

**Système de surveillance basé sur une caméra
rotative**



Etudiants :

JOUANNEAU Maxime HEDIN Louis
VALENTIN Célian FOURNIER Thomas
SOURDRILLE Nathan MOUMSIK Yassine

Enseignant-responsable du projet :

LAGHMARA Hind

Date de remise du rapport : **13/06/24**

Référence du projet : **STPI/P6/2024 – 020**

Intitulé du projet : **Système de surveillance basé sur une caméra rotative**

Type de projet : **expérimental**

Objectifs du projet :

Créer un système de surveillance capable de détecter une intrusion, de la prendre en photo et de l'afficher sur un écran.

Mots-clefs du projet :

Surveillance / Modélisation / Programmation / Expérimentation

TABLE DES MATIERES

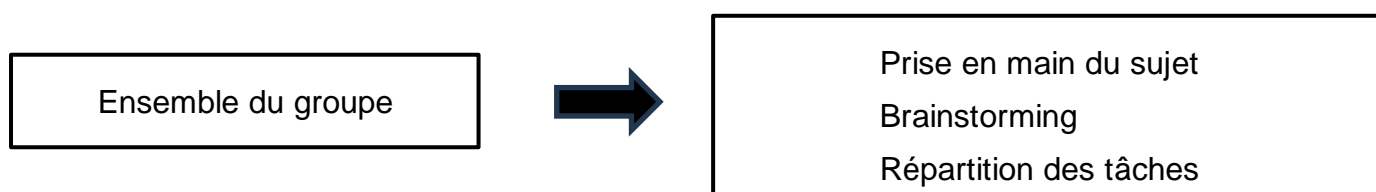
1. Méthodologie / Organisation du travail	5
1.1. Semaines 1 et 2.....	5
2.2 Semaines 3 à 14.....	5
2. Choix et Commande du matériel.....	6
2.1. Choix du matériel.....	6
2.2. Commande du matériel	7
4. Travail realise et resultats.....	8
4.1 Modélisation 3D.....	8
4.1.1 Modélisation	8
4.1.2 Impression et assemblage.....	9
4.2 Programmation.....	11
4.2.1 Confection de la partie moteur et capteur : code et montage.....	11
4.2.2 Programmation gestion affichage.....	13
5. Problèmes rencontrés	14
6. Conclusions et perspectives	15
7. Remerciements	16

Introduction :

Ce projet expérimental représente une manière de travailler différente de ce dont nous avons l'habitude : **créer un système de A à Z**. Nous avons donc mélangé de nombreuses compétences (programmation, logistique, modélisation, expériences...) pour arriver au fonctionnement du prototype final. Celui-ci doit être capable de fonctionner en autonomie une fois branché sur secteur afin de capturer la présence d'un potentiel intrus. Pour cela, le système s'appuie sur une détection de mouvement qui déclenche la prise d'une photo directement enregistrée et accessible.

1. METHODOLOGIE / ORGANISATION DU TRAVAIL

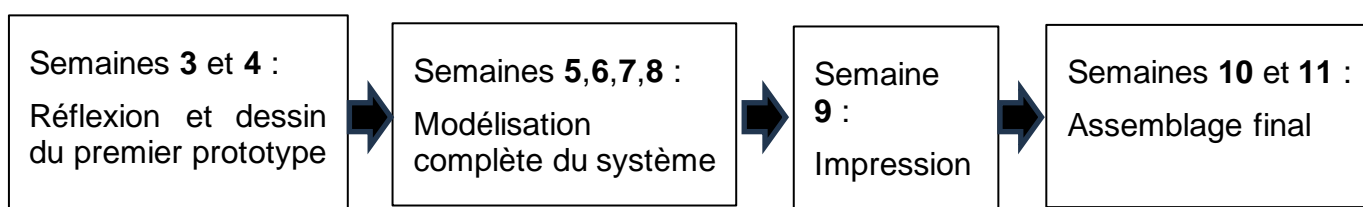
1.1. Semaines 1 et 2



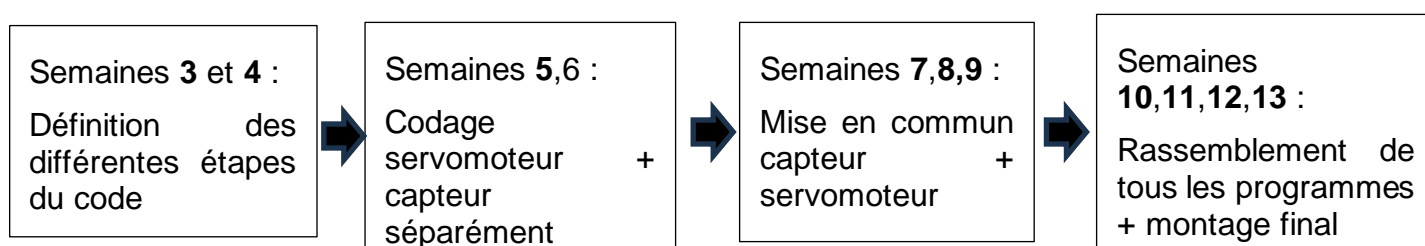
2.2 Semaines 3 à 14

Les semaines suivantes, le travail fut théoriquement réparti de la manière suivante :

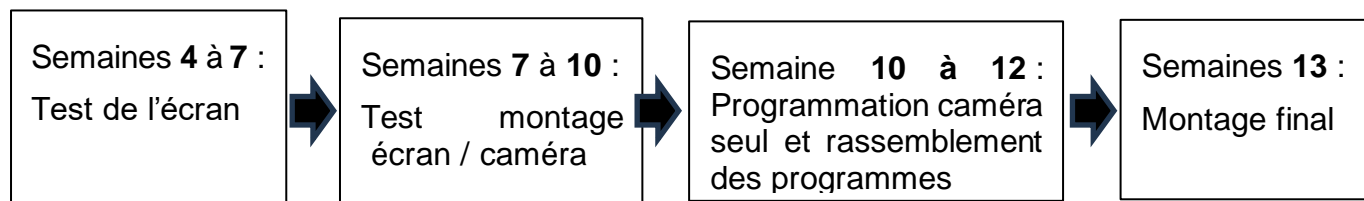
- **Organisation théorique de la Modélisation 3D** (Louis HEDIN et Célian VALENTIN)



- **Organisation théorique de la programmation servomoteur + capteur** (Maxime JOUANNEAU et Yassine MOUMSIK)



- **Organisation théorique de la programmation affichage** (Thomas FOURNIER et Nathan SOURDRILLE)



2. CHOIX ET COMMANDE DU MATERIEL

2.1. Choix du matériel

L'objectif de notre projet étant de réaliser une caméra de surveillance rotative intelligente, avec des contraintes de temps, de budget et de compétence, le choix du matériel était crucial. Nous avons principalement pris en compte la compatibilité entre les différents composants et leur coût, tout en respectant le cahier des charges.

Pour notre caméra de surveillance, nous avons sélectionné les composants suivants :

- Une caméra
- Un détecteur de mouvement
- Un servo-moteur
- Une alimentation
- Une LED
- Une carte Arduino Uno

Caméra :

La caméra devait remplir plusieurs critères : elle devait être économique, compatible avec une carte Arduino Uno et avec l'écran LED. Elle devait également respecter des critères de taille pour faciliter l'assemblage du système.

Détecteur de mouvement :

Le détecteur de mouvement devait être économique et avoir une grande portée de détection en distance euclidienne. Idéalement, il aurait un angle de détection faible, mais celui-ci pouvait être limité manuellement. De plus, il devait être compatible avec Arduino, tant au niveau de la bibliothèque que de l'alimentation.

Servo-moteur :

Le servo-moteur, responsable du mouvement rotatif de la caméra, devait générer un couple suffisant pour permettre la rotation de la partie supérieure de la caméra. Il devait également être économique et compatible avec Arduino.

Alimentation :

L'alimentation devait être économique et compatible avec Arduino.

LED :

Une LED classique.

Arduino Uno :

La carte de contrôle Arduino était quasiment imposée. Une version plus complexe ou plus adaptée aurait peut-être pu être trouvée, mais la carte Arduino Uno avait l'avantage d'être relativement simple pour des débutants.

Buzzer :

Un buzzer classique, émet un signal sonore.

2.2. Commande du matériel

Liste des composants sélectionnés :

Composant	Lien	Prix	Point Fort
Servomoteur	https://www.gotronic.fr/art-servomoteur-s07nf-25182.htm	12.4€	Gros couple nécessaire pour supporter la partie haute de la caméra
Caméra	https://www.lextronic.fr/camera-couleur-ov7670-pour-arduino-64731.html	9.6€	Compatible Arduino + faible prix
Capteur Mouvement	https://www.gotronic.fr/art-detecteur-infrarouge-gravity-sen0018-19369.htm#compte_desc	4.9€	Longue portée + faible prix
Arduino Uno	https://www.gotronic.fr/art-arduino-uno-a000066-12420.htm	23.90€	
Alimentation	https://www.gotronic.fr/art-adaptateur-ps1220n-25293.htm	14.9€	Compatible Arduino
Platine	(fournie par l'INSA)	0€	
Ecran	https://www.gotronic.fr/art-ecran-1-8-rb-tft1-8-26567.htm#compte_desc	12.8€	Petite dimension + bonne qualité visuelle
Buzzer	(fourni par l'INSA)	0€	
LED	(fournie par l'INSA)	0€	

A propos du servomoteur, nous avons hésité avec le modèle suivant : <https://www.gotronic.fr/art-servomoteur-180-ser0049-32636.htm> car il est 2 fois moins cher. Cependant, il possède un couple bien inférieur à celui choisi : 1.2kg.cm contre 5.5kg.cm.

A noter que certaines pièces étant déjà disponibles à l'INSA, nous avons largement respecté le budget de 100€.

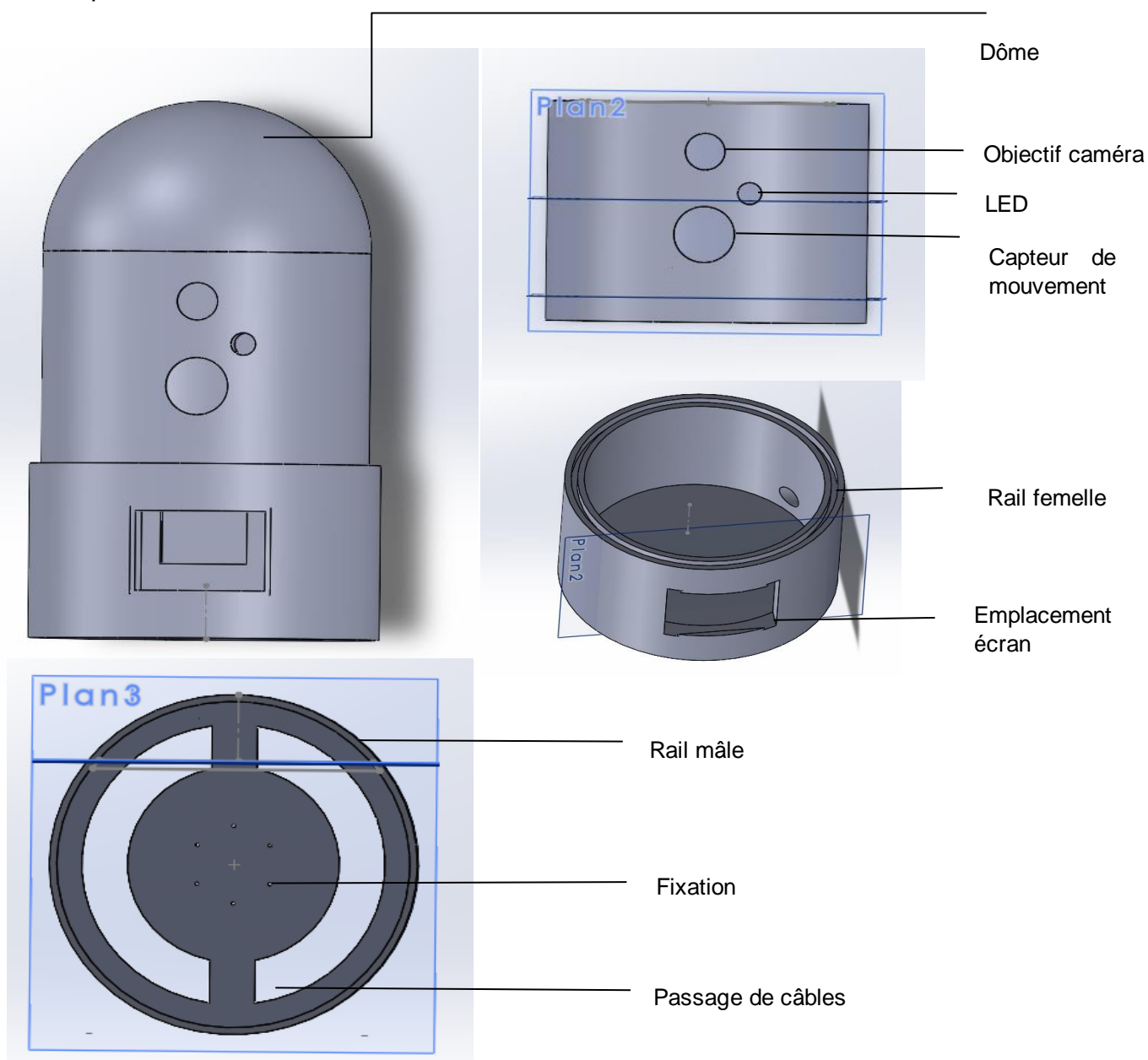
4. TRAVAIL REALISE ET RESULTATS

4.1 Modélisation 3D

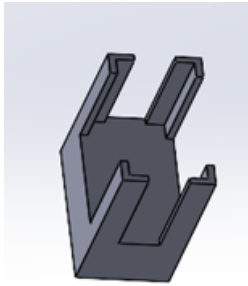
4.1.1 Modélisation

Pour modéliser l'armature 3D de la caméra nous avons utilisé le logiciel SOLIDWORKS que nous avons découvert en début d'année grâce à l'option CTI3.

Après de nombreuses réflexions et de nombreux essais non concluants, nous avons proposé la version suivante à l'ensemble du groupe afin de la valider pour l'impression :



Dans la partie **inférieure** de notre système, on retrouve les pièces suivantes :

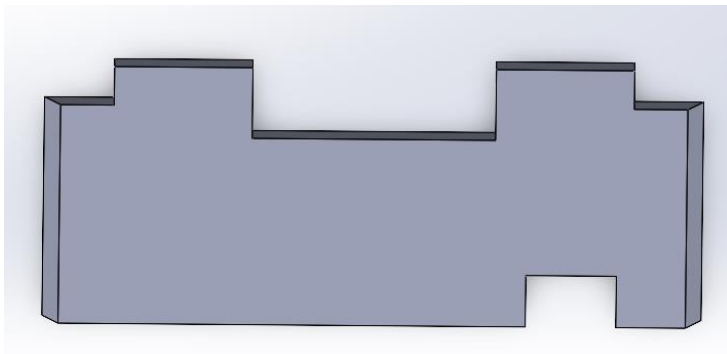


Support permettant
d'encastrer le moteur



Mur permettant de fixer la carte
électronique

Dans la partie supérieure de notre système, on retrouve la pièce suivante :



Mur permettant de fixer
la caméra

4.1.2 Impression et assemblage

Lors de notre demande d'impression auprès du technicien, nous avons réalisé que nos pièces possédaient des délais trop importants pour l'imprimante à notre disposition, en cela nous avons réduit au maximum les pièces en modélisant séparément tous les composants au sein d'une seule pièce.

Malgré cela, deux de nos pièces dépassaient encore les délais, c'est pourquoi après de nombreux échanges nous avons décidé avec le technicien de lancer l'impression malgré tout.

Finalement, malgré quelques détails de précision, l'impression s'est bien déroulée et les délais théoriques ont été plus ou moins respectés.

Plusieurs de ces détails nous ont toutefois fait dépasser des délais :

- Les trous initialement prévus pour fixer la partie supérieure de la caméra au servomoteur ne se sont pas faits. Nous avons donc percé la caméra nous-même.
- Les contours des deux armatures principales cylindriques n'étaient pas lisses, le jeu que nous avons laissé afin d'obtenir une liaison pivot entre les deux n'était donc plus suffisant. Par conséquent, nous avons limé les deux pièces en question afin que la rotation puisse avoir lieu.



Absence de trous
pour la fixation du
servomoteur



Imperfection sans
conséquence



Contour
non-lisse

Nous avons ensuite pu assembler notre système comme prévu initialement :



4.2 Programmation

4.2.1 Confection de la partie moteur et capteur : code et montage

Avant-propos :

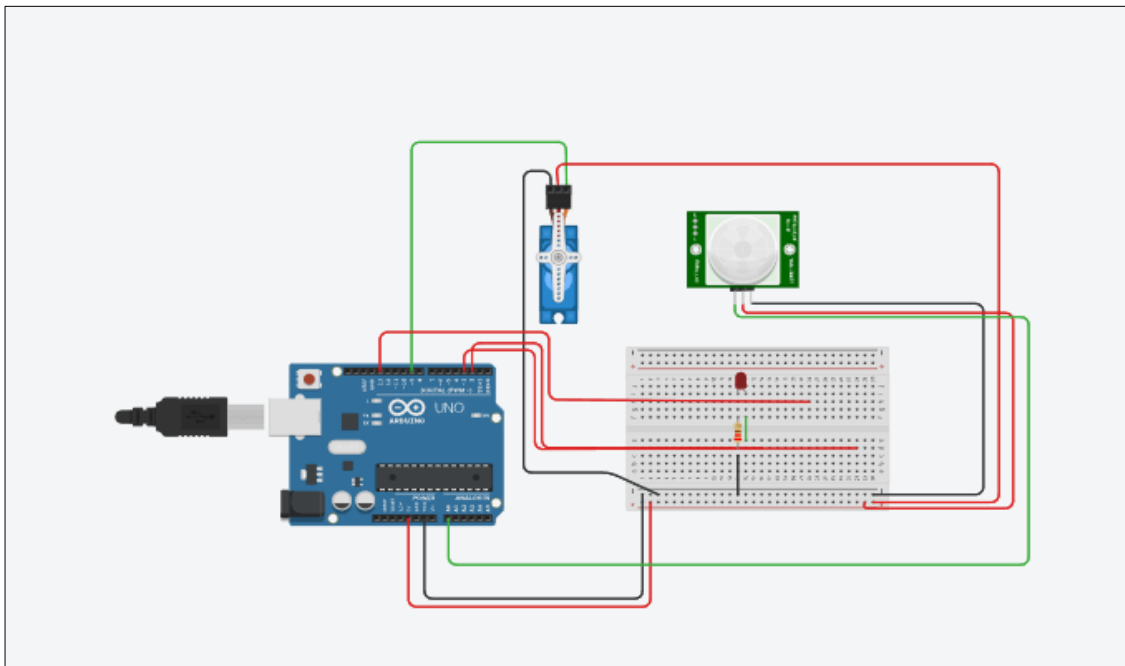
Pour faire fonctionner la caméra, l'idée est la suivante : le détecteur de mouvement, renvoyant un signal binaire (oui ou non), nous indique s'il détecte un mouvement. Le problème étant que nous ne pouvons pas déterminer où le mouvement a été détecté. Pour pallier ce problème, l'idée était de limiter l'angle de détection du capteur de sorte que, s'il détecte un mouvement, celui-ci soit directement en face du capteur. De cette manière, en alignant la caméra au capteur et en les faisant tourner ensemble, si le capteur détecte un mouvement, la caméra est activée.

Confection :

La première étape consistait à être capable de contrôler séparément la rotation du moteur et comprendre la détection du capteur. Chaque composant est branché sur la carte Arduino avec trois câbles : un pour l'alimentation, un pour la masse et un pour le signal. Pour le capteur, le câble signal est un câble d'entrée (input). Si le capteur détecte un mouvement, il envoie un signal TRUE à la carte. Pour le moteur, le câble de signal est un câble de sortie (output), où la carte envoie des informations de rotation au moteur.

Une fois ces premiers principes maîtrisés, nous avons réalisé un montage avec les deux composants. L'idée était de faire tourner le moteur jusqu'à ce qu'un mouvement soit détecté. Les premiers tests ne furent pas concluants en raison d'un problème avec le capteur de mouvement qui renvoyait toujours TRUE. Après quelques recherches dans la fiche technique du capteur, nous avons découvert qu'un potentiomètre au dos du capteur permettait de régler sa sensibilité. Problème résolu.

Schéma du montage avec les deux composants :



Code :

Le code sous Arduino est relativement simple pour les tâches basiques. Pour la partie gérant le moteur et le capteur, l'utilisation de variables booléennes permet un contrôle optimal. Nous utilisons une variable "rotation" indiquant par sa valeur si le moteur tourne ou non, et une variable "détection" indiquant si la détection du capteur est prise en compte ou non.

Amélioration du système :

Une fois cette partie préliminaire importante terminée, de nouveaux objectifs et de nouvelles contraintes se sont ajoutées. Premièrement, nous avons dû remplacer le servo moteur utilisé pour les tests par un nouveau, qui après plusieurs tests, ne fonctionnait pas.

Après l'acquisition d'un nouveau servo-moteur, censé marcher, de nouveaux problèmes sont apparus. Ce nouveau moteur, était à fonctionnement continue. Contrairement à l'ancien, qui avait besoin d'une donnée 'angle' en entrée pour fonctionner, celui-ci fonctionne avec une donnée 'pulse', lui indiquant une vitesse de rotation. De plus, ce nouveau moteur n'est pas limite en rotation, et peut tourner sur lui-même indéfiniment.

Des modifications dans le code ont alors été faites. Premièrement, il était important de limiter cette rotation du moteur, pour re créer le va et vient du moteur précédent. Un premier constant fut que le moteur ne tournait pas à la même vitesse dans les deux sens. Pour résoudre ce problème, le moteur effectue dix incréments dans un sens, puis 13 dans l'autre. Cela permet une constance dans la rotation.

Ce problème simple, peut en générer bien d'autres, notamment si le système tend à tourner plus dans un sens que l'autre, les câbles s'emmêleraient et la caméra ne serait plus fonctionnelle.

Pour ce qui est du capteur, une étape clé à la réussite du projet était de réussir à brider son angle de détection. Après le teste de différents matériaux adhésifs, un scotch noir a fini par faire l'affaire. Il présentait l'avantage de parfaitement brider l'angle, d'être facile à installer / retirer et était peu coûteux.

Photo du capteur une fois l'angle bridé :



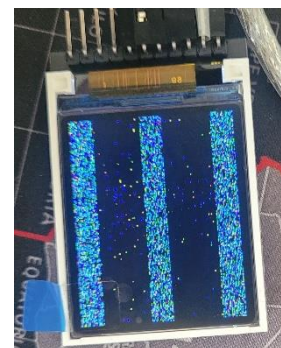
Partie finale : mise en commun du système :

Pour pouvoir aisément mettre en commun cette partie de détection et de mouvement du système, le code a été pré-aménagé pour permettre une compatibilité optimale. En effet, des fonctions « nulle » furent insérées dans le code, pour garantir son fonctionnement avec celle-ci. Une fois le circuit entier constitué, la fusion des 2 codes nécessitent simplement de remplacer ces fonctions par les bonnes.

4.2.2 Programmation gestion affichage

Pour continuer, cette partie du code a pour objectif de connecter la caméra et la carte Arduino afin de prendre une photo lorsqu'un intrus est détecté par le capteur, cela dans le but d'afficher ensuite l'image de l'intrus sur un écran.

Tout d'abord, notre approche a été de connecter uniquement l'écran à la carte Arduino, afin de tester son bon fonctionnement. Ainsi, son fonctionnement était correct, nous avons notamment la capacité de contrôler, à partir de la carte Arduino, des affichages à l'écran, sa couleur par exemple. Cependant, nous avons rencontré des difficultés lors de l'affichage direct d'une image, car il est complexe de la transmettre directement à partir de l'IDE Arduino. En cela, une seconde approche a été de passer par une carte micro SD qui servirait d'intermédiaire entre l'écran et la carte Arduino, afin de sauvegarder l'image, mais c'est un processus plus complexe encore, que nous avons abandonné.



Erreur écran

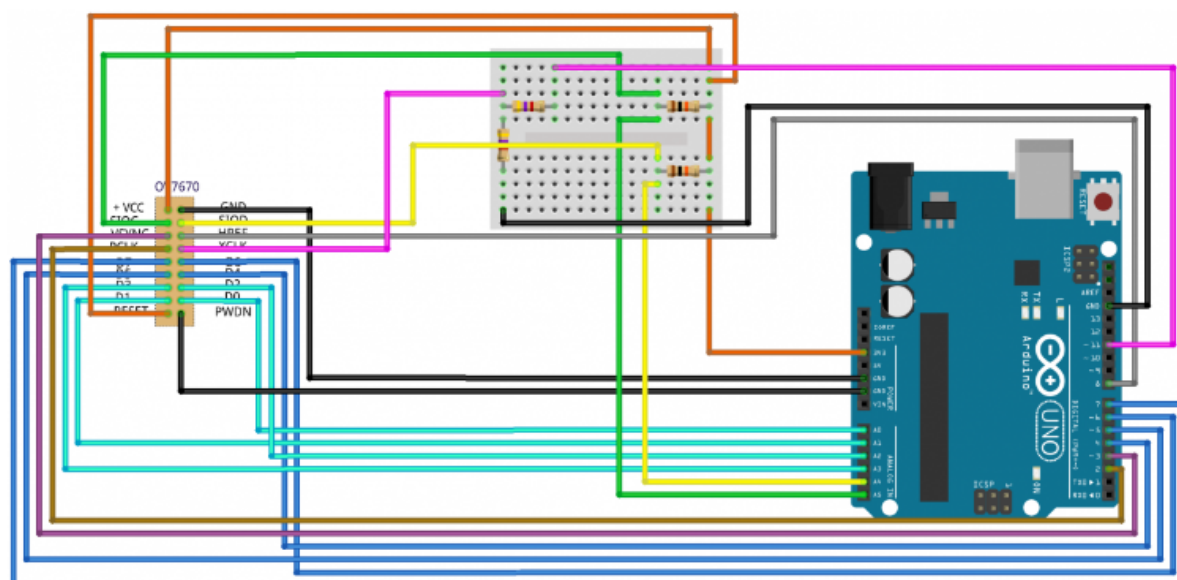
Ainsi, une fois la caméra reçue, notre dessein a été de tester directement les deux composants (écran, caméra) ensemble, car il s'agit de l'objectif initial. Cependant des difficultés sont apparues, dans un premier temps le fait qu'il n'existe pas de bibliothèques universelles destinées à cette caméra mais uniquement des bibliothèques créées pour des

projets indépendants, ce qui signifie qu'elles étaient conçues uniquement dans le cadre de ces projets et donc non générales et adaptables. De plus, les branchements essentiels pour la caméra étaient assez complexes et nécessitent un grand nombre de câbles et plusieurs résistances. Il était donc assez difficile d'entrevoir, durant les phases de test, si la caméra était non fonctionnelle ou si le montage n'était pas adapté.

Finalement, nous nous sommes résolus à laisser de côté l'affichage sur un écran de l'image photographié par la caméra et avons opté pour l'enregistrement de l'image directement sur un ordinateur. En effet, l'ensemble des broches de la caméra et de l'écran étaient nécessaires à leur fonctionnement, or ces branchements occupaient la majeure partie des broches de la carte Arduino, ce que nous ne pouvons pas mettre en place dans l'optique du branchement d'autres composants du projet, notamment le servomoteur. Alors, nous avons décidé de remplacer le composant écran par le composant buzzer, qui permet de fournir un signal sonore lors de la détection d'un intrus.

En ce qui concerne le mode de communication entre la caméra et la carte arduino, nous utilisons le protocole I2C qui permet une communication en seulement 2 broches, la broche SDA (Serial Data Line) qui permet l'envoi et réception de données ainsi que la broche SCL (Serial Clock Line) qui synchronise la transmission des bits de données.

Schéma montage caméra :



5. PROBLEMES RENCONTRES

Nous avons rencontré différents problèmes dont les principaux sont listés ci-dessous :

- Le servomoteur initialement commandé ne fonctionnait pas et nous avons dû le remplacer par un servomoteur disponible à l'INSA. Cela a entraîné la modification du support moteur initialement modélisé.

- Initialement, la caméra devait posséder un écran permettant d'afficher les éventuelles infractions capturées. Cependant, faute de réussite, notre prototype ne permet que d'afficher ces images sur un écran d'ordinateur via une connexion filaire.
- Le servomoteur possède un défaut : il tourne plus dans un sens que dans l'autre. Pour régler ce problème nous l'avons fait tourner plus de fois dans un sens que dans l'autre. En effet, cela évite que notre caméra effectue un 360 degrés et que les câbles s'emmêlent.
- Le nombre de ports de la Arduino Uno n'était pas suffisant pour notre projet. En effet, nous pensions pouvoir sélectionner certains ports de la caméra afin de libérer certains ports de la Arduino. Ce n'était finalement pas le cas et nous avons donc utilisé une deuxième carte Arduino.
- Enfin, lorsque nous avons voulu fusionner les programmes des deux parties, nous avons été confrontés au problème suivant : lors de l'appel de la fonction qui active la caméra pour prendre une photo, le code bascule sur le petit logiciel qui permet de visionner et de sauvegarder les images. Cependant, une fois basculé, le code n'arrivait plus à appliquer les instructions sur les autres composants ce qui rendait notre système obsolète.

Pour pallier à ce problème, nous avons décidé de faire fonctionner deux cartes Arduino en simultanément, une avec la partie moteur-capteur et l'autre avec la caméra. Cela a permis de résoudre notre problème et de faire fonctionner notre système.

6. CONCLUSIONS ET PERSPECTIVES

En conclusion, bien que notre projet ne soit pas abouti à 100%, nous pouvons être fier du travail réalisé. En effet, il y a moins de 6 mois, aucun de nous ne possédait aucune des compétences nécessaires à la réalisation de ce projet (CAO, Arduino, traitement d'images, gestion autonome de projet...).

Toutefois, nous avons réussi à fournir un prototype fonctionnel.

Sur le plan personnel, nous aurons appris à gérer un projet assez conséquent de manière autonome. Chacun d'entre nous aura également développé des compétences techniques poussées selon le travail qu'il a réalisé.

Si ce projet venait à être poursuivi, nous aimerions l'aboutir en rajoutant l'écran initialement prévu et nécessaire à l'affichage des images directement sur notre caméra.

Pour finir, il est important de rappeler dans tout échec se cachent des réussites et que nous avons beaucoup appris de nos erreurs durant ce projet.

7. REMERCIEMENTS

L'ensemble du groupe remercie Madame LAGHMARA pour son aide et ses conseils tout au long du projet.

Nous souhaitons aussi remercier Monsieur WILLIAMS sans qui l'impression 3D n'aurait été possible.

Merci également à Monsieur JOLLY pour son accompagnement dans les parties techniques et manuelles du projet.