UART1 PAGE 1

1 ;====================================================================

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : November 2001

6 ;

7 ; File : UART.asm

8 ;

9 ; Hardware : ADuC836

10 ;

11 ; Description : This Program saves 16 numbers in order initially

12 ; starting with 0 into memory locations 40h to 50h.

13 ; When finished the values in these locations are

14 ; transmitted down the UART in ASCII form to the PC

15 ; where they can be viewed using the preconfigured

16 ; Hyperterminal program. (c:\ADuC\9600com1.ht)

17 ;

18 ; After the transmission of the 16 bytes a 5 second

19 ; delay is called and the process is repeated, this

20 ; time starting with the saving of 10h to location

21 ; 40h.

22 ;

23 ;====================================================================

24 ;

25 $MOD836 ;Use 8052 predefined Symbols

26

00B4 27 LED EQU P3.4

28

29 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

30 ; BEGINNING OF CODE

---- 31 CSEG

0000 32 ORG 0000H

33

0000 020060 34 JMP MAIN

35

0060 36 ORG 0060H ; Start code at address above interrupts

37

38

0060 39 MAIN: ; Main program

40 ; configure UART for 9600 using Timer3

0060 759E82 41 MOV T3CON,#82h

0063 759D12 42 MOV T3FD,#12h

0066 759852 43 MOV SCON,#52h

44

0069 7800 45 MOV R0, #00 ; start output data at 0

006B 7940 46 MOV R1, #40h ; initialise R1 to 40 to store the

47 ; input data from memory location 40

006D 48 SAVENOS:

006D E8 49 MOV A,R0

006E F7 50 MOV @R1, A ; move R0 into memory location R1

006F 09 51 INC R1 ; increment memory location and data so

52 ; new data is stored in new address

0070 08 53 INC R0

0071 B950F9 54 CJNE R1, #50H, SAVENOS ; reset memory location to 40h

55 ; when memory location reaches 50h

56 ; saving 16 bytes of data

57

58 ; Transmit the values in locations 40h->50h up the UART wait for

UART1 PAGE 2

59 ; 5 seconds and then repeat

60

61

0074 B2B4 62 START: CPL LED ;CPL LED with each transmission

0076 9000ED 63 MOV DPTR, #TITLE

0079 1200A1 64 CALL SENDSTRING ; write title block on screen

65

007C 7940 66 MOV R1, #40h ; move value at address 40 into R2

007E E7 67 MOV A, @R1

007F FA 68 MOV R2, A

69

0080 70 NEXT: ; Put new value on a new line

0080 740A 71 MOV A, #10 ; Transmit a linefeed (= ASCII 10)

0082 1200B9 72 CALL SENDCHAR

0085 740D 73 MOV A, #13 ;Transmit a carriage return (=ASCII 13)

0087 1200B9 74 CALL SENDCHAR

75

008A EA 76 MOV A, R2 ; Transmit R2 i.e. value @ address R1

008B 1200C1 77 CALL SENDVAL

008E 09 78 INC R1 ; Increment address

008F E7 79 MOV A, @R1

0090 FA 80 MOV R2, A ; R2 holds the value @ addrR1

81

0091 E9 82 MOV A, R1 ; Check if at address 50h

0092 B450EB 83 CJNE A, #50h, NEXT ; if not jump to Next

0095 020098 84 JMP WAIT5S ; if so wait 5s and repeat

85

0098 7432 86 WAIT5S: MOV A, #50

009A 1200E1 87 CALL DELAY ; Wait 5 seconds

009D 7940 88 MOV R1, #40h

009F 80CC 89 JMP SAVENOS ; Resave new numbers to same addresses

90

91

92 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

93 ; SENDSTRING

94

00A1 95 SENDSTRING: ; sends ASCII string to UART starting at location

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

97

00A1 C0E0 98 PUSH ACC

00A3 C0F0 99 PUSH B

00A5 E4 100 CLR A

00A6 F5F0 101 MOV B,A

00A8 E5F0 102 IO0010: MOV A,B

00AA 05F0 103 INC B

00AC 93 104 MOVC A,@A+DPTR

00AD 6005 105 JZ IO0020

00AF 1200B9 106 CALL SENDCHAR

00B2 80F4 107 JMP IO0010

00B4 D0F0 108 IO0020: POP B

00B6 D0E0 109 POP ACC

110

00B8 22 111 RET

112

113 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

114 ; SENDCHAR

115

00B9 116 SENDCHAR: ; sends ASCII value contained in A to UART

UART1 PAGE 3

117

00B9 3099FD 118 JNB TI,$ ; wait til present char gone

00BC C299 119 CLR TI ; must clear TI

00BE F599 120 MOV SBUF,A

121

00C0 22 122 RET

123

124 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

125 ; SENDVAL

126

00C1 127 SENDVAL: ; converts the hex value of A into two ASCII chars,

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

129 ; does not change the value of A.

130

00C1 C0E0 131 PUSH ACC

00C3 C4 132 SWAP A

00C4 1200D5 133 CALL HEX2ASCII

00C7 11B9 134 CALL SENDCHAR ; send high nibble

00C9 D0E0 135 POP ACC

00CB C0E0 136 PUSH ACC

00CD 1200D5 137 CALL HEX2ASCII

00D0 11B9 138 CALL SENDCHAR ; send low nibble

00D2 D0E0 139 POP ACC

140

00D4 22 141 RET

142

143

144 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

145 ; HEX2ASCII

146

00D5 147 HEX2ASCII: ; converts A into the hex character representing the

148 ; value of A's least significant nibble

149

00D5 540F 150 ANL A,#00Fh

00D7 B40A00 151 CJNE A,#00Ah,$+3

00DA 4002 152 JC IO0030

00DC 2407 153 ADD A,#007h

00DE 2430 154 IO0030: ADD A,#'0'

155

00E0 22 156 RET

157

158 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

159 ; DELAY

160

00E1 161 DELAY: ; Delays by 100ms \* A

162 ; 100mSec based on 1.5728MHZ

163 ; Core Clock

164 ; i.e. default ADuC824 Clock

165

00E1 F9 166 MOV R1,A ; Acc holds delay variable

00E2 7A19 167 DLY0: MOV R2,#019h ; Set up delay loop0

00E4 7BFE 168 DLY1: MOV R3,#0FEh ; Set up delay loop1

00E6 DBFE 169 DJNZ R3,$ ; Dec R3 & Jump here until R2 is 0

00E8 DAFA 170 DJNZ R2,DLY1 ; Dec R2 & Jump DLY1 until R1 is 0

00EA D9F6 171 DJNZ R1,DLY0 ; Dec R1 & Jump DLY0 until R0 is 0

00EC 22 172 RET ; Return from subroutine

173

174 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

UART1 PAGE 4

175

176

00ED 0A0A0D5F 177 TITLE: DB 10,10,13,'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_',10,13

00F1 5F5F5F5F

00F5 5F5F5F5F

00F9 5F5F5F5F

00FD 5F5F5F5F

0101 5F5F5F5F

0105 5F5F5F5F

0109 5F5F5F5F

010D 5F5F5F5F

0111 5F5F5F0A

0115 0D

0116 416E616C 178 DB 'Analog Devices MicroConverter ADuC836',10,13

011A 6F672044

011E 65766963

0122 6573204D

0126 6963726F

012A 436F6E76

012E 65727465

0132 72204144

0136 75433833

013A 360A0D

013D 20202020 179 DB ' UART Demo Routine',10,13

0141 20202020

0145 20554152

0149 54204465

014D 6D6F2052

0151 6F757469

0155 6E650A0D

0159 20204461 180 DB ' Data Stored in Memory in Hex Form',10,13,0

015D 74612053

0161 746F7265

0165 6420696E

0169 204D656D

016D 6F727920

0171 696E2048

0175 65782046

0179 6F726D0A

017D 0D00

181

182

183 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

UART1 PAGE 5

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

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

DELAY. . . . . . . . . . . . . . C ADDR 00E1H

DLY0 . . . . . . . . . . . . . . C ADDR 00E2H

DLY1 . . . . . . . . . . . . . . C ADDR 00E4H

HEX2ASCII. . . . . . . . . . . . C ADDR 00D5H

IO0010 . . . . . . . . . . . . . C ADDR 00A8H

IO0020 . . . . . . . . . . . . . C ADDR 00B4H

IO0030 . . . . . . . . . . . . . C ADDR 00DEH

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

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

NEXT . . . . . . . . . . . . . . C ADDR 0080H

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

SAVENOS. . . . . . . . . . . . . C ADDR 006DH

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 00B9H

SENDSTRING . . . . . . . . . . . C ADDR 00A1H

SENDVAL. . . . . . . . . . . . . C ADDR 00C1H

START. . . . . . . . . . . . . . C ADDR 0074H NOT USED

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

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

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

TITLE. . . . . . . . . . . . . . C ADDR 00EDH

WAIT5S . . . . . . . . . . . . . C ADDR 0098H