I2CSLAVE PAGE 1

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

2 ;

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

4 ;

5 ; Date : Oct 2000

6 ;

7 ; File : i2cslave.asm

8 ;

9 ; Hardware : ADuC836 (commented out = ADuC812)

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 $MOD836

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 ; configure the UART ADuC812

80 ; MOV SCON,#52h ; configure UART for 9600baud..

81 ; MOV TMOD,#20h ; ..assuming 11.0592MHz crystal

82 ; MOV TH1,#-3

83 ; SETB TR1

84

85 ; configure the UART ADuC824/ADuC816

86 ; MOV RCAP2H,#0FFh ; config UART for 9830baud

87 ; MOV RCAP2L,#-5 ; (close enough to 9600baud)

88 ; MOV TH2,#0FFh

89 ; MOV TL2,#-5

90 ; MOV SCON,#52h

91 ; MOV T2CON,#34h

92

93 ; configure UART for 9600 using Timer3

0060 759E82 94 MOV T3CON,#82h

0063 759D12 95 MOV T3FD,#12h

0066 759852 96 MOV SCON,#52h

97

98 ;configure and enable interrupts

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

0069 75A901 100 MOV IEIP2,#01h ; enable I2C interrupt

006C D2A8 101 SETB EX0 ; enable INT0

006E D288 102 SETB IT0 ; INT0 edge triggered

0070 D2AF 103 SETB EA ; allow all the interrupts

104

105 ;initialize settings

0072 759B44 106 MOV I2CADD,#044h ; slave address is 44h

0075 75E800 107 MOV I2CCON,#00h ; slave mode (default=>not necessary)

0078 C200 108 CLR GO ; clear flag to wait for interrupt

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

007A D201 110 SETB FIRST ; FIRST is cleared after receiving the

111 ; first SLAVE receiver interrupt

112

007C 753200 113 MOV OUTPUT,#0 ; first byte to be transmitted is 40h

007F C2B4 114 CLR LED

115

0081 116 WAITFORDATA:

I2CSLAVE PAGE 3

0081 3000FD 117 JNB GO,$ ; ----- wait for i2c interrupt ------

118 ; If it is in receive mode, it will

119 ; wait here for a second interrupt (as

120 ; the first interrupt only contains the

121 ; slave address in I2CDAT).

122 ; In transmit mode the tranmission will

123 ; occur after the first interrupt.

0084 D201 124 SETB FIRST ; re-initialise flags

0086 C200 125 CLR GO

0088 20E9F6 126 JB I2CTX,WAITFORDATA

127 ; if the slave has just transmitted then

128 ; wait to receive a byte

129 ; if the slave has just received then

130 ; send input up the UART

131

008B 132 SENDUART:

008B B2B4 133 CPL LED ; LED changes each time one byte has been

134 ; received and another transmitted

135

008D E531 136 MOV A,INPUT ; send value received out the UART

008F 1200BA 137 CALL SENDVAL

0092 740A 138 MOV A,#10

0094 1200A6 139 CALL SENDCHAR ; send LF + CR

0097 740D 140 MOV A,#13

0099 1200A6 141 CALL SENDCHAR

142

009C 3098E2 143 JNB RI, WAITFORDATA ; repeat (unless UART data received)

144

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

146

009F 859932 147 MOV OUTPUT, SBUF ; update OUTPUT byte to new value

00A2 C298 148 CLR RI ; must clear RI

00A4 80DB 149 JMP WAITFORDATA ; back to main loop

150

151

152 ;======================================================================

153 ; SUBROUTINES

154 ;======================================================================

155

156 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

157 ; SENDCHAR

158 ; sends ASCII value contained in A to UART

159

00A6 160 SENDCHAR:

00A6 3099FD 161 JNB TI,$ ; wait 'til present char gone

00A9 C299 162 CLR TI ; must clear TI

00AB F599 163 MOV SBUF,A

00AD 22 164 RET

165 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

166 ; HEX2ASCII

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

168 ; least significant nibble

169

00AE 170 HEX2ASCII:

00AE 540F 171 ANL A,#00Fh

00B0 B40A00 172 CJNE A,#00Ah,$+3

00B3 4002 173 JC IO0030

00B5 2407 174 ADD A,#007h

I2CSLAVE PAGE 4

00B7 2430 175 IO0030: ADD A,#'0'

00B9 22 176 RET

177 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

178 ; SENDVAL

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

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

181

00BA 182 SENDVAL:

00BA C0E0 183 PUSH ACC

00BC C4 184 SWAP A

00BD 11AE 185 CALL HEX2ASCII

00BF 11A6 186 CALL SENDCHAR ; send high nibble

00C1 D0E0 187 POP ACC

00C3 C0E0 188 PUSH ACC

00C5 11AE 189 CALL HEX2ASCII

00C7 11A6 190 CALL SENDCHAR ; send low nibble

00C9 D0E0 191 POP ACC

00CB 22 192 RET

193 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

194

195 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

I2CSLAVE PAGE 5

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

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

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 00AEH

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 00B7H

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

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

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

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

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

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

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

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

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

SENDUART . . . . . . . . . . . . C ADDR 008BH NOT USED

SENDVAL. . . . . . . . . . . . . C ADDR 00BAH

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

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

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

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

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

WAITFORDATA. . . . . . . . . . . C ADDR 0081H