

Interfacing Serial EEPROM Using AX1001 SPI Interface



AN-AX1000-002

May 2005

Version 1.0

1.0 Introduction

AX1001 has two SERDES (serializer/deserializer) units. Both units support SPI master/slave mode, UART, and I2C. For SERDES-1, RA0 (SCK), RA1 (SI), RA2 (SO) are used. For SERDES-2, RA3 (SCK), RA4 (SI), RA5 (SO) are used.

This application note describes the implementation of SPI interface using 32K x 8-bit SPI serial EEPROM (AT25256). AT25256 address range is from 0x0000

to 0x7FFF. In the example of this application note, the program is using SERDES-1 to perform EEPROM read and write operation (master mode).

Figure 1 shows the SPI connection between AX1001 and EEPROM using SERDES-1. SI is serial data input pin. SO is serial data output pin. SCK is serial clock pin. CS_ is chip select pin (active low). In this case, AX1001 is in master mode and EEPROM is in slave mode.

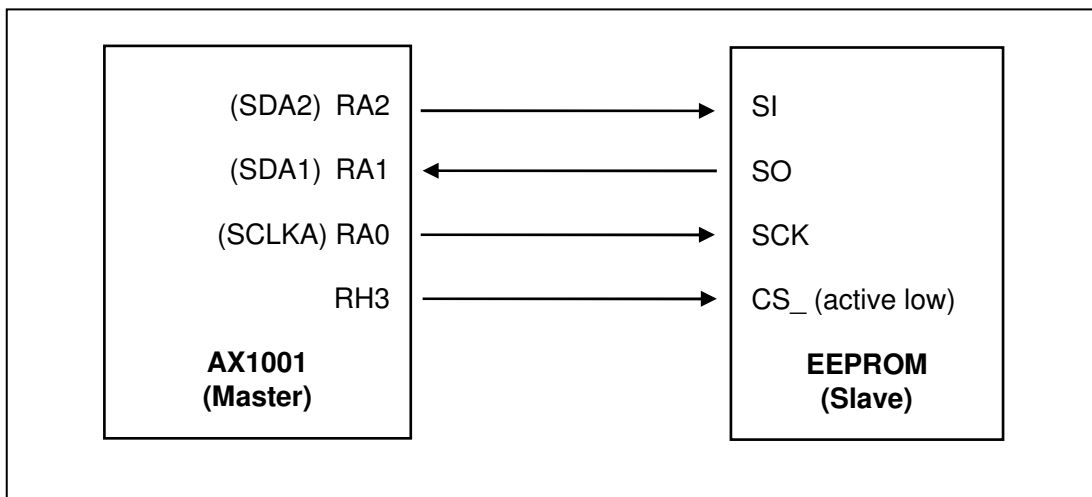


Figure 1 SPI interface connection between AX1001 and EEPROM

2.0 Programming Implementation

The following AX1001 data RAM addresses are used in the example:

- SPI_addrH and SPI_addrL – stores EEPROM reading and writing address.
- AFSRH_temp and AFSRL_temp – stores AX1001 data RAM reading and writing address.
- SPI_byte_cnt – counter that stores the number of bytes to be read from or written to EEPROM.
- Time_SPI – delay counter used in SPI_DELAY subroutine.
- SPI_status – stores value of SPISTATUS SFR.

The address of the 7 registers mentioned above can be defined by user. In the example, the registers are located in bank 0 of data RAM.

To control SERDES-1, the following SFR's are used: SERDES1_CTRL, SERDES_CNT, SERDES1_BRC, SERDES1_BUFH, SERDES1_BUFL. For more details regarding the SFR's, user may refer to the latest version of **AX1001 Development & Application User Guide**.

2.1 Reading data from EEPROM to AX1001

The example is trying to read 2 bytes of data at address 0x3005 - 0x3006, and stores to AX1001 data RAM address 0x010A - 0x010B. Here is the set up

procedure in the example:

- Set SPI_byte_cnt = 0x02. If the number of bytes is 256, set SPI_byte_cnt = 0x00.
- Set SPI_addrH = 0x30, SPI_addrL = 0x05.
- Set AFSRH_temp = 0x01, AFSRL_temp = 0x0A. The subroutine uses pointer as indirect addressing.
- Call SPI2DRAM_NBYTE subroutine.

2.2 Writing data to EEPROM from AX1001

The example is trying to write 2 bytes of data to EEPROM address 0x3005 - 0x3006, from AX1001 data RAM address 0x0108 - 0x0109. Here is the set up procedure in the example:

- Set SPI_byte_cnt = 0x02.
- Set SPI_addrH = 0x30, SPI_addrL = 0x05.
- Set AFSRH_temp = 0x01, AFSRL_temp = 0x09. The subroutine uses pointer as indirect addressing.
- Call DRAM2SPI_NBYTE subroutine.

2.3 Setting up SERDES for Reading EEPROM

Figure 2 shows the timing diagram for READ operation. AX1001 sends 3 bytes of data to EEPROM, 1 byte for instruction and 2 bytes for byte address. The instruction byte for READ operation is 0x03 (user may refer to AT25256 specification for more details on instruction sets).

In the example, CONFIG_SPI_BAUDRATE subroutine sets up Timer 2 for SERDES-1. SPI_READ subroutine handles READ operation. Here is a brief description in SPI_READ subroutine:

- Write 0x28 to SERDES1_CNT to set up SPI mode and bit counter (8 bits for instruction byte).
- 0x03 is written to SERDES1_BUFH (MSB comes first, as set up by SERDES1_BRC SFR in CONFIG_SPI_BAUDRATE subroutine).
- Write 0x95 to SERDES1_CTRL to start transmission.
- SPI_DONE subroutine is called to check if transmission is complete. Bit 6 of SERDES1_CTRL should be cleared by software manually once completed.
- Write 0x30 to SERDES1_CNT to set up SPI mode and bit counter (16 bits for EEPROM address).
- Write content of SPI_addrH and SPI_addrL to SERDES1_BUFH and SERDES1_BUFL, respectively.
- Write 0x95 to SERDES1_CTRL to start transmission.
- Call SPI_DONE to check if transmission is complete.
- Call CONFIG_AFSR to copy content of AFSRH_temp and AFSRL_temp to data pointer AFSRH and AFSRL, respectively.
- Write 0x28 to SERDES1_CNT to set up SPI mode and bit counter (8 bits for instruction byte)
- Write 0x00 to SERDES1_BUFH to clear buffer.
- Write 0x91 to SERDES1_CTRL to start reception (data read from EEPROM to AX1001).
- Call SPI_DONE to check if reception is complete.
- Once complete move data from SERDES1_BUFL to data RAM.
- Decrement SPI_byte_cnt and repeat the previously 5 steps until all the bytes are received.

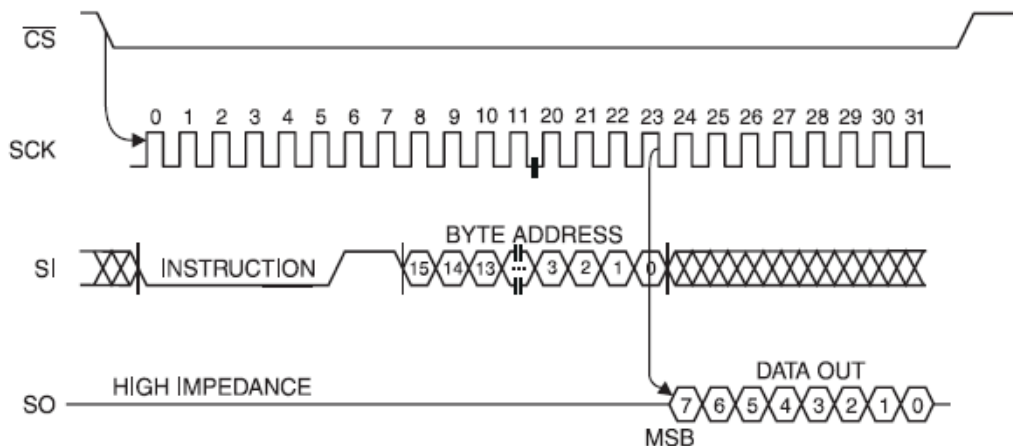


Figure 2 READ Timing Diagram

2.4 Setting up SERDES for Writing EEPROM

Figure 3 shows the timing diagram for WRITE operation. In DRAM2SPI_NBYTE subroutine, SPI_PROGRAM subroutine handles WRITE operation to EEPROM. All the operations in

SPI_PROGRAM is similar to SPI_READ, except that the WRITE operation instruction byte is 0x02.

Before calling SPI_PROGRAM, SPI_WREN is called to send a "write enabled" instruction to EEPROM. After calling SPI_PROGRAM, SPI_RDSR is called to read the EEPROM status and check if write cycle is complete.

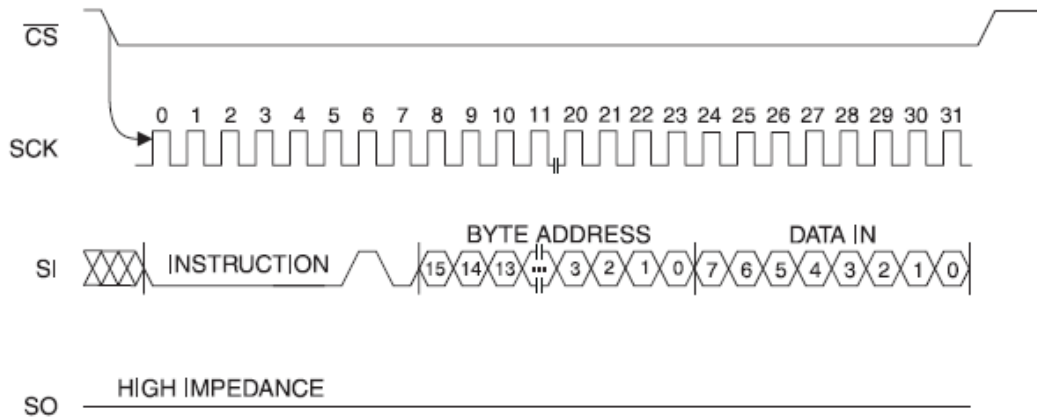


Figure 3 WRITE Timing Diagram

3.0 Source Code Example

```
#include "ax1001.inc"
;Symbols definition
#define CS_      RH,3      ;external SPI EEPROM chip select signal
;Registers definition(These registers' address is decided by users and can be changed)
SPI_addrH      EQU 100h   ;register to store high byte of SPI EEPROM address to be accessed
SPI_addrL      EQU 101h   ;register to store low byte of SPI EEPROM address to be accessed
AFSRH_temp     EQU 102h   ;register to store the AFSRH value of AX1001 DRAM to be accessed
AFSRL_temp     EQU 103h   ;register to store the AFSRL value of AX1001 DRAM to be accessed
SPI_byte_cnt   EQU 104h   ;register to store how many bytes to be read or write
Time_SPI       EQU 105h   ;used in delay subroutine as a counter
SPI_status     EQU 106h   ;register to store external SPI EEPROM STATUS register value
;Registers used for test in this code example
B64_cnt        EQU 107h   ;count how many 64 bytes data to be read or write
write_temp1    EQU 108h
write_temp2    EQU 109h
read_temp1     EQU 10Ah
read_temp2     EQU 10Bh

        org     0fffh      ;ISD mode
        jmp     MAIN

        org     0010h
;*****
;*          MAIN PROGRAM          *
;*****
MAIN PROC
;Init AX1001 PORTH(RH3---CS_)
        bs      CS_
        clr    RHDIR
;write 2BYTE(86H,90H) to external SPI EEPROM address 3005H and 3006H*****
        movw   86h        write_temp1,w    ;the address of temp1: AFSRH=01H, AFSRL=08H
        movw   90h
        mov     write_temp2,w    ;the address of temp2: AFSRH=01H, AFSRH=09H
        movw   02h
        mov     SPI_byte_cnt,w    ;write 2 Bytes
        movw   30h
        mov     SPI_addrH,w      ;the high byte of the SPI EEPROM address to be accessed
        movw   05h
        mov     SPI_addrL,w      ;the low byte of the SPI EEPROM address to be accessed
        movw   01h
```



```

mov     AFSRH_temp,w      ;store the AFSRH value of the address of temp1
movw   08h
mov     AFSRL_temp,w      ;store the AFSRL value of the address of temp1
call   DRAM2SPI_NBYTE     ;write n bytes from AX1001 Data RAM to external SPI EEPROM
;read 2BYTE from SPI EEPROM address 3005H and 3006H to AX1001 Data RAM**8*****
movw  02h
mov     SPI_byte_cnt,w    ;read 2 Bytes
movw   30h
mov     SPI_addrH,w      ;the high byte of the SPI EEPROM address to be accessed
movw   05h
mov     SPI_addrL,w      ;the low byte of the SPI EEPROM address to be accessed
movw   01h
mov     AFSRH_temp,w      ;store the AFSRH value of the address of temp1
movw   0Ah
mov     AFSRL_temp,w      ;store the AFSRL value of the address of temp1
call   SPI2DRAM_NBYTE     ;read n bytes from external SPI EEPROM to AX1001 Data RAM
;write 256BYTE(00H~FFH) from AX1001 Data RAM bank1 to external SPI EEPROM (1000H~10FFH)**
call   STORE2BANK1        ;for test use(Store 256 byte data(00H~FFH) to AX1001 data
                           ;ram bank1(300H~3FFH))

movw   10h
mov     SPI_addrH,w      ;the high byte of the SPI EEPROM address to be accessed
movw   00h
mov     SPI_addrL,w      ;the low byte of the SPI EEPROM address to be accessed
movw   03h
mov     AFSRH_temp,w      ;store the AFSRH value of the address of temp1
movw   00h
mov     AFSRL_temp,w      ;store the AFSRL value of the address of temp1
call   DRAM2SPI_256BYTE   ;write 256 byte data from AX1001 DRAM to ex SPI EEPROM
;read 256BYTE from external SPI EEPROM to AX1001 Data RAM bank2*****
movw   00h
mov     SPI_byte_cnt,w    ;read 256 Bytes
movw   10h
mov     SPI_addrH,w      ;the high byte of the SPI EEPROM address to be accessed
movw   00h
mov     SPI_addrL,w      ;the low byte of the SPI EEPROM address to be accessed
movw   05h
mov     AFSRH_temp,w      ;store the AFSRH value of the address of temp1
movw   00h
mov     AFSRL_temp,w      ;store the AFSRL value of the address of temp1
call   SPI2DRAM_NBYTE     ;read 256 byte data from external SPI EEPROM to AX1001 Data RAM
jmp     $

MAIN ENDP
;*****
; STORE2BANK1 SUBROUTINE *
;*****
;Store 256 byte data(00H~FFH) to AX1001 data ram bank1(300H~3FFH)
STORE2BANK1 PROC
movw   03h
mov     AFSRH,w
clr     AFSRL
mov     w,AFSRL
mov     AINDF,w
incjz   AFSRL
jmp     $-3
clr     AFSRH
clr     AFSRL
ret

STORE2BANK1 ENDP
;*****
; DRAM2SPI_256BYTE SUBROUTINE *
;*****
;Write 256 Bytes from AX1001 DRAM bank specified to external SPI EEPROM subroutine
DRAM2SPI_256BYTE PROC
movw   04h
mov     B64_cnt,w
LP1_DRAM2SPI_256BYTE:
movw   40h
mov     SPI_byte_cnt,w
call   DRAM2SPI_NBYTE
movw   40h
add     AFSRL_temp,w
add     SPI_addrL,w
decjz   B64_cnt
jmp     LP1_DRAM2SPI_256BYTE
ret

DRAM2SPI_256BYTE ENDP
;*****

```



```

;*****
;   SPI2DRAM_NBYTE SUBROUTINE *
;*****
;Read SPI_byte_cnt bytes data from external SPI EEPROM to AX1001 DATA RAM subroutine
SPI2DRAM_NBYTE   PROC
    call    CONFIG_SPI_BAUDRATE
    call    SPI_READ
    call    CLEAR_SPI_BAUDRATE
    ret
SPI2DRAM_NBYTE   ENDP
;*****
;   DRAM2SPI_NBYTE SUBROUTINE *
;*****
;Write SPI_byte_cnt bytes data from AX1001 DATA RAM to external SPI EEPROM subroutine
DRAM2SPI_NBYTE   PROC
    call    CONFIG_SPI_BAUDRATE
    call    SPI_WREN
    call    SPI_PROGRAM
LP1_DRAM2SPI_NBYTE:
    call    SPI_RDSR
    jbc     SPI_status,0
    jmp     LP1_DRAM2SPI_NBYTE
    call    CLEAR_SPI_BAUDRATE
    ret
DRAM2SPI_NBYTE   ENDP
;*****
;   CONFIG_SPI_BAUDRATE SUBROUTINE *
;*****
CONFIG_SPI_BAUDRATE   PROC ;SPI Baudrate Configuration Subroutine
    clr     TMR2_CNT
    clr     TMR2_PSR
    movw    05h
    mov     TMR2_PR,w
    movw    01h
    mov     TMR2_CTRL,w
    movw    04h
    mov     SERDES1_BRC,w    ;MSB first, Timer2 overflow signal as SERDES baudrate clock
    ret
CONFIG_SPI_BAUDRATE   ENDP
;*****
;   CLEAR_SPI_BAUDRATE SUBROUTINE *
;*****
CLEAR_SPI_BAUDRATE   PROC ;SPI Baudrate Clear Subroutine
    bc     SERDES1_CTRL,0    ;Disable serdes1
    clr     TMR2_CTRL
    clr     TMR2_CNT
    clr     TMR2_PSR
    clr     TMR2_PR
    ret
CLEAR_SPI_BAUDRATE   ENDP
;*****
;   SPI_DELAY SUBROUTINE *
;*****
SPI_DELAY   PROC ;SPI Delay Subroutine
    movw    0A0h
    mov     Time_SPI,w
    decjz   Time_SPI
    jmp     $-1
    ret
SPI_DELAY   ENDP
;*****
;   SPI_DONE SUBROUTINE *
;*****
SPI_DONE PROC ;SPI Done Check Subroutine
    jbs     SERDES1_CTRL,6    ;Check serdes1 done flag(1---serdes transmit/receive done)
    jmp     $-1
    bc     SERDES1_CTRL,6    ;Clear serdes1 done flag
    ret
SPI_DONE ENDP
;*****
;   CONFIG_AFSR SUBROUTINE *
;*****
CONFIG_AFSR   PROC ;AFSRH & AFSRL register configuration subroutine
    mov     w,AFSRL_temp
    mov     AFSRL,w
    mov     w,AFSRH_temp

```



```
        mov     AFSRH,w
        ret
CONFIG_AFSR ENDP
;*****
;   CLEAR_AFSR SUBROUTINE   *
;*****
CLEAR_AFSR PROC           ;Clear AFSRH & AFSRL register subroutine
        clr     AFSRH
        clr     AFSRL
        ret
CLEAR_AFSR ENDP
;*****
;   SPI_WREN SUBROUTINE   *
;*****
SPI_WREN PROC           ;Write enable external SPI EEPROM subroutine
        bc     CS_           ;External SPI EEPROM chip select signal(low active)
        movw   28h
        mov    SERDES1_CNT,w ;3pin mode SPI, 8bits transmit or receive
        movw   06h
        mov    SERDES1_BUFH,w ;External SPI EEPROM wren opcode: 0000x110
        movw   95h
        mov    SERDES1_CTRL,w ;Start Tx/Rx, Disable INT, Transmit mode, enable SERDES1
        call   SPI_DONE      ;Check serdes1 done flag
        bs     CS_
        call   SPI_DELAY     ;SPI delay subroutine
        ret
SPI_WREN ENDP
;*****
;   SPI_RDSR SUBROUTINE   *
;*****
SPI_RDSR PROC           ;Read STATUS register of external SPI EEPROM subroutine
        bc     CS_           ;External SPI EEPROM chip select signal(low active)
        movw   28h
        mov    SERDES1_CNT,w ;3pin mode SPI, 8bits transmit or receive
        movw   05h
        mov    SERDES1_BUFH,w ;External SPI EEPROM rdsr opcode: 0000x101
        movw   95h
        mov    SERDES1_CTRL,w ;Start Tx/Rx, Disable INT, Transmit mode, enable SERDES1
        call   SPI_DONE      ;Check serdes1 done flag
        movw   00h
        mov    SERDES1_BUFH,w
        movw   91h
        mov    SERDES1_CTRL,w ;Start Tx/Rx, Disable INT, Receive mode, enable SERDES1
        call   SPI_DONE      ;Check serdes1 done flag
        mov    w,SERDES1_BUFL
        mov    SPI_status,w  ;Store status register value of SPI EEPROM to SPI_status
        bs     CS_
        call   SPI_DELAY     ;SPI delay subroutine
        ret
SPI_RDSR ENDP
;*****
;   SPI_READ SUBROUTINE   *
;*****
SPI_READ PROC           ;Read data from External SPI EEPROM subroutine
        bc     CS_           ;Send read command & read address to external SPI EEPROM
        movw   28h
        mov    SERDES1_CNT,w ;3pin mode SPI, 8bits transmit or receive
        movw   03h
        mov    SERDES1_BUFH,w ;External SPI EEPROM read opcode: 0000x011
        movw   95h
        mov    SERDES1_CTRL,w ;Start Tx/Rx, Disable INT, Transmit mode, enable SERDES1
        call   SPI_DONE      ;Check serdes1 done flag
        movw   30h
        mov    SERDES1_CNT,w ;3pin mode SPI, 16bits transmit or receive
        mov    w,SPI_addrH    ;External SPI EEPROM high byte of address to be accessed
        mov    SERDES1_BUFH,w
        mov    W,SPI_addrL    ;External SPI EEPROM low byte of address to be accessed
        mov    SERDES1_BUFL,w
        movw   95h
        mov    SERDES1_CTRL,w ;Start Tx/Rx, Disable INT, Transmit mode, enable SERDES1
        call   SPI_DONE      ;Check SERDES1 done flag
        call   CONFIG_AFSR   ;Configure AX1001 DATA memory address to be accessed
LOOP_SPI_READ:         ;Read data from external SPI EEPROM
        movw   28h
        mov    SERDES1_CNT,w ;3pin mode SPI, 8bits transmit or receive
        movw   00h
        mov    SERDES1_BUFH,w
```



```
    movw    91h
    mov     SERDES1_CTRL,w    ;Start Tx/Rx, Disable INT, Receive mode, enable SERDES1
    call    SPI_DONE          ;Check serdes1 done flag
    mov     w, SERDES1_BUFH
    mov     AINDF,W          ;read data from external SPI EEPROM into AX1001 Data RAM
    inc     AFSRL
    decjz   SPI_byte_cnt
    jmp     LOOP_SPI_READ
    bs     CS_
    call    CLEAR_AFSR        ;Clear AFSRH & AFSRL register
    call    SPI_DELAY         ;SPI delay subroutine
    ret
SPI_READ ENDP
;*****
;   SPI_PROGRAM SUBROUTINE *
;*****
SPI_PROGRAM PROC          ;Program data to external SPI EEPROM subroutine
    bc     CS_                ;Send program command & program address to SPI EEPROM
    movw   28h
    mov     SERDES1_CNT,w    ;3pin mode SPI, 8bits transmit or receive
    movw   02h
    mov     SERDES1_BUFH,w  ;External SPI EEPROM program opcode: 0000x010
    movw   95h
    mov     SERDES1_CTRL,w  ;Start Tx/Rx, Disable INT, Transmit mode, enable SERDES1
    call    SPI_DONE        ;Check serdes1 done flag
    movw   30h
    mov     SERDES1_CNT,w   ;3pin mode SPI, 16bits transmit or receive
    mov     w, SPI_addrH    ;External SPI EEPROM high byte of address to be accessed
    mov     SERDES1_BUFH,w
    mov     W, SPI_addrL    ;External SPI EEPROM low byte of address to be accessed
    mov     SERDES1_BUFH,w
    movw   95h
    mov     SERDES1_CTRL,w  ;Start Tx/Rx, Disable INT, Transmit mode, enable SERDES1
    call    SPI_DONE        ;Check serdes1 done flag
    call    CONFIG_AFSR     ;Configure AX1001 DATA memory address to be accessed
LOOP_SPI_PROGRAM:
    mov     w, AINDF        ;Program data into external SPI EEPROM
    mov     SERDES1_BUFH,w
    movw   28h
    mov     SERDES1_CNT,w   ;3pin mode SPI, 8bits transmit or receive
    movw   95h
    mov     SERDES1_CTRL,w  ;Start T/R, Disable INT, Transmit mode, enable SERDES1
    call    SPI_DONE        ;Check SERDES1 done flag
    inc     AFSRL
    decjz   SPI_byte_cnt
    jmp     LOOP_SPI_PROGRAM
    bs     CS_
    call    CLEAR_AFSR     ;Clear AFSRH & AFSRL register
    call    SPI_DELAY      ;SPI delay subroutine
    ret
SPI_PROGRAM ENDP
```



Sales and Support

AppoTech Limited

Unit 110-111, 1/F, IC Development Centre,
No. 6 Science Park West Ave.,
Hong Kong Science Park,
Pak Shek Kok, N.T., Hong Kong
Tel: +(852) 2607 4090
Fax: +(852) 2607 4096

Buildwin International (Zhuhai) Limited

4/F, 3D Tech Building,
No. 184 Bailian Road,
JiDa Zhuhai,
GuangDong, PRC
Tel: +86 (756) 3882190
Fax: +86 (756) 3882193

Buildwin International (ShenZhen) Limited

Room 922, Youse Building, 6013 ShenNan Road,
Futian District, Shenzhen,
GuangDong, PRC
Tel: +86 (755) 25334930 / 25334970
Fax: +86 (755) 25334931

E-mail: info@appotech.com

Website: www.appotech.com