

# Writing and Reading to the HT24 EEPROM with the HT48 MCU Series

D/N : HA0016E

## Introduction

The HT24 Series of EEPROMs are a standard series of EEPROMs conforming to standard I<sup>2</sup>C protocol. The HT48 series of microcontrollers can easily interface to I<sup>2</sup>C devices, such as the HT24 EEPROMs, and communicate with them using their flexible and programmable CMOS I/O ports.

The HT24 series EEPROM have a total of 8 pins, three of which are device address pins A0, A1, A2. The serial data, SDA pin is bi-directional and is used to transfer addresses and data into and out of the device. The SCL line is the clock line, which is used to clock data in and out of the EEPROM. The write protect, WP pin, is used to inhibit write operations to the memory. Both read and write operations can be implemented when this pin is connected to ground. When the WP pin is high, the device can only implement read operations, with write operations inhibited. The remaining 2 pins are the VCC and VSS power supply pins. When using a Holtek HT48 series microcontroller to interface to the HT24 series of EEPROMS, the VCC, VSS, WP, A0, A1 and A2 pins are used according to the user requirements. The data and clock pins SDA and SCL are connected to two of the MCUs I/O pins for data transfer operations.

Pin Name	I/O	Description
A0-A2	I	Device Address inputs
SDA	I/O	Serial data I/O
SCL	I	Serial clock input
WP	I	Write protect
VSS	I	Negative power supply
VCC	I	Positive power supply

The HT24 series of EEPROMs is comprised of different Part Nos. according to their individual EEPROM capacity. When the EEPROM size is bigger than 1 page (256 bytes), equal to 2048 bits, the HT48 series MCU needs to control A0, A1 and A2 to determine which page is being written to. The range of EEPROM capacities within the HT24 series is shown in the following table:

Part No.	A0, A1 and A2 Operating Description	ROM Capacity
HT24LC02	The A0, A1 and A2 pins are the device address inputs. The chip select bits in the control byte, transmitted on the SDA line, must correspond with the logic levels on the A0, A1 and A2 pins for the device to respond.	2K (256x8)
HT24LC04	The A1 and A2 pins are the device address inputs. The chip select bits in the control byte, transmitted on the SDA line, must correspond with the logic levels on the A1 and A2 pins for the device to respond. The A0 pin is left floating.	4K (512x8, 2 pages)
HT24LC08	The A2 pin is the device address input. The chip select bits in the control byte, transmitted on the SDA line, must correspond with the logic levels on the A2 pin for the device to respond. The A0 and A1 pins are left floating.	8K (1024x8, 4 pages)
HT24LC16	The A0, A1 and A2 pins require no connection and are left floating.	16K (2048x8, 8 pages)

## Operating Description

This application note shows the HT48R30A-1 microcontroller interfaced to the HT24LC04 EEPROM device. The standard I<sup>2</sup>C protocol uses a two-wire interface consisting of a serial data line (SDA) and a serial clock line (SCL). When interfacing the MCU to the HT24LC04, the device address input pins, A1 and A2, are used according to the user requirements. In this example both A1 and A2 are connected to VSS. For data transfer purposes, the SCL and SDA pins are connected to pins PA.3 and PA.1 on the MCU respectively. This application program works by first writing the data 55H to a certain address within the EEPROM. After the write operation has completed, a read operation is then implemented at the same address. The data that is read back is then compared with 55H. If the data is not the same, the program will jump to fail\_out, otherwise the program will jump to ok\_end.

The program only performs a write operation to a specific address within page 0, after which the data at the same address will be read out again. The HT24 series of EEPROMs is capable of implementing other operations as long as the correct sequence of events are strictly adhered to. Special note must be made of the fact that when sending the control byte on the SDA line, the chip select bits in the control byte must correspond to the logic levels on the corresponding A2 and A1 pins for the device to respond.

## Program Example

```

;-----
;mask option:
;WDT: disabled
;WDTinstr: one clear instruction
;PA wake up: none
;pullhigh: all
;WDT OSC : on chip RC
;OSC: crystal
;sysvolt: 5.000V
;sysfreq: 1000kHz,internal
;product: 24SKDIP_B
include ht48r30a-1.inc
;file name: 4810wr.asm
;Author :Holtek Semiconductor (Shanghai) Software Department
;Purpose : Using the HT48 MCU series to interface with the HT24 series
of EEPROMS

EEPROM .section 'data'
    scl          equ    pa.3      ;define pa.3 as clock pin,
                                ;connect pa3 to scl

    scl_c        equ    pac.3
    sda          equ    pa.1      ;define pa.1 as data pin,
                                ;connect pa1 to sda

    sda_c        equ    pac.1
    read_out     equ    [70h]     ;read data register
    write_in     equ    [71h]     ;write data register
    word_address equ    [72h]     ;read/write address register
    data_8       equ    [73h]
    delay_5      equ    [77h]
    delay        equ    [78h]

EEPROMc .section 'code'
    org    00h
    jmp    start
    org    020h
start:
    mov    a,55h
    mov    write_in,a            ;write 55H
    mov    a,14h                ;write 14H as EEPROM address
    mov    word_address,a
write_data:
    clr    sda_c
    clr    scl_c

    set    sda
    set    scl
    clr    sda                    ;start signal
    clr    scl

```

```
        set    sda                ;1
        set    scl
        clr    scl

        clr    sda                ;0
        set    scl
        clr    scl

        set    sda                ;1
        set    scl
        clr    scl

        clr    sda                ;0
        set    scl
        clr    scl

        clr    sda                ;A2, A1, A0=000
        set    scl
        clr    scl

        set    scl
        clr    scl

        set    scl
        clr    scl

        set    scl                ;write 0, set as write mode
        clr    scl

        set    sda_c
        set    scl
wait_ack:
        sz     sda                ;wait for response
        jmp    wait_ack
        clr    scl
        clr    sda_c
        mov    a,08h                ;set transmission data length to 8
        mov    data_8,a
random_write:
        clr    sda
        sz     word_address.7
        set    sda
        set    scl
        clr    scl

        rl     word_address
        sdz    data_8
        jmp    random_write
        set    sda_c
```

```
    set    scl
fdev:
    sz     sda                ;wait for response
    jmp   fdev
    clr   scl
    clr   sda_c
    mov   a,08h
    mov   data_8,a
dtat_in:
    clr   sda
    sz    write_in.7
    set   sda
    set   scl
    clr   scl
    rl    write_in
    sdz   data_8
    jmp   dtat_in

    set   scl
    clr   scl

    clr   sda
    set   scl
    set   sda                ;stop signal

    mov   a,30h
    mov   delay_5,a
    mov   a,05h
    mov   delay,a
delay1:
    sdz   delay_5
    jmp   delay1
    sdz   delay
    jmp   delay1
;-----
;read
do_read:
    clr   read_out
    clr   sda_c
    clr   scl_c

    set   sda
    set   scl
    clr   sda                ;start signal

    clr   scl
    set   sda                ;1
    set   scl
    clr   scl
```

```
    clr    sda                ;0
    set    scl
    clr    scl

    set    sda                ;1
    set    scl
    clr    scl

    clr    sda                ;0
    set    scl
    clr    scl

    clr    sda
    set    scl                ;A0, A1, A2=0
    clr    scl

    set    scl                ;0
    clr    scl

    set    scl                ;0
    clr    scl

    clr    sda                ;write mode, write address
    set    scl
    clr    scl

    set    sda_c
    set    scl

wait:
    sz     sda
    jmp    wait
    clr    scl
    mov    a,08h
    mov    data_8,a
    clr    sda_c
read_address_in:
    clr    sda
    sz     word_address.7
    set    sda
    set    scl

    clr    scl
    rl     word_address
    sdz    data_8
    jmp    read_address_in

    set    sda_c
```

```
        set    scl

ack:
    sz    sda
    jmp   ack
    clr   scl
    clr   sda_c

read_data:
    set   sda
    set   scl
    clr   sda                ;start bit
    clr   scl

    set   sda                ;1
    set   scl
    clr   scl

    clr   sda                ;0
    set   scl
    clr   scl

    set   sda                ;1
    set   scl
    clr   scl

    clr   sda                ;0
    set   scl
    clr   scl

    clr   sda                ;A2, A1, A0
    set   scl
    clr   scl

    set   scl
    clr   scl

    set   scl
    clr   scl

    set   sda                ;read mode
    set   scl
    clr   scl

    set   sda_c
w_ack:
    sz    sda
    jmp   w_ack
```

```
        set    scl                ;down_edge data out
        mov    a,08h
        mov    data_8,a
        set    sda_c
random_out:
        set    scl
        call   del
        clr    scl
        call   del
        rl     read_out
        clr    read_out.0
        sz     sda
        set    read_out.0
        sdz    data_8
        jmp    random_out
        mov    a,read_out
        mov    [41h],a
        clr    sda_c                ;for stop
        set    scl
        set    sda
        clr    scl
        clr    sda
        set    sda                ;stop end

        mov    a,055h
        xor    a,read_out
        sz     acc
        jmp    fail_out
        jmp    ok_end

fail_out:
        jmp    $
ok_end:
        jmp    $

del:                ;for delay
        nop
        nop
        nop
ret
```

- Note:**
1. Take note of how pin A0 is used in the HT24 series of EEPROMs. For the HT24LC04 device pin A0 is left floating
  2. Data is read on the falling edge and written on the rising edge
  3. When a read operation is implemented, the clock frequency should not be too high. During display, the system clock frequency is 1.0MHz, however if the system clock frequency is too high, then time delays should be added in the operations, otherwise errors might occur when the data is read.
- 

This example shows read/write operations at a specific address in page 0 of the HT24 series EEPROM. If operations are to be carried out on other pages, then the corresponding changes must be made to the values of the data A0, A1, A2 transmitted on the SDA line.