

# How to Access PHY Register

## Application Note

Version 1.0.0



<http://www.wiznet.co.kr>

## Table of Contents

1	Introduction .....	4
2	How to Access PHY Register .....	5
2.1	Write Access.....	5
2.2	Read Access .....	5
2.3	Get PHY Address.....	5
	Document History Information.....	7

## List of table

Table 1 Interface Format .....	4
--------------------------------	---

## List of figures

Figure 1 MDC/MDIO Write Format.....	4
Figure 2 MDC/MDIO Read Format .....	4

# 1 Introduction

W7500P PHY's Register can be accessed by MDC and MDIO. Users can control MDC/MDIO through GPIOx. MDC/MDIO format is shown in the below table. To access PHY Register in W7500P, MDC should be at least one more cycle than MDIO. That is, a complete command consists of 32 bits MDIO data and at least 33 MDC clocks. When this interface is idle, MDIO is in high impedance.

Table 1 Interface Format

Frame format	<Idle><start><op code><PHY address><Register address><turnaround><data><Idle>
Read Operation	<Idle><01><10><A <sub>4</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub> ><R <sub>4</sub> R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> ><Z0><b <sub>15</sub> b <sub>14</sub> b <sub>13</sub> b <sub>12</sub> b <sub>11</sub> b <sub>10</sub> b <sub>9</sub> b <sub>8</sub> b <sub>7</sub> b <sub>6</sub> b <sub>5</sub> b <sub>4</sub> b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub> ><Idle>
Write Operation	<Idle><01><01><A <sub>4</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub> ><R <sub>4</sub> R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> ><10><b <sub>15</sub> b <sub>14</sub> b <sub>13</sub> b <sub>12</sub> b <sub>11</sub> b <sub>10</sub> b <sub>9</sub> b <sub>8</sub> b <sub>7</sub> b <sub>6</sub> b <sub>5</sub> b <sub>4</sub> b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub> ><Idle>

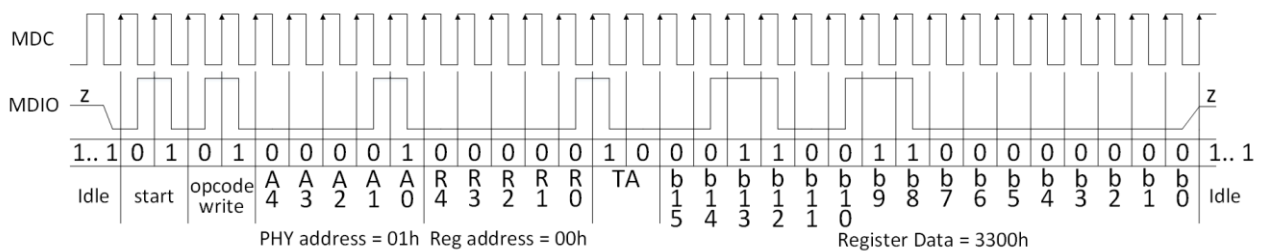


Figure 1 MDC/MDIO Write Format

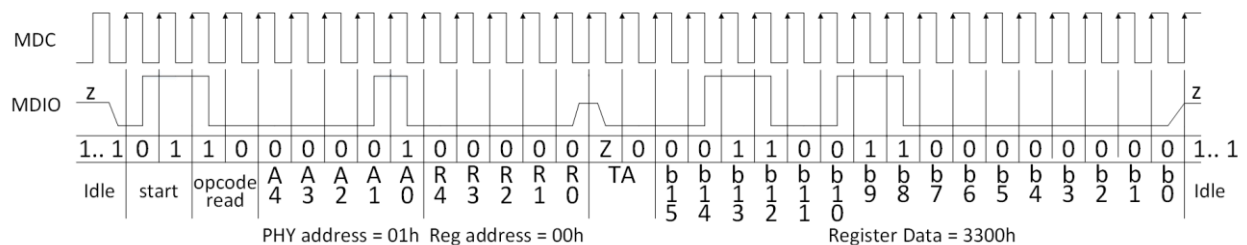


Figure 2 MDC/MDIO Read Format

## 2 How to Access PHY Register

### 2.1 Write Access

```
{
START:
    // Set GPIO(Value, Length)
    Set GPIO(0x05, 4); // Start bits 01, Write Access 01
    Set GPIO(PHY address, 5); // PHY address
    Set GPIO(REG address, 5); // MII register
    Set GPIO(0x02, 2); // turnaround bits 10
    Set GPIO(DATA, 16);
}
```

### 2.2 Read Access

```
{
START:
    // Set GPIO(Value, Length)
    Set GPIO(0x06, 4); // Start bits 01, Read Access 10
    Set GPIO(PHY address, 5); // PHY register
    Set GPIO(REG address, 5); // MII register
    Set GPIO(CLR, 2); // turnaround bits high impedance
    Val = Get GPIO(DATA, 16);
}
```

### 2.3 Get PHY Address

```
{
START:
    // Loop to find PHY address
    for(i=0; i<8; i++)
    {
        Set GPIO(0x05, 4); // Read Access
        Set GPIO(i, 5); // PHY address
        Set GPIO(0x01, 5); // PHY Status Register (0x01)
        Set GPIO(CLR, 2); // turnaround bits high impedance
        Val = Get GPIO(DATA, 16); // To check LINK bit in PHY Status Register.
        if(Val != 0) return i; // i is PHY address
    }
}
```

}

## Document History Information

Version	Date	Descriptions
Ver. 1.0.0	7AUG2018	Initial Release

---

## Copyright Notice

Copyright 2015 WIZnet Co., Ltd. All Rights Reserved.

Technical Support: <http://wizwiki.net/forum>

Sales & Distribution: [sales@wiznet.co.kr](mailto:sales@wiznet.co.kr)

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