



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 Rodrigues Pereira

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 memórias escolha a alternativa falsa:

- a) Se minha memória tem 16 pinos de endereço e 8 pinos de dados então ela consegue armazenar 65536 bytes de informação.
- b) Uma memória com 32 pinos de endereço e 32 bits de dados consegue armazenar 16GB de informação.
- c) Uma memória com 32 pinos de endereço e 32 bits de dados consegue armazenar 4GB de informação.
- d) Uma memória RAM do tipo dinâmica depende de atualizações periódicas conhecidas como *refresh* para manter suas informações armazenadas.
- e) Toda memória ROM foi uma PROM até ter sido fabricada.

Questão 2

Considerando a seguinte rotina de interrupção de *overflow* associada ao `Timer2`, que o `Timer2` está configurado para operar no modo normal (modo timer) e que a frequência do *clock* do `Timer2` é de 16 MHz, qual fator de prescaler faz com que o estado do LED conectado ao pino digital 13 comute a (aproximadamente) 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));  
    }  
}
```

- a) 1.
- b) 16.
- c) 64.
- d) 256.

e) 1024.

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

Acerca de entradas e saídas digitais no Arduino e a conexão de periféricos ao microcontrolador, avalie as seguintes assertivas:

- I. O comando `pinMode(13, OUTPUT)` configura o pino 13 como saída.
- II. Para uma conexão em série do pino +5V do Arduino a um resistor de 330Ω passando pelo ânodo de um LED, em sequência pelo cátodo do LED e finalizando no pino 13 do Arduino, o comando `digitalWrite(13, HIGH)` acenderá o LED se ele estiver apagado.
- III. Para uma conexão em série do pino +5V do Arduino a um resistor de 330Ω passando pelo ânodo de um LED, em sequência pelo cátodo do LED e finalizando no pino 13 do Arduino, o comando `digitalWrite(13, HIGH)` apagará o LED se ele estiver aceso.
- IV. O comando `digitalWrite(13, !digitalRead(13))` inverte o estado do dispositivo conectado ao pino 13 qualquer que seja seu modo de conexão (ligado ao +5V ou ao GND do Arduino).

São corretas as assertivas:

- a) I e II, apenas.

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

A respeito do ambiente de desenvolvimento do Arduino, da sua linguagem de programação *sketch* e de demais conceitos relacionados, julgue os itens a seguir.

- I. A função `setup()` é executada uma única vez, quando o programa é inicializado;
- II. A função `setup()` é usada para inicializar configurações e preparar o estado inicial do programa;
- III. A função `loop()` é executada continuamente, em um loop infinito, até que o microcontrolador seja desligado;
- IV. A instrução `pinMode(3, INPUT_PULLUP);` configura o pino 3 como entrada digital com nível lógico invertido;
- V. O Arduino pode acionar diretamente atuadores — elementos de carga — de baixa potência e que suportem o nível de tensão de saída do microcontrolador;
- VI. Para atuadores de média / alta potência é necessário o uso de circuitos auxiliares de acionamento comumente baseados em transistores.

Estão incorretos:

- a) Nenhum item está incorreto.
- b) I, III e V.
- c) II, IV e VI.
- d) I, II, III, V e VI.
- e) Todos os itens estão incorretos.