UART2 PAGE 1

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

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : January 2001

6 ;

7 ; File : UART2.asm

8 ;

9 ; Hardware : ADuC816

10 ;

11 ; Description : This Program transmits the numbers 0->7F (starting

12 ; with 0) down the UART in ASCII form to the PC where

13 ; they can be viewed using the preconfigured

14 ; Hyperterminal program. (c:\ADuC\_Beta816\9600com1.ht)

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

16 ; delay is called and the process is repeated.

17 ;

18 ;====================================================================

19 ;

20 $MOD52 ;Use 8052 predefined Symbols

21

00B4 22 LED EQU P3.4

23

24 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

25 ; BEGINNING OF CODE

---- 26 CSEG

0000 27 ORG 0000H

28

0000 020060 29 JMP MAIN

30

0060 31 ORG 0060H ; Start code at address above interrupts

32

33

0060 34 MAIN: ; Main program

35

0060 75CBFF 36 MOV RCAP2H,#0FFh ; config UART for 9830baud

0063 75CAFB 37 MOV RCAP2L,#-5 ; (close enough to 9600baud)

0066 75CDFF 38 MOV TH2,#0FFh

0069 75CCFB 39 MOV TL2,#-5

006C 759852 40 MOV SCON,#52h

006F 75C834 41 MOV T2CON,#34h

42

0072 43 START:

0072 B2B4 44 CPL LED ; CPL LED with each transmission

0074 9000F4 45 MOV DPTR, #TITLE

0077 1200A8 46 CALL SENDSTRING ; write title block on screen

47

007A 7800 48 MOV R0, #00H ; Start transmissions from 0

007C 7908 49 MOV R1, #08H ; Start a new line after 8 transmissions

50

007E 51 LOOP1: ; Every eight transmissions start on a

52 ; new line

007E 740A 53 MOV A, #10 ; Transmit a linefeed

0080 1200C0 54 CALL SENDCHAR

0083 740D 55 MOV A, #13 ; Transmit a carriage return

0085 1200C0 56 CALL SENDCHAR

57

0088 7908 58 MOV R1, #08H

UART2 PAGE 2

59

008A 60 LOOP2:

008A 7420 61 MOV A, #20H ; Transmit a SPACE (=ASCII 20) between

62 ; transmissions on same line

008C 1200C0 63 CALL SENDCHAR

64

008F E8 65 MOV A, R0 ; Transmit R0 = present data

0090 1200C8 66 CALL SENDVAL

0093 08 67 INC R0 ; increment data

68

0094 B47F03 69 CJNE A, #7FH, CONT ; check if data =7F, if no continue

0097 0200A1 70 JMP WAIT5S ; if = 7F wait 5s and repeat

71

009A 19 72 CONT: DEC R1 ; decrement R1....

009B E9 73 MOV A, R1

009C B400EB 74 CJNE A, #00H, LOOP2 ; and check if new line is required

75 ; jump to loop 2 for a space

009F 80DD 76 JMP LOOP1 ; jump to loop 1 for a new line

77

78

00A1 7432 79 WAIT5S: MOV A, #50 ; wait 5s

00A3 1200E8 80 CALL DELAY

00A6 80CA 81 JMP START ; start transmissions again

82

83

84 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

85 ; SENDSTRING

86

00A8 87 SENDSTRING: ; sends ASCII string to UART starting at location

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

89

00A8 C0E0 90 PUSH ACC

00AA C0F0 91 PUSH B

00AC E4 92 CLR A

00AD F5F0 93 MOV B,A

00AF E5F0 94 IO0010: MOV A,B

00B1 05F0 95 INC B

00B3 93 96 MOVC A,@A+DPTR

00B4 6005 97 JZ IO0020

00B6 1200C0 98 CALL SENDCHAR

00B9 80F4 99 JMP IO0010

00BB D0F0 100 IO0020: POP B

00BD D0E0 101 POP ACC

102

00BF 22 103 RET

104

105 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

106 ; SENDCHAR

107

00C0 108 SENDCHAR: ; sends ASCII value contained in A to UART

109

00C0 3099FD 110 JNB TI,$ ; wait til present char gone

00C3 C299 111 CLR TI ; must clear TI

00C5 F599 112 MOV SBUF,A

113

00C7 22 114 RET

115

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

UART2 PAGE 3

117 ; SENDVAL

118

00C8 119 SENDVAL: ; converts the hex value of A into two ASCII chars,

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

121 ; does not change the value of A.

122

00C8 C0E0 123 PUSH ACC

00CA C4 124 SWAP A

00CB 1200DC 125 CALL HEX2ASCII

00CE 11C0 126 CALL SENDCHAR ; send high nibble

00D0 D0E0 127 POP ACC

00D2 C0E0 128 PUSH ACC

00D4 1200DC 129 CALL HEX2ASCII

00D7 11C0 130 CALL SENDCHAR ; send low nibble

00D9 D0E0 131 POP ACC

132

00DB 22 133 RET

134

135

136 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

137 ; HEX2ASCII

138

00DC 139 HEX2ASCII: ; converts A into the hex character representing the

140 ; value of A's least significant nibble

141

00DC 540F 142 ANL A,#00Fh

00DE B40A00 143 CJNE A,#00Ah,$+3

00E1 4002 144 JC IO0030

00E3 2407 145 ADD A,#007h

00E5 2430 146 IO0030: ADD A,#'0'

147

00E7 22 148 RET

149

150 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

151 ; DELAY

00E8 152 DELAY: ; Delays by 100ms \* A

153 ; 100mSec based on 1.573MHZ Core Clock

154

155

00E8 FA 156 MOV R2,A ; Acc holds delay variable

00E9 7B32 157 DLY0: MOV R3,#50 ; Set up delay loop0

00EB 7C83 158 DLY1: MOV R4,#131 ; Set up delay loop1

00ED DCFE 159 DJNZ R4,$ ; Dec R4 & Jump here until R4 is 0

160 ; wait here for 131\*15.3us=2ms

00EF DBFA 161 DJNZ R3,DLY1 ; Dec R3 & Jump DLY1 until R3 is 0

162 ; Wait for 50\*2ms

00F1 DAF6 163 DJNZ R2,DLY0 ; Dec R2 & Jump DLY0 until R2 is 0

164 ; wait for ACC\*100ms

00F3 22 165 RET ; Return from subroutine

166

167 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

168

169

00F4 0A0A0D5F 170 TITLE: DB 10,10,13,'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_',10,13

00F8 5F5F5F5F

00FC 5F5F5F5F

0100 5F5F5F5F

0104 5F5F5F5F

UART2 PAGE 4

0108 5F5F5F5F

010C 5F5F5F5F

0110 5F5F5F5F

0114 5F5F5F5F

0118 5F5F5F0A

011C 0D

011D 416E616C 171 DB 'Analog Devices MicroConverter ADuC816',10,13

0121 6F672044

0125 65766963

0129 6573204D

012D 6963726F

0131 436F6E76

0135 65727465

0139 72204144

013D 75433831

0141 360A0D

0144 20202020 172 DB ' UART Demo Routine',10,13

0148 20202020

014C 20554152

0150 54204465

0154 6D6F2052

0158 6F757469

015C 6E650A0D

0160 20205472 173 DB ' Transmission of Data from 0 to 7F',10,13,0

0164 616E736D

0168 69737369

016C 6F6E206F

0170 66204461

0174 74612066

0178 726F6D20

017C 3020746F

0180 2037460A

0184 0D00

174

175 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

UART2 PAGE 5

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

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

CONT . . . . . . . . . . . . . . C ADDR 009AH

DELAY. . . . . . . . . . . . . . C ADDR 00E8H

DLY0 . . . . . . . . . . . . . . C ADDR 00E9H

DLY1 . . . . . . . . . . . . . . C ADDR 00EBH

HEX2ASCII. . . . . . . . . . . . C ADDR 00DCH

IO0010 . . . . . . . . . . . . . C ADDR 00AFH

IO0020 . . . . . . . . . . . . . C ADDR 00BBH

IO0030 . . . . . . . . . . . . . C ADDR 00E5H

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

LOOP1. . . . . . . . . . . . . . C ADDR 007EH

LOOP2. . . . . . . . . . . . . . C ADDR 008AH

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

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

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

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

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 00C0H

SENDSTRING . . . . . . . . . . . C ADDR 00A8H

SENDVAL. . . . . . . . . . . . . C ADDR 00C8H

START. . . . . . . . . . . . . . C ADDR 0072H

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

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

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

TITLE. . . . . . . . . . . . . . C ADDR 00F4H

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

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