816CTEST PAGE 1

1 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : January 2001

6 ;

7 ; File : 816CTest.asm

8 ;

9 ; Description : QuickStart development kit basic test routine.

10 ;

11 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12

13 $MOD816 ; Use predefined Symbols

14

0000 15 FLAG BIT 00h

0001 16 FIRSTGO BIT 01H

00B4 17 LED EQU P3.4

18

19 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

20 ; DEFINE VARIABLES IN INTERNAL RAM

---- 21 DSEG

22

0060 23 ORG 0060h

0060 24 COUNT: DS 1 ; define 1 byte for the count

0061 25 DACOUT: DS 1 ; define 1 byte for the DAC o/p

0062 26 DPH1: DS 1 ; \ define 2 bytes for the

0063 27 DPL1: DS 1 ; / temporary storage of DPTR

0064 28 DATAOUT: DS 1 ; the current data put into mem

29

30 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

---- 31 CSEG ; (beginning of code)

32

0000 33 ORG 0000h

0000 020060 34 JMP MAIN

35

36 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

37

0003 38 ORG 0003h ; (INT0 ISR)

0003 C200 39 CLR FLAG ; clear flag

0005 32 40 RETI

41 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

42

0060 43 ORG 0060h ; (subroutines...)

44

45

46 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

47

0060 48 MAIN: ; (main program)

49

50

51 ; CONFIGURE UART

0060 75CBFF 52 MOV RCAP2H,#0FFh ; config UART for 9830baud

0063 75CAFB 53 MOV RCAP2L,#-5 ; (close enough to 9600baud)

0066 75CDFF 54 MOV TH2,#0FFh

0069 75CCFB 55 MOV TL2,#-5

006C 759852 56 MOV SCON,#52h

006F 75C834 57 MOV T2CON,#34h

58

816CTEST PAGE 2

59 ; CONFIGURE INTERRUPT 0...

60

0072 D288 61 SETB IT0 ; INT0 edge triggered

0074 D2A8 62 SETB EX0 ; enable INT0 interrupt

63

64 ; ENABLE INTERRUPTS & ENTER MAIN LOOP...

65

0076 D2AF 66 SETB EA ; enable inturrupts

67

0078 D200 68 SETB FLAG

007A 7401 69 MOV A, #01h

70

007C D299 71 SETB TI ; set TI flag to indicate that the

72 ; buffer is ready to transmit data.

007E 900C86 73 MOV DPTR, #GOMENU ; send a press INT0 message

0081 120D5A 74 CALL SENDSTRING

0084 D201 75 SETB FIRSTGO

76

0086 B2B4 77 BLINK: CPL LED ; blink LED until INT0 button pressed

0088 120D4E 78 CALL DELAY

008B 2000F8 79 JB FLAG, BLINK

80

008E D2B4 81 SETB LED ; turn on LED

82 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0090 83 MAINMENU:

0090 20010E 84 JB FIRSTGO, SKIP1

0093 900CAC 85 MOV DPTR, #RETMENU ; send a press INT0 message

0096 120D5A 86 CALL SENDSTRING

0099 D200 87 SETB FLAG

009B 2000FD 88 JB FLAG, $ ; wait for INT0

009E 0200B5 89 JMP SKIP2

00A1 90 SKIP1:

00A1 C201 91 CLR FIRSTGO

00A3 9003FB 92 MOV DPTR,#TITLE ; display title

00A6 120D5A 93 CALL SENDSTRING

00A9 900512 94 MOV DPTR,#LINKS1 ; display link options

00AC 120D5A 95 CALL SENDSTRING

00AF 9005A9 96 MOV DPTR,#LINKS2 ; display link options

00B2 120D5A 97 CALL SENDSTRING

00B5 98 SKIP2:

00B5 900471 99 MOV DPTR,#MENU ; display menu

00B8 120D5A 100 CALL SENDSTRING

101

00BB 3098FD 102 JNB RI, $ ; wait here for a key press

00BE E599 103 MOV A, SBUF ; read input

00C0 C298 104 CLR RI ; clear RI flag for the next input

105

00C2 B43103 106 CJNE A, #'1', $+6 ; if a 1 (31H) is pressed jmp TEST1

00C5 0200E2 107 JMP TEST1

00C8 B43203 108 CJNE A, #'2', $+6 ; if a 2 (32H) is pressed jmp TEST2

00CB 0201CC 109 JMP TEST2

00CE B43303 110 CJNE A, #'3', $+6 ; if a 3 (33H) is pressed jmp TEST3

00D1 0202C5 111 JMP TEST3

00D4 B43403 112 CJNE A, #'4', $+6 ; if a 4 (34H) is pressed jmp TEST4

00D7 020342 113 JMP TEST4

00DA B43503 114 CJNE A, #'5', $+6 ; if a 5 (35H) is pressed jmp TEST5

00DD 02038C 115 JMP TEST5

00E0 80AE 116 JMP MAINMENU

816CTEST PAGE 3

117

118 ;====================================================================

119 ; TEST 1: ADC AND DAC TEST

120 ;====================================================================

121

00E2 122 TEST1:

123

124

00E2 900730 125 MOV DPTR,#PRIMADC ; display ADC title

00E5 120D5A 126 CALL SENDSTRING

127

00E8 75FD17 128 MOV DACCON, #17h ; configure DAC for

129 ; 0-Vdd (5V), 12 bits,

00EB 75D247 130 MOV ADC0CON, #47h ; configure primary ADC for

131 ; external ref 2.5V, Ain1-Ain2,

132 ; bipolar, +-2.56V

00EE D200 133 SETB FLAG

00F0 2000FD 134 JB FLAG, $ ; wait for INT0

135

00F3 75610C 136 MOV DACOUT, #0Ch ; start of prim ADC conv

00F6 756005 137 MOV COUNT, #05h ; # of conv

138

00F9 120132 139 CALL TESTPRIMADC

140

141 ;aux channel Ain3

00FC 900802 142 MOV DPTR,#AUX3ADC

00FF 120D5A 143 CALL SENDSTRING

144

0102 75D348 145 MOV ADC1CON,#48h ; AIN3, unipolar, ext ref

0105 75FD13 146 MOV DACCON,#13h ; turn DAC on, 0-2.5V

147

0108 D200 148 SETB FLAG

010A 2000FD 149 JB FLAG, $ ; wait for INT0

150

010D 756105 151 MOV DACOUT, #05H

0110 12016A 152 CALL TESTAUXCHAN

153

154 ;aux channel Ain5

0113 900862 155 MOV DPTR,#AUX5ADC

0116 120D5A 156 CALL SENDSTRING

157

0119 75D378 158 MOV ADC1CON,#78h ; AIN5, unipolar, ext ref

011C 75FD13 159 MOV DACCON,#13h ; turn DAC on, 0-2.5V

160

011F D200 161 SETB FLAG

0121 2000FD 162 JB FLAG, $ ; wait for INT0

163

0124 756105 164 MOV DACOUT, #05H

0127 12016A 165 CALL TESTAUXCHAN

166

167 ; END OF TEST 1

012A 75D100 168 MOV ADCMODE,#00h ; turn ADCs off

012D 75FD00 169 MOV DACCON, #00h ; turn offthe DAC

0130 0190 170 JMP MAINMENU ; return

171

172

173 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

174

816CTEST PAGE 4

0132 175 TESTPRIMADC:

176 ; output voltage on DAC

0132 8561FC 177 MOV DACH,DACOUT ; change hi-byte only

0135 75FB00 178 MOV DACL,#00h

0138 7810 179 MOV R0,#10h ; allow for settling

013A D8FE 180 DJNZ R0,$

181

182 ; do a single conversion on the DAC voltage. Ain2 biased to

183 ; 2.5V. Hence the ADC conv voltage is the DAC voltage - 2.5V

013C 75D122 184 MOV ADCMODE,#22h ; single conv on prim chan

013F 30DFFD 185 JNB RDY0,$ ; wait for ADC result

0142 C2DF 186 CLR RDY0

187

188 ; display the DAC output on the screen

0144 900CD4 189 MOV DPTR,#DACMSG ; "DAC="

0147 120D5A 190 CALL SENDSTRING

014A 7E00 191 MOV R6, #00h

014C AF61 192 MOV R7, DACOUT ; display DAC value

014E 120DB5 193 CALL SEND12BITS

194

195 ; display the ADC conv result on the screen

0151 900CDA 196 MOV DPTR,#ADC0MSG ; " -> ADC0 = "

0154 120D5A 197 CALL SENDSTRING

198

0157 E5DB 199 MOV A, ADC0H ; send the 24 bit result upo the UART

0159 120D8A 200 CALL SENDVAL

015C E5DA 201 MOV A, ADC0M

015E 120D8A 202 CALL SENDVAL

203

0161 1201A0 204 CALL CHECKADC0 ; see if results are "OK" or "ERROR!"

205

0164 1561 206 DEC DACOUT

207

0166 D560C9 208 DJNZ COUNT, TESTPRIMADC

0169 22 209 RET

210

211

016A 212 TESTAUXCHAN:

213 ; output voltage on DAC

016A 8561FC 214 MOV DACH,DACOUT ; change hi-byte only

016D 75FB00 215 MOV DACL,#00h ; load DAC

0170 7810 216 MOV R0,#10h ; allow for settling

0172 D8FE 217 DJNZ R0, $

218

219 ; do a single conversion on the DAC voltage. The DAC voltage

220 ; appears between Ain3 and ground. Therefore the aux conv

221 ; should equal the DAC o/p

0174 75D112 222 MOV ADCMODE,#12h ; single conv on aux chan

0177 30DEFD 223 JNB RDY1,$ ; wait for ADC result

017A C2DE 224 CLR RDY1

225

226 ; display the DAC output on the screen

017C 900CD4 227 MOV DPTR,#DACMSG ; "DAC="

017F 120D5A 228 CALL SENDSTRING

0182 7E00 229 MOV R6, #00h

0184 AF61 230 MOV R7, DACOUT ; display DAC value

0186 120DB5 231 CALL SEND12BITS

232

816CTEST PAGE 5

233 ; display the ADC conv result on the screen

0189 900CEB 234 MOV DPTR,#ADC1MSG ; "ADC1="

018C 120D5A 235 CALL SENDSTRING

018F E5DD 236 MOV A, ADC1H ; display ADC results

0191 120D8A 237 CALL SENDVAL

0194 E5DC 238 MOV A, ADC1L

0196 120D8A 239 CALL SENDVAL

240

0199 1201B6 241 CALL CHECKADC1 ; "ok" or "ERROR!"

242

019C D561CB 243 DJNZ DACOUT, TESTAUXCHAN

019F 22 244 RET

245 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

01A0 246 CHECKADC0:

01A0 E561 247 MOV A, DACOUT

01A2 C4 248 SWAP A ; Swap Daccout from 0Ch to C0h etc

01A3 9408 249 SUBB A, #08h

01A5 FA 250 MOV R2, A ; R2 holds B8h (for DACH=0C)

01A6 E5DB 251 MOV A, ADC0H ; A holds say C1h

01A8 9A 252 SUBB A, R2

01A9 B41000 253 CJNE A, #10h, $+3 ; C is set if A<10H i.e. OK

01AC 4004 254 JC $+6

01AE 120D47 255 CALL PRINTERROR

01B1 22 256 RET

01B2 120D40 257 CALL PRINTOK

01B5 22 258 RET

259 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

01B6 260 CHECKADC1:

261 ; check if the ADC result is within 8LSBs of the DAC op

01B6 E561 262 MOV A, DACOUT

01B8 C4 263 SWAP A

01B9 9408 264 SUBB A, #08h

01BB FA 265 MOV R2, A

01BC E5DD 266 MOV A, ADC1H

01BE 9A 267 SUBB A, R2

01BF B41000 268 CJNE A, #10h, $+3 ; C is set if A<10H i.e. OK

01C2 4004 269 JC $+6

01C4 120D47 270 CALL PRINTERROR

01C7 22 271 RET

01C8 120D40 272 CALL PRINTOK

01CB 22 273 RET

274 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

275

276

277

278 ;====================================================================

279 ; TEST 2: EXTERNAL DATA MEMORY TEST

280 ;====================================================================

281

01CC 282 TEST2:

283 ; save particular data into internal RAM location 50h->59h

01CC 284 EXTDATA:

01CC 7850 285 MOV R0, #50h

01CE 76FF 286 MOV @R0, #0FFh ; store FFh into 50H

01D0 08 287 INC R0

01D1 7600 288 MOV @R0, #00h ; store 00h into 51h

01D3 08 289 INC R0

01D4 76AA 290 MOV @R0, #0AAh ; store AAh into 52h

816CTEST PAGE 6

01D6 08 291 INC R0

01D7 7655 292 MOV @R0, #55h ; store 55h into 53h

01D9 08 293 INC R0

01DA 76CC 294 MOV @R0, #0CCh ; store CCh into 54h

01DC 08 295 INC R0

01DD 7633 296 MOV @R0, #33h ; store 33h into 55h

01DF 08 297 INC R0

01E0 763F 298 MOV @R0, #3Fh ; store 3Fh into 56h

01E2 08 299 INC R0

01E3 7666 300 MOV @R0, #66h ; store 66h into 57h

301

01E5 9008C1 302 MOV DPTR,#MEMTITLE ; pick a memory mode

01E8 120D5A 303 CALL SENDSTRING

304 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

305

01EB 306 LK9A:

01EB 90092E 307 MOV DPTR,#LINK9A ; "LK9 to position A"

01EE 120D5A 308 CALL SENDSTRING

01F1 D200 309 SETB FLAG

01F3 2000FD 310 JB FLAG, $ ; wait for INT0

311

01F6 758400 312 MOV DPP,#00h

01F9 7850 313 MOV R0, #50h

01FB 9005FF 314 MOV DPTR, #05FFH

315

316 ; save data in RAM into the following locations

01FE 317 XRAMIN1:

01FE 758400 318 MOV DPP,#00h

0201 E6 319 MOV A, @R0

0202 8664 320 MOV DATAOUT, @R0

0204 F0 321 MOVX @DPTR, A

0205 858362 322 MOV DPH1, DPH ; DPH1 and DPL1 hold the present

0208 858263 323 MOV DPL1, DPL ; address in XRAM

324

020B 900D24 325 MOV DPTR, #ADDRMSG

020E 120D5A 326 CALL SENDSTRING

327

0211 E584 328 MOV A, DPP

0213 120D8A 329 CALL SENDVAL

0216 E562 330 MOV A, DPH1

0218 120D8A 331 CALL SENDVAL

021B E563 332 MOV A, DPL1

021D 120D8A 333 CALL SENDVAL

334

0220 900D2C 335 MOV DPTR, #WRITTENMSG

0223 120D5A 336 CALL SENDSTRING

337

0226 E564 338 MOV A, DATAOUT

0228 120D8A 339 CALL SENDVAL

340

022B 900D38 341 MOV DPTR, #READMSG

022E 120D5A 342 CALL SENDSTRING

343

0231 758455 344 MOV DPP, #55H ; change the page number. DPP does

0234 856283 345 MOV DPH, DPH1 ; not hold any address lines in this mode

0237 856382 346 MOV DPL, DPL1

023A E0 347 MOVX A, @DPTR

023B 120D8A 348 CALL SENDVAL

816CTEST PAGE 7

349

023E B56406 350 CJNE A, DATAOUT, ERROR2A

0241 120D40 351 CALL PRINTOK

0244 02024A 352 JMP $+6

353

0247 354 ERROR2A:

0247 120D47 355 CALL PRINTERROR

356

024A 856283 357 MOV DPH, DPH1

024D 856382 358 MOV DPL, DPL1

0250 08 359 INC R0

0251 E583 360 MOV A, DPH

0253 2420 361 ADD A, #20h

0255 F583 362 MOV DPH, A

363

0257 B858A4 364 CJNE R0, #58H, XRAMIN1

365 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

366

025A 900963 367 LK9B: MOV DPTR,#LINK9B ; "LK9 to position B"

025D 120D5A 368 CALL SENDSTRING

0260 D200 369 SETB FLAG

0262 2000FD 370 JB FLAG, $ ; wait for INT0

371

0265 758405 372 MOV DPP,#05h

0268 9000FF 373 MOV DPTR, #00FFH

026B 7857 374 MOV R0, #57h ; R0 starts at 57h

375

376 ; save data in RAM into the following locations

026D 377 XRAMIN2:

026D E6 378 MOV A, @R0

026E 8664 379 MOV DATAOUT, @R0

0270 F0 380 MOVX @DPTR, A

0271 858362 381 MOV DPH1, DPH ; DPH1 --> DPL1 hold the present

0274 858263 382 MOV DPL1, DPL ; address in XRAM

383

384

0277 900D24 385 MOV DPTR, #ADDRMSG

027A 120D5A 386 CALL SENDSTRING

387

027D E584 388 MOV A, DPP

027F 120D8A 389 CALL SENDVAL

0282 E562 390 MOV A, DPH1

0284 120D8A 391 CALL SENDVAL

0287 E563 392 MOV A, DPL1

0289 120D8A 393 CALL SENDVAL

394

028C 900D2C 395 MOV DPTR, #WRITTENMSG

028F 120D5A 396 CALL SENDSTRING

397

0292 E564 398 MOV A, DATAOUT

0294 120D8A 399 CALL SENDVAL

400

0297 900D38 401 MOV DPTR, #READMSG

029A 120D5A 402 CALL SENDSTRING

403

029D 758355 404 MOV DPH, #55h ; corrupt DPH as DPP contains the

02A0 856382 405 MOV DPL, DPL1 ; high address lines in this case

02A3 E0 406 MOVX A, @DPTR

816CTEST PAGE 8

02A4 120D8A 407 CALL SENDVAL

408

02A7 B56406 409 CJNE A, DATAOUT, ERROR2B

02AA 120D40 410 CALL PRINTOK

02AD 0202B3 411 JMP $+6

02B0 412 ERROR2B:

02B0 120D47 413 CALL PRINTERROR

414

02B3 856283 415 MOV DPH, DPH1

02B6 856382 416 MOV DPL, DPL1

02B9 18 417 DEC R0

02BA E584 418 MOV A, DPP

02BC 2420 419 ADD A, #20h

02BE F584 420 MOV DPP, A

421

02C0 B84FAA 422 CJNE R0, #4FH, XRAMIN2

02C3 0190 423 JMP MAINMENU

424

425

426 ;====================================================================

427 ; TEST 3: SPI Interface TEST

428 ;====================================================================

429

02C5 430 TEST3:

02C5 900998 431 MOV DPTR,#SPITITLE ; display SPI title

02C8 120D5A 432 CALL SENDSTRING

433

02CB D200 434 SETB FLAG

02CD 2000FD 435 JB FLAG, $ ; wait for INT0

436

02D0 75F833 437 MOV SPICON,#33h ; configure SPI port Master Mode

438 ; slowest bit rate

439

440 ; Test SCLOCK

02D3 75F700 441 MOV SPIDAT, #00h ; transmit 00h

02D6 442 WAITFORLOW:

02D6 20B30B 443 JB P3.3, SCLOCKHIGH; wait for SCLOCK to go low

02D9 444 WAITFORHIGH:

02D9 30B30D 445 JNB P3.3, SCLOCKLOW ; wait for SCLOCK to go high

02DC 30FFFD 446 JNB ISPI, $

02DF C2FF 447 CLR ISPI

02E1 0202EE 448 JMP TESTSDATA

02E4 449 SCLOCKHIGH:

02E4 20FF50 450 JB ISPI, SPIERROR

02E7 80ED 451 JMP WAITFORLOW

02E9 452 SCLOCKLOW:

02E9 20FF4B 453 JB ISPI, SPIERROR

02EC 80EB 454 JMP WAITFORHIGH

455

456

457 ; Test SDATA

02EE 458 TESTSDATA:

02EE 900A4D 459 MOV DPTR, #SDATAMSG

02F1 120D5A 460 CALL SENDSTRING

02F4 D200 461 SETB FLAG

02F6 2000FD 462 JB FLAG, $ ; wait for INT0

463

464 ; ; reinitialise the SPIPORT

816CTEST PAGE 9

465 ; MOV SPICON, #0

466 ; MOV SPICON,#33h ; configure SPI port Master Mode

467 ; slowest bit rate

02F9 75F70F 468 MOV SPIDAT, #0Fh ; transmit 0Fh

02FC 469 WAITFORLOW1:

02FC 20B30B 470 JB P3.3, SDATAHIGH; wait for SCLOCK to go low

02FF 471 WAITFORHIGH1:

02FF 30B30D 472 JNB P3.3, SDATALOW ; wait for SCLOCK to go high

0302 30FFFD 473 JNB ISPI, $

0305 C2FF 474 CLR ISPI

0307 020314 475 JMP TESTSS

030A 476 SDATAHIGH:

030A 20FF2A 477 JB ISPI, SPIERROR

030D 80ED 478 JMP WAITFORLOW1

030F 479 SDATALOW:

030F 20FF25 480 JB ISPI, SPIERROR

0312 80EB 481 JMP WAITFORHIGH1

482

0314 483 TESTSS:

484 ; connect p3.3 to SS

0314 900A75 485 MOV DPTR, #SSMSG

0317 120D5A 486 CALL SENDSTRING

031A D200 487 SETB FLAG

031C 2000FD 488 JB FLAG, $ ; wait for INT0

489

031F 75B0FF 490 MOV P3, #0FFh ; set all P3 as inputs

0322 D2B5 491 SETB P3.5

0324 30B310 492 JNB P3.3, SPIERROR

0327 C2B5 493 CLR P3.5

0329 20B30B 494 JB P3.3, SPIERROR

495

032C 496 SPIOK:

032C 900A9A 497 MOV DPTR, #SPIMSG

032F 120D5A 498 CALL SENDSTRING

0332 120D40 499 CALL PRINTOK

0335 0190 500 JMP MAINMENU

501

0337 502 SPIERROR:

0337 900A9A 503 MOV DPTR, #SPIMSG

033A 120D5A 504 CALL SENDSTRING

033D 120D47 505 CALL PRINTERROR

0340 0190 506 JMP MAINMENU

507

508

509 ;====================================================================

510 ; TEST 4: RTD Demo Circuit TEST

511 ;====================================================================

512

0342 513 TEST4:

0342 900AB9 514 MOV DPTR,#RTDTITLE ; display RTD title

0345 120D5A 515 CALL SENDSTRING

516

0348 900B20 517 MOV DPTR,#RTDLINKS ; display RTD links

034B 120D5A 518 CALL SENDSTRING

519

520 ; Configure Current source

034E 75D501 521 MOV ICON, #01h ; put 200uA out on pin 3

522

816CTEST PAGE 10

523

0351 D200 524 SETB FLAG

0353 2000FD 525 JB FLAG, $ ; wait for INT0

526

527

528 ; Configure ADC

0356 75D24A 529 MOV ADC0CON, #4AH ; use external ref (R11)

530 ; unpolar mode

531 ; fixed 0->80mV range

532

0359 75D122 533 MOV ADCMODE, #22H ; initiate a single prim chan conv

035C 30DFFD 534 JNB RDY0,$ ; Wait for conversion results

535

035F 900CF8 536 MOV DPTR, #RTDSENSOR ; Send temp up UART

0362 120D5A 537 CALL SENDSTRING

0365 E5DB 538 MOV A, ADC0H

0367 120D8A 539 CALL SENDVAL

036A E5DA 540 MOV A, ADC0M

036C 120D8A 541 CALL SENDVAL

542

543 ; a value in AD0H of between 88h and A7h represents temps

544 ; between -11degC and 47degC. Only these temps should

545 ; pass the test.

036F E5DB 546 MOV A, ADC0H

0371 C2D7 547 CLR CY

0373 9408 548 SUBB A, #08h ; A7->9Fh, 88->80h

0375 54F0 549 ANL A, #0F0h ; correct data is now either 80h or 90h

0377 B49005 550 CJNE A, #90h, CHECK80 ; if fails 90 check A0

037A 120D40 551 CALL PRINTOK

037D 0190 552 JMP MAINMENU

037F 553 CHECK80:

037F B48005 554 CJNE A, #80h, ERROR4 ; fails 80 and 90 => fail

0382 120D40 555 CALL PRINTOK

0385 0190 556 JMP MAINMENU

0387 120D47 557 ERROR4: CALL PRINTERROR

038A 0190 558 JMP MAINMENU

559

560

561 ;====================================================================

562 ; TEST 5: Automatic Checks

563 ;====================================================================

038C 564 TEST5:

038C 900BCD 565 MOV DPTR, #AUTO

038F 120D5A 566 CALL SENDSTRING

567

568 ;test port 1 for shorts

0392 900C45 569 MOV DPTR, #PORT1

0395 120D5A 570 CALL SENDSTRING

0398 D291 571 SETB P1.1

039A C290 572 CLR P1.0

039C E590 573 MOV A, P1

039E 5403 574 ANL A, #03h ; zero acc.2 -> acc.7

03A0 B40205 575 CJNE A, #02h, $+8 ; p1.1=1, p1.0=0 =>OK

03A3 120D40 576 CALL PRINTOK

03A6 61AB 577 AJMP $+5

03A8 120D47 578 CALL PRINTERROR

579

580 ;test port 3 for shorts

816CTEST PAGE 11

03AB 900C28 581 MOV DPTR, #PORT3

03AE 120D5A 582 CALL SENDSTRING

03B1 D2B2 583 SETB P3.2 ; do not alter TXD and RXD lines

03B3 C2B3 584 CLR P3.3

03B5 D2B4 585 SETB P3.4

03B7 C2B5 586 CLR P3.5

03B9 D2B6 587 SETB P3.6

03BB C2B7 588 CLR P3.7

03BD E5B0 589 MOV A, P3

03BF 54FC 590 ANL A, #0FCh ; zero acc.0 and acc.1

03C1 B45405 591 CJNE A, #54h, $+8

03C4 120D40 592 CALL PRINTOK

03C7 61CC 593 AJMP $+5

03C9 120D47 594 CALL PRINTERROR

595

596 ;use TIC to test if crystal is oscillating

03CC 900C6C 597 MOV DPTR, #CRYSTAL

03CF 120D5A 598 CALL SENDSTRING

599

03D2 75A60B 600 MOV INTVAL, #0BH ; SET INTVAL FOR 12/128TH =93ms

03D5 75A103 601 MOV TIMECON, #03H ; MEASURE IN 1/128THS SECS

03D8 7401 602 MOV A, #01H

03DA 120D4E 603 CALL DELAY ;SOFTWARE 100ms delay => TII set

03DD E5A1 604 MOV A, TIMECON

03DF 20E205 605 JB ACC.2, $+8

03E2 120D47 606 CALL PRINTERROR

03E5 61EA 607 AJMP $+5

03E7 120D40 608 CALL PRINTOK

03EA 75A100 609 MOV TIMECON, #00h ; this clears TII and disables TIC

610

611 ;any other automatic tests go here

612

613

03ED 900627 614 MOV DPTR, #REMOVELINKS1

03F0 120D5A 615 CALL SENDSTRING

03F3 9006BA 616 MOV DPTR, #REMOVELINKS2

03F6 120D5A 617 CALL SENDSTRING

03F9 0190 618 JMP MAINMENU

619

620 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

621

03FB 0A0A0A0D 622 TITLE: DB 10,10,10,13,'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_',10,13

03FF 5F5F5F5F

0403 5F5F5F5F

0407 5F5F5F5F

040B 5F5F5F5F

040F 5F5F5F5F

0413 5F5F5F5F

0417 5F5F5F5F

041B 5F5F5F5F

041F 5F5F5F5F

0423 5F5F5F0A

0427 0D

0428 20416E61 623 DB ' Analog Devices MicroConverter ADuC824',10,13

042C 6C6F6720

0430 44657669

0434 63657320

0438 4D696372

816CTEST PAGE 12

043C 6F436F6E

0440 76657274

0444 65722041

0448 44754338

044C 32340A0D

0450 20202045 624 DB ' Evaluation Board Test Routine',0

0454 76616C75

0458 6174696F

045C 6E20426F

0460 61726420

0464 54657374

0468 20526F75

046C 74696E65

0470 00

0471 625 Menu:

0471 0A0A0D4D 626 DB 10,10,13,'Main Menu:', 10, 13

0475 61696E20

0479 4D656E75

047D 3A0A0D

0480 20203120 627 DB ' 1 - Test ADCs and DAC',10,13

0484 2D205465

0488 73742041

048C 44437320

0490 616E6420

0494 4441430A

0498 0D

0499 20203220 628 DB ' 2 - Test External Data Memory' ,10,13

049D 2D205465

04A1 73742045

04A5 78746572

04A9 6E616C20

04AD 44617461

04B1 204D656D

04B5 6F72790A

04B9 0D

04BA 20203320 629 DB ' 3 - Test SPI Interface', 10,13

04BE 2D205465

04C2 73742053

04C6 50492049

04CA 6E746572

04CE 66616365

04D2 0A0D

04D4 20203420 630 DB ' 4 - Test RTD Demo Circuit',10,13

04D8 2D205465

04DC 73742052

04E0 54442044

04E4 656D6F20

04E8 43697263

04EC 7569740A

04F0 0D

04F1 20203520 631 DB ' 5 - Automatic Checks',10,13

04F5 2D204175

04F9 746F6D61

04FD 74696320

0501 43686563

0505 6B730A0D

0509 53656C65 632 DB 'Select: ',0

050D 63743A20

816CTEST PAGE 13

0511 00

633

0512 0A0A0D45 634 LINKS1: DB 10,10,13,'Ensure that ONLY the following links are inserted'

0516 6E737572

051A 65207468

051E 6174204F

0522 4E4C5920

0526 74686520

052A 666F6C6C

052E 6F77696E

0532 67206C69

0536 6E6B7320

053A 61726520

053E 696E7365

0542 72746564

0546 0A0D202D 635 DB 10,13,' - Slide LK1 to ON Posn'

054A 20536C69

054E 6465204C

0552 4B312074

0556 6F204F4E

055A 20506F73

055E 6E

055F 0A0D202D 636 DB 10,13,' - Slide LK5 to ON Posn'

0563 20536C69

0567 6465204C

056B 4B352074

056F 6F204F4E

0573 20506F73

0577 6E

0578 0A0D202D 637 DB 10,13,' - Slide LK6 to Posn A'

057C 20536C69

0580 6465204C

0584 4B362074

0588 6F20506F

058C 736E2041

0590 0A0D202D 638 DB 10,13,' - Slide LK7 to Posn A',0

0594 20536C69

0598 6465204C

059C 4B372074

05A0 6F20506F

05A4 736E2041

05A8 00

05A9 0A0D202D 639 LINKS2: DB 10,13,' - Slide LK8 to Posn A'

05AD 20536C69

05B1 6465204C

05B5 4B382074

05B9 6F20506F

05BD 736E2041

05C1 0A0D202D 640 DB 10,13,' - Slide LK9 to Posn A'

05C5 20536C69

05C9 6465204C

05CD 4B392074

05D1 6F20506F

05D5 736E2041

05D9 0A0D202D 641 DB 10,13,' - Slide LK10 to Posn A'

05DD 20536C69

05E1 6465204C

05E5 4B313020

816CTEST PAGE 14

05E9 746F2050

05ED 6F736E20

05F1 41

05F2 0A0D202D 642 DB 10,13,' - Slide LK11 to ON Posn'

05F6 20536C69

05FA 6465204C

05FE 4B313120

0602 746F204F

0606 4E20506F

060A 736E

060C 0A0D202D 643 DB 10,13,' - Slide LK12 to ON Posn',0

0610 20536C69

0614 6465204C

0618 4B313220

061C 746F204F

0620 4E20506F

0624 736E00

644

0627 645 REMOVELINKS1:

0627 0A0A0D49 646 DB 10,10,13,'If finished INSERT only the following Links'

062B 66206669

062F 6E697368

0633 65642049

0637 4E534552

063B 54206F6E

063F 6C792074

0643 68652066

0647 6F6C6C6F

064B 77696E67

064F 204C696E

0653 6B73

0655 0A0D202D 647 DB 10,13,' - Slide LK1 to ON Posn'

0659 20536C69

065D 6465204C

0661 4B312074

0665 6F204F4E

0669 20506F73

066D 6E

066E 0A0D202D 648 DB 10,13,' - Slide LK3 to ON Posn'

0672 20536C69

0676 6465204C

067A 4B332074

067E 6F204F4E

0682 20506F73

0686 6E

0687 0A0D202D 649 DB 10,13,' - Slide LK5 to ON Posn'

068B 20536C69

068F 6465204C

0693 4B352074

0697 6F204F4E

069B 20506F73

069F 6E

06A0 0A0D202D 650 DB 10,13,' - Slide LK6 to ON Posn',0

06A4 20536C69

06A8 6465204C

06AC 4B362074

06B0 6F204F4E

06B4 20506F73

816CTEST PAGE 15

06B8 6E00

06BA 651 REMOVELINKS2:

06BA 0A0D202D 652 DB 10,13,' - Slide LK7 to Posn A'

06BE 20536C69

06C2 6465204C

06C6 4B372074

06CA 6F20506F

06CE 736E2041

06D2 0A0D202D 653 DB 10,13,' - Slide LK8 to Posn A'

06D6 20536C69

06DA 6465204C

06DE 4B382074

06E2 6F20506F

06E6 736E2041

06EA 0A0D202D 654 DB 10,13,' - Slide LK9 to Posn A'

06EE 20536C69

06F2 6465204C

06F6 4B392074

06FA 6F20506F

06FE 736E2041

0702 0A0D416C 655 DB 10,13,'All other links should be in the OFF posn',10,13,0

0706 6C206F74

070A 68657220

070E 6C696E6B

0712 73207368

0716 6F756C64

071A 20626520

071E 696E2074

0722 6865204F

0726 46462070

072A 6F736E0A

072E 0D00

656

657

0730 0A0A0D54 658 PRIMADC: DB 10,10,13,'TEST 1: Testing ADCs and DAC in a Feedback Configuration'

0734 45535420

0738 313A2054

073C 65737469

0740 6E672041

0744 44437320

0748 616E6420

074C 44414320

0750 696E2061

0754 20466565

0758 64626163

075C 6B20436F

0760 6E666967

0764 75726174

0768 696F6E

076B 0A0D2D2D 659 DB 10,13,'-------------------------------------------------------',10,13

076F 2D2D2D2D

0773 2D2D2D2D

0777 2D2D2D2D

077B 2D2D2D2D

077F 2D2D2D2D

0783 2D2D2D2D

0787 2D2D2D2D

078B 2D2D2D2D

816CTEST PAGE 16

078F 2D2D2D2D

0793 2D2D2D2D

0797 2D2D2D2D

079B 2D2D2D2D

079F 2D2D2D2D

07A3 2D0A0D

07A6 436F6E6E 660 DB 'Connect the DAC (J2-13) to the Primary ADC (Ain1 J2-9).',10,13

07AA 65637420

07AE 74686520

07B2 44414320

07B6 284A322D

07BA 31332920

07BE 746F2074

07C2 68652050

07C6 72696D61

07CA 72792041

07CE 44432028

07D2 41696E31

07D6 204A322D

07DA 39292E0A

07DE 0D

07DF 50726573 661 DB 'Press the INT0 button when ready',10,13,0

07E3 73207468

07E7 6520494E

07EB 54302062

07EF 7574746F

07F3 6E207768

07F7 656E2072

07FB 65616479

07FF 0A0D00

662

0802 0A0A0D43 663 AUX3ADC: DB 10,10,13,'Connect the DAC (J2-13) to the Auxilary ADC (Ain3 J2-11)',10,13

0806 6F6E6E65

080A 63742074

080E 68652044

0812 41432028

0816 4A322D31

081A 33292074

081E 6F207468

0822 65204175

0826 78696C61

082A 72792041

082E 44432028

0832 41696E33

0836 204A322D

083A 3131290A

083E 0D

083F 50726573 664 DB 'Press the INT0 button when ready',10,13,0

0843 73207468

0847 6520494E

084B 54302062

084F 7574746F

0853 6E207768

0857 656E2072

085B 65616479

085F 0A0D00

665

0862 0A0A0D43 666 AUX5ADC: DB 10,10,13,'Connect the DAC (J2-13) to the Auxilary ADC (Ain5 J2-6)',10,13

816CTEST PAGE 17

0866 6F6E6E65

086A 63742074

086E 68652044

0872 41432028

0876 4A322D31

087A 33292074

087E 6F207468

0882 65204175

0886 78696C61

088A 72792041

088E 44432028

0892 41696E35

0896 204A322D

089A 36290A0D

089E 50726573 667 DB 'Press the INT0 button when ready',10,13,0

08A2 73207468

08A6 6520494E

08AA 54302062

08AE 7574746F

08B2 6E207768

08B6 656E2072

08BA 65616479

08BE 0A0D00

668

08C1 0A0A0D20 669 MEMTITLE: DB 10,10,13,' TEST 2: Testing the External Data Memory'

08C5 20202020

08C9 20544553

08CD 5420323A

08D1 20205465

08D5 7374696E

08D9 67207468

08DD 65204578

08E1 7465726E

08E5 616C2044

08E9 61746120

08ED 4D656D6F

08F1 7279

08F3 0A0D2D2D 670 DB 10,13,'------------------------------------------------------',10,13,0

08F7 2D2D2D2D

08FB 2D2D2D2D

08FF 2D2D2D2D

0903 2D2D2D2D

0907 2D2D2D2D

090B 2D2D2D2D

090F 2D2D2D2D

0913 2D2D2D2D

0917 2D2D2D2D

091B 2D2D2D2D

091F 2D2D2D2D

0923 2D2D2D2D

0927 2D2D2D2D

092B 0A0D00

092E 0A0D536C 671 LINK9A: DB 10,13,'Slide LK9 to position A. Press INT0 when ready.',10,13,0

0932 69646520

0936 4C4B3920

093A 746F2070

093E 6F736974

0942 696F6E20

816CTEST PAGE 18

0946 412E2020

094A 50726573

094E 7320494E

0952 54302077

0956 68656E20

095A 72656164

095E 792E0A0D

0962 00

0963 0A0D536C 672 LINK9B: DB 10,13,'Slide LK9 to position B. Press INT0 when ready.',10,13,0

0967 69646520

096B 4C4B3920

096F 746F2070

0973 6F736974

0977 696F6E20

097B 422E2020

097F 50726573

0983 7320494E

0987 54302077

098B 68656E20

098F 72656164

0993 792E0A0D

0997 00

673

0998 0A0A0D20 674 SPITITLE: DB 10,10,13,' TEST 3: Testing the SPI/I2C Interface'

099C 20202020

09A0 20205445

09A4 53542033

09A8 3A205465

09AC 7374696E

09B0 67207468

09B4 65205350

09B8 492F4932

09BC 4320496E

09C0 74657266

09C4 616365

09C7 0A0D2D2D 675 DB 10,13,'------------------------------------------------------',10,13

09CB 2D2D2D2D

09CF 2D2D2D2D

09D3 2D2D2D2D

09D7 2D2D2D2D

09DB 2D2D2D2D

09DF 2D2D2D2D

09E3 2D2D2D2D

09E7 2D2D2D2D

09EB 2D2D2D2D

09EF 2D2D2D2D

09F3 2D2D2D2D

09F7 2D2D2D2D

09FB 2D2D2D2D

09FF 0A0D

0A01 0A0D436F 676 DB 10,13,'Connect SCLOCK (J7-1) to P3.3 (J3-4).'

0A05 6E6E6563

0A09 74205343

0A0D 4C4F434B

0A11 20284A37

0A15 2D312920

0A19 746F2050

0A1D 332E3320

816CTEST PAGE 19

0A21 284A332D

0A25 34292E

0A28 0A0D5072 677 DB 10,13,'Press the INT0 button when ready',10,13,0

0A2C 65737320

0A30 74686520

0A34 494E5430

0A38 20627574

0A3C 746F6E20

0A40 7768656E

0A44 20726561

0A48 64790A0D

0A4C 00

0A4D 0A0D436F 678 SDATAMSG: DB 10,13,'Connect SDATA (J7-3) to P3.3 (J3-4)',10,13,0

0A51 6E6E6563

0A55 74205344

0A59 41544120

0A5D 284A372D

0A61 33292074

0A65 6F205033

0A69 2E332028

0A6D 4A332D34

0A71 290A0D00

0A75 0A0D436F 679 SSMSG: DB 10,13,'Connect SS (J7-7) to P3.3 (J3-4)',10,13,0

0A79 6E6E6563

0A7D 74205353

0A81 20284A37

0A85 2D372920

0A89 746F2050

0A8D 332E3320

0A91 284A332D

0A95 34290A0D

0A99 00

0A9A 0A0D5465 680 SPIMSG: DB 10,13,'Testing SPI/I2C Interface...',0

0A9E 7374696E

0AA2 67205350

0AA6 492F4932

0AAA 4320496E

0AAE 74657266

0AB2 6163652E

0AB6 2E2E00

681

0AB9 0A0A0D20 682 RTDTITLE: DB 10,10,13,' TEST4: Testing the RTD Demo Circuit'

0ABD 20202020

0AC1 20202054

0AC5 45535434

0AC9 3A205465

0ACD 7374696E

0AD1 67207468

0AD5 65205254

0AD9 44204465

0ADD 6D6F2043

0AE1 69726375

0AE5 6974

0AE7 0A0D2D2D 683 DB 10,13,'------------------------------------------------------',0

0AEB 2D2D2D2D

0AEF 2D2D2D2D

0AF3 2D2D2D2D

0AF7 2D2D2D2D

816CTEST PAGE 20

0AFB 2D2D2D2D

0AFF 2D2D2D2D

0B03 2D2D2D2D

0B07 2D2D2D2D

0B0B 2D2D2D2D

0B0F 2D2D2D2D

0B13 2D2D2D2D

0B17 2D2D2D2D

0B1B 2D2D2D2D

0B1F 00

0B20 684 RTDLINKS:

0B20 0A0D4368 685 DB 10,13,'Change the following Links:'

0B24 616E6765

0B28 20746865

0B2C 20666F6C

0B30 6C6F7769

0B34 6E67204C

0B38 696E6B73

0B3C 3A

0B3D 0A0D202D 686 DB 10,13,' - Move LK7 into position B'

0B41 204D6F76

0B45 65204C4B

0B49 3720696E

0B4D 746F2070

0B51 6F736974

0B55 696F6E20

0B59 42

0B5A 0A0D202D 687 DB 10,13,' - Move LK8 into position B'

0B5E 204D6F76

0B62 65204C4B

0B66 3820696E

0B6A 746F2070

0B6E 6F736974

0B72 696F6E20

0B76 42

0B77 0A0D202D 688 DB 10,13,' - REMOVE LK5'

0B7B 2052454D

0B7F 4F564520

0B83 4C4B35

0B86 0A0D4E4F 689 DB 10,13,'NOTE: Remove any connector used in test 1'

0B8A 54453A20

0B8E 52656D6F

0B92 76652061

0B96 6E792063

0B9A 6F6E6E65

0B9E 63746F72

0BA2 20757365

0BA6 6420696E

0BAA 20746573

0BAE 742031

0BB1 0A0A0D50 690 DB 10,10,13,'Press INT0 when ready.',10,13,0

0BB5 72657373

0BB9 20494E54

0BBD 30207768

0BC1 656E2072

0BC5 65616479

0BC9 2E0A0D00

691

816CTEST PAGE 21

0BCD 0A0A0D20 692 AUTO: DB 10,10,13,' TEST5: Automatic Checks'

0BD1 20202020

0BD5 20202054

0BD9 45535435

0BDD 3A204175

0BE1 746F6D61

0BE5 74696320

0BE9 43686563

0BED 6B73

0BEF 0A0D2D2D 693 DB 10,13,'------------------------------------------------------',0

0BF3 2D2D2D2D

0BF7 2D2D2D2D

0BFB 2D2D2D2D

0BFF 2D2D2D2D

0C03 2D2D2D2D

0C07 2D2D2D2D

0C0B 2D2D2D2D

0C0F 2D2D2D2D

0C13 2D2D2D2D

0C17 2D2D2D2D

0C1B 2D2D2D2D

0C1F 2D2D2D2D

0C23 2D2D2D2D

0C27 00

0C28 0A0D5465 694 PORT3: DB 10,13,'Testing Port 3 for shorts ', 0

0C2C 7374696E

0C30 6720506F

0C34 72742033

0C38 20666F72

0C3C 2073686F

0C40 72747320

0C44 00

0C45 0A0D5465 695 PORT1: DB 10,13,'Testing Port1.0 and P1.1 for shorts ', 0

0C49 7374696E

0C4D 6720506F

0C51 7274312E

0C55 3020616E

0C59 64205031

0C5D 2E312066

0C61 6F722073

0C65 686F7274

0C69 732000

0C6C 0A0D5465 696 CRYSTAL: DB 10,13,'Testing crystal ciruit ',0

0C70 7374696E

0C74 67206372

0C78 79737461

0C7C 6C206369

0C80 72756974

0C84 2000

697

0C86 0A0A0A0D 698 GOMENU: DB 10,10,10,13,'Press INT0 to go to the Main Menu',0

0C8A 50726573

0C8E 7320494E

0C92 54302074

0C96 6F20676F

0C9A 20746F20

0C9E 74686520

0CA2 4D61696E

816CTEST PAGE 22

0CA6 204D656E

0CAA 7500

0CAC 0A0D5072 699 RETMENU: DB 10,13,'Press INT0 to return to the Main Menu',0

0CB0 65737320

0CB4 494E5430

0CB8 20746F20

0CBC 72657475

0CC0 726E2074

0CC4 6F207468

0CC8 65204D61

0CCC 696E204D

0CD0 656E7500

700

0CD4 20444143 701 DACMSG: DB ' DAC=',0

0CD8 3D00

0CDA 202D3E20 702 ADC0MSG: DB ' -> Primary ADC=',0

0CDE 5072696D

0CE2 61727920

0CE6 4144433D

0CEA 00

0CEB 202D3E20 703 ADC1MSG: DB ' -> Aux ADC=',0

0CEF 41757820

0CF3 4144433D

0CF7 00

0CF8 2054656D 704 RTDSENSOR: DB ' Temperature Conversion = ',0

0CFC 70657261

0D00 74757265

0D04 20436F6E

0D08 76657273

0D0C 696F6E20

0D10 3D2000

705

0D13 204F4B0A 706 OKMSG: DB ' OK',10,13,0

0D17 0D00

0D19 20455252 707 ERRORMSG: DB ' ERROR!!',10,13,0

0D1D 4F522121

0D21 0A0D00

708

0D24 0D204144 709 ADDRMSG: DB 13,' ADDR=',0

0D28 44523D00

0D2C 3A202057 710 WRITTENMSG: DB ': WRITTEN=',0

0D30 52495454

0D34 454E3D00

0D38 2C205245 711 READMSG: DB ', READ=',0

0D3C 41443D00

712

713

714

715

716

717

718

719 ;====================================================================

720 ; FUNCTIONS

721 ;====================================================================

722

723

724 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

816CTEST PAGE 23

725 ; PRINT OK AND ERROR MESSAGES

0D40 726 PRINTOK:

0D40 900D13 727 MOV DPTR,#OKMSG

0D43 120D5A 728 CALL SENDSTRING

0D46 22 729 RET

0D47 730 PRINTERROR:

0D47 900D19 731 MOV DPTR,#ERRORMSG

0D4A 120D5A 732 CALL SENDSTRING

0D4D 22 733 RET

734 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

735 ; DELAY

0D4E 736 DELAY: ; Delays by 100ms \* A

737 ; 100mSec based on 1.573MHZ Core Clock

738

0D4E FA 739 MOV R2,A ; Acc holds delay variable

0D4F 7B32 740 DLY0: MOV R3,#50 ; Set up delay loop0

0D51 7C83 741 DLY1: MOV R4,#131 ; Set up delay loop1

0D53 DCFE 742 DJNZ R4, $ ; Dec R4 & Jump here until R4 is 0

743 ; wait here for 131\*15.3us=2ms

0D55 DBFA 744 DJNZ R3, DLY1 ; Dec R3 & Jump DLY1 until R3 is 0

745 ; Wait for 50\*2ms

0D57 DAF6 746 DJNZ R2,DLY0 ; Dec R2 & Jump DLY0 until R2 is 0

747 ; wait for ACC\*100ms

0D59 22 748 RET ; Return from subroutine

749

750 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0D5A 751 SENDSTRING: ; sends ASCII string to UART starting at location DPTR

752 ; and ending with a null (0) value

753

0D5A C0E0 754 PUSH ACC

0D5C C0F0 755 PUSH B

0D5E E4 756 CLR A

0D5F F5F0 757 MOV B,A

0D61 E5F0 758 SENDNEXT: MOV A,B

0D63 05F0 759 INC B

0D65 93 760 MOVC A,@A+DPTR

0D66 6009 761 JZ SENDDONE

0D68 3099FD 762 JNB TI,$ ; wait til present char gone

0D6B C299 763 CLR TI ; must clear TI

0D6D F599 764 MOV SBUF,A ; transmit byte

0D6F 80F0 765 JMP SENDNEXT

0D71 D0F0 766 SENDDONE: POP B

0D73 D0E0 767 POP ACC

768

0D75 22 769 RET

770

771 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0D76 772 SENDCHAR: ; sends ASCII value contained in A to UART

773

0D76 3099FD 774 JNB TI,$ ; wait til present char gone

0D79 C299 775 CLR TI ; must clear TI

0D7B F599 776 MOV SBUF,A

777

0D7D 22 778 RET

779

780 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0D7E 781 HEX2ASCII: ; converts A into the hex character representing the

782 ; previous value of A's least significant nibble

816CTEST PAGE 24

783

0D7E 540F 784 ANL A,#00Fh

0D80 B40A00 785 CJNE A,#00Ah,$+3

0D83 4002 786 JC CONV

0D85 2407 787 ADD A,#007h

0D87 2430 788 CONV: ADD A,#'0'

789

0D89 22 790 RET

791

792 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0D8A 793 SENDVAL: ; uses the above two subroutines to....

794 ; 1) convert the hex value of A into two ASCII chars

795 ; 2) spew converted chars out the UART

796

0D8A C0E0 797 PUSH ACC

0D8C C4 798 SWAP A

0D8D 540F 799 ANL A,#00Fh

0D8F B17E 800 CALL HEX2ASCII

0D91 B176 801 CALL SENDCHAR

0D93 D0E0 802 POP ACC

0D95 C0E0 803 PUSH ACC

0D97 540F 804 ANL A,#00Fh

0D99 B17E 805 CALL HEX2ASCII

0D9B B176 806 CALL SENDCHAR

0D9D D0E0 807 POP ACC

808

0D9F 22 809 RET

810 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

811 ; shifts the 12MSBs of R7,R6 to the 12LSBs of R7,R6.

812 ; A zero is read into the 4 MSBs of R7

0DA0 813 SHIFT:

0DA0 C0E0 814 PUSH ACC

0DA2 EE 815 MOV A,R6

0DA3 54F0 816 ANL A,#0F0h

0DA5 C4 817 SWAP A

0DA6 FE 818 MOV R6,A

0DA7 EF 819 MOV A,R7

0DA8 C4 820 SWAP A

0DA9 54F0 821 ANL A,#0F0h

0DAB 4E 822 ORL A,R6

0DAC FE 823 MOV R6,A

0DAD EF 824 MOV A,R7

0DAE C4 825 SWAP A

0DAF 540F 826 ANL A,#0Fh

0DB1 FF 827 MOV R7,A

0DB2 D0E0 828 POP ACC

829

0DB4 22 830 RET

831 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

832 ; sends the 12 LSBs of R7,R6 up the UART in ASCII form

833

0DB5 834 SEND12BITS:

0DB5 C0E0 835 PUSH ACC

0DB7 EF 836 MOV A,R7

0DB8 540F 837 ANL A,#0Fh

0DBA B17E 838 CALL HEX2ASCII

0DBC B176 839 CALL SENDCHAR

0DBE EE 840 MOV A,R6

816CTEST PAGE 25

0DBF C4 841 SWAP A

0DC0 540F 842 ANL A,#0Fh

0DC2 B17E 843 CALL HEX2ASCII

0DC4 B176 844 CALL SENDCHAR

0DC6 EE 845 MOV A,R6

0DC7 540F 846 ANL A,#0Fh

0DC9 B17E 847 CALL HEX2ASCII

0DCB B176 848 CALL SENDCHAR

0DCD D0E0 849 POP ACC

0DCF 22 850 RET

851 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

852

853 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

816CTEST PAGE 26

ACC. . . . . . . . . . . . . . . D ADDR 00E0H PREDEFINED

ADC0CON. . . . . . . . . . . . . D ADDR 00D2H PREDEFINED

ADC0H. . . . . . . . . . . . . . D ADDR 00DBH PREDEFINED

ADC0M. . . . . . . . . . . . . . D ADDR 00DAH PREDEFINED

ADC0MSG. . . . . . . . . . . . . C ADDR 0CDAH

ADC1CON. . . . . . . . . . . . . D ADDR 00D3H PREDEFINED

ADC1H. . . . . . . . . . . . . . D ADDR 00DDH PREDEFINED

ADC1L. . . . . . . . . . . . . . D ADDR 00DCH PREDEFINED

ADC1MSG. . . . . . . . . . . . . C ADDR 0CEBH

ADCMODE. . . . . . . . . . . . . D ADDR 00D1H PREDEFINED

ADDRMSG. . . . . . . . . . . . . C ADDR 0D24H

AUTO . . . . . . . . . . . . . . C ADDR 0BCDH

AUX3ADC. . . . . . . . . . . . . C ADDR 0802H

AUX5ADC. . . . . . . . . . . . . C ADDR 0862H

B. . . . . . . . . . . . . . . . D ADDR 00F0H PREDEFINED

BLINK. . . . . . . . . . . . . . C ADDR 0086H

CHECK80. . . . . . . . . . . . . C ADDR 037FH

CHECKADC0. . . . . . . . . . . . C ADDR 01A0H

CHECKADC1. . . . . . . . . . . . C ADDR 01B6H

CONV . . . . . . . . . . . . . . C ADDR 0D87H

COUNT. . . . . . . . . . . . . . D ADDR 0060H

CRYSTAL. . . . . . . . . . . . . C ADDR 0C6CH

CY . . . . . . . . . . . . . . . B ADDR 00D7H PREDEFINED

DACCON . . . . . . . . . . . . . D ADDR 00FDH PREDEFINED

DACH . . . . . . . . . . . . . . D ADDR 00FCH PREDEFINED

DACL . . . . . . . . . . . . . . D ADDR 00FBH PREDEFINED

DACMSG . . . . . . . . . . . . . C ADDR 0CD4H

DACOUT . . . . . . . . . . . . . D ADDR 0061H

DATAOUT. . . . . . . . . . . . . D ADDR 0064H

DELAY. . . . . . . . . . . . . . C ADDR 0D4EH

DLY0 . . . . . . . . . . . . . . C ADDR 0D4FH

DLY1 . . . . . . . . . . . . . . C ADDR 0D51H

DPH. . . . . . . . . . . . . . . D ADDR 0083H PREDEFINED

DPH1 . . . . . . . . . . . . . . D ADDR 0062H

DPL. . . . . . . . . . . . . . . D ADDR 0082H PREDEFINED

DPL1 . . . . . . . . . . . . . . D ADDR 0063H

DPP. . . . . . . . . . . . . . . D ADDR 0084H PREDEFINED

EA . . . . . . . . . . . . . . . B ADDR 00AFH PREDEFINED

ERROR2A. . . . . . . . . . . . . C ADDR 0247H

ERROR2B. . . . . . . . . . . . . C ADDR 02B0H

ERROR4 . . . . . . . . . . . . . C ADDR 0387H

ERRORMSG . . . . . . . . . . . . C ADDR 0D19H

EX0. . . . . . . . . . . . . . . B ADDR 00A8H PREDEFINED

EXTDATA. . . . . . . . . . . . . C ADDR 01CCH NOT USED

FIRSTGO. . . . . . . . . . . . . B ADDR 0001H

FLAG . . . . . . . . . . . . . . B ADDR 0000H

GOMENU . . . . . . . . . . . . . C ADDR 0C86H

HEX2ASCII. . . . . . . . . . . . C ADDR 0D7EH

ICON . . . . . . . . . . . . . . D ADDR 00D5H PREDEFINED

INTVAL . . . . . . . . . . . . . D ADDR 00A6H PREDEFINED

ISPI . . . . . . . . . . . . . . B ADDR 00FFH PREDEFINED

IT0. . . . . . . . . . . . . . . B ADDR 0088H PREDEFINED

LED. . . . . . . . . . . . . . . NUMB 00B4H

LINK9A . . . . . . . . . . . . . C ADDR 092EH

LINK9B . . . . . . . . . . . . . C ADDR 0963H

LINKS1 . . . . . . . . . . . . . C ADDR 0512H

LINKS2 . . . . . . . . . . . . . C ADDR 05A9H

LK9A . . . . . . . . . . . . . . C ADDR 01EBH NOT USED

816CTEST PAGE 27

LK9B . . . . . . . . . . . . . . C ADDR 025AH NOT USED

MAIN . . . . . . . . . . . . . . C ADDR 0060H

MAINMENU . . . . . . . . . . . . C ADDR 0090H

MEMTITLE . . . . . . . . . . . . C ADDR 08C1H

MENU . . . . . . . . . . . . . . C ADDR 0471H

OKMSG. . . . . . . . . . . . . . C ADDR 0D13H

P1 . . . . . . . . . . . . . . . D ADDR 0090H PREDEFINED

P3 . . . . . . . . . . . . . . . D ADDR 00B0H PREDEFINED

PORT1. . . . . . . . . . . . . . C ADDR 0C45H

PORT3. . . . . . . . . . . . . . C ADDR 0C28H

PRIMADC. . . . . . . . . . . . . C ADDR 0730H

PRINTERROR . . . . . . . . . . . C ADDR 0D47H

PRINTOK. . . . . . . . . . . . . C ADDR 0D40H

RCAP2H . . . . . . . . . . . . . D ADDR 00CBH PREDEFINED

RCAP2L . . . . . . . . . . . . . D ADDR 00CAH PREDEFINED

RDY0 . . . . . . . . . . . . . . B ADDR 00DFH PREDEFINED

RDY1 . . . . . . . . . . . . . . B ADDR 00DEH PREDEFINED

READMSG. . . . . . . . . . . . . C ADDR 0D38H

REMOVELINKS1 . . . . . . . . . . C ADDR 0627H

REMOVELINKS2 . . . . . . . . . . C ADDR 06BAH

RETMENU. . . . . . . . . . . . . C ADDR 0CACH

RI . . . . . . . . . . . . . . . B ADDR 0098H PREDEFINED

RTDLINKS . . . . . . . . . . . . C ADDR 0B20H

RTDSENSOR. . . . . . . . . . . . C ADDR 0CF8H

RTDTITLE . . . . . . . . . . . . C ADDR 0AB9H

SBUF . . . . . . . . . . . . . . D ADDR 0099H PREDEFINED

SCLOCKHIGH . . . . . . . . . . . C ADDR 02E4H

SCLOCKLOW. . . . . . . . . . . . C ADDR 02E9H

SCON . . . . . . . . . . . . . . D ADDR 0098H PREDEFINED

SDATAHIGH. . . . . . . . . . . . C ADDR 030AH

SDATALOW . . . . . . . . . . . . C ADDR 030FH

SDATAMSG . . . . . . . . . . . . C ADDR 0A4DH

SEND12BITS . . . . . . . . . . . C ADDR 0DB5H

SENDCHAR . . . . . . . . . . . . C ADDR 0D76H

SENDDONE . . . . . . . . . . . . C ADDR 0D71H

SENDNEXT . . . . . . . . . . . . C ADDR 0D61H

SENDSTRING . . . . . . . . . . . C ADDR 0D5AH

SENDVAL. . . . . . . . . . . . . C ADDR 0D8AH

SHIFT. . . . . . . . . . . . . . C ADDR 0DA0H NOT USED

SKIP1. . . . . . . . . . . . . . C ADDR 00A1H

SKIP2. . . . . . . . . . . . . . C ADDR 00B5H

SPICON . . . . . . . . . . . . . D ADDR 00F8H PREDEFINED

SPIDAT . . . . . . . . . . . . . D ADDR 00F7H PREDEFINED

SPIERROR . . . . . . . . . . . . C ADDR 0337H

SPIMSG . . . . . . . . . . . . . C ADDR 0A9AH

SPIOK. . . . . . . . . . . . . . C ADDR 032CH NOT USED

SPITITLE . . . . . . . . . . . . C ADDR 0998H

SSMSG. . . . . . . . . . . . . . C ADDR 0A75H

T2CON. . . . . . . . . . . . . . D ADDR 00C8H PREDEFINED

TEST1. . . . . . . . . . . . . . C ADDR 00E2H

TEST2. . . . . . . . . . . . . . C ADDR 01CCH

TEST3. . . . . . . . . . . . . . C ADDR 02C5H

TEST4. . . . . . . . . . . . . . C ADDR 0342H

TEST5. . . . . . . . . . . . . . C ADDR 038CH

TESTAUXCHAN. . . . . . . . . . . C ADDR 016AH

TESTPRIMADC. . . . . . . . . . . C ADDR 0132H

TESTSDATA. . . . . . . . . . . . C ADDR 02EEH

TESTSS . . . . . . . . . . . . . C ADDR 0314H

816CTEST PAGE 28

TH2. . . . . . . . . . . . . . . D ADDR 00CDH PREDEFINED

TI . . . . . . . . . . . . . . . B ADDR 0099H PREDEFINED

TIMECON. . . . . . . . . . . . . D ADDR 00A1H PREDEFINED

TITLE. . . . . . . . . . . . . . C ADDR 03FBH

TL2. . . . . . . . . . . . . . . D ADDR 00CCH PREDEFINED

WAITFORHIGH. . . . . . . . . . . C ADDR 02D9H

WAITFORHIGH1 . . . . . . . . . . C ADDR 02FFH

WAITFORLOW . . . . . . . . . . . C ADDR 02D6H

WAITFORLOW1. . . . . . . . . . . C ADDR 02FCH

WRITTENMSG . . . . . . . . . . . C ADDR 0D2CH

XRAMIN1. . . . . . . . . . . . . C ADDR 01FEH

XRAMIN2. . . . . . . . . . . . . C ADDR 026DH