

Relatório do Experimento 1

Introdução ao Hardware (FRDM KL25 e shield EA871)

Henrique Aires Silva
RA: 169574 - Turma T
March 13, 2017

QUESTÃO 3

Vamos praticar mais o que acabamos de ver? Quais alterações você faria no programa 4 para substituir a cor vermelha pela cor verde do led RGB, para alternar o estado do led com uso de GPIOx_PTOR e para evitar uso desnecessário do qualificador volatile ?

Para alterar a cor do LED verde ao invés do vermelho, devemos configurar e atuar no pino correspondente a ele (PORTB_19). Primeiro, devemos criar outra macro para controlar o MUX do pino 19 do Port B, do seguinte modo:

```
/*! MUX do pino PTB19 (Reg. PORTB_PCR19) */  
#define PORTB_PCR19 (*(unsigned int volatile *) 0x4004A04C)
```

Então devemos definir a macro GPIOB_PTOR para o endereço de memória do mesmo no microcontrolador:

```
/*! Inverte o estado do bit nos pinos da porta PORTB (Reg. GPIOB_PTOR) */  
#define GPIOB_PTOR (*(unsigned int volatile *) 0x400FF04C)
```

Na função main devemos configurar o pino 19 da Port B como GPIO usando o registrador PORTB_PCR19 e sua direção como saída pelo registrador GPIOB_PDDR:

```
PORTB_PCR19 = PORTB_PCR19 & 0xFFFFF8FF; /*! Zera bits 10, 9 e 8 (MUX) de PTB19 */  
PORTB_PCR19 = PORTB_PCR19 | 0x00000100; /*! Configura PTB19 como GPIO (seta bit 8) */  
GPIOB_PDDR = GPIOB_PDDR | (1<<19); /*! Seta direção do PTB19 como saída */
```

Dentro do laço eterno da função main, escrevemos no bit 19 do registrador GPIOB_PTOR para inverter o estado do pino 19 e aplicamos um delay para percepção visual:

```
for(;;)
{
    GPIOB_PTOR = (1<<19); /*! Inverte estado da saída 19 (LED verde) */
    delay(500000);          /*! Espera um tempo */
}
```