

# Projeto de uma Estação Meteorológica

-**Integrantes:** Enzo Cunha, Thomas Chabro, Felipe Catapano, Felipe Drummond.

## Introdução

- Criação de uma estação meteorológica que fosse capaz de medir a temperatura, umidade, pressão e luminosidade local. Utilizando sensores propícios para fazer tais medições, estes tiveram primeiramente que ser calibrados e posteriormente colocados em um sistema com uma protoboard e um arduino já com as equações de calibração obtidas.

## O que foi usado?

- 1 Arduino Uno
- 1 sensor LM35(temperatura)
- 1 sensor DHT22(umidade)
- 1 sensor BMP280 (pressão)
- 1 sensor LDR(luminosidade)
- 1 Display LCD
- 1 resistor de 100 kOhms
- 1 resistor de 330 Ohms

## Método de calibração:

- Primeiramente calcular as incertezas:

Incerteza padrão =

$$\sqrt{(Incerteza\ tipo\ A)^2 + (Incerteza\ tipo\ B)^2 + (Incerteza\ tipo\ B)^2} - (\text{fórmula 1})$$

Incerteza do tipo B = média das medições · Precisão + Resolução- (fórmula 2)

$$Incerteza\ tipo\ A = \frac{desvio\ padrão}{\sqrt{número\ de\ medições}} - (\text{fórmula 3})$$

•Método dos Mínimos Quadrados:

$$\begin{aligned}
 S_\sigma &= \sum_{i=1}^n \frac{1}{\sigma_i^2} \\
 S_x &= \sum_{i=1}^n \frac{x_i}{\sigma_i^2} \\
 S_{x^2} &= \sum_{i=1}^n \frac{x_i^2}{\sigma_i^2} \\
 S_y &= \sum_{i=1}^n \frac{y_i}{\sigma_i^2} \\
 S_{xy} &= \sum_{i=1}^n \frac{x_i y_i}{\sigma_i^2}
 \end{aligned}
 \quad \left. \right\} \quad
 \begin{aligned}
 f(x) &= ax + b \\
 a &= \frac{1}{\Delta} (S_\sigma S_{xy} - S_x S_y) \\
 b &= \frac{1}{\Delta} (S_{x^2} S_y - S_x S_{xy}) \\
 \Delta &= (S_\sigma S_{x^2} - S_x S_x)
 \end{aligned}$$

Nesta etapa, calcula-se consecutivamente: As somatórias, a variação (delta), coeficiente angular e termo independente da função.

## Como as equações calibradas foram encontradas para cada sensor:

### Sensor LM35

#### Calibração:

- O objetivo foi de achar uma relação linear entre a tensão lida pelo arduino e a temperatura apresentada pelo TMP-36. No entanto, o equipamento TMP-36, assim como o arduino tem uma incerteza B relacionada a ele. Então foi necessário também calibrar o valor da temperatura do sensor.
- **Necessidade de encontrar o valor da incerteza padrão:**
  - Incertezas tipo b: já fornecidas.
  - Incerteza tipo a: a calcular.
- Para isso, como mostra a tabela a seguir, seis temperaturas foram expostas ao LM35 que realizou seis medições em cada. Suas voltagens equivalentes apresentadas na figura abaixo foram conseguidas através do arduino.

Medições da Temperatura						
T (°C)	LM35(V)					
4,87	0,0495	0,0505	0,0516	0,0462	0,0495	0,0473
15,26	0,1516	0,1538	0,1548	0,1484	0,1538	0,1527
25,18	0,2538	0,2538	0,2495	0,2516	0,2527	0,2484
36,78	0,3688	0,3656	0,3677	0,3667	0,3645	0,3699
45,48	0,4570	0,4581	0,4548	0,4570	0,4548	0,4559
53,30	0,5355	0,5333	0,5366	0,5355	0,5333	0,5376

- A seguir, uma outra tabela com os valores até agora obtidos.

Média	Desv. padrão	$\sigma_A$	$\sigma_B$ (LM35)	$\sigma_B$ (ARDUINO)	$\sigma$ Padrão
0,0491	0,00201	0,00082	0,00500	0,00215	0,00550
0,1525	0,00229	0,00094	0,00500	0,00215	0,00552
0,2516	0,00226	0,00092	0,00500	0,00215	0,00552
0,3672	0,00201	0,00082	0,00500	0,00215	0,00550
0,4563	0,00133	0,00054	0,00500	0,00215	0,00547
0,5353	0,00174	0,00071	0,00500	0,00215	0,00549

- Finalmente, invertendo a função obtida por meio do MMQ, chegou-se a função desejada da temperatura pela voltagem, expressa pela equação a seguir, calibrada e utilizada no código do arduino:  $Y = 99,62571 * x + 0,005754$

## Sensor DHT22

### Calibração:

- Nosso intuito com essa calibração foi calcular uma relação linear entre a umidade lida pelo sensor e a umidade de referência descrita na tabela a seguir.
- Importante esclarecer que a incerteza do sensor DHT22 já é sabida: 2%

		Umidade (%)			
Solução salina saturada <sup>(1)</sup>		5°C	15°C	20°C	25°C
Hidróxido de potássio	KOH	13,0	9,0	~	8,0
Acetato de potássio	KAc	24,8	23,5	23,0	23,0
Cloreto de magnésio	MgCl	34,0	33,5	33,0	32,5
Nitrito de cálcio	Ca(NO <sub>3</sub> ) <sub>2</sub>	61,0	58,0	56,0	52,2
Cloreto de sódio	NaCl	76,0	75,5	75,3	75,1
Cloreto de potássio	KCl	87,8	86,0	85,3	85,0

- Como apresentado na imagem acima, para fazer a calibração foram utilizados 6 sais. Através do programa “calibra DHT22” presente no excel, 5 medidas de umidade e temperatura foram feitas para cada sal. Fazendo o cálculo da incerteza A e incerteza padrão da umidade lida, os resultados estão apresentados na figura abaixo(a tabela está separada em duas imagens devido ao seu tamanho comparado com a dimensão da página):

Sol. Salina	Referência			Medidas			
	w	x	y1	y2	y3	y4	y5
T (°C)	Ur (%)	U1 (%)	U2 (%)	U3 (%)	U4(%)	U5 (%)	
KAc	25	23	28,1	27	24,1	24	26,3
Ca(NO <sub>3</sub> ) <sub>2</sub>	23,8	53,112	66,4	65,2	61,7	61,9	64,2
KCl	22,8	85,132	82,8	84,4	89,2	89,3	86,4
KOH	25	8	8,2	7,7	6,6	6,7	7,3
MgCl	22,2	29,25	33,4	33,9	34,2	34,4	34
NaCl	23,2	69,69	71	73	78,6	78,7	74,4

y4	Média das Med.		Desvio Padrão	Incerteza do Tipo A	Incerteza do Tipo B	Incerteza Padrão	
	U4(%)	U5 (%)	Um (%)	σ (%)	σA (%)	σB (%)	σPm (%)
24	26,3	25,9	1,8069311	0,808084154	2,0	2,157081361	
61,9	64,2	63,88	2,05353354	0,918368118	2,0	2,200772592	
89,3	86,4	86,42	2,881319142	1,288565093	2,0	2,379159515	
6,7	7,3	7,3	0,674536878	0,301662063	2,0	2,022622061	
34,4	34	33,98	0,376828874	0,168522995	2,0	2,007087442	
78,7	74,4	75,14	3,424616767	1,531535178	2,0	2,519047439	

- O próximo passo foi de transferir a incerteza da temperatura para a umidade de referência, para isso bastou fazer o cálculo de tal transferência, resultando na tabela a seguir:

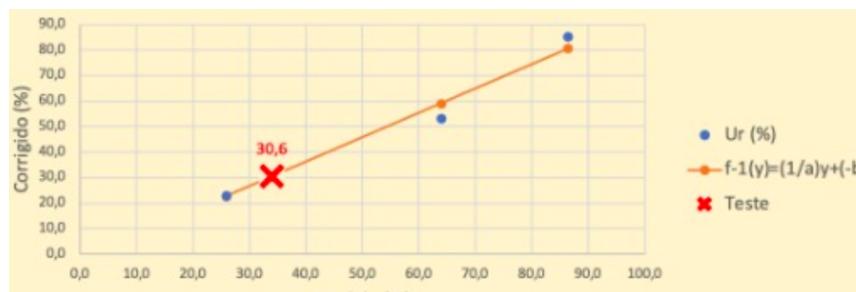
Sol. Salina	Referência		Incerteza da Temperatura da inc. da temp.		Transferência
	w	x	σT (°C)	σw	para x
T (°C)	Ur (%)	σT (°C)	σw	σw-->x	σr (%)
KAc	25	23	0,5	0	
Ca(NO <sub>3</sub> ) <sub>2</sub>	23,8	53,112	0,5	-0,38	
KCl	22,8	85,132	0,5	-0,03	
KOH	25	8	0,5	-0,05	
MgCl	22,2	29,25	0,5	-0,05	
NaCl	23,2	69,69	0,5	-0,02	

- Agora, a transferência da incerteza da umidade de referência para a umidade lida. Nesta etapa, o coeficiente angular é desconhecido, logo fizemos um "chute" de um valor e o 1 foi escolhido por normalmente estar perto de coeficientes angulares similares.

Sol. Salina	Referência		Transf. da inc. da inc. da temp. da temp. para x		Transf. da inc. da temp. para y (inclinação 1/1)
	w	x	σr (%)	σw-->x	σw-->y
T (°C)	Ur (%)	σT (°C)	σw	σw-->x	σw-->y
KAc	22,4	23	0	0	0
Ca(NO <sub>3</sub> ) <sub>2</sub>	22,4	53,112	-0,38	-0,38	-0,38
NaCl	23,2	69,69	-0,02	-0,02	-0,02
KOH	25	8	-0,05	-0,05	-0,05
MgCl	22,2	29,25	-0,05	-0,05	-0,05
KCl	22,4	85,132	-0,03	-0,03	-0,03
				Chute	1

- A partir da incerteza final obtida e invertendo a função resultante do MMQ, a equação calibrada e seu respectivo gráfico são:

Coeficiente	Coeficiente	Entra Medida	Sai Referência	Sai Curva de Cal. Final
		y	x	$f^{-1}(y) = (1/a)y + (-b/a)$
1/a	-b/a	Um (%)	Ur (%)	$f^{-1}(y) = (1/a)y + (-b/a)$
0,954248905	-1,870287298	25,9	23,0	22,8
		63,9	53,1	59,1
		86,4	85,1	80,6
<b>Teste</b>		<b>34</b>		<b>30,6</b>



Equação calibrada:

$$\text{Umidade} = 0,954 \times \text{medida} + (-1,870)$$

## Sensor LDR

### Calibração:

- O LDR é um sensor de luminosidade, que varia a resistência, conforme a intensidade da luz captada. A resistência do sensor varia de  $100\Omega$  a  $1M\Omega$ .

Foi necessário fazer um divisor de tensão para possibilitar a medição da luminosidade. Foi utilizado um resistor de  $100\text{ k}\Omega$  em série com o sensor, para se obter uma tensão dividida devida.. A função que relaciona a luminosidade pela resistência é:

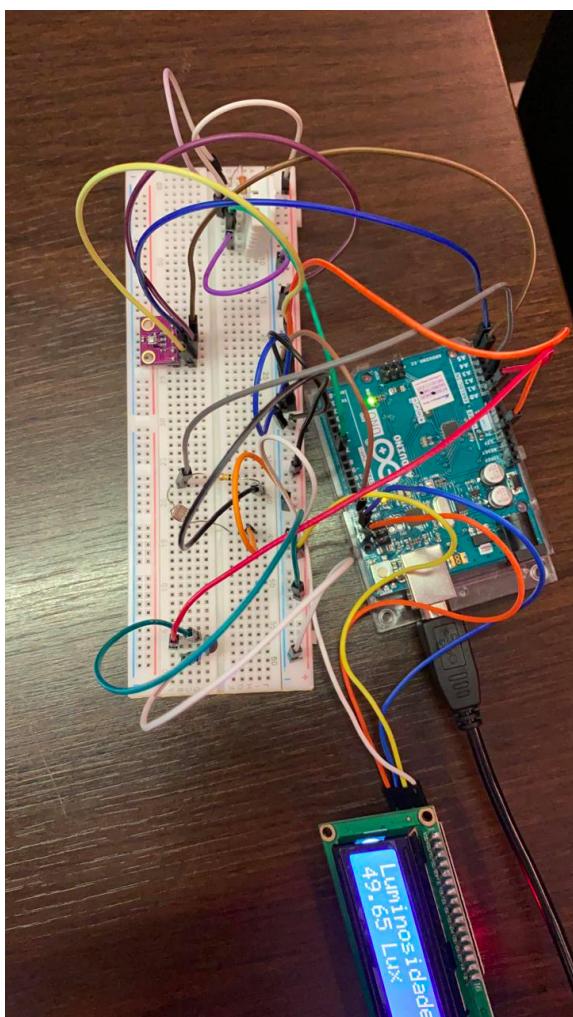
$$L = \text{pow}(10, 6.5 - 1.25 \cdot \log(R))$$

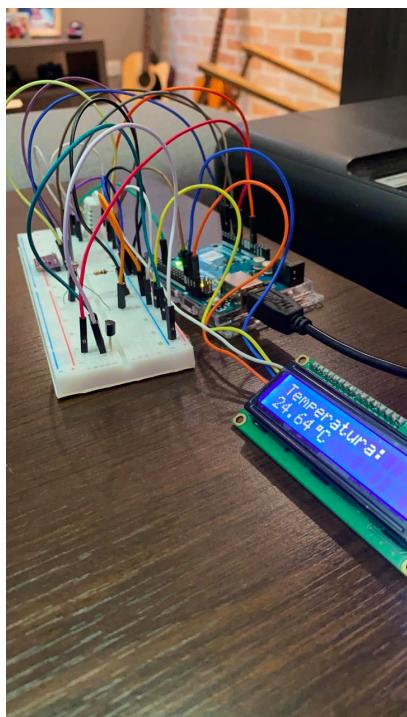
## Sensor BMP280

### Calibração:

- Este sensor ja e previamente calibrado.
- Constantes e equações previamente prontas e presentes na biblioteca "Adafruit BMP280" importada direta ao código.

O circuito completo final e o código utilizado no arduino:





```
// Incluindo as bibliotecas necessárias
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_BMP280.h>
#include <LiquidCrystal_I2C.h>

// Definindo constantes do código
LiquidCrystal_I2C lcd(0x27, 16, 2);
Adafruit_BMP280 bmp;
const int DHT_pino = 7;
DHT dht(DHT_pino, DHT22);
uint8_t SimboloGrau[] = {0x07,0x05,0x07,0x00,0x00,0x00,0x00,0x00};
uint8_t SimboloOhm[] = {0x00,0x00,0x0E,0x0A,0x1B,0x00,0x00,0x00};

float Temperatura, umid;
int le_lum; float tensao, res, Luminosidade;
```

```

// Criando função que mostra os nomes dos integrantes, série e o nome do projeto.
void NomesTurmaProjeto(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Felipe Catapano");
    lcd.setCursor(0, 1);
    lcd.print("Thomas Chabro");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enzo Cunha");
    lcd.setCursor(0, 1);
    lcd.print("Felipe Drummond");
    delay(2000);
    lcd.clear();
    lcd.print("TURMA B");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Projeto Estacao");
    lcd.setCursor(0, 1);
    lcd.print("Metereologica!");
    delay(2000);
}

void setup() {
    analogReference(INTERNAL);
    dht.begin();
    bmp.begin(0x76);
    lcd.init();
    lcd.backlight();
    lcd.createChar(0, SimboloGrau);
    lcd.createChar(1, SimboloOhm);
    NomesTurmaProjeto();
}

void loop() {
    // Medindo a temperatura (Sensor LM35)
    int bin = analogRead(A0);
    // Convertendo o valor lido para uma tensão
    float ddp = (bin*1.1)/1023.0;
    // Calculando a temperatura junto com sua calibração
    Temperatura = ddp * 99.62571869 + 0.0057542347;
    // Lendo a temperatura no display LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print ("Temperatura: ");
    lcd.setCursor (0, 1);
    lcd.print (Temperatura);
    lcd.write(0);
    lcd.print("C");
    delay(2000);
}

```

```

// Medindo a umidade (Sensor DHT22)
float le_umid = dht.readHumidity();
umid = le_umid*0.954248905 -1.870287298;
// Lendo a umidade no display LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print ("Umidade: ");
lcd.setCursor (0, 1);
lcd.print (umid);
lcd.print(" %");
delay(2000);

// Medindo a luminosidade
le_lum = analogRead(A1);
tensao = (le_lum/1023.0)*1.1;
res = tensao*100000.0/(5.0 - tensao);
Luminosidade = pow(10, 6.5 - 1.25*log10(res));
// Lendo a luminosidade no display LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print ("Luminosidade: ");
lcd.setCursor (0, 1);
lcd.print(Luminosidade);
lcd.print(" Lux");
delay(2000);

// Medindo a luminosidade
le_lum = analogRead(A1);
tensao = (le_lum/1023.0)*1.1;
res = tensao*100000.0/(5.0 - tensao);
Luminosidade = pow(10, 6.5 - 1.25*log10(res));
// Lendo a luminosidade no display LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print ("Luminosidade: ");
lcd.setCursor (0, 1);
lcd.print(Luminosidade);
lcd.print(" Lux");
delay(2000);

// Medindo a pressão
float press = (bmp.readPressure())/100.0;
// Lendo a luminosidade no display LCD
lcd.clear();
lcd.setCursor (0, 0);
lcd.print("Pressao: ");
lcd.setCursor (0, 1);
lcd.print (press);
lcd.print(" hPa");
delay(2000);
}

```

