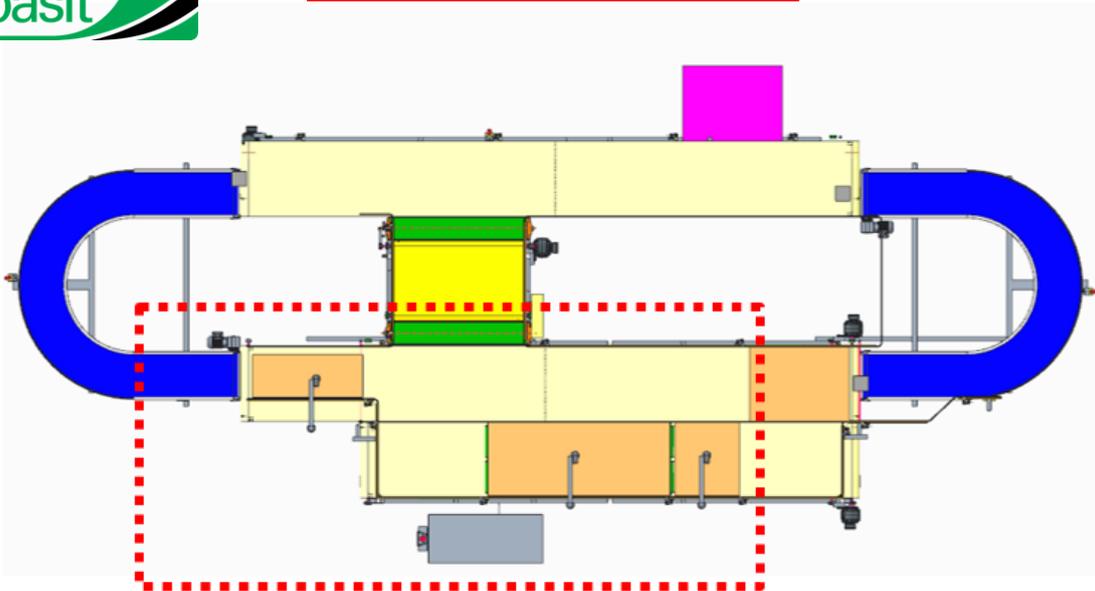




RAPPORT DE PROJET



Identification et mesure de position d'un produit
sur bandes roulantes

Sujet proposé par :
Habasit

Diriger par :
Dietmar Elsner
Frédéric Siegrist
4153 Reinach Switzerland

Nom des étudiants :
METZLER Noa
DAGHOUR Yann
SAUTTER Lukas

Session 2023/2024

SOMMAIRE (partie commune)

Table des matières

1.	REMERCIEMENTS (partie commune)	3
2.	PRÉSENTATION DU PROJET (partie commune)	4
1.	A.1 Cahier des charges (partie commune).....	4
2.	A.2 Analyse du cahier de charges - Principes mis en œuvre (partie commune).....	5
3.	A.3:Analyse fonctionnelle du système	7
3.	B.Détection du carton et modélisation du support caméra (Partie de Noa)	8
1.	B.1 : Définir le champ observable de la caméra.....	8
2.	B.2 : création du support pour la caméra.....	12
3.	B.3: Création du programme de détection du carton.....	18
4.	C) Partie éclairage (Partie de Lukas)	20
1.	Choix de la position de l'éclairage :	20
2.	Choix de l'éclairage.....	24
3.	CONDITIONS D'UTILISATIONS :	28
4.	Protocole éclairage :.....	28
5.	NETTOYAGE (PRODUIT HORS TENSION).....	28
5.	D. Partie programmation (Partie de Yann).....	29
6.	E.Déroulement des travaux. (partie commune)	40
7.	F. Conclusion (partie commune)	41

1. REMERCIEMENTS (partie commune)

Nous remercions Mr Frédéric Siegrist de l'entreprise Habasit de nous avoir confié ce projet et de nous avoir suivi tout au long du projet.

Nous tenions à remercier les professeurs du BTS systèmes photoniques pour leurs dévouements et leurs confiances.

Nous remercions Mr Sigwarth pour son aide apportée pour ses conseils sur la partie théorie et pratique pour la partie des phénomènes physiques.

Nous remercions Mr Siegrist pour son aide apportée dans la partie électronique avec l'éclairage.

Nous remercions Mme Bareux pour son aide dans la gestion du projet et pour la partie mécanique du projet.

2. PRÉSENTATION DU PROJET (partie commune)

1. A.1 Cahier des charges (partie commune)

	Noa	Lukas	Yann
Prise en main du sujet et test de faisabilité	X	X	X
Elaboration du diagramme de blocs du système	X	X	X
Elaboration du diagramme de blocs internes du système			X
Caractériser la zone observable de caméra sick	X		
Choisir l'éclairage optimal (couleur, inclinaison)		X	
Modéliser le support d'éclairage		X	
Modéliser le support camera	X		
Modéliser le système complet			X
Traitement d'image détecter carton	X		
Traitement d'image coordonnées x,y d'un point du paquet			X
Traitement d'image écart angulaire par rapport à l'axe de translation de la bande			X
Réaliser le support caméra	X		
Réaliser le support éclairage		X	
Assembler tous les éléments			X
Etude de coût	X	X	X
Notice d'utilisation	X	X	X
Validation du système	X	X	X

2. A.2 Analyse du cahier de charges - Principes mis en œuvre (partie commune)

Analyse de la position d'un produit sur la bande roulante avec sphères :

Les sphères sur cette bande permettent de modifier l'orientation d'un produit qui s'y trouve pendant son déplacement.

L'objectif est de définir une liste de paramètres permettant la rotation d'un produit jusqu'à ce qu'il soit bien orienté (maximum 90°).

Le premier objectif sera de trouver un moyen d'identifier **la forme extérieure du paquet** (carton brun) ainsi que sa position sur la bande (en XY + orientation).

Les étudiants pourront suggérer les points de référence pour la position, mais nous suggérons le côté du tapis roulant pour l'orientation et le centre du paquet.

Idéalement, il faut faire un suivi en temps réel et indiquer le temps de suivi pour une position donnée de la caméra.

Le temps de suivi doit être suffisant pour observer le changement d'orientation du paquet. La vitesse maximale du paquet est de 40 mètres par minute.

La précision de la position doit être de 10 mm pour les paquets de 400 mm de côté.

Une meilleure précision est attendue pour de plus petits paquets. (Pas de précision sur l'angle d'orientation)

Le programme doit donner la position X et Y d'un point du paquet et son angle d'orientation par rapport à une direction de référence.

Les résultats seront utilisés pour faire tourner le paquet dans une direction donnée (petit ou grand côté parallèle au côté du tapis roulant).

Le type d'éclairage et la structure le portant (tunnel, porte, etc...) devront être définis.

Le système doit fonctionner pour toutes les couleurs de cartons et particulièrement la couleur brune.

Le projet consiste à détecter des boîtes en carton se déplaçant sur une bande transporteuse. On souhaite récupérer les coordonnées du carton ainsi que son orientation.

Pour cela il faut mettre en place un système de visualisation optique, éclairer la scène de manière optimale tout en respectant la sécurité des ouvriers.

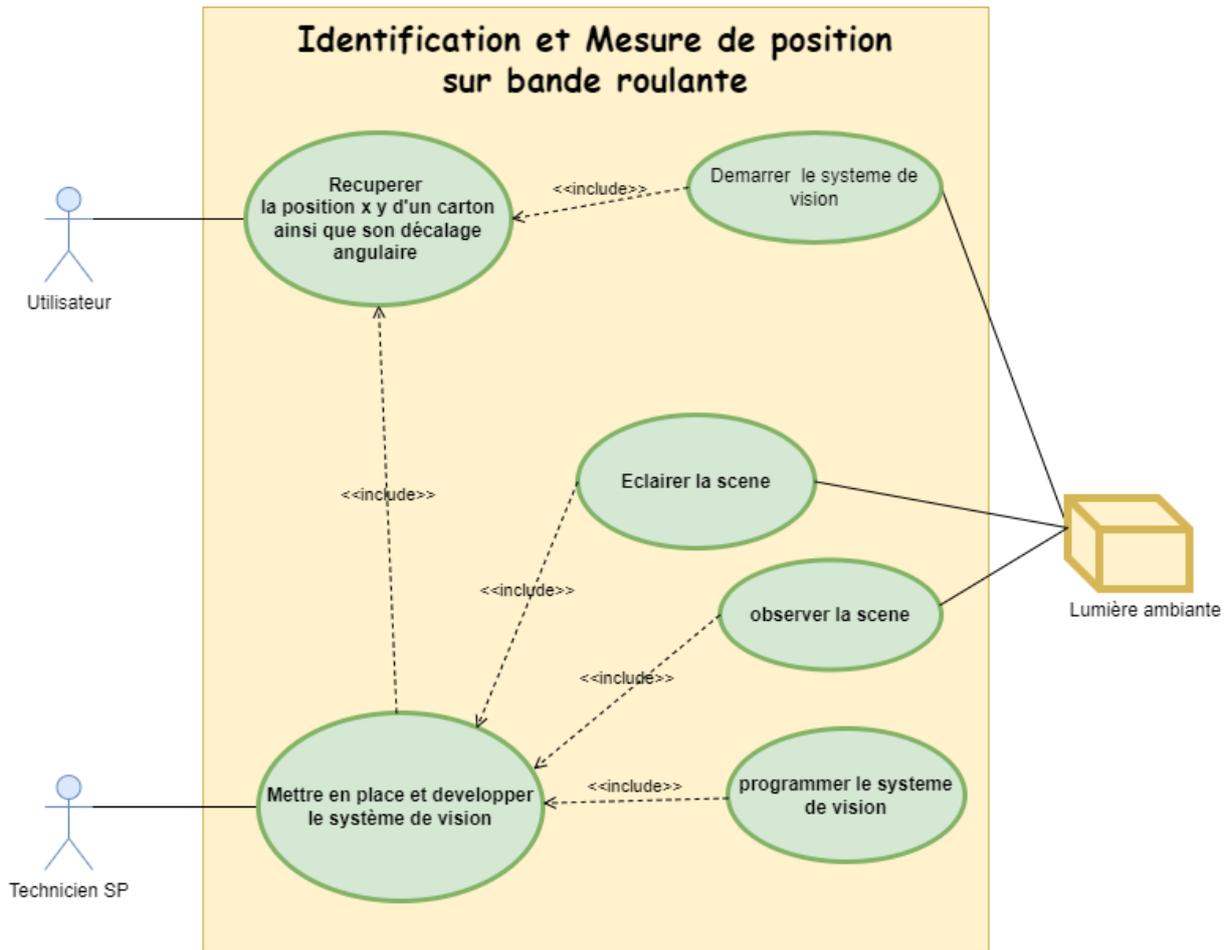
Pour mettre en place le système il nous faut un support mécanique pour la caméra ainsi que l'éclairage, il faut aussi prendre en compte le bruit lumineux extérieur, il faut prendre en compte le champ observable par la caméra pour savoir où la positionner en termes de hauteur et de la profondeur de champ.

Afin de déterminer cette hauteur, nous avons utilisé les lois trigonométriques pour avoir l'angle de champ de la caméra.

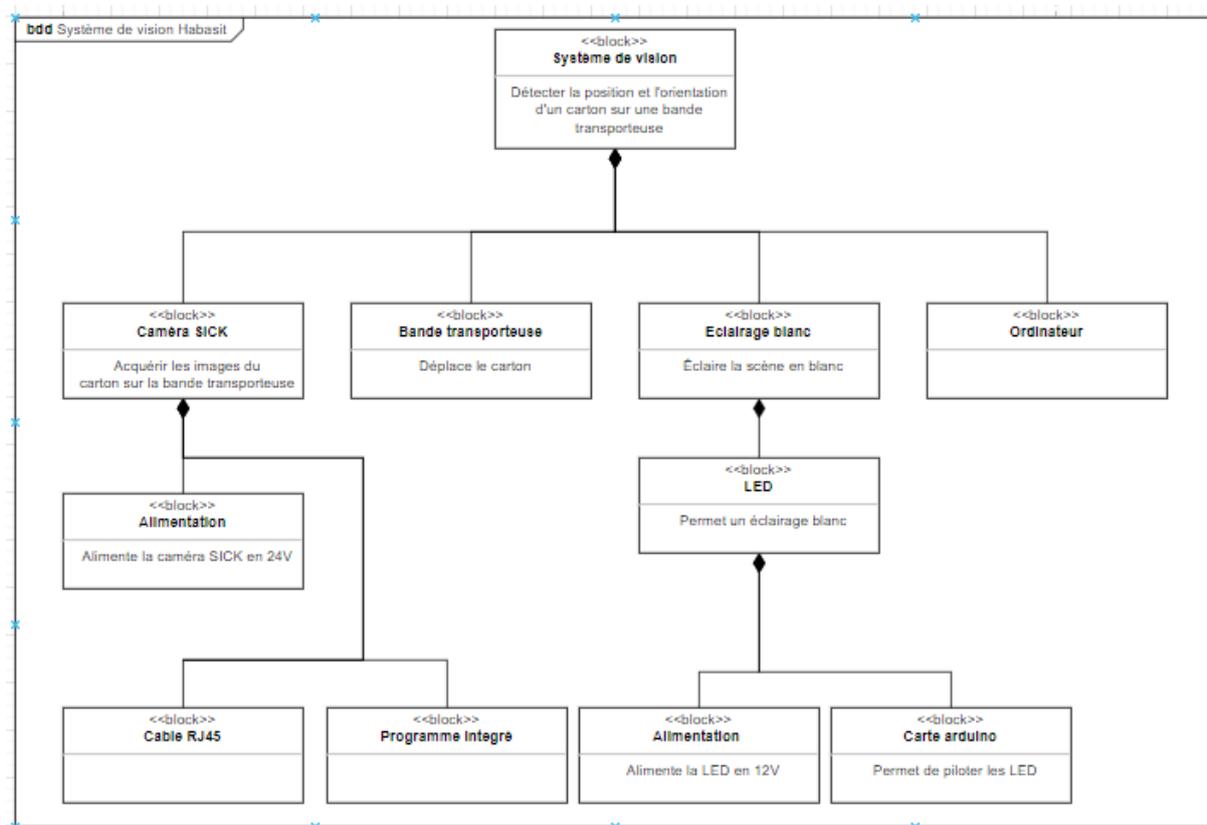
Dans le but d'obtenir un éclairage optimal, nous avons utilisé les lois de Beer-Lambert afin de déterminer la couleur optimale.

Pour permettre la détection du carton, il faut créer un programme qui permettrait la détection de celle-ci en moins de 300ms. Le logiciel de la caméra est couplé à un programme PYTHON, le logiciel permet la détection du carton grâce aux outils du logiciel.

Dans tous les points cités, il faut assurer la sécurité de l'utilisateur du point de vue mécanique, optique et électrique.



3. A.3:Analyse fonctionnelle du système



Le système de vision utilisé doit permettre de détecter un carton sur une bande transporteuse.

Pour cela on utilise une caméra de la marque SICK, qui permet d'acquérir les images du carton sur la bande transporteuse et de l'envoyer sur le PC via un câble RJ45. Le PC va traiter ces images grâce à un programme intégré.

La bande transporteuse permet de déplacer le carton.

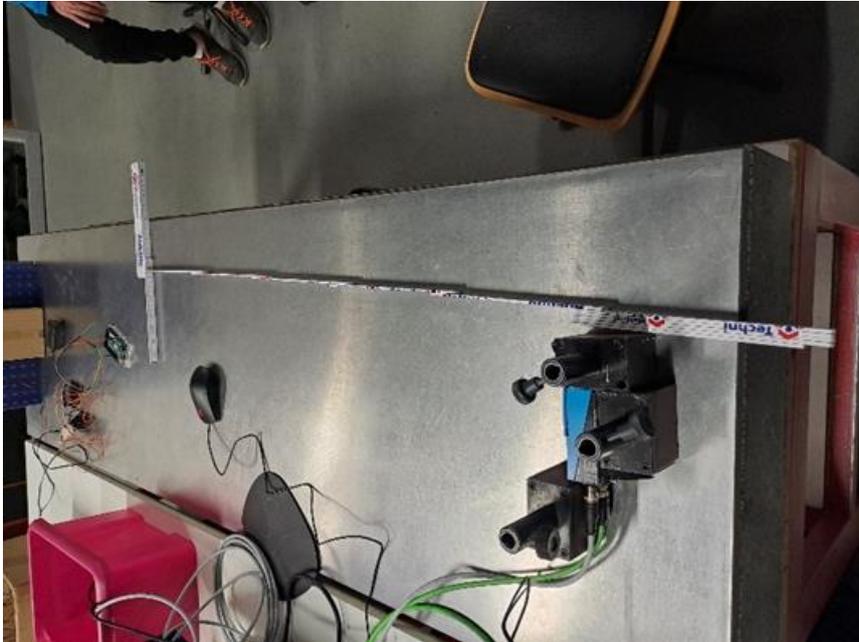
L'éclairage permet d'éclairer la scène en blanc, afin d'obtenir une détection optimale du carton, pour cela nous éclairons la scène avec des LEDs, qui sont alimentées en 12V et qui sont pilotées grâce à la carte Arduino.

L'ordinateur permet de traiter ces images de façon optimale.

3. B.Détection du carton et modélisation du support caméra (Partie de Noa)

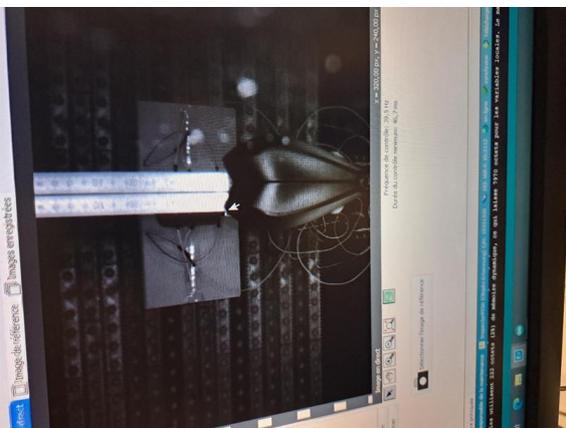
1. B.1 : Définir le champ observable de la caméra

Pour déterminer le champ observable de la caméra, j'ai procédé ainsi :

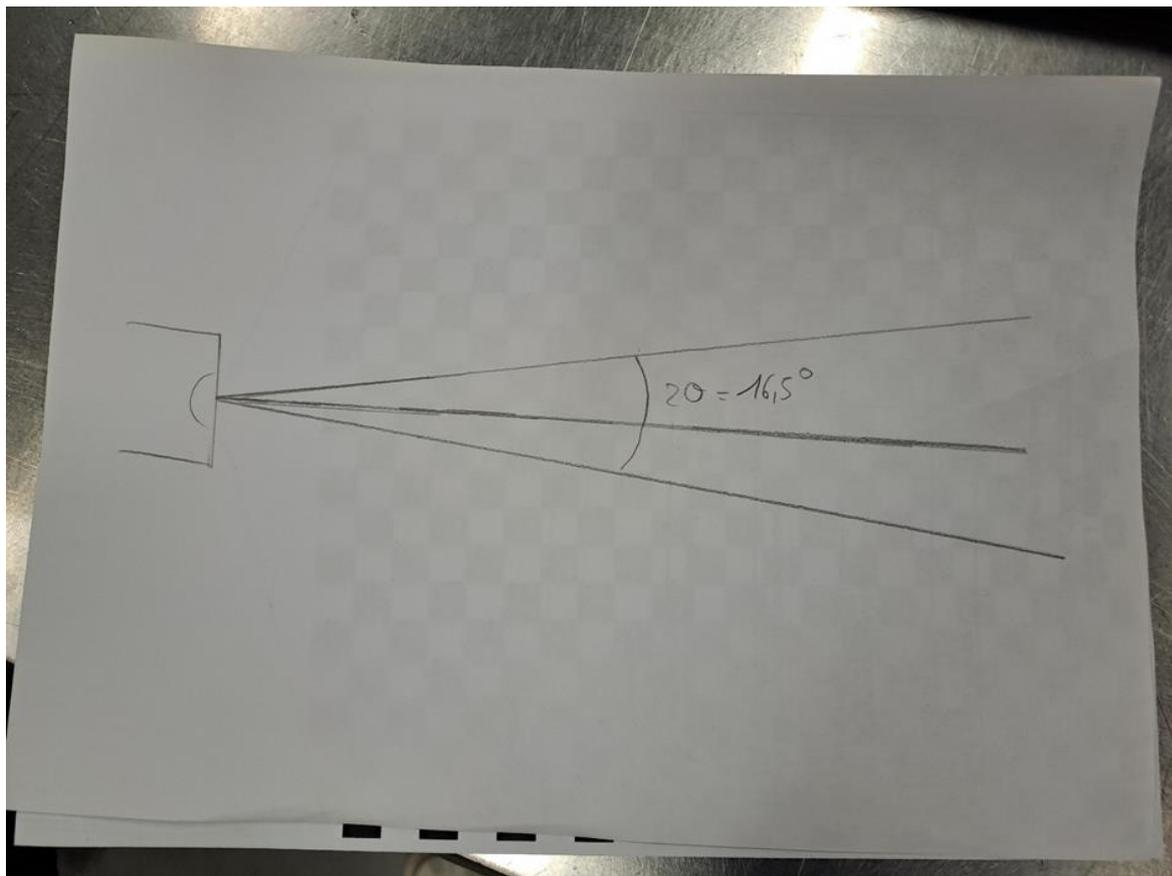


- Je place la caméra sur le marbre de manière stable et sécurisé
- je place un mètre le long de la caméra
- je place un second mètre à 1 mètre de la caméra, en perpendiculaire à la caméra.
- Puis en observant l'image, on peut voir qu'elle largeur la caméra peut observer à un mètre de distance.

Après avoir trouvé la largeur observable à 1 mètre, qui est de 29 cm.



On peut réaliser un schéma pour caractériser l'angle de champs :

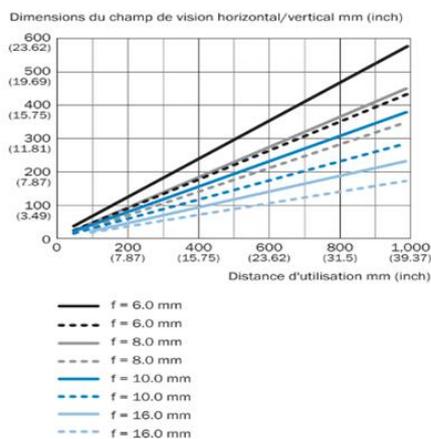


Avec les mesures réalisées, on peut calculer le demi-angle par un calcul de trigonométrie :

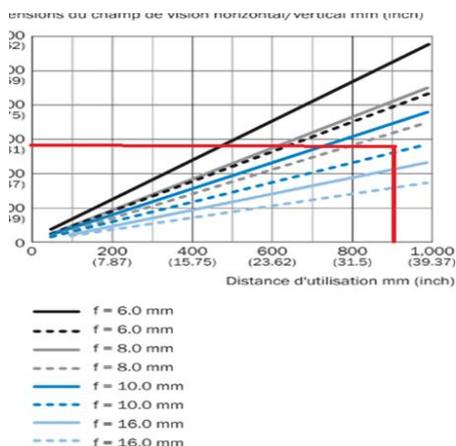
$$\text{TAN} = \text{opposée} / \text{adjacent} = (29/2) / 100 = 8.25^\circ$$

On a un demi-angle qui est de 8.25° , pour avoir l'angle total on multiplie cet angle par 2 et on obtient un angle de 16.5° .

Grâce à ces calculs on va pouvoir chercher dans la documentation technique de la caméra la distance focale de l'objectif de la caméra.

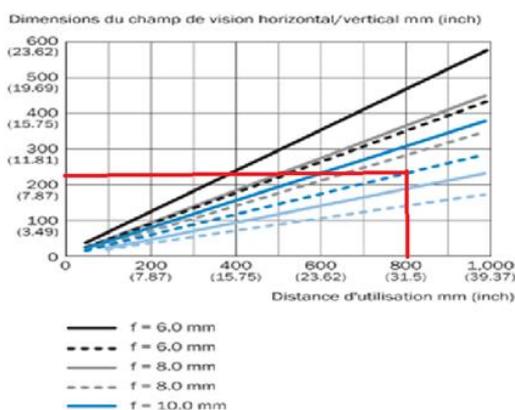


Pour confirmer les calculs, j'ai réalisé la même démarche mais cette fois-ci à une distance de 90 cm, on observe un objet de 25 cm.



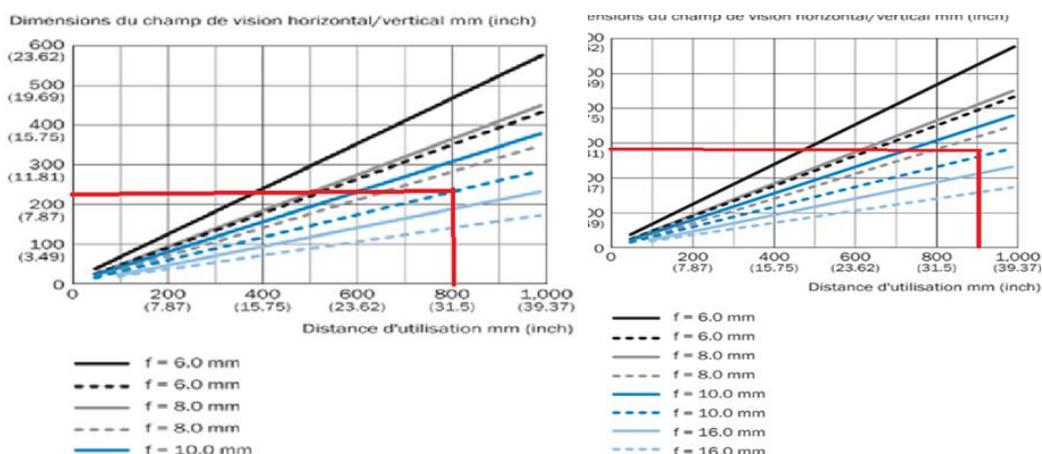
Essai à 900 mm de la caméra.

Puis, j'ai refait un essai à 80 cm de la caméra, la caméra observe objet de 22 cm



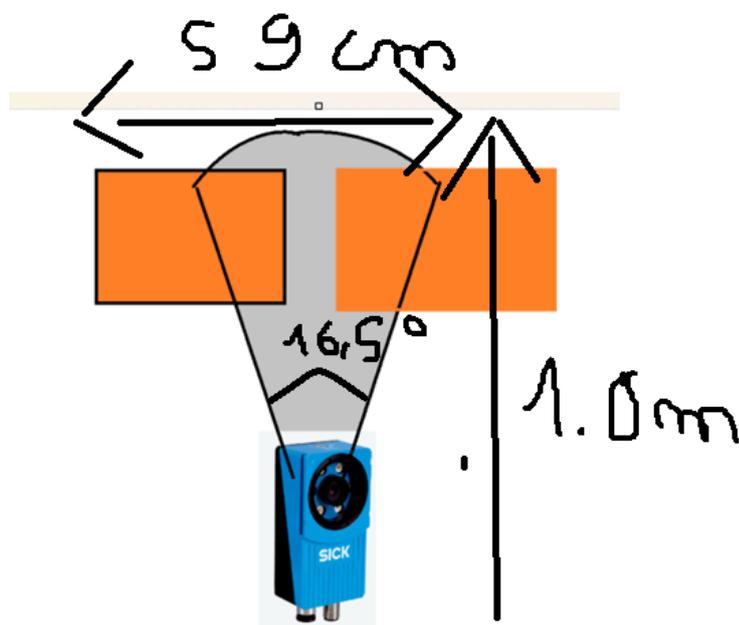
Essaie à 800 mm de la caméra.

En comparant, les deux résultats, on remarque que l'essai a 800 mm et l'essai à 900mm son sur la même courbe qui est la focale de 10.0mm.



On peut donc conclure que la focale utilisée est celle qui est mise sur la caméra par défaut qui est la focale de 10 mm.

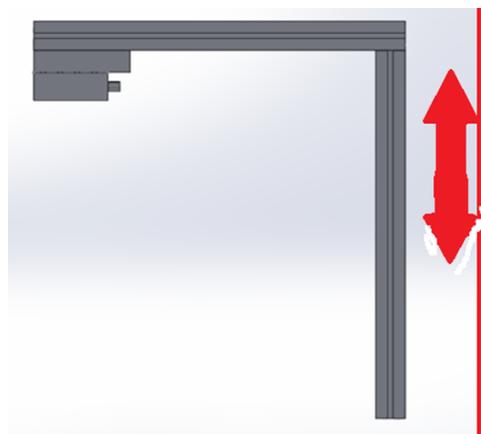
On peut donc réaliser un schéma pour imaginer la caméra avec la distance d'utilisation.



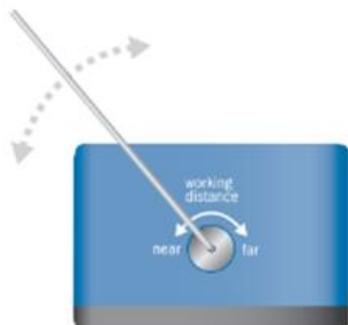
Grâce à ces démarches, je vais pouvoir régler de manière optimale la caméra, afin de détecter le carton avec le bon réglage afin d'avoir une image nette et précise.

Pour ajuster la profondeur de champs, il y a plusieurs étapes à suivre :

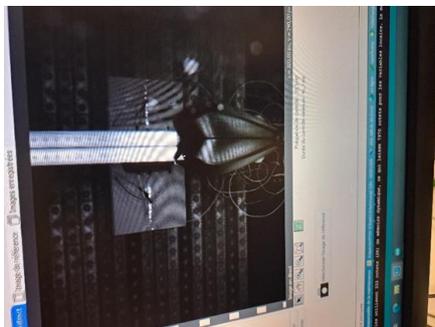
Etape 1 : il faut ajuster la hauteur du profilé sur lequel est fixé la caméra*



Etape 2 : A l'aide d'une clé Allen, on va ajuster la profondeur de champ afin que la caméra puisse voir le carton de manière net et que le tapis ne soit pas vu ou de manière floue.



Etape 3 : Observer sur l'écran et régler le gain ainsi que l'exposition de la caméra

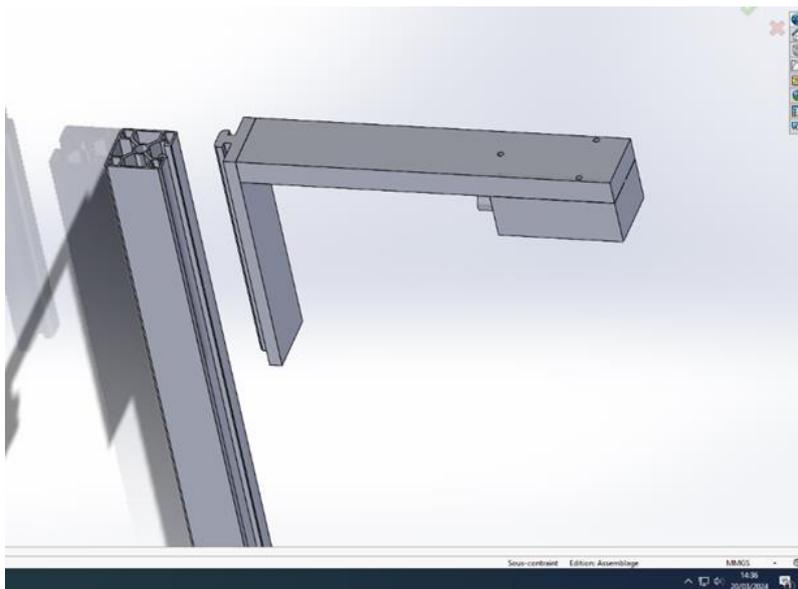


2. B.2 : création du support pour la caméra

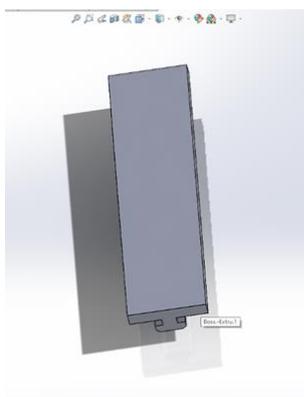
Pour pouvoir régler la caméra, j'ai dû créer un support qui puisse se glisser dans le profilée

J'ai réalisé deux prototypes :

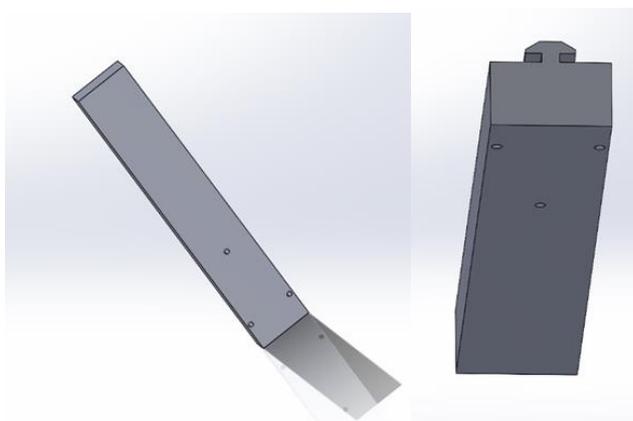
Le premier :



Il contient un support poteaux



Un support pour la caméra qui est fixe



Et le profilé qui est déjà présent à l'entreprise.

En réalisant l'assemblage sur SolidWorks, je me suis rendu compte que cela était trop compliqué, car il demande beaucoup de précision et les contraintes mécaniques étaient élevées. Il fallait rajouter des équerres pour soutenir le support car il est imprimé en 3D et que la caméra est trop lourde.

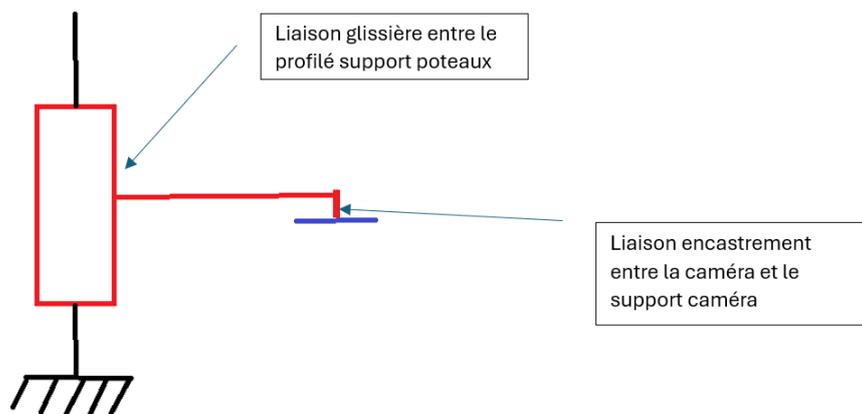


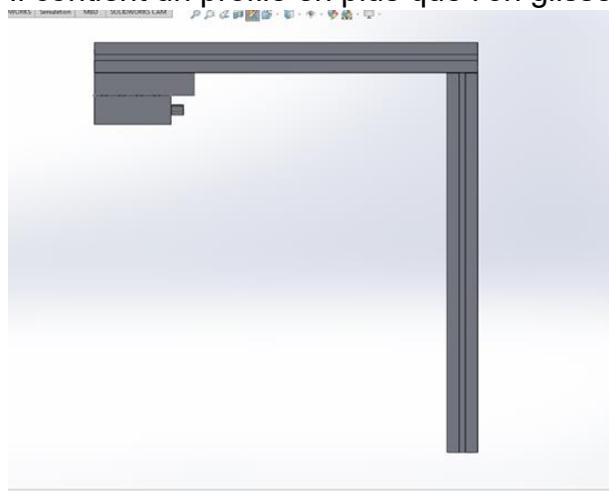
Schéma cinématique du premier prototype

Ce prototype avait une contrainte en plus, car la caméra ne pouvait pas être réglée en longueur il fallait donc que la pièce qui tient la caméra soit aux bonnes dimensions.

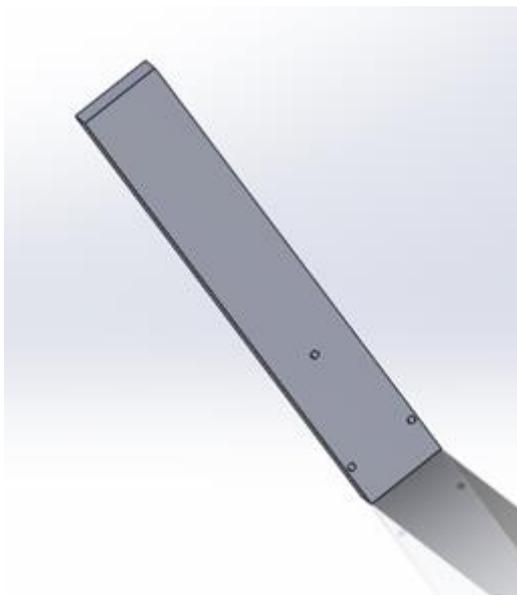
Il a fallu penser à une autre solution.

Le deuxième prototype se présente différemment :

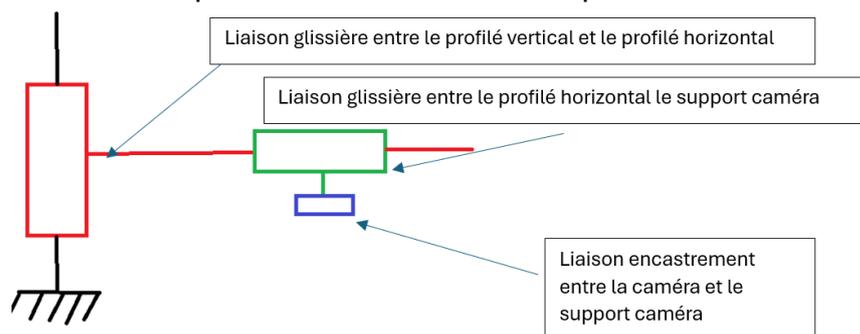
Il contient un profilé en plus que l'on glisse dans le premier



Un support pour la caméra



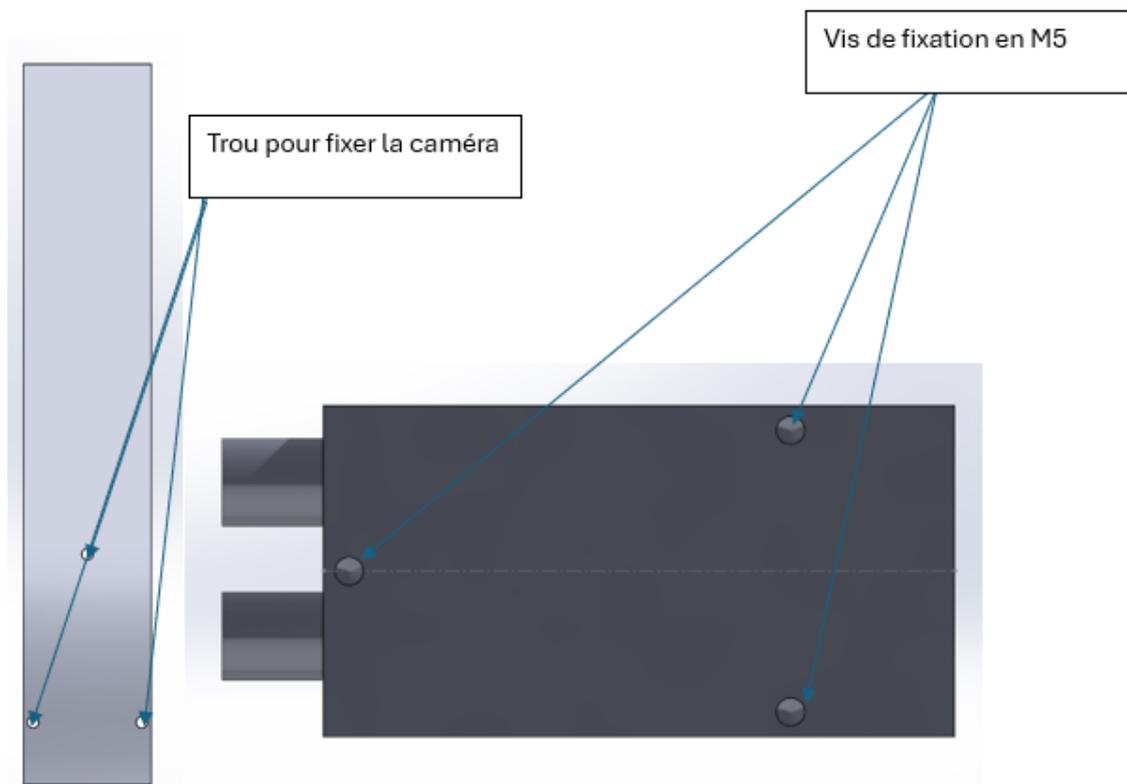
Ce deuxième prototype est dans sa conception beaucoup plus simple, car il a moins de contraintes et permet à l'ensemble d'être plus libre en termes de mouvement.



Schéma

cinématique du 2e prototype

Il y a maintenant le support caméra qui a subi des modifications, pour des raisons d'impression.

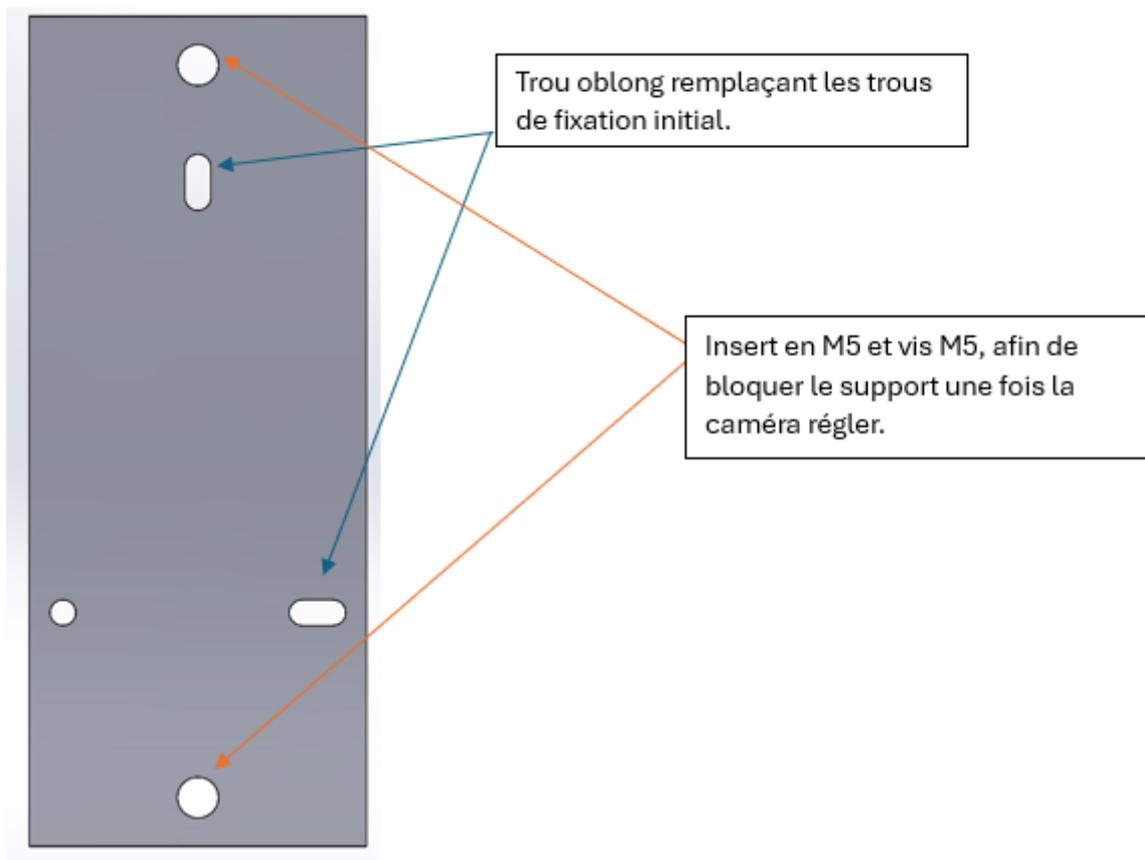


Comme on peut le voir sur la photo il y a trois trous qui servent à fixer la caméra dessus.

Une fois imprimer, en essayant de fixer la caméra, je me suis rendu compte que l'imprimante présentée un défaut : elle n'a pas la précision nécessaire pour que chaque trou corresponde aux vis de la caméra.

Il a donc fallu adapter la pièce afin que les vis puissent être fixer sur le support.

Pour remédier à cette contrainte, j'ai remplacé deux des trois perçages par des trous oblongs.



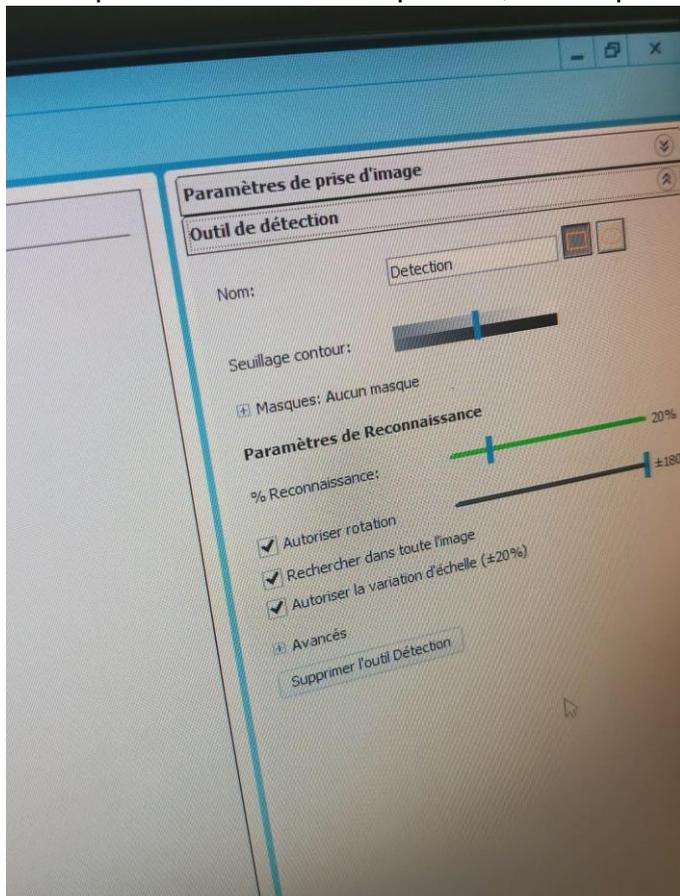
Le fait de mettre des trous oblongs permet de mettre les vis avec plus de facilité, car si la vis n'est pas coaxiale avec son perçage, on peut toujours faire coulisser la vis le long du trou afin de faire correspondre la vis avec son perçage.

3. B.3: Création du programme de détection du carton

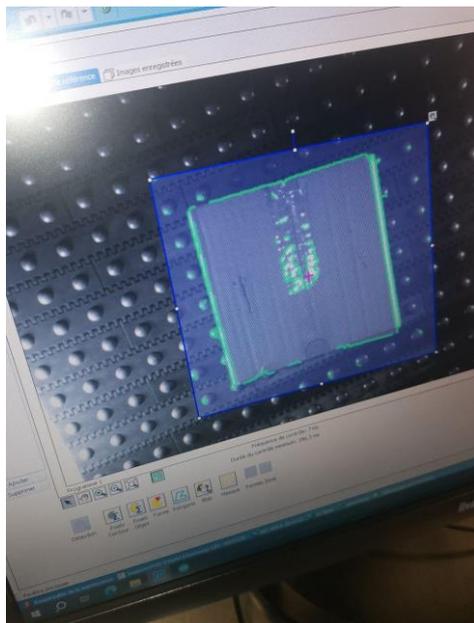
Pour pouvoir détecter le carton, il va falloir utiliser les outils que le logiciel de la caméra.

Dans la doc technique de la caméra, il y a l'outil détection qui est disponible et nous allons l'utiliser.

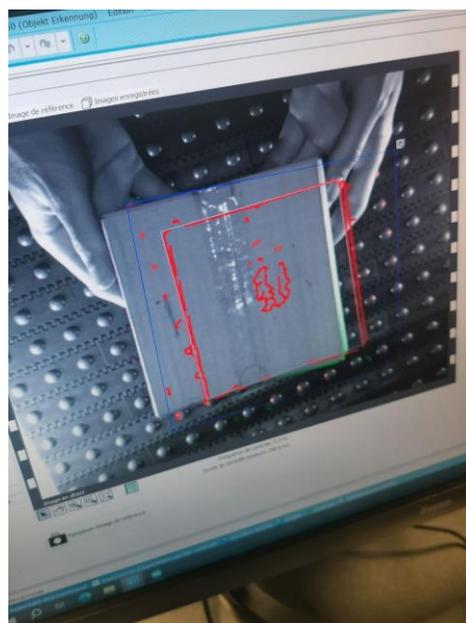
Pour que la détection soit optimale, il faut que cocher ces cases la



il faut mettre le seuil de détection à 25% maximum, sinon les imperfections du cartons peuvent ne pas le détecter.



Ici le carton est détecté



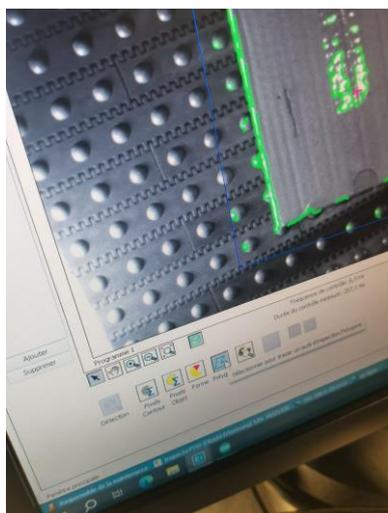
ici le carton n'est pas détecté

-La case "rechercher dans toute l'image" permet de détecter les cartons, n'importe où sur l'image tant que le carton est vu en entier.

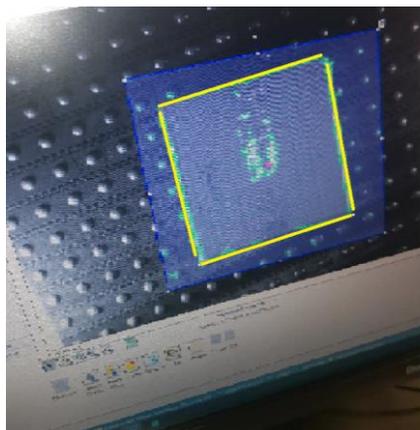
-La case "autoriser la variation d'échelle" permet la variation de taille des cartons et cela permet de détecter le carton même s'il a une taille différente du précédent.

-Enfin le curseur "autoriser rotation" permet de détecter le carton peu importe son angle d'orientation.

On va utiliser aussi l'outil polygone.



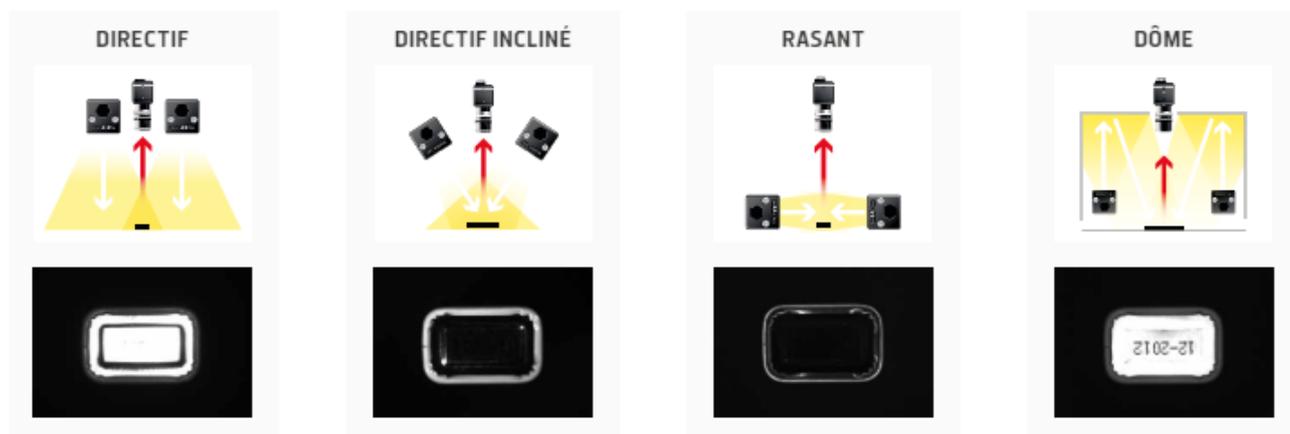
Pour l'utiliser on va placer un carton le plus droit possible sur le tapis, ensuite on va tirer un trait sur un des côtés du carton, on répète cette opération sur toutes les faces du carton.



Une fois ces outils mis en place, le carton est détecté de manière optimale et le programme python (partie de Yann) va permettre de donner les coordonnées du carton ainsi que son orientation

4. C) Partie éclairage (Partie de Lukas)

1. Choix de la position de l'éclairage :



Nous avons ici quelques moyens d'éclairage pour notre carton sur la bande transporteuse

Il y a l'éclairage directif qui est un éclairage qui est de face et donc qui ne pourrait pas convaincre avec notre système ou la caméra sera positionné au-dessus de la bande transporteuse.

L'éclairage directif incliné est un éclairage qui permet de bien éclairer le carton mais il serait plus difficile à mettre en place en termes de moyen, et il pourrait éclairer le carton plus qu'on le voudrait et pourrait dé calibrer la caméra.

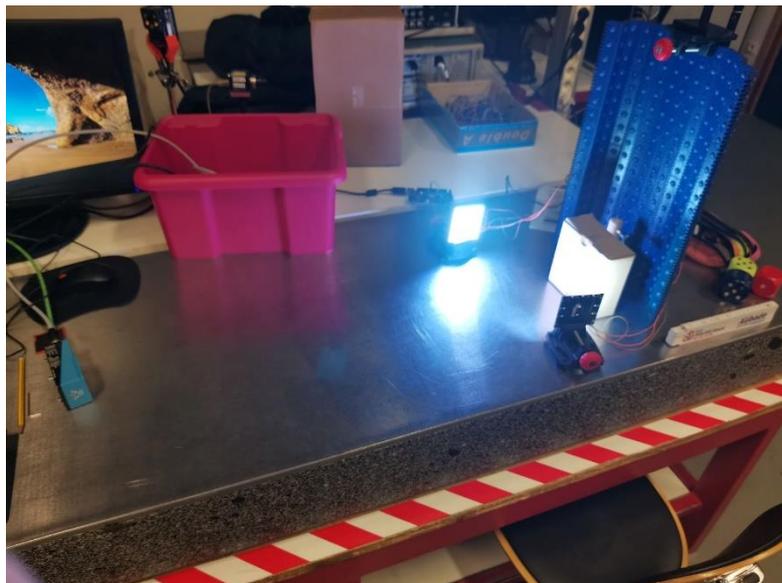
L'éclairage dôme ne nous n'ai pas utile car il éclaire la surface du dessus du carton et ne nous permet pas d'avoir les bordures du carton, mais encore l'éclairage pourrait refléter et induire en erreur notre caméra.

L'éclairage rasant est le choix que nous avons fait pour notre projet, les éclairages seront positionnés au bord des tapis, et pour avoir un bon éclairage adéquat il y en aura 1 de chaque côté des bandes transporteuses donc 2 éclairages.

Lycée Jean Mermoz -68300 SAINT-LOUIS

Identification et mesure de position d'un produit sur bandes roulantes

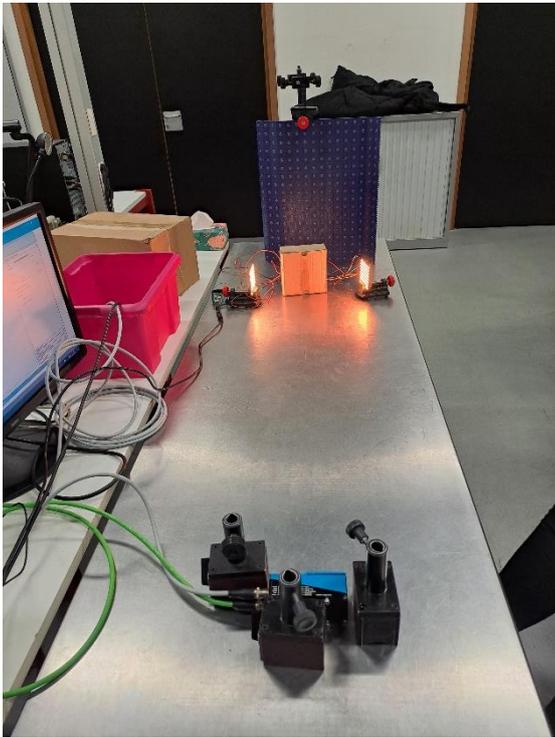
METZLER Noa - DAGHOUR Yann - SAUTTER Lukas



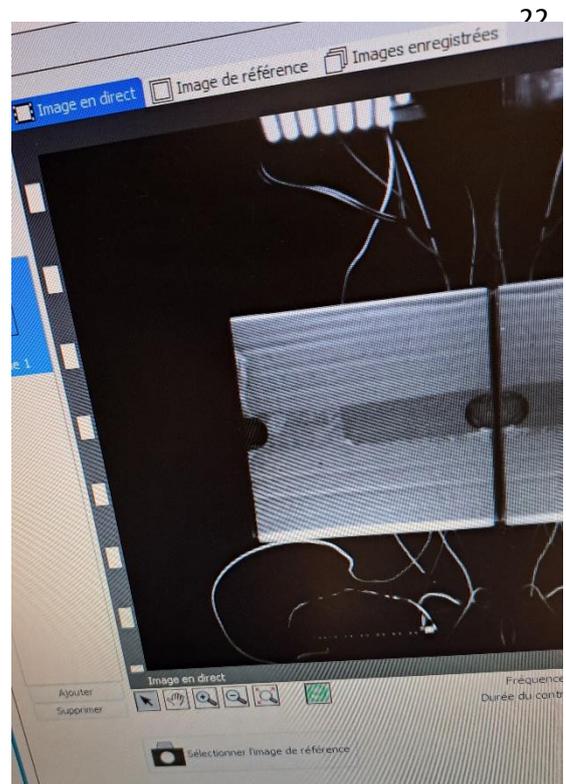
Premier test

Le rouge et le jaune orange sont des couleurs qui se rapproche pas mal de la couleur du carton donc ce sont des couleurs que nous observons bien à la caméra, mais le blanc est la meilleure car elle regroupe toutes les couleurs donc contient celles qui ressemblent le plus au couleur du carton.

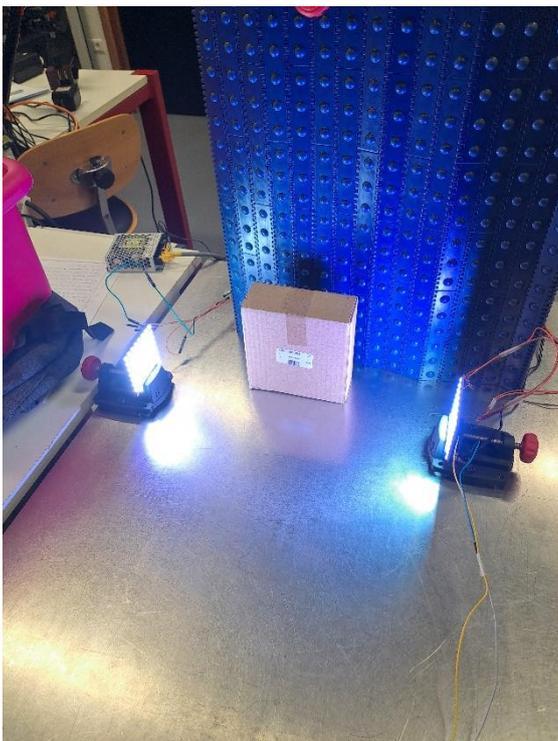




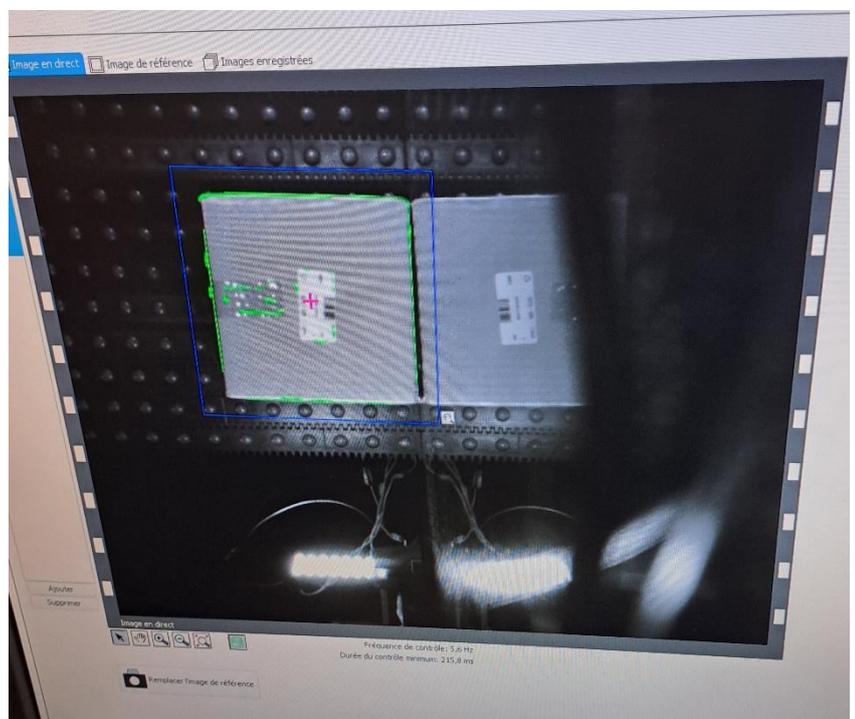
Vu de l'ensemble avec couleur des LEDs jaune-orangé



Vue de la caméra avec éclairage jaune-orangé

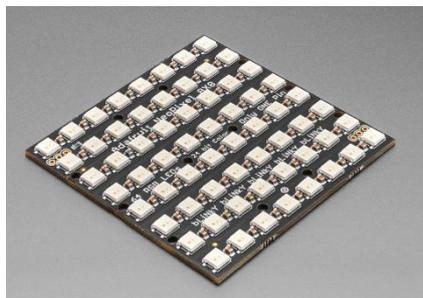


LED blanche



Vue de la caméra avec éclairage blanc

Pour notre maquette et nos tests, nous avons utilisé des LEDs Adafruit Neopixel 8x8 (photo ci-contre)



Bande LED
Adafruit
Neopixel

Ces LED consomment 60mA par LED de couleur blanc à la puissance maximum, donc on a $128 \times 60 \times 10^{-3} = 7.68 \text{ A}$, à la puissance max.

Mais notre transformateur nous pouvant accepter plus de 2 A nous a obliger à prendre un transformateur de 10A car il y avait trop d'intensité. Celui de 2 A n'aurais pas tenu une telle intensité.

Nous avons aussi pris en compte qu'il y aura déjà de l'éclairage qui sont : la lumière extérieure (du soleil) et la lumière des lampes au plafond comme chez HABASIT. Nos premiers tests ont été effectuer dans nos labos, donc sans prise en compte de la lumière extérieure, et en ajustant comme nous voulions les lampes au plafond. Puis on a aussi fait des tests avec notre maquette avec de l'éclairage extérieure et avec ou sans la lumière artificielle des lampes au plafond.



Grâce à notre maquette, et à l'aide d'un Luxmètre nous avons déterminé la puissance minimale qu'il nous faudrait pour éclairer. Nous aurons besoin au minimum de 3200 Lux.

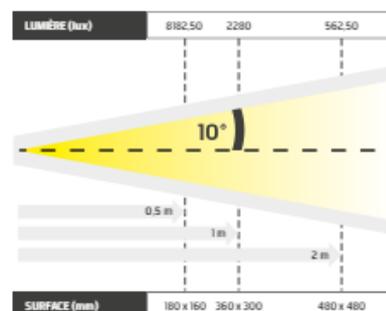
Les bandes LEDs seront au bord des bandes transporteuses qui font 160cm de largeur, le carton positionné au milieu fera en taille maximum 40cm, donc $160 - 40 = 120\text{cm} / 2$ car nous avons 2 bandes, $120 / 2 = 60$, elles seront à 60cm du carton.

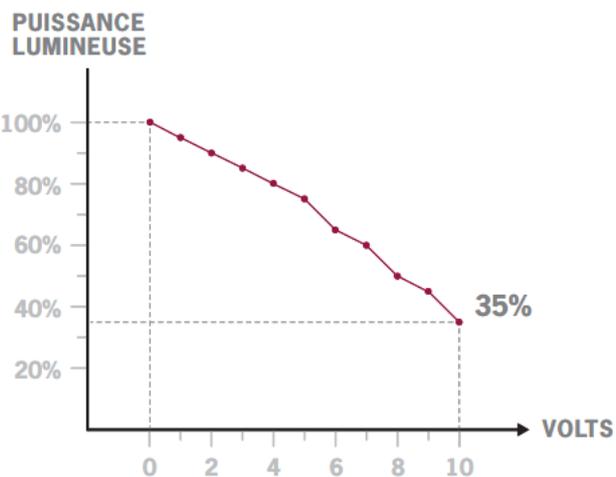


Bande
transporteuse de
HABASIT

2. Choix de l'éclairage

J'ai regardé sur plusieurs sites internet donné par l'intermédiaire de SICK France, et c'est sur le site TPL Vision que j'ai trouvé l'éclairage qu'il nous faudrait. Ça sera donc 2 bandes LEDs EBAR+500





Voltage (VDC)	0	1	2	3	4	5	6	7	8	9	10
Luminosité	100%	95%	90%	85%	80%	75%	65%	60%	50%	45%	35%

EBAR+	Distance : 0,5 m		Distance : 1 m		Distance : 2 m	
	Valeur Max	Zone éclairée	Valeur Max	Zone éclairée	Valeur Max	Zone éclairée
EBAR+ 500	14302,50	400x170 mm	6120,00	400x260 mm	1890,00	560x480 mm

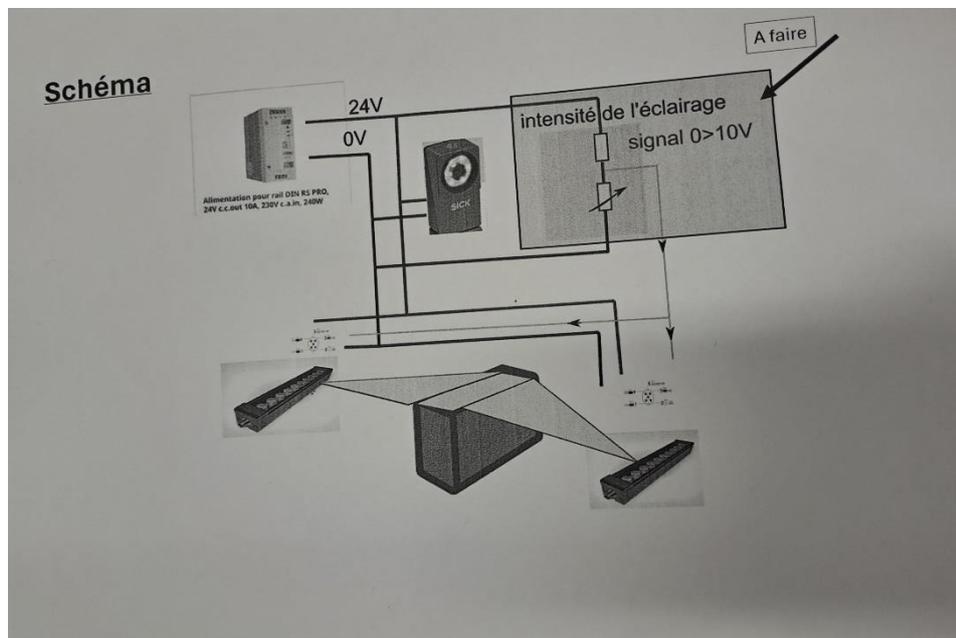
Notre choix qui sera la EBAR +500, qui possède un éclairage maximum de 14302.5 Lux pour une distance de 50cm. Donc une puissance qui sera assez suffisante pour éclairer le carton.

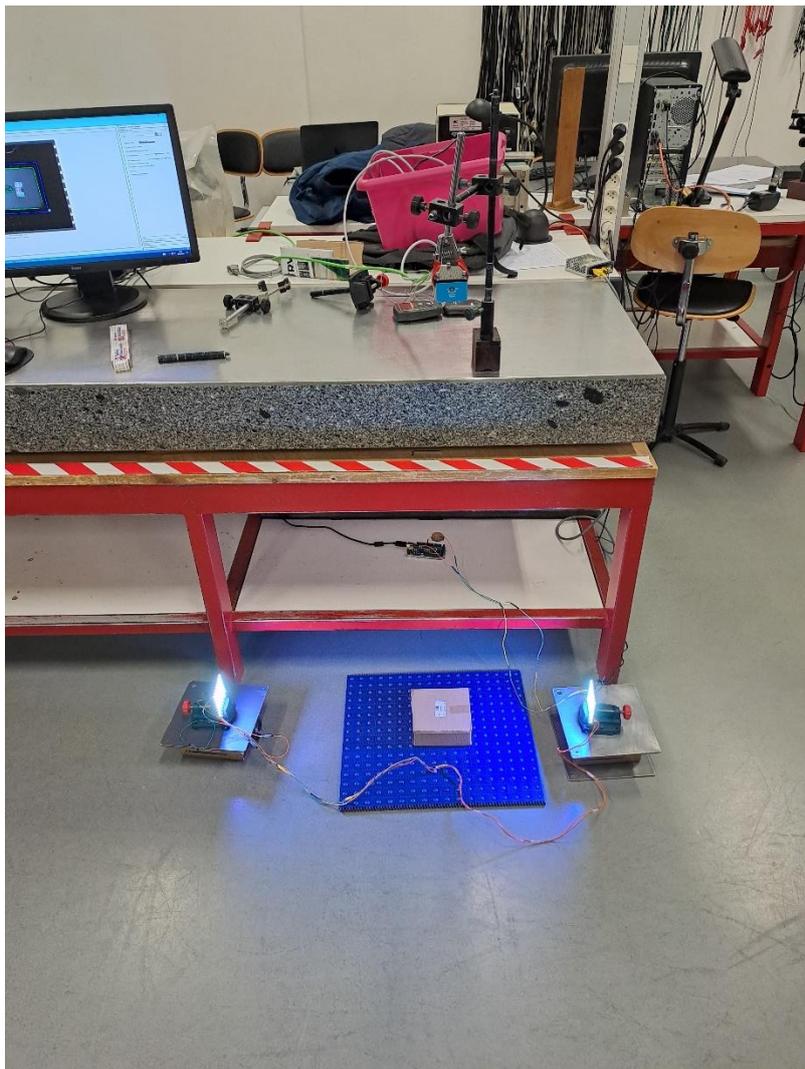
Lorsque la puissance lumineuse est maximale, les potentiomètres sont ouverts donc tout le courant passe. Donc le signal 0V correspond au 100% de la puissance lumineuse.

Quand les potentiomètres sont réglés au maximum la puissance lumineuse sera réglée à 35% de la puissance maximale.

On a créé un montage électrique afin de pouvoir faire varier la puissance lumineuse des bandes LEDs EBAR +500.

A l'aide de 2 potentiomètres (voir encadré rouge du schéma) on transforme la tension d'alimentation 0-24V en un signal modulable sur 0-10V.





Voici le montage avec la caméra suspendu au-dessus du sol comme elle sera positionné chez HABASIT. Il y a les LEDs Adafruit Neopixel qui sont posés au sol pour faire l'éclairage rasant que nous voulons pour éclairer le carton.

3. CONDITIONS D'UTILISATIONS :

- -10° à +40°C / 80% d'humidité sans condensation.
- Pas de choc thermique (variation de température max de 10°C en 24h).
- Non adapté à une utilisation en extérieur.

4. Protocole éclairage :

Respecter les tensions d'alimentation et les bornes de branchement.

Ne pas modifier ou démonter tout ou une partie du produit.

Ne pas connecter ou nettoyer sous tension.

Ne pas regarder la source lumineuse directement et suivre les préconisations suivantes :

•

Interposer, dans la mesure où le poste de travail le permet, un filtre permettant de bloquer les rayonnements émis par l'éclairage sous cadre fixe ou réglable entre la source et l'opérateur.

- Lorsque la mise en œuvre des dispositions précédentes n'est pas possible, fournir aux opérateurs des lunettes de catégorie 4 (disponibles à la vente par TPL Vision).
- Interdire ou limiter tant que possible l'accès direct à la source (exposition dans l'axe du rayonnement).
- Établir un périmètre de sécurité afin d'éviter aux opérateurs de s'approcher de la source au-delà des distances nominales de risque oculaire préconisées par le constructeur
- Dans tous les cas, faire en sorte que les moyens utilisés atténuent convenablement les grandeurs d'exposition (caractéristiques des écrans ou lunettes à choisir en fonction des longueurs d'ondes auxquelles les opérateurs sont exposés).

5. NETTOYAGE (PRODUIT HORS TENSION)

Utiliser un chiffon doux et sec.

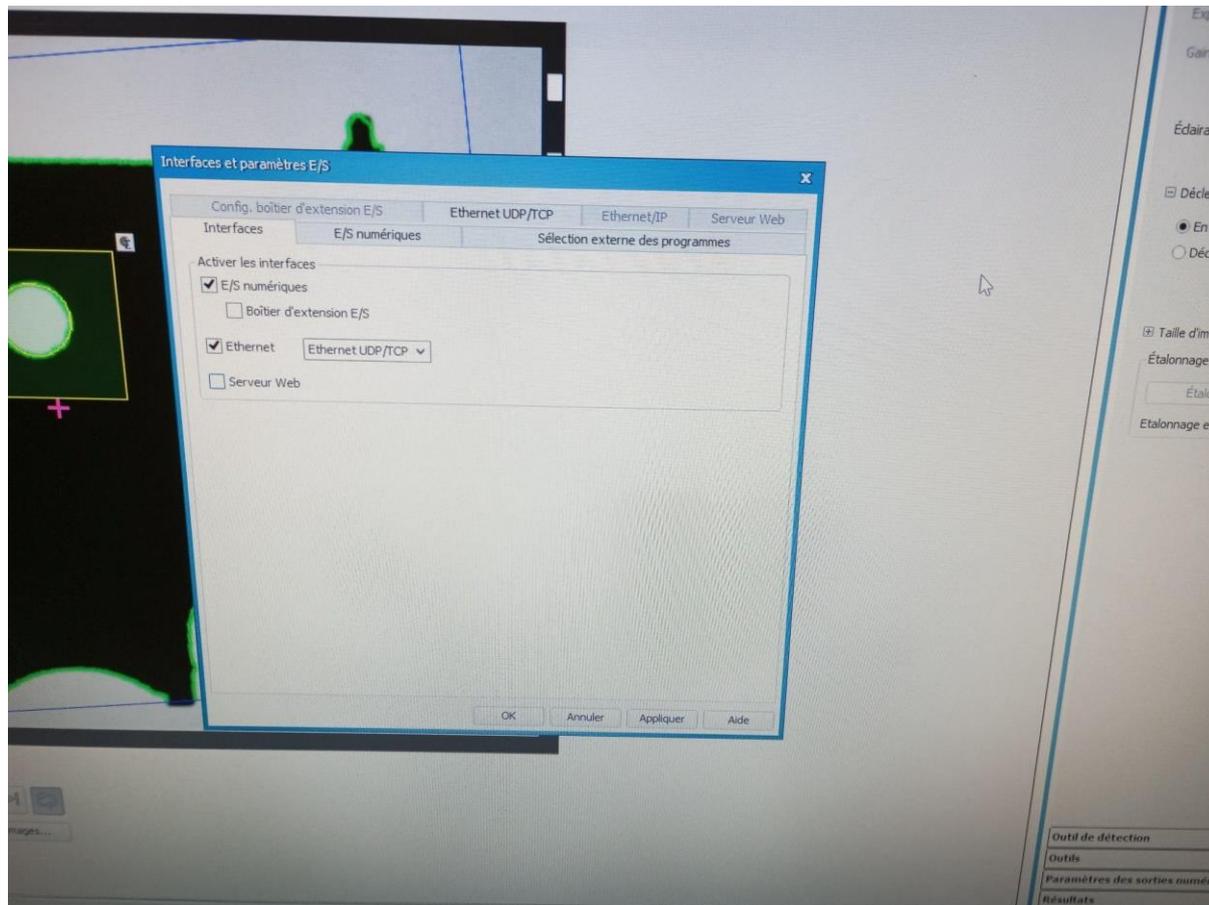
Ne pas utiliser de matériau abrasif.

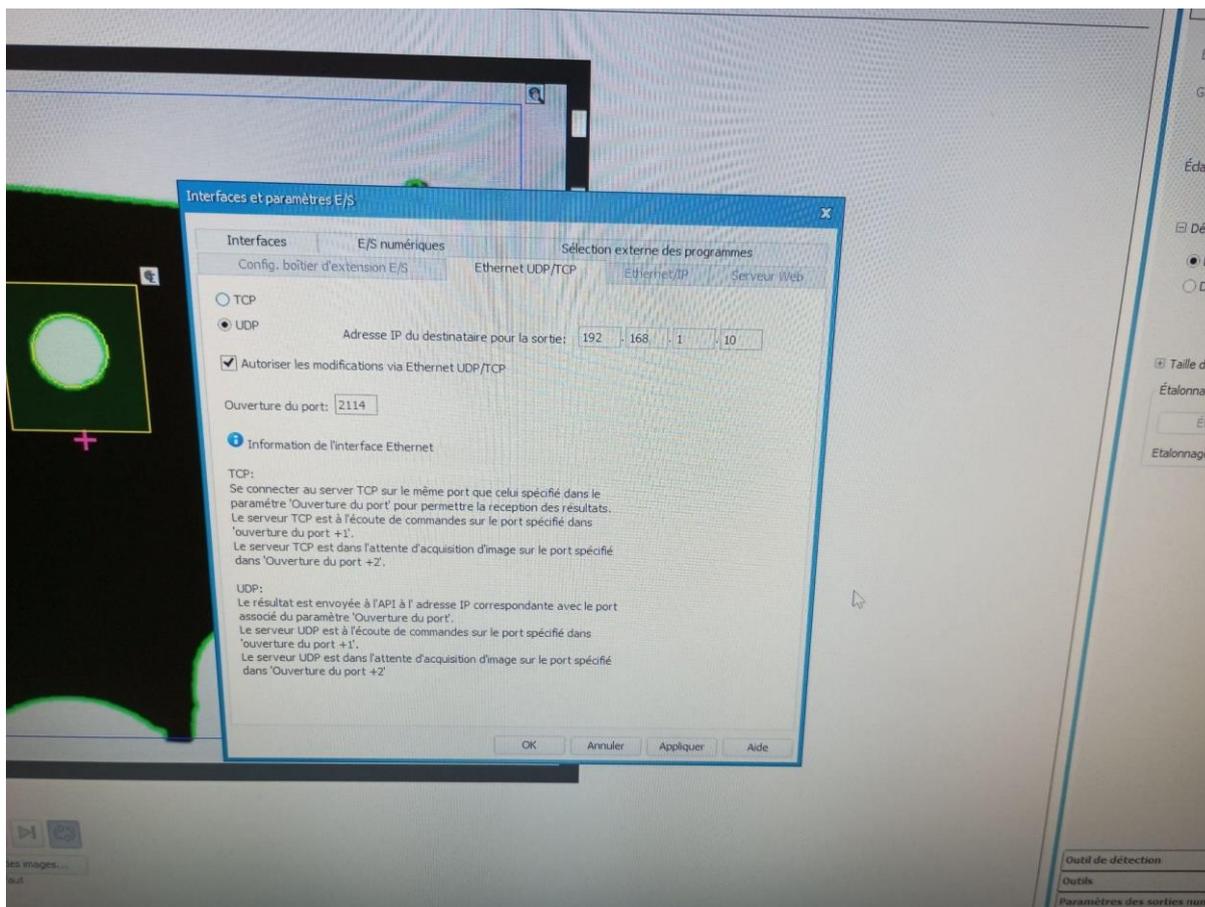
Ne pas utiliser de solvant ou de produit chimique agressif.

TPL Vision préconise l'utilisation d'alcool isopropylique.

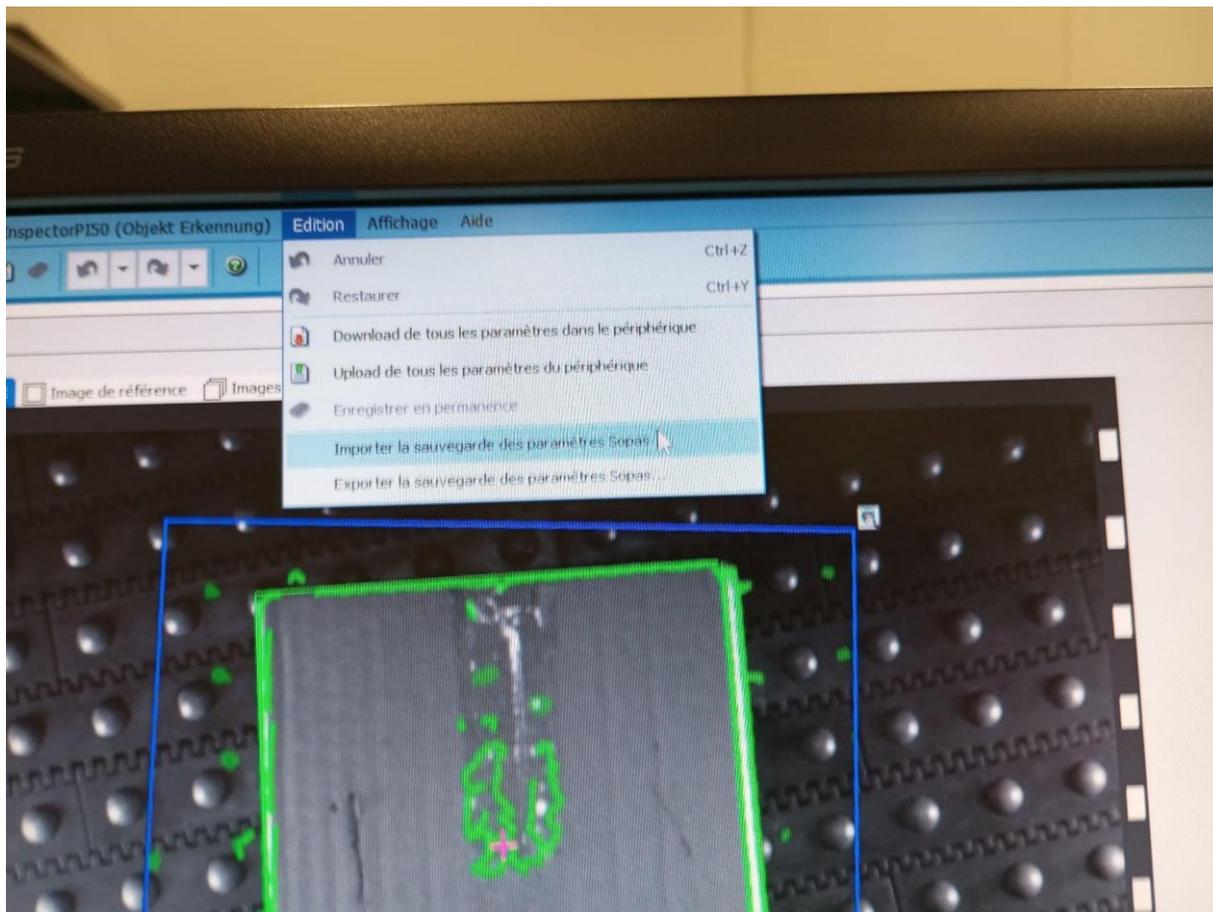
5. D. Partie programmation (Partie de Yann)

La première partie concerne la procédure pour communiquer avec la caméra et le transfert de données

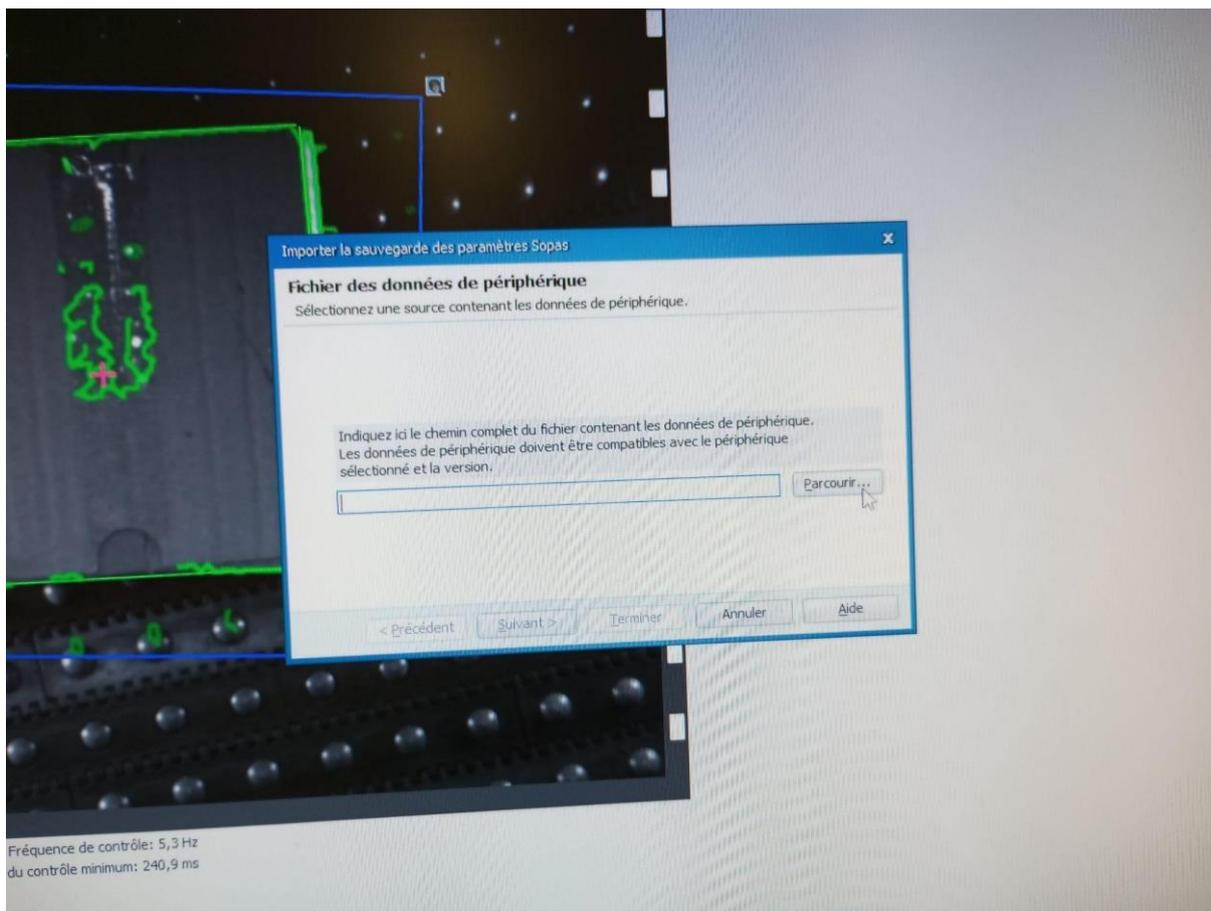




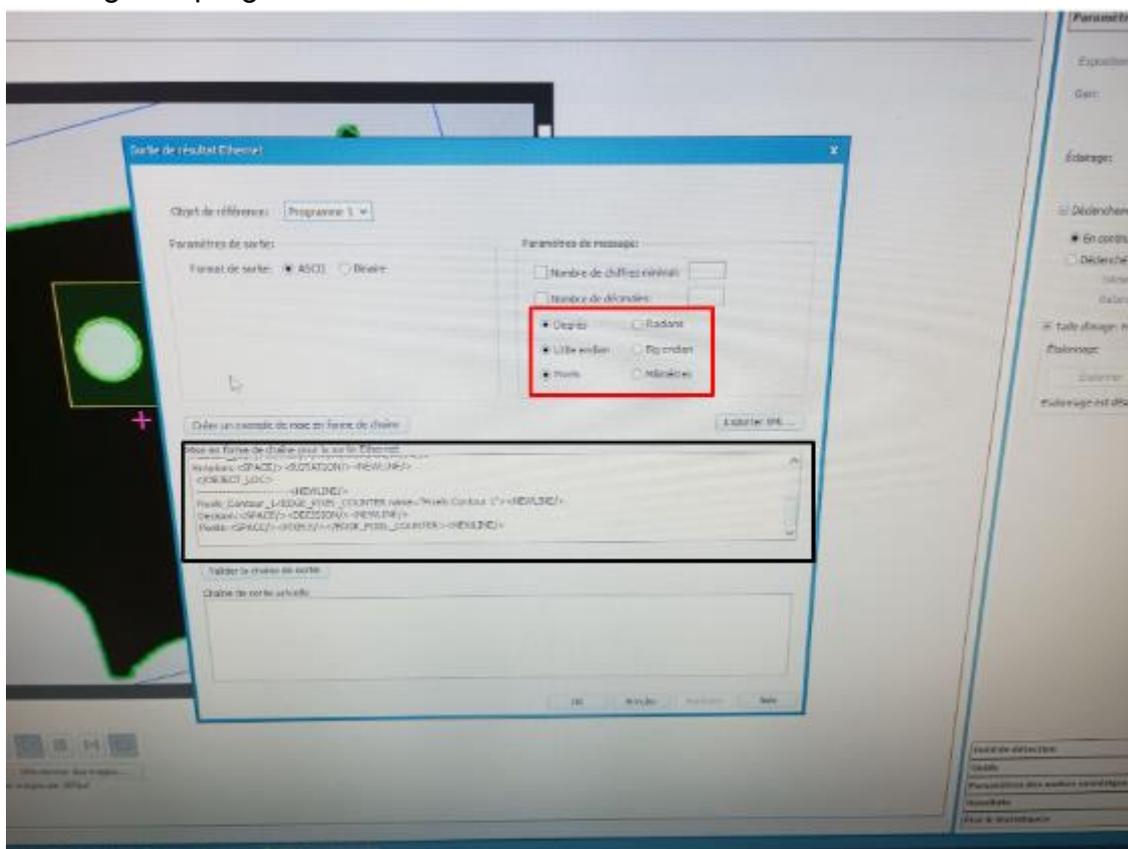
Dans l'onglet Ethernet UDP/TCP il faudra préciser l'adresse du destinataire et le port à ouvrir



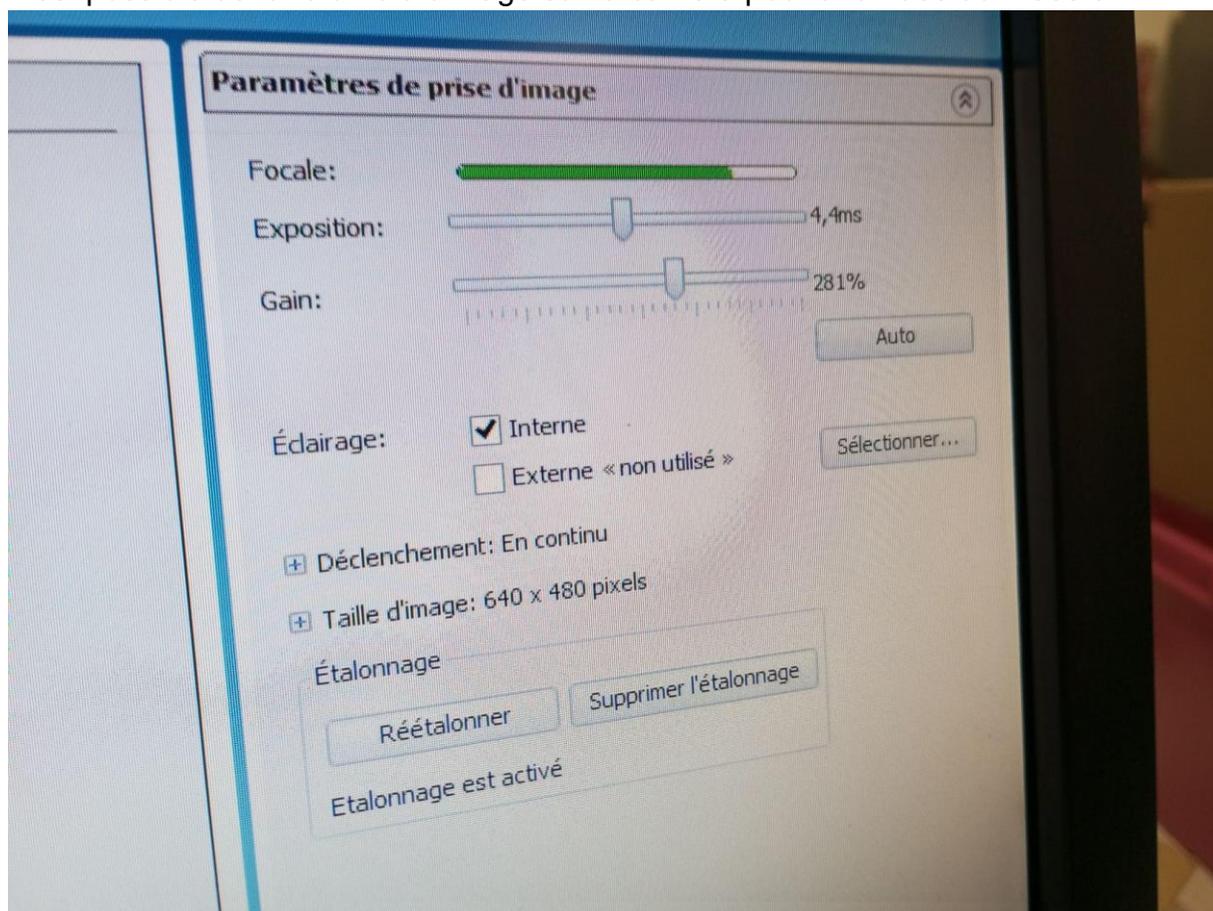
Pour importer le programme, il faudra aller dans l'onglet ci-dessus



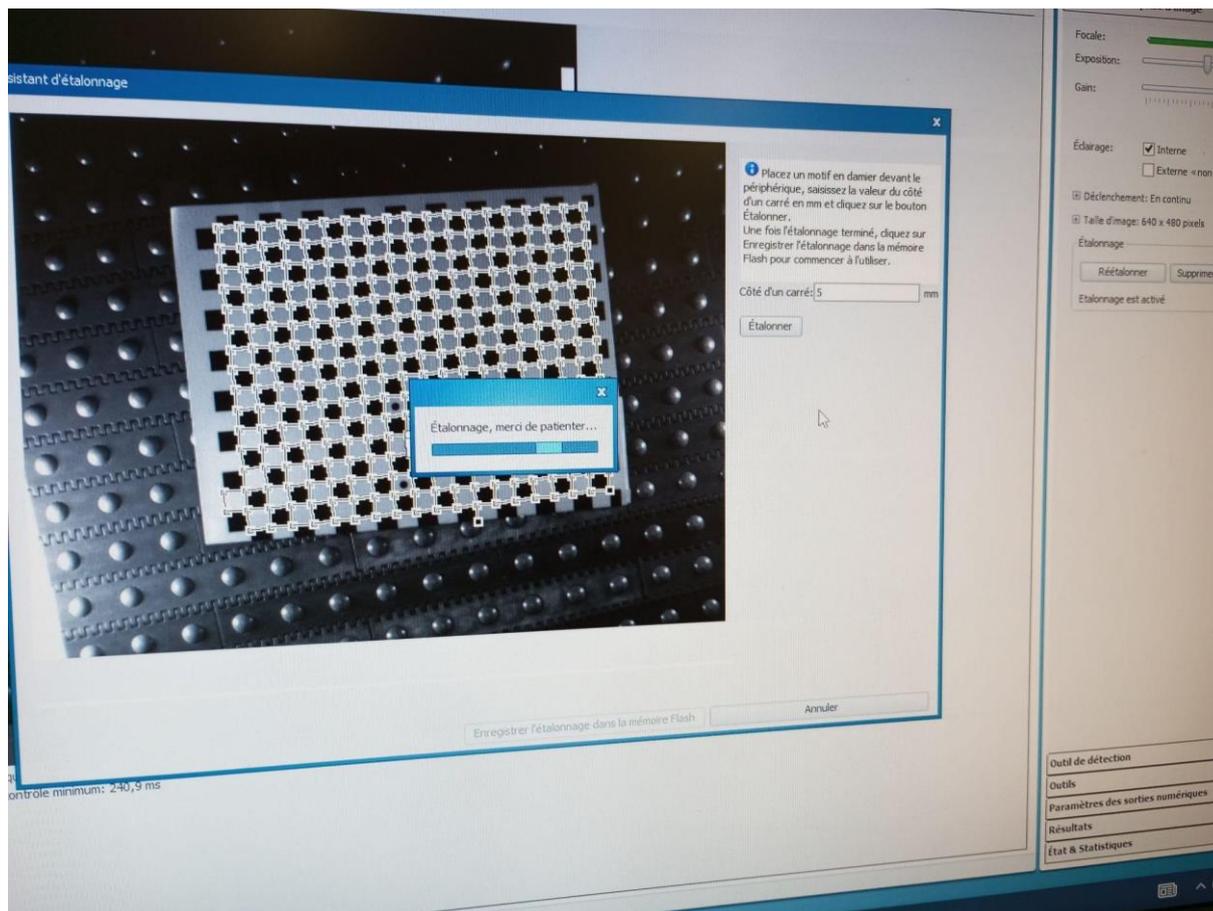
Il suffira ensuite de sélectionner la sauvegarde voulue, dans notre cas ce sera “sauvegarde programme”



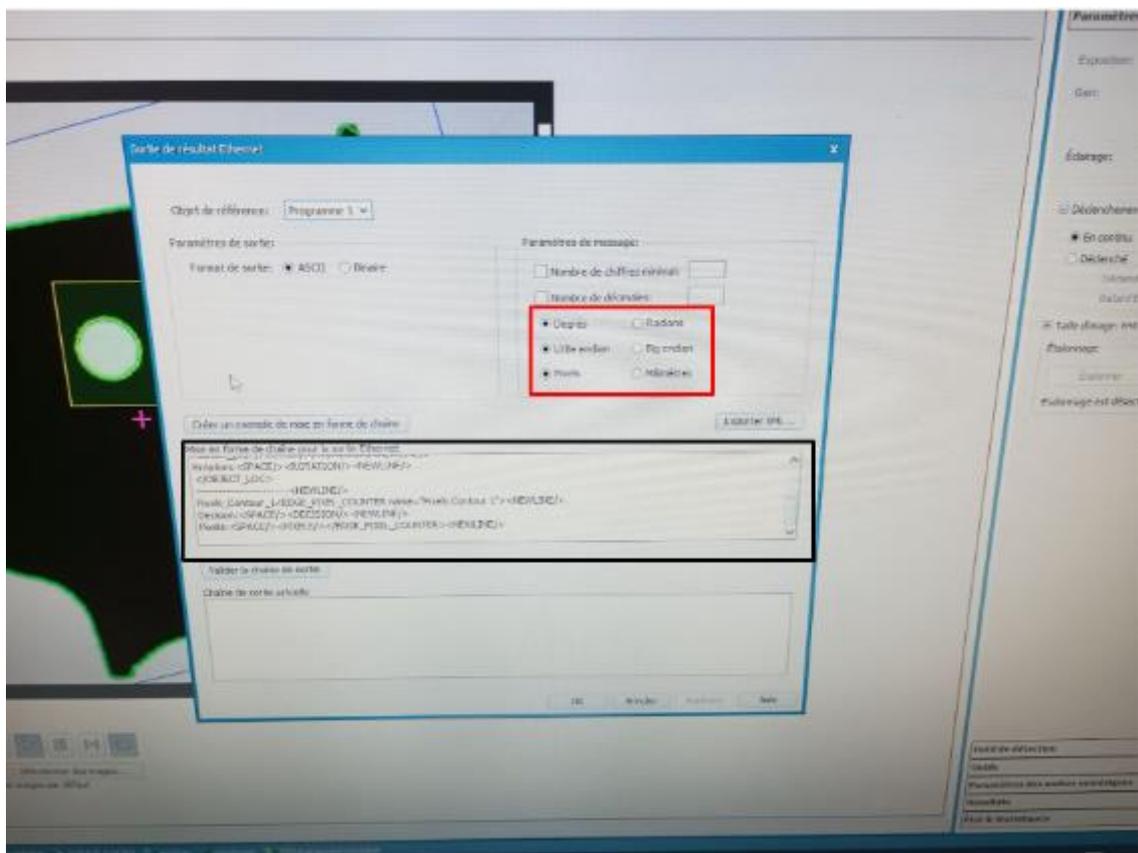
Les cases du cadre rouge permettent de modifier l'unité des données envoyées.
Il est possible de faire un étalonnage sur la caméra pour avoir des données en mm.



Il faudra cliquer sur étalonner



Nous avons placé un quadrillage sur une feuille de papier afin d'avoir une référence, dans notre cas les cases mesurent 5 mm, cette valeur sera à préciser lors d'un nouvel étalonnage



Dans le cadre noir on peut modifier le code afin de sélectionner les données voulues et de les mettre en forme pour les extraire avec un programme python

Le programme python devra servir à extraire les coordonnées du carton et son orientation

le début du programme traite la communication avec la caméra

```
# coding: utf-8

import socket
import string
import time

host = '192.168.0.10'
port = 2114

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((host, port))
    #s.sendall("Hello world".encode('utf-8'))
    l=0
    numeronem=0
    while 1:
        buff = s.recv(1024)
        #print(buff.decode())
        c= buff.splitlines()
        #print(c)
        #extraire le numero d'image
        class Camera():
            def __init__(self, a,b,elt,texte):
                self.a=a
                self.b=b
                self.elt=elt # tableau contenant les élément chaîne à analyser
                self.texte=texte
                self.position_1=''
            def distance(self):
                for i in range (len(self.elt)):
                    if "Position" in self.elt[i]:
                        if self.position_1=='':
                            self.position_1=(self.elt[i].decode()) # on transforme la chaîne byte en str plus facile à traiter
                #on a trouvé un emplacement avec le mot position reste à extraire x et y
                # les positions sont maintenant ds str plus facile pour extraire les nombre
```

Vert = cette partie vérifie si la caméra et le pc communiquent correctement

blanc = sélection de l'adresse ip et du port

rose = réception des paquets de données et mise sous forme de ligne

bleu = mise des données sous forme de tableau et conversion en chaîne byte pour simplifier l'extraction

la suite parlera de l'extraction de la chaîne

```

c=str(c)
ind_1=c.find('1(')#recherche de l'indice de :
print(ind_1)
ind_2=c.find(')1')#recherche de l'indice de :
print(ind_2)
ind_3=c.find(';1;')#recherche de l'indice de :
print(ind_3)
ind_4=c.find('2(')#recherche de l'indice de :
print(ind_4)
ind_5=c.find(')2')#recherche de l'indice de :
print(ind_5)
ind_6=c.find(';2;')#recherche de l'indice de :
print(ind_6)
ind_7=c.find('3(')#recherche e l'indice de :
print(ind_7)
ind_8=c.find(')3')#recherche de l'indice de :
print(ind_8)
ind_9=c.find(';3;')#recherche de l'indice de :
print(ind_9)

```

vert = caractère spéciaux insérés dans la chaîne

rouge = fonction utilisée pour la recherche des caractères

```

print('XY')

sous_1=float(c[:(ind_1)+2] :((ind_3)))] #j'ai X1 sous forme de flottant
print(sous_1)
sous_2=float(c[:(ind_3)+3] :((ind_2)))] #j'ai Y1
print(sous_2)
sous_3=float(c[:(ind_4)+2] :((ind_6)))] #j'ai X2
print(sous_3)
sous_4=float(c[:(ind_6)+3] :((ind_5)))] #j'ai Y2
print(sous_4)

```

bleu = conversion en float de la chaîne extraite

orange = sélection des bornes, la valeur à rechercher sera située entre ces 2 caractères

La chaîne d'entrée est codée en ASCII

Lycée Jean Mermoz -68300 SAINT-LOUIS

Identification et mesure de position d'un produit sur bandes roulantes

METZLER Noa - DAGHOUR Yann - SAUTTER Lukas

Chaîne:

```

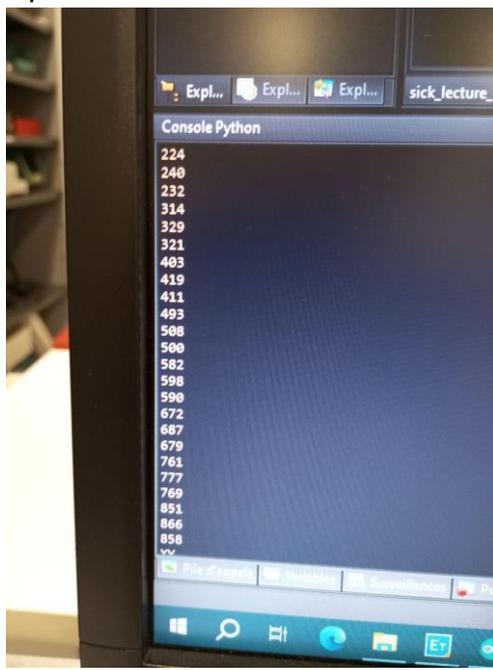
<MESSAGE unit="degrees" endianness="little"
coordUnit="mm">Numero_d'image:<SPACE/><IMAGE_NUMBER/><NEWLINE/>
Detecteur_d'objet.<NEWLINE/>
<OBJECT_LOC>
Localise:<SPACE/><DECISION/><NEWLINE/>
Taux:<SPACE/><SCORE/><NEWLINE/>
Echelle:<SPACE/><SCALE/><NEWLINE/>
Position_(X,Y):<SPACE/><X/>,<Y/><NEWLINE/>
Rotation:<SPACE/><ROTATION/><NEWLINE/>
</OBJECT_LOC>
-----<NEWLINE/>
Polygone_1<POLYGON name="Polygone 1"><NEWLINE/>
Corners:<SPACE/><NUM_CORNERS/><NEWLINE/>
<CORNERS corners="0">(X,Y):<SPACE/>1<X/>;1<Y/>)1<NEWLINE/>
</CORNERS>
</POLYGON>
-----<NEWLINE/>
Polygone_1<POLYGON name="Polygone 1"><NEWLINE/>
Corners:<SPACE/><NUM_CORNERS/><NEWLINE/>
<CORNERS corners="1">(X,Y):<SPACE/>2(<X/>;2;<Y/>)2<NEWLINE/>
</CORNERS>
</POLYGON>
-----<NEWLINE/>
Polygone_2<POLYGON name="Polygone 1"><NEWLINE/>
Corners:<SPACE/><NUM_CORNERS/><NEWLINE/>
<CORNERS corners="0">(X,Y):<SPACE/>3(<X/>;3;<Y/>)3<NEWLINE/>
</CORNERS>
</POLYGON>
-----<NEWLINE/>
Polygone_2<POLYGON name="Polygone 1"><NEWLINE/>
Corners:<SPACE/><NUM_CORNERS/><NEWLINE/>
<CORNERS corners="1">(X,Y):<SPACE/>4(<X/>;4;<Y/>)4<NEWLINE/>
</CORNERS>
</POLYGON>
-----<NEWLINE/>
Polygone_3<POLYGON name="Polygone 1"><NEWLINE/>
Corners:<SPACE/><NUM_CORNERS/><NEWLINE/>
<CORNERS corners="0">(X,Y):<SPACE/>5(<X/>;5;<Y/>)5<NEWLINE/>
</CORNERS>
</POLYGON>
-----<NEWLINE/>
Polygone_3<POLYGON name="Polygone 1"><NEWLINE/>
Corners:<SPACE/><NUM_CORNERS/><NEWLINE/>
<CORNERS corners="1">(X,Y):<SPACE/>6(<X/>;6;<Y/>)6<NEWLINE/>
</CORNERS>
</POLYGON>
-----<NEWLINE/>
Polygone_4<POLYGON name="Polygone 1"><NEWLINE/>
Corners:<SPACE/><NUM_CORNERS/><NEWLINE/>
<CORNERS corners="0">(X,Y):<SPACE/>7(<X/>;7;<Y/>)7<NEWLINE/>
</CORNERS>
</POLYGON>
-----<NEWLINE/>
Polygone_4<POLYGON name="Polygone 1"><NEWLINE/>
Corners:<SPACE/><NUM_CORNERS/><NEWLINE/>
<CORNERS corners="1">(X,Y):<SPACE/>8(<X/>;8;<Y/>)8<NEWLINE/>
</CORNERS>
</POLYGON></MESSAGE></MESSAGE>

```

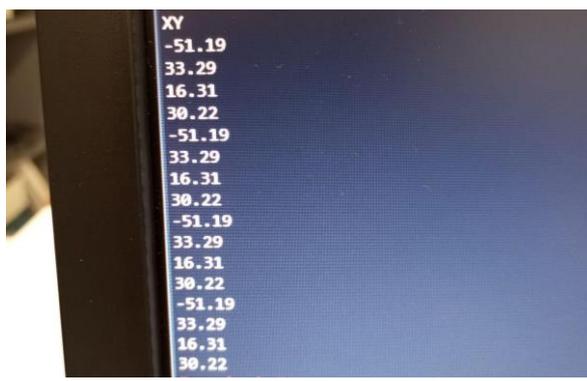
Les caractères surlignés en vert sont ceux qui ont été rajoutés pour récupérer les données situées entre. Pour chaque extrémité de la fonction polygone on a les chiffres de 1 à 8.

La ligne de code surlignée en bleu est l'orientation du carton en fonction de l'axe vertical de la caméra

Après le traitement on a en sortie :



C'est la première partie de la chaîne de sortie qui représente les indices qui correspondent à chaque caractère



C'est la deuxième partie de la chaîne de sortie qui représente les coordonnées en mm des extrémités du carton

Il est ensuite possible d'obtenir la taille du carton et son orientation.

6. E.Déroulement des travaux. (partie commune)

Séances	Noa	Lukas	Yann
1	Analyse du cahier des charges	Analyse du cahier des charges	Analyse du cahier des charges
2	Caractérisation du champs de vision de la caméra	Caractérisation du champs de vision de la caméra	Caractérisation du champs de vision de la caméra
3	Elaboration du BDD	Elaboration du BDD	Elaboration du BDD
4	Premier test éclairage	Premier test éclairage	Elaboration des IBD
5	Premier test éclairage	Premier test éclairage	Elaboration du programme python et sick
6	Prise en main du programme de la caméra	Choix de l'éclairage optimal (choix des LEDs, couleurs)	Elaboration du programme python et sick
7	Essaie de détecter le carton avec le logiciel caméra.	Choix de l'éclairage optimal (choix des LEDs, couleurs) et modélisation du support éclairage	Elaboration du programme python et sick
8	Détection du carton et modélisation du support caméra	Modélisation du support éclairage et choix définitif des LEDs	Elaboration du programme python et sick
9	Modélisation du support caméra	Modélisation du support éclairage	Elaboration du programme python et sick
10	Modélisation du support caméra	Modélisation du système complet	Elaboration du programme python et sick
11	Impression du support caméra et études de coût	Études de coût	Études de coût
12	Assemblage de la maquette	Assemblage de la maquette	Assemblage de la maquette

13	Test de la maquette et notice d'utilisation	Test de la maquette et notice d'utilisation	Test de la maquette et notice d'utilisation
14	Validation du système	Validation du système	Validation du système

7. F. Conclusion (partie commune)

En conclusion, le projet a abouti et réussi.

Nous avons réussi à détecter le carton sur la bande transporteuse, on y a rajouté un éclairage extérieur afin de limiter le bruit ambiant, quand la caméra est bien réglée la caméra détecte le carton en moins de 300ms.

Le programme écrit en PYTHON permet d'extraire la position en X et Y du carton, le logiciel SICK permet de détecter le carton.

Le support de la caméra permet de la régler en hauteur et en largeur grâce au profilé sur lequel elle est fixé, puis grâce au réglage de la distance focale, on peut permettre la détection du carton sans que le tapis ne dérange durant la détection.

La seule tâche qui n'a pas pu être réalisée est le tableau Excel et les calculs supplémentaires de coordonnées.