

Acendendo o Led com Arduino.

Nesse tutorial aprenderemos a ligar e desligar um Led com o Arduino.

Para isso utilizaremos os seguintes componentes e ferramentas:

- Arduino



Figura 1 – Arduino UNO.

Placa de prototipação utilizada nos nossos projetos eletrônicos desse curso.

- Protoboard



Figura 2

Nessa placa faremos a montagem de nossos exemplos de circuitos elétricos durante o curso.

- Jumper

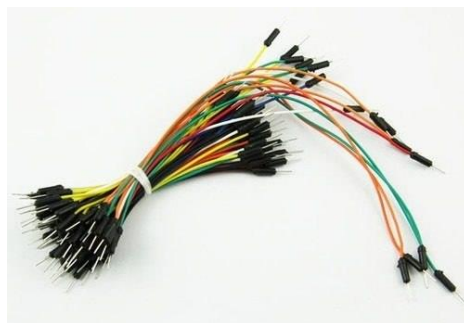
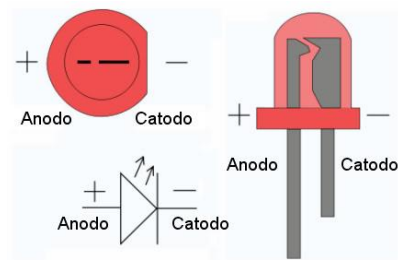
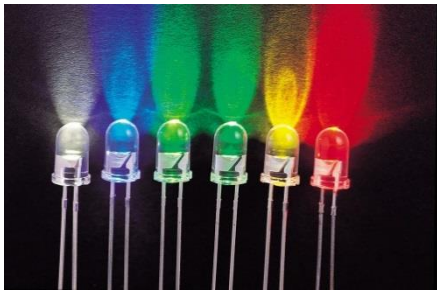


Figura 3

São os conectores que utilizaremos para ligar os componentes elétricos que vamos utilizar.

- Led



Figuras 4 – Led's

É um componente eletrônico emissor de luz, utilizado para ser a “lâmpada” do circuito.

- Resistor



Figura 5 – Resistor

Componente utilizado para limitar a corrente elétrica que atravessará o Led.

Agora que já listamos os materiais que utilizaremos, vamos fazer um desenho de como deve ficar nosso circuito elétrico e partir pra prática.

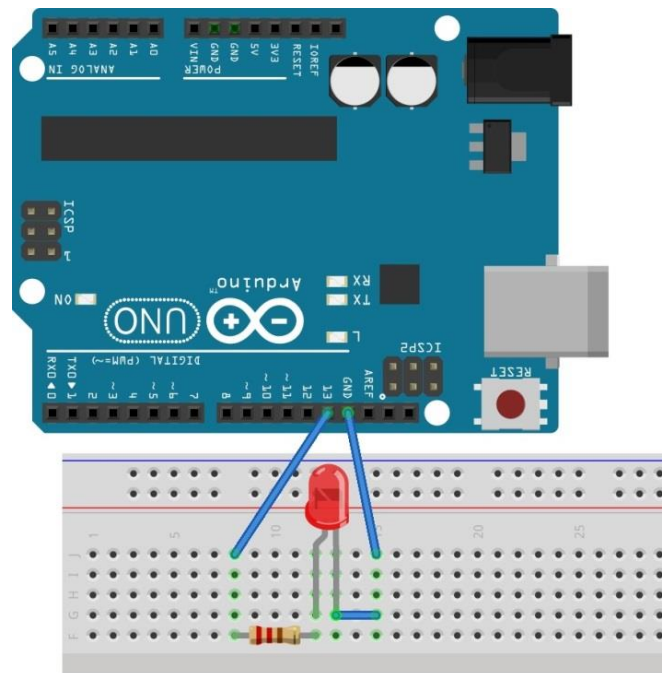


Figura 6 – Desenho do Circuito

Como podem ver, temos um resistor em nosso circuito, então vamos calcular seu valor através dos conhecimentos já obtidos no tutorial de eletrônica.

Para calcular a resistência associada em serie com o Led, faremos o seguinte calculo:

$$R1 = \frac{VR1}{I}$$

Onde, para achar a queda e tensão no resistor (VR1) basta subtrair a tensão de saída da porta digital do Arduino, que é de 5V, com a queda de tensão do Led que é de 2V. A corrente utilizada pelo Led também pode ser encontrada no Datasheet, é de 20mA.

$$R1 = \frac{Vreg - Vled}{Iled} \qquad R1 = \frac{5 - 2}{20mA}$$

$$R1 = \frac{3}{0,02} \qquad R1 = 150\Omega$$

Como não encontramos resistores com o valor calculado 150Ω, no mercado, basta utilizar um próximo. No caso usaremos o de 220Ω.



O que mudará em nosso projeto se utilizar o resistor de 220Ω?

Para obter uma iluminação satisfatória deveríamos utilizar o resistor de 150Ω, mas como iremos utilizar o resistor de 220Ω o nosso Led não irá brilhar com a mesma intensidade, pois como a resistência é maior a corrente será menor. Mas essa diferença de resistência tem efeito quase imperceptível ao nosso olhar.

Agora que calculamos nossa resistência, vamos montar nosso circuito.

Conecte o Led em duas colunas, se quiser pode dobrar as “pernas” ou cortar para diminuir seu tamanho.

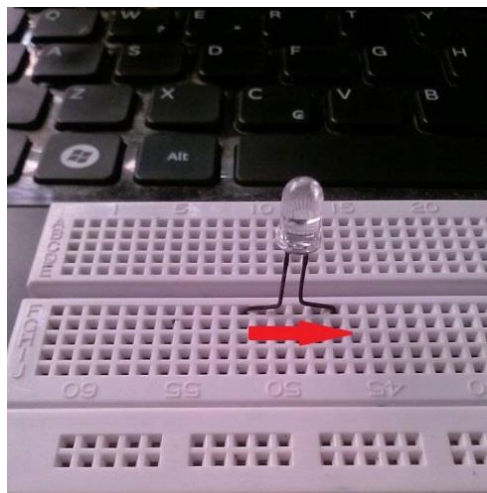


Figura 7 – Ligando Led.

Perceba que o sentido da corrente no Led é indicado pela seta, entrando pelo terminal Anodo (redondo) e saindo pelo terminal Catodo (Achatado). Como estamos fazendo uma associação em série, podemos ligar o resistor antes ou depois do Led, mas vamos seguir o desenho inicial e conecta-lo antes, no terminal Anodo do Led.



Figura 8 – Conectando Resistor.

Veja que o terminal do resistor esta na mesma coluna do terminal do Led, então eles estão conectados em série.



O que acontece se ligarmos o Led de maneira contraria em nosso circuito?

De maneira geral, se você estiver alimentando seu circuito com a tensão de saída do Arduino, que é de 5 Volts, dificilmente isso causará danos ao seu Led, pois a tensão típica reversa suportada pelo componente é de 5V. Mas, tome muito cuidado quando usar fontes externas, leia atentamente o Datasheet do componente em questão e sempre avalie **Tensão Máxima**, assim como **Corrente Máxima**, direta e/ou reversa, suportada e o modo de uso, para ligá-lo corretamente em seu circuito.

Por fim, vamos utilizar dois jumper para conectar o pino 13 do Arduino no resistor e, o outro, para conectar a outra “perna” do Led (Catodo) no GND do Arduino. Nosso circuito deve ficar da seguinte maneira.

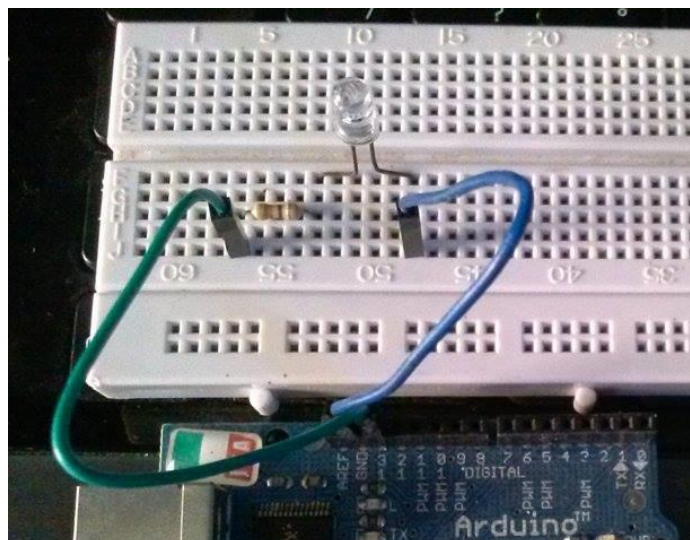


Figura 9 – Circuito Completo.

Agora, para terminar nosso projeto, devemos programar o Arduino e fazer os testes. 😊

Abra a IDE do Arduino e clique em **Arquivo** → **01.Basics** → **Blink** que é o arquivo exemplo de acender o Led.

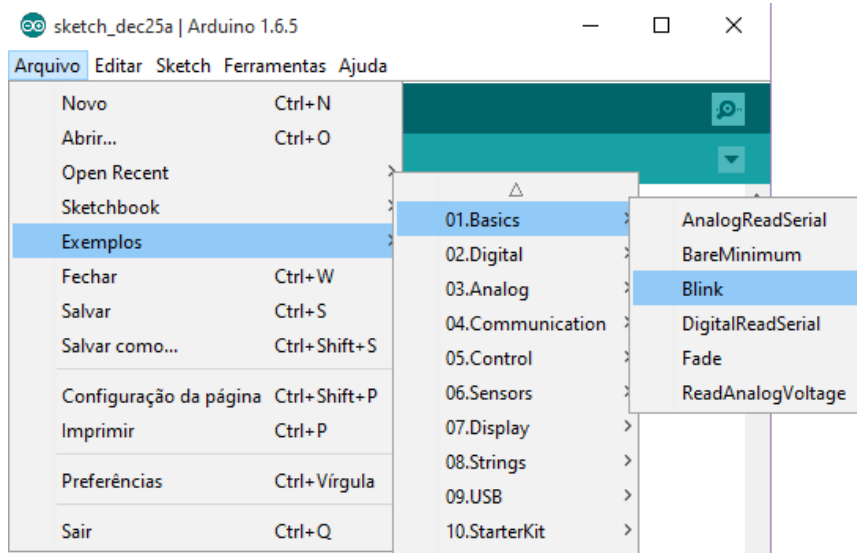


Figura 10 – IDE Arduino.

Vamos entender o código apresentado.

1. Todo o código, ou conjunto de palavras escrito entre os símbolos `/* */` é visto pelo compilador do Arduino como “**Comentário de Código**”, ou seja, não é levado em consideração na geração do código binário **.hex**. Então, a parte inicial apresentada é um comentário feito pelo autor do código, explicando-o.
2. Após isso o autor declara a função **void setup()**, onde declara todas as condições iniciais do código.

// a função de configuração é executada uma vez quando você pressionar reset ou quando liga o Arduino.

```
void setup() {
  // inicializa o pino digital 13 como Saída.
  pinMode(13, OUTPUT);
}
```

3. Agora declaramos a função **void loop()**, onde estarão presentes os comandos para ligar e desligar o Led automaticamente durante 1 segundo.

// a função loop é executada indefinidamente.

```
void loop() {
  digitalWrite(13, HIGH); // ligar o LED em nível de tensão alto (HIGH)
  delay(1000);           // esperar por 1 segundo
  digitalWrite(13, LOW); // ligar o LED em nível de tensão baixo (LOW)
  delay(1000);           // esperar por 1 segundo
}
```

Com esse código faremos o Led piscar indefinidamente, acendendo e apagando durante 1 segundo.



Robô na Escola - UFPA - 2015

Autor: Alisson Ricardo da Silva Souza

Bibliografias

Fotos – Própria Autoria.