# HS-WIZ610IO User Manual

(Version 1.1)

微信公众号

手机淘宝商城

# 1. Introduction

HS-WIZ610IO is the internet offload network module that includes W6100 (TCP/IP hardwired chip, include PHY), MAG-JACK (RJ45 with X'FMR) with other glue logics. It can be used as a component and no effort is required to interface W6100 and Transformer.

The HS-WIZ610IO is an ideal option for users who want to develop their Internet enabling systems rapidly.

**For the detailed information on implementation of Hardware TCP/IP, refer to the W6100 Datasheet.**

HS-WIZ610IO consists of W6100 and MAG-JACK.

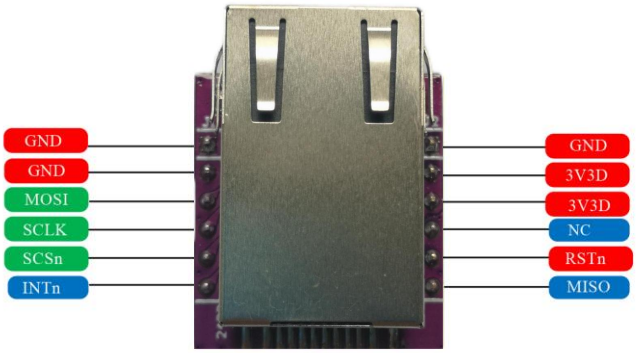
- Hardware TCP/IP, Ethernet MAC : Included in W6100
- Ethernet PHY : Included in W6100
- Connector : MAG-JACK(RJ45 with Transformer)

## 1.1 Feature

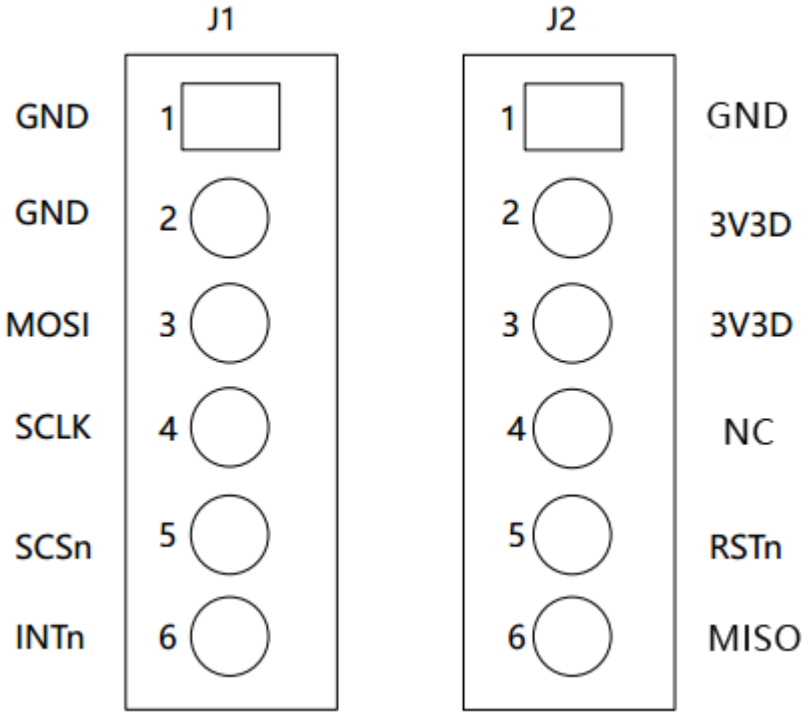
- Support Hardwired TCP/IP Protocols : TCP, UDP, IPv6, IPv4, ICMPv6, ICMPv4, IGMP, MLDv1, ARP, PPPoE
- Support IPv4/IPv6 Dual Stack
- Support 8 independent SOCKETs simultaneously with 32KB Memory
- Support SOCKET-less Command:
  ARP, PING, ICMPv6(PING, ARP,DAD,NA,RS) Command for IPv6 Auto-configuration& Network Monitoring
- Support Ethernet Power Down Mode & Main Clock Switching for power save
- Support Wake on LAN over UDP
- Support Serial Interface: High Speed SPI(MODE 0/3)
- Internal 16Kbytes Memory for TX/ RX Buffers
- 10BaseT/100BaseTX Ethernet PHY Integrated
- Support Auto Negotiation (Full and half duplex, 10 and 100-based )
- Support Auto-MDIX only on Auto-Negotiation Mode
- 3V operation with 5V I/O signal tolerance
- Network Indicator LEDs (Full/Half Duplex, Link, 10/100 Speed, Active)
- Interfaces with two 2.54mm pitch 1 x 10 header pin
- Temperature : -40 ~ 85℃(Operating)

# 2. Pin assignment & description

## 2.1 HS-**WIZ610**IO **Pin assignment**




**< TOP side view >**




**< Pin assignment >**

Pin Type Notation

| Type | Description |
|---|---|
| I | Input |
| O | Output |
| P | Power & Ground |

## 2.1.2 HS-WIZ610IO Pin description

| Pin No. | | I/O | Pin Name | Description |
|---|---|---|---|---|
| J1 | 1 | P | GND | **Ground** |
| | 2 | P | GND | **Ground** |
| | 3 | I | MOSI | **SPI Master Out Slave In**<br>This pin is used to SPI MOSI signal pin. |
| | 4 | I | SCLK | **SPI Clock**<br>This pin is used to SPI Clock Signal pin. |
| | 5 | I | SCSn | **SPI Slave Select :** Active Low<br>This pin is used to SPI Slave Select signal Pin when using SPI interface. |
| | 6 | O | INTn | **Interrupt** : Active low<br>When the event occur during W6100 Ethernet Communication, INTn notices to HOST.<br>Low : Interrupt Occurred<br>High : No Interrupt<br>Refer to IEN (Interrupt pin Enable) in SYCR1 (System Config Register1), INTPTMR (Interrupt Pending Time Register), IR (Interrupt Register), SIR (Socket Interrupt Register), SLIR (SOCKET-less Interrupt Register). |
| J2 | 1 | P | GND | **Ground** |
| | 2 | P | 3V3D | **Power** : Digital 3.3V power |
| | 3 | P | 3V3D | **Power** : Digital 3.3V power |
| | 4 | I | NC | **NC** |
| | 5 | I | RSTn | **Reset :** RSTn initializes W6100. RSTn must |

| | | | | |
|---|---|---|---|---|
| | | | | be asserted to Low longer than 1.0us. After asserted RSTn, W6100 spends 60.3ms for initialization.<br>Low：W6100 initialized.<br>High：Normal Operation. |
| | **6** | **O** | **MISO** | **SPI Master In Slave Out**<br>This pin is used to SPI MISO signal pin. |

## Tip:

For more information about chip W6100 and its application, please visit http://www.hschip.com/down.aspx?TypeId=56&Fld=t14:56:14

Datesheet：


W6100数据手册 2019-06-11 点击下载


W6100_ds_v100e 2019-03-04 点击下载

Drives:


1-W6100_TCPS(STM32F103) 2020-03-18 点击下载


2-W6100_TCPC(STM32F103) 2020-03-18 点击下载


3-W6100_UDP(STM32F103) 2020-03-18 点击下载

# 3. HS-WIZ610IO SPI operations

HS-WIZ610IO is controlled by a set of instruction that is sent from a external host , commonly referred to as the SPI Master. The SPI Master communicates with W6100 via the SPI bus, which is composed of four signal lines: Slave Chip Select (SCSn), Serial Clock (SCLK), MOSI (Master Out Slave In ) and MISO (Master In Slave Out).

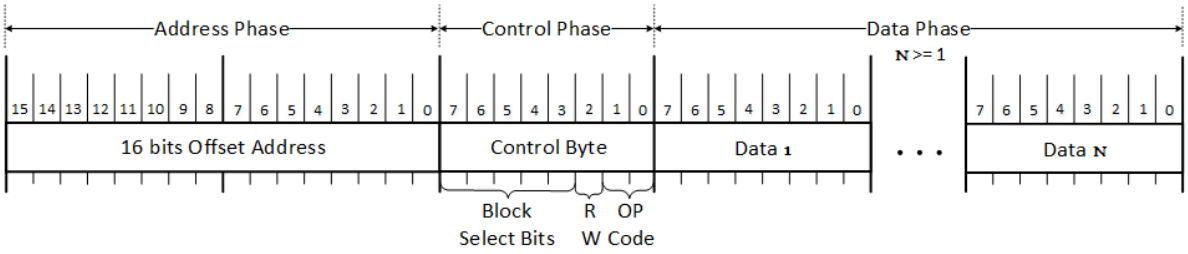
The SPI protocol defines four modes for its operation (Mode 0-3). Each mode differs according to the SCLK polarity and phase - how the polarity and phase

control the flow of data on the SPI bus. The W6100 operates as SPI Slave device and supports the most common modes - SPI Mode 0 and 3.

The only difference between SPI Mode 0 and 3 is the polarity of the SCLK signal at the inactive state. With SPI Mode 0 and 3, data is always latched in on the rising edge of SCLK and always output on the falling edge of SCLK.

## 3.1 Process of using general SPI Master device

1. Configure Input/Output direction on SPI Master Device pins
2. Configure SCSn as 'High' on inactive
3. Write target address for transmission on SPDR register (SPI Data Register)
4. Write Control Byte for transmission on SPDR register
5. Write desired data for transmission on SPDR register
6. Configure SCSn as 'Low' (data transfer start)
7. Wait for reception complete
8. If all data transmission ends, configure SCSn as 'High'




**< W6100 SPI Frame Format >**

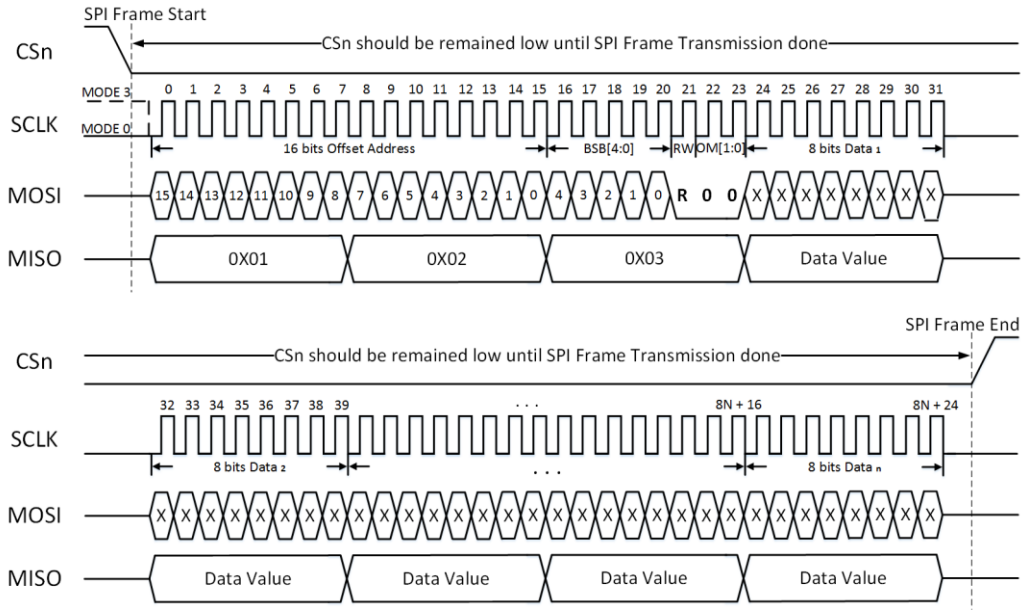
## 3.2 Read processing

In the VDM, CSn(High-to-Low) by HOST informs the start of SPI frame and CSn(Low to High) by HOST informs the end of SPI frame to W6100.

In the control phase, RW is "0" to indicate read access and OP[1:0] is "00" to indicate VDM.

The data bits received through MISO are synchronized to SCLK (Falling-Edge).

If more than one byte of data is transmitted continuously, it supports sequential data write.




**< Read Sequence >**

```
uint8_t tAD[3];   //send buffer
// Address
tAD[0] = (uint8_t)((AddrSel & 0x00FF0000) >> 16);
tAD[1] = (uint8_t)((AddrSel & 0x0000FF00) >> 8);
tAD[2] = (uint8_t)(AddrSel & 0x000000ff);

wizchip_cs_select(); //CS=0, SPI start

tAD[2] |= (_W6100_SPI_READ_ | _W6100_SPI_OP_);   // Control Byte

wizchip_spi_write_buf(tAD,3); //write address
for(idx = 0; idx < len; idx++)                    // Write data in loop
{
      buf[idx] = IINCHIP_SpiSendData(0x00);
 }
```

```
wizchip_cs_deselect();//CS=1, SPI end
```

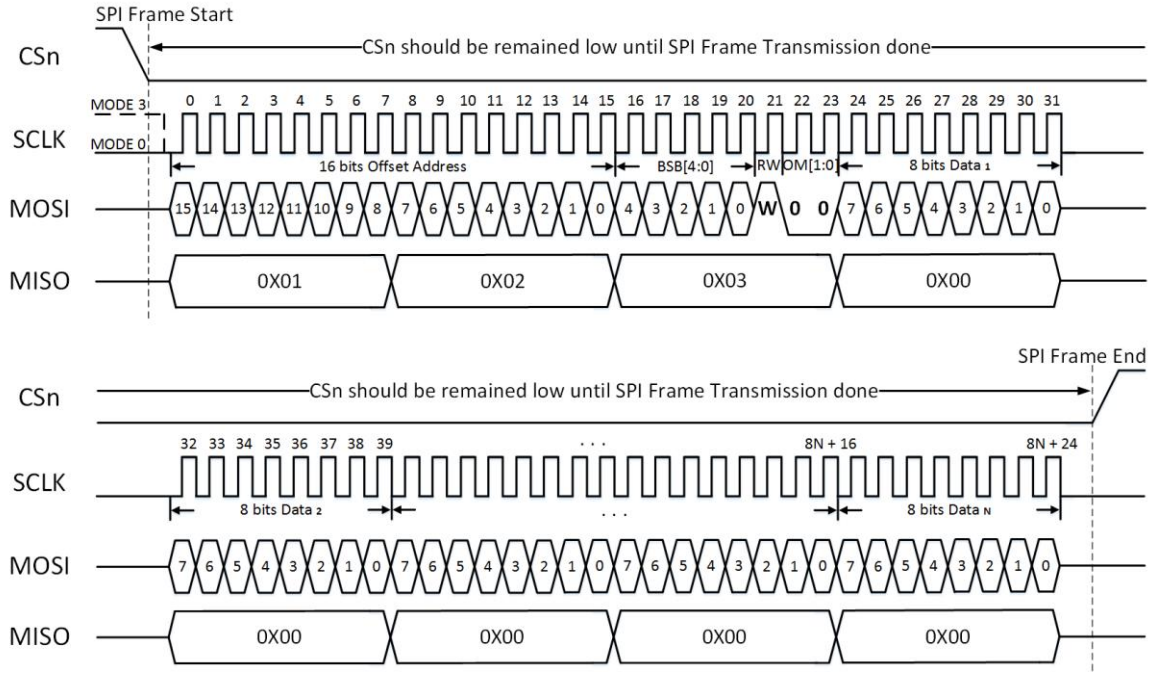
## 3.3 Write processing

In the VDM, CSn(High-to-Low) by HOST informs the start of SPI frame and CSn(Low to High) by HOST informs the end of SPI Frame to W6100.

In the control phase, RW is "1" to indicate write access and OM[1:0] is "00" to indicate VDM.

The data bits transmitted through MOSI are synchronized to the SCLK (Falling-Edge).

If more than one byte of data is transmitted continuously, it supports sequential data write.



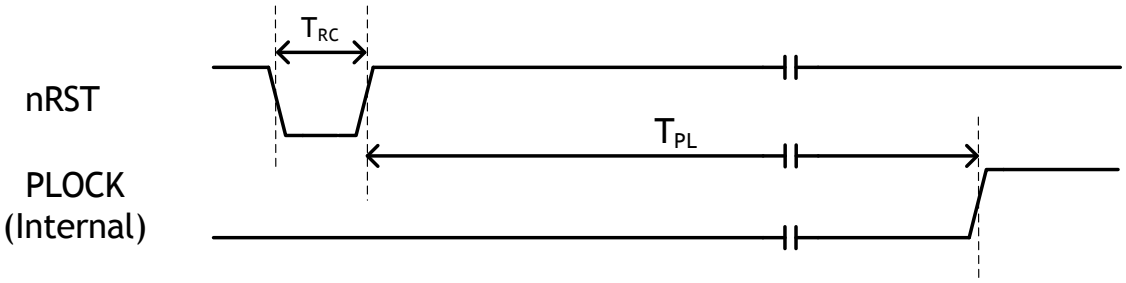

**< Write Sequence >**

```
uint8_t tAD[3];   //send buffer
 // Address
 tAD[0] = (uint8_t)((AddrSel & 0x00FF0000) >> 16);
 tAD[1] = (uint8_t)((AddrSel & 0x0000FF00) >> 8);
 tAD[2] = (uint8_t)(AddrSel & 0x000000ff);
 wizchip_cs_select(); //CS=0, SPI start

 tAD[2] |= (_W6100_SPI_WRITE_ | _W6100_SPI_OP_);   // Control Byte
```

```
    wizchip_spi_write_buf(tAD, 3);   //write data
    for(idx = 0; idx < len; idx++)                    // Write data in loop
    {
        IINCHIP_SpiSendData(buf[idx]);
    }

wizchip_cs_deselect();//CS=1, SPI end
```

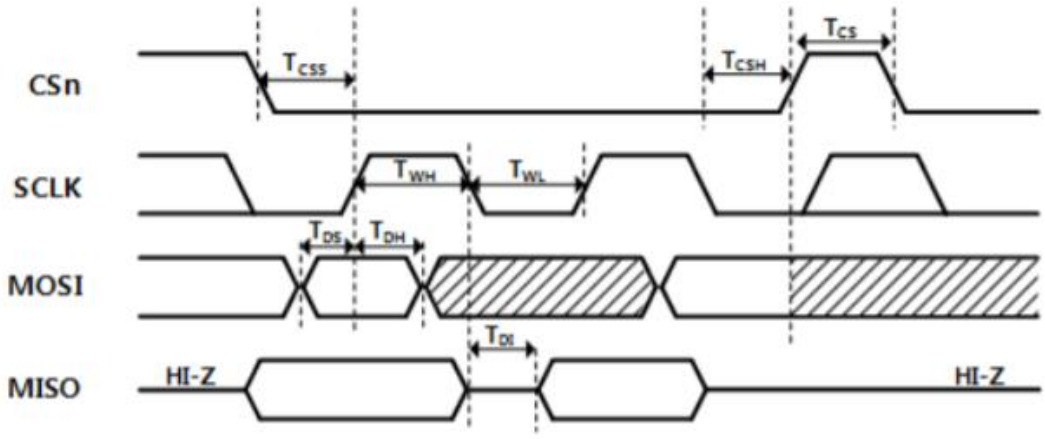
# 4. Timing diagram

## 4.1 Reset Timing




| Symbol | Description | Min | Typ | Max |
|---|---|---|---|---|
| $T_{RC}$ | Reset Time | 350ns | 580ns | 1.0us |
| $T_{PL}$ | Stable Time | - | - | 60.3ms |

## 4.2 SPI Timing



| Symbol | Description | Min | Max | Units |
|---|---|---|---|---|
| $F_{SCK}$ | SClK Clock Frequency | | 70 | MHz |
| $T_{CSS}$ | CSn Setup Time | 3 SYS_CLK | - | ns |
| $T_{CSH}$ | CSn Hold Time | 2 SYS_CLK | | ns |
| $T_{CS}$ | CSn High Time | 2 SYS_CLK | | ns |
| $T_{WH}$ | SCLK High time | 3 | | ns |
| $T_{WL}$ | SCLK Low Time | 3 | | ns |
| $T_{DS}$ | Data Setup Time | 3 | | ns |
| $T_{DH}$ | Data In Hold Time | 3 | | ns |
| TDI | Data Invalid Time | 7 | | ns |

# 5. Dimensions