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 : ADuC816/ADuC824 (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 ;$MOD812

23 ;$MOD816

24 ;$MOD824

25 $MOD834

26

27 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

28 ; DEFINE VARIABLES IN INTERNAL RAM

29

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

0031 31 INPUT DATA 31h ; data recieved from master

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

33

0000 34 GO BIT 00h ; flag to wait for interrupts

0001 35 FIRST BIT 01h ; flag to indicate first receive Int

36

00B4 37 LED EQU P3.4 ; P3.4 drives the LED on eval board

38

39 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

40 ; BEGINNING OF CODE

---- 41 CSEG

0000 42 ORG 0000h

0000 020060 43 JMP MAIN

44 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

45 ; INT0 ISR

0003 46 ORG 0003h

0003 0532 47 INC OUTPUT

0005 32 48 RETI

49 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

50 ; I2C ISR

003B 51 ORG 003Bh

52

003B 20E90B 53 JB I2CTX, SLAVE\_TRANSMITTER

54

003E 55 SLAVE\_RECEIVER:

003E 200110 56 JB FIRST, ENDINT1 ; if first INT then wait for next int

0041 D200 57 SETB GO ; reception complete

0043 859A31 58 MOV INPUT, I2CDAT ; store data received in INPUT

I2CSLAVE PAGE 2

0046 020051 59 JMP ENDINT1

60

0049 61 SLAVE\_TRANSMITTER:

0049 D200 62 SETB GO ; transmission complete

004B 85329A 63 MOV I2CDAT, OUTPUT ; move data to be transmitted into I2CDAT

004E 020053 64 JMP ENDINT2 ; Note: On the ADuC824/816 the read or

65 ; write of I2CDAT register

66 ; automatically clears i2ci. If

67 ; I2CI is cleared twice then the

68 ; microconverter will hang.)

69

0051 70 ENDINT1:

0051 C2E8 71 CLR I2CI ; clear I2C interrupt bit (812 only)

0053 72 ENDINT2:

0053 C201 73 CLR FIRST ; address has already been received

0055 32 74 RETI

75

76 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

77 ; MAIN PROGRAM

0060 78 ORG 0060h

0060 79 MAIN:

80

81 ; configure the UART ADuC812

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

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

84 ; MOV TH1,#-3

85 ; SETB TR1

86

87 ; configure the UART ADuC824/ADuC816

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

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

90 ; MOV TH2,#0FFh

91 ; MOV TL2,#-5

92 ; MOV SCON,#52h

93 ; MOV T2CON,#34h

94

95 ; configure UART for 9600 using Timer3

0060 759E82 96 MOV T3CON,#82h

0063 759D12 97 MOV T3FD,#12h

0066 759852 98 MOV SCON,#52h

99

100 ;configure and enable interrupts

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

0069 75A901 102 MOV IEIP2,#01h ; enable I2C interrupt

006C D2A8 103 SETB EX0 ; enable INT0

006E D288 104 SETB IT0 ; INT0 edge triggered

0070 D2AF 105 SETB EA ; allow all the interrupts

106

107 ;initialize settings

0072 759B44 108 MOV I2CADD,#044h ; slave address is 44h

0075 75E800 109 MOV I2CCON,#00h ; slave mode (default=>not necessary)

0078 C200 110 CLR GO ; clear flag to wait for interrupt

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

007A D201 112 SETB FIRST ; FIRST is cleared after receiving the

113 ; first SLAVE receiver interrupt

114

007C 753200 115 MOV OUTPUT,#0 ; first byte to be transmitted is 40h

007F C2B4 116 CLR LED

I2CSLAVE PAGE 3

117

0081 118 WAITFORDATA:

0081 3000FD 119 JNB GO,$ ; ----- wait for i2c interrupt ------

120 ; If it is in receive mode, it will

121 ; wait here for a second interrupt (as

122 ; the first interrupt only contains the

123 ; slave address in I2CDAT).

124 ; In transmit mode the tranmission will

125 ; occur after the first interrupt.

0084 D201 126 SETB FIRST ; re-initialise flags

0086 C200 127 CLR GO

0088 20E9F6 128 JB I2CTX,WAITFORDATA

129 ; if the slave has just transmitted then

130 ; wait to receive a byte

131 ; if the slave has just received then

132 ; send input up the UART

133

008B 134 SENDUART:

008B B2B4 135 CPL LED ; LED changes each time one byte has been

136 ; received and another transmitted

137

008D E531 138 MOV A,INPUT ; send value received out the UART

008F 1200BA 139 CALL SENDVAL

0092 740A 140 MOV A,#10

0094 1200A6 141 CALL SENDCHAR ; send LF + CR

0097 740D 142 MOV A,#13

0099 1200A6 143 CALL SENDCHAR

144

009C 3098E2 145 JNB RI, WAITFORDATA ; repeat (unless UART data received)

146

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

148

009F 859932 149 MOV OUTPUT, SBUF ; update OUTPUT byte to new value

00A2 C298 150 CLR RI ; must clear RI

00A4 80DB 151 JMP WAITFORDATA ; back to main loop

152

153

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

155 ; SUBROUTINES

156 ;======================================================================

157

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

159 ; SENDCHAR

160 ; sends ASCII value contained in A to UART

161

00A6 162 SENDCHAR:

00A6 3099FD 163 JNB TI,$ ; wait 'til present char gone

00A9 C299 164 CLR TI ; must clear TI

00AB F599 165 MOV SBUF,A

00AD 22 166 RET

167 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

168 ; HEX2ASCII

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

170 ; least significant nibble

171

00AE 172 HEX2ASCII:

00AE 540F 173 ANL A,#00Fh

00B0 B40A00 174 CJNE A,#00Ah,$+3

I2CSLAVE PAGE 4

00B3 4002 175 JC IO0030

00B5 2407 176 ADD A,#007h

00B7 2430 177 IO0030: ADD A,#'0'

00B9 22 178 RET

179 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

180 ; SENDVAL

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

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

183

00BA 184 SENDVAL:

00BA C0E0 185 PUSH ACC

00BC C4 186 SWAP A

00BD 11AE 187 CALL HEX2ASCII

00BF 11A6 188 CALL SENDCHAR ; send high nibble

00C1 D0E0 189 POP ACC

00C3 C0E0 190 PUSH ACC

00C5 11AE 191 CALL HEX2ASCII

00C7 11A6 192 CALL SENDCHAR ; send low nibble

00C9 D0E0 193 POP ACC

00CB 22 194 RET

195 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

196

197 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