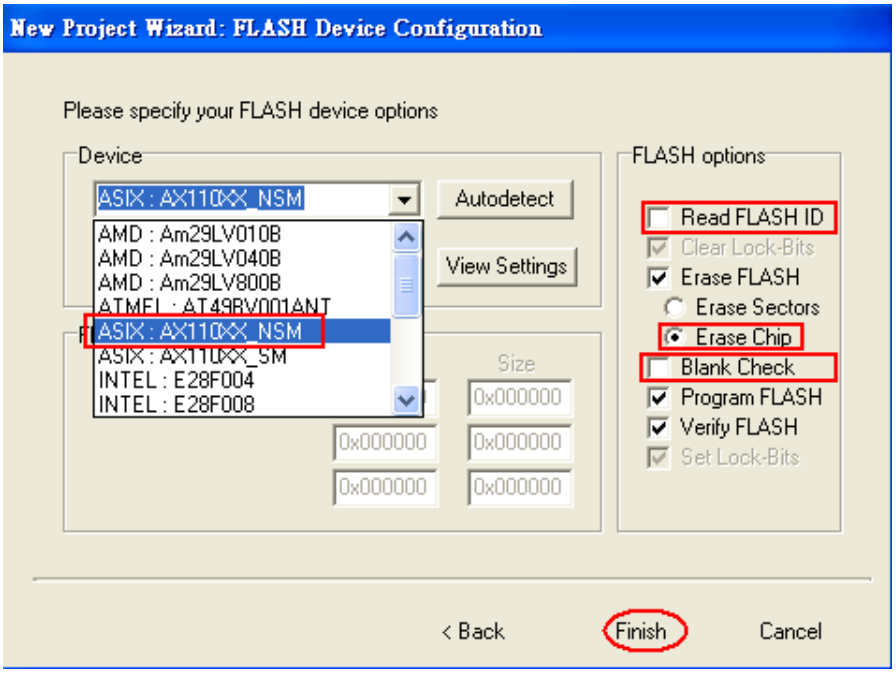


Note:

1. The project file of the HAD2 program <filename>.DOCD has to be at the same folder as the Keil project file .uV2 file (e.g. ax110xx.uv2) because the .omf file (e.g. ax11000.omf) has the information of all source files with respect to Keil project file location; otherwise, you might not do the source debugging with HAD2 debugger.
2. In order to build an “ax11000.omf” file, you should modify the Options of the AX110xx software project file (e.g. ax11000.uv2) to change the Output Executable File Name to “ax11000.omf” in the Keil uVision environment. (Refer to below pictures)



17. Select the "ASIX: AX110XX_NSM" FLASH device, check the "Erase Chip" box and uncheck the "Read FLASH ID" and "Blank Check" boxes, and then press "Finish" button to complete the project file creation.



18. If the connection between AX110xx board, DoCD HAD2 debugger and WinXP/2K machine is not established, you will see below Connection error message.

Note: If you couldn't get a connection between AX110xx and DoCD HAD2 debugger, please check below points to isolate this issue,

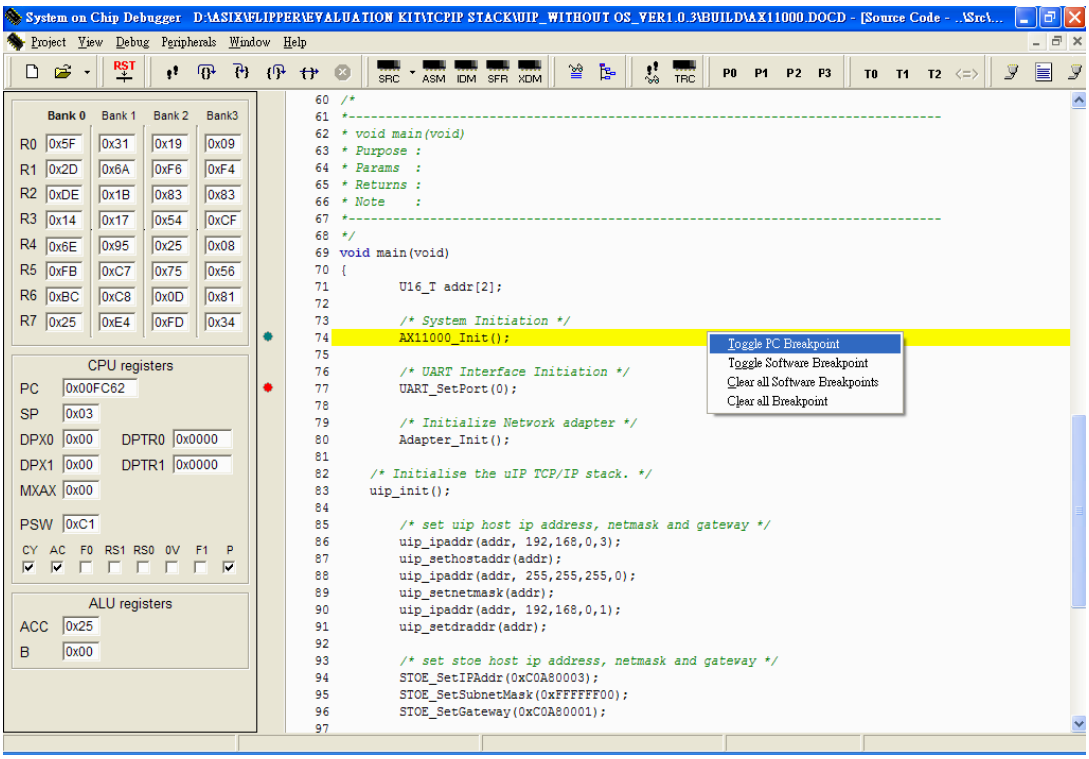
1. Make sure if the version number of AX110xx boot loader code (bl_docd.bin) is v1.20 or later for DoCD HAD2 debugger or not. If your AX110xx software doesn't run with AX110xx Ethernet boot loader code, please check the following statement in the **cpu\start390.a51** file of AX110xx demo firmware source code to make sure if the AX110xx Read Protection function was disabled or not. The Read Protection function of AX110xx should be disabled (i.e. CODEBLOCK = 8Ch) in order to run AX110xx software with the DoCD HAD2 debugger. If the Read Protection function of AX110xx was enabled (i.e. CODEBLOCK = 0Ch), DoCD HAD2 debugger couldn't get a connection with AX110xx development board.

   CODEBLOCK EQU 8CH ; the block size of code space.

2. Make sure bit 7 of AX110xx I2C Configuration EEPROM offset 01h is set to 0.
3. Make sure the direction of HAD2 cable was correctly connected to AX110xx and HAD2 debugger.
4. Please contact DCD's support guys or visit DCD's web site (http://www.dcd.pl/aappln.php) to get more information about how to setup the HAD2 debugger environment.

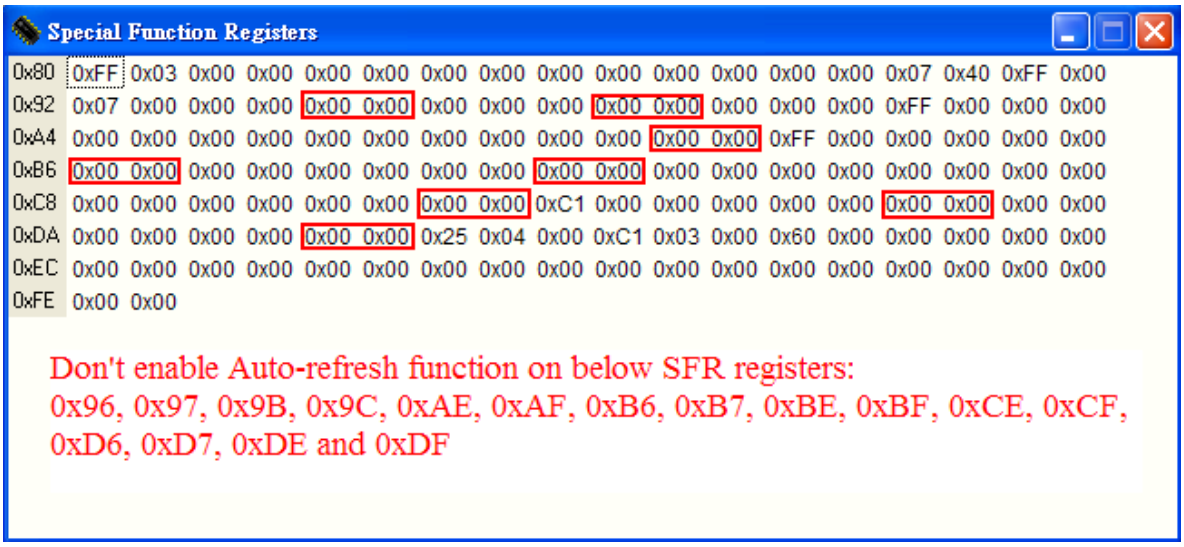
19. If the connection between AX110xx board, DoCD HAD2 debugger and WinXP/2K machine is established successfully, you will see below "Flash memory upload" and "SFR download" status bars.

20. When the "SFR download" status bar shows "100%", the AX110xx firmware code is uploaded successfully and is waiting for debugging now.

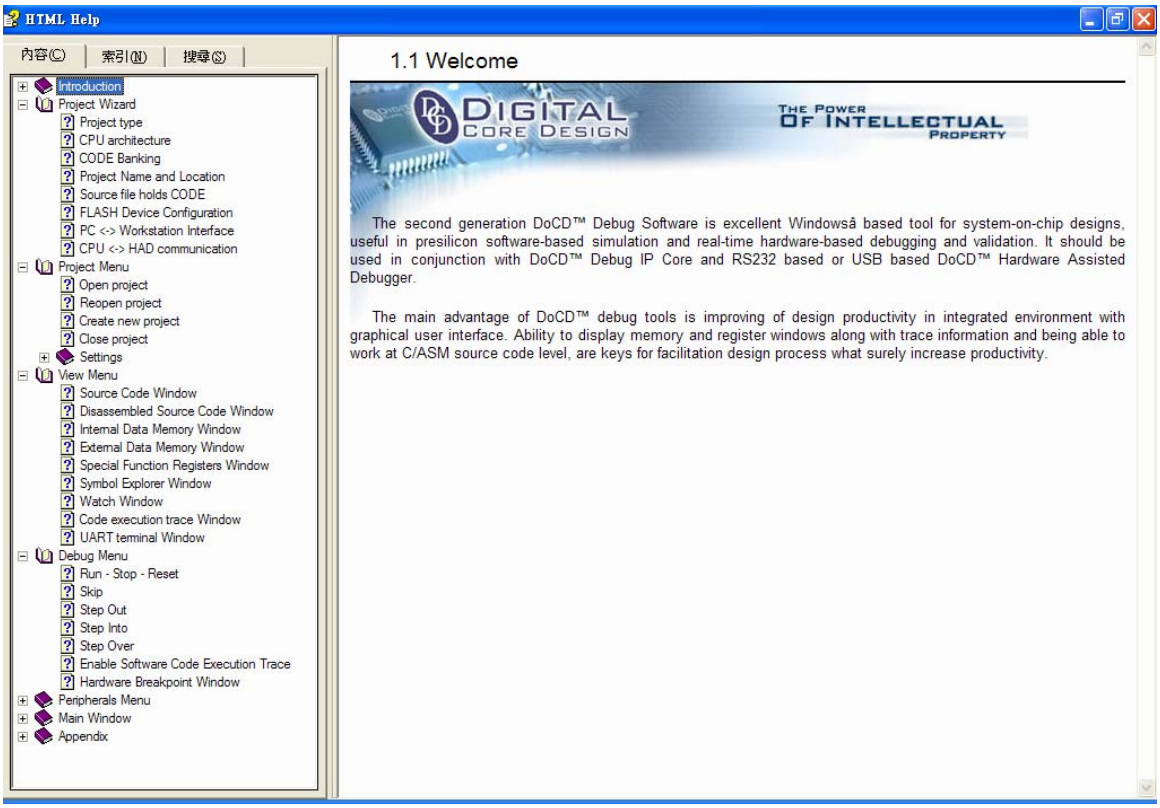



```
/*
*-------------------------------------------------------------------------------
* void main(void)
* Purpose :
* Params  :
* Returns :
* Note    :
*-------------------------------------------------------------------------------
*/
void main(void)
{
        U16_T addr[2];

        /* System Initiation */
        AX11000_Init();

        /* UART Interface Initiation */
        UART_SetPort(0);

        /* Initialize Network adapter */
        Adapter_Init();

    /* Initialise the uIP TCP/IP stack. */
    uip_init();

        /* set uip host ip address, netmask and gateway */
        uip_ipaddr(addr, 192,168,0,3);
        uip_sethostaddr(addr);
        uip_ipaddr(addr, 255,255,255,0);
        uip_setnetmask(addr);
        uip_ipaddr(addr, 192,168,0,1);
        uip_setdraddr(addr);

        /* set stoe host ip address, netmask and gateway */
        STOE_SetIPAddr(0xC0A80003);
        STOE_SetSubnetMask(0xFFFFFF00);
        STOE_SetGateway(0xC0A80001);
```

Note: Don't enable the auto-refresh function of SFR window when the firmware is accessing the following AX110xx peripheral related SFR registers (0x96, 0x97, 0x9B, 0x9C, 0xAE, 0xAF, 0xB6, 0xB7, 0xBE, 0xBF, 0xCE, 0xCF, 0xD6, 0xD7, 0xDE and 0xDF); otherwise, the AX110xx peripheral interfaces might not be configured properly through these indirect index/data registers.



21. You press “F1” key to display the DoCD Help screen and refer to the DoCD Help for the detail information about how to use the DoCD Software Debugger utility.

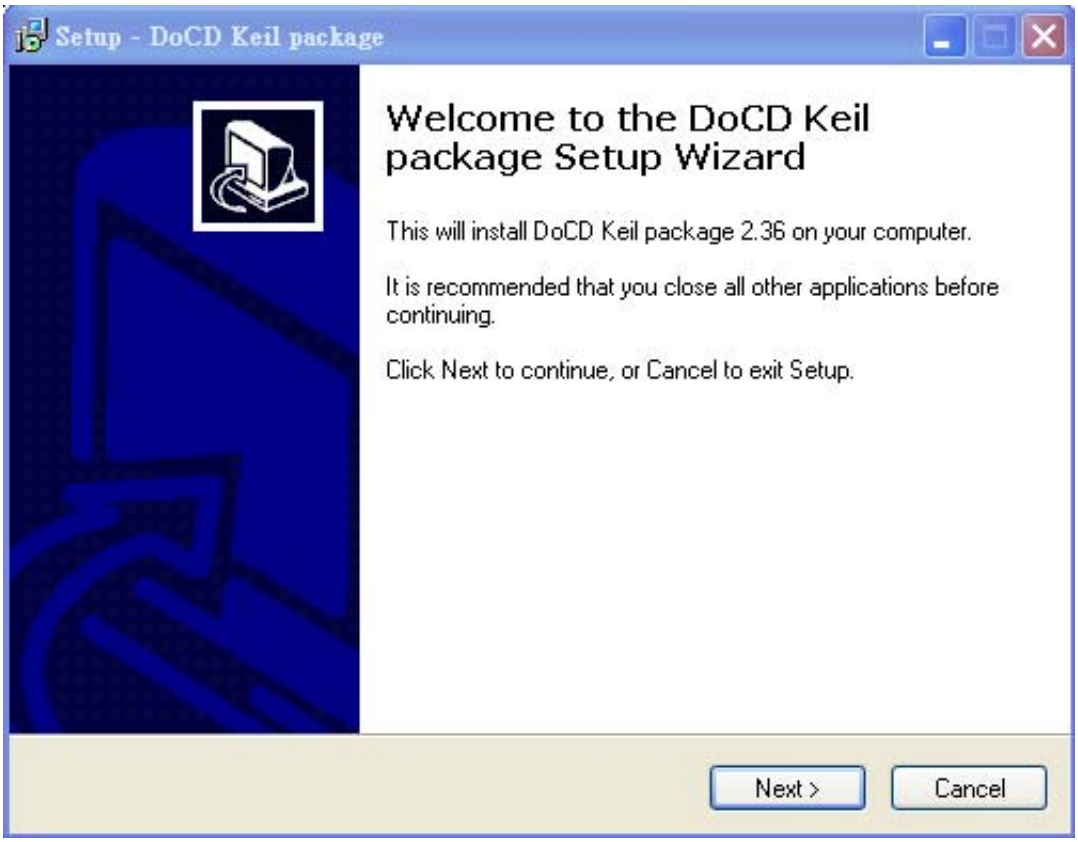


9-5. How to use DoCD HAD2 Debugger with Keil IDE?

All of the AX110XX Cores are now supported by Keil uVision AGSI interface. It allows Keil software users to work within uVision environment without switching between Keil compiler and external DoCD software. Everything (compilation, simulation, debugging) is under control of a single application. This section describes how to download your design with AX110xx development board in Keil IDE without an external DoCD software.

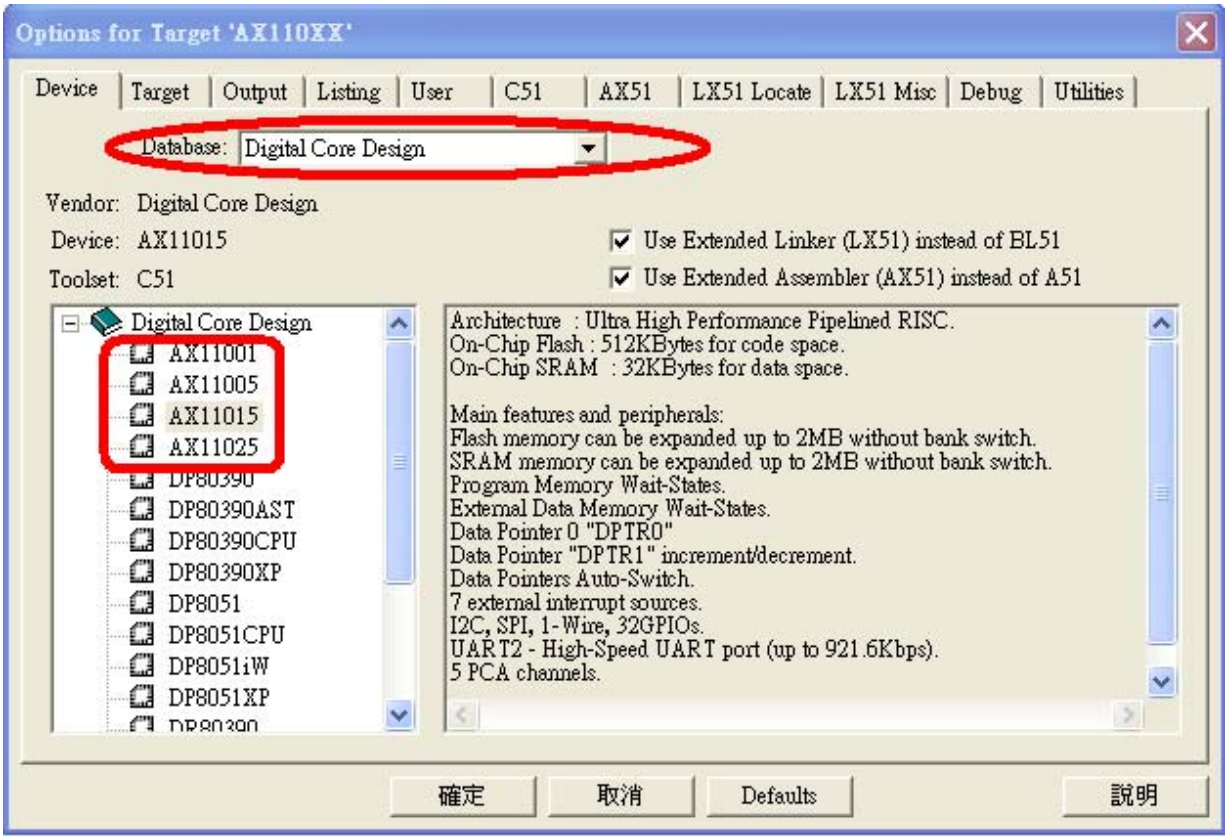
Manual installation of DoCD executable program "DoCDKeil-setup.exe", and all steps are performed automatically by setup program. After installation completing, the related Keil DLL files will includes and can be use to debug for AX110xx development board. The following directories can be found inside package:

.**C51\BIN**\ : Contains simulation DLL files
.**C51\INC\DCD**\ : Contains DCD's 8051/80390 include files
.**C51\EXAMPLES\DCDsim**\ : Contains example C/ASM 8051 application
.**DATASHTS\DCD**\ : Contains DCD's 8051/80390 related PDF files
.**UV2**\ : Contains DCD's 8051/80390 uVision2/uVision3 database

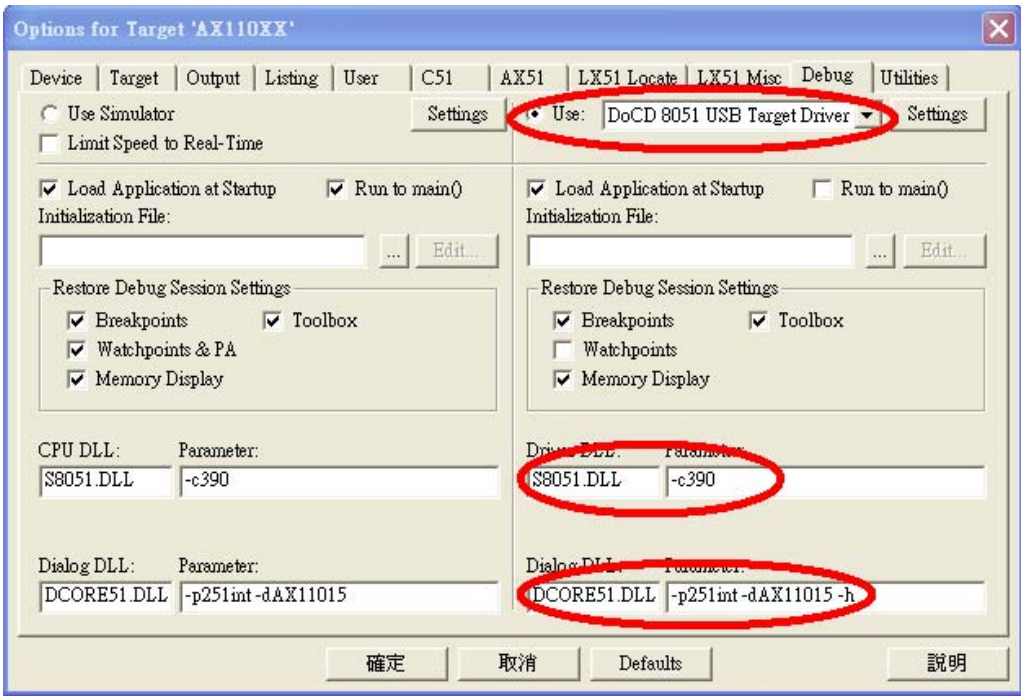


Note: The latest DoCD debugger software supported related Keil DLL files. If your DoCD debugger software doesn't install Keil DLL files. Please contact DoCD's guys to get the appropriate support.

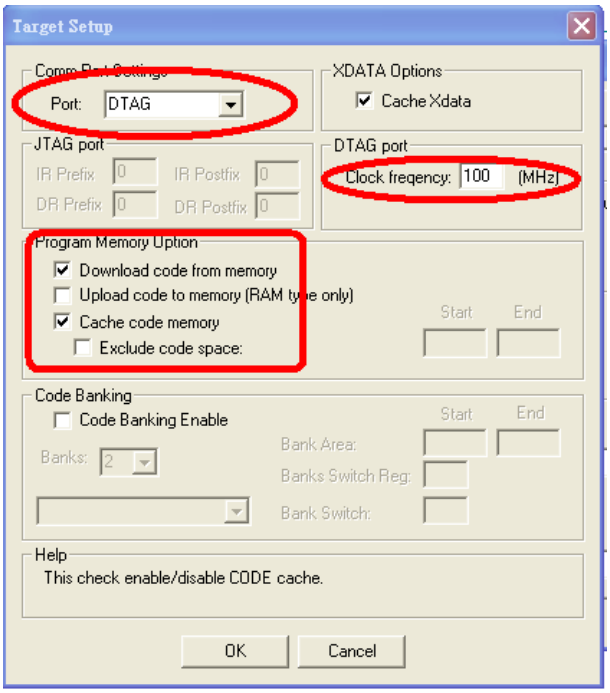
1. Run Keil uVision IDE. Select “Project – Open Project”, the Select Project dialog comes up. Select your project file to work.
2. Select “Options for Target – Device”. From the “Database” list-box, select “Digital Core Design” which is DoCD’s devices database. Select required microcontroller type (AX110xx) you actually have.



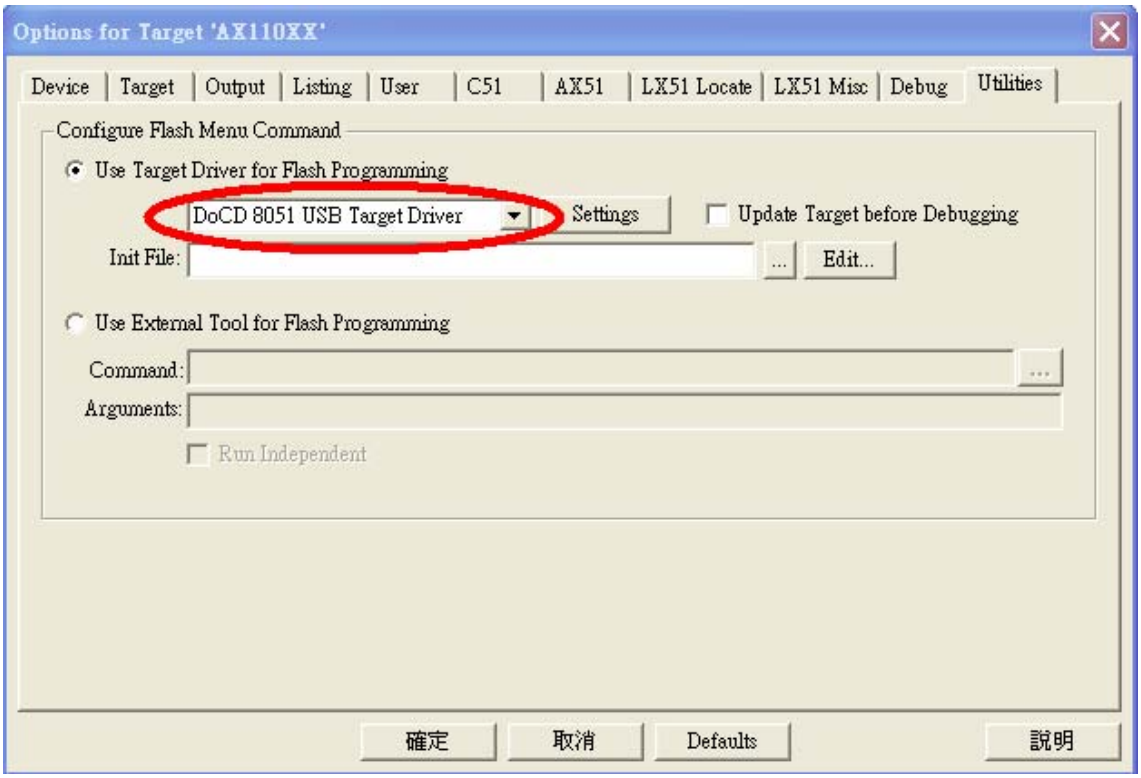
3. Select “Options for Target – Debug”. Choosing the “Use DoCD 8051 USB Target Driver”. If you have installed the DoCD setup executable program, the item “DoCD 8051 USB Target Driver” will be shown in the list box.
4. CPU DLL is “S8051.DLL”, and Parameter is “-c390”.
5. Dialog DLL is “DCORE51.DLL”, and Parameter is “-p251int -dAX11015 -h” if you are selected AX11015 microcontroller type.



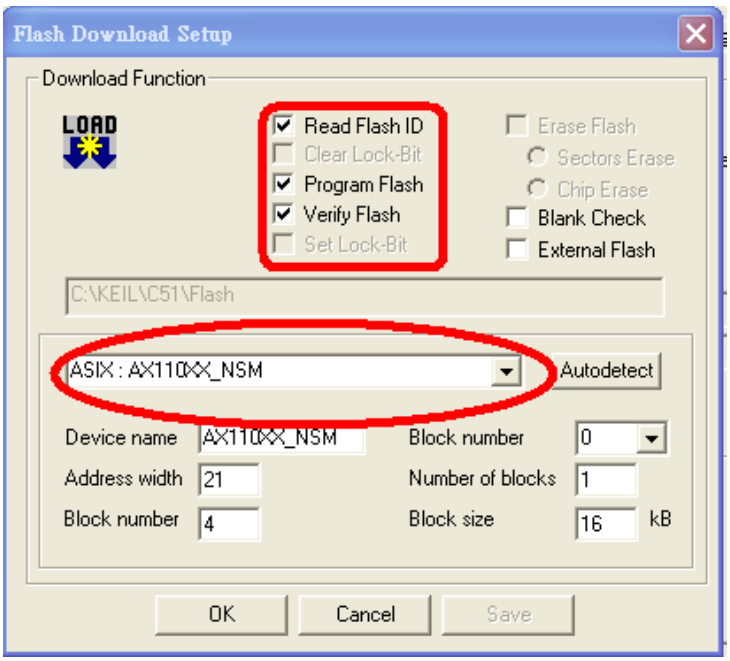
6. Click “Settings” to set your hardware debugger.
7. Comm Port Settings ：DTAG
8. Clock Frequency : meet your system. (AX110XX has 25MHz, 50MHz, and 100MHz)



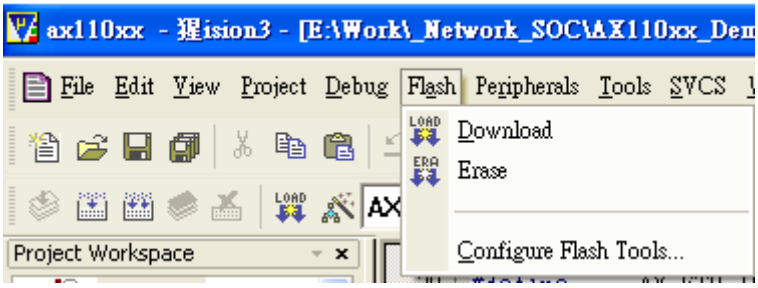
9. Select “Options for Target – Utilities”. Check “Use Target Driver for Flash Programming” and choose the “Use DoCD 8051 USB Target Driver”



10. Click “Settings” to set your Flash Programming.
11. Choose Flash device：ASIX:AX110XX_NSM for Non-Shadow mode.
    ASIX:AX110XX_SM for Shadow mode.
12. Check items “Read Flash ID”, “Program Flash” and “Verify Flash”. The “External Flash” item must only be checked if you have an external flash.



13. After completing these setting, you can download source code into target device by clicking “download” button or selecting “Flash—Download” on tool bar to run hardware debugger on AX110xx development board in Keil IDE.



10. AX110xx Mass Production Solutions

To support the mass production for those products using AX110xx chips. ASIX Electronics provides 3rd Party Universal Programmer and AX110xx Manufacture Program solutions for AX110xx customers. This section provides a brief introduction for both solutions. Please refer to "**AX110xx Mass Production Application Note**" for details.

10-1. Advantech Equipment Corporation's Flash Programmer Solution

AX110xx family is supported by Advantech Equipment Corp.'s LABTOOL-848XP Turbo Gang Programmer. The LABTOOL-848XP programmer supports to program up to 8 AX110xx chips at the same time. Please contact Advantech Equipment support guys (www.aec.com.tw) to get the detailed user manual and update the software if necessary.

Users can purchase LABTOOL-848XP Turbo Gang Programmer and AX11015/AX11025 128-pin socket board (i.e. **SDP-AX110-128LQ**) from Advantech Equipment Corp.

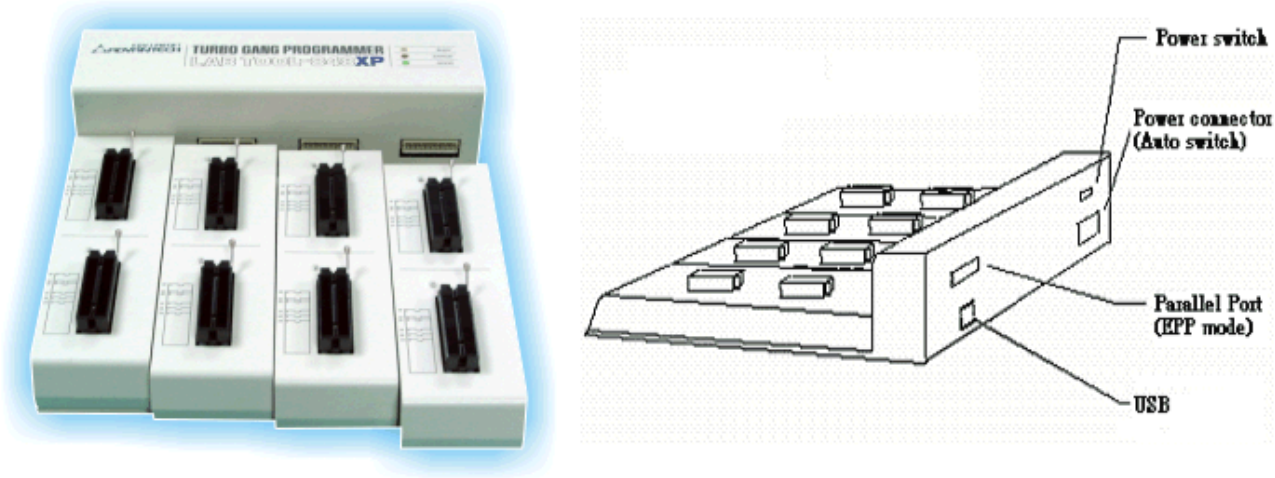


Figure 24. AEC's LABTOOL-848XP Turbo Gang Programmer

10-2. ELNEC's Flash Programmer Solution

AX110xx family is supported by ELNEC's device programmer series such as BeeHive, and BeeProg, etc. These device programmer series support up to 8 AX110xx chips Flash programming operation at one time. Please visit ELNEC's website at www.elnec.com to get the detailed user manual and update the software if necessary.

Users can purchase ELNEC's Device Programmer and AX11001/AX11005 80-pin socket board (i.e. **LQFP80**) or AX11015 128-pin socket board (i.e. **LQFP128**) from ELNEC (http://www.elnec.com/sw/dev_html/asix_electronics_dev.htm).

Figure 25. ELNEC's BeeHive 208S and 204 Programmer

10-3. AX110xx Manufacture Program Solution

To ease the testing in mass production for those products using AX110xx chip, a sample manufacture program is provided for AX110xx customers. The AX110xx manufacture program is a Windows dialog-based software tool that can be used to test the AX110xx chip on a product. Source codes as well as the binary files are provided that customers can modify the sample codes to meet the real requirements in mass production.

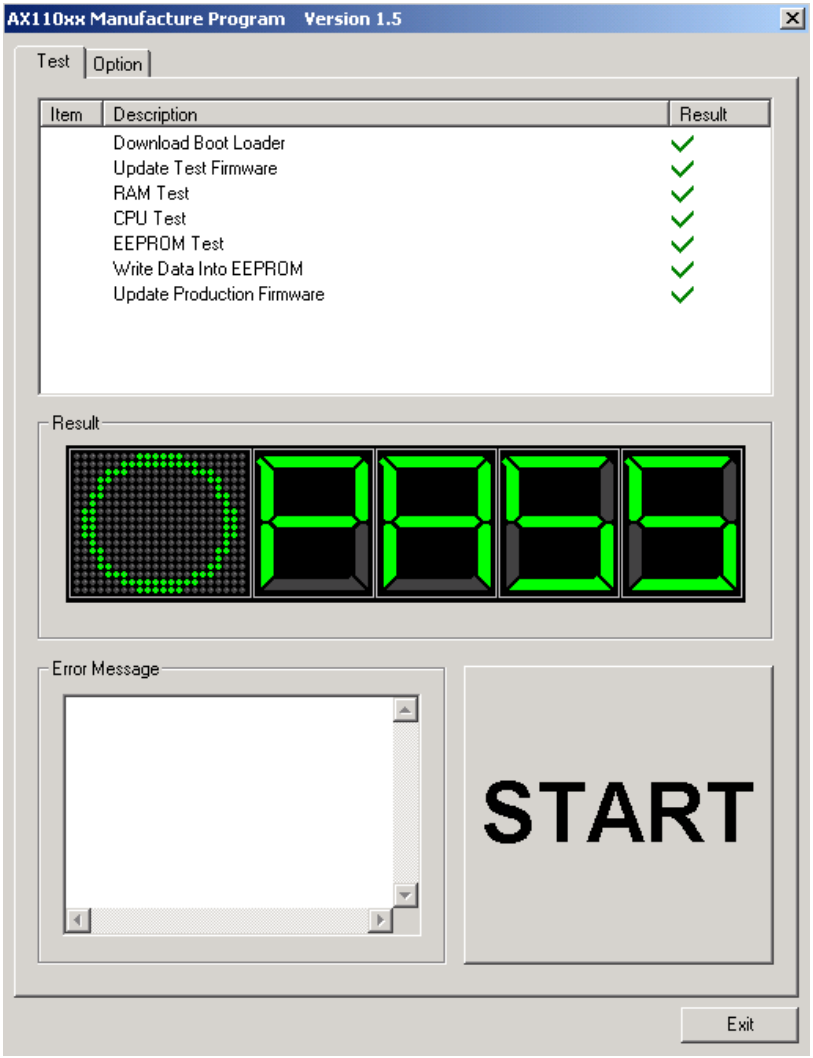


Figure 26. AX110xx Manufacture Program

Appendix A. Software Availability

ASIX Electronics provides the following utilities, software modules and TCP/IP stack sample codes for customers' reference. Please contact ASIX's Sales (sales@asix.com.tw) for detailed information.

AX110xx Software	Source Code Release	Need to sign a NDA
CPU Module	YES	NO
Ethernet Module	YES	NO
S/W DMA Module	YES	NO
MS Timer Module	YES	NO
Local Bus Module	YES	NO
I2C Module	YES	NO
SPI Module	YES	NO
1-Wire Module	YES	NO
CAN Module	YES	NO
UART2 Module	YES	NO
UART Module	YES	NO
PCA Module	YES	NO
Buffer Module	YES	NO
Adapter Module	Available upon request	NO
TCP/IP Module	NO	NO
PPPoE Module	Available upon request	NO
DHCP Client Module	Available upon request	NO
FTP Client Module	Available upon request	NO
HTTP Module	Available upon request	NO
DNS Module	Available upon request	NO
DYNDNS Module	Available upon request	NO
SMTP Module	Available upon request	NO
SNTP Module	Available upon request	NO
UPNP Module	Available upon request	NO
Boot Loader Code	NO	NO
I2C Module Sample Code	YES	NO
SPI Module Sample Code	YES	NO
GPIO Port 3 LED Control Sample Code	YES	NO
AX11001/AX11005 80-pin Development Board Demo Firmware	YES	NO
AX11015 128-pin Development Board Demo Firmware	YES	NO
AX11025 128-pin Development Board Demo Firmware	YES	NO
AX11005 FreeRTOS RTOS Demo Firmware	Available upon request	YES
AX11015 FreeRTOS RTOS Demo Firmware	Available upon request	YES
AX11025 FreeRTOS RTOS Demo Firmware	Available upon request	YES
AX11005 uCOSII RTOS Demo Firmware	Available upon request (Note 1)	YES
AX11015 uCOSII RTOS Demo Firmware	Available upon request (Note 1)	YES
AX11025 uCOSII RTOS Demo Firmware	Available upon request (Note 1)	YES

AX110xx RS-232 to Ethernet Reference Design Demo Firmware	Available upon request	Yes
AX110xx AXR2E Configuration Utility	NO	NO
uIP TCP/IP Stack without OS	YES	NO
LWIP TCP/IP Stack without OS	YES	NO
Manufacture Program	Available upon request	YES
Windows ISP	NO	NO
DOS Flash Programming Utility (UARTL.EXE)	NO	NO
DOS Flash Programming Utility for 921K bps Baud Rate (UARTH.EXE)	Available upon request	NO
Device Finder Utility (DEVICEFINDER.EXE)	NO	NO

Figure 27. AX110xx Software Availability

Note 1: The uC/OS-II related sample codes don't include uC/OS-II source files. If user wants to implement AX110xx software with uC/OS-II, one may need to license the uC/OS-II source code from the Micrium (http://www.micrium.com/).

Appendix B. AX1100x 80-pin Development Board EEPROM Default Setting

EEPROM Offset	Description	Value
0x00	Length	0x21
0x01	Flag	0xFC
0x02	Multi-function Pin Setting 0	0x00
0x03	Multi-function Pin Setting 1	0x80
0x04	Programmable Output Driving Strength	0x30
0x05	Reserved	0x00
0x0B~0x06	Node ID 5 ~ ID 0 (Note)	0x00 0x00 0x00 0x00 0x00 0x01
0x0D~0x0C	Maximum Packet Size	0x05 0xF2
0x0E	Primary PHY Type and PHY ID	0x10
0x0F	Secondary PHY Type and PHY ID	0xE0
0x10	Pause Frame High Water Mark	0x1D
0x11	Pause Frame Low Water Mark	0x19
0x12	Reserved	0x87
0x13	Reserved	0x00
0x15~0x14	TOE TX VLAN Tag	0xFF 0xFF
0x17~0x16	TOE RX VLAN Tag	0xFF 0xFF
0x18	TOE ARP Cache Timeout	0x10
0x1C~0x19	TOE Source IP Address	0xC0 0xA8 0x00 0x03
0x20~0x1D	TOE Subnet Mask	0xFF 0xFF 0xFF 0x00
0x21	TOE L4 DMA Transfer Gap	0x04
0x2F~0x22	Reserved for H/W future use	0xFF 0xFF … 0xFF
0x7F~0x30	Reserved for S/W and Driver Settings	0xFF 0xFF … 0xFF

Figure 28. AX1100x 80-pin Development Board EEPROM Default Setting

Note: Users should assign a unique Node ID address for each AX110xx device.

Appendix C. AX11015 128-pin Development Board EEPROM Default Setting

EEPROM Offset	Description	Value
0x00	Length	0x21
0x01	Flag	0xBC
0x02	Multi-function Pin Setting 0	0x00
0x03	Multi-function Pin Setting 1	0x80
0x04	Programmable Output Driving Strength	0x30
0x05	Reserved	0x00
0x0B~0x06	Node ID 5 ~ ID 0 (Note)	0x00 0x00 0x00 0x00 0x00 0x01
0x0D~0x0C	Maximum Packet Size	0x05 0xF2
0x0E	Primary PHY Type and PHY ID	0x10
0x0F	Secondary PHY Type and PHY ID	0xE0
0x10	Pause Frame High Water Mark	0x1D
0x11	Pause Frame Low Water Mark	0x19
0x12	Local Bus Setting 0	0x87
0x13	Local Bus Setting 1	0x00
0x15~0x14	TOE TX VLAN Tag	0xFF 0xFF
0x17~0x16	TOE RX VLAN Tag	0xFF 0xFF
0x18	TOE ARP Cache Timeout	0x10
0x1C~0x19	TOE Source IP Address	0xC0 0xA8 0x00 0x03
0x20~0x1D	TOE Subnet Mask	0xFF 0xFF 0xFF 0x00
0x21	TOE L4 DMA Transfer Gap	0x04
0x2F~0x22	Reserved for H/W future use	0xFF 0xFF … 0xFF
0x7F~0x30	Reserved for S/W and Driver Settings	0xFF 0xFF … 0xFF

Figure 29. AX11015 128-pin Development Board EEPROM Default Setting

Note: Users should assign a unique Node ID address for each AX110xx device.

Appendix D. AX11025 128-pin Development Board EEPROM Default Setting

EEPROM Offset	Description	Value
0x00	Length	0x21
0x01	Flag	0xBC
0x02	Multi-function Pin Setting 0	0x00
0x03	Multi-function Pin Setting 1	0xC0
0x04	Programmable Output Driving Strength	0x30
0x05	Reserved	0x00
0x0B~0x06	Node ID 5 ~ ID 0 (Note)	0x00 0x00 0x00 0x00 0x00 0x01
0x0D~0x0C	Maximum Packet Size	0x05 0xF2
0x0E	Primary PHY Type and PHY ID	0x10
0x0F	Secondary PHY Type and PHY ID	0xE0
0x10	Pause Frame High Water Mark	0x1D
0x11	Pause Frame Low Water Mark	0x19
0x12	Local Bus Setting 0	0x87
0x13	Local Bus Setting 1	0x00
0x15~0x14	TOE TX VLAN Tag	0xFF 0xFF
0x17~0x16	TOE RX VLAN Tag	0xFF 0xFF
0x18	TOE ARP Cache Timeout	0x10
0x1C~0x19	TOE Source IP Address	0xC0 0xA8 0x00 0x03
0x20~0x1D	TOE Subnet Mask	0xFF 0xFF 0xFF 0x00
0x21	TOE L4 DMA Transfer Gap	0x04
0x2F~0x22	Reserved for H/W future use	0xFF 0xFF … 0xFF
0x7F~0x30	Reserved for S/W and Driver Settings	0xFF 0xFF … 0xFF

Figure 30. AX11025 128-pin Development Board EEPROM Default Setting

Note:
1. Users should assign a unique Node ID address for each AX110xx device.
2. The Multi-function Pin Setting 1 (offset 0x03) of AX11025 development board is set to enable the 1-Wire interface for 1-Wire temperature sensor demonstration. Users should write 0xC0 to AX11025 EEPROM offset 0x03 and set a proper setting on S8 switch on AX11025 development board to enable the CAN interface if necessary. Please refer to Section 5-2-7 for details.

4F, No.8, Hsin Ann Rd., Hsinchu Science Park,
Hsinchu, Taiwan, R.O.C.

TEL: +886-3-5799500
FAX: +886-3-5799558

Email: support@asix.com.tw
Web: http://www.asix.com.tw