

Insper

Handout 1 PISCA LED

Computação Embarcada
- 14 de agosto de 2017 -

Rachel Bottino

Engenharia da Computação - 2017

CARREGANDO UM EXEMPLO

1. Dê uma olhada nos exemplos fornecidos. Algum interessante?

CÓDIGOS

1. Explore os códigos contidos neste projeto.

ASF

1. Quais as bibliotecas carregadas no pisca LED?

- *asf.h*
- *stdio_serial.h*
- *conf_board.h*
- *conf_clock.h*

ENTENDENDO O CÓDIGO

1. Qual a frequência de operação do uC?
2. Qual a frequência com que o LED pisca?

O LED pisca a uma frequência de 2MHz.

3. Altere a frequência para o dobro da atual.

A frequência pode ser alterada para o dobro da atual, diminuindo pela metade o período de delay.

4. Qual a parte do código responsável por gerar a frequência com que o LED pisca? Como isso funciona?

A parte do código responsável por gerar a frequência com que o LED pisca é a função `mdelay()`:

```
// [main_ms_delay]
static void mdelay(uint32_t ul_dly_ticks)
{
    uint32_t ul_cur_ticks;

    ul_cur_ticks = g_ul_ms_ticks;
    while ((g_ul_ms_ticks - ul_cur_ticks) < ul_dly_ticks);
}
// [main_ms_delay]
```

O argumento passado na função é o período em que o LED pisca.

5. Executando e analisando o código, informe qual o uso do botão SW0?

O botão SW0 faz o LED da placa começar a piscar, se ele estiver apagado, ou parar de piscar, caso contrário.

6. Como é feita a detecção do estado do botão? Qual função é responsável por lidar com isso?

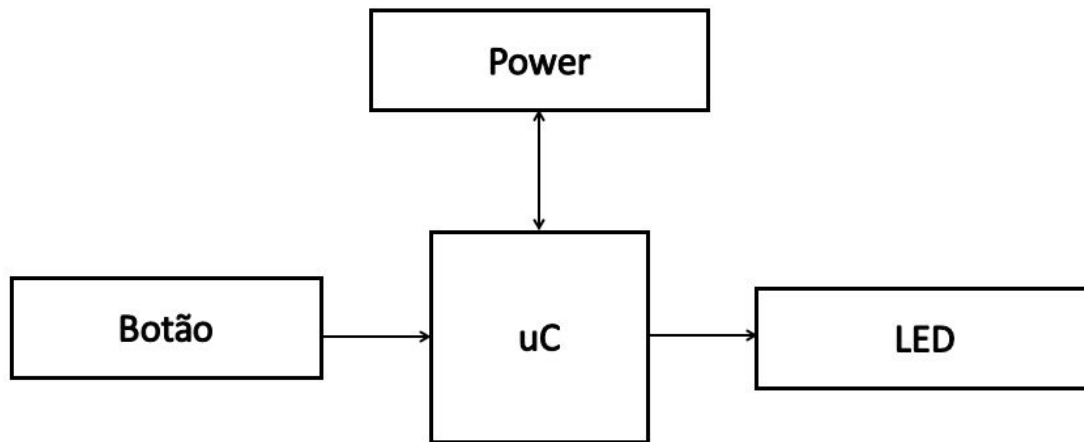
A detecção do estado do botão é feita pela `Button1_handler()`. A função que lida com isso é a `ProcessButtonEvt()`

```
static void ProcessButtonEvt(uint8_t uc_button)
{
    // [main_button1_evnt_process]
    if (uc_button == 0) {
        g_b_led0_active = !g_b_led0_active;
        if (!g_b_led0_active) {
            ioport_set_pin_level(LED0_GPIO, IOPORT_PIN_LEVEL_HIGH);
        }
    }
}
```

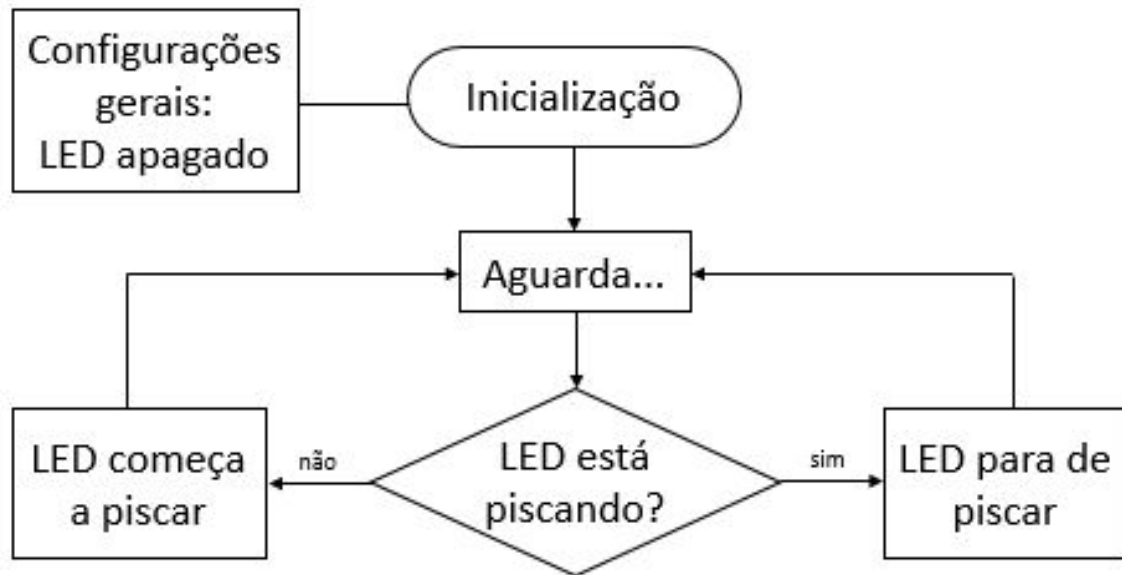
7. Faça com que a frequência do led mude toda vez que o botão foi pressionado.
8. Identifique os botões responsáveis por acionar o LED e por ler o botão, classifique os pinos como entrada e saída e identifique com função responsável por cada um.

	Pino	PIO	Entrada/Saída	Função (handler)
Botão	11	PIOA	Entrada	Button1_Handler()
LED	8	PIOC	Saída	mdelay()

9. Esboce um diagrama de blocos do projeto pisca LED.



10. Esboce um fluxograma do firmware.



11. Remova os trechos de código não utilizados, limpe também excesso de comentários e adicione comentários melhores.