



Réalisé par: Salmane Baba

Code CNC: SF054T

Encadré par: Oualifi Khadija







# Introduction:



Figure 1: La Draisienne du Baron Karl Drais



Figure 2: Vélo VTT

# Pourquoi changer le rapport de transmission ?

On a:  $\overline{F_{\text{cycliste}}} = \overline{F_{\text{normale}}} + \overline{F_{\text{tangentielle}}}$  $C = F \text{ tangentielle } x d = F \text{ cycliste } x \sin \alpha x d$ 

la force transmise sur la roue est :

Froue / route = F cycliste x d x  $\sin \alpha$  /(rapport de transmission x R)

or F motrice = F roue / route = F route / roue

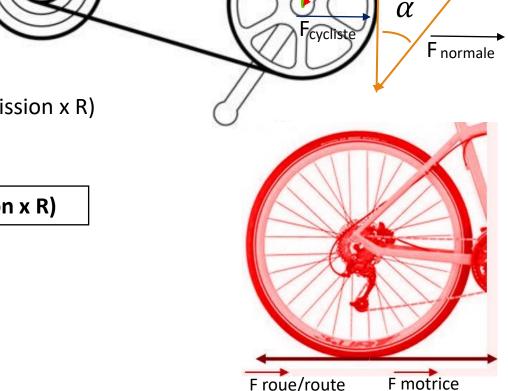
Donc : F motrice = F cycliste x d x sin $\alpha$  / (rapport de transmission x R)

Avec:

C: couple en N/m; d: longeur de la manivelle;

lpha : angle de la manivelle avec la vertical

R: rayon de la roue



Ftangentielle



### Sur une pente;

Si **V=Cte** :

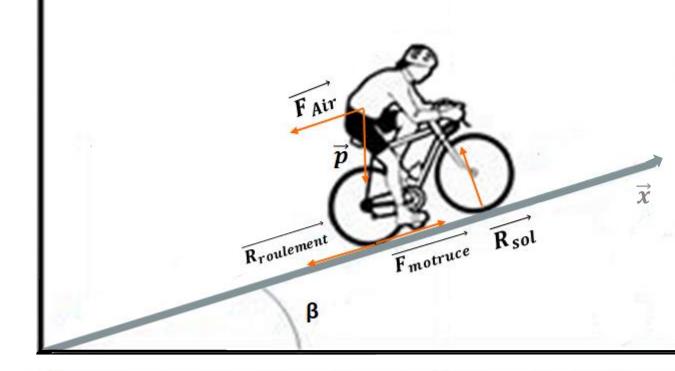
Loi de newton:  $\sum \vec{F} = \vec{0}$ 

On applique la loi au système vélo, dans le référentiel terrestre supposé galiléen:

$$\overrightarrow{F_{motrice}} + \overrightarrow{F_{air}} + \overrightarrow{R_{sol}} + \overrightarrow{P} + \overrightarrow{R_{roulement}} = \overrightarrow{0}$$

Projection sur l'axe  $\overrightarrow{OX}$ :

Fmotrice – Fair – Rroulement – mgsin  $\beta = 0$ 



F motrice = F air + R roulment + mg sin  $\beta$ 

F motrice = F cycliste x d x sin  $\alpha$ / (rapport de transmission x R)



### Les inconvénients des dérailleurs manuels





**Apprentissage requis** 

### Problématique:

Comment développer un dérailleur automatique, capable d'ajuster automatiquement les vitesses en fonction de la cadence de pédalage.



### Les objectifs:

Etudier le système de mesure de la cadence de pédalage.

Etudier la régulation de position du câble du dérailleur.

02

04

01

Présenter le système et élaborer le cahier des charges.

03

Etudier le système de tirage et de libération du câble du dérailleur.

05

Dimensionnement et choix du batterie.

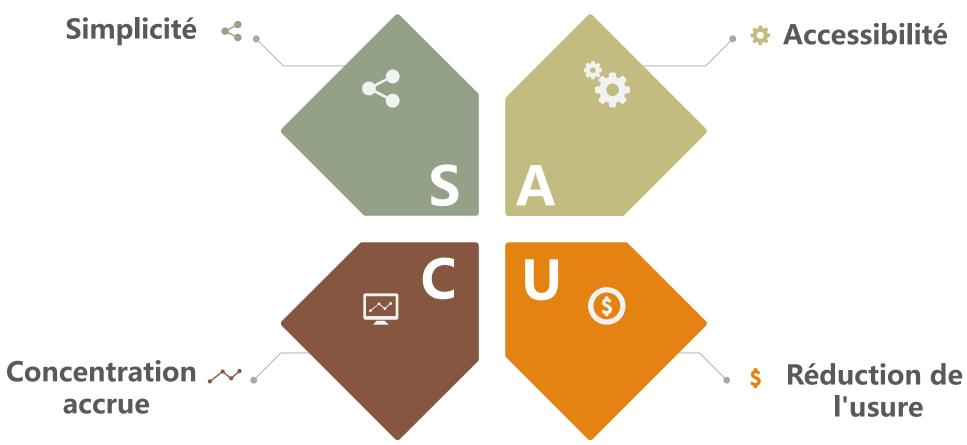
### Objectif N°O1

Présenter le système et élaborer le cahier des charges.





### Les avantages du derailleur automatique



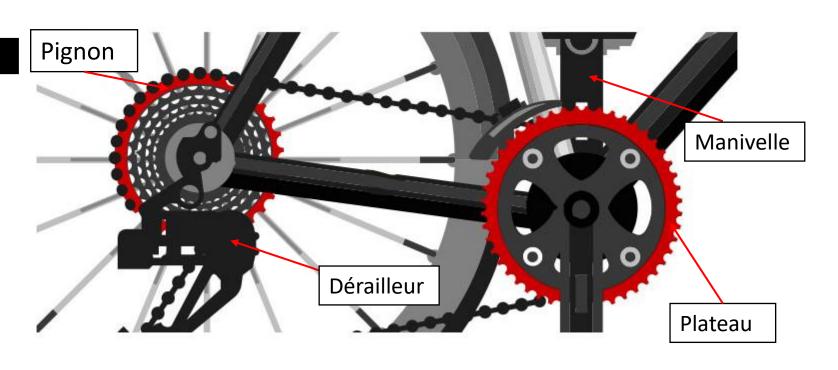


### Vélo à étudié:

Type de dérailleur

**Shimano Altus** 

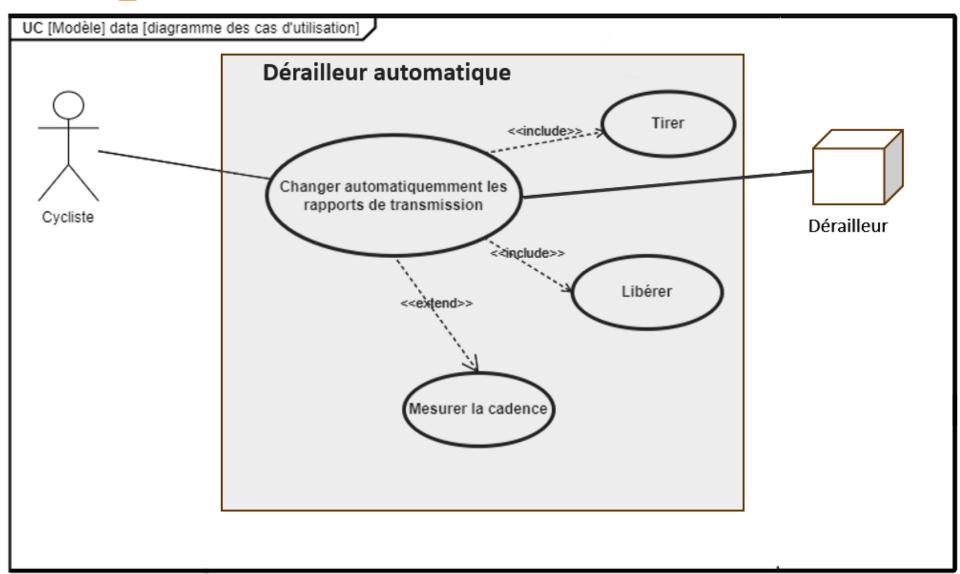
Pignons	Nombre de dents
1	11
2	13
3	15
4	17
5	20
6	23
7	26
8	30



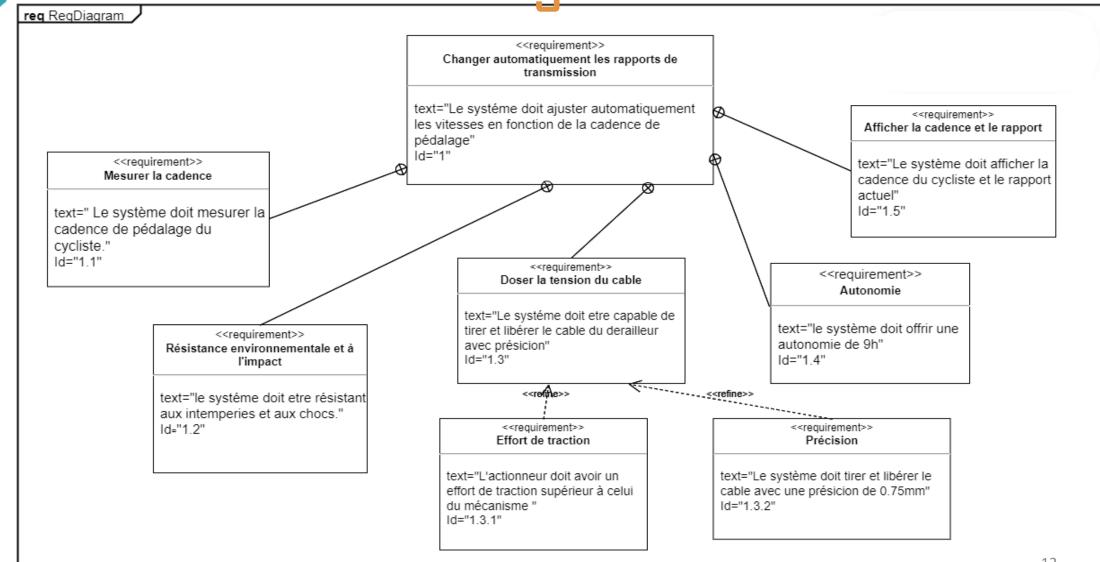
Plateau	Nombre de dents
1	34



## Diagramme de cas d'utilisation:



### Cahier de charge:



### Objectif N°02

Etudier le système de mesure de la cadence de pédalage.





### Capteur ILS et aimants

#### Capteur MDCG-4-13-23:

- Tension maximal: 200V
- Courant maximal: 0.5A
- Puissance maximal: 10W
- Fréquence maximal : 500Hz
- Vibration resistance(10-2000Hz):30G
- Senibilité nominale : 20 AT

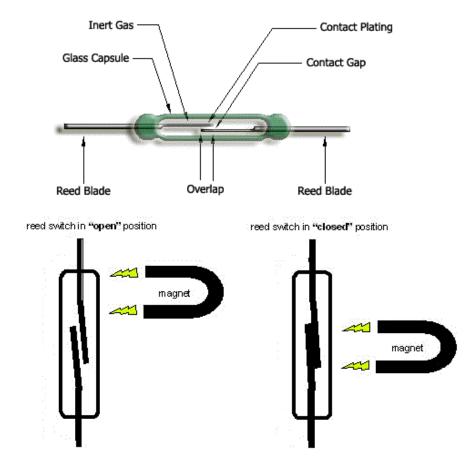
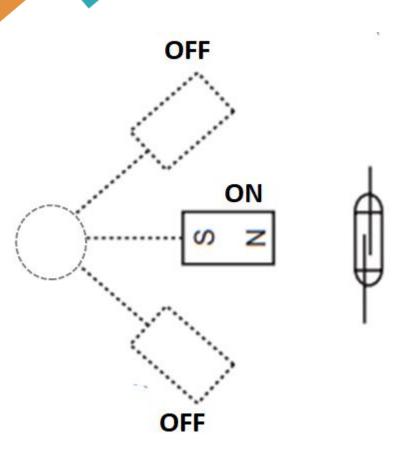


Figure 4: Principe de fonctionnement du capteur

## Montage du capteur:



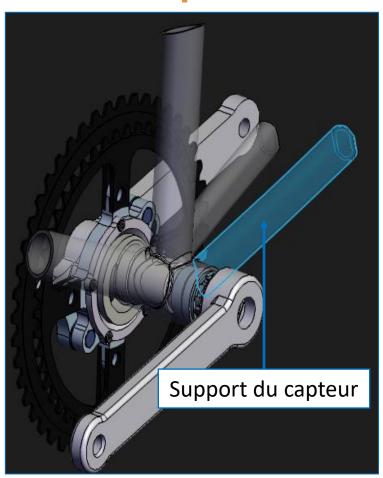
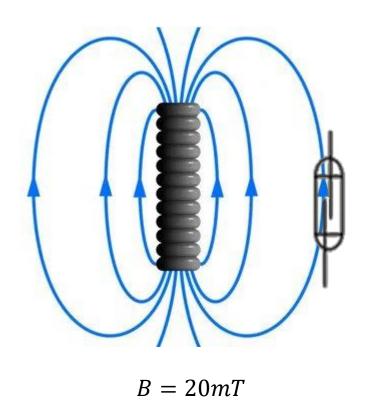


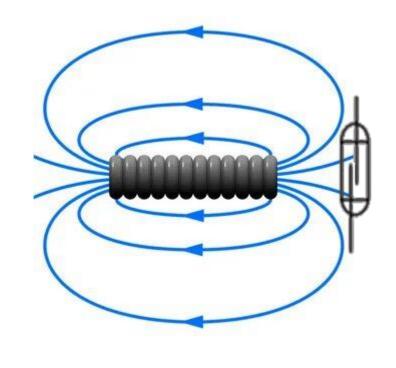


Figure 5 et 6



# Dimensionnement d'un Aimant pour un capteur MDCG-4-13-23:





B = 40mT



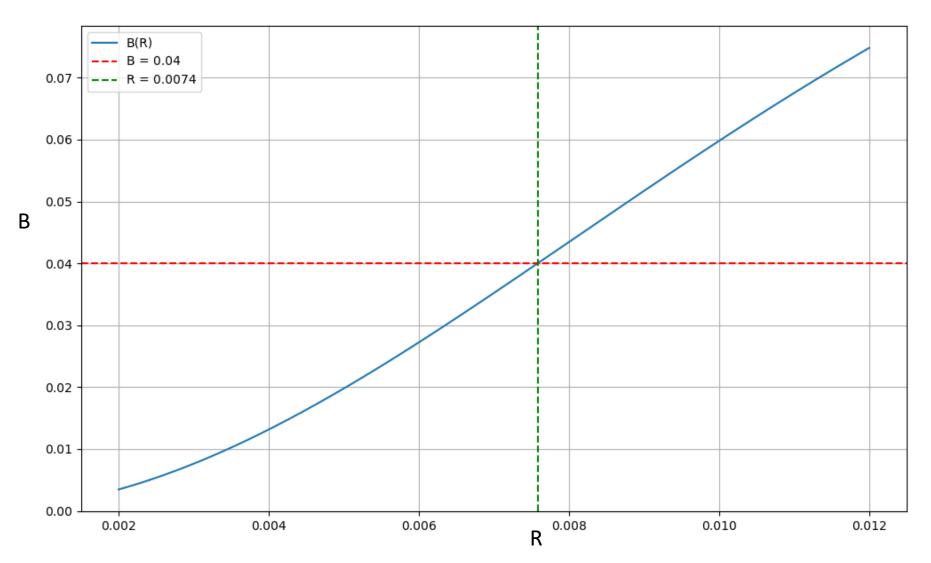
#### Paramètres et hypothèses

- Distance entre capteur et manivelle d=2.5cm
- Largeur de la manivelle L=2.5cm
- Aimant cylindrique en néodyme N35(Br=1.24T)
- Epaisseur de l'aimant e=10mm
- Distance entre la surface de l'aimant et le capteur x=15mm

#### **Equation du champs axial d'un aimant cylindrique:**

$$B(x) = \frac{B_r}{2} \times \left(\frac{e + x}{\sqrt{R^2 + (e + x)^2}} - \frac{x}{\sqrt{R^2 + x^2}}\right)$$

# Courbe de B(R):





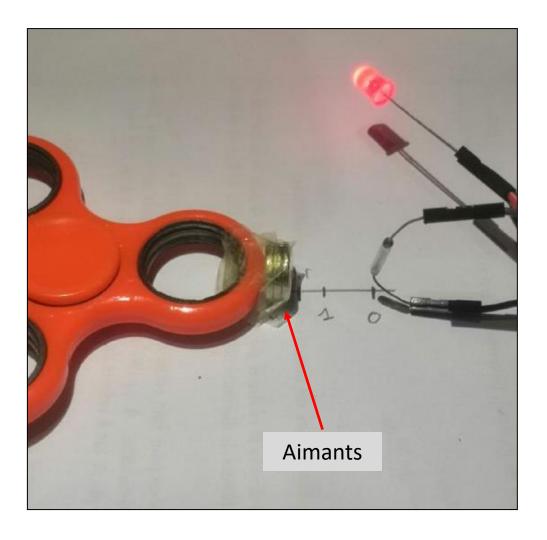
### Choix de l'aimant

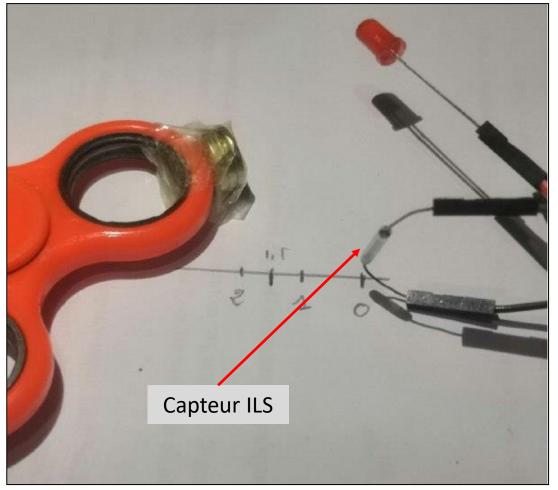
### **Information produit**

- Matériau NdFeB
- > Former un disque
- > Diamètre 15 mm
- > Hauteur 5 mm
- > Direction de l'aimantation axiale (parallèle à la hauteur)
- > Revêtement Nickel (Ni-Cu-Ni)
- > Magnétisation N35
- > Max. température de travail 80°C



## Vérification des résultats:





### Objectif N°03

Etudier le système de tirage et de libération du câble du dérailleur.

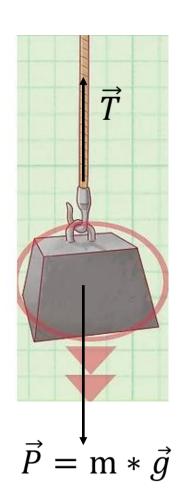
• • •



### Nesurer la force de traction sur le câble :

Appliquant la 1ére loi de Newton :  $\sum \vec{F} = \vec{0}$  On trouve:

$$T = m * g$$
  
=  $6kg * 10m/s^2$   
= **60N**



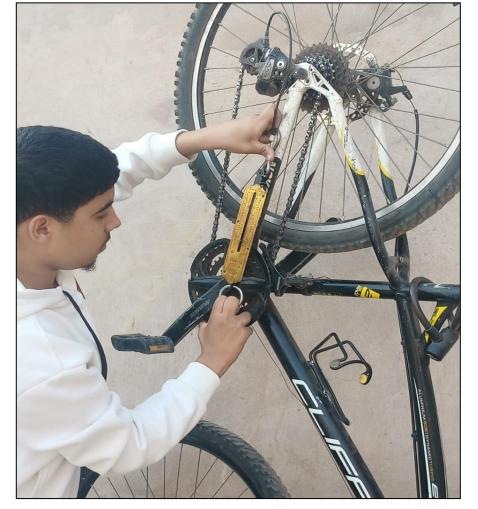


Figure 8: Mesure du force de traction



### Mesure des distances de déplacement du câble pour les différents rapports de transmission:

Pignon	Distance totale de relachement du cable(mm)
1	0
2	3.7
3	7.3
4	10.9
5	14.6
6	18.2
7	21.8
8	25.5



### Choix de l'actionneur:

#### Hypothèses:

- Masse de la poulie est négligeable
- Angle de rotation du moteur: 180
- Rayon de la poulie R=1.5cm

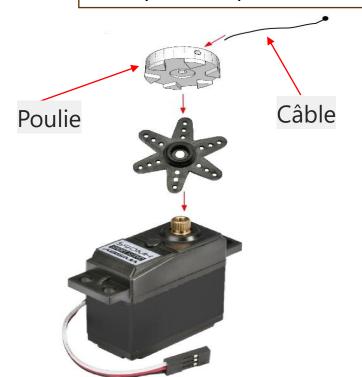
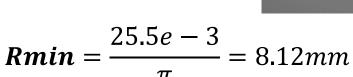


Figure 9: Montage du poulie

#### Déplacement



• 
$$Rmin = \frac{D}{\theta}$$
  
AN:



Le couple moteur:

$$C = T * R$$
  
 $C = 60N * 0.015m$   
 $= 0.9Nm$ 



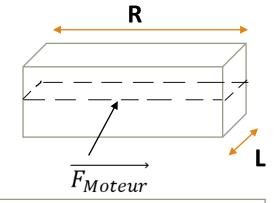
### Choix du moteur:

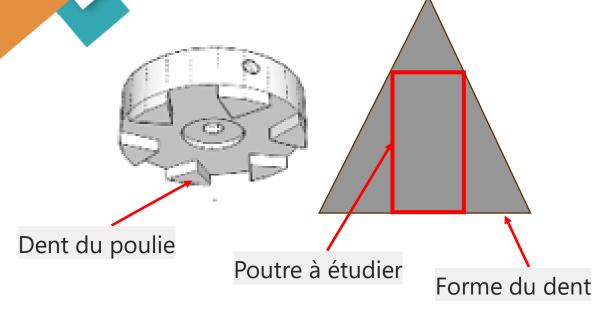
### MG995R

- Voltage: 4.8-6.6V
- Courant: 170-1200 mA
- Angle de rotation: 180°
- Couple: 0.92-1.07Nm



Etude de la résistance du poulie :





Hypothèse:

- La dent est modélisée par une poutre rectangulaire de longueur R=8mm et de largeur L=3mm
- Coefficient de s écurité: s=3

La dent est sollicité au cisaillement sous l'action du câble et de l'action du moteur

Condition de résistance: 
$$\tau_{max} \leq \frac{R_{pg}}{s} = \frac{R_e}{2s}$$

$$\Rightarrow R_e \geq \tau_{max} \times 2s$$

$$F_{Moteur} = \frac{C_{max}}{R/2} = \frac{0.92}{4 \times 10^{-3}} = 230N$$

$$\tau_{max} = \frac{F_{Moteur}}{Surface} = \frac{230}{3 \times 8} = 9.58 \text{ MPa}$$

$$\Rightarrow R_e \geq 9.58 \times 2s = 57.5 \text{ MPa}$$



### Choix du matériaux: PTFE ou PLA+

#### Polytétrafluoroéthylène (PTFE)

Propriétés Générales				
Masse Volumique	2140	-	2200	kg/m^3
Prix	124.5	-	224.1	MAD/kg
Propriétés Mécaniques				
Module de Young	0.4	-	0.552	GPa
Limite élastique	15	-	25	MPa
Résistance en traction	20	-	30	MPa
Allongement	200	-	400	%
Mesure de dureté Vickers	5.9	-	6.5	HV
Limite de fatigue	5.75	-	7	MPa
Ténacité	* 1.315	-	1.8	MPa.m^1/2



- Mass volumique  $\rho = 2140 2200kg/m^3$
- Resistance élastique:  $R_e = 15 25 MPa$

#### acide polylactique amélioré(PLA+)

Propriétés Générales				
Masse Volumique	1240	-	1300	kg/m^3
Prix	200	-	400	MAD/kg
Propriétés Mécaniques				
Module de Young	2.5	-	3.5	GPa
Limite élastique	50	-	70	MPa
Résistance en traction	55	-	75	MPa
Allongement	5	-	10	%
Mesure de dureté Vickers		-		HV
Limite de fatigue		-		MPa
Ténacité	* 2.0	-	4.0	MPa.m^1/2

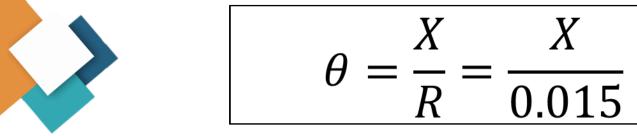


- Mass volumique  $\rho = 1240 1300 kg/m^3$
- Resistance élastique:  $R_{\rho} = 50 70 MPa$

### Objectif N°04

Etudier la régulation de position du câble du dérailleur.





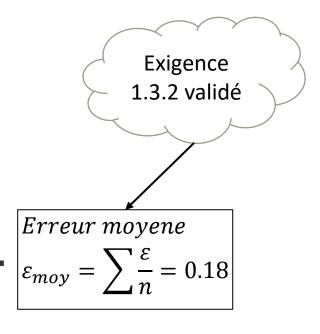
Pignon	Distance totale de tirage du Câble (mm)	Angle de rotaion
1	0	0
2	3.7	14.13°
3	7.3	27.88°
4	10.9	41.63°
5	14.6	55.76°
6	18.2	69.51°
7	21.8	83.26°
8	25.5	97.4°

### Mesure des distance:

Angle de rotation	La distance souhaité	La distance mesurée	Erreur $oldsymbol{arepsilon}$
0	0	0	0
14.13°	3.7	3.5	0.2
27.88°	7.3	7	0.3
41.63°	10.9	11	0.1
55.76°	14.6	14.5	0.1
69.51°	18.2	18	0.2
83.26°	21.8	22	0.2
97.4°	25.5	25. 3	0.2

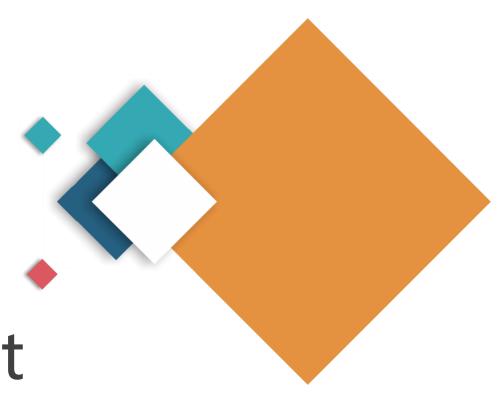
<<requirement>>
Précision

text="Le système doit tirer et libérer le cable avec une présicion de 0.75mm" ld="1.3.2"



### Objectif N°05

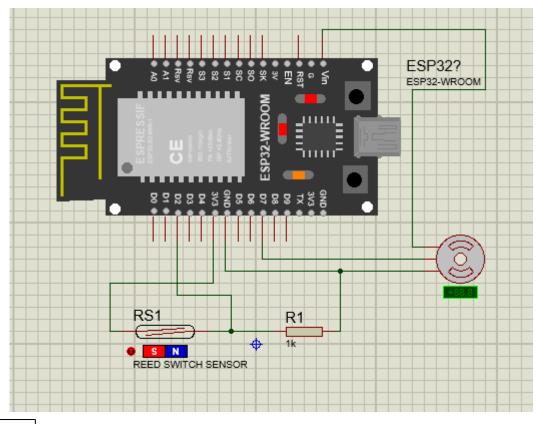
Dimensionnement et choix du batterie.





### 🔪 Calculer la capacité de la batterie:

Composant	Courant maximal consommé	Tension necessaire
ESP32	170mA	5V
Capteur	3.3mA	3.3V
Servo moteur	500mA	5V
Totalité:	673.3.3mA	5V



$$E_{tot} = I \times V \times t = 30.3Wh$$

$$E_{tot} = \frac{E_{tot}}{\eta \times DoD \times V} = \frac{30.3}{0.73 * 0.85 * 5} = 9766mAh$$

Figure 13: Montage du système dans ISIS



### > Choix de la batterie:

#### **Utilisation d'un Power Bank est un bon choix**:

- Disponible dans le marché
- Facilement rechargeables
- Compact et léger
- Haute capacité

#### **KONFULON Power Bank:**

• Type: Li-polymer

Tension de sortie: 5V

Capacité: 10000mAh

Capacité nette: 7300mAh

Profondeur de décharge: 80%

• Temps de charge: 3h

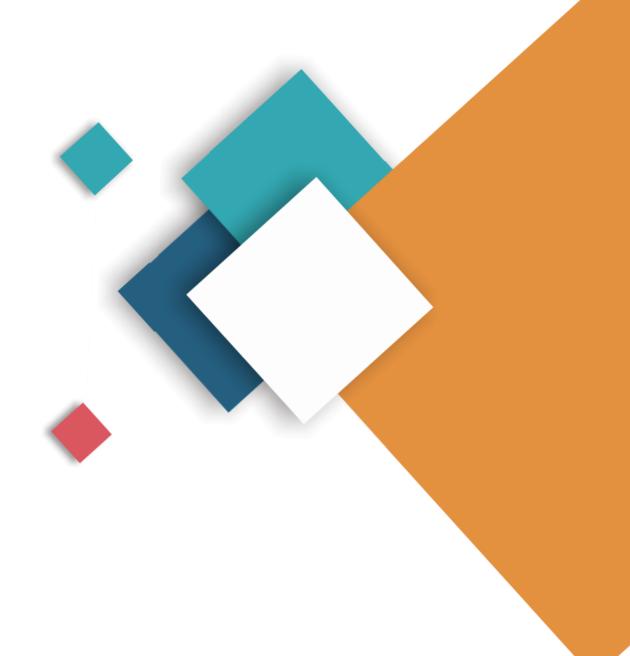


#### **Avantage de Li-polymer batterie**:

- Poids léger
- Sécurité améliorée
- Longue durée de vie
- Alimentation stable et fiable

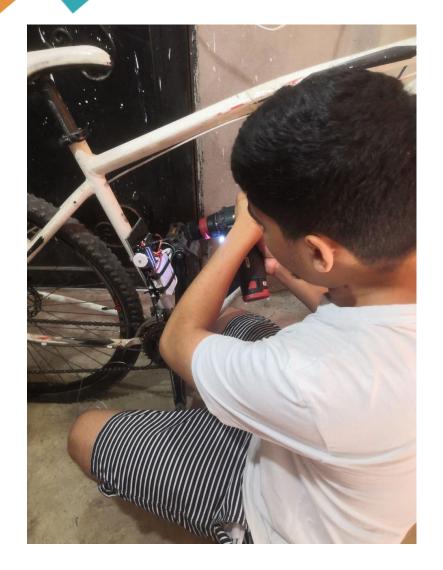
# Prototype

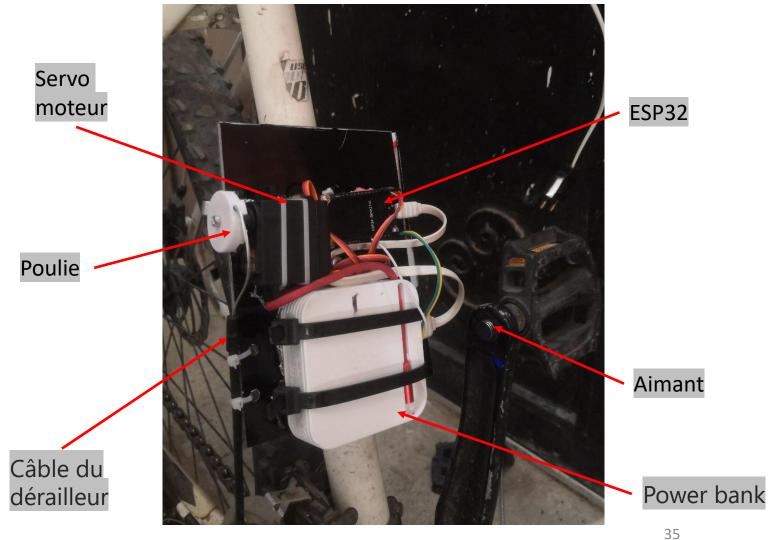






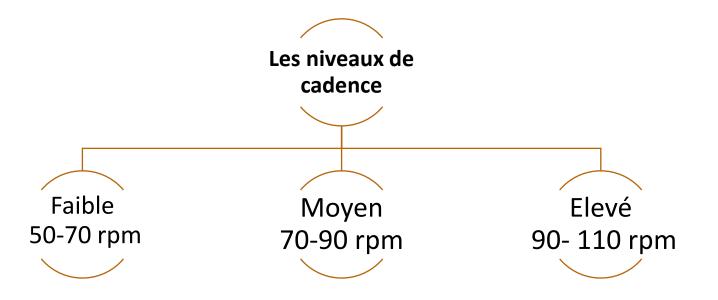
# Réalisation du système:

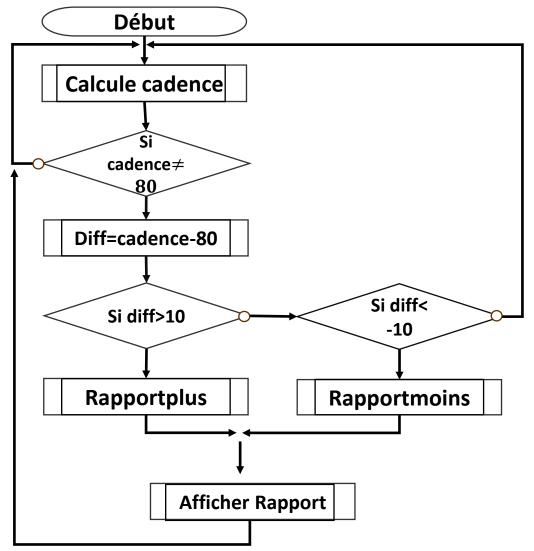






# Organigramme

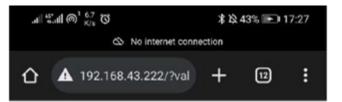






# Essai d'experience:





#### **ESP32 Cadence Control**

Current Cadence: 60 RPM

Current Gear: 6

Low Cadence Medium Cadence High Cadence

Set Target Cadence

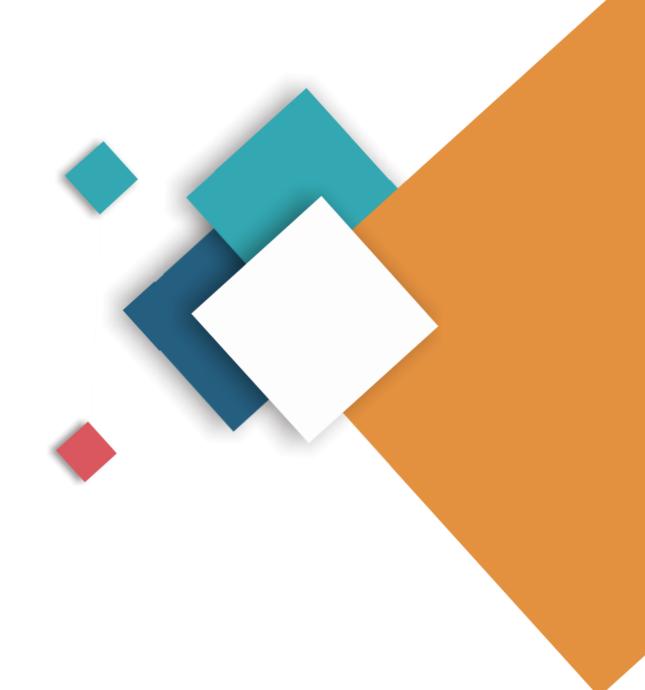






# Annexe





#### **Code ESP32 qui traduit l'organigramme:**

```
#include <ESP32Servo.h>
     #include <WiFi.h>
     #include <WebServer.h>
     const char* ssid = "salmanewifi";
     const char* password = "12345678";
     const int reedPin = 5;
     const int servoPin = 13;
     volatile int pedalRevolutions = 0;
     unsigned long lastTime = 0;
     unsigned long debounceTime = 100;
     unsigned long lastReedSwitchTime = 0;
     int targetCadence = 80;
     float gearAngles[] = {97.4, 83.26, 69.51, 55.76, 41.63, 27.88, 14.13, 0};
     int currentGear = 0;
     int numGears = sizeof(gearAngles) / sizeof(gearAngles[0]);
17
     Servo myServo;
     WebServer server(80);
     void setup() {
       pinMode(reedPin, INPUT PULLUP);
       attachInterrupt(digitalPinToInterrupt(reedPin), countPedalRevolution, FALLING);
       Serial.begin(9600);
       myServo.attach(servoPin);
       WiFi.begin(ssid, password);
       while (WiFi.status() != WL CONNECTED) {
         delay(1000);
         Serial.println("Connecting to WiFi...");
```

```
Serial.println(WiFi.localIP());
34
       server.on("/", HTTP GET, handleRoot);
       server.on("/setTargetCadence", HTTP GET, handleSetTargetCadence);
       server.on("/getCadence", HTTP GET, handleGetCadence);
       server.on("/getGear", HTTP_GET, handleGetGear);
       server.on("/setLowCadence", HTTP_GET, []() { handleSetPresetCadence(60); });
       server.on("/setMediumCadence", HTTP_GET, []() { handleSetPresetCadence(80); });
       server.on("/setHighCadence", HTTP_GET, []() { handleSetPresetCadence(100); });
       server.begin();
     void loop() {
         unsigned long currentTime = millis();
         if (currentTime - lastTime >= 1000) {
             detachInterrupt(digitalPinToInterrupt(reedPin));
             int cadence = pedalRevolutions * 60;
             Serial.print("Cadence: ");
             Serial.println(cadence);
             int diff = cadence - targetCadence;
             if (diff > 10) {
                 increaseGear();
             } else if (diff < -10) {
                 decreaseGear();
             pedalRevolutions = 0;
             lastTime = currentTime:
             attachInterrupt(digitalPinToInterrupt(reedPin), countPedalRevolution, FALLING);
         server.handleClient();
```

```
html += "Current Gear: <span id='currentGear'>0</span>";
void countPedalRevolution() {
                                                                                                   html += "<button onclick=\"setLowCadence()\">Low Cadence</button>";
    if ((millis() - lastReedSwitchTime) > debounceTime) {
                                                                                                   html += "<button onclick=\"setMediumCadence()\">Medium Cadence</button>";
         pedalRevolutions++;
                                                                                                   html += "<button onclick=\"setHighCadence()\">High Cadence</button><br>>';
         lastReedSwitchTime = millis();
                                                                                                   html += "<form id=\"targetCadenceForm\">";
                                                                                                   html += "<input type=\"number\" name=\"value\" value=\"" + String(targetCadence) + "\">";
                                                                                                   html += "<input type=\"button\" value=\"Set Target Cadence\" onclick=\"setTargetCadence()\">";
void increaseGear() {
                                                                                                   html += "</form>";
    if (currentGear < numGears) {</pre>
                                                                                                   html += "<script>";
                                                                                                   html += "function setTargetCadence() {";
         currentGear++;
                                                                                                   html += " var value = document.querySelector('input[name=\"value\"]').value;";
         myServo.write(gearAngles[currentGear - 1]);
                                                                                                   html += " fetch('/setTargetCadence?value=' + value).then(response => response.text()).then(data => {";
        Serial.print("Increasing Gear to: ");
                                                                                                   html += " console.log('Target Cadence Updated:', data);";
        Serial.println(currentGear);
        Serial.print("Servo Angle: ");
                                                                                                   html += " });";
                                                                                          113
        Serial.println(gearAngles[currentGear - 1]);
                                                                                                   html += "}";
                                                                                                   html += "function setLowCadence() {";
                                                                                                   html += " fetch('/setLowCadence').then(response => response.text()).then(data => {";
                                                                                                   html += " console.log('Low Cadence Set:', data);";
void decreaseGear() {
                                                                                          117
                                                                                                   html += " });";
    if (currentGear > 1) {
                                                                                                   html += "}";
        currentGear--;
                                                                                                   html += "function setMediumCadence() {";
        myServo.write(gearAngles[currentGear - 1]);
                                                                                                   html += " fetch('/setMediumCadence').then(response => response.text()).then(data => {";
        Serial.print("Decreasing Gear to: ");
                                                                                                   html += " console.log('Medium Cadence Set:', data);";
        Serial.println(currentGear);
                                                                                                   html += " });";
        Serial.print("Servo Angle: ");
                                                                                                   html += "}";
         Serial.println(gearAngles[currentGear - 1]);
                                                                                                   html += "function setHighCadence() {";
                                                                                                   html += " fetch('/setHighCadence').then(response => response.text()).then(data => {";
                                                                                                   html += " console.log('High Cadence Set:', data);";
                                                                                                   html += " });";
void handleRoot() {
                                                                                                   html += "}";
    String html = "<html><body>";
                                                                                                   html += "setInterval(function() {";
    html += "<h2>ESP32 Cadence Control</h2>";
                                                                                                   html += " fetch('/getCadence').then(response => response.text()).then(data => {";
    html += "Current Cadence: <span id='currentCadence'>0</span> RPM";
                                                                                                   html += "
                                                                                                               document.getElementById('currentCadence').innerText = data;";
```

```
html += " });";
         html += " fetch('/getGear').then(response => response.text()).then(data => {";
                      document.getElementById('currentGear').innerText = data;";
         html += " });";
         html += "}, 1000);";
         html += "</script>";
         html += "</body></html>";
         server.send(200, "text/html", html);
      void handleSetTargetCadence() {
         if (server.hasArg("value")) {
             targetCadence = server.arg("value").toInt();
             Serial.print("Target Cadence Set to: ");
             Serial.println(targetCadence);
          server.send(200, "text/plain", "Target Cadence Updated");
      void handleGetCadence() {
          int currentCadence = pedalRevolutions * 60;
         server.send(200, "text/plain", String(currentCadence));
     void handleGetGear() {
          server.send(200, "text/plain", String(currentGear));
     void handleSetPresetCadence(int cadence) {
         targetCadence = cadence;
         Serial.print("Target Cadence Set to: ");
         Serial.println(targetCadence);
         server.send(200, "text/plain", "Target Cadence Updated");
161 }
```

#### **Programe en Python**: Traçage du courbe B(R):

```
import numpy as np
import matplotlib.pyplot as plt
Br = 1.24
x = 0.015
e = 0.01
R values = np.linspace(0.002, 0.012, 500)
B = (Br / 2) * ((e + x) / np.sqrt(R_values**2 + (e + x)**2) - x / np.sqrt(R_values**2 + x**2))
B target = 0.04
R_target = np.interp(B_target, B, R_values)
plt.figure(figsize=(10, 6))
plt.plot(R_values, B, label='B(R)')
plt.axhline(y=B target, color='r', linestyle='--', label=f'B = {B target}')
plt.axvline(x=R target, color='g', linestyle='--', label=f'R = {R_target:.4f}')
plt.legend()
plt.grid(True)
plt.show()
```