projeto – Pisca-pisca remoto

# identificação

Disciplina:

* Laboratório de Sistemas Microprocessados – Turma A

Aluno:

* Estevam Galvão Albuquerque – 16/0005663

Professores:

* Ricardo Zelenovsky
* Daniel Chaves Café

# proposta inicial

O propósito do projeto é aprofundar o entendimento do protocolo de comunicação UART através da implementação de um sistema de controle de pisca-pisca de árvore de natal controlado remotamente pelo celular via conexão *Bluetooth*. Nesse sentido, os pisca-pisca, propriamente ditos, serão simulados pelos *LED* verde e vermelho da MSP430FR5594 e a configuração destes deverá poder ser realizada a qualquer momento através de envio de *bytes* pelo celular. É pretendido desenvolver 3 modos de operação. Alternado, simultâneo e pulsante.

# descrição

O funcionamento do programa consiste em controlar a atividade de um pisca-pisca de uma árvore de natal remotamente através do uso dos periféricos *timer*, *enhaced universal serial communication interface (e-USCI)* e os *LED* da *launchpad* MSP430FR5594.

Assim sendo, empregamos o *Timer\_A1* para controlar o sinal de PWM através de interrupções que ascenderão e apagarão os *LED,* criando o efeito visual de pisca-pisca na *launchpad*. O *eUSCI\_A*3, em modo *UART*, para estabelecer a comunicação com o módulo *Bluetooth* HC-05 e permitir o usuário controlar, através do envio de bytes , a configuração do *Timer\_A1* e, consequentemente, a PWM que rege o efeito de pisca-pisca. Ademais, foram elaborados os 3 modos de operação propostos inicialmente. O alternado, que consiste em ascender uma *LED* enquanto a outra está apagada e vice-versa, o simultâneo, que consiste em ascender e apagar as duas *LED* ao mesmo tempo e o pulsante, que consiste em alterar o *duty-cycle* da *PWM* durante a execução do código, gerando um efeito de pulso.

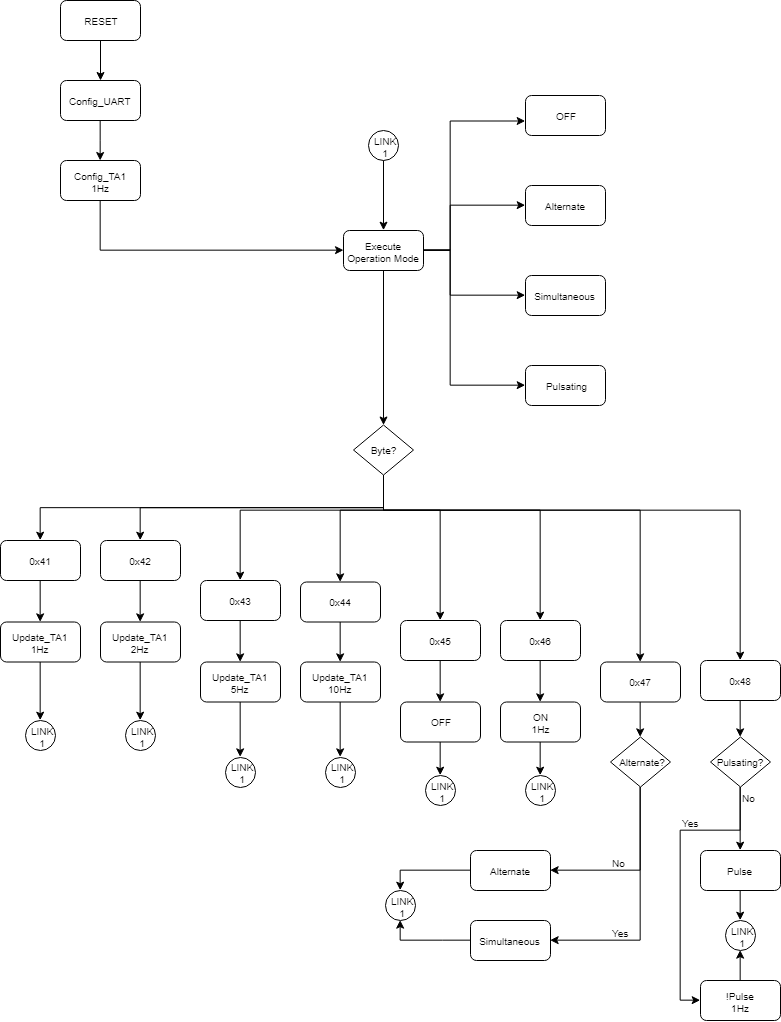
Dito isso, inicializamos o código configurando os pinos *LED* como saída e fazemos com que comecem apagados. Em seguida, configuramos a comunicação *UART* a uma taxa de transmissão de 9600 *bits* por segundo, enviando a informação do *bit* menos significativo ao mais significativo *(LSB first)*, sem *bit* de paridade, com apenas 1 *bit* de *STOP* e 8 *bits* de dados por transmissão. Além disso, também inicializamos o *Timer\_A1* a 1 *hertz* utilizando o *ACLK* e habilitamos as interrupções dos canais “0” e “1”.

Após as configurações iniciais, prendemos o código em uma trava de execução infinita para trabalharmos apenas com interrupções. Desse modo, as interrupções do *Timer\_A1, que* consistem puramente nas interrupções dos dois canais configurados, serão as responsáveis por controlar o sinal de *PWM*. O canal “0” será responsável pelo teto de contagem e por apagar as *LED*, com exceção do modo alternado, onde uma *LED* será apagada e outra será acesa. O canal “1” será responsável por coordenar o *duty-cycle* no modo pulsante e por ascender as *LED*, com exceção do modo alternado, onde uma *LED* será acesa e outra será apagada. E finalmente, as interrupções da interface de comunicação *eUSCI\_A*3 serão responsáveis por tratar os dados enviados através do módulo *Bluetooth* epermitir que o usuário controle remotamente o pisca-pisca.

Portanto, foram configurados 8 conjuntos de bytes responsivos, representados na tabela abaixo, para realizar, de fato, o controle da atividade do pisca-pisca.

|  |  |
| --- | --- |
| **0x41** | Atualiza a frequência de operação para 1 *hertz* e mantém o modo de operação atual. |
| **0x42** | Atualiza a frequência de operação para 2 *hertz* e mantém o modo de operação atual. |
| **0x43** | Atualiza a frequência de operação para 5 *hertz* e mantém o modo de operação atual. |
| **0x44** | Atualiza a frequência de operação para 10 *hertz* e mantém o modo de operação atual. |
| **0x45** | Desliga o “pisca-pisca”. |
| **0x46** | Liga o “pisca-pisca”, retornando ao modo de operação anterior e atualizando a frequência para 1 *hertz*. |
| **0x47** | Altera o modo de operação entre alternado ou simultâneo e mantém a frequência de operação atual. |
| **0x48** | Entra em modo pulsante. |

# Fluxograma:



# recursos utilizados

* Módulo Bluetooth Serial HC-05 (Mestre/Escravo).
* Launchpad MSP430FR5994
* Software Serial Bluetooth Terminal v1.25