UART2 PAGE 1

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

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : October 2003

6 ;

7 ; File : UART.asm

8 ;

9 ; Hardware : ADuC842/ADuC843

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\_Beta84x\9600com1.ht)

15 ; After the transmission of the 16 bytes a 2 second

16 ; delay is called and the process is repeated.

17 ;

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

19 ;

20 $MOD842 ;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 75D703 34 MAIN: MOV PLLCON,#03H ; core freq of 2.097152MHz

35 ; Main program

36 ;Configure Uart for 9600 baud at core freq of 2.097152MHz

0063 759E83 37 MOV T3CON,#83h

0066 759D2D 38 MOV T3FD,#02Dh

0069 759852 39 MOV SCON,#52h

40

006C 41 START:

006C B2B4 42 CPL LED ; CPL LED with each transmission

006E 9000EE 43 MOV DPTR, #TITLE

0071 1200A2 44 CALL SENDSTRING ; write title block on screen

45

0074 7800 46 MOV R0, #00H ; Start transmissions from 0

0076 7908 47 MOV R1, #08H ; Start a new line after 8 transmissions

48

0078 49 LOOP1: ; Every eight transmissions start on a

50 ; new line

0078 740A 51 MOV A, #10 ; Transmit a linefeed

007A 1200BA 52 CALL SENDCHAR

007D 740D 53 MOV A, #13 ; Transmit a carriage return

007F 1200BA 54 CALL SENDCHAR

55

0082 7908 56 MOV R1, #08H

57

0084 58 LOOP2:

UART2 PAGE 2

0084 7420 59 MOV A, #20H ; Transmit a SPACE (=ASCII 20) between

60 ; transmissions on same line

0086 1200BA 61 CALL SENDCHAR

62

0089 E8 63 MOV A, R0 ; Transmit R0 = present data

008A 1200C2 64 CALL SENDVAL

008D 08 65 INC R0 ; increment data

66

008E B47F03 67 CJNE A, #7FH, CONT ; check if data =7F, if no continue

0091 02009B 68 JMP WAIT5S ; if = 7F wait 5s and repeat

69

0094 19 70 CONT: DEC R1 ; decrement R1....

0095 E9 71 MOV A, R1

0096 B400EB 72 CJNE A, #00H, LOOP2 ; and check if new line is required

73 ; jump to loop 2 for a space

0099 80DD 74 JMP LOOP1 ; jump to loop 1 for a new line

75

76

009B 74C8 77 WAIT5S: MOV A, #200 ; wait 2s

009D 1200E2 78 CALL DELAY

00A0 80CA 79 JMP START ; start transmissions again

80

81

82 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

83 ; SENDSTRING

84

00A2 85 SENDSTRING: ; sends ASCII string to UART starting at location

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

87

00A2 C0E0 88 PUSH ACC

00A4 C0F0 89 PUSH B

00A6 E4 90 CLR A

00A7 F5F0 91 MOV B,A

00A9 E5F0 92 IO0010: MOV A,B

00AB 05F0 93 INC B

00AD 93 94 MOVC A,@A+DPTR

00AE 6005 95 JZ IO0020

00B0 1200BA 96 CALL SENDCHAR

00B3 80F4 97 JMP IO0010

00B5 D0F0 98 IO0020: POP B

00B7 D0E0 99 POP ACC

100

00B9 22 101 RET

102

103 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

104 ; SENDCHAR

105

00BA 106 SENDCHAR: ; sends ASCII value contained in A to UART

107

00BA 3099FD 108 JNB TI,$ ; wait til present char gone

00BD C299 109 CLR TI ; must clear TI

00BF F599 110 MOV SBUF,A

111

00C1 22 112 RET

113

114 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

115 ; SENDVAL

116

UART2 PAGE 3

00C2 117 SENDVAL: ; converts the hex value of A into two ASCII chars,

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

119 ; does not change the value of A.

120

00C2 C0E0 121 PUSH ACC

00C4 C4 122 SWAP A

00C5 1200D6 123 CALL HEX2ASCII

00C8 11BA 124 CALL SENDCHAR ; send high nibble

00CA D0E0 125 POP ACC

00CC C0E0 126 PUSH ACC

00CE 1200D6 127 CALL HEX2ASCII

00D1 11BA 128 CALL SENDCHAR ; send low nibble

00D3 D0E0 129 POP ACC

130

00D5 22 131 RET

132

133

134 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

135 ; HEX2ASCII

136

00D6 137 HEX2ASCII: ; converts A into the hex character representing the

138 ; value of A's least significant nibble

139

00D6 540F 140 ANL A,#00Fh

00D8 B40A00 141 CJNE A,#00Ah,$+3

00DB 4002 142 JC IO0030

00DD 2407 143 ADD A,#007h

00DF 2430 144 IO0030: ADD A,#'0'

145

00E1 22 146 RET

147

148 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

149

00E2 150 DELAY: ; Delays by 100ms \* A

151 ; 100mSec based on 2.097152MHZ

152 ; Core Clock

153 ;

154

00E2 FD 155 MOV R5,A ; Acc holds delay variable (1 clock)

00E3 7E1B 156 DLY0: MOV R6,#01Bh ; Set up delay loop0 (2 clocks)

00E5 7FFF 157 DLY1: MOV R7,#0FFh ; Set up delay loop1 (2 clocks)

00E7 DFFE 158 DJNZ R7,$ ; Dec R7 & Jump here until R7 is 0 (3 clocks)

00E9 DEFA 159 DJNZ R6,DLY1 ; Dec R6 & Jump DLY1 until R6 is 0 (3 clocks)

00EB DDF6 160 DJNZ R5,DLY0 ; Dec R5 & Jump DLY0 until R5 is 0 (3 clocks)

00ED 22 161 RET ; Return from subroutine

; DELAY

162 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

163

164

00EE 0A0A0D5F 165 TITLE: DB 10,10,13,'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_',10,13

00F2 5F5F5F5F

00F6 5F5F5F5F

00FA 5F5F5F5F

00FE 5F5F5F5F

0102 5F5F5F5F

0106 5F5F5F5F

010A 5F5F5F5F

UART2 PAGE 4

010E 5F5F5F5F

0112 5F5F5F0A

0116 0D

0117 416E616C 166 DB 'Analog Devices MicroConverter ADuC842',10,13

011B 6F672044

011F 65766963

0123 6573204D

0127 6963726F

012B 436F6E76

012F 65727465

0133 72204144

0137 75433834

013B 320A0D

013E 20202020 167 DB ' UART Demo Routine',10,13

0142 20202020

0146 20554152

014A 54204465

014E 6D6F2052

0152 6F757469

0156 6E650A0D

015A 20205472 168 DB ' Transmission of Data from 0 to 7F',10,13,0

015E 616E736D

0162 69737369

0166 6F6E206F

016A 66204461

016E 74612066

0172 726F6D20

0176 3020746F

017A 2037460A

017E 0D00

169

170 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 0094H

DELAY. . . . . . . . . . . . . . C ADDR 00E2H

DLY0 . . . . . . . . . . . . . . C ADDR 00E3H

DLY1 . . . . . . . . . . . . . . C ADDR 00E5H

HEX2ASCII. . . . . . . . . . . . C ADDR 00D6H

IO0010 . . . . . . . . . . . . . C ADDR 00A9H

IO0020 . . . . . . . . . . . . . C ADDR 00B5H

IO0030 . . . . . . . . . . . . . C ADDR 00DFH

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

LOOP1. . . . . . . . . . . . . . C ADDR 0078H

LOOP2. . . . . . . . . . . . . . C ADDR 0084H

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

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

PLLCON . . . . . . . . . . . . . D ADDR 00D7H PREDEFINED

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 00BAH

SENDSTRING . . . . . . . . . . . C ADDR 00A2H

SENDVAL. . . . . . . . . . . . . C ADDR 00C2H

START. . . . . . . . . . . . . . C ADDR 006CH

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

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

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

TITLE. . . . . . . . . . . . . . C ADDR 00EEH

WAIT5S . . . . . . . . . . . . . C ADDR 009BH