

**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

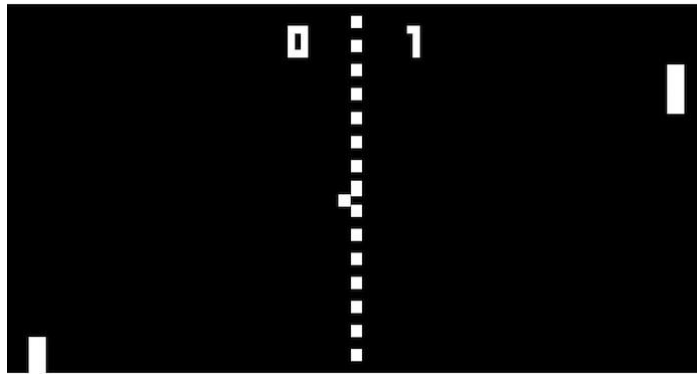
## **1. INTRODUÇÃO**

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.

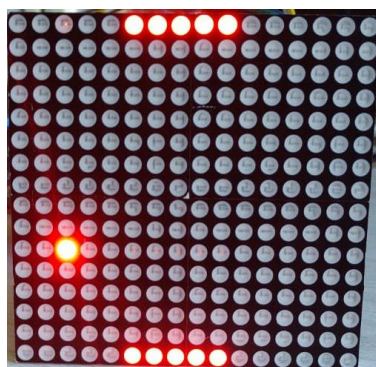
O jogo PONG original foi o primeiro videogame lucrativo da história, dando origem a um novo setor da indústria. Foi de importância fundamental na história dos videogames. Foi criado por Nolan Bushnell e Ted Dabney na forma de um console ligado a um monitor, movido a moedas. A primeira instalação em um bar de San Francisco, Califórnia, mostrou aos dois a possibilidade de lucro da criação. Assim, em 27 de Junho de 1972, a empresa Atari foi fundada.

## 2. DESENVOLVIMENTO

O jogo PONG original é um jogo eletrônico de esporte em duas dimensões que simula um tênis 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 90° 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.

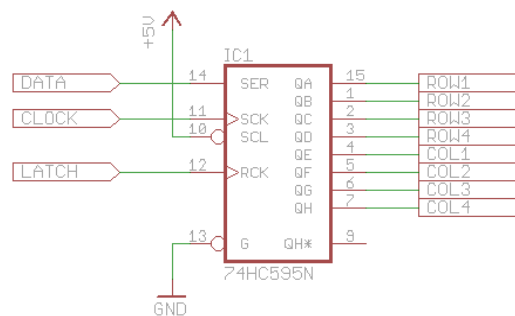


Para nossa versão do PONG fizemos algumas mudanças, ao invés de o jogador controlar de maneira vertical, optamos por fazer na horizontal, pois facilitaria a implementação na nossa matriz de Leds, também optamos por colocar uma pontuação de “Game Over” para que o jogo tivesse um final e para incrementarmos 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.



## 2.1. 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.

```

197 void SPI_Init1(void)
198 {
199
200     //1a. Ativar o clock para a porta setando o bit correspondente no registrador RCGCGPIO
201     //1b. após isso verificar no PRGPIO se a porta está pronta para uso.
202     SYSCCTL_RCGCGPIO_R |= SYSCCTL_RCGCGPIO_R3;
203     while((SYSCCTL_PRGPIO_R & (SYSCCTL_RCGCGPIO_R3)) != (SYSCCTL_RCGCGPIO_R3)){};
204     // 2. Desabilitar a Funcionalidade analógica dos pinos do GPIO no registrador GPIOAMSEL.
205     GPIO_PORTD_AHB_AMSEL_R = 0x00;
206     // 3. Preencher a função alternativa dos pinos do GPIO, para o DAT1, DAT0, Fss e clk, no registrador
207     // GPIOOCTL (verificar a tabela 10-2 no datasheet páginas 743-746)
208     //COLOCAR 1111 NO primeiro,segundo,terceiro e quarto BLOCO (PD 0,1,2,3) CONFORME TABELA E PAG 788
209     GPIO_PORTD_AHB_PCTL_R = 0xFFFF;
210     // 4. Habilitar os bits de função alternativa no registrador GPIOAFSEL para o pino do GPIO. PINOS 0-3 - 1111 = F
211     GPIO_PORTD_AHB_AFSEL_R = 0xF;
212     // 5. Habilitar a função digital no pino do GPIO no registrador GPIODEN. PINOS 0-3 - 1111 = F
213     GPIO_PORTD_AHB_DEN_R = 0xF;
214     // 6. Habilitar o Clock no módulo SSI no registrador RCGCSSI(cada bit representa uma SSI) //SSI2
215     //e esperar até que a respectiva SSI esteja pronta para ser acessada no registrador PRSSI(cada bit representa uma SSI).
216     SYSCCTL_RCGCSSI_R |= SYSCCTL_RCGCSSI_R2;
217     while((SYSCCTL_RCGCSSI_R & (SYSCCTL_RCGCSSI_R2)) != (SYSCCTL_RCGCSSI_R2)){};
218     // 7.Desabilitar o SSI no registrador SSICR1 escrevendo 0no bit SSE.
219     SSI2_CR1_R &= ~(1<<1);
220     // 8.Configurar o SSI como master no registrador SSICR1 escrevendo 0 no bit MS.
221     SSI2_CR1_R = 0x00;
222     // 9.Configurar o registrador de prescalerSSICPSR como divisor de clock, e o campo SCR no registrador SSICR0 segundo a equação de baudrate.
223     SSI2_CR0_R = 0x00;
224     SSI2_CPSR_R = 100;
225     // 10.Configurar o registrador SSICR0, os bits SPH, SPO para 0 (se o SPI estiver sendo executado no modo normal) e o campo FRF para 0.
226     SSI2_CR0_R = 0x0000;
227     // 11.Configurar o tamanho do número de bits na FIFO no campo DSS do registrador SSICR0.
228     SSI2_CR0_R = (0xF<<0);
229     // 12.Habilitar o SSI no registrador SSICR1 escrevendo 0no bit SSE.
230     SSI2_CR1_R |= (1<<1);
231 }
232
233 void writeSPI_dado(int data){
234     SSI2_DR_R = data;
235     while((SSI2_SR_R & (1<<0))==0);
236 }
237
238 void writeSPI_linha(int data){
239     SSI0_DR_R = data;
240     while((SSI0_SR_R & (1<<0))==0);
241 }

```

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:

Endereço	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497
----------	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

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.

```

61 const int UTFPR[16]      = {0x9FC0,0x943E,0x95ED,0x953E,0x9525,0xF525,0x0000,0xFFFF,0x0000,0x445C,0xAAD4,0x2A5C,0x4A44,0x8A44,0xE459,0x0000};
62 const int PONG[16]      = {0xFFFF,0x87F1,0xBBED,0xBBED,0xBBED,0x87F0,0xBFFD,0xA2D9,0xAAD5,0x5555,0x5555,0x5555,0x5555,0x5555,0x5555,0x5555};
63 const int GAMEMODE[16]  = {0x6107,0x91E4,0x8157,0xB154,0x9557,0x6A00,0x0E00,0x0A00,0x0A00,0x8000,0xF19C,0x5555,0x5555,0x5555,0x5555,0x5555};
64 const int GAMEOVER[16]  = {0x6107,0x91E4,0x8157,0xB154,0x9557,0x6A00,0x0E00,0x0A00,0x6A76,0xF045,0x9545,0x9566,0x9566,0x9545,0xF745,0x6275};

```

```

112
113
114
115
116
117
118
119
120
121
122
123
124
125

```

```

for(x=0;x<25;x++){
    for(i=0;i<16;i++){
        writeSPI_dado(~(PONG[i])); //Dado
        writeSPI_linha((linha>>i)); //LINHA
        SysTick_Wait1ms(1);
    }
}
for(x=25;x>0;x--){
    for(i=0;i<16;i++){
        writeSPI_dado((PONG[i])); //Dado
        writeSPI_linha((linha>>i)); //LINHA
        SysTick_Wait1ms(1);
    }
}

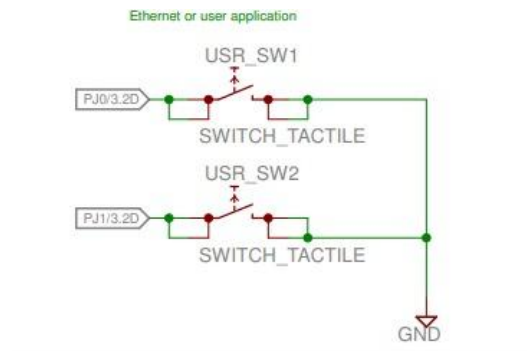
```

## 2.2. 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 buttons usando resistor de pull-up assim como que já foi feito para a o teclado matricial.

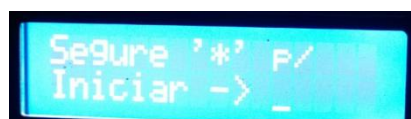


### 2.3. Menu e Interrupção

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.





## 2.4. 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.

```
419 //-----MOVIMENTO AI-----
420
421 if(BolaLin==0x100 && y==1) //quando a bola chega na linha 0x100 (metade da matriz de led) ativa a IA
422 {
423     ativa_AI=1;
424     if(BolaCol & 0xFF00){bola_AI=1;} //bola na metade esquerda do display
425     if(BolaCol & 0x00FF){bola_AI=2;} //bola na metade direita do display
426 }
427 //quando AI está ativa
428 // da posição inicial 0xF00 temos >>6 (bits) e <<5 (bits)
429 //11 posições no total de deslocamento
430 if(ativa_AI==1 && y==1)
431 {
432     if(sorteio_AI==0){
433         mov_AI=RandomBall()%12; //sorteia uma quantidade de mov aleatória
434         sorteio_AI=1;
435     }
436     if(x==0 && mov_AI!=0 && bola_AI==1) //bola indo pra esq da metade esq
437     {
438         dir_AI=1; //movendo pra esquerda
439         if(AI!=0xF800) //LIMITE DA ESQUERDA
440             AI=AI<<1;
441     }
442     if(x==0 && mov_AI!=0 && bola_AI==2) //bola indo pra esq da metade direita
443     {
444         dir_AI=1; //movendo pra esquerda
445         if(AI!=0xF800) //LIMITE DA ESQUERDA
446             AI=AI<<1;
447     }
448 }
449 if(x==1 && mov_AI!=0 && bola_AI==1) //bola indo pra direita da metade esq
450 {
451     dir_AI=2; //movendo pra direita
452 }
453 if(AI!=0x1F) //LIMITE DA DIREITA
454     AI=AI>>1;
455 }
456 if(x==1 && mov_AI!=0 && bola_AI==2) //bola indo pra direita da metade direita
457 {
458     dir_AI=2; //movendo pra direita
459     if(AI!=0x1F) //LIMITE DA DIREITA
460         AI=AI>>1;
461 }
462 if(mov_AI==0)
463 {
464     ativa_AI=0; //desativa AI
465     bola_AI=0;
466 }
467 mov_AI--;
468 }
469 }
```

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.

```

399 //-----MOVIMENTO PLAYER1-----
400 if(aux==0xBE) //tecla 3 ESQUERDA
401 {
402     dir_jogador=1; //movendo pra esquerda
403
404     if(player1!=0xF800) //LIMITE DA ESQUERDA
405         player1=player1<<1;
406
407     aux=0xFF;
408 }
409 if(aux==0xEE) //tecla 1 DIREITA
410 {
411     dir_jogador=2; //movendo pra direita
412
413     if(player1!=0x1F) //LIMITE DA DIREITA
414         player1=player1>>1;
415
416     aux=0xFF;
417 }

```

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.

```

471 //-----MOVIMENTO BOLA-----
472 //DESCE
473 if(BolaLin!=0x0 && y==0)
474 {
475     BolaLin=BolaLin>>1;
476     //COLIDIU COM O JOGADOR
477     if(BolaLin==0x1 && ((BolaCol & player1) || ((BolaCol>>1) & player1)
478         || ((BolaCol<<1) & player1) )) //verifica se colidiu com o player
479     {
480         y=1;
481         BolaLin=0x2;
482         //Analisa o sentido do jogador para mudar direção da bolinha
483         //só muda quando os dois estão vindo de direções opostas
484         //mantem a enquanto está parado ou vindo na mesma direção
485         if(dir_jogador==2 && x==0){x=1;dir_jogador=0;}
486         if(dir_jogador==1 && x==1){x=0;dir_jogador=0;}
487     }
488     //NÃO COLIDIU COM O JOGADOR
489     if(BolaLin==0x1 && ~( (BolaCol & player1) ||
490         ((BolaCol>>1) & player1) || ((BolaCol<<1) & player1) )) //se não colidiu conta ponto
491     {
492         dir_jogador=0;
493         if(x==0)
494         {
495             writeSPI_dado(~(BolaCol<<1));
496             writeSPI_linha(BolaLin);
497             SysTick_Waitlms(200);
498             ponto=1;
499             Ponto_AI++;
500         }
501         if(x==1)
502         {
503             writeSPI_dado(~(BolaCol>>1));
504             writeSPI_linha(BolaLin);
505             SysTick_Waitlms(200);
506             ponto=1;
507             Ponto_AI++;
508         }
509     }
510 }

```

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

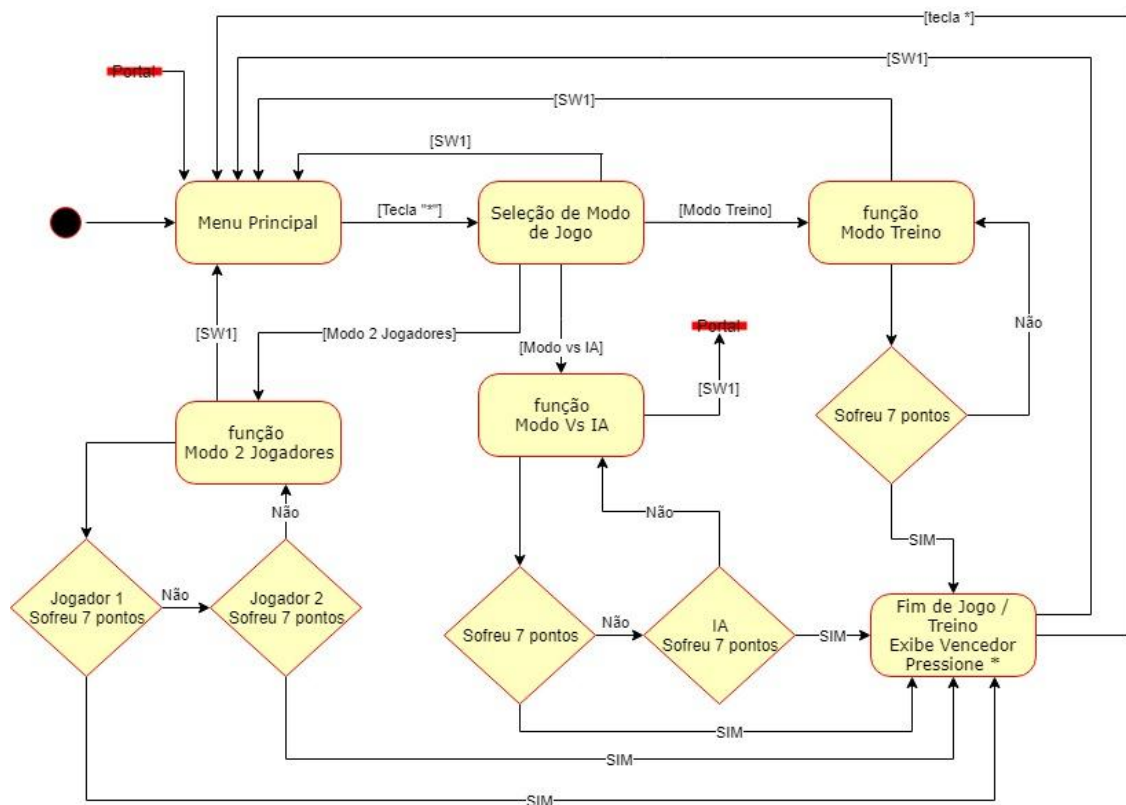
```

557
558 //ESQUERDA
559 if(BolaCol!=(0x8000<<1) && x==0)
560 {
561     BolaCol=BolaCol<<1;
562     if(BolaCol==(0x8000<<1)) {
563         x=1;
564         BolaCol=0x8000;
565     }
566 }
567
568 //DIREITA
569 if(BolaCol!=0x1 && x==1)
570 {
571     BolaCol=BolaCol>>1;
572     if(BolaCol==0x1)
573         x=0;
574 }
575
576
577

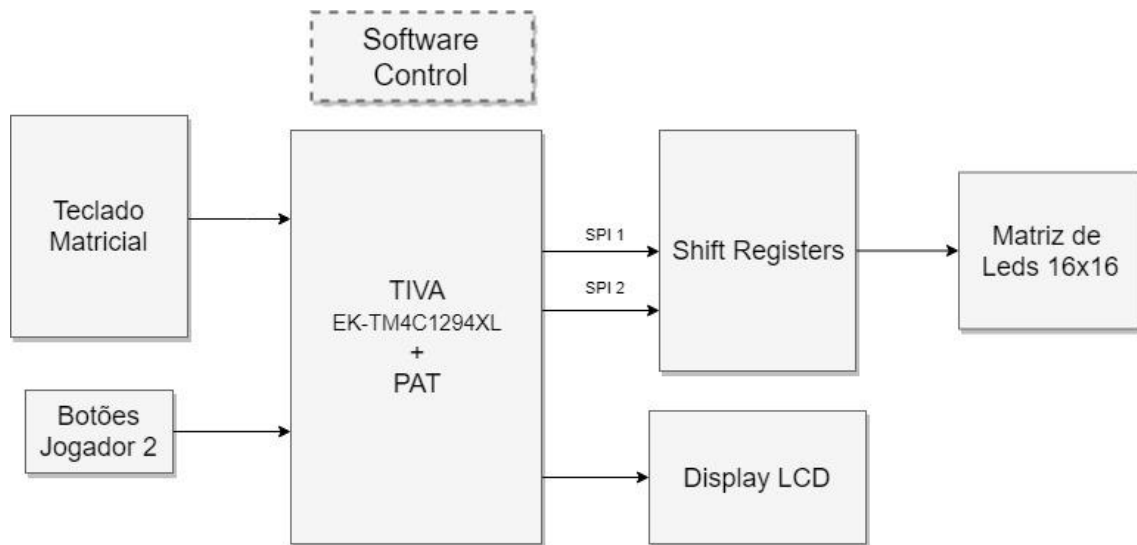
```

## 2.5. 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:

