



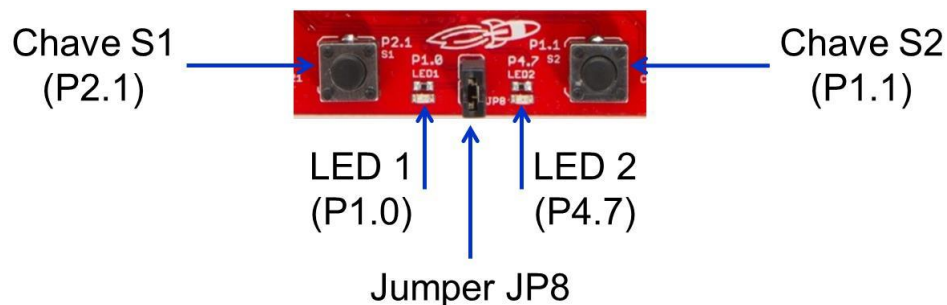
## Experimento 4. Leds e Chaves em C

### OBJETIVO:

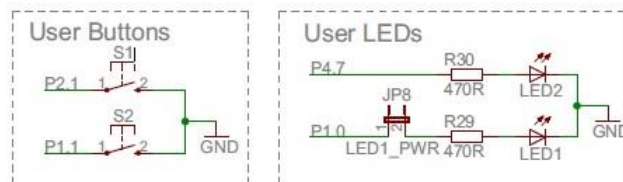
Praticar o acionamento de Leds e Chaves com programas em C. Gerar atrasos com laços de programa.

### DADOS:

A placa (Launch Pad) em uso tem duas chaves e dois leds. Veja a localização na figura abaixo.



De acordo com o esquemático abaixo, retirado do userguide da LaunchPad, os LEDs acendem quando o pino correspondente é colocado em nível alto. As chaves, quando acionadas são lidas como ZERO. Enquanto estiverem abertas, não possuem nenhum valor pré-definido. O pino do microcontrolador fica flutuando. É necessário então colocar um resistor de pull-up para que neste estado (chave aberta) a leitura do pino seja levada para um valor conhecido, ou seja 3,3V ou '1' lógico. Para isso, podemos utilizar os próprios resistores internos do MSP.



É uma boa prática de programação, colocar no início do programa as constantes que serão usadas. Por exemplo, na listagem abaixo, foi declarada a constante DELAY. Toda vez que a palavra DELAY for encontrada, o compilador a substitui pelo seu valor (1234). Note o uso da palavra reservada #define.

```
-----  
; Main loop here  
-----  
#define DELAY      1234
```



## PEDIDOS:

### **Programa 12:**

Escreva em linguagem C um programa que pisca o LED1 na frequência de 1 Hz, ou seja, 0,5 s aceso e 0,5 s apagado. Para facilitar o programa, escreva uma função, denominada **T500ms**, que consome aproximadamente 500 ms (use a constante DELAY). Com um cronômetro, conte as piscadas durante 10 segundos, para verificar se está próximo ao que se deseja. Se for o caso, faça ajustes na função. Dica: verifique o jumper JP8

### **Programa 13:**

Escreva um programa em C que faça o seguinte controle dos leds.

- Chave S1: a cada acionamento ( $A \rightarrow F$ ), inverte o estado do LED 1;
- Chave S2: a cada acionamento ( $A \rightarrow F$ ), inverte o estado do LED 2;
- Enquanto ambas as chaves estiverem acionadas (ambas fechadas), os leds piscam de forma complementar, na frequência de 1 Hz. Por complementar se entende um led aceso e o outro apagado. Quando ambas as chaves forem liberadas, os leds voltam para o estado anterior e voltam a obedecer aos controles das chaves S1 e S2.

## SUGESTÕES:

- Identificar do esquemático dos circuitos envolvidos.
- Esboçar um fluxograma para o problema.
- Testar cada pequena etapa do seu programa.
- As chaves S1 e S2 são “sujas”, isto é, não contém nenhum circuito de *debouncing*. Escreva uma subrotina para “limpar” a leitura das chaves S1 e S2, tanto na subida ( $F \rightarrow A$ ) quanto na descida ( $A \rightarrow F$ ). É recomendado escrever essa subrotina da melhor forma possível, pois ela será utilizada para todos os outros experimentos.

## RELATÓRIO

O relatório é individual, e deve ser entregue impresso (ou feito à mão). Em hipótese alguma será admitida a entrega do relatório de forma eletrônica.

### **Questão 1 (4 pontos)**

Apresente o **fluxograma** dos Programas 12 e 13, comentando as partes mais importantes.

### **Questão 2 (2 pontos)**

Os registradores do MSP se encontram em endereços específicos. Porém, a linguagem C não permite que se aloque uma variável em uma determinada posição de memória. Explique como foi feito para escrever nos registradores do MSP utilizando a linguagem C.

### **Questão 3 (2 pontos)**

A linguagem C não contém comandos para escrever nos bits de uma variável, apenas na variável inteira. Explique como podemos setar, apagar e inverter um determinado bit de uma variável, sem mexer nos demais bits.

### **Questão 4 (2 pontos)**

Explique o que significa o comando "volatile" e por que ele é importante no seu código.