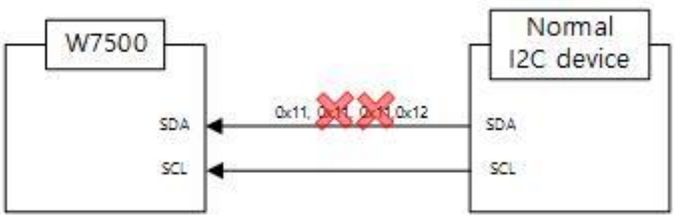
W7500P Errata Sheet

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

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

<table>
<tr><th colspan="2">Erratum 1</th></tr>
<tr><td><b>Phenomenon</b></td><td><b>Receiving repeating data in continuative data transmission causes I2C communication problem.</b></td></tr>
<tr><td><b>Condition</b></td><td>

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.</td></tr>
<tr><td><b>Solution & Recommendation</b></td><td>To avoid this issue, W7500 uses GPIO instead of I2C. In this case, SCL has limited speed, 100KHz.

Example pseudo code:
<pre>
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;

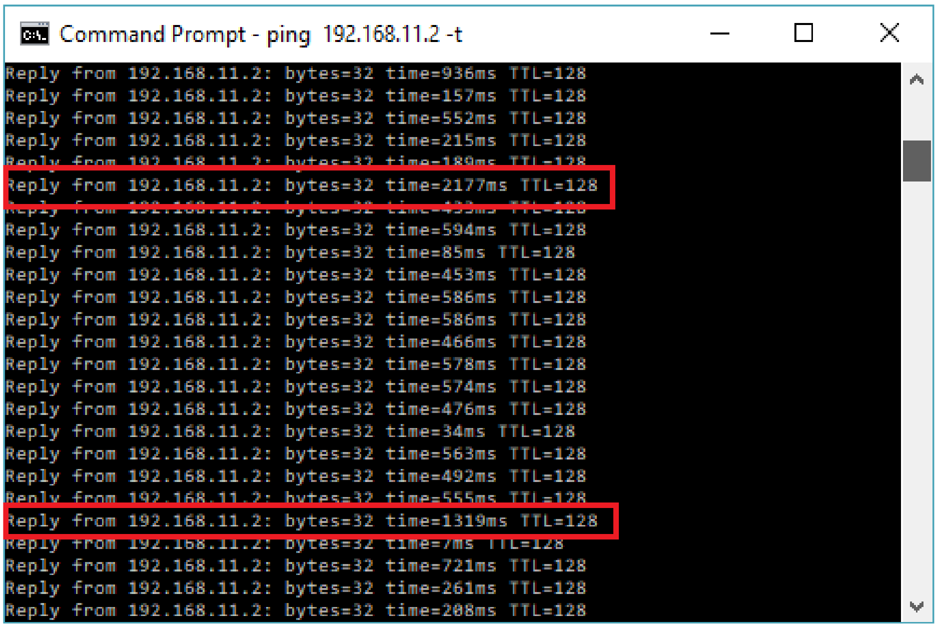
    if(scl_port_num == 0)
    {
        GPIO_Init(GPIOA, &GPIO_InitDef);
        GPIO_SetBits(GPIOA, scl_pin_index);
    }
….
//SDA setting
</pre></td></tr>
</table>

```
    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);
}
……
```

<table>
<tr><td colspan="2">Erratum 2</td></tr>
<tr><td colspan="2">Transmission Delay Case</td></tr>
<tr><td>Phenomenon</td><td>

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

<pre>
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
</pre>

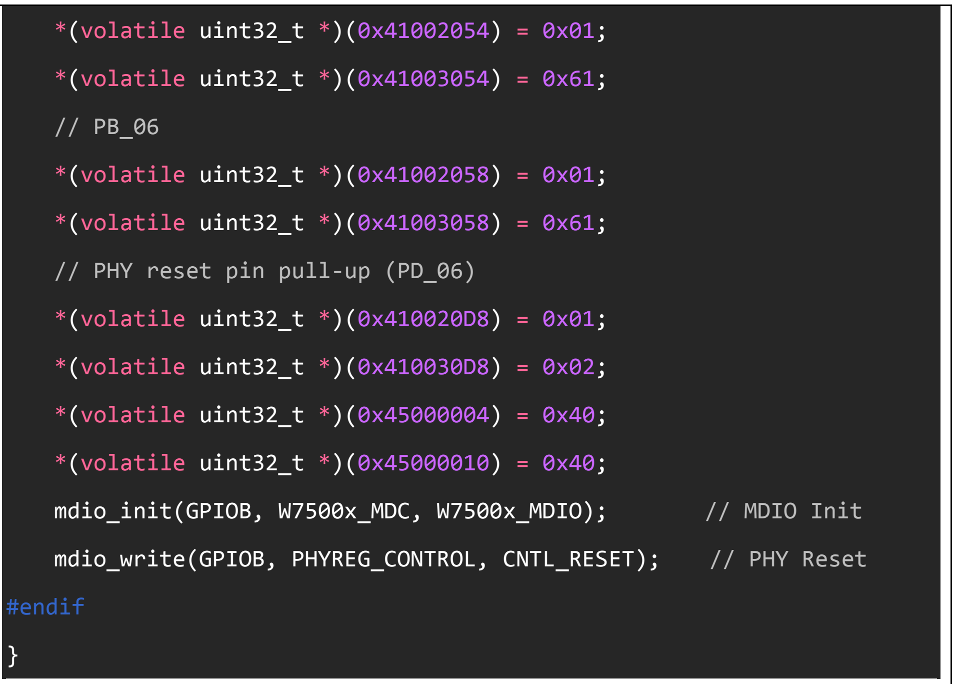
As shown above, there are random caseswhere the ping replyis delayed over 3 secondsand occurs irregularly.

</td></tr>
<tr><td>Condition</td><td>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.</td></tr>
<tr><td>Solution & Recommendation</td><td>

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

<pre>
void PHY_Init(void)
{
#ifdef __W7500P__ // W7500P only
    // PB_12
    *(volatile uint32_t *)(0x41003070) = 0x61; // RXDV: set pull down
    // PB_05
</pre>

</td></tr>
</table>



```
    *(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