

```
1: sbit LCD_RS at RD2_bit;
2: sbit LCD_EN at RD3_bit;
3: sbit LCD_D4 at RD4_bit;
4: sbit LCD_D5 at RD5_bit;
5: sbit LCD_D6 at RD6_bit;
6: sbit LCD_D7 at RD7_bit;
7:
8: sbit LCD_RS_Direction at TRISD2_bit;
9: sbit LCD_EN_Direction at TRISD3_bit;
10: sbit LCD_D4_Direction at TRISD4_bit;
11: sbit LCD_D5_Direction at TRISD5_bit;
12: sbit LCD_D6_Direction at TRISD6_bit;
13: sbit LCD_D7_Direction at TRISD7_bit;
14:
15: void LeitorRFID();
16: void LeituraEEPROM();
17: void statusTAG();
18: void Candidatos();
19: void gravarVOTO();
20: void armazenamento();
21: void gravarTAG();
22: void ContarVOTO();
23: void mostrarVOTO();
24: void menu();
25: void botao();
26: void zerarURNA();
27: void clearLCD ();
28: void checarEEPROM ();
29:
30: int etapa=0,v,play=0;
31: char i,confirma;
32: char rfid[7];
33: int votou,escolheu=0,local;
34: int aa,b,k,e,nn,j,o,u,hh,cc;
35: int votos;
36: int can1,can2,can3,can4,can5,can6,can7,can8,can9;
37: char txt1[4],txt2[4],txt3[4],txt4[4],txt5[4],txt6[4],txt7[4],txt8[4],txt9[4];
38:
39: void botao()
40: {
41:     for(v=0;v<1;){
42:         if (RA4_bit==0) {
43:             v = 2;          // V=2 Botão SETA pressionado
44:         }
45:         if (RA5_bit==0) {
46:             v = 3;          // V=3 Botão ENTER pressionado
47:         }
48:     }
49: }
50:
51: void clearLCD ()
52: {
53:     Lcd_Cmd(_LCD_CURSOR_OFF);
54:     Lcd_Cmd(_LCD_CLEAR);
55: }
56:
57: void menu()
58: {
59:     b=0;          // variável de seleção do menu
60:     play=0;
61:     Lcd_Out(1,1,"          MENU          ");
62:     Lcd_Out(2,2,"Exibir relatorio ");
```

```
63:     Lcd_Out(3,2,"Continuar votacao");
64:     Lcd_Out(4,2,"Inicar nova votacao");
65:     Lcd_Cmd(_LCD_BLINK_CURSOR_ON);
66:
67:     while (play==0) {
68:
69:         if (b==0) { // OPÇÃO 1 >> Exibir relatório
70:             Lcd_Cmd(_LCD_SECOND_ROW);
71:             botao();
72:             if (v==2) { // SETA pressionado encaminha proxima opção
73:                 b++;
74:             }
75:             else {
76:                 clearLCD ();
77:                 ContarVOTO();
78:                 mostrarVOTO();
79:                 play=1; //Encerra o while e entra novamente no menu
80:             }
81:         }
82:         if (b==1) { // OPÇÃO 2 >> Continuar votação
83:             Lcd_Cmd(_LCD_THIRD_ROW);
84:             botao();
85:             if (v==2) {
86:                 b++;
87:             }
88:             else {
89:                 clearLCD ();
90:                 play=2; // Encerra o while e prossegue com a votação
91:             }
92:         }
93:         if (b==2) { // OPÇÃO 3 >> Inicar nova votação
94:             Lcd_Cmd(_LCD_FOURTH_ROW);
95:             botao();
96:             if (v==2) {
97:                 b=0;
98:             }
99:             else {
100:                 clearLCD ();
101:                 zerarURNA();
102:             }
103:         }
104:     }
105: }
106:
107:
108: void main()
109: {
110:     int take,e,f,g,r,stop,receber,a,b,k;
111:     int resultado;
112:     Lcd_Init();
113:     clearLCD ();
114:
115:     UART1_Init(9600);
116:     rfid[6] = '\0';
117:     I2C1_Init(100000);
118:     ADCON1=0x0F;
119:
120:     while(1)
121:     {
122:         menu();
123:         while(play==2)
124:         {
```

```
125:      LeitorRFID();      // rotina de leitura da tag do cartão
126:      checarEEPROM ();
127:      if (etapa == 1) {
128:          Lcd_Cmd( _LCD_CLEAR);
129:          LeituraEEPROM();
130:          statusTAG();
131:          Lcd_Cmd( _LCD_CLEAR);
132:      }
133:      if (etapa == 2) {
134:          Candidatos();
135:          Lcd_Cmd( _LCD_CLEAR);
136:          gravarVOTO();
137:          Lcd_Cmd( _LCD_CLEAR);
138:      }
139:
140:      if (etapa == 3) {
141:          gravarTAG();
142:          Lcd_Cmd( _LCD_CLEAR);
143:      }
144:  }
145:  Lcd_Cmd( _LCD_CLEAR);
146:  }
147: }
148:
149: void LeitorRFID()
150: {
151:     etapa=0;
152:
153:     while (etapa == 0)
154:     {
155:         Lcd_Out(2,1,"    APROXIME SEU    ");
156:         Lcd_Out(3,1,"    CARTAO        ");
157:         if (RA2_bit==0) { // checar se o botão MENU foi pressionado
158:             etapa=10;
159:             play=0;
160:         }
161:         if(UART1_Data_Ready())
162:         {
163:             for(i=0;i<6;)      // algoritmo para ler os 6 caracteres da tag
164:             {
165:                 if(UART1_Data_Ready())
166:                 {
167:                     rfid[i] = UART1_Read(); // grava a tag lida para a string
168:                     i++;
169:                 }
170:             }
171:
172:             if((rfid[0] ^ rfid[2] ^ rfid[4] ^ rfid[6] ^ rfid[8] == rfid[10]) && (rfid[
173: 1] ^ rfid[3] ^ rfid[5] ^ rfid[7] ^ rfid[9] == rfid[11]))
174:             {
175:                 etapa = 1; // lido a tag corretamente avança a etapa
176:             }
177:             else
178:             {
179:                 Lcd_Out(2,1,"Error ");
180:             }
181:         }
182:     }
183: }
184:
185: void LeituraEEPROM()
```

```
186: {
187:   Lcd_Out(1,7,"Checando");
188:   Lcd_Cmd(_LCD_BLINK_CURSOR_ON);
189:   k=0;
190:   votou=0; // votou=1 significa que a pessoa já votou
191:   while (k<252)
192:   {
193:     aa=0;
194:     b = k;
195:     for (e=0;e<6;) // algoritmo para checar se os 6 caracteres da tag atual
196:     { // já esta salva na EEPROM
197:       I2C1_Start();
198:       I2C1_Wr(0xA2);
199:       I2C1_Wr(b);
200:       I2C1_Repeated_Start();
201:       I2C1_Wr(0xA3);
202:       delay_ms(30);
203:
204:       if (RFID[e] == I2C1_Rd(0u)) { // Sendo igual o caracter atual, da
205:         b++;
206:         e++;
207:         aa++;
208:       }
209:
210:       else { // caso o caracter que esta sendo comparado for diferente
211:         já desloca 6 posições a frente na EEPROM
212:         k = k + 6;
213:         e = 8;
214:       }
215:       I2C1_Stop();
216:     }
217:
218:     if (aa == 6) { // aa=6 significa que foi encontrado os 6 caracteres da
219:       tag já gravado na EEPROM externa
220:       k = 259;
221:       votou = 1; // portanto votou=1 pois a pessoa já votou
222:     }
223:   }
224:   clearLCD ();
225: }
226:
227: void statusTAG()
228: {
229:   if (votou == 0) { // se a pessoa não votou, avança a etapa
230:     Lcd_Out(3,1,"      VOTO PENDENTE  ");
231:     delay_ms(2000);
232:     etapa++;
233:   }
234:
235:   else { // caso contrário volta para a etapa 0
236:     Lcd_Out(2,1,"      SEU VOTO JA FOI  ");
237:     Lcd_Out(3,1,"      REGISTRADO      ");
238:     Delay_ms(3500);
239:   }
240: }
241:
242: void Candidatos()
243: {
```

```
244:   confirma=0; // confirmar=1 significa que o foi confirmado o candidato
245:   while (confirma == 0)
246:   {
247:       Lcd_Out(2,1,"ESCOLHA SEU");
248:       Lcd_Out(3,1,"CANDIDATO:");
249:       escolheu=0;
250:       while(escolheu==0) {
251:
252:           if (RB0_bit==0) {
253:               Lcd_Cmd(_LCD_CLEAR);
254:               escolheu = 1;
255:               Lcd_Out(2,1,"CANDIDATO 1");
256:           }
257:
258:           if (RB1_bit==0) {
259:               Lcd_Cmd(_LCD_CLEAR);
260:               escolheu = 2;
261:               Lcd_Out(2,1,"CANDIDATO 2");
262:           }
263:
264:           if (RB2_bit==0) {
265:               Lcd_Cmd(_LCD_CLEAR);
266:               escolheu = 3;
267:               Lcd_Out(2,1,"CANDIDATO 3");
268:           }
269:
270:           if (RB3_bit==0) {
271:               Lcd_Cmd(_LCD_CLEAR);
272:               escolheu = 4;
273:               Lcd_Out(2,1,"CANDIDATO 4");
274:           }
275:
276:           if (RB4_bit==0) {
277:               Lcd_Cmd(_LCD_CLEAR);
278:               escolheu = 5;
279:               Lcd_Out(2,1,"CANDIDATO 5");
280:           }
281:
282:           if (RB5_bit==0) {
283:               Lcd_Cmd(_LCD_CLEAR);
284:               escolheu = 6;
285:               Lcd_Out(2,1,"CANDIDATO 6");
286:           }
287:
288:           if (RB6_bit==0) {
289:               Lcd_Cmd(_LCD_CLEAR);
290:               escolheu = 7;
291:               Lcd_Out(2,1,"CANDIDATO 7");
292:           }
293:
294:           if (RB7_bit==0) {
295:               Lcd_Cmd(_LCD_CLEAR);
296:               escolheu = 8;
297:               Lcd_Out(2,1,"CANDIDATO 8");
298:           }
299:       }
300:       Lcd_Out(3,1,"CONFIRMA?");
301:       for(v=0;v<1;){
302:           if (RA1_bit==0) { // CORRIGIR pressionado retorna para o começo do w
while
303:               escolheu = 0;
304:               Lcd_Cmd(_LCD_CLEAR);
```

```
305:         v++;
306:     }
307:     if (RA0_bit==0) { // CONFIRMAR pressionado encerra o while e prosseg
gue
308:         confirma = 1;
309:         v++;
310:     }
311: }
312: }
313: }
314:
315: void gravarVOTO() // VOTOS são armazenados na EEPROM Interna
316: {
317:     votos = EEPROM_Read(escolheu); // Lê os votos do Candidato escolhido
318:     votos++; // Incrementa o voto
319:     EEPROM_Write(escolheu,votos); // Grava o valor atualizado dos votos do
o Candidato
320:     Lcd_Out(3,1," VOTO REGISTRADO ");
321:     delay_ms(1000);
322:     votos = EEPROM_Read(9); //Lê o número total de votos
323:     votos++;
324:     EEPROM_Write(9,votos);
325:     etapa++; // Avança para próxima etapa
326: }
327:
328: void gravarTAG() // Todas as tags RFID são grvadas na EEPROM Externa
329: {
330:     hh = EEPROM_Read(20); // Valor que contém a posição livre da EEPROM e
externa
331:     for(cc=0;cc<6;cc++) {
332:         I2C1_Start();
333:         I2C1_Wr(0xA2);
334:         I2C1_Wr(hh);
335:         I2C1_Wr(rfid[cc]); // string com a tag lida pelo leitor
336:         I2C1_Stop();
337:         Delay_ms(20);
338:         hh++;
339:     }
340:     EEPROM_Write(20,hh); // grava a nova posição da EEPROM externa
341: }
342:
343: void ContarVOTO() // Lê os votos da EEPROM & converte para as strings
344: {
345:     can1 = EEPROM_Read(1);
346:     can2 = EEPROM_Read(2);
347:     can3 = EEPROM_Read(3);
348:     can4 = EEPROM_Read(4);
349:     can5 = EEPROM_Read(5);
350:     can6 = EEPROM_Read(6);
351:     can7 = EEPROM_Read(7);
352:     can8 = EEPROM_Read(8);
353:     can9 = EEPROM_Read(9);
354:     ByteToStr(can1, txt1);
355:     ltrim(txt1);
356:     ByteToStr(can2, txt2);
357:     ltrim(txt2);
358:     ByteToStr(can3, txt3);
359:     ltrim(txt3);
360:     ByteToStr(can4, txt4);
361:     ltrim(txt4);
362:     ByteToStr(can5, txt5);
363:     ltrim(txt5);
```

```
364:         ByteToStr(can6, txt6);
365:         ltrim(txt6);
366:         ByteToStr(can7, txt7);
367:         ltrim(txt7);
368:         ByteToStr(can8, txt8);
369:         ltrim(txt8);
370:         delay_ms(500);
371: }
372:
373:
374: void mostrarVOTO()
375: {
376:     while(RA3_bit==1) // VOLTAR pressionado termina de exibir
377:     {
378:         Lcd_out(1,1,"1:");
379:         Lcd_out(1,4,txt1);
380:         Lcd_out(2,1,"2:");
381:         Lcd_out(2,4,txt2);
382:         Lcd_out(3,1,"3:");
383:         Lcd_out(3,4,txt3);
384:         Lcd_out(1,7,"4:");
385:         Lcd_out(1,10,txt4);
386:         Lcd_out(2,7,"5:");
387:         Lcd_out(2,10,txt5);
388:         Lcd_out(3,7,"6:");
389:         Lcd_out(3,10,txt6);
390:         Lcd_out(1,13,"7:");
391:         Lcd_out(1,16,txt7);
392:         Lcd_out(2,13,"8:");
393:         Lcd_out(2,16,txt8);
394:     }
395:     Lcd_Cmd(_LCD_CLEAR);
396: }
397:
398: void zerarURNA()
399: {
400:     Lcd_Out(2,1,"* Insira o codigo");
401:     Lcd_Out(3,1,"* Pressione Voltar");
402:     for (v=0;v<1;)
403:     {
404:         if (PORTB==0b01111100) // Combinação para zerar a urna
405:         {
406:             Lcd_Cmd(_LCD_CLEAR);
407:             Lcd_Out(3,7,"Aguarde");
408:             Lcd_Cmd(_LCD_BLINK_CURSOR_ON);
409:             for(e=0;e<252;e++) { // Apaga EEPROM externa
410:                 I2C1_Start();
411:                 I2C1_Wr(0xA2);
412:                 I2C1_Wr(e);
413:                 I2C1_Wr(0);
414:                 I2C1_Stop();
415:                 Delay_ms(30);
416:             }
417:             for(e=0;e<25;e++) { // Apaga EEPROM interna
418:                 EEPROM_Write(e,0);
419:             }
420:             clearLCD ();
421:             v++;
422:             play=2; // Encerra while do menu e prossegue para votação
423:         }
424:         if (RA3_bit==0) { // VOLTAR pressionado volta para o menu
425:             clearLCD ();
```

```
426:         play=1; //Encerra while do menu e entra novamente no menu
427:         v++;
428:     }
429: }
430:
431: }
432:
433: void checarEEPROM ()
434: {
435:     hh = EEPROM_Read(20); // posição da EEPROM interna onde fica salvo o próximo endereço livre da EEPROM externa
436:     if (hh>251) {
437:         Lcd_Cmd(_LCD_CLEAR);
438:         Lcd_Out(3,5,"MEMORIA CHEIA");
439:         delay_ms(3000);
440:         etapa=0; // irá voltar para a etapa 0 até iniciar uma nova votação
441:         Lcd_Cmd(_LCD_CLEAR);
442:     }
443: }
```