

PROJET ROBOT PONG

ABDUL HAKIM Panna



Avec BARETTE Alexandra & ALAOUA Othmane

Lycée Jean Rostand 2017-2018 – BTS SNIR 2

Sommaires

- ❖ Introduction
- ❖ Schéma
- ❖ Présentation des parties
- ❖ Diagramme de séquence
- ❖ Ma Partie
- ❖ Programmation
- ❖ Conclusion



Introduction

- ❖ Le Robot Pong 1040 est composé d'une table de Ping Pong et du robot situé en face du joueur, elle est télécommandé depuis une manette de contrôle, elle a comme particularité de changer la vitesse d'envoi des balles et la fréquence oscillations d'envoi des zones.
- ❖ Notre Objectif est d'améliorer son système télécommander, rendre plus dynamique pour le joueur, ajouter comme fonctionnalité de consulter des statistiques avoir un compte (abonnement), historique détailler de chaque match pour étudier ces évolution et avoir le contrôle du robot.
- ❖ Nous avons décidé d'équiper le robot, Capteur de Force afin de déterminer la zone joué pour son entraînement, Capteur IR détection sortie de balle, un Cerveau Moteur et un Moteur.
- ❖ Remplacer complètement la télécommande par une carte Arduino.
- ❖ Récolter des données comme le score, qui seront stocker dans une BDD(Raspberry), et consultables via une Application.



Visuelle

Avant



Télécommande

Relier Par une carte Arduino

Raspberry

Application Android

Capteur IR

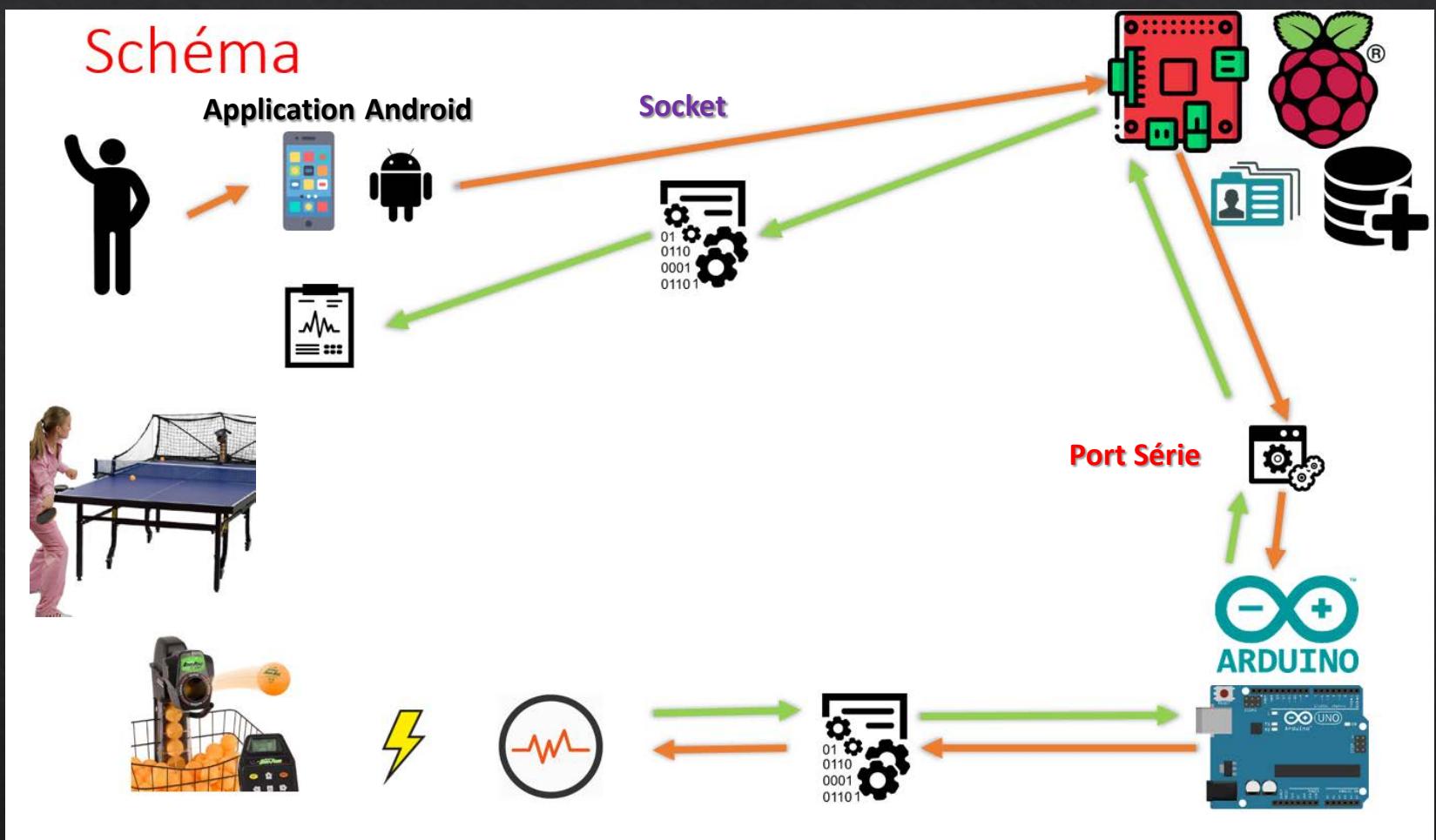
Cerveau Moteur

Apres



Capteur de Force

Moteur



Ici l'utilisateur se connecte dans la base de données qui est contenue dans la Raspberry, puis la Raspberry met à jour la base données avec les paramètres envoyé depuis l'application et en parallèle il envoi vers la carte Arduino via un port série, et la carte Arduino traite les données de la Raspberry et s'engage, lors de la partie l'Arduino renvoi le score final que celui-ci sera envoyé à la carte Raspberry et mettre à jour dans la base données puis l'utilisateur pourra consulté dans l'application.

Présentation des parties

- ❖ Application Android
- ❖ -Interfaces login et MDP
- ❖ Connexion vers le serveur centrale via une socket.
- ❖ -création d'un compte, modification ou supprimer.
- ❖ Ajout des paramètre, vitesse, fréquence, zone, nombre de balles.
- ❖ Commander le robot en temps réel.
- ❖ Affichages des statistiques.



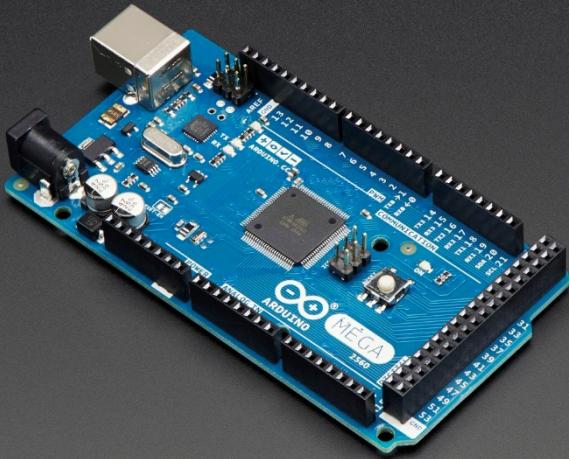
The figure shows two screenshots of the Android application. The left screenshot shows the main control screen with 'Control Mode: Precision'. It has fields for 'Delay (Seconds)' (0) and 'Duration (Minutes)' (5), and buttons for 'START', 'SAVE', 'LOAD', 'BACK', and 'CLEAR'. The right screenshot shows a detailed configuration screen for 'Position #1' with various parameters: '# Balls' (1), 'Wait Time' (1000), 'Top Motor' (75), 'Bottom Motor' (0), 'Oscillation' (50), 'Height' (50), and 'Rotation' (0). Each parameter has a '+' and '-' button to adjust its value.

Socket

BDD & serveur TCP/IP



- ❖ Arduino(Commande les capteurs) :
- ❖ Commander ces cerveaux moteur, comptage de balles capteur IR, capteur de détection des rebonds.
- ❖ Application des donnée : vitesse, fréquence, zone, nombre de balles.
- ❖ communique avec la Raspberry en port série et renvoi le score finale.



Série

BDD & serveur TCP/IP



Raspberry pi(mini ordinateur)



- ❖ Serveur TCP/IP communique avec l'application.
- ❖ Elaborer une structure base de données.
- ❖ Récupérer des données paramètre de l'application enregistrer dans la base de données puis envoyer à l'Arduino.



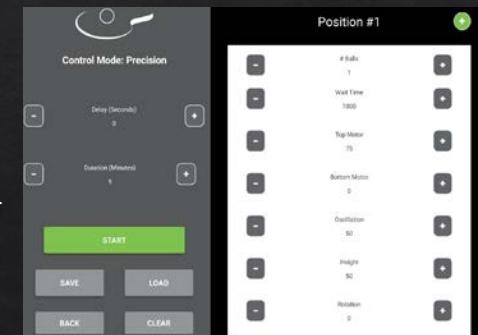
Série

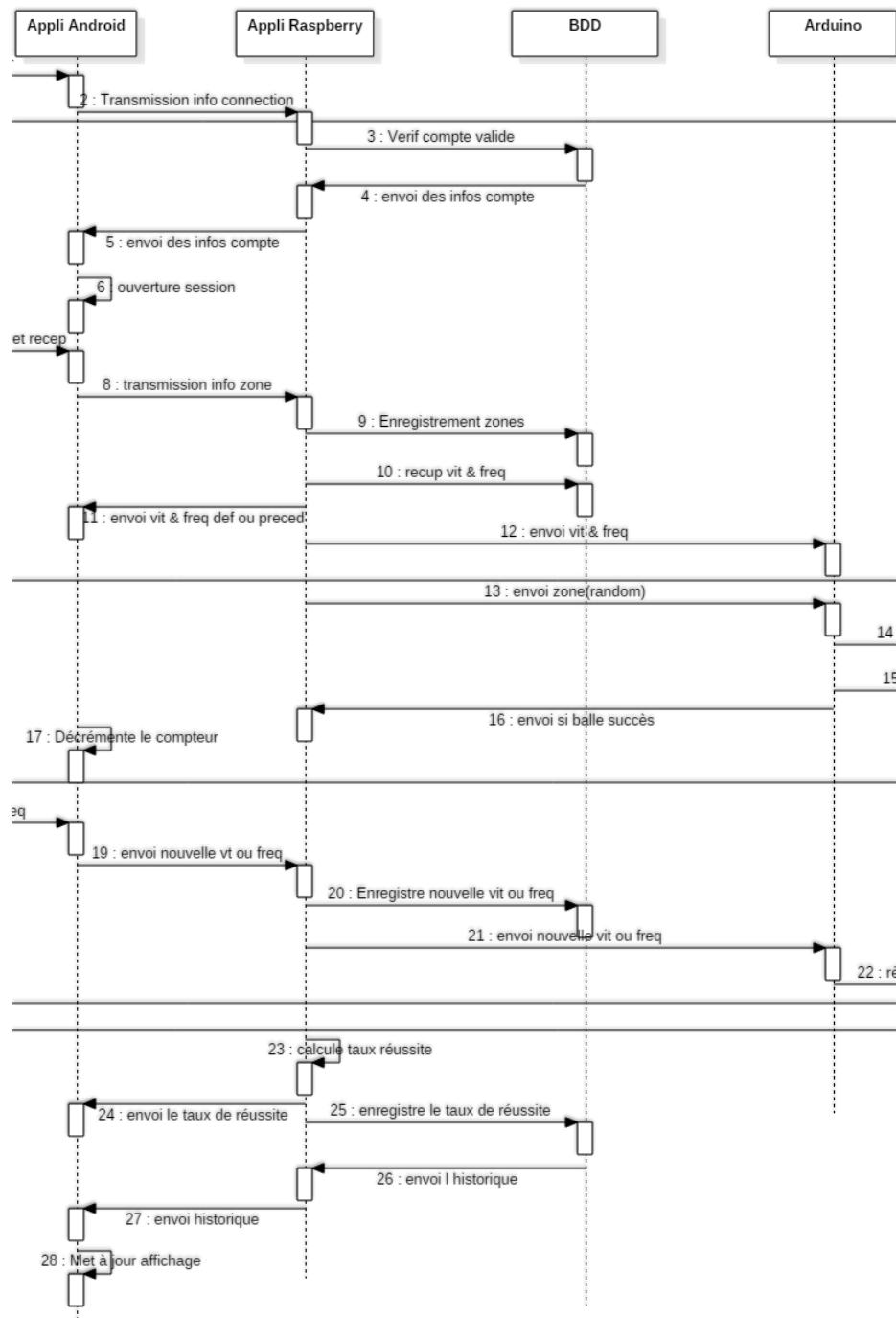


Socket

BDD & serveur TCP/IP

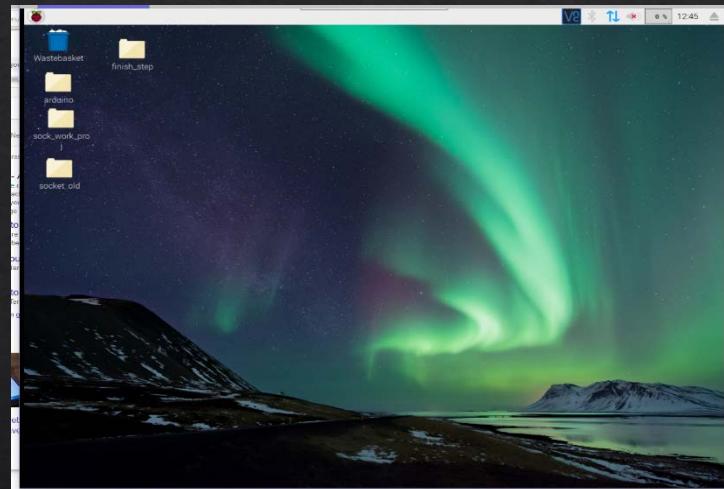
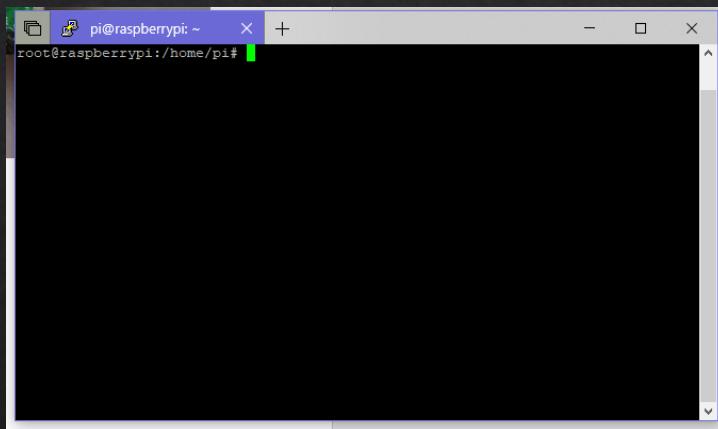
Application Android





Ma partie

- ❖ Installation de l'os Raspbian
- ❖ Installation de MySQL serveur
- ❖ Création d'utilisateur, crée la base donnée puis les tables et lui donner des privilèges et rendre les connexion extérieur possible.
- ❖ Connexion via PuTTY en cmd sinon bureau :



connexion MySQL

```
root@raspberrypi:/etc/mysql# mysql -u user -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

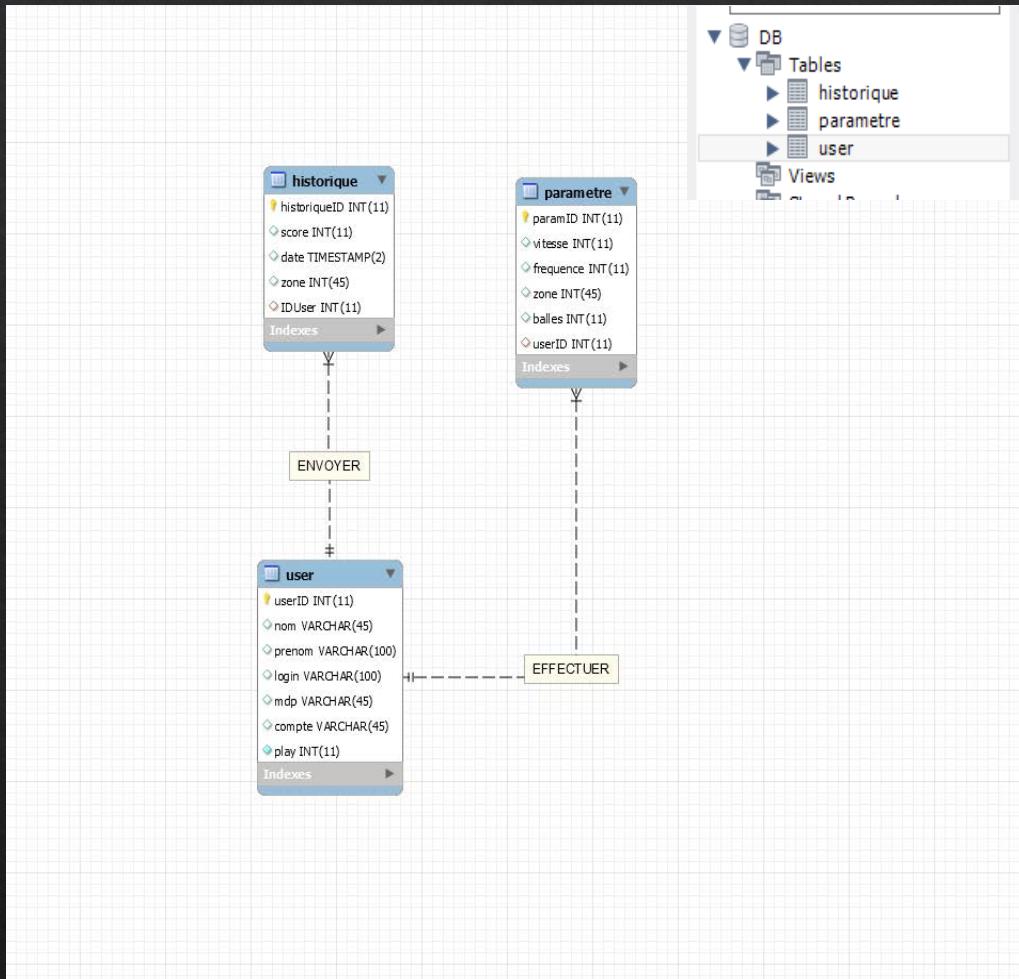
MariaDB [(none)]> use db;
ERROR 1049 (42000): Unknown database 'db'
MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| DB           |
| information_schema |
| mysql         |
| performance_schema |
| phpmyadmin    |
+-----+
5 rows in set (0.00 sec)

MariaDB [(none)]> use DB;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [DB]> show tables;
+-----+
| Tables_in_DB |
+-----+
| historique   |
| parametre    |
| user          |
+-----+
3 rows in set (0.00 sec)

MariaDB [DB]> █
```

Structure MySQL



userID	nom	prenom	login	mdp	compte	play
0	mohammed	Yousef	MY	Mvoussef	user	0
1	david	ib	dio	123456	user	0
2	Valerie	Edouard	Vedouard	edouard123	user	0
3		Alex	Alex	Alex123	admin	0
4		Othmane	Othmane	Othmane123	admin	0
5	NULL	Panna	Panna	Panna123	admin	0
6	Entraineur	Admin	admin321	admin123	admin	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

paramID	vitesse	frequence	zone	balles	userID
1	3	7	10111000	10	1
2	10	2	10111011	12	0
4	5	5	10101010	10	3
5	1	1	11100010	10	4
6	9	4	10111011	10	5
9	2	8	10111011	10	2
NULL	NULL	NULL	NULL	NULL	NULL

historiqueID	score	date	zone	IDUser
1	39	2018-06-05 08:17:24.59	10111011	0
2	86	2018-06-16 09:14:49.38	10111011	0
3	20	2018-05-18 16:30:00.00	10110011	1
4	65	2018-06-05 13:15:21.70	10111001	0
5	56	2018-06-09 05:46:00.53	11110001	1
NULL	NULL	NULL	NULL	NULL

Par exemple pour avoir le score de l'utilisateur Edouard on devra faire une requête :
 Select score from historique where historique.IDUser=user.userID and nom=«Edouard»;

Programmation :

❖ Socket : donner reçu de l'application Android

Donnée reçu depuis l'application Android :
u.1.z.11111010.v.100.f.100.play
U= Numéro de l'utilisateur
Z=les zones (entrée : 1111 et réception 1010)
V=vitesse d'envoi des balles
F= fréquence pour l'oscillation

```
prydaz@prydaz: ~/Downloads/threads_server$ ./server
[+]Socket est Crée
[+]Creation Bind par succes
[+]En Attente d'une connexion client...
[+]Connexion accepter de 127.0.0.1:42902
[+]Thread assigné
u.1.z.11111010.v.100.f.100.play
data recu : id : [1] zoneER : [e1111r1010] vit : [100] freq : [100]
requete OK
Erreur d'ouverture du port serie: No such file or directory
data recu : id : [1] zoneER : [e1111r1010] vit : [100] freq : [100]
[+]my PID : 2146
f.232
data modif frequence : id : [1] zoneER : [e1111r1010] vit : [100] freq : [232]
requete Frequence OK
Erreur d'ouverture du port serie: No such file or directory
data recu : id : [1] zoneER : [e1111r1010] vit : [100] freq : [232]
[+]mv PID : 2146
v.400
data modif vitesse : id : [1] zoneER : [e1111r1010] vit : [400] freq : [232]
requete vitesse OK
Erreur d'ouverture du port serie: No such file or directory
data recu : id : [1] zoneER : [e1111r1010] vit : [400] freq : [232]
[+]my PID : 2146
```

- Première connexion reçoit id_user.zone l'id (db. User) utilisateur et la zone (db.parametre)
- Deuxième Connexion on reçoit comme paramètre la vitesse(parametre.vitesse) et la fréquence(parametre.frequence).
- Troisième tentative de modification des données fréquence et la vitesse.

socket

```
printf("Connexion accepter de %s:%d\n", inet_ntoa(newAddr.sin_addr), ntohs(newAddr.sin_port));
user++; //pour chaque connection
if((childpid = fork()) == 0){
    close(sockfd);
    int i = 0;
    while(1){
        recv(newSocket, buffer, BUFFER_SIZE, 0);

        if(strcmp(buffer, "q") == 0){
            printf("Deconnexion de %s:%d\n", inet_ntoa(newAddr.sin_addr), ntohs(newAddr.sin_port));
            break;
        }else{
            printf("%s\n", buffer);//Client recu;
            send(newSocket, buffer, strlen(buffer), 0);
            bzero(buffer, sizeof(buffer));

            db[i]=buffer[i];//copie du buffer au db
            printf("%c",db[i]);

            i++;
            sleep(1);
        }
    }
}
```

Connexion vers le serveur, j'utilise fork() pour connexions multiples via le socket TCP/IP,
Puis la recv() permet de recevoir des données (BUFFER) à partir du TCP, et j'affiche les données envoyé

Connexion en série

Initialisation du port série:

```
port_serie = open("/dev/ttyACM0", O_RDWR | O_NOCTTY | O_NDELAY); //lecture et écriture | pas controlling terminal | ne pas attendre DCD
```

Programme de lecture et écriture via la port série tourne en boucle pour recevoir et envoi des données vers l'Arduino via le port série.

```
while(!stop){
    cout << "Donnees a envoyer : ";
    cin.getline(data, 20);

    if(strcmp(data, ":exit") == 0) {
        stop = true;
    }
    else {      //écriture
        if (write(port_serie, data, sizeof(data)) < 0){perror("ERROR\n");}
        cout << "En attente de la reponse... " << endl;
        sleep(1);
        //lecture
        fcntl(port_serie,F_SETFL,10); //mode bloquant pour la fonction read
        reception=read(port_serie,buffer,20); //buffer = donnees; 20 = nb octets
        cout << "Donnees recues : " << buffer << endl;

        if (reception == -1)
            {printf("Erreur lecture port serie\n");}
    }
    cout << "Deconnexion du port" << endl;
}

close(port_serie); //fermeture du port serie
```

Connexion BDD

Connexion de vers la bases données on récupère les paramètre on applique dans mysql_query() puis on la fetch() vers la BDD.

```
void sql (){  
    MySQL *conn;  
    MySQL_RES *res;  
    MySQL_ROW row;  
  
    char *server = "192.168.0.182";  
    char *user = "user";  
    char *password = "rootroot"; /* set me first */  
    char *database = "DB";  
  
    conn = mysql_init(NULL);  
  
    /* Connect to database */  
    if (!mysql_real_connect(conn, server,  
                           user, password, database, 0, NULL, 0)) {  
        fprintf(stderr, "%s\n", mysql_error(conn));  
        exit(1);  
    }  
  
    /* send SQL query */  
    if (mysql_query(conn, "Select * from raspberry;")) { //show tables  
        fprintf(stderr, "%s\n", mysql_error(conn));  
        exit(1);  
    }  
  
    res = mysql_use_result(conn);  
  
    /* output table name */  
  
    printf("Les Tables de MySQL :\n");  
    while ((row = mysql_fetch_row(res)) != NULL)  
        printf("[+] [%s] [%s] [%s]\n", row[0],row[1],row[2]);  
  
    /* close connection */  
    mysql_free_result(res);  
    mysql_close(conn);  
}
```

conclusion

- ❖ Apres avoir travailler sur ce projet robot Pong et intégration de toutes parties et des modifications effectués, on a mis en place une application Android d'où l'utilisateur pourra contrôlé depuis cette application et tout ces informations émise vers le serveur centrale d'où celui-ci stock dans une BDD et qui ordonne au robot gérer par l'Arduino. Plus tard fin de partie l'utilisateur pourra consulter ses statistiques relevé lors de la parties. On en déduire qu'elle est très efficaces lancer une partie et de joué .

- ❖ Remerciement au professeur d'informatique et physique pour l'avancement de ce projet et de leurs aides fournis.