



## User Guide

### Babel Buster 3

**Model BB3-6101-GW**  
**Model MX-61-GW**

**Modbus TCP to RTU**  
**Transparent Gateway**  
**Rev. 1.0 – April 2021**

© 2021 Control Solutions, Inc.

### User Guide Contents

#### [1 Introduction](#)

- 1.1 How to Use This Guide
- 1.2 Important Safety Notice
- 1.3 Warranty

#### [2 Connecting Gateway for the First Time](#)

- 2.1 Where to Start
- 2.2 Overview of Model BB3-6101-GW/MX-61-GW
  - 2.2.1 Application of the BB3-6101-GW/MX-61-GW
  - 2.2.2 How RTU Translates to TCP or Vice Versa
- 2.3 What is New in Model BB3-6101/MX-61
- 2.4 Connectors and Indicators
- 2.5 Opening the Web User Interface

#### [3 System Configuration and Resources](#)

- 3.1 Using the File Manager
  - 3.1.1 Load, Save, Create XML Configuration File
  - 3.1.2 Select Startup Configuration
  - 3.1.3 Delete a File
  - 3.1.4 Clear Configuration
- 3.2 Configuration Files and Restoring Default Settings
- 3.3 Network Configuration
  - 3.3.1 IPv4, IPv6 Settings
  - 3.3.2 NTP Time Server Settings
  - 3.3.3 Port Settings
- 3.4 Resource Allocation
- 3.5 User Login Passwords

#### [4 Accessing RTU Devices from TCP](#)

- 4.1 Set Mode and Port Parameters
- 4.2 How It Works

#### [5 Accessing TCP Devices from RTU](#)

- 5.1 Set Mode and Port Parameters
- 5.2 Create TCP Device Map
- 5.3 How It Works

#### [6 Error Counts and Packet Log](#)

- 6.1 Reviewing Error Counts
- 6.2 Reviewing TCP Device Status
- 6.3 Reviewing the Packet Log

#### [Appendix A Hardware Details](#)

- A.1 Wiring
- A.2 Front Panel LED Indicators
- A.3 RS-485 Line Termination and Bias
- A.4 Soft Configuration Reset
- A.5 Discovering Lost IP Address
- A.6 Forced Hard Configuration Reset
- A.7 Firmware Update Notes

#### [Appendix B Modbus Reference Information](#)

- B.1 Function Codes, Error Codes, and More

#### [Appendix C Trouble Shooting](#)

- C.1 Modbus RTU Trouble Shooting
- C.2 Modbus TCP Trouble Shooting

- C.3 Wireshark Hardware Requirements
- C.4 Example of Using Wireshark

#### [Appendix D SSL Certificates for Secure Web \(HTTPS\)](#)

- D.1 X.509 Auto-Certificate Generation
- D.2 External Certificates
- D.3 Certificate Generation Script (Linux)



# 1. Introduction

## 1.1 How to Use This Guide

This user guide provides background information on how the gateway works, and an overview of the configuration process. There are several sections for groups of tabs found in the web interface in the gateway which is accessed by opening a web browser and browsing to the IP address of the device.

You should at least read Sections 2 and 3, and other sections specific to your intended use. There is a "Quick Help" section at the bottom of each web page in the gateway which is generally sufficient for quick reference in setting up the gateway.

## 1.2 Important Safety Notice

**Proper system design is required for reliable and safe operation of distributed control systems incorporating any Control Solutions product. It is extremely important for the user and system designer to consider the effects of loss of power, loss of communications, and failure of components in the design of any monitoring or control application. This is especially important where the potential for property damage, personal injury, or loss of life may exist. By using ANY Control Solutions, Inc., product, the user has agreed to assume all risk and responsibility for proper system design as well as any consequence for improper system design.**

## 1.3 Warranty

**This documentation is provided "as is,"** without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Control Solutions may make improvements and/or changes in this documentation at any time. This documentation could include technical inaccuracies, typographical errors, and the like. Changes are periodically made to the information herein; these changes may be made without notice.

**Product Warranty:** All Control Solutions products are warranted against defects in materials and workmanship for a period of time from date of shipment from factory as follows: Two years on non-mechanical parts, one year on mechanical parts (e.g. relays). Defective units will be repaired or replaced, at manufacturer's discretion, at no cost to user except when negligence or improper use has resulted in damage. The express warranty stated herein is in lieu of all other warranties, express or implied,

including without limitation any warranties of merchantability or fitness for a particular purpose and all other warranties are hereby disclaimed and excluded by Control Solutions, Inc.

Configuration errors made by customer are not covered under warranty. Damage caused by incorrect electrical connection is not covered under warranty. Removing circuit boards from their enclosures will void the warranty - the complete product with all of its original circuit boards and components must be returned for warranty consideration.



## 2. Connecting Gateway for the First Time

### 2.1 Where to Start

The Babel Buster BB3-6101-GW or MX-61-GW is used to directly route Modbus RTU messages to Modbus TCP and vice versa. This version of gateway does not do any data translation or remapping of registers, nor does it even look at the data being transferred. If you are looking for a gateway to translate Modbus to some other protocol, or need to remap Modbus registers between TCP and RTU, then you need to start by looking for a different model number.

NOTE: There are several versions of BB3-6101, most of which are SNMP to Modbus gateways. It should be noted that SNMP is disabled in the BB3-6101-GW because there is no means of directly routing Modbus to SNMP without the mapping found in the other models of BB3-6101.

Start by getting familiar with this User Guide. Be sure to review the remainder of this section. Online videos are also available to demonstrate key operations in setting up the BB3-6101-GW or MX-61-GW.

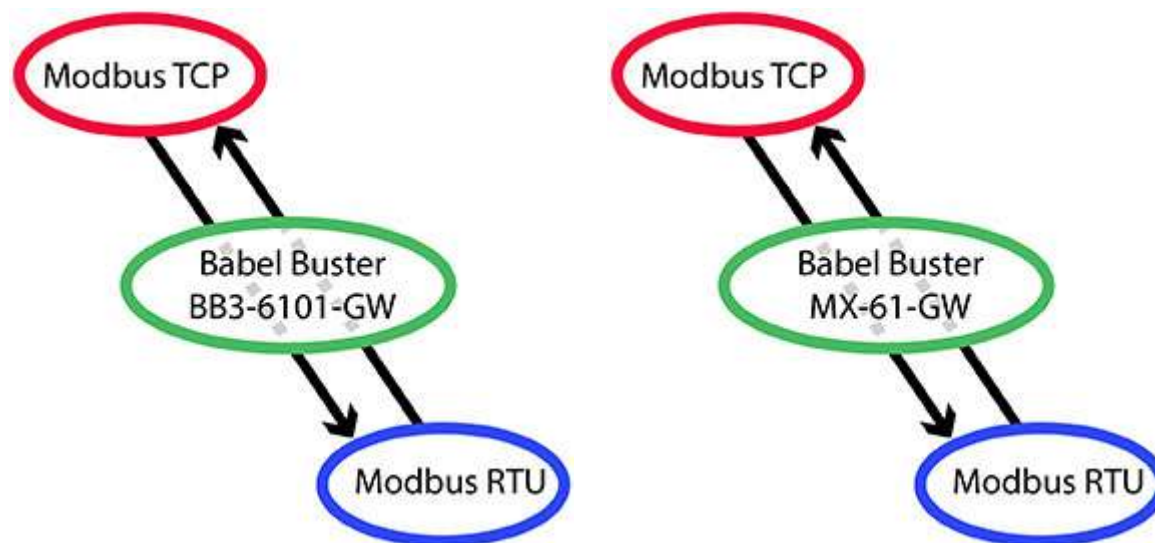
If you get stuck, you can open a support ticket at <https://ticket.csimn.com> where response time is generally 24 hours or less, and often as little as 2 hours, and at no cost.

NOTE: Screen shots throughout this User Guide illustrate BB3-6101-GW; however, the screens in the MX-61-GW are identical with the only exception being model number indicated at the top of the page.

### 2.2 Overview of Model BB3-6101-GW/MX-61-GW

#### 2.2.1 Application of the BB3-6101-GW/MX-61-GW

The Babel Buster BB3-6101-GW/MX-61-GW is a non-mapping gateway used to directly route Modbus RTU messages to Modbus TCP and vice versa. This version of gateway does not do any data translation or remapping of registers, nor does it even look at the data being transferred.



Your requirement can be categorized as one or the other of these two options:

- Accessing RTU devices from TCP
- Accessing TCP devices from RTU

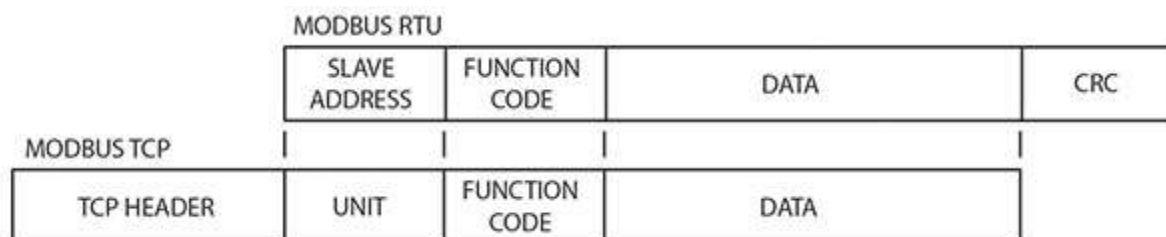
To access RTU devices from TCP, you need only set the IP address of the gateway, and set the RTU port parameters such as baud rate. Refer to section 4 in this user guide for instructions if you will be accessing RTU devices from TCP.

To access TCP devices from RTU, in addition to setting the IP address of the gateway itself and setting the RTU port parameters, you will need to create a table that maps IP addresses of the TCP devices to an RTU addresses. This becomes a lookup table that the gateway uses to forward RTU requests to the correct TCP device. Refer to section 5 in this user guide for instructions if you will be accessing TCP devices from RTU.

### 2.2.2 How RTU Translates to TCP or Vice Versa

The Babel Buster BB3-6101-GW and MX-61-GW are non-mapping Modbus gateways used to simply forward Modbus RTU requests and responses to Modbus TCP, and vice versa. Most Control Solutions gateways involve mapping, and the gateway itself contains registers or objects which hold copies of data found in other devices. This intermediate data buffering is what allows access to the same data from multiple protocols. The non-mapping gateway discussed here does not contain any of its own registers. It simply forwards whatever request it receives to the other side by simply repackaging and retransmitting exactly the same request (regardless of whether it was a correct request).

The process of "repackaging" the Modbus request or response is illustrated below. The core of a Modbus data packet is the same for RTU and TCP. It contains a slave address (or unit number), a function code, and some data. The "data" is most often a starting register number, register count, and register data (if writing).



If the data packet is being sent via Modbus RTU, the first character transmitted is the slave address, and the last two characters are a CRC type checksum. If the data packet is being sent via Modbus TCP, there is a TCP header at the beginning of the packet, and the last byte of that packet is the same slave address or unit number that would have been sent via RTU. The RTU checksum is not included because Ethernet has its own checksum that covers the entire Ethernet transmission.

The process of translating RTU to TCP or vice versa is simply a matter of adding or subtracting TCP header and RTU checksum. The only configuration required in this type of gateway is to create an association between RTU slave addresses and TCP IP addresses.

## 2.3 What is New in Model BB3-6101/MX-61

The BB3-6101 is a significant enhancement over its predecessor, the BB2-6010. The MX-61 is the equivalent enhancement over its predecessor, the SPX. The hardware includes a faster processor and hardware encryption engine for efficient rendering of secure web pages. The software includes numerous enhancements.

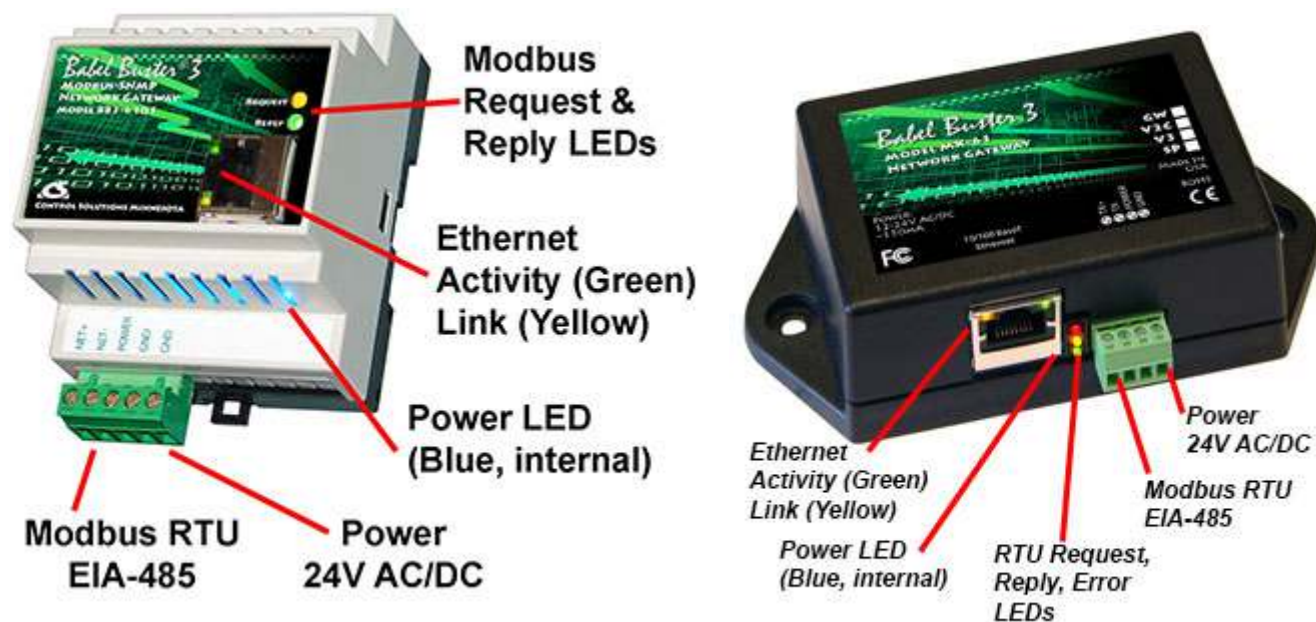
- IPv6 support
- Secure (HTTPS) web server
- Faster processor
- Added features specific to the application

## 2.4 Connectors and Indicators

Follow these steps to make the initial connection to the BB3-6101/MX-61.

(a) Connect power. Apply +12VDC to +24VDC or 24VAC to the terminal marked "POWER", and common or ground to one of the terminals marked "GND".





(b) Connect a CAT5 cable between the RJ-45 jack on the gateway, and your network switch or hub. You cannot connect directly to your PC unless you use a "crossover" cable (or your PC supports auto-MDX, which many newer laptops do).

(c) Apply power.

A blue LED inside the case should light indicating power is present.

If the link LED on the RJ45 jack is not on, check your Ethernet cable connections. Both link and activity LEDs on the RJ45 jack will be on solid for a short time during boot-up. The entire bootup process will take 1-2 minutes, during which time you will not be able to connect with a browser.

Ethernet link LED is the yellow LED integrated into the CAT5 connector. Ethernet activity LED is the green LED integrated into the CAT5 connector.

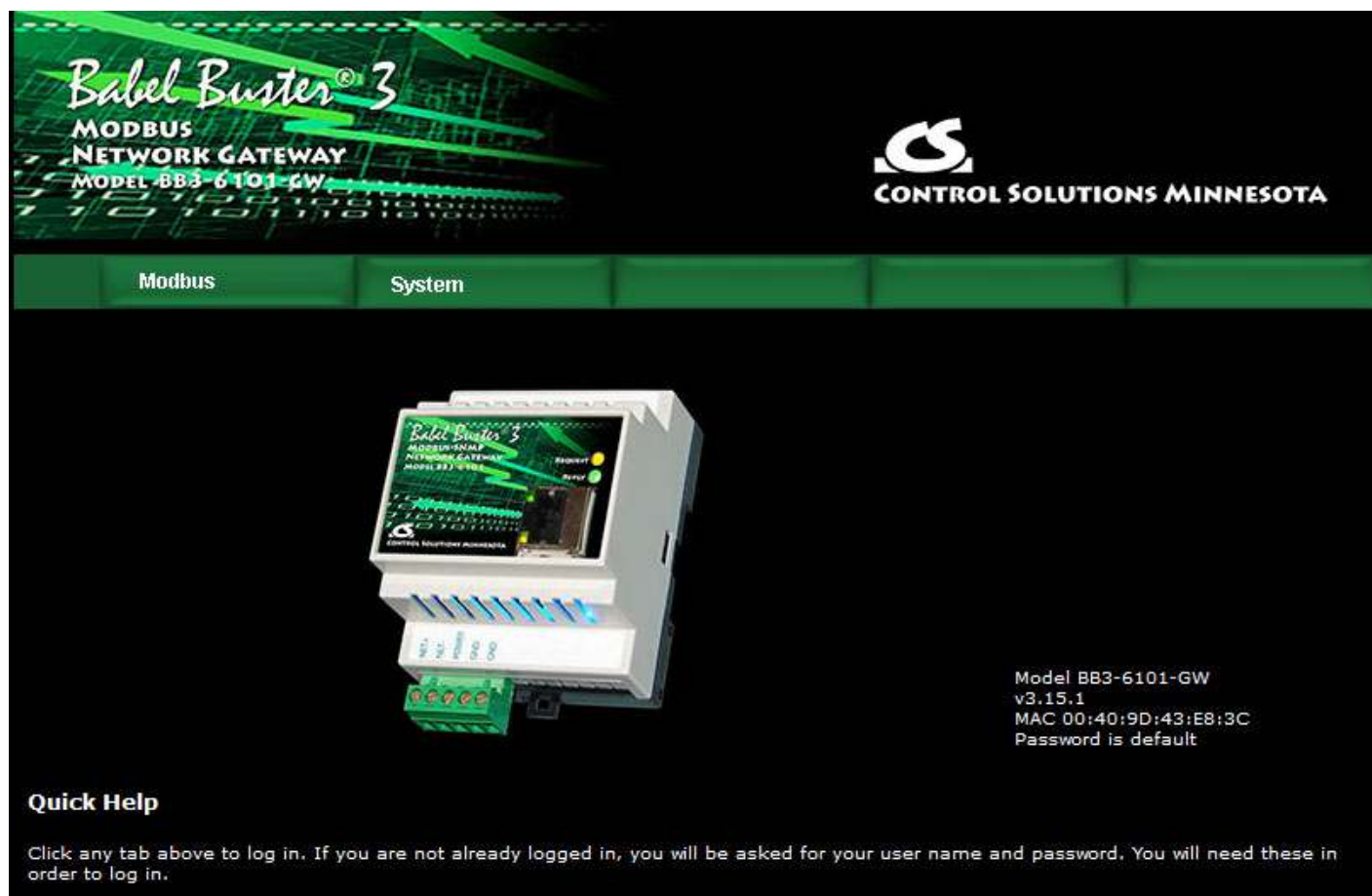
Refer to Appendix A for additional detail pertaining to connections and indicators as well as optional internal jumper settings.

## 2.5 Opening the Web User Interface

The default IP address as shipped is 10.0.0.101. Open your browser, and enter "http://10.0.0.101/" in the address window. You should see a page with the "Babel Buster 3" header shown below. From this point, you will find help on each page in the web site contained within the product.

If your PC is not already on the 10.0.0.0 domain, and you are unable to connect, you may need to temporarily change your computer's IP address to a static IP address that starts with 10.0.0. and ends with anything but 101.





When you click on any of the page tabs such as System, you will be asked for a user name and password. The only login as shipped is user name "root" with a unique password generated specifically for your Babel Buster. Your password should be included on a document included with the gateway, or on a label attached to the gateway.

If the unique automatically generated password is currently in effect for user "root", it will be indicated by "Password is default" as shown in the above screen shot. If you have changed the root password to something of your own making, then this line is absent.

There is no way to get the BB3-6101/MX-61 to show you what the default root password is. If you have lost track of it, make a note of the MAC address, and open a support ticket at <https://ticket.csinn.com> to request the default root password (you will need to provide the MAC address in order to obtain the password).

To change the IP address of the gateway, go to the Network page under System :: System Setup. The following page should appear (only top portion illustrated here). Change the IP address, and subnet mask and gateway if applicable. Click Change IP to save the changes. The process of programming this into Flash takes around half a minute. The new IP address only takes effect following the next system restart or power cycle.

**Babel Buster<sup>®</sup> 3**  
MODBUS  
NETWORK GATEWAY  
MODEL BB3-6101-GW

**CONTROL SOLUTIONS MINNESOTA**

Modbus System  
File Manager **Network** User

**IPv4 Settings** ☒ Automatic ☐ Static

IPv4 Static IP Address  IPv4 Configured IP Address **192.168.1.115**

IPv4 Static Subnet Mask  IPv4 Subnet Mask **255.255.255.0**

IPv4 Static Gateway  IPv4 Gateway **192.168.1.1**

**IPv6 Settings** ☐ Disabled ☐ Automatic ☒ Static

IPv6 Link-Local IP Address **fe80::240:9dff:fe43:e83c**

IPv6 Configured IP Address **fec0::9**

Most changes are stored in an XML configuration file in the device's Flash file system. Only a few are stored differently, and the IP address is one of those. Normally, clicking Update on any configuration page only stores that configuration information to a temporary RAM copy of the configuration file. To make your changes other than IP address permanent, you must select your file, select the Save XML Config File action, and then click Execute on the File Manager page. Refer to Section 3 for more about the File Manager.

NOTE: The BB3-6101-GW and MX-61-GW require rather minimal configuration compared to mapping gateways. In some cases, you might not need any configuration file at all. Refer to sections 4 and 5 as applicable for your need.

Modbus System  
**File Manager** Network User

Free space: 1.36 MB

File Directory:  Filtered by:

Selected File:  Action:

Boot configuration  ☐ Confirm



## 3. System Configuration and Resources

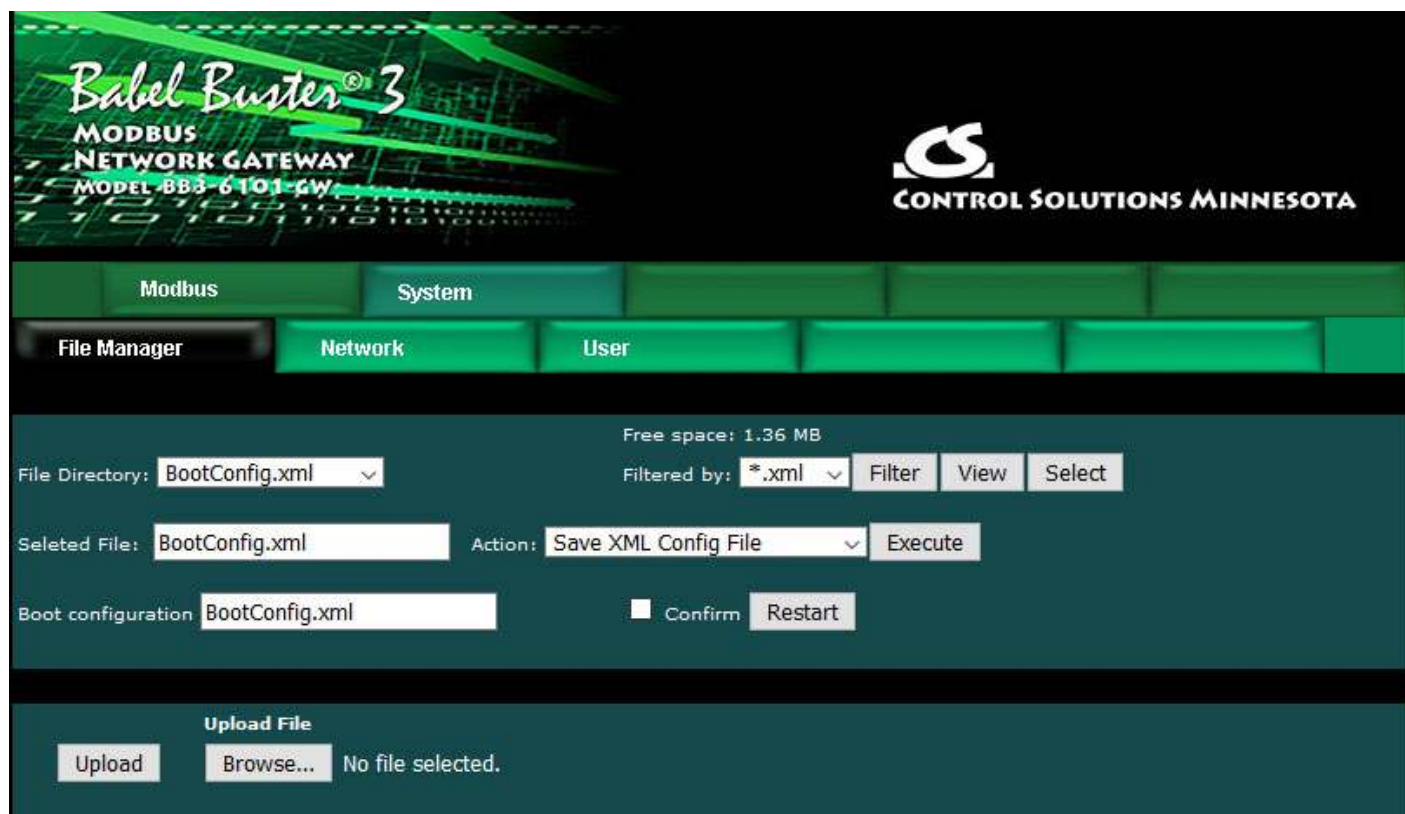
### 3.1 Using the File Manager

The File Manager page is probably one of the most important pages to know about. Among other things, this is where you tell the gateway to save all of the changes you have made. The various "Update" buttons on the many pages in the web user interface only copy your configuration from your PC's browser to temporary memory in the gateway. To retain those changes indefinitely (i.e. through restart or power cycle), you need to tell the gateway to save those changes in a configuration file.

The configuration files are stored in non-volatile (Flash) memory. The process of reprogramming the Flash takes a little time. It would be cumbersome to rewrite that file every time you made a minor change. Therefore, in the interest of being more responsive, and in the interest of extending the life of the Flash, configuration is only saved to Flash when you direct it to do so.

The File Manager is used primarily to manage your XML configuration files, but you can also upload SSL certificates here.

**NOTE:** Most Babel Buster gateways include a significant amount of configuration and the gateway has little use without the content of an XML file. The BB3-6101-GW and MX-61-GW are a rare exception depending on mode of operation. If using this gateway to access RTU devices from TCP, then no XML file is necessary. The only settings are mode and port settings (e.g. baud rate) and these settings are saved in non-volatile configuration memory separate from XML file storage. If you are accessing TCP devices from RTU, then you need to create a TCP Device Map, and this map must be saved in the XML configuration file.



The File Directory is a list of files that are currently stored in the Babel Buster's Flash file system. To filter files by type, select a type from the Filtered by list, and click Filter.



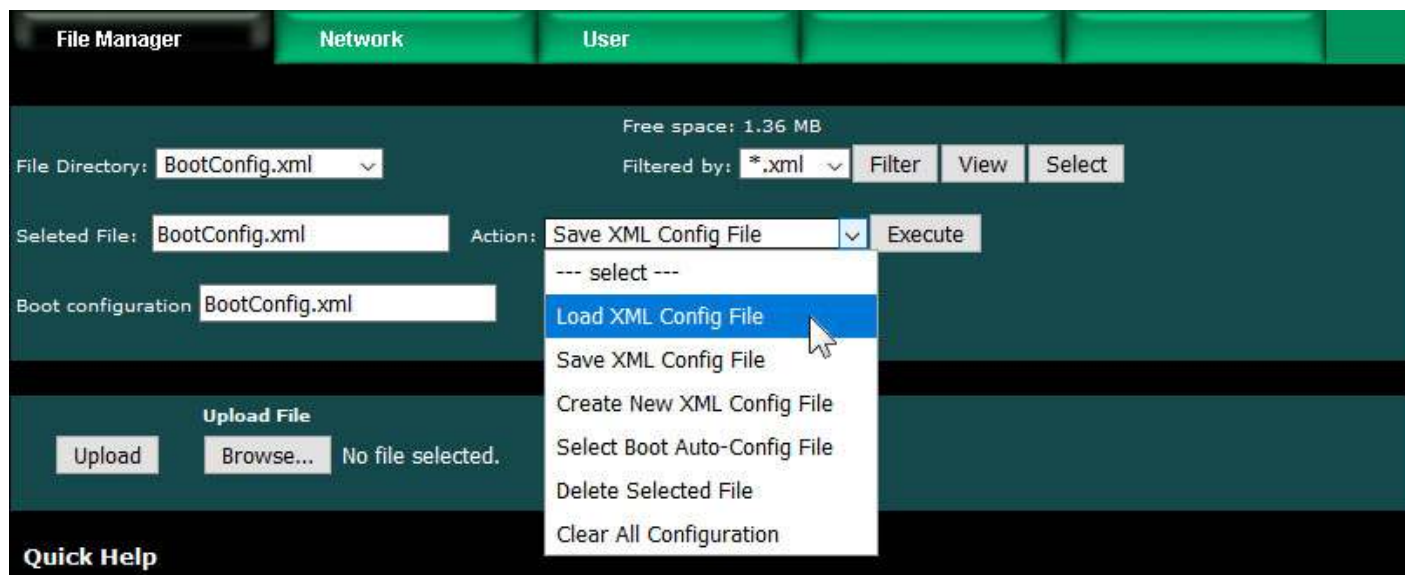
File type filters are as follows:

- \*.xml XML configuration files
- \*.pem SSL certificates
- \*.\* Display all files

There are several file related actions you may take. To take action with a certain file, select that file from the File Directory list, and click Select. That file should now show up in the Selected File window.

Once a file has been selected, choose your action from the Action list, and click Execute.





You must use the Select button to populate the Selected File window prior to executing any action from the list. Choose a file from the drop down list that shows all available files, then click the Select button. You may then act on that file.

You do not need to use the Select button to simply View a file. Clicking View will cause your browser to display the file chosen from the drop down list.

**Upload File:** To upload a file from your PC to this gateway, use the Browse button to find the file on your PC, open the file in the PC's file dialog box, and then click Upload.

**NOTE:** If you get a "File upload error: -1" message, click the browser's "back" button, then simply click the View button to view any file (does not matter which file), and then click browser's "back" button again to return to this page. This gets the browser and HTTP server back in sync, and this requirement generally happens only once following power-up.

**Restart:** To restart the gateway, check Confirm and click Restart. This is a hard reset that will accomplish the same thing as a power cycle without physically disconnecting and reconnecting power.

### 3.1.1 Load, Save, Create XML Configuration File

**NOTE:** Configuration files from a BB2-6010-GW or SPX-GW can be loaded into the BB3-6101-GW or MX-61-GW, but will be saved in a new format when saved.

**NOTE:** The only thing saved in the XML file for a BB3-6101-GW or MX-61-GW is the TCP Device Maps. RTU port settings are saved in non-volatile configuration memory in this model gateway. Therefore, if you are not using the TCP Device Mapping, you have no need for any XML files.

**Load XML Config File:** The configuration file shown in the "Boot configuration" window will be loaded automatically at startup. If you have uploaded a new configuration file and wish to use it without restarting, select that file (choose from list, click Select), select this action, and click Execute.

**Save XML Config File:** Any time you have made configuration changes that you want to retain as permanent, you need to come here, select the file from the directory list, and execute this Save action.

**Create New XML Config File:** You have the option to create a totally new configuration file. This is often suitable if you started with an existing configuration, made changes, and want to save your changes without replacing the original configuration. To create a new file, rather than selecting a file from the directory list, simply type a new name into the Selected file window. The name cannot contain spaces or special characters, and be sure to use the correct file suffix. Enter the name and execute this action.

### 3.1.2 Select Startup Configuration

**Select Boot Auto-Config File:** This is where you tell the Babel Buster what configuration to automatically load upon startup. To set the Boot configuration, select the XML file from the list, and execute this action. The name of the startup file, along with a few other important things like the gateway's own IP address, are stored in a different area of Flash that is not part of the file system.

When selecting a new Boot configuration file, it is a good idea to select the file, and execute Load XML Config File. If there are errors, they will be displayed. If there are errors in the file but you do not fix them, then the gateway will not fully start up the next time it restarts. The web user interface will be available, but it will not be talking to Modbus devices.

### 3.1.3 Delete a File

Remove a file from the Flash file system by selecting it from the list and executing the Delete Selected File action.

### 3.1.4 Clear Configuration

**Clear All Configuration:** Execute this action to completely wipe out all configuration. This includes all Modbus TCP Device Maps. This will put you back to a "reset to factory" condition with the exception that your IP address is left unchanged. (See Appendix A, Section A.6, regarding forced hard configuration reset that includes IP address and root password.) If you want to make the now empty configuration permanent, select the file that is also selected as Boot configuration, and execute the Save XML Config File action.

The other means of completely wiping out all saved configuration is to simply delete the file named as the Boot configuration file, and then restart or power cycle the Babel Buster. Upon restart, a new empty configuration file will be created automatically (provided the default file name BootConfig.xml is being used as the boot file - see next section).

## 3.2 Configuration Files and Restoring Default Settings

There is a means of restoring the Babel Buster to "manufacturer's default settings". First of all, make sure that the Boot configuration file is set to "BootConfig.xml". Then, after selecting this file as the boot file, delete it. Now restart the gateway. Upon restart, and upon finding that the boot configuration name is BootConfig.xml, and it does not exist, the gateway will automatically create one with default parameters. The automatic creation of a default file will not occur with any other file name.

**Manual Editing:** It is possible to manually edit the XML file outside of the gateway. However, doing so is very prone to errors. If there are errors in the XML file, it will not load successfully on startup. If the configuration does not load on startup, none of the scanners will begin scanning. Because they are all blocked by configuration failure, entering new configuration via the web pages will not result in functionality being restored. You must successfully load a configuration file before the gateway will become functional. To check for errors, select the file here, select Load XML Config File, and click Execute. Error messages that would have been discarded by the automatic loading at startup will now be displayed on an error page if there are any.

**Backup Copy of XML Config File:** To save a copy of the configuration to your PC, select the file and click the View button. Your browser will now display the XML file. DO NOT do a text copy/paste to try to create an XML file - doing so will result in an invalid file format that cannot be loaded again. You must use the browser's "save as" or "save page" function. The browser should default to wanting to save a file with a .xml suffix. If correctly saved on your PC, you should be able to double click on the saved file and it will result in opening the file automatically in your browser. It was saved correctly if the browser does not give any error messages when displaying the XML (which should now look exactly as it did when you first clicked the View button). Saving the configuration file to your PC, and then uploading on a different device, is a quick and easy way to configure two Babel Busters the same way.

**Note about caching:** Your browser may cache files. If you view a file, make configuration changes, save the file, then view the file again, you may see the old file cached by the browser. To see the updated file, go to "Options" in your browser's tools menu, and delete temporary Internet files (or delete cache files). Also, if you upload a file, make changes on your PC, and re-upload the same file, the browser may send the old file. Again, you will need to find the button inside your browser options that lets you delete the cached files from your PC. To upload a configuration file from your PC to the gateway, use the Browse button to find the file on your PC, open the file in the PC's file dialog box, and then click Upload.

### 3.3 Network Configuration

The Network Configuration page is where you set the Babel Buster's IP address as well as a few other important things.



**Babel Buster<sup>®</sup> 3**  
MODBUS  
NETWORK GATEWAY  
MODEL BB3-6101-MX

**CONTROL SOLUTIONS MINNESOTA**

Modbus System  
File Manager Network User

**IPv4 Settings** ☒ Automatic ☐ Static

IPv4 Static IP Address  IPv4 Configured IP Address **192.168.1.115**

IPv4 Static Subnet Mask  IPv4 Subnet Mask **255.255.255.0**

IPv4 Static Gateway  IPv4 Gateway **192.168.1.1**

**IPv6 Settings** ☐ Disabled ☐ Automatic ☒ Static

IPv6 Link-Local IP Address **fe80::240:9dff:fe43:e83c**

IPv6 Configured IP Address **fec0::9**

IPv6 Static IP Address

IPv6 Prefix Length

IPv6 Gateway Tunnel

**DNS Settings**

Primary DNS

Secondary DNS

### 3.3.1 IPv4, IPv6 Settings

To change the IP address(es) of this device, make the applicable entries and click Apply. The "automatic" selection means DHCP. Changes to the IPv4 IP address will take effect upon the next system restart.

If IPv6 is enabled, IPv6 will always have a Link-Local address, plus one configured address. The configured address will be either the static IP address, or an IPv6 address obtained from an IPv6 DHCP server. If no configured address appears, the DHCP server may have been unreachable.

The IPv6 static IP address window is the configured static address. If "Static" is selected and a new IP address entered as the static address, this new address will not take effect until the next system restart.

The numbers shown to the right of the IPv4 input windows are the actual numbers currently in use. If static IP addresses have been entered but the gateway has not been restarted yet, these numbers will not be the same.

The place to enter DNS Server IP addresses is provided; however, DNS is not used in this particular gateway. The DNS feature is simply part of the standard network configuration for Babel Buster 3 gateways in general.

### 3.3.2 NTP Time Server Settings

The Babel Buster maintains time and date via SNTP services.

Primary NTP Server	<input type="text" value="129.6.15.28"/>	Secondary NTP Server	<input type="text" value="132.163.97.1"/>
Daylight Time Start Rule	<input type="text" value="3.2.0/02:00:00"/>	Daylight Time End Rule	<input type="text" value="11.1.0/02:00:00"/>
Standard GMT Offset	<input type="text" value="-360"/> Minutes	Daylight GMT Offset	<input type="text" value="-300"/> Minutes
NTP Refresh Period	<input type="text" value="300"/> Minutes	<input type="button" value="Set NTP"/>	
Current Local Time		<b>2021-04-28 14:55:25</b> <input type="button" value="Refresh"/>	

NTP setup: Enter a primary and secondary IP address of NTP servers, such as those found at [www.nist.gov](http://www.nist.gov) (go to <http://tf.nist.gov/tf-cgi/servers.cgi> to find more). Enter daylight start/end rules, and offset from GMT for both standard and daylight time. Offset is a negative number in the western hemisphere. Enter an NTP update time in minutes. Do not set NTP to update too frequently or you risk being denied service by the NTP server. Click the Set NTP button after all settings have been made. The Flash update will take several seconds. The initial update of local time may take a minute or two.

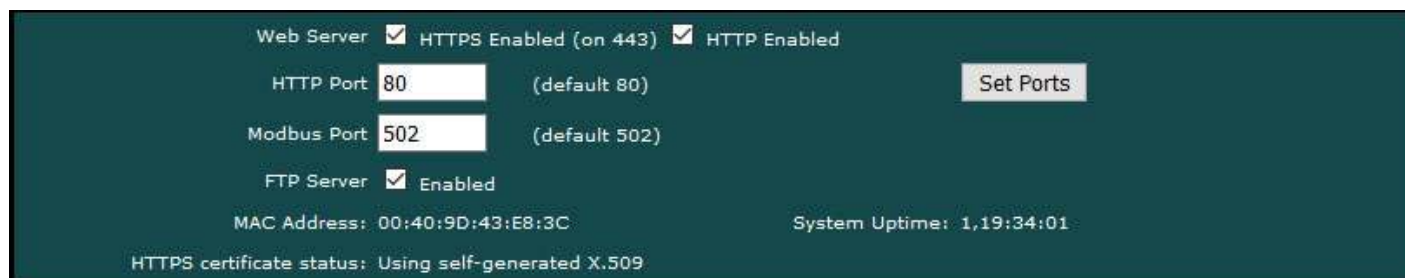
Daylight savings time start/end rules consist of "date/time" where the date (m.n.d) indicates the day when summer time starts or ends, and time (hour:min:sec) is the current local time when summer time starts/ends. The date portion of the rule is formatted as follows:

- m indicates the month ( $1 \leq m \leq 12$ )
- n indicates which week of the month ( $1 \leq n \leq 5$ ). 5 = the last week in the month.
- d indicates what day of the week ( $0 \leq d \leq 6$ ). 0 = Sunday

For example: Start "4.1.0/02:00:00", end "10.5.0/02:00:00" means summer time starts at 2am on the first Sunday in April and ends at 2am on last Sunday in October. That was the old US rule. The new US rule is start "3.2.0/02:00:00" and end "11.1.0/02:00:00", which is start at 2am on the second Sunday in March, end at 2am on the first Sunday in November.

Note about time maintained here: Modbus gateway functionality has no use for time and date. The only time you have a need for valid time and date is when using secure connections. If you are using a secure web connection and having trouble connecting, be sure NTP is set up here. If using the internally self-generated X.509 certificate for HTTPS, NTP is not needed.

### 3.3.3 Port Settings



The screenshot shows a configuration page for a Babel Buster 3 gateway. It features a dark teal background with white text and input fields. At the top, there are two checked checkboxes: 'Web Server' and 'HTTPS Enabled (on 443)'. Below these, the 'HTTP Port' is set to 80 (default 80) and the 'Modbus Port' is set to 502 (default 502). There is a 'Set Ports' button to the right of the port fields. The 'FTP Server' checkbox is also checked and labeled 'Enabled'. At the bottom, the 'MAC Address' is 00:40:9D:43:E8:3C and the 'System Uptime' is 1,19:34:01. A status line at the very bottom indicates 'HTTPS certificate status: Using self-generated X.509'.

Web Server	<input checked="" type="checkbox"/>	HTTPS Enabled (on 443)	<input checked="" type="checkbox"/>	HTTP Enabled
HTTP Port	<input type="text" value="80"/>	(default 80)	<button>Set Ports</button>	
Modbus Port	<input type="text" value="502"/>	(default 502)		
FTP Server	<input checked="" type="checkbox"/>	Enabled		
MAC Address:		00:40:9D:43:E8:3C	System Uptime: 1,19:34:01	
HTTPS certificate status: Using self-generated X.509				

Secure browsing can be enabled here, and non-secure can be disabled. You cannot disable both, and a forced configuration reset will restore HTTP (non-secure) web browsing. In order to use HTTPS, you must first upload the necessary SSL certificates (see Appendix E) or allow the certificates to be self-generated by explicitly deleting existing certificates.

**IMPORTANT:** It is highly recommended that in making the transition from HTTP to HTTPS, you enable both until you confirm HTTPS is functional. If there is a problem with the SSL certificates provided for HTTPS, then HTTPS will not run and you will find an error message on the "HTTPS certificate status" line. If you disable standard HTTP without first verifying that HTTPS is functional, you may end up locked out and will then need to do a forced hard reset (Appendix A.6).

The HTTP port for browsing the user interface can be moved away from the default HTTP port 80. Select a different port, click Set Ports, and then restart the gateway to make that new port take effect. Don't forget to append the port number to the gateway's IP address when attempting to browse the web user interface if it has been moved away from port 80.

The Modbus port to which this device responds as a Modbus TCP server is entered here. The standard port is 502. To enter a non-standard port number, enter that here and click Set Ports to set the Modbus port. The device needs to be restarted after changing the Modbus TCP port.

FTP is enabled by default to allow firmware update uploads. It may be optionally disabled here. Just remember to enable it again before attempting a firmware update.

Any changes to these port numbers or enabling/disabling features requires restarting the Babel Buster before they will take effect.

### 3.4 Resource Allocation

Most models of Babel Buster 3 gateways have a Resource Allocation page. Because the BB3-6101-GW requires minimal resources to function, it always has the maximum permitted resources already permanently allocated.

Modbus protocol specification permits RTU addresses (or unit numbers in TCP) to be from 1 to 247 and this full range is supported by the BB3-6101-GW.

Support for up to 120 simultaneous Modbus TCP connections (unique IP addresses) is provided by the BB3-6101-GW.

### 3.5 User Login Passwords

There is only one default login provided initially, namely the username "root" with a unique password generated specifically for your particular Babel Buster. This password is provided to you in either external documentation included with the gateway, or it may be found on a label attached to the gateway. Network security laws in some jurisdictions require that Internet connected (or connectable) devices be shipped with unique default passwords, and the BB3-6101/MX-61 complies with this requirement.

Additional user logins may be created. The privilege level Administrator lets that user see and change anything. The privilege level Maintenance allows the user to log in and see (and change) values in the local objects via the Local Objects page, but cannot access any other pages. The Restricted level has no meaning in the BB3-6101/MX-61 (other than block access to everything) since it does not operate as a user defined web server.

You also have the option of IP filtering. If set, then the user can only access Babel Buster's web pages from that IP address. Leave set to 0.0.0.0 to disable filtering.

User Name	Password	Privilege Level	IP Filter	Confirm Change
		Restricted	0.0.0.0	<input type="checkbox"/>
		Restricted	0.0.0.0	<input type="checkbox"/>
		Restricted	0.0.0.0	<input type="checkbox"/>
		Restricted	0.0.0.0	<input type="checkbox"/>
		Restricted	0.0.0.0	<input type="checkbox"/>
root	.....	Unrestricted	0.0.0.0	<input type="checkbox"/>
root confirm				





## 4. Accessing RTU Devices from TCP

### 4.1 Set Mode and Port Parameters

Accessing one or more Modbus RTU slaves from TCP with the external Modbus TCP device acting as client (master) means the gateway needs to act as an RTU master on the RTU network. Therefore, select "I act as the RTU master" for mode.

The screenshot shows the 'Modbus' configuration page of the Babel Buster 3 software. The page has a dark green header with the product name and logo. Below the header is a navigation bar with tabs: 'Modbus', 'System', 'Error Counts', 'Packet Log', and 'Update'. The 'Modbus' tab is selected, and the 'Mode / Port Info' sub-tab is active. The main content area is divided into two columns. The left column contains settings for 'Baud Rate' (38400), 'Parity' (None, 1 Stop Bit), 'Timeout' (1.000 Seconds), and 'Server Connections' (1). The right column contains two radio button options: 'I act as the RTU master.' (selected) and 'I act as one or more RTU slaves.' Each option has a descriptive paragraph explaining its function. The 'I act as the RTU master' option states that it receives requests on the TCP port and forwards them to one or more RTU slaves connected to the RTU port. The 'I act as one or more RTU slaves' option states that it receives requests on the RTU port and forwards them to TCP ports at IP addresses listed on the TCP Device Map.

Select the baud rate and parity setting that all of the RTU slaves on the network are set for. Set a timeout value. This is the amount of time that the gateway will wait for a response from an RTU slave before calling it a "no-response" error.

For this mode, you should also enter the number of server connections to support. This will default to one. If more than one Modbus TCP client will be polling RTU devices at the same time through this gateway, then increase this number in order to allow more connections. If your TCP client opens and closes a different connection for each request (considered bad behavior by Modbus protocol but some devices do it anyway) then you may need to increase this number to sustain reliable connections. The gateway must be restarted after changing the number of server connections. (Note that setting the connection count substantially higher than what is actually in use will result in increased latency due to the gateway taking time to check for traffic on all of the unused connections).

**IMPORTANT:** Set timeout to something long enough for the device. If too short, the gateway will not wait long enough for a response from the Modbus slave device, and the result will be a lot of "no response" errors from the device even though the device is perfectly functional.

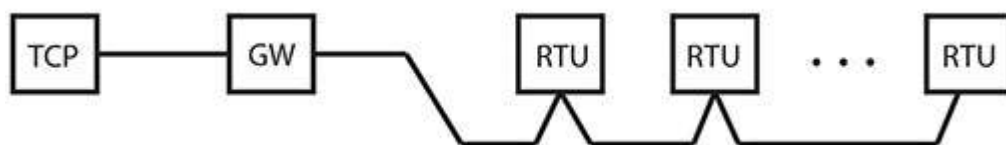
**IMPORTANT:** The timeout you set in your Modbus TCP client must be longer than the timeout set on this page. If the RTU slave queried times out, it will be tabulated as a no-response error here in the gateway, but the gateway will return an exception 11 to the TCP client telling it that the RTU slave timed out. If your TCP client timeout is shorter than the timeout set above, then your client will not receive that exception report.

Device #	First RTU Address In	First RTU Address Out	No. Units at this IP	Type	IP Address	Port	Status
1	0	0	0	IPv4	0.0.0.0	0	0

You do not need to do anything on the TCP Device Map page for this mode.

## 4.2 How It Works

When the gateway receives a request from the TCP device, the gateway will remove the TCP header, add an RTU checksum, and transmit the request on the RTU network. The gateway will then wait for a response from the RTU slave.



If the slave does not respond, the gateway will return exception code 11 to the TCP client (master). Exception code 11 is defined by Modbus protocol to mean "Gateway target device failed to respond". This is how a gateway reports a timeout or no-response error back to the requesting TCP device.

If the slave does respond, the checksum in the response will be verified. If the checksum test fails, meaning the message is corrupt, this fault will be reported back to the requesting TCP client with exception code 4 which is defined as "Slave device failure".

If a good response is received from the RTU slave and its checksum test passes, then the TCP header will be added to the response and it will be transmitted back to the requesting TCP client.





## 5. Accessing TCP Devices from RTU

### 5.1 Set Mode and Port Parameters

Accessing one or more Modbus TCP server devices from an external Modbus RTU master means the gateway needs to act as one or more RTU slaves on the RTU network. Therefore, select "I act as one or more RTU slaves" for mode.

The screenshot shows the 'Modbus' configuration screen of the Babel Buster 3. The top navigation bar has 'Modbus' selected. Below it, a sub-menu bar includes 'Mode / Port Info' (selected), 'TCP Device Map', 'Error Counts', and 'Packet Log'. An 'Update' button is in the top right. The main area has two columns. The left column is for 'I act as the RTU master.' and the right for 'I act as one or more RTU slaves.' Both columns have a description of the mode. At the top, 'Baud Rate' is set to 38400 and 'Parity' is set to 'None, 1 Stop Bit'. Below the mode descriptions, 'Timeout' is set to 1.000 seconds. At the bottom, 'Server Connections' is set to 1. Each mode has a corresponding explanation of the timeout and server connections settings.

Select the baud rate and parity setting that all of the RTU slaves on the network are set for. Set a timeout value. This is the amount of time that the gateway will wait for a response from a TCP server before calling it a "no-response" error.

The server connections setting on this page does not apply in this mode - the connection count is established automatically by the TCP Device Map.

**IMPORTANT:** Set timeout to something long enough for the device. If too short, the gateway will not wait long enough for a response from the Modbus server, and the result will be a lot of "no response" errors from the device even though the device is perfectly functional.

**IMPORTANT:** The timeout you set in your Modbus RTU master must be longer than the timeout set on this page. If the TCP server queried times out, it will be tabulated as a no-response error here in the gateway, but the gateway will return an exception 11 to the RTU master telling it that the TCP server timed out. If your RTU master timeout is shorter than the timeout set above, then your master will not receive that exception report. Yet, that exception report will still be sent upon timeout, and may then result in a collision on the network if the RTU master has started sending the next request before waiting to receive the exception reply to the previous request. This known collision will take on the appearance of further communication problems including further no-response errors and possibly CRC errors. These will be avoided with timeout settings that do not conflict.

## 5.2 Create TCP Device Map

The gateway can present one or more TCP servers as one or more RTU slaves. There can be a one to one mapping, or a many to one mapping. If the TCP device is itself another gateway, then it may contain multiple slaves within its one TCP network address. The most common application will be a one to one mapping of RTU address to TCP address.

The TCP Device Map is where you establish your association between RTU address and TCP device IP address. When operating in this mode, if an RTU address is not found in this table, then the gateway will simply not respond. The expected behavior for RTU slave devices is that they remain silent if the request does not contain their address (aka slave ID or unit number). This gateway may be acting as multiple RTU slaves, but it will remain silent if the RTU address is not found in this table.

A typical one to one mapping of single RTU address to TCP address is illustrated below.




Modbus System

Mode / Port Info TCP Device Map Error Counts Packet Log

The following TCP devices are mapped. Showing 1 to 2 of 2 Update < Prev Next >

Device #	First RTU Address In	First RTU Address Out	No. Units at this IP	Type	IP Address	Port	Status
1	1	1	1	IPv4	192.168.1.168	502	0
2	0	0	0	IPv4	0.0.0.0	0	0

Reset

The first column, "First RTU Address In", should be the RTU slave address that you want the gateway to respond to. Assuming you do not want to translate addresses as they pass through the gateway, the First RTU Address Out would be the same address. The count ("No. Units at this IP") for a one to one association is simply one. Then select IP address type, enter that address in notation standard for the address type, and port number. The standard port normally reserved for Modbus TCP is 502. If no port is given, the gateway will use 502 by default. A non-standard port number can also be used.

An example for mapping an IPv6 address is illustrated below.




Modbus System

Mode / Port Info TCP Device Map Error Counts Packet Log

The following TCP devices are mapped. Showing 1 to 2 of 2 Update < Prev Next >

Device #	First RTU Address In	First RTU Address Out	No. Units at this IP	Type	IP Address	Port	Status
1	1	1	1	IPv6	fec0::6	502	0
2	0	0	0	IPv4	0.0.0.0	0	0

Reset

Each device must have a unique RTU address on the RTU network. Therefore, when mapping multiple TCP servers as RTU slaves, the First RTU Address In should be the unique RTU address you want to associate with a specific TCP device.

The First RTU Address In is the address to which the gateway will respond as an RTU slave. But the TCP servers may all expect to see unit number one in the request (or some other specific unit). In this case, the "First RTU Address Out" should be the address that the TCP server wants to see. The RTU address in the incoming request will be replaced with the First RTU Address Out in the request that is forwarded to the TCP server.

Numbers in the First RTU Address In column must be unique. If an RTU address is duplicated, only the first entry will be used.

Modbus		System					
Mode / Port Info		TCP Device Map		Error Counts		Packet Log	
The following TCP devices are mapped.				Showing 1 to 6 of 6		<input type="button" value="Update"/> <input type="button" value=" &lt; Prev"/> <input type="button" value="Next &gt;"/>	
Device #	First RTU Address In	First RTU Address Out	No. Units at this IP	Type	IP Address	Port	Status
1	1	1	1	IPv4 ▾	192.168.1.21	0	0
2	2	1	1	IPv4 ▾	192.168.1.33	0	0
3	3	1	1	IPv4 ▾	192.168.1.39	0	0
4	4	1	1	IPv4 ▾	192.168.1.62	0	0
5	5	1	1	IPv4 ▾	192.168.1.87	0	0
6	0	0	0	IPv4 ▾	0.0.0.0	0	0
<input type="button" value="Reset"/>							

When you have the unique situation of the TCP device being another gateway, then you have the option of mapping a range of RTU addresses to a single TCP IP address. DO NOT duplicate IP addresses in the table. Duplicate IP addresses will not function properly.

In the example below, the gateway will respond as RTU slaves 1 through 5. The unit numbers in the requests forwarded to the TCP server at the given IP address will be 20 through 25.

Modbus		System					
Mode / Port Info		TCP Device Map		Error Counts		Packet Log	
The following TCP devices are mapped.				Showing 1 to 2 of 2		<input type="button" value="Update"/> <input type="button" value=" &lt; Prev"/> <input type="button" value="Next &gt;"/>	
Device #	First RTU Address In	First RTU Address Out	No. Units at this IP	Type	IP Address	Port	Status
1	1	20	5	IPv4 ▾	192.168.1.112	0	0
2	0	0	0	IPv4 ▾	0.0.0.0	0	0
<input type="button" value="Reset"/>							

When you have a long list of TCP devices, use the Prev and Next buttons to scroll through the list. This gateway will support up to 120 connections to different TCP devices.



# Babel Buster® 3

MODBUS  
NETWORK GATEWAY  
MODEL BB3-6101-6W

**CONTROL SOLUTIONS MINNESOTA**

Modbus

System

Mode / Port Info

TCP Device Map

Error Counts

Packet Log

The following TCP devices are mapped. Showing 1 to 15 of 50 Update < Prev Next >

Device #	First RTU Address In	First RTU Address Out	No. Units at this IP	Type	IP Address	Port	Status
1	1	1	1	IPv4 ▾	192.168.1.168	502	0
2	2	1	1	IPv4 ▾	192.168.1.134	502	0
3	3	1	1	IPv4 ▾	192.168.1.64	502	0
4	4	1	1	IPv4 ▾	192.168.1.67	502	0
5	5	1	1	IPv4 ▾	192.168.1.148	502	0
6	6	1	1	IPv4 ▾	192.168.1.38	502	0
7	7	1	1	IPv4 ▾	192.168.1.65	502	0
8	8	1	1	IPv4 ▾	192.168.1.43	502	0
9	9	1	1	IPv4 ▾	192.168.1.23	502	0
10	10	1	1	IPv4 ▾	192.168.1.24	502	0
11	11	1	1	IPv4 ▾	192.168.1.14	502	0
12	12	1	1	IPv4 ▾	192.168.1.87	502	0
13	13	1	1	IPv4 ▾	192.168.1.178	502	0
14	14	1	1	IPv4 ▾	192.168.1.156	502	0
15	15	1	1	IPv4 ▾	192.168.1.33	502	0

Reset

To remove a device from the list, simply set its count ("No. Units at this IP") to zero. When this configuration is saved, any lines with a count of zero are skipped in writing the XML file. The next time the file is reloaded, the lines with zero counts will be completely gone. Any lines with a count of zero are skipped in address lookup.

### 5.3 How It Works

When the gateway receives a request from the RTU master, it will first verify the checksum. The message will be discarded if there is a checksum error and there will be no response. Next, the gateway scans the TCP Device Map table to see if the RTU address in the request matches any RTU address in the table. If no match is found, the message is disregarded and assumed to be intended for some other device on the RTU network.

If the message passes the checksum test and the RTU address is found in the TCP Device Map table, the gateway will then remove the RTU checksum and add the TCP header. The unit number in the TCP request will be replaced with the

corresponding unit number from the map table. The gateway will then transmit the request on the TCP network and then wait for a response.



If the TCP device fails to connect or times out, the gateway will return exception code 11 to the RTU master. Exception code 11 is defined by Modbus protocol to mean "Gateway target device failed to respond". This is how a gateway reports a timeout or no-response error back to the requesting device.

If a good response is received from the TCP server, then the TCP header is removed, the original RTU address is restored, a checksum is added, and the response is transmitted back to the RTU master.




## 6. Error Counts and Packet Log

### 6.1 Reviewing Error Counts

The Error Counts page is the first place to look if you are not getting the responses you expect. Messages are tabulated by RTU address regardless of operating mode. Total messages tabulates all messages including exception messages and requests that get no response. The number of times a device fails to respond is tabulated. The number of times a CRC error is detected on the RTU link is tabulated. The number of times an exception message is sent is tabulated, and the exception code found in the most recent exception message is logged. Exceptions can be generated by this gateway, or by the device to which a request was sent. Regardless of the origin of the exception, it is tabulated and logged here.



Modbus

System

Mode / Port Info

TCP Device Map

**Error Counts**

Packet Log

Showing devices from 

Update

< Prev

Next >

Unit #	Reset	Total Messages	No Responses	CRC Errors	Exceptions	Last Exception
1	<input type="checkbox"/>	731	0	0	0	0
2	<input type="checkbox"/>	731	0	0	0	0
3	<input type="checkbox"/>	731	0	0	0	0
4	<input type="checkbox"/>	731	0	0	0	0
5	<input type="checkbox"/>	731	0	0	0	0
6	<input type="checkbox"/>	731	0	0	731	11
7	<input type="checkbox"/>	729	0	0	0	0
8	<input type="checkbox"/>	731	0	0	0	0
9	<input type="checkbox"/>	730	0	0	0	0
10	<input type="checkbox"/>	730	0	0	0	0
11	<input type="checkbox"/>	730	0	0	0	0
12	<input type="checkbox"/>	730	0	0	0	0
13	<input type="checkbox"/>	730	0	0	730	2
14	<input type="checkbox"/>	730	0	0	0	0
15	<input type="checkbox"/>	730	0	0	0	0

Non-specific CRC errors: 14
 

Clear All

**IMPORTANT:** If you observe timeouts (no-response errors) at your Modbus RTU master or Modbus TCP client, but do not see no-response errors tabulated here, then the timeout setting in your master or client may be too short. If the device on the other side of this gateway fails to respond, it will be logged as a no-response error here, and your master/client should receive an exception 11 report. You should never see just a timeout on the master/client side of this gateway. Refer to section 4 or 5 as applicable for additional notes about timeout settings.

## 6.2 Reviewing TCP Device Status

When accessing TCP devices from RTU, an additional diagnostic is available on the TCP Device Map page. If a TCP device has failed to connect or failed to respond, its status in the Status column will be some number other than zero (0=no error detected). Status codes will normally clear on their own, but you can also force them to clear by clicking the Reset button. If the condition still exists, the status code will return to its non-zero value upon the next attempt to connect.

**Babel Buster<sup>®</sup> 3**  
MODBUS  
NETWORK GATEWAY  
MODEL BB3-6101-GW

**CONTROL SOLUTIONS MINNESOTA**

Modbus System

Mode / Port Info TCP Device Map Error Counts Packet Log

The following TCP devices are mapped. Showing 1 to 2 of 2 Update < Prev Next >

Device #	First RTU Address In	First RTU Address Out	No. Units at this IP	Type	IP Address	Port	Status
1	1	1	1	IPv4	192.168.1.168	502	111
2	0	0	0	IPv4	0.0.0.0	0	0

Reset

Connection status codes you may see include:

5 = Connection attempt timed out, unable to establish connection (usually means remote device not connected or not reachable)

104 = Connection reset by peer

111 = Connection refused

113 = Connection aborted

114 = Network is unreachable

115 = Network interface not configured

116 = Connection timed out

118 = Host is unreachable

125 = Address not available

### 6.3 Reviewing the Packet Log

To further aid in diagnostics, a packet log is provided where you can see the last 200 or so raw messages. The message's unit or slave address, function code, and data are displayed. RTU checksum and TCP header are omitted from the display to aid in clarity.

The most recent message is always displayed at the top, and elapsed time gives a relative indication of when the message was sent. The Source/Dest column indicates the "from" and "to". Content is the actual message sent.


Indicator

-> Request from master to slave

<- Response from slave

The example packet log below shows what the packet log can look like when accessing RTU devices from TCP. The RTU slave address is indicated as "RTU x". There is no TCP number in this case because the TCP Device Map is not used in this mode.

**Babel Buster® 3**  
**MODBUS**  
**NETWORK GATEWAY**  
**MODEL BB3-6101 GW**

  
**CONTROL SOLUTIONS MINNESOTA**

Modbus

System

Mode / Port Info

TCP Device Map

Error Counts

Packet Log

Top

Next >

Elapsed Time	Source/Dest	Content
00:00:02	TCP <- RTU 1	01 03 0A 00 65 00 00 00 00 00 00 00
00:00:02	TCP -> RTU 1	01 03 00 00 00 05
00:00:02	TCP <- RTU 3	03 03 0A 01 2D 00 00 00 00 00 00 00
00:00:02	TCP -> RTU 3	03 03 00 00 00 05
00:00:02	TCP <- RTU 4	04 03 0A 01 91 00 00 00 00 00 00 00
00:00:02	TCP -> RTU 4	04 03 00 00 00 05
00:00:02	TCP <- RTU 2	02 03 0A 00 C9 00 00 00 00 00 00 00
00:00:02	TCP -> RTU 2	02 03 00 00 00 05
00:00:05	TCP <- RTU 1	01 03 0A 00 65 00 00 00 00 00 00 00
00:00:05	TCP -> RTU 1	01 03 00 00 00 05
00:00:05	TCP <- RTU 3	03 03 0A 01 2D 00 00 00 00 00 00 00
00:00:05	TCP -> RTU 3	03 03 00 00 00 05
00:00:05	TCP <- RTU 4	04 03 0A 01 91 00 00 00 00 00 00 00
00:00:05	TCP -> RTU 4	04 03 00 00 00 05
00:00:05	TCP <- RTU 2	02 03 0A 00 C9 00 00 00 00 00 00 00

When accessing TCP devices from RTU, then there is a number with "TCP" in the Source/Dest column. This number is the device number as listed on the TCP Device Map. The number indicated along with "RTU" is the RTU slave address as recognized on the RTU link.

Mode / Port Info		TCP Device Map	Error Counts	Packet Log	
					Top Next >
Elapsed Time	Source/Dest	Content			
00:00:08	RTU 41 <- TCP 41	29 03 04 00 00 00 00			
00:00:08	RTU 41 -> TCP 41	29 03 00 00 00 00 01			
00:00:08	RTU 40 <- TCP 40	28 83 02			
00:00:08	RTU 40 -> TCP 40	28 03 00 00 00 00 01			
00:00:08	RTU 39 <- TCP 39	27 83 02			
00:00:08	RTU 39 -> TCP 39	27 03 00 00 00 00 01			
00:00:08	RTU 38 <- TCP 38	26 03 02 00 00			
00:00:08	RTU 38 -> TCP 38	26 03 00 00 00 00 01			
00:00:08	RTU 37 <- TCP 37	25 83 02			
00:00:08	RTU 37 -> TCP 37	25 03 00 00 00 00 01			
00:00:09	RTU 36 <- TCP 36	24 03 02 00 00			
00:00:09	RTU 36 -> TCP 36	24 03 00 00 00 00 01			
00:00:09	RTU 35 <- TCP 35	23 03 02 00 00			
00:00:09	RTU 35 -> TCP 35	23 03 00 00 00 00 01			
00:00:09	RTU 34 <- TCP 34	22 03 02 00 00			
00:00:09	RTU 34 -> TCP 34	22 03 00 00 00 00 01			
00:00:11	RTU 33 -> TCP 33	21 03 00 00 00 00 01			
00:00:11	RTU 32 <- TCP 32	20 03 02 EA 3D			
00:00:12	RTU 32 -> TCP 32	20 03 00 00 00 00 01			
00:00:12	RTU 31 <- TCP 31	1F 03 02 00 14			
00:00:12	RTU 31 -> TCP 31	1F 03 00 00 00 00 01			
00:00:12	RTU 30 <- TCP 30	1E 03 04 00 00 00 00			
00:00:12	RTU 30 -> TCP 30	1E 03 00 00 00 00 01			
00:00:12	RTU 29 <- TCP 29	1D 03 02 00 00			
00:00:12	RTU 29 -> TCP 29	1D 03 00 00 00 00 01			
00:00:12	RTU 28 <- TCP 28	1C 03 02 00 00			
00:00:12	RTU 28 -> TCP 28	1C 03 00 00 00 00 01			
00:00:12	RTU 27 <- TCP 27	1B 03 02 E8 2C			
00:00:13	RTU 27 -> TCP 27	1B 03 00 00 00 00 01			
00:00:13	RTU 26 <- TCP 26	1A 03 02 00 00			

The above examples illustrated short messages reading only a couple of registers. Longer messages will also be displayed in their entirety. In this particular example, the Modbus holding registers contained a lot of uneventful zeroes.



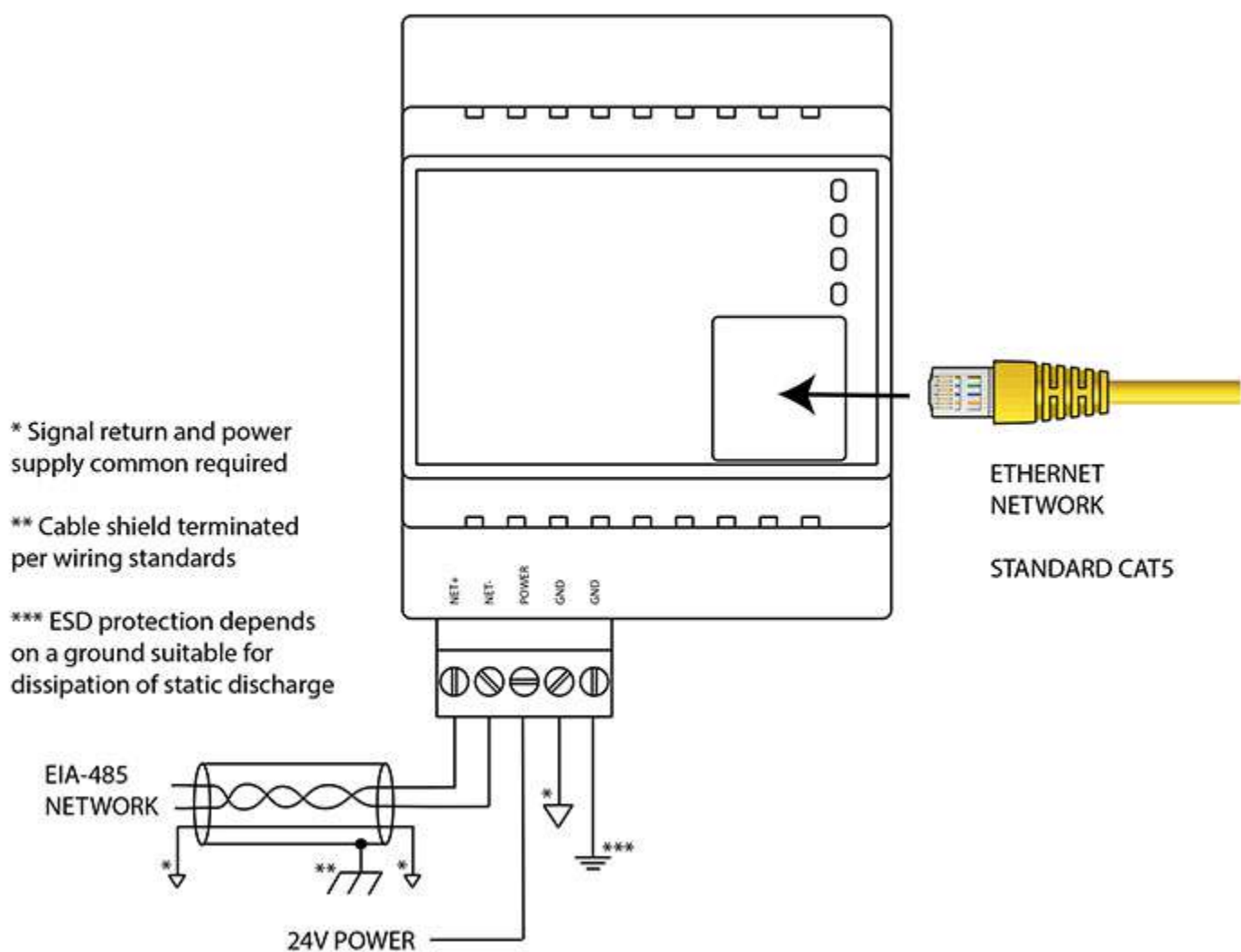
5/4/2021, 9:56 AM



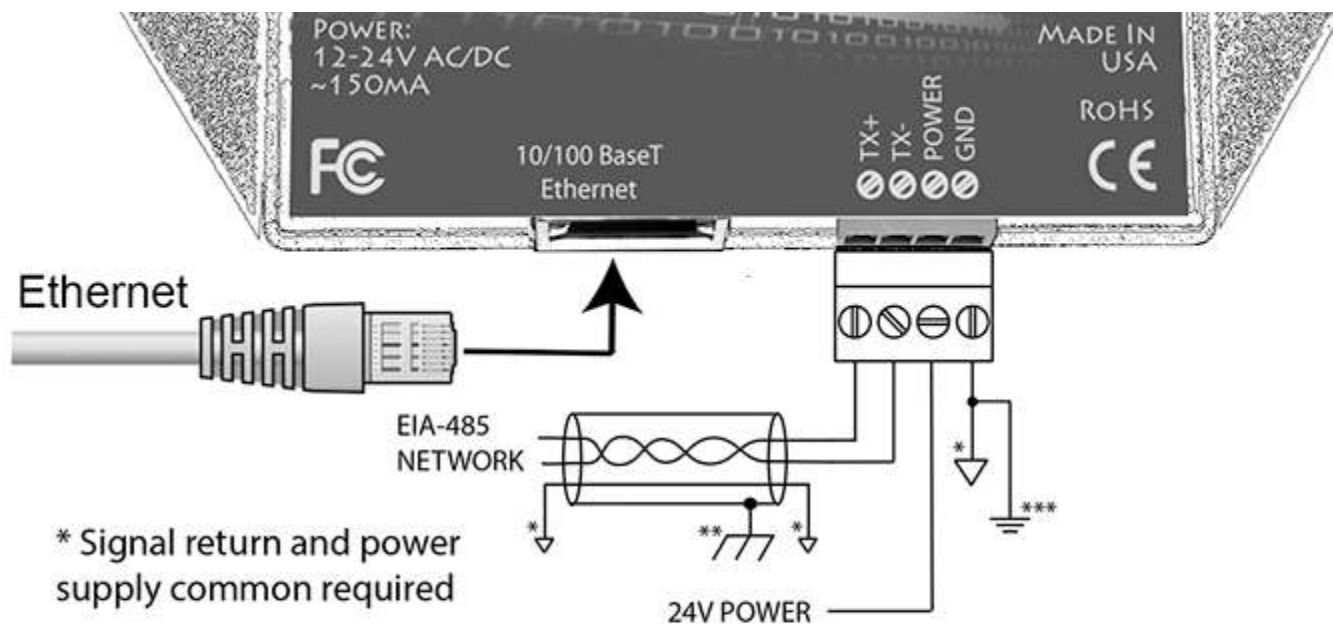
## Appendix A Hardware Details

### A.1 Wiring

Wiring for the Babel Buster BB3-6101 is illustrated below.



Wiring for the Babel Buster MX-61 is illustrated below.



\* Signal return and power supply common required

\*\* Cable shield terminated per wiring standards

\*\*\* ESD protection depends on a ground suitable for dissipation of static discharge

Wire the gateway as illustrated. Follow all conventional standards for wiring of EIA-485 networks when connecting the Modbus RTU EIA-485 (RS485) network. This includes use and termination of shield, termination of the network, and grounding.

**IMPORTANT:** Although EIA-485 (RS485) is thought of as a 2-wire network, you **MUST** include a third conductor connected to GND or common at each device so that all devices are operating at close to the same ground potential. Proper grounding of equipment should ensure proper operation without the third conductor; however, proper grounding often cannot be relied upon. If large common mode voltages are present, you may even need to insert optically isolated repeaters between EIA-485 devices.

Use standard CAT5 cables for Ethernet connections. Use control wire as applicable for local electrical codes for connecting the 24V (AC or DC) power supply.

Note that in addition to connecting power supply common to a GND terminal, you must also connect a GND terminal to earth ground in order to ensure proper ESD protection.

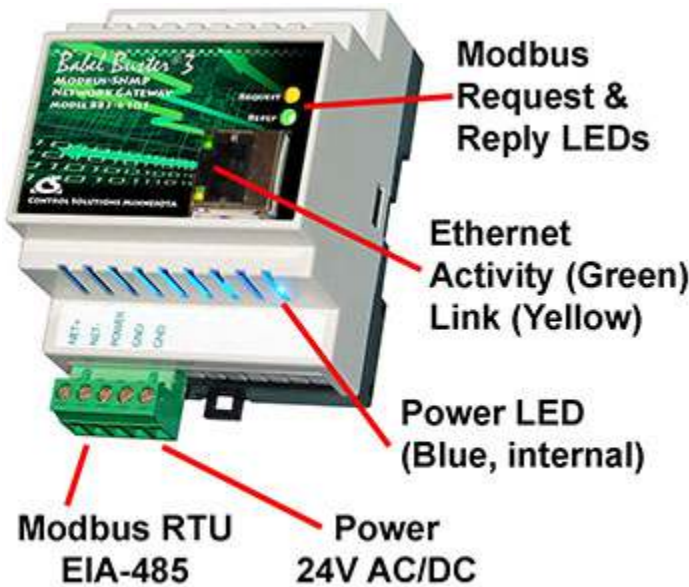
**BB3-6101-232:** The standard BB3-6101 Modbus RTU port uses RS-485. The RS-232 version replaces the RS-485 transceiver with an RS-232 transceiver. The NET+/NET- terminals are replaced by TXD and RXD on the -232 version. TXD is data out from the BB3-6101-232, and RXD is data in to the gateway. Hardware handshake is not supported.



## A.2 Front Panel LED Indicators

### A.2.1 BB3-6101 LED Indicators

Power-up LED behavior: On power up, the Reply LED will remain on solid red for about 20 seconds, then the Request and Reply LEDs will do a "lamp test" where Request is yellow and Reply is Red simultaneously for about 1 second, and then both Request and Reply turn green simultaneously for about 1 second. The LEDs will then begin to operate according to their normal functionality.



Babel Buster BB3-6101 Request and Reply LEDs reflect Modbus RTU traffic, and the Ethernet activity LED will indicate network traffic in general. If Modbus RTU is not being used at all, then the Request and Reply LEDs will indicate TCP traffic. If Modbus RTU is in use, then the Request and Reply LEDs will indicate Modbus RTU traffic while the Ethernet LEDs will be the only indication of TCP traffic.

Babel Buster BB3-6101 LEDs indicate as follows (LEDs are bi-color):

REQUEST	Flashes yellow each time a request is sent when operating as Modbus Master, or each time a request is received when operating as Modbus Slave.
REPLY	<p>Operating as Modbus Master, flashes green each time a good response is received, or red when an error code is received, the request times out, or there is a flaw in the response such as CRC error.</p> <p>Operating as Modbus Slave, flashes green each time a good response is sent, or red if an exception code is sent (meaning the received request resulted in an error).</p>
Ethernet Activity	Green LED is on solid during portions of the boot-up process, and then flashes briefly when Ethernet network traffic is

	detected.
Ethernet Link	Yellow LED indicates an Ethernet link is present. This indicator will light if a link is present regardless of processor or network activity. If not lit, check network wiring.
Status	Blue LED (internal) on any time power is present and internal power supply is functioning.

### A.2.2 MX-61 LED Indicators

Power-up LED behavior: On power up, the Request, Reply and Error LEDs will remain off for about 20 seconds, then all three LEDs will do a "lamp test" where they all turn on simultaneously for about 1 second. The LEDs will then begin to operate according to their normal functionality.



Babel Buster MX-61 Request, Reply and Error LEDs reflect Modbus RTU traffic, and the Ethernet activity LED will indicate network traffic in general.

Babel Buster MX-61 LEDs indicate as follows (LEDs are each a single color):

Error (red)	<p>Operating as Modbus Master, flashes red when an error code is received, the request times out, or there is a flaw in the response such as CRC error.</p> <p>Operating as Modbus Slave, flashes red if an exception code is sent (meaning the received request resulted in an error).</p>
Request (yellow)	Flashes yellow each time a request is sent when operating as Modbus Master, or each time a request is received when operating as Modbus Slave.

Reply (green)	<p>Operating as Modbus Master, flashes green each time a good response is received.</p> <p>Operating as Modbus Slave, flashes green each time a good response is sent.</p>
Ethernet Activity	Green LED is on solid during portions of the boot-up process, and then flashes briefly when Ethernet network traffic is detected.
Ethernet Link	Yellow LED indicates an Ethernet link is present. This indicator will light if a link is present regardless of processor or network activity. If not lit, check network wiring.
Status	Blue LED (internal) on any time power is present and internal power supply is functioning.

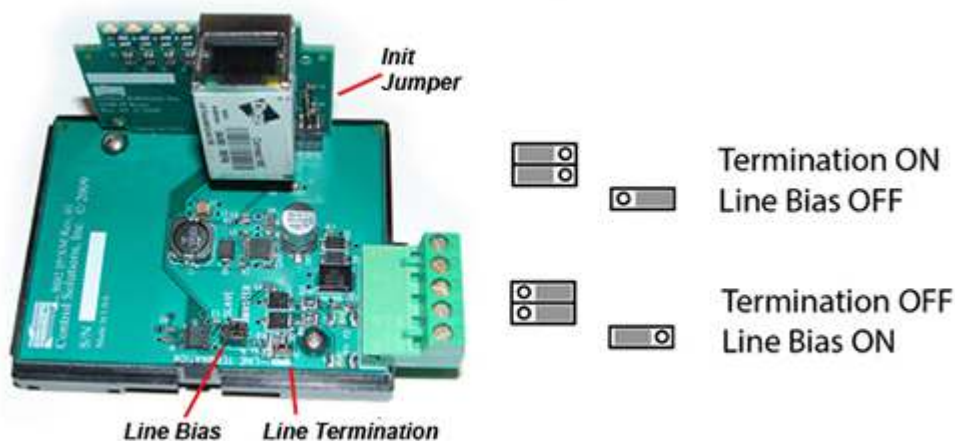
### A.3 RS-485 Line Termination & Bias

Enable line termination only when this device is placed at the end of the network. Termination should only be enabled at two points on the network, and these two points must be specifically the end points.

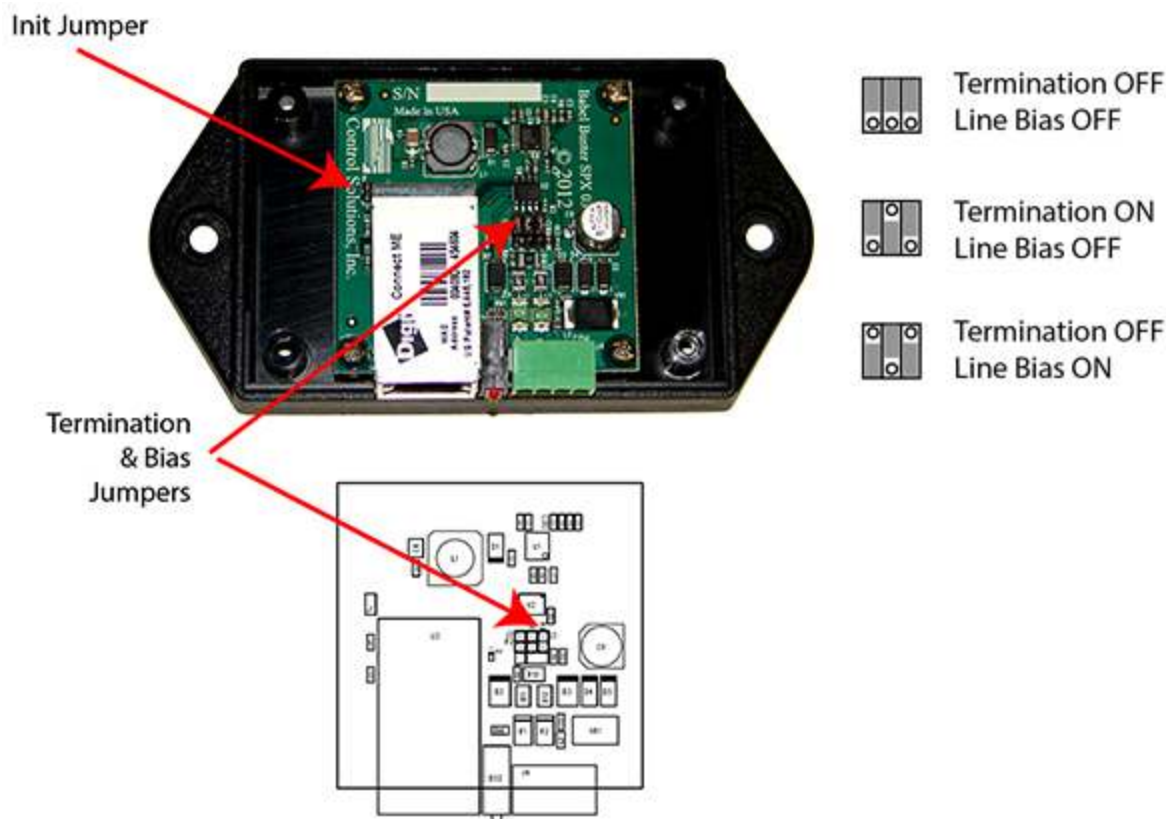
Enable line bias when needed. Line bias should only be enabled at one point on the network, and does not have to be the end point. Line bias holds the line in a known neutral state when no devices are transmitting. Without bias, the transition from offline to online by a transmitter can look like a false start bit and cause loss of communication.

The line conditioning options are enabled when the respective shunt is moved to the position indicated by the diagrams below.

Jumper locations for Babel Buster BB3-6101:



Jumper locations for Babel Buster MX-61:



## A.4 Soft Configuration Reset

Soft reset should be used to remove all configuration information any time you do have the ability to connect to the gateway's web user interface. The "Clear Configuration" action is described in Section 3.1.5. Using the forced hard reset should only be used as a last resort if you are unable to connect to the gateway because the SSL certificates are invalid for a secure connection or you are unable to recover the lost IP address.

## A.5 Discovering Lost IP Address

You can use Wireshark to discover a lost IP address if the gateway is still functional. Connect the gateway directly to your PC running Wireshark using a cross-over cable (or standard CAT5 cable if your PC supports auto-MDX). With Wireshark running, power up the gateway.

Upon power up, BB3-6101/MX-61 will ping its own IP address one or more times. This is part of its duplicate address resolution mechanism. If it finds another device with its own IP address, it will set its own IP address to a default pseudo-random address generally starting with 192.

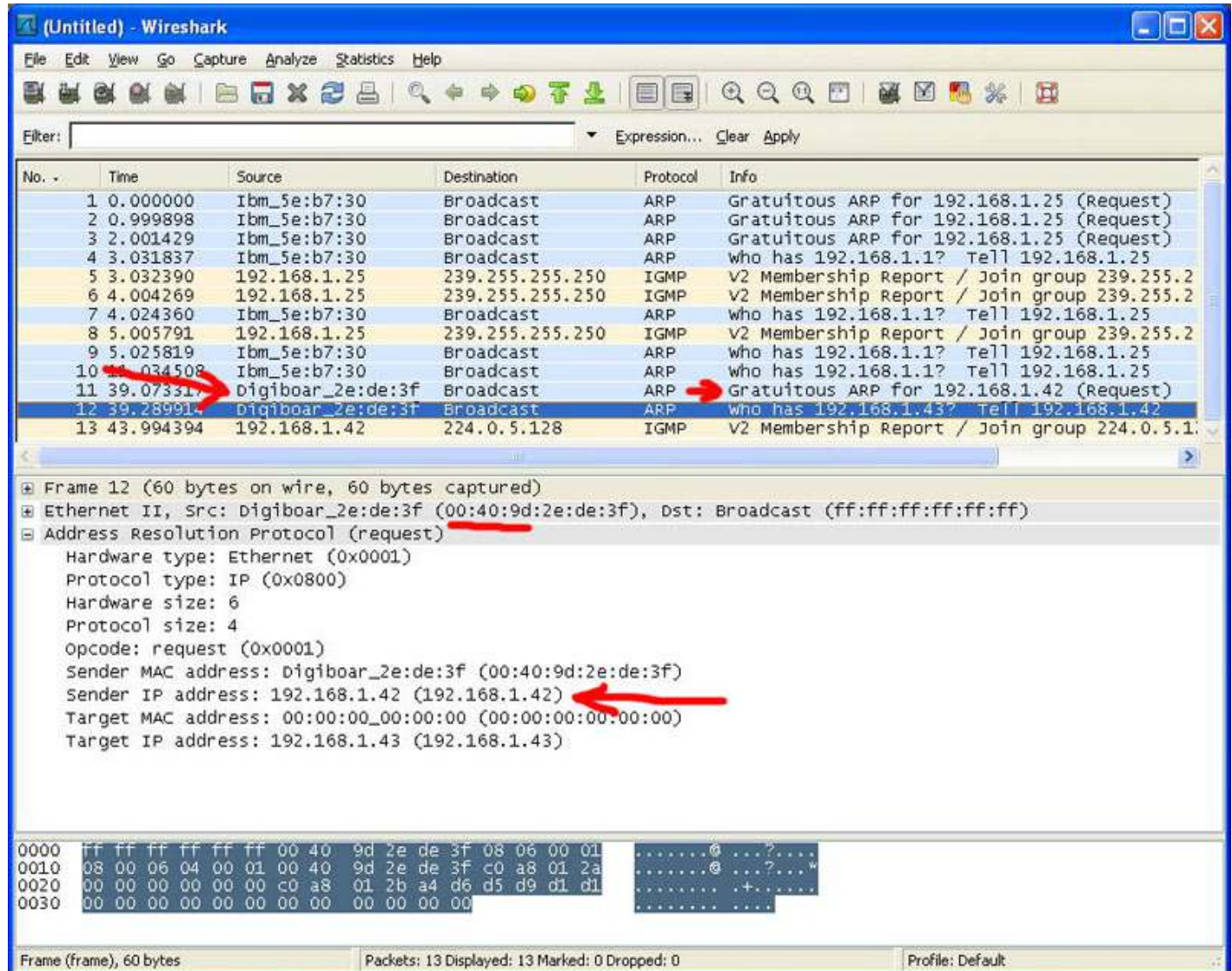
Wait until you are certain BB3-6101/MX-61 has booted up, or wait 2-3 minutes to be sure if you don't recognize the bootup LED sequence. Now look for the ARP packets and note what IP address they came from. This is your device. (To make sure it is your device, connect only the BB3-6101/MX-61 to your PC while doing this exercise.)

Your device will have a MAC address that starts with 00:40:9D, also labeled with a



source that starts with "Digiboar\_". This label comes from the fact that the server modules used on Control Solutions IP products are made by Digi International, previously known as "Digiboard".

There will usually be one or more "pings" or ARP packets to the device's own IP address, and one last ping to its own address plus one. In the illustration here, the BB3-6101/MX-61 is located at 192.168.1.42.



## A.6 Forced Hard Configuration Reset

**IMPORTANT:** Before considering the forced hard reset, be sure you have considered soft configuration reset, or discovering lost IP address if applicable.

The "Init" jumper inside the BB3-6101/MX-61 serves two purposes, and what it does depends on whether you apply the jumper before or after the BB3-6101/MX-61 boots up.

### Hard Configuration Reset:

Installing the jumper after bootup causes the BB3-6101/MX-61 to do a hard reset on



its configuration memory. The IPv4 address will be reset to 10.0.0.101. The root password will be reset to the original default password. After clearing all configuration, the BB3-6101/MX-61 will automatically restart. Remove the jumper when you see the indication of restart after about 30 seconds, which is both LEDs coming on solid on the RJ45 Ethernet connector and remaining on for a couple of seconds. If you miss the start of reboot, both LEDs on the RJ45 will come on and stay on. It will now be attempting the firmware update, but you can abort that by simply powering down the BB3-6101/MX-61. If both LEDs on the RJ45 jack come on and remain on, remove the jumper and then power cycle the BB3-6101/MX-61.

Once you have regained access to the device, go to the File Manager page, execute the Clear All configuration action, then select the file named as "Boot configuration" and execute the Save XML Config File action to wipe out any configuration normally saved in the XML configuration file.

Note: The forced hard reset will restore HTTP web access and disable HTTPS web access. The forced hard reset will also restore FTP access to allow FTP firmware uploads if needed.

Note: The hard reset of configuration also means all of your resource allocations are reset to original factory defaults. If you want resource allocations that are different, you will need to repeat the allocation setup as described in Section 3.4.

### **Firmware Update Recovery:**

Installing this jumper prior to power-up causes the server to go into TFTP firmware update mode. Normally you would perform a firmware update by simply uploading a new image.bin file (provided by Control Solutions tech support) using the BB3-6101/MX-61's internal FTP server and a command line FTP session on your PC (Linux or Windows command line). Detailed instructions are included in the zip file that also contains the applicable image.bin file.

Should the FTP upload fail for some reason, then you need to resort to the TFTP upload method as the fallback method. Full details on how to go about this can be found under the topic "Restoring a corrupt application image" at <https://info.csimn.com>.

### **Additional maintenance page:**

Go to [http\(s\)://10.0.0.101/html/pgRestoreAddr.html](http(s)://10.0.0.101/html/pgRestoreAddr.html) to find the following page (substituting your IP address). It serves two purposes as noted below, which ideally you will never have a use for.



The screenshot shows the Babel Buster 3 web interface. At the top left, it says "Babel Buster® 3 MODBUS-SNMP NETWORK GATEWAY MODEL BB3-6101". At the top right is the "CONTROL SOLUTIONS MINNESOTA" logo. Below this, there are two input fields. The first is labeled "Valid MAC Address" and contains the text "00:40:9D:45:45:EA". To its right is a button labeled "Restore". The second input field is labeled "Reformat Flash file system" and is empty. To its right is a button labeled "Wipe".

## File System Wipe:

On rare occasion, the Flash file system has been observed to get corrupted as a result of losing power while a write operation was in progress. This is most effectively confirmed by opening a command prompt FTP session (Windows 10 PowerShell) to try to view the files in the Flash file system. If FTP fails to show any files, in addition to other problems saving or loading files, it may be that the file system has gotten corrupted. If this happens, go to the page pictured above, and enter the Reformat key, then click Wipe, and then power cycle the device (or restart from the File Manager page). The reformat key is 55AAAA55. Simply type that into the window next to the Wipe button.

## MAC Address Restore:

In the event the MAC address has been reset due to NVRAM checksum failure, this page will permit restoring the MAC address to its original address as printed on the component label internal to this device, or on the default password label found on the outside or on external documentation included with the device.

If the MAC address is deemed to be valid, the window will be labeled "Valid MAC Address" and you will not be allowed to change it. If the MAC address is deemed to be invalid, the window will be labeled "Restore MAC Address" and you should then enter the correct MAC address and click Restore. A restart is then needed.

## A.7 Firmware Update Notes

The most up to date firmware is shipped with all new devices. This isn't like a new laptop where you spent the first half a day updating software on a computer you thought was brand new. If you believe you have discovered an issue that you believe a firmware update might fix, contact technical support first to confirm whether that is the case, and then to get a login to the firmware update support site.

The brute force approach to updating firmware using TFTP as noted in the section above is always available, but the more graceful approach is to use FTP to upload the new image.bin file. There is one minor problem: The upload wants to buffer the entire file in RAM while it proceeds to reprogram the Flash memory. **If the memory**

**utilization indicated on the Resources page in your device is above about 30%, the FTP upload will fail, and thus the firmware update will not take place.**

You have two choices: (1) Use the TFTP approach, or (2) Temporarily reconfigure your gateway to use a minimum of resources to free up space to temporarily buffer the image.bin file upload.

More detailed instructions for the FTP upload are included in the zip file you will download to obtain the firmware update. Instructions for the TFTP upload are available in our knowledgebase at <https://info.csimn.com>.



## Appendix B Modbus Reference Information

### B.1 Function Codes, Error Codes, and More

#### Modbus Register Types

The types of registers referenced in Modbus devices include the following:

- Coil (Discrete Output)
- Discrete Input
- Input Register
- Holding Register

Whether a particular device includes all of these register types is up to the manufacturer. It is very common to find all I/O mapped to holding registers only. Coils are 1-bit registers, are used to control discrete outputs, and may be read or written. Discrete Inputs are 1-bit registers used as inputs, and may only be read. Input registers are 16-bit registers used for input, and may only be read. Holding registers are the most universal 16-bit register, may be read or written, and may be used for a variety of things including inputs, outputs, configuration data, or any requirement for "holding" data.

#### Modbus Function Codes

Modbus protocol defines several function codes for accessing Modbus registers. There are four different data blocks defined by Modbus, and the addresses or register numbers in each of those overlap. Therefore, a complete definition of where to find a piece of data requires both the address (or register number) and function code (or register type).

The function codes most commonly recognized by Modbus devices are indicated in the table below. This is only a subset of the codes available - several of the codes have special applications that most often do not apply.

Function Code	Register Type
1	Read Coil
2	Read Discrete Input
3	Read Holding Registers
4	Read Input Registers
5	Write Single Coil
6	Write Single Holding Register

15	Write Multiple Coils
16	Write Multiple Holding Registers

## Modbus Exception (error) Codes

When a Modbus slave recognizes a packet, but determines that there is an error in the request, it will return an exception code reply instead of a data reply. The exception reply consists of the slave address or unit number, a copy of the function code with the high bit set, and an exception code. If the function code was 3, for example, the function code in the exception reply will be 0x83. The exception codes will be one of the following:

1	Illegal Function	The function code received in the query is not recognized by the slave or is not allowed by the slave.
2	Illegal Data Address	The data address (register number) received in the query is not an allowed address for the slave, i.e., the register does not exist. If multiple registers were requested, at least one was not permitted.
3	Illegal Data Value	The value contained in the query's data field is not acceptable to the slave.
4	Slave Device Failure	An unrecoverable error occurred.
6	Slave Device Busy	The slave is engaged in processing a long-duration command. The master should try again later.
10 (hex 0A)	Gateway Path Unavailable	Gateway could not establish communication with target device.
11 (hex 0B)	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways, indicates no response was received from the target device.
17 (hex 11)	Gateway Target Device Failed to Respond	No response from slave, request timed out.

## Modicon convention notation for Modbus registers

Modbus was originally developed by Gould-Modicon, which is presently Schneider Electric. The notation originally used by Modicon is still often used today, even though considered obsolete by present Modbus standards. The advantage in using the Modicon notation is that two pieces of information are included in a single number: (a) The register type; (b) The register number. A register number offset defines the type.

The types of registers referenced in Modbus devices, and supported by Babel Buster gateways, include the following:

- Coil (Discrete Output)
- Discrete Input
- Input Register
- Holding Register



Valid address ranges as originally defined for Modbus were 0 to 9999 for each of the above register types. Valid ranges allowed in the current specification are 0 to 65,535. The address range applies to each type of register, and one needs to look at the function code in the Modbus message packet to determine what register type is being referenced. The Modicon convention uses the first digit of a register reference to identify the register type.

Register types and reference ranges recognized by Modicon notation are as follows:

0x = Coil = 00001-09999  
1x = Discrete Input = 10001-19999  
3x = Input Register = 30001-39999  
4x = Holding Register = 40001-49999

Translating references to addresses, reference 40001 selects the holding register at address 0000, most often referred to as holding register number 1.

On occasion, it was necessary to access more than 10,000 of a register type using Modicon notation. Based on the original convention, there is another defacto standard that looks very similar. Additional register types and reference ranges recognized by Modicon notation are as follows:

0x = Coil = 000001-065535  
1x = Discrete Input = 100001-165535  
3x = Input Register = 300001-365535  
4x = Holding Register = 400001-465535

Translating references to addresses, reference 400001 selects the holding register at address 0000, most often referred to as holding register number 1.

### **If registers are 16-bits, how does one read Floating Point or 32-bit data?**

Modbus protocol defines a holding register as 16 bits wide; however, there is a widely used defacto standard for reading and writing data wider than 16 bits. The most common are IEEE 754 floating point, and 32-bit integer. The convention may also be extended to double precision floating point and 64-bit integer data.

The wide data simply consists of two consecutive "registers" treated as a single wide register. Floating point in 32-bit IEEE 754 standard, and 32-bit integer data, are widely used. Although the convention of register pairs is widely recognized, agreement on whether the high order or low order register should come first is not standardized. For this reason, many devices, including all Control Solutions gateways, support register "swapping". This means you simply check the "swapped" option (aka "High reg first" in some devices) if the other device treats wide data in the opposite order relative to Control Solutions default order.

Control Solutions Modbus products all default to placing the high order register first, or in the lower numbered register. This is known as "big endian", and is consistent with Modbus protocol which is by definition big endian.

### **Deciphering Modbus Documentation**

Documentation for Modbus is not well standardized. Actually there is a standard, but not well followed when it comes to documentation. You will have to do one or more of the following to decipher which register a manufacturer is really referring to:

a) Look for the register description, such as holding register, coil, etc. If the documentation says #1, and tells you they are holding registers, then you have holding register #1. You also have user friendly documentation.

b) Look at the numbers themselves. If you see the first register on the list having a number 40001, that really tells you register #1, and it is a holding register. This form of notation is often referred to as the old Modicon convention.

c) Look for a definition of function codes to be used. If you see a register #1, along with notation telling you to use function codes 3 and 16, that also tells you it is holding register #1.

**IMPORTANT:** Register 1 is address 0. Read on...

d) Do the numbers in your documentation refer to the register number or address? Register #1 is address zero. If it is not clear whether your documentation refers to register or address, and you are not getting the expected result, try plus or minus one for register number. All Control Solutions products refer to register numbers in configuration software or web pages. However, some manufacturers document their devices showing address, not register numbers. When you have addresses, you must add one when entering that register into configuration software from Control Solutions.

### **Can I put 2 gateways on the same Modbus network?**

You can not have more than one Master on a Modbus RTU (RS-485) network. Therefore, if the gateway is to be configured as the Master, you can only have 1 gateway. You cannot use multiple gateways to read more points from the same Modbus slave device.

Multiple gateways configured as slaves can reside on the same Modbus RS-485 network.

If you are using RS-232 devices, you can have only two devices total, regardless of how they are configured. RS-232 is not multi-drop.

### **How many devices can I have on a Modbus RTU network?**

Logically you can address over 250 devices; however, the RS-485 transceivers are not capable of physically driving that many devices. Modbus protocol states that the limit is 32 devices, and most RS-485 transceivers will agree with this. Only if all devices on the network have low load transceivers can you have more than 32 devices.



## Appendix C      Trouble Shooting

### C.1      Modbus RTU Trouble Shooting

You will find message and error counters listed on the Error Counts (see section 6). There is also the packet log where you can see what messages are passing back and forth.

The most frequent problem is "no response" or timeout. This means the master and slave are not connecting for any of several possible reasons: (a) There is a wiring problem; (b) Port parameters are not configured the same (baud rate, etc); (c) Master's timeout setting is too short.

When it comes to wiring, remember that RS-458 is NOT truly a 2-wire interface as it is commonly referred to. Refer to the RS-485 FAQ under Support at csimn.com if you have questions or concerns about wiring.

If you are getting CRC errors, that is almost always a wiring problem, but can be a port problem such as mismatched parity setting.

If you are getting exception errors, that is somewhat good news - it means that at least you are successfully communicating. The exception code itself will provide a clue as to the problem.

### C.2      Modbus TCP Trouble Shooting

Since there is always a one to one correlation between TCP message and RTU message in this type of gateway, the Error Counts page applies regardless of source and destination of the message. You will find message and error counters listed on the Error Counts (see section 6). There is also the packet log where you can see what messages are passing back and forth.

The most frequent problem is "no response" or timeout. The most common cause of this problem for Modbus TCP is a network configuration problem, such as incorrect IP address or IP address that cannot be reached as configured. The problem sometimes lies outside the Babel Buster and may require consulting with the IT personnel responsible for the network if on a large network.

If you are getting exception errors, that is somewhat good news - it means that at least you are successfully communicating. The exception code itself will provide a clue as to the problem.

### C.3 Wireshark Hardware Requirements

There are no particular hardware requirements regarding the PC you run Wireshark on. Basically anything running any version of Windows can run Wireshark. There are also Linux and Mac versions.

The "hardware requirement" that is of most concern is the means of connecting to the network. We typically just connect everything Ethernet to a switch and don't worry about it. However, switches are really unmanaged routers, and they filter traffic. Therefore, your PC will not see traffic passing back and forth between two other devices that are not the PC. In order to see that network traffic using Wireshark, you need to come up with the right kind of network connection.

If your PC itself is one end of the network conversation you wish to capture, for example when running the MIB Browser, then Wireshark will capture all network traffic to and from the PC however connected. It is when your PC wants to simply "eavesdrop" that you run into problems with the network switch.

A while back, 10BaseT hubs were common. A 10BaseT hub is not as smart as a switch and does not filter traffic. If you have an old 10BaseT hub collecting dust somewhere, you now have a new use for it. It will let Wireshark see all traffic from the PC that goes between any other devices connected to that 10BaseT hub. Beware of devices that call themselves "hubs" but support 100BaseT connections. These are switches.

Since manufacturers of hubs decided nobody should have a use for them anymore, they are generally out of production. Finding a 10BaseT hub for sale is not easy (try eBay). But there are other alternatives.

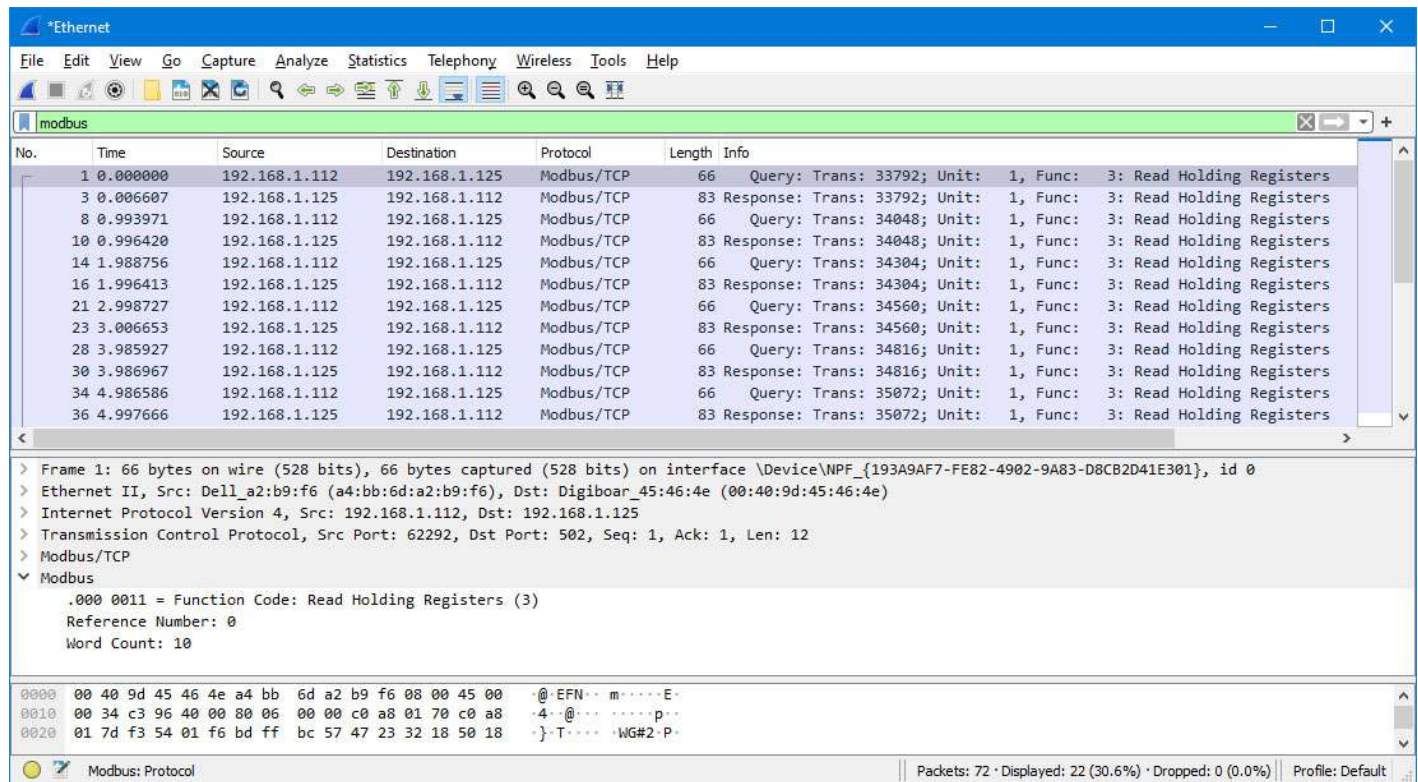
One means of monitoring network traffic is to get a managed switch that supports "port mirroring". One such device we have tested is the TP-LINK model TL-SG105E. Setting it up requires utility software (provided with the switch) and takes a little effort to get configured. But once configured, it works well without any further monkeying around. And it is inexpensive.

The other means of monitoring traffic is with the use of a device made specifically for use with Wireshark. The "SharkTap" provides two connections for the network pass-through, and a third "tap" connection where you connect your PC running Wireshark. There is no configuration required. It is the simplest way to monitor network traffic, and it is a current production item available on Amazon (as of 2021).



## C.4 Example of Using Wireshark

If you use Wireshark to capture Modbus TCP traffic, and set the filter to "modbus", your display will be something similar to what is pictured below. Click on any request or reply, and you can expand the interpretation of the Modbus message in the middle section of the screen.



The image shows a Wireshark capture of Modbus TCP traffic. The packet list on the left shows 36 packets, alternating between queries and responses. The packet details pane on the right shows the expanded view of the first packet (No. 1), which is a Modbus/TCP query. The packet structure is as follows:

- Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF\_{193A9AF7-FE82-4902-9A83-D8CB2D41E301}, id 0
- Ethernet II, Src: Dell\_a2:b9:f6 (a4:bb:6d:a2:b9:f6), Dst: Digiboar\_45:46:4e (00:40:9d:45:46:4e)
- Internet Protocol Version 4, Src: 192.168.1.112, Dst: 192.168.1.125
- Transmission Control Protocol, Src Port: 62292, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
- Modbus/TCP
  - Modbus
    - .000 0011 = Function Code: Read Holding Registers (3)
    - Reference Number: 0
    - Word Count: 10

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII:

```
0000 00 40 9d 45 46 4e a4 bb 6d a2 b9 f6 08 00 45 00  @EFN...m....E.
0010 00 34 c3 96 40 00 80 06 00 00 c0 a8 01 70 c0 a8  4..@...p..
0020 01 7d f3 54 01 f6 bd ff bc 57 47 23 32 18 50 18  }-T...WG#2-P.
```

The status bar at the bottom indicates: Modbus: Protocol, Packets: 72 · Displayed: 22 (30.6%) · Dropped: 0 (0.0%) · Profile: Default





## Appendix D SSL Certificates for Secure Web (HTTPS)

The secure web server (HTTPS) requires SSL certificates in order to establish secure connections. The HTTPS certificates are only required if HTTPS is enabled on the Network configuration page in the Babel Buster BB3-6101/MX-61.

### D.1 X.509 Auto-Certificate Generation

The Babel Buster BB3-6101/MX-61 Gateway will automatically generate X.509 certificates if no external certificates are found or could not be loaded correctly. These will be generated one time and saved in the Flash file system for subsequent reuse. When the self-generated X.509 certificates are in use, this will be indicated at the bottom of the Network configuration page.

The screenshot shows a network configuration interface with a dark teal background. At the top, it says 'Web Server' followed by two checked checkboxes: 'HTTPS Enabled (on 443)' and 'HTTP Enabled'. Below this, there are two input fields: 'HTTP Port' with the value '80' and '(default 80)', and 'Modbus Port' with the value '502' and '(default 502)'. To the right of these fields is a 'Set Ports' button. Below the ports, it says 'FTP Server' followed by a checked checkbox and the word 'Enabled'. At the bottom left, it shows 'MAC Address: 00:40:9D:45:46:4E'. At the bottom right, it shows 'System Uptime: 0,01:23:30'. At the very bottom, it says 'HTTPS certificate status: Using self-generated X.509'.

If there is a need to delete the self-generated certificates, you can do so by logging in via FTP. Change directory to /FLASH0, then to .cfg. The two certificate files that were self-generated are ssl.cert and ssl.key.

```

C:\Users\Jim Hogenson\My Documents\config files>ftp
ftp> open 192.168.1.120
Connected to 192.168.1.120.
220 NET+OS 7.5.2.2 FTP server ready.
User (192.168.1.120:(none)): root
331 User root OK, send password.
Password:
230 Password OK.
ftp> cd /FLASH0
250 Directory is changed
ftp> dir .cfg
200 PORT command Ok.
150 File Listing Follows in ASCII mode
-rwlrwl--- 1 noone      group2 447      Dec 31 1969 ssl.cert
-rwlrwl--- 1 noone      group2 465      Dec 31 1969 ssl.key
226 Transfer complete.
ftp: 119 bytes received in 0.11Seconds 1.09Kbytes/sec.
ftp>

```

## D.2 External Certificates

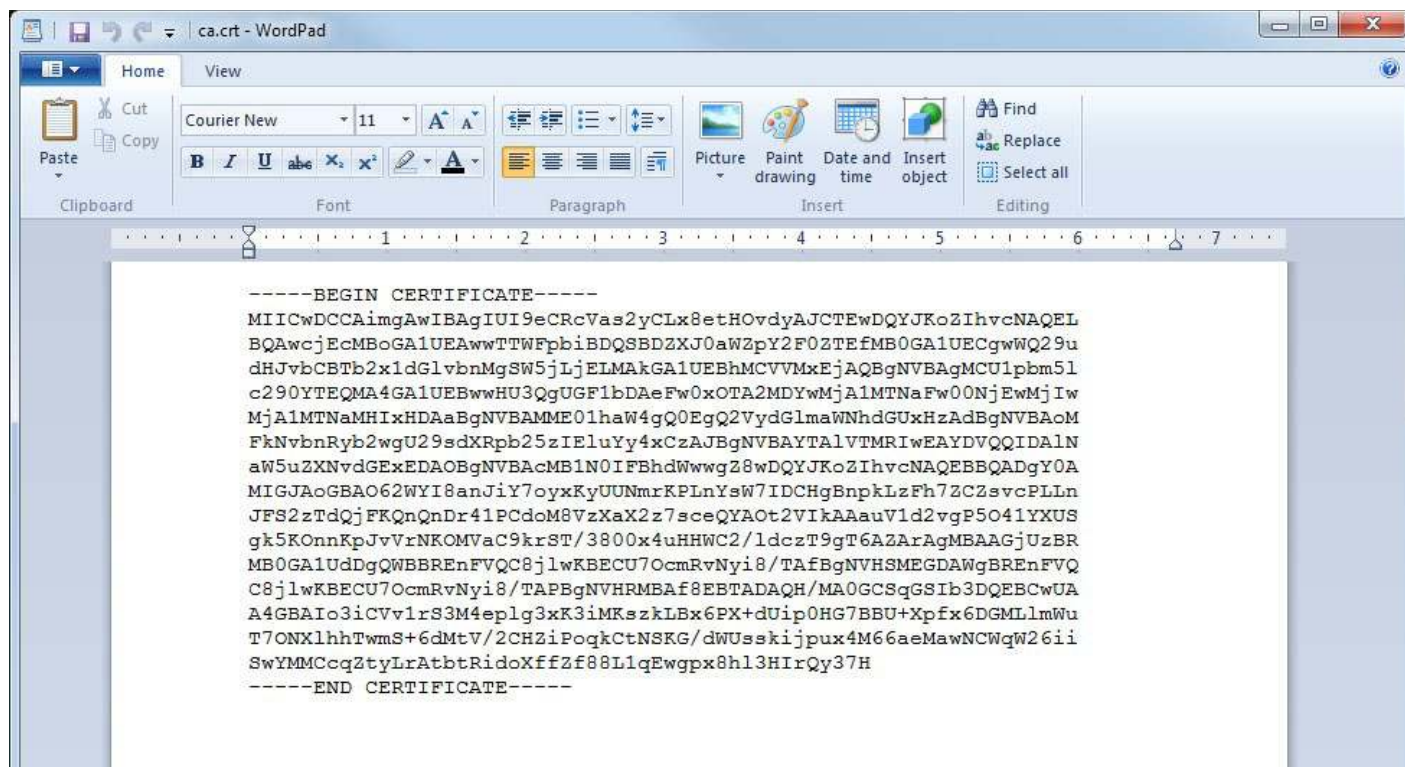
There are three certificates that you must generate and upload to use SSL certificates other than the self-generated X.509 certificates.



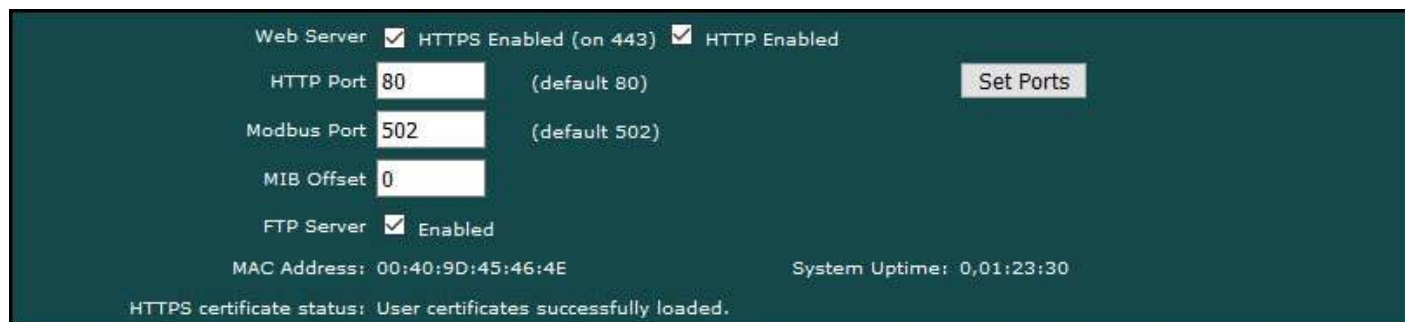
The required certificates are as follows, and must use exactly these names.

ca.crt	CA Root certificate in PEM format
server.crt	Server certificate in PEM format
server.key	Server private key in PEM format

The content of each certificate file will look something like the screen shot below. If you require external certificates for your secure web server, the requirement was likely imposed by your IT department. They should be able to provide the necessary certificates for you. For globally accessed use, the Root CA would come from somebody like GoDaddy or DigiCert (formerly Symantec).



If external certificates were loaded successfully, that will be indicated at the bottom of the Network configuration page.



### D.3 Certificate Generation Script (Linux)

The art and science of generating SSL certificates is beyond the scope of this document. An example SSL certificate generation script is provided here as a reference.

The following script, run on a Linux system with OpenSSL installed, will generate the three required SSL certificate files. It will generate a number of intermediate files as well - you don't need to upload them. Replace references to Control Solutions in this script with your own company name.

```
#!/bin/bash
echo hello
# This will create some self signed certs, using one master CA.
#
# these can be the webserver DNS name, or an IP address, however you
access
```

```

# the resource, this needs to match.
if [ -z "$1" ] || [ -z "$2" ]; then
echo 'Usage: gen.sh <server-name> <client-name>'
echo ' <server-name> and <client-name> can be IP addresses'
echo ' or DNS names.'
exit 1
fi
SNAME=$1
CNAME=$2
#
# Bits for strength, 1024, 2048, 4096, etc.. (suggest 2k or 4k for web
servers)
BITS=1024
#
# HASH - Options are sha256, sha512, sha1, md5
HASH="sha256"
SN=`date +%Y%m%d%H%M%S`
#####
# below is the entry for the CRL
# Do not use http://www.csimn.com/crl.pem for production keys and
certificates
# cat <<EOF >> extensions.cnf
# [ extensions_section ]
# crlDistributionPoints = URI:http://www.csimn.com/crl.pem
#
# basicConstraints = CA:FALSE
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment
# subjectAltName = DNS:${SNAME},IP:${SNAME}
# EOF
#####
#####
# first, lets generate some private keys...
openssl genrsa -out server.key ${BITS}
openssl genrsa -out client.key ${BITS}
# ok, and now the MAIN CA
openssl req -x509 -${HASH} -nodes -days 10000 -newkey rsa:${BITS} -keyout
ca.key -out ca.crt -subj "/CN=Main CA Certificate/O=Control Solutions
Inc./C=US/ST=Minnesota/L=St Paul"
#####
#
# Create a CSR for both server and client
# Replace these values with one appropriate for your organization
openssl req -out server.csr -key server.key -new -subj "/CN=${SNAME}
/O=Control Solutions Inc./C=US/ST=Minnesota/L=St Paul"
openssl req -out client.csr -key client.key -new -subj "/CN=${CNAME}
/O=Control Solutions Inc./C=US/ST=Minnesota/L=St Paul"
#
#
#####
# Sign the keys with the CA
openssl x509 -req -days 3650 -in server.csr -CA ca.crt -CAkey ca.key
-set_serial ${SN}01 -out server.crt -${HASH}
openssl x509 -req -days 3650 -in client.csr -CA ca.crt -CAkey ca.key

```

```
-set_serial ${SN}02 -out client.crt -${HASH}
# Create a windows file to import the client keys if needed in this
format
openssl pkcs12 -export -clcerts -in client.crt -inkey client.key -out
client.p12
# Create the client keys as a complete pem file if needed in this format
openssl pkcs12 -in client.p12 -out client-full.pem -clcerts
# mv -f server.key svrkey.pem
# mv -f server.crt svrcert.pem
# mv -f client.key clntkey.pem
# mv -f client.crt clntcert.pem
# cp -f ca.crt cacert.pem
####
# cleanup
# rm -f client.csr server.csr
#DLS 20160420
echo '*****'
echo '* WARNING: Do not use this script to generate production *'
echo '* keys and certificates. This script is for *'
echo '* demonstration purposes only. *'
echo '*****'
```