

Bobineuse minimale Version Arduino-Visseuse

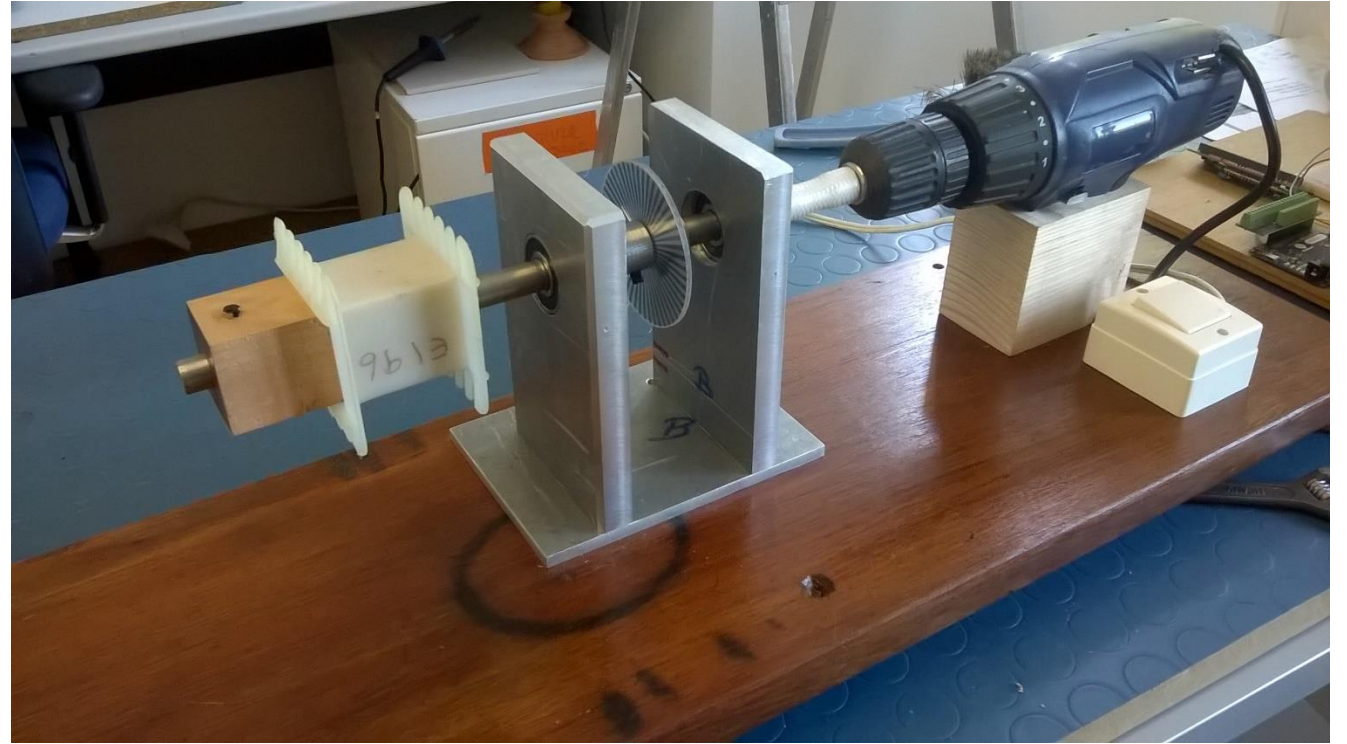
2017

Cahier des charges

- Ne pas nécessiter un PC, donc ARDUINO pour faire le pilotage
- Entraînement broche simple, peut-être manuel ou par visseuse
- Donc pilotage du guide-fil en fonction de la rotation de la broche (encodeur optique)

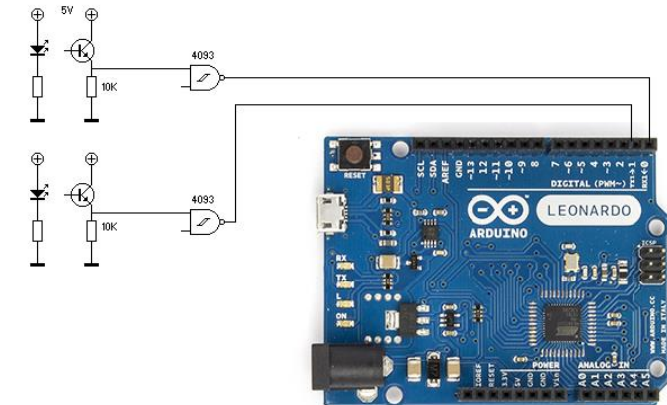
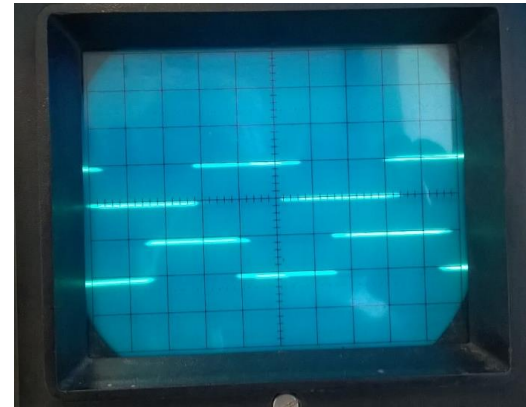
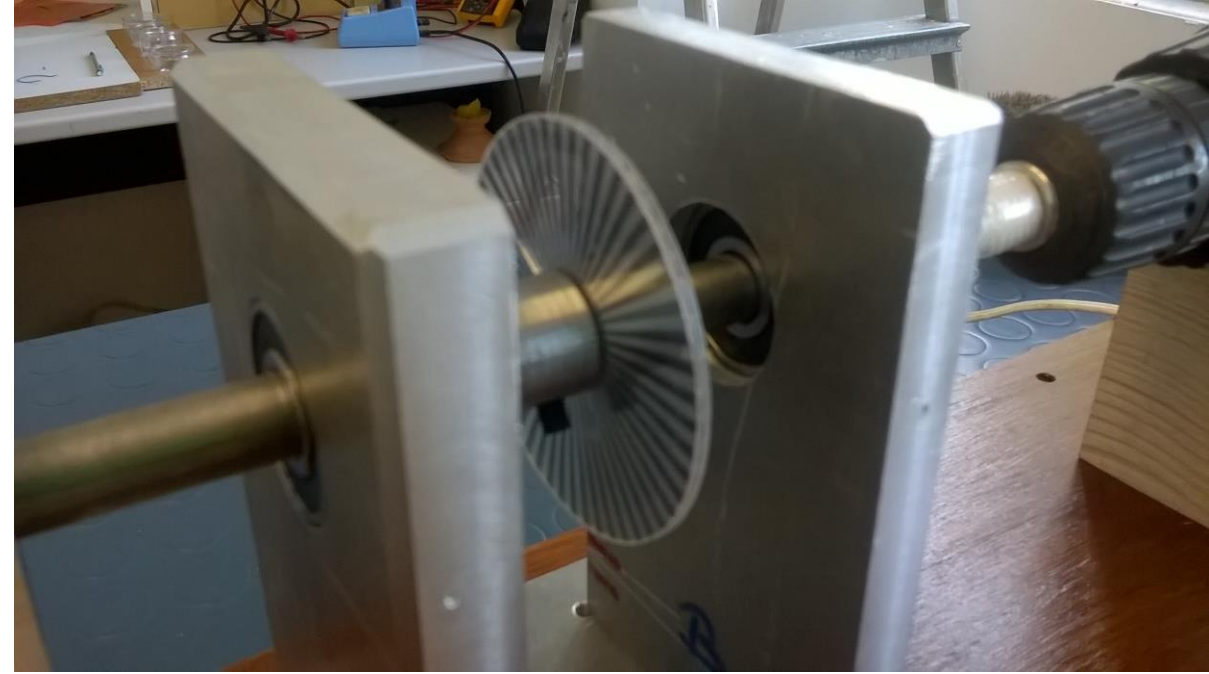
BROCHE

- Usinage de flasques en alu avec axe diam 12 sur roulements à billes
- Entraînement souple (flexible de douche!) par moteur ancienne visseuse (batterie HS)
- Variateur de vitesse natif (Triac) sur gâchette intégré à une pédale
- Alim sous-voltée: 12V au lieu de 18V



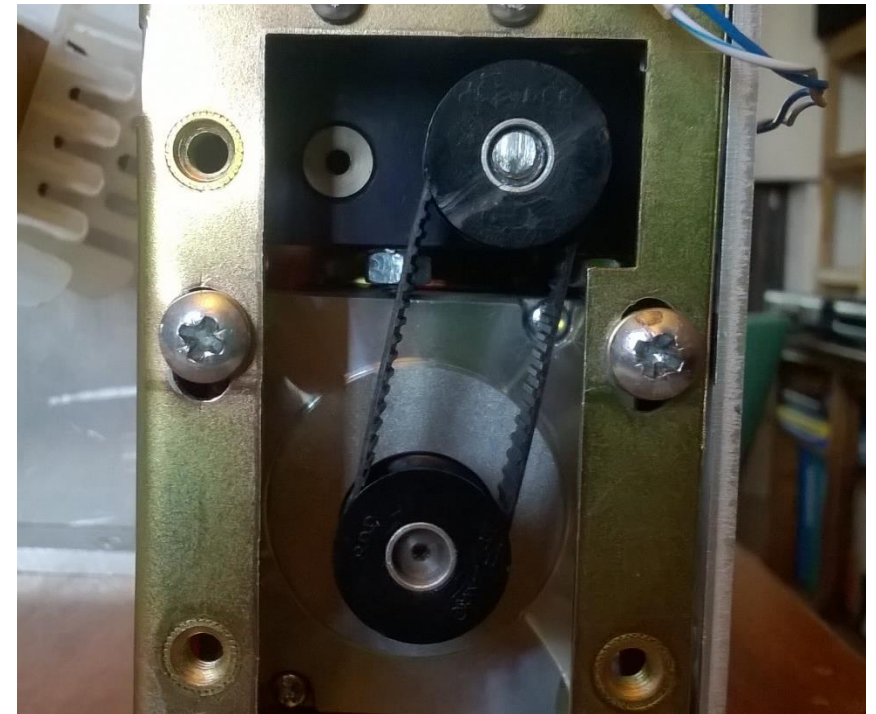
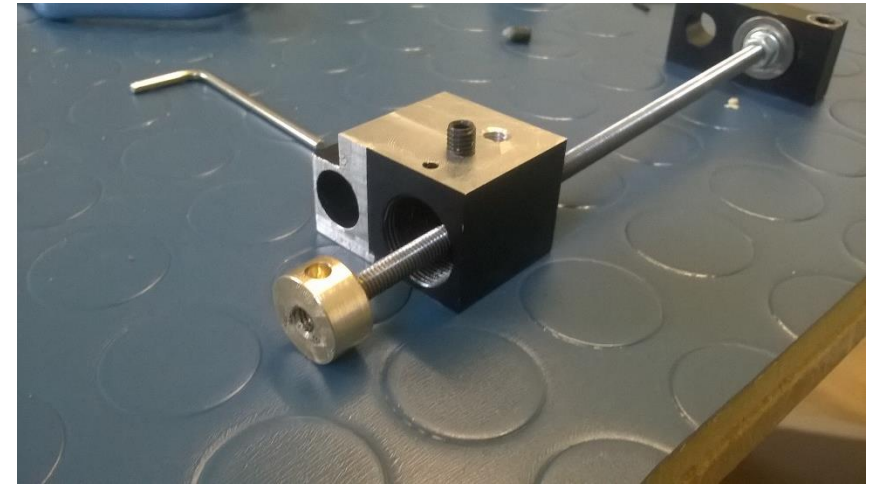
Capteur

- Encodeur maison sur l'axe, 5 fentes par tour
(la photo montre la version initiale avec 40 fentes qui s'est avérée peu performante: trop d'interruptions à gérer par l'Arduino. Avec 5 fentes c'est OK)
- Les CI des fourches optiques incluent un 4093 qui fait Trigger de Schmidt et fournit des X et Y carrés propres (0-5V)
sans oublier de relier la patte inutilisée au HIGH



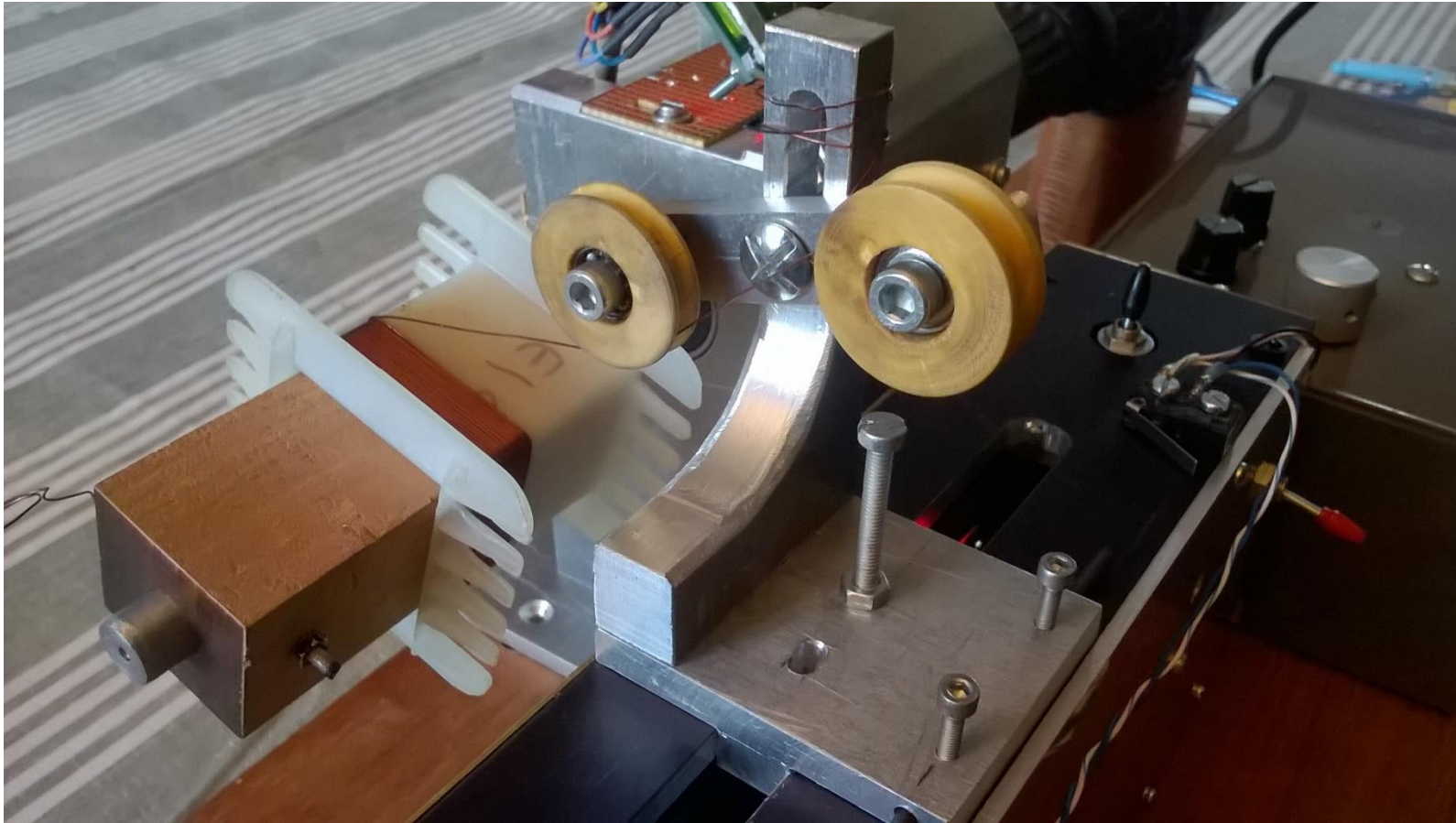
Guide-fil: vis de guidage

- PAP NEMA23 et driver Gecko G251
- Transmission par courroie crantée
- Fabrication d'une vis de guidage 0,8mm (M4) avec noix bronze maison
- Contacts de fin de course qui coupent le moteur afin de ne pas endommager le mécanisme en bout de course



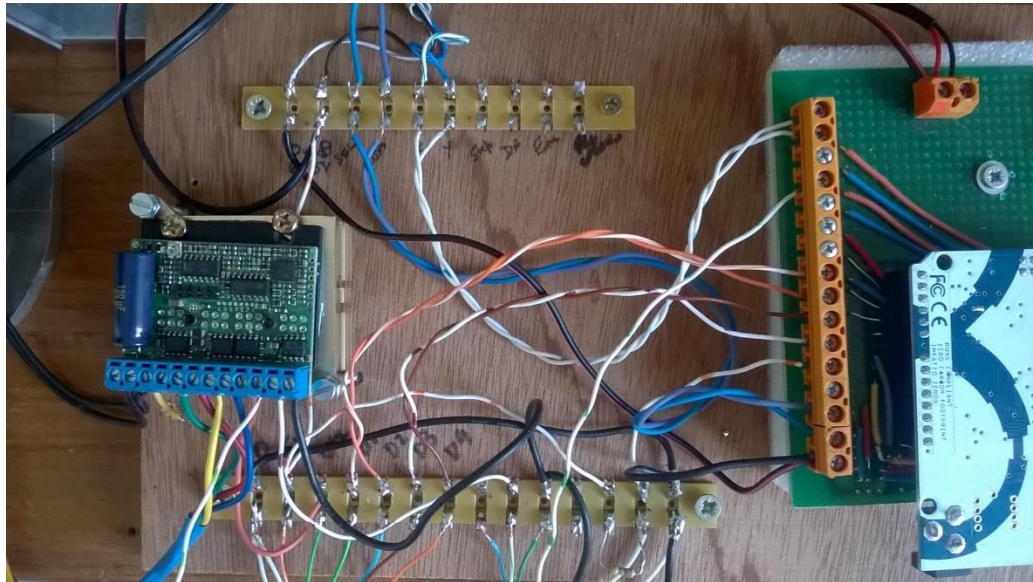
Guide-fil: roulettes

- Roulettes buis « maison » montées sur roulement



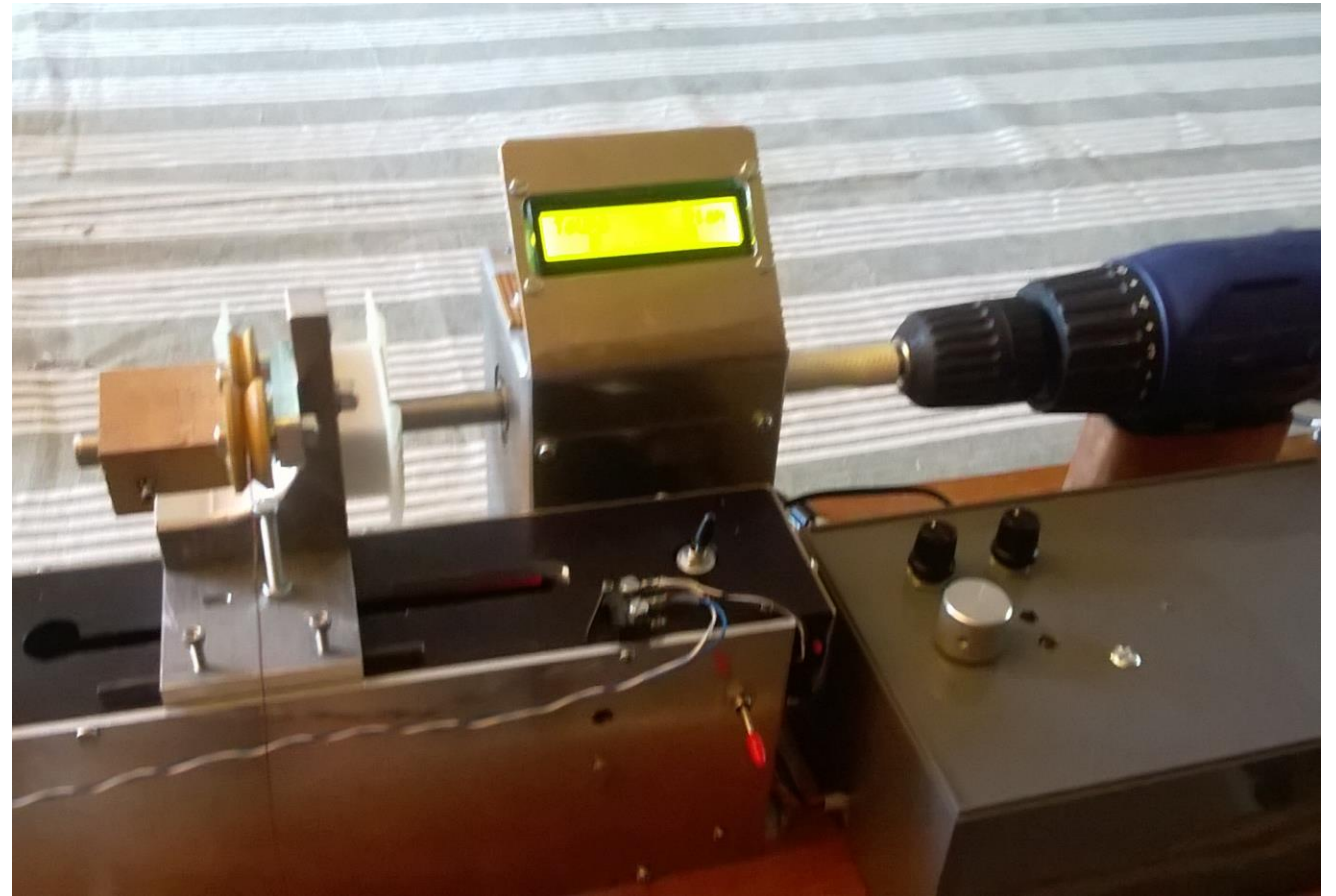
Boitier de contrôle Arduino

- Affichage LCD pour compte tours et saisie du diamètre (= avance)
- Pilotage du guide-fil (envoi des steps)
- = fonctions annexe: RAZ compteur, Avance rapide du guide



Interface Homme Machine (ben tiens)

1. Réglage Diamètre sur 2 encodeurs (dizaine, unité)
2. Sens du guide GD ou DG
3. Mise au repos du PAP (Disable pour éviter de le laisser chauffer dans les phases intermède)
4. Raz compteur
5. Acquiescement fin de course par déconnexion furtive de ceux-ci
6. Avance rapide
 1. Activation par Poussoir
 2. Potentiomètre réglage vitesse



Connexions Arduino

- Connexions:
 - LCD sur SCL/SDA (résistances Pullup nécessaires)
 - Encodeur Broche sur pins 0, 1 (interruptibles)
 - Encodeur Bouton Diam sur pins A0 et A1 + A2 et A3
 - Envoi STEP sur pin 5
 - Poussoir « Avance rapide » sur pin 8 (pullup)
 - Poussoir RAZ compteur sur pin 12 (pullup)
 - Pot avance rapide sur pin A5

Résultat?

- Ca marche nickel!



Code 1/5: En-tête

```
#define ENCODER_USE_INTERRUPTS
#define ENCODER_OPTIMIZE_INTERRUPTS
#include <Encoder.h>
Encoder CompteurBroche(0, 1); //capteur compte-tours
Encoder knobLeft(A0, A1); // bouton DIAM dizaines
Encoder knobRight(A2, A3); //bouton DIAM unites

#include <Wire.h> // Comes with Arduino IDE

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Set the LCD I2C address


static char outstr[15];
long positionCompteur = 0L;
long newCompteur = 0L;
long Compteur = 0L;
int DiamLeft = -999;
int newLeft = 0;
int DiamRight = -999;
int newRight = 0;
int Diametre = 0;
int RazCptValue = 1;
int AvanceRapide = 1;
int VitesseAvance = 0;
int Occ = 0;
```


Code 2/5: SETUP

```
void setup() {  
    lcd.begin(16, 2); // initialize the lcd for 16 chars 2 lines, turn on backlight  
    lcd.clear();  
    lcd.setCursor(0, 0); //Start at character 0 on line 0  
    lcd.print("Tours      Diam");  
  
    pinMode(5, OUTPUT); //broche 5 pour sortie STEP  
    pinMode(A5, INPUT); //broche A5 pour lecture POT  
    pinMode(8, INPUT); // broche 8 bouton poussoir Avance Rapide  
    pinMode(12, INPUT); // broche 12 bouton poussoir RAZ compteur  
}
```

Code 3/5: LOOP (1)

```
void loop() {
  // on surveille surtout le Compteur BROCHE (sur Pins Interrupt capable)
  newCompteur = CompteurBroche.read();
  if (newCompteur != positionCompteur) {    // nouvelle interruption
    positionCompteur = newCompteur;
    Occ = newCompteur % 4          ;        // 20 modulo 5 = 5 séries par tour
    if (Occ == 0) {                // à la fin de la periode (cinquieme de tour), avance de diametre*5
      Salve(Diametre * 5);         // voir la feuille XL PULSES pour calculs
    }
  }                                     // fin de l'événement avance BROCHE

  //on surveille aussi la mise à jour du DIAM
  newLeft = knobLeft.read();
  newRight = knobRight.read();
  if (newLeft != DiamLeft | newRight != DiamRight) {
    DiamLeft = newLeft;
    DiamRight = newRight;
    Diametre = (DiamLeft * 10) + (DiamRight);
    lcd.setCursor(12, 1); // maj affichage diametre uniquement s'il a change
    lcd.print("      ");
    lcd.setCursor(12, 1);
    lcd.print(Diametre);
  }                                     // fin de l'événement changement DIAM
}
```

Code 4/5: LOOP (2)

```
// Événement Avance Rapide
AvanceRapide = digitalRead(8); //
if (AvanceRapide < 1) {
    VitesseAvance = analogRead(A5); //lecture du POT pour vitesse avance
    Salve(VitesseAvance);           //
}                                  // fin de l'événement AvanceRapide

// Événement RAZ Compteur
RazCptValue = digitalRead(12);
if (RazCptValue < 1) {
    CompteurBroche.write(0);
    lcd.print("          "); //effacement de tous les caractères jusque -999.9
} // fin de l'événement RAZ Compteur

// Mise à jour des Affichages
dtostrf(newCompteur / 20, 4, 0, outstr);
lcd.setCursor(0, 1);
lcd.print(outstr);

} // fin de LOOP
```


Code 5/5: fonction Salve

```
void Salve(int param1) {           // fonction Salve qui envoie les steps au moteur sur Pin 5
  for (int compteur = 0; compteur < param1; compteur++)
  {
    digitalWrite(5, LOW);
    delayMicroseconds(50);         //100 microsecondes au total (50 LOW 50 HIGH) semble le plancher
    digitalWrite(5, HIGH);
    delayMicroseconds(50);
  }
}
```