

Resumo Arduino Challenge

HARDWARE:

- *Dispositivo de entrada: Le informações, envia informações pro Arduino.
 - *Dispositivo de saída: Executa ações(led, motores).
 - *Alimentação: Dar energia a algo para que este possa executar alguma função.
 - VCC: Positivo(+).
 - GND: Negativo(-).
- Protoboard: placa de montagem.

SOFTWARE:

- * #define led 12 --> define o led no pino 12 do Arduino
- * #define sensor 0 --> define o sensor no pino A0 do Arduino
- * pinMode(pino, CONSTANTE);
 - pino: onde liga o componente no Arduino.
 - CONSTANTE: -INPUT: Entrada, leitura de sensor.
 - OUTPUT: Saida, acionar algum dispositivo.
 - Ex.: pinMode(12, OUTPUT);
pinMode(led, OUTPUT);
- * digitalWrite(pino, CONSTANTE); --> Ligado ou Desligado.
 - pino: onde liga o componente no Arduino ou nome do pino.
 - CONSTANTE: -HIGH: Ligado.
 - LOW: Desligado.
 - Ex.: digitalWrite(12, HIGH);
digitalWrite(led, HIGH);
- * analogWrite(pino, variável); --> Controla a intensidade
 - pino: onde liga o componente no Arduino ou nome do pino.
 - variável: valores de 0 a 255.
 - Ex.: analogWrite(12, 0); --> led apagado
analogWrite(led, 150); --> intensidade media
analogWrite(led, 255); --> intensidade máxima

FIQUE ATENTO!!!!

O analogWrite(); não funciona em todos os pinos do Arduino. Isso acontece, porque para o usarmos alternando a intensidade de seus valores, usamos o conceito de PWM (Modulação por largura de Pulso é o que permite a essa característica). Os pinos que podem ser usados são: 3, 5, 6, 9, 10 e 11.

- * delay(1000); --> Para o programa por um tempo que é definido em milissegundos(1000 milissegundos = 1 segundo).

* Sensor de refletancia: Percebe se a superfície se encontra clara ou escura. Toda vez que ele ler um valor acima de 300 é uma superfície escura e menor que 300 é uma superfície clara. Os valores lidos pode ser de 0 a 1023.

- * analogRead(pin); --> Leitura de um pino analógico.
 - pin: pino em que o sensor esta ligado no Arduino.
 - Ex.: analogRead(3);
analogRead(sensor);

- * Estrutura if(): Tomar decisões, faz comparações entre informações
- ```
Se(Minhanota > NotadoMeuAmigo)
{
 OferecerAjuda;
}
```

| Operador de Comparação | Simbologia                          |
|------------------------|-------------------------------------|
| Maior que              | >                                   |
| Menor que              | <                                   |
| Maior ou igual a       | >=                                  |
| Menor ou igual a       | <=                                  |
| Igual a                | == (dois sinais de iguais seguidos) |
| Diferente de           | !=                                  |

### -A ação OfercerAjuda só acontecera de for verdadeira.

Exemplo:

```
#define led 12 // nomeação dos pinos
#define sensor 3
void setup()
{
 pinMode(led, OUTPUT); // configuração dos pinos
 pinMode(sensor, INPUT);
}

void loop()
{
 if(analogRead(sensor) > 300) // condição a ser atendida
 {
 digitalWrite(led, HIGH); // Ação implementada
 }

 if(analogRead(sensor) < 300)
 {
 digitalWrite(led, LOW);
 }
}
```

### \*Operadores

AND: Relação de inclusão. "Estou participando do torneio e estou gostando."

Ambas as coisas estão acontecendo.

Simbologia: &&

```
if(analogRead(sensor1) > 300 && analogRead(sensor2) > 300)
{
 digitalWrite(led, HIGH);
}
```

Neste caso, o led só será aceso se o sensor1 e o sensor2 lerem uma "faixa escura". Se por acaso, o sensor1 ler uma "faixa clara" e o sensor2 ler uma "faixa escura", ou se o sensor 1 ler uma "faixa escura" e o sensor2 ler uma "faixa clara", o led não se acenderá. Para que a ação aconteça, é necessário que ambas as condições sejam atendidas.

Com isso percebemos o seguinte:

| Sensor 1     | Sensor 2     | Led        |
|--------------|--------------|------------|
| Faixa clara  | Faixa clara  | Não acende |
| Faixa clara  | Faixa escura | Não acende |
| Faixa escura | Faixa clara  | Não acende |
| Faixa escura | Faixa escura | Acende     |

OR: Relação de independência. Se apenas um estiver verdadeiro a ação é realizada.

Simbologia: ||

```
if(analogRead(sensor1) > 300 || analogRead(sensor2) > 300)
{
 digitalWrite(led, HIGH);
}
```

Neste caso, o led só não será acesso se o sensor1 e o sensor2 lerem "faixa clara". Se por ventura, qualquer um dos dois lerem uma "faixa escura", então o led deverá ser aceso.

Com isso percebemos o seguinte:

| Sensor 1     | Sensor 2     | Led        |
|--------------|--------------|------------|
| Faixa clara  | Faixa clara  | Não acende |
| Faixa clara  | Faixa escura | Acende     |
| Faixa escura | Faixa clara  | Acende     |
| Faixa escura | Faixa escura | Acende     |

\*`millis()` --> Retorna o tempo em que o Arduino esta ligado. Retorna em milissegundos

\*Estrutura `else()`

```
 se(estiver chovendo)
 {
 Abre a sombrinha;
 }senão{
 Deixa fechada;
 }
```

\*Variáveis: locais que armazenam valores que serão usados posteriormente no programa.

-Para a declararmos, usamos a seguinte estrutura:

<tipo\_de\_dado> <nome\_variável>;

-Onde:

<tipo\_de\_dado>: equivale se ela será do tipo `int` ou do tipo `float`;

<nome\_variável>: como queremos nomeá-la.

O nome de uma variável, por padrão:

-Deve ser composto apenas por letras, números e sublinhado ('\_').

-Deve sempre começar por uma letra;

-Não pode ser igual a nenhuma palavra reservada, ou seja, aquelas palavras que já existem dentro da linguagem de programação, como por exemplo, `if`, `analogRead`, ou algo assim.

\*\*Lembre-se sempre: toda declaração de variável deve ser finalizada com ponto e vírgula.

Exemplo: `float leitura_sensor1;`

`float leitura_sensor2;`

\*Tipos de variáveis:

-Tipo inteiro(`int`): Conjunto dos números inteiros.

-Ex.: `int x = 5;`

-Tipo ponto flutuante(`float`): Números reais que contem uma vírgula, números fracionados.

-Ex.: `float y = 10,2;`

-Até o momento, aprendermos o que é uma variável e como fazemos a sua declaração. Mas no caso do nosso exemplo, para usarmos a variável, é interessante colocarmos o valor da leitura dos sensores dentro das mesmas, para que assim ela possa armazená-lo.

-Para fazermos isso, seguimos a estrutura:<nome\_var> = <valor>;

-Onde:<nome\_var>: equivale a variável que armazenará o valor;

= : corresponde a atribuição do valor à variável;

<valor>: o valor que desejamos armazenar.

-No nosso exemplo ficaria:`leitura_sensor1 = analogRead(sensor1);`

`leitura_sensor2 = analogRead(sensor2);`

\*Comunicação Serial: Transmite dados do computador para o Arduino e vice versa.

-`Serial.begin()`; --> Resolução de taxa de transmissão. Sera usado a taxa 9600. Sera colocada dentro do void `setup()`.

-`Serial.println()`; --> Escrita na tela, mostra os dados com quebra de linha(pula uma linha).