

# Corrigé du TD2 Liaison série

## 1 Le fichier d'entête serie.h

```
1  /*
2   * File:   serie.h
3   * Author: psimier
4   *
5   * Created on 10 décembre 2018, 08:57
6   */
7
8  #ifndef SERIE_H
9  #define SERIE_H
10
11  typedef enum {NOECHO, ECHO} typeEcho;
12
13  #include <stdio.h>
14  #include <stdlib.h>
15  #include <unistd.h> //read write open
16  #include <string.h> // strlen
17  #include <fcntl.h>
18  #include <errno.h> // erreur sytème
19  #include <termios.h> // structure pour configuration serie
20  #include <sys/ioctl.h> // fonctions pour accéder au driver
21
22
23  int OuvrirPort(const char *device);
24  void configurerSerie (int fd, const int baud);
25  void recevoirMessage(int fd, char *message, char fin, typeEcho echo);
26  void envoyerCaractere (const int fd, const unsigned char c);
27  int  envoyerMessage (const int fd, const char *s);
28  void viderBuffer (const int fd);
29
30  #endif /* SERIE_H */
```

## 2 le fichier serie.c

```
7  #include "serie.h"
8
9  int OuvrirPort(const char *device){
10     int fd = -1;
11     fd = open(device, O_RDWR | O_NOCTTY);
12     if ( fd == -1 ) {
13         printf("pb ouverture: %s\n", strerror(errno));
14         exit(EXIT_FAILURE);
15     }
16     return fd;
17 }
18
```

### La fonction **configurerSerie**

```
void configurerSerie (int fd, const int baud){

    struct termios term;
    speed_t myBaud;

    switch (baud){
        case 50:    myBaud =  B50 ; break ;
        case 75:    myBaud =  B75 ; break ;
        case 110:   myBaud =  B110 ; break ;
        case 134:   myBaud =  B134 ; break ;
        case 150:   myBaud =  B150 ; break ;
        case 200:   myBaud =  B200 ; break ;
        case 300:   myBaud =  B300 ; break ;
        case 600:   myBaud =  B600 ; break ;
        case 1200:  myBaud =  B1200 ; break ;
        case 1800:  myBaud =  B1800 ; break ;
        case 2400:  myBaud =  B2400 ; break ;
        case 4800:  myBaud =  B4800 ; break ;
        case 9600:  myBaud =  B9600 ; break ;
        case 19200: myBaud =  B19200 ; break ;
        case 38400: myBaud =  B38400 ; break ;
        case 57600: myBaud =  B57600 ; break ;
        case 115200: myBaud = B115200 ; break ;
        case 230400: myBaud = B230400 ; break ;

        default:
            myBaud =  B9600;
    }

    tcgetattr(fd, &term);
```

```
/* mode RAW, pas de mode canonique, pas d'écho */
term.c_iflag = IGNBRK;
term.c_lflag = 0;
term.c_oflag = 0;

/* 1 seul caractère suffit */
term.c_cc[VMIN] = 1;

/* Donnée disponible immédiatement */
term.c_cc[VTIME] = 0;

/* Inhibe le controle de flux XON/XOFF */
term.c_iflag &= ~(IXON|IXOFF|IXANY);

/* 8 bits de données, pas de parité */
term.c_cflag &= ~(PARENB | CSIZE);
term.c_cflag |= CS8;

/* vitesse de la transmission */
term.c_cflag |= myBaud;
tcsetattr(fd, TCSANOW, &term);

// play with DTR
int iFlags;

// turn off DTR
iFlags = TIOCM_DTR;
ioctl(fd, TIOCMBIC, &iFlags);
}
```

```

84 void recevoirMessage(int fd, char *message, char fin, typeEcho echo){
85     int erreur = 0;
86     char caractere_recu;
87     do{
88         erreur = read(fd,message,1);
89
90         if (erreur == -1){
91             printf("Erreur de reception\n");
92             exit(-1);
93         }
94         caractere_recu = *message;
95         if (echo == ECHO){
96             envoyerCaractere(fd,*message);
97         }
98         message++; // On passe au caractère suivant
99     }
100     while(caractere_recu != fin);
101     *message = '\0'; // fin de chaîne de caractère
102 }
103
104 void envoyerCaractere (const int fd, const unsigned char c){
105     write(fd,&c,1);
106 }
107
108 int envoyerMessage (const int fd, const char *s){
109     int longueur = 0;
110     int nb = 0;
111     longueur = strlen(s);
112     nb = write(fd, s, longueur );
113     if ( nb == -1 ) {
114         printf("pb avec write: %s\n", strerror(errno));
115         exit(EXIT_FAILURE);
116     }
117     return nb;
118 }
119
120 void viderBuffer (const int fd){
121     tcflush (fd, TCIOFLUSH) ;
122 }
123

```

### 3 Exemple de programme principale :

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "serie.h"
4
5  int main(int argc, char** argv) {
6
7      int fdSerie;
8      char message[1000];
9
10     fdSerie = OuvrirPort("/dev/ttyS0");
11     configurerSerie(fdSerie, 9600);
12     printf("la liaison série ttyS0 est configurée à 9600 8N1\n");
13
14     // reception de message avec echo des caractères reçus
15     do{
16         recevoirMessage(fdSerie, message, '\r', ECHO);
17         printf("Message reçu : %s\n", message);
18         envoyerMessage(fdSerie, "\r\nBien reçu\r\n");
19     }
20     while(strncmp(message, "bye", 3));
21     // tant que le message reçu est différent de bye
22
23     return (EXIT_SUCCESS);
24 }
25
```