

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DO MARANHÃO  
ENGENHARIA INDUSTRIAL ELÉTRICA**

LUIS HENRIQUE COSTA – 20211EE0002

MATHEUS ESPINDOLA ROCHA PEREIRA – 20211EE0005

REINIER SOARES BERTHIER - 20211EE0021

**Implementação do Semáforo**

São Luís -MA

2022

LUIS HENRIQUE COSTA – 20211EE0002  
MATHEUS ESPINDOLA ROCHA PEREIRA – 20211EE0005  
REINIER SOARES BERTHIER - 20211EE0021

### **Implementação do Semáforo**

Projeto apresentado à disciplina de Arquitetura de Sistemas Digitais, ministrada pela prof<sup>a</sup> Lucilene Mouzinho- IFMA, São Luís - Monte Castelo.

São Luís -MA

2022

## PROJETO DE IMPLEMENTAÇÃO DE SEMÁFORO

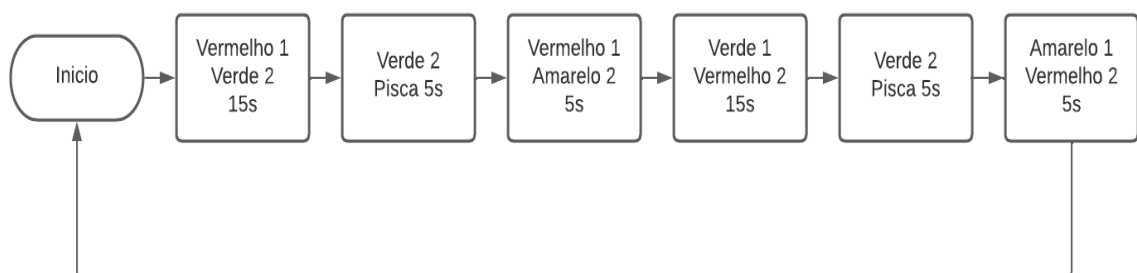
Grupo	Descrição do projeto	Material
Luis, Matheus, Reinier	a. Implementação do semáforo	AVR Studio, Proteus, fritzing, placa Arduino, leds, resistores,....

### Descrição:

Implemente um sistema que opere 2 semáforos com 3 cores básicas (vermelho, amarelo, verde), seguindo a sequência de tempo, conforme tabela abaixo.

Vermelho1	Amarelo1	Verde1	vermelho2	Amarelo2	Verde2	Tempo(s)
1	0	0	0	0	1	15
1	0	0	0	0	pisca	5
1	0	0	0	1	0	5
0	0	1	1	0	0	15
0	0	pisca	1	0	0	5
0	1	0	1	0	0	5

### • Fluxograma:



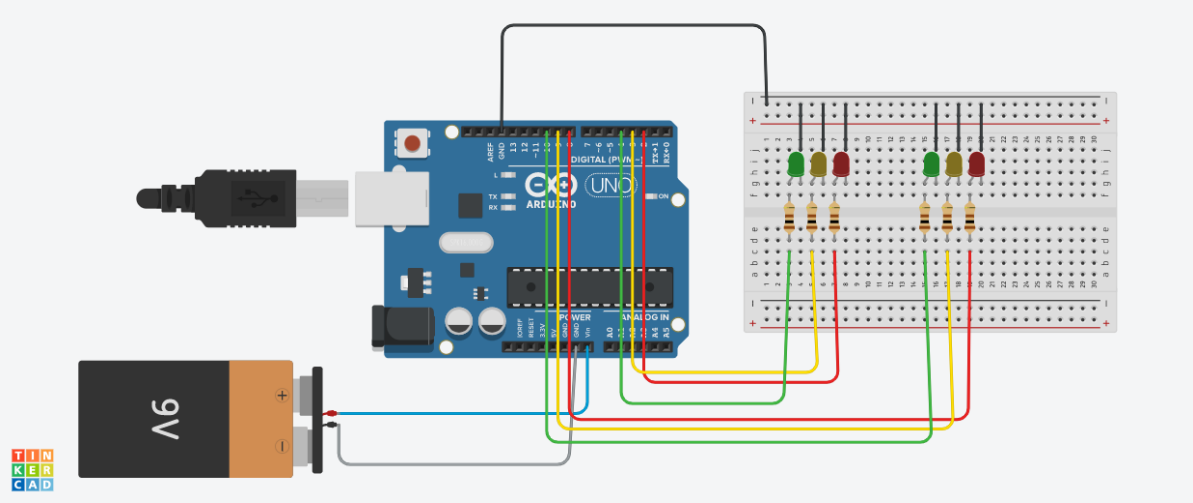
**OBS:** Todas as simulações gravadas serão enviadas na plataforma Classroom com seus respectivos nomes de acordo com a forma como o projeto foi executado. Dentre os 3 videos um foi gravado no dia 5 de Julho no kit do laboratório , outro na plataforma Tinkercad com a programação em C++ e o último no PROTEUS com o código em Assembly. Todos os códigos também serão mostrados a seguir e serão enviados na plataforma Classroom em formato de Pdf juntamente com seus arquivos.

- **Simulador TINKERCAD**

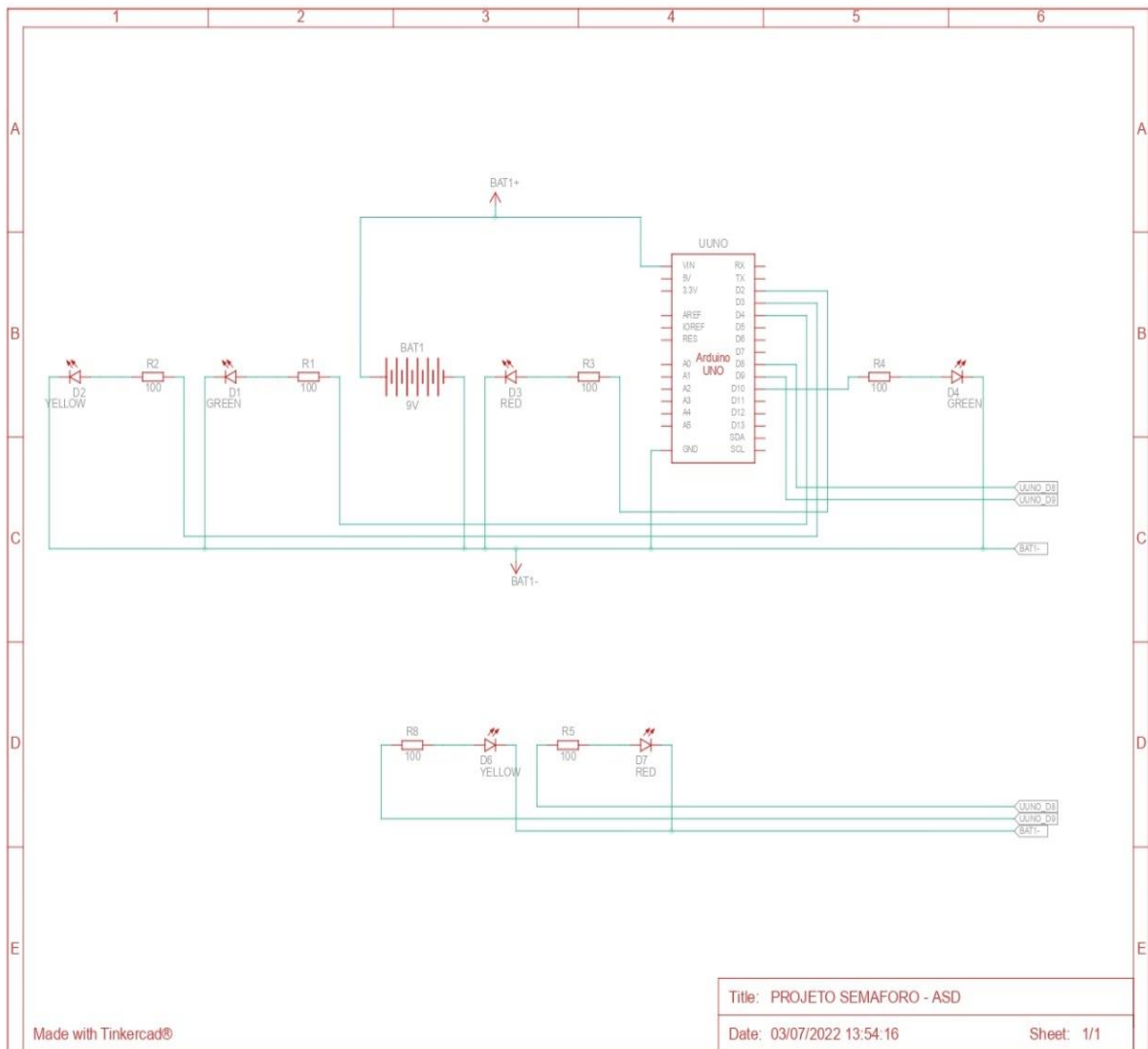
**Lista de Componentes:**

Nome	Quantidade	Componente
UUno	1	Arduino Uno R3
R1 R2 R3 R4 R8 R5	6	100 $\Omega$ Resistor
D1 D4	2	Verde LED
D2 D6	2	Amarelo LED
D3 D7	2	Vermelho LED
BAT1	1	Bateria 9V

**Vista do Circuito:**



### Vista Esquemática:



### Programa em C++ (IDE ARDUINO):

```
void setup() {
  pinMode(2, OUTPUT); // vermelho_1
  pinMode(10, OUTPUT); // verde_2
  pinMode(9, OUTPUT); // amarelo_2
  pinMode(8, OUTPUT); // vermelho_2
  pinMode(4, OUTPUT); // verde_1
  pinMode(3, OUTPUT); // amarelo_1
}

void loop() {
  digitalWrite(2, HIGH); // vermelho_1
  digitalWrite(10, HIGH); // verde_2
  delay(15000); // Esperar por 15000 milisegundos
  for (int i = 0; i <= 4; i++) { // VERDE 2 PISCANDO
    digitalWrite(10, LOW);
    delay(1000);
    digitalWrite(10, HIGH);
  }
  digitalWrite(10, LOW); // verde_2
  digitalWrite(9, HIGH); // amarelo_2
  delay(5000); // Esperar por 5000 milisegundos
  digitalWrite(9, LOW); // amarelo_2
  digitalWrite(2, LOW); // vermelho_1
  digitalWrite(8, HIGH); // vermelho_2
  digitalWrite(4, HIGH); // verde_1
  delay(15000); // Esperar por 15000 milisegundos
  for (int i = 0; i <= 4; i++) { // VERDE 1 PISCANDO
    digitalWrite(4, LOW);
    delay(1000);
    digitalWrite(4, HIGH);
  }
  digitalWrite(4, LOW); // verde_1
  digitalWrite(3, HIGH); // amarelo_1
  delay(5000); // Esperar por 5000 milisegundos
  digitalWrite(3, LOW); // amarelo_1
  digitalWrite(8, LOW); // vermelho_2
}
```

## Programa em ASSEMBLY:

```
;IFMA - CAMPUS MONTE CASTELO - ENGENHARIA ELÉTRICA INDUSTRIAL
;Período - 2022.1 - Professora: Lucilene Mouzinho
;Integrantes: Reinier Soares(20211EE0021),
;             Luis Henrique(20211EE0002),
;             Matheus Espíndola(20211EE0005)

//-----Projeto Semaforo-----//
.INCLUDE "M328DEF.INC"
.org 0x0000

//Instrução----Operando-----Comentário-----

// Vermelho 1
SBI          DDRB,0          ;Bit=1, faz pino 0 da portaB (PB0) como pino de saída

// Amarelo 1
SBI          DDRB,1          ;Bit=1, faz pino 1 da portaB (PB1) como pino de saída

// Verde 1
SBI          DDRB,2          ;Bit=1, faz pino 2 da portaB (PB2) como pino de saída

// Vermelho 2
SBI          DDRC,0          ;Bit=1, faz pino 0 da portaC (PC0) como pino de saída

// Amarelo 2
SBI          DDRC,1          ;Bit=1, faz pino 1 da portaC (PC1) como pino de saída

// Verde 2
SBI          DDRC,2          ;Bit=1, faz pino 2 da portaC (PC2) como pino de saída

L1:
    SBI      PORTB, 0        ;Vermelho 1 Acende
```

```
SBI    PORTC, 2    ;Verde 2 Acende

RCALL   DELAY_15s   ;Chama a Função DELAY_15s

RCALL PISCA_VER2   ;Chama a Função PISCA_VERDE_2

CBI     PORTC, 2    ;Verde 2 Apaga

SBI     PORTC, 1    ;Amarelo 2 Acende

RCALL DELAY_5s      ;Chama a Função DELAY_5s

CBI     PORTC, 1    ;Amarelo 2 Apaga

CBI     PORTB, 0    ;Vermelho 1 Apaga

SBI     PORTC, 0    ;Vermelho 2 Acende

SBI     PORTB, 2    ;Verde 1 Acende

RCALL DELAY_15s     ;Chama a Função DELAY_15s

RCALL PISCA_VER1   ;Chama a Função PISCA_VERDE_1

CBI     PORTB, 2    ;Verde 1 Apaga

SBI     PORTB, 1    ;Amarelo 1 Acende

RCALL DELAY_5s      ;Chama a Função DELAY_5s

CBI     PORTB, 1    ;Apagar Amarelo 1

CBI     PORTC, 0    ;Apagar Vermelho 2
```



```

        RJMP    L1                ;Jump L1

// Função para o Delay de 1 segundo
DELAY_1s:
        LDI     R20,15            ;R20 = 21(decimal)

delay1:
        LDI     R19,105          ;R19 = 252(decimal)

delay2:
        LDI     R18,105          ;R18 = 251(decimal)

delay3:
        DEC     R18              ;Decrementa em R18

        BRNE    delay3          ;Vai pra delay3 se Z=0, ou seja quando R18!=0

        DEC     R19              ;Decrementa em R19

        BRNE    delay2          ;Vai pra delay3 se Z=0, ou seja quando R19!=0
2 |
        DEC     R20              ;Decrementa em R20

        BRNE    delay1          ;Vai pra delay3 se Z=0, ou seja quando R20!=0

        RET                     ;RETORNA

// No BRNE se a condição for verdadeira o ciclo é 2, senão o ciclo é 1.

// Função para o Delay de 15 segundos
DELAY_15s:
        RCALL   DELAY_5s        ;Chama a Função DELAY_5s

        RCALL   DELAY_5s        ;Chama a Função DELAY_5s

        RCALL   DELAY_5s        ;Chama a Função DELAY_5s
        RET                     ;RETORNA

```

```

// Função para o Delay de 5 segundos
DELAY_5s:
    RCALL        DELAY_1s    ;Chama a Função DELAY_1s

    RCALL        DELAY_1s    ;Chama a Função DELAY_1s

    RCALL        DELAY_1s    ;Chama a Função DELAY_1s

    RCALL        DELAY_1s    ;Chama a Função DELAY_1s

    RCALL        DELAY_1s    ;Chama a Função DELAY_1s

    RET          ;RETORNA

PISCA_VER1:

    CBI          PORTB, 2
    RCALL        DELAY_1s
    SBI          PORTB, 2
    RCALL        DELAY_1s
    CBI          PORTB, 2
    RCALL        DELAY_1s
    SBI          PORTB, 2
    RCALL        DELAY_1s
    CBI          PORTB, 2
    RCALL        DELAY_1s
    SBI          PORTB, 2
    RET

PISCA_VER2:

    CBI          PORTC, 2
    RCALL        DELAY_1s
    SBI          PORTC, 2
    RCALL        DELAY_1s
    CBI          PORTC, 2
    RCALL        DELAY_1s
    SBI          PORTC, 2
    RCALL        DELAY_1s

```

```

CBI        PORTC, 2
RCALL      DELAY_1s
SBI        PORTC, 2

RET        ;RETORNA

```

- **Build:**









```

ATmega328 memory use summary [bytes]:
Segment  Begin    End      Code  Data  Used    Size  Use%
-----
[.cseg]  0x000000  0x00008a   138    0   138   32768  0.4%
[.dseg]  0x000100  0x000100    0    0    0    2048  0.0%
[.eseg]  0x000000  0x000000    0    0    0    1024  0.0%


● Assembly complete, 0 errors. 0 warnings

```

- **Tamanho dos Arquivos:**

	semaforo	12/07/2022 23:40	Arquivo AWS	1 KB
	Semaforo	12/07/2022 23:41	aps File	4 KB
	AvrBuild	12/07/2022 23:49	Arquivo em Lotes ...	1 KB
	Semaforo	12/07/2022 23:49	Arquivo HEX	1 KB
	Semaforo	12/07/2022 23:49	Arquivo MAP	20 KB
	Semaforo	12/07/2022 23:49	3D Object	1 KB
	labels	12/07/2022 23:49	Arquivo TMP	3 KB
	Semaforo	12/07/2022 23:58	Arquivo ASM	4 KB

- **Arquivo HEX:**

 Semaforo - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```

|:0200000020000FC
:10000000209A219A229A389A399A3A9A289A429AA8
:100010001AD02FD04298419A1AD041982898409AE5
:100020002A9A11D01AD02A98299A11D02998409842
:10003000EDCF4FE039E629E62A95F1F73A95D9F761
:100040004A95C1F7089503D002D001D00895F1DF99
:10005000F0DFEFDFEEDFEDDF08952A98EADF2A9A7E
:10006000E8DF2A98E6DF2A9AE4DF2A98E2DF2A9A74
:1000700008954298DEDF429ADCDF4298DADF429A46
:0A008000D8DF4298D6DF429A0895B7
:000000001FF

```

- **Modelo Esquemático no PROTEUS:**

