LKP-302/LKP-302-BP 보드 설명서

LKP-302 HW Ver 1.0

LKP-302 BP v1.0 or v1.1

2012년 09월 07일

True Leader of Technology

|주|엘케이일레븐

알 림

여기에 실린 내용은 제품의 성능 향상과 신뢰도의 증대를 위하여 예고없이 변경될 수도 있습니다.

㈜엘케이일레븐
경기도 성남시 중원구 상대원동 191-1 SKn테크노파크 메가동 1306호

목 차

그림 목차

표 목차

1. 개요

LKP-302 보드는 PMC 슬롯을 통해 RS-485 4포트를 지원하는 보드입니다. LKP-302 보드의 RS-485 포트는 PMC 슬롯을 제공하는 CPU 보드의 전원과는 분리되어 있으며, 16550 UART와 호환되는 레지스터를 제공합니다.

2. 사양

2.1. 보드 규격

표 1. 보드 규격

구분	설명
RS-485	Isolated RS-485 4포트

2.2. 보드 기능

LKP-302 보드는 아래에 열거한 기능들을 지원합니다.

- RS-485

  LKP-302 보드는 전원 분리된 RS-485 4포트를 지원합니다. 최대 통신 속도 115,200bps까지 지원하며, 종단 저항을 점퍼를 이용하여 사용하지 않게 설정할 수 있습니다.

2.3. LKP-302 보드 Top PCB Silk

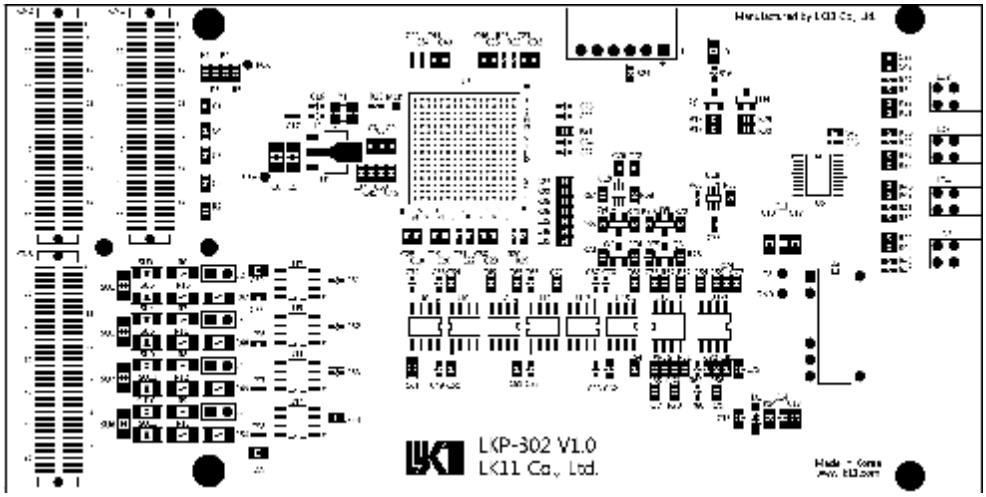

그림 1. LKP-302 보드 전면 Silk

2.4. LKP-302-BP 보드 Bottom PCB Silk

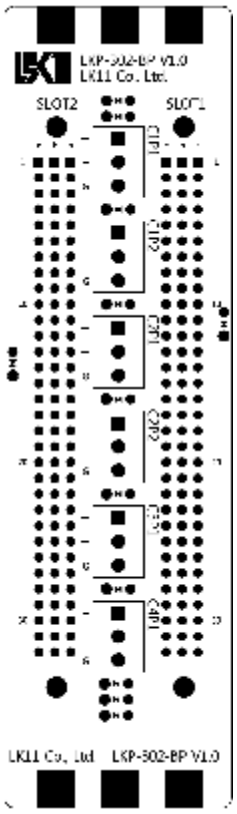

그림 2. LKP-302-BP 보드 후면 Silk

2.5. LKP-302 보드 전면판

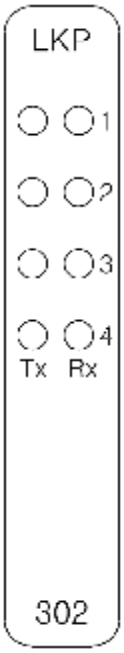

그림 3. LKP-302 보드 전면판

3. 외부 연결 방법

3.1. PMC Slot 핀 번호

표 2. 핀 번호

PMC No.	VME P2 No.	LKP-302 Pin Assignment	PMC No.	VME P2 No.	LKP-302 Pin Assignment
1	C1	GND	2	A1	**RS485_C1-**
3	C2	**RS485_C1+**	4	A2	**RS485_C1-**
5	C3	**RS485_C1+**	6	A3	GND
7	C4	Reserved	8	A4	Reserved
9	C5	Reserved	10	A5	Reserved
11	C6	GND	12	A6	**RS485_C2-**
13	C7	**RS485_C2+**	14	A7	**RS485_C2-**
15	C8	**RS485_C2+**	16	A8	GND
17	C9	Reserved	18	A9	Reserved
19	C10	Reserved	20	A10	Reserved
21	C11	GND	22	A11	**RS485_C3-**
23	C12	**RS485_C3+**	24	A12	**RS485_C3-**
25	C13	**RS485_C3+**	26	A13	GND
27	C14	Reserved	28	A14	Reserved
29	C15	Reserved	30	A15	Reserved
31	C16	GND	32	A16	**RS485_C4-**
33	C17	**RS485_C4+**	34	A17	**RS485_C4-**
35	C18	**RS485_C4+**	36	A18	GND
37	C19	Reserved	38	A19	Reserved
39	C20	Reserved	40	A20	Reserved
41	C21	GND	42	A21	GND
43	C22	Reserved	44	A22	Reserved
45	C23	Reserved	46	A23	Reserved
47	C24	Reserved	48	A24	Reserved
49	C25	Reserved	50	A25	Reserved
51	C26	Reserved	52	A26	Reserved
53	C27	Reserved	54	A27	Reserved
55	C28	Reserved	56	A28	Reserved
57	C29	Reserved	58	A29	Reserved
59	C30	Reserved	60	A30	Reserved
61	C31	Reserved	62	A31	Reserved
63	C32	Reserved	64	A32	Reserved

PMC 슬롯의 CN1, CN2는 PICMG의 PMC 규격을 만족하는 PCI Interface 용 커넥터입니다. CN3은 LKP-080B 보드의 P2와 연결되며 아래와 같은 핀 번호로 구성되어 있습니다. RS-485는 각 채널 당

(+), (-)로 구성되어 있으며, Reserved 단자는 사용하시면 안됩니다.

3.2. Jumper 설정 방법

LKP-302 보드의 J1은 FPGA 로직을 다운로드 하는 커넥터입니다. 이 커넥터는 사용자가 사용할 수 없습니다.

J2 ~ J5는 RS-485 Termination 저항의 삽입여부를 결정하는 Jumper 입니다. RS-485 포트를 여러 개를 연결할 시에 전압 강하가 심할 경우 이 Jumper를 제거하시고 사용하실 수 있습니다. 제거하는 방법은 그림 4에 있는 점퍼를 뽑으시면 됩니다.

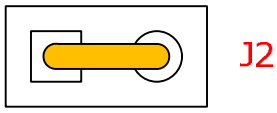

그림 4. 점퍼 설정 상태

3.3. LKP-302-BP와 연결 방법

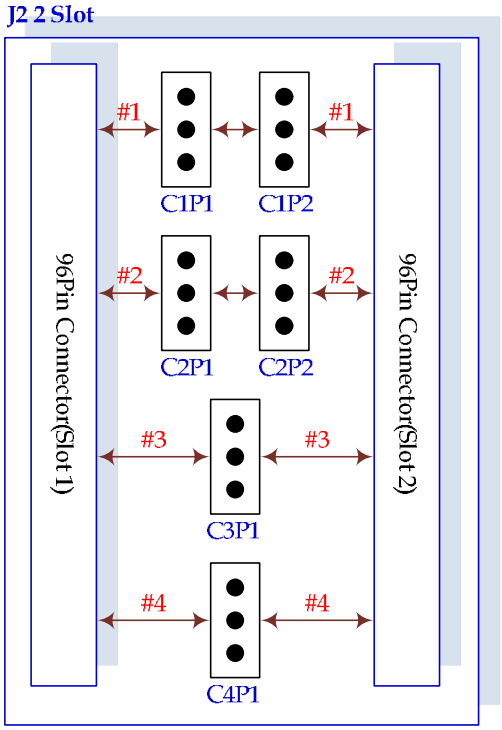

그림 5. LKP-302-BP 포트 연결 상태

LKP-302-BP 보드는 LKP-302 보드와 연결하여 사용할 수 있는 2-Slot J2 Backplane 입니다. 외부와 연결하는 터미널 블록은 Euroclamp 사의 MVS133-3.81-V 를 사용하였으면 스크류를 이용하여 단자를 고정할 수 있는 타입입니다. 커넥터 번호는 CnPm으로 되어 있으며, n은 채널 번호 m은 포트 번호입니다. 1번과 2번 채널은 서로 연결된 2개의 포트를 제공하고, 3번과 4번 채널은 서로 연결된 하나의 포트를 제공합니다.

각 터미널 블록의 1번은 RS-485 (+) 단자, 2번은 (-)단자, 3번은 GND로 3번 단자는 특별한 경우 외에는 사용하지 않습니다.

4. Memory Map

LKP-302의 메모리는 두 개의 영역으로 나누어 집니다. 이미지 0는 PCI Configuration공간이고, 이미지 1은 UART Control Register 영역입니다.

4.1. 이미지 0

이미지 0는 PCI 설정 레지스터, 주소 변환, 인터럽트 설정 등에 관련된 레지스터입니다. 이미지 0의 메모리 맵은 아래와 같습니다.

표 3. 이미지 0 메모리 맵

이름	주소	넓이	접근	설명
PCI 설정 공간	0x000 – 0x0FF			PCI 규격 Rev. 2.2 설정 공간
ICR	0x1EC	32	R/W	인터럽트 제어 레지스터
ISR	0x1F0	32	R/W	인터럽트 상태 레지스터

4.2. 이미지 1

이미지 1의 메모리 지도는 아래와 같습니다.

표 4. 이미지 1 메모리 지도

이름	주소	넓이	접근	설명
UART0_RBR	0x0000	8	R	UART 0 받기 버퍼 레지스터
UART0_THR	0x0000	8	W	UART 0 보내기 홀딩 레지스터
UART0_IER	0x0001	8	R/W	UART 0 인터럽트 가능 레지스터
UART0_IIR	0x0002	8	R	UART 0 인터럽트 구별 레지스터

UART0_FCR	0x0002	8	W	UART 0 FIFO 제어 레지스터
UART0_LCR	0x0003	8	R/W	UART 0 선 제어 레지스터
UART0_MCR	0x0004	8	R/W	UART 0 모뎀 제어 레지스터
UART0_LSR	0x0005	8	R/W	UART 0 선 상태 레지스터
UART0_MSR	0x0006	8	R/W	UART 0 모뎀 상태 레지스터
UART0_SCR	0x0007	8	R/W	UART 0 스크래치 레지스터
UART0_DLL	0x0000	8	R/W	UART 0 나누기 레지스터(LSB)
UART0_DLM	0x0001	8	R/W	UART 0 나누기 레지스터(MSB)
UART1_RBR	0x0008	8	R	UART 1 받기 버퍼 레지스터
UART1_THR	0x0008	8	W	UART 1 보내기 홀딩 레지스터
UART1_IER	0x0009	8	R/W	UART 1 인터럽트 가능 레지스터
UART1_IIR	0x000A	8	R	UART 1 인터럽트 구별 레지스터
UART1_FCR	0x000A	8	W	UART 1 FIFO 제어 레지스터
UART1_LCR	0x000B	8	R/W	UART 1 선 제어 레지스터
UART1_MCR	0x000C	8	R/W	UART 1 모뎀 제어 레지스터
UART1_LSR	0x000D	8	R/W	UART 1 선 상태 레지스터
UART1_MSR	0x000E	8	R/W	UART 1 모뎀 상태 레지스터
UART1_SCR	0x000F	8	R/W	UART 1 스크래치 레지스터
UART1_DLL	0x0008	8	R/W	UART 1 나누기 레지스터(LSB)
UART1_DLM	0x0009	8	R/W	UART 1 나누기 레지스터(MSB)
UART2_RBR	0x0010	8	R	UART 2 받기 버퍼 레지스터
UART2_THR	0x0010	8	W	UART 2 보내기 홀딩 레지스터
UART2_IER	0x0011	8	R/W	UART 2 인터럽트 가능 레지스터
UART2_IIR	0x0012	8	R	UART 2 인터럽트 구별 레지스터
UART2_FCR	0x0012	8	W	UART 2 FIFO 제어 레지스터
UART2_LCR	0x0013	8	R/W	UART 2 선 제어 레지스터
UART2_MCR	0x0014	8	R/W	UART 2 모뎀 제어 레지스터
UART2_LSR	0x0015	8	R/W	UART 2 선 상태 레지스터
UART2_MSR	0x0016	8	R/W	UART 2 모뎀 상태 레지스터
UART2_SCR	0x0017	8	R/W	UART 2 스크래치 레지스터
UART2_DLL	0x0010	8	R/W	UART 2 나누기 레지스터(LSB)
UART2_DLM	0x0011	8	R/W	UART 2 나누기 레지스터(MSB)
UART3_RBR	0x0018	8	R	UART 3 받기 버퍼 레지스터
UART3_THR	0x0018	8	W	UART 3 보내기 홀딩 레지스터
UART3_IER	0x0019	8	R/W	UART 3 인터럽트 가능 레지스터
UART3_IIR	0x001A	8	R	UART 3 인터럽트 구별 레지스터
UART3_FCR	0x001A	8	W	UART 3 FIFO 제어 레지스터
UART3_LCR	0x001B	8	R/W	UART 3 선 제어 레지스터
UART3_MCR	0x001C	8	R/W	UART 3 모뎀 제어 레지스터

UART3_LSR	0x001D	8	R/W	UART 3 선 상태 레지스터
UART3_MSR	0x001E	8	R/W	UART 3 모뎀 상태 레지스터
UART3_SCR	0x001F	8	R/W	UART 3 스크래치 레지스터
UART3_DLL	0x0018	8	R/W	UART 3 나누기 레지스터(LSB)
UART3_DLM	0x0019	8	R/W	UART 3 나누기 레지스터(MSB)
UART4_RBR	0x0020	8	R	UART 4 받기 버퍼 레지스터
UART4_THR	0x0020	8	W	UART 4 보내기 홀딩 레지스터
UART4_IER	0x0021	8	R/W	UART 4 인터럽트 가능 레지스터
UART4_IIR	0x0022	8	R	UART 4 인터럽트 구별 레지스터
UART4_FCR	0x0022	8	W	UART 4 FIFO 제어 레지스터
UART4_LCR	0x0023	8	R/W	UART 4 선 제어 레지스터
UART4_MCR	0x0024	8	R/W	UART 4 모뎀 제어 레지스터
UART4_LSR	0x0025	8	R/W	UART 4 선 상태 레지스터
UART4_MSR	0x0026	8	R/W	UART 4 모뎀 상태 레지스터
UART4_SCR	0x0027	8	R/W	UART 4 스크래치 레지스터
UART4_DLL	0x0020	8	R/W	UART 4 나누기 레지스터(LSB)
UART4_DLM	0x0021	8	R/W	UART 4 나누기 레지스터(MSB)

5. 레지스터 설명

5.1. PCI 설정 공간

PCI 설정 공간은 PCI 베이스 주소, 인터럽트 등 PCI 버스에서 요구하는 각종 설정을 저장하는 공간으로 PCI 규격 2.2를 따릅니다.

5.2. 인터럽트 제어 레지스터(ICR)

인터럽트 제어 레지스터(ICR)는 인터럽트를 허용 하거나 금지 하는 역할을 합니다.

표 5. ICR 비트 정의

Bit No.	이름	설명
0	인터럽트 전달 허용	1이면, UART의 인터럽트가 INTA#로 전달
2	PCI 에러 인터럽트 허용	1이면, 쓰기 사이클 동안 에러가 발생했을 경우 인터럽트를 발생.
3	패리티 에러 인터럽트 허용	1이면, 패리티 에러가 발견되었을 때 인터럽트를 발생.
31	소프트웨어 Reset	1이면, 소프트웨어 Reset이 발생.

5.3. 인터럽트 상태 레지스터(ISR)

인터럽트 상태 레지스터(ISR)는 인터럽트의 상태를 나타냅니다.

표 6. ISR 비트 정의

Bit No.	이름	설명
0	인터럽트	1이면, UART에서 인터럽트가 발생했음을 나타냄.
2	PCI 에러 인터럽트	1이면, 쓰기 사이클 동안 에러가 발견되어 인터럽트가 발생 했음을 나타냄.
3	패리티 에러 인터럽트	1이면, 패리티 에러에 의해 인터럽트가 발생 했음을 나타냄.

5.4. UART 받기 버퍼 레지스터(UART_RBR)

UART_RBR은 UART에서 RX를 통해 받은 데이터를 저장합니다.

5.5. UART 보내기 홀딩 레지스터(UART_THR)

UART_THR은 UART에서 TX로 보낼 데이터를 저장합니다.

5.6. UART 인터럽트 가능 레지스터(UART_IER)

UART_IER은 인터럽트를 허용하거나 금지하는 역할을 합니다. UART 인터럽트는 데이터 받음 인터럽트, 보내기 홀딩 레지스터 빔 인터럽트, 선 상태 인터럽트 및 모뎀 상태 인터럽트의 4가지가 있는데 각각의 인터럽트에 할당된 비트는 다음 표와 같습니다.

표 7. UART_IER 비트 정의

Bit No.	이름	설명
0	IER_RDI	데이터 받음 인터럽트
1	IER_THRI	보내기 홀딩 레지스터 빔 인터럽트
2	IER_RLSI	선 상태 인터럽트
3	IER_MSI	모뎀 상태 인터럽트

5.7. UART 인터럽트 구별 레지스터(UART_IIR)

UART_IIR은 인터럽트의 발생여부와 어떤 인터럽트가 발생 했는지를 알려주는 역할을 한다. 또한 FIFO가 활성화 되어 있는지를 알려줍니다.

표 8. UART_IIR 비트 정의

Bit No.	이름	설명
0	IIR_NO_INT	처리되지 않은 인터럽트가 없으면 1
1	IIR_ID0	인터럽트 ID 비트 0
2	IIR_ID1	인터럽트 ID 비트 1
3	IIR_ID2	인터럽트 ID 비트 2

인터럽트 ID는 다음 표와 같이 정의됩니다. 여러 인터럽트가 동시에 발생했을 경우 우선순위가 높은 인터럽트 ID가 읽혀집니다.

문자시간 초과 인터럽트는 4개의 문자를 받을 수 있는 시간 동안 데이터 받기 레지스터에 데이터가 있는데 읽어가지 않을 경우 발생합니다.

표 9. IIR_ID

IIR ID	이름	우선순위	설명
011	IIR_RLSI	1	선 상태 인터럽트
010	IIR_RDI	2	데이터 받음 인터럽트
110	IIR_CTOI	2	문자 시간 초과 인터럽트
001	IIR_THRI	3	보내기 홀딩 레지스터 빔 인터럽트
000	IIR_MSI	4	모뎀 상태 인터럽트

5.8. UART FIFO 제어 레지스터(UART_FCR)

UART_FCR은 FIFO를 제어하는 역할을 합니다. LKP-302 보드는 FIFO Mode를 지원하지 않습니다.

5.9. UART Line 제어 레지스터(UART_LCR)

UART_LCR은 워드 크기, 정지 비트, 패리티 등을 설정하는 역할을 합니다. 각 비트의 정의는 아래 표와 같습니다.

표 10. UART_LCR 비트 정의

Bit No.	이름	설명
0	LCR_WLEN0	워드 크기 비트 0
1	LCR_WLEN1	워드 크기 비트 1
2	LCR_STOP	정지 비트
3	LCR_PARITY	패리티
4	LCR_EPAR	짝수 패리티
5	LCR_SPAR	고정 패리티
6	LCR_SBRK	브레이크
7	LCR_DLAB	나누기 레지스터 접근

LCR_WLEN은 아래 표와 같이 설정됩니다.

표 11. LCR_WLEN

LCR_WLEN	워드 크기
00	5 비트
01	6 비트
10	7 비트
11	8 비트

LCR_STOP은 0일 경우 정지 비트가 1비트로 설정되고, 1일 경우 정지비트가 2비트로 설정됩니다.
LCR_PARITY는 1일 경우 패리티가 활성화 됩니다.
LCR_EPAR는 1일 경우 짝수 패리티가, 0일 경우 홀수 패리티가 설정됩니다.
LCR_SPAR는 1일 경우 LCR_EPAR이 1이면 패리티 비트가 무조건 0이 되고, 0이면 패리티 비트가 무조건 1이 됩니다.
LCR_BREAK는 1일 경우 TX로 무조건 0이 나갑니다.
LCR_DLAB는 1이면 UART_DLL, UART_DLM 레지스터가 선택되고, 0이면 UART_RBR, UART_THR, UART_IER 레지스터가 선택됩니다.

5.10. UART 모뎀 제어 레지스터(UART_MCR)

UART_MCR은 모뎀 제어 신호를 제어하는 역할을 합니다. 비트 정의는 다음과 같습니다.

표 12. UART_MCR 비트 정의

Bit No.	이름	설명
1	MCR_RTS	1이면 RTS 신호에 0, 0이면 RTS 신호에 1이 나감.

5.11. UART 선 상태 레지스터(UART_LSR)

UART_LSR은 데이터 전송의 상태를 나타냅니다. 비트 정의는 다음 표와 같습니다.

표 13. UART_LSR 비트 정의

Bit No.	이름	설명
0	LSR_DR	데이터가 준비됨
1	LSR_OE	데이터가 RBR에 겹쳐 써짐
2	LSR_PE	패리티 에리
3	LSR_FE	프레임 에러
4	LSR_BI	브레이크 인터럽트
5	LSR_THRE	보내기 홀딩 레지스터 빔
6	LSR_TEMT	보내기 시프트 레지스터 까지 빔
7	LSR_FIFO_ERR	FIFO안에 에러가 있음

5.12. UART 모뎀 상태 레지스터(UART_MSR)

UART_MSR은 LKP-302 보드는 RS-485 전용 보드이므로 사용되지 않습니다.

5.13. UART 스크래치 레지스터(UART_SCR)

UART_SCR은 쓰기 접근할 때 써지지 않고, 읽을 때는 항상 0값이 읽혀집니다.

5.14. UART 나누기 레지스터 LSB(UART_DLL)

UART_DLL은 UART 나누기 레지스터의 LSB 값을 저장합니다. 접근 할 때는 FCR_DLAB를 1로 설정하고 접근 해야 0합니다. 나누기 레지스터의 값은 다음과 같은 공식으로 구할 수 있습니다.

DIV = FREQ / (BAUD * 16)

이 때 FREQ 값은 18,432,000입니다.

5.15. UART 나누기 레지스터 MSB(UART_DLM)

UART_DLM은 UART 나누기 레지스터의 MSB 값을 저장합니다. 접근 할 때는 FCR_DLAB를 1로 설정하고 접근 해야 합니다.

6. 디바이스 드라이버

6.1. 디바이스 초기화 함수

LKP-302 보드의 초기화 함수는 다음과 같다.

- **STATUS sweeperInit (void)**
- Return Value : OK – 초기화 성공
  ERROR – 초기화 실패

(주의) LKP-302를 사용하기 위해서는 응용프로그램에서는 **sweeperInit()** 함수를 반드시 1회만 호출한다. 여러 번 호출하게 되면, 동작 중에 예상치 못한 에러를 발생시킨다.

6.2. 디바이스 인식 확인 함수

LKP-302 보드가 PMC 슬롯을 통해서 PCI 디바이스로 정확하게 인식되었는지를 확인하는 함수는 다음과 같다.

- **STATUS sweeperCheck (void)**
- Return Value : OK –성공적으로 인식되었음
  ERROR – 인식 실패

6.3. 시리얼 통신 관련 함수

VxWorks에서 제공하는 표준 API를 사용해서 사용자가 프로그램밍을 한다.

6.4. 예제 프로그램

드라이버 소스 코드를 컴파일하고 사용하는 방법은 아래와 같다.

NOTE) 아래의 예제는 모두 LKV-080A 보드에 LKP-302 PMC 보드를 장착하고 테스트하는 경우에 대한 것이다.

6.3.1. 드라이버 소스 컴파일

① 드라이버 소스 디렉토리로 이동
cd LKP_302_1.0.0

② Makefile에서 CPU= 부분을 사용하는 CPU 보드의 CPU에 맞게 수정한다.
CPU=PPC603

③ 소스를 컴파일해서 lkp_302.o 파일을 생성한다.
make

④ 생성된 lkp_302.o 파일을 CPU 보드가 FTP로 접속하는 디렉토리에 복사한 후에, CPU 보드의 콘솔에서 드라이버를 로드한다.
-> ld < lkp_302.o

⑤ sweeperInit을 실행한다.
-> sweeperInit
value = 0 = 0x0

6.3.2. 드라이버 확인

CPU 보드에서 sweeperInit을 성공적으로 호출한 경우에, VxWorks에서 제공하는 "devs"명령을 사용해서 사용 가능한 tty 드라이버에 LKP-302 보드의 드라이버도 사용 가능한지 확인할 수 있다.

아래는 LKP-080A 보드에 장착된 LKP-302 보드를 위한 예이다.

```
-> devs
drv name
  0 /null
  1 /tyCo/0
  1 /tyCo/1
  1 /tyCo/2
  1 /tyCo/3
  5 host:
  8 /vio
  1 /tyCo/4
  1 /tyCo/5
  1 /tyCo/6
  1 /tyCo/7
value = 25 = 0x19
```

'/tyCo/0'에서 '/tyCo/3' 까지는 보드에 LKV-080A에 내장되어 있는 시리얼 포트이고, '/tyCo/4'에서 '/tyCo/7'까지는 LKP-302의 시리얼 포트이다.

6.3.3. write() 함수 예제

UART로 값을 보낼 때는 open 함수로 디바이스를 열고, write 함수로 쓰면 된다. 다음은 LKP-302의 첫 번째 포트에 "Hello, World!"를 출력하는 예제이다.

```
int
uart_example1 (void)
```

```
{
        int fd = open("/tyCo/4", O_RDWR, 0);
        if (fd   < 0)
        {
                printf("Error opening /tyCo/4 \ n");
                return -1;
        }
        write(fd, "Hello World! \ n", 13);
        close(fd);
        return 0;
}
```

6.3.4. read() 함수 예제

UART에서 받을 때는 open 함수로 디바이스를 열고, read 함수로 읽으면 된다. 다음은 LKP-302의 두 번째 포트에서 값을 읽어오는 예제이다.

```
int
uart_example2 (void)
{
        char c;
        int fd = open("/tyCo/5", O_RDWR, 0);
        if (fd   < 0)
        {
                printf("Error opening /tyCo/5 \ n");
                return -1;
        }
        FOREVER {
                read(fd, &c, 1);
                putchar(c);
        }
        close(fd);
        return 0;
}
```

6.3.5. ioctl() 함수

tty 디바이스는 ioctl 함수를 통해 다양한 기능이 가능하다. tty가 지원하는 기능은 다음과 같다.

표 14. ioctl 기능

FIOBAUDRATE	Baud rate 설정

FIOCANCEL	Read/Write 동작 취소
FIOFLUSH	입력과 출력 버퍼의 모든 값을 버림
FIOGETNAME	fd의 이름을 얻음
FIOGETOPTIONS	옵션을 읽음
FIONREAD	입력 버퍼에서 읽지 않은 바이트 수를 구함
FIONWRITE	출력 버퍼에서 출력되지 않은 바이트 수를 구함
FIOSETOPTIONS	옵션을 설정.

다음은 LKP-302 보드의 세 번째 포트의 Baud rate를 115200으로 변경하고, "Hello World!"를 출력하는 예제이다.

```
int
uart_ example3(void)
{
        int fd = open("/tyCo/6", O_RDWR, 0);
        if (fd   < 0)
        {
                printf("Error opening /tyCo/6 \ n");
                return -1;
        }
        ioctl(fd, FIOBAUDRATE, 115200);
        write(fd, "Hello World! \ n", 13);
        close(fd);
        return 0;
}
```

옵션 값은 다음과 같다.

표 15. ioctl 옵션

OPT_ECHO	입력 에코
OPT_CRMOD	lf를 crlf로 변경
OPT_TANDEM	^S/^Q 플로우 컨트롤 프로토콜
OPT_7_BIT	입력에서 8번째 비트를 없앰
OPT_MON_TRAP	^X 활성화
OPT_ABORT	^C 활성화
OPT_TERMINAL	위의 모든 옵션 셋
OPT_RAW	위의 모든 옵션 셋 안 함

다음은 LKP-302 보드의 옵션을 읽어오는 예제이다.

```
int
uart_ example4(void)
{
        int opt;
        int fd = open("/tyCo/7", O_RDWR, 0);
        if (fd < 0)
        {
                printf("Error opening /tyCo/7. \ n");
                return -1;
        }
        opt = ioctl(fd, FIOGETOPTIONS, 0);
        if (opt & OPT_ECHO) printf("OPT_ECHO \ n");
        if (opt & OPT_CRMOD) printf("OPT_CRMOD \ n");
        if (opt & OPT_TANDEM) printf("OPT_TANDEM \ n");
        if (opt & OPT_7_BIT) printf("OPT_7_BIT \ n");
        if (opt & OPT_MON_TRAP) printf("OPT_MON_TRAP \ n");
        if (opt & OPT_ABORT) printf("OPT_ABORT \ n");
        if (opt & OPT_LINE) printf("OPT_LINE \ n");
        if (opt == 0) printf("OPT_RAW \ n");
        close(fd);
        return 0;
}
```

6.5. 보트 테스트

① 백플레인 J2에 LKP-302용 터미널 보드를 장착한다.
② CPU 보드의 PMC slot에 LKP-302 보드를 장착한다.
③ CPU 보드에 적합한 LKP-302용 드라이버 이미지를 컴파일하고, 로드할 수 있는 환경을 구성한다.
④ CPU 보드가 부팅한 후에, LKP-302용 드라이버를 load 한다.
-> ld < lkp_302.o
⑤ 드라이버와 함께 제공된 sw_test.c를 드라이버와 같이 컴파일했다면, lkp302_Test() 함수를 호출해서 테스트를 수행하고, 그렇지 않다면 사용자가 작성한 테스트 프로그램으로 테스트를 수행한다. lkp302_Test() 함수는 인자로 테스트 횟수를 입력받아 테스트를 수행하는 것에 주의한다.
⑥ sw_test.c의 시작함수에서 LKP-302용 드라이버 초기화 함수인 sweeperInit()를 호출하지 않았다면, ⑤의 과정 앞에서 콘솔상에서 sweeperInit()을 입력해서 반드시 드라이버를 초기화해야 한다.

7. 주의 사항

★ LKP-302 보드와 LKP-302-BP 보드는 CPU 보드와 전원 분리된 구조로 되어 있습니다. 어떠한 LKP-302 보드만 사용하시기 위해서는 사용하시는 CPU 보드의 PMC 슬롯 관계를 파악하셔야 합니다. 표 2는 LKV-080B 보드와의 연결을 나타낸 것이며, 타 CPU는 별도의 연결 방법이 있을 수 있습니다.

★ Reserved 되어 있는 핀은 LKP-302 보드가 내부적으로 사용할 수 있기 때문에 사용자가 절대로 사용하시면 안됩니다.

★ LKP-302-BP 보드는 300V Surge Absorber가 삽입되어 있습니다. 그 이상의 Surge가 인가되지 않도록 주의하시기 바랍니다.