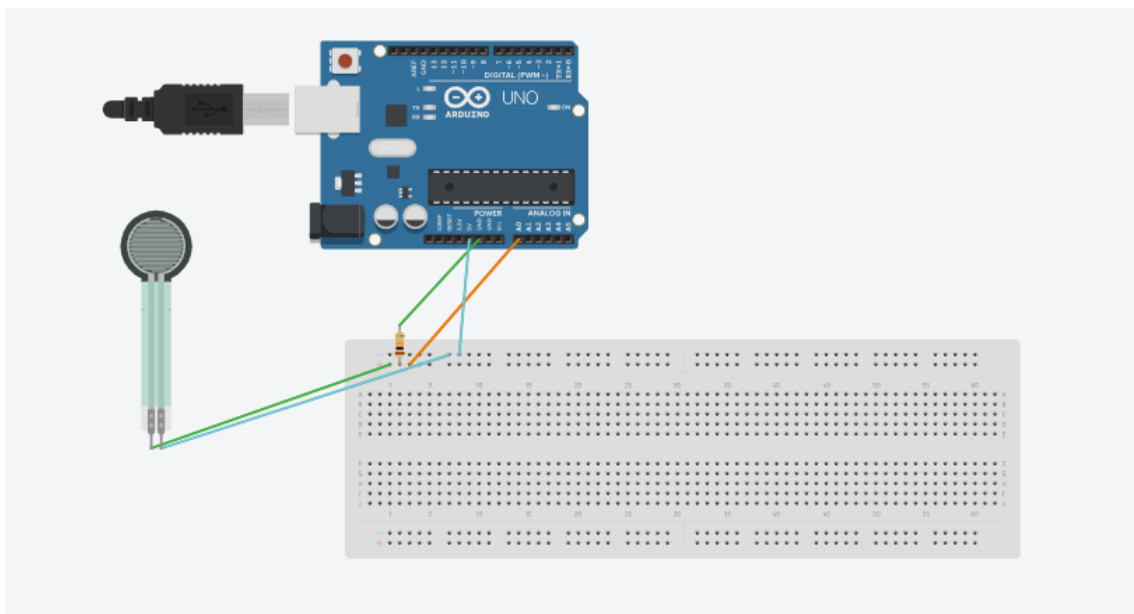


Relatório de Aprendizagem: Uso de um FSR para Medição de Força

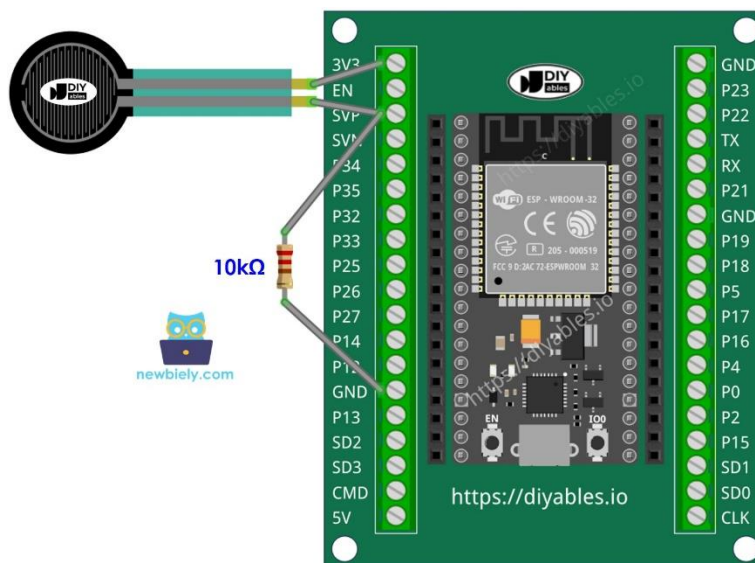
Este relatório tem por objetivo descrever minha experiência de aprendizagem no uso de um sensor FSR (Force-Sensitive Resistor) para medir força utilizando um ESP32. O objetivo principal é avaliar se o FSR pode ser empregado para medir força de forma mais detalhada.

Iniciei os estudos utilizando o Tinkercad, pois não tinha acesso a uma plataforma física no momento. Entretanto, encontrei um problema: o ESP32 não está disponível no Tinkercad, o que me levou a usar um Arduino UNO R3 como alternativa. Tentei explorar outras plataformas, como WOWKI, mas elas também não possuem suporte ao sensor FSR.

Para começar, montei o circuito no Arduino UNO com um resistor de 10 Ω .



Exemplo da montagem de circuito no Esp32:



Fonte: <https://esp32io.com/tutorials/esp32-force-sensor>.

Durante os testes, percebi que o FSR não fornece diretamente um valor de força ao controlador. Em vez disso, ele mede a força aplicada por meio da variação de tensão. Basicamente, quanto maior a força aplicada, maior a tensão no circuito. Essa variação é interpretada pelo microcontrolador, gerando um valor lógico entre 0 e 1023, proporcional à tensão medida.

Segue o código utilizado no ESP32 para interpretar os valores de pressão:

```
1 /*
2  * This ESP32 code is created by esp32io.com
3  *
4  * This ESP32 code is released in the public domain
5  *
6  * For more detail (instruction and wiring diagram), visit https://esp32io.com/tutorials/esp32-force-sensor
7  */
8
9 #define FORCE_SENSOR_PIN 36 // ESP32 pin GPIO36 (ADC0): the FSR and 10K pulldown are connected to A0
10
11 void setup() {
12   Serial.begin(9600);
13
14   // set the ADC attenuation to 11 dB (up to ~3.3V input)
15   analogSetAttenuation(ADC_11db);
16 }
17
18 void loop() {
19   int analogReading = analogRead(FORCE_SENSOR_PIN);
20
21   Serial.print("The force sensor value = ");
22   Serial.print(analogReading); // print the raw analog reading
23
24   if (analogReading < 10) // from 0 to 9
25     Serial.println(" → no pressure");
26   else if (analogReading < 200) // from 10 to 199
27     Serial.println(" → light touch");
28   else if (analogReading < 500) // from 200 to 499
29     Serial.println(" → light squeeze");
30   else if (analogReading < 800) // from 500 to 799
31     Serial.println(" → medium squeeze");
32   else // from 800 to 1023
33     Serial.println(" → big squeeze");
34
35   delay(1000);
36 }
37
```

Fonte: <https://esp32io.com/tutorials/esp32-force-sensor>.

De acordo com minha pesquisa, o sensor FSR não é ideal para medições detalhadas de força, pois o controlador só consegue obter valores analógicos derivados de grandes intervalos da tensão. Esses valores apresentam limitações sendo só útil para identificar se o sensor está recebendo pouca ou muita pressão por exemplo.

Right now you don't have pressure units. You just have a unitless quantity between 0 and 1023 that represents a voltage between 0 and 5V. You'll have to look at the datasheet for the sensor and find out how to convert from voltage to some more usable unit. Or do some sort of calibration by measuring some known force values and calculating the calibration curve yourself. – Delta_G Apr 29, 2020 at 22:08 ✍

To convert analog readings to a voltage use $(\text{analogReading} * 5.0 / 1024.0)$ Assuming you have a 5V reference. If you're using some other reference voltage substitute it for the 5.0 – Delta_G Apr 29, 2020 at 22:11 ✍

No one knows because only you have the manufacturer's datasheet on this sensor. You DO have it, right? – TomServo Apr 29, 2020 at 22:22

Fonte: [sensors - What is the unit of measure for the pressure data I collect using an FSR sensor with an Arduino - Stack Overflow](#)

Aprofundando meus estudos, descobri que é possível calibrar o sensor para obter valores mais precisos. No entanto, mesmo após realizar o mapeamento, não consegui garantir que o microcontrolador interpretasse sempre o valor correto. Ainda há uma margem de erro considerável, próxima de 1 N.

```
1  /* FSR testing sketch.
2
3  Connect one end of FSR to power, the other end to Analog 0.
4  Then connect one end of a 10K resistor from Analog 0 to ground
5
6  For more information see www.ladyada.net/learn/sensors/fsr.html */
7
8  int fsrPin = 0;      // the FSR and 10K pulldown are connected to a0
9  int fsrReading;      // the analog reading from the FSR resistor divider
10 int fsrVoltage;      // the analog reading converted to voltage
11 unsigned long fsrResistance; // The voltage converted to resistance, can be very big so make "long"
12 unsigned long fsrConductance;
13 long fsrForce;      // Finally, the resistance converted to force
14
15 void setup(void) {
16   Serial.begin(9600); // We'll send debugging information via the Serial monitor
17 }
18
19 void loop(void) {
20   fsrReading = analogRead(fsrPin);
21   Serial.print("Analog reading = ");
22   Serial.println(fsrReading);
23
24   // A leitura analógica varia de 0 a 1023, mapeando para 0V a 5V (5000mV)
25   fsrVoltage = map(fsrReading, 0, 1023, 0, 5000);
26   Serial.print("Voltage reading in mV = ");
27   Serial.println(fsrVoltage);
28
29   if (fsrVoltage == 0) {
30     Serial.println("No pressure");
31   } else {
32     // Calculando a resistência do FSR
33     // The voltage = Vcc * R / (R + FSR) where R = 10K and Vcc = 5V
34     // so FSR = ((Vcc - V) * R) / V      yay math!
35     fsrResistance = 5000 - fsrVoltage; // Subtrai a tensão medida de 5V
36     fsrResistance *= 10000; // Multiplica pelo resistor de 10K ohm
37     fsrResistance /= fsrVoltage; // Divide pela tensão medida
38     Serial.print("FSR resistance in ohms = ");
39     Serial.println(fsrResistance);
40     // Calculando a condutância (1 / resistência)
41     fsrConductance = 1000000; // micromhos
42     fsrConductance /= fsrResistance;
43     Serial.print("Conductance in microMhos: ");
44     Serial.println(fsrConductance);
45
46     // Estimativa da força com base na condutância
47     if (fsrConductance <= 1000) {
48       fsrForce = fsrConductance / 80;
49       Serial.print("Force in Newtons: ");
50       Serial.println(fsrForce);
51     } else {
52       fsrForce = fsrConductance - 1000;
53       fsrForce /= 30;
54       Serial.print("Force in Newtons: ");
55       Serial.println(fsrForce);
56     }
57   }
58   Serial.println("-----");
59   delay(10000);
60 }
```

Ainda tentei utilizar um código de uma empresa que produz os sensores, mas mesmo esse ainda apresenta erros como, não conseguir ler valores abaixo 1N (consegui corrigir isso, mas ainda não o outro problema), e dependo do valor ele ainda extrapola os valores (alguns casos ele chegou a errar por 1.5N).

Código com a correção

```
1 // Estimativa da força com base na condutância
2 if (fsrConductance ≤ 1000) {
3     fsrForce = (float)fsrConductance / 80.0;
4 } else {
5     fsrForce = (float)(fsrConductance - 1000) / 30.0;
6 }
7
8 //laço para arredondar o valor
9 if (fsrForce < 1.0) {
10     fsrForce = 1.0;
11 } else {
12     fsrForce = round(fsrForce);
13 }
```

Vídeo: [Video do codigo da empresa](#)

A minha última solução foi tentar mapear valores específicos de força e tensão para tentar fazer o programa ter uma leitura mais precisa:

```
1 const int pinSensor = A0;
2 // Array com pontos específicos de tensão e força
3 float tensoes[] = {0.0,2.02, 2.9, 3.46,3.82, 3.91, 4.04,4.21, 4.32,4.36,4.41, 4.47};
4 float forcas[] = {0.0,0.35, 0.77, 1.53,2.58, 2.99, 3.78,5.23, 6.76,7.42,8.48, 10.0};
5
6 void setup() {
7     Serial.begin(9600);
8 }
9
10 void loop() {
11     int valorBruto = analogRead(pinSensor); // Lê o valor bruto do ADC
12     float tensao = valorBruto * (5.0 / 1023.0); // Converte o valor bruto para tensão (5V)
13
14     // Cálculo da força
15     float forca = interpolarForca(tensao);
16
17     // Exibe os valores de tensão e força no serial
18     Serial.print(" | ");
19     Serial.print(tensao, 2); // Exibe a tensão com 2 casas decimais
20     Serial.print(" V | ");
21     Serial.print(forca, 2); // Exibe a força com 2 casas decimais
22     Serial.println(" N");
23
24     delay(1000);
25 }
26
27 // Função de interpolação linear
28 float interpolarForca(float tensao) {
29     // Interpolação linear entre os pontos conhecidos
30     for (int i = 0; i < 11; i++) { // Atualizado para corresponder ao tamanho do array
31         if (tensao ≥ tensoes[i] && tensao < tensoes[i + 1]) {
32             // A interpolação linear é dada pela fórmula:
33             // f = f1 + (tensao - t1) * (f2 - f1) / (t2 - t1)
34             return forcas[i] + (tensao - tensoes[i]) * (forcas[i + 1] - forcas[i]) / (tensoes[i + 1] - tensoes[i]);
35         }
36     }
37 }
38
```

Vídeo: [Assista ao vídeo no YouTube](#)

Mesmo com o mapeamento, o programa ainda apresenta erros nos valores, embora com uma precisão muito maior, em torno de 0,15 N. No entanto, essa solução não é a mais viável, já que seria necessário realizar todo o processo de recalibração do sensor a cada novo uso no projeto.

A partir da pesquisa, concluí que o sensor FSR pode não ser a melhor alternativa, pois não consegue realizar uma leitura tão precisa dos dados. Ele seria mais adequado para identificar apenas se o paciente está aplicando pressão no sensor com o pé.