UART3 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 a number (starting at 0)

12 ; down the UART every second. Pressing the INT0

13 ; button increases the number being outputted.

14 ; Pressing a key on the keyboard causes the ASCII

15 ; char to be transmitted. Eg Pressing the 'A' button

16 ; causes the number '41h' to appear on the

17 ; hyperterminal program.

18 ;

19 ;====================================================================

20 ;

21 $MOD842 ;Use 8052 predefined Symbols

22

00B4 23 LED EQU P3.4

24

25 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

26 ; DEFINE VARIABLES IN INTERNAL RAM

---- 27 DSEG

0060 28 ORG 0060h

0060 29 INPUT: DS 1 ; data byte received by SPI

0061 30 OUTPUT: DS 1 ; data byte to send by SPI

31

32 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

33 ; BEGINNING OF CODE

---- 34 CSEG

0000 35 ORG 0000H

36

0000 020060 37 JMP MAIN

38 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

39 ; INTERRUPT VECTOR SPACE

0003 40 ORG 0003h ; (.................... INT0 ISR)

41

0003 0561 42 INC OUTPUT

0005 32 43 RETI

44 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

45

0060 46 ORG 0060H ; Start code at address above interrupts

47

48

0060 49 MAIN: ; Main program

0060 75D703 50 MOV PLLCON,#03H

0063 759E83 51 MOV T3CON,#83h

0066 759D2D 52 MOV T3FD,#02Dh

0069 759852 53 MOV SCON,#52h

54

55

56 ; CONFIGURE INTERRUPT 0...

57

006C D288 58 SETB IT0 ; INT0 edge triggered

UART3 PAGE 2

006E D2A8 59 SETB EX0 ; enable INT0 interrupt

60

61 ; ENABLE INTERRUPTS & ENTER MAIN LOOP...

62

0070 756100 63 MOV OUTPUT,#0 ; set initial value for output byte

0073 D2AF 64 SETB EA ; enable inturrupts

65

0075 66 TXDATA:

0075 B2B4 67 CPL LED ; CPL LED with each transmission

68

0077 E561 69 MOV A, OUTPUT ; output value

0079 1200B1 70 CALL SENDVAL

007C 9000DD 71 MOV DPTR,#SEPERATOR ; send line-feed & crdg-return..

007F 120091 72 CALL SENDSTRING ; ..out the UART

73

0082 7464 74 MOV A, #100

0084 1200D1 75 CALL DELAY ; delay for 1s

76

0087 3098EB 77 JNB RI, TXDATA ; check if data is received from keyboard

78

008A 859961 79 MOV OUTPUT, SBUF

008D C298 80 CLR RI

008F 80E4 81 JMP TXDATA

82

83

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

85 ; SENDSTRING

86

0091 87 SENDSTRING: ; sends ASCII string to UART starting at location

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

89

0091 C0E0 90 PUSH ACC

0093 C0F0 91 PUSH B

0095 E4 92 CLR A

0096 F5F0 93 MOV B,A

0098 E5F0 94 IO0010: MOV A,B

009A 05F0 95 INC B

009C 93 96 MOVC A,@A+DPTR

009D 6005 97 JZ IO0020

009F 1200A9 98 CALL SENDCHAR

00A2 80F4 99 JMP IO0010

00A4 D0F0 100 IO0020: POP B

00A6 D0E0 101 POP ACC

102

00A8 22 103 RET

104

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

106 ; SENDCHAR

107

00A9 108 SENDCHAR: ; sends ASCII value contained in A to UART

109

00A9 3099FD 110 JNB TI,$ ; wait til present char gone

00AC C299 111 CLR TI ; must clear TI

00AE F599 112 MOV SBUF,A

113

00B0 22 114 RET

115

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

UART3 PAGE 3

117 ; SENDVAL

118

00B1 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

00B1 C0E0 123 PUSH ACC

00B3 C4 124 SWAP A

00B4 1200C5 125 CALL HEX2ASCII

00B7 11A9 126 CALL SENDCHAR ; send high nibble

00B9 D0E0 127 POP ACC

00BB C0E0 128 PUSH ACC

00BD 1200C5 129 CALL HEX2ASCII

00C0 11A9 130 CALL SENDCHAR ; send low nibble

00C2 D0E0 131 POP ACC

132

00C4 22 133 RET

134

135

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

137 ; HEX2ASCII

138

00C5 139 HEX2ASCII: ; converts A into the hex character representing the

140 ; value of A's least significant nibble

141

00C5 540F 142 ANL A,#00Fh

00C7 B40A00 143 CJNE A,#00Ah,$+3

00CA 4002 144 JC IO0030

00CC 2407 145 ADD A,#007h

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

147

00D0 22 148 RET

149

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

151 ; DELAY

00D1 152 DELAY: ; Delays by 100ms \* A

153 ; 100mSec based on 2.097152MHZ

154 ; Core Clock

155

00D1 FD 156 MOV R5,A ; Acc holds delay variable (1 clock)

00D2 7E1B 157 DLY0: MOV R6,#01Bh ; Set up delay loop0 (2 clocks)

00D4 7FFF 158 DLY1: MOV R7,#0FFh ; Set up delay loop1 (2 clocks)

00D6 DFFE 159 DJNZ R7,$ ; Dec R7 & Jump here until R7 is 0 (3 clocks)

00D8 DEFA 160 DJNZ R6,DLY1 ; Dec R6 & Jump DLY1 until R6 is 0 (3 clocks)

00DA DDF6 161 DJNZ R5,DLY0 ; Dec R5 & Jump DLY0 until R5 is 0 (3 clocks)

00DC 22 162 RET ; Return from subroutine

163 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

164

00DD 0A0D00 165 SEPERATOR: DB 10,13,0

166

167

168 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

UART3 PAGE 4

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

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

DELAY. . . . . . . . . . . . . . C ADDR 00D1H

DLY0 . . . . . . . . . . . . . . C ADDR 00D2H

DLY1 . . . . . . . . . . . . . . C ADDR 00D4H

EA . . . . . . . . . . . . . . . B ADDR 00AFH PREDEFINED

EX0. . . . . . . . . . . . . . . B ADDR 00A8H PREDEFINED

HEX2ASCII. . . . . . . . . . . . C ADDR 00C5H

INPUT. . . . . . . . . . . . . . D ADDR 0060H NOT USED

IO0010 . . . . . . . . . . . . . C ADDR 0098H

IO0020 . . . . . . . . . . . . . C ADDR 00A4H

IO0030 . . . . . . . . . . . . . C ADDR 00CEH

IT0. . . . . . . . . . . . . . . B ADDR 0088H PREDEFINED

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

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

OUTPUT . . . . . . . . . . . . . D ADDR 0061H

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

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

RI . . . . . . . . . . . . . . . B ADDR 0098H PREDEFINED

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 00A9H

SENDSTRING . . . . . . . . . . . C ADDR 0091H

SENDVAL. . . . . . . . . . . . . C ADDR 00B1H

SEPERATOR. . . . . . . . . . . . C ADDR 00DDH

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

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

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

TXDATA . . . . . . . . . . . . . C ADDR 0075H