Connecting to an iMS over Ethernet after installing the SDK

Simple configuration

Host PC

Ethernet Switch

iMS
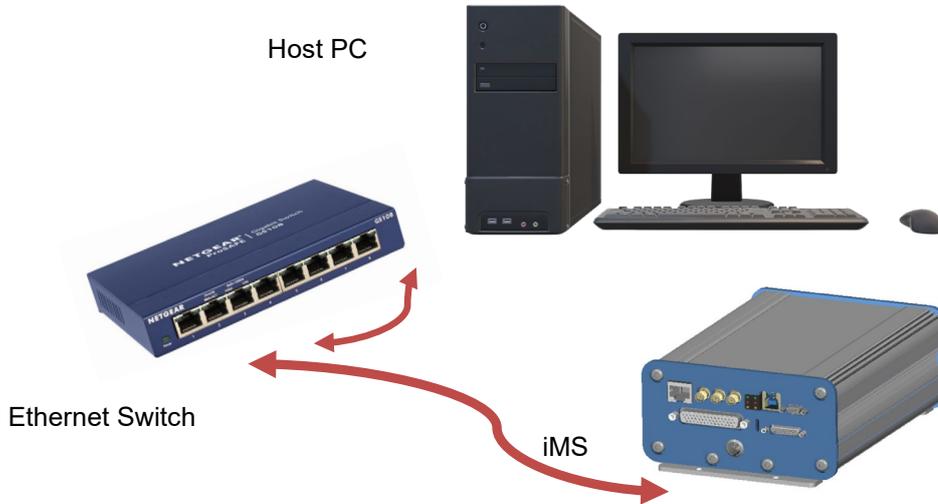
Considerations:

(a) IP Address

By default, iMS-P Controllers are configured to acquire an IP Address from DHCP. If there is no DHCP Server on the network, the iMS-P will default to its internal IP Address which is pre-configured at Isomet to 192.168.1.10

- **Configuring the IP address: SDK utility over USB.**

The SDK includes the *iMS IP Configuration* utility which can be found in the start menu. This allows the IP address and subnet of the iMS to be configured over USB.
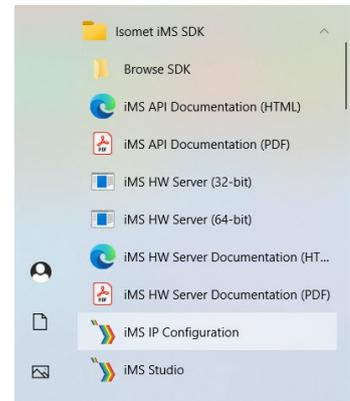
1: Check *Include USB in Scan* and hit *SCAN*
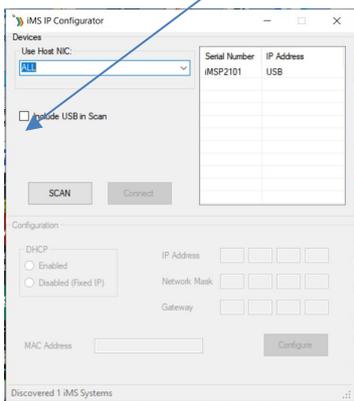
2: Select (highlight) the iMS and hit *Connect.*
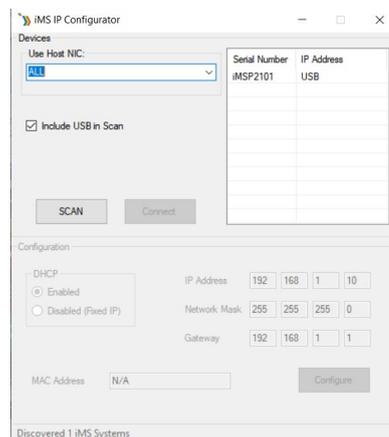
3:   Configure as required
e.g. Change to fixed IP 192.168.1.10
Hit *Configure*
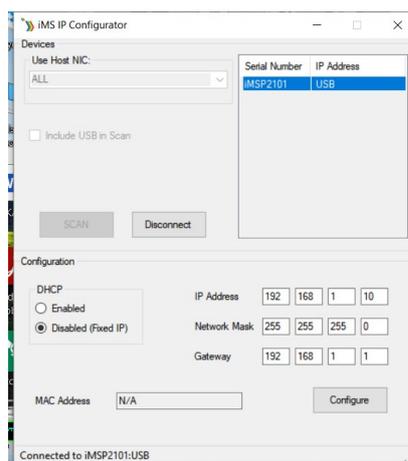
-   or -

- **Configuring the IP address: Coding in C++**

The iMS can be configured by the user C++ code to use a static IP address and also modify the IP network mask and gateway.

To retrieve and apply an IP address makes use of ***IMSSystem::ApplySettings*** and ***IMSSystem::RetrieveSetting*s** methods through which the CS_ETH object is passed with the desired IP address parameters. These will store the settings in non-volatile memory on the iMS Controller so that it will start up with the new parameters on next boot. E.g.

> CS_ETH cs_eth;
>
> // Get existing settings from iMS Controller
> myiMS.RetrieveSettings(cs_eth);
> std::cout << "Current IP Address: " << cs_eth.Address() << std::endl;
>
> cs_eth.Address("192.168.2.100");
> std::cout << "New IP Address: " << cs_eth.Address() << std::endl;
>
> myiMS.ApplySettings(cs_eth);

See code documentation in ***CS_ETH.h*** for details.
 (e.g. ***C:\Program Files\Isomet\iMS_SDK\v1.x.x\include\CS_ETH.h***).

To apply above change:
4:   Cycle the DC power (Off and On)

(b) Enabling/ disabling comms interface and selecting defined port in C++ code

```
/*
 * Uncomment these defines to include the types of interface on which you wish to communicate with the iMS
 */
//#define  ENABLE_SERIAL_PORT_SCAN
#define  ENABLE_ETHERNET_SCAN
// #define  ENABLE_USB_SCAN

/*
 * Uncomment these defines to restrict iMS scan to specific interface ports (using conventional interface type port naming)
 * Note that it is possible to add additional ports by inserting extra lines of "PortMask.push_back(...)" code below
 */
// #define SERIAL_PORT_MASK "COM3"
#define ETHERNET_PORT_MASK  "192.168.2.100"
```

Note: An equivalent control is not yet a feature within the GUI (SDK 1.8.9)
(c) Firewall

The iMS Studio and its accompanying *ims_hw_server* process requires a port open in the PC's
firewall to enable the scan for iMS systems procedure to complete successfully. Normally this is done
during the installation process (there is a checkbox *"Add Firewall Exception"* that defaults to
enabled but can be disabled if required by the installer).

To add the firewall exception manually,
do the following for Windows 7.
(Windows10 should be similar):

From the start menu type *"Firewall"*
to bring up a list of actions and select
 "Allow a program through Windows Firewall"
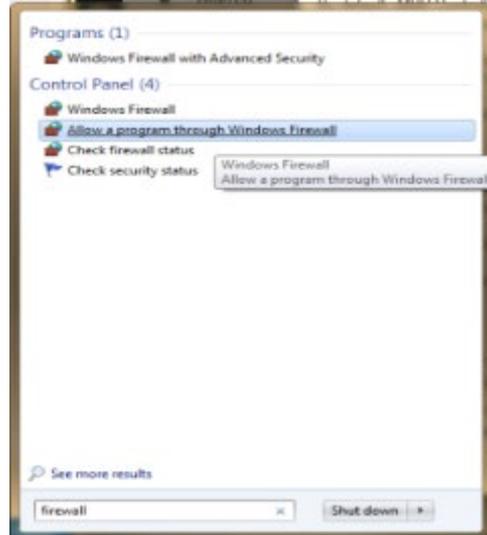
Click "Allow Another Program..."
then "Browse..." and navigate to *"C:\Program Files\Isomet\iMS_SDK\v1.x.x\utils\iMS_Studio"*
(assuming default SDK install location).

Choose *"ims_hw_server.exe"*

If you receive an error message
"The application is already in the list
of exceptions" look for
"iMS Studio HW Server"
in the list and make sure the
checkbox alongside it is checked.
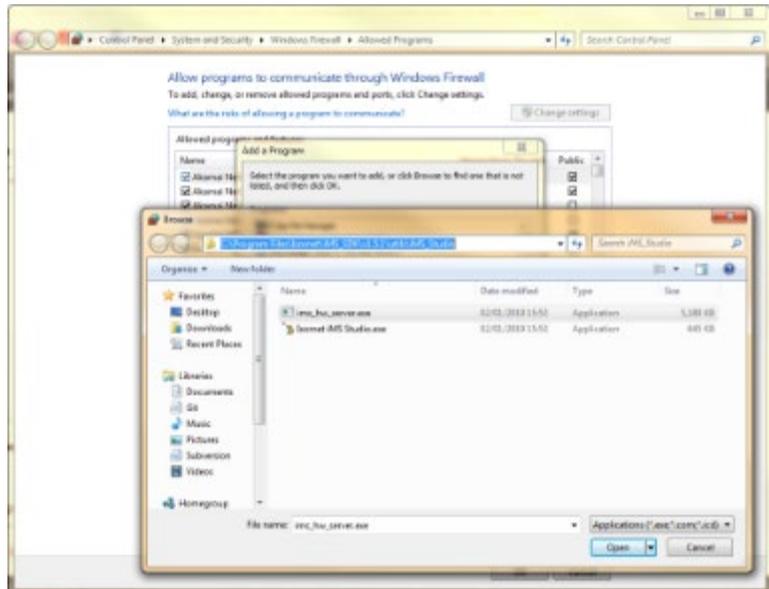
☑ iMS Studio	☐	☑
☑ iMS Studio HW Server	☑	☐
☑ iMS Studio HW Server	☐	☑
☑ ims_fw_upgrade	☑	☐

Also, check for any third party firewall or antivirus software which may be blocking communications
with the *ims_hw_server.exe* application.