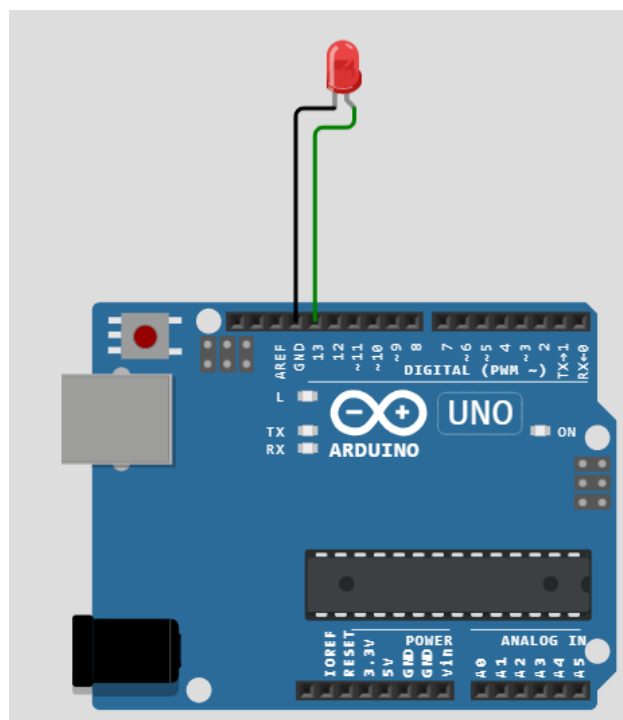


## Relatório de Sistemas de Controle – Acionamento de LED.

Grupo: Ana Carolina Cardoso & Marco Bockoski & Mateus Cardoso

### EXPLICAÇÃO

Foi desenvolvido em sala um circuito eletrônico no qual o objetivo é acionar um LED, utiliza-se um resistor para limitação de corrente que flui pelo LED, prevenindo a sobrecarga do LED. Um microcontrolador Arduino foi disposto para realizar as operações de acionamentos do LED, portanto, eles são conectados entre si, juntamente com o resistor mencionado acima.

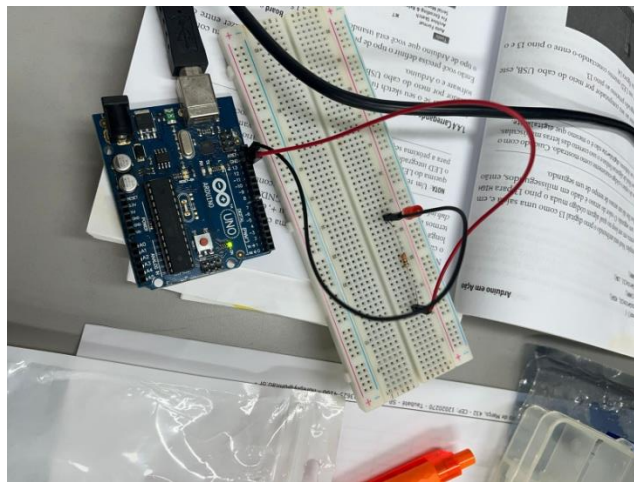


### MATERIAIS UTILIZADOS

- 2 fios condutores
- 1 cabo USB
- 1 Microcontrolador Arduino Uno
- 1 resistor de  $330\Omega$
- 1 LED vermelho
- 1 protoboard

## MONTAGEM

Utiliza-se a entrada digital 13 do Arduino e o terra (GND), a entrada digital 13 recebe um fio condutor vermelho que tem sua outra extremidade colocada na Protoboard. A Protoboard é utilizada para gerar ligações entre componentes do circuito sem necessidade de soldagem, assim, conecta-se em série o resistor e o LED. Ênfase na conexão do LED que tem seu Anodo ligado ao resistor e o Catodo (terminal com o chanfro) tem ligação com um fio condutor preto, este por fim é conectado ao terminal terra do Arduino (GND).



## CÓDIGO

```
int pinoLED = 13;

void setup() {

  Serial.begin(9600);
  pinMode(pinoLED, OUTPUT);

}

void loop() {
  digitalWrite(pinoLED, HIGH);
  delay(100);
  digitalWrite(pinoLED, LOW);
  delay(100);
}
```

O Script enviado para o Arduino é separável em 3 partes:

- Declaração de Variáveis
- Ligação com o Arduino e Definição de entrada/saída
- Acionamento do LED

A primeira etapa se trata de criar uma variável correspondente ao terminal digital 13 do Arduino no qual o LED está conectado. Essa variável será processada no decorrer do código.

A segunda parte apresenta a função Setup, que corresponde aos processos anteriores à devida execução do código, geralmente nesse espaço declara-se especificações de sensores, declarações de LCDs. Nesse caso, foi determinado como o terminal digital 13 seria utilizado, que no caso, funciona como saída. Isso é definido pela função pinMode().

Por fim, a função loop() corresponde aos comandos que serão executados por tempo indefinido no Arduino, no corpo desta função, há apenas 2 tipos de comandos:

- digitalWrite()
- delay()

Sendo a primeira função responsável por modificar o estado do LED, assim, o primeiro argumento da função corresponde ao terminal em que o LED está conectado e o segundo argumento da função corresponde aos estados digitais HIGH e LOW, que seriam respectivamente, ligado e desligado.

A segunda função corresponde no atraso (em milissegundos) do código. Pois o processamento do script é muito rápido, para as modificações no sistema serem perceptíveis ao olho humano, essas funções são responsáveis por permitir visualizar os comandos serem executados.