



BOSCH
Invented for life

Aula 3.3 - Entradas digitais

inside.Docupedia Export

Author: Lundgren Daniel (CtP/ETS)
Date: 15-Dec-2021 13:01

Table of Contents

1 Código	4
2 Sensores	5
3 Sensor de movimento de presença PIR	9
4 Circuito	10
5 Código	11

Como visto na aula anterior, a ESP32 possui diversas GPIOs (General Purpose Input/Output), que são pinos que podem ser configurados ou como entrada, ou como saída.

Quando configurado como saída, o pino regula a tensão de saída do pino de acordo com o valor recebido. Esse valor é definido no comando **write**, podendo ter o valor **HIGH** (3,3V), ou **LOW** (0V).

Já quando configurado como entrada, o pino espera ler um valor de tensão, que significa se está em HIGH ou LOW. O pino configurado como INPUT (entrada) estará sempre recebendo dados, geralmente de um sensor.

1 Código

Primeiro temos que configurar o pino como entrada. Dentro da função **setup** informamos o número correspondente do pino, e damos o valor de **INPUT** :

```
1 pinMode(4, INPUT);
```

Agora com nosso pino configurado, podemos fazer a leitura a qualquer momento, e atribuir esse valor a uma variável:

```
1 valorEntrada = digitalRead(4);
```

2 Sensores

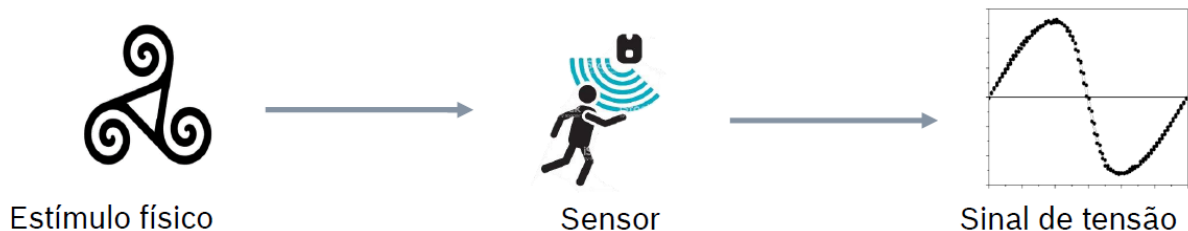
Um sensor é um dispositivo que transforma uma grandeza física (temperatura, pressão, movimento, luz, etc.) em uma grandeza física mensurável (sinal elétrico, mecânico, ótico, etc.).

Exemplo:

Um termômetro é um exemplo de sensor, um dispositivo que transforma **temperatura** em uma grandeza mensurável: altura do mercúrio líquido, a partir da sua graduação é possível ler o seu valor.



Em sistemas embarcados estamos sempre interessados em um valor de tensão, os sensores serão responsáveis por converter um estímulo físico em um sinal elétrico.



Os sensores podem se dividir em dois tipos:

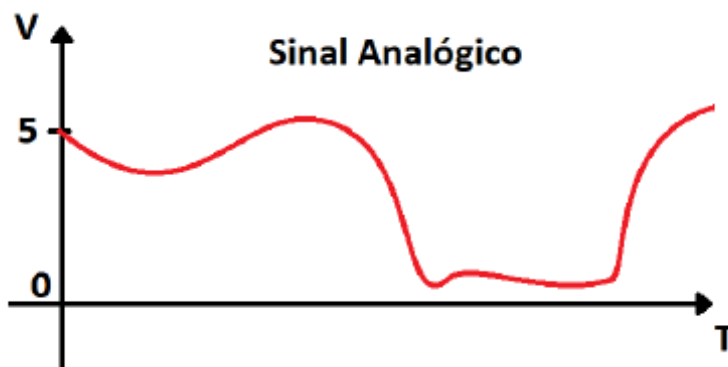
- **Ativos:** transformam diretamente o estímulo em energia elétrica. Exemplos: sensores piezoelétricos, termopar.
- **Passivos:** são resistências que variam seu valor de acordo com o estímulo que recebem, para medir o seu valor é necessário um circuito de alimentação. Exemplos: LDR, sensor infravermelho.

Também podemos dividir em sensores **digitais** e **analógicos**:

- **Digitais:** são sensores cuja saída pode assumir apenas dois valores: 0 ou 1 (0V ou 3,3V no caso da ESP32). Exemplos: chaves fim de curso, sensor de presença.

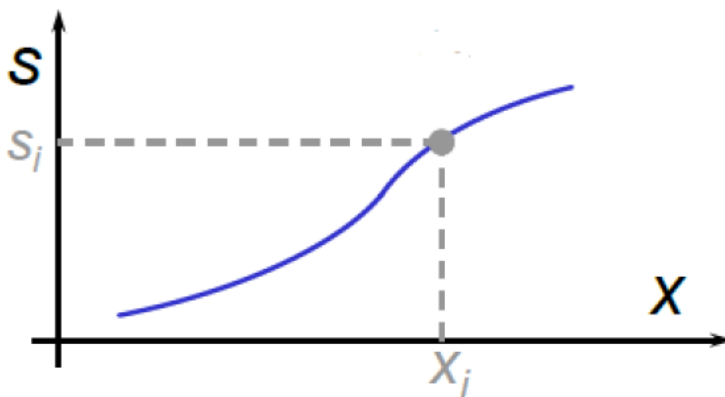


- **Analógicos:** são sensores cuja saída pode assumir qualquer valor dentro da escala entre o valor mínimo e máximo de tensão. Exemplos: potenciômetros, sensores de temperatura, etc.

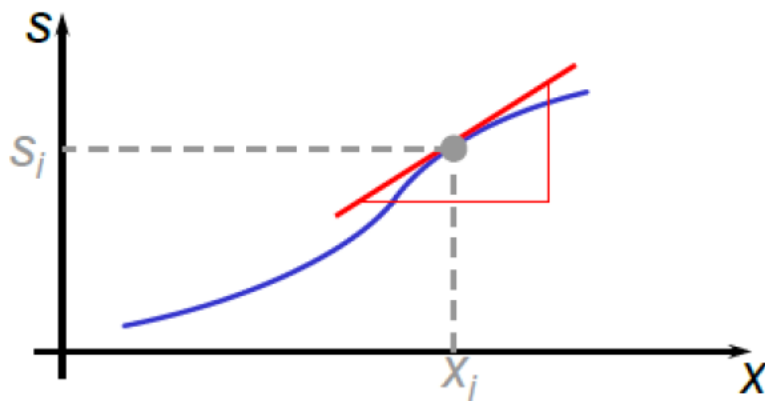


Os sensores possuem algumas características importantes, que devem ser conhecidas e diferenciadas umas das outras para não haver confusão quando um sensor for escolhido.

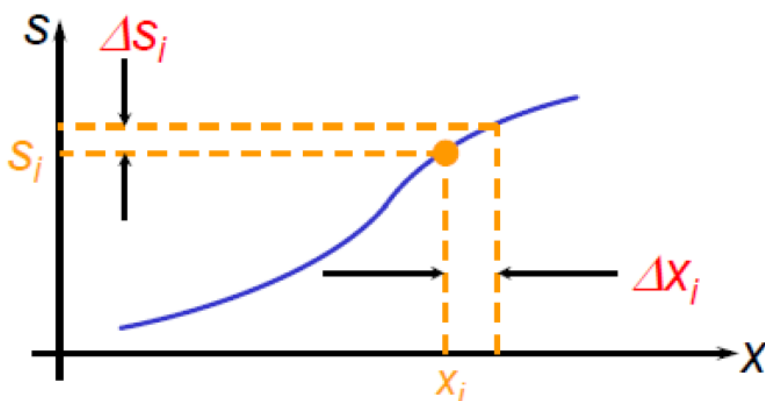
Vamos analisar uma função entre o estímulo físico medido (X) e o valor de tensão medido (S):



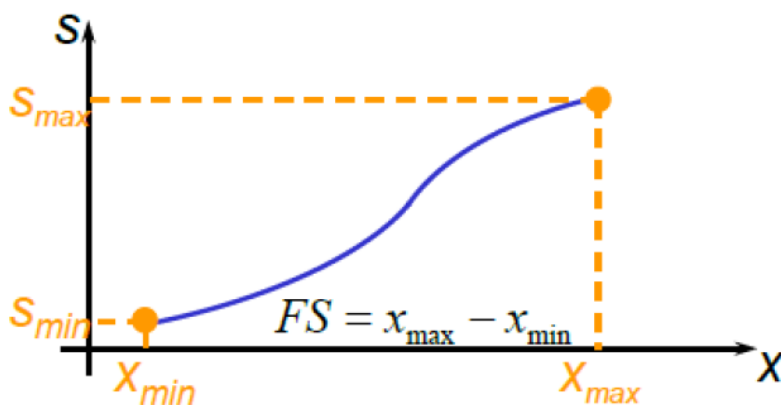
Sensibilidade se trata do quanto o sensor faz variar o valor de tensão criado a partir da variação do estímulo físico. Se uma pequena mudança do estímulo (da temperatura, por exemplo) fizer o sinal variar bastante, dizemos que o sensor tem uma alta sensibilidade. Ele é indicado na curva pela reta tangente em vermelho (quanto mais inclinada, maior a sensibilidade).



Resolução é o menor incremento de estímulo (X) que provoca alguma variação detectável do sinal (S).
Exemplo: para um sensor de distância, numa aplicação em que o objeto medido se move alguns centímetros, é interessante que a resolução seja em milímetros. Se o objeto medido precisa se mover por metros até que se note alguma variação no sinal, não é um sensor recomendável para este caso.

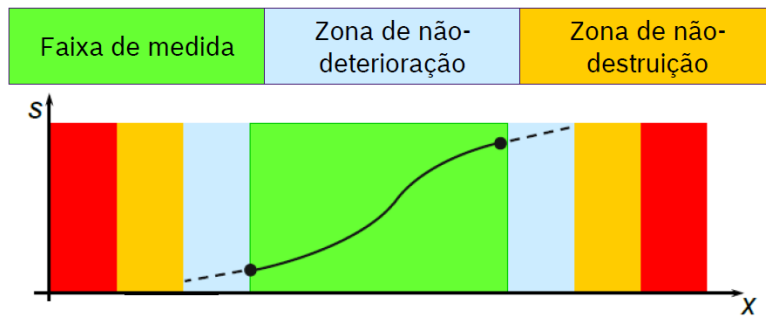


Faixa de medida é a diferença entre a mínima e a máxima medida que um sensor é capaz de detectar.
Exemplo: se queremos medir a temperatura de um forno que pode aquecer até 200°C, e de um freezer que chega a -20°C, um sensor com faixa de medida de 0°C a 50°C não é recomendável. Porém, para medir a temperatura ambiente em uma sala de reunião, esse sensor é uma boa escolha.



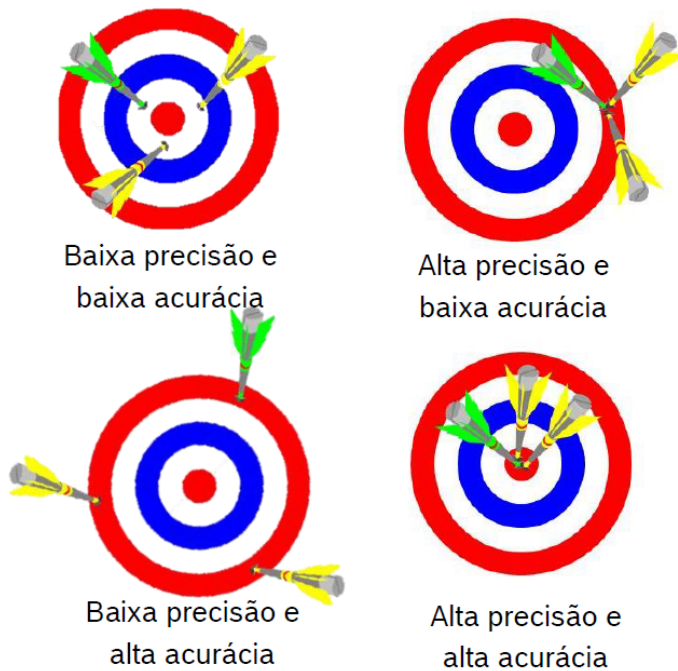
A **zona de utilização** possui 3 faixas: a faixa de medida, a zona de não-deterioração e a zona de não-destruição:

- Faixa de medida é a faixa onde o sensor deve trabalhar.
- Zona de não-deterioração é a faixa onde as medidas não são confiáveis, porém o sensor continua totalmente funcional.
- Zona de não-destruição é a faixa onde o sensor é deteriorado, e pode fornecer medidas errôneas mesmo quando volta a operar na faixa de medida. Se passar por essa zona, necessita de calibração.



Precisão é a capacidade do sensor de dar o mesmo resultado para medições que foram feitas em momentos diferentes, porém sob as mesmas condições.

Acurácia é a capacidade do sensor de obter medidas próximas ao valor real do estímulo.

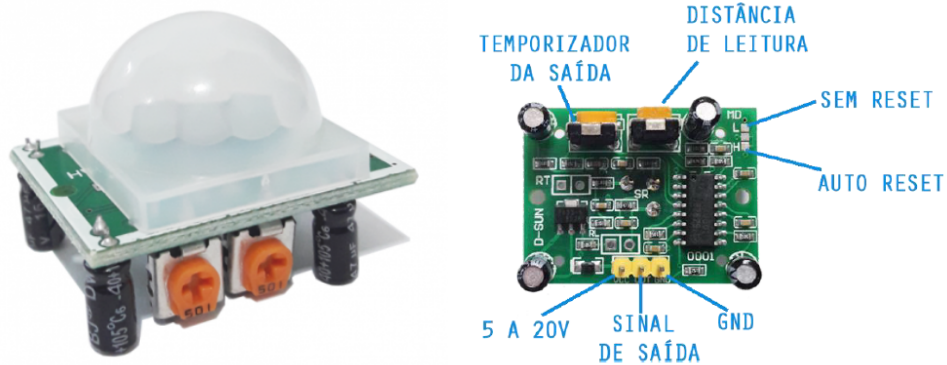


As características de cada sensor podem ser consultadas em seu **datasheet**, que são facilmente encontrados na internet.

3 Sensor de movimento de presença PIR

O Sensor de Movimento PIR DYP-ME003 consegue detectar o movimento de objetos que estejam em uma área de até 7 metros! Caso algo ou alguém se movimentar nesta área o pino de alarme é ativado.

[Datasheet](#)

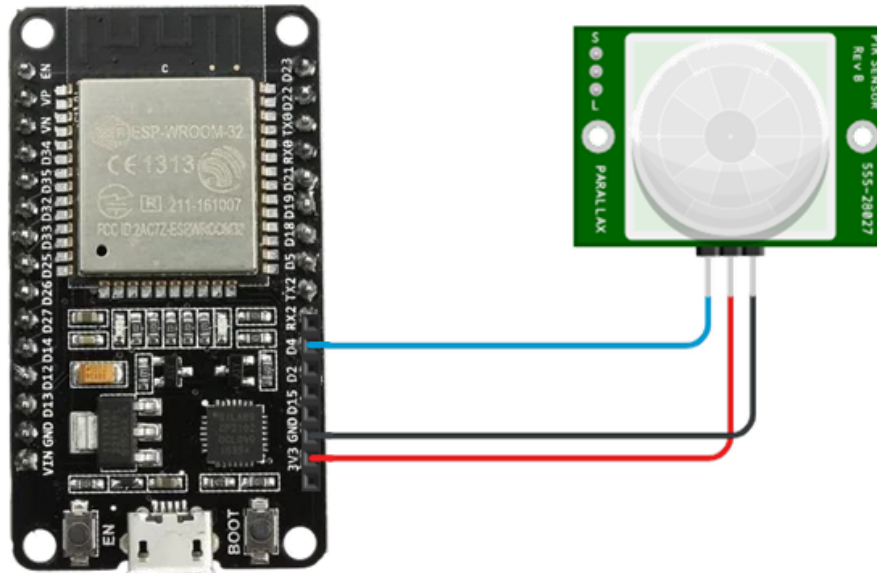


É possível ajustar a duração do tempo de espera para estabilização do PIR através do potenciômetro laranja em baixo do sensor bem como sua sensibilidade. A estabilização pode variar entre 5-200 seg e entre 3 e 7 metros.

Quando a saída é acionada pelo movimento detectado esta ficará em alto por um curto período de tempo, mesmo se não haja mais movimento. O tempo em alto pode ser setado variando o potenciômetro 'Time', sendo que o outro altera a sensibilidade.

4 Circuito

A montagem do circuito deve ser feita como na imagem abaixo. Vamos utilizar a porta digital D4 para ler os dados do sensor. Além disso devemos conectar o sensor nos pinos 3V3 e GND para fazer sua alimentação. Como vamos utilizar o LED built-in da ESP não vamos precisar montar um LED externamente, mas sinta-se a vontade se quiser utilizar.



5 Código

Como conectamos o pino de dados do PIR na porta 4 da ESP, precisamos declarar isso no código:

```
1  bool in;  
2  
3  void setup(){  
4      pinMode(4, INPUT);  
5      pinMode(2, OUTPUT);  
6  }
```

Criamos uma variável booleana na qual será armazenado o valor de leitura do sensor. No setup declaramos o pino 4 como entrada (conectado no PIR) e o pino 2 como saída (LED built-in da ESP).

```
1  void loop(){  
2      in = digitalRead(4);  
3      digitalWrite(2, in);  
4  }
```

No loop estaremos sempre lendo o valor da porta 4, e mandando esse valor para a porta 2. Ou seja, quando o sensor enviar HIGH, a porta 2 receberá HIGH, acendendo o LED. Quando o sensor enviar LOW, a porta 2 receberá LOW, apagando o LED.

Assim temos um sistema muito parecido com o sistema de alarme de uma casa, a diferença é que nesse caso a saída é um alto falante. Você pode tentar reproduzir algo parecido utilizando o buzzer.