

# **STANDALONE SpaceWire to Gigabit Ether H/W specification**

Rev1.60

SHIMAFUJI Electric Inc  
KC Bldg.5F, 8-1-15, Nishikamata, Ota-ku,  
Tokyo, Japan 144-00051  
TEL: +81-3-3733-8308  
FAX: +81-3-3733-8318  
E-mail [info@shimafuji.co.jp](mailto:info@shimafuji.co.jp)  
URL <http://www.shimafuji.co.jp>

Document No. EN210439003-0  
Date Published July. , 2012  
SHIMAFUJI Electric Inc  
Printed in Japan



## **Notices on use and handling**

**You should follow those remarks to use this product safely. If you are not following those remarks, you may cause electrical shock, injury, fire or trouble.**



**Lightning**

**While lightning has occurred , never installing the product or connecting cables, it may cause an electrical shock by the thunderbolt.**



**Handle with care**

**Does not either drop, hit or give a strong shock to the product.**



**Caution to electrostatic discharge**

**This product mounted electrostatic sensitive parts.**

**The parts are possibly destroyed by electrostatic discharge; do not touch directly to contact area of the connectors and the parts.**



**Caution to connect or dis-connect cables**

**First turn off the power to this product to connect connectors on board and cables. If connecting or dis-connecting cables to this board without turn off the power, it may destroy this board and connected product.**



**Pay attention to touch this product**

**While the product is turned on or straight after the operation, it may cause an electrical shock or scald. (Some parts become higher temp.)**



### **Unplug the Power source**

**Immediately unplug from the Power source when it smells or smokes. If continually keep supply power on while it smells or smokes, it may cause fire, an electrical shock or serious influence on this board and other equipments.**



### **Do not use or store in the following places.**

- Do not expose in direct sunlight**
- Do not place where the temperature changes rapidly and wets with dew.**
- Do not expose to rain or moisture.**
- Do not place rolled or vibrated.**
- Do not place dusty or carpet laid places cause electrostatic obstacles.**
- Do not place where corrosive gas outbreaks.**
- Do not directly place this product on the electro conductive materials (it may cause trouble)**



### **Remark on operation of this product**

**The maximum operating temperature of this Microprocessor (EC-4350) is 80 degrees Celsius on case; it has to operate under this temperature (it may need air cooling system to operate in high temp.)**

Revision History

Version	Date	Contents	Note
Rev0.0	2012/07/18	New Release	

## Contents

---

1.	OVERVIEW	1
2.	SPECIFICATION	1
3.	BLOCK DIAGRAM	2
4.	FUNCTION	3
4.1.	SPACEWIRE ROUTER	3
4.2.	RMAP	3
4.3.	TX DATA FORMAT	3
4.3.1.	APPLICATION HEADER AREA	3
4.3.2.	SPACEWIRE DATA AREA	4
4.4.	RX DATA FORMAT	5
4.5.	TIME CODE	5
4.6.	EXAMPLE OF TCP/IP PACKET	6
4.6.1.	EXAMPLE OF WRITE COMMAND PACKET	6
4.6.2.	EXAMPLE OF READ COMMAND PACKET	6
4.7.	CHANGE IP ADDRESS	7
4.8.	MEMORY MAP	11
4.9.	ROUTER CONFIGURATION REGISTER	12
4.9.1.	SPACEWIRE LINK CTL STS 1 to 4 REGISTER	13
4.9.2.	SPACEWIRE LINK CTL STS6 REGISTER	13
4.9.3.	SPACEWIRE LINK CTL STS7 REGISTER	14
4.9.4.	SPACEWIRE Time Code REGISTER	14
4.9.5.	SPACEWIRE Time Code En REGISTER	14
4.9.6.	SPACEWIRE Port0 Key ID REGISTER	15
4.9.7.	SPACEWIRE DEVISE ID REGISTER	15
4.9.8.	SPACEWIRE _FPGA_REV REGISTER	15
4.9.9.	SPACEWIRE _IP_REV REGISTER	16
4.9.10.	SPACEWIRE DEVISE ID REGISTER	16
4.10.	SpW PORT LINK LED	17
4.11.	DIP SWITCH	18
4.12.	POWER	19
5.	SAMPLE PROGRAM	20
5.1.	OVER VIEW	20
5.2.	OPERATING METHOD	21
5.3.	CONNECTION	23
5.4.	RMA PHEADER	24
5.5.	RMAP COMMAND	27
5.6.	TimeCode	29
5.7.	EXCUTE MACRO	30
5.8.	CREATE MACRO FILE	31
5.9.	EXAMPLE (test01.mac)	31
6.	PRINT CIRCUIT BOARD TOP VIEW	32
7.	EXTERNAL VIEW	33

## 1. OVERVIEW

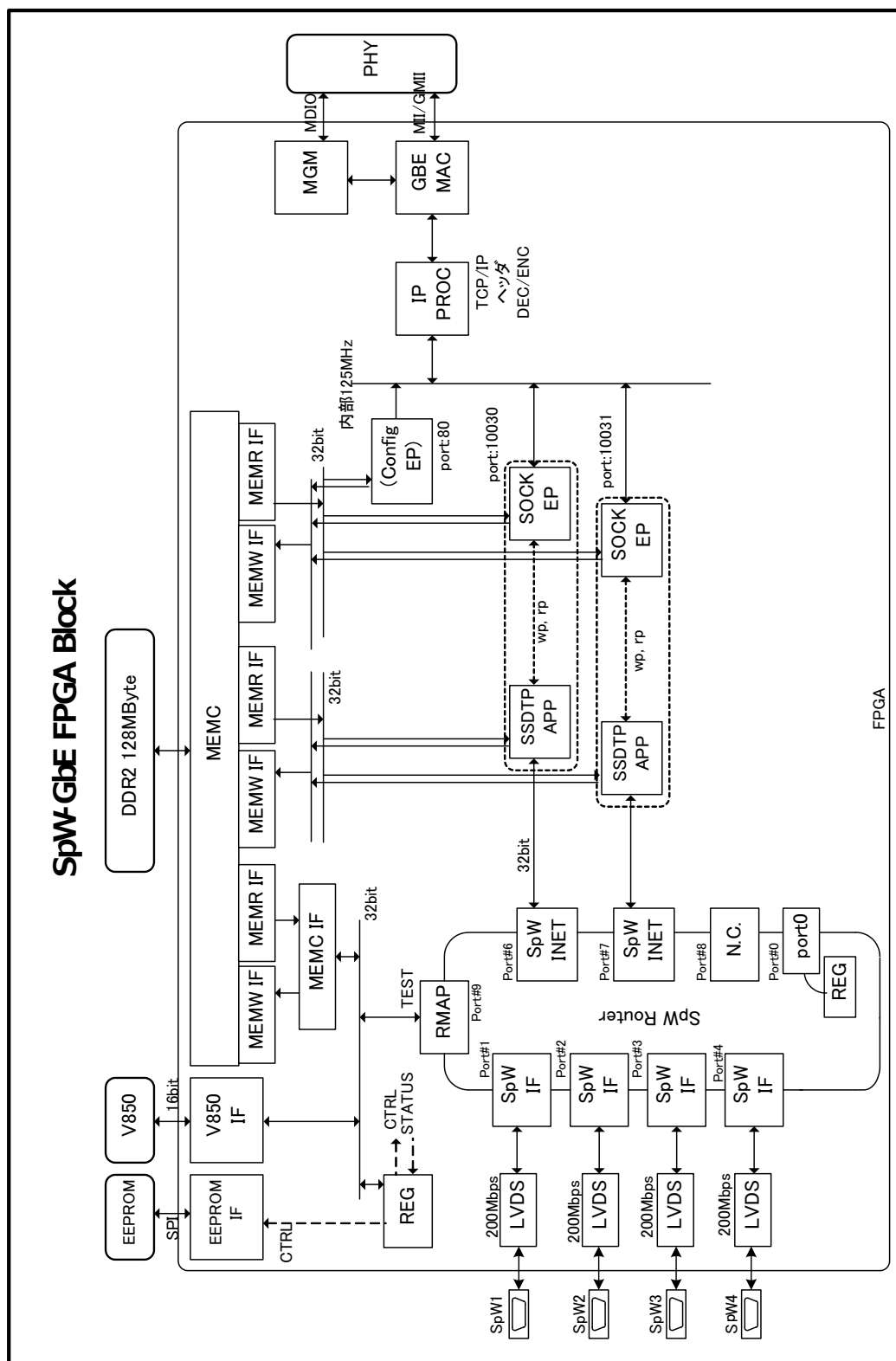
This STANDALONE SpaceWire to Gigabit Ether is the bridge between SpaceWire and Giga-bit Ether.  
The bridge function is embedded on FPGA and it transfer over 800 M bps from PC without device driver.

## 2. SPECIFICATION

Function		Specification / Feature	Remarks
FPGA		Spartan6 XC6SLX75-3FGG484	
ETHERNET	PHY	10/100/1000Base-T/Tx 1Port	
	IP Address	192.168.1.100 (Default)	Reconfigurable
	Connector	RJ-45	
Space Wire	Port	4port	
	Link Speed	MAX 200Mbps	Change transfer speed by register setting
	Connector	MDM 4port	
Size		136mm×75mm×25mm	Without protrusion
Weight		340g	
Power / Consumed current		5V 1.3A(typ)	
Operation temperature		0~45 degree (Celsius)	

### 3. BLOCK DIAGRAM

Block diagram of this Unit is described as follows.





## 4. FUNCTION

### 4.1. SPACEWIRE ROUTER

The SpaceWire and Giga-bit Ether are connected as router port in the FPGA.

Those port address are below table

Router port addresses

Port address	Space Wire Port	Remarks
Port #0	Router configuration port	Router setting port
Port #1	SpaceWire 1	MDM connector
Port #2	SpaceWire 2	MDM connector
Port #3	SpaceWire 3	MDM connector
Port #4	SpaceWire 4	MDM connector
Port #5	Reserved	N.C
Port #6	Ether HOST1	TCP port number #10030
Port #7	Ether HOST2	TCP port number #10031
Port #8	Reserved	Not open to user
Port #9	Reserved	Not open to user

### 4.2. RMAP

RMAP port (Logical address, Key ID)

RMA Port	Logical Address	Key ID	Remarks
Comfit Port	0xFE	0x02	Router seating

### 4.3. TX DATA FORMAT

SpaceWire packet is insert in TCP packet.

This is the data format.

FLAG 8bit	RSV 8bit	SIZE 80bit	DATA0 8bit	DATA1 8bit	DATA2 8bit	–	–	–	Data [SIZE-1]
Application header area			DATA0 to Data = SpaceWire Data area						

#### 4.3.1. APPLICATION HEADER AREA

The Application header consists 12 byte of information.

- FLAG (1 byte) : Packet information of this packet
  - 0x00 : add EOP (Send packet with EOP which added end of this packet by FPGA)
  - 0x01 : add EEP (Send packet with EEP which added end of this packet by FPGA)
  - 0x02 : add Continuation Packet Flag (send packet with 0x02 because SpaceWire data length is larger than TCP packet length )
  - Other : Reserved
- RSV (1byte) : Fixed to 0x00

### 4.3.2. SPACEWIRE DATA AREA

SpaceWire data area is configured by length data which specified by SIZE. The data configure as SpaceWire format. DATA0 have to specify this STANDALONE SpaceWire to Gigabit Ether port number.

- DATA0 : Specify this STANDALONE SpaceWire to Gigabit Ether port number
- 0x00 : Router Configuration Port
- 0x01 : SpaceWire 1
- 0x02 : SpaceWire 2
- 0x03 : SpaceWire 3
- 0x04 : SpaceWire 4
- 0x05 : Reserved
- 0x06 : --
- 0x07 : --
- 0x08 : Reserved
- 0x09 : Reserved
- Other : Reserved

#### ■ Transaction example 1

This data send to TCP socket when send the packet of SIZE=2048 byte with EOP to SpaceWire port 1.

FLAG 0x00	RSV 0x00	SIZE 0x800	DATA0 0x01	DATA 1	DATA 2	—	—	—	DATA 2047
--------------	-------------	---------------	---------------	-----------	-----------	---	---	---	--------------

#### ■ Transaction example 2

This data send to TCP socket when send the packet of SIZE=4000 byte with EEP to SpaceWire port 1.

FLAG 0x01	RSV 0x00	SIZE 0x00——0FA0	DATA0 0x01	DATA 1	DATA 2	—	—	—	DATA 3999
--------------	-------------	--------------------	---------------	-----------	-----------	---	---	---	--------------

#### ■ Transaction example 3

This data send to TCP socket when send the packet of SIZE=2048 byte with Continuation Packet Flag to SpaceWire port 4. ( It have to send the packet with EOP to send to TCP socket after packet with Continuation Packet Flag)

FLAG 0x02	RSV 0x00	SIZE 0x800	DATA0 0x04	DATA 1	DATA 2	—	—	—	DATA 2047
--------------	-------------	---------------	---------------	-----------	-----------	---	---	---	--------------

#### 4.4. RX DATA FORMAT

This is RX packet format of the SpaceWire replay packet

This is an example of the RX packet of SIZE=2048 byte with EOP.

FLAG 0x00	RSV 0x00	SIZE 0x800	DATA 0	DATA 1	DATA 2	--	--	--	DATA 2047
--------------	-------------	---------------	-----------	-----------	-----------	----	----	----	--------------

Note: It might be 2 SpaceWire replay packet in a TCP/IP replay packet.

#### 4.5. TIME CODE

The STANDALONE SpaceWire to Gigabit Ether can send and receive TimeCode.

It set flag to "0x30" when send TimeCode as TCP packet, and set flag to "0x31" when receive TCP packet.

This is TimeCode format

### Sending TimeCode data format

FLAG 0x30	RSV 0x00	SIZE9 0x00	SIZE8 0x00	SIZE7 0x00	SIZE6 0x00	SIZE5 0x00	SIZE4 0x00	SIZE3 0x00	SIZE2 0x00	SIZE1 0x00	SIZE0 0x02
TimeCode 8bit *1		RSV 0x00									

Note \*1 : Set TimeCode MSB 2bits to "0" and use LSB 6bit for TimeCode.

## Receiving TimeCode data format

FLAG 0x31	RSV 0x00	SIZE9 0x00	SIZE8 0x00	SIZE7 0x00	SIZE6 0x00	SIZE5 0x00	SIZE4 0x00	SIZE3 0x00	SIZE2 0x00	SIZE1 0x00	SIZE0 0x02
TimeCode 8bit		RSV 0x00									

#### 4.6. EXAMPLE OF TCP/IP PACKET

It describe example of TCP/IP packet.

#### 4.6.1. EXAMPLE OF WRITE COMMAND PACKET

```

Target address  : 0x30
Initiator address : 0xfe
Address         : 0x0000-0000
Data length     : 4 byte
Write data      : 0x12345678
Instruction     : 0x71

```

Flag	Reserved	Size[9]	Size[8]	Size[7]	Size[6]	Size[5]	Size[4]	Size[3]	Size[2]	Size[1]	Size[0]
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x1a
Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]	Data[10]	Data[11]
0x01	0x30	0x01	0x71	0x02	0x00	0x00	0x00	0x06	0xfe	0x00	0x00
Data[12]	Data[13]	Data[14]	Data[15]	Data[16]	Data[17]	Data[18]	Data[19]	Data[20]	Data[21]	Data[22]	Data[23]
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x04	0xc6	0x78	0x56	0x34
Data[24]	Data[25]										
0x12	0x32										

#### 4.6.2. EXAMPLE OF READ COMMAND PACKET

```
Target address  : 0x30
Initiator address : 0xfe
Address        : 0x0000-0000
Data length    : 4 byte
Instruction     : 0x4d
```

Flag	Reserved	Size[9]	Size[8]	Size[7]	Size[6]	Size[5]	Size[4]	Size[3]	Size[2]	Size[1]	Size[0]
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x15
Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]	Data[10]	Data[11]
0x01	0x30	0x01	0x4d	0x02	0x00	0x00	0x00	0x06	0xfe	0x00	0x00
Data[12]	Data[13]	Data[14]	Data[15]	Data[16]	Data[17]	Data[18]	Data[19]	Data[20]			
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x04	0x30			

<Read replay>

Flag	Reserved	Size[9]	Size[8]	Size[7]	Size[6]	Size[5]	Size[4]	Size[3]	Size[2]	Size[1]	Size[0]
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x11
Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]	Data[10]	Data[11]
0xfe	0x01	0x0c	0x00	0x030	0x00	0x00	0x00	0x00	0x00	0x04	0xa8
Data[12]	Data[13]	Data[14]	Data[15]	Data[16]							
0x78	0x56	0x34	0x12	0x32							

## 4.7. CHANGE IP ADDRESS

The default IP address is "192.168.1.100" of this unit.

The default IP address can change via web browser, use pertinence IP address to mach with network configuration.

< Network setting for PC >

IP address : 192.168.1.101

(Select IP address a part from "1000", because The default IP address is 192.168.1.100 of this unit.)

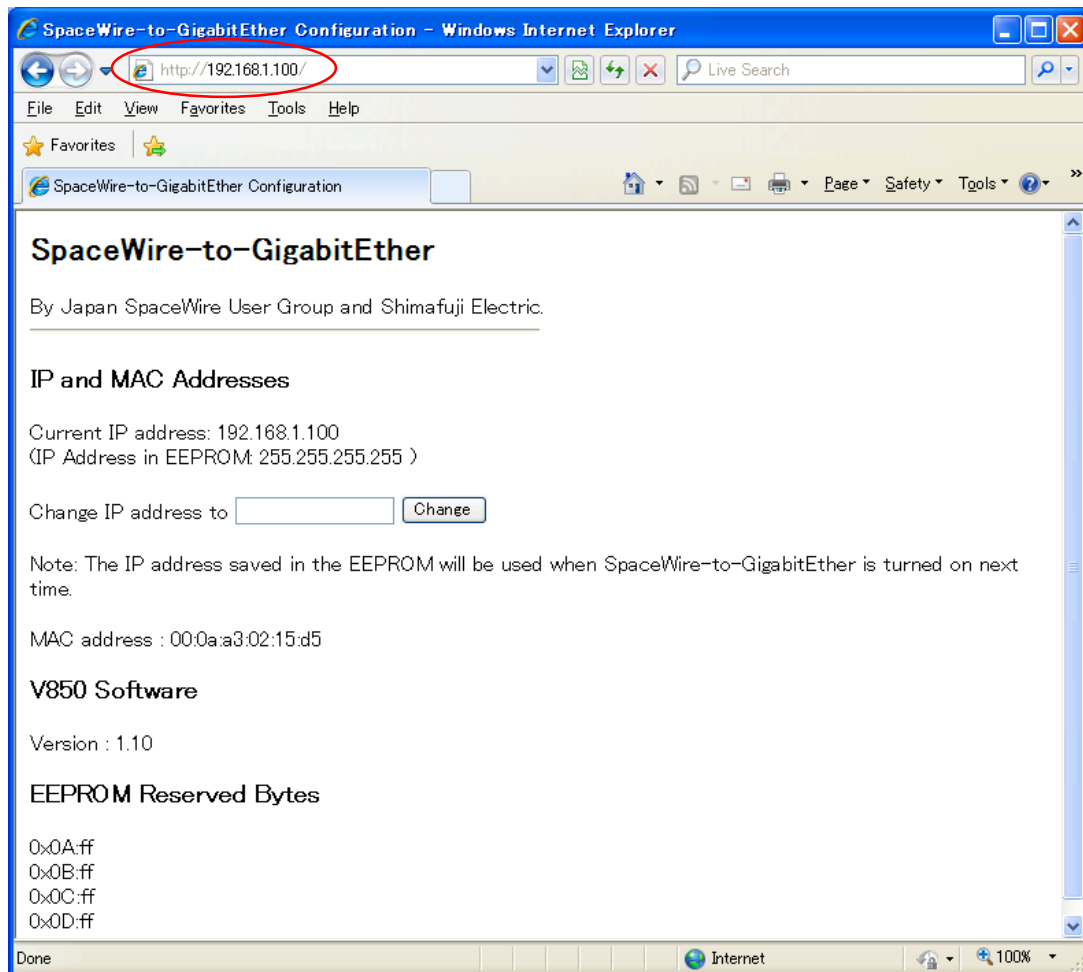
Subnet address : 255.255.255.0 ( It is not necessary to change.)

Default gateway : 192.168.0.1 (It is not necessary to change.)

Access 192.168.1.100 via web browser after connect PC and this unit by Ethernet cable. Please note "Safari" (MAC OS default browser) is not supported, so it mifgt be use something like "Camino" (<http://caminobrowser.org/>. )

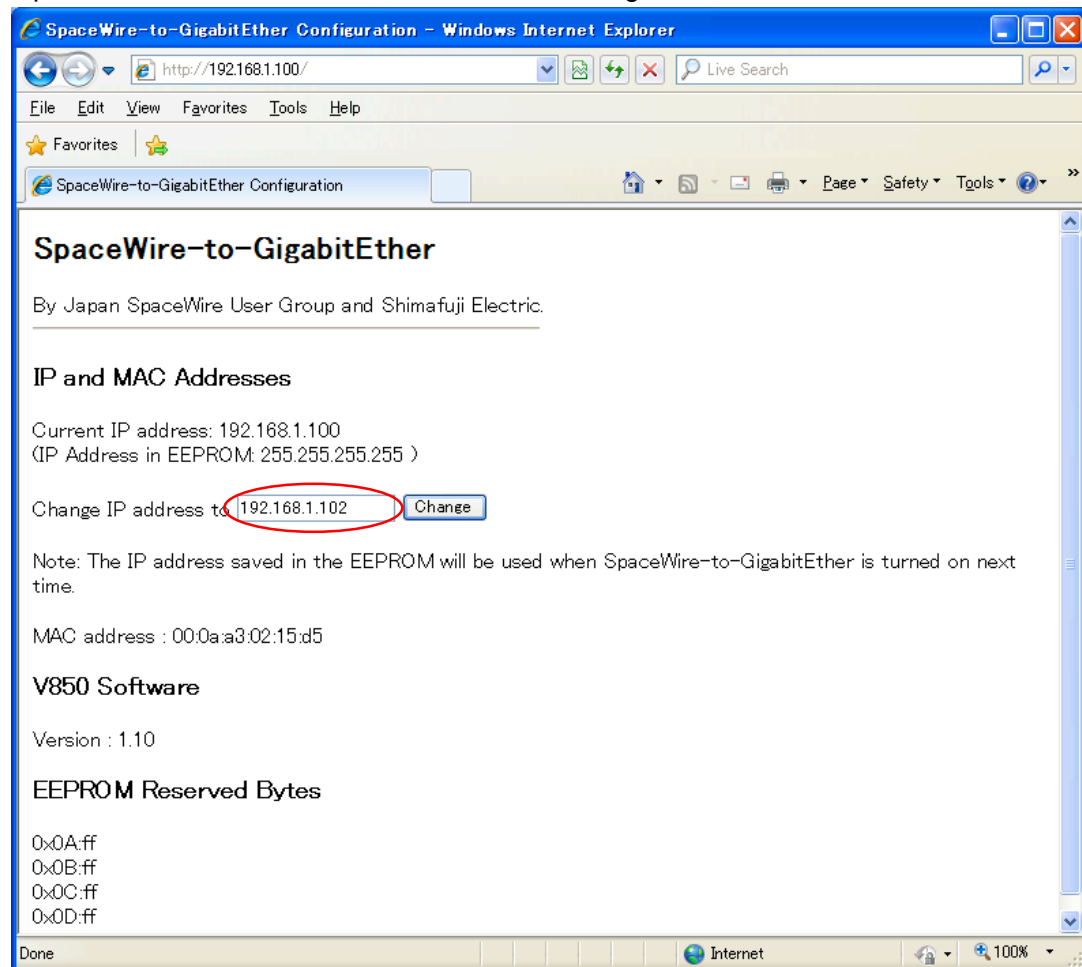
< Web server start up screen >

Input "192.168.1.100" in address bar.

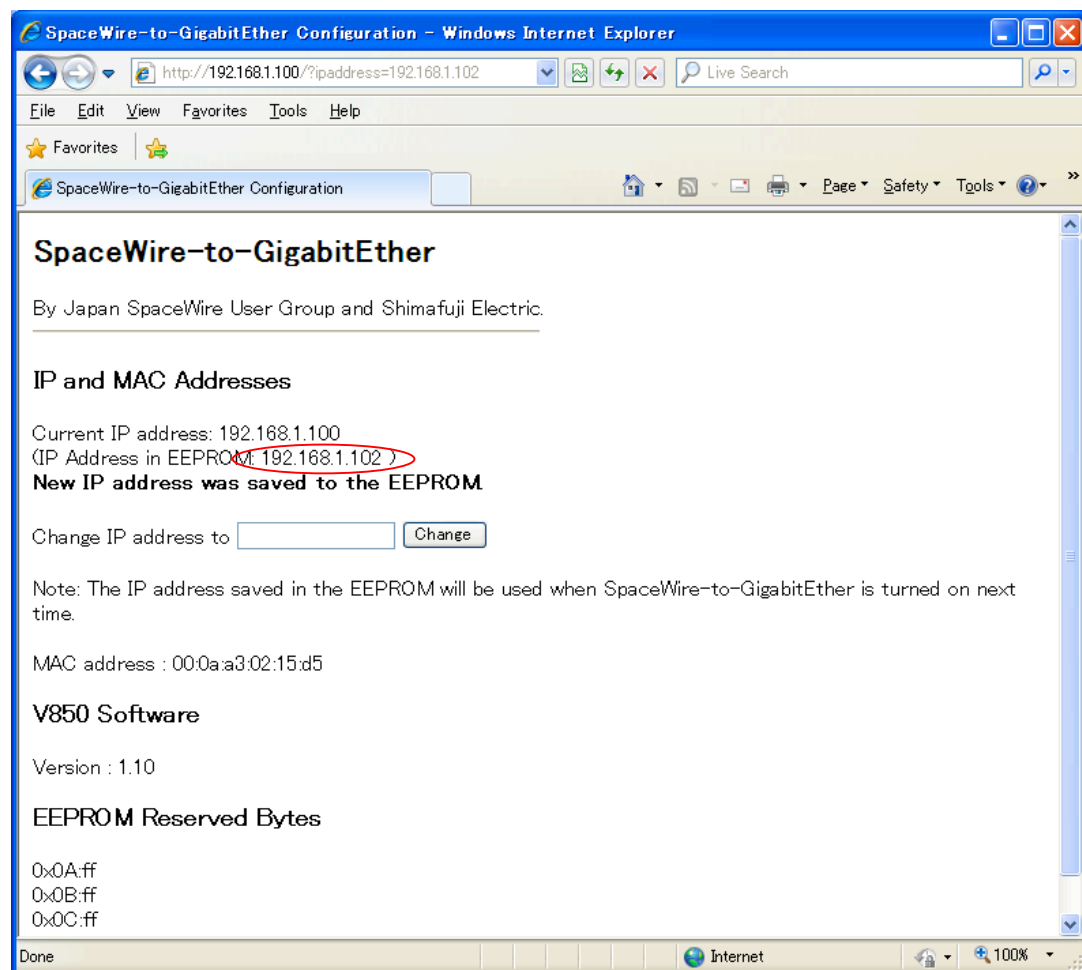


< Change IP address to "192.168.0.102" screen >

Input "192.168.0.102" in red circle and then click "change".



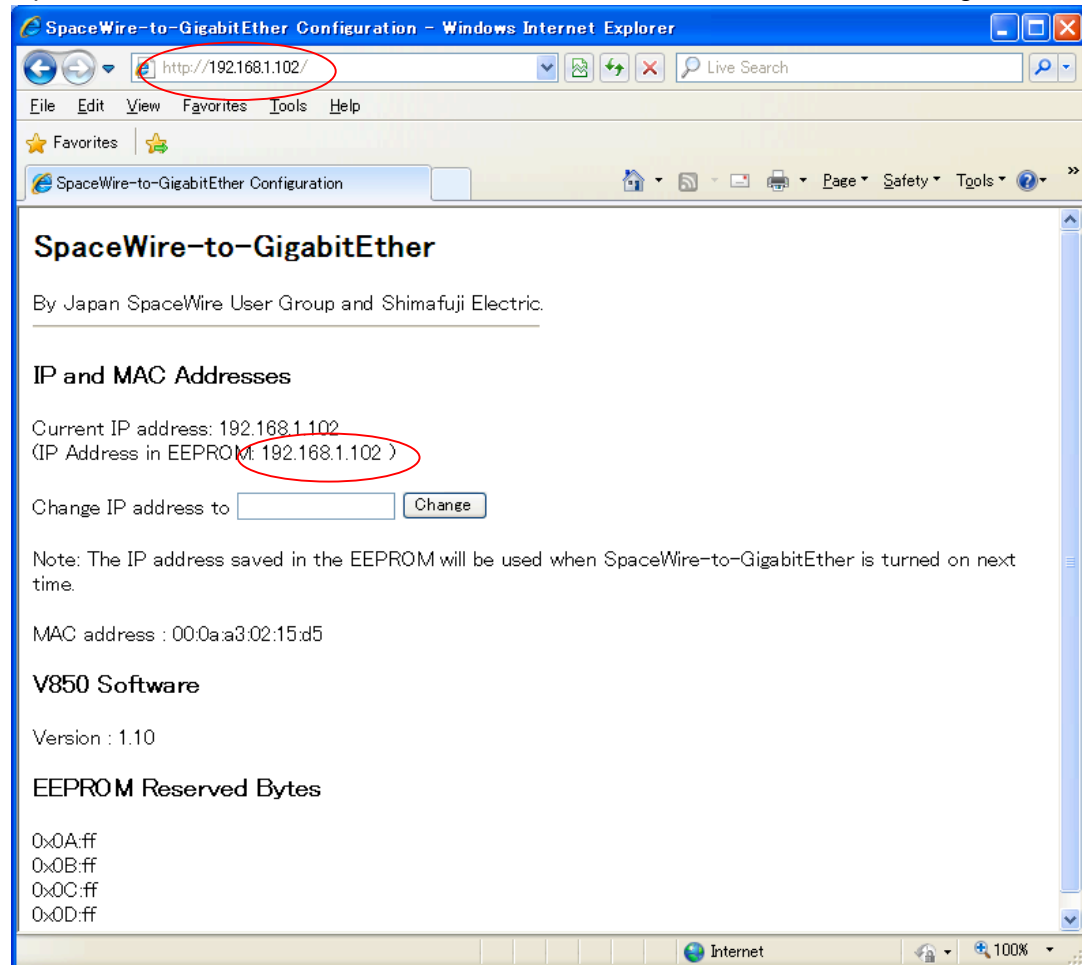
< Changed IP address to "192.168.0.102" >



< Confirm changing address to "192.168.0.102" >

Turn OFF the power and then turn ON the power again.

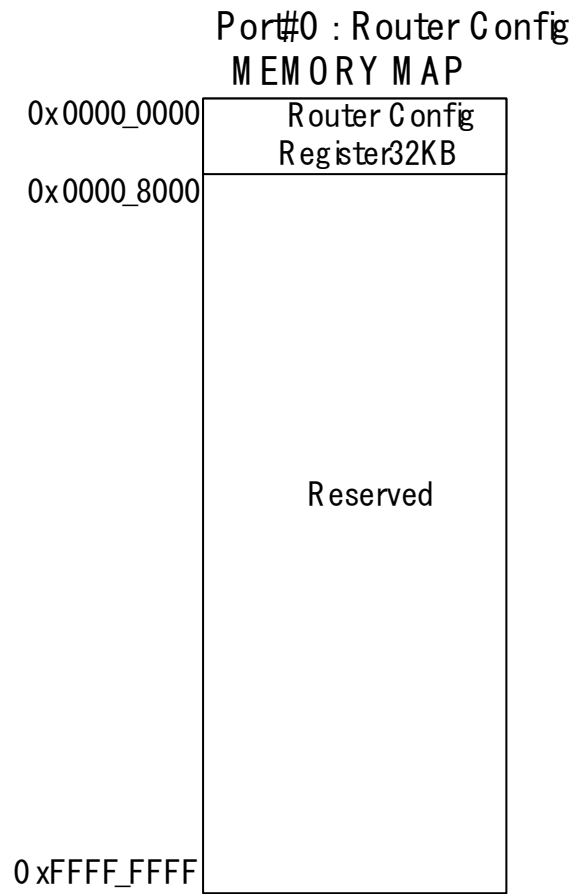
Input "192.168.0.102" in address bar in web browser, and confirm IP address is changed on the screen





## 4.8. MEMORY MAP

This is the memory map of STANDALONE SpaceWire to Gigabit Ether



## 4.9. ROUTER CONFIGURATION REGISTER

This is the Router Configuration register table.  
it can access to this register by 32 bits access only.

Offset.	Register NAME	SYMBOL	R/W	Reset Value	Remarks
0x0004	SPACEWIRE LINK CTL STS1	SPACEWIRE_LINKCSR1	R/W	0x1305_----	
0x0008	SPACEWIRE LINK CTL STS2	SPACEWIRE_LINKCSR2	R/W	0x1305_----	
0x000C	SPACEWIRE LINK CTL STS3	SPACEWIRE_LINKCSR3	R/W	0x1305_----	
0x0010	SPACEWIRE LINK CTL STS4	SPACEWIRE_LINKCSR4	R/W	0x1305_----	
0x0018	SPACEWIRE LINK CTL STS6	SPACEWIRE_LINKCSR6	R	0x0000_0001	
0x001C	SPACEWIRE LINK CTL STS7	SPACEWIRE_LINKCSR7	R	0x0000_0000	
0x0410	SPACEWIRE Time Code	SPACEWIRE_TCODE	R	0x0000_0000	
0x041C	SPACEWIRE Time Code En	SPACEWIRE_TCODEEN	R/W	0x0000_00FE	
0x0424	SPACEWIRE Port0 Key ID	SPACEWIRE_PORTKEY	R/W	0x8000_FE02	
0x0430	SPACEWIRE Device ID	SPACEWIRE_DEVICEID	R	0x4762_4553	
0x0434	SPACEWIRE FPGA Revision	SPACEWIRE_FPGA_REV	R	0x4022_0101	
0x0438	SPACEWIRE IP Revision	SPACEWIRE_IP_REV	R	0x4022_0201	
0x043c	SPACEWIRE RMAP Revision	SPACEWIRE_RMAP_REV	R	0x4022_0200	

#### 4.9.1. SPACEWIRE LINK CTL STS 1 to 4 REGISTER

This is the register's to control SpaceWire link status and link speed.

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0004 0x0008 0x000c 0x0010	R/W	SPACEWIRE LINK CTL STS1 SPACEWIRE LINK CTL STS2 SPACEWIRE LINK CTL STS3 SPACEWIRE LINK CTL STS4	link_control1 link_control2 link_control3 link_control4																error_status1 error_status2 error_status3 error_status4								Link_status1 Link_status2 Link_status3 Link_status4							

Bit		name	R/W	default value	Function
[31:30]	Link_control[15:0]	valid	R	00	
[29:24]		txck_div	R/W	010011	SPACEWIRE Transfer clock = 200 / (txck_div+1) [MHz]
[23:20]		valid	R	00	
[19]		link_reset	R/W	0	1 => link_reset
[18]		auto_start	R/W	1	1 => auto_start
[17]		link_disable	R/W	0	1 => link_disable
[16]		link_start	R/W	1	1 => link_start
[15]		valid	R	0	
[14]	error_status[7:0]	escape_error	R	*	1 => Escape error
[13]		disconnect_error	R	*	1 => Disconnect error
[12]		parity_error	R	*	1 => Parity error
[11]		valid	R	0	
[10]		rx_err	R	*	1 => Receiver error
[9]		credit_error	R	*	1 => Credit error
[8]		sequence_error	R	*	1 => Sequence error
[7]		reset	R	*	0 => Reset
[6]	Link_status[7:0]	valid	R	0	
[5]		send_time_codes	R	*	1 => Send TIME CODES
[4]		send_n_chars	R	*	1 => Send N CHARACTERS
[3]		send_fcts	R	*	1 => Send FCTS
[2]		send_nulls	R	*	1 => Send NULLS
[1]		enable_rx	R	*	1 => Receiver enable
[0]		enable_tx	R	*	1 => Transmitter enable

#### 4.9.2. SPACEWIRE LINK CTL STS6 REGISTER

This is the link status registers for Ether HOST1.

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0018	R	SPACEWIRE LINK CTL STS6	all 0																															link_on[6]

Bit	name	R/W	default value	Function
[0]	link_on[6]	R	0	Host1 link on status Linkstatus="1"

### 4.9.3. SPACEWIRE LINK CTL STS7 REGISTER

This is the link status registers for Ether HOST2.

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x001c	R	SPACEWIRE LINK CTL STS7	all 0																																			link on[7]

Bit	name	R/W	default value	Function
[0]	link_on[7]	R	0	Host2 link on status Linkstatus="1"

### 4.9.4. SPACEWIRE Time Code REGISTER

This is the registers to store TimeCode for SPACEWIRE router in the STANDALONE SpaceWire to Gigabit Ether.

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0410	R	SPACEWIRE Time Code	all 0																								TIME_CODE[7:0]							

Bit	name	R/W	default value	Function
[7:0]	TIME_CODE[7:0]	R	0x00	TimeCode

### 4.9.5. SPACEWIRE Time Code En REGISTER

This is the register to set "enable / unable" sending TimeCode from SpaceWire router Port#1 to #4 of STANDALONE SpaceWire to Gigabit Ether.

Each bit correspond each port number.

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x041c	R/W	SPACEWIRE Time Code En	all 0																						0	0	1	1	0	TCODE_EN[4:1]		0		

Bit	name	R/W	default value	Function
[9:0]	TCODE_EN[9:0]	R/W	0011011110	TimeCode enable/unable setting : "1"= enable to send TimeCode "0"= unable to send TimeCode

#### 4.9.6. SPACEWIRE Port0 Key ID REGISTER

This is the registers to set RMAP (CRC / address / Key) for SPACEWIRE router in the STANDALONE SpaceWire to Gigabit Ether.

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0424	R/W	SPACEWIRE Port0 Key ID	CRC_REV	all 0																PORT_LADR[7:0]								PORT_KEY[7:0]							

Bit	name	R/W	default value	Function
[31]	CRC_REV	R/W	1	CRC Flag
[15:8]	PORT_LADR[7:0]	R/W	0xfe	RMAP port logical address
[7:0]	PORT_KEY[7:0]	R/W	0x02	RMAP port D-key

#### 4.9.7. SPACEWIRE DEVISE ID REGISTER

This is the revision registers for this device (SpaceWire to Giga-bit Ether Unit)

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0430	R	SPACEWIRE_DEVISEID	SPACEWIRE_DEVISEID[31:0]																															

Bit	name	R/W	default value	Function
[31:0]	SPACEWIRE_DEVISEID[31:0]	R	0x4762_4553	SPACEWIRE_DEVISEID

#### 4.9.8. SPACEWIRE \_FPGA\_REV REGISTER

This is the revision registers for SPACEWIRE \_FPGA.

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0434	R	SPACEWIRE _FPGA_REV	SPACEWIRE _FPGA_REV [31:0]																															

Bit	name	R/W	default value	Function
[31:0]	SPACEWIRE _FPGA_REV [31:0]	R	0x4022_0101	SPACEWIRE _FPGA_REV

#### 4.9.9. SPACEWIRE \_IP\_REV REGISTER

This is the revision registers for SPACEWIRE \_IP.

Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0438	R	SPACEWIRE _IP_REV	SPACEWIRE _IP_REV [31:0]																															

Bit	name	R/W	default value	Function
[31:0]	SPACEWIRE _IP_REV [31:0]	R	0x4022_0201	SPACEWIRE _IP_REV

#### 4.9.10. SPACEWIRE DEVISE ID REGISTER

This is the revision registers for SPACEWIRE \_RMAP.

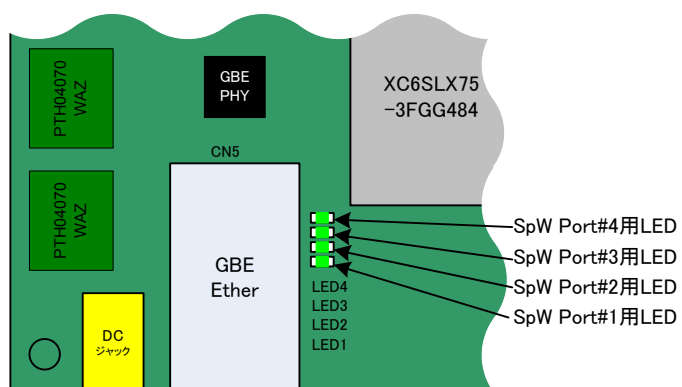
Offset Address	R/W	Function	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x043c	R	SPACEWIRE _RMAP_REV	SPACEWIRE _RMAP_REV [31:0]																															

Bit	name	R/W	default value	Function
[31:0]	SPACEWIRE _RMAP_REV [31:0]	R	0x4022_0200	SPACEWIRE _RMAP_REV

## 4.10. SpW PORT LINK LED

Those are LEDs to display link status of SpW Port #1 to #4.

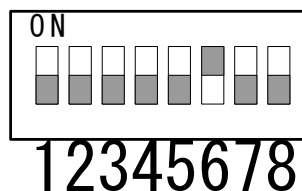
LED No.	Function	Remarks
LED1 (Green)	SpaceWire Port#1link status	Illuminate = SpaceWire link is established Not illuminate = SpaceWire link is not established
LED2 (Green)	SpaceWire Port#2link status	
LED3 (Green)	SpaceWire Port#3link status	
LED4 (Green)	SpaceWire Port#4link status	



## 4.11. DIP SWITCH

There are a DIP Switch (DSW1) on board.

DIP switch (DSW1) setting table



Bit	Function	Setting	Default
1	IP ADDRESS	ON=Fix IP address (192.168.1.100) OFF=User setting IP4 address	OFF
2	Reserved		OFF
3	Reserved		OFF
4	Reserved		OFF
5	LINK Speed[0]	Transfer clock setting after Link established	OFF
6	LINK Speed[1]		ON
7	LED	Use this bits for debugging a part from "00". Set OFF for normal use	OFF
8	LED		OFF

Bit5	Bit6	Comment
x	ON	10MHz Transfer clock setting value (default setting)
ON	OFF	100MHz Transfer clock setting value
OFF	OFF	200MHz Transfer clock setting value

Note : This setting is default value after turn power on.

It can change transfer clock speed by software to set transfer clock register.

Please refer "4.7.1SPACEWIRE LINK CTL STS1 to4 REGISTER" for details.

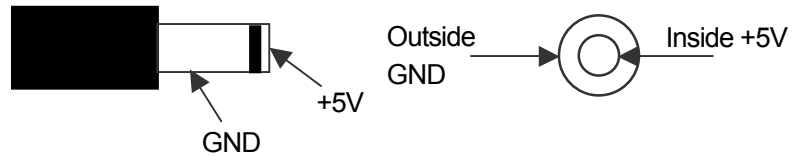


## 4.12. POWER

It will supply the power to STANDALONE SpaceWire to Gigabit Ether from attached AC adapter.

### Specification

- Power supply voltage :  $5V \pm 5\%$  / Ripple less than 300 mV
- Consumed current : MAX2A
- Plug :  $\phi 1.7\text{mm}$  DC Plug (outer diameter  $\phi 4\text{mm}$ )
- Jack (on PCB) : HEC3600-010020 (Hoshiden)
- polarity : Center +



## 5. SAMPLE PROGRAM

There are sample program for windows PC to download from web page which is written on the attached document come with SpaceWire to Giga bit Ether UNIT.

This chapter describe operation method of this sample program "SpaceWire RMAP TEST"

### 5.1. OVER VIEW

The "SpaceWire RMAP TEST" program is the program to test read or write on RMAP protocol on SpaceWire.

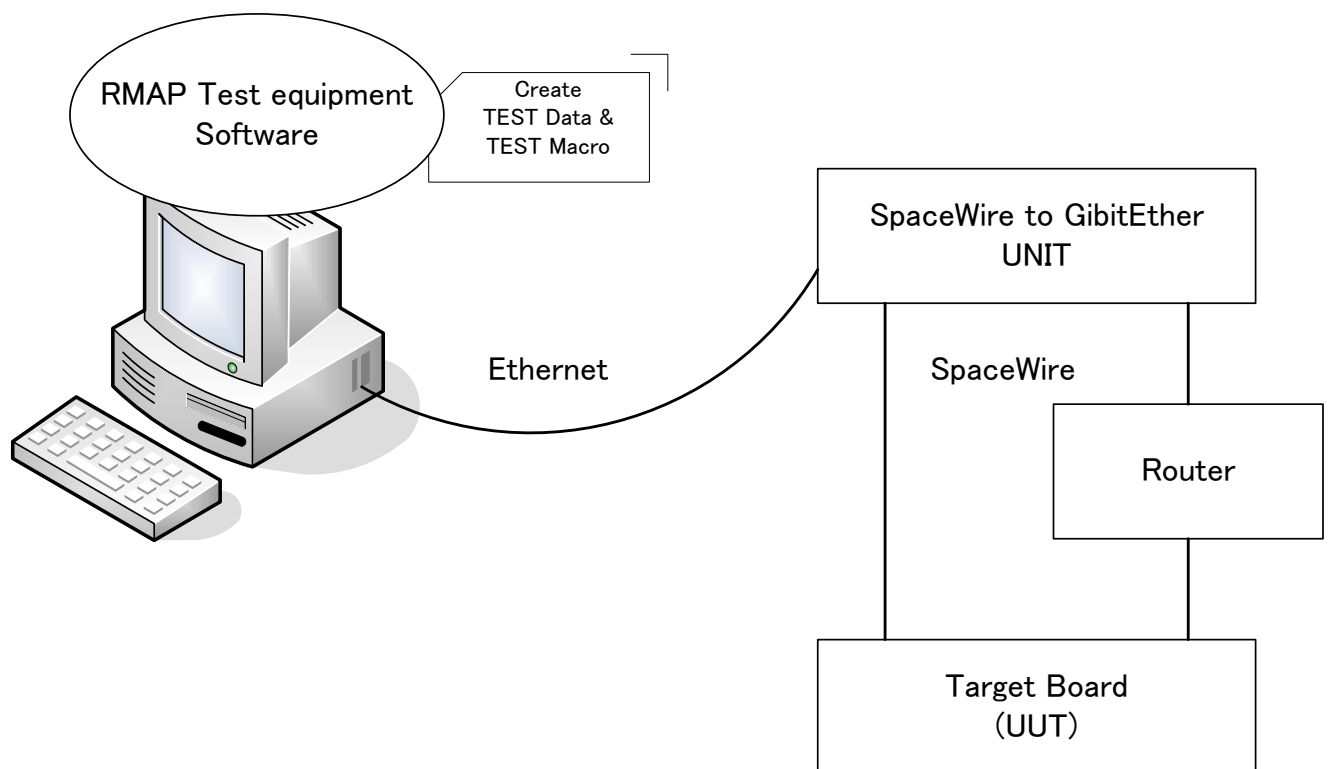
- It can operate on WindowsPC (confirmed on WindowsXP)
- This program is compiled by BorlandC++Bulder5.

It also attach source code (project) for reference to understand method to access SpaceWire etc. The attachment source is OPEN SOURCE.

#### Configuration

This diagram is for inspecting SpW Target Unit by this SpaceWire to Giga bit Ether UNIT.

It can connect to SpW Target Unit, directly or via router.



## 5.2. OPERATING METHOD

The "SpaceWire RMAP TEST" program will display this window after start up.

**SpaceWire RMAP TEST**

**Ether setting**  
 IP Address: 192 . 168 . 1 . 100  
 Port no: 10030  
 Status (red button)  
 Connect (button)  
 Exit (button)

**System**  
 Ver1.01 2012/5/11

**RMAP | Macro**

**RMAP setting**  
 SPW PATH ADDRESS (Input:010203)  
 09 HED Read  
 0xaa 0x01 0011b 0x02  
 Reply Addr: 4  

0x00	0x00	0x00	0x06
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00
0xFE	0x0000	0x00	
0x01000000			
0x000400			

 CRC

**RMAP Command**  
☒ Read Send  
☐ Write : File File select  
☐ Write : Input 0x (32bit only)

**TimeCode**  
 Value(64bit) 0 Send  
 Count 1

**Replay Data** Clear

**Communication-Monitor** Clear

- Ether Setting : Set Ether configuration to connect this SpaceWire to Giga bit Ether UNIT.
- System : Display Version information and close program to click "Exit" button.
- RMAP setting : Set RMAP header information to Tx.
- RMAP Command : Issue Read/Write command of RAMP
- Time Code : Issue Time Code
- ReplyData : Display reply response data
- Communication-Monitor : Display result of operation

Display bellow wondow then click MACRO tab.

The screenshot shows the 'SpaceWire RMAP TEST' application window. At the top, there's a title bar with standard window controls. Below it, the 'Ether setting' section contains IP address fields (192, 168, 1, 100) and a port number field (10030). To the right of these fields are 'Status' and 'Connect' buttons. Further right, a 'System' section displays 'Ver1.01 2012/5/11' and an 'Exit' button. Below the settings, there are two tabs: 'RMAP' and 'Macro'. The 'Macro' tab is active, showing a 'Macro' section with a text input field, a 'File select' button, an 'Exec Macro' button, a checked 'Error Stop' checkbox, and an 'Error Count' field set to 0. To the right of the 'Macro' tab, there are two large empty rectangular areas labeled 'Replay Data' and 'Communication-Monitor', each with a 'Clear' button in its top right corner.

Macro : Excute selected file Macro (test script)

### 5.3. CONNECTION

Set Ether net configuration (IP address, port number) to connect this UNIT.

The status display will be turn GREEN when success to connect after click "connect" button.

The screenshot shows the 'SpaceWire RMAP TEST' application window. It features a top status bar with a green 'Status' button and a 'Disconnect' button. Below this, the 'Ether setting' section includes IP address fields (192, 168, 1, 100) and a port number field (10030). To the right, the 'System' information displays 'Ver1.01 2012/5/11' and an 'Exit' button. The main interface is divided into two panes. The left pane, titled 'RMAP | Macro', contains the 'RMAP setting' section with a table for SPW PATH ADDRESS (Input: 010203) and a 'HED Read' button. The table has columns for '09', '0xaa', '0x01', '0011b', and '0x02'. Below the table is a 'Reply Addr' dropdown set to '4'. The 'RMAP Command' section has radio buttons for 'Read', 'Write : File', and 'Write : Input', with a 'Send' button. The 'TimeCode' section has a 'Value(64bit)' field set to '0' and a 'Count' field set to '1', with a 'Send' button. The right pane contains a 'Replay Data' section with a 'Clear' button and a 'Communication-Monitor' section with a 'Clear' button and a text area showing '>Ether Connection'.

SPW PATH ADDRESS (Input: 010203)				
09				
0xaa	0x01	0011b		0x02
Reply Addr: 4				
0x00	0x00	0x00		0x06
0x00	0x00	0x00		0x00
0x00	0x00	0x00		0x00
0xFE	0x0000			0x00
0x01000000				
0x000400				
CRC				

## 5.4. RMA PHEADER

This "RMAP setting" is same order as RMAP Tx header.

Input those ① to ⑩ data in each field.

The screenshot shows the "RMAP setting" dialog box. The title bar says "RMAP setting". Below it, "SPW PATH ADDRESS (Input:010203)" is displayed. The main area contains several fields and a "HED Read" button. The fields are as follows:

- Field 1: A text box containing "09".
- Field 2: A text box containing "0xaa".
- Field 3: A dropdown menu showing "0x01".
- Field 4: A dropdown menu showing "0011b".
- Field 5: A dropdown menu showing "0x02".
- Field 6: A text box containing "0x00".
- Field 7: A text box containing "0x00".
- Field 8: A text box containing "0x00".
- Field 9: A text box containing "0x06".
- Field 10: A text box containing "0x00".
- Field 11: A text box containing "0x00".
- Field 12: A text box containing "0x00".
- Field 13: A text box containing "0x00".
- Field 14: A text box containing "0x00".
- Field 15: A text box containing "0xFE".
- Field 16: A text box containing "0x0000".
- Field 17: A text box containing "0x01000000".
- Field 18: A text box containing "0x000400".
- Field 19: A text box containing "CRC".

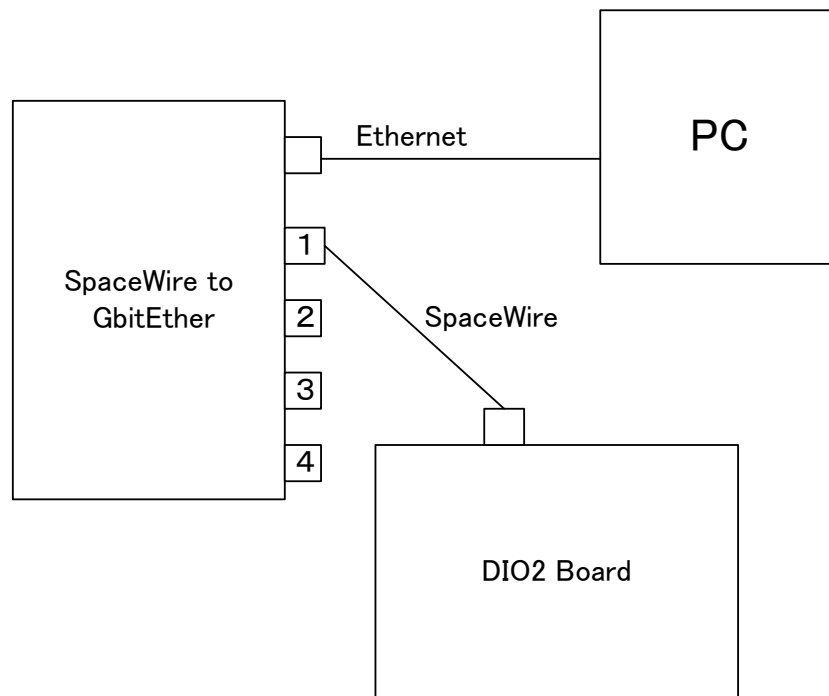
The "HED Read" button is located to the right of the first row of fields. The "Reply Addr:" label is positioned above the field containing "4".

- ① Target SpaceWire address field  
Set pass address in Hexadecimal, it could set multiple pass address.  
Example to set multiple pass address : 0102  
The pass address of this UNIT is as bellow.  
01 : SPW port 1  
02 : SPW port 2  
03 : SPW port 3  
04 : SPW port 4  
06 : Ether port
- ② Target logical address field  
Specified target logical address
- ③ Command field  
This field is automatically set.
- ④ Key field  
Set target key
- ⑤ Replay address length field  
Select from 0,4,8,12
- ⑥ Replay address field  
Specified replay address  
"0x06" is Ether port pass address of this UNIT
- ⑦ Initiator logical address field  
Specified SpW address of this UNIT.
- ⑧ Transaction identifier field  
Specified packet identifier

- ⑨ Address field  
Specified target address
- ⑩ Data Length field  
Specified data length

This is example setting to read/write first 1k bytes from/to SpaceWireDIO2 board (shimafuji Electric Inc) memory.

Connection diagram :



Setting :

RMAP setting

SPW PATH ADDRESS (Input:010203)

01

0x30 0x01 0011b 0x02

Logical address=0x30

0x00 0x00 0x00 0x06

0x00 0x00 0x00 0x00

0x00 0x00 0x00 0x00

0xFE 0x0000 0x00

0x00000000

0x000400 CRC

Pass address=1

HED Read

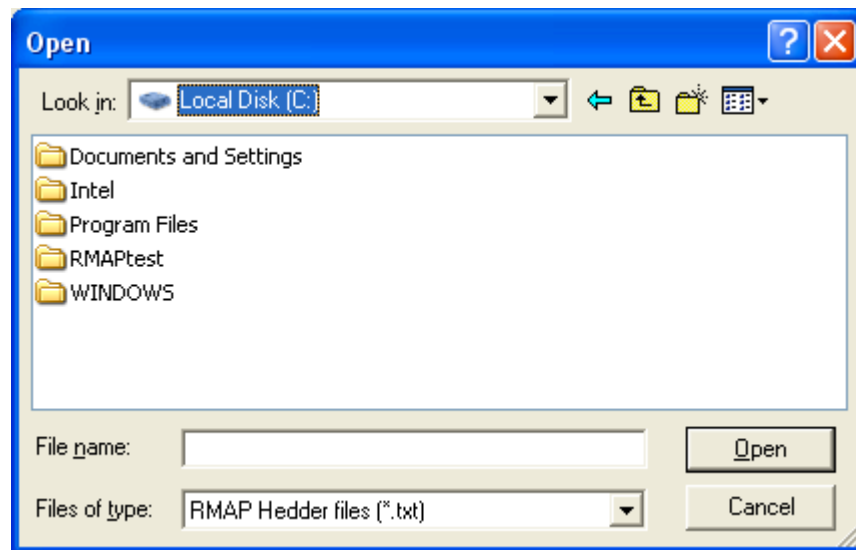
Key=0x02

Replay

Start address=0x0

Set size = 1k bytes

It also could load setting file which has setting information already.



RMAP header should make "key name = value" format

Example of RmapHed01.txt file.

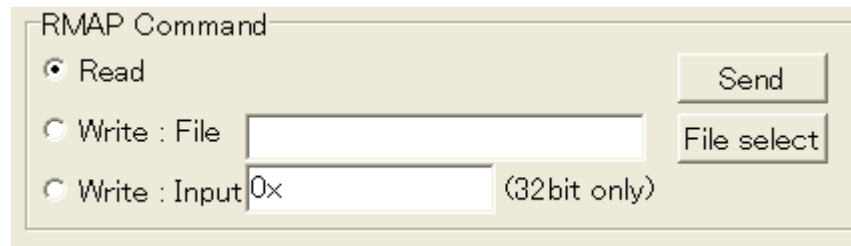
PATH_ADDR=010106	----- Pass address
TARGET_LOG_ADDR=0xaa	----- Target logical address
KEY=0x02	----- Key
REPLY_ADDR=00000006	----- Replay address
INITIATOR_ADDR=0xfe	----- Initiator logical address
TRANSACTION=0x0000	----- Transaction identifier
RMAP_ADDR=0x1000000	----- Address
DATA_SIZE=0x400	----- Data Length

The pass address is hexadecimal without "0x".

The replay address is hexadecimal without "0x". It automatically generate replay address length.



## 5.5. RMAP COMMAND



There are 3 kind of command

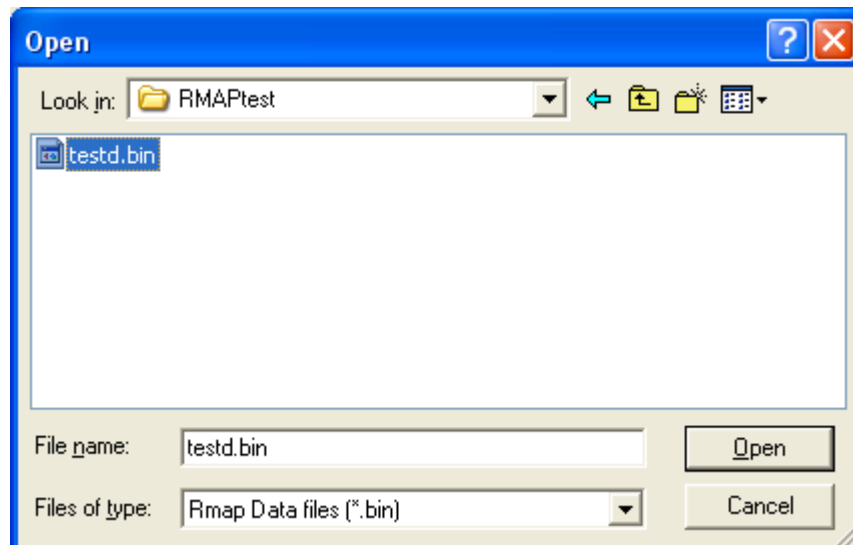
Click "Send" for issuing command.

- Read : Issue Read command flowing RMAP header setting  
Display read data in replay area.
- WriteFile : Specified RMAP data file by " FileSelect" button  
The data file is binary data.
- Write Input : Write 32 bit specified data. \*Writing little Indian

Those are example of read/write 1024 bytes increment data.

① Write data

Specified RMAP data file by " FileSelect" button



Set data length=0x400, Click "Send" button to write in the WriteFile mode.  
The testd.bin file is increment data which is byte unit..

② Read data

Read from the address which is same address as written in ①.

Click "Send" button in the Read mode, It will be display read data in ReplyData area for data confirmation.

**SpaceWire RMAP TEST**

Ether setting  
 IP Address: 192 . 168 . 1 . 100  
 Port no: 10030  
 Status  
 Disconnect  
 System: Ver1.01 2012/5/11  
 Exit

RMAP | Macro

RMAP setting  
 SPW PATH ADDRESS (Input:010203)  
 09 HED Read  
 0xaa 0x01 0011b 0x02  
 Reply Addr: 4  
 0x00 0x00 0x00 0x06  
 0x00 0x00 0x00 0x00  
 0x00 0x00 0x00 0x00  
 0xFE 0x0000 0x00  
 0x01000000  
 0x000400 CRC

RMAP Command  
☒ Read Send  
☐ Write : File File select  
☐ Write : Input 0x (32bit only)

TimeCode  
 Value(0-63) 0 Send  
 Count 1

Replay Data Clear  
 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F  
 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F  
 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F  
 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF  
 B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF  
 C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF  
 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF  
 E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF  
 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF  
 56

Communication-Monitor Clear  
 >Ether Connection

## 5.6. TimeCode

A screenshot of a software interface titled "TimeCode". It contains two input fields: "Value(0-63)" with the value "0" and "Count" with the value "1". To the right of these fields is a "Send" button.

TimeCode

Value(0-63) 0

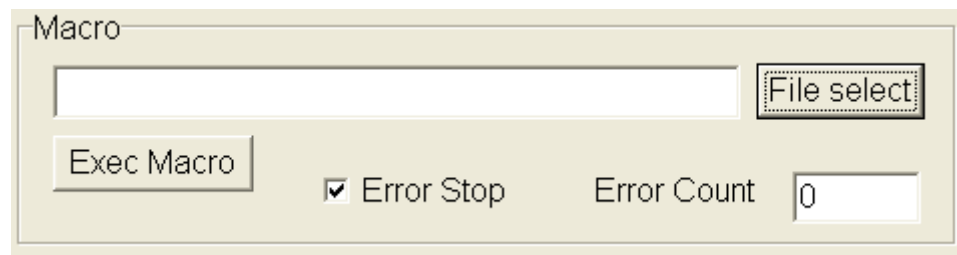
Count 1

Send

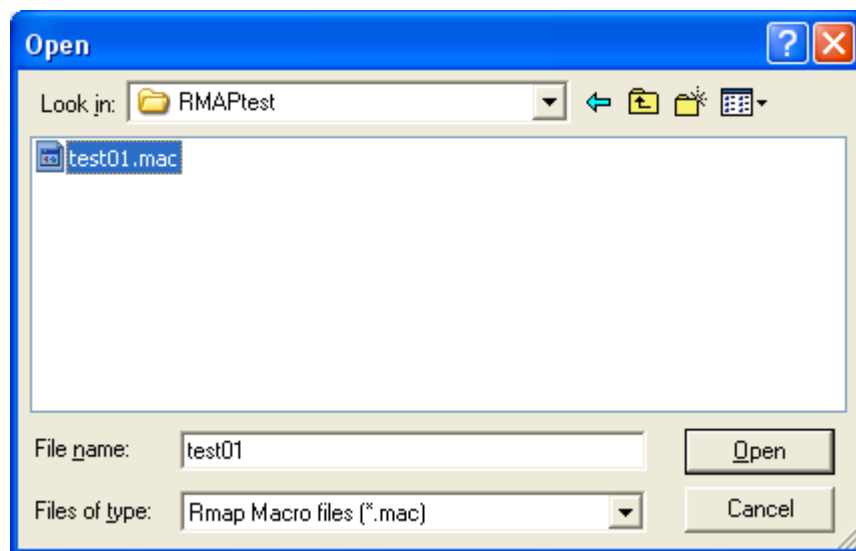
- Value : Set default value between 0 to 63
- Count : Dispatch number of TimeCode
- Send button : Issue TimeCode  
Increment "Value" each TimeCode send until number of "count" value times.

If send "3", "4", "5" and "6" TimeCodes, to set Value=3 and Count=4 then click "Send" button.  
The "Value" will be "7" after send TimeCode 4 times.

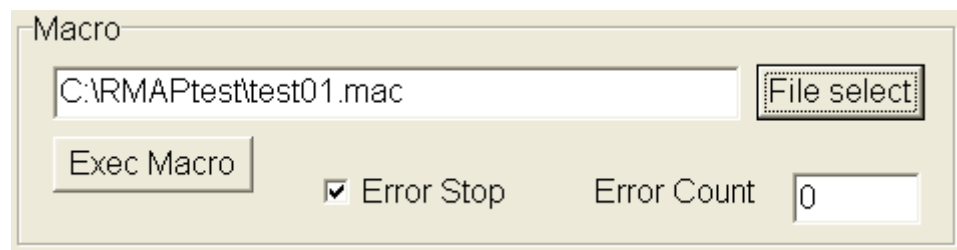
## 5.7. EXECUTE MACRO



Specify Macro file pressing "FileSelect" then click "ExecMacro" button.



The "ExecMacro" button will be change to "STOP" button.  
It could enforce to STOP Macro clicking "STOP" button.



Check "Error Stop" for stop Macro automatically in the error occur.  
The "Error Count" is count up error event and display number of errors after excute macro.

## 5.8. CREATE MACRO FILE

Write test script in command format to Macro file.

test script and command list

No	Command function	Command parameter	Remarks
1	Update write heeder information	HED,"File Name"	
2	Write	WT, Address, Size ,"File Name"	
3	Read	RD, Address, Size ,"File Name"	Save read data to Spacified file
4	Compare files	CMP, Size ,"File Name" ,"File Name"	
5	End Macro	END	End Macro If there are no this command, it will repeat Macro.

## 5.9. EXAMPLE (test01.mac)

Excute the script (test01.mac)

```
HED,"RmapHed01.txt"
WT,0x1000000,0x400,"RmapTestData001.bin"
RD,0x1000000,0x400,"Resp001.bin"
CMP,0x400,"RmapTestData001.bin","Resp001.bin"
```

First Line (L1) : Up date header information  
Second Line (L2) : Write 1024 byte data to start address 0x1000000  
Third Line (L3) : Read 1024 byte data from start address 0x1000000  
Forth Line (L4) : Compere write data and read data.

Excute test script when click " ExecMacro" button.

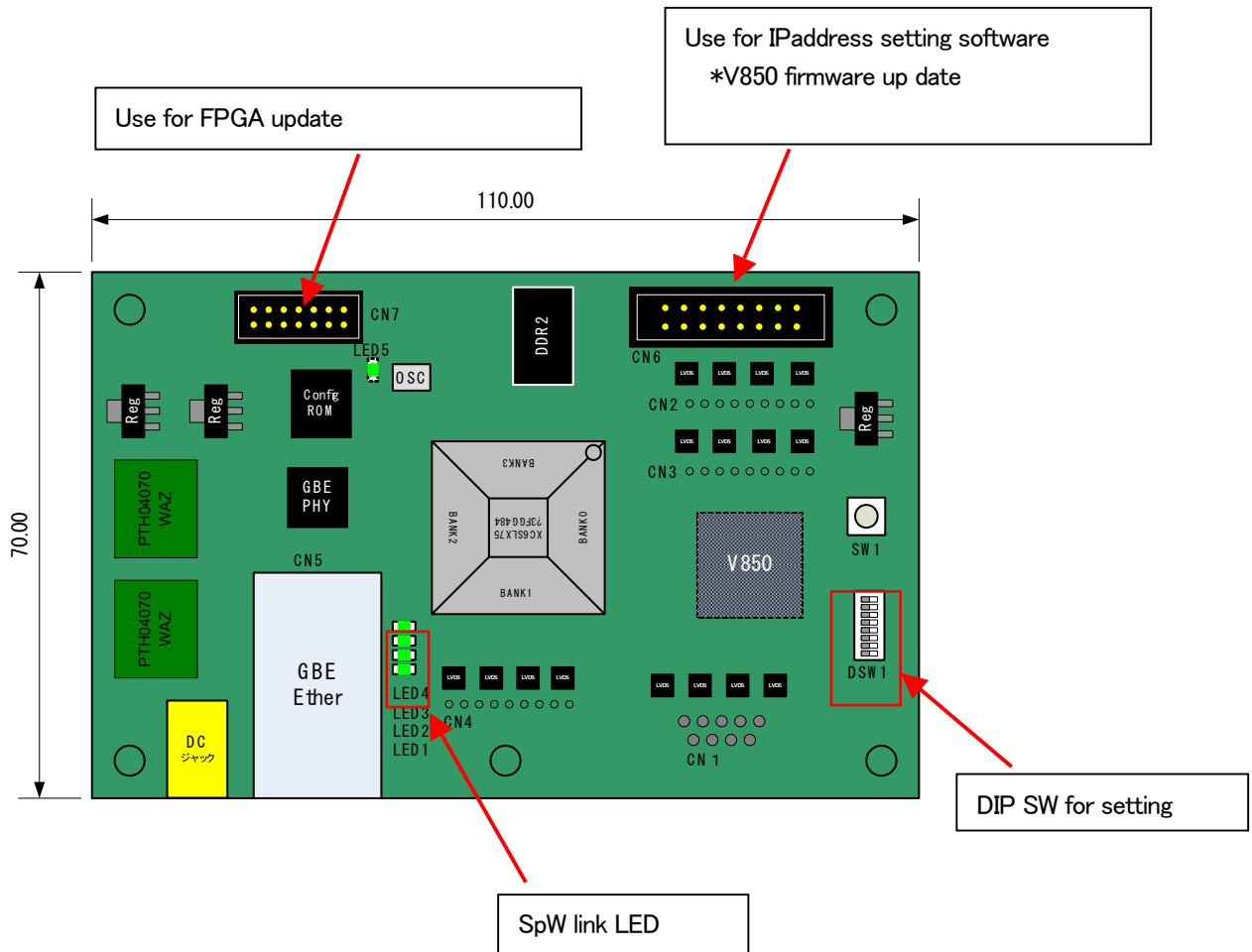
The result od test will display in " Communication-Monitor".

```
L1: OK: read hedder
L2: OK: write command
L3: OK: read command
L4: OK: Compare
```

If "END" is written in fifth line, Test will be terminate.

If there are no "END" end of the script, test repeat from 1st line

## 6. PRINT CIRCUIT BOARD TOP VIEW



## 7. EXTERNAL VIEW

STANDALONE SpaceWire to Gigabit Ether case external view.

