AN-280

Configuring HTTPS Connection to the Protege WX Controller

Application Note

Last Published: 04-Nov-24 2:56 PM

Contents

Introduction

HTTPS (Hypertext Transfer Protocol Secure), an extension of the HTTP protocol, encrypts data transfer between web server and browser for better information security. Browsers now strongly discourage the use of unencrypted HTTP, making HTTPS the basis of the modern web.

Protege WX controllers come preconfigured with a self-signed certificate and HTTPS enabled by default, so that communications between the controller and the web browser are always encrypted. However, an alternative certificate can be installed if preferred. Installing a third-party certificate on the controller will remove the security warning which you may see in your browser when accessing a controller with a factory certificate.

This application note describes how to install a new HTTPS certificate on a controller to replace the default factory certificate. For older controllers without a default HTTPS certificate it may be possible to install an HTTPS certificate after upgrading the controller's operating system. This is **strongly recommended** for any controller that is connected to internal or external networks via a router.

Two different certification methods are available, each of which can be configured directly within the web interface:

- Validating and installing a third-party certificate obtained from a certificate authority.
- Installing a self-signed certificate (recommended for testing only).

If the controller is factory defaulted any user-created HTTPS certificates are removed and its default certificate is reloaded. Custom certificates will need to be reinstalled.

Prerequisites

The following controllers support custom HTTPS certificates.

Controller	Firmware Version
Protege WX Controller	4.00.452 or higher

HTTPS support also depends on the controller's operating system version. Read the information below before attempting to implement HTTPS.

Additional Requirements for HTTPS Support

The HTTPS encryption feature is dependent on the controller's operating system (OS) version. The supported operating systems depend on the hardware type of the controller, as older controller hardware does not support newer operating systems.

OS Version	Two-door with USB port	Two-door without USB port	One-door with USB port	One-door without USB port
2.0.25 or higher	✅	N/A	✅	N/A
2.0.0 - 2.0.24	☑	N/A	N/A	N/A
1.33.145 or lower	N/A	⊗	N/A	⊗

✅ = Full support for HTTPS with TLS 1.2

☑ = HTTPS is not supported, but it is possible to upgrade the OS to a supported version.

⊗ = No support for HTTPS. It is not possible to upgrade the OS to a supported version.

N/A = OS and hardware versions not compatible

Checking Controller Compatibility

Before implementing HTTPS you should check each controller to ensure that it supports this feature.

> If it is not practical to check each controller individually, contact ICT support with a list of the controller serial numbers.

First, identify the hardware type of the controller. Newer controllers have a USB port next to the ethernet port.

- Controllers with USB ports **may** support HTTPS (depending on the operating system).
- Controllers without USB ports **do not** support HTTPS and cannot be retrofitted.

To identify your controller's operating system version:

1. Log in to the web interface and navigate to **Application Software**.
2. Click on the **Current Version** number. This should expand to reveal additional versioning information.
3. Check and record the **OS** version.

   > If no additional versioning information is displayed when you click on the **Current Version**, the controller's OS is lower than 2.0.20.

Based on the controller's OS version, you will have one of two results:

- **Version 2.0.25 or higher**: HTTPS is fully supported with TLS 1.2. Third-party and self-signed certificates are supported.
  - No action is required.
- **Version 2.0.24 or lower**: HTTPS is not supported, but the OS can be upgraded.
  - Contact ICT Technical Support for information about upgrading your controller's operating system to a supported version.

Connectivity Requirements for HTTPS

To acquire a third-party certificate for HTTPS connection to the controller's web interface, the controller must be accessible over the internet. This section discusses some of these requirements so that the system can be properly prepared for HTTPS implementation.

Operating on an active network requires knowledge of the configuration and structure of the network. Always consult the network or system administrator before you begin.

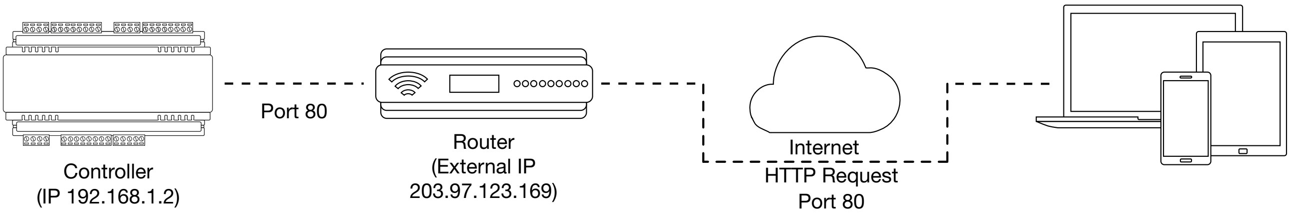
More Information

- For detailed networking information, see the Protege WX Network Administrator Guide.
- For basic information on Protege WX controller networking see AN-189: Protege WX Connectivity Guide.

Port Forwarding Requirements

In order for the controller to be accessible externally, port forwarding must be configured at the router. Port forwarding is a method of mapping an IP address and port on a local subnet to an external port, so that the networked device is accessible over the internet.

In particular, validating a third-party certificate generally requires the controller to be accessible via **external port 80**. This is the default port for HTTP requests. This external port must be set up to forward traffic to an internal port on the controller that accepts HTTP requests. By default this is **internal port 80**; however, if required this can be changed in the **System Settings**.



Once this port has been forwarded, the controller will be accessible via the external IP address of the network. In this example, typing 203.97.123.169 into an external web browser will open the controller's web interface.

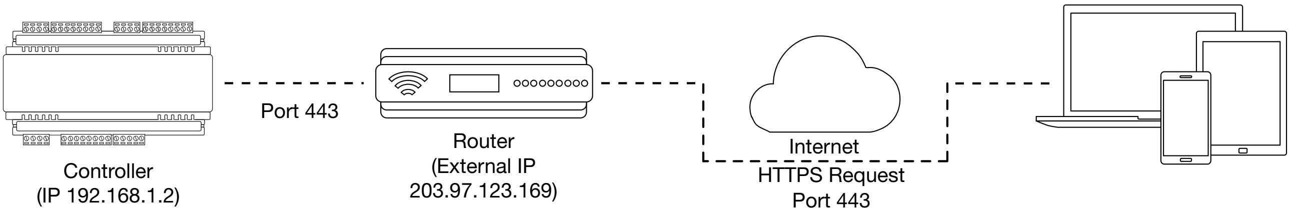
External access via HTTP is only required in order to validate and install your certificate. Once the certificate has been installed, HTTP access will be disabled because the more secure HTTPS connection is available. Therefore it will no longer be necessary to forward external port 80 to the controller.

Port forwarding is configured from the router's utility interface, which can be accessed by browsing to the router's IP address. Different routers have different interfaces, so it is recommended that you consult the documentation for your router.

Optional Port Forwarding

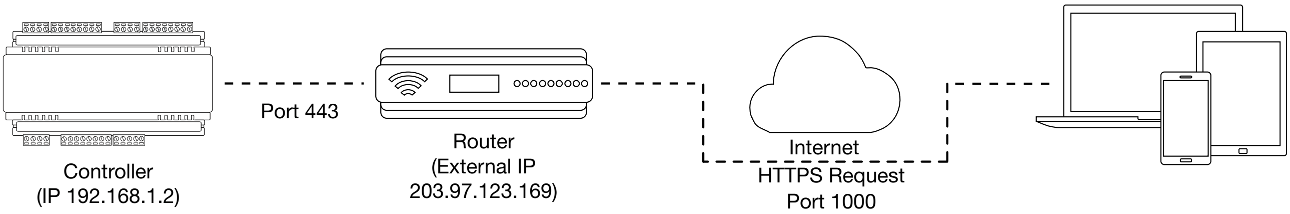
After you have installed a certificate and established an HTTPS connection to the controller, you may wish to continue accessing the controller over the internet. To achieve this, the controller must be accessible via its HTTPS port. The default HTTPS port is **internal port 443**, but this can be changed if necessary in the **System Settings** (available once **Use HTTPS** is enabled).

The easiest method is to configure the router to forward all traffic from **external port 443** (the default HTTPS port) to the controller's internal HTTPS port, as in the image below.



In this case, all traffic directed to the external HTTPS IP address will be forwarded to the controller. The controller's web interface could be accessed by typing https://203.97.123.169 into an external web browser.

However, it is possible to grant external access by forwarding any external port to the controller's HTTPS port. This is especially useful if external port 443 is not available on your network.



In this case, any traffic directed to **external port 1000** will be forwarded to the controller's HTTPS port. The controller's web interface can be accessed simply by appending the external port number onto the end of the URL: e.g. https://203.97.123.169:1000.

Note: If the controller does not have a factory loaded certificate, it will not be accessible via HTTPS until an HTTPS certificate has been installed, regardless of whether port forwarding has been configured.

Controller Default Gateway

In order for the controller to send and receive external communications via the router, its default gateway needs to be set to the router's **internal** IP address.

1. Log in to the controller's web interface.
2. Navigate to the **System Settings | Adaptor - Onboard Ethernet** tab.
3. In the **Default Gateway** field, enter the IP address of the router.
4. **Save** the configuration and **Restart** the controller.

Note: The default gateway must be set to the router's internal IP address that identifies it on the local internal network, not the external IP address used to connect over the internet.

Mapping an IP Address to a Domain

In order to achieve third-party HTTPS certification, it is necessary to map the controller's externally accessible IP address to a domain. The domain name becomes the **hostname** for the controller: a fixed, human readable point of access to the device.

Domain names can be purchased from Domain Name Registrars and assigned to a **static IP address**, usually for an annual fee. For example, the IP address 203.97.123.169 could be assigned the domain name controller.com, and would then be accessible by typing that domain name into a browser address bar.

However, typically routers are assigned a **dynamic IP address**. This IP address is not static: internet service providers may reassign the address whenever the router is reset or even more frequently. A fixed domain name would have to be constantly monitored and updated, as the IP address it is mapped to will change unpredictably. If necessary, a **static IP address** may be purchased from your internet service provider.

Alternatively, you may use a **Dynamic Domain Name Server (DDNS)**, which allows a dynamic IP address to be mapped to a static domain name. Generally a DDNS service will provide a client application which runs on the web server PC and automatically updates the domain's IP address mapping whenever the external IP address changes. Controllers also have an **integrated DDNS client** which supports several free DDNS providers.

Setting Up Integrated DDNS

DDNS (Dynamic Domain Name Server) is a method which allows you to create a static hostname even when the external IP address of the controller is not fixed. The controller contains an integrated DDNS client which automatically updates the DDNS provider whenever the IP address changes.

Controllers currently support two DDNS providers: Duck DNS (free provider) and No-IP (free accounts available, paid plans for further services).

In order to set up DDNS, the controller must be port forwarded so that it is externally accessible.

Setting Up Duck DNS

Duck DNS can be used for HTTPS certification via third-party certificates.

1. Browse to Duck DNS and create a free account by signing in with Google or another existing account.

   Take note of the **Token** that is generated when you create your account.
2. Create a new **subdomain**. The full hostname will have the form [subdomain].duckdns.org.
3. The **Current IP** field should automatically populate with the external IP address of your network. Ensure that this is the controller's externally accessible IP address.
4. Access the controller's web interface by typing its **IP address** into the address bar of a web browser, then log in with your username and password.
5. Navigate to the **System Settings**.
6. In the **Adaptor - Onboard Ethernet** tab, select the **Enable DDNS** checkbox.
7. Enter the **Hostname** [subdomain].duckdns.org and **DDNS Server** duckdns.org.
8. Leave the **DDNS Username** blank. For the **DDNS Password**, enter the **Token** generated by your Duck DNS account.
9. **Save** your settings.
10. Confirm that the controller is externally accessible by browsing to the hostname on another PC.

    If the controller's external port is not the default port, you will need to append the port number to the URL (e.g. controller.duckdns.org:1000).

Setting Up No-IP

The free No-IP Dynamic DNS service does not support third-party certification. This is only supported with the additional Plus Managed DNS service.

1. Browse to No-IP and create a **Dynamic DNS** account (free or paid as required).

   Free Dynamic DNS hostnames provided by No-IP require confirmation every 30 days, whereas paid accounts do not.
2. Create a new **Hostname** and select a **Domain**.
3. Ensure that the **IP Address** matches the controller's externally accessible IP address.
4. Access the controller's web interface by typing its **IP address** into the address bar of a web browser, then log in with your username and password.
5. Navigate to the **System Settings**.

6. In the **Adaptor - Onboard Ethernet** tab, select the **Enable DDNS** checkbox.
7. Enter the **Hostname** and **DDNS Server**.
8. Enter the **Username** and **Password** that you used to sign up to No-IP.
9. **Save** your settings.
10. Confirm that the controller is externally accessible by browsing to the hostname on another PC.

    If the controller's external port is not the default port, you will need to append the port number to the URL (e.g. controller.ddns.org:1000).

Configuring the HTTPS Connection

HTTPS uses certificates to encrypt the data channels between a website and the browser, ensuring that others on the internet cannot read the communications. When a connection is requested the controller's web page will "handshake" with the browser and exchange the encryption keys contained in the certificates. When both ends have exchanged the appropriate data and everything matches up, the channel is encrypted and normal data exchange can begin.

Establishing an HTTPS connection therefore requires a certificate to be installed on the controller. This certificate can be generated by either a trusted certificate authority (third-party certificate) or the user themselves (self-signed certificate).

This section outlines the methods of HTTPS connection which can be used and how to set up each method within your controller's web interface:

- Third-Party Certificate (see below)
- Self-Signed Certificate (see page 13)

Once HTTPS is successfully configured, the normal HTTP connection will be disabled.

Third-Party Certificate

This method uses a certificate generated by a recognized third-party certificate authority (CA) to encrypt the HTTPS connection. Unlike the self-signed certificate method, third-party certificates generally require an annual fee; however, they are trusted by web browsers.

The process has five main stages:

1. The installer generates a private/public encryption key pair and certificate signing request for their domain.
2. The installer submits the certificate signing request to the certificate authority.
3. The certificate authority provides a validation file which is loaded onto the controller.
4. The certificate authority validates the domain and provides the certificate.
5. Finally, the installer converts the certificate format (if necessary) and installs the certificate onto the controller.

Requirements for Third-Party Certificates

- The controller must be exposed to the internet via external port 80.
- The controller must be externally accessible via a hostname.

  Either static IP or DDNS (see page 8) can be used to assign this hostname.
- The operator must renew the certificate whenever it expires.
- Different certificate authorities may have different requirements. For example, some CAs do not require manual validation of domain names, allowing you to skip the certificate authentication stage. It is recommended that you carefully note all requirements for your chosen CA before beginning.

If you need help when obtaining and loading a third-party certificate, consult your IT support. ICT Technical Support cannot assist with this process.

Creating a Private Key and Certificate Signing Request

To begin, it is necessary to generate the private/public encryption key pair which will be the basis for the HTTPS encryption. The public key will be integrated into a certificate signing request which will be submitted to the CA.

The following instructions will use the free OpenSSL utility. The latest version of OpenSSL for Windows can be downloaded from this page.

1. Download and install the OpenSSL utility.
2. Navigate to the installation directory, open the **bin** folder, locate the **openssl** executable and run it as an administrator. This will open the OpenSSL command prompt.
3. To **generate the key pair**, enter the following command, replacing `[name]` with your desired filenames:

   ```
   req -newkey rsa:2048 -keyout [name].key -out [name].csr
   ```

   This generates a new 2048-bit private key (.key file) and certificate signing request (.csr file). The files should appear in the current OpenSSL directory.
4. Enter a **passphrase** for the private key. This is a phrase used to encrypt the private key to protect it against anyone with access to your local system. It will be required whenever the private key is used.

   Note that passphrase characters will not be displayed in the console. Only alphanumeric characters are supported for the passphrase.
5. Enter your **location and identity information** as requested. These details will be incorporated into your certificate and publicly viewable from the web browser.

   Ensure that the **Common Name** is the same as the **Domain Name** which is being used for the controller.

   Some details are optional. Confirm with your CA which fields are required.
6. **Save** both files in a safe, known location, as both are required for the following steps. It is especially important that the private key is not publicly accessible.

Purchasing a Certificate

Below are very basic instructions for purchasing a third-party certificate from a CA. Every CA will have different processes and requirements - this is only intended to be a rough guide to what is required for implementation on a controller.

1. Begin the process of generating a certificate from a recognized CA such as:
   - **GoDaddy**: https://nz.godaddy.com/web-security/ssl-certificate
   - **Network Solutions**: https://www.networksolutions.com/
   - **RapidSSL**: https://www.rapidsslonline.com/

   It is important that you select **File-Based or HTTP-based Validation** (or equivalent) when asked to choose an authentication/validation method. You will require a .txt file to upload to the controller.
2. When prompted, upload the text of your **Certificate Signing Request** (.csr).
3. Follow the CA's instructions to complete the request. You should be prompted to download a **.txt** validation file.

   DO NOT change the name or contents of this file.

Authenticating the Certificate

The .txt file that you received in the previous steps must be uploaded to a known directory on your domain (in this case, the controller) so that it can be viewed by the CA. This verifies that you are the owner of the domain in question.

1. Access the controller's web interface by typing its **IP address** into the address bar of a web browser, then log in with your username and password.
2. Navigate to the **System Settings**.
3. In the **General** tab, select the **Use HTTPS** checkbox (if not already enabled).
4. Enter an appropriate **HTTPS Port**. The default is port 443, which is commonly used for this purpose. You should retain the default port unless you are required to use another port by your system administrator.
5. Click **Load Validation File** and browse to the .txt validation file to load it onto the controller.

6. Open the **Adaptor - Onboard Ethernet** tab. Enter the controller's domain name in the **Controller Hostname** field.
7. Confirm that the file is publicly accessible by using another machine to navigate to [domainname]/.wellknown/pki-validation/[filename].txt. You should be able to view the content of your validation file.

Once the CA has verified that your domain is accessible, you will be sent the signed certificate. Wait times can vary between providers, but will typically take from one hour to several hours.

Converting the Certificate Format

The controller requires a file with the .pfx extension. Your CA may have provided a different file type, potentially several files such as a certificate (e.g. .cer, .crt or .pem) and an intermediate certificate. These must be combined with the private key generated with your certificate request to create a .pfx file. The following instructions will use the OpenSSL utility installed above.

1. Navigate to the installation directory, open the **bin** folder, locate the **openssl** executable and run it as an administrator. This will open the OpenSSL command prompt.
2. **Export** your certificate as a .pfx file using the following command, replacing `[name]` with your filenames:

   ```
   pkcs12 -export -certpbe PBE-SHA1-3DES -keypbe PBE-SHA1-3DES -nomac -out
   [name].pfx -inkey [name].key -in [name].[cer/crt/pem]
   ```

   Replace `[cer/crt/pem]` with the extension on your certificate file as required.

   > Always include the `-certpbe`, `-keypbe` and `-nomac` arguments so that the certificate is encrypted in a way that the controller can interpret. This does not affect the encryption of the HTTPS connection.

   Note: If you have been provided with an intermediate certificate you **must** include intermediate certificates by appending to the end of the command: `-certfile [intermediatename].[cer/crt/pem]` as shown below.

   ```
   pkcs12 -export -certpbe PBE-SHA1-3DES -keypbe PBE-SHA1-3DES -nomac -out
   [name].pfx -inkey [name].key -in [name].[cer/crt/pem] -certfile
   [intermediatename].[cer/crt/pem]
   ```

   > Android devices will fail to connect if intermediate certificates are not included in the certificate loaded onto the device.

3. Enter the **passphrase** for the private key (set above) to continue.

   > Note that passphrase characters will not be displayed in the console.

4. Enter an **export password** when requested. This will be required when installing the certificate on the controller.
5. This process will generate a [name].pfx file in the current OpenSSL directory. This is your third-party certificate. Store this file in a safe, known location.

Installing the Certificate on the Controller

1. Log in to the controller's web interface and navigate to the **System Settings**.
2. Scroll to the **Certificate File** section. Click **Install Certificate** and browse to the .pfx certificate file to install it on the controller.
3. Enter the **export password** that you created when generating the certificate file.
4. Click **Save**, then **restart the controller** using the button on the top right to implement the new settings.

   > Once the restart process is complete, the controller will restart but the web page will not automatically refresh.

5. Browse to the controller web page by adding the prefix https:// to the beginning of the IP address or URL.

A lock or similar icon in the browser toolbar should indicate that the connection is secure. Click on this icon to see details about the certificate, including the information you entered in the certificate signing request.

Self-Signed Certificate

Self-signed certificates do not require the certificate to be validated by an authority, or for the controller to be accessible over the internet. They can also be created for free. However, self-signed certificates are not considered secure by web browsers, which will generate warnings whenever the web interface is accessed. This method is fine for testing and development but is **not recommended** for live sites.

Requirements for Self-Signed Certificates

- There is no requirement for the controller to be externally accessible.
- The operator must manually renew the certificate whenever it expires.

Generating a Self-Signed Certificate with OpenSSL

The following instructions will use the free OpenSSL utility. The latest version of OpenSSL for Windows can be downloaded from this page.

1. Download and install the OpenSSL utility.
2. Navigate to the installation directory, open the **bin** folder, locate the **openssl** executable and run it as an administrator. This will open the OpenSSL command prompt.
3. To **generate** your certificate, enter the following command:

   ```
   req -new -newkey rsa:2048 -x509 -sha256 -subj "/C=[Country code]/CN=
   [Common name]" -days 365 -out [name].crt -keyout [name].key
   ```

   - Replace `[name]` with your desired filenames
   - The country code is optional, but recommended best practice. You can find your country code here.
   - The common name is typically in the form [hostname].[domain name]. For a self-signed certificate this does not need to be an externally accessible hostname. For example, you could use secure.controller.com.

   This generates a new key pair (.crt certificate and .key private key) with 2048-bit encryption that will expire after 365 days. The files should appear in the current OpenSSL directory.
4. Enter a **passphrase** for the private key. This is a phrase used to encrypt the private key to protect it against anyone with access to your local system. It will be required whenever the private key is used.

   Note that passphrase characters will not be displayed in the console. Only alphanumeric characters are supported for the passphrase.
5. Enter your **location and identity information** as requested. These details will be incorporated into your certificate and publicly viewable from the web browser.

   Ensure that the **Common Name** is the same as the **Domain Name** which is being used for the controller, if any.
6. To **export** your certificate, enter the following command, replacing `[name]` with your desired filename:

   ```
   pkcs12 -export -certpbe PBE-SHA1-3DES -keypbe PBE-SHA1-3DES -nomac -out
   [name].pfx -inkey [name].key -in [name].crt
   ```

   Always include the `-certpbe`, `-keypbe` and `-nomac` arguments so that the certificate is encrypted in a way that the controller can interpret. This does not affect the encryption of the HTTPS connection.
7. Enter the **passphrase** assigned above when prompted.
8. Create an **export password** when prompted. This will be required when installing the certificate on the controller.

   This process will generate a [name].pfx file in the current OpenSSL directory. This is your self-signed certificate. Store this file in a safe, known location.

Installing the Self-Signed Certificate to the Controller

1. Access the controller's web interface by typing its **IP address** into the address bar of a web browser, then log in with your username and password.
2. Navigate to the **System Settings**.
3. In the **General** tab, select the **Use HTTPS** checkbox (if not already enabled).
4. Enter an appropriate **HTTPS Port**. The default is port 443, which is commonly used for this purpose. You should retain the default port unless you are required to use another port by your system administrator.
5. Click **Install Certificate** and browse to the .pfx certificate file to install it on the controller.

   No .txt validation file is required for this method, as the connection is not validated by a third party.

6. Enter the **export password** that you created when generating the certificate file.
7. Click **Save**, then **restart the controller** using the button on the top right to implement the new settings.

   Once the restart process is complete, the controller will restart but the web page will not automatically refresh.

8. Browse to the controller web page by adding the prefix https:// to the beginning of the IP address or URL.

When using a self-signed certificate, you will likely be presented with a security warning if you attempt to access the HTTPS web page. The connection is still encrypted, but the browser has flagged the certificate as untrustworthy as it lacks third-party validation.

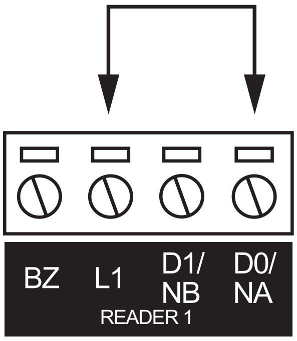
Disabling HTTPS

In some cases you may need to disable HTTPS to gain access to the controller's web interface. For example, this may be necessary if the controller is using an older, unsupported version of HTTPS which is blocked by the web browser.

In this situation you can temporarily default the controller's IP settings. This will revert the controller to HTTP operation until it is next power cycled, allowing you to access the controller and, if required, disable HTTPS.

Defaulting the IP Address of a Two Door Controller

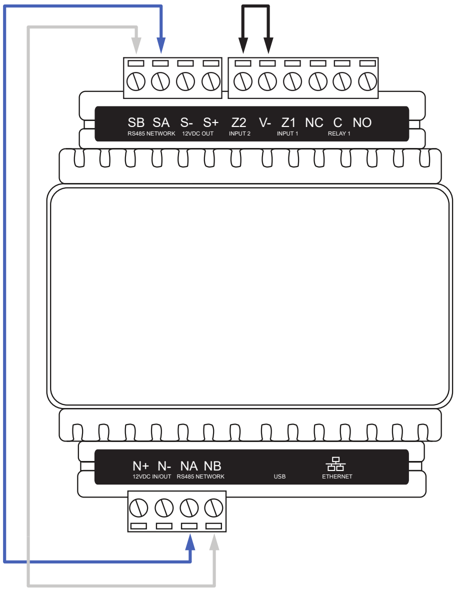
1. Remove power to the controller by disconnecting the 12V DC input.
2. Wait until the power indicator is off.
3. Connect a wire link between **Reader 1** D0 input and **Reader 1** L1 output.



4. Power up the controller. Wait for the status indicator to begin flashing steadily.

Defaulting the IP Address of a Single Door Controller

1. Remove power to the controller by disconnecting the 12V DC input.
2. Wait until the power indicator is off.
3. Connect a wire link between **NA** of the module network and **SA** of the reader network, and between **NB** of the module network and **SB** of the reader network.
4. Connect **Input 2** to ground.



5. Power up the controller. Wait for the status indicator to begin flashing steadily.

Accessing the Controller

5. When the controller starts up it will use the following temporary settings:
   - **IP Address**: 192.168.111.222
   - **Subnet Mask**: 255.255.255.0
   - **Gateway**: 192.168.111.254
   - **DHCP**: Disabled
   - **Use HTTPS**: Disabled
6. Connect to the controller by entering http://192.168.111.222 into the address bar of your web browser, and view or change the IP address and other network settings as required.

   > Remember to change the subnet of your PC or laptop to match the subnet of the controller.

7. Remove the wire link(s) and power cycle the controller again.

   The controller will now use the configured network settings.

Designers & manufacturers of integrated electronic access control, security and automation products.
Designed & manufactured by Integrated Control Technology Ltd.

Disclaimer: Whilst every effort has been made to ensure accuracy in the representation of this product, neither Integrated Control Technology Ltd nor its employees shall be liable under any circumstances to any party in respect of decisions or actions they may make as a result of using this information. In accordance with the ICT policy of enhanced development, design and specifications are subject to change without notice.