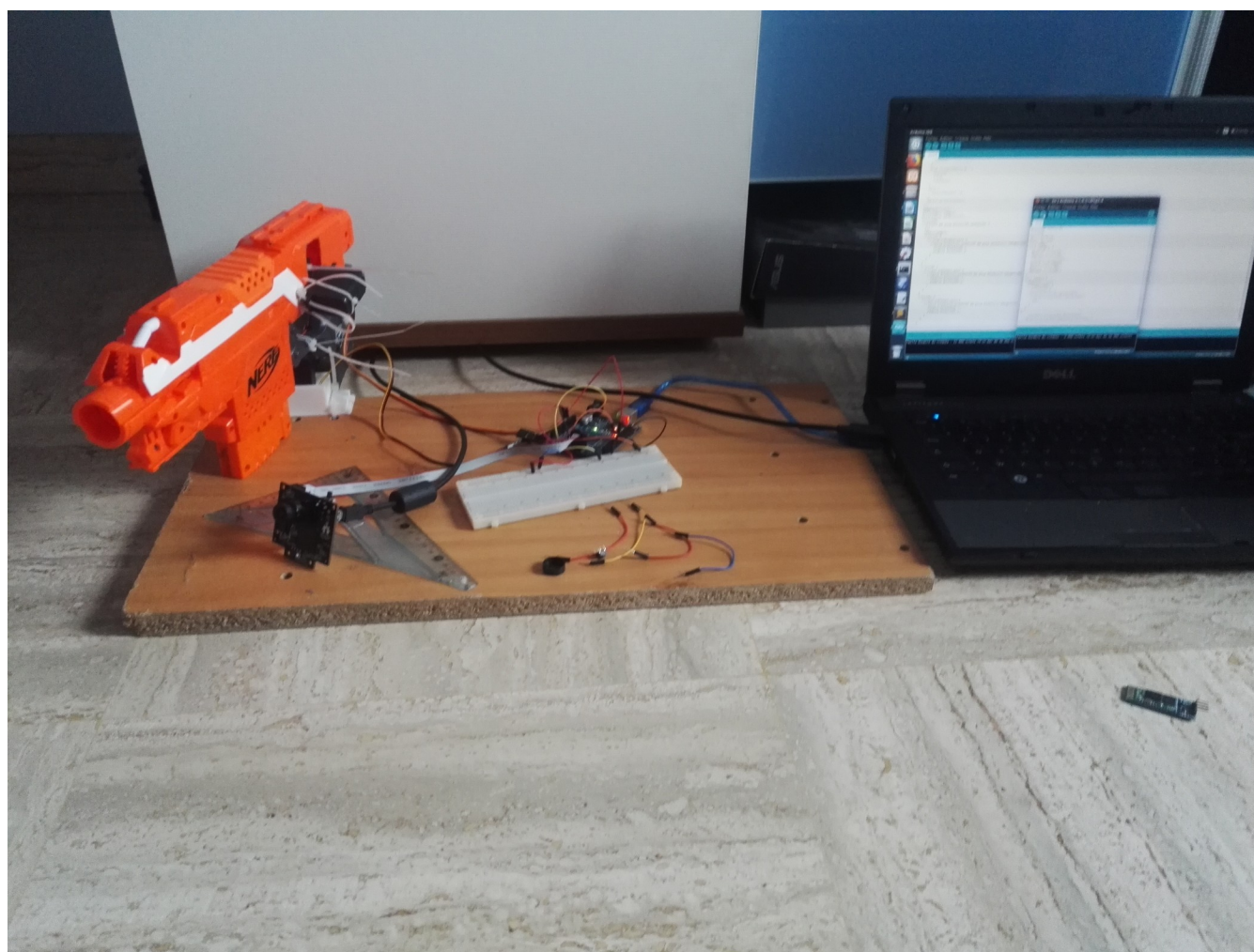


Rakotoarivony Lucas
Groupe 2

Rapport de projet : Tourelle Arduino



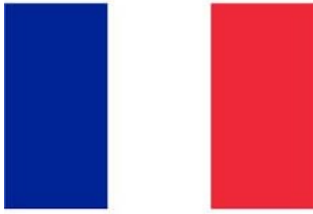
Encadrant: Pascal Masson et Fabien Ferrero
Année : 2017-2018



Sommaire :

1. Description
2. Les objectifs
3. Le matériel
4. Le planning
5. Les difficulté et les solutions
6. Les perspectives
7. Le code
8. Conclusion

Description :

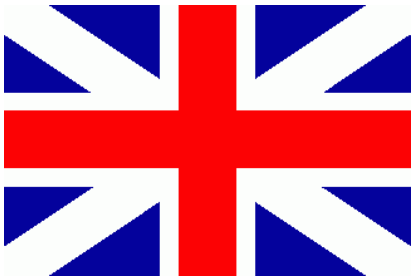


Mon projet est intitulé la Tourelle Arduino. Le principe est simple détecter un objet de couleur et lui tirer dessus à l'aide d'un Nerf.

L'objet en question sera détecté à l'aide de la caméra Pixy, sa trajectoire sera calculé à l'aide de l'Arduino et ensuite la tourelle se décalera et tirera dans la bonne direction.

L'application Android servira à choisir quelles couleurs seront détectés et arrêter la tourelle sur commande. L'application sera relié par Bluetooth à mon téléphone.

Lors de la démonstration je ferai passer des objets de couleur devant la tourelle et ensuite le Nerf tirera sur cet objet en question.



My project is entitled the Turret Arduino. The principle is simple to detect an object of color and to shoot it by means of a Nerve.

The object in question will be detected by means of the camera Pixy, its trajectory will be calculated by means of Arduino and then the turret will move and will fire in the right direction.

The Android application will serve to choose which colors will be detected and to stop the turret to order. The application will be connected by Bluetooth with my telephone.

During the demonstration I shall make cross objects of color in front of the turret and then the Nerve will fire at this object in question.

Objectifs :

- Réaliser un objet par moi-même
- Réussir à détecter un objet en mouvement
- Réussir à gérer différents moteurs simultanément
- Construire la structure de la tourelle

Matériel :

- un Nerf
- deux servomoteurs (un pour tirer et un pour déplacer la tourelle).
- une planche en bois (support pour la tourelle)
- une caméra Pixy (pour détecter l'objet)
- un support pour le moteur (pour bloquer le moteur)
- une carte Arduino Uno
- un HC-06 (pour faire la liaison Bluetooth)
- une plaque à trous
- plusieurs fils
- plusieurs serres-câbles (pour fixer tout les moteurs)
- le logiciel Pixymon (pour avoir la vision de la caméra)

Planning :

	Prévision	Réalité
Janvier :	Déclencher le tir du nerf	Fait en mars
Février :	Partie code avec la Pixy	Fait dans les temps
Mars :	Structure de la tourelle	Sera fait en avril
Avril :	Commencer les test	Sera fait en mai
Mai :	Fonctionnalités supplémentaires	Non fait

Difficultés et solutions :

Difficultés / solutions

Réussir à bloquer les moteurs / Utilisation de structures créées avec Autodesk et usage de serre-cable

Calculer la trajectoire de l'objet / Utilisation de la librairie Pixy et du logiciel Pixymon

Déclencher le tir sur commande / Usage de servomoteurs puissants et fixer l'embout à l'aide de serre-cable

Créer l'application bluetooth et la faire fonctionner en parallèle de la caméra / Pas encore résolu, tout envoi de donnée à travers l'application stoppe mon code

Perspectives :

- Réussir à faire fonctionner l'application et la caméra Pixy simultanément.
- Utiliser un moteur plus puissant pour faciliter le tir.
- Calculer de manière plus précise la trajectoire de l'objet en prenant plus de points en paramètre.
- Améliorer la précision de la tourelle en utilisant une caméra et de meilleure qualité et déterminer de manière optimale la trajectoire de l'objet.
- Réussir à gérer plusieurs objets simultanément et ne pas prendre en compte la couleurs des murs.
- Améliorer le coté esthétique de la tourelle.
- Ajouter un module ultrason (HC-SR04) pour connaître la distance entre l'objet et la tourelle (la tourelle gère uniquement les déplacements latéraux).

Code :

pixy

```
#include <SPI.h>
#include <Pixy.h>
#include <Servo.h>
#include <SoftwareSerial.h> //Software Serial Port
#define RxD 10 //Pin 10 pour RX, PB2 sur votre board, a brancher sur le TX du HC-06
#define TxD 11 //Pin 11 pour TX, PB3 sur votre board, a brancher sur le RX du HC-06
SoftwareSerial BTSerie(RxD,TxD);

Servo myservo; //servo se chargeant du tir
Servo deplace; //servo se chargeant de pivoter
Pixy pixy; //la caméra
uint16_t blocks;
int i=0;
int j;
int k=0;
float x1;
float y1;
float x2;
float y2;
float x3;
float y3; //on définit les coordonnées

float a;
float b;
float c; //on définit les coefficients

float delta;
float r1;
float r2; //on définit les racines

float cible=50;
int pos=-100;
int final=-5;
int color=2; //variable non utilisée mais utile si application bluetooth
int start=1;

long temps1;
long temps2;
long diff; //permet de gérer le temps

int tir=0;
int calme=95; //les positions de tir et de rechargement
```

```

int tir=0;
int calme=95; //les positions de tir et de rechargement
char recvChar;

void setup()
{
  Serial.begin(9600);
  Serial.println("Début");
  pixy.init();
  myservo.attach(8);
  myservo.write(calme);
  deplace.attach(9);
  deplace.write(30);

  //on met les servo au bon endroit

  /* pinMode(RxD, INPUT);
  pinMode(TxD, OUTPUT);
  BTSerie.begin(9600);
  Serial.println("En attente de la commandes AT");
  delay(100);
  BTSerie.print("AT+BAUD4");
  delay(500); */
  //cette partie servait pour le bluetooth
}

void loop() {
  /* if (BTSerie.available()) {
  }
  recvChar = BTSerie.read();
  if(recvChar=='S') {
    start=1;
  }
  else if(recvChar=='N') {
    start=0;
    //i=0;
    //color=8;
  }
  else {
    color=recvChar-'0';
  }
  Serial.println(color);
  */
  }
  else {
    color=recvChar-'0';
  }
  Serial.println(color);
  */
  //cette partie servait pour le bluetooth

  temps1=millis();
  diff=temps1-temps2;//on calcule la durée entre deux détections

  blocks=pixy.getBlocks(); //c'est cette fonction qui bloque le bluetooth
  delay(30);
  if(blocks && pixy.blocks[0].width>20) {
    i++;
    temps2=temps1;
    if(i==2) {
      for(j=0;j<blocks;j++) {
        if(pixy.blocks[j].width>20 && pixy.blocks[j].height>20) {
          x1=pixy.blocks[0].x;
          y1=pixy.blocks[0].y;
        }
      }
    }
    if(i==4) {
      for(j=0;j<blocks;j++) {
        if(pixy.blocks[j].width>20 && pixy.blocks[j].height>20) {
          x2=pixy.blocks[0].x;
          y2=pixy.blocks[0].y;
        }
      }
    }
    if(i==6) {
      for(j=0;j<blocks;j++) {
        if(pixy.blocks[j].width>20 && pixy.blocks[j].height>20) {
          x3=pixy.blocks[0].x;
          y3=pixy.blocks[0].y;
        }
      }
    }
  }
  }
  //dans cette partie on affecte les coordonnées de l'objet toutes les 60ms

```



```

if(start==1) {
  if(x1!=0 && x2!=0 && x3!=0 && x1<x3) {
    c=y1-a*x1*x1-b*x1;
    b=(y2-a*x2*x2-c)/x2;
    a=(y3-b*x3-c)/(x3*x3);
    delta=b*b-4*a*(c-cible);
    if(delta<=0) {
      Serial.println("Cible intouchable");
    }
    else {
      r1=(-b-sqrt(delta))/(2*a);
      r2=(-b+sqrt(delta))/(2*a);
      if(x3-x1>0) {
        Serial.println(r2);
        pos=(int) (r2/14);
      }
      else {
        Serial.println(r1);
        pos=(int) (r1/14);
      }
    }
  }
  //on calcule la trajectoire de l'objet et on détermine à quelle moment
  //il pourrait être touché par le Nerf et on détermine pos

  else if(x1!=0 && x2!=0 && x3!=0 && x1>x3) {
    c=y3-a*x3*x3-b*x3;
    b=(y2-a*x2*x2-c)/x2;
    a=(y1-b*x1-c)/(x1*x1);
    delta=b*b-4*a*(c-cible);
    if(delta<=0) {
      Serial.println("Cible intouchable");
    }
    else {
      r1=(-b-sqrt(delta))/(2*a);
      r2=(-b+sqrt(delta))/(2*a);
      if(x3-x1>0) {
        Serial.println(r2);
        pos=(int) (r2/14);
      }
      else {
        Serial.println(r1);
        pos=(int) (r1/14);
      }
    }
  }
  //idem mais dans l'autre sens,
  //si on ne fait pas ce cas l'arduino détecte l'objet à l'infini dans un sens

  if(pos<20 && pos>-20) {pos+=20;}
  if(pos>80 && pos<100) {pos-=20;}
  if(pos>0 && pos<=60) {
    deplace.write(pos);
    delay(20);
    while(k<1) {
      myservo.write(tir);
      delay(1200);
      myservo.write(calme);
      delay(800);
      Serial.println("tir");
      k++;
    }
    a=0;
    b=0;
    c=0;
    i=0;
    pos=0;
  }
  //si l'objet est dans la range de tir du nerf on le décale et on tire

  Serial.print("Cible touchable en ");
  Serial.print(r1);
  Serial.print(" ou ");
  Serial.println(r2);
  Serial.print("pos :");
  Serial.println(pos);
  Serial.print("y=");Serial.print(a);Serial.print("x*x+");
  Serial.print(b);Serial.print("x+");Serial.println(c);
  Serial.println(i);
  if(diff>3000) {
    i=0;
    a=0;
    b=0;
    c=0;
    pos=0;
  }
}

```

```

Serial.print("Cible touchable en ");
Serial.print(r1);
Serial.print(" ou ");
Serial.println(r2);
Serial.print("pos :");
Serial.println(pos);
Serial.print("y=");Serial.print(a);Serial.print("x*x+");
Serial.print(b);Serial.print("x+");Serial.println(c);
Serial.println(i);
    if(diff>3000) {
        i=0;
        a=0;
        b=0;
        c=0;
        k=0;
        //si on ne détecte rien pendant 3s on réinitialise les valeurs;
    }
}
}

```

Il est possible que le code rendu sur Github soit légèrement modifié pour causes d'ajustement.

Conclusion :

Actuellement l'objectif de détecter est de calculer la trajectoire de l'objet est atteint. La tourelle détecte l'objet et le vise. Cependant ma tourelle a encore plein de défauts.

Tout d'abord elle n'est pas assez précise pour toucher un objet en mouvement. De plus paradoxalement plus l'objet est volumineux, moins le calcul de la trajectoire sera précis.

En conclusion ce projet a été une expérience bénéfique, il m'a permis d'être autonome du début à la fin.

De plus même si la tourelle peut encore être améliorée, je considère ce projet comme une réussite personnelle.