

Emissor Infravermelho

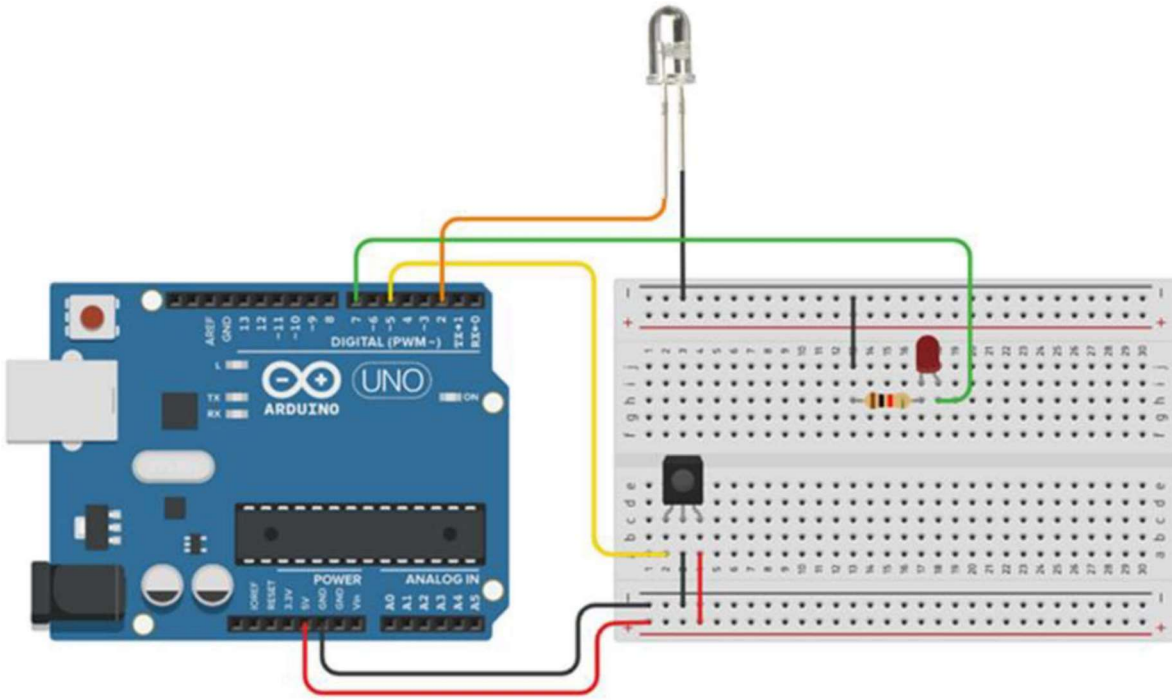


O emissor infravermelho é um fotodiodo capaz de emitir luz infravermelha, invisível à nós humanos, mas perceptível por câmeras, lentes e fotodiodos receptores. Quanto a aparência, não se difere de um led de alto brilho (com invólucro transparente) e o seu funcionamento básico é similar: alimentado com tensão, o mesmo “acende”. Esse componente é de fundamental importância e base dos chamados “controle remotos”. O controle remoto nada mais é do que um circuito eletrônico programado para que cada botão pressionado emita uma sequência de “piscadas” infravermelhas, com

diferentes tempos de duração e diferentes espaçamentos. Estas piscadas podem então ser percebidas por um receptor infravermelho, através do qual estimamos essas durações e espaçamentos. Assim, temos uma forma eficiente de enviar um sinal codificado através de um dispositivo e este sinal ser recebido por outro dispositivo com a capacidade de o decodificar. Isso torna possível programar ações diferentes no dispositivo que percebe o sinal para cada código recebido através da modulação infravermelha. Um controle de televisão de uma determinada marca, por exemplo, está preparado para emitir sinais com uma codificação para cada uma de suas teclas e as televisões dessa marca, por sua vez, estão preparadas para ler esses sinais e executar uma ação para cada um dos códigos de cada uma das teclas deste controle.



Não nos interessa aqui aprofundar em como ocorre essa codificação e decodificação, os tempos, frequências e durações dos pulsos. O interessante para nós, em nossos projetos de IOT, é que um controle remoto pode ser “clonado”, descobrindo o sinal que cada tecla emite. Isso permite duas coisas na prática após a clonagem: utilizar o controle remoto para controlar o Arduino ou usar o Arduino para controlar os dispositivos preparados para usar aquele controle remoto. Vamos ver estes dois processos em separado, até mesmo por utilizarem duas técnicas diferentes. Para os dois exemplos, vamos montar um mesmo circuito eletrônico de clonagem e teste como o abaixo:



Neste circuito, temos na parte de baixo um receptor infravermelho, responsável por perceber os sinais luminosos. Este sensor com 3 pinos (sinal, que ligamos na porta digital 5, VCC e GND) geralmente é mais adequado pois como conta com um pino adicional para alimentação, tem uma maior capacidade de percepção com sinais fracos. Porém, um receptor de 2 pinos também poderia ser utilizado, obrigando apenas que o controle seja pressionado mais próximo a ele. Acima, temos um emissor infravermelho de 5mm (existem também de 3mm, porém emitem sinais mais fracos e não podem ser alimentados diretamente com 5V) com a perna mais curta ligada ao negativo e a perna mais longa ligada à porta digital 2. Ele é o responsável por emitir sinais infravermelhos e através dele vamos tentar controlar algum equipamento que responda ao controle remoto clonado. Já na porta 7 temos um led que iremos utilizar para testar se através de um controle clonado conseguimos controlar o Arduino, ligando o led com uma de suas teclas e desligando com outra. Utilizaremos a biblioteca especial IRRemote que fornece diversos métodos para trabalhar com o envio, codificação e decodificação do sinal infravermelho. Esta biblioteca pode ser baixada no link: <https://bit.ly/arduinoqi>

Com isso, podemos começar os nossos experimentos. Aqui não é objetivo o aprofundamento nos códigos que descobrem as teclas ou geram as sequências e frequências de envio e codificação, mas sim ter um passo a passo do processo para posteriormente poder adaptá-lo e utilizá-lo em nossos projetos.

Clonando o controle remoto para controlar o Arduino

Neste primeiro experimento, iremos utilizar o circuito montado para descobrir através do receptor infravermelho os códigos enviados pelas teclas de um controle remoto quando pressionadas. A partir daí, vamos escolher duas delas para que liguem ou desliguem o led conectado na porta 7 do Arduino.

Passo 1: descobrir o código hexadecimal das teclas

Primeiramente, se desejamos “clonar” um controle para utilizar suas teclas de modo a comandarem o Arduino, necessitamos descobrir qual o código enviado em infravermelho por cada uma delas. Neste exemplo, vamos utilizar um controle remoto universal da LG para televisores, mas pode-se utilizar qualquer controle remoto desejado. Abra o Arduino IDE, cole o seguinte código e envie para a placa:

```
#include <IRremote.h>

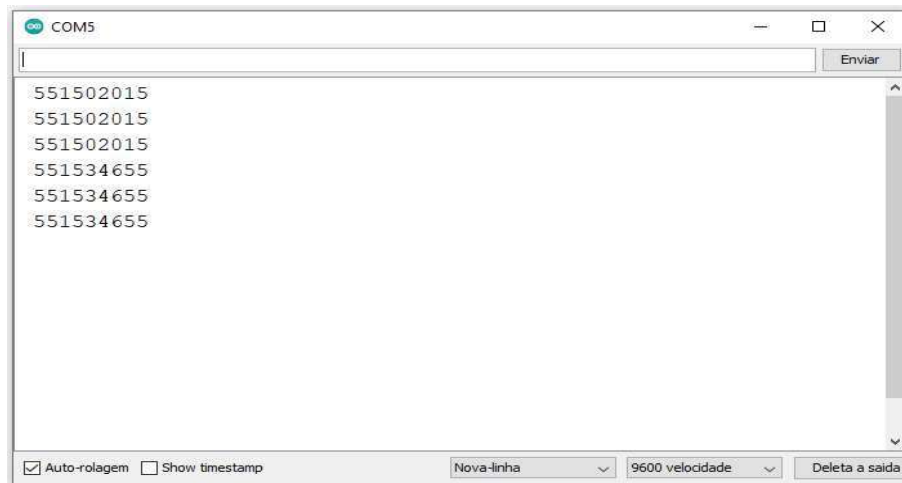
IRrecv receptor(5);
decode_results resultado;

void setup() {
  Serial.begin(9600);

  receptor.enableIRIn();
}

void loop() {
  if(receptor.decode(&resultado)) {
    Serial.println(resultado.value);
    receptor.resume();
  }
  delay(100);
}
```

Após, abra o monitor Serial e pressione rapidamente (sem segurar) as teclas do seu controle remoto. Para cada tecla pressionada será mostrado um número que é o resultado da decodificação daquela tecla. Para garantir que o código está correto, pressione algumas vezes a mesma tecla para verificar se os códigos conferem, pois é normal o aparecimento de ruídos que modifiquem o valor final lido. No exemplo abaixo, no controle de televisores LG, estes foram os códigos exibidos ao pressionar os botões volume para cima e volume para baixo:



Com isso descobrimos que sempre que pressionamos a tecla de volume para cima, recebemos um código 551502015 no Arduino e sempre que pressionamos a tecla volume para baixo recebemos um código 551534655.

Passo 2: usando o código descoberto das teclas para controlar o Arduino

Depois de mapear as teclas que desejamos, podemos criar um programa no Arduino que sempre que recebe um código infravermelho de uma tecla, testa qual o valor desse código e executa algo. Em nosso exemplo, iremos fazer com que sempre que a tecla volume para cima for pressionada (código 551502015) o led da porta 7 acenda e sempre que a tecla volume para baixo for pressionada (código 551534655) o led da porta 7 apague.

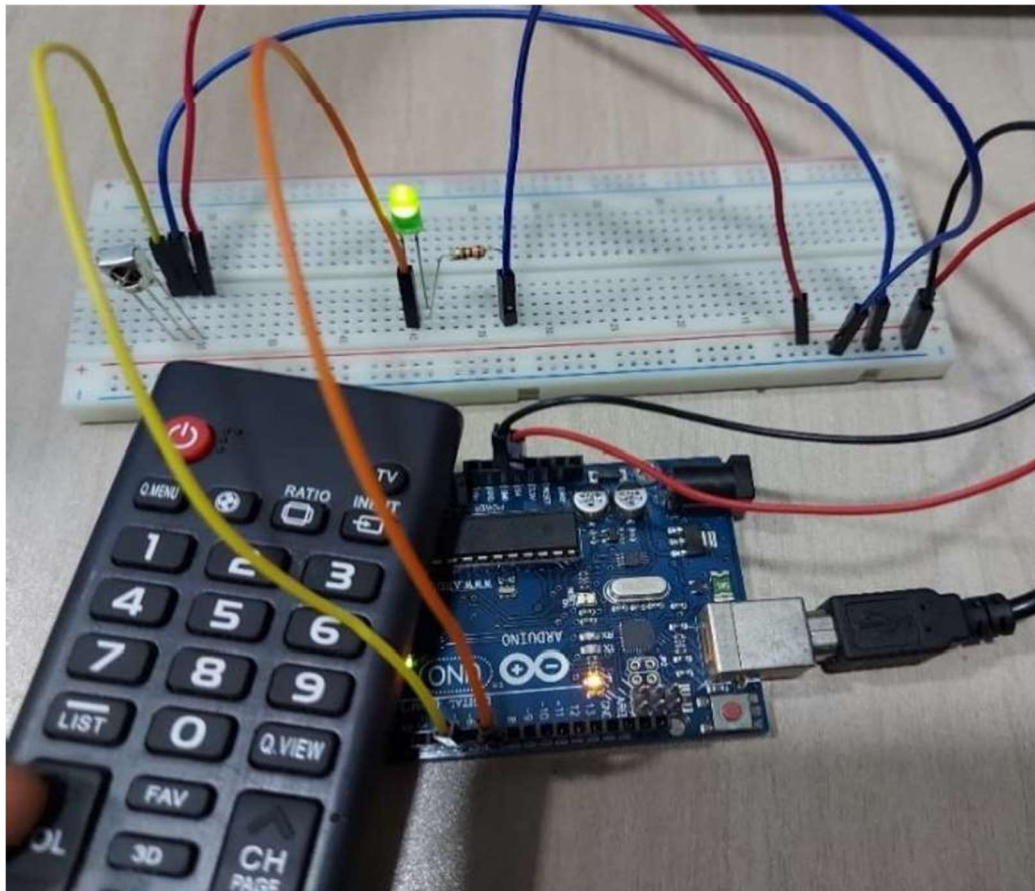
```
#include <IRremote.h>

IRrecv receptor(5);
decode_results resultado;

void setup() {
  Serial.begin(9600);
  pinMode(7, OUTPUT);
  digitalWrite(7, LOW);
  receptor.enableIRIn();
}

void loop() {
  if(receptor.decode(&resultado)) {
    if(resultado.value == 551502015){ //volume para cima
      digitalWrite(7, HIGH);
    }
    if(resultado.value == 551534655){ //volume para baixo
      digitalWrite(7, LOW);
    }
    receptor.resume();
  }
  delay(100);
}
```

Se tudo estiver certo e as teclas tiverem sido corretamente mapeadas, o led pode ser ligado e desligado com o controle. Abaixo, nosso exemplo com o controle da TV LG funcionando e com o led ligado através da tecla de volume para cima.



Clonando o controle remoto para controlar equipamentos

Neste segundo experimento, iremos utilizar o circuito montado para descobrir, (através do receptor infravermelho) as sequências enviadas pelas teclas e gerar automaticamente um código que consiga reproduzir (através do emissor infravermelho) exatamente as mesmas piscadas com as mesmas durações e frequências. Deste modo, conseguimos posteriormente com o Arduino reproduzir essas sequências infravermelhas através de seu emissor, se tornando o próprio Arduino um controle remoto capaz de operar dispositivos eletroeletrônicos.

Passo 1: gerar o código que reproduz a sequência de piscadas infravermelhas de cada uma das teclas

Primeiramente, precisamos realizar a clonagem, gerando os códigos que reproduzam as sequências infravermelhas exatas (piscadas de determinada duração e frequência) de cada tecla. Para isso, abra o Arduino IDE, cole o código abaixo e envie para a placa:

```
#define MAXPULSE 65000
#define RESOLUTION 20
#define IRpin_PIN PIND
#define IRpin 5
uint16_t pulses[100][2];
uint8_t currentpulse = 0;

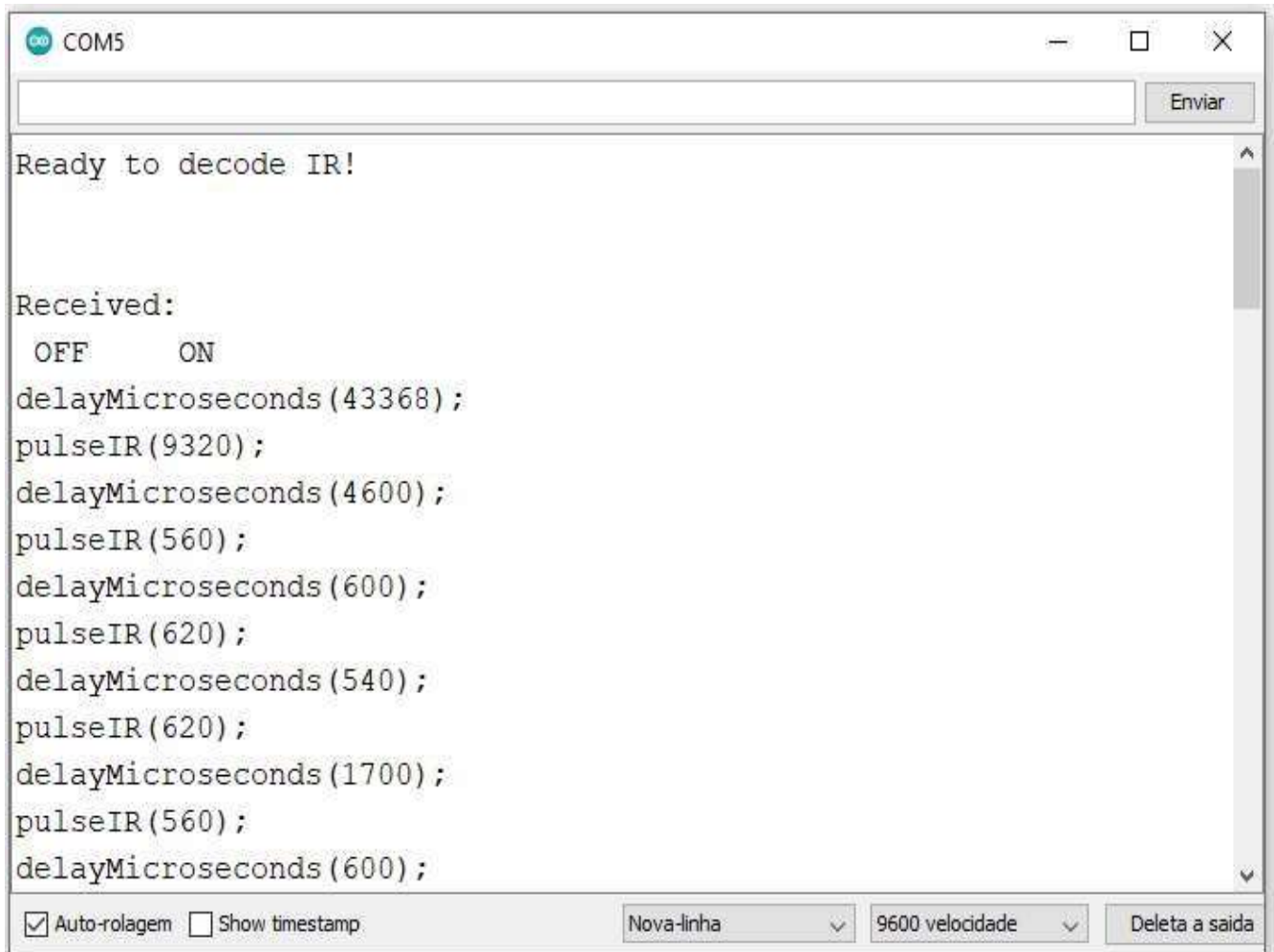
void setup(void){
    Serial.begin(9600);
    Serial.println("Ready to decode IR!");
}

void loop(){
    uint16_t highpulse = 0;
    uint16_t lowpulse = 0;
    while (IRpin_PIN & _BV(IRpin)) {
        highpulse++;
        delayMicroseconds(RESOLUTION);
        if ((highpulse >= MAXPULSE) && (currentpulse != 0)) {
            printpulses();
            currentpulse = 0;
            return;
        }
    }
    pulses[currentpulse][0] = highpulse;
    while (! (IRpin_PIN & _BV(IRpin))) {
        lowpulse++;
        delayMicroseconds(RESOLUTION);
        if ((lowpulse >= MAXPULSE) && (currentpulse != 0)) {
            printpulses();
            currentpulse = 0;
            return;
        }
    }
    pulses[currentpulse][1] = lowpulse;
    currentpulse++;
}

void printpulses(void) {

    Serial.println("\n\r\n\rReceived: \n\r OFF \t ON");
    for(uint8_t i = 0; i < currentpulse; i++) {
        Serial.print("delayMicroseconds(");
        Serial.print(pulses[i][0] * RESOLUTION, DEC);
        Serial.print(");\n");
        Serial.print("pulseIR(");
        Serial.print(pulses[i][1] * RESOLUTION, DEC);
        Serial.print(");\n");
    }
}
```

Agora abra o monitor Serial e pressione rapidamente (sem segurar) as teclas do seu controle remoto. Para cada tecla pressionada será gerado um código com diversas linhas, que reproduzem a emissão do infravermelho recebida. Copie esse código para cada tecla desejada (logo abaixo da linha OFF ON). Em nosso exemplo, vamos utilizar apenas a tecla liga/desliga.



The screenshot shows a serial monitor window with the title 'COM5'. At the top, there is an input field and an 'Enviar' button. The main area displays the text 'Ready to decode IR!' followed by 'Received:'. Below this, the text 'OFF ON' is shown. The rest of the window contains a series of Arduino code lines: `delayMicroseconds(43368);`, `pulseIR(9320);`, `delayMicroseconds(4600);`, `pulseIR(560);`, `delayMicroseconds(600);`, `pulseIR(620);`, `delayMicroseconds(540);`, `pulseIR(620);`, `delayMicroseconds(1700);`, `pulseIR(560);`, and `delayMicroseconds(600);`. At the bottom, there are checkboxes for 'Auto-rolagem' (checked) and 'Show timestamp' (unchecked), along with buttons for 'Nova-linha', '9600 velocidade', and 'Deleta a saída'.

```
Ready to decode IR!

Received:
  OFF    ON
delayMicroseconds(43368);
pulseIR(9320);
delayMicroseconds(4600);
pulseIR(560);
delayMicroseconds(600);
pulseIR(620);
delayMicroseconds(540);
pulseIR(620);
delayMicroseconds(1700);
pulseIR(560);
delayMicroseconds(600);
```

Passo 2: usando o código descoberto das teclas para controlar o Arduino

Depois de copiar os códigos gerados para cada tecla, vamos criar um programa no Arduino que execute esses códigos, testando se o emissor infravermelho, através deles, consegue controlar o equipamento do controle remoto clonado. Podemos, para melhor organização, criar um métodos para a execução destes códigos clonados de cada tecla. No nosso caso, vamos criar um método *ligarDesligar()* que receberá o código da tecla Liga/Desliga copiado anteriormente e dentro do loop, a cada 10 segundos, vamos chamar esse método, ligando ou desligado a televisão.

```
void setup(){
  pinMode(2, OUTPUT);
  digitalWrite(2, LOW);
}

void pulseIR(long microsecs){
  cli();
  while(microsecs > 0){
    digitalWrite(2, HIGH);
    delayMicroseconds(10);
    digitalWrite(2, LOW);
    delayMicroseconds(10);
    microsecs -= 26;
  }
  sei();
}

void ligaDesliga(){ //código gerado para a tecla Liga/Desliga colado no método
  delayMicroseconds(7984); pulseIR(9260); delayMicroseconds(4660); pulseIR(560);
  delayMicroseconds(600); pulseIR(560); delayMicroseconds(620); pulseIR(560);
  delayMicroseconds(1740); pulseIR(560); delayMicroseconds(600); pulseIR(580);
  delayMicroseconds(580); pulseIR(580); delayMicroseconds(600); pulseIR(560);
  delayMicroseconds(600); pulseIR(580); delayMicroseconds(600); pulseIR(560);
  delayMicroseconds(1740); pulseIR(560); delayMicroseconds(1760); pulseIR(560);
  delayMicroseconds(600); pulseIR(560); delayMicroseconds(1740); pulseIR(560);
  delayMicroseconds(1760); pulseIR(560); delayMicroseconds(1740); pulseIR(560);
  delayMicroseconds(1760); pulseIR(560); delayMicroseconds(1740); pulseIR(560);
  delayMicroseconds(600); pulseIR(560); delayMicroseconds(600); pulseIR(580);
  delayMicroseconds(600); pulseIR(560); delayMicroseconds(1740); pulseIR(560);
  delayMicroseconds(620); pulseIR(560); delayMicroseconds(600); pulseIR(560);
  delayMicroseconds(600); pulseIR(560); delayMicroseconds(620); pulseIR(560);
  delayMicroseconds(1740); pulseIR(560); delayMicroseconds(1740); pulseIR(580);
  delayMicroseconds(1740); pulseIR(560); delayMicroseconds(640); pulseIR(520);
  delayMicroseconds(1760); pulseIR(560); delayMicroseconds(1740); pulseIR(560);
  delayMicroseconds(1760); pulseIR(560); delayMicroseconds(1740); pulseIR(560);
  delayMicroseconds(41040); pulseIR(9200); delayMicroseconds(2420); pulseIR(520);
  delayMicroseconds(33524); pulseIR(9260); delayMicroseconds(2600); pulseIR(280);
}

void loop(){

  //a cada 10 segundos o Arduino liga ou desliga a TV
  delay(10000);
  ligaDesliga();

}
```

Se tudo estiver ok e o mapeamento realizado de forma correta, o Arduino conseguirá, através do seu emissor infravermelho, controlar o equipamento como se fosse o seu próprio controle remoto. Em nosso exemplo, na imagem ao lado, podemos ver a televisão ligada pelo Arduino através da clonagem do botão Liga/Desliga do controle remoto universal da LG. Como visto, mapeamos apenas uma tecla, mas adaptando nosso exemplo, podemos controlar totalmente um ou mais aparelhos de marcas diferentes com um único emissor infravermelho.

