LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 1

ASSEMBLER CODE PACKING OF MODULE: CC6 (MAIN)

----- FUNCTION T2\_vInit (BEGIN) -----

FILE: 'T2.C'

130: void T2\_vInit(void)

131: {

132: // USER CODE BEGIN (T2\_Init,2)

133:

134: // USER CODE END

135:

136: // -----------------------------------------------------------------------

137: // Operating Mode

138: // -----------------------------------------------------------------------

139:

140: /// 16-bit timer function with automatic reload when timer 2 overflows

141: /// Prescaler enabled - input clock = fPCLK/16

142: /// the timer 2 resolution is 0,766 µs

143: /// the timer 2 overflow is 50197,787 µs

144: /// timer 2 interrupt: enabled

145: /// timer 2 will be started

146:

147: // -----------------------------------------------------------------------

148: // Register Initialization

149: // -----------------------------------------------------------------------

150:

151: // the reset values of RC2H and RC2L are not changed

152: T2\_T2MOD = 0x18; // load timer 2 mode register

000003 75C118 MOV T2\_T2MOD,#018H

153:

154: ET2 = 1; // Enable interrupt

000006 D2AD SETB ET2

155:

156: // USER CODE BEGIN (T2\_Init,3)

157:

158: // USER CODE END

159:

160: TR2 = 1; // start timer

000008 D2C2 SETB TR2

161:

162: } // End of function T2\_vInit

00000A 22 RET

----- FUNCTION T2\_vInit (END) -------

00000B 0200E6 LJMP 0E6H

00001B 02011D LJMP 011DH

----- FUNCTION \_SSC\_vSendData (BEGIN) -----

FILE: 'SSC.c'

228: void SSC\_vSendData(ubyte Data)

229: {

;---- Variable 'Data' assigned to Register 'R7' ----

230: SSC\_TBL = Data; // load transmit buffer register

231:

00001E 8FAC MOV SSC\_TBL,R7

232: } // End of function SSC\_vSendData

000020 22 RET

----- FUNCTION \_SSC\_vSendData (END) -------

00002B 02013A LJMP 013AH

00003B 020296 LJMP 0296H

----- FUNCTION T01\_vInit (BEGIN) -----

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 2

FILE: 'T01.C'

132: void T01\_vInit(void)

133: {

134: // USER CODE BEGIN (T01\_Init,2)

135:

136: // USER CODE END

137:

138: /// -----------------------------------------------------------------------

139: /// Timer 0

140: /// -----------------------------------------------------------------------

141: /// Mode 0: 8-bit timer (TH0) with a divide-by-32 prescaler (TL0)

142: /// the timer 0 overflow is 65,362 µs

143:

144:

145: /// -----------------------------------------------------------------------

146: /// Timer 1

147: /// -----------------------------------------------------------------------

148: /// Mode 0: 8-bit timer (TH1) with a divide-by-32 prescaler (TL1)

149: /// the timer 1 overflow is 65,362 µs

150:

151:

152: TMOD = 0x00;

00003E E4 CLR A

00003F F589 MOV TMOD,A

153:

154: TL0 = 0x00;

000041 F58A MOV TL0,A

155: TH0 = 0x00;

000043 F58C MOV TH0,A

156:

157: TL1 = 0x00;

000045 F58B MOV TL1,A

158: TH1 = 0x00;

000047 F58D MOV TH1,A

159:

160: /// Enable Timer 0 interrupt

161: ET0 = 1;

000049 D2A9 SETB ET0

162: /// Enable Timer 1 interrupt

163: ET1 = 1;

00004B D2AB SETB ET1

164:

165: // USER CODE BEGIN (T01\_Init,3)

166:

167: // USER CODE END

168:

169: /// Start Timer 0

170: TR0 = 1;

00004D D28C SETB TR0

171: /// Start Timer 1

172: TR1 = 1;

00004F D28E SETB TR1

173: } // End of function T01\_vInit

000051 22 RET

----- FUNCTION T01\_vInit (END) -------

000053 0201F1 LJMP 01F1H

----- FUNCTION MAIN\_vInit (BEGIN) -----

FILE: 'MAIN.C'

121: void MAIN\_vInit(void)

122: {

123: // USER CODE BEGIN (MAIN\_Init,2)

124:

125: // USER CODE END

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 3

126:

127: /// -----------------------------------------------------------------------

128: /// Configuration of the System Clock:

129: /// -----------------------------------------------------------------------

130: /// - On Chip Osc is Selected

131: /// - PLL Mode, NDIV = 2

132: /// - input frequency is 9.4 MHz

133:

134:

135:

136: /// Initialization of module 'GPIO'

137: IO\_vInit();

000056 12008E LCALL IO\_vInit

138:

139: /// Initialization of module 'Timer 0/1'

140: T01\_vInit();

000059 12003E LCALL T01\_vInit

141:

142: /// Initialization of module 'Timer 2'

143: T2\_vInit();

00005C 120003 LCALL T2\_vInit

144:

145: /// Initialization of module 'Capture / Compare Unit 6 (CCU6)'

146: CC6\_vInit();

00005F 120166 LCALL CC6\_vInit

147:

148: SSC\_vInit ();

000062 120262 LCALL SSC\_vInit

149:

150: // Interrupt Priority

151:

152: IP = 0x00; // load Interrupt Priority Register

000065 E4 CLR A

000066 F5B8 MOV IP,A

153: IPH = 0x00; // load Interrupt Priority High Register

000068 F5B9 MOV IPH,A

154: IP1 = 0x00; // load Interrupt Priority 1 Register

00006A F5F8 MOV IP1,A

155: IPH1 = 0x00; // load Interrupt Priority 1 High Register

00006C F5F9 MOV IPH1,A

156:

157:

158: // USER CODE BEGIN (MAIN\_Init,3)

159:

160: // USER CODE END

161:

162: // globally enable interrupts

163: EA = 1;

00006E D2AF SETB EA

164:

165: } // End of function MAIN\_vInit

000070 22 RET

----- FUNCTION MAIN\_vInit (END) -------

----- FUNCTION main (BEGIN) -----

FILE: 'MAIN.C'

189: void main(void)

190: {

191: // USER CODE BEGIN (MAIN\_Main,2)

192: unsigned int toto;

193: // USER CODE END

194:

195: MAIN\_vInit();

000071 120056 LCALL MAIN\_vInit

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 4

196:

197:

198: // USER CODE BEGIN (MAIN\_Main,3)

199: P3\_DATA = 0x00;

000074 E4 CLR A

000075 F5B0 MOV P3\_DATA,A

000077 ?C0002?MAIN:

200: while (1) {

201: for (toto=0x8000; toto>0; toto--);

;---- Variable 'toto' assigned to Register 'R6/R7' ----

000077 7F00 MOV R7,#00H

000079 7E80 MOV R6,#080H

00007B ?C0004?MAIN:

00007B EF MOV A,R7

00007C 1F DEC R7

00007D 7001 JNZ ?C0008?MAIN

00007F 1E DEC R6

000080 ?C0008?MAIN:

000080 EF MOV A,R7

000081 4E ORL A,R6

000082 70F7 JNZ ?C0004?MAIN

000084 ?C0005?MAIN:

202: P3\_DATA ^= 0x01;

000084 63B001 XRL P3\_DATA,#01H

203: SSC\_vSendData(P3\_DATA);

000087 AFB0 MOV R7,P3\_DATA

000089 12001E LCALL \_SSC\_vSendData

204: }

00008C 80E9 SJMP ?C0002?MAIN

----- FUNCTION main (END) -------

----- FUNCTION IO\_vInit (BEGIN) -----

FILE: 'IO.C'

122: void IO\_vInit(void)

123: {

124: // USER CODE BEGIN (IO\_Init,2)

125:

126: // USER CODE END

127:

128: /// -----------------------------------------------------------------------

129: /// Configuration of Port P0:

130: /// -----------------------------------------------------------------------

131: /// - no pin of port P0 is used

132:

133: SFR\_PAGE(\_pp3, SST2); // switch to page 3

00008E 75B2A3 MOV PORT\_PAGE,#0A3H

134: P0\_OD = 0x00; // load open-drain register

000091 E4 CLR A

000092 F580 MOV P0\_OD,A

135:

136: SFR\_PAGE(\_pp1, SST2); // switch to page 1

000094 75B2A1 MOV PORT\_PAGE,#0A1H

137: P0\_PUDSEL = 0xFF; // load pullup/pulldown select register

000097 7580FF MOV P0\_PUDSEL,#0FFH

138: P0\_PUDEN = 0xC4; // load pullup/pulldown enable register

00009A 7586C4 MOV P0\_PUDEN,#0C4H

139:

140: SFR\_PAGE(\_pp2, SST2); // switch to page 2

00009D 75B2A2 MOV PORT\_PAGE,#0A2H

141: P0\_ALTSEL0 = 0x00; // load alternate output function select

0000A0 F580 MOV P0\_ALTSEL0,A

142: // register 0

143: P0\_ALTSEL1 = 0x00; // load alternate output function select

0000A2 F586 MOV P0\_ALTSEL1,A

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 5

144: // register 1

145:

146: SFR\_PAGE(\_pp0, SST2); // switch to page 0

0000A4 75B2A0 MOV PORT\_PAGE,#0A0H

147: P0\_DATA = 0x00; // load data output register

0000A7 F580 MOV P0\_DATA,A

148: P0\_DIR = 0x00; // load direction register

0000A9 F586 MOV P0\_DIR,A

149:

150: /// -----------------------------------------------------------------------

151: /// Configuration of Port P1:

152: /// -----------------------------------------------------------------------

153: /// - no pin of port P1 is used

154:

155: SFR\_PAGE(\_pp3, SST2); // switch to page 3

0000AB 75B2A3 MOV PORT\_PAGE,#0A3H

156: P1\_OD = 0x00; // load open-drain register

0000AE F590 MOV P1\_OD,A

157:

158: SFR\_PAGE(\_pp1, SST2); // switch to page 1

0000B0 75B2A1 MOV PORT\_PAGE,#0A1H

159: P1\_PUDSEL = 0xFF; // load pullup/pulldown select register

0000B3 7590FF MOV P1\_PUDSEL,#0FFH

160: P1\_PUDEN = 0xFF; // load pullup/pulldown enable register

0000B6 7591FF MOV P1\_PUDEN,#0FFH

161:

162: SFR\_PAGE(\_pp2, SST2); // switch to page 2

0000B9 75B2A2 MOV PORT\_PAGE,#0A2H

163: P1\_ALTSEL0 = 0x00; // load alternate output function select

0000BC F590 MOV P1\_ALTSEL0,A

164: // register 0

165: P1\_ALTSEL1 = 0x00; // load alternate output function select

0000BE F591 MOV P1\_ALTSEL1,A

166: // register 1

167:

168: SFR\_PAGE(\_pp0, SST2); // switch to page 0

0000C0 75B2A0 MOV PORT\_PAGE,#0A0H

169: P1\_DATA = 0x00; // load data output register

0000C3 F590 MOV P1\_DATA,A

170: P1\_DIR = 0x00; // load direction register

0000C5 F591 MOV P1\_DIR,A

171:

172: /// -----------------------------------------------------------------------

173: /// Configuration of Port P2:

174: /// -----------------------------------------------------------------------

175: /// - no pin of port P2 is used

176:

177:

178: SFR\_PAGE(\_pp1, SST2); // switch to page 1

0000C7 75B2A1 MOV PORT\_PAGE,#0A1H

179: P2\_PUDSEL = 0xFF; // load pullup/pulldown select register

0000CA 75A0FF MOV P2\_PUDSEL,#0FFH

180: P2\_PUDEN = 0x00; // load pullup/pulldown enable register

0000CD F5A1 MOV P2\_PUDEN,A

181:

182: SFR\_PAGE(\_pp0, SST2); // switch to page 0

0000CF 75B2A0 MOV PORT\_PAGE,#0A0H

183: P2\_DATA = 0x00; // load data output register

0000D2 F5A0 MOV P2\_DATA,A

184:

185: /// -----------------------------------------------------------------------

186: /// Configuration of Port P3:

187: /// -----------------------------------------------------------------------

188: /// P3.0:

189: /// - is used as general purpose output

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 6

190: /// - open drain output is selected

191: /// - the pin status is low level

192: /// - pull-up device is assigned

193: /// P3.1:

194: /// - is used as general purpose output

195: /// - open drain output is selected

196: /// - the pin status is low level

197: /// - pull-up device is assigned

198: /// P3.2:

199: /// - is used as general purpose output

200: /// - open drain output is selected

201: /// - the pin status is low level

202: /// - pull-up device is assigned

203: /// P3.3:

204: /// - is used as general purpose output

205: /// - open drain output is selected

206: /// - the pin status is low level

207: /// - pull-up device is assigned

208: /// P3.4:

209: /// - is used as general purpose output

210: /// - open drain output is selected

211: /// - the pin status is low level

212: /// - pull-up device is assigned

213: /// P3.5:

214: /// - is used as general purpose output

215: /// - open drain output is selected

216: /// - the pin status is low level

217: /// - pull-up device is assigned

218: /// P3.6:

219: /// - is used as general purpose output

220: /// - open drain output is selected

221: /// - the pin status is low level

222: /// - pull-up device is assigned

223: /// P3.7:

224: /// - is used as alternate output for the Output of Capture/Compare

225: /// channel 3

226: /// - push/pull output is selected

227: /// - pull-up device is assigned

228:

229: // SFR\_PAGE(\_pp3, SST2); // switch to page 3

230: // P3\_OD = 0x7F; // load open-drain register

231:

232: SFR\_PAGE(\_pp1, SST2); // switch to page 1

0000D4 75B2A1 MOV PORT\_PAGE,#0A1H

233: P3\_PUDSEL = 0xFF; // load pullup/pulldown select register

0000D7 75B0FF MOV P3\_PUDSEL,#0FFH

234: P3\_PUDEN = 0xFF; // load pullup/pulldown enable register

0000DA 75B1FF MOV P3\_PUDEN,#0FFH

235:

236: // SFR\_PAGE(\_pp2, SST2); // switch to page 2

237: // P3\_ALTSEL0 = 0x80; // load alternate output function select

238: // register 0

239: // P3\_ALTSEL1 = 0x00; // load alternate output function select

240: // register 1

241: // USER CODE BEGIN (IO\_Init,2)

242: // P3\_ALTSEL0 = 0x10; // P37 to ALT1-COUT63

243:

244: // USER CODE END

245:

246:

247:

248: SFR\_PAGE(\_pp0, SST2); // switch to page 0

0000DD 75B2A0 MOV PORT\_PAGE,#0A0H

249: P3\_DATA = 0x00; // load data output register

0000E0 F5B0 MOV P3\_DATA,A

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 7

250: P3\_DIR = 0xFF; // load direction register

0000E2 75B1FF MOV P3\_DIR,#0FFH

251:

252:

253: // USER CODE BEGIN (IO\_Init,3)

254:

255: // USER CODE END

256:

257: } // End of function IO\_vInit

0000E5 22 RET

----- FUNCTION IO\_vInit (END) -------

----- FUNCTION T01\_viTmr0 (BEGIN) -----

FILE: 'T01.C'

202: void T01\_viTmr0(void) interrupt T0INT

0000E6 C0E0 PUSH ACC

0000E8 C0D0 PUSH PSW

0000EA 75D000 MOV PSW,#00H

0000ED C006 PUSH AR6

203: {

204:

205: // USER CODE BEGIN (T01\_IsrTmr0,2)

206: if (T0count++ > 0x100) P3\_DATA ^= 0x02; // toggle every 256 T0

0000EF 050E INC T0count+01H

0000F1 E50E MOV A,T0count+01H

0000F3 AE0D MOV R6,T0count

0000F5 7002 JNZ ?C0007?T01

0000F7 050D INC T0count

0000F9 ?C0007?T01:

0000F9 14 DEC A

0000FA D3 SETB C

0000FB 9400 SUBB A,#00H

0000FD EE MOV A,R6

0000FE 9401 SUBB A,#01H

000100 4003 JC ?C0002?T01

000102 63B002 XRL P3\_DATA,#02H

000105 ?C0002?T01:

207: if (T0count > 0x200) T0count = 0;

000105 D3 SETB C

000106 E50E MOV A,T0count+01H

000108 9400 SUBB A,#00H

00010A E50D MOV A,T0count

00010C 9402 SUBB A,#02H

00010E 4006 JC ?C0004?T01

000110 750D00 MOV T0count,#00H

000113 750E00 MOV T0count+01H,#00H

208:

209: // USER CODE END

210:

211: } // End of function T01\_viTmr0

000116 ?C0004?T01:

000116 D006 POP AR6

000118 D0D0 POP PSW

00011A D0E0 POP ACC

00011C 32 RETI

----- FUNCTION T01\_viTmr0 (END) -------

----- FUNCTION T01\_viTmr1 (BEGIN) -----

FILE: 'T01.C'

243: void T01\_viTmr1(void) interrupt T1INT

00011D C0E0 PUSH ACC

00011F C0D0 PUSH PSW

000121 75D000 MOV PSW,#00H

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 8

000124 C007 PUSH AR7

244: {

245:

246: // USER CODE BEGIN (T01\_IsrTmr1,2)

247: if (T1count++ > 0x80) P3\_DATA ^= 0x02; // toggle for 128 T1count

000126 AF0F MOV R7,T1count

000128 050F INC T1count

00012A EF MOV A,R7

00012B D3 SETB C

00012C 9480 SUBB A,#080H

00012E 4003 JC ?C0006?T01

000130 63B002 XRL P3\_DATA,#02H

248:

249: // USER CODE END

250:

251: } // End of function T01\_viTmr1

000133 ?C0006?T01:

000133 D007 POP AR7

000135 D0D0 POP PSW

000137 D0E0 POP ACC

000139 32 RETI

----- FUNCTION T01\_viTmr1 (END) -------

----- FUNCTION T2\_viTmr (BEGIN) -----

FILE: 'T2.C'

192: void T2\_viTmr(void) interrupt T2INT

00013A C0E0 PUSH ACC

00013C C0D0 PUSH PSW

00013E 75D000 MOV PSW,#00H

000141 C007 PUSH AR7

193: {

194: // USER CODE BEGIN (T2\_IsrTmr,2)

195: ubyte T2count;

196: // USER CODE END

197:

198: if (TF2)

000143 30C719 JNB TF2,?C0005?T2

199: {

200: // a timer 2 overflow has occured

201: // USER CODE BEGIN (T2\_IsrTmr,3)

202: if (T2count++ > 0x02) P3\_DATA ^= 0x04; // toggle every 2nd T2OF

000146 AF10 MOV R7,T2count

000148 0510 INC T2count

00014A EF MOV A,R7

00014B D3 SETB C

00014C 9402 SUBB A,#02H

00014E 4003 JC ?C0003?T2

000150 63B004 XRL P3\_DATA,#04H

000153 ?C0003?T2:

203: if (T2count > 0x04) T2count = 0;

000153 E510 MOV A,T2count

000155 D3 SETB C

000156 9404 SUBB A,#04H

000158 4003 JC ?C0004?T2

00015A 751000 MOV T2count,#00H

00015D ?C0004?T2:

204:

205: // SFR\_PAGE(\_cc2, SST2); // switch to page 2

206: // CCU6\_PAGE = 0x02;

207: // CCU6\_ISSH |= 0x02; // set T13PM

208: // SFR\_PAGE(\_cc2, RST2); // restore the old CCU page

209:

210: // USER CODE END

211:

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 9

212: TF2 = 0;

00015D C2C7 CLR TF2

213: }

214:

215: // USER CODE BEGIN (T2\_IsrTmr,5)

216:

217: // USER CODE END

218:

219: } // End of function T2\_viTmr

00015F ?C0005?T2:

00015F D007 POP AR7

000161 D0D0 POP PSW

000163 D0E0 POP ACC

000165 32 RETI

----- FUNCTION T2\_viTmr (END) -------

----- FUNCTION CC6\_vInit (BEGIN) -----

FILE: 'CC6.C'

122: void CC6\_vInit(void)

123: {

124: // USER CODE BEGIN (CCU6\_General,9)

125:

126: // USER CODE END

127:

128:

129: SFR\_PAGE(\_cc1, SST2); // switch to page 1

000166 75A3A1 MOV CCU6\_PAGE,#0A1H

130:

131: //T12 period

132: CCU6\_T12PRL = 0x00; // load CCU6 T12 period register low

000169 E4 CLR A

00016A F59C MOV CCU6\_T12PRL,A

133: CCU6\_T12PRH = 0xF0; // load CCU6 T12 period register lhigh

00016C 759DF0 MOV CCU6\_T12PRH,#0F0H

134: //T13 period

135: CCU6\_T13PRL = 0xFF; // load CCU6 T13 period register low

00016F 759EFF MOV CCU6\_T13PRL,#0FFH

136: CCU6\_T13PRH = 0xFF; // load CCU6 T13 period register high

000172 759FFF MOV CCU6\_T13PRH,#0FFH

137: // T12 prescaler and center aligned

138: CCU6\_TCTR0L = 0x84; // load CCU6 timer control register 0 low

000175 75A684 MOV CCU6\_TCTR0L,#084H

139: // T13 prescaler

140: CCU6\_TCTR0H = 0x03; // load CCU6 timer control register 0 high

000178 75A703 MOV CCU6\_TCTR0H,#03H

141: //// SFR\_PAGE(\_cc1, SST2); // switch to page 1

142: // T12 dead time DTM

143: CCU6\_T12DTCL = 0x10; // load CCU6 dead time control register low

00017B 75A410 MOV CCU6\_T12DTCL,#010H

144: // T12 dead time channel select

145: CCU6\_T12DTCH = 0x04; // load CCU6 dead time control register high

00017E 75A504 MOV CCU6\_T12DTCH,#04H

146:

147: SFR\_PAGE(\_cc2, SST2); // switch to page 2

000181 75A3A2 MOV CCU6\_PAGE,#0A2H

148: // T12/T12 modes and external trigger

149: CCU6\_TCTR2L = 0x00; // load CCU6 timer control register 2 low

000184 F5FA MOV CCU6\_TCTR2L,A

150: CCU6\_TCTR2H = 0x00; // load CCU6 timer control register 2 high

000186 F5FB MOV CCU6\_TCTR2H,A

151: // T12 Modulation Enable & MCMEN

152: CCU6\_MODCTRL = 0x30; // load CCU6 modulation control register low

000188 75FC30 MOV CCU6\_MODCTRL,#030H

153: // T13 Modulation Enable

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 10

154: CCU6\_MODCTRH = 0x80; // load CCU6 modulation control register high

00018B 75FD80 MOV CCU6\_MODCTRH,#080H

155: /// SFR\_PAGE(\_cc2, SST2); // switch to page 2

156: // MultiChannel Synch & Sel

157: CCU6\_MCMCTR = 0x00; // load CCU6 multi channel mode control

00018E F5A7 MOV CCU6\_MCMCTR,A

158: // register

159: /// SFR\_PAGE(\_cc2, SST2); // switch to page 2

160: // T12 capcom mode ch0 / ch1

161: CCU6\_T12MSELL = 0x00; // load CCU6 T12 campture/compare mode

000190 F59A MOV CCU6\_T12MSELL,A

162: // T12 capcom mode ch2 / hallsynch

163: CCU6\_T12MSELH = 0x03; // load CCU6 T12 campture/compare mode

000192 759B03 MOV CCU6\_T12MSELH,#03H

164: ///SFR\_PAGE(\_cc2, SST2); // switch to page 2

165: // Passive state level

166: CCU6\_PSLR = 0x00; // load CCU6 passive state level register low

000195 F5A6 MOV CCU6\_PSLR,A

167:

168:

169: SFR\_PAGE(\_cc0, SST2); // switch to page 0

000197 75A3A0 MOV CCU6\_PAGE,#0A0H

170: // T12 shadow

171: CCU6\_CC62SRL = 0x00; // load CCU6 capture/compare shadow register

00019A F5FE MOV CCU6\_CC62SRL,A

172: CCU6\_CC62SRH = 0x78; // load CCU6 capture/compare shadow register

00019C 75FF78 MOV CCU6\_CC62SRH,#078H

173: // T13 shadow

174: CCU6\_CC63SRL = 0x00; // load CCU6 capture/compare shadow register

00019F F59A MOV CCU6\_CC63SRL,A

175: CCU6\_CC63SRH = 0x19; // load CCU6 capture/compare shadow register

0001A1 759B19 MOV CCU6\_CC63SRH,#019H

176: /// SFR\_PAGE(\_cc0, SST2); // switch to page 0

177: // MultiChannel PWM Pattern

178: CCU6\_MCMOUTSL = 0x00; // load CCU6 multi channel mode output

0001A4 F59E MOV CCU6\_MCMOUTSL,A

179: // MultiChannel Hall Pattern

180: CCU6\_MCMOUTSH = 0x00; // load CCU6 multi channel mode output

0001A6 F59F MOV CCU6\_MCMOUTSH,A

181: // control register high

182:

183: SFR\_PAGE(\_cc2, SST2); // switch to page 2

0001A8 75A3A2 MOV CCU6\_PAGE,#0A2H

184: // Trap Control Leave

185: CCU6\_TRPCTRL = 0x00; // load CCU6 trap control register low

0001AB F5FE MOV CCU6\_TRPCTRL,A

186: // Trap Control Enable

187: CCU6\_TRPCTRH = 0x00; // load CCU6 trap control register high

0001AD F5FF MOV CCU6\_TRPCTRH,A

188:

189: // Interrupt node pointer

190: CCU6\_INPL = 0x00; // load CCU6 capture/compare interrupt node

0001AF F59E MOV CCU6\_INPL,A

191: CCU6\_INPH = 0x00; // load CCU6 capture/compare interrupt node

0001B1 F59F MOV CCU6\_INPH,A

192:

193: // T12 interrupts enable

194: CCU6\_IENL = 0x80; // load CCU6 capture/compare interrupt

0001B3 759C80 MOV CCU6\_IENL,#080H

195: // T13 & others interrupts enable

196: CCU6\_IENH = 0x02; // load CCU6 capture/compare interrupt

0001B6 759D02 MOV CCU6\_IENH,#02H

197: // enable register high

198:

199: SFR\_PAGE(\_cc3, SST2); // switch to page 3

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 11

0001B9 75A3A3 MOV CCU6\_PAGE,#0A3H

200: // Compare Passive State Select

201: CCU6\_CMPSTATH = 0x20; // load CCU6 compare status register high

0001BC 75FF20 MOV CCU6\_CMPSTATH,#020H

202:

203: // input capture and trap input

204: CCU6\_PISEL0L = 0x00; // load CCU6 Port Input Select Register 0 Low

0001BF F59E MOV CCU6\_PISEL0L,A

205: // input CCPOS and T12HR

206: CCU6\_PISEL0H = 0x00; // load CCU6 Port Input Select Register 0

0001C1 F59F MOV CCU6\_PISEL0H,A

207: // input T13HR

208: CCU6\_PISEL2 = 0x00; // load CCU6 Port Input Select Register 2

0001C3 F5A4 MOV CCU6\_PISEL2,A

209: /// SFR\_PAGE(\_cc3, SST2); // switch to page 3

210:

211: /// Pin P3.4, P3.5, P3.7

212: PORT\_PAGE = 0x02; // switch to page 2

0001C5 75B202 MOV PORT\_PAGE,#02H

213: P3\_ALTSEL0 |= 0x10; // set AltSel0

0001C8 43B010 ORL P3\_ALTSEL0,#010H

214: P3\_ALTSEL1 &= ~(ubyte) 0x10; // set AltSel1

0001CB 53B1EF ANL P3\_ALTSEL1,#0EFH

215: P3\_ALTSEL0 |= 0x20; // set AltSel0

0001CE 43B020 ORL P3\_ALTSEL0,#020H

216: P3\_ALTSEL1 &= ~(ubyte) 0x20; // set AltSel1

0001D1 53B1DF ANL P3\_ALTSEL1,#0DFH

217: P3\_ALTSEL0 |= 0x80; // set AltSel0

0001D4 43B080 ORL P3\_ALTSEL0,#080H

218: P3\_ALTSEL1 &= ~(ubyte) 0x80; // set AltSel1

0001D7 53B17F ANL P3\_ALTSEL1,#07FH

219: PORT\_PAGE = 0x00; // switch to page 0

0001DA F5B2 MOV PORT\_PAGE,A

220: P3\_DIR |= 0x10; // set P3.4 OutPut

0001DC 43B110 ORL P3\_DIR,#010H

221: P3\_DIR |= 0x20; // set P3.5 OutPut

0001DF 43B120 ORL P3\_DIR,#020H

222: P3\_DIR |= 0x80; // set P3.7 OutPut

0001E2 43B180 ORL P3\_DIR,#080H

223:

224:

225: SFR\_PAGE(\_cc0, noSST); // switch to cc6 page 0 as working page

0001E5 F5A3 MOV CCU6\_PAGE,A

226: /// SFR\_PAGE(\_cc0, SST2); // switch to page 0

227: // T12 shadow transfer & run control

228: CCU6\_TCTR4L = 0x42; // enable Timer T12 shadow transfer

0001E7 759C42 MOV CCU6\_TCTR4L,#042H

229: // T13 shadow transfer & run control

230: CCU6\_TCTR4H = 0x42; // T13 run & enable Timer T13 shadow transfer

0001EA 759D42 MOV CCU6\_TCTR4H,#042H

231:

232: // node interrupt enable

233: IEN1 |= 0x10; // load interrupt enable register 2

0001ED 43E810 ORL IEN1,#010H

234:

235: // USER CODE END

236:

237: } // End of function CC6\_vInit

0001F0 22 RET

----- FUNCTION CC6\_vInit (END) -------

----- FUNCTION CC6\_viNodeI0 (BEGIN) -----

FILE: 'CC6.C'

268: void CC6\_viNodeI0(void) interrupt CCU6\_NodeI0\_INT

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 12

0001F1 C0E0 PUSH ACC

0001F3 C0D0 PUSH PSW

0001F5 75D000 MOV PSW,#00H

0001F8 C007 PUSH AR7

269: {

270:

271: // USER CODE BEGIN (NodeI0,2)

272: static ubyte T13count;

273: static uword T12CC2, T13CC3;

274: // USER CODE END

275:

276: SFR\_PAGE(\_cc3, SST2); // switch to page 3

0001FA 75A3A3 MOV CCU6\_PAGE,#0A3H

277:

278: if(CCU6\_ISL & 0x80) //if ISL\_T12PM

0001FD E59C MOV A,CCU6\_ISL

0001FF 30E722 JNB ACC.7,?C0002?CC6

279: {

280: //timer T12 period match detection

281:

282: SFR\_PAGE(\_cc0, SST3); // switch to page 0

000202 75A3B0 MOV CCU6\_PAGE,#0B0H

283:

284: CCU6\_ISRL = 0x80; //clear flag ISL\_T12PM

000205 75A480 MOV CCU6\_ISRL,#080H

285:

286: T12CC2 += 0x0100;

000208 E4 CLR A

000209 250A ADD A,T12CC2+01H

00020B F50A MOV T12CC2+01H,A

00020D 7401 MOV A,#01H

00020F 3509 ADDC A,T12CC2

000211 F509 MOV T12CC2,A

287: SFR\_PAGE(\_cc0, noSST); // restore the old CCU page

000213 75A300 MOV CCU6\_PAGE,#00H

288: CCU6\_CC62SRH = (ubyte) (T12CC2 >> 8); // Load Channel Shadow Register CC63

000216 E509 MOV A,T12CC2

000218 F5FF MOV CCU6\_CC62SRH,A

289: CCU6\_CC62SRL = (ubyte) (T12CC2 & 0x00FF); // Load Channel Shadow Register CC63

00021A E50A MOV A,T12CC2+01H

00021C F5FE MOV CCU6\_CC62SRL,A

290: CCU6\_TCTR4L |= 0x40; // enable Timer T13 shadow transfer

00021E 439C40 ORL CCU6\_TCTR4L,#040H

291:

292: // USER CODE BEGIN (NodeI0,12)

293: P3\_DATA ^= 0x40;

000221 63B040 XRL P3\_DATA,#040H

294: // USER CODE END

295: }

000224 ?C0002?CC6:

296:

297: SFR\_PAGE(\_cc3, SST2); // switch to page 3

000224 75A3A3 MOV CCU6\_PAGE,#0A3H

298:

299: if(CCU6\_ISH & 0x02) //if ISH\_T13PM

000227 E59D MOV A,CCU6\_ISH

000229 30E12F JNB ACC.1,?C0005?CC6

300: {

301: //timer T13 period match detection

302:

303: SFR\_PAGE(\_cc0, SST3); // switch to page 0

00022C 75A3B0 MOV CCU6\_PAGE,#0B0H

304:

305: CCU6\_ISRH = 0x02; //clear flag ISH\_T13PM

00022F 75A502 MOV CCU6\_ISRH,#02H

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 13

306:

307:

308: // USER CODE BEGIN (NodeI0,12)

309: T13CC3 += 0x0100;

000232 E4 CLR A

000233 250C ADD A,T13CC3+01H

000235 F50C MOV T13CC3+01H,A

000237 7401 MOV A,#01H

000239 350B ADDC A,T13CC3

00023B F50B MOV T13CC3,A

310: SFR\_PAGE(\_cc0, noSST); // restore the old CCU page

00023D 75A300 MOV CCU6\_PAGE,#00H

311: CCU6\_CC63SRH = (ubyte) (T13CC3 >> 8); // Load Channel Shadow Register CC63

000240 E50B MOV A,T13CC3

000242 F59B MOV CCU6\_CC63SRH,A

312: CCU6\_CC63SRL = (ubyte) (T13CC3 & 0x00FF); // Load Channel Shadow Register CC63

000244 E50C MOV A,T13CC3+01H

000246 F59A MOV CCU6\_CC63SRL,A

313: CCU6\_TCTR4H |= 0x40; // enable Timer T13 shadow transfer

000248 439D40 ORL CCU6\_TCTR4H,#040H

314:

315: if (T13count++ > 0x10) {

00024B AF08 MOV R7,T13count

00024D 0508 INC T13count

00024F EF MOV A,R7

000250 D3 SETB C

000251 9410 SUBB A,#010H

000253 4006 JC ?C0005?CC6

316: T13count = 0;

000255 750800 MOV T13count,#00H

317: P3\_DATA ^= 0x08;

000258 63B008 XRL P3\_DATA,#08H

318: }

319:

320: // USER CODE END

321:

322: }

323:

324:

325: } // End of function CC6\_viNodeI0

00025B ?C0005?CC6:

00025B D007 POP AR7

00025D D0D0 POP PSW

00025F D0E0 POP ACC

000261 32 RETI

----- FUNCTION CC6\_viNodeI0 (END) -------

----- FUNCTION SSC\_vInit (BEGIN) -----

FILE: 'SSC.c'

122: void SSC\_vInit(void)

123: {

124: // USER CODE BEGIN (Init,2)

125:

126: // USER CODE END

127:

128: SSC\_CONH\_P &= ~(ubyte)0x80; // enable access to control bits

000262 53AB7F ANL SSC\_CONH\_P,#07FH

129:

130: /// -----------------------------------------------------------------------

131: /// Configuration of the SSC Baud Rate Generator:

132: /// -----------------------------------------------------------------------

133: /// - required baud rate = 125333,333 kbaud

134: /// - real baud rate = 125333,333 kbaud

135: /// - deviation = 0,000 %

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 14

136:

137: SSC\_BRH = 0x00; // load SSC baud rate time reload register

000265 E4 CLR A

000266 F5AF MOV SSC\_BRH,A

138: // high

139: SSC\_BRL = 0x00; // load SSC baud rate time reload register

000268 F5AE MOV SSC\_BRL,A

140: // low

141:

142: /// -----------------------------------------------------------------------

143: /// Configuration of the SSC Operation Mode:

144: /// -----------------------------------------------------------------------

145: /// - this device is configured as SSC master

146: /// - transfer data width is 2 bit

147: /// - transfer/receive LSB first

148: /// - shift transmit data on the leading clock edge, latch on trailing edge

149: /// - idle clock line is low, leading clock edge is low-to-high transition

150: /// - ignore receive error

151: /// - ignore phase error

152:

153:

154: SSC\_CONH\_P = 0x40; // load SSC control register

00026A 75AB40 MOV SSC\_CONH\_P,#040H

155:

156: SSC\_CONL\_P = 0x81; // load SSC control register

00026D 75AA81 MOV SSC\_CONL\_P,#081H

157:

158: SSC\_CONH\_P |= 0x80; // disable access to control bits

000270 43AB80 ORL SSC\_CONH\_P,#080H

159:

160: /// - Port Selection

161: SSC\_PISEL = 0x01; // load SSC Port Input Select Register to

000273 75A901 MOV SSC\_PISEL,#01H

162: // Port A

163:

164: /// -----------------------------------------------------------------------

165: /// Configuration of the used SSC Port Pins:

166: /// -----------------------------------------------------------------------

167: /// Pin SCLK (P0.3) is selected for SCLK ouput

168: /// Pin MTSR\_1 (P0.4) is selected for Master Transmit Output

169:

170: PORT\_PAGE = 0x02; // switch to page 2

000276 75B202 MOV PORT\_PAGE,#02H

171: P0\_ALTSEL0 |= 0x10; // set AltSel0

000279 438010 ORL P0\_ALTSEL0,#010H

172: P0\_ALTSEL1 &= ~(ubyte)0x10; // set AltSel1

00027C 5386EF ANL P0\_ALTSEL1,#0EFH

173:

174: PORT\_PAGE = 0x00; // switch to page 0

00027F F5B2 MOV PORT\_PAGE,A

175: P0\_DIR |= 0x10; // set P0.4 OutPut

000281 438610 ORL P0\_DIR,#010H

176:

177:

178: PORT\_PAGE = 0x02; // switch to page 2

000284 75B202 MOV PORT\_PAGE,#02H

179: P0\_ALTSEL0 |= 0x08; // set AltSel0

000287 438008 ORL P0\_ALTSEL0,#08H

180: P0\_ALTSEL1 &= ~(ubyte)0x08; // set AltSel1

00028A 5386F7 ANL P0\_ALTSEL1,#0F7H

181:

182: PORT\_PAGE = 0x00; // switch to page 0

00028D F5B2 MOV PORT\_PAGE,A

183: P0\_DIR |= 0x08; // set P0.3 OutPut

00028F 438608 ORL P0\_DIR,#08H

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 15

184:

185: /// Pin MRST\_1 (P0.5) is selected for Master Receive Input

186:

187: /// - SSC interrupt is enabled

188: IEN1 |= 0x02; // Enable SSC interrupt

000292 43E802 ORL IEN1,#02H

189:

190: // USER CODE BEGIN (Init,3)

191:

192: // USER CODE END

193:

194: } // End of function SSC\_vInit

000295 22 RET

----- FUNCTION SSC\_vInit (END) -------

----- FUNCTION SSC\_vIsr (BEGIN) -----

FILE: 'SSC.c'

293: void SSC\_vIsr(void) interrupt SSCINT

000296 C0E0 PUSH ACC

294: {

295: SFR\_PAGE(\_su0, SST2); // switch to page 0

000298 75BFA0 MOV SCU\_PAGE,#0A0H

296:

297: if (IRCON1 & 0x01) // if error

00029B E5B5 MOV A,IRCON1

00029D 30E006 JNB ACC.0,?C0004?SSC

298: {

299: IRCON1 &= ~(ubyte)0x01;

0002A0 53B5FE ANL IRCON1,#0FEH

300:

301: SFR\_PAGE(\_su0, RST2); // restore the old SCU page

0002A3 75BFE0 MOV SCU\_PAGE,#0E0H

302:

303: // USER CODE BEGIN (Isr,2)

304:

305: // USER CODE END

306: }

0002A6 ?C0004?SSC:

307:

308: SFR\_PAGE(\_su0, SST2); // switch to page 0

0002A6 75BFA0 MOV SCU\_PAGE,#0A0H

309:

310: if (IRCON1 & 0x02) // if Transmit Interrupt Request

0002A9 E5B5 MOV A,IRCON1

0002AB 30E106 JNB ACC.1,?C0005?SSC

311: {

312: IRCON1 &= ~(ubyte)0x02;

0002AE 53B5FD ANL IRCON1,#0FDH

313:

314: SFR\_PAGE(\_su0, RST2); // restore the old SCU page

0002B1 75BFE0 MOV SCU\_PAGE,#0E0H

315:

316: // USER CODE BEGIN (Isr,3)

317:

318: // USER CODE END

319: }

0002B4 ?C0005?SSC:

320:

321: SFR\_PAGE(\_su0, SST2); // switch to page 0

0002B4 75BFA0 MOV SCU\_PAGE,#0A0H

322:

323: if (IRCON1 & 0x04) // if Receive Interrupt Request

0002B7 E5B5 MOV A,IRCON1

0002B9 30E206 JNB ACC.2,?C0007?SSC

LX51 LINKER/LOCATER V4.20 11/09/2007 16:45:28 PAGE 16

324: {

325: IRCON1 &= ~(ubyte)0x04;

0002BC 53B5FB ANL IRCON1,#0FBH

326:

327: SFR\_PAGE(\_su0, RST2); // restore the old SCU page

0002BF 75BFE0 MOV SCU\_PAGE,#0E0H

328:

329: // USER CODE BEGIN (Isr,4)

330:

331: // USER CODE END

332: }

333:

334: } // End of function SSC\_vIsr

0002C2 ?C0007?SSC:

0002C2 D0E0 POP ACC

0002C4 32 RETI

----- FUNCTION SSC\_vIsr (END) -------