

Dossier technique

SHOOTING STARS[®]

Projet réalisé par

Marie MAZZOLINI
Apolline MELLOTT
Clément NARBAUD
Estelle LE GALL

SOMMAIRE

PARTIE 1 : ORGANISATION

1.1 Répartition des tâches	3
1.2 Planning détaillé	3

PARTIE 2 : RAPPORT TECHNIQUE

2.1 Bloc pistolet laser	5
2.2 Bloc photodétection	7
2.3 Bloc actionnement des pancartes	8

PARTIE 3 : NOTICE D'UTILISATION

ANNEXES

PARTIE 1 : ORGANISATION

1.1 Répartition des tâches

Durant le projet, nous avons décidé de former des groupes de deux personnes la plupart du temps. Ainsi, nous pouvions avancer plus vite sur un bloc en particulier en mettant en commun l'ensemble de nos compétences.

Pour les quatre premières séances, Marie et Clément ont travaillé ensemble notamment sur la partie du pistolet laser tandis qu'Apolline et Estelle ont travaillé surtout sur le montage de photodétection et le programme informatique global ainsi que la mécanique de l'abaissement des pancartes.

Durant la cinquième séance, nous avons travaillé sur les divers éléments restants pour améliorer notre jeu et le rendre plus esthétique. Pour cela, Apolline et Marie ont travaillé sur le design, Clément a travaillé sur le code principal pour rendre le jeu limité dans le temps et Estelle a élaboré un socle pour assurer une bonne liaison mécanique des composants.

Le travail de la dernière séance fut principalement d'harmoniser le projet en rassemblant les divers blocs et en réglant les derniers problèmes techniques rencontrés.

1.2 Planning détaillé

De manière générale nous avons essayé de nous conformer au planning en réalisant les tâches dans l'ordre de priorité établi dans le planning initial.

Nous avons dû tout de même effectuer des changements en temps réel sur certains points.

Tout d'abord, nous avons passé plus de temps que prévu sur le pistolet laser car nous souhaitions qu'il soit le plus autonome et compact possible (soudure du circuit électronique, pistolet en impression 3D).

Ensuite, nous avons dû adapter le planning en fonction des problèmes rencontrés comme par exemple dans la partie mécanique. Nous avons passé beaucoup de temps sur les tourelles qui permettaient d'assurer la liaison entre le servo-moteur et la pancarte car le servo-moteur n'avait pas la force nécessaire pour soulever la pancarte via la tourelle. Nous avons finalement opté pour une autre solution plus efficace compte tenu du temps restant.

Le planning détaillé ci-dessous regroupe l'ensemble de notre travail sur les 6 séances.

Planning détaillé

SÉANCE 2

Apolline et Estelle ont commencé à rédiger le code pour l'actionnement des pancartes.

Clément et Marie ont soudé leur circuit et fait les branchements du laser.

L'ensemble du groupe a testé et amélioré le code des pancartes avec plusieurs servo-moteurs. Tous les circuits ont été optimisés en taille.

SÉANCE 4

Clément et Marie ont finalisé leur montage électrique en fixant une diode laser et en alimentant le circuit de manière autonome (pile). Ils ont aussi fait une maquette de pistolet sur SolidWorks.

Estelle et Apolline ont réalisé 3 circuits indépendants (en optimisant la taille) puis ont testé le montage avec 3 servo-moteurs et 3 photodiodes.

SÉANCE 6

Harmonisation du projet.

Il a fallu resouder la plaquette du pistolet car elle était trop grande.

Assemblage final avec soudure des fils des photodiodes et test final.

12/03

12/03

11/04

02/05

09/05

16/05

SÉANCE 1

Apolline et Estelle ont réalisé le circuit de photodétection (bande passante maximisée et comparaison à un seuil fixé) et ont commandé les servo-moteurs et les tourelles mécaniques pour les pancartes.

Clément et Marie ont réalisé un circuit électrique permettant de contrôler le laser.

SÉANCE 3

Clément et Marie ont testé leur circuit laser avec un bouton poussoir et une LED. Ils ont étudié les possibilités de source et ont opté pour une diode laser. Apolline et Estelle ont réparé les problèmes sur le circuit de photodétection (ALI grillés). Elles ont modifié le programme pour deux pancartes différentes.

SÉANCE 5

L'impression 3D du pistolet est lancée. Marie et Apolline ont travaillé sur le design des pancartes et sur l'esthétique du jeu (boîte pour cacher les composants).

Clément a travaillé sur le code principal pour afficher un compteur et limiter le jeu dans le temps.

Estelle a travaillé sur les liaisons mécaniques en trouvant comment remplacer les tourelles.

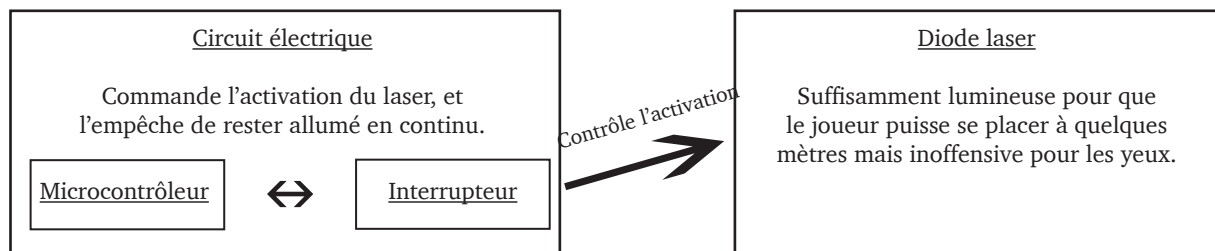
PARTIE 2 : RAPPORT TECHNIQUE

2.1 Bloc pistolet laser

Introduction

Dans notre jeu, le pistolet-laser est un composant essentiel : il permet au joueur d'activer les photodiodes en tirant dessus afin de marquer des points. Son schéma fonctionnel est composé de trois blocs principaux qui vont être présentés ci-dessous. Le système présenté ne se résume pas à un simple laser mais contient un circuit électrique afin de pénaliser d'éventuels tricheurs.

Schéma fonctionnel



Armature

Son premier but est de maintenir l'ensemble du circuit électrique + diode laser mais également de donner un aspect esthétique au système. Pour le concevoir nous avons utilisé Solidworks ainsi que l'impression 3D afin d'avoir une armature réaliste qui permet une prise en main confortable.

Circuit électrique

C'est le cœur du système. Dès le début du projet nous voulions réaliser un système « anti-triche » : en effet, en maintenant l'interrupteur enfoncé, le laser reste allumé en continu, il suffit donc juste de balader le laser entre les différentes cibles sans réellement chercher à viser ce qui facilite grandement le jeu. Le but était donc de créer un système qui force l'arrêt du laser au bout d'un certain temps et ce même si le joueur garde l'interrupteur enfoncé. Le joueur est donc obligé d'appuyer régulièrement sur la gâchette, ce qui l'oblige à rester attentif.

L'élément principal de ce circuit est le microcontrôleur PIC16F. En particulier ce sont les interruptions qui nous intéressent ici : c'est un mode de fonctionnement durant lequel le microcontrôleur a la capacité d'interrompre la boucle principale lorsqu'un évènement interne, ou externe survient.

Dans notre cas nous avons utilisé les deux types d'interruptions. L'interrupteur est responsable d'une interruption externe et va déclencher ainsi l'allumage de la diode laser. En parallèle, nous allons nous servir de l'horloge interne du microcon-

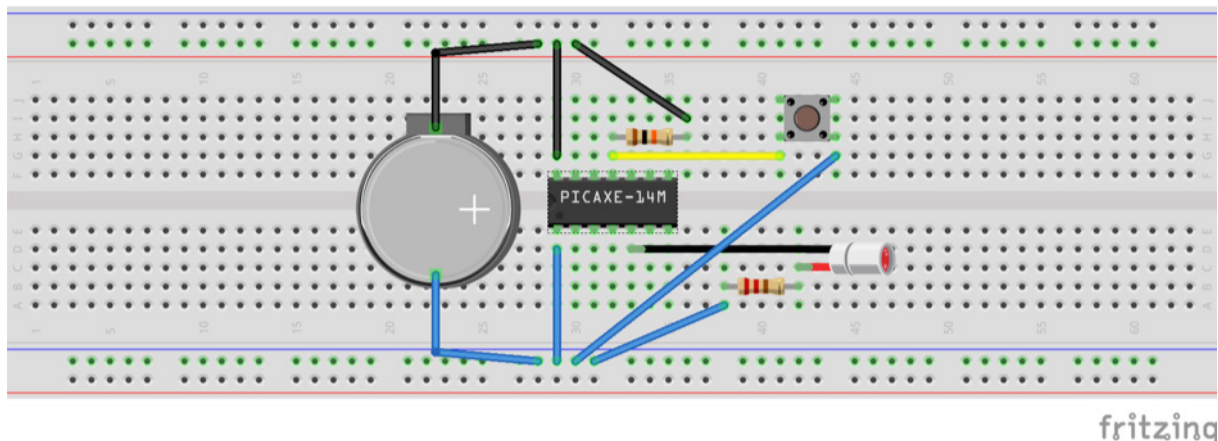
trôleur afin de créer des interruptions internes. Ainsi, lorsqu'une valeur de temps (choisie au préalable) est dépassée, la diode laser s'éteint dans tous les cas. Enfin relâcher l'interrupteur aura pour effet de réinitialiser les routines d'interruption.

Diode laser

Son activation est donc contrôlée par le circuit électrique. Nous avons besoin d'un composant qui puisse émettre un faisceau assez puissant sur quelques mètres sans pour autant être trop dangereux, la diode laser semblait donc offrir le meilleur compromis.

Schéma électrique

La principale contrainte dans la réalisation du circuit était de le rendre compact et portable. Nous avons donc décidé de souder sur une plaque d'essai. Enfin, le circuit est alimenté par deux piles bouton 3V, toujours dans une idée de rendre le système compact.



fritzing

Tous les fils bleus sont reliés au pôle positif de la pile, tandis que tous les fils noirs sont reliés au pôle négatif.

Le microcontrôleur utilisé ici est un PIC16F1503, 4 de ses sorties sont utilisées ici. Deux sont dédiées à l'alimentation. La sortie reliée à l'interrupteur est dédiée aux interruptions externes : c'est ainsi que nous contrôlons l'activation de la LED. Enfin la sortie reliée au laser, est une sortie analogique du microcontrôleur dédiée aux dispositifs d'éclairage. L'idée d'utiliser un microcontrôleur nous est venue rapidement, car nous l'avons déjà vu pour de telles utilisations en TP.

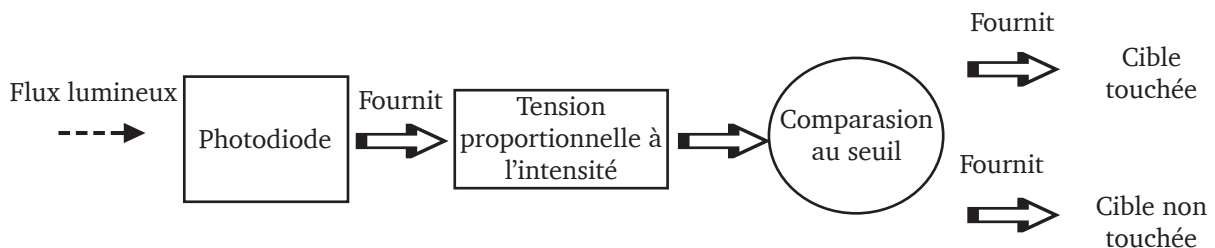
Nous avons également décidé de souder tous les composants à une plaquette afin de rendre le tout solide si le pistolet venait à être utilisé brusquement. Enfin, nous avons dû nous y reprendre à deux fois pour faire le montage et la soudure afin d'obtenir le circuit le plus compact possible et le faire rentrer dans le pistolet.

2.2 Bloc photodétection

Introduction

Le but de notre jeu étant de viser des pancartes avec un laser, nous avons dû mettre en place un système de photodétection qui perçoit si la cible a été touchée ou non. Pour cela nous avons placé sur les pancartes des photodiodes qui sont reliées à un montage optimal permettant d'obtenir une sortie de 1 (cible touchée) ou de 0 (cible non touchée).

Schéma fonctionnel



Circuit électrique

Dans le circuit émetteur, le laser va donc émettre un flux lumineux, et les photons vont induire un courant dans le circuit récepteur en arrivant sur une photodiode. Avec un circuit classique, du type filtre linéaire passe-bas, on ne peut pas augmenter la bande passante sans détériorer la sensibilité, et la bande passante est assez faible

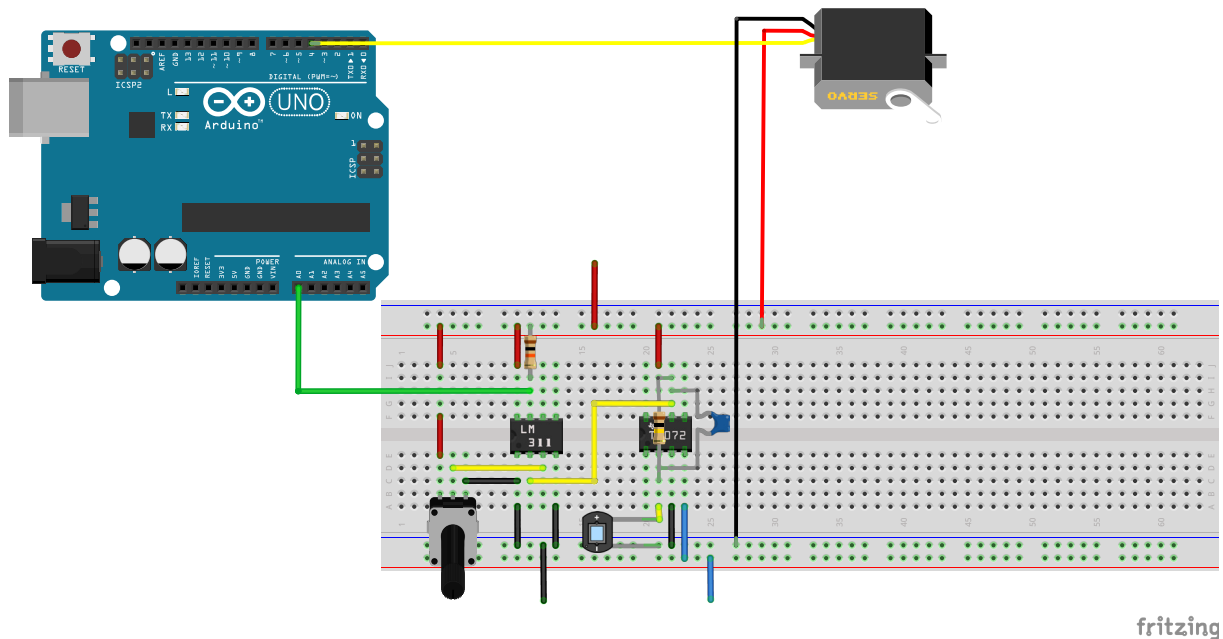
On utilise donc un montage transimpédance, qui convertit le courant issu de la photodiode en tension, associé à une capacité. Ce montage permet d'améliorer la bande passante et la sensibilité du capteur tout en limitant la résonance qu'il provoque.

Seuil de détection

En sortie du montage transimpédance, on obtient une tension proportionnelle à l'éclairement de la photodiode qui va être traitée par Arduino dans la suite. Cependant le jeu ne se joue pas dans le noir, donc il est nécessaire de s'affranchir de la lumière ambiante.

Pour cela nous avons ajouté un circuit comparateur dont le seuil est modifiable grâce à un potentiomètre. Ce qui nous intéresse est le fait qu'une photodiode soit touchée par le laser ou non, il suffit donc d'avoir en sortie de notre montage électrique deux valeurs distinctes : l'une nulle correspondant à la photodiode non touchée et l'autre d'une valeur constante (ici égale à 5V) correspondant à la photodiode touchée. Ainsi, cette sortie binaire est envoyée en entrée de la partie informatique et est associée à l'abaissement ou non de la pancarte.

Schéma électrique



2.2 Bloc actionnement des pancartes

Introduction

Nous souhaitons placer les photodiodes sur des pancartes rotatives. Nous avons donc dû créer un système permettant de faire tourner aléatoirement les pancartes. Il a également fallu mettre en place un support mécanique pour les servomoteurs.

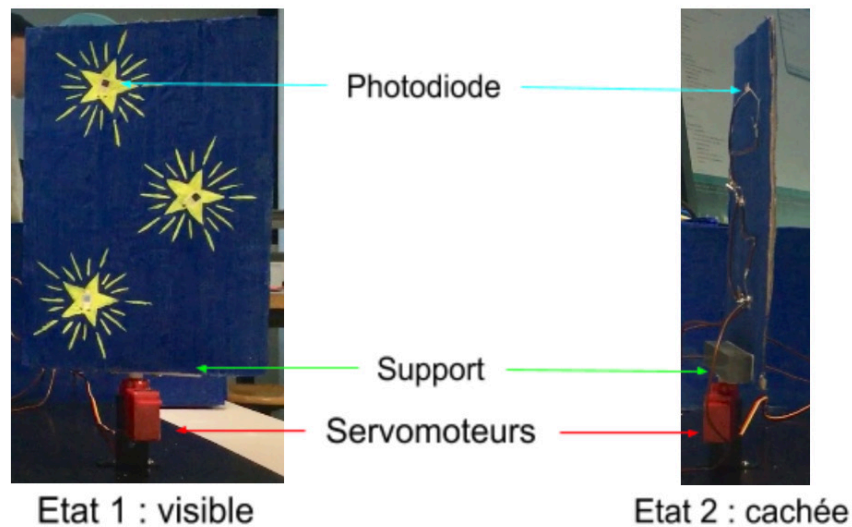
Partie mécanique

Nous avons utilisé des servomoteurs Nano car ils étaient adaptés à des tourelles permettant d'avoir une rotation des pancartes. Ces composants nous permettent de placer la pancarte dans une position, de la maintenir puis de revenir à la position initiale. Les servomoteurs sont reliés à une carte Arduino et pilotés grâce à un programme informatique.

Nous n'avons malheureusement pas pu utiliser les tourelles mécaniques car après plusieurs tests, le servo-moteurs n'était pas assez puissant pour contrer les frottements de la tourelle et la pancarte ne pouvait donc jamais se lever ou s'abaisser. Une autre solution fut de visser des blocs de plastiques sur chaque servo-moteurs sur lesquels du scotch double face permettait de fixer les pancartes.

Les servomoteurs ont ensuite été vissés sur un socle en plastique par l'intermédiaire d'une pièce de la tourelle. Le servomoteur était donc immobile et seul la pancarte bougeait.

Voici ci-dessous deux photographies de notre montage mécanique.



Partie informatique

Du point de vue informatique, les pancartes se retournent chacune leur tour selon une séquence aléatoire pendant une période que nous définissons dans le programme. Lorsque qu'une pancarte retournée est touchée, elle revient à sa position initiale instantanément, un compteur de points s'incrémente et la séquence aléatoire reprend son cours. Le programme pilotant les pancartes accompagné de commentaires se situe en annexes, il a pour entrée une valeur sortant du montage de photodétection et pour sortie une position angulaire que doit prendre le servomoteur.

PARTIE 3 : NOTICE D'UTILISATION

2 joueurs et +

Le jeu contient :



☆1 pistolet laser



☆ 3 pancartes



☆1 support de jeu

But du jeu : Toucher le plus de cibles possible à l'aide du pistolet laser.
Attention le temps est compté !

Règles du jeu :

Chaque joueur joue une partie à tour de rôle, quand c'est à ton tour de jouer, il faut te placer à **2 mètres** du plateau de jeu.

Lorsqu'une **pancarte** se tourne face à toi, vise le centre de la **cible** représentée par une étoile.

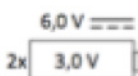
Si la pancarte se retourne avant que tu ne réussisses, tu ne gagnes pas de point, il faut alors attendre de pouvoir viser une autre pancarte. Si au contraire tu atteins la cible avec le pistolet laser, **c'est gagné** ! Tu marques un point.

Le vainqueur :

Le vainqueur est le joueur qui a marqué le plus de points durant sa partie.



Attention, ne convient pas aux enfants de moins de 3 ans



Fonctionne avec 2 piles (piles fournies)



©2018 IOGS
Tous droits réservés

ANNEXES

Programme pancarte

/*

Le programme a pour but de commander les trois servomoteurs par impulsion. A chaque début de boucle principale, le programme choisit aléatoirement l'un des trois servomoteurs à actionner.

Nanomaître 2015

Le servo est connecté au pin 8 de l'arduino

*/

```
int periode=20000; // période entre chaque début d'impulsion en microsecondes
int pinServo1=8; // variable pour le pin connecté à la commande du servomoteur
int pinServo2=7;
int pinServo3=6;
int compteur = 0; // Compteur de point
int go = 1; //permet de démarrer/arrêter le jeu
unsigned long timer = 0; //timer principal, il régit la durée totale du jeu
```

```
void setup() {
  pinMode(pinServo1,OUTPUT); // on prépare le pin en mode OUTPUT
  pinMode(pinServo2,OUTPUT); // on prépare le pin en mode OUTPUT
  pinMode(pinServo3,OUTPUT); // on prépare le pin en mode OUTPUT
  digitalWrite(pinServo1,LOW); // on initialise à l'état bas le moteur 1
  digitalWrite(pinServo2,LOW); // on initialise à l'état bas le moteur 2
  digitalWrite(pinServo3,LOW); // on initialise à l'état bas le moteur 3
  Serial.begin(9600);
}
```

//boucle principale

```
void loop() {
```

```
  while (timer<60000){ //le jeu va durer 1 minute environ
```

```
    int c=random(1,4); //l'un des trois servomoteurs est choisi aléatoirement
```

```
    int ct = 0; //compteur de temps pour les boucles while
```

```
    setAngle1(0);
```

```
    setAngle2(0);
```

```
    setAngle3(0);
```

```
    Serial.print("Temps écoulé =");
```

```
    timer = millis();
```

```
    Serial.print(timer);
```

```
    Serial.print("\n");
```

```
    delay(2000);
```

```
    Serial.println(c);
```

```
    if (c==1){ //le premier servomoteur a été choisi, la pancarte qui lui correspond est tournée et le joueur peut tirer dessus pendant un court instant
```

```
      setAngle1(200); //la pancarte est tournée
```

```

while (ct <= 10000){ //définit la durée pendant laquelle la pancarte est tournée
  //Serial.println(analogRead(A1));
  if (analogRead(A1) >= 900){ //seuil de détection choisi, la condition est vérifiée
seulement si le joueur tire sur la cible avec le pistolet laser
    setAngle1(0);
    compteur+=1;
    Serial.print("Points =");
    Serial.println(compteur);
    break; //le joueur marque un point et on sort de la boucle
  }
  ct+=1;
}
setAngle1(0);
} //si le joueur ne tire pas sur la cible, la pancarte revient dans sa position initiale après
un certain temps

if (c==2){ //le second servomoteur a été choisi

setAngle2(200);
while (ct <= 1500){
  //Serial.println(analogRead(A2));
  if (analogRead(A2) >= 900){
    setAngle2(0);
    compteur+=1;
    Serial.print("Points =");
    Serial.println(compteur);
    break;
  }
  ct+=1;
}
setAngle2(0);
}

if (c==3){ //le troisième servomoteur a été choisi
setAngle3(200);
while (ct <= 1500){
  //Serial.println(analogRead(A1));
  if (analogRead(A3) >= 900){
    setAngle3(0);
    compteur+=1;
    Serial.print("Points =");
    Serial.println(compteur);
    break;
  }
  ct+=1;
}
setAngle3(0);
}
}
if (go==1){ //le temps imparti est écoulé, toutes les pancartes sont remises dans
leur état initial et le moniteur affiche le message "GAME OVER" : le jeu est terminé.
  setAngle1(0);
  setAngle2(0);
  setAngle3(0);

```

```

Serial.print("GAME OVER");
go = 0;
}
}

```

//fonction setAngle pour envoyer les impulsions

```

void setAngle1(int a){

    int duree=map(a,0,179,1000,2000);           // on transforme l'angle en microsec-
onnes et on stocke dans la variable duree
    digitalWrite(pinServo1,LOW);               //on met le pin à l'état bas
    // la boucle qui suit est nécessaire
    // pour laisser le temps au servomoteur d'aller à sa position
    for (int t=0;t<300;t++){
        digitalWrite(pinServo1,HIGH);         // on envoie l'impulsion
        delayMicroseconds(duree);            // pendant la bonne durée
        digitalWrite(pinServo1,LOW);         // on stoppe l'impulsion
        delayMicroseconds(periode-duree);    // on attend le temps restant pour atteindre la
période
    }
}

```

```

void setAngle2(int a){

```

```

    int duree=map(a,0,179,1000,2000);           // on transforme l'angle en microsec-
onnes et on stocke dans la variable duree
    digitalWrite(pinServo2,LOW);               //on met le pin à l'état bas
    // la boucle qui suit est nécessaire
    // pour laisser le temps au servo d'aller à sa position
    for (int t=0;t<300;t++){
        digitalWrite(pinServo2,HIGH);         // on envoie l'impulsion
        delayMicroseconds(duree);            // pendant la bonne durée
        digitalWrite(pinServo2,LOW);         // on stoppe l'impulsion
        delayMicroseconds(periode-duree);    // on attend le temps restant pour atteindre la
période
    }
}

```

```

void setAngle3(int a){

```

```

    int duree=map(a,0,179,1000,2000);           // on transforme l'angle en microsec-
onnes et on stocke dans la variable duree
    digitalWrite(pinServo3,LOW);               //on met le pin à l'état bas
    // la boucle qui suit est nécessaire
    // pour laisser le temps au servo d'aller à sa position
    for (int t=0;t<300;t++){
        digitalWrite(pinServo3,HIGH);         // on envoie l'impulsion
        delayMicroseconds(duree);            // pendant la bonne durée
        digitalWrite(pinServo3,LOW);         // on stoppe l'impulsion
        delayMicroseconds(periode-duree);    // on attend le temps restant pour atteindre la
période
    }
}

```