824UART2 PAGE 1

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

2 ;

3 ; Author : ADI - Apps www.analog.com/MicroConverter

4 ;

5 ; Date : 22 September 1999

6 ;

7 ; File : 824uart.asm

8 ;

9 ; Hardware : ADuC824

10 ;

11 ; Description : sample program that performs ADC conversions in

12 ; continuous mode and sends results to a PC via the

13 ; UART. program also accepts commands from the PC

14 ; via incoming characters on the UART. an ASCII "0"

15 ; innitiates a zero-scale calibration, and an ASCII

16 ; "1" innitiates a full-scale calibration.

17 ;

18 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

19

20 $MOD824 ; Use 8052&ADuC824 predefined symbols

21

00B4 22 LED EQU P3.4 ; P3.4 drives red LED on eval board

23

24 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

25 ; BEGINNING OF CODE

---- 26 CSEG

27

0000 28 ORG 0000h

29

0000 020100 30 JMP MAIN

31

32 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

33 ; INTERRUPT VECTOR SPACE

34

0033 35 ORG 0033h ; (ADC ISR)

36

0033 B2B4 37 CPL LED

38

0035 900209 39 MOV DPTR,#SEPERATOR ; send linefeed+CR out UART

0038 12016E 40 CALL SENDSTRING

41

003B E5DB 42 MOV A,ADC0H ; send ADC data via UART

003D 12018E 43 CALL SENDVAL

0040 E5DA 44 MOV A,ADC0M

0042 12018E 45 CALL SENDVAL

0045 E5D9 46 MOV A,ADC0L

0047 12018E 47 CALL SENDVAL

48

004A C2DF 49 CLR RDY0

004C 32 50 RETI

51

52 ;====================================================================

53 ; MAIN PROGRAM

0100 54 ORG 0100h

55

0100 56 MAIN:

0100 75817F 57 MOV SP,#127

58

824UART2 PAGE 2

59 ; CONFIGURE UART....

60

0103 75CBFF 61 MOV RCAP2H,#0FFh ; config UART for 9830baud

0106 75CAFB 62 MOV RCAP2L,#-5 ; (close enough to 9600baud)

0109 75CDFF 63 MOV TH2,#0FFh

010C 75CCFB 64 MOV TL2,#-5

010F 759852 65 MOV SCON,#01010010b

0112 75C834 66 MOV T2CON,#00110100b

67

68 ; CONFIGURE ADC....

69

0115 75D4C8 70 MOV SF,#200 ; 6.8266667Hz ADC data rate

0118 75D245 71 MOV ADC0CON,#045h ; externalVref, bipolar, ±640mV

011B D2AE 72 SETB EADC ; enable ADC interrupt (trig on RDY0)

73

74 ; START CONVERTING & WAIT FOR INTERRUPTS OR INCOMING UART COMS....

75

011D D2AF 76 WAIT: SETB EA

011F C2DF 77 CLR RDY0

0121 75D123 78 MOV ADCMODE,#023h ; continuous conversion mode

0124 C298 79 CLR RI

0126 3098FD 80 JNB RI,$ ; wait here for UART command..

81 ; ..or ADC interrupt

82

83 ; WHEN UART COMMAND RECEIVED....

84

0129 C2AF 85 CLR EA ; disable interrupts

012B 75D121 86 MOV ADCMODE,#021h ; put ADC in idle mode

012E E599 87 MOV A,SBUF

0130 B43005 88 CJNE A,#'0',NEXT01 ; if "0" received..

0133 120140 89 CALL CALZERO ; ..perform zero calibration

0136 80E5 90 JMP WAIT

0138 B431E2 91 NEXT01: CJNE A,#'1',WAIT ; if "1" received..

013B 120157 92 CALL CALGAIN ; ..perform gain calibration

013E 80DD 93 JMP WAIT

94

95 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

96 ; SUBROUTINES

97

0140 98 CALZERO: ; perform zero-scale ADC calibration

99

0140 90020C 100 MOV DPTR,#CALZSMSG ; send char to indicate..

0143 12016E 101 CALL SENDSTRING ; ..begin zero calibration

102

0146 C2DF 103 CLR RDY0

0148 75D122 104 MOV ADCMODE,#034 ; zero-scale self cal

105 ; MOV ADCMODE,#036h ; zero-scale system cal

014B 30DFFD 106 JNB RDY0,$

014E C2DF 107 CLR RDY0

108

0150 900210 109 MOV DPTR,#CALDONEMSG ; send char to indicate..

0153 12016E 110 CALL SENDSTRING ; ..calibraion complete

111

0156 22 112 RET

113

114 ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

115

0157 116 CALGAIN: ; perform full-scale ADC system calibration

824UART2 PAGE 3

117

0157 90020E 118 MOV DPTR,#CALGNMSG ; send char to indicate..

015A 12016E 119 CALL SENDSTRING ; ..begin gain calibration

120

015D C2DF 121 CLR RDY0

015F 75D123 122 MOV ADCMODE,#035 ; full-scale self cal

123 ; MOV ADCMODE,#037h ; full-scale system cal

0162 30DFFD 124 JNB RDY0,$

0165 C2DF 125 CLR RDY0

126

0167 900210 127 MOV DPTR,#CALDONEMSG ; send char to indicate..

016A 12016E 128 CALL SENDSTRING ; ..calibration complete

129

016D 22 130 RET

131

132 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

133 ; SUBROUTINE INCLUDE FILE

134

=1 135 $INCLUDE(UARTIO.asm)

=1 136 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

=1 137 ;

=1 138 ; Author : ADI - Apps www.analog.com/MicroConverter

=1 139 ;

=1 140 ; Date : 12 October 1999

=1 141 ;

=1 142 ; File : UARTIO.hex

=1 143 ;

=1 144 ; Hardware : any 8051 based microcontroller or MicroConverter

=1 145 ;

=1 146 ; Description : standard UART I/O subroutines. total size of this

=1 147 ; code when assembled is 155 bytes. routines for use

=1 148 ; external to this file are:

=1 149 ;

=1 150 ; SENDSTRING - sends a string of characters

=1 151 ; SENDCHAR - sends a single character

=1 152 ; SENDVAL - sends a byte as 2 ASCII characters

=1 153 ; HEX2ASCII - converts from HEX to ASCII

=1 154 ; ASCII2HEX - converts from ASCII to HEX

=1 155 ; GETCHAR - gets a single character

=1 156 ; GETVAL - gets a byte as 2 ASCII characters

=1 157 ;

=1 158 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

=1 159

=1 160 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

=1 161 ; SENDSTRING

=1 162

016E =1 163 SENDSTRING: ; sends ASCII string to UART starting at location

=1 164 ; DPTR and ending with a null (0) value

=1 165

016E C0E0 =1 166 PUSH ACC

0170 C0F0 =1 167 PUSH B

0172 E4 =1 168 CLR A

0173 F5F0 =1 169 MOV B,A

0175 E5F0 =1 170 IO0010: MOV A,B

0177 05F0 =1 171 INC B

0179 93 =1 172 MOVC A,@A+DPTR

017A 6005 =1 173 JZ IO0020

017C 120186 =1 174 CALL SENDCHAR

824UART2 PAGE 4

017F 80F4 =1 175 JMP IO0010

0181 D0F0 =1 176 IO0020: POP B

0183 D0E0 =1 177 POP ACC

=1 178

0185 22 =1 179 RET

=1 180

=1 181 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

=1 182 ; SENDCHAR

=1 183

0186 =1 184 SENDCHAR: ; sends ASCII value contained in A to UART

=1 185

0186 3099FD =1 186 JNB TI,$ ; wait til present char gone

0189 C299 =1 187 CLR TI ; must clear TI

018B F599 =1 188 MOV SBUF,A

=1 189

018D 22 =1 190 RET

=1 191

=1 192 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

=1 193 ; SENDVAL

=1 194

018E =1 195 SENDVAL: ; converts the hex value of A into two ASCII chars,

=1 196 ; and then spits these two characters up the UART.

=1 197 ; does not change the value of A.

=1 198

018E C0E0 =1 199 PUSH ACC

0190 C4 =1 200 SWAP A

0191 1201A2 =1 201 CALL HEX2ASCII

0194 3186 =1 202 CALL SENDCHAR ; send high nibble

0196 D0E0 =1 203 POP ACC

0198 C0E0 =1 204 PUSH ACC

019A 1201A2 =1 205 CALL HEX2ASCII

019D 3186 =1 206 CALL SENDCHAR ; send low nibble

019F D0E0 =1 207 POP ACC

=1 208

01A1 22 =1 209 RET

=1 210

=1 211 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

=1 212 ; HEX2ASCII

=1 213

01A2 =1 214 HEX2ASCII: ; converts A into the hex character representing the

=1 215 ; value of A's least significant nibble

=1 216

01A2 540F =1 217 ANL A,#00Fh

01A4 B40A00 =1 218 CJNE A,#00Ah,$+3

01A7 4002 =1 219 JC IO0030

01A9 2407 =1 220 ADD A,#007h

01AB 2430 =1 221 IO0030: ADD A,#'0'

=1 222

01AD 22 =1 223 RET

=1 224

=1 225 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

=1 226 ; ASCII2HEX

=1 227

01AE =1 228 ASCII2HEX: ; converts A from an ASCII digit ('0'-'9' or 'A'-'F')

=1 229 ; into the corresponding number (0-15). returns C=1

=1 230 ; when input is other than an ASCII digit,

=1 231 ; indicating invalid output (returned as 255).

=1 232

824UART2 PAGE 5

01AE C3 =1 233 CLR C

01AF 9430 =1 234 SUBB A,#'0'

01B1 B40A00 =1 235 CJNE A,#10,$+3

01B4 401B =1 236 JC IO0050 ; if '0'<=char<='9', return OK

01B6 B41100 =1 237 CJNE A,#17,$+3

01B9 4013 =1 238 JC IO0040 ; if '9'<char<'A', return FAIL

01BB 9407 =1 239 SUBB A,#7

01BD B41000 =1 240 CJNE A,#10h,$+3

01C0 400F =1 241 JC IO0050 ; if 'A'<=char<='F', return OK

01C2 B42A00 =1 242 CJNE A,#42,$+3

01C5 4007 =1 243 JC IO0040 ; if 'F'<char<'a', return FAIL

01C7 9420 =1 244 SUBB A,#20h

01C9 B41000 =1 245 CJNE A,#10h,$+3

01CC 4003 =1 246 JC IO0050 ; if 'a'<=char<='f', return OK..

=1 247

01CE C3 =1 248 IO0040: CLR C ; ..else return FAIL

01CF 74FF =1 249 MOV A,#0FFh

=1 250

01D1 B3 =1 251 IO0050: CPL C

01D2 22 =1 252 RET

=1 253

=1 254 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

=1 255 ; GETCHAR

=1 256

01D3 =1 257 GETCHAR: ; waits for a single ASCII character to be received

=1 258 ; by the UART. places this character into A.

=1 259

01D3 3098FD =1 260 JNB RI,$

01D6 E599 =1 261 MOV A,SBUF

01D8 C298 =1 262 CLR RI

=1 263

01DA 22 =1 264 RET

=1 265

=1 266 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

=1 267 ; GETVAL

=1 268

01DB =1 269 GETVAL: ; waits for two ASCII hex digits to be received by

=1 270 ; the UART. returns the hex value in A.

=1 271

01DB C0F0 =1 272 PUSH B

01DD C000 =1 273 PUSH 0

01DF C298 =1 274 IO0060: CLR RI

01E1 31D3 =1 275 CALL GETCHAR ; first nibble

01E3 F500 =1 276 MOV 0,A ; store received char

01E5 31AE =1 277 CALL ASCII2HEX

01E7 40F6 =1 278 JC IO0060 ; if not '0' thru 'F', don't accept

01E9 C4 =1 279 SWAP A ; swap nibbles

01EA F5F0 =1 280 MOV B,A ; store nibble in B

01EC E500 =1 281 MOV A,0 ; echo received char

01EE 3186 =1 282 CALL SENDCHAR

01F0 C298 =1 283 IO0070: CLR RI

01F2 31D3 =1 284 CALL GETCHAR ; second nibble

01F4 F500 =1 285 MOV 0,A ; store received char

01F6 31AE =1 286 CALL ASCII2HEX

01F8 40F6 =1 287 JC IO0070 ; if not '0' thru 'F', don't accept

01FA 45F0 =1 288 ORL A,B ; combine nibbles

01FC F5F0 =1 289 MOV B,A ; store results in B

01FE E500 =1 290 MOV A,0 ; echo received char

824UART2 PAGE 6

0200 3186 =1 291 CALL SENDCHAR

0202 E5F0 =1 292 MOV A,B ; final result

0204 D000 =1 293 POP 0

0206 D0F0 =1 294 POP B

=1 295

0208 22 =1 296 RET

=1 297

298

299 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

300 ; TEXT DATA TABLES

301

0209 0A0D00 302 SEPERATOR: DB 10,13,0

020C 5A00 303 CALZSMSG: DB 'Z',0

020E 4700 304 CALGNMSG: DB 'G',0

0210 7800 305 CALDONEMSG: DB 'x',0

306

307 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

308

309 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

824UART2 PAGE 7

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

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

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

ADC0L. . . . . . . . . . . . . . D ADDR 00D9H PREDEFINED

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

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

ASCII2HEX. . . . . . . . . . . . C ADDR 01AEH

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

CALDONEMSG . . . . . . . . . . . C ADDR 0210H

CALGAIN. . . . . . . . . . . . . C ADDR 0157H

CALGNMSG . . . . . . . . . . . . C ADDR 020EH

CALZERO. . . . . . . . . . . . . C ADDR 0140H

CALZSMSG . . . . . . . . . . . . C ADDR 020CH

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

EADC . . . . . . . . . . . . . . B ADDR 00AEH PREDEFINED

GETCHAR. . . . . . . . . . . . . C ADDR 01D3H

GETVAL . . . . . . . . . . . . . C ADDR 01DBH NOT USED

HEX2ASCII. . . . . . . . . . . . C ADDR 01A2H

IO0010 . . . . . . . . . . . . . C ADDR 0175H

IO0020 . . . . . . . . . . . . . C ADDR 0181H

IO0030 . . . . . . . . . . . . . C ADDR 01ABH

IO0040 . . . . . . . . . . . . . C ADDR 01CEH

IO0050 . . . . . . . . . . . . . C ADDR 01D1H

IO0060 . . . . . . . . . . . . . C ADDR 01DFH

IO0070 . . . . . . . . . . . . . C ADDR 01F0H

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

MAIN . . . . . . . . . . . . . . C ADDR 0100H

NEXT01 . . . . . . . . . . . . . C ADDR 0138H

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

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

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

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

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

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 0186H

SENDSTRING . . . . . . . . . . . C ADDR 016EH

SENDVAL. . . . . . . . . . . . . C ADDR 018EH

SEPERATOR. . . . . . . . . . . . C ADDR 0209H

SF . . . . . . . . . . . . . . . D ADDR 00D4H PREDEFINED

SP . . . . . . . . . . . . . . . D ADDR 0081H PREDEFINED

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

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

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

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

WAIT . . . . . . . . . . . . . . C ADDR 011DH