

```

int threshold = 730; //Poid d'une tasse d'eau sur le capteur,
int attente = 250; //Vitesse de clignotement de la led en
attente
int distribution = 10;
int calibrating = 10;

int tempo = 1000; //Vitesse de clignotement de la led

//Constants:
const int ledPin = 13;    //pin de la LED integrée
const int sensorPin = A0; //pin A0 pour l'entrée capteur
const int buttonPin = 7;  //pin pour la lecture du bouton
poussoir

void setup() {
  pinMode(4, OUTPUT); // On initialise les broches D4 et D5
en sorties
  pinMode(5, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(115200);
  pinMode(buttonPin, INPUT);
}

//
int reglage_angle = 200; // Seuil initial du compteur
int current_angle = 0;   //Compteur en pas moteur

bool Button_pressed = false;
bool Turning = false;
bool fini = false;
int buttonstate;

int value;

void decharge_the() {
  if (not Turning) {
    Turning = true;
    tempo = distribution;
  }
  else {
    if (current_angle < reglage_angle) {

```

```

        for (int i = 0; i <= 5; i++) {
            digitalWrite(5, LOW);
            delay(1); // On envoie 200 impulsions pour faire
tourner le moteur de 200 pas dans un sens
            digitalWrite(5, HIGH); // La valeur de tempo
determine le temps entre 2 impulsions en ms
            delay(1);
            current_angle++;
        }
    }
    else {
        arreter_distrib();
    }
}
}

```

```

void arreter_distrib() {
    Turning = false;
    current_angle = 0;
    tempo = attente;
    fini = true;
    Button_pressed = false;
}

```

```

void calibration() {
    buttonstate = digitalRead(buttonPin);
    if (buttonstate == HIGH) {
        Button_pressed = true;
        if (not Turning) {
            Turning = true;
            tempo = calibrating;
        }
    }
    else {
        for (int i = 0; i <= 5; i++) {
            digitalWrite(5, LOW);
            delay(1); // On envoie 200 impulsions pour faire
tourner le moteur de 200 pas dans un sens
            digitalWrite(5, HIGH); // La valeur de tempo
determine le temps entre 2 impulsions en ms
            delay(1);
            current_angle++;
        }
    }
}

```

```

        Serial.println(current_angle);
    }

}

else {
    Button_pressed = false;
    reglage_angle = current_angle;
    arreter_distrib();
}
}

void loop() {
    value = analogRead(sensorPin);           //Read and save analog
value from potentiometer
    Serial.println(value);
    digitalWrite(ledPin, HIGH);
    delay(tempo);
    digitalWrite(ledPin, LOW);
    delay(tempo);
    // On met la broche DIR à l'état haut
    digitalWrite(4, LOW);

    if (value > threshold) {
        buttonstate = digitalRead(buttonPin);
        if (not Turning and not fini) {
            if (buttonstate == LOW) {
                decharge_the();
            }
            else {
                calibration();
            }
        }
        if (Turning and not fini) {
            if (Button_pressed == HIGH) {
                if (buttonstate == HIGH) {
                    calibration();
                } else {

```

```
        reglage_angle = current_angle;
        arreter_distrib();
    }
}
else {
    if (buttonstate == LOW) {
        decharge_the();
    } else {
        arreter_distrib();
    }
}
}
} else {
    arreter_distrib();
    fini = false;
}
}
```