TEMPUART PAGE 1

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

2 ;

3 ; Author : ADI - Apps

4 ;

5 ; Date : August 2000

6 ;

7 ; File : TempUart.asm

8 ;

9 ; Hardware : ADuC824

10 ;

11 ; Description : This Program takes a temperature measurement every

12 ; second from the on-chip temp sensor and sends the

13 ; temp in degrees Celcius up the UART to the PC where

14 ; it can be read using hyperterminal

15 ;====================================================================

16 ;

17 $MOD824 ; Use 8052 / ADuC824 predefined Symbols

18

00B4 19 LED EQU P3.4

0000 20 FLAG EQU 00h

21

22 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

23 ; DEFINE VARIABLES IN INTERNAL RAM

---- 24 DSEG

25

0033 26 ORG 0033h

0033 27 COUNT1: DS 1

0034 28 COUNT2: DS 1

0035 29 COUNT3: DS 1

0036 30 DIG1: DS 1

0037 31 DIG2: DS 1

0038 32 DIG3: DS 1

33 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

34 ; BEGINNING OF CODE

---- 35 CSEG

0000 36 ORG 0000H

37

0000 020060 38 JMP MAIN

39

40 ;====================================================================

41

0060 42 ORG 0060H ; Start code at address above interrupts

43

44

45

0060 46 MAIN: ; Main program

47

0060 75CBFF 48 MOV RCAP2H,#0FFh ; config UART for 9830baud

0063 75CAFB 49 MOV RCAP2L,#-5 ; (close enough to 9600baud)

0066 75CDFF 50 MOV TH2,#0FFh

0069 75CCFB 51 MOV TL2,#-5

006C 759852 52 MOV SCON,#52h

006F 75C834 53 MOV T2CON,#34h

54

55 ; Configure ADC

0072 75D110 56 MOV ADCMODE, #10H ; ENABLE AUX Mode - Power down

0075 75D320 57 MOV ADC1CON, #20H ; USE INTERNAL REFERENCE

58 ; PTAT(+) --> PTAT(-)

TEMPUART PAGE 2

59 ; BIPOLAR MODE

60 ; Fixed +/- 2.5V range

61

0078 90010A 62 MOV DPTR, #TITLE

007B 1200E6 63 CALL SENDSTRING ; write title block on screen

64

65 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

66 ; TEMP MEASURE LOOP

007E 67 TEMPLOOP:

007E 75D112 68 MOV ADCMODE, #12H ; INITIATE A SINGLE AUX CONV

0081 30DEFD 69 JNB RDY1,$ ; Wait for conversion results

70

71 ; conversion result ready

72 ; a value of 80h in AD1H=0degC

73

0084 E5DD 74 MOV A, ADC1H ; 80h=0, FFh=+127, 00h=-128

0086 C3 75 CLR C

0087 9480 76 SUBB A, #80H ; convert to 2's comp

77 ; FFh=-1, 80h=-128, 00h=0, 7Fh=+127

78

79

0089 80 SENDDECs: ; SENDs the signed decimal number in Acc up UART

81 ; -128->127

0089 C0F0 82 PUSH B

008B C0E0 83 PUSH ACC

008D 30E70B 84 JNB ACC.7, HUNDREDS

0090 742D 85 MOV A, #'-' ; transmit minus sign

0092 1200DE 86 CALL SENDCHAR

0095 D0E0 87 POP ACC ; restore original value of A

0097 C0E0 88 PUSH ACC ; remember original value of A

0099 F4 89 CPL A

009A 04 90 INC A

91

009B 92 HUNDREDS: ; check #hundreds

009B 75F064 93 MOV B, #100 ; divide remainder by 100

009E 84 94 DIV AB ; A receives integer quotient

95 ; B receives the remainder

009F D2D5 96 SETB F0

00A1 6007 97 JZ TENS ; if ACC=0 then num=0xx

00A3 C2D5 98 CLR F0

00A5 2430 99 ADD A, #'0'

00A7 1200DE 100 LCALL SENDCHAR

101

00AA 102 TENS: ; check tens

00AA E5F0 103 MOV A,B

00AC 75F00A 104 MOV B,#10

00AF 84 105 DIV AB ; divide remainder by 10

00B0 30D502 106 JNB F0, SEND0 ; if F0 is cleared the a number

107 ; exists in the 100s

00B3 6005 108 JZ UNITS

109

00B5 2430 110 SEND0: ADD A, #'0' ; only send a zero if number

00B7 1200DE 111 CALL SENDCHAR ; existed in the 100s

112

00BA E5F0 113 UNITS: MOV A,B ; send remainder (even if 0)

00BC 2430 114 ADD A, #'0'

00BE 1200DE 115 CALL SENDCHAR

00C1 D0E0 116 POP ACC

TEMPUART PAGE 3

00C3 D0F0 117 POP B

118

00C5 9000FD 119 MOV DPTR, #DEGREES

00C8 1200E6 120 CALL SENDSTRING

00CB 7401 121 MOV A, #01

00CD 1200D2 122 CALL DELAY

123

00D0 80AC 124 JMP TEMPLOOP

125

126 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

127 ; DELAY

128

00D2 129 DELAY: ; Delays by 100ms \* A

130 ; 100mSec based on 1.573MHZ Core Clock

131

00D2 FA 132 MOV R2,A ; Acc holds delay variable

00D3 7B32 133 DLY0: MOV R3,#50 ; Set up delay loop0

00D5 7C83 134 DLY1: MOV R4,#131 ; Set up delay loop1

00D7 DCFE 135 DJNZ R4,$ ; Dec R4 & Jump here until R4 is 0

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

00D9 DBFA 137 DJNZ R3,DLY1 ; Dec R3 & Jump DLY1 until R3 is 0

138 ; Wait for 50\*2ms

00DB DAF6 139 DJNZ R2,DLY0 ; Dec R2 & Jump DLY0 until R2 is 0

140 ; wait for ACC\*100ms

00DD 22 141 RET ; Return from subroutine

142

143 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

144 ; SENDCHAR

145

00DE 146 SENDCHAR: ; sends ASCII value contained in A to UART

147

00DE 3099FD 148 JNB TI,$ ; wait til present char gone

00E1 C299 149 CLR TI ; must clear TI

00E3 F599 150 MOV SBUF,A

151

00E5 22 152 RET

153

154 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

155 ; SENDSTRING

156

00E6 157 SENDSTRING: ; sends ASCII string to UART starting at location

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

159

00E6 C0E0 160 PUSH ACC

00E8 C0F0 161 PUSH B

00EA E4 162 CLR A

00EB F5F0 163 MOV B,A

00ED E5F0 164 IO0010: MOV A,B

00EF 05F0 165 INC B

00F1 93 166 MOVC A,@A+DPTR

00F2 6004 167 JZ IO0020

00F4 11DE 168 CALL SENDCHAR

00F6 80F5 169 JMP IO0010

00F8 D0F0 170 IO0020: POP B

00FA D0E0 171 POP ACC

172

00FC 22 173 RET

174

TEMPUART PAGE 4

175 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

176

00FD 20646567 177 DEGREES: DB ' degrees C',10,13,0

0101 72656573

0105 20430A0D

0109 00

178

179

010A 0A0A0D5F 180 TITLE: DB 10,10,13,'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_',10,13

010E 5F5F5F5F

0112 5F5F5F5F

0116 5F5F5F5F

011A 5F5F5F5F

011E 5F5F5F5F

0122 5F5F5F5F

0126 5F5F5F5F

012A 5F5F5F5F

012E 5F5F5F0A

0132 0D

0133 416E616C 181 DB 'Analog Devices MicroConverter ADuC824',10,13

0137 6F672044

013B 65766963

013F 6573204D

0143 6963726F

0147 436F6E76

014B 65727465

014F 72204144

0153 75433832

0157 340A0D

015A 20202020 182 DB ' Temp Sensor Demo Routine',10,13,0

015E 20205465

0162 6D702053

0166 656E736F

016A 72204465

016E 6D6F2052

0172 6F757469

0176 6E650A0D

017A 00

183

184 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

TEMPUART PAGE 5

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

ADC1CON. . . . . . . . . . . . . D ADDR 00D3H PREDEFINED

ADC1H. . . . . . . . . . . . . . D ADDR 00DDH PREDEFINED

ADCMODE. . . . . . . . . . . . . D ADDR 00D1H PREDEFINED

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

COUNT1 . . . . . . . . . . . . . D ADDR 0033H NOT USED

COUNT2 . . . . . . . . . . . . . D ADDR 0034H NOT USED

COUNT3 . . . . . . . . . . . . . D ADDR 0035H NOT USED

DEGREES. . . . . . . . . . . . . C ADDR 00FDH

DELAY. . . . . . . . . . . . . . C ADDR 00D2H

DIG1 . . . . . . . . . . . . . . D ADDR 0036H NOT USED

DIG2 . . . . . . . . . . . . . . D ADDR 0037H NOT USED

DIG3 . . . . . . . . . . . . . . D ADDR 0038H NOT USED

DLY0 . . . . . . . . . . . . . . C ADDR 00D3H

DLY1 . . . . . . . . . . . . . . C ADDR 00D5H

F0 . . . . . . . . . . . . . . . B ADDR 00D5H PREDEFINED

FLAG . . . . . . . . . . . . . . NUMB 0000H NOT USED

HUNDREDS . . . . . . . . . . . . C ADDR 009BH

IO0010 . . . . . . . . . . . . . C ADDR 00EDH

IO0020 . . . . . . . . . . . . . C ADDR 00F8H

LED. . . . . . . . . . . . . . . NUMB 00B4H NOT USED

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

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

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

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

RDY1 . . . . . . . . . . . . . . B ADDR 00DEH PREDEFINED

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

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

SEND0. . . . . . . . . . . . . . C ADDR 00B5H

SENDCHAR . . . . . . . . . . . . C ADDR 00DEH

SENDDECS . . . . . . . . . . . . C ADDR 0089H NOT USED

SENDSTRING . . . . . . . . . . . C ADDR 00E6H

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

TEMPLOOP . . . . . . . . . . . . C ADDR 007EH

TENS . . . . . . . . . . . . . . C ADDR 00AAH

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

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

TITLE. . . . . . . . . . . . . . C ADDR 010AH

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

UNITS. . . . . . . . . . . . . . C ADDR 00BAH