**Semana Carranca**

**Arduíno Prático**

Equipe de Robótica Carranca

[carrancanear@gmail.com](mailto:carrancanear@gmail.com)

facebook.com/carranca22

**1° Dia**

Entendendo as portas do arduino

1. Sinal Digital

0 ou 1, HIGH ou LOW. O sinal digital só possui dois estados lógicos, ou eu mando energia ou eu não mando energia. Pode-se também receber um sinal digital, os botões só enviam sinais digitais. Outros equipamentos como sensor de presença também só trabalham com sinais digitais.



1. Sinal Analógico

Diferente do sinal digital, o sinal analógico pode variar. Quando estamos **RECEBENDO** o sinal é reconhecido pelo arduino entre :

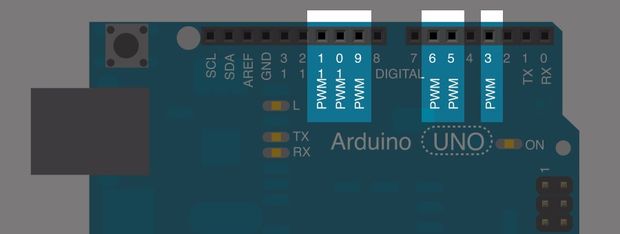
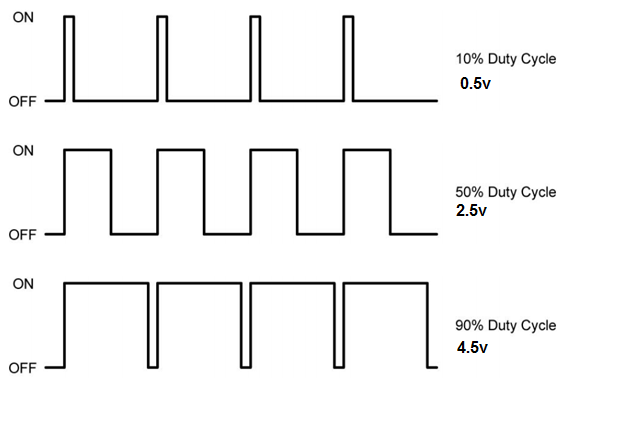
* 1. , quando estamos enviado, entre 0-255.
  2. Portas Analógicas de Entrada

Essas portas só recebem sinal ( 0- 1023).

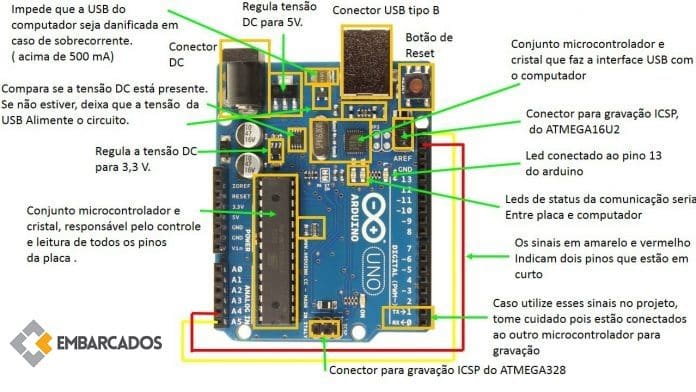


* 1. Portas PWM

As portas PWM (Pulse Width Modulation) “Modulação por largura de pulso” Essas portas são capazes de simular uma tensão menor que a do arduino, modificando muito rapidamente entre Ligado e Desligado.



1. Todos os componentes do Arduino



PROJETOS DE ENTRADAS

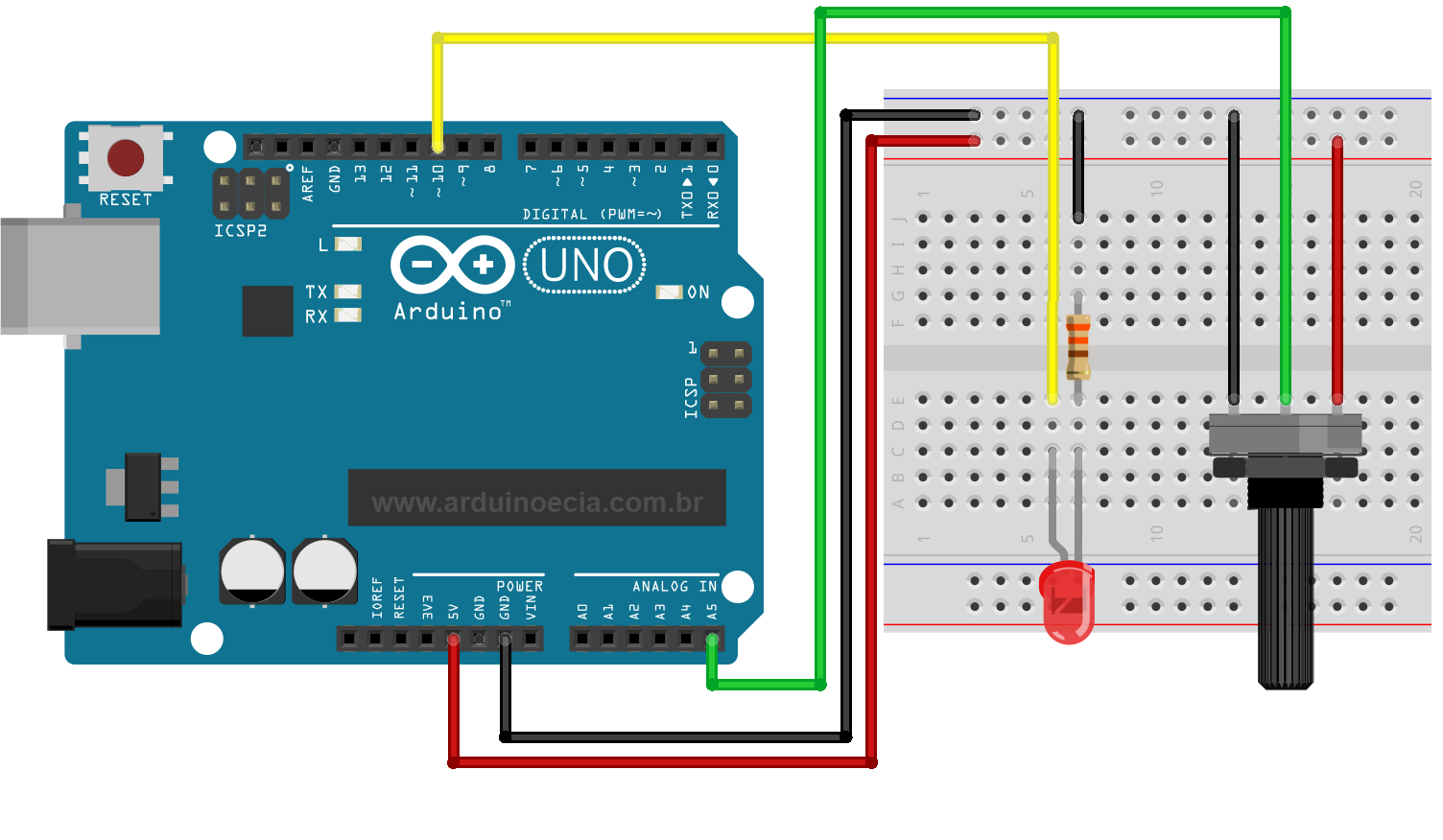
E1) Potenciômetro

Vamos usar o seguinte material....

**1 potenciometro de 10 K**

**1 led**

**1 resistor de 220 ohms**



*// Projeto : Controle de luminosidade de led com potenciometro*

*// Autor : Arduino e Cia*

**int** pinoled = 10; *//Pino ligado ao anodo do led*

**int** pinopot = 5; *//Pino ligado ao pino central do potenciometro*

**int** valorpot = 0; *// valor lido do potenciometro, entre 0 e 1023*

**float** luminosidade = 0; *//Valor de luminosidade do led*

**void** setup()

{

Serial.begin(9600); *//Inicializa a serial*

pinMode(pinoled, OUTPUT); *//Define o pino do led como saída*

pinMode(pinopot, INPUT); *//Define o pino do pot como entrada*

}

**void** loop()

{

*// Le o valor - analogico - do pino do potenciometro*

valorpot = analogRead(pinopot);

*//Converte e atribui para a variavel "luminosidade" o*

*// valor lido do potenciometro*

luminosidade = valorpot/4;

Serial.print("Valor lido do potenciometro : ");

Serial.print(valorpot);

Serial.print(" = Luminosidade : ");

Serial.println(luminosidade);

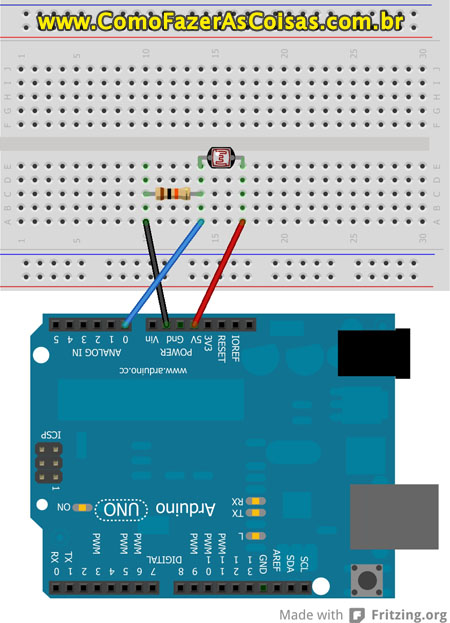
*//Envia sinal analogico para a saída do led, com luminosidade variavel*

analogWrite(pinoled, luminosidade);

}

E2) LDR sensor de luz

* um [Arduino](http://www.comofazerascoisas.com.br/introducao-ao-arduino.html);
* uma protoboard;
* 3 fios;
* um [resistor](http://www.comofazerascoisas.com.br/resistor-o-que-e-e-para-que-serve-introducao-aplicacao.html) de 10K ohms;
* um sensor de luz LDR de 5 ou 10 mm.

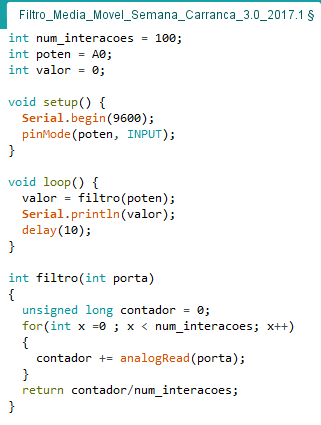


|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | /\*  Projeto Arduino sensor de luz.  Por Jota  ----------------------------------------  --=<| www.ComoFazerAsCoisas.com.br |>=--  ----------------------------------------  \*/    //Pino analógico em que o sensor está conectado.  int sensor = 0;    //variável usada para ler o valor do sensor em tempo real.  int valorSensor = 0;    //Método setup, executado uma vez ao ligar o Arduino.  void setup(){    //Ativando o serial monitor que exibirá os    //valores lidos no sensor.    Serial.begin(9600);  }    //Método loop, executado enquanto o Arduino estiver ligado.  void loop(){      //Lendo o valor do sensor.    int valorSensor = analogRead(sensor);      //Exibindo o valor do sensor no serial monitor.    Serial.println(valorSensor);      delay(500);  } |

Projetos de processo

P1) Filtro de média móvel

O filtro de média móvel é a média aritmética periódica de um parâmetro. É utilizada principalmente para eliminar ruídos indesejáveis e para estabilizar um sinal.



P2) Interrupção

A ideia é que é possível interromper a programação independentemente de onde ela esteja , por um atuador externo, um botão por exemplo.

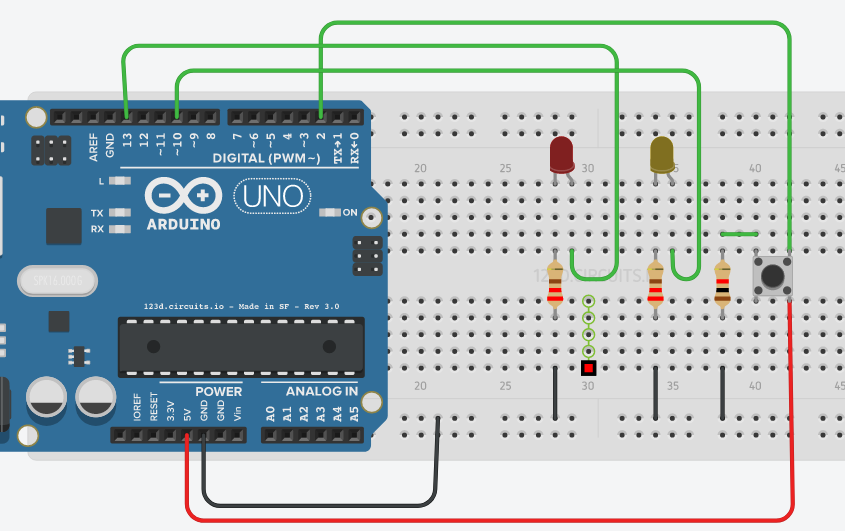
Vamos usar:

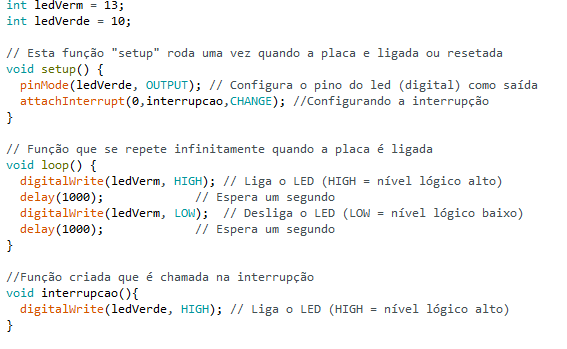
1– Botão

2– LED

2- Resistor 220 Ohms

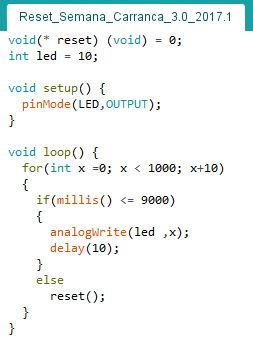
1 -Resistor 1k Ohms





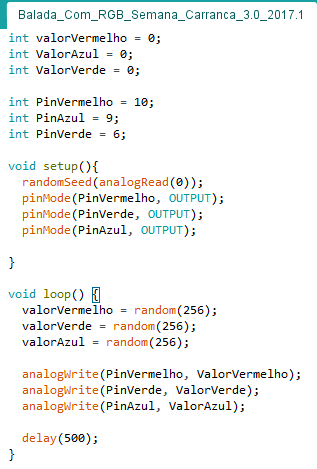
P3) Reset pela programação

A ideia aqui é que é possível resetar a programação utilizando da própria programação sem a necessidade de um agente externo.



P4) Randomizar

O arduino permite utilizar randomização, sendo muito útil em sorteios.



**2° Dia**

# Pulos PWM

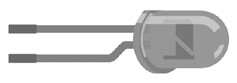


## Projeto 7 – Lâmpada pulsante

Você agora tentará um método mais avançado de controle de seus LEDs. Até aqui você simplesmente os acendeu e apagou. Não seria interessante ajustar o brilho de um LED? Será que isso pode ser feito com o Arduino? Certamente.

Hora de retornar aos fundamentos.

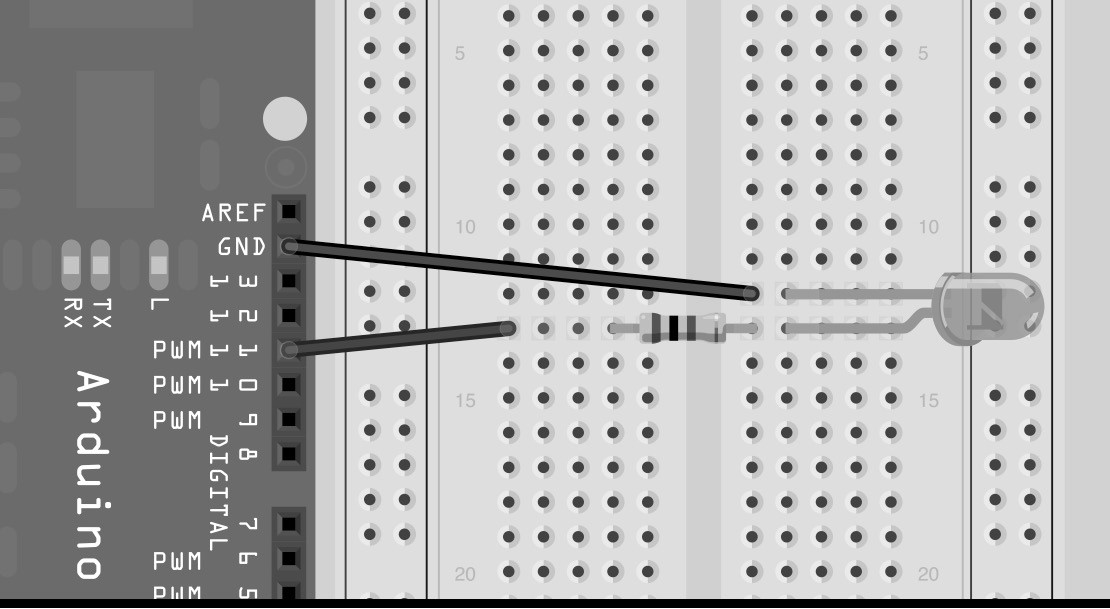
### 1.1.2 Componentes necessários

LED verde difuso de 5 mm 

Resistores limitadores de corrente 

### 1.1.3. Conectando os componentes

O circuito para este projeto apresenta simplesmente um LED verde conectando, por meio de um resistor limitador de corrente, o terra e o pino digital 11 (Figura 3.3).



### 1.1.4. Digite o código

Abra o IDE de seu Arduino e digite o código da listagem 3.3. **Listagem 3.3 – Código para o projeto 7**

int ledPin = 11;

float sinVal;

int ledVal;

void setup() {

pinMode(ledPin, OUTPUT);

}

void loop() {

for (int x=0;x<180; x++) {

// converte graus para radianos e, então, obtém o valor do seno

sinVal = (sin(x\*(3.1416/180)));

ledVal = int(sinVal\*255);

analogWrite(ledPin, ledVal);

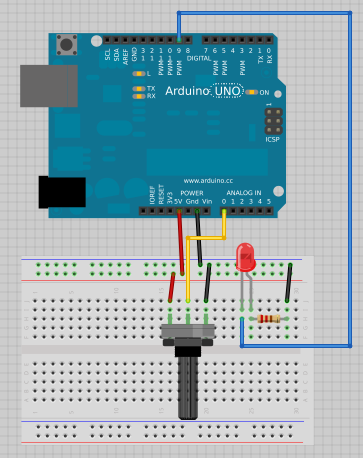
delay(25);

}

}

**Projeto 7– Lampada pulsante com potenciômetro**

**Esquema:**



**Programação:**

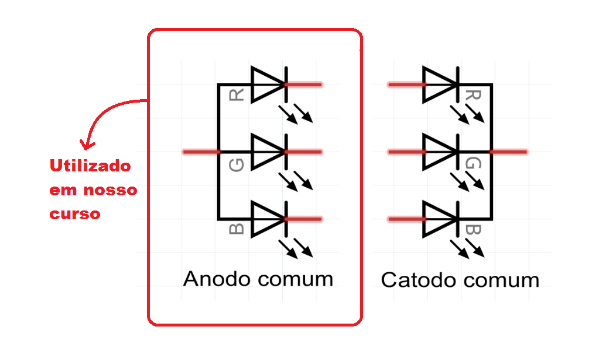
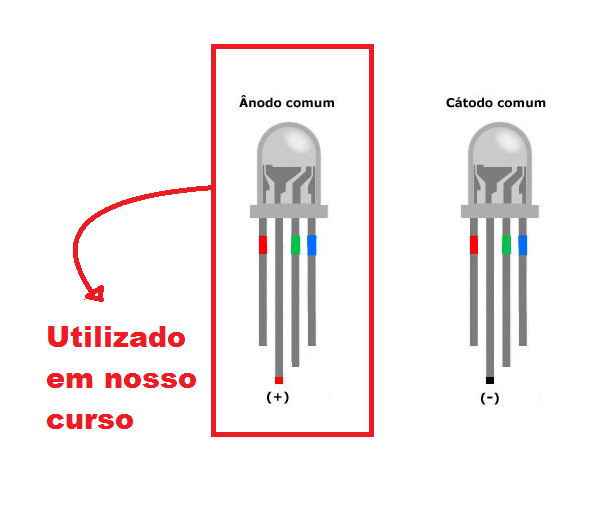
int Ledpin=9;  
int analogpin=A0;  
int value=0;  
int x;

void setup()  
{  
 Serial.begin(9600);   
 pinMode(Ledpin, OUTPUT);  
}

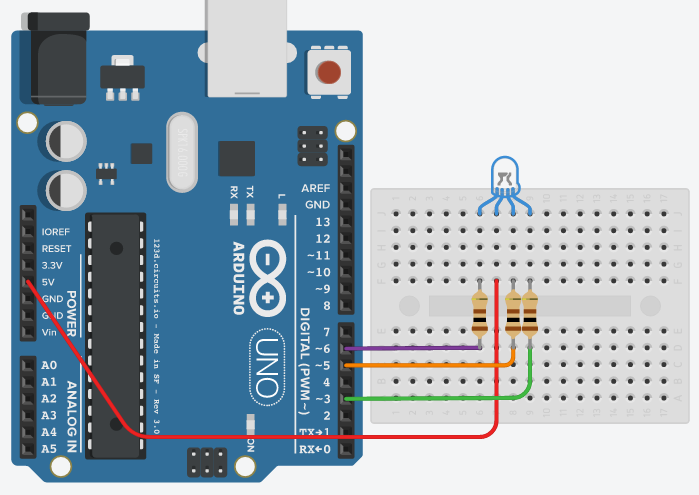
void loop()  
{  
 value=analogRead(analogpin);  
 x=map(value,0,1023,0,255);  
 analogWrite(Ledpin,x);  
 Serial.println(x);  
 delay(100);  
}

# Projeto 8 – O senhor das cores ,LED RGB

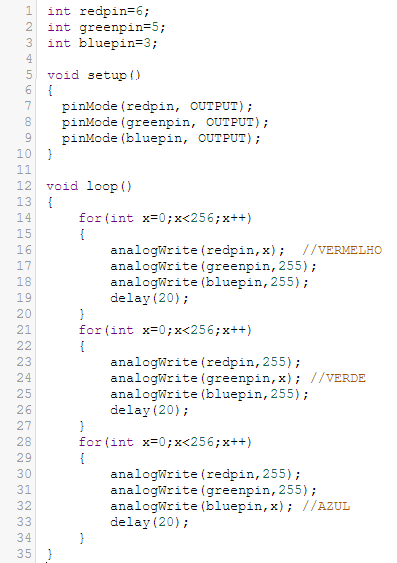
**RGB** é a abreviatura do sistema de cores aditivas formado por Vermelho (**R**ed), Verde (**G**reen) e Azul (**B**lue). O propósito principal do sistema RGB é a reprodução de cores em dispositivos eletrônicos como monitores de TV e computador, retroprojetores, scanners e câmeras digitais, assim como na fotografia tradicional. Geralmente, a partição é de 8 bits para cada uma das cores (vermelho, verde e azul), dando um alcance de 256 possíveis valores, ou intensidades, para cada tom. Com este sistema, mais de 16 milhões (**16.777.216** ou **256³**) diferentes combinações de tons, saturação e brilho podem ser especificadas, mesmo que não sejam facilmente distinguidos.Os leds RGB possuem 4 terminais, um corresponde a cor vermelha, cor azul, cor verde e um cátodo ou anodo comum.



ESQUEMA:

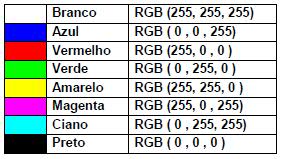


PROGRAMAÇÃO BÁSICA



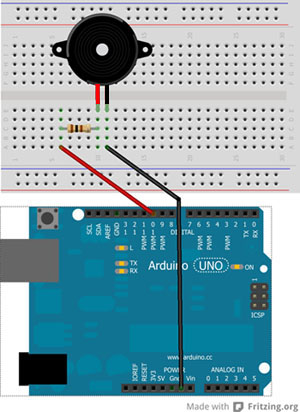
EXPERIMENTE MISTURAR AS CORES SÃO APENAS 16 MILHÕES DE OPÇÕES!!

Algumas....



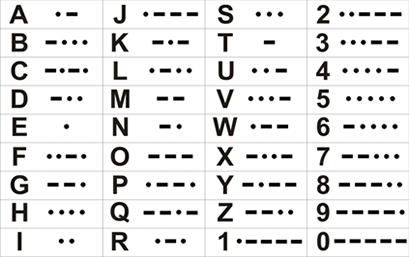
# PROJETO 9 – BUZZER SAIDA DE SOM

**Esquema**

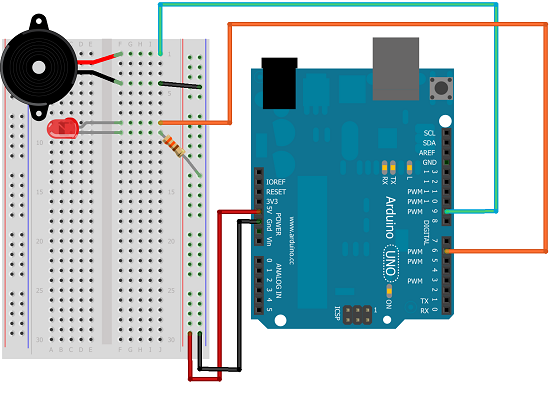


|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | const int buzzer = 10;  void setup() {    //Definindo o pino buzzer como de saída.    pinMode(buzzer,OUTPUT);  }    //Método loop, executado enquanto o Arduino estiver ligado.  void loop() {    //Ligando o buzzer com uma frequencia de 1500 hz.    tone(buzzer,1500);    delay(500);      //Desligando o buzzer.    noTone(buzzer);    delay(500);  } |

**PROJETO 4 – CODIGO MORSE**



**Esquema**



**Programação**

int buzzerPin = 9;  
int ledPin = 7;  
int tonefreq = 523;  
int dotlength = 100; //duração do delay  
int dashlength = dotlength \* 3;  
int inter = dotlength;   
int lgap = dotlength \* 2;  
int wgap = dotlength \* 4;  
  
void **setup**(){  
  pinMode(buzzerPin, OUTPUT);  
  pinMode(ledPin, OUTPUT);  
  **Serial**.begin(9600);  
}  
void **loop**(){  
  char thischar;  
  if (**Serial**.available())  
  {  
    thischar = **Serial**.read();  
    if (thischar>='a' && thischar<='z')  
    {  
      thischar = thischar -32;   
    }  
    soundLetter(thischar);  
    delay(lgap);  
  }  
}  
void dot(){  
  tone(buzzerPin, tonefreq);  
  digitalWrite(ledPin, HIGH);  
  delay(dotlength);  
  noTone(buzzerPin);  
  digitalWrite(ledPin, LOW);  
  delay(inter);  
}  
void dash(){  
  tone(buzzerPin, tonefreq);  
  digitalWrite(ledPin, HIGH);  
  delay(dashlength);  
  noTone(buzzerPin);  
  digitalWrite(ledPin, LOW);  
  delay(inter);  
}  
void soundLetter(char letter){  
  switch(letter)  
  {  
  case 'O':  
    dash();  
    dash();  
    dash();  
    return;   
  case 'S':  
    dot();  
    dot();  
    dot();  
    return;  
  case ' ':  
    delay(wgap);  
    return;   
  }  
}