

## Table of contents

[1. calibracao.ino](#)

[2. balanca\\_aero.ino](#)

## calibracao.ino

Esse código é usado para calibrar as duas células de carga a partir de massas conhecidas.

```
#include "HX711.h"                                // Biblioteca HX711

//Pinos de dados
#define DOUT1 A0
#define DOUT2 A4
//pinos de clock
#define CLK1 A1
#define CLK2 A5
HX711 ARRASTO_HX711; // define instancia balança HX711
HX711 SUSTENTACAO_HX711;

float fator_de_calibracao_arrasto = 1863150.00; // fator de calibração para teste inicial
float fator_de_calibracao_sustentacao = 302200.00;

void setup()
{
  Serial.begin(9600); // monitor serial 9600 Bps
  ARRASTO_HX711.begin(DOUT1, CLK1);
  SUSTENTACAO_HX711.begin(DOUT2, CLK2);

  Serial.println("\nCalibração");
  Serial.println("Remova as massas");
  Serial.println("Depois que as leituras começarem, coloque uma massa conhecida sobre a Balança");
  Serial.println("Fator de calibração para o arrasto:");
  Serial.println("Pressione a,s,d,f para aumentar Fator de Calibração por 10,100,1000,10000 respectivamente");
  Serial.println("Pressione z,x,c,v para diminuir Fator de Calibração por 10,100,1000,10000 respectivamente");
  delay(3000);
  Serial.println("Fator de calibração para a sustentação:");
  Serial.println("Pressione h,j,k,l para aumentar Fator de Calibração por 10,100,1000,10000 respectivamente");
  Serial.println("Pressione u,i,o,p para diminuir Fator de Calibração por 10,100,1000,10000 respectivamente");
  Serial.println("Após leitura correta das massas, pressione y e/ou t para TARA(zerar) ");
  delay(3000);

  ARRASTO_HX711.set_scale(); // configura a escala da Balança
  SUSTENTACAO_HX711.set_scale();

  ARRASTO_HX711.tare();
  SUSTENTACAO_HX711.tare();
}

void loop()
{
  ARRASTO_HX711.set_scale(fator_de_calibracao_arrasto); // ajusta fator de calibração
  SUSTENTACAO_HX711.set_scale(fator_de_calibracao_sustentacao);

  Serial.print("Arrasto Massa: ");
  Serial.print(ARRASTO_HX711.get_units(3),4);
  Serial.print(" kg");
  Serial.print("\t\tFator de Calibração: "); // imprime no monitor serial
  Serial.println(fator_de_calibracao_arrasto); // imprime fator de calibração
  Serial.print("Sustentação Massa: ");
  Serial.print(SUSTENTACAO_HX711.get_units(3),4);
  Serial.print(" kg");
  Serial.print("\t\tFator de Calibração: "); // imprime no monitor serial
  Serial.println(fator_de_calibracao_sustentacao); // imprime fator de calibração
  Serial.print("\n");
  delay(2000); // atraso de 2 segundos

  if (Serial.available()) // reconhece letra para ajuste do fator de calibração
  {
    char temp = Serial.read();

    // ARRASTO
    if (temp == '+' || temp == 'a') // a = aumenta 10
      fator_de_calibracao_arrasto += 10;
    else if (temp == '-' || temp == 'z') // z = diminui 10
```

```

    fator_de_calibracao_arrasto -= 10;
else if (temp == 's') // s = aumenta 100
    fator_de_calibracao_arrasto += 100;
else if (temp == 'x') // x = diminui 100
    fator_de_calibracao_arrasto -= 100;
else if (temp == 'd') // d = aumenta 1000
    fator_de_calibracao_arrasto += 1000;
else if (temp == 'c') // c = diminui 1000
    fator_de_calibracao_arrasto -= 1000;
else if (temp == 'f') // f = aumenta 10000
    fator_de_calibracao_arrasto += 10000;
else if (temp == 'v') // v = diminui 10000
    fator_de_calibracao_arrasto -= 10000;
else if (temp == 't')
    ARRASTO_HX711.tare(); // t = zera a Balança

// SUSTENTAÇÃO
else if (temp == 'u') // z = diminui 10
    fator_de_calibracao_sustentacao -= 10;
else if (temp == 'j') // j = aumenta 100
    fator_de_calibracao_sustentacao += 100;
else if (temp == 'i') // i = diminui 100
    fator_de_calibracao_sustentacao -= 100;
else if (temp == 'k') // k = aumenta 1000
    fator_de_calibracao_sustentacao += 1000;
else if (temp == 'o') // o = diminui 1000
    fator_de_calibracao_sustentacao -= 1000;
else if (temp == 'l') // l = aumenta 10000
    fator_de_calibracao_sustentacao += 10000;
else if (temp == 'p') // p = diminui 10000
    fator_de_calibracao_sustentacao -= 10000;
else if (temp == 'y')
    SUSTENTACAO_HX711.tare();
}
}

```

## balanca\_aero.ino

Esse é o código principal da solução de software. A partir dele obtém-se as massas medidas pelas células de carga, bem como os valores das forças de arrasto e de sustentação, dentre outros, conforme os objetivos do trabalho.

```

#include "HX711.h"

//pinos de entrada
#define pino_entrada1 A0
#define pino_entrada2 A4
//pinos de clock
#define pino_clock1 A1
#define pino_clock2 A5

float fator_de_calibracao1 = 1863150.00;
float fator_de_calibracao2 = 390200.00;

HX711 balanca_aero_arrasto;
HX711 balanca_aero_sustentacao;

//dados para as funções de arrasto e de sustentacao
double fg = 9.81; // m/s^2 - aceleracao gravitacional
double corda = 0.079; // m - 8 cm
double densidade_do_fluido = 1.225/1.11; // kg/m^3
float velocidade = 11.15; // m/s
double velocidade_2 = velocidade*velocidade; // m^2/s^2 - velocidade ao quadrad
float envergadura = 0.145;
float massal = 0.0, massa2 = 0.0;

void setup() {
    Serial.begin(9600);

    balanca_aero_arrasto.begin(pino_entrada1, pino_clock1);
    balanca_aero_sustentacao.begin(pino_entrada2, pino_clock2);

    Serial.print("\t\tProjeto Integrador de Engenharia 1\n");
    Serial.print("\t\tUniversidade de Brasília - Faculdade do Gama\n\n");
    Serial.print("\t\t\tBALANÇA AERODINÂMICA\n\n");
    balanca_aero_arrasto.set_scale();
    balanca_aero_sustentacao.set_scale();

    //define o fator de calibracao

```

```

    balanca_aero_arrasto.tare();
    balanca_aero_sustentacao.tare();
}

void loop() {
    //zera a balanca
    balanca_aero_arrasto.set_scale(fator_de_calibracao1);
    balanca_aero_sustentacao.set_scale(fator_de_calibracao2);

    //get_units realizara a media dos 15 valores de massas medidas
    massa1 = balanca_aero_arrasto.get_units(15);
    massa2 = balanca_aero_sustentacao.get_units(15);
    massa1 = abs(massa1);
    massa2 = abs(massa2);

    Serial.print("Massas aferidas pelas células de carga (kg) :\n\n");
    Serial.print("\t\tMassa 1 (ARRASTO) = ");
    Serial.print(massa1,4);
    Serial.print("\t Força= ");
    Serial.println(massa1*fg,4);
    Serial.print("\t\tMassa 2 (SUSTENTAÇÃO)= ");
    Serial.print(massa2,4);
    Serial.print("\t Força= ");
    Serial.print(massa2*fg,4);
    Serial.print("\n\n");

    //chamada das funcoes
    double sustentacao, arrasto ;
    sustentacao = coef_sustentacao(massa2);
    arrasto = coef_arrasto(massa1);

    //imprime os coeficientes
    Serial.print("Coeficientes calculados:\n\n");
    Serial.print("\t\tCoeficiente de arrasto: ");
    Serial.println(arrasto,4);
    Serial.print("\n\t\tCoeficiente de sustentação: ");
    Serial.println(sustentacao,4);
    //Atraso de 5,5 segundos
    delay(5500);
    Serial.print("-----//-----\n\n");
}

double coef_sustentacao(double massa_da_celula){
    return ((massa2*fg)/((0.5)*(densidade_do_fluido)*(velocidade_2)*(corda)*(envergadura)));
}

double coef_arrasto(double massa_da_celula){
    return ((massa1*fg)/((0.5)*(densidade_do_fluido)*(velocidade_2)*(corda)*(envergadura)));
}

```

[Back to top](#)

Made with [Docsie.io](#)