

Новосибирский Государственный Университет

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Курс “ЭВМ и периферийные устройства”

Лабораторная работа №6.2

**«ПРОГРАММИРОВАНИЕ КОММУНИКАЦИЙ С ПЕРИФЕРИЙНЫМИ
УСТРОЙСТВАМИ ЧЕРЕЗ ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ ПЕРЕДАЧИ
ДАННЫХ»**

Выполнил:

Пятаев Егор, гр. 15206

Преподаватель:

Городничев Максим Александрович

Новосибирск 2016

Цели работы

1. Изучить протокол передачи данных для последовательного канала передачи данных;
2. Научиться программировать обмен данными между компьютером и внешним устройством через последовательный канал передачи данных, используя программный интерфейс ОС Linux.

Листинг реализованной программы

main.c:

```
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define BAUDRATE B9600
#define MODEMDEVICE "/dev/ttyACM0"
#define _POSIX_SOURCE 1
#define FALSE 0
#define TRUE 1
volatile int STOP=FALSE;

int main(){
    int fd; int c; int res;
    struct termios oldtio, newtio;
    char buf[255];
    fd = open(MODEMDEVICE, O_RDWR | O_NOCTTY );
    if (fd <0) {
        perror(MODEMDEVICE); exit(-1);
    }
    tcgetattr(fd,&oldtio);
    memset(&newtio, 0, sizeof(newtio));
    newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
    newtio.c_iflag = IGNPAR | ICRNL;
    newtio.c_oflag = 0;
    newtio.c_lflag = ICANON;
    newtio.c_cc[VINTR]= 0;
    newtio.c_cc[VQUIT]= 0;
    newtio.c_cc[VERASE]= 0;
    newtio.c_cc[VKILL]= 0;
    newtio.c_cc[VEOF]= 4;
    newtio.c_cc[VTIME]= 0;
    newtio.c_cc[VMIN]= 1;
    newtio.c_cc[VSWTC]= 0;
    newtio.c_cc[VSTART]= 0;
    newtio.c_cc[VSTOP]= 0;
    newtio.c_cc[VSUSP]= 0;
    newtio.c_cc[VEOL]= 0;
```

```

newtio.c_cc[VREPRINT]= 0;
newtio.c_cc[VDISCARD]= 0;
newtio.c_cc[VWERASE]= 0;
newtio.c_cc[VLNEXT]= 0;
newtio.c_cc[VEOL2]= 0;
tcflush(fd, TCIFLUSH);
tcsetattr(fd,TCSANOW,&newtio);

printf("Write data in [[[data]]] format.\n");
while (STOP==FALSE) {
    printf("DATA: ");
    scanf("%s", buf);
    if(write(fd, buf, strlen(buf)) == -1){
        fprintf(stderr, "failed to write to port\n");
        break;
    }
    usleep(500000);

    res = read(fd,buf,255);
    buf[res]=0;

    if(res == 1){
        res = read(fd,buf,255);
        buf[res]=0;
        printf("2:%s:%d\n", buf, res);
    }

    else printf("1:%s:%d\n", buf, res);

    if (buf[3]=='z') STOP=TRUE;
}

tcsetattr(fd,TCSANOW,&oldtio);

return 0;
}

```

Выводы

Для выполнения поставленных целей была написана программа для передачи данных по последовательного канала передачи данных RS-232. Было произведено считывание и передача символов на устройство, находящееся на удаленном сервере.