

Fall 2004 Embedded Systems

실험 3. Serial Communication

1. Purpose

PC Linux 에서 Serial port 를 이용하여 file 을 송수신하는 프로그램을 작성하여, serial 통신에 대한 기본개념을 확고히 한다.

2. Problem 3

Write an application program named "recvs" in user space of Linux PC, which is a receiver program for serial file downloading. An application program named "sends" will be provided, which is a sender program.

3. Hardware 설정

Serial downloading 을 수행하려면 두 대의 컴퓨터가 필요하며, 각각의 serial interface 를 serial cable 로 연결하여야 함이 원칙이다.

그러나 본 실험에서는 한대의 PC 사용하기로 한다. Sender program 은 serial port 0 (COM1)를 사용하고, receiver program 은 serial port 1 (COM2)를 사용한다. 이 두 serial port 는 null modem cable 로 연결한다. EZ-X5 용 serial cable 이 null modem cable 이므로 이 cable 로 PC 의 COM1 과 COM2 를 직접 연결하라.

결과적인 연결은 다음과 같다.

PC (Sender terminal) -> COM1 -> Null modem cable -> COM2 -> PC (Receiver terminal)

EZ-X5 는 이 실험에서는 사용하지 않는다.

4. 실험 순서

4.1 Linux command 에 의한 file 전송

1) PC 에서 X-window 를 수행시킨다.

```
# startx
```

2) X-window 에서 두개의 command window 를 만든다. X-window menu 에서 System Tools - Terminal 을 두 번 선택한다. 하나는 sender 가, 다른 하나는 receiver 가 사용하기로 한다. Sender terminal 의 prompt 를 '#'로, receiver terminal 의 prompt 를 '@'로 나타내기로 한다.

3) Sender terminal 에서 다음과 같은 command 로 serial port 0 (COM1)을 설정할 수 있다.

```
# stty -F /dev/ttyS0 115200 cs8 -parenb -cstopb line 0 -brkint -imaxbel -icrnl
```

4) Receiver terminal 에서 다음과 같은 command 로 serial port 1 (COM2)를 설정할 수 있다.

```
@ stty -F /dev/ttyS1 115200 cs8 -parenb -cstopb line 0 -brkint -imaxbel -icrnl
```

5) 이 command 의 각 항목의 의미를 설명하라. Web 에서 <http://www.yolinux.com/HOWTO/Serial-HOWTO.html> 참조.

6) 설정된 serial parameter 를 확인한다.

```
# stty -F /dev/ttyS0
```

```
@ stty -F /dev/ttyS1
```

7) Receiver terminal 에서 다음 command 로 serial receiver 를 구동시킨다.

```
@ cat /dev/ttyS1
```

8) Sender terminal 에서 다음 command 로 serial sender 를 구동시킨다. source_ascii_file 은 첨부된 sends.c 를 사용할 수 있다.

```
# cat source_ascii_file >/dev/ttyS0
```

- 9) Receiver terminal 에 source_ascii_file 의 내용이 보이면 성공!
- 10) Receiver 를 정지시키려면, receiver terminal 에서 Ctrl+C 를 입력한다.

4.2 Application program sender 와 Linux command receiver 에 의한 전송

- 1) Receiver terminal 에서는 4.1 과 동일하게 입력한다.
- 2) Sender terminal 에서는 첨부된 sends.c file 을 compile 한다. 물론 compiler 는 arm-linux-gcc 가 아니고 gcc 를 사용하여야 한다.
gcc -o sends sends.c
- 3) Sender terminal 에서 sender program 을 수행시킨다. source_ascii_file 은 sends.c 를 사용할 수 있다.
./sends source_ascii_file
- 4) Receiver terminal 에 source_ascii_file 의 내용이 보이면 성공!

4.3 Application program sender 와 application program receiver 에 의한 file 전송.

- 1) Web 에서 Serial-Programming-HOWTO.html 문서를 찾아 읽는다.
- 2) Sends.c program 을 이해한다.
- 3) 이를 바탕으로 Application program 인 recvx.c program 을 작성한다.
@ vi recvs.c
- 4) recvs.c 를 compile 한다.
@ gcc -o recvs recvs.c
- 5) Receiver terminal 에서 recvs program 을 수행시킨다.
@ ./recvs destination_ascii_file
- 6) Sender terminal 에서 sends program 을 수행한다.
./sends source_ascii_file
- 7) Receiver terminal 에서 destination_ascii_file 이 생성되고, 그 내용이 source_ascii_file 과 동일하면 성공!
- 8) Binary file (예를 들어 sends file) 도 같은 program 으로 전송이 가능한지 test 하라.

5. Demonstration (To TA, 10/28 Thu. 4 PM – 6 PM)

6. Report (To TA, Due 11/1 Mon 6 PM)

Each student should prepare his own report containing:

- Purpose
- Experiment sequence
- Experimental results
- Discussion: should be different even for each member of the same team.
- References.

7. 첨부: sends.c program

```

/*          File sends.c
*
*   File send program without transmitting the file length
*
*   Programmed by Byung Kook Kim,      Oct. 23, 2003
*/

#include<stdio.h>

```

```

#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<sys/ioctl.h>
#include<fcntl.h>
#include<linux/types.h>
#include<linux/serial.h>
#include<termios.h>
#include<string.h>
#include<sys/signal.h>

#define BAUDRATE    B115200                //BAUD RATE setting
#define MODEMDEVICE "/dev/ttyS0"          //serial port
#define _POSIX_SOURCE    1                //POSIX source
#define FALSE 0
#define TRUE 1

//open dev setting...
#define _O_RDONLY    0x0001
#define _O_WRONLY    0x0002
#define _O_RDWR     0x0004
#define _O_CREAT     0x0100
#define _O_TRUNC     0x0200
#define _O_EXCEL     0x0400
#define _O_APPEND    0x0800
#define _O_TEXT      0x4000
#define _O_BINARY    0x8000

volatile int stop=FALSE;

int main(int argc, char *argv[])
{
    FILE *ffile;
    int fd,i;
    struct termios oldtio,newtio;          //termios structure
    char buffer[258];
    unsigned long total_len;              //total length of size.
    unsigned long len,imsi_len;           //temporary length.

// 1. Check arguments
    if (argc != 2) {
        fprintf(stderr, "Usage: sends <source_filename>Wn");
        exit(-1);
    }

// 2. Open input datafile
    if (!(ffile=fopen(argv[1],"rb"))) { //read file as binary
        fclose(ffile);
        fprintf(stderr, "Cannot open %sWn", argv[1]);
        exit(-2);
    }
}

```

```

    }
    fprintf(stderr, "Starting send2 %s ...WrWn", argv[1]);

// 3. Open SIO
fd=open(MODEMDEVICE, _O_RDONLY | _O_BINARY ); //open serial port
if (fd < 0)
{
    perror(MODEMDEVICE);
    exit(-3);
}

tcgetattr(fd,&oldtio);          //save current setting to oldtio
bzero(&newtio,sizeof(newtio));  //clear struct for new port
newtio.c_cflag=BAUDRATE | CS8 | CLOCAL | CREAD;
                                //control mode flag
newtio.c_iflag=IGNPAR;         //input mode flag
newtio.c_oflag=0;              //output mode flag
newtio.c_lflag=0;              //local mode flag
newtio.c_cc[VTIME]=10;         // Timer between character
newtio.c_cc[VMIN]=255;         //block for next 255 character

//init modem line & port setting..
tcflush(fd,TCIFLUSH);
tcsetattr(fd,TCSANOW,&newtio);

/* Terminal settings done      */

// 5. Read the source file and send it iteratively
while (stop == FALSE) {
    len = fread(buffer, sizeof(char), 256, ffile);
                                //binary stream input
    write(fd, buffer, len);      //send char to receiver
    fprintf(stderr, "%d ", len);
    if (len <= 0) stop = TRUE;
}
printf("Wn");

// 8. Restore the old port settings
tcsetattr(fd,TCSANOW,&oldtio);

// 9. Close files
fclose(ffile);
close(fd);

printf("Sends completedWn");
return 0;
}

```
