

Instituição: Instituto Federal de Educação, Ciência e Tecnologia - Paraíba (IFPB).

Disciplina: Microprocessadores e microcontroladores.

Professor: Fagner de Araujo Pereira.

Aluno (a): _____

Exercício avaliativo 2 (Peso 100 pontos) - Reposição

1. Explique como ocorre o processo de boot nos microcontroladores STM32 com núcleos Cortex M3/M4 até que eles possam executar o código do usuário.
2. Explique por que um projetista pode optar por usar um resistor de pull-up externo ao invés do resistor do pull-up interno do STM32F407 ao configurar um pino como saída em dreno aberto.
3. Sobre microcontroladores e sistemas microcontrolados, assinale a(s) alternativa(s) correta(s).
 - (a) O STM32 é uma plataforma de código aberto que foi concebida com o objetivo de facilitar o desenvolvimento de sistemas embarcados complexos.
 - (b) Microcontroladores da família STM32 não podem se comunicar com microcontroladores de outros fabricantes devido a sua arquitetura interna baseada em processadores ARM.
 - (c) A quantidade de memória RAM interna encontrada nos microcontroladores STM32 é um destaque positivo, permitindo sua utilização até em sistemas mais complexos, como em núcleos de processamento de placas de vídeo de notebooks.
 - (d) Microcontroladores STM32 vêm substituindo, com sucesso, até mesmo portas lógicas básicas, realizando as mesmas operações com maior velocidade.
 - (e) A maioria dos microcontroladores STM32 possui diversos periféricos integrados de grande utilidade, como conversores AD, Timers, portas USB, Ethernet, portas seriais SPI, I2C, entre outros.
4. Apresente um código que realiza a configuração do pino PD4 para operar como saída em dreno aberto em *medium speed* e com resistor de *pull-up* habilitado no STM32F407. (*Obs. Não é permitido a alteração das configurações de nenhum outro pino do microcontrolador*).
5. O código mostrado a seguir configura o sistema de clock de um microcontrolador STM32F407 que usa um cristal externo de 8MHz.

```
void Configure_Clock(void)
{
    #define PLL_M      2
    #define PLL_N      60
    #define PLL_P      3

    //Configura a fonte de clock (HSE), os fatores de divisão e multiplicação do PLL,
    //prescalers e dos barramentos AHB, APB
    RCC->CR |= 0x00010000;          //habilita HSE

    RCC->CFGR |= 0x00009400;          //HCLK = SYSCLK/1, PCLK2 = HCLK/2, PCLK1 = HCLK/4

    //configura a fonte de clock e os parâmetros do PLL
    RCC->PLLCFGR = PLL_M | (PLL_N << 6) | (((PLL_P >> 1) - 1) << 16) | (0x00400000)
    RCC->CR |= 0x01000000;          //habilita o PLL

    //seleciona o PLL como fonte de SYSCLK
    RCC->CFGR |= 0x00000002;
}
```

Considerando que o pino PC10 foi previamente configurado como saída digital e em seguida foi colocado para alternar o seu estado lógico pelo código mostrado abaixo, determine por quanto tempo o pino permanece em nível lógico alto em cada ciclo, considerando que cada instrução é executada a cada pulso de clock.

```
while(1)
{
    GPIOC->ODR ^= (1 << 10);
}
```