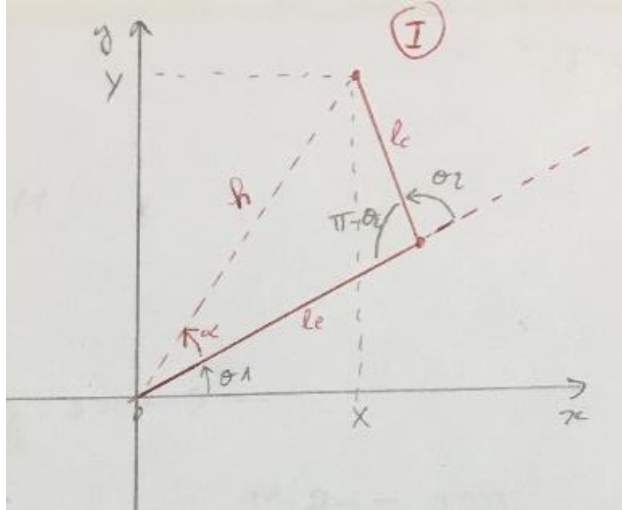


SEANCE 7

Objectif : L'objectif de cette séance était de comprendre ce travail que vous avez fait et de savoir l'utiliser. Par la suite, il fallait commencer à traiter le module Bluetooth.

TRAVAIL PRELIMINAIRE : Compte rendu du travail que vous avez effectué.



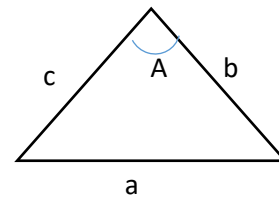
(1) Pythagore $h = \sqrt{x^2 + y^2}$

(2) Al Kashi : $a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos(A)$

Avec (2) on obtient $h^2 = le^2 + lc^2 - 2 \cdot lc \cdot le \cdot \cos(\pi - \theta_2)$

$$\Leftrightarrow \cos(\pi - \theta_2) = \frac{h^2 - le^2 - lc^2}{-2 \cdot lc \cdot le}$$

$$\Leftrightarrow \theta_2 = \text{Teta}_c = \pi - \cos^{-1}\left(\frac{-h^2 + le^2 + lc^2}{2 \cdot lc \cdot le}\right)$$



Le cercle trigo permet d'écrire :

$$\theta_1 = \cos^{-1}\left(\frac{y}{h}\right) - \alpha \quad (4)$$

On obtient α avec (2) :

$$lc^2 = h^2 + le^2 - 2 \cdot h \cdot le \cdot \cos \alpha$$

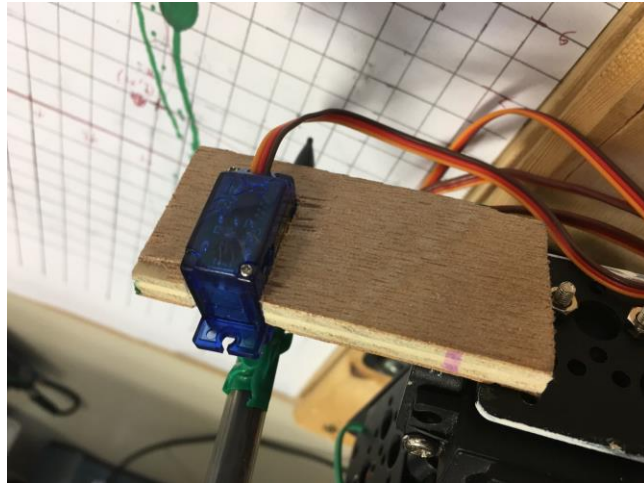
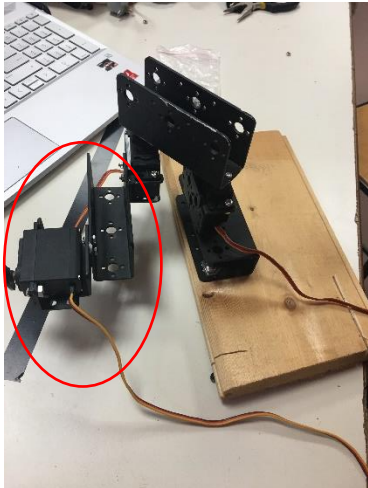
$$\Leftrightarrow \alpha = \cos^{-1}\left(\frac{h^2 + le^2 - lc^2}{2 \cdot h \cdot le}\right) \quad (5)$$

avec (5) et (4) on obtient :

$$\theta_1 = \text{Teta}_e = \cos^{-1}\left(\frac{y}{h}\right) - \cos^{-1}\left(\frac{h^2 + le^2 - lc^2}{2 \cdot h \cdot le}\right)$$

Etape 1 : Changer la forme de notre coude.

Vous nous avez conseillé de changer la forme du coude : enlever la partie métallique et la remplacer par un morceau de bois dans lequel s'encastre un servomoteur plus petit qui permet de lever le stylo. La première partie de la séance, j'ai donc découpé la plaque de bois aux mesure qu'il nous fallait pour ensuite la visser sur le servomoteur du coude (cf photo).



Etape 2 : Comprendre votre code.

Code qui est finalement assez court et plutôt simple à comprendre une fois que vous avez trouvé l'expression des angles.

```
#include <Servo.h>
#include <math.h>
Servo epaule;
Servo coude;
Servo poignet;

float tetaE=50;
float tetaC=50;
double Alpha;//angle intermediaire
float tetaE_micro, tetaC_micro;

const float le=7, lc=6.5; //longueurs en cm
float h = 0; //longueur intermédiaire
float x, y; //coordonnées du stylo
const float pi = M_PI; //Store pi in a less annoying format

void loop() {
  for(int i=0; i<100; i++){
    x=(0.11*i)-5;
    y=9;

    h=sqrt(sq(x)+sq(y));
    Alpha=(acos( (sq(h) + sq(le) - sq(lc)) / (2*le*h) )) * (180/pi);
    tetaC= (pi - (acos( (sq(le) + sq(lc) - sq(h)) / (2*le*lc) ))) * (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);

    Serial.print("x = ");
    Serial.println(x);
    Serial.print(" y = ");
    Serial.println(y);
```

Après modification de la forme du bras, j'ai juste dû changer la longueur lc.

L'exécution entraine bien une droite des points (-5,9) à (5,9), cependant un décalage de quasiment un carreau apparaît. Imprécision à traiter.

J'ai par la suite essayé de tracer un trait non plus vertical mais horizontal des points (5,7) à (5,14) (14 étant la distance max atteignable).

```
for(int i=0; i<100; i++){
    x=5;
    y=0.07*i + 7;
    //permet de tracer un trait horizontal de la position (5,7) à (5,14)

    h=sqrt(sq(x)+sq(y));
    Alpha=(acos( (sq(h) + sq(le) - sq(lc)) / (2*le*h) )) * (180/pi);
    tetaC= (pi - (acos( (sq(le) + sq(lc) - sq(h)) / (2*le*lc) ))) * (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);
}
```

La ligne est très belle.

Etape 3 : Module Bluetooth.

Nous avons décidé avec Camille de recevoir un mot reçu via une application Bluetooth que le bras devra ensuite écrire.

J'ai donc commencé par brancher le module et vérifier son bon fonctionnement. Pour cela j'ai suivi les étapes de votre cours.

```
#include <SoftwareSerial.h>

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

void setup() {
    // ...
}

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

L'envoi de « AT » m'a bien renvoyé un OK

Après un AT+NAME j'ai pu définir le nom de notre module et j'ai changé le code pin avec AT+PIN. Mon téléphone a bien pu se connecter.

Prochaine objectif des vacances : définir des fonctions qui me permettent de tracer chaque lettre de l'alphabet, toutes de même taille. Ensuite faire en sorte que lorsque le module reçoit une lettre, il sache tracer celle qui correspond. Il va falloir réfléchir à comment faire pour que le bras sache tracer les lettres peu importe sa coordonnée de départ. Pour l'instant, on choisit de quel point à quel point on souhaite tracer notre droite. Il va falloir créer des fonctions générales qui permettent de tracer des traits verticaux/horizontaux de même longueur à partir d'une coordonnée de départ. De plus on souhaite écrire des mots, il va donc falloir faire en sorte que le bras se déplace vers la droite d'une certaine distance à chaque fois qu'il aura fini une lettre. Le plus simple sera de faire en sorte que l'écriture de chaque lettre débute par le haut.