

**Nome: Adriano Ulrich do Prado Wiedmann**

**Matrícula: 202014824**

## **Projeto Laboratório de Sistemas Microprocessados**

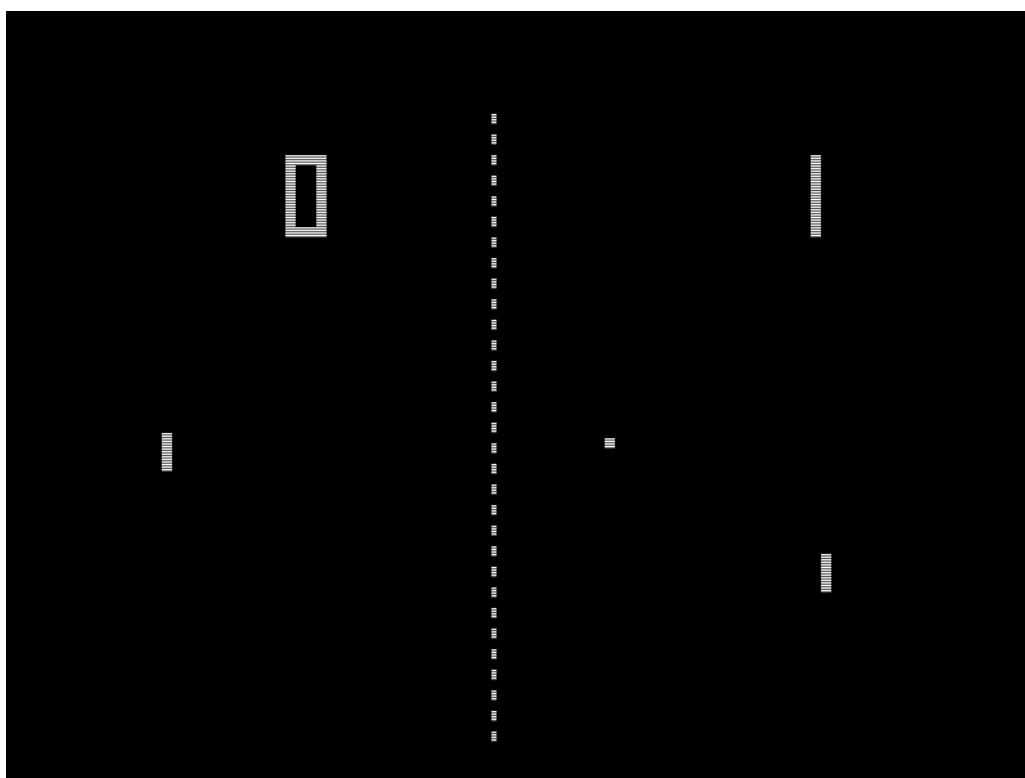


**Universidade De Brasília – Faculdade de Tecnologia (ENE)**

**Brasília, Brasil**

## INTRODUÇÃO

Ao longo do curso de **Laboratório de Sistemas Microprocessados**, foi colocado em prática vários assuntos, por exemplo, DMA, UART, ADC etc. Assuntos que auxiliaram no projeto realizado. Tal projeto visou a realização do jogo **Pong**, primeiro jogo comercialmente bem-sucedido. Além disso, é um jogo eletrônico de esporte em duas dimensões que simula um tênis de mesa, a figura 1 ilustra o jogo. O jogador controla uma paleta no jogo movendo-a verticalmente.



*Figura 1 – Jogabilidade do Pong*

É objetivo deste relatório, demonstrar a realização do projeto, a sua implementação, os dispositivos utilizados etc.

## Dispositivos Utilizados

1. **MSP430 F5529:** O principal dispositivo para a realização do curso foi a MSP430 F5529 (figura 2), o principal microcontrolador utilizado durante o semestre. Com ele podemos fazer todas as conexões possíveis com os outros dispositivos, que serão abordados;



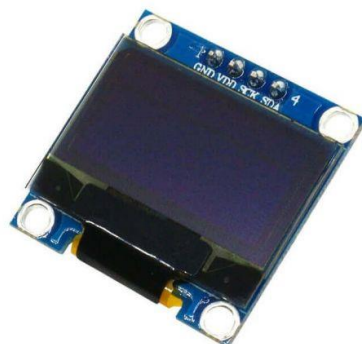
*Figura 2 – MSP430 F5529*

2. **2 Joysticks:** Para o usuário movimentar a paleta, foram utilizados 2 **joysticks**, um **joystick** para movimentar a paleta do lado esquerdo e o outro **joystick** para movimentar a paleta do lado direito, figura 3;



*Figura 3 – Joystick utilizado*

3. **Display OLED:** Para que o jogo pudesse ser visível, foi utilizado um **Display OLED** (figura 4), com isso será possível visualizar as paletas (esquerda e direita), a delimitação do campo e outras informações.



*Figura 4 – Display OLED*

## **Conexões dos Dispositivos**

1. **Joystick:** Para que pudesse verificar a movimentação do joystick, com a utilização do ADC, foi ligado o **VRx** no pino P6.1. Mais detalhes.

**GND -> GND**

**+5v -> 3v3**

**VRx -> P6.1**

**VRy -> P6.2**

2. **Outro Joystick:** Para que pudesse verificar a movimentação do outro joystick, com a utilização do ADC, foi ligado o **VRx** no pino P6.3. Mais detalhes.

**GND -> GND**

**+5v -> 3v3**

**VRx -> P6.3**

**VRy -> P6.4**

3. **Display OLED:** Para a configuração e exibição dos dados no display, foram feitas as devidas conexões.

**GND -> GND**

**VCC -> 5V**

**SCL -> P4.2**

**SDA -> P4.1**

## Programa

Para que pudesse fazer a delimitação do campo, foi utilizado a função ***oled\_linha***, localizado no arquivo Oled.c. Essa função foi utilizada no arquivo principal, Pong.c:

```
70 oled_linha(0,64,125,64,PX_ON); //LINHA HORIZONTAL SUPERIOR
71 oled_linha(0,127,125,127,PX_ON); //LINHA HORIZONTAL INFERIOR
72 oled_linha(125,0,125,125,PX_ON); //LINHA VERTICAL LADO DIREITO
73 oled_linha(0,0,0,125,PX_ON); //LINHA VERTICAL LADO ESQUERDO
74
```

*Figura 5 – delimitação do campo*

Para iniciar o jogo é necessário segurar a chave **SW1** do MSP430. Para isso foi utilizado a função ***sw1\_estado*** que verifica o estado da chave **SW1**. As verificações foram feitas através do laço condicional ***while***, enquanto a chave não é pressionada, o jogo não inicia.

```
101 while(sw1_estado == ABERTA){ //Esperar acionar SW1
102     //oled_frase("PONG",55,8,PX_ON);
103     oled_frase_delay10ms("PONG",55,8,PX_ON,10000);
104     oled_frase_delay10ms("PONG",55,8,PX_OFF,10000);
105     oled_frase("HOLD SW1 TO PLAY",31,33,PX_ON);
106     espera_10ms(80);
107     oled_frase("HOLD SW1 TO PLAY",31,33,PX_OFF);
108     espera_10ms(40);
109 }
```

*Figura 6 – Laço que verifica se a chave SW1 foi pressionada.*

Após isso o programa entra em um laço infinito, agora serão citadas partes importantes desse loop.

A partir de uma matriz 3x3, foram utilizados dois vetores, uma para a linha (**bola\_lin**) e outra para a coluna (**bola\_col**). Com isso é possível realizar a movimentação da bola, figura 7.

```
162         //BOLA
163         //espera_10ms(5);
164         if(direcao%2==0){
165             apagar_bola();
166             bola_lin[0]=bola_lin[0]+1;
167             bola_lin[1]=bola_lin[1]+1;
168             bola_lin[2]=bola_lin[2]+1;
169             if(lado%2!=0){
170                 bola_col[0]=bola_col[0]+1;
171                 bola_col[1]=bola_col[1]+1;
172                 bola_col[2]=bola_col[2]+1;
173             }
174             else{
175                 bola_col[0]=bola_col[0]-1;
176                 bola_col[1]=bola_col[1]-1;
177                 bola_col[2]=bola_col[2]-1;
178             }
179             mover_bola();
180         }
181         else{
182             apagar_bola();
183             bola_lin[0]=bola_lin[0]-1;
184             bola_lin[1]=bola_lin[1]-1;
185             bola_lin[2]=bola_lin[2]-1;
186             if(lado%2!=0){
187                 bola_col[0]=bola_col[0]+1;
188                 bola_col[1]=bola_col[1]+1;
189                 bola_col[2]=bola_col[2]+1;
190             }
191             else{
192                 bola_col[0]=bola_col[0]-1;
193                 bola_col[1]=bola_col[1]-1;
194                 bola_col[2]=bola_col[2]-1;
195             }
196             mover_bola();
197         }
```

*Figura 7 – Movimentação da bola.*

Caso algum jogador tenha feito ponto, o jogo então reinicia quando a bola volta ao meio do campo. Para isso foi feito o seguinte trecho de código, onde verifica se a bola chegou no limite do lado esquerdo ou do direito. E caso tenha chegado no limite, o jogo é reiniciado e a bola vai ao meio de campo.

```
126     if(bola_col[1]==2){
127         random++;
128         visitante++;
129         apagar_bola();
130         bola_lin[0]=94;
131         bola_lin[1]=95;
132         bola_lin[2]=96;
133         bola_col[0]=63;
134         bola_col[1]=64;
135         bola_col[2]=65;
136         mover_bola();
137         direcao = rrand_fx(random);
138         lado = rrand_fx(random);
139     }
140     else if(bola_col[1]==123){
141         random++;
142         casa++;
143         apagar_bola();
144         bola_lin[0]=94;
145         bola_lin[1]=95;
146         bola_lin[2]=96;
147         bola_col[0]=63;
148         bola_col[1]=64;
149         bola_col[2]=65;
150         mover_bola();
151         direcao = rrand_fx(random);
152         lado = rrand_fx(random);
153     }
```

*Figura 8 – Bola reiniciada.*



Caso o valor da variável **move** seja 0, significa que a paleta vai se mover para cima, caso seja igual a 2, vai mover para baixo. E caso seja igual a 1, a paleta permanece parada.

Variável **move** é específico para a paleta da esquerda, já o **move2** para a paleta da direita.

```
if(move==0){//mover para baixo

oled_linha(5,linha,5,linha,PX_OFF);    //Raquete lado esquerdo
oled_linha(5,linha+1,5,linha+1,PX_OFF);//Raquete lado esquerdo
oled_linha(5,linha+2,5,linha+2,PX_OFF);//Raquete lado esquerdo
oled_linha(5,linha+3,5,linha+3,PX_OFF);//Raquete lado esquerdo
oled_linha(5,linha+4,5,linha+4,PX_OFF);//Raquete lado esquerdo
oled_linha(5,linha+5,5,linha+5,PX_OFF);//Raquete lado esquerdo
oled_linha(5,linha+6,5,linha+6,PX_OFF);//Raquete lado esquerdo
oled_linha(5,linha+7,5,linha+7,PX_OFF);//Raquete lado esquerdo

linha=linha-7;

if(linha<=64){
    dif=64-linha;
    linha=linha+(1+dif);
    //linha=linha-8;
}

oled_linha(5,linha,5,linha,PX_ON);    //Raquete lado esquerdo
oled_linha(5,linha+1,5,linha+1,PX_ON);//Raquete lado esquerdo
oled_linha(5,linha+2,5,linha+2,PX_ON);//Raquete lado esquerdo
oled_linha(5,linha+3,5,linha+3,PX_ON);//Raquete lado esquerdo
oled_linha(5,linha+4,5,linha+4,PX_ON);//Raquete lado esquerdo
oled_linha(5,linha+5,5,linha+5,PX_ON);//Raquete lado esquerdo
oled_linha(5,linha+6,5,linha+6,PX_ON);//Raquete lado esquerdo
oled_linha(5,linha+7,5,linha+7,PX_ON);//Raquete lado esquerdo
raquete_touch();
//espera_10ms(25);
```

*Figura 9 – Movimentação da paleta.*

Várias funções parecidas foram utilizadas para verificar se a bola tocou na paleta, caso tenha tocado, a movimentação muda, por exemplo, a bolinha ia em direção ao canto esquerdo, quando tocou na paleta, começou a ir em direção ao canto direito.

```
493 void raquete_touch(){
494     if((bola_lin[0] == linha-2) ||
495         (bola_lin[0] == linha-1) ||
496         (bola_lin[0] == linha ) ||
497         (bola_lin[0] == linha+1) ||
498         (bola_lin[0] == linha+2) ||
499         (bola_lin[0] == linha+3) ||
500         (bola_lin[0] == linha+4) ||
501         (bola_lin[0] == linha+5) ||
502         (bola_lin[0] == linha+6) ||
503         (bola_lin[0] == linha+7)
504     ){
505         if(bola_col[0] <= 9){
506             lado=3;
507         }
508         else if(bola_col[2] == 9){
509             lado=3;
510         }
511     }
512 }
```

*Figura 10 – Verificação se a bola tocou na paleta.*

Caso a bolinha tenha tocado nas delimitações do campo, na linha horizontal superior ou inferior a bola muda de direção, por exemplo, bola estar descendo, toca na linha horizontal inferior, a bola começa a subir.

```
155     if(bola_lin[2] == 126){
156         direcao=1;
157     }
158     else if(bola_lin[0] == 65){
159         direcao=0;
160     }
```

*Figura 11 – Defini direção da bola.*

Há uma função que verifica quem ganhou o jogo, **Player 1** (paleta da esquerda) ou **Player 2** (paleta da direita). Ganha o jogo, o jogador que fizer 3 pontos primeiro. E no fim mostra o resultado no Display.

```
380 void gerar_placar(){
381     if(casa==0){
382         oled_frase("0",54,74,PX_ON);
383         oled_frase("X",64,74,PX_ON);
384     }
385     else if(casa==1){
386         oled_frase("1",54,74,PX_ON);
387         oled_frase("X",64,74,PX_ON);
388     }
389     else if(casa==2){
390         oled_frase("2",54,74,PX_ON);
391         oled_frase("X",64,74,PX_ON);
392     }
393     else{
394         oled_frase("3",54,74,PX_ON);
395         oled_frase("X",64,74,PX_ON);
396         oled_frase("PLAYER 1 WINS!",36,94,PX_ON);
397     }
398
399     if(visitante==0){
400         oled_frase("0",74,74,PX_ON);
401     }
402     else if(visitante==1){
403         oled_frase("1",74,74,PX_ON);
404     }
405     else if(visitante==2){
406         oled_frase("2",74,74,PX_ON);
407     }
408     else{
409         oled_frase("3",74,74,PX_ON);
410         oled_frase("PLAYER 2 WINS!",36,94,PX_ON);
411     }
412
413     while(sw1_estado == ABERTA){ //Esperar acionar SW1
414         oled_frase("*HOLD SW1 TO PLAY",28,104,PX_ON);
415         espera_10ms(80);
416         oled_frase("*HOLD SW1 TO PLAY",28,104,PX_OFF);
417         espera_10ms(40);
418     }
419 }
```

*Figura 12 – Verifica e mostra o placar.*

De forma resumida, o programa inicia mostrando no Display o nome do jogo, **Pong**. E solicita ao usuário que pressione SW1 para que o jogo se inicie.

Enquanto o nenhum dos jogadores tiverem 3 pontos, o jogo continuará. Caso contrário, é mostrado no display o resultado do jogo, e solicita ao usuário, novamente, para que pressione **SW1** para começar uma nova partida.

## Conclusão

Ao decorrer da criação do código, foi adquirido vários conhecimentos, principalmente por estar utilizando vários conhecimentos abordados no curso teórico e também por utilizar um dispositivo nunca antes utilizado.

Por outro lado, o jogo **Pong** foi bem efetuado, faltando algumas implementações e outras correções. Entretanto, o jogo se assemelha ao jogo original, podendo assim então, se entreter de forma rápida com o joguinho de mesa virtual.