

Transmisiones Flexibles

```
#include "HX711.h"

HX711 scale(4, 5);

float calibration_factor = 21000; // this calibration factor is adjusted according to my load cell
float units;
float ounces;
const int irPin = 2;
int triggers = 0;
unsigned long elapsedTime;
int laststate = 0;
const unsigned long sampleTime = 1000;
int rpmMax = 0;

void setup() {
    Serial.begin(9600);
    pinMode(irPin, INPUT);
    Serial.println("HX711 calibration sketch");
    Serial.println("Remove all weight from scale");
    Serial.println("After readings begin, place known weight on scale");
    Serial.println("Press + or a to increase calibration factor");
    Serial.println("Press - or z to decrease calibration factor");
    scale.set_scale();
    scale.tare();
    long zero_factor = scale.read_average();
    Serial.print("Zero factor: ");
    Serial.println(zero_factor);
}

void loop() {

    scale.set_scale(calibration_factor);
    Serial.print("Reading: ");
    units = scale.get_units(), 10;
    if (units < 0)
    {
        units = 0.00;
    }
    ounces = units * 0.035274;
    Serial.print(ounces);
    Serial.print(" kg");
    Serial.print(" calibration_factor: ");
    Serial.print(calibration_factor);
    Serial.println();
    int rpm = rpmGet();
    if (rpm > rpmMax)
```

```

    {
        rpmMax = rpm;
    }
    Serial.print("rpm: ");
    Serial.println(rpm);
}

int rpmGet()
{
    unsigned long currentTime = 0;
    unsigned long startTime = millis();
    while (currentTime <= sampleTime)
    {
        int val = digitalRead(irPin);
        if(!val)
        {
            if(laststate){
                triggers++;
                laststate=0;
            }
        }
        else
        {
            laststate = 1;
        }
        currentTime = millis() - startTime;
    }
    int countRpm = int(120000/float(sampleTime))*(triggers/2);
    triggers=0;
    return countRpm;
}

```

// Nota. La zona del código que se encarga de calcular las RPM fue tomado de: DIY Torquemeter - Device to Measure Dynamic Torque, por INTEGZA, Febrero 2019, (<https://www.hackster.io/integza/diy-torquemeter-device-to-measure-dynamic-torque-6608ae#code>)

// Nota 2: La zona del código que se encarga del tratamiento de señal de la celda de carga a través del amplificador HX711 fue tomado de: Celda de Carga 50 Kg y Arduino UNO, por PicaioVideos, (https://www.youtube.com/watch?v=43xcnJxuvjY&ab_channel=PicaioVideos)

Banco de Taladro

```
int LecUp = 0;
int LecDown = 0;
const int UpPin = 4;
const int DownPin = 2;
unsigned long InicialTime = 0;
unsigned long LastTime = 0;
unsigned long Differential = 0;

void setup()
{
    Serial.begin(9600);
    pinMode(UpPin, INPUT);
    pinMode(DownPin, INPUT);
}

void loop()
{
    LecUp=digitalRead(UpPin);
    while(LecUp == 1)
    {
        LecUp=digitalRead(UpPin);
    }
    InicialTime=micros();
    LecDown=digitalRead(DownPin);
    while(LecDown==1)
    {
        LecDown=digitalRead(DownPin);
    }
    LastTime=micros();
    Differential=LastTime-InicialTime;
    Serial.print("\n");
    Serial.print("Diferencial: ");
    Serial.println(Differential);
}
```