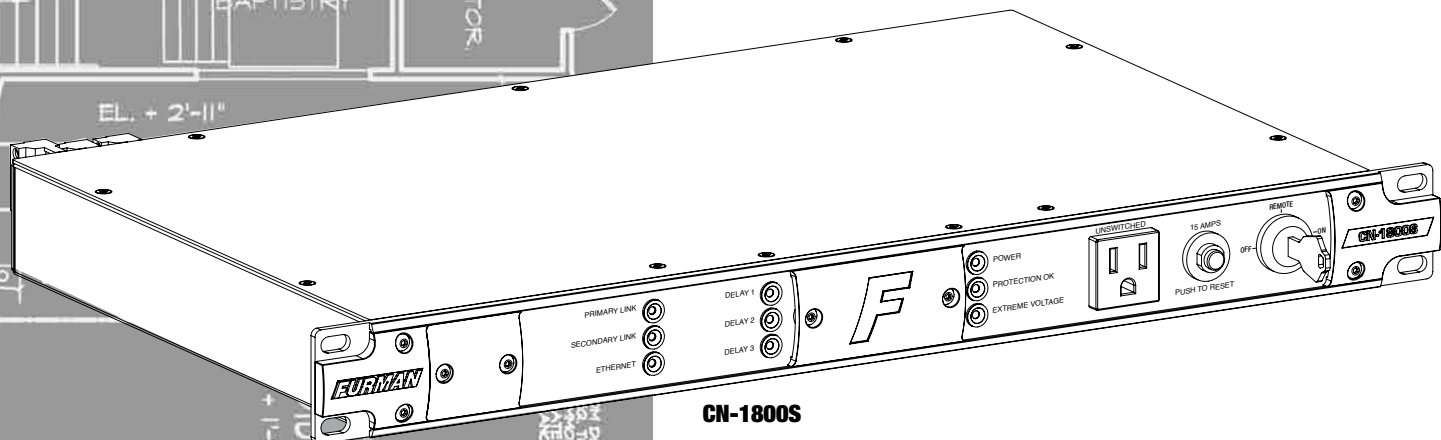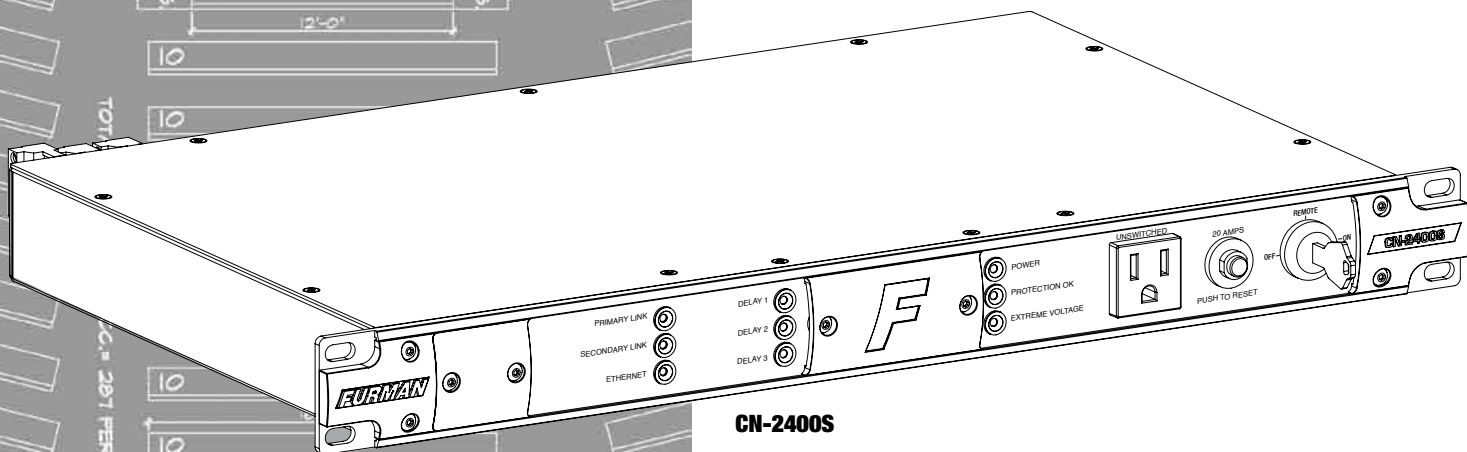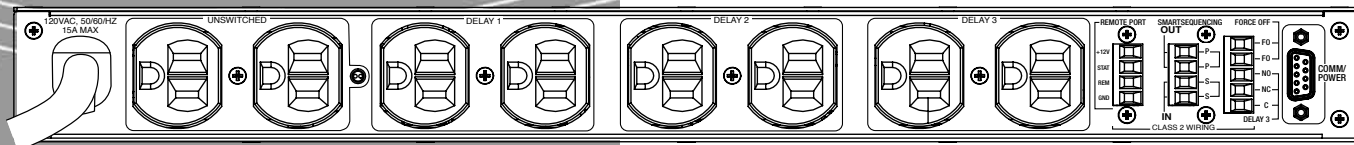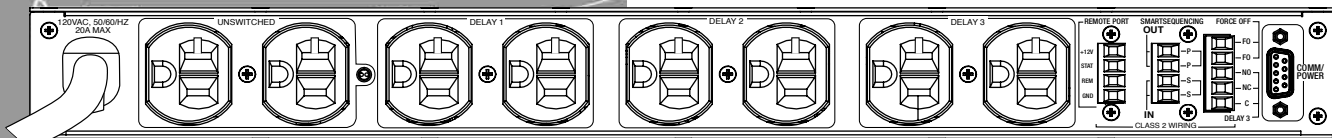# CN-1800S and CN-2400S
Command Set /
Protocol Specification

CN-1800S

CN-2400S

FURMAN®

0. OVERVIEW

The purpose of this document is to outline the command set used to communicate with and control the Furman Contractor Series of products. The data communication feature will most often be used to interface with automation systems. It may also be used to control and configure the unit manually using a standard terminal emulation program (i.e. Windows Hyper-Terminal or Hercules)

1. MESSAGE CONSTRUCTS

All messages are in the form of ASCII character strings that start with a symbol (!,?,$) to indicate the type of message and are terminated with a carriage return <CR> ASCII character 0Dh (hex), 13d (decimal).

1.1   Incoming messages (to the Primary Sequencer) must be terminated with one of the following characters: NUL (00h, 00d), carriage return (<CR>, 0Dh, 13d) or line feed (<LF>, 0Ah, 10d).

     1.2   The Primary Sequencer will discard the incoming message under the following conditions:

     1.2.1   The message overruns the receiver buffer (32 characters).

     1.2.2   No terminating character (NUL, <CR>, <LF>) is received within 500ms of receipt of the last character.

1.3   There are three data types transmitted between the Contractor and the controller hardware:

     1.3.1   **COMMAND**: A message sent to the Primary Sequencer **from** the controller requesting a specific action by the Primary Sequencer. Command strings start with an exclamation point (!, 21h, 33d). Some commands require a parameter to be included in the message. The parameter must be separated from the command with a space character (<SP>, 20h, 32d). In the command descriptions, parameters will be described in ***bold italic*** font.

     1.3.2   **QUERY:** A message sent to the Primary Sequencer from the controller requesting a status message to be returned. Query strings start with a question mark (?, 3Fh, 63d). Some queries require a parameter to be included in the message. The parameter must be separated from the command with a space character (<SP>, 20h, 32d). In the query descriptions, parameters will be described in ***bold italic*** font.

     1.3.3   **RESPONSE:** A message sent from the Primary Sequencer to the controller indicating the current status. Response strings start with a dollar sign ($, 24h, 36d).

     1.3.4   **EVENT:** A message sent **from** Sequencer indicating a **change** has occurred. Response strings start with an asterisk (*, 2Ah, 42d). Some events require a parameter to be included in the message. The parameter must be separated from the command with a space character (<SP>, 20h, 32d).

2. RS232 CONFIGURATION

2.1   Connector Type: DE-9 Female

2.2   Connector Pin-out: Pin 1, Accessory power (-12VDC)

     Pin 2, TXD. CONTRACTOR transmits data on this pin.

     Pin 3, RXD. CONTRACTOR receives data on this pin

     Pin 4, Accessory power only.

     Pin 5, SG (Signal ground).

     Pin 6, Accessory power only.

     Pin 7, RTS. Request to send.

     Pin 8, CTS. Clear to send

     Pin 9, Accessory power (-12VDC)

2.3   Comm. System: Asynchronous, half-duplex

2.4   Baud Rate: 19,200bps

2.5   Start Bits: 1

2.6   Data Bits: 8

2.7   Stop Bits: 1

2.8   Parity: None

2.9   Flow Control: None

2.10   Cable: DE-9 Straight Male to Female (1:1, 2:2, 3:3 etc.)

3. CONTROLLER COMMANDS

The following are commands sent by the controlling equipment to the Primary Sequencer.

Note: The Primary Sequencer must be configured in Primary mode (DIP Switch #8 ON) and the front panel key switch must be in the REMOTE position for communications to be enabled.

3.0   SEQUENCE ON

Activates DELAY 1, DELAY 2, and DELAY 3 outlets in sequence and according to the time delay set by the front panel DIP switches and the DLY ADJ pot setting. If the ***sequencer*** argument is omitted, the command will be executed by the Primary Sequencer and then propagated to all connected Secondary Sequencers. This will cause the entire chain of sequencers to sequence on in the order in which they are connected to the Primary Sequencer.

3.0.1   Command: !SEQ_ON ***sequencer*** <CR>

     sequencer = {0..n}

     Example: !SEQ_ON<CR> (Sequences on all units)

     Example: !SEQ_ON 2<CR> (Sequences on all outlets on sequencer 2)

3.0.2   Action: Activates DELAY 1, DELAY 2, and DELAY 3 outlets in sequence. Has the same effect as if someone pressed the START SEQUENCE button.

3.0.3   Response: !SEQ_ON<CR>: $SEQ_ON_DONE<CR>

     !SEQ_ON ***sequencer*** <CR>:
     $ACK ***sequencer***, SEQ_ON_DONE<CR>

3.1   SEQUENCE OFF

Deactivates DELAY 1, DELAY 2, and DELAY 3 outlets in reverse order and according to the time delay set by the front panel DIP switches and the DLY ADJ pot setting. If the sequencer argument is omitted, the command will be executed by the Primary Sequencer and then propagated to all connected Secondary Sequencers. This will cause the entire chain of sequencers to sequence off starting with Secondary Sequencer furthest away from Primary Sequencer.

3.1.1   Command: !SEQ_OFF ***sequencer*** <CR>

     sequencer = {0..n}

     Example: !SEQ_OFF<CR> (Sequences off all units connected to Primary)

     Example: !SEQ_OFF 1<CR> (Sequences off all outlets on sequencer 1)

FURMAN

3.1.2	Action:	Deactivates DELAY 1, DELAY 2, and DELAY 3 outlets in reverse order. Has the same effect as if someone pressed the START SEQUENCE button.

3.1.3   Response: !SEQ_OFF<CR>: $SEQ_OFF_DONE<CR>
!SEQ_OFF**sequencer** <CR>:
   $ACK **sequencer** ,SEQ_OFF_DONE<CR>

3.2   ENUMERATE

Provides an enumerated list of all sequencers connected to the Primary Sequencer and assigns unit numbers to all Secondary Sequencers. This command also identifies the Last Secondary Sequencer within a chain of connected sequencers by appending the text "LAST" to the last sequencer. This command may take several seconds to complete depending on the number of Secondary Sequencers that are connected to the Primary Sequencer.

3.2.1   Command: !ENUMERATE<CR>

Example: !ENUMERATE<CR> (Starts the enumeration process)

3.2.2   Action: Provides an enumerated list of all sequencers connected to the Primary Sequencer.

3.2.3   Response: For an individual sequencer: $ACK 0,CN-1800S,LAST <CR>

For a group of n sequencers: $ACK 0,CN-1800S<CR>

$ACK 1,CN-2400S<CR>

$ACK n,CN-1800S,LAST <CR>

3.3   BANK ON

Immediately activates the DELAY output number **outlet** on the sequencer specified by **sequencer**.

3.3.1   Command: !BANK_ON **sequencer outlet** <CR>

outlet = {1..3}

sequencer= {0..n}

Example: !BANK_ON 0 2<CR> (turns on outlet 2 on sequencer 0)

3.3.2   Action: Specified **outlet** is activated on specified **sequencer.**

3.3.3   Response: $ACK **sequencer**, BANK_ON **outlet** =ON<CR>

3.4   BANK OFF
Immediately deactivates the DELAY output number **outlet** on the sequencer specified by **sequencer**

3.4.1   Command: !BANK_OFF **sequencer outlet**<CR>

outlet = {1..3}

sequencer= {0..n}

Example: BANK_OFF 0 2<CR> (turns on outlet 2 on sequencer 0)

3.4.2   Action: Specified **outlet** is deactivated on specified **sequencer**.

3.4.3   Response: $ACK **sequencer**, BANK_OFF **outlet** =ON<CR>

3.4   ALL OFF
Immediately turns off all of the DELAY outlets on the sequencer specified by **sequencer**. If the **sequencer** argument is omitted, the command will be executed by the Primary Sequencer.

3.4.1   Command: !ALL_OFF **sequencer**<CR>

sequencer = {0..n}

Example: !ALL_OFF 2<CR> (turns off all DELAY outlets on sequencer 2)

3.4.2   Action: All DELAY outlets on **sequencer** will be deactivated.

3.4.3   Response: $ACK **sequencer**, ALL_OFF

3.5   EVENTS ENABLE
Enables the reporting of events and warnings on the sequencer specified by **sequencer**. If the argument **sequencer** is omitted the command will be executed by the Primary Sequencer.

3.5.1   Command: !EVENTS_ENABLE **sequencer** <CR>

sequencer= {0..n}

Example: !EVENTS_ENABLE 9<CR> (Enables Events on sequencer 9)

3.5.2   Action: Events are reported.

3.5.3   Response: $ACK **sequencer**, EVENTS_ENABLE<CR>

3.6   EVENTS DISABLE
Disables the reporting of events and warnings on the sequencer specified by **sequencer**. If **sequencer** is omitted the command will be executed by the Primary Sequencer.

3.6.1   Command: !EVENTS_DISABLE **sequencer** <CR>

sequencer = {0..n}

Example: !EVENTS_DISABLE 7<CR> (Disables Events on sequencer 7)

3.6.2   Action: Events are no longer reported.

3.6.3   Response: $ACK **sequencer**, EVENTS_DISABLE<CR>

3.7   RESET
Resets the sequencer specified by sequencer. If **sequencer** is omitted the RESET command will be executed by the Primary Sequencer. It should be noted that if event reporting is enabled, the Primary Sequencer will issue a "SEQ_CHANGE" error message as it will temporarily be unable to communicate with the Secondary Sequencer that is undergoing reset. This is a temporary condition.

3.7.1   Command: !RESET **sequencer**<CR>

sequencer = {0..n}

Example: !RESET 79<CR> (Resets sequencer number 79)

3.7.2   Action: Specified unit will perform a power-on reset
Delay outputs will be deactivated until reactivated
Primary unit will issue a SEQ_CHANGE error as the unit undergoes reset
The SmartLink LEDs will blink red for a few moments

3.7.3   Response: $ACK **sequencer**, RESET<CR>

3.8 CLEAR_PER

Clears the communications log percentage on the sequencer specified by **sequencer**. If the **sequencer** argument is omitted the command will be executed by the Primary Sequencer. The log can be viewed by invoking the ?STATUS command.

3.8.1   Command: !CLEAR_PER **sequencer** <CR>

　　　　　　　　　　sequencer = {0..n}

3.8.2   Action: Specified unit will clear out its parameters.

3.8.3   Response: $ACK **sequencer**, CLEAR_PER<CR>

---

4.   QUERIES - Responses end with <EOT>

4.0   VOLTAGE

Requests the AC voltage (in Volts) measurement on the sequencer specified by **sequencer**. The **sequencer** number must be provided.

4.0.1   Query: ?VOLTAGE **sequencer** <CR>

　　　　　　　　　　sequencer = {0..n}

　　　　　　　　　　Example: ?VOLTAGE 0<CR> (get voltage reading from sequencer 0)

4.0.2   Response: $ACK **sequencer** VOLTAGE=118.3
　　　　　　　　　　<EOT>

4.1   CURRENT

Requests the AC current (in Amps) measurement on the sequencer specified by **sequencer**. The **sequencer** number must be provided.

4.1.1   Query: ?CURRENT **sequencer** <CR>

　　　　　　　　　　sequencer = {0..n}

　　　　　　　　　　Example: ?CURRENT 2<CR> (get current reading from sequencer 2)

4.1.2   Response: $ACK **sequencer**, CURRENT=2.4
　　　　　　　　　　<EOT>

4.2   POWER

Requests the RMS power (in Watts) measurement on the sequencer specified by **sequencer**. The **sequencer** number must be provided.

4.2.1   Query: ?POWER **sequencer** <CR>

　　　　　　　　　　sequencer = {0..n}

　　　　　　　　　　Example: ?POWER 23<CR> (get RMS power reading from sequencer 23)

4.2.2   Response: $ACK **sequencer**, WATTS=1325
　　　　　　　　　　<EOT>

4.3   POWER_VA

Requests the VA Power measurement on the sequencer specified by **sequencer**. The **sequencer** number must be provided.

4.3.1   Query: ?POWER_VA **sequencer** <CR>

　　　　　　　　　　sequencer = {0..n}

　　　　　　　　　　Example: ?POWER_VA 5<CR> (get VA power reading from sequencer 5)

4.3.2   Response: $ACK **sequencer**, VA=125
　　　　　　　　　　<EOT>

4.4   POWER FACTOR

Requests the Power Factor measurement on the sequencer specified by the **sequencer** argument. The **sequencer** number must be provided.

4.4.1   Query: ?PF **sequencer** <CR>

　　　　　　　　　　sequencer = {0..n}

　　　　　　　　　　Example: ?PF 17<CR> (get VA power reading from sequencer 17)

4.4.2   Response: $ACK **sequencer**, PF=0.91
　　　　　　　　　　<EOT>

4.5   ID

Requests the device identification of the sequencer specified by **sequencer**.

4.5.1   Query: ?ID **sequencer** <CR>

　　　　　　　　　　sequencer = {0..n}

　　　　　　　　　　Example: ?ID 2<CR> (get ID data from sequencer 2)

4.5.2   Response: $ACK **sequencer**, ID

FURMAN	-> Brand
CN-1800S	-> Model
12345678901234	-> Serial number
1.1.1	-> Firmware revision
Fri Mar 16 10:47:00 2012	-> Rev Date
<EOT>	

4.6   SETTINGS

Requests the settings of the sequencer specified by **sequencer**. The **sequencer** number must be provided.

4.6.1   Query: ?SETTINGS **sequencer** <CR>

　　　　　　　　　　sequencer = {0..n}

　　　　　　　　　　Example: ?SETTINGS 4<CR> (get settings data from sequencer 4)

4.6.2   Response: $ACK **sequencer**, SETTINGS

POT ADJ=6%	-> POT Setting
DIP MAX DELAY=10 SECONDS	-> DIP 1-3 setting
TOTAL DELAY=0.6 SECONDS	-> Total delay setting
DIP MODE=MOM	-> DIP 7 Position
DIP SEQ=PRIM	-> DIP 8 Position
DIP ALARM=N. OPEN	-> State of Force Off pins
DIP EVS=AUTO	-> DIP 9 Position
KEY=REM	-> Key Switch position
<EOT>	

FURMAN

4.7 STATUS

Requests the current status of the sequencer specified by **sequencer**. The **sequencer** number must be provided.

4.7.1   Query: ?STATUS **sequencer** <CR>

sequencer = {0..n}

Example: ?STATUS 3<CR> (get status data from sequencer 3)

4.7.2   Response: $ACK **sequencer**, STATUS

SEQ=PRIM,LAST	> Sequencer position
PROTECT=OK	-> State of protection
EVS=OFF	-> State of EVS
SMP RLY=ON	-> State of SMP relay
ALARM=OFF	-> State of Force Off input
BANK1=OFF	-> State of DELAY 1 Output
BANK2=ON	-> State of DELAY 2 Output
BANK3=OFF	-> State of DELAY 3 Output
REMOTE=0V	-> State of Legacy remote input
PUSHBUTTON=OFF	-> State of START SEQUENCE button
SECLINK=NOTOK	-> Connection state of Secondary Smartlink
UART0 PER=0.00%	-> % of Primary missed messages
UART1 PER=0%	-> % of Secondary missed messages
<EOT>	

4.8 BANK STATUS

Requests the status (ON or OFF) of the of the DELAY Outputs on the **sequencer** specified by sequencer. The **sequencer** number must be provided.

4.8.1   Query: ?BANK_STAT **sequencer** <CR>

sequencer = {0..n}

Example: ?BANK_STAT 1<CR> (get bank status data from sequencer 1)

4.8.2   Response: $ACK **sequencer**, BANK_STAT

BANK1=OFF	-> State of DELAY 1 output on **sequencer**
BANK2=ON	-> State of DELAY 2 output on **sequencer**
BANK3=OFF	-> State of DELAY 3 output on **sequencer**
<EOT>	

4.9 ROLLCALL

Creates a list of all sequencers attached to the Primary Sequencer. No sequencer argument is used.

4.9.1   Query: ?ROLLCALL<CR>

Example: ?ROLLCALL<CR> (get a list of all sequencers attached)

4.9.2   Response: $ACK sequencer, (model) LAST <CR>

For a chain of n units:

$ACK 0,CN-1800S	-> Primary unit
$ACK 1,CN-2400S	-> Secondary unit
$ACK n-1,CN-2400S,LAST <EOT>	-> Last Secondary unit

4.10 HELP

Creates a list of all available commands. No sequencer argument is used.

4.10.1   Query: ?HELP<CR>

Example: ?HELP<CR> (get a list of all commands)

4.10.2   Response: Commands

!SEQ_ON	-> See section 3.0
!SEQ_OFF	-> See section 3.1
!ENUMERATE	-> See section 3.2
BANK_ON	-> See section 3.3
!BANK_OFF	-> See section 3.4
!ALL_OFF	-> See section 3.5
!EVENTS_ENABLE	-> See section 3.6
!EVENTS_DISABLE	-> See section 3.7
!RESET	-> See section 3.8
!CLEAR_PER	-> See section 3.9

Queries

?VOLTAGE	-> See section 4.0
?CURRENT	-> See section 4.1
?POWER	-> See section 4.2
?POWER_VA	-> See section 4.3
?PF	-> See section 4.4
?ID	-> See section 4.5
?SETTINGS	-> See section 4.6
?STATUS	-> See section 4.7
?BANK_STAT	-> See section 4.8
?ROLLCALL	-> See section 4.9
?HELP	-> See section 4.10
<EOT>	