



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 FERNANDES BARBOSA DE ANDRADE

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

Quanto a entradas digitais no Arduino, é correto afirmar que:

- a) Esperar um tempo após a leitura de um pino digital — técnica conhecida como **debounce** — é uma boa prática para garantir que o valor lido seja estável.
- b) A comutação de chaves mecânicas é imune ao aparecimento de ruídos, efeito conhecido como **bouncing**.
- c) Resistores de *pull-up* fazem com que o estado padrão de uma entrada digital seja nível lógico BAIXO.
- d) Resistores de *pull-down* fazem com que o estado padrão de uma entrada digital seja nível lógico ALTO.
- e) Nenhum dos itens anteriores está correto.

Questão 2

A respeito das fontes de interrupção no Arduino avalie as seguintes afirmativas:

- I. O temporizador do Arduino pode ser configurado para gerar interrupções em intervalos regulares de tempo.
- II. O ADC pode ser configurado para gerar uma interrupção quando a conversão analógico-digital é concluída.
- III. Quaisquer pinos do Arduino UNO podem ser configurados como pinos de interrupção externos.
- IV. Não é possível associar interrupções para o recebimento de dados pela porta serial.

Quais são verdadeiras?

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

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 = 0b00000000;`
 - b) `TCCR2B = 0b00000001;`
 - c) `TCCR2B = 0b00000011;`
 - d) `TCCR2B = 0b00000111;`
 - e) `TCCR2B = 0b00000110;`
-

Questão 4

A respeito de saídas digitais do arduino escolha a alternativa correta:

- a) O arduino é capaz de acionar diretamente quaisquer tipos de cargas, independente da corrente que elas demandem.
 - b) Não é possível realizar configurações de acionamento de cargas em nível lógico BAIXO.
 - c) Em um circuito para acionamento de um motor de corrente contínua não é necessário adicionar um diodo para proteção do circuito.
 - d) As saídas digitais do Arduino podem, virtualmente, controlar quaisquer tipos de dispositivos atuadores, cada qual com suas particularidades.
 - e) Nenhuma das alternativas anteriores.
-

Questão 5

Júlio procurou seu professor de microcontroladores durante a aula alegando que o código que ele havia escrito para piscar o led interno (Led 13) do Arduino de 500 em 500ms não estava funcionando. O professor pediu para que Júlio mostrasse o código que havia escrito e assim, Júlio o fez:

```
void setup() {  
    pinMode(13, INPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(500);  
    digitalWrite(13, LOW);  
}
```

Qual das seguintes alternativas resolve o problema de Júlio?

a) Trocar a função `pinMode(13, INPUT);` por `pinMode(13, OUTPUT);` — na linha 2 — da função `setup()`.

b) Adicionar uma instrução `delay(500);` logo abaixo da instrução `digitalWrite(13, LOW);` — que está na linha 8 — da função `loop()`.

c) Retirar a instrução `delay(500)` — que está na linha 7 — da função `loop()`.

d) Trocar a função `pinMode(13, INPUT);` por `pinMode(13, OUTPUT);` — na linha 2 — da função `setup()` e reescrever a função `loop()` da seguinte forma:

```
void loop() {  
    digitalWrite(  
        13, !digitalRead(13)  
    );  
    delay(500);  
}
```

e) Nenhuma das alternativas anteriores resolve o problema de Júlio

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.