CURTIS

Curtis enGage® NX VII (NX7)

Model 3750T User Guide

Digital Instrumentation

Version 1.4

Notices

Curtis Instruments, Inc.
200 Kisco Avenue
Mt. Kisco, NY 10549
https://www.curtisinstruments.com/

Part Number: 53260
Revision: Rev B, June 2025

CAUTION:

Read Instructions Carefully!

Contents

List of Tables

List of Figures

1 — Overview

The Curtis enGage® NX VII (NX7) is a 7 inch, 1000 nit touchscreen display that utilizes a high-performance i.MX 8M+ processor, memory, and peripherals to enable innovative features such as digital video, machine learning, advanced graphical animations, and wireless connectivity. The NX7 is customizable in a multitude of ways, allowing OEMs to realize unique brand identity quickly.

The display is equipped with CAN-FD, the latest CANbus technology, which provides faster data rates and increased reliability than CAN 2.0. Optional Bluetooth and WiFi enable communication with a mobile application or fleet telematics portal. Optional RFID enables vehicle security and operator identification through an RFID tag or mobile phone.

The NX7 represents the most advanced level of the unified enGage® NX product portfolio. All enGage NX products share a common user experience as well as a set of companion tools, such as the mobile application and data server, making it easy to move between product platforms within the portfolio.

For more information on the device's features, see the Features section and the NX7 datasheet on the Curtis Instruments CAN & Serial Instrumentation page.

Figure 1. Curtis enGage® NX VII (NX7)

About this Manual

The following table summarizes the information provided by the manual.

Section	Description
Features	Major product features.
Installation	Mounting instructions and diagram.
Wiring and I/Os	Includes the following information: • Wiring diagram • Connectors • I/O specifications and descriptions
Screens	The screens included in the NX7's generic application.
Configuration	**DCF Editor**, which is a component of the web application used to configure the NX7.
CANopen Communications	The device's CAN functionality, including PDOs and CANopen parameters.

Technical Support

For technical support, contact the Curtis distributor where you obtained your controller or the Curtis sales-support office in your region. The Curtis sales-support office contact information is available from the Curtis Instruments website on the Contacts web page:

https://www.curtisinstruments.com/contact/

Conventions

The manual uses the following conventions.

Numeral System Notation

The following table describes how this manual denotes decimal, binary, and hexadecimal numbers.

Note:

The letter *n* in the format column represents a digit.

Numeral System	Format	Example
Decimal	Either of the following: • *nnn* • *nnn*d	• 127 • 127d
Hexadecimal	Either of the following: • *nnn*h • 0x*nnn*	• 62Ah • 0x62A
Binary	*nnn*b	1011b

In addition, some CANopen examples have hexadecimal values without notation. Those examples are formatted with a monospace font and with the bytes delimited by spaces, as shown in the following example:

```
21 FF 01 11 22 01 00 00
```

Miscellaneous Conventions

- *RO* means read-only.
- *RW* means read-write.
- *N/A* means not applicable.

2 — Features

The following sections describe the NX7's hardware and software features.

Flexible I/Os (Multi-Purpose Pins)

The NX7 includes several multi-purpose pins, known as *flexible I/Os*, that can be used for the following types of I/Os:

- Switch inputs
- Sender inputs (voltage-based or resistance-based)
- Frequency inputs
- Metal-Oxide Semiconductor Field Effect Transistor (MOSFET) outputs

For information on the multi-purpose pins, see J2 Connector.

Related information

Flexible I/Os

LCD

The LCD has a backlight and an ambient light sensor. The brightness level, whether the light sensor is enabled, and the light levels that control the sensor are configurable. Vehicle operators can also use the **Settings** screen to specify the brightness and enable the light sensor.

Related information

Display Configuration

Battery Discharge Indicator (BDI)

The device can display a battery discharge indicator (BDI) that shows the battery's state-of-charge (SoC). The state-of-charge is specified with a CANopen parameter.

Note:

If your application requires an alternative method to calculate the state-of-charge, contact technical support.

Related information

BDI Percentage Parameter

Real-Time Clock

The device provides a real-time clock (RTC) that indicates the current time and date. The RTC is backed by an internal battery, and will track time for ten years with the power removed. The time and date can be specified with the **Settings** screen or with CAN.

Note:

The RTC accounts for leap years.

Related information

Real-Time Clock Parameters

Audible Alarm

The device provides a speaker that is used to sound alarms. The alarm can be commanded by the alarm_command parameter.

Video

The device has inputs for two video sources. The inputs support the NTSC and PAL video formats.

Video can be displayed in normal or mirrored mode. If two cameras are connected, either a switch input or a CANopen parameter can be used to switch between the cameras. The video image can be scaled as designed in the application.

Digital and analog cameras are supported. Analog cameras can be connected to the video inputs on the J3 connector. For information on using digital cameras, contact Curtis technical support.

Multicolor LED

The device provides a multicolor LED. The LED is used by the device to communicate system messages.

The LED can also be used to flash custom signals. The LED's colors and blinking behavior can be controlled through CAN.

Related information

LED Parameters

CAN Features

The NX7 provides two 29-bit CAN channels and is designed to the following standards:

- SAE J1939
- CAN 2.0B
- CAN FD
- CiA DS301

Some models provide an isolated CAN interface that allows each CAN channel to be referenced to a separate isolated ground.

Related information

CAN Connections
CANopen Communications

J1939

The NX7 provides support for transmitting (Tx) and receiving (Rx) J1939 messages. Application developers can define PGNs for Rx and Tx messages and design callbacks that respond to Rx messages.

When the device starts up, it registers ten J1939 variables that store data for Tx messages and ten variables for Rx messages. Message data can be manipulated with bitmasks and bit shifts. The following settings are configured with **DCF Editor**:

- Address claiming for both CAN channels.
- Rx node filters for both CAN channels.
- PGNs
- Rx bitmasks and bit shifts.

In the device's J1939 implementation, the following bit-fields of J1939 29-bit IDs are always 0:

- Extended data page (EDP)
- Data page (DP)

Note:

Reception of data does not depend upon the producer's node ID. The controller does not store the data in non-volatile memory.

Related information

J1939 Configuration
J1939 Parameters

Maintenance Monitors

The NX7 provides three independently-configurable maintenance monitors. The maintenance monitors are for tracking the time until maintenance is due.

A maintenance monitor can be set to a value specified by a CANopen parameter, or can begin counting time when a specified switch input is active or when the monitor is enabled by a CANopen parameter.

The time intervals for required maintenance are configurable. For each maintenance monitor, you can specify two maintenance intervals:

- Initial maintenance
- Subsequent, regular maintenance

The maximum value of the intervals is 1,193,046 hours. Other maintenance monitor features, such as whether a maintenance monitor is resettable or historical, or counts up or down from 0, can be configured.

Related information

Maintenance Monitor Configuration

Hour Meters

The NX7 provides four independently-configurable hour meters. The hour meters measure up to 1,000,000 hours. An hour meter can be configured as resettable or historical. An hour meter can be set to a value specified by a CANopen parameter, or can begin counting time when a specified switch input is active or when the hour meter is enabled by a CANopen parameter.

Related information

Hour Meter Configuration

Speedometers

The NX7 provides two independently-configurable speedometers. A speedometer can be configured as resettable or historical, and can use either CAN or a frequency input as its data source. The speedometers' pulses-per-km and number of samples to be averaged are also configurable.

Note:

The **Units of Measure** setting specifies whether the NX7 displays speedometer and odometer data in miles-per-hour (MPH) or kilometers-per-hour (km/h). The unit of measurement can be specified with **DCF Editor** or the **Settings** screen.

Related information

Speedometer Configuration

Odometers

The NX7 provides three independently-configurable odometers. The odometers measure up to 999,999.9km. An odometer can be configured as resettable or historical, and can use either CAN or a speedometer as its data source.

Note:

The **Units of Measure** setting specifies whether the NX7 displays speedometer and odometer data in miles-per-hour (MPH) or kilometers-per-hour (km/h). The unit of measurement can be specified with **DCF Editor** or the **Settings** screen.

Related information

Odometer Configuration

Tachometers

The NX7 provides two independently-configurable tachometers. The tachometers measure up to 65,535 RPM.

Either CAN or a frequency input can be configured as a tachometer's data source. The pulses-per-revolution and number of samples to average are also configurable.

Related information

Tachometer Configuration

User Interface Customization

OEMs can fully customize the user interface. The Qt framework is used for customization.

For information on customizing the NX7, contact Curtis technical support.

3 — Installation

The enGage NX7 can be mounted with either a panel mount bracket or a RAM mount. The diagrams in Figure 2 through Figure 4 show the NX7's dimensions. For mounting diagrams and instructions, see the Panel Mount Bracket Installation and RAM Mount Installation sections.

Figure 2. Dimensions — Front View (mm)

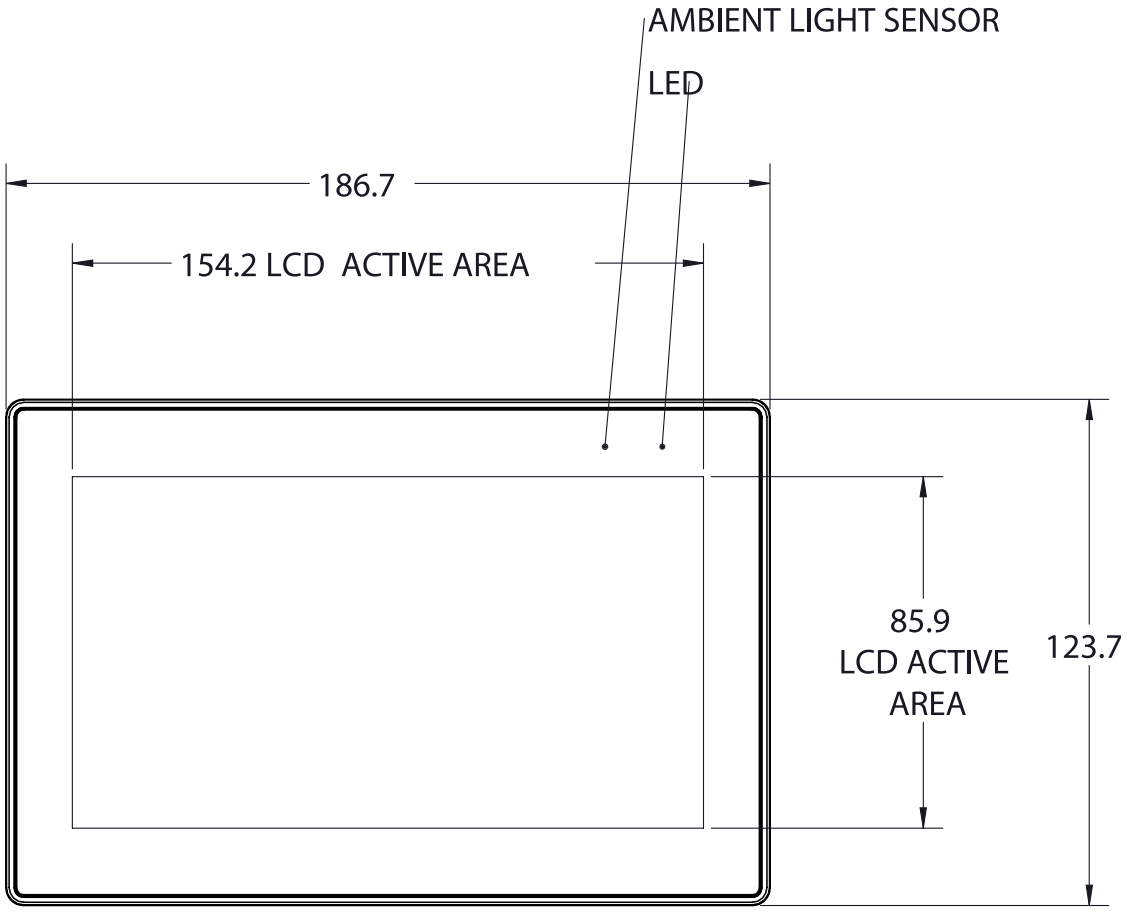


Figure 3. Dimensions — Rear View (mm)

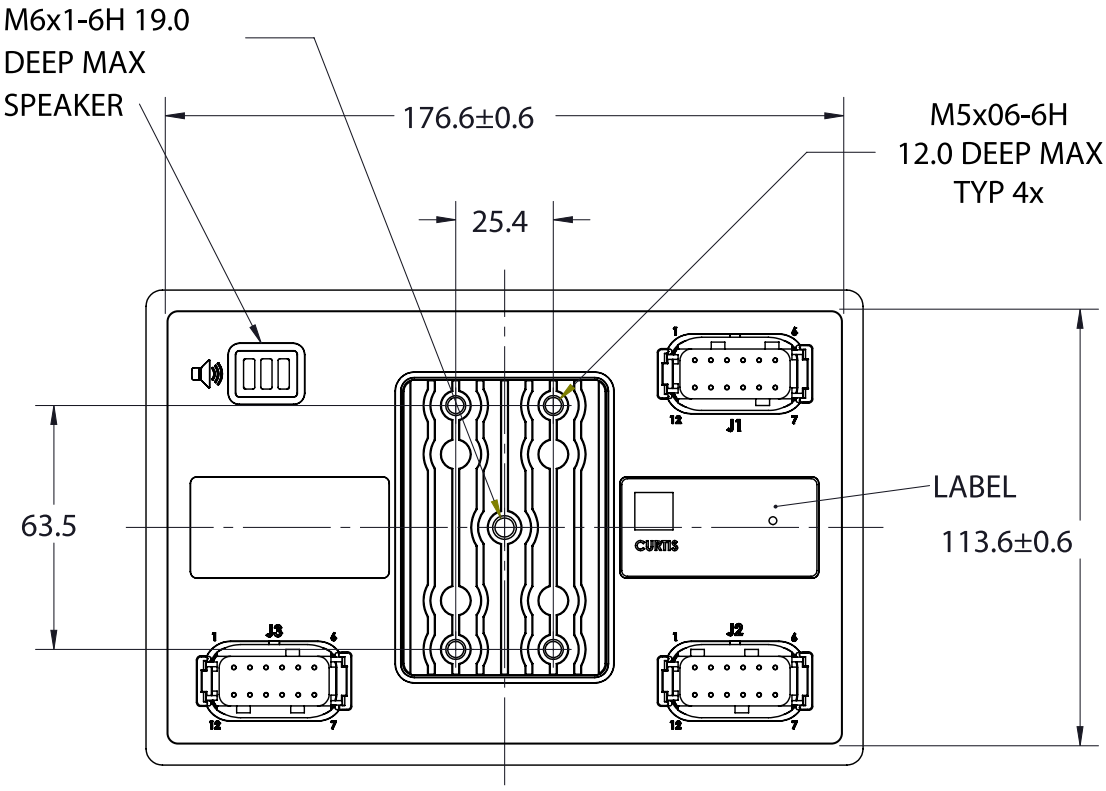
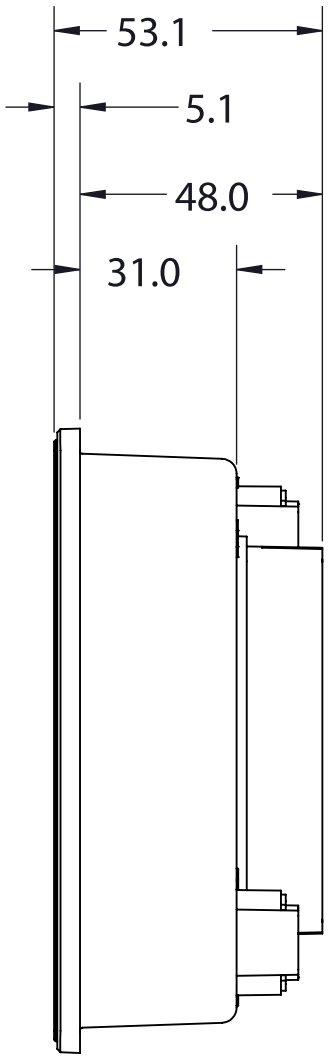


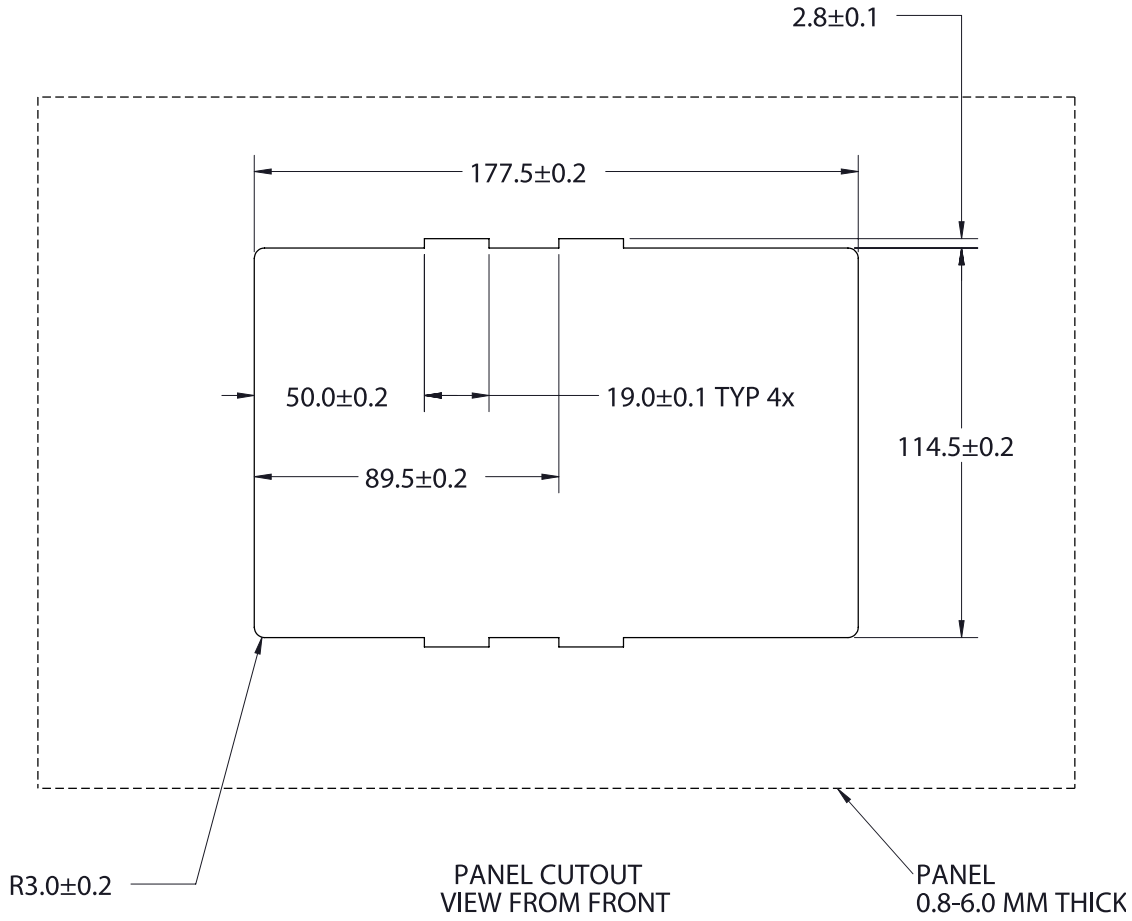
Figure 4. Dimensions — Side View (mm)



Panel Mount Bracket Installation

The panel mount bracket is designed to work with panel thicknesses ranging from 0.8mm–6.4mm. The following diagram shows the panel cutout dimensions.

Figure 5. Panel Cutout Dimensions (mm)



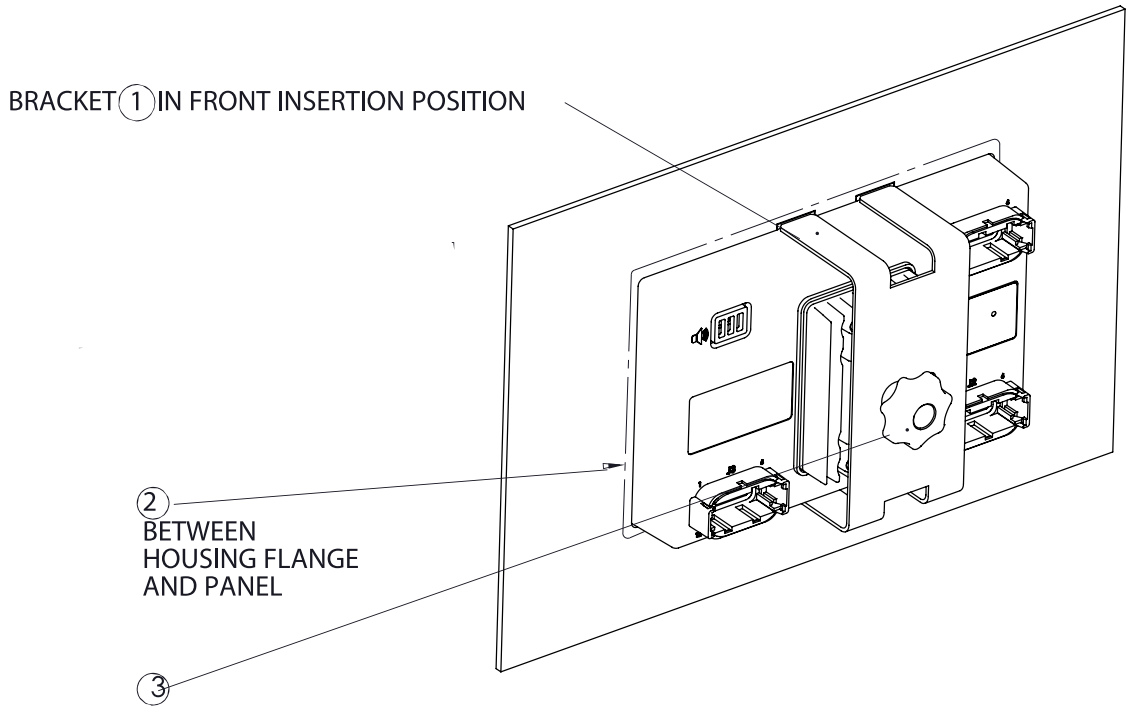
The following table describes the parts for the panel mount bracket.

Table 1. Panel Mount Bracket Parts

Curtis Part Number	Quantity	Description
17778364	1	Panel mount bracket
17778363	1	Gasket
12515-HS-01	1	Hand screw, M6x20
12077WB-0005-M6	1	Bellville washer, M6

The callouts in the following diagram show where the parts are installed.

Figure 6. Panel Mount Bracket

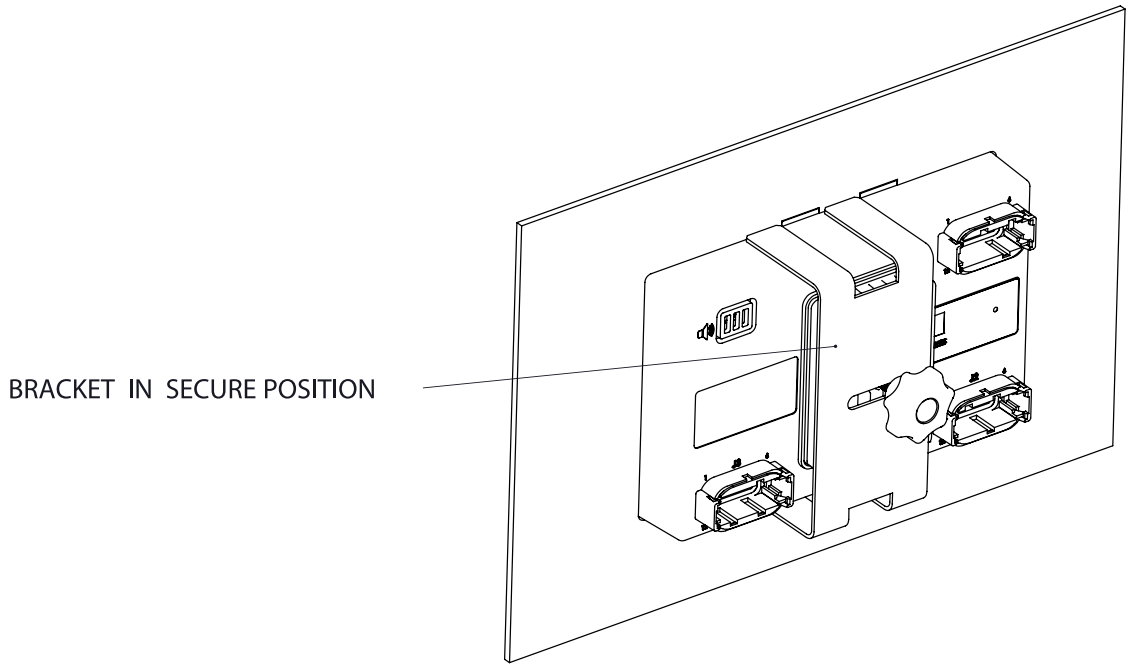


The following table describes the callouts.

Callout	Description
1	The bracket's front insertion point.
2	The gasket is installed between the housing flange and panel.
3	The location where the screw secures the bracket.

The following diagram shows the bracket secured to the NX7.

Figure 7. Panel Mount Bracket — Secured



RAM Mount Installation

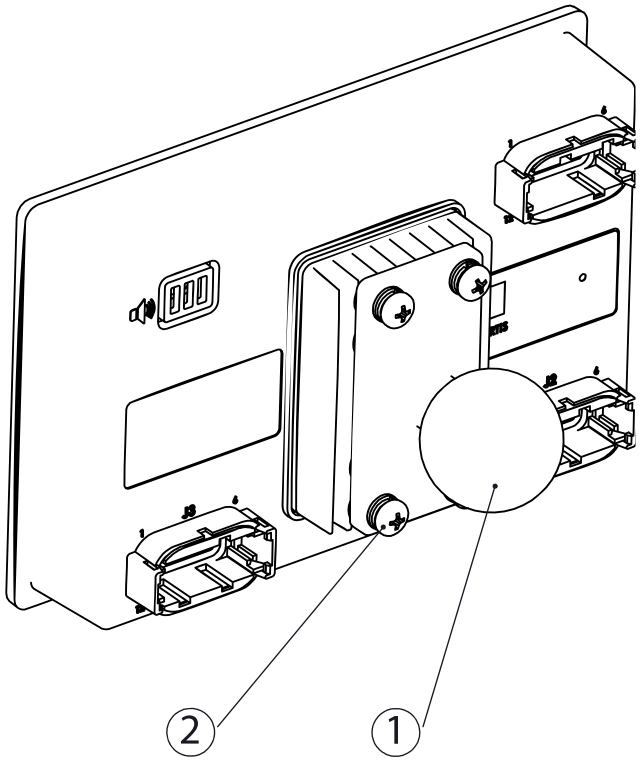
The following table describes the parts for the RAM mount.

Table 2. RAM Mount Parts

Curtis Part Number	Quantity	Description
17778362	1	RAM mount
12057MP-9001-M5-16	4	SEMS screw with spring and flat washer, M5x16

The callouts in the following diagram show where the parts are installed.

Figure 8. RAM Mount



The following table describes the callouts.

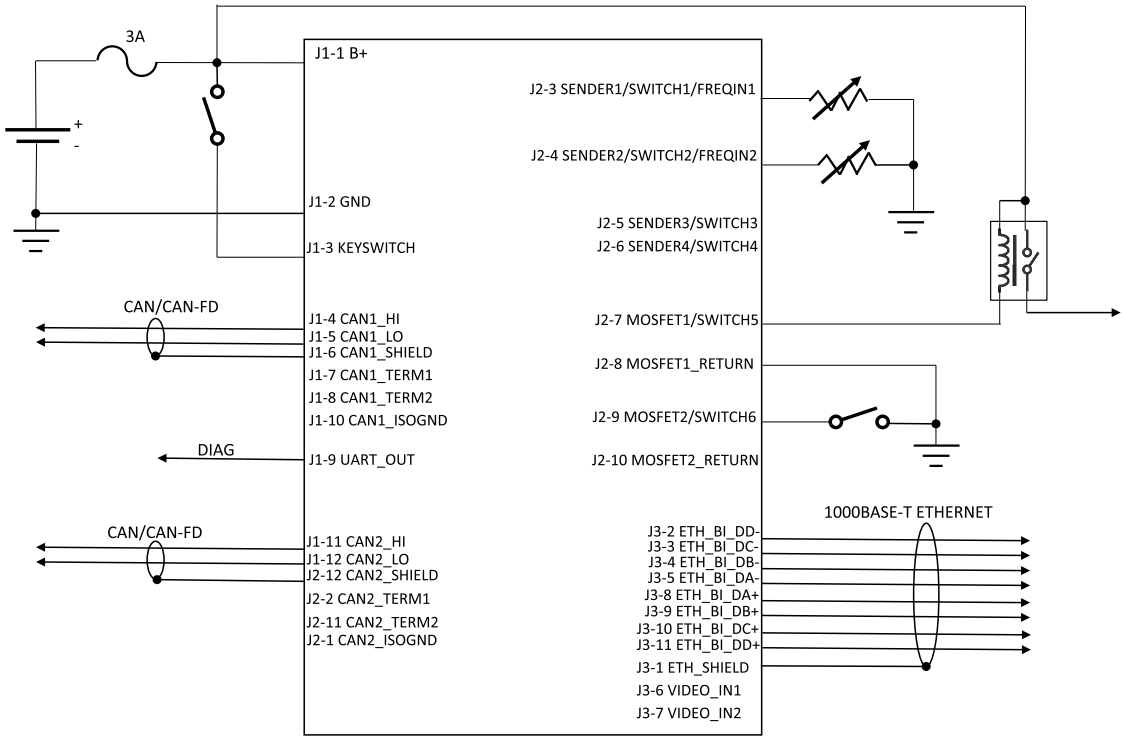
Callout	Description
1	The location of the RAM mount.
2	One of the four screws.

4 — Wiring and I/Os

Wiring Diagram

Following is a typical wiring diagram for the NX7.

Figure 9. Wiring Diagram



The following topics describe the I/Os.

Connectors

The NX7 is available with I/Os or as a CAN bus-only unit. Models with I/Os have three 12-pin connectors, labelled J1, J2, and J3. The CAN bus-only unit has only the J1 connector.

The mating connectors are from the Deutsch DTM series, and require Deutsch wedge locks and sockets. The following tables describe the Deutsch and Curtis part numbers.

Table 3. Mating Connectors

Connector	Deutsch Part Number	Curtis Part Number
J1	DTM06-12SA	12690FH140
J2	DTM06-12SB	12690FH141

Table 3. Mating Connectors (continued)

Connector	Deutsch Part Number	Curtis Part Number
J3	DTM06-12SC	12690FH142

Table 4. Wedge Lock and Socket for Mating Connectors

Part	Deutsch Part Number	Curtis Part Number
Wedge lock	WM-12S	12690CA29
Socket	Either of the following: • 0462-201-20141 • 1062-20-0122	The following list shows the respective Curtis part numbers: • 12690FC59 • *There is not a corresponding Curtis part number for 1062-20-0122.*

J1 Connector

The following table describes the J1 connector's pins.

Table 5. J1 Connector

Pin Number	Description
J1-1	Battery Voltage (B+)
J1-2	Battery Common (B–)
J1-3	Keyswitch
J1-4	CAN 1 High
J1-5	CAN 1 Low
J1-6	CAN 1 Shield
J1-7	CAN 1, Terminating Resistor 1
J1-8	CAN 1, Terminating Resistor 2
J1-9	*Reserved*
J1-10	Isolated Ground, CAN 1
J1-11	CAN 2 High
J1-12	CAN 2 Low

J2 Connector

The J2 connector includes several flexible I/Os (multi-purpose pins). For example, pin J2-3 can be used as a sender, switch, or frequency input. The following table describes the J2 connector's pins.

Table 6. J2 Connector

Pin Number	Description
J2-1	Isolated Ground, CAN 2
J2-2	CAN 2, Terminating Resistor 1
J2-3	Sender 1 / Switch 1 / Frequency 1 / Digital Output 1
J2-4	Sender 2 / Switch 2 / Frequency 2 / Digital Output 2
J2-5	Sender 3 / Switch 3 / Digital Output 3
J2-6	Sender 4 / Switch 4 / Digital Output 4
J2-7	MOSFET 1 / Switch 5 / Digital Output 5
J2-8	MOSFET 1 Return
J2-9	MOSFET 2 / Switch 6 / Digital Output 6
J2-10	MOSFET 2 Return
J2-11	CAN 2, Terminating Resistor 2
J2-12	CAN 2 Shield

J3 Connector

The following table describes the J3 connector's pins.

Table 7. J3 Connector

Pin Number	Description
J3-1	Ethernet Shield
J3-2	Ethernet BI DD–
J3-3	Ethernet BI DC–
J3-4	Ethernet BI DB–
J3-5	Ethernet BI DA–
J3-6	Video Input 1 Signal
J3-7	Video Input 2 Signal
J3-8	Ethernet BI DA+
J3-9	Ethernet BI DB+
J3-10	Ethernet BI DC+

Table 7. J3 Connector (continued)

Pin Number	Description
J3-11	Ethernet BI DD+
J3-12	*Reserved*

Operating Voltage

The following table describes the operating voltages.

Table 8. Operating Voltage

Voltage Rating	Minimum Operating Voltage	Maximum Operating Voltage
12–48V	9V	60V

Note:

Unless otherwise specified, the voltages listed in this manual are DC voltages.

The NX7 withstands reverse polarity connections continuously and the jumper start conditions listed in SAE J1455 (Table 3A and Table 3B).

The device includes an overvoltage protection circuit that turns off power in the event of a potentially damaging overvoltage condition. Power is restored when the overvoltage condition is removed. Due to circuit tolerances, the overvoltage trip point may vary from device to device and by temperature. The following table describes the overvoltage trip point.

Minimum	Nominal	Maximum
66.4V	69.1V	73.0V

Operating Current

The following table describes the typical and maximum operating currents.

Table 9. Operating Current

B+ (VDC)	Typical Operating Current[1] (mA)		Maximum Operating Current[2] (mA)	
	Keyswitch Off	Keyswitch On	Keyswitch Off	Keyswitch On
9	72	950	75	2000
12	60	660	65	1500
24	50	350	60	780
36	48	240	55	516
48	45	200	50	400
60	42	175	45	350

I/Os

The following sections describe the I/Os.

Power Connections

Connect the power supply to the B+ and B– inputs (pins J1-1 and J1-2). Curtis recommends that you include a fuse in the circuit that connects the battery to the B+ input, as shown in Figure 9. The fuse protects the power system from external shorts. Size the fuse according to the application's requirements.

For information on operating voltages and overvoltage protection, see Operating Voltage.

Keyswitch Input

The keyswitch input is active when it is switched to B+. Connect the keyswitch to pin J1-3.

1. All the flexible I/Os are configured as switch inputs and are open. Ambient temperature: 25°C.

2. Flexible I/Os 1, 2, 3, and 4 are configured as sender inputs and are connected to 0Ω. Flexible I/Os 5 and 6 are configured as switch inputs and are closed. The LCD is at its maximum brightness level. Audio, GPU, and wireless are active. The ambient temperature is in the range of –40°C to 70°C

The keyswitch specifications are the same as those for the switch inputs; see Table 11. Various functions, such as hour meters and maintenance monitors, can be configured to activate when the keyswitch is active.

Flexible I/Os

The J2 connector provides six flexible I/Os. The following table lists the flexible I/O pins and their corresponding functions.

Table 10. Flexible I/Os

Name	Pin	Sender Input	Switch Input	Frequency Input	MOSFET Output	Digital Output
Flexible I/O 1	J2-3	✓	✓	✓		✓
Flexible I/O 2	J2-4	✓	✓	✓		✓
Flexible I/O 3	J2-5	✓	✓			✓
Flexible I/O 4	J2-6	✓	✓			✓
Flexible I/O 5	J2-7		✓		✓	✓
Flexible I/O 6	J2-9		✓		✓	✓

The following topics describe the input and output functions.

Related information

Flexible I/Os (Multi-Purpose Pins)
Flexible I/O Configuration

Switch Inputs

The J2 connector includes six pins that can be configured as digital switch inputs. The inputs can either be active low (switched to B–) or active high (switched to B+).

The statuses of the switch inputs are available on the CAN bus. The statuses can be used for functions such as displaying and hiding vehicle status icons.

Table 11. Switch Input Specifications

Specification	Value
Input Range	0–60V
Active High Threshold	8.0V
Active Low Threshold	1.0V

Table 11. Switch Input Specifications (continued)

Specification	Value
Input Impedance	The impedance depends upon the switch: • Switches 1–4: 58kΩ–60kΩ • Switches 5–6: 741kΩ–819kΩ

Related information

Flexible I/O Configuration
Switch Input Parameters

Sender Inputs

Pins J2-3 through J2-6 can be used as analog sender inputs. Sender inputs must be referenced to the NX7's system ground (B–). Each sender input can be independently configured as a resistive or voltage-based input.

Table 12. Sender Input Specifications

Specification	Value
Voltage Input Range	0–60V
Voltage Measurement Range	0–10V
Voltage Resolution	10mV
Voltage Measurement Error	±(1% + 40mV)
Resistance Measurement Range	0–10kΩ
Resistance Resolution (0–1200Ω)	0.2–5.0Ω
Resistance Resolution (1.2kΩ–10kΩ)	5–35Ω
Resistance Measurement Error	±(3% + 2Ω)

Related information

Flexible I/O Configuration
Sender Input Parameters

Frequency Inputs

Pins J2-3 and J2-4 can be configured as frequency inputs. Frequency inputs can be used as data sources for display elements such as speedometers and tachometers.

The NX7 filters and conditions frequency input signals, then sends the signals to an internal timer that measures the period of the signals' waveforms. Values are measured in microseconds and are available on the CAN bus.

Note:

The high and low threshold levels have two configurable options.

Table 13. Frequency Input Specifications

Specification	Value
Active High Threshold 1	3.28V
Active Low Threshold 1	2.95V
Active High Threshold 2	1.38V
Active Low Threshold 2	1.04V
Input Impedance	42.7kΩ–60kΩ
Maximum Input Voltage	60V
Frequency	1–10 kHz
Duty Cycle	10–90%
Resolution	1 µsec
Accuracy	0.5%

Related information

Flexible I/O Configuration
Frequency Input Parameters

MOSFET Outputs

The NX7 has two independent open-drain MOSFET outputs that are PWM-controlled. These outputs can also be configured as binary push-pull outputs. Pins J2-7 and J2-8 are for MOSFET 1 and pins J2-9 and J2-10 are for MOSFET 2.

When MOSFET outputs are activated, the outputs connect to their return pins. The MOSFET return pins must be connected to B– externally. All of the MOSFET pins incorporate an inductive spike protection diode connected to B+. The MOSFET outputs have current limiting circuitry.

Table 14. MOSFET Output Specifications

Specification	Value
Continuous Current	0–2A
Off Voltage	60V
On Voltage (reverse polarity protected)	1.0–2.0V (with 2A of current)

Table 14. MOSFET Output Specifications (continued)

Specification	Value
PWM Output Frequency	65 kHz

Related information

Flexible I/O Configuration
MOSFET Output Parameters

Digital Outputs

All of the flexible I/Os can be configured as digital outputs. The outputs emulate digital "push-pull" outputs by enabling and disabling a pull-up resistor to 5V. The digital outputs are high-impedance and are designed to interface with digital controllers that have high impedance inputs. A digital output's status can be specified with CANopen parameters or internal logic and read with CANopen parameters.

The following table describes the digital output specifications.

Table 15. Digital Output Specifications

Specification	Value
On Voltage, Open Circuit Load	4.1–4.5V
On Voltage, 10kΩ Load (Flexible I/Os 1, 2, 3, and 4)	4.0–4.3V
On Voltage, 100kΩ Load (Flexible I/Os 5 and 6)	3.2–3.5V
Off Voltage, any Load	0–0.3V
Current (sourced)	10mA

Ethernet Ports

The J3 connector provides two 1000BASE-T Ethernet ports. The ports can be used for digital cameras and to update the NX7's software.

Note:

For information on connecting digital cameras, contact Curtis technical support.

Video Inputs

The J3 connector has inputs for two analog video sources. The inputs support the NTSC and PAL video formats. Analog cameras are plug and play.

CAN Connections

The following table describes the CAN connection pins for both CAN channels:

Function	CAN Channel 1	CAN Channel 2	Comments
CAN bus signals	Pins J1-4 (CAN 1 High) and J1-5 (CAN 1 Low).	Pins J1-11 (CAN 2 High) and J1-12 (CAN 2 Low).	Use twisted-pair wiring to minimize the likelihood of picking up a voltage bias on only one signal.
Internal 120Ω termination resistor	Short pins J1-7 and J1-8.	Short pins J2-1 and J2-11.	
CAN cable shield termination network	Pin J1-6.	Pin J2-12	The cable shield termination is a 1Ω resistor in series with a 0.68 µF capacitor connected to ground.

Isolated CAN Interface

The optional isolated CAN interface allows each CAN channel to be referenced to a separate isolated ground. The isolated CAN bus circuitry has its own power supply and provides galvanic isolation from the rest of the unit's circuitry. The ground for the isolated power supply is connected to pin J1-10 for CAN channel 1 and pin J2-1 for CAN channel 2.

The isolated CAN bus's dielectric insulation voltage is 1500Vrms for a one-minute duration.

Related information

CANopen Communications
CAN Features

5 — Screens

Users can change screens by tapping the buttons on the bottom of the screen. The following table describes the buttons.

Button	Screen
	**Home**
	**Settings**
	**Faults**
	**Video**
	**Library**

The NX7 includes a generic application. The following topics describe the generic application's screens.

Home Screen

The **Home** screen displays data such as the vehicle speed, driving direction, time, and vehicle status. The content of the **Home** screen depends upon the application.

The **Home** screen of the generic application includes the following screen elements:

- Speedometer
- Tachometer
- Battery discharge indicator (BDI)
- Hour meter (in the upper left corner)
- Real-time clock

- Icons that display above the buttons. Some of the application's icons are visible, others are hidden. The visible icons depend upon the vehicle system status.

  In the following example, the icons indicate the vehicle is driving in reverse with the headlights on:

Settings Screen

The **Settings** screen provides the following screens, which are accessed by tapping the menu items on the left:

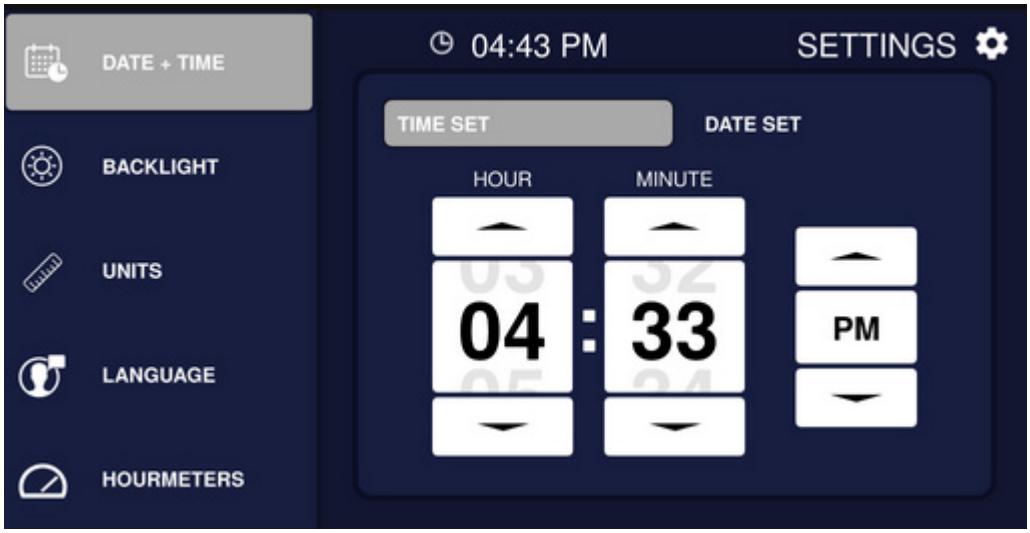
- DATE + TIME Screen
- BACKLIGHT Screen
- UNITS Screen
- LANGUAGE Screen
- HOURMETERS Screen
- ABOUT Screen

The following topics describe the screens.

DATE + TIME Screen

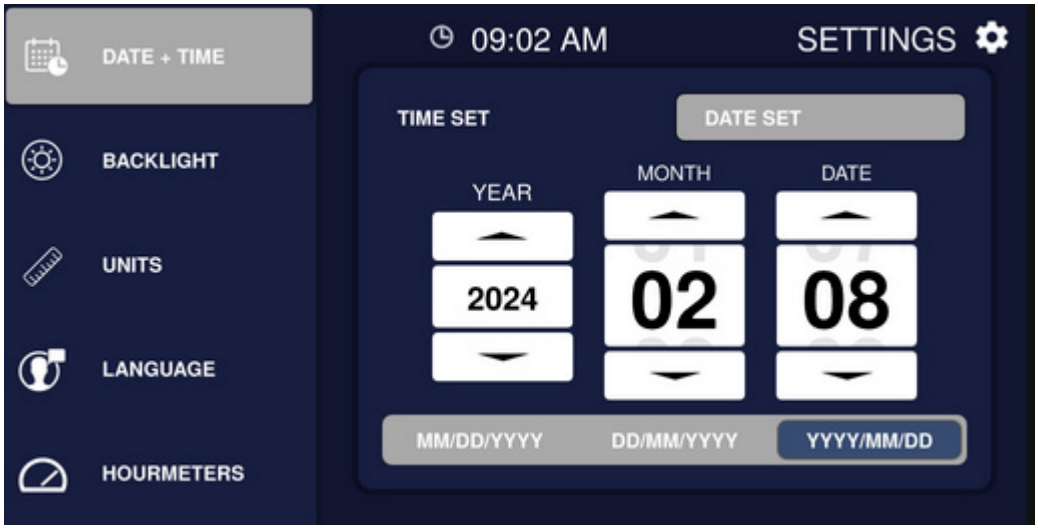
The **DATE + TIME** screen is for specifying the date and time.

To update the time, tap **TIME SET**. Specify the time by scrolling the **HOUR**, **MINUTE**, and **AM/PM** controls:



To update the date, tap **DATE SET**. Specify the date by scrolling the **MONTH**, **DATE**, and **YEAR** controls.

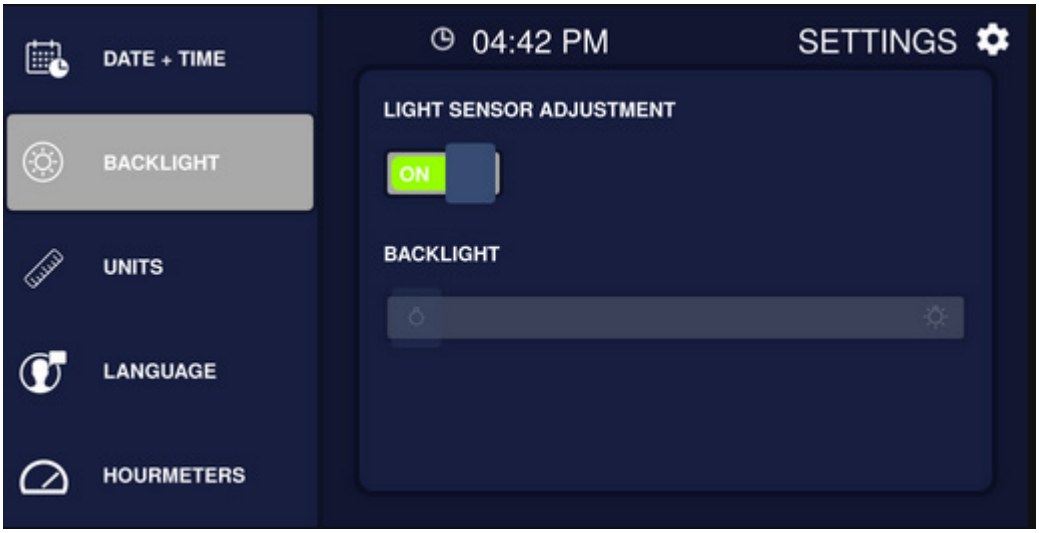
To specify the date format, tap the applicable format. The order of the scrollable controls corresponds to the date format. The following example shows the screen when the YYYY/MM/DD format is selected:



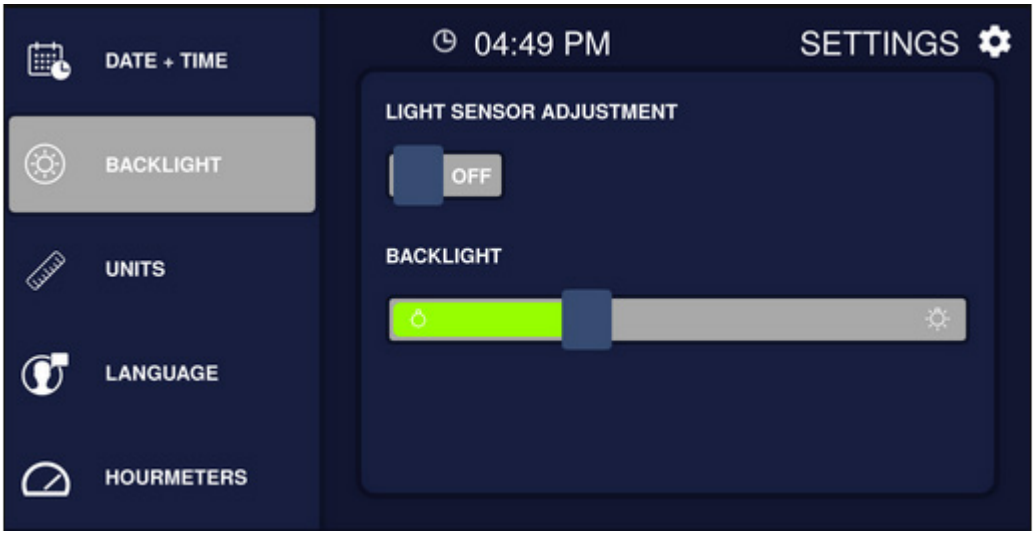
BACKLIGHT Screen

The **BACKLIGHT** screen is for enabling and disabling the ambient light sensor, and for adjusting the brightness level when the ambient light sensor is disabled.

- When the **LIGHT SENSOR ADJUSTMENT** button is set to ON, the ambient light sensor is enabled and the **BACKLIGHT** slider is disabled:

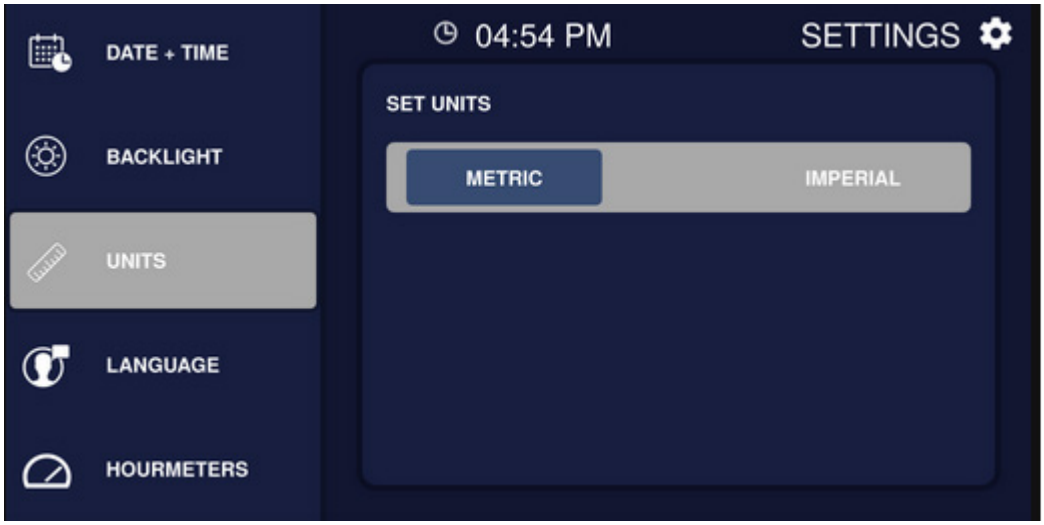


- When the **LIGHT SENSOR ADJUSTMENT** button is set to OFF, the ambient light sensor is disabled and the **BACKLIGHT** slider is enabled. To specify the brightness label, move the slider:



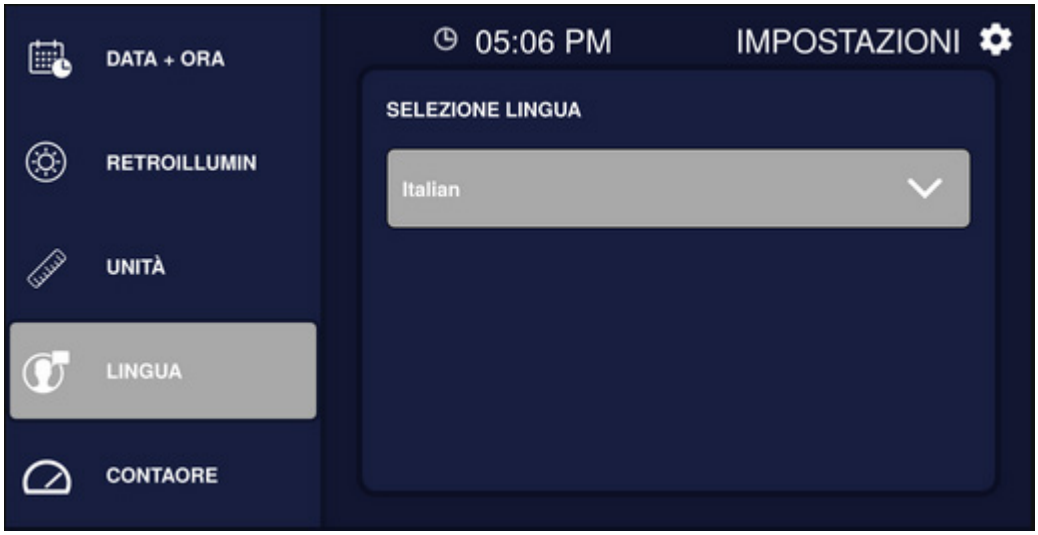
UNITS Screen

The **UNITS** screen is for specifying whether the device displays data using the metric system or the imperial system. To specify, tap **METRIC** or **IMPERIAL**.



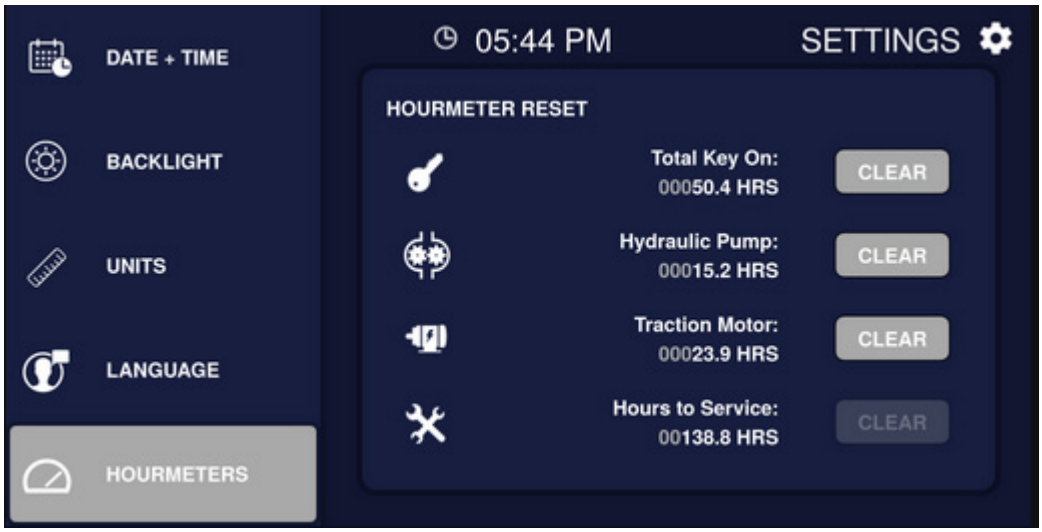
LANGUAGE Screen

The **LANGUAGE** screen provides a dropdown list from which users can select the language in which the device displays text.



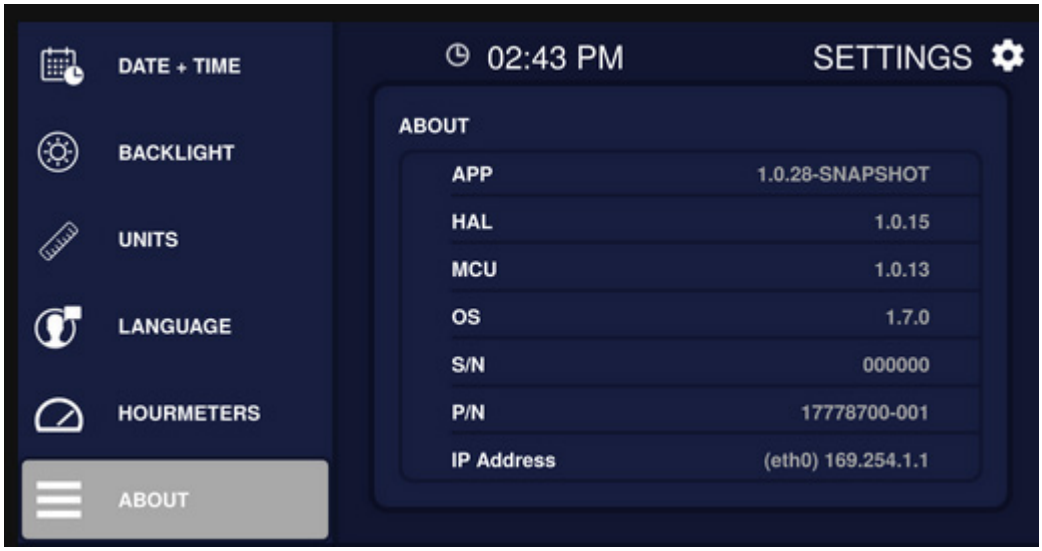
HOURMETERS Screen

The **HOURMETERS** screen displays data for three hour meters and the **Hours to Service** maintenance monitor. Resettable hour meters and maintenance monitors have active **CLEAR** buttons. To reset an hour meter or maintenance monitor to 0.0 hours, press its **CLEAR** button.



ABOUT Screen

The **ABOUT** screen displays system information such as the NX7's software version numbers, device's serial and part numbers, and IP address.

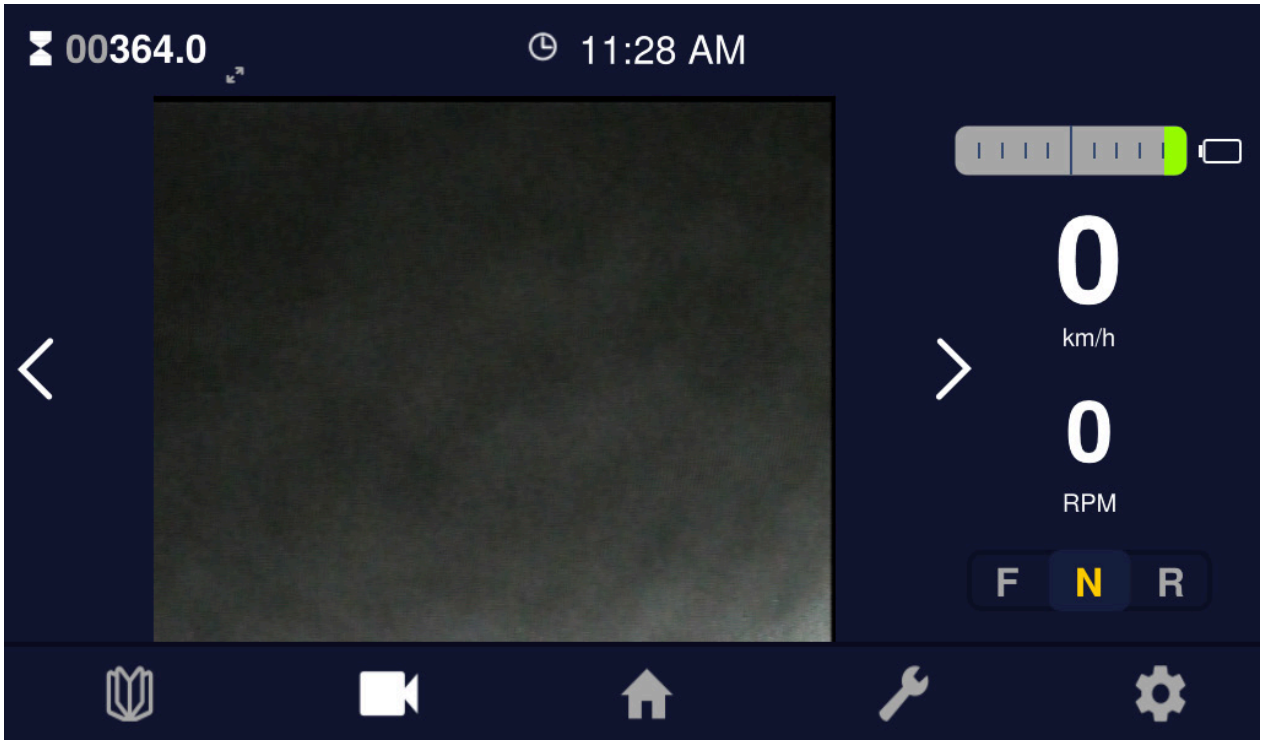


Faults Screen

The **Faults** screen displays data for faults for which CANopen emergency messages have been sent. The screen displays active faults' events, times and dates, and fault codes.

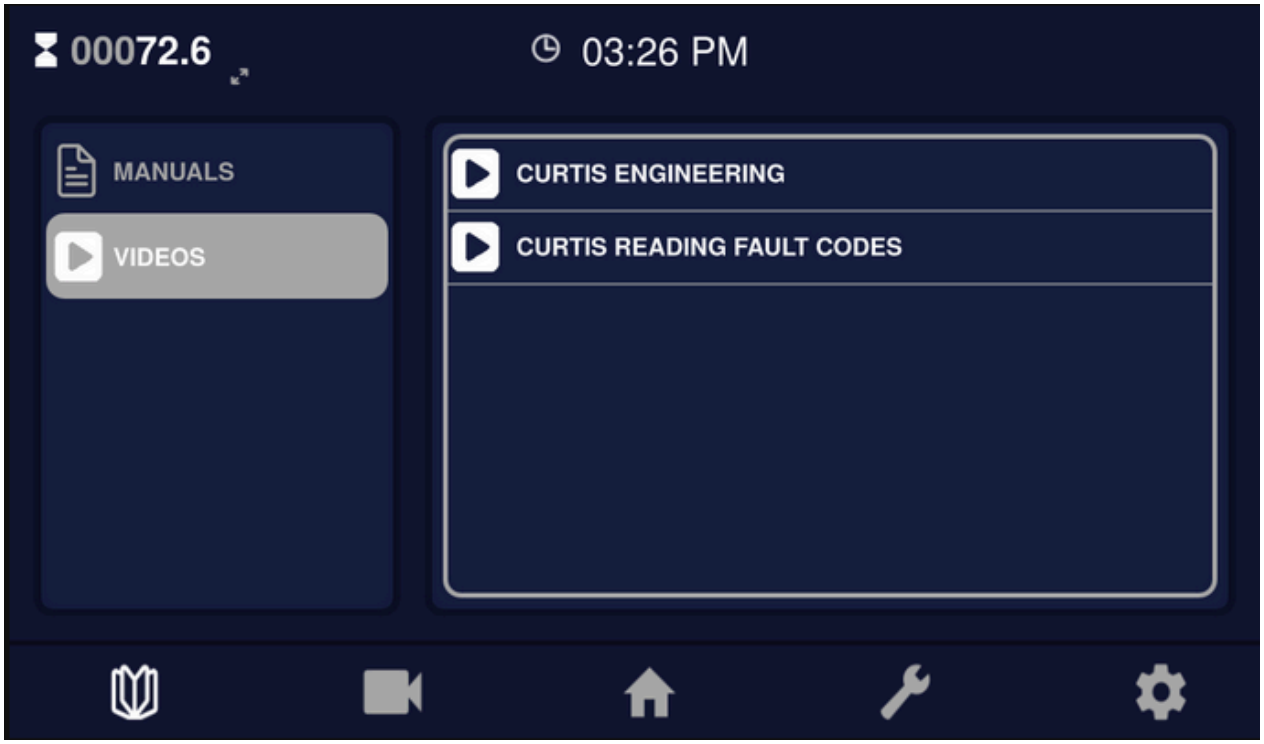
Video Screen

The following example shows the **Video** screen with two cameras connected. The user can switch between the cameras by tapping the arrows to the left and right of the displayed image.



Library Screen

The **Library** screen displays documents and videos that apply to an application. The following image shows an example. The content will vary by application.



6 — Configuration

The NX7 includes a configuration web application. Contact Curtis technical support for the URL and login credentials.

The web application's **DCF Editor** page is used to configure the device. **DCF Editor** contains the following tabs:

- CAN Settings
- J1939 Configuration
- Counters (maintenance monitors and hour meters)
- FlexIO (switch inputs, sender inputs, frequency inputs, and MOSFET outputs)
- Revolution Counters (tachometers, odometers, and speedometers)
- Display
- Icons
- Custom

Notice:

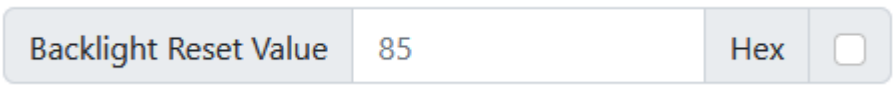
Do not use the web application pages other than **DCF Editor** without guidance from Curtis technical support.

Most of the settings are contained by collapsible sections, which are collapsed by default. After you change settings, press the **Save to Device** button.

Decimal and Hexadecimal Values

Many settings can be specified with either decimal or hexadecimal values. These settings include the text “Hex” to the left of a check box. When the check box is selected, the value is a hexadecimal number:

When the check box is cleared, the value is a decimal number:



CAN Configuration

The **CAN Settings** tab in **DCF Editor** configures the following items:

- Heartbeat rate
- PDO settings
- PDO mappings
- CAN 1 and CAN 2 channels

Heartbeat Rate

The **Heartbeat Rate** section specifies the heartbeat rate for both CAN channels. Valid values are 0–2000ms; the default value is 128ms.. 0 specifies no heartbeat.

PDO Settings

The **PDO 1 Rx Settings** through **PDO 8 Rx Settings** sections configure the RPDOs' COB-IDs and timeouts. The **PDO 1 Tx Settings** through **PDO 8 Tx Settings** sections configure the TPDOs' COB-IDs. The following list describes the valid values:

- PDO COB IDs: 128–4294967295
- PDO Timeouts: 0–65535ms

Note:

These sections correspond to the PDO communication objects. See PDOs.

PDO Mappings

The **PDO 1 Rx Mappings** through **PDO 8 Rx Mappings** sections and **PDO 1 Tx Mappings** through **PDO 8 Tx Mappings** sections specify the CAN objects for which the PDOs send and receive data.

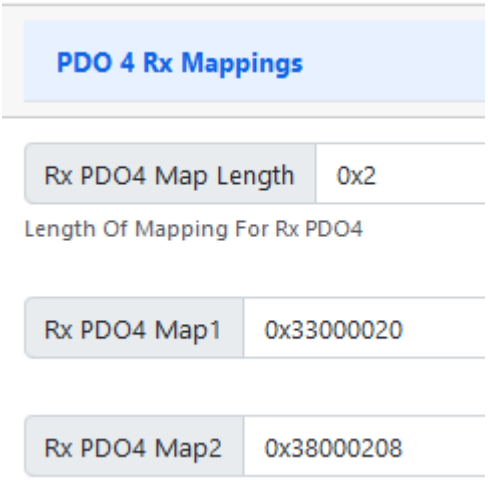
Note:

The **PDO Mappings** sections correspond to the PDO mapping objects described in Table 17.

The **PDO Mappings** sections consist of the following parameters:

- A **Map Length** setting that specifies the number of CAN objects for which the PDO will transmit or receive data.
- Eight **Map*n*** settings, which specify the CAN objects. The first four characters specify the object's CAN index, and the fifth and sixth characters specify the sub-index. The last two characters specify the object's data size.

Consider the following example for RPDO 4:



The following list describes the example:

- The **Rx PDO4 Map Length** setting specifies that the PDO receives data for the objects specified with the **Rx PDO4 Map1** and **Rx PDO4 Map2** settings
- **Rx PDO4 Map1** specifies a 32-bit CAN object with an index and sub-index of `0x3300:00`.
- **Rx PDO4 Map2** specifies an 8-bit CAN object with an index and sub-index of `0x3800:02`.
- Since **Rx PDO4 Map Length** specifies 2, the objects specified with the **Rx PDO4 Map3** through **Rx PDO4 Map8** settings do not receive data.

CAN 1 and CAN 2 Channels

The **Channel 1 Settings** and **Channel 2 Settings** sections allow the following properties to be independently configured for each channel.

Property	Setting Name	Values
Node ID	**Node Id 1/2**	0x01–0x7F
Baud rate	**Baud Rate Ch1/2**	One of the following values: 100 Kbps 125 Kbps 250 Kbps 500 Kbps (default) 800 Kbps 1 Mbps
NMT state upon startup	**Start Operational 1/2**	One of the following values: 0 = Pre-operational NMT state 1 = Operational NMT state

J1939 Configuration

The **J1939 Settings** tab in **DCF Editor** configures the following items:

- Address claiming for both CAN channels.
- Rx node filters for both CAN channels.
- PGNs
- Rx bitmasks and bit shifts.

The **J1939 Settings** tab contains the settings and sections described below.

J1939 Address Claiming Start

The **J1939 Address Claiming Start** settings specify the starting addresses for CAN channels 1 and 2.

J1939 Receive Node Filter

The **J1939 Receive Node Filter** settings specify node filters for CAN channels 1 and 2. To receive data from any node, specify 0xFF.

Custom J1939 Receive PGN/SPN

The device supports ten custom variables that store data for received J1939 messages. The **Custom J1939 Receive PGN/SPN** section contains the **J1939 Custom Rx *n*** settings, where ***n*** is the number at the end of the variable name. To register a J1939 Rx custom variable, specify the following:

- In most significant 16 bits, specify the PGN.
- In the least significant 16 bits, specify the SPN.

Note:

To register a variable that receives the most significant 32 bits of data, you must also register a variable that receives the least significant 32 bits. You can however register a variable that receives the least significant 32 bits without registering one for the most significant 32 bits.

Custom J1939 Receive Filters

The **Custom J1939 Receive Filters** section contains the **J1939 Filter Rx *n*** settings. The settings specify bitmasks for the corresponding Rx messages.

The device performs a bitwise AND operation with the **J1939 Filter Rx *n*** value and the data that results from the bit shift specified by the corresponding **J1939 Shift Rx *n*** setting.

Custom J1939 Receive Shift

The most significant or least significant 32 bits of a message's data can be bit-shifted to the right. The **J1939 Shift Rx *n*** settings in the **Custom J1939 Receive Shift** section specify the number of bits to shift for the corresponding **J1939 Custom Rx *n*** message's data. After the bits are shifted, the device performs a bitwise AND operation on the resulting data and the corresponding **J1939 Filter Rx *n*** bitmask.

For example, suppose the data for the **J1939 Custom Rx 2** message is 0x100 and the **J1939 Filter Rx 2** setting specifies 0xFF as the bitmask:

- If **J1939 Shift Rx 2** specifies 0x04, the result of the bit shift is 0x10. The result of the bitwise AND with the **J1939 Filter Rx 2** bitmask is 0x10.
- If **J1939 Shift Rx 2** specifies 0, the result of the bit shift is 0x100. The result of the bitwise AND with the bitmask is 0.

Custom Transmit J1939 PGNs

The device supports ten custom variables for data transmitted by J1939 messages. The **Custom Transmit J1939 PGNs** section contains the **J1939 Custom Tx *n*** settings, where ***n*** is the number at the end of the variable name. In these settings, specify the PGNs for transmit messages.

Related information

J1939
J1939 Parameters

Maintenance Monitor and Hour Meter Configuration

The **Counters** tab of **DCF Editor** is used to configure the maintenance monitors and hour meters.

Maintenance Monitor Configuration

The **Counters** tab includes sections for configuring the maintenance monitors. Since the maintenance monitors can be independently configured, each section has settings for maintenance monitors 1, 2, and 3. The following table describes the sections.

Note:

The maximum value for maintenance monitor settings is 1,193,046 hours.

Section	Description
Maintenance Monitor Resettable	Specifies whether the maintenance monitors are resettable or historical.
Maintenance Monitors Direction	Specifies whether the maintenance intervals increment from or decrement to zero.

Section	Description
Maint Monitor Initial Reset Interval	Specifies the initial maintenance intervals, which are for the first time the vehicle requires maintenance. The intervals are specified in hours.
Maint Monitor Reset Interval	Specifies the maintenance intervals for regular maintenance, after the initial maintenance intervals expire. The intervals are specified in hours.
Maint Monitor Source	Specifies the inputs that trigger the maintenance monitors to start counting. For each monitor, you can either disable the monitor or specify one the following inputs: • CAN • CanEnabled: Data increments when the corresponding `MaintMon`***`n`***`Enable` parameter is set to 1. The starting value is specified with the `MaintMon`***`n`***`SetManualVal` parameter. • Any of the switch inputs. For a switch input, specify either high or low (B+ or ground).

Related information

Maintenance Monitors

Hour Meter Configuration

The **Counters** tab includes sections for configuring the hour meters. Since the hour meters can be independently configured, each section has settings for hour meters 1, 2, 3, and 4. The following table describes the sections.

Section	Description
Hourmeter Source	Specifies the inputs that trigger the hour meters to start counting. For each hour meter, you can either disable the meter or specify one the following inputs: • CAN • CanEnabled: Data increments when the corresponding `HourMeter`***`n`***`Enable` parameter is set to 1. The starting value is specified with the `Hour`***`n`***`SetManualVal` parameter. • Any of the switch inputs. For a switch input, specify either high or low (B+ or ground).
Hourmeter Resettable	Specifies whether the hour meters are resettable or historical.

Related information

Hour Meters

Flexible I/O Configuration

The following sections of the **FlexIO** tab of **DCF Editor** are used to configure the flexible I/Os.

FlexIO 1 Settings through FlexIO 4 Settings

The **FlexIO 1 Settings** through **FlexIO 4 Settings** sections contain the following settings, which configure pins J2-3 through J2-6:

- **FlexIO Mode 1** through **FlexIO Mode 4** configure the functions for which the pins are used. The following table describes the pins and their corresponding functions:

Setting	Pin	Sender Input	Switch Input	Frequency Input	Digital Output
FlexIO Mode 1	J2-3	✓	✓	✓	✓
FlexIO Mode 2	J2-4	✓	✓	✓	✓
FlexIO Mode 3	J2-5	✓	✓		✓
FlexIO Mode 4	J2-6	✓	✓		✓

  Flexible I/Os configured as digital switch inputs may be used as data sources for icons. For example, if flexible I/Os 1 and 3 are configured as digital switch inputs, Sw1 and Sw3 are contained by the **Icon *n* Source** dropdown lists on the Icons tab.

  Note:

  If a pin is unused, specify Disabled.

- **Flex IO 1 Samples to Average** through **Flex IO 4 Samples to Average** specify the number of samples required. Valid values are 1–128. For instant values, specify 1.

Mosfet 1 and Mosfet 2

The **Mosfet 1** and **Mosfet 2** sections contain the **Mosfet1 Configuration** and **Mosfet2 Configuration** settings, which configure how pins J2-7 and J2-9, respectively, are used. The pins can be configured for the following functions. If a pin is unused, specify Disabled.

- PWM output
- Switch input
- Digital output

Mosfet PWM

The **Mosfet PWM** section contains the following settings, which apply to flexible I/Os configured as Mosfet outputs.

Setting	Description
Mosfet PWM Output Current Threshold	Valid values are 0-4000mA.
Mosfet PWM Output Frequency	Valid values are 1–65 kHz.

Related information

Switch Inputs
Sender Inputs
Frequency Inputs
MOSFET Outputs
Flexible I/Os

Tachometer, Odometer, and Speedometer Configuration

The **Revolution Counters** tab of **DCF Editor** is used to configure the device's tachometers, odometers, and speedometers.

Tachometer Configuration

The **Tachometer 1** and **Tachometer 2** sections of the **Revolution Counters** tab configure the tachometers. These sections contain the following settings.

Setting	Description
Tach 1/2 Source	Specifies whether the source of a tachometer's data is CAN or one of the frequency inputs, or whether a tachometer is disabled. To use a frequency input as a data source, the corresponding flexible I/O must be configured as a flexible input.
Tach 1/2 Sample To Average	Specifies the number of samples to average. Valid values are 1–100.
Tach 1/2 Pulses Per Rev	Specifies the pulses-per-revolution. Valid values are 1–100.

Related information

Tachometers

Odometer Configuration

The Odometer 1 through Odometer 3 sections of the **Revolution Counters** tab configure the odometers. These sections contain the following settings.

Setting	Description
Odom 1/2/3 Source	Specifies whether the source of an odometer's data is CAN or one of the speedometers, or whether an odometer is disabled.
Odom 1/2/3 Resettable	Specifies whether an odometer is resettable or historical.

Related information

Odometers

Speedometer Configuration

The **Speedometer 1** and **Speedometer 2** sections of the **Revolution Counters** tab configure the speedometers. These sections contain the following settings.

Setting	Description
Speed 1/2 Source	Specifies whether the source of a speedometer's data is CAN or one of the frequency inputs, or whether a speedometer is disabled. To use a frequency input as a data source, the corresponding flexible I/O must be configured as a flexible input.
Speed 1/2 Samples	Specifies the number of samples to be averaged. Valid values are 0–100.
Speed 1/2 Pulses Per Km	Specifies the pulses-per-km. Valid values are 0–100.

Related information

Speedometers

Display Configuration

The **Display** tab of **DCF Editor** is used to specify the screen that displays when the application starts up and to configure the ambient light sensor, unit of measurement, and other display settings.

Application Start Screen

The **Application Start Screen** setting specifies the screen that displays when the device starts up. The valid values depend upon the application.

Application Start Sub Screen

A screen can have multiple sub-screens. The **Application Start Sub Screen** setting specifies the sub-screen that displays when the device starts up. The valid values depend upon the application and the screen specified with the **Application Start Screen** setting.

Backlight Configuration

The **Backlight** section contains the following settings.

Setting	Description
Backlight Reset Value	The default brightness. The value is a percentage of the maximum brightness. Valid values are 0–100. Users can override the default with the **Settings** screen.
Light Sensor Used	Specifies whether the ambient light sensor is used by default. Valid values are Yes and No. Users can override the default with the **Settings** screen.
ALS Step 1/2/3/4	The **ALS Step** settings provide four ALS values. When the ambient light sensor is enabled and it detects that the ambient lighting is at or below the ALS value, the device changes the brightness level to the level specified with the corresponding **Backlight Step** setting. Valid values are 0–65535. For example, if the ambient light level is at or below the **ALS Step 3** level, the device changes the brightness level to the **Backlight Step 3** level.
Backlight Step 1/2/3/4	The brightness level to apply when the ambient light level is at or below the ALS value specified by the corresponding **ALS Step** setting. Valid values are 0–100.

Display Options Configuration

The **Options** section contains the following settings.

Setting	Description
Language Selection	Specifies the default language in which the device displays text. Users can override the default with the **Settings** screen.
Units Of Measure	Specifies the device's default unit of measurement for functions such as speedometers and odometers. Valid values are Metric and Imperial. Users can override the default with the **Settings** screen.
Logo Screen Time	Specifies how long the logo is displayed after the device starts up. Valid values are 0–255s. 0 specifies that a logo is not displayed.

Related information

LCD

Icons Configuration

The **Icons** tab of **DCF Editor** is used to configure the following items:

- Custom icons' data sources.
- The conditions that activate icons that use switch inputs as data sources.

The device supports 32 custom icons, thus the tab contains 32 **Icon *n* Source** sections. Each **Icon *n* Source** section contains pairs of **Icon *n* Source** and **Icon *n* Activation Configuration** dropdown lists.

Note:

For custom icons, the manual does not describe application-specific functions such as adding icon images or performing custom calculations. For information on using and configuring icons, contact technical support.

The **Icon *n* Source** dropdown lists specify the source of an icon's data. You can specify the following sources:

- CAN
- Switch input 1 (Sw1)
- Switch input 2 (Sw2)
- Switch input 3 (Sw3)
- Switch input 4 (Sw4)
- Switch input 5 (Sw5)
- Switch input 6 (Sw6)
- Custom Calculation (CustomCalc)

Note:

The available switch inputs depend upon which flexible I/Os are configured as switch inputs on the FlexIO tab. For example, if only flexible I/Os 1 and 3 are configured as switch inputs, the **Icon *n* Source** dropdown lists will include Sw1 and Sw3.

An icon's **Icon *n* Activation Configuration** setting is enabled when the icon's data source is a switch input. The **Icon *n* Activation Configuration** settings specify the switch input condition that activates the icon. You can select from the following values:

- ActiveOpen: Activate when the switch is open.
- ActiveBPlus: Activate when switched to B+.
- ActiveGnd: Activate when switched to ground.

Related information

Icon Status Parameter

Custom Configuration

The **Custom** tab of **DCF Editor** is used to specify values for user variables. The tab contains the **CAN User Variables** section.

CAN User Variables

The device supports 100 CAN user variables. The **CAN User Variables** section contains the **User CAN Object *n*** settings, which specify values for the user variables. Valid values are 0–4294967295.

Related information

CAN User Variable Parameters

7 — CANopen Communications

The device complies with the CAN in Automation (CiA) CANopen 301 specification. This chapter describes the device's CANopen features.

Some familiarity with CANopen is a prerequisite. For CANopen information, see the CiA web site.

Byte and Bit Sequence Order

CANopen message byte sequences are transmitted with the least significant byte first (little-endian format).

Note:

This manual uses the LSB 0 Numbering convention when referring to byte and bit numbers.

For example, the following table shows an SDO that writes `0x029E` to the object with the index and sub-index `0x31B0:04`.

0	1	2	3	4	5	6	7
Control Byte	Index		Sub-index	Data			
22h	B0h	31h	04h	9Eh	02h	00h	00h

ASCII strings are read from left to right. The following example shows an SDO segment for the string "Curtis".

1	2	3	4	5	6	7	8
Control Byte	Data						
00h	43h (= "C")	75h (= "u")	72h (= "r")	74h (= "t")	69h (= "i")	73h (= "s")	00h

Bit sequences are transmitted from most significant to least significant bit (big-endian format). The following example shows how the device transmits `0x2B` in binary.

7	6	5	4	3	2	1	0
0	0	1	0	1	0	1	1

Related information

CAN Connections
CAN Features

Expedited SDOs

The device provides SDO channels for sending and receiving SDO messages. The COB-IDs of the SDO channels are specified as follows:

- Send SDO: The sum of 0x600 and the CAN channel's configured node ID.
- Receive SDO: The sum of 0x580 and the CAN channel's node ID.

For expedited SDOs, the least significant byte is known as the control byte. The following table describes the control byte's bits and their corresponding fields:

Bit	7	6	5	4	3	2	1	0
Field	*Command Specifier*			0	*n*		*e*	*s*

The following table describes the control byte fields:

Field	**Description**
Command Specifier	Indicates the SDO's transfer type. Table 16 describes valid values.
Bit 4	Must be 0.
n	0 for messages that do not contain data. For messages that contain data, specify either 0 or the number of unused data bytes.
e	1 for messages that contain data, 0 otherwise.
s	0 for messages in which *n* does not specify the number of unused bytes, 1 otherwise.

Table 16. SDO Transfer Types

Transfer Type	**Value**
Write data to a device	`001b`
Confirm a write	`011b`
Request data from a device	`010b`
Device responds with requested data	`010b`
Abort SDO	`100b`

For example, a message that writes two bytes of data can have a control byte of either `0x22` or `0x2B`.

Related information

CAN Configuration

PDOs

The device provides eight RPDOs and eight TPDOs. The PDOs are available on both CAN channels.

Each PDO is represented by a PDO communication object and a PDO mapping object that specifies the CAN objects for which data is transmitted or received. The following table describes the indexes of the PDO communication and mapping objects, as well as the PDOs' default COB-IDS:

Table 17. PDO Objects and Default COB-IDS

PDO	PDO Communication Object Index	PDO Mapping Object Index	Default COB-ID
RPDO1	1400h	1600h	0x261
RPDO2	1401h	1601h	0x361
RPDO3	1402h	1602h	0x461
RPDO4	1403h	1603h	0x561
RPDO5	1404h	1604h	0x262
RPDO6	1405h	1605h	0x362
RPDO7	1406h	1606h	0x462
RPDO8	1407h	1607h	0x562
TPDO1	1800h	1A00h	0x1E1
TPDO2	1801h	1A01h	0x2E1
TPDO3	1802h	1A02h	0x3E1
TPDO4	1803h	1A03h	0x4E1
TPDO5	1804h	1A04h	0x1E2
TPDO6	1805h	1A05h	0x2E2
TPDO7	1806h	1A06h	0x3E2
TPDO8	1807h	1A07h	0x4E2

Note:

Sub-index 0x01 of a PDO communication object specifies the PDO's COB-ID, and sub-index 0x05 specifies the PDO timeout.

The PDO communication objects are configured with the **PDO *n* Rx Settings** and **PDO *n* Tx Settings** sections of the CAN Settings tab of **DCF Editor**. The mapped CAN objects are configured with the **PDO *n* Rx Mappings** and **PDO *n* Tx Mappings** sections.

CANopen Parameters

The following list explains the data used in the parameter descriptions in the following sections.

- Parameters are referred to by their EDS names.
- Parameters' CAN indexes and EDS names are listed before parameter descriptions.
- CAN indexes and sub-indexes are described in the following format:

  `0xiiii:ss`, where *`iiii`* is the index and *`ss`* is the sub-index.
- Parameter descriptions are followed by descriptions of valid values and whether parameters are read-only or writable.

Screen Selection Parameters

The screen selection parameters specify the active screen and sub-screen.

current_screen

`0x3200:00`
`current_screen`

Specifies the active screen. Screens are identified by screen IDs. The specified screen can be a standard or custom screen.

Note:

The standard screens have IDs that cannot be changed. Custom screens have application-specific IDs.

Values	0–255 The following screen IDs identify the standard screens. 0 = **Library** screen 1 = **Home** screen 2 = **Faults** screen 3 = **Settings** screen 253 = **Video** screen
Read/Write	RW

The following example changes the active screen to the Video screen, then displays the sub-screen identified by the application's 0x01 sub-screen ID.

```
22 00 32 00 FD 00 00 00
22 01 32 00 01 00 00 00
```

current_sub_screen

`0x3201:00`
`current_sub_screen`

Specifies the sub-screen of the active screen. Sub-screens are identified by application-specific sub-screen IDs.

Values	0–255
Read/Write	RW

See the example for the current_screen parameter.

Backlight Parameters

The backlight parameters indicate whether the ambient light sensor is enabled and the backlight's brightness. The brightness can only be adjusted if the ambient light sensor is disabled.

backlight_value

`0x3210:01`
`backlight_value`

Specifies the backlight's brightness as a percentage of the maximum brightness.

Values	0–100
Read/Write	RW

The following example changes the brightness level to 50% of the maximum level.

```
22 10 32 01 32 00 00 00
```

light_sensor_used

`0x3210:02`
`light_sensor_used`

Indicates whether the ambient light sensor is disabled.

Values	0–1 0 = Disabled 1 = Enabled
Read/Write	RO

LED Parameters

The LED parameters control the multi-colored LED's colors and blinking behavior. The parameters specify the LED colors for various states and the number of blinks in a flash sequence.

Note:

A flash sequence is defined by the *`LedsOnColor`*, *`LedsOffColor`*, and *`NumBlinks`* parameters, and is initiated with the BlinkEnable parameter.

LedsColor

`0x3215:01`
`LedsColor`

Specifies the LED color and brightness when the LED is steadily on.

Bytes 4, 5, and 6 specify an RGB color's red, green, and blue primary colors, respectively. The bytes specify the primary colors' brightness, with 0xFF indicating the maximum brightness.

Values 0–0xFFFFFF
Read/Write RW

The following example specifies that the LED's RGB color is 0x800080 (purple).

```
22 15 32 01 80 00 80 00
```

LedsOnColor

`0x3215:02`
`LedsOnColor`

Specifies the LED color that indicates On when the LED is blinking, as well as the color brightness.

Bytes 4, 5, and 6 specify an RGB color's red, green, and blue primary colors, respectively. The bytes specify the primary colors' brightness, with 0xFF indicating the maximum brightness.

Values 0–0xFFFFFF
Read/Write RW

LedsOffColor

`0x3215:03`
`LedsOffColor`

Specifies the LED color that indicates Off when the LED is blinking, as well as the color brightness.

Bytes 4, 5, and 6 specify an RGB color's red, green, and blue primary colors, respectively. The bytes specify the primary colors' brightness, with 0xFF indicating the maximum brightness.

Values 0–0xFFFFFF
Read/Write RW

NumBlinks

`0x3215:04`
`NumBlinks`

Specifies the number of blinks in a flash sequence.

Values 0–100
Read/Write RW

BlinkEnable

`0x3215:05`
`BlinkEnable`

Enables a flash sequence.

1 indicates enable. The value reverts to 0 after the flash sequence finishes.

Values 0–1
Read/Write RW

The following example specifies a red-green sequence that flashes 10 times.

```
22 15 32 02 FF 00 00 00
22 15 32 03 00 FF 00 00
```

```
22 15 32 04 0A 00 00 00
22 15 32 05 01 00 00 00
```

LedsKeyswitchOffColor

`0x3215:06`
`LedsKeyswitchOffColor`

Specifies the LED color when the keyswitch is off.

Bytes 4, 5, and 6 specify an RGB color's red, green, and blue primary colors, respectively. The bytes specify the primary colors' brightness, with 0xFF indicating the maximum brightness.

Values	0 -0xFFFFFF
Read/Write	RW

The following example specifies that the LED is blue, at 50% of maximum brightness, when the keyswitch is off.

```
22 15 32 06 00 00 80 00
```

Icon Status Parameter

The device provides a parameter that indicates icon statuses. The parameter is named *`IconStatus`*.

IconStatus

`0x3300:00`
`IconStatus`

Specifies which icons are displayed.

NX7 applications support up to 32 icons. Each bit corresponds to an icon; for example, bit 0 corresponds to icon 1. When an icon's data source is configured as CAN, the icon is displayed when its corresponding bit is set to 1.

Values	0–0xFFFFFFFF
Read/Write	RW

The following example displays icons 3, 7, and 9 and hides the other icons.

```
22 00 33 00 44 01 00 00
```

Related information

Icons Configuration

Switch Input Parameters

The switch input parameters indicate the statuses of the switch inputs.

SwitchInputValue1

`0x3350:01`
`SwitchInputValue1`

Indicates the status of switch 1.

Values 0–3
0 = Open (high impedance)
1 = Switched to ground
2 = Switched to B+
3 = The pin is not configured as a switch input

Read/Write RO

SwitchInputValue2

`0x3350:02`
`SwitchInputValue2`

Indicates the status of switch 2.

Values 0–3
0 = Open (high impedance)
1 = Switched to ground
2 = Switched to B+
3 = The pin is not configured as a switch input

Read/Write RO

SwitchInputValue3

`0x3350:03`
`SwitchInputValue3`

Indicates the status of switch 3.

Values 0–3
0 = Open (high impedance)
1 = Switched to ground
2 = Switched to B+
3 = The pin is not configured as a switch input

Read/Write RO

SwitchInputValue4

`0x3350:04`
`SwitchInputValue4`

Indicates the status of switch 4.

Values 0–3
0 = Open (high impedance)
1 = Switched to ground
2 = Switched to B+
3 = The pin is not configured as a switch input

Read/Write RO

SwitchInputValue5

`0x3350:05`
`SwitchInputValue5`

Indicates the status of switch 5.

Values	0–3 0 = Open (high impedance) 1 = Switched to ground 2 = Switched to B+ 3 = The pin is not configured as a switch input
Read/Write	RO

SwitchInputValue6

`0x3350:06`
`SwitchInputValue6`

Indicates the status of switch 6.

Values	0–3 0 = Open (high impedance) 1 = Switched to ground 2 = Switched to B+ 3 = The pin is not configured as a switch input
Read/Write	RO

Related information

Switch Inputs

Sender Input Parameters

The sender input parameters indicate the voltages and resistances of the sender inputs. Since these inputs can also be configured as digital outputs, there also are parameters that specify digital output statuses.

flexio1_voltage

`0x3179:02`
`flexio1_voltage`

Indicates the voltage of sender input 1.

Values	0–65535mV
Read/Write	RO

flexio1_resistance

`0x3179:03`
`flexio1_resistance`

Indicates the resistance of sender input 1.

Values	0–65535Ω
Read/Write	RO

flexio1_output

`0x3179:04`
`flexio1_output`

Specifies the status of digital output 1.

Values 0–1
Read/Write RW

The following example sets the output's status to 1.

```
22 79 31 04 01 00 00 00
```

flexio2_voltage

`0x317A:02`
`flexio2_voltage`

Indicates the voltage of sender input 2.

Values 0–65535mV
Read/Write RO

flexio2_resistance

`0x317A:03`
`flexio2_resistance`

Indicates the resistance of sender input 2.

Values 0–65535Ω
Read/Write RO

flexio2_output

`0x317A:04`
`flexio2_output`

Specifies the status of digital output 2.

Values 0–1
Read/Write RW

The following example sets the output's status to 1.

```
22 7A 31 04 01 00 00 00
```

flexio3_voltage

`0x317B:02`
`flexio3_voltage`

Indicates the voltage of sender input 3.

Values 0–65535mV
Read/Write RO

flexio3_resistance

`0x317B:03`
`flexio3_resistance`

Indicates the resistance of sender input 3.

Values 0–65535Ω
Read/Write RO

flexio3_output

`0x317B:04`
`flexio3_output`

Specifies the status of digital output 3.

Values	0–1
Read/Write	RW

The following example sets the output's status to 0.

```
22 7B 31 04 00 00 00 00
```

flexio4_voltage

`0x317C:02`
`flexio4_voltage`

Indicates the voltage of sender input 4.

Values	0–65535mV
Read/Write	RO

flexio4_resistance

`0x317C:03`
`flexio4_resistance`

Indicates the resistance of sender input 4.

Values	0–65535Ω
Read/Write	RO

flexio4_output

`0x317C:04`
`flexio4_output`

Specifies the status of digital output 4.

Values	0–1
Read/Write	RW

The following example sets the output's status to 1.

```
22 7C 31 04 01 00 00 00
```

Related information

Sender Inputs

Frequency Input Parameters

The frequency input parameters indicate the inputs' frequencies.

frequency_value1

`0x3403:01`
`frequency_value1`

Indicates the frequency of frequency input 1.

Values	0–4294967295µs
Read/Write	RO

frequency_value2

`0x3403:02`
`frequency_value2`

Indicates the frequency of frequency input 2.

Values	0–4294967295µs
Read/Write	RO

Related information
Frequency Inputs

MOSFET Output Parameters

The MOSFET parameters specify data for the MOSFET outputs. The parameter that applies to an output depends upon the output's data source.

Mosfet1Pwm

`0x3375:02`
`Mosfet1Pwm`

Specifies the PWM duty cycle for MOSFET 1 when the output is configured as a PWM output.

Values	0–100%
Read/Write	RW

The following example sets the duty cycle to 75%.

```
22 75 33 02 4B 00 00 00
```

Mosfet1SwitchOutput

`0x3375:03`
`Mosfet1SwitchOutput`

Indicates the state of MOSFET 1 when the output is configured as a digital output. 1 indicates the output is on.

Values	0–1
Read/Write	RW

The following example turns on MOSFET 1 when the output is configured as a digital output.

```
22 75 33 03 01 00 00 00
```

Related information
MOSFET Outputs

Hour Meter Parameters

Each hour meter has parameters for the following functions:

- Specify the hour meter's data source.
- Enable or disable the hour meter.
- Specify the hour meter value when the hour meter's data source is configured as CAN.
- Overwrite the hour meter value when the hour meter's data source is not configured as CAN.
- Reset the hour meter.

Related information

Hour Meters
Hour Meter Configuration

Hour Meter 1 Parameters

The hour meter 1 parameters enable, disable, and reset hour meter 1, and specify its data source and value. The parameter used to specify the value depends upon the hour meter's data source.

Hour1Source

`0x3170:01`
`Hour1Source`

Specifies the data source for hour meter 1.

Values	0–14 0 = CAN 1 = EnableByCAN 2 = Switch 1 ground 3 = Switch 1 B+ 4 = Switch 2 ground 5 = Switch 2 B+ 6 = Switch 3 ground 7 = Switch 3 B+ 8 = Switch 4 ground 9 = Switch 4 B+ 10 = Switch 5 ground 11 = Switch 5 B+ 12 = Switch 6 ground 13 = Switch 6 B+ 14 = Invalid
Read/Write	RW

Hour1Enable

`0x3172:01`
`Hour1Enable`

Enables or disables hour meter 1 if the hour meter's source is EnableByCAN.

Values	0–1 0 = Disables the hour meter. 1 = Enables the hour meter.
Read/Write	RW

The following example enables hour meter 1.

```
22 72 31 01 01 00 00 00
```

HourMeter1Value

`0x3174:01`
`HourMeter1Value`

Specifies the value for hour meter 1 if the hour meter's source is CAN.

Values	0–3600000000s
Read/Write	RW

The following example sets hour meter 1 to 86,400 seconds.

```
22 74 31 01 80 51 01 00
```

ResetHourMeter1

`0x3176:01`
`ResetHourMeter1`

Resets hour meter 1 if it is resettable.

Values	0–1 0 = Resetting is disabled. 1 = Resets the hour meter. After the reset, the value reverts to 0.
Read/Write	RW

The following example resets hour meter 1.

```
22 76 31 01 01 00 00 00
```

HourMeter1OverwriteEnable

`0x3178:01`
`HourMeter1OverwriteEnable`

Enables overwriting the hour meter 1 value if the hour meter's source is other than CAN. Set this parameter to 1 before overwriting; see the example for Hour1SetManualVal.

Values	0–1 0 = Resetting is disabled. 1 = Resets the hour meter. After the reset, the value reverts to 0.
Read/Write	RW

Hour1SetManualVal

`0x3175:01`
`Hour1SetManualVal`

Overwrites the hour meter 1 value if the hour meter's source is other than CAN. Enable overwriting with HourMeter1OverwriteEnable before setting the value.

Values	0–3600000000s
Read/Write	RW

The following example overwrites the hour meter 1 value.

```
22 78 31 01 01 00 00 00
22 75 31 01 AE 40 01 00
```

Hour Meter 2 Parameters

The hour meter 2 parameters enable, disable, and reset hour meter 2, and specify its data source and value. The parameter used to specify the value depends upon the hour meter's data source.

Hour2Source

`0x3170:02`
`Hour2Source`

Specifies the data source for hour meter 2.

Values	0–14 0 = CAN 1 = EnableByCAN 2 = Switch 1 ground 3 = Switch 1 B+ 4 = Switch 2 ground 5 = Switch 2 B+ 6 = Switch 3 ground 7 = Switch 3 B+ 8 = Switch 4 ground 9 = Switch 4 B+ 10 = Switch 5 ground 11 = Switch 5 B+ 12 = Switch 6 ground 13 = Switch 6 B+ 14 = Invalid
Read/Write	RW

Hour2Enable

`0x3172:02`
`Hour2Enable`

Enables or disables hour meter 2 if the hour meter's data source is EnableByCAN.

Values	0–1 0 = Disables hour meter 2. 1 = Enables hour meter 2.
Read/Write	RW

The following example enables hour meter 2.

```
22 72 31 02 01 00 00 00
```

HourMeter2Value

`0x3174:02`

`HourMeter2Value`

Specifies the value for hour meter 2 if the hour meter's data source is CAN.

Values	0–3600000000s
Read/Write	RW

The following example sets hour meter 2 to 3600 seconds.

```
22 74 31 02 10 0E 00 00
```

ResetHourMeter2

`0x3176:02`

`ResetHourMeter2`

Resets hour meter 2 if it is resettable.

Values	0–1 0 = Resetting is disabled. 1 = Resets the hour meter. After the reset, the value reverts to 0.
Read/Write	RW

The following example resets hour meter 2.

```
22 76 31 02 01 00 00 00
```

HourMeter2OverwriteEnable

`0x3178:02`

`HourMeter2OverwriteEnable`

Enables overwriting the hour meter 2 value if the hour meter's source is other than CAN. Set this object to 1 before overwriting; see the example for Hour2SetManualVal.

Values	0–1 0 = Overwriting is disabled. 1 = Enables overwriting. The value reverts to 0 after the hour meter value is overwritten.
Read/Write	RW

Hour2SetManualVal

`0x3175:02`

`Hour2SetManualVal`

Overwrites the hour meter 2 value if the hour meter's source is other than CAN. Enable overwriting with HourMeter2OverwriteEnable before setting the value.

Values	0–3600000000s
Read/Write	RW

The following example overwrites the hour meter 2 value.

```
22 78 31 02 01 00 00 00
22 75 31 02 01 30 43 00
```

Hour Meter 3 Parameters

The hour meter 3 parameters enable, disable, and reset hour meter 3, and specify its data source and value. The parameter used to specify the value depends upon the hour meter's data source.

Hour3Source

`0x3170:03`
`Hour3Source`

Specifies the data source for hour meter 3.

Values 0–14
0 = CAN
1 = EnableByCAN
2 = Switch 1 ground
3 = Switch 1 B+
4 = Switch 2 ground
5 = Switch 2 B+
6 = Switch 3 ground
7 = Switch 3 B+
8 = Switch 4 ground
9 = Switch 4 B+
10 = Switch 5 ground
11 = Switch 5 B+
12 = Switch 6 ground
13 = Switch 6 B+
14 = Invalid

Read/Write RW

Hour3Enable

`0x3172:03`
`Hour3Enable`

Enables or disables hour meter 3 if the hour meter's data source is EnableByCAN.

Values 0–1
0 = Hour meter 3 is disabled.
1 = Hour meter 3 is enabled.

Read/Write RW

The following example disables hour meter 3.

```
22 72 31 03 00 00 00 00
```

HourMeter3Value

`0x3174:03`
`HourMeter3Value`

Specifies the value for hour meter 3 if the hour meter's data source is CAN.

Values 0–3600000000s

Read/Write RW

The following example sets hour meter 3 to 3600 seconds.

```
22 74 31 03 10 0E 00 00
```

ResetHourMeter3

`0x3176:03`
`ResetHourMeter3`

Resets hour meter 3 if it is resettable.

Values	0–1 0 = Resetting is disabled. 1 = Resets the hour meter. After the reset, the value reverts to 0.
Read/Write	RW

The following example resets hour meter 3.

```
22 76 31 03 01 00 00 00
```

HourMeter3OverwriteEnable

`0x3178:03`
`HourMeter3OverwriteEnable`

Enables overwriting the hour meter 3 value if the hour meter's source is other than CAN. Set this object to 1 before overwriting; see the example for Hour3SetManualVal.

Values	0–1 0 = Overwriting is disabled. 1 = Enables overwriting. The value reverts to 0 after the hour meter value is overwritten.
Read/Write	RW

Hour3SetManualVal

`0x3175:03`
`Hour3SetManualVal`

Overwrites the hour meter 3 value if the hour meter's source is other than CAN. Enable overwriting with HourMeter3OverwriteEnable before setting the value.

Values	0–3600000000s
Read/Write	RW

The following example overwrites the hour meter 3 value.

```
22 78 31 03 01 00 00 00
22 75 31 03 81 22 00 00
```

Hour Meter 4 Parameters

The hour meter 4 parameters enable, disable, and reset hour meter 4, and specify its data source and value. The parameter used to specify the value depends upon the hour meter's data source.

Hour4Source

`0x3170:04`
`Hour4Source`

Specifies the data source for hour meter 4.

Values 0–14
0 = CAN
1 = EnableByCAN
2 = Switch 1 ground
3 = Switch 1 B+
4 = Switch 2 ground
5 = Switch 2 B+
6 = Switch 3 ground
7 = Switch 3 B+
8 = Switch 4 ground
9 = Switch 4 B+
10 = Switch 5 ground
11 = Switch 5 B+
12 = Switch 6 ground
13 = Switch 6 B+
14 = Invalid

Read/Write RW

Hour4Enable

`0x3172:04`
`Hour4Enable`

Enables or disables hour meter 4 if the hour meter's data source is EnableByCAN.

Values 0–1
0 = Disables hour meter 4.
1 = Enables hour meter 4.

Read/Write RW

The following example disables hour meter 4.

```
22 72 31 04 00 00 00 00
```

HourMeter4Value

`0x3174:04`
`HourMeter4Value`

Specifies the value for hour meter 4 if the hour meter's data source is CAN.

Values 0–3600000000s

Read/Write RW

The following example sets hour meter 4 to 604,800 seconds.

```
22 74 31 04 80 3A 09 00
```

ResetHourMeter4

`0x3176:04`
`ResetHourMeter4`

Resets hour meter 4 if it is resettable.

Values	0–1 0 = Resetting is disabled. 1 = Resets the hour meter. After the reset, the value reverts to 0.
Read/Write	RW

The following example resets hour meter 4.

```
22 76 31 04 01 00 00 00
```

HourMeter4OverwriteEnable

`0x3178:04`
`HourMeter4OverwriteEnable`

Enables overwriting the hour meter 4 value if the hour meter's source is other than CAN. Set this object to 1 before overwriting; see the example for Hour4SetManualVal.

Values	0–1 0 = Overwriting is disabled. 1 = Enables overwriting. The value reverts to 0 after the hour meter value is overwritten.
Read/Write	RW

Hour4SetManualVal

`0x3175:04`
`Hour4SetManualVal`

Overwrites the hour meter 4 value if the hour meter's source is other than CAN. Enable overwriting with HourMeter4OverwriteEnable before setting the value.

Values	0–3600000000s
Read/Write	RW

The following example overwrites the hour meter 4 value.

```
22 78 31 04 01 00 00 00
22 75 31 04 22 88 28 00
```

Maintenance Monitor Parameters

Each maintenance monitor has parameters for the following functions:

- Specify the maintenance monitor's data source.
- Enable or disable the maintenance monitor.
- Specify the maintenance monitor value.
- Reset the maintenance monitor.

Related information

Maintenance Monitors
Maintenance Monitor Configuration

Maintenance Monitor 1 Parameters

The maintenance monitor 1 parameters enable, disable, and reset maintenance monitor 1, and specify its data source and value. The object used to specify the value depends upon the monitor's data source.

MaintMon1Source

`0x316A:01`
`MaintMon1Source`

Specifies the data source for maintenance monitor 1.

Values 0–14
0 = CAN
1 = EnableByCAN
2 = Switch 1 ground
3 = Switch 1 B+
4 = Switch 2 ground
5 = Switch 2 B+
6 = Switch 3 ground
7 = Switch 3 B+
8 = Switch 4 ground
9 = Switch 4 B+
10 = Switch 5 ground
11 = Switch 5 B+
12 = Switch 6 ground
13 = Switch 6 B+
14 = Invalid

Read/Write RW

MaintMon1Enable

`0x3166:01`
`MaintMon1Enable`

Enables maintenance monitor 1 when its data source is EnableByCAN.

Values 0–1
0 = Disables maintenance monitor 1.
1 = Enables maintenance monitor 1.

Read/Write RW

The following example enables maintenance monitor 1.

```
22 66 31 01 01 00 00 00
```

MaintMon1Value

`0x3164:01`
`MaintMon1Value`

Specifies the maintenance monitor 1 value if the monitor's data source is CAN.

Values 0–4294965600s

Read/Write RW

The following example sets maintenance monitor 1 to 360,000 seconds (100 hours).

```
22 64 31 01 40 7E 05 00
```

ResetMaintMon1

`0x316C:01`
`ResetMaintMon1`

Resets maintenance monitor 1 when it is resettable.

Values	0–1 0 = Resetting is disabled. 1 = Resets the maintenance monitor.
Read/Write	RW

The following example resets maintenance monitor 1.

```
22 6C 31 01 01 00 00 00
```

MaintMon1OverwriteEnable

`0x316D:01`
`MaintMon1OverwriteEnable`

Enables overwriting the maintenance monitor 1 value if the monitor's data source is other than CAN. Set this parameter to 1 before overwriting; see the example for MaintMon1SetManualVal.

Values	0–1 0 = Overwriting is disabled. 1 = Enables overwriting. The value reverts to 0 after the maintenance monitor value is overwritten.
Read/Write	RW

MaintMon1SetManualVal

`0x316B:01`
`MaintMon1SetManualVal`

Overwrites the maintenance monitor 1 value if the monitor's data source is other than CAN. Before overwriting, set *`MaintMon1OverwriteEnable`* to 1.

Values	0–4294965600s
Read/Write	RW

The following example sets the maintenance monitor 1 value to 360,000 seconds (100 hours).

```
22 6D 31 01 01 00 00 00
22 6B 31 01 40 7E 05 00
```

Maintenance Monitor 2 Parameters

The maintenance monitor 2 parameters enable, disable, and reset maintenance monitor 2, and specify its data source and value. The object used to specify the value depends upon the monitor's data source.

MaintMon2Source

`MaintMon2Source`

Specifies the data source for maintenance monitor 2.

Values 0–14
0 = CAN
1 = EnableByCAN
2 = Switch 1 ground
3 = Switch 1 B+
4 = Switch 2 ground
5 = Switch 2 B+
6 = Switch 3 ground
7 = Switch 3 B+
8 = Switch 4 ground
9 = Switch 4 B+
10 = Switch 5 ground
11 = Switch 5 B+
12 = Switch 6 ground
13 = Switch 6 B+
14 = Invalid

Read/Write RW

MaintMon2Enable

`0x3166:02`
`MaintMon2Enable`

Enables maintenance monitor 2 when its data source is EnableByCAN.

Values 0–1
0 = Disables maintenance monitor 2.
1 = Enables maintenance monitor 2.

Read/Write RW

The following example enables maintenance monitor 2.

```
22 66 31 02 01 00 00 00
```

MaintMon2Value

`0x3164:02`
`MaintMon2Value`

Specifies the maintenance monitor 2 value if the monitor's data source is CAN.

Values 0–4294965600s

Read/Write RW

The following example sets maintenance monitor 2 to 360,000 seconds (100 hours).

```
22 64 31 02 40 7E 05 00
```

ResetMaintMon2

`0x316C:02`
`ResetMaintMon2`

Resets maintenance monitor 2 when it is resettable.

Values	0–1 0 = Resetting is disabled. 1 = Resets the maintenance monitor.
Read/Write	RW

The following example resets maintenance monitor 2.

```
22 6C 31 02 01 00 00 00
```

MaintMon2OverwriteEnable

`0x316D:02`
`MaintMon2OverwriteEnable`

Enables overwriting the maintenance monitor 2 value if the monitor's data source is other than CAN. Set this object to 1 before overwriting; see the example for MaintMon2SetManualVal.

Values	0–1 0 = Overwriting is disabled. 1 = Enables overwriting. The value reverts to 0 after the maintenance monitor value is overwritten.
Read/Write	RW

MaintMon2SetManualVal

`0x316B:02`
`MaintMon2SetManualVal`

Overwrites the maintenance monitor 2 value if the monitor's data source is other than CAN. Before overwriting, set *`MaintMon2OverwriteEnable`* to 1.

Values	0–4294965600s
Read/Write	RW

The following example sets the maintenance monitor 2 value to 360,000 seconds (100 hours).

```
22 6D 31 02 01 00 00 00
22 6B 31 02 40 7E 05 00
```

Maintenance Monitor 3 Parameters

The maintenance monitor 3 parameters enable, disable, and reset maintenance monitor 3, and specify its data source and value. The object used to specify the value depends upon the monitor's data source.

MaintMon3Source

`0x316A:03`
`MaintMon3Source`

Values 0–14
0 = CAN
1 = EnableByCAN
2 = Switch 1 ground
3 = Switch 1 B+
4 = Switch 2 ground
5 = Switch 2 B+
6 = Switch 3 ground
7 = Switch 3 B+
8 = Switch 4 ground
9 = Switch 4 B+
10 = Switch 5 ground
11 = Switch 5 B+
12 = Switch 6 ground
13 = Switch 6 B+
14 = Invalid

Read/Write RW

MaintMon3Enable

`0x3166:03`
`MaintMon3Enable`

Enables maintenance monitor 3 when its data source is EnableByCAN.

Values 0–1
0 = Disables maintenance monitor 3.
1 = Enables maintenance monitor 3.

Read/Write RW

The following example enables maintenance monitor 3.

```
22 66 31 03 01 00 00 00
```

MaintMon3Value

`0x3164:03`
`MaintMon3Value`

Specifies the maintenance monitor 3 value if the monitor's data source is CAN.

Values 0–4294965600s

Read/Write RW

The following example sets maintenance monitor 3 to 360,000 seconds (100 hours).

```
22 64 31 03 40 7E 05 00
```

ResetMaintMon3

`0x316C:03`
`ResetMaintMon3`

Resets maintenance monitor 3 when it is resettable.

Values	0–1 0 = Resetting is disabled. 1 = Resets the maintenance monitor.
Read/Write	RW

The following example resets maintenance monitor 3.

```
22 6C 31 03 01 00 00 00
```

MaintMon3OverwriteEnable

`0x316D:03`
`MaintMon3OverwriteEnable`

Enables overwriting the maintenance monitor 3 value if the monitor's data source is other than CAN. Set this object to 1 before overwriting; see the example for MaintMon3SetManualVal.

Values	0–1 0 = Overwriting is disabled. 1 = Enables overwriting. The value reverts to 0 after the maintenance monitor value is overwritten.
Read/Write	RW

MaintMon3SetManualVal

`0x316B:03`
`MaintMon3SetManualVal`

Overwrites the maintenance monitor 3 value if the monitor's data source is other than CAN. Before overwriting, set *`MaintMon3OverwriteEnable`* to 1.

Values	0–4294965600s
Read/Write	RW

The following example sets the maintenance monitor 3 value to 360,000 seconds (100 hours).

```
22 6D 31 03 01 00 00 00
22 6B 31 03 40 7E 05 00
```

Speedometer Parameters

Each speedometer has parameters that specify its data source and the current speed.

Note:

The **Units of Measure** setting specifies whether the NX7 displays speedometer and odometer data in miles-per-hour (MPH) or kilometers-per-hour (km/h). The unit of measurement can be specified with **DCF Editor** or the **Settings** screen.

speed1_source

`0x31D0:02`
`speed1_source`

Specifies the data source for speedometer 1.

Values 0–3
0 = CAN
1 = Frequency 1 input

Note:
Flexible I/O 1 must be configured as a frequency input.

2 = Frequency 2 input

Note:
Flexible I/O 2 must be configured as a frequency input.

3 = Disabled

Read/Write RO

speed1_kph

`0x31D0:01`
`speed1_kph`

Specifies the speed for speedometer 1.

Values 0–65635 km/h
Read/Write RW (Values are read-only if the speedometer's data source is not CAN.)

The following example sets speedometer 1 to 50 km/h.

```
22 D0 31 01 32 00 00 00
```

speed2_source

`0x31D1:02`
`speed2_source`

Specifies the data source for speedometer 2.

Values 0–3
0 = CAN
1 = Frequency 1 input

Note:
Flexible I/O 1 must be configured as a frequency input.

2 = Frequency 2 input

Note:
Flexible I/O 2 must be configured as a frequency input.

3 = Disabled

Read/Write RO

speed2_kph

`0x31D1:01`
`speed2_kph`

Specifies the speed for speedometer 2.

Values 0–65635 km/h
Read/Write RW (Values are read-only if the speedometer's data source is not CAN.)

The following example sets speedometer 2 to 35 km/h.

```
22 D1 31 01 23 00 00 00
```

Related information

Speedometers
Speedometer Configuration

Odometer Parameters

Each odometer has parameters that specify the odometer's value and data source.

Note:

The **Units of Measure** setting specifies whether the NX7 displays speedometer and odometer data in miles-per-hour (MPH) or kilometers-per-hour (km/h). The unit of measurement can be specified with **DCF Editor** or the **Settings** screen.

odom1_source

`0x31C0:01`
`odom1_source`

Specifies the data source for odometer 1.

Values	0–3 0 = CAN 1 = Calculate using the speedometer 1 data. 2 = Calculate using the speedometer 2 data. 3 = Disabled
Read/Write	RO

odom1_km

`0x31C0:02`
`odom1_km`

Specifies the value for odometer 1.

Values	0.0–999,999.9km, in increments of 0.1km.
Read/Write	RW (Values are read-only if the odometer's data source is not CAN.)

The following example sets odometer 1 to 3.2km.

```
22 C0 31 02 20 00 00 00
```

odom2_source

`0x31C1:01`
`odom2_source`

Specifies the data source for odometer 2.

Values	0–3 0 = CAN 1 = Calculate using the speedometer 1 data. 2 = Calculate using the speedometer 2 data. 3 = Disabled
Read/Write	RO

odom2_km

`0x31C1:02`
`odom2_km`

Specifies the value for odometer 2.

Values	0.0–999,999.9km, in increments of 0.1km.
Read/Write	RW (Values are read-only if the odometer's data source is not CAN.)

The following example sets odometer 2 to 100.3km.

```
22 C1 31 02 EB 03 00 00
```

odom3_source

`0x31C2:01`
`odom3_source`

Specifies the data source for odometer 3.

Values	0–3 0 = CAN 1 = Calculate using the speedometer 1 data. 2 = Calculate using the speedometer 2 data. 3 = Disabled
Read/Write	RO

odom3_km

`0x31C2:02`
`odom3_km`

Specifies the value for odometer 3.

Values	0.0–999,999.9km, in increments of 0.1km.
Read/Write	RW (Values are read-only if the odometer's data source is not CAN.)

The following example sets odometer 3 to 376.8km.

```
22 C2 31 02 B8 0E 00 00
```

Tachometer Parameters

Each tachometer has parameters that specify the tachometer's value and data source.

tach1_source

`0x31B0:01`
`tach1_source`

Specifies the data source for tachometer 1.

Values 0–3
0 = CAN
1 = Frequency 1 input

Note:
Flexible I/O 1 must be configured as a frequency input.

2 = Frequency 2 input

Note:
Flexible I/O 2 must be configured as a frequency input.

Read/Write RO

tach1_value

`0x31B0:04`
`tach1_value`

Specifies the value for tachometer 1.

Values 0–65,535 RPM
Read/Write RW (Values are read-only if the tachometer's data source is not CAN.)

The following example sets tachometer 1 to 4000 RPM.

```
22 B0 31 04 A0 0F 00 00
```

tach2_source

`0x31B1:01`
`tach2_source`

Specifies the data source for tachometer 2.

Values 0–3
0 = CAN
1 = Frequency 1 input

Note:
Flexible I/O 1 must be configured as a frequency input.

2 = Frequency 2 input

Note:
Flexible I/O 2 must be configured as a frequency input.

Read/Write RO

tach2_value

`0x31B1:04`
`tach2_value`

Specifies the value for tachometer 2.

Values 0–65,535 RPM
Read/Write RW (Values are read-only if the tachometer's data source is not CAN.)

The following example sets tachometer 2 to 670 RPM.

```
22 B1 31 04 9E 02 00 00
```

Related information

Tachometers
Tachometer Configuration

Real-Time Clock Parameters

The real-time-clock parameters specify the clock's time and date.

RTC_hours_minutes_seconds

`0x3700:01`
`RTC_hours_minutes_seconds`

Indicates the clock's time to the second.

Values The time in the *hh mm ss* format. The value consists of three bytes:
- *hh* is the least significant byte. Valid values are 0–23.
- *mm* is the second byte. Valid values are 0–59.
- *dd* is the most significant byte. Valid values are 0–59.

Read/Write RO

The following example reads the time.

```
40 00 37 01 00 00 00 00
```

RTC_set_hours_minutes_seconds

`0x3700:03`
`RTC_set_hours_minutes_seconds`

Specifies the clock's time to the second. After specifying the time, activate it by setting RTC_set_time_command to 1.

Values The time in the *hh mm ss* format. The value consists of three bytes:
- *hh* is the least significant byte. Valid values are 0–23.
- *mm* is the second byte. Valid values are 0–59.
- *dd* is the most significant byte. Valid values are 0–59.

Read/Write RW

The following example sets the time to 13:05:06

```
22 00 37 03 0D 05 06 00
22 00 37 05 01 00 00 00
```

RTC_set_time_command

`0x3700:05`
`RTC_set_time_command`

Activates the time specified by RTC_set_hours_minutes_seconds. The device displays the specified time after *`RTC_set_time_command`* is set to 1.

Values 0–1. 1 activates the specified time.

Read/Write RW

RTC_year_month_day

`0x3700:02`

`RTC_year_month_day`

Indicates the date.

Values The date in the *yy mm dd* format. The value consists of three bytes:
- *yy* is the least significant byte. The years count from 1900. For example, 0x7C indicates 2024.
- *mm* is the second byte. Valid values are 1–12.
- *dd* is the most significant byte. Valid values are values are 1–31.

Read/Write RO

The following example reads the date.

```
40 00 37 02 00 00 00 00
```

RTC_set_year_month_day

`0x3700:04`

`RTC_set_year_month_day`

Specifies the date. After specifying the date, activate it by setting RTC_set_date_command to 1.

Values The date in the *yy mm dd* format. The value consists of three bytes:
- *yy* is the least significant byte. The years count from 1900. For example, 0x7C indicates 2024.
- *mm* is the second byte. Valid values are 1–12.
- *dd* is the most significant byte. Valid values are values are 1–31.

Read/Write RW

The following example sets the date to January 24, 2024.

```
22 00 37 04 7C 01 18 00
22 00 37 06 01 00 00 00
```

RTC_set_date_command

`0x3700:06`

`RTC_set_date_command`

Activates the date specified by RTC_set_year_month_day. The device displays the specified date after *`RTC_set_date_command`* is set to 1.

Values 0–1. 1 activates the specified date.

Read/Write RW

Related information

Real-Time Clock

Audible Alarm Parameter

alarm_command

`0x3408:01`

`alarm_command`

Sounds and silences the alarm. The parameter specifies the number of beeps or a continuous beep.

Values 0–255
The value specifies the number of beeps, and decrements after each beep until the value reaches 0.
0xFF specifies a continuous beep. The value does not decrement. To stop the beeping, specify 0.

BDI Percentage Parameter

BDI_percent

`0x31A0:00`
`BDI_percent`

Specifies the BDI percentage, which is a measurement of the battery's state-of-charge.

Values 0–1000. The values scale to 0–100%.
Read/Write RW

The following example specifies that the state-of-charge is 90%.

```
22 A0 31 00 84 03 00 00
```

Related information
Battery Discharge Indicator (BDI)

Custom Value Parameters

The device provides three custom values that are saved to EEPROM. The Custom Value parameters specify the custom values' data.

The following table describes the parameters.

Item	Description
CAN index and sub-index	The index of all custom value parameters is `0x3800`. Parameter sub-indexes consist of the numbers that identify the corresponding custom values. For example, the index and sub-index of custom value 5 is `0x3800:05`.
EDS Name	The EDS names are *`custom`**`n`***, where ***`n`*** is the number that identifies the custom value. For example, the EDS name of custom value 2 is *`custom2`*.
Values	–2147483648 through 2147483647
Read / Write	RW

CAN User Variable Parameters

The device provides 100 CAN user variables. The CAN user variables specify the data stored by the variables.

The following table describes the parameters.

Item	Description
CAN index and sub-index	The index of all CAN user variable parameters is `0x4400`. Parameter sub-indexes consist of the numbers that identify the corresponding user variables. For example, the index and sub-index of user variable 1 is `0x4400:01`.
EDS Name	The EDS names are *`p_user_`**n***, where ***n*** is the number that identifies the user variable. For example, the EDS name of user variable 1 is *`p_user_1`*.
Values	0–4294967295
Read / Write	RW

Related information

Custom Configuration

J1939 Parameters

The device provides parameters for the following J1939 functions:

- Rx messages: PGNs, bitmasks, and bit shifts.
- Tx messages: PGNs.

The following topics describe these parameters.

Related information

J1939
J1939 Configuration

J1939 Rx PGN Parameters

The device provides ten J1939 parameters for receive (Rx) message PGNs. The parameters correspond to the J1939 custom Rx variables. These variables are configured with the **J1939 Custom Rx *n*** settings, which are contained by the **Custom J1939 Receive PGN** section of the **J1939** tab in **DCF Editor**.

The following table describes the parameters.

Item	Description
CAN index and sub-index	The index of all J1939 Rx PGN parameters is `0x4500`. Parameter sub-indexes consist of the numbers that identify the corresponding **J1939 Custom Rx *n*** settings. For example, the index and sub-index of the parameter corresponding to the **J1939 Custom Rx 3** setting is `0x4500:03`.
EDS Name	The EDS names are *`J1939_CustomRxPGN_`**n***, where ***n*** is the number that identifies the J1939 Custom Rx variable. For example, the EDS name of the parameter corresponding to the **J1939 Custom Rx 1** setting is *`J1939_CustomRxPGN_1`*.
Values	0–4294967295
Read / Write	RO

J1939 Rx Bitmask Parameters

The device provides ten J1939 parameters for receive (Rx) message bitmasks. The parameters correspond to the **DCF Editor** settings in the **Custom J1939 Receive Filters** section of the **J1939 Settings** tab.

The following table describes the parameters:

Item	Description
CAN index and sub-index	The index of all J1939 Rx bitmask parameters is `0x4501`. Parameter sub-indexes consist of the numbers that identify the corresponding **J1939 Filter Rx *n*** settings. For example, the index and sub-index of the parameter corresponding to the **J1939 Filter Rx 2** setting is `0x4501:02`.
EDS Name	The EDS names are *`J1939_CustomRxFilter_`**n***, where ***n*** is the number that identifies the corresponding **J1939 Filter Rx *n*** setting. For example, the EDS name that corresponds to the **J1939 Filter Rx 2** setting is *`J1939_CustomRxFilter_2`*.
Values	1–65535
Read / Write	RO

Related information

J1939 Rx Bit Shift Parameters

J1939 Rx Bit Shift Parameters

The device provides ten J1939 parameters for bit shifts of receive (Rx) message data. The parameters correspond to the **DCF Editor** settings in the **Custom J1939 Receive Shift** section of the **J1939 Settings** tab.

The following table describes the parameters:

Item	Description
CAN index and sub-index	The index of all J1939 Rx bit shift parameters is `0x4502`. Parameter sub-indexes consist of the numbers that identify the corresponding **J1939 Shift Rx *n*** settings. For example, the index and sub-index of the parameter corresponding to the **J1939 Shift Rx 3** setting is `0x4502:03`.
EDS Name	The EDS names are *`J1939_CustomRxShift_`**n***, where ***n*** is the number that identifies the corresponding **J1939 Shift Rx *n*** settings. For example, the EDS name that corresponds to the **J1939 Shift Rx 3** setting is *`J1939_CustomRxFilter_3`*.
Values	0–64
Read / Write	RO

Related information

J1939 Rx Bitmask Parameters

J1939 Tx PGN Parameters

The device provides ten J1939 parameters for transmit (Tx) message PGNs. The J1939 Tx parameters store data for the corresponding J1939 custom variables. These variables are configured with the **J1939 Custom Tx *n*** settings, which are contained by the **Custom Transmit J1939 PGNs** section of the **J1939** tab in **DCF Editor**.

The following table describes the parameters.

Item	Description
CAN index and sub-index	The index of all J1939 Tx parameters is `0x4503`. Parameter sub-indexes consist of the numbers that identify the corresponding **J1939 Tx *n*** settings. For example, the index and sub-index of the parameter corresponding to the **J1939 Tx 3** setting is `0x4503:03`.
EDS Name	The EDS names are *`J1939_CustomTxPGN_`**n***, where ***n*** is the number that identifies the corresponding **J1939 Tx *n*** setting. For example, the EDS name of the parameter corresponding to the **J1939 Custom Tx 3** setting is *`J1939_CustomTxPGN_3`*.

Item	Description
Values	1–4294967295
Read / Write	RO

Device Identity Parameters

The parameters described in the following sections indicate the device's hardware and firmware data.

canopen_mandatory_identity_vendor_id

`0x1018:01`
`canopen_mandatory_identity_vendor_id`

Indicates the vendor ID, which is 0x4349 for Curtis devices.

Values 0x4349
Read/Write RO

canopen_mandatory_family_number

`0x1018:02`
`canopen_mandatory_family_number`

Indicates the device's part number.

Values 0–999999999
Read/Write RO

canopen_mandatory_revision_number

`0x1018:03`
`canopen_mandatory_revision_number`

Indicates the hardware revision number.

Values 0–999
Read/Write RO

canopen_mandatory_serial_number

`0x1018:04`
`canopen_mandatory_serial_number`

Indicates the device's serial number.

Values 0–999999
Read/Write RO

manufacturing_datecode

`0x1018:05`
`manufacturing_datecode`

Indicates the date the unit was manufactured.

Values 0–99999. The value is a 5-digit number coded as *yyddd*:

- *yy* indicates the last two digits of the year.
- *ddd* indicates number of the day, which ranges from 1–365 in most years and 1–366 in leap years.

For example, if the manufacturing date is September 30, 2023, the parameter value is 23273.

system_configuration

`0x1018:07`
`system_configuration`

Describes the hardware options on an NX7 unit.

Values The following bits describe the installed options. 1 indicates that the unit includes the feature:

0 = Analog Video
1 = Wi-Fi and Bluetooth
2 = *reserved*
3 = NFC
4 = Isolated CAN 1
5 = Isolated CAN 2

Read/Write RO

Version Parameters

The device provides CAN objects to read the major, minor, and patch versions of the following components:

- Customer application
- Microcontroller unit (MC) firmware
- OS firmware
- Backend library

Customer Application Version Parameters

The customer application parameters read the application's major, minor, and patch versions.

customerApplicationVersion_major

`0x2010:01`
`customerApplicationVersion_major`

Indicates the customer application's major version.

Values 0–429496729

Read/Write RO

customerApplicationVersion_minor

`0x2010:02`
`customerApplicationVersion_minor`

Indicates the customer application's minor version.

Values 0–429496729

Read/Write RO

customerApplicationVersion_patch

`0x2010:03`

`customerApplicationVersion_patch`

Indicates the customer application's patch version.

Values 0–429496729

Read/Write RO

MCU Firmware Version Parameters

The MCU firmware parameters read the firmware's major, minor, and patch versions.

mcuFirmwareVersion_major

`0x2011:01`

`mcuFirmwareVersion_major`

Indicates the MCU firmware's major version.

Values 0–429496729

Read/Write RO

mcuFirmwareVersion_minor

`0x2011:02`

`mcuFirmwareVersion_minor`

Indicates the MCU firmware's minor version.

Values 0–429496729

Read/Write RO

mcuFirmwareVersion_patch

`0x2011:03`

`mcuFirmwareVersion_patch`

Indicates the MCU firmware's patch version.

Values 0–429496729

Read/Write RO

OS Version Parameters

The OS version parameters read the operating system's major, minor, and patch versions.

osVersion_major

`0x2012:01`

`osVersion_major`

Indicates the operating system's major version.

Values 0–429496729

Read/Write RO

osVersion_minor

`0x2012:02`

`osVersion_minor`

Indicates the operating system's minor version.

Values 0–429496729

Read/Write RO

osVersion_patch

`0x2012:03`

`osVersion_patch`

Indicates the operating system's patch version.

Values 0–429496729

Read/Write RO

Backend Library Version Parameters

The backend library version parameters read the library's major, minor, and patch versions.

backendLibraryVersion_major

`0x2013:01`

`backendLibraryVersion_major`

Indicates the backend library's major version.

Values 0–429496729

Read/Write RO

backendLibraryVersion_minor

`0x2013:02`

`backendLibraryVersion_minor`

Indicates the backend library's minor version.

Values 0–429496729

Read/Write RO

backendLibraryVersion_patch

`0x2013:03`

`backendLibraryVersion_patch`

Indicates the backend library's patch version.

Values 0–429496729

Read/Write RO

A — Specifications

Nominal Voltage	12–48V
Minimum Voltage	9V
Maximum Voltage	60V
Operating Current	• Typical operating current: ◦ Keyswitch off: 66–210mA ◦ Keyswitch on: 175–950mA • Maximum operating current: ◦ Keyswitch off: 66–210mA ◦ Keyswitch on: 350–2000mA
Dimensions, W x L x H	186.7 x 123.7 x 53.1 mm
Operating Temperature	−40°C to +70°C
Storage Temperature	–40°C to +85°C
Humidity	Designed to the following requirements: • **Soak**: EN 60068-2-78 • **Cyclic**: EN 60068-2-30
Ingress Protection	IP67
Shock	Designed to the requirements of EN 60068-2-27
Vibration	Designed to the following requirements: • **General**: EN 60068-2-6 • **Random**: EN 60068-2-64 • **Resonance**: EN 60068-2-6
EMC	Designed to the following requirements: • **Radiated Emissions**: ISO 13766-1:2018 • **Radiated Immunity**: ISO 13766-1:2018 • **Conducted Immunity**: ISO 13766-1:2018 • **Frequency Magnetic Field Immunity**: EN 12895:2013 • **ESD**: EN 12895:2013
CE	Designed to the following requirements: • EMC Directive 2014/30/EU • Low Voltage Directive (LVD) 2014/35/EU • RoHS directive 2015/863/EU (RoHS 3)
UL	UL recognized component per UL583

Note:

Regulatory compliance of the complete system with the NX7 installed is the responsibility of the OEM.

Model Encodement

The model number encodement is 3750T*VIRWT-XXX*. The *italicized* characters indicate which features are available in NX7 models, and are described in the following table.

Character	Description
V	Whether the model provides video: • Blank= no video • V = video
I	Whether the model provides isolated CAN: • Blank = no isolated CAN • I = Isolated CAN
R	Whether the model provides RFID: • Blank = no RFID • R = RFID
W	Whether the model provides WiFi or Bluetooth (BT), as well as supported standards: • 0 = no WiFi or BT • 4 = WiFi 4 / BT 4.2 • 5 = WiFi 5 / BT 4.2
T	Whether the model includes a touchscreen: • Blank = no touch panel • T = touchscreen
XXX	A random sequence of numbers.

For example, model number 3750TVI4T-196 is for a model with the following features:

- Video
- Isolated CAN
- WiFi 4 / BT 4.2
- Touchscreen