

Internet das Coisas

Aula 14 – Leitura de sensores analógicos





Leitura de sensores analógicos

Os sensores analógicos, como já comentado anteriormente, conseguem produzir valores quaisquer de tensão entre 0V e 5V. Essa tensão, quando recebida pela porta analógica, é então transformada em um valor entre 0 e 1023. As vezes, o uso direto desse valor sem a conversão em uma grandeza específica nos é o suficiente. Outras vezes, necessitamos convertê-lo em uma grandeza que seja mais útil para nós.

Leitura de sensores analógicos

Independentemente do caso, a leitura é simples e necessita apenas do método `analogRead` apresentado abaixo, indicando a porta analógica desejada e recebendo de volta a leitura em forma de um valor entre 0 e 1023.

```
analogRead( <porta> );
```

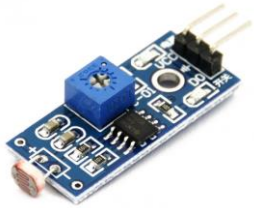
```
<porta>
```

```
porta à qual o sensor analógico está ligado
```

```
resposta:
```

```
a resposta do método será um valor entre 0 e 1023
```

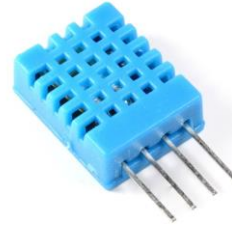
Principais sensores analógicos



sensor de luz (LDR)



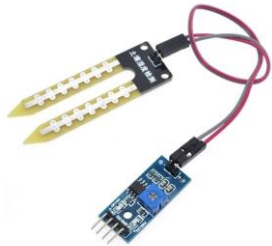
sensor de temperatura



sensor de umidade do ar



sensor de pressão do ar



sensor umidade do solo



sensor de respingos



sensor de PH



sensor óptico-reflexivo

Principais sensores analógicos



sensor de gás



sensor de fumaça



sensor piezoelétrico



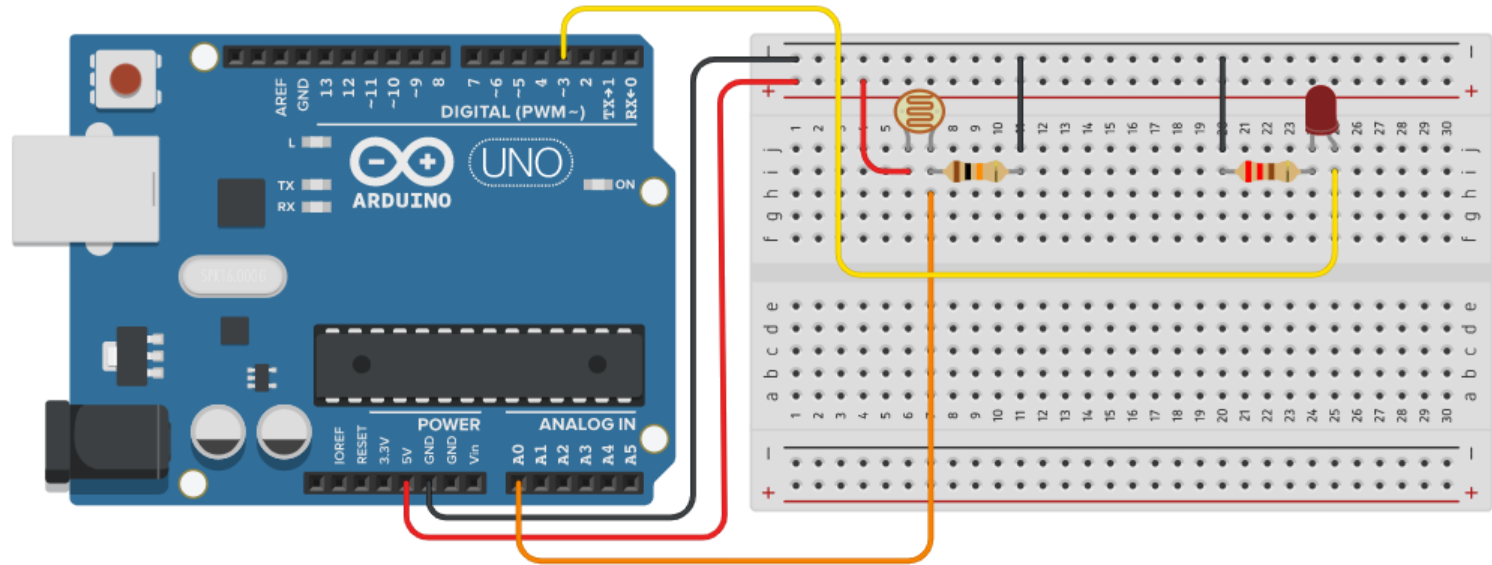
sensor joystick

Exemplo 1: Sensor de luz (LDR)

Exemplo 1: queremos utilizar um sensor LDR para capturar um valor que nos dê ideia da luminosidade do ambiente, acendendo um led quando esta for muito baixa. Para isso, vamos conectar um dos seus pinos em uma porta analógica do Arduino (no exemplo, a porta A0). A montagem de uso de um LDR depende se o mesmo é um sensor avulso (apenas o LDR), tendo neste caso dois pinos, ou se o mesmo foi encapsulado em uma placa, tendo neste caso 3 pinos. A grande diferença é que no segundo caso temos já na placa um resistor de pullup de $10k\Omega$. Caso tenhamos apenas o LDR avulso, necessitamos incluir esse resistor em nosso projeto, em série com o pino de dados e com sua saída no negativo.

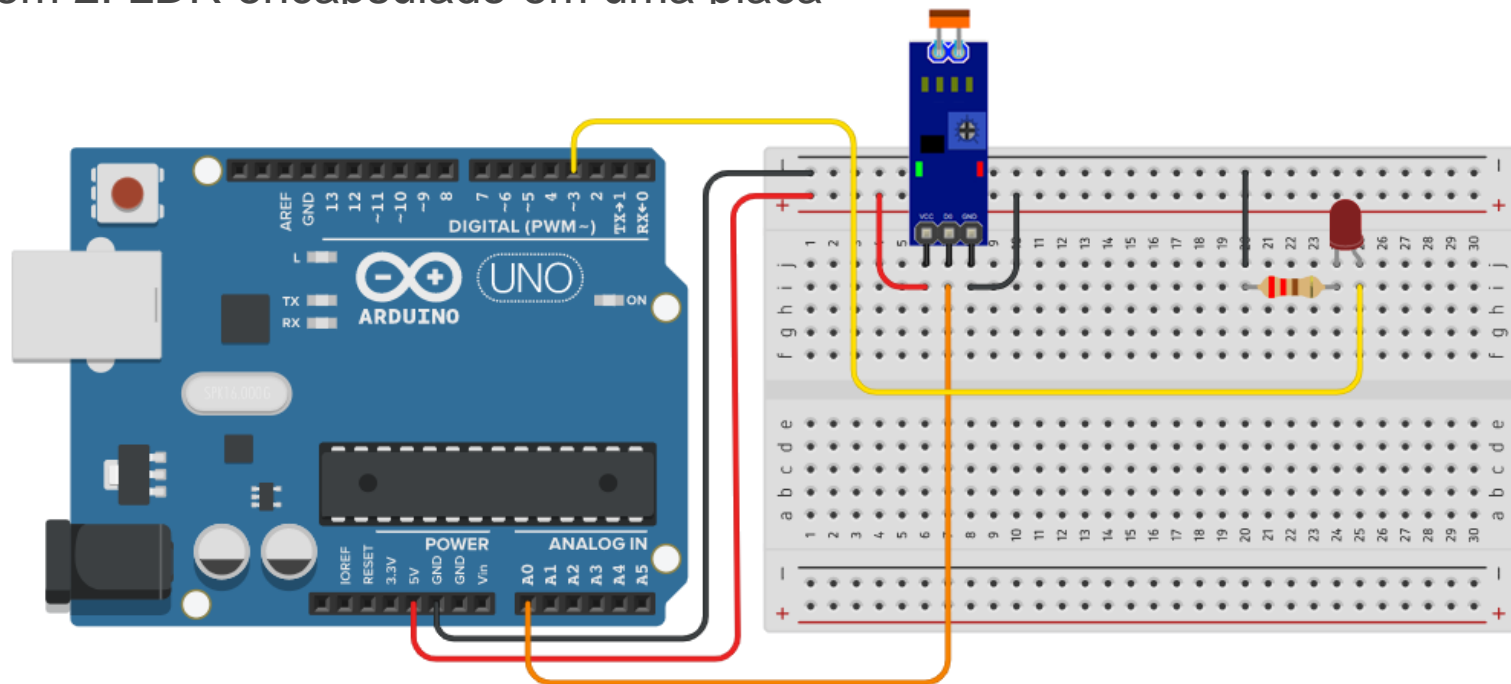
Exemplo 1: Sensor de luz (LDR)

Montagem 1: LDR avulso



Exemplo 1: Sensor de luz (LDR)

Montagem 2: LDR encapsulado em uma placa





Exemplo 1: Sensor de luz (LDR)

Independentemente da montagem, o nosso código é baseado em ler o valor do sensor LDR através da porta A0 à qual está conectado. Após a leitura, verificamos se a luminosidade percebida é muito baixa (vamos considerar um valor abaixo de 100 em nosso exemplo). Quando isso ocorre, o led conectado à porta digital 2 deve ser ligado, senão, deve permanecer desligado. Assim, temos:

Exemplo 1: Sensor de luz (LDR)

```
void setup(){
    pinMode(A0, INPUT);
    pinMode(3, OUTPUT);
    digitalWrite(3, LOW);
    Serial.begin(9600);
}

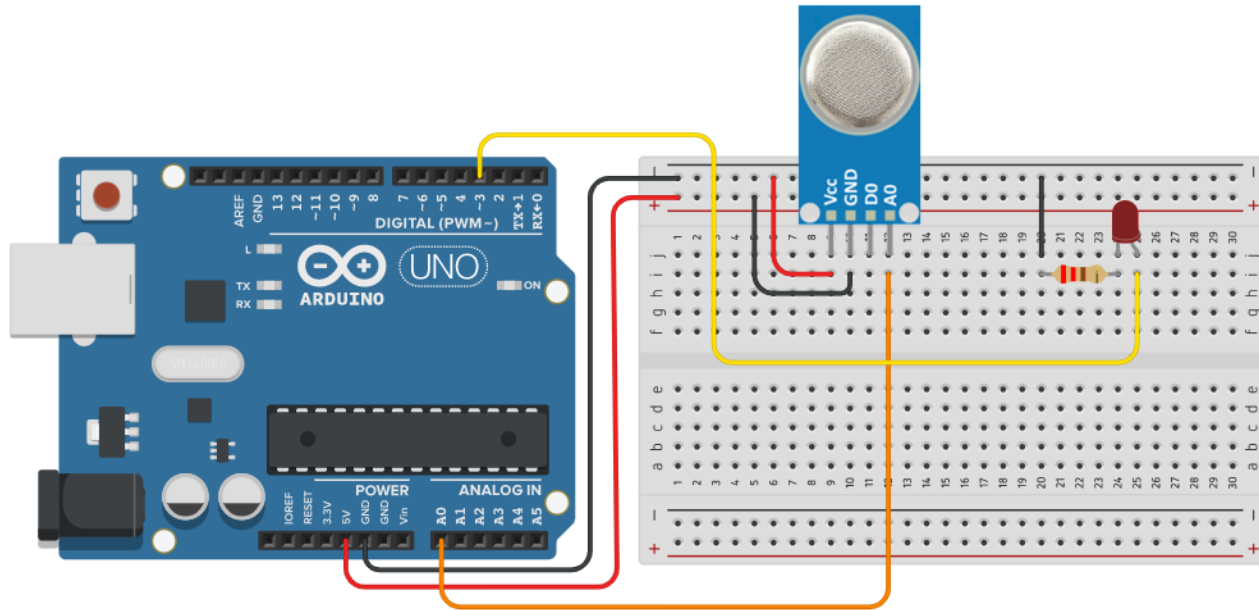
void loop(){
    delay(100);
    int valorLido = analogRead(A0);
    Serial.println("Valor da luminosidade: " + (String)valorLido);

    if(valorLido < 100) {
        digitalWrite(3, HIGH);
    } else {
        digitalWrite(3, LOW);
    }
}
```

Exemplo 2: Sensor de gás

Exemplo 2: queremos utilizar um sensor de gás para detectar um tipo específico de gás (cada sensor é construído especificamente para perceber um tipo), no nosso caso, monóxido de carbono. Este sensor será colocado em um ambiente fechado (uma sala com portas e janelas trancadas e lareira ligada, por exemplo) e ao identificar uma concentração grande desse gás e a possibilidade de envenenamento por ele (que é um envenenamento silencioso), liga um led para avisar. Para isso, iremos ligar o pino de saída (dados) do sensor de gás em uma porta analógica (no nosso caso a A0) e o led em uma porta digital (em nosso caso a 3). Assim, teremos o seguinte circuito eletrônico.

Exemplo 2: Sensor de gás



Exemplo 2: Sensor de gás

O nosso código é baseado em ler o valor do sensor de gás através da porta A0 à qual está conectado. Após a leitura, verificamos se a concentração de monóxido de carbono percebida é muito alta (vamos considerar um valor acima de 400 em nosso exemplo). Quando isso ocorre, o led conectado à porta digital 2 deve ser ligado, senão, deve permanecer desligado. Assim, temos:

Exemplo 2: Sensor de gás

```
void setup(){
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  digitalWrite(3, LOW);
  Serial.begin(9600);
}

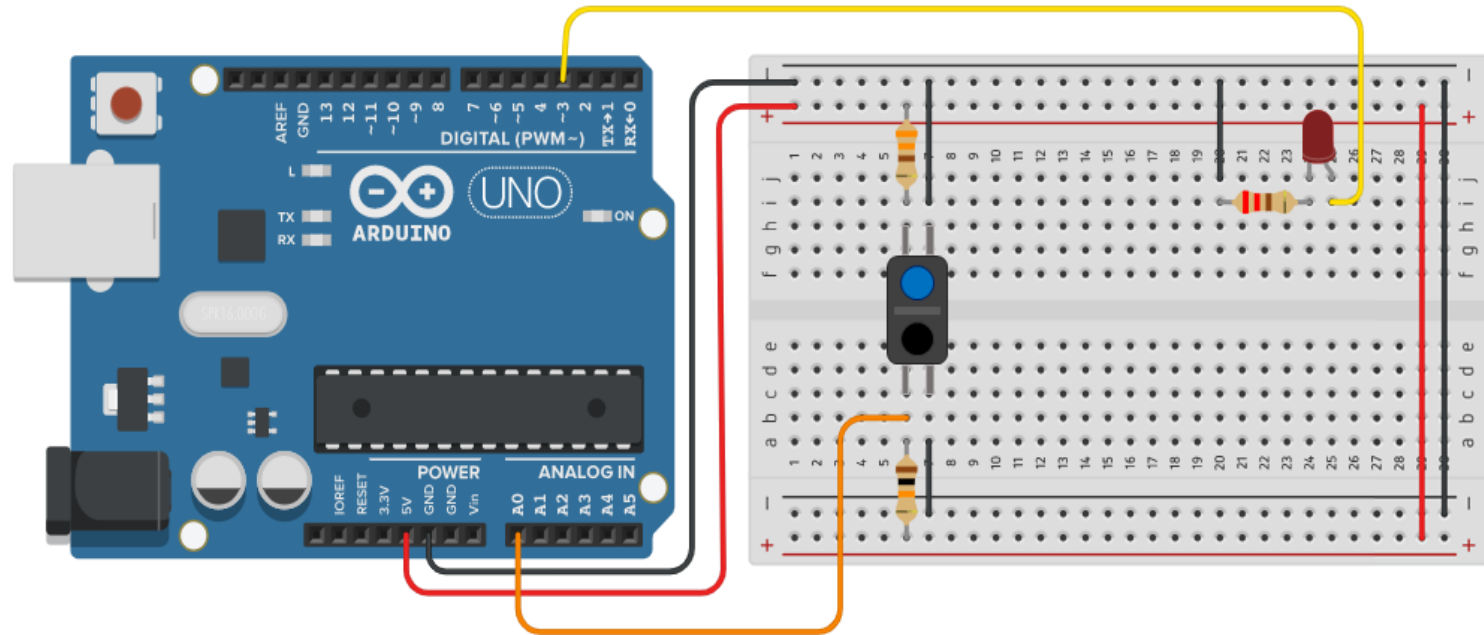
void loop(){
  delay(100);
  int valorLido = analogRead(A0);
  Serial.println("Concentração de monóxido de carbono: " + (String)valorLido);

  if(valorLido > 400) {
    digitalWrite(3, HIGH);
  } else {
    digitalWrite(3, LOW);
  }
}
```

Exemplo 3: Sensor óptico reflexivo

Exemplo 3: queremos utilizar um sensor óptico reflexivo para identificar quando um objeto é aproximado do mesmo. Sensores deste tipo são equipados com 2 elementos: um emissor infravermelho (led azul) e um receptor infravermelho (led preto). O emissor, continuamente envia luz infravermelha que só é percebida pelo receptor quando esta reflete em algum objeto próximo. O emissor deve então estar permanentemente ligado (com um resistor geralmente de 220Ω ou 330Ω para proteção de correntes altas). O receptor deve estar conectado à uma porta analógica (no nosso caso a A0) com o uso de um resistor de pullup (geralmente de $10k\Omega$). Assim, temos o seguinte circuito eletrônico:

Exemplo 3: Sensor óptico reflexivo



Exemplo 3: Sensor óptico reflexivo

Quando um objeto é aproximado do sensor óptico reflexivo, parte da luz infravermelha dela é absorvida e parte dela é refletida de volta. Assim, a distância do objeto e até mesmo o material que o compõe, podem devolver valores com pequenas variações que podem ser percebidas. Para ler esses valores, podemos utilizar o seguinte código:

Exemplo 3: Sensor óptico reflexivo

```
void setup() {  
  pinMode(A0, INPUT);  
  pinMode(3, OUTPUT);  
  digitalWrite(3, LOW);  
  Serial.begin(9600);  
}  
  
void loop() {  
  delay(100);  
  int valorLido = analogRead(A0);  
  Serial.println("Quantidade de luz IR percebida: " + (String)valorLido);  
}
```

Já para detectar um objeto e informar a proximidade acendendo o led, necessitamos definir uma faixa de leitura no sensor de acordo com a mudança de leitura provocada pelo objeto (no nosso caso de 900 à 1000) . Assim, temos:

Exemplo 3: Sensor óptico reflexivo

```
void setup(){
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  digitalWrite(3, LOW);
  Serial.begin(9600);
}

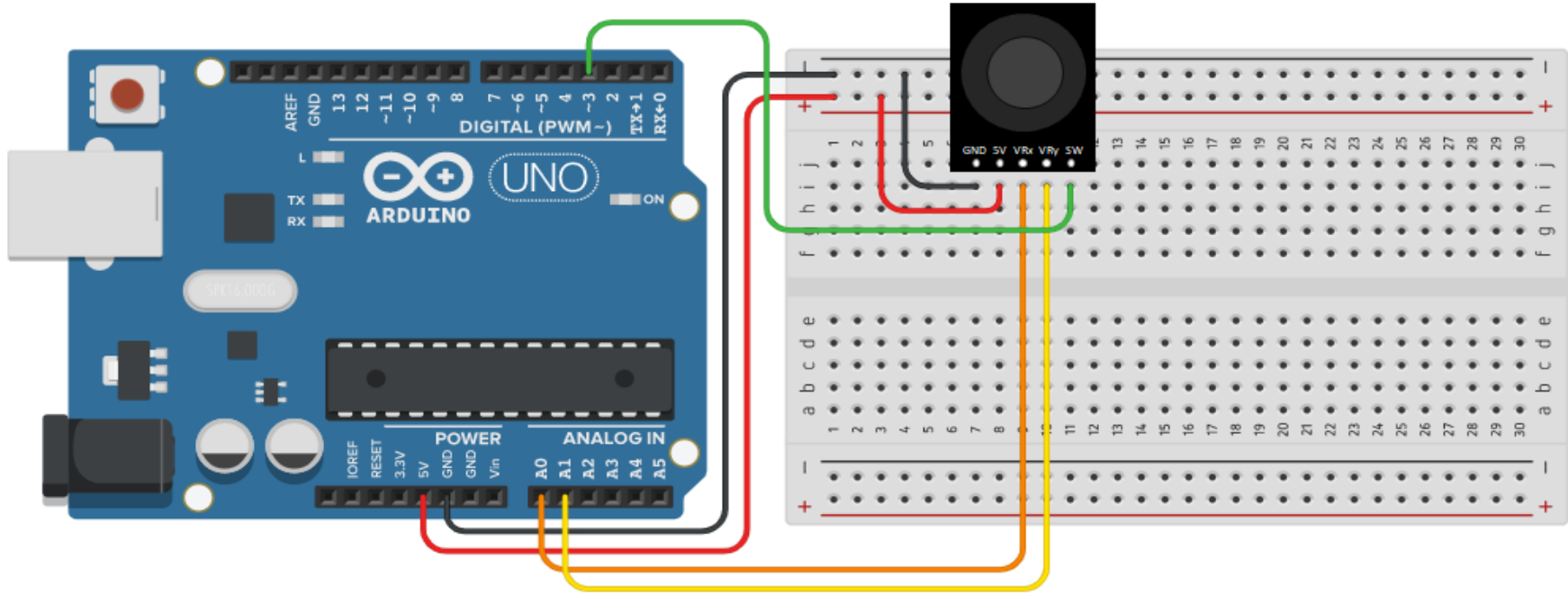
void loop(){
  delay(100);
  int valorLido = analogRead(A0);
  Serial.println("Quantidade de luz IR percebida: " + (String)valorLido);

  if(valorLido >= 900 && valorLido <= 1000) {
    digitalWrite(3, HIGH);
  } else {
    digitalWrite(3, LOW);
  }
}
```

Exemplo 4: Sensor joystick

Exemplo 3: queremos utilizar um sensor joystick para detectar cada um dos movimentos comuns que o usuário executa nele (eixo x, eixo y e eixo z) e escrever o movimento percebido. Este tipo de sensor possui 3 pinos que nos interessam: o pino analógico VRx (que controla o movimento em x, ou seja, esquerda e direita), o pino analógico VRy (que controla o movimento em y, ou seja, para cima e para baixo) e o pino digital SW (que controla o movimento em z, ou seja, botão clicado ou não clicado). Em nosso exemplo, vamos conectar o VRx na porta A0, o VRy na porta A1 e o SW na porta 3. Teremos então um circuito eletrônico como este:

Exemplo 4: Sensor joystick



Exemplo 4: Sensor joystick

Para identificarmos os movimentos, devemos capturar os valores dos 3 pinos e analisar. O VRx nos entrega valores entre 0 e 1023, sendo que quanto mais perto de 0, mais à esquerda (e consequentemente, quanto mais perto de 1023, mais à direita). O VRy também nos entrega valores entre 0 e 1023, sendo que quanto mais perto de 0, mais acima (e consequentemente, quanto mais perto de 1023, mais abaixo). O SW, sendo digital, nos entrega LOW (botão não pressionado) ou HIGH (botão pressionado). Deste modo, para detectarmos os movimentos executados pelo usuário, poderíamos ter um código como este:

Exemplo 4: Sensor joystick

```
void setup() {  
  pinMode(A0, INPUT);  
  pinMode(A1, INPUT);  
  pinMode(3, INPUT_PULLUP);  
  Serial.begin(9600);  
}  
  
void loop() {  
  if(analogRead(A0) == 0) {  
    Serial.println("Esquerda");  
  }  
  if(analogRead(A0) == 1023) {  
    Serial.println("Direita");  
  }  
  if(analogRead(A1) == 0) {  
    Serial.println("Para cima");  
  }  
  if(analogRead(A1) == 1023) {  
    Serial.println("Para baixo");  
  }  
  if(digitalRead(3) == LOW) {  
    Serial.println("Botão clicado");  
  }  
}
```



Mapping de valores

Em todos os sensores vistos até agora, fizemos uso direto dos valores entre 0 e 1023 lidos na porta analógica. Porém, muitas vezes necessitamos que este valor de referência seja convertido para uma determinada unidade que represente a grandeza medida. O grande problema é que isso as vezes é complexo. Imagine um sensor de temperatura que meça temperaturas entre 10°C e 100°C . Testando o sensor (ou olhando seu datasheet), descobrimos que quando a temperatura mínima de 10°C é medida, ele provoca um valor de entrada de 55 na porta analógica e que quando a temperatura máxima é medida, ele provoca um valor de entrada de 985 na porta analógica.



Mapping de valores

Agora, imagine a complexidade e a quantidade cálculos necessária para descobrir a quantos graus celsius equivale uma leitura de 800 na porta analógica, por exemplo? Necessitaríamos de regras de 3 combinadas, uso de ajuste por offset e até mesmos integração matemática, tornando essa conversão bem complexa.



Mapping de valores

Para resolver esse problema, o Arduino possui um método que permite realizar o chamado “mapping de valores”, que nada mais é do que um mapeamento entre os valores mínimo e máximo que o leitor pode ler (na unidade que expresse a grandeza medida) e os consequentes valores mínimo e máximo provocados na entrada analógica por eles. Esse método é o map e funciona da seguinte forma:

Mapping de valores

```
map(<valorLido>, <minPorta>, <maxPorta>, <minSensor>, <maxSensor>);
```

`<valorLido>`

valor lido na entrada analógica que queremos descobrir a quanto equivale

`<minPorta>`

valor gerado na porta digital quando o valor mínimo da grandeza desejada é lido

`<maxPorta>`

valor gerado na porta digital quando o valor máximo da grandeza desejada é lido

`<minSensor>`

valor mínimo da grandeza desejada lido pelo sensor

`<maxSensor>`

valor máximo da grandeza desejada lido pelo sensor

Mapping de valores

No nosso exemplo apresentado, para descobrir a quanto equivale em graus celsius a leitura de 800 na porta analógica provocada pelo sensor de temperatura que lê valores entre 10°C e 100°C (que provocam um valor mínimo de 55 e máximo de 985), basta fazermos:

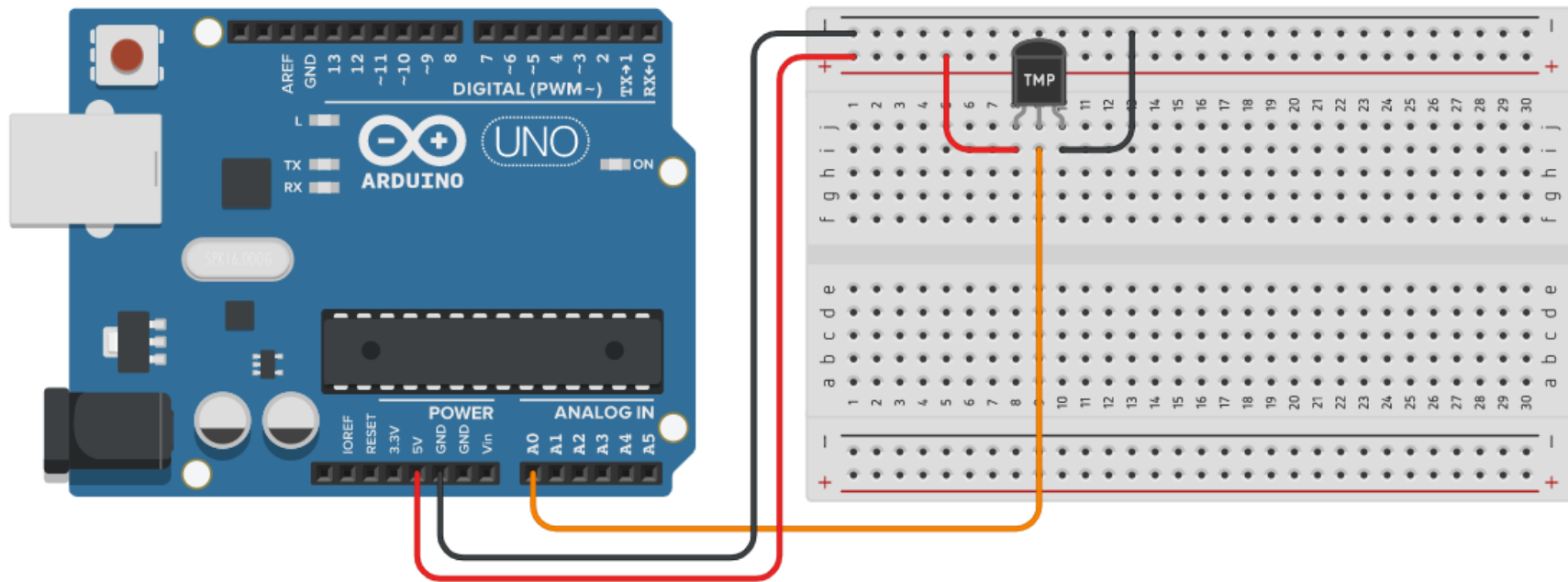
$$\text{float graus} = \text{map}(800, 55, 985, 10, 100)$$

Assim, dentro da variável grau, teríamos 82, descobrindo facilmente que o valor 800 lido na porta analógica, equivale a 82°C.

Exemplo 1: sensor de temperatura LM35

O sensor LM35 é um sensor de temperatura capaz de realizar medições entre 0°C e 150°C, com um erro de menos de meio grau (ou seja, muito preciso). Seu datasheet nos informa que o mesmo é ajustado para provocar um valor de 0 na porta analógica quando o valor mínimo de 0°C é lido e provocar um valor de 350 na porta analógica quando o valor máximo de 150°C é lido. Esse valor é capturado através do seu pino de dados. Vamos então tentar utilizar esse sensor analógico para nos devolver a temperatura ambiente percebida. Para isso, vamos ligar seu pino de dados na porta A0.

Exemplo 1: sensor de temperatura LM35



Exemplo 1: sensor de temperatura LM35

Agora, em nosso código, configuramos a porta A0 como entrada e no loop lemos o seu valor analógico (entre 0 e 1023) e logo em seguida, usamos o map fornecendo as informações de máximos e mínimos, para que o valor lido seja convertido (em °C):

```
void setup() {  
  pinMode(A0, INPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  delay(100);  
  
  int valorLido = analogRead(A0);  
  float temperatura = map(valorLido, 0, 150, 0, 350);  
  
  Serial.println("Valor lido: " + (String)valorLido);  
  Serial.println("Temperatura: " + (String)temperatura);  
}
```

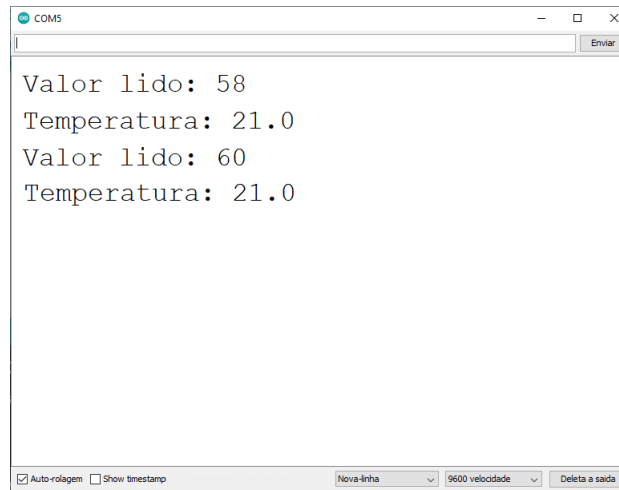
Exemplo 1: sensor de temperatura LM35

Entendendo o código: o map, torna o código muito simples e evita a realização de cálculos massivos e complexos na conversão. No setup a porta A0 está configurada como entrada pois através dela que receberemos o valor vindo de sensor LM35. Inicializamos também a Serial para poder escrever nela (e posteriormente visualizar) os resultados obtidos. Dentro do loop, a cada 100 ms, lemos a porta A0 e armazenamos o valor na variável valorLido.

Exemplo 1: sensor de temperatura LM35

Entendendo o código: Agora, necessitamos converter esse valor lido (qualquer coisa entre 0 e 1023) em graus celsius. Como conhecemos as temperaturas mínima e máxima passíveis de leitura pelo LM35 (0°C à 150°C na faixa de valor que queremos utilizar) e qual o valor máximo e mínimo que as leituras provocam na entrada da porta analógica (0 a 350 conforme datasheet do componente), utilizamos o método map para que o valor lido seja convertido e armazenado na variável temperatura. Após, escrevemos na Serial qual o valor lido através do sensor e qual a temperatura respectiva em graus celsius. Se tudo ocorreu bem, teremos no nosso Monitor Serial uma saída como esta:

Exemplo 1: sensor de temperatura LM35



COM5

Valor lido: 58
Temperatura: 21.0

Enviar

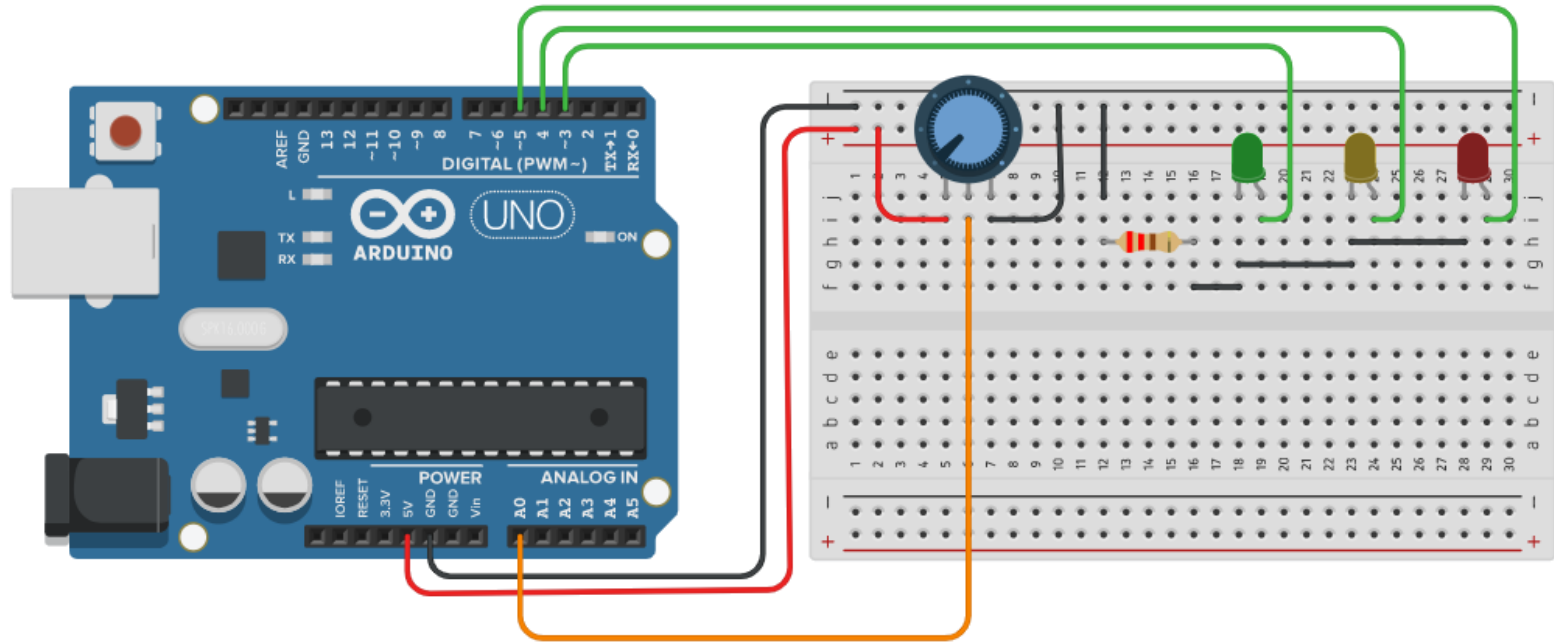
☒ Auto-rolagem ☐ Show timestamp

Nova-linha 9600 velocidade Deleta a saída

Exemplo 2: potenciômetro

Apesar dos potenciômetros serem componentes eletrônicos do tipo resistor variável, podemos utilizá-los em nossos projetos como se fossem sensores dos quais conseguimos detectar o giro. Para isso, basta conectar as suas extremidades ao 5V e o GND e o seu pino central à uma porta analógica. Isso porque, ao girar o pino, a resistência gerada cria um divisor de tensão entre o pino 5V e central e entre o central o GND. Deste modo, conseguimos modificar a tensão que chega à porta digital, conforme o nosso giro. Para testar, podemos criar um circuito eletrônico como o mostrado abaixo, com o pino central do potenciômetro ligado à porta analógica A0 do Arduino e 3 leds ligados às portas digitais 3, 4 e 5.

Exemplo 2: potenciômetro



Exemplo 2: potenciômetro

Nessa configuração, independente do valor do nosso potenciômetro (que no exemplo será de $5K\Omega$), quando seu pino estiver totalmente para a esquerda, o valor lido pela porta analógica será 1000 (mínima resistência implica em máxima tensão) e quando seu pino estiver totalmente para a direita o valor lido pela porta analógica será 0 (máxima resistência implica em mínima tensão). Para inverter, bastaria subtrair o valor lido de 1000 que, com isso, teríamos um valor que aumenta conforme o pino é girado para a direita.

Exemplo 2: potenciômetro

Agora, para converter esse valor entre 0 e 1000 em uma resistência entre 0Ω e $5k\Omega$, podemos utilizar um mapping, mapeando o valor lido com o método map com base na escala de valores possíveis na leitura analógica em comparação com a escala de valores possíveis da resistência do potenciômetro. Colocando isso em código, teríamos:

```
void setup(){
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  delay(100);
  int valorLido = 1000 - analogRead(A0);
  float resistencia = map(valorLido, 0, 1000, 0, 5000);

  Serial.println("Valor lido: " + (String)valorLido);
  Serial.println("Resistência: " + (String)resistencia);
}
```



Exemplo 2: potenciômetro

Com o método `map`, o mapping do valor lido (algo entre 0 e 100) para transformação em ohms (entre 0 e 5000) tornou-se simples. Agora, imagine que dentro do loop, logo após as mensagens enviadas para a Serial com os valores, desejássemos que os leds fossem ligados conforme o pino do potenciômetro fosse girado em sentido horário. Aqui, poderíamos utilizar duas lógicas:

Exemplo 2: potenciômetro

Lógica 1: a primeira possibilidade (e mais convencional), é dividir os possíveis valores lidos (de 0 a 1000) em três grupos de valores, sendo o primeiro de 0 a 333 (ligando o led 1), o segundo de 334 à 666 (ligando o led 2) e o terceiro de 667 à 1000 (ligando o led 3). Assim, basta testaria testar o valor lido com três condicionais para identificar em qual grupo (faixa de valor) está e qual o respectivo led que deve ser ligado. Assim, teríamos no loop:

Exemplo 2: potenciômetro

```
if(valorLido >= 0 && valorLido <= 333) {  
    digitalWrite(3, HIGH);  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
}  
  
if(valorLido >= 334 && valorLido <= 666) {  
    digitalWrite(3, LOW);  
    digitalWrite(4, HIGH);  
    digitalWrite(5, LOW);  
}  
  
if(valorLido >= 667 && valorLido <= 1000) {  
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(5, HIGH);  
}
```

Exemplo 2: potenciômetro

Lógica 2: deixar o código mais elegante, poderíamos utilizar um mapping de valores para converter o valor lido (algo entre 0 e 1000), em um valor entre 1 e 3, que vai nos indicar exatamente qual led ligar. Apesar de necessitarmos de uma linha a mais de código do que na solução anterior, o condicional fica mais simples e não necessita de condição encadeada para testar faixas de valores. Assim, temos:

Exemplo 2: potenciômetro

```
int faixaLed = map(valorLido, 0, 1000, 1, 3);

if(faixaLed == 1) {
    digitalWrite(3, HIGH);
    digitalWrite(5, LOW);
    digitalWrite(3, LOW);
}

if(faixaLed == 2) {
    digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
}

if(faixaLed == 3) {
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
}
```



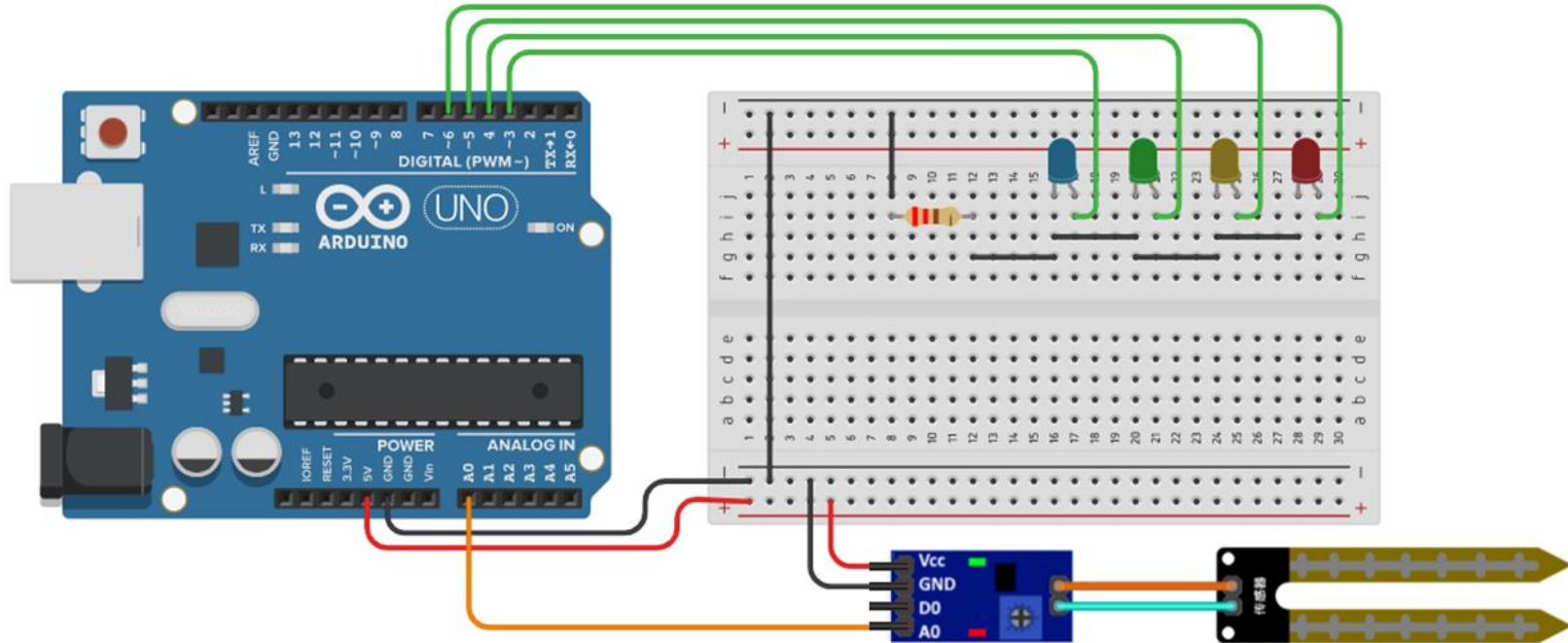
Exemplo 3: sensor de umidade do solo

O sensor de umidade do solo é especialmente desenvolvido para ser introduzido no solo a ser monitorado e, de acordo com a umidade percebida, nos devolver um valor analógico que varia de 0 (totalmente seco) até 1023 (totalmente encharcado). Este sensor é composto de duas partes distintas: uma haste, responsável por ficar em contato com o solo e uma placa lógica que captura as micro correntes geradas pela haste, convertendo-as em informação sobre a umidade percebida.

Exemplo 3: sensor de umidade do solo

A placa, geralmente possui 4 pinos, sendo eles o VCC (5V), o GND, o analógico e um pino digital que pode ser utilizado caso não se deseje saber o valor da umidade, mas apenas mudar o estado do sensor quando uma umidade máxima (ajustada através do trimpot presente em seu corpo) for atingida. No nosso exemplo, não iremos utilizar esse pino digital, sendo o pino analógico ligado à nossa porta A0. Para chamar a atenção e ilustrar melhor o nível de umidade do solo, iremos fazer algo similar ao exemplo do potenciômetro, colocando 4 leds que são acionados de acordo com 4 faixas de umidade que vamos estabelecer. Estes leds, serão então ligados nas portas digitais 3, 4, 5 e 6.

Exemplo 3: sensor de umidade do solo



Exemplo 3: sensor de umidade do solo

Para o nosso mapeamento, vamos definir 4 faixas que representem a umidade do solo. Se dividirmos os 1024 valores possíveis por 4, teremos que cada faixa deve contemplar 256 unidades de valor. Assim, poderíamos considerar que valores lidos entre 0 e 255 representam o solo muito seco, valores lidos entre 256 e 511 representam o solo ressecado, valores lidos entre 512 e 765 representam o solo úmido e valores lidos entre 766 e 1023 representam o solo encharcado. Porém esse trabalho todo não é necessário, pois podemos simplificar nossa lógica com o mapping dos valores, transformando entradas entre 0 e 1023 em faixas entre 1 e 4:

Exemplo 3: sensor de umidade do solo

```
void setup(){
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  delay(100);
  int valorLido = analogRead(A0);
  int faixa = map(valorLido, 0, 1023, 1, 4);

  Serial.println("Valor lido: " + (String)valorLido);
  Serial.println("Faixa de umidade: " + (String)faixa);

  if(faixa == 1) {
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, HIGH);
  }
}
```


Exemplo 3: sensor de umidade do solo

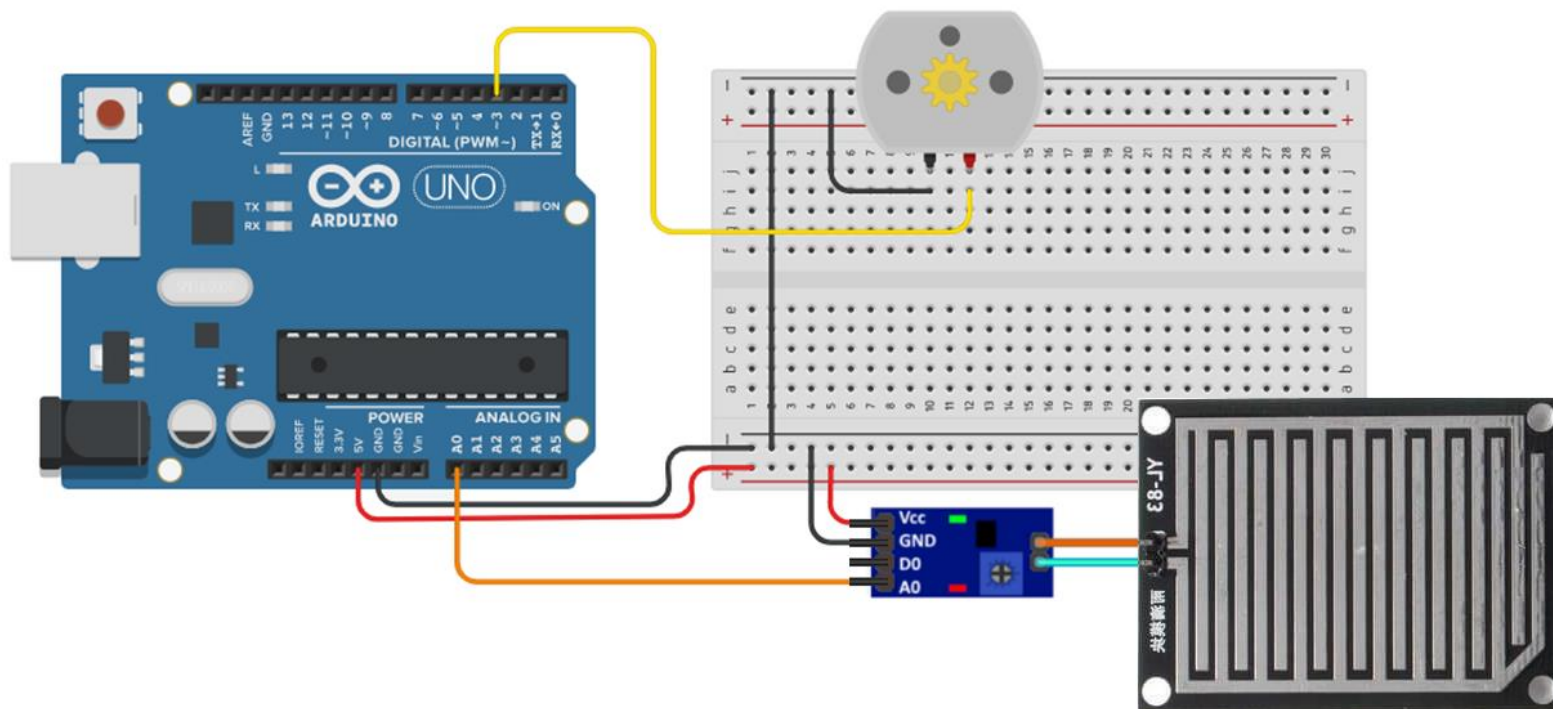
```
if(faixa == 2) {  
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(5, HIGH);  
    digitalWrite(6, LOW);  
}  
  
if(faixa == 3) {  
    digitalWrite(3, LOW);  
    digitalWrite(4, HIGH);  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
}  
  
if(faixa == 4) {  
    digitalWrite(3, HIGH);  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
}  
}
```

Aqui, estamos apenas ligando e desligando leds como alerta para a umidade percebida, lembrando que quando mais alta a faixa, mais úmido o solo (por causa do modo como os valores analógicos são lidos, com o 0 sendo a menor umidade e o 1023 sendo a maior umidade). Poderíamos modificar o exemplo para, através de um relé ligado por uma das portas digitais, ativar uma válvula solenoide ou bomba de água para que o solo seja regado.

Exemplo 4: sensor de respingos

O sensor de respingos é muito similar ao sensor de umidade do solo e conta com uma placa de detecção de líquidos (cujas gotas permitem a passagem de micro correntes) e uma placa que transforma essas micro correntes em informação, devolvendo um valor entre 0 (totalmente seca) e 1023 (totalmente molhada). Os pinos deste sensor são análogos ao do sensor anterior e a nossa única mudança neste exemplo é que em lugar dos leds indicativos, queremos um motor DC alimentado pela porta 3 e que tenha força de, ao perceber um início de chuva, mover uma polia que puxe a corda de um varal para proteger as roupas.

Exemplo 4: sensor de respingos



Exemplo 4: sensor de respingos

Vamos realizar o map com uma lógica um pouco diferente, transformando o valor lido em um percentual de chuva que vai de 0 a 100. O problema é que não é o 0 que representa a placa totalmente seca e o 1023 a placa totalmente molhada e sim o inverso. Como fazemos o mapeamento das duas escalas, se uma está invertida em relação à outra? Simples! Invertendo os valores de uma das escalas.

Exemplo 4: sensor de respingos

Então, em nosso código, após ler o valor da porta analógica, vamos transformá-lo em um percentual de 0 a 100 mas usando a escala de valor lido na porta invertida (1023 e 0 e não 0 e 1023). Após, caso o percentual de respingos na placa (chuva no nosso caso) for maior do que 20 (que representa 20%) acionamos o motor por 10 segundos para puxar a corda e proteger as roupas estendidas.

Exemplo 4: sensor de respingos

```
void setup(){
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  digitalWrite(3, HIGH);

  Serial.begin(9600);
}

void loop(){
  delay(100);
  int valorLido = analogRead(A0);
  float percentual = map(valorLido, 1023, 0, 1, 100);

  Serial.println("Valor lido: " + (String)valorLido);
  Serial.println("Percentual de chuva: " + (String)percentual);

  if(percentual > 20) {
    digitalWrite(3, HIGH);
    delay(10000);
    digitalWrite(3, LOW);
  }
}
```