ECM4000

Ethernet Control Module

Hardware and Software Manual

Software Version 1.03

Scanivalve

PREFACE

WARNINGS, CAUTIONS, AND NOTES

The WARNING! symbol indicates that danger of injury to persons and the environment and/or considerable damage (mortal danger, danger of injury) could occur if the respective safety precautions are not taken.

The CAUTION! symbol indicates danger to the system and material if the respective safety precautions are not taken.

The ESD note symbol indicates that proper precautions for handling Electrostatic Sensitive Devices need to be taken when performing the related operation. This includes the use of grounded work surfaces and personal wrist straps to prevent damage to sensitive electronic components.

WARRANTY

Scanivalve Corporation, Liberty Lake, Washington, hereafter referred to as Seller, warrants to the Buyer and the first end user that its products will be free from defects in workmanship and material for a period of twelve (12) months from date of delivery. Written notice of any claimed defect must be received by Seller within thirty (30) days after such defect is first discovered. The claimed defective product must be returned by prepaid transportation to Seller within ninety (90) days after the defect is first discovered. Seller's obligations under this Warranty are limited to repairing or replacing, at its option, any product or component part thereof that is proven to be other than as herein warranted.

Surface transportation charges covering any repaired or replacement product or component part shall be at Seller's expense; however, inspection, testing and return transportation charges covering any product or component part returned and redelivered, which proves not to be defective, shall be at the expense of Buyer or the end user, whomever has returned such product or component part.

This Warranty does not extend to any Seller product or component part thereof which has been subjected to misuse, accident or improper installation, maintenance or application; or to any product or component part thereof which has been repaired or altered outside of Seller's facilities unless authorized in writing by Seller, or unless such installation, repair or alteration is performed by Seller; or to any labor charges whatsoever, whether for removal and/or reinstallation of the defective product or component part or otherwise, except for Seller's labor charges for repair or replacement in accordance with the Warranty. Any repaired or replacement product or component part thereof provided by Seller under this Warranty shall, upon redelivery to Buyer, be warranted for the unexpired portion of the original product warranty.

THIS WARRANTY IS IN LIEU OF AND EXCLUDES ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, ARISING

BY OPERATION OF LAW OR OTHERWISE, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND IN NO EVENT SHALL SELLER BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES.

In the event of a failure:

     1) Notify Scanivalve Corporation, Customer Service Department. Include model number and serial number. On receipt of this information, service data or shipping instructions will be forwarded. This may be transacted by telephone or e-mail.

     2) On receipt of shipping instructions, forward the product, transportation prepaid. Repairs will be made and the product returned.

     3) All shipments should be made via "Best Way." The product should be shipped in the original packing container or wrapped in protective material and surrounded by a minimum of four (4) inches of a shock absorbing material.

TRADEMARKS ® AND COPYRIGHTS ©

Scanivalve is a registered trademark of Scanivalve Corporation.

LabVIEW® is a registered trademark of National Instruments.

Swagelok® is a registered trademark of Swagelok Company.

All other brand and product names are trademarks or registered trademarks of their respective companies.

PACKAGING FOR SHIPMENT

If the product must be shipped, whether being returned to Scanivalve or relocated to another location, it must be packaged properly to minimize the risk of damage. The recommended method of packing is to place the instrument in a container, surrounded on all sides with at lease four inches of shock attenuating material such as Styrofoam peanuts.

IMPORTANT NOTICE

Please note that the product specifications and other information contained in this manual are subject to change without notice. Scanivalve Corporation makes an effort and strives to provide complete and current information for the proper use of the equipment. If there are any questions regarding this manual or the proper use of the equipment, contact Scanivalve Corporation.

CONTACT INFORMATION

If there are any questions or concerns regarding any Scanivalve product, please do not hesitate to contact us at the following:

Scanivalve Corp.
1722 N. Madson Street
Liberty Lake, WA 99019
Telephone: (800)935-5151   (509)891-9970
Fax: (509)891-9481
scanco@scanivalve.com
www.scanivalve.com

Scanivalve Corporation is an ISO 9001:2015 certified company.

[INTENTIONALLY LEFT BLANK]

TABLE OF CONTENTS

Scanivalve

Scanivalve

[INTENTIONALLY LEFT BLANK]

SECTION 1: SPECIFICATIONS

GENERAL SPECIFICATIONS

Size (WxHxD)
ECM4000   6.63" x 2.15" x 8.03"
                                     [168.3mm x 54.6mm x 204.1mm]

ECM4000RM   19" x 3.5" x 12.8"
                                     [483mm x 88mm x 326mm]

Weight
        ECM4000   2.31lbs [1.05kg]
        ECM4000 with Heater 2.43lbs [1.10kg]
        ECM4000RM   6.5lbs [2.95kg]

Supply Air Input Fitting   1/4" Swagelok
                             Compression Fitting

Max Supply Air Input   120psi

Solenoid Vent Fitting   1/4" Swagelok
                             Compression Fitting

Pneumatic Output Fittings   Push-in style for
                             3/16" OD Tube
                             or 0.063" bulged
                             tubulations

Electrical Connections
        Serial   Amphenol JTP02RE8-6P
        Serial - Mate   Amphenol JT06RE8-6S
        Power   Amphenol PT02A-8-3P
        Power - Mate   Amphenol PT02A-8-3S
        Digital Out   M12, 12 pin, A-Code, Socket
        Digital Out - Mate   M12, 12 pin, A-Code, Pin
        Digital Ins   M12, 12 pin, A-Code, Pin
        Digital Ins - Mate   M12, 12 pin, A-Code, Socket

Ethernet Connections
        ECM4000   M12, 4 pin, D-Code, Female
        ECM4000RM   RJ45, Female

Communication Protocol
        Ethernet   TCP/IP, UDP
        Serial   RS-232 (limited function)
        Synchronization   IEEE1588v2 PTP

Ethernet Connection   100baseT, MDIX auto-crossing

Power Requirements

ECM4000   9-36Vdc, 6.5W
ECM4000 with Heater   20-36Vdc, 65W
ECM4000RM   9-36Vdc, 6.5W

Digital Out Voltages   +5Vdc
                                 AND/OR
                            Supplied Power to ECM
                               (+9-36Vdc)

ENVIRONMENT SPECIFICATIONS

Temperature
        Operation   0 to 70 °C
                            -40°C to 70°C with Heater
        Storage   0 to 80°C

Humidity
        Operation   5 to 95% RH, Non-Condensing
        Storage   5 to 95% RH, Non-Condensing

Maximum
Environmental Pressure   100psia (690kPa absolute)

Minimum
Environmental Pressure   0.50psia (3.45kPa absolute)

Scanivalve

FIGURE 1-1: ECM4000 DIMENSIONAL DRAWING

NOTES:
1. MATING CONNECTOR OR CABLE WITH FLYING LEADS PROVIDED.
2. POWER INPUT REQUIREMENTS:
   1. 9-36VDC
   2. 6.5W MAX
   3. 65W MAX W/ HEATER
3. WEIGHT:
   1. 2.31 LBS [1.05KG] STANDARD
   2. 2.43 LBS [1.10KG] WITH HEATER

-1 ECM4000 ETHERNET CONTROL MODULE
-2 ECM4000 ETHERNET CONTROL MODULE (WITH HEATER)

POWER, LINK, LAN INDICATOR LEDS

DOUT STATE INDICATOR LEDS

SOLENOID OUTLET PORTS (FOR Ø0.125 O.D. [3.2] TUBING)

POWER BUTTON

DIGITAL OUTPUT CONNECTOR
M12 12 CONTACT, A-CODE, SOCKET
M12 12 CONTACT, A-CODE, PIN (MATE)

DIGITAL INPUT CONNECTOR
M12 12 CONTACT, A-CODE, PIN
M12 12 CONTACT, A-CODE, SOCKET (MATE)

[168.3]
6.63

[166.4]
6.55

[204.1]
8.03

[191.1]
7.53

SLOT: Ø.25 X .44
[Ø6.4 X 11.1]

[58]
2.29

[54.6]
2.15

SERIAL CONNECTOR
PT02A-8-5P
PT06A-8-5S (MATE)

ETHERNET CONNECTOR
M12 RECEPTACLE, 4 CONTACT, D-CODE
M12 MALE, 4 CONTACT, D-CODE (MATE)

POWER CONNECTOR
PT02A-8-3P
PT06A-8-3S (MATE)

SOLENOID VENT FITTING
(1/4" SWAGELOK COMPRESSION)

SUPPLY AIR INPUT FITTING
(1/4" SWAGELOK COMPRESSION)

REVISIONS				
REV	DESCRIPTION	DATE	APPROVED	
-	INITIAL RELEASE	4-20-20	AA	
A	SERIAL CONNECTOR	11-17-20	RP	

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
TWO PLACE DECIMAL ± .015
THREE PLACE DECIMAL ± .005

SHEETMETAL PARTS ± .010
BEND RADIUS .06 MAX
ANGULARITY ± 1.5°

VALUED CUSTOMER:

CUSTOMER APPROVAL:

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	RAP	7-10-19
CHECKED	AA	4-20-20
ENGINEER	RP	

CAD: SOLIDWORKS 2018

Scanivalve
LIBERTY LAKE, WA

TITLE: ECM4000 - ETHERNET CONTROL MODULE OUTLINE DRAWING

SIZE: B　　DWG. NO. 80143　　REV A

SCALE: N/A　　ERO#:　　SHEET 1 OF 1

FIGURE 1-2: *ECM4000-RM DIMENSIONAL DRAWING*

NOTES:
1. ALL DIMENSIONS IN INCHES [MM].
2. REFERENCE THE ECM4000 USER MANUAL.
3. MATING CONNECTOR OR CABLE WITH FLYING LEADS PROVIDED.
4. POWER REQUIREMENTS: 9-36VDC, 6.5W.
5. WEIGHT: 6.5 LBS [2.95KG].

SOLENOID SUPPLY INPUT FITTING
[120PSI MAX]
1/4" SWAGELOK COMPRESSION

SOLENOID VENT FITTING
1/4" SWAGELOK COMPRESSION

DIGITAL INPUT CONNECTOR
M12 12 CONTACT, A-CODE, SOCKET
M12 12 CONTACT, A-CODE, PIN (MATE)

4X SLOT .250 X .437
[6.4 X 11.1]

SOLENOID OUTLET PORTS
(FOR 3/16" [4.8] O.D. TUBING)

DIGITAL OUTPUT CONNECTOR
M12 12 CONTACT, A-CODE, PIN
M12 12 CONTACT, A-CODE, SOCKET (MATE)

SERIAL CONNECTOR
PT02A-8-5P
PT06A-8-5S (MATE)

POWER CONNECTOR
PT02A-8-3P
PT06A-8-3S (MATE)

ETHERNET CONNECTION
RJ45

DOUT STATE INDICATOR LEDS

POWER, LINK, LAN INDICATORS LEDS

POWER BUTTON

SOLENOID STATE LEDS

ECM4000RM - ETHERNET CONTROL MODULE

[76.2]
3.0

[483]
19.0

[465.8]
18.3

[88]
3.5

[304]
12.0

[326]
12.8

	REVISIONS					80174-1
REV	DESCRIPTION			DATE	APPROVED	
-	INITIAL RELEASE			7-8-21	AA	

ECM4000 RACKMOUNT OUTLINE

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
TWO PLACE DECIMAL ± .015
THREE PLACE DECIMAL ± .005
ANGULARITY ± 1.5°
BEND RADIUS .06 MAX
SHEETMETAL PARTS ± .025

	NAME	DATE
DRAWN	AA	7-8-21
CHECKED	MG	7-8-21
ENGINEER	AA	7-8-21

CUSTOMER APPROVAL:

VALUED CUSTOMER:

DATE:

CAD: SOLIDWORKS 2019

Scanivalve
LIBERTY LAKE, WA

TITLE: ECM4000 RACKMOUNT OUTLINE

SIZE: B

DWG. NO. 80174

REV -

SCALE: N/A   ERO#:   SHEET 1 OF 1

DO NOT SCALE DRAWING

SECTION 2: INTRODUCTION

OVERVIEW

The ECM4000 (Ethernet Control Module) is a single module system orchestrator designed to be used alongside other measurement devices and instrumentation. It provides a single point of communication for multiple pressure and temperature scanners as well as a simple solution for managing control pressures to pressure scanners, triggering external devices using digital outputs, and running sequences using digital input signals.

The ECM4000's hardware provides 8 discrete digital inputs, 8 discrete digital outputs, and 3 discrete pneumatic outputs which are controlled by internal solenoids. Commands and configurations unique to the ECM4000's software allow each of these inputs and outputs to be controlled manually, as part of a sequence, or from within a script.

The ECM4000 is IEEE1588v2 Precision Time Protocol compliant. The Precision Time Protocol features allow it to be configured as a PTP salve when a PTP Grandmaster is present on a network. Alternatively, the ECM may be configured to act as a PTP master in a Scanivalve system where high accuracy data synchronization is required. Four of the digital outputs can be configured as fixed frequency generators, capable of providing a 50% duty cycle waveform at rates up to 20kHz. The waveform outputs can be synchronized with PTP time.

The ECM4000 also features the ability to create and execute scripts which can include responding to digital input signals, cycling digital outputs, cycling control pressures, sending Ethernet commands, querying pressure scanners, or inserting timed delays. Multiple scripts can be created allowing inputs from software commands or digital input switches to initiate tasks such as:

- Perform a CALZ (zero offset calibration) on all pressure scanners within a system.
- Perform a purge sequence to clear debris from measurement lines.
- Initiate data collection from multiple scanners, synchronized by the ECM4000's IEEE1588 synchronization.

ECM4000 - MPS4000 Typical System Layout
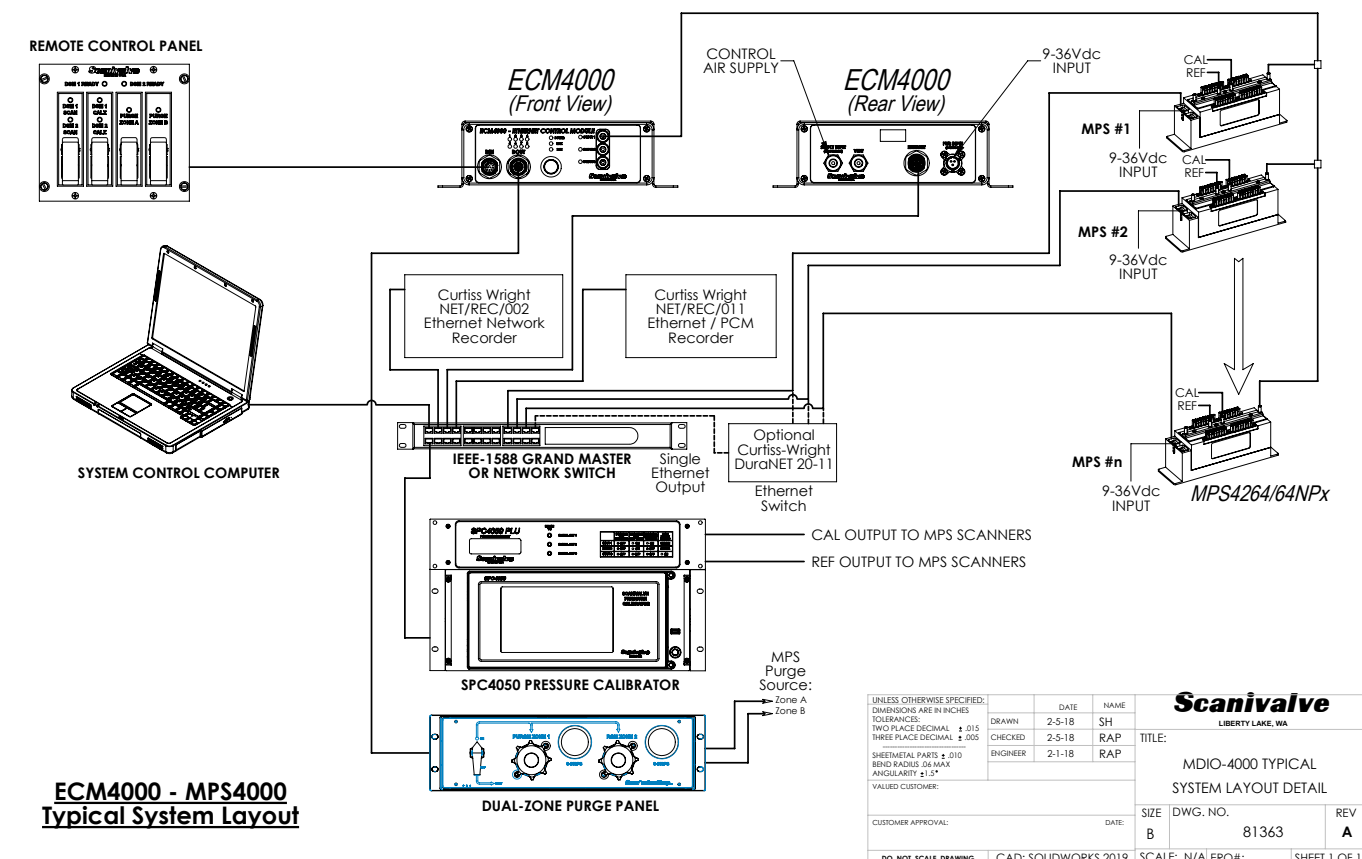
ECM FRONT PANEL

The ECM's front panel houses several different interfaces and indicators.

Status LEDs

Power, Ethernet, and Activity Indicator LEDs

Pneumatic Output Indicator LEDs

Digital Input (DIN) Connector

Digital Output (DOUT) Connector

Power On/Off Switch

Pneumatic Outputs (POUT)
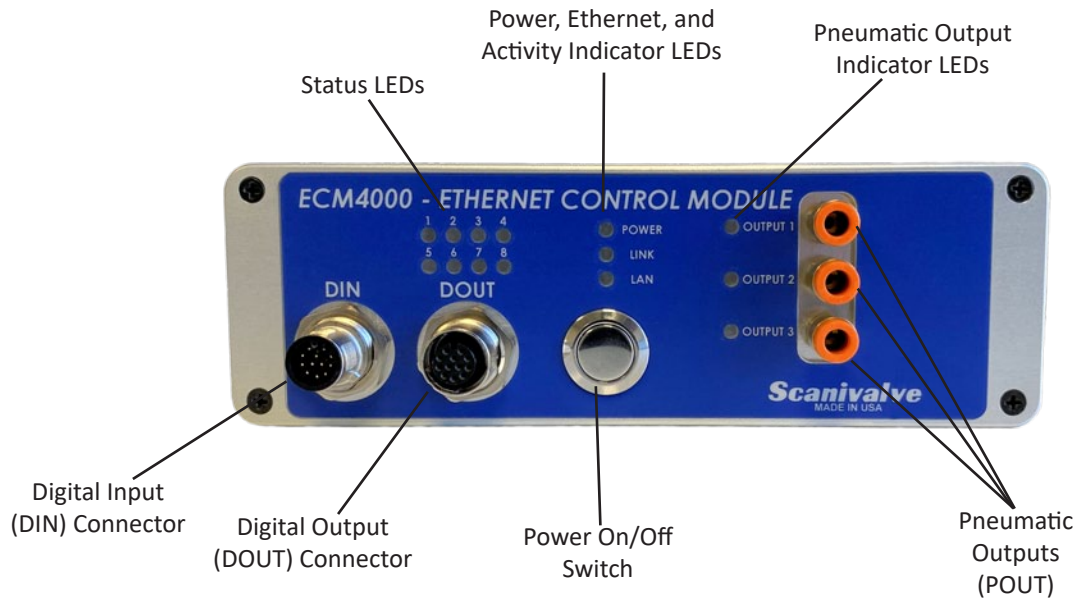
ECM REAR PANEL

The ECM's rear panel houses several different connectors and identifiers.

Model and Serial Number Tag

Pneumatic Supply Input

Pneumatic "Vent" Port

Serial Connector

Ethernet Connector

ECM Power Input
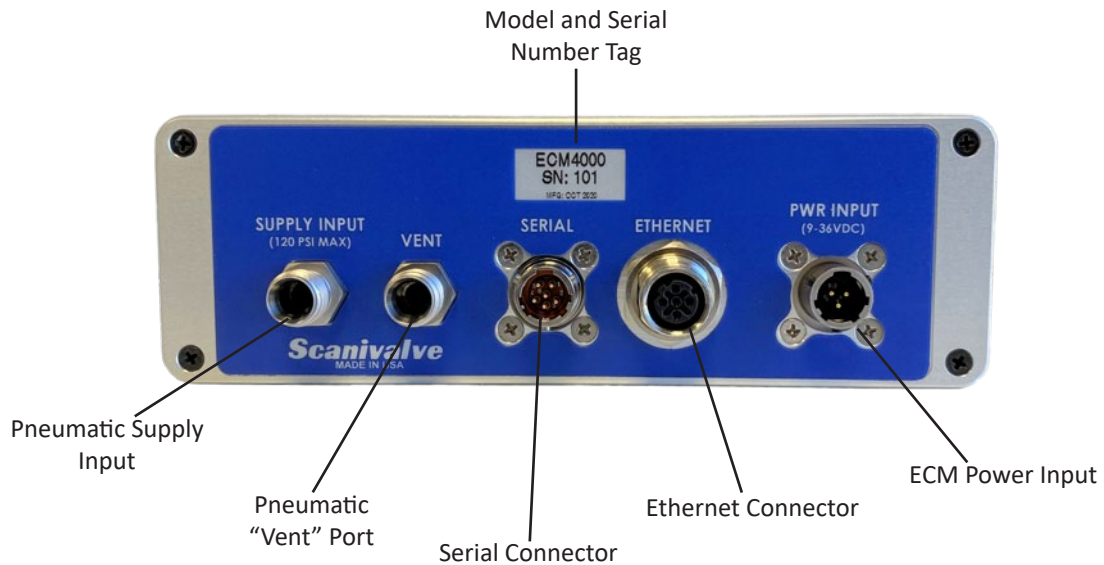
[INTENTIONALLY LEFT BLANK]

SECTION 3: OPERATION

UNPACK & INVENTORY

When first unpacking the ECM4000 module, begin by inspecting and inventorying the contents of the package. If any visible damage is immediately noticed, or if any contents are missing, please contact Scanivalve before proceeding. As a minimum, ECM modules are shipped with the following contents:

1. ECM4000 module
2. M12, 12 pin, female flying leads cable, 1.5m
3. M12, 12 pin, male flying leads cable, 1.5m
4. M12, 4 Pin, RJ45 Ethernet cable, 1m*
5. CO-613 3 Pin, female power mating connector
6. 20325-01 ECM Mounting Strap*
7. CO-767 6 Pin, Serial mating connector
8. ECM4000 Resource Disk (USB)

Item not included with the rack mount variation of the ECM.

ENVIRONMENTAL CONSIDERATIONS

ECM4000 modules are constructed with a rugged, aluminum case. It is designed to withstand normal industrial, flight test, educational, wind tunnel, or similar applications. The module is not waterproof and it should be protected from any splash, spray, or mist. If any moisture gets spilled or splashed on the ECM module, wipe it dry immediately to prevent damage to the module.

The ECM module should not be mounted in a location where it may be subjected to extreme temperature shifts or ambient temperatures outside limits defined in "Section 1: Specifications" on page 6. If necessary, the ECM can be configured with an internal heater and controller that will allow it to be operated in temperatures below 0°C. The heater's primary purpose is to prevent the electronics and solenoids from freezing.

CAUTION! Mounting the ECM module inadequately or in an environment that does not conform to the recommendations can results in permanent damage to the module.

MOUNTING

A stainless steel mounting strap will is provided with the ECM4000 to allow the module to be mounted in a permanent location. The mounting straps tabs are approximately

0.050" (1.27mm) thick. The mounting holes are centered on either side of the strap and provide a 0.25" by 0.44" through slot (6.4mm x 11.1mm).
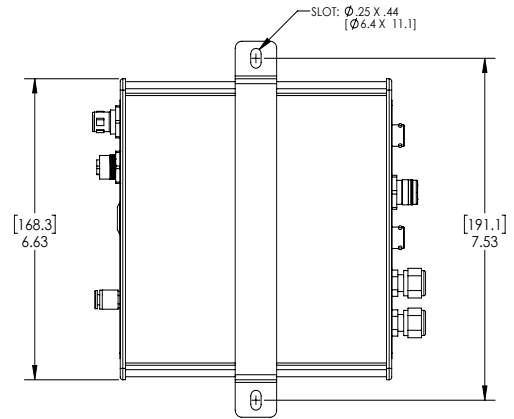
FIGURE 3-1: MOUNTING BRACKET DIMENSIONS

POWER-UP

Due to the design of the ECM4000, power-up time is minimal. After power has been connected to the module, the power button on the front panel can be depressed to start the power up sequence. Do not apply the power connection to the ECM with the power button in the "ON" or depressed state.

When powered on, the ECM will enter READY mode. This is true with the default Scanivalve configuration unless otherwise specified. The ECM may be configured to perform one of the following alternative options on power-up:
*   Load a script file, then enter READY mode
*   Load a script file, then RUN a script

The behavior of the ECM4000 on power-up can be configured by the variable **SET AUTORUN** in **LIST CONFIG**. Please refer to Section 5: Software for more information on power-up configurations.

COMMUNICATIONS

The ECM4000 module is designed primarily for Ethernet communications. Ethernet communications provide a means to configure the ECM module as well as run scripts, sequences, or control digital and pneumatic outputs.

A serial (RS-232) connection is also supported. The serial connection is designed to provide emergency communications with the module for cases in which an Ethernet connection cannot be established. If the internal memory of the ECM becomes corrupted, the serial connection

Scanivalve

allows the user to format the memory and restore the operating files.

SERIAL COMMUNICATIONS

Every ECM4000 module has an serial connector. It is available through the 'Serial' connector on the back panel of the module. The serial connection can be used to find and set the module's "Group IP" settings, including the IP address. The serial connection can also be used for general debugging and troubleshooting purposes.

The module's internal memory can be formatted and restored through a serial connection using the "Firmware Installation via Serial" procedure in Section 6: Maintenance.

Settings for establishing a serial connection to the ECM module are as follows:

    Bits per second: 9600
    Data bits: 8
    Parity: none
    Stop bits: 1
    Flow control: none

A serial and Ethernet connection can be established simultaneously. All commands accepted in Section 5 are accepted via serial communications.

Information on the connector type, wiring diagrams, and other information on the serial connector can be found in Section 4: Hardware.

ETHERNET COMMUNICATIONS

The ECM4000 has one Ethernet 100Base-T with MDIX auto-crossing connection. Ethernet is the primary means of communication with the ECM. All ECM modules use a M12, 4 pin, D-Code socket series connector for the Ethernet connection. This is a latching or threaded connector to prevent the connector from becoming unseated when subjected to vibrations. If creating a custom manufactured cable, a Shielded Category 5e cable or better is recommended. More information on the connector and wiring can be found in Section 4: Hardware.

SETTING UP AN ETHERNET CONNECTION

Before an Ethernet connection can be established, the IP address must be configured. This can be done manually with a "static" IP address, or automatically using a "dynamic" IP address, which is automatically assigned to the ECM by a DHCP server. If a "static" IP address is desired, the IP address of the ECM must be set manually. The IP address assigned to the ECM must be compatible with the network/host computer. If a "dynamic" IP address

is desired, the ECM must be configured to obtain an IP address automatically, and it must be on a network with a DHCP server present.

USING A STATIC IPV4 ADDRESS

If a static IP address is desired, the IP address must be manually configured. The IP address that is configured in the ECM must be compatible with the network the ECM is being connected to. If simply connecting the ECM directly to a host computer, the IP address of the ECM and the host computer must be compatible, which may require manual configuration of the ECM and/or computer.

The range of compatible IP addresses is defined by the subnet mask. The standard default subnet mask is 255.255.0.0. This default subnet mask requires that the IP address of the module and host computer must share the first two octets, or sets of numbers. The third and fourth octets of the IP address is variable with this subnet, although it is typically recommended that the third octet also be shared between the host computer and the module. The subnet mask digits of "255" define that the two IP addresses must have matching digits in those positions, and the subnet mask digit of "0" allows the two IP addresses to have unique values for those octets and still be compatible. No two devices on a single network can share the same IP address. Below are some examples of compatible and non-compatible IP addresses:

Example of matching the first three octets:
        Subnet mask: 255.255.255.0
        Host computer: 191.30.66.90
        ECM module: 191.30.66.100
Example of matching the first two octets:
        Subnet mask: 255.255.0.0
        Host computer: 191.30.1.1
        ECM module: 191.30.66.125
Example of NON-COMPATIBLE IP addresses:
        Subnet mask: 255.255.255.0
        Host computer: 191.30.1.1
        ECM module: 191.30.66.5

When configuring the network, the IP address of the ECM module can be modified to match the network, or the IP address of the host computer can be modified to be compatible with the ECM module.

The IP address of a Windows 7, 8, or 10 host computer can be changed under:
    Control Panel -> Network and Sharing Center -> Local
      Area Connection/Ethernet -> Properties -> Internet
      Protocol Version 4 (TCP/IPv4) -> Properties

Changing the IP address of the host computer may require administrative rights.

The IP address of the computer can quickly be found using Windows Command Prompt and the command "ipconfig."

ECM4000 modules are shipped with a pre-configured IP address that uses the following format:

> 191.30.66.xxx

The first two octets used are unique to all Scanivalve products (191.30). The third octet is the device family code (66). The fourth octet is the last three digits of the ECM's serial number. *"xxx" is only a placeholder in the example as all modules will have different serial numbers.*

Based on the examples above, the host computer can be configured to communicate with the shipped IP address of the ECM.

In situations where the IP address of the host computer cannot be altered, the ECM's address must be changed to establish an Ethernet connection. The ECM's IP address is set with the **IPADD** variable in the IP Group of variables (**LIST IP**). This can be changed through an existing Ethernet connection or via serial connection.

If an existing Ethernet connection can be established (by changing the host computer's IP address, using an alternate computer, or by some other means), it is the preferred method for changing the IP address of the ECM. If Ethernet communication is not possible, a serial connection must be established with the ECM following the guidelines in the "Serial Communications" section.

If the IPADD variable is changed either through an Ethernet connection or a serial connection, the new address will not be effective until a **SAVE IP** command has been issued and the power is cycled. For more information on changing the **IPADD** variable, see the "Changing the IP address" procedure.

CLIENT/HOST OPTIONS

Once the module has been assigned a proper IP address and is connected to power and a host computer or network, Ethernet communications can be established with the ECM module. Communications can be made through several software packages including:

- PC - Web Server
- PC - TCP/IP
- PC - ScanTel (Scanivalve PN: 155406-01)
- PC - LabVIEW
- PC - Windows HyperTerminal (alt. PuTTY)

PC - WEB SERVER

The ECM4000 is integrated with a built-in Web Server with an easy to use graphical interface. The Web Server can be accessed using a number of web browsers, including Google Chrome and Microsoft Edge. All functionality available in the ECM can be accessed using the Web Server including additional features such as a dashboard for writing and editing scripts, terminal window for simple command-line communication, and the ability to update the ECM's firmware. This is the recommended means of communications for most users.

An internet connection is not required when accessing the ECM's Web Server. Not all web browsers are supported. Please see the section dedicated to the Web Server for more information.

PC - TCP/IP

The user may write their own TCP/IP interface using the software specification portion of this manual. This interface could allow the user to:

- Issue commands to any or all ECM modules on the network.
- Display returned information from the ECM module(s).
- Write returned information to the client/host in TCP/IP format.

PC - SCANTEL

ScanTel is a free communications utility designed by Scanivalve to communicate with Scanivalve products including the ECM4000. It is a text based, command line program that allows users to connect to a single ECM module. ScanTel can be downloaded from the Scanivalve website.

PC - LABVIEW®

The ECM4000 can be integrated into a LabVIEW® based program, or a custom LabVIEW® program can be developed to operate the ECM4000. Users can easily create a custom graphical interface to configure the unit or execute scripts.

PC - HYPERTERMINAL

HyperTerminal is a Windows program included as part of Windows 2000, XP, and Vista Operating Systems. It is a text based command line program. This program permits a user to connect to a single ECM module, modify the configuration variables, format the internal memory, or troubleshooting. HyperTerminal provides a means for both Serial RS-232 and Ethernet connections. Terminal emulators like PuTTY can be downloaded for free from various sources online if HyperTerminal is not available.

CHANGING THE IP ADDRESS

The following procedure will change the IP address of the ECM4000.

1. Establish a connection to the ECM4000 using a serial or Ethernet connection. Software that can be used is discussed in the previous section.
2. Issue the command:
   LIST IP<ENTER>
3. The IP address and communication variables will be displayed.
4. Issue the command:
   SET IPADD xxx.xxx.xxx.xxx<ENTER>
   *where **xxx.xxx.xxx.xxx** is the IP address desired.*
5. Issue the command:
   LIST IP<ENTER>
   Confirm that the IP address set is correct.
6. Issue the command:
   SAVE IP<ENTER>
   The IP address has now been saved.
7. Issue the command:
   REBOOT<ENTER>
   - or - Cycle the ECM power.
   The ECM will now boot with the new IP address.

PNEUMATIC INTERFACE

The ECM4000 has 3 independently controlled pneumatic outputs that can be used for a wide range of applications. One of the primary uses of the pneumatic outs is to provide orchestrated control pressures to Scanivalve pressure scanners.

Clean, dry instrument grade air must be supplied to the "Supply Input" on the back panel with a maximum input limit of 120psi. Each pneumatic output can be controlled with the use of software commands (**POUT**). These commands can be sent manually, used within a script, or controlled through the Web Server interface.

An LED indicator is provided on the front panel of the ECM to display the current status of the solenoid/output. When the LED is lit green, the solenoid will be open and air will flow through the output. When the light is off, the solenoid is closed and air will no longer be flowing. When the solenoid closes, any back pressure in the line will be vented through the "Vent" port on the back panel of the ECM.

INTERFACING WITH SCANIVALVE SCANNERS

The 3 discrete pneumatic outputs are ideal for providing control pressures to one or more Scanivalve scanners. The following table shows different Scanivalve scanners and the control pressure requirements for each Scanivalve device.

Module	Control Ports	Required Control Pressure (PSI)
MPS4000/NPx	(1) CALCTL	90 - 120
MPS4000/CPx	(2) CALCTL, PxCTL	65 min.
MPS4000/EPx	N/A	N/A
DSA3217-PTP / DSA3218-PTP*	(3) CTL1, CTL2, PRGCTL	90 - 120
ZOC33/22	(2) CALCTL, PxA	55 - 65
ZOC33 Duplex	(3) CALCTL, PxA, PxB	55 - 65
ZOC17	(3) CTL1, CTL2, PRGCTL	90 -120
DSA5000	(1) CTL Supply	90 - 120
DSAENCL/3016	(1) CTL Supply	90 - 120
SPCPLU	(1) Supply Input	Varies (65 or 120)

FIGURE 3-4: SCANIVALVE SCANNERS CONTROL TABLE
* Typical configurations. Also applies for legacy DSA3200s.

DIGITAL INPUTS/OUTPUTS

The ECM4000 has 8 digital outputs and 8 digital inputs that can be configured to perform a number of duties.

DIGITAL OUTPUTS

The ECM's 8 digital outputs can be used to operate LED panels, actuators, solenoids, or a number of other external devices. Each of digital outputs can be controlled independently or all can be changed with a single command (**DOUT**).

Each of the digital outs can supply either +5Vdc or the ECMs input power supply (+9-36Vdc). The max output current is 500mA per channel.

There are two states for all digital outputs: on and off. The first 4 digital outputs have the option of a third state; trigger. Digital outputs 1 - 4 can be configured independently to output a 50% duty cycle waveform at a fixed rate (expressed in Hz). The maximum rate at which a waveform can be generated is 20kHz. The lowest rate is 0.001Hz. The waveform output from the digital outputs is synchronized with the internal clock of the ECM, which can be synchronized using Precision Time Protocol. The trigger out settings can be set and controlled using the **TOUT** and **DOUT** commands. More information on the digital output hardware can be found in Section 4: Hardware.

DIGITAL INPUTS

The ECM's 8 digital inputs can be configured to trigger

scripts or other events. Each of the digital inputs supplies +5Vdc. These digital inputs can also be powered using the ECM's input power supply with the use of additional resistor (see Section 4: Hardware for more information).

The digital inputs are assigned using the **DEVENT** variable in **LIST DEVENT**. Each digital input can be configured to accept a "edge sensing" trigger or "high/low" logic. This allows for either momentary or latching switches to be used in a variety of configurations. The digital inputs can also be configured to follow a sequence of events. There are two sequences available for configuration: first and second. Each sequence has a high and low trigger setting.

When the ECM is first powered on, each digital input will start with "high" logic. When a switch that is connected to a digital input is activated, the ECM logic will go "low." This is referred to as; 0 0 (low logic, first sequence). When the switch is deactivated, the logic will return "high" in the first sequence; 1 0 (high logic, first sequence). If the switch is activated again, it will return to "low," but will be in the second sequence; 0 1 (low logic, second sequence). When the switch is deactivated, the logic will return to "high" on the second sequence; 1 1 (high logic, second sequence). The next time the switch is operated, it will return to 0 0 (low logic, first sequence) and repeat.

Scripts can be configured to be executed any of the four logic/sequence states: 0 0, 1 0, 0 1, or 1 1. When a script is configured to run in the first sequence (either high or low logic), the ECM will automatically assign the script to run in the second sequence in the same logic state. For example, Script-A is configured to be executed on 0 0. The ECM will default to execute the same script on 0 1 (same logic state, second sequence). Any time the digital input goes low, this script will be executed. If a different script (Script-B) is configured to execute on 0 1, it will override the previous scenario. Script-A will only execute when 0 0, and Script-B will only execute when 0 1.

Examples of digital input use can be found in the Appendix A, as well as Section 5: Software.

Status LEDs

The ECM4000 has 8 status LED indicators on the front panel that can be independently controlled. Each of the status LEDs can be turned on or off using a software command (**DISP**). In the ECM's default configuration, the 8 status LEDs follow the digital out statuses. If the digital out is configured to act as a frequency generator, the corresponding LED will flash at a fixed rate as an indication and does not represent the actual triggering frequency.

Scripts

The ECM4000 is designed to run user configured and preprogrammed scripts that are commanded from the serial interface, the digital inputs, or the Ethernet interface. Script files can be created using Notepad or similar text editor program and saved as a file. The file is then transferred to the ECM using the Web Server. Alternatively, script files can be written, edited, and saved using the Web Server only.

Individual scripts are contained within a single script file. A script specifies a series of commands or events that are executed in sequence. Scripts can execute a sequence of commands that can perform the following:
- Control digital outputs or LEDs.
- Control internal solenoids.
- Send a command via TCP over the network.
- Wait a period of time, wait for an external device to be READY, or wait until a digital input state changes. Time-outs are programmable in each situation.
- Start a 50% duty cycle waveform on one of the digital outputs to be used for external triggering.
- Run other scripts within the script file.
- Change configuration variables in the ECM.

Scripts can be configured to:
- Run automatically on power-up.
- Run when a digital input becomes asserted.
- Run when the **RUN** command is issued.

Creating Scripts

A script within a script file begins with the command:
> **BEGIN <script name>**

The script name is user definable and is used as a parameter for the **RUN** command or when setting up a digital in event (**LIST DEVENT**).
A script ends with the command:
> **END**

The syntax to define a script is:
> **BEGIN <script name>**
> <command desired>
> **END**

A double slash // can be used anywhere in a script file to denote comments. The ECM will not process any characters that proceed a double slash. Comments can be used on individual lines or proceeding commands. The following example line is not executed when a script is run:

// Turn on digital output 1

The following example will execute the command and ignore the comments following the //:

Scanivalve

DOUT 1 1 // Turn on digital output 1

Script files and script creation guidelines:
- A script file's filename cannot exceed 32 characters and cannot contain spaces.
- Script files must be ".txt" file format.
- A script file must contain at least one script.
- A maximum of 16 scripts can be contained in a single script file.
- Each script can be a maximum of 127 lines, excluding comments, but including **BEGIN** and **END**.
- Each line in a script cannot exceed a length of 60 characters including comments.
- The last **END** command in a script file must be followed with a new line otherwise the last **END** command will not be recognized.

Examples covering basic script creation and proper syntax can be found in Appendix A. More information on the commands available for use can be found in the Software section.

Loading and Running Scripts
Before a script can be run, the script file must be loaded as an active script. Only one script file can be active at any time. Any scripts that are referenced will call from the currently active script.

Scripts are loaded using the command:
      LOAD <script filename>
The script's filename is case insensitive (SCRIPT.TXT is the same as script.txt). The filename must be included in the file extension.

A script file can also be loaded when the ECM powers on. This is configured using the **SET AUTORUN** variable in **LIST CONFIG**.

Once a script file is active, scripts can be run from the active script file. Only one script can be run at any given time. Scripts can be run manually using the command:
      RUN <script name>
The **RUN** command can also be used during a script to call another script.

Scripts can be triggered to run using a digital input. This requires proper configuration in the digital in event group (**LIST DEVENT**). A script name must be assigned to a digital input number, level, and sequence.

A script can be run when the ECM powers on. This is configured using the **SET AUTORUN** variable in **LIST CONFIG**.

Pre-programmed Scripts
Scanivalve has integrated pre-programmed scripts into the ECM, which are designed to be specifically used with Scanivalve devices.

CALZ (Zero-Offset Calibration)
CALZ is a pre-programmed script in the ECM. The CALZ script was implemented to allow a simple way for the ECM to orchestrate and perform CALZ's (Zero-Offset Calibrations) on one or more Scanivalve pressure scanners. The CALZ group (**LIST CALZ**) specifies the properties of this group, and provides the variables available for configuration. The internal CALZ script can be executed from a digital in, a **RUN CALZ** command, or simply a **CALZ** command.

If a script titled "CALZ" is manually created in a script file, the pre-programmed CALZ script will be disabled.

Note: Using the embedded CALZ script with a DMON Instance/State of STATUS/CALZ can have mixed results. It is best to use one or the other to determine the CALZ state.

Scan
Scan is a pre-programmed script in the ECM. The scan script was implemented to allow a simple way for the ECM to orchestrate scanning and data collection on one or more Scanivalve scanners. The scan group (**LIST SCAN**) specifies the properties of this group, and provides the variables available for configuration. The internal scan script can be executed from a digital in or a **scan** command.

If a script titled "SCAN" is manually created in a script file, the pre-programmed scan script will be disabled.

Note: Using the embedded scan script with a DMON Instance/State of STATUS/SCAN can have mixed results. Due to this, DMON is temporarily disabled if the internal scan script is used.

Optionally, PTP synchronization can be used with the embedded scan script to help synchronize the start of the scan across multiple devices. This requires that PTP is running on the network, and all compatible devices are synchronized via PTP.

There are two methods to utilize PTP with the scan script:

With the use of **SCANPTP** and **SCANDELAY**.
    When **SCANPTP** is enabled and the scan script is executed, the ECM will retrieve its internal time, add the setting for **SCANDELAY** (in seconds) and use the future date and time to set the SST/SSD variables in the devices configured under **SCANDEVICE**. After SST/SSD

is set, it will issue the scan command to all devices. This will cause all compatible devices to enter scan mode and wait until the set time and date before releasing data. As long as all devices are correct synchronized via PTP, all devices will start scanning at the same time, synchronously.

An optional argument after the scan command.
The **SCAN** command, to start the embedded scan script, will accept an argument containing a date/time string to set SST/SSD in all configured devices (same function as **SCANPTP** + **SCANDELAY**). For this to work correctly, **SCANPTP** must be set to 0. See the **SCAN** command for details on the optional argument syntax.

The last configuration for the scan script is the **SCANTIMEOUT** variable. Any time the scan script is used, the ECM will poll that status of all configured devices. When all devices return to STATUS: READY, the ECM will also return to ready. If one or more devices does not return to READY, **SCANTIMEOUT** may be triggered to return the ECM to ready, but will log an error to the errorlog. **SCANTIMEOUT** is set in seconds.

If the SCAN script is executed but must be stopped manually, the command **STOP SCAN** will stop all devices and return all devices to ready (including the ECM).

TCP/IP Commands

A unique feature of the ECM4000 is the ability to send text based commands to other TCP/IP enabled devices. External devices must be configured with an IP address and port, enabled in the device list (**LIST DEVICE**), present on the network, and ready to accept a TCP connection for a command to be sent successfully. A simple command can be used to send commands to other Scanivalve devices such as a DSA-PTP, MPS4264, or a second ECM. Any commands accepted by these devices can be sent using the **TCPOUT** command. The commands sent using the **TCPOUT** command can be sent to a single module or to all enabled devices simultaneously.

When the **TCPOUT** command is issued for the first time, the ECM will establish a TCP/IP connection to the specified device(s) using the IP address and port number assigned to the device(s). The TCP connection(s) will remain connected until a **TCPCLOSE** command is issued. If the TCP/IP is just to be open and established, the command **TCPOPEN** can be used.

The TCP/IP connection to an enabled deivce will only close if the **TCPCLOSE** command is issued, if the device is **DISABLED**, or if the **REBOOT** command is issued.

In the event that a device connection cannot be established, the ECM will issue an ERROR followed by a TCP error code. These error codes can be referred to in Appendix B of this manual.

Warning and Error Handling

Errors and warnings are caused when incorrect syntax is encountered, or when TCP/IP network devices time-out. Errors and warnings are handled differently based on the mode and settings of the ECM4000.

ECM Modes

The ECM has two modes: READY and SCRIPT. When in READY mode, the ECM is ready to accept commands and connections. When a script is run, the ECM will change to SCRIPT mode. The only available commands accepted by the ECM in SCRIPT mode are **STATUS** and **STOP**. If a **STOP** command is issued, the ECM will terminate the script sequence and return to READY mode.

Syntax Errors

When a syntax error occurs in READY mode, the error will be returned immediately. When a syntax error occurs in SCRIPT mode, errors will be handled differently based on the **TOSTOP** variable in **LIST CONFIG**.
- If TOSTOP == 0, the script will continue through the error and write the error to the error log.
  - or -
- If TOSTOP == 1, the script is stopped and the error is written to the error log. The error generated will note that the script has been stopped.

Time-out Errors

When a device becomes timed-out in READY mode, the error will be returned immediately. When a device becomes timed-out in SCRIPT mode, the ECM may generate an error or a warning based on the setting of the TOSTOP variable.
- If TOSTOP == 0 and a device becomes timed-out, the ECM will write a warning to the error log and continue through the script.
- If TOSTOP == 1 and a device becomes timed-out, or a DMON error is generated, the ECM will write an error to the error log and stop the script. The error generated will note that the script has been stopped.

When a time-out warning or error occurs in either READY or SCRIPT mode, the device that timed-out is disabled from future communication until the **CLEAR** command is executed. Errors will remain in the error log until a **CLEAR** command is issued. The errors may be viewed by using the **ERROR** command. The **STATUS** command shows the

Scanivalve

number of errors that have occurred since the last **CLEAR** command.

WARNING AND ERROR ROUTING

The following table shows the routing of error messages for READY and SCRIPT state:

Command Received From	Destination when READY	Destination when SCRIPT
Serial	Serial	Serial
Network* (TCP/IP)	Network	Network
DIN	N/A	Serial and Log
AUTORUN	N/A	Serial and Log
Web Server	Web Server	Web Server

* Multiple TCP/IP connections can be made to the ECM at one time. The error or warning will be returned to the connection that the command was sent from.

IEEE1588v2 PRECISION TIME PROTOCOL

The ECM4000 is compliant with IEEE1588v2 Precision Time Protocol (PTP). The ECM's PTP engine can be configured as follows:

- Disabled.
- Enabled as a PTP slave device.
- Enabled as a PTP master device.

When the ECM4000 is configured as a PTP slave device (**SET PTPEN 1**), the ECM will listen for PTPv2 packets being multicasted on the network. This allows for the ECM's internal time and date to be set and synchronized when a PTP Grandmaster is active on the same network. This synchronization will allow the waveform outputs on digital outputs 1 through 4 to be synchronized with PTP time.

When the ECM4000 is configured as a PTP master device (**SET PTPEN 2**), the ECM will multicast PTPv2 packets over the network, similar to a PTP Grandmaster. The ECM will use its internal time and date when multicasting these packets for synchronization. The ECM configured as a PTP master allows other PTP enabled Scanivalve devices to be synchronized using the internal time and date of the ECM.

Note: Only one PTP Grandmaster or PTP master can be present on a network at any given time.

When the ECM's PTP engine is disabled or enabled as a PTP master, the ECM's time and date can be manually set using the **SETTIME** command, or using the "Setup" page on the Web Server.

Other PTP related commands and settings can be referenced in LIST PTP and PTP Commands in Section 5: Software.

DEVICE MONITOR (DMON)

Device Monitor, or **DMON**, is a configurable monitoring system that can be used to monitor the state or status of any enabled and connected device, and can change the status of the 8 digital outputs based on the device's response. This feature can be useful in applications where an external panel of status LEDs is required to indicate the state of connected device.

Device monitor is configured using "instances." The "instance" name is the command used to poll any enabled and connected devices, in which the ECM expects a response. Every "instance" created will have "states," which are the expected responses from the connected device after the "instance" command is sent. Each "state" will have an assigned bit pattern that will determine the states of the digital outputs, which may be on, off, not changed, and more if all devices match this "state."

When an "instance" and "state" are configured and the "instance" is enabled, the ECM will send the "instance" command in 1 second intervals to all enabled devices and will read the response from each device. If the response matches a configured "state," the ECM will confirm that all enabled devices match the same state and set the digital outputs according to the configured "state."

Each "instance" created will be preconfigured with two embedded "states": ERROR and TRANS (transition). Each of these states has an assigned digital output pattern which can be changed by the user, however the state itself cannot be removed.

The ERROR state is when all connected devices do not respond to the "instance" command, when a device is timed-out, if a device's response is not configured as a "state," or when all connected devices do not meet the same state. The ERROR state is cleared when all enabled devices match the same state.

The transition state (TRANS) is the transitional period where devices may be changing or are in transition to a different state. The amount of time that determines the transitional state is controlled by the **DMON TIMEOUT** variable. When DMON is notified that a state has changed, DMON will enter the transition state and poll for the amount of time set in the **DMON TIMEOUT**. Once in the transitional state, the following can occur:

- If all devices match the same configured state before the timeout, DMON will update to the current matched state.
- If devices do not match the same configured state by the timeout, DMON updates to the ERROR state.
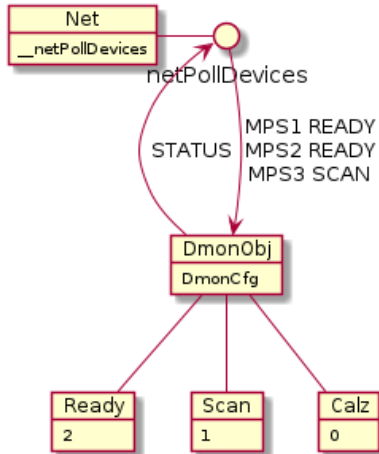

Figure 3-5: DMON Polling Process

When a DMON instance is enabled, the process will run in the background and update the digital outputs as configured. A "one-shot" poll command is available that will poll the instance one time, and display the response from each device, output to the destination from where the command was issued. The **DMON POLL** command, in conjunction with the **STATUS D** command, are useful tools that can be used to troubleshoot devices when errors are present, or if states are not updating correctly.

DMON POLL can also can be used to validate the state of enabled devices. If the command is used to validate a state, it will either:
- Return nothing if the condition is true
- ECM will provide an error if the condition is not true.

This can be used in conjunction with the **TOSTOP** variable to validate device's states before continuing in scripts, whereas a successfully validated state will have no effect.

Another unique function of DMON is the ability to maintain and re-establish connections to enabled TCP devices. If a device and a DMON instance is enabled, the DMON instance will always open and maintain the TCP connection to the device. For example; if a device is enabled, not-timed out, and connected, DMON is enabled and operating, and the device loses connection, DMON will move to the ERROR state, disconnect the socket, and attempt to reopen the socket. If the device is reconnected to the network and becomes ready for a new connection, DMON will eventually establish this connection and update the state.

Examples of DMON configuration and uses with other Scanivalve devices can be found in Appendix A of the manual.

DMON POLL can also can be used to validate the state of enabled devices. If the command is used to validate a state, it will either:
- Return nothing if the condition is true
- ECM will provide an error if the condition is not true.

This can be used in conjunction with the **TOSTOP** variable to validate device's states before continuing in scripts, whereas a successfully validated state will have no effect.

Another unique function of DMON is the ability to maintain and re-establish connections to enabled TCP devices. If a device and a DMON instance is enabled, the DMON instance will always open and maintain the TCP connection to the device. For example; if a device is enabled, not-timed out, and connected, DMON is enabled and operating, and the device loses connection, DMON will move to the ERROR state, disconnect the socket, and attempt to reopen the socket. If the device is reconnected to the network and becomes ready for a new connection, DMON will eventually establish this connection and update the state.

Examples of DMON configuration and uses with other Scanivalve devices can be found in Appendix A of the manual.

Scanivalve

[INTENTIONALLY LEFT BLANK]

Web Server

The ECM4000 is configured with a single user Web Server application which provides a simple method for communication, configurations, and operations from virtually any host computer. The Web Server requires no additional software be installed on the host computer and provides a simple graphical interface for the user to use the ECM to its full capacity.
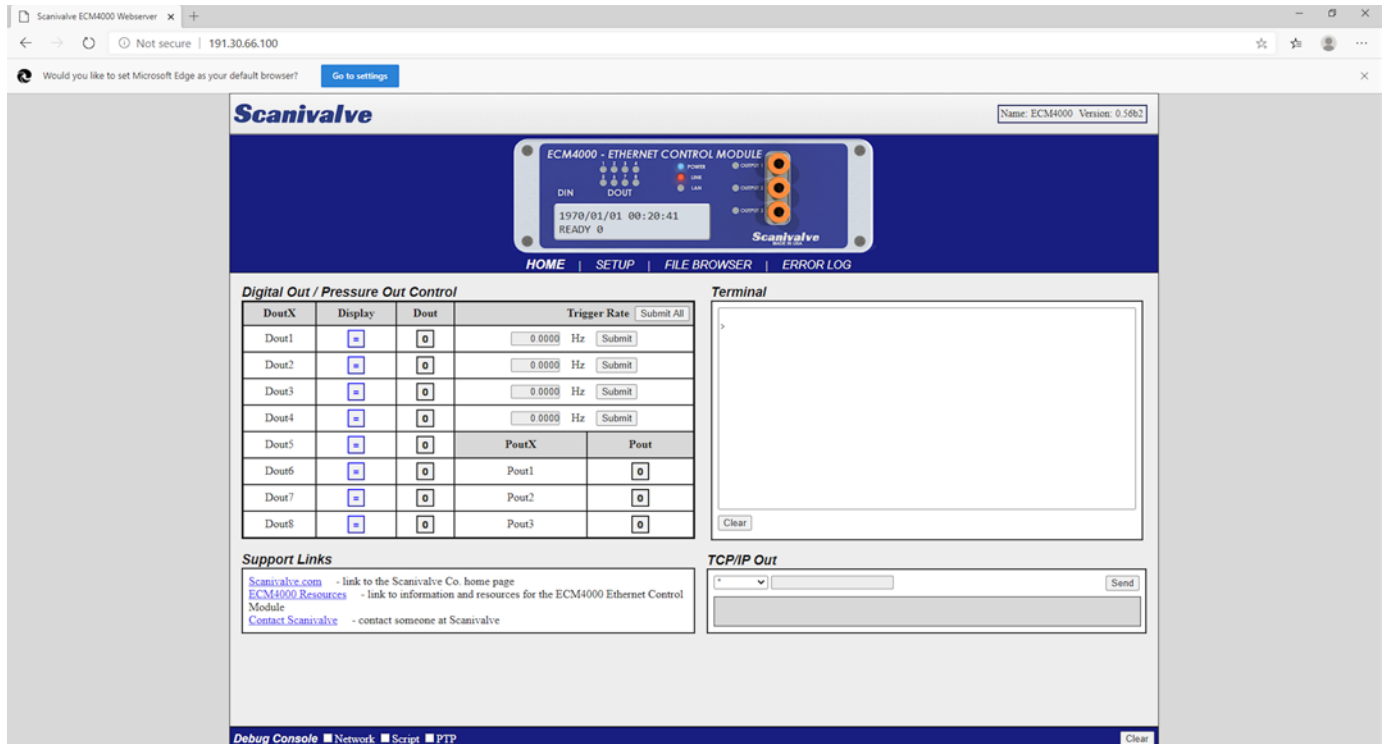
Figure 3-6: Web Server Home Screen

The Web Server has been tested with several web browsers to verify functionality and compatibility. Those that were tested are:

Google Chrome
Microsoft Edge
Mozilla Firefox

To connect to the ECM4000's Web Server, simply open one of the supported browsers and enter the ECM's IP address into the address bar.

Once connected, the Web Server's Home screen will be displayed as shown in figure 3-5.

Basic module information will be displayed in the top right corner of the window. This will include the model number and current firmware version.

Scanivalve

WEB SERVER: MAIN DISPLAY

FIGURE 3-7: MAIN DISPLAY

The top portion of the Web Server is the "Main Display." This display will remain constant through all pages of the Web Server. This display provides a digital representation of the ECM's front panel and will provide visual indicators for the digital outs, status LEDs, and pneumatic outs. It also includes a status window that will display the current time in the ECM as well as the status. The main display is updated once per second to reflect the most current status of the ECM.
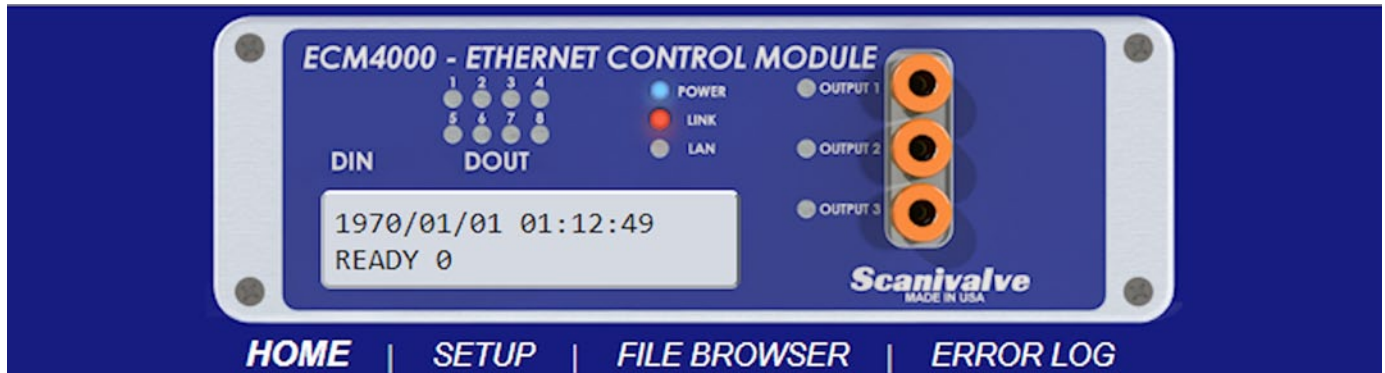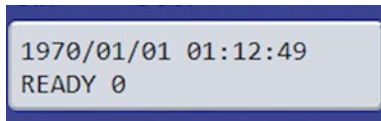
 -- This area shows the status of the Status LEDs (default configuration will follow the digital out statuses).
    If the digital out is OFF, it will be GRAY (shown).
    If the digital out is ON, it will be GREEN.
    If the digital out is providing a trigger waveform, it will flash GREEN at a fixed rate.

 -- This area shows the status of the ECM.
    POWER - The indicator will be lit BLUE if the ECM is powered on.
    LINK   - This indicator will be lit RED if a Ethernet connection is detected.
    LAN   - This indicator will flash GREEN if communication is possible.

 -- This area shows the status of the pneumatic outs.
    Each output has a corresponding indicator.
    If the indicator is GRAY (shown), the pneumatic output is not active and the solenoid is closed.
    If the indicator is GREEN, the pneumatic output is active and the solenoid is open.

 -- This area shows the ECM's date and time, and the status and error count of the ECM.

The main display also provides the links to the other pages available in the Web Server:
HOME, **SETUP**, **FILE BROWSER**, and **ERROR LOG**.
Each of these pages will be discussed in the following sub-sections.

The bottom of each page will display the Debug console bar. This bar will remain constant through all pages of the ECM.

When a debug option is checked (Network, Script, PTP), the console will expand and will display debug information for the selected item. If no options are checked, the console will collapse. The clear button will clear any content in the console.

For more information on the debug options, please see **SET DEBUG** in the software section of the manual.

WEB SERVER: HOME

The home page is the first page loaded when connecting to the ECM Web Server. This page provides manual controls for the digital and pneumatic outputs, the ability to send TCP commands to configured and enabled devices, a "Terminal," and links for ECM4000 support.

HOME | SETUP | FILE BROWSER | ERROR LOG

Digital Out / Pressure Out Control

DoutX	Display	Dout	Trigger Rate		Submit All
Dout1	=	0	0.0000	Hz	Submit
Dout2	=	0	0.0000	Hz	Submit
Dout3	=	0	0.0000	Hz	Submit
Dout4	=	0	0.0000	Hz	Submit
Dout5	=	0	**PoutX**		**Pout**
Dout6	=	0	Pout1		0
Dout7	=	0	Pout2		0
Dout8	=	0	Pout3		0

This table allows for manual control over the digital and pneumatic outputs.

DOUTX -- This column displays the 8 available digital outputs.
DISPLAY -- This column sets the status for each LED Indicator. Clicking a box will toggle through three settings:

- [=] -- Set the indicator (LED) to follow the DOUT status
- [1] -- Turns the indicator (LED) ON, regardless of DOUT status
- [0] -- Turns the indicator (LED) OFF, regardless of DOUT status

DOUT -- This column selects the status of each digital output. Clicking a box will toggle through three settings:

- [T] -- Set the digital out as a trigger (per configured rate)
- [1] -- Turns the Digital Out ON
- [0] -- Turns the Digital Out OFF

TRIGGER RATE -- This column allows the trigger rate to be configured (in Hz). Once a value has been entered, use the SUBMIT button to apply these changes, or use the SUBMIT ALL button to set all four trigger rates. The digital out will only trigger at the set rate if the DOUT status is set to [T].
POUTX -- This column displays the three available pneumatic outputs.
POUT -- This column selects the status of each pneumatic output. Clicking a box will toggle through two settings:

- [1] -- Turns the Pneumatic Out ON
- [0] -- Turns the Pneumatic Out OFF

Support Links

Scanivalve.com   - link to the Scanivalve Co. home page
ECM4000 Resources   - link to information and resources for the ECM4000 Ethernet Control Module
Contact Scanivalve   - contact someone at Scanivalve

This section provides links that will be directed to different resources on Scanivalve's website. These pages will provide general information, ECM4000 resources (such as links to the manual or data sheet), and a link to Scanivalve's contact form.
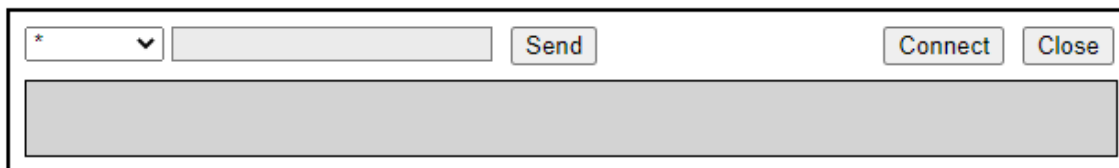
NOTE: These hyperlinks will only work successfully if an internet connection is available.

Scanivalve

WEB SERVER: HOME (CONT.)

HOME | SETUP | FILE BROWSER | ERROR LOG

Terminal

>

Clear

This section provides a "Terminal" emulator screen, allowing the user to send simple text commands to the ECM. While the Web Server pages provide many configurations and commands, not all variables are available in the GUI. The "Terminal" screen is a simple means to check or change variables or issue commands manually.

NOTE: The ">" prompt in the Terminal section is fixed and is unaffected by the **SET PROMPT** variable in **LIST CONFIG**.

TCP Client

| * ∨ | | Send | | Connect | Close |

This section allows for TCP/IP commands to be sent to any or all configured and enabled devices. It also provides buttons to open and connect to devices (Connect), or close the TCP connection to active devices (Close). These buttons simulate the **TCPOUT**, **TCPOPEN,** and **TCPCLOSE** commands.

The drop down box will provide a list of enabled devices. Select the device to open or close the TCP connection, or send a command to. If the name of a device does not appear in this list, then the device is not enabled or configured. The default setting is the wildcard metacharacter (*), which will perform the selected function to all enabled devices.

The desired command to be sent to the device(s) can be entered in the field next to the device drop down box. Once the command is entered, click the SEND button to send the command to the device(s). The field below will display confirmation if the command is sent successfully. If the command could not be sent, an error will be displayed.

WEB SERVER: SETUP

The Setup page is used for configuring devices, configuring Digital In assignments, and configuring Precision Time Protocol settings.

HOME | **SETUP** | FILE BROWSER | ERROR LOG

Device List

Name	Ip:Port	Type	Enabled	Delete
DSA16	191.30.82.3:23	DSA	☑	🗑
DTS32	191.30.105.117:23	DTS	☑	🗑
SPC4050	191.30.44.45:1025	SPC	☑	🗑
MPS123	191.30.90.123:23	MPSE	☑	🗑
MPS21	191.30.90.21:23	MPS	☑	🗑

Add Remote Device

Name [   ]   IpAddr [   ]

Type [DSM ▼]   Port [   ]

[Submit]   [Save]

This section displays configured remote devices, allows device properties to be changed, or new devices to be added.

NAME -- This column displays the name of the configured device.

IP:PORT -- This column displays the IP address and port number of the configured device.

TYPE -- This column displays the device type.

ENABLED -- This column allows for devices to be ENABLED or DISABLED. Checking the box will ENABLE the device. Un-checking the box will DISABLE the device.

DELETE -- This column provides a link to delete a device.

ADD REMOTE DEVICE -- This section allows for new remote devices to be added or for existing devices to be altered.

DEVICE NAME -- Enter the name of the device. *Do not use spaces in this field.*

IPADDR -- Enter the IP address of the remote device. *IPv4 addresses only.*

PORT -- Enter the TCP Port number of the device.

TYPE -- Select the proper device type of the remote device in this list.

When all fields have been filled out correctly, click SUBMIT to add the device.

To change a device, select the name of the device by clicking the name of the device in the **NAME** column. This action will populate the **ADD REMOTE DEVICE** fields. Only the **IPADDR**, **PORT**, and **TYPE** can be altered. Once these settings are changed, click SUBMIT. If the **DEVICE NAME** is altered and SUBMIT is clicked, a new device will be added to the list.

Once the device configurations are completed, use the SAVE button below the table to save the device list. This action will only save the device list (LIST DEVICE). Other settings will not be saved, including the Din Event Matrix and PTP settings.

The maximum amount of devices supported in the ECM is 32.

Scanivalve

WEB SERVER: SETUP (CONT.)

Each digital input has a field that will allow for script names to be entered, based on the digital input level and sequence number. Only script names can be entered into these fields.

To assign a script to a digital input event, enter the name of the script in a gray empty box in the same row and click SUBMIT. Once a Digital In Event has been assigned, the gray box will disappear and will only display the name of the script. Examples of submitted events would be POUT, ZCDIN, DOUT, and LED on digital inputs 1 through 4, level 0, sequence 1. This is also true with ZCDIN, level 0, sequence 2.

When a script is assigned to sequence 1 (either 0 or 1 level), it will also populate the second sequence temporarily. The script name will be displayed in parenthesis as filler text in the field - example: (POUT). The first time the digital in goes low, the POUT script will be run. The second time the digital input goes low, the script POUT will run again, fulfilling the sequence and restarting. A different script can be added to this field if a sequence is needed. Enter the name of the script into the desired field and click SUBMIT.

A digital in can be changed or deleted by clicking on the desired text. Clicking on script LED in level 0, sequence 1 will populate the field at the bottom of the table. The information in this field will be displayed in the same syntax as needed in LIST DEVENT (see the DEVENT in the software section of this manual for more information).
The event can either be altered and submitted, or the event can be deleted using the DELETE DIN EVENT button.
The wildcard metacharacter (*) can be entered into the lower field, and DELETE DIN EVENT clicked to delete all digital input events.
The SAVE button can be used to save the digital in event settings. Other settings will not be saved.

ZCDIN will always be displayed in RED. This is the pre-programmed CALZ routine that can be configured in **LIST CALZ**. The only way ZCDIN can be removed or reassigned is by configuring ZCDIN in LIST CALZ. Please see the ZCDIN variable in the software section of this manual.

WEB SERVER: SETUP (CONT.)

HOME | **SETUP** | FILE BROWSER | ERROR LOG

IEEE-1588 Precision Time Protocol

PTP Engine ○ SLAVE ○ MASTER ● DISABLED
Statistical Output ○ SERIAL ○ TCP/IP ● NONE
☐ UTC Offset [00:00:00]
☐ Set UTC Time [YYYY/MM/DD hh:mm:ss]

[Submit] [Save]   [Retrieve Network Time]   [Set Network Time]

This section will control the IEEE-1588v2 Precision Time Protocol variables.

PTP ENGINE -- The PTP Engine can be enabled/disabled by using the three radio buttons:

SLAVE -- Will assign the ECM to listen for PTPv2 packets on a network and synchronize with a PTP Grandmaster.

MASTER -- Will assign the ECM to act as a PTP master. It will broadcast and sync slave devices using its internal time.

DISABLED -- PTP functionality will be disabled.

If any of these settings are changed, the SUBMIT button must be used to activate the changes.
In order for the ECM to change PTP states from Slave to Master, or Master to Slave, the PTP engine must be disabled and submitted first.

STATISTICAL OUTPUT -- PTP Statistical information can be output from the ECM when enabled as a slave. Use the radio buttons to direct the PTP statistical data.

SERIAL -- Will display PTP statistical data to the serial port.
TCP/IP -- Will display PTP statistical data to the last connected TCP/IP port.
NONE -- No PTP statistical data will be output.

If any of these settings are changed, the SUBMIT button must be used to activate the changes.

UTC OFFSET -- This field can be used to set the UTC Offset. The box must be checked in order to fill in the field. The box has filler text to show the correct syntax. Once the field is filled out, use the SUBMIT button to submit these changes. The check box will clear after being set. The SUBMIT button will not submit these settings if the box is unchecked.
Note: The SAVE button will save this setting to the ECM's memory.

SET UTC TIME -- This field can be used to set the internal time of the ECM. This time can be configured as desired. The box must be checked to fill in the field. The field has filler text to show the correct syntax. Once this field is filled out, use the SUBMIT button to submit these changed. The check box will clear after being set. The SUBMIT button will not submit these settings if the box is unchecked.
Note: The SAVE button will not save this setting to the ECM's memory. Date/time cannot be maintained through a power cycle.

RETRIEVE NETWORK TIME -- If this button is clicked, the ECM will retrieve the network time from the browser and populate the **SET UTC TIME** and **UTCOFFSET**. Each field will be checked and can be edited or submitted.

SET NETWORK TIME -- If this button is clicked, the ECM will retrieve the network time from the browser and automatically set the **UTC TIME** and **UTCOFFSET**. Once clicked, the changes will be reflected in the status area of the main display.
Note: the host computer connected to the ECM does not require an internet connection to retrieve the time. Most web browsers will get time from the computer.

The SAVE button can be used to save the Precision Time Protocol settings. Other settings will not be saved.

Scanivalve

WEB SERVER: FILE BROWSER

The File Browser page is used to view files and scripts stored in the ECM, edit files or scripts, create and edit new scripts, upload files, and update the ECM's firmware.

HOME | SETUP | **FILE BROWSER** | ERROR LOG

Filesystem

Config Files	Size	Download	Delete
ECM4000.hex	1197836B	⬇	🗑
ptp.cfg	50B	⬇	🗑
config.cfg	93B	⬇	🗑
devent.cfg	22B	⬇	🗑
device.cfg	83B	⬇	🗑
calz.cfg	93B	⬇	🗑
ip.cfg	91B	⬇	🗑

Script Files	Size	Download	Delete
Demo.txt	2703B	⬇	🗑

Upload File

[Choose File] No file chosen   [Upload]

This section displays all files that are currently in the ECM's memory. It is broken up into two file types: **CONFIG FILES** and **SCRIPT FILES.** There is also a section where files can be uploaded.

CONFIG FILES -- This column displays the names of the configuration files.
SIZE -- This column displays the size of the file in bytes (B).
DOWNLOAD -- This column provides a link that will initiate a download of a file to the host computer.
DELETE -- This column provides a link that will delete a file.

Only .cfg (device configuration) and .hex (ECM application) files will be displayed in this table. All cfg files contain ECM settings, including the IP address. Deleting these files may have adverse effects.

SCRIPT FILES -- This column displays the names of the script files.
SIZE -- This column displays the size of the file in bytes (B).
DOWNLOAD -- This column provides a link that will initiate a download of a file to the host computer.
DELETE -- This column provides a link that will delete a file.

Only .txt (script) files will be displayed in this table.

UPLOAD FILE -- This area allows for files to be uploaded to the ECM, including firmware files, script files, and configuration files.

All files can be double clicked to activate a function:
> **CONFIG (.cfg) FILES** -- When a .cfg file is double clicked, it will open the "Text Editor" in the adjacent console. The contents of this file will be displayed. This file can be edited and saved using the SAVE button in the console. Remember that editing or changing these files may have adverse effects.
> **APPLICATION (.hex) FILE** -- When the ECM4000.hex file is double clicked, a dialog box will appear. Double clicking will begin a firmware update. Press OK to continue, or CANCEL to cancel. More information on the firmware update process can be found using the "Firmware Installation via Web Server" procedure.
> Once firmware has been updated, this file may be deleted. This file does not have to be stored on the ECM for operation.
> **SCRIPT (.txt) FILE** -- When a .txt file is double clicked, it will open the "Script Console" in the adjacent console AND load a script (see LOAD command). This will provide a means to view all scripts in the file and run scripts. The "Text Editor" tab will also be available in order to edit script files or add additional scripts. Any edits to the script file can be saved using the SAVE button in the "Text Editor" console. If the filename does not change, it will overwrite the designated file. If a new filename is entered, a new file will be created and saved. Script file names cannot include spaces and cannot exceed 32 characters in length.

Note: Files that do NOT match .cfg, .txt, and ECM4000.hex are considered unsupported and the ECM will not upload these files.

WEB SERVER: FILE BROWSER (CONT.)

SCRIPT CONSOLE

This window will allow you to view all scripts within a script file and provide the ability to run and stop scripts.

The script console tab will only be available when a script file (.txt) is selected and will be the default window.

FILENAME -- This field will display the filename of the script file.

RUN -- This column provides a button to run a script. When the button is clicked, the script will run and the button will change to a stop button. All other run buttons will be disabled while a script is running. The stop button can be used to manually stop a running script.

NAME -- This column shows the script names.

SHOW -- This column provides a button to display all commands in the corresponding script.

TEXT EDITOR

This window will allow you to view, edit, or create script (.txt) or configuration (.cfg) files.

The text editor tab will always be available and will be the default window when configuration files are selected.

FILENAME -- This field will display the filename of the script file. Filenames must include the file extension (.txt or .cfg). The filename cannot exceed 32 characters and cannot contain spaces.

SAVE -- This button is used to save the file being created or edited.

The edit window will respond like a generic text editor application.
Script lines cannot exceed 60 characters.
Scripts cannot exceed 127 lines.
Script files cannot exceed 16 scripts.
Using text after "//" will be ignored and can be used for notes or comments.

Scanivalve

WEB SERVER: ERROR LOG

The Error Log page is used to view errors messages stored in the ECM. A red error light will appear at the end of "LOG" whenever errors are listed on this page. The errors will be listed using the same syntax outlined in the **ERROR** command.

HOME | SETUP | **FILE BROWSER** | ERROR LOG

Error Log [Clear]

No Errors

The error log section will simply display any logged errors. Any errors or warnings generated will be displayed in the box and will include basic information about the error.

A maximum of 100 errors will be displayed. The "Clear" button can be used to clear all errors.

Information on error reporting can be found in the "Warning and Error Handling" section, and the error syntax can be referenced in the **ERROR** command description.

[INTENTIONALLY LEFT BLANK]

SECTION 4: HARDWARE

PNEUMATIC CONNECTIONS

The ECM4000 has five different pneumatic connections that are available: Supply input, Vent, and outputs one through three. The supply input is tied to all three outputs whose flow is electronically controlled by three individual solenoids. When a solenoid(s) is activated, the supply air will flow through and out the pneumatic out(s). When the solenoid is closed, any trapped air between the pneumatic output and the plumbed source will be vented out the "Vent" port on the back of the module.

Air supplied to the Supply Input port must be clean, dry instrument grade air. Nitrogen is a very good alternative when instrument grade air is not available. The maximum allowable input pressure is 120psi.

SUPPLY INPUT AND VENT

FIGURE 4-1: SUPPLY INPUT AND VENT

The "Supply Input" and "Vent" ports are located on the back panel of the ECM. The "Supply Input" port is the ECM's air supply connection. Clean, dry instrument grade air or nitrogen should be used. Air supplied to this port can be passed through any of the three pneumatic outputs with the use of internal solenoids. These solenoids can be activated and deactivated using software commands. The pressure supplied to the "Supply Input" must not exceed 120psi.

The "Vent" port on the ECM allows air supplied from the pneumatic outputs to be vented when the solenoids are deactivated. Any back pressure from the pneumatic outs will vent through this port. When the solenoids are closed, the air supplied at the "Supply Input" will not pass through to this port. If this port is plugged, air directed to the outputs will not properly vent when the solenoid is closed, and may result in trapped air on the output lines.

These pneumatic connections are 1/4" Swagelok® compression fittings.

MAKING SWAGELOK® COMPRESSION CONNECTIONS

Each Swagelok® compression fitting includes 4 parts: the nut, the front ferrule, the back ferrule, and the body. Squarely cut the end of the tubing being used. Slide the nut and both ferrules up on the tubing, then insert it all of the way into the Swagelok® body. Tighten the nut finger tight to the point at which the tube cannot be removed by hand. Finally, use a wrench to tighten the nut by 3/4 to 1-1/4 more turns. There is no need to tighten the nut any more than this.
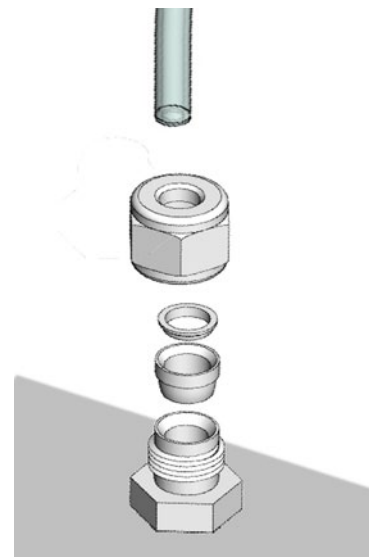
FIGURE 4-2: SWAGELOK® COMPRESSION CONNECTIONS

Many types of 1/4" OD tubing can be connected to the 1/4" compression fitting. Scanivalve offers several styles of adaptors to convert 1/4" Swagelok® compression fittings to common Scanivalve bulged stainless steel tubulation sizes such as 0.063" OD and 0.125" OD. Please see the Scanivalve Accessory Catalog for more information on available tubing and adaptors.

PNEUMATIC OUTPUTS

FIGURE 4-3: PNEUMATIC OUTPUTS

There are 3 pneumatic outputs available. These three outputs are located on the front panel of the ECM. The outputs are controlled by individual internal solenoids which can be controlled with the use of software commands manually or within a script. When a solenoid is activated, pressure supplied to the "Supply Input" will flow through the pneumatic output that has been activated. When the solenoid is deactivated, any back pressure in the line will be vented through the "Vent" port on the back of the ECM. All three solenoids can be operated independently or together. The command used to operate these solenoids is **POUT**, which can be referenced in the software section of this manual.

Each pneumatic output has a LED indicator to show the active status of the internal solenoid. When the corresponding LED is lit, the solenoid is active and open. When the LED is not lit, the solenoid is closed.

All three pneumatic outputs can be configured with the Push-In style fittings made to accept 3/16" OD tubing (standard) or using the 0.063" bulged tubulations (optional).

MAKING TUBING CONNECTIONS TO PUSH-IN STYLE FITTINGS
Each Push-In style fitting allows for many types of 3/16" OD tubing to connect with ease. Squarely cut the end of the tubing being used. Insert the tubing into the center of the fitting and depress until the tubing bottoms out. The tubing will now be secured in place.

To remove the tubing, simply depress the collar of the fitting and remove the tubing.
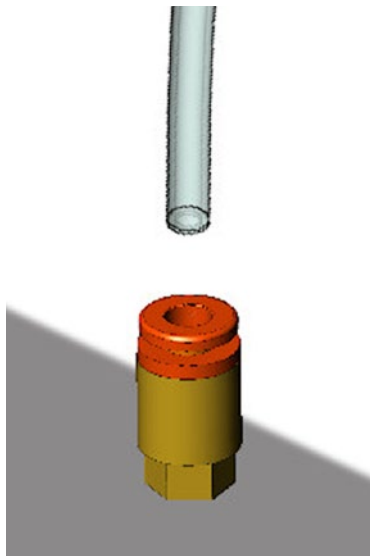
FIGURE 4-4: PUSH-IN STYLE CONNECTIONS

ELECTRICAL CONNECTORS

The ECM4000 has four electrical connectors located on the front and back panel of the module. The back panel contains the connectors for the ECM power and Ethernet connections. The front panel contains the connectors for the digital input and output connections.

POWER CONNECTOR & REQUIREMENTS
The power connector is located on the back panel of the module. The connector is an Amphenol PT02A-8-3P. A mating power connector is supplied with each ECM if DC voltage is to be supplied by the user. The Pin-Out diagram for the power connector can be found in Figure 4-5.

CAUTION! Do not make or break the power connector with power applied! Doing so risks damage to the module.

The standard ECM with no heater requires +9-36Vdc power and will consume no more than 6.5W. Power consumption will be at its greatest when all three solenoids are "On."

The ECM with integrated heater requires 20 - 36Vdc power and will consume no more than 65W. Power consumption will be at its greatest when all three solenoids are "On" and the heaters are on at full capacity. The power consumption will drop when the module has reached a normal operating temperature. If the module is used in an environment where the ambient temperature is 0°C or less, power requirements could continuously remain near the 65W max.

CO-615
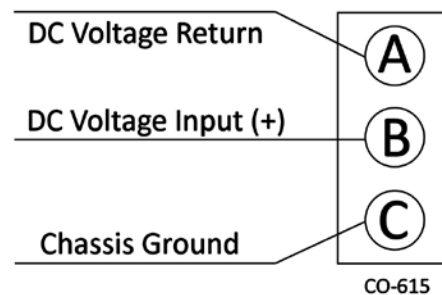
FIGURE 4-5 - POWER WIRING

ETHERNET CONNECTOR
The primary means of communication with the ECM4000 module is the 100Base-T Ethernet port with MDIX auto-crossing. The ECM uses a M12, 4-pin, D-Code connector for the Ethernet connection. This is a latching or threaded

Scanivalve

connector to prevent the connector from becoming unseated when subjected to vibrations. A 3ft M12 to male RJ45 cable is provided with every ECM4000.

Additional Ethernet cables for the ECM4000 modules can be ordered from Scanivalve in any length up to 100 feet (30 meters). Scanivalve's part number for these cables is 155035.

M12 cables, connectors, and parts can also be purchased off-the-shelf from third party vendors. Custom Ethernet cables can be built using the mating connectors and a shielded Category 5e cable or better. The Ethernet Pin-Out diagram is shown in Figure 4-6.
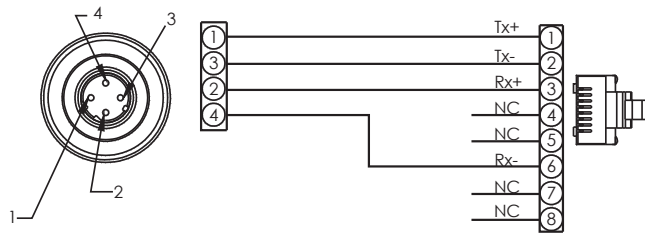
FIGURE 4-6: 100BASE-T ETHERNET CABLES

DIGITAL OUT CONNECTOR

The Digital Out connector provides 8 discrete digital out signals to be used with external devices. The ECM uses a M12, 12-pin, A Code Socket connector for the digital out connection. This is a latching or threaded connector to prevent the connector from becoming unseated when subjected to vibrations. The mating connector (cable connector) is a M12, 12-pin A Code Pin connector. A 5ft (1.5m) flying leads mating cable is provided with every ECM.

Different voltages are provided at the digital out connector to be used with external devices. This includes a single lead for +5Vdc or two leads to directly use the ECM's DC input power supply (+9-36Vdc). Each channel has a max current rating of 500mA. Each of the digital outs may be operated or controlled using software commands manually or within a script. Information on the **DOUT** command can be found in the Software section of this manual.

Mating connectors or cables can be purchased through Scanivalve or third party vendors. Custom digital out cables can be manufactured using the Pin-Out diagram shown in Figure 4-7.
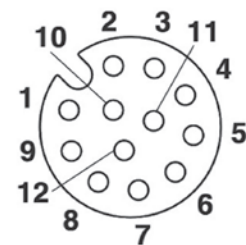
FIGURE 4-7: DIGITAL OUT WIRING

DIGITAL IN CONNECTOR

The Digital In connector allows for 8 discrete digital inputs. The ECM uses a M12, 12-pin, A Code Pin connector for the digital out connection. This is a latching or threaded connector to prevent the connector from becoming unseated when subjected to vibrations. The mating connector (cable connector) is a M12, 12-pin A Code Socket connector. A 5ft (1.5m) flying leads mating cable is provided with every ECM.

Each of the 8 digital ins will supply +5Vdc. Two common ground lines are available to complete the circuit. The connection also provides a separate constant +5Vdc line, as well as the ECM input supply power (+9-36Vdc). These constant connections can be used to tie to the digital input for higher power usage. If used, a 100KΩ resistor must be used in series, per channel used. Refer to Figure 4-9 for an example schematic.
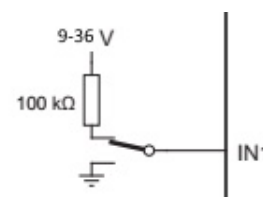
FIGURE 4-8: 9-36VDC DIGITAL IN

Each digital in connection can be used to trigger events or scripts. Each digital in can also be configured to be "edge-sensing" or allow "high/low" logic to allow for momentary or latching switches to be used. The digital in configurations are assigned in **LIST DEVENT**. These configurations can be found in the Software section of the manual.

Mating connectors or cables can be purchased through Scanivalve or third party vendors. Custom digital in cables can be manufactured using the Pin-Out diagram shown in Figure 4-9.
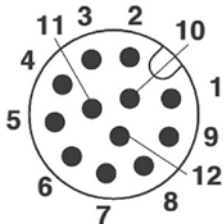
FIGURE 4-9: DIGITAL IN WIRING

<u>**SERIAL CONNECTOR**</u>

The serial connector on the back panel of the ECM provides a secondary means of communications with the module. The serial connection is primarily used for troubleshooting, or finding or changing the IP address of the module if the IP address is not known.

The bulkhead connector is an Amphenol JTP02RE8-6P. All ECM4000 modules are shipped with a mating connector (Amphenol JT06RE8-6S-SR or MS27473T8F6S) that can be used to fabricate a Serial Communications cable. Alternately, a Serial Communication cable can be ordered from Scanivalve using Scanivalve part number 155829.

The wiring diagram for the RS-232 output is shown in figure 4-10. The cable wiring must connect the Tx output from the host computer to the Rx input of the ECM4000. Also,

the RX input of the host computer must connect to the Tx output of the ECM4000.
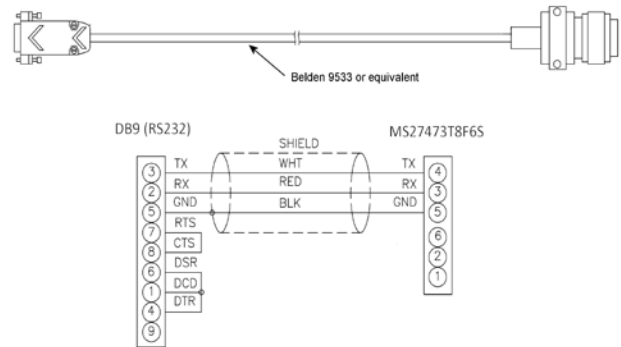
FIGURE 4-10: SERIAL COMMUNICATIONS CABLE

Settings for establishing a serial connection to the ECM module are as follows:

> Bits per second: 9600
> Data bits: 8
> Parity: none
> Stop bits: 1
> Flow control: none

A serial and Ethernet connection can be established simultaneously.

Scanivalve

[INTENTIONALLY LEFT BLANK]

SECTION 5: SOFTWARE

OVERVIEW

The ECM4000 is essentially a single user network server that supports connections from a Telnet Client and Web Browser Client. The ECM4000 supports serial communication that can be used to set configuration variables, diagnostics and trouble-shooting, or when the ECM's IP address is not known.

COMMAND FORMAT

Each of the commands are explained with the following sections: command, syntax, arguments, description, and returns.

COMMAND lists the name of the command.

SYNTAX lists the format of the command. The following conventions are used:

  BP - Boldface letters indicate command keywords and operators. Within the discussion of syntax, bold type indicates that the text must be entered exactly as shown.

  - Words in < > indicate place holders for parameters that must be supplied.

  [parameter] - Words in [ ] indicate place holders for optional parameters that are not required.

  data - Example data is shown in *italics*.

  data - Example commands are show in **bold**.

  , - Commas separate options, only one of the options may be used.

  <CR> - The carriage-return key, sometimes marked as a bent arrow, Enter, or Return on the keyboard, is called <CR>.

  Spaces, as used in the syntax, are entered as spaces.

DESCRIPTION describes the function of the command.

RETURNS lists the format of the information that the unit returns to the host.

  A PROMPT may be output when the ECM4000 is ready to accept a command. The prompt in the ECM is configurable. This manual shows all returns and examples as if **SET PROMPT 1 >** is configured.

VALID MODES is the state of the ECM that will accept the command.

EXAMPLES are example commands and do not necessarily apply to all applications. When "//" is used, this is a comment to explain the commands that proceed it and are not used in the command sequence.

NOTES are additional comments relevant to the command or variables.

TCP/IP does not guarantee that packet boundaries will be maintained between a Host and a ECM4000. Therefore, ALL commands from a Host MUST be terminated properly with one of two options using the NL configuration variable. The two options are:

> CR-LF (ASCII 13 - ASCII 10) or CR (ASCII 13)

The maximum string length for commands sent to the ECM4000 is 79 characters. Any command received that is longer than 79 characters will be discarded and an error will be generated.

When a communications variable is modified (those in the LIST IP group), the ECM4000 software must be restarted in order for the changes to take effect.

Several commands in the ECM accept metacharacters that have uniquely defined actions. These metacharacters include:

  * - The asterisk is used as a "wildcard" option. When used in most cases, the command will apply to all available values for the specified command (e.g., ENABLE * will enable all configured devices).

  ? - The question mark is used to query the status of the applicable command it is used in (e.g., DOUT ? will provide the current status of the digital outputs).

  # - The pound sign (otcothorpe) is use to set a bit sequence for the applicable command it is used in (e.g., DOUT # <bit sequence>).

The metacharacters will be listed as accepted values under all applicable commands.

Scanivalve

GENERAL CONTROL COMMANDS

LOAD SCRIPT

COMMAND	LOAD SCRIPT
DESCRIPTION	The ECM4000 will load the specified script file and make the script file active. When active, scripts within the file can be run.
SYNTAX	LOAD <script filename> Where: <script filename> - the name of the script file to LOAD and become the active script file that contains the scripts to execute. This must include the extension of the filename.
RETURNS	Prompt when ready for the next command.
VALID MODES	READY SCRIPT (when called from script)
EXAMPLE	**LOAD Script1.txt** >

RUN SCRIPT

COMMAND	RUN SCRIPT
DESCRIPTION	Executes the script contained in the script name. This will place the ECM4000 in SCRIPT mode. This continues until the script is finished, the STOP command is received, or an error is encountered when TOSTOP == 1.
SYNTAX	RUN <script name> Where: <script name> - the name of the script to run.
RETURNS	Prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**RUN Spt123** >

STOP

COMMAND	STOP
DESCRIPTION	Cancels all commands, stops a script, and returns the ECM4000 to READY mode.
SYNTAX	STOP [<NAME>} Where he optional <NAME> is an embedded script. When used, it will first attempt to send the STOP command to all enabled devices, and then stop the embedded script. Currently, only the embedded SCAN script is supported. For all user scripts, the optional argument is not supported and will not stop the devices.
RETURNS	Prompt when ready for the next command.
VALID MODES	READY SCRIPT
EXAMPLE	**STOP** >

STATUS

COMMAND	STATUS
DESCRIPTION	Displays the current mode of the ECM. - or - Displays the current mode of the ECM and the status of the configured devices.
SYNTAX	STATUS [D]
RETURNS	STATUS: <mode> <error Count> Where: <mode> - READY or SCRIPT <error count> - shows the current error count since the last CLEAR command. When the option D is used, the following information will also appear: SET DEVICE <index> <device Name> <enable status> <time-out> <connection status> Where: <Index> - assigned device number. <Device Name> - the name of the device. <enable status> - whether the device is ENABLED or DISABLED. <time-out> - whether the device is TIMED-OUT or NOT-TIMED-OUT. (*the CLEAR command will reset the time-out status*) <connection status> - whether the device is CONNECTED or DISCONNECTED via TCP.
VALID MODES	READY SCRIPT
EXAMPLE	**STATUS** *STATUS: Ready 1* *>*
EXAMPLE	*// If script "Spt123" is running with no errors* **STATUS** *STATUS: SCRIPT 0* *>*
EXAMPLE	**STATUS D** *STATUS: Ready 0* *SET DEVICE 0 DTS64 DISABLED NOT-TIMED-OUT DISCONNECTED* *SET DEVICE 1 MPS64 ENABLED NOT-TIMED-OUT CONNECTED* *SET DEVICE 2 DSA16 ENABLED TIMED-OUT CONNECTED* *>*

VERSION

COMMAND	VERSION
DESCRIPTION	Shows the current software version.
SYNTAX	VER
RETURNS	The version followed by the prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**VER** *ECM Scanivalve (c) 2020 Ver 1.00* *>*

Scanivalve

WAIT

COMMAND	WAIT
DESCRIPTION	Pauses execution for the length of time specified in the <time in seconds> field, until a configured device is ready, or until the digital input goes to a specific level.

SYNTAX	WAIT <time in seconds> [<option>]
	
	Where:
	<time in seconds> - the delay in seconds to wait.
	       Min value - 1
	       Max value - 86400
	
	[<option>] - when the option is a device name, the ECM4000 will poll that device for READY mode and use the time in seconds as a maximum wait time.
	
	when the <option> is DIN <DIN channel> <level>, the ECM will poll and wait for the digital input channel's level and use the time in seconds as a maximum wait time.

RETURNS	Nothing
VALID MODES	SCRIPT (when called from a script)
EXAMPLE	// Wait 60 seconds **WAIT 60**
EXAMPLE	// Wait until MPS1 is READY, with a 10 second timeout **WAIT 10 MPS1**
EXAMPLE	// Wait until Digital Input 2 is low, with a 20 second timeout **WAIT 20 DIN 2 0**

ERROR

COMMAND	ERROR
DESCRIPTION	Lists the errors that have occurred since the last CLEAR command. Only the first 100 errors will be listed. Any errors generated after 100 will be ignored.
SYNTAX	ERROR
RETURNS	ERROR: <error message>, [Stopping script,] <command>, <script name> : : : : ERROR: <error message>, [Stopping script,] <command>, <script name> Where: <error message> - is the error message. [Stopping script,] - this message will indicate that the script was stopped due to the error. This message will only be displayed when TOSTOP = 1. <command> - command used that resulted in an error. <script name> - script that the error was generated from. Followed by a prompt when ready.
VALID MODES	READY
EXAMPLE	**ERROR** *ERROR: Invalid command, SDIN, ScriptA* >

CLEAR

COMMAND	CLEAR
DESCRIPTION	Commands the ECM to clear all errors and timed-out device statuses.
SYNTAX	CLEAR
RETURNS	Prompt when ready for next command.
VALID MODES	READY
EXAMPLE	**CLEAR** >

TCP COMMAND OUTPUT

COMMAND	TCP OUT
DESCRIPTION	Sends a specified command to the named device via TCP/IP connection. This command will open a connection to the device if one is not open. The connection is closed when the TCPCLOSE command is executed, if a device is DISABLED, or if the REBOOT command is used.
SYNTAX	TCPOUT <name> <command to output> Where: <name> - the name of the target device as stored in the device settings. Can also be wildcard metacharacter (*) specifier that will reference all active devices in the device settings. <command to output> - the command to send.
RETURNS	Prompt when ready for the next command.
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Issue a SCAN command on MPS1 **TCPOUT MPS1 SCAN** >
EXAMPLE	// Issue a STOP command to all active devices **TCPOUT * STOP** >

OPEN TCP CONNECTION

COMMAND	TCP OPEN
DESCRIPTION	Opens the TCP connection to a specified device.
SYNTAX	TCPOPEN <name> Where: <name> - the name of the target device as stored in the device settings. The wildcard metacharacter (*) that references all enabled devices in the device list can also be used.
RETURNS	Prompt when ready for the next command.
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Opens the connection to device MPS12 **TCPOPEN MPS12** >

Scanivalve

EXAMPLE	// Open all connections to enabled devices **TCPOPEN *** >

CLOSE TCP CONNECTION

COMMAND	TCP CLOSE
DESCRIPTION	Closes the TCP connection to a specified device.
SYNTAX	TCPCLOSE <name> Where: <name> - the name of the target device as stored in the device settings. The wildcard metacha-racter (*) that references all enabled devices in the device list can also be used.
RETURNS	Prompt when ready for the next command.
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Close connection to device MPS12 **TCPCLOSE MPS12** >
EXAMPLE	// Close all connections to enabled devices **TCPCLOSE *** >

SET

COMMAND	SET
DESCRIPTION	Commands the ECM4000 to set one of the many configuration variables. Configuration variables are described in a subsequent section.
SYNTAX	SET <name> <value> Where: <name> - the name of the configuration variable to be set or modified. <value> - the value of that configuration variable.
RETURNS	Prompt when ready for the next command.
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Set the PTPEN variable to 1 **SET PTPEN 1** >
EXAMPLE	// Set the IP address variable to 191.30.65.1 **SET IPADDR 191.30.65.1** >

LIST

COMMAND	LIST
DESCRIPTION	Commands the ECM4000 to list all settings in a specified group.

SYNTAX	LIST <name> Where: <name> - the name of the configuration group to be listed.
RETURNS	LIST <name> ::::: LIST <name> Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	// List the IP variables **LIST IP** *SET IPADD 192.168.0.240* *SET SUBNET 255.0.0.0* *SET MAC 0.96.93.46.0.223* *SET GW 0.0.0.0* >

FILE DIRECTORY CONTENTS

COMMAND	DIRECTORY
DESCRIPTION	Commands the ECM4000 to list the filenames and sizes (in bytes) on the flash drive.
SYNTAX	DIR
RETURNS	<file size> <file name> ::::: <file size> <file name> Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**DIR** *219 ip.cfg* *200 Spt123.txt* >

DELETE

COMMAND	DELETE
DESCRIPTION	Commands the ECM4000 to delete a specified object.
SYNTAX	DIR
RETURNS	DELETE <event> <name> Where: <event> - is the object to delete. Valid values are FILE, DEVENT, or DEVICE. <name> - name of the object to delete. When DEVENT or DEVICE is used in the <event> field, the wildcard metacharacter (*) can be used to delete all devices.
VALID MODES	READY SCRIPT
EXAMPLE	// Delete ditigal event EvtT1 **DELETE DEVENT EvtT1** >

EXAMPLE	// Delete all devices **DELETE DEVICE *** >

SAVE

COMMAND	SAVE
DESCRIPTION	Saves all or a particular configuration to flash.
SYNTAX	SAVE [<group>] When no configuration is present, most[†] configurations are saved. When a [<group>] is present, only that group of variables is saved. The following table shows the [<group>] configuration that is valid and their associated files on flash. <table><tr><td>IP</td><td>ip.cfg</td></tr><tr><td>ID</td><td>id.cfg</td></tr><tr><td>PTP</td><td>ptp.cfg</td></tr><tr><td>DEVENT</td><td>devent.cfg</td></tr><tr><td>DEVICE</td><td>device.cfg</td></tr><tr><td>DMON</td><td>dmon.cfg</td></tr><tr><td>CALZ</td><td>calz.cfg</td></tr><tr><td>SCAN</td><td>scan.cfg</td></tr><tr><td>CONFIG</td><td>config.cfg</td></tr></table> [†]A SAVE command will not save the IP group variables. These variables must be saved using a SAVE IP.
RETURNS	Prompt when saving is complete and ready for the next command.
VALID MODES	READY
EXAMPLE	// To save all of the settings of all groups **SAVE** >
EXAMPLE	// To save only the "IP" settings to the file ip.cfg **SAVE IP** >

TYPE

COMMAND	TYPE
DESCRIPTION	Commands the ECM4000 to return the contents of the named file from flash.
SYNTAX	TYPE <filename> Where: <filename> - the file to return to the host.
RETURNS	Contents of the specified <filename> followed by a prompt when ready for the next command.
VALID MODES	READY

EXAMPLE	**TYPE IP.cfg**
	SET IPADD 192.168.0.210
	SET SUBNET 255.255.255.0
	SET MAC 0.96.93.105.0.100
	SET GW 0.0.0.0

FORMAT DISK

COMMAND	FDISK
DESCRIPTION	Erases and formats the internal memory.
	WARNING: All data will be lost, including script files and configuration files.
	Once the command is issued, the ECM will respond with a confirmation message to approve the FDISK command or to cancel.
SYNTAX	FDISK
RETURNS	Type FDISKCONFIRM to confirm FDISK or STOP to escape FDISK Formatting…
	Prompt when ready for next command.
VALID MODES	READY
EXAMPLE	**FDISK**
	Type FDISKCONFIRM to confirm FDISK or STOP to escape FDISK Formatting…
	>
NOTES	FDISK will not remove the installed firmware application.

REBOOT ECM

COMMAND	REBOOT
DESCRIPTION	Reboots the ECM (soft reboot). This will close all open TCP connections to enabled devices.
SYNTAX	REBOOT
RETURNS	Nothing
VALID MODES	READY
EXAMPLE	**REBOOT**

READ CURRENT SCRIPT

COMMAND	SCRIPT
DESCRIPTION	Displays the names of the scripts in the currently loaded script file.
	When verbose [V] option is used, the name and the content of each script will be returned.
SYNTAX	SCRIPT [V]
	Where:
	[V] will return
RETURNS	Returns the current loaded script file followed by a prompt when ready for next command.
VALID MODES	READY
EXAMPLE	**SCRIPT**
	SCPFile
	1 ScpCalz
	2 ScpPurge
	>

EXAMPLE	**SCRIPT V**
	CalzFile
	 1 ScpCalz
	   1 DOUT 1 1
	   2 DOUT 2 1
	 2 ScpPurge
	   1 DOUT 4 0
	   2 DOUT 1 1
	>

ENABLE DEVICE

COMMAND	ENABLE
DESCRIPTION	Commands the ECM to enable specified devices in the device list. This command allows the wild-card metacharacter (*) to be used to enable all devices in LIST DEVICE.
SYNTAX	ENABLE <device name> where <device name> is the name of a device in LIST DEVICE, or the wildcard metacharacter (*).
RETURNS	Prompt when ready for next command.
VALID MODES	READY
EXAMPLE	// Enable MPS1 in device list **ENABLE MPS1** >
EXAMPLE	// Enable all devices in device list **ENABLE *** >

DISABLE DEVICE

COMMAND	DISABLE
DESCRIPTION	Commands the ECM to disable specified devices in the device list. This command allows the wild-card metacharacter (*) to be used to disable all devices in LIST DEVICE. If a device is connected, this command will disconnect from the device (TCP Connection).
SYNTAX	DISABLE <device name> where <device name> is the name of a device in LIST DEVICE, or the wildcard metacharacter (*).
RETURNS	Prompt when ready for next command.
VALID MODES	READY
EXAMPLE	// Disable MPS1 in device list **DISABLE MPS1** >
EXAMPLE	// Disable all devices in device list **DISABLE *** >

CALZ (ZERO-OFFSET CORRECTION)

COMMAND	CALZ
DESCRIPTION	Commands the ECM to run the pre-programmed CALZ script based on the settings in LIST CALZ.
SYNTAX	[RUN] CALZ
RETURNS	Prompt when ready for next command.

VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Perform the CALZ routine **CALZ** >
NOTES	When a script file contains a script titled "CALZ," the internal CALZ routine will be disabled.

SCAN

COMMAND	SCAN
DESCRIPTION	Commands the ECM to run the pre-programmed SCAN script based on the settings in LIST SCAN. See the pre-programmed scripts section for more information.

SYNTAX	SCAN [<time-date string>]
	
	Where <time-date string> is optional, and is used in conjunction with PTP synchronization to start a scan at a time and date in the future. The syntax for the time-date string is:
	    YYYY/MM/DD_hh:mm:ss
	Where:
	    YYYY - year
	    MM - month
	    DD - day
	    hh - hour
	    mm - minute
	    ss - second

RETURNS	Prompt when ready for next command.
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Perform the CALZ routine **SCAN** >
NOTES	When a script file contains a script titled "SCAN," the internal SCAN routine will be disabled.
NOTES	If the ECM enters the SCAN script, a STOP command will return the ECM to ready, but not the external devices. If the external devices are also to be stopped, STOP SCAN command must be used.

BEGIN SCRIPT

COMMAND	BEGIN
DESCRIPTION	Command used in a script file to mark the beginning of a script. The command must include the user-defined script name.
SYNTAX	BEGIN <script name> where <script name> is defined by the user. Script names cannot contain spaces.
RETURNS	Nothing.
VALID MODES	N/A
EXAMPLE	// FIrst line of a Script titled "Startup" **BEGIN Startup**
NOTES	Command cannot be executed manually.

END SCRIPT

Scanivalve

COMMAND	END
DESCRIPTION	Command used in a script file to mark the end of a script.
SYNTAX	END
RETURNS	Nothing.
VALID MODES	N/A
EXAMPLE	// End the last line of a script. **END**
NOTES	Command cannot be executed manually.

DIGITAL IN/OUT COMMANDS

DIGITAL OUT

COMMAND	DIGITAL OUT
DESCRIPTION	Sets or displays the digital out configurations.

SYNTAX	DOUT <channel> <value>
	
	Where:
	<channel> - Is the digital output channel 1 through 8, the # symbol, or the ? symbol.
	
	    When a ? metacharacter is entered, the current digital output values are returned.
	
	    When a # metacharacter is entered, the value is a bit pattern of the digital outputs starting with channel 1 as the left most bit. Omitted bits are assumed to be X, and do not change.
	
	<value> - When a channel number is entered the value pertains only to that digital output channel. The options for <value> are:
	
	    0 – Off (default)
	    1 – On
	    T – Follow PHY trigger (channels 1 through 4 only - see Trigger Out)
	    X – Don't change

RETURNS	Prompt when ready for the next command. - or - Status of the digital outs in a bit pattern shows as: DOUT # <DOUT 1 value><DOUT 2 value>...<DOUT 8 value>
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Turn digital output 1 on **DOUT 1 1** >
EXAMPLE	// Turn digital output 2 off **DOUT 2 0** >
EXAMPLE	// Assign digital output 2 to follow the PHY signal **DOUT 2 T** >
EXAMPLE	// Set DOUT 1 on, 2 off, 8 on and do not change the others **DOUT # 10XXXXX1** >
EXAMPLE	// Return the current status of the digital outputs **DOUT ?** *DOUT # TT101010* >

Scanivalve

COMMAND	PRESSURE OUT
DESCRIPTION	Sets or displays the pressure output (internal solenoids) configurations.
SYNTAX	POUT <channel> <value> Where: <channel> - Is the digital output channel 1 through 3, the # symbol, or the ? symbol. When a ? metacharacter is entered, the current digital output values are returned. When a # metacharacter is entered, the value is a bit pattern of the pressure outputs starting with channel 1 as the left most bit. Omitted bits are assumed to be X and do not change. <value> - When a channel number is entered the value pertains only to that digital output channel. The options for <value> are: 0 – Off (default) 1 – On X – Don't change
RETURNS	Prompt when ready for the next command. - or - Status of the pressure solenoids in a bit pattern shows as: POUT # <POUT 1 value><POUT 2 value><POUT 3 value>
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Turn pressure solenoid 1 on **POUT 1 1** >
EXAMPLE	// Turn pressure solenoid 2 off **POUT 2 0** >
EXAMPLE	// Turn pressure solenoid 1 on, 3 off, and do not change the others **POUT # 1x0** >
EXAMPLE	// Return the current status of the pressure outputs **POUT ?** *POUT #* 110 >

DISPLAY OUT

COMMAND	DISPLAY OUT
DESCRIPTION	Sets or displays the 8 display LED settings.

SYNTAX	DISP <channel> <value>
	
	Where:
	<channel> - Is the digital output channel 1 through 8, the # symbol, or the ? symbol.
	
	    When a ? metacharacter is entered, the current digital output values are returned.
	
	    When a # metacharacter is entered, the value is a bit pattern of the digital outputs starting with channel 1 as the left most bit. Omitted bits are assumed to be X and do not change.
	
	<value> - When a channel number is entered the value pertains only to that digital output channel. The options for <value> are:
	
	    0 – Off
	    1 – On
	    = – Follow the digital output for this channel (default)
	    X – Don't change

RETURNS	Prompt when ready for the next command. - or - Status of the digital outs in a bit pattern shows as: DISP # <DISP 1 value><DISP 2 value>...<DISP 8 value>
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Turn digital output 1 on **DISP 1 1** >
EXAMPLE	// Follow the digital outputs on all channels except channel 1 **DISP # x=======** >
EXAMPLE	// Return the current status of the digital outputs **DISP?** *DISP # 1===0=11* >

Scanivalve

COMMAND	TRIGGER OUT
DESCRIPTION	Sets or displays the 4 trigger out options.

SYNTAX	TOUT <channel> <rate>
	
	Where:
	<channel> - Is the digital output channel 1 through 4, the # symbol, or the ? symbol.
	
	    When a # metacharacter is entered, the rate (in Hz) for each channel can be entered as follows:
	    TOUT # <channel 1 rate> <channel 2 rate> <channel 3 rate> <channel 4 rate>
	
	    When a ? metacharacter is entered, the current trigger rate values are returned as follows:
	    TOUT # <channel 1 rate> <channel 2 rate> <channel 3 rate> <channel 4 rate>
	
	<rate> - integer frequency that aligns on the second boundary. When 0 is entered, the output waveform is stopped (0 = default).
	    This value is range checked and can accept rate settings from 0.001 to 20000 (Hz).

RETURNS	Prompt when ready for the next command. - or - Status of the trigger outs.
VALID MODES	READY SCRIPT (when called from a script)
EXAMPLE	// Start a 10Hz trigger on channel 1 **TOUT 1 10** >
EXAMPLE	// Set a trigger rate for all 4 channels **TOUT # 1 5 10 100** >
EXAMPLE	// Display all TOUT settings **TOUT ?** *TOUT # 10 1 100 0*
NOTES	When a trigger out is enabled, and the display out is set to =, the corresponding channel's display LED will flash when the trigger out is active. The LED will flash at a fixed rate and does not flash at the configured rate setting.

Digital In

COMMAND	DIGITAL IN STATUS
DESCRIPTION	Reads the digital input channels and returns the values to the host.
SYNTAX	DIN [?]
RETURNS	DIN # <binary pattern>
VALID MODES	READY
EXAMPLE	**DIN** *DIN # 10100110* >

PTP (PRECISION TIME PROTOCOL) COMMANDS

CLEAR ACCUMULATED PTP STATISTICAL DATA

COMMAND	PTPCLEAR
DESCRIPTION	Clears the accumulated data in the histogram buffers and restarts sampling.
SYNTAX	PTPCLEAR
RETURNS	Prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**PTPCLEAR** *PTP Histogram Cleared* *>*

DISPLAY PTP STATISTICS

COMMAND	PTPSTAT
DESCRIPTION	Displays the accumulated data in histogram form.
SYNTAX	PTPSTAT
RETURNS	OFM, INTERVAL, <interval value>, <interval -6>, <interval -5>, <interval -4>, <interval -3>, <interval -2>, <interval -1>, <interval 0>, <interval 1>, <interval 2>, <interval 3>, <interval 4>, <interval 5>, <interval 6> Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**PTPSTAT** *OFM, INTERVAL, 100000, 114, 13, 10, 23, 21, 15, 36, 6, 12, 9, 23, 10, 129* *>*

GET PTP TIME

COMMAND	GETTIME
DESCRIPTION	Gets the current PTP time as used by the ECM. Time is adjusted by UTCOFFSET.
SYNTAX	GETTIME
RETURNS	Current Time yyyy/mm/dd hh:mm:ss sec nnnn ns Followed by a prompt when ready for next command.
VALID MODES	READY
EXAMPLE	**GETTIME** *Current Time 2020/2/6 0:23:52.0 sec 3335 ns 439809720* *>*

GET UTC OFFSET

COMMAND	GETUTCO
DESCRIPTION	Gets the current UTC offset.
SYNTAX	GETUTCO

RETURNS	Current UTC Offset <a> <b> <c>
	<a> - The current difference between TAI time and UTC time, in seconds, as supplied by the local grandmaster. <b> - UTC updated flag <c> - Set when UTC offset is valid. Followed by a prompt when ready for next command.
VALID MODES	READY
EXAMPLE	**GETUTCO** *Current UTC Offset 0 0 0* *>*

SET PTP TIME

COMMAND	SETTIME
DESCRIPTION	Sets the current PTP time as used by the ECM. Time is adjusted by UTCOFFSET. When a grandmaster is active on the network, the time set by the SETTIME command is over written by the grandmaster time.
SYNTAX	SETTIME yyyy/mm/dd hh:mm:ss
RETURNS	Current Time yyyy/mm/dd hh:mm:ss sec nnnn ns Followed by prompt when ready for next command.
VALID MODES	READY
EXAMPLE	**SETTIME 2019/9/22 14:20:0.00** *Entered 2019/9/22 14:20:0* *Time is 1466605200* *>*

LIST CONFIGURATIONS

This section defines the different configuration groups that are returned when using the LIST command.

LIST CONFIGURATION GROUP VARIABLES

COMMAND	List Configuration Group Variables
DESCRIPTION	Lists the CONFIG configuration variables.
SYNTAX	LIST CONFIG
RETURNS	SET DEBUG <hex value> SET PROMPT <0, 1, 2, or 3> SET AUTORUN <start up script file> <start up script name> SET NAME <name of this device> SET TOSTOP <0 or 1> Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**LIST CONFIG** *SET DEBUG 4* *SET PROMPT 1* *SET AUTORUN 0 0* *SET NAME ECM0000* *SET TOSTOP 0* *>*

LIST DEVICE GROUP VARIABLES

COMMAND	List Device Group Variables
DESCRIPTION	Lists the DEVICE configuration variables.
SYNTAX	LIST DEVICE
RETURNS	SET DEVICE <name> <IP Address:port> <type> <enable status> : : : SET DEVICE <name> <IP Address:port> <type> <enable status> Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**LIST DEVICE** *SET DEVICE DSM1 192.168.0.140:23 DSM 0* *SET DEVICE MPS2 192.168.0.150:23 MPS 1* *>*

LIST PRECISION TIME PROTOCOL GROUP VARIABLES

COMMAND	List PTP Group Variables
DESCRIPTION	Lists the PTP configuration variables.
SYNTAX	LIST PTP
RETURNS	SET PTPEN <0, 1 or 2> SET STAT <0, 1 or 2> SET UTCOFFSET <hh:mm:ss> Followed by a prompt when ready for the next command.
VALID MODES	READY

Scanivalve

EXAMPLE	**LIST PTP**
	SET PTPEN 1
	SET STAT 0
	SET UTCOFFSET 00:00:00
	>

LIST BOOT PARAMETERS GROUP VARIABLES

COMMAND	List IP Group Variables
DESCRIPTION	Lists the IP configuration variables.
SYNTAX	LIST IP
RETURNS	SET IPADD <ip address>
	SET SUBNET <subnet mask>
	SET MAC <MAC address>
	SET GW <gateway>
	Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**LIST PTP**
	SET IPADD 192.168.0.240
	SET SUBNET 255.0.0.0
	SET MAC 00:12:34:56:78:01
	SET GW 0.0.0.0
	>

LIST DIGITAL EVENT GROUP VARIABLES

COMMAND	List DEVENT Group Variables
DESCRIPTION	Lists the DEVENT configuration variables.
SYNTAX	LIST DEVENT
RETURNS	SET DEVENT <script name> <channel> <level> <sequence>
	: : :
	SET DEVENT <script name> <channel> <level> <sequence>
	Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**LIST DEVENT**
	SET DEVENT EvtD1 2 1 1
	SET DEVENT Evt5 3 0 1
	SET DEVENT CalLight 1 1 0
	>

LIST CALZ CONFIGURATION VARIABLES

COMMAND	List CALZ Group Variables
DESCRIPTION	Lists the pre-programmed CALZ configuration variables.
SYNTAX	LIST CALZ
RETURNS	SET ZCSTARTDOUT <binary pattern>
	SET ZCSTARTPOUT <binary pattern>
	SET ZCDWELL <time in seconds>
	SET ZCDEVICE <device name or *>
	SET ZCDIN <digital input number> <level>
	Followed by a prompt when ready for the next command.

VALID MODES	READY
EXAMPLE	**LIST CALZ** *SET ZCSTARTDOUT 10000000* *SET ZCSTARTPOUT 110* *SET ZCDWELL 5* *SET ZCDEVICE ** *SET ZCDIN 2 0* *>*

LIST ID CONFIGURATION VARIABLES

COMMAND	List ID Group Variables
DESCRIPTION	Lists the identification configuration variables.
SYNTAX	LIST ID
RETURNS	SET MODEL <model> SET SN <serial number> SET MCAST <address> Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**LIST ID** *SET MODEL ECM4000* *SET SN 102* *SET MCAST 224.1.1.1* *>*

LIST SCAN CONFIGURATION VARIABLES

COMMAND	List Scan Group Variables
DESCRIPTION	Lists the embedded scan configuration variables.
SYNTAX	LIST SCAN
RETURNS	SET SCANDOUT <binary pattern> SET SCANDIN <digital input number> <level> SET SCANDEVICE <device name or *> SET SCANPTP <off/on> SET SCANDELAY <seconds> SET SCANTIMEOUT <seconds> Followed by a prompt when ready for the next command.
VALID MODES	READY
EXAMPLE	**LIST SCAN** *SET SCANDOUT 10000000* *SET SCANDIN 0 0* *SET SCANDEVICE ** *SET SCANPTP 0* *SET SCANDELAY 5* *SET SCANTIMEOUT 60* *>*

Scanivalve

IP CONFIGURATION VARIABLES

The IP settings control the IP address and associated network identification settings of the ECM4000. The IP configuration is accessed by the command LIST IP.

The variables in this group are not saved when a SAVE command is issued. They may only be saved by using the SAVE IP command. Changes to these variables do not take effect until the power has been cycled. These variables are saved in the ip.cfg file.

SET IP ADDRESS OF ECM

VARIABLE	IPADD
DESCRIPTION	Sets the IPv4 address of the ECM4000. *IPv6 addresses are generated from the MAC address.*

SYNTAX	SET IPADD <ip address>
	
	    Where:
	    <ip address> is any valid IP address in the form xxx.xxx.xxx.xxx
	
	    When <ip address> is set to 0.0.0.0, DHCP and DNS is used. See "Using a Dynamic IP Address"
	      section for information on DHCP.

RETURNS	Prompt when ready to accept a new command
EXAMPLE	**SET IPADD 10.0.1.222** >
DEFAULT	191.30.66.xxx Where: xxx is the last three digits of the serial number.
RANGE	Any valid IP address

SET SUBNET OF ECM

VARIABLE	SUBNET
DESCRIPTION	Sets the subnet mask for the ECM4000. This only applies to IPv4.
SYNTAX	SET SUBNET <mask> Where: <mask> is in the form nnn.nnn.nnn.nnn
RETURNS	Prompt when ready to accept a new command
EXAMPLE	**SET SUBNET 255.255.0.0** >
DEFAULT	255.255.0.0
RANGE	Any valid subnet mask

VARIABLE	MAC
DESCRIPTION	Sets the MAC address for the ECM.
SYNTAX	SET MAC <mac address>
RETURNS	Prompt when ready to accept a new command
EXAMPLE	**SET MAC 12:34:B3:78:A4** >
DEFAULT	This value is configured by Scanivalve and may be different from unit to unit.

Set Gateway (GW)

DESCRIPTION	This is the IP address of the network gateway used for network communication. If not present 0.0.0.0 must be used.
SYNTAX	SET GW <ip address> Where <ipadd> is any valid IP address in the form xxx.xxxx.xxx.xxx
RETURNS	Prompt when ready to accept a new command
EXAMPLE	**SET GW 10.0.2.123** >
DEFAULT	0.0.0.0

Scanivalve

PRECISION TIME PROTOCOL VARIABLES (LIST PTP)

The Precision Time Protocol settings control the PTP IEEE1588 settings of the ECM4000. The PTP configuration is accessed by the command LIST PTP.

The variables in this group are saved when a SAVE command is issued. It can alternatively be saved using the SAVE PTP command. These variables are saved in the ptp.cfg file.

SET PTP ENABLE

VARIABLE	PTPEN
DESCRIPTION	Enables the precision time protocol engine in the ECM4000. When PTPEN is set to 2, the ECM will serve as a PTP master

SYNTAX	SET PTPEN <value>
	
	Where:
	  0 - disabled
	  1 - enabled as PTP slave
	  2 - enabled as PTP master

RETURNS	Prompt when ready to accept a new command
EXAMPLE	**SET PTPEN 1** >
DEFAULT	0
RANGE	0, 1, or 2
NOTES	The PTP engine must be disabled before switching states. The PTP engine cannot switch from 1 to 2 or 2 to 1 without disabling PTP first (0).

SET PTP STATISTICAL OUTPUT

VARIABLE	STAT
DESCRIPTION	Sets the destination of PTP statistical output

SYNTAX	SET STAT <level code>
	
	Where:
	    <level code> is:
	      0 – off
	      1 – serial
	      2 – network
	
	    The statistical data is output in the following format:
	    <ofm>, <msd>, <smd>, <mpd>
	    ofm - Offset From Master. Time that slave and master differ in units of microseconds.
	    msd - Master to Slave Delay. Filtered packet delivery time from master to slave.
	    smd - Slave to Master Delay. Filtered packet delivery time from slave to master.
	    mpd - Mean Path Delay. Filtered mean path delay time.

RETURNS	Prompt when ready to accept a new command
EXAMPLE	**SET STAT 0** >
DEFAULT	0
RANGE	0, 1, or 2
NOTES	The statistical output will be directed to the network connection that issued the command.

SET UTC OFFSET

VARIABLE	UTCOFFSET
DESCRIPTION	Sets the offset to be added or subtracted from network time.

SYNTAX	SET UTCOFFSET hh:mm:ss
	Where:
	    hh - Hours from -12 to 12
	    mm - Minutes from 0 to 59
	    ss – Seconds from 0 to 59

RETURNS	Prompt when ready to accept a new command
EXAMPLE	**SET UTCOFFSET -9:0:0** >
DEFAULT	0:0:0
RANGE	Any valid time

Scanivalve

GENERAL CONFIGURATION VARIABLES (LIST CONFIG)

This group contains the general configuration variables. Changes to the variables in this group take effect immediately.

The variables in this group are saved when a SAVE command is issued. It can alternatively be saved using the SAVE CONFIG command. These variables are saved in the config.cfg file.

DEBUG CONFIGURATION

VARIABLE	DEBUG
DESCRIPTION	When set, running a script produces verbose output describing each command as it is executed.

SYNTAX	SET DEBUG <hex value>
	
	Where:
	    <hex value> is:
	        0 = Network activity
	        1 = Script activity
	        2 = PTP activity

RETURNS	Prompt when ready to accept a new command.
DEFAULT	0 *(no debug)*
EXAMPLE	// Activate PTP debug output **SET DEBUG 4** >
EXAMPLE	// Activates script debug output **SET DEBUG 2** >
EXAMPLE	// Activates PTP and script debug output **SET DEBUG 6** >

RETURN PROMPT CONFIGURATION

VARIABLE	PROMPT
DESCRIPTION	Determines how the ECM responds to commands.

SYNTAX	SET PROMPT <value> [<character>]
	
	Where <value>:
	    0 = nothing
	    1 = CR
	    2 = LF
	    3 = CR-LF
	
	Where [<character>] is optional. If a character is entered in this field, the character will be sent after the value setting. Only one character can be set.

RETURNS	Prompt when ready to accept a new command
DEFAULT	0
EXAMPLE	**SET PROMPT 1 >** >
NOTES	This variable will not affect the Web Server's Terminal. The Web Server has a fixed response.

AutoRun

VARIABLE	AUTORUN
DESCRIPTION	This determines the name of the script file and the script to run at boot up. If no file is desired to be loaded, 0 is entered for the script file. If a file is desired to be loaded but not run, the file name is entered and 0 is entered for the script name.
SYNTAX	SET AUTORUN <script file name> <script name>
RETURNS	Prompt when ready to accept a new command.
DEFAULT	0
EXAMPLE	// Spt883.txt script file is loaded and the script "PowerOn" is run. **SET AUTORUN Spt883.txt PowerOn** >
EXAMPLE	// No script is loaded. **SET AUTORUN 0 0** >
EXAMPLE	// PwrUp.txt script file is loaded and no script is run. **SET AUTORUN PwrUp.txt 0** >

ECM4000 Name

VARIABLE	NAME
DESCRIPTION	This determines the name of the ECM. It is also used for access when using IPv6 protocol.
SYNTAX	SET NAME <ECM4000 name>
RETURNS	Prompt when ready to accept a new command.
DEFAULT	ECM0000
EXAMPLE	**SET NAME ECM142** >

Script Error Action

VARIABLE	TOSTOP
DESCRIPTION	This determines how the ECM handles an error while a script is running.
SYNTAX	SET TOSTOP <0 or 1> Where: 0 - the ECM will log the error and continue through the script. 1 - the ECM will stop the script and log the error.
RETURNS	Prompt when ready to accept a new command.
DEFAULT	0
EXAMPLE	**SET TOSTOP 0** >

Scanivalve

DEVICE CONFIGURATION VARIABLES (LIST DEVICE)

This group contains the device configuration variable(s). Changes to the variable(s) in this group take effect immediately.

This group associates a TCP device capable of accepting commands via TCP using an IPv4 address and a port. The device name may then be used as a reference in other commands. This does not apply to IPv6 address because the name of the device can be used directly.

The variables in this group are saved when a SAVE command is issued. It can alternatively be saved using the SAVE DEVICE command. These variables are saved in the device.cfg file.

DEVICE CONFIGURATION

VARIABLE	DEVICE
DESCRIPTION	Associates a device name with the IP address IPv4 and port, device type, and if the device is enabled or disabled. This command will also edit any configured devices when a pre-existing device name is used.

SYNTAX	SET DEVICE <device name> <IP address:port> <device type> <active>
	
	Where:
	    <device name> - the name of the device, used for reference in TCP commands. Spaces cannot be used within this field.
	
	    <IP address:port> - the IP address and port number of the specified device. Spaces cannot be used within this field.
	
	    <device type> - the type of device. Valid entries are MPS, MPSE, DSA, DTS, DSM, ENCL, SPC, RAD, and ECM. Spaces cannot be used within this field.
	
	    <active> - determines the active state of the device.
	        0 = disabled
	        1 = enabled
	        When 1 is used, the device will be active and will be referenced by any applicable wildcard (*) commands used.

RETURNS	Prompt when ready to accept a new command.
DEFAULT	Empty - no devices configured.
EXAMPLE	// Add new MPS device with IP address of 191.30.90.123 using port 23, and enabling the device. **SET DEVICE MPS123 191.30.90.123:23 MPS 1** >
EXAMPLE	// Add DSA3217PTP device with IP address of 191.30.82.99 using port 23, but not enabling the device. **SET DEVICE DSA3217PTP 191.30.82.99:23 DSA 0** >
EXAMPLE	// Change the IP address of the already configured DSA3217PTP, and enabled the device. **SET DEVICE DSA3217PTP 191.30.80.143:23 DSA 1** >

DIGITAL EVENT CONFIGURATION VARIABLES (LIST DEVENT)

This group contains the digital event configuration variables.

The variables in this group are saved when a SAVE command is issued. It can alternatively be saved using the SAVE DEVENT command. These variables are saved in the devent.cfg file.

DIGITAL EVENT CONFIGURATION

VARIABLE	DEVENT
DESCRIPTION	SET DEVENT causes the specified event to become active. When SET DEVENT is called within a script, the event does not become active until the script ends execution and the device returns to the READY state.
SYNTAX	SET DEVENT <script name> <channel> <level> <sequence> Where: <script name> - the name of this event, assigned by the user. This must match the name of the script to run when this digital event occurs. <channel> - the digital input channel to trigger the event. Valid values are 1 through 8. <level> - the digital input channel's logic level to trigger the event. Valid values are: 1 = high, 0 = low. <sequence> - the sequence number specifies what event to trigger first in the case in which the digital channel and the level are the same. Valid values are 0 or 1, where 0 specifies the event that should occur the first time the digital in is at the specified level.
RETURNS	Prompt when ready to accept a new command.
DEFAULT	Empty - no events active.
EXAMPLE	// Run script E1 when channel 2 goes high. **SET DEVENT E1 2 1** >
EXAMPLE	// Run script Purge on the first occurrence of channel 2 going low. **SET DEVENT Purge 2 0 0** >
EXAMPLE	// Start event E3 on the second occurrence of channel 2 going low. **SET DEVENT E3 2 0 1** > *// Note: In this example, the next occurrence of digital input 2 going to a 0 will run the Purge script.*
NOTES	The digital in status of all digital in's defaults high (1). When the ECM is first powered on, all digital inputs will be high (1).

CALZ CONFIGURATION VARIABLES (LIST CALZ)

This group contains the CALZ (Zero-Offset Calibration) configuration variables. The CALZ group is a predefined script for running a zero calibration on applicable devices. Changes to the variables in this group take effect immediately.

The variables in this group are saved when a SAVE command is issued. They can also be saved by using the SAVE CALZ command. These variables are saved in the calz.cfg file.

CALZ START - DIGITAL OUTS

VARIABLE	ZCSTARTDOUT
DESCRIPTION	Sets the digital output pattern when a CALZ script is run.
SYNTAX	SET ZCSTARTDOUT <binary pattern>

Where:
    <pattern> - the binary pattern of the (8) digital outputs when starting a zero-offset calibration. Where each digital out can be set as:
    1 = On
    2 = Off
    X = No change |
| RETURNS | Prompt when ready to accept a new command. |
| EXAMPLE | // Enable DOUT 3 and disable all other DOUT's when starting a CALZ.
SET ZCSTARTDOUT 00100000
> |
| DEFAULT | 0 |
| NOTES | After the CALZ Script has been performed, the ECM will return the digital outs to the last known state before the CALZ script was executed. |

CALZ START - PRESSURE OUTS

VARIABLE	ZCSTARTPOUT
DESCRIPTION	Sets the pressure output pattern when a CALZ script is run.
SYNTAX	SET ZCSTARTPOUT <pattern>

Where:
    <pattern> - the binary pattern of the (3) pressure outputs when starting a zero-offset calibration. |
| RETURNS | Prompt when ready to accept a new command. |
| EXAMPLE | // Enable POUT 1 and 2 and disable POUT 3 when starting a CALZ.
SET ZCSTARTPOUT 110
> |
| DEFAULT | 0 |
| NOTES | After the CALZ Script has been performed, the ECM will return the pneumatic outs to the last known state before the CALZ script was executed. |

CALZ Dwell

VARIABLE	ZCDWELL
DESCRIPTION	Sets the zero offset calibration dwell.
SYNTAX	SET ZCDWELL <dwell in seconds>

Where:
   <dwell in seconds> - the dwell time in seconds from the start of the CALZ sequence (post-DOUT and POUT settings), to the time when the CALZ command is issued to applicable devices (used to help stabilize devices when control pressure has just been applied). |
| RETURNS | Prompt when ready to accept a new command. |
| EXAMPLE | **SET ZCDWELL 10**
> |
| DEFAULT | 5 |

CALZ Devices

VARIABLE	ZCDEVICE
DESCRIPTION	Sets the target device(s) for the CALZ pre-programmed script.
SYNTAX	SET ZCDEVICE <name>

Where:
   <name> - the target device for the CALZ (please use the device name specified in LIST DEVICE). The wildcard metacharacter (*) may be used to target all enabled devices. |
| RETURNS | Prompt when ready to accept a new command. |
| EXAMPLE | // When a CALZ script is started, it will send the CALZ command to device MPS1 only.
SET ZCDEVICE MPS1
> |
| EXAMPLE | // When a CALZ script is started, it will send the CALZ command to all enabled devices.
SET ZCDEVICE *
> |
| DEFAULT | * (wildcard) |

CALZ Digital Input

VARIABLE	ZCDIN
DESCRIPTION	Sets the digital input channel to start the CALZ pre-programmed script.
SYNTAX	SET ZCDIN <channel> <level>

Where:
   <channel> - the digital input channel to start the CALZ script. When 0, the CALZ pre-programmed script is disabled. Valid values are 1 through 8.

   <level> - the digital level to trigger the CALZ event. Valid values are 0 or 1. |
| RETURNS | Prompt when ready to accept a new command. |
| EXAMPLE | // Start a CALZ when digital input channel 1 goes high.
SET ZCDIN 1 1
> |
| DEFAULT | 0 0 |
| NOTES | After the CALZ Script has been performed, the ECM will return the devices to the last known state before the CALZ script was executed. |

Scanivalve

IDENTIFICATION CONFIGURATION VARIABLES (LIST ID)

This group contains the ID (Identification) configuration variables. This is used to store variables that are unique to the ECM, such as the serial number and model.

The variables in this group are saved when a SAVE command is issued. They can also be saved by using the SAVE ID command. These variables are saved in the id.cfg file.

MODEL

VARIABLE	MODEL
DESCRIPTION	Sets the model name of the ECM.
SYNTAX	SET MODEL <name> Where <name> is an ascii string up to 7 characters long.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	**SET MODEL ECM4000** >
DEFAULT	ECM4000
NOTES	This variable is set at the factory and should not be changed unless required.

SERIAL NUMBER (SN)

VARIABLE	SN
DESCRIPTION	Sets the serial number of the ECM.
SYNTAX	SET SN <integer> Where <integer> is the serial number of the device.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	**SET SN 123** >
DEFAULT	100
NOTES	This variable is set at the factory and should not be changed unless required.

MULTICAST ADDRESS

VARIABLE	MCAST
DESCRIPTION	Sets the multicast address of the ECM. Primarily use to find and locate other MCAST supported modules.
SYNTAX	SET MCAST <address> Where <address> is a supported multicast address. Valid multicast addresses span from 224.0.0.0 to 239.255.255.255.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	**SET MCAST 224.1.2.3** >
DEFAULT	224.1.1.11

SCAN CONFIGURATION VARIABLES (LIST SCAN)

This group contains the scan configuration variables. The scan group is a predefined script to perform scanning on applicable devices. Changes to the variables in this group take effect immediately.

The variables in this group are saved when a SAVE command is issued. They can also be saved by using the SAVE SCAN command. These variables are saved in the scan.cfg file.

SCAN DIGITAL OUTS

VARIABLE	SCANDOUT
DESCRIPTION	Sets the digital output pattern when the embedded Scan script is run.

SYNTAX	SET SCANDOUT <binary pattern>
	
	Where:
	    <pattern> - the binary pattern of the (8) digital outputs when starting a zero-offset calibration. Where each digital out can be set as:
	    1 = On
	    2 = Off
	    X = No change

RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Enable DOUT 3, do not change DOUT 4 and 5, and disable all other DOUT's when starting a SCAN. **SET ZCSTARTDOUT 001XX000** >
DEFAULT	00000000
NOTES	After the SCAN Script has been performed, the ECM will return the digital outs to the last known state before the SCAN script was executed.

SCAN DIGITAL INPUT

VARIABLE	SCANDIN
DESCRIPTION	Sets the digital input channel to start the embedded Scan script.
SYNTAX	SET SCANDIN <channel> <level> Where: <channel> - the digital input channel to start the scan script. When 0, the scan pre-programmed script is disabled. Valid values are 1 through 8. <level> - the digital level to trigger the scan event. Valid values are 0 or 1.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Start a scan when digital input channel 1 goes high. **SET SCANDIN 1 1** >
DEFAULT	0 0

SCAN DEVICES

VARIABLE	SCANDEVICE
DESCRIPTION	Sets the target device(s) for the embedded scan script.

Scanivalve

SYNTAX	SET SCANDEVICE <name> Where: <name> - the target device to scan (please use the device name specified in LIST DEVICE). The wildcard metacharacter (*) may be used to target all enabled devices.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// When a scan script is started, it will send the scan command to device MPS1 only. **SET SCANDEVICE MPS1** >
EXAMPLE	// When a scan script is started, it will send the scan command to all enabled devices. **SET SCANDEVICE *** >
DEFAULT	* (wildcard)

Scan PTP

VARIABLE	SCANPTP
DESCRIPTION	Enables or disables the use of PTP in order to scan multiple devices synchronously. When enabled, and the scan script is executed, the ECM will retrieve its currently internal date/time, adjust the time by the SCANDELAY variable, and send the command to all compatible devices to start a scan for a specific time and date in the future.
SYNTAX	SET SCANPTP <status> Where: <status> - is 0 for disabled, or 1 for enabled.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Disable scnaning using the future scan start time/date with PTP **SET SCANPTP 0** >
DEFAULT	0

Scan Delay

VARIABLE	SCANDELAY
DESCRIPTION	This variable is only used when SCANPTP == 1. When SCANPTP is enabled, and a scan is executed, this variable will indicate the amount of seconds in the future the scan will start.
SYNTAX	SET SCANDELAY <seconds> Where: <seconds> - time in seconds. Valid values are 0 to 60.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// When using PTP, begin the scan 5 seconds after receiving the Scan command. **SET SCANDELAY 5** >
DEFAULT	5

Scan Timeout

VARIABLE	SCANTIMEOUT
DESCRIPTION	Sets the amount of seconds for the ECM to timeout during the scan script, if all devices to not return to ready.

SYNTAX	SET SCANTIMEOUT <seconds> Where: <seconds> - time in seconds. Valid values are 1 to 3600.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// To set the scan script timeout of 30 seconds **SET SCANTIMEOUT 30** >
DEFAULT	35

Device Monitor Configuration Variables and Commands (DMON)

This section contains the Device Monitor (DMON) commands and configuration variables. The DMON settings are highly configurable. One device monitor is pre-configured in the ECM4000 that is applicable for most Scanivalve systems. Changes to the variables in this group take effect immediately and can be saved.

The variables in this group are saved when a SAVE command is issued. They can also be saved by using the SAVE DMON command. These variables are saved in the dmon.cfg file.

List DMON Configuration

VARIABLE	DMON
DESCRIPTION	Lists the current DMON configuration(s).

SYNTAX	[LIST] DMON [<monitor>]
	
	Where:
	    [<monitor>] - list only the device monitor specified. If the monitor does not exist, and error will
	        be output.

RETURNS	The device monitor configuration in JSON formatted tables for all configured device monitors. This includes any configured states and patterns. The standard format for a configuration (with no user states) will be shown as: FORMAT <output> COMMAND <monitor> ENABLED <true or false> TIMEOUT <seconds> STATES NAME DOUT # - <state> <bit pattern> Followed by a prompt when ready to accept a new command.

EXAMPLE	// List the "STATUS" DMON configuration
	DMON STATUS
	FORMAT TABLE
	COMMAND STATUS
	ENABLED TRUE
	TIMEOUT 5
	STATES NAME DOUT #
	- -
	
	        READY XXXX0001
	        ERROR XXXXXX1X
	        TRANS XXXXNNXN
	>

DEFAULT	COMMAND STATUS ENABLED FALSE TIMEOUT 3 STATES NAME DOUT # - READY XXXXXXXX SCAN XXXXXXXX CALZ XXXXXXXX ERROR XXXXXXXX TRANS XXXXXXXX

DMON ENABLE

VARIABLE	DMON ENABLE
DESCRIPTION	Enable the automatic polling of devices using the configured device monitor instances.
SYNTAX	DMON ENABLE [<monitor>] Where: <monitor> - is optional, and allows for a single monitor to be enabled.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Enable all device monitors **DMON ENABLE** >
NOTES	When a monitor is enabled, the DMON configuration response will display as ENABLED TRUE.

DMON DISABLE

VARIABLE	DMON DISABLE
DESCRIPTION	Disable the automatic polling of devices using the configured device monitor instances. This will leave the digital output configuration is the last known state.
SYNTAX	DMON ENABLE [<monitor>] Where: <monitor> - is optional, and allows for a single monitor to be disabled.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Disable all device monitors **DMON DISABLE** >
NOTES	When a monitor is disabled, the DMON configuration response will display as ENABLED FALSE.

DMON ADD MONITOR

VARIABLE	DMON ADD MONITOR
DESCRIPTION	Create a new device monitor instance. Adding a device monitor will auto-create the ERROR and TRANS states for that monitor.
SYNTAX	DMON ADD <monitor> Where: <monitor> - the command used when polling remote devices (also used as the instance name). This command cannot exceed 15 characters and cannot contain spaces.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Create a device monitor for "STATUS" **DMON ADD STATUS** >
EXAMPLE	// Create a device monitor for "VALVESTATE" **DMON ADD VALVESTATE** >

DMON REMOVE MONITOR

VARIABLE	DMON REMOVE MONITOR

DESCRIPTION	Remove a device monitor instance. Removing a device monitor will remove all states and configurations for the specified monitor.
SYNTAX	DMON REMOVE <monitor>

Where:
    <monitor> - the monitor name to be removed. This command cannot exceed 15 characters and cannot contain spaces. |
| RETURNS | Prompt when ready to accept a new command. |
| EXAMPLE | // Remove the "STATUS" device monitor
DMON REMOVE STATUS
> |

DMON ADD STATE

VARIABLE	DMON ADD STATE
DESCRIPTION	Create a new state for a specified device monitor instance. The state is the expected response from the device when being polled and the desired control this response has in the ECM (digital out control only). A device monitor instance must exist for a state to be created. Duplicate states in each instance cannot be added.
SYNTAX	DMON <monitor> ADD <state> <bit pattern>

Where:
    <monitor> - the device monitor instance.
    <state> - the state name, which is the expected response from the device(s). The state is case-sensitive.
    <bit pattern> - the digital output (DOUT) bit pattern set when all enabled devices return this state. The bit pattern is the same as using the DOUT # command, where the pattern is the digital outputs starting with channel 1 as the left most bit. Each out the digital outs can be set with one of the following actions:
        0 – Off
        1 – On
        T – Follow PHY trigger (channels 1 through 4 only - see Trigger Out)
        X – Do not change state
        N – Next state |
| RETURNS | Prompt when ready to accept a new command. |
| EXAMPLE | // Create a "READY" state in the "STATUS" monitor instance, that turns on digital output 8 and turns off all other digital outputs
DMON STATUS ADD READY 00000001
> |
| EXAMPLE | // Create a "CALZ" state in the "STATUS" monitor instance, that turns on digital output 2, turns off digital outputs 1, 3, and 4, and does not change any other digital output
DMON STATUS ADD CALZ 0100XXXX
> |
| EXAMPLE | // Create a "SCAN" state in the "STATUS" monitor instance, that turns on digital output 3 using the PHY trigger, turns off digital outputs 1, 2, 4, and 8 and does not change any other digital output
DMON STATUS ADD CALZ 00T0XXX0
> |
| EXAMPLE | // Create a "PX" device monitor for a "VALVESTATE" monitor instance that does not change any of the digital out states
DMON VALVESTATE ADD PX XXXXXXXX
> |

DMON REMOVE STATE

VARIABLE	DMON REMOVE STATE
DESCRIPTION	Remove a state for a specified device monitor instance. The preconfigrued ERROR and TRANS states cannot be removed.
SYNTAX	DMON <monitor> REMOVE <state> Where: <monitor> - the device monitor instance. <state> - the state name to be removed. This is case-sensitive.
RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Remove the "READY" state from the "STATUS" device monitor instance. **DMON STATUS REMOVE READY** >

DMON MODIFY STATE BIT PATTERN

VARIABLE	DMON MODIFY STATE
DESCRIPTION	Modifies the DOUT bit pattern for an existing monitor/state configuration. This is used to alter the predefined ERROR and TRANS states, as well as any user created state.

SYNTAX	DMON <monitor> <state> <bit pattern>
	
	Where:
	    <monitor> - the device monitor instance.
	    <state> - the state name. The state is case-sensitive.
	    <bit pattern> - the digital output (DOUT) bit pattern set when all enabled devices return this state. The bit pattern is the same as using the **DOUT #** command, where the pattern is the digital outputs starting with channel 1 as the left most bit. Each out the digital outs can be set with one of the following actions:
	        0 – Off
	        1 – On
	        T – Follow PHY trigger (channels 1 through 4 only - see Trigger Out)
	        X – Do not change state
	        N – Next State

RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Change the "READY" bit pattern under the "STATUS" monitor to turn on digital output 1, turn off digital outputs 2 - 4, and do not change outputs 5 - 8. **DMON STATUS READY 1000XXXX** >
EXAMPLE	// Change the "ERROR" bit pattern under the "STATUS" monitor to turn on digital outputs 1 - 4 to flash with the PHY trigger, turn off digital outputs 5 and 6, and do not change digital outputs 7 and 8. **DMON STATUS ERROR TTTT00XX** >
NOTES	If the <bit pattern> is excluded from the command string, the ECM will output the current bit pattern for the specified state.

DMON MONITOR TIMEOUT

VARIABLE	DMON TIMEOUT

Scanivalve

DESCRIPTION	List or configure the timeout for a specified DMON monitor state transition. This controls the amount of time that the ECM will continually poll for a state before changing to the next or displaying the ERROR state if all devices are not matching.
SYNTAX	DMON <monitor> TIMEOUT [<seconds>] Where: 　　<monitor> - the device monitor instance. 　　<seconds> - the timeout value in seconds. If this optional argument is not included, the ECM 　　　will output the current setting.
RETURNS	The DMON timeout for the specified monitor and/or a prompt when ready to accept a new command.
EXAMPLE	// Set the "STATUS" timeout value to 10 seconds **DMON STATUS TIMEOUT 10** >
EXAMPLE	// Retrieve the current timeout setting for the "VALVESTATE" monitor **DMON VALVESTATE TIMEOUT** *3* >

DMON POLL

VARIABLE	DMON POLL
DESCRIPTION	One-shot poll of all or specified DMON instances that will change state depending on the response, and will output the response from each device in JSON format. The DMON instance being polled does not have to be enabled to use this one-shot poll. This command can alternatively be used to validate the state condition of a dmon instance. If a monitor and state is specified, the ECM will validate the current states of each device. If the condition entered is not met, the ECM will output an error and change the state to ERROR.
SYNTAX	DMON POLL [<monitor>] [<state>] Where: 　　<monitor> - the device monitor instance. If no instance is provided, all instances will be polled. 　　<state> - optional state argument. If a state is provided for a specified monitor, the ECM will 　　　validate whether all devices meet this state condition.

RETURNS	The current state of each enabled device in the set format. For FORMAT TABLE: DEVICE \<monitor name\> - \<device\> \<state\> :: :: :: :: \<device\> \<state\> For FORMAT JSON: [{"\<monitor name\>":[{"\<device\>":"\<state\>"}]}] Where: \<device\> - the device name. Only enabled devices will appear here. \<state\> - the current state of the device listed. If the optional state is included for a validation and the condition is not met, it will output an ERROR. Followed by a prompt when ready to accept a new command.

EXAMPLE	// Poll the current state of the devices using the "STATUS" DMON instance in TABLE format
	DMON POLL STATUS
	DEVICE STATUS
	- -
	MPS1 READY
	MPS2 READY
	MPS3 SCAN
	MPS4 ERROR
	>
	
	// This would set the digital outputs to the ERROR state.

EXAMPLE	// Poll the current state of the devices using all DMON instances in JSON format
	DMON POLL
	[{"STATUS":[{"M1":"READY"},{"M2":"READY"},{"M3":"READY"},{"M4":"READY"},{"M5":"READY"},{ "M6":"READY"},{"M7":"READY"},{"M8":"READY"}]},{"VALVESTATE":[{"M1":"PX"},{"M2":"PX"},{"M 3":"PX"},{"M4":"PX"},{"M5":"PX"},{"M6":"PX"},{"M7":"PX"},{"M8":"PX"}]}]
	>
	
	// This would set the digital outputs to the READY and PX states.

EXAMPLE	// Validate that the current state of all devices is "READY" using the "STATUS" DMON instance
	DMON POLL STATUS READY
	ERROR: STATUS/READY condition returned false, DMON POLL STATUS READY
	>
	
	// This means that all devices are not currently in "READY" mode.

EXAMPLE	// Validate that the current state of all devices is "PX" using the "VALVESTATE" DMON instance
	DMON POLL VALVESTATE PX
	>
	
	// This means that all devices are currently in "PX" mode.

Scanivalve

NOTES	The response for <state> in the poll can also include a NO_CONN response, which means the device does not have a valid connection, and is not timed-out. If the device is timed-out and not connected, it will show as an ERROR.

DMON FORMAT

VARIABLE	DMON FORMAT
DESCRIPTION	Sets the output format for DMON POLL responses (all DMON instances)

SYNTAX	DMON FORMAT <format>
	
	Where <format>:
	    TABLE = Standard Scanivalve Table format (default).
	    JSON = JSON format

RETURNS	Prompt when ready to accept a new command.
EXAMPLE	// Set format to TABLE **DMON FORMAT TABLE** >

SECTION 6: MAINTENANCE

FIRMWARE INSTALLATION (WINDOWS 7, 8, 10)

The ECM's firmware can be upload using two options: ECM Web Server via Ethernet connection, or using CodeSkin's C2Prog via Serial connection. The Web Server upload process is the recommended path unless communications with the ECM are impossible. Please contact Scanivalve for the latest ECM firmware version.

FIRMWARE INSTALLATION VIA WEB SERVER

1. Connect the ECM to the host computer via Ethernet connection.

2. Open the web browser of choice:
   Google Chrome
   Microsoft Edge
   Mozilla Firefox
   Note: Verify pop-ups are allowed in the browser.

3. Enter in the ECM's IP address into the address bar and press enter.

4. Navigate to the File Browser page on the ECM's Web Server.
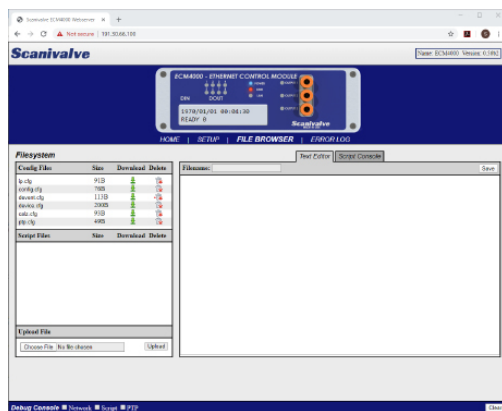
FIGURE 6-1: FILE BROWSER PAGE

5. In the "Upload File" section, click "Choose File."
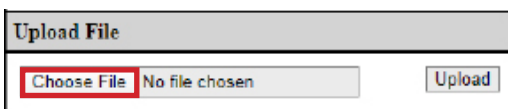
FIGURE 6-2: UPLOAD FILE

6. Navigate to the location where the ECM's firmware file is stored. The file will be called ECM4000.hex. Select the file and click "Open."
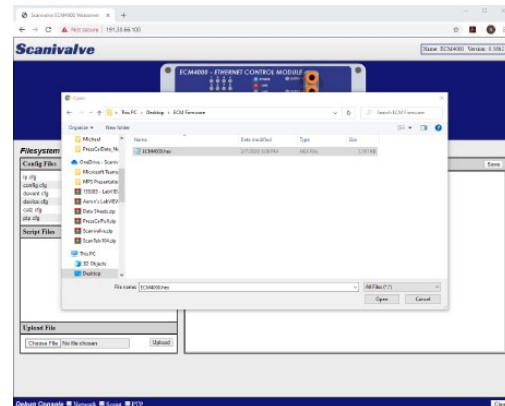
FIGURE 6-3: ECM4000.HEX

7. The ECM4000.hex file will now appear in the "Upload File" section. Click "Upload." A dialog box will appear with information.
   If there is an existing ECM4000.hex file on the ECM, the upload process will overwrite the existing file.
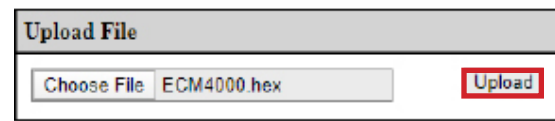
FIGURE 6-4: UPLOAD FILE

8. The ECM4000.hex file will now appear in the "Config Files" table. To initiate the upload process, double click the ECM4000.hex filename in the table.

FIGURE 6-5: CONFIG FILES

9. A dialog box will appear and ask for firmware update confirmation. Click "OK" to continue.
   A new dialog box will appear. The ECM will begin updating the firmware. When the update is complete, the dialog box will disappear and the browser will refresh and reload the ECM Home Page.
   The "Version" in the top right corner of the ECM web page should now reflect the latest version.
   Note: The ECM4000.hex file can be deleted from the File System once the update is complete. It does not need to reside in the memory for operation.

Scanivalve

the ECM's application.

In order to upload the ECM's firmware via Serial connection, the dip switches need to be placed in the following positions:

1	2	3	4
ON	OFF	ON	ON

In this state, no other communications will be possible.

FIGURE 6-10: SERIAL FIRMWARE UPLOAD MODE

UPLOADING FIRMWARE VIA SERIAL

1. Make a connection from the host computer to the Serial port on the ECM. Verify that the ECM's dip switches have been adjusted correctly for the serial firmware upload to be possible (see Figure 6-10).
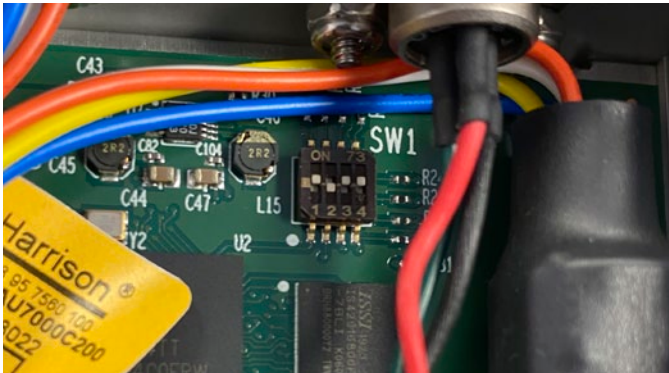
2. Reconnect the power connection to the ECM and power on.

3. In the host computer, determine the COM port that is being used to communicate with the ECM.

WARNING! Live voltages will be present in the ECM. Do not touch any internal part of the ECM until power is removed.

COM ports are typically assigned by the computer but can be manually adjusted. If you do not know the COM port being used, please use Device Manager in the host computer to determine the correct port.

4. Navigate to the installation location of the Codeskin C2Prog executable that was installed. Run this program. The C2Prog application will start and the home page will be displayed.
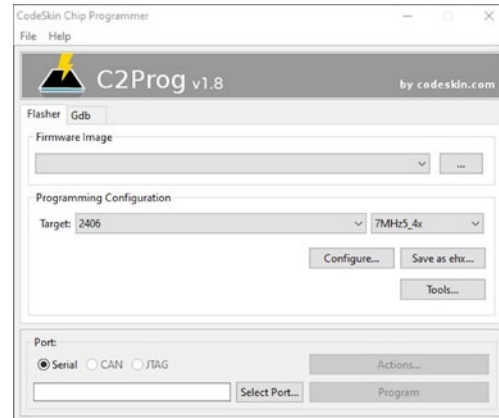
FIGURE 6-11: C2PROG HOME SCREEN

5. On the 'Flasher' tab, enter the following information into the designated fields:
   FIRMWARE IMAGE: Click the box with the ellipses (...). Navigate to the "ECM4000.hex" firmware file and select the file.
   PROGRAMMING CONFIGURATION: Click 'Target' and select "F28M36X-M3-1024K" from the drop down list.
   Click the next box over and select "20MHz" from the drop down list.
   PORT: Select the 'Serial' radio button.
   Click 'Select Port...' and choose the COM port that is connected to the ECM.
   See Figure 6-12 for an example configuration.

FIGURE 6-12: ECM SETTINGS

6. Click 'Program.'
   C2Prog will now begin programming the ECM.
   A progress bar will appear in a new window and information on the upload process will be displayed.
   The upload process may take a few minutes.
   The dialog box will notify when the programming is completed. See Figure 6-13 for a completed program.

Scanivalve

Figure 6-13: Programming Completed
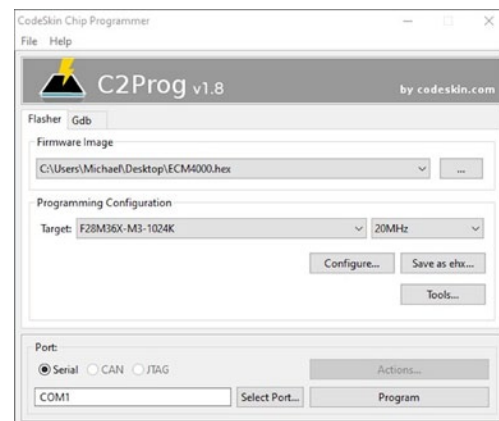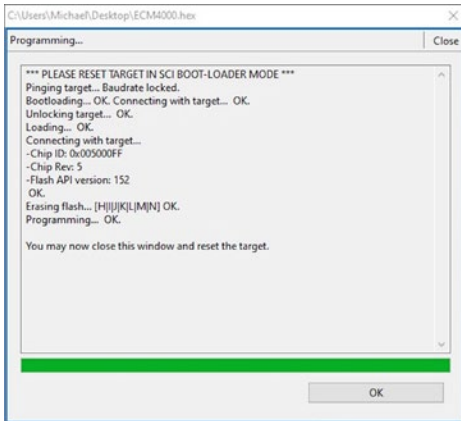
Click 'OK' and close C2Prog.

7. Power off the ECM.

8. Configure the dip switches for application run mode:

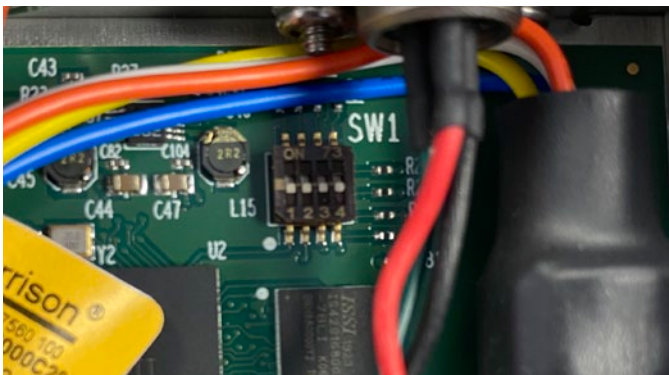| 1	2	3	4
OFF	OFF	OFF	OFF

Figure 6-14: Application Run Mode

9. Using HyperTerminal or puTTy (or similar Terminal Emulator program), open a serial connection to the ECM using the same COM port. The serial communication settings are:
    BAUD - 9600
    Data bits - 8
    Parity - None
    Stop bits - 1
    Flow Control - None

10. Power on the ECM4000. The following should display in the terminal screen:
    ECM4000
    >

11. Issue the command:
    LIST IP<CR>

12. Document the current IP address of the ECM, or alter the IP address if needed. To change the IP address, issue the command:

    SET IPADD xxx.xxx.xxx.xxx<CR>
    Where xxx.xxx.xxx.xxx is the new IP address

    Once this step is completed, issue the commands:

    SAVE IP<CR>
    STATUS<CR>

    After the ECM responds with *STATUS: READY*, power off the ECM and disconnect the power connection.

13. *(OPTIONAL) Before disconnecting the power connection, power on the ECM one more time and check LIST IP to verify that the IP address has been correctly changed.*

14. Disconnect the serial connection and reassemble the ECM using the steps provided in the "DIP Switch Access" section in reverse.

Once the ECM is fully assembled, power and Ethernet connections can be made and the ECM can be powered on. Using the web browser of choice, establish an Ethernet connection to the ECM's Web Server using Section 3 of this manual. All backup script files and configuration files may be re-uploaded to the ECM using the Web Server.

[INTENTIONALLY LEFT BLANK]

APPENDIX

APPENDIX A: PROGRAMMING AND SCRIPT EXAMPLES

The following examples are all performed using Notepad, ScanTel (or similar terminal emulator), the Web Server, and a TCP/IP connection. The integrated Web Server must be used for certain steps. In most cases, the Web Server can be used for all steps using the correct pages of the Web Server for script writing, loading, and configuration.

These examples were created to show manual configuration.

PROGRAMMING A START-UP SCRIPT

EXAMPLE REQUIREMENT: A script is required to turn on pneumatic out (POUT) 2 and turn off pneumatic outs 1 and 3 when the ECM is powered on. It must also set a 20 Hz trigger on DOUT 2 and activate the waveform output.

SOLUTION: The following is the script is created and saved to a file named Test.txt

Test.txt
BEGIN StartUp POUT # 010 TOUT 2 20 DOUT 2 T END

Note: There must be an additional blank line proceeding "END"; otherwise the "END" command will not be recognized.

This script file contains a script titled "StartUp." When this script is executed, it will perform the following:
1. Activate pneumatic output 2 and deactivate pneumatic outputs 1 and 3.
2. Set a trigger rate of 20Hz on digital output 2.
3. Activate the waveform output on digital output 2.

The script file is uploaded to the ECM4000 using the Web Server interface.
The following commands will configure the ECM4000 to run script StartUp at boot time:

SET AUTORUN Test.txt StartUp<CR>
SAVE CONFIG<CR>

The ECM4000 may be rebooted and the script will run at start up.

CREATING A LIST OF DEVICES

The following commands will create a list of devices that can be referenced by name or when the wildcard metacharacter (*) is used.

SET DEVICE DSA16 191.30.82.3:23 DSA 1<CR>
SET DEVICE DSM512 191.30.41.105:23 DSM 0<CR>
SET DEVICE MPS64 191.30.90.111:23 MPS 1<CR>

These commands have configured the following:
A DSA with an IP address of 191.30.82.3 using port 23 was added and enabled.
A DSM with an IP address of 191.30.41.105 using port 23 was added and disabled.
A MPS with an IP address of 191.30.90.111 using port 23 was added and enabled.

These configurations can be viewed by issuing a **LIST DEVICE** command.
The status of these devices can be found by issuing a **STATUS D** command.
These devices can be saved with a **SAVE DEVICE** command.

SENDING TCP COMMANDS

Once devices are configured and enabled in **LIST DEVICE**, TCP commands can be sent to enabled devices. The **TCPOUT** command is used to perform this. The examples below will be in reference to the devices configured in the example above.

// To start a SCAN of the DSA only.
TCPOUT DSA16 SCAN

// To start a SCAN of the DSA and MPS.
TCPOUT * SCAN

Note: the DSM is disabled and will not accept the command as the wildcard (*) does not apply to disabled devices.

// To set the Frames Per Scan (FPS) of the DSA to 100.
TCPOUT DSA16 SET FPS 100

Once a command has been sent to a device, the TCP connection will remain open to allow for multiple commands to be sent to the devices. Once all communications have been performed and the connection is no longer necessary, the connections can be closed using the **TCPCLOSE** command.

// Close the TCP connection to all enabled devices.
TCPCLOSE *

If there are issues when sending TCP command, please reference the Error Log and **STATUS D** for more information. Further troubleshooting can be performed using the **SET DEBUG** variable in **LIST CONFIG** and enabling Network debug information (**SET DEBUG 1**).

Scanivalve

STARTING A SCRIPT FROM A DIGITAL INPUT

EXAMPLE REQUIREMENT: A simple script is required to issue a **SCAN** command to all enabled TCP devices. The SCAN script must be executed using digital input 1. Another script is needed to send a STOP command to all enabled TCP devices. The scripts must be available after the StartUp script is run from the previous example.

SOLUTION: In Test.txt, add in new scripts after the last END:

BEGIN SCAN
TCPOUT * SCAN
END
BEGIN STOP
TCPOUT * STOP
END

Test.txt
BEGIN StartUp POUT # 010 TOUT 2 20 DOUT 2 T END BEGIN SCAN TCPOUT * SCAN END BEGIN STOP TCPOUT * STOP END

Note: Script files with multiple scripts need no additional separation between lines/scripts. The ECM uses the commands BEGIN and END to determine scripts and will find the next script using the next BEGIN and END commands. Extra spaces can be included to separate the scripts if desired. // can also be used to separate notes, comments, or descriptions. The ECM will not process any information that proceeds //.

Using the Webs Server interface, delete the existing Test.txt file and upload the newly edited file.

In order to execute the SCAN script from a digital input, a DEVENT must be created:
SET DEVENT SCAN 1 0 0

The ECM is now configured to run the SCAN script when digital input 1 goes low.

STARTING SCRIPTS FROM A DIGITAL INPUT USING SEQUENCING

A single digital input can be used to run two scripts. This may be useful when a momentary switch is intended to start and stop a scan. The DEVENT is given an additional parameter to indicate which of the two scripts are to run on the first event.

EXAMPLE REQUIREMENT: A SCAN script must be created to send a SCAN command to all devices. This script must be executed when a momentary switch is pressed. The second time the switch is pressed, a script to send a STOP command to all devices must be executed.

SOLUTION: A script file titled Scan.txt with two scripts can be made. These scripts are:

BEGIN SCAN
TCPOUT * SCAN
END
BEGIN STOP
TCPOUT * STOP
END

Scan.txt
BEGIN SCAN TCPOUT * SCAN END BEGIN STOP TCPOUT * STOP END

This file must be uploaded to the ECM using the Web Server interface. It must then be loaded into the ECM by using the LOAD command or by double-clicking the script file in the Web Server. Using manual command to load script:

LOAD Scan.txt

Two DEVENTs must be created:

SET DEVENT SCAN 1 0 0
SET DEVENT STOP 1 0 1

With this configuration, the first-time digital input 1 goes low, the SCAN script will run (sending a SCAN command to all enabled devices). The second time the digital in goes low, the STOP script will run (sending a STOP command to all enabled devices). Once these are configured, a **SAVE DEVENT** will save these configurations.

Note: The ECM's digital inputs always default to high (1). When the ECM is powered on, all digital inputs levels are high (1).

An alternate configuration could be used to trigger the SCAN script when the digital input 1 goes low, and then trigger the STOP script when the digital input returns high. This would require the DEVENT to be set as follows:

SET DEVENT SCAN 1 0 0
SET DEVENT STOP 1 1 0

If a momentary switch is used: when the switch is pressed, the SCAN script will be triggered. As soon as the switch is released, the STOP script will be triggered.
If a latching switch is used: when the switch is pressed, the scan script will be triggered. When the switch is pressed again, the STOP script will be triggered.

Scanivalve

This example will show a basic script that can be used in a system for scanning and purging. This example will be written to assume the following conditions:

1. Only DSA3217-PTP devices are being used.
2. An external pneumatic supply for PURGE SUPPLY is being controlled with an pneumatic electronic solenoid. This line is connected to all three DSA's "PRG SUPPLY" ports.
3. An LED Display panel is being used to show when the devices are in READY mode, SCAN mode, or PURGE mode.

In this example system, the following DSA modules are configured:

LIST DEVICE
SET DEVICE DSA1 191.30.82.1:23 DSA 1
SET DEVICE DSA2 191.30.82.2:23 DSA 1
SET DEVICE DSA3 191.30.82.3:23 DSA 1

The ECM's three POUTs (Pneumatic Outs) are plumbed as follows:

POUT 1 - All DSA's CTL1 ports
POUT 2 - All DSA's CTL2 ports
POUT 3 - All DSA's PRGCTL ports

A regulated 120psi supply of instrument grade air is connected to the ECM's "Supply Input."

The digital outs of the ECM are connected as follows:

DOUT 1- connected to external solenoid to control on/off of PURGE SUPPLY
DOUT 2-5 - no connection
DOUT 6 - connected to LED on external panel to indicate READY mode.
DOUT 7 - connected to LED on external panel to indicate SCAN mode.
DOUT 8 - connected to LED on external panel to indicated PURGE mode.

A script file titled "System" will be created, and the following scripts can be configured:

System.txt
BEGIN SCAN TCPOUT * SCAN DOUT # 00000010 END BEGIN STOP TCPOUT * STOP DOUT # 00000100 END BEGIN PURGE DOUT # 00000001 WAIT 5 POUT # 111 WAIT 10 DOUT 1 1 WAIT 120 DOUT 1 0 WAIT 10 POUT # 000 WAIT 5 DOUT # 00000100 END

This script file contains three scripts: SCAN, STOP, and PURGE. Each can be started with the RUN command (as no digital in events are configured).

SCAN script: The SCAN script will send a SCAN command to all enabled DSA devices. It will then enable DOUT 7 (the external SCAN LED) and disable all other DOUTs using the **DOUT #** command. The script will end.

STOP script: The STOP script will send a STOP command to all enabled DSA devices. It will then enable DOUT 6 (the external READY LED) and disable all other DOUTs using the **DOUT #** command. The script will end.

PURGE script: The PURGE script will perform an automated purge sequence to purge the DSA devices for 2 minutes. The script will perform the following:
    1. The DOUT command will enable DOUT 8 (the external PURGE LED) and disable all other DOUTs.
    2. It will then wait 5 seconds.
    3. All three POUTs will be opened to allow the 120psi control pressure to flow to each DSAs CTL1, CTL2, and PRGCTL ports.
     This will place the DSAs into "Purge" mode.
    4. It will then wait 10 seconds to allow the control pressure to stabilize.
    5. DOUT 1 will then be activated, switching an external solenoid to open and allow the purge supply to flow to the DSAs.
    6. The ECM will then wait 120 seconds to allow the DSAs input lines to be purged.
    7. After 120 seconds, DOUT 1 will be disabled, stopping the purge supply flow.
    8. It will then wait 10 seconds for the air to stabilize.
    9. All three POUTs are disabled, stopping the pneumatic control supply flow to the DSAs and allowing the DSAs to return
     to "Measurement" mode.
    10. It will then wait 5 second to stabilize.
    11. DOUT 6 (the external READY LED) will then be activated and all out DOUTs disabled.
    12. The script will end.

Scanivalve

Using Device Monitor to Indicate the Status of Devices

Device Monitor (DMON) can be used to constantly poll enabled devices and change the digital output statuses based on the device's response. This may be useful when using many Ethernet based Scanivalve devices and a panel of LED lights that can indicate the status of the external devices, and if signaling if there is an issue with one or more of the devices. DMON can be configured to poll the status of these devices in the background at all times.

EXAMPLE REQUIREMENT: DMON must be configured to poll "STATUS" from enabled devices and turn on/off LED lights on a external panel to indicate the current state of all external devices. There are 5 LEDs on the panel that indicate the following states:

READY, SCANNING, CALZ, PURGE, ERROR

SOLUTION: Configure DMON to control the digital outputs which will display the current states of enabled devices on an external LED panel.

This example will show a basic script and DMON configuration that can be used in a system for scanning, CALZ, and purging. This example will be written to assume the following conditions:
1. Only MPS4264/CPx devices are being used.
2. An external pneumatic supply for PURGE SUPPLY is being controlled with an pneumatic electronic solenoid. This pneumatic line is connected to the "PRG" ports on all MPS devices.
3. An external LED display panel is being used to show when the devices are in READY mode, SCAN mode, CALZ mode, PURGE mode, or if there is an ERROR.

In this example system, the following MPS/CPx devices are configured:

LIST DEVICE
SET DEVICE MPS1 191.30.90.1:23 MPS 1
SET DEVICE MPS2 191.30.90.2:23 MPS 1
SET DEVICE MPS3 191.30.90.3:23 MPS 1

The ECM's three POUTs (Pneumatic Outs) are plumbed as follows:

POUT 1 - All MPS's CALCTL ports
POUT 2 - All MPS's PXCTL ports
POUT 3 - No Connection

A regulated 65psi supply of instrument grade air is connected to the ECM's "Supply Input."

The digital outs of the ECM are connected as follows:

DOUT 1 - connected to external solenoid to control on/off of PURGE SUPPLY
DOUT 2 - no connection
DOUT 3 - connected to LED on external panel to indicate PURGE state.
DOUT 4 - no connection
DOUT 5 - connected to LED on external panel to indicate an ERROR state.
DOUT 6 - connected to LED on external panel to indicate READY state.
DOUT 7 - connected to LED on external panel to indicate SCAN state.
DOUT 8 - connected to LED on external panel to indicated CALZ state.

The following commands must be issued to the ECM to configure DMON for these conditions:
DMON ADD STATUS
DMON STATUS TIMEOUT 3
DMON STATUS ERROR XXXX1XXX
DMON STATUS TRANS XXXXX000
DMON STATUS ADD READY XX0X0100

DMON STATUS ADD SCAN XX0X0010
DMON STATUS ADD CALZ XX0X0001

Explanation of the DMON STATUS configurations:

    DMON ADD STATUS - the device monitor instance "STATUS" is created, which will send a STATUS command to enabled devices.

    DMON STATUS TIMEOUT 3 - uses a 3 second transitional timeout, meaning that all devices have 3 seconds to report the same states before the DMON state is updated to ERROR.

    DMON STATUS ERROR XXXX1XXX - the ERROR state for STATUS DMON is set to not change digital outputs 1-4 and 6-8. It will activate digital output 5 to light the ERROR LED.

    DMON STATUS TRANS XXXXX000 - the TRANS (transition) state for STATUS DMON is set to not change digital outputs 1-5 and turn off digital outputs 6-8. This will leave the purge and error lights in their current state and turn all other LEDs off when devices are in transition from one state to another i.e. from READY to SCAN. If there is an ERROR, the ERROR LED will not be affected by the transition state.

    DMON STATUS ADD READY XX0X0100 - a READY state is created (a response from MPS's when the STATUS command is sent) for STATUS DMON, and is set to not change digital outputs 1, 2 and 4, turn off digital outputs 3, 5, 7 and 8, and turn on digital output 6 (READY LED). When all devices respond with READY, this digital output sequence will activate only illuminating the READY LED.

    DMON STATUS ADD SCAN XX0X0010 - a SCAN state is created (a response from MPS's when the STATUS command is sent) for STATUS DMON, and is set to not change digital outputs 1, 2 and 4, turn off digital outputs 3, 5, 6 and 8, and turn on digital output 7 (SCAN LED). When all devices respond with SCAN, this digital output sequence will activate only illuminating the SCAN LED.

    DMON STATUS ADD CALZ XX0X0001 - a CALZ state is created (a response from MPS's when the STATUS command is sent) for STATUS DMON, and is set to not change digital outputs 1, 2 and 4, turn off digital outputs 3 and 5-7, and turn on digital output 8 (CALZ LED). When all devices respond with CALZ, this digital output sequence will activate only illuminating the CALZ LED.

After DMON is configured, and all devices are enabled and powered on, DMON can be enabled using the command:
DMON ENABLE STATUS. The device monitor will begin to run and poll STATUS for all devices. If devices are not connected, DMON will open any TCP connections to enabled devices.

A script file titled "MPSSystem" will be created with the following scripts:

MPSSystem.txt

```
BEGIN SCAN
TCPOUT * SCAN
END

BEGIN STOP
TCPOUT * STOP
END

BEGIN CALZ
POUT # 100
WAIT 5
TCPOUT * CALZ
WAIT 30 *
POUT # 010
WAIT 5
END
```

```
BEGIN PURGE
DMON DISABLE
DOUT # 00100000
POUT # 100
WAIT 5
DOUT # 10100000
WAIT 60
DOUT # 00100000
WAIT 5
POUT # 010
WAIT 5
DOUT # 00000000
DMON ENABLE STATUS
END
```

Scanivalve

This script file contains four scripts: SCAN, STOP, CALZ, and PURGE. Each can be started with the RUN command (as no digital in events are configured). *Note: These scripts are in the same file, but shown in columns to save space.*

SCAN script: The SCAN script will send a SCAN command to all enabled MPS devices. The script will end.

STOP script: The STOP script will send a STOP command to all enabled MPS devices. The script will end.

CALZ script: The CALZ script will perform an automated zero-calibration sequence to CALZ the MPS devices. The script will perform the following:
1. The POUT command will enable control pressure to flow out Output 1, placing the devices into CAL mode.
2. It will then wait 5 seconds for control pressure and the valves to stabilize.
3. A "CALZ" command is send to all enabled devices to start the CALZ.
4. The WAIT command is used to wait for all enabled devices to respond with "STATUS: READY" with a timeout of 30 seconds. Once either of these conditions is met, the script will continue.
5. The POUT command will enable control pressure to flow out Output 2 and turn off the other outputs, placing the devices into PX mode.
6. The ECM will then wait 5 seconds for control pressure to stabilize and then END the script.

PURGE script: The PURGE script will perform an automated purge sequence to purge the MPS devices for 1 minute. The script will perform the following:
1. Disable DMON. There is no status in MPS devices that reads PURGE, so the state does not exist and is not configured in DMON. To prevent the constant update of DMON, it is disabled for this script.
2. The DOUT command is used to turn off all digital outputs and turn on digital output 3 (PURGE LED).
3. The POUT command is used to turn on pneumatic output 1 (to place MPS devices into CAL/PRG mode), and turn off all other pneumatic outputs.
4. It will then wait 5 seconds to allow the control pressure to stabilize.
5. The DOUT command is used to activate digital outputs 1 and 3, switching an external solenoid to open and allow the purge supply to flow to the MPSs, along with leaving the PURGE LED lit.
6. The ECM will then wait 60 seconds to allow the MPSs input lines to be purged.
7. The DOUT command is then used to disabled digital output 1, stopping the purge supply flow.
8. It will then wait 5 seconds for the air to stabilize.
9. The POUT command is used to turn on pneumatic output 2 (to place MPS devices into PX mode), and turn off all other pneumatic outputs.
10. It will then wait 5 second to stabilize.
11. The DOUT command is used to turn off all digital outputs (to show a transitional state).
12. DMON ENABLE STATUS is used to enable this DMON monitor. It will begin the polling again and activate the current state of the devices, updating the digital output sequence.

With DMON enabled, the corresponding state of all devices should activate the correct LED on the external panel to indicate the status of the devices. When a script is run, SCAN for example and all devices enter into SCAN mode, DMON will retrieve the SCAN response and change the DMON state to show SCAN mode.

The PURGE script is unique as there is no STATUS response for a PURGE mode in the MPS, and the digital output states must be controlled manually - this is why DMON is disabled at the beginning of the script. However, at the end of the script, DMON is enabled to update the state of the devices. Enabling and Disabling DMON in scripts can be used for manually controlling digital outputs when needed.

USING DEVICE MONITOR TO STOP A SCRIPT

Device Monitor (DMON) can be used to perform a "one-shot" poll (**DMON POLL**) when device monitor is enabled or disabled. One feature of the "one-shot" poll is the ability to validate the state of a specified monitor/state. in conjunction with the **TOSTOP** variable under **LIST CONFIG**, this command can be used during a script to validate a state before continuing the script. If an error is generated due to a validation poll not being met, the ECM will generate and error which can stop the script for a user to investigate. This can potentially save equipment from being damaged if something is improperly configured.

EXAMPLE REQUIREMENT: In the previous PURGE script in the MPSSytem.txt script file, the valvestate of the MPS must be validated to ensure the devices are in the correct state before and after activating the purge supply solenoid.

SOLUTION: Create another DMON monitor instance configuration for VALVESTATE that can be used to validate the valvestate of MPS devices using the "one-shot" validation poll.

Using all of the same configurations and scripts as the previous section's example "*USING DEVICE MONITOR TO INDICATE THE STATUS OF DEVICES*", configure a new device monitor for VALVESTATE using the following commands:

DMON ADD VALVESTATE
DMON VALVESTATE TIMEOUT 3
DMON VALVESTATE ERROR 00000010
DMON VALVESTATE TRANS XXXXXXXX
DMON VALVESTATE ADD CAL XXXXXXXX
DMON VALVESTATE ADD PX XXXXXXXX

Explanation of the DMON VALVESTATE settings:

  DMON ADD VALVESTATE - a device monitor called "VALVESTATE" is created, which will send a VALVESTATE command to enabled devices.
  DMON STATUS TIMEOUT 3 - provides a 3 second timeout, meaning that all devices have 3 seconds to report the same states before the DMON state is moved to ERROR.
  DMON STATUS ERROR 00001000- the ERROR state for this DMON is set to turn off all digital outputs except for digital output 5 (ERROR LED). If the VALVESTATE monitor is ERROR, it will turn off all potentially connected digital outputs and signal an ERROR.
  DMON STATUS TRANS XXXXXXXX - the TRANS (transition) state for this DMON is set to not change any digital outputs because the transition notification or change is not required.
  DMON STATUS ADD CAL XXXXXXXX - a CAL state is created (a response from MPSs when the VALVESTATE command is sent) for this DMON, and is set to not change any of the digital outputs if all devices are in this state. This is only used for validation, and does not need to change any external digital outputs.
  DMON STATUS ADD PX XXXXXXXX -a PX state is created (a response from MPSs when the VALVESTATE command is sent) for this DMON, and is set to not change any of the digital outputs if all devices are in this state. This is only used for validation, and does not need to change any external digital outputs.

This monitor will not be enabled as it will only be used to validate the valvestate of the MPSs during a script. The "one-shot" poll command can be used to show the current state of each enabled and connected MPS:

DMON POLL VALVESTATE

This will display the current state of each enabled device.

To use the "one-shot" validation poll in the PURGE script, the script must be edited to run as follows (changes are shown in **BOLD**):

Scanivalve

MPSSystem.txt
BEGIN PURGE DMON DISABLE **SET TOSTOP 1** DOUT # 00100000 POUT # 100 WAIT 5 **DMON POLL VALVESTATE CAL** DOUT # 10100000 WAIT 60 DOUT # 00100000 WAIT 5 POUT # 010 WAIT 5 **DMON POLL VALVESTATE PX** DOUT # 00000000 **SET TOSTOP 0** DMON ENABLE STATUS END

Modified PURGE script: The PURGE script will perform an automated purge sequence to purge the MPS devices for 1 minute. The script will also validate whether the MPS devices are in the correct valvestate before continuing with the script. The script will perform the following:

1. Disable DMON. There is no status in MPS devices that reads PURGE, so the state does not exist and is not configured in DMON. To prevent the constant update of DMON, it is disabled for this script. The "one-shot" validation poll can be run when DMON is disabled.
2. SET TOSTOP 1 is issued to change the error action behavior of the ECM. If an ERROR is generated, the SCRIPT will terminate when the ERROR is generated (used for the validation).
3. The DOUT command is used to turn off all digital outputs and turn on digital output 3 (PURGE LED).
4. The POUT command is used to turn on pneumatic output 1 (to place MPS devices into CAL/PRG mode), and turn off all other pneumatic outputs.
5. It will then wait 5 seconds to allow the control pressure to stabilize.
6. DMON POLL VAVLESTATE CAL command is used to validate the states of the MPS devices. The DMON monitor will send a VALVESTATE command to all enabled devices and validate whether all devices read CAL or not.
   If one or more devices does not respond with CAL, the condition is not met and will generate an ERROR (which will stop the script and change the digital output configuration to the DMON VALVESTATE ERROR pattern). This prevents the next line in the script (applying the Purge Supply pressure) from being applied to devices that are not in the correct valvestate.
   If all devices respond with CAL, the condition is met and the script will continue...
7. The DOUT command is used to activate digital outputs 1 and 3, switching an external solenoid to open and allow the purge supply to flow to the MPSs, along with leaving the PURGE LED lit.
8. The ECM will then wait 60 seconds to allow the MPSs input lines to be purged.
9. The DOUT command is then used to disabled digital output 1, stopping the purge supply flow.
10. It will then wait 5 seconds for the air to stabilize.
11. The POUT command is used to turn on pneumatic output 2 (to place MPS devices into PX mode), and turn off all other pneumatic outputs.
12. It will then wait 5 second to stabilize.
13. DMON POLL VAVLESTATE PX command is used to validate the states of the MPS devices again. The DMON monitor will send a VALVESTATE command to all enabled devices and validate whether all devices read PX or not.
   If one or more devices does not respond with PX, the condition is not met and will generate an ERROR (which will stop the script and change the digital output configuration to the DMON VALVESTATE ERROR pattern). If all devices respond with CPX, the condition is met and the script will continue...

14. The DOUT command is used to turn off all digital outputs (to show a transitional state).
15. SET TOSTOP is returned back to 0 (scripts will not stop on an error).
16. DMON ENABLE STATUS is used to enable this DMON monitor. It will begin the polling again and activate the current state of the devices, updating the digital output sequence.
17. The script will end and return the ECM to ready mode.

Using **TOSTOP** and **DMON POLL** in a script can help prevent scripts from performing duties that may cause accidental damage to devices, or used to stop scripts if all devices are not in the same state.

APPENDIX B - TCP ERROR TABLE

Error Code Number (=)	Error Short	Description
4	EINTR	Interrupted system call
6	ENXIO	Device no configured
9	EBADF	Bad file descriptor
12	ENOMEM	Cannot allocate memory
13	EACCES	Permission denied
22	EINVAL	Invalid Argument
24	EMFILE	Too many open files
32	EPIPE	Broken pipe
35	EWOULDBLOCK	Operation would block
	EAGAIN	Try Again
36	EINPROGRESS	Operation in progress
37	EALREADY	Operation already in progress
38	ENOTSOCK	Socket operation on non-socket
40	EMGSIZE	Message too long
41	EPROTOTYPE	Protocol wrong type for socket
42	ENOPROTOOPT	Protocol not available
44	ESOCKTNOSUPPORT	Socket type not supported
45	EOPNOTSUPP	Operation no supported
46	EPFNOSUPPORT	Protocol family no supported
48	EADDRINUSE	Address already in use
53	ECONNABORTED	Software caused connection abort
54	ECONNRESET	Connection reset by peer
55	ENOBUFS	No buffer space available
56	EISCONN	Socket is already connected
57	ENOTCONN	Socket is not connected
58	ESHUTDOWN	Can't send after socket shutdown
60	ETIMEDOUT	Operation timed out
61	ECONNREFUSED	Connection refused
64	EHOSTDOWN	Host is down
65	EHOSTUNREACH	No route to host
-1	EAI_BADFLAGS	Invalid value for 'ai_flags' field
-2	EAI_NONAME	Name or service is unknown
-3	EAI_AGAIN	Temporary failure in name resolution
-4	EAI_FAIL	Non-recoverable failure in name resolution
-5	EAI_NODATA	No address associated with name
-6	EAI_FAMILY	'ai_family' not supported
-7	EAI_SOCKTYPE	'ai_socktype' not supported
-8	EAI_SERVICE	Service not supported for 'ai_socktype'
-9	EAI_ADDRFAMILY	Address family for name not supported
-10	EAI_MEMORY	Memory allocation failure
-11	EAI_SYSTEM	System error returned in 'errno'
-12	EAI_OVERFLOW	Argument buffer overflow

Appendix C - Software Revision Log

Version 1.00 - Initial Release (April 2020)

Version 1.01 - Added Device Monitor (DMON) functions. Added the TCPOPEN command. Removed the automatic TCPCLOSE function when a script is ended. Added TCPOPEN and TCPCLOSE buttons to the Home page on the web server. Enabled the automatic close of TCP connections to external devices when a device is disabled or if a reboot command is issued. Minor bug fixes (April 2021).

Version 1.02 - Support DSA-PTP binary status packet in DMON. Modified default internal date to 2015/1/1. (June 2022)

Version 1.03 - Added LIST ID and associated variables. Added embedded Scan script and associated variables. Add features associated with 70565. (February 2024)

Scanivalve

[INTENTIONALLY LEFT BLANK]

Scanivalve

1722 N. Madson St.
Liberty Lake, WA 99019

Phone: 1-800-935-5151
1-509-891-9970
Fax: 1-509-891-9481

scanco@scanivalve.com
www.scanivalve.com