

Semáforo e Buzzer

Neste projeto será construído um protótipo de semáforo com um diferencial, quando o LED verde que representa o sinal de pedestres for ligado, será ativado simultaneamente o sensor de bip Buzzer, com o propósito de avisar aqueles que não estão atentos ao sinal ou que por alguma necessidade não consegue ver o mesmo.

Componentes e equipamentos

Descrição
Arduino Uno
02 LEDs vermelhos
02 LEDs verdes
01 LED amarelo
06 Resistores 1k
07 Jumpers (fios de conexão)
Buzzer
Protoboard

Hardware:

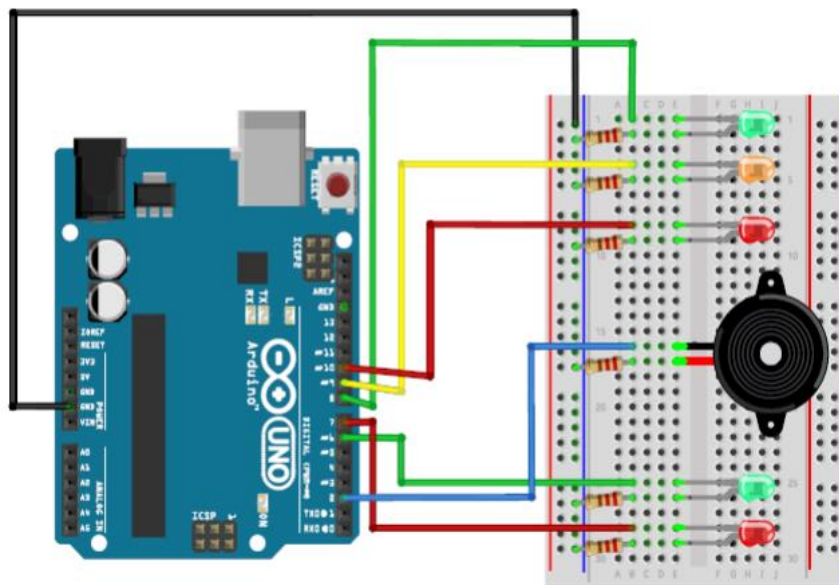


Figura 1 Disposição de componentes na Protoboard

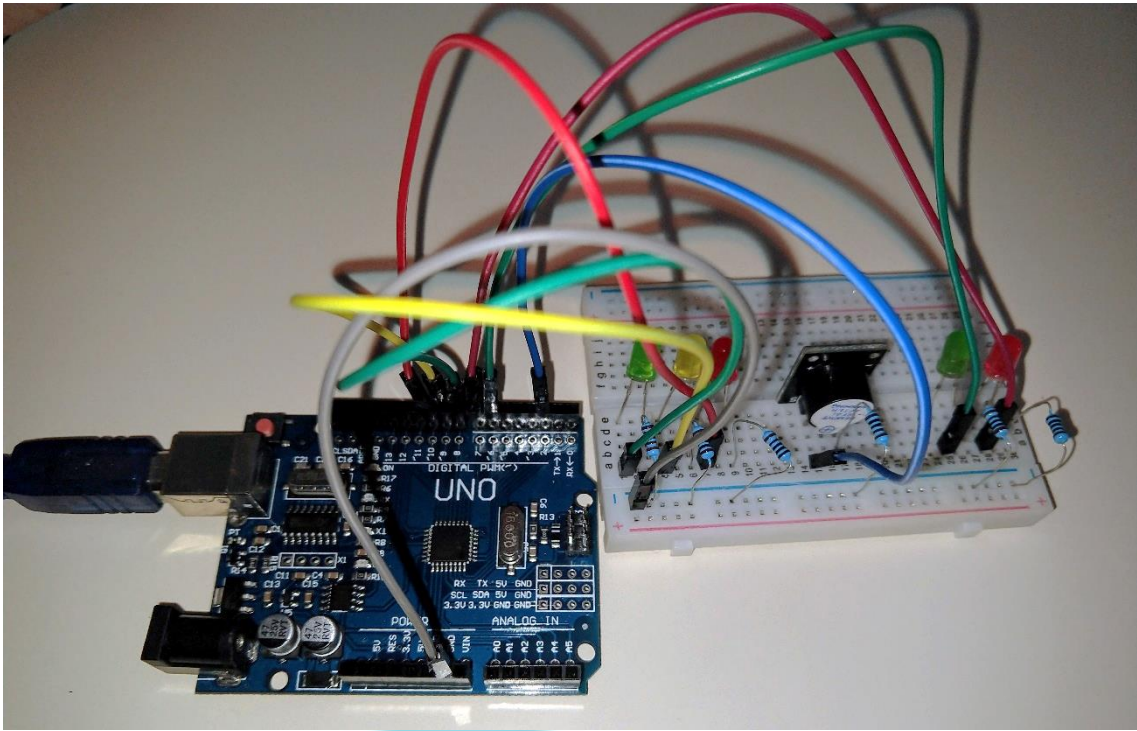


Figura 2

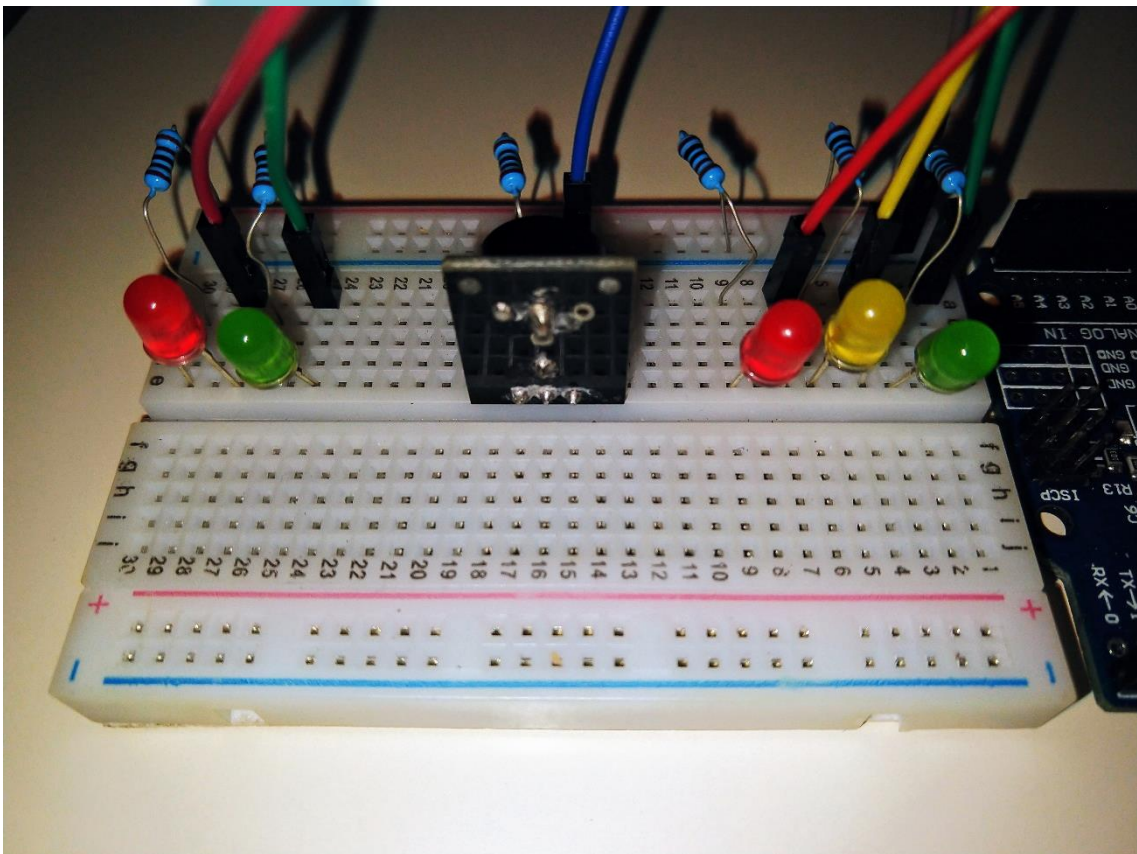


Figura 3

Software:

No ambiente de desenvolvimento do Arduino utiliza-se um compilador, que tem por finalidade converter o código escrito em C para o código de máquina que será executado no microcontrolador.

Entendendo o código...

Passo 1: Inicialmente é dado um nome a um pino de saída digital onde conectaremos um LED, o nome facilitará a identificação de qual pino estamos utilizando em qual LED.

```
Semaforo_Buzz
```

```
int greenPin1 = 8;  
int yellowPin = 9;  
int redPin1 = 10;  
int greenPin2 = 6;  
int redPin2 = 7;  
int buzzPin = 2;
```

Passo 2: A função **setup** tem como finalidade inicializar e declarar os valores e ações iniciais. A finalidade da **pinMode** é permitir que a definição do comportamento de um pino, como entrada ou saída e a **digitalWrite** escreve um valor HIGH (alto) ou LOW (baixo) em um pino digital que tenha sido configurado anteriormente como OUTPUT (saída).

```
void setup() {  
  
  pinMode(greenPin1, OUTPUT); //define o pino 8 como saída  
  pinMode(yellowPin, OUTPUT); //define o pino 9 como saída  
  pinMode(redPin1, OUTPUT); //define o pino 10 como saída  
  pinMode(greenPin2, OUTPUT); //define o pino 6 como saída  
  pinMode(redPin2, OUTPUT); //define o pino 7 como saída  
  pinMode(buzzPin, OUTPUT); //define o buzz  
  
  digitalWrite(greenPin1, HIGH); //liga o led verde  
  
}
```

Passo 3: A função loop como o nome já nos indica permite que seu programa funcione dinamicamente, já que ela se repete continuamente.

```
void loop() {
  //liga led vermelho2
  digitalWrite(redPin2,HIGH);

  //controle do led verde1
  digitalWrite(greenPin1,HIGH); //acende o led
  delay(4000); //espera 4 segundos
  digitalWrite(greenPin1,LOW); //apaga o led

  //controle do led amarelo1
  digitalWrite(yellowPin,HIGH); //acende o led
  delay(2000); //espera 2 segundos
  digitalWrite(yellowPin,LOW); //apaga o led //Controle do led vermelhol

  //desliga o vermelho2
  digitalWrite(redPin2,LOW); //apaga o led

  //liga o verde2
  digitalWrite(greenPin2,HIGH); //acende o led
  //liga o vermelhol
  digitalWrite(redPin1,HIGH); //acende o led
```

Ainda na função loop temos as funções Tone e a no Tone que tem como propósito respectivamente gerar tons de frequências pré-estabelecidas e interromper esses mesmos tons.

```
//ligar buzzer
tone (buzzPin, 440);
delay (1800);
noTone (buzzPin);
delay (200);
tone (buzzPin, 440);
delay (450);
delay (50);
tone (buzzPin, 440);
delay (450);
noTone (buzzPin);
delay (50);
tone (buzzPin, 440);
delay (450);
noTone (buzzPin);
noTone (buzzPin);
delay (50);

//delay(4000); //espera 4 segundos
digitalWrite (redPin1, LOW); //apaga o led
//desligar led verde2
digitalWrite (greenPin2, LOW);
```

```
}
```