mCore®SDR
Operations Manual

mCore® SDR

Thank you for purchasing **mCore®SDR** (Secure Data Router). Your device comes preconfigured and ready to install.

Prior to use, thoroughly read the instructions in this manual to connect and use this product correctly. Please retain this manual for future reference and make sure that this manual is available to all users. To ensure the safety and proper operation of the device and any connected equipment, installation or relocation should be performed by qualified personnel only.

Table of Contents

Section 1: Safety Precautions 4
Section 1.1: Warnings 5
Section 2: Introduction 6
Section 2.1: LED Indicator Table 7
Section 3: Installation 8
Section 3.1: Mounting 9
Section 3.2: Wiring 11
Section 3.2.1: Special Note : Required for Terminating Resistors 14
Section 3.2.2: Wiring Kit 15
Section 3.3: Grounding 17
Section 4: Parts 18
Section 4.1: Parts Table 18
Section 5: Power Requirements 19
Section 6: User Interface 20
Section 6.1: Connecting To The User Interface 21
Section 6.2: Setting Computer's Network Adapter 22
Section 6.3: Change Password 26
Section 6.4: Account Sign Off 27
Section 6.5: General Setting 27
Section 6.6: Firewall 28
Section 6.7: To change the IP address and related settings 30
Section 7: Web Server 31
Section 7.1: mDNS/LLMNR Support 31
Section 7.2: SSL and HTTP Redirect 32
Section 7.3: Upload SSL Certificate 33
Section 7.4: External IP and External Name 34
Section 8: Protocols 35
Section 8.1: Adding A Protocol 35
Section 8.2: CDL 37
Section 8.2.1: CDL Traffic Log 39
Section 8.3: Modbus Setup 40

Click to Return to Top of Document

Section 8.4: SAE J1939 Setup 43
Section 8.4.1: Connections 45
Section 8.4.2: J1939 Traffic Log 48
Section 8.5: OMF 48
Section 8.6: AB Tag Client 52
Section 8.7: MQTT Protocols 56
Section 8.7.1: MQTT-Generic 56
Section 8.7.2: MQTT Amazon Web Services 61
Section 8.7.3: MQTT-Sparkplug B 65
Section 8.8: Configuring mCoreSDR to use Azure IoT Hub 70
Section 9: Live Data Viewer 76
Section 10: Internal Tags 78
Section 10.1: Function Table 80
Section 11: Customer Logging 81
Section 11.1: Export Customer Logs 83
Section 12: ACI eRCM™ Express Configuration 84
Section 12.1: Adding License File 84
Section 12.2: Viewer File(s) 85
Section 12.3: Modbus TCP Slave – PLC Setup and Sequence of Operations – Version 1.0 and 1.1 Legacy 87
Section 12.4: Modbus TCP Slave – PLC Setup and Sequence of Operations – Version 1.1 88
Section 12.5: AB Tag Client - PLC Setup and Sequence of Operations 88
Section 13: System Functions 90
Section 13.1: Commit All Changes 90
Section 13.2: Restore Last Commit 91
Section 13.3: Export User Data 92
Section 13.4: Import User Data 94
Section 13.5: Clear User Data 96
Section 13.6: Export System Logs 97
Section 13.7: Set Device Time 99
Section 13.8: Update Firmware 100
Section 14: Factory Reset 103

SECTION 1: SAFETY PRECAUTIONS

Thoroughly read and follow the safety precautions and operating instructions listed in this manual before using the product.

- Do not use or mount the product in any manner or location not intended.
- After reading, retain this manual for future reference.
- This manual should be available for anyone operating, repairing or relocating the product.
- The product should be installed, repaired, or removed by qualified personnel only.
- Do not disassemble or modify the product.
- Improper installation or repair may cause injury, damage, electric shock or fire.
- To ensure proper operation and avoid damage to the product, use appropriate tools along with recommended connectors and recommended wire gauge.
- Do not open or break the seal between the housing and the connector header.
- Do not operate a unit that has been damaged.
- Extended environmental conditions: wet location, outdoor use, ambient temperature - 40° C to +70° C (at or below 28 VDC input), and altitude up to 2000m.

Click to Return to Top of Document

SECTION 1.1: WARNINGS

⚠ **WARNING – EXPLOSION HAZARD. DO NOT CONNECT OR DISCONNECT WHEN ENERGIZED.**

AVERTISSEMENT – RISQUE D'EXPLOSION. NE PAS BRANCER OU DÉBRANCHER LORSQUE KE CIRCUIT EST SOUS TENSION.

⚠ **WARNING – EXPLOSION HAZARD. DO NOT DISCONNECT WHILE THE CIRCUIT IS LIVE OR UNLESS THE AREA IS FREE OF IGNITABLE CONCENTRATIONS.**

AVERTISSEMENT – RISQUE D'EXPLOSION. NE PAS DÉBRANCHER SI LE CIRCUIT EST SOUS TENSION, À MOINS QUE LE MILIEU SOIT LIBRE DE SUBSTANCES INFLAMMABLES CONCENTRÉES.

⚠ **WARNING – EXPLOSION HAZARD. DO NOT REMOVE OR REPLACE BATTERIES OR PLUG-IN MODULES (AS APPLICABLE) UNLESS POWER HAS BEEN DISCONNECTED OR THE AREA IS FREE OF IGNITIBLE CONCENTRATIONS.**

AVERTISSEMENT – RISQUE D'EXPLOSION. NE PAS RETIRER OU REMPLACER DES BATTERIES OU DES MODULES DE PLUG-IN (COMME APPLICABLES) À MOINS QUE LA PUISSANCE A ETE DÉCONNECTEE OU LA ZONE EST LIBRE DE CONCENTRATIONS IGNITIVE.

⚠ **WARNING – DO NOT REMOVE THE AMPSEAL CONNECTORS UNLESS THE EQUIPMENT NEEDS TO UNDERGO REPAIR OR MAINTENANCE. AFTER REPAIR OR MAINTENANCE HAS BEEN DONE, PLEASE CHECK THE BARE PINS ON THE EQUIPMENT TO ENSURE THERE IS NO DUST OR WATER PRESENT. IP RATING OF THE EQUIPMENT IS ONLY MAINTAINED WITH THE AMPSEAL CONNECTORS ATTACHED TO THE EQUIPMENT.**

AVERTISSEMENT - NE RETIREZ PAS LES CONNECTEURS AMPSEAL À MOINS QUE L'ÉQUIPEMENT NE SOIT RÉPARÉ OU ENTRETENU. UNE FOIS LA RÉPARATION OU L'ENTRETIEN EFFECTUÉE, VÉRIFIEZ LES BROCHES NUES SUR L'ÉQUIPEMENT POUR VOUS ASSURER QU'IL N'Y A PAS DE POUSSIÈRE OU D'EAU PRÉSENTE. LA CLASSIFICATION IP DE L'ÉQUIPEMENT EST UNIQUEMENT ENTRETENUE AVEC LES CONNECTEURS AMPSEAL ATTACHÉS À L'ÉQUIPEMENT.

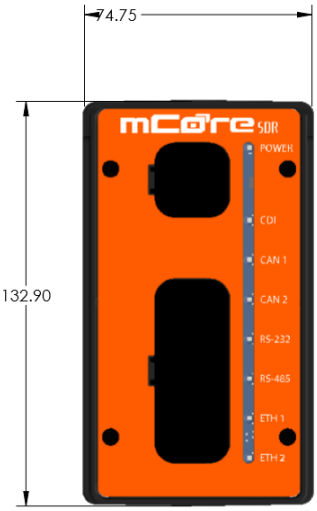
⚠ **WARNING – TO MAINTAIN CLASS 1, DIVISION 2 RATING, THE UNIT MUST BE INSTALLED IN A TOOL-SECURED ENCLOSURE USING ONE OF THE NEC WIRING METHODS THAT IS OUTLINED IN THIS MANUAL.**

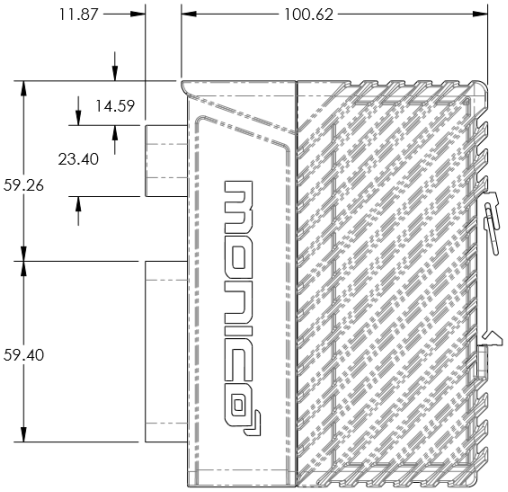
AVERTISSEMENT - POUR MAINTENIR L'INDICE DE CLASSE 1, DIVISION 2, L'UNITÉ DOIT ÊTRE INSTALLÉE DANS UN BOÎTIER SÉCURISÉ EN UTILISANT L'UNE DES MÉTHODES DE CÂBLAGE NEC DÉCRITES DANS CE MANUEL.

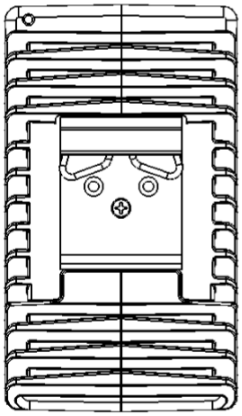
SECTION 2: INTRODUCTION

mCore is designed for industrial applications requiring protocol translation between CDL (CAT® Data Link), SAE J1939, Allen Bradley Tags, Modbus RTU, Modbus TCP, and OMF (OSIsoft® Message Format). Each unit comes to you preconfigured and ready to install.

The mCore is easy to mount, with two mounting options, and environmentally sealed to protect against dust ingress and temporary immersion in up to 1 meter (IP66 & IP67) of water. Designed specifically for heavy-duty industrial applications. The device displays LED indicating lights configured to provide positive confirmation of power, connectivity, and data transmit/data receive.







Weight = 2.0 lbs. (.907 kg)

Click to Return to Top of Document

SECTION 2.1: LED INDICATOR TABLE

	Solid Green	Blinking Green	Solid Amber	Blinking Amber	Solid Red	Blinking Red	LED Off
Power	Powered On, Booted Up, and Ready to Use		Powered On and Booting Up	Key-switch Has Initiated Shutdown or Software Upgrade In Progress	Final Shutdown Sequence and Initial Reboot		Unit Does Not Have Power
CDL		Good Communications	Actively Scanning CDL Bus for Responding Parameters	Indicates Responding Parameters During Scanning	Cable Not Connected, Wired Incorrectly, or CDL Device Is Powered Off		Protocol Not Configured
CAN1		Good Communications			Cable Disconnected or J1939 Device Is Powered Off		Protocol Not Configured
CAN2		Good Communications			Cable Disconnected or J1939 Device Is Powered Off		Protocol Not Configured
Modbus Master (RS-485/232)	Actively Polling But Receiving No Replies	Good Communications			Upon Bootup Only: No Activity On Bus, Cable Not Connected, or Incorrect Mapping For Slave Device	CRC Error	Protocol Not Configured
Modbus Slave (RS-485/232)	Will Be Temporarily Solid When Other Devices On The Network Are Being Polled	Good Communications	No Polls Being Received, Cable Not Connected, Master Device Powered Off, or Incorrect Registers Being Requested			CRC Error	Protocol Not Configured
Ethernet 1	Link Established But Not Actively Communicating	Good Communications			No Link Established or Cable Disconnected		
Ethernet 2	Link Established But Not Actively Communicating	Good Communications			No Link Established or Cable Disconnected		
Factory Reset		Factory Reset Is Complete (All Protocol LEDs)				Factory Reset Initiated (All Protocol LEDs)	

Click to Return to Top of Document

SECTION 3: INSTALLATION

mCore is approved for Class I Div. 2 Group A, B, C, and D. In order to maintain this rating, the unit must be installed in a separate tool secured enclosure and comply with the one of the following NEC wiring methods:

- Extra-hard usage cord – Section 501.140 of the NEC and Rule J18-160 of the CE Code Part I.

- Instrumentation tray cable (Type ITC or CIC) – Section 501.10(B) and 501.105(B)(6) of the NEC and Rule J18-152 of the CE Code Part I.

- Power-limited tray cable (Type PLTC) – Article 725 of the NEC.

- Tray cable (Type TC) installed per Article 336 of the NEC and Rule 12-2202 of the CE Code Part I.

DIN Rail Mount:

The mCore unit should be mounted vertically on a horizontal DIN rail. Tilt the unit to a 45-degree angle and insert the top lip of the DIN rail bracket onto the DIN rail. While pushing down, attach the bottom lip to the DIN rail to snap the unit into place, as illustrated in Figure 1. A minimum of 1" of space should be maintained on all sides of the unit to ensure proper heat regulation. The device's mounting pad is shown below in Figure 2.

To remove the unit from the DIN rail, push down on the unit until the bottom lip is clear from the rail. Then pull out from the bottom. The unit should remove with ease.

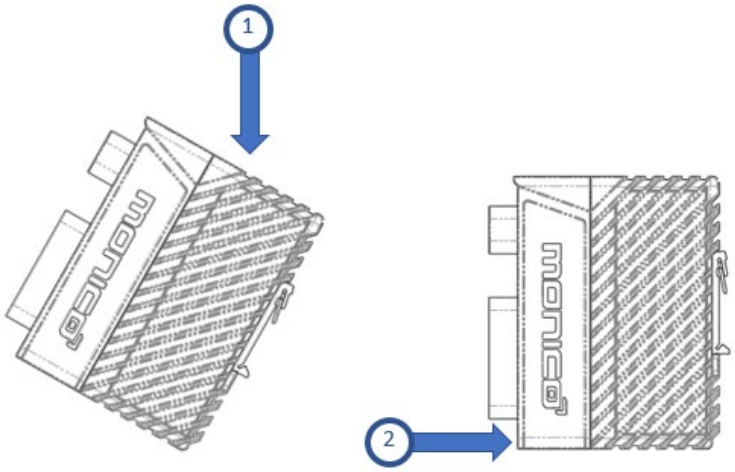


Figure 1

Click to Return to Top of Document

Figure 2

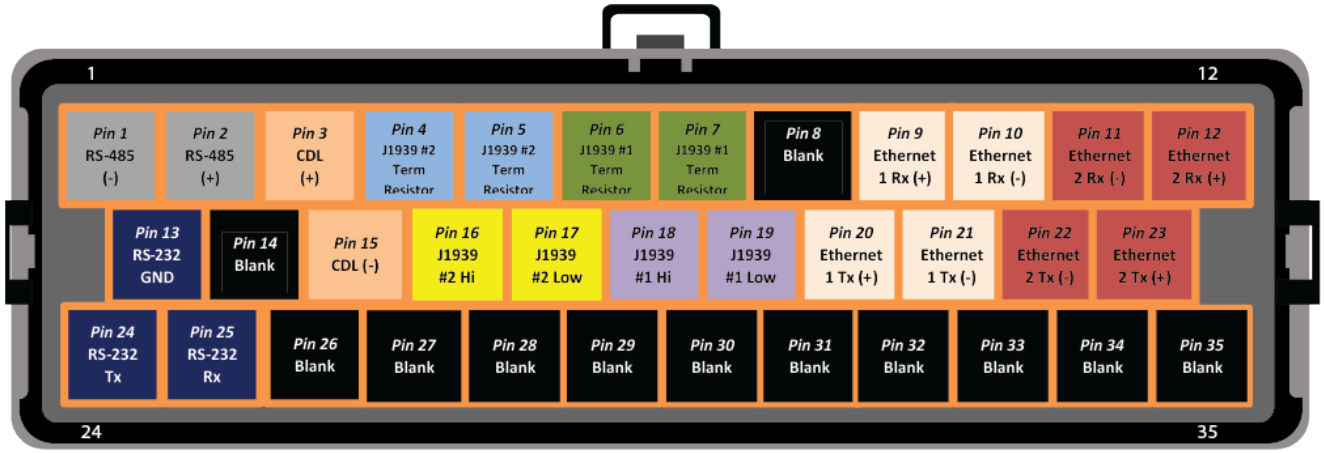
Surface-Mount:

The unit should be mounted securely against a flat surface, using two ¼" fasteners (not provided), in a suitable location as close to the engine as possible. DO NOT mount directly to the engine block. The unit should be mounted, as shown in Figure 3, either horizontally or vertically. Horizonal is the optimal mounting orientation considering LED visibility and heat dissipation. However, other mounting orientations are acceptable.

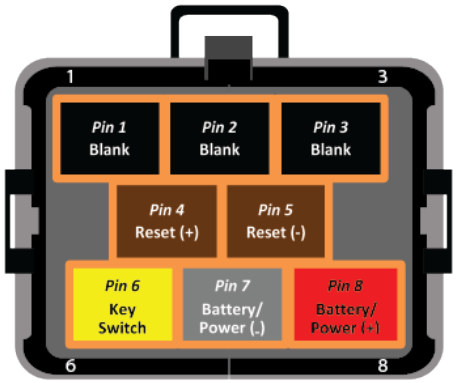
Figure 3

SECTION 3.2: WIRING

The unit should be connected to the desired ports according to the pinout below using the recommended connectors shown in Section 4.1: Parts Table.



Key Connector B



Key Connector A

The above pinout is depicting the female header connectors and respective key orientations on your mCore that will mate to the male connectors. Monico recommends connecting mCore to the batteries when available so that mCore will be booted up before the engine is turned on so it can capture the startup process. mCore takes about 15 seconds to boot up. The mCore's power is supplied through Pins 6, 7, and 8 on Key Connector A, as shown in the Pin Out Guide above. Pin 7 will be connected to Battery/Power (-). Pin 6 is for a Key Switch to be able to power cycle the mCore without removing the connector. Pin 6 must be energized to allow mCore to boot up. When power is removed from Pin 6, mCore will power down. If a Key Switch will not be in line with Pin 6, then connect Pin 6 with Pin 8 to Battery/Power (+).

Be sure to follow the AMPSEAL “How-to Instructions” located at:

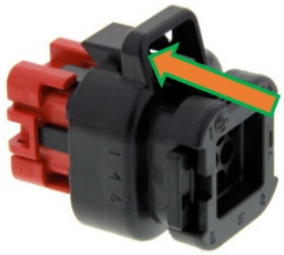
https://www.te.com/usa-en/videos/transportation/ampseal.html

https://www.te.com/usa-en/videos/transportation/ampseal-16-contact-crimping.html

These instructions include steps for proper inserting, crimping, & removing of wires into the pin connectors. For proper hand crimping, an AMPSEAL 2119118-1 hand crimping tool (not included) must be used.

Plugging In:

To connect power to the mCore, push the connector until you hear the click of the locking ear snapping into place. It may require a good amount of force as these connectors are waterproof and have a gasket that must be overcome.



Unplugging:

To disconnect power to the mCore, insert a small flathead screwdriver into the locking ear on the side of Key Connector A and pry outwards gently while pulling on the connector.

SECTION 3.2.1: SPECIAL NOTE : REQUIRED FOR TERMINATING RESISTORS

For all mCore serial numbers requiring a terminating resistor on CAN1, users should follow the pinout guide to enable the CAN1 terminating resistor by installing the terminating jumper between pins 6 and 7 on the 35 pin connector. A terminating resistor jumper is available from Monico if required.

For all mCore's with a serial number < MC-0000-2000, users should follow a modified pinout guide to enable the CAN2 terminating resistor by connecting a jumper between pin 4 (J1939#2 term resistor) and pin 17 (J1939#2 low). A special modified jumper is available from Monico if required.

SECTION 3.2.2: WIRING KIT

Identified below are the wiring components that your mCore®SDR may include depending on your configuration. Please contact Monico for additional or different wiring needs.

SPECIAL NOTE: All Ampseal contacts come preinstalled into the Key Connectors from Monico and the customer will not need to populate. However documentation on the pinouts can be found in section 3.2 Wiring.

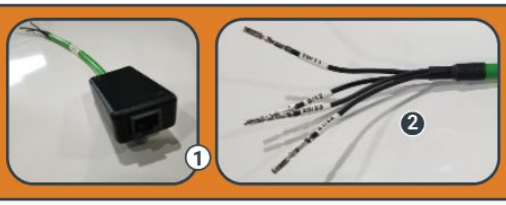
Ⓧ Legend: White backed number is customer end of wiring

Power Wires

1. Includes 3 wires; Key Switch, Positive and Ground
2. Ampseal contact connects to mCore®SDR 8 pin connector (Key Connector A)
3. Ferrule connectors

Key Connector A

Ethernet Dongle



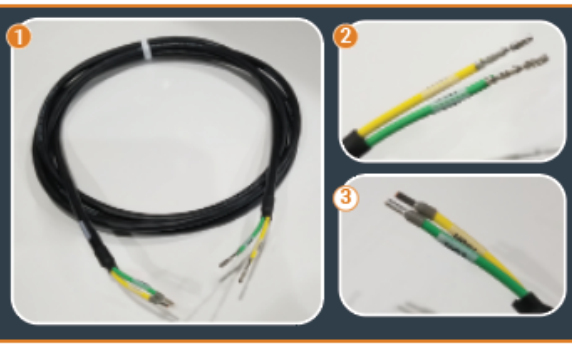
1. Dongle includes RJ45 connection
2. Ampseal contact connects to mCore®SDR 35 pin connector (Key Connector B)

Key Connector B

CDL

1. 10' CDL Cable for connecting to Cat® Data Link port
2. Ampseal contact connects to mCore®SDR 35 pin connector (Key Connector B)
3. Ferrule ends for connecting to CDL

CAN



1. 10' CAN Cable for connecting to J1939 CAN
2. Ampseal contact connects to mCore®SDR 35 pin connector (Key Connector B)
3. Ferrule ends for connecting to CANbus

RS485

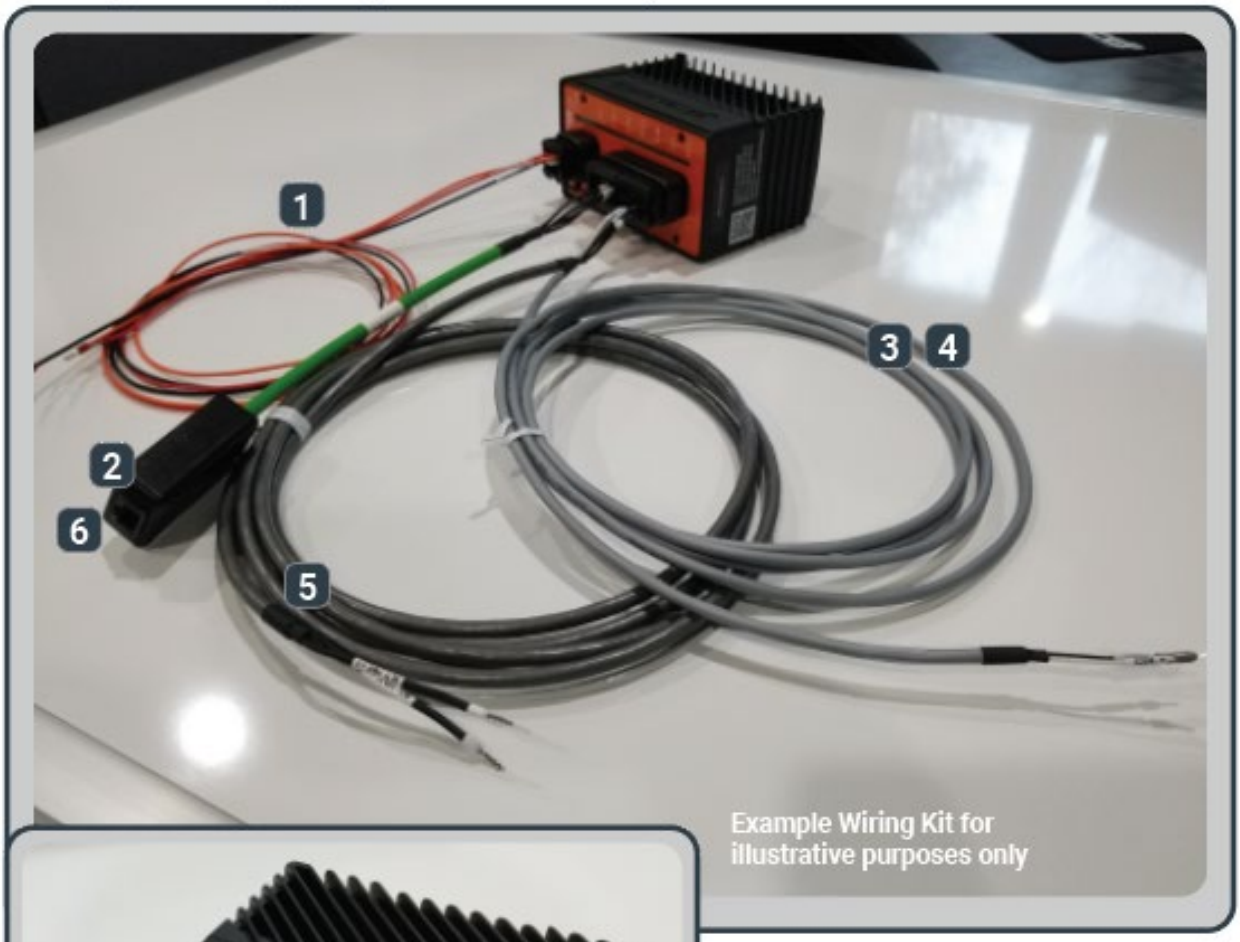
1. 10' RS485 Cable with flying leads
2. Ampseal contact connects to mCore®SDR 35 pin connector (Key Connector B)
3. Ferrule ends for connecting terminal

RS232 Dongle

1. Dongle includes RJ12 connection
2. Ampseal contact connects to mCore®SDR 35 pin connector (Key Connector B)

Click to Return to Top of Document

EXAMPLE WIRING KIT (RS485 and CDL)



1. Power Wires
2. Ethernet Dongle
3. CDL Cable
4. J1939 Cable (Not Depicted)
5. RS485 Cable
6. RS232 Dongle (Not Depicted)

SPECIAL NOTE: There is an optional alternative to flying leads for CDL and RS485 to use a RJ45 Dongle, similiar to the RS 232 Dongle depicted on opposite page

Click to Return to Top of Document

SECTION 3.3: GROUNDING

After mounting and connecting mCore, the unit must be grounded in compliance with local and national electrical codes. It must be externally grounded using a customer-supplied ground wire before any power is applied. Contact the appropriate electrical inspection authority if you are uncertain that suitable grounding is available.

Tools Required:

QTY: 1	Grounding Lug (included)
QTY: 1	6-Gauge Ground Wire (not included)
QTY: 1	Nut Driver (not included)
QTY: 1	Pliers or Crimping Tool (not included)

1. Use the Pliers or Crimping Tool to crimp the 6-Gauge Ground Wire to the Grounding Lug.

2. Connect the Grounding Wire to the mCore's Grounding Lug connection point, shown in Figure 4 below.

3. Place the Grounding Wire over the Grounding Lug and tighten these components using a nut driver. Tighten to 9.6 in-lbs.

4. Connect the other end of the wire to a reliable earth ground if possible. For most effective grounding, use the grounding standards listed below.

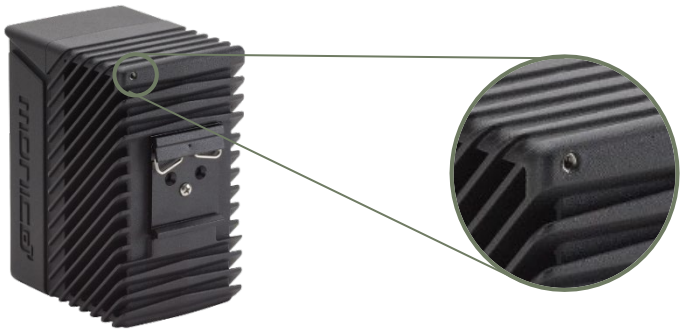


Figure 4

*Safety guidelines for proper grounding are outlined in OSHA Standard 1926.962.

*National standards for proper grounding are outlined in IEEE Standard 142.

Click to Return to Top of Document

SECTION 4: PARTS

SECTION 4.1: PARTS TABLE

Connection Parts (AMPSEAL*)		
Item	**Description**	**AMPSEAL Part Number**
	Plug, 8 Pin, Key Connector A	**776286-1**
	Plug, 35 Pin, Key Connector B	**776164-1**
	Sealing Plug**	**770678-1**
	Back shell, 8 Pin **Back shell, 35 Pin**	**2138529-1** **776463-1**
	Connector	**770854-1**

** Connectors are AMPSEAL brand; available through LADD Industries*

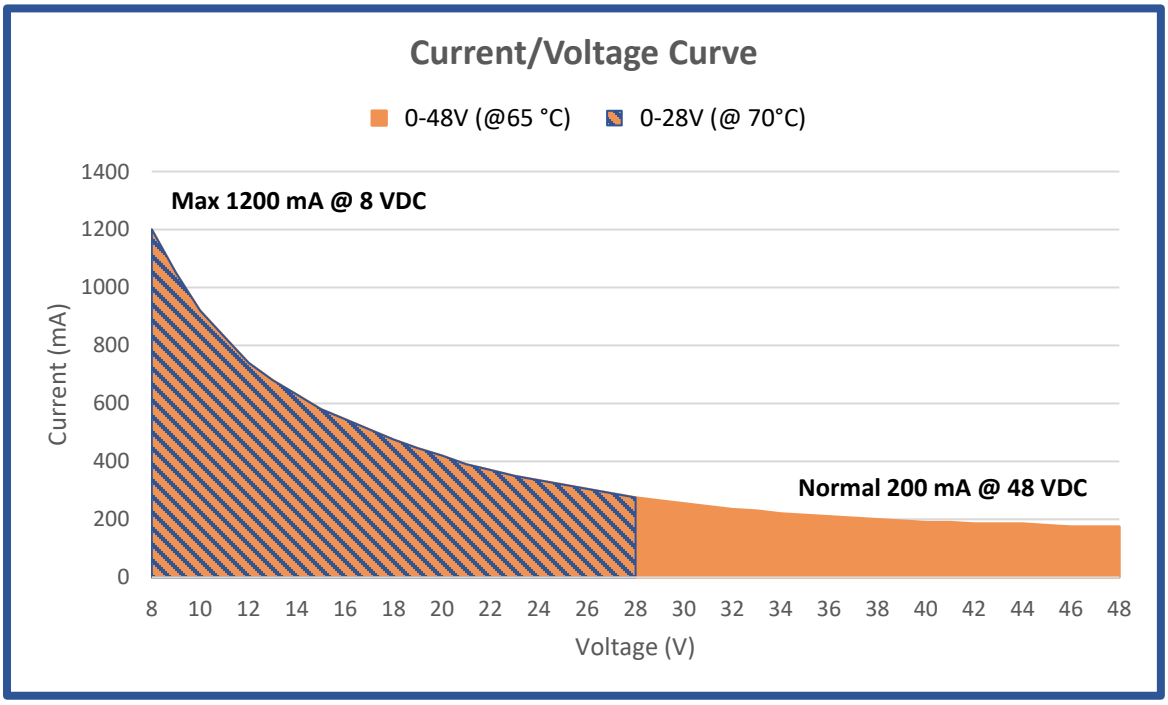
*** Only needed if Connector Seal was punctured due to mis-wiring or pins no longer being used. This will re-seal the connector against water ingress.*

Click to Return to Top of Document

SECTION 5: POWER REQUIREMENTS

The input voltage requirements are 8 – 48 VDC for operating temperatures between -40oC and +65oC. The input voltage requirements are limited to 8 – 28 VDC for operating temperatures between -40oC and +70oC. The mCore has internal reverse polarity protection but will not operate under reverse polarity conditions.

****User is required to install a 2 Amp fast blow fuse rated for at least 50VDC (or the maximum voltage of the supplied power) to protect against short circuit.****



Using the KeySwitch

In addition to the battery positive and battery negative leads, there is a key switch wire that must be wired into the positive terminal of the power source or wired to a switch connected to the positive terminal of the power source.

mCore®SDR will not boot if the key switch is not powered.

IMPORTANT NOTE: In stationary installations where there is NOT a Keyswitch input, connect the Keyswitch wire to the Power +.

Click to Return to Top of Document

SECTION 6: USER INTERFACE

The mCore has a web-based UI (User Interface) for configuring the unit that can be accessed using a web browser. Your mCore User Interface can be accessed via an Ethernet connection. There are two methods for accessing the User Interface for the mCore over an Ethernet connection; one uses a static IP (default configuration) and another that uses DHCP (modified configuration).

Default Configuration: By default your mCore is set for Ethernet 1 to use Static IP, follow the default configuration steps in section 6.1 to access the User Inteface.

Modified Configuration: If, your mCore is configured for Ethernet 1 to use DHCP, use the Modified Configuration steps in section 6.1 to access the User Interface.

On the bottom of the mCore will be a sticker with useful information and a QR code.

The label will contain the following:

- Serial Number
- MAC-1 Address
- MAC-2 Address
- Username
- Password

Special Note about Saving Changes:

STORE CHANGES: ***When making modifications to the User Interface, there are two steps to saving changes. On each page or tab, there is a STORE CHANGES button. This button just saves the changes to the User Interface and are NOT sent to the System level. If you do not click on STORE CHANGES before leaving each page or tab, your changes will be lost.***

Commit All Changes: ***In order to send User Interface changes to the system level, you must use the Commit All Changes on the left-hand menu. When you click on this button, you will be given an opportunity to review any network changes you have made and choose whether to implement these network changes during the commit. You may be working on an offline unit that you do not want to implement these changes. If you do commit these network changes, you may need to make changes to your computer to reach the unit after the commit.***

Click to Return to Top of Document

SECTION 6.1: CONNECTING TO THE USER INTERFACE

Default Configuration:

1. Plug the mCore's Ethernet 1 directly into your computer or network switch using an ethernet cable. Ethernet 1 is configured as a static IP.

2. To connect to a static IP address, you will need to make sure your computer's IP address is setup properly. In either case, your computer will need to be on the same subnet as the mCore target. See Section 6.2 for how to change your computer's IP address.

3. Open an internet browser, enter the default static ip address of 192.168.10.220

4. You should now see the mCore Login page.
    - Default Username: admin
    - Default Password is unique per mCore and will be on the label sticker on the bottom of the mCore

Modified Configuration: (only use if Ethernet 1 is configured as DHCP)

1. Plug the mCore's Ethernet 1 directly into your computer or network switch using an ethernet cable.

2. In order to connect to mCore using DCHP your PC must be setup for DHCP as well. See Section 6.2 for how to access your network adapter settings.

3. Locate the Serial Number written on the label sticker on the bottom of the mCore.

4. Open an internet browser and using the example below as a guide, enter your serial number.

   Example: If the mCore serial number is mc-1234-5678, you would enter, without quotes, "http://mc-1234-5678.local" or "https://mc-1234-5678.local".

5. You should now see the mCore Login page.
    - Default Username: admin
    - Default Password is unique per mCore and will be on the label sticker on the bottom of the mCore.

***Note: We recommend that the user change the username and password after the first login to a secure credentials. See Section 6.3: Change Password for how to do this. It is recommended to use a minimum of 8 characters in length and utilize a combination of capitalized letters, special characters, and numbers. Should the password be forgotten, you can reset mCore back to the password listed on the label sticker by performing a Factory Reset. See Section 14 for instructions.**

SECTION 6.2: SETTING COMPUTER'S NETWORK ADAPTER

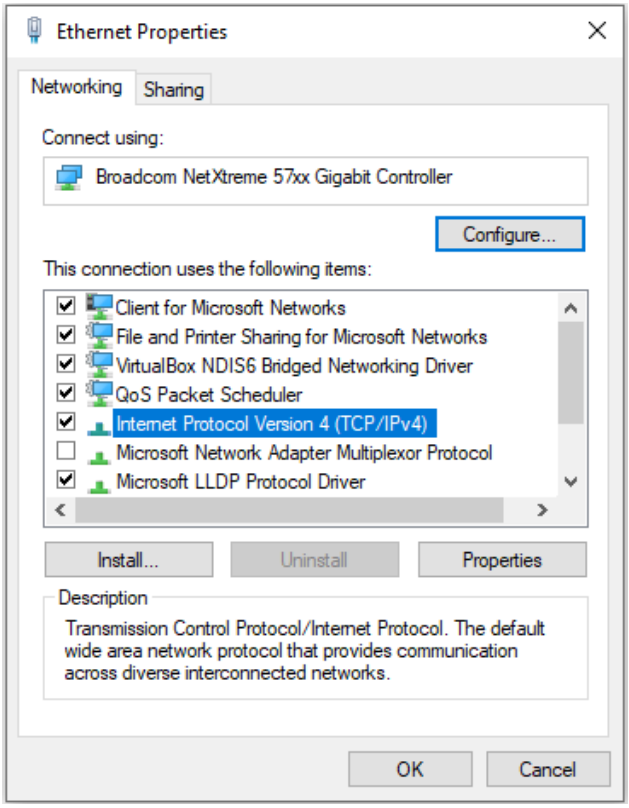
Static IP

The section is only needed if connecting to an mCore that previously had its IP address set to be static instead of DHCP. You will need to change your computer's IP address to be on the same subnet.

1. Select Start, then select Settings > Network & Internet > Ethernet.

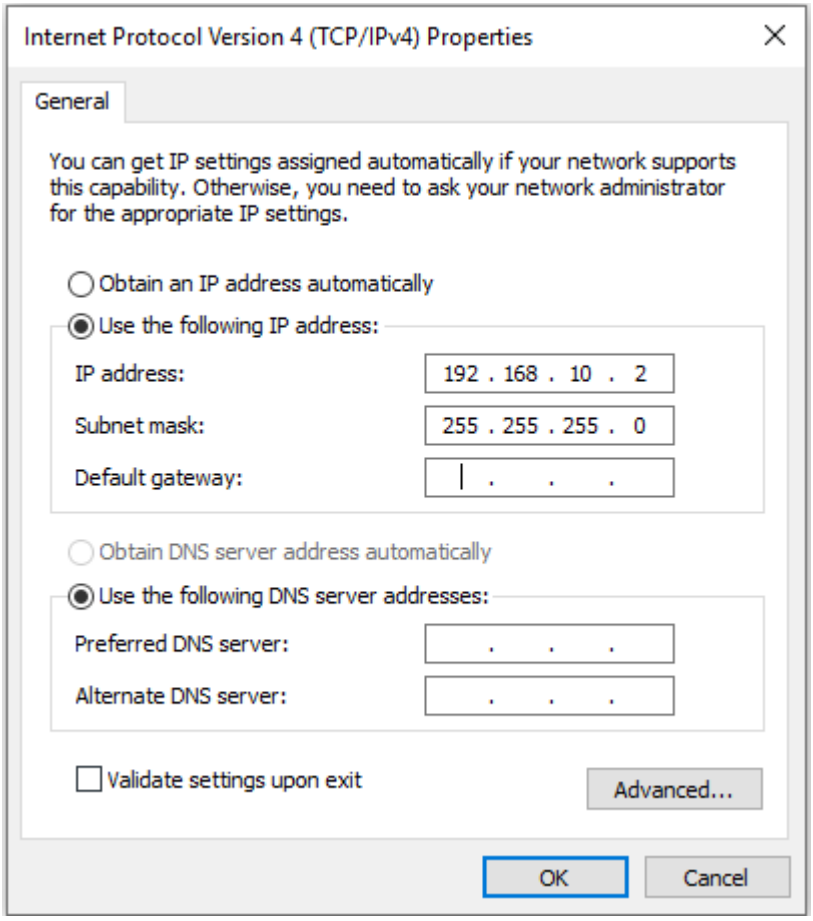
2. Select Change Adapter Options.

3. Select the Ethernet adapter from the list, and Right Click, select Properties.



4. Select Internet Protocol Version 4 (TCP/IPv4), click Properties.

5. Select the "**Use the following IP Address**" option.



6. Enter an IP Address where the last number is different from mCore's, then hit Tab. Example: use the IP depicted above.

7. Enter a Subnet Mask. Typically, it is: 255.255.255.0. Then hit Tab.

8. Leave the Default Gateway blank.

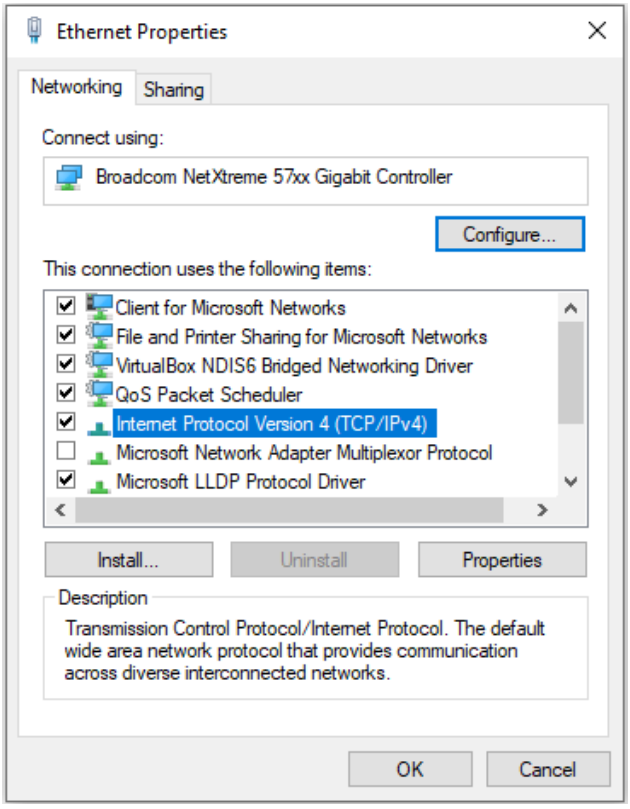
9. Click OK to save, then Close to save.

DHCP Settings

1. Select Start, then select Settings > Network & Internet > Ethernet.

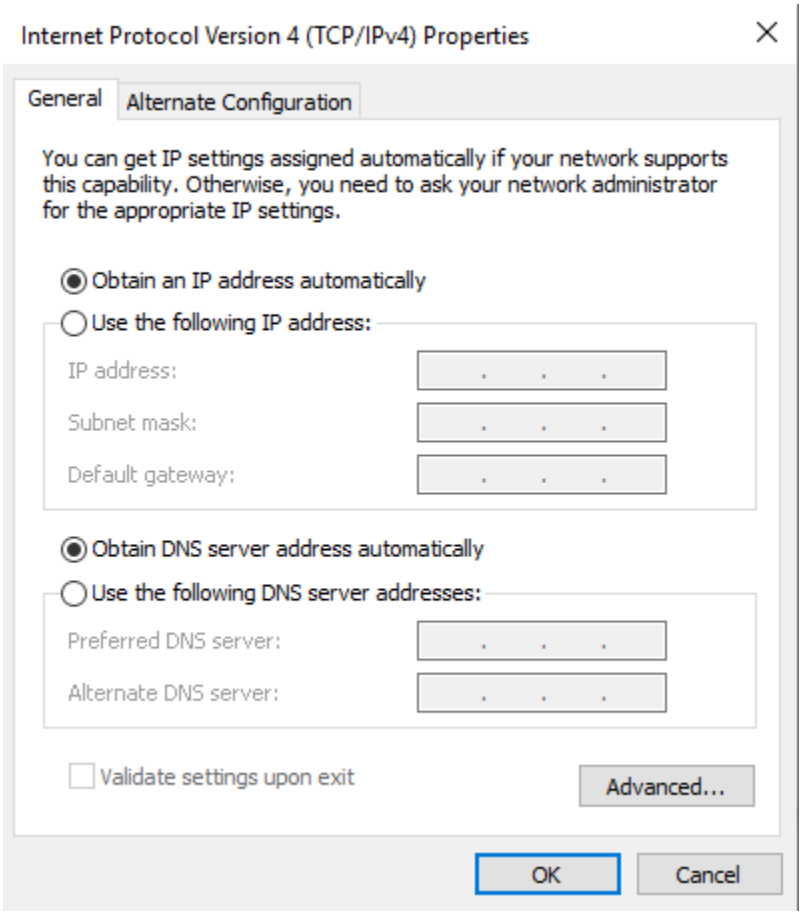
2. Select Change Adapter Options.

3. Select the Ethernet adapter from the list, and Right Click, select Properties.



4. Select Internet Protocol Version 4 (TCP/IPv4), click Properties.

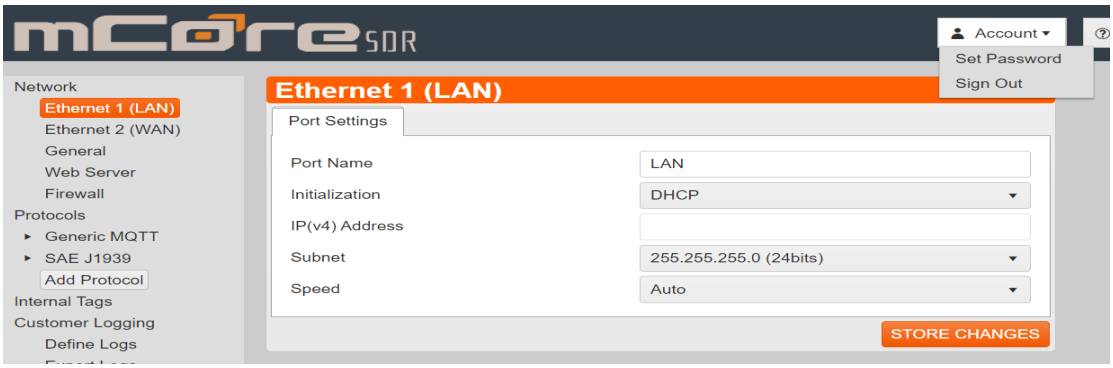
5. Select the "**Obtain an IP address automatically**" option.



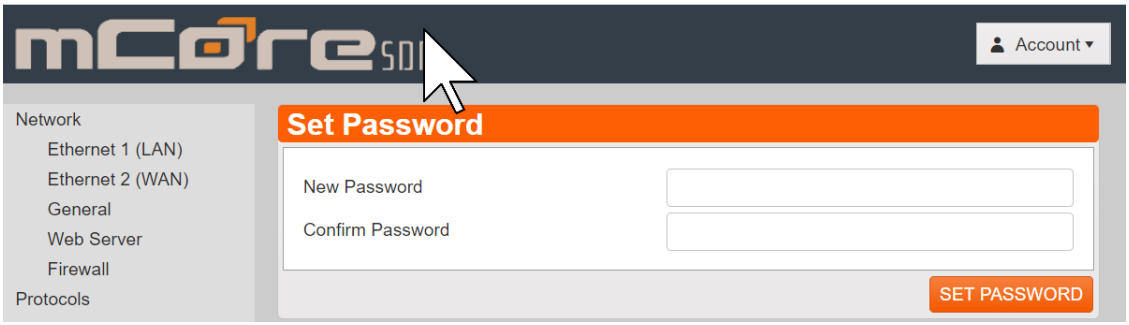
6. Click OK to save, then Close to save.

SECTION 6.3: CHANGE PASSWORD

1. To change the password, click the Account button in the top right, then click Set Password.



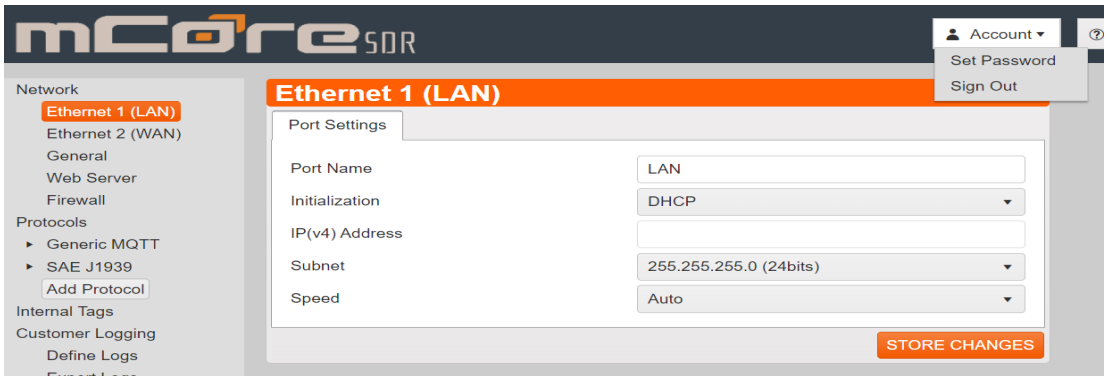
2. The following screen will appear. Enter the new password in each box. Click the Set Password button in the center menu when done.



3. ***The password has now been changed but will not take effect until you either reboot or Sign Out and then log back in.***

SECTION 6.4: ACCOUNT SIGN OFF

Account Sign Off allows you to log off the mCore and requires you to re-enter your login username and password to log back in.



1. To Sign Off, click the Account button in the top right, then click the Sign Out button as shown in the following picture.

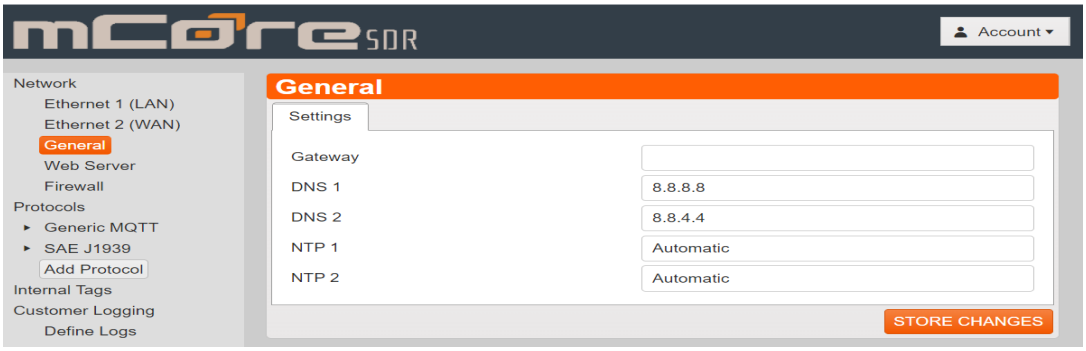
2. After clicking the Sign Out button, a pop box will appear. Click Ok to continue.

3. Once you have clicked Ok, you will be signed out and brought back to the Login screen.

SECTION 6.5: GENERAL SETTING

The General Setting tab is used to set several options:

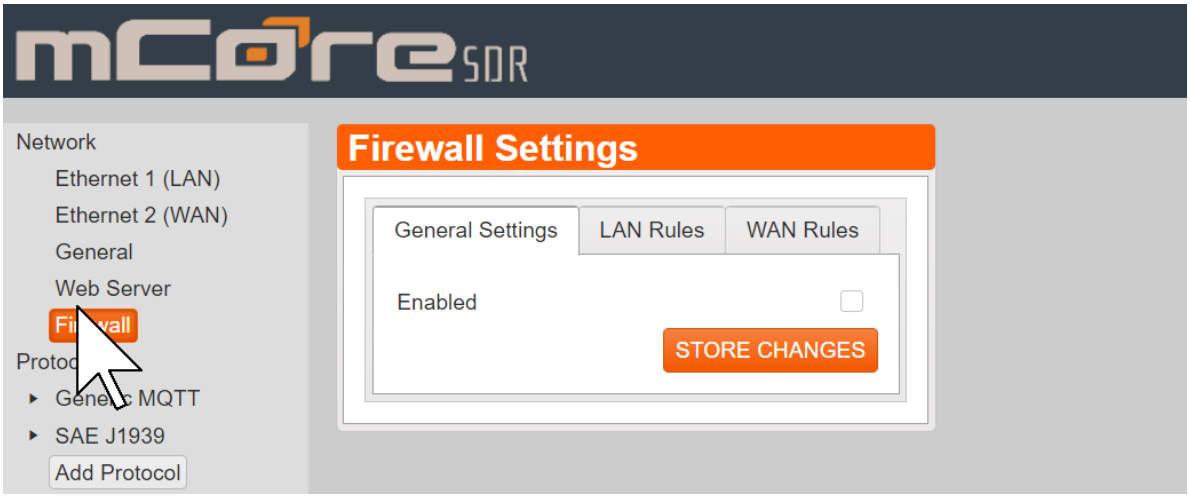
- Gateway: Enter the IP of the default gateway. This is normally the address of the primary router on your network.
- DNS Servers: Enter IP addresses of up to two DNS servers. If your device is not connected to the Internet, these servers must be reachable from mCore.
- Time Servers: Enter the IP addresses or names of up to two NTP servers that mCore will use to set its Real-Time Clock. Leaving both at Automatic will result in mCore using the NIST servers. If your device is not connected to the Internet, these servers must be reachable from the mCore.



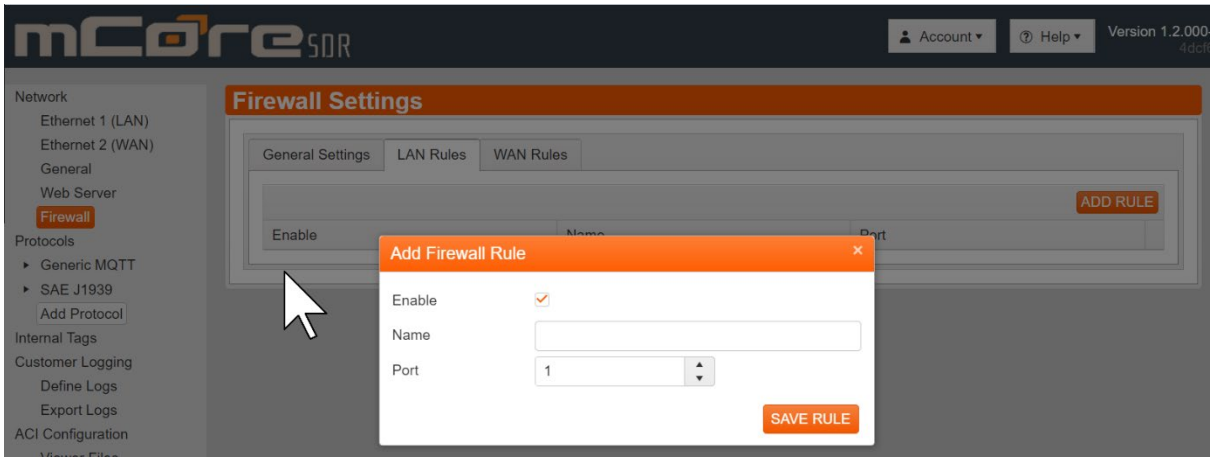
SECTION 6.6: FIREWALL

The mCore's Firewall give the user the ability to limit inbound connections. This is done by limiting what ports can be accessed on the mCore. Once the Firewall is enabled, no incoming connections will be allowed unless you enter a new rule in the Firewall section to allow the connection to a specific port. For example, Modbus TCP usually runs on port 502. If the Firewall is enabled, but no rule entered to allow port 502, then the Modbus TCP communications will not work. The port must be opened on LAN and/or WAN port, depending which network it is on.

1. To Enable the Firewall, click the Firewall button in the left menu towards the top. The following screen will appear. Click the check box and then click Store Changes in the center menu.



2. To enter a new rule, click the LAN rules tab in the center menu. Then click the Add Rule button in the top right of the center menu. The following screen will appear.



3. The Enable check box should be defaulted to checked, if not, check it. Enter a name for the rule and then enter the Port number that you would like allowed an incoming connection to.

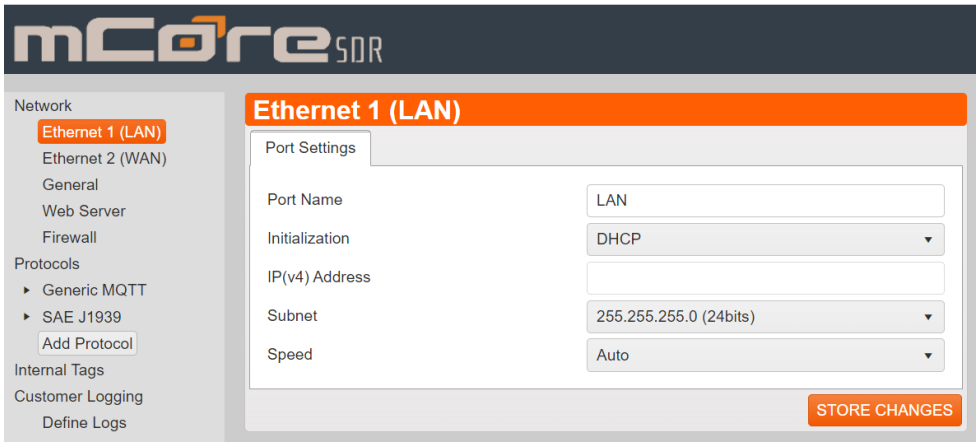
4. When done, click the Save Rule button in the bottom right.

5. Repeat Steps 2 through 4 to add any additional rules required. Follow the same steps for entering rules in the WAN tab as well.

Once you are done entering all rules and ready for them to take effect, click the Commit All Changes button in the left menu towards the bottom. A pop-up box will appear. Click Ok to continue. mCore will then reboot with the rules now taking effect.

SECTION 6.7: TO CHANGE THE IP ADDRESS AND RELATED SETTINGS

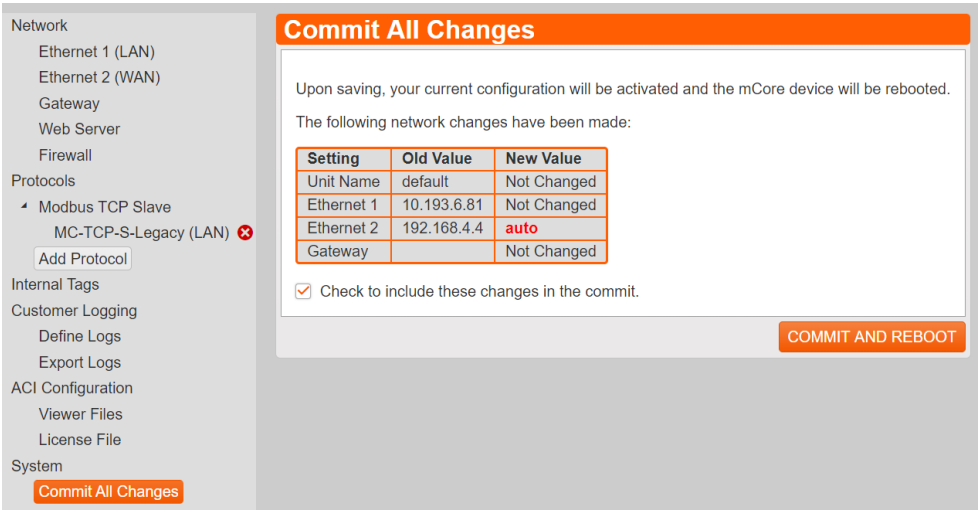
1. Click the Ethernet 1 (LAN) button in the left menu toward the top. The following screen will appear. Change the IP address and Subnet to the desired settings. Click the Store Changes button in the center menu at the bottom right.



2. Continue to Ethernet 2 (WAN) port in the left menu if applicable for your application or leave it as is.

3. The next setting is ***General*** in the left menu. Most sites are set to a default of xxx.xxx.xxx.1 as compared to the IP address. Ask your network administrator to be sure.

4. Once you've entered your IP Address and **stored changes**, you will need to **Commit All Changes** for the new IP address to take effect. Do this by clicking the Commit All Changes button in the left menu toward the bottom. Make sure to check the box to include the new network setting changes. If you do not check the box, the old network settings will stay in effect. Then click the Commit and Reboot button. A pop-up box will appear. Click Ok to continue. mCore will then reboot and the new IP address will now be in effect.



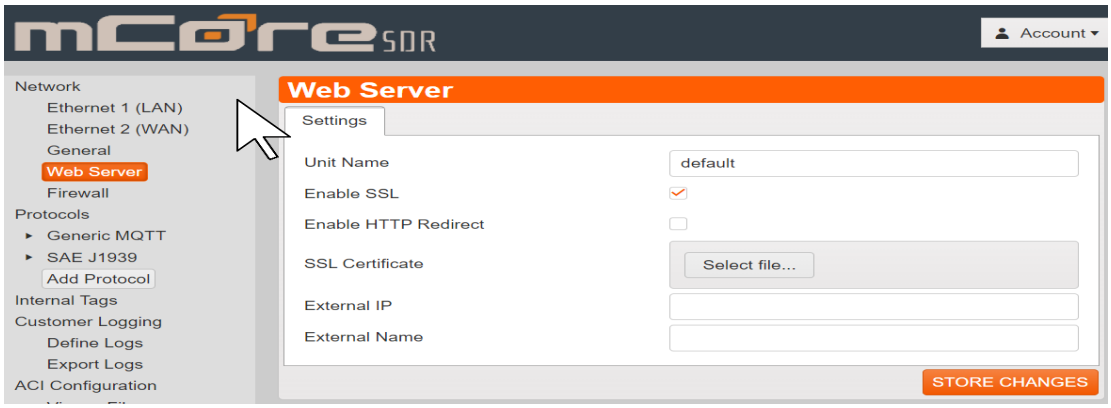
SECTION 7: WEB SERVER

The Web Server, located in the left menu towards the top, provides the following functions: the setup of mDNS and LLMNR support for connecting to the mCore, enable or disable SSL and HTTP redirect, allows the user to upload and use their own SSL certificate, and allows the user to give the mCore an external IP address and name for connecting remotely.

SECTION 7.1: MDNS/LLMNR SUPPORT

mDNS and LLMNR give the user the ability to connect to the mCore's UI (User Interface) via a web browser using its Unit Name/Serial Number or IP Address. The mCore will come with the Unit Name set to the mCore's serial number. For the following example assume the serial number is mc-1234-5678. In your browser's address bar, type in the following without the quotes:
"http://mc-1234-5678.local" or "https://mc-1234-5678.local". This will take you to the Login page. If you are unable to connect, contact Monico Support.

1. The Unit Name will show as Default out of the box, which is the mCore's serial number. To change the Unit Name, click the Web Server button in the left menu toward the top. The following screen will appear.

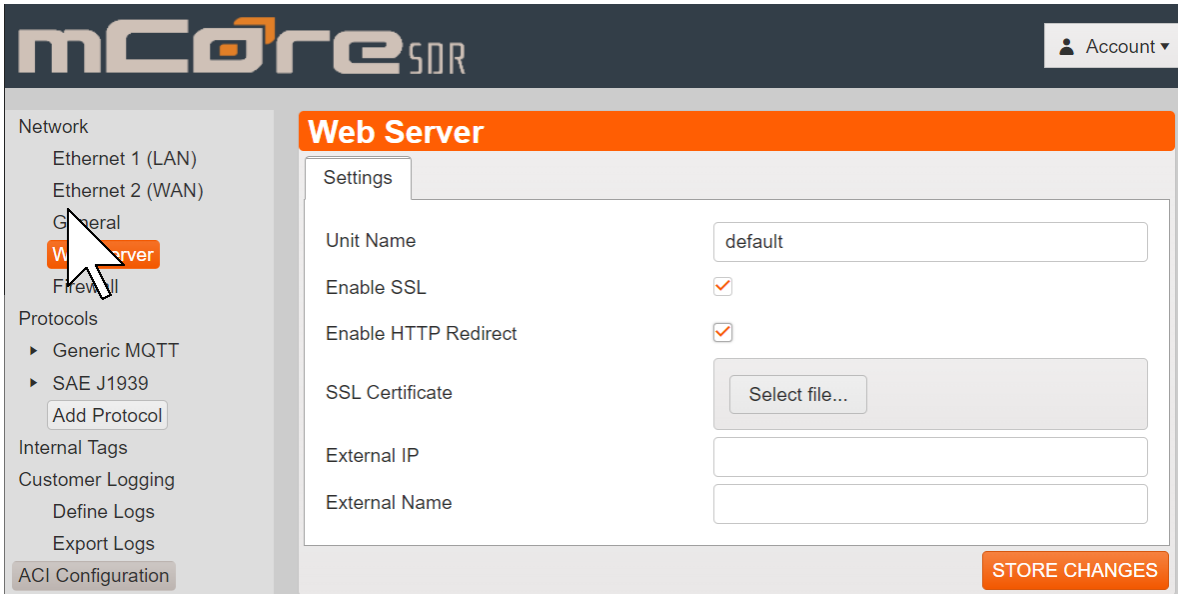


2. Enter the new name in the Unit Name box. Click the Store Changes button in the center menu when done.

3. To activate the changes, you will have to click the Commit All Changes button in the left menu towards the bottom. It will show a pop box. Click Ok to continue. mCore will reboot and the new Unit Name will now be active.

SECTION 7.2: SSL AND HTTP REDIRECT

1. By default, SSL will come enabled. Since the Enable HTTP Redirect is not checked, you will be able to access the UI via both http and https. To disable, uncheck the Enable SSL box. When done, click the Store Changes button in the center menu.



2. If you want your computer to always go to https, you can check the Enable http Redirect check box. So, if you enter http:// to connect to mCore's UI (User Interface), you will be redirected to https://. When done, click the Store Changes button in the center menu.

3. To activate the changes, you will have to click the Commit All Changes button in the left menu towards the bottom. It will show a pop box. Click Ok to continue. mCore will then reboot.

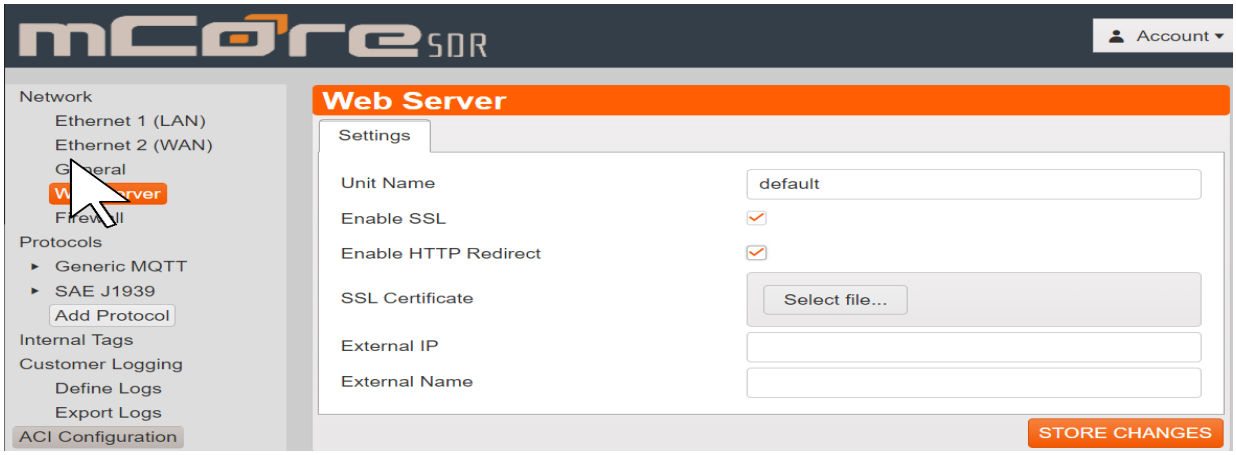
4. Reconnect to the mCore using https:// instead of http:// with either the serial number, the IP address, or the Unit Name if changed from default.

Click to Return to Top of Document

SECTION 7.3: UPLOAD SSL CERTIFICATE

SSL connections require a certificate that proves to the web browser that the unit to which you are connecting is what it claims to be. This is to avoid, for example, you're sending a password to another device that is pretending to be your mCore. You can either provide your mCore with a certificate file manually using the Certificate setting, or you can allow it to generate its own. If you provide a certificate, the file you provide must contain both the certificate itself and the associated private key. If these are in separate files, you can combine them by simply concatenating them into a single text file. If allow the mCore to generate its certificate, it will sign this certificate using its own copy of the Monico Local Root certificate. This means that downloading the Monico Local Root and installing on your PC as a Trusted Root will allow your web browser to accept any certificate created by an mCore device.

1. To upload an SSL Certificate, click the Web Server button in the left menu toward the top. The following screen will appear.



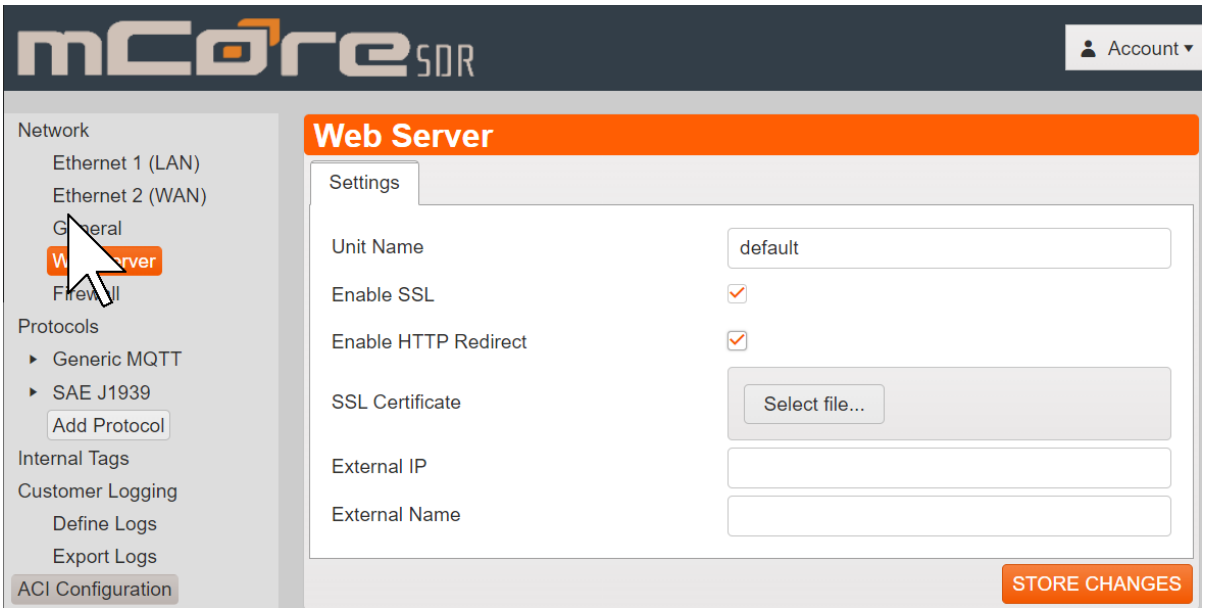
2. Next to the SSL Certificate label in the center menu, click the Select file button. A pop-up window will appear. Navigate to the SSL file and then click Open.

3. When done, click the Store Changes button in the center menu. To activate the changes, you will have to click the Commit All Changes button in the left menu towards the bottom. It will show a pop box. Click Ok to continue. mCore will then reboot.

SECTION 7.4: EXTERNAL IP AND EXTERNAL NAME

The External IP and External Name fields can be used to specify additional data to be included in an automatically generated certificate's Subject Alternate Name field. They are not used if you provide a certificate yourself, but if you are allowing the mCore to generate its own certificate, you can use these fields to ensure that the name by which you refer to the unit is considered valid by a web browser. If, for example, you are accessing the mCore via a cell router that exposes it to the Internet using an IP address, you should enter that address as the External IP. If you use a name to refer to that IP address, either via your own DNS infrastructure or perhaps a hosts file, you can include that name as the External Name. The automatically generated certificate will already include the unit's name, the unit's name followed by ".local", and any static IP's allocated to Ethernet 1 or Ethernet 2. If the unit name is left at default, the serial number of the unit will be used.

1. To change the External IP and/or External Name, click the Web Server button in the left menu toward the top. The following screen will appear.
   Enter or edit an External IP and/or External Name in their respective boxes.



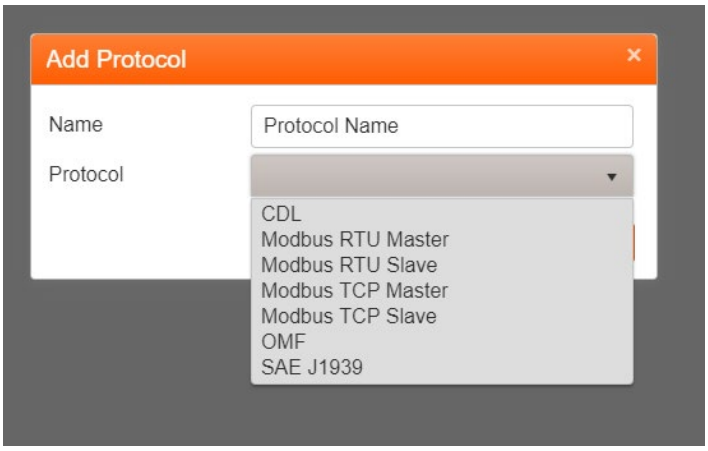
2. When done, click the Store Changes button in the center menu. To activate the changes, you will have to click the Commit All Changes button in the left menu towards the bottom. It will show a pop box. Click Ok to continue. mCore will then reboot.

SECTION 8: PROTOCOLS

SECTION 8.1: ADDING A PROTOCOL

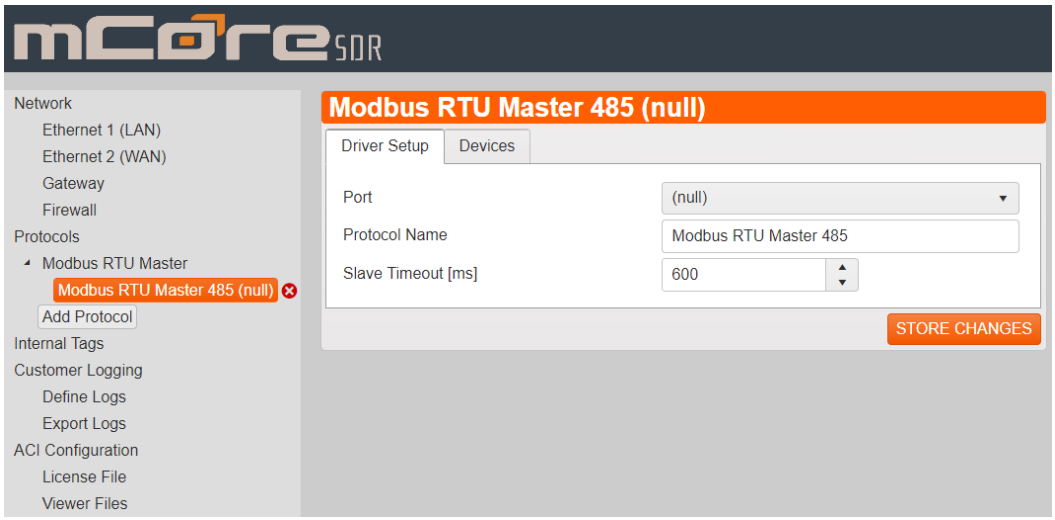
1. To add a protocol, click the Add Protocol button under the section Protocols in the left menu and the following pop up box will appear. Name the protocol and select the protocol from the drop-down menu. Once named and the protocol selected, click Add Protocol.

2. Once you've clicked Add Protocol, the protocol will show up under the Protocols section in the left menu.



Click to Return to Top of Document

3. Click the name or the arrow beside the protocol name to expand list and select the next level that appears. This will take you to the protocol setup section. Refer to the relevant protocol sections that follow this section for further setup instructions.



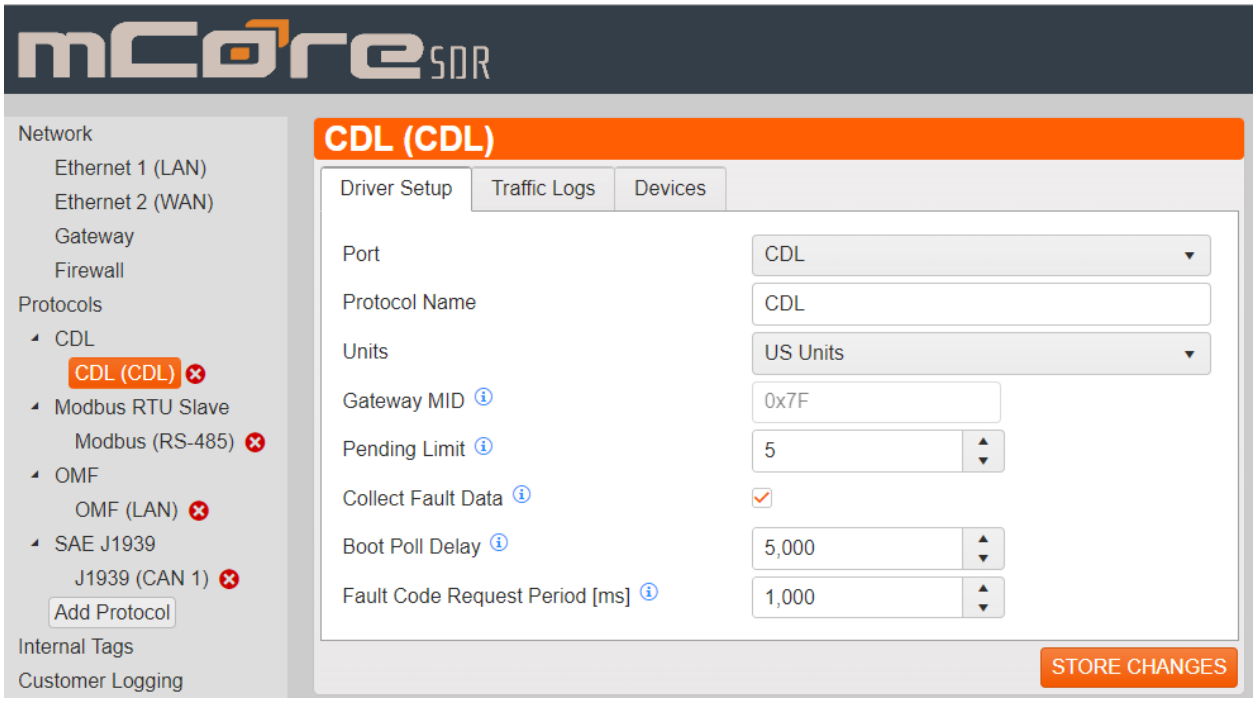
SECTION 8.2: CDL

CDL Theory of Operations:

- Cat Data Link is a proprietary protocol developed by Caterpillar®
- Most mCore units will come with CDL already mapped out, but the following will demonstrate how to setup this protocol.

1. After adding the Protocol per the Add Protocol section, click the second level of the protocol.

2. mCore automatically scales and converts the CDL data to the units of your choice. Choose between US Units (psi, °F, etc.) or Metric (kPa, °C, etc.).



3. The Pending Limit is essentially a communications speed regulator. Some older CDL ECMs need to be talked to slower to communicate well. The range is 1 – 8. Default is set to a 5. If communications seem to drop out or become erratic, try lowering the pending limit.

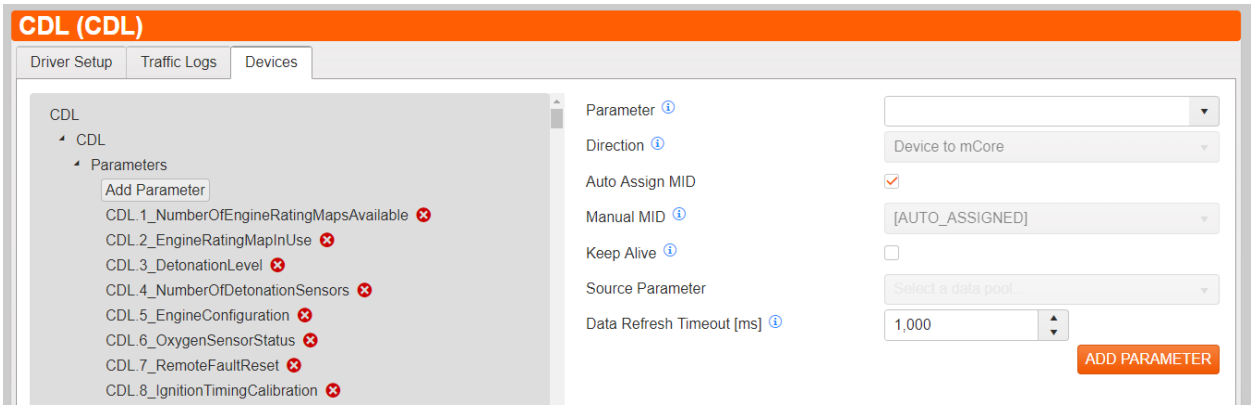
4. Collect Fault Data turns on the ability to receive fault information. This is set to On as default.

5. Boot Poll Delay is the amount of time mCore will wait after being booted up before it starts communicating over the CDL bus.

6. Fault Code Request Period is how frequent mCore checks for Faults.

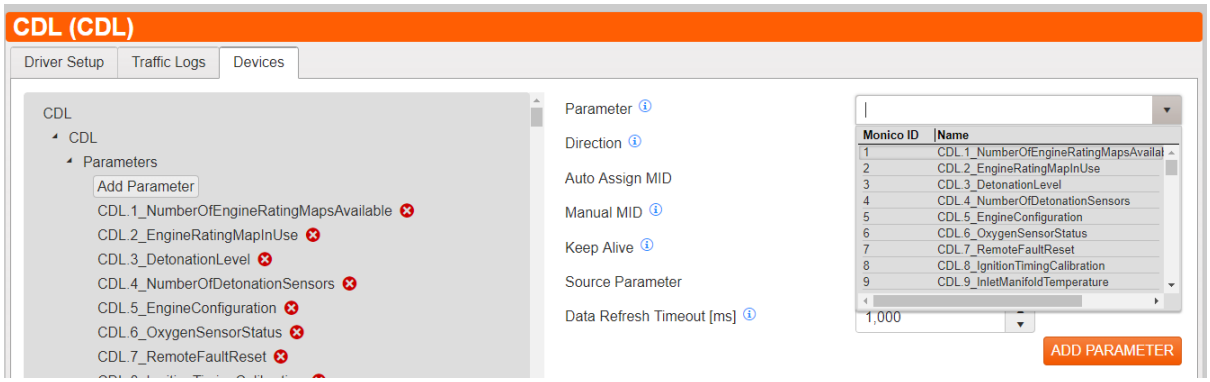
7. Click the Store Changes button when done.

8. Click the Devices tab. The following screen will appear. To add a parameter to an existing or blank CDL configuration, click the Add Parameter button located in the center menu. A blank parameter screen will appear on the right side.



9. Click the top drown down box labeled Parameter. This will contain a search box as well as a scrollable list containing all available CDL parameters in the driver.

10. Choose the desired parameter and click the Add Parameter button in the bottom right.



11. Repeat Steps 1 through 3 until you've added all desired CDL parameters you wish to monitor. Once done, you must click the Commit All Changes button in the left menu towards the bottom for the changes to take effect.

SECTION 8.2.1: CDL TRAFFIC LOG

This is a diagnostics tool for Monico Support staff should the need arise. If asked to get a CDL Traffic Log, follow the steps below for enabling the Traffic Log. With CDL it is important to follow a specific procedure to gather adequate traffic logs. For Instructions on exporting the Traffic Logs, see Section 13.6.

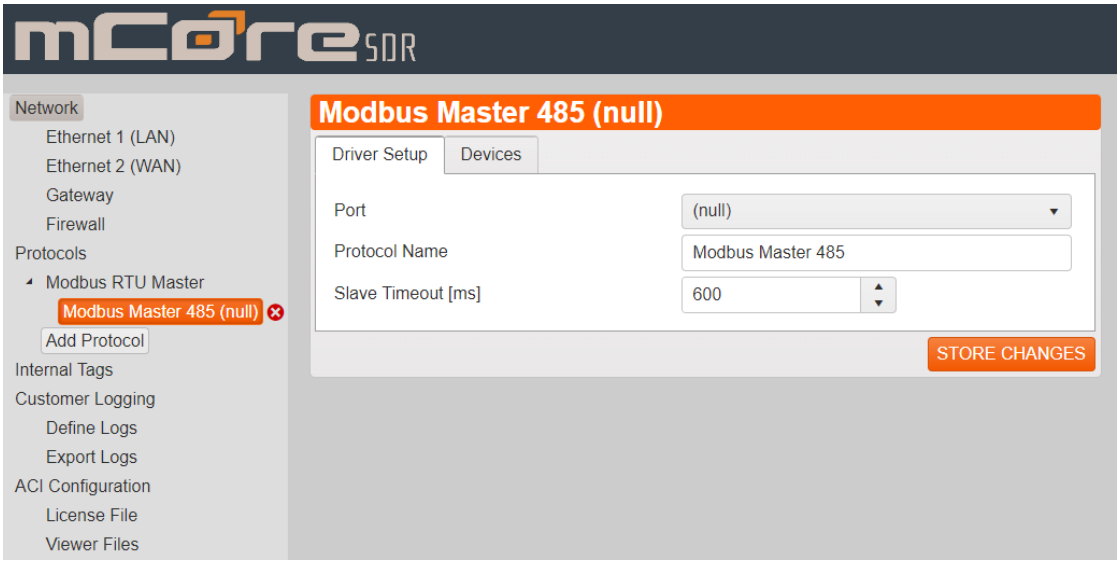
1. Ensure that **Keyswitch** on the machine is **OFF** to make sure the ECM's are powered down. mCore should be powered at this point for the following steps while the ECM's are powered down. If you are using the Service Tool Connector, the power from this connector is not switched, so it remains powered regardless of keyswitch.
2. Click the check box next to Collect Traffic Log, then click the Store Changes button.
3. Go to Export System Logs menu and click Clear System Logs. Store Changes.
4. Commit All Changes and wait for mCore to restart.
5. Turn the Keyswitch ON to power the ECM's and leave running for 5-15 minutes.
6. Go to the Export System Logs menu and download the System Logs and send to Support@monicoinc.com.

SECTION 8.3: MODBUS SETUP

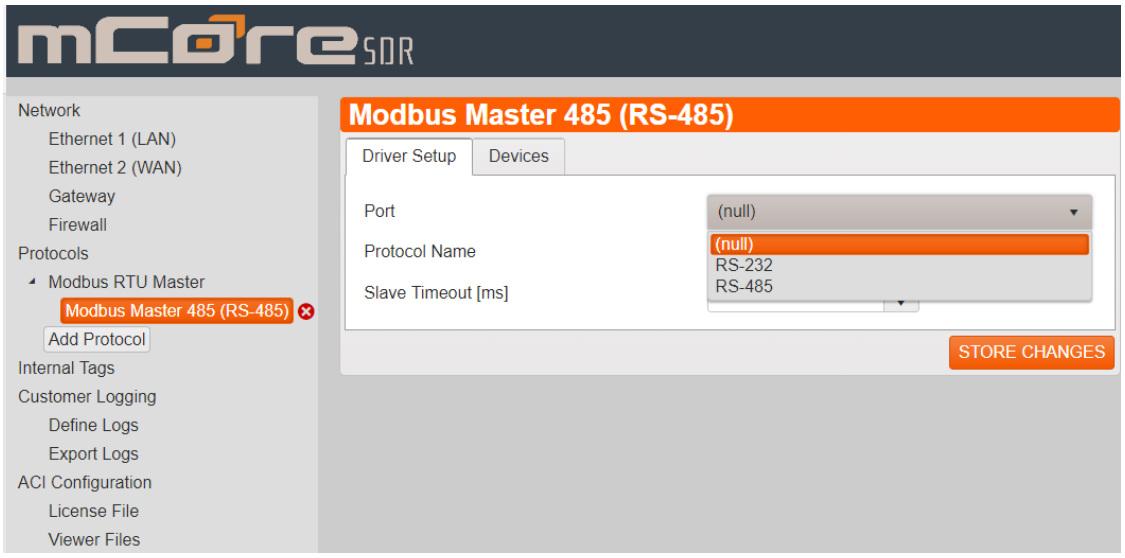
Modbus Theory of Operations:

Modbus RTU is a Single Master, but multiple slave protocol that is always a poll/response where the master sends a poll request and nothing else happens until the slave responds or a timeout setting is reached. Modbus TCP simply updated the protocol to work over TCP/IP Ethernet and added the ability to have multiple masters polling slave devices.

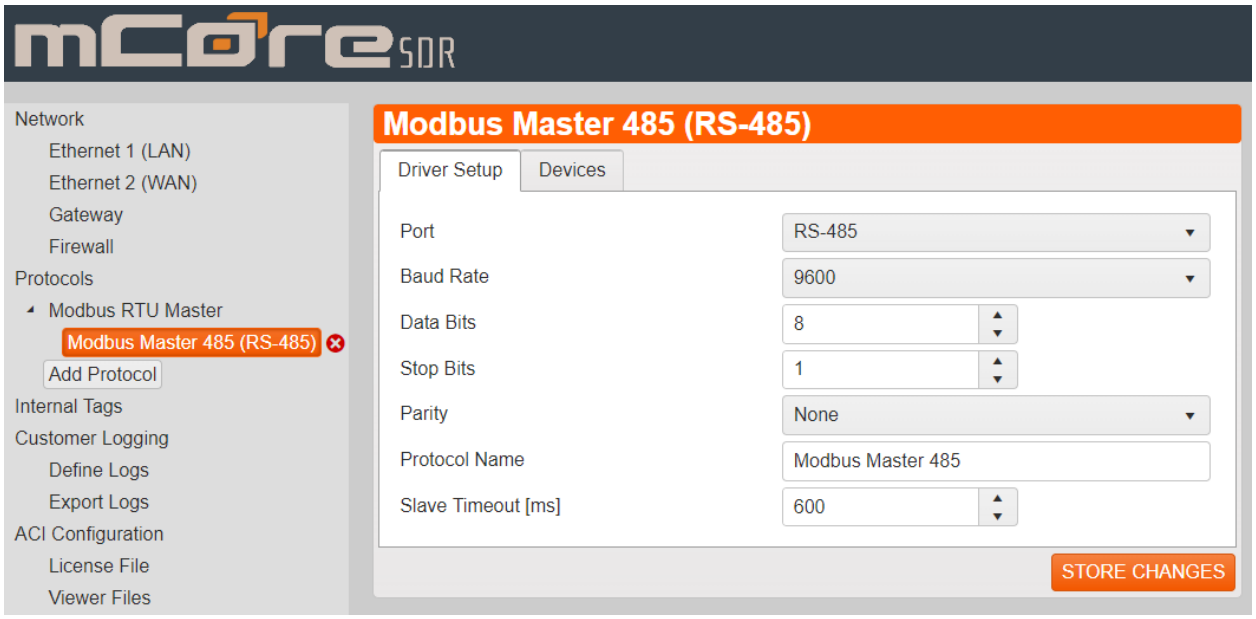
1. After adding the Protocol per the Add Protocol section, click the second level of the protocol.



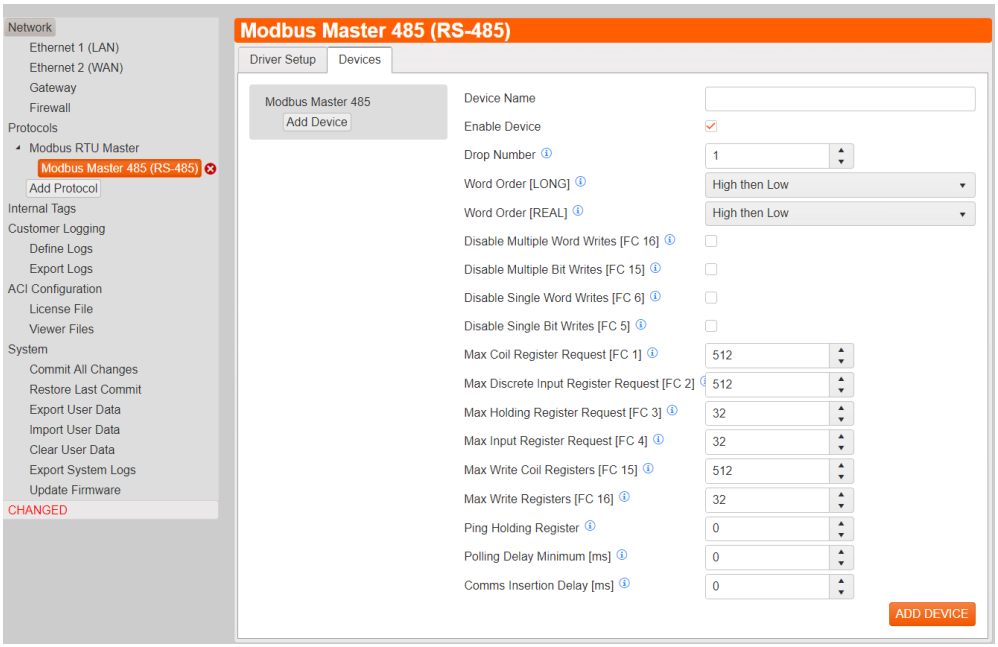
2. Choose the desired port, either 485 or 232 depending on the application.



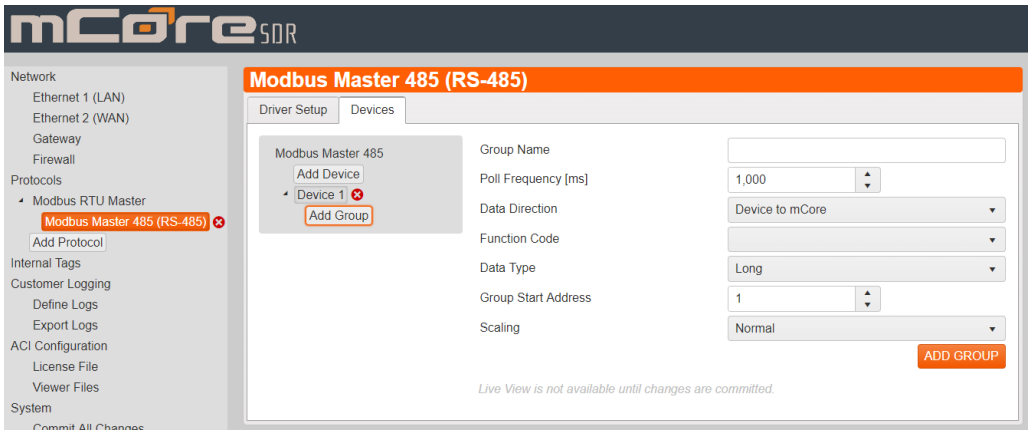
3. Once the port is chosen, other settings will appear. Make sure these settings match the settings of the device(s) that mCore will be communicating with. Click Store Changes button at the bottom right when you are done.



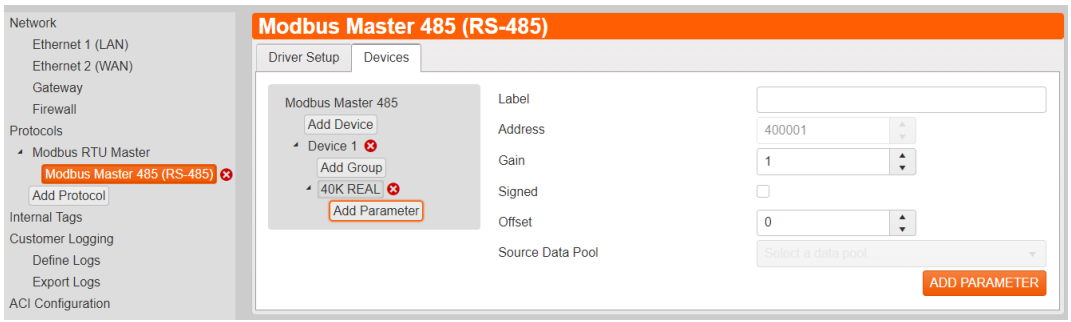
4. Click the Devices tab, then click the Add Devices button. A similar screen to below, depending on the Modbus protocol, will appear. When you are done, click the Add Device button at the bottom right. If you have multiple devices, repeat this step.



5. Once the device is added, a Add Group button will appear under the Add Device button. Click Add Group. The following will appear. Here you will choose the data type (word, real/float, or long), the direction (read or write), the function code (currently only function codes 3 and 4 are supported), and the starting address of the Modbus Register. Scaling Normal versus Scaling Reverse reverses the math operation. Contact Customer Support for more information if needed. Click Add Group in the bottom right when done.



6. The button Add Parameter will appear under the Add Group button in the top left as shown. Here you can start mapping out the individual registers starting at the number specified at the group settings in the above step. If this was a Slave application, the Source Data Pool drop down menu would be where you choose the parameter from a different protocol to map out. The UI (User Interface) will count for you as you add registers. The Gain and Offset allow you to apply scaling if needed. If you need to skip a large number of registers in your mapping, start a new group. If a small skip is needed, create an internal tag with a meaningless value to use as a filler tag to skip unneeded registers. When you are done, click the Add Parameter button in the bottom right.



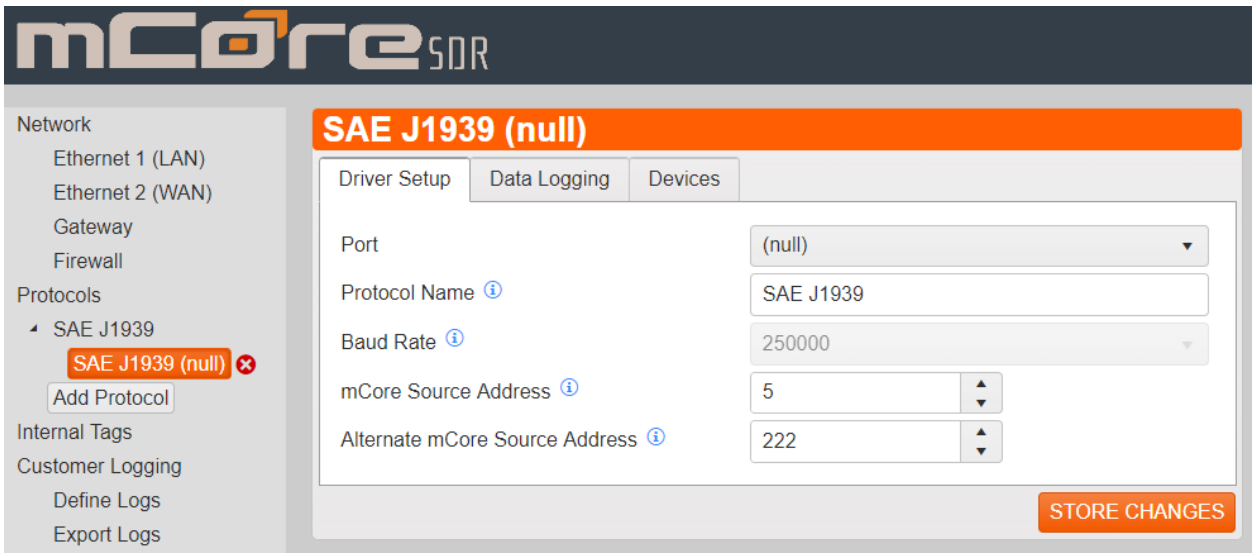
7. Repeat steps 5 and 6 until you have added all registers. If talking to multiple devices, repeat steps 4 through 6. Once done, you must click the Commit All Changes button in the left menu towards the bottom for the changes to take effect.

SECTION 8.4: SAE J1939 SETUP

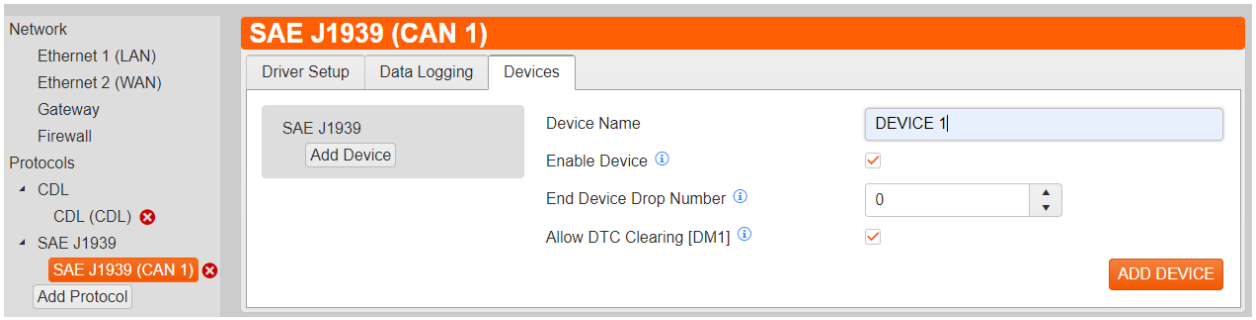
S.A.E. J1939 Theory of Operations:

J1939 is one of many protocols that run on a CANbus network. J1939 is primarily a broadcast protocol where there are a very large number of Parameter Group Numbers (PGN) and each PGN is assigned a number of Suspect Parameter Numbers (SPN) which are publicly defined and maintained by S.A.E.org. One unique characteristic of CANbus networks is they have a low-level transport protocol that allows for acknowledgments to confirm packet receipt that is very similar to Ethernet TCP/IP transport layer.

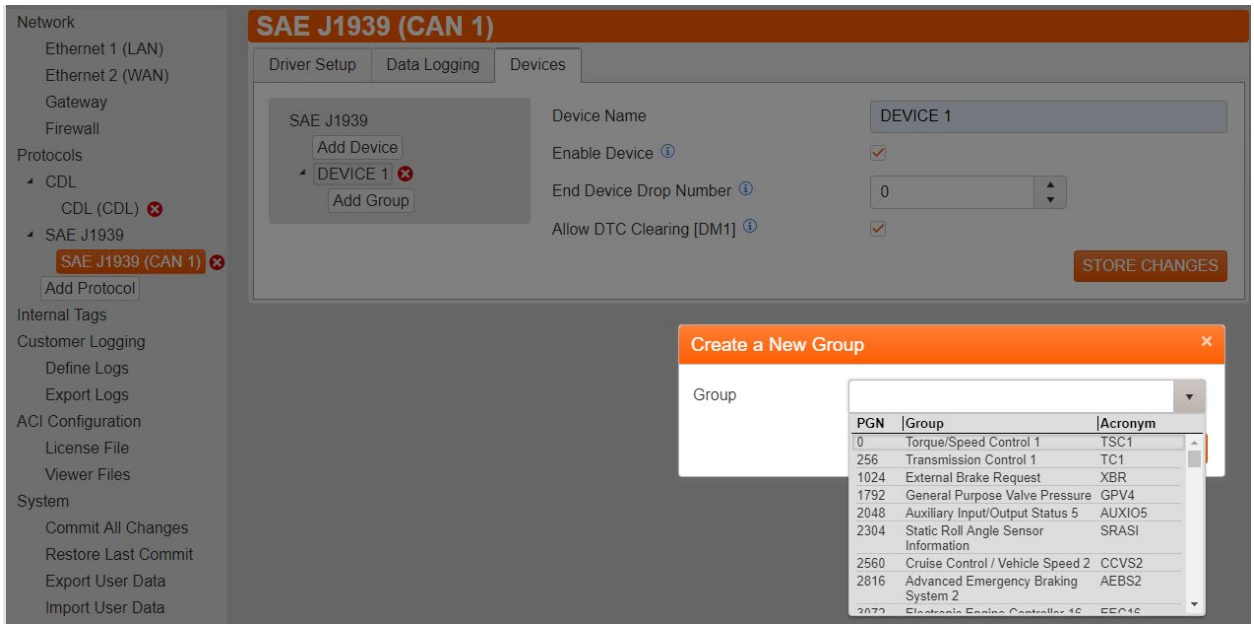
1. After adding the Protocol per Add Protocol section, click the second level of the protocol. Here you will select which CAN port, 1 or 2, you are physically wired into. Also, you can also edit the mCore's Source Address and Alternate Source address. Click Store Changes in the bottom right when done.



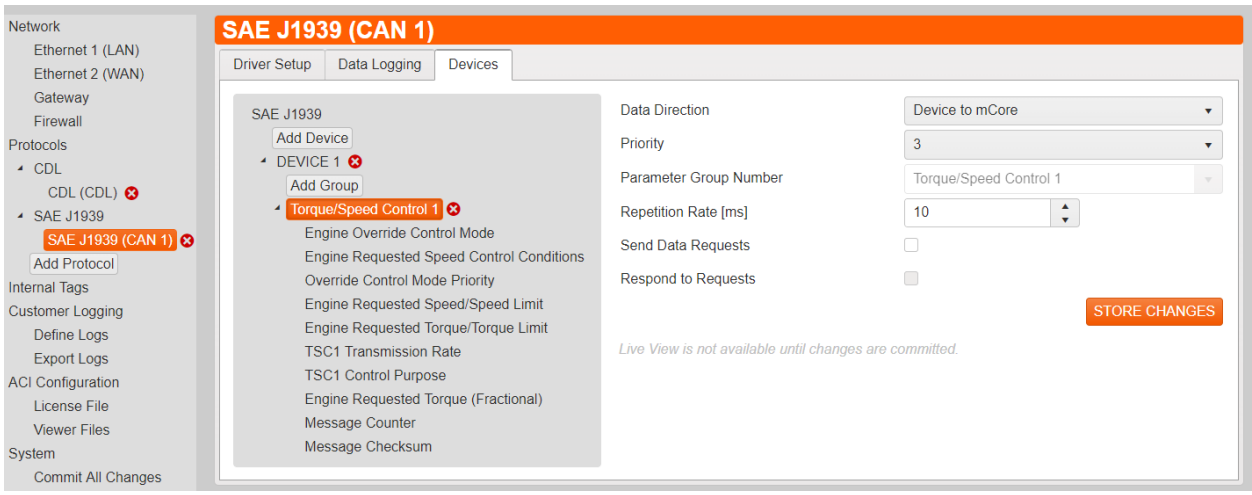
2. Click the Devices tab, then click the Add Device button. Here you can name the device, change the device drop number, and allow DTC clearing. When done, click the Add Device button in the bottom right.



3. Next click the Add Group button that appeared under device name. A pop-up window will appear. Click the drop-down menu to view the PGN list or use the search bar to find the PGN you are looking for. You can search by description, PNG number, and PGN Acronym. Once you are done, click Add Group.

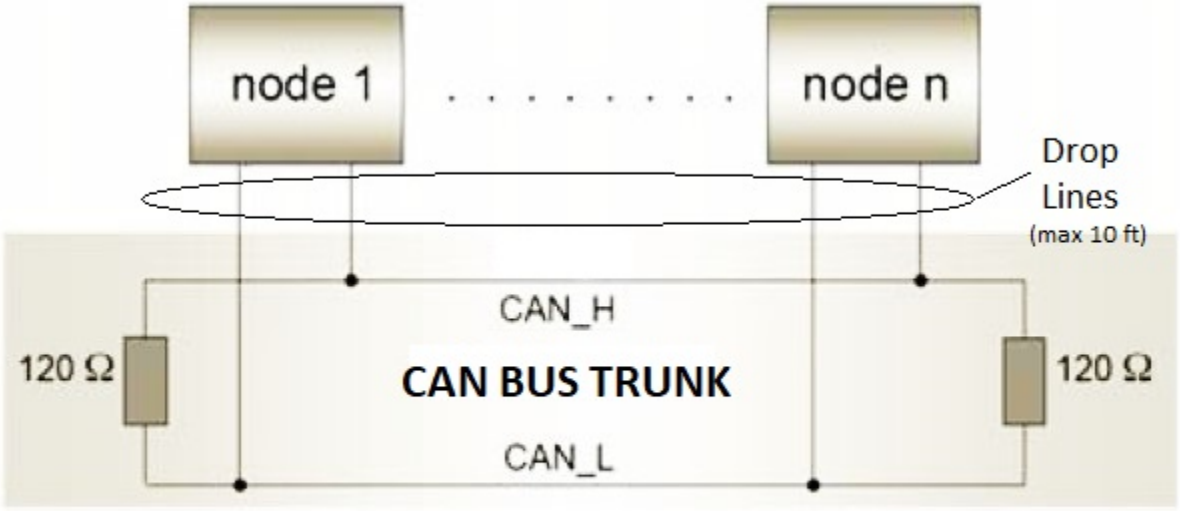


4. Next, you'll see the PGN in the list. Here you can change the data direction for outbound or inbound, change priority, repetition rate, and select if send data requests or respond to requests is needed. Click the Store Changes button when done.



SECTION 8.4.1: CONNECTIONS

CAN Bus is a multi-master serial bus used for connecting various ECMs and ECUs, also known as nodes, together for communication. The nodes are connected via unterminated drop lines to the main trunk. The trunk is terminated at both both ends with a single termination resistor, 120 ohms in J1939 applications, as shown in the figure below. Termination resistors reduce signal reflections on the bus which can cause significant ringing and increased error rates. It is imperative that correct cabling be utilized within the network on all drop lines and within the trunk. Use only the cable drop type shipped with the mCore or Northwire FJ1939182-002 in all applications.



You must locate and connect to the Primary CAN bus trunk of your equipment. Manufacturers generally label these as CAN_H (+) or CAN_L (-). There are times when it becomes difficult in identifying exactly which terminals to connect to within the equipment. Please check with your engine manufacturer as to which pins to connect into. There are a total of 2 CAN ports on the mCore and your mCore should already come prewired with the. Should it be necessary to install an additional line, please contact Monico and reference sections 3 and 4 for adding an additional CAN port line.

Extending the CAN Bus Trunk

As stated, your mCore should come pre-populated with a maximum 10 foot drop length for each CAN port you ordered. This is fine if you are within 10 feet of the Primary CAN Bus Trunk (Figure A) however, if you are more than 10 feet away, you will have to extend the CAN Bus Trunk and properly terminate the new extended trunk. Make sure that you are within max distance compliance and use the proper cable for any added lengths (Figure B). When all connections are made properly, your mCore's CANx LED will be Blinking Green.

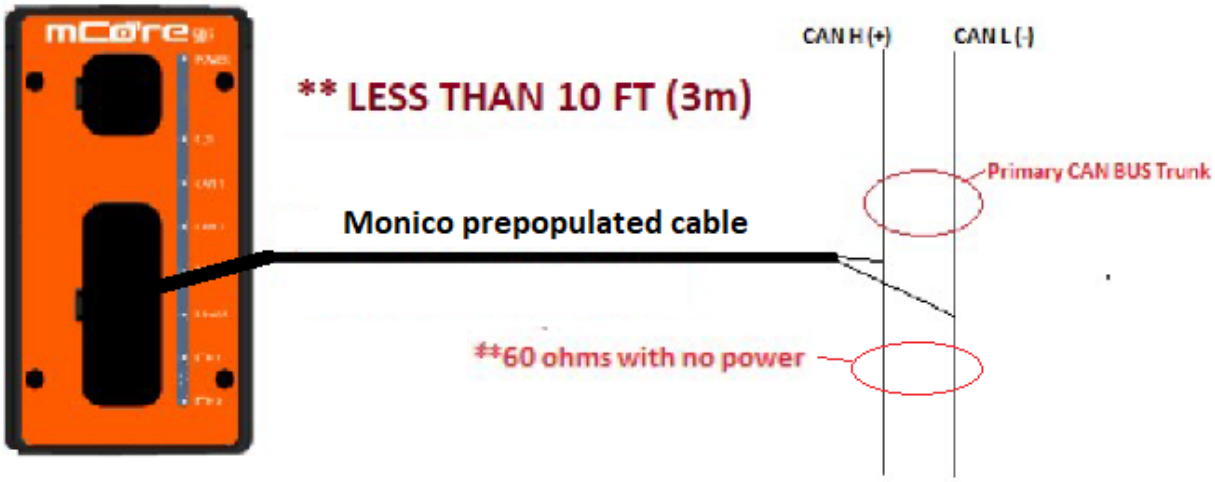


Figure A

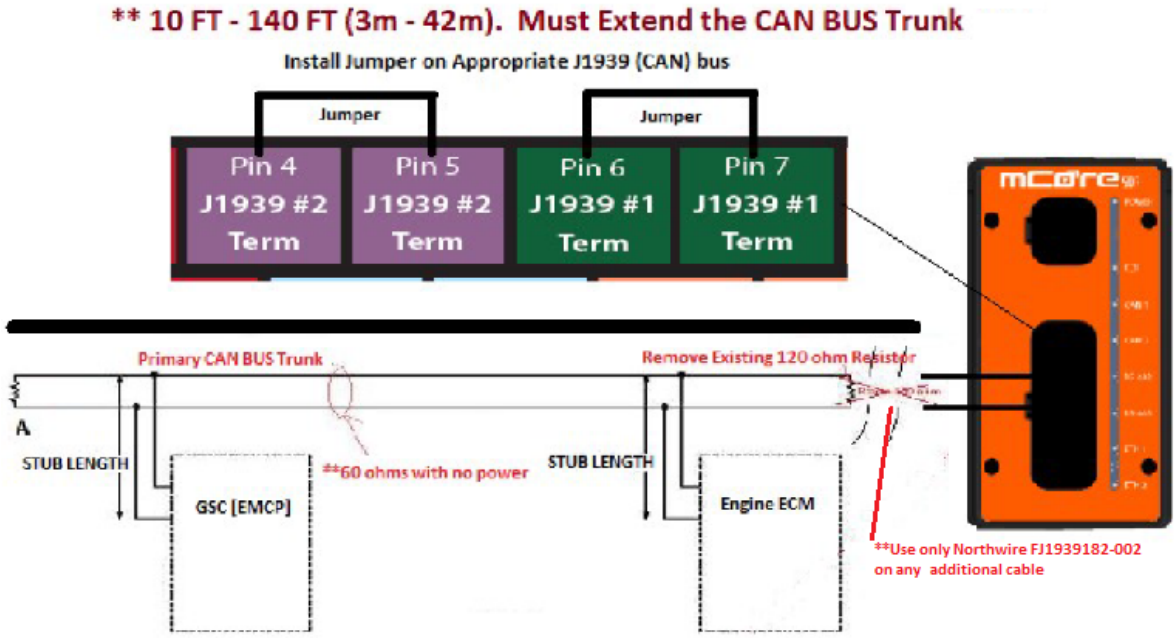
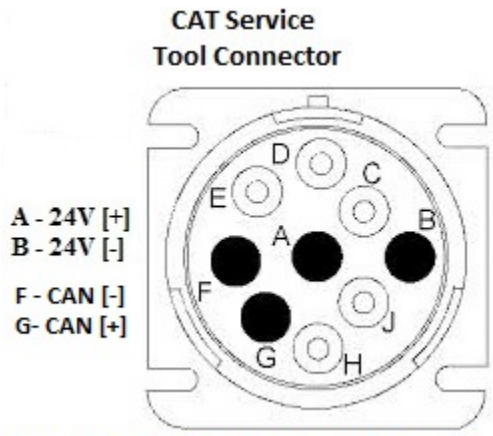


Figure B

Using the CAT Service Tool Connector as a reference

Please note below showing a typical 9 pin Deutsch connector on CAT. You can reference the back of the Service Tool Connector for the associated Wire Label and Colors. You can Trace these back to the terminal block for the proper connection point.



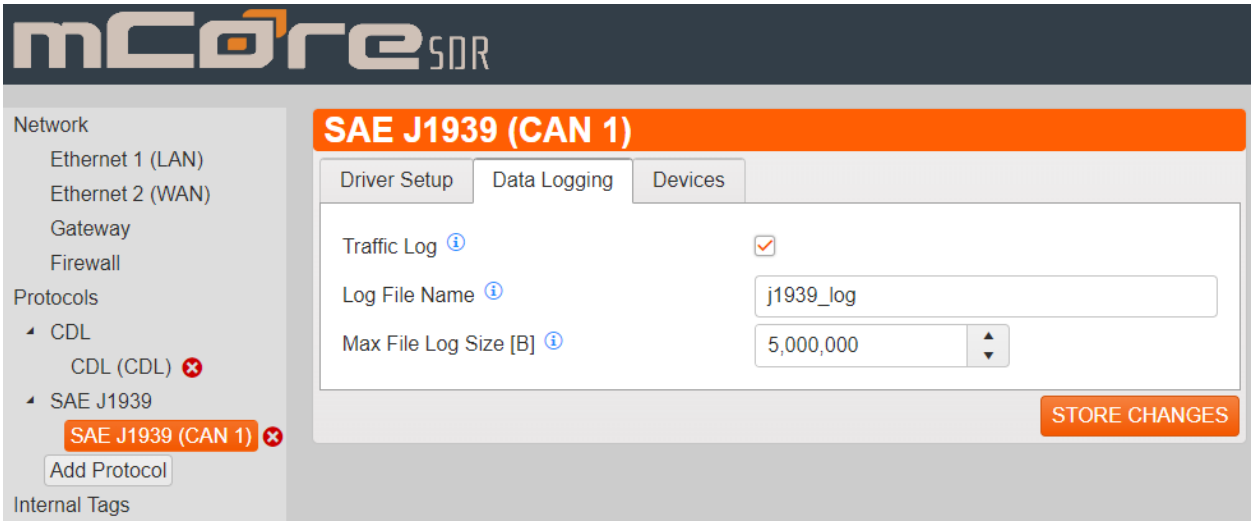
Click to Return to Top of Document

SECTION 8.4.2: J1939 TRAFFIC LOG

This is a diagnostics tool for Monico Support staff should the need arise. If asked to get a J1939 Traffic Log, follow the steps below for how to enable the Traffic Log. To export the Traffic Log, see Section 13.6.

1. Click the check box next to Traffic Log. Enter a name in the Log File Name box. We recommend leaving the Max File Log Size at the default value but could need it increased for special applications.

2. Click the Store Changes button when done. You must then click the Commit All Changes button in the left menu towards the bottom for the change to take effect.



SECTION 8.5: OMF

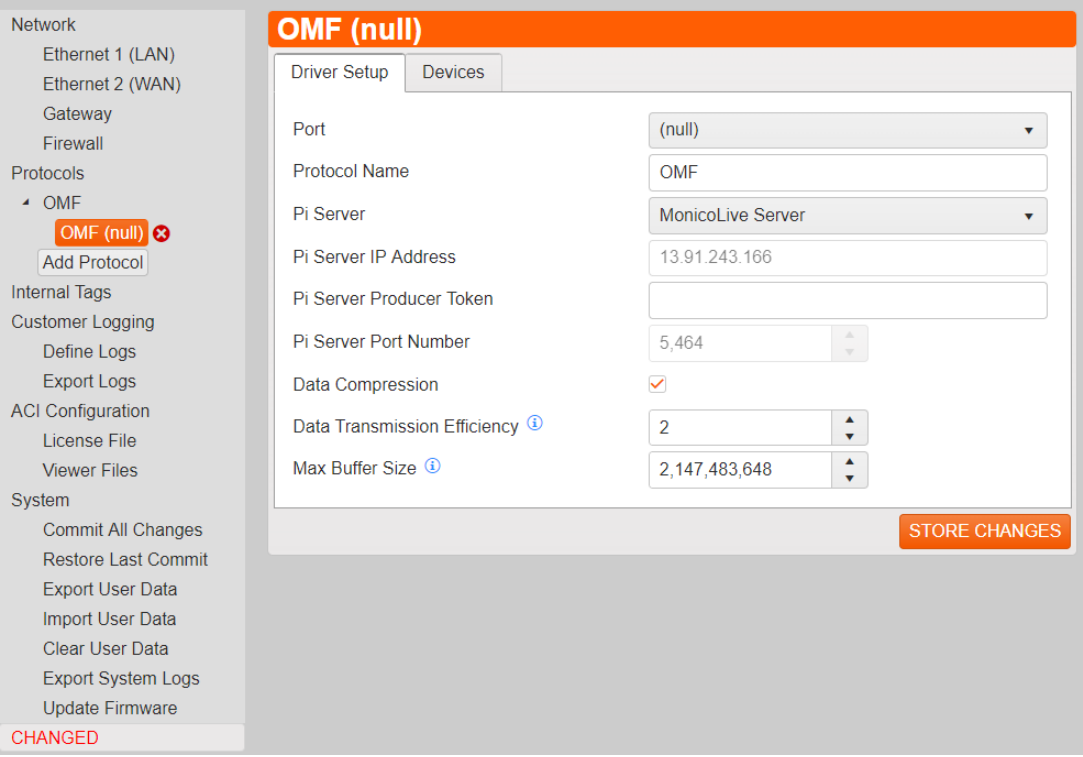
OMF Theory of Operations:

OMF (OSIsoft® Message Format) is a data transfer protocol developed to allow individual devices to send data to the PI Server directly without the need for an intermediary Interface or Connector device. In this protocol, Asset Frameworks (AF) elements and Pi Archive tags are created by mCore in the first message alleviating the need for exhaustive creation before the device connects. OMF is a “one-way” protocol that does not have any capability to send data back from the Pi Server to the end device. OMF uses ONLY an OUTBOUND SOCKET connection that is usually SSL/TLS encrypted. With a maximum message size of 192kB, it is a very efficient protocol for transmitting large volumes of data over an Ethernet network.

mCore’s implementation of OMF adds features that increase the efficiency of the protocol and improve data fidelity. To increase data efficiency, mCore offers “Exception reporting” which means that each data block is adjustable as to how it transmits and how often each parameter is checked for exception. The second major feature is data buffering to prevent packet loss. When a parameter is judged to have met the reporting

requirements, it is immediately time-stamped and sent to a memory buffer. If a connection is currently available, the message goes out. If not, the message is buffered with the original timestamp and transmitted when a connection is reestablished.

1. After adding the Protocol per the Add Protocol section, click the OMF protocol add in the left menu, then click the 2nd level. The following screen will appear.



2. Set the Port based off which Ethernet port will have network access to the PI Server you intend to connect to.

3. If you will be using the MonicoLive Server, leave that default selection alone. If you are using a customer owned server, click the dropdown and select Customer Supplied PI Server. You can then fill in the IP address for that server.

4. If you are using a Customer Supplied PI Server, enter the PI Server Port Number. If you are using MonicoLive, it will be filled in and un-editable.

5. The PI Server Producer Token is a security code for the data to be able to get into the PI Server. If you are using MonicoLive we will either fill this in before shipping or email this to the customer for them to put in onsite. If this is a Customer Supplied PI Server application, contact your PI administrator to get a Producer Token.

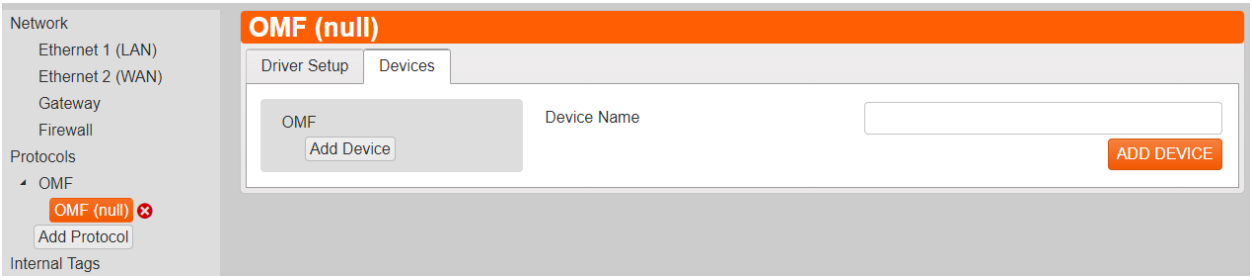
6. Data compression is set to default ON. We recommend using compression but can be turned off if desired. The amount of data sent will increase by magnitudes of 200% or more potentially.

7. Data Transmission Efficiency is a setting that will help reduce data usage. This is applicable when using cell modems and satellite connections. We recommend starting at 2 and working your way up until you are happy with the performance. Ranges from 2 - 10.

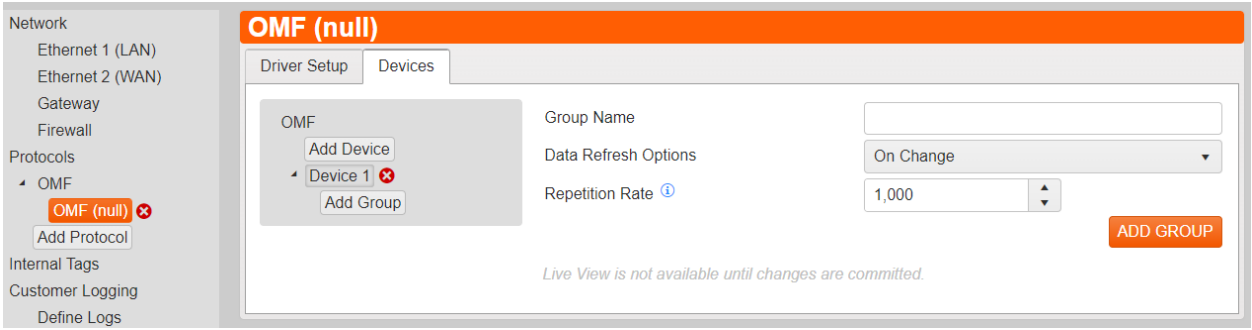
8. Max Buffer Size is the amount of space allocated on the unit to buffer data if the network cable is disconnected, or cell modem drops out. mCore will continue to retrieve data and store it until the connection is re-established. We recommend leaving this setting at the default unless your application will be doing a lot of logging. Once the connection is back, mCore will send the stored data. Depending on how long the connection was down and the speed of the connection, will determine how long it will take to backfill the data to the PI Server.

9. Click the Store Changes button in the center menu when done.

10. Click Devices Tab in the center menu. Then click the Add Device button. You will see the following screen. Name the device and click Add Device.



11. Next you will see a Add Group button appear under the Add Device button. Click Add Group. You will see the following screen.



12. Here you will start adding groups which will contain your tags. After naming the group you will choose between sending this group and all the tags in the group at a fixed Timed Interval (i.e. every 30 seconds) or On Change.

13. The Repetition Rate is dependent on the Data Refresh Option chosen. For Timed Interval this is the time between each data transmission sent for that group. For example, if 2 minutes go by with Repetition Rate set to 30,000ms or 30 s, you will have sent all tags in that group 4 times in a 2-minute span.

    For On Change, the Repetition Rate is how often mCore will look at the data and check to see if it has changed an equivalent or greater amount than the Threshold specified per parameter. The Threshold is set in future steps.

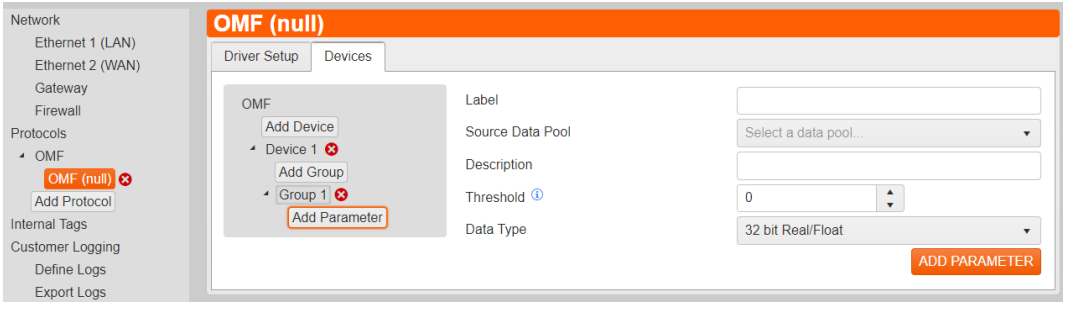
14. Click Store Changes when done.

15. Next click Add Parameter in the center menu. The following screen will appear. Choose from the Source Data Pool drop down menu the tag previously created from other protocols that you would like to send to the PI Server. The Label will autofill with the same name as the Source Data Pool but can be changed if desired. A description is not required but can be added. The threshold is used as plus or minus the data value.

    For example: if the value is 10 and the threshold is 5, that means only values that are either ≥ 15 or ≤ 5 will trigger the entire group to send all data values. If there is only 1 tag per group, then only that tag's value gets sent. If there are 10 tags in the group, then 1 tag exceeding its threshold will trigger all 10 tags' values to be sent at the same time.

16. Click Add Parameter when done.

17. Repeat Steps 11 through 15 until you've added all tags and groups desired. Click Commit All Changes in the left menu toward the bottom when done and ready to start feeding data to the PI Server.



Note: The most efficient mapping method when using On Change is 1 parameter per 1 group. This keeps each tag independent and will only send 1 data point when triggered versus all tags in the entire group.

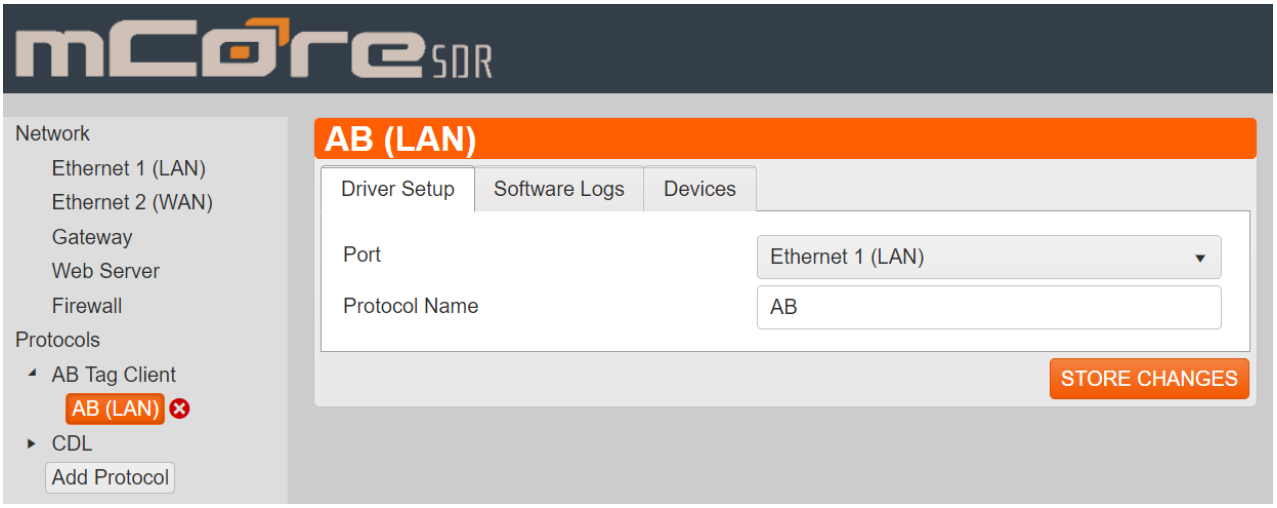
SECTION 8.6: AB TAG CLIENT

AB Tag Client Theory of Operations:

The Allen Bradley Tag Client performs as a master to write and read data to and from a Compactlogix or ControlLogix PLC. The required setup in the PLC program is to create the needed Controller Tag Arrays noted in the mCoreSDR data map. mCoreSDR supports Controller Tags and Controller Tag arrays with Integer, Double Integer, and Real Floating-point data types.

Data is written to the PLC immediately on connection, but subsequent writes are only made when the source data changes. Therefore, if you overwrite a value in the PLC, mCore will NOT overwrite that value until the source data changes which triggers a new write.

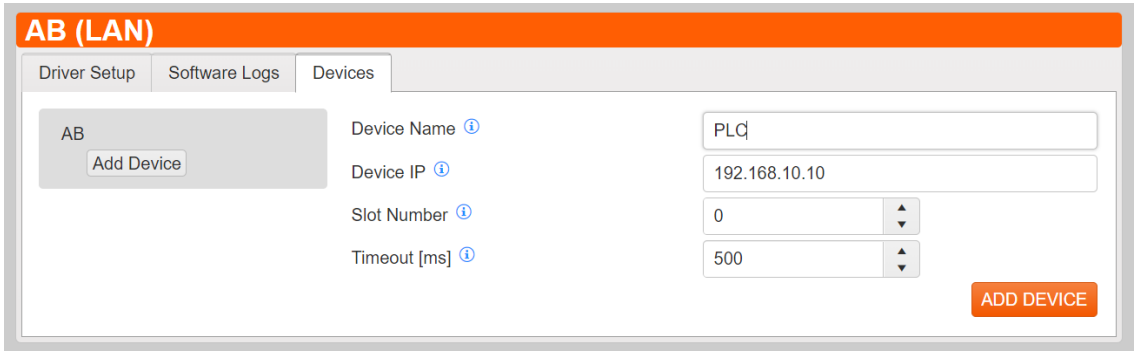
1. After adding the Protocol per the Add Protocol section, click the AB Tag Client protocol in the left menu, then click the 2nd level. The following screen will appear.



2. Select the ethernet port that the mCore will be communicating with the PLC on.

   Note: The Software Logs tab should only be used if Monico Technical Support asks you to turn it on to help troubleshoot a problem. It does not provide customer accessible information. It is strictly a troubleshooting tool.

3. Click the Devices tab, click the Add Device button. You will see the following screen.



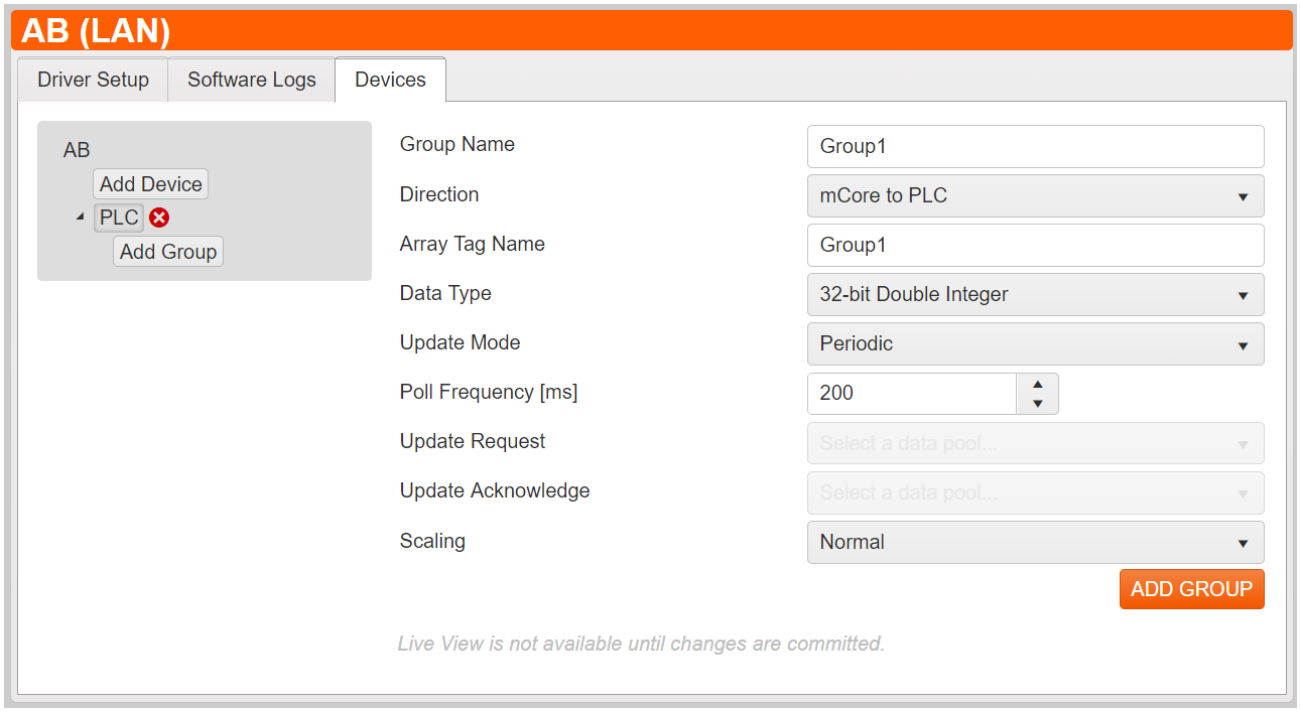
4. Give the device a name, then enter the PLC's IP address.

5. The Slot Number does not matter for CompactLogix PLCs so leave it as 0. If you are using a ControlLogix PLC, enter the slot number of the processor module.

6. The Timeout is defaulted to 500 ms. Adjust if necessary.

7. When done, click the Add Device button in the bottom right.

8. Next, click the Add Group button in the center menu towards the left under the device added previously. The following screen will appear.



9. Name the group of parameters you will be adding in the Group Name. It is a good practice to make the Group Name the same as the Array Tag Name to prevent confusion.

10. The Array Tag Name will be the name of the Array that you will have to enter in the PLC. It must be exactly as you enter it in this box, including capitalization, spelling, spaces, special characters, etc. It is a good practice to make the Array Tag Name the same as the Group Name to prevent confusion.

11. Choose the relevant data type. Make sure your PLC data type matches as well.

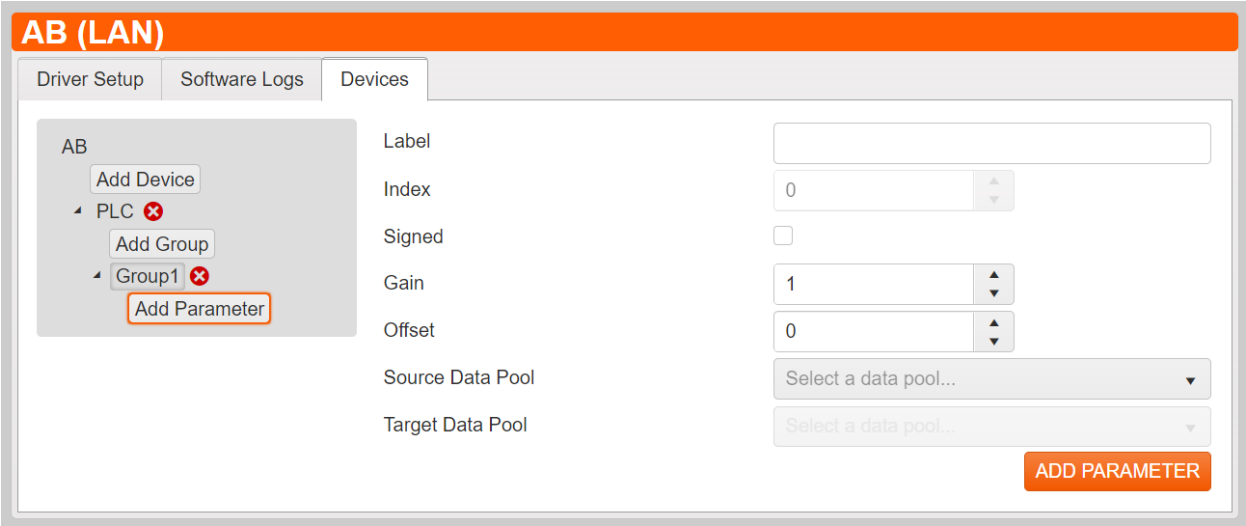
12. The Periodic Update Mode is a send-on-change mode with a Poll Frequency that limits how many changes will be sent in each time interval. mCore is checking for change as fast as it can and will send a value when a change occurs but will not send changing values any faster than the Poll Frequency setting. So, if a value hasn't changed in a minute and the Poll Frequency is set to 5 seconds, then a change occurs, that value is immediately sent.
    However, 1 second later, the value changes again. This value will not be sent until the 5 second Poll Frequency timer has expired. If during that waiting period, the value changes again before the timer is met, the latest/newest changed value will be sent at the end of the 5 seconds.

    To simplify this, set it to the desired update rate you would like to see data refresh at. If the data changes every second and you put the Poll Frequency to one second, then you'll more than likely get a one second refresh rate.

13. The other Update Mode, Triggered, is reserved for use with ACI eRCM Express applications only and should not be used for applications other than that.

14. Click the Add Group button when done.

15. Next click the Add Parameter button in the center menu under the Group that was created in the previous step. You will see the following screen.



16. The index numbering is automatic, so starting with 0, you will start adding your parameters to each index number by using the drop-down menu labeled Source Data Pool. Navigate through this drop-down menu, then click the desired parameter.

17. You can also apply a gain/multiplier and offset to the value before it is sent to the PLC as well as choose between signed and unsigned when relevant.

18. Click the Add Parameter button when done.

19. Repeat Steps 14 through 18 until you've added all groups and tags desired. Click Commit All Changes in the left menu toward the bottom when done.

SECTION 8.7: MQTT PROTOCOLS

MQTT Theory of Operations:

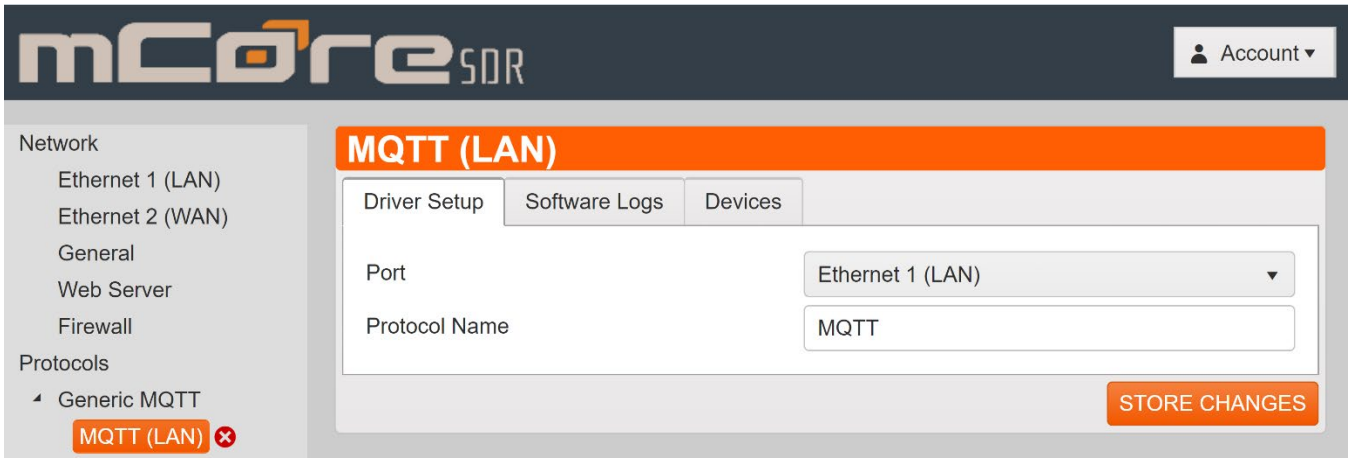
MQTT is a higher-level transport protocol for Publisher/Subscriber networks. However, since it is a transport protocol it does not define a payload. All MQTT protocols require there to be a publisher, a subscriber, and a broker. In most cases, the publisher publishes one or more topics to the broker. One or more subscribers register a subscription to one or more topics. Some things to keep in mind:

- The publisher opens a connection to the broker only. It has no concept of what the subscriber is doing.
- The broker does not care about payload type or arrangement, but just passes through whatever is wrapped in the transport layer.
- The broker can be setup to buffer, but it can get very complicated, so most brokers are in reliable proximity with the subscriber.
- The subscriber does not really have a concept of the publisher, but only subscribes to one or more topics.
- If a publisher publishes a topic to a broker, but the subscriber is not online, the message can be lost unless there is a specific setup to buffer that topic for that subscriber.

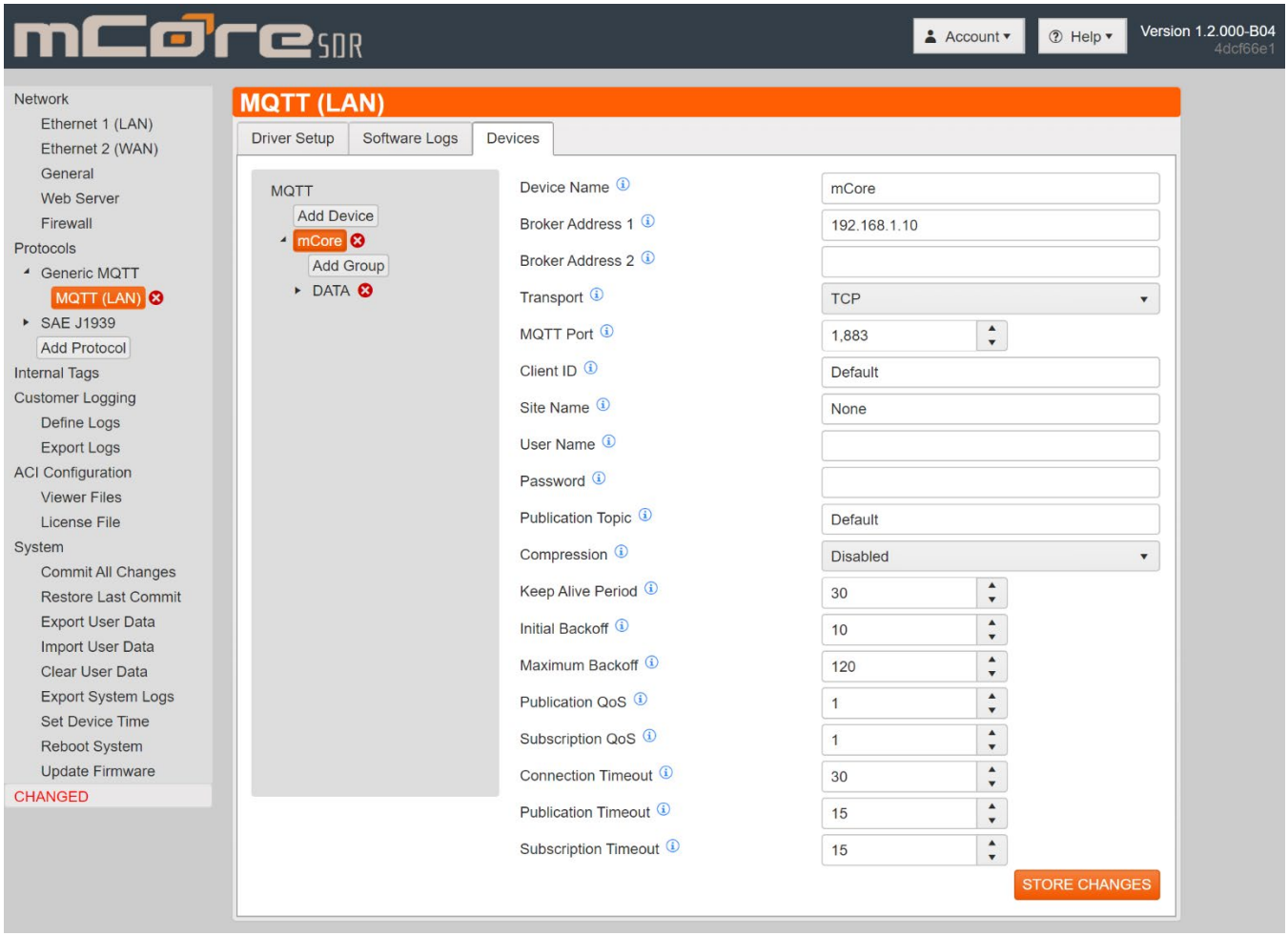
SECTION 8.7.1: MQTT-GENERIC

Generic MQTT is designed to be rather flexible implementation with a JSON payload that is relatively simple.

1. ***Driver Setup*** is very straight forward where you name the protocol anything you wish. This name is only for the User Interface and does not affect the topic name. Choose which Ethernet port you would like to use.



2. When you create a device, this will create a topic. If you want to publish more than one topic, you simply create more than one device.

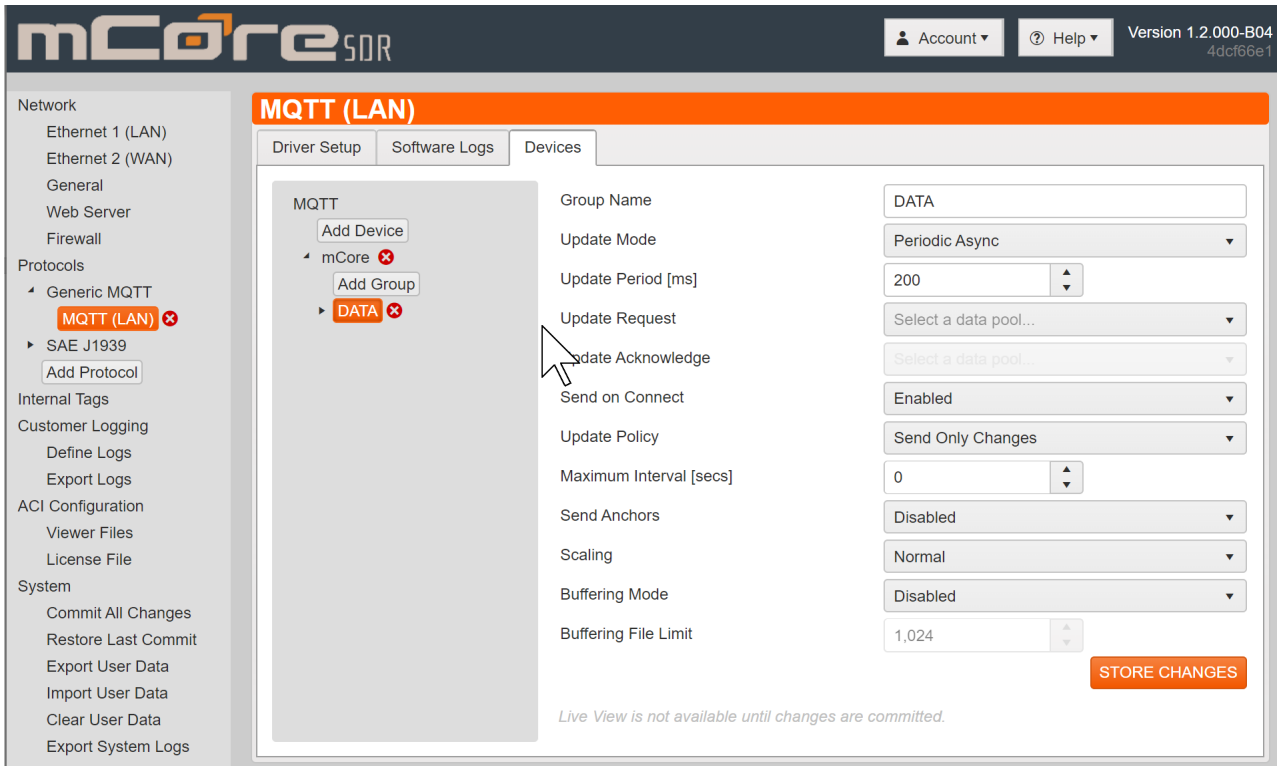


3. In the Device setup screen there are numerous settings:

   a. ***Device Name***: This name is for User Interface only and has no significance as to the topic name.
   b. ***Broker Address***: There are two slots for Broker Addresses. mCore will try to open a connection to Broker Address 1 and then will cycle to Broker Address 2 based on the settings described in more detail below.
   c. ***Transport:*** This protocol can be use on unencrypted TCP/IP or you can choose TLS-SSL.
   d. ***MQTT Port***: you can set any port you want if it is setup on both the broker and the publisher. Standard ports are 1883 for TCP/IP and 8883 for TLS-SSL.
   e. ***Client ID***: This is a value included in the header of each message. It is used as part of the default topic name. In addition, it can just be used by the broker to help identify a publisher. If left at *Default*, mCore's serial number will be used.

f. ***Site Name***: This value is a string that is inserted into the JSON packet and can be used as a "hint" as to the source, but is not part of the topic name..
g. ***Username and Password***: These are the credentials to authenticate the publisher to the broker.
h. ***Publication Topic***: This field is where you enter a string that will be the topic name for the data published by mCore. If left at *Default*, the topic will be *mCore/client-id/pub*, where client-id is the Client ID setting. Since this itself defaults to the serial number, leaving both at default will produce a topic of *mCore/mc-0000-0000/pub* where *mc-0000-0000* is the serial number of mCore.
i. ***Compression***: We do offer two types of compression, gzip and zlib. However, you do not have to use any compression and the subscriber MUST be able to decompress the packets. Compression is transparent to the broker.
j. ***Keep Alive Period***: If there are no packets sent within this time, mCore will send a small packet to keep the connection open and prevent disconnection from the broker.
k. ***Initial Backoff***: If mCore cannot establish a connection with the broker it will back off for this time period and then try again. If it still does not get a connection, it will double this value and will continue to try at longer and longer multiples of this time setting.
l. ***Maximum Backoff***: this setting is the maximum amount of time it will back off when a connection cannot be established.
m. ***Publication QoS***: MQTT has three Quality of Service settings. They are:
    i. 0=*Fire and Forget.*
    ii. 1=*Send At Least* Once by using an acknowledge to ensure that the packet was received. Note that with this setting you are guaranteed to get a packet once, but it could be received MORE than once. This level can slow data flow considerably depending on the network latency, especially when replaying buffered data.
    iii. 2=*Send Exactly Once* by using a more complex acknowledge mechanism. Note that this ensures you will get the packet ONLY ONCE, but it also slows down the data flow even more.
n. ***Subscription QoS***: Same as above when acting as a subscriber.
o. ***Connection Timeout***: this is the setting for how long to wait for a connection to be established before entering the Initial Back off period.
p. ***Publication Timeout***: This setting describes the amount of time to wait for an acknowledgement from a publication message.
q. ***Subscription Timeout***: This setting describes the amount of time to wait for an acknowledgement from a subscription message.

4. At the Group Level, you can setup how you want each data group to trigger to send.

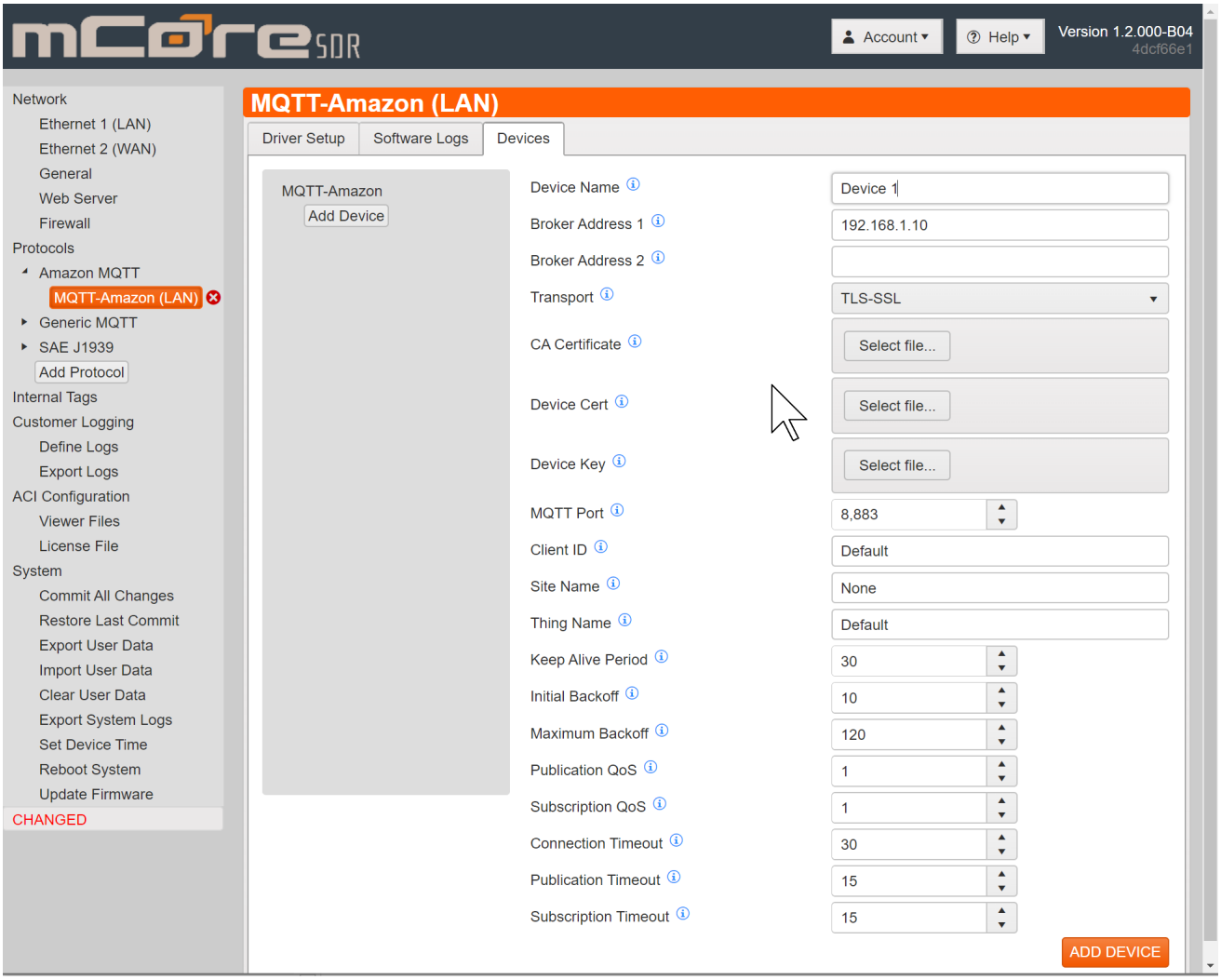


a. ***Group Name***: This name is only for the User Interface to reference.
b. ***Update Mode***: This option indicates how the entire group will be processes for exception reporting. Setting are:
    i. *Periodic Async*: This method sets an update period where during this period each value will be continuously checked for exception. If a parameter meets the exception test, it is sent immediately no matter where it happens inside the update period. In this mode, you may get a message anytime during the update period, but you will never get a message more often than once every period.
    ii. *Periodic Sync*: Very similar to Async, but the exception checks only take place at the end of each update period and that time period starts and stops on the REAL TIME CLOCK period. So, if you choose a 5000 ms update period, the first check for exception will be at 12:00:00. If any parameters meet exception, they will be transmitted at this time stamp. The next exception check will happen at 12:00:05 and so on. So, you may NOT get a message at the end of each time period, but all messages will be sent on the real time clock period end.
    iii. *Triggered*: This mode is used when you want a group to be processed only when a certain condition exists. In this mode, the Update Request is an edge-sensitive trigger for an update, and Update Acknowledgement is the response telling the system it triggered. This group will not be processed again until the trigger is reset to zero and then returns to non-zero once again.

c. ***Update Period***: This is the time period of action taken on the group as defined by the Update Mode above.
d. ***Update Request***: If the Update Mode is set to Periodic Sync or Async, then the Update Request is an "ENABLE" for this data group. I.e. If you only wanted to check a group while an engine is running, you may create an internal tag that says Is RPM>500. When you put this internal tag into the Update Request, the group will only be processed if the Internal tag is true and the update period is satisfied.
e. ***Update Acknowledgement***: This sets the acknowledge that the group has triggered and can be reset to rearm the triggered group for the next triggered event.
f. ***Send on Connect***: This setting controls whether to send data immediately upon connection. Enable or Disable.
g. ***Update Policy***: This sets the policy determining what values in a group are sent when an exception is met:
    i. *Send Only Changes*: This policy is most common and means that upon each evaluation of the group, only parameters that have met the exception will be time stamped and sent.
    ii. *Send All On Any Change*: This policy that if any parameter meets exception values, it will send all parameters in the group.
    iii. *Send All Each Time*: This policy will cause all parameters to be sent at each time period whether any exception is met or not. To be clear, all parameters will be sent every time period.
h. ***Maximum Interval***: This setting will cause the group to send each parameter in a group at least this often. Note that this feature is implemented on a per-parameter basis, such that it does not force the entire group to be sent every so-many seconds, but rather forces additional transmissions so that each parameter is sent at least that often. The timing and contents of this transmissions will depend on whether each parameter has already changed within the specified period.
i. ***Send Anchors***: This feature causes not only the value that met the exception for a parameter to be sent, but also the previous value it was compared to during the evaluation. The values will be sent with sequential time stamps. This is a feature that is similar to the OSIsoft Interface and should be used with caution.
j. ***Scaling***: This setting allows for normal or reverse scaling of each parameter.
k. ***Buffering Mode***: This setting enables or disables data buffering and allows a selection for which medium is used for buffering. Data buffering is where we store data and timestamps if a connection is not available.
    i. *To Memory*: Buffers data in RAM memory, which is faster, but is volatile, so if mCore loses power the buffered data will be lost.
    ii. *To Disk*: Buffers data to the eMMC non-volatile storage. Considerably slower, but easily fast enough for most applications.
l. ***Buffering File Limit***: Allows you to choose the amount of memory dedicated to buffering. Keep in mind that you if you use Data Logging, it will take some memory as well.

MQTT-Amazon Theory of Operations:

This version of MQTT is a JSON payload format that is very similar to the Generic version. The main differences on in the Authentication options and requirements of the AWS.

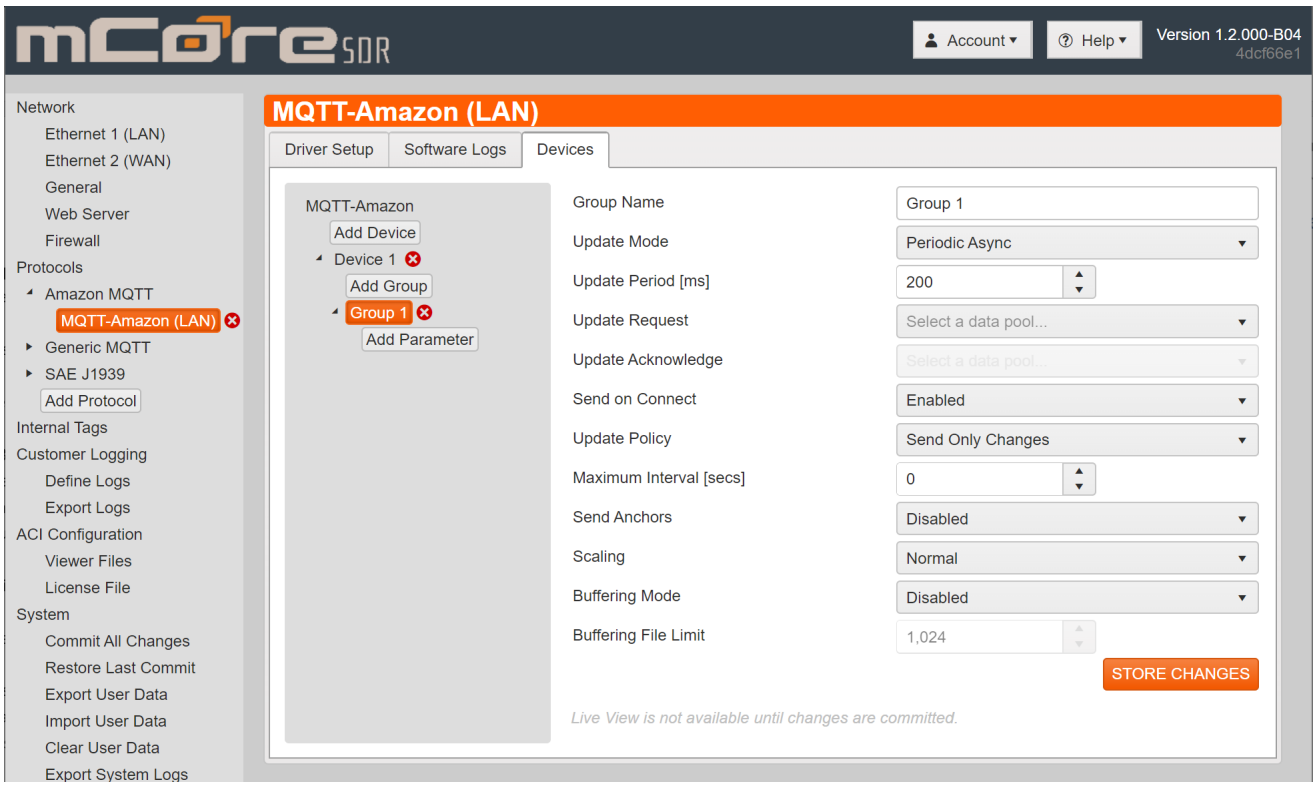


1. ***Driver Setup*** is very straight forward where you name the protocol anything you wish. This name is only for the User Interface and does not affect the topic name. Choose which Ethernet port you would like to use.
2. When you create a device, this will create a topic. If you want to publish more than one topic, you simply create more than one device.
3. In the Device setup screen there are numerous settings:

    a. ***Device Name***: This name is for User Interface only and has no significance as to the topic name.

b. ***Broker Address***: There are two slots for Broker Addresses. mCore will try to open a connection to Broker Address 1 and then will cycle to Broker Address 2 based on the settings described in more detail below.
c. ***Transport:*** This protocol can be use on unencrypted TCP/IP or you can choose TLS-SSL.
d. ***CA Certificate:*** To upload the root certificate(s) to be used to validate the broker's certificate.
e. ***Client Certificate:*** To upload a certificate to be used to validate the device to the broker.
f. ***Device Key:*** To upload the Private Key associated with the Client Certificate.
g. ***MQTT Port***: you can set any port you want if it is setup on both the broker and the publisher. Standard ports are 1883 for TCP/IP and 8883 for TLS-SSL.
h. ***Client ID***: This is a value included in the header of each message. It is used as part of the default topic name. In addition, it can just be used by the broker to help identify a publisher. If left at *Default*, mCore's serial number will be used.
i. ***Site Name***: This value is a string that is inserted into the JSON packet and can be used as a "hint" as to the source, but is not part of the topic name.
j. ***Thing Name***: The name of the thing within the AWS Server. This is REQUIRED by AWS and is included in the topic name. However, the rest of the topic name is determined by AWS.
k. ***Keep Alive Period***: If there are no packets sent within this time, mCore will send a small packet to keep the connection open and prevent disconnection from the broker.
l. ***Initial Backoff***: If mCore cannot establish a connection with the broker it will back off for this time period and then try again. If it still does not get a connection, it will double this value and will continue to try at longer and longer multiples of this time setting.
m. ***Maximum Backoff***: this setting is the maximum amount of time it will back off when a connection cannot be established.
n. ***Publication QoS***: MQTT has three Quality of Service settings. They are:
o. 0=Fire and Forget.
p. 1=*Send At Least* Once by using an acknowledge to ensure that the packet was received. Note that with this setting you are guaranteed to get a packet once, but it could be received MORE than once. This level can slow data flow considerably depending on the network latency, especially when replaying buffered data.
q. 2=*Send Exactly Once* by using a more complex acknowledge mechanism. Note that this ensures you will get the packet ONLY ONCE, but it also slows down the data flow even more.
r. ***Subscription QoS***: Same as above when acting as a subscriber.
s. ***Connection Timeout***: this is the setting for how long to wait for a connection to be established before entering the Initial Back off period.
t. ***Publication Timeout***: This setting describes the amount of time to wait for an acknowledgement from a publication message.
u. ***Subscription Timeout***: This setting describes the amount of time to wait for an acknowledgement from a subscription message.

4. At the Group Level, you can setup how you want each data group to trigger to send.



a. ***Group Name***: This name is only for the User Interface to reference.

b. ***Update Mode***: This option indicates how the entire group will be processes for exception reporting. Setting are:

   i. *Periodic Async*: This method sets an update period where during this period each value will be continuously checked for exception. If a parameter meets the exception test, it is sent immediately no matter where it happens inside the update period. In this mode, you may get a message anytime during the update period, but you will never get a message more often than once every period.

   ii. *Periodic Sync*: Very similar to Async, but the exception checks only take place at the end of each update period and that time period starts and stops on the REAL TIME CLOCK period. So, if you choose a 5000 ms update period, the first check for exception will be at 12:00:00. If any parameters meet exception, they will be transmitted at this time stamp. The next exception check will happen at 12:00:05 and so on. So, you may NOT get a message at the end of each time period, but all messages will be sent on the real time clock period end.

   iii. *Triggered*: This mode is used when you want a group to be processed only when a certain condition exists. In this mode, the Update Request is an edge-sensitive trigger for an update, and Update Acknowledgement is the response telling the system it triggered. This group will not be processed again until the trigger is reset to zero and then returns to non-zero once again.

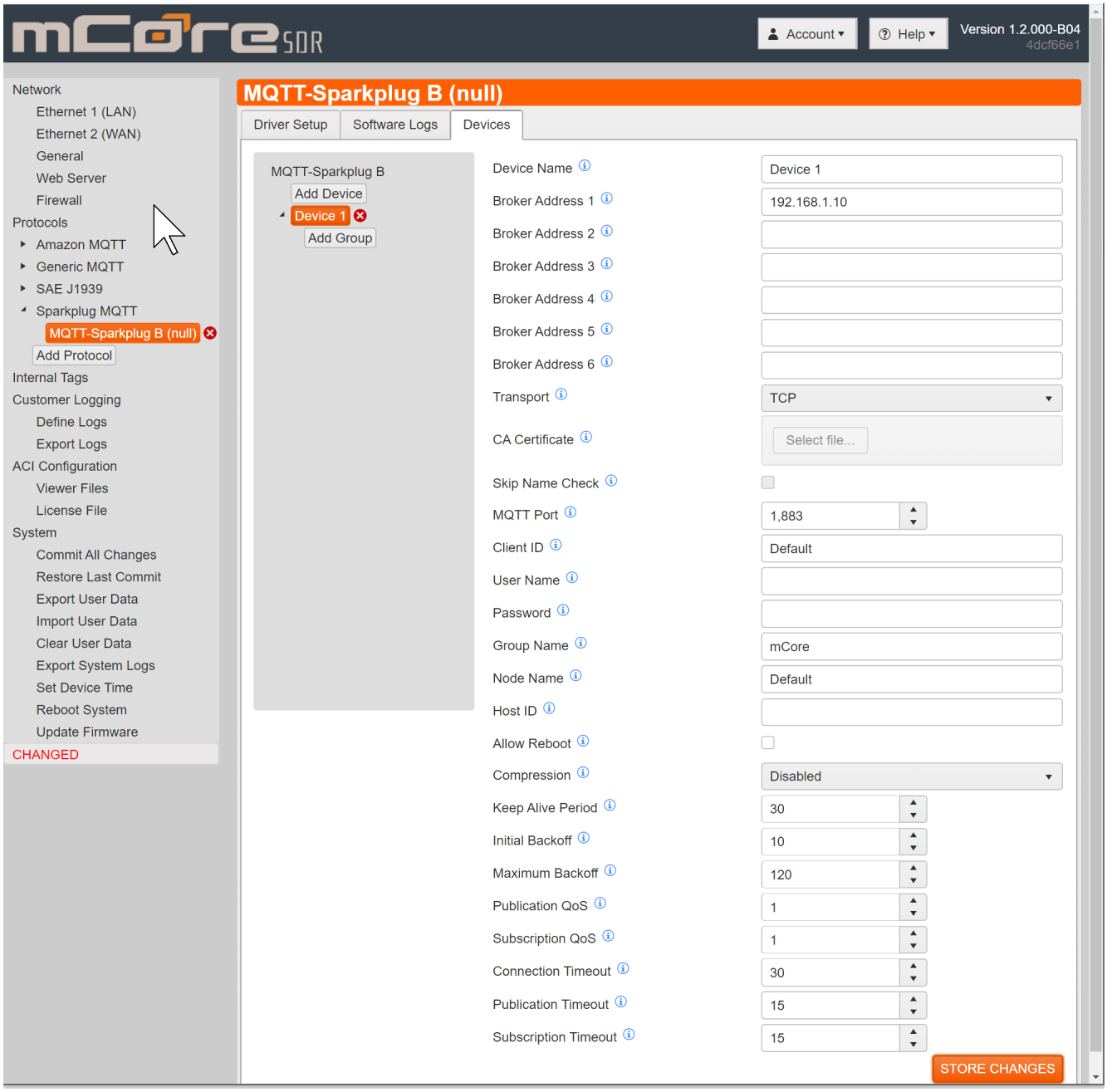
c. ***Update Period***: This is the time period of action taken on the group as defined by the Update Mode above.

d. ***Update Request***: If the Update Mode is set to Periodic Sync or Async, then the Update Request is an "ENABLE" for this data group. I.e. If you only wanted to check a group while an engine is running, you may create an internal tag that says Is RPM>500. When you put this internal tag into the Update Request, the group will only be processed if the Internal tag is true and the update period is satisfied.
e. ***Update Acknowledgement***: This sets the acknowledge that the group has triggered and can be reset to rearm the triggered group for the next triggered event.
f. ***Send on Connect***: This setting controls whether to send data immediately upon connection. Enable or Disable.
g. ***Update Policy***: This sets the policy determining what values in a group are sent when an exception is met:
    i. *Send Only Changes*: This policy is most common and means that upon each evaluation of the group, only parameters that have net the exception will be time stamped and sent.
    ii. *Send All On Any Change*: This policy that if any parameter meets exception values, it will send all parameters in the group.
    iii. *Send All Each Time*: This policy will cause all parameters to be sent at each time period whether any exception is met or not. To be clear, all parameters will be sent every time period.
h. ***Maximum Interval***: This setting will cause the group to send each parameter in a group at least this often. Note that this feature is implemented on a per-parameter basis, such that it does not force the entire group to be sent every so-many seconds, but rather forces additional transmissions so that each parameter is sent at least that often. The timing and contents of this transmissions will depend on whether each parameter has already changed within the specified period.
i. ***Send Anchors***: This feature causes not only the value that met the exception for a parameter to be sent, but also the previous value it was compared to during the evaluation. The values will be sent with sequential time stamps. This is a feature that is similar to the OSIsoft Interface and should be used with caution.
j. ***Scaling***: This setting allows for normal or reverse scaling of each parameter
k. ***Buffering Mode***: This setting enables or disables data buffering and allows a selection for which medium is used for buffering. Data buffering is where we store data and timestamps if a connection is not available.
    i. *To Memory*: Buffers data in RAM memory, which is faster, but is volatile, so if mCore loses power the buffered data will be lost.
    ii. *To Disk*: Buffers data to the eMMC non-volatile storage. Considerably slower, but easily fast enough for most applications.
l. ***Buffering File Limit***: Allows you to choose the amount of memory dedicated to buffering. Keep in mind that you if you use Data Logging, it will take some memory as well.

SECTION 8.7.3: MQTT-SPARKPLUG B

MQTT Sparkplug B Theory of Operations:

Sparkplug B is a special, open source Eclipse project, that is aimed at the industrial SCADA market. It still uses the MQTT Transport protocol but defines a very specific Namespace and is a binary payload. It offers many features such as Auto-Discovery of nodes and automatic health updates of each node. It is also intended to be used bi-directionally as required by a SCADA system.



Click to Return to Top of Document

1. ***Driver Setup*** is very straight forward where you name the protocol anything you wish. This name is only for the User Interface and does not affect the topic name. Choose which Ethernet port you would like to use.
2. When you create a device, this will create a topic. If you want to publish more than one topic, you simply create more than one device.
3. In the Device setup screen there are numerous settings:
   a. ***Device Name***: This name is for User Interface only and has no significance as to the topic name.
   b. ***Broker Address***: There are two slots for Broker Addresses. mCore will try to open a connection to Broker Address 1 and then will cycle to Broker Address 2 based on the settings described in more detail below. And will continue through all the broker addresses until if finds a connection.
   c. ***Transport:*** This protocol can be use on unencrypted TCP/IP or you can choose TLS-SSL.
   d. ***CA Certificate:*** To upload the root certificate(s) to be used to validate the broker's certificate.
   e. ***Skip Name Check:*** This check box disables the checking of the peer name in the broker certificate.
   f. ***MQTT Port***: you can set any port you want if it is setup on both the broker and the publisher. Standard ports are 1883 for TCP/IP and 8883 for TLS-SSL**.**
   g. ***Client ID***: This is a value included in the header of each message. It is used as part of the default topic name. In addition, it can just be used by the broker to help identify a publisher. If left at *Default*, mCore's serial number will be used.
   h. ***Username and Password***: These are the credentials to authenticate the publisher to the broker.
   i. ***Group Name***: The group name under which the device will publish data.
   j. ***Node Name***: The node name under which the device will publish data.
   k. ***Host ID***: The Host ID that the client will look for on the server.
   l. ***Allow Reboot Checkbox***: Allows the server to reboot this device.
   m. ***Compression***: We do offer two types of compression, gzip and zlib. However, you do not have to use any compression and the subscriber MUST be able to decompress the packets. Compression is transparent to the broker.
   n. ***Keep Alive Period***: If there are no packets sent within this time, mCore will send a small packet to keep the connection open and prevent disconnection from the broker.
   o. ***Initial Backoff***: If mCore cannot establish a connection with the broker it will back off for this time period and then try again. If it still does not get a connection, it will double this value and will continue to try at longer and longer multiples of this time setting.
   p. ***Maximum Backoff***: this setting is the maximum amount of time it will back off when a connection cannot be established.
   q. ***Publication QoS***: MQTT has three Quality of Service settings. They are:
      i. 0=Fire and Forget.

ii. 1=Fire and Wait for acknowledgement. Note: with this setting you are guaranteed to get a packet once, but it could be received MORE than once. This level can slow data flow considerably depending on the network latency.

iii. 2=Fire and Wait for acknowledgement only ONCE. Note: this ensures you will get the packet ONLY ONCE guaranteed, but it also slows down the data flow drastically depending on the system latency.

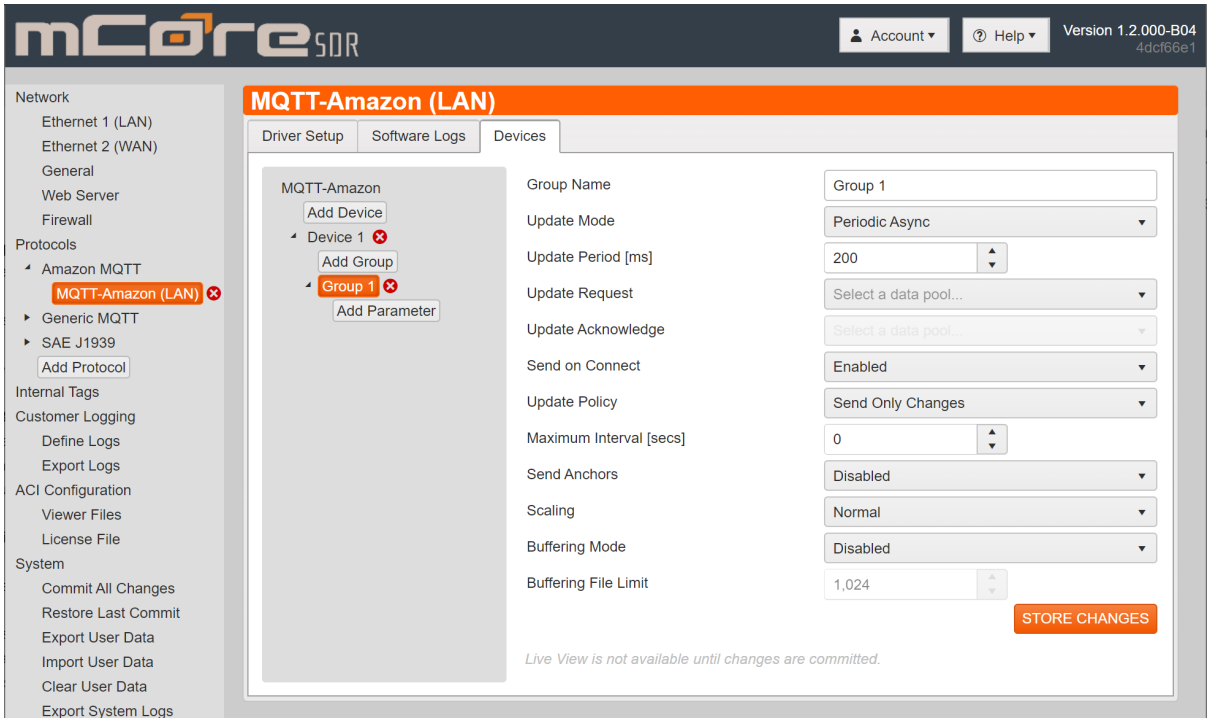
r. ***Subscription QoS***: Same as above when acting as a subscriber.

s. ***Connection Timeout***: this is the setting for how long to wait for a connection to be established before entering the Initial Back off period.

t. ***Publication Timeout***: This setting describes the amount of time to wait for an acknowledgement from a publication message.

u. ***Subscription Timeout***: This setting describes the amount of time to wait for an acknowledgement from a subscription message.

4. At the Group Level, you can setup how you want each data group to trigger to send.



a. ***Group Name***: This name is only for the User Interface to reference.

b. ***Update Mode***: This option indicates how the entire group will be processes for exception reporting. Setting are:

i. *Periodic Async*: This method sets an update period where during this period each value will be continuously checked for exception. If a parameter meets the exception test, it is sent immediately no matter where it happens inside the update period. In this mode, you may get a message anytime during the update period, but you will never get a message more often than once every period.

ii. *Periodic Sync*: Very similar to Async, but the exception checks only take place at the end of each update period and that time period starts and stops on the REAL TIME CLOCK period. So, if you choose a 5000 ms update period, the first check for exception will be at 12:00:00. If any parameters meet exception, they will be transmitted at this time stamp. The next exception check will happen at 12:00:05 and so on. So, you may NOT get a message at the end of each time period, but all messages will be sent on the real time clock period end.

iii. *Triggered*: This mode is used when you want a group to be processed only when a certain condition exists. In this mode, the Update Request is an edge-sensitive trigger for an update, and Update Acknowledgement is the response telling the system it triggered. This group will not be processed again until the trigger is reset to zero and then returns to non-zero once again.

c. ***Update Period***: This is the time period of action taken on the group as defined by the Update Mode above.

d. ***Update Request***: If the Update Mode is set to Periodic Sync or Async, then the Update Request is an "ENABLE" for this data group. I.e. If you only wanted to check a group while an engine is running, you may create an internal tag that says Is RPM>500. When you put this internal tag into the Update Request, the group will only be processed if the Internal tag is true and the update period is satisfied.

e. ***Update Acknowledgement***: This sets the acknowledge that the group has triggered and can be reset to rearm the triggered group for the next triggered event.

f. ***Send on Connect***: This setting controls whether to send data immediately upon connection. Enable or Disable.

g. ***Update Policy***: This sets the policy determining what values in a group are sent when an exception is met:

i. *Send Only Changes*: This policy is most common and means that upon each evaluation of the group, only parameters that have net the exception will be time stamped and sent.

ii. *Send All On Any Change*: This policy that if any parameter meets exception values, it will send all parameters in the group.

iii. *Send All Each Time*: This policy will cause all parameters to be sent at each time period whether any exception is met or not. To be clear, all parameters will be sent every time period.

h. ***Maximum Interval***: This setting will cause the group to send each parameter in a group at least this often. Note that this feature is implemented on a per-parameter basis, such that it does not force the entire group to be sent every so-many seconds, but rather forces additional transmissions so that each parameter is sent at least that often. The timing and contents of this transmissions will depend on whether each parameter has already changed within the specified period.

i. ***Send Anchors***: This feature causes not only the value that met the exception for a parameter to be sent, but also the previous value it was compared to during the evaluation. The values will be sent with sequential time stamps. This is a feature that is similar to the OSIsoft Interface and should be used with caution.

j. ***Scaling***: This setting allows for normal or reverse scaling of each parameter.

k. ***Buffering Mode***: This setting enables or disables data buffering and allows a selection for which medium is used for buffering. Data buffering is where we store data and timestamps if a connection is not available.

   I. *To Memory*: Buffers data in RAM memory, which is faster, but is volatile, so if mCore loses power the buffered data will be lost.

   II. *To Disk*: Buffers data to the eMMC non-volatile storage. Considerably slower, but easily fast enough for most applications. However, if buffering is constant and used a great deal, then it will eventually damage the eMMC storage.

   i. ***Buffering File Limit***: Allows you to choose the amount of memory dedicated to buffering. Keep in mind that you if you use Data Logging, it will take some memory as well.

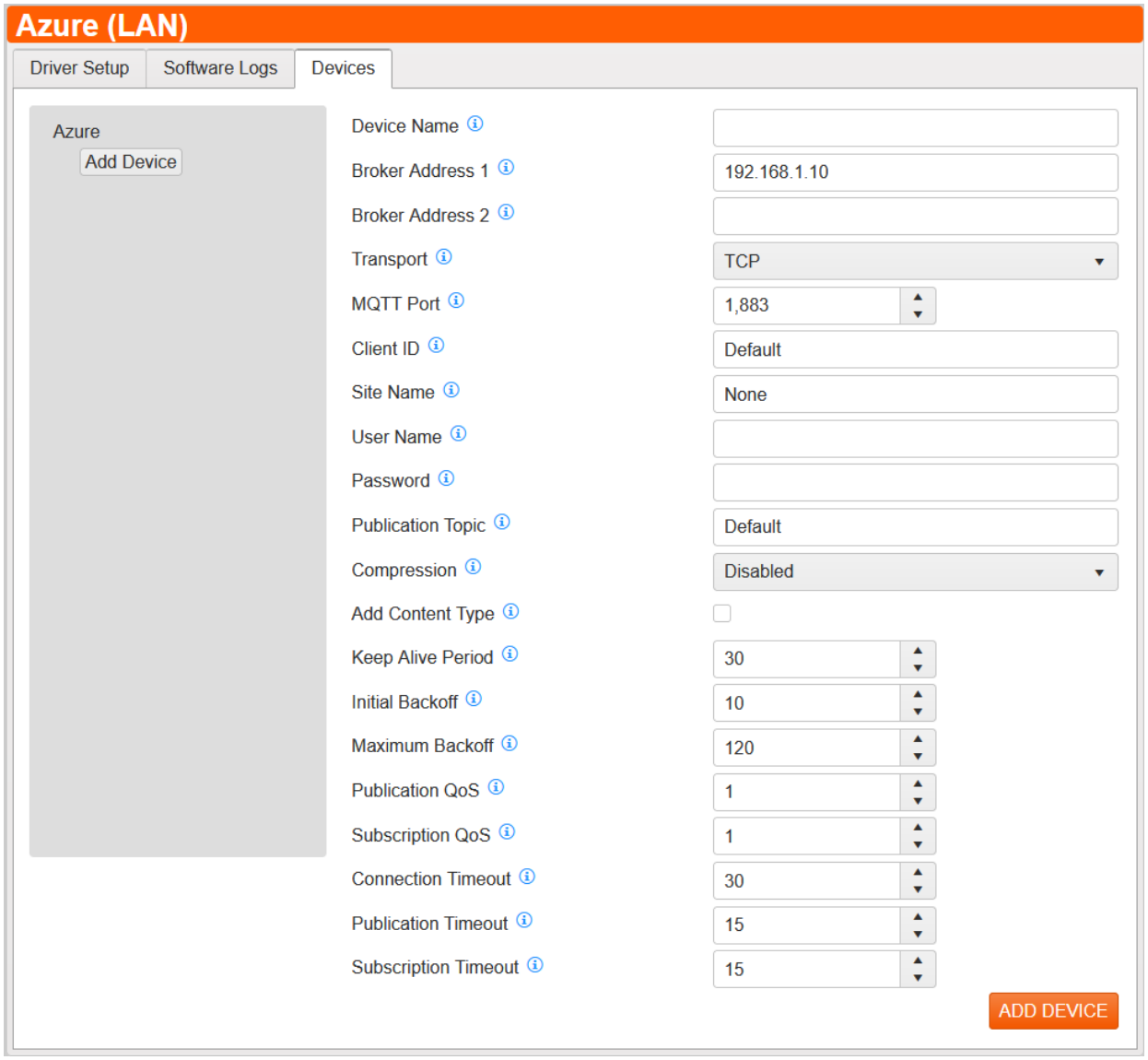
Communicate with your IoT hub using the MQTT protocol

Microsoft URL: [Understand Azure IoT Hub MQTT support | Microsoft Docs]

Initial configuration screen for the Azure IoT Hub configuration in the mCore using the Generic MQTT protocol.

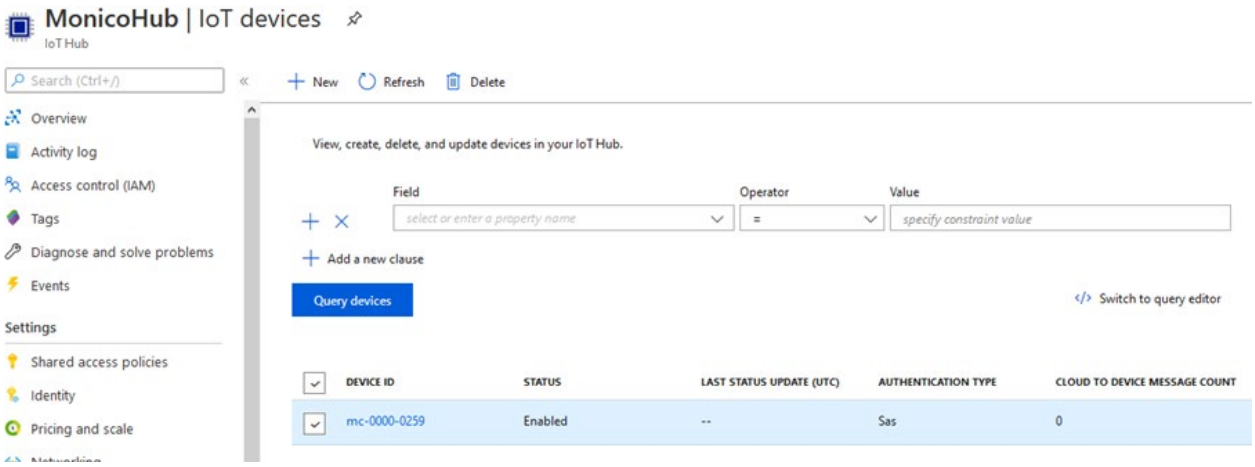
Transport and MQTT Port

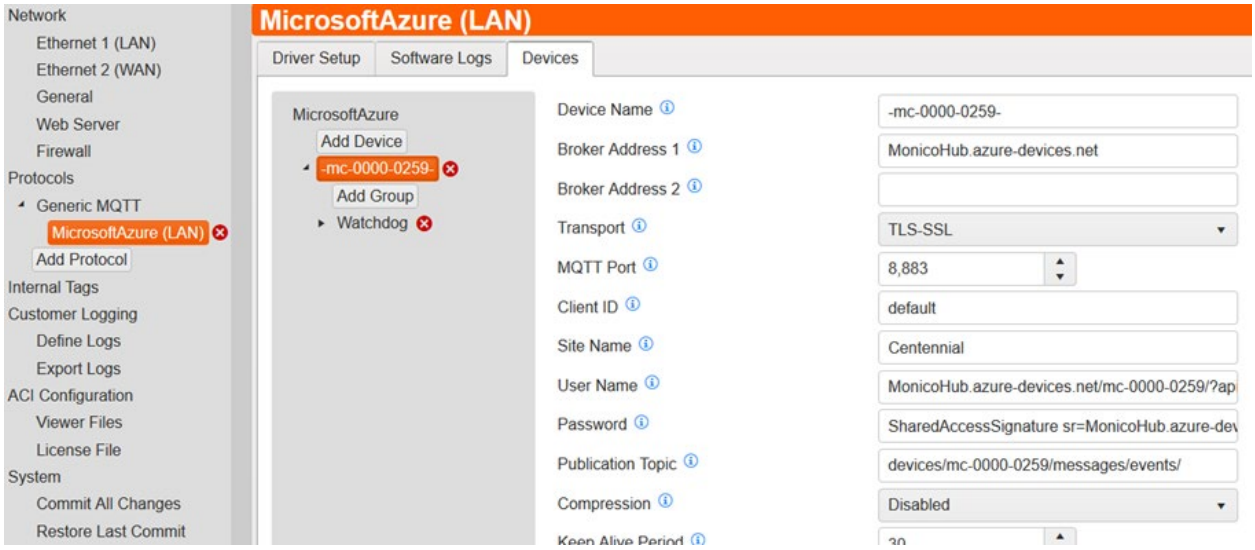
1. The Transport will be TLS-SSL and the MQTT Port will be 8883 for connections to Azure.



Client ID

- For the **ClientId** field, use the **deviceId** from Azure. The deviceId is created in Azure when you add the device. You can add a device using the mCore serial number if desired. This helps because when you configure the MQTT protocol in mCore, the Client ID defaults to using the serial number of the device. Bottom line is that the deviceId in Azure MUST match the Device ID field on the mCore and they must be a unique value from any other device.





Username

- For the **Username** field, use this format:

  `{iothubhostname}/{device_id}/?api-version=2018-06-30`,

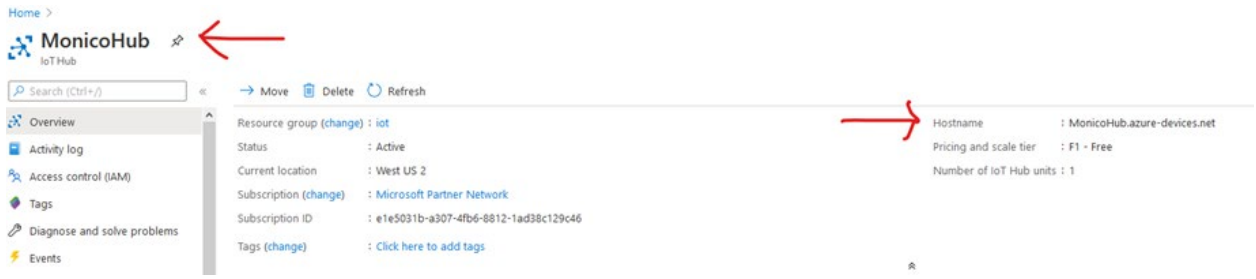
where `{iothubhostname}` is the full Hostname of the IoT hub, in this example, it is ***MonicoHub.azure-devices.net***, the name we used in Azure to create the IoT hub.

For example, if the name of your IoT hub is **contoso.azure-devices.net** and if the name of your device is **MyDevice01**, the full **Username** field should contain:

```
contoso.azure-devices.net/MyDevice01/?api-version=2018-06-30
```

Or in our example:

```
MonicoHub.azure-devices.net/mc-0000-0259/?api-version=2018-06-30
```



Password

- For the **Password** field, use a SAS token. The format of the SAS token is the same as for both the HTTPS and AMQP protocols:

  `SharedAccessSignature sig={signature-string}&se={expiry}&sr={URL-encoded-resourceURI}`

  This is where things get complicated. To create the SAS token, you need to install software, either Microsoft Visual Studio Code editor (it's a free editor from Microsoft) and an Azure IOT tool package or the Azure CLI software. The easiest method is the Azure CLI. Visual Studio Code also has the ability to view IoT Hub data at the endpoint to confirm that the mCore is communicating with the Azure IoT Hub.

  Install Azure CLI on Windows –

  Microsoft URL: Install the Azure CLI for Windows | Microsoft Docs

  Generate a SAS token using Azure CLI –

  Microsoft URL: az iot hub | Microsoft Docs

  For Azure IoT Tools

1. Expand the **AZURE IOT HUB DEVICES** tab in the bottom left corner of Visual Studio Code.
2. Right-click your device and select **Generate SAS Token for Device**.
3. Set **expiration time** and press 'Enter'.
4. The SAS token is created and copied to clipboard.

   The SAS token that's generated has the following structure:

   ```
   HostName={your hub name}.azure-
   devices.net;DeviceId=javadevice;SharedAccessSignature=SharedAccessSigna
   ture sr={your hub name}.azure-devices.net%2Fdevices%2FMyDevice01%2Fapi-
   version%3D2016-11-14&sig=vSgHBMUG.....Ntg%3d&se=1456481802
   ```

   The part of this token to use as the **Password** field to connect using MQTT starts with SharedAccessSignature:

   ```
   SharedAccessSignature sr={your hub name}.azure-
   devices.net%2Fdevices%2FMyDevice01%2Fapi-version%3D2016-11-
   14&sig=vSgHBMUG.....Ntg%3d&se=1456481802
   ```

NOTE: **SAS tokens created in Visual Studio Code have an expiration date and will need to be renewed from time to time. The expiration time is in hours, so 8760 hours is equal to one year. Multiple 8760 according for how many years the token is valid.**

Here are screen outputs of the token generation and MQTT being monitored in real time using Visual Studio Code editor.

```
[SASToken] SAS token for [mc-0000-0259] is generated and copied to clipboard:

SharedAccessSignature sr=MonicoHub.azure-devices.net%2Fdevices%2Fmc-0000-
0259&sig=LPuzDomwgkmWE63wuGnSs80w%2Fd0Wxi%2BbcU3MqgtGd0o%3D&se=1624539884



[IoTHubMonitor] [8:49:58 AM] Message received from [mc-0000-0259]:

{
  "device": {
    "siteName": "Centennial"
  },
  "tags": {
    "One_Second_Watchdog": "76",
    "timestamp": "1593010198.227",
    "historic": false,
    "adhoc": false
  },
  "online": true,
  "timestamp": "1593010198.228"
}
[IoTHubMonitor] [8:49:59 AM] Message received from [mc-0000-0259]:

{
  "device": {
    "siteName": "Centennial"
  },
  "tags": {
    "One_Second_Watchdog": "77",
```

```
    "timestamp": "1593010199.242",
    "historic": false,
    "adhoc": false
  },
  "online": true,
  "timestamp": "1593010199.243"
}
[IoTHubMonitor] [8:49:59 AM] Message received from [mc-0000-0259]:
{
  "body": {
    "online": false
  },
  "applicationProperties": {
    "iothub-MessageType": "Will"
  }
}
[IoTHubMonitor] Stopping built-in event endpoint monitoring...
```

Publication Topic

1. As far as the Topic goes for publishing, there are also set requirements, it's not arbitrary. There are built in 'endpoints' in Azure that can be used, the default one is called **events**. The full path for the topic 'events' is:

   `devices/{device_id}/messages/events/`

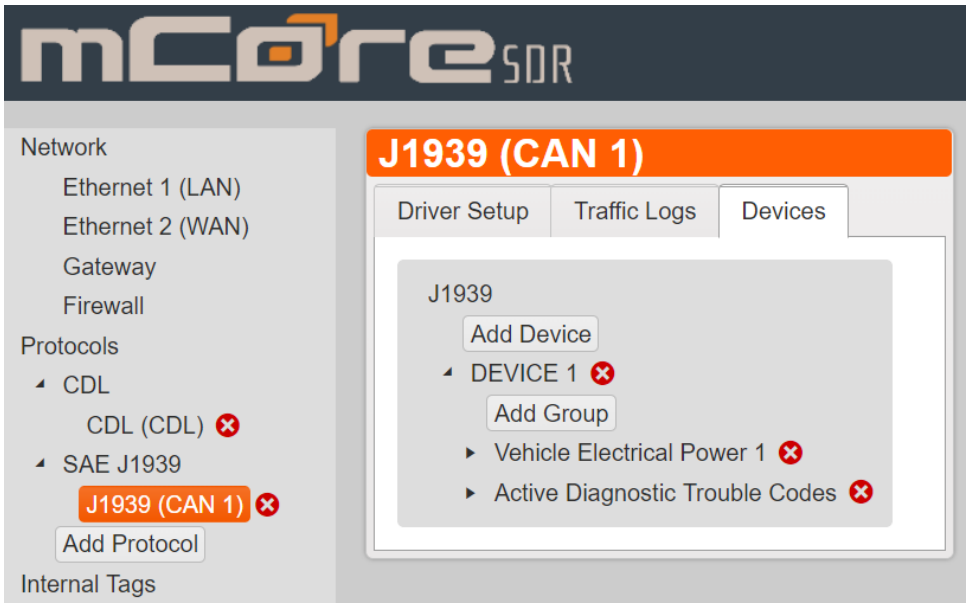
   In our example:

   `devices/mc-0000-0259/messages/events/`

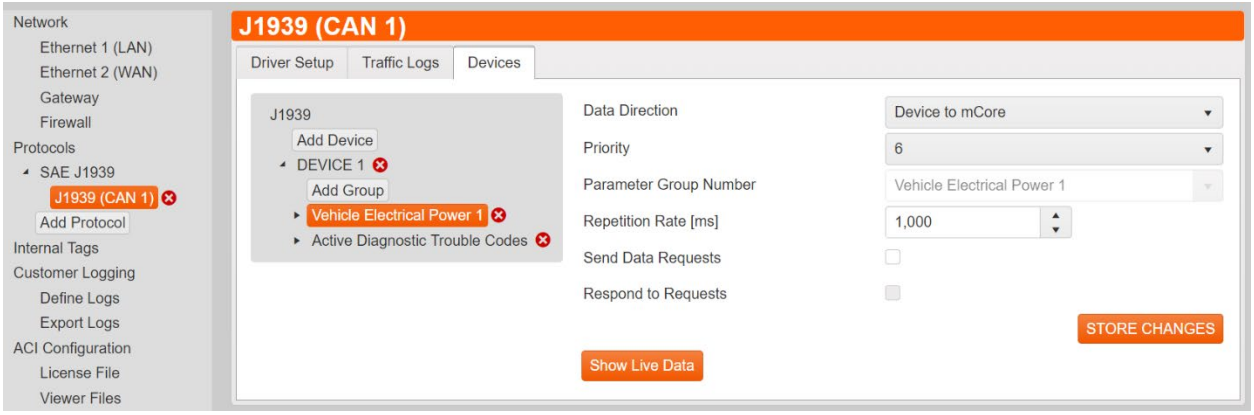
SECTION 9: LIVE DATA VIEWER

Live Data Viewer is a tool used to view live data either when connected to mCore locally or remotely using an internet browser. Once you've added a group with mapped out parameters and committed all changes, you can view live data. ***To be clear, the bottom left-hand menu must read COMMITTED in order to view live data***. Refresh rates are typically between .5s to 1s.

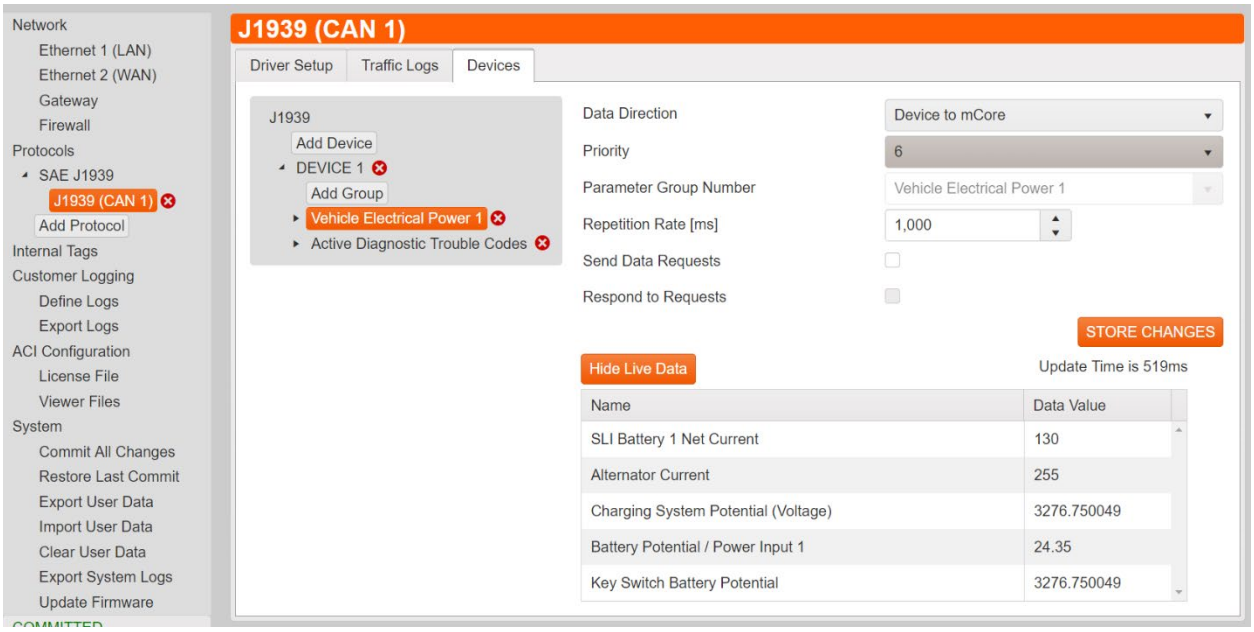
1. To view live data, select any protocol in the left menu. Click the secondary level to show the center menu, then click the Devices tab as shown below.



2. Next click the group, in this example the PGN name, that contains the parameters you would like to see data for. See example below.



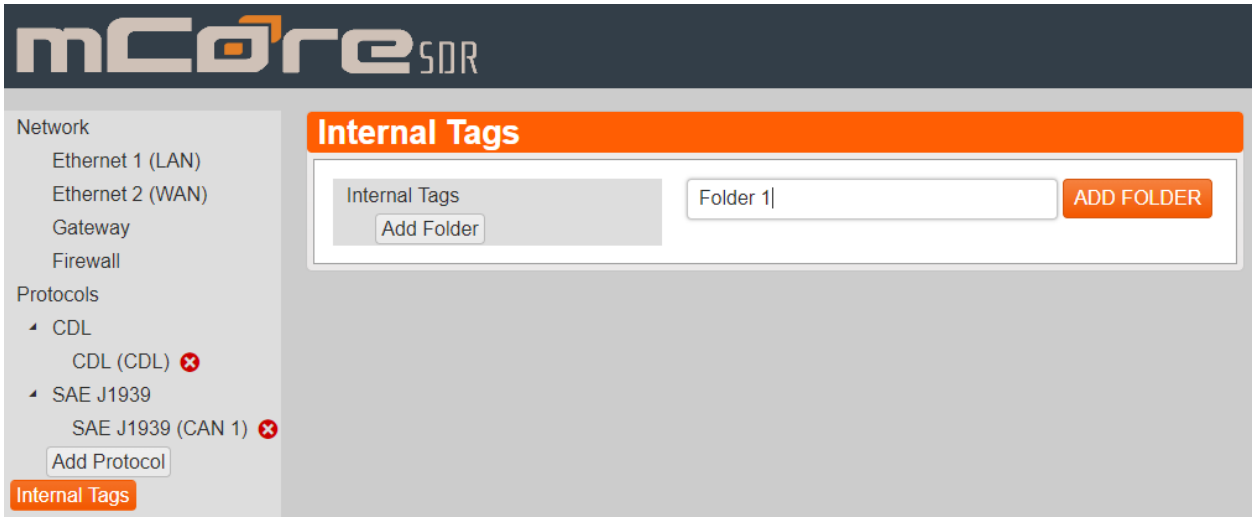
3. Click the Show Live Data button in the center menu toward the bottom.
   A table of the parameters and values will appear. See example below.



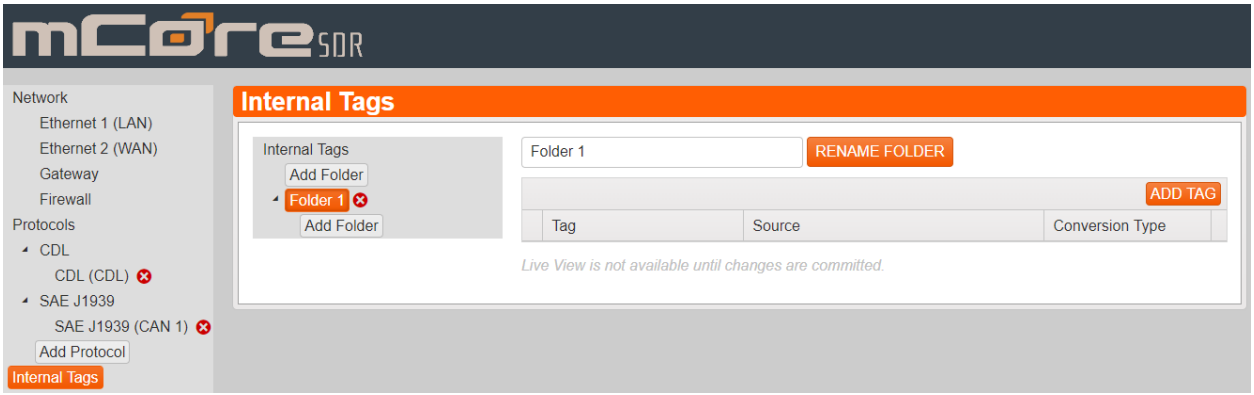
4. To stop watching live data, click the Hide Live Data button in the center menu.

SECTION 10: INTERNAL TAGS

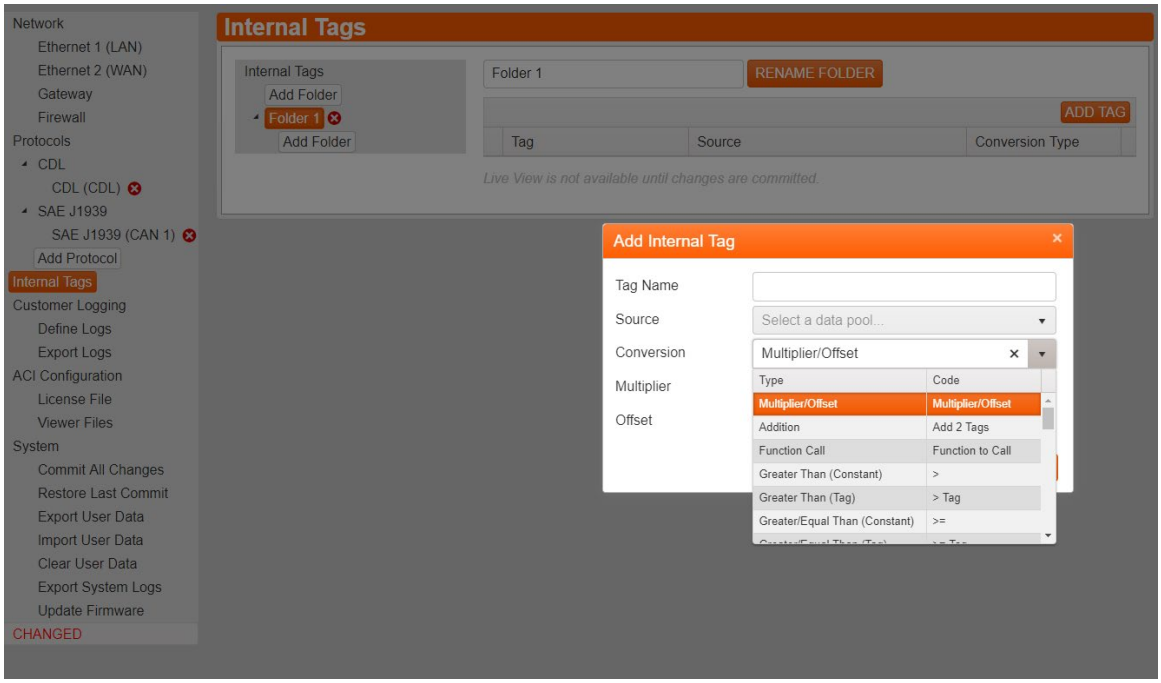
1. Click the Internal Tags button in the left menu. The following screen will appear. Click Add Folder and input a name for the folder. This will serve as an organizational tool. Click the orange Add Folder button to the right of the name box to save and create the folder.



2. The following screen will appear where you can start adding tags. Click the Add Tag button located to the right of the screen.



3. Once you've clicked Add Tag, the following pop up will appear. Click the Conversion drop down menu to see all the different functions that can be used including multiplier/offset for scaling and unit conversion, add 2 tags together, etc. When done, click the Store Changes button in the pop-up window.



Note: See Section 10.1: Function Table for examples and full list of functions. Contact Customer Support if you have questions about any of the functions.

4. Once you've stored that change, you will see the internal tag in the folder's line up. Repeat Step 3 to add as many internal tags as desired.

5. In order to create multiple step logic, take the resulting tag from the first step and use it as the source for the second step and so on.

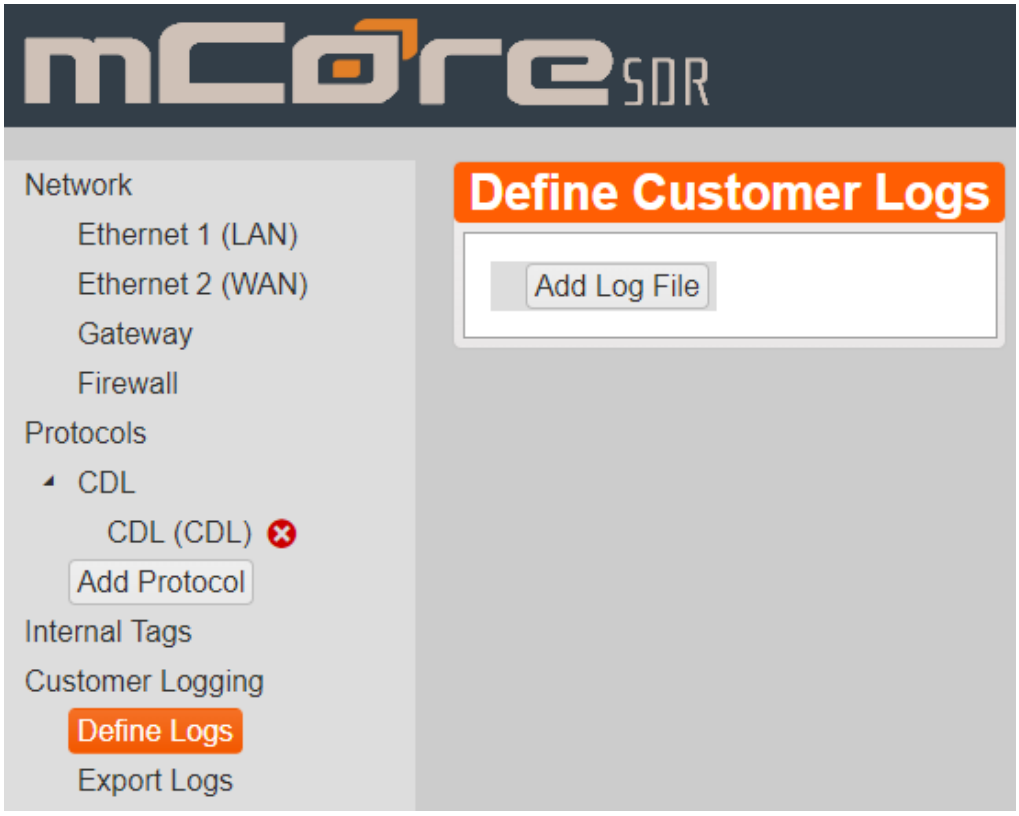
SECTION 10.1: FUNCTION TABLE

Label	Function		Label	Function
Multiplier/Offset	(m*Tag1) + o		Bitwise AND (Constant)	Tag1 & k
Reverse Multiplier/Offset	(Tag1 – o) / m		Bitwise AND (Tag)	Tag1 & Tag2
Addition (Constant)	Tag1 + k		Bitwise OR (Constant)	Tag1 | k
Addition (Tag)	Tag1 + Tag2		Bitwise OR (Tag)	Tag1 | Tag2
Subtraction (Constant)	Tag1 - k		Bitwise XOR (Constant)	Tag1 ^ k
Subtraction (Tag)	Tag1 - Tag2		Bitwise XOR (Tag)	Tag1 ^ Tag2
Find Percentage (Constant)	Tag1 / k * 100		Bitwise Shift Left (Constant)	Tag1 << k
Find Percentage (Tag)	Tag1/Tag2 * 100		Bitwise Shift Left (Tag)	Tag1 << Tag2
Take Percentage (Constant)	Tag1 * k / 100		Bitwise Shift Right (Constant)	Tag1 >> k
Take Percentage (Tag)	Tag1 * Tag2 /100		Bitwise Shift Right (Tag)	Tag1 >> Tag2
Remainder (Constant)	Tag1 % k		Reverse Bits	n/a
Remainder (Tag)	Tag1 % Tag2		Reverse Bytes	n/a
Greater Than (Constant)	Tag1 > k		Reverse Words	n/a
Greater Than (Tag)	Tag1 > Tag2			
Greater Than or Equal (Constant)	Tag1 >= k			
Greater Than or Equal (Tag)	Tag1 >= Tag2			
Less Than (Constant)	Tag1 < k			
Less Than (Tag)	Tag1 < Tag2			
Less Than or Equal (Constant)	Tag1 <= k			
Less Than or Equal (Tag)	Tag1 <= Tag2			
Data Match (Constant)	Tag1 = k			
Data Match (Tag)	Tag1 = Tag2			
Data Mismatch (Constant)	Tag1 != k			
Data Mismatch (Tag)	Tag1 != Tag2			
Logical Inverse (NOT)	! Tag1			
Logical AND (Tag)	Tag1 && Tag2			
Logical OR (Tag)	Tag1 || Tag2			
Invert Bits	~ Tag1			

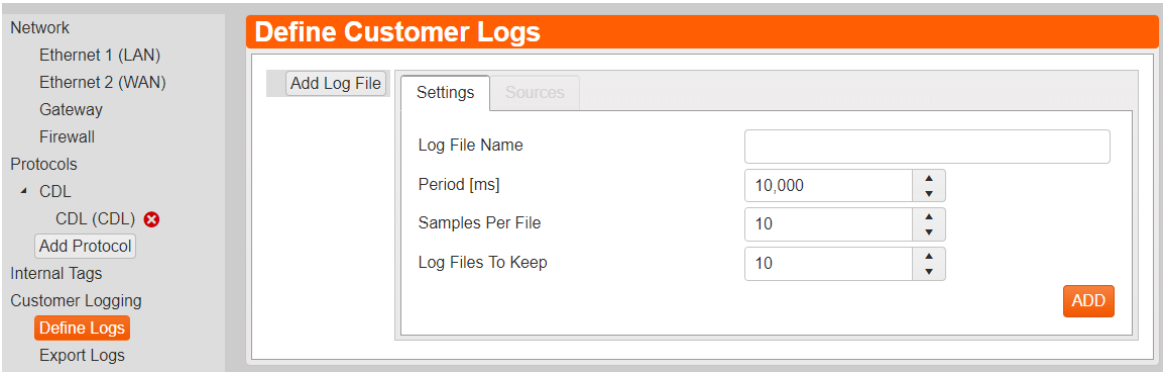
Click to Return to Top of Document

SECTION 11: CUSTOMER LOGGING

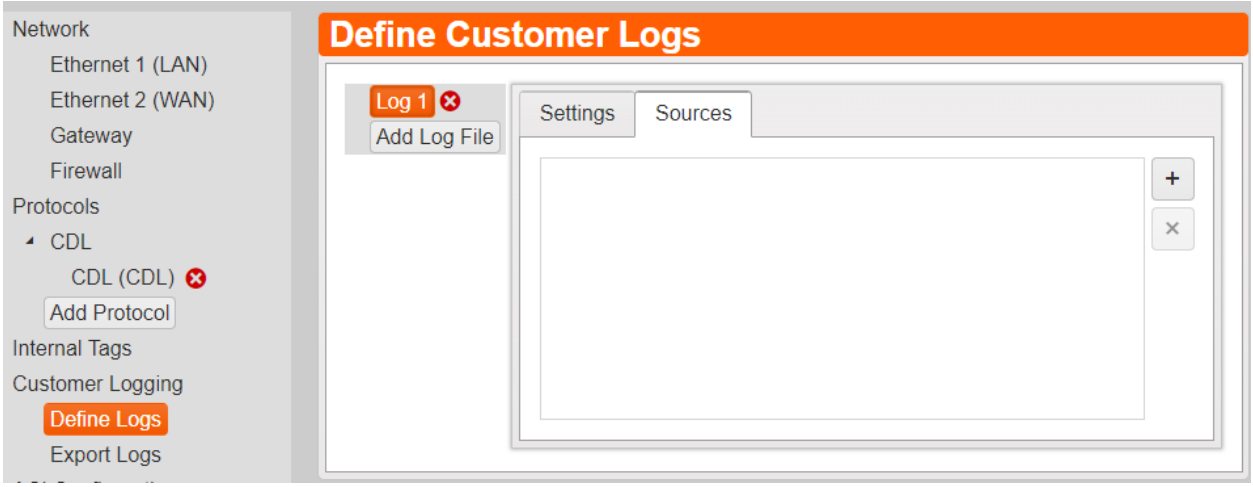
1. Click the Customer Logging button in the left menu. This will bring you to the following screen.



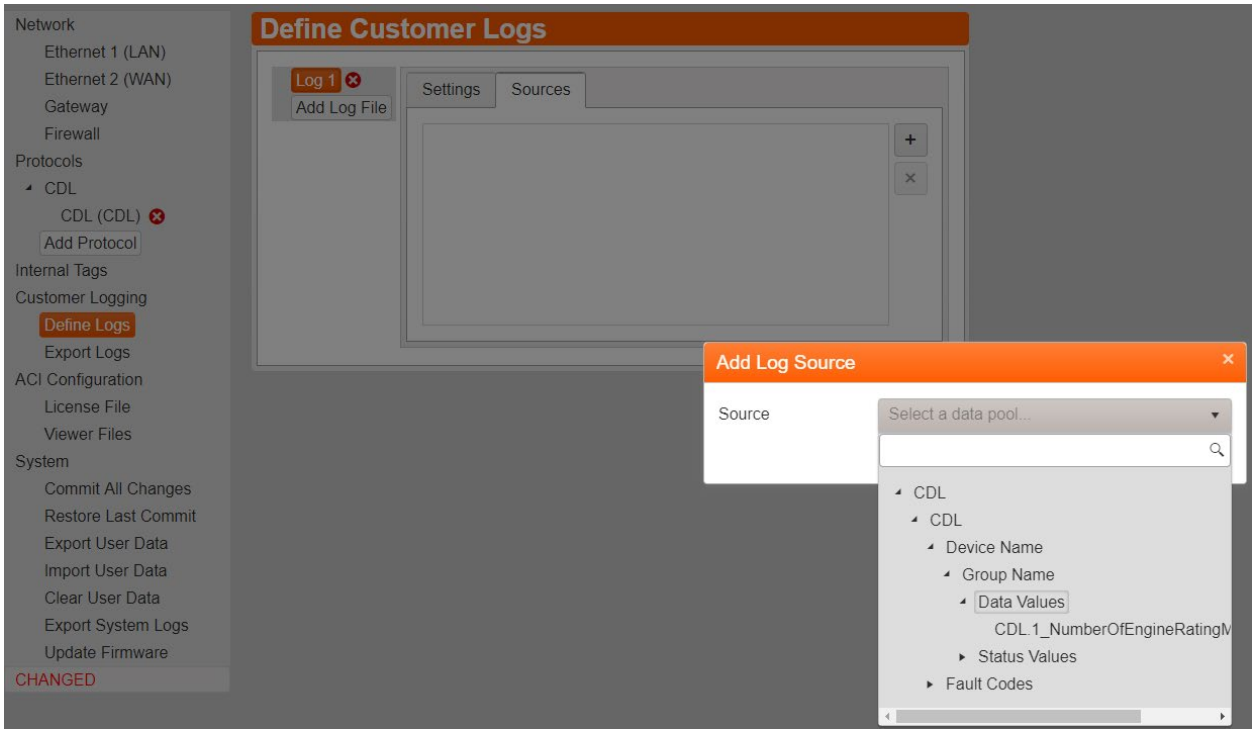
2. Click the Add Log File button. Here you can name the log file and adjust the logging settings. The Period is how often, in milliseconds, mCore will record the values of all tags listed for logging. The Samples Per File is the number of samples mCore will keep in one file before it creates and start saving to another file. The Log Files To Keep is the number of total files to save. These files will be overwritten once the last file is full. This means if you have 10 log files, once #10 is full, #1 will be overwritten and the logging will continue. Click Add when done.



3. When you click add, the Sources tab will appear. Click the Sources tab. This is where you will add the tags you want to log. Click the + in the top right to add a tag.



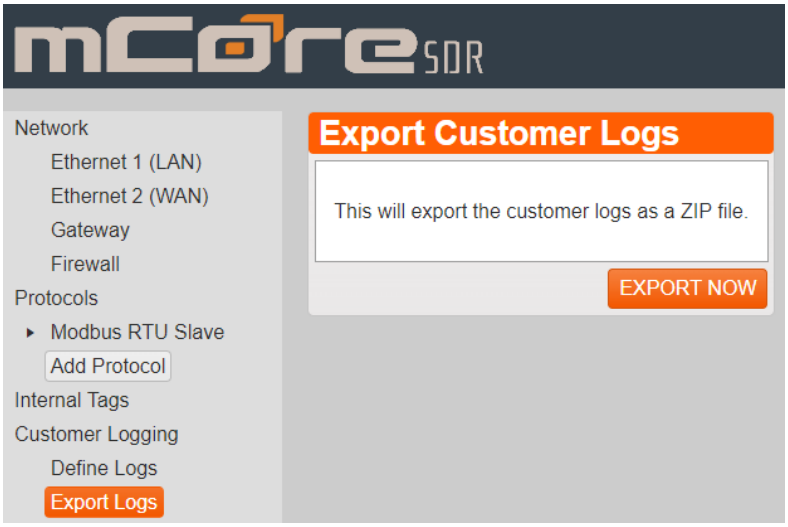
4. The following pop up box will appear. You can either use the search bar or navigate through the tree structure to find the tag you want to add. Click the Add button in the pop-up box after selecting the desired tag.



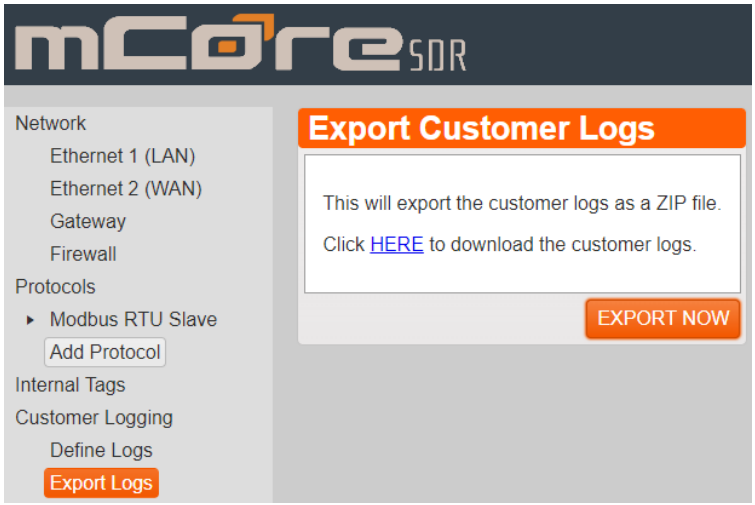
5. Repeat Step 4 to add all desired tags to be logged. You must then click the Commit All Changes button in the left menu towards the bottom for the change to take effect.

SECTION 11.1: EXPORT CUSTOMER LOGS

1. Click Export Logs in the left menu under the Customer Logging section. The following screen will appear.



2. Click the Export Now button in the bottom right. The following "Click HERE" blue link will appear. Click this link. A pop-up box will appear asking where you want to save the zip file and what to name it.



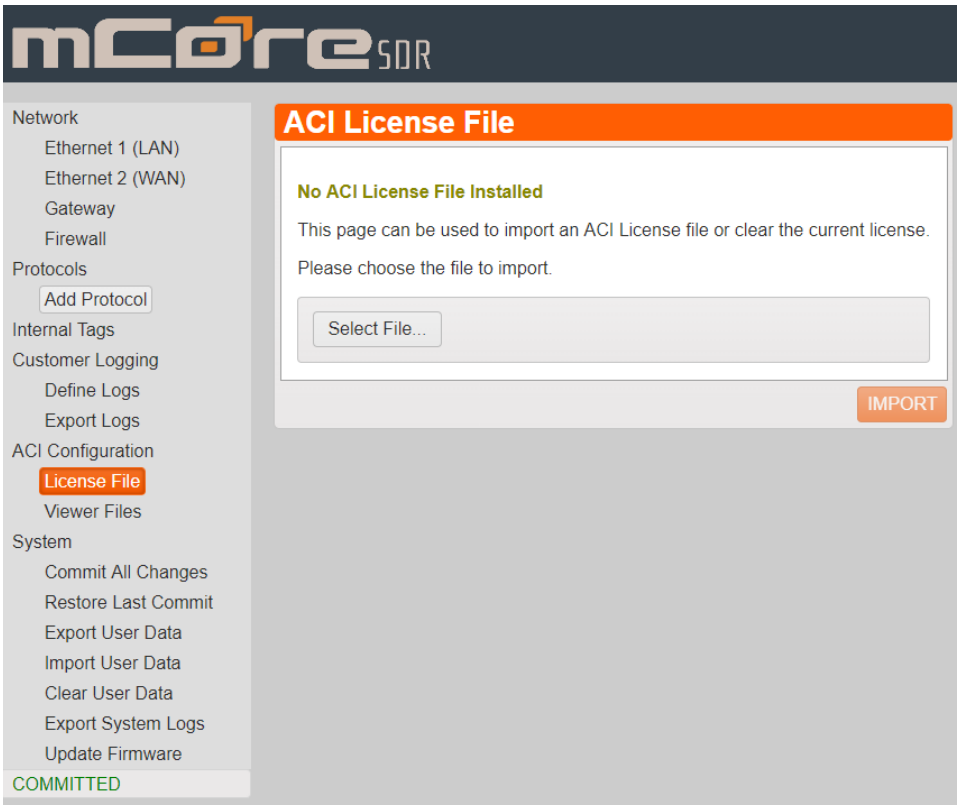
3. Once the file is saved, navigate to the location that you saved the zip file on your computer. Most Windows computers will automatically unzip files for you if you double click on the zip file. If this does not work, try right clicking the file and selecting Extract All. This will pull all files out of the zip file and put them in the same folder/directory that you saved the zip file in.

SECTION 12: ACI ERCM™ EXPRESS CONFIGURATION

Monico has partnered with ACI to provide the hardware for running eRCM Express™ Compressor Modeling software. This program communicates with a PLC and runs the modeling software using compressor and engine data. mCore communicates with the PLCs either over Modbus TCP or using our AB Tag Client to speak directly to an Allen Bradley PLC using Tag Arrays.

SECTION 12.1: ADDING LICENSE FILE

1. Find the ACI Configuration section in the left-hand menu. Click License File. The following screen will appear. This screen will tell you if a license file is installed, what the name of the file is if one is installed, and whether it matches the MAC address of the mCore. The License files are unique per mCore and cannot be used in other mCores.

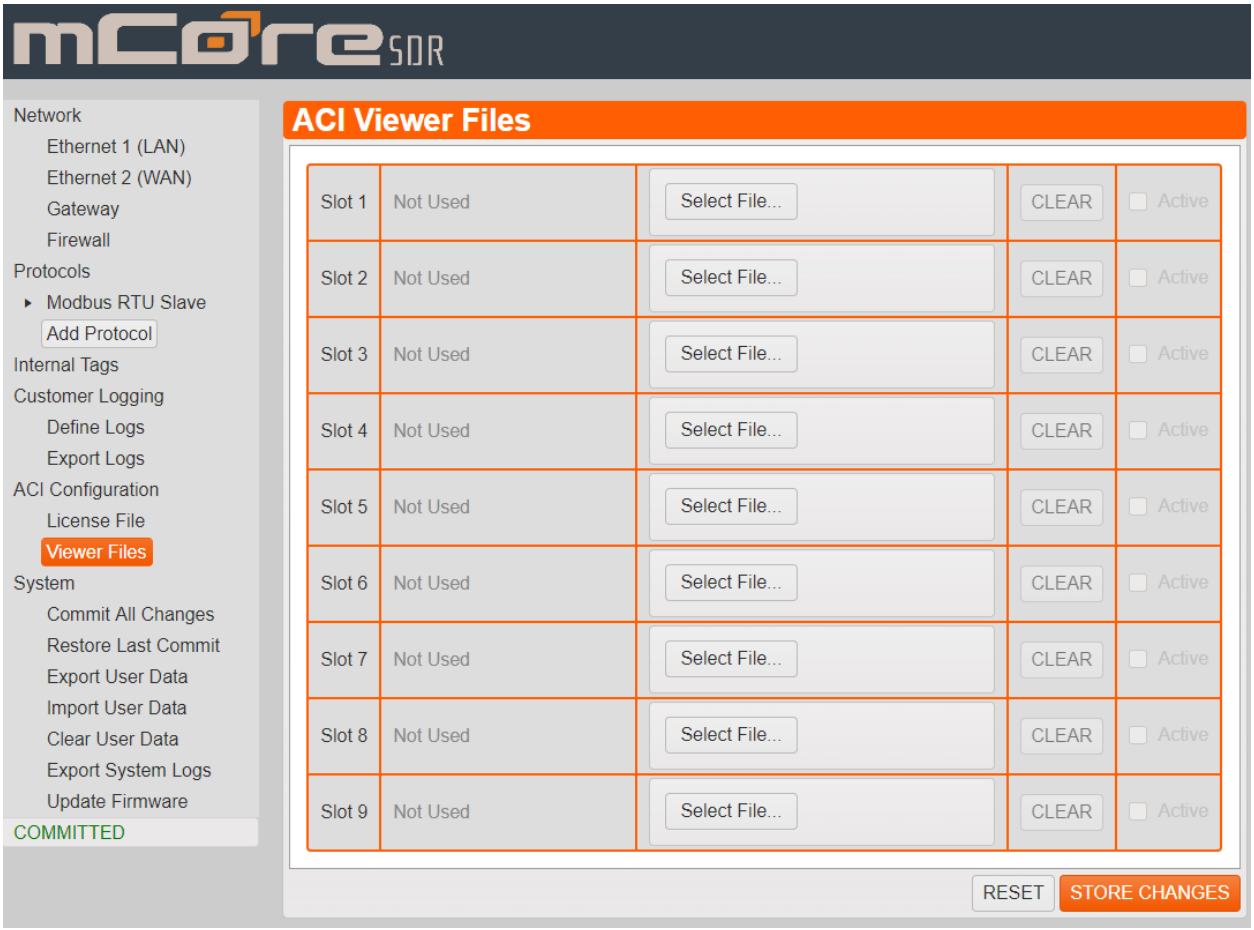


2. To add a License file from your computer, click the Select File button. A pop up will appear with File Explorer. Navigate to where the License File is located on your computer, select it, then click Open in the bottom right.

3. Once the file has finished uploading, click the Import button in the center menu towards the bottom right. A pop up will appear. Click Ok to continue.

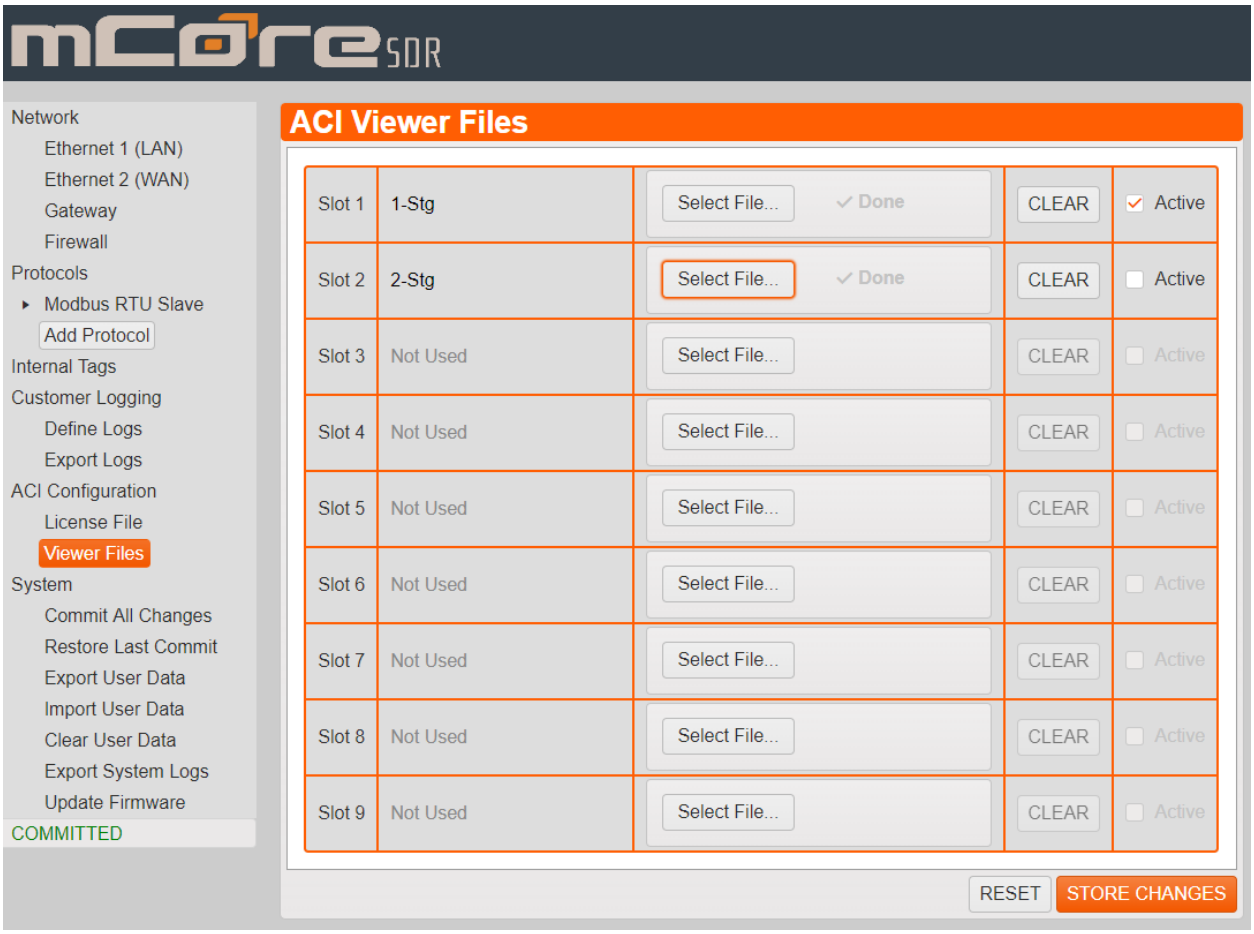
SECTION 12.2: VIEWER FILE(S)

1. Click Viewer Files under ACI Configuration in the left menu. The following screen will appear.



2. Click one of the Select File buttons. A pop up will appear with File Explorer. Navigate to where the Viewer File is located on your computer, select it, then click Open in the bottom right.

3. The Viewer File will then appear in the screen shown below. Click the check box next to the Viewer File that you want to be active. Multiple Viewer files can be present at one time but only one can be set to active at a time. However, if only 1 file is uploaded and clicked to be Active, it will not be able to be set to inactive until at least a 2nd Viewer file is uploaded.



4. Once you've clicked a check box to set a Viewer file active, click the Store Changes button in the bottom right. A pop up will appear asking if you are sure you want the recently selected Viewer file to be active. Click Ok to continue. mCore will now be fully setup for ERCM Express and will now start running calculations.

SECTION 12.3: MODBUS TCP SLAVE – PLC SETUP AND SEQUENCE OF OPERATIONS – VERSION 1.0 AND 1.1 LEGACY

Note: mCore will not perform calculations unless a License file AND at least 1 Viewer file is uploaded, set to active, and clicked Store Changes.

This version will work the same way as the older firmware version of mCore running V.5 or V1.0. The next section uses a different and improved method that requires a different User Data file.

1. Setup the PLC to write the compressor values to the corresponding registers (40K Reals/Floats) per the Data Map's Writes tab.
2. Once all the compressor values are written and you are ready to start the calculations, write a 1 (40k Real/Float) to either register 400075 or 400273.
3. When calculations are done, mCore will set the force calculations registers back to zero. When the force calculations register(s) gets put back to a zero, have the PLC watching for it to turn to zero and have this trigger the PLC to read the new calculated values.
4. Repeat Steps 1-3 as often as you want to get new calculated values. Typical calculation time is 500ms or .5s.

SECTION 12.4: MODBUS TCP SLAVE – PLC SETUP AND SEQUENCE OF OPERATIONS – VERSION 1.1

Note: mCore will not perform calculations unless a License file AND at least 1 Viewer file is uploaded, set to active, and clicked Store Changes.

1. Setup the PLC to write the compressor values to the corresponding registers (40k Reals/Floats) per the Data Map's Writes tab.
2. Once all the compressor values are written and you are ready to start the calculations, write a 1 to register 403649 (40k word/16-bit integer).
3. Register 403650 (40k word/16-bit integer) will go to a 1 indicating the calculation is in progress. Register 403651 will go to a 1 when the calculations are done.
4. Once the calculation is done, write a zero to register 403649 (40k word/16-bit integer).
5. Repeat steps 1-4 as often as you want to get new calculated values. Typical calculation time is 500 ms or .5s.

SECTION 12.5: AB TAG CLIENT - PLC SETUP AND SEQUENCE OF OPERATIONS

Note: mCore will not perform calculations unless a License file AND at least 1 Viewer file is uploaded, set to active, and clicked Store Changes.

1. The PLC Programmer will need to first setup their PLC with the following Tag Arrays. Pay special attention to the spelling, capitalization, and the use of underscores instead of spaces for each tag array name as they must be identical in the PLC program. A Data Map will be available to show what each array index is for what.

Array Name	Data Type [Quantity]
ACI_Inputs	REAL [137]
ACI_Control	REAL [25]
ACI_Misc	REAL [30]
ACI_Cond	REAL [30]
ACI_ErrorArray	DINT [100]
ACI_LoadArray	REAL [100]
ACI_FlowArray	REAL [100]
ACI_StageArray	REAL [180]
ACI_ThrowArray	REAL [300]
ACI_CylHEArray	REAL [400]
ACI_CylCEArray	REAL [400]
ACI_Ranges	REAL [25]
ACI_PLC_REQ	INT [4]
ACI_PLC_ACK	INT [4]

Click to Return to Top of Document

2. Have the PLC write a 1 to ACI_PLC_REQ[0] array index to tell mCore to run calculations. mCore will first read the compressor data from the PLC's ACI_Inputs array and then start the calculation process.

3. Then ACI_PLC_ACK[0] will go to a 1, meaning the calculations are running and to standby.

Then ACI_PLC_ACK[1] will go to a 1 when the calculation is done.

4. Once you receive the 1 from ACI_PLC_ACK[1] for the calculations being done, write a 0 to ACI_PLC_REQ[0], then wait for ACI_PLC_ACK[1] to go to 0.
5. Repeat Steps 2-5 as often as you want to run calculations. Typical calculation time is 500ms or .5s.

Note: A simplified approach in the ladder logic would be to set up the following logic and mCore will continuously calculate as fast as it can. This logic below will start the calculation, wait for the acknowledgement that the calculation is done, then will restart the process.

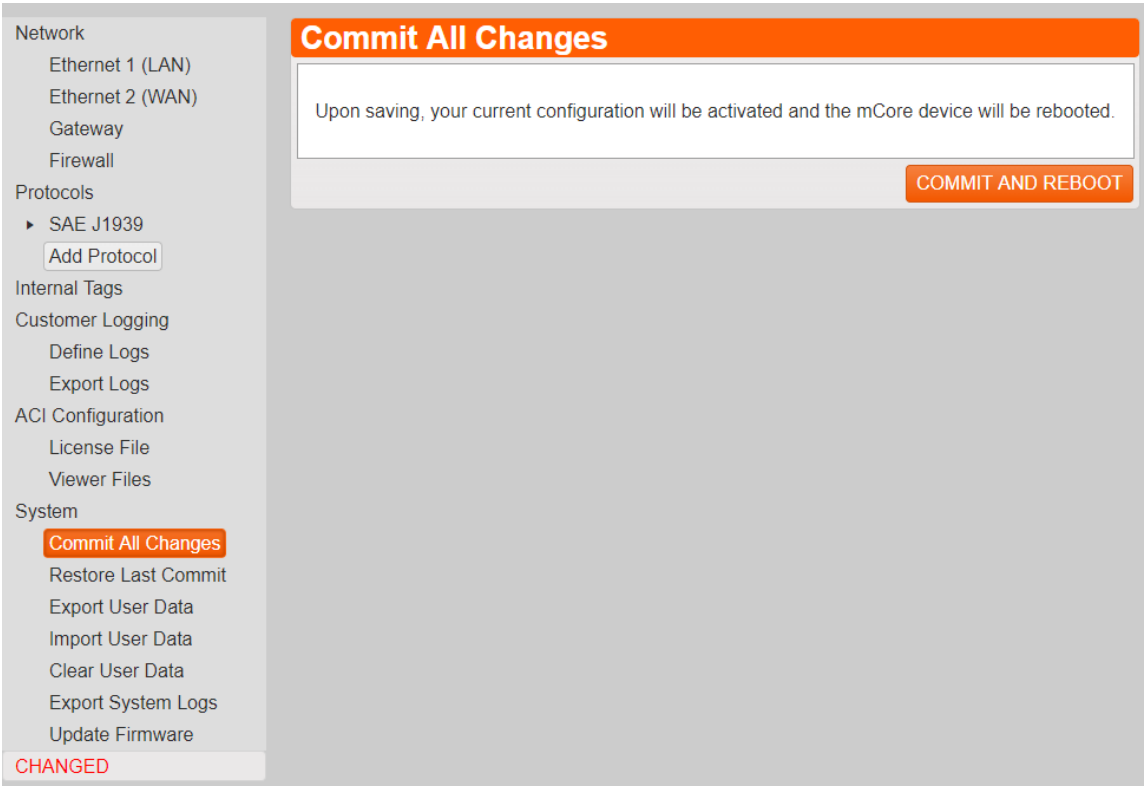
1. If ACI_PLC_ACK[1] is 0, then write a 1 to ACI_PLC_REQ[0].
2. When ACI_PLC_ACK[1] goes to a 1, write a 0 to ACI_PLC_REQ[0].
3. Repeat Steps 1-2. (If you want to slow down the calculation time then add a timer before repeating the steps.)

SECTION 13: SYSTEM FUNCTIONS

SECTION 13.1: COMMIT ALL CHANGES

The Commit All Changes button will implement the changes made to the configuration and reboot the mCore, depending on changes made and firmware revision level, to activate the new changes. In firmware version 1.1.xxx, mCore will reboot to implement configuration changes. In firmware version 1.2.xxx, mCore does not require reboot to commit changes, so it is much faster. This is the final step to configure an mCore. Please double check your IP Address settings prior to clicking the Commit All Changes button.

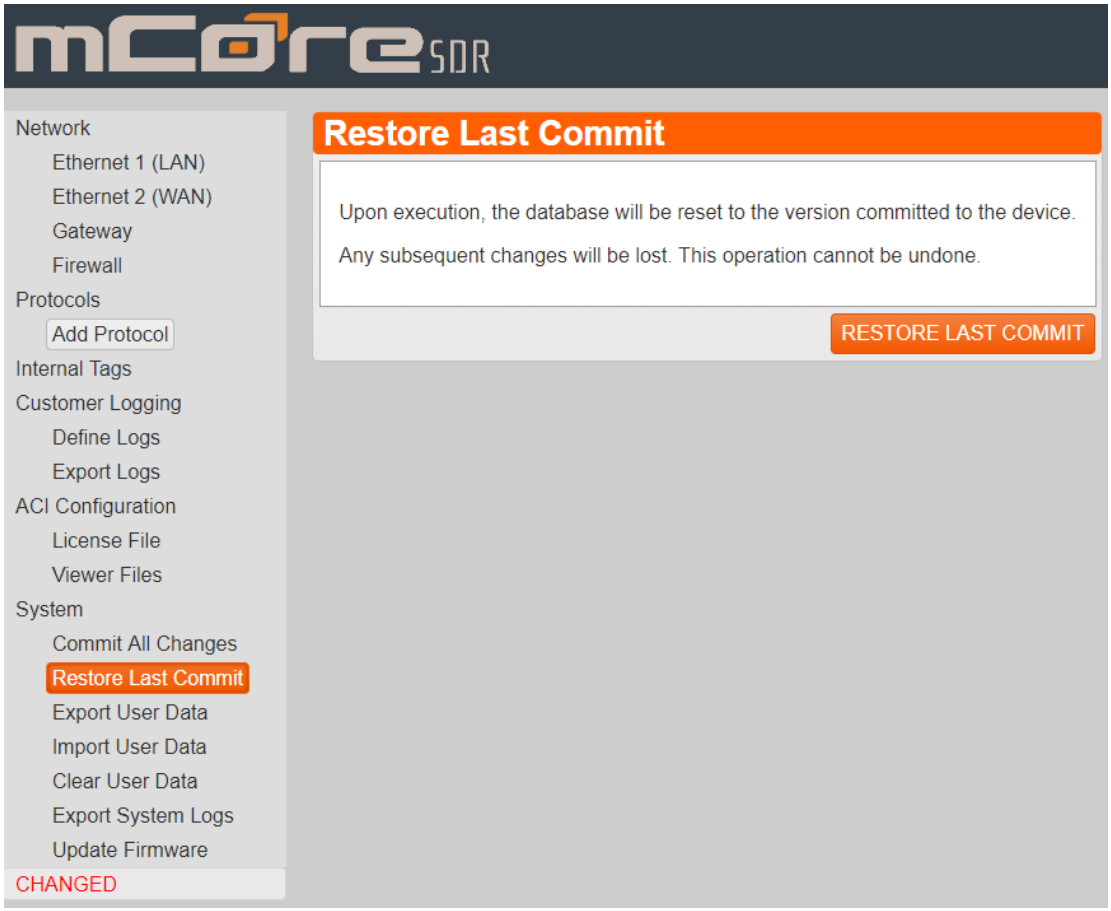
1. Once you've made your changes, click the Commit All Changes button in the left menu toward the bottom. The following screen will appear. Click the Commit and Reboot button in the center menu. A pop-up box will appear warning the user that the mCore will repeat and lose communications. Click Ok to continue.
2. Once you've clicked Ok, mCore will reboot and all changes will take effect.



SECTION 13.2: RESTORE LAST COMMIT

Restore Last Commit is essentially an undo button. It allows the user to go back to the last Committed configuration. Anything entered, changed, and/or stored will be lost and reset to what was Committed last.

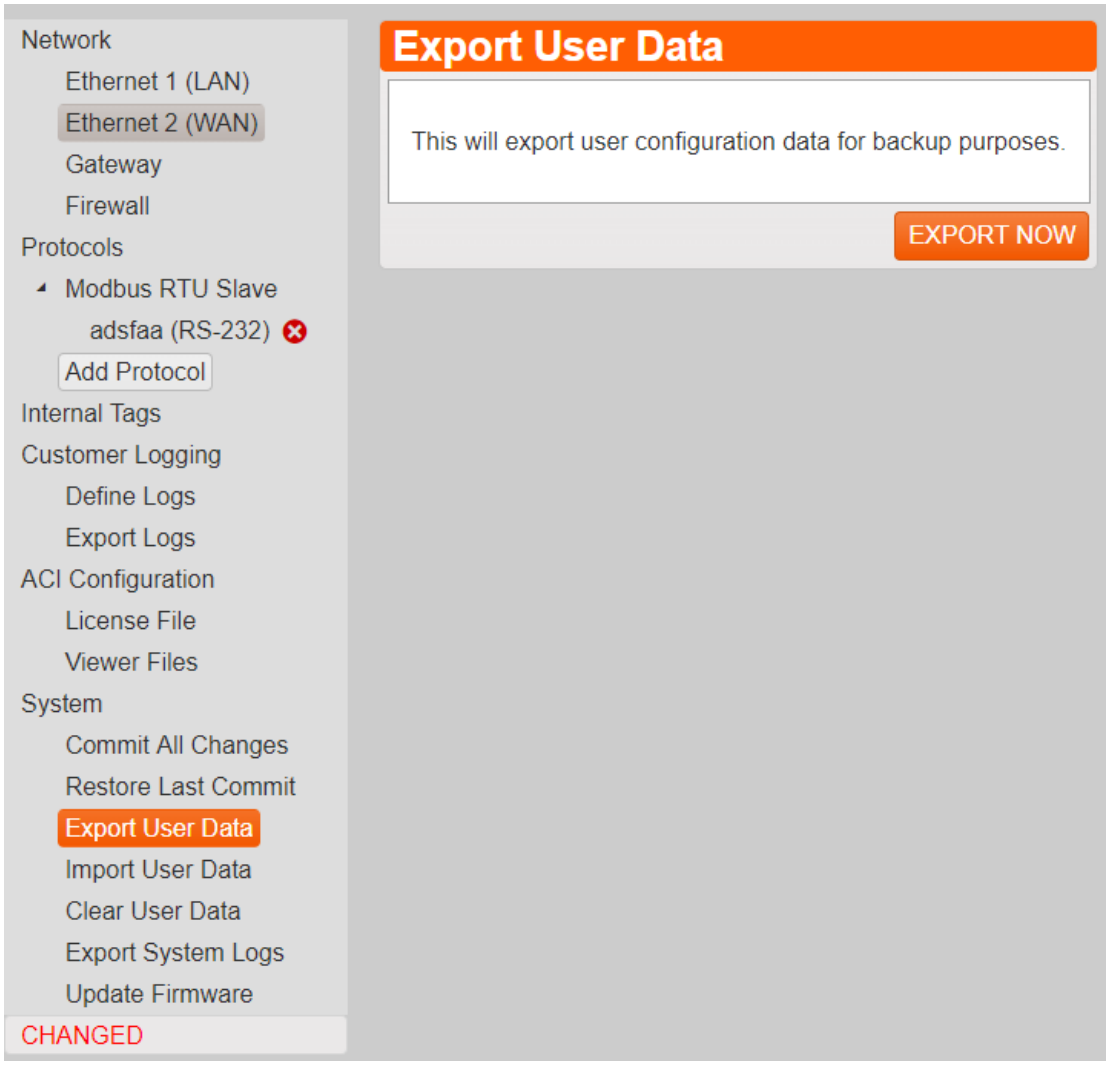
1. To Restore Last Commit, go to the left menu and towards the bottom. Click the Restore Last Commit button. A pop-up box will appear. Click Ok to continue.



SECTION 13.3: EXPORT USER DATA

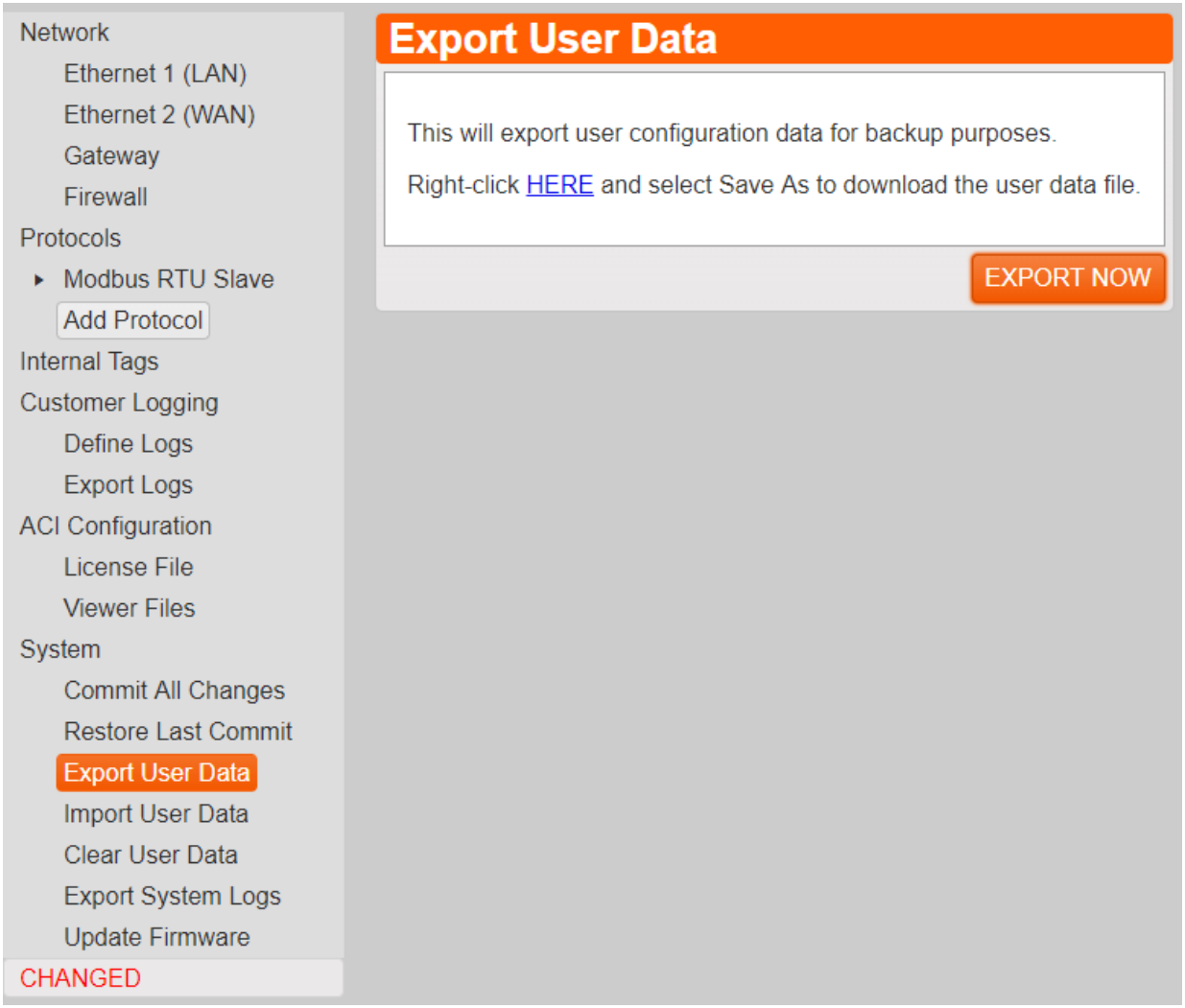
Exporting User Data will give the customer the ability to create a backup of their mCore's configuration/settings, share their setup for troubleshooting purposes, or even use it to setup another mCore that is being used for an identical application.

1. Click the Export User Data button in the left menu toward the bottom. The following screen will appear.



Click to Return to Top of Document

2. Click the Export Now button in the center menu. The following screen will appear.



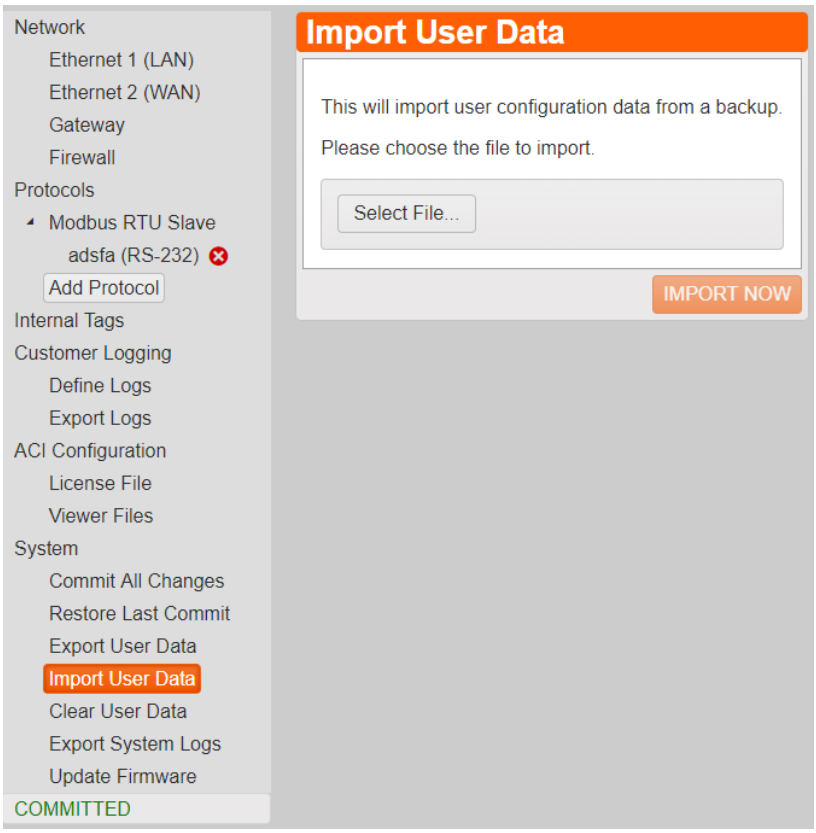
3. Click the blue link labeled “Here”. A pop-up box will appear asking where to save the User Data file and what you want to name it. Click Save when done.

Note: Keep in mind if this file is to be used on another mCore on the same network, the IP address and related network information of the old mCore will be in this User Data file. So before clicking Commit All Changes in the new mCore, check the IP credentials to make sure they are different from the previous mCore. If this is a replacement unit, then the IP credentials can stay as they are.

SECTION 13.4: IMPORT USER DATA

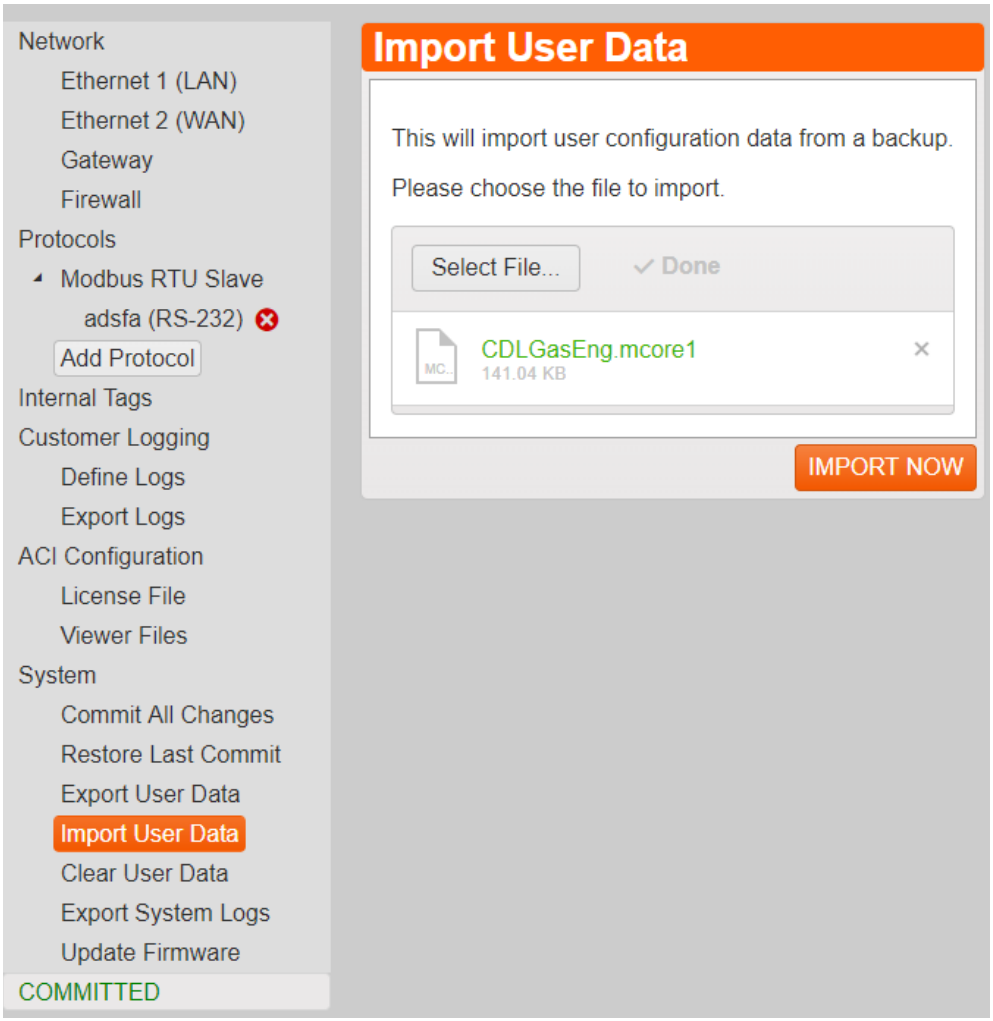
1. User Data is the configuration database that contains all the settings in the UI (User Interface). The user can import User Data files that were either saved as a backup previously or import User Data files sent with changes. Typical changes would include parameter mapping changes like for Modbus registers or even adding an entire protocol can be done quickly by being sent an already done User Data file and importing it.

2. The User Data file you would like to import must be saved to a location on your computer or on a reachable network drive. Click the Import User Data button in the left menu toward the bottom. The following screen will appear.



3. Click the Select File button in the center menu. A pop-up box will appear. Navigate to the saved User Data file, select it, then click Open. User Data files are the file type ".mcore1" on the end of the file.

4. Once the file is uploaded, it will show up in the center menu as shown below. Click the Import Now button at the bottom right. A pop up will appear asking if you want to continue with the change. Click Ok to continue.



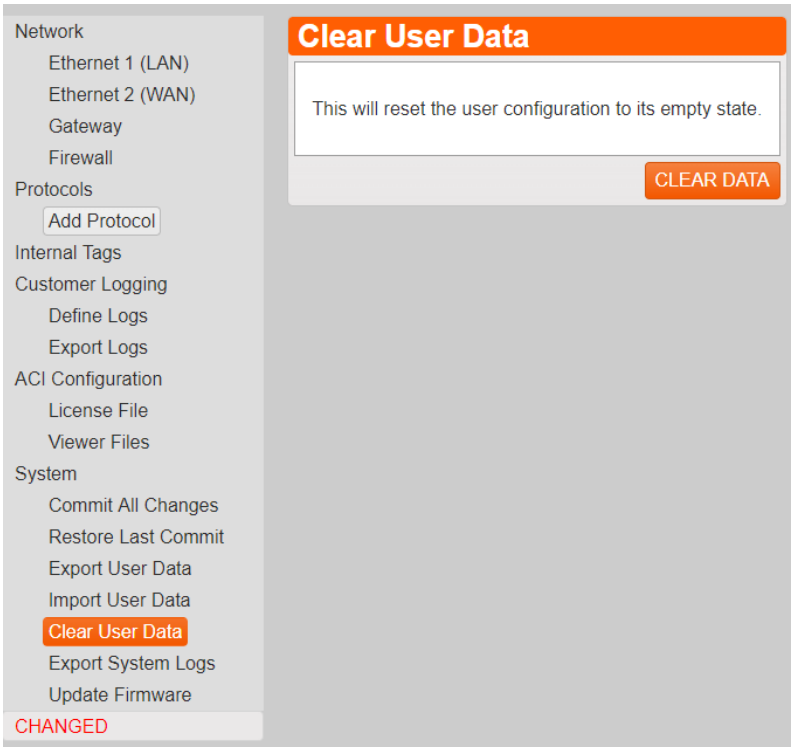
5. It may take up to about 10 seconds for the file to import and show all the information on the screen. Once it is done, you will see your new changes present. Double check your network settings after every User Data import. Since it copies all settings, it could have incorrect IP address settings copied from an office computer that are incompatible with the site network where the mCore is installed.

6. Once you've verified all the new settings, click the Commit All Changes button in the left menu toward the bottom. A pop-up box will appear asking if you want to continue with the changes and reboot the mCore so all the changes can take effect. Click Ok to apply the new changes.

7. After clicking Ok, the mCore will reboot and take around 15 to 20 seconds to reboot. If the UI (User Interface) does not come back to the Login page after 30 seconds, refresh your browser page by clicking the Refresh button. It is typically located either to the right or left of the address bar depending on the browser.

SECTION 13.5: CLEAR USER DATA

1. Clear User Data is a quick way to clear out the settings/configuration of the mCore to an empty state. This will remove all protocols, tags, and reset the IP credentials back to a factory reset of 192.168.10.220 on Ethernet 1 and DHCP for Ethernet 2.
2. Click Clear User Data in the left menu toward the bottom. The following screen will appear.



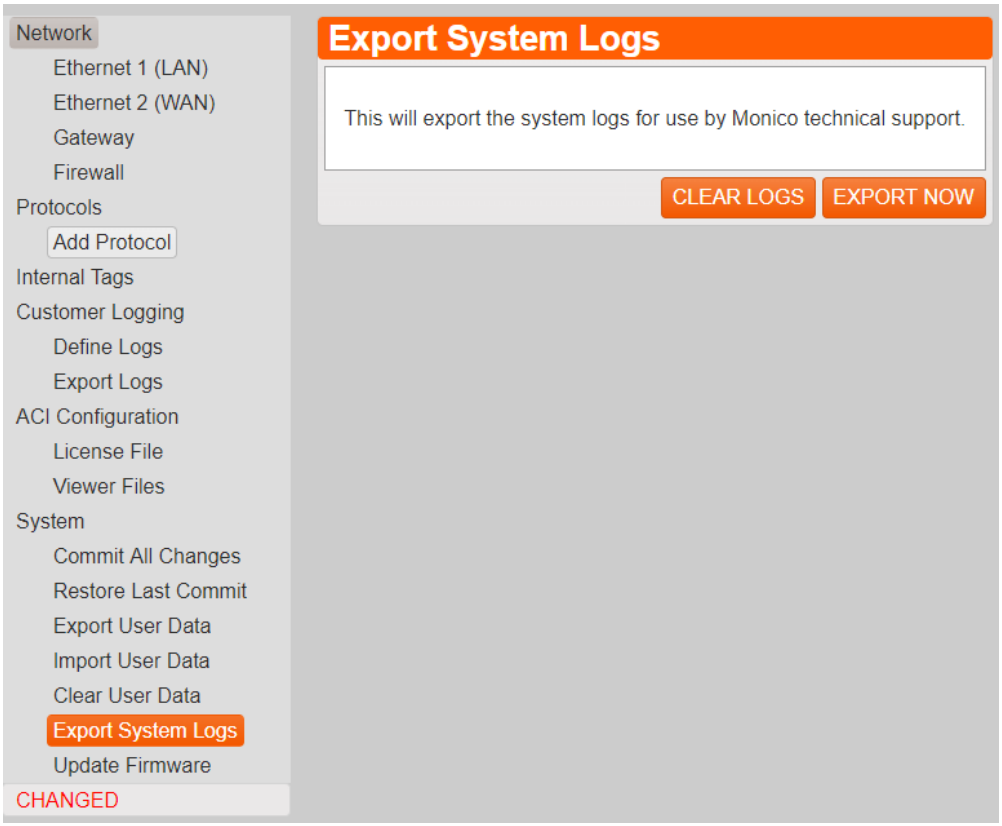
3. Click Clear Data in the center menu. A pop-up box will appear asking if you want to continue with clearing mCore. Click Ok to continue.

4. After you've clicked OK, it will take a few seconds and then all protocols will be gone, and all settings reset to factory defaults. The first thing to do before continuing with making a new configuration is to change the IP credentials back to useable settings as they have now been reverted to factory defaults.

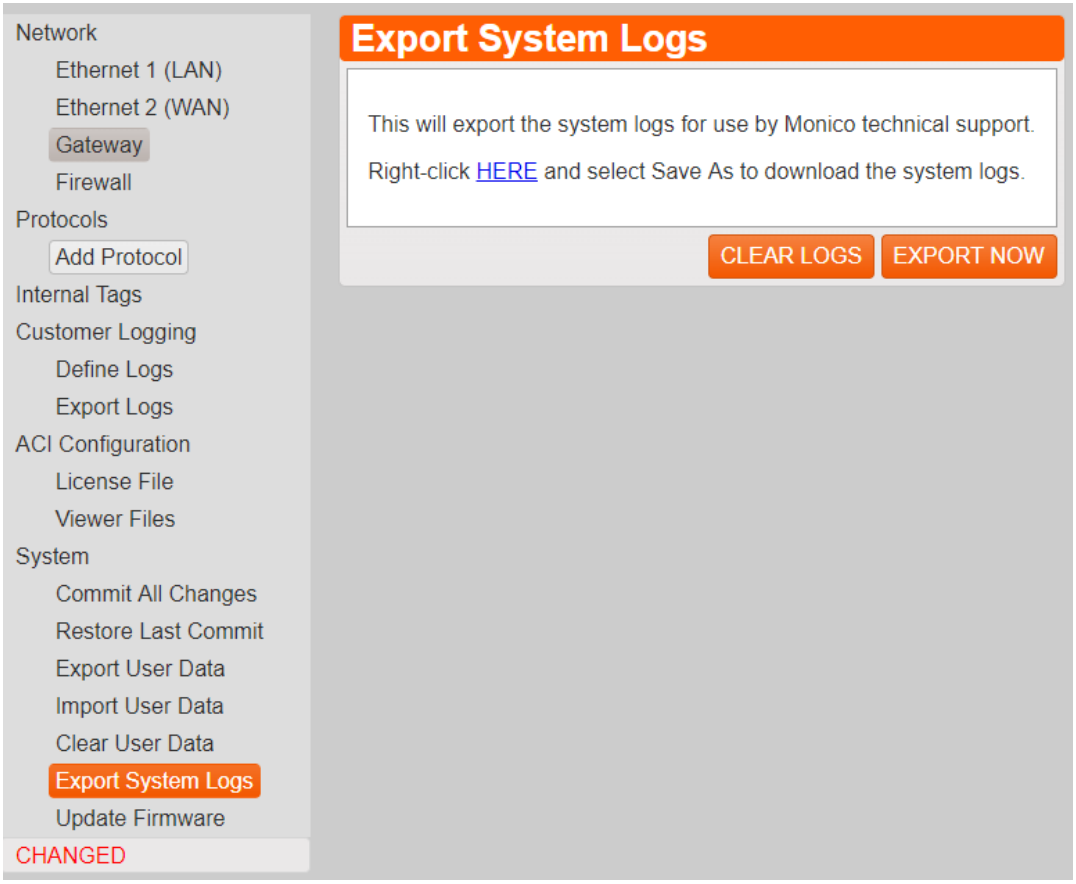
SECTION 13.6: EXPORT SYSTEM LOGS

System logs are what we call Traffic Logs of RAW data at the driver level. The current protocols that support this feature are SAE J1939 and CDL. The logs are enabled by selecting the Traffic Logs tab under the protocol and enabling the logs by clicking the check box. See the respective protocol section in this manual for how to steps for enabling Traffic Logs. Allow mCore to run for at least 5 minutes after enabling Traffic Logging and mCore rebooting before attempting to Export System Logs.

1. Click the Export System Logs button in the left menu toward the bottom. The following screen will appear.



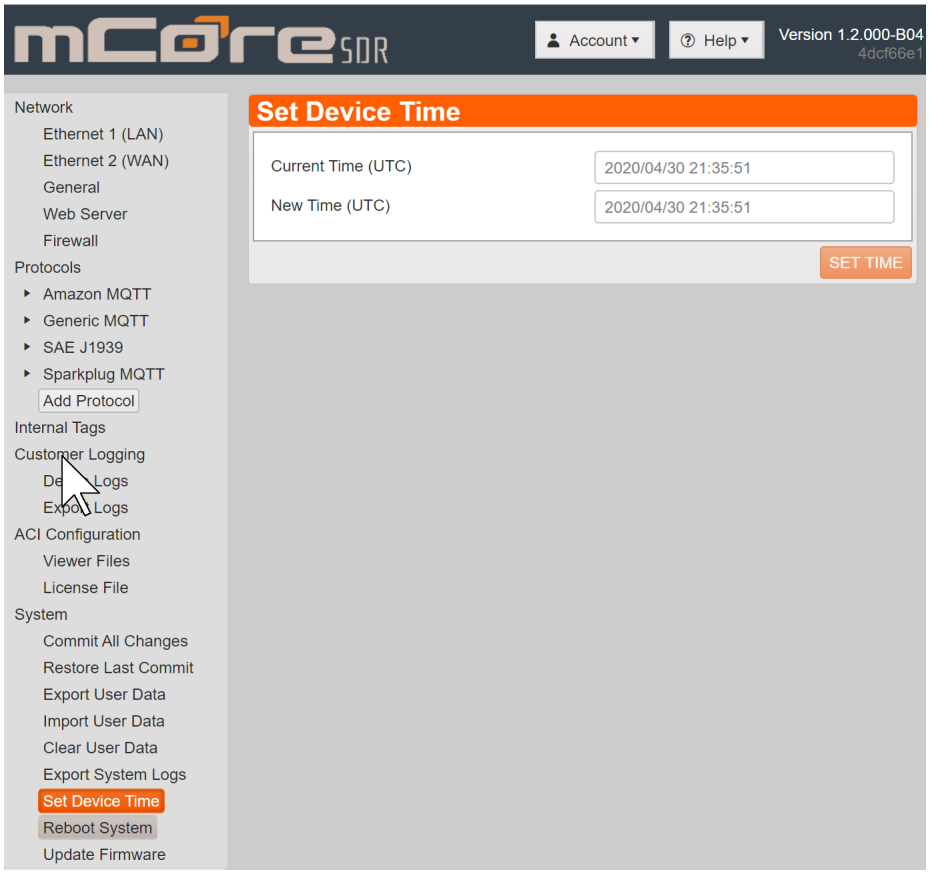
2. Click the Export Now button in the middle menu. The following screen will appear. Click the blue hyperlink labeled “HERE”. A pop-up box will appear asking where you want to save the log file and what to name it. Click Save.
    a. If multiple sets of logs are needed over an extended time frame, you will need to delete the old logs before each run to obtain traffic logs. Make sure to have already backed up the current log files before doing this step. To do this, click the Clear Logs button. Click Ok to continue.



3. Once you’ve saved the log to your computer, send them to support@monicoinc.com. These log files are encrypted as they contain proprietary information. We will decrypt, analyze the logs, and reply. Please contact us at 281-350-8751 ext.1 if you need more rapid assistance.

SECTION 13.7: SET DEVICE TIME

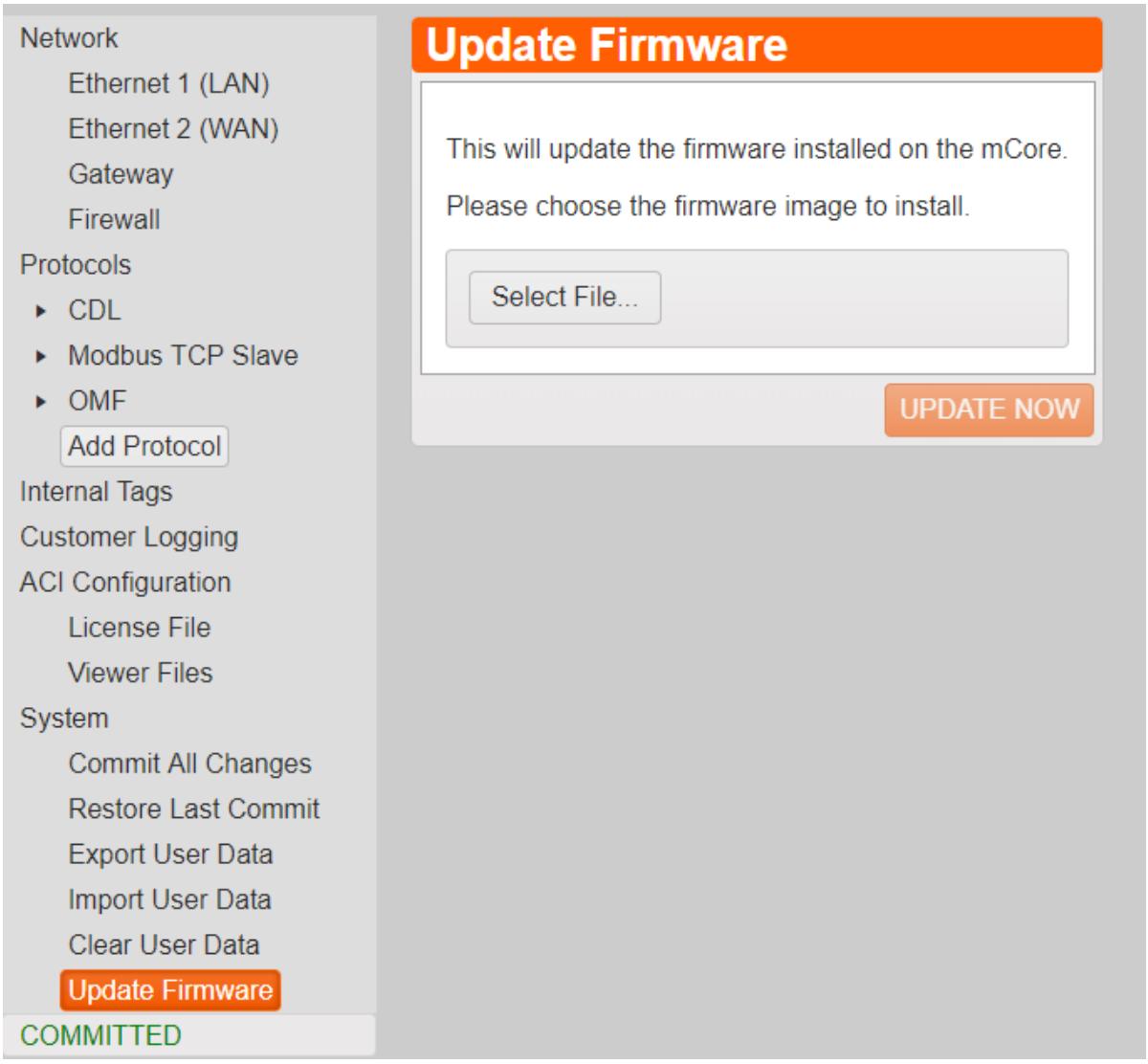
This feature offers the ability to set the real-time clock (RTC) manually if an NTP server is not available to automatically and regularly updating the clock. The RTC will naturally drift over time, so manually setting it is not a long-term solution if the accurate time stamps are important. Simply enter the time in UTC.



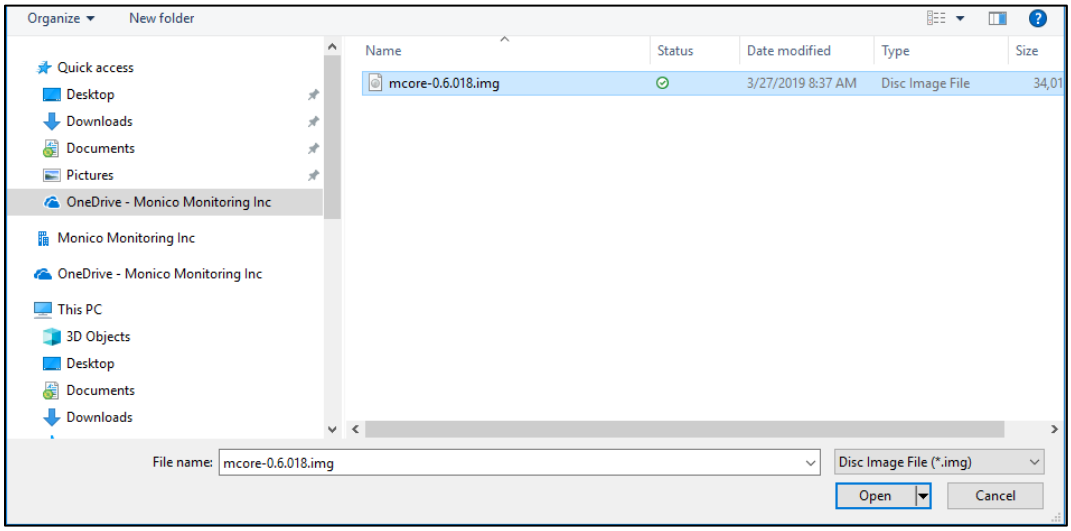
Click to Return to Top of Document

SECTION 13.8: UPDATE FIRMWARE

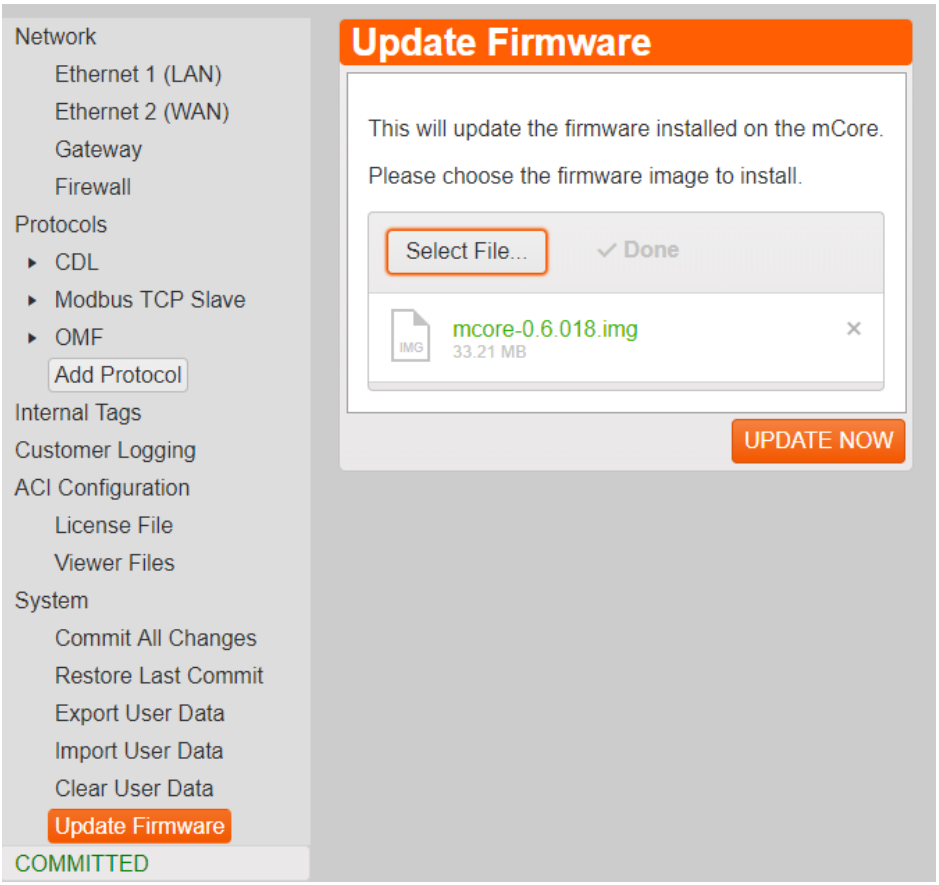
1. From the System menu select Update Firmware.



2. Select the file that want you to upload. All firmware updates will have the suffix .img in the filename i.e. mcore-0.6.018.img. After selecting the file, click Open to begin the upload process.



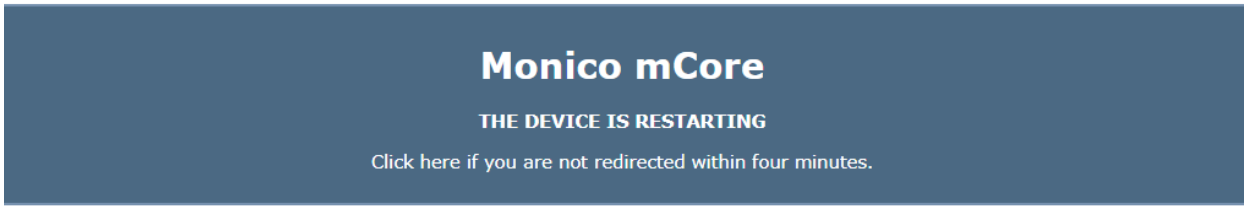
3. Once the file has completed the upload 100%, click UPDATE NOW to begin the upgrade process.



4. You will then be prompted with a warning that you are about to upgrade your firmware. Click OK to continue.

   DO NOT REMOVE POWER WHILE THE mCORE IS UPDATING FIRMWARE.

5. While the firmware upgrade is in progress, you will see the following message. It will take approximately four (4) minutes for the upgrade to complete, at which time the mCore will automatically reboot and the screen with refresh back to the initial landing page.

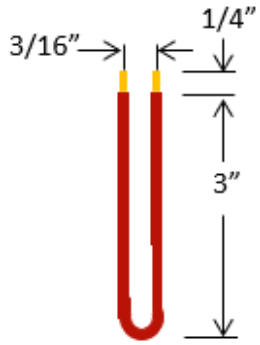


SECTION 14: FACTORY RESET

mCore can be reset to a Factory state should the need arise. A Factory Reset will put the IP address for Ethernet 1 and 2 to DHCP. The Username and Password will be reset to the defaults as listed on the label sticker on the bottom of the mCore. All protocols and other settings will not appear after initial reboot. To get the previous configuration back, click the Restore Last Commit button. Contact Monico Support should you have questions or need assistance with performing this reset.

1. Power off the mCore.

2. You will need a 6" piece of 18-gauge wire with both ends stripped back by an 1/4 inch and bent into the shape of a "U". See picture below.



3. The Power Connector - Key Connector A will have 2 seal plugs already installed from the factory in the Factory Reset pins 4 and 5. Remove those plugs by gently pulling them out. Save these plugs to be re-installed after the Reset is performed.

Push the wire through the waterproof gasket at Pins 4 and 5 where the seal plugs were removed on the Power Connector - Key Connector A until you bottom out. The pin out is shown in Section 3.2.

Click to Return to Top of Document

4. While applying gentle force on the jumper wire to make sure the wire maintains contact with both Pins 4 and 5, power on the mCore. The Power LED will briefly go Solid Red then go Solid Amber. All LEDs below the Power LED will be flashing Red then go flashing Green. Once you see flashing Green, the unit has been reset.

5. Remove the jumper wire and the mCore will reboot automatically.

6. Seal the pierced gasket holes for Pins 4 and 5 with the sealing plugs that were removed previously. Gently push the plugs into the pin holes. Make sure the thicker tapered portion is inserted face down toward the pins and the thinner portion is sticking out of the connector. Once re-installed, the plugs should look like the picture in Step 3.

7. The username and password have also been reset back to the default as shown on the label sticker on the bottom of the mCore. To connect, open an internet browser and type in the serial number of the mCore followed by ".local". For example, if the mCore serial number is mc-1234-5678, you would enter, without quotes, "http://mc-1234-5678.local" or "https://mc-1234-5678.local".