

Tournicayou

Construction d'un appareil photogrammétrique dans le cadre de l'UE Atelier Fablab LU3SV062

Informations

- **Étudiants :**
 - Zoé Albenga
 - Lucas Gorini
- **Encadrants :**
 - Responsables de l'UE : Fabrice Minoletti et Pierre Théry
 - Responsable du projet : Pierre Théry
- **Parcours :** SV/ST : UE_FabLab : LU3ST062
- **Date :** S6



Objectifs

Notre objectif de réalisation pour cette UE est la d'un socle pour la photogrammétrie. Celui-ci aurait comme application dans un premier temps la réalisation "d'image en 3D" pour certaines pierres ponces, puis à usage plus large par la suite.

Construction

Étape 1 : S'approprier Arduino ainsi que le projet avec la modélisation de notre appareil à photogrammétrie.

Étape 2 : *Réflexion/Réalisation des premières pièces par modélisation 3D ainsi qu'un travail de recherche sur le code du moteur pas à pas.*

Étape 3 : *Modification et ajustement des pièces du projet et réalisation du code pour la (les) caméra(s).*

Étape 4 : *Assemblage de toutes les pièces modélisées ainsi que des deux codes, afin de réaliser les premiers tests.*

Étape 5 : *Réaliser les premiers tests avec un objet à prendre en photo avec un logiciel de photogrammétrie permettant la réalisation de l'objet en 3D.*

Étape 6 : *Paramétrer une application permettant la réalisation des photos.*

Étape 7 : *C'est déjà un beau projet*

Matériel

- Carte Arduino Uno
- Module caméra ArduCam mini 5MP Plus OV5642
- Moteur NEMA
- Motor shield (--> branchement pour le moteur et la caméra)
- Câblages
- Logiciel de modélisation 3D

Résultats

Lors de ce dernier semestre de licence nous avons eu la chance travailler sur un projet sur la photogrammétrie, le but étant de réaliser le maximum d'étapes prédéfinies (pas obligatoirement la fin du projet).

Journal de bord

30/01/26 _ Premières manipulations

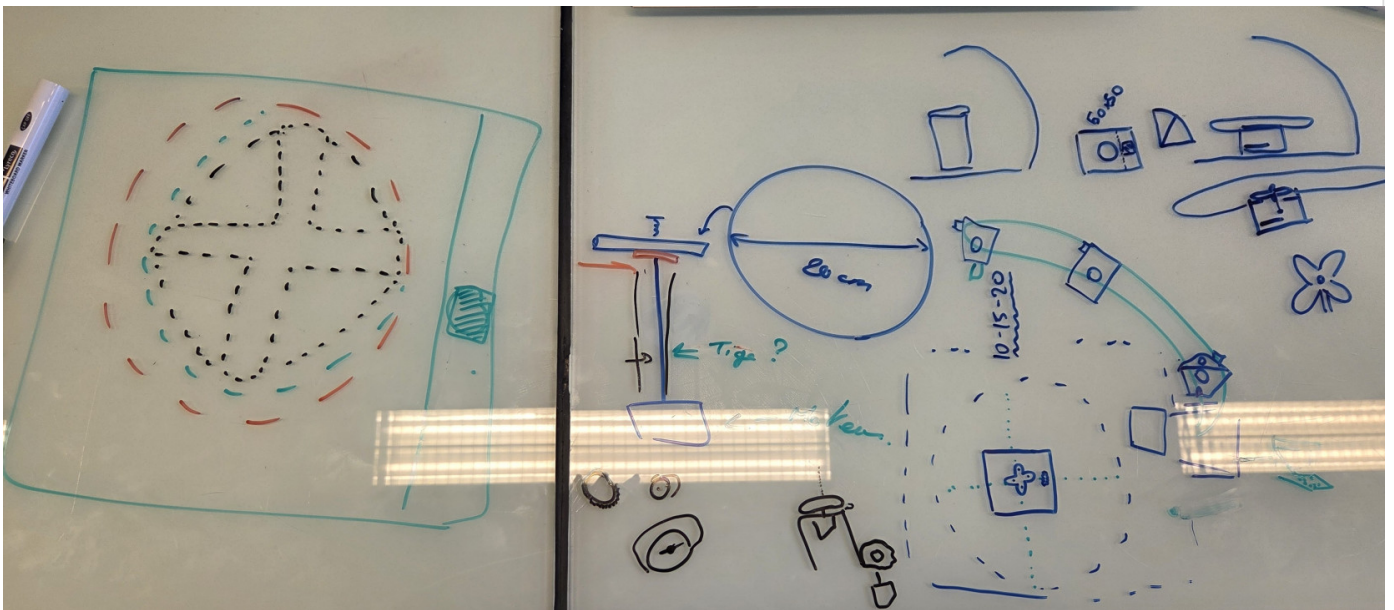
1ère Séance : Sujet Photogrammétrie (Schéma) + Découverte de ARDUINO

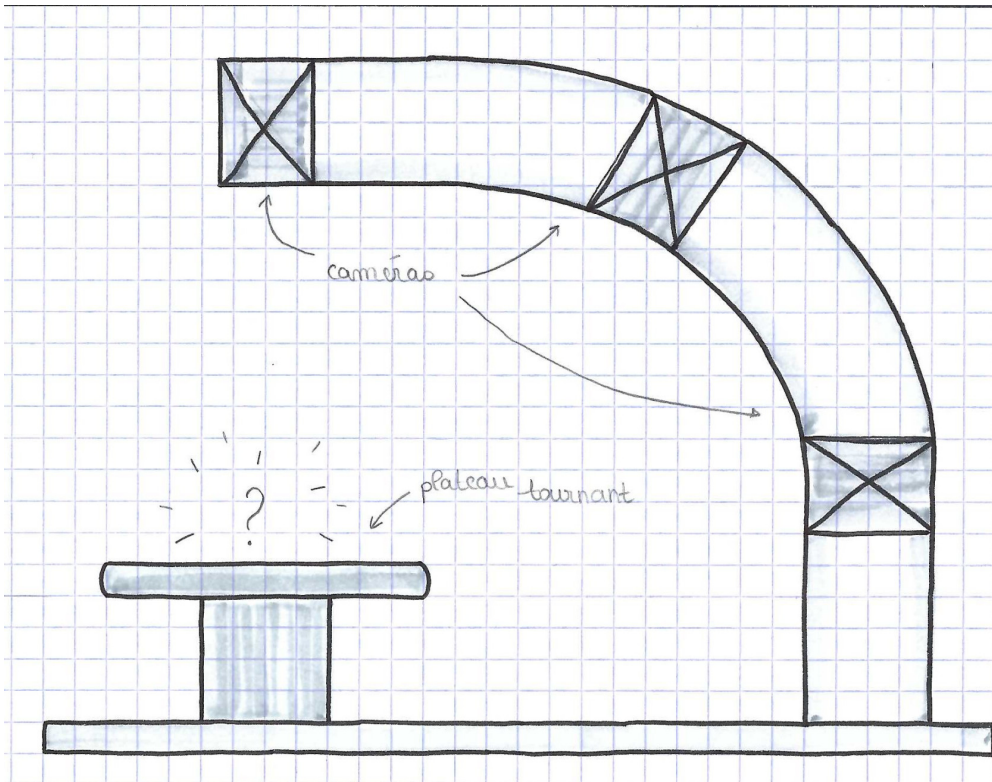
- Premières réflexions réalisation sur la de ce support à photogrammétrie.
- Réalisation de plusieurs dessins "idées" (voir photos).
- Première manipulation avec le logiciel ARDUINO, premier codage. Bien difficile....

```
// the loop function runs over and over again forever
void loop() {
  for (int t=1; t<=3; t++) {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(dur_high_short);           // wait for a second
    digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
    delay(dur_low);
  }
  for (int t=1; t<=3; t++) {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(dur_high_long);           // wait for a second
    digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
    delay(dur_low);
  }
  for (int t=1; t<=3; t++) {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(dur_high_short);           // wait for a second
    digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
    delay(dur_low);
  }
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(2000);                        // wait for a second
}
```



Codage avec Arduino, led clignotant SOS en morse.

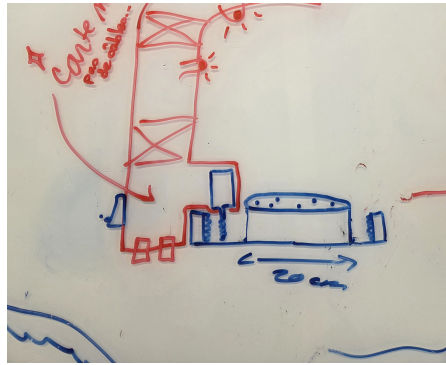
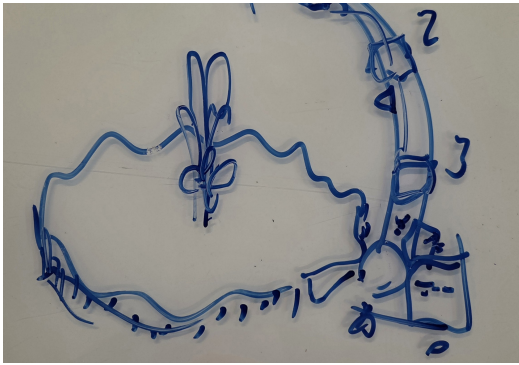




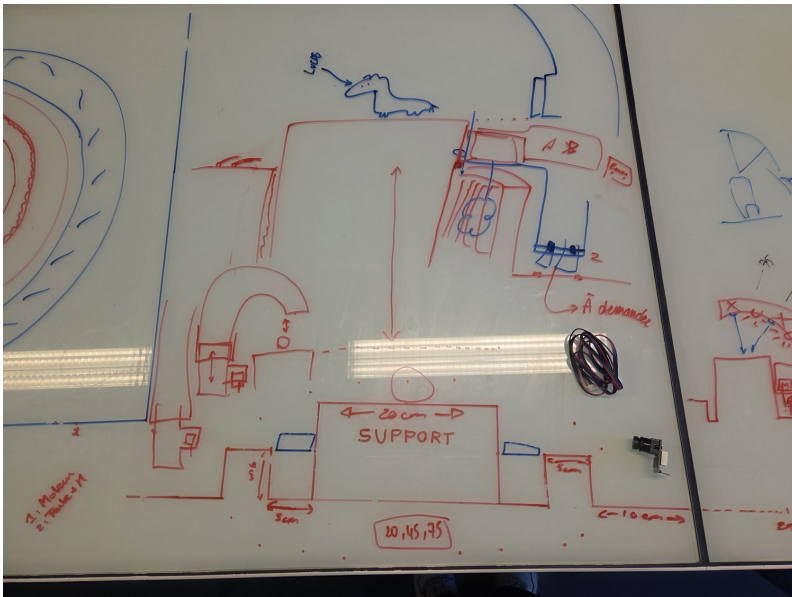
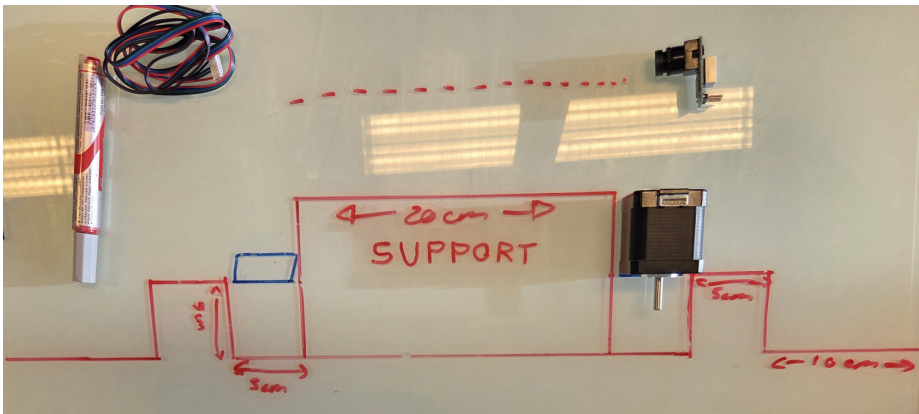
06/02/26_Mise en place du plan d'attaque pour le projet

2ème Séance : Retour sur le support (Schémas) + Mise en place des différentes étapes

- Reprise de la structure du support de photogrammétrie --> bras tournant autour du support !! et plus un plateau tournant => permet de faire la photogrammétrie sur des objets plus léger
- Mise en place des différentes étapes :
 - Moteur + programmation
 - Moule à Tarte
 - Arduino + caméras + affichage ordi à voir ?
 - Bras : support caméra + LED
 - Bonus : écran + angles + focus caméras



*



13/02/2026_1ères modélisation 3D et manipulation Arduino pour le moteur

3ème Séance : Gratuit Gratuit Freecad + Arduino

FREECAD

Apprentissage des fonctions de base du logiciel FreeCAD. Plusieurs essais infructueux de la conception de la pièce crantée permettant le support du moteur autour de l'objet à photographier.

Vidéos youtube visionnées pour le modèle FreeCAD :

- <https://www.youtube.com/watch?v=3wPDnRFfgEg>
- <https://www.youtube.com/@cadgab/playlistshttps://www.youtube.com/@cadgab/playlists>

LE COULOMMIERS EST TERMINÉ (18/02) - Côte Mauvaise

Pour créer le coulommiers, qui fait office d'engrenage interne permettant le déplacement du bras, il faut créer deux corps différents sur FreeCAD. Les corps sont créés sur le module **Part design**

- Un cercle de 600 mm de diamètre et 200 mm de haut
- Un engrenage à développer de $D_p = 495$ mm, $M = 2,5$ mm, $z = 198$ et 200 mm de haut

Une fois les deux corps formés, il faut utiliser une booléenne pour former le rouage final grâce au module **Part**

Attention : Les valeurs inscrites sont les valeurs erronées et le fichiers FCStd a été réduit en hauteur

Fichiers STL et FCStd de la pièce décrite :

[Coulommiers-Support.stl](#)

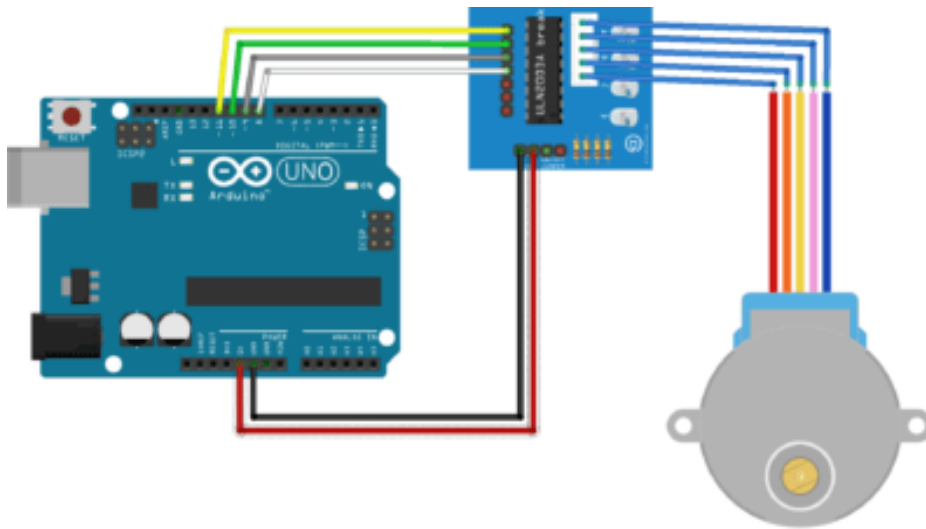
[Coulomier final.FCStd](#)

ARDUINO

Familiarisation réalisation avec le logiciel Arduino et d'un premier montage :

- Matériels :
 - Arduino Uno + câble USB de liaison Arduino/Ordinateur
 - Câbles de connexion
 - Moteur pas à pas 28 BYJ-48

- Schéma de connexion :



- Code :

- Ce code permet au moteur de réaliser une rotation dans le sens des aiguilles d'une montre, attendre 1 seconde(s), puis de faire une rotation dans le sens contraire des aiguilles d'une montre, puis de nouveau attendre 1 s.

```
#include <Stepper.h>

double stepsPerRevolution = 2048;

Stepper myStepper(stepsPerRevolution, 8, 10, 9, 11); // Pin inversion to make the library work

void setup() {
  myStepper.setSpeed(10);
  Serial.begin(9600);
}

void loop() {
  // 1 rotation counterclockwise:
  Serial.println("counterclockwise");
  myStepper.step(stepsPerRevolution);
  delay(1000);

  // 1 rotation clockwise:
  Serial.println("clockwise");
  myStepper.step(-stepsPerRevolution);
  delay(1000);
}
```

```
}
```

Une librairie Stepper.h existe.

- Succès de ce premier montage et code associé. Manipulation du code nous permettant d'obtenir une rotation dans le sens des aiguilles d'une montre avec arrêt de 1 s. (suppression de la première partie de la void loop)
- Rapidement essayez de changer de vitesse = ÉCHEC --> comprendre pourquoi à la prochaine séance.

--> Prochaine étape : Réaliser le montage avec ajout d'un potentiomètre permettant de contrôler la vitesse de rotation ainsi que le sens (--> potentiomètre vers la droite = sens des aiguilles d'une montre ; potentiomètre à gauche = sens inverse)

20/02/2026_ Réalisation du Coulommiers et 1ères ébauches code moteur

4ème Séance : Gratuit ecad + Arduino

Partie Freecad :

Modélisation du rouage pour le moteur pas à pas, et reprise de dimension du coulommiers. Pour calculer le nombre de dents de chaque rouage, externe (Moteur) comme interne (Coulommiers) nous avons suivi l'équation suivante :

$$D_p/M = z$$

D_p => Le diamètre primitif ; M => Module du rouage ; z => Nombre de dents

Pour que deux engrenages puissent fonctionner ensemble **il faut respecter 3 conditions :**

- Ils doivent avoir un module égal
- Ils doivent avoir un angle de pression égal (par convention 20°)
- Leur diamètre primitif doit être tangent

Pour le dernier point, la distance entre le centre des rouages **$D_r = ((z_1 + z_2) * M) / 2$**

Pour le coulommiers :

La modélisation comporte deux corps.

- Un cercle de 300 mm de diamètre et 100 mm de haut
- Un engrenage à développé de $D_p = 250$ mm, $M = 2,5$ mm, $z = 100$ et 100 mm de haut

Pour former le rouage interne, il faut faire la différence des deux corps grâce à la booléenne dans le module **Part** . (cf <https://www.youtube.com/watch?v=3wPDnRFfgEg>)

Pour le rouage externe ce dernier doit être fixé sur la "tige" du moteur faisant 19 mm de long. Il faut creuser un cylindre de 5mm de diamètre au centre du rouage avec une corde de 3mm afin de respecter l'aspect de la tige.

La modélisation comporte deux corps.

- Un cylindre de 5mm de diamètre coupé d'une corde de 3mm et faisant 19mm de haut
- Un engrenage externe de $D_p = 45$ mm, $M = 2,5$ mm, $z = 18$ et 19 mm de haut

Impression 3D des pièces pour essais ultérieurs.

Fichiers STL et FCStd de la pièce décrite :

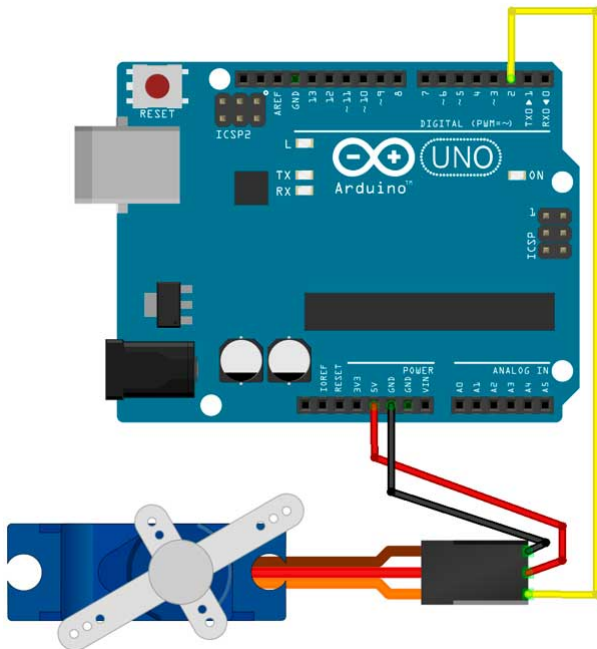
ARDUINO

Retour sur le montage réalisation et la du circuit de la semaine dernière = ÉCHEC. Nous n'avons pas trouvé la raison pour laquelle la réussite de la semaine passée n'était plus réalisable. (Changement de moteur, vérification et retour sur les explications du code, changement de code, vérification du fonctionnement de la carte Arduino uno...)

Réalisation d'un autre montage avec le moteur Servo :

- Matériels :
 - Arduino Uno + câble USB de liaison Arduino/Ordinateur
 - Câbles de connexion
 - Microservo à moteur pas à pas SG90

- Schéma de connexion :



- Code :
 - Ce code permet au moteur de réaliser une rotation dans le sens des aiguilles d'une montre, attendre 1 seconde(s), puis de faire une rotation dans le sens contraire des aiguilles d'une montre, puis de nouveau attendre 1 s.

Après plusieurs échecs, et des changements de codes, succès de la manipulation. Notre moteur se déplace de l'angle 0 à l'angle 90, puis à celui 180 avant de revenir à 0 et de recommencer sa boucle. Après chaque déplacement celui-ci fait une pause de 1 s. Notre moteur avance degré par degré.

--> Prochaine étape : Modification de notre code pour réaliser un mouvement de 20° en 20° avec 1 s de délai entre chaque déplacement.

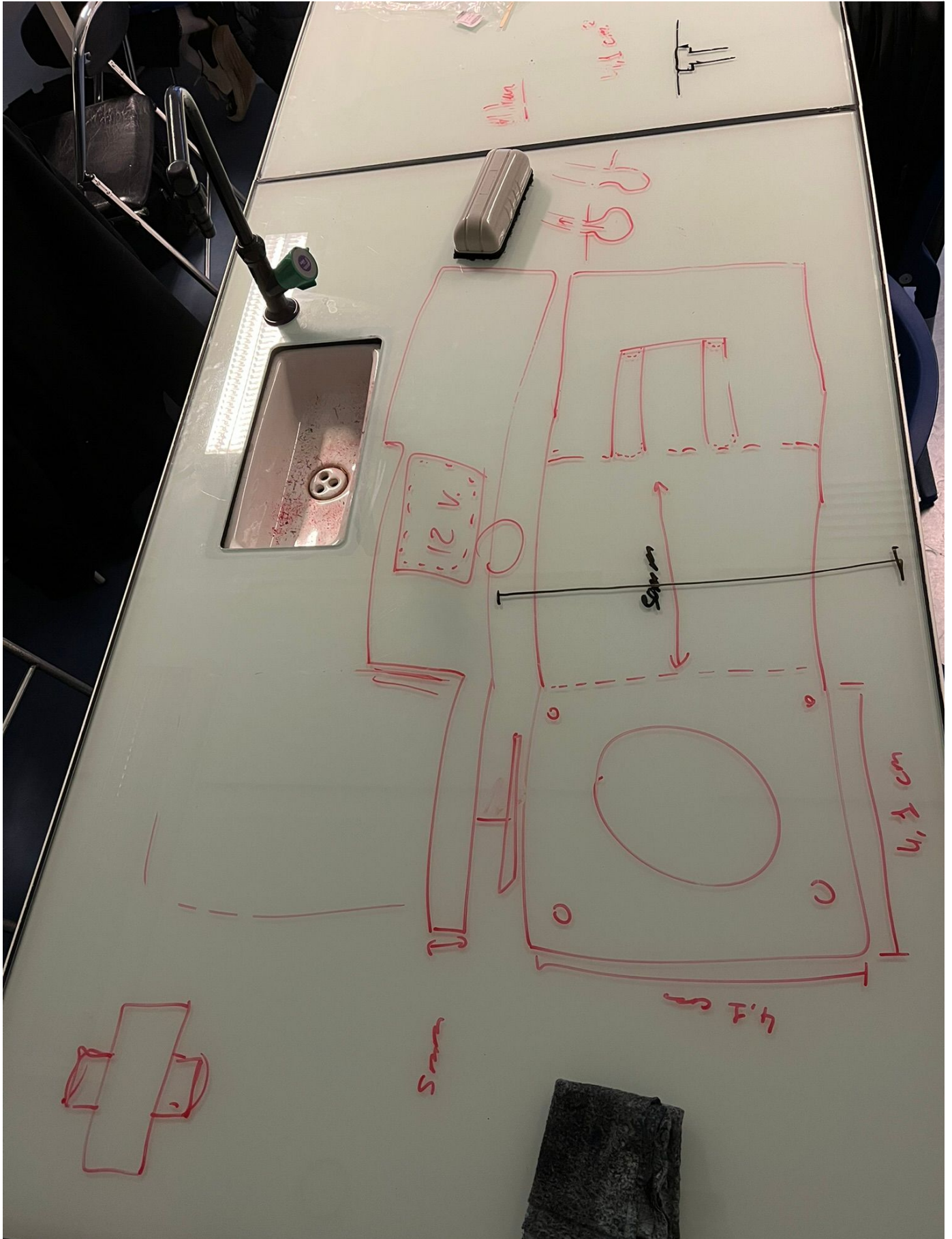
27/02/2026_Support moteur et 1er test moteur NEMA (pas à pas)

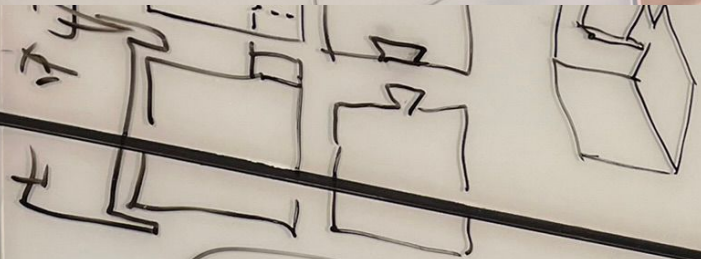
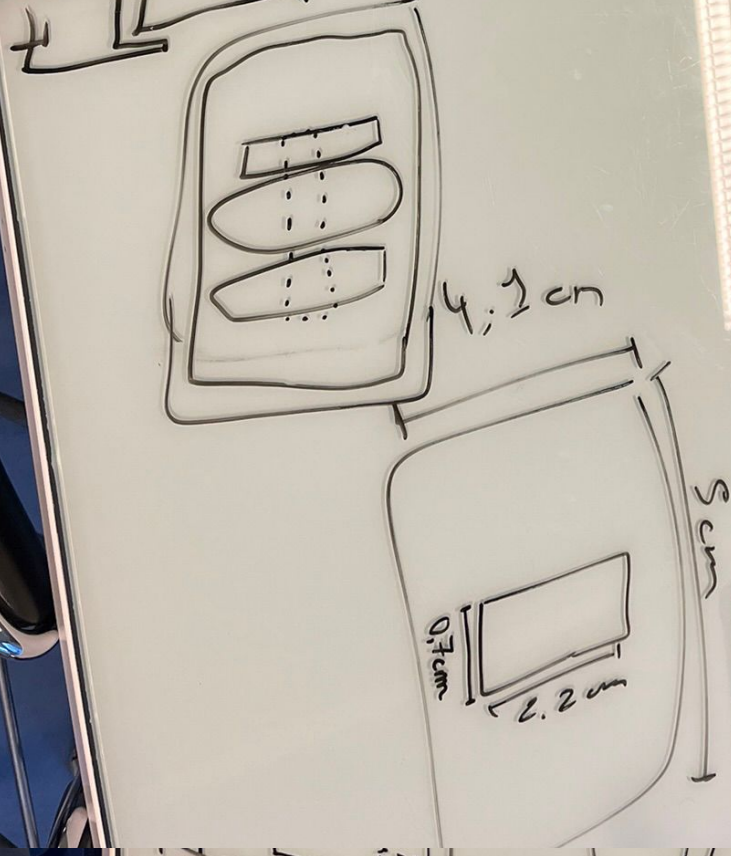
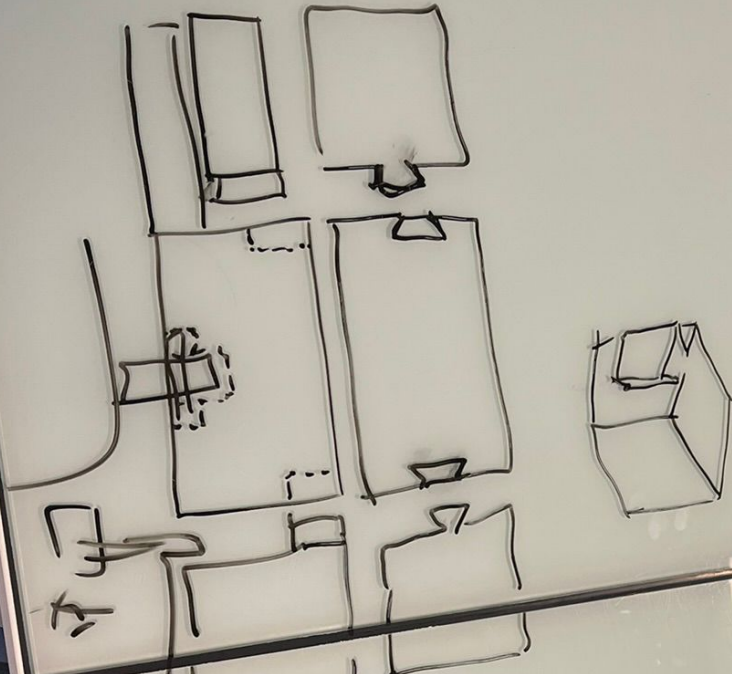
Objectif de la séance : **Finaliser le code arduino pour le moteur, tester les pièces 3D, conception du bras**

Nous avons commencé la réalisation des plans quant à la partie qui tourne autour du coulommiers. Trois soirées :

- Le support même du moteur, permettant sa fixation par vis.
- La partie centrale, dotée d'une roue permettant au socle tournant d'avancer sur les coulommiers.
- Le support bras, avec des fentes permettant l'insertion du bras photo et des emplacement pour piles et carte arduino.

L'idée est de créer deux pièces s'assemblant comme un puzzle. La première composée du socle moteur et de la partie centrale. L'autre du socle bras.





ARDUINO

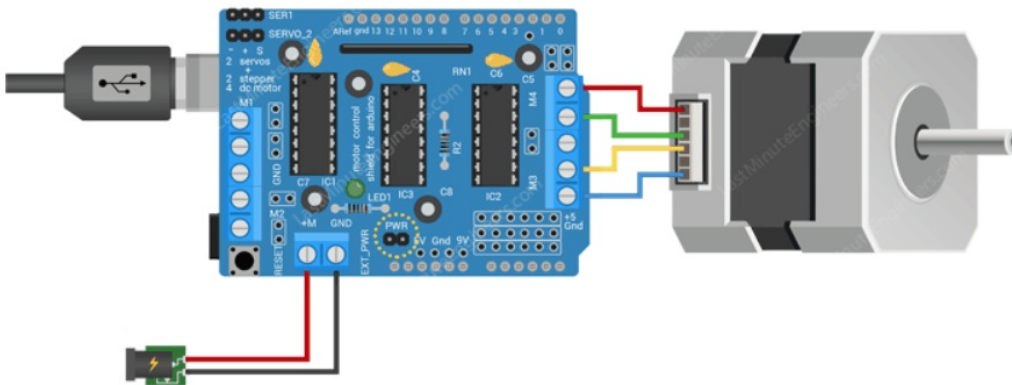
Une partie des coulommiers dentés et le rouage du moteur sont désormais imprimés ! Aujourd'hui sur un pu tester le gros moteur de 12 V, qui sera notre moteur final, avec donc le rouage modélisé par Lucas.

J'ai cramé la batterie brique de 12 V... Surement un problème dans mon montage.

Réalisation d'un autre montage avec le moteur NEMA 17, moteur pas à pas de 12 V :

- Matériels :
 - Arduino Uno + câble USB de liaison Arduino/Ordinateur
 - Câbles de connexion
 - Moteur Shield
 - Moteur NEMA 17 pas à pas

- Schéma de connexion :



- Code :
 - Ce code permet au moteur de réaliser une rotation dans le sens des aiguilles d'une montre, attendre 1 seconde(s), puis de faire une rotation dans le sens contraire des aiguilles d'une montre, puis de nouveau attendre 1 s.

```
#include <AFMotor_R4.h>
```

```
const int stepsPerRevolution = 200; // nombre de pas du moteur
```

```

AF_Stepper motor(stepsPerRevolution, 2);

void setup() {
  Serial.begin(9600);
  motor.setSpeed(10); // vitesse en RPM
}

void loop() {

  // Calcul du nombre de pas pour 20°
  int stepsFor20Degrees = stepsPerRevolution / 18;

  // Rotation sens horaire (aiguilles d'une montre)
  motor.step(stepsFor20Degrees, FORWARD, SINGLE);

  delay(1000); // arrêt de 1 seconde

}

```

- Ce code fonctionne plutôt très bien, il réalise donc une rotation dans le sens des aiguilles d'une montre ; fait 20° puis fait une pause pour la photo.

```

#include <AFMotor_R4.h>

const int stepsPerRevolution = 200; // nombre de pas du moteur

AF_Stepper motor(stepsPerRevolution, 2);

void setup() {
  Serial.begin(9600);
  motor.setSpeed(10); // vitesse en RPM
}

void loop() {

  int stepsFor20Degrees = stepsPerRevolution / 18; // 20° ≈ 11 pas

  // Un tour complet = 360° = 18 segments de 20°
  for(int i = 0; i < 18; i++) {

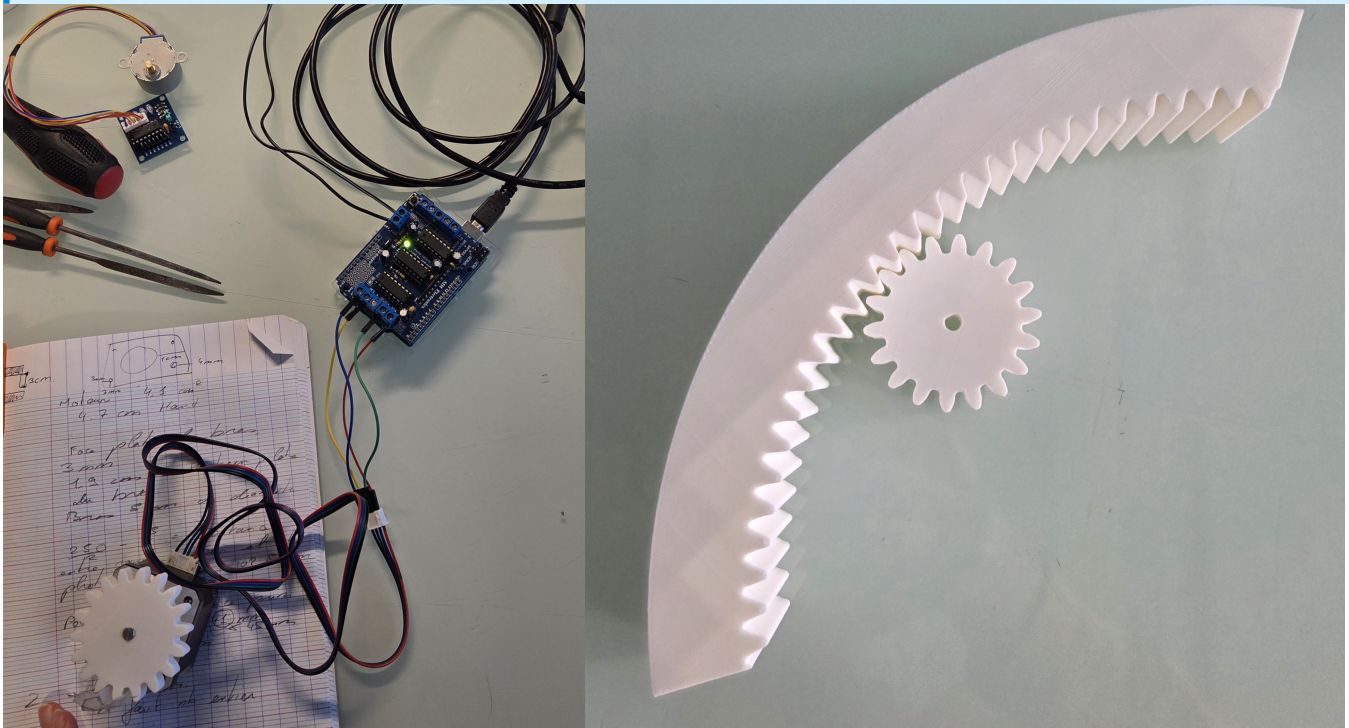
```

```
motor.step(stepsFor20Degrees, FORWARD, SINGLE); // avance 20°
delay(2000); // pause de 2 secondes
}

// Pause longue à la fin du tour
delay(10000); // pause de 10 secondes
}
```

Ce 2ème code est très probablement mieux mais manque de fluidité de mouvement : il ne réalise pas forcément tout le temps 20° degré mais parfois moins et de manière saccadé. Il réalise donc 11 pas pour faire les 20° demandées ; pause de 2 secondes ; puis reprends pour faire un tour complet de 20 par 20 ; pause de 10 s.

Ce code rend le moteur extrêmement bruyant en plus du mouvement saccadé.



06/03/2026_Conception du socle moteur

Partie conception 3D : **Le socle moteur**

Conception du socle moteur sur Freecad. Les mesures sont encore à vérifier. Le socle est composé d'un assemblage Freecad de deux pièces. La première correspond au support même du moteur, et dispose de 4 trous de fixation permettant par vis et un cinquième plus grand permettant au bras du moteur de passer. (Le moteur est fixé par la surface où se situe le bras). L'autre pièce est un rectangle permettant l'insertion d'une roue à billes, servant au roulement du support moteur sur les coulommiers. Cette pièce possède un trou permettant le passage d'une tige maintenant la roue en place. La conception de cette tige en 3D est réalisée, mais reste à voir pour sa conception (Impression 3D, métal, autre ?)(Au final Pierre a arrangé une vis pour la faire maintenir la roue). Le rectangle possède un creux en forme de puzzle servant à l'insertion du socle du bras, devant encore être modélisé.

Fichiers STL et FCStd de la pièce décrite :

[Assemblage socle moteur + Tige-Assembly.stl](#)

[Assemblage socle moteur + Tige.FCStd](#)

13/03/2026_Modélisation de pièce soutien bras, fluidification du code moteur et recherches caméras

Lucas : J'ai limé pendant deux heures le bloc, puis début de la conception des plan pour la pièce arrière.

Partie conception 3D : Création de la pièce arrière

La pièce arrière a pour but de maintenir la machine autour du coulommiers et de servir de contre-poids au moteur. Celle-ci est composée de deux parties principales, formant une pièce unique :

1. La partie la plus importante est le gros boîtier principal. Celui-ci fait ... de large, ... de haut et ... de long. À l'intérieur une grande cavité est laissée pour permettre de tenir les différents composants électriques. L'avant de la pièce possède la pièce de puzzle complémentaire au support moteur, le rond de à un diamètre de 15,5mm. À partir du haut, au bout de ...mm, la face avant recule de 4mm pour que permette à un roulement à billes de rouler sur le bord du coulommiers. Sur la partie basse, deux roues sont placés sur un arc de cercle fictif de 735mm de diamètre. Cet arc a le même foyer que le coulommiers, les roues ainsi inclinées d'un angle de $6,60^\circ$ à partir de l'axe central de la pièce servent à la pièce à tourner plus facilement.
2. La deuxième partie est sur la partie haute de la pièce. Il s'agit d'un socle ayant à l'avant de trous de 8,25mm de rayon permettant l'encoche du bras photographique.

Fichier STL et FCStd de la pièce décrite :

Pièce arrière.stl

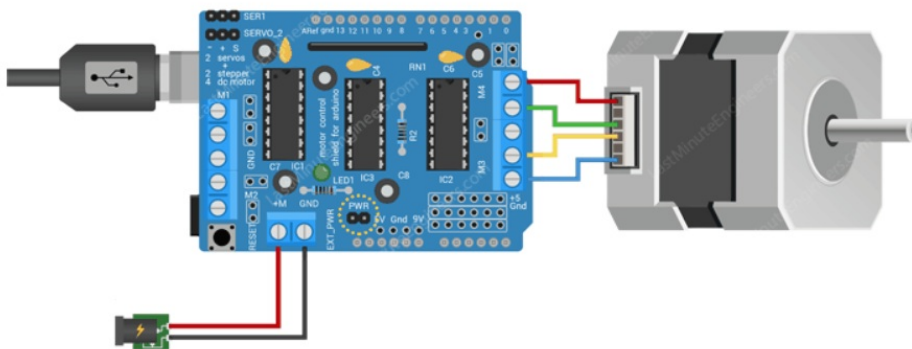
Pièce arrière.FCStd

ARDUINO

Aujourd'hui j'ai revu le code pour notre moteur afin que celui-ci réalise 20° sur les coulommiers et non par rapport à l'axe du moteur.

Réalisation d'un autre montage avec le moteur NEMA 17, moteur pas à pas de 12 V :

- Matériels :
 - Arduino Uno + câble USB de liaison Arduino/Ordinateur
 - Câbles de connexion
 - Moteur Shield
 - Moteur NEMA 17 pas à pas
- Schéma de connexion :



- Code :
 - Petite partie calcule pour réaliser le code et essayer de trouver une façon pour que le moteur soit un mouvement plus fluide et plus saccadé et bruyant.

coulommiers = 100 bosses ; rouage = 18 bosses ; moteur pas à pas = 200 pas / tour

$100/18=5.55$ tours --> le bébé rouage effectuera donc 5.5 tour pour réaliser un tour complet du tour interne du coulommiers.

$200 \times 5.55 = 1111$ pas --> le moteur lui devra donc réaliser 1111 pas pour faire 360° donc un tour complet du coulommiers.

$1111 \times 20/360 = 61.7$ pas --> le moteur devra réalisé environ 62 pas pour réaliser 20° sur le coulommiers.

62 pas x 18 positions = 1 116 pas # 1 111 pas ! => possibilité d'alternance entre 62 et 61 pas pour obtenir pile 360°

- Pour éviter que le moteur soit saccadé on peut voir pour augmenter la vitesse de déplacement/ ou la diminuer.

```
#include <AFMotor_R4.h>

const int stepsPerRevolution = 200; // pas moteur
const int stepsFor20Degrees = 62; // calcul avec engrenage

AF_Stepper motor(stepsPerRevolution, 2);

void setup() {
  Serial.begin(9600);
  motor.setSpeed(10); // vitesse RPM
}

void loop() {

  // 360° / 20° = 18 positions
  for(int i = 0; i < 18; i++) {

    motor.step(stepsFor20Degrees, FORWARD, SINGLE);

    Serial.print("Position : ");
    Serial.println(i + 1);

    delay(2000); // pause entre chaque mouvement
  }

  Serial.println("Tour complet terminé");

  delay(10000);
}
```

- 1er code testé : **FONCTIONNEL !**
- Testez un autre code avec une alternance de 61 et 62 pas pour avoir pile 360° et voir si celui ci est plus fluide ou non.

```
#include <AFMotor_R4.h>

const int stepsPerRevolution = 200;

AF_Stepper motor(stepsPerRevolution, 2);

int positionIndex = 0;

// Séquence corrigée pour obtenir exactement 360°
int stepSequence[18] = {
  62,62,62,62,62,62,
  62,62,62,
  61,
  62,62,62,62,62,
  62,62,
  61
};

void setup() {

  Serial.begin(9600);

  motor.setSpeed(10);

  Serial.println("Systeme de rotation initialise");

}

void loop() {

  int steps = stepSequence[positionIndex];

  motor.step(steps, FORWARD, INTERLEAVE);

  Serial.print("Position : ");
  Serial.println(positionIndex + 1);

  positionIndex++;
}
```

```
delay(2000); // pause de 2 s entre chaque mouvement
```

```
if(positionIndex >= 18)
```

```
{
```

```
  positionIndex = 0;
```

```
  Serial.println("Tour complet termine");
```

```
  delay(10000); // pause de 10 s après le tour complet
```

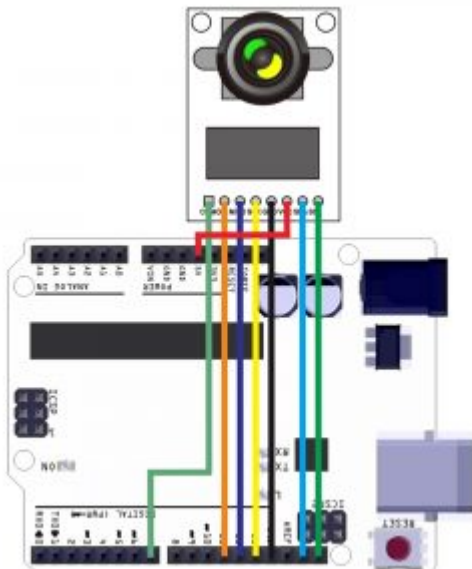
```
}
```

```
}
```

Toujours un problème de fluidité, voir pour améliorer le 1er code ou le 2ème.

Test de différentes vitesses : 5 et 30 plus ou moins OK, toujours saccadé et bruyant --> voir microstep ?

*Recherche pour les caméras : **Arducam Shield Mini 5MP Plus** :*



- o Carte SD nécessaire ?

- Dans ce montage, le module caméra 5MP serait utilisé avec une carte Arduino UNO pour capturer des images à intervalles réguliers. Sans carte microSD, les photos ne peuvent pas être stockées directement sur le système embarqué.
- À la place, les données de l'image au format JPEG seront envoyées via la liaison série (USB) vers un ordinateur. L'Arduino déclenche une prise de vue, transmet les données binaires correspondantes, puis attend 30 secondes avant de recommencer (pour notre projet).
- Comme l'Arduino Uno possède une mémoire limitée, il est nécessaire de réduire la résolution des images afin d'assurer un fonctionnement stable. --> utilisation d'une autre carte ?
- Les images reçues sur l'ordinateur peuvent ensuite être reconstruites et visualisées à l'aide d'un programme externe, par exemple en Python.

Prochaine séance --> Test de 25 - 5° pour améliorer la fluidité + CAMÉRAS !
redemander les bibliothèques.

20/03/2026_Retour des recherches sur les caméras et conception du bras

On galère avec les caméras....

Partie conception 3D : **Création du bras photographique**

Ce bras s'encoche dans la partie haute de la pièce arrière. Il est conçu en une pièce unique (dû à la limitation de taille d'impression de l'imprimante 3D, la pièce a été scindé en deux pour les tests) en trois parties.

1. La partie basale est composée de deux tubes permettant l'insertion à la pièce arrière. Les encoches étant de 8,25mm de rayon, le tubes ont un rayon de 7,25mm.
2. Le bras incurvé, arc de cercle de 692 mm de diamètre. Il forme un angle de 72,70°. Afin d'alléger la structure, mais aussi de limiter l'utilisation de plastique, des triangles équilatéraux d'une arête de 34,1738mm sont disposés de façon alterne. Cinq triangles avec la pointe distal au support central, quatre avec une pointe qui lui est proximal.
3. Deux nouvelles encoches à 19,96° et 59,39° de 7mm de diamètre permettront l'insertion des caméras.

Le bras a due être sectionné pour l'impression !

Fichiers STL et FCStd de la pièce décrite :

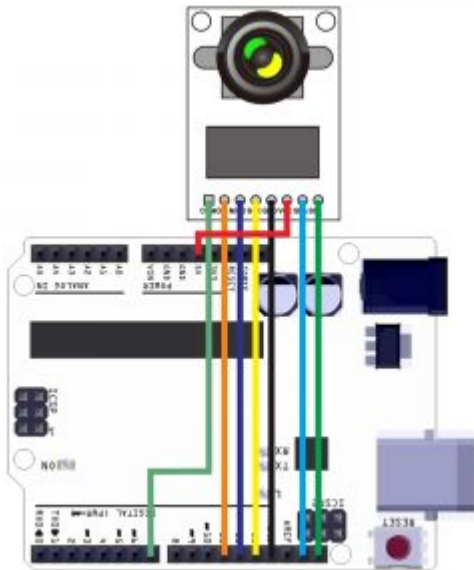
[Bras.stl](#)

27/03/2026_Comprendre la caméra...

Le bras du Tournicayou est imprimé !

!!!! Comprendre les caméras !!!!

- Matériels :
 - Arduino Uno + câble USB de liaison Arduino/Ordinateur
 - Câbles de connexion
 - Module caméra Mini 5MP Plus OV5642
- Schéma de connexion :



- Code :
 - Essais avec énormément de codes différents, beaucoup de difficultés pour ce qui est de comprendre l'utilisation des différentes fonctions lié à la caméra.

Super code pour notre caméra de l'enfer

```
#include <Wire.h>
#include <SPI.h>
#include <ArduCAM.h>
#include "memorysaver.h"

#if !(defined OV5642_MINI_5MP_PLUS)
  #error Please select the hardware platform and camera module in memorysaver.h
#endif

#define BMPIMAGEOFFSET 66

const int CS = 7;
bool is_header = false;
int mode = 0;
uint8_t start_capture = 0;

ArduCAM myCAM(OV5642, CS);

uint8_t read_fifo_burst(ArduCAM myCAM);

void setup() {
  uint8_t vid, pid;
  uint8_t temp;

  Wire.begin();
  Serial.begin(921600);

  Serial.println(F("ACK CMD ArduCAM Start! END"));

  pinMode(CS, OUTPUT);
  digitalWrite(CS, HIGH);

  SPI.begin();

  // Reset caméra
  myCAM.write_reg(0x07, 0x80);
  delay(100);
```

```
myCAM.write_reg(0x07, 0x00);
delay(100);

// Test SPI
while (1) {
  myCAM.write_reg(ARDUCHIP_TEST1, 0x55);
  temp = myCAM.read_reg(ARDUCHIP_TEST1);
  if (temp != 0x55) {
    Serial.println(F("ACK CMD SPI interface Error! END"));
    delay(1000);
  } else {
    Serial.println(F("ACK CMD SPI interface OK. END"));
    break;
  }
}

// Vérif caméra
while (1) {
  myCAM.wrSensorReg16_8(0xff, 0x01);
  myCAM.rdSensorReg16_8(OV5642_CHIPID_HIGH, &vid);
  myCAM.rdSensorReg16_8(OV5642_CHIPID_LOW, &pid);

  if ((vid != 0x56) || (pid != 0x42)) {
    Serial.println(F("ACK CMD Can't find OV5642 module! END"));
    delay(1000);
  } else {
    Serial.println(F("ACK CMD OV5642 detected. END"));
    break;
  }
}

myCAM.set_format(JPEG);
myCAM.InitCAM();
myCAM.OV5642_set_JPEG_size(OV5642_320x240);

delay(1000);

myCAM.clear_fifo_flag();
```

```
}

void loop() {
  uint8_t temp;

  if (Serial.available()) {
    temp = Serial.read();

    switch (temp) {

      case 0x10:
        mode = 1;
        start_capture = 1;
        Serial.println(F("ACK CMD CAM start single shoot. END"));
        break;

      case 0x11:
        myCAM.set_format(JPEG);
        myCAM.InitCAM();
        break;

      default:
        break;
    }
  }

  if (mode == 1) {
    if (start_capture == 1) {
      myCAM.flush_fifo();
      myCAM.clear_fifo_flag();
      myCAM.start_capture();
      start_capture = 0;
    }

    if (myCAM.get_bit(ARDUCHIP_TRIG, CAP_DONE_MASK)) {
      Serial.println(F("ACK CMD CAM Capture Done. END"));
      read_fifo_burst(myCAM);
      myCAM.clear_fifo_flag();
    }
  }
}
```

```

    }
  }
}

// --- Version corrigée pour Host App ---
uint8_t read_fifo_burst(ArduCAM myCAM) {
  uint8_t temp = 0, temp_last = 0;
  uint32_t length = myCAM.read_fifo_length();

  if (length == 0 || length >= MAX_FIFO_SIZE) {
    Serial.println(F("ACK CMD Size error. END"));
    return 0;
  }

  myCAM.CS_LOW();
  myCAM.set_fifo_burst();

  temp = SPI.transfer(0x00);
  length--;

  while (length--) {
    temp_last = temp;
    temp = SPI.transfer(0x00);

    if (is_header) {
      Serial.write(temp); // flux JPEG pur
    } else if ((temp == 0xD8) && (temp_last == 0xFF)) {
      is_header = true;
      Serial.write(temp_last);
      Serial.write(temp); // début JPEG
    }

    if ((temp == 0xD9) && (temp_last == 0xFF)) break; // fin JPEG
  }

  myCAM.CS_HIGH();
  is_header = false;
}

```

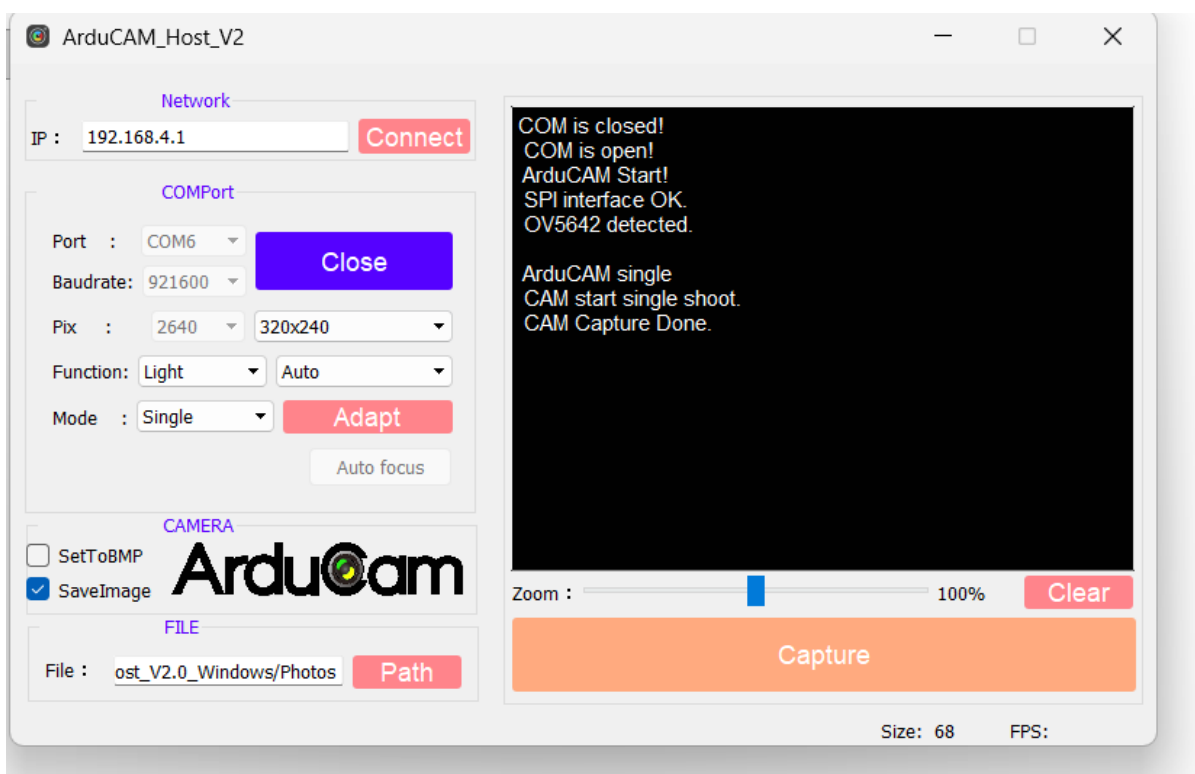
```
return 1;  
}
```

- Voici ci-dessus le dernier code utilisé pour obtenir des photos et qui est lié avec l'ArduCam Host App.
- Dernier test plutôt positifs avec l'ArduCam Host App mais toujours pas un succès...

D'après le logiciel on arrive à prendre des photos (ArduCam Host App --> photo ci-dessous)

Où sont les photos !? C'est le plus gros des problèmes, réussir à retrouver les photos et savoir si celle-ci se sont réellement enregistrées ou non, ou si plus généralement on arrive vraiment à prendre des photos.

Pour la prochaine séance --> résoudre le problème : où sont les photos !?!



Objectif : se débarrasser du problème CAMÉRA ! Où sont les photos sur l'ordinateur ? Les photos sont-elles prises ?

<https://www.arducam.com/camera-modules/5mp-ov5642/>

Fab a du fil à retordre

Le code photo fonctionne, il est possible de prendre des photos !!!!! On sait où elles sont !

Nos différents problèmes avec les photos maintenant qu'on sait où elles sont :

- Au bout d'un certain nombre de photos, il n'est plus possible d'en prendre (D'après Claude c'est un problème de FIFO où Host App/Arduino essaye de lire la photo avant qu'elle ne soit prise).
- Il faut trouver un moyen d'automatiser les photos pour un tour complet de coulommiers.
- Reste à savoir si fonctionne avec deux caméras.

Partie conception 3D : Révision du coulommiers

La première version du coulommiers avait un léger problème de dimension. Notamment avec ses 20 centimètres de haut. En réalité, le coulommiers ne nécessite que 4 centimètres de haut. Autre problème auquel faire face... 60 centimètre de diamètre sort du plateau d'impression de l'imprimante 3D.

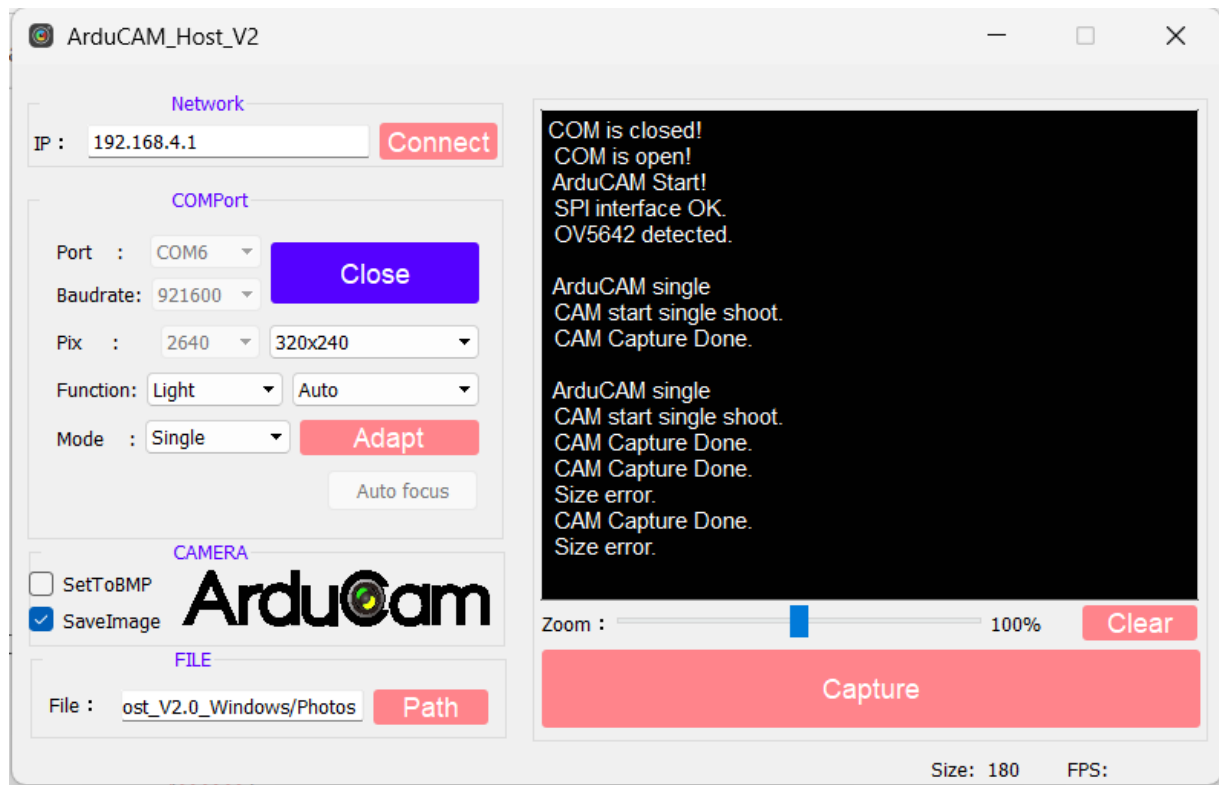
Ainsi le nouveau coulommiers ne fait "que" 4 centimètre de haut, est monté sur un un plateau support de 5 millimètres de diamètre permettant le roulement de la pièce arrière. La structure montante du coulommiers est creuse, permettant d'user de moins de plastique. Le coulommiers a été modélisé en quart, avec des pièces mâles femelle basales permettant un montage en puzzle. Ces structures sont ainsi disposées pour limiter un maximum au support moteur de rouler sur des failles. Cependant la partie mâle dépasse toujours du support d'impression.

La pièce est encore à revoir. Durant cette révision, le plateau support sera retravaillé afin de limiter au mieux l'usage de plastique.

Fichiers STL et FCStd de la pièce décrite :

[Coulommiers rerererererevisité-Cut.stl](#)

[Coulommiers rerererererevisité.FCStd](#)



10/04/2026_ Caméra OK = Réflexion sur la suite

!!!! Rappel important !!!! : Nous avons désormais des photos et on sait où on peut les retrouver !

Les différents problèmes de la dernière fois qu'on a réussi à régler :

- En reprenant le code de la dernière fois pour faire les photos, ce n'était pas possible de faire des photos multiples. On a donc cherché un autre code, que l'on a par la suite légèrement modifié afin de prendre plusieurs photos à la suite. Grâce à ce code on a donc pu prendre plusieurs photos d'affiler sans problème de FIFO mais également d'un sens à automatiser pour réaliser un tour complet du Coulommiers. On peut régler le nombre de photos que l'on veut réaliser et le temps d'attente donc entre chacune d'entre elles.

Code super méga ultra trop long pour caméra

```

// ArduCAM Mini demo (C)2017 Lee
// Web: http://www.ArduCAM.com
// This program is a demo of how to use most of the functions
// of the library with ArduCAM Mini camera, and can run on any Arduino platform.
// This demo was made for ArduCAM_Mini_5MP_Plus.
// It needs to be used in combination with PC software.
// It can test OV5642 functions.
//

// This program requires the ArduCAM V4.0.0 (or later) library and ArduCAM_Mini_5MP_Plus
// and use Arduino IDE 1.6.8 compiler or above
#include <Wire.h>
#include <ArduCAM.h>
#include <SPI.h>
#include "memorysaver.h"
//This demo can only work on OV5642_MINI_5MP_Plus platform.
//#if !(defined OV5642_MINI_5MP_PLUS)
// #error Please select the hardware platform and camera module in the
// ..libraries/ArduCAM/memorysaver.h file
//#endif
#define BMPIMAGEOFFSET 66
const char bmp_header[BMPIMAGEOFFSET] PROGMEM =
{
    0x42, 0x4D, 0x36, 0x58, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x42, 0x00, 0x00, 0x00, 0x28, 0x00,
    0x00, 0x00, 0x40, 0x01, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x01, 0x00, 0x10, 0x00, 0x03, 0x00,
    0x00, 0x00, 0x00, 0x58, 0x02, 0x00, 0xC4, 0x0E, 0x00, 0x00, 0xC4, 0x0E, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xE0, 0x07, 0x00, 0x00, 0x1F, 0x00,
    0x00, 0x00
};
// set pin 7 as the slave select for the digital pot:
const int CS = 7;
bool is_header = false;
int mode = 0;
uint8_t start_capture = 0;
ArduCAM myCAM( OV5642, CS );
// === MULTI PHOTO ===
unsigned long lastCapture = 0;

```

```

const int interval = 3000; // délai entre photos (ms)
const int nbPhotos = 20; // nombre de photos
int compteur = 0;
bool multiShot = false;
uint8_t read_fifo_burst(ArduCAM myCAM);
void setup() {
// put your setup code here, to run once:
uint8_t vid, pid;
uint8_t temp;
#ifdef __SAM3X8E__
  Wire1.begin();
  Serial.begin(115200);
#else
  Wire.begin();
  Serial.begin(921600);
#endif
Serial.println(F("ACK CMD ArduCAM Start! END"));
// set the CS as an output:
pinMode(CS, OUTPUT);
digitalWrite(CS, HIGH);
// initialize SPI:
SPI.begin();
//Reset the CPLD
myCAM.write_reg(0x07, 0x80);
delay(100);
myCAM.write_reg(0x07, 0x00);
delay(100);
while(1){
  //Check if the ArduCAM SPI bus is OK
  myCAM.write_reg(ARDUCHIP_TEST1, 0x55);
  temp = myCAM.read_reg(ARDUCHIP_TEST1);
  if (temp != 0x55){
    Serial.println(F("ACK CMD SPI interface Error! END"));
    delay(1000);continue;
  }else{
    Serial.println(F("ACK CMD SPI interface OK. END"));break;
  }
}
}

```

```

while(1){
  //Check if the camera module type is OV5642
  myCAM.wrSensorReg16_8(0xff, 0x01);
  myCAM.rdSensorReg16_8(OV5642_CHIPID_HIGH, &vid);
  myCAM.rdSensorReg16_8(OV5642_CHIPID_LOW, &pid);
  if((vid != 0x56) || (pid != 0x42)){
    Serial.println(F("ACK CMD Can't find OV5642 module! END"));
    delay(1000);continue;
  }
  else{
    Serial.println(F("ACK CMD OV5642 detected. END"));break;
  }
}
//Change to JPEG capture mode and initialize the OV5642 module
myCAM.set_format(JPEG);
myCAM.InitCAM();

myCAM.write_reg(ARDUCHIP_TIM, VSYNC_LEVEL_MASK); //VSYNC is active HIGH
myCAM.OV5642_set_JPEG_size(OV5642_320x240);
delay(1000);
myCAM.clear_fifo_flag();
myCAM.write_reg(ARDUCHIP_FRAMES,0x00);
}
void loop() {
// put your main code here, to run repeatedly:

uint8_t temp = 0xff, temp_last = 0;
bool is_header = false;
if (Serial.available())
{
temp = Serial.read();
switch (temp)
{
case 0:
myCAM.OV5642_set_JPEG_size(OV5642_320x240);delay(1000);
Serial.println(F("ACK CMD switch to OV5642_320x240 END"));
temp = 0xff;
break;
}
}
}

```

```
case 1:
  myCAM.OV5642_set_JPEG_size(OV5642_640x480);delay(1000);
  Serial.println(F("ACK CMD switch to OV5642_640x480 END"));
temp = 0xff;
break;
case 2:
  myCAM.OV5642_set_JPEG_size(OV5642_1024x768);delay(1000);
  Serial.println(F("ACK CMD switch to OV5642_1024x768 END"));
temp = 0xff;
break;
case 3:
temp = 0xff;
  myCAM.OV5642_set_JPEG_size(OV5642_1280x960);delay(1000);
  Serial.println(F("ACK CMD switch to OV5642_1280x960 END"));
break;
case 4:
temp = 0xff;
  myCAM.OV5642_set_JPEG_size(OV5642_1600x1200);delay(1000);
  Serial.println(F("ACK CMD switch to OV5642_1600x1200 END"));
break;
case 5:
temp = 0xff;
  myCAM.OV5642_set_JPEG_size(OV5642_2048x1536);delay(1000);
  Serial.println(F("ACK CMD switch to OV5642_2048x1536 END"));
break;
case 6:
temp = 0xff;
  myCAM.OV5642_set_JPEG_size(OV5642_2592x1944);delay(1000);
  Serial.println(F("ACK CMD switch to OV5642_2592x1944 END"));
break;
case 0x10:
mode = 1;
temp = 0xff;
start_capture = 1;
Serial.println(F("ACK CMD CAM start single shoot. END"));
break;
case 0x11:
temp = 0xff;
```

```
myCAM.set_format(JPEG);
myCAM.InitCAM();
#if !(defined (OV2640_MINI_2MP))
myCAM.set_bit(ARDUCHIP_TIM, VSYNC_LEVEL_MASK);
#endif
break;
case 0x20:
mode = 2;
temp = 0xff;
start_capture = 2;
Serial.println(F("ACK CMD CAM start video streaming. END"));
break;
case 0x30:
mode = 3;
temp = 0xff;
start_capture = 3;
Serial.println(F("ACK CMD CAM start single shoot. END"));
break;
case 0x31:
temp = 0xff;
myCAM.set_format(BMP);
myCAM.InitCAM();
myCAM.clear_bit(ARDUCHIP_TIM, VSYNC_LEVEL_MASK);
myCAM.wrSensorReg16_8(0x3818, 0x81);
myCAM.wrSensorReg16_8(0x3621, 0xA7);
break;
case 0x40:
myCAM.OV5642_set_Light_Mode(Advanced_AWB);temp = 0xff;
Serial.println(F("ACK CMD Set to Advanced_AWB END"));break;
case 0x41:
myCAM.OV5642_set_Light_Mode(Simple_AWB);temp = 0xff;
Serial.println(F("ACK CMD Set to Simple_AWB END"));break;
case 0x42:
myCAM.OV5642_set_Light_Mode(Manual_day);temp = 0xff;
Serial.println(F("ACK CMD Set to Manual_day END"));break;
case 0x43:
myCAM.OV5642_set_Light_Mode(Manual_A);temp = 0xff;
Serial.println(F("ACK CMD Set to Manual_A END"));break;
```

```
case 0x44:
myCAM.OV5642_set_Light_Mode(Manual_cwf);temp = 0xff;
Serial.println(F("ACK CMD Set to Manual_cwf END"));break;
case 0x45:
myCAM.OV5642_set_Light_Mode(Manual_cloudy);temp = 0xff;
Serial.println(F("ACK CMD Set to Manual_cloudy END"));break;
case 0x50:
myCAM.OV5642_set_Color_Saturation(Saturation4);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+4 END"));break;
case 0x51:
myCAM.OV5642_set_Color_Saturation(Saturation3);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+3 END"));break;
case 0x52:
myCAM.OV5642_set_Color_Saturation(Saturation2);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+2 END"));break;
case 0x53:
myCAM.OV5642_set_Color_Saturation(Saturation1);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+1 END"));break;
case 0x54:
myCAM.OV5642_set_Color_Saturation(Saturation0);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+0 END"));break;
case 0x55:
myCAM.OV5642_set_Color_Saturation(Saturation_1);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation-1 END"));break;
case 0x56:
myCAM.OV5642_set_Color_Saturation(Saturation_2);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation-2"));break;
case 0x57:
myCAM.OV5642_set_Color_Saturation(Saturation_3);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation-3 END"));break;
case 0x58:
myCAM.OV5642_set_Light_Mode(Saturation_4);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation-4 END"));break;
case 0x60:
myCAM.OV5642_set_Brightness(Brightness4);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness+4 END"));break;
case 0x61:
myCAM.OV5642_set_Brightness(Brightness3);temp = 0xff;
```

```
Serial.println(F("ACK CMD Set to Brightness+3 END"));break;
case 0x62:
myCAM.OV5642_set_Brightness(Brightness2);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness+2 END"));break;
case 0x63:
myCAM.OV5642_set_Brightness(Brightness1);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness+1 END"));break;
case 0x64:
myCAM.OV5642_set_Brightness(Brightness0);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness+0 END"));break;
case 0x65:
myCAM.OV5642_set_Brightness(Brightness_1);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness-1 END"));break;
case 0x66:
myCAM.OV5642_set_Brightness(Brightness_2);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness-2 END"));break;
case 0x67:
myCAM.OV5642_set_Brightness(Brightness_3);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness-3 END"));break;
case 0x68:
myCAM.OV5642_set_Brightness(Brightness_4);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness-4 END"));break;
case 0x70:
myCAM.OV5642_set_Contrast(Contrast4);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+4 END"));break;
case 0x71:
myCAM.OV5642_set_Contrast(Contrast3);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+3 END"));break;
case 0x72:
myCAM.OV5642_set_Contrast(Contrast2);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+2 END"));break;
case 0x73:
myCAM.OV5642_set_Contrast(Contrast1);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+1 END"));break;
case 0x74:
myCAM.OV5642_set_Contrast(Contrast0);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+0 END"));break;
case 0x75:
```

```
myCAM.OV5642_set_Contrast(Contrast_1);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast-1 END"));break;
case 0x76:
myCAM.OV5642_set_Contrast(Contrast_2);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast-2 END"));break;
case 0x77:
myCAM.OV5642_set_Contrast(Contrast_3);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast-3 END"));break;
case 0x78:
myCAM.OV5642_set_Contrast(Contrast_4);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast-4 END"));break;
case 0x80:
myCAM.OV5642_set_hue(degree_180);temp = 0xff;
Serial.println(F("ACK CMD Set to -180 degree END"));break;
case 0x81:
myCAM.OV5642_set_hue(degree_150);temp = 0xff;
Serial.println(F("ACK CMD Set to -150 degree END"));break;
case 0x82:
myCAM.OV5642_set_hue(degree_120);temp = 0xff;
Serial.println(F("ACK CMD Set to -120 degree END"));break;
case 0x83:
myCAM.OV5642_set_hue(degree_90);temp = 0xff;
Serial.println(F("ACK CMD Set to -90 degree END"));break;
case 0x84:
myCAM.OV5642_set_hue(degree_60);temp = 0xff;
Serial.println(F("ACK CMD Set to -60 degree END"));break;
case 0x85:
myCAM.OV5642_set_hue(degree_30);temp = 0xff;
Serial.println(F("ACK CMD Set to -30 degree END"));break;
case 0x86:
myCAM.OV5642_set_hue(degree_0);temp = 0xff;
Serial.println(F("ACK CMD Set to 0 degree END"));break;
case 0x87:
myCAM.OV5642_set_hue(degree30);temp = 0xff;
Serial.println(F("ACK CMD Set to 30 degree END"));break;
case 0x88:
myCAM.OV5642_set_hue(degree60);temp = 0xff;
Serial.println(F("ACK CMD Set to 60 degree END"));break;
```

```
case 0x89:
myCAM.OV5642_set_hue(degree90);temp = 0xff;
  Serial.println(F("ACK CMD Set to 90 degree END"));break;
case 0x8a:
myCAM.OV5642_set_hue(degree120);temp = 0xff;
  Serial.println(F("ACK CMD Set to 120 degree END"));break ;
case 0x8b:
myCAM.OV5642_set_hue(degree150);temp = 0xff;
  Serial.println(F("ACK CMD Set to 150 degree END"));break ;
case 0x90:
myCAM.OV5642_set_Special_effects(Normal);temp = 0xff;
  Serial.println(F("ACK CMD Set to Normal END"));break ;
  case 0x91:
myCAM.OV5642_set_Special_effects(BW);temp = 0xff;
  Serial.println(F("ACK CMD Set to BW END"));break ;
case 0x92:
myCAM.OV5642_set_Special_effects(Bluish);temp = 0xff;
  Serial.println(F("ACK CMD Set to Bluish END"));break ;
  case 0x93:
myCAM.OV5642_set_Special_effects(Sepia);temp = 0xff;
  Serial.println(F("ACK CMD Set to Sepia END"));break ;
case 0x94:
myCAM.OV5642_set_Special_effects(Reddish);temp = 0xff;
  Serial.println(F("ACK CMD Set to Reddish END"));break ;
case 0x95:
myCAM.OV5642_set_Special_effects(Greenish);temp = 0xff;
  Serial.println(F("ACK CMD Set to Greenish END"));break ;
case 0x96:
myCAM.OV5642_set_Special_effects(Negative);temp = 0xff;
  Serial.println(F("ACK CMD Set to Negative END"));break ;
case 0xA0:
myCAM.OV5642_set_Exposure_level(Exposure_17_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to -1.7EV"));break ;
  case 0xA1:
myCAM.OV5642_set_Exposure_level(Exposure_13_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to -1.3EV END"));break ;
  case 0xA2:
myCAM.OV5642_set_Exposure_level(Exposure_10_EV);temp = 0xff;
```

```
Serial.println(F("ACK CMD Set to -1.0EV END"));break ;
case 0xA3:
myCAM.OV5642_set_Exposure_level(Exposure_07_EV);temp = 0xff;
Serial.println(F("ACK CMD Set to -0.7EV END"));break ;
case 0xA4:
myCAM.OV5642_set_Exposure_level(Exposure_03_EV);temp = 0xff;
Serial.println(F("ACK CMD Set to -0.3EV END"));break ;
case 0xA5:
myCAM.OV5642_set_Exposure_level(Exposure_default);temp = 0xff;
Serial.println(F("ACK CMD Set to -Exposure_default END"));break ;
case 0xA6:
myCAM.OV5642_set_Exposure_level(Exposure07_EV);temp = 0xff;
Serial.println(F("ACK CMD Set to 0.7EV END"));break ;
case 0xA7:
myCAM.OV5642_set_Exposure_level(Exposure10_EV);temp = 0xff;
Serial.println(F("ACK CMD Set to 1.0EV END"));break ;
case 0xA8:
myCAM.OV5642_set_Exposure_level(Exposure13_EV);temp = 0xff;
Serial.println(F("ACK CMD Set to 1.3EV END"));break ;
case 0xA9:
myCAM.OV5642_set_Exposure_level(Exposure17_EV);temp = 0xff;
Serial.println(F("ACK CMD Set to 1.7EV END"));break ;
case 0xB0:
myCAM.OV5642_set_Sharpness(Auto_Sharpness_default);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Sharpness default END"));break ;
case 0xB1:
myCAM.OV5642_set_Sharpness(Auto_Sharpness1);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Sharpness +1 END"));break ;
case 0xB2:
myCAM.OV5642_set_Sharpness(Auto_Sharpness2);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Sharpness +2 END"));break ;
case 0xB3:
myCAM.OV5642_set_Sharpness(Manual_Sharpnessoff);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness off END"));break ;
case 0xB4:
myCAM.OV5642_set_Sharpness(Manual_Sharpness1);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +1 END"));break ;
case 0xB5:
```

```
myCAM.OV5642_set_Sharpness(Manual_Sharpness2);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +2 END"));break ;
case 0xB6:
myCAM.OV5642_set_Sharpness(Manual_Sharpness3);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +3 END"));break ;
case 0xB7:
myCAM.OV5642_set_Sharpness(Manual_Sharpness4);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +4 END"));break ;
case 0xB8:
myCAM.OV5642_set_Sharpness(Manual_Sharpness5);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +5 END"));break ;
case 0xC0:
myCAM.OV5642_set_Mirror_Flip(MIRROR);temp = 0xff;
Serial.println(F("ACK CMD Set to MIRROR END"));break ;
case 0xC1:
myCAM.OV5642_set_Mirror_Flip(FLIP);temp = 0xff;
Serial.println(F("ACK CMD Set to FLIP END"));break ;
case 0xC2:
myCAM.OV5642_set_Mirror_Flip(MIRROR_FLIP);temp = 0xff;
Serial.println(F("ACK CMD Set to MIRROR&FLIP END"));break ;
case 0xC3:
myCAM.OV5642_set_Mirror_Flip(Normal);temp = 0xff;
Serial.println(F("ACK CMD Set to Normal END"));break ;
case 0xD0:
myCAM.OV5642_set_Compress_quality(high_quality);temp = 0xff;
Serial.println(F("ACK CMD Set to high quality END"));break ;
case 0xD1:
myCAM.OV5642_set_Compress_quality(default_quality);temp = 0xff;
Serial.println(F("ACK CMD Set to default quality END"));break ;
case 0xD2:
myCAM.OV5642_set_Compress_quality(low_quality);temp = 0xff;
Serial.println(F("ACK CMD Set to low quality END"));break ;

case 0xE0:
myCAM.OV5642_Test_Pattern(Color_bar);temp = 0xff;
Serial.println(F("ACK CMD Set to Color bar END"));break ;
case 0xE1:
myCAM.OV5642_Test_Pattern(Color_square);temp = 0xff;
```

```

Serial.println(F("ACK CMD Set to Color square END"));break ;
case 0xE2:
myCAM.OV5642_Test_Pattern(BW_square);temp = 0xff;
Serial.println(F("ACK CMD Set to BW square END"));break ;
case 0xE3:
myCAM.OV5642_Test_Pattern(DLI);temp = 0xff;
Serial.println(F("ACK CMD Set to DLI END"));break ;
default:
break;
}
}
if (mode == 1)
{
// Lancement du mode multi-photo
if (start_capture == 1)
{
compteur = 0;
multiShot = true;
lastCapture = millis();
start_capture = 0;
}

// Boucle de capture avec délai
if (multiShot)
{
if (millis() - lastCapture >= interval)
{
lastCapture = millis();

// Capture
myCAM.flush_fifo();
myCAM.clear_fifo_flag();
myCAM.start_capture();

// Attendre fin capture
while (!myCAM.get_bit(ARDUCHIP_TRIG, CAP_DONE_MASK));

Serial.println(F("ACK CMD CAM Capture Done. END"));
}
}
}
}

```

```

// Envoi vers Host App
read_fifo_burst(myCAM);

myCAM.clear_fifo_flag();

compteur++;

if (compteur >= nbPhotos)
{
    multiShot = false;
    Serial.println(F("ACK CMD Multi capture finished. END"));
}
}
}
else if (mode == 2)
{
    while (1)
    {
        temp = Serial.read();
        if (temp == 0x21)
        {
            start_capture = 0;
            mode = 0;
            Serial.println(F("ACK CMD CAM stop video streaming. END"));
            break;
        }
        switch(temp){
            case 0x40:
                myCAM.OV5642_set_Light_Mode(Advanced_AWB);temp = 0xff;
                Serial.println(F("ACK CMD Set to Advanced_AWB END"));break;
            case 0x41:
                myCAM.OV5642_set_Light_Mode(Simple_AWB);temp = 0xff;
                Serial.println(F("ACK CMD Set to Simple_AWB END"));break;
            case 0x42:
                myCAM.OV5642_set_Light_Mode(Manual_day);temp = 0xff;
                Serial.println(F("ACK CMD Set to Manual_day END"));break;

```

```
case 0x43:
myCAM.OV5642_set_Light_Mode(Manual_A);temp = 0xff;
Serial.println(F("ACK CMD Set to Manual_A END"));break;
case 0x44:
myCAM.OV5642_set_Light_Mode(Manual_cwf);temp = 0xff;
Serial.println(F("ACK CMD Set to Manual_cwf END"));break;
case 0x45:
myCAM.OV5642_set_Light_Mode(Manual_cloudy);temp = 0xff;
Serial.println(F("ACK CMD Set to Manual_cloudy END"));break;
case 0x50:
myCAM.OV5642_set_Color_Saturation(Saturation4);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+4 END"));break;
case 0x51:
myCAM.OV5642_set_Color_Saturation(Saturation3);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+3 END"));break;
case 0x52:
myCAM.OV5642_set_Color_Saturation(Saturation2);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+2 END"));break;
case 0x53:
myCAM.OV5642_set_Color_Saturation(Saturation1);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+1 END"));break;
case 0x54:
myCAM.OV5642_set_Color_Saturation(Saturation0);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation+0 END"));break;
case 0x55:
myCAM.OV5642_set_Color_Saturation(Saturation_1);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation-1 END"));break;
case 0x56:
myCAM.OV5642_set_Color_Saturation(Saturation_2);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation-2 END"));break;
case 0x57:
myCAM.OV5642_set_Color_Saturation(Saturation_3);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation-3 END"));break;
case 0x58:
myCAM.OV5642_set_Light_Mode(Saturation_4);temp = 0xff;
Serial.println(F("ACK CMD Set to Saturation-4 END"));break;
case 0x60:
myCAM.OV5642_set_Brightness(Brightness4);temp = 0xff;
```

```
Serial.println(F("ACK CMD Set to Brightness+4 END"));break;
case 0x61:
myCAM.OV5642_set_Brightness(Brightness3);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness+3 END"));break;
case 0x62:
myCAM.OV5642_set_Brightness(Brightness2);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness+2 END"));break;
case 0x63:
myCAM.OV5642_set_Brightness(Brightness1);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness+1 END"));break;
case 0x64:
myCAM.OV5642_set_Brightness(Brightness0);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness+0 END"));break;
case 0x65:
myCAM.OV5642_set_Brightness(Brightness_1);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness-1 END"));break;
case 0x66:
myCAM.OV5642_set_Brightness(Brightness_2);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness-2 END"));break;
case 0x67:
myCAM.OV5642_set_Brightness(Brightness_3);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness-3 END"));break;
case 0x68:
myCAM.OV5642_set_Brightness(Brightness_4);temp = 0xff;
Serial.println(F("ACK CMD Set to Brightness-4 END"));break;
case 0x70:
myCAM.OV5642_set_Contrast(Contrast4);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+4 END"));break;
case 0x71:
myCAM.OV5642_set_Contrast(Contrast3);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+3 END"));break;
case 0x72:
myCAM.OV5642_set_Contrast(Contrast2);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+2 END"));break;
case 0x73:
myCAM.OV5642_set_Contrast(Contrast1);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+1 END"));break;
case 0x74:
```

```
myCAM.OV5642_set_Contrast(Contrast0);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast+0 END"));break;
case 0x75:
myCAM.OV5642_set_Contrast(Contrast_1);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast-1 END"));break;
case 0x76:
myCAM.OV5642_set_Contrast(Contrast_2);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast-2 END"));break;
case 0x77:
myCAM.OV5642_set_Contrast(Contrast_3);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast-3 END"));break;
case 0x78:
myCAM.OV5642_set_Contrast(Contrast_4);temp = 0xff;
Serial.println(F("ACK CMD Set to Contrast-4 END"));break;
case 0x80:
myCAM.OV5642_set_hue(degree_180);temp = 0xff;
Serial.println(F("ACK CMD Set to -180 degree END"));break;
case 0x81:
myCAM.OV5642_set_hue(degree_150);temp = 0xff;
Serial.println(F("ACK CMD Set to -150 degree END"));break;
case 0x82:
myCAM.OV5642_set_hue(degree_120);temp = 0xff;
Serial.println(F("ACK CMD Set to -120 degree END"));break;
case 0x83:
myCAM.OV5642_set_hue(degree_90);temp = 0xff;
Serial.println(F("ACK CMD Set to -90 degree END"));break;
case 0x84:
myCAM.OV5642_set_hue(degree_60);temp = 0xff;
Serial.println(F("ACK CMD Set to -60 degree END"));break;
case 0x85:
myCAM.OV5642_set_hue(degree_30);temp = 0xff;
Serial.println(F("ACK CMD Set to -30 degree END"));break;
case 0x86:
myCAM.OV5642_set_hue(degree_0);temp = 0xff;
Serial.println(F("ACK CMD Set to 0 degree END"));break;
case 0x87:
myCAM.OV5642_set_hue(degree30);temp = 0xff;
Serial.println(F("ACK CMD Set to 30 degree END"));break;
```

```
case 0x88:
myCAM.OV5642_set_hue(degree60);temp = 0xff;
  Serial.println(F("ACK CMD Set to 60 degree END"));break;
case 0x89:
myCAM.OV5642_set_hue(degree90);temp = 0xff;
  Serial.println(F("ACK CMD Set to 90 degree END"));break;
  case 0x8a:
myCAM.OV5642_set_hue(degree120);temp = 0xff;
  Serial.println(F("ACK CMD Set to 120 degree END"));break ;
case 0x8b:
myCAM.OV5642_set_hue(degree150);temp = 0xff;
  Serial.println(F("ACK CMD Set to 150 degree END"));break ;
case 0x90:
myCAM.OV5642_set_Special_effects(Normal);temp = 0xff;
  Serial.println(F("ACK CMD Set to Normal END"));break ;
  case 0x91:
myCAM.OV5642_set_Special_effects(BW);temp = 0xff;
  Serial.println(F("ACK CMD Set to BW END"));break ;
case 0x92:
myCAM.OV5642_set_Special_effects(Bluish);temp = 0xff;
  Serial.println(F("ACK CMD Set to Bluish END"));break ;
  case 0x93:
myCAM.OV5642_set_Special_effects(Sepia);temp = 0xff;
  Serial.println(F("ACK CMD Set to Sepia END"));break ;
case 0x94:
myCAM.OV5642_set_Special_effects(Reddish);temp = 0xff;
  Serial.println(F("ACK CMD Set to Reddish END"));break ;
case 0x95:
myCAM.OV5642_set_Special_effects(Greenish);temp = 0xff;
  Serial.println(F("ACK CMD Set to Greenish END"));break ;
case 0x96:
myCAM.OV5642_set_Special_effects(Negative);temp = 0xff;
  Serial.println(F("ACK CMD Set to Negative END"));break ;
case 0xA0:
myCAM.OV5642_set_Exposure_level(Exposure_17_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to -1.7EV END"));break ;
  case 0xA1:
myCAM.OV5642_set_Exposure_level(Exposure_13_EV);temp = 0xff;
```

```
Serial.println(F("ACK CMD Set to -1.3EV END"));break ;
  case 0xA2:
myCAM.OV5642_set_Exposure_level(Exposure_10_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to -1.0EV END"));break ;
  case 0xA3:
myCAM.OV5642_set_Exposure_level(Exposure_07_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to -0.7EV END"));break ;
  case 0xA4:
myCAM.OV5642_set_Exposure_level(Exposure_03_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to -0.3EV END"));break ;
  case 0xA5:
myCAM.OV5642_set_Exposure_level(Exposure_default);temp = 0xff;
  Serial.println(F("ACK CMD Set to -Exposure_default END"));break ;
  case 0xA6:
myCAM.OV5642_set_Exposure_level(Exposure07_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to 0.7EV END"));break ;
  case 0xA7:
myCAM.OV5642_set_Exposure_level(Exposure10_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to 1.0EV END"));break ;
  case 0xA8:
myCAM.OV5642_set_Exposure_level(Exposure13_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to 1.3EV END"));break ;
  case 0xA9:
myCAM.OV5642_set_Exposure_level(Exposure17_EV);temp = 0xff;
  Serial.println(F("ACK CMD Set to 1.7EV END"));break ;
  case 0xB0:
myCAM.OV5642_set_Sharpness(Auto_Sharpness_default);temp = 0xff;
  Serial.println(F("ACK CMD Set to Auto Sharpness default END"));break ;
  case 0xB1:
myCAM.OV5642_set_Sharpness(Auto_Sharpness1);temp = 0xff;
  Serial.println(F("ACK CMD Set to Auto Sharpness +1 END"));break ;
  case 0xB2:
myCAM.OV5642_set_Sharpness(Auto_Sharpness2);temp = 0xff;
  Serial.println(F("ACK CMD Set to Auto Sharpness +2 END"));break ;
  case 0xB3:
myCAM.OV5642_set_Sharpness(Manual_Sharpnessoff);temp = 0xff;
  Serial.println(F("ACK CMD Set to Auto Manual Sharpness off END"));break ;
  case 0xB4:
```

```
myCAM.OV5642_set_Sharpness(Manual_Sharpness1);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +1 END"));break ;
case 0xB5:
myCAM.OV5642_set_Sharpness(Manual_Sharpness2);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +2 END"));break ;
case 0xB6:
myCAM.OV5642_set_Sharpness(Manual_Sharpness3);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +3 END"));break ;
case 0xB7:
myCAM.OV5642_set_Sharpness(Manual_Sharpness4);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +4 END"));break ;
case 0xB8:
myCAM.OV5642_set_Sharpness(Manual_Sharpness5);temp = 0xff;
Serial.println(F("ACK CMD Set to Auto Manual Sharpness +5 END"));break ;
case 0xC0:
myCAM.OV5642_set_Mirror_Flip(MIRROR);temp = 0xff;
Serial.println(F("ACK CMD Set to MIRROR END"));break ;
case 0xC1:
myCAM.OV5642_set_Mirror_Flip(FLIP);temp = 0xff;
Serial.println(F("ACK CMD Set to FLIP END"));break ;
case 0xC2:
myCAM.OV5642_set_Mirror_Flip(MIRROR_FLIP);temp = 0xff;
Serial.println(F("ACK CMD Set to MIRROR&FLIP END"));break ;
case 0xC3:
myCAM.OV5642_set_Mirror_Flip(Normal);temp = 0xff;
Serial.println(F("ACK CMD Set to Normal END"));break ;
case 0xD0:
myCAM.OV5642_set_Compress_quality(high_quality);temp = 0xff;
Serial.println(F("ACK CMD Set to high quality END"));break ;
case 0xD1:
myCAM.OV5642_set_Compress_quality(default_quality);temp = 0xff;
Serial.println(F("ACK CMD Set to default quality END"));break ;
case 0xD2:
myCAM.OV5642_set_Compress_quality(low_quality);temp = 0xff;
Serial.println(F("ACK CMD Set to low quality END"));break ;

case 0xE0:
myCAM.OV5642_Test_Pattern(Color_bar);temp = 0xff;
```

```

Serial.println(F("ACK CMD Set to Color bar END"));break ;
case 0xE1:
myCAM.OV5642_Test_Pattern(Color_square);temp = 0xff;
Serial.println(F("ACK CMD Set to Color square END"));break ;
case 0xE2:
myCAM.OV5642_Test_Pattern(BW_square);temp = 0xff;
Serial.println(F("ACK CMD Set to BW square END"));break ;
case 0xE3:
myCAM.OV5642_Test_Pattern(DLI);temp = 0xff;
Serial.println(F("ACK CMD Set to DLI END"));break ;

}
if (start_capture == 2)
{
myCAM.flush_fifo();
myCAM.clear_fifo_flag();
//Start capture
myCAM.start_capture();
start_capture = 0;
}
if (myCAM.get_bit(ARDUCHIP_TRIG, CAP_DONE_MASK))
{
uint32_t length = 0;
length = myCAM.read_fifo_length();
if ((length >= MAX_FIFO_SIZE) | (length == 0))
{
myCAM.clear_fifo_flag();
start_capture = 2;
continue;
}
myCAM.CS_LOW();
myCAM.set_fifo_burst();//Set fifo burst mode
temp = SPI.transfer(0x00);
length --;
while ( length-- )
{
temp_last = temp;
temp = SPI.transfer(0x00);
}
}

```



```

length = myCAM.read_fifo_length();
if (length >= MAX_FIFO_SIZE )
{
  Serial.println(F("ACK CMD Over size. END"));
  myCAM.clear_fifo_flag();
  return;
}
if (length == 0 ) //0 kb
{
  Serial.println(F("ACK CMD Size is 0. END"));
  myCAM.clear_fifo_flag();
  return;
}
myCAM.CS_LOW();
myCAM.set_fifo_burst();//Set fifo burst mode

Serial.write(0xFF);
Serial.write(0xAA);
for (temp = 0; temp < BMPIMAGEOFFSET; temp++)
{
  Serial.write(pgm_read_byte(&bmp_header[temp]));
}
//SPI.transfer(0x00);
char VH, VL;
int i = 0, j = 0;
for (i = 0; i < 240; i++)
{
  for (j = 0; j < 320; j++)
  {
    VH = SPI.transfer(0x00);;
    VL = SPI.transfer(0x00);;
    Serial.write(VL);
    delayMicroseconds(12);
    Serial.write(VH);
    delayMicroseconds(12);
  }
}
Serial.write(0xBB);

```

```

Serial.write(0xCC);
myCAM.CS_HIGH();
//Clear the capture done flag
myCAM.clear_fifo_flag();
}
}
}
uint8_t read_fifo_burst(ArduCAM myCAM)
{
uint8_t temp = 0, temp_last = 0;
uint32_t length = 0;
length = myCAM.read_fifo_length();
Serial.println(length, DEC);
if (length >= MAX_FIFO_SIZE) //512 kb
{
Serial.println(F("ACK CMD Over size. END"));
return 0;
}
if (length == 0 ) //0 kb
{
Serial.println(F("ACK CMD Size is 0. END"));
return 0;
}
myCAM.CS_LOW();
myCAM.set_fifo_burst();//Set fifo burst mode
temp = SPI.transfer(0x00);
length --;
while ( length-- )
{
temp_last = temp;
temp = SPI.transfer(0x00);
if (is_header == true)
{
Serial.write(temp);
}
else if ((temp == 0xD8) & (temp_last == 0xFF))
{
is_header = true;
}
}
}
}
}

```

```
Serial.println(F("ACK IMG END"));
Serial.write(temp_last);
Serial.write(temp);
}
if ( (temp == 0xD9) && (temp_last == 0xFF) ) //If find the end ,break while,
break;
delayMicroseconds(15);
}
myCAM.CS_HIGH();
is_header = false;
return 1;
}
```

LES SOUCIS :

- On ne sait pas si c'est toujours possible d'avoir deux caméras pour le bras.
- Il va falloir combiner le code du moteur avec le code de la caméra en réfléchissant au problème suivant : une ou deux Arduino ? Comment faire avec le Shield motor ? Est ce que la caméra acceptera l'alimentation en 12 V nécessaire pour le moteur ? (Parce que pas de possibilité de changer de Shield motor --> seul compatible avec le 12 V pour le moteur NEMA) Arduino maitre/esclave ?
- Il faut revoir le mouvement du moteur sur le Coulommiers terminé pour être sur de ses différents déplacement et degré de rotation : seulement une partie pour le moment.
- Faire premier test de montage des différentes pièces de notre Tournicayou

Pour la prochaine séance --> Test d'ajouts de pin sur le Shield moteur pour brancher donc le motor NEMA et la caméra dessus avec les pin analytiques + Imprimer un Coulommiers entier pour pouvoir réaliser les tests moteurs sur tout le tour du Coulommiers et par la suite combiner code moteur et code caméra !

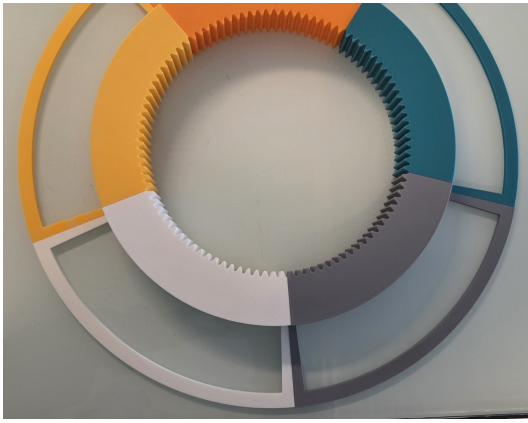
Notre joli coulommiers est arrivé, il est tout beau et tout coloré ! (Photo ci-joint)

Nos petits problèmes de la séance : 1er vrai test montage de notre "Tounicayou"

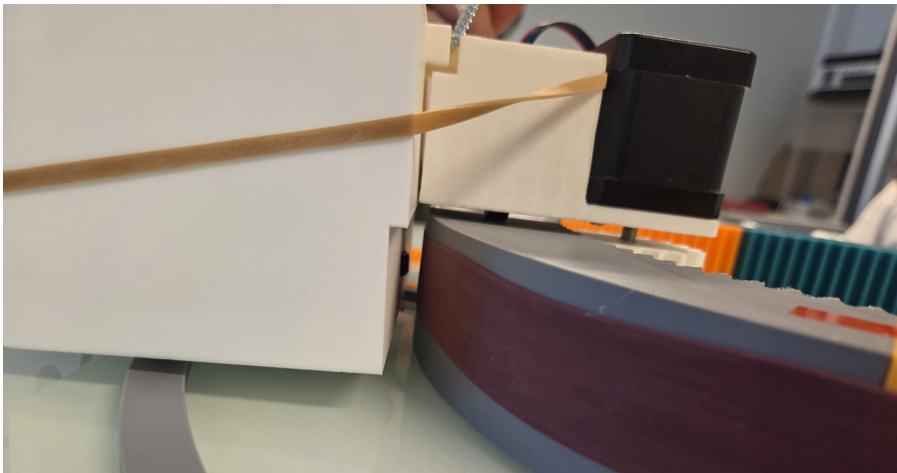
Nécessité de re modélisation de certaines pièces

Aujourd'hui nous avons essayé de monter les différentes pièces toutes ensembles. Plusieurs problèmes se sont alors présentés à nous:

- Les puzzles (Coulommiers) ont des marges d'erreurs sur estimées. Il faudrait à l'avenir les repenser afin de faciliter le montage. Toutes les complémentarités sont à revoir, nonobstant, celle entre le socle moteur et la pièce arrière est primordiale.
- La pièce arrière avait été pensée pour avoir une roue permettant de faire encoche avec le coulommiers, permettant alors de maintenir le tout sans branlements. Il faudrait donc, soit retirer la buté, soit redesigner la pièce dans son ensemble.
- Le support moteur est à retravailler. Imaginé au départ comme nécessitant une pièce complémentaire pour être stable, il devra dorénavant obtenir une stabilité autonome. L'idée sera d'ajouter des butes sur le bord distal au foyer du coulommiers. Plusieurs modèle de butes seront testés.
- Le coulommiers en lui même ne sera pas modifié, mais des explications supplémentaires quant à sa conception peuvent être ajoutées. Le principe du coulommiers est de permettre une rotation d'un bras photographique autour du support -substituant la rotation du support lui même-. Les dents sont sur la surface interne du coulommiers **Pourquoi ?** --> Pour permettre un ancrage par le rouage et une autre encoche à la surface opposée. L'idée de rouage sur sur le coté haut du coulommiers a été proposée. Elle ne nous semble que peu adéquate au projet. En effet, elle poserait plus de problèmes, nécessitant deux encoche, et un rouage circulaire à la Citroën. IMPOSSIBLE.



Le coulommiers tout coloré et Lucas



02/05/2026_Modification de la pièce moteur

Pour le moteur baroque, 3 idées sont pour l'instant modélisées. Un plaque (à poncer), deux encoches, un méga anneau. En plus de cela, le support a également été repensé avec l'ajout

d'une encoche permettant l'insertion du bras photo sur celui-ci. Cette structure n'est là que pour l'idée d'améliorer le projet dans le futur, pour l'instant le bras si insérant n'a pas été conçu.

A noter donc, cette encoche fait :

- 20 mm de large
- 29 mm de long
- 10 mm de haut
- L'encoche fait 25 mm de long
- L'encoche fait 8 mm de diamètre, son foyer est centré par rapport aux côtés droit et gauche. Cependant le foyer se trouve à 3.25 mm du haut. L'encoche est circulaire, du fait que ce soit plus simple à designer

[Barroque motor.FCStd](#) - Plaque

[Barroque motor.stl](#) - Plaque

[Barroque motor - 2 coins.FCStd](#)

[Barroque motor - 2 coins.stl](#)

[Barroque motor - anneau.FCStd](#)

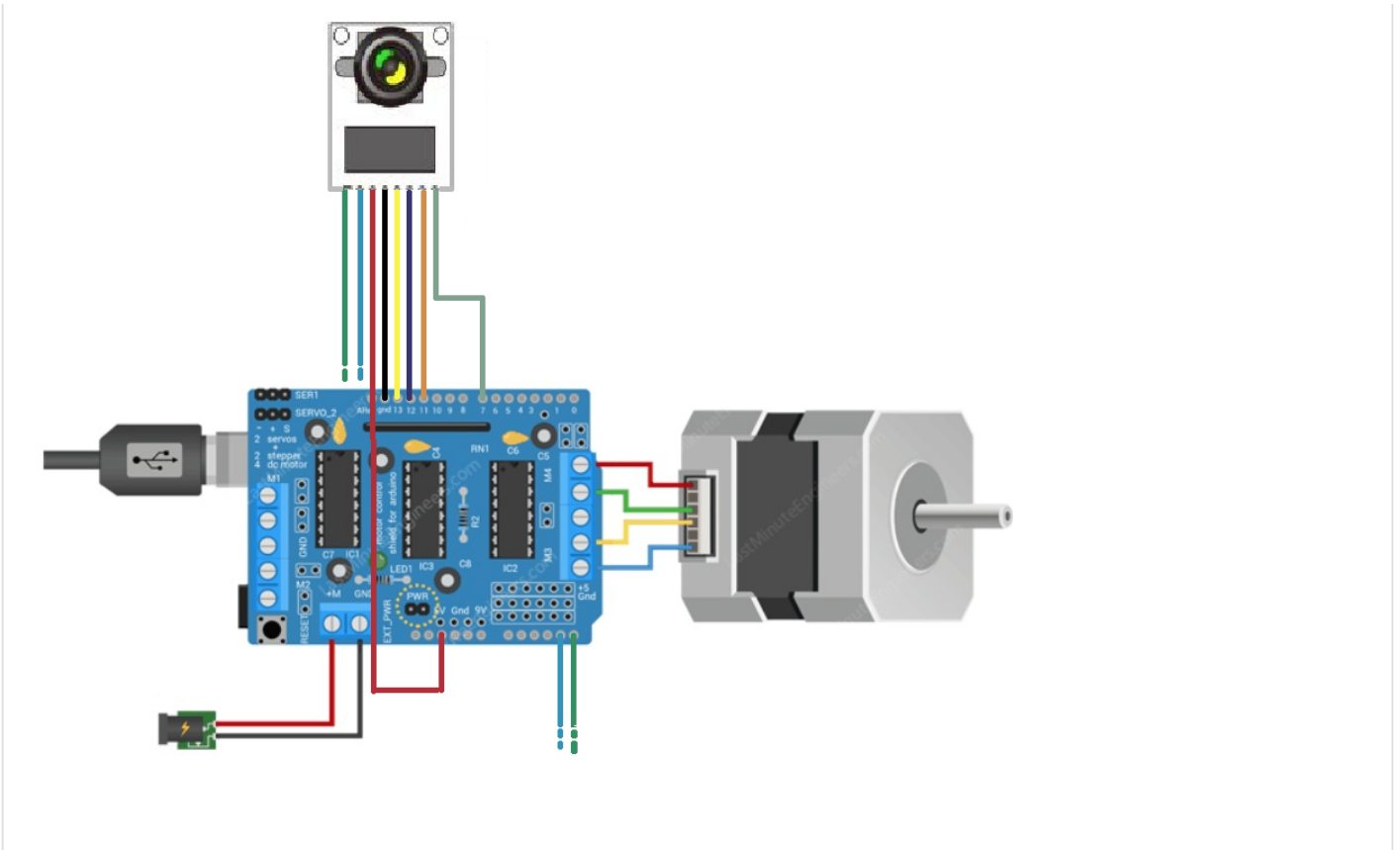
[Barroque motor - anneau.stl](#)

22/05/2026

Objectif sur le montage du **Tournicayou** pour les tests moteur

Modification de la pièce support moteur pendant environ 3h, amélioration du wiki en parallèle. Dernière heure modification de la pièce support du bras pour faire nos tests, si on y arrive.

Réalisation du montage final pour la partie programmation !



Les liens utiles :

- https://github.com/ArduCAM/Arducam_mini
- <https://github.com/ArduCAM/Arduino>
- <https://www.youtube.com/watch?v=Flfrkj08pnQ>
- <https://github.com/ArduCAM/Arduino/issues/528>

Revision #73

Created 30 January 2026 16:00:27 by Albenga Zoe

Updated 25 May 2026 11:27:27 by Albenga Zoe