UART3 PAGE 1

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

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : May 2002

6 ;

7 ; File : UART.asm

8 ;

9 ; Hardware : ADuC831

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 $MOD831 ;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

50

0060 759E85 51 MOV T3CON,#85h

0063 759D08 52 MOV T3FD,#08h

0066 759852 53 MOV SCON,#52h

54

55

56 ; CONFIGURE INTERRUPT 0...

57

0069 D288 58 SETB IT0 ; INT0 edge triggered

UART3 PAGE 2

006B D2A8 59 SETB EX0 ; enable INT0 interrupt

60

61 ; ENABLE INTERRUPTS & ENTER MAIN LOOP...

62

006D 756100 63 MOV OUTPUT,#0 ; set initial value for output byte

0070 D2AF 64 SETB EA ; enable inturrupts

65

0072 66 TXDATA:

0072 B2B4 67 CPL LED ; CPL LED with each transmission

68

0074 E561 69 MOV A, OUTPUT ; output value

0076 1200AE 70 CALL SENDVAL

0079 9000DA 71 MOV DPTR,#SEPERATOR ; send line-feed & crdg-return..

007C 12008E 72 CALL SENDSTRING ; ..out the UART

73

007F 740A 74 MOV A, #0Ah

0081 1200CE 75 CALL DELAY ; delay for 1s

76

0084 3098EB 77 JNB RI, TXDATA ; check if data is received from keyboard

78

0087 859961 79 MOV OUTPUT, SBUF

008A C298 80 CLR RI

008C 80E4 81 JMP TXDATA

82

83

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

85 ; SENDSTRING

86

008E 87 SENDSTRING: ; sends ASCII string to UART starting at location

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

89

008E C0E0 90 PUSH ACC

0090 C0F0 91 PUSH B

0092 E4 92 CLR A

0093 F5F0 93 MOV B,A

0095 E5F0 94 IO0010: MOV A,B

0097 05F0 95 INC B

0099 93 96 MOVC A,@A+DPTR

009A 6005 97 JZ IO0020

009C 1200A6 98 CALL SENDCHAR

009F 80F4 99 JMP IO0010

00A1 D0F0 100 IO0020: POP B

00A3 D0E0 101 POP ACC

102

00A5 22 103 RET

104

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

106 ; SENDCHAR

107

00A6 108 SENDCHAR: ; sends ASCII value contained in A to UART

109

00A6 3099FD 110 JNB TI,$ ; wait til present char gone

00A9 C299 111 CLR TI ; must clear TI

00AB F599 112 MOV SBUF,A

113

00AD 22 114 RET

115

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

UART3 PAGE 3

117 ; SENDVAL

118

00AE 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

00AE C0E0 123 PUSH ACC

00B0 C4 124 SWAP A

00B1 1200C2 125 CALL HEX2ASCII

00B4 11A6 126 CALL SENDCHAR ; send high nibble

00B6 D0E0 127 POP ACC

00B8 C0E0 128 PUSH ACC

00BA 1200C2 129 CALL HEX2ASCII

00BD 11A6 130 CALL SENDCHAR ; send low nibble

00BF D0E0 131 POP ACC

132

00C1 22 133 RET

134

135

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

137 ; HEX2ASCII

138

00C2 139 HEX2ASCII: ; converts A into the hex character representing the

140 ; value of A's least significant nibble

141

00C2 540F 142 ANL A,#00Fh

00C4 B40A00 143 CJNE A,#00Ah,$+3

00C7 4002 144 JC IO0030

00C9 2407 145 ADD A,#007h

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

147

00CD 22 148 RET

149

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

151 ; DELAY

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

153 ; 100mSec based on 11.0592MHZ

154 ; Core Clock

155

00CE FD 156 MOV R5,A ; Acc holds delay variable

00CF 7EC8 157 DLY0: MOV R6,#200 ; Set up delay loop0

00D1 7FE5 158 DLY1: MOV R7,#229 ; Set up delay loop1

00D3 DFFE 159 DJNZ R7,$ ; Dec R2 until R2 is zero

00D5 DEFA 160 DJNZ R6,DLY1 ; Dec R1 & Jump DLY1 until R1 is 0

00D7 DDF6 161 DJNZ R5,DLY0 ; Dec R0 & Jump DLY0 until R0 is 0

00D9 22 162 RET ; Return from subroutine

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

164

00DA 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 00CEH

DLY0 . . . . . . . . . . . . . . C ADDR 00CFH

DLY1 . . . . . . . . . . . . . . C ADDR 00D1H

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

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

HEX2ASCII. . . . . . . . . . . . C ADDR 00C2H

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

IO0010 . . . . . . . . . . . . . C ADDR 0095H

IO0020 . . . . . . . . . . . . . C ADDR 00A1H

IO0030 . . . . . . . . . . . . . C ADDR 00CBH

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

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

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

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

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

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

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 00A6H

SENDSTRING . . . . . . . . . . . C ADDR 008EH

SENDVAL. . . . . . . . . . . . . C ADDR 00AEH

SEPERATOR. . . . . . . . . . . . C ADDR 00DAH

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

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

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

TXDATA . . . . . . . . . . . . . C ADDR 0072H