Migration from W3100A to W5100

This application note describes what designers and programmers should consider when migrating from W3100A to W5100. For additional information about the W5100, please refer to the W5100 datasheet.

This document contains the following topics:

Section 1, "Comparison"

Section 2, "Hardware Considerations"

Section 3, "Software Considerations"

1. Comparison

1.1 Advantages and Challenges of Migration

This section highlights the various advantages and challenges involved when migrating from W3100A to the W5100.

The W5100 provides a higher level of performance while maintaining many characteristics of the W3100A's architecture. Following is a list of the advantages of migrating to the W5100.

- Cost effective.
- Built-in Ethernet PHY
- Provides a higher level of integration.
- Added Tx free size register and Rx received size register. Users can directly read them and do not need to calculate the value any more.
- TCP sequence and ACK number is automatically processed. Users do not need to calculate the values any more.
- New functions (PPPoE/IGMP/SPI Interface/Keepalive, etc).

Following is a list of differences that may present challenges in migrating from the W3100A to the W5100:

- Different pinout. See Section 2, "Hardware Considerations" for information on addressing these differences.
- Library migration. See section 3, "Software considerations" for information on addressing these differences.

1.2 Summary and Feature Comparison tables

Table 1 includes information comparing some of the features of W3100A and W5100.

Table 1. W3100A and W5100 Comparison

		W3100A	W5100
H/W	Core	H/W IPv4	
	Voltage	3.3v	
	H/W Reset	High active	Low active
	Ethernet core Clock	25Mhz	
	Performance (full-duplex)	20Mbps	25Mbps
	Package	Small 64-pin LQFP	Small 80-pin LQFP
	MCU bus I/F	Direct / Indirect / I2C	Direct / Indirect / SPI
	clocked mode for MCU interface	clocked/external/non-clocked mode	Integrated clocked mode
	Tx/Rx memory	16KBytes	
	Socket number	4	
Function	TCP/IP protocol	TCP, UDP, IP, ARP, ICMP, MAC	TCP, UDP, IP, ARP, ICMP, MAC, IGMP, PPPoE
	SEND_OK interrupt	Support	
	No delayed ACK	Support in TCP mode	
	Interrupt register clear	By writing “1”	
	shadow register	Support	Removed
	ACK pointer register	Support	Removed
	RD/WR Pointer register	4 byte registers	2 byte registers
	Receive Data Size / Free Tx Buffer Size register	None	Added
	Keep alive command	None	Added
	Direct send command without ARP	None	Added (SEND_MAC command)

2. H/W consideration

2.1 Voltage

Both W3100A and W5100 is 3.3v device.

2.2 TCP/IP Core

Both W3100A and W5100 use WIZnet's unique H/W IPv4 core. However, W5100 upgraded this core and fixed weak points which are listed in documentation, "W3100A errata and limitation". Users can open this file by clicking the link below,

http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/W3100A_errata_limitation_list1[2][0].pdf

2.3 Package and Pinout

a) The pinout has a huge difference. Two main factors made this difference.

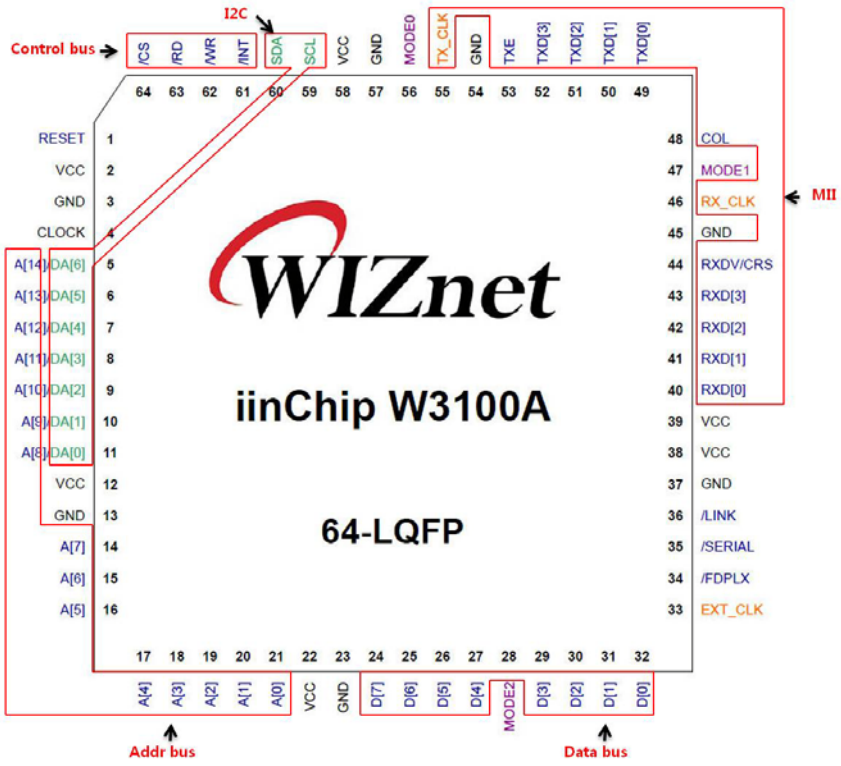
- As PHY is embedded in W5100, MII is replaced by RJ45 interface.
- 4-wire SPI interface replaced I2C interface.

b) Pinout considerations:

There are three parts that users should note when migrate W3100A to W5100. They are MCU interface, PHY interface and 1.8V power source interface. MCU interface consists of Control bus, Address bus, Data bus and I2C (W3100A) or SPI (W5100). PHY interface consists of PHY mode setting interface, MII (W3100A) or RJ45 (W5100) and PHY Status LED. As long as these parts were fully understood, users could easily do the H/W migration from W3100A to W5100.

Fig.1 shows the difference between the pinout of W3100A and pinout of W5100.

Please see the comments in red color.



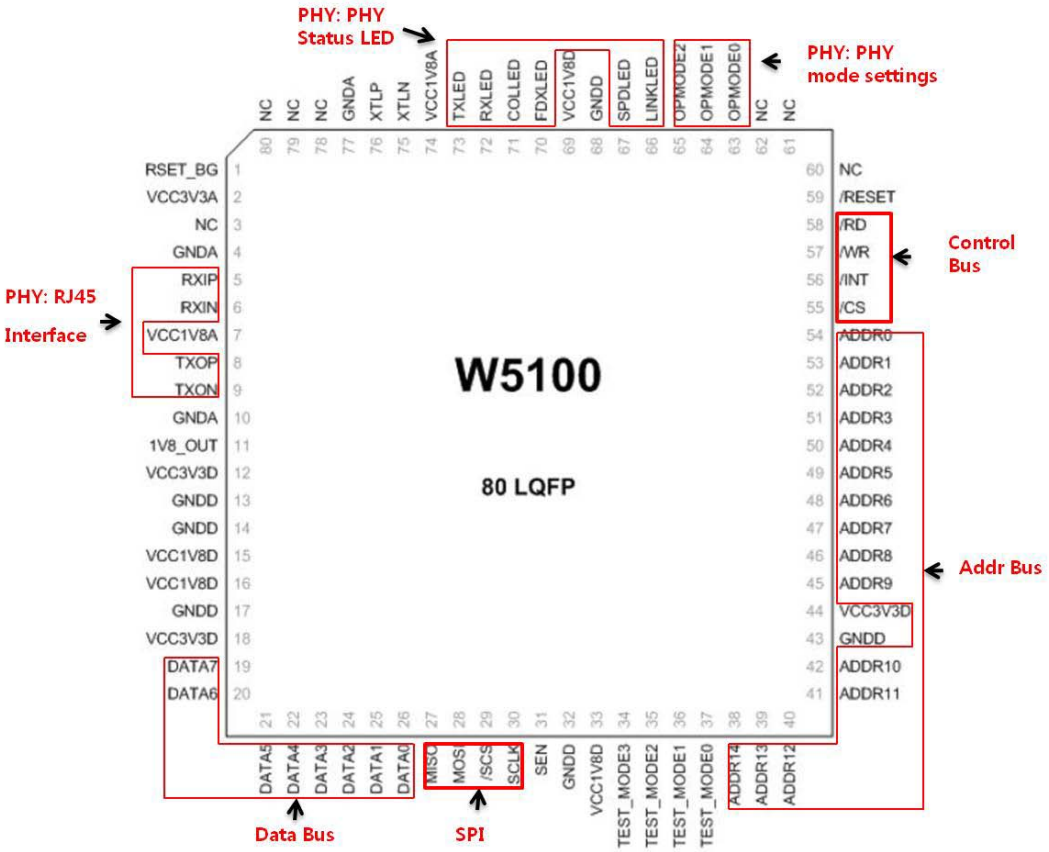


Fig. 1 Pinout considerations when migrate W3100A to W5100

In the pinout of W5100, pin 7, 11, 15, 16, 69 and 74 are 1.8v power source-related pins which W3100A doesn't have. Thus, users should pay more attention to Fig 2 and properly handle these new pins.

Please refer to Fig 2.

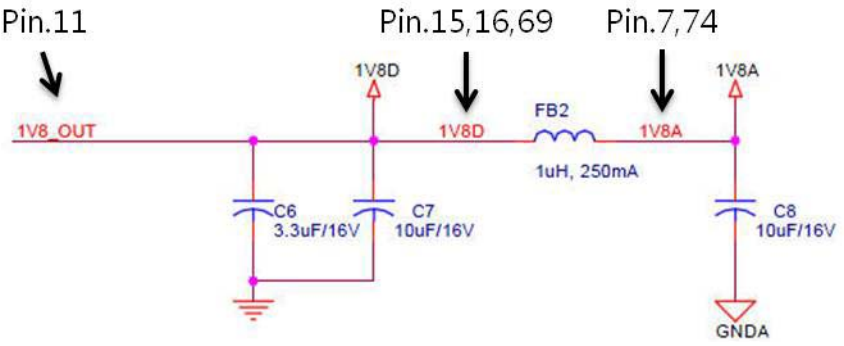


Fig 2. Reference schematic for 1.8v power input

Note: Place C6, C7, FB2 close to 1V8OUT and place C8 close to 1V8A

In terms of PHY mode settings of W5100, Pin.OPMODE0~2 may provide 8 different PHY modes with users. Usually, "auto-negotiation enable with all capabilities" can meet the needs of most applications. That is to say, connect OPMODE0~2 to ground. For more information about PHY mode settings, refer to Chapter 1.2 PHY signals in W5100 datasheet.

2.4 Interface

2.4.1 Host interface

W3100A supports MCU bus I/F and I2C I/F, whereas W5100 supports MCU bus I/F and SPI I/F. Thus,

- If you were using I2C I/F to communicate with your host MCU, you can use your host MCU's SPI to interface with W5100. SPI is a 4-wire interface which is more popular nowadays and easy to use.
- If your host MCU doesn't contain SPI controller, SPI interface can be implemented by GPIO.
- If you were using MCU bus I/F, besides a few H/W modification, all your remaining work is just driver porting that is very simple.

Please see table 2.

Table 2 Host interface considerations

W3100A		H/W	S/W		W5100
I2C	→	redesign	WIZnet driver porting	→	SPI or MCU bus I/F
Clocked	→	In W3100A, Pin 28, 33, 47 and 56 are used to set MCU bus I/F's clocked mode. However, as above mentioned, W5100 removed clocked mode settings as well as these pins.		→	Integrated clocked mode
External clocked	→			→	
Non-clocked	→			→	

2.4.2 MII interface

W3100A supports two kinds of MII. One is a serial MII and the other is a nibble MII; but W5100 already contains Ethernet PHY, so W5100 doesn't support external MII.

2.5 System features

2.5.1 Clock

W3100A may have two clock source pins(Pin #4 and #33). But the W5100 removed an external clocked mode, thus W5100 has only one clock source: 25MHz (No. 75 and 76 pin).

2.5.2 System Reset

Both W3100A and W5100support H/W and S/W reset.

- For H/W reset signal, W3100A is HIGH active, but W5100 is LOW active.
- For S/W reset, both of them implement this function by writing '1' to the 7th bit in certain registers. W3100A is in C0_CR register (offset 0x00). W5100 is in MR register (offset: 0x00).
-

2.5.3 Register write/read timing

W3100A supports six kinds of MCU bus I/F mode including Direct and Indirect, whereas W5100 supports two kinds of MCU bus I/F mode. In this case, write/read timing has a huge difference.

MCU bus I/F	W3100A	W5100
Direct bus	Clocked mode	Integrated clocked mode
	External clocked mode	
	Non-clocked mode	
Indirect bus	Clocked mode	Integrated clocked mode
	External clocked mode	
	Non-clocked mode	
Num of Bus mode	6 bus modes	2 bus modes

When migrating from W3100A to W5100, users should follow W5100's timing. For more information, please refer to W5100's datasheet: Chapter 7 Electrical Specifications (Register/Memory Read timing & Register/Memory Write Timing.

2.5.4 Tx/Rx memory

Both W3100A and W5100 have 16KB Tx/Rx memory. Thus, do not consider the memory limitation when migrating from W3100A to W5100.

2.6 Register map

W3100A consists a Control register, Pointer register, System register, Channel register, and Tx/Rx memory, whereas W5100 consists a Common register, Socket register, and Tx/Rx memory.

Fig.3 shows the Register map of W3100A and W5100.

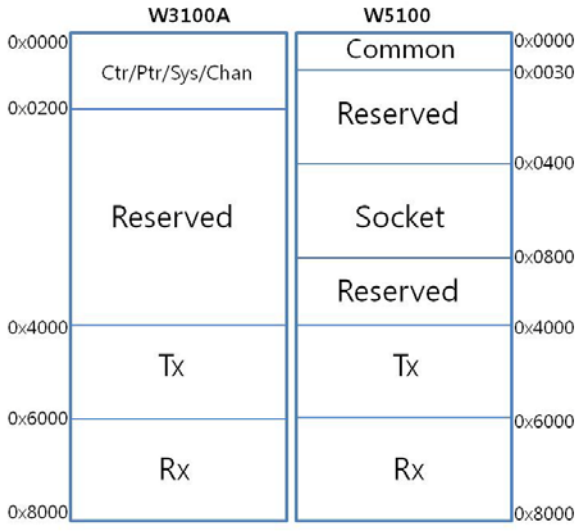


Fig.3 The register map of W3100A and W5100

2.6.1 Function Compatible registers

Table.3 shows the compatible registers. These function compatible registers may have different offset addresses and names in the two chipsets.

Table 3 Function Compatible registers

offset	W3100A	W5100	offset
0x0080~0x0083	GAR		0x0001~0x0004
0x0088~0x008D	SHAR		0x0009~0x000E
0x008E~0x0091	SIPR		0x000F~0x0012
0X0094	RCR		0x0019
0X0095	RMSR		0x001A
0X0096	TMSR		0x001B
0X0084~0X0087	SMR	SUBR	0x0005~0x0008
0X0092~0X0093	IRTR	RTR	0x0017~0x0018
	Cn_DIR	Sn_DIPR	
	Cn_DPR	Sn_DPORT	
	Cn_SPR	Sn_PORT	
	Cn_IPR	Sn_PROTO	
	Cn_TOSR	Sn_TOS	
	Cn_MSSR	Sn_MSR	

2.6.2 Function different registers

IR:

Table.4 IR comparison between W3100A and W5100

	W3100A	W5100
Bit7	C3R	CONFLICT
Bit6	C2R	UNREACH
Bit5	C1R	PPPoE
Bit4	C0R	Reserved
Bit3	C3	S3_INT
Bit2	C2	S2_INT
Bit1	C1	S1_INT
Bit0	C0	S0_INT

When migrating from W3100A to W5100, bit 7~4 cannot be used to even if the WIZnet chip received data or not. Because the function of Bit 7~4 has been replaced by the 2nd bit of Sn_IR(0~3).

IMR:

Table.5 IMR comparison between W3100A and W5100

	W3100A	W5100
Bit7	IMR_C3R	IM_IR7
Bit6	IMR_C2R	IM_IR6
Bit5	IMR_C1R	IM_IR5
Bit4	IMR_C0R	Reserved
Bit3	IMR_C3	IM_IR3
Bit2	IMR_C2	IM_IR2
Bit1	IMR_C1	IM_IR1
Bit0	IMR_C0	IM_IR0

In the IMR register of W5100, the bit 7~4 cannot be used to disable the interrupt, which indicates whether the WIZnet chip received data or not.

Cn_CR/Sn_CR:

Table.6 Command Register comparison between W3100A and W5100

	W3100A		W5100
Bit7	S/W reset (sock 0 only)	→	7th bit in MR
	Memory Test (sock 1 only)	removed	N/A
Bit6	Recv	→	Sn_CR=0x40 (RECV)
Bit5	Send	→	Sn_CR=0x20 (SEND)
Bit4	Close	→	Sn_CR=0x10 (CLOSE)
Bit3	Listen	→	Sn_CR=0x02 (LISTEN)
Bit2	Connect	→	Sn_CR=0x04 (CONNECT)
Bit1	Sock_Init	→	Sn_CR=0x01 (OPEN)
Bit0	Sys_Init (sock 0 only)	removed	N/A

In addition, W5100 added some new commands: DISCON/SEND_MAC/SEND_KEEP.

If users are interested in these commands, please refer to the description of Sn_CR in W5100 datasheet.

Cn_ISR/Sn_IR:

Table.7 Socket Interrupt Register comparison between W3100A and W5100

	W3100A		W5100
Bit7	Reserved	→	Reserved
Bit6	Recv_OK	→	Reserved
Bit5	Send_OK	→	Reserved
Bit4	Timeout	→	SEND_OK
Bit3	Closed	→	TIMEOUT
Bit2	Established	→	RECV
Bit1	SInit_OK	removed	DISCON
Bit0	Init_OK	removed	CON

Bit Recv_OK just reflects the completion of Recv command. It does not indicate whether the WIZnet chip received data or not. However, RECV is the flag that indicates whether the WIZnet chip received data or not.

Bit Established is renamed as CON.

Bit Closed is renamed and upgraded as DISCON.

IDM_OR/MR:

Table.8 Mode Register comparison between W3100A and W5100

	W3100A		W5100
Bit7	IND_EN	→ (bit 0)	RST (S/W reset)
Bit6	Reserved	→	Reserved
Bit5	Reserved	→	Reserved
Bit4	Reserved	→	PB
Bit3	Reserved	→	PPPoE
Bit2	Reserved	→	Reserved
Bit1	L/B	removed	AI
Bit0	AUTO_INC	→ (bit 1)	IND

Bit IND_EN is renamed as IND and placed at bit0 of MR register.

Bit L/B is removed. W5100 only supports big-endian. Thus, if users were using little-endian to access to indirect register, they must use big-endian when migrating from W3100A to W5100.

Bit AUTO_INC is renamed as AI and is placed at bit1 of MR register.

Cn_TW_PR/Sn_TX_WR:

- Size: Cn_TW_PR is a 4-byte register, while Sn_TX_WR is a 2-byte register.
- Function: There are no changes; please see Fig. 4.

Cn_TA_PR/Sn_TX_RD :

- Size: Cn_TA_PR is a 4-byte register, while Sn_TX_RD is a 2-byte register.
- Function: There are no changes; please see Fig.4

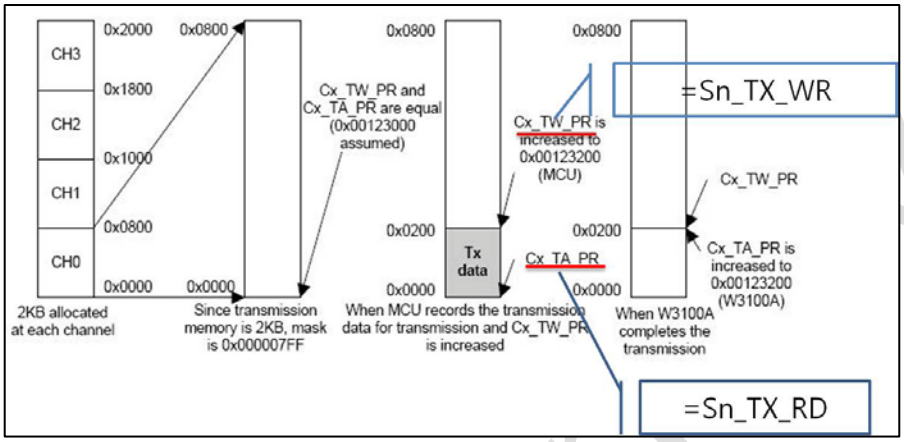


Fig.4 Pointer Management during TCP Transmission

Cn_RW_PR/Sn_RX_WR:

- Size: Cn_RW_PR is a 4-byte register, while Sn_RX_WR is a 2-byte register.
- Function: There are no changes.

Cn_RR_PR/Sn_RX_RD:

- Size: Cn_RR_PR is a 4-byte register, while Sn_RX_RD is a 2-byte register.
- Function: There are no changes.

Cn_SSR/Sn_SR:

Please see the below picture to check the differences of socket status between W3100A and W5100.

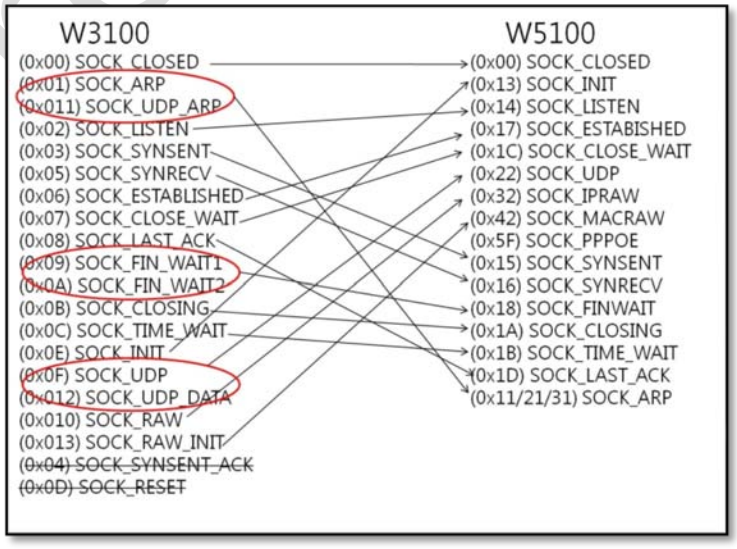


Fig .5 Socket status comparisons

Cn_SOPR/Sn_MR:

Table.9 Socket mode Register comparison between W3100A and W5100 (a)

	W3100A		W5100
Bit7	Broadcast/ERR	removed	MULTI
Bit6	NDTimeout	Timeout(Dynamic/Fixed)→Fixed	Reserved
Bit5	NDAck	→	ND/MC
Bit4	SWS/P	removed	reserved
Bit3	Reserved	See below Table	P3
Bit2	Protocol	See below Table	P2
Bit1	Protocol	See below Table	P1
Bit0	Protocol	See below Table	P0

Table.10 Socket mode Register comparison between W3100A and W5100 (b)

W3100A				W5100			
Bit2	Bit1	Bit0	Meaning	P3	P2	P1	P0
0	0	0	Closed	0	0	0	0
0	0	1	TCP	0	0	0	1
0	1	0	UDP	0	0	1	0
0	1	1	IP_RAW	0	0	1	1
1	0	0	MAC_RAW	0	1	0	0
-	-	-	PPPOE	0	1	0	1

2.6.3 New register and new functions

W5100 has some new registers and new functions:

- PPPOE mode enable bit: 3rd bit in MR register
- PPPOE Authentication register (PATR)
- PPPOE LCP Request Timer register (PTIMEG)
- PPPOE LCP Magic number register (PMAGIC)
- Multicasting enable bit: 7th bit in Sn_MR register
- Multicasting protocol version selection: 5th bit in Sn_MR register
- Unreachable IP address and port register (UIPR and UPORT)
- Socket n TX free size register (Sn_TX_FSR)
- Socket n RX received size register (Sn_RX_RSR)
- Socket n TTL register (Sn_TTL)

If users are interested in these registers and functions, please refer to W5100 datasheet.

This documentation mainly focuses on the migration from the W3100A to W5100.

3. S/W consideration

3.1 S/W migration

Essentially, S/W migration refers to driver migration, which is downloadable through WIZnet website. Thus, for each WIZnet chip, S/W migration is very simple:

3.1.1 How to migrate

<Step1>: Remove W3100A driver

<Step2>: Download W5100 driver

<Step3>: Porting W5100 driver to your host MCU

<Step4>: List up all W3100A's API that your application is using

<Step5>: List up all W5100's API which has the same functions with W3100A's API (See Table 11)

<Step6>: Replace W3100A's API with W5100's API one by one.

<Step7>: Completed

Table.11 Primary APIs comparison between W3100A and W5100

Functions	W3100A	W5100
Initialization of Network	InitW3100A()	iinchip_init();
	setMACAddr()	setSHAR()
	setIP()	setSIPR()
	setgateway()	setGAR()
	setsubmask()	setSUBR()
	getsubmask()	getSUBR()
	GetIPAddress()	getSIPR
	GetGWAddress()	getGAR()
Open Sockets	socket() The symbol of 2nd parameter of function socket is different.	socket()
	SOCK_STREAM	Sn_MR_TCP
	SOCK_CLOSEDM	Sn_MR_CLOSE
	SOCK_DGRAM	Sn_MR_UDP
	SOCK_IPL_RAW	Sn_MR_IPRAW
	SOCK_MACL_RAW	Sn_MR_MACRAW
Basic socket APIs	Listen()	Listen()
	Connect()	Connect()
	Close()	Close()
Socket status	select(i,SEL_Control)	getSn_SR(i)
Socket Tx free size	Select(i,SEL_SEND)	getSn_TX_FSR()
Socket Rx received data size	Select(i,SEL_RECV)	getSn_RX_RSR()

Data transmission	send()/sendto()	send()/sendto()
Data receive	recv()/recvfrom()	recv()/recvfrom()
Check IR	INT_REG	iichip_read(IR)
Check socket IR	INT_REG&0x01(check socket 0)	iichip_read(IR)&IR_SOCK(0)
	INT_REG&0x02(check socket 1)	iichip_read(IR)&IR_SOCK(1)
	INT_REG&0x04(check socket 2)	iichip_read(IR)&IR_SOCK(2)
	INT_REG&0x08(check socket 3)	iichip_read(IR)&IR_SOCK(3)
Check socket INT name	I_STATUS[i]&SESTABLISHED	getSn_IR(i)&Sn_IR_CON
	I_STATUS[i]&SCLOSED	getSn_IR(i)&Sn_IR_DISCON
	I_STATUS[i]&SSEND_OK	getSn_IR(i)& Sn_IR_SEND_OK
	I_STATUS[i]&STIMEOUT	getSn_IR(i)& Sn_IR_TIMEOUT
	INT_REG&0x10	getSn_IR(0)&Sn_IR_RECV
	INT_REG&0x20	getSn_IR(1)&Sn_IR_RECV
	INT_REG&0x40	getSn_IR(2)&Sn_IR_RECV
	INT_REG&0x80	getSn_IR(3)&Sn_IR_RECV
Clear socket IR flag	Direct value assignment Ex: clear SEND_OK flag INT_STATUS(0)=0x20	By iichip_write(): Ex: clear SEND_OK flag iichip_write(Sn_IR(0),0x10)
Clear IR flag	Direct value assignment Ex: INT_REG=0XFF	By iichip_write(): Ex: iichip_write(IR,0xFF)

3.1.2 Where to download WIZnet driver

W3100A libraries:

http://www.wiznet.co.kr/W3100A-LF/download

W5100 libraries:

http://www.wiznet.co.kr/W5100/download