CEFET-MG — Centro Federal de Educação Tecnológica de Minas Gerais

DEPARTAMENTO DE COMPUTAÇÃO DE DIVINÓPOLIS — DECOM-DV

**Microprocessadores e Microcontroladores**

**Primeira Atividade Avaliativa**

**Aluno:** EDUARDO RABELO GUIMARÃES

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**

A respeito de interrupções:

a) A função `attachInterrupt()` é a única forma de se configurar uma interrupção no Arduino.

b) `HIGH` é um modo válido de acionamento de interrupção no Arduino UNO.

c) As *callbacks* de interrupção podem ter quantos argumentos forem necessários.

d) Só é possível tratar interrupções externas no Arduino UNO pelos pinos 2 e 3.

e) As *callbacks* de interrupção podem retornar quantos valores forem necessários.

---

**Questão 2**

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 3**

(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 seqüência de instruções de máquina; uma instrução é lida da memória de acordo com a ordem dessa seqüência e, quando é executada, passa-se, então, para a próxima instrução na seqüência.

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 seqüência de instruções de máquina; uma instrução é lida da memória de acordo com a ordem dessa seqüência e, quando é executada, passa-se, então, para a próxima instrução na seqüência.

---

**Questão 4**

Avalie as assertivas:

I. Todos os sinais de circuitos são elétricos, porém, podem ser categorizados em dois tipos: analógicos e digitais.

II. Não é possível para sinais analógicos assumirem qualquer valor arbitrário dentro de um intervalo especificado.

III. Sinais digitais são representados por valores contínuos.

IV. Sinais digitais são representados por valores discretos.

V. A tensão digital em nível lógico ALTO em circuitos digitais de lógica CMOS é de +5V.

Assinale a alternativa que contém todas as assertivas corretas:

a) I, II, III e V.

b) I, III e V.

c) II, III, IV e V.

d) I, IV e V.

e) I e IV.

---

**Questão 5**

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 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`.