UART824 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 the numbers 0->7F (starting

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

13 ; they can be viewed using Hyperterminal.

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

15 ; delay is called and the process is repeated.

16 ;

17 ;====================================================================

18 ;

19 $MOD824 ;Use 8052 predefined Symbols

20

00B4 21 LED EQU P3.4

22

23 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

24 ; BEGINNING OF CODE

---- 25 CSEG

0000 26 ORG 0000H

27

0000 020060 28 JMP MAIN

29

0060 30 ORG 0060H ; Start code at address above interrupts

31

32

0060 33 MAIN: ; Main program

34

0060 75CBFF 35 MOV RCAP2H,#0FFh ; config UART for 9830baud

0063 75CAFB 36 MOV RCAP2L,#-5 ; (close enough to 9600baud)

0066 75CDFF 37 MOV TH2,#0FFh

0069 75CCFB 38 MOV TL2,#-5

006C 759852 39 MOV SCON,#52h

006F 75C834 40 MOV T2CON,#34h

41

0072 42 START:

0072 B2B4 43 CPL LED ; CPL LED with each transmission

0074 9000F4 44 MOV DPTR, #TITLE

0077 1200A8 45 CALL SENDSTRING ; write title block on screen

46

007A 7800 47 MOV R0, #00H ; Start transmissions from 0

007C 7908 48 MOV R1, #08H ; Start a new line after 8 transmissions

49

007E 50 LOOP1: ; Every eight transmissions start on a

51 ; new line

007E 740A 52 MOV A, #10 ; Transmit a linefeed

0080 1200C0 53 CALL SENDCHAR

0083 740D 54 MOV A, #13 ; Transmit a carriage return

0085 1200C0 55 CALL SENDCHAR

56

0088 7908 57 MOV R1, #08H

58

UART824 PAGE 2

008A 59 LOOP2:

008A 7420 60 MOV A, #20H ; Transmit a SPACE (=ASCII 20) between

61 ; transmissions on same line

008C 1200C0 62 CALL SENDCHAR

63

008F E8 64 MOV A, R0 ; Transmit R0 = present data

0090 1200C8 65 CALL SENDVAL

0093 08 66 INC R0 ; increment data

67

0094 B47F03 68 CJNE A, #7FH, CONT ; check if data =7F, if no continue

0097 0200A1 69 JMP WAIT5S ; if = 7F wait 5s and repeat

70

009A 19 71 CONT: DEC R1 ; decrement R1....

009B E9 72 MOV A, R1

009C B400EB 73 CJNE A, #00H, LOOP2 ; and check if new line is required

74 ; jump to loop 2 for a space

009F 80DD 75 JMP LOOP1 ; jump to loop 1 for a new line

76

77

00A1 7432 78 WAIT5S: MOV A, #50 ; wait 5s

00A3 1200E8 79 CALL DELAY

00A6 80CA 80 JMP START ; start transmissions again

81

82

83 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

84 ; SENDSTRING

85

00A8 86 SENDSTRING: ; sends ASCII string to UART starting at location

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

88

00A8 C0E0 89 PUSH ACC

00AA C0F0 90 PUSH B

00AC E4 91 CLR A

00AD F5F0 92 MOV B,A

00AF E5F0 93 IO0010: MOV A,B

00B1 05F0 94 INC B

00B3 93 95 MOVC A,@A+DPTR

00B4 6005 96 JZ IO0020

00B6 1200C0 97 CALL SENDCHAR

00B9 80F4 98 JMP IO0010

00BB D0F0 99 IO0020: POP B

00BD D0E0 100 POP ACC

101

00BF 22 102 RET

103

104 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

105 ; SENDCHAR

106

00C0 107 SENDCHAR: ; sends ASCII value contained in A to UART

108

00C0 3099FD 109 JNB TI,$ ; wait til present char gone

00C3 C299 110 CLR TI ; must clear TI

00C5 F599 111 MOV SBUF,A

112

00C7 22 113 RET

114

115 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

116 ; SENDVAL

UART824 PAGE 3

117

00C8 118 SENDVAL: ; converts the hex value of A into two ASCII chars,

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

120 ; does not change the value of A.

121

00C8 C0E0 122 PUSH ACC

00CA C4 123 SWAP A

00CB 1200DC 124 CALL HEX2ASCII

00CE 11C0 125 CALL SENDCHAR ; send high nibble

00D0 D0E0 126 POP ACC

00D2 C0E0 127 PUSH ACC

00D4 1200DC 128 CALL HEX2ASCII

00D7 11C0 129 CALL SENDCHAR ; send low nibble

00D9 D0E0 130 POP ACC

131

00DB 22 132 RET

133

134

135 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

136 ; HEX2ASCII

137

00DC 138 HEX2ASCII: ; converts A into the hex character representing the

139 ; value of A's least significant nibble

140

00DC 540F 141 ANL A,#00Fh

00DE B40A00 142 CJNE A,#00Ah,$+3

00E1 4002 143 JC IO0030

00E3 2407 144 ADD A,#007h

00E5 2430 145 IO0030: ADD A,#'0'

146

00E7 22 147 RET

148

149 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

150 ; DELAY

00E8 151 DELAY: ; Delays by 100ms \* A

152 ; 100mSec based on 1.573MHZ Core Clock

153

154

00E8 FA 155 MOV R2,A ; Acc holds delay variable

00E9 7B32 156 DLY0: MOV R3,#50 ; Set up delay loop0

00EB 7C83 157 DLY1: MOV R4,#131 ; Set up delay loop1

00ED DCFE 158 DJNZ R4,$ ; Dec R4 & Jump here until R4 is 0

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

00EF DBFA 160 DJNZ R3,DLY1 ; Dec R3 & Jump DLY1 until R3 is 0

161 ; Wait for 50\*2ms

00F1 DAF6 162 DJNZ R2,DLY0 ; Dec R2 & Jump DLY0 until R2 is 0

163 ; wait for ACC\*100ms

00F3 22 164 RET ; Return from subroutine

165

166 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

167

168

00F4 0A0A0D5F 169 TITLE: DB 10,10,13,'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_',10,13

00F8 5F5F5F5F

00FC 5F5F5F5F

0100 5F5F5F5F

0104 5F5F5F5F

0108 5F5F5F5F

UART824 PAGE 4

010C 5F5F5F5F

0110 5F5F5F5F

0114 5F5F5F5F

0118 5F5F5F0A

011C 0D

011D 416E616C 170 DB 'Analog Devices MicroConverter ADuC824',10,13

0121 6F672044

0125 65766963

0129 6573204D

012D 6963726F

0131 436F6E76

0135 65727465

0139 72204144

013D 75433832

0141 340A0D

0144 20202020 171 DB ' UART Demo Routine',10,13

0148 20202020

014C 20554152

0150 54204465

0154 6D6F2052

0158 6F757469

015C 6E650A0D

0160 20205472 172 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

173

174 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

UART824 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