

Dary Jean-Léo n°26212

AutoShoes : Les chaussures AutoLaçantes

Une nouvelle façon
de se déplacer

Plan



Structure Globale

Comment mettre en lien les différentes parties de notre système ?



Interface Utilisateurs

Comment l'utilisateur pourra t'il se (dé)lacer ?
présence ou non d'une communication sans fil ?



Capteurs

Par quel dispositif la chaussure évaluera t'elle du bon maintien du pied, quelle sera sa fiabilité ?



Système de serrage

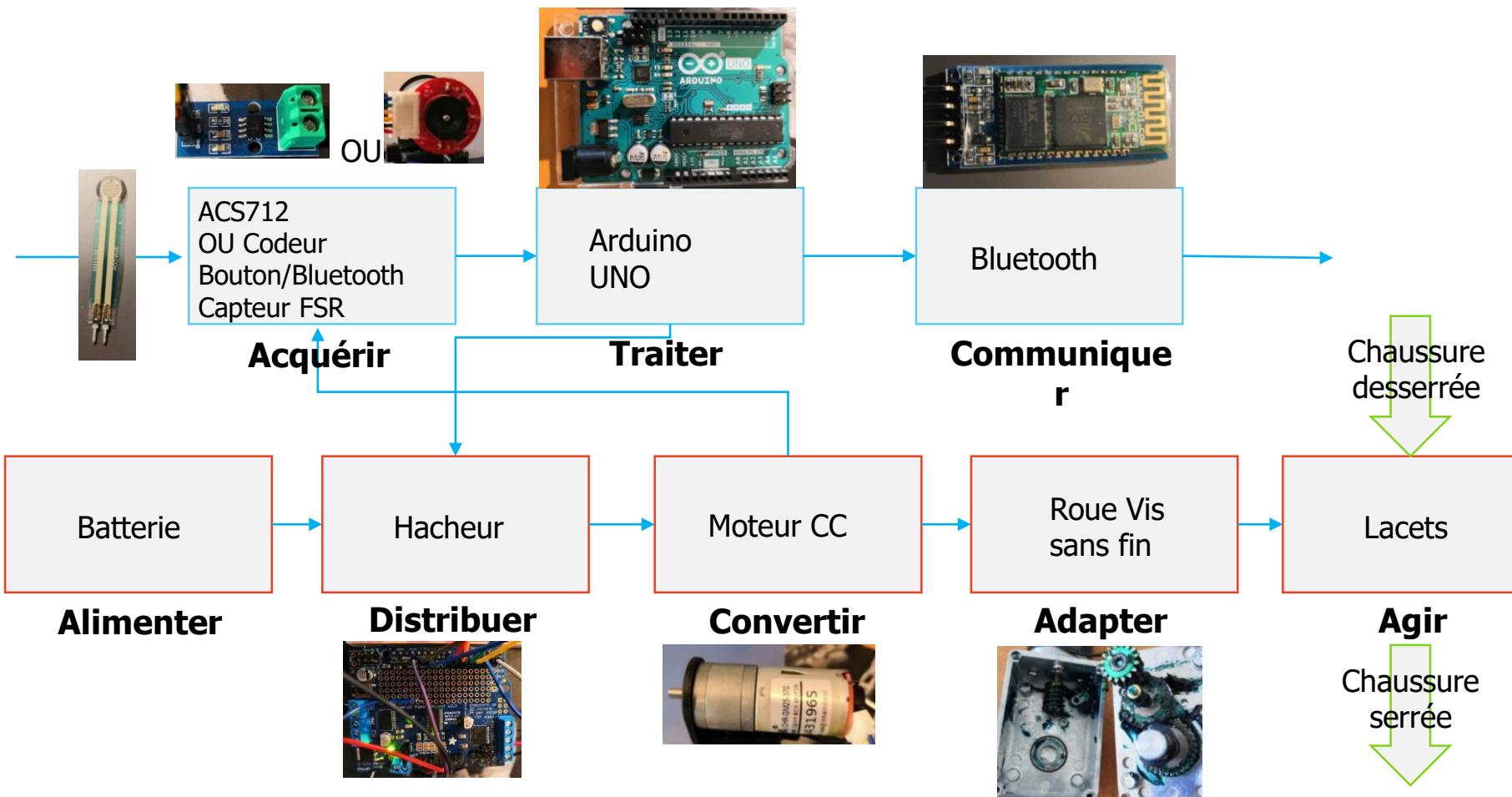
Comment le système assurera le maintien du pied ?



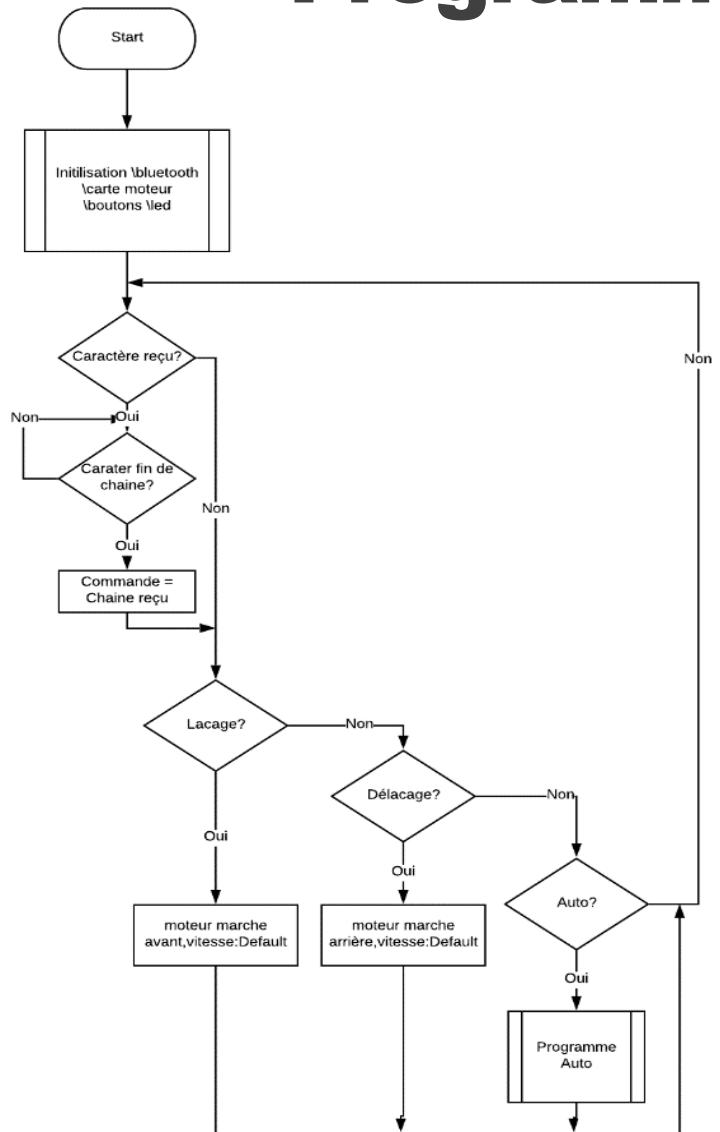
Structure Globale



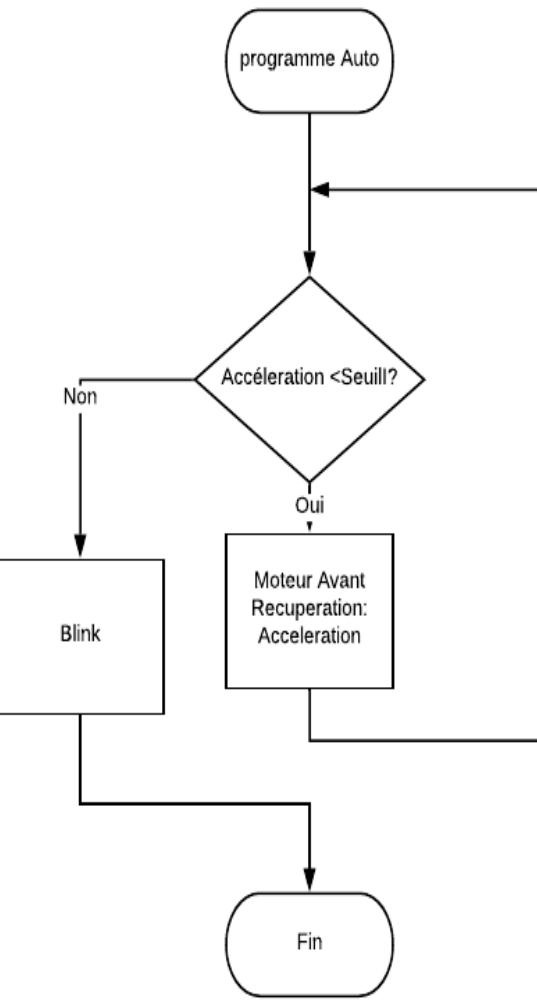
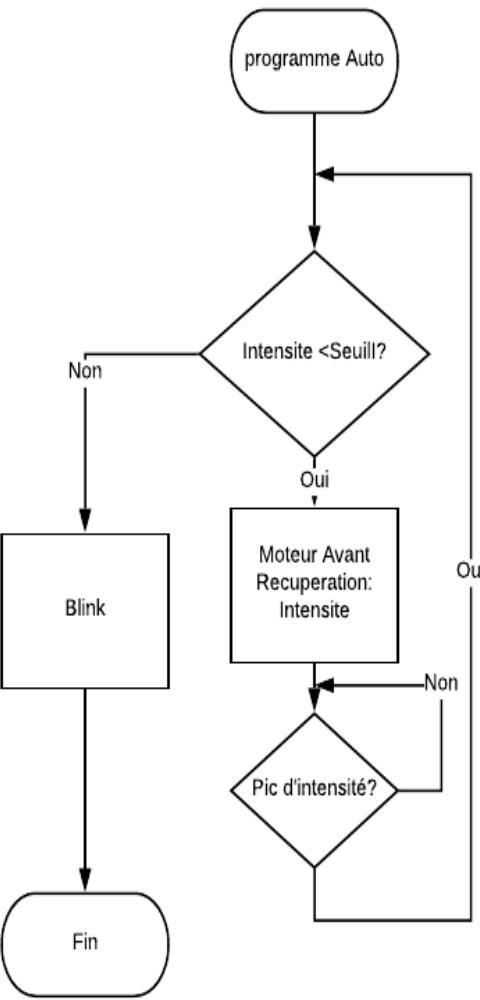
Structure globale



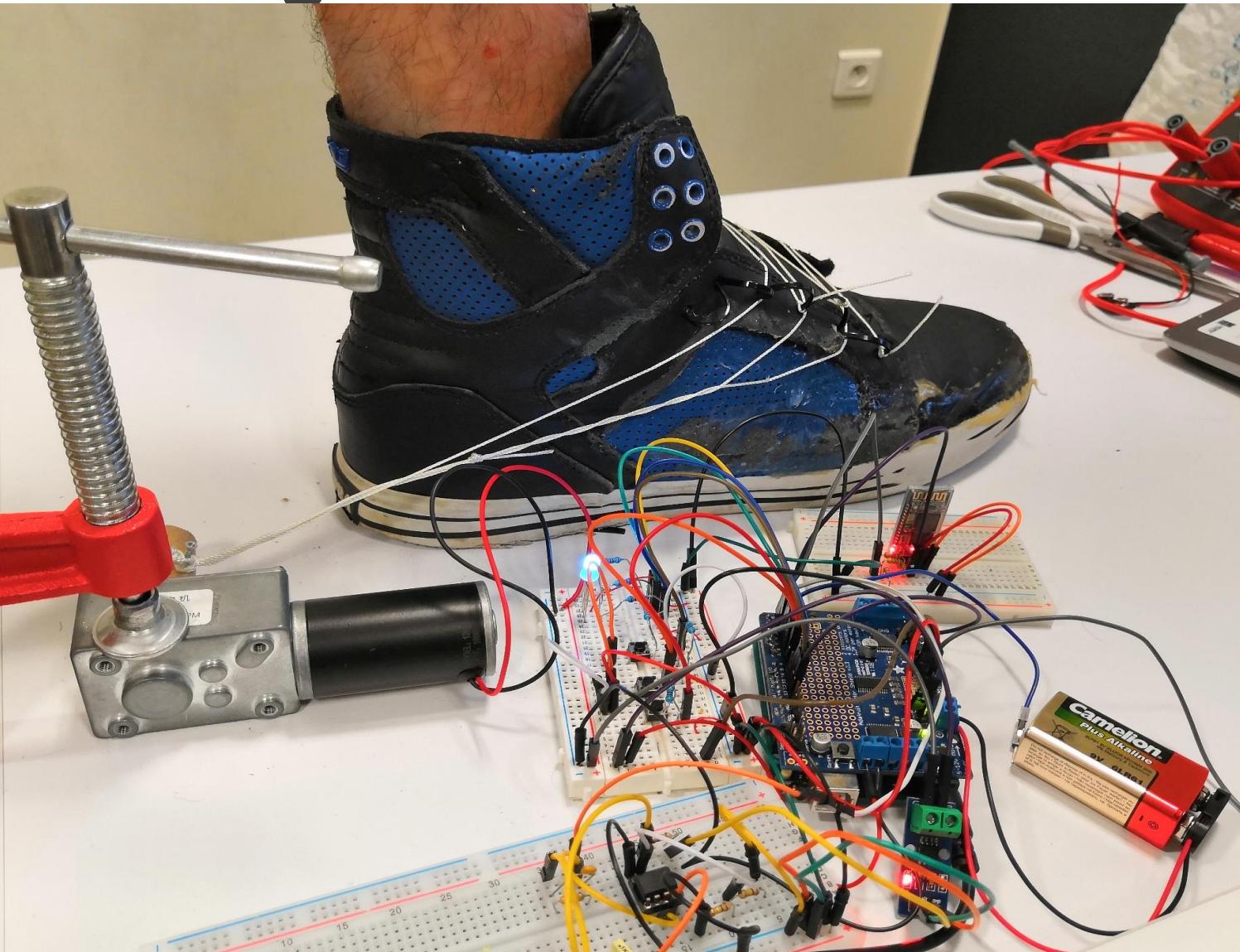
Programmation



2 Possibilités



Montage





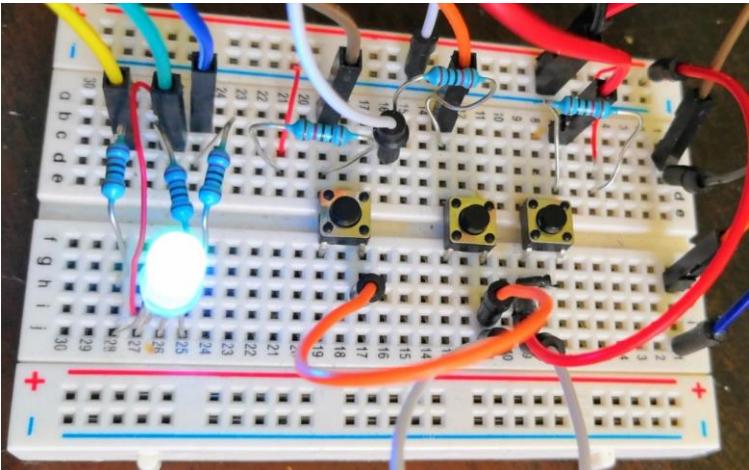
Interface utilisateur



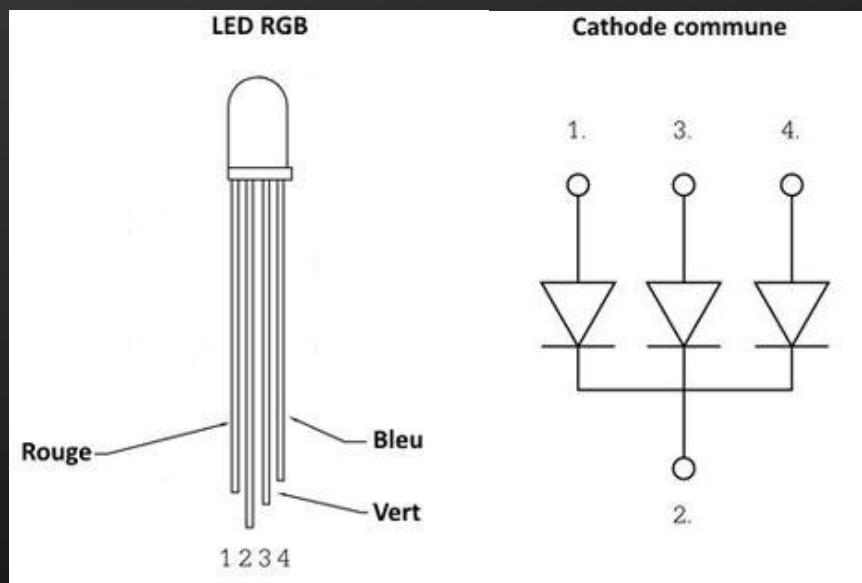
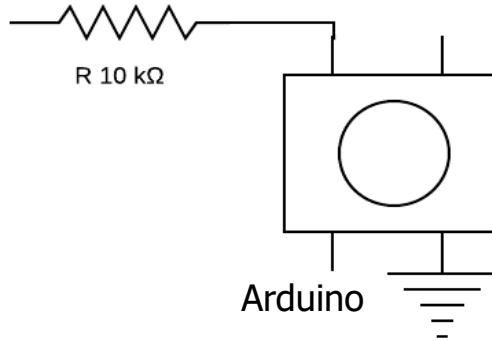
Boutons « physiques »

Trois boutons sont présents sur la chaussure:

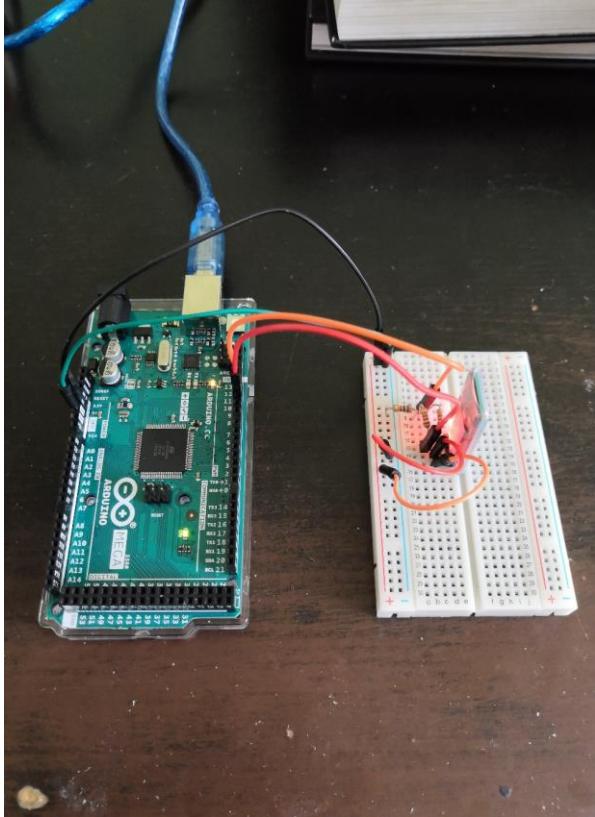
- Bouton de laçage manuel (+)
- Bouton de délaçage manuel (-)
- Bouton automatique (Auto)



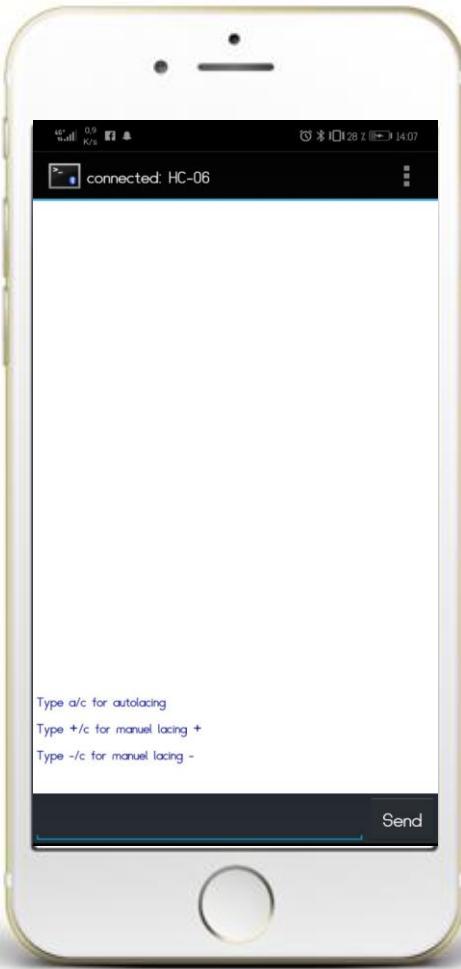
Résistance « Pull Up »



Bluetooth



Montage HC06





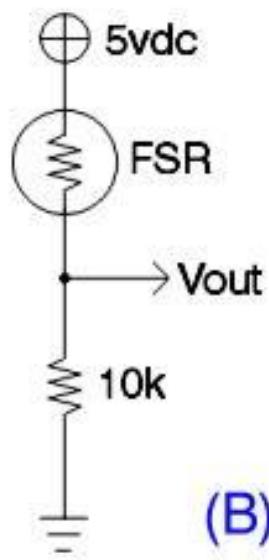
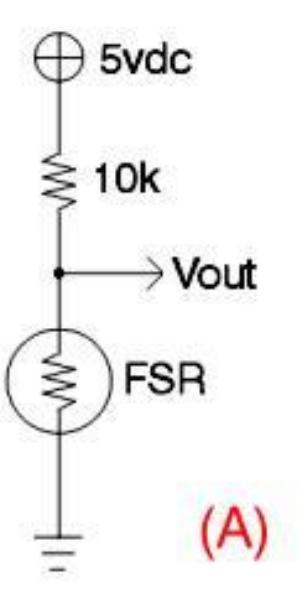
Capteurs

Capteur de Force



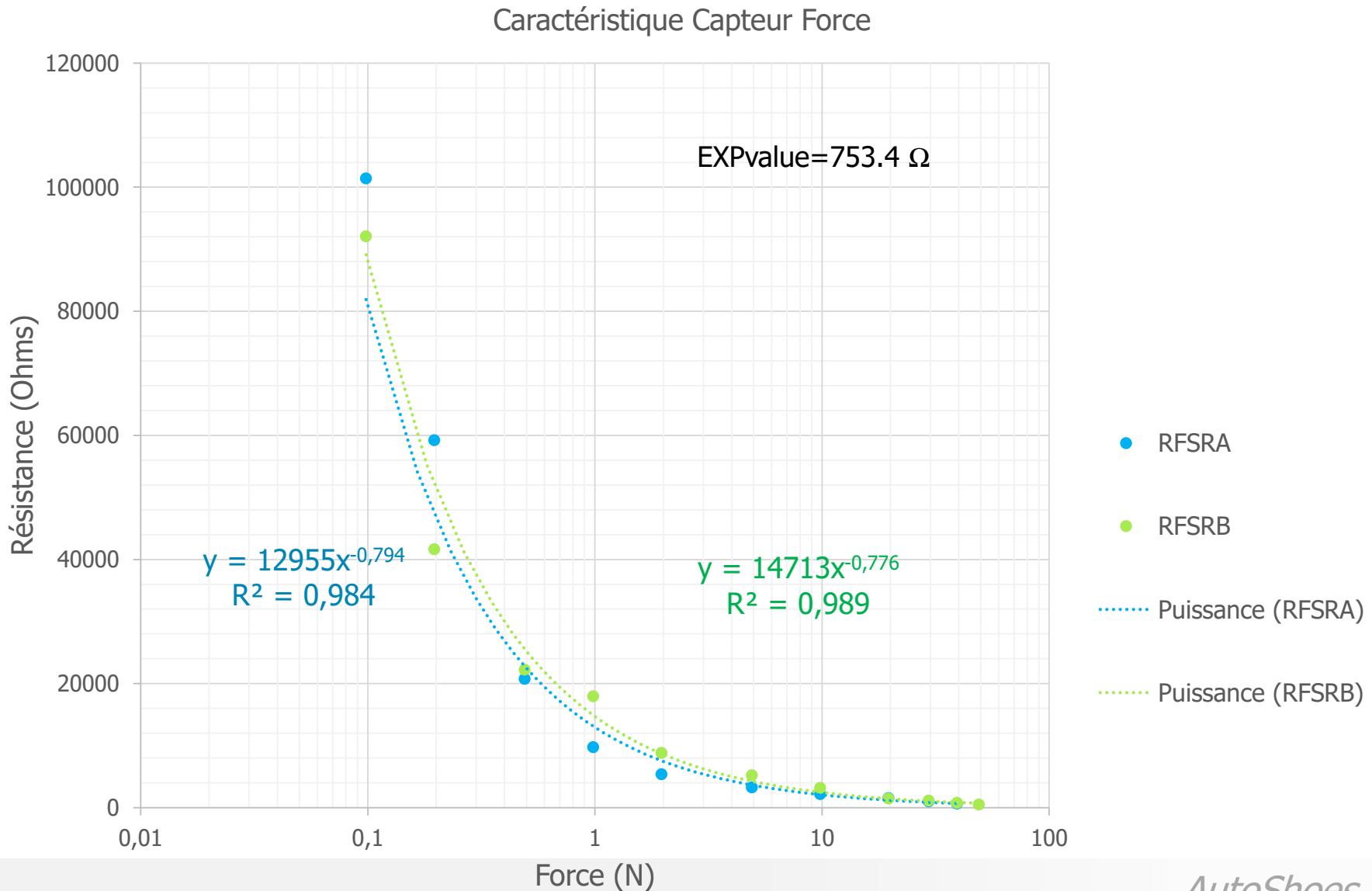
- Permet de détecter le pied de l'utilisateur

Mesure de la relation entre le poids et le voltage de sortie du capteur



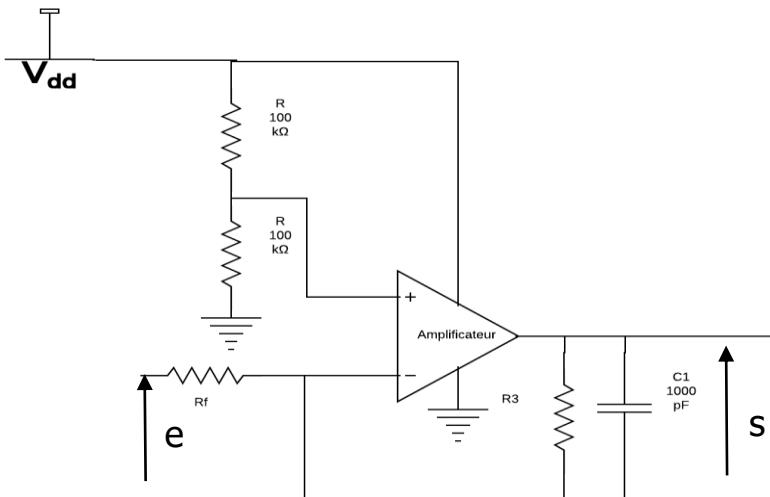
Montage « Poutre pour capteur de force »

Caractéristique Capteur de Force



Capteur de Courant

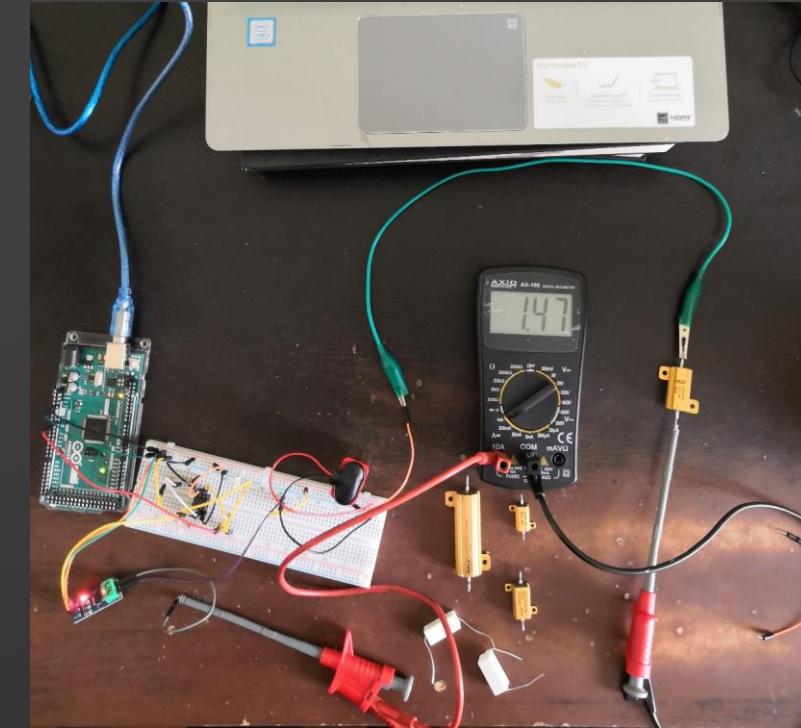
- Permet de détecter le couple résistant



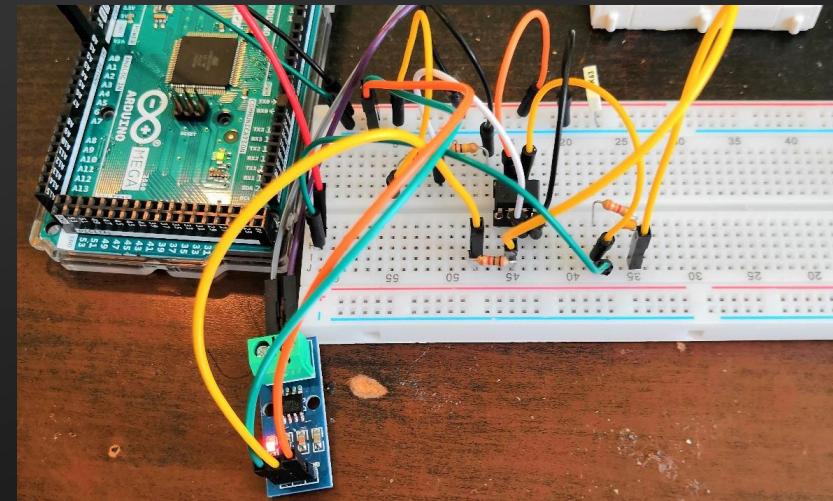
$$H(j\omega) = -\frac{R_3}{R_f} \frac{1}{(1 + j\omega R_3 C_1)}$$

$$H_0 = -\frac{R_3}{R_f}$$

$$f_0 = \frac{1}{2\pi R_3 C_1} = 43\text{kHz}$$



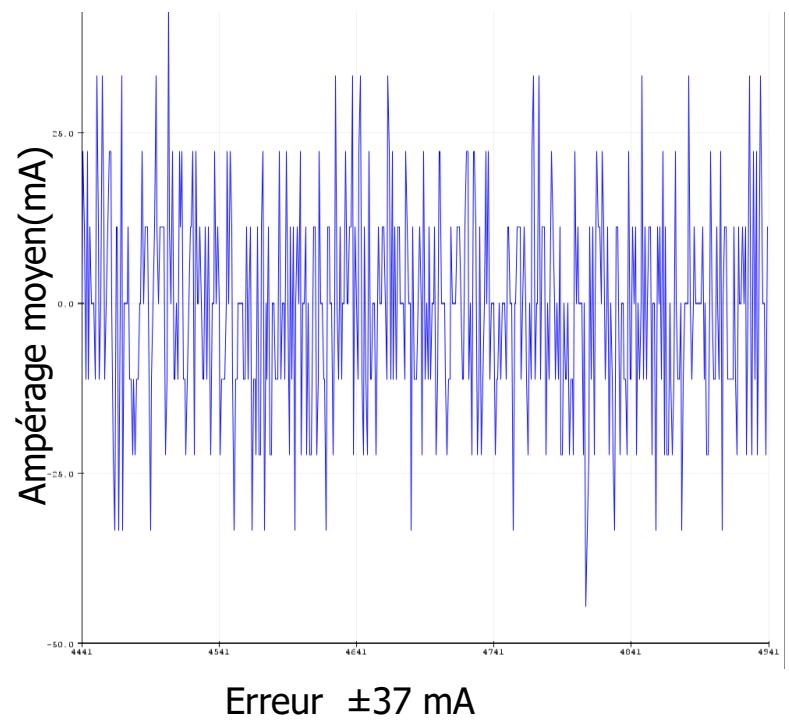
Montage « Résistance de Puissance »



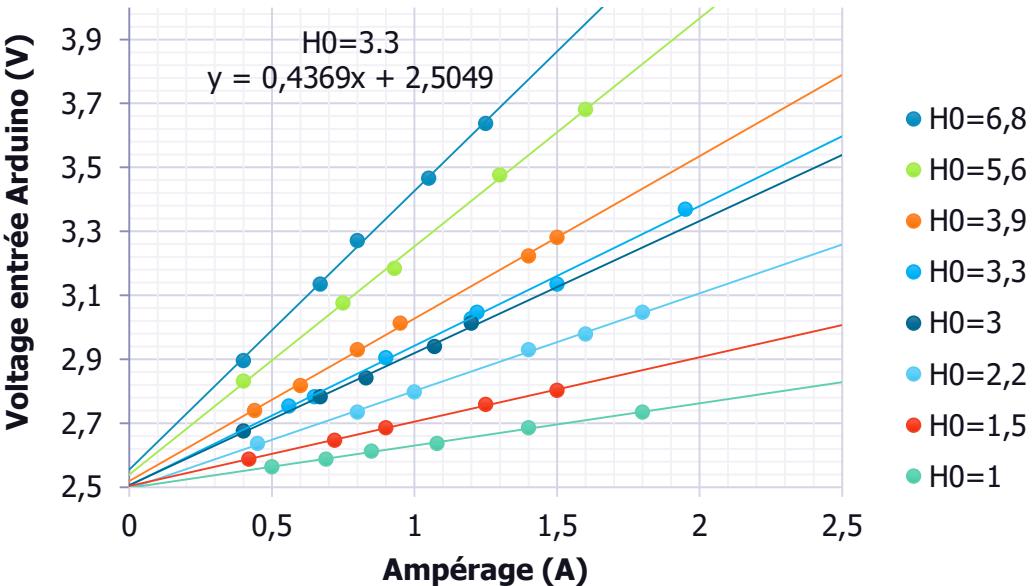
ACS712+Filtre Amplificateur

Caractéristique et Erreur

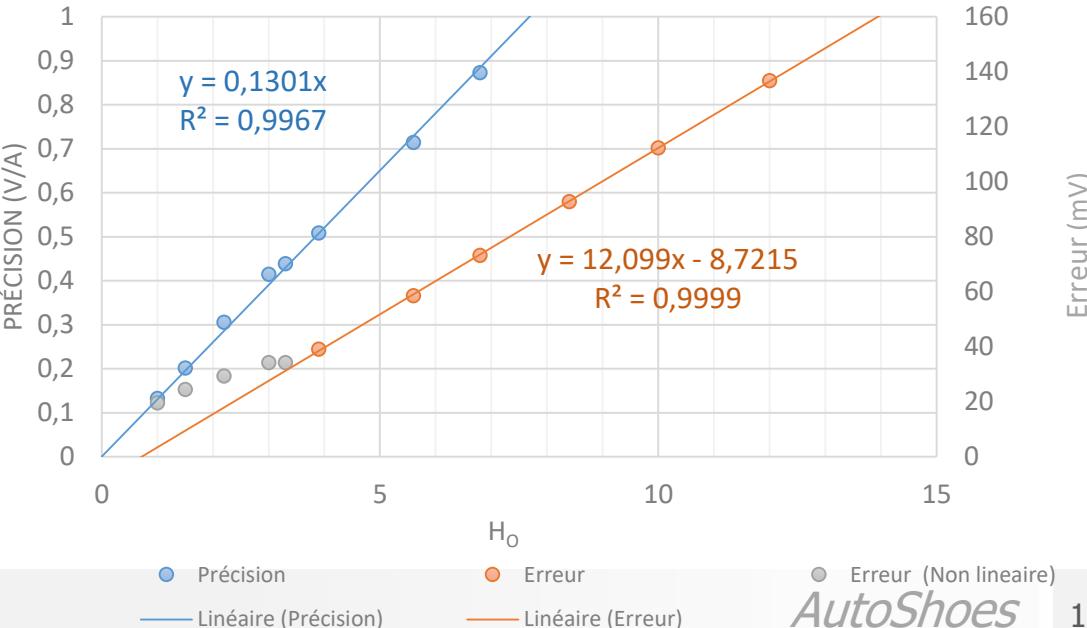
Précision: 437 mV/A



Caractéristique ACS712 pour plusieurs H_o

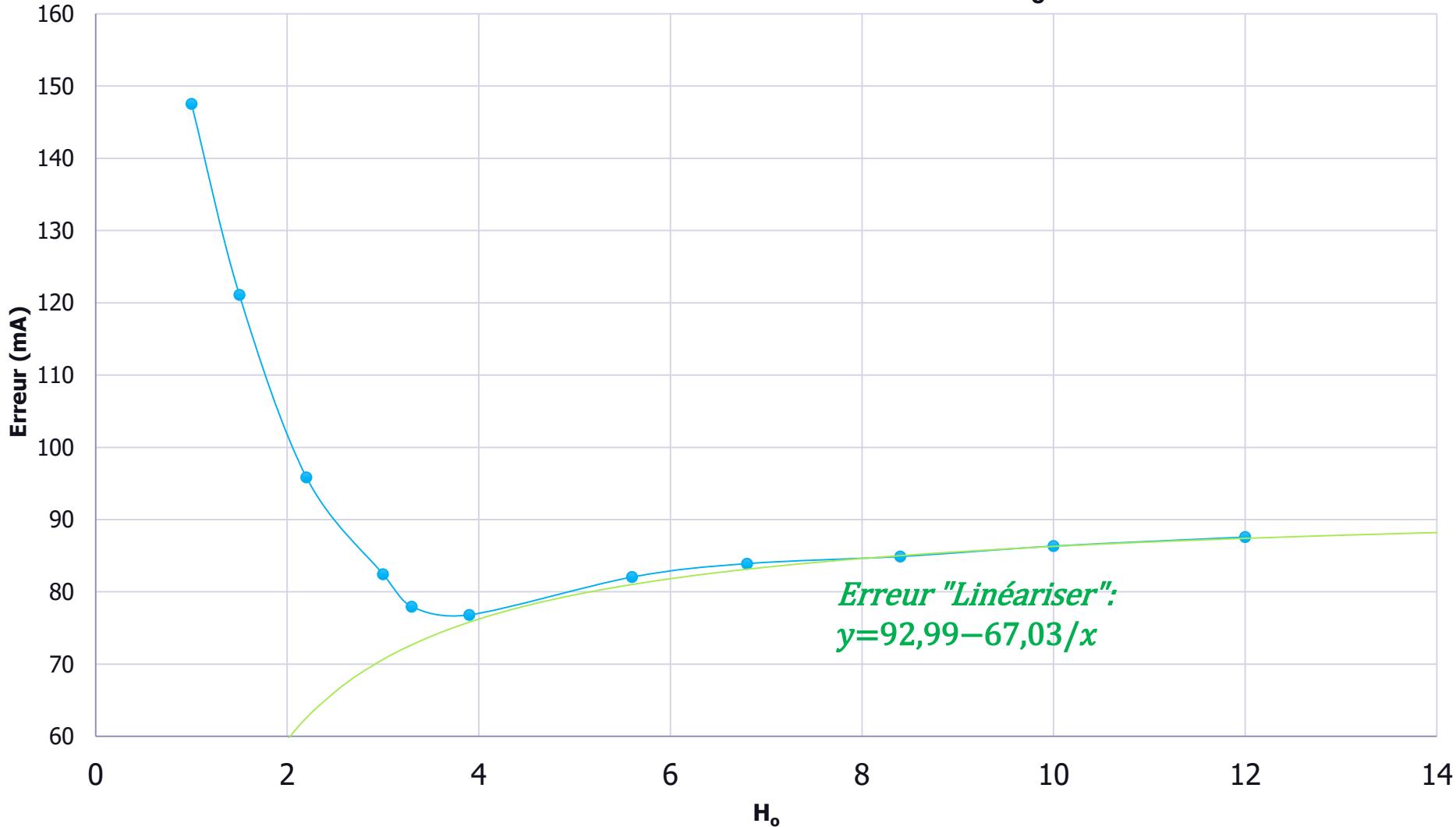


Précision et erreur de quantification en fonction de H_o



Erreurs Capteur de Courant

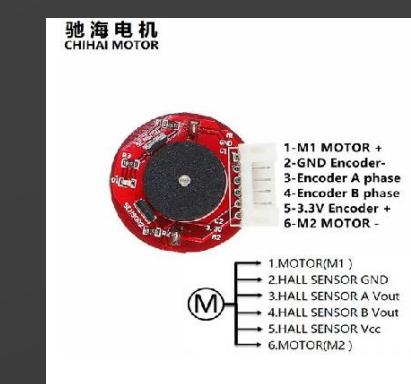
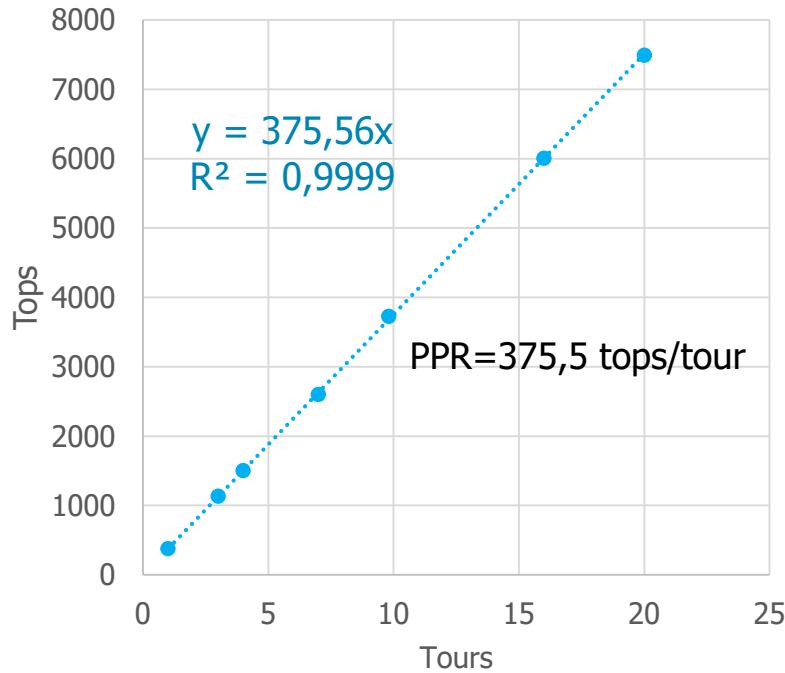
Erreurs ACS712 en fonction de H_o



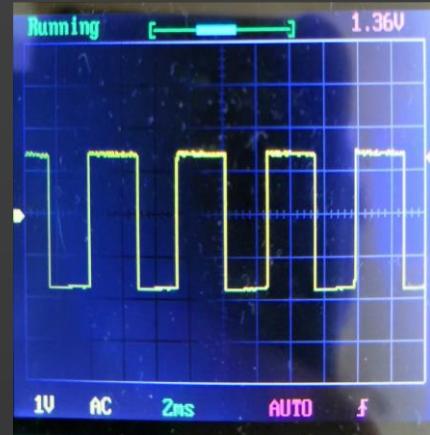
Capteur de vitesse

- Déetecter une décélération

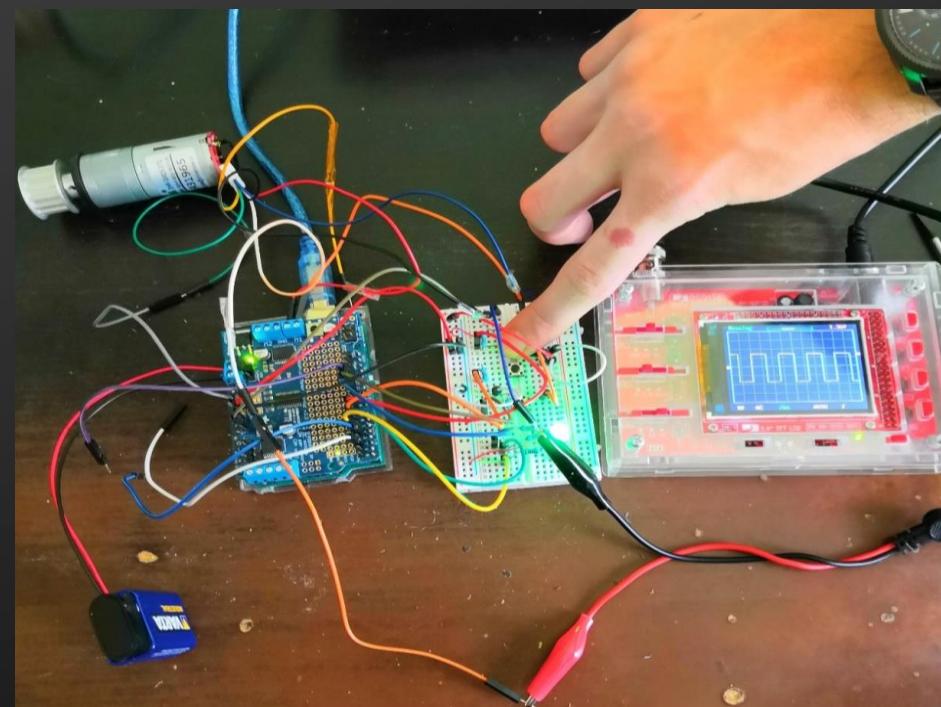
Caractéristique Codeur



Codeur à effet Hall

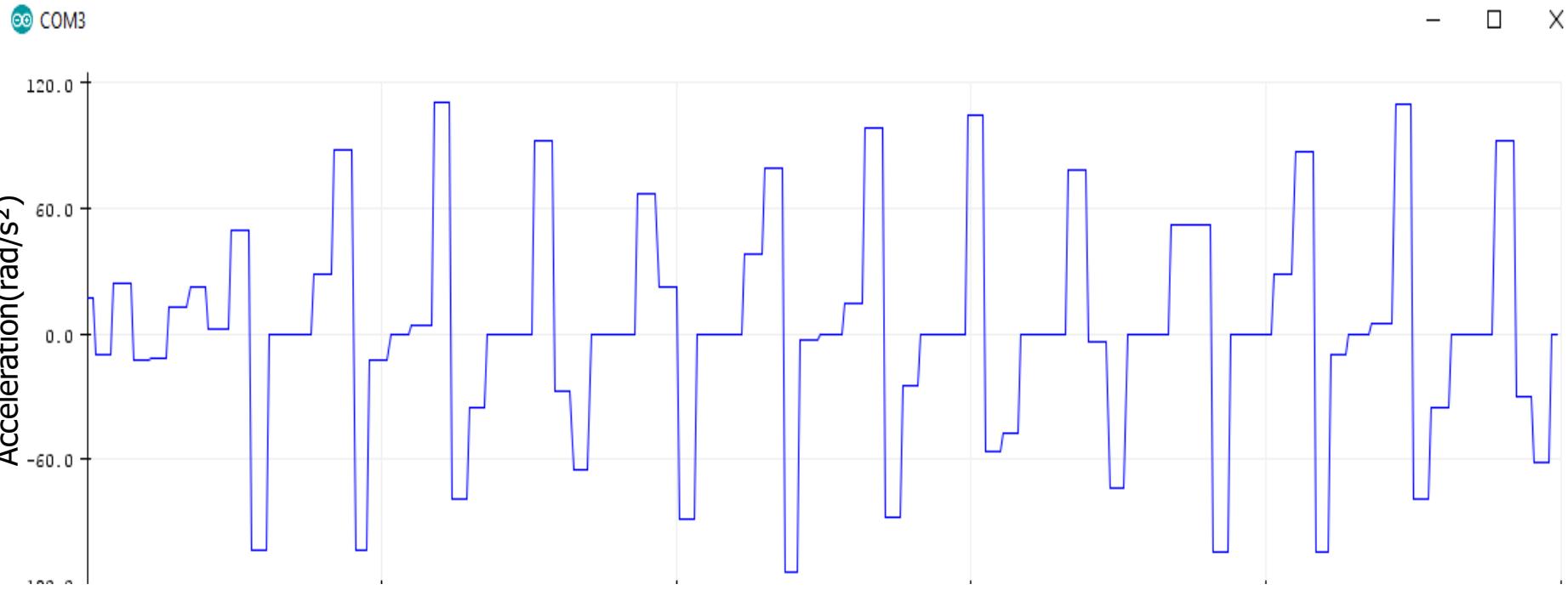


Signal créneaux du capteur



Montage codeur incrémental

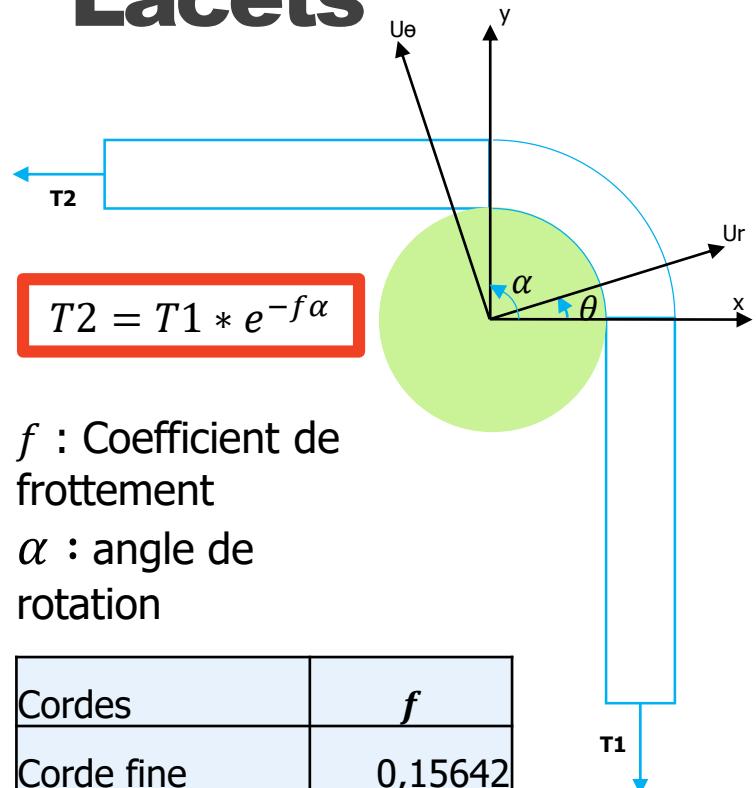
Problème





Serrage des lacets

Lacets



f : Coefficient de frottement
 α : angle de rotation

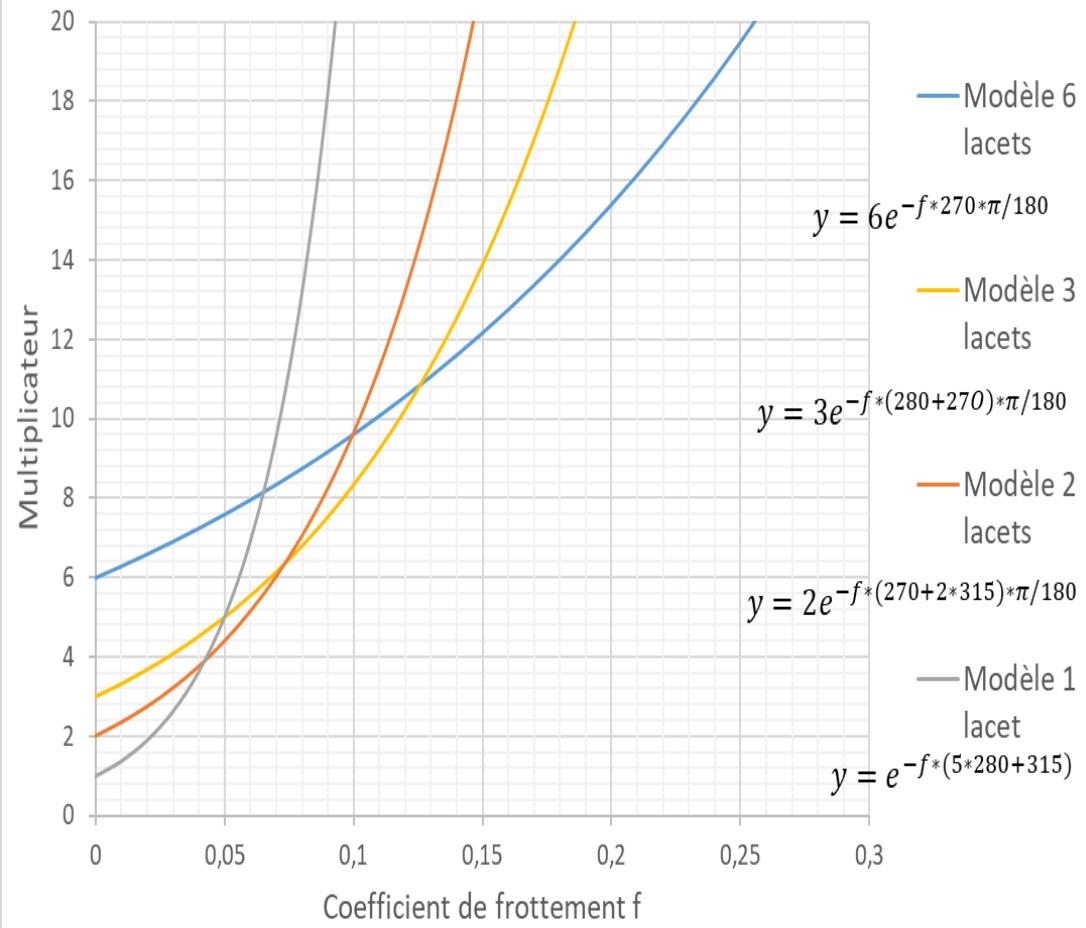
Cordes	f
Corde fine	0,15642
Lacet Noir	0,27977
Corde fine Rouge	0,29377



Montage mesure du frottement

Modèles de laçage

Evolution de l'amplification de la force en fonction du coefficient f

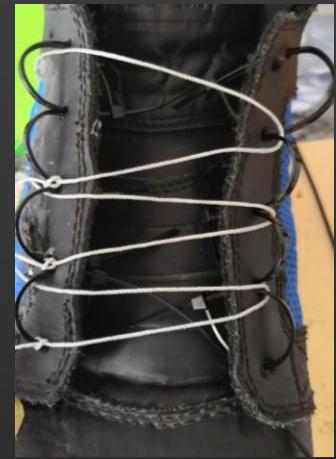


Angle ouverture $\approx 270^\circ$

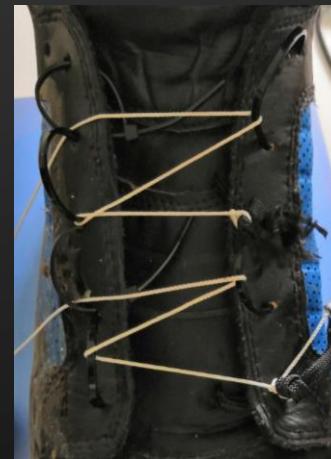


Modèle 6 Lacets

Angle lacet $\approx 315^\circ$



Modèle 3 Lacets

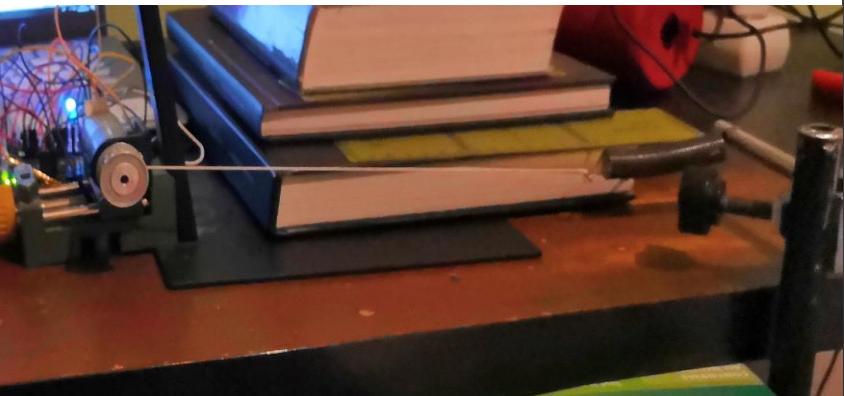


Modèle 2 Lacets



Modèle 1 Lacet

Montages

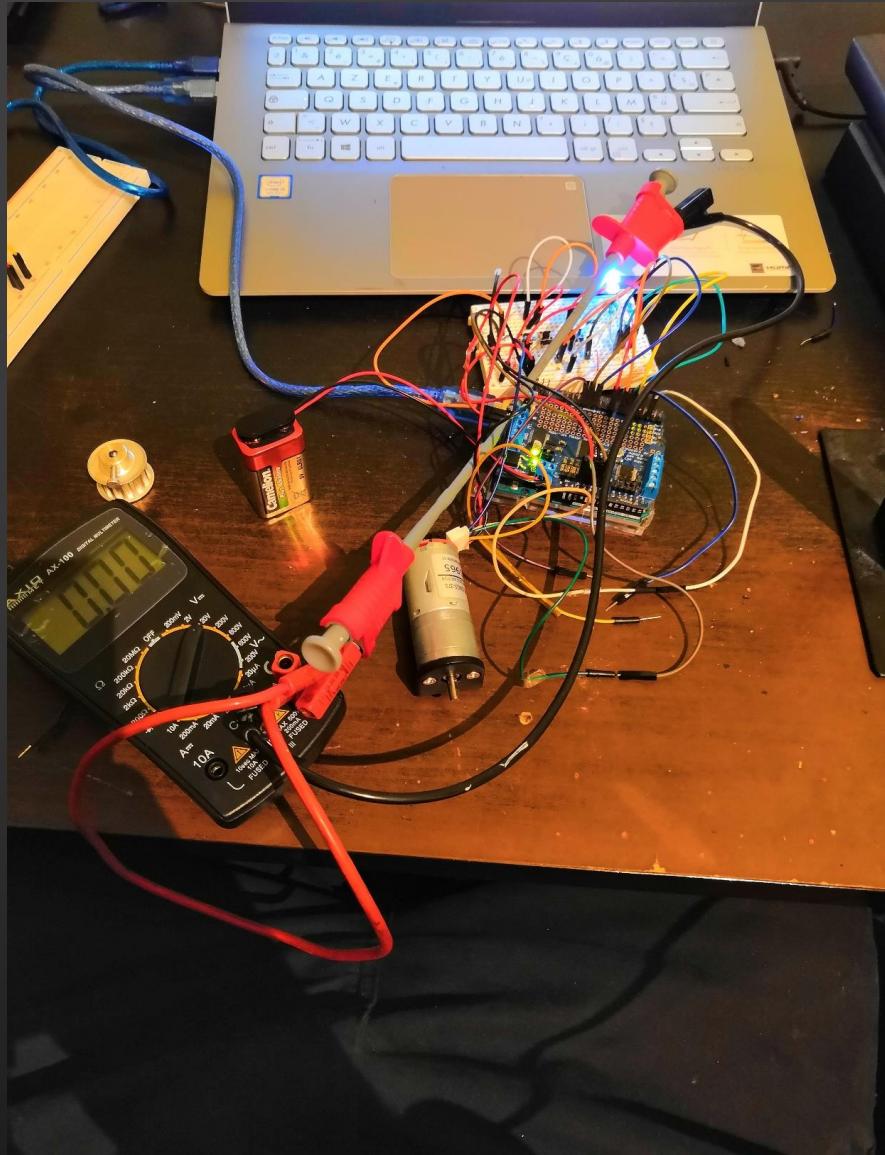


Montage « Couple grâce à un ressort »

Loi de Hooke:

$$F = k\Delta x$$

$$C = F * R = k\Delta x R$$

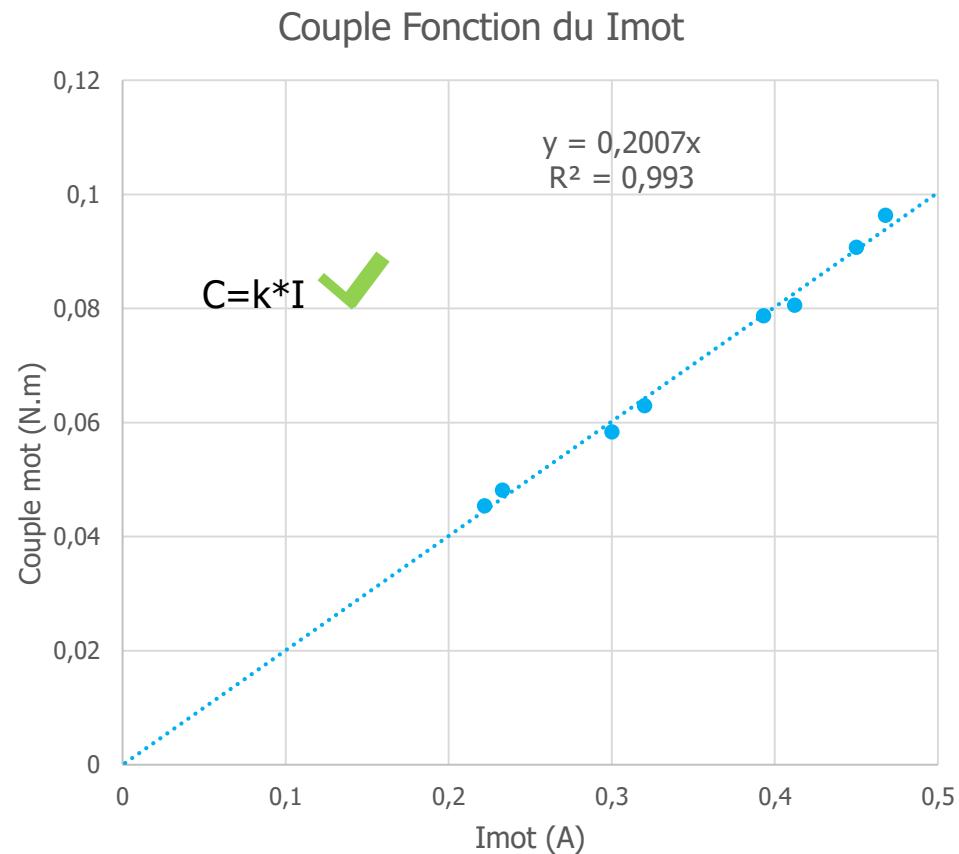
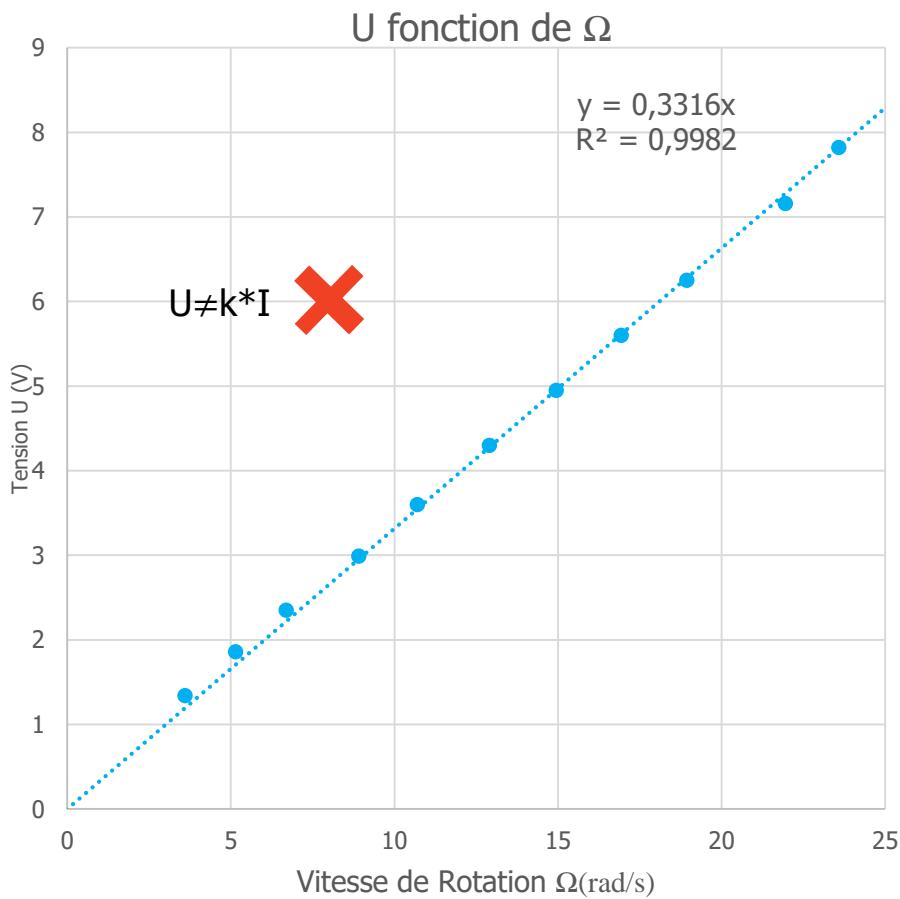


Montage « Tension , codeur incrémental »

Caractéristique MCC



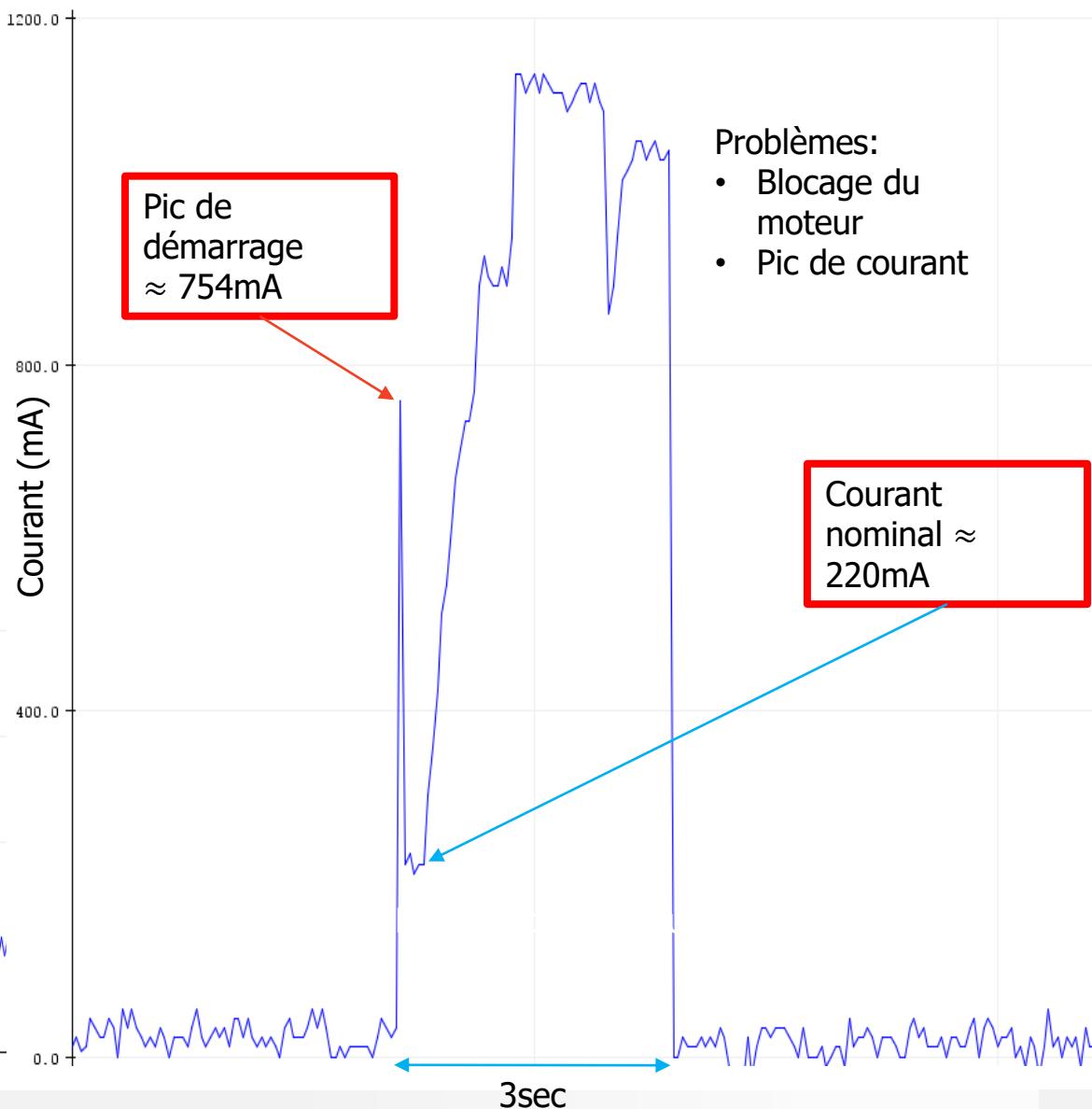
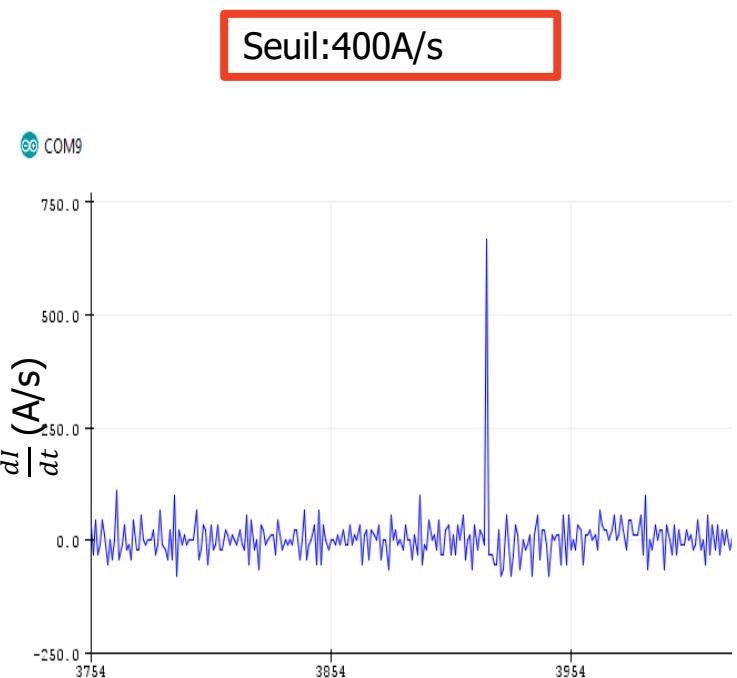
Moteur



Résultats

$F \approx 120N$ Pour $C=0.18N.m$

Courant de coupure $\approx 900mA$



Système bloquant

Rendement $\eta=79\%$

Rapport de réduction $r=35$

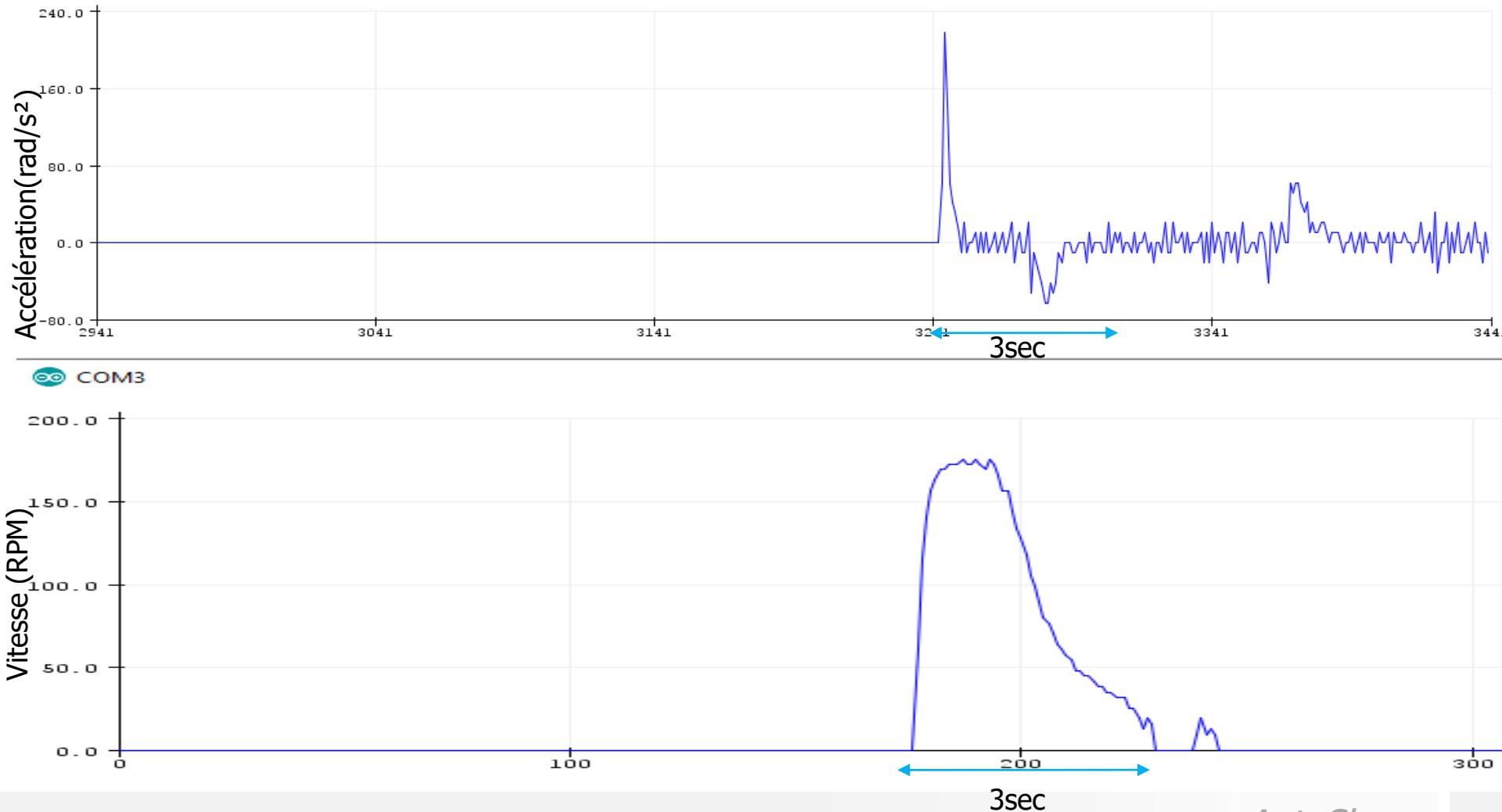


Roue vis sans fin graissé

Résultats

$T_e = 0,05 \text{ sec}$

Accélération de coupure $\approx -20 \text{ rads/s}^2$





Dary Jean-Léo n°26212

Merci de
votre
attention

Annexe 1

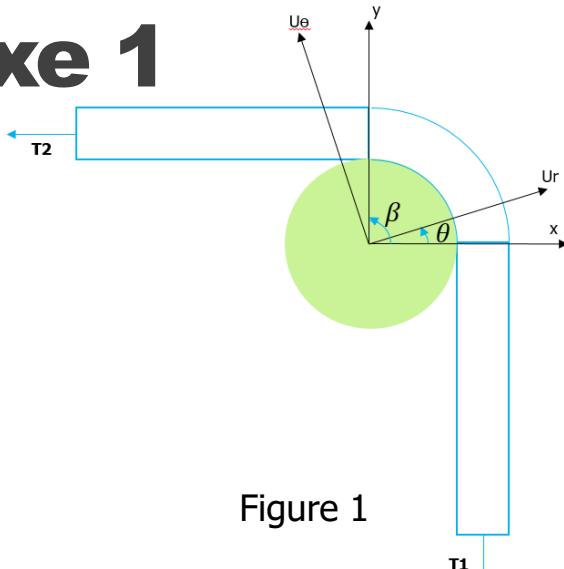


Figure 1

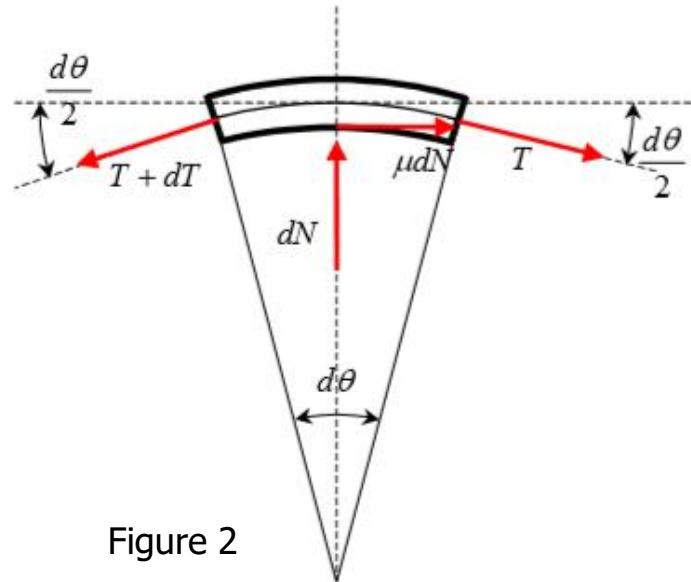


Figure 2

A partir du diagramme du corps libre d'un élément de courroie de longueur $rd\theta$ représenté à la figure 2, écrivons les équations d'équilibre suivant la tangente et la normale à l'élément de courroie :

$$\begin{cases} T \cos \frac{d\theta}{2} + \mu dN = (T + dT) \cos \frac{d\theta}{2} \\ dN = (T + dT) \sin \frac{d\theta}{2} + T \sin \frac{d\theta}{2} \end{cases}$$

Etant donné qu'à la limite $\cos \frac{d\theta}{2}$ tend vers 1 et $\sin \frac{d\theta}{2}$ tend vers $\frac{d\theta}{2}$, et que le produit de deux différentielles peut être négligé en regard du terme du premier ordre:

$$\begin{cases} \mu dN = dT \\ dN = T d\theta \end{cases} \quad \frac{dT}{T} = \mu d\theta \quad \boxed{T_2 = T_1 e^{\mu \beta}}$$

Annexe 2

```
/*
AutoShoes program on arduino uno

*/
//--- Timer --
#include <TimerOne.h>
#define Te 40000

//--- Capteur vitesse --
volatile long int count      = 0;
volatile double vitesse = 0;
volatile double acceleration=0;
#define interrupt 0
#define PPR 341.2
#define PPR 375.56

//--- Moteur CC --
/*
Modèle fait selon la bibliothèque Adafruit
Carte : AFMS
Motor M1 : myMotor

*/
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"

Adafruit_MotorShield AFMS = Adafruit_MotorShield();
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);

//--- Capteur courant --
float OFFSET =2.500650408-0.0480952;
#define pinC A2

//--- Interface Users --
#define BoutonLACE 12
#define BoutonUN 7
#define BoutonPULSE 8
boolean Lace;
boolean pulse;
boolean un;

//--- RGB --
#define PIN_LED_R 5
#define PIN_LED_G 6
#define PIN_LED_B 11

//--- Bluetooth --
#include <SoftwareSerial.h>
#define Rx 13
#define Tx 4
#define baudRate 9600
SoftwareSerial BTserial(Rx,Tx);
long unsigned int temps=0;

void setup()
{
    //---Init Bluetooth --
    Serial.begin(115200);
    BTserial.begin(baudRate);
    Serial.print("Sketch:  "); Serial.println(__FILE__);
    Serial.print("Uploaded: "); Serial.println(__DATE__);
    Serial.print("BTserial started at "); Serial.println(baudRate);
    while (!Serial) {
        ; //Need for native USB
    }
    Serial.println("Goodnight moon!"); //Test
    BTserial.println("Hello, world?");
    //Activation Carte Adafruit Motor Shield v2.3
    AFMS.begin();

    //init MOTOR
    myMotor->setSpeed(255);

    //Capteur vitesse
    attachInterrupt(interrupt,tops,RISING);
    pinMode(2,INPUT_PULLUP);
    Timer1.initialize(Te);
    Timer1.attachInterrupt( timerIsr );

    //Interface Users
    pinMode(BoutonLACE,INPUT);
    pinMode(BoutonUN,INPUT);
    pinMode(BoutonPULSE,INPUT);
}

void loop()
{
    //Default setting
    myMotor->run(RELEASE);
    displayColor(15, 32, 32);
    initialise();
    String commande="";

    //Scan users
    Lace      = digitalRead(BoutonLACE);
    pulse     = digitalRead(BoutonPULSE);
    un       = digitalRead(BoutonUN);
    if (BTserial.available()){
        int len = 0;
        while (commande.substring(len-4,len-2) != String("/c")){
            commande +=BTserial.readString();
            len = commande.length();
        }
        commande.remove(len-4,len-2);
        Serial.print(commande);
    }
    if ((millis()-temps)>10000){
        BTserial.println("Type a/c for autolacing");
        BTserial.println("Type +/c for manuel lacing +");
        BTserial.println("Type -/c for manuel lacing -");
        temps = millis();
    }
}
```

Annexe 2(Suite)

```
//--Action users--                                // -- Mesure une moyenne sur x mesure sur la dite broche -- // -- LACAGE --
// MODE LACAGE                                         void lacage(int vit,int moy,char broche)
while (Lace || commande[0] == 'a'){                  {
    displayColor(128,0,0);                           long unsigned int Sensormoy = 0;
    lacage(150,255,pinC);                          float mesureB(int x,char broche)
    Lace      = digitalRead(BoutonLACE);           {
    blinkRGB(2,3);                                 long unsigned int sensorValue = analogRead(broche);
    commande="";                                    Sensormoy+=sensorValue;
}                                                 for (int i=1;i<=x;i++){
double temps =0;                                 double moy = Sensormoy/x;
                                                    return moy;
}
// MODE REGLAGE MANUELLE +                           void initialise()
while (pulse || commande[0] == '+'){               {
    displayColor(0,128,0);                         OFFSET=mesureB(50,pinC);
    myMotor->run(FORWARD);                      }
    pulse      = digitalRead(BoutonPULSE);         void tops ()
    commande="";                                 {
}
// MODE REGLAGE MANUELLE -                           count++;
while (un || commande[0] == '-'){
    displayColor(0,128,0);                         void timerIsr()
    myMotor->run(BACKWARD);                     {
    un       = digitalRead(BoutonUN);              acceleration=count-vitesse;
    commande="";                                 vitesse = count;
}
-- Permet de changer de Couleur de la RGB          count=0;
id displayColor(byte r, byte g, byte b)           }
// Version anode commune
analogWrite(PIN_LED_R, r);
analogWrite(PIN_LED_G, g);
analogWrite(PIN_LED_B, b);

// -- Fonctions Capteur de vitesse --
void tops ()                                     {
{
    count++;                                 acceleration=count-vitesse;
}
void timerIsr()                               vitesse = count;
{
    acceleration=count-vitesse;
    vitesse = count;
    count=0;
}

// -- Mesure Courant
initialise();                                     count=0;
                                                    long unsigned int timer=micros();
                                                    int peak=0;

//Demarage Moteur
myMotor->setSpeed(vit);
myMotor->run(FORWARD);
BTserial.println("Autolacing...");

double mesure=0;
double acc=0;
double mesureold=(mesureB(50,pinC)-OFFSET)*11.235;
long unsigned int timer=micros();
int peak=0;

//Attente de saturation tension ou vitesse trop basse
while((mesure <= 900 || peak==0) && (acc > -20))
{
    //Mesure Courant
    mesure =(mesureB(50,pinC)-OFFSET)*11.235;//10^3*(5/1023)/0.435

    //Detection du pic
    double derivative=(mesureold-mesure)/(micros()-timer);
    if (derivative >400){
        peak=1;
    }

    timer =micros();
    mesureold=mesure;

    //mesure acceleration
    acc=acceleration*10.456;//2*pi/PPR/(Te)^2
}
myMotor->run(RELEASE);
BTserial.println("Lacing DONE");
}
```