



CEFET-MG — Centro Federal de Educação Tecnológica de Minas Gerais  
DEPARTAMENTO DE COMPUTAÇÃO DE DIVINÓPOLIS — DECOM-DV

## Microprocessadores e Microcontroladores

### Primeira Atividade Avaliativa

Aluno: PEDRO HENRIQUE PIRES DIAS

Valor: 30 pontos (cada questão vale 5 pontos)

Turma: 2024/1

Prof. M. Sc. Diego Ascânio Santos

Respostas:

1      2      3      4      5      6

-----

### Questão 1

Cláudia perguntou ao professor Ascânio se sua rotina de interrupções de overflow do `Timer2` estava correta para contar intervalos de tempo de 10 em 10 segundos, pois, estava com dúvidas se seus cálculos de quantidade de *overflows* e o modo de *prescaling* que havia definido para o `Timer2` estavam corretos. O professor Ascânio verificou o código apresentado por Cláudia e disse que tanto os cálculos quanto o *prescaling* estavam corretos, mas, que a rotina de interrupção de overflow do `Timer2` — `Flag TOIE2` — estava desabilitada.

Qual deve ser a instrução que Cláudia deve adicionar à função `setup()` de seu código para habilitar a interrupção de overflow do `Timer2`?

- a) `TIMSK2 = TIMSK2 | 0b00000001;`
- b) `TIMSK2 = 0b00000000;`
- c) `TIMSK2 = TIMSK2 | 0b00000000;`
- d) `TIMSK2 = TIMSK2 | 0b00000010;`
- e) `TIMSK2 = TIMSK2 | 0b00000100;`

### Questão 2

(ENADE 2005 - 11) Apesar de todo o desenvolvimento, a construção de computadores e processadores continua, basicamente, seguindo a arquitetura clássica de von Neumann. As exceções a essa regra encontram-se em computadores de propósitos específicos e nos desenvolvidos em centros de pesquisa. Assinale a opção em que estão corretamente apresentadas características da operação básica de um processador clássico:

- a) Instruções e dados estão em uma memória física única; um programa é constituído de uma sequência de instruções de máquina; uma instrução é lida da memória de acordo com a ordem dessa sequência e, quando é executada, passa-se, então, para a próxima instrução na sequência.
- b) Instruções e dados estão em memórias físicas distintas; um programa é constituído de um conjunto de instruções de máquina; uma instrução é lida da memória quando o seu operando-destino necessita ser recalculado; essa instrução é executada e o resultado é escrito no operando de destino, passando-se, então, para o próximo operando a ser recalculado.
- c) Instruções e dados estão em uma memória física única; um programa é constituído de um conjunto de instruções de máquina; uma instrução é lida da memória quando todos os seus operandos-fonte estiverem prontos e disponíveis; essa instrução é executada e o resultado é escrito no operando de destino, passando-se, então, para a instrução seguinte que tiver todos seus

operandos disponíveis.

d) Instruções e dados estão em memórias físicas distintas; um programa é constituído de um conjunto de instruções de máquina; uma instrução é lida da memória quando todos os seus operandos-fonte estiverem prontos e disponíveis; essa instrução é executada e o resultado é escrito no operando de destino, passando-se, então, para a instrução seguinte que estiver com todos os seus operandos disponíveis.

e) Instruções e dados estão em memórias físicas distintas; um programa é constituído de uma sequência de instruções de máquina; uma instrução é lida da memória de acordo com a ordem dessa sequência e, quando é executada, passa-se, então, para a próxima instrução na sequência.

---

### Questão 3

Leia o seguinte código em Arduino:

```
int greenLED = 3, redLED = 4;
int leftButton = 6, rightButton = 7;

void setup() {
    pinMode(greenLED, OUTPUT);
    pinMode(redLED, OUTPUT);
    pinMode(leftButton, INPUT_PULLUP);
    pinMode(rightButton, INPUT_PULLUP);
}

void loop() {
    while (digitalRead(leftButton) == LOW) {
        digitalWrite(redLED, HIGH);
    }
    while (digitalRead(rightButton) == LOW) {
        digitalWrite(greenLED, HIGH);
        delay(1000);
        digitalWrite(greenLED, LOW);
        delay(1000);
    }
    while (digitalRead(leftButton) == HIGH && digitalRead(rightButton) == HIGH) {
        digitalWrite(greenLED, LOW);
        digitalWrite(redLED, LOW);
    }
}
```

O que este programa faz?

- a) Acende o LED verde quando `rightButton` é pressionado e apaga o LED vermelho quando `leftButton` é pressionado.
- b) Apaga os LEDs se nenhuma das teclas for pressionada, acende o LED vermelho enquanto `leftButton` é pressionado e pisca o LED verde de 1 em 1 segundo enquanto `rightButton` é pressionado.
- c) Acende o LED verde quando `rightButton` é pressionado e apaga o LED vermelho quando `leftButton` é pressionado, mas não apaga os LEDs se nenhuma das teclas for pressionada.
- d) Apaga os LEDs se as teclas forem processionadas, acende o LED vermelho enquanto `leftButton` não é pressionado e pisca o LED verde de 1 em 1 segundo enquanto `rightButton` não é pressionado.
- e) Nenhuma das alternativas anteriores, pois, não existe configuração de entrada do tipo `INPUT_PULLUP`.

---

### Questão 4

Quanto as interrupções, avalie as assertivas:

- I. Interrupção é um mecanismo que permite a uma entidade externa interromper a execução de um programa sendo executado.
- II. Chegada de dados em uma porta de entrada/saída pode ser um exemplo de interrupção.
- III. Jammais podem ser associadas a eventos assíncronos.
- IV. O pressionamento de um botão pode ser um exemplo de interrupção.

Quais são verdadeiras?

- a) I, II, III e IV.
  - b) I, II e IV.
  - c) I e II apenas.
  - d) I e IV apenas.
  - e) II e IV apenas.
- 

### Questão 5

Raul deseja fazer um LED comutar de estado a cada 5 segundos. Sem muito critério do entendimento viu nos códigos dos slides do prof. Ascânio o seguinte ISR que fazia o LED comutar de estado a cada 5 segundos:

#### Rotina de interrupção de Overflow associada ao `Timer2`

```
ISR(TIMER2_OVF_vect) {
    overflows++;
    if (overflows == 306) {
        overflows = 0;
        // Toggle the LED state
        digitalWrite(13, !digitalRead(13));
    }
}
```

Entretanto, ao copiar os códigos, não se atentou ao modo correto de funcionamento e configurou — sem querer — o prescaler do `Timer2` para o fator de 256, através do registrador de controle `TCCR2B` pela seguinte instrução na função `setup()`:

```
TCCR2B = 0b00000110;
```

Com isso, em vez do LED comutar de estado a cada 5 segundos, ele comutava a cada 1.25 segundos (aproximadamente). Qual dos seguintes modos de *prescaling* do `Timer2` faz com que o LED comute de estado a cada 5 segundos, como deseja Raul?

- a) `TCCR2B = 0b00000001;`
  - b) `TCCR2B = 0b00000111;`
  - c) `TCCR2B = 0b00000011;`
  - d) `TCCR2B = 0b00000000;`
  - e) `TCCR2B = 0b00000010;`
- 

### Questão 6

Interrupções no arduino podem ser habilitadas e associadas a funções de *callback* — que são executadas quando interrupções ocorrem — através do **ISR** (Interrupt Service Routine). Em relações aos recursos e limitações que as funções de *callback* apresentam, é correto afirmar que:

- a) Funções de *callback* de interrupções podem receber argumentos e retornar valores.
- b) É possível executar múltiplos *callbacks* ao mesmo tempo.
- c) Funções de *callback* não podem receber argumentos e nem retornar valores.
- d) Podem ser interrompidas por outras interrupções.
- e) Nenhum tipo de modificador de variável precisa ser utilizado para permitir a modificação de variáveis globais na função de *callback* e na função principal.