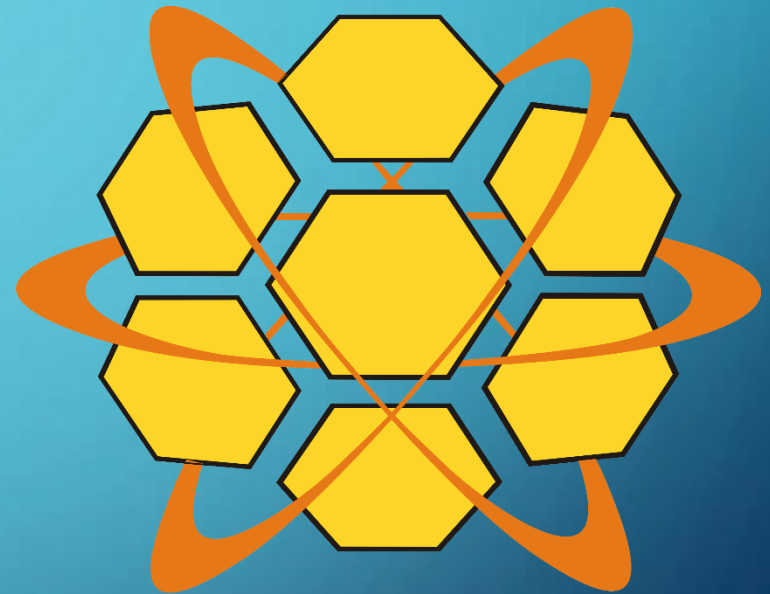


# MINICURSO ARDUINO AVANÇADO

Ministrantes:

- Henrique Wippel Parucker da Silva
- Murilo de Oliveira do Nascimento
- Ícaro Cristofolini



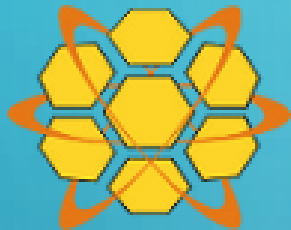
## COLMÉIA



**UDESC**  
UNIVERSIDADE  
DO ESTADO DE  
SANTA CATARINA

JOINVILLE  
CENTRO DE CIÊNCIAS  
TECNOLÓGICAS

# COLMÉIA



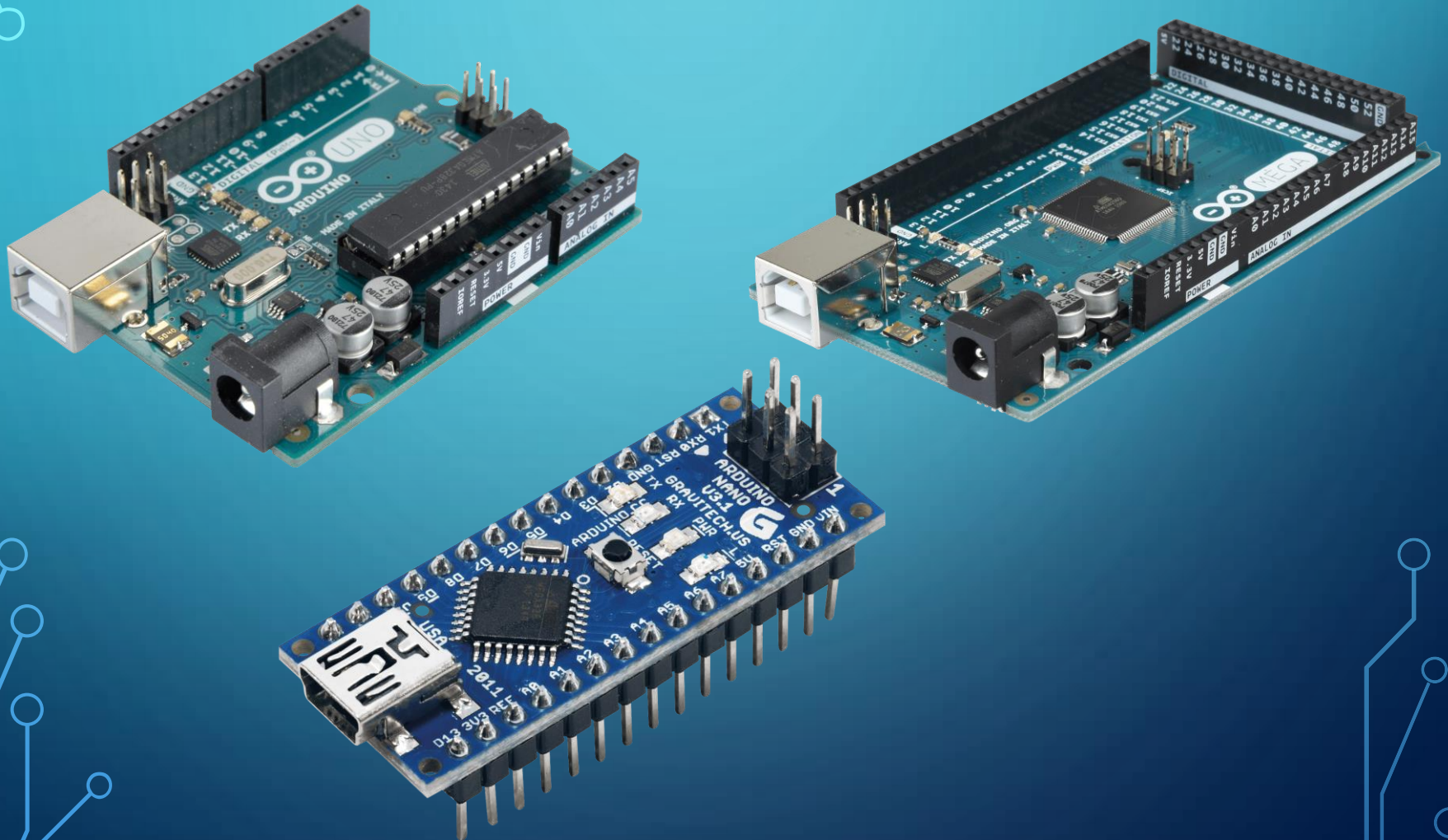
# COLMÉIA

Grupo de Pesquisa em Software Livre

Quem somos?

O que fazemos?

# PLATAFORMA DE PROTOTIPAGEM ARDUINO



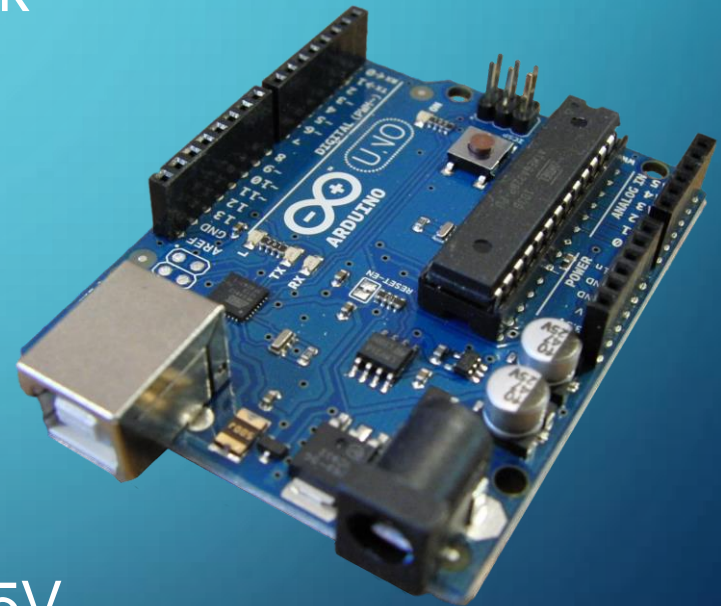
# ARDUINO

- Projeto iniciado em 2005.
- Alternativa barata e simples de usar.
- Conjunto de ferramentas para desenvolvimento de aplicações diversas.
- Diversos modelos (universal).
- Hardware Livre.



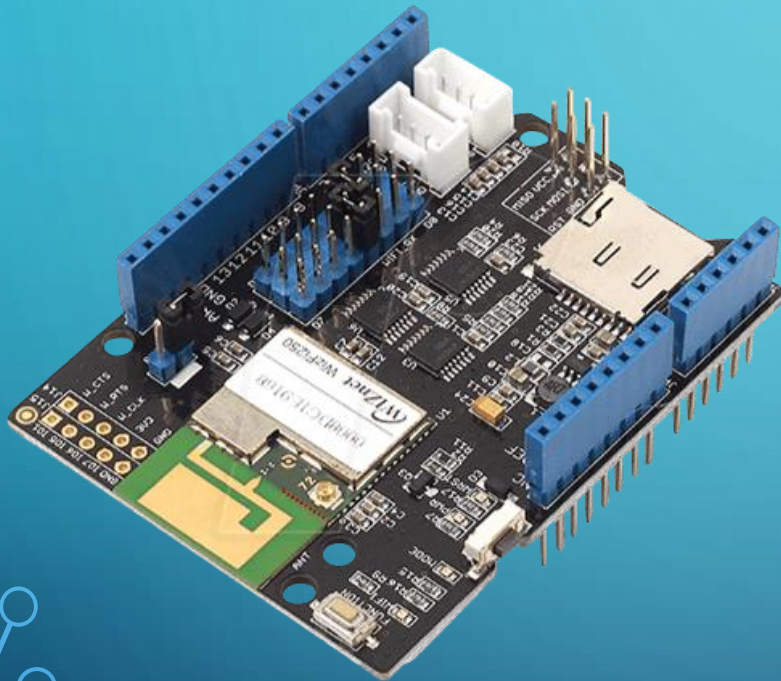
# ARDUINO UNO

- Controlador: ATmega328 (32k flash)
- 14 pinos digitais
  - 6 pinos PWM
- 6 pinos de entrada analógica
- 2 pinos alimentação
- 1 pino reset
- Entrada USB e alimentação 5V





# SHIELDS

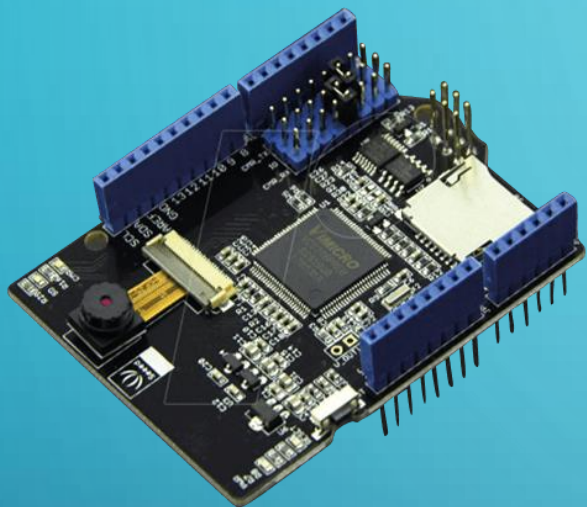


Wifi

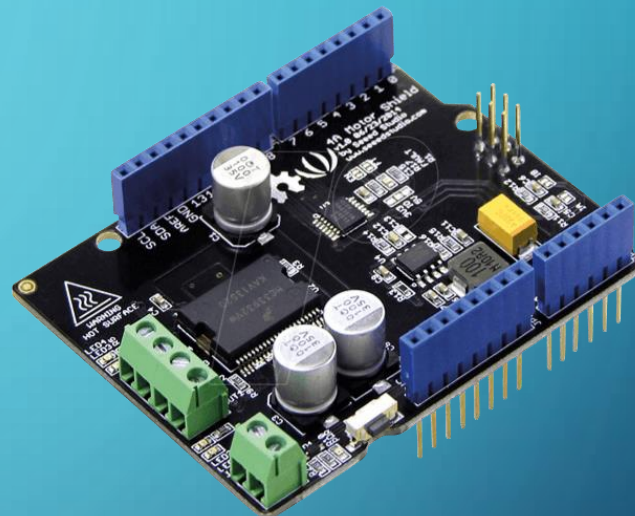


Bluetooth

# SHIELDS



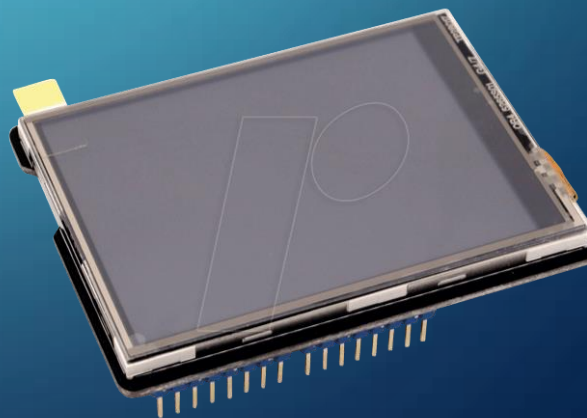
Câmera 640x480



Motor



40 LEDs



Touch Screen

# PROJETO 1 - DHT-11 SENSOR DE TEMPERATURA E UMIDADE

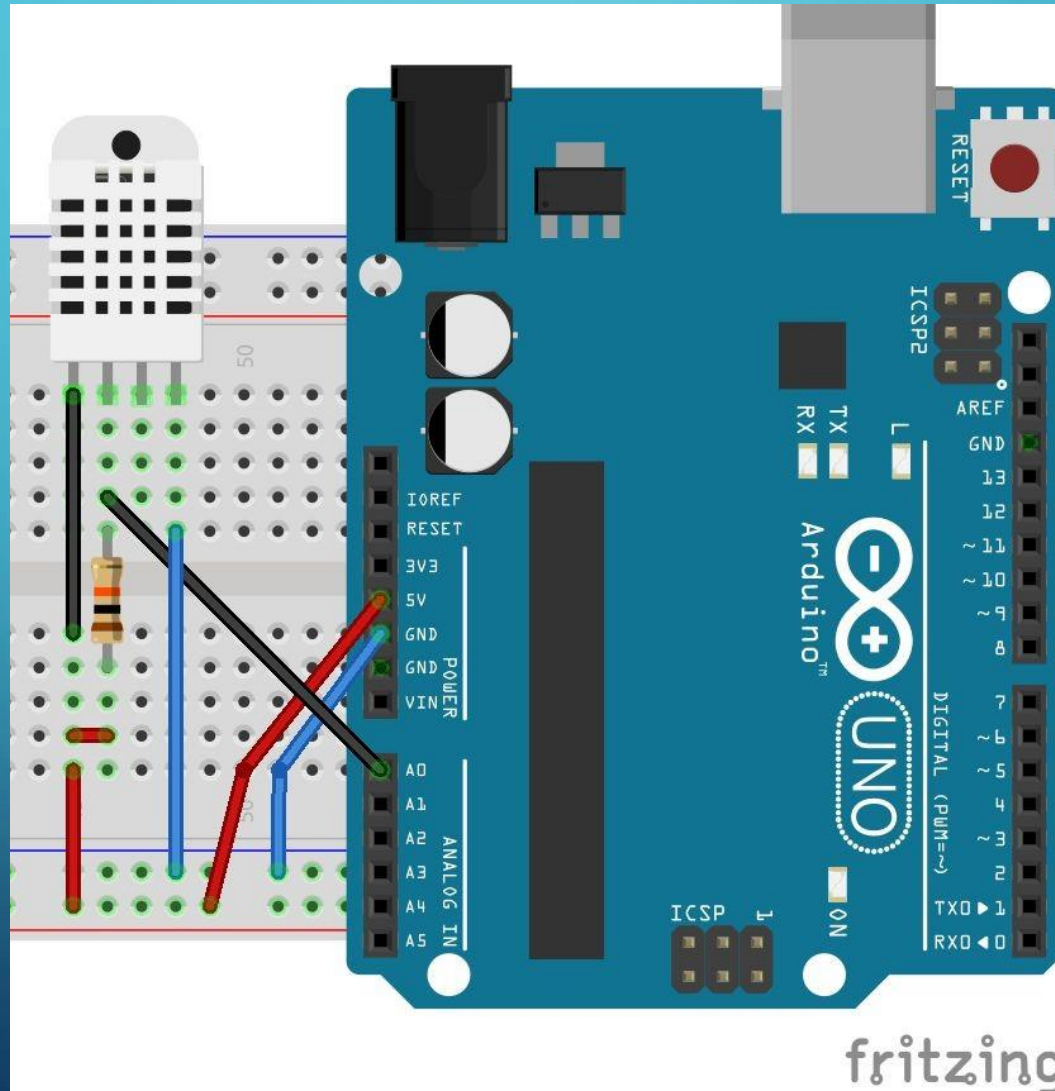
O sensor da linha DHT faz a leitura de temperatura e umidade através de uma saída digital calibrada.

O sistema de interface é feito no modo one wire, tornando a ligação mais simples.

Temperatura de 0 a 50°C, erro de  $\pm 2^\circ\text{C}$ ,  
Umidade: 20 a 90% RH, erro de  $\pm 5\%$ ;



# PROJETO 1 - DHT-11 SENSOR DE TEMPERATURA E UMIDADE



# PROJETO 1 - DHT-11 SENSOR DE TEMPERATURA E UMIDADE

```
#include "DHT.h"
```

```
#define DHTPIN 7 // pino que estamos conectado
```

```
#define DHTTYPE DHT11 // DHT 11
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup(){  
  Serial.begin(9600);  
  dht.begin();  
}
```

# PROJETO 1 - DHT-11 SENSOR DE TEMPERATURA E UMIDADE

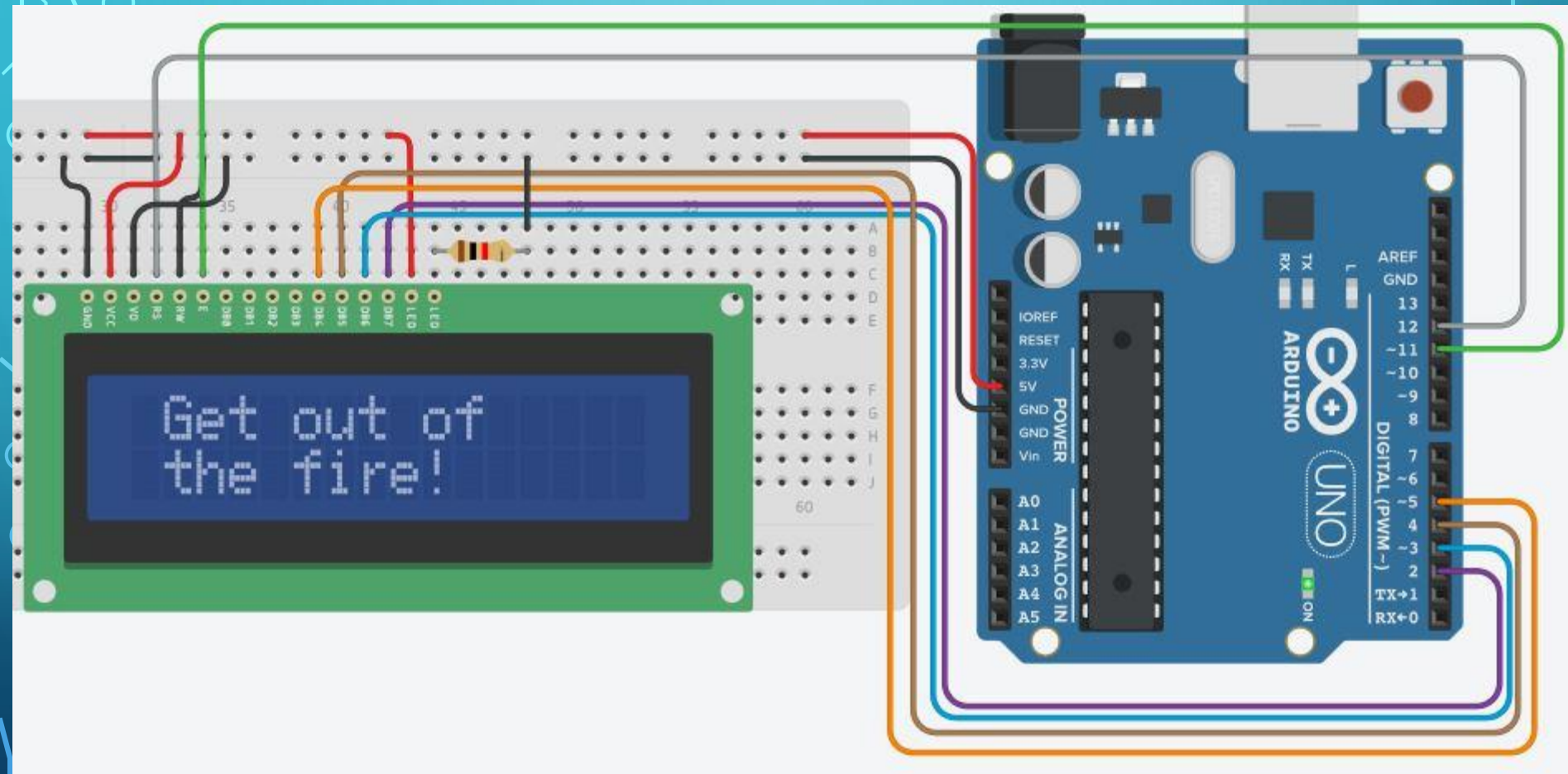
```
void loop(){  
  // A leitura da temperatura e umidade pode levar 250ms!  
  // O atraso do sensor pode chegar a 2 segundos.  
  float h = dht.readHumidity();  
  float t = dht.readTemperature();  
  // testa se retorno é valido, caso contrário algo está errado.  
  if (isnan(t) || isnan(h)) {  
    Serial.println("Failed to read from DHT");  
  } else{  
    Serial.print("Umidade: ");  
    Serial.print(h);  
    Serial.println(" %t");  
    Serial.print("Temperatura: ");  
    Serial.print(t);  
    Serial.println(" *C");  
  }  
  delay(2500);  
}
```

## PROJETO 2 - TELA LCD

Um diferencial em qualquer projeto microcontrolado é utilização de um display LCD para indicação de parâmetros e informações diversas.

O modelo utilizado neste material é o JHD 162A, possui backlight com ajuste de luminosidade e contraste dos caracteres controláveis. É fácil de encontrá-lo no mercado por ter o preço mais acessível.

# PROJETO 2 - TELA LCD





# PROJETO 2 - TELA LCD

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //Configura a tela LCD com os pinos conectados
int l = 0, c = 0, t = 0;
char frase[2][17] = {"  Colmeia", "  UDESC"}; //tamanho máximo da frase 16

void setup() {
  lcd.begin(16, 2);
  Serial.begin(9600);
}
```

# PROJETO 2 - TELA LCD

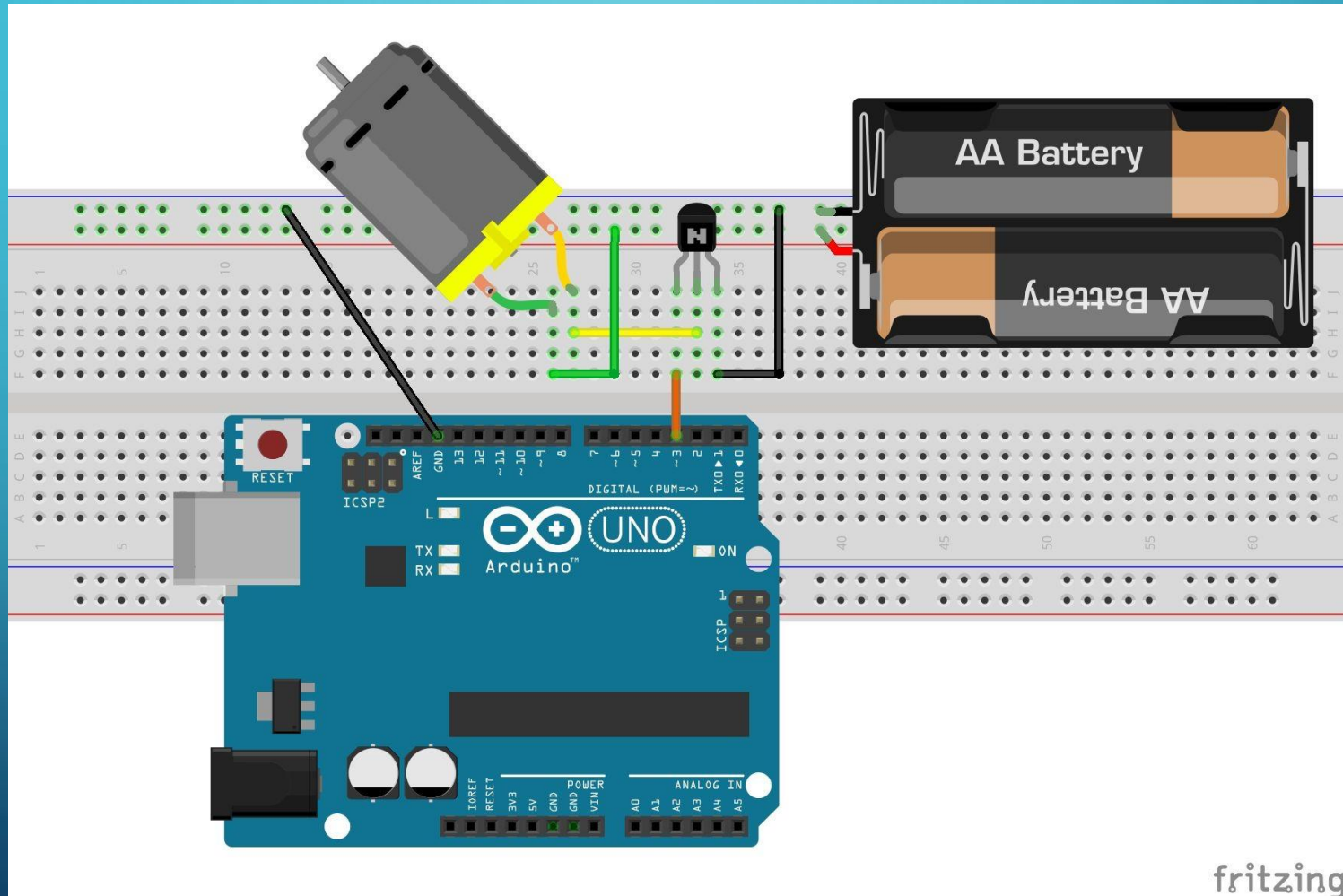
```
void loop() {  
  Serial.println(frase[l]);  
  lcd.setCursor(c, l);  
  lcd.print(frase[l]);  
  delay(1000);
```

```
  if(l){  
    l = 0;  
    lcd.clear();  
  }  
  else  
    l = 1;  
  delay(1000);  
}
```

# PROJETO 3 - MOTOR DC

- Para controlar componentes que requerem uma corrente elétrica mais alta, como um motor, ou para controlar dispositivos em circuitos externos, é possível utilizar relés e transistores.
- Para controlar o motor será usado o transistor TIP122.

# PROJETO 3 - MOTOR DC



# PROJETO 3 - MOTOR DC

```
int MOTOR= 3;
```

```
void setup(){
```

```
    pinMode(MOTOR, OUTPUT);
```

```
    pinMode(13, OUTPUT); //só para o led embutido piscar junto
```

```
}
```



# PROJETO 3 - MOTOR DC

```
void loop(){  
    digitalWrite(13,HIGH);  
    digitalWrite(MOTOR, HIGH);  
    delay(3000);  
    digitalWrite(13,LOW);  
    digitalWrite(MOTOR, LOW);  
    delay(3000);  
}
```

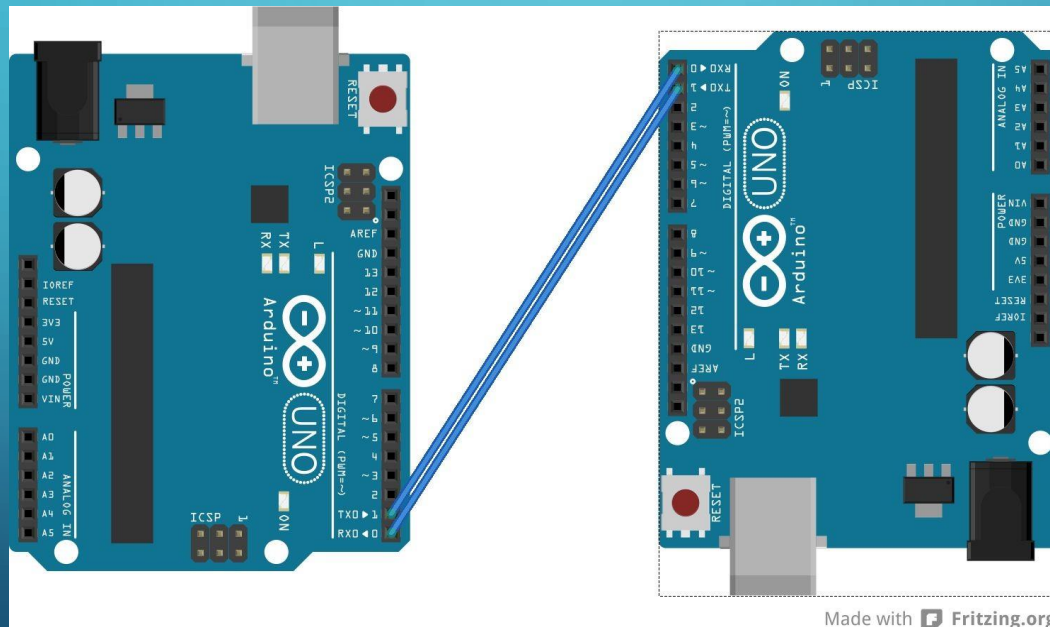
# COMUNICAÇÃO DE DADOS

Quando falamos de dispositivos programáveis é importante salientar a utilização da comunicação de dados, pois podemos realizar inúmeras aplicações com as mesmas. Esta comunicação na maioria das vezes ocorre em um meio de transmissão comum, podendo ser o ar ou um fio qualquer.

Para iniciarmos esta ideia utilizaremos o padrão de comunicação mais simples entre o Arduíno que é a comunicação RX/TX.

# COMUNICAÇÃO RX/TX

Esta comunicação consiste em ligarmos dois Arduínos utilizando dois cabos e ligando os mesmos da seguinte forma representada pela figura abaixo:



# COMUNICAÇÃO RX/TX

## Código receptor

```
void setup(){  
  Serial.begin(9600);  
}  
  
void loop(){  
  if(Serial.available()){  
    Serial.print("Valor Recebido:");  
    Serial.println(Serial.read());  
    delay(500);  
  }else{  
    Serial.println("Aguardando Recebimento.");  
    delay(500);  
  }  
}
```

# COMUNICAÇÃO RX/TX

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

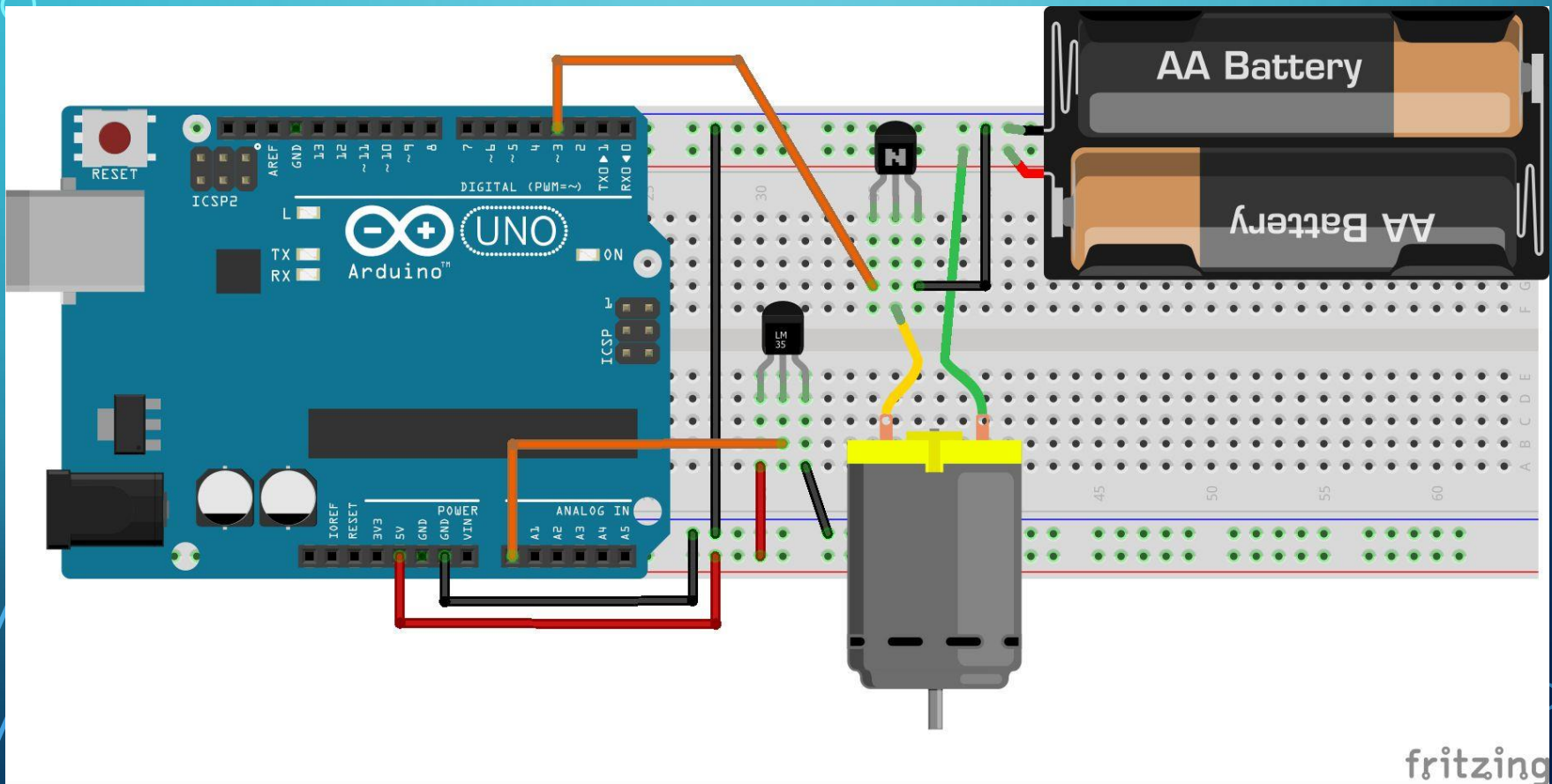
```
void loop(){  
  Serial.write(100);  
  delay(500);  
}
```



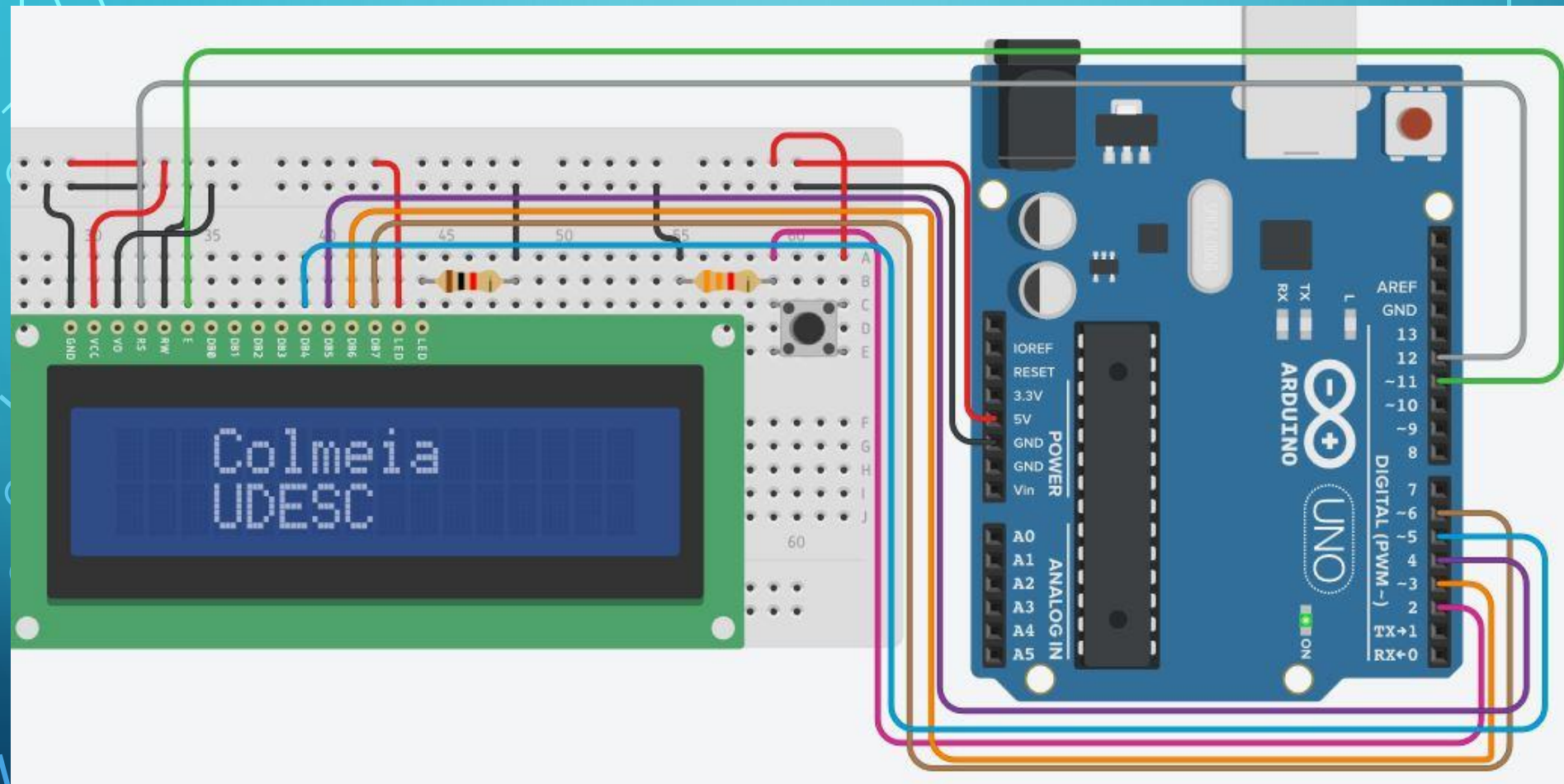
# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA

- Um arduino controla o motor e lê a temperatura.
- Um arduino recebe o comando de desligar e ligar o motor por um botão e mostra em uma tela LCD o estado do motor e a temperatura.

# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA



# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA



# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA

## Código Emissor

```
#include<LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 6);
```

```
int t = 0, motorLigado = 0;
```

```
const int BOTAO = 2;
```

```
const char FraseMotorEstado[2][17] = {"Motor: ON", "Motor:  
OFF"}; //tamanho maximo da frase 16
```

```
const char FraseTemperatura[] = "Temp: ";
```

```
float temperatura = 0.0;
```

# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA

## Código Emissor

```
void setup() {  
  attachInterrupt(0, iniciaMotor, FALLING);  
  lcd.begin(16, 2);  
  pinMode(BOTAO, INPUT);  
  Serial.begin(9600);  
}
```

```
void iniciaMotor(){  
  motorLigado = !motorLigado;  
  Serial.write(motorLigado);  
}
```



# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA

## Código Emissor

```
void atualizaTemperatura(){  
    int tempread = Serial.read();  
    temperatura = (tempread * 5) * 10.0 / 1024;  
}  
  
void loop() {  
  
    if(Serial.available()){  
        atualizaTemperatura();  
    }  
}
```

# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA

## Código Emissor

```
lcd.setCursor(0, 0);  
lcd.print(FraseMotorEstado[!motorLigado]);  
lcd.setCursor(0, 1);  
lcd.print(FraseTemperatura);  
lcd.setCursor(6, 1);  
lcd.print(temperatura);  
lcd.setCursor(13, 1);  
lcd.print(" °C");  
delay(500);  
lcd.clear();  
}
```

# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA

## Código Receptor

```
const int MOTOR= 3, LED = 4, LM35 = A0;  
int temperatura;  
int estadoMotor = 0;
```

```
void setup(){  
  pinMode(MOTOR, OUTPUT);  
  pinMode(LED, OUTPUT);  
  pinMode(LM35, INPUT);  
  Serial.begin(9600);  
  
}
```

# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA

## Código Receptor

```
void ligaMotor(){  
    estadoMotor = 1;  
    digitalWrite(LED, HIGH);  
    digitalWrite(MOTOR, HIGH);  
}
```

```
void desligaMotor(){  
    estadoMotor = 0;  
    digitalWrite(LED, LOW);  
    digitalWrite(MOTOR, LOW);  
}
```

# PROJETO 4 - COMUNICAÇÃO RX/TX APLICADA

## Código Receptor

```
void leituraTemperatura(){
    temperatura = analogRead(LM35);
    Serial.write(temperatura);
}
void loop(){
    if(Serial.available()){
        int leitura = Serial.read();
        if(leitura == 1){
            ligaMotor();
        }else if(leitura == 0){
            desligaMotor();
        }
    }
    leituraTemperatura();
}
```

# REFERÊNCIAS

SADIKU, Matthew N. O. “Fundamentos de circuito elétricos”. Local: Mcgraw Hill Brasil técnicos 3a Ed.; 2008.

IRWIN, J.D.; "Análise de Circuitos em Engenharia". Makron Books do Brasil Editora LTDA; 4a Ed.; 2000.

Site Arduino: <http://www.arduino.cc/>

Software apra desenho de circuitos: <http://fritzing.org/>



# PERGUNTAS?

Site : [www.colmeia.udesc.br](http://www.colmeia.udesc.br)

E-mail Murilo: [murilo.on@hotmail.com](mailto:murilo.on@hotmail.com)

E-mail Ícaro: [icaroaj@hotmail.com](mailto:icaroaj@hotmail.com)

E-mail Henrique: [henriquewps@gmail.com](mailto:henriquewps@gmail.com)

## Obrigado!