MASTER PAGE 1

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

2 ;

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

4 ;

5 ; Date : 30 April 1999

6 ;

7 ; File : master.asm

8 ;

9 ; Hardware : ADuC812

10 ;

11 ; Description : Code for a master in an I2C system.

12 ;

13 ; Reference : Tech Note, uC001: "Using the ADuC812 I2C Interface"

14 ; find it at www.analog.com/microconverter

15 ;

16 ;======================================================================

17

18 $MOD812 ; use ADuC812 & 8052 predefined symbols

19

0008 20 BITCNT DATA 8h ; bit counter for I2C routines

0030 21 BYTECNT DATA 030h ; byte counter for I2C routines

0032 22 SLAVEADD DATA 032h ; slave address for I2C routines

23

0028 24 FLAGS DATA 28h

0040 25 NOACK BIT FLAGS.0 ; I2C no acknowledge flag

0041 26 BUSY BIT FLAGS.1 ; I2C busy flag

0042 27 ERROR BIT FLAGS.2 ; I2C error flag

00B4 28 MISTAKE BIT P3.4

29

30 ;======================================================================

31

---- 32 CSEG

33

0000 34 ORG 0000H

35

0000 020122 36 JMP START

37

38 ;======================================================================

39

007B 40 ORG 007BH ; Subroutines

41

42 ;----------------------------------------------------------------------

43 ; DELAY: Create a delay for the main signals ( SCLOCK , SDATA )

44 ;----------------------------------------------------------------------

45

007B 46 DELAY:

47

007B 00 48 NOP

007C 22 49 RET

50

51 ;----------------------------------------------------------------------

52 ; SENDSTOP: Send the bit stop of the transmission

53 ;----------------------------------------------------------------------

54

007D 55 SENDSTOP:

56

007D D2EE 57 SETB MDE ; to enable SDATA pin as an output

007F C2EF 58 CLR MDO ; get SDATA ready for stop

MASTER PAGE 2

0081 D2ED 59 SETB MCO ; set clock for stop

0083 117B 60 ACALL DELAY

0085 D2EF 61 SETB MDO ; this is the stop bit

0087 C241 62 CLR BUSY ; bus should be released

0089 22 63 RET

64

65 ;----------------------------------------------------------------------

66 ; SENDBYTE: Send a 8-bits word to the slave

67 ;----------------------------------------------------------------------

68

008A 69 SENDBYTE:

70

008A 750808 71 MOV BITCNT,#8 ; 8 bits in a byte

72

008D D2EE 73 SETB MDE ; to enable SDATA pin as an output

008F C2EF 74 CLR MDO

0091 C2ED 75 CLR MCO

0093 33 76 LOOP: RLC A ; send one bit

0094 92EF 77 MOV MDO,C ; put data bit on pin

0096 D2ED 78 SETB MCO ; send clock

0098 C2ED 79 CLR MCO ; clock is off

009A D508F6 80 DJNZ BITCNT,LOOP

81

009D C2EE 82 CLR MDE ; release data line for acknowledge

009F D2ED 83 SETB MCO ; send clock for acknowledge

00A1 30EC02 84 JNB MDI,NEXT ; this is a check

00A4 D240 85 SETB NOACK ; no acknowledge

00A6 C2ED 86 NEXT: CLR MCO ; clock for acknowledge

00A8 22 87 RET

88

89

90 ;----------------------------------------------------------------------

91 ; BITSTART: Send the bit start of the transmission and the slave

92 ; address to the slave

93 ;----------------------------------------------------------------------

94

00A9 95 BITSTART:

96

00A9 D241 97 SETB BUSY ; I2C is in progress

00AB D2EE 98 SETB MDE ; to enable SDATA pin as an output

00AD C240 99 CLR NOACK

00AF C242 100 CLR ERROR

00B1 30ED09 101 JNB MCO,FAULT

00B4 30EF06 102 JNB MDO,FAULT

00B7 C2EF 103 CLR MDO ; this is

00B9 117B 104 ACALL DELAY ; the

00BB C2ED 105 CLR MCO ; start bit

00BD C2B4 106 FAULT: CLR MISTAKE ; set error flag

00BF E532 107 MOV A,SLAVEADD ; Get slave address

00C1 118A 108 ACALL SENDBYTE ; call routine to send slave addr. byte

00C3 22 109 RET

110

111 ;----------------------------------------------------------------------

112 ; SENDATA: Send all the sequence to the slave ( slave address + data )

113 ;----------------------------------------------------------------------

114

00C4 115 SENDATA:

116

MASTER PAGE 3

00C4 11A9 117 ACALL BITSTART

00C6 20EC0C 118 JB MDI,NEXT1

00C9 7400 119 MOV A,#00

00CB E0 120 SLOOP: MOVX A,@DPTR

00CC 118A 121 ACALL SENDBYTE

00CE A3 122 INC DPTR

00CF 204003 123 JB NOACK,NEXT1

00D2 D530F6 124 DJNZ BYTECNT,SLOOP

125

00D5 117D 126 NEXT1: ACALL SENDSTOP

00D7 E528 127 MOV A,FLAGS

00D9 5407 128 ANL A,#07h

00DB 6004 129 JZ RETOUR

00DD C2B4 130 CLR P3.4

00DF C2EA 131 CLR I2CRS

00E1 22 132 RETOUR: RET

133

134 ;----------------------------------------------------------------------

135 ; RCVBYTE: receives one byte of data from an I2C slave device.

136 ;----------------------------------------------------------------------

137

00E2 138 RCVBYTE:

139

00E2 750808 140 MOV BITCNT,#8 ;Set bit count.

141

00E5 C2EE 142 CLR MDE ;Data pin of the master is now an input

00E7 C2ED 143 CLR MCO

00E9 D2ED 144 LOOP2: SETB MCO

00EB C2ED 145 CLR MCO

00ED A2EC 146 MOV C,MDI ;Get data bit from pin.

00EF 33 147 RLC A ;Rotate bit into result byte.

148

00F0 D508F6 149 DJNZ BITCNT,LOOP2 ;Repeat until all bits received.

150

151 ;result byte is in the accumulator

152

00F3 C0E0 153 PUSH ACC ;Save result byte in the stack

154

00F5 D2EE 155 SETB MDE ;Data pin of the master must be an..

156 ;..output for the acknowledge

00F7 E530 157 MOV A,BYTECNT

00F9 B40104 158 CJNE A,#1,SACK ;Check for last byte of frame.

00FC D2EF 159 SETB MDO ;Send no acknowledge on last byte.

00FE 8002 160 SJMP NACK

161

0100 C2EF 162 SACK: CLR MDO ;Send acknowledge bit.

163

0102 D2ED 164 NACK: SETB MCO ;Send acknowledge clock.

0104 D0E0 165 POP ACC ;Restore accumulator

0106 117B 166 ACALL DELAY

0108 C2ED 167 CLR MCO

010A D2EF 168 SETB MDO ;Clear acknowledge bit.

010C 117B 169 ACALL DELAY

010E C2EE 170 CLR MDE

171

0110 22 172 RET

173

174 ;----------------------------------------------------------------------

MASTER PAGE 4

175 ; RCVDATA: receives one or more bytes of data from an I2C slave device.

176 ;----------------------------------------------------------------------

177

0111 0532 178 RCVDATA: INC SLAVEADD ;Set for READ of slave.

0113 11A9 179 ACALL BITSTART ;Acquire bus and send slave address.

0115 204007 180 JB NoAck,RDEX ;Check for slave not responding.

181

0118 11E2 182 RDLoop: ACALL RCVBYTE ;Receive next data byte.

011A F7 183 MOV @R1,A ;Save data byte in buffer.

011B 09 184 INC R1 ;Advance buffer pointer.

011C D530F9 185 DJNZ BYTECNT,RDLoop ;Repeat untill all bytes received.

186

011F 117D 187 RDEX: ACALL SENDSTOP ;Done, send an I2C stop.

0121 22 188 RET

189

190 ;======================================================================

191 ; Main program

192 ;======================================================================

193

0122 194 START:

195

0122 758140 196 MOV SP,#040h

0125 C240 197 CLR NOACK

0127 753288 198 MOV SLAVEADD,#088H

012A 753003 199 MOV BYTECNT,#3

012D 75E8A8 200 MOV I2CCON,#0A8h

201

202 ; code for a write mode ( master-transmitter to slave-receiver )

203

204 ; MOV DPTR,#080H ; master transmits to slave

205 ; MOV A,#055H ; datas which are located in

206 ; MOVX @DPTR,A ; the external memory

207 ; MOV DPTR,#081H

208 ; MOV A,#044H

209 ; MOVX @DPTR,A

210 ; MOV DPTR,#082H

211 ; MOV A,#033H

212 ; MOVX @DPTR,A

213

214 ; MOV DPTR,#080h

215 ; ACALL SENDATA

216

217 ; code for a read mode ( master reads immediately after first byte )

218

0130 7935 219 MOV R1,#035h

0132 3111 220 ACALL RCVDATA

221

222 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

MASTER PAGE 5

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

BITCNT . . . . . . . . . . . . . D ADDR 0008H

BITSTART . . . . . . . . . . . . C ADDR 00A9H

BUSY . . . . . . . . . . . . . . B ADDR 0041H

BYTECNT. . . . . . . . . . . . . D ADDR 0030H

DELAY. . . . . . . . . . . . . . C ADDR 007BH

ERROR. . . . . . . . . . . . . . B ADDR 0042H

FAULT. . . . . . . . . . . . . . C ADDR 00BDH

FLAGS. . . . . . . . . . . . . . D ADDR 0028H

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

I2CRS. . . . . . . . . . . . . . B ADDR 00EAH PREDEFINED

LOOP . . . . . . . . . . . . . . C ADDR 0093H

LOOP2. . . . . . . . . . . . . . C ADDR 00E9H

MCO. . . . . . . . . . . . . . . B ADDR 00EDH PREDEFINED

MDE. . . . . . . . . . . . . . . B ADDR 00EEH PREDEFINED

MDI. . . . . . . . . . . . . . . B ADDR 00ECH PREDEFINED

MDO. . . . . . . . . . . . . . . B ADDR 00EFH PREDEFINED

MISTAKE. . . . . . . . . . . . . B ADDR 00B4H

NACK . . . . . . . . . . . . . . C ADDR 0102H

NEXT . . . . . . . . . . . . . . C ADDR 00A6H

NEXT1. . . . . . . . . . . . . . C ADDR 00D5H

NOACK. . . . . . . . . . . . . . B ADDR 0040H

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

RCVBYTE. . . . . . . . . . . . . C ADDR 00E2H

RCVDATA. . . . . . . . . . . . . C ADDR 0111H

RDEX . . . . . . . . . . . . . . C ADDR 011FH

RDLOOP . . . . . . . . . . . . . C ADDR 0118H

RETOUR . . . . . . . . . . . . . C ADDR 00E1H

SACK . . . . . . . . . . . . . . C ADDR 0100H

SENDATA. . . . . . . . . . . . . C ADDR 00C4H NOT USED

SENDBYTE . . . . . . . . . . . . C ADDR 008AH

SENDSTOP . . . . . . . . . . . . C ADDR 007DH

SLAVEADD . . . . . . . . . . . . D ADDR 0032H

SLOOP. . . . . . . . . . . . . . C ADDR 00CBH

SP . . . . . . . . . . . . . . . D ADDR 0081H PREDEFINED

START. . . . . . . . . . . . . . C ADDR 0122H