

# Rapport Technique Convoyeur et Trieur



**BEN ALI Mohamed Aziz, HIA Inès, XIAO Zijun, SUN Chenyu**

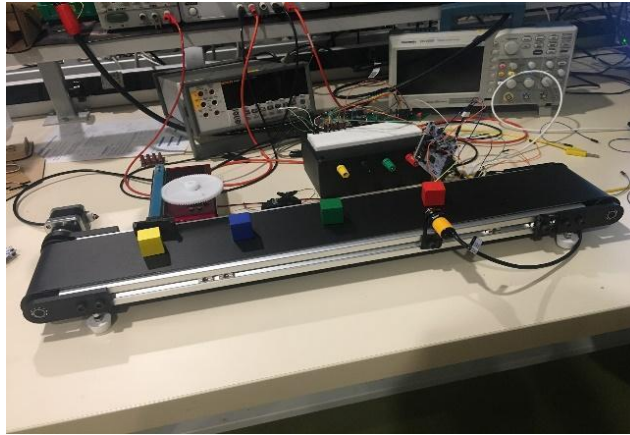
# **SOMMAIRE**

I - Introduction du projet : problématique et objectifs.....	p. 2
II - Matériels utilisés.....	p. 3
III - Schéma Bloc.....	p. 4
IV - Planning et Organisation.....	p. 4
V - Déroulement et problèmes rencontrés.....	p. 5
VI - Module de détection des couleurs et des formes.....	p. 6
VII - Schéma électrique du câblage.....	p. 9
VIII - Résultats et pistes d'améliorations.....	p. 9
IX - Ressenti sur le projet.....	p. 10
ANNEXE.....	p. 11

## I - Problématique et objectifs du système

Nous nous sommes penchés sur la conception **d'un convoyeur trieur** qui servira à trier automatiquement une chaîne de produits par couleur (voire par forme).

Il s'agirait d'une détection de pièces par un détecteur infrarouge et une caméra pour traduire la forme et la couleur.



On s'est fixé comme **objectifs** :




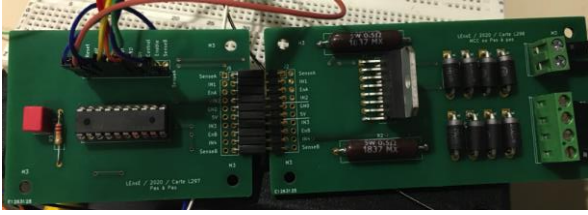

1. Faire fonctionner le convoyeur
2. Arriver à détecter la présence d'une pièce
3. Proposer une solution pour différencier les couleurs et les formes des pièces
4. Faire fonctionner les distributeurs trieurs
5. Développer une interface Humaine-Machine

Parmi les **contraintes** imposées dans le projet :

1. Arriver à analyser au minimum 10 pièces par minute
2. Utiliser un capteur pour détecter la présence d'une pièce et une caméra pour les couleurs et formes
3. Une erreur d'une pièce sur 1000 est tolérée sur la détection des couleurs

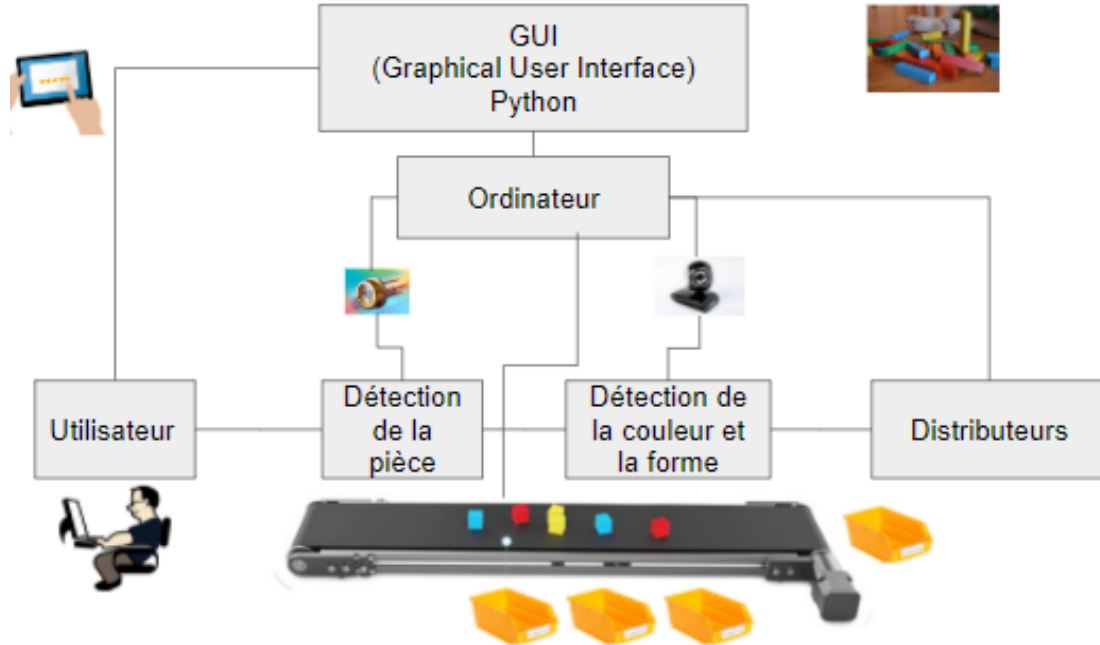
## II – Matériels

Durant notre projet, nous avons utilisé le matériel suivant :

<p><b>Convoyeur</b></p> 	<p><b>Distributeur avec moteur PARALLAX Standard Servo</b></p> 	<p><b>Capteur infrarouge E18-D80NK</b></p> 	<p><b>Caméra BASETech Model N°SC-626</b></p> 
<p><b>Carte Nucléo</b></p> 	<p><b>Raspberry Pi 3B</b></p> 	<p><b>Cartes L297 et L298</b></p>  <p><b>Deux générateurs de tension : un pour le moteur du convoyeur et un pour la carte Nucléo</b></p> 	

### III - Schéma Bloc Synthétique

Au départ, nous avons réfléchi et dessiné le schéma bloc suivant :



### IV - Planning et Organisation

Nous avons par ailleurs décidé lors de cette première séance de suivre le planning et l'organisation ci-dessous :

→ **Planning :**

15 Mars	Proposer une solution pour détecter la présence et la couleur d'une pièce
18 Mars	Collecter les informations et faire fonctionner les distributeurs
22 Mars	Avoir une interface Humain-Machine prête
29 Mars	Arriver à analyser 10 pièces par minute + Commencer les livrables
1 Avril	Atteindre une erreur d'une pièce sur 1000 + Finaliser les livrables
12 Avril	Présenter notre produit et le vendre

→ **Organisation :**

<b>Aziz &amp; Inès</b>	Réflexion par rapport à l'établissement d'une solution de détection et de tri et le codage de la carte Nucleo pour assurer le bon fonctionnement du banc
<b>Zijun</b>	Documentation sur le matériel disponible et la réalisation du câblage du système

<b>Chenyu</b>	<p>Réalisation du module de vision industrielle basée sur Raspberry Pi.</p> <p>Réalisation de la détection des couleurs et des formes des pièces.</p> <p>L'achèvement préliminaire de l'interface graphique.</p>
---------------	--

## V - Déroulement & problèmes rencontrés

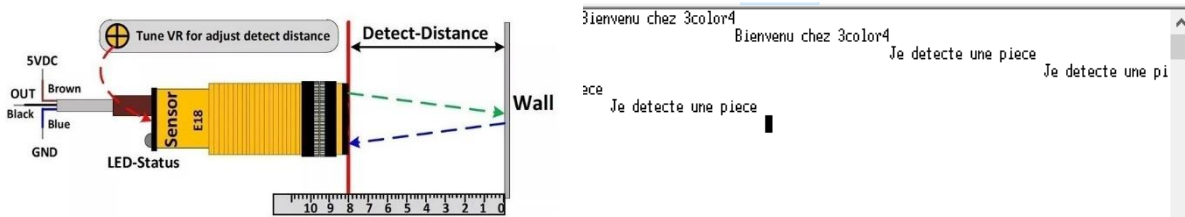
Le projet s'est alors déroulé selon les étapes suivantes :

1. Faire fonctionner le moteur du convoyeur avec un générateur de front d'onde, puis en codant sur MBED

⇒ Problème de rapidité : le moteur du convoyeur a une vitesse maximale assez faible. **Impossible d'analyser 10 pièces par minutes**

2. Mise en place du capteur infrarouge E18-D80NK et du programme permettant de **détecter le passage d'une pièce** sur le convoyeur

⇒ Tera Term nous a permis de vérifier le fonctionnement du code

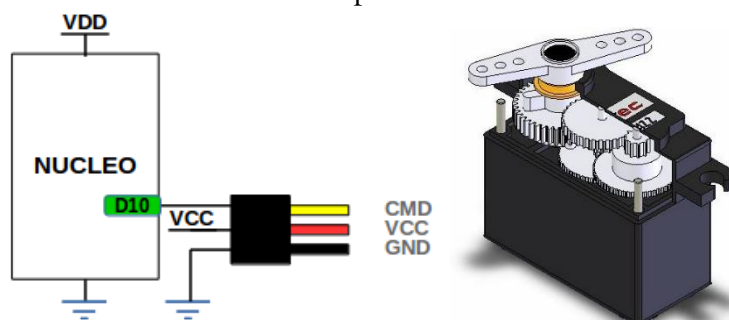


### Fonctionnement du capteur infrarouge

3. Faire fonctionner un distributeur pendant une certaine durée après la détection de la pièce par le capteur

⇒ On utilise des servomoteurs qui permettent de réaliser une rotation d'un angle calibré en fonction d'une commande externe

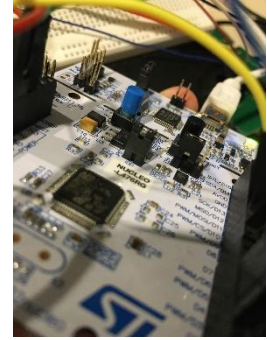
⇒ Synchronisation du moment de la détection avec le moment d'éjection de la pièce : Calcul du temps nécessaire pour atteindre un distributeur après détection



Branchement du servomoteur sur la carte Nucléo

4. Automatisation du fonctionnement du convoyeur :

⇒ **Rendre la carte Nucléo autonome et ainsi le banc indépendant**



5. Programmation en Python-OpenCV sur Raspberry PI afin d'arriver à différencier les couleurs et les formes de chaque pièce

6. Transfert d'informations données par la caméra, de la Raspberry à la carte Nucléo : La Nucléo est codée en C, la Raspberry en Python

⇒ **Quel moyen de communication choisir?**

a) Tout basculer sur Python ⇒ Perte de temps et devoir tout recommencer dès le début

b) Une connexion série entre les deux cartes ⇒ On n'arrive pas à mettre en place un code fonctionnel

c) **Utiliser GPIO pour envoyer des signaux sur la carte Nucléo** ⇒ la meilleure solution

7. Mise en marche de l'ensemble du système

## VI - Module de détection des couleurs et des formes

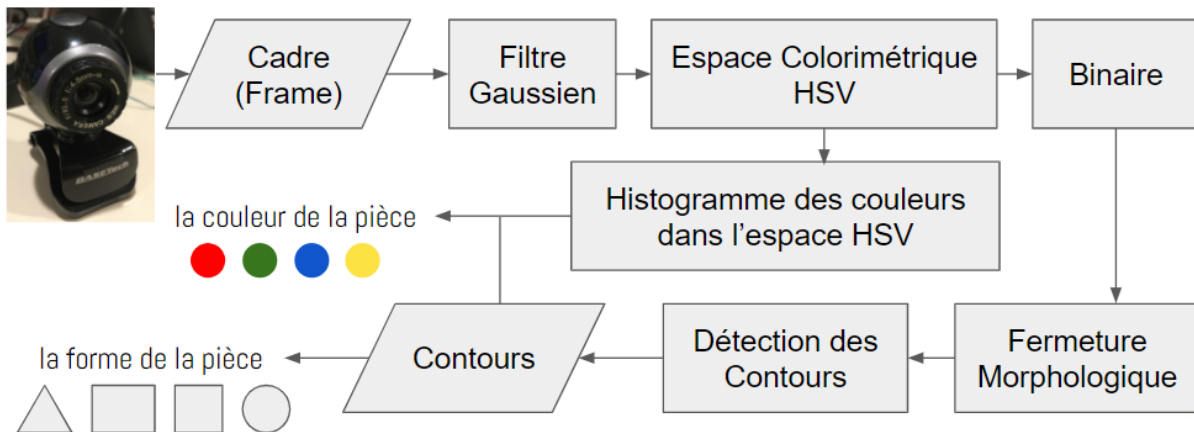


Schéma blocs du programme de détection

Afin de réaliser la vision industrielle, nous utilisons une caméra pour acquérir des images et effectuer un traitement d'image via la programmation python-opencv sur le Raspberry Pi.

(Environnement de développement : linux (Raspbian), python 3, OpenCV 2)

Tout d'abord, on utilise le Raspberry Pi pour lire chaque image de la caméra, puis on fait passer le filtre gaussien pour réduire le bruit.

Ensuite, nous effectuons la détection des contours et la reconnaissance des couleurs dans l'espace colorimétrique HSV

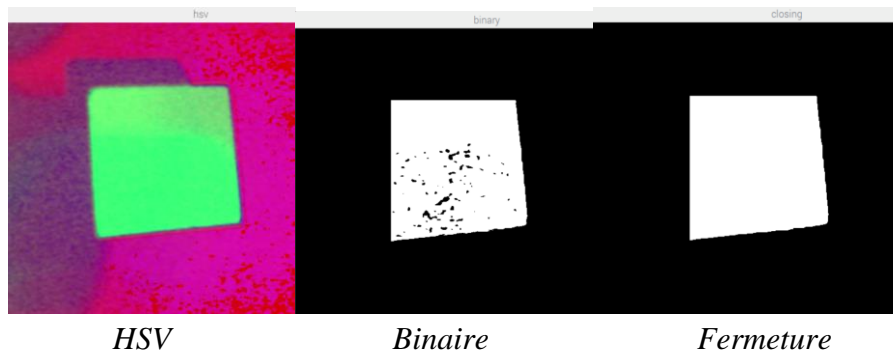
## a) Détection des formes

Afin de réaliser la détection des contours, l'image est d'abord binarisée.

Nous pouvons constater qu'il y a encore de nombreux petits « trous » dans l'image après la binarisation, nous utilisons donc des opérations de fermeture morphologique pour les éliminer.

L'opération de fermeture de la morphologie consiste à se dilater d'abord puis à s'éroder.

⇒ On peut constater que l'opération de fermeture forme très bien le seul contour fermé.



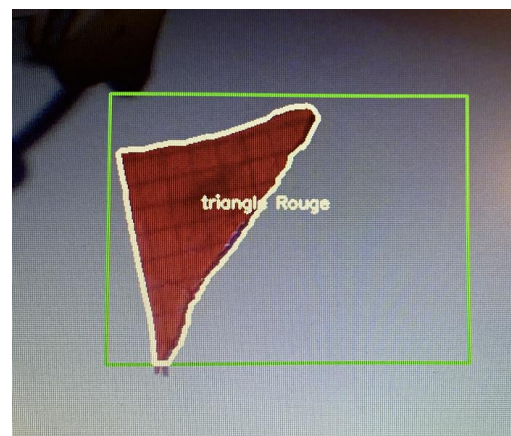
Grâce à OpenCV, nous pouvons non seulement obtenir des arêtes, mais également obtenir le nombre d'arêtes de ligne droite par ajustement polynomial. En fonction du nombre d'arêtes différents, nous définissons différentes formes :

**3** : triangulaire,

**4** : rectangulaire (carré si le rapport hauteur / largeur est inférieur à 1,1),

**> 5** : circulaire,

**Autre** : indéfini, exprimé en "forme"



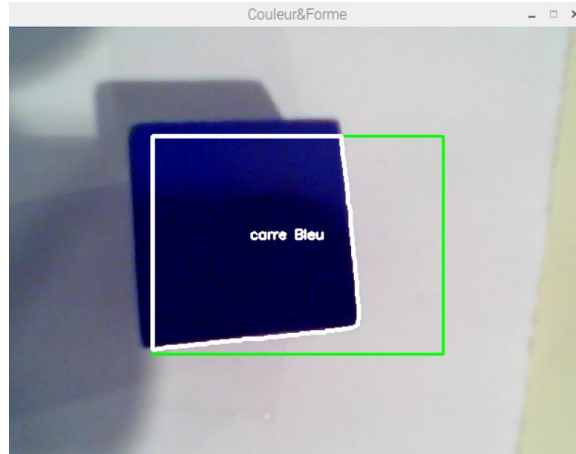
## b) Détection des couleurs

Le composant H dans l'espace colorimétrique HSV représente la couleur. Nous calculons l'histogramme de distribution de la composante H dans le contour détecté et trouvons la composante H avec la plus grande fréquence, qui est considérée comme la couleur de la pièce.



Différentes gammes de couleurs ont des valeurs H différentes. Nous pouvons trouver la valeur dont nous avons besoin à l'URL suivante : <http://colorizer.org/>

Dans le programme, après avoir testé et lu la couleur d'une pièce spécifique, nous avons choisi une gamme de couleurs plus large pour faciliter la distinction du rouge, du jaune, du bleu et du vert (On a juste de ces quatre couleurs).



*Détection de la pièce carré bleu*

De plus, le programme limite également la mesure de la seule pièce à l'intérieur du rectangle vert au centre pour éviter de détecter le bord du convoyeur.

### Codes

Si on modifie ou améliore le code à l'avenir, on le mettra à jour sur GitHub. Tous les commentaires de code sont en anglais.

### GitHub

[https://github.com/csun19/Detection\\_Tri\\_ProTIS2021\\_IOGS.git](https://github.com/csun19/Detection_Tri_ProTIS2021_IOGS.git)

### Google Drive

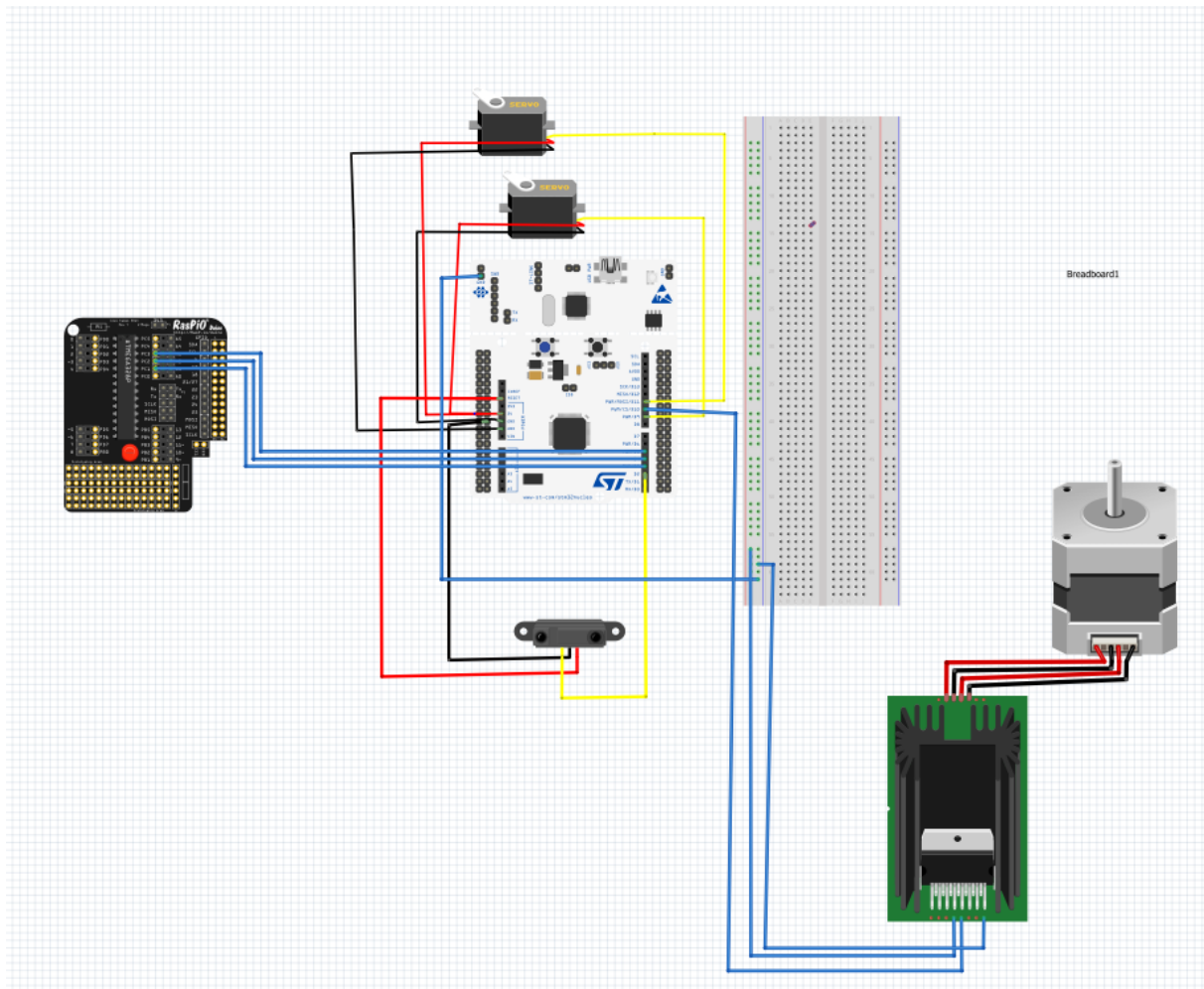
[https://drive.google.com/drive/folders/1X\\_UUGf-3DiuyXIKcevVBJ5hGRIWfo-F8?usp=sharing](https://drive.google.com/drive/folders/1X_UUGf-3DiuyXIKcevVBJ5hGRIWfo-F8?usp=sharing)

On peut voir sur les photos qui suivent plus d'exemples de détection de couleurs.



## VII) Schéma électrique du câblage

Finalement, on a le schéma électrique suivant pour le fonctionnement de notre système :



## VIII – Résultats et pistes d'améliorations

Notre équipe a réussi, en se répartissant les tâches, à avancer progressivement sur le projet et à avoir un banc indépendant fonctionnel, qui permet de détecter la présence des pièces avec la couleur (4 couleurs) et la forme (4 formes).

Pour le moment, il ne répond pas à certains critères qu'on s'était fixés, mais il répond au besoin de convoyeur trieur.

→ **Pistes d'amélioration :**

- Travailler avec un meilleur moteur ⇒ Pour arriver à analyser 10 pièces par minute
- Avoir les 3 servomoteurs, les 3 distributeurs, calibrer chacun selon la distance qui les sépare du capteur de position pour programmer
- Réussir à détecter plus de formes et de couleurs
- L'interface humain-machine : Recevoir plusieurs informations : nombre de pièces détectés selon

couleur et forme : La Raspberry PI suite à la détection peut calculer le nombre et l'afficher  
⇒ Un fichier texte peut être généré pour enregistrer toutes les données

- Améliorer l'algorithme de différenciation des formes et des couleurs afin de minimiser le nombre d'erreur lors du tri des pièces
- Réussir à différencier les formes et les couleurs avec n'importe quel éclairage et sur le tapis du convoyeur noir

## VIII– Ressenti sur le projet

Après les 6 séances de travail, chaque membre de l'équipe a fait un bilan personnel à propos du ressenti sur ce projet :

*“ Je trouve que la vision industrielle est très intéressante. En effet, dans les processus de production industrielle répétitifs de masse, les méthodes d'inspection par vision industrielle peuvent considérablement améliorer l'efficacité et l'automatisation de la production. Cela peut alléger le travail des gens. De plus, j'ai acquis des connaissances sur le système d'exploitation Linux et python grâce à ProTIS. C'est vraiment fantastique! ” - Chenyu SUN*

*“ J'ai participé à la construction du circuit et à la programmation de l'ensemble du projet. Grâce à cela, j'ai une compréhension plus profonde du processus de production d'un produit électronique. Non seulement j'ai passé en revue ce que j'ai appris, mais j'ai également une meilleure compréhension de ce nouveau domaine. En même temps, je pense également que l'ensemble du projet est très intéressant et j'espère que tout le produit sera 100% bien. ” - Zijun XIAO*

*“ Ce fut une expérience assez riche en termes d'acquisition de connaissances et de savoir-faire. Il y a eu plusieurs hauts et bas. Entre les moments de réflexion et les discussions internes, on a pu beaucoup échanger entre nous et juger des meilleures solutions. Travailler sur un produit concret, de la phase de conception à la mise en œuvre m'a beaucoup plu. Cela m'a permis de mettre en place des connaissances dans le domaine électronique, la programmation, et tout ce qui est industriel. Préparer un produit et réussir à le vendre est un tout un processus assez riche. Dans l'idéal, on pourrait améliorer le produit et jouer sur ces performances. En tous cas, je retiens du projet ProTIS que de bons souvenirs pour clore la 2A. ” - Aziz BEN ALI*

*“ Ce projet a été pour moi une vraie découverte de l'électronique et de l'utilisation de carte électronique. En effet, étant arrivée à SupOptique seulement en 2A, je n'avais encore jamais fait aucun TP et projet dans ce domaine. J'ai pu par ailleurs découvrir et apprendre le langage informatique C. Ainsi, mener un projet d'une demande d'un prototype, puis de la réflexion pour sa mise en place, et enfin de la conception et mise en fonctionnement, a été une expérience riche et pas toujours facile. Mais le travail d'équipe a été essentiel dans ce projet et nous a permis d'avoir un produit fonctionnel. ” - Inès HIA*

# ANNEXE

## CODE CARTE NUCLEO

```
#include "mbed.h"

DigitalIn detecteur(D2); // sortie du capteur
InterruptIn CR(D3); // detecteur rouge comme interruption
InterruptIn CV(D4); // detecteur vert comme interruption
InterruptIn CB(D5); // Detecteur bleu comme interruption
PwmOut convoyeur_pwm(D10); // entrée pour le moteur du convoyeur
PwmOut servo_motR(D11); // entrée pour le moteur du distributeur R

Serial pc(USBTX, USBRX); // Communication série avec l'ordinateur
void distribuer(void); // Déclaration de la fonction d'interruption 1

int etatPrecedent = 0; // état initial si pas de mouvement
int rouge,vert,bleu; // couleur détectée

void distribuer(){
    rouge=CR;
    vert=CV;
    bleu=CB;
    if(vert==1) {
        pc.printf("la piece est verte\n ");
    }
    if(rouge==1) {
        pc.printf("la piece est rouge\n ");
    }
    if(bleu==1){
        pc.printf("la piece est bleue\n ");
    }
    wait_us(30000000);
    servo_motR.pulsewidth_us(2000); // Angle positif pour faire sortir la tige du distributeur
    wait_us(1000000);
    servo_motR.pulsewidth_us(900); // Angle négatif
}

int main()
{
    pc.printf("Bienvenu chez 3color4\n");
    convoyeur_pwm.period_ms(5); // un signal de periode 5ms
    convoyeur_pwm.pulsewidth_ms(2); // un pulse de durée 2 ms
    servo_motR.period_ms(20); // Initialisation de la période
}
```

```
servo_motR.pulsewidth_us(1900);//Initialisation en position 0
/*servo_motV.period_ms(20); // Initialisation de la période
servo_motV.pulsewidth_us(1900);//initialisation en position 0*/
while(1) {
if (detecteur == 1)
{// pas mouvement détecté

servo_motR.pulsewidth_us(900);// Angle négatif
/*servo_motV.pulsewidth_us(900);// Angle négatif*/
}

else {
// un mouvement détecté
if (etatPrecedent == 0) { // l'état vient de changer
pc.printf("Je detecte une piece \n");
etatPrecedent = 1;
CR.rise(&distribuer);
CV.rise(&distribuer);
CB.rise(&distribuer);

}
}
etatPrecedent = 0;// revenir à l'état initial
rouge=0;// revenir à l'état initial
vert=0;// revenir à l'état initial
bleu=0;// revenir à l'état initial

}

}
```

## **CODE Raspberry PI**

### **GitHub**

[https://github.com/csun19/Detection\\_Tri\\_ProTIS2021\\_IOGS.git](https://github.com/csun19/Detection_Tri_ProTIS2021_IOGS.git)

### **Google Drive**

[https://drive.google.com/drive/folders/1X\\_UUGf-3DiuyXIKcevVBJ5hGRIWfo-F8?usp=sharing](https://drive.google.com/drive/folders/1X_UUGf-3DiuyXIKcevVBJ5hGRIWfo-F8?usp=sharing)

### **Vidéo**

[https://drive.google.com/drive/folders/1-C6BIyBIIvBhcRTDPxUGh\\_OwiltTw1Hv?usp=sharing](https://drive.google.com/drive/folders/1-C6BIyBIIvBhcRTDPxUGh_OwiltTw1Hv?usp=sharing)