UART1 PAGE 1

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

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : April 2002

6 ;

7 ; File : UART.asm

8 ;

9 ; Hardware : ADuC832

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\_Beta832\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 $MOD832 ;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

41

42 ;Configure uart for 9600 baud at core clk = 2.097152MHz

0060 759E82 43 MOV T3CON,#82h

0063 759D2D 44 MOV T3FD,#2Dh

0066 759852 45 MOV SCON,#52h

46

0069 7800 47 MOV R0, #00 ; start output data at 0

006B 7940 48 MOV R1, #40h ; initialise R1 to 40 to store the

49 ; input data from memory location 40

006D 50 SAVENOS:

006D E8 51 MOV A,R0

006E F7 52 MOV @R1, A ; move R0 into memory location R1

006F 09 53 INC R1 ; increment memory location and data so

54 ; new data is stored in new address

0070 08 55 INC R0

0071 B950F9 56 CJNE R1, #50H, SAVENOS ; reset memory location to 40h

57 ; when memory location reaches 50h

58 ; saving 16 bytes of data

UART1 PAGE 2

59

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

61 ; 5 seconds and then repeat

62

63

0074 B2B4 64 START: CPL LED ;CPL LED with each transmission

0076 9000ED 65 MOV DPTR, #TITLE

0079 1200A1 66 CALL SENDSTRING ; write title block on screen

67

007C 7940 68 MOV R1, #40h ; move value at address 40 into R2

007E E7 69 MOV A, @R1

007F FA 70 MOV R2, A

71

0080 72 NEXT: ; Put new value on a new line

0080 740A 73 MOV A, #10 ; Transmit a linefeed (= ASCII 10)

0082 1200B9 74 CALL SENDCHAR

0085 740D 75 MOV A, #13 ;Transmit a carriage return (=ASCII 13)

0087 1200B9 76 CALL SENDCHAR

77

008A EA 78 MOV A, R2 ; Transmit R2 i.e. value @ address R1

008B 1200C1 79 CALL SENDVAL

008E 09 80 INC R1 ; Increment address

008F E7 81 MOV A, @R1

0090 FA 82 MOV R2, A ; R2 holds the value @ addrR1

83

0091 E9 84 MOV A, R1 ; Check if at address 50h

0092 B450EB 85 CJNE A, #50h, NEXT ; if not jump to Next

0095 020098 86 JMP WAIT5S ; if so wait 5s and repeat

87

0098 7432 88 WAIT5S: MOV A, #50

009A 1200E1 89 CALL DELAY ; Wait 5 seconds

009D 7940 90 MOV R1, #40h

009F 80CC 91 JMP SAVENOS ; Resave new numbers to same addresses

92

93

94 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

95 ; SENDSTRING

96

00A1 97 SENDSTRING: ; sends ASCII string to UART starting at location

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

99

00A1 C0E0 100 PUSH ACC

00A3 C0F0 101 PUSH B

00A5 E4 102 CLR A

00A6 F5F0 103 MOV B,A

00A8 E5F0 104 IO0010: MOV A,B

00AA 05F0 105 INC B

00AC 93 106 MOVC A,@A+DPTR

00AD 6005 107 JZ IO0020

00AF 1200B9 108 CALL SENDCHAR

00B2 80F4 109 JMP IO0010

00B4 D0F0 110 IO0020: POP B

00B6 D0E0 111 POP ACC

112

00B8 22 113 RET

114

115 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

116 ; SENDCHAR

UART1 PAGE 3

117

00B9 118 SENDCHAR: ; sends ASCII value contained in A to UART

119

00B9 3099FD 120 JNB TI,$ ; wait til present char gone

00BC C299 121 CLR TI ; must clear TI

00BE F599 122 MOV SBUF,A

123

00C0 22 124 RET

125

126 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

127 ; SENDVAL

128

00C1 129 SENDVAL: ; converts the hex value of A into two ASCII chars,

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

131 ; does not change the value of A.

132

00C1 C0E0 133 PUSH ACC

00C3 C4 134 SWAP A

00C4 1200D5 135 CALL HEX2ASCII

00C7 11B9 136 CALL SENDCHAR ; send high nibble

00C9 D0E0 137 POP ACC

00CB C0E0 138 PUSH ACC

00CD 1200D5 139 CALL HEX2ASCII

00D0 11B9 140 CALL SENDCHAR ; send low nibble

00D2 D0E0 141 POP ACC

142

00D4 22 143 RET

144

145

146 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

147 ; HEX2ASCII

148

00D5 149 HEX2ASCII: ; converts A into the hex character representing the

150 ; value of A's least significant nibble

151

00D5 540F 152 ANL A,#00Fh

00D7 B40A00 153 CJNE A,#00Ah,$+3

00DA 4002 154 JC IO0030

00DC 2407 155 ADD A,#007h

00DE 2430 156 IO0030: ADD A,#'0'

157

00E0 22 158 RET

159

160 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

161 ; DELAY

00E1 162 DELAY: ; Delays by 100ms \* A

163 ; 100mSec based on 2.097152MHZ

164 ; Core Clock

165 ; i.e. default ADuC832 Clock

166

00E1 FD 167 MOV R5,A ; Acc holds delay variable

00E2 7E22 168 DLY0: MOV R6,#022h ; Set up delay loop0

00E4 7FFF 169 DLY1: MOV R7,#0FFh ; Set up delay loop1

00E6 DFFE 170 DJNZ R7,$ ; Dec R2 until R2 is zero

00E8 DEFA 171 DJNZ R6,DLY1 ; Dec R1 & Jump DLY1 until R1 is 0

00EA DDF6 172 DJNZ R5,DLY0 ; Dec R0 & Jump DLY0 until R0 is 0

00EC 22 173 RET ; Return from subroutine

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 ADuC832',10,13

011A 6F672044

011E 65766963

0122 6573204D

0126 6963726F

012A 436F6E76

012E 65727465

0132 72204144

0136 75433833

013A 320A0D

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