UART3 PAGE 1

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

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : 11/17/99

6 ;

7 ; File : UART.asm

8 ;

9 ; Hardware : ADuC824

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 $MOD824 ;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 75CBFF 51 MOV RCAP2H,#0FFh ; config UART for 9830baud

0063 75CAFB 52 MOV RCAP2L,#-5 ; (close enough to 9600baud)

0066 75CDFF 53 MOV TH2,#0FFh

0069 75CCFB 54 MOV TL2,#-5

006C 759852 55 MOV SCON,#52h

006F 75C834 56 MOV T2CON,#34h

57

58

UART3 PAGE 2

59 ; CONFIGURE INTERRUPT 0...

60

0072 D288 61 SETB IT0 ; INT0 edge triggered

0074 D2A8 62 SETB EX0 ; enable INT0 interrupt

63

64 ; ENABLE INTERRUPTS & ENTER MAIN LOOP...

65

0076 756100 66 MOV OUTPUT,#0 ; set initial value for output byte

0079 D2AF 67 SETB EA ; enable inturrupts

68

007B 69 TXDATA:

007B B2B4 70 CPL LED ; CPL LED with each transmission

71

007D E561 72 MOV A, OUTPUT ; output value

007F 1200B7 73 CALL SENDVAL

0082 9000E3 74 MOV DPTR,#SEPERATOR ; send line-feed & crdg-return..

0085 120097 75 CALL SENDSTRING ; ..out the UART

76

0088 740A 77 MOV A, #0Ah

008A 1200D7 78 CALL DELAY ; delay for 1s

79

008D 3098EB 80 JNB RI, TXDATA ; check if data is received from keyboard

81

0090 859961 82 MOV OUTPUT, SBUF

0093 C298 83 CLR RI

0095 80E4 84 JMP TXDATA

85

86

87 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

88 ; SENDSTRING

89

0097 90 SENDSTRING: ; sends ASCII string to UART starting at location

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

92

0097 C0E0 93 PUSH ACC

0099 C0F0 94 PUSH B

009B E4 95 CLR A

009C F5F0 96 MOV B,A

009E E5F0 97 IO0010: MOV A,B

00A0 05F0 98 INC B

00A2 93 99 MOVC A,@A+DPTR

00A3 6005 100 JZ IO0020

00A5 1200AF 101 CALL SENDCHAR

00A8 80F4 102 JMP IO0010

00AA D0F0 103 IO0020: POP B

00AC D0E0 104 POP ACC

105

00AE 22 106 RET

107

108 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

109 ; SENDCHAR

110

00AF 111 SENDCHAR: ; sends ASCII value contained in A to UART

112

00AF 3099FD 113 JNB TI,$ ; wait til present char gone

00B2 C299 114 CLR TI ; must clear TI

00B4 F599 115 MOV SBUF,A

116

UART3 PAGE 3

00B6 22 117 RET

118

119 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

120 ; SENDVAL

121

00B7 122 SENDVAL: ; converts the hex value of A into two ASCII chars,

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

124 ; does not change the value of A.

125

00B7 C0E0 126 PUSH ACC

00B9 C4 127 SWAP A

00BA 1200CB 128 CALL HEX2ASCII

00BD 11AF 129 CALL SENDCHAR ; send high nibble

00BF D0E0 130 POP ACC

00C1 C0E0 131 PUSH ACC

00C3 1200CB 132 CALL HEX2ASCII

00C6 11AF 133 CALL SENDCHAR ; send low nibble

00C8 D0E0 134 POP ACC

135

00CA 22 136 RET

137

138

139 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

140 ; HEX2ASCII

141

00CB 142 HEX2ASCII: ; converts A into the hex character representing the

143 ; value of A's least significant nibble

144

00CB 540F 145 ANL A,#00Fh

00CD B40A00 146 CJNE A,#00Ah,$+3

00D0 4002 147 JC IO0030

00D2 2407 148 ADD A,#007h

00D4 2430 149 IO0030: ADD A,#'0'

150

00D6 22 151 RET

152

153 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

154 ; DELAY

00D7 155 DELAY: ; Delays by 100ms \* A

156 ; 100mSec based on 1.573MHZ Core Clock

157

158

00D7 FA 159 MOV R2,A ; Acc holds delay variable

00D8 7B32 160 DLY0: MOV R3,#50 ; Set up delay loop0

00DA 7C83 161 DLY1: MOV R4,#131 ; Set up delay loop1

00DC DCFE 162 DJNZ R4,$ ; Dec R4 & Jump here until R4 is 0

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

00DE DBFA 164 DJNZ R3,DLY1 ; Dec R3 & Jump DLY1 until R3 is 0

165 ; Wait for 50\*2ms

00E0 DAF6 166 DJNZ R2,DLY0 ; Dec R2 & Jump DLY0 until R2 is 0

167 ; wait for ACC\*100ms

00E2 22 168 RET ; Return from subroutine

169

170 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

171

00E3 0A0D00 172 SEPERATOR: DB 10,13,0

173

174

UART3 PAGE 4

175 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

UART3 PAGE 5

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

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

DELAY. . . . . . . . . . . . . . C ADDR 00D7H

DLY0 . . . . . . . . . . . . . . C ADDR 00D8H

DLY1 . . . . . . . . . . . . . . C ADDR 00DAH

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

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

HEX2ASCII. . . . . . . . . . . . C ADDR 00CBH

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

IO0010 . . . . . . . . . . . . . C ADDR 009EH

IO0020 . . . . . . . . . . . . . C ADDR 00AAH

IO0030 . . . . . . . . . . . . . C ADDR 00D4H

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

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

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

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

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

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

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

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

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

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

SENDCHAR . . . . . . . . . . . . C ADDR 00AFH

SENDSTRING . . . . . . . . . . . C ADDR 0097H

SENDVAL. . . . . . . . . . . . . C ADDR 00B7H

SEPERATOR. . . . . . . . . . . . C ADDR 00E3H

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

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

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

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

TXDATA . . . . . . . . . . . . . C ADDR 007BH