

# Aula 08: Aquisição de dados com microcontroladores

## Apresentação

---

Microcontroladores, quando ligados a sensores externos, ganham um novo mundo de aplicações, onde se destacam as aplicações de monitoração e controle. Todas as automações que você observa a seu redor, desde as mais simples, como o acender de uma luz com a percepção de movimento, até aquelas que, provavelmente, você ainda não experimentou, como carros autônomos, passam pela utilização de sensores ligados a microcontroladores.

Com base em exemplos de sensores presentes no simulador da plataforma Tinkercad, teremos uma ideia de como funções em C se aplicam com facilidade à programação de software básico com sensores e microcontroladores em sistemas de monitoração e controle.

## Objetivos

---

- Discutir o básico sobre aquisição de dados;
- Desenvolver programas em C para monitoração e controle;
- Experimentar o uso de sensores, transdutores e conversores analógico/digital (A/D).

## Conceito de sensores e transdutores

---

Para que um circuito ou sistema eletrônico realize qualquer tarefa ou função útil, ele precisa ser capaz de se comunicar com o “mundo real”, seja lendo um sinal de entrada de um interruptor on/off, seja ativando alguma forma de dispositivo de saída para acender uma luz, por exemplo. Em outras palavras, um sistema eletrônico ou circuito deve ser capaz de “realizar” algo, e sensores e transdutores são os componentes perfeitos para isso.

## Transdutor



A palavra transdutor é o termo coletivo usado para sensores – que podem ser usados para detectar uma ampla gama de diferentes formas de energia, como movimento, sinais elétricos, energia radiante e energia térmica ou magnética – e atuadores – que podem ser usados para alternar tensões ou correntes, “atuando” em um sistema.

## Sensor



Sensores analógicos estão presentes nos mais diversos equipamentos e sistemas. Eles produzem um sinal de saída contínua ou tensão que é geralmente proporcional à quantidade sendo medida.

Quantidades físicas como temperatura, velocidade, pressão, deslocamento, deformação etc. são todas grandezas analógicas, pois tendem a ser de natureza contínua. Por exemplo, a temperatura de um líquido pode ser medida usando um termopar, que responde continuamente, com variações de tensão, às variações de temperatura, na medida em que o líquido é aquecido ou resfriado.

Os sensores digitais produzem uma saída discreta representando um número ou dígito binário, como um nível lógico 0 ou um nível lógico 1.

## Exemplo

Um exemplo é o passive infrared sensor (PIR), que permite perceber movimento, quase sempre usado para detectar se um humano entrou ou saiu da faixa de sensores. Eles são pequenos, baratos, de baixa potência, fáceis de usar e não se desgastam. Por essa razão, eles são comumente encontrados em aparelhos usados em residências ou empresas.

O sensor em um detector de movimento é dividido em duas metades. A razão para isso é que estamos procurando detectar o movimento (mudança), e não os níveis médios de RI. As duas metades estão ligadas de modo que se anulam mutuamente. Se uma metade perceber mais ou menos RI do que a outra, situação típica de movimento, a saída será alta ou baixa. A figura 1 ilustra a atuação do sensor PIR.

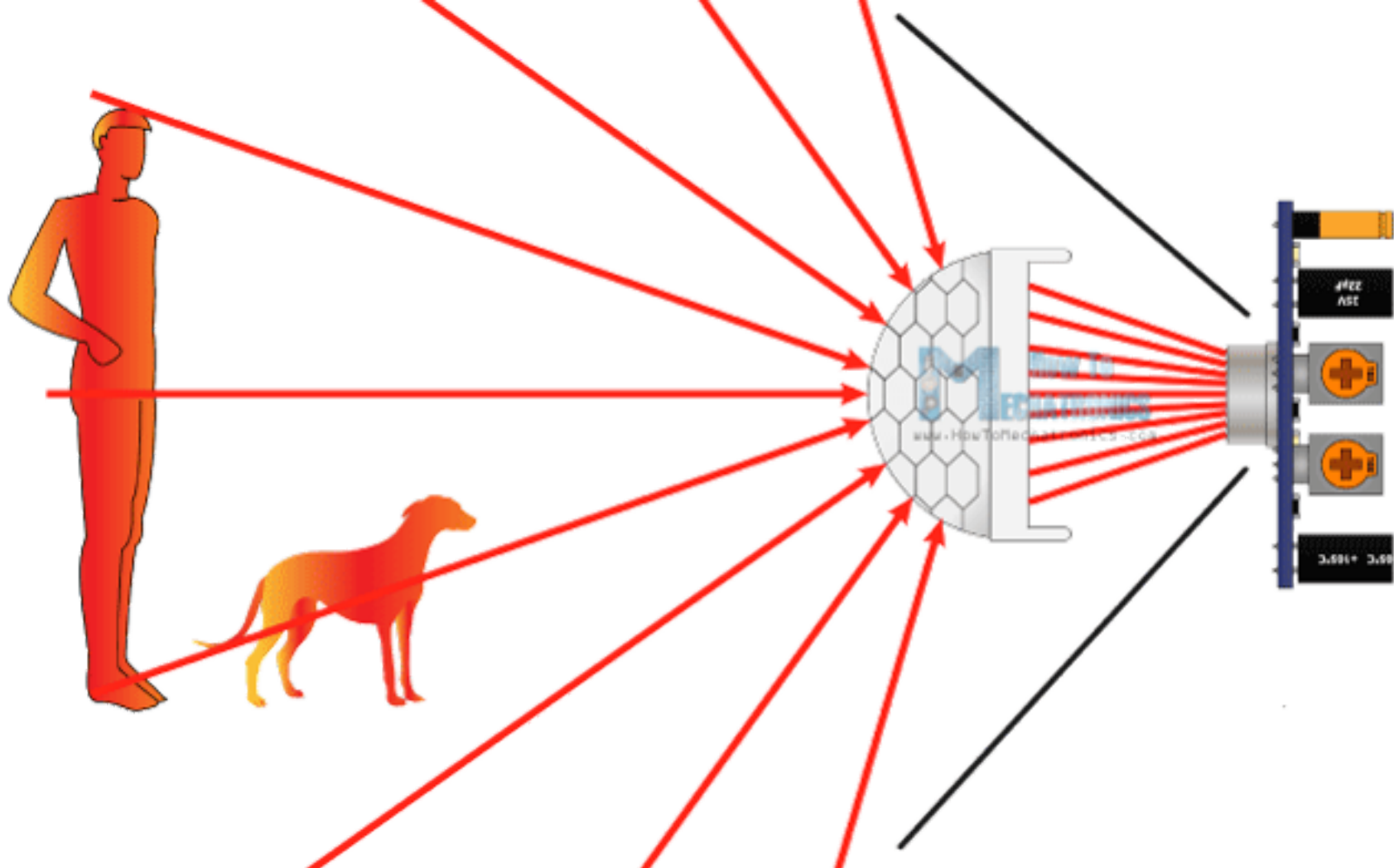


Figura 1. Atuação do sensor PIR, (Fonte: How to mechatronics).

## Sensor PIR com Arduino

A montagem da figura 2 mostra um exemplo de teste com o uso de sensores digitais com microcontroladores. Nessa figura, um sensor PIR está ligado a uma placa Arduino Uno. Os fios vermelho e preto ligados ao sensor PIR são a alimentação do sensor, retirada do Arduino. O fio amarelo é a saída do sensor PIR, que informa nível lógico alto quando percebe movimento.

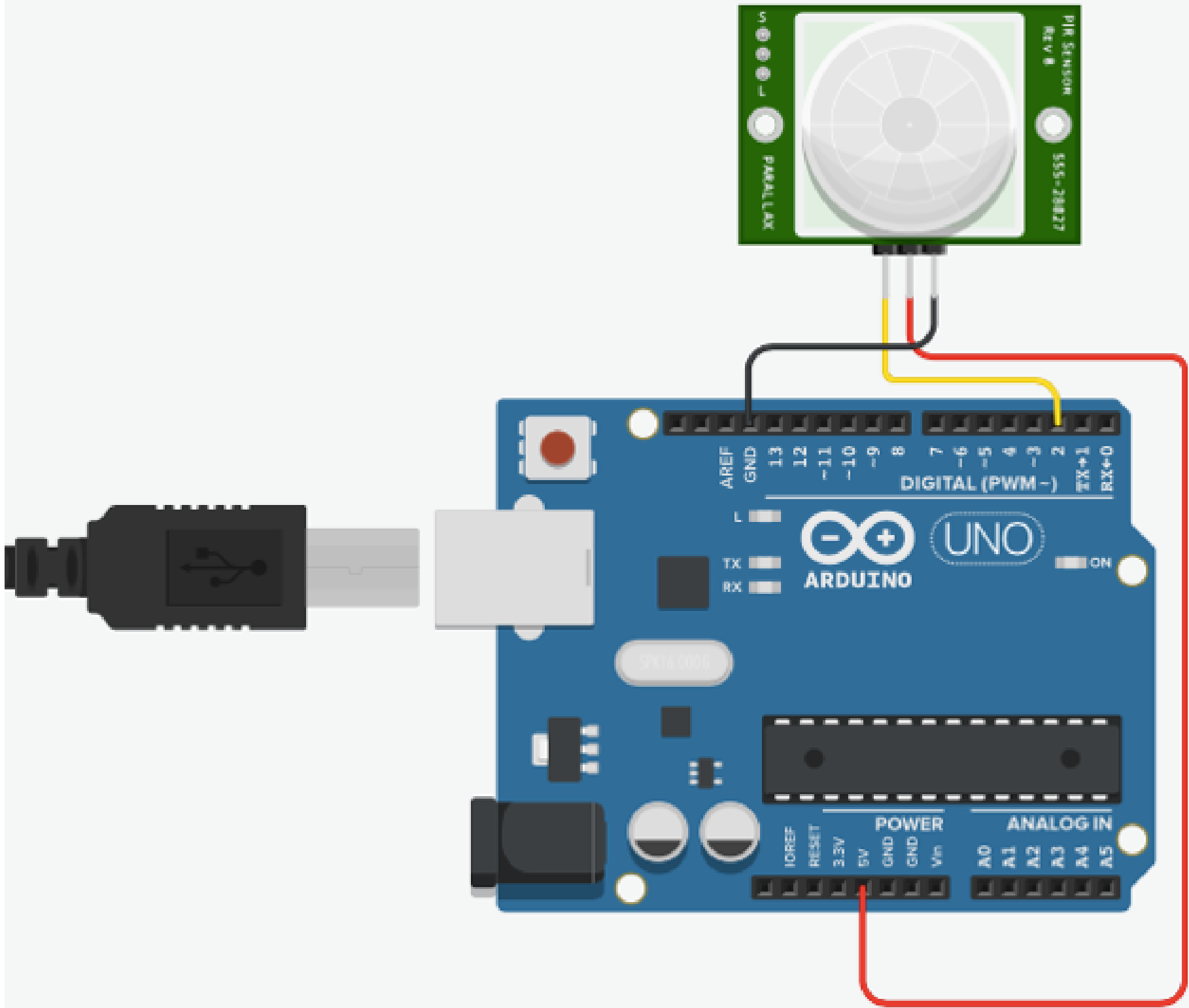


Figura 2 Montagem de sensor PIR com Arduino

O programa a seguir é usado para simular a aquisição de dados do sensor PIR.

TÍTULO

```
void setup() {
  pinMode(2, INPUT); //Pino 2 como entrada
  pinMode(13, OUTPUT); //Pino 3 como saída
}

void loop() {

  if (digitalRead(2) == HIGH) //Verifica se o sensor PIR está acionado
  {
    digitalWrite(13, HIGH); //Liga o LED da placa
    delay(100); //Espera 100 milissegundos
    digitalWrite(13, LOW); //Desliga o LED da placa
    delay(100); //Espera 100 milissegundos
  }
}
```

O exemplo pode ser copiado e editado na plataforma Tinkercad no link <https://www.tinkercad.com/things/jfAPRg9BkpV>.

A simulação da percepção de movimento pelo sensor PIR é feita clicando-se no centro do sensor e movimentando um círculo que aparece em área sombreada em sua frente, como pode ser visto na figura 3. Talvez seja necessário diminuir o zoom do campo onde é vista a montagem para ter acesso à área sombreada. Pelo código, a percepção de movimento faz com que o LED da placa pisque.

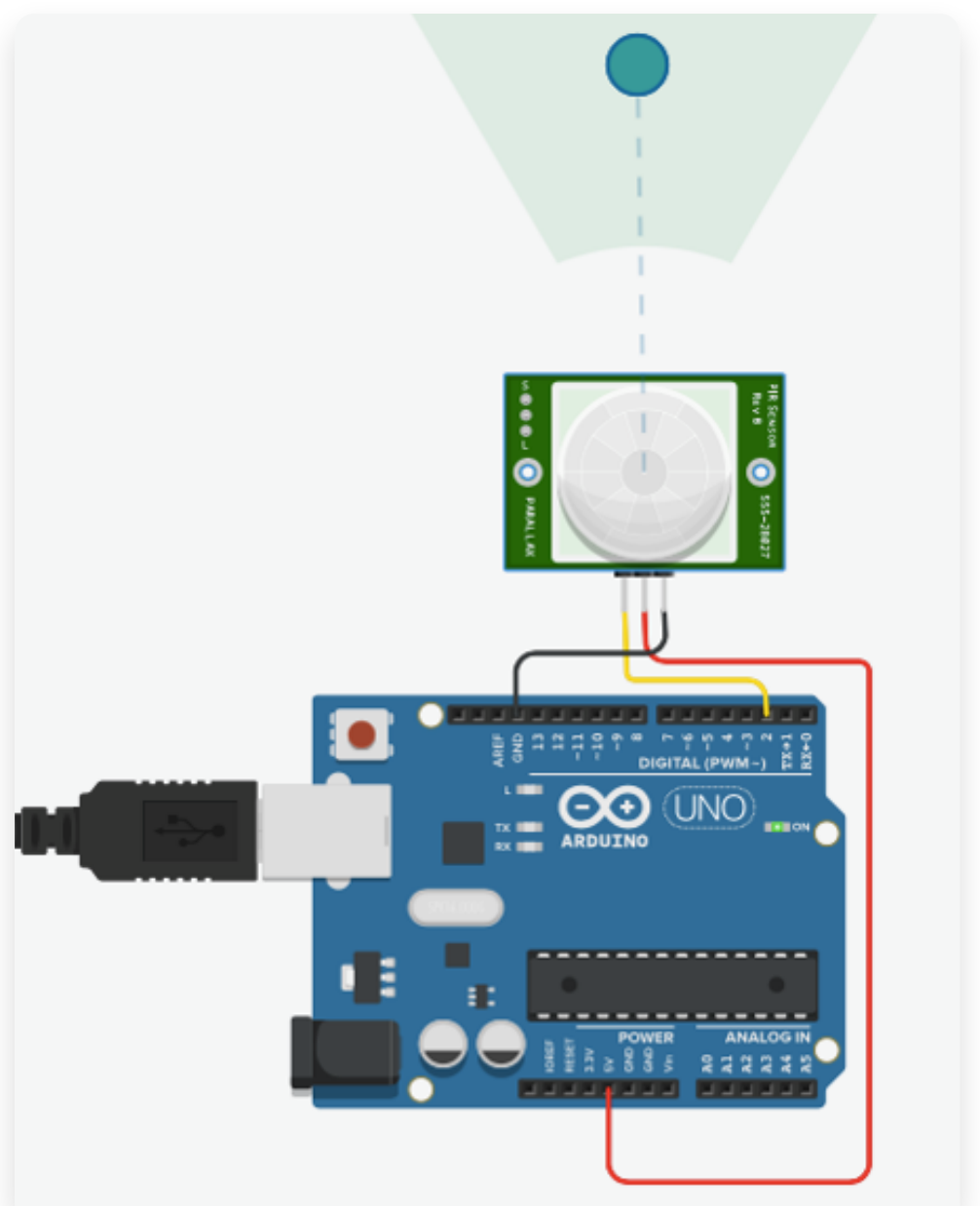


Figura 3 Sensor PIR sendo acionado na simulação no Tinkercad.

## Conversor A/D ou ADC

Microcontroladores são capazes de detectar sinais binários, como botão pressionado ou não. Estes são sinais digitais. Quando um microcontrolador é alimentado a partir de cinco volts, ele entende zero volt (0 V) como nível lógico 0 e cinco volts (5 V) como nível lógico 1.

Contudo, muitas vezes precisamos medir os sinais que variam, como a temperatura, os sinais analógicos que foram citados. Um sensor analógico de 5 V pode produzir 0,01 V ou 4,99 V, ou qualquer coisa entre eles. Felizmente, quase todos os microcontroladores têm um dispositivo embutido que nos permite converter essas tensões em valores que podemos usar em um programa para tomar uma decisão: Um conversor analógico para digital (ADC).

**Este é um recurso muito útil, que converte uma tensão analógica em um pino para um número digital. Ao converter do mundo analógico para o mundo digital, podemos começar a usar a eletrônica para interagir com o mundo analógico a nosso redor.**

# Princípio de funcionamento de um ADC

Como vimos, quase todos os parâmetros mensuráveis do ambiente estão em forma analógica, como temperatura, som, pressão, luz etc.

Considere um sistema de monitoramento de temperatura em que aquisição, análise e processamento de dados de temperatura de sensores não sejam possíveis com computadores e processadores digitais.

Esse sistema precisaria de um dispositivo intermediário para converter os dados analógicos de temperatura em dados digitais, a fim de se comunicar com os processadores digitais, como microcontroladores e microprocessadores.

01

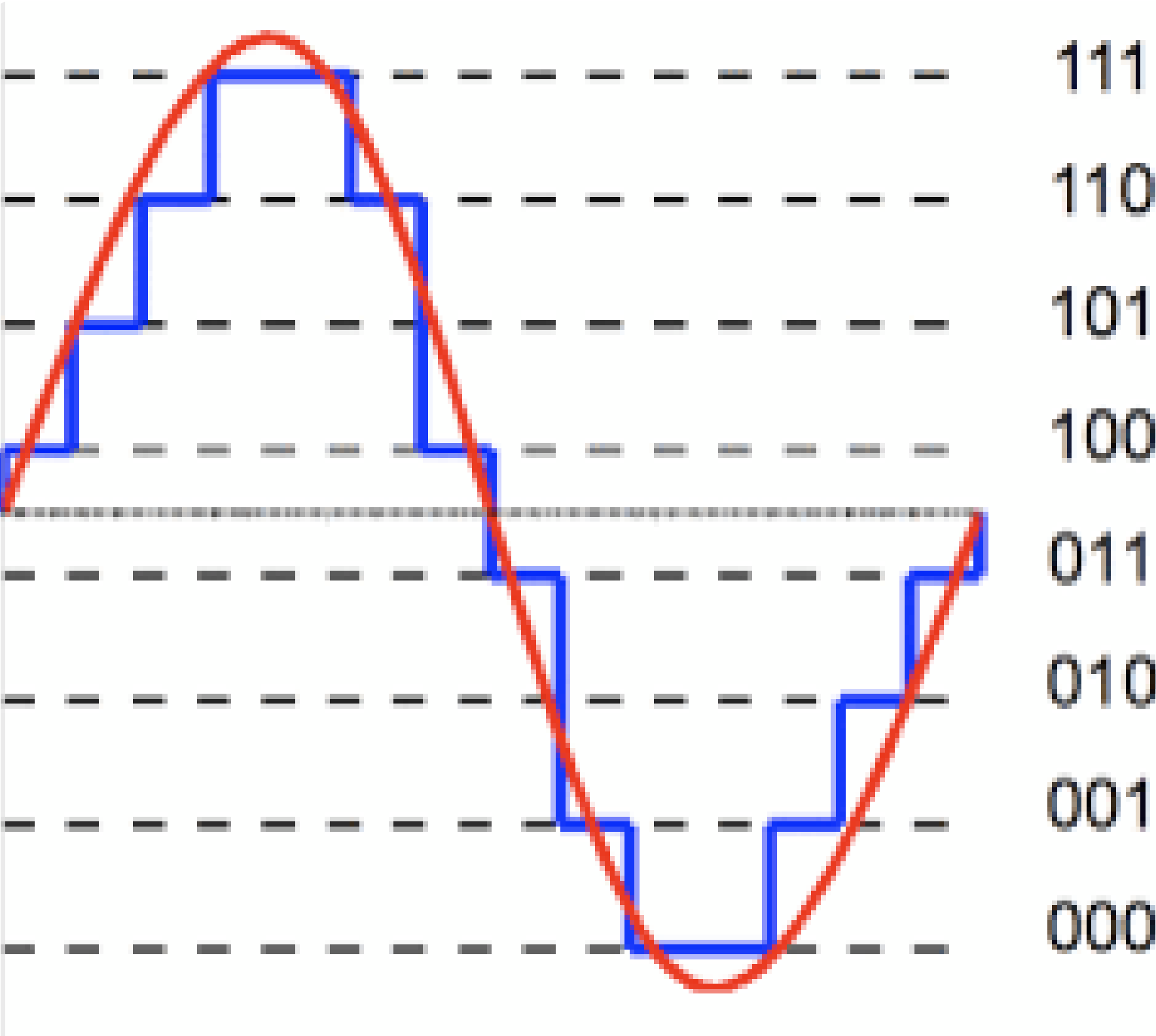
O ADC é um circuito eletrônico integrado usado para converter sinais analógicos, como tensões, em formato digital ou binário, consistindo em 1 s e 0 s. A maioria dos ADCs recebe uma entrada de tensão de 0 V a 10 V, -5 V a +5 V etc. e, de forma correspondente, produz saída digital como uma espécie de número binário.

02

O ADC faz uma amostragem do sinal analógico em cada extremidade decrescente ou ascendente do clock do sistema. Em cada ciclo, o ADC obtém o sinal analógico, mede-o e o converte em um valor digital. Esse circuito converte os dados de saída em uma série de valores digitais aproximando o sinal com precisão fixa.

03

Nos ADCs, dois fatores determinam a precisão do valor digital que captura o sinal analógico original: O nível de quantização (ou taxa de bits); e a taxa de amostragem. A figura 4 mostra como ocorre a conversão analógica para digital. A taxa de bits define a resolução da saída digitalizada e você pode observar onde o ADC de três bits é usado para converter o sinal analógico.



Suponha que um sinal seja convertido para digital usando o ADC de três bits com máximo de 1 V, como mostrado na figura 4. Portanto, um total de 23, isto é, oito divisões, está disponível para a produção de 1 V de saída. Isso resulta que 1/8, ou 0,125 V, é chamado como nível mínimo de alteração ou quantização, representado para cada divisão como 000 para 0 V, 001 para 0,125 V e 111 para 1 V.

Se aumentarmos as taxas de bits como 6, 8, 12, 14, 16 etc., obteremos uma precisão melhor do sinal. Assim, a taxa de bits ou quantização permite a menor alteração de saída no valor do sinal analógico que resulta de uma alteração na representação digital.

Suponha que o sinal é de cerca de 0 V-5 V e usamos um ADC de 10 bits. A saída binária de 5 V é 1023, e para 3 V é 614, como calculado por uma regra de três simples:

$$\begin{array}{l} 5 \text{ V} \rightarrow 1023 \\ 3 \text{ V} \rightarrow x \end{array}$$

Daí,  $x = (1023 \times 3)/5 = 613,8$ , que, arredondando, será 614 em valor binário.

## Comentário

Existe uma chance absoluta de deturpar o sinal de entrada no lado de saída, se for amostrado em uma frequência diferente da desejada, como pode ser percebido na figura 4. A linha azul mostra como fica a reconstrução do sinal senoidal com a frequência dada e com um ADC de três bits. Portanto, outra consideração importante do ADC é a taxa de amostragem.

O teorema de Nyquist afirma que a reconstrução de sinal adquirida introduz distorção, a menos que seja amostrada (no mínimo) com duas vezes a taxa do maior conteúdo de frequência do sinal. Assim, uma taxa de amostragem baixa demais causa uma reconstrução incorreta da forma de onda.

Em geral, é uma boa ideia fazer amostragem em uma taxa de cinco a dez vezes maior que a frequência do sinal para uma boa reconstrução, como mostrado na figura 5, que compara um sinal amostrado na mesma frequência do sinal original (100 Hz), com duas vezes a frequência e com 10 vezes a frequência do sinal.

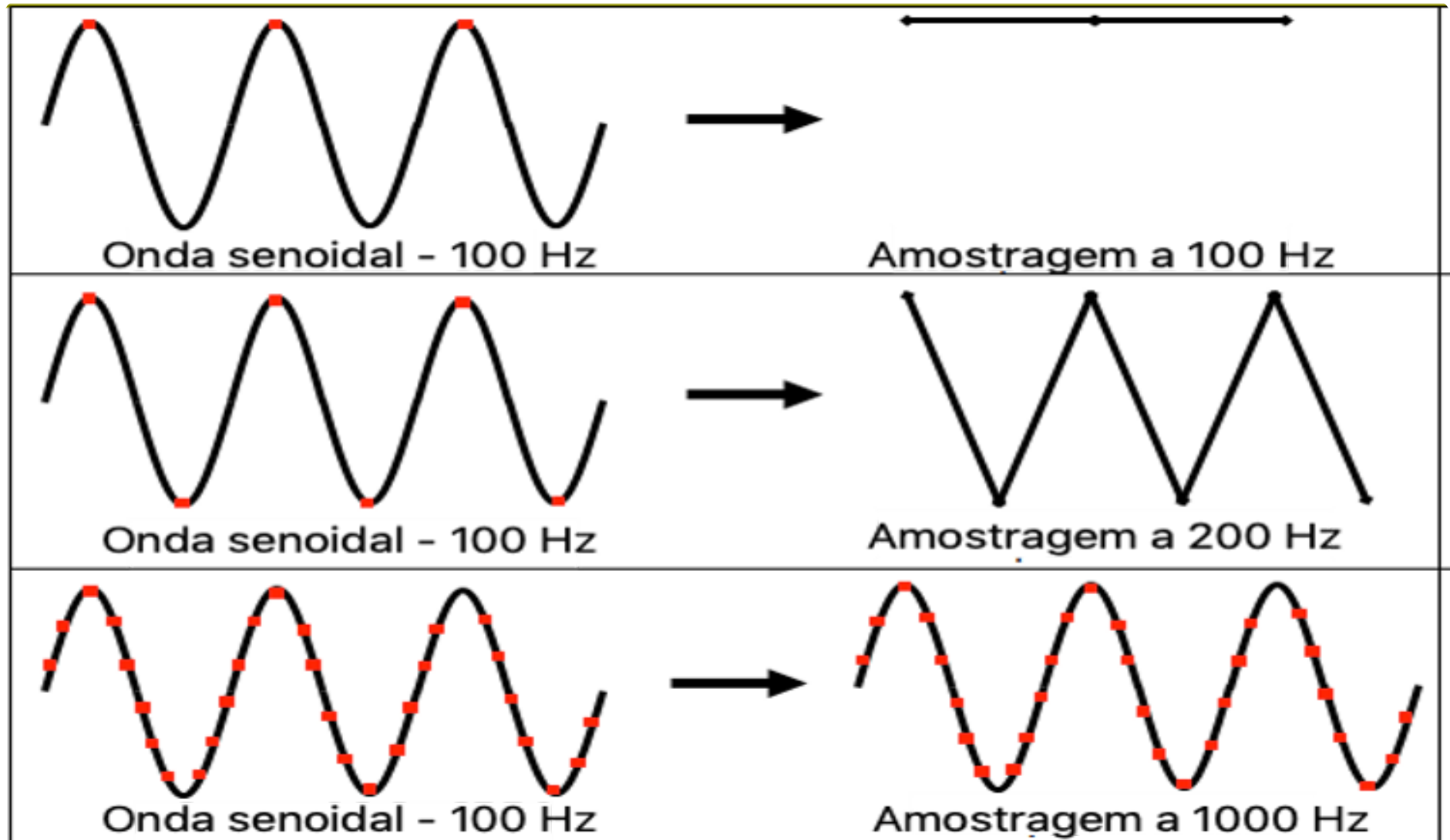


Figura 5. Avaliação de diferentes taxas de amostragem do sinal por um ADC. (Fonte: Adaptado de mashavuspirometer).

## ADC na plataforma Arduino Uno

Nem todos os pinos de um microcontrolador têm a capacidade de fazer conversões analógicas para digitais. Na placa do Arduino Uno são seis pinos, que têm um A na frente de sua etiqueta (A0 a A5).

ADCs podem variar muito entre os microcontroladores. O ADC no Arduino é de 10 bits, o que significa que tem capacidade para detectar 1.024 níveis analógicos discretos. O ADC, assumindo 5 V em sua entrada, converte para um valor binário de 0 a 1023, e qualquer coisa menor que 5 V será uma relação entre 5 V e 1023.

Para um Arduino de 16 MHz de clock de cristal, o clock para o ADC é definido como  $16 \text{ MHz} / 128 = 125 \text{ kHz}$ . Cada conversão no microcontrolador AVR leva 13 ciclos de clock do ADC, então  $125 \text{ kHz} / 13 = 9.615 \text{ Hz}$ .

Esta é a taxa de amostragem máxima possível, mas a taxa de amostragem real em sua aplicação depende do intervalo entre chamadas de conversões sucessivas.

Quando você lê o resultado e o envia pela porta serial, por exemplo, você recebe um atraso que aumenta à medida que a taxa de transmissão diminui. Quanto menor a taxa de transmissão, mais tempo levará para enviar o mesmo comprimento de dados e mais tempo levará para chamar a próxima conversão ADC.



Em uma montagem real, uma solução fácil para determinar a taxa real de conversão é alternar um pino digital toda vez que você executa uma conversão e medir a frequência com que o pino digital se alterna com um equipamento externo, o osciloscópio.

## Exemplos de uso de sensores e ADC

A montagem da figura 6 mostra um exemplo de teste com o uso de sensores digitais e ADC em sistema com microcontrolador. Nessa figura, um sensor PIR está ligado a uma placa Arduino Uno. Os fios vermelho e preto ligados ao sensor PIR são a alimentação do sensor, retirada do Arduino. O fio amarelo é a saída do sensor PIR, que informa nível lógico alto quando percebe movimento.

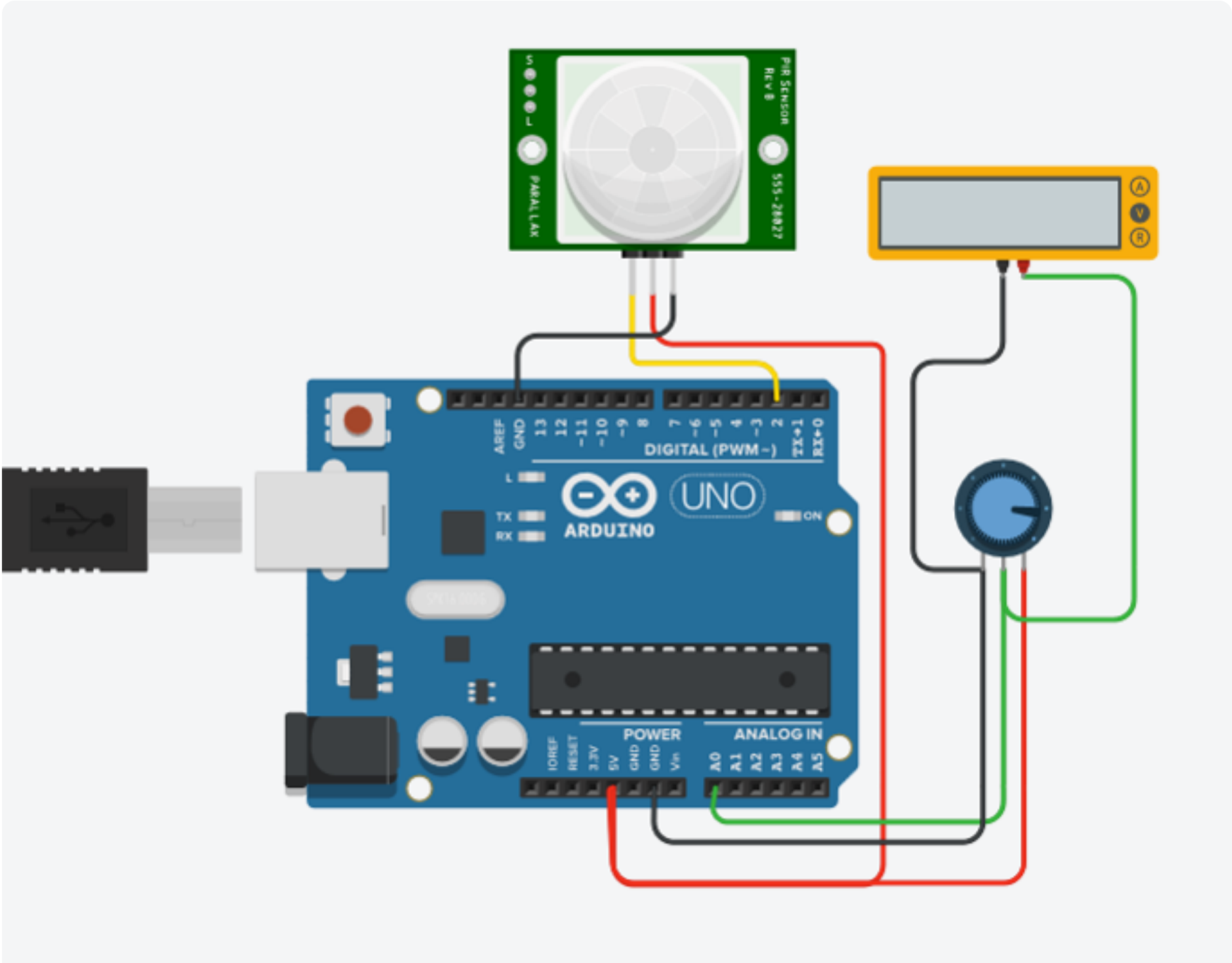


Figura 6 Montagem com sensor PIR, potenciômetro e medidor de tensão

Para testar a entrada do ADC, a ligação de um potenciômetro rotativo permite que se varie a tensão na entrada A0 entre 0 V e 5 V. Um medidor de tensão acrescentado mede a tensão no mesmo ponto, servindo para avaliar o ADC, verificando se a medida de tensão está sendo realizada corretamente pelo microcontrolador.

O programa para simular a aquisição de dados do sensor PIR e pelo ADC é mostrado a seguir. O valor de tensão, calculado a partir do valor fornecido pelo ADC, é enviado para a porta serial. Esse valor pode ser visto no simulador Tinkercad ao clicar em Monitor serial, abaixo do campo com o código.

```
void setup() {
  pinMode(2, INPUT); //Pino 2 como entrada
  pinMode(13, OUTPUT); //Pino 3 como saída
  Serial.begin(9600);
```

```
}

void loop() {
  if (digitalRead(2) == HIGH) //Verifica se o sensor PIR está acionado
  {
    digitalWrite(13, HIGH); //Liga o LED da placa
    delay(100); //Espera 100 milissegundos
    digitalWrite(13, LOW); //Desliga o LED da placa
    delay(100); //Espera 100 milissegundos
  }

  int sensorValor = analogRead(A0);
  //Converte a leitura analógica (que vai de 0 a 1023) para tensão (0 V-5 V):
  float tensao = sensorValor * (5.0 / 1023.0);

  Serial.println(tensao); //Manda dado pela porta serial
}
```

## Comentário

O exemplo pode ser copiado e editado na plataforma Tinkercad no link <https://www.tinkercad.com/things/9PcXqJobsjy-adc-e-sensor-pir>.

## Sensor de temperatura com Arduino

Quando usamos sensores analógicos, precisamos medir, em geral, uma tensão que é proporcional à quantidade sendo medida. Este é o caso dos sensores de temperatura.

O LM35 ou o TMP36, por exemplo, são sensores de temperatura linear analógicos. Isso significa que a tensão de saída é proporcional à temperatura. A tensão de saída aumenta em 10 mV para cada aumento de 1 °C na temperatura, em um range de -40 °C a +125 °C. A fórmula de conversão de tensão fornecida pelo sensor para temperatura, dada pelo fabricante, é:

$$\text{Temperatura Celsius} = [(tensão\ analógica\ em\ mV) - 500] / 10$$

O Arduino pode ler a entrada de 0 V a 5 V com suas entradas analógicas ligadas aos ADCs. O Arduino armazena isso como um número de 10 bits (0-1023). O sensor de temperatura pode ser montado como indicado na figura 7.

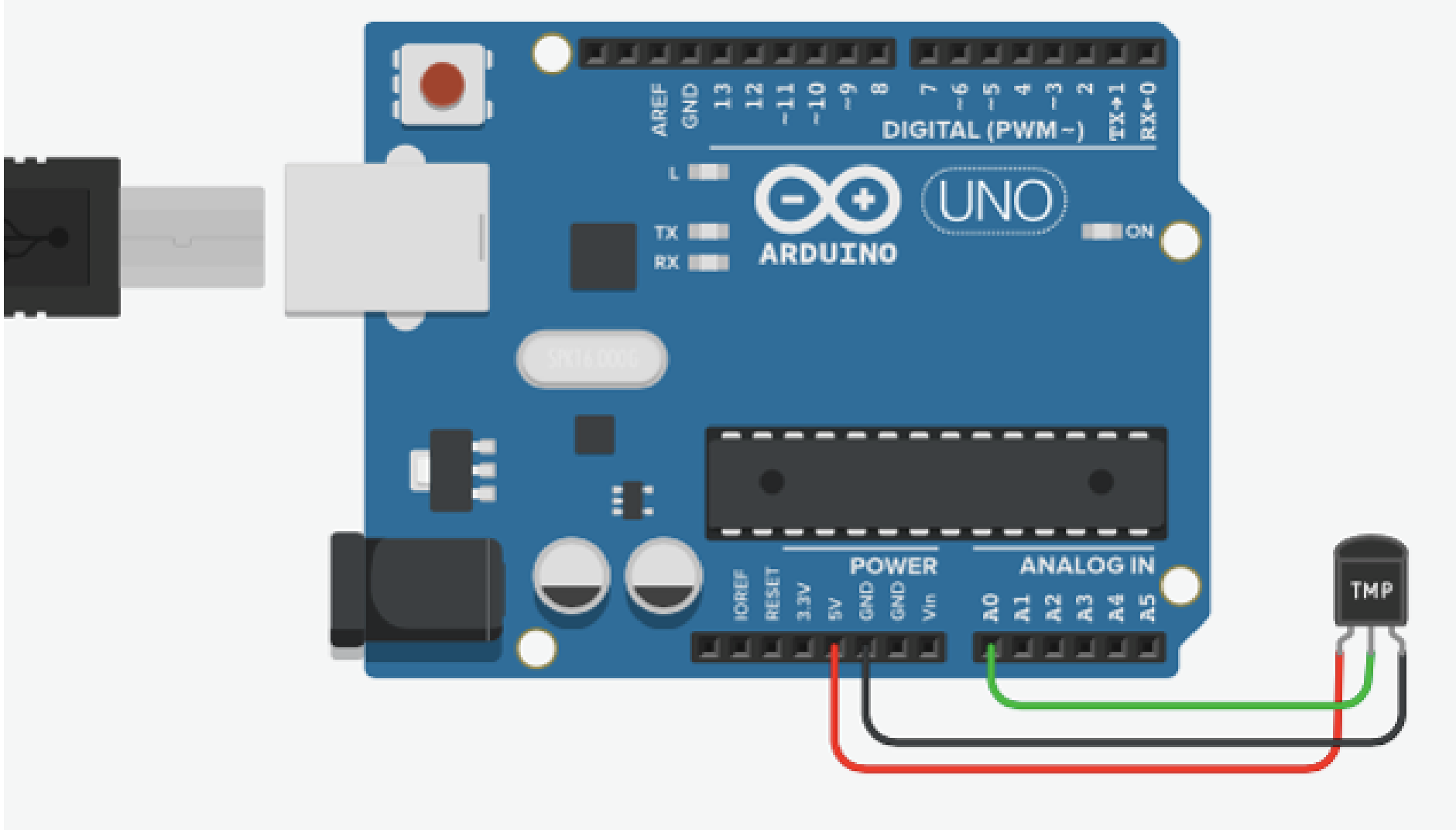


Figura 7 Sensor de temperatura ligado ao Arduino

O código a seguir mostra a aquisição dos dados do sensor e a conversão para temperatura, com envio do valor em graus Celsius para a porta serial.

```
char grau = 176; //Valor ASCII de grau
void setup()
{
    pinMode(A0, INPUT);
    Serial.begin(9600);
}

void loop()
{
    int tmp = analogRead(A0);
    float voltage = (tmp * 5.0)/1024; //(5*temp)/1024 para converter em tensão
    float milliVolt = voltage * 1000; //Para converter em milivolts

    float tmpCel = (milliVolt-500)/10 ; //Para sensor TMP36. Range (-40 °C a +125 °C)

    Serial.print("Celsius: ");
    Serial.print(tmpCel);
    Serial.println(grau);

    delay(1000);
}
```

Comentário

O código pode ser copiado e editado no Tinkercad pelo link <https://www.tinkercad.com/things/jjnow40JSiK>.

O valor de temperatura, simulado no sensor, pode ser alterado ao clicar nele durante a simulação no Tinkercad, modificando o valor como mostrado na figura 8.

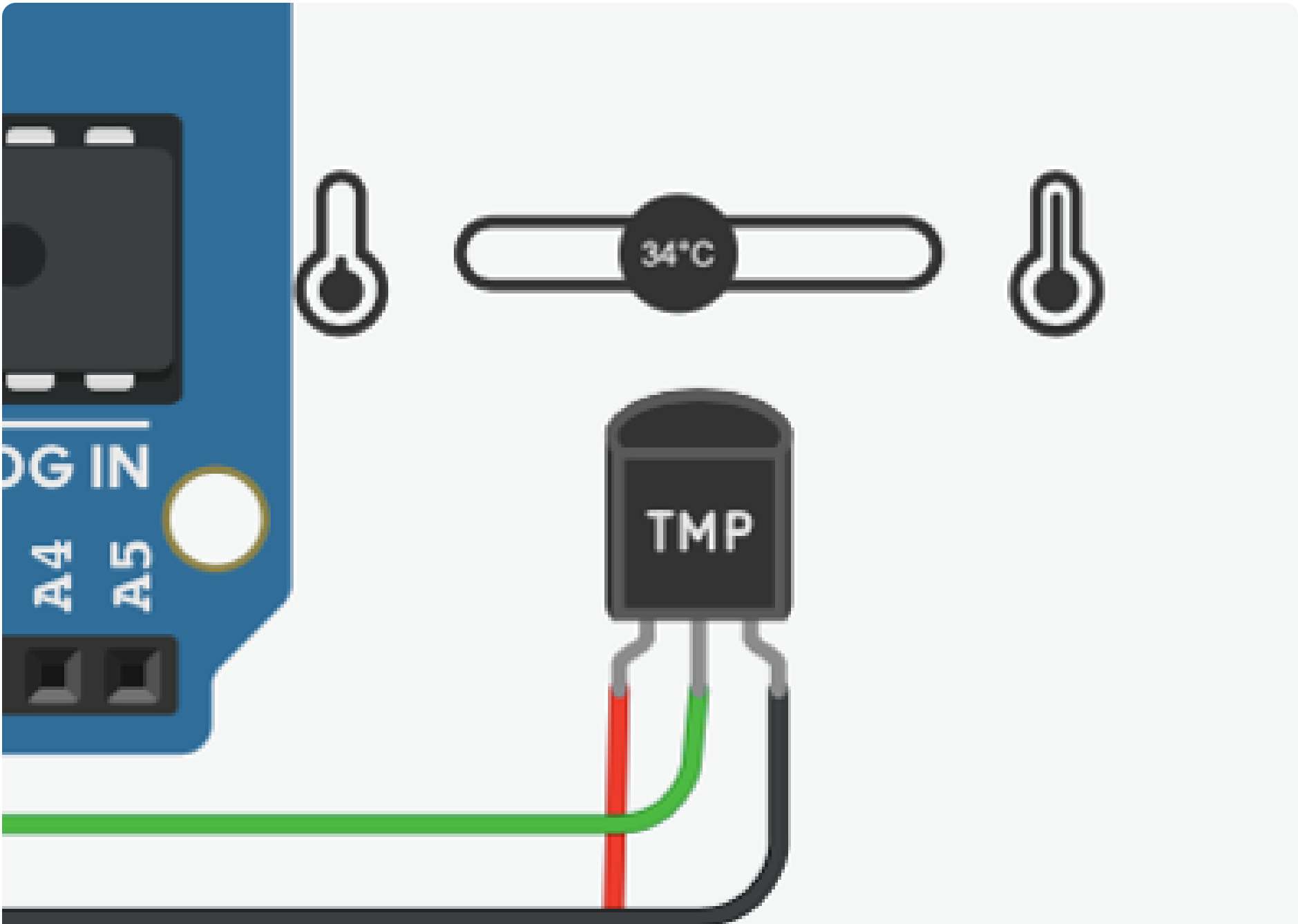


 Figura 8 Variação no valor de temperatura, simulada no Tinkercad

## Sensor de distância

Um sensor muito utilizado em projetos de robótica é o sensor ultrassônico HC-SR04. Ele usa sonar para determinar a distância de um objeto, assim como os morcegos. Oferece excelente detecção de alcance sem contato, com alta precisão e leituras estáveis em um pacote fácil de usar. Ele vem completo com módulos transmissores e receptores ultrassônicos. Veja o funcionamento na figura 9.

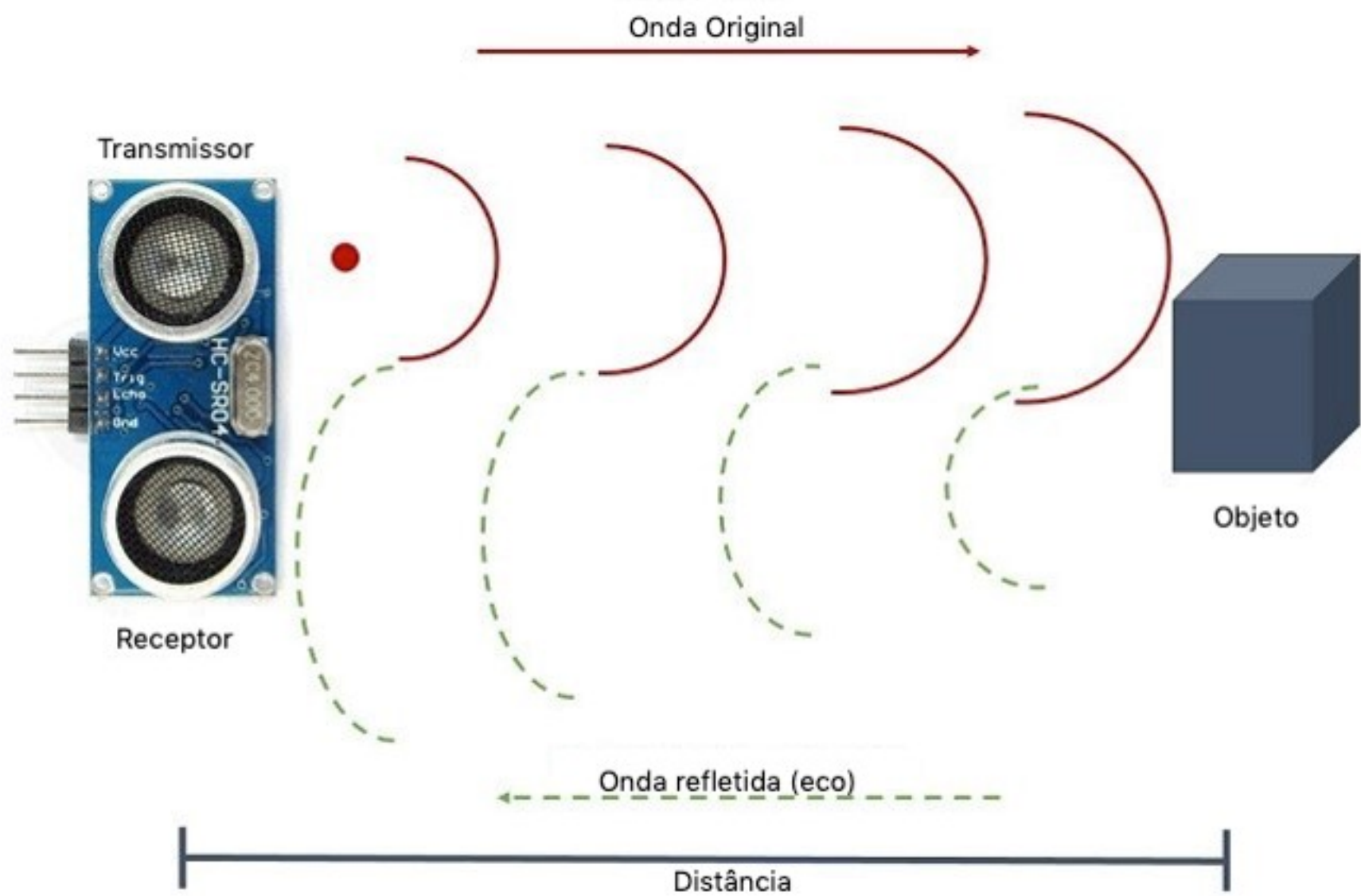


 Figura 9. Princípio de funcionamento do sensor de distância por ultrassom. (Fonte: Adaptado de How 2 electronics).

O transmissor (pino de trigger) envia um sinal, um som de alta frequência. Quando o sinal encontra um objeto, o receptor (pino de eco) o recebe o sinal refletido. O tempo entre a transmissão e a recepção do sinal nos permite calcular a distância até um objeto. Isso é possível porque sabemos a velocidade do som no ar, que é de aproximadamente 343 m/s.

## Atividade

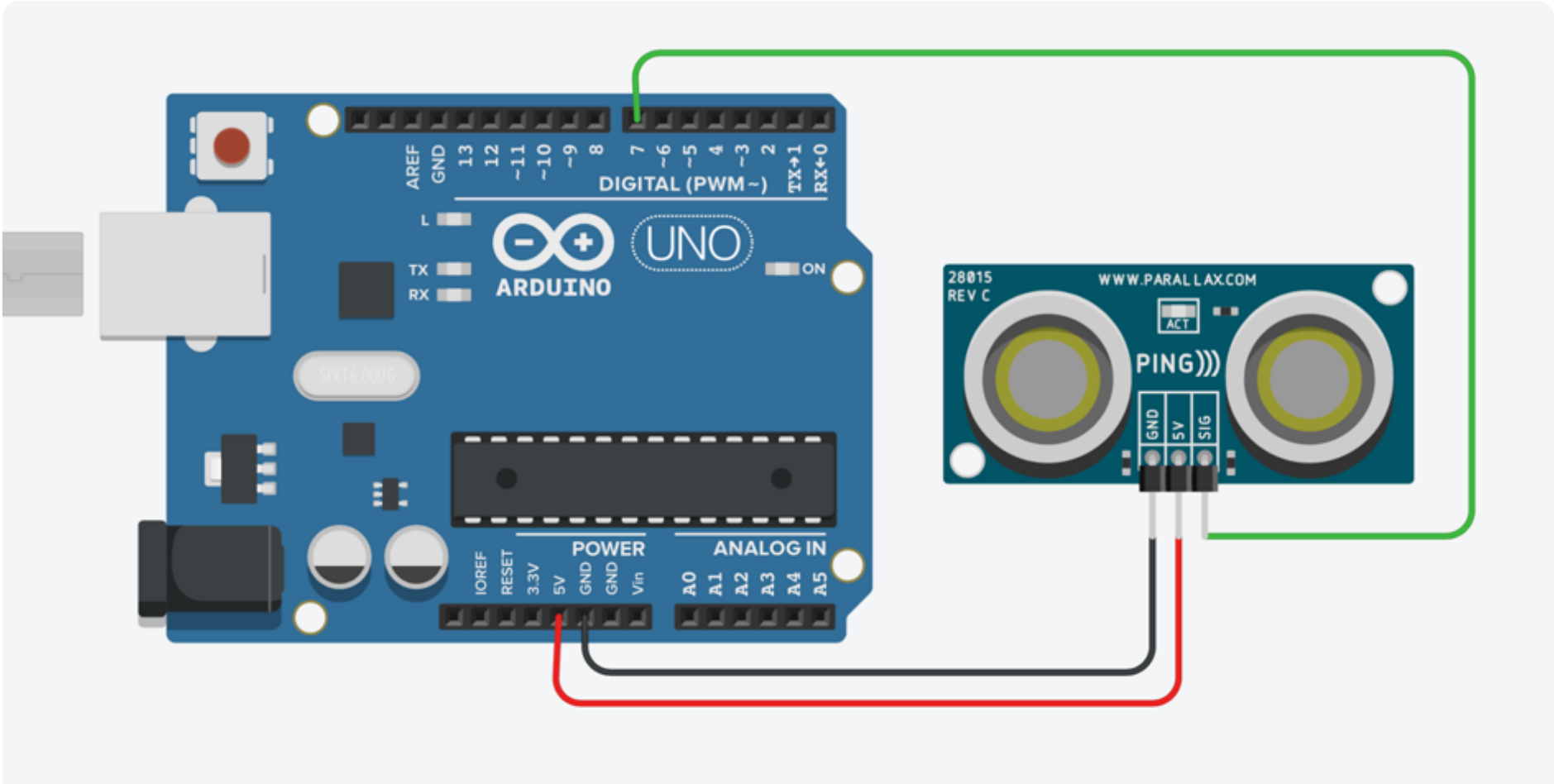
1. O ouvido humano mais sensível pode ouvir frequências até 20 kHz. Qual deveria ser a taxa de amostragem mínima do sinal de gravação de uma música para que tenha uma qualidade adequada de reprodução?
2. Qual é a resolução de um ADC de 12 bits em percentual da tensão de entrada?
3. O termopar tipo K é um termopar de uso genérico. Tem um baixo custo e, em razão de sua popularidade, está disponível em variados sistemas para medida de temperatura. Cobre temperaturas entre os -200 °C e os 1.200 °C, tendo uma sensibilidade de aproximadamente 41  $\mu\text{V}/^\circ\text{C}$ .

Suponha que o sistema de medida de temperatura, que foi solicitado para você criar a aquisição de dados, tenha que medir temperaturas positivas e que a temperatura de 0 °C gere 0 V no termopar (em um termopar real, não é assim, mas usaremos dessa forma para facilitar).

Qual o número de bits mínimo do ADC de que precisamos para podermos ter pelo menos 0,1 °C de resolução na medida, sabendo que o range de tensão de entrada do ADC pode ser ajustado múltiplos de 0,5 V.

4. Use um LED RGB no simulador Tinkercad (esse LED pode acender em vermelho, verde ou azul), alterando o programa para que acenda em verde para tensões menores ou iguais a 2 V e em vermelho para tensões maiores que 2 V. Essas tensões são lidas em uma entrada que varia com um potenciômetro, como no exemplo desta aula.

5. O sensor de distância do simulador Tinkercad usa o mesmo pino para transmitir e receber o sinal. A figura 10 mostra uma possível montagem. Pesquise e tente criar um programa, com essa montagem, para enviar a medida da distância pela porta serial.



Montagem do sensor de distância na plataforma Tinkercad

Notas

Título modal <sup>1</sup>

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

Título modal <sup>1</sup>

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

Referências

DI RENNA, Roberto Brauer; PAIVA, Lorraine de Miranda. **Tópicos especiais em eletrônica II:** Introdução ao microcontrolador Arduino. Niterói, 2014. Disponível em: [https://www.telacom.ufrr.br/act/nativa/downloads/anotacoes/arduino/anotacao\\_programacao\\_arduino.pdf](https://www.telacom.ufrr.br/act/nativa/downloads/anotacoes/arduino/anotacao_programacao_arduino.pdf). Acesso em: 20

[https://www.telecom.uff.br/pel/pelws/downloads/apostilas/arduino/apostila\\_de\\_programacao\\_arduino.pdf](https://www.telecom.uff.br/pel/pelws/downloads/apostilas/arduino/apostila_de_programacao_arduino.pdf). Acesso em: 29 dez. 2019.

**FORMATAÇÃO de fonte.** Disponível em: <https://paginas.fe.up.pt/~sam/Tele2/apontamentos/Amostragens.pdf>. Acesso em: 28 dez. 2019.

OLIVEIRA, Cláudio Luís Vieira; ZANETTI, Humberto Augusto Piovesana. **Arduino descomplicado**: Como elaborar projetos de eletrônica. São Paulo: Érica, 2015.

**TUTORIAL Arduino.** Disponível em: [//www.dsc.ufcg.edu.br/~joseana/IC\\_TutorialArduino\\_TinkerCad.pdf](http://www.dsc.ufcg.edu.br/~joseana/IC_TutorialArduino_TinkerCad.pdf). Acesso em: 28 dez. 2019.

## Próxima aula

---

- Comunicação cliente/servidor com a utilização de socket e protocolos UDP e HTTP.
- Processos em sistemas operacionais.

## Explore mais

---

O uso de sensores está em tudo o que nos cerca, você já sabe. Ligado a microcontroladores, como o Arduino, abre-se uma infinidade de aplicações possíveis de controle e monitoração. Uma das mais promissoras é nos carros autônomos. Os materiais a seguir mostram algumas dessas aplicações e referências sobre a variedade de sensores existentes.

- [Conheça os sensores do Arduino #ManualMaker Aula 6, Vídeo 1.](#)
- [Sensor.](#)
- [Apostila de Sensores.](#)
- [Como funciona a visão de um carro autônomo?](#)