



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

Microprocessadores e Microcontroladores

Primeira Atividade Avaliativa

Aluno: LUCAS FARINELLI CRIVELLARI DE PINHO

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

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 2

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 3

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 +5V do microcontrolador.

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

a)

```

void setup() {
    pinMode(ledPin, OUTPUT);
    attachInterrupt(
        digitalPinToInterrupt(interruptPin),
        blink,
        RISING
    );
}

```

b)

```

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

```

c)

```

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

```

d)

```

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

```

e)

```

void setup() {
    pinMode(ledPin, OUTPUT);
    attachInterrupt(
        digitalPinToInterrupt(interruptPin),
        blink,
        COMMUTING
    );
}

```

Questão 4

A respeito de entradas e saídas digitais do Arduino, resistores pull-up e pull-down, contatos normalmente abertos e normalmente fechados, avalie as assertivas:

I. Por padrão as entradas digitais do Arduino estão preparadas para receber sinais digitais em nível lógico TTL (0V a 5V).

II. Não é necessário realizar quaisquer tipos de adaptações para conectar circuitos digitais não-TTL

(por exemplo, CMOS) ao Arduino.

III. Um contato normalmente aberto é um contato que, em repouso, não permite a passagem de corrente elétrica.

IV. O nível lógico de uma entrada digital do Arduino conectada a um contato normalmente aberto em seu estado de repouso é sempre 0V.

V. Resistores pull-up são utilizados para garantir que uma entrada digital do Arduino esteja sempre em nível lógico alto em seu estado padrão.

VI. O Arduino não dispõe de resistores pull-up internos, sendo necessário adicionar resistores externos para este fim.

São falsas as assertivas:

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

Questão 5

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 6

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