UART1 PAGE 1

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

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : January 2001

6 ;

7 ; File : UART1.asm

8 ;

9 ; Hardware : ADuC816

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 $MOD52 ;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

0060 75CBFF 41 MOV RCAP2H,#0FFh ; config UART for 9830baud

0063 75CAFB 42 MOV RCAP2L,#-5 ; (close enough to 9600baud)

0066 75CDFF 43 MOV TH2,#0FFh

0069 75CCFB 44 MOV TL2,#-5

006C 759852 45 MOV SCON,#52h

006F 75C834 46 MOV T2CON,#34h

47

0072 7800 48 MOV R0, #00 ; start output data at 0

0074 7940 49 MOV R1, #40h ; initialise R1 to 40 to store the

50 ; input data from memory location 40

0076 51 SAVENOS:

0076 E8 52 MOV A,R0

0077 F7 53 MOV @R1, A ; move R0 into memory location R1

0078 09 54 INC R1 ; increment memory location and data so

55 ; new data is stored in new address

0079 08 56 INC R0

007A B950F9 57 CJNE R1, #50H, SAVENOS ; reset memory location to 40h

58 ; when memory location reaches 50h

UART1 PAGE 2

59 ; saving 16 bytes of data

60

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

62 ; 5 seconds and then repeat

63

64

007D B2B4 65 START: CPL LED ;CPL LED with each transmission

007F 9000F6 66 MOV DPTR, #TITLE

0082 1200AA 67 CALL SENDSTRING ; write title block on screen

68

0085 7940 69 MOV R1, #40h ; move value at address 40 into R2

0087 E7 70 MOV A, @R1

0088 FA 71 MOV R2, A

72

0089 73 NEXT: ; Put new value on a new line

0089 740A 74 MOV A, #10 ; Transmit a linefeed (= ASCII 10)

008B 1200C2 75 CALL SENDCHAR

008E 740D 76 MOV A, #13 ;Transmit a carriage return (=ASCII 13)

0090 1200C2 77 CALL SENDCHAR

78

0093 EA 79 MOV A, R2 ; Transmit R2 i.e. value @ address R1

0094 1200CA 80 CALL SENDVAL

0097 09 81 INC R1 ; Increment address

0098 E7 82 MOV A, @R1

0099 FA 83 MOV R2, A ; R2 holds the value @ addrR1

84

009A E9 85 MOV A, R1 ; Check if at address 50h

009B B450EB 86 CJNE A, #50h, NEXT ; if not jump to Next

009E 0200A1 87 JMP WAIT5S ; if so wait 5s and repeat

88

00A1 7432 89 WAIT5S: MOV A, #50

00A3 1200EA 90 CALL DELAY ; Wait 5 seconds

00A6 7940 91 MOV R1, #40h

00A8 80CC 92 JMP SAVENOS ; Resave new numbers to same addresses

93

94

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

96 ; SENDSTRING

97

00AA 98 SENDSTRING: ; sends ASCII string to UART starting at location

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

100

00AA C0E0 101 PUSH ACC

00AC C0F0 102 PUSH B

00AE E4 103 CLR A

00AF F5F0 104 MOV B,A

00B1 E5F0 105 IO0010: MOV A,B

00B3 05F0 106 INC B

00B5 93 107 MOVC A,@A+DPTR

00B6 6005 108 JZ IO0020

00B8 1200C2 109 CALL SENDCHAR

00BB 80F4 110 JMP IO0010

00BD D0F0 111 IO0020: POP B

00BF D0E0 112 POP ACC

113

00C1 22 114 RET

115

116 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

UART1 PAGE 3

117 ; SENDCHAR

118

00C2 119 SENDCHAR: ; sends ASCII value contained in A to UART

120

00C2 3099FD 121 JNB TI,$ ; wait til present char gone

00C5 C299 122 CLR TI ; must clear TI

00C7 F599 123 MOV SBUF,A

124

00C9 22 125 RET

126

127 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

128 ; SENDVAL

129

00CA 130 SENDVAL: ; converts the hex value of A into two ASCII chars,

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

132 ; does not change the value of A.

133

00CA C0E0 134 PUSH ACC

00CC C4 135 SWAP A

00CD 1200DE 136 CALL HEX2ASCII

00D0 11C2 137 CALL SENDCHAR ; send high nibble

00D2 D0E0 138 POP ACC

00D4 C0E0 139 PUSH ACC

00D6 1200DE 140 CALL HEX2ASCII

00D9 11C2 141 CALL SENDCHAR ; send low nibble

00DB D0E0 142 POP ACC

143

00DD 22 144 RET

145

146

147 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

148 ; HEX2ASCII

149

00DE 150 HEX2ASCII: ; converts A into the hex character representing the

151 ; value of A's least significant nibble

152

00DE 540F 153 ANL A,#00Fh

00E0 B40A00 154 CJNE A,#00Ah,$+3

00E3 4002 155 JC IO0030

00E5 2407 156 ADD A,#007h

00E7 2430 157 IO0030: ADD A,#'0'

158

00E9 22 159 RET

160

161 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

162 ; DELAY

163

00EA 164 DELAY: ; Delays by 100ms \* A

165 ; 100mSec based on 1.5728MHZ

166 ; Core Clock

167 ; i.e. default ADuC824 Clock

168

00EA F9 169 MOV R1,A ; Acc holds delay variable

00EB 7A19 170 DLY0: MOV R2,#019h ; Set up delay loop0

00ED 7BFE 171 DLY1: MOV R3,#0FEh ; Set up delay loop1

00EF DBFE 172 DJNZ R3,$ ; Dec R3 & Jump here until R2 is 0

00F1 DAFA 173 DJNZ R2,DLY1 ; Dec R2 & Jump DLY1 until R1 is 0

00F3 D9F6 174 DJNZ R1,DLY0 ; Dec R1 & Jump DLY0 until R0 is 0

UART1 PAGE 4

00F5 22 175 RET ; Return from subroutine

176

177 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

178

179

00F6 0A0A0D5F 180 TITLE: DB 10,10,13,'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_',10,13

00FA 5F5F5F5F

00FE 5F5F5F5F

0102 5F5F5F5F

0106 5F5F5F5F

010A 5F5F5F5F

010E 5F5F5F5F

0112 5F5F5F5F

0116 5F5F5F5F

011A 5F5F5F0A

011E 0D

011F 416E616C 181 DB 'Analog Devices MicroConverter ADuC816',10,13

0123 6F672044

0127 65766963

012B 6573204D

012F 6963726F

0133 436F6E76

0137 65727465

013B 72204144

013F 75433831

0143 360A0D

0146 20202020 182 DB ' UART Demo Routine',10,13

014A 20202020

014E 20554152

0152 54204465

0156 6D6F2052

015A 6F757469

015E 6E650A0D

0162 20204461 183 DB ' Data Stored in Memory in Hex Form',10,13,0

0166 74612053

016A 746F7265

016E 6420696E

0172 204D656D

0176 6F727920

017A 696E2048

017E 65782046

0182 6F726D0A

0186 0D00

184

185

186 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 00EAH

DLY0 . . . . . . . . . . . . . . C ADDR 00EBH

DLY1 . . . . . . . . . . . . . . C ADDR 00EDH

HEX2ASCII. . . . . . . . . . . . C ADDR 00DEH

IO0010 . . . . . . . . . . . . . C ADDR 00B1H

IO0020 . . . . . . . . . . . . . C ADDR 00BDH

IO0030 . . . . . . . . . . . . . C ADDR 00E7H

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

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

NEXT . . . . . . . . . . . . . . C ADDR 0089H

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

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

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

SAVENOS. . . . . . . . . . . . . C ADDR 0076H

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 00C2H

SENDSTRING . . . . . . . . . . . C ADDR 00AAH

SENDVAL. . . . . . . . . . . . . C ADDR 00CAH

START. . . . . . . . . . . . . . C ADDR 007DH NOT USED

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

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

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

TITLE. . . . . . . . . . . . . . C ADDR 00F6H

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

WAIT5S . . . . . . . . . . . . . C ADDR 00A1H