



**UNIVERSIDADE FEDERAL DO PARÁ - UFPA**

**INSTITUTO DE TECNOLOGIA – ITEC**

**FACULDADE DE ENGENHARIA DA COMPUTAÇÃO**

**RELATÓRIO DO PROJETO DE SEMÁFORO PARA O  
CRUZAMENTO DE DUAS VIAS**

Bruno Santos de Sousa

Dilton Vilhena Brandão

Ivan Pedro Lima de Sousa

Izidio Sousa de Carvalho

Jamile Lima Leite

Marcelo Henrique de Paula dos Santos

**BELÉM - PARÁ**

**2016**

# Sumário

<b>1. Introdução.....</b>	<b>3</b>
<b>2. Esquemático do Circuito Eletrônico.....</b>	<b>4</b>
2.1. Ilustração da Placa Arduino .....	4
2.2. Ilustração do Circuito Eletrônico.....	4
<b>3. Memória de Cálculos .....</b>	<b>5</b>
3.1. Cálculo da resistência em relação aos LEDs.....	5
3.2. Tabela de Cálculo: Tempo do Ciclo do Semáforo .....	6
3.3. Tabela de Cálculo: Tempo dos Pedestres.....	6
<b>4. Descrição do Código Solução.....</b>	<b>7</b>
4.1. Explicação do Código Solução .....	7
4.2. Descrição Narrativa do Código Solução .....	7
<b>5. Conclusão .....</b>	<b>9</b>
<b>6. Anexo I: Código Solução .....</b>	<b>9</b>

## 1. Introdução

Este trabalho visa uma simulação de um semáforo em um cruzamento de duas vias, onde em uma das vias temos uma faixa para pedestres com sinalização. A sinalização para pedestres é ativada através de um interruptor, fechando o sinal das duas vias para a travessia do pedestre. Porém, o interruptor somente funcionará caso ele não tenha sido pressionado nos últimos dez segundos. Algumas medidas de segurança são necessárias, como a permanência do sinal fechado para ambas as vias por dois segundos na transição da sinalização de siga para pare.

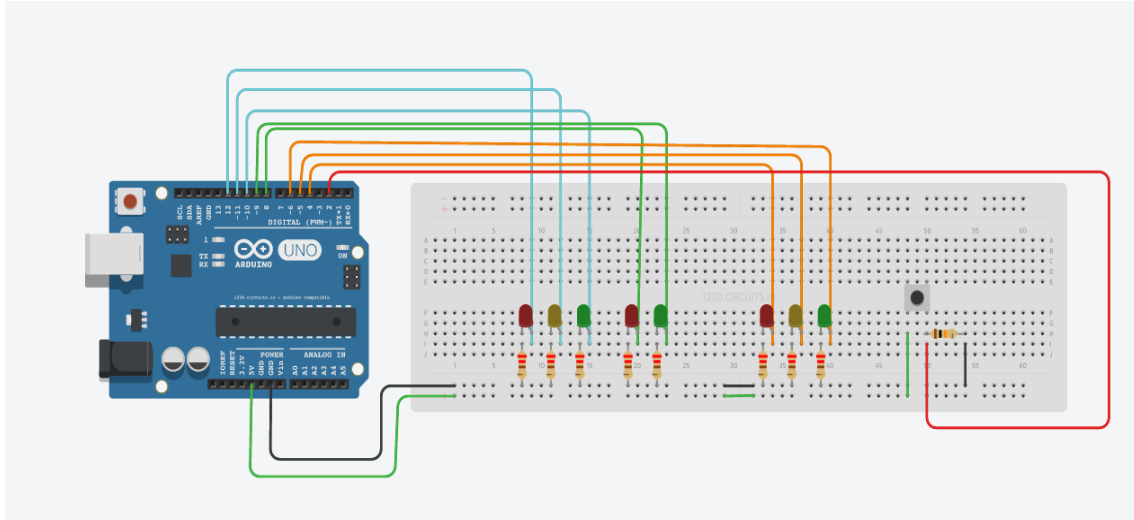
Para isso, foi necessário o uso da plataforma de prototipagem eletrônica, Arduino, especificamente o modelo Arduino Uno. Os componentes utilizados foram uma placa com conexões condutoras (protoboard), cabos jumper, LEDs, resistores de  $200\Omega$ , um resistor de  $10K\Omega$ , e um interruptor (push-button).

As sinalizações em ambas as vias foram feitas através de LEDs coloridos. Os cabos jumper foram usados na conexão entre a placa Arduino e a protoboard. Para controle da corrente elétrica foram utilizados resistores. O push-button foi utilizado como o botão que ativa a sinalização para pedestres.

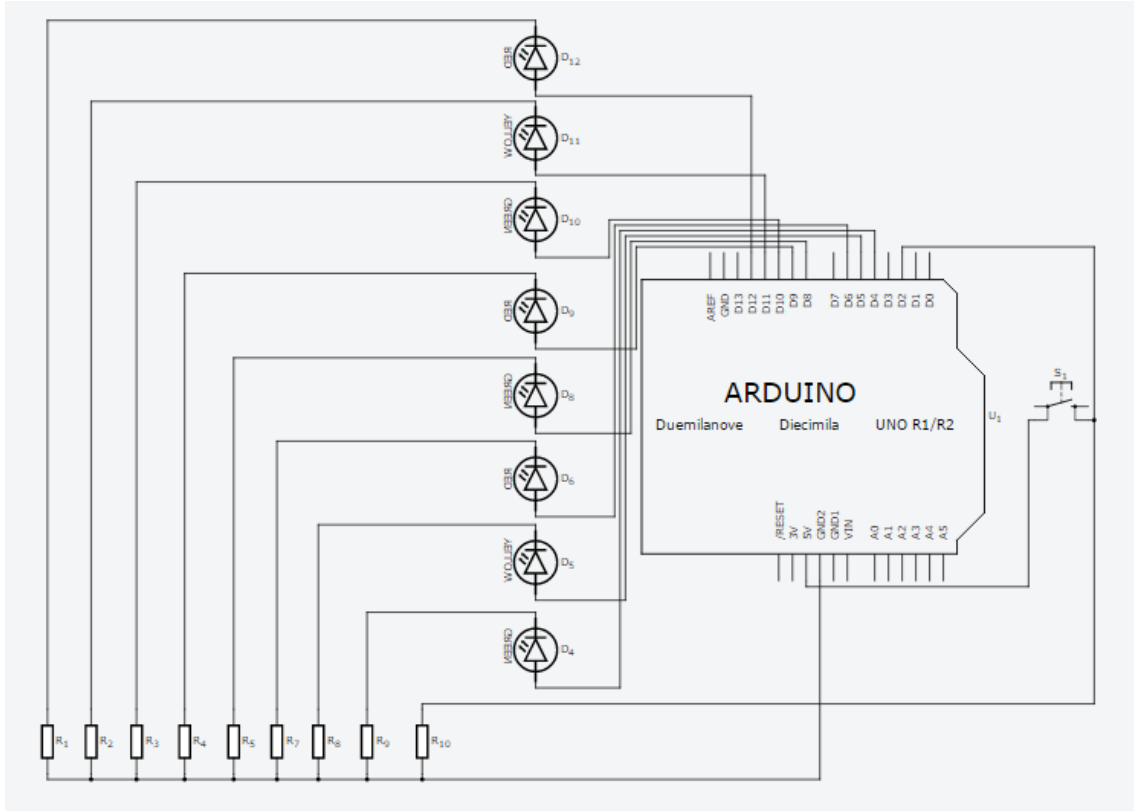
A familiarização com o projeto e as simulações foram feitas através do software Proteus e da plataforma 123D Circuits (Autodesk Circuits). A programação foi feita através da multiplataforma Arduino IDE.

## 2. Esquemático do Circuito Eletrônico

2.1. Ilustração do Arduino com suas componentes utilizadas durante o projeto, obtida a partir de uma simulação na plataforma 123D Circuits (Autodesk 123D).



2.2. Esquemático do circuito elétrico, obtida a partir de uma simulação na plataforma 123D Circuits (Autodesk 123D).



### 3. Memória de Cálculos

$$R = (V_a - V_l)/i$$

$V_a$  [Tensão de alimentação do circuito em Volts]

$V_l$  [Tensão de funcionamento do LED, também em Volts]

$i$  [Corrente de funcionamento do LED em Ampère]

**3.0.1. Observação:** para todos os LEDs, a corrente de funcionamento será a mesma.

**3.0.2. Observação:** uma tensão de 5V é fornecida pelo circuito, e será igual para todos os LEDs.

#### 3.1. Cálculo da resistência em relação aos LEDs

##### 3.1.1. Para o LED vermelho:

- Funciona a uma tensão de 2,0V e uma corrente elétrica de 20,0mA.

Logo, sendo  $i = 20\text{mA}$  e convertendo para Ampère, temos que  $i = 0,02\text{A}$ , então, pela fórmula apresentada anteriormente:

$$R = (5\text{V} - 2\text{V})/0.02\text{A} = 150\Omega$$

Essa será a resistência aproximada que deverá ser ligada em série com o LED vermelho. Nesse caso, será o valor dos resistores R1, R4 e R7.

##### 3.1.2. Para o LED amarelo:

- Funciona a uma tensão de 2,0V e uma corrente elétrica de 20mA.

Logo, sendo a tensão fornecida pelo circuito igual em todos os pontos, note que:

$$R = (5\text{V} - 2\text{V})/0,02\text{A} = 150\Omega$$

Essa será a resistência aproximada a ser ligada em série com o LED amarelo. Nesse caso, será o valor dos resistores R2 e R8.

##### 3.1.3. Para o LED verde:

- Funciona a uma tensão de 2,5V e uma corrente elétrica de 20mA.

Logo, com a tensão fornecida igual em todos os pontos, temos:


$$R = (5\text{V} - 2,5\text{V})/0,02\text{A} = 125\Omega$$


Resistência a ser ligada em série com o LED verde para melhor desempenho na luminosidade de Led. Nesse caso, será o Valor dos resistores R3, R5 e R9.

### 3.2. Tabela de Cálculo: Tempo do Ciclo do Semáforo

Semáforo A	Semáforo B	Tempo (em segundos)	Caso o botão seja acionado	Tempo de processamento (em segundos)	Semáforo A	Semáforo B	Execução da função pedestre
Verde	Vermelho	30		3	Amarelo	Vermelho	
Amarelo	Vermelho	3		3	Amarelo	Vermelho	
Vermelho	Vermelho	2					
Vermelho	Verde	10		3	Vermelho	Amarelo	
Vermelho	Amarelo	3		3	Vermelho	Amarelo	
Vermelho	Vermelho	2					

Legenda:

 O que ocorre quando o botão é pressionado com os semáforos nas configurações especificadas.

 Execução da função pedestre.

**Observação 3.2.1:** Durante todo o ciclo, o semáforo do pedestre ficará no vermelho caso a função pedestre não seja acionada.

### 3.3. Tabela de Cálculo: Tempo dos Pedestres

Tempo (em segundos)	Semáforo A	Semáforo B	Pedestre
2	Vermelho	Vermelho	Vermelho
15	Vermelho	Vermelho	Verde
3	Vermelho	Vermelho	Verde/Vermelho

**Observação 3.3.1:** Depois desse ciclo, o semáforo do pedestre terá o "siga" desligado e se manterá no vermelho até que a função pedestre seja acionada novamente.

## **4. Descrição do Código Solução**

### **4.1. Explicação do Código Solução**

O código começa com a declaração das variáveis dos LEDs e do botão para definir os pinos de cada componente. Além disso, também é declarada uma variável que auxiliará a função millis mais à frente no código. Após isso temos a inicialização através da função setup, os LEDs começam como entradas e o botão como saída, para verificar se está tudo correto, eles acendem por cinco segundos, apagam, e após cinco segundos com eles apagados, o LED vermelho dos pedestres acende, e então a função loop é iniciada.

A função loop inicia fazendo o sinal vermelho da via B, e o verde da via A ficarem acesos. Após trinta segundos a função segue, fazendo o amarelo da via A acender, apagando o verde. Após três segundos, o amarelo de A apaga, acendendo o vermelho. Por dois segundos as duas vias ficam em vermelho, e então o vermelho da via B apaga e o verde acende por dez segundos, após esses dez segundos o verde da via B desliga, acendendo o amarelo por três segundos, e então apaga, acendendo o vermelho da via B, as duas vias ficam dois segundos em vermelho, e então voltamos ao início do loop.

Caso o botão de pedestres seja pressionado, se alguma das vias estiver com sinal verde, ele fica amarelo por três segundos e ativa a função pedestre, que deixa os sinais vermelhos por dois segundos. Se o botão for pressionado e as vias estiverem vermelhas, a função pedestre é ativada. Se o botão for ativado e algum dos sinais estiver amarelo, ele fica mais três segundos em amarelo e ativa a função pedestre. A função pedestre faz com que o sinal dos pedestres fique verde por quinze segundos, então o sinal vermelho dos pedestres acende, ficando os dois acesos por três segundos, e então o sinal verde é desligado. No final previousmillis assume o valor da função millis, impedindo que o botão seja pressionado antes de dez segundos após ele ter sido pressionado.

### **4.2. Descrição Narrativa do Código Solução**

- Todos os LEDs acendem por 5 segundos;
- Todos os LEDs apagam;
- Após 5 segundos o LED vermelho dos pedestres acende;

- Se o botão dos pedestres for pressionado e o intervalo em relação a última vez que foi pressionado for maior que dez segundos, enquanto o sinal estiver verde, o sinal fica amarelo por 3 segundos, e ativa a função pedestre;
- Se o botão dos pedestres for pressionado e o intervalo em relação a última vez que foi pressionado for maior que dez segundos, enquanto o sinal estiver vermelho, ativa a função pedestre;
- Se o botão dos pedestres for pressionado e o intervalo em relação a última vez que foi pressionado for maior que dez segundos, enquanto o sinal estiver amarelo, o sinal fica amarelo por mais 3 segundos, e ativa a função pedestre;
- A função pedestre fecha o sinal das duas vias, e deixa ambas com o sinal vermelho por dois segundos, apaga o sinal vermelho dos pedestres e acende o verde por 15 segundos. Após os 15 segundos o sinal vermelho acende por 3 segundos junto ao sinal verde. Após os 3 segundos o verde apaga, e o vermelho continua acesso.
- O loop inicia com o vermelho da via B e o verde da via A acesos;
- Se houverem passado 30 segundos e o botão não foi pressionado, o LED amarelo da via A acende e o verde apaga;
- Após 3 segundos, e se o botão não foi pressionado, o amarelo apaga e o vermelho acende;
- Após 2 segundos, e se o botão não foi pressionado, o vermelho da via A acende, o vermelho da via B desliga, acendendo o verde;
- Se houverem passado 10 segundos e o botão não foi pressionado, o LED amarelo da via B acende e o verde apaga;
- Após 3 segundos, e se o botão não foi pressionado, o amarelo apaga e o vermelho acende;
- Após 2 segundos, e se o botão não foi pressionado, o vermelho da via B acende, o vermelho da via A desliga.



## 5. Conclusão

Neste projeto, aplicamos alguns conceitos da engenharia, em especial a eletrônica e programação. Isso permitiu uma familiarização com a placa Arduino e com a programação baseada em C, além da interpretação e análise de circuitos elétricos, ou seja, saímos dos termos técnicos da sala de aula para uma prática que incorpora as áreas de conhecimento do curso de Engenharia da Computação.

O objetivo principal era conseguir utilizar comandos básicos do Arduino para resolver uma solução lógica de uma situação real e aplicá-las a uma linguagem de programação. Conseguimos resolver o problema de sinalização das vias, adquirindo conhecimentos sobre as funções principais na programação com Arduino, além de imaginar soluções para outras situações a partir desta. Com a cooperação de todos os membros da equipe chegamos em um resultado satisfatório tanto em conhecimento como em prática. Em suma, o objetivo final foi cumprido.

## 6. Anexo I: Código Solução

```
const int G1 =10;
const int Y1 =11;
const int R1 =12;
const int Gp =8;
const int Rp =9;
const int R2 =6;
const int Y2 =5;
const int G2 =4;
const int buttonPin =2;

unsigned long previousMillis = 0;

int cont = 0;

void setup() {
  pinMode(G1, OUTPUT);
  pinMode(Y1, OUTPUT);
  pinMode(R1, OUTPUT);
  pinMode(Gp, OUTPUT);
  pinMode(Rp, OUTPUT);
  pinMode(R2, OUTPUT);
  pinMode(Y2, OUTPUT);
  pinMode(G2, OUTPUT);
  pinMode(buttonPin, INPUT);

  /*Os leds são ligados para verificar se estão funcionando e depois desligados para todos
  iniciarem assim.*/
  digitalWrite(G1, HIGH);
  digitalWrite(Y1, HIGH);
  digitalWrite(R1, HIGH);
  digitalWrite(Gp, HIGH);
  digitalWrite(Rp, HIGH);
```

```

digitalWrite(R2, HIGH);
digitalWrite(Y2, HIGH);
digitalWrite(G2, HIGH);

delay(5000);

digitalWrite(G1, LOW);
digitalWrite(Y1, LOW);
digitalWrite(R1, LOW);
digitalWrite(Gp, LOW);
digitalWrite(Rp, LOW);
digitalWrite(R2, LOW);
digitalWrite(Y2, LOW);
digitalWrite(G2, LOW);

delay(5000);

/*
 * A led vermelha da via para pedestres já é definida na função 'setup' pois o único
momento em que ela será apagada é na função 'pedestre',
 * e no final dessa mesma função, a led vermelha para os pedestres é ligada novamente.
 */
digitalWrite(Rp, HIGH);
}

void loop() {

/*
 * Para cada combinação de leds apagados e/ou acessos, haverá uma função 'for'
diferente. Dentro do 'for', o contador e o delay farão com
 * que tal função fique num loop por um determinado tempo com os leds que queremos
acesos, e a cada loop, é verificado se o botão é
 * pressionado e se o tempo passado desde a ultima vez q a via para pedestres foi
aberta é maior que 10 segundos. Caso as duas afirmações
 * sejam verdadeiras, a função pedestre é chamada, que é a função que abrirá a via para
os pedestres.
 */

/*
 * A rua 1(A) começará com a via aberta e as outras fechadas por 30 segundos, então,
seu led verde e os leds vermelhos das outras vias
 * são/continuarão ligados.
 * Se as condições para a via para pedestres abrir forem verdadeiras, o semáforo da
rua 1 primeiro ficará amarelo por 3
 * segundos e depois a função pedestre será executada. Depois de sua execução, o
contador receberá o valor de 3002, não estando mais na
 * condição da função 'for' e assim saindo dela(isto acontecerá em todas as funções
'for' quando a função pedestre for executada, porém
 * com valores diferentes).
 */
for(cont=1;cont<=3000;cont++){
    digitalWrite(R1,LOW);
    digitalWrite(G1,HIGH);
    digitalWrite(R2,HIGH);
    delay(10);
    if(digitalRead(buttonPin)==HIGH && millis() - previousMillis > 10000){
        for(cont=1;cont<=300;cont++){
            digitalWrite(G1,LOW);
            digitalWrite(Y1,HIGH);
            delay(10);
        }
        pedestre();
        cont = 3002;
    }
}

/*Caso o botão tenha sido pressionado durante a execução da função acima, a próxima
função começará a ser executada com o semáforo da rua 1 estando
 * no vermelho (olhar descrição da função pedestre), então seria desnecessário ele mudar
para amarelo. Se o botão não for pressionado durante
 * a função acima, então o último valor que a variável 'cont' receberá é 3001, e se o
botão tiver sido pressionado e a função pedestre chamada,
 * então o ultimo valor que 'cont' receberá é 3002. Logo, a função 'if' fará com que o
semáforo da rua 1 só vá para o amarelo se ele estivesse
 * verde antes(o botão não foi pressionado).
 */
if(cont==3001){

```

```

for(cont=1;cont<=300;cont++){
    digitalWrite(R1,LOW);
    digitalWrite(G1,LOW);
    digitalWrite(Y1,HIGH);
    delay(10);
    if(digitalRead(buttonPin)==HIGH && millis() - previousMillis > 10000){
        for(cont=1;cont<=300;cont++){
            delay(10);
        }
        pedestre();
        cont = 301;
    }
}

/*O led vermelho da rua 1 é ligado, e as duas ruas ficam com semáforo fechado por 2 segundos */
for(cont=1;cont<=200;cont++){
    digitalWrite(Y1,LOW);
    digitalWrite(R1,HIGH);
    delay(10);
    if(digitalRead(buttonPin)==HIGH && millis() - previousMillis > 10000){
        pedestre();
        cont = 2001;
    }
}

/*Agora a rua 2(B) é aberta, seu semáforo fica verde por 10 segundos.Se as condições para a via para pedestres abrir forem verdadeiras, o
* semáforo da rua 1 primeiro ficará amarelo por 3 segundos e depois a função pedestre será executada.
*/
for(cont=1;cont<=1000;cont++){
    digitalWrite(R2,LOW);
    digitalWrite(G2,HIGH);
    delay(10);
    if(digitalRead(buttonPin)==HIGH && millis() - previousMillis > 10000){
        for(cont=1;cont<=300;cont++){
            digitalWrite(G2,LOW);
            digitalWrite(Y2,HIGH);
            delay(10);
        }
        pedestre();
        cont = 1002;
    }
}

/*Caso o botão tenha sido pressionado durante a execução da função acima, a próxima função começará a ser executada com o semáforo da rua 2 estando
* no vermelho (olhar descrição da função pedestre), então seria desnecessário ele mudar para amarelo. Se o botão não for pressionado durante
* a função acima, então o último valor que a variável 'cont' receberá é 1001, e se o botão tiver sido pressionado e a função pedestre chamada,
* então o último valor que 'cont' receberá é 1002. Logo, a função 'if' fará com que o semáforo da rua 2 só vá para o amarelo se ele estivesse
* verde antes(o botão não foi pressionado).
*/
if(cont==1001){
    for(cont=1;cont<=300;cont++){
        digitalWrite(R2,LOW);
        digitalWrite(G2,LOW);
        digitalWrite(Y2,HIGH);
        delay(10);
        if(digitalRead(buttonPin)==HIGH && millis() - previousMillis > 10000){
            for(cont=1;cont<=300;cont++){
                delay(10);
            }
            pedestre();
            cont = 301;
        }
    }
}

/*O led vermelho da rua 2 é ligado, e as duas ruas ficam com semáforo fechado por 2 segundos.*/
for(cont=1;cont<=200;cont++){
    digitalWrite(Y2,LOW);

```

```

    digitalWrite(R2,HIGH);
    delay(10);
    if(digitalRead(buttonPin)==HIGH && millis() - previousMillis > 10000){
        pedestre();
        cont = 201;
    }
}
}

void pedestre() {

/*As duas ruas primeiro ficarão fechadas por 2 segundos junto com a via para
pedestres.*/
for(cont=1;cont<=200;cont++){
    digitalWrite(Y1,LOW);
    digitalWrite(R1,HIGH);
    digitalWrite(Y2,LOW);
    digitalWrite(R2,HIGH);
    delay(10);
}

/*A led verde do semáforo dos pedestres ficará acessa por 15 segundos.*/
for(cont=1;cont<=1500;cont++){
    digitalWrite(Rp,LOW);
    digitalWrite(Gp, HIGH);
    delay(10);
}

/*A led verde e a vermelha do semáforo dos pedestres ficarão acessas por 3 segundos,
indicando 'alerta'*/
for(cont=1;cont<=300;cont++){
    digitalWrite(Rp,HIGH);
    delay(10);
}

/*A led verde é desligada*/
    digitalWrite(Gp,LOW);

/*Para a função 'pedestre' ser chamada, o botão deve ser pressionado e 10 segundo devem
ter sidos transcorridos após sua última execução,
* então o valor do tempo atual(função 'millis') é dado a uma variável quando a execução
da função 'pedestre' chega ao fim, então a diferença
* entre 'millis()' e essa variável dirá quanto tempo passou desde a última vez que a
via para pedestres foi aberta(função 'pedetre' executada).
*/
    previousMillis = millis();
}

```