

## SEANCE 8

**Objectif :** L'objectif de cette séance était de finaliser la réception de données par Bluetooth et l'écriture réalisée par le robot.

**TRAVAIL PRELIMINAIRE :** Durant les vacances, j'ai essayé d'avancer au maximum. La tâche principale était de faire en sorte que le bras écrive dans un premier temps la lettre qu'il reçoit via une application Bluetooth.

**Etape 1 :** Créer mes fonctions de base à partir du code initial. Un travail assez simple finalement. J'ai mis le code initial dans une fonction qui prend en paramètre le point où je veux aller :

```
void goTo(int x1,int y1){
    x=x1;
    y=y1;
    h=sqrt(sq(x)+sq(y));
    Alpha=(acos( (sq(h) + sq(1e) - sq(1c)) / (2*1e*h) )) * (180/pi);
    tetaC= (pi - (acos ( (sq(1e) + sq(1c) - sq(h)) / (2*1e*1c) ))) * (180/pi);
    tetaE= acos( x / h) * (180/pi) - Alpha;

    tetaE_micro= 9.2339*tetaE+1100.3;
    tetaC_micro= -10.023*tetaC+2155.8;
    epaule.writeMicroseconds(tetaE_micro);
    coude.writeMicroseconds(tetaC_micro);
    delay(10);
}
```

Pour tracer des traits verticaux et horizontaux, je n'ai donc plus qu'à appeler ces fonctions :

```
void ligneH(int x1, int y1){
    goTo(x1,y1);
    delay(500);
    setDown();
    delay(1000);
    for(int i=0; i<100; i++){
        x=x1;
        y=0.02*i + y1;

        h=sqrt(sq(x)+sq(y));
        Alpha=(acos( (sq(h) + sq(1e) - sq(1c)) / (2*1e*h) )) * (180/pi);
        tetaC= (pi - (acos ( (sq(1e) + sq(1c) - sq(h)) / (2*1e*1c) ))) * (180/pi);
        tetaE= acos( x / h) * (180/pi) - Alpha;

        tetaE_micro= 9.2339*tetaE+1100.3;
        tetaC_micro= -10.023*tetaC+2155.8;
        epaule.writeMicroseconds(tetaE_micro);
        coude.writeMicroseconds(tetaC_micro);
        delay(10);
    }
}
```

```
void ligneV(int x1, int y1){
    goTo(x1,y1);
    delay(500);
    setDown();
    delay(1000);
    for(int i=0; i<100; i++){
        x=(0.03*i)+x1;
        y=y1;

        h=sqrt(sq(x)+sq(y));
        Alpha=(acos( (sq(h) + sq(1e) - sq(1c)) / (2*1e*h) )) * (180/pi);
        tetaC= (pi - (acos ( (sq(1e) + sq(1c) - sq(h)) / (2*1e*1c) ))) * (180/pi);
        tetaE= acos( x / h) * (180/pi) - Alpha;

        tetaE_micro= 9.2339*tetaE+1100.3;
        tetaC_micro= -10.023*tetaC+2155.8;
        epaule.writeMicroseconds(tetaE_micro);
        coude.writeMicroseconds(tetaC_micro);

        delay(10);
    }
}
```

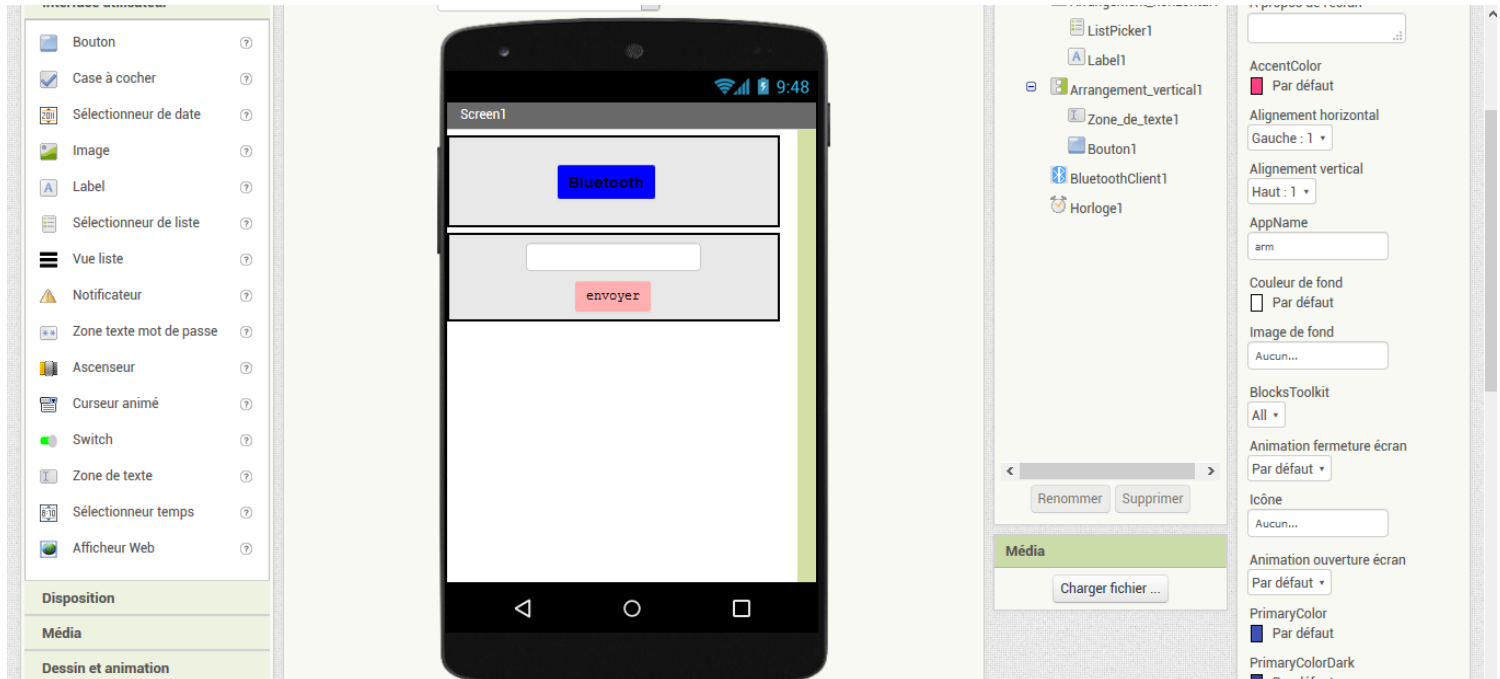
De même, pour tracer des traits obliques :

```
void ligneObliqueAS(int x1, int y1){ //permet de tracer un anti-slash
    for(int i=0; i<100; i++){
        x=0.03*i + x1;
        y=0.02*i + y1;

        h=sqrt(sq(x)+sq(y));
        Alpha=(acos( (sq(h) + sq(1e) - sq(1c)) / (2*1e*h) )) * (180/pi);
        tetaC= (pi - (acos ( (sq(1e) + sq(1c) - sq(h)) / (2*1e*1c) ))) * (180/pi);
        tetaE= acos( x / h) * (180/pi) - Alpha;
```

Cette fonction me permet de tracer un \ mais lorsque je programme celle qui me permet de tracer un /, le bras ne trace pas du tout un trait droit. Pour cela il vaut donc mieux que j'utilise la fonction goTo.

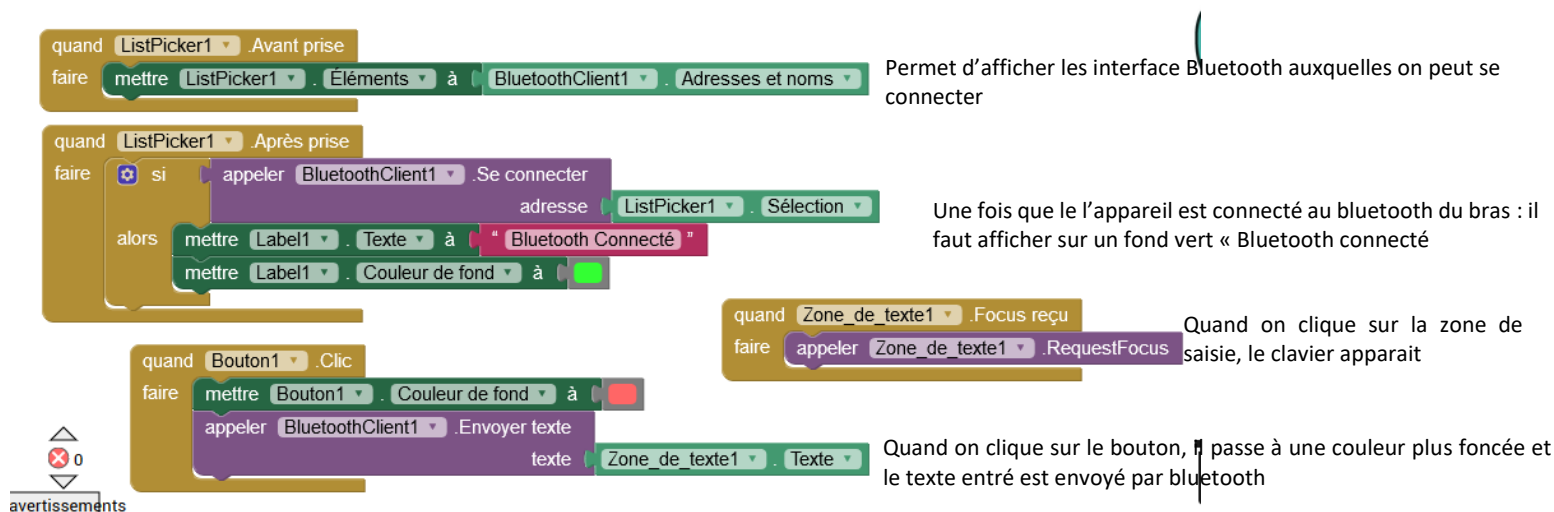
Etape 2 : Créer une application sur MIT app inventor. Camille avait déjà essayé de créer une application au début du projet mais elle n'était pas sûre de son bon fonctionnement. De plus, elle était en vacances et je n'arrivais pas à me connecter avec son adresse mail pour avoir accès à son projet. Il a donc fallu que je regarde plusieurs tutos pour apprendre à utiliser ce « logiciel ». En m'inspirant de ce qu'elle avait fait, j'ai créé cette application :



L'interface se présente comme ceci. Lorsque l'utilisateur appuie sur le bouton bluetooth, il peut choisir notre module bluetooth nommé « arm ». Une fois qu'il clique dessus et que son téléphone se connecte, un message « bluetooth connecté » apparaît à l'écran.

Ensuite, il clique sur la barre de saisie, entre la lettre qu'il veut écrire et clique sur le bouton « envoyer ».

Comment l'application fonctionne ? J'ai dû définir des « blocs » qui correspondent à différentes actions à effectuer :



```

//--MODULE BLUETOOTH
const int RX=3;
const int TX=11;
char Data;
SoftwareSerial BlueT(RX,TX);

void setup() {
    epaule.attach(9,0,180);
    coude.attach(10,0,180);
    poignet.attach(8,0,180);
    Serial.begin(9600);
    Serial.println("Bonjour -Pret pour les commandes AT");
    BlueT.begin(9600);
    delay(500);
}

void loop() {
    raiseP();
    while (BlueT.available()) {
        Data=BlueT.read();
        Serial.print(Data);
    }
    while (Serial.available()) {
        BlueT.write(char(Serial.read()));
    }
}

```

Etape 3 : J'ai ensuite commencé à créer un début d'alphabet :

<pre> void L(int x, int y){     goTo(x,y);     delay(1000);     setDown();     ligneV(x,y);     delay(1000);     ligneH(x+3,y);     delay(1000);     raiseP();     delay(100); } </pre>	<pre> void f(int x, int y){     raiseP();     goTo(x,y+2);     setDown();     delay(1000);     ligneV(x,y);     delay(1000);     ligneH(x+1,y);     delay(1000); } </pre>	<pre> void e(int x, int y){     raiseP();     goTo(x,y+2);     setDown();     delay(1000);     ligneV(x,y);     delay(1000);     ligneH(x+3,y);     delay(1000);     raiseP();     delay(1000);     ligneH(x+1,y);     delay(1000);     raiseP();     delay(100); } </pre>	<pre> void H(int x,int y){     ligneV(x,y);     delay(1000);     ligneV(x,y+2);     delay(1000);     ligneH(x+1,y);     delay(1000); } </pre>
<pre> void I(int x, int y){     ligneV(x,y);     delay(1000); } </pre>	<pre> void A(int x,int y){     goTo(x,y);     delay(1000);     goTo(x+3,y);     delay(1000);     setDown();     goTo(x,y+1);     delay(1000);     goTo(x+3,y+2);     delay(1000);     raiseP();     delay(1000);     ligneH(x+1,y);     delay(1000);     raiseP();     delay(100); } </pre>	<pre> void U(int x,int y){     goTo(x,y);     delay(1000);     setDown();     ligneV(x,y);     delay(1000);     ligneH(x+3,y);     delay(1000);     ligneV(x,y+2);     delay(1000);     raiseP();     delay(100); } </pre>	<pre> void N(int x, int y){     goTo(x,y);     delay(1000);     setDown();     ligneV(x,y);     delay(1000);     ligneObliqueAS(x,y);     delay(1000);     goTo(x,y+2);     delay(1000);     raiseP();     delay(100); } </pre>

Puis j'ai commencé à tester en envoyant qu'une seule lettre.

J'ai d'ailleurs également créé les fonctions qui me permettent de lever et baisser le stylo :

```

void raiseP(){
    poignet.writeMicroseconds(1500);
    delay(500);
}

void setDown(){
    poignet.writeMicroseconds(1700);
    delay(500);
}

```

**Problèmes :** → je perdais la connexion avec le module Bluetooth alors que je souhaiter envoyer une autre lettre.

→ le bras a des problèmes pour se repérer : il va placer un même point à deux endroits proches mais non identiques lorsqu'il doit effectuer plusieurs actions. Lorsque je fais par exemple : ligneH(x,y) puis ligneV(x,y), bien que les paramètres passés aux deux fonctions soient identiques, le bras ne trace pas la ligne vertical pile au debut du trait horizontal. C'est pour cela que pour tracer un E par exemple je ne fais pas : ligneH(x,y) puis ligneV(x,y) mais je le fais aller à (x,y+2) pour ensuite abaisser le stylo et revenir en (x,y) ce qui me permet de tracer le même trait mais en faisant en sorte que les deux lignes se touchent.

## **SEANCE :**

Etape 1 : Durant la première partie de la séance, Camille souhaitait que je lui explique le travail réalisé durant son absence. Je lui ai donc présenté l'ensemble du travail que j'avais réalisé afin qu'elle comprenne le code.

Etape 2 : Ensuite, nous avons réglé le problème de la déconnection du Bluetooth. Vous nous avez rappelé qu'il était nécessaire de placer un condensateur afin d'éviter ce problème. Nous avons d'abord observé comment le bras se comportait lorsqu'il recevait plusieurs lettres à la fois et il semblerait qu'il ne trace que la dernière reçue. Il faut donc pour écrire un mot, envoyer lettre par lettre.

Etape 3 : Il a fallu réfléchir à la manière dont on voulait écrire suite aux différentes contraintes que le bras nous impose. En effet, le bras est censé atteindre tous les points de coordonnées : x appartenant à [-7,+7] et y appartenant à [3, 14]. En réalité le bras lorsqu'il trace atteint les points de coordonnées : x appartenant à [-5,+5] environ et y appartenant à [7, 11] ce qui nous laisse une faible marge de manœuvre sachant que nos lettres sont assez grosses. On a donc choisi d'écrire les lettres les unes sous les autres.

Etape 4 : Définir de nouvelles lettres :

```
void C(int x,int y){
    goTo(x,y+2);
    delay(1000);
    setDown();
    delay(1000);
    goTo(x,y);
    delay(1000);
    ligneV(x,y);
    delay(1000);
    ligneH(x+3,y);
    delay(1000);
}

void X(int x, int y){
    goTo(x,y);
    delay(1000);
    setDown();
    ligneObliqueAS(x,y);
    delay(1000);
    raiseP();
    goTo(x+1,y+2);
    delay(1000);
    setDown();
    delay(1000);
    goTo(x+2,y);
    delay(1000);
    raiseP();
    delay(500);
}

void M(int x, int y){
    goTo(x,y);
    delay(1000);
    setDown();
    ligneV(x,y);
    delay(1000);
    ligneObliqueAS(x,y);
    delay(1000);
    goTo(x,y+3);
    delay(1000);
    ligneV(x,y+3);
    delay(1000);
    raiseP();
    delay(100);
}
```

Etape 5 : Faire en sorte que le stylo se déplace tout seul vers le bas entre chaque lettre : pour cela, on a déclaré des variables x2 et y2 d'abord égale aux coordonnées x,y du point de départ (i.e le point ou le stylo tracera la première lettre).

```
int x2=-5;
int y2=8 ;

if (Data=='v'){
    V(x2,y2);
    delay(1000);
    x2+=4;
    y2-=1;
}
if (Data=='e'){
    e(x2,y2);
    delay(1000);
    x2+=4;
    y2-=1;
}

if (Data=='n'){
    N(x2,y2);
    delay(1000);
    x2+=4;
    y2-=1;
}
if (Data=='m'){
    M(x2,y2);
    delay(1000);
    x2+=4;
    y2-=1;
}
```

Puis on fait en sorte que le x s'incrmente de 4 a chaque fois.

Pourquoi décrémenter le y de 1 ? parce que étrangement le stylo se décale vers la droite à chaque fois et n'écrit pas exactement dessous sinon.

**Problèmes :** → les traits horizontaux ne sont pas exactement droits

→ Quelques incohérences de placement font que les lettres ne sont pas très belles

→ Le fait de lever le stylo ou le baisser trace des traits que l'on ne souhaite pas.  
En effet le stylo est totalement droit lorsque l'on écrit ce qui fait que le soulever un peu trace forcément un trait.