;======================================================================

;

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

;

; Date : 30 April 1999

;

; File : slave.asm

;

; Hardware : ADuC812

;

; Description : Code for a slave in an I2C system.

;

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

; find it at www.analog.com/microconverter

;

;======================================================================

$MOD812 ; use ADuC812 & 8052 predefined symbols

BYTECNT DATA 030h ; byte counter for I2C routines

FLAGS DATA 28h

GO BIT FLAGS.0 ; flag for all the interrupts

RC BIT FLAGS.1 ; flag for Write mode interrupt

TR BIT FLAGS.2 ; flag for Read mode interrupt

;======================================================================

CSEG

ORG 0000H

JMP START

;======================================================================

ORG 003Bh ; I2C slave interrupt

JB RC,RECEIVE ; depending on flags there

JB TR,TRANSMIT ; are two different interrupts

;======================================================================

ORG 007BH ; Subroutines

;----------------------------------------------------------------------

; RECEIVE: receive interrupt routine

;----------------------------------------------------------------------

RECEIVE:

SETB GO

MOV @R1,I2CDAT ; move data on internal RAM

CLR I2CI ; clear interrupt bit

RETI

;----------------------------------------------------------------------

; TRANSMIT: transmit interrupt routine

;----------------------------------------------------------------------

TRANSMIT:

SETB GO

MOV I2CDAT,R0

CLR I2CI ; clear interrupt bit

RETI

;----------------------------------------------------------------------

; RCVBYTE2: receive byte routine for read mode

;----------------------------------------------------------------------

RCVBYTE2:

NOP

RET

;----------------------------------------------------------------------

; RCVBYTE: receive byte routine

;----------------------------------------------------------------------

RCVBYTE:

JNB GO,$ ; wait for the interrupt

INC R1 ; next storage will be on 41h then 42h

CLR GO ; flag cleared for the next interrupt

RET

;----------------------------------------------------------------------

; RCVDATA: receive bytes routine

;----------------------------------------------------------------------

RCVDATA:

MOV BYTECNT,#4 ; 4 bytes : address + 3 datas

LOOP2: ACALL RCVBYTE

DJNZ BYTECNT,LOOP2

RET

;----------------------------------------------------------------------

; SENDBYTE: byte transmit routine

;----------------------------------------------------------------------

SENDBYTE:

JNB GO,$ ; wait for the interrupt

INC R0 ; 2nd data is 34h and 3rd data is 35h

CLR GO

RET

;----------------------------------------------------------------------

;SENDATA:bytes transmit routine

;----------------------------------------------------------------------

SENDATA:

MOV BYTECNT,#3 ; 3 data will be send by the slave

LOOP: ACALL SENDBYTE

DJNZ BYTECNT,LOOP

RET

;======================================================================

;Main program

;======================================================================

START:

CLR GO ; clear flag used in the interrupt

MOV I2CADD,#044h ; slave address

MOV SP,#020h

MOV IE,#80h ; enable all the interrupts

MOV IE2,#01h ; enable I2C interrupt

MOV I2CCON,#000h ; slave mode

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

; SETB RC ; specific flag for interrupt routine

; MOV R1,#040h ; first data to be stored in RAM at 40h

; ACALL RCVDATA ; slave receives his address + 3 datas

; code for read mode ( master reads slave immediately after 1st byte )

SETB RC ; specific flag for interrupt routine

MOV R0,#033h ; first data send is 33h

ACALL RCVBYTE2 ; slave receives address send by master

CLR RC

SETB TR

ACALL SENDATA ; slave sends 3 datas

SETB P3.4 ; led is off, everything is OK

END