

Approfondissement sur la liaison série : La librairie termios

Application à la configuration du port
RS232 en C

tcgetattr

- Ces paramètres sont stockés en mémoire par l'OS et influent directement sur le **pilotage matériel** du port série.
- 2 fonctions importantes :
 - `tcgetattr(int fildes, struct termios *termios_p);`
Permet de récupérer les attribus du port *fildes* et de les copier dans la structure *termios_p* créée au préalable

```
struct termios toptions;  
// on suppose ici que le port /dev/ttyACM0 a été ouvert avec open et que le  
//descripteur du port est stocké dans sc->port (un entier)
```

```
tcgetattr(sc->port, &toptions);
```

toptions

9600
8 N 1

tcgetattr

Données système en mémoire

9600
8 N 1

man termios

- Le port série est configuré par un ensemble de paramètres. Dans notre cas (comme vu avec minicom) :

9600 8N1

9600 : le débit d'envoi et de reception

8 : la taille des octets envoyés (ici 8 bits)

N : le niveau de parité, ici N pour "None"

1 : le nombre de stopbits, ici 1

Données système en mémoire

9600
8 N 1

tcsetattr

- Ces paramètres sont stockés en mémoire par l'OS et influent directement sur le **pilotage matériel** du port série.
- 2 fonctions importantes :
 - `int tcsetattr(int fd, int optional_actions, const struct termios *termios_p);`
Permet d'écrire les valeurs contenue par *termios_p* dans le descripteur de fichier du port *fildes*

```
struct termios toptions;  
//...
```

```
// on suppose ici que toption contient les valeurs souhaitées  
tcsetattr(sc->port, TCSANOW, &toptions);
```

toptions

9600
8 N 2

tcsetattr

Données système en mémoire

9600
8 N 1

Important

- On n'écrase pas (sans `tcsetattr`) les valeurs du port aveuglément !

Care must be taken when changing the terminal attributes. Applications should always do a `tcgetattr()`, save the **termios** structure values returned, and then do a `tcsetattr()`, [changing only the necessary fields. The application should use the values saved from the `tcgetattr\(\)` to reset the terminal state whenever it is done with the terminal. This is necessary because terminal attributes apply to the underlying port and not to each individual open instance; that is, all processes that have used the terminal see the latest attribute changes.](#)

A program that uses these functions should be written to catch all signals and take other appropriate actions to ensure that when the program terminates, whether planned or not, the terminal device's state is restored to its original state.

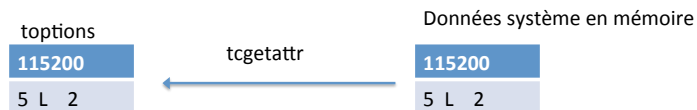
On crée toption

toptions	
0000	
0 0 0	

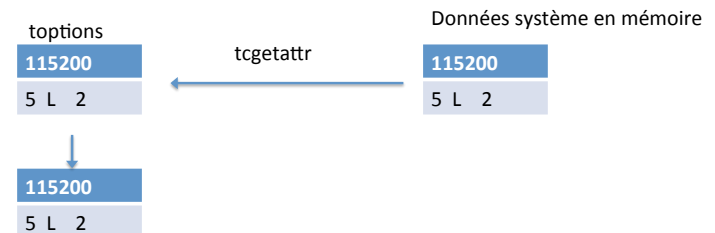
Données système en mémoire

115200
5 L 2

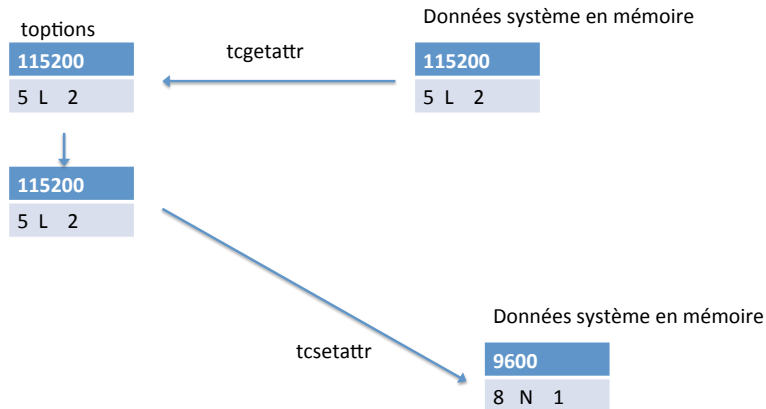
On recopie les valeurs dans toption



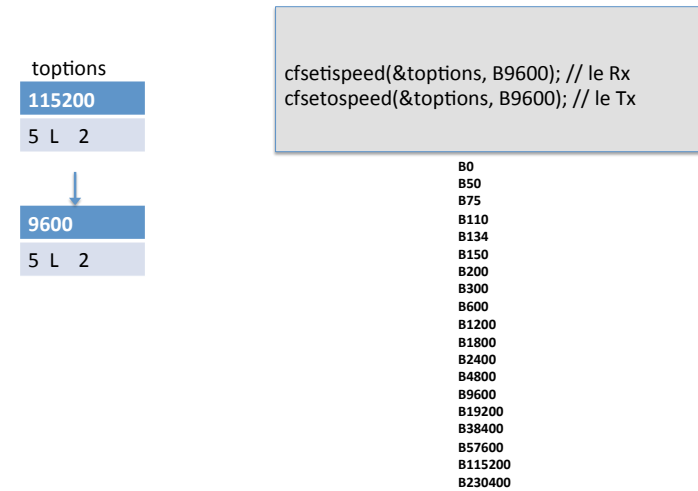
On change les valeurs dans toption



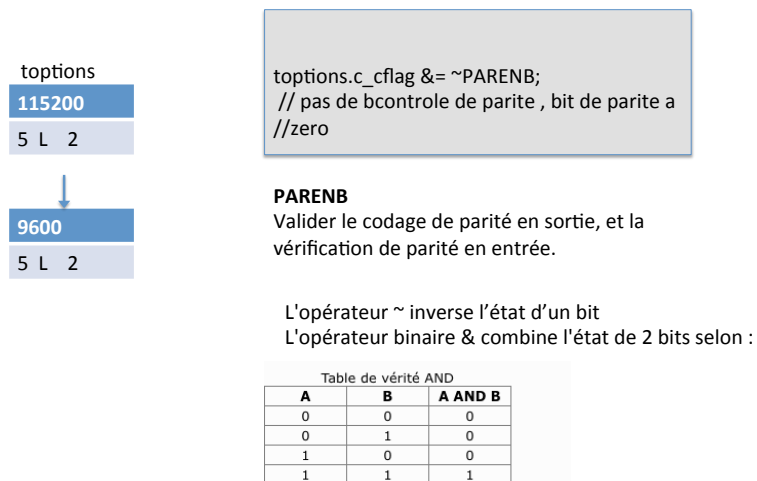
On écrase les données système



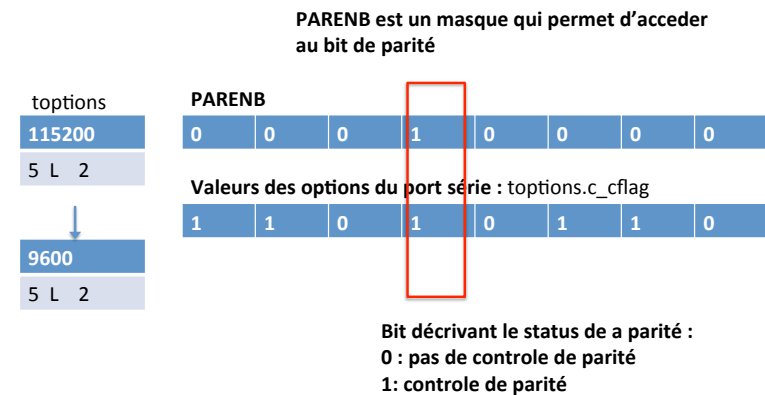
On change les valeurs dans toption



On change les valeurs dans toption



Détail :



Détail :

On souhaite affecter seulement le bit de parité, et le mettre a 0

toptions
115200
5 L 2

↓

9600
5 L 2

PARENB							
0	0	0	1	0	0	0	0

Valeurs des options du port série : toptions.c_cflag

1	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---

Bit décrivant le status de a parité :
0 : pas de controle de parité
1: controle de parité

Détail :

On prends la négation de PARENB

toptions
115200
5 L 2

↓

9600
5 L 2

~PARENB							
1	1	1	0	1	1	1	1

Valeurs des options du port série : toptions.c_cflag

1	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---

Bit décrivant le status de a parité :
0 : pas de controle de parité
1: controle de parité

Détail :

On applique un ET logique & bit a bit avec le registre

toptions
115200
5 L 2

↓

9600
5 L 2

~PARENB							
1	1	1	0	1	1	1	1

Valeurs des options du port série : toptions.c_cflag

1	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---

Bit décrivant le status de a parité :
0 : pas de controle de parité
1: pas de controle de parité

Détail :

On applique un ET logique & bit a bit avec le registre

toptions
115200
5 L 2

↓

9600
5 L 2

~PARENB							
1	1	1	0	1	1	1	1

Valeurs des options du port série : toptions.c_cflag

1	1	0	1	0	1	1	0
1	?	?	?	?	?	?	?

Bit décrivant le status de a parité :
0 : pas de controle de parité
1: pas de controle de parité

toptions.c_cflag &= ~PARENB;
Qui s'écrit : toptions.c_cflag = toptions.c_cflag & ~PARENB;

Détail :

On applique un ET logique & bit a bit avec le registre

toptions	~PARENb							
115200	1	1	1	0	1	1	1	1
5 L 2	Valeurs des options du port série : toptions.c_cflag							
	1	1	0	1	0	1	1	0
↓	1	1	?	?	?	?	?	?
9600								
5 L 2								

Bit décrivant le status de a parité :
0 : pas de controle de parité
1: pas de controle de parité

toptions.c_cflag &= ~PARENb;
Qui s'écrit : toptions.c_cflag = toptions.c_cflag & ~PARENb;

Détail :

On applique un ET logique & bit a bit avec le registre

toptions	~PARENb							
115200	1	1	1	0	1	1	1	1
5 L 2	Valeurs des options du port série : toptions.c_cflag							
	1	1	0	1	0	1	1	0
↓	1	1	0	?	?	?	?	?
9600								
5 L 2								

Bit décrivant le status de a parité :
0 : pas de controle de parité
1: pas de controle de parité

toptions.c_cflag &= ~PARENb;
Qui s'écrit : toptions.c_cflag = toptions.c_cflag & ~PARENb;

Détail :

On applique un ET logique & bit a bit avec le registre

toptions	~PARENb							
115200	1	1	1	0	1	1	1	1
5 L 2	Valeurs des options du port série : toptions.c_cflag							
	1	1	0	1	0	1	1	0
↓	1	1	0	0	?	?	?	?
9600								
5 L 2								

Bit décrivant le status de a parité :
0 : pas de controle de parité
1: pas de controle de parité

toptions.c_cflag &= ~PARENb;
Qui s'écrit : toptions.c_cflag = toptions.c_cflag & ~PARENb;

Détail :

On applique un ET logique & bit a bit avec le registre

toptions	~PARENb							
115200	1	1	1	0	1	1	1	1
5 L 2	Valeurs des options du port série : toptions.c_cflag							
	1	1	0	1	0	1	1	0
↓	1	1	0	0	?	?	?	?
9600								
5 L 2								

Bit décrivant le status de a parité :
0 : pas de controle de parité
1: pas de controle de parité

toptions.c_cflag &= ~PARENb;
Qui s'écrit : toptions.c_cflag = toptions.c_cflag & ~PARENb;
C'est a dire, une recopie de toptions.c_cflag sauf le bit de parité qui est mis a 0

Détail :

Quand on fera le `tcsetattr`

toptions	~PARENB							
115200	1	1	1	0	1	1	1	1
5 L 2	Valeurs des options du port série : <code>toptions.c_cflag</code>							
	1	1	0	1	0	1	1	0
9600	1	1	0	0	?	?	?	?
5 N 2								

Bit décrivant le status de a parité :
 0 : pas de controle de parité
 1: pas de controle de parité

`toptions.c_cflag &= ~PARENB;`
 Qui s'écrit : `toptions.c_cflag = toptions.c_cflag & ~PARENB;`
 C'est a dire, une recopie de `toptions.c_cflag` sauf le bit de parité qui est mis a 0

Bits de stop :

CSTOPB

Utiliser deux bits de stop plutôt qu'un.

toptions	<code>toptions.c_cflag &= ~CSTOPB;</code>							
115200								
5 L 2								
9600								
5 N 1								

C'est à dire :
 Conserver tous les bits de toptions à l'identique sauf le bit de stop qui sera mis à la négation de CSTOPB
 Donc "pas CSTOPB"
 Donc pas 2 bits de stop
 Donc 1 bit de stop

Bits de stop :

CSIZE :Masque de longueur des caractères. Les valeurs sont **CS5**, **CS6**, **CS7** ou **CS8**

toptions	<code>toptions.c_cflag &= ~CSIZE;</code> <code>toptions.c_cflag = CS8;</code>							
115200								
5 L 2								
9600								
5 N 1								

Pas de masque de longueur,
Mots de 8 bits

Sachant que `|` est le ou inclusif, donnez la forme du masque CS8.

Bits de stop :

```
tcgetattr(sc->port, &toptions);
/* ----- configuratin en 9600 8N1----- */

cfsetispeed(&toptions, B9600);
cfsetospeed(&toptions, B9600);
toptions.c_cflag &= ~PARENB; // pas de bcontrol de parite , bit de parite a zero
toptions.c_cflag &= ~CSTOPB; // 1 stop bit
toptions.c_cflag &= ~CSIZE; // mots de 8 bits
toptions.c_cflag |= CS8; // mots de 8 bits

toptions.c_cflag &= ~CRTSCTS; // pas de controle de flux

tcsetattr(sc->port, TCSANOW, &toptions);
```