I2CSLAVE PAGE 1

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

2 ;

3 ; Author : ADI - Apps www.analog.com/MicroConverter

4 ;

5 ; Date : May 2002

6 ;

7 ; File : i2cslave.asm

8 ;

9 ; Hardware : ADuC814

10 ;

11 ; Description : Code for a slave in an I2C system. This code will

12 ; continuously receive and transmit a byte over the I2C

13 ; interface, then send the received byte out the UART,

14 ; then check if a character had been entered in the UART.

15 ; If so, it will send the ASCII value of the character

16 ; entered to the slave, the next time it transmits a byte.

17 ;

18 ; Reference : Tech Note, uC001: "MicroConverter I2C Compatible

19 ; Interface" find it at www.analog.com/microconverter

20 ;

21 ;======================================================================

22

23 $MOD814

24

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

26 ; DEFINE VARIABLES IN INTERNAL RAM

27

0030 28 BYTECNT DATA 30h ; byte counter for I2C routines

0031 29 INPUT DATA 31h ; data recieved from master

0032 30 OUTPUT DATA 32h ; data to be transmitted to master

31

0000 32 GO BIT 00h ; flag to wait for interrupts

0001 33 FIRST BIT 01h ; flag to indicate first receive Int

34

00B4 35 LED EQU P3.4 ; P3.4 drives the LED on eval board

36

37 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

38 ; BEGINNING OF CODE

---- 39 CSEG

0000 40 ORG 0000h

0000 020060 41 JMP MAIN

42 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

43 ; INT0 ISR

0003 44 ORG 0003h

0003 0532 45 INC OUTPUT

0005 32 46 RETI

47 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

48 ; I2C ISR

003B 49 ORG 003Bh

50

003B 20E90B 51 JB I2CTX, SLAVE\_TRANSMITTER

52

003E 53 SLAVE\_RECEIVER:

003E 200110 54 JB FIRST, ENDINT1 ; if first INT then wait for next int

0041 D200 55 SETB GO ; reception complete

0043 859A31 56 MOV INPUT, I2CDAT ; store data received in INPUT

0046 020051 57 JMP ENDINT1

58

I2CSLAVE PAGE 2

0049 59 SLAVE\_TRANSMITTER:

0049 D200 60 SETB GO ; transmission complete

004B 85329A 61 MOV I2CDAT, OUTPUT ; move data to be transmitted into I2CDAT

004E 020053 62 JMP ENDINT2 ; Note: On the ADuC824/816 the read or

63 ; write of I2CDAT register

64 ; automatically clears i2ci. If

65 ; I2CI is cleared twice then the

66 ; microconverter will hang.)

67

0051 68 ENDINT1:

0051 C2E8 69 CLR I2CI ; clear I2C interrupt bit (812 only)

0053 70 ENDINT2:

0053 C201 71 CLR FIRST ; address has already been received

0055 32 72 RETI

73

74 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

75 ; MAIN PROGRAM

0060 76 ORG 0060h

0060 77 MAIN:

78

79

80 ; configure the UART ADuC812s

0060 75CBFF 81 MOV RCAP2H,#0FFh ; config UART for 9830baud

0063 75CAF9 82 MOV RCAP2L,#-7 ; (close enough to 9600baud)

0066 75CDFF 83 MOV TH2,#0FFh

0069 75CCF9 84 MOV TL2,#-7

006C 759852 85 MOV SCON,#52h

006F 75C834 86 MOV T2CON,#34h

87

88 ; configure pins for 812s

0072 759C01 89 MOV CFG814,#01H

90

91 ;configure and enable interrupts

92 ; MOV IE2,#01h ; enable I2C interrupt

0075 75A901 93 MOV IEIP2,#01h ; enable I2C interrupt

0078 D2A8 94 SETB EX0 ; enable INT0

007A D288 95 SETB IT0 ; INT0 edge triggered

007C D2AF 96 SETB EA ; allow all the interrupts

97

98 ;initialize settings

007E 759B44 99 MOV I2CADD,#044h ; slave address is 44h

0081 75E800 100 MOV I2CCON,#00h ; slave mode (default=>not necessary)

0084 C200 101 CLR GO ; clear flag to wait for interrupt

102 ; GO is set once data is TX'd or RX'd

0086 D201 103 SETB FIRST ; FIRST is cleared after receiving the

104 ; first SLAVE receiver interrupt

105

0088 753200 106 MOV OUTPUT,#0 ; first byte to be transmitted is 40h

008B C2B4 107 CLR LED

108

008D 109 WAITFORDATA:

008D 3000FD 110 JNB GO,$ ; ----- wait for i2c interrupt ------

111 ; If it is in receive mode, it will

112 ; wait here for a second interrupt (as

113 ; the first interrupt only contains the

114 ; slave address in I2CDAT).

115 ; In transmit mode the tranmission will

116 ; occur after the first interrupt.

I2CSLAVE PAGE 3

0090 D201 117 SETB FIRST ; re-initialise flags

0092 C200 118 CLR GO

0094 20E9F6 119 JB I2CTX,WAITFORDATA

120 ; if the slave has just transmitted then

121 ; wait to receive a byte

122 ; if the slave has just received then

123 ; send input up the UART

124

0097 125 SENDUART:

0097 B2B4 126 CPL LED ; LED changes each time one byte has been

127 ; received and another transmitted

128

0099 E531 129 MOV A,INPUT ; send value received out the UART

009B 1200C6 130 CALL SENDVAL

009E 740A 131 MOV A,#10

00A0 1200B2 132 CALL SENDCHAR ; send LF + CR

00A3 740D 133 MOV A,#13

00A5 1200B2 134 CALL SENDCHAR

135

00A8 3098E2 136 JNB RI, WAITFORDATA ; repeat (unless UART data received)

137

138 ; WHEN UART DATA RECEIVED, MOVE DATA TO I2C OUTPUT...

139

00AB 859932 140 MOV OUTPUT, SBUF ; update OUTPUT byte to new value

00AE C298 141 CLR RI ; must clear RI

00B0 80DB 142 JMP WAITFORDATA ; back to main loop

143

144

145 ;======================================================================

146 ; SUBROUTINES

147 ;======================================================================

148

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

150 ; SENDCHAR

151 ; sends ASCII value contained in A to UART

152

00B2 153 SENDCHAR:

00B2 3099FD 154 JNB TI,$ ; wait 'til present char gone

00B5 C299 155 CLR TI ; must clear TI

00B7 F599 156 MOV SBUF,A

00B9 22 157 RET

158 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

159 ; HEX2ASCII

160 ; converts A into the hex character representing the value of A's

161 ; least significant nibble

162

00BA 163 HEX2ASCII:

00BA 540F 164 ANL A,#00Fh

00BC B40A00 165 CJNE A,#00Ah,$+3

00BF 4002 166 JC IO0030

00C1 2407 167 ADD A,#007h

00C3 2430 168 IO0030: ADD A,#'0'

00C5 22 169 RET

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

171 ; SENDVAL

172 ; converts the hex value of A into two ASCII chars, and then spits

173 ; these two characters up the UART. does not change the value of A.

174

I2CSLAVE PAGE 4

00C6 175 SENDVAL:

00C6 C0E0 176 PUSH ACC

00C8 C4 177 SWAP A

00C9 11BA 178 CALL HEX2ASCII

00CB 11B2 179 CALL SENDCHAR ; send high nibble

00CD D0E0 180 POP ACC

00CF C0E0 181 PUSH ACC

00D1 11BA 182 CALL HEX2ASCII

00D3 11B2 183 CALL SENDCHAR ; send low nibble

00D5 D0E0 184 POP ACC

00D7 22 185 RET

186 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

187

188 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

I2CSLAVE PAGE 5

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

BYTECNT. . . . . . . . . . . . . D ADDR 0030H NOT USED

CFG814 . . . . . . . . . . . . . D ADDR 009CH PREDEFINED

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

ENDINT1. . . . . . . . . . . . . C ADDR 0051H

ENDINT2. . . . . . . . . . . . . C ADDR 0053H

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

FIRST. . . . . . . . . . . . . . B ADDR 0001H

GO . . . . . . . . . . . . . . . B ADDR 0000H

HEX2ASCII. . . . . . . . . . . . C ADDR 00BAH

I2CADD . . . . . . . . . . . . . D ADDR 009BH PREDEFINED

I2CCON . . . . . . . . . . . . . D ADDR 00E8H PREDEFINED

I2CDAT . . . . . . . . . . . . . D ADDR 009AH PREDEFINED

I2CI . . . . . . . . . . . . . . B ADDR 00E8H PREDEFINED

I2CTX. . . . . . . . . . . . . . B ADDR 00E9H PREDEFINED

IEIP2. . . . . . . . . . . . . . D ADDR 00A9H PREDEFINED

INPUT. . . . . . . . . . . . . . D ADDR 0031H

IO0030 . . . . . . . . . . . . . C ADDR 00C3H

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

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

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

OUTPUT . . . . . . . . . . . . . D ADDR 0032H

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 00B2H

SENDUART . . . . . . . . . . . . C ADDR 0097H NOT USED

SENDVAL. . . . . . . . . . . . . C ADDR 00C6H

SLAVE\_RECEIVER . . . . . . . . . C ADDR 003EH NOT USED

SLAVE\_TRANSMITTER. . . . . . . . C ADDR 0049H

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

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

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

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

WAITFORDATA. . . . . . . . . . . C ADDR 008DH