

Envoie de float

Fonctionnement de la simulation

Initialiser les 2 cartes (voir démarche dans le fichier initialisation_des_cartes.pdf)

Faire make all sur les deux cartes (dans la console en se plaçant dans le dossier correspondant avec la commande cd)

Lancer en premier le programme sur le PC2 (reception) et ensuite le programme sur le PC1 (le PC2 attend de recevoir quelque chose sur le CAN pour passer à la suite).

Description des fonctions

Sur le PC2 (reception)

Créer un char port contenant la valeur du port a ouvrir (canA ou canB)

Attendre la réception du nombre de floats que le PC1 va envoyer
int taille = (int)recvFloat(port);

Fonctionnement de recFloat(port) :

Ouvre le port canA ou canB (argument en entrée de la fonction)

Rentre dans la boucle while(running)

Attend que quelqu'un parle le bus can (avec ioctl =0 si on parle)

Dès qu'une trame est reçu sauvegarde la trame dans data avec ioctl

Convertie la trame en float

Ferme le can

Renvoie le float

Sur le PC1 (émission)

Lire le nombre de lignes du fichier contenant des nombres qui sont convertis en floats.

NbLignes("fichier_a_lire_PC1.txt")

Lire le fichier et stocker les valeurs dans un tableau de float : float *TABLEAU=new float[taille]

TABLEAU=LectureData("fichier_a_lire_PC1.txt");

Envoyer au PC2 le nombre de floats qui sera transmit

sendFloat(taille);

Sur le PC2 (reception)

Créer une boucle for de taille le nombre de floats à recevoir
float r_tab[taille];

```
for(int ind=0 ; ind<taille ; ind++){  
float r=recvFloat(port);  
r_tab[ind]=r;  
}
```

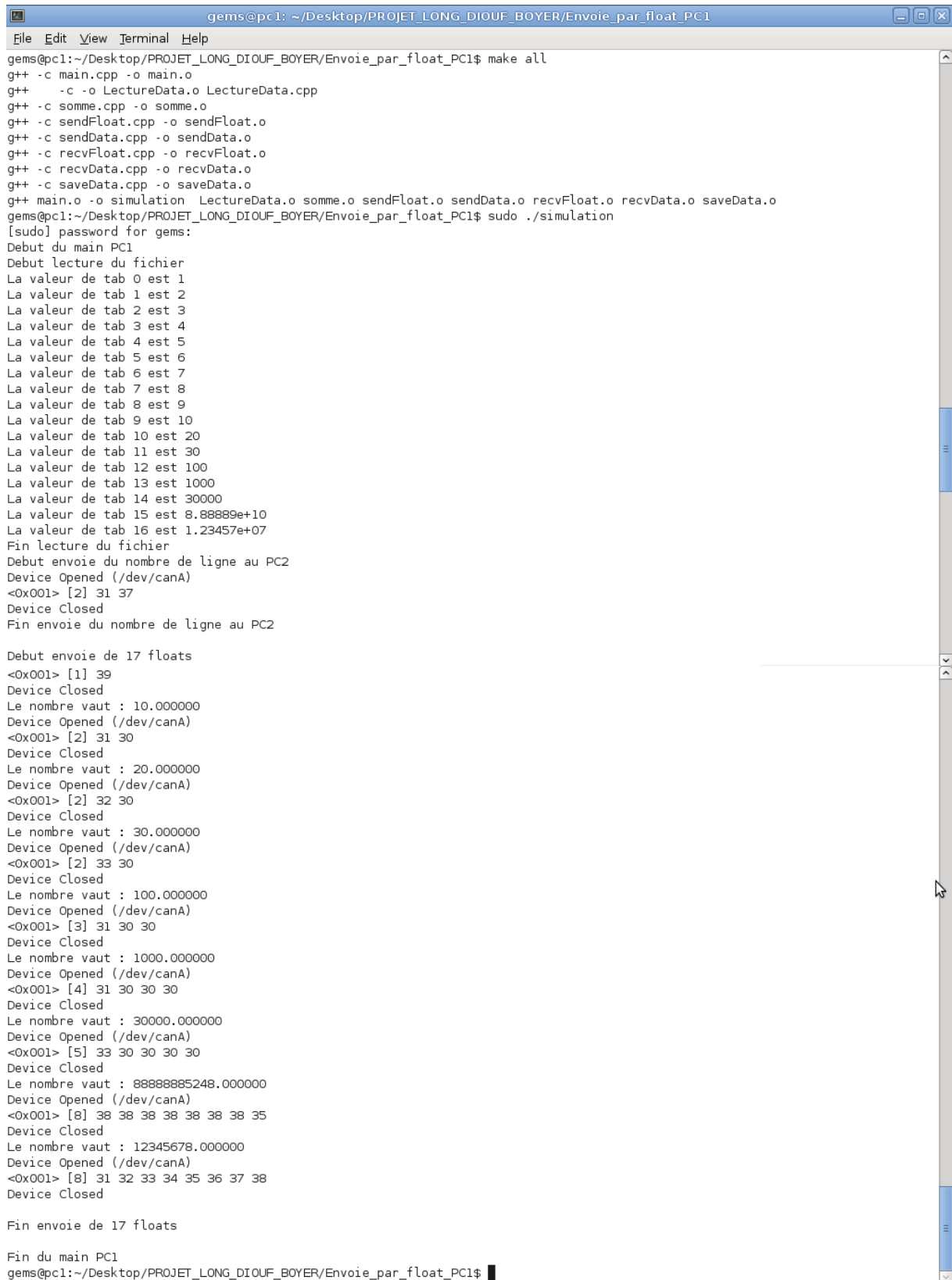
Sur le PC1 (émission)

Créer une boucle for de taille le nombre de floats à émettre

```
for(int ind=0 ; ind<taille ; ind++){  
sendFloat(TABLEAU[ind]);  
usleep(1000);  
}
```

On obtient dans la console :

sur le PC1



```
gems@pc1: ~/Desktop/PROJET_LONG_DIOUF_BOYER/Envoie_par_float_PC1
File Edit View Terminal Help
gems@pc1:~/Desktop/PROJET_LONG_DIOUF_BOYER/Envoie_par_float_PC1$ make all
g++ -c main.cpp -o main.o
g++ -c -o LectureData.o LectureData.cpp
g++ -c somme.cpp -o somme.o
g++ -c sendFloat.cpp -o sendFloat.o
g++ -c sendData.cpp -o sendData.o
g++ -c recvFloat.cpp -o recvFloat.o
g++ -c recvData.cpp -o recvData.o
g++ -c saveData.cpp -o saveData.o
g++ main.o -o simulation LectureData.o somme.o sendFloat.o sendData.o recvFloat.o recvData.o saveData.o
gems@pc1:~/Desktop/PROJET_LONG_DIOUF_BOYER/Envoie_par_float_PC1$ sudo ./simulation
[sudo] password for gems:
Debut du main PC1
Debut lecture du fichier
La valeur de tab 0 est 1
La valeur de tab 1 est 2
La valeur de tab 2 est 3
La valeur de tab 3 est 4
La valeur de tab 4 est 5
La valeur de tab 5 est 6
La valeur de tab 6 est 7
La valeur de tab 7 est 8
La valeur de tab 8 est 9
La valeur de tab 9 est 10
La valeur de tab 10 est 20
La valeur de tab 11 est 30
La valeur de tab 12 est 100
La valeur de tab 13 est 1000
La valeur de tab 14 est 30000
La valeur de tab 15 est 8.88889e+10
La valeur de tab 16 est 1.23457e+07
Fin lecture du fichier
Debut envoie du nombre de ligne au PC2
Device Opened (/dev/canA)
<0x001> [2] 31 37
Device Closed
Fin envoie du nombre de ligne au PC2

Debut envoie de 17 floats
<0x001> [1] 39
Device Closed
Le nombre vaut : 10.000000
Device Opened (/dev/canA)
<0x001> [2] 31 30
Device Closed
Le nombre vaut : 20.000000
Device Opened (/dev/canA)
<0x001> [2] 32 30
Device Closed
Le nombre vaut : 30.000000
Device Opened (/dev/canA)
<0x001> [2] 33 30
Device Closed
Le nombre vaut : 100.000000
Device Opened (/dev/canA)
<0x001> [3] 31 30 30
Device Closed
Le nombre vaut : 1000.000000
Device Opened (/dev/canA)
<0x001> [4] 31 30 30 30
Device Closed
Le nombre vaut : 30000.000000
Device Opened (/dev/canA)
<0x001> [5] 33 30 30 30 30
Device Closed
Le nombre vaut : 8888885248.000000
Device Opened (/dev/canA)
<0x001> [8] 38 38 38 38 38 38 38 35
Device Closed
Le nombre vaut : 12345678.000000
Device Opened (/dev/canA)
<0x001> [8] 31 32 33 34 35 36 37 38
Device Closed

Fin envoie de 17 floats

Fin du main PC1
gems@pc1:~/Desktop/PROJET_LONG_DIOUF_BOYER/Envoie_par_float_PC1$
```

sur le PC2

```
Applications Places System Thu Mar 5, 3:26 PM
gems@pc2: ~/Desktop/PROJET LONG DIOUF BOYER 2015/Envoie_par_float_PC2
File Edit View Terminal Help
gems@pc2:~/Desktop/PROJET LONG DIOUF BOYER 2015/Envoie_par_float_PC2$ make all
g++ -c main.cpp -o main.o
g++ -c -o LectureData.o LectureData.cpp
g++ -c somme.cpp -o somme.o
g++ -c sendFloat.cpp -o sendFloat.o
g++ -c sendData.cpp -o sendData.o
g++ -c recvFloat.cpp -o recvFloat.o
g++ -c recvData.cpp -o recvData.o
g++ -c saveData.cpp -o saveData.o
g++ main.o -o simulation LectureData.o somme.o sendFloat.o sendData.o recvFloat.o recvData.o saveData.o
gems@pc2:~/Desktop/PROJET LONG DIOUF BOYER 2015/Envoie_par_float_PC2$ sudo ./simulation
[sudo] password for gems:
Debut du main PC1
Debut reception du nombre de lignes
appDeviceName_rcv : /dev/canA
Device Opened (/dev/canA)
<0x001> [2] 31 37
Device Closed
la chaine vaut : 17
Fin reception du nombre de lignes
Debut reception de 17 floats
appDeviceName_rcv : /dev/canA
Device Opened (/dev/canA)
<0x001> [1] 31
Device Closed
la chaine vaut : 1
appDeviceName_rcv : /dev/canA
Device Opened (/dev/canA)
<0x001> [1] 32
Device Closed
la chaine vaut : 2
appDeviceName_rcv : /dev/canA
Device Opened (/dev/canA)
<0x001> [1] 33
Device Closed
la chaine vaut : 3
appDeviceName_rcv : /dev/canA
Device Opened (/dev/canA)
<0x001> [1] 34
<0x001> [4] 31 30 30 30
Device Closed
la chaine vaut : 1000
appDeviceName_rcv : /dev/canA
Device Opened (/dev/canA)
<0x001> [5] 33 30 30 30 30
Device Closed
la chaine vaut : 30000
appDeviceName_rcv : /dev/canA
Device Opened (/dev/canA)
<0x001> [8] 38 38 38 38 38 38 38 35
Device Closed
la chaine vaut : 88888885
appDeviceName_rcv : /dev/canA
Device Opened (/dev/canA)
<0x001> [8] 31 32 33 34 35 36 37 38
Device Closed
la chaine vaut : 12345678
Fin reception de 17 floats
Debut du print du tableau
Le float 0 vaut avec printf : 1.000000
Le float 1 vaut avec printf : 2.000000
Le float 2 vaut avec printf : 3.000000
Le float 3 vaut avec printf : 4.000000
Le float 4 vaut avec printf : 5.000000
Le float 5 vaut avec printf : 6.000000
Le float 6 vaut avec printf : 7.000000
Le float 7 vaut avec printf : 8.000000
Le float 8 vaut avec printf : 9.000000
Le float 9 vaut avec printf : 10.000000
Le float 10 vaut avec printf : 20.000000
Le float 11 vaut avec printf : 30.000000
Le float 12 vaut avec printf : 100.000000
Le float 13 vaut avec printf : 1000.000000
Le float 14 vaut avec printf : 30000.000000
Le float 15 vaut avec printf : 88888888.000000
Le float 16 vaut avec printf : 12345678.000000
Fin du main PC1
gems@pc2:~/Desktop/PROJET LONG DIOUF BOYER 2015/Envoie_par_float_PC2$
```