



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

## Microprocessadores e Microcontroladores

### Primeira Atividade Avaliativa

**Aluno:** GABRIEL HENRIQUE SILVA GONTIJO

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

Considere o código abaixo:

```
const int ledPin = 13;
const int interruptPin = 2; // only pin 2 and 3 can be used for interrupts

volatile int state = LOW;

void blink() { // ISR function
    state = !state; // toggle the state
}

// missing setup function

void loop() {
    digitalWrite(ledPin, state);
}
```

É desejado que o LED conectado ao pino 13 comute de estado a cada vez que o botão conectado ao pino 2 for pressionado. O *pushbutton* conectado ao pino 2 do arduino também está conectado ao pino GND do microcontrolador.

Qual alternativa contém a implementação da função `setup()` que atende a esse requisito no Arduino UNO?

a)

```
void setup() {
    pinMode(2, INPUT_PULLUP);
    pinMode(ledPin, OUTPUT);
    attachInterrupt(
        digitalPinToInterrupt(interruptPin),
        blink,
        KEEPING
    );
}
```

b)

```
void setup() {
    pinMode(2, INPUT_PULLUP);
    pinMode(ledPin, OUTPUT);
    attachInterrupt(
        digitalPinToInterrupt(interruptPin),
        blink,
        CHANGE
    );
}
```

```
}
```

c)

```
void setup() {  
    pinMode(2, INPUT_PULLUP);  
    pinMode(ledPin, OUTPUT);  
    attachInterrupt(  
        digitalPinToInterrupt(interruptPin),  
        blink,  
        HIGH  
    );  
}
```

d)

```
void setup() {  
    pinMode(2, INPUT_PULLUP);  
    pinMode(ledPin, OUTPUT);  
    attachInterrupt(  
        digitalPinToInterrupt(interruptPin),  
        blink,  
        FALLING  
    );  
}
```

e)

```
void setup() {  
    pinMode(2, INPUT_PULLUP);  
    pinMode(ledPin, OUTPUT);  
    attachInterrupt(  
        digitalPinToInterrupt(interruptPin),  
        blink,  
        LOW  
    );  
}
```

---

## Questão 2

A respeito de circuitos digitais — portas lógicas, armazenadores de estado, registradores, memórias, somadores (subtratores), processadores, dentre outros — julgue os itens a seguir:

- I. A porta lógica XNOR produz nível lógico alto na saída quando as entradas são diferentes.
- II. A porta lógica XNOR produz nível lógico alto na saída quando as entradas são iguais.
- III. *Flip-flops* são armazenadores de estado que podem ser utilizados para armazenar apenas um bit de informação.
- IV. Registradores são armazenadores de estado capazes de armazenar  $n$  bits de informação construídos tipicamente a partir da associação de uma quantidade  $n$  de *flip-flops*.
- V. A quantidade de *flip-flops* em um registrador não define a quantidade de bits de informação que ele pode armazenar.
- VI. Em um registrador comercial de 8 bits cuja habilitação de escrita é ativa em nível lógico baixo, quando o sinal de habilitação de escrita está em nível lógico alto, o registrador comporta-se como elemento inexistente no circuito, como um elemento de alta impedância.
- VII. Memórias de acesso aleatório (RAM) possuem barramentos distintos para escrita de dados e leitura de dados.
- VIII. Memórias de acesso aleatório (RAM) usam o mesmo barramento para escrita e leitura de dados.
- IX. Em um processador de 8 bits a unidade lógica aritmética consegue por padrão realizar operações de soma e subtração de números inteiros que variam de  $-128$  a  $+127$ .
- X. Em um processador de 8 bits a unidade lógica aritmética consegue por padrão realizar operações de soma e subtração de números naturais que variam de  $0$  a  $+255$ .

Assinale a alternativa que contém somente os itens corretos.

- a) I, II, III, IV, V, VI, VII, VIII, IX e X.
- b) I, III, IV, V, VI, VII, VIII, IX e X.

- c) II, III, IV, VI, VII, IX e X.
- d) II, III, IV, VII, e IX.
- e) II, III, IV, VI, VIII, IX e X.

---

### Questão 3

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 4

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 = 0b00000000;`

- b) `TCCR2B = 0b00000001;`
  - c) `TCCR2B = 0b00000011;`
  - d) `TCCR2B = 0b00000111;`
  - e) `TCCR2B = 0b00000110;`
- 

### Questão 5

Interrupções no arduino podem ser habilitadas e associadas a funções *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* podem receber argumentos, mas, não podem retornar valores.
  - d) Podem ser interrompidas por outras interrupções.
  - e) O modificador de variável `volatile` precisa ser utilizado para permitir a modificação de variáveis globais nas funções de *callback* e na função principal.
- 

### Questão 6

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`.