

RECEPTOR DE INFRAVERMELHO ACIONADO POR CONTROLE REMOTO

1. Mapeamento do controle remoto

2. Acendendo e apagando leds remotamente

Esta aula visa mostrar como funciona o receptor de IR comandado por controle remoto, utilizando o Arduino UNO R3.

O acionamento pode ser via controle remoto de qualquer tipo de TV ou mesmo de leitor de DVD, bem como outros controles que transmitam dados através de IR.

O primeiro passo é associar as teclas do controle remoto ao código numérico, de forma que ao acionar a tecla escolhida no controle remoto é possível controlar qualquer dispositivo via Arduino.

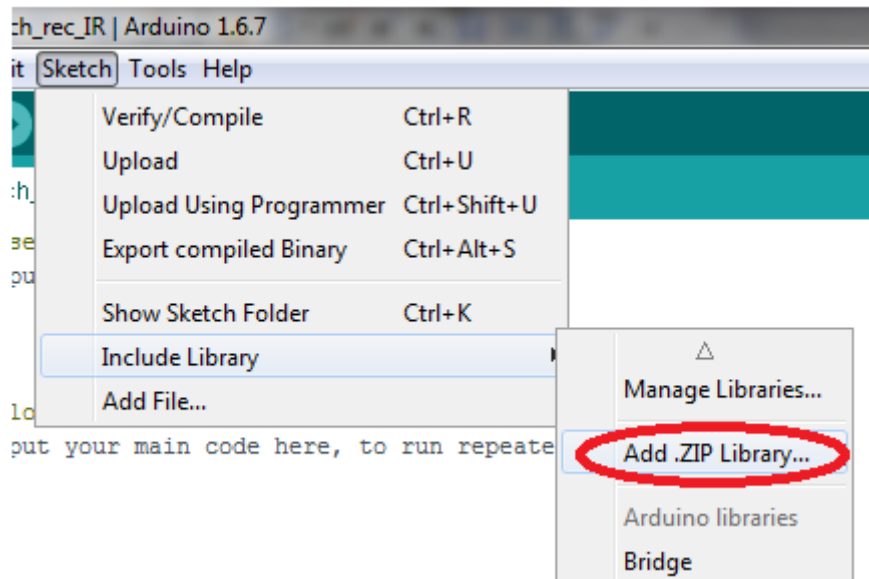


Para que o receptor de IR funcione no Arduino é preciso que seja usada a biblioteca para o referido dispositivo.

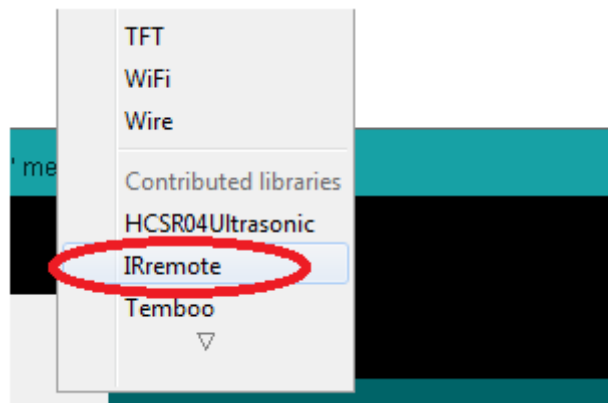
A biblioteca para esse dispositivo está disponível em diversos sites especializados na Web. Estarei disponibilizando além dos sketches essa biblioteca.

Para inserir a biblioteca no Arduino, siga os passos:

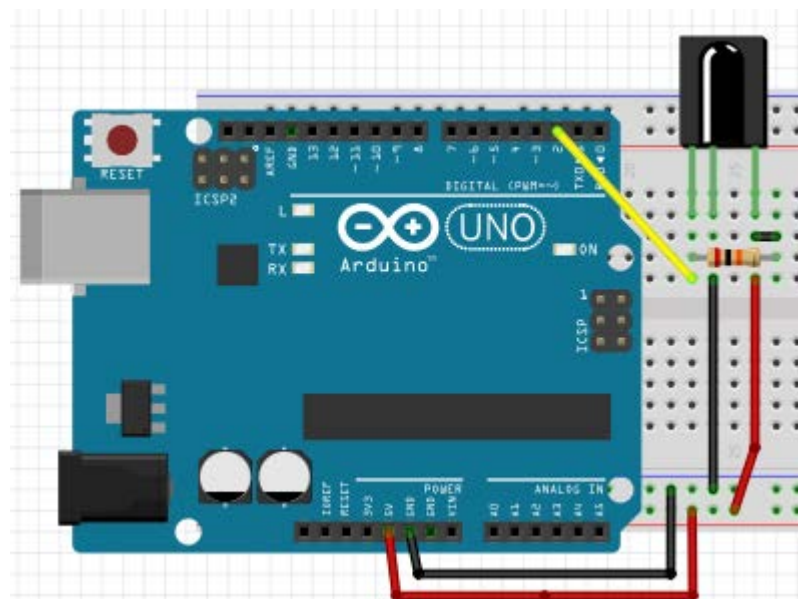
Arduino – Sketch – Include Library – Add .ZIPLibrary



A figura abaixo mostra a biblioteca do IRremote disponível para o desenvolvimento do projeto.



A figura a seguir mostra a montagem do circuito no Fritzing.



1. Mapeamento do controle remoto

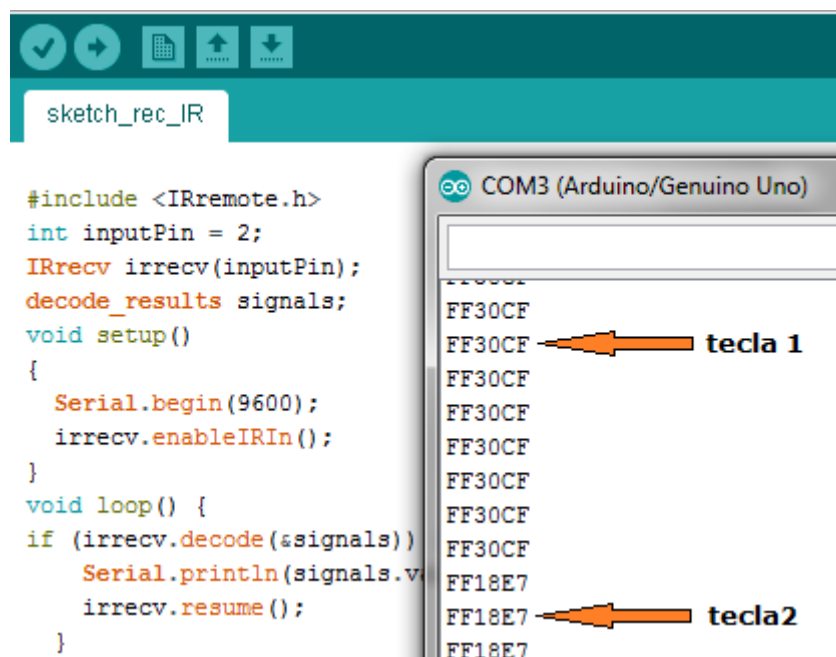
O próximo passo é mapear o controle remoto, ou seja, identificar os códigos para cada uma das teclas que se deseja usar.

Existem sites pela Web que fornecem esse mapeamento para vários modelos de controle remoto, mas, às vezes torna-se mais seguro fazer o mapeamento através de uma rotina bastante simples mostrada abaixo.

Sketch para verificação dos dados recebidos pelo Arduino a partir do controle remoto:

```
#include <IRremote.h> // inclui a biblioteca
int inputPin = 2; // define o pino do Arduino que vai receber o sinal do receptor IR
IRrecv irrecv(inputPin); // habilita o pino para receber o sinal do controle remoto
decode_results signals; // armazena a informação enviada pelo controle remoto
void setup()
{
  Serial.begin(9600); //inicia o serial monitor
  irrecv.enableIRIn();
}
void loop() {
  if (irrecv.decode(&signals)) {
    Serial.println(signals.value, HEX); //mostra o valor em hexadecimal no serial monitor
    irrecv.resume(); //recebe o próximo valor ao apertar uma tecla do controle remoto
  }
  delay(100); //ajusta o tempo para mostrar o valor no serial monitor (opcional)
}
```

A figura a seguir mostra o código hexadecimal das teclas 1 e 2 do controle remoto.



Veja a seguir os códigos HEXA das teclas 0 a 9 do controle remoto:

Tecla 1 = FF30CF
Tecla 2 = FF18E7
Tecla 3 = FF7A85
Tecla 4 = FF10EF
Tecla 5 = FF38C7
Tecla 6 = FF5AA5
Tecla 7 = FF42BD
Tecla 8 = FF4AB5
Tecla 9 = FF52AD
Tecla 0 = FF6897

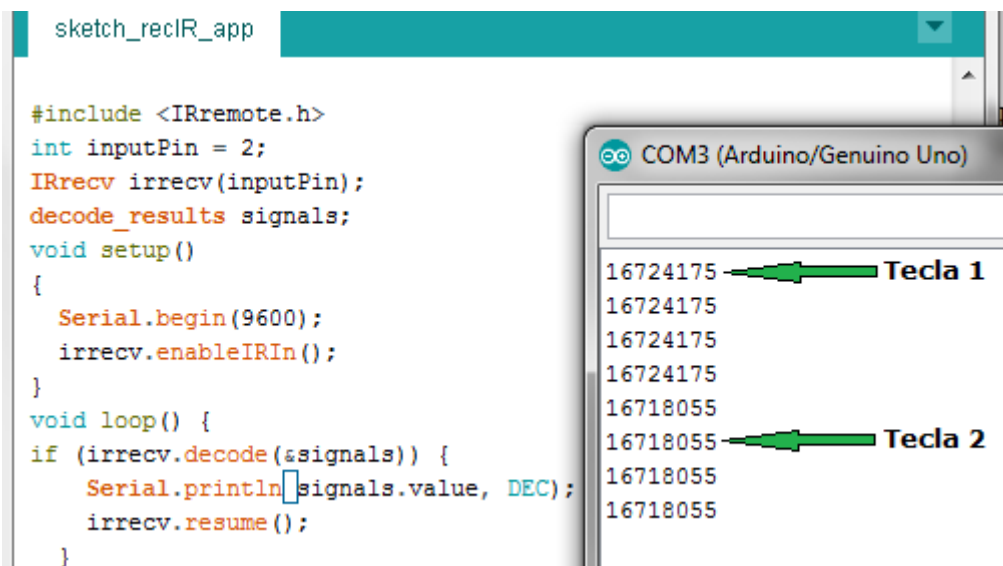
De maneira similar, pode ser visto o código de qualquer outra tecla do controle remoto, que não seja número, como por exemplo, PLAY/PAUSE, VOL+, etc.

Tecla Play/Pause = FFC23D
Tecla Vol + = FFA857

Os códigos podem ser visualizados em decimal (DEC) e em binário (BIN). Para isso basta modificar a linha:

```
void loop() {  
  if (irrecv.decode(&signals)) {  
    Serial.println(signals.value, DEC); // mostra o valor em decimal  
    irrecv.resume();  
  }  
}
```

Veja abaixo o valor em decimal para as tecla 1 e 2



```
sketch_reclR_app  
  
#include <IRremote.h>  
int inputPin = 2;  
IRrecv irrecv(inputPin);  
decode_results signals;  
void setup()  
{  
  Serial.begin(9600);  
  irrecv.enableIRIn();  
}  
void loop() {  
  if (irrecv.decode(&signals)) {  
    Serial.println(signals.value, DEC);  
    irrecv.resume();  
  }  
}
```

COM3 (Arduino/Genuino Uno)

16724175 → Tecla 1
16724175
16724175
16724175
16718055
16718055 → Tecla 2
16718055
16718055

Tecla 1 = 16724175
Tecla 2 = 16718055

Da mesma forma, podemos obter os códigos em binário, modificando a mesma linha:

```
void loop() {  
if (irrecv.decode(&signals)) {  
  Serial.println(signals.value, BIN); // mostra o valor em binário  
  irrecv.resume();  
}
```

Para as teclas 1 e 2 teremos então os códigos em binário:

Tecla 1 = 111111110011000011001111
Tecla 2 = 111111110001100011100111

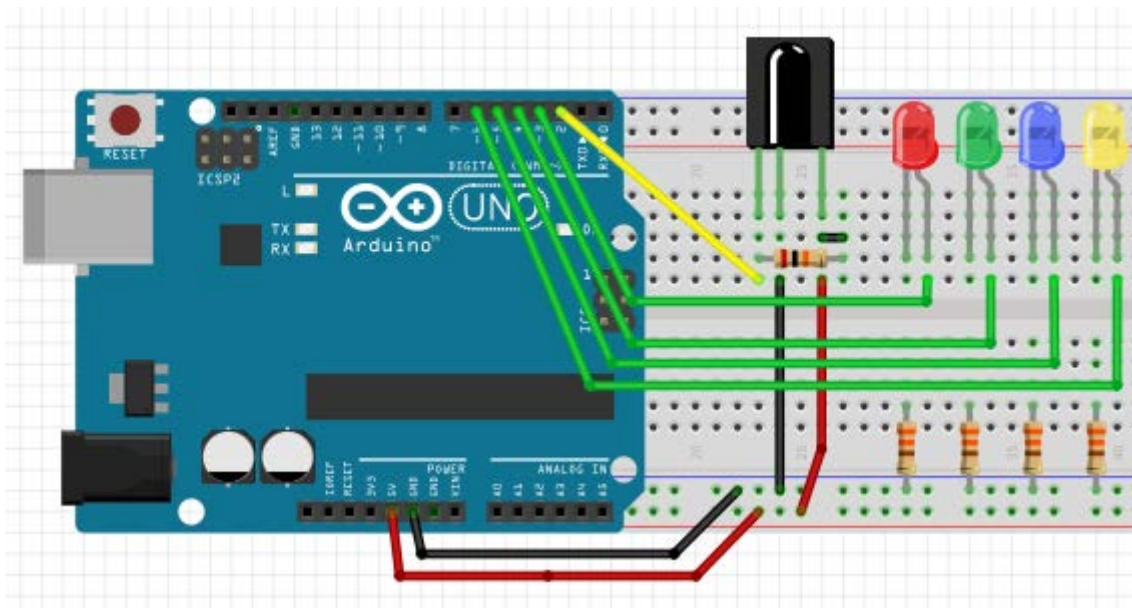
Concluindo: é mais prático utilizar a codificação em hexadecimal ou decimal do que em binário.

2. Acionando leds por controle remoto

Nesta segunda parte vamos acionar 4 leds com cores diferentes: Vermelho, Verde, Azul e Amarelo, e para cada um definiremos uma tecla do controle remoto.

Para isso é preciso modificar o sketch com os códigos de programação.

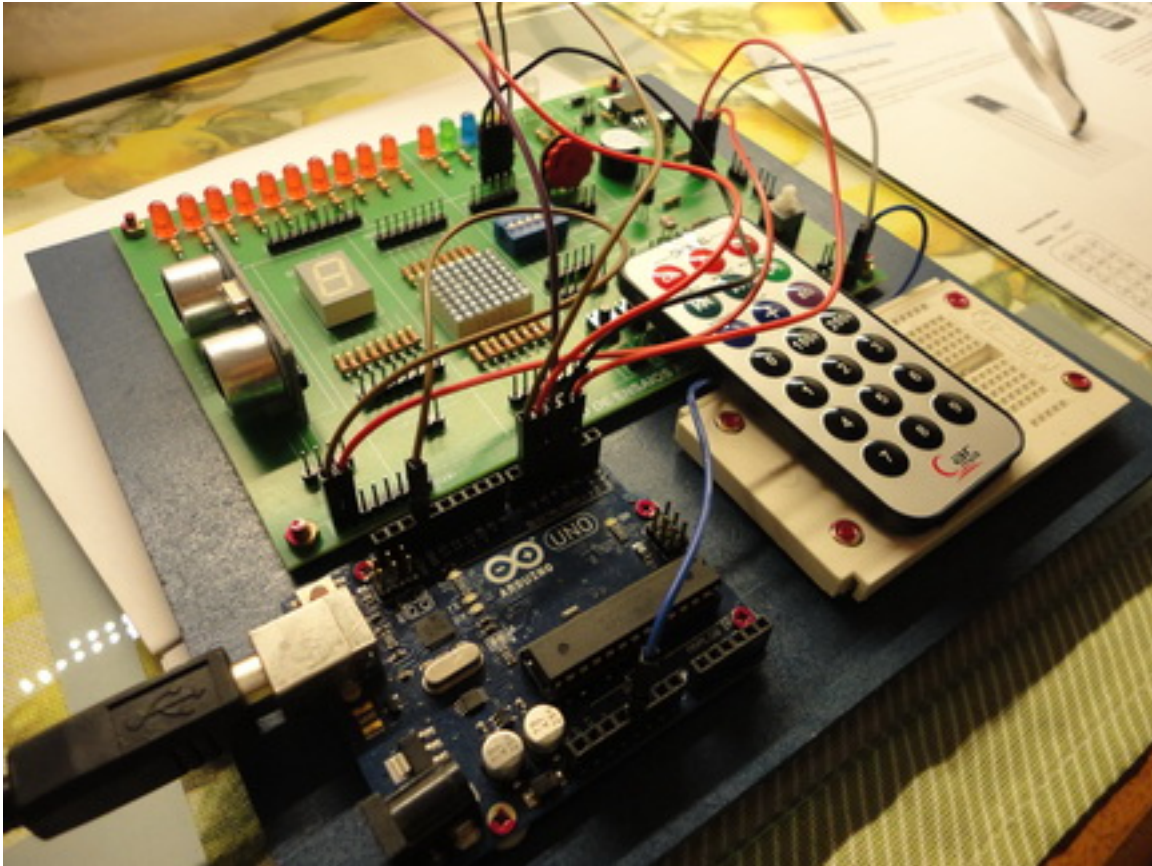
A figura a seguir mostra o projeto montado no Fritzing.



Tecla 1 – para acender o led vermelho
Tecla 2 – para acender o led verde
Tecla 3 – para acender o led azul
Tecla 4 – para acender o led amarelo

A tecla 0 servirá para apagar cada um dos leds, ou se todos estiverem acesos o acionamento da tecla 0 apagará todos de uma só vez.

Veja a montagem do projeto no Módulo de Ensaio Arduino:



As modificações necessárias são mostradas a seguir.

Na realidade nada mais é do que acrescentar as linhas de código necessárias.

As linhas acrescentadas estão em azul.

```
#include <IRremote.h>
int inputPin = 2;
IRrecv irrecv(inputPin);
  int ledVM=3;
  int ledVD=4;
  int ledAZ=5;
  int ledAM=6;
decode_results signals;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
  pinMode(ledVM,OUTPUT);
  pinMode(ledVD,OUTPUT);
  pinMode(ledAZ,OUTPUT);
  pinMode(ledAM,OUTPUT);
}
void loop() {
```

```

if (irrecv.decode(&signals)) {
  Serial.println(signals.value,HEX);
  irrecv.resume();
}
if (signals.value==0xFF30CF){
  digitalWrite(ledVM,HIGH);
  delay(500);
}
if (signals.value==0xFF6897){
  digitalWrite(ledVM,LOW);
  delay(500);
}
if (signals.value==0xFF18E7){
  digitalWrite(ledVD,HIGH);
  delay(500);
}
if (signals.value==0xFF6897){
  digitalWrite(ledVD,LOW);
  delay(500);
}
if (signals.value==0xFF7A85){
  digitalWrite(ledAZ,HIGH);
  delay(500);
}
if (signals.value==0xFF6897){
  digitalWrite(ledAZ,LOW);
  delay(500);
}
if (signals.value==0xFF10EF){
  digitalWrite(ledAM,HIGH);
  delay(500);
}
if (signals.value==0xFF6897){
  digitalWrite(ledAM,LOW);
  delay(500);
}
}

```

1. Observe que foi acrescentado 0x na frente do código HEX, caso contrário o Arduino não vai entender o código, além de provocar erro de compilação.
2. O delay de 500ms no fim de cada instrução "if" é opcional.
3. Se por exemplo, todos os leds estiverem acesos, mantendo o delay de 500ms ao acionar a tecla 0 do controle remoto os leds apagarão com intervalo de 1/2 segundo.
4. Se o delay for eliminado, os leds apagarão simultaneamente.

Utilizando controle remoto de TV – NET Digital HD com codificação decimal

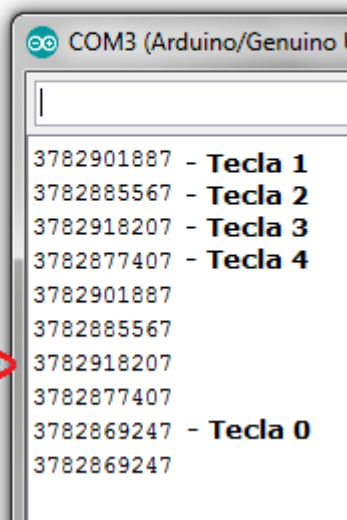
Para melhor exemplificar a versatilidade do projeto, veja abaixo o mesmo projeto porém com codificação decimal, utilizando um controle remoto de TV a Cabo – NET Digital.



Basta alterar a linha destacada em vermelho de HEX para DEC

```
sketch_reclR_cod1

#include <IRremote.h>
int inputPin = 2;
IRrecv irrecv(inputPin);
decode_results signals;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
}
void loop() {
  if (irrecv.decode(&signals)) {
    Serial.println(signals.value, DEC);
    irrecv.resume();
  }
  delay(1000);
}
```



Segue abaixo o código de programação para acionamento remoto dos leds vermelho e verde.

```
#include <IRremote.h>
int inputPin = 2;
IRrecv irrecv(inputPin);
  int ledVM=3;
  int ledVD=4;
decode_results signals;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
  pinMode(ledVM,OUTPUT);
  pinMode(ledVD,OUTPUT);
}
void loop() {
if (irrecv.decode(&signals)) {
  Serial.println(signals.value,DEC); //→ DEC substituindo o HEX
  irrecv.resume();
}
if (signals.value==3782901887){
  digitalWrite(ledVM,HIGH);
  delay(500);
}
if (signals.value==3782869247){
  digitalWrite(ledVM,LOW);
  delay(500);
}
if (signals.value==3782885567){
  digitalWrite(ledVD,HIGH);
  delay(500);
}
if (signals.value==3782869247){
  digitalWrite(ledVD,LOW);
  delay(500);
}
}
```

UTILIZANDO DOIS CONTROLES DIFERENTES NA MESMA PROGRAMAÇÃO

No exemplo a seguir poderemos optar por um dos dois controles vistos anteriormente no mesmo código de programa.

Vamos exemplificar apenas com os leds vermelho e verde, mas poderão ser acrescentados quantos leds ou quaisquer outros dispositivos na saída que se desejar.

Da mesma forma, poderão ser acrescentados mais controles, de diferentes modelos ou marcas.

Vamos então definir os códigos HEXA para os leds vermelho e verde.

Usando as teclas 1 e 2 (acender) e 0 (apagar) temos:

Controle 1 (Car)	Controle 2 (NET)
Tecla 1 (VM) = 0xFF30CF Tecla 2 (VD) = 0xFF18E7 Tecla 0 (OFF) = 0xFF6897	Tecla 1 (VM) = 0xE17A807F Tecla 2 (VD) = 0xE17A40BF Tecla 0 (OFF) = 0xE17A00FF

Veja como fica a programação:

```
#include <IRremote.h>
int inputPin = 2;
IRrecv irrecv(inputPin);
  int ledVM=3;
  int ledVD=4;
decode_results signals;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
  pinMode(ledVM,OUTPUT);
  pinMode(ledVD,OUTPUT);
}
void loop() {
if (irrecv.decode(&signals)) {
  Serial.println(signals.value,HEX);
  irrecv.resume();
}
if ((signals.value==0xFF30CF)||signals.value==0xE17A807F){ //compara qual dos dois
controles está sendo usado, para acionar o led
  digitalWrite(ledVM,HIGH);
  delay(500);
}
if ((signals.value==0xFF6897)||signals.value==0xE17A00FF){ //compara qual dos dois
controles está sendo usado, para apagar o led

  digitalWrite(ledVM,LOW);
  delay(500);
}
if ((signals.value==0xFF18E7)||signals.value==0xE17A40BF){
  digitalWrite(ledVD,HIGH);
  delay(500);
}
if ((signals.value==0xFF6897)||signals.value==0xE17A00FF){
  digitalWrite(ledVD,LOW);
  delay(500);
}
}
```

OBS: A versatilidade fica por conta de que é possível acender o led ou os dois leds com um controle e apagar com o outro.