



ProTIS - Rapport technique

---

# Convoyeur et trieur

---

*Auteurs :*

M<sup>me</sup> Maëlys ALQUIER

M. Louis FOLLET

M<sup>me</sup> Rayanne MARTINY

M. Thomas TROCELLO



Version du  
15 avril 2021

## Table des matières

<b>1</b>	<b>Introduction et motivation</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Chaîne fonctionnelle . . . . .	3
1.3	Planning prévisonnel . . . . .	3
1.4	Planning réalisé . . . . .	4
<b>2</b>	<b>Partie électronique</b>	<b>7</b>
2.1	Tapis roulant . . . . .	7
2.2	Boîte tournante . . . . .	8
2.3	Ecran LCD . . . . .	9
<b>3</b>	<b>Partie mécanique</b>	<b>10</b>
3.1	Boîte à compartiments . . . . .	10
3.2	Boîte à éclairage constant . . . . .	11
<b>4</b>	<b>Partie informatique</b>	<b>12</b>
<b>5</b>	<b>Résultats et analyse</b>	<b>12</b>
<b>6</b>	<b>Bilan - Retour d'expérience</b>	<b>13</b>
	<b>Annexe</b>	<b>15</b>
	Programme n°1 : Tapis roulant . . . . .	15
	Programme n°2 : Boîte de tri et Interface Humain-Machine . . . . .	18
	Programme n°3 : Détection des couleurs . . . . .	28

# 1 Introduction et motivation

## 1.1 Motivation

Dans le cadre du partenariat entre l'Institut d'Optique Graduate School et la société SOLEC, nous avons été amenés à choisir un système à réaliser dans le cadre du module ProTIS. Notre groupe de quatre étudiants - Louis, Thomas, Rayanne et Maëlys - voulait travailler sur un sujet qui nous parlait à tous et que nous trouverions sympathique et intéressant à produire.

Après concertation, nous nous sommes mis d'accord pour conceptualiser et produire un convoyeur et trieur. L'objectif que nous nous sommes alors fixés était le développement d'un système permettant la détection de cubes de différentes couleurs (rouge, vert, jaune et bleu) qui seront ensuite automatiquement placés dans le conteneur associé à leur couleur.

Pour pimenter le projet, un cahier des charges a été mis en place permettant ainsi de contraindre notre système sur son efficacité et sa rapidité à trier. En effet, nous imaginons un système renvoyant des informations sur le tri effectué via une interface avec au minimum 10 pièces par minute analysées avec une marge d'erreur d'une pièce sur 1000 pour la détection des couleurs.

Tout en sachant que ce produit pourra être amélioré via les pistes suivantes : distinction d'un plus grand nombre de couleurs ou tri selon les formes des pièces (cube, pavé, étoile, rond. . .) voir tri selon forme et couleur!

### 1.2 Chaîne fonctionnelle

La chaîne fonctionnelle de notre système se présente de la manière suivante :

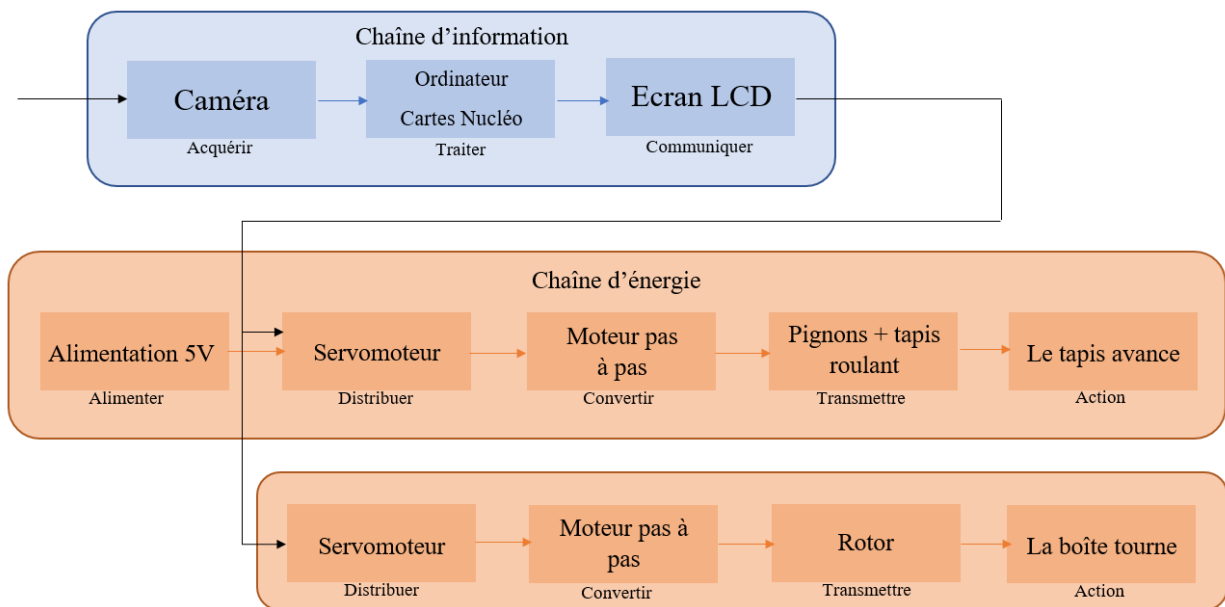


FIGURE 1 – Chaîne fonctionnelle

### 1.3 Planning prévisionnel

#### Séance 1 (11 mars)

- Mise en route du tapis roulant
- Prise en main des capteurs
- Test du capteur de présence
- Rédaction de la 1ère fonction d'interruption

#### Séance 2 (16 mars)

- Codage de la fonction et du pivot de tri
- Test du capteur de couleurs
- Interface Humain-Machine

#### Séance 3 (30 mars)

- Interface Humain-Machine
- Assemblage et test global

- Débogage
- Vérification du cahier des charges

**Séance 4 (06 avril)**

- Conception de la boîte
- Réflexions sur la détection de formes
- Création de nouvelles pièces de test
- Rédaction des livrables

**Séance 5 (08 avril)**

- Réalisation et test de détection de formes
- Préparation de la présentation finale
- Rédaction des livrables

**Séance 6 (13 avril)**

- Présentation finale

**1.4 *Planning réalisé*****Séance 1 (Jeudi 11 mars)**Pôle : Capteurs (Louis et Thomas)

1. Prise en main du capteur de position et du capteur de couleur : montage électrique, tests et codes ont été réalisés.
  - Le capteur de couleur ne captant que 3 couleurs par sa configuration, cela ne correspondait pas au cahier des charges.
  - Solution retenue : Utilisation d'une caméra remplaçant les deux types de capteurs.
2. Acquisition de la vidéo fournie par la caméra via le logiciel Matlab : installation de modules, suivi de tutoriels.

Pôle : Tapis roulant (Maëlys et Rayanne)

1. Mise en fonctionnement du tapis roulant via un servo-moteur : compréhension physique du fonctionnement du servo-moteur, codes rédigés sous MBed.
2. Contrôle du tapis roulant : sens de fonctionnement et choix des paramètres pour la vitesse afin de respecter le cahier des charges (le nombre de pièces analysées par minute par notre système).

**Séance 2 (Mardi 16 mars)**Pôle : Détection des couleurs (Louis et Thomas)

1. Caméra : implémentation de la liaison Matlab-Carte Nucléo et traitement vidéo sur Matlab pour obtenir la couleur du cube.
2. Intégration de la caméra et du code associé à celui du tapis roulant réalisé à la séance précédente.
  - De multiples tests avec des cubes de différentes couleurs ont été réalisés afin de paramétrer les valeurs nécessaires au traitement d'image pour obtenir la couleur du cube mais ces valeurs sont modifiées avec l'éclairage.
  - Solution retenue : Nécessité d'un éclairage constant pour ne pas avoir à modifier ces paramètres à chaque séance donc création d'une boîte avec un éclairage LEDs à l'intérieur avec rangement de la caméra pour qu'elle ait une position fixe.

Pôle : Boîte tournante de compartiments (Maëlys et Rayanne)

1. Faire tourner la boîte : mise en fonctionnement du moteur pas à pas pour faire tourner la boîte contenant les 4 compartiments (montage électrique et écriture du code sous MBed).
2. Création de la boîte à la découpe laser : choix des dimensions et réalisation.
  - Des tests ont été réalisés : on sait paramétrer le code MBed en fonction du nombre de tours à réaliser.

**Séance 3 (Mardi 30 mars)**Pôle : Eclairage constant (Louis et Rayanne)

1. Création d'un compartiment à la découpe laser : celui-ci peut être posé sur le tapis roulant, contient un emplacement pour y placer la caméra et un emplacement pour faire passer les fils électriques alimentant l'éclairage LED.
2. Choix et mise en place de l'éclairage LED au plafond de la boîte.

Pôle : Interface Humain-Machine (Maëlys et Rayanne)

1. Mise en place de l'interface avec écran LCD : montage électrique, écriture du code sous MBed.

- Il permet d'afficher le nombre de cubes analysés par la machine en fonction de sa couleur.

Pôle : Tri (Thomas)

1. Ecriture du programme général de notre système en fonction des codes écrits pour les différents modules.
  - Des tests ont été réalisés et notre système fonctionne sans erreur si on considère la situation suivante : quand la caméra déclare au programme qu'elle a détecté un cube de couleur (imaginons rouge), la boîte de compartiment se met à tourner pour mettre le bon comportement (ici rouge) à la fin du tapis roulant.
2. Intégration de la nouvelle boîte pour éclairage constant.
  - Des tests ont été réalisés pour modifier les paramètres de lecture vidéo de la caméra car il arrive 1 fois sur 4 qu'un cube rouge soit capté comme un cube jaune et inversement par la caméra. En revanche, il y a aucun problème avec les cubes de couleur bleu et vert.
  - Solution retenue : changer la disposition de l'éclairage LED et le mettre latéralement plutôt que frontalement pour permettre une meilleure détection des couleurs. Prendre en compte l'effet du poids qui entraîne une erreur lors de la rotation de la boîte après 5 cubes.

**Séance 4 (Mardi 06 avril)**

Pôle : Rédaction des livrables (Rayanne et Maëlys)

1. Rédaction sur le livrable technique : partie électrique et mécanique (schémas électriques, explications, photos ...)

Pôle : Tri (Thomas et Louis)

1. Disposition finale du système : disposition latérale de l'éclairage LED dans la boîte posée sur le tapis et mise à jour des coefficients sur le programme final pour que la bonne couleur soit détectée. Cela a bien fonctionné!
2. Prise en compte du poids qui se rajoute avec les cubes au fur et à mesure : jusqu'à 40 pièces, aucune erreur!
3. Test global : aucune erreur sur toutes les pièces disponibles et performance de 14 pièces triées en 1 minute.

## 2 Partie électronique

### 2.1 Tapis roulant

#### Liste du matériel et logiciel

- Tapis roulant du Conveyor Belt DEMO;
- Servo-moteur ;
- Carte Nucléo (référence : L476RG);
- Ordinateur + MBed;
- Fils électriques.

#### Montage électrique

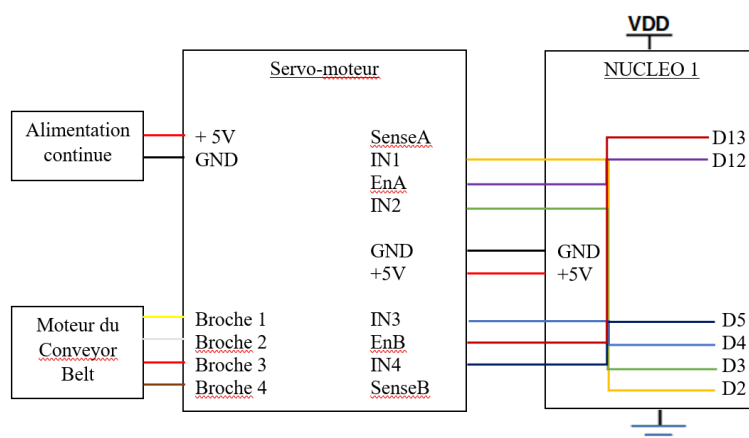


FIGURE 2 – Schéma du montage électrique - fonctionnement du moteur

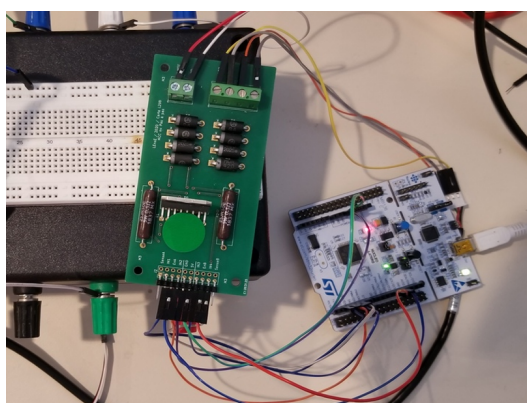


FIGURE 3 – Photo du montage électrique associé

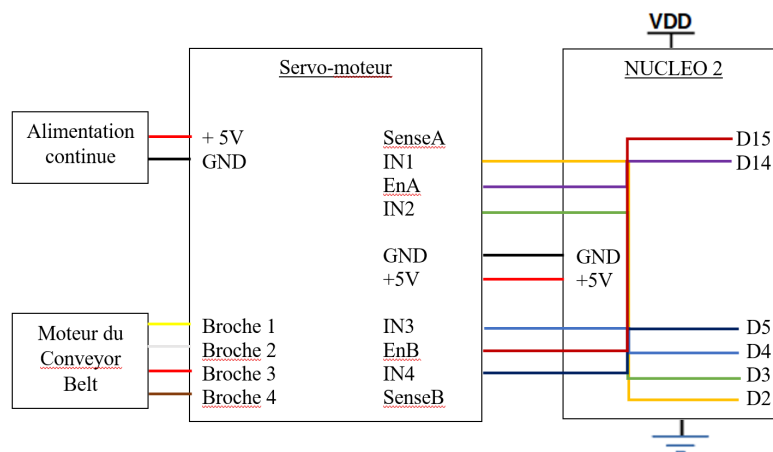


## 2.2 Boîte tournante

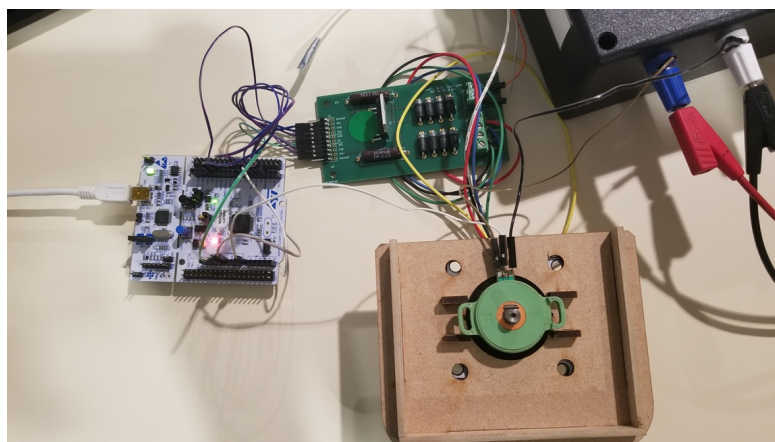
### Liste du matériel et logiciel

- Moteur pas-à-pas (100 pas - 12V - 0.9A/phase);
- Servo-moteur;
- Carte Nucléo (référence : L476RG);
- Ordinateur + MBed;
- Fils électriques.

### Montage électrique



**FIGURE 4** – Schéma du montage électrique permettant le fonctionnement du moteur



**FIGURE 5** – Photo du montage électrique associé

## 2.3 Ecran LCD

### Liste du matériel et logiciel

- Ecran LCD;
- Carte Nucléo (référence : L476RG);
- Ordinateur + Mbed;
- Fils électriques.

### Montage électrique

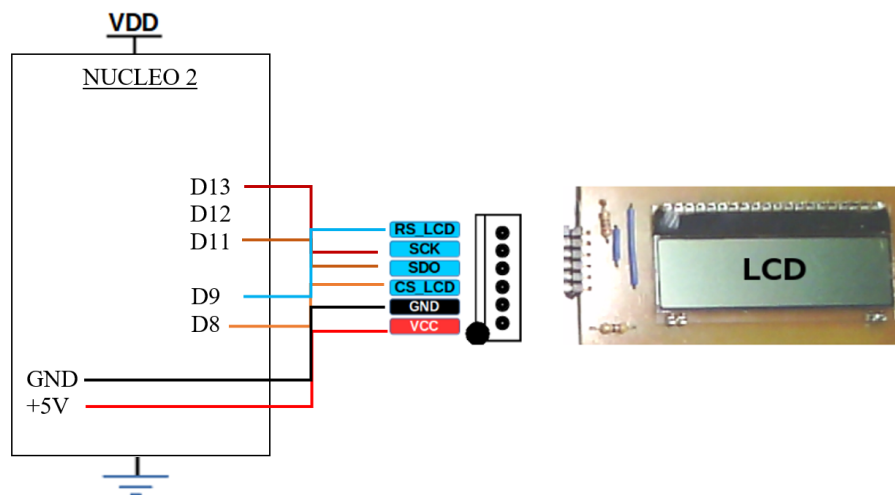


FIGURE 6 – Schéma du montage électrique permettant le fonctionnement du moteur

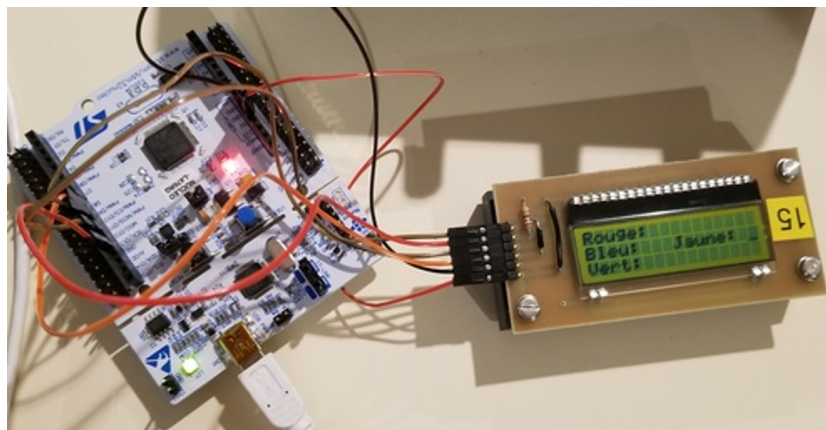


FIGURE 7 – Photo du montage électrique associé

### 3 Partie mécanique

#### 3.1 Boîte à compartiments

Nous avons construit à la découpe laser une boîte de quatre compartiments qui est un pavé droit les dimensions sont de 16 cm de longueur et largeur et 10 cm de hauteur. Le schéma de la découpe (cf figure 8) ainsi que les photos de la boîte obtenue (cf figures 9 et 10) sont détaillés ci-dessous :

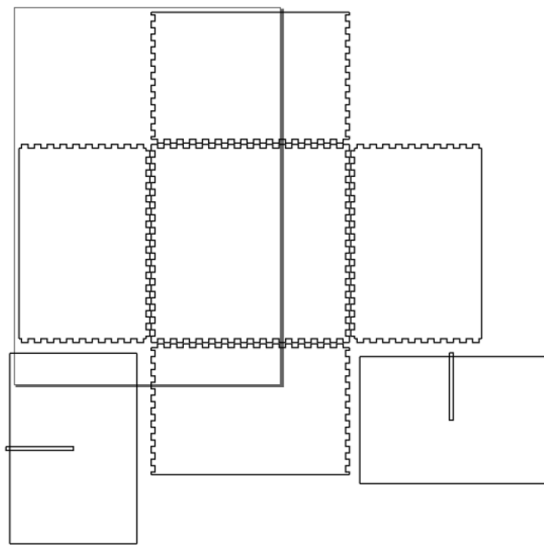


FIGURE 8 – Plan de la boîte pour la découpe laser



FIGURE 9 – Boîte finale - vue de dessus

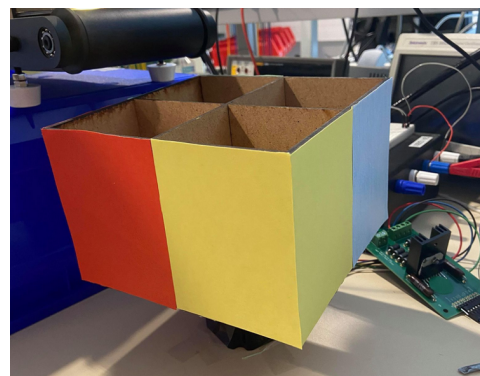


FIGURE 10 – Boîte finale - vue globale

### 3.2 Boîte à éclairage constant

Nous avons construit à la découpe laser une boîte permettant de fixer la caméra et d'installer un éclairage constant (ampoule LED, référence DS-G4-12LT-NW, alimentation 12V), permettant une meilleure détection des couleurs, ainsi qu'une indépendance de l'éclairage extérieur. Les plans (cf figure 11) ainsi que le résultat final obtenus (cf figure 12) sont détaillés ci-dessous :

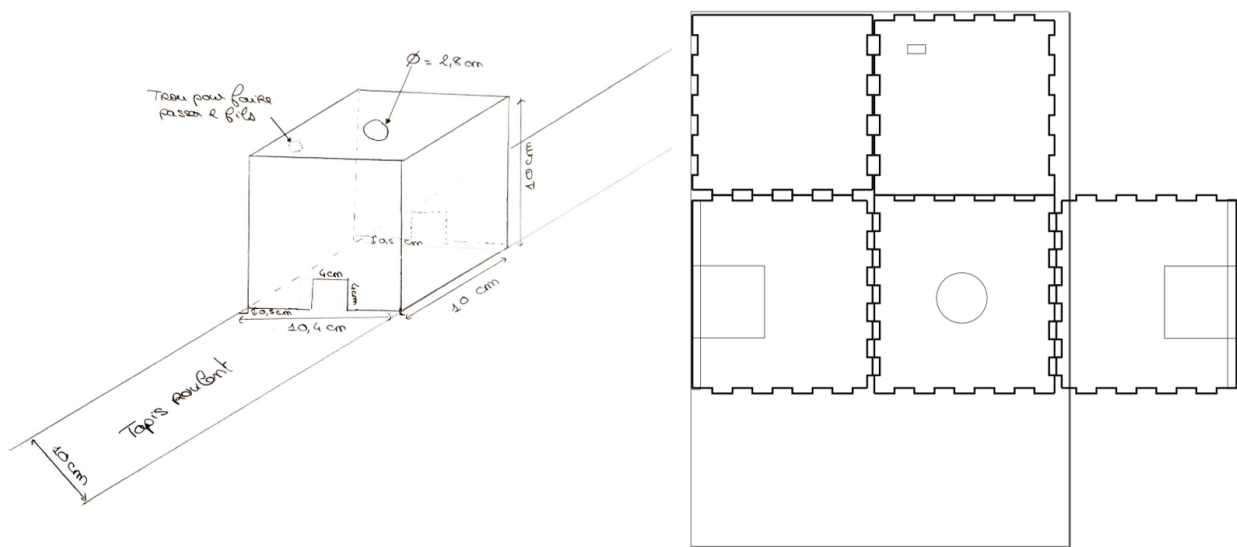


FIGURE 11 – Plan de la boîte pour la découpe laser

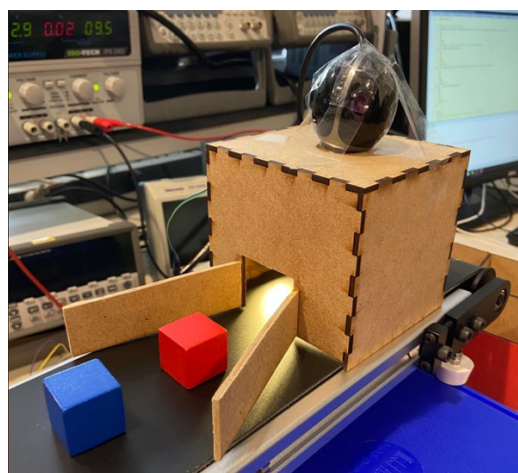


FIGURE 12 – Boîte à éclairage constant finale

## 4 Partie informatique

Pour faire fonctionner notre système, trois programmes distincts sont nécessaires.

Le premier (disponible [ici](#)) sert à faire fonctionner le tapis roulant au moyen d'une première carte Nucléo (Nucléo 1).

Le deuxième programme (disponible [ici](#)), indépendant de ce dernier, est en charge de faire fonctionner le tri, la boîte tournante, l'écran LCD et les compteurs. Il est nécessaire que ce programme soit envoyé sur une carte Nucléo différente (Nucléo 2), afin d'avoir assez de ports pour effectuer tous les branchements.

Enfin, en parallèle, un code (disponible [ici](#)) tourne sur Matlab pour permettre le traitement des images obtenues avec la caméra. Trois filtres colorés sont appliqués à l'image, qui est ensuite moyennée sur tous ses pixels, ce qui permet de déterminer la couleur dominante sur l'image, en fonction des moyennes obtenues sur les composantes RGB. L'information de couleur est ensuite renvoyée sur la Nucléo 2, où se trouve l'algorithme de tri. En fonction de la couleur reçue et de la position actuelle du compartiment, la boîte tournante reçoit une instruction de rotation adaptée.

**Remarque :** Pour utiliser l'algorithme sur Matlab, il est nécessaire d'avoir préalablement installé l'add-on « MATLAB Support Package for USB Webcams » qui permet l'acquisition de la vidéo et son traitement.

## 5 Résultats et analyse

Notre système de convoyeur-trieur est capable de trier 14 pièces par minute ce qui rentre bien dans notre cahier de charge, qui demandait un tri d'au minimum 10 pièces par minute. Nos pièces sont espacées d'environ 4 cm. De plus, notre système a été capable de ranger 40 pièces sans aucune erreur. Par ailleurs, un écran LCD a été installé afin de permettre une lecture du nombre de pièces triées par couleur.

Ainsi, notre système répond au cahier des charges.

Cependant, il arrive parfois qu'un cube tombe sur une des tranches de la boîte du fait de l'ajustement de la rotation de la boîte tournante avec le poids des cubes. Un asservissement de la rotation pourrait pallier ce problème.

Nous avons également pensé à des améliorations de notre système. D'une part, une nouvelle boîte à éclairage pourrait être créée de façon à fixer la caméra de manière permanente sans que l'on ait à utiliser du scotch pour cela. Cette boîte contiendrait deux planches permettant de guider la trajectoire du cube au sein de la boîte en plus d'améliorer la détection numérique des couleurs.

Enfin, pour la partie informatique, une mémoire enregistrant l'ordre des pièces de couleur pourrait être mise en place dans le cas où les pièces sont collées les unes aux autres avec un système de barrière permettant de les verser une à une.

## 6 Bilan - Retour d'expérience

Pour conclure, nous pouvons affirmer avec fierté que nous avons mené à bien ce projet. Ce dernier fut très agréable à effectuer et été adapté à notre expertise d'élèves en 2ème année. Les missions de recrutement proposées par SOLEC étaient en accord avec les connaissances requises pour aborder ce projet, qui nous a également permis de développer de nouvelles compétences.

Tout d'abord, nous avons acquis des compétences techniques. En électronique et programmation, on peut citer par exemple : le contrôle d'un moteur pas-à-pas, le codage d'une interface humain-machine, l'acquisition et l'analyse d'images ainsi que l'envoi des données via une liaison Matlab-Nucléo.

Ce projet nous a également été formateur concernant des compétences transverses.

En effet, nous avons par exemple dû rechercher des informations au sein des datasheets constructeur.

Ensuite, nous avons pu développer notre capacité d'adaptation. Comme dans tout projet, il nous a fallu faire face à des problèmes non attendus, et donc réagir rapidement afin de trouver une solution fonctionnelle tout en respectant les délais fixés par notre planning prévisionnel.

Cela a été rendu possible par un travail d'équipe : nous travaillions la plu-

part du temps en binôme, afin de toujours confronter plusieurs points de vue, et nous avons appris à appliquer une communication claire et régulière entre nous quatre.

Enfin, le suivi régulier de notre projet ainsi que la rédaction des différents livrables nous ont permis de développer une rigueur organisationnelle et rédactionnelle.

Nous remercions l'équipe de SOLEC pour leur aide et leur bienveillance, et espérons réitérer notre collaboration très vite.

## Annexe

### Programme n°1 : Tapis roulant

```
1
2 //   Nucléo responsable du tapis roulant
3 //   * Auteurs : Maëlys ALQUIER, Louis FOLLET, Rayanne MARTINY et
   Thomas TROCELLO
4 //   * Date : 11 mars 2021, Dernière modification : 11 mars 2021
5
6
7 /* Déclaration des ressources externes */
8 #include "mbed.h"
9
10 /* Déclaration des entrées/ sorties */
11
12 DigitalOut In1(D2); /* broche In1 du servomoteur : 1er pont H */
13 DigitalOut In2(D3); /* broche In2 du servomoteur : 1er pont H */
14 DigitalOut In3(D4); /* broche In3 du servomoteur : 2e pont H */
15 DigitalOut In4(D5); /* broche In4 du servomoteur : 2e pont H */
16
17 DigitalOut Enable1(D13); /* broche EnableA du servomoteur : 1er pont H
   */
18 DigitalOut Enable2(D12); /* broche EnableB du servomoteur : 2ème pont
   H */
19
20 Ticker toggle_int_ticker; /* Pour réaliser des interruptions */
21 void toggle_int(void);
22 /* Fonction principale */
23
24 int main(){
25
26
27     toggle_int_ticker.attach(&toggle_int, 0.00001);
28
29     while(1){}
30 }
31
32
33 void toggle_int(){
34     Enable1=1; /* Initialisation du 1er pont H */
```



```
35     Enable2=1; /* Initialisation du 2ème pont H */
36
37     int temps=2;
38
39     /* Marche Avant : 50 s pour 10 pièces espacées de 4-5 cm */
40
41     /* position initiale : 1/8 */
42
43     In1 = 0;
44     In2 = 0;
45     In3 = 1;
46     In4 = 0;
47     wait_ms(temps);
48
49     /* position suivante : 2/8 */
50
51     In1 = 0;
52     In2 = 1;
53     In3 = 1;
54     In4 = 0;
55     wait_ms(temps);
56
57     /* position suivante : 3/8 */
58
59     In1 = 0;
60     In2 = 1;
61     In3 = 0;
62     In4 = 0;
63     wait_ms(temps);
64
65     /* position suivante : 4/8 */
66
67     In1 = 0;
68     In2 = 1;
69     In3 = 0;
70     In4 = 1;
71     wait_ms(temps);
72
```

```
73  /* position suivante : 5/8 */
74
75  In1 = 0;
76  In2 = 0;
77  In3 = 0;
78  In4 = 1;
79  wait_ms(temps);
80
81  /* position suivante : 6/8 */
82
83  In1 = 1;
84  In2 = 0;
85  In3 = 0;
86  In4 = 1;
87  wait_ms(temps);
88
89  /* position suivante : 7/8 */
90
91  In1 = 1;
92  In2 = 0;
93  In3 = 0;
94  In4 = 0;
95  wait_ms(temps);
96
97  /* position suivante : 8/8 */
98
99  In1 = 1;
100 In2 = 0;
101 In3 = 1;
102 In4 = 0;
103 wait_ms(temps);
104
105
106 }
107
```

## Programme n°2 : Boîte de tri et Interface Humain-Machine

```

1
2 // * Programme final : Nucléo responsable de la boîte tournante, du tri
  et des compteurs, ainsi que de l'écran LCD
3
4 // * Auteurs : Maëlys ALQUIER, Louis FOLLET, Rayanne MARTINY et
  Thomas TROCELLO
5 // * Date : 11 mars 2021
6 // * Dernière modification : 08 avril 2021
7
8
9
10 /* ---Déclaration des ressources externes --- */
11
12 #include "mbed.h"
13
14 /* ---Déclaration des entrées/ sorties --- */
15
16 DigitalOut In1(D2); /* broche In1 du servomoteur : 1er pont H */
17 DigitalOut In2(D3); /* broche In2 du servomoteur : 1er pont H */
18 DigitalOut In3(D4); /* broche In3 du servomoteur : 2e pont H */
19 DigitalOut In4(D5); /* broche In4 du servomoteur : 2e pont H */
20
21 DigitalOut Enable1(D14); /* broche EnableA du servomoteur : 1er pont H
  */
22 DigitalOut Enable2(D15); /* broche EnableB du servomoteur : 2ème pont
  H */
23
24 SPI spi(D11,D12,D13); // mosi, miso, sck
25 DigitalOut CS_LCD(D8); // LCD
26 DigitalOut RS_LCD(D9); // LCD
27
28 /* ---Déclaration des fonctions --- */
29
30 void Tourne(int tour); // Fonction de rotation de la boîte
31 void Tourne_inverse(int tour); // Fonction de rotation inverse de la bo
  îte 32
33 void LCD_DOG_init(void); // Fonction d' initialisation de l'écran LCD

```

```
34 void LCD_DOG_setPosition(char ligne, char colonne); //Fonction de d
    éplacement de curseur
35 void LCD_DOG_writeStr(char *c, char ligne, char colonne); //
    Fonction d'écriture de caractère
36 void LCD_DOG_writeChar(char val); //Fonction d'écriture d'une chaîne
    de caractères
37 void LCD_DOG_clear(void); //Fonction pour effacer l'écran
38
39 Serial Matlab(USBTX, USBRX); //Liaison Nucléo-Matlab
40
41 /* ---Fonction principale --- */
42
43 int main(){
44     LCD_DOG_init(); // Initialisation de l'écran LCD
45     int i;
46     char ch[10];
47     spi.format(8,0);
48     spi.frequency(100000);
49     RS_LCD = 1;
50     CS_LCD = 1;
51     LCD_DOG_init();
52
53     Matlab.baud(112500); // Même valeur à rentrer dans Matlab
54
55     int C0=1; //Le compartiment rouge fait face au tapis
56     int s=0; // Variable indiquant le déplacement à faire entre la
    position actuelle et celle souhaitée
57     int tour; // Variable des fonctions de rotation de la boîte tournante
58     int r=0,v=0,b=0,j=0; //Compteurs de couleur
59     int k=0,d=0; // Variable venant accélérer la boîte dès qu'elle s'
    alourdit
60
61     while(1){
62         int C= Matlab.getc(); //Acquisition de la "valeur de couleur" de
    Matlab
63
64         if (C==1){r+=1;} //Le 1 équivaut au rouge
65         if (C==2){v+=1;} //Le 2 équivaut au vert
66         if (C==3){b+=1;} //Le 3 équivaut au bleu
67         if (C==4){j+=1;} //Le 4 équivaut au jaune
```

```
68
69     if (C!=0){ //Une couleur est détectée
70         if ((C-C0>0)|| (C-C0==0)){
71             s=(C-C0)%4;} // Calcul du déplacement s à faire , sachant la
// position actuelle C0 de la boîte et la couleur C souhaitée
72         if (C-C0<0){
73             s=4-(C0-C)%4;} //Même chose dans le cas où le compartiment
// est en amont
74
75
76         k=k+3; // Au bout de 3 cubes ...
77         d=k/9; // ... la variable de rotation change
78
79         tour=12*s+d; // Variable de rotation + prise en compte de l'
// alourdissement
80         if (d==2){k=k-15;} //Régulation sur la modification de vitesse
81
82         C0=C; // Nouvelle valeur indiquant la position actuelle de la
// boîte
83
84         if (s==3){Tourne_inverse(12);} //Fonction permettant d'éviter
// le trois quarts de tour
85
86         else{Tourne(tour);} /*Fonction de rotation de la boîte */
87
88         i++; /* Fonctions d'affichage des compteurs */
89
90         sprintf(ch, "Rouge:%d",r);
91         LCD_DOG_writeStr(ch, 1, 1);
92         sprintf(ch, "Bleu:%d",b);
93         LCD_DOG_writeStr(ch, 2, 1);
94         sprintf(ch, "Vert:%d",v);
95         LCD_DOG_writeStr(ch, 3, 1);
96         sprintf(ch, "Jaune:%d",j);
97         LCD_DOG_writeStr(ch, 2, 9);
98         wait(1.2); //Fréquence entre deux rotations de boîte
99         LCD_DOG_clear();
100     }
101 }
102
103
```

```
104 void Tourne(int tour){
105
106     int i;
107
108     for (i=0 ; i<tour ; i++)
109
110         // i=12 : un quart de tour ;
111         // i=25 : un demi tour ;
112         // i=37 : trois quarts de tour;
113         // i=50 : un tour complet :
114
115
116     {
117
118         Enable1=1; /* Initialisation du 1er pont H */
119         Enable2=1; /* Initialisation du 2ème pont H */
120
121         int temps=5; //Impact sur la vitesse de la boîte
122
123         /* Marche Avant : 60 s pour 15 pièces espacées de 3 cm */
124
125         /* position initiale : 1/8 */
126
127         In1 = 0;
128         In2 = 0;
129         In3 = 1;
130         In4 = 0;
131         wait_ms(temps);
132
133         /* position suivante : 2/8 */
134
135         In1 = 0;
136         In2 = 1;
137         In3 = 1;
138         In4 = 0;
139         wait_ms(temps);
140
141         /* position suivante : 3/8 */
142
143         In1 = 0;
```

```
144     In2 = 1;
145     In3 = 0;
146     In4 = 0;
147     wait_ms(temps);
148
149 /* position suivante : 4/8 */
150
151     In1 = 0;
152     In2 = 1;
153     In3 = 0;
154     In4 = 1;
155     wait_ms(temps);
156
157
158 /* position suivante : 5/8 */
159
160     In1 = 0;
161     In2 = 0;
162     In3 = 0;
163     In4 = 1;
164     wait_ms(temps);
165
166
167 /* position suivante : 6/8 */
168
169
170     In1 = 1;
171     In2 = 0;
172     In3 = 0;
173     In4 = 1;
174     wait_ms(temps);
175
176
177 /* position suivante : 7/8 */
178
179
180     In1 = 1;
181     In2 = 0;
182     In3 = 0;
183     In4 = 0;
184     wait_ms(temps);
```

```
185
186
187
188 /* position suivante : 8/8 */
189
190
191     In1 = 1;
192     In2 = 0;
193     In3 = 1;
194     In4 = 0;
195     wait_ms(temps);
196 }
197 }
198 void Tourne_inverse(int tour){
199
200     int i;
201
202     for (i=0 ; i<tour ; i++)
203
204         //i=12 : un quart de tour ;
205         //i=25 : un demi tour ;
206         //i=37 : trois quarts de tour ;
207         //i=50 : un tour complet :
208
209
210 {
211
212     Enable1=1; /* Initialisation du 1er pont H */
213     Enable2=1; /* Initialisation du 2ème pont H */
214
215     int temps=5;
216
217 /* Marche Avant : 50 s pour 10 pièces espacées de 4–5 cm */
218
219 /* position initiale : 1/8 */
220
221     In1 = 1;
222     In2 = 0;
223     In3 = 0;
224     In4 = 0;
225     wait_ms(temps);
```



```
226
227 /* position suivante : 2/8 */
228
229     In1 = 1;
230     In2 = 0;
231     In3 = 0;
232     In4 = 1;
233     wait_ms(temps);
234
235 /* position suivante : 3/8 */
236
237     In1 = 0;
238     In2 = 0;
239     In3 = 0;
240     In4 = 1;
241     wait_ms(temps);
242
243 /* position suivante : 4/8 */
244
245     In1 = 0;
246     In2 = 1;
247     In3 = 0;
248     In4 = 1;
249     wait_ms(temps);
250
251
252 /* position suivante : 5/8 */
253
254     In1 = 0;
255     In2 = 1;
256     In3 = 0;
257     In4 = 0;
258     wait_ms(temps);
259
260
261 /* position suivante : 6/8 */
262
263
264     In1 = 0;
265     In2 = 1;
266     In3 = 1;
```

```
267     In4 = 0;
268     wait_ms(temps);
269
270
271     /* position suivante : 7/8 */
272
273
274     In1 = 0;
275     In2 = 0;
276     In3 = 1;
277     In4 = 0;
278     wait_ms(temps);
279
280
281     /* position suivante : 8/8 */
282
283
284     In1 = 1;
285     In2 = 0;
286     In3 = 1;
287     In4 = 0;
288     wait_ms(temps);
289 }
290 }
291
292 /* Initialisation de l'écran */
293
294 void LCD_DOG_init(void){
295     CS_LCD = 0;
296     RS_LCD = 0;
297     spi.write(0x29); //Function Set - Table 1
298     spi.write(0x1D); //Bias Set
299     spi.write(0x50); //Power Control
300     spi.write(0x6C); //Follower Control
301     spi.write(0x7C); //Contrast Set
302     spi.write(0x03); //Function Set - Table 0
303     spi.write(0x0F); //Display On
304     spi.write(0x01); //Clear Display
305     wait_ms(2);
306     spi.write(0x06); //Cursor On
```

```
307     wait_ms(10);
308     RS_LCD=1;
309     CS_LCD=1;
310     wait_ms(10);
311 }
312
313 /* Déplacer le curseur */
314
315 void LCD_DOG_setPosition(char ligne, char colonne){
316     char address = 0x80 + ((ligne - 1) * 16) + (colonne - 1);
317     CS_LCD = 0;
318     RS_LCD = 0;
319     spi.write(address);
320     CS_LCD=1;
321     RS_LCD = 1;
322 }
323
324 /* Ecriture d'un caractère à l'écran */
325
326 void LCD_DOG_writeChar(char val){
327     CS_LCD = 0;
328     spi.write(val);
329     CS_LCD=1;
330 }
331
332
333
334 /* Pour écrire une chaîne de caractères */
335
336 void LCD_DOG_writeStr(char *c, char ligne, char colonne){
337     char i = 0;
338     LCD_DOG_setPosition(ligne,colonne);
339     while(c[i] != '\0') {
340         LCD_DOG_writeChar(c[i]);
341         i++;
342     }
343 }
```

```
344
345
346 /* Pour effacer l'écran */
347
348 void LCD_DOG_clear(void){
349     CS_LCD = 0;
350     RS_LCD = 0;
351     spi.write(0x01); //Clear Display
352     wait_ms(2); //delay 2 ms
353     RS_LCD = 1;
354     CS_LCD = 1;
355 }
```

## Programme n°3 : Détection des couleurs

```
%% Code Matlab pour la détection des couleurs
%% Auteurs : Maëlys ALQUIER, Louis FOLLET, Rayanne MARTINY et Thomas
TROCELLO

1 close all; clc;
2
3 %% Ouverture du port série
4
5 cam = webcam(2) % Utiliser webcamlist pour trouver le chiffre adéquat
6 preview(cam); % Aperçu de la zone filmée
7 s = serialport("COM8",112500); % Valeur de 112 500 bauds, utiliser
serialportlist pour la valeur de COM
8 %% Détection de la couleur dominante de l'image
9
10 while 1
11
12     C=0;
13     % Obtenir l'image actuelle
14     img=snapshot(cam); % Photo de la zone filmée
15     im_rgb0=imresize(img,0.99); % Réduction de 99%
16     im_rgb= imcrop(im_rgb0,[200 150 200 200]); % Image rognée
17
18     % Récupérer les composantes R, G et B
19     im_R=im_rgb(:,:,1);
20     im_G=im_rgb(:,:,2);
21     im_B=im_rgb(:,:,3);
22
23     R=mean(mean(im_R));
24     G=mean(mean(im_G)); % On récupère la valeur moyenne des pixels
25     B=mean(mean(im_B));
26
27 % [a,b,c]=rgb2hsv(im_rgb); % HSV
28 % m=360*a;
29 % d=mean(mean(m));
30 % n=mean(mean(c));
```

```
31
32     % Identifier la couleur détectée
33     if ((R>B)&&(G>B)&&(R>160)&&(G>160)&&((R-G)<30)&&(R-B>20)&&(G-
34         B>20)) % A changer en fonction de la luminosité
35         R
36         G
37         B
38         C=[4]
39         pause(1.3) % Pour ne pas détecter 2 fois la même pièce
40     elseif ((R>B)&&(R>G)&&(R>180)&&((R-B)>20)&&((R-G)>20))
41         R
42         G
43         B
44         C=[1]
45         pause(1.3)
46     elseif ((G>B)&&(G>R)&&(G>110)&&((G-R)>10)&&((G-B)>10))
47         R
48         G
49         B
50         C=[2]
51         pause(1.3)
52     elseif ((B>R)&&(B>G)&&(B>115)&&((B-R)>15)&&((B-G)>15))
53         R
54         G
55         B
56         C=[3]
57         pause(1.3)
58     end
59
60     % Transfert de la valeur vers la carte Nucléo
61     write(s,C,"uint8")
62
63     pause(0.05) % Période d'acquisition des images
64 end
```

```
65 %% Libération des ports
66
67 clear cam
68 clear s
69 % A exécuter pour libérer les ports
```

---

Institut d'Optique Graduate School  
2 avenue Augustin Fresnel  
91127 Palaiseau