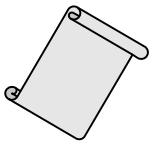
# EIP-2000 Quick Start

**V1.0, February 2023**

## Packing List

In addition to this guide, the package includes the following items:

EIP-2000



Quick Start

Screw Driver

CN-1824

EIP-2019/S Only

## Technical Support

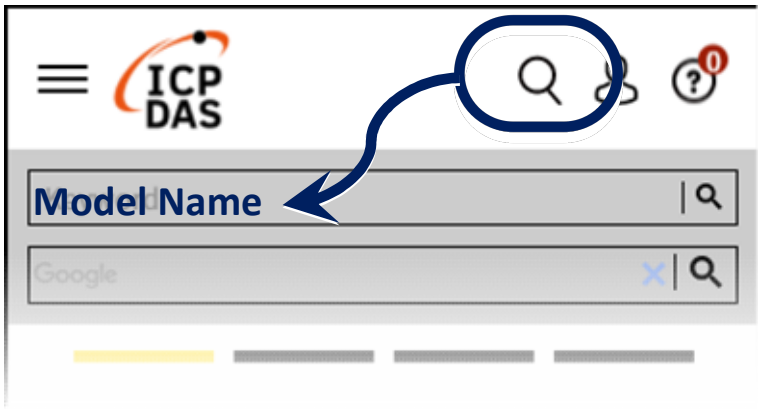
service@icpdas.com

www.icpdas.com

## Resources

How to search for drivers, manuals and spec information on ICP DAS website.

- For Mobile Web




- For Desktop Web

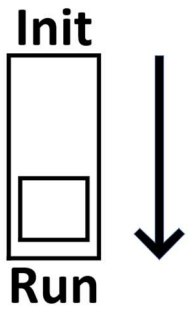

## Installing Software on your PC

**Install EIP-2000 Utility:**

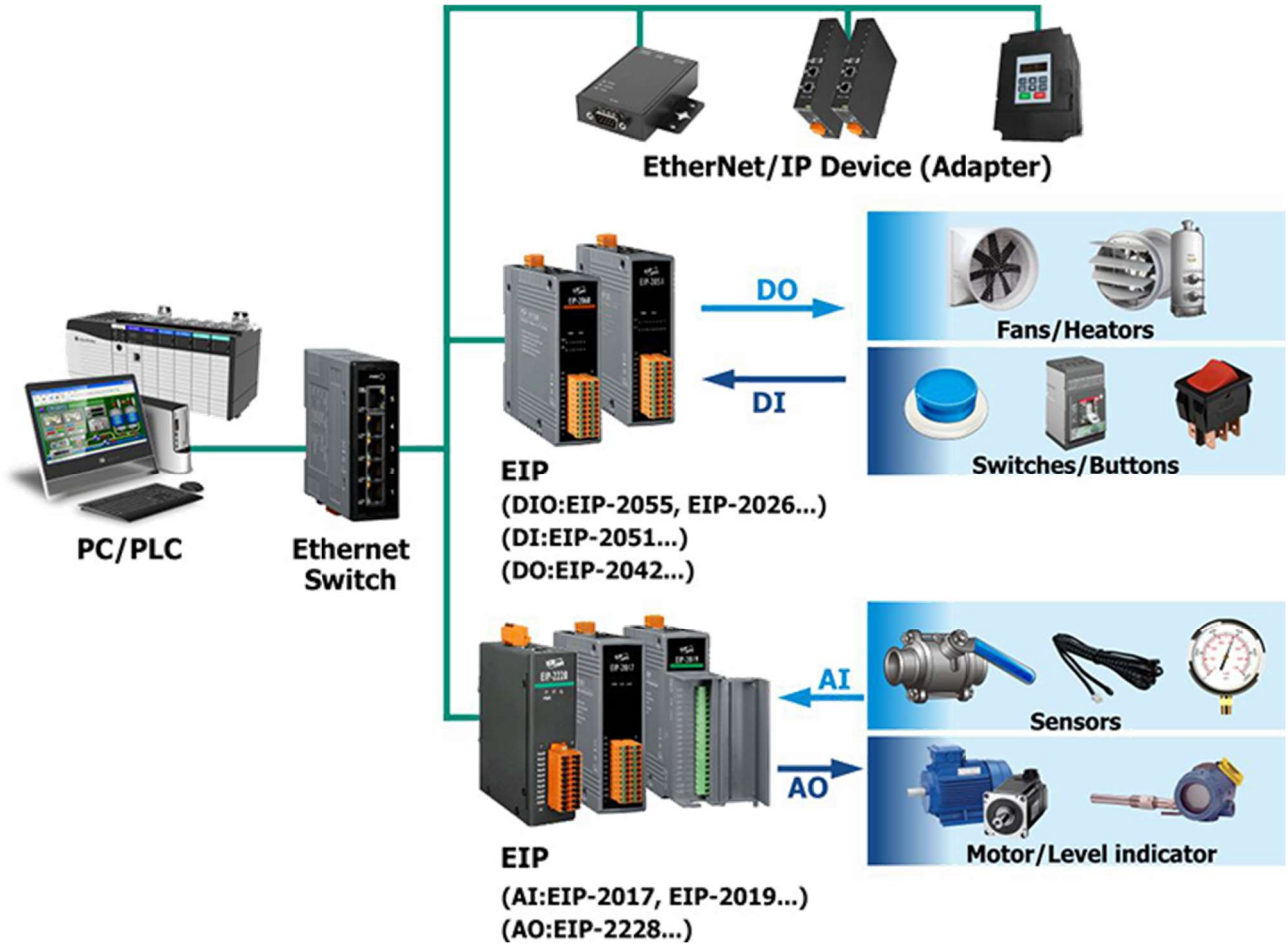
http://www.icpdas.com/en/download/show.php?num=3080

## Connecting the Power and PC

1. Make sure your PC has workable network settings.
2. Disable or well configure your Windows firewall and anti-virus firewall first, else the “Network Scan” may not work. (Please contact with your system Administrator)
3. Check Init/Run DIP switch if it is on Run position.




4. Connect both the EIP-2000 and your computer to the same sub network or the same Ethernet switch, and power the EIP-2000 on.



5. Please download the datasheet from the ICPDAS website for more pin-assignment and wire-connection details. The LED indicator table is shown in the following table.

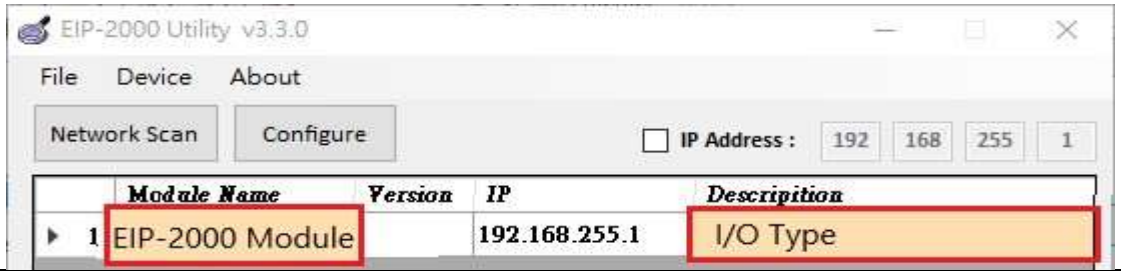
| LED Indicator | | |
|---|---|---|
| **LED** | **LED Status** | **Description** |
| **Power LED** | Always On | Module is in Run mode. |
| | Flashing | Module is in Init mode. |
| **Status LED** | Always On | EtherNet/IP connection is failed. |
| | Blink per second | EtherNet/IP connection is successful. |
| | Blink per 300 ms | EtherNet/IP disconnected during communication but still in Safe-Delay time. |
| | Blink per 100 ms | Module is about to reboot. |
| **I/O status LED** | On | The DI/DO is activated. |
| | Off | The DI/DO is inactivated. |

## EIP-2000 Utility

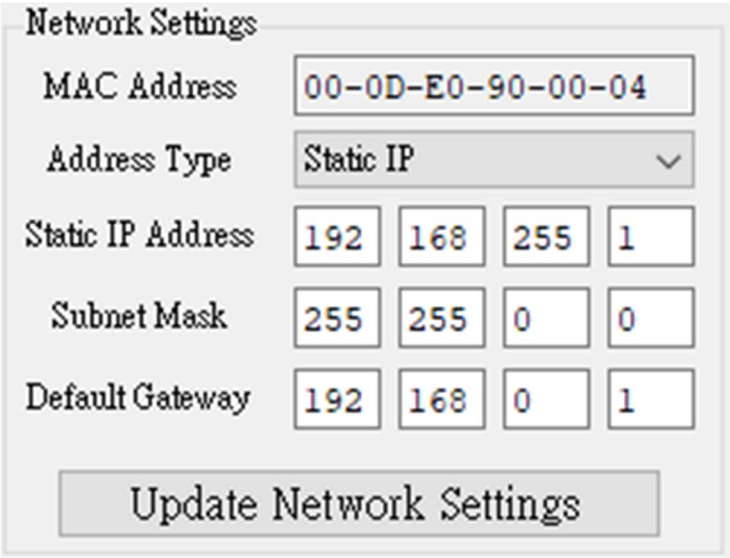
1. Double click the EIP-2000 Utility shortcut on the desktop.
2. The factory default IP address of the EIP-2000 modules.

| Network Settings | |
|---|---|
| **Item** | **Settings (default)** |
| **IP** | 192.168.255.1 |
| **Gateway** | 192.168.0.1 |
| **Mask** | 255.255.0.0 |

3. Click the "**Network Scan**" button to search your EIP-2000 modules. Or check "**IP Address**" to enter the IP address of the module, and then click the "**Configure**" button.

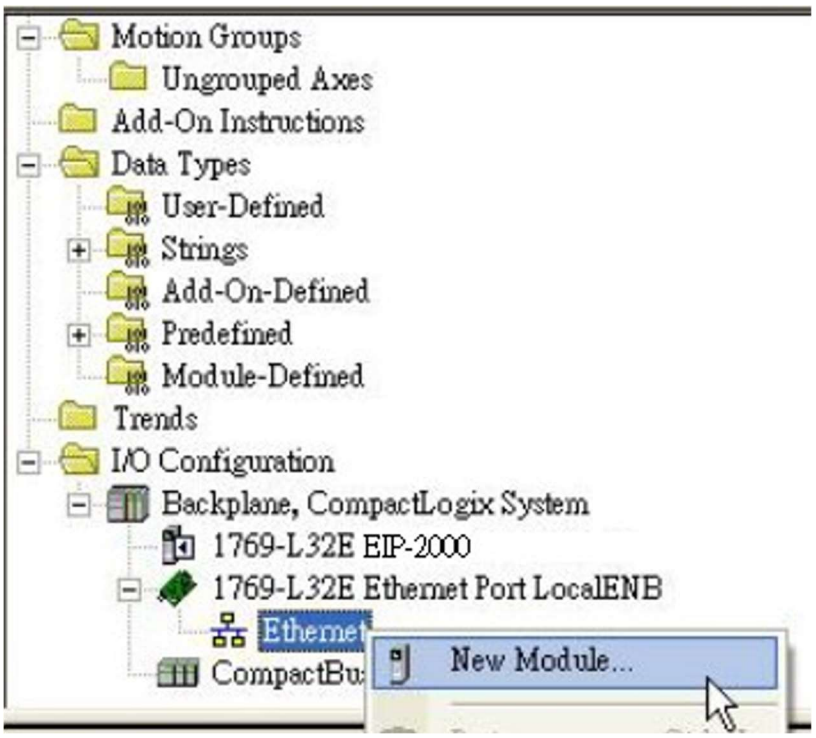


4. Users can edit the “Network Settings” for changing the modules’s IP address. And click the **“Update Network Settings”** button to update the configuration and reboot the module.

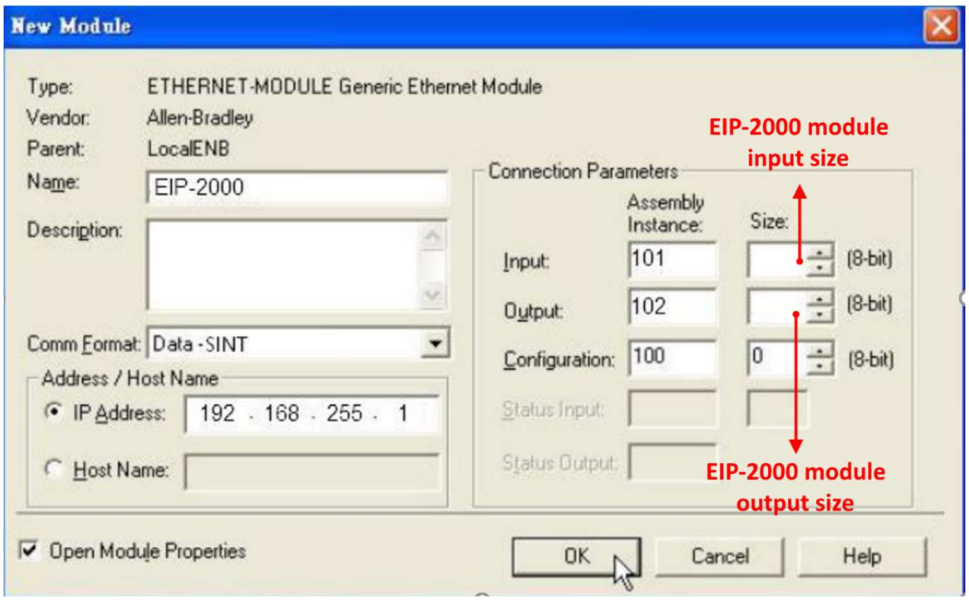



## How to connect with Allen-Bradley PLC?

1. Open RSLogix 5000 (AB PLC tool) and create a new project, and create a new module in the “Ethernet” item.



2. Set the parameters of the new module. The input assembly instain is 101, the output assembly instance is 102 and the configuration assembly instance is 100.




Please find the IO size from the following tables, and the configuration size is always 0.

- EIP-2017 data size

| Data Assembly | Byte count | Description |
|---|---|---|
| Input Assembly | 53 | 1st ~ 16th Byte: AI status (AI0 ~ AI7) for DIFF. or S.E. mode. |
| | | 17th ~ 32nd Byte: AI status (AI8 ~ AI15) for S.E. mode only. |
| | | 33rd ~ 40th Byte: AI Type Code ( AI0 ~ AI7) for DIFF. or S.E. mode. |
| | | 41st ~ 48th Byte: AI Type Code (AI0 ~ AI7) for S.E. mode only. |
| | | 49th Byte: AI filters status. |
| | | 50th Byte: Channel mode status. |
| | | 51st Byte: AI representation. |
| | | 52nd Byte: Channel selection (AI0 ~ AI7). |
| | | 53rd Byte: Channel selection (AI8 ~ AI15). |
| Output Assembly | 22 | 1st Byte: Set value to the module. |
| | | 2nd ~ 17th Byte: Set type code to AI0 ~ AI15. |

| | | |
|---|---|---|
| | | 18th Byte: Filter selections of AI |
| | | 19th Byte: Channel mode selection DIFF. or S.E. |
| | | 20th Byte: AI representations |
| | | 21st Byte: AI channel selection (AI0 ~ AI7) |
| | | 22nd Byte: AI channel selection (AI8 ~ AI15) |

- EIP-2019 data size

| Data Assembly | Byte count | Description |
|---|---|---|
| **Input Assembly** | **41** | 1st ~ 16th Byte: AI status(AI0 ~ AI7). |
| | | 17th ~ 18th Byte: The broken wire status. |
| | | 19th ~ 20th Byte: CJC status. |
| | | 21st ~ 28th Byte: AI type code(AI0 ~ AI7). |
| | | 29th Byte: AI filter status. |
| | | 30th Byte: AI representation. |
| | | 31st Byte: Wire break detector. |
| | | 32nd Byte: CJC switch. |
| | | 33rd Byte: CJC increment. |
| | | 34th ~ 41st Byte: CJC offset(AI0 ~ AI7). |
| **Output Assembly** | **23** | 1st Byte: Set value to the module. |
| | | 2nd ~ 9th Byte: Set type code to Ch0 ~ Ch7. |
| | | 10th Byte: Filter selection of AI |
| | | 11st Byte: Wire break detector |
| | | 12nd Byte: AI representation |
| | | 13rd Byte: Select AI channel to be short |
| | | 14th Byte: CJC switch |
| | | 15th Byte: CJC increment |
| | | 16th ~ 23rd Byte: CJC Offset |

- EIP-2042 data size

| Data Assembly | Byte count | Description |
|---|---|---|
| **Input Assembly** | **2** | 1st Byte: DO status read back (DO0 ~ DO7). |
| | | 2nd Byte: DO status read back (DO8 ~ DO15). |
| **Output Assembly** | **2** | 1st Byte: DO status (DO0 ~ DO7). |
| | | 2nd Byte: DO status (DO8 ~ DO15). |

- EIP-2051 data size

| Data Assembly | Byte count | Description |
|---|---|---|
| **Input Assembly** | **66** | 1st Byte: DI status(DI0 ~ DI7). |
| | | 2nd Byte: DI status(DI8 ~ DI15). |
| | | 3rd ~ 65th Byte: DI counters. |
| **Output Assembly** | **2** | 1st Byte: to set DI counters zero (DI0 ~ DI7). |
| | | 2nd Byte: to set DI counters zero (DI8 ~ DI15). |

- EIP-2055 data size

| Data Assembly | Byte count | Description |
|---|---|---|
| **Input Assembly** | **34** | 1st Byte: DI status. |
| | | 2nd Byte: DO status read back |
| | | 3rd ~ 34th Byte: DI counters. |
| **Output Assembly** | **2** | 1st Byte: DO status. |
| | | 2nd Byte: to set DI counters zero. |

- EIP-2060 data size

| Data Assembly | Byte count | Description |
|---|---|---|
| **Input Assembly** | **26** | 1st Byte: DI status. |
| | | 2nd Byte: DO status read back |
| | | 3rd ~ 26th Byte: DI counters. |
| **Output Assembly** | **2** | 1st Byte: DO status. |
| | | 2nd Byte: to set DI counters zero. |

The Instance ID of the EIP-2000 module.

| Implicit Message Information of EIP-2000 | | |
|---|---|---|
| **Instance** | **Instance ID** | **Data length** |
| **Input(T->O)** | 65hex (101) | Depends on modules. |
| **Output(O->T)** | 66hex (102) | Depends on modules. |
| **Configuration** | 64hex (100) | |