**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**DEPARTAMENTO ACADÊMICO DE ELETRÔNICA**

**BACHARELADO EM ENGENHARIA ELETRÔNICA**

**ANDERSON KMETIUK**

**GIOVANNI DE ROSSO UNRUH**

**PONG**

**Curitiba**

**2019**

**ANDERSON KMETIUK**

**GIOVANNI DE ROSSO UNRUH**

**PONG**

Relatório presente como requisito parcial na composição de notas na disciplina EL66A - Microcontroladores do curso de Engenharia Eletrônica, ministrado pelo Departamento Acadêmico de Eletrônica do Campus Curitiba, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Guilherme Peron

Curitiba

2019

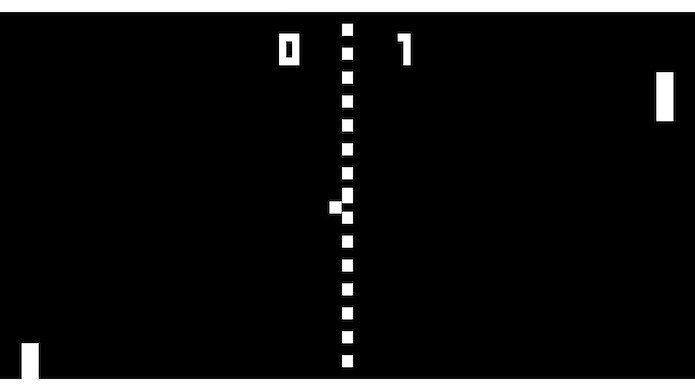
1. **INTRODUÇÃO**

A ideia inicial do projeto era desenvolver uma versão do jogo clássico PONG usando a placa estudada ao longo do semestre da Texas Instruments EK-TM4C1294XL (TIVA) e uma matriz de Leds de 16x16.

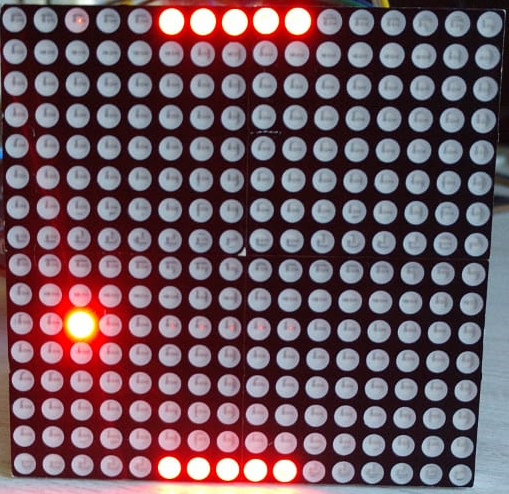
O jogo PONG original foi o primeiro [videojogo](https://pt.wikipedia.org/wiki/Jogo_eletr%C3%B4nico) lucrativo da história, dando origem a um novo setor da indústria. Foi de importância fundamental na [história dos videogames.](https://pt.wikipedia.org/wiki/Hist%C3%B3ria_do_videogame) Foi criado por [Nolan Bushnell](https://pt.wikipedia.org/wiki/Nolan_Bushnell" \o "Nolan Bushnell) e [Ted Dabney](https://pt.wikipedia.org/wiki/Ted_Dabney) na forma de um [console](https://pt.wikipedia.org/wiki/Console) ligado a um [monitor](https://pt.wikipedia.org/wiki/Monitor), movido a moedas. A primeira instalação em um bar de [San Francisco](https://pt.wikipedia.org/wiki/San_Francisco" \o "San Francisco), [Califórnia](https://pt.wikipedia.org/wiki/Calif%C3%B3rnia), mostrou aos dois a possibilidade de lucro da criação. Assim, em [27 de Junho](https://pt.wikipedia.org/wiki/27_de_Junho) de [1972](https://pt.wikipedia.org/wiki/1972_nos_jogos_eletr%C3%B4nicos), a empresa [Atari](https://pt.wikipedia.org/wiki/Atari" \o "Atari) foi fundada.

1. **DESENVOLVIMENTO**

O jogo PONG original é um [jogo eletrônico de esporte](https://pt.wikipedia.org/wiki/Jogo_eletr%C3%B4nico_de_esporte) em duas dimensões que simula um [tênis de mesa](https://pt.wikipedia.org/wiki/T%C3%AAnis_de_mesa). O jogador controla uma paleta (barra vertical) no jogo movendo-a verticalmente no lado esquerdo da tela, e compete contra o computador ou outro jogador que controlam uma segunda raquete no lado oposto. Os jogadores usam suas paletas para acertar a esfera (bola) e mandá-la para o outro lado. A paleta é dividida em oito segmentos, com o segmento central retornando a bola em um ângulo de 90º em relação a paleta e os segmentos externos retornando a bola em ângulos cada vez menores. A bola aumenta de velocidade cada vez que é rebatida, reiniciando a velocidade caso algum dos jogadores não acerte a bola. O objetivo é fazer mais pontos que seu oponente, fazendo com que o oponente não consiga retornar a bola para o outro lado.

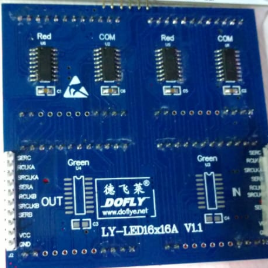


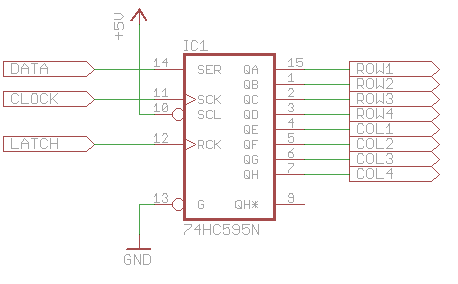
Para nossa versão do PONG fizemos algumas mudanças, ao invés de o jogador controlar de maneira vertical, optamos por fazer na horizontal, pois facilitaria a implementação na nossa matriz de Leds, também optamos por colocar uma pontuação de “Game Over” para que o jogo tivesse um final e para incrementarmos fizemos um menu com três modos, modo Treino, modo Vs AI, e modo dois jogadores, todos plenamente funcionais, apenas no modo dois jogadores que implementamos a opção onde a bola acelera com o tempo.



* 1. **Escrita na Matriz 16x16**

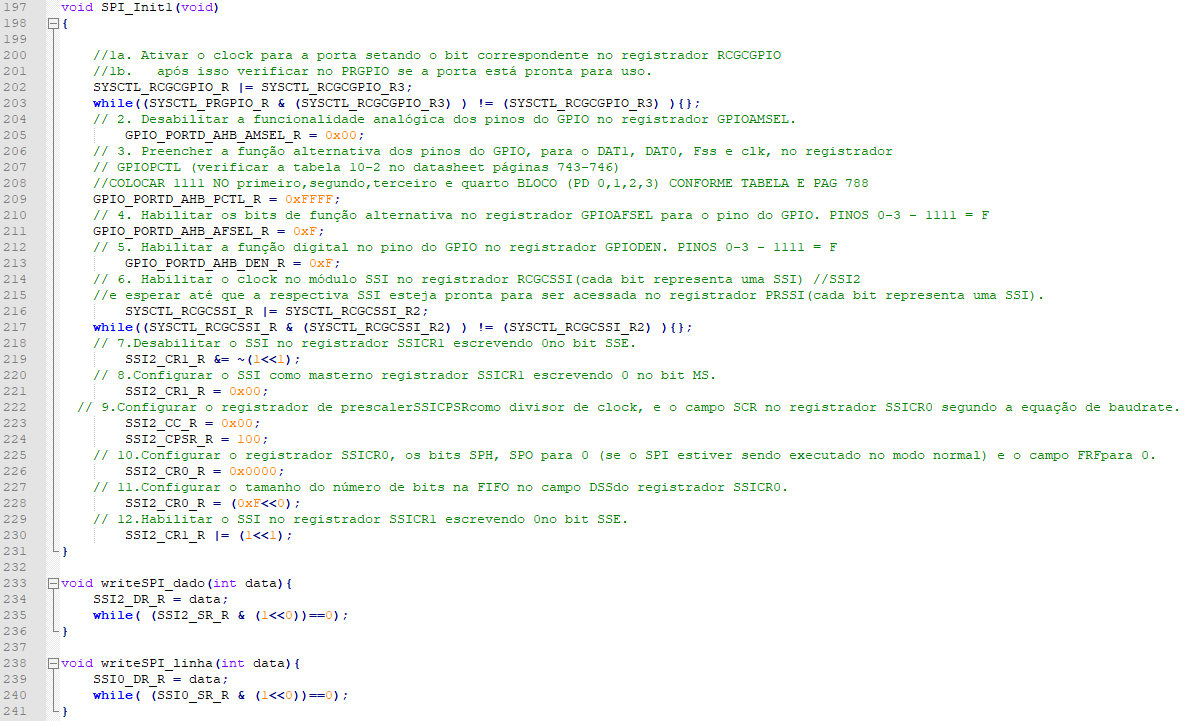
Para realizarmos o controle da matriz de Leds de 16x16 utilizamos comunicação via protocolo SPI e o Shift Register 74HC595, foram necessárias duas comunicações SPI para fazer a varredura completa da matriz, em uma enviamos os BITS que representavam as colunas e na outra os que representavam as linhas, a matriz que utilizamos já possuía integrada quatro 74HC595, assim só foi necessário descobrir o funcionamento desse componente.



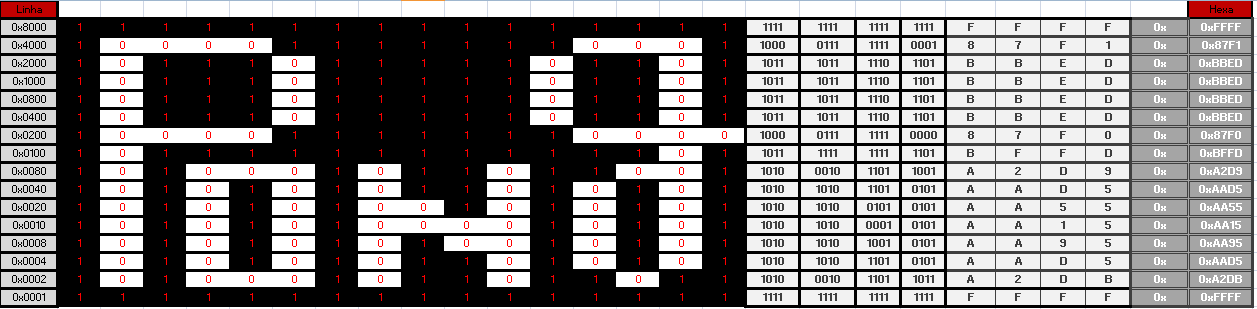


Como não seria necessária a verificação se os Bits foram recebidos de maneira adequada pela matriz nós não precisamos nos preocupar com o MISO (Master Input Slave Output), assim a conexão era usando apenas os pinos DAT0, Fss e Clk da PAT e os pinos P10 (DAT0), P8(Fss) e P6(Clk) da TIVA para utilizar as duas SPI.

Essa foi a configuração realizada para uma das comunicações SPI e as duas funções de transmissão.



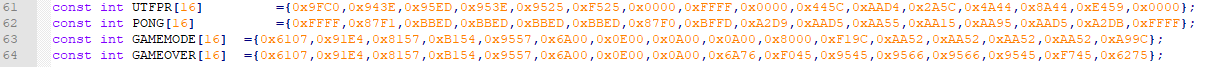
Para auxiliar na obtenção dos valores em hexadecimal que seriam enviados para a matriz optamos em fazer uma tabela do Excel que gerasse os valores mais rapidamente, assim conseguimos essa estrutura, que se encontra na pasta Documentos enviada junto com o projeto:

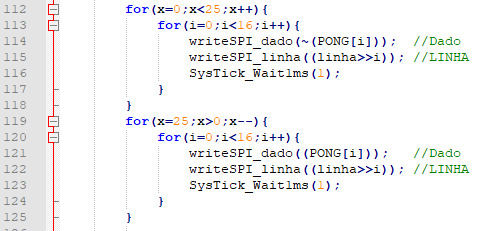


O resultado na matriz foi esse:

Os valores obtidos na tabela do Excel eram colocados em forma de vetor e para realizar a impressão na matriz bastava percorrer o vetor e enviar o valor via SPI. Para que os leds ficassem acessos de forma continua e não piscando numa velocidade que tornava desagradável a visualização fizemos que ele repetisse o envio do dado varias vezes antes de seguir para o próximo, como se fosse uma “Taxa de atualização” da matriz.



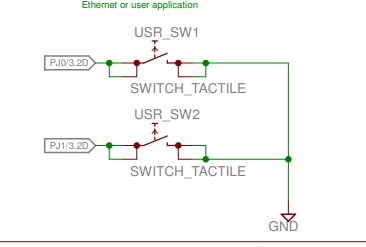


* 1. **Controles**

Inicialmente planejávamos fazer os controles do Jogador 1 pelo teclado matricial (pelos botões 1 e 3), como já era normal para o menu do jogo e todas as praticas estudadas ao longo do semestre, e o controle do Jogador 2 seria feito utilizando a UART, porem com o desenvolvimento do projeto percebemos que a UART não funcionaria da maneira que gostaríamos, pois adicionaria alguns delays ao jogo deixando o jogador 1 com vantagem, alem de que o tempo para o termino do prazo de entrega fez com que optássemos em mudar o jogador 2 para dois botões.



Utilizamos os Pinos Rx, Tx e GND da interface UART da PAT para inserir dois push bottons usando resistor de pull-up assim como que já foi feito para a o teclado matricial.



* 1. **Menu e Interrupção**

Para esse projeto fizemos um menu bem simples que usaria apenas o display LCD para exibição e o teclado matricial para controle.

Na tela inicial é solicitado que o jogador segure o botão “\*” do teclado matricial então muda para a opção de escolha do modo de jogo onde ele pode optar pelo 3 modos disponíveis do citados anteriormente, para cada um dos modos durante o andamento do jogo o display LCD mostra qual modo está sendo executado, e no final do modo aparece uma mensagem respectiva de cada modo, Fim de Treino para modo treino; Perdeu ou Ganhou para o modo Vs AI; Venceu Player 1 ou Player 2 para modo dois jogadores dependendo do vencedor; logo em seguida em todos os modos é solicitado que o jogador pressione “\*” para voltar ao menu principal.

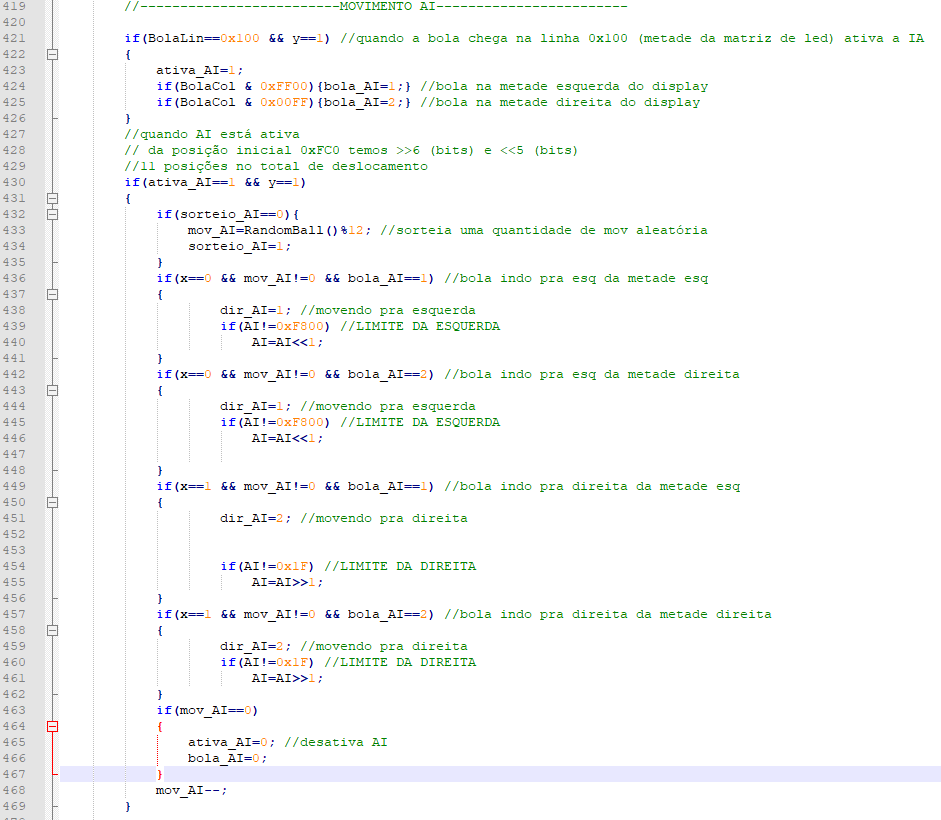
A qualquer momento do jogo o usuário pode pressionar a tecla SW1 da TIVA que será acionada uma interrupção que retorna para o Menu Principal.

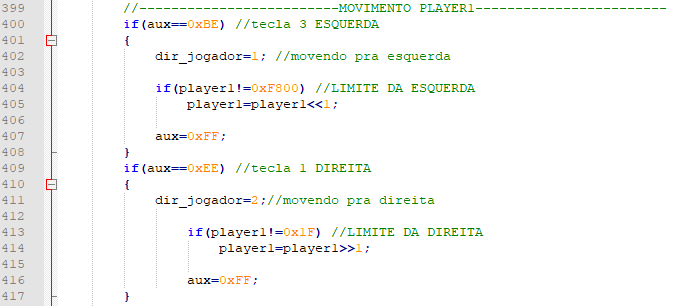


* 1. **O Jogo e a IA**

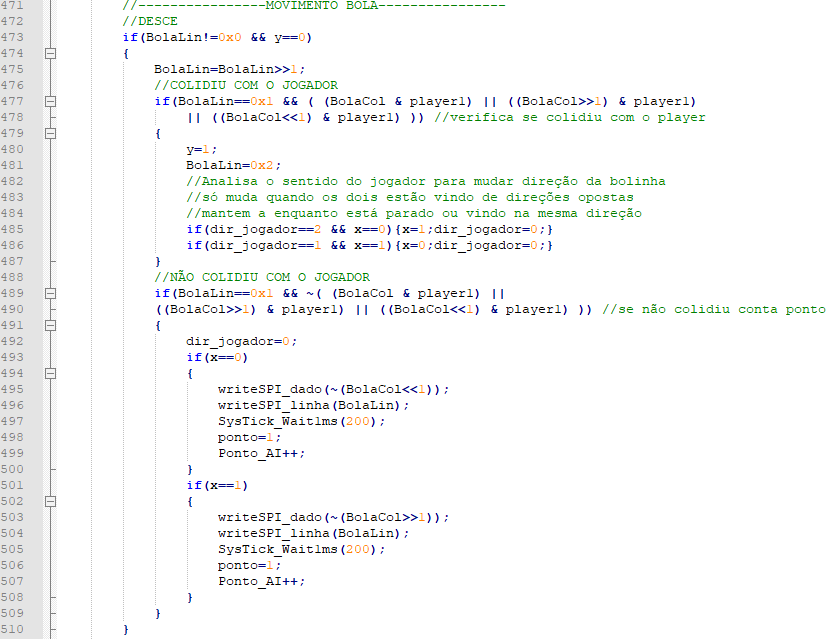
A IA funciona a partir do método de sorteio, utilizamos a função que sorteava a posição inicial da bola e geramos um número de 0 a 11 que correspondia ao número de movimentos que a IA faria dependendo de onde a bola estivesse.

Para que a IA não ficasse impossível de se vencer, fizemos com que ela só fosse ativada uma vez, quando a bola cruzava a linha que dividia a metade de cima e de baixo da matriz de leds. Para obter mais precisão, foi necessário também dividir a matriz de leds em metade esquerda e direita. Assim, ficava mais fácil de prever de onde a bola estava vindo e como a IA deveria reagir para defendê-la. 

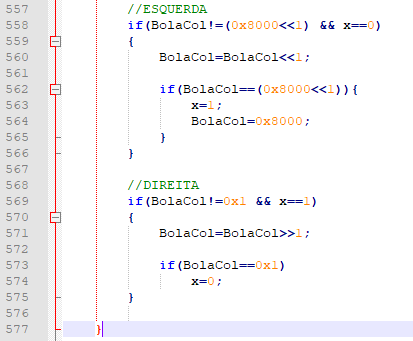
Para movimentar o jogador foi feito por meio de deslocamento de Bit, conforme o teclado é varrido e detecta as teclas 1 ou 3 precisonadas.



E para bolinha foram realizadas verificações de colisão fazendo operações lógicas AND entre o bit q representa a bolinha e os bits que representam os jogadores ou a IA ou as “Paredes” da matriz.

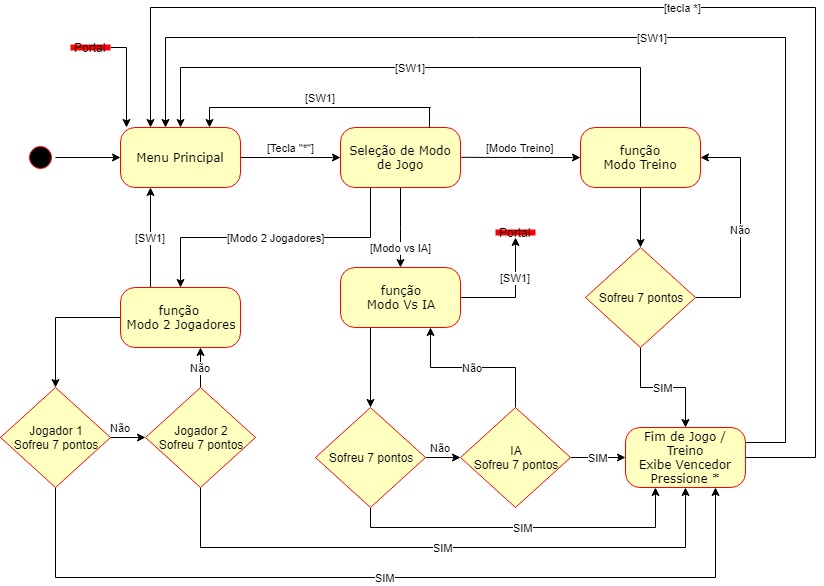


Assim como o jogador a bolinha se move com deslocamento de Bits, mudando a direção em caso de colisão.

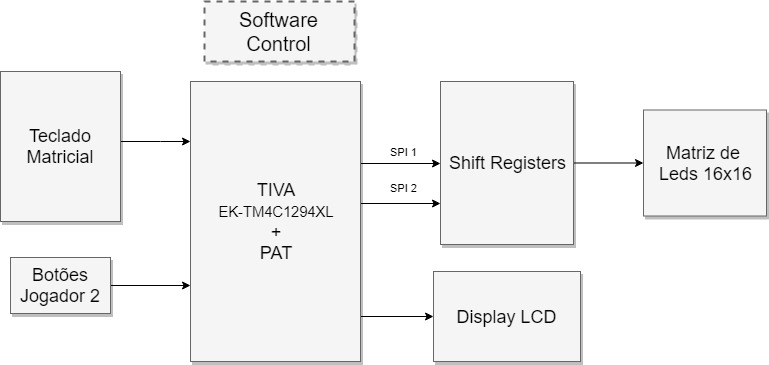


* 1. **Diagramas e Esquemático**

A maquina de estados do projeto após rever como seria o funcionamento do jogo ficou da seguinte maneira:



O nosso diagrama de blocos ficou bem simples pois não necessita de muitos componentes alem da Tiva:



E por fim o esquemático das conexões do nosso projeto, feito utilizando o software Altium Designer:

