

# LKV-402

## SW 사용자 메뉴얼

Board Rev. 2.4  
BSP Version 2.2.2

2009년 03월 05일

True Leader of Technology



주엘케이일레븐

## 알림

---

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

여기에 실린 내용의 일부라도 엘케이일레브의 사전 허락없이 어떠한 유형의 매체에 복사되거나 저장될 수 없으며 전기적, 기계적, 광학적, 화학적인 어떤 방법으로도 전송될 수 없습니다.

㈜엘케이일레브

경기도 성남시 중원구 상대원동 191-1 SKn테크노파크 메가동 1306호

LKV-402은 ㈜엘케이일레브의 등록상표입니다.

# 차 례

---

알 림 .....	2
1. INTRODUCTION.....	6
1.1. LKV-402 보드 소개 .....	6
1.2. 본 문서의 내용 .....	6
2. BOOT PARAMETER .....	7
2.1. Boot Parameter 정보.....	7
2.2. Boot Parameter 입력 예 .....	8
3. LKV-402 보드 설정 방법 .....	10
3.1. LKV-402 보드 Address Map .....	10
3.1.1. Flash Memory .....	11
3.1.1.1. Flash Memory 기능 .....	11
3.1.1.2. Flash Memory Control Function .....	11
3.1.1.3. makeFlashBoot Test.....	12
3.1.2. NVRAM Memory .....	13
3.1.2.1. NVRAM의 기능 .....	13
3.1.2.2. NVRAM Register Map .....	13
3.1.2.3. NVRAM Control Function .....	14
3.1.3. SRAM Memory .....	15
3.1.3.1. SRAM의 기능 .....	15
3.1.3.2. SRAM Test Example .....	15
3.1.4. Serial EEPROM (93c46).....	15
3.2. LKV-402 보드 I/O 제어.....	16
3.2.1. WatchDog time out Reset Control.....	16
3.2.1.1. Watchdog time Reset 기능.....	16
3.2.1.2. WatchDog time control Function.....	16
3.2.1.3. ....	16
3.2.2. Digital IN/OUT 제어 .....	17

3.2.2.1. Function Description .....	17
3.2.2.2. Digital IN/OUT Test Example .....	18
3.2.3. Dip Switch 제어 .....	19
3.2.3.1. Function Description .....	19
3.2.3.2. Dip Switch Test Example .....	19
<b>4. LKV-402 보드 시리얼 제어 방법 .....</b>	<b>21</b>
<b>4.1. LKV-402 보드 시리얼 드라이버 초기화 .....</b>	<b>21</b>
<b>4.2. VxWorks Serial I/O 시스템 .....</b>	<b>22</b>
4.2.1. open () .....	22
4.2.2. close () .....	22
4.2.3. read () .....	23
4.2.4. write () .....	23
4.2.5. ioctl () .....	24
<b>4.3. Serial I/O 제어 예제 .....</b>	<b>25</b>
4.3.1. Example A .....	25
4.3.1.1. open () .....	25
4.3.1.2. close () .....	25
4.3.2. Example B .....	26
4.3.2.1. read () / write () .....	26
<b>5. 주의 사항 .....</b>	<b>29</b>

## 그림 목차

그림 1. Boot Parameter 정보 예제.....	7
그림 2. Boot Parameter 확인 예제.....	8
그림 3. 네트워크의 FTP 를 이용한 부팅 예제 .....	9
그림 4. LKV-402 B/D Serial Driver 초기화 순서 .....	21

## 표 목차

표 1. LKV-402 보드 Address Map.....	10
표 2. DS1647(RTC) Register Map.....	13

## 1. Introduction

### 1.1. LKV-402 보드 소개

LKV-402 보드는 Freescale사의 MPC860 CPU를 사용하여 제작된 VMEbus System Controller 보드로서 여러 종류의 Application을 적용하여 사용 가능한 상용 보드 컴퓨터이다.

VMEbus 상에서 Master나 Slave로 동작할 수 있으며, 16bit의 데이터와 short 및 extended mode의 address로 VMEbus 로 접근할 수 있다. 현재 LKV-402 보드는 Ethernet 2 Port, HDLC 1 Port, UART 6 Port를 내장하고 있으며 사용자에게 따라 선택 가능하게 되어 있다. 또한 Wind River사의 Real Time OS인 VxWorks가 Porting 되어 사용자는 VxWorks 상에서 사용자가 직접 제작한 Application Program을 사용할 수 있다.

### 1.2. 본 문서의 내용

LKV-402 보드는 보드 내에 32MByte의 SDRAM, 4MB의 Flash Memory, Battery Backup된 2MB의 SRAM, 128KB의 RTC/NVRAM, 1Kbyte의 EEPROM이 내장되어 있다. 또한 Serial Interface와 VMEbus Interface가 구현되어 있어 Master/Slave 보드로 사용 가능합니다. 따라서 본 문서에서는 이러한 보드의 구성에 의해 1장에서는 LKV-402 보드의 소개 및 본 문서의 구성에 대한 내용이 수록 되어있으며, 2장에서는 보드 사용시 처음에 셋팅 되어야 할 Boot Parameter에 관해 자세한 설명이 되어 있다. 3장에서는 LKV-402 보드에서 지원되는 다양한 메모리 종류에 따른 컨트롤 방법 및 예가 주어지며, 그 밖의 기능의 IO(Digital I/O, DipSW, WatchDog) 컨트롤 방법에 대해 설명한다. 4장에서는 Serial Port에 관한 사용법 및 셋팅 방법에 대해 소개하며, 그에 따른 예가 주어진다.

## 2. Boot Parameter

### 2.1. Boot Parameter 정보

boot device	:	cpm
unit number	:	0
processor number	:	0
host name	:	hjahn
file name	:	E:Wftp_rootWworkWvxworksWLKV-402Wver_3.0WvxWorks
inet on ethernet (e)	:	220.76.45.97
host inet (h)	:	220.76.45.36
user (u)	:	hjahn
ftp password (pw)	:	1234
flags (f)	:	0x0
other (o)	:	e2h0u4:220.76.45.98

그림 1. Boot Parameter 정보 예제

위의 내용은 Boot Parameter VxWorks BSP를 이용하여 부팅하기 위한 정보들이다. 여기서 Boot device는 Ethernet 드라이버의 name을 가리키며 보드나 드라이버에 따라서 달라질 수 있다. 아래의 그림 2는 VxWorks BSP를 이용한 부팅 화면이다. BSP는 FTP를 이용하여 host로부터 OS+Application image를 다운로드 받으며, 후에 이를 실행한다.

LKV-402 보드는 다른 상용보드와는 달리 각 포트의 모드를 설정해 주어야 하며 이 설정 모드에 따라 각 포트의 기능이 정해진다.

- Ethernet, HDLC, UART의 Port를 선택하여 사용 가능함.
- Mode 설정: Boot Parameter의 other에 설정한다.

- 1) e0h0u6 : Ethernet 0, HDLC 0, UART 6
- 2) e0h1u5 : Ethernet 0, HDLC 1, UART 5
- 3) e1h0u5 : Ethernet 1, HDLC 0, UART 5
- 4) e1h1u4 : Ethernet 1, HDLC 1, UART 4
- 5) e2h0u4 : Ethernet 2, HDLC 0, UART 4
- 6) e2h1u3 : Ethernet 2, HDLC 1, UART 3

- Default는 e1h0u5 Mode로 셋팅
- Ethernet을 두 개의 Port로 사용시 other에 e2xxxx : ??? . ??? . ??? . ??? 으로 설정 가능함  
Ex) other(o) : e2h0u4:220.76.45.98
- Ethernet Port 2를 이용하여 통신은 가능하며 Booting 정보는 항상 Ethernet Port 1을 통해 가능하다.
- Ethernet Port 2를 사용할 경우에는 보드의 J12, J13, J19, J22 점퍼를 변경한다.

자세한 내용은 하드웨어 사용자 매뉴얼 점퍼 설정에 대한 내용을 참조 하십시오.

## 2.2. Boot Parameter 입력 예

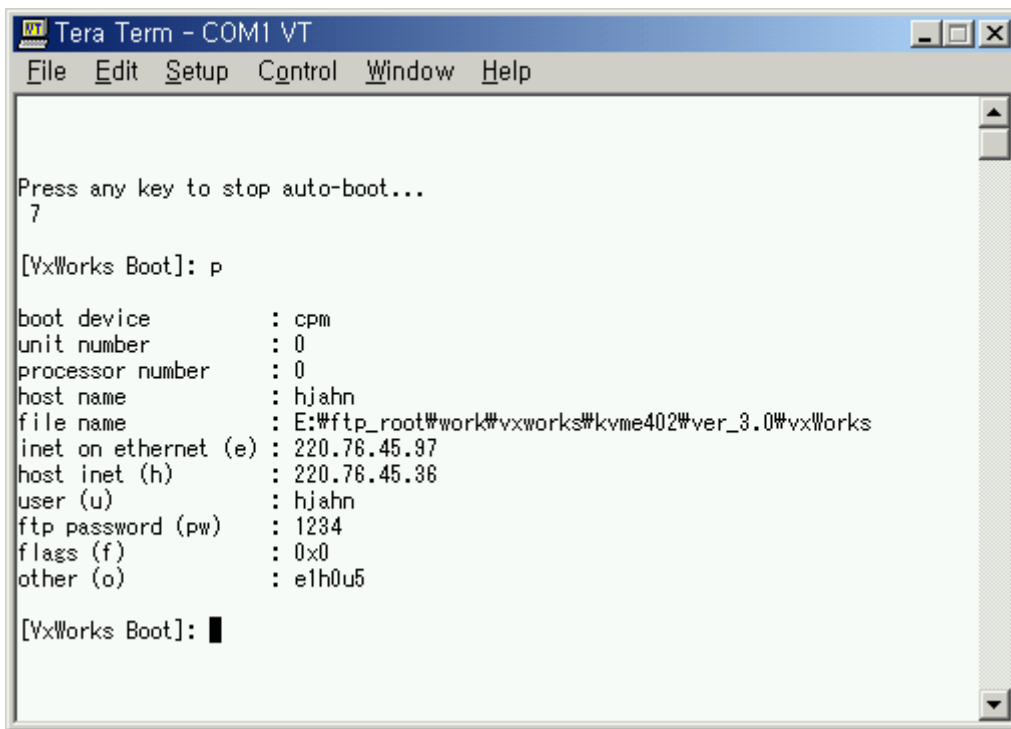


그림 2. Boot Parameter 확인 예제

VxWorks Boot 프롬프트([VxWorks Boot]:) 상에서 'p'를 입력하면 그림 2와 같이 Boot Parameter에 관한 정보를 볼 수 있다. 프롬프트 상에서 'c'를 입력하면 Boot Parameter 에 대한 정보를 user에 맞게 설정할 수 있도록 되어 있다. 다음 그림 3은 실제 VxWorks image가 boot된 후에 target Shell이 실행된 모습이다. Parameter의 입력이 끝난 후 프롬프트 상에서 '@'를 입력하면 Ethernet을 통해 부팅이 실행되는 것을 확인할 수 있다.



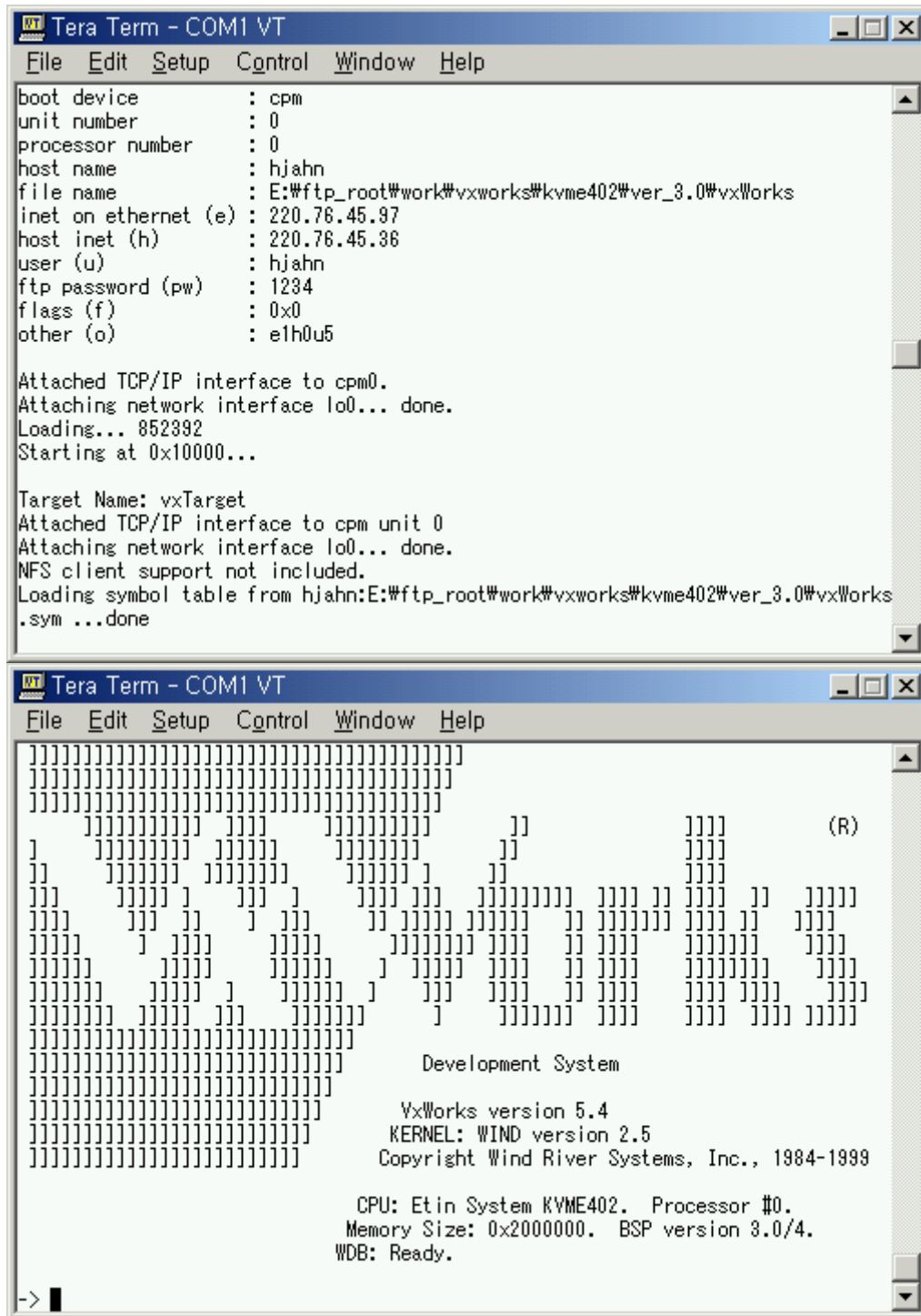


그림 3. 네트워크의 FTP를 이용한 부팅 예제

### 3. LKV-402 보드 설정 방법

#### 3.1. LKV-402 보드 Address Map

표 1. LKV-402 보드 Address Map

Address Range	Size	Description
0x00000000 ~ 0x01FFFFFF	32 Mbytes	SDRAM
0x10000000 ~ 0xEFFFFFFF	3.6 Gbytes	VMEbus Extended Address
0xF0000000 ~ 0xF0FFFFFF	16 Mbytes	VMEbus Standard Address
0xF1000000 ~ 0xF11FFFFF	2 Mbytes	SRAM
0xF2000000 ~ 0xF201FFFF	128 Kbytes	NVRAM (128 Kbyte 타입 사용시)
0xF2000000 ~ 0xF207FFFF	512 Kbytes	NVRAM (512 Kbyte 타입 사용시)
0xF3000000	1 Byte	RESERVED
0xF4000000 ~ 0xF400FFFF	64 Kbytes	VMEbus Short Address
0xF5000000	1 Byte	FAIL LED OFF
0xF6000000	1 Byte	Watch dog timer Enable
0xF7000000	1 Byte	FAIL LED ON
0xF800000X	1 Byte	VMEbus Interrupt Acknowledge
0xF9000000	1 Byte	DIP Switch & DI Read
0xFA000000 ~ 0xFA3FFFFFFF	4 Mbytes	Flash Memory
0xFB000000	1 Byte	VMEbus Interrupt Vector register
0xFC000000	1 Byte	VMEbus Interrupt Request register
0xFD000000	1 Byte	VMEbus AM Code 2 generate register
0xFE000000	1 Byte	RESERVED
0xFF000000	1 Byte	IMMR
0xFFF00000 ~ 0xFFFFFFFF	1 Mbytes	EPROM

### 3.1.1. Flash Memory

#### 3.1.1.1. Flash Memory 기능

LKV-402 보드의 Flash Memory는 총 4Mbyte의 용량을 가지고 있으며, 각 블록은 128Kbyte의 용량을 가지며 32블록으로 구성되어 있다. 또한 LKV-402 보드는 Flash Memory를 이용하여 Flash Boot 기능으로 사용할 수도 있다.

#### 3.1.1.2. Flash Memory Control Function

##### 1) makeFlashBoot();

Flash에 bootrom.bin 파일을 플래쉬 메모리에 다운로드하는 함수이다.

LKV-402 보드의 Flash Booting을 하기 위해서는 아래와 같은 절차를 수행해야 한다.

순서 1) DOS 프롬프트 상에서 "WWINDBASEW\hostWx86-win32WbinWtorVars.bat"를 실행하여 Path를 잡아준다.

순서 2) VxWorks bootrom.hex 파일을 Binary 형식으로 만든다. DOS 프롬프트 상에서 BSP 파일이 있는 디렉토리에서 다음을 실행한다.

```
Ex) make bootrom.hex
```

순서 3) 아래와 같은 명령으로 bootrom.bin 파일을 생성한다.

```
Ex) elfToBin <inFile> outfile_bsd  
    elfToBin <bootrom> bootrom.bin
```

실행시 aout file이 Binary 파일 형식으로 바뀐. 도스 명령의 가운데 인자는 롬파일 만들 때 생성된 bootrom 파일이다.

순서 4) 생성된 Binary 파일을 Target Board에서 Load 할 수 있는 위치로 옮긴다.

순서 5) Console 상에서 "ls" 명령시 다음과 같이 binary 파일이 보이는지 확인한다

```
-> ls  
bootrom.bin
```

순서 6) makeFlashBoot 함수를 이용하여 Binary 파일을 Flash Memory에 기록한다.

순서 7) 아래의 makeFlashBoot Test와 같은 메시지가 나오면 정상적으로 플래쉬 메모리에 기록된다. 아래 상태에서 롬의 우측 하단부에 있는 점퍼 2개를 밑으로 옮기면 플래쉬 부팅이 이루어진다.

### 3.1.1.3. makeFlashBoot Test

```
-> makeFlashBoot("bootrom.bin")
  Start binary file to Flash
  Load File Name : bootrom.bin - size : 271508 byte
  Open & read binary file OK
  Bin File size OK
  Flash Manufacture Code    = 0x90
  Flash Device Code        = 0x14
  Flash Block Erase OK

  271508 Byte Write. addr : 0xfa342592
  Flash Write OK.
  Start Addr = 0xfa300100, End Addr = 0xfa342593,
  Total Size = 271508 Byte
  Flash Buf Write OK!!!
  Data Varify OK!!
```

### 3.1.2. NVRAM Memory

#### 3.1.2.1. NVRAM의 기능

NVRAM은 Boot Parameter의 저장을 주 목적으로 사용하며, User가 Consol의 Buad Rate를 조절할 수 있게 한 영역으로서 Consol Port 지원하고, Real Time Clock을 지원하는 Timer 영역을 제공하고 있다.

LKV-402 보드에서는 Dallas 사의 DS1647을 사용하며, 메모리는 512 Kbyte, access time은 150ns 이하의 것을 사용한다.

#### 3.1.2.2. NVRAM Register Map

NVRAM 의 Register에 시간정보를 설정할 수 있으며, 시간 정보는 표 2와 같다.

표 2. DS1647(RTC) Register Map

ADDRESS	DATA								FUNCTION	
	B7	B6	B5	B4	B3	B2	B1	B0		
7FFFF	-	-	-	-	-	-	-	-	YEAR	00-99
7FFFE	0	0	0	-	-	-	-	-	MONTH	01-12
7FFFD	0	0	-	-	-	-	-	-	DATE	01-31
7FFFC	0	FT	0	0	0	-	-	-	DAY	01-07
7FFFB	KS	0	-	-	-	-	-	-	HOUR	00-23
7FFFA	0	-	-	-	-	-	-	-	MINUTES	00-59
7FFF9	ST	-	-	-	-	-	-	-	SECONDS	00-59
7FFF8	W	R	S	-	-	-	-	-	CONTROL	A

**ST** : Stop Bit      **R** : Read Bit      **FT** : Frequency Test  
**W** : Write Bit      **S** : Sign Bit      **KS** : Kick Start

### 3.1.2.3. NVRAM Control Function

#### 1) STATUS timeSet (char \*p)

RTC의 시간 정보를 설정하는 함수이다. 설정값은 순서는 MM DD YY HH MM의 순서로 설정된다.

Ex ) timeSet("0721030555")

MM : Month

DD : Day

YY : Year

HH : Hour

MM : Minutes

#### 2) void timeShow (void)

현재 NVRAM에 저장되어 있는 시간 정보를 출력해 주는 함수이다.

```
-> timeShow
```

```
Mon Jul 21 5:55:3 2003
```

#### 3) STATUS timeStop (void)

NVRAM(DS1647)은 real time clock기능을 정지시키는 함수이다.

#### 4) STATUS timeStart (void)

정지되어 있는 NVRAM의 real time clock 기능을 다시 ON 시키는 함수이다.

### 3.1.3. SRAM Memory

#### 3.1.3.1. SRAM의 기능

2 Mbyte의 저전력 SRAM이 장착되어 있으며, SRAM은 MPC860 또는 VMEbus에서 access 가능하다. SRAM의 access time은 70 ns이다. 또한 SRAM에는 3V의 Lithium battery가 연결되어 있어서 전원이 off 된 상태에서도 데이터를 유지할 수 있다.

#### 3.1.3.2. SRAM Test Example

SRAM 영역에 테스트 데이터를 기록하며, SRAM에서 데이터를 Read하여 테스트 데이터와 비교하여 이상유무를 확인한다.

```
-> testSRAM
SRAM Test Data Write OK!
SRAM Test OK!!!!
value = 18 = 0x12
```

#### 3.1.4. Serial EEPROM (93c46)

Ethernet의 MAC Address가 저장되는 곳으로 93c46에서 NVRAM으로 복사된다.

## 3.2. LKV-402 보드 I/O 제어

LKV-402 보드에서 제공하는 I/O들에 대한 제어 방법은 아래와 같다.

### 3.2.1. WatchDog time out Reset Control

#### 3.2.1.1. Watchdog time Reset 기능

LKV-402 보드에서는 Maxim사의 MAX690A라는 칩을 사용하여 WatchDog time out 기능을 구현하였다. 보드 설정은 J20의 shunt를 연결하면 reset 기능은 enable 상태로 되며, WatchDog reset time은 1.6초이다.

#### 3.2.1.2. WatchDog time control Function

- 1) void watchDogTimer\_Enable(void)  
WatchDog timer Enable Function으로서 이 함수를 Call 하면 WatchDog Timer 가 On 된다.
- 2) void watchDogTimer\_Disable(void)  
WatchDog timer Disable Function으로서 이 함수를 Call하면 WatchDog Timer가 Off된다.
- 3) STATUS watchDogTimer\_Clear\_Set(void)  
WatchDog timer Clear 시켜주는 Function. 이 함수를 Call하면 WatchDog Timer가 Clear되며, WatchDog Reset(H/W)이 걸리지 않게 된다. 실행시 Hold Led가 On/Off되며 WatchDog Timer 를 Clear하게 된다.
- 4) STATUS watchDogTimer\_Clear\_Free(void)  
Enable된 watchDogTimer\_Clear\_Set function을 멈추게 하는 Function이다. 함수 실행시 만일 WatchDog Timer가 Enable된 상태라면 약 1.6초 후에 Board는 Reset 된다.

#### 3.2.1.3. WatchDog time out Test Example

```
extern void watchDogTimer_Enable();
extern void watchDogTimer_Clear_Set();

void watchDogTest()
{
    sysClkRateSet(1000);
}
```



```

watchDogTimer_Enable();
watchDogTimer_Clear_Set();

while(1){
    memShow(1);
    break;
}
}

```

### 3.2.2. Digital IN/OUT 제어

#### 3.2.2.1. Function Description

LKV-402 보드에서는 4개의 Digital Input, 4개의 Digital Output Channel이 존재한다. 이 Channel을 Control 하기 위하여 다음과 같은 Function을 사용하면 된다.

```

#define DI_ON      1
#define DI_OFF     0
#define DO_ON      1
#define DO_OFF     0

```

1) char DigitInChan(int chan)

Digital Input Channel의 값을 1개 Channel 단위로 Return 시켜주는 Function이다. Argument로 0~3의 Channel 값을 넣어 주어야 한다.

2) char DigitIn(void)

Digital Input Channel의 값을 4개 Channel 단위로 Return 시켜주는 Function이다. Argument 없이 사용한다.

3) int DigitOutChan(int chan, char value)

Digital Input Channel의 값을 1개 Channel 단위로 Fill 시켜주는 Function 이다. Argument로 0~3의 Channel 값과 Channel의 ON/OFF Value을 넣어 주어야 한다.

4) int DigitOut(char value)

Digital Input Channel의 값을 4개 Channel 단위로 Fill 시켜주는 Function 이다. Argument로 4개 Channel 모두의 ON/OFF Value를 넣어 주어야 합니다.

### 3.2.2.2. Digital IN/OUT Test Example

아래의 테스트 프로그램은 Digital Input, Output Channel 의 상태에 대한 정보를 볼 수 있다.

```

void    Digital_IO_Test()
{
    int    cnt;
    char    di_value, temp_value[8], bit_value[8];

    *PDPAR(vxImmrGet()) &= ~(0xf);
    *PDDIR(vxImmrGet()) |= 0xf;
    *PDDAT(vxImmrGet()) &= ~(0xf);

    for(cnt = 0; cnt < 4; cnt++) {
        printf("Diag1 Out %d ON!!WrWn", cnt);
        *PDDAT(vxImmrGet()) |= (0x1 << cnt);
        taskDelay(50);
    }

    for(cnt = 0; cnt < 4; cnt++) {
        printf("Diag1 Out %d OFF!!WrWn", cnt);
        *PDDAT(vxImmrGet()) &= ~(0x1 << cnt);
        taskDelay(50);
    }

    di_value = *(char *)DIP_SW_READ_ADDR;
    printf("di_value = %xWn", di_value);

    for(cnt = 0; cnt < 8; cnt++) {
        temp_value[cnt] = di_value & (0x1 << cnt);
    }

    for(cnt = 0; cnt < 8; cnt++) {
        bit_value[cnt] = temp_value[7 - cnt];
    }

    for(cnt = 4; cnt < 8; cnt++) {
        if(bit_value[cnt] != 0)    printf("Digital In %d : OFFWn", cnt - 4);
    }
}

```

```

else          printf("Digital In %d : ONWn", cnt - 4);
}
}
    
```

### 3.2.3. Dip Switch 제어

#### 3.2.3.1. Function Description

LKV-402 보드에서는 4개의 Dip Switch로 구성되어 있으며, Switch on시 data는 0으로 read 되며, switch off시 data는 1로 read 된다. 이 Channel을 Control 하기 위하여 아래와 같은 Function을 사용하면 된다.

##### 1) DipSwValue(void)

Dip Switch Value의 셋팅 설정과 Ethernet을 disable 시켜주는 Function 이다.

□: ON, ■: OFF	
□□□□ : 0x8F	1000
□□□■ : 0x9F	1001
□□■■■ : 0xBF	1011
□■■■■ : 0xFF	1111

Ex) Dip Switch Value

#### 3.2.3.2. Dip Switch Test Example

아래의 테스트 프로그램은 Dip Switch의 셋팅 상태에 대한 정보를 볼 수 있다.

```

void Dip_SW_Test()
{
    int cnt;
    char dip_value, temp_value[8], bit_value[8];

    dip_value = *(char *)DIP_SW_READ_ADDR;

    for(cnt = 0; cnt < 8; cnt++) {
        temp_value[cnt] = dip_value & (0x1 << cnt);
    }
}
    
```

```
for(cnt = 0; cnt < 8; cnt++){  
    bit_value[cnt] = temp_value[7 - cnt];  
}  
  
for(cnt = 0; cnt < MAX_DIP_SW_NUM; cnt++){  
if(bit_value[cnt] != 0)    printf("Dip SW %d : OFF\n", cnt);  
else                      printf("Dip SW %d : ON\n", cnt);  
}  
}
```

## 4. LKV-402 보드 시리얼 제어 방법

### 4.1. LKV-402 보드 시리얼 드라이버 초기화

LKV-402 보드에는 6개의 Serial Port가 기본적으로 장착되어 있으며 각각의 Serial Port의 모든 신호선에는 surge protection chip이 장착되어 있다. Baud rate generation을 위한 기본 클럭은 7.3728Mhz가 공급된다. 6개의 Serial Port중 Port6은 보드 Debug를 위한 Console 포트로서 사용되며, Port5와 함께 RS-232 전용이다. 그리고 Port1와 Port2은 Ethernet 또는 RS-232 Serial Port로 사용할 수 있으며, Port3와 Port4은 RS-232 또는 RS-422 Serial Port로 사용할 수 있다. 사용자의 점퍼 설정 방법에 따라 선택적으로 사용 가능하다.

사용자의 점퍼 설정 방법에 대한 설명은 LKV-402 하드웨어 매뉴얼을 참조 하십시오.

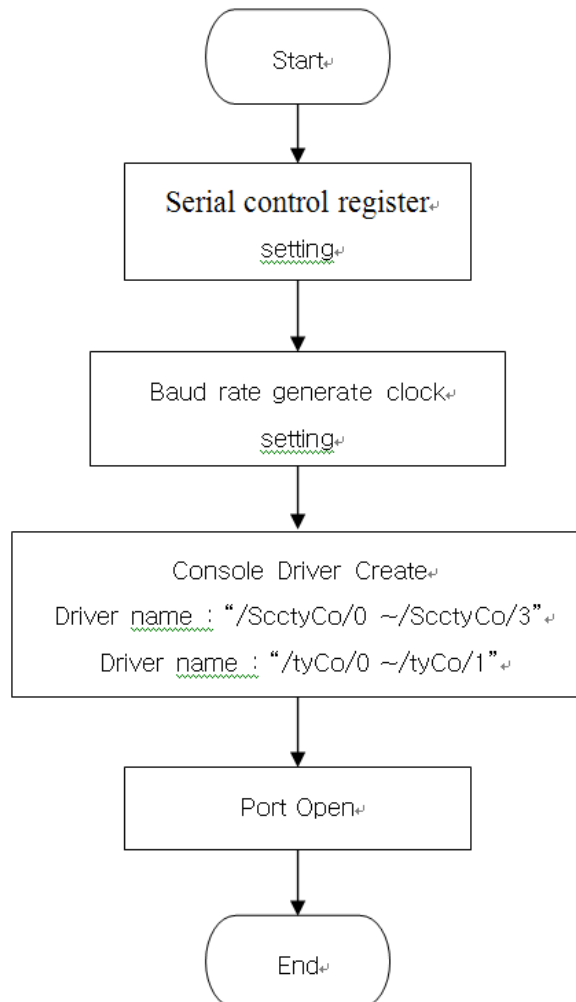


그림 4. LKV-402 B/D Serial Driver 초기화 순서

## 4.2. VxWorks Serial I/O 시스템

LKV-402 보드는 IO System을 통해서 create(), remove() open(), close(), read(), write(), ioctl() 함수를 지원하며 VxWorks 에서 일반적으로 지원하는 기능과 같다.

### 4.2.1. open ()

#### 1) Synopsis

```
int open ( const char * name, int flag, int mode )
```

#### 2) Description

##### ⓐ name

현재 생성되어야 될 프로세서의 드라이버를 가리킨다. 드라이버의 이름은 각각의 프로세서마다 다르며 파일단위로 생성됨. Name을 통해서 드라이버에 접근 가능함.

##### ⓑ flag

O\_RDONLY(0) : 드라이버 읽기 전용으로 생성

O\_WRONLY(1) : 드라이버 쓰기 전용으로 생성

O\_RDWR(2) : 드라이버 읽기/쓰기로 생성

O\_CREATE(0x0200) : 드라이버를 파일단위로 생성

##### ⓒ mode

UNIX 시스템에서 파일접근 허가를 나타내는 숫자를 수록함.

#### 3) Return Value

##### ⓐ number

파일단위로 생성된 디바이스의 개수를 나타냄.

##### ⓑ ERROR

해당 디바이스가 없거나 파일네임과 맞지 않을 경우를 나타냄.

### 4.2.2. close ()

#### 1) Synopsis

```
STATUS close ( int fd )
```

#### 2) Description

##### ⓐ fd

시스템으로부터 드라이버 파일의 사용이 끝났음을 알림.

#### 2) Return Value

##### ⓐ OK

호출이 성공적으로 이루어 졌을 경우를 나타냄.

ⓑ ERROR

해당 드라이버가 없을 경우 또는 파일 기술자가 아닐 경우.

#### 4.2.3. read ()

1) Synopsis

```
int read ( int fd, char * buffer, size_t maxbytes )
```

2) Description

개방된 드라이버 파일로부터 일정 수의 바이트를 버퍼로 복사하기 위해 사용됨.

ⓐ fd

버퍼로 읽어 들일 드라이버의 파일명.

ⓑ buffer

읽어 들일 파일의 저장공간으로 char type의 포인터로 정식 선언되며 1문자와 1바이트는 구별 없이 사용할 수 있다. 즉 buffer는 자료가 복사되어질 문자배열에 대한 포인터 임.

ⓒ maxbytes

파일로부터 읽혀질 바이트의 수를 나타내는 양의 정수임.

3) Return Value

ⓐ number

파일단위로 생성된 디바이스에서 버퍼에 저장된 바이트의 개수를 나타냄.

ⓑ ERROR

해당 디바이스가 없거나 파일네임과 맞지 않을 경우를 나타냄.

#### 4.2.4. write ()

1) Synopsis

```
int write ( int fd, char * buffer, size_t nbytes)
```

2) Description

write의 호출은 read와는 반대로 문자배열로 선언된 프로그램 버퍼로부터 개방된 파일 드라이버를 통해 일정 수의 바이트를 출력하거나 쓰기 위해 사용됨.

ⓐ fd

출력하거나 쓰기 위한 드라이버의 파일명.

ⓑ buffer

문자배열로 선언된 프로그램 버퍼로 char type의 포인터로 정식 선언되며 1문자와 1바이트는 구별 없이 사용할 수 있다. 즉 buffer는 출력될 자료의 문자배열에 대한 포인터 임.

ⓒ nbytes

파일로부터 출력되어야 할 바이트의 수를 나타내는 양의 정수임.

### 3) Return Value

ⓐ number

프로그램 버퍼로부터 디바이스에 쓰여진 바이트의 개수를 나타냄.

ⓑ ERROR

해당 디바이스가 없거나 파일네임과 맞지 않을 경우를 나타냄.

## 4.2.5. ioctl ()

### 1) Synopsis

int ioctl ( int fd, int function, int arg )

### 2) Description

Device 의 I/O 컨트롤 함수로 매우 유용하게 쓰인다.

ⓐ fd

컨트롤 할 디바이스의 터미널 명.

ⓑ function

I/O 컨트롤 함수는 디바이스 드라이버에 따라 각 함수에 대한 옵션이 정해져 있으며 제공되는 드라이버의 옵션은 다음과 같다.

- FIOBAUDRATE

Baud Rate 을 Set 한다.

- FIOGETOPTIONS

각 채널에 대한 옵션을 얻어옴.

- FIOSETOPTIONS

각 채널에 대한 옵션을 셋팅함.

- FIOFLUSH

RX Buffer clear

- FIONREAD

RX Buffer에 Read되지 못한 바이트의 수를 구함.

※ 주의 : 통신 도중에 에러가 발생해서 데이터 수신이 불안정한 경우에는 기본적으로 사용 중이던 포트를 close() API를 이용해서 사용 중지하고, 다시 open() API를 사용해서 해당 통신 포트를 초기화한 후에 read()/write() API 등을 사용해서 통신을 수행하는 것이 보다 안전합니다.



### 4.3. Serial I/O 제어 예제

#### 4.3.1. Example A

##### 4.3.4.1. open ()

다음의 예제를 통해서 위 함수의 기능을 알 수 있으며 각 포트를 쓰기 전에 OPEN 하는 방법을 알 수 있다.

```

void openPort(int channel)
{
    switch(channel){
        case 1 : fp_port[0]=open("/ScctyCo/1",2,0);
                break;
        case 2 : fp_port[1]=open("/ScctyCo/2",2,0);
                break;
        case 3 : fp_port[2]=open("/ScctyCo/3",2,0);
                break;
        case 4 : fp_port[3]=open("/tyCo/1",2,0);
                break;
    }
}
    
```

LKV-402 보드는 I/O name 이 “/ScctyCo/0 ~ /ScctyCo/3”와 “/tyCo/0 ~ /tyCo/1” 모두 6포트를 지원한다. 위 예제에서는 포트 4개의 드라이버를 오픈 할 경우 각각의 포트 name에 따라 셋팅 되어있다.

##### 4.3.1.2. close ()

```

void    UARTClose(int channel)
{
    switch(channel){
        case 1 :
            if(close(fp_port[0]) == ERROR){
                printf("Close fp_port[%d] Error!\n",channel);
                return;
            }
            break;

        case 2 :
            if(close(fp_port[1]) == ERROR){
                printf("Close fp_port[%d] Error!\n",channel);
                return;
            }
            break;

        case 3 :
            if(close(fp_port[2]) == ERROR){
                printf("Close fp_port[%d] Error!\n",channel);
                return;
            }
            break;

        case 4 :
            if(close(fp_port[3]) == ERROR){
                printf("Close fp_port[%d] Error!\n",channel);
                return;
            }
            break;

    }

    printf("Close UART %d OK!\n",channel);
}
    
```

open () 과 마찬가지로 포트 4개를 모두 반환할 경우 다음과 같이 close ()함수를 이용한다.

## 4.2.2. Example B

### 4.2.2.1. read () / write ()

다음 함수는 open된 파일 드라이버를 통해 데이터를 읽고 쓰며 count 의 정해진 횟수만큼 루프를

돈다. 다음 함수를 통하여 어떻게 시리얼 각각의 채널을 통해 읽고 쓰는지 알 수 있다.

```
void loopTest(int from, int to, int loop_cnt)
{
    char *buf;
    char *str="1234567890abcdef1234567890abcdef";
    int    k, r_byte, i=0, j=0;
    int    temp, baud, *cur_baud;

    LOOP_FLAG = 0;

    openPort(from);
    openPort(to);

    if( (from < 1 || from > 4) || (to < 1 || from > 4) ) {
        printf("Select correct port number : 1 ~ 4 Wn");
        return;
    }

    while(1) {
        if(loop_cnt == -1) {
            printf("UART Loop Test port = %d, port = %d complete!!Wn", from, to);
        }

        Write (fp_port[from-1],str+i,16);
        taskDelay(10);
        r_byte = read(fp_port[to-1],buf,16);

        for(k=0;k<r_byte;k+=1) {
            if(*(str+i+k)!=*(buf+k)) {
                printf("ERROR OCCUR - Write : %c, Read : %cWn",*(str+i+k),*(buf+k));
                break;
            }
        }

        i++;
        if(i==16) {
            LOOP_FLAG = 1;
        }
    }
}
```

```

        if(loop_cnt != -1)    break;

        i=0;
        j++;
        if (j==5) {
            j = 0;

            if(loop_cnt != -1) {
                ioctl (fp_port[from - 1], SIO_BAUD_SET, getAutoBaudRate(from));
                ioctl (fp_port[to - 1], SIO_BAUD_SET, getAutoBaudRate(to));
                ioctl (fp_port[from - 1], SIO_BAUD_GET, cur_baud);
                printf("channel %d Set BaudRate : %dWn", from - 1, *cur_baud);
                ioctl (fp_port[to - 1], SIO_BAUD_GET, cur_baud);
                printf("channel %d Set BaudRate : %dWn", to - 1, *cur_baud);
            }
        }
    }
    temp = to;
    to = from;
    from = temp;
}

UARTClose(from);
UARTClose(to);
}

```

from은 시작 포트를 나타내며, to은 마지막으로 검사할 채널의 번호를 나타낸다. BaudRate는 test할 채널의 baud rate로써 다양하게 지원하지만 default로 9600으로 지정되어 있다.

위의 loopTest 함수는 write 함수와 read 함수를 통해 각 채널의 드라이버 상태를 검사 할 수 있으며, 예로 0번 채널을 통해 드라이버에 write한 후 1번 채널에서 read 하여 write된 바이트 수와 read된 바이트 수를 비교하게 된다. 이러한 과정을 각 채널별로 하며 모든 채널에 대해 수행하게 된다.

## 5. 주의 사항

### CAUTION

- VME Rack에서 가장 왼쪽에 위치한 보드는 System Controller로 설정할 것.
- System Controller 보드는 하나의 Rack에 하나만 장착할 것.
- System Controller 보드 이외의 보드는 마스터/Slave 보드로 설정할 것.
- LKV-402 보드가 장착되는 Rack의 GND는 FGND와 직접 연결 하거나 Capacitor를 통해 연결할 것.

### WARNING

- LKV-402 보드는 정전기(Electrostatic Discharge)에 취약할 수 있으니, 보드 취급 시 주의할 것.
- LKV-402 보드를 Rack에 장착할 시에는 가능하면 전원을 끄고 작업할 것.
- Rack이 접지되지 않았을 경우 감전의 우려가 있으므로, 반드시 접지 여부를 확인하고, 물이나 땀이 묻은 손으로 작업하지 말 것.



**㈜엘케이일레븐**

138-809 서울특별시 송파구 가락2동 545-5번지 동명빌딩 3층

<http://www.lk11.com>, 전화:02-3012-3788