

```
// Project: MES-RC Cristino -  
Versão enxuta 20251125 -  
WORKING!!!!!!  
// Board: ESP32-C3 Mini  
// Sensores: AD8232 (EMG),  
Screw_nut (Stop open), Push_Claw  
(Stop cloe)  
  
#include <Arduino.h>  
#include <filters.h> //  
https://github.com/MartinBloedorn/  
libFilter  
  
//Pinos  
const int EMG = 0; // Entrada do  
sinal EMG  
const int tilt_ = 3;  
const int potPin = 2;  
const int Screw_nut_ = 1;  
const int Push_Claw_ = 4;  
const int led = 8;  
const int H1 = 20, H2 = 21;
```

```
// Filtros
const float fHP = 10.0;    //
High-pass cutoff (Hz)
const float fLP = 40.0;    //
Low-pass cutoff (Hz)
const float Ts = 0.01;     // Tempo
de amostragem (s)
Filter hp(fHP, Ts,
IIR::ORDER::OD3,
IIR::TYPE::HIGHPASS);
Filter lp(fLP, Ts,
IIR::ORDER::OD3);
int flag = 0;

// Função para ler e filtrar N
amostras do EMG
float readMES(int N = 20) {
    float sum = 0;
    for (int i = 0; i < N; i++) {
        float val = analogRead(EMG);
        val = lp.filterIn(val);
```

```
    val = hp.filterIn(val);
    sum += abs(val);
    delay(1);
}

return (sum / N) * 10; // ganho
ajustável conforme sensor
}

void setup() {
    Serial.begin(9600);
    pinMode(led, OUTPUT);
    pinMode(H1, OUTPUT);
    pinMode(H2, OUTPUT);
    digitalWrite(H1, LOW);
    digitalWrite(H2, LOW);
    pinMode(Screw_nut_, INPUT);
    pinMode(Push_Claw_, INPUT);
}

void loop() {
    int Push_Claw =
digitalRead(Push_Claw_);
```

```
int Screw_nut =
digitalRead(Screw_nut_);
int tilt = digitalRead(tilt_);
float MES = readMES();
float MESTh = analogRead(potPin)
* 0.0244; // threshold ajustável
delay(1);

float MES1 = (readMES() - 5);
float MES2 = (readMES() - 10);
float MES3 = (readMES() - 15);
float MES4 = (readMES() - 20);

digitalWrite(H1, LOW);
digitalWrite(H2, LOW);
delay(1);

if (flag == 0 && MES >= MESTh &&
MES1 >= MESTh && MES3 >= MESTh &&
MES4 >= MESTh && tilt == 0 &&
Screw_nut == 1) // retirado
Battery >= Battery_sd
```

```
{  
    while (MES >= MESTh) {  
        MES = 0;  
        Screw_nut =  
digitalRead(Screw_nut_);  
        if (Screw_nut == 0 || tilt ==  
1) { break; }  
        delay(2);  
        MES = readMES();  
        MESTh = analogRead(potPin) *  
0.0244; // threshold ajustável  
        flag = 1; // Abrindo  
        digitalWrite(led, HIGH);  
        delay(2);  
        digitalWrite(H1, LOW);  
        digitalWrite(H2, HIGH);  
        delay(200);  
        digitalWrite(H1, LOW);  
        digitalWrite(H2, LOW);  
        digitalWrite(led, LOW);  
    }  
    flag = 0;
```

```
}

if (MES < MESTh) {
    digitalWrite(H1, LOW);
    digitalWrite(H2, LOW);
    flag = 2;
}

if (flag == 2 && MES < MESTh) {
    while (MES <= MESTh) {
        Push_Claw =
digitalRead(Push_Claw_);
        if (Push_Claw == 1) { break;
    }
    MES = readMES();
    MESTh = analogRead(potPin) *
0.0244; // threshold ajustável
    delay(2);
    digitalWrite(H1, HIGH);
    digitalWrite(H2, LOW);
    delay(200);
    digitalWrite(H1, LOW);
```

```
    digitalWrite(H2, LOW);
    delay(2);
}
flag = 0;
}
delay(2);
}
```