

Exercicio 8 – resposta parcial.

// 1. Ative o *clock* para a porta de GPIO usado pelo canal ADC.

```
SIM_SCGC5 |= 0x400 ; /clock porta B
```

// 2. Defina o bit MUX do PORTX_PCRn para o pino de entrada do ADC.

```
PORTB_PCR0 &= 0xFFFF F8FF ;
```

// Configuração do pino PTB19 (Led verde) para acionar o LED.

```
PORTB_PCR19 |= 0x100;
```

```
PORTB_PCR19 &= 0xFFFF F9FF;
```

```
GPIO_PDDR |= 0x8 0000 ;
```

//3. Ative o *clock* para o módulo ADC usando o registrador SIM_SCGC6.

```
SIM_SCGC6 |= 0x800 0000 ;
```

// 4. Escolha o tipo de trigger para iniciar a conversão analógica-digital usando o registrador ADC0_SC2.

```
ADC0_SC2 &= 0xFFFF FFBF ; // ou ~(0x40)
```

// 5. Escolha a fonte de *clock* e a resolução usando o registrador ADC0_CFG1.

```
ADC0_CFG1 &= 0xFFFF FF9F ;
```

// 6. Selecione o canal de entrada ADC usando o registro ADC0_SC1A. Certifique-se que a interrupção não está habilitada e que está usando a opção-single ended.

```
While(1) { //9. Repita as etapas 6 a 8 para a próxima conversão.
```

```
ADC0_SC1A &= 0xFFFFF9F ; //interrupção e diferencial desligados
```

```
ADC0_SC1A &= 0xFFFF FFE8 ;
```

```
ADC0_SC1A |= 0x8 ;
```

```
int saida;
```

// 7. Monitore quando o *flag* de fim de conversão (COCO) no registrador ADC0_SC1A.

```
while (~(ADC0_SC1A & 0x80)) {}
```

// 8. Quando o *flag* COCO for setado, leia o resultado da conversão em ADC0_RA e salve-o.

```
saida = ADC0_RA ;
```

```
if (saida > 0xF0)
```

```
{
```

```
GPIO_PDOR &= 0xFFFF7 FFFF ;
```

```
}
```

