

PCI/PCIe Serial Card

리눅스 사용자 매뉴얼



Revision History

수정일	버전	페이지	설명
04/04/2013	1.8	부분 수정	내용 추가
05/14/2013	1.9	부분 수정	내용 추가
01/15/2014	2.0	부분 수정	내용 추가
01/30/2014	2.1	부분 수정	내용 추가
02/07/2014	2.2	부분 수정	내용 추가
02/12/2016	2.3	부분 수정	Ubuntu 14.04 LTS (kernel 3.13) 추가 Ubuntu 14.10 (kernel 3.16) 추가 Ubuntu 15.04 (kernel 3.19.3) 추가 Ubuntu 15.10 (kernel 4.2.0) 추가 Ubuntu latest version + last update (kernel 4.3.x) 추가 리눅스 커널 4.3 지원.
08/02/2018	2.4	부분 수정	UBuntu16.04 LTS (kernel 4.4) 추가 Ubuntu18.04 LTS (Kernel 4.15) 추가 Ubuntu latest version + last update (kernel 4.16) 추가 CentOS7 (kernel 3.10) 추가 CentOS7 (kernel 4.16) 추가 OpenSuse 42.3 (kernel 4.4) 추가 OpenSuse 15 (kernel 4.12) 추가
12/12/2018	2.5	부분 수정	CentOS 7 kernel 버전 내용 수정
02/26/2020	2.6	부분 수정	CentOS 7.7 (kernel 3.10) 추가 CentOS 8.1 (kernel 4.18) 추가 Ubuntu 19.04 (kernel 5.0) 추가 Ubuntu 19.10 (kernel 5.3) 추가 Ubuntu latest version + last update (kernel 5.3) 추가 OpenSuse 15.1 (kernel 4.12.14) 추가 Fedora 28 (kernel 4.16) 추가 Fedora 29 (kernel 4.18) 추가 Fedora 31 (kernel 5.3) 추가

Copyright 2006-2018 SystemBase Co., Ltd. All rights reserved.

홈페이지 <http://www.sysbas.com/>

전화 (02) 855-0501

FAX 82-2-855-0580

서울시 구로구 디지털로 288, 대륭포스트타워 1 차 1601 호

기술 문의 용 email tech@sysbas.com

목차

1. 개요	2
1.1 지원 리눅스 커널	2
2. 드라이버 설치 환경	4
3. 드라이버 설치	6
4. 드라이버 삭제	7
5. 드라이버 테스트	8

1. 개요

시스템베이스의 시리얼 카드는 PC의 PCI/PCIe 슬롯에 장착하여 하나의 카드로 1, 2, 4, 8, 16, 24 또는 32 개의 RS232/RS422/RS485 시리얼 통신 포트를 제공합니다.

PCI/PCIe 시리얼카드 드라이버는 스크립트 형태로 제공되고 있으며, 고객이 편리하게 드라이버를 설치할 수 있습니다.

1.1 지원 리눅스 커널

Supported Kernel: From version 2.4.18 to 5.3.0

Tested Linux		
Linux distribution	Kernel version	Architecture
Red hat 9.0	-	i386
Ubuntu 11.10	-	i386
CentOS 6	-	i386
CentOS 7	3.10.0-862	x86_64
	3.10.0-1062	x86_64
	4.16	x86_64
CentOS 8.1	4.18.0-147	x86_64
Fedora 28	4.16	x86_64
Fedora 29	4.18	x86_64
Fedora 31	5.3	x86_64
OpenSuse 42.3	4.4	x86_64
OpenSuse 15	4.12	x86_64
OpenSuse 15.1	4.12.14	x86_64
Ubuntu 6.06.1	2.6.15	i386/amd64
Ubuntu 6.10	2.6.17	i386/amd64
Ubuntu 7.04	2.6.20	i386/amd64
Ubuntu 9.10	2.6.31	i386/amd64
Ubuntu 10.10	2.6.35	i386/amd64
Ubuntu 11.10	3.0.0	i386/amd64
Ubuntu 12.04.1 LTS	3.2.0	i386/amd64
Ubuntu 12.10	3.5.0	i386/amd64
Ubuntu 13.04	3.8.0	i386/amd64
Ubuntu 13.10	3.11.0	i386/amd64
Ubuntu 14.04 LTS	3.13.0	i386/amd64
Ubuntu 14.10	3.16.0	i386/amd64

Ubuntu 15.04	3.19.3	i386/amd64
Ubuntu 15.10	4.2.0	i386/amd64
Ubuntu 16.04 LTS	4.4.0	i386/amd64
Ubuntu 17.10	4.13.0	i386/amd64
Ubuntu 18.04 LTS	4.16.0	amd64
Ubuntu 19.04	5.0.0	amd64
Ubuntu 19.10	5.3.0	amd64

커널 2.4 이하 버전을 사용하기 위해서는 sysbas_mpdv.v21 버전을 이용하시기 바랍니다.

2. 드라이버 설치 환경

1. 루트(root) 사용자 로그인 (Superuser)

루트 계정으로 로그인해야 하며 프롬프트에 #이 표시되는 것을 확인할 수 있습니다.

루트 계정으로 로그인 하지 않으면, "\$"이 표시되며 드라이버가 정상적으로 설치되지 않습니다.

2. GCC (GNU C Compiler) 설치

드라이버 설치 전에 GCC를 설치합니다. 만일 'gcc -v'를 입력했을 때 "gcc: command not found"가 출력되면 GCC가 설치되지 않았기 때문입니다.

```
sysbas@utu:/tmp# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/i686-linux-gnu/4.6.1/lto-wrapper
Target: i686-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 4.6.1-9ubuntu3' --with-
bugurl=file:///usr/share/doc/gcc-4.6/README.Bugs --enable-languages=c,c++,fortran,objc,obj-c++,go --
prefix=/usr --program-suffix=-4.6 --enable-shared --enable-linker-build-id --with-system-zlib --libexecdir=/usr/lib --
without-included-gettext --enable-threads=posix --with-gxx-include-dir=/usr/include/c++/4.6 --libdir=/usr/lib --
enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-
plugin --enable-objc-gc --enable-targets=all --disable-werror --with-arch-32=i686 --with-tune=generic --enable-
checking=release --build=i686-linux-gnu --host=i686-linux-gnu --target=i686-linux-gnu
Thread model: posix
gcc version 4.6.1 (Ubuntu/Linaro 4.6.1-9ubuntu3)
```

<GCC가 정상적으로 설치된 경우>

```
sysbas@utu:/tmp# gcc -v
gcc: command not found
```

<GCC가 설치되지 않은 경우>

3. 커널 소스 설치

드라이버 설치 전에 커널소스 파일들을 설치하고 'cd /usr/src'를 입력하여 파일들이 정상적으로 설치된 것을 확인합니다.

```
sysbas@utu:/tmp# cd /usr/src
sysbas@utu:/src# ls
linux-headers-3.0.0-12          linux-headers-3.0.0-12-generic
```

<커널소스가 정상적으로 설치 된 경우>

```
sysbas@utu:/tmp# cd /usr/src
sysbas@utu:/src# ls
```

<커널소스가 설치되지 않았을 경우>

4. Make 유틸리티 설치

드라이버 설치 전에 "make"가 설치되었는지 확인합니다.

'make -v'를 입력했을 때 "make: command not found"가 출력되면 "make"가 설치되지 않은 것 입니다.

```
sysbas@utu:/tmp# make -v
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for i686-pc-linux-gnu
```

<Make가 정상적으로 설치 된 경우>

```
sysbas@utu:/tmp# make -v
make: command not found
```

<Make가 설치되지 않은 경우>

3. 드라이버 설치

1. 시스템 전원을 끕니다.
2. PCI/PCIe 시리얼카드를 PCI/PCIe 슬롯에 장착합니다.
3. 시스템 전원을 켭니다.
4. 리눅스 운영체제가 부팅하면, 관리자(root) 계정으로 로그인을 합니다.
5. “./sysbas_mpdv.v**(version).sh”을 입력하여 디바이스 드라이버 설치파일을 실행합니다.
드라이버가 설치되면 시리얼카드의 모델명, RS232/RS422/RS485 인터페이스 종류, 포트명과 버전을 확인 할 수 있습니다.

```

root@utu:/tmp# ./sysbas_mpdv.v23.0.sh
Verifying archive integrity... All good.
Uncompressing Enhanced Async Multi-Port(PCI/PCIe) Linux device driver.....
=====
Enhanced Async Multi-Port(PCI/PCIe) Linux Device Driver
Version : 23.0      revision: 2018-05-15
=====
1 board(s) installed
Board No.1 : Multi-4 PCI (rev b0)
    /dev/ttyMP0 (RS232 , 16C105X)
    /dev/ttyMP1 (RS232 , 16C105X)
    /dev/ttyMP2 (RS232 , 16C105X)
    /dev/ttyMP3 (RS232 , 16C105X)
root@utu:/tmp#

```

설치된 시리얼포트는 `dmesg | grep tty` 명령어로 확인할 수 있습니다.

```

root@utu:/tmp# dmesg | grep tty
[ 0.000000] console [tty0] enabled
[ 0.627793] 00:07: ttyS0 at I/O 0x3f8 (irp = 4, base_baud = 115200) is a 16550A
[ 12.993516] ttyMP0 at I/O 0xe040 (irp = 16) is a SB16C1050
[ 12.993967] ttyMP1 at I/O 0xe048 (irp = 16) is a SB16C1050

```


4. 드라이버 삭제

1. 드라이버가 설치된 경로로 이동한다. 만일 “~/tmp”경로에 드라이버가 설치되었다면 “~/tmp/sysbas_mpdrv.v**(version)”라고 입력하여 해당 경로로 이동합니다.

```
#cd ~/tmp/sysbas_mpdrv.v23.0
```

```
root@utu:# cd ~/tmp/sysbas_mpdrv.v23.0
root@utu:/tmp/sysbas_mpdrv.v23.0# ls
Install  Remove async_multiport ioctl multidrop_test
```

2. 아래와 같이 드라이버 삭제 명령을 실행합니다.

```
#!/Remove
```

```
root@utu:/tmp/sysbas_mpdrv.v23.0# ls
Install  Remove async_multiport ioctl multidrop_test
root@utu:/tmp/sysbas_mpdrv.v23.0# ./Remove
```

3. Remove 명령이 실행되면 설치된 모든 드라이버 파일들이 자동으로 삭제됩니다.

```
Remove Multiports PCI/PCle Driver..!!
remove device(/dev)....done
modify rc.local....done
root@utu:/tmp/sysbas_mpdrv.v23.0#
```

5. 드라이버 테스트

```
#cd async_multiport
```

```
#./sb_test [Port Name] [Baudrate] [TestMode]
```

sb_test 프로그램 사용법을 확인하려면 “./sb_test”만 입력하면 됩니다.

```
root@utu:/tmp/sysbas _mpdrv.v23.0/async_multiport# ./sb_test
Usage: ./sb_test [Port Name] [Baudrate] [TestMode]
Port Name : /dev/ttyMP0 ~ /dev/ttyMP32
Baudrate  : 9600, 19200, ...
TestMode  : 0(Loopback) 1(Send) 2(Recv)
root@utu:/tmp/sysbas _mpdrv.v23.0/async_multiport#
```

Usage: ./sb_test [Port Name] [Baudrate] [TestMode]

```
Port Name      : /dev/ttyMP0 ~ /dev/ttyMP32
Baudrate       : 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600
TestMode       : 0 (Loopback)
                  1 (Send)
                  2 (Receive)
```

사용예제:

```
#./sb_test /dev/ttyMP0 9600 0
#./sb_test /dev/ttyMP5 921600 0
#./sb_test /dev/ttyMP3 115200 1
```

참고:

- “./sb_test”뒤와 “/dev/”앞에는 공백을 둡니다.
- 32포트 시리얼카드 경우, RS485 echo 모드는 지원하지 않습니다.

루프백 모드를 사용하여 테스트하고자 하는 시리얼 포트에 루프백 테스트를 할 수 있습니다.

테스트 패턴은 “abcdefghijklmnopqrstuvwxyz”이며 프로그램이 “a”에서 “z”까지 한 문자씩 증가하는 반복적인 문자열을 전송합니다.

```
root@utu:/tmp/sysbas_mpdrv.v23.0/async_multiport# ./sb_test /dev/ttyMP0 9600 0
Loopback Test Mode  !
a
ab
abc
abcd
abcde
abcdef
abcdefg
abcdefgh
abcdefghi
abcdefghij
abcdefghijk
abcdefghijkl
abcdefghijklm
abcdefghijklmn
abcdefghijklmno
abcdefghijklmnop
```