

# W7200-EVB User's Guide

Version 1.1



© 2011 WIZnet Co., Ltd. All Rights Reserved.

For more information, visit our website at <http://www.wiznet.co.kr>

## Table of Contents

1	Introduction.....	4
2	Specification.....	5
3	W7200-EVB Block Diagram .....	6
4	Hardware Layout.....	7
5	Expansion Port Interface .....	8
6	Development environment .....	10
6.1	IDE.....	10
6.2	Install Flash loader Demonstrator .....	10
6.3	USB-to-UART interface IC Driver .....	10
7	W7200 SPI .....	12
8	Reference Firmware.....	14
8.1	W7200 Socket API .....	14
8.2	Firmware Structure.....	14
8.3	Firmware Functions .....	15
8.4	Firmware Build and Upload .....	17
8.4.1	Build - IAR Embedded Workbench IDE.....	17
8.4.2	Upload - Flash Loader Demonstrator .....	18
9	Application Demonstration.....	21
9.1	Flow of Demonstration.....	21
9.2	Ping TEST .....	21
9.2.1	Network configuration for TEST PC.....	21
9.2.2	Ping Test at Command Prompt.....	22
9.2.3	App. Test - Loopback TEST .....	22
10	Physical Specification .....	25
11	Schematic .....	26
	Document History Information .....	27

## Table of Figures

Figure 1 W7200-EVB Block Diagram .....	6
Figure 2 W7200-EVB Layout.....	7
Figure 3 Download Flash loader Demonstrator .....	10
Figure 4 Currently Supported VCP Drivers (16DEC2011) .....	11
Figure 5 W7200 SPI Interface .....	12
Figure 6 TCP Server / Client.....	14
Figure 7 Compile on IAR Embedded Workbench IDE.....	17
Figure 8 Flow Chart of Demonstration.....	21
Figure 9 Internet Protocol Properties .....	22
Figure 10 Ping Test at Command Prompt .....	22
Figure 11 W7200-EVB Board Dimension.....	25

# 1 Introduction

W7200-EVB is the evaluation board for testing iMCU W7200 and prototyping development. W7200-EVB is composed of a W7200 based on ARM Cortex M3 CPU core, a W7200 which acts as Hardwired TCP/IP embedded Ethernet controller, and a FT232R which acts as USB-to-UART interface IC. W7200 has been proven in various fields to work as a fully hardwired TCP/IP implemented chip that processes various protocols such as TCP, UDP, IPv4, ICMP, ARP, IGMP, PPPoE, and etc.

The USB-to-UART interface IC in W7200-EVB can be used for UART communication. And the extension pin header (total of 40 pins) allows the user for easy connection and testing.

W7200-EVB can provide simple example codes based on ANSI C to implement various internet application programs based on W5200. W5200 can be used as a small embedded device in Power down mode to save power consumption.

## Main features;

- W7200 (Cortex-M3 + W5200 core)
- RJ-45 which is integrated transformer.
- USB-to-UART interface IC.
- 40pin expanded header.
- 2 user LEDs, 2 Serial TX/RX LEDs, 1 Power Indicate LED
- Mode S/W, Reset S/W
- Power source: USB BUS power (500mA) or External VIN (5V)

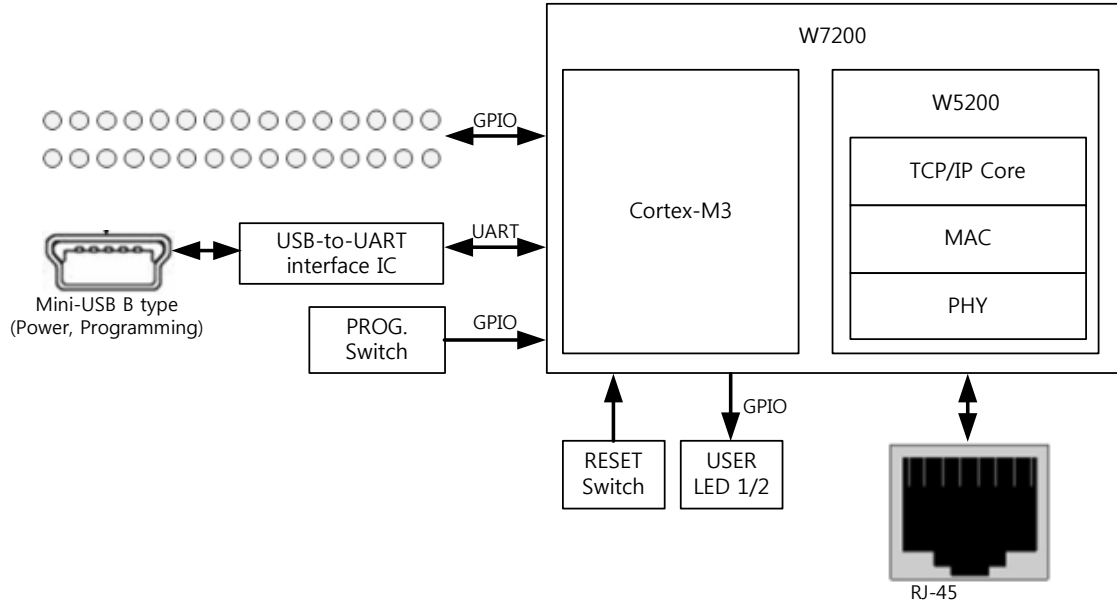
## 2 Specification

**Table 1 List of Items Contained in the EVB**

Item	Description	Remark
MCU	W7200 (STM32F103CB + W5200)	
USB-to-Serial Converter	On board USB-to-UART interface IC, USB bus power	FT232RQ
Ethernet	On board RJ-45 which is integrated transformer	-
LED	User LED 2Ea Serial Status LED 2Ea	-
Button	Reset Switch 1Ea Program Enable Switch 1Ea	-
Expansion Port	MCU port expansion	2.54mm Pitch
PCB	28mm * 52mm Size	-

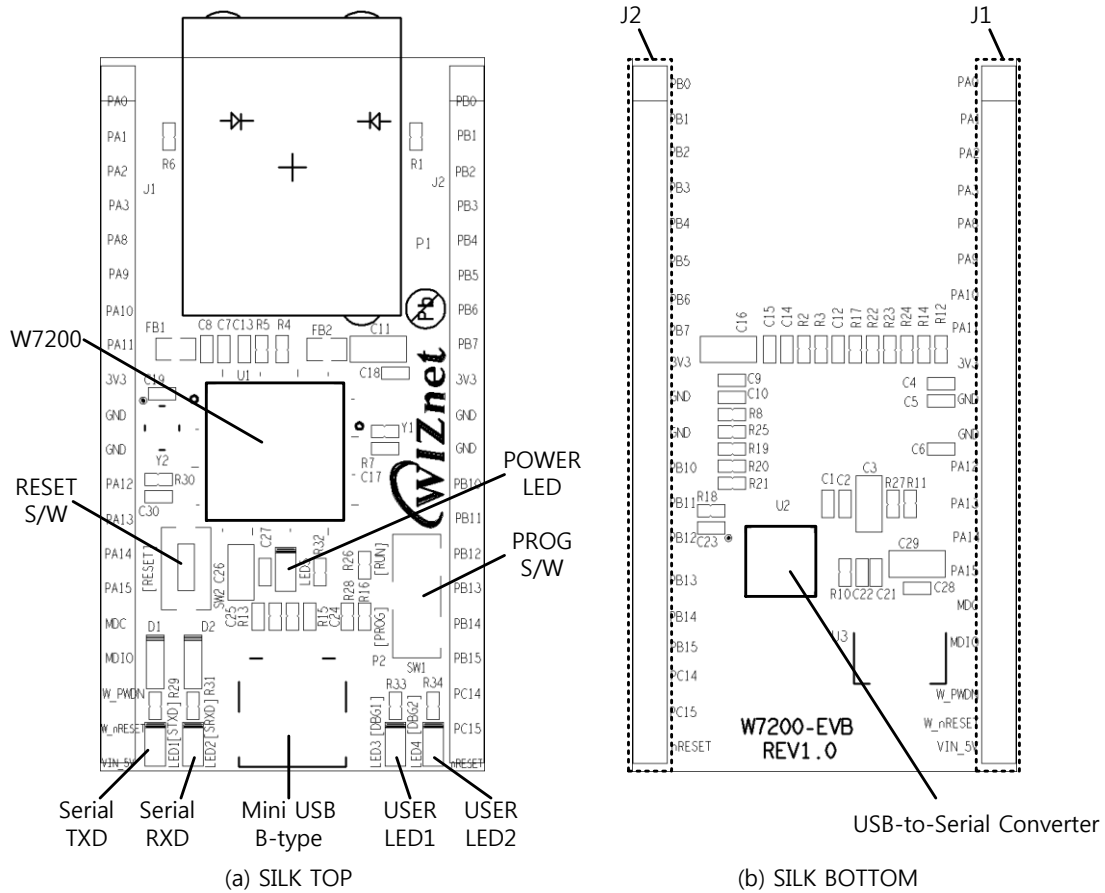
### 3 W7200-EVB Block Diagram

The block diagram is shown below.



**Figure 1 W7200-EVB Block Diagram**

## 4 Hardware Layout



**Figure 2 W7200-EVB Layout**

**Table 2 Hardware Description**

Symbols	Description	Symbols	Description
W5200	Hardwired TCP/IP Core	STM32F103C8	Cortex-M3 MCU
RESET S/W	Reset Switch	User LED1 / User LED2	User LED 2Ea
PROG S/W	Enable Programming Switch - PROG: Program Enable - RUN: User APP. Enable	FT232RQ	USB-to-UART Interface IC
Ethernet Port	RJ-45 ( integrated transformer )	POWER LED	POWER Indicate LED
Serial TXD / Serial RXD	Serial status LED 2Ea	J1 / J2	20 Pin 2.54mm Pitch Expanded Headers
Mini USB B-type	USB Connector		

## 5 Expansion Port Interface

The expansion port has 2.54mm Pitch Pin-Header Hole.

Note.

1. Some of the expanded pin headers are shared by on board peripherals.
2. Refer to W7200's datasheet for more detailed information about alternative functions of pin header

**Table 3 Expanded pin header**

J1	Alternative Functions	Shared by	J2	Alternative Functions	Shared by
PA0	WKUP/ USART2_CTS/ ADC12_IN0/ TIM_CH1_ETR	-	PB0	ADC12_IN8/ TIM3_CH3	LED3
PA1	USART2_RTS/ ADC12_IN1/ TIM_CH2	-	PB1	ADC12_IN9/ TIM3_CH4	LED4
PA2	USART2_TX/ ADC12_IN2/ TIM_CH3	-	PB2	BOOT1	-
PA3	USART2_RX/ ADC12_IN3/ TIM_CH4	-	PB3	JTDO	-
PA8	USART1_CK/ TIM1_CH1/MCO	-	PB4	JNTRST	-
PA9	UART1_TX/ TIM1_CH2	-	PB5	I2C1_SMBAL	-
PA10	UART1_RX/ TIM1_CH3	-	PB6	I2C1_SCL/ TIM4_CH1	-
PA11	UART1_CTS/ CANRX/ USBDM/ TIM1_CH4	-	PB7	I2C1_SDA/ TIM4_CH2	-
3V3	-	-	3V3	-	-
GND	-	-	GND	-	-
GND	-	-	GND	-	-



PA12	UART1_RTS/ CANTX/ USBDP/ TIM1_ETR	-	PB10	I2C2_SCL/ USART3_TX	-
PA13	JTMS/SWDIO	-	PB11	I2C2_SDA/ USART3_RX	-
PA14	JTCK/SWCLK	-	PB12	SPI2_NSS/ I2C2_SMBAL/ USART3_CK/ TIM1_BKIN	-
PA15	JTDI	-	PB13	SPI2_SCK/ USART3_CTS/ TIM1_CH1N	-
MDC	-	-	PB14	SPI2_MISO/ USART3_RTS/ TIM1_CH2N	-
MDIO	-	-	PB15	SPI2_MOSI/ TIM1_CH3N	-
W_PW DN	-	-	PC14	OSC32_IN	-
W_nRE SET	-	-	PC15	OSC32_OUT	-
VIN_5V	External VIN (5V)	-	nRESET	-	-

## 6 Development environment

### 6.1 IDE

The IAR Embedded Workbench for ARM IDE is currently supported. (Other IDE tools for ARM IDE will be supported as like Keil.) The W5200E01-M3 software package is released the version of IAR Embedded Workbench for ARM 5.41. Refer to IAR's own documentation on how to use it.

The W5200E01-M3 software package contains various examples for using W5200

### 6.2 Install Flash loader Demonstrator

Flash Loader demonstrator is used to program for W5200E01-M3.


Note:

Refer to UM0462 User manual at [www.st.com](http://www.st.com) for more detailed information about STM32F103xx Flash Loader demonstrator

- Download: UM0462 Flash loader demonstrator

<http://www.st.com/internet/mcu/product/216817.jsp>

Click "Design Support" -> SW DEMOS (Bottom end of page)

SW DEMOS			
	Description	Version	Size
	STM32F101xx and STM32F103xx Flash loader demonstrator: Contains the Demo GUI, Command line and header source files	2.2.0	7867KB

**Figure 3 Download Flash loader Demonstrator**

### 6.3 USB-to-UART interface IC Driver

When the mini-USB is connect to USB-equipped Windows computer, the Device Manager will properly installed USB-to-Serial driver. If USB-to-Serial adaptor not works as expected, you can download the USB-to-Serial driver at [www.ftdichip.com](http://www.ftdichip.com).

Note:

Refer to Installation Guides at [www.fuducguo.com](http://www.fuducguo.com) more detailed information about USB-to-Serial converter.

- Download Installation Guides:

1. [www.fidichip.com](http://www.fidichip.com)

2. Click "Support->Documents-> Installation Guides"

3. Download up to your operation system.

- Download Driver

1. [www.fidichip.com](http://www.fidichip.com)

2. Click "Drivers->VCP Drivers"

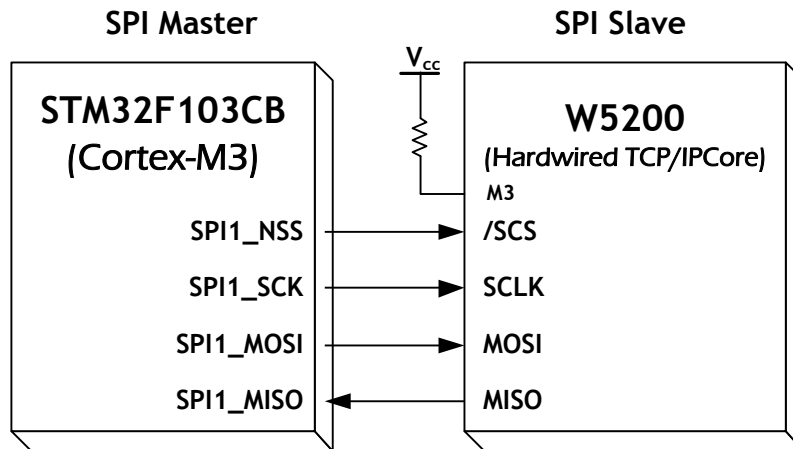
3. Download up to your operation system.

Operating System	Release Date	Processor Architecture						
		x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIV	SH4
Windows*	2011-04-12	2.08.14	2.08.14	-	-	-	-	-
	2011-08-26	<a href="#">2.08.17 (Beta)</a>	<a href="#">2.08.17 (Beta)</a>	-	-	-	-	-
Linux	2009-05-14	1.5.0	1.5.0	-	-	-	-	-
Mac OS X	2011-02-28	2.2.16	2.2.16	2.2.16	-	-	-	-
Windows CE 4.2-5.2**	2010-02-11	1.1.0.6	-	-	1.1.0.6	1.1.0.6	1.1.0.6	1.1.0.6
Windows CE 6.0	2010-02-11	1.1.0.6	-	-	1.1.0.6	1.1.0.6	1.1.0.6	1.1.0.6

**Figure 4 Currently Supported VCP Drivers (16DEC2011)**

## 7 W7200 SPI

Below diagram show internal SPI connection between STM32F103CB and W5200.



**Figure 5 W7200 SPI Interface**

A pseudo code for read/write with SPI is shown below. Check the W5200 documentation for SPI burst mode, and how to use it.

### Code 1 Pseudo code for Read with SPI interface

```
#define data_read_command 0x00
uint16 addr; //address : 16bits
int16 data_len; //data_length :15bits
uint8 data_buf[]; // array for data
SpiSendData(); //send data from MCU to W5200
SpiRecvData(); //Receive data from W5200 to MCU
/* Pseudo Code for Read data of 8bit per packet */
{
    ISR_DISABLE(); // Interrupt Service Routine disable
    CSoff(); // CS=0, SPI start
    //SpiSendData
    SpiSendData(((addr+idx) & 0xFF00) >> 8); //Address byte 1
    SpiSendData((addr+idx) & 0x00FF); //Address byte 2
    // data write command + data length upper 7bits
    SpiSendData((data_read_command | ((data_len & 0x7F00) >> 8)));
    // data length bottom 8bits
    SpiSendData((data_len & 0x00FF));
}
```

```

SpiSendData(0); //dummy data
data_buf[idx] = SpiRecvData(idx); //READ data
CSon(); // CS=1, SPI end
ISR_ENABLE();// Interrupt Service Routine disable
}
    
```

### Code 2 Pseudo code for write with SPI interface

```

#define data_write_command 0x80
uint16 addr; //address : 16bits
int16 data_len; //data_length :15bits
uint8 data_buf[]; // array for data
SpiSendData(); //send data from MCU to W5200
SpiRecvData(); //Receive data from W5200 to MCU
/* Pseudo Code for Read data of 8bit per packet */
{
SpiSendData(); //send data from MCU to W5200
ISR_DISABLE(); // Interrupt Service Routine disable
CSoff(); // CS=0, SPI start
SpiSendData(((addr+idx) & 0xFF00) >> 8); //Address byte 1
SpiSendData((addr+idx) & 0x00FF); //Address byte 2
// data write command + data length upper 7bits
SpiSendData((data_write_command | ((data_len & 0x7F00) >> 8)));
// data length bottom 8bits
SpiSendData((data_len & 0x00FF));
SpiSendData(data_buf[idx]);
CSon(); // CS=1, SPI end
IINCHIP_ISR_ENABLE(); // Interrupt Service Routine disable
}
    
```

## 8 Reference Firmware

- The TCP (Transmission Control Protocol) RFC 793 of IETF
- TCP Server / Client Loopback

The TCP protocol of W5200 supports both server mode and client mode, user can select one and use for its application. The difference between server mode and client mode are shown below.

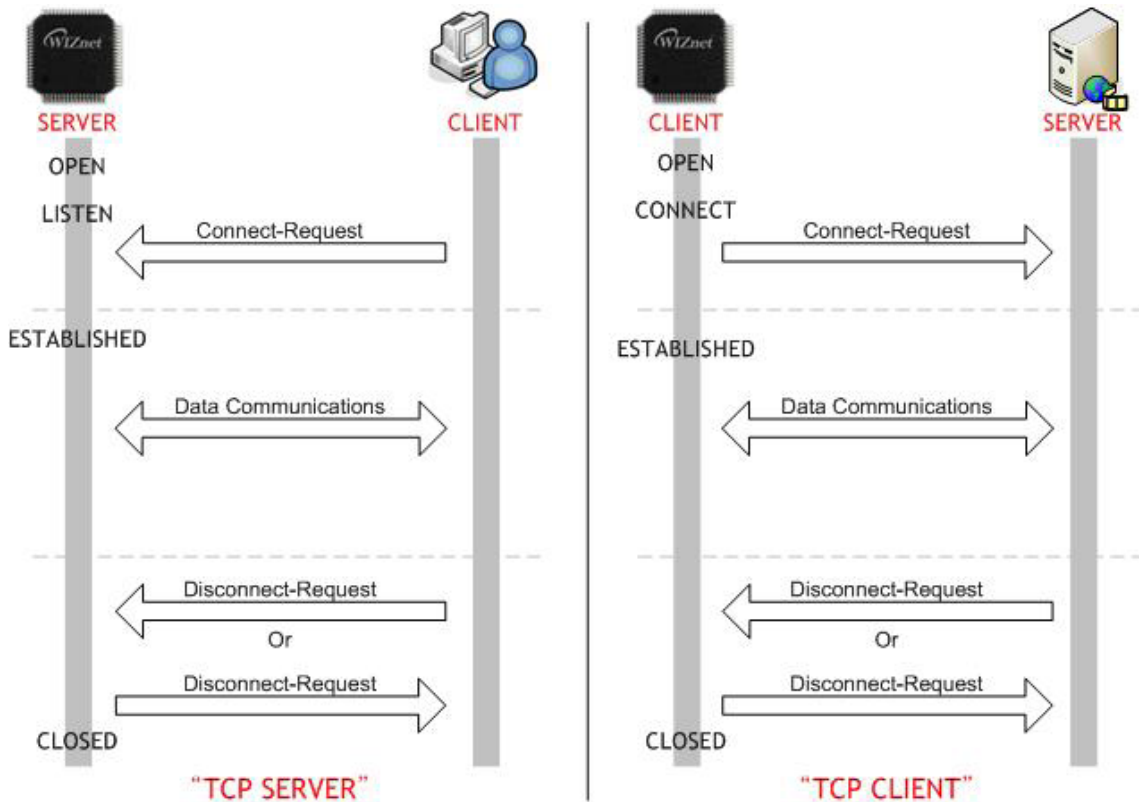


Figure 6 TCP Server / Client

### 8.1 W7200 Socket API

Please refer to "Socket API for ioPlatform" document.

### 8.2 Firmware Structure

Table 4 Project Hierarchy

Directory	Files	Decryption
USER	main.c	W7200-EVB main function
	W5200.c, W5200.h	I/O functions for W5200

	socket.c, socket.h	Socket APIs for W5200
	loopback.c, loopback.h	TCP, UDP Loopback Apps implementation
	SPI1.c	STM32F103x SPI Interface initialization
	util.c, util.h	Utilities
	dhcp.c, dhcp.h	DHCP App implementation
	md5.c, md5.h	md5 hash algorithm implementation for PPPoE
	stm32f10x_it.c	Main Interrupt Service Routines
	system_stm32f10x.c	Cortex-M3 Device Peripheral Access Layer System
CMSIS	core_cm3.c	Cortex™ Microcontroller Software Interface Standard
EWARMv5.4	startup_stm32f10x_md.s	STM32F10x Medium Density Devices vector table
EWARMv5.4/ StdPeriph_Driver	misc.c	miscellaneous firmware functions
	stm32f10x_bkp.c	BKP firmware functions
	stm32f10x_flash.c	FLASH firmware functions
	stm32f10x_gpio.c	GPIO firmware functions
	stm32f10x_rcc.c	RCC firmware functions
	stm32f10x_spi.c	SPI firmware functions
	stm32f10x_tim.c	TIM firmware functions
	stm32f10xusart.c	USART firmware functions

## 8.3 Firmware Functions

**Table 5 Functions in main.c**

Function	Description
RCC_Configuration	Configure the system clocks
NVIC_Configuration	Nested Vectored Interrupt Controller configuration
GPIO_Configuration	Configure the General Purpose I/O Pin
Reset_W5200	W5200 Reset Function
UART1_Init	UART Interface Initialization
WIZ_SPI_Init	W5200 SPI Interface Initialization
Timer_Configuration	Timer Configuration
LED3_onoff/LED4_onoff	USER LED n Control Function
Set_network	Configure Network In formations for W5200
WIZ_Config	Configure Network In formations

**Table 6 Key Variables for network Configuration**

Variable	Description	Example Code (Location: main.c)
MAC[6]	MAC address	MAC[6] = {0x00, 0x08, 0xDC, 0x01, 0x02, 0x03};
IP[4]	Local IP address	IP[4] = {192, 168, 11, 4};
GateWay[4]	Gateway address	GateWay[4] = {192, 168, 11, 1};
SubNet[4]	Sub	SubNet[4] = {255, 255, 255, 0};

- Note : MAC address should be defined even if DHCP mode.

**Table 7 Functions in Loopback.c**

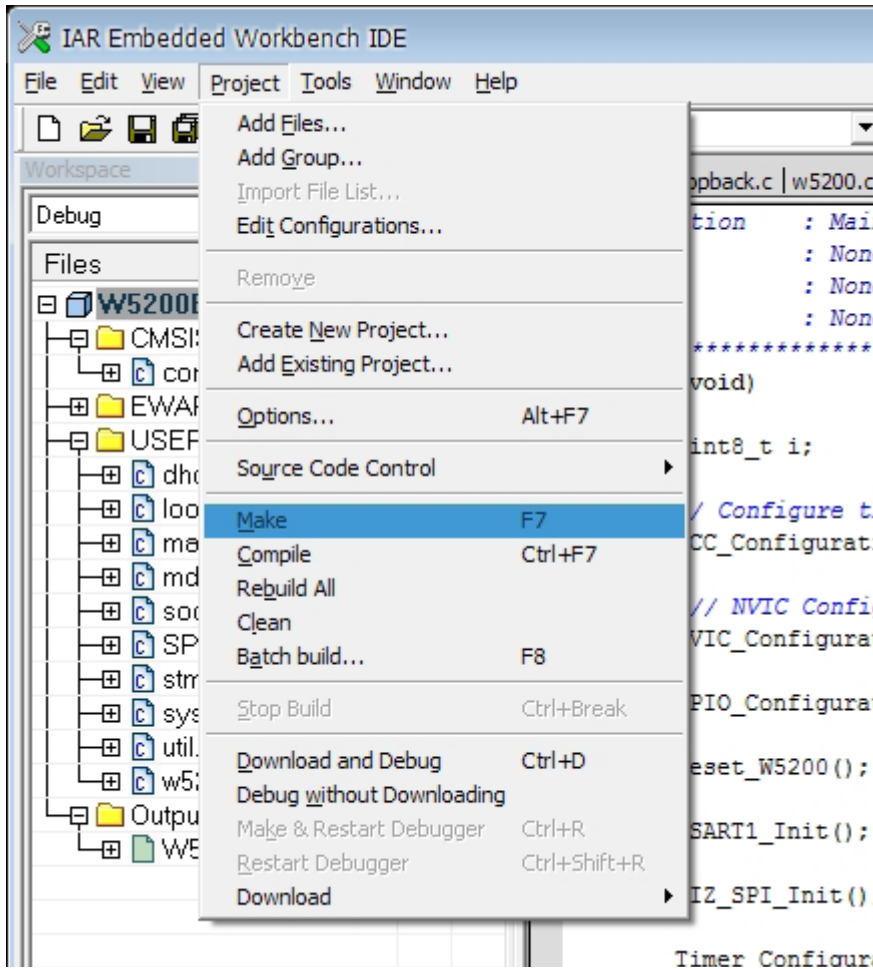
Function	Description	Example Code (Location: loopback.c)
loopback_tcps	TCP Loopback server mode	- ch : socket number [0-7] - port : source port loopback_tcps(uint8 ch, uint16 port)
loopback_tcpclient	TCP Loopback client mode	- ch : socket number [0-7] - ChConfig.destip : Destination IP - ChConfig.port : Destination Port loopback_tcpclient(uint8 ch, CHConfig_TypeDef* ChConfig)
loopback_udp	UDP (User Datagram Protocol) Loopback server mode	- ch : socket number [0-7] - port : source port loopback_udp(uint8 ch, uint16 port)



## 8.4 Firmware Build and Upload

### 8.4.1 Build - IAR Embedded Workbench IDE

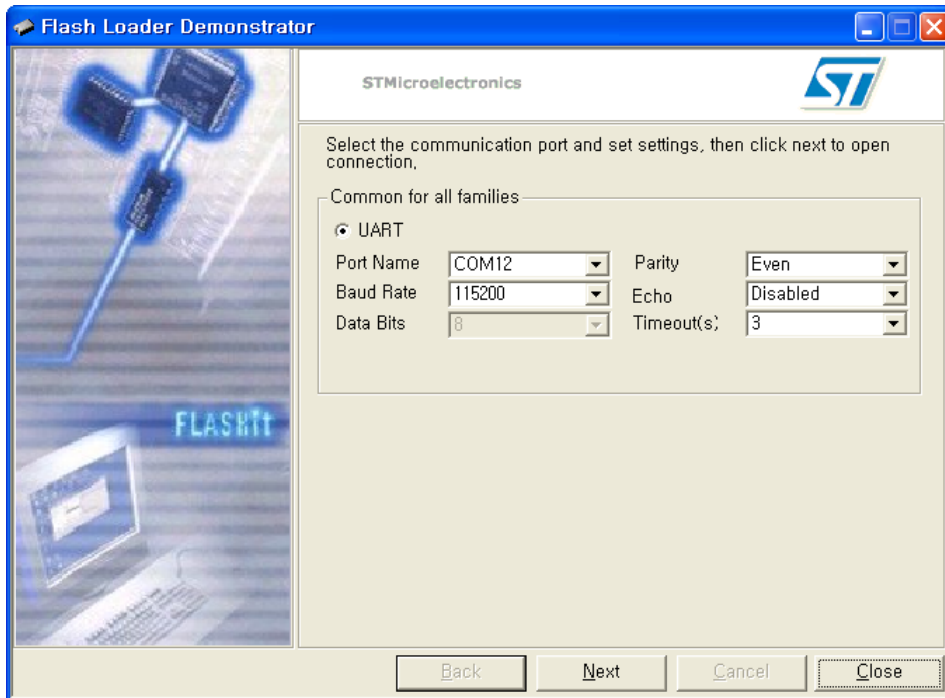
To build and link your project choose "Make" from the "Project" menu, or press F7.



**Figure 7 Compile on IAR Embedded Workbench IDE**

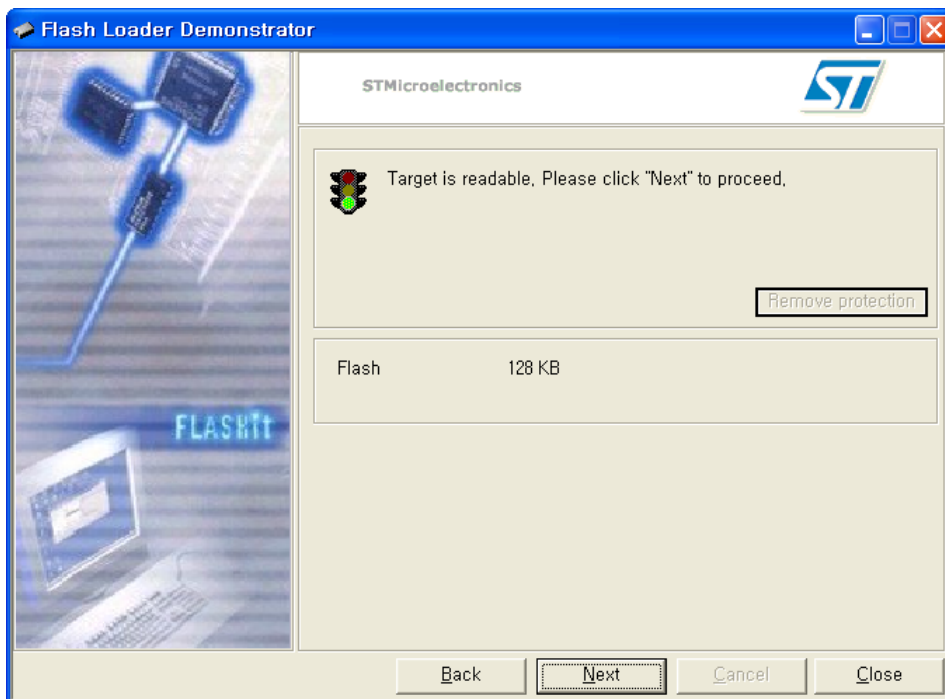
## 8.4.2 Upload - Flash Loader Demonstrator

### Upload (1) Select the Communication port and set setting

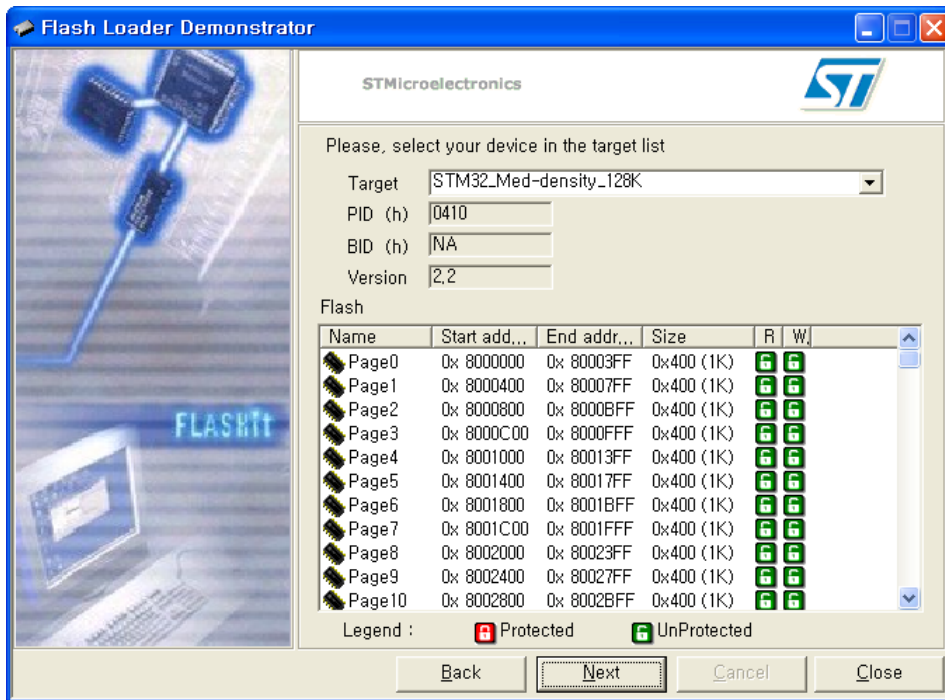


*Note - PROG S/W should be selected 'PROG' to connect to W7200-EVB with PC.*

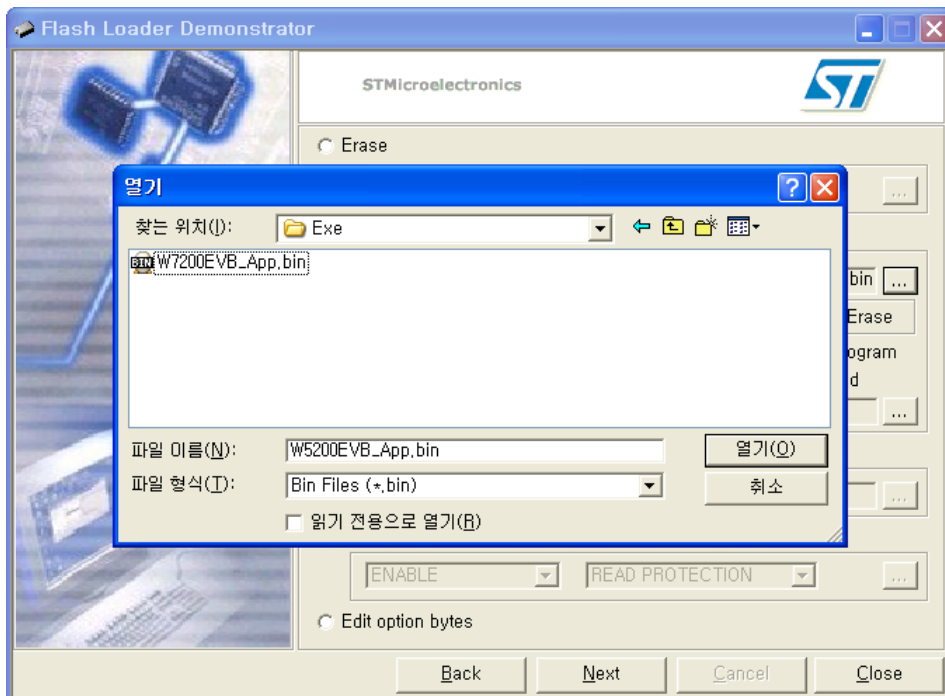
### Upload (2) Check target readable



**Upload (3) Select device in the target**

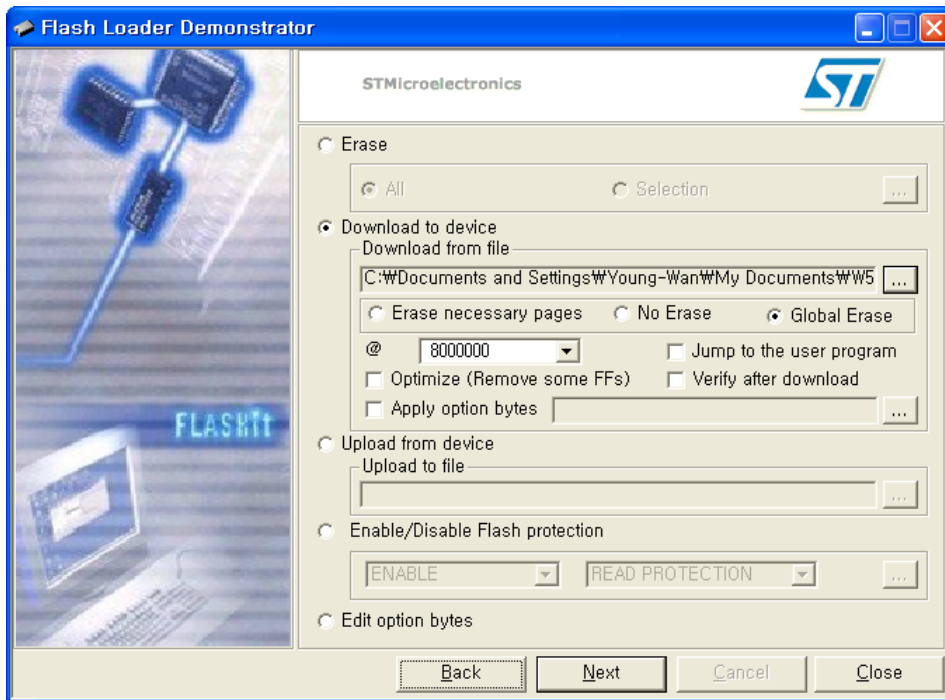


**Upload (4) Choose Binary image file in work project**

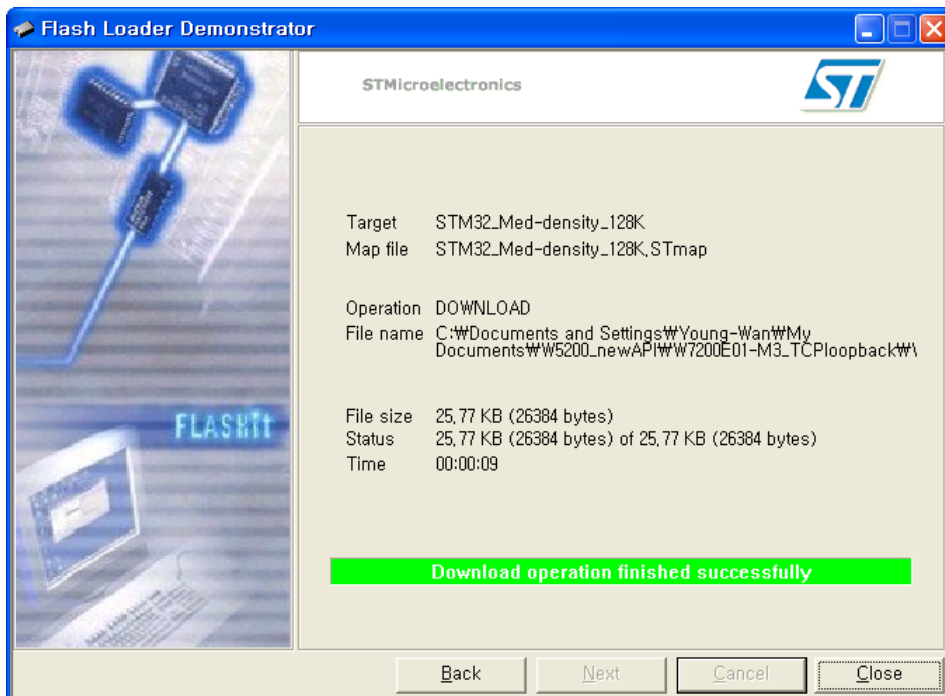


*Note - Binary image file's location: ...\\Work\\App\\Debug\\Exe in project directory*

**Upload (5) Select "@0x08000000" in memory address list**



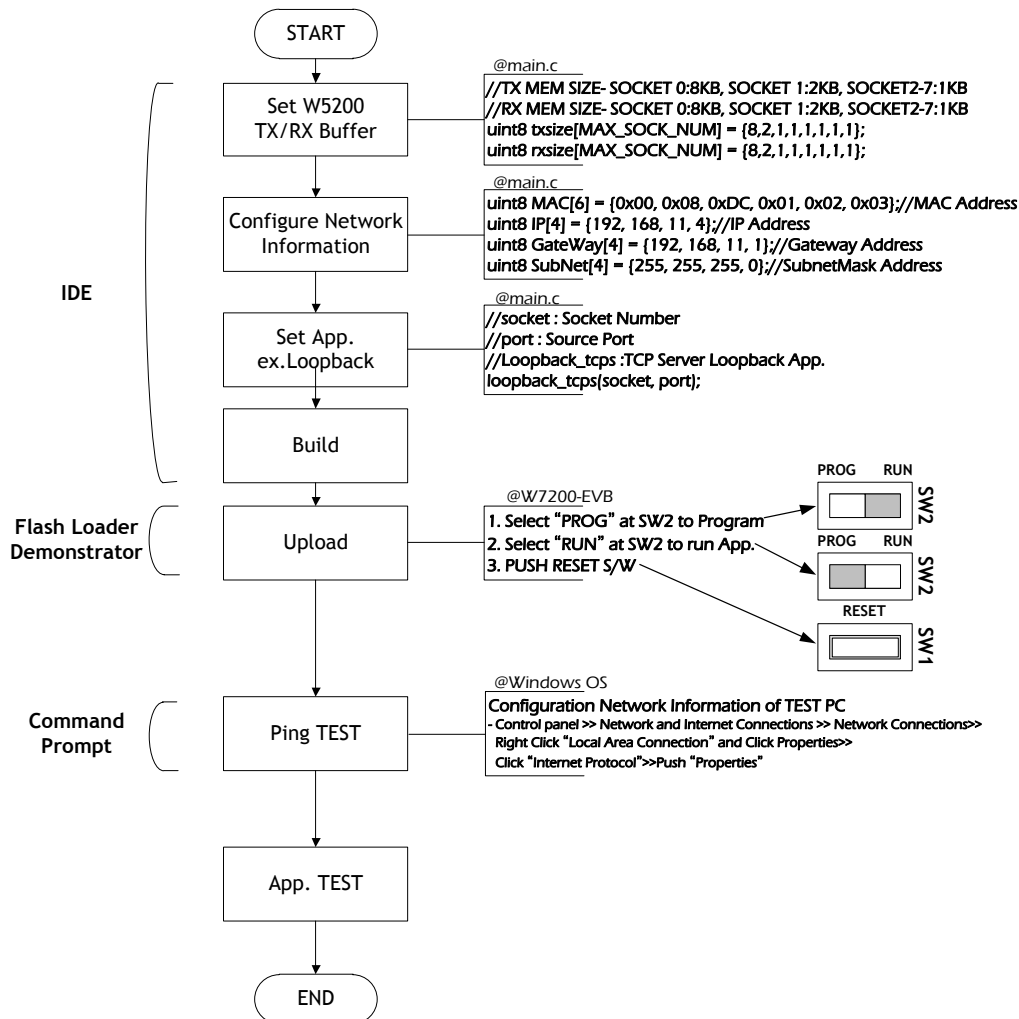
**Upload (6) Select "Next" to upload the binart image file**



**Note - After finishing 'Download', PROG S/W should be selected 'RUN' to run User APP.**

## 9 Application Demonstration

### 9.1 Flow of Demonstration



**Figure 8 Flow Chart of Demonstration**

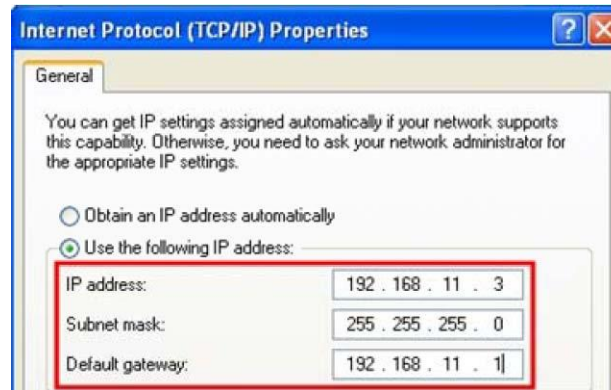
### 9.2 Ping TEST

A ping test determines whether your test PC can communicate with the W7200-EVB over the network.

#### 9.2.1 Network configuration for TEST PC

1. Access the "Start" menu and click "Control Panel."
2. Click "Network Connections"
3. Right-click the name of your network in the list of available networks. Select "Properties."
4. Navigate to the "General" tab. Scroll down through the list of connection types and locate the "Internet Protocol (TCP/IP)" entry.

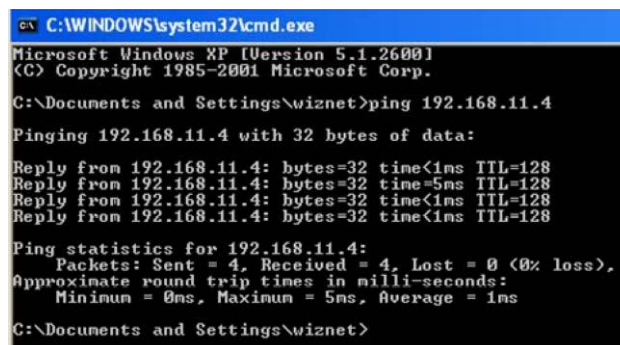
5. Right-click the entry and select "Properties."
6. Configure IP address, subnet mask and Default gateway as follow figure.



**Figure 9 Internet Protocol Properties**

### 9.2.2 Ping Test at Command Prompt

1. Access the "Start" in the menu, click "Run".
2. Enter "cmd" in the "Open:" field,
3. Type "ping 192.168.11.4" (W7200-EVB IP address) in Command Prompt window

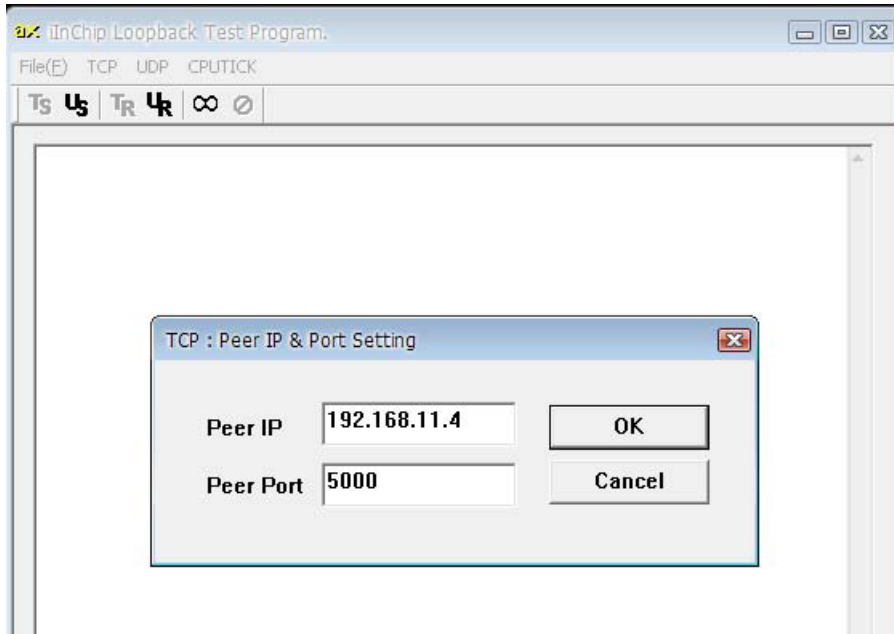


**Figure 10 Ping Test at Command Prompt**

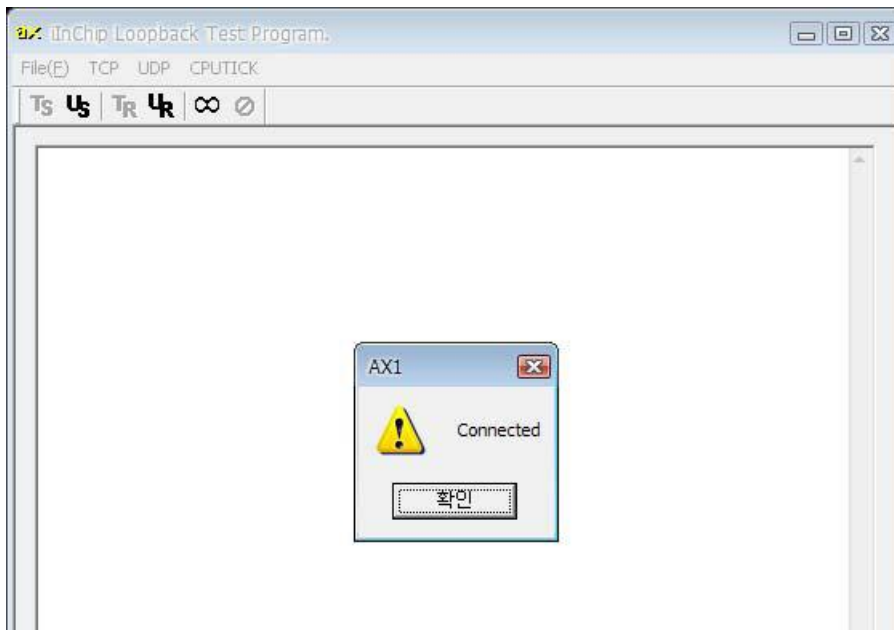
### 9.2.3 App. Test - Loopback TEST

- AX1 : Loopback test program
- Download URL : [http://www.wiznet.co.kr/UpLoad\\_Files/ReferenceFiles/AX1.zip](http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/AX1.zip)
  - AX1.zip : AXInstallV3.1.exe, AX1 Manual V3.1.pdf

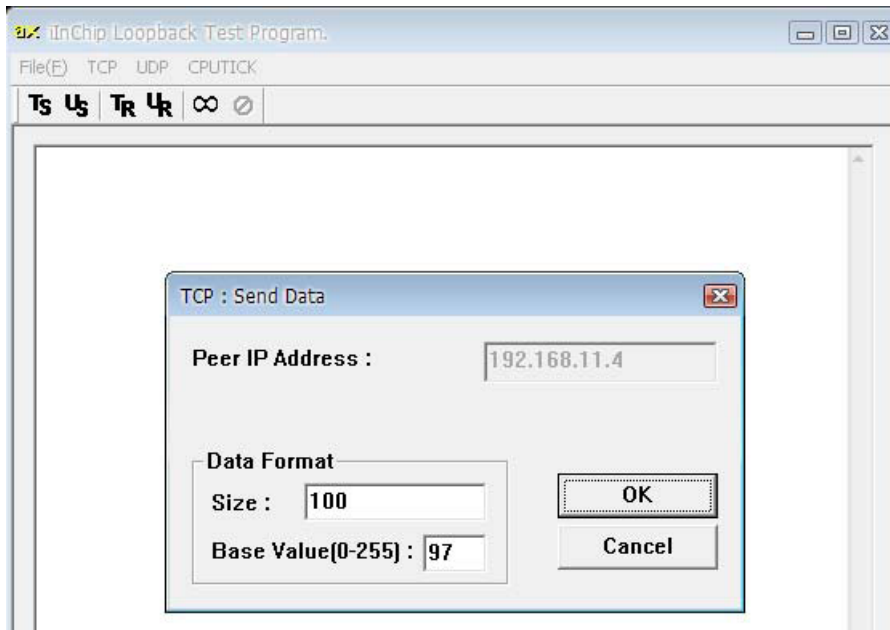
### Run (1) Input W7200-EVB IP and Port



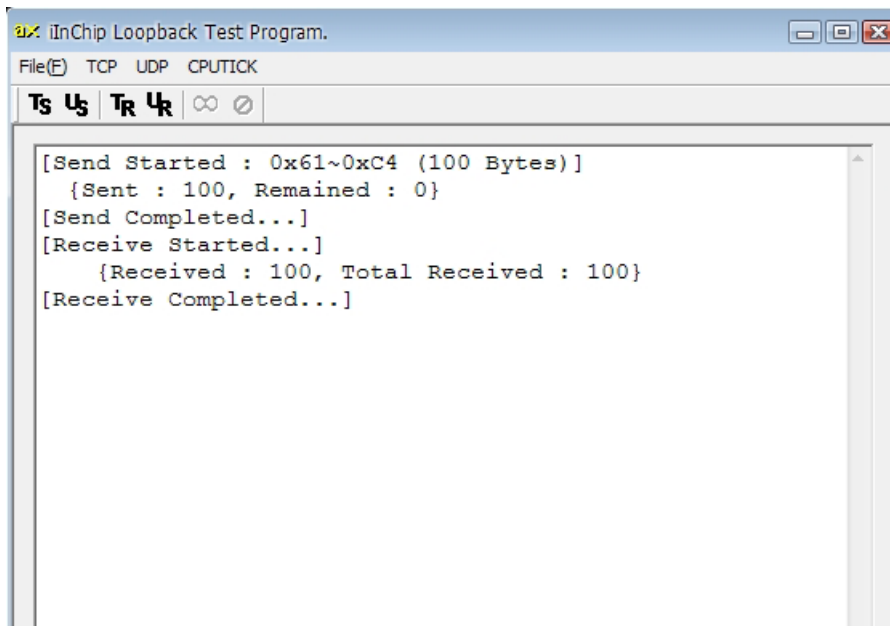
### Run (2) Check "Connected" message



### Run (3) Set Data format; size and base value

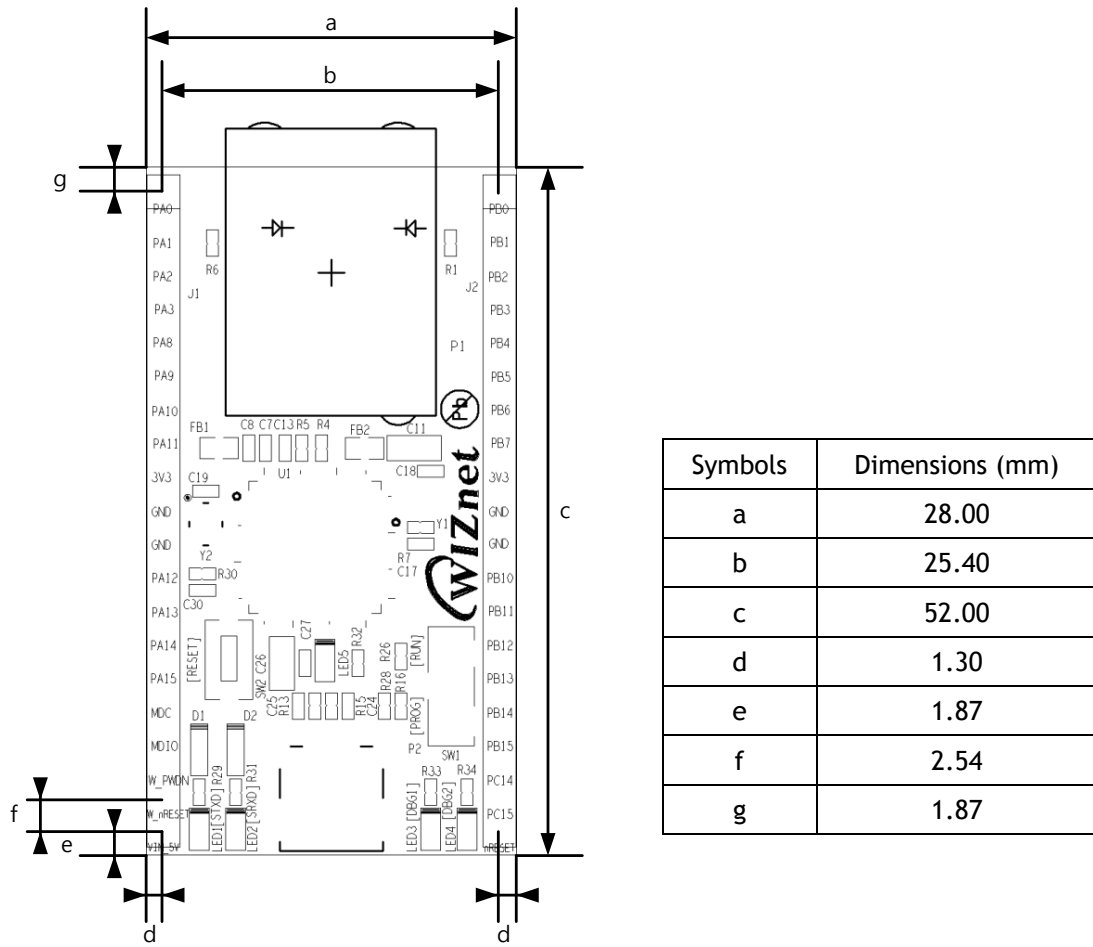


### Run (4) Check the status message in dialog window





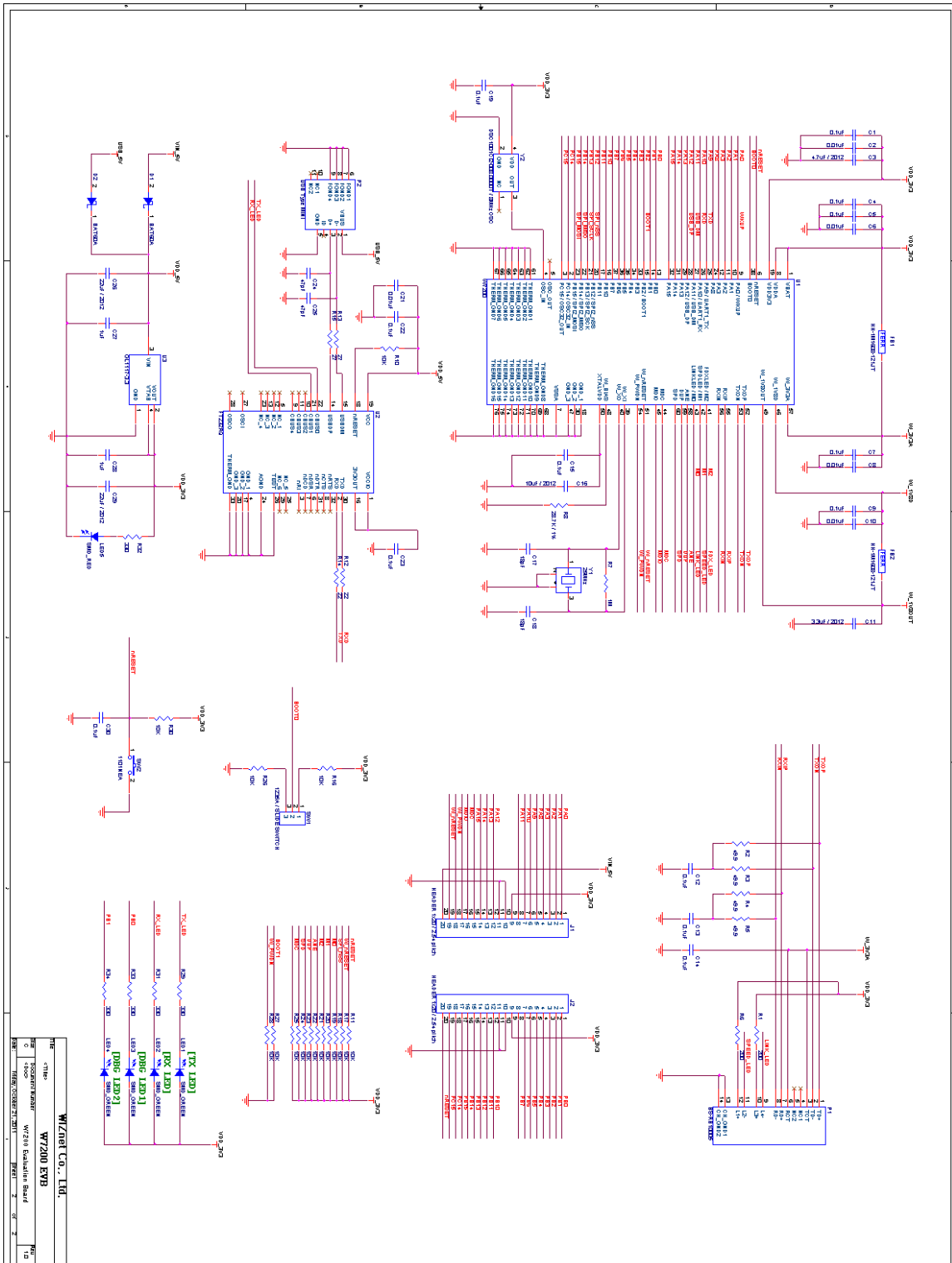
## 10 Physical Specification



**Figure 11 W7200-EVB Board Dimension**

# 11 Schematic

W7200-EVB schematic can be downloading at WIZnet Homepage, [www.wiznet.co.kr](http://www.wiznet.co.kr)



## Document History Information

Version	Date	Descriptions
Ver.1.0	Dec. 2011	Release
Ver.1.1	Dec. 2011	Modify the figures

## Copyright Notice

Copyright 2011 WIZnet, Ltd. All Rights Reserved.

For more information, visit our website at <http://www.wiznet.co.kr>

Technical Support: <http://www.wiznet.co.kr>

Sales & Distribution: [sales@wiznet.co.kr](mailto:sales@wiznet.co.kr)