

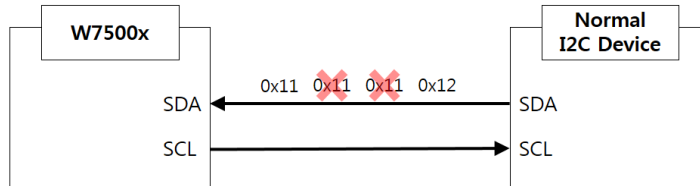
W7500x Errata Sheet

Document History

Ver 1.0.0 (July.11, 2016)	First release, erratum 1 - W7500x I2C
Ver 1.0.1 (Dec.08, 2016)	Correct SCL speed
Ver 1.1.0 (Jun.18, 2018)	erratum 2 - W7500P Transmission Delay Case

© 2016 WIZnet Co., Inc. All Rights Reserved.

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

Erratum 1	
W7500x I2C	
Phenomenon	Receiving repeating data in continuative data transmission causes I2C communication problem.
Condition	<div data-bbox="547 521 1249 705" data-label="Diagram">  <pre> graph LR W7500x[W7500x] <--> SDA Device[Normal I2C Device] W7500x <--> SCL Device subgraph Data_Flow direction LR D1[0x11] --> D2[0x11] D2 --> D3[0x11] D3 --> D4[0x12] end style D2 stroke-dasharray: 5 5 style D3 stroke-dasharray: 5 5 style D2 stroke:red,stroke-width:2px style D3 stroke:red,stroke-width:2px </pre> </div> <p>W7500x 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.</p>
Solution & Recommendation	<p>To avoid this issue, W7500x uses GPIO instead of I2C. In this case, SCL has limited speed, 100KHz.</p> <p>Example pseudo code:</p> <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); } ... } </pre>

```
//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);

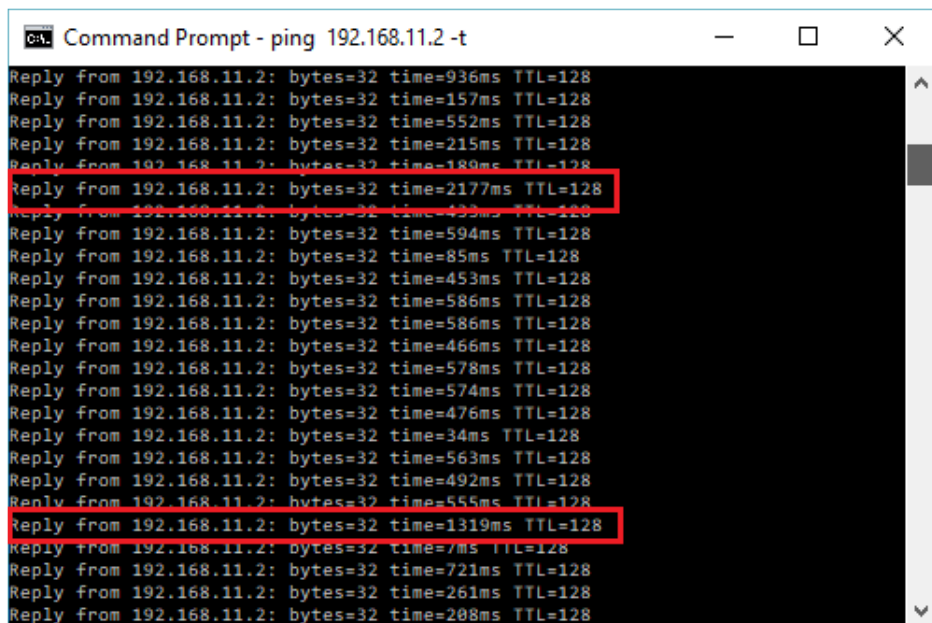
    I2C_WriteBitSCL(conf, 1);
    I2C_WriteBitSDA(conf, 1);
}
.....
```

Erratum 2

W7500P Transmission Delay Case

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

Phenomenon



As shown above, there are random cases where the ping reply is delayed over 3 seconds and 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 W7500 and Ethernet 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 W7500P shows 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