

Controle para basculante – Relatório final

Gilson da Rosa Webber¹, Rafael da Fonte Lopes da Silva¹

¹Instituto de Informática/Departamento de Engenharia Elétrica
Universidade Federal do Rio Grande do Sul (UFRGS)

gilson.webber@inf.ufrgs.br, rflsilva@inf.ufrgs.br

1. Objetivo

O objetivo inicial da nossa proposta era o de projetar e construir um dispositivo utilizando um microcontrolador **PIC-16F684** da *MicroChipTM* que fosse capaz de gerenciar a abertura de uma janela do tipo basculante. Este controle seria feito através de medidas da luminosidade do ambiente, de acordo com preferências definidas pelo usuário. Os modos sob os quais ele iria operar seriam os seguintes:

- Controle manual da abertura, feito diretamente pelo usuário;
- Controle automático realizado pelo PIC, de acordo com o grau de iluminação ambiente percebido pelo sensor.

2. Metodologia e resultados

2.1. Circuito

O circuito utilizado no projeto foi desenvolvido de acordo com as ideias que foram agregadas. Inicialmente, o projeto restringia-se ao controle automático pelo PIC da abertura da basculante de acordo com a luminosidade informada pelo LDR. Aos poucos, novas ideias foram integradas, agregando valor e complexidade ao trabalho. Dentre estas, adicionamos a possibilidade de o usuário do dispositivo configurá-lo através do uso de um controle remoto, trazendo assim a filosofia de dois modos de operação diferentes: um manual (onde o usuário pode controlar a abertura da basculante de forma customizada) e outro automático (no qual o usuário seta um valor de luminosidade a ser atingido pelo LDR, deixando o controle da abertura apenas a cargo do PIC). Além disso, adicionamos um display duplo de sete segmentos, através do qual o usuário poderá ler os valores de luminosidade lidos pelo LDR instantaneamente, além de facilitar a configuração do dispositivo.

A Figura 4 mostra o resultado final do esquemático do projeto. Os componentes principais utilizados na confecção do projeto são listados a seguir, considerando todas as adições que fizemos ao longo do tempo:

- Microcontrolador **PIC-16F684**;
- Um LDR;
- Um receiver de infravermelho TSOP-1836;
- Um IC Driver L293D;
- Um motor de passo;
- Dois IC Shifters 74HC595;
- Um display de 7 segmentos duplo A-562G.

Além destes, componentes como resistores, capacitores e transistores também foram introduzidos, de modo a prover certas características de tensão/corrente convenientes (como no caso do LDR, onde o resistor de pull-up limita a tensão aplicada à entrada do PIC, realizando um divisor de tensão, e o capacitor em paralelo com ele, que ajuda na remoção de possíveis flutuações nesta tensão). Como podemos ver pelo esquemático do projeto, ao lado esquerdo do PIC estão os dispositivos que geram sinais a serem processados (LDR, receiver IR). Ao lado esquerdo, estão todos os sinais de saída do PIC utilizadas. Ele controla diretamente o driver do motor de passo, o qual será capaz de fornecer a corrente necessária e “isolar” o motor do resto do circuito. Finalmente, o PIC ainda fornece os sinais para os displays de sete segmentos de forma serial (os registradores se encarregam de prover os valores corretos para os displays).

2.2. Programa desenvolvido

Para este projeto de disciplina, era obrigatório que se realizasse a programação do microcontrolador utilizando-se sua respectiva linguagem assembly. De modo que pudéssemos suportar todos os periféricos citados, tivemos que fazer diversas adições ao código assembly inicial ao longo do projeto. Os eventos que causam interrupções no PIC são aqueles relacionados à entrada do controle remoto (teste de entrada de valor ou do amostrador de bits). Também foi necessário desenvolver trechos de código para realizar a conversão de valores BCD para a leitura a partir do conversor A/D alimentado pelo sinal do LDR.

O controle das bobinas do motor também é realizado por software, havendo a possibilidade de se utilizar diferentes sequências de passo de acordo com a necessidade da aplicação. Outro item importante do código é a recepção de comandos do controle remoto que, da forma como está organizada, permite o desenvolvimento de novas rotinas, facilitando uma melhor configurabilidade do dispositivo.

2.3. Ambiente de teste

Além do circuito microcontrolado proposto, desejávamos confeccionar um ambiente próprio para testar nossa solução e validar a ideia do projeto. Deste modo, decidimos por construir um pequeno sistema basculante de teste, o qual permitisse que o circuito microcontrolado pudesse realizar o controle de aletas móveis. O movimento delas iria, então, alterar o grau de luminosidade do interior de uma caixa sobre a qual colocaríamos este sistema de aletas e, dentro desta caixa, fixaríamos o sensor do dispositivo.

As figuras a seguir ilustram a construção deste sistema. A Figura 1 mostra em detalhe parte do mecanismo utilizado para movimentar em conjunto as aletas da basculante. A Figura 2, por sua vez, tem por finalidade mostrar como as aletas são fixadas numa estrutura única. Já a Figura 3 dá a ideia de como a estrutura de aletas é movimentada pelo motor de passo, além de definir seus limites de curso. As posições limites das aletas são: completamente fechadas (luminosidade mínima) e abertas em 90° (luminosidade máxima, caso a fonte de luz emita raios perpendiculares ao plano superior da caixa).

Como no nosso caso de testes a distância entre as aletas e o fundo da caixa onde está o sensor é muito pequena, a luz incidente no interior dela pouco se difunde, sendo fortemente coletada nas regiões diretamente abaixo das frestas entre as aletas. Isso faz com que haja pouca gradação das intensidades de luz na região do sensor, e não permitindo que se atinja certos valores de quantidade de luz lidos pelo LDR (como ocorreria

num ambiente mais amplo). Desta forma, ainda precisamos utilizar papel alumínio na base da caixa para tornar a distribuição de luz mais uniforme e difusa pelo seu interior, e ainda garantir que o LDR fosse estimulado por este tipo de luz apenas. Com essas decisões, foi possível garantir que houvesse uma gradação mais suave dos níveis de luz dentro da caixa, permitindo que as aletas assumissem diversas posições diferentes de acordo com a configuração programada no modo automático do PIC.



Figura 1. Teste do esquema de movimentação das aletas.



Figura 2. Detalhe da montagem das aletas da basculante.

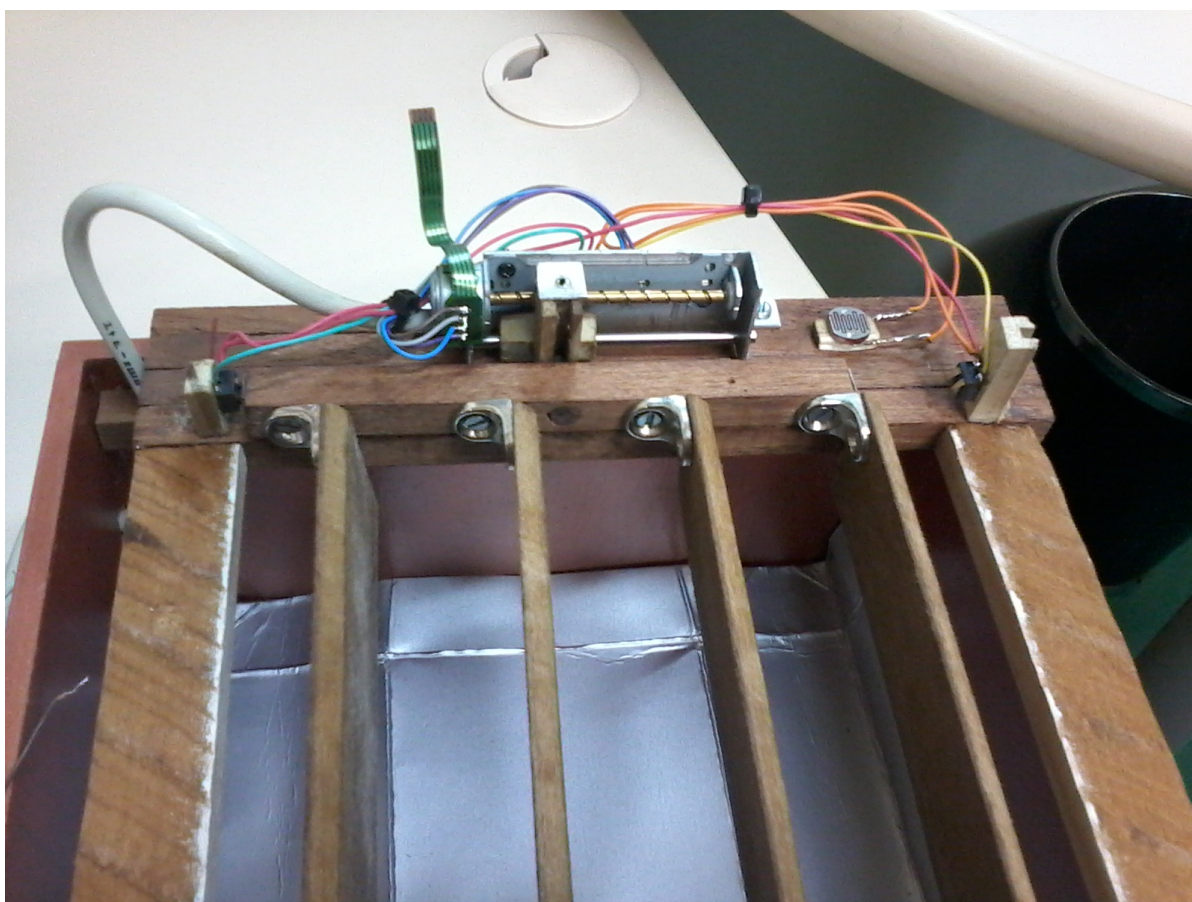


Figura 3. Detalhe do motor linear, da estrutura de movimentação das aletas e dos sensores de limite de curso.

3. Código do PIC

```
1 ; Firmware de controle de basculante automática/manual por sensor de
2 ; luminosidade .
3 ;
4 ; Desenvolvedores :
5 ; Gilson da Rosa Webber
6 ; Rafael da Fonte Lopes da Silva
7 ;
8 ; Data: Jun 2014
9 ;
10 ; Universidade Federal do Rio Grande do Sul – UFRGS
11 ; Instituto de Informática – INF
12 ; Departamento de Engenharia Elétrica – DELET
13 ;
14 ; ENG04056 – Microcontroladores
15 ; Prof.: Gilson Inácio Wirth
16 ;
17 ;
18 #include <p16f684.inc>
19
20 __CONFIG _FOSC_INTOSCIO & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF & _CPD_OFF &
21
22 errorlevel -302 ; suppress message 302 from list file
23
24 ADDR_EE_POS EQU 0x00
25 IrDAb EQU RA2 ; bit da porta de recepção do controle
26 IrDap EQU PORTA ; porta de recepção do controle
27 ENA_DISP EQU RA5 ; bit da porta de controle do display
28 DISPLAYp EQU PORTA ; porta de controle do display
29 DEVICE EQU 0x01 ; MODO TV DO CONTROLE SONY RMT-V297C
30 ST_AUTO EQU 0 ; Estado de operação automática
31 ST_COMMAND EQU 1 ; Estado de execução de um comando
32 ST_DISP_SET EQU 2
33 ST_WRITE_CHANGE EQU 3
34 ST_MOVE_ALL EQU 4
35
36
37
38
39 UDATA 0x50; Inicio da área de variáveis (RAM) – Bloco 0
40 IrDA_bits res 1 ; quantidade de bits enviados pelo controle
41 IrDA_discard res 1 ; contador para discarte de envio multiplo do controle
42 IrDA_DATA res 3 ; dados recebidos do controle remoto
43 IrDA_command res 1 ; commando enviado pelo controle remoto
44 SMCON res 1 ; controle da máquina de estados
45 count_disp res 1 ; contador de bits enviados para o display
46 w_disp res 1 ; temporário para operações de envio serial
47
48 pos_value res 1 ; 0 desligado , outros define o nível de luminosidade
49 motor_pos res 1 ; quais bobinas foram acionadas na ultima chamada
50 counter res 1 ; contador multiuso
51
52 B1 res 1 ; digito um do display 7-seg
53 B2 res 1 ; digito dois do display 7-seg
54 B1_temp res 1 ; temporário
```



```

55 B2_temp          res 1      ; temporário
56 LOOP_BCD_COUNTER res 1      ; ?
57 VALUE_BIN        res 1      ; valor binário do display , não usado no momento
58 com_routine      res 1      ; armazena a função a ser executado pelo comando
59
60
61                  UDATA_SHR   ; área de memória compartilhada entre blocos
62 w_temp           res 1      ; acumulador temporário
63 w_bk             res 1      ; armazena backup do acumulador
64 status_bk        res 1      ; backup do registrador de status
65
66
67 record_data       res 1      ; dado a ser gravado na eeprom
68 record_addr       res 1      ; endereço a ser gravado na eeprom
69
70
71 RESET_VECTOR      CODE      0x0000 ; ponto de execução em caso de reset
72          goto      start          ; salta para o início da execução
73
74
75 ;=====
76 ; Seção de tratamento de interrupções
77 ;
78 ;   ESTA ESTRUTURA FOI UTILIZADA PARA NÃO PERMITIR CHAMADAS CONTINUAS DENTRO
79 ;   DESTA FUNÇÃO CASO SEJA CHAMADO MAIS DE UMA INTERRUPÇÃO, SERVE PARA MANTER
80 ;   A HIERARQUIA DE PROCESSAMENTO
81 ;
82 ;=====
83 INT_VECTOR        CODE      0x0004 ; interrupt vector location
84
85 INTERRUPT
86          BCF      INTCON, GIE      ; DESABILITA INTERRUPÇÃO GLOBAL
87          BCF      WDTCN, SWDTEN    ; DESTIVA O WATCH-DOG
88
89          movwf    w_bk              ; salva o conteudo do acumulador
90          movf      STATUS, W        ; copia o conteudo do registrador de status
91          movwf     status_bk        ; salva o conteúdo do registrador de status
92
93          BTFSC     INTCON, INTF      ; Interrupção pelo controle remoto
94          GOTO      IrDA_FALLING
95          BTFSC     PIR1, TMR2IF      ; Interrupção pelo contador (controle remoto)
96          GOTO      IrDA_TIME
97
98          ;RESOLVER                  ; interrupção não tratável , fica em loop
99          END_INTERRUPT
100         movf      status_bk , W      ; retrieve copy of STATUS register
101         movwf     STATUS              ; restore pre-isr STATUS register contents
102         swapf     w_bk , F            ;
103         swapf     w_bk , W            ; restore pre-isr W register contents
104
105         BSF      INTCON, GIE          ; HABILITA INTERRUPÇÃO GLOBAL
106         retfie                      ; return from interrupt
107
108 IrDA_FALLING
109         BCF      INTCON, INTF          ; zera bit da interrupção por controle remoto

```

```

110          CALL    IrDA_FALL      ; tratamento da recepção
111          GOTO    END_INTERRUPT
112 IrDA_TIME
113          BCF     PIR1, TMR2IF    ; zera bit de interrupção pelo contador
114          CALL    IrDA_SAMPLER   ; executa leitura do bit recebido
115          GOTO    END_INTERRUPT
116
117
118
119
120 ;=====
121 ; bootstrap da máquina
122 ;
123 ;
124 ;
125 ;=====
126
127 MAIN      CODE
128
129 start     ; PRÉ-SET DO AMBIENTE DE EXECUÇÃO
130          CLRF    INTCON
131
132          BSF     STATUS, RP0      ; APONTA PARA BLOCO 1
133          MOVLW   b'00000001'
134          MOVWF   OPTION_REG      ; DEFINE PRESCALER DO TIMER 0, COLOCA INT/RA2 EM FALL
135          MOVLW   0X1F             ; b'00011111'
136          MOVWF   TRISA           ; DEFINE I/O DA PORTA A
137          CLRF    TRISC           ; DEFINE PORTA C COMO SAÍDAS
138          MOVLW   b'00001000'
139          MOVWF   ANSEL           ; DEFINE AS PORTAS QUE SÃO ANALÓGICAS
140          CLRF    IOCA            ; Desativa InterruptOnChange para todas as portas
141          MOVLW   b'00000111'
142          MOVWF   WPUA           ; Ajusta PULL-UP
143          MOVLW   0X10
144          MOVWF   ADCON1         ; DEFINE OSCILADOR 1/8 ~ 2us PARA O CONVERSOR A/D
145          MOVLW   b'00000010'
146          MOVWF   PIE1           ; HABILITA INTERRUPÇÃO POR TIMER2
147          MOVLW   0XE1
148          MOVWF   PR2            ; COLOCA 160 NO COMPARADOR DO TIMER 2 (IrDA) 2560 us F
149
150
151          BCF     STATUS, RP0      ; APONTA PARA O BANCO 0
152          CLRF    PORTA           ; INIT PORTA
153          CLRF    PORTC
154          MOVLW   0X07
155          MOVWF   CMCON0         ; DESLIGA OS COMPARADORES
156          MOVLW   0X0C
157          MOVWF   ADCON0         ; HABILITA ENTRADA ANALÓGICA 3 E AJUSTA PARA SAIDA JU
158          MOVLW   0X16
159          MOVWF   WDTCN          ; AJUSTA PRESCALER DO WATCH-DOG PARA 1:65536 ~ 2 segun
160          CLRF    PIR1           ; ZERA FLAGS DE INTERRUPÇÕES DOS PERIFÉRICOS
161
162          MOVLW   b'01010000'
163          MOVWF   INTCON         ; HABILITA INT/RA2, PERIFÉRICOS
164

```



```

165         ; Inicialização de variáveis
166
167         ; Inicalização da RAM
168         ;BTFSC    EECON1, WRERR    ; TESTA SE HOUE ERRO DE GRAVAÇÃO DURANTE RESET
169         ;CALL     RECORD_POSITION
170         ;CALL     READ_POSITION
171
172         ; Teste da interação
173         MOVLW     0X80              ; 50%
174         MOVWF     pos_value
175         ;CALL     RECORD_POSITION
176
177     ;        CLRF      record_addr
178
179         CLRF      SMCON              ; inicializa estrutura da maquina de estados
180         MOVLW     0xFF
181         MOVWF     IrDA_command      ; inicializa comando do controle remoto
182         CALL      INIT_IrDA         ; inicializa estruturas de recepção do controle remoto
183         CLRF      B1                ; inicializa byte 1 do display 7-seg
184         CLRF      B2                ; inicializa byte 2 do 7-seg
185
186         BSF       DISPLAYp, ENA_DISP
187
188         BSF       INTCON, GIE        ; Habilita interrupções
189
190 MAIN_1
191         BTFSC     SMCON, ST_COMMAND
192         call      COMMANDS
193         BTFSS     SMCON, ST_DISP_SET
194         call      READ_LDR
195         BTFSC     SMCON, ST_AUTO
196         call      AUTO_REFRESH
197         goto      MAIN_1
198
199         GOTO      start              ; ocorreu algum erro , resetar
200
201
202 ;=====
203 ;
204 ; Seção com funções
205 ;
206 ;
207 ;=====
208
209 FUNCTIONS CODE
210
211 ;=====
212 ;
213 ; Delay de 1000 ciclos de clock de instrução
214 ;
215 ; Presupõe que a interrupção de overflow do timer0 está desativada , caso
216 ; contrário o comportamento é imprevisível.
217 ;
218 ; O timer utiliza um ajuste para se obter o consumo de 1000 ciclos desde
219 ; a instrução de chamada até o retorno na instrução seguinte
220 ;

```

```

221 ;=====
222
223 DELAYms: ; com preescaler em 1:4 desde o call até o retorno gasta 1000 ciclos
224
225 ; AJUSTA PARÂMETROS DO CONTADOR
226 ;
227 ; PRESCALER 1:4
228 ; INCREMENTA NA DESCIDA DO CLOCK
229 ; USA O CLOCK DE INSTRUÇÃO COMO BATE DE TEMPO
230 ;
231 MOVLW    b'11000000'
232 ANDWF    OPTION_REG, W
233 IORLW    b'00000001'
234 MOVWF    OPTION_REG
235
236 ;laço do timer
237 MOVLW    0X0A                ; AJUSTA CONTADOR PARA GERAR A SAÍDA CORRETA
238 MOVWF    TMR0                ; ATUALIZA CONTADOR
239 BCF      INTCON, T0IF
240 BTFSS    INTCON, T0IF
241 GOTO     $-1
242 RETURN
243
244
245 ;=====
246 ;
247 ; INICIALIZA ESTRUTURAS PARA RECEPÇÃO DE UM COMANDO DO CONTROLE REMOTO
248 ;
249 ;=====
250 INIT_IrDA:
251 CLRF     IrDA_bits
252 MOVLW    b'00010001'        ; PRESCALER 1:4 POSTSCALER 1:3
253 MOVWF    T2CON
254 CLRF     TMR2                ; ZERA CONTADOR
255 RETURN
256
257 ;=====
258 ;
259 ; TRATAMENTO DA RECEPÇÃO DOS DADOS DO CONTROLE REMOTO
260 ; PRESUPOE QUE TIMER 2 NÃO É USADO POR MAIS NINGUEM
261 ;
262 ;=====
263 IrDA_FALL:
264 BTFSC    T2CON, 6            ; testa se está em aguardo de novo comando 1:16
265 CALL     INIT_IrDA
266
267 BTFSC    T2CON, 5
268 GOTO     IrDA_FALL_SAMPLE_1 ; se 1:5
269
270 BTFSC    T2CON, TMR2ON
271 GOTO     IrDA_FALL_ABORT    ; se não é 1:5 e está com o timer ligado
272
273 BTFSC    T2CON, 4
274 GOTO     IrDA_FALL_SAMPLE   ; se 1:3
275

```

```

276 IrDA_FALL_ABORT
277     CALL     INIT_IrDA
278     RETURN
279
280 IrDA_FALL_SAMPLE_1
281     BCF      T2CON, 5           ; passa para 1:1
282 IrDA_FALL_SAMPLE
283     CLRF     TMR2
284     BSF      T2CON, TMR2ON
285     RETURN
286
287 ;=====
288 ;
289 ;   ROTINA QUE TRATA DA LEITURA DO BIT DO IrDA NA PORTA
290 ;   BYTE 0 – DEVICE
291 ;   BYTE 1 – COMANDO
292 ;   BYTE 2 – EXTENÇÃO (SÓ EM 20 BITS)
293 ;
294 ;   INTERRUPÇÃO DEVE CESSAR DEPOIS DE ALTERAR ESTADO DA MÁQUINA DE ESTADOS
295 ;=====
296
297 IrDA_SAMPLER:
298     BTFSC    T2CON, 6
299     GOTO     IrDA_SAMPLER_RESET ; tem que executar INIT e zerar contador
300     BTFSC    T2CON, 5           ; TESTA SE É 1:5
301     GOTO     IrDA_SAMPLER_5    ; 1:5
302     BTFSS    T2CON, 4           ;TESTA DE É 1:3
303     GOTO     IrDA_SAMPLER_1
304     BTFSC    IrDAp, IrDAb       ; TESTA NÍVEL LÓGICO DA PORTA, DEVE SER ALTO
305     GOTO     IrDA_SAMPLER_3
306     goto     IrDA_SAMPLER_RESET
307     RETURN
308
309 IrDA_SAMPLER_1
310     INCF     IrDA_bits, F
311
312     BCF      STATUS, C          ; INVERTE BIT VINDO DA LÓGICA
313     BTFSS    IrDAp, IrDAb
314     BSF      STATUS, C
315
316     ; COLOCA BIT NO ARRAY
317     RRF      IrDA_DATA, F
318     RRF      IrDA_DATA+1, F
319     RRF      IrDA_DATA+2, F
320
321 IrDA_SAMPLER_3
322     BCF      T2CON, 4           ; passa para 1:1
323     BSF      T2CON, 5           ; ATIVA POSTSCALER 1:5
324     CLRF     TMR2               ; mantém contador ativo
325     RETURN
326
327
328 ; PRESCALER DE 1:5
329 ; ATINGIU O LIMITE DE TEMPO DE ESPERA
330 ; VERIFICA SE A QUANTIDADE DE BITS SE ENQUADRA EM ALGUM PADRÃO

```

```

331 IrDA_SAMPLER_5
332     MOVLW    b'01111111'
333     MOVWF    T2CON           ; ativa pos e pre 1:16
334
335     MOVLW    0X0C           ; 12 BITS
336     XORWF    IrDA_bits, W
337     BTFSC    STATUS, Z
338     GOTO     IrDA_COMMAND_12_BITS
339
340     MOVLW    0X0F           ; 15 BITS
341     XORWF    IrDA_bits, W
342     BTFSC    STATUS, Z
343     GOTO     IrDA_COMMAND_15_BITS
344
345     MOVLW    0X14           ; 20 BITS
346     XORWF    IrDA_bits, W
347     BTFSS    STATUS, Z
348     goto     IrDA_SAMPLER_RESET
349
350 ; INTERVALO DE CÓDIGO QUE CORRIGE POSIÇÃO DOS BITS
351
352     MOVF      IrDA_DATA, W
353     MOVWF    w_temp
354     MOVF      IrDA_DATA+1, W
355     MOVWF    IrDA_DATA
356     MOVF      IrDA_DATA+2, W
357     MOVWF    IrDA_DATA+1
358     MOVF      w_temp, W
359     MOVWF    IrDA_DATA+2
360
361 IrDA_COMMAND_12_BITS
362     MOVLW    b'11110000'
363     ANDWF    IrDA_DATA+1, F
364     BCF      STATUS, C
365     RRF      IrDA_DATA, F
366     RRF      IrDA_DATA+1, F
367     RRF      IrDA_DATA, F
368     RRF      IrDA_DATA+1, F
369     RRF      IrDA_DATA, F
370     RRF      IrDA_DATA+1, F
371
372 IrDA_COMMAND_15_BITS
373
374     BCF      STATUS, C
375     RRF      IrDA_DATA+1, F
376
377 ; FIM DAS CORREÇÕES DE POSIÇÃO
378
379
380     MOVF      IrDA_DATA, W
381     XORLW    DEVICE
382     BTFSS    STATUS, Z           ; testa dispositivo
383     RETURN
384
385 ; TODO: tratar melhor o descarte do envio triplo

```

```

386      ; comando diferente do anterior , zera contador
387      ; commando igual incrementa e testa
388      ; caso atinja 2, zera comando
389      MOVF      IrDA_command, W
390      XORWF     IrDA_DATA+1, W
391      BTFSS     STATUS, Z           ; testa se comando é igual ao anterior
392      CLRF      IrDA_discard
393      INCF      IrDA_discard, F
394      MOVLW     0x03
395      XORWF     IrDA_discard, W
396      BTFSC     STATUS, Z
397      CLRF      IrDA_discard
398      MOVLW     0x01
399      XORWF     IrDA_discard, W
400      BTFSS     STATUS, Z
401      RETURN
402
403      MOVF      IrDA_DATA+1, W
404      MOVWF     IrDA_command
405
406      ; LOCALIZAR AÇÃO NA TABELA
407      ;MOVF      IrDA_DATA+1, W
408      BTFSC     SMCON, ST_AUTO      ; testa qual tabela deve ser consultada auto/manua
409      CALL      REMOTE_CONTROL_COMMANDS.AUTO
410      BTFSS     SMCON, ST_AUTO
411      CALL      REMOTE_CONTROL_COMMANDS.MANUAL
412
413      MOVWF     com_routine
414
415      ANDLW     0xFF
416      BTFSS     STATUS, Z
417      ;CALL      COMMANDS
418      BSF       SMCON, ST_COMMAND
419      RETURN
420
421 IrDA_SAMPLER_RESET
422      ;MOVLW     0xFF
423      ;MOVWF     IrDA_command
424      CLRF      IrDA_discard
425      CALL      INIT_IrDA           ; INICIALIZA ESTRUTURAS
426      RETURN
427
428
429
430 ;=====
431 ;
432 ;   GERENCIA SEQUENCIA PARA LEITURA E GRAVAÇÃO NA EEPROM
433 ;
434 ;=====
435
436 RECORD_EEPROM:
437      BCF       INTCON, GIE         ; DESATIVA INTERRUPÇÃO GLOBAL
438      BTFSC     INTCON, GIE         ; TESTA SE INTERRUPÇÃO FOI DESABILITADA
439      GOTO      $-2                ; REPETE SE AINDA NÃO ESTÁ DESATIVADO
440      BSF       STATUS, RP0         ; APONTA PARA BLOCO 1

```

```

441      BCF      EECON1, WRERR      ; REMOVE MARCAÇÃO DE ERRO DE GRAVAÇÃO
442      BSF      EECON1, WREN      ; HABILITA GRAVAÇÃO
443      ;MOVWF   pos_value, W      ; COLOCA O DADO NO REGISTRADOR DE DADOS
444      MOVWF    EEDAT
445      MOVF     record_addr, W      ; COLOCA A POSIÇÃO NO REGISTRADOR DE POSIÇÃO
446      MOVWF    EEADR
447      ; SEQUENCIA DE INICIALIZAÇÃO EXIGIDA PARA A GRAVAÇÃO NA EEPROM
448      MOVLW    0X55
449      MOVWF    EECON2
450      MOVLW    0XAA
451      MOVWF    EECON2
452      BSF      EECON1, WR      ; ATIVA GRAVAÇÃO
453      ; FIM DA SEQUENCIA EXIGIDA
454      BTFSC    EECON1, WR      ; TESTA SE GRAVAÇÃO FOI CONCLUIDA
455      GOTO     $-1
456      BCF      STATUS, RP0
457      BSF      INTCON, GIE      ; HABILITA INTERRUPÇÃO GLOBAL
458      RETURN
459
460 READ_EEPROM:
461      BCF      INTCON, GIE
462      BSF      STATUS, RP0      ; APONTA PARA BLOCO 1
463      MOVLW    ADDR_EE_POS
464      MOVWF    EEADR
465      BSF      EECON1, RD
466      BTFSC    EECON1, RD
467      GOTO     $-1
468      MOVF     EEDAT, W
469      MOVWF    pos_value
470      BCF      STATUS, RP0
471      BSF      INTCON, GIE
472      RETURN
473
474 ;=====
475 ;
476 ; Rotina que implementa a conversão A/D do sensor de luminosidade e permanece
477 ; até que se atinja o nível de luminosidade desejado
478 ;
479 ;=====
480 AUTO_REFRESH:
481      BSF      ADCON0, ADON      ; LIGA CONVERSADOR A/D
482      BSF      ADCON0, GO_NOT_DONE ; INICIA CONVERSÃO
483
484 AD_DONE
485      BTFSC    ADCON0, GO_NOT_DONE ; TESTA SE TERMINOU
486      GOTO     AD_DONE
487      MOVF     ADRESH, W      ; ARMAZENA O RESULTADO DA CONVERSÃO
488
489
490
491      SUBWF    pos_value, W      ; COMPARA OS VALORES
492      BTFSC    STATUS, Z      ; SE SÃO IGUAIS NÃO FAZ NADA
493      GOTO     REFRESH_watch
494
495      BTFSS    STATUS, C

```



```

496         goto    AD_FORWARD
497
498         BTFSS    PORTA, RA1
499         goto    REFRESH_END
500         CALL     MOTOR_REVERSE        ; fechar
501         goto    REFRESH_END
502
503 AD_FORWARD
504         BTFSS    PORTA, RA0
505         goto    REFRESH_END
506         CALL     MOTOR_DIRECT         ; abrir
507
508 REFRESH_END
509         BCF      ADCON0, ADON        ; DESLIGA CONVERSOR
510         ;CALL    MOTOR_OFF           ; REMOVE CONSUMO DE CORRENTE
511         CALL     DELAYms
512         CALL     DELAYms
513         CALL     DELAYms
514         CALL     DELAYms
515         CALL     DELAYms
516         CALL     DELAYms
517         CALL     DELAYms
518         CALL     DELAYms
519         CALL     DELAYms
520         CALL     DELAYms
521         CALL     DELAYms
522         CALL     DELAYms
523         CALL     DELAYms
524         CALL     DELAYms
525         CALL     DELAYms
526         RETURN
527
528 REFRESH_watch
529         BCF      ADCON0, ADON        ; DESLIGA CONVERSOR
530
531         BSF      WDTCON, SWDTEN
532         SLEEP
533         BCF      WDTCON, SWDTEN
534
535         return
536
537 ;=====
538 ;
539 ;   GERENCIA O MOTOR DE PASSO CONECTADO AS SAÍDAS RC2~RC5
540
541 MOTOR_OFF:
542         CLRF     w_temp
543         GOTO     MOTOR_DO
544 MOTOR_DIRECT:
545         BTFSS    PORTA, RA0
546         RETURN
547         INCF     motor_pos, F
548         GOTO     $+4
549 MOTOR_REVERSE:
550         BTFSS    PORTA, RA1
551         RETURN

```

```

552      DECF      motor_pos , F
553      MOVF      motor_pos , W
554      CALL     BIT_PATTERN
555      MOVWF     w_temp
556
557 MOTOR_DO
558      MOVLW     b'11000011'
559      ANDWF     PORTC, W
560      IORWF     w_temp, W
561      MOVWF     PORTC
562      RETURN
563
564 ;-----
565 ;
566 ;   Conversor binário -> BCD
567 ;
568 ;-----
569
570 BCD_CONVERT:
571     MOVWF VALUE_BIN
572     ;inicializa laço (8 vezes)
573     MOVLW 0x80
574     MOVWF LOOP_BCD_COUNTER
575     CLRF B1                                ;b1; - primeiro byte com dois nibbles de bcd
576     CLRF B2                                ;b2; - segundo byte, com um unico nibble de bcd
577
578     GOTO KEEP_SHIFTING
579 LOOP_SHIFT
580     BTFSC B1, 3                            ;Testa se primeiro nibble eh maior ou igual a oito
581     GOTO ADD_B1_0
582     BTFSS B1, 2                            ;Testa se eh maior ou igual a quatro
583     GOTO TEST_B1_1                        ;nibble ok, continuar
584     BTFSC B1, 1
585     GOTO ADD_B1_0                        ;tem que somar
586     BTFSS B1, 0
587     GOTO TEST_B1_1
588
589 ADD_B1_0                                ;primeiro nibble eh maior que 5
590     MOVLW 0x03
591     ADDWF B1, f                            ;adiciona 3 ao nibble
592
593 TEST_B1_1
594     BTFSC B1, 7                            ;Testa se primeiro nibble eh maior ou igual a oito
595     GOTO ADD_B1_1
596     BTFSS B1, 6                            ;Testa se eh maior ou igual a quatro
597     GOTO TEST_B2                        ;nibble ok, continuar
598     BTFSC B1, 5
599     GOTO ADD_B1_1                        ;tem que somar
600     BTFSS B1, 4
601     GOTO TEST_B2
602
603 ADD_B1_1                                ;segundo nibble eh maior que 5
604     MOVLW 0x30
605     ADDWF B1, f                            ;adiciona 3 ao nibble
606
607 TEST_B2

```

```

608         BTFSC B2, 3                ;Testa se primeiro nibble eh maior ou igual a oito
609         GOTO ADD_B2
610         BTFSS B2, 2                ;Testa se eh maior ou igual a quatro
611         GOTO KEEP_SHIFTING         ;nibble ok, continuar
612         BTFSC B2, 1
613         GOTO ADD_B2                ;tem que somar
614         BTFSS B2, 0
615         GOTO KEEP_SHIFTING
616
617 ADD_B2
618         MOVLW 0x03
619         ADDWF B2, f                ;adiciona 3 ao nibble
620
621 KEEP_SHIFTING
622         RLF VALUE_BIN
623         RLF B1                    ;joga bit de carry no primeiro bit do registrador
624         RLF B2                    ;de novo, passa ultimo bit adiante
625
626         RRF LOOP_BCD_COUNTER, f
627         MOVF LOOP_BCD_COUNTER, f    ;testa se valor do registrador eh zero
628         BTFSS STATUS, Z            ;testa se chegamos ao fim da execucao
629         GOTO LOOP_SHIFT
630 final_da_execucao                ;realizar tratamentos finais
631         RETURN
632
633 ;=====
634 ;
635 ; Envia os valores bcd em B1 e B2 para o display
636 ;
637 ; a sensibilidade do clock do receptor do display é a borda de subida
638 ; para habilitar a saída manter enable em 0 para atualizar o display enviar pulso
639 ; de clock na linha de enable.
640 ;
641 ; o shift é de 16 bits
642 ;
643 ;=====
644 SEND_DISPLAY:
645         BSF     SMCON, 3
646         SWAPF   B1, F
647
648 START_SEND
649         MOVF    B1, W
650         CALL    SEVEN_SEG_CODE_LOOKUP
651         MOVWF   w_disp
652         MOVLW   0X08
653         MOVWF   count_disp
654 NEXT_BIT
655         RRF     w_disp, F
656
657         ; DEFINE BIT DE DADOS
658         BTFSS   STATUS, C
659         BCF     PORTC, 0
660         BTFSC   STATUS, C
661         BSF     PORTC, 0
662
663         ; APLICA TRANSIÇÃO

```

```

664      BCF      PORTC, 1
665      BSF      PORTC, 1
666      BCF      PORTC, 1
667
668      DECF      count_disp, F
669      BTFSS     STATUS, Z
670      GOTO      NEXT_BIT
671
672      SWAPF     B1, F
673      BTFSS     SMCON, 3
674      GOTO      END_SEND
675      BCF      SMCON, 3
676      GOTO      START_SEND
677
678
679  END_SEND
680      ; envia pulso de clock para o display atualizar
681      SWAPF     B1, F
682      BCF      DISPLAYp, ENA_DISP
683      BSF      DISPLAYp, ENA_DISP
684      BCF      DISPLAYp, ENA_DISP
685      RETURN
686
687
688  ;=====
689  ;
690      ;      Envia para o display a leitura instantânea do sensor de luminosidade
691  ;
692  ;=====
693  READ_LDR:
694      BTFSC     DISPLAYp, ENA_DISP
695      return
696      BSF      ADCON0, ADON          ; LIGA CONVERSADOR A/D
697      CALL     DELAYms
698      BSF      ADCON0, GO_NOT_DONE ; INICIA CONVERSÃO
699      BTFSC     ADCON0, GO_NOT_DONE ; TESTA SE TERMINOU
700      GOTO      $-1
701      BCF      ADCON0, ADON          ; desliga CONVERSADOR A/D
702      MOVF      ADRESH, W            ; ARMAZENA O RESULTADO DA CONVERSÃO
703      MOVWF     B1
704      CLRF      B2
705      CALL     SEND_DISPLAY
706      RETURN
707
708
709  ;=====
710  ;
711      ;      padrão de ativação das bobinas do motor de passo para meio passo
712  ;
713  ;=====
714      org      0x0200 ; AJUSTA INICIO DAS TABELAS, 255 POSIÇÕES
715  BIT_PATTERN:
716      CLRF      PCLATH
717      BSF      PCLATH, 1 ; ACERTA DESLOCAMENTO SOMADO AO PCL
718      ANDLW     0X07     ; remove possíveis saltos para fora da tabela

```

```

719 ADDWF PCL, F ; move o PC para a posição desejada
720 RETLW b'00000100' ; A
721 RETLW b'00001100' ; A~B
722 RETLW b'00001000' ; ~B
723 RETLW b'00011000' ; ~A~B
724 RETLW b'00010000' ; ~A
725 RETLW b'00110000' ; ~AB
726 RETLW b'00100000' ; B
727 RETLW b'00100100' ; AB

```

```

728 ;
729 ;=====
730 ;
731 ; CONTROLE REMOTO SONY RMT-V297C
732 ;

```

```

733 ; CÓDIGO DE DISPOSITIVO
734 ; TV
735 ; SIRC12 00001
736 ;
737 ; VCR
738 ; SIRC12 01011
739 ; SIRC15 10111010
740 ;

```

```

741 ;
742 ; KEYMAP:
743 ;

```

| | | | |
|-------|---------------|------|--------|
| 744 ; | 1 | 0x00 | VCR/TV |
| 745 ; | 2 | 0x01 | VCR/TV |
| 746 ; | 3 | 0x02 | VCR/TV |
| 747 ; | 4 | 0x03 | VCR/TV |
| 748 ; | 5 | 0x04 | VCR/TV |
| 749 ; | 6 | 0x05 | VCR/TV |
| 750 ; | 7 | 0x06 | VCR/TV |
| 751 ; | 8 | 0x07 | VCR/TV |
| 752 ; | 9 | 0x08 | VCR/TV |
| 753 ; | 0 | 0x09 | VCR/TV |
| 754 ; | ENTER | 0x0B | VCR/TV |
| 755 ; | | | |
| 756 ; | CH+ | 0x10 | VCR/TV |
| 757 ; | CH- | 0x11 | VCR/TV |
| 758 ; | VOL+ | 0x12 | TV |
| 759 ; | VOL- | 0x13 | TV |
| 760 ; | X2 | 0x14 | VCR |
| 761 ; | POWER | 0x15 | VCR/TV |
| 762 ; | EJECT | 0x16 | VCR |
| 763 ; | AUDIO MONITOR | 0x17 | VCR/TV |
| 764 ; | REC | 0x1D | VCR |
| 765 ; | SLOW | 0x23 | VCR |
| 766 ; | TV/VIDEO | 0x25 | TV |
| 767 ; | TV/VIDEO | 0x2A | VCR |
| 768 ; | DISPLAY | 0x3A | TV |
| 769 ; | MENU | 0x4D | VCR |
| 770 ; | INPUT SELECT | 0x4F | VCR |
| 771 ; | INDEX RIGHT | 0x56 | VCR |
| 772 ; | INDEX LEFT | 0x57 | VCR |
| 773 ; | SP/EP | 0x58 | VCR |
| 774 ; | DISPLAY | 0x5A | VCR |

| | | | | | | | |
|-----|---|-------------------------------|----------------|------|---|--------|-------------|
| 775 | ; | EASY TIMER | 0x60 | VCR | | | |
| 776 | ; | CLEAR | | 0x63 | VCR | | |
| 777 | ; | | | | | | |
| 778 | ; | UP | | 0x0E | VCR | SIRC15 | |
| 779 | ; | DOWN | | 0x0F | VCR | SIRC15 | |
| 780 | ; | LEFT | | 0x10 | VCR | SIRC15 | |
| 781 | ; | RIGHT | | 0x11 | VCR | SIRC15 | |
| 782 | ; | PLAY/OK | | 0x18 | VCR | SIRC15 | |
| 783 | ; | | | | | | |
| 784 | ; | | | | | | |
| 785 | ; | ===== | | | | | |
| 786 | | | | | | | |
| 787 | | REMOTE_CONTROL_COMMANDS.AUTO: | | | | | |
| 788 | | CLRF | PCLATH | | | | |
| 789 | | BSF | PCLATH, 1 | ; | ACERTA O DESLOCAMENTO SOMADO AO PCL | | |
| 790 | | ANDLW | 0X3F | ; | remove possíveis saltos para fora da tabela | | |
| 791 | | ADDWF | PCL, F | ; | move o PC para a posição desejada | | |
| 792 | | RETLW | cmd_digit_1 | ; | 1 | 0x00 | VCR/TV |
| 793 | | RETLW | cmd_digit_2 | ; | 2 | 0x01 | VCR/TV |
| 794 | | RETLW | cmd_digit_3 | ; | 3 | 0x02 | VCR/TV |
| 795 | | RETLW | cmd_digit_4 | ; | 4 | 0x03 | VCR/TV |
| 796 | | RETLW | cmd_digit_5 | ; | 5 | 0x04 | VCR/TV |
| 797 | | RETLW | cmd_digit_6 | ; | 6 | 0x05 | VCR/TV |
| 798 | | RETLW | cmd_digit_7 | ; | 7 | 0x06 | VCR/TV |
| 799 | | RETLW | cmd_digit_8 | ; | 8 | 0x07 | VCR/TV |
| 800 | | RETLW | cmd_digit_9 | ; | 9 | 0x08 | VCR/TV |
| 801 | | RETLW | cmd_digit_0 | ; | 0 | 0x09 | VCR/TV |
| 802 | | RETLW | 0X00 | ; | | 0x0A | |
| 803 | | RETLW | cmd_apply | ; | ENTER | 0x0B | VCR/TV |
| 804 | | RETLW | 0X00 | ; | | 0x0C | |
| 805 | | RETLW | 0X00 | ; | | 0x0D | |
| 806 | | RETLW | 0X00 | ; | | 0x0E | |
| 807 | | RETLW | 0X00 | ; | | 0x0F | |
| 808 | | RETLW | cmd_upper_unit | ; | CH+ | | 0x10 VCR/TV |
| 809 | | RETLW | cmd_lower_unit | ; | CH- | | 0x11 VCR/TV |
| 810 | | RETLW | cmd_upper | ; | VOL+ | 0x12 | TV |
| 811 | | RETLW | cmd_lower | ; | VOL- | 0x13 | TV |
| 812 | | RETLW | 0X00 | ; | | 0x14 | |
| 813 | | RETLW | cmd_manual | ; | POWER | 0x15 | VCR/TV |
| 814 | | RETLW | 0X00 | ; | | 0x16 | |
| 815 | | RETLW | 0X00 | ; | AUDIO MONITOR | 0x17 | VCR/TV |
| 816 | | RETLW | 0X00 | ; | | 0x18 | |
| 817 | | RETLW | 0X00 | ; | | 0x19 | |
| 818 | | RETLW | 0X00 | ; | | 0x1A | |
| 819 | | RETLW | 0X00 | ; | | 0x1B | |
| 820 | | RETLW | 0X00 | ; | | 0x1C | |
| 821 | | RETLW | 0X00 | ; | | 0x1D | |
| 822 | | RETLW | 0X00 | ; | | 0x1E | |
| 823 | | RETLW | 0X00 | ; | | 0x1F | |
| 824 | | RETLW | 0X00 | ; | | 0x20 | |
| 825 | | RETLW | 0X00 | ; | | 0x21 | |
| 826 | | RETLW | 0X00 | ; | | 0x22 | |
| 827 | | RETLW | 0X00 | ; | | 0x23 | |
| 828 | | RETLW | 0X00 | ; | | 0x24 | |
| 829 | | RETLW | 0X00 | ; | TV/VIDEO | 0x25 | TV |
| 830 | | RETLW | 0X00 | ; | | 0x26 | |

| | | | | | | | |
|-----|---------------------------------|---------------|---|---|------|------|--------|
| 831 | RETLW | 0X00 | ; | | 0x27 | | |
| 832 | RETLW | 0X00 | ; | | 0x28 | | |
| 833 | RETLW | 0X00 | ; | | 0x29 | | |
| 834 | RETLW | 0X00 | ; | | 0x2A | | |
| 835 | RETLW | 0X00 | ; | | 0x2B | | |
| 836 | RETLW | 0X00 | ; | | 0x2C | | |
| 837 | RETLW | 0X00 | ; | | 0x2D | | |
| 838 | RETLW | 0X00 | ; | | 0x2E | | |
| 839 | RETLW | 0X00 | ; | | 0x2F | | |
| 840 | RETLW | 0X00 | ; | | 0x30 | | |
| 841 | RETLW | 0X00 | ; | | 0x31 | | |
| 842 | RETLW | 0X00 | ; | | 0x32 | | |
| 843 | RETLW | 0X00 | ; | | 0x33 | | |
| 844 | RETLW | 0X00 | ; | | 0x34 | | |
| 845 | RETLW | 0X00 | ; | | 0x35 | | |
| 846 | RETLW | 0X00 | ; | | 0x36 | | |
| 847 | RETLW | 0X00 | ; | | 0x37 | | |
| 848 | RETLW | 0X00 | ; | | 0x38 | | |
| 849 | RETLW | 0X00 | ; | | 0x39 | | |
| 850 | RETLW | cmd_display | ; | DISPLAY | | 0x3A | TV |
| 851 | RETLW | 0X00 | ; | | 0x3B | | |
| 852 | RETLW | 0X00 | ; | | 0x3C | | |
| 853 | RETLW | 0X00 | ; | | 0x3D | | |
| 854 | RETLW | 0X00 | ; | | 0x3E | | |
| 855 | RETLW | 0X00 | ; | | 0x3F | | |
| 856 | | | | | | | |
| 857 | REMOTE.CONTROL.COMMANDS.MANUAL: | | | | | | |
| 858 | CLRF | PCLATH | | | | | |
| 859 | BSF | PCLATH, 1 | ; | ACERTA O DESLOCAMENTO SOMADO AO PCL | | | |
| 860 | ANDLW | 0X3F | ; | remove possíveis saltos para fora da tabela | | | |
| 861 | ADDWF | PCL, F | ; | move o PC para a posição desejada | | | |
| 862 | RETLW | 0X00 | ; | 1 | 0x00 | | VCR/TV |
| 863 | RETLW | 0X00 | ; | 2 | 0x01 | | VCR/TV |
| 864 | RETLW | 0X00 | ; | 3 | 0x02 | | VCR/TV |
| 865 | RETLW | 0X00 | ; | 4 | 0x03 | | VCR/TV |
| 866 | RETLW | 0X00 | ; | 5 | 0x04 | | VCR/TV |
| 867 | RETLW | 0X00 | ; | 6 | 0x05 | | VCR/TV |
| 868 | RETLW | 0X00 | ; | 7 | 0x06 | | VCR/TV |
| 869 | RETLW | 0X00 | ; | 8 | 0x07 | | VCR/TV |
| 870 | RETLW | 0X00 | ; | 9 | 0x08 | | VCR/TV |
| 871 | RETLW | 0X00 | ; | 0 | 0x09 | | VCR/TV |
| 872 | RETLW | 0X00 | ; | | 0x0A | | |
| 873 | RETLW | 0X00 | ; | ENTER | | 0x0B | VCR/TV |
| 874 | RETLW | 0X00 | ; | | 0x0C | | |
| 875 | RETLW | 0X00 | ; | | 0x0D | | |
| 876 | RETLW | 0X00 | ; | | 0x0E | | |
| 877 | RETLW | 0X00 | ; | | 0x0F | | |
| 878 | RETLW | cmd_open_all | ; | CH+ | | 0x10 | VCR/TV |
| 879 | RETLW | cmd_close_all | ; | CH- | | 0x11 | VCR/TV |
| 880 | RETLW | cmd_open | ; | VOL+ | | 0x12 | TV |
| 881 | RETLW | cmd_close | ; | VOL- | | 0x13 | TV |
| 882 | RETLW | 0X00 | ; | | 0x14 | | |
| 883 | RETLW | cmd_auto | ; | POWER | | 0x15 | VCR/TV |
| 884 | RETLW | 0X00 | ; | | 0x16 | | |
| 885 | RETLW | 0X00 | ; | AUDIO MONITOR | | 0x17 | VCR/TV |
| 886 | RETLW | 0X00 | ; | | 0x18 | | |

| | | | | | | | |
|-----|--|-------------|---|----------|------|------|----|
| 887 | RETLW | 0X00 | ; | | 0x19 | | |
| 888 | RETLW | 0X00 | ; | | 0x1A | | |
| 889 | RETLW | 0X00 | ; | | 0x1B | | |
| 890 | RETLW | 0X00 | ; | | 0x1C | | |
| 891 | RETLW | 0X00 | ; | | 0x1D | | |
| 892 | RETLW | 0X00 | ; | | 0x1E | | |
| 893 | RETLW | 0X00 | ; | | 0x1F | | |
| 894 | RETLW | 0X00 | ; | | 0x20 | | |
| 895 | RETLW | 0X00 | ; | | 0x21 | | |
| 896 | RETLW | 0X00 | ; | | 0x22 | | |
| 897 | RETLW | 0X00 | ; | | 0x23 | | |
| 898 | RETLW | 0X00 | ; | | 0x24 | | |
| 899 | RETLW | 0X00 | ; | TV/VIDEO | | 0x25 | TV |
| 900 | RETLW | 0X00 | ; | | 0x26 | | |
| 901 | RETLW | 0X00 | ; | | 0x27 | | |
| 902 | RETLW | 0X00 | ; | | 0x28 | | |
| 903 | RETLW | 0X00 | ; | | 0x29 | | |
| 904 | RETLW | 0X00 | ; | | 0x2A | | |
| 905 | RETLW | 0X00 | ; | | 0x2B | | |
| 906 | RETLW | 0X00 | ; | | 0x2C | | |
| 907 | RETLW | 0X00 | ; | | 0x2D | | |
| 908 | RETLW | 0X00 | ; | | 0x2E | | |
| 909 | RETLW | 0X00 | ; | | 0x2F | | |
| 910 | RETLW | 0X00 | ; | | 0x30 | | |
| 911 | RETLW | 0X00 | ; | | 0x31 | | |
| 912 | RETLW | 0X00 | ; | | 0x32 | | |
| 913 | RETLW | 0X00 | ; | | 0x33 | | |
| 914 | RETLW | 0X00 | ; | | 0x34 | | |
| 915 | RETLW | 0X00 | ; | | 0x35 | | |
| 916 | RETLW | 0X00 | ; | | 0x36 | | |
| 917 | RETLW | 0X00 | ; | | 0x37 | | |
| 918 | RETLW | 0X00 | ; | | 0x38 | | |
| 919 | RETLW | 0X00 | ; | | 0x39 | | |
| 920 | RETLW | cmd_display | ; | DISPLAY | | 0x3A | TV |
| 921 | RETLW | 0X00 | ; | | 0x3B | | |
| 922 | RETLW | 0X00 | ; | | 0x3C | | |
| 923 | RETLW | 0X00 | ; | | 0x3D | | |
| 924 | RETLW | 0X00 | ; | | 0x3E | | |
| 925 | RETLW | 0X00 | ; | | 0x3F | | |
| 926 | | | | | | | |
| 927 | | | | | | | |
| 928 | | | | | | | |
| 929 | ;TABELA PARA O DISPLAY DE SETE SEGMENTOS | | | | | | |
| 930 | ; | | | | | | |
| 931 | ; Lógica negada | | | | | | |
| 932 | ; | | | | | | |
| 933 | ; a | | | | | | |
| 934 | ; f b | | | | | | |
| 935 | ; g | | | | | | |
| 936 | ; e c | | | | | | |
| 937 | ; d | | | | | | |
| 938 | ; | | | | | | |
| 939 | ; byte = (dot) abcdefg | | | | | | |
| 940 | | | | | | | |
| 941 | SEVEN_SEG_CODE_LOOKUP: | | | | | | |
| 942 | CLRF | PCLATH | | | | | |

```

943     BSF PCLATH, 1
944     ANDLW 0x0F
945     ADDWF PCL, F
946     RETLW b'10000001'      ;0
947     RETLW b'11001111'      ;1
948     RETLW b'10010010'      ;2
949     RETLW b'10000110'      ;3
950     RETLW b'11001100'      ;4
951     RETLW b'10100100'      ;5
952     RETLW b'10100000'      ;6
953     RETLW b'10001111'      ;7
954     RETLW b'10000000'      ;8
955     RETLW b'10000100'      ;9
956     RETLW b'10001000'      ;a
957     RETLW b'11100000'      ;b
958     RETLW b'10110001'      ;c
959     RETLW b'11000010'      ;d
960     RETLW b'10110000'      ;e
961     RETLW b'10111000'      ;f
962
963 ; tabela com valores a serem somados no conversor bcd -> binário
964 BCD_TO_BINARY_DOZENS:
965     MOVWF w_temp
966     SUBLW 0X09
967     BTFSC STATUS, C
968     RETLW 0xFF
969     BSF PCLATH, 1
970     ADDWF PCL, F
971     RETLW 0x00      ; 0
972     RETLW 0x0A      ; 10
973     RETLW 0x14      ; 20
974     RETLW 0x1E      ; 30
975     RETLW 0x28      ; 40
976     RETLW 0x32      ; 50
977     RETLW 0x3C      ; 60
978     RETLW 0x46      ; 70
979     RETLW 0x50      ; 80
980     RETLW 0x5A      ; 90
981
982 ;
983 ;
984 ; Funções que devem ser chamadas alterando o pcl, não devem ser chamadas com
985 ; CALL, ver limite de endereço ao gerar o binário para se necessário ajustar o
986 ; PCLATCH
987 ;
988 ;
989 ;
990     org 0x0300
991 COMMANDS:
992     BCF SMCON, ST_COMMAND
993     MOVLW 0X03
994     MOVWF PCLATH
995     MOVF com_routine, W
996     MOVWF PCL
997
998 cmd_upper_unit

```

```

999      INCF      pos_value , W
1000     BTFSC     SMCON, ST_WRITE_CHANGE
1001     INCF      B1_temp , W
1002     goto      cmd_unit
1003 cmd_lower_unit
1004     DECF      pos_value , W
1005     BTFSC     SMCON, ST_WRITE_CHANGE
1006     DECF      B1_temp , W
1007 cmd_unit
1008     ANDLW     0x0F
1009     MOVWF     w_temp
1010
1011     MOVLW     0xF0
1012
1013     BTFSS     SMCON, ST_WRITE_CHANGE
1014     ANDWF     pos_value , W
1015     BTFSC     SMCON, ST_WRITE_CHANGE
1016     ANDWF     B1_temp , W
1017
1018     IORWF     w_temp , W
1019     MOVWF     B1_temp
1020     MOVWF     B1
1021     CALL      SEND_DISPLAY
1022     BSF       SMCON, ST_WRITE_CHANGE
1023     RETURN
1024
1025 cmd_digit_0
1026     MOVLW     0x00
1027     goto      cmd_digit
1028 cmd_digit_1
1029     MOVLW     0x10
1030     goto      cmd_digit
1031 cmd_digit_2
1032     MOVLW     0x20
1033     goto      cmd_digit
1034 cmd_digit_3
1035     MOVLW     0x30
1036     goto      cmd_digit
1037 cmd_digit_4
1038     MOVLW     0x40
1039     goto      cmd_digit
1040 cmd_digit_5
1041     MOVLW     0x50
1042     goto      cmd_digit
1043 cmd_digit_6
1044     MOVLW     0x60
1045     goto      cmd_digit
1046 cmd_digit_7
1047     MOVLW     0x70
1048     goto      cmd_digit
1049 cmd_digit_8
1050     MOVLW     0x80
1051     goto      cmd_digit
1052 cmd_digit_9
1053     MOVLW     0x90
1054

```

```

1055 cmd_digit
1056     MOVWF    w_temp
1057     MOVF     B1_temp, W
1058
1059     BTFSS    SMCON, ST_WRITE_CHANGE ; testa se já houve modificação anterior
1060     MOVF     pos_value, W
1061
1062 CONTINUE_DIGIT
1063     ANDLW    0x0F
1064     IORWF    w_temp, F
1065     SWAPF    w_temp, W
1066     MOVWF    B1_temp
1067     MOVWF    B1
1068     CALL     SEND_DISPLAY
1069     BSF      SMCON, ST_WRITE_CHANGE ; ativa dado a ser gravado
1070     RETURN
1071
1072 cmd_apply
1073     ; tratar conversão bcd
1074     BTFSS    SMCON, ST_WRITE_CHANGE ; ativo se há mudança a ser salva
1075     RETURN
1076     MOVF     B1_temp, W
1077     MOVWF    pos_value
1078     goto     cmd_change_record
1079 cmd_upper
1080     INCF     pos_value, W
1081     BTFSS    STATUS, Z
1082     MOVWF    pos_value
1083     goto     cmd_change_record
1084 cmd_lower
1085     MOVF     pos_value, F
1086     BTFSS    STATUS, Z
1087     DECF     pos_value, F
1088 cmd_change_record
1089     ; gravar na eeprom
1090     BCF      SMCON, ST_WRITE_CHANGE ; informa que não há nada a ser gravado
1091     CLRF     B1_temp ; descarta temporarios
1092     CLRF     B2_temp
1093     goto     cmd_display_show
1094
1095 cmd_display
1096     BTFSS    DISPLAYp, ENA_DISP
1097     goto     $+3
1098     BCF      DISPLAYp, ENA_DISP
1099     return
1100
1101     MOVLW    b'00000100'
1102     XORWF    SMCON, F
1103
1104     BTFSS    SMCON, ST_DISP_SET
1105     BSF      DISPLAYp, ENA_DISP
1106
1107     BTFSC    DISPLAYp, ENA_DISP
1108     RETURN ; se ação foi desligar o display não faz mais nada
1109 cmd_display_show
1110     ; converter valor especificado para bcd

```

```

1111         MOVF      pos_value , W
1112         MOVWF     B1
1113         CLRF      B2
1114         CALL      SEND_DISPLAY
1115         RETURN
1116
1117 cmd_manual
1118         BCF       SMCON, ST_AUTO
1119         RETURN
1120
1121 ;*****
1122 ;*
1123 ;* Comandos Manuais
1124 ;*
1125 ;*****
1126
1127 ; para o movimento ficar suave manter o tempo total desta execucao em 4ms
1128 cmd_open_all
1129         BSF       SMCON, ST_MOVE_ALL
1130 cmd_open
1131         BTFSS     PORTA, RA0
1132         BCF       SMCON, ST_MOVE_ALL
1133
1134         call      MOTOR_DIRECT
1135
1136         call      DELAYms
1137         call      DELAYms
1138         call      DELAYms
1139         call      DELAYms
1140
1141         BTFSC     SMCON, ST_MOVE_ALL
1142         GOTO      cmd_open
1143
1144         ; enquanto não terminar os tres envios fica em loop
1145         MOVF      IrDA_discard , F
1146         btfss     STATUS, Z
1147         goto      cmd_open
1148
1149         RETURN
1150
1151 ; para o movimento ficar suave manter o tempo total desta execucao em 4ms
1152 cmd_close_all
1153         BSF       SMCON, ST_MOVE_ALL
1154 cmd_close
1155         BTFSS     PORTA, RA1
1156         BCF       SMCON, ST_MOVE_ALL
1157
1158         call      MOTOR_REVERSE
1159
1160         call      DELAYms
1161         call      DELAYms
1162         call      DELAYms
1163         call      DELAYms
1164
1165         BTFSC     SMCON, ST_MOVE_ALL
1166         GOTO      cmd_close

```



```

1167
1168         ; enquanto não terminar os tres envios do controle fica em loop
1169     MOVF    IrDA_discard , F
1170     btfss  STATUS, Z
1171     goto   cmd_close
1172
1173     RETURN
1174
1175 cmd_auto
1176     BSF     SMCON, ST_AUTO           ; coloca em modo automático
1177     RETURN
1178
1179
1180     END                               ; directive 'end of program'

```

4. Considerações finais

A escolha da ideia do projeto nos permitiu explorar uma área considerável de problemas/soluções de projetos eletrônicos microcontrolados. Tivemos a oportunidade de trabalhar, testar e refinar nossa solução ao longo do projeto, o que trouxe um resultado muito interessante ao final.

As questões de configuração do PIC, apesar de um tanto complicadas no começo, aos poucos foram sendo resolvidas. Foi muito interessante ver como o projeto se desenvolveu enquanto agregávamos ideias que permitiram melhor configurabilidade do dispositivo pelo usuário, e isso motivou a ideia de se criar o cenário de exemplo para teste da solução obtida. Aos poucos pudemos nos acostumar com o estilo de programação do dispositivo, que é bem mais restritivo do que aquele ao qual estávamos habituados do microcontrolador 8051 da Intel.

Talvez um próximo passo possa ser integrar um sistema semelhante a um cenário real, onde os princípios absorvidos com o desenvolvimento de nosso sistema possa ser útil (tanto que utilizamos um problema real de um componente do grupo como motivação para a escolha do projeto). Outros itens que podem ser considerados em iterações futuras podem envolver, por exemplo, a integração de controle de tempo no sistema (tanto cronológico quanto meteorológico). Isso poderia permitir um ajuste do grau de luminosidade desejado pelo usuário de acordo com o horário e as condições do tempo medidos pelo PIC.

Referências

SONY SIRC infrared protocol. Disponível em: <http://picprojects.org.uk/projects/sirc/sonysirc.pdf>.

MICROCHIP. 16F684 Data Sheet. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/41202F-print.pdf>.

PARA LIGHT ELECTRONICS CO. 0.56 INCH DUAL DIGITS DISPLAY. Disponível em: <http://www.paralight.us/uploads/pdf/A-562G.pdf>.

STMicroelectronics. L293D, L293DD - PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES. Disponível em: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000059.pdf>.

STMicroelectronics. **M74HC595 - 8-bit shift register with output latches (3-state).**

Disponível em: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000339.pdf>.

Vishay Semiconductor GmbH. **TSOP18.** Disponível em: <http://pdf1.alldatasheet.com/datasheet-pdf/view/26603/VISHAY/TSOP1836.html>.