User Manual
for the
HDLC Serial Board (PMC)

Shinbo Documents No	SB_PW-13023(PMC)
Document issue	0.94
Issue Date	2012-01-31
Print Date	
File Name	vxhsbpmc.obj
Distribution List No.	

Signature Sheet

Name	Signature	Date
Completed by	Project Engineer	
Accepted by	Project Manager	
Accepted by		

History

Issue	Desc	Date	Note
1.0	First		

Tables

Abbreviation and Acronyms

API	Application Program Interface
BIT	Built-In-Test
CRC	Cyclic Redundancy Check
HDLC	High Level Data Link Control
HSB	HDLC Serial Board
PCI	Peripheral Component Interconnect
PMC	PCI Mezzanine Card

1. HDLC Serial Board

HDCL Serial Board는 동기식과 비동기식을 지원하는 4 채널 시리얼 통신카드이다.
주 장치와의 Interface는 PMC(PCI Mezzanine Card) 형태를 따른다. PCI Interface는 "33MHz, 32Bit Bus Target Specification V 2.2" 을 지원한다. 사용자 시스템은 기본적으로 PCI 33MHz을 지원하여야 하며, 32비트 체계를 지원하는 시스템 이어야 한다. PMC의 P4를 통해서 데이터 입출력이 가능하며, 전면부(front) 쪽으로 68핀의 입출력 포트를 지원하며, 후면부(Rare) 쪽으로 64 핀을 지원한다.

HDLC Serial Board는 기본적으로 동기식 모드를 지원하며, 추가적으로 비동기 모드를 지원한다. 동기식 모드에서는 HDLC(High-level Data Link Control) Protocol을 지원하며, **ISO 3309**에 의한 Frame Structure을 따르며, ISO/IEC 13239(High-level data link control procedures) 규약의 일부를 지원한다.

HDLC Serial Board는 V.11, RS-530, RS-530A, X.21, V.35, V36, RS-232(V.28)과 다양한 Physical layer protocol을 사용할 수 있다.

HSB는 기본적으로 Motorola사가 제공하는 Single Board Computer인 MVME-6100/MVME-5500에서 사용 가능하다.

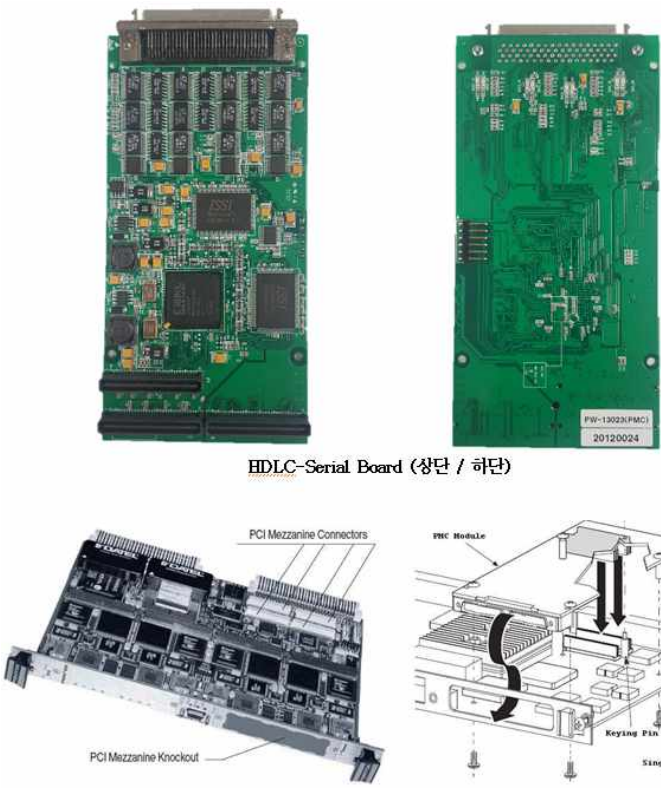

그림 1 HDLC Serial Board

2． HDLC Serial Board(HSB) Register

HDLC Serial Board의 동작과 설정을 담당하는 Register 들은 다음과 같다.
Offset value는 사용자 시스템에서 제공하는 PCI Configuration에 따른 base address에 기본으로 한다.

2－1．HSB Register Map

다음은 HSB에서 사용되는 global register 및 Interrupt 관련 register 이다.

Offset	Registers	
0x104	Interrupt enable/disable register.	IER
0x110	Channel 1 interrupt mask enable/disable register.	IMR1
0x114	Channel 2 interrupt mask enable/disable register.	IMR2
0x118	Channel 3 interrupt mask enable/disable register.	IMR3
0x11C	Channel 4 interrupt mask enable/disable register.	IMR4
0x120	Channel 1 interrupt status register.	IIR1
0x124	Channel 2 interrupt status register.	IIR2
0x128	Channel 3 interrupt status register.	IIR3
0x12C	Channel 4 interrupt status register.	IIR4
0x300	Channel Select register.	CSR
0x7A0	Channel 1 Tx interrupt mask enable/disable register.	ITM1
0x7A4	Channel 1 Rx interrupt mask enable/disable register.	IRM1
0x7A8	Channel 2 Tx interrupt mask enable/disable register.	ITM2
0x7AC	Channel 2 Rx interrupt mask enable/disable register.	IRM2
0x7B0	Channel 3 Tx interrupt mask enable/disable register.	ITM3
0x7B4	Channel 3 Rx interrupt mask enable/disable register.	IRM3
0x7B8	Channel 4 Tx interrupt mask enable/disable register.	ITM4
0x7BC	Channel 4 Rx interrupt mask enable/disable register.	IRM4
0x7F0	HSB version register.	HVR

표 **1. global register** 및 **Interrupt** 관련 **register**

다음은 Serial Communication Controller 관련된 register들이다.
Offset 항목의 n은 channel에 따른 값이며 1번 채널의 값은 3이며 2번 채널은 4, 3번 채널은 5, 4번 채널은 6이다. 예를 들어 1번 채널 SCC의 Baud rate register의 Offset 값은 0x304의 값이된다.

Offset	Registers	
0x0*n*04	Baud rate register.	BRR
0x0*n*08	Line Control Register.	LCR
0x0*n*0C	Physical Mode select Register	PMR
0x0*n*10	Synchronous Transmit Control Register.	TXR
0x0*n*14	Synchronous Transmit Address field Register.	TAR
0x0*n*18	Synchronous Transmit Control field Register.	TCR
0x0*n*1C	Synchronous Transmit Frame Length register.	TFL
0x0*n*20	Transmit Data/Receiving Data Register.	RWR
0x0*n*24	Flow Control Register.	MCR
0x0*n*28	Flow Control Status Register.	MSR
0x0*n*2C	Line Status Register.	LSR
0x0*n*30	Synchronous Receive Control Register.	RXR
0x0*n*34	Synchronous Receive Address field Register.	RAR
0x0*n*38	Synchronous Receive Control field Register	RCR
0x0*n*40	FIFO Control/Status register.	FCR
0x0*n*44	Sync/Async Communication Mode Select Register.	SMS

0x0*n*48	Synchronous Receive FCNs(CRC) Register.	sRWR
0x0*n*4C	Synchronous Transmit Pause Length register.	TPL
0x0*n*50	Received Address Mask Register.	RAM
0x0*n*54	Frame encoding Register.	TFE
0x0*n*58	Sync preamble option register.	PRE
0x0*n*60	Sync Flag Register.	SFR
0x0*n*64	Received Timeout Register.	RTM
0x0*n*68	Internal Loopback Register.	ILB

표 **2. Serial Communication Controller** 관련 **register**

다음은 FiFo Counting 관련된 Register들이다.

Offset	**Registers**	
0x710	Channel 1 Received FIFO Read Pointer Register.	FRRP
0x720	Channel 2 Received FIFO Read Pointer Register.	FRRP
0x730	Channel 3 Received FIFO Read Pointer Register.	FRRP
0x740	Channel 4 Received FIFO Read Pointer Register.	FRRP
0x714	Channel 1 Received FIFO Write Pointer Register.	FRWP
0x724	Channel 2 Received FIFO Write Pointer Register.	FRWP
0x734	Channel 3 Received FIFO Write Pointer Register.	FRWP
0x744	Channel 4 Received FIFO Write Pointer Register.	FRWP
0x718	Channel 1 Transmit FIFO Read Pointer Register.	FTRP
0x728	Channel 2 Transmit FIFO Read Pointer Register.	FTRP
0x738	Channel 3 Transmit FIFO Read Pointer Register.	FTRP
0x748	Channel 4 Transmit FIFO Read Pointer Register.	FTRP
0x71C	Channel 1 Transmit FIFO Write Pointer Register.	FTWP
0x72C	Channel 2 Transmit FIFO Write Pointer Register.	FTWP
0x73C	Channel 3 Transmit FIFO Write Pointer Register.	FTWP
0x74C	Channel 4 Transmit FIFO Write Pointer Register.	FTWP

표 **3. FiFo Counting**

2-2. HSB Register Description

2-2-1. Interrupt Enable/Disable Register (IER)

Offset 0x104
Size 32Bit
I/O Read/Write
DESC IER Register 는 HSB의 interrupt를 enable/disable 한다. 각각의 채널 별로 8bit씩 할당되어 있으며 하위 2bit을 이용하여 Reception(RX) interrupt와 Transmission(TX) interrupt enable/disable 한다.

bit / offset	31 24	23 16	15 08	07 00
0x104	**CH4 IER**	**CH3 IER**	**CH2 IER**	**CH1 IER**

BIT	**DESC**
IER[7-2]	Reserved
IER[1]	Receive interrupt enables or disables 1: Rx Interrupt Enable 0: Rx Interrupt Disable
IER[0]	Transmit Interrupt enables or disables 1: Tx Interrupt Enable 0: Tx Interrupt Disable

2-2-2. Interrupt Mask Enable/Disable Register (IMR*n*)

Offset 0x110(CH1), 0x114(CH2), 0x118(CH3), 0x11C(CH4)
Size 32Bit
I/O Read/Write
DESC IMR Register는 RX interrupt 와 TX interrupt 의 각 항목에 대한 Mask를 enable/disable 한다. 하위 16Bit을 사용하여 16 bit의 상위 8비트는 RX interrupt mask를, 하위 8비트는 TX interrupt mask을 다룬다.

bit / offset	31 24	23 16	15 08	07 00
0x110(CH1)	x	x	**IMR1 RX**	**IMR1 TX**
0x114(CH2)	x	x	**IMR2 RX**	**IMR2 TX**
0x118(CH3)	x	x	**IMR3 RX**	**IMR3 TX**
0x11C(CH4)	x	x	**IMR4 RX**	**IMR4 TX**

BIT	DESC
IMR[15]	RX timeout interrupt mask.
IMR[14]	RX error interrupt mask.
IMR[13]	RX synchronous character detected interrupt mask.
IMR[12]	RX data available interrupt mask.
IMR[11]	Reserved
IMR[10]	RX buffer full interrupt mask.
IMR[9]	RX buffer frame interrupt mask.
IMR[8]	RX buffer empty interrupt mask.
IMR[7]	TX timeout interrupt mask.
IMR[6]	Reserved
IMR[5]	Reserved
IMR[4]	Reserved
IMR[3]	Reserved
IMR[2]	TX done interrupt mask.
IMR[1]	TX complete interrupt mask.
IMR[0]	TX buffer empty interrupt mask.

2-2-3. Interrupt Indicator Register (IIR*n*)

Offset 0x120(CH1), 0x124(CH2), 0x128(CH3), 0x12C(CH4)
Size 32Bit
Read only
DESC IIR Register는 IER과 IMR이 세트되어 있을 경우에 interrupt source 의 상태를 확인한다.

bit / offset	31 24	23 16	15 08	07 00
0x120(CH1)	x	x	**IIR1 RX**	**IIR1 TX**
0x124(CH2)	x	x	**IIR2 RX**	**IIR2 TX**
0x128(CH3)	x	x	**IIR3 RX**	**IIR3 TX**
0x12C(CH4)	x	x	**IIR4 RX**	**IIR4 TX**

BIT	DESC
IIR[15]	RX timeout interrupt status.
IIR[14]	RX error interrupt status.

IIR[13]	RX synchronous character detected interrupt status.
IIR[12]	RX data available interrupt status.
IIR[11]	Reserved
IIR[10]	RX buffer full interrupt status.
IIR[9]	RX buffer frame interrupt status.
IIR[8]	RX buffer empty interrupt status.
IIR[7]	TX timeout interrupt status.
IIR[6]	Reserved
IIR[5]	Reserved
IIR[4]	Reserved
IIR[3]	Reserved
IIR[2]	TX done interrupt status.
IIR[1]	TX complete interrupt status.
IIR[0]	TX buffer empty interrupt status.

2－2－4．Channel(Port) Select Register (CSR)

Offset 0x300
Size 32Bit
I/O Read/Write
DESC CSR register는 Channel의 physical layer protocol을 선택하거나 변경할 경우, 각 채널의 CSR register를 On하여야 한다.

bit / offset	31 24	23 16	15 08	07 00
0x300(CSR)	x	x	x	**CSR**

BIT	DESC
CSR[7]	CH4 Channel On(High)/Off(Low)
CSR[6]	CH3 Channel On(High)/Off(Low)
CSR[5]	CH2 Channel On(High)/Off(Low)
CSR[4]	CH1 Channel On(High)/Off(Low)
CSR[3]	CH4 Channel Select On(High)/Off(Low)
CSR[2]	CH3 Channel Select On(High)/Off(Low)
CSR[1]	CH2 Channel Select On(High)/Off(Low)
CSR[0]	CH1 Channel Select On(High)/Off(Low)

11h-ch0, 0x00-ch5(Engineering Terminal)
22h-ch1, if all channel set : FFh
44h-ch2,
88h-ch3,

2－2－5．Baud Rate Register (BRR*n*)

Offset 0x304(CH1), 0x404(CH2), 0x504(CH3), 0x604(CH4)
Size 32Bit
I/O Read/Write
DESC BRR register를 통해서 각 채널의 통신 속도를 설정한다.

bit / offset	31 24	23 16	15 08	07 00
0x304(CH1)	x	x	x	**BRR1**

0x404(CH2)	x	x	x	**BRR2**
0x504(CH3)	x	x	x	**BRR3**
0x604(CH4)	x	x	x	**BRR4**

Baud Rate Formula
Clock Freq : 14745600 Hz **Baud rate: 115200 bps** **Synchronous mode : DIV = Clock Freq/Baud rate** **Asynchronous mode: DIV = Clock Freq/(Baud rate *16)**

Standard Clock		External Clock	
Asynchronous	synchronous	Asynchronous	synchronous
4800			
9600			
14400			
28800			

2－2－6．Line Control Register (LCR*n*)

Offset 0x308(CH1), 0x408(CH2), 0x508(CH3), 0x608(CH4)
Size 32Bit
I/O Read/Write
DESC LCR register는 비동기 UART 모드에서 각 채널의 전송형태를 결정한다.

bit / offset	31 24	23 16	15 08	07 00
0x308(CH1)	x	x	x	**LCR1**
0x408(CH2)	x	x	x	**LCR2**
0x508(CH3)	x	x	x	**LCR3**
0x608(CH4)	x	x	x	**LCR4**

BIT	DESC
LCR[7]	Reserved
LCR[6]	Reserved
LCR[5:4]	11: Space Parity 10: Mark Parity 01: Even Parity 00: Odd Parity
LCR[3]	Parity enable/disable bit If **LCR[3]** is **"0",** parity is disable**.** If **LCR[3]** is **"1",** parity is enable**.**
LCR[2]	These bits control stop bit If **LCR[2]** is **"0",** stop bit is 1 bit**.** If **LCR[2]** is **"1",** stop bit is 2 bits**.**
LCR[1:0]	These bits control data length. 11:8bit 10:7bit 01:6bit 00:5bit

2-2-7. Physical Mode select Register (PMR*n*)

Offset 0x30C(CH1), 0x40C(CH2), 0x50C(CH3), 0x60C(CH4)
Size 32Bit
I/O Read/Write
DESC PMR register는 CSR register가 세트된 후 사용자가 원하는 physical layer를 선택한다. 사용하고자 하는 Board system clock을 선택할 경우도 PMR register의 해당 비트를 세트 한다.
Default: 0x00000016

bit / offset	31 24	23 16	15 08	07 00
0x30C(CH1)	x	x	x	**PMR1**
0x40C(CH2)	x	x	x	**PMR2**
0x50C(CH3)	x	x	x	**PMR3**
0x60C(CH4)	X	x	x	**PMR4**

BIT	DESC
PMR[7:6]	These bits field is board system clock. 00: Standard Clock[14.7456MHz] 01: External Clock 10: User Clock – default: [16.0000MHz]
PMR[5]	Reserved.
PMR[4]	This bit set DTE/DCE. 0: DTE (Data Terminal Equipment, Receiver) 1: DCE (Data Circuit-terminating Equipment, Driver)
PMR[3]	Reserved
PMR[2:0]	This bit field determines the physical layer protocol. PMR[2:0] = ‘ 000’ is V.11 PMR[2:0] = ‘ 001’ is RS- 530A PMR[2:0] = ‘ 010’ is RS-530 PMR[2:0] = ‘ 011’ is X.21 PMR[2:0] = ‘ 100’ is V.35 PMR[2:0] = ‘ 101’ is V.36 PMR[2:0] = ‘ 110’ is RS-232(V.28)

❾ External Clock Mode와 User Clock Mode가 동시에 설정될 경우에는 External Clock Mode가 우선권을 가진다.

2-2-8. Synchronous Transmit Control Register (TXR*n*)

Offset 0x310(CH1), 0x410(CH2), 0x510(CH3), x610(CH4)
Size 32Bit
I/O Read/Write
DESC This register is set to control a HDLC transmit frame.
Address field, Control field, Preamble and Pause time HDLC transmit frame are controlled.

bit / offset	31 24	23 16	15 08	07 00
0x310(CH1)	x	x	x	**TXR1**
0x410(CH2)	x	x	x	**TXR2**
0x510(CH3)	x	x	x	**TXR3**
0x610(CH4)	x	x	x	**TXR4**

BIT	DESC
TXR[7]	This bit enables or disables the preamble when transmitting data. TXR[7] is set to ‘ 1’ , if transmit preamble is used. TXR[7] is set to ‘ 0’ , if transmit preamble is not used.
TXR[6:5]	This bit field is only valid if the TXR[7] is ‘ 1’ . This bit field determines the preamble length when transmitting data. TXR[6:5]= ‘ 00’ – 1 byte TXR[6:5]= ‘ 01’ – 2 byte TXR[6:5]= ‘ 10’
TXR[4]	This bit enables or disables the transmit pause. TXR[4] is set to ‘ 1’ , if transmit pause is used. TXR[4] is set to ‘ 0’ , if transmit pause is not used.
TXR[3]	This bit enables or disables the **control field** of a HDLC frame when transmitting data. TXR[3] is set to ‘ 1’ if the control filed of a HDLC transmit frame is used. TXR[3] is set to ‘ 0’ if the control filed of a HDLC transmit frame is not used.
TXR[2]	This bit is only valid if TXR[3] is ‘ 1’ . This bit determines the Transmit control field length of HDLC frame. TXR[2] is set to ‘ 1’ , if the control field length of HDLC transmit frame is 16 bits. TXR[2] is set to ‘ 0’ , if the control field length of HDLC transmit frame is 8 bits.
TXR[1]	This bit enables or disables the **address field** of a HDLC frame when transmitting data. TXR[1] is set to ‘ 1’ if that enable the address field is used a HDLC transmit frame. TXR[1] is set to ‘ 0’ , if the address field is not used a HDLC transmit frame.
TXR[0]	This bit is only valid if TXR[1] is ‘ 1’ . This bit determines the address field length of a HDLC frame when transmitting data. TXR[0] is set to ‘ 1’ , if the address field length of a HDLC transmit frame is 16 bits. TXR[0] is set to ‘ 0’ , if the address field length of a HDLC transmit frame is 8 bits.

2-2-9. synchronous Transmit Address field Register (TAR*n*)

Offset 0x314(CH1), 0x414(CH2), 0x514(CH3), 0x614(CH4)
Size 16 bits or 8 bits.
I/O Read/Write
DESC This register set to a target address.
A target address is inserted into the address field of a HDLC transmit frame.

bit / offset	31 24	23 16	15 08	07 00
0x314(CH1)	x	x	**TAR1(H)**	**TAR1(L)**
0x414(CH2)	x	x	**TAR2(H)**	**TAR2(L)**
0x514(CH3)	x	x	**TAR3(H)**	**TAR3(L)**
0x614(CH4)	x	x	**TAR4(H)**	**TAR4(L)**

BIT	DESC
TAR(H)[15:8]	전송하는 HDLC 프레임의 Address 필드 값의 상위 바이트.
TAR(L)[7:0]	전송하는 HDLC 프레임의 Address 필드 값의 하위 바이트

2-2-10. synchronous Transmit Control field Register (TCR*n*)

Offset 0x318(CH1), 0x418(CH2), 0x518(CH3), 0x618(CH4)
Size 16 bits or 8bits
I/O Read/Write
DESC This register set to a control command.
A control command is inserted into the control field of a HDLC transmit frame.

bit / offset	31 24	23 16	15 08	07 00
0x318(CH1)	x	x	**TCR1(H)**	**TCR1(L)**
0x418(CH2)	x	x	**TCR2(H)**	**TCR2(L)**
0x518(CH3)	x	x	**TCR3(H)**	**TCR3(L)**
0x618(CH4)	x	x	**TCR4(H)**	**TCR4(L)**

BIT	DESC
TCR[15:8]	전송하는 HDLC 프레임의 Control/Command 필드 값의 상위 바이트
TCR[7:0]	전송하는 HDLC 프레임의 Control/Command 필드 값의 하위 바이트

2-2-11. Synchronous Transmit Frame Length register (TFL*n*)

Offset 0x31C(CH1), 0x41C(CH2), 0x51C(CH3), x61C(CH4)
Size 32Bit
I/O Read/Write
DESC Synchronous HDLC mode에서 전송할 Frame의 길이를 설정하는 register.

bit / offset	31 24	23 16	15 08	07 00
0x318(CH1)	x	x	**TFL1**	
0x418(CH2)	x	x	**TFL2**	
0x518(CH3)	x	x	**TFL3**	
0x618(CH4)	x	x	**TFL4**	

HEX	DESC
TFL[15:0]	HDLC Frame Length

2-2-12. Transmit Data/Receiving Data Register (RWR*n*)

Offset 0x320(CH1), 0x420(CH2), 0x520(CH3), 0x620(CH4)
Size 32Bit
I/O Read/Write
DESC This register reads or writes a data in the received or transmits FiFo.

bit / offset	31 24	23 16	15 08	07 00
0x320(CH1)	x	x	x	**RWR1**
0x420(CH2)	x	x	x	**RWR2**
0x520(CH3)	x	x	x	**RWR3**
0x620(CH4)	x	x	x	**RWR4**

BIT	DESC

RWR[7:0]	When HSB has been Received data, this register read data from FiFo
	When HSB transmit data, this register write data to FiFo

2-2-13. Flow Control Register (MCR*n*)

Offset 0x324(CH1), 0x424(CH2), 0x524(CH3), 0x624(CH4)
Size 32Bit
I/O Read/Write
DESC This register control Hardware line flow control.

bit / offset	31 24	23 16	15 08	07 00
0x324(CH1)	x	x	x	**MCR1**
0x424(CH2)	x	x	x	**MCR2**
0x524(CH3)	x	x	x	**MCR3**
0x624(CH4)	x	x	x	**MCR4**

BIT	DESC
MCR[7]	Ring indicator
MCR[6]	DCD indicator
MCR[5]	DSR indicator
MCR[4]	CTS indicator
MCR[3]	RTS
MCR[2]	DTR
MCR[1]	Auto flow control : 1, Manual flow control: 0
MCR[0]	If flow Control is On, this bit set "1". Else this bit set to "0"

2-2-14. Flow Control Status Register (MSR*n*)

Offset 0x328(CH1), 0x428(CH2), 0x528(CH3), 0x628(CH4)
Size 32 bit
I/O Read/Write
DESC This register show a status of line flow control.

bit / offset	31 24	23 16	15 08	07 00
0x328(CH1)	x	x	x	**MSR1**
0x428(CH1)	x	x	x	**MSR2**
0x528(CH1)	x	x	x	**MSR3**
0x628(CH1)	x	x	x	**MSR4**

BIT	DESC
MSR[7]	DCD input
MSR[6]	RI input
MSR[5]	DSR input
MSR[4]	CTS input
MSR[3]	DCD change
MSR[2]	RI change
MSR[1]	DSR change
MSR[0]	CTS change

2-2-15. Line Status Register (LSR*n*)

Offset 0x32c(CH1), 0x42c(CH2), 0x52c(CH3), 0x62c(CH4)
Size 32 bit
I/O Read/Write
DESC This register is the state of Line. Check the bit of FiFo status, Error.

bit / offset	31 24	23 16	15 08	07 00
0x32C(CH1)	x	x	x	**LSR1**
0x42C(CH2)	x	x	x	**LSR2**
0x52C(CH3)	x	x	x	**LSR3**
0x62C(CH4)	x	x	x	**LSR4**

Asynchronous communication mode.

BIT	DESC
LSR[7]	Tx FiFo overflow error
LSR[6]	Tx empty
LSR[5]	Reserved
LSR[4]	Break
LSR[3]	Byte frame error
LSR[2]	Parity error
LSR[1]	Rx FiFo overflow error
LSR[0]	Rx data available

Synchronous communication mode.

BIT	DESC
LSR[7]	Tx FiFo overflow error.
LSR[6]	Tx done. This bit is that Tx FiFo is empty.
LSR[5]	Reserved
LSR[4]	Rx abort.
LSR[3]	Rx frame error.
LSR[2]	Rx FCS Error (CRC error).
LSR[1]	Rx FiFo Overflow Error.
LSR[0]	Rx data available. This bit is that Rx FiFo not empty.

2-2-16. Synchronous Receive Control Register (RXR*n*)

Offset 0x330(CH1), 0x430(CH2), 0x530(CH3), 0x630(CH4)
Size 32Bit
I/O Read/Write
DESC This register is set to control a HDLC transmit frame.
Address field and Control field of received HDLC frame are controlled.

bit / offset	31 24	23 16	15 08	07 00
0x330(CH1)	x	x	x	**RXR1**
0x430(CH2)	x	x	x	**RXR2**
0x530(CH3)	x	x	x	**RXR3**
0x630(CH4)	x	x	x	**RXR4**

BIT	DESC
RXR[7]	If this bit is set to 1, the control field of a HDLC frame is filled into FiFo.
RXR[6]	This bit is only valid if RXR[5] is ‘ 1’ . This bit determines the **control field length** of HDLC frame. RXR[6] is set to ‘ 1’ , if the control field length of a received HDLC frame is **16 bits**. RXR[6] is set to ‘ 0’ , if the control field length of a received HDLC frame is **8 bits**.
RXR[5]	This bit enables or disables **the control field** of a received HDLC frame. RXR[5] is set to ‘ 1’ , if the control filed of a received HDLC frame is used. RXR[5] is set to ‘ 0’ , if the control filed of a received HDLC frame is not used.
RXR[4]	Received Uniquest enable bit.
RXR[3]	Received Address Mask enable bit
RXR[2]	Rx Address FIFO enable bit.
RXR[1]	This bit is only valid if RXR[0] is ‘ 1’ . This bit determines the **address field length** of HDLC frame. RXR[1] is set to ‘ 1’ , if the control field length of a received HDLC frame is **16 bits**. RXR[1] is set to ‘ 0’ , if the control field length of a received HDLC frame is **8 bits**.
RXR[0]	This bit enables or disables **the address field** of a received HDLC frame. TXR[0] is set to ‘ 1’ , if the address filed of a received HDLC frame is used. TXR[0] is set to ‘ 0’ , if the address filed of a received HDLC frame is not used.

2-2-17. Synchronous Receive Address field Register (RAR*n*)

Offset 0x334(CH1), 0x434(CH2), 0x534(CH3), 0x634(CH4)
Size 32Bit
I/O Read/Write
DESC This register is the HSB address on synchronous HDLC mode.

bit / offset	31 24	23 16	15 08	07 00
0x334(CH1)	x	X	**RAR1**	
0x434(CH2)	x	X	**RAR2**	
0x534(CH3)	x	X	**RAR3**	
0x634(CH4)	x	X	**RAR4**	

BIT	DESC
RAR[15:8]	HSB의 HDLC Address의 상위 비트
RAR[7:0]	HSB의 HDLC Address의 하위 비트

2-2-18. synchronous Receive Control field Register (RCR)

Offset 0x338(CH1), 0x438(CH2), 0x538(CH3), 0x638(CH4)
Size 32Bit
Read/Write
DESC This Register is the control field of a received HDLC frame

bit / offset	31 24	23 16	15 08	07 00
0x338(CH1)	x	x	**RCR1**	
0x438(CH2)	x	x	**RCR2**	

0x538(CH3)	x	x	**RCR3**
0x638(CH4)	x	x	**RCR4**

BIT	DESC
RCR[15:8]	수신 HDLC 프레임의 Control/Command 필드의 상위 바이트
RCR[7:0]	수신 HDLC 프레임의 Control/Command 필드의 하위 바이트

2-2-19. Synchronous Received Frame Length register (RFL*n*)

Offset 0x33C(CH1), 0x43C(CH2), 0x53C(CH3), 0x63C(CH4)
Size 32Bit
I/O Read/Write
DESC This Register is that the Length a received HDLC frame.

bit / offset	31 24	23 16	15 08	07 00
0x33C(CH1)	x	x	**RFL1**	
0x43C(CH2)	x	x	**RFL2**	
0x53C(CH3)	x	x	**RFL3**	
0x63C(CH4)	x	x	**RFL4**	

2-2-20. FIFO Control/Status register (FCR*n*)

Offset 0x340(CH1), 0x440(CH2), 0x540(CH3), 0x640(CH4)
Size 32Bit
I/O Read/Write
DESC This Register is that control or show the FiFo.

bit / offset	31 24	23 16	15 08	07 00
0x340(CH1)	x	x	x	**FCR1**
0x440(CH2)	x	x	x	**FCR2**
0x540(CH3)	x	x	x	**FCR3**
0x640(CH4)	x	x	x	**FCR4**

BIT	DESC
FCR[7]	Rx FiFo clear bits If bit is set “1”, you can clear Rx FiFo.
FCR[6]	Rx FiFo empty status bit This bit is set “1”, if Rx buffer is empty.
FCR[5]	Rx FiFo half full status bit This bit is set “1”, if Rx buffer is half full.
FCR[4]	Rx FiFo full status bit This bit is set “1”, if Rx buffer is full.
FCR[3]	Tx FiFo clear If bit is set “1”, you can clear Tx FiFo.
FCR[2]	Tx FiFo empty bit This bit is set “1”, if Tx buffer is empty.
FCR[1]	Tx FiFo half full bit This bit is set “1”, if Tx buffer is half full.
FCR[0]	Tx FiFo full bit This bit is set “1”, if Tx buffer is full.

2-2-21. Sync/Async Communication Mode Select Register (SMS*n*)

Offset 0x344(CH1), 0x444(CH2), 0x544(CH3), 0x644(CH4)
Size 32Bit
I/O Read/Write
DESC This register is used that select mode of Synchronous communication or Asynchronous communication. Can enable or disable FiFo.
Default – 0x00000010

bit / offset	31 24	23 16	15 08	07 00
0x344	x	x	x	**SMS1**
0x444	x	x	x	**SMS2**
0x544	x	x	x	**SMS3**
0x644	x	x	x	**SMS4**

BIT	DESC
SMS[7]	Reserved
SMS[6]	Reserved
SMS[5]	Reserved
SMS[4]	enable/disable FiFo bit. **SMS[4]** = **"0"** is that FiFo disable. **SMS[4]** = **"1"** is that FiFo enable.
SMS[3]	Reserved
SMS[2]	Reserved
SMS[1]	Reserved
SMS[0]	Sync/Async Mode Select bit. **SMS[0]** = **"0"** is Asynchronous communication mode **SMS[0]** = **"1"** is synchronous communication mode

2-2-22. Synchronous Receive FCNs(CRC) Register (sRWR*n*)

Offset 0x348(CH1), 0x448(CH2), 0x548(CH3), 0x648(CH4)
Size 32Bit
I/O Read/Write
DESC RWR register는 각 채널의 RX/TX을 enable/disable 할 수 있으며 FCS 모드를 설정할 수 있다.

bit / offset	31 24	23 16	15 08	07 00
0x348(CH1)	x	x	x	**sRWR1**
0x448(CH2)	x	x	x	**sRWR2**
0x548(CH3)	x	x	x	**sRWR3**
0x648(CH4)	x	x	x	**sRWR4**

BIT	DESC
sRWR[7]	Tx FCS mode select bit **sRWR[7] = "1"**이면 Tx FCS에 **CRC32** 모드를 사용하며, **"0"** 이면 **CRC16** 모드를 사용한다.
sRWR[6]	Tx FCS enable bit **sRWR[6] = "1"**이면 HDLC 모드에서 Frame의 Tx FCS에 CRC를 사용하지 않고 전송한다. 주로 Burst모드 전송에서 사용된다.
sRWR[5]	Tx Abort bit
sRWR[4]	Tx enable bit Synchronous 모드에서 Tx를 enable/disable 할 수 있다.

sRWR[3]	Rx FCS mode select bit **sRWR[3]** = **"1"**이면 Rx FCS에 **CRC32** 모드를 사용하며, **"0"** 이면 **CRC16** 모드를 사용한다.
sRWR[2]	Rx FCS enable bit **sRWR[2]** = **"1"**이면 HDLC 모드에서 Frame에서 CRC를 사용하지 않는다.
sRWR[1]	Reserved
sRWR[0]	Rx Enable bit Synchronous 모드에서 Rx를 enable/disable 할 수 있다.

2-2-23. Synchronous Transmit Pause Length register (TPL*n*)

Offset 0x34C(CH1), 0x44C(CH2), 0x54C(CH3), 0x64C(CH4)
Size 32Bit
I/O Read/Write
DESC

bit / offset	31 24	23 16	15 08	07 00
0x34C	x	x	x	**TPL1**
0x44C	x	x	x	**TPL2**
0x54C	x	x	x	**TPL3**
0x64C	x	x	x	**TPL4**

BIT	DESC
TFL[7:0]	Synchronous Transmit Pause Length

2-2-24. Received Address Mask Register (RAM*n*)

Offset 0x350(CH1), 0x450(CH2), 0x550(CH3), 0x650(CH4)
Size 32Bit
Read/Write
DESC 수신할 HDLC frame의 주소 대역을 설정하는 register이다.

bit / offset	31 24	23 16	15 08	07 00
0x350	x	x	**RAM1**	
0x450	x	x	**RAM2**	
0x550	x	x	**RAM3**	
0x650	x	x	**RAM4**	

BIT	DESC
RAM[15:8]	Synchronous Received Mask Address High Bit
RAM[7:0]	Synchronous Received Mask Address Low Bit

2-2-25. Frame encoding Register (TFE*n*)

Offset 0x354(CH1), 0x450(CH2), 0x550(CH3), 0x650(CH4)
Size 32Bit
I/O Read/Write
DESC This register is used that select a Frame Encoding Mode when HDLC mode
Default – 0x00000000

bit / offset	31 24	23 16	15 08	07 00
0x354	x	x	x	**TFE1**
0x454	x	x	x	**TFE2**
0x554	x	x	x	**TFE3**
0x654	x	x	x	**TFE4**

BIT	DESC
TFE[7]	Reserved
TFE[6]	Reserved
TFE[5]	Reserved
TFE[4]	Reserved
TFE[3:0]	0x08(1000) **Diff_Manchester** 0x04(0100) **Manchester** 0x02(0010) **FFM0** 0x01(0001) **NRZI** 0x00(0000) **NRZ(default)**

2－2－26. Sync preamble option register (PRE*n*)

Offset 0x358(CH1), 0x458(CH2), 0x558(CH3), 0x658(CH4)
Size 32Bit
I/O Read/Write
DESC This register is used that preamble value.

bit / offset	31 24	23 16	15 08	07 00
0x358(CH1)	x	x	x	**PRE1**
0x458(CH2)	x	x	x	**PRE2**
0x558(CH3)	x	x	x	**PRE3**
0x658(CH4)	x	x	x	**PRE4**

BIT	DESC
PRE[7:0]	Preamble value.

2－2－27. Sync Flag Register (SFR*n*)

Offset 0x360(CH1), 0x460(CH2), 0x560(CH3), 0x660(CH4)
Size 32Bit
I/O Read/Write
DESC This register used to set the flag value of Synchronous communication.

bit / offset	31 24	23 16	15 08	07 00
0x360(CH1)	x	x	x	**SFR1**
0x460(CH2)	x	x	x	**SFR2**
0x560(CH3)	x	x	x	**SFR3**
0x660(CH4)	x	x	x	**SFR4**

BIT	DESC
SFR[7:0]	This bits field set the flag value of Synchronous communication.

2-2-28. Received Timeout Register (RTM*n*)

Offset 0x364(CH1), 0x464(CH2), 0x564(CH3), 0x664(CH4)
Size 32Bit
I/O Read/Write
DESC This register is used to value of RX Timeout interrupt.

bit / offset	31 24	23 16	15 08	07 00
0x364	x	x	x	**RTM1**
0x464	x	x	x	**RTM2**
0x564	x	x	x	**RTM3**
0x664	x	x	x	**RTM4**

BIT	DESC
RTM[31:0]	value of RX Timeout interrupt

2-2-29. Internal Loopback Register (ILB*n*)

Offset 0x368(CH1), 0x468(CH2), 0x568(CH3), 0x668(CH4)
Size 32Bit
I/O Read/Write
DESC This register used to internal loopback enable/disable.

bit / offset	31 24	23 16	15 08	07 00
0x364(CH1)	x	x	x	**ILB1**
0x460(CH2)	x	x	x	**ILB2**
0x560(CH3)	x	x	x	**ILB3**
0x660(CH4)	x	x	x	**ILB4**

BIT	DESC
ILB[7:1]	Reserved
ILB[0]	This bit set “1”, if internal loop back enable. This bit set “0, if internal loop back disable.

2-2-30. Received FIFO Read Pointer Register (FRRP*n*)

Offset 0x710(CH1), 0x720(CH2), 0x730(CH3), 0x740(CH4)
Size 32Bit
I/O Read/Write
DESC Read pointer of Rx Circular buffer. 수신부의 FiFo에서 사용자 시스템이 읽어가는 데이터의 카운터를 나타낸다.

bit / offset	31 24	23 16	15 08	07 00
0x710(CH1)	x	X	x	**FRRP1**
0x720(CH2)	x	X	x	**FRRP2**
0x730(CH3)	x	X	x	**FRRP3**
0x740(CH4)	x	X	x	**FRRP4**

2-2-31. Received FIFO Write Pointer Register (FRWP*n*)

Offset 0x714(CH1), 0x724(CH2), 0x734(CH3), 0x744(CH4)
Size 32Bit
I/O Read/Write
DESC Write pointer of Rx Circular buffer. 수신부의 FiFo로 들어오는 데이터의 카운터를 나타낸다.

bit / offset	31 24	23 16	15 08	07 00
0x714(CH1)	x	X	x	**FRWP1**
0x724(CH2)	x	X	x	**FRWP2**
0x734(CH3)	x	X	x	**FRWP3**
0x744(CH4)	x	x	x	**FRWP4**

2-2-32. Transmit FIFO Read Pointer Register (FTRP*n*)

Offset 0x718(CH1), 0x728(CH2), 0x738(CH3), 0x748(CH4)
Size 32Bit
I/O Read/Write
DESC Read pointer of Tx Circular buffer. 송신부의 FiFo에서 통신상으로 출력되는 데이터의 카운터를 나타낸다.

bit / offset	31 24	23 16	15 08	07 00
0x718(CH1)	x	x	x	**FTRP1**
0x728(CH2)	x	x	x	**FTRP2**
0x738(CH3)	x	x	x	**FTRP3**
0x748(CH4)	x	x	x	**FTRP4**

2-2-33. Transmit FIFO Write Pointer Register (FTWP*n*)

Offset 0x71C(CH1), 0x72C(CH2), 0x73C(CH3), 0x74C(CH4)
Size 32Bit
I/O Read/Write
DESC Write pointer of Tx Circular buffer. 송신부의 FiFo에 사용자 시스템 에서 출력하는 데이터의 카운터를 나타낸다.

bit / offset	31 24	23 16	15 08	07 00
0x71C(CH1)	x	x	x	**FTWP1**
0x72C(CH2)	x	x	x	**FTWP2**
0x73C(CH3)	x	x	x	**FTWP3**
0x74C(CH4)	x	x	x	**FTWP4**

2-2-34. Transmit Interrupt Mask enable/disable register (ITM*n*)

Offset 0x7A0(CH1), 0x7A8(CH2), 0x7B0(CH3), 0x7B8(CH4)
Size 32Bit
I/O Read/Write
DESC .

bit / offset	31 24	23 16	15 08	07 00
0x7A0(CH1)	x	x	x	**ITM1**
0x7A8(CH2)	x	x	x	**ITM2**

0x7B0(CH3)	x	x	x	**ITM3**
0x7B8(CH4)	x	x	x	**ITM4**

2-2-35. Received Interrupt Mask enable/disable register (IRM*n*)

Offset 0x7A4(CH1), 0x7AC(CH2), 0x7B4(CH3), 0x7BC(CH4)
Size 32Bit
I/O Read/Write
DESC .

bit / offset	31 24	23 16	15 08	07 00
0x7A4(CH1)	x	x	x	**IRM1**
0x7AC(CH2)	x	x	x	**IRM2**
0x7B4(CH3)	x	x	x	**IRM3**
0x7BC(CH4)	X	x	x	**IRM4**

2-2-36. HSB Version Register (HVR)

Offset 0x7F0
Size 32Bit
I/O Read/Write
DESC This Register is the HSB internal logic Version Number.

bit / offset	31 24	23 16	15 08	07 00
0x7F0	**HVR**			

3. HDLC Serial Board(HSB) Device Driver.

3-1. Device Driver Install and Remove

3-1-1. HSB Device Driver Install

hsb_create() 함수는 SBC 시스템에 HSB device driver를 install한다.

function **int hsb_create(void)**
purpose
Device driver Install function.
arguments
<void>
returns
OK Device Driver is successfully installed on the system.
ERROR Device Driver is not successfully installed on the system.

hsbdrv_create() 함수는 hsb_create 함수와 동일하게 SBC 시스템에 HSB device driver를 install 한다. Install 이후에 internal loopback 기능을 이용하여 통신이 성공적으로 이루어 지는 가에 대한 검정 절차를 밟는다.

function **int hsbdrv_create(int nBit)**
purpose
Device driver Install function.
arguments
<internal BIT>
"0" – hsb_create와 동일한 기능을 한다.
"1" – internal loopback 기능을 이용한 진단 기능 포함.
returns
OK Device Driver is successfully installed on the system.
ERROR Device Driver is not successfully installed on the system.

Ex)
-> hsb_create()
-> hsbdrv_create(1)

3-1-2. HSB Device Driver Remove

hsb_remove 함수는 SBC 시스템에서 HSB 디바이스 드라이버를 제거하는 기능을 하는 함수이다.

function **int hsb_remove(void)**
purpose
Device driver Install function.
arguments
<void>
returns
OK Device Driver is successfully removed on the system.
ERROR Device Driver is not successfully removed on the system.

Ex)
-> hsb_remove
"/hsb/0" deleted
"/hsb/1" deleted
"/hsb/2" deleted

"/hsb/3" deleted
hsbpmc7036 driver Removed!
value = 0 = 0x0

3－2 . HSB Device Driver Level functions.

HSB Device Driver는 VxWorks/Linux의 기본적인 Device Driver 구조를 따르고 있다, 따라서, open, write, read, ioctl의 Device Driver 인터페이스를 제공한다.

3-2-1. open

function purpose **int open (char *name, int flags, int mode)**

Used to open a HDLC-Serial Board device for read and/or write operation.
When called the previously configured port (by calling hsb_create()) is enabled for read and write operation function.

arguments

The name argument is the device file name associated with the requested serial device. The device names are as follows:

<Name >

/hsb/0	Serial Device 0 (channel 1)
/hsb/1	Serial Device 1 (channel 2)
/hsb/2	Serial Device 2 (channel 3)
/hsb/3	Serial Device 3 (channel 4)

<flags>

Read	O_RD_ONLY(0)
Write	O_WR_ONLY(1)
Read/Write	O_RDWR (2).
Mode	file permissions(ignored by the driver).

returns

<value>	The integer returned is the file descriptor for the selected device
ERROR	an error was encountered.

3－2－2．close

function **int close (int fd)**

purpose

Used to disable the specified device from read and/or write operation.

arguments

<int fd>

The fd argument is the file descriptor obtained by the open function call

returns

OK	The operation was successful
ERROR	An error was encountered.

3-2-3. read

function **int read (int fd, char *buffer, int maxbytes)**

purpose

Used to read data from the specified device.
In synchronous mode however data is received in frames ranging from 1 to 4K bytes in size.
The driver will return less than the requested number of bytes if the end of a frame is detected.
Typical synchronous operation would be to request more than the maximum frame size and use the return value to determine the actual size of the frame received.

arguments

<int fd>
The fd argument is the file descriptor obtained by the open function call.
<char * buffer>
The buffer argument is a pointer to the application read buffer.
<int maxbytes>
The maxbytes argument indicates the maximum number of bytes to be read from the device. Keep in mind that in synchronous modes the CRC is returned as part of the data and must be accounted for. In synchronous modes this value should be larger than the maximum frame size and in transparent mode it should be the specified driver buffer size

returns

<value>	The integer returned is the number of bytes read from the device
ERROR	An error was encountered

To determine the error type the following ioctl call can be used to return the status of the previous read.

3-2-4. write

Function purpose **int write (int fd, char *buffer, int numbytes)**

Used to write data to the specified device.
The data is copied from the specified buffer directly into the drivers write buffer

arguments

<int fd>
The fd argument is the file descriptor obtained by the open function call.
<char *buffer>
The buffer argument is a pointer to the application write buffer.
< int numbytes>
The numbytes argument indicates the number of bytes to be written to the device.
In synchronous modes this value should never be larger than the maximum frame size.

returns

<value>	The integer returned is the number of bytes written to the device
ERROR	An error was encountered

To determine the error type the following ioctl call can be used to return the status of the previous write.

3-2-5. ioctl

function **int ioctl (int fd, int function, int argument)**

purpose

Used to perform I/O control functions on the specified device.
The I/O functions supported by the HS-Serial module are listed below

arguments

<int fd>
The fd argument is the file descriptor obtained by the open function call
<int function>
The function argument indicates which function the application wishes to execute. The functions are defined in the table above.
< int argument>
The argument is the value to be passed to the selected function.

returns

<Value> The integer returned is the either the return value from the selected function
ERROR an error was encounter

IOCTL function	code	function
IOCTL_HSB_SYNC_MODE_SET/GET	500/600	Synchronous / Asynchronous mode Select
IOCTL_HSB_BAUD_RATE_SET/GET	501/601	Baud Rate.
IOCTL_HSB_PHYC_MODE_SET/GET	502/602	Physical Interface.
IOCTL_HSB_SYTX_AREN_SET/GET	510/610	Transmit frame address field enable.
IOCTL_HSB_SYTX_CMEN_SET/GET	511/611	Transmit frame control field enable.
IOCTL_HSB_SYTX_FNEN_SET/GET	512/612	Transmit frame FCS field enable.
IOCTL_HSB_SYTX_ADDR_SET/GET	513/613	Transmit frame address field value.
IOCTL_HSB_SYTX_COMM_SET/GET	514/614	Transmit frame control field value.
IOCTL_HSB_SYTX_ARSZ_SET/GET	515/615	Transmit frame address field size.
IOCTL_HSB_SYTX_CMSZ_SET/GET	516/616	Transmit frame control field size.
IOCTL_HSB_SYTX_FNSZ_SET/GET	517/617	Transmit frame FCS field size.
IOCTL_HSB_SYTX_CRC_SET	518	Transmit frame FCS field value.
IOCTL_HSB_SYRX_AREN_SET/GET	520/620	Receiving frame address field enable.
IOCTL_HSB_SYRX_CMEN_SET/GET	521/621	Receiving frame control field enable.
IOCTL_HSB_SYRX_FNEN_SET/GET	522/622	Receiving frame FCS field enable.
IOCTL_HSB_SYRX_ADDR_SET/GET	523/623	Receiving frame address field value.
IOCTL_HSB_SYRX_COMM_SET/GET	524/624	Receiving frame control field value.
IOCTL_HSB_SYRX_ARSZ_SET/GET	525/625	Receiving frame address field size.
IOCTL_HSB_SYRX_CMSZ_SET/GET	526/626	Receiving frame control field size.
IOCTL_HSB_SYRX_FNSZ_SET/GET	527/627	Receiving frame FCS field size.
IOCTL_HSB_SYRX_CRC_SET	528	
IOCTL_HSB_SYRX_ARFF_SET/GET	530/630	Receiving frame address field enable in FiFo
IOCTL_HSB_SYRX_CMFF_SET/GET	531/631	Receiving frame control field enable in FiFo
IOCTL_HSB_SYRX_FNFF_SET/GET	532/632	Receiving frame FCS field enable in FiFo
IOCTL_HSB_LOOP_BACK_SET/GET	550/650	Internal loopback
IOCTL_HSB_SYTX_PEEN_SET/GET	551/651	Preamble enable
IOCTL_HSB_SYTX_PESZ_SET/GET	552/652	Preamble Size
IOCTL_HSB_SYTX_PREM_SET/GET	553/653	Preamble Pattern Value
IOCTL_HSB_PAUSE_ENBL_SET/GET	554/654	Pause time enable
IOCTL_HSB_PAUSE_LENG_SET/GET	555/655	Pause time Length
IOCTL_HSB_SYRX_ARMK_SET/GET	556/656	Receiving Address Mask value
IOCTL_HSB_SYRX_ARMK_EN_SET/GET	557/657	Receiving Address Mask enable
IOCTL_HSB_SYRX_ARUQ_SET/GET	558/658	Receiving Address uni-quest enable
IOCTL_HSB_SYNC_FLAG_SET/GET	559/659	Synchronous communication flag
IOCTL_HSB_SOFT_RESET	560/	Reset
IOCTL_HSB_FIFO_CLEAR_SET/GET	561/	FiFo Clear
IOCTL_HSB_RXHW_TOUT_SET/GET	562/662	Rx Time out

IOCTL_HSB_TXHW_TOUT_SET/GET	563/663	Tx Time out
IOCTL_HSB_FRM_ENCODE_SET/GET	566/666	Tx Frame encoding type
IOCTL_HSB_DATA_BITS_SET/GET	570/670	BIT
IOCTL_HSB_PARITY_SET/GET	571/671	Parity
IOCTL_HSB_STOPBIT_SET/GET	572/672	Stop Bits
IOCTL_HSB_FLOW_CTRL_EN_SET/GET	573/673	Flow Control enable
IOCTL_HSB_FLOW_CTRL_SET/GET	574/674	Flow Control
IOCTL_HSB_FLOW_CTRL_STAT_GET	/675	Flow Control Status
IOCTL_HSB_LINE_STAT_GET	/676	Line Status
IOCTL_HSB_LINE_CTRL_GET	/677	Line Control
IOCTL_HSB_LINE_RTS_SET	578	Set Line Control RTS
IOCTL_HSB_LINE_DTR_SET	579	Set Line Control DTR
IOCTL_HSB_INTM_TX_FFEM_SET	580	Interrupt Mask (FiFo empty)
IOCTL_HSB_INTM_TX_COMP_SET	581	Interrupt Mask (TX Complete)
IOCTL_HSB_INTM_RX_FFEM_SET	582	Interrupt Mask (Rx FiFo empty)
IOCTL_HSB_INTM_RX_FFFU_SET	583	Interrupt Mask (Rx FiFo Full)
IOCTL_HSB_INTM_RX_EROR_SET	584	Interrupt Mask (Rx Error)
IOCTL_HSB_CLOCK_MODE_SET/GET	590/690	Set/Get HSB Clock Mode 0: Standard Clock, 1: External Clock, 2:User Clock
IOCTL_HSB_CLOCK_SPD_SET/GET	591/691	Set/Get HSB Oscillator Clock Speed

표 **4. IOCTL code Table**

3 – 2 – 6 . HSB Device Driver Level helper functions.

function **int hsb_resetChannel(char * dev)**

purpose

입력된 채널 디바이스의 name을 입력 받아 해당 채널을 Reset 한다.

arguments

<char * dev> - HSB device driver channel name

returns

OK - Successfully reset

ERROR - An error was encounter

Ex)
-> hsb_resetChannel("/hsb/3")

function **int hsb_devReset ()**
purpose
모든 채널 디바이스를 Reset 한다.
arguments
<char * dev> - HSB device driver channel name
returns
OK - Successfully reset
ERROR - An error was encounter

Ex)
-> hsb_devReset
"/hsb/0" deleted
"/hsb/1" deleted
"/hsb/2" deleted
"/hsb/3" deleted
hsbpmc7036 driver Removed!
"/hsb/0" created.
"/hsb/1" created.
"/hsb/2" created.
"/hsb/3" created.
HDLC Serial Board Logic Version(2)

function **void hsb_getFiFoStat(int nChanNum)**
purpose

지정한 채널번호의 현재 FiFo의 정보를 출력한다.

arguments

< channel number – int nChanNum>
채널 번호

returns

<void>

Ex)
-> hsb_getFiFoStat(2)
FiFo Status
: Rx Used = 0
: Rx Remain = 4095
: Tx Used = 0
: Tx Remain = 4095
: Rx Empty = 1
: Rx Half Full = 0
: Rx Full = 0
: Tx Empty = 1
: Tx Half Full = 0
: Tx Full = 0
value = 0 = 0x0

function **void hsb_write_print(UINT32 *uAddr, UINT32 value)**

purpose

Register의 offset과 value를 해당 register의 값으로 설정한다.
설정된 내용이 출력된다.

arguments

<UINT32 *uAddr> - Registger offset address.
<UNT32 value> - Registger value.

returns

<void>

function **void hsb_write_value(UINT32 *uAddr, UINT32 value)**

purpose

Register의 offset과 value를 해당 register의 값으로 설정한다.
설정된 내용이 출력되지 않는다.

arguments

<UINT32 * uAddr> - Registger offset address.
<UNT32 value> - Registger value.

returns

<void>

function **UINT32 hsb_read_print(UINT32 *readAddr)**

purpose

Register의 offset과 value를 해당 register의 값.
설정된 내용이 출력된다.

arguments

<UINT32 *readAddr> - Register offset address.

returns

<UNT32 value> - Register value.

Ex)
-> hsb_read_print(0x304)
ADDRESS[0xa1000304] = VALUE:0x18
value = 24 = 0x18

function **void hsb_regInfo(int nChan)**
purpose 입력된 채널 번호의 HSB의 register 값을 출력한다.

arguments
<int nChan> - 채널 번호
returns
<void>

Ex)
-> hsb_regInfo(3)
ADDRESS[0xa1000104] = VALUE:0x00
ADDRESS[0xa1000118] = VALUE:0x00
ADDRESS[0xa1000504] = VALUE:0x18
ADDRESS[0xa1000508] = VALUE:0xf3
ADDRESS[0xa100050c] = VALUE:0x12
ADDRESS[0xa1000510] = VALUE:0xea
ADDRESS[0xa1000514] = VALUE:0x02
ADDRESS[0xa1000518] = VALUE:0x01
ADDRESS[0xa100051c] = VALUE:0xfff
ADDRESS[0xa1000524] = VALUE:0x70
ADDRESS[0xa1000528] = VALUE:0xbb
ADDRESS[0xa100052c] = VALUE:0x40
ADDRESS[0xa1000530] = VALUE:0x21
ADDRESS[0xa1000534] = VALUE:0x01
ADDRESS[0xa1000538] = VALUE:0x00
ADDRESS[0xa100053c] = VALUE:0x00
ADDRESS[0xa1000540] = VALUE:0x144
ADDRESS[0xa1000544] = VALUE:0x01
ADDRESS[0xa1000548] = VALUE:0x51
ADDRESS[0xa1000550] = VALUE:0xffff
ADDRESS[0xa1000554] = VALUE:0x00
ADDRESS[0xa1000558] = VALUE:0xff
ADDRESS[0xa1000560] = VALUE:0x7e
ADDRESS[0xa1000564] = VALUE:0x00
ADDRESS[0xa100056c] = VALUE:0x00
ADDRESS[0xa1000730] = VALUE:0x00
ADDRESS[0xa1000734] = VALUE:0x00
ADDRESS[0xa1000738] = VALUE:0x00
ADDRESS[0xa100073c] = VALUE:0x00

function **void hsb_devInfo(char * dev)**
purpose

입력된 채널 디바이스의 name을 입력 받아 device의 정보를 출력한다.

arguments

<char * dev> - HSB device driver channel name

returns

<void>

Ex)

```
-> hsb_devInfo("/hsb/0")
SCC Information.....
SCC[uChannel         ]= 0
SCC[baseAddr         ]= 0xa1000000
SCC[baseCSR          ]= 0xa1000300
SCC[baseIER_Tx       ]= 0xa1000780:0x0
SCC[baseIER_Rx       ]= 0xa1000784:0x0
SCC[baseIMR          ]= 0xa1000110
SCC[baseIMR_Tx       ]= 0xa10007a0:0x0
SCC[baseIMR_Rx       ]= 0xa10007a4:0x0
SCC[uRxAsyncFrm      ]= 0xa10007c0:0x10
SCC[uSyncMode        ]= 0x1
SCC[uBaudRate        ]= 307200
SCC[uHsbClock        ]= 0x0
SCC[uHsbClockSpd ]= 0xe10000
SCC[uPhysMode        ]= 0x2
SCC[uLineCtrl        ]= 0xffffff3
SCC[uLoopback        ]= 0x0
SCC[uSyncFlag        ]= 0x0
SCC[uOpenFlag        ]= 1
SCC[uTxStatus        ]= 0x11
SCC[uRxStatus        ]= 0x11
SCC[uTxCount         ]= 0
SCC[uTxWriteCnt      ]= 0
SCC[uTxWritePtr      ]= 0
SCC[uRxCount         ]= 0
SCC[uRxReadCnt       ]= 0
SCC[uRxReadPtr       ]= 0
SCC[uTxTimeout       ]= 0
SCC[uTxHwTimeout ]= 0
SCC[uIntTxTimeout]= 0
SCC[uRxTimeout       ]= 0
SCC[uRxHwTimeout ]= 0
SCC[uIntRxTimeout]= 0
SCC[uRxCurFiFoPtr]= 0
SCC[uTxTotCnt        ]= 0
SCC[uTxFrmCnt        ]= 0
SCC[uRxTotCnt        ]= 0
SCC[uRxFrmCnt        ]= 0
```

function **void hsb_channelInfo(int nChan)**
purpose
입력된 채널 번호 입력 받아 device의 정보를 출력한다.
arguments
<int nChan> - HSB device driver channel number
returns
<void>

Ex)
-> hsb_channelInfo(3)
HDLC-Serial Board Channel(3)(/hsb/2)
MODE: SYNC | 307200
PHYSICAL MODE: RS-530
CLOCK: Statndrad(14.7456Mhz)
HDLC Synchronous MODE
HDLC FRAME(T): 7e | Address(1 byte) | Control(1 byte) | DATA | CRC(16) | 7e
HDLC FRAME(R): 7e | Address(1 byte) | Control(1 byte) | DATA | CRC(16) | 7e
Address(S): 0x1
Address(M): 0xffff, disable
Address(D): 0x2
Control : 0x1
Preemble enalbe,Length 0x8
Pause disable,Length 0x0
TX enalbe
RX enalbe
FRAME ENCODING: NRZ
Timeout(RX): WAIT_FOREVER
Timeout(TX): WAIT_FOREVER
Line Status: Value = 0x40
: Tx Overflow = 0
: Sync Tx Done = 1
: Sync Tx Empty = 0
: Rx Abort = 0
: Rx Frame Error = 0
: Rx FCS Error = 0
: Rx Overflow = 0
: Rx Rx Ready = 0
FiFo Status
: Rx Used = 0
: Rx Remain = 4095
: Tx Used = 0
: Tx Remain = 4095
: Rx Empty = 1
: Rx Half Full = 0
: Rx Full = 0
: Tx Empty = 1
: Tx Half Full = 0
: Tx Full = 0
value = 0 = 0x0

function **void hsb_drvShow()**
purpose
전체 채널(all channel)의 정보를 출력한다.
arguments
<void>
returns
<void>

Ex)
-> hsb_drvShow
PMC HS-Serial Device Status Show!
This Device Driver support Sync/ASync UART/HDLC Mode
Device Driver Version 1.0.2
Installed PMC HS-Serial Device Number 1

==========
HDLC-Serial Board Channel(1)(/hsb/0) installed...
MODE: SYNC | 307200
PHYSICAL MODE: RS-530
CLOCK: Statndrad(14.7456Mhz)
HDLC Synchronous MODE
HDLC FRAME(T): 7e | Address(1 byte) | Control(1 byte) | DATA | CRC(16) | 7e
HDLC FRAME(R): 7e | Address(1 byte) | Control(1 byte) | DATA | CRC(16) | 7e
Address(S): 0x1
Address(M): 0xffff, disable
Address(D): 0x2
Control : 0x1
Preemble enalbe,Length 0x8
Pause disable,Length 0x0
TX enalbe
RX enalbe
FRAME ENCODING: NRZ
Timeout(RX): WAIT_FOREVER
Timeout(TX): WAIT_FOREVER
…
HDLC-Serial Board Channel(4)(/hsb/3) installed...
MODE: SYNC | 307200
PHYSICAL MODE: RS-530
CLOCK: Statndrad(14.7456Mhz)
HDLC Synchronous MODE
HDLC FRAME(T): 7e | Address(1 byte) | Control(1 byte) | DATA | CRC(16) | 7e
HDLC FRAME(R): 7e | Address(1 byte) | Control(1 byte) | DATA | CRC(16) | 7e
Address(S): 0x1
Address(M): 0xffff, disable
Address(D): 0x2
Control : 0x1
Preemble enalbe,Length 0x8
Pause disable,Length 0x0
TX enalbe
RX enalbe
FRAME ENCODING: NRZ
Timeout(RX): WAIT_FOREVER
Timeout(TX): WAIT_FOREVER
value = 0 = 0x0

function **int hsb_syncSetup(char * szDevName, int nPhys, int nBaud, int Modulation)**

purpose

지정한 채널의 통신설정을 HDLC 모드로 변경한다.

arguments

<char * szDevName>	-	Channel name, 채널 이름
<int nPhys>	-	Physical Interface. 물리 인터페이스.
<int nBaud>	-	Baud Rate. 전송속도
<int Modulation>	-	Frame encoding Modulation. 부호화 모드

returns

OK
ERROR

Ex)
-> hsb_syncSetup("/hsb/2", 2, 307200, 0);

function **int hsb_asyncSetup(char * szDevName, int nPhys, int nBaud, int nData, int nPar, int nStop)**

purpose

Asynchonous 통신 모드를 설정한다.

arguments

<char * szDevName>	-	채널 이름
<int nPhys>	-	물리 계층 인터페이스
<int nBaud>	-	전송속도
< int nData>	-	데이터 크기
<int nPar>	-	Parity 설정
<int nStop>	-	stop 비트 설정

returns

OK
ERROR

function **void hsb_rxFrameSetup(char * strDevName,**
int rxAdrEn, int rxAdrLen, int rxAdrFiF, int rxAdr,
int rxCmdEn, int rxCmdLen, int rxCmdFiF,
int rxFCSEn, int rxFCSLen)

purpose

지정한 채널의 HDLC Rx Frame을 구성하기 위한 환경을 설정한다.

arguments

char * strDevName	-	채널 이름
int rxAdrEn	-	HDLC Frame address field enable/disable
int rxAdrLen	-	address length 8/16 [0 -> 8 bits, 1 -> 16 bits]
int rxAdrFiF	-	address FiFo enable/disable
int rxAdr	-	address value
int rxCmdEn	-	HDLC Frame control field enable/disable
int rxCmdLen	-	control field length 8/16 [0 -> 8 bits, 1 -> 16 bits]
int rxCmdFiF	-	control field FiFo enable/disable
int rxFCSEn	-	HDLC Frame FCS Field enable/disable
int rxFCSLen	-	FCS field length CRC16/CRC32 0 -> CRC16, 1 -> CRC32]

returns

void

Ex)
-> hsb_rxFrameSetup("/hsb/0",0x1, 0x0, 0x0, 0x01, 0x1, 0x0, 0x0, 0x1,0x0)

function **void hsb_txFrameSetup(char * strDevName,**
int txAdrEn, int txAdrLen, int txAdr,
int txCmdEn, int txCmdLen, int txCmd,
int txFCSEn , int txFCSLen)

purpose

지정한 채널의 HDLC Tx Frame을 구성하기 위한 환경을 설정한다.

arguments

char * strDevName	-	채널 이름
int txAdrEn	-	HDLC Tx Frame address field enable/disable
int txAdrLen	-	address length [0 -> 8 bits, 1 -> 16 bits]
int txAdr	-	address value
int rxCmdEn	-	HDLC Tx Frame control field enable/disable
int rxCmdLen	-	control field length 8/16 [0 -> 8 bits, 1 -> 16 bits]
int rxFCSEn	-	HDLC Frame FCS Field enable/disable
int rxFCSLen	-	FCS field length CRC16/CRC32 [0 -> CRC16, 1 -> CRC32]

returns

<void>

Ex)
-> hsb_txFrameSetup("/hsb/0",0x1, 0x0, 0x2, 0x1, 0x0, 0x1, 0x1, 0x0)

function **void hsb_syncDisplay(char * dev)**
purpose
입력된 채널 디바이스의 name을 입력 받아 hdlc 설정에 대한 정보를 출력한다.
arguments
<char * dev> - HSB device driver channel name
returns
<void>

Ex)
-> hsb_syncDisplay(1)
HDLC-Serial Board Channel(1)(/hsb/0)
MODE: SYNC | 307200 [RS-530]
HDLC FRAME(T): 7e | Address(1 byte) | Control(1 byte) | DATA | CRC(16) | 7e
HDLC FRAME(R): 7e | Address(1 byte) | Control(1 byte) | DATA | CRC(16) | 7e
Address(S): 0x1
Address(M): 0xffff, disable
Address(D): 0x2
Control : 0x1
Preemble enalbe[0x8:0x0]
Pause disable
FRAME ENCODING: NRZ
value = 20 = 0x14

4 . HDLC Serial Board(HSB) Connector Signal Assignment.

HSB 의 Serial I/O Interface는 전면 연결부(front)와 후면 연결부/ (rare)를 통해서 이루어진다.
전면 연결부는 SCSI III style connector 68핀 타입이며 후면 연결부는 PMC P4 64 핀 타입이다..

4 – 1 . HSB Front Panel Serial I/O Pin Assignment

전면 연결부의 SCSI III style connector 68핀은 다음과 같이 정의한다.

Front Panel Connector J1 Pin#	Port #	Signal Mnemonics			Front Panel Connector J1 Pin#	Port #	Signal Mnemonics		
		RS-232	RS-530	V.35			RS-232	RS-530	V.35
1	1	CD	CDA/-	CD	35	2	CD	CDA/-	CD
2			CDB/+		36			CDB/+	
3		RXD	RXDA/	RXDA/	37		RXD	RXDA/	RXDA/
4		RTS	RTSA/-	RTS	38		RTS	RTSA/-	RTS
5		TXD	TXDA/	TXDA/	39		TXD	TXDA/-	TXDA/
6		CTS	CTSA/-	CTS	40		CTS	CTSA/-	CTS
7			RTSB/+		41			RTSB/+	
8			CTSB/+		42			CTSB/+	
9		GND	GND	GND	43		GND	GND	GND
10			TXDB/+	TXDB/+	44			TXDB/+	TXDB/+
11			RXDB/+	RXDB/+	45			RXDB/+	RXDB/+
12		TXC	TXCA/	TXCA/	46		TXC	TXCA/	TXCA/
13			TXCB/+	TXCB/+	47			TXCB/+	TXCB/+
14			GND	GND	48			GND	GND
15		RXC	RXCA/	RXCA/	49		RXC	RXCA/	RXCA/-
16			RXCB/+	RXCB/+	50			RXCB/+	RXCB/+
17	3	CD	CDA/-	CD	51	4	CD	CDA/-	CD
18			CDB/+		52			CDB/+	
19		RXD	RXDA/	RXDA/	53		RXD	RXDA/	RXDA/
20		RTS	RTSA/-	RTS	54		RTS	RTSA/-	RTS
21		TXD	TXDA/	TXDA/	55		TXD	TXDA/-	TXDA/
22		CTS	CTSA/-	CTS	56		CTS	CTSA/-	CTS
23			RTSB/+		57			RTSB/+	
24			CTSB/+		58			CTSB/+	
25		GND	GND	GND	59		GND	GND	GND
26			TXDB/+	TXDB/+	60			TXDB/+	TXDB/+
27			RXDB/+	RXDB/+	61			RXDB/+	RXDB/+
28		TXC	TXCA/	TXCA/	62		TXC	TXCA/	TXCA/
29			TXCB/+	TXCB/+	63			TXCB/+	TXCB/+
30			GND	GND	64			GND	GND
31		RXC	RXCA/	RXCA/	65		RXC	RXCA/	RXCA/-
32		RXCB/+	RXCB/+		66			RXCB/+	RXCB/+
33	4	Reserved (INB/+)			67		Reserved (OUTB/+)		
34		Reserved (INA/-)			68		Reserved (OUTA/-)		

표 5 Front Panel Serial I/O Pin Assignment

4 - 2 . HSB Rare Panel Serial I/O Pin Assignment

후면 연결부의 PMC P4 64 핀의 정의는 다음과 같다.

Rare Panel Connector J1 Pin#	Port #	Signal Mnemonics			Rare Panel Connector J1 Pin#	Port #	Signal Mnemonics		
		RS-232	RS-530	V.35			RS-232	RS-530	V.35
1	1	CD	CDA/	CD	33	3	CD	CDA/-	CD
2		CDB/+			34			CDB/+	
3		RXD	RXDA/	RXDA/	35		RXD	RXDA/	RXDA/
4		RTS	RTSA/	RTS	36		RTS	RTSA/-	RTS
5		TXD	TXDA/	TXDA/	37		TXD	TXDA/	TXDA/
6		CTS	CTSA/	CTS	38		CTS	CTSA/-	CTS
7			RTSB/+		39			RTSB/+	
8			CTSB/+		40			CTSB/+	
9		GND	GND	GND	41		GND	GND	GND
10			TXDB/+	TXDB/+	42			TXDB/+	TXDB/+
11			RXDB/+	RXDB/+	43			RXDB/+	RXDB/+
12		TXC	TXCA/	TXCA/	44		TXC	TXCA/	TXCA/
13			TXCB/+	TXCB/+	45			TXCB/+	TXCB/+
14			GND	GND	46			GND	GND
15		RXC	RXCA/	RXCA/	47		RXC	RXCA/	RXCA/
16			RXCB/+	RXCB/+	48			RXCB/+	RXCB/+
17	2	CD	CDA/	CD	49	4	CD	CDA/-	CD
18			CDB/+		50			CDB/+	
19		RXD	RXDA/	RXDA/	51		RXD	RXDA/	RXDA/
20		RTS	RTSA/	RTS	52		RTS	RTSA/-	RTS
21		TXD	TXDA/	TXDA/	53		TXD	TXDA/	TXDA/
22		CTS	CTSA/	CTS	54		CTS	CTSA/-	CTS
23			RTSB/+		55			RTSB/+	
24			CTSB/+		56			CTSB/+	
25		GND	GND	GND	57		GND	GND	GND
26			TXDB/+	TXDB/+	58			TXDB/+	TXDB/+
27			RXDB/+	RXDB/+	59			RXDB/+	RXDB/+
28		TXC	TXCA/	TXCA/	60		TXC	TXCA/	TXCA/
29			TXCB/+	TXCB/+	61			TXCB/+	TXCB/+
30			GND	GND	62			GND	GND
31		RXC	RXCA/	RXCA/	63		RXC	RXCA/	RXCA/
32			RXCB/+	RXCB/+	64			RXCB/+	RXCB/+

표 6 Rear Panel Serial I/0 Pin Assignment

4－3．PMC Connector Signal Assignment

4－3－1．PMC P1 Connector Pin Assignment

Pin #	Signal Name	Signal Name	Pin #
1 (N/C)	TCK	-12V	2 (BP)
3	Ground	INTA#	4
5 (N/C)	INTB#	INTC#	6 (N/C)
7	BUSMODE1#	+5V	8
9 (N/C)	INTD#	PCI-RSVD	10 (N/C)
11	Ground	PCI-RSVD	12 (N/C)
13	CLK	Ground	14
15	Ground	GNT#	16
17	REQ#	+5V	18
19 (N/C)	V (I/O)	AD[31]	20
21	AD[28]	AD[27]	22
23	AD[25]	Ground	24
25	Ground	C/BE[3]#	26
27	AD[22]	AD[21]	28
29	AD[19]	+5V	30
31 (N/C)	V (I/O)	AD[17]	32
33	FRAME#	Ground	34
35	Ground	IRDY#	36
37	DEVSEL#	+5V	38
39	Ground	LOCK#	40 (N/C)
41 (N/C)	SDONE#	SBO#	42 (N/C)
43	PAR	Ground	44
45 (N/C)	V (I/O)	AD[15]	46
47	AD[12]	AD[11]	48
49	AD[09]	+5V	50
51	Ground	C/BE[O]#	52
53	AD[06]	AD[05]	54
55	AD[04]	Ground	56
57 (N/C)	V (I/O)	AD[03]	58
59	AD[02]	AD[01]	60
61	AD[00]	+5V	62
63	Ground	REQ64#	64 (N/C)

표 7 PMC P2 Connector PCI Pin Assignment

4-3-2. PMC P2 Connector Pin Assignment

Pin #	Signal Name	Signal Name	Pin #
1 (BP)	+12V	TRST#	2 (N/C)
3 (N/C)	TMS	TDO	4 (N/C)
5 (N/C)	TDI	Ground	6
7	Ground	PCI-RSVD	8 (N/C)
9 (N/C)	PCI-RSVD	PCI-RSVD	10 (N/C)
11 (N/C)	BUSMODE2#	+3.3V	12 (BP)
13	RST#	BUSMODE3#	14
15 (BP)	3.3V	BUSMODE4#	16
17 (N/C)	PCI-RSVD	Ground	18
19	AD[30],	AD[29]	20
21	Ground	AD[26]	22
23	AD[24]	+3.3V	24 (BP)
25	IDSEL	AD[23]	26
27 (BP)	+3.3V	AD[20]	28
29	AD[18]	Ground	30
31	AD[16]	C/BE[2]#	32
33	Ground	PMC-RSVD	34 (N/C)
35	TRDY#	+3.3V	36 (BP)
37	Ground	STOP#	38
39	PERR#	Ground	40
41 (BP)	+3.3V	SERR#	42
43	C/BE[1]#	Ground	44
45	AD[14]	AD[13]	46
47	Ground	AD[10]	48
49	AD[08]	+3.3V	50 (BP)
51	AD[07]	PMC-RSVD	52 (N/C)
53 (BP)	+3.3V	PMC-RSVD	54 (N/C)
55 (N/C)	PMC-RSVD	Ground	56
57 (N/C)	PMC-RSVD	PMC-RSVD	58 (N/C)
59	Ground	PMC-RSVD	60 (N/C)
61 (N/C)	ACK64#	+3.3V	62 (BP)
63	Ground	PMC-RSVD	64 (N/C)

표 8 PMC P2 Connector PCI Pin Assignment

4 – 4 . Front Panel Cable

4 – 4 – 1 . C68M-4XD25M

HSB의 Front Panel의 Connector와 연결되는 케이블은 GE Fanuc 에서 제공하고 있는 C68M-4XD25M와 호환성을 유지하는 케이블이어야 한다.

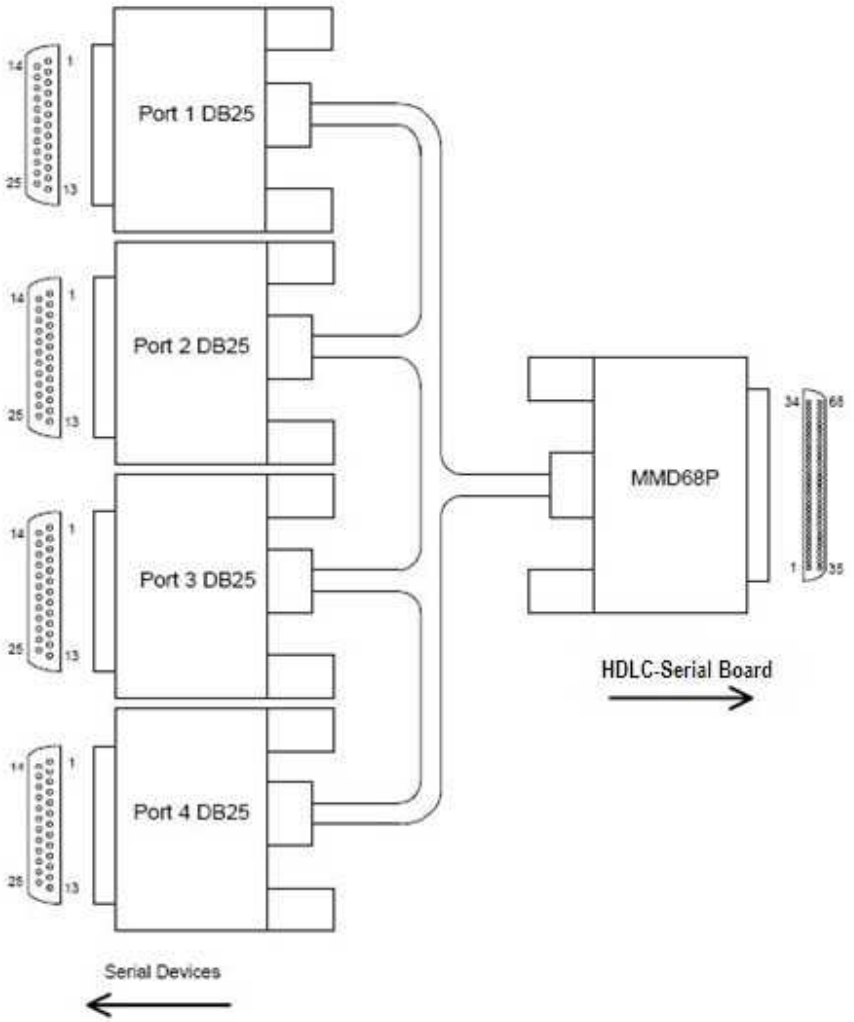

그림 2 C68M-4XD25M Cable

4-4-2. Front Cable Signal Assignment

DB25 Connector / Port #	DB25 Pin #	MMD-68 Pin #	Signal Mnemonic	DB25 Connector/ Port #	DB25 Pin #	MMD-68 Pin #	Signal Mnemonic
1	2	5	TXDA/-	2	2	39	TXDA/-
	3	3	RXDA/-		3	37	RXDA/-
	4	4	RTSA/-		4	38	RTSA/-
	5	6	CTSA/-		5	40	CTSA/-
	7	9	GND		7	43	GND
	8	1	CDA/-		8	35	CDA/-
	9	16	RXCB/+		9	50	RXCB/+
	10	2	CDB/+		10	36	CDB/+
	12	13	TXCB/+		12	47	TXCB/+
	13	8	CTSB/+		13	42	CTSB/+
	14	10	TXDB/+		14	44	TXDB/+
	15	12	TXCA/-		15	46	TXCA/-
	16	11	RXDB/+		16	45	RXDB/+
	17	15	RXCA/-		17	49	RXCA/-
	19	7	RTSB/+		19	41	RTSB/+
3	2	21	TXDA/-	4	2	55	TXDA/-
	3	19	RXDA/-		3	53	RXDA/-
	4	20	RTSA/-		4	54	RTSA/-
	5	22	CTSA/-		5	56	CTSA/-
	7	25	GND		7	59	GND
	8	17	CDA/-		8	51	CDA/-
	9	32	RXCB/+		9	66	RXCB/+
	10	18	CDB/+		10	52	CDB/+
	12	29	TXCB/+		12	63	TXCB/+
	13	24	CTSB/+		13	58	CTSB/+
	14	26	TXDB/+		14	60	TXDB/+
	15	28	TXCA/-		15	62	TXCA/-
	16	27	RXDB/+		16	61	RXDB/+
	17	31	RXCA/-		17	65	RXCA/-
	19	23	RTSB/+		19	57	RTSB/+

표 9 Cable Signal Assignment