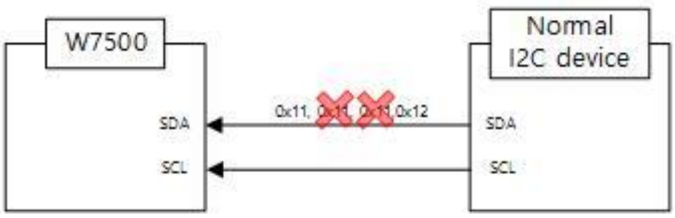
W7500x Errata Sheet

Document History

Ver 1.0.0 (July.11, 2016)	First release (erratum 1) – I2C
Ver 1.0.1 (Dec.08, 2016)	Correct SCL speed
Ver 1.1.0 (Jun.18, 2018)	erratum 2 – Transmission Delay
Ver 1.2.0 (May.12. 2019)	Erratum 3 – IAP Function Call Failure Erratum 4 – Two Image Banks Failure Erratum 5 – Cold Booting Failure

For more information, visit our website at http://www.wiznet.co.kr

Erratum 1	
Phenomenon	**Receiving repeating data in continuative data transmission causes I2C communication problem.**
Condition	 W7500 Normal I2C device SDA SCL 0x11, 0x11, 0x11, 0x12 SDA SCL W7500 receives the first repeating data but starts to discards from the 2nd repeating data to next different data in continuative data transmission. It causes data loss.
Solution & Recommendation	To avoid this issue, W7500 uses GPIO instead of I2C. In this case, SCL has limited speed, 100KHz. Example pseudo code:

```
Function Initialize_I2C ( )
{
…
    scl_port_num = I2C_PORT(conf->scl);
    scl_pin_index = I2C_PIN_INDEX(conf->scl);

    sda_port_num = I2C_PORT(conf->sda);
    sda_pin_index = I2C_PIN_INDEX(conf->sda);

    //SCL setting
    GPIO_InitDef.GPIO_Pin = scl_pin_index;
    GPIO_InitDef.GPIO_Mode = GPIO_Mode_OUT;

    if(scl_port_num == 0)
    {
        GPIO_Init(GPIOA, &GPIO_InitDef);
        GPIO_SetBits(GPIOA, scl_pin_index);
    }
….
//SDA setting
```

```
    GPIO_InitDef.GPIO_Pin = sda_pin_index;
    GPIO_InitDef.GPIO_Mode = GPIO_Mode_IN;
    if(sda_port_num == 0)
    {
        GPIO_Init(GPIOA, &GPIO_InitDef);
        GPIO_ResetBits(GPIOA, sda_pin_index);
    }
…..
}

/* SCL function */
Function I2C_SCL
{
…
    if(scl_port_num == 0)
    {
        if(data == 1)
            GPIO_SetBits(GPIOA, scl_pin_index);
        else
            GPIO_ResetBits(GPIOA, scl_pin_index);
    }
…
}

/* SDA function */
Function I2C_SDA
{
…
     if(sda_port_num == 0)
    {
        if(data == 1)
            GPIOA->OUTENCLR = sda_pin_index;
        else
            GPIOA->OUTENSET = sda_pin_index;
    }
…
```

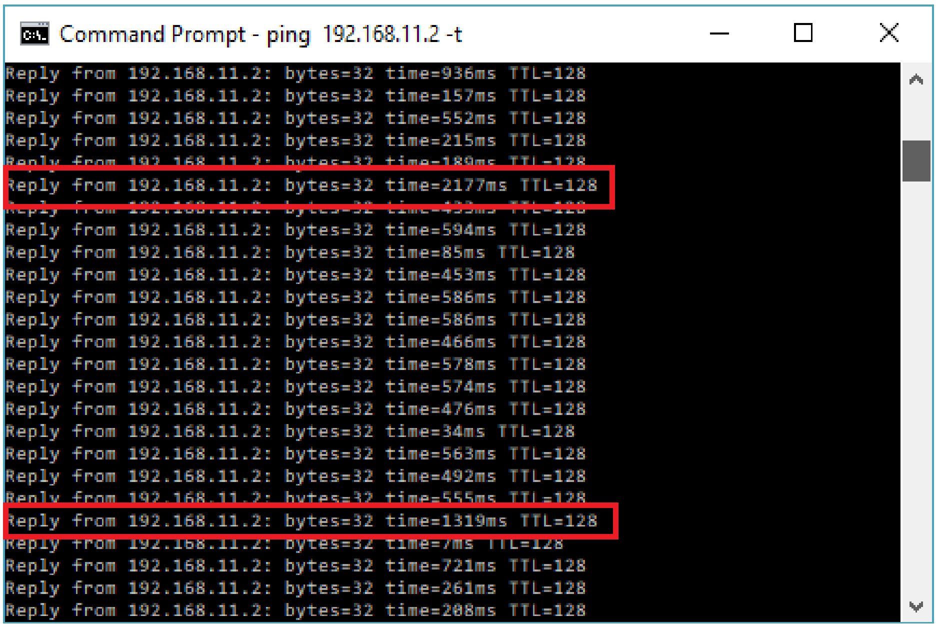
```
}
/* START function */
Function I2C_START

void I2C_Start(I2C_ConfigStruct* conf)
{
    I2C_WriteBitSCL(conf, 1);
    I2C_WriteBitSDA(conf, 1);

    I2C_WriteBitSDA(conf, 0);
    I2C_WriteBitSCL(conf, 0);
}
/* STOP function */
Function I2C_STOP

void I2C_Stop(I2C_ConfigStruct* conf)
{
    I2C_WriteBitSCL(conf, 0);
    I2C_WriteBitSDA(conf, 0);

    I2C_WriteBitSCL(conf, 1);
    I2C_WriteBitSDA(conf, 1);
}
……
```

Erratum 2

Transmission Delay Case

Phenomenon

There are some cases of data transmission delay when W7500P is connected to a particular switchor router.(The router that was used for the below test is"TP_LINK AC750")



Command Prompt - ping 192.168.11.2 -t

```
Reply from 192.168.11.2: bytes=32 time=936ms TTL=128
Reply from 192.168.11.2: bytes=32 time=157ms TTL=128
Reply from 192.168.11.2: bytes=32 time=552ms TTL=128
Reply from 192.168.11.2: bytes=32 time=215ms TTL=128
Reply from 192.168.11.2: bytes=32 time=189ms TTL=128
Reply from 192.168.11.2: bytes=32 time=2177ms TTL=128
Reply from 192.168.11.2: bytes=32 time=433ms TTL=128
Reply from 192.168.11.2: bytes=32 time=594ms TTL=128
Reply from 192.168.11.2: bytes=32 time=85ms TTL=128
Reply from 192.168.11.2: bytes=32 time=453ms TTL=128
Reply from 192.168.11.2: bytes=32 time=586ms TTL=128
Reply from 192.168.11.2: bytes=32 time=586ms TTL=128
Reply from 192.168.11.2: bytes=32 time=466ms TTL=128
Reply from 192.168.11.2: bytes=32 time=578ms TTL=128
Reply from 192.168.11.2: bytes=32 time=574ms TTL=128
Reply from 192.168.11.2: bytes=32 time=476ms TTL=128
Reply from 192.168.11.2: bytes=32 time=34ms TTL=128
Reply from 192.168.11.2: bytes=32 time=563ms TTL=128
Reply from 192.168.11.2: bytes=32 time=492ms TTL=128
Reply from 192.168.11.2: bytes=32 time=555ms TTL=128
Reply from 192.168.11.2: bytes=32 time=1319ms TTL=128
Reply from 192.168.11.2: bytes=32 time=7ms TTL=128
Reply from 192.168.11.2: bytes=32 time=721ms TTL=128
Reply from 192.168.11.2: bytes=32 time=261ms TTL=128
Reply from 192.168.11.2: bytes=32 time=208ms TTL=128
```

As shown above, there are random caseswhere the ping replyis delayed over 3 secondsand occurs irregularly.

Condition

The cause of this phenomenon is due to NC(Not Connected) pads& the connection problems related to PHY MII signals inside the chip(W7500P is silicon-in-package product and it includes W7500andEthernet PHY circuit inside.); By Collision handling due to wrong detection of duplex mode, the transmission packets are delayed.

Solution & Recommendation

In order to resolve this phenomenon, users **MUST**add the following initialization code.

```
void PHY_Init(void)
{
#ifdef __W7500P__ // W7500P only
    // PB_12
    *(volatile uint32_t *)(0x41003070) = 0x61; // RXDV: set pull down
    // PB_05
    *(volatile uint32_t *)(0x41002054) = 0x01;
    *(volatile uint32_t *)(0x41003054) = 0x61;
    // PB_06
    *(volatile uint32_t *)(0x41002058) = 0x01;
    *(volatile uint32_t *)(0x41003058) = 0x61;
    // PHY reset pin pull-up (PD_06)
    *(volatile uint32_t *)(0x410020D8) = 0x01;
    *(volatile uint32_t *)(0x410030D8) = 0x02;
    *(volatile uint32_t *)(0x45000004) = 0x40;
    *(volatile uint32_t *)(0x45000010) = 0x40;
    mdio_init(GPIOB, W7500x_MDC, W7500x_MDIO);        // MDIO Init
    mdio_write(GPIOB, PHYREG_CONTROL, CNTL_RESET);    // PHY Reset
#endif
}
```

The DUP pin(pin 15)of W7500Pshows what duplex mode it operates with the switch or router as,The value is as below.

- DUP pin = ‘1’ (HIGH) : Full duplex mode
- DUP pin = ‘0’ (LOW) : Half duplex mode

Erratum 3	
IAP Function Call Issue	
Phenomenon	The processor hangs while calling IAP function to write to internal flash memory.
Condition	The processor failure may occur when an external clock source is used.
Solution & Recommendation	Changing the clock source to internal 8MHz is the solution. 1. Change the clock source to internal 8MHz. 2. Call the IAP function for internal Flash write operations. 3. Then restore the clock source to the previous clock source. Following are example codes for before IAP function call and after it.

```
static void flash_update_start(void)
{
    /* System Core Clock Update */
    SystemCoreClockUpdate_User(CLOCK_SOURCE_INTERNAL, PLL_SOURCE_8MHz,
                                   SYSTEM_CLOCK_8MHz);

    /* SysTick_Config */
    SysTick_Config((GetSystemClock()/1000));

    /* Backup Interrupt Set Pending Register */
    temp_interrupt = (NVIC->ISPR[0]);
    (NVIC->ISPR[0]) = (uint32_t)0xFFFFFFFF;
}

/* System Core Clock Update - Restore */
static void flash_update_end(void)
{
    /* System Core Clock Update */
    SystemCoreClockUpdate_User(DEVICE_CLOCK_SELECT,
                   DEVICE_PLL_SOURCE_CLOCK, DEVICE_TARGET_SYSTEM_CLOCK);

    /* SysTick_Config */
```

```
        SysTick_Config((GetSystemClock()/1000));

        /* Restore Interrupt Set Pending Register */
        (NVIC->ISPR[0]) = temp_interrupt;
}
```

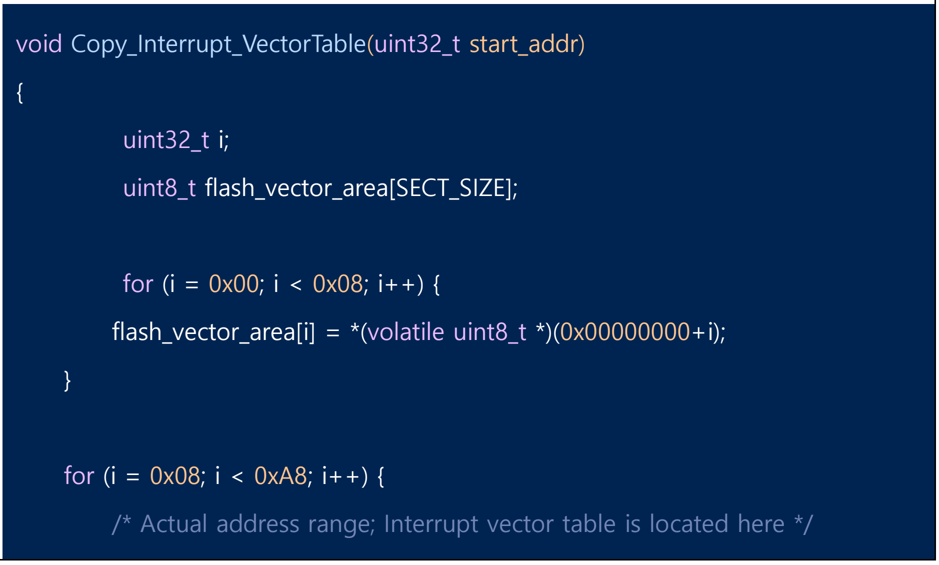
For reference,
https://github.com/Wiznet/WIZ750SR/blob/master/Projects/S2E_App/src/Configuration/segcp.c, line 1438

<table>
<tr><th colspan="2">Erratum 4</th></tr>
<tr><th colspan="2">Two Image Banks Issue</th></tr>
<tr><td><b>Phenomenon</b></td><td>The processor hangs during Bank 2 operation if the internal flash memory is divided into dual banks to implement functions such as firmware update.</td></tr>
<tr><td><b>Condition</b></td><td>This problem occurs because W7500x does not support 'interrupt vector table remaps'. If internal flash memory is divided and used as a dual bank, Bank 1 application starting from address 0 operates without problems. However, if an interrupt occurs while the Bank 2 application is operating, a process stuck occurs because the interrupt vector table at address 0 is referenced.</td></tr>
<tr><td><b>Solution & Recommendation</b></td><td>
A user can avoid the problem by copying the interrupt vector table in the bank 2 area to address 0 if the user utilizes the Bank 1 as the boot area and Bank 2 as the application area.

The following conditions must be met:

1. In the case of most functions operate on bank 2 and bank 1 only uses internal flash update
2. All interrupts must be disabled in bank 1 code
3. The solution of Erratum 3 must be guaranteed since the IAP operation is performed when copying the interrupt vector table.

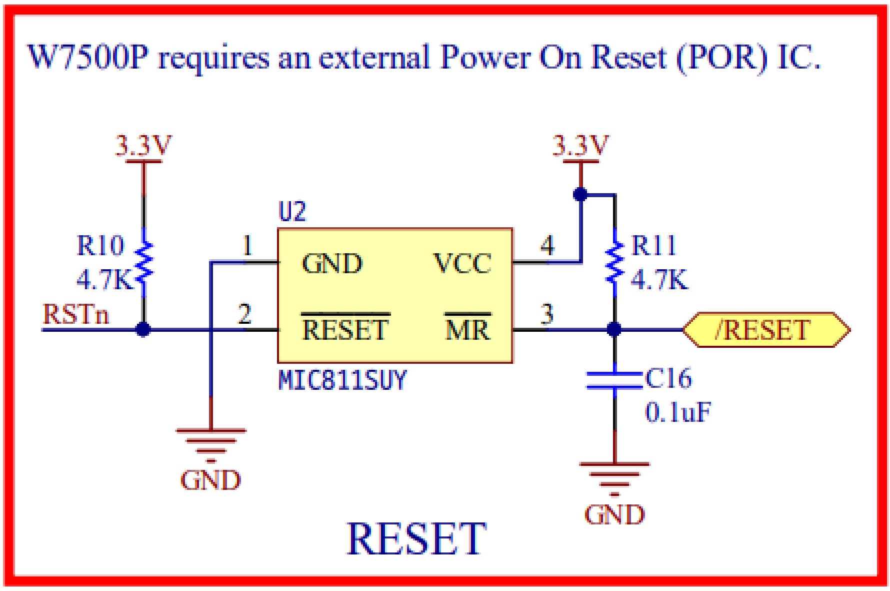
Following is an example.



```
void Copy_Interrupt_VectorTable(uint32_t start_addr)
{
        uint32_t i;
        uint8_t flash_vector_area[SECT_SIZE];

        for (i = 0x00; i < 0x08; i++) {
       flash_vector_area[i] = *(volatile uint8_t *)(0x00000000+i);
    }

    for (i = 0x08; i < 0xA8; i++) {
       /* Actual address range; Interrupt vector table is located here */
```
</td></tr>
</table>

```
            flash_vector_area[i] = *(volatile uint8_t *)(start_addr+i);
        }

             for (i = 0xA8; i < SECT_SIZE; i++) {
            flash_vector_area[i] = *(volatile uint8_t *)(0x00000000+i);
        }

           /* Global interrupt disabled */
             __disable_irq();

           /* Erase the interrupt vector table area : Sector 0 */
             DO_IAP(IAP_ERAS_SECT, 0x00000000, 0, 0);

           /* Write the applicaion vector table to 0x00000000 */
             DO_IAP(IAP_PROG, 0x00000000, flash_vector_area , SECT_SIZE);

           /* Global interrupt enabled */
             __enable_irq();
}
```

For reference,
https://github.com/Wiznet/WIZ750SR/blob/master/Projects/S2E_Boot/src/main.c, line 532

Erratum 5	
Cold Booting Failure	
Phenomenon	Cold booting failure
Condition	If the time that Power supply raises to the operating supply voltage, 2.7V is longer than 20ms or a different phase power source over 1V is supplied to any peripheral pin before Power on, the internal reset logic of the MCU doesn't run and this happens.
Solution & Recommendation	The best solution is to add POR(Power On Reset) chip  W7500P requires an external Power On Reset (POR) IC. 3.3V 3.3V U2 R10 4.7K RSTn 1 2 GND VCC RESET MR 4 3 R11 4.7K /RESET MIC811SUY C16 0.1uF GND GND RESET To avoid to add POR, you should guarantee that 0-to-2.7V time is shorter than 20ms and peripheral pins are isolated from external power source. For reference, https://github.com/Wiznet/Hardware-Files-of-WIZnet/tree/master/01_iMCU/W7500P/Reference%20Schematic