11BITSP PAGE 1

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

2 ;

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

4 ;

5 ; Date : May 2002

6 ;

7 ; File : DualDPTR.asm

8 ;

9 ; Hardware : ADuC831

10 ;

11 ; Description : Sample Program to show the new ADuC831 feature

12 ; of the extended 11-bit Stack Pointer

13 ;

14 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

15

16 $MOD831 ; Use 8052&ADuC831 predefined symbols

17

00B4 18 LED EQU P3.4 ; P3.4 drives red LED on eval board

19

20 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

21 ; BEGINNING OF CODE

---- 22 CSEG

23

0000 24 ORG 0000h

25

26 ; Configure UART

0000 759E85 27 MOV T3CON,#85h

0003 759D08 28 MOV T3FD,#08h

0006 759852 29 MOV SCON,#52h

30

31 ; enable the intenal On-Chip XRAM, keep Flash Timing correct

32

0009 43AF01 33 ORL CFG831, #01h

34

000C 9000F8 35 MOV DPTR, #STACK8MSG

000F 1200B6 36 CALL SENDSTRING

37

38 ; initialise SP

0012 7581F0 39 MOV SP, #0F0h

0015 1200A1 40 CALL SENDSP

41

0018 12002F 42 CALL LEVEL1

001B 1200A1 43 CALL SENDSP ; stack should be back at F0 here

44 ; (it will print as F2 because calling SENDSP

45 ; will increment the stack twice)

46

47 ; now enable the stack to rollover into XRAM

001E 43AF80 48 ORL CFG831, #80h

49

0021 90010F 50 MOV DPTR, #STACK11MSG

0024 1200B6 51 CALL SENDSTRING

52

53 ; print out new stack tree

0027 12002F 54 CALL LEVEL1

002A 1200A1 55 CALL SENDSP

002D 80FE 56 JMP $ ; will eventually return to here

57

58

11BITSP PAGE 2

002F 59 LEVEL1:

002F 1200A1 60 CALL SENDSP

0032 120039 61 CALL LEVEL2

0035 1200A1 62 CALL SENDSP

0038 22 63 RET

0039 64 LEVEL2:

0039 1200A1 65 CALL SENDSP

003C 120043 66 CALL LEVEL3

003F 1200A1 67 CALL SENDSP

0042 22 68 RET

0043 69 LEVEL3:

0043 1200A1 70 CALL SENDSP

0046 12004D 71 CALL LEVEL4

0049 1200A1 72 CALL SENDSP

004C 22 73 RET

004D 74 LEVEL4:

004D 1200A1 75 CALL SENDSP

0050 120057 76 CALL LEVEL5

0053 1200A1 77 CALL SENDSP

0056 22 78 RET

0057 79 LEVEL5:

0057 1200A1 80 CALL SENDSP

005A 120061 81 CALL LEVEL6

005D 1200A1 82 CALL SENDSP

0060 22 83 RET

0061 84 LEVEL6:

0061 1200A1 85 CALL SENDSP

0064 12006B 86 CALL LEVEL7

0067 1200A1 87 CALL SENDSP

006A 22 88 RET

006B 89 LEVEL7:

006B 1200A1 90 CALL SENDSP

006E 120075 91 CALL LEVEL8

0071 1200A1 92 CALL SENDSP

0074 22 93 RET

0075 94 LEVEL8:

0075 1200A1 95 CALL SENDSP

0078 12007F 96 CALL LEVEL9

007B 1200A1 97 CALL SENDSP

007E 22 98 RET

007F 99 LEVEL9:

007F 1200A1 100 CALL SENDSP

0082 120089 101 CALL LEVEL10

0085 1200A1 102 CALL SENDSP

0088 22 103 RET

0089 104 LEVEL10:

0089 1200A1 105 CALL SENDSP

008C 120093 106 CALL LEVEL11

008F 1200A1 107 CALL SENDSP

0092 22 108 RET

0093 109 LEVEL11:

0093 1200A1 110 CALL SENDSP

0096 12009D 111 CALL LEVEL12

0099 1200A1 112 CALL SENDSP

009C 22 113 RET

009D 114 LEVEL12:

009D 1200A1 115 CALL SENDSP

00A0 22 116 RET

11BITSP PAGE 3

117

118

119 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

120 ; SENDSP

00A1 121 SENDSP:

122 ; send line feed

00A1 740A 123 MOV A, #10

00A3 1200E4 124 CALL SENDCHAR

00A6 740D 125 MOV A, #13

00A8 1200E4 126 CALL SENDCHAR

127 ; send SP

00AB E5B7 128 MOV A, SPH

00AD 1200CE 129 CALL SENDVAL

00B0 E581 130 MOV A, SP

00B2 1200CE 131 CALL SENDVAL

00B5 22 132 RET

133 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

134 ; SENDSTRING

135

00B6 136 SENDSTRING: ; sends ASCII string to UART starting at location

137 ; DPTR and ending with a null (0) value

138

00B6 C0E0 139 PUSH ACC

00B8 C0F0 140 PUSH B

00BA E4 141 CLR A

00BB F5F0 142 MOV B,A

00BD E5F0 143 IO0010: MOV A,B

00BF 05F0 144 INC B

00C1 93 145 MOVC A,@A+DPTR

00C2 6005 146 JZ IO0020

00C4 1200E4 147 CALL SENDCHAR

00C7 80F4 148 JMP IO0010

00C9 D0F0 149 IO0020: POP B

00CB D0E0 150 POP ACC

151

00CD 22 152 RET

153

154 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

155 ; SENDVAL

156

00CE 157 SENDVAL: ; converts the hex value of A into two ASCII chars,

158 ; and then spits these two characters up the UART.

159 ; does not change the value of A.

160

00CE C0E0 161 PUSH ACC

00D0 C4 162 SWAP A

00D1 1200EC 163 CALL HEX2ASCII

00D4 1200E4 164 CALL SENDCHAR ; send high nibble

00D7 D0E0 165 POP ACC

00D9 C0E0 166 PUSH ACC

00DB 1200EC 167 CALL HEX2ASCII

00DE 1200E4 168 CALL SENDCHAR ; send low nibble

00E1 D0E0 169 POP ACC

170

00E3 22 171 RET

172 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

173 ; SENDCHAR

174

11BITSP PAGE 4

00E4 175 SENDCHAR: ; sends ASCII value contained in A to UART

176

00E4 3099FD 177 JNB TI,$ ; wait til present char gone

00E7 C299 178 CLR TI ; must clear TI

00E9 F599 179 MOV SBUF,A

180

00EB 22 181 RET

182 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

183 ; HEX2ASCII

184

00EC 185 HEX2ASCII: ; converts A into the hex character representing the

186 ; value of A's least significant nibble

187

00EC 540F 188 ANL A,#00Fh

00EE B40A00 189 CJNE A,#00Ah,$+3

00F1 4002 190 JC IO0030

00F3 2407 191 ADD A,#007h

00F5 2430 192 IO0030: ADD A,#'0'

193

00F7 22 194 RET

195

196

197

198 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

199

00F8 0A0A0D38 200 STACK8MSG: DB 10,10,13,'8-BIT STACK POINTER',0

00FC 2D424954

0100 20535441

0104 434B2050

0108 4F494E54

010C 455200

010F 0A0A0D31 201 STACK11MSG: DB 10,10,13,'11-BIT STACK POINTER',0

0113 312D4249

0117 54205354

011B 41434B20

011F 504F494E

0123 54455200

202

203

204 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

11BITSP PAGE 5

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

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

CFG831 . . . . . . . . . . . . . D ADDR 00AFH PREDEFINED

HEX2ASCII. . . . . . . . . . . . C ADDR 00ECH

IO0010 . . . . . . . . . . . . . C ADDR 00BDH

IO0020 . . . . . . . . . . . . . C ADDR 00C9H

IO0030 . . . . . . . . . . . . . C ADDR 00F5H

LED. . . . . . . . . . . . . . . NUMB 00B4H NOT USED

LEVEL1 . . . . . . . . . . . . . C ADDR 002FH

LEVEL10. . . . . . . . . . . . . C ADDR 0089H

LEVEL11. . . . . . . . . . . . . C ADDR 0093H

LEVEL12. . . . . . . . . . . . . C ADDR 009DH

LEVEL2 . . . . . . . . . . . . . C ADDR 0039H

LEVEL3 . . . . . . . . . . . . . C ADDR 0043H

LEVEL4 . . . . . . . . . . . . . C ADDR 004DH

LEVEL5 . . . . . . . . . . . . . C ADDR 0057H

LEVEL6 . . . . . . . . . . . . . C ADDR 0061H

LEVEL7 . . . . . . . . . . . . . C ADDR 006BH

LEVEL8 . . . . . . . . . . . . . C ADDR 0075H

LEVEL9 . . . . . . . . . . . . . C ADDR 007FH

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

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 00E4H

SENDSP . . . . . . . . . . . . . C ADDR 00A1H

SENDSTRING . . . . . . . . . . . C ADDR 00B6H

SENDVAL. . . . . . . . . . . . . C ADDR 00CEH

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

SPH. . . . . . . . . . . . . . . D ADDR 00B7H PREDEFINED

STACK11MSG . . . . . . . . . . . C ADDR 010FH

STACK8MSG. . . . . . . . . . . . C ADDR 00F8H

T3CON. . . . . . . . . . . . . . D ADDR 009EH PREDEFINED

T3FD . . . . . . . . . . . . . . D ADDR 009DH PREDEFINED

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