

**Docente: José Kleiton Ewerton da Costa Martins**  
**Discentes: Ernane Ferreira, Thiago Lopes e Quelita Míriam**

### **Lista 01 – Portas de saída**

#### **Relatório (DCA0119): Comparação de Programas para Controle de Semáforo**

Este relatório visa comparar dois programas que têm como objetivo controlar um semáforo de trânsito com três estados: luz vermelha, luz amarela e luz verde. Ambos os programas alcançam a mesma funcionalidade, mas adotam abordagens diferentes para alcançar esse objetivo.

#### **Programa 1 - Manipulação Direta dos Registradores**

O primeiro programa utiliza a técnica de manipulação direta dos registradores para controlar os LEDs do semáforo. Neste caso, os LEDs estão conectados aos pinos 4, 5 e 6 do Arduino, que correspondem aos registradores DDRG, DDRE e DDRH. Os comandos equivalentes para configuração dos pinos como saída, ligar e desligar os LEDs e criar atrasos são realizados por meio de operações diretas com esses registradores. Esse método é mais complexo e requer um conhecimento mais profundo do hardware subjacente.

#### **Programa 2 - Utilização de Comandos do Arduino**

O segundo programa utiliza comandos nativos do Arduino para realizar as mesmas tarefas. Ele utiliza as funções `pinMode()` e `digitalWrite()` para configurar os pinos como saída e controlar os LEDs. Essas funções abstraem a manipulação direta dos registradores, tornando o código mais legível e acessível. Além disso, as funções `delay()` são utilizadas para criar atrasos temporais, tornando o código mais simples de compreender.

#### **Comandos Equivalentes**

Ambos os programas possuem comandos equivalentes, que desempenham funções semelhantes:

1. Configuração dos pinos como saída: Ambos utilizam comandos para configurar os pinos dos LEDs como saída.

Programa 1:

```
DDRG = 0x20; // Configura o pino 4 como saída (LED vermelho)
DDRE = 0x8;  // Configura o pino 5 como saída (LED amarelo)
DDRH = 0x8;  // Configura o pino 6 como saída (LED verde)
```

Programa 2:

```
pinMode(PIN_LED_RED, OUTPUT);  
pinMode(PIN_LED_YELLOW, OUTPUT);  
pinMode(PIN_LED_GREEN, OUTPUT);
```

2. Controle dos LEDs: Ambos utilizam comandos para ligar e desligar os LEDs.

Programa 1:

```
PORTG = 0x20; // Liga o LED vermelho  
PORTG = 0x00; // Desliga o LED vermelho  
PORTE = 0x8; // Liga o LED amarelo  
PORTE = 0x00; // Desliga o LED amarelo  
PORTH = 0x8; // Liga o LED verde  
PORTH = 0x00; // Desliga todos os LEDs
```

Programa 2:

```
digitalWrite(PIN_LED_RED, HIGH);  
digitalWrite(PIN_LED_RED, LOW);  
digitalWrite(PIN_LED_YELLOW, HIGH);  
digitalWrite(PIN_LED_YELLOW, LOW);  
digitalWrite(PIN_LED_GREEN, HIGH);  
digitalWrite(PIN_LED_GREEN, LOW);
```

3. Atrasos (Delays): Ambos utilizam comandos para criar atrasos temporais

Programa 1:

```
_delay_ms(5000); // Espera 5 segundos  
_delay_ms(2000); // Espera 2 segundos  
_delay_ms(5000); // Espera 5 segundos
```

Programa 2:

```
delay(5000); // Espera 5 segundos  
delay(2000); // Espera 2 segundos  
delay(5000); // Espera 5 segundos
```

## **Conclusão**

A escolha entre os programas depende das preferências do programador e dos requisitos do projeto. O Programa 1, com manipulação direta dos registradores, é mais adequado quando é necessário um controle preciso de baixo nível sobre o hardware. Por outro lado, o Programa 2, que utiliza comandos do Arduino, é mais simples de entender e é preferível em situações em que a facilidade de desenvolvimento e a legibilidade do código são prioritárias.