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

23 ;$MOD812 ; use ADuC812 & 8052 predefined symbols

24 ;$MOD816

25 $MOD824

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

0060 75CBFF 88 MOV RCAP2H,#0FFh ; config UART for 9830baud

0063 75CAFB 89 MOV RCAP2L,#-5 ; (close enough to 9600baud)

0066 75CDFF 90 MOV TH2,#0FFh

0069 75CCFB 91 MOV TL2,#-5

006C 759852 92 MOV SCON,#52h

006F 75C834 93 MOV T2CON,#34h

94

95 ;configure and enable interrupts

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

0072 75A901 97 MOV IEIP2,#01h ; enable I2C interrupt

0075 D2A8 98 SETB EX0 ; enable INT0

0077 D288 99 SETB IT0 ; INT0 edge triggered

0079 D2AF 100 SETB EA ; allow all the interrupts

101

102 ;initialize settings

007B 759B44 103 MOV I2CADD,#044h ; slave address is 44h

007E 75E800 104 MOV I2CCON,#00h ; slave mode (default=>not necessary)

0081 C200 105 CLR GO ; clear flag to wait for interrupt

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

0083 D201 107 SETB FIRST ; FIRST is cleared after receiving the

108 ; first SLAVE receiver interrupt

109

0085 753200 110 MOV OUTPUT,#0 ; first byte to be transmitted is 40h

0088 C2B4 111 CLR LED

112

008A 113 WAITFORDATA:

008A 3000FD 114 JNB GO,$ ; ----- wait for i2c interrupt ------

115 ; If it is in receive mode, it will

116 ; wait here for a second interrupt (as

I2CSLAVE PAGE 3

117 ; the first interrupt only contains the

118 ; slave address in I2CDAT).

119 ; In transmit mode the tranmission will

120 ; occur after the first interrupt.

008D D201 121 SETB FIRST ; re-initialise flags

008F C200 122 CLR GO

0091 20E9F6 123 JB I2CTX,WAITFORDATA

124 ; if the slave has just transmitted then

125 ; wait to receive a byte

126 ; if the slave has just received then

127 ; send input up the UART

128

0094 129 SENDUART:

0094 B2B4 130 CPL LED ; LED changes each time one byte has been

131 ; received and another transmitted

132

0096 E531 133 MOV A,INPUT ; send value received out the UART

0098 1200C3 134 CALL SENDVAL

009B 740A 135 MOV A,#10

009D 1200AF 136 CALL SENDCHAR ; send LF + CR

00A0 740D 137 MOV A,#13

00A2 1200AF 138 CALL SENDCHAR

139

00A5 3098E2 140 JNB RI, WAITFORDATA ; repeat (unless UART data received)

141

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

143

00A8 859932 144 MOV OUTPUT, SBUF ; update OUTPUT byte to new value

00AB C298 145 CLR RI ; must clear RI

00AD 80DB 146 JMP WAITFORDATA ; back to main loop

147

148

149 ;======================================================================

150 ; SUBROUTINES

151 ;======================================================================

152

153 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

154 ; SENDCHAR

155 ; sends ASCII value contained in A to UART

156

00AF 157 SENDCHAR:

00AF 3099FD 158 JNB TI,$ ; wait 'til present char gone

00B2 C299 159 CLR TI ; must clear TI

00B4 F599 160 MOV SBUF,A

00B6 22 161 RET

162 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

163 ; HEX2ASCII

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

165 ; least significant nibble

166

00B7 167 HEX2ASCII:

00B7 540F 168 ANL A,#00Fh

00B9 B40A00 169 CJNE A,#00Ah,$+3

00BC 4002 170 JC IO0030

00BE 2407 171 ADD A,#007h

00C0 2430 172 IO0030: ADD A,#'0'

00C2 22 173 RET

174 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I2CSLAVE PAGE 4

175 ; SENDVAL

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

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

178

00C3 179 SENDVAL:

00C3 C0E0 180 PUSH ACC

00C5 C4 181 SWAP A

00C6 11B7 182 CALL HEX2ASCII

00C8 11AF 183 CALL SENDCHAR ; send high nibble

00CA D0E0 184 POP ACC

00CC C0E0 185 PUSH ACC

00CE 11B7 186 CALL HEX2ASCII

00D0 11AF 187 CALL SENDCHAR ; send low nibble

00D2 D0E0 188 POP ACC

00D4 22 189 RET

190 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

191

192 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 00B7H

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 00C0H

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 00AFH

SENDUART . . . . . . . . . . . . C ADDR 0094H NOT USED

SENDVAL. . . . . . . . . . . . . C ADDR 00C3H

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 008AH