11BITSP PAGE 1

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

2 ;

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

4 ;

5 ; Date : April 2001

6 ;

7 ; File : DualDPTR.asm

8 ;

9 ; Hardware : ADuC836

10 ;

11 ; Description : Sample Program to show the new ADuC836 feature

12 ; of the extended 11-bit Stack Pointer

13 ;

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

15

16 $MOD836 ; Use 8052&ADuC836 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 759E82 27 MOV T3CON,#82h

0003 759D12 28 MOV T3FD,#12h

0006 759852 29 MOV SCON,#52h

30

31 ; enable the intenal On-Chip XRAM

0009 75AF01 32 MOV CFG836, #01h

33

000C 9000F8 34 MOV DPTR, #STACK8MSG

000F 1200B6 35 CALL SENDSTRING

36

37 ; initialise SP

0012 7581F0 38 MOV SP, #0F0h

0015 1200A1 39 CALL SENDSP

40

0018 12002F 41 CALL LEVEL1

001B 1200A1 42 CALL SENDSP ; stack should be back at F0 here

43 ; (it will print as F2 because calling SENDSP

44 ; will increment the stack twice)

45

46 ; now enable the stack to rollover into XRAM

001E 75AF81 47 MOV CFG836, #81h

48

0021 90010F 49 MOV DPTR, #STACK11MSG

0024 1200B6 50 CALL SENDSTRING

51

52 ; print out new stack tree

0027 12002F 53 CALL LEVEL1

002A 1200A1 54 CALL SENDSP

002D 80FE 55 JMP $ ; will eventually return to here

56

57

002F 58 LEVEL1:

11BITSP PAGE 2

002F 1200A1 59 CALL SENDSP

0032 120039 60 CALL LEVEL2

0035 1200A1 61 CALL SENDSP

0038 22 62 RET

0039 63 LEVEL2:

0039 1200A1 64 CALL SENDSP

003C 120043 65 CALL LEVEL3

003F 1200A1 66 CALL SENDSP

0042 22 67 RET

0043 68 LEVEL3:

0043 1200A1 69 CALL SENDSP

0046 12004D 70 CALL LEVEL4

0049 1200A1 71 CALL SENDSP

004C 22 72 RET

004D 73 LEVEL4:

004D 1200A1 74 CALL SENDSP

0050 120057 75 CALL LEVEL5

0053 1200A1 76 CALL SENDSP

0056 22 77 RET

0057 78 LEVEL5:

0057 1200A1 79 CALL SENDSP

005A 120061 80 CALL LEVEL6

005D 1200A1 81 CALL SENDSP

0060 22 82 RET

0061 83 LEVEL6:

0061 1200A1 84 CALL SENDSP

0064 12006B 85 CALL LEVEL7

0067 1200A1 86 CALL SENDSP

006A 22 87 RET

006B 88 LEVEL7:

006B 1200A1 89 CALL SENDSP

006E 120075 90 CALL LEVEL8

0071 1200A1 91 CALL SENDSP

0074 22 92 RET

0075 93 LEVEL8:

0075 1200A1 94 CALL SENDSP

0078 12007F 95 CALL LEVEL9

007B 1200A1 96 CALL SENDSP

007E 22 97 RET

007F 98 LEVEL9:

007F 1200A1 99 CALL SENDSP

0082 120089 100 CALL LEVEL10

0085 1200A1 101 CALL SENDSP

0088 22 102 RET

0089 103 LEVEL10:

0089 1200A1 104 CALL SENDSP

008C 120093 105 CALL LEVEL11

008F 1200A1 106 CALL SENDSP

0092 22 107 RET

0093 108 LEVEL11:

0093 1200A1 109 CALL SENDSP

0096 12009D 110 CALL LEVEL12

0099 1200A1 111 CALL SENDSP

009C 22 112 RET

009D 113 LEVEL12:

009D 1200A1 114 CALL SENDSP

00A0 22 115 RET

116

11BITSP PAGE 3

117

118 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

119 ; SENDSP

00A1 120 SENDSP:

121 ; send line feed

00A1 740A 122 MOV A, #10

00A3 1200E4 123 CALL SENDCHAR

00A6 740D 124 MOV A, #13

00A8 1200E4 125 CALL SENDCHAR

126 ; send SP

00AB E5B7 127 MOV A, SPH

00AD 1200CE 128 CALL SENDVAL

00B0 E581 129 MOV A, SP

00B2 1200CE 130 CALL SENDVAL

00B5 22 131 RET

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

133 ; SENDSTRING

134

00B6 135 SENDSTRING: ; sends ASCII string to UART starting at location

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

137

00B6 C0E0 138 PUSH ACC

00B8 C0F0 139 PUSH B

00BA E4 140 CLR A

00BB F5F0 141 MOV B,A

00BD E5F0 142 IO0010: MOV A,B

00BF 05F0 143 INC B

00C1 93 144 MOVC A,@A+DPTR

00C2 6005 145 JZ IO0020

00C4 1200E4 146 CALL SENDCHAR

00C7 80F4 147 JMP IO0010

00C9 D0F0 148 IO0020: POP B

00CB D0E0 149 POP ACC

150

00CD 22 151 RET

152

153 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

154 ; SENDVAL

155

00CE 156 SENDVAL: ; converts the hex value of A into two ASCII chars,

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

158 ; does not change the value of A.

159

00CE C0E0 160 PUSH ACC

00D0 C4 161 SWAP A

00D1 1200EC 162 CALL HEX2ASCII

00D4 1200E4 163 CALL SENDCHAR ; send high nibble

00D7 D0E0 164 POP ACC

00D9 C0E0 165 PUSH ACC

00DB 1200EC 166 CALL HEX2ASCII

00DE 1200E4 167 CALL SENDCHAR ; send low nibble

00E1 D0E0 168 POP ACC

169

00E3 22 170 RET

171 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

172 ; SENDCHAR

173

00E4 174 SENDCHAR: ; sends ASCII value contained in A to UART

11BITSP PAGE 4

175

00E4 3099FD 176 JNB TI,$ ; wait til present char gone

00E7 C299 177 CLR TI ; must clear TI

00E9 F599 178 MOV SBUF,A

179

00EB 22 180 RET

181 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

182 ; HEX2ASCII

183

00EC 184 HEX2ASCII: ; converts A into the hex character representing the

185 ; value of A's least significant nibble

186

00EC 540F 187 ANL A,#00Fh

00EE B40A00 188 CJNE A,#00Ah,$+3

00F1 4002 189 JC IO0030

00F3 2407 190 ADD A,#007h

00F5 2430 191 IO0030: ADD A,#'0'

192

00F7 22 193 RET

194

195

196

197 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

198

00F8 0A0A0D38 199 STACK8MSG: DB 10,10,13,'8-BIT STACK POINTER',0

00FC 2D424954

0100 20535441

0104 434B2050

0108 4F494E54

010C 455200

010F 0A0A0D31 200 STACK11MSG: DB 10,10,13,'11-BIT STACK POINTER',0

0113 312D4249

0117 54205354

011B 41434B20

011F 504F494E

0123 54455200

201

202

203 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

11BITSP PAGE 5

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

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

CFG836 . . . . . . . . . . . . . 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