

## Migration from W3100A to W3150A+

This application note describes what designers and programmers should consider when migrating from W3100A to W3150A+. For additional information about the W3150A+, please refer to the W3150A+ 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 W3150A+. The W3150A+ 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 W3150A+.

- Cost effective.
- 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).
- Standard driver library for future W5100 migration. The driver library of W3150A+ and W5100 is fully compatible. Thus, users do not need to consider firmware change after migrating from W3150A+ to W5100.

Following is a list of differences that may present challenges in migrating from the W3100A to the W3150A+:

- Different pinout. See Section 2, "Hardware Considerations" for information on addressing these differences.
- Library migration. There is a huge difference between the driver library of W3100A and W3150A+. 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 W3150A+.

**Table 1. W3100A and W3150A+ Comparison**

		W3100A	W3150A+
H/W	Core	H/W IPv4	H/W IPv4
	Voltage	3.3v	3.3v
	H/W Reset	High active	Low active
	Ethernet core Clock	25Mhz	25MHz
	MCU bus I/F Clock	25MHz or>25MHz	25MHz
	Performance (full-duplex)	20Mbps	25Mbps
	Package	64-pin LQFP	64-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	16k byte	16k byte
	Socket number	4	4
	MII	Serial and Nibble	Nibble
Function	TCP/IP protocol	TCP.UDP.IP.ARP.ICMP.MAC	TCP.UDP.IP.ARP.ICMP.MAC. IGMP(UDP Multicasting) PPPoE(ADSL)
	SEND_OK interrupt	Support	Support
	No delayed ACK	Support in TCP mode	Support in TCP mode
	Interrupt register clear	By writing "1"	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 models are 3.3v device.

### 2.2 TCP/IP Core

Both models use WIZnet's unique H/W IPv4 core. However, W3150A+ 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/W3100A-LF/download/>

### 2.3 Package and Pinout

- Both models are 64-pin LQFP
- Pinout considerations:

Fig.1 shows the difference between the pinout of W3100A and pinout of W3150A+.

Please see the comments in red color.

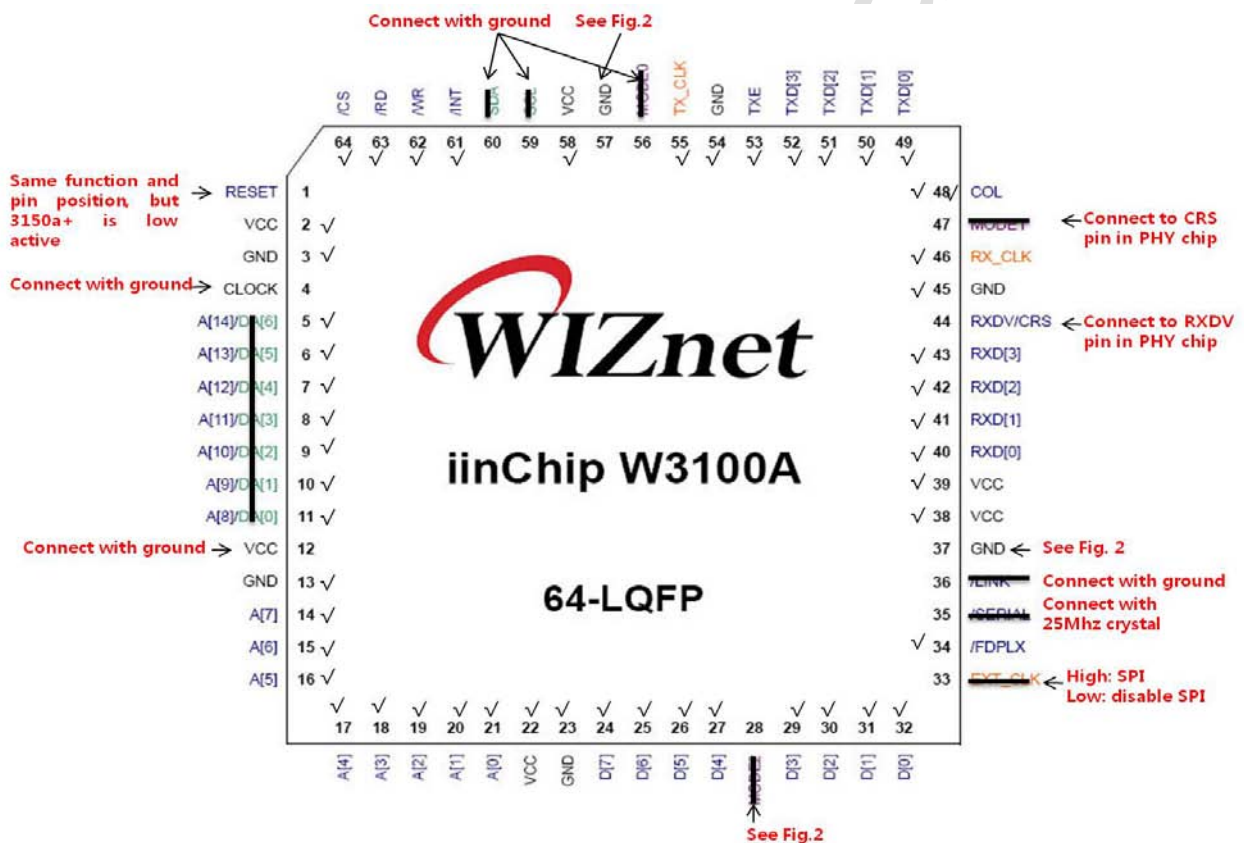


Fig. 1 Pinout considerations when migrate W3100A to W3150A+

- "  " means "removed function"
- " / " means "no change",
- " ←+comments " means "the changed part"

In the pinout of W3150A+, pin 28, 37, and 57 are 1.8v power source-related pins.

Please refer to Fig 2.

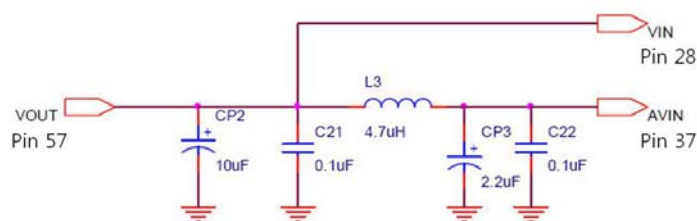


Fig 2. Reference schematic for 1.8v power input

## 2.4 Interface

### 2.4.1 Host interface

W3100A supports MCU bus I/F and I2C I/F, whereas W3150A+ 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 W3150A+. SPI is a 4-wire interface which is more popular nowadays and easy to use. SPI interface also 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	F/W		W3150A+
I2C	→	redesign	WIZnet driver porting	→	MCU bus I/F or SPI
Clocked	→	Refer to comments in pin 56,47,28,33 of Fig.1		→	Non-clocked
External clocked	→			→	
Non-clocked	→			→	

### 2.4.2 MII interface

The W3100A supports two kinds of MII. One is a serial MII and the other is a nibble MII; but the W3150A+ only supports nibble MII. Thus, if you were using serial MII mode, you have to adjust your PHY chip to the nibble type.

In addition, W3150A+ has two independent pins: RXDV and CRS. Users should connect pin RXDV (No. 44) to pin RXDV (PHY chip side), and pin CRS (No.47) to pin CRS (PHY chip side).

## 2.5 System features

### 2.5.1 Clock

The W3100A may have two clock pins(Pin #4 and #33). But the W3150A+ removed the external clocked mode. Thus, W3150A+ has only one clock source: 25MHz (No. 35 pin).

## 2.5.2 System Reset

Both models support H/W and S/W reset.

- For H/W reset signal, W3100A is HIGH active, but W3150A+ is LOW active.
- For S/W reset, both of them implement this function by writing '1' to the 7<sup>th</sup> bit in certain registers. W3100A is in CO\_CR register (offset 0x00). W3150A+ is in MR register (offset: 0x00).

## 2.5.3 Register write/read timing

The W3100A supports six kinds of MCU bus I/F mode including Direct and Indirect, whereas W3150A+ supports two kinds of MCU bus I/F mode. In this case, write/read timing has a huge difference.

When migrating from W3100A to W3150A+, users should follow W3150A+'s timing. For more information, please refer to W3150A+'s datasheet: 7.4.2 Register/Memory Read timing & 7.4.3 Register/Memory Write Timing.

## 2.5.4 Tx/Rx memory

Both models have 16KBytes Tx/Rx memory. Thus, do not consider the memory limitation when migrating from W3100A to W3150A+.

## 2.6 Register map

W3100A consists a Control register, Pointer register, System register, Channel register, and Tx/Rx memory, whereas W3150A+ consists a Common register, Socket register, and Tx/Rx memory.

Fig.3 shows the Register map of W3100A and W3150A+.

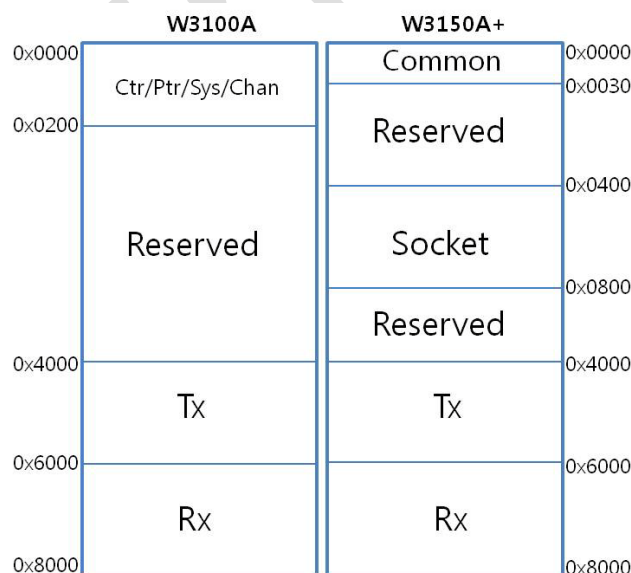


Fig.3 The register map of W3100A and W3150A+

### 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	W3150A+	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 W3150A+**

	W3100A	W3150A+
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 W3150A+, bit 7-4 cannot be used even if the WIZnet chip received data or not. The function of Bit 7-4 has been replaced by the 2<sup>nd</sup> bit of Sn\_IR(0-3).

IMR:

Table.5 IMR comparison between W3100A and W3150A+

	W3100A	W3150A+
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 W3150A+, 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 W3150A+

	W3100A		W3150A+
Bit7	S/W reset (sock 0 only )	→	7 <sup>th</sup> 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, W3150a+ 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 W3150A+ datasheet.

Cn\_ISR/Sn\_IR:

**Table.7 Socket Interrupt Register comparison between W3100A and W3150A+**

	W3100A		W3150A+
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 W3150A+**

	W3100A		W3150A+
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. W3150A+ 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 W3150A+.

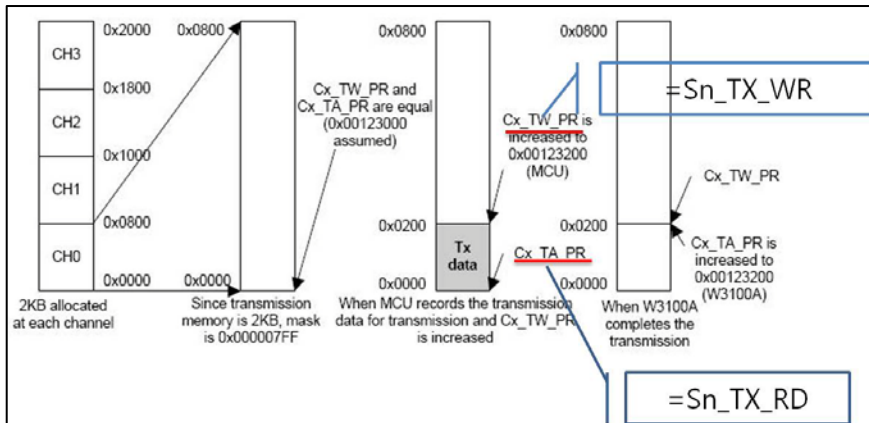
Bit AUTO\_INC is renamed as AI and is placed at bit1 of MR register.

**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:**

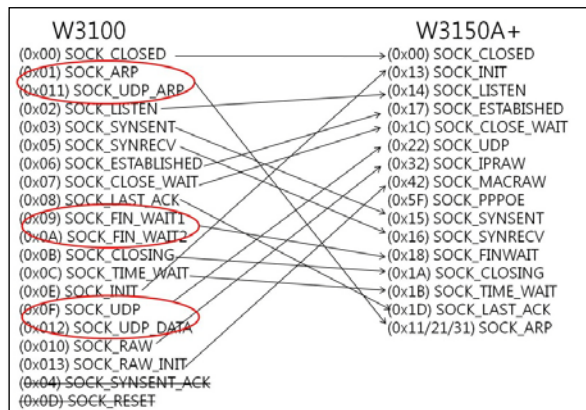
- Size: Cn\_RW\_PR is a 4-byte register, while Sn\_RX\_WR is a 2-byte register.
- Function: There are no changes.

**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 W3150A+.



**Fig .5 Socket status comparisons**

Cn\_SOPR/Sn\_MR:

**Table.9 Socket mode Register comparison between W3100A and W3150A+ (a)**

	W3100A		W3150A+
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 W3150A+ (b)**

W3100A				W3150A+			
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

W3150A+ has some new registers and new functions:

- PPPoE mode enable bit: 3<sup>rd</sup> bit in MR register
- PPPoE Authentication register (PATR)
- PPPoE LCP Request Timer register (PTIMEG)
- PPPoE LCP Magic number register (PMAGIC)
- Multicasting enable bit: 7<sup>th</sup> bit in Sn\_MR register
- Multicasting protocol version selection: 5<sup>th</sup> 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 W3150A+ datasheet.

This documentation mainly focuses on the migration from the W3100A to W3150A+.

### 3. S/W consideration

#### 3.1 Software library

Both W3100A and W3150A+ provide users with drivers. By using the API functions in the driver, users may conveniently develop their application.

In this section, we focus on the driver migration from the W3100A to W3150A+.

##### 3.1.1 How to migrate

Users should update driver files first, and then change old API functions (those in application-related files, such as main.c, dhcp.c, dns.c, http.c and target\_application.c) into new API functions one by one.

#### 1> Driver updating

Remove all W3100A driver files and replace them by W3150A+driver.

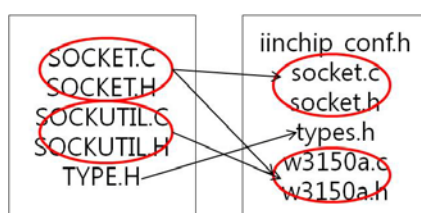


Fig.6 Driver source tree comparison

iinchip\_conf.h is a new added header in the W3150A+ driver. It defines the system options for W3150A+ driver, such as the MACRO for direct mode or indirect mode or SPI mode and etc. W3100A had three different drivers for three different modes, whereas W3150A+ has a higher level integration driver built in.

#### 2> Primary API functions updating

Table.11 Primary APIs comparison between W3100A and W3150A+

Functions	W3100A	W3150A+
Initialization of Network	InitW3100AA()	iinchip_init();
	setMACAddr()	setSHAR()
	setIP()	setSIPR()
	setgateway()	setGAR()
	setsubmask()	setSUBR()
	getsubmask()	getSUBR()
	GetIPAddress()	getSIPR
	GetGWAddress()	getGAR()
Open Sockets	socket() The symbol of 2 <sup>nd</sup> 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()
Disable external INT	EX0=0 (Note: W3100AA is 8051 external INTO based/ W3150A+ is Atmel128 external INT4 based)	IINCHIP_ISR_DISABLE()
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]&SEND_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	By iichip_write():

	Ex: clear SEND_OK flag INT_STATUS(0)=0x20	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)
Enable external INT	EX0=1	IINCHIP_ISR_ENABLE()

### 3.1.2 Where to download library

W3100A library and errata:

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

W3150A+ library and errata:

<http://www.wiznet.co.kr/W3150A+/download>

**Note: S/W for W5100 and W3150A+ is 100% compatible.**

**Developers only need to change their H/W design.**

W5100 library and errata:

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

Preliminary