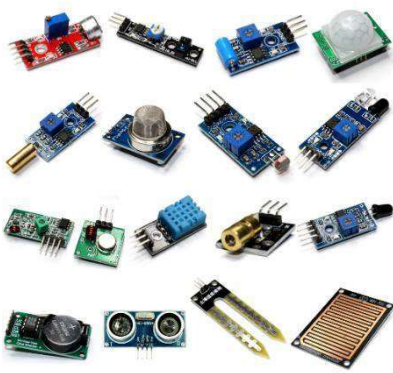


SENSORES

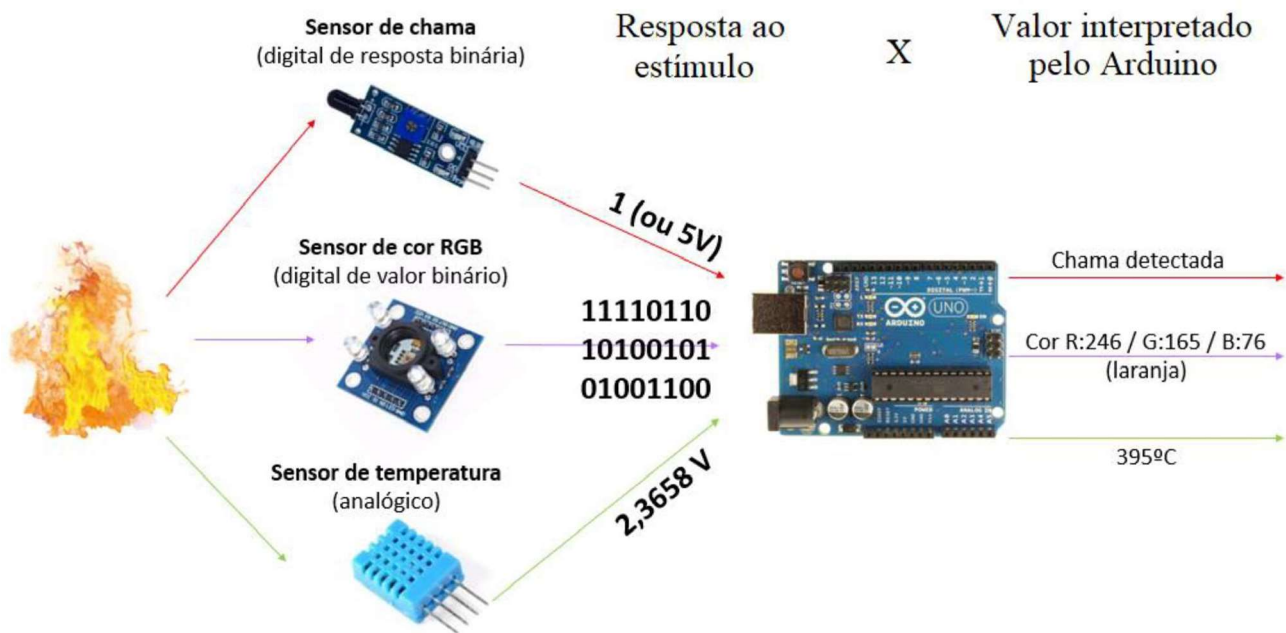
TRABALHANDO COM SENSORES



Os sensores são dispositivos que tem como função detectar um determinado fenômeno e responder ao estímulo recebido. Cada sensor é construído, então, com um propósito específico e bem definido. Deste modo, para cada situação que desejamos monitorar, devemos agregar um ou mais sensores, conseguindo assim, obter as informações desejadas do mundo externo. Nos últimos anos, principalmente por conta das placas de prototipagem, mas

também pela profusão de dispositivos eletroeletrônicos cada vez mais “inteligentes”, surgiram diversos sensores de baixo custo e fácil uso. Queremos neste capítulo, trazer uma coletânea dos principais sensores utilizados em conjunto com o Arduino (e em outras placas similares em projetos Maker) e os códigos e técnicas utilizados para acessar suas informações. Mas para isso, precisamos primeiramente entender como o Arduino lê as informações dos mesmos.

Os sensores do Arduino podem ser de 2 tipos distintos: digitais e analógicos. A diferença básica é que os sensores analógicos conseguem devolver ao Arduino qualquer tensão entre 0V e 5V como resposta ao estímulo do fenômeno medido (e esse valor, como veremos adiante, é mapeado para um número entre 0 e 1023) e os sensores digitais apenas conseguem devolver ao Arduino ou 0V ou 5V (chamados de estado baixo e estado alto, respectivamente). Assim, estes sensores digitais podem ser sensores de resposta binária (aqueles que devolvem apenas duas respostas como sim ou não, detectado ou não detectado, etc) ou sensores que utilizam combinações dos estados baixo e alto para o envio de bits 0 ou 1 em uma sequência binária que após é mapeada em um valor. Abaixo, temos um esquema que sintetiza então os modos de resposta destes sensores:



Os três sensores são de propósitos diferentes: o de cima é um sensor que informa a detecção de chama, o do meio é um sensor que informa o valor RGB da cor detectada e o de baixo é um sensor que mede a temperatura. Apesar de sujeitos ao mesmo estímulo (uma mesma chama), cada um enviará ao Arduino um valor diferente, de uma forma diferente e que representa uma informação diferente. O sensor de chama é um sensor de resposta binária e ao detectar fogo envia ao Arduino 5V (equivalente ao estado alto, ou 1); já o sensor de cor RGB é um sensor de valor binário e alterna seus estados altos e baixos para enviar ao Arduino, de modo binário, os valores de composição das 3 matizes que compõe a cor (no caso 246 de vermelho, 165 de verde e 76 de azul, resultando em um tom de laranja); finalmente o sensor de temperatura varia a sua tensão de acordo com a temperatura medida e envia ao Arduino 2,3658 volts, que serão convertidos em um valor entre 0 e 1023 (adiante entenderemos isso) que finalmente é mapeado para a temperatura medida (no caso, 395°C). Isso é importante para ilustrar que para podermos trabalhar facilmente com sensores no Arduino, precisamos entender não só como os diferentes tipos de sensores enviam os dados gerados em resposta aos estímulos mas, principalmente, como o Arduino deve lê-los e interpretá-los.

Leitura de sensores digitais de resposta binária

Para a leitura de sensores digitais de resposta binária, utilizamos o método `digitalRead`, informando sua porta e tendo como resposta um estado baixo (0 / LOW) ou um estado alto (1 / HIGH). Alguns exemplos deste tipo são o sensor PIR (objeto próximo detectado ou não detectado), o sensor de vibração (vibração detectada ou não detectada) e

o push button (botão pressionado ou não pressionado).

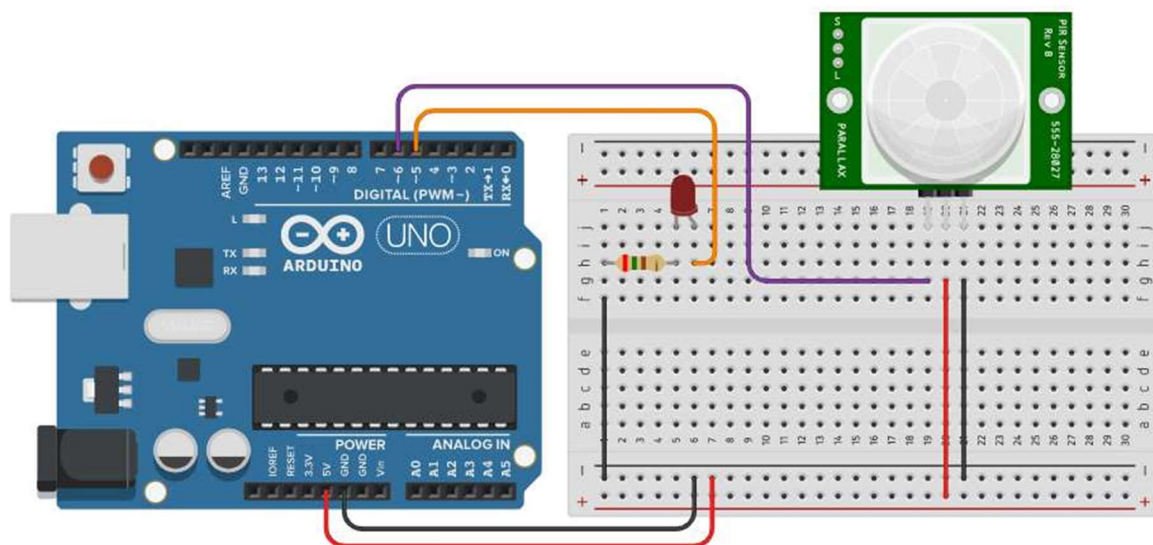
```
digitalRead( <porta> );

<porta>
    porta à qual o sensor de resposta binário está ligado

resposta:
    a resposta do método será o estado LOW (0) ou HIGH (1)
```

Sensor de proximidade PIR: Exemplo detalhado

O sensor PIR é um sensor que altera o seu estado para HIGH quando detecta algum movimento em seu raio de ação. Imagine então que queremos que um sensor de presença do tipo PIR, ao detectar um objeto em movimento próximo a si, ligue um led como indicativo. Os sensores digitais de resposta binária geralmente possuem 3 pinos: GND, VCC (ou ++ ou 5V) e um pino de envio de sinal, que deve ser conectado em uma porta digital qualquer. No caso do PIR, o pino 1 é este pino de sinal (por onde ele nos dará a resposta binária da sua detecção, no caso, estado baixo se não detectar nada ou estado alto caso detectar), enquanto o pino 2 (central) é o 5V e o pino 3 é o GND. Em nosso exemplo, o pino de sinal será ligado na porta 6, enquanto o positivo do led será ligado à porta 5 (com seu o negativo em série com um resistor de 250 Ω). Deste modo, temos um circuito como o esquematizado a seguir:



Com a ligação eletrônica concluída, devemos programar o Arduino para que quando o PIR detectar movimento em seu entorno, o led indicativo ser aceso. Previamente, devemos informar que a porta de sinal do PIR (6) é uma porta de entrada (INPUT), enquanto a porta do led (5) é uma porta de saída (OUTPUT). Já no loop realizamos a leitura digital da porta

de sinal para detectar se o mesmo está baixo (neste caso desligamos o led) ou alto (neste caso, ligamos o led). Assim, nosso código será como o apresentado a seguir:

```
void setup(){
  pinMode(5, OUTPUT);
  pinMode(6, INPUT);
  digitalWrite(5, LOW);
}

void loop(){
  delay(1000);
  if(digitalRead(6) == HIGH) {
    digitalWrite(5, HIGH);
  } else {
    digitalWrite(5, LOW);
  }
}
```

Importante novamente ressaltar, que todos os sensores de resposta binária, utilizam exatamente a mesma lógica. Então se o PIR fosse substituído por um sensor de chama, ou um sensor de som, ou qualquer outro do mesmo tipo, mantida a porta 6 no pino de sinal, o código para a leitura do estado seria exatamente o mesmo, sem qualquer mudança. Então, vamos conhecer alguns dos principais sensores desta categoria.

Sensor de som



Este sensor nos informa quando um som em uma determinada altura é detectado e quando isso ocorre, modifica o seu estado para HIGH. A sua sensibilidade pode ser ajustada por meio de um trimpot presente em seu corpo e com isso podemos definir a altura mínima para que a detecção possa ocorrer (som mais alto ou mais baixo, de acordo com nossa necessidade).

Exemplos de projetos onde pode ser utilizado:

- Sistema de detecção de batidas na porta
- Sistema de ligamento / desligamento de luz ambiente com palmas
- Sistema de monitoramento de barulho em quartos hospitalares ou bibliotecas

Sensor de chama



Este sensor nos informa quando uma chama é detectada, modificando seu estado para HIGH quando isso acontece. Esta detecção é possível através da análise do registro da cor infravermelha emitida pela temperatura da chama e pelo seu movimento. Assim como no sensor de som, podemos regular a sensibilidade através de um trimpot. Alguns

sensores especiais deste tipo, possuem um segundo sensor de luz e um segundo trimpot onde é possível o ajuste da cor da chama a ser detectada (azul, laranja, amarela ou vermelha geralmente).

Exemplos de projetos onde pode ser utilizado:

- Sistema de detecção de incêndio
- Sistema de controle de tempo de uso de um fogão
- Sistema de monitoramento de apagamento accidental de chama (cortando o gás)

Sensor de vibração



Este sensor nos informa quando uma vibração com determinada força é detectada, modificando seu estado para HIGH quando isso acontece. Esta detecção é possível por conta de uma mola que existe dentro do seu receptáculo de metal, que vibra e toca as paredes permitindo a passagem de corrente. Também conta com um trimpot para ajuste da sensibilidade de vibração.

Exemplos de projetos onde pode ser utilizado:

- Sistema de monitoramento de vibração de máquinas (para evitar quebras)
- Sistema de detecção de quedas (de pessoas ou objetos)
- Sistema de contagem, frequência e ritmo de passos

Sensor de inclinação (TILT)



Este sensor é composto de uma cápsula com dois contatos internos em aberto e uma pequena bolinha de metal. Quando o mesmo é inclinado, após um determinado ângulo, a bolinha rola e toca os contatos, fechando o circuito e modificando o seu estado para HIGH.

Com isso, este sensor pode ser utilizado para, ao ser posicionado e ajustado à um determinado objeto, controlar quando uma inclinação máxima desejada for atingida. Conta com um trimpot para regular o tempo mínimo de contato da bola de metal para que o estado seja alterado.

Exemplos de projetos onde pode ser utilizado:

- Monitoramento de inclinação de uma cadeira de rodas para evitar quedas
- Sistema de detecção de capotagem ou quedas de maquinário ou automóveis
- Detecção de vibrações

Chave magnética (reed-switch)



A chave magnética do tipo reed-switch é composta de duas finas placas de metal internas a um bulbo de vidro, que ficam muito próximas entre si. Quando a chave é exposta a um campo magnético de força suficiente, as placas são atraídas ou repelidas e tocam-se, alterando o estado para HIGH neste momento. Assim, a chave reed-switch é utilizada para a detecção de campos magnéticos.

Exemplos de projetos onde pode ser utilizado:

- Detector de metais
- Detector de minerais polarizados (ímãs)
- Acesso a data center para evitar entrada com materiais magnéticos

Sensor magnético de proximidade



Este sensor é composto de duas partes: uma base (equipada com uma chave reed-switch interna) e um elemento móvel (que contém um ímã). Quando as duas estão próximas, as chapas internas do reed-switch são atraídas pelo ímã e o circuito fica fechado, mas ao serem afastadas, o estado do sensor é modificado para HIGH. Assim, fixando cada uma delas em elementos diferentes, conseguimos controlar quando os elementos são afastados.

Exemplos de projetos onde pode ser utilizado:

- Sistemas de alarme de invasão de residências (abertura de portas e janelas)
- Monitoramento de abertura de locais protegidos (cofres, gavetas, etc)
- Sistemas antifurto de peças de valor expostas

Push button



O push button é um tipo de botão muito popular em projetos eletrônicos por não guardar estado (ou seja, é diferente de um interruptor que clicamos uma vez para ligar e fechar o circuito e outra vez para desligar e abrir o circuito). Sua característica é alterar o estado para LOW quando é pressionado e para HIGH assim que é solto, detectando, deste modo, o clique.

Exemplos de projetos onde pode ser utilizado:

- Qualquer tipo de projeto que necessite detectar clique de botão para executar, ajustar ou programar algo.

Chave tátil capacitiva



Muito similar ao push button quanto à funcionamento: ao ser tocado, muda o seu estado para LOW e ao ser solto muda o seu estado para HIGH, detectando, desta forma, o clique. Sua única diferença é que ao invés de utilizar um dispositivo mecânico para esse acionamento (com partes móveis), utiliza um circuito eletrônico com um elemento capacitivo para conseguir detectar em que momento está sendo tocado.

Exemplos de projetos onde pode ser utilizado:

- Qualquer tipo de projeto que necessite detectar click de botão para executar, ajustar ou programar algo, assim com o push button.

Chave de 2 estados



A chave de dois estados é um botão similar a um interruptor pois guarda o seu estado. Porém suas ligações possibilitam ao mesmo tempo, dependendo da posição de dois pares distintos, ter um circuito aberto de um lado e fechado de outro, invertendo quando o botão for clicado novamente. Assim, podemos também utilizá-lo para detectar cliques, mas principalmente como chave liga-desliga de circuitos ou sub circuitos ou mesmo como chave de configuração, cujas combinações de estados representem diferentes ajustes.

Exemplos de projetos onde pode ser utilizado:

- Qualquer tipo de projeto que necessite ligar ou desligar um circuito ou sub circuito através de um botão ou que necessite de combinações de estado para configurações.

Chave de fim de curso



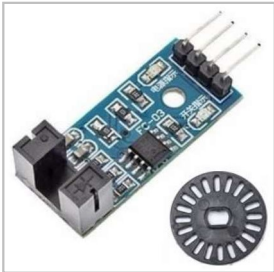
A chave de fim de curso possui um push button posicionado logo abaixo de uma haste móvel, que pode movimentar-se em apenas um eixo (para cima e para baixo). Quando algum objeto se aproxima e a empurra, em um determinado ponto, o push button é pressionado e o seu estado muda para HIGH. Assim, torna-se muito adequada quando necessitamos detectar quando um objeto chega a um determinado ponto que desejamos para pará-lo, desacelerá-lo ou mudar o seu curso. Impressoras, por exemplo, utilizam duas chaves deste tipo para que o cabeçote de impressão saiba quando chegou ao limite da folha.

Exemplos de projetos onde pode ser utilizado:

- Sistemas de monitoramento de aberturas (portas, janelas, gavetas)

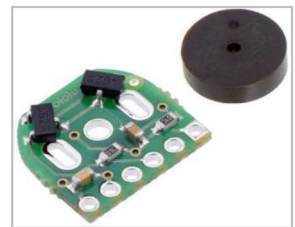
- Braços mecânicos ou sistemas de peças móveis para definir limites máximos
- Elevadores de carga para saber quando desligar o motor na subida e descida

Encoder ótico e encoder magnético



Os encoders são sensores capazes de transformar movimento mecânico em pulsos (geralmente através de sinais digitais, ou seja, estado LOW ou HIGH), que são utilizados para estimar posição, deslocamento ou velocidade. No encoder ótico, existe um feixe de luz que incide sobre um sensor de luz e quando algo passa por eles interrompendo o feixe, é contado um pulso modificando o estado para

HIGH (geralmente utiliza-se um disco perfurado ou um pino achatado, presos à parte móvel). Já no encoder magnético, existe um ímã que é preso na parte móvel e um dispositivo eletrônico com uma chave reed switch (ou um outro sensor magnético) que detecta quando este ímã passa por ele, alterando então o estado para HIGH. Ambos são muito úteis quando trabalhamos com motores, sistemas mecânicos com giros em eixos ou mesmo para estimar a frequência de funcionamento de máquinas e ferramentas com partes móveis.



Sensor de vazão



O sensor de vazão por efeito hall tem um funcionamento similar a um encoder magnético, pois internamente possui uma hélice que contém um pequeno ímã que força a disparada de um pulso a cada volta completa (mudando seu estado de LOW para HIGH). Assim, quando o ar ou a água passam por dentro do sensor, movem a hélice. Contar o total

de pulsos (voltas) em um segundo, nos permite estimar então o volume de ar ou água que passou pela válvula que passou pelo sensor.

Leitura de sensores de resposta binária com uso de interrupções

Apesar de praticamente todos os sensores digitais de resposta binária permitirem a leitura do seu estado a partir do método `digitalRead`, de forma simples, temos algumas exceções que necessitam de uma técnica um pouco mais elaborada. Eles são justamente os dois sensores vistos por último na relação anterior: os encoders e o sensor de vazão. Um botão, um sensor de som, um sensor de chama, etc, são ativados poucas vezes por segundo e além disso geralmente não nos interessa quantas vezes isso ocorreu. Portanto,

a técnica que utilizamos é dentro do loop, com um tempo de espera pequeno, ler a porta consecutivas vezes até perceber uma mudança de estado e, a partir daí, tomar uma ação. Já nos sensores do tipo encoder e no sensor de vazão, o total de mudanças de estado (chamadas de pulsos), é fundamental para estimar as grandezas que estes sensores medem (frequência, velocidade, rotação, vazão, etc). E é aí que temos um problema: não podemos ler o estado dentro de um loop e a cada vez que HIGH for detectado mudar o estado, pois não sabemos com que frequência isso deverá ser feito (e ela muda de acordo com a velocidade do que está sendo medido). E mesmo que a gente saiba, pode acontecer de uma mesma mudança de estado ser lida duas ou mais vezes apenas porque a velocidade