

TIPE 2025



DÉRAILLEUR INTELLIGENT

RÉALISÉ PAR : Yahya Raoudi

CODE CNC : MH006T

TIPE 2025



PLAN DE PRÉSENTATION

Introduction et fonctionnement

problématique

Cahier de charges (SYSML)

Objectifs:

1 : Électrification du dérailleur

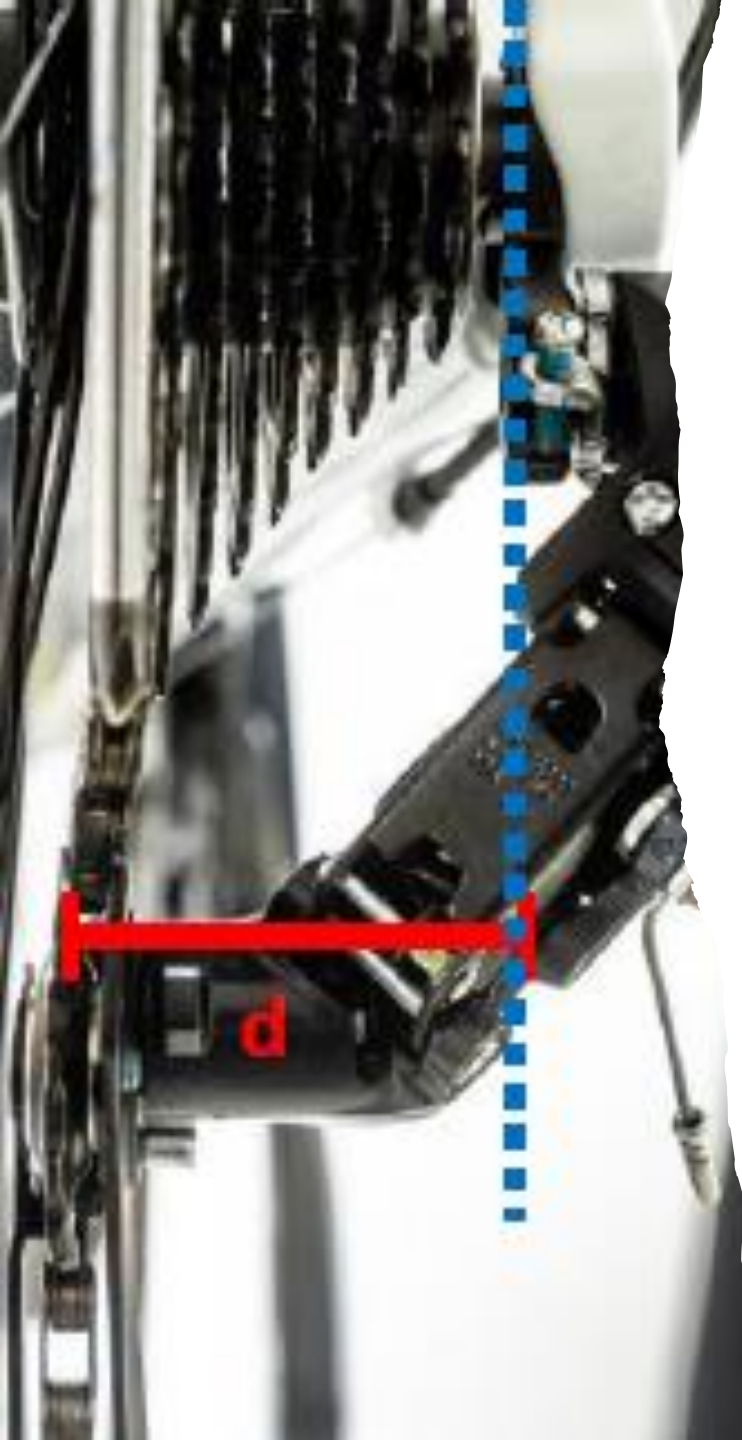
2 : Automatisation du changement de rapport

ANNEXE

FONCTIONNEMENT

- Lorsque l'on tire sur le câble, il déplace le dérailleur du plus grand rapport vers le plus petit (par exemple du 5^e au 1^{er}).

- Lorsque l'on relâche le câble, c'est un ressort intégré dans le mécanisme qui ramène le dérailleur du petit rapport vers le grand.



Problématique

Le dérailleur classique est un défi pour les cyclistes qui nécessite une rapide réflexion pour le bien commander alors comment l'adapter automatiquement pour optimiser le passage des vitesses ?



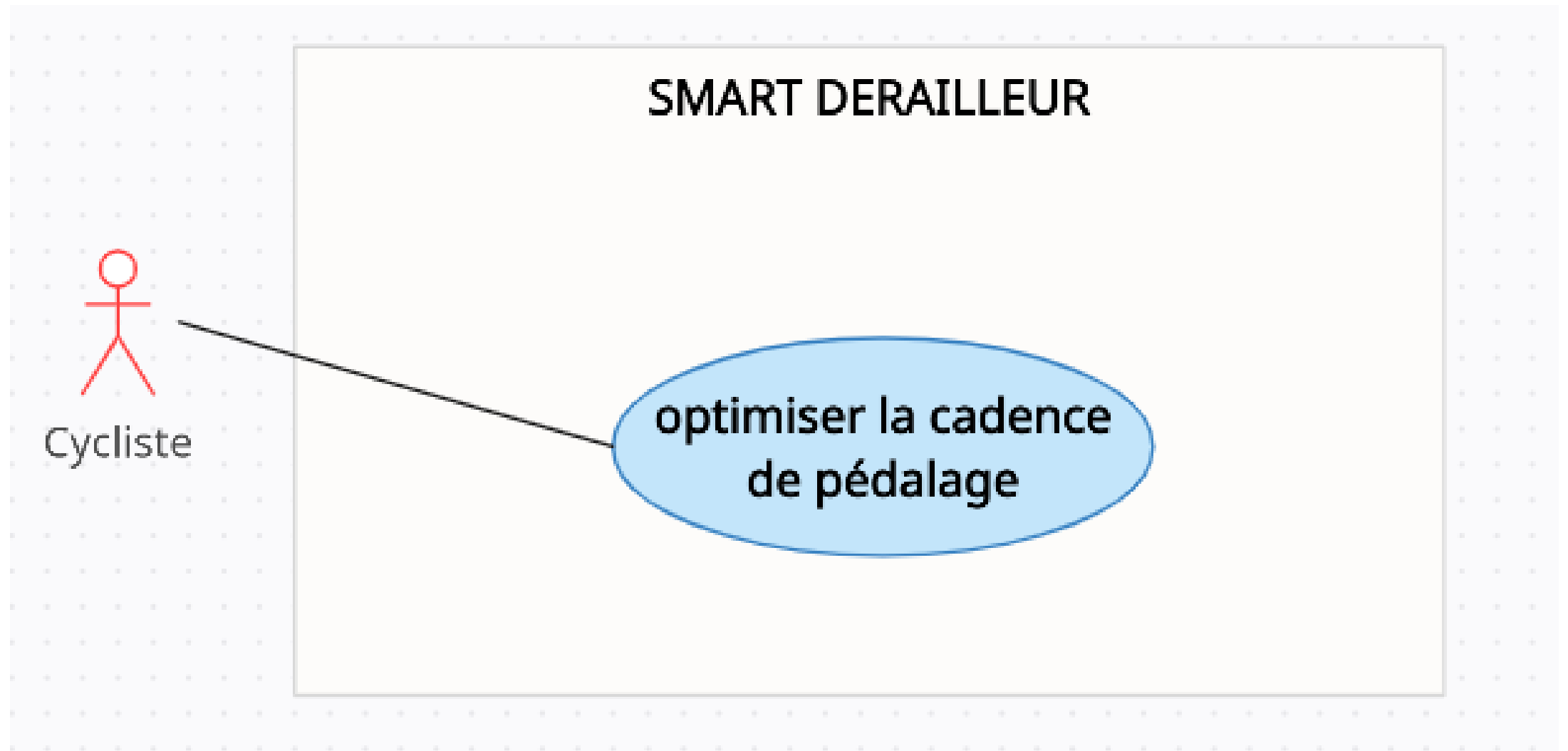


Figure 1 : Diagramme de Cas d'utilisation

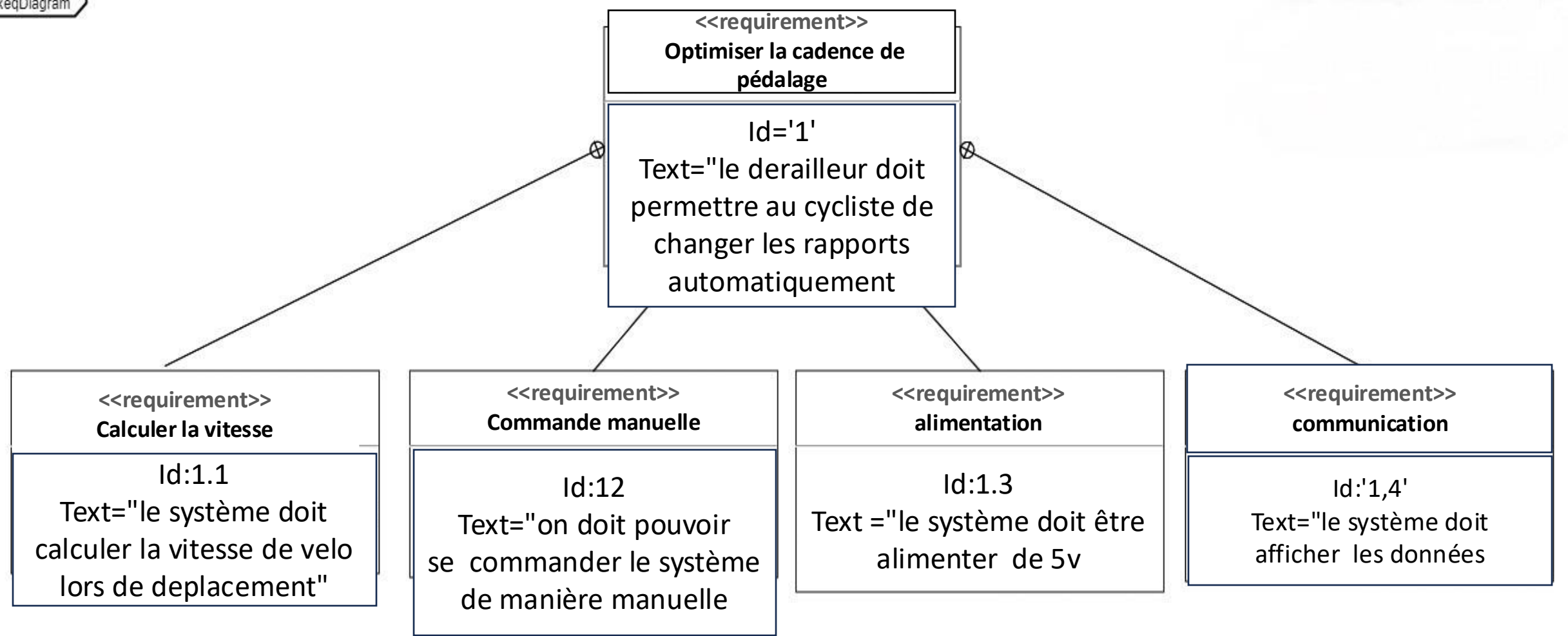
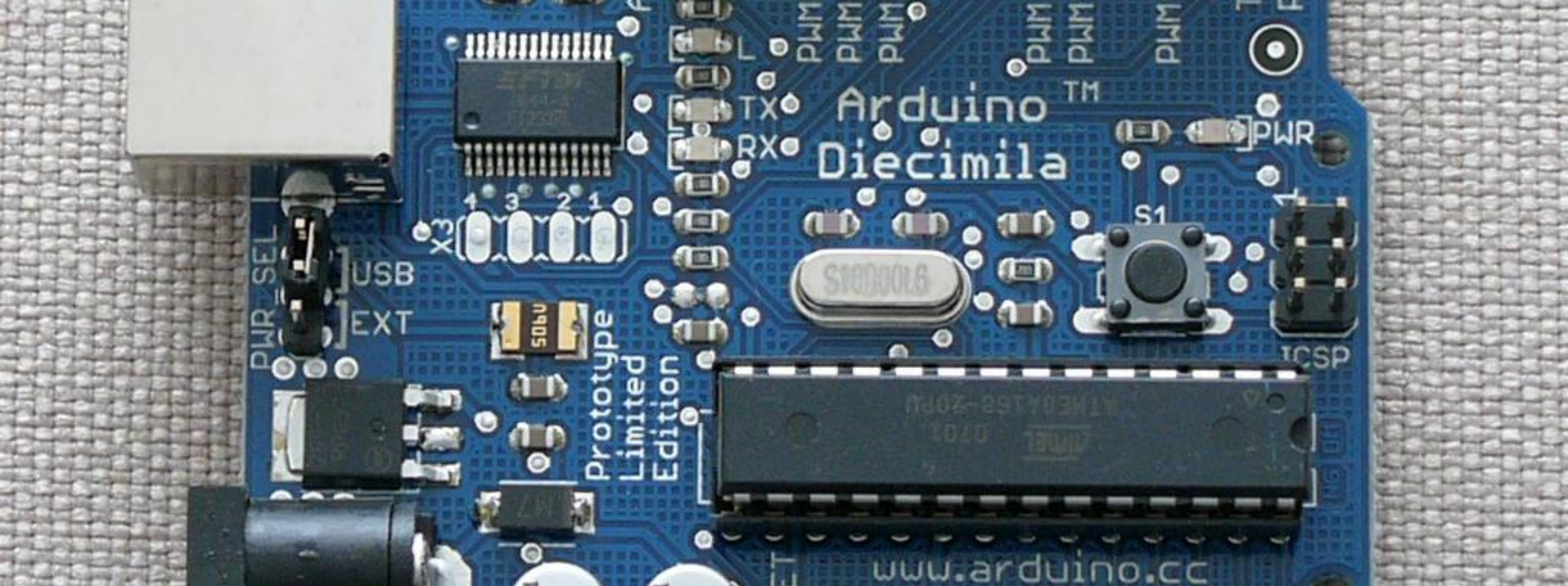


Figure 2 : Diagramme des Exigences



1^{er} Objectif : • **Électrification du dérailleur**

CHOIX DU DERAILLEUR

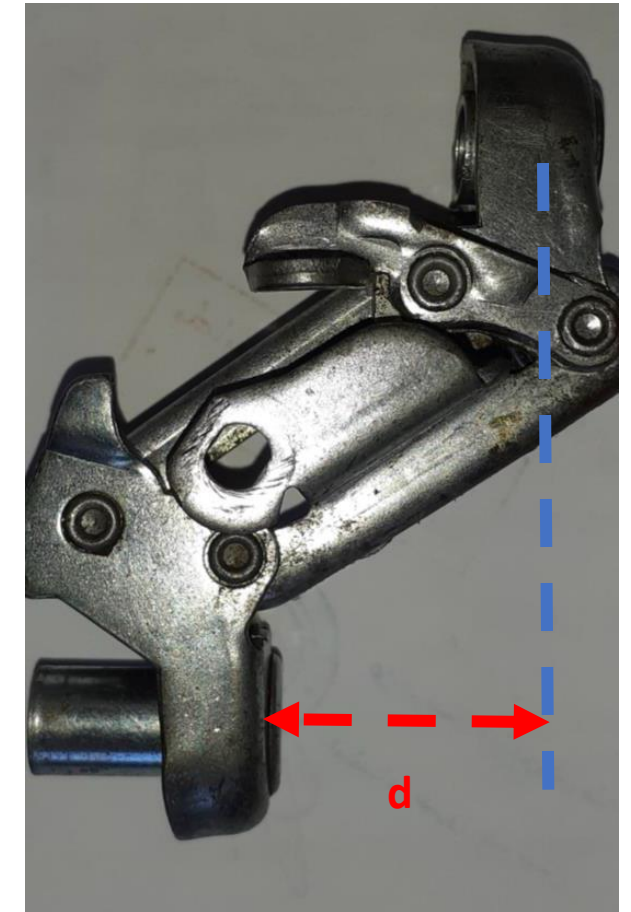
Figure 3 : Shimano Tourney (TY15 GS)

Au cours de cette étude on va travailler sur le dérailleur Shimano Tourney (TY15 GS) qui est compatible avec une roue libre de 5 vitesses. Et justement sur la partie mobile qui fait varier le dérailleur entre les différents rapports pour cela on va enlever le ressort et le câble de transmission.

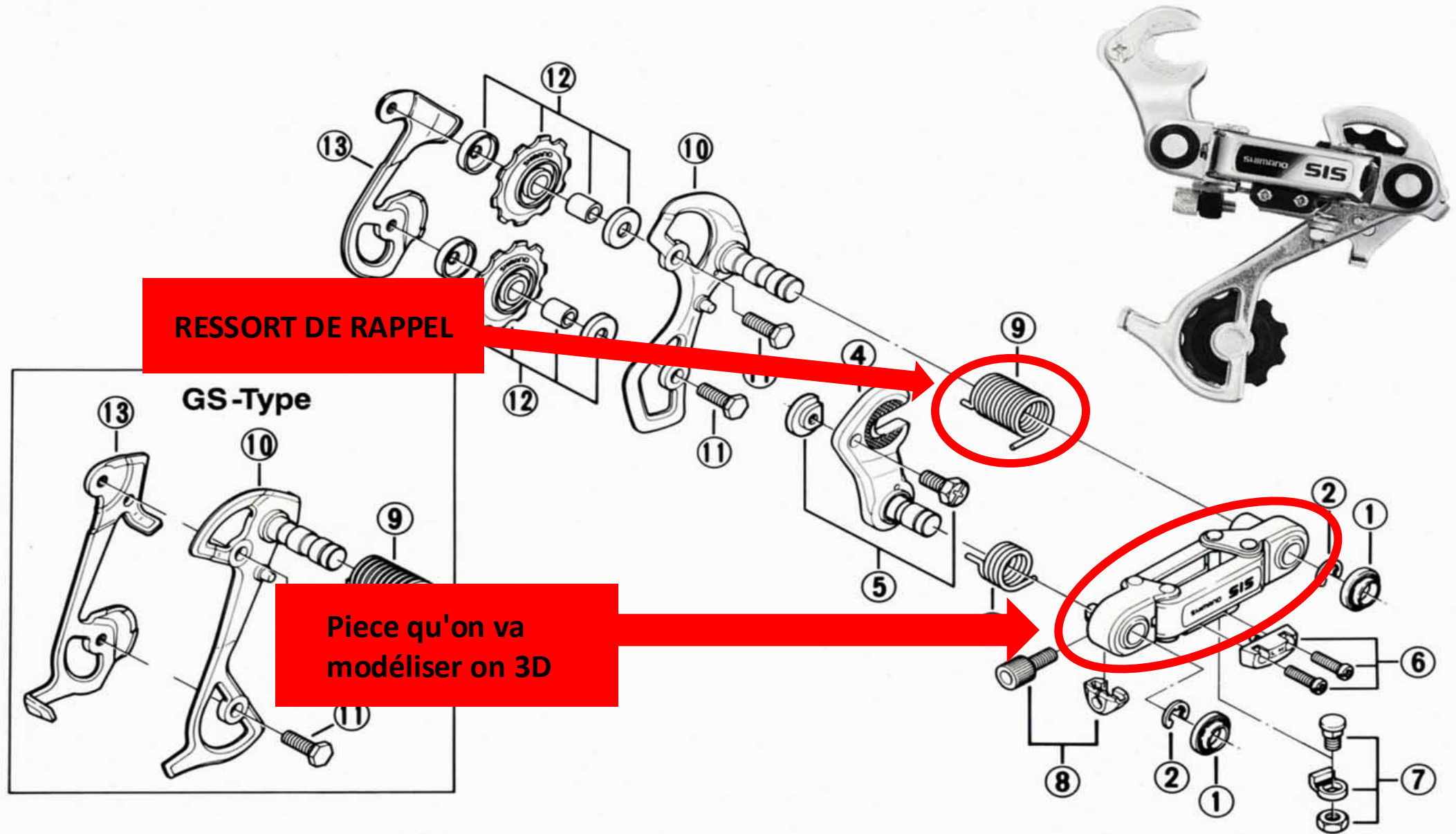


Etude de dérailleur

- Cette pièce de dérailleur est la pièce responsable de la translation circulaire de l'ensemble, alors l'étude sera centrée autour de cette pièce



RD-TY15-GS Long Cage



MODÉLISATION 3D

A l'aide du logiciel CATIA V5 de conception assistée par ordinateur et la pièce réel nous faisons la modélisation de cette pièce selon une échelle « vraie grandeur » 1 : 1.

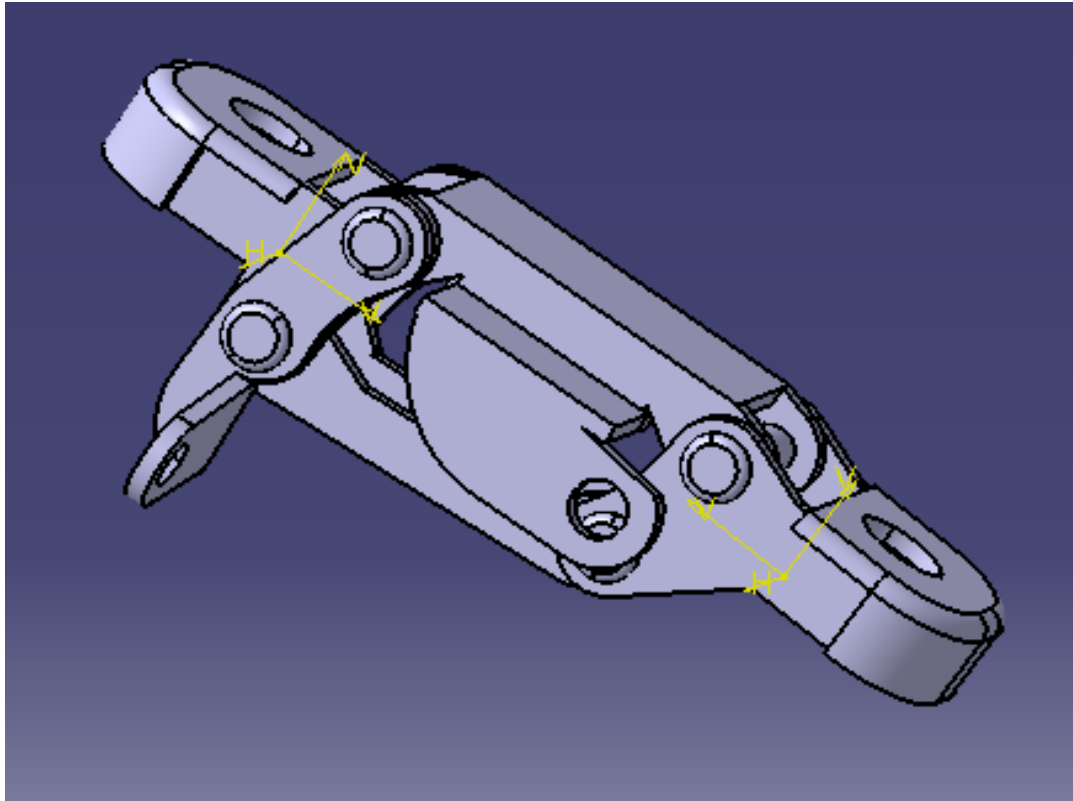


Figure 5 : Conception 3D de pièce



Figure 4 : Pièce réel

Choix de l'actionneur :

Servomoteur SG90

Composant du servomoteur :

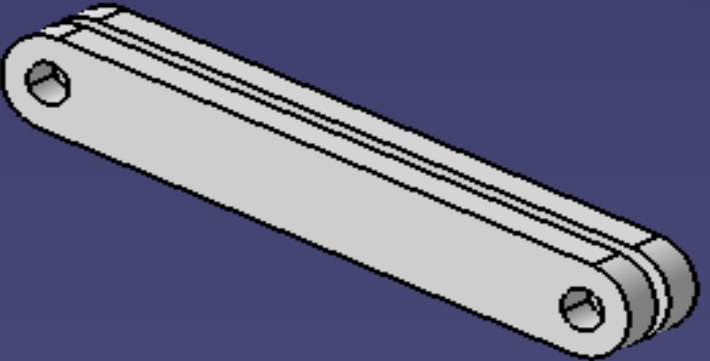
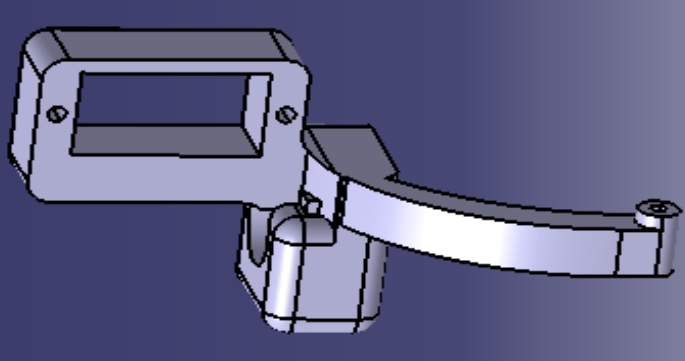
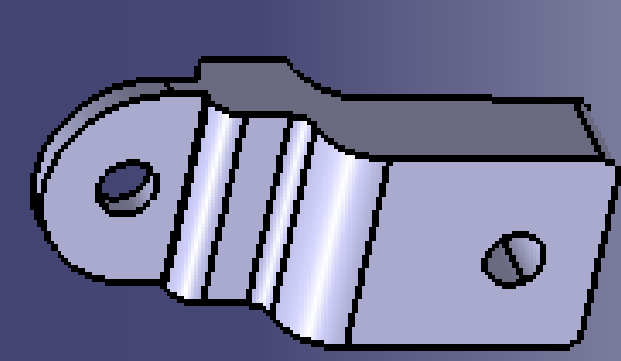
- Moteur électrique à courant continu
- Engrenage pour former un réducteur
- Capteur de position de l'angle d'orientation de l'axe
- Carte électronique pour le contrôle de position

Pourquoi le servomoteur ?

- ✓ • Angle de rotation entre 0 et 180
- ✓ • Grande couple à la sortie
- ✓ • Contrôle de position avec haute précision



Pièces Mécaniques Imprimées en 3D et Fonctions Associées:

Tige :	SUPPORT:	BRAS:
<p>Permettre de transmettre le mouvement du servomoteur</p>	<p>Permettre de fixer le servomoteur avec le corps fixe de dérailleur</p>	<p>Transmettre le mouvement de la tige au dérailleur</p>
		

Impression des pièces:



MÉCANISME:

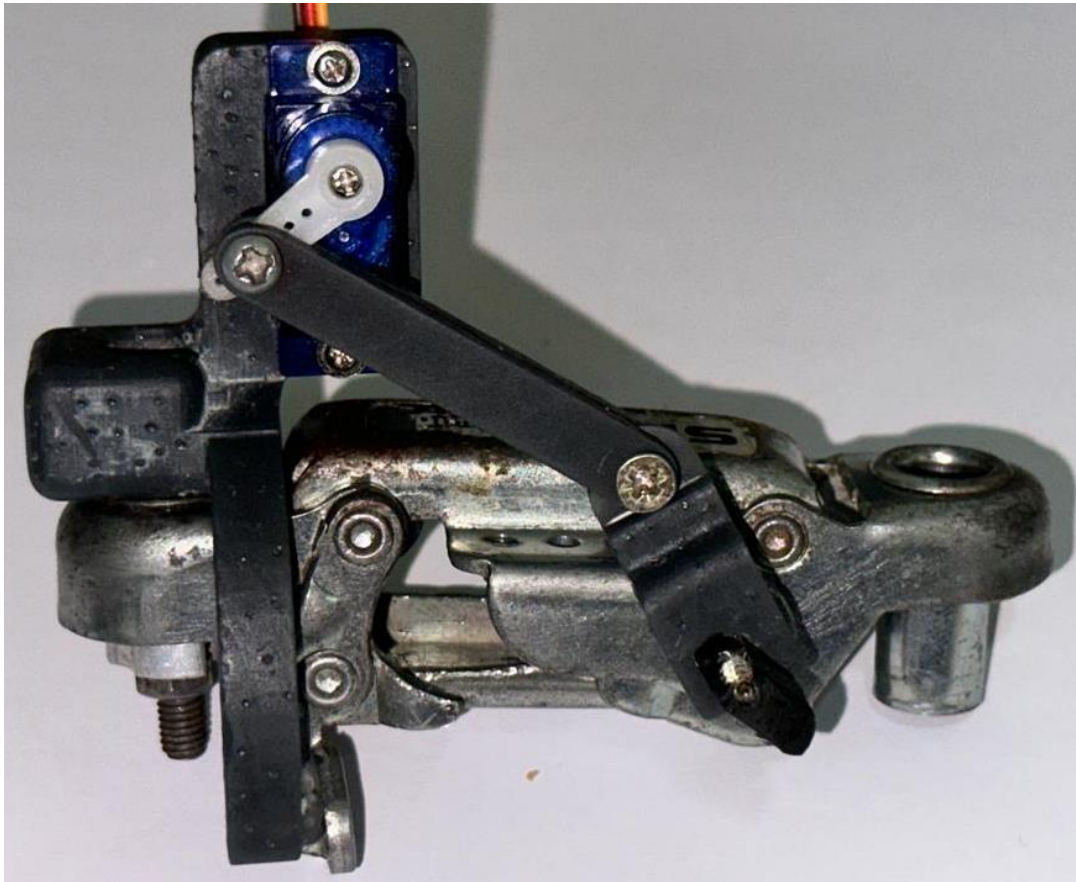


Figure 9: mécanisme réel

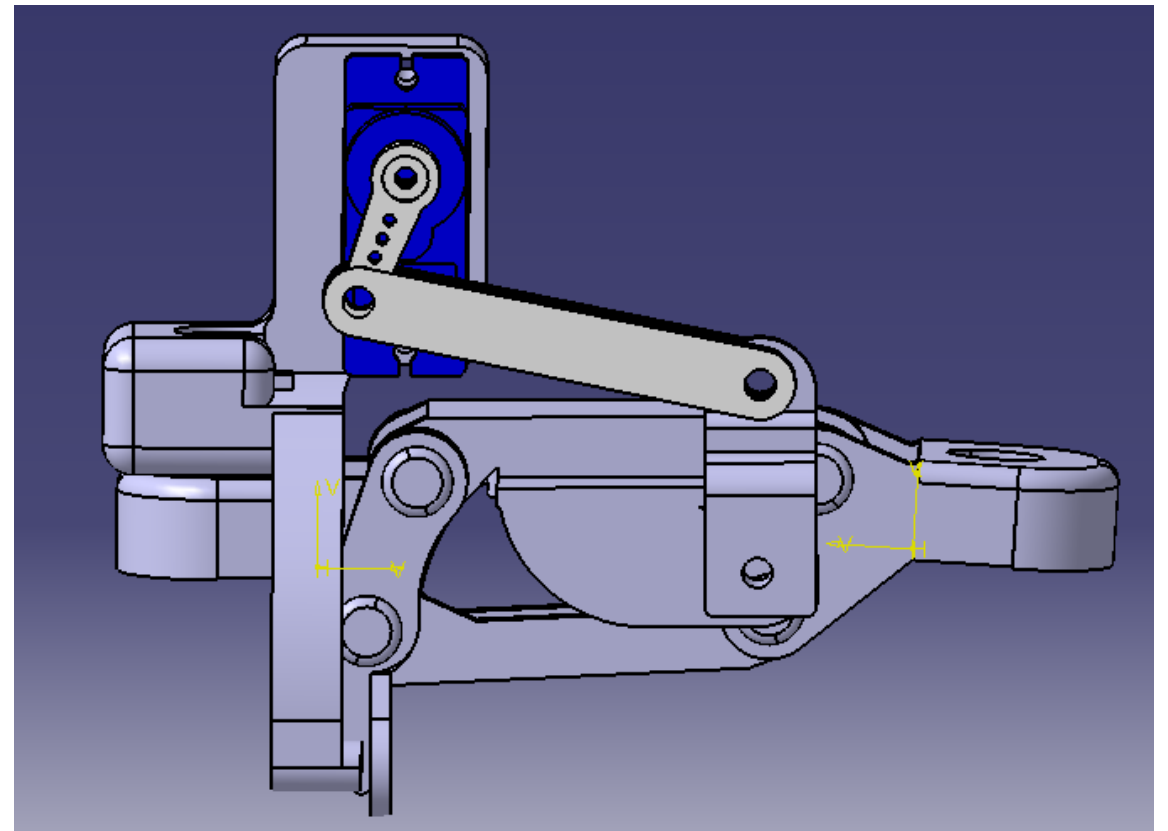


Figure 10: conception du mécanisme en 3D

Test du mécanisme:

1^{ER} RAPPORT:

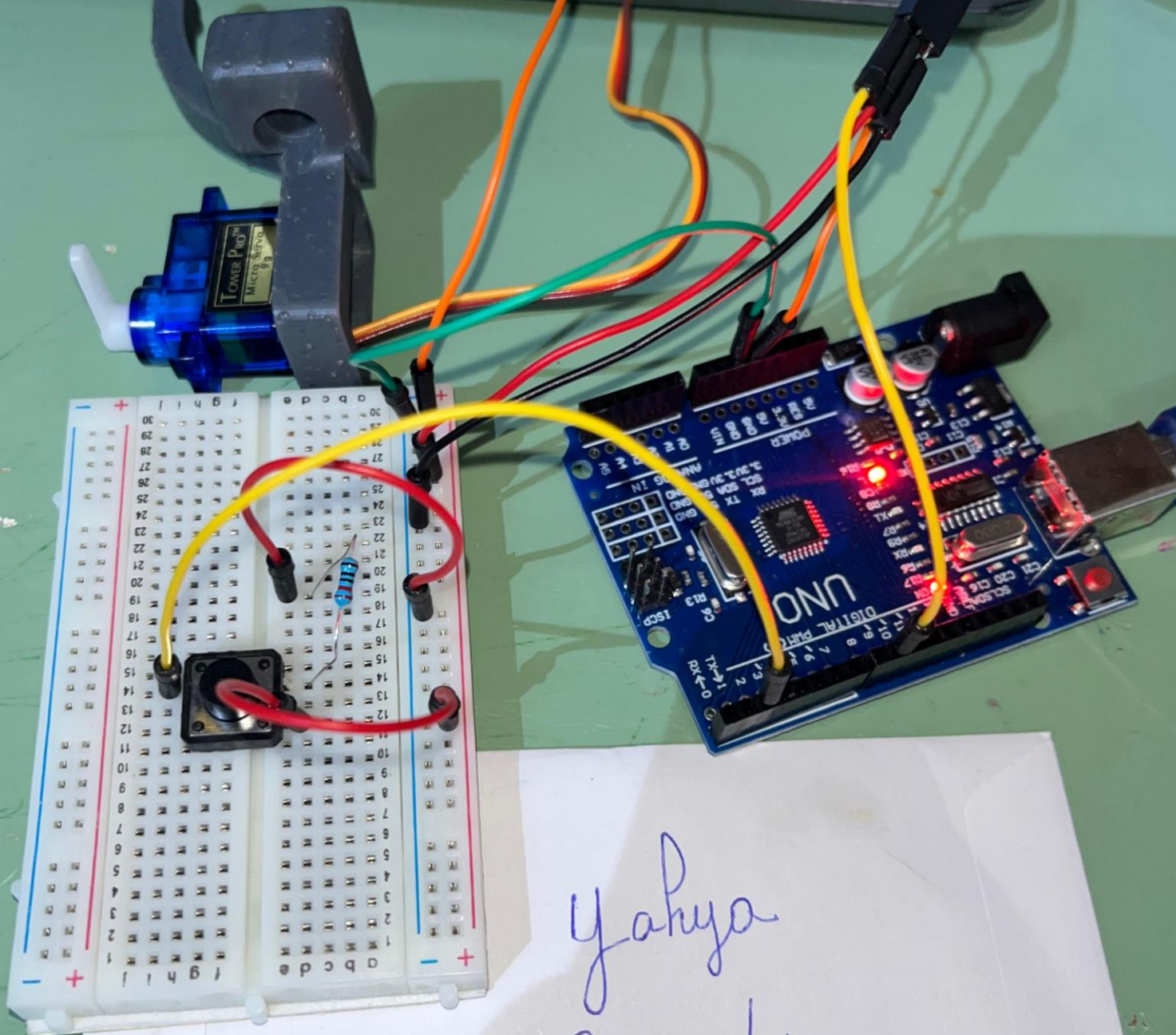


Le mécanisme crée est capable de créer de varier la position du dérailleur



5^{EM} RAPPORT:





POSITION DU SERVOMOTEUR:

Grâce à ce circuit, je peux commander le servomoteur angle par angle afin de trouver la position exacte permettant de s'intercaler sur chaque rapport.

POSITION DES RAPPORTS:

En commençant par un angle de 0 degré, j'incrmente de 2 degrés jusqu'à trouver les angles suivants :

1ER RAPPORT:	2EM RAPPORT	3EM RAPPORT	4EM RAPPORT	5EM RAPPORT
10°	25°	40°	55°	70°

```
4 int pos = 0;
5 int boutonPlusPin = 7;
6
7 void setup() {
8   myservo.attach(9);
9   pinMode(boutonPlusPin, INPUT);
10  Serial.begin(9600);
11 }
12
13 void loop() {
14   // Bouton pour augmenter l'angle
15   if (digitalRead(boutonPlusPin) == HIGH) {
16     pos += 2;
17     delay(250); // délai pour éviter les rebonds
18     Serial.println(pos);
19   }
20
21   myservo.write(pos);
22   delay(15); // Délai pour le mouvement du servo
```

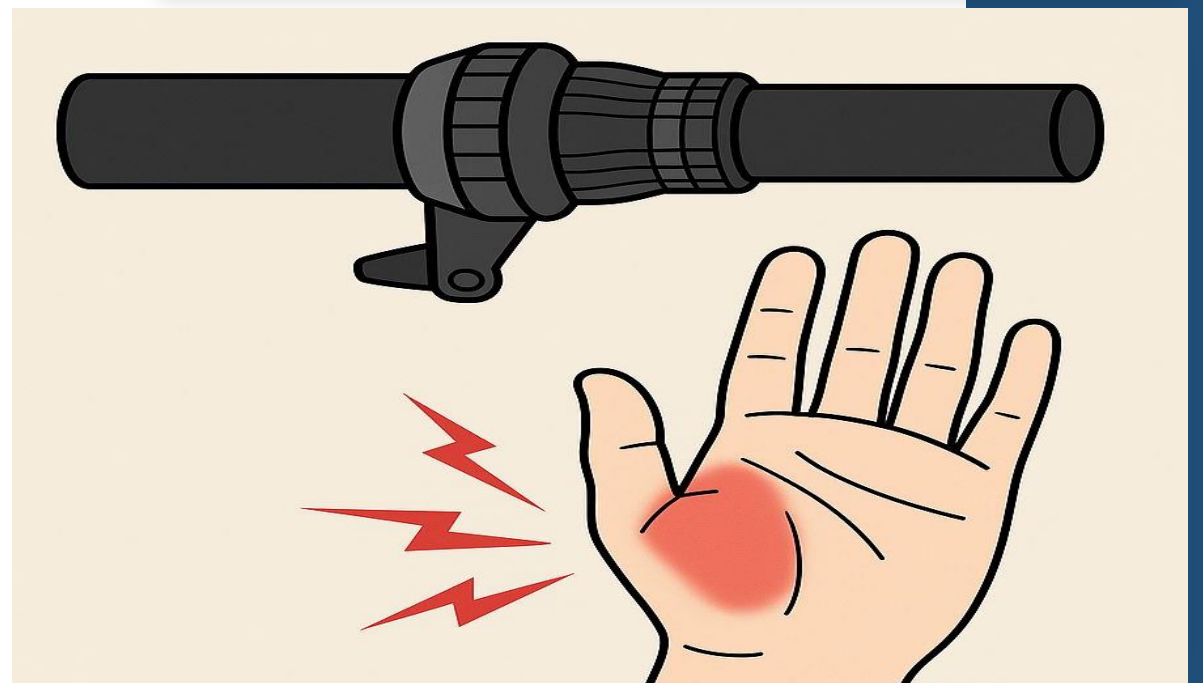
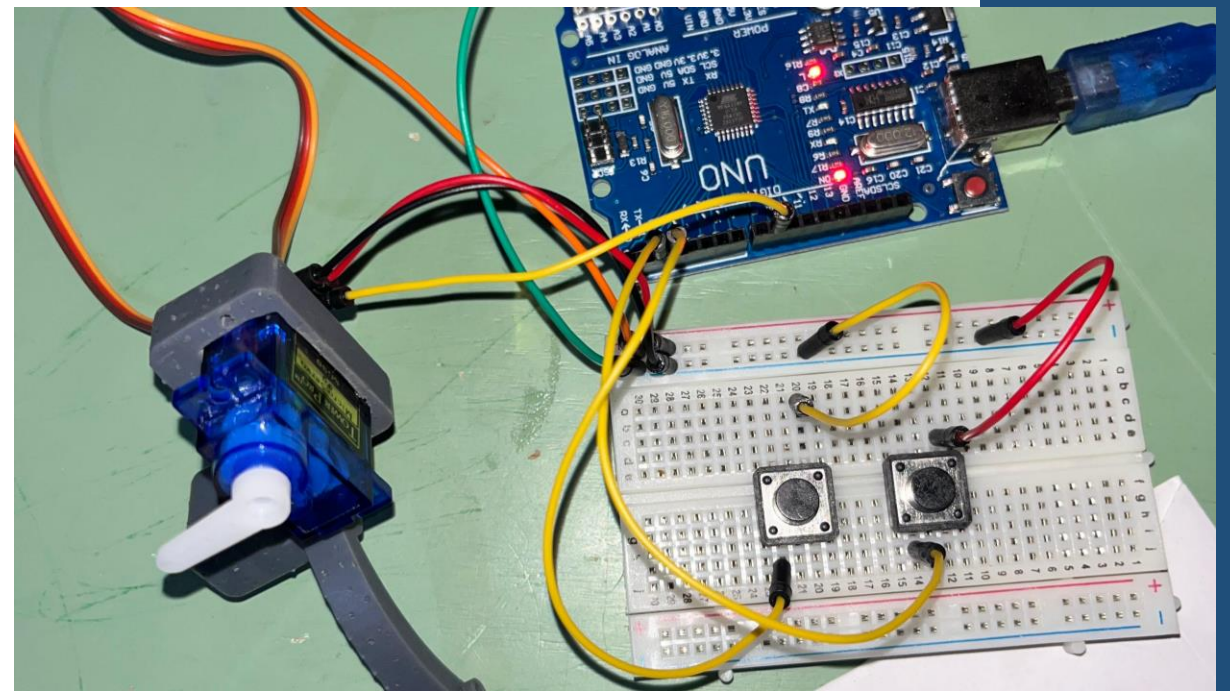
Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM4')

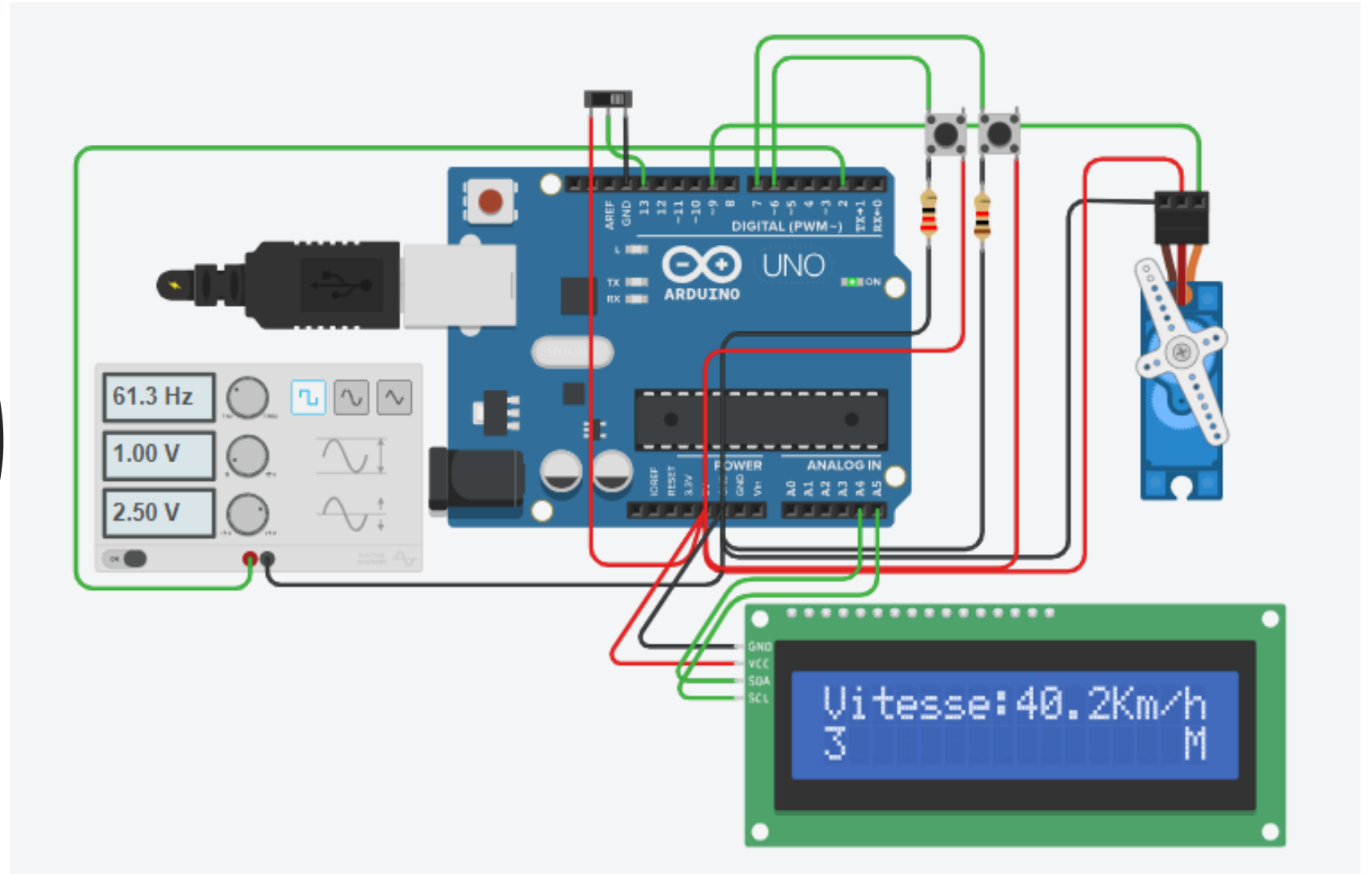
2
4
6
8
10

COMMANDE:

La manette rotative peut provoquer des douleurs à la main à cause de l'effort de rotation répété, surtout en conditions difficiles. Pour améliorer le confort et la simplicité d'usage, j'ai choisi de la remplacer par des boutons-poussoirs.



Simulation:

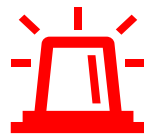


ALIMENTATION:



JUOVI power bank

Caractéristique	Valeur
Type de connecteurs	USB Type-C ×2, USB Type-A ×2
Marque	JUOVI
Capacité de la batterie	20 000 mAh
Couleur	Noir
Tension de sortie	5 Volts

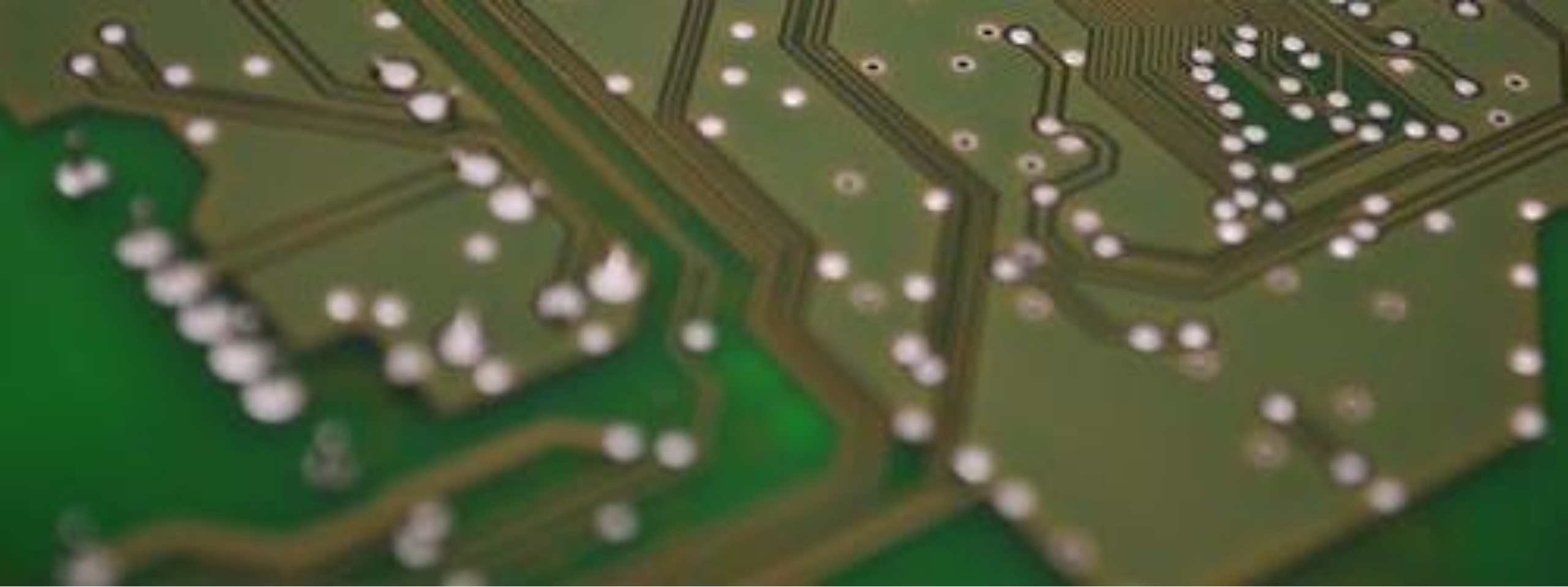


52h d'autonomie si le servo est très actif

The image shows a blue Arduino Uno R3 microcontroller board connected to a white breadboard. Two servomotors are mounted on the breadboard, and a complex mechanical linkage mechanism is attached to them. The mechanism consists of metal plates, gears, and a black plastic handle. A green checkmark icon is positioned above the main text. The background is a plain, light-colored surface.

Conclusion sur le 1^{ER} OBJECTIF:

Grace à un mécanisme adapté et un servomoteur j'ai réussi à rendre électrique le dérailleur.



2^{ém} Objectif : • Automatisation du
changement de rapport

CHOIX DU PLATEAU:

Critère	Monoplateau	Double Plateau	Triple Plateau
Simplicité mécanique	✓	✗	✗
Poids réduit	✓	✗	✗
Moins d'entretien	✓	✗	✗
Moins de risques de déraillement	✓	✗	✗



ETUDE DE VITESSE:

Selon un sondage que j'ai réalisé sur un groupe Facebook, la majorité des participants préfèrent une cadence de pédalage autour de 80 tours par minute.

$$\text{Dveloppement} = \left(\frac{N_{\text{plateau}}}{N_{\text{pignon}}} \right) \times \text{circonfrence de la roue}$$

$$\text{Vitesse (m/min)} = \text{Cadence (tr/min)} \times \text{Dveloppement (m)}$$

Groupe de Cyclisme Marocain
Yahya Raoudi · 11 min · 🌐

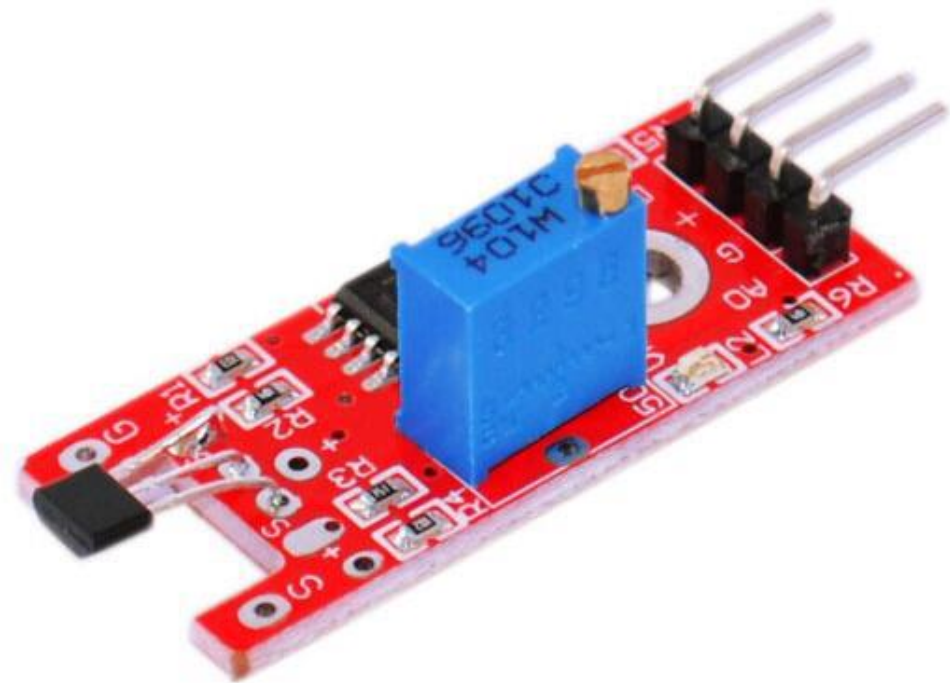
La meilleure cadence moyenne de pédalage selon vous?

- 80tr/min**  > ✕
- 90tr/min** 0 % > ✕
- 70tr/min**  > ✕

CALCUL DE VITESSE MAX DE CHAQUE RAPPORT:

Pignon (dents)	Rapport (plateau/pignon)	Développement (m)	Vitesse maximale (km/h)
11	2.91	6.12	29.39
13	2.46	5.18	24.87
15	2.13	4.49	21.55
18	1.78	3.74	17.96
21	1.52	3.21	15.40

CAPTEUR DE VITESSE:

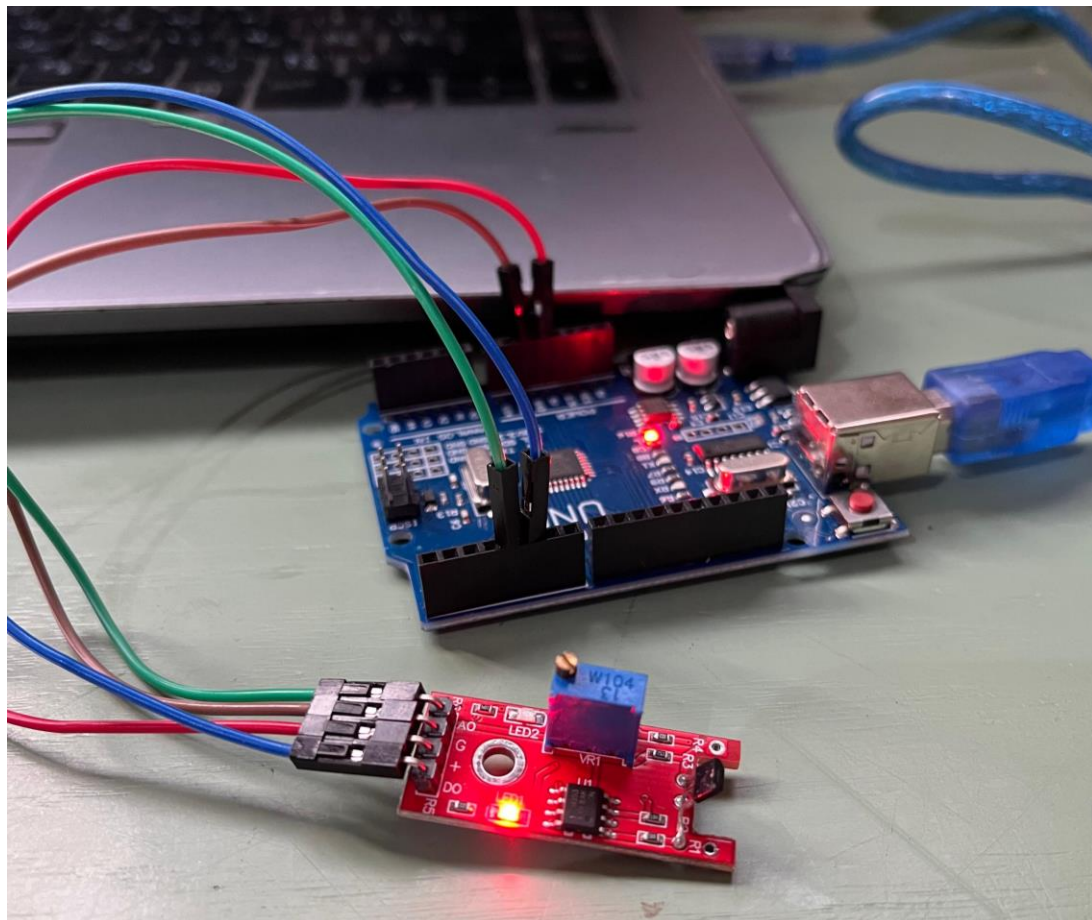


KY-024

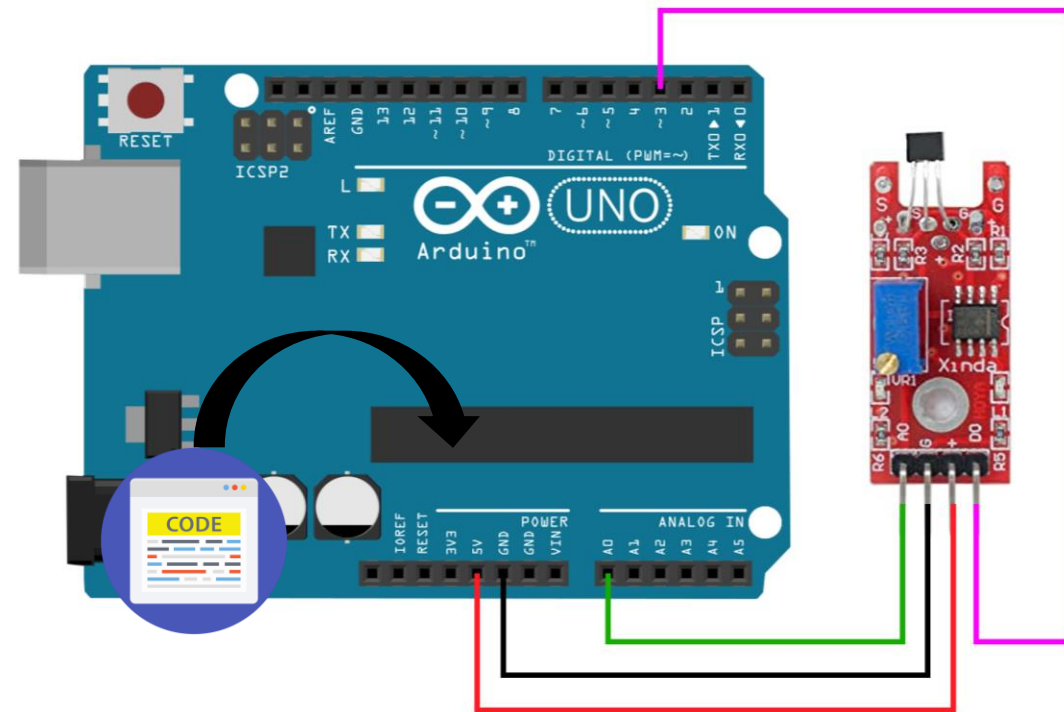
Caractéristique	Valeur / Description
Nom	KY-024 – Linear Magnetic Hall Sensor
Capteur Hall	SS49E (ou équivalent)
Tension d'alimentation	3.3 V à 5 V
Sorties	- AO : signal analogique (0–Vcc) - DO : signal digital (HIGH/LOW selon seuil)
Sensibilité	Proportionnelle à la densité de flux magnétique
Temps de réponse	~3 μ s (capteur Hall)
Type de détection	Champ magnétique axial (N/S face au capteur)
LEDs	- Power (rouge) - Signal (verte quand seuil dépassé)
Potentiomètre	Ajuste le seuil de détection pour la sortie DO
Plage analogique utile (SS49E)	\approx 0.88 V à 4.2 V autour de 2.5 V (sans champ)

CAPTEUR DE VITESSE:

Circuit réel

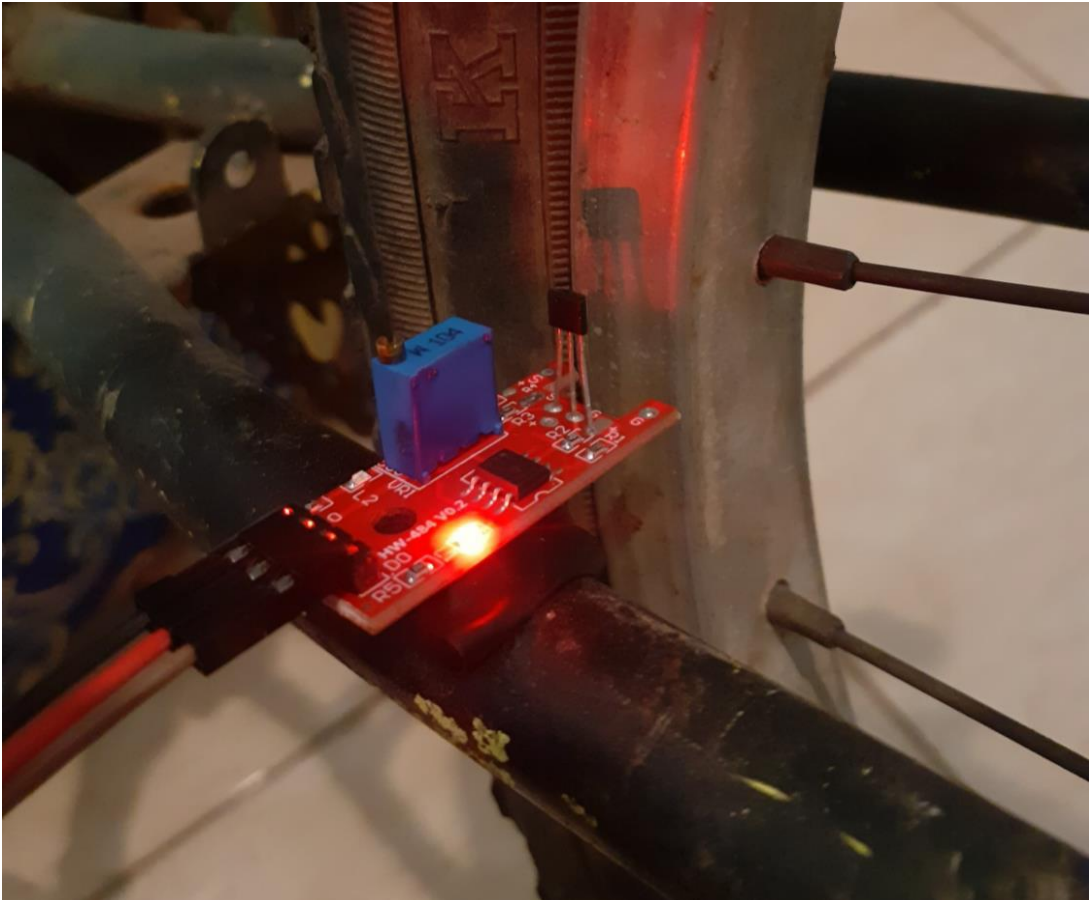


simulation de circuit

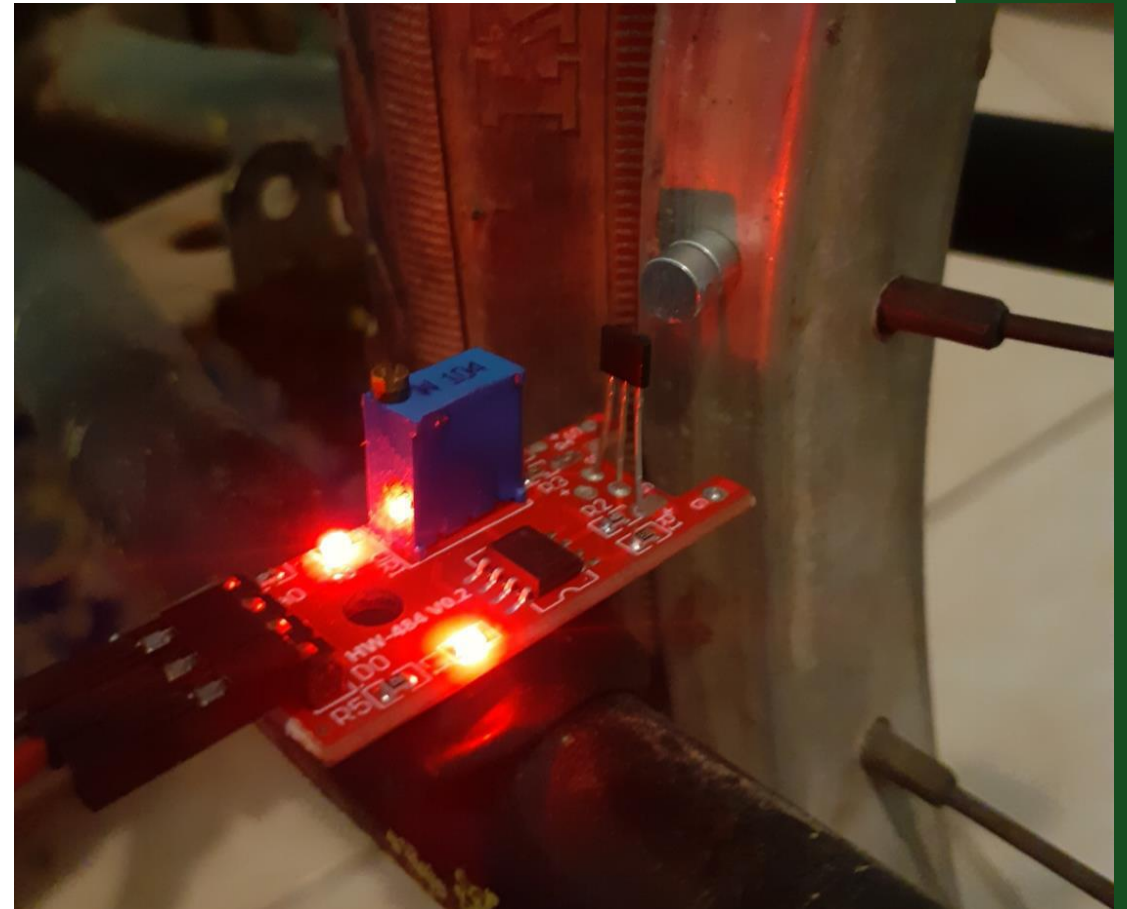


CAPTEUR DE VITESSE:

Aucune présence de champ magnétique



Présence de champ magnétique



CALCUL DE VITESSE:

Formule pour calculer la vitesse du vélo :

$$V = \frac{N \times C}{T}$$

- V : vitesse du vélo (en m/s)
- N : nombre de tour (nombre de pulsations par seconde)
- C : circonférence de la roue (en m)
- T : temps (en s)

COMMANDE DU MODE:



**UN INTEREPTEUR PERMET DE PASSER DU
MODE MANUEL EN MODE AUTOMATIC**



Conclusion sur le 2^{EM} OBJECTIF:

À l'aide du calcul précédent, une carte Arduino et un programme, j'ai réussi à commandé le dérailleur de façon automatique.

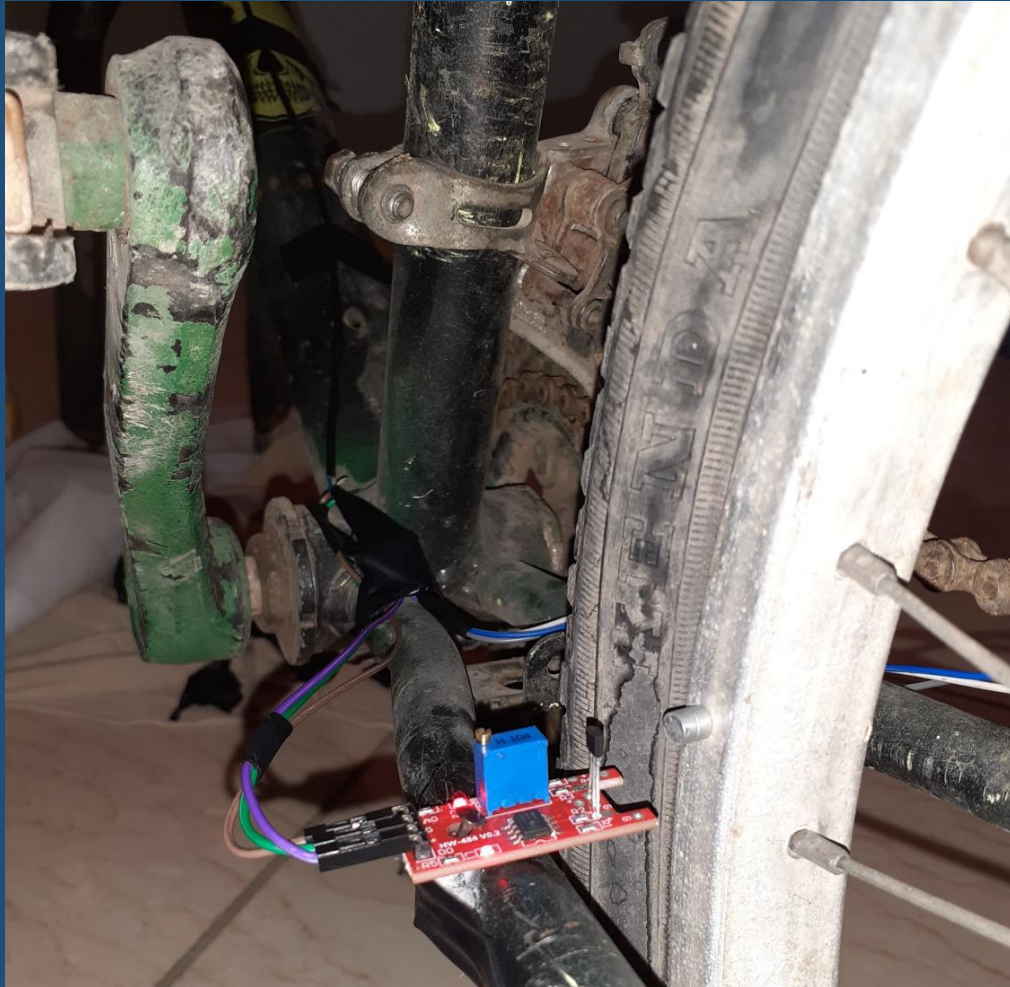
RÉALISATION

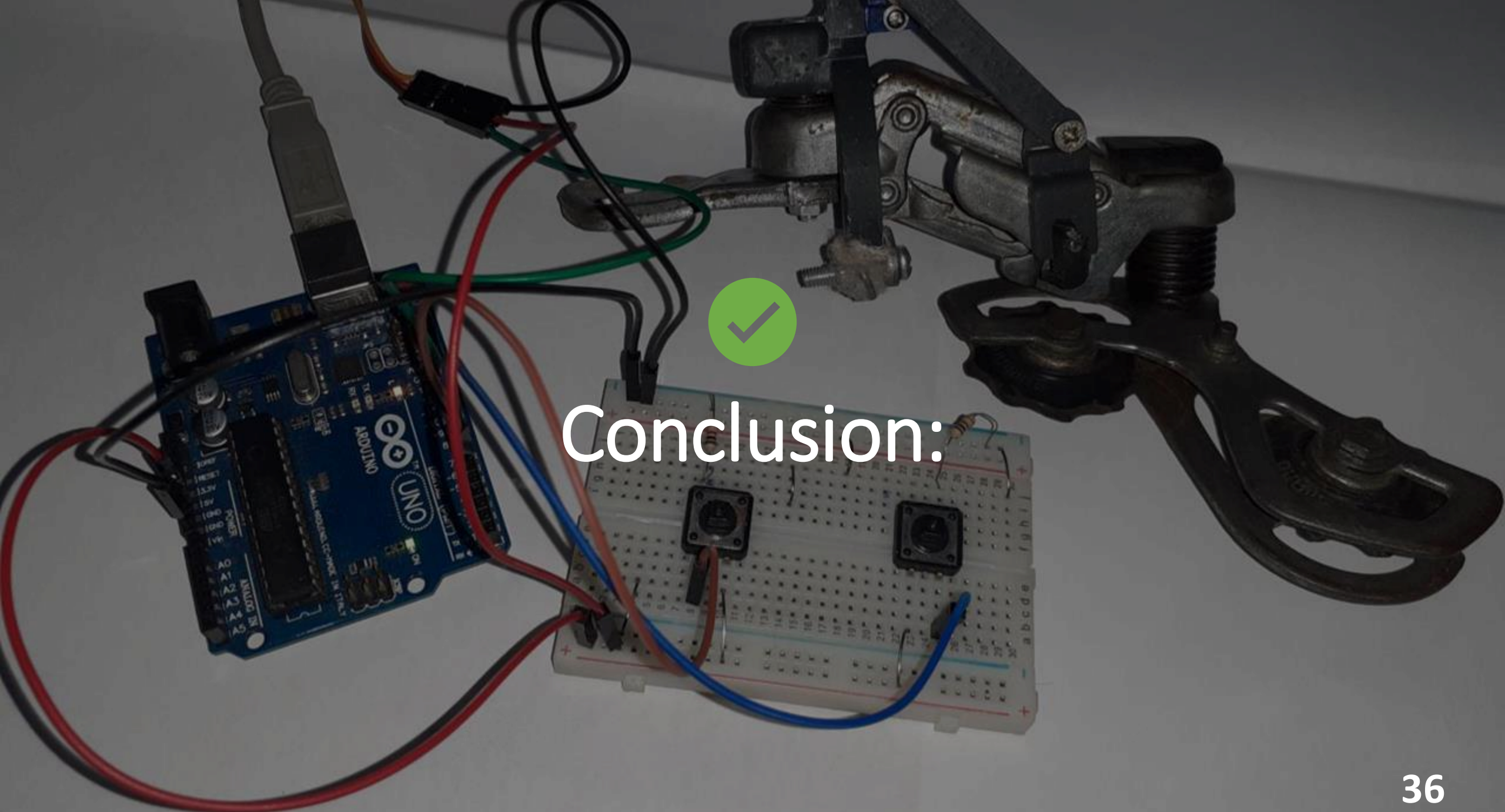


AFFICHAGE DES INFORMATIONS:



RÉALISATION





Conclusion:

```
File Edit Sketch Tools Help
sketch_jun12a.ino
1 #include <Servo.h>
2
3 const float C = 1.0; // 1 impulsion = 1 tour (1 aimant)
4 volatile int impulsions = 0;
5 unsigned long lastMillis = 0;
6 const unsigned long intervalle = 500; // mesure sur 0.5 s
7 float cadence = 0;
8 Servo derailleur;
9 int vitesses[5] = {10, 25, 40, 55, 70};
10 int vitesse_actuelle = -1;
11 void setup() {
12     pinMode(2, INPUT_PULLUP); // capteur effet Hall sur D2
13     attachInterrupt(digitalPinToInterrupt(2), detectImpulsion, FALLING);
14     derailleur.attach(9); // servomoteur branché sur D9
15     derailleur.write(vitesses[0]); // position initiale
16     Serial.begin(9600);
17 }
18 void loop() {
19     if (millis() - lastMillis >= intervalle) {
20         noInterrupts();
21         int N = impulsions;
22         impulsions = 0;
23         interrupts();
24         lastMillis = millis();
25         cadence = N * 120.0; // car 0.5 s → ×120 = tr/min
26         int rapport = choisirRapport(cadence);
27         if (rapport != vitesse_actuelle) {
28             derailleur.write(vitesses[rapport - 1]); // adapter l'angle
29             vitesse_actuelle = rapport;
30             Serial.print("Changement : rapport ");
31             Serial.print(rapport);
32             Serial.print(" (cadence : ");
33             Serial.print(cadence);
34             Serial.println(" tr/min");
35         }
36     }
37 }
38 void detectImpulsion() {
39     impulsions++;
40 }
41
42 int choisirRapport(float c) {
43     if (c < 50) return 1;
44     else if (c < 60) return 2;
45     else if (c < 70) return 3;
46     else if (c < 85) return 4;
47     else return 5;
48 }
49
```

ANNEXE

- CODE ARDUINO POUR FONCTIONNEMENT AUTOMATIQUE

```
Arduino Uno
Button.ino
1 #include <Servo.h>
2
3 Servo myservo;
4
5 const int boutonPlus = 2;
6 const int boutonMoins = 3;
7 const int nbRapports = 5;
8
9 int rapport = 0;
10 int rapports[nbRapports] = {10, 25, 40, 55, 70};
11
12 void setup() {
13     myservo.attach(9);
14
15     // Utilise les résistances internes (pas besoin de résistance externe)
16     pinMode(boutonPlus, INPUT_PULLUP);
17     pinMode(boutonMoins, INPUT_PULLUP);
18
19     Serial.begin(9600);
20
21     myservo.write(rapports[rapport]);
22     Serial.print("Démarrage → Rapport ");
23     Serial.print(rapport + 1);
24     Serial.print(" | Angle : ");
25     Serial.println(rapports[rapport]);
26 }
27
28 void loop() {
29     // Bouton appuyé = LOW (car pullup actif)
30     if (digitalRead(boutonPlus) == LOW) {
31         if (rapport < nbRapports - 1) {
32             rapport++;
33             myservo.write(rapports[rapport]);
34             Serial.print("↑ Rapport : ");
35             Serial.print(rapport + 1);
36             Serial.print(" | Angle : ");
37             Serial.println(rapports[rapport]);
38             delay(300); // Anti-rebond
39         }
40     }
41
42     if (digitalRead(boutonMoins) == LOW) {
43         if (rapport > 0) {
44             rapport--;
45             myservo.write(rapports[rapport]);
46             Serial.print("↓ Rapport : ");
47             Serial.print(rapport + 1);
48             Serial.print(" | Angle : ");
49             Serial.println(rapports[rapport]);
50             delay(300);
51         }
52     }
53
54     delay(10);

```

ANNEXE

- CODE ARDUINO POUR FONCTIONNEMENT MANUEL



MERCI POUR
VOTRE
ATTENTION

