

---

# **python\_ics Documentation**

***Release 3.1***

**David Rebbe**

**Apr 24, 2019**



---

## Contents

---

<b>1</b>	<b>v802</b>	<b>3</b>
<b>2</b>	<b>v803</b>	<b>5</b>
<b>3</b>	<b>v900</b>	<b>7</b>
<b>4</b>	<b>Installation on Windows</b>	<b>9</b>
4.1	Building from source . . . . .	9
4.2	Intrepid icsneo40 Library . . . . .	9
<b>5</b>	<b>Installation on Linux</b>	<b>11</b>
5.1	Fedora Dependencies (FC28) . . . . .	11
5.2	Debian/Ubuntu Dependencies . . . . .	11
5.3	Others (Required dependencies) . . . . .	11
5.4	Installation . . . . .	11
5.5	Intrepid libicsneoapi.so Library . . . . .	12
<b>6</b>	<b>Getting Started</b>	<b>13</b>
<b>7</b>	<b>Module Function List</b>	<b>15</b>
<b>8</b>	<b>Module Documentation</b>	<b>23</b>
<b>9</b>	<b>Module Variables</b>	<b>73</b>
	<b>Python Module Index</b>	<b>85</b>



Minor differences can occur between different icsnVC40.h versions. These differences are usually just structures and constant differences. Below is a list of how the python\_ics version correlates to the icsnVC40.h version:



# CHAPTER 1

---

v802

---

```
pip install 'python_ics>=2.0,<3.0' --force-reinstall
```

**Note:** Refer to platform specific installation if not on Windows





## CHAPTER 2

---

v803

---

```
pip install 'python_ics>=3.0,<4.0' --force-reinstall
```

**Note:** Refer to platform specific installation if not on Windows



## CHAPTER 3

---

v900

---

```
pip install 'python_ics>=4.0,<5.0' --force-reinstall
```

**Note:** Refer to platform specific installation if not on Windows



---

### Installation on Windows

---

PyPi provides binary packages for Windows. You can simply install the `python_ics` module by running the following command:

```
pip install python_ics
```

**Note:** `pip.exe` is usually located under the `Scripts` directory under the Python installation directory.

#### 4.1 Building from source

Building from source on windows is not usually need so it won't really be covered here in detail. As a starting point you'll need to match the compiler version used to build the official Python binaries (MSVC). If the build environment is setup correctly, you should be able to run `python setup.py build` like usual.

#### 4.2 Intrepid icsneo40 Library

`python_ics` module looks for `icsneo40.dll` in the normal windows DLL search paths. The module will throw an exception if its not found.



python\_ics does not provide binaries for linux distributions so we will have to compile from source. This can be easily achieved by utilizing Python's PIP. First we need to make sure we have some base packages installed.

### 5.1 Fedora Dependencies (FC28)

```
sudo dnf install redhat-rpm-config gcc g++ python3-devel
```

### 5.2 Debian/Ubuntu Dependencies

```
sudo apt install build-essential python-dev
```

### 5.3 Others (Required dependencies)

- GCC
- G++
- Python Development packages (We Need to link to Python.h)

### 5.4 Installation

After dependencies are installed we can run the following pip command:

```
pip install python_ics
```

**Note:** A lot of distributions have Python 2 and 3 installed side by side. As of this writing without a version suffix the commands still default to version 2 of the Python binaries. In order to utilize the Python 3 binaries you must append a 3 after the binary names (python3 and pip3 instead of just python and pip).

## 5.5 Intrepid libicsneoapi.so Library

Please see <https://github.com/intrepidcs/icsneoapi> for more details.



## CHAPTER 6

---

### Getting Started

---

Please see [https://github.com/intrepidcs/python\\_ics/tree/master/examples](https://github.com/intrepidcs/python_ics/tree/master/examples) for simple examples on how to use this module. Most function documentation has a simple example on how its intended to be used. Every function was designed to be as close as possible to its C counterpart unless it was deemed to make the function more pythonic in nature.

For those experienced with the C API `ics.open_device (icsneoOpenNeoDevice())` behavior has been changed the most (no parameters makes it auto utilize `ics.find_devices (icsneoFindNeoDevices())` and open the first device). Also since python is a object oriented language the module utilizes this and auto cleans up device handles when going out of scope so there is usually no need to call `ics.close_device()`.



## CHAPTER 7

### Module Function List

<code>ics.close_device</code>	Closes the device.
<code>ics.coremini_clear</code>	Clears the CoreMini into the device.
<code>ics.coremini_get_fblock_status</code>	Gets the status of a Coremini Function Block at <i>index</i> on <i>device</i> .
<code>ics.coremini_get_status</code>	Gets the status of the CoreMini in the device.
<code>ics.coremini_load</code>	Loads the CoreMini into the device.
<code>ics.coremini_read_app_signal</code>	Gets the value of a Coremini application signal at <i>index</i> on <i>device</i> .
<code>ics.coremini_read_rx_message</code>	Gets the value of a Coremini Message at <i>index</i> on <i>device</i> .
<code>ics.coremini_read_tx_message</code>	Gets the value of a Coremini Message at <i>index</i> on <i>device</i> .
<code>ics.coremini_start</code>	Starts the CoreMini into the device.
<code>ics.coremini_start_fblock</code>	Starts a Coremini Function Block at <i>index</i> on <i>device</i> .
<code>ics.coremini_stop</code>	Stops the CoreMini into the device.
<code>ics.coremini_stop_fblock</code>	Stops a Coremini Function Block at <i>index</i> on <i>device</i> .
<code>ics.coremini_write_app_signal</code>	Sets the value of a Coremini application signal at <i>index</i> on <i>device</i> .
<code>ics.coremini_write_rx_message</code>	TODO
<code>ics.coremini_write_tx_message</code>	TODO
<code>ics.create_neovi_radio_message</code>	Python API only.
<code>ics.find_devices</code>	Finds all connected devices and returns a tuple of <code>ics.NeoDevice</code> for use in <code>ics.open_device()</code>
<code>ics.firmware_update_required</code>	Determines if the device firmware needs flashing.
<code>ics.force_firmware_update</code>	Forces the device to flash firmware.
<code>ics.get_active_vnet_channel</code>	Gets active vnet channel for the device.
<code>ics.get_backup_power_enabled</code>	Returns the device backup power enabled for the device.
<code>ics.get_backup_power_ready</code>	Returns the device backup power is ready for the device.
<code>ics.get_device_settings</code>	Gets the settings in the device.
<code>ics.get_device_status</code>	Returns the device status.

Continued on next page

Table 1 – continued from previous page

<code>ics.get_dll_firmware_info</code>	Returns the DLL firmware info for the device.
<code>ics.get_dll_version</code>	Gets the DLL version.
<code>ics.get_error_messages</code>	Gets the error message(s) on the device.
<code>ics.get_hw_firmware_info</code>	Returns the device firmware info for the device.
<code>ics.get_last_api_error</code>	Gets the error message from the last API call.
<code>ics.get_library_path</code>	
<code>ics.get_messages</code>	Gets the message(s) on the device.
<code>ics.get_performance_parameters</code>	Gets the Performance Parameters on <i>device</i> .
<code>ics.get_rtc</code>	Gets the Real-Time Clock of the device.
<code>ics.get_script_status</code>	Accepts a <code>ics.NeoDevice</code> , exception on error.
<code>ics.get_serial_number</code>	Gets the serial number out of the device.
<code>ics.get_timestamp_for_msg</code>	Calculates the timestamp for a message.
<code>ics.iso15765_disable_networks</code>	Disables ISO15765 networks.
<code>ics.iso15765_enable_networks</code>	Enables ISO15765 networks.
<code>ics.iso15765_receive_message</code>	Setup rx ISO15765 Message.
<code>ics.iso15765_transmit_message</code>	Transmits an ISO15765 Message.
<code>ics.load_default_settings</code>	Load the default settings in the device.
<code>ics.open_device</code>	Opens the device.
<code>ics.override_library_name</code>	Sets active vnet channel for the device.
<code>ics.read_sdcard</code>	<code>icsneoReadSDCard()</code> , Accepts a <code>ics.NeoDevice</code> and sector index.
<code>ics.request_enter_sleep_mode</code>	Signal neoVI to immediate go to sleep.
<code>ics.set_active_vnet_channel</code>	Sets active vnet channel for the device.
<code>ics.set_backup_power_enabled</code>	Sets the device backup power enabled for the device.
<code>ics.set_bit_rate</code>	Sets the bitrate for a given Network ID on the device..
<code>ics.set_bit_rate_ex</code>	<code>ics.set_fd_bit_rate_ex(device, BitRate, NetworkID, iOptions)</code>
<code>ics.set_context</code>	Sets the “context” of how <code>icsneoFindNeoDevices(Ex)</code> and <code>icsneoOpenNeoDevice(Ex)</code> function.
<code>ics.set_device_settings</code>	Sets the settings in the device.
<code>ics.set_fd_bit_rate</code>	Sets the FD bitrate for a given Network ID on the device..
<code>ics.set_reflash_callback</code>	Sets the reflash display callback.
<code>ics.set_rtc</code>	Sets the Real-Time Clock of the device.
<code>ics.transmit_messages</code>	Transmits message(s) on the device.
<code>ics.validate_hobject</code>	Validates the handle is valid for a <i>device</i> .
<code>ics.write_sdcard</code>	<code>icsneoReadSDCard()</code> , Accepts a <code>ics.NeoDevice</code> , sector index, and a bytearray of 512 bytes.
<code>ics.ClosePort</code>	

---

**Note:** Compatibility Function

---



---

`ics.FindNeoDevices`

---



---

**Note:** Compatibility Function

---

Continued on next page

Table 1 – continued from previous page

---

<i>ics.GetDLLVersion</i>	<b>Note:</b> Compatibility Function
<i>ics.GetErrorMessages</i>	<b>Note:</b> Compatibility Function
<i>ics.GetHWFirmwareInfo</i>	<b>Note:</b> Compatibility Function
<i>ics.GetLastAPIError</i>	<b>Note:</b> Compatibility Function
<i>ics.GetMessages</i>	<b>Note:</b> Compatibility Function
<i>ics.GetPerformanceParameters</i>	<b>Note:</b> Compatibility Function
<i>ics.GetRTC</i>	<b>Note:</b> Compatibility Function
<i>ics.GetSerialNumber</i>	<b>Note:</b> Compatibility Function
<i>ics.OpenNeoDevice</i>	<b>Note:</b> Compatibility Function
<i>ics.RequestEnterSleepMode</i>	<b>Note:</b> Compatibility Function

---

Continued on next page

Table 1 – continued from previous page

---

*ics.ScriptClear*

---

**Note:** Compatibility Function

---

*ics.ScriptGetFBlockStatus*

---

**Note:** Compatibility Function

---

*ics.ScriptGetScriptStatus*

---

**Note:** Compatibility Function

---

*ics.ScriptLoad*

---

**Note:** Compatibility Function

---

*ics.ScriptReadAppSignal*

---

**Note:** Compatibility Function

---

*ics.ScriptReadRxMessage*

---

**Note:** Compatibility Function

---

*ics.ScriptReadTxMessage*

---

**Note:** Compatibility Function

---

*ics.ScriptStart*

---

**Note:** Compatibility Function

---

*ics.ScriptStartFBlock*

---

**Note:** Compatibility Function

---

*ics.ScriptStop*

---

**Note:** Compatibility Function

---

---

Continued on next page

---

Table 1 – continued from previous page

<i>ics.ScriptStopFBlock</i>	<b>Note:</b> Compatibility Function
<i>ics.ScriptWriteAppSignal</i>	<b>Note:</b> Compatibility Function
<i>ics.ScriptWriteRxMessage</i>	<b>Note:</b> Compatibility Function
<i>ics.ScriptWriteTxMessage</i>	<b>Note:</b> Compatibility Function
<i>ics.SetRTC</i>	<b>Note:</b> Compatibility Function
<i>ics.SetReflashDisplayCallback</i>	<b>Note:</b> Compatibility Function
<i>ics.TxMessages</i>	<b>Note:</b> Compatibility Function
<i>ics.ValidateHObject</i>	<b>Note:</b> Compatibility Function
<i>ics.base36enc</i>	Converts a decimal serial number to base36.
<i>ics.icsneoFirmwareUpdateRequired</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoForceFirmwareUpdate</i>	<b>Note:</b> Compatibility Function

Continued on next page

Table 1 – continued from previous page

---

<i>ics.icsneoGetActiveVNETChannel</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoGetBackupPowerEnabled</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoGetBackupPowerReady</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoGetDLLFirmwareInfo</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoGetDeviceStatus</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoGetFireSettings</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoGetTimeStampForMsg</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoGetVCAN3Settings</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoISO15765_DisableNetworks</i>	<b>Note:</b> Compatibility Function
<hr/>	
<i>ics.icsneoISO15765_EnableNetworks</i>	<b>Note:</b> Compatibility Function
<hr/>	

---

Continued on next page



Table 1 – continued from previous page

<i>ics.icsneoISO15765_ReceiveMessage</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoISO15765_TransmitMessage</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoLoadDefaultSettings</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoReadSDCard</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoScriptGetScriptStatusEx</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoSetActiveVNETChannel</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoSetBackupPowerEnabled</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoSetBitRate</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoSetBitRateEx</i>	<b>Note:</b> Compatibility Function
<i>ics.icsneoSetContext</i>	<b>Note:</b> Compatibility Function

Continued on next page

Table 1 – continued from previous page

---

*ics.icsneoSetFDBitRate*

---

**Note:** Compatibility Function

---

---

*ics.icsneoSetFireSettings*

---

**Note:** Compatibility Function

---

---

*ics.icsneoSetVCAN3Settings*

---

**Note:** Compatibility Function

---

---

*ics.icsneoWriteSDCard*

---

**Note:** Compatibility Function

---

---

## Module Documentation

---

Python C Code module for interfacing to the icsneo40 dynamic library. Code tries to respect PEP 8 (<http://python.org/dev/peps/pep-0008>). Function naming convention does not follow the tradition c style icsneo40 naming convention as pyics module name acts as the namespace (icsneo portion of the function) and function names are suppose to be lowercase with underscores instead of mixedCase like icsneo API.

**C API can be mimiced almost identically by doing the following:**

```
>>> import ics as icsneo
>>> devices = icsneo.FindNeoDevices()
>>> for device in devices:
...     print(device.Name, device.SerialNumber)
...
neoVI FIRE 59886
```

**Recommended *Python* way by doing the following:**

```
>>> import ics
>>> devices = ics.find_devices()
>>> for device in devices:
...     print(device.Name, device.SerialNumber)
...
neoVI FIRE 59886
```

It should be noted that *ics.NeoDevice* is used a little bit differently than the C API. *ics.NeoDevice* contains two extra members:

*ics.NeoDevice.AutoHandleClose* and *ics.NeoDevice.\_Handle*

The handle normally returned from *icsneoOpenNeoDevice()* is stored inside *\_Handle* and setting *AutoHandleClose* to True (Default) will automatically close the handle when the *ics.NeoDevice* goes out of scope.

Installation:

```
pip install python_ics
```

<https://pypi.python.org/pypi/python-ics>

**exception** `ics.ArgumentError`

Bases: `Exception`

**exception** `ics.RuntimeError`

Bases: `Exception`

**class** `ics.ApiFirmwareInfo`

Bases: `object`

ApiFirmwareInfo object

**iAppMajor**

**iAppMinor**

**iBoardRevMajor**

**iBoardRevMinor**

**iBootLoaderVersionMajor**

**iBootLoaderVersionMinor**

**iMainFirmChkSum**

**iMainFirmDateDay**

**iMainFirmDateHour**

**iMainFirmDateMin**

**iMainFirmDateMonth**

**iMainFirmDateSecond**

**iMainFirmDateYear**

**iMainVnetHWrevMajor**

**iMainVnetHWrevMinor**

**iMainVnetSRAMSize**

**iManufactureDay**

**iManufactureMonth**

**iManufactureYear**

**iType**

**class** `ics.CanFdSettings`

Bases: `object`

CanFdSettings object

**FDBRP**

**FDBaudrate**

**FDMode**

**FDTqProp**

**FDTqSeg1**

**FDTqSeg2**

**FDTqSync**

**class ics.CanSettings**

Bases: object

CanSettings object

**BRP**

**Baudrate**

The bit rate of a CAN channel can be selected from a list of common bit rates Write the correct enumeration for the desired bit rate and ensure that SetBaudrate is 1(auto)

**Mode**

CAN controller mode when the neoVI device goes online or runs a CoreMini script. Normal=0 Disabled=1 Listen Only=3 Listen All=7

**SetBaudrate**

The bit rate of a CAN channel can be selected one of two ways. It can either be selected from a list of common bit rates (SetBaudrate=1) or the user can specify the CAN timing parameters (SetBaudrate=0)

**TqProp**

Propagation delay

**TqSeg1**

Phase 1 segment

**TqSeg2**

Phase 2 segment

**TqSync**

Syncro jump width

**auto\_baud**

Enables the auto bitrate feature. 1 = enable, 0 = disable.

**innerFrameDelay25us**

**transceiver\_mode**

Currently Not used.

**class ics.CmISO157652RxMessage**

Bases: object

CmISO157652RxMessage object

**blockSize**

Overrides the block size that the receiver reports, see overrideBlockSize. Set to J2534's BS\_TX if <= 0xFF

**cf\_timeout**

max timeout (ms) for waiting on consecutive frame. Set this to N\_CR\_MAX's value in J2534

**extendedAddress**

Extended Address byte of transmitter. see ext\_address\_enable, not supported

**fc\_id**

flow control arbid to transmit in flow control (from neoVI to ECU)

**flags**

**flowControlExtendedAddress**

Expected Extended Address byte of response from receiver. see fc\_ext\_address\_enable, not supported

**id**

arbid of transmitted frames (CAN id to transmit to)

**id\_mask**

ArbId filter mask for frames from transmitter (from ECU to neoVI)

**padding**

The padding byte to use to fill the unused portion of \* transmitted CAN frames (flow control), see paddingEnable.

**reserved****stMin**

Minimum separation time (between consecutive frames) to report in flow control response

**vs\_netid**

The netid of the message (determines which network to decode receives), not supported

**class ics.CmISO157652TxMessage**

Bases: object

CmISO157652TxMessage object

**blockSize**

Overrides the block size that the receiver reports, see overrideBlockSize. Set to J2534's BS\_TX if <= 0xFF

**data**

The data

**extendedAddress**

Extended Address byte of transmitter. see ext\_address\_enable, not supported

**fc\_id**

flow control arb id filter value (response id from receiver)

**fc\_id\_mask**

The flow control arb filter mask (response id from receiver)

**flags****flowControlExtendedAddress**

Expected Extended Address byte of response from receiver. see fc\_ext\_address\_enable, not supported

**fs\_timeout**

max timeout (ms) for waiting on flow control responses. Set this to N\_BS\_MAX's value if J2534

**fs\_wait**

max timeout (ms) for waiting on flow control response after receiving flow control \* with flow status set to WAIT. Set this to N\_BS\_MAX's value if J2534.

**id**

arbId of transmitted frames (CAN id to transmit to)

**num\_bytes**

Number of data bytes

**padding**

The padding byte to use to fill the unused portion of \* transmitted CAN frames (single frame, first frame, consecutive frame) \*

**stMin**

Overrides the stMin that the receiver reports, see overrideSTmin. Set to J2534's STMIN\_TX if <= 0xFF

**tx\_index****vs\_netid**

The netid of the message (determines which network to transmit on), not supported

```
class ics.CyanSettings
    Bases: object

    CyanSettings object

    ain_sample_period

    ain_threshold

    can1
        ics.CanSettings Object

    can2
        ics.CanSettings Object

    can3
        ics.CanSettings Object

    can4
        ics.CanSettings Object

    can5
        ics.CanSettings Object

    can6
        ics.CanSettings Object

    can7
        ics.CanSettings Object

    can8
        ics.CanSettings Object

    can_switch_mode

    canfd1
        ics.CanFdSettings Object

    canfd2
        ics.CanFdSettings Object

    canfd3
        ics.CanFdSettings Object

    canfd4
        ics.CanFdSettings Object

    canfd5
        ics.CanFdSettings Object

    canfd6
        ics.CanFdSettings Object

    canfd7
        ics.CanFdSettings Object

    canfd8
        ics.CanFdSettings Object

    digitalIoThresholdEnable

    digitalIoThresholdTicks

    disableUsbCheckOnBoot

    enableLatencyTest
```

**ethernet**  
ics.EthernetSettings Object

**idle\_wakeup\_network\_enables\_3**

**iso15765\_separation\_time\_offset**

**iso9141\_kwp\_settings\_1**  
Iso9141Keyword2000Settings Object

**iso9141\_kwp\_settings\_2**  
Iso9141Keyword2000Settings Object

**iso9141\_kwp\_settings\_3**  
Iso9141Keyword2000Settings Object

**iso9141\_kwp\_settings\_4**  
Iso9141Keyword2000Settings Object

**iso\_msg\_termination\_1**  
0 - use inner frame time, 1 - GME CIM-SCL

**iso\_msg\_termination\_2**  
0 - use inner frame time, 1 - GME CIM-SCL

**iso\_msg\_termination\_3**  
0 - use inner frame time, 1 - GME CIM-SCL

**iso\_msg\_termination\_4**  
0 - use inner frame time, 1 - GME CIM-SCL

**iso\_parity\_1**  
0 - no parity, 1 - event, 2 - odd

**iso\_parity\_2**  
0 - no parity, 1 - event, 2 - odd

**iso\_parity\_3**  
0 - no parity, 1 - event, 2 - odd

**iso\_parity\_4**  
0 - no parity, 1 - event, 2 - odd

**lin1**  
ics.LinSettings Object

**lin2**  
ics.LinSettings Object

**lin3**  
ics.LinSettings Object

**lin4**  
ics.LinSettings Object

**lin5**  
ics.LinSettings Object

**lin6**  
ics.LinSettings Object

**lsft1**  
ics.CanSettings Object



**lsft2**  
ics.CanSettings Object

**misc\_io\_analog\_enable**

**misc\_io\_initial\_ddr**

**misc\_io\_initial\_latch**

**misc\_io\_on\_report\_events**

**misc\_io\_report\_period**

**network\_enabled\_on\_boot**

**network\_enables**

**network\_enables\_2**

**network\_enables\_3**

**perf\_en**

**pwr\_man\_enable**

**pwr\_man\_timeout**

**reserved**

**slaveVnetA**

**slaveVnetB**

**swcan1**  
ics.SWCanSettings Object

**swcan2**  
ics.SWCanSettings Object

**termination\_enables**

**text\_api**  
ics.TextApiSettings Object

**class ics.DeviceSettings**  
Bases: object

DeviceSettings object

**DeviceSettingType**

**cyan**  
ics.CyanSettings Object

**fire**  
ics.FireSettings Object

**radgalaxy**  
ics.RadGalaxySettings Object

**radstar2**  
ics.RadStar2Settings Object

**vcan3**  
ics.Vcan3Settings Object

**vcan4**  
ics.Vcan4Settings Object

**vcan4\_12**  
ics.Vcan412Settings Object

**vividcan**  
ics.VividCANSettings Object

**class ics.EthernetSettings**

Bases: object

EthernetSettings object

**auto\_neg**

**duplex**

**led\_mode**

**link\_speed**

**rsvd**

**class ics.Fire2DeviceStatus**

Bases: object

Fire2DeviceStatus object

**backupPowerEnabled**

**backupPowerGood**

**ethernetActivationLineEnabled**

**usbHostPowerEnabled**

**class ics.FireSettings**

Bases: object

FireSettings object

**ain\_sample\_period**

**ain\_threshold**

**can1**  
ics.CanSettings Object

**can2**  
ics.CanSettings Object

**can3**  
ics.CanSettings Object

**can4**  
ics.CanSettings Object

**cgi\_baud**

**cgi\_chksum\_enable**

**cgi\_enable\_reserved**

**cgi\_rx\_ifs\_bit\_times**

**cgi\_tx\_ifs\_bit\_times**

**fast\_init\_network\_enables\_1**

**fast\_init\_network\_enables\_2**

**iso15765\_separation\_time\_offset**  
**iso9141\_kwp\_enable\_reserved**  
**iso9141\_kwp\_settings**  
Iso9141Keyword2000Settings Object  
**iso9141\_kwp\_settings\_2**  
Iso9141Keyword2000Settings Object  
**iso9141\_kwp\_settings\_3**  
Iso9141Keyword2000Settings Object  
**iso9141\_kwp\_settings\_4**  
Iso9141Keyword2000Settings Object  
**iso\_msg\_termination**  
0 - use inner frame time, 1 - GME CIM-SCL  
**iso\_msg\_termination\_2**  
0 - use inner frame time, 1 - GME CIM-SCL  
**iso\_msg\_termination\_3**  
0 - use inner frame time, 1 - GME CIM-SCL  
**iso\_msg\_termination\_4**  
0 - use inner frame time, 1 - GME CIM-SCL  
**iso\_parity**  
0 - no parity, 1 - event, 2 - odd  
**iso\_parity\_2**  
0 - no parity, 1 - event, 2 - odd  
**iso\_parity\_3**  
0 - no parity, 1 - event, 2 - odd  
**iso\_parity\_4**  
0 - no parity, 1 - event, 2 - odd  
**iso\_tester\_pullup\_enable**  
**lin1**  
ics.LinSettings Object  
**lin2**  
ics.LinSettings Object  
**lin3**  
ics.LinSettings Object  
**lin4**  
ics.LinSettings Object  
**lsft**  
ics.CanSettings Object  
**misc\_io\_analog\_enable**  
**misc\_io\_initial\_ddr**  
**misc\_io\_initial\_latch**  
**misc\_io\_on\_report\_events**  
**misc\_io\_report\_period**

```
network_enabled_on_boot
network_enables
network_enables_2
perf_en
pwm_man_timeout
pwr_man_enable
swcan
    ics.SWCanSettings Object
text_api
    ics.TextApiSettings Object
uart
    ics.UartSettings Object
uart2
    ics.UartSettings Object
vnetBits
class ics.IcsDeviceStatus
    Bases: object
    IcsDeviceStatus object
    fire2Status
        ics.Fire2DeviceStatus Object
    vcan4Status
        ics.Vcan4DeviceStatus Object
class ics.Iso9141Keyword2000InitSteps
    Bases: object
    Iso9141Keyword2000InitSteps object
    k
    l
    time_500us
class ics.Iso9141Keyword2000Settings
    Bases: object
    Iso9141Keyword2000Settings object
    Baudrate
    brgh
    checksum_enabled
    init_steps
        Tuple of Iso9141Keyword2000InitSteps
    p2_500us
    p3_500us
    p4_500us
```

**spbrg**

**class** ics.LinSettings

Bases: object

LinSettings object

**Baudrate**

**MasterResistor**

**Mode**

**brgh**

**spbrg**

**class** ics.NeoDevice

Bases: object

NeoDevice object

**AutoHandleClose**

When NeoDevice is freed the handle will automatically be closed, if true.

**DeviceType**

**Handle**

**IsOpen**

This contains the handle returned from icsneoOpenDevice() API. If uncertain, don't use this.

**MaxAllowedClients**

**Name**

String describing DeviceType, extension to Python api only.

**NumberOfClients**

**SerialNumber**

**class** ics.OpEthGeneralSettings

Bases: object

OpEthGeneralSettings object

**bEnReportLinkQuality**

**bTapEnPtp**

**bTapEnSwitch**

**reserved0**

**tapPair0**

**tapPair1**

**tapPair2**

**tapPair3**

**tapPair4**

**tapPair5**

**ucInterfaceType**

**class** ics.OpEthSettings

Bases: object

OpEthSettings object

**preemption\_en**

**reserved0**

**ucConfigMode**

**class** ics.RadGalaxySettings

Bases: object

RadGalaxySettings object

**ain\_sample\_period**

**ain\_threshold**

**can1**

ics.CanSettings Object

**can2**

ics.CanSettings Object

**can3**

ics.CanSettings Object

**can4**

ics.CanSettings Object

**can5**

ics.CanSettings Object

**can6**

ics.CanSettings Object

**can7**

ics.CanSettings Object

**can8**

ics.CanSettings Object

**can\_switch\_mode**

**canfd1**

ics.CanFdSettings Object

**canfd2**

ics.CanFdSettings Object

**canfd3**

ics.CanFdSettings Object

**canfd4**

ics.CanFdSettings Object

**canfd5**

ics.CanFdSettings Object

**canfd6**

ics.CanFdSettings Object

**canfd7**

ics.CanFdSettings Object

**canfd8**  
ics.CanFdSettings Object

**idle\_wakeup\_network\_enables\_1**

**idle\_wakeup\_network\_enables\_2**

**idle\_wakeup\_network\_enables\_3**

**iso15765\_separation\_time\_offset**

**iso9141\_kwp\_settings\_1**  
Iso9141Keyword2000Settings Object

**iso\_msg\_termination\_1**  
0 - use inner frame time, 1 - GME CIM-SCL

**iso\_parity\_1**  
0 - no parity, 1 - event, 2 - odd

**lin1**  
ics.LinSettings Object

**misc\_io\_analog\_enable**

**misc\_io\_initial\_ddr**

**misc\_io\_initial\_latch**

**misc\_io\_on\_report\_events**

**misc\_io\_report\_period**

**network\_enabled\_on\_boot**

**network\_enables**

**network\_enables\_2**

**network\_enables\_3**

**opEth1**  
ics.OpEthSettings Object

**opEth10**  
ics.OpEthSettings Object

**opEth11**  
ics.OpEthSettings Object

**opEth12**  
ics.OpEthSettings Object

**opEth2**  
ics.OpEthSettings Object

**opEth3**  
ics.OpEthSettings Object

**opEth4**  
ics.OpEthSettings Object

**opEth5**  
ics.OpEthSettings Object

**opEth6**  
ics.OpEthSettings Object

**opEth7**  
ics.OpEthSettings Object

**opEth8**  
ics.OpEthSettings Object

**opEth9**  
ics.OpEthSettings Object

**opEthGen**  
ics.OpEthGeneralSettings Object

**perf\_en**

**pwr\_man\_enable**

**pwr\_man\_timeout**

**swcan1**  
ics.SWCanSettings Object

**swcan2**  
ics.SWCanSettings Object

**text\_api**  
ics.TextApiSettings Object

**class ics.RadStar2Settings**  
Bases: object

RadStar2Settings object

**ain\_sample\_period**

**ain\_threshold**

**can1**  
ics.CanSettings Object

**can2**  
ics.CanSettings Object

**can\_switch\_mode**

**canfd1**  
ics.CanFdSettings Object

**canfd2**  
ics.CanFdSettings Object

**hwComLatencyTestEn**

**idle\_wakeup\_network\_enables\_1**

**idle\_wakeup\_network\_enables\_2**

**idle\_wakeup\_network\_enables\_3**

**iso15765\_separation\_time\_offset**

**iso9141\_kwp\_settings\_1**  
Iso9141Keyword2000Settings Object

**iso\_9141\_kwp\_enable\_reserved**

**iso\_msg\_termination\_1**  
0 - use inner frame time, 1 - GME CIM-SCL



**iso\_parity\_1**  
0 - no parity, 1 - event, 2 - odd

**lin1**  
ics.LinSettings Object

**misc\_io\_analog\_enable**

**misc\_io\_initial\_ddr**

**misc\_io\_initial\_latch**

**misc\_io\_on\_report\_events**

**misc\_io\_report\_period**

**network\_enabled\_on\_boot**

**network\_enables**

**network\_enables\_2**

**network\_enables\_3**

**opEth1**  
ics.OpEthSettings Object

**opEth2**  
ics.OpEthSettings Object

**opEthGen**  
ics.OpEthGeneralSettings Object

**pc\_com\_mode**

**perf\_en**

**pwr\_man\_enable**

**pwr\_man\_timeout**

**text\_api**  
ics.TextApiSettings Object

**timeSyncSettings**  
ics.TimesyncSettings Object

**class ics.SWCanSettings**

Bases: object

SWCanSettings object

**BRP**

**Baudrate**

The bit rate of a CAN channel can be selected from a list of common bit rates Write the correct enumeration for the desired bit rate and ensure that SetBaudrate is 1(auto)

**Mode**

CAN controller mode when the neoVI device goes online or runs a CoreMini script. Normal=0 Disabled=1 Listen Only=3 Listen All=7

**RESERVED**

**SetBaudrate**

The bit rate of a CAN channel can be selected one of two ways. It can either be selected from a list of common bit rates (SetBaudrate=1) or the user can specify the CAN timing parameters (SetBaudrate=0)

**TqProp**

Propagation delay

**TqSeg1**

Phase 1 segment

**TqSeg2**

Phase 2 segment

**TqSync**

Syncro jump width

**auto\_baud**

Enables the auto bitrate feature. 1 = enable, 0 = disable.

**high\_speed\_auto\_switch****transceiver\_mode**

Currently Not used.

**class ics.SpyMessage**

Bases: object

SpyMessage object

**AckBytes****ArbIDOrHeader****Data****DescriptionID**

Not Used

**ExtraDataPtr****ExtraDataPtrEnabled****MessagePieceID**

Not Used

**MiscData****NetworkID**

This value is used to identify which network this message was received on.

**NetworkID2**

This value is used to identify which network this message was received on.

**NodeID**

Not Used

**NumberBytesData**

Holds the number of bytes in the Data(1 to 8) array or the number of bytes in a CAN remote frame (The DLC).

**NumberBytesHeader**

Used for J1850/ISO messages. It indicates how many bytes are stored in the Header(1 to 4) array.

**Protocol**

Valid values are SPY\_PROTOCOL\_CAN, SPY\_PROTOCOL\_J1850VPW, and SPY\_PROTOCOL\_ISO9141.

**StatusBitField****StatusBitField2**

**StatusBitField3**

**StatusBitField4**

**TimeHardware**

Hardware time stamp. The TimeStamp is reset on device open

**TimeHardware2**

Hardware time stamp. The TimeStamp is reset on device open

**TimeStampHardwareID**

This is an identifier of what type of hardware timestamp is used. Since neoVI's timestamp is always the same, this doesn't change.

**TimeStampSystemID**

This is an identifier of what type of system timestamp is used. Since WIN32 neoVI's timestamp is always the same, from the timeGetTime API, this doesn't change.

**TimeSystem**

TimeSystem is loaded with the value received from the timeGetTime call in the WIN32 multimedia API.

**TimeSystem2**

TimeSystem is loaded with the value received from the timeGetTime call in the WIN32 multimedia API.

**noExtraDataPtrCleanup**

Tells Python to not clean up ExtraDataPtrMemory, If this is enabled. Ignore, if unsure.

**class** ics.**SpyMessageJ1850**

Bases: object

SpyMessageJ1850 object

**AckBytes**

**Data**

**DescriptionID**

Not Used

**ExtraDataPtr**

**ExtraDataPtrEnabled**

**Header**

**MessagePieceID**

Not Used

**MiscData**

**NetworkID**

This value is used to identify which network this message was received on.

**NetworkID2**

This value is used to identify which network this message was received on.

**NodeID**

Not Used

**NumberBytesData**

Holds the number of bytes in the Data(1 to 8) array or the number of bytes in a CAN remote frame (The DLC).

**NumberBytesHeader**

Used for J1850/ISO messages. It indicates how many bytes are stored in the Header(1 to 4) array.

**Protocol**

Valid values are SPY\_PROTOCOL\_CAN, SPY\_PROTOCOL\_J1850VPW, and SPY\_PROTOCOL\_ISO9141.

**StatusBitField****StatusBitField2****StatusBitField3****StatusBitField4****TimeHardware**

Hardware time stamp. The TimeStamp is reset on device open

**TimeHardware2**

Hardware time stamp. The TimeStamp is reset on device open

**TimeStampHardwareID**

This is an identifier of what type of hardware timestamp is used. Since neoVI's timestamp is always the same, this doesn't change.

**TimeStampSystemID**

This is an identifier of what type of system timestamp is used. Since WIN32 neoVI's timestamp is always the same, from the timeGetTime API, this doesn't change.

**TimeSystem**

TimeSystem is loaded with the value received from the timeGetTime call in the WIN32 multimedia API.

**TimeSystem2**

TimeSystem is loaded with the value received from the timeGetTime call in the WIN32 multimedia API.

**noExtraDataPtrCleanup**

Tells Python to not clean up ExtraDataPtrMemory, If this is enabled. Ignore, if unsure.

**class ics.TextApiSettings**

Bases: object

TextApiSettings object

**can1\_options**

Sets the length of the Arbitration ID's. Set to 1 for Extended and 0 for Standard

**can1\_rx\_id**

Sets or Reads the Arbitration ID for Sending Receiving API commands

**can1\_tx\_id**

Sets or Reads the Arbitration ID for Sending Text API commands

**can2\_options****can2\_rx\_id****can2\_tx\_id****can3\_options****can3\_rx\_id****can3\_tx\_id****can4\_options****can4\_rx\_id****can4\_tx\_id**

**network\_enables**

Bitfield telling which network to support Text API.

**class ics.TimesyncSettings**

Bases: object

TimesyncSettings object

**MasterEnable****MasterNetwork****SlaveEnable****SlaveNetwork****class ics.UartSettings**

Bases: object

UartSettings object

**Baudrate**

Holds the baud rate for the UART Connection. An example value could be 10417 or 9600

**bOptions**

Bitfield containing UART Options Invert TX=1, Invert RX=2, Half Duplex=4

**brgh****flow\_control**

Set to 0 for no flow control and 1 for simple CTS RTS

**parity**

Sets the Parity type. Valid values are None=0, Even=1, Odd=2

**reserved\_1****spbrg****stop\_bits**

Sets the number of stop bits to use. Valid values are One=1, Two=2

**class ics.Vcan3Settings**

Bases: object

Vcan3Settings object

**can1**

ics.CanSettings Object

**can2**

ics.CanSettings Object

**isol5765\_separation\_time\_offset****misc\_io\_initial\_ddr****misc\_io\_initial\_latch****misc\_io\_on\_report\_events****misc\_io\_report\_period****network\_enabled\_on\_boot****network\_enables****perf\_en**

```
class ics.Vcan412Settings
    Bases: object
    Vcan412Settings object

    can1
        ics.CanSettings Object

    can2
        ics.CanSettings Object

    canfd1
        ics.CanFdSettings Object

    canfd2
        ics.CanFdSettings Object

    disableUsbCheckOnBoot
        flags

    enableLatencyTest
        flags

    iso15765_separation_time_offset

    network_enabled_on_boot

    network_enables

    perf_en

    pwr_man_enable

    pwr_man_timeout

    reserved
        flags

    termination_enables

    text_api
        ics.TextApiSettings Object

class ics.Vcan4DeviceStatus
    Bases: object
    Vcan4DeviceStatus object

    ethernetActivationLineEnabled

class ics.Vcan4Settings
    Bases: object
    Vcan4Settings object

    can1
        ics.CanSettings Object

    can2
        ics.CanSettings Object

    can3
        ics.CanSettings Object

    can4
        ics.CanSettings Object
```

**canfd1**  
ics.CanFdSettings Object

**canfd2**  
ics.CanFdSettings Object

**canfd3**  
ics.CanFdSettings Object

**canfd4**  
ics.CanFdSettings Object

**enableLatencyTest**  
flags

**enablePcEthernetComm**  
flags

**ethernet**  
ics.EthernetSettings Object

**iso15765\_separation\_time\_offset**

**iso9141\_kwp\_settings\_1**  
Iso9141Keyword2000Settings Object

**iso\_9141\_kwp\_enable\_reserved**

**iso\_msg\_termination\_1**

**iso\_parity\_1**

**lin1**  
ics.LinSettings Object

**network\_enabled\_on\_boot**

**network\_enables**

**network\_enables\_2**

**network\_enables\_3**

**perf\_en**

**pwr\_man\_enable**

**pwr\_man\_timeout**

**reserved**  
flags

**termination\_enables**

**text\_api**  
ics.TextApiSettings Object

**class ics.VcanRFSettings**

Bases: object

VcanRFSettings object

**can1**  
ics.CanSettings Object

**can2**  
ics.CanSettings Object

**can3**  
ics.CanSettings Object

**can4**  
ics.CanSettings Object

**idle\_wakeup\_network\_enables\_1**

**idle\_wakeup\_network\_enables\_2**

**iso15765\_separation\_time\_offset**

**iso9141\_kwp\_enable\_reserved**

**iso9141\_kwp\_settings**  
ics.Iso9141Keyword2000Settings Object

**iso9141\_kwp\_settings\_2**  
ics.Iso9141Keyword2000Settings Object

**iso\_msg\_termination**  
0 - use inner frame time, 1 - GME CIM-SCL

**iso\_msg\_termination\_2**  
0 - use inner frame time, 1 - GME CIM-SCL

**iso\_parity**  
0 - no parity, 1 - event, 2 - odd

**iso\_parity\_2**  
0 - no parity, 1 - event, 2 - odd

**iso\_tester\_pullup\_enable**

**lin1**  
ics.LinSettings Object

**lin2**  
ics.LinSettings Object

**misc\_io\_analog\_enable**

**misc\_io\_initial\_ddr**

**misc\_io\_initial\_latch**

**misc\_io\_on\_report\_events**

**misc\_io\_report\_period**

**network\_enabled\_on\_boot**

**network\_enables**

**network\_enables\_2**

**perf\_en**

**pwr\_man\_enable**  
0 - off, 1 - sleep enabled, 2- idle enabled (fast wakeup)

**pwr\_man\_timeout**

**class ics.VividCANSettings**  
Bases: object  
VividCANSettings object



**can1**  
ics.CanSettings Object

**can\_switch\_mode**

**disableUsbCheckOnBoot**  
flags

**ecu\_id**

**enableLatencyTest**  
flags

**iso15765\_separation\_time\_offset**

**lsftcan1**  
ics.CanSettings Object

**network\_enabled\_on\_boot**

**network\_enables**

**perf\_en**

**pwr\_man\_enable**

**pwr\_man\_timeout**

**reserved**  
flags

**swcan1**  
ics.CanSettings Object

**termination\_enables**

**ics.ClosePort()**

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.close\_device()* method.

---

**ics.EnableNetworkCom()**

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.enable\_network\_com()* method.

---

**ics.FindNeoDevices()**

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.find\_devices()* method.

---

**ics.FirmwareUpdateRequired()**

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.firmware\_update\_required()* method.

---

`ics.ForceFirmwareUpdate()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.force_firmware_update()` method.

---

`ics.GetActiveVNETChannel()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_active_vnet_channel()` method.

---

`ics.GetBackupPowerEnabled()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_backup_power_enabled()` method.

---

`ics.GetBackupPowerReady()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_backup_power_ready()` method.

---

`ics.GetDLLFirmwareInfo()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_dll_firmware_info()` method.

---

`ics.GetDLLVersion()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_dll_version()` method.

---

`ics.GetDeviceStatus()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_device_status()` method.

---

`ics.GetErrorMessages()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_error_messages()` method.

---

`ics.GetFireSettings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_device\_settings()* method.

---

`ics.GetHWFirmwareInfo()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_hw\_firmware\_info()* method.

---

`ics.GetLastError()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_last\_api\_error()* method.

---

`ics.GetMessages()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_messages()* method.

---

`ics.GetPerformanceParameters()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_performance\_parameters()* method.

---

`ics.GetRTC()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_rtc()* method.

---

`ics.GetSerialNumber()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_serial\_number()* method.

---

`ics.GetTimeStampForMsg()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_timestamp\_for\_msg()* method.

---

`ics.GetVCAN3Settings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_device\_settings()* method.

---

`ics.ISO15765_DisableNetworks()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.iso15765_disable_networks()` method.

---

`ics.ISO15765_EnableNetworks()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.iso15765_enable_networks()` method.

---

`ics.ISO15765_ReceiveMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.iso15765_receive_message()` method.

---

`ics.ISO15765_TransmitMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.iso15765_transmit_message()` method.

---

`ics.LoadDefaultSettings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.load_default_settings()` method.

---

`ics.OpenNeoDevice()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.open_device()` method.

---

`ics.ReadSDCard()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.read_sdcard()` method.

---

`ics.RequestEnterSleepMode()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.request_enter_sleep_mode()` method.

---

`ics.ScriptClear()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_clear()` method.

---

`ics.ScriptGetFBlockStatus()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_get_fblock_status()` method.

---

`ics.ScriptGetScriptStatus()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_get_status()` method.

---

`ics.ScriptGetScriptStatusEx()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_script_status()` method.

---

`ics.ScriptLoad()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_load()` method.

---

`ics.ScriptReadAppSignal()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_read_app_signal()` method.

---

`ics.ScriptReadRxMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_read_rx_message()` method.

---

`ics.ScriptReadTxMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_read_tx_message()` method.

---

`ics.ScriptStart()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_start()` method.

---

`ics.ScriptStartFBlock()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_start_fblock()` method.

---

`ics.ScriptStop()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_stop()` method.

---

`ics.ScriptStopFBlock()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_stop_fblock()` method.

---

`ics.ScriptWriteAppSignal()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_write_app_signal()` method.

---

`ics.ScriptWriteRxMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_write_rx_message()` method.

---

`ics.ScriptWriteTxMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_write_tx_message()` method.

---

`ics.SetActiveVNETChannel()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_active_vnet_channel()` method.

---

`ics.SetBackupPowerEnabled()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_backup_power_enabled()` method.

---

`ics.SetBitRate()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_bit_rate()` method.

---

`ics.SetBitRateEx()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_bit_rate_ex()` method.

---

`ics.SetContext()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_context()` method.

---

`ics.SetFDBitRate()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_fd_bit_rate()` method.

---

`ics.SetFireSettings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_device_settings()` method.

---

`ics.SetRTC()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_rtc()` method.

---

`ics.SetReflashDisplayCallback()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_reflash_callback()` method.

---

`ics.SetVCAN3Settings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_device_settings()` method.

---

`ics.TxMessages()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.transmit_messages()` method.

---

`ics.ValidateHObject()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.validate_hobject()` method.

---

`ics.WriteSDCard()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.write_sdcard()` method.

---

`ics.base36enc(serial)`

Converts a decimal serial number to base36.

**Args:** serial (int): serial number.

**Raises:** `ics.RuntimeError`

**Returns:** Str: Serial Number

```
>>> ics.base36enc(device.SerialNumber)
CY0024
```

`ics.close_device(device)`

Closes the device.

**Args:** device (`ics.NeoDevice`): `ics.NeoDevice`

**Raises:** `ics.RuntimeError`

**Returns:** Error Count (int)

```
>>> for device in ics.find_devices():
...     ics.open_device(device)
...     # Do something with the device...
...     ics.close_device(device)
... 
```

---

**Note:** `ics.NeoDevice` will automatically close the device when it goes out of scope.

---

`ics.coremini_clear(device, location)`

Clears the CoreMini into the device.

**Args:** device (`ics.NeoDevice`): `ics.NeoDevice`

location (int): Accepts `ics.SCRIPT_LOCATION_FLASH_MEM`, `ics.SCRIPT_LOCATION_SDCARD`, or `ics.SCRIPT_LOCATION_VCAN3_MEM`

**Raises:** `ics.RuntimeError`

**Returns:** None.



```
>>> device = ics.open_device()
>>> ics.coremini_clear(device, ics.SCRIPT_LOCATION_SDCARD)
```

**ics.coremini\_get\_fblock\_status** (*device*, *index*)  
Gets the status of a Coremini Function Block at *index* on *device*.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*index* (int): Index of the function block.

**Raises:** *ics.RuntimeError*

**Returns:** None on Success.

```
>>> device = ics.open_device()
>>> ics.coremini_get_fblock_status(device, 1)
True
```

**ics.coremini\_get\_status** (*device*)  
Gets the status of the CoreMini in the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** True if running, otherwise False.

```
>>> device = ics.open_device()
>>> ics.coremini_get_status(device)
>>>
```

**ics.coremini\_load** (*device*, *coremini*, *location*)  
Loads the CoreMini into the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*coremini* (str/tuple): Use string to load from file, Use Tuple if file data.

*location* (int): Accepts *ics.SCRIPT\_LOCATION\_FLASH\_MEM*, *ics.SCRIPT\_LOCATION\_SDCARD*, or *ics.SCRIPT\_LOCATION\_VCAN3\_MEM*

**Raises:** *ics.RuntimeError*

**Returns:** None.

```
>>> device = ics.open_device()
>>> ics.coremini_load(device, 'cmvspy.vs3cmb', ics.SCRIPT_LOCATION_SDCARD)
```

**ics.coremini\_read\_app\_signal** (*device*, *index*)  
Gets the value of a Coremini application signal at *index* on *device*.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*index* (int): Index of the application signal.

**Raises:** *ics.RuntimeError*

**Returns:** int on Success.

```
>>> device = ics.open_device()
>>> ics.coremini_read_app_signal(device, 1)
52
```

**ics.coremini\_read\_rx\_message** (*device*, *index*, *j1850=False*)

Gets the value of a Coremini Message at *index* on *device*.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*index* (int): Index of the application signal.

*j1850* (bool): Use *ics.SpyMessageJ1850* instead.

**Raises:** *ics.RuntimeError*

**Returns:** *ics.SpyMessage* Success.

```
>>> device = ics.open_device()
>>> msg = ics.coremini_read_tx_message(device, 0)
```

**ics.coremini\_read\_tx\_message** (*device*, *index*, *j1850=False*)

Gets the value of a Coremini Message at *index* on *device*.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*index* (int): Index of the application signal.

*j1850* (bool): Use *ics.SpyMessageJ1850* instead.

**Raises:** *ics.RuntimeError*

**Returns:** *ics.SpyMessage* Success.

```
>>> device = ics.open_device()
>>> msg = ics.coremini_read_tx_message(device, 0)
```

**ics.coremini\_start** (*device*, *location*)

Starts the CoreMini into the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*location* (int): Accepts *ics.SCRIPT\_LOCATION\_FLASH\_MEM*, *ics.SCRIPT\_LOCATION\_SDCARD*, or *ics.SCRIPT\_LOCATION\_VCAN3\_MEM*

**Raises:** *ics.RuntimeError*

**Returns:** None.

```
>>> device = ics.open_device()
>>> ics.coremini_start(device, ics.SCRIPT_LOCATION_SDCARD)
```

**ics.coremini\_start\_fblock** (*device*, *index*)

Starts a Coremini Function Block at *index* on *device*.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*index* (int): Index of the function block.

**Raises:** *ics.RuntimeError*

**Returns:** None on Success.

```
>>> device = ics.open_device()
>>> ics.coremini_start_fblock(device, 1)
```

**ics.coremini\_stop** (*device*)

Stops the CoreMini into the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** `ics.RuntimeError`

**Returns:** None.

```
>>> device = ics.open_device()
>>> ics.coremini_stop(device)
```

`ics.coremini_stop_fblock` (*device*, *index*)  
Stops a Coremini Function Block at *index* on *device*.

**Args:** *device* (`ics.NeoDevice`): `ics.NeoDevice`  
*index* (int): Index of the function block.

**Raises:** `ics.RuntimeError`

**Returns:** None on Success.

```
>>> device = ics.open_device()
>>> ics.coremini_stop_fblock(device, 1)
```

`ics.coremini_write_app_signal` (*device*, *index*, *value*)  
Sets the value of a Coremini application signal at *index* on *device*.

**Args:** *device* (`ics.NeoDevice`): `ics.NeoDevice`  
*index* (int): Index of the application signal.  
*value* (int): New value of the application signal.

**Raises:** `ics.RuntimeError`

**Returns:** None on Success.

```
>>> device = ics.open_device()
>>> ics.coremini_write_app_signal(device, 1, 52)
>>>
```

`ics.coremini_write_rx_message` (*device*, *index*, *TODO*)  
TODO

`ics.coremini_write_tx_message` (*device*, *index*, *msg*)  
TODO

`ics.create_neovi_radio_message` (*Relay1*, *Relay2*, *Relay3*, *Relay4*, *Relay5*, *LED6*,  
*LED5*, *MSB\_report\_rate*, *LSB\_report\_rate*, *ana-*  
*log\_change\_report\_rate*, *relay\_timeout*)

Python API only. Generates data bytes for use with neoVI RADI/O CAN Messages

**Kwargs:** *Relay1* (boolean): Enable/Disable Relay1

*Relay2* (boolean): Enable/Disable Relay2

*Relay3* (boolean): Enable/Disable Relay3

*Relay4* (boolean): Enable/Disable Relay4

*Relay5* (boolean): Enable/Disable Relay5

*LED5* (boolean): Enable/Disable LED5

*LED6* (boolean): Enable/Disable LED6

*MSB\_report\_rate* (int): MSB Report Rate in ms (0-255)

*LSB\_report\_rate* (int): LSB Report Rate in ms (0-255)

analog\_change\_report\_rate (int): Analog change report rate

relay\_timeout (int): Relay Timeout (0-255)\*255ms

Returns:

Tuple of data bytes for use with `ics.SpyMessage`

Raises: `ics.RuntimeError`

```
>>> msg = ics.SpyMessage()
>>> msg.Data = ics.create_neovi_radio_message(Relay1=True, Relay4=False,
↪ LED6=True, MSB_report_rate=10)
>>> msg.Data
(65, 10, 0, 0, 0)
```

`ics.enable_network_com(device, enable, net_id)`

Enable or disable network communication.

Args: device (`ics.NeoDevice`): `ics.NeoDevice`

enable (bool): bool

net\_id (int): int: Optional. If left blank, disables/enables all networks.

Raises: `ics.RuntimeError`

Returns: None.

```
>>> import ics
>>> d = ics.open_device()
>>> status = ics.enable_network_com(d, True)
>>>
```

`ics.find_devices(device_type=ics.NEODEVICE_ALL)`

Finds all connected devices and returns a tuple of `ics.NeoDevice` for use in `ics.open_device()`

Args: device\_type (int): Accepts `ics.NEODEVICE_*` Macros

*New in 3.0 (803):*

device\_type (List/Tuple): Accepts a Container of `ics.NEODEVICE_*` Macros

stOptionsOpenNeoEx (int): Usually `ics.NETID_CAN`, if needed

Raises: `ics.RuntimeError`

Returns: Tuple of `ics.NeoDevice` for use in `ics.open_device()`

```
>>> for device in ics.find_devices():
...     print(device.Name, device.SerialNumber)
...
neoVI FIRE 59886
```

*New in 3.0 (803):*

```
>>> for device in ics.find_devices([ics.NEODEVICE_FIRE, ics.NEODEVICE_VCAN3]):
...     print(device.Name, device.SerialNumber)
...
neoVI FIRE 59886
```

`ics.firmware_update_required(device)`

Determines if the device firmware needs flashing.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True on success, False on failure.

```
>>> ics.force_firmware_update(device)
True
```

**ics.force\_firmware\_update**(device)

Forces the device to flash firmware.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True on success, False on failure.

```
>>> ics.force_firmware_update(device)
True
```

**ics.get\_active\_vnet\_channel**(device)

Gets active vnet channel for the device.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Int: Returns active vnet channel.

**ics.get\_backup\_power\_enabled**(device)

Returns the device backup power enabled for the device.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True on success, False on failure.

**ics.get\_backup\_power\_ready**(device)

Returns the device backup power is ready for the device.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True on success, False on failure.

**ics.get\_device\_settings**(device, vnet\_slot)

Gets the settings in the device. vnet\_slot defaults to ics.PlasmaIonVnetChannelMain

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** *ics.DeviceSettings*

```
>>> d = ics.open_device()
>>> d.Name
'neoVI ION'
>>> d.SerialNumber
404444
>>> s = ics.get_device_settings(d)
>>> s.DeviceSettingType
2
```

(continues on next page)

(continued from previous page)

```
>>> s.cyan
<ics.CyanSettings object at 0x01E61B40>
>>> s.cyan.canfd1.FDMode
4
>>> s2.cyan
<ics.CyanSettings object at 0x02B113C8>
>>> s2 = ics.get_device_settings(d, ics.PlasmaIonVnetChannelA)
>>> s2.DeviceSettingType
2
>>> s2.cyan.canfd1.FDMode
4
```

**ics.get\_device\_status** (*device*)

Returns the device status.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice***Raises:** *ics.RuntimeError***Returns:** (*ics.IcsDeviceStatus*).

```
>>> import ics
>>> d = ics.open_device()
>>> status = ics.get_device_status(d)
>>> status.fire2Status.ethernetActivationLineEnabled
0
```

**ics.get\_dll\_firmware\_info** (*device*)

Returns the DLL firmware info for the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice***Raises:** *ics.RuntimeError***Returns:** (*ics.ApiFirmwareInfo*)

```
>>> device = ics.open_device()
>>> info = ics.get_dll_firmware_info(device)
>>> info.iAppMajor
7
>>> info.iAppMinor
55
>>>
```

**ics.get\_dll\_version** (*device*)

Gets the DLL version.

**Args:** None**Raises:** *ics.RuntimeError***Returns:** Int: DLL Version

```
>>> ics.get_dll_version()
700
```

**ics.get\_error\_messages** (*device*[, *j1850*, *timeout*])

Gets the error message(s) on the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** list of tuple`s. :class:`tuple` contents: (error\_number, description\_short, description\_long, severity, restart\_needed)

```
>>> device = ics.open_device()
>>> errors = ics.get_error_messages(device)
```

**ics.get\_hw\_firmware\_info(device)**

Returns the device firmware info for the device.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** (*ics.ApiFirmwareInfo*)

```
>>> device = ics.open_device()
>>> info = ics.get_hw_firmware_info(device)
>>> info.iAppMajor
7
>>> info.iAppMinor
55
>>>
```

**ics.get\_last\_api\_error(device)**

Gets the error message from the last API call.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Tuple: (error, description short, description long, severity, restart needed)

```
>>> device = ics.open_device()
>>> try:
...     msg = ics.coremini_read_tx_message(device, 0)
... except ics.RuntimeError as ex:
...     print(ex)
...     print(ics.get_last_api_error(device))
...
Error: coremini_read_tx_message(): icsneoScriptReadTxMessage() Failed
(224, 'Invalid Message Index for script.', 'Invalid Message Index for script.
↳', 16, 0)
```

**ics.get\_library\_path()**

**ics.get\_messages(device[, j1850, timeout])**

Gets the message(s) on the device.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

j1850 (bool): Return *ics.SpyMessageJ1850* instead.

imeout (float): Optional timeout to wait for messages in seconds (0.1 = 100ms).

**Raises:** *ics.RuntimeError*

**Returns:** tuple of two items. First item is a tuple of *ics.SpyMessage* and second is the error count.

```
>>> device = ics.open_device()
>>> messages, errors = ics.get_messages(device)
```

(continues on next page)

(continued from previous page)

```
>>> len(messages)
14
>>> hex(messages[0].ArbIDOrHeader)
'0x160'
>>> messages[0].Data
(36, 11, 11, 177, 37, 3, 11, 199)
>>> errors
0
```

**ics.get\_performance\_parameters** (*device*)

Gets the Performance Parameters on *device*.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Tuple on Success: (buffer count, buffer max, overflow count, reserved, reserved, reserved, reserved, reserved)

```
>>> device = ics.open_device()
>>> ics.get_performance_parameters(device)
(0, 24576, 0, 0, 0, 0, 0, 0)
```

**ics.get\_rtc** (*device*)

Gets the Real-Time Clock of the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Tuple: (datetime.datetime object, offset in seconds)

```
>>> device = ics.open_device()
>>> ics.get_rtc(device)
(datetime.datetime(2014, 9, 10, 17, 45, 45), 3)
```

**ics.get\_script\_status** ()

Accepts a *ics.NeoDevice*, exception on error. Returns a list of values of what *ulParameters* would hold

**ics.get\_serial\_number** (*device*)

Gets the serial number out of the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Int: Serial Number Version

```
>>> ics.get_serial_number(device)
53123
```

**ics.get\_timestamp\_for\_msg** (*device*, *msg*)

Calculates the timestamp for a message.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*msg* (*ics.SpyMessage*): *ics.SpyMessage*

**Raises:** *ics.RuntimeError*

**Returns:** Float: Timestamp for the message.



```
>>> import ics
>>> d = ics.open_device()
>>> msgs, error_count = ics.get_messages(d)
>>> ics.get_timestamp_for_msg(d, msgs[0])
354577568.9145524
```

`ics.icsneoClosePort()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.close_device()` method.

---

`ics.icsneoEnableNetworkCom()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.enable_network_com()` method.

---

`ics.icsneoFindNeoDevices()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.find_devices()` method.

---

`ics.icsneoFirmwareUpdateRequired()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.firmware_update_required()` method.

---

`ics.icsneoForceFirmwareUpdate()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.force_firmware_update()` method.

---

`ics.icsneoGetActiveVNETChannel()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_active_vnet_channel()` method.

---

`ics.icsneoGetBackupPowerEnabled()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_backup_power_enabled()` method.

---

`ics.icsneoGetBackupPowerReady()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_backup\_power\_ready()* method.

---

`ics.icsneoGetDLLFirmwareInfo()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_dll\_firmware\_info()* method.

---

`ics.icsneoGetDLLVersion()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_dll\_version()* method.

---

`ics.icsneoGetDeviceStatus()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_device\_status()* method.

---

`ics.icsneoGetErrorMessages()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_error\_messages()* method.

---

`ics.icsneoGetFireSettings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_device\_settings()* method.

---

`ics.icsneoGetHWFirmwareInfo()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_hw\_firmware\_info()* method.

---

`ics.icsneoGetLastAPIError()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_last\_api\_error()* method.

---

`ics.icsneoGetMessages()`

---

**Note:** Compatibility Function Identical to PEP8 compliant *ics.get\_messages()* method.

---

`ics.icsneoGetPerformanceParameters()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_performance_parameters()` method.

---

`ics.icsneoGetRTC()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_rtc()` method.

---

`ics.icsneoGetSerialNumber()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_serial_number()` method.

---

`ics.icsneoGetTimeStampForMsg()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_timestamp_for_msg()` method.

---

`ics.icsneoGetVCAN3Settings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_device_settings()` method.

---

`ics.icsneoISO15765_DisableNetworks()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.iso15765_disable_networks()` method.

---

`ics.icsneoISO15765_EnableNetworks()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.iso15765_enable_networks()` method.

---

`ics.icsneoISO15765_ReceiveMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.iso15765_receive_message()` method.

---

`ics.icsneoISO15765_TransmitMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.iso15765_transmit_message()` method.

---

`ics.icsneoLoadDefaultSettings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.load_default_settings()` method.

---

`ics.icsneoOpenNeoDevice()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.open_device()` method.

---

`ics.icsneoReadSDCard()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.read_sdcard()` method.

---

`ics.icsneoRequestEnterSleepMode()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.request_enter_sleep_mode()` method.

---

`ics.icsneoScriptClear()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_clear()` method.

---

`ics.icsneoScriptGetFBlockStatus()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_get_fblock_status()` method.

---

`ics.icsneoScriptGetScriptStatus()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_get_status()` method.

---

`ics.icsneoScriptGetScriptStatusEx()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.get_script_status()` method.

---

`ics.icsneoScriptLoad()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_load()` method.

---

`ics.icsneoScriptReadAppSignal()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_read_app_signal()` method.

---

`ics.icsneoScriptReadRxMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_read_rx_message()` method.

---

`ics.icsneoScriptReadTxMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_read_tx_message()` method.

---

`ics.icsneoScriptStart()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_start()` method.

---

`ics.icsneoScriptStartFBlock()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_start_fblock()` method.

---

`ics.icsneoScriptStop()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_stop()` method.

---

`ics.icsneoScriptStopFBlock()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_stop_fblock()` method.

---

`ics.icsneoScriptWriteAppSignal()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_write_app_signal()` method.

---

`ics.icsneoScriptWriteRxMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_write_rx_message()` method.

---

`ics.icsneoScriptWriteTxMessage()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.coremini_write_tx_message()` method.

---

`ics.icsneoSetActiveVNETChannel()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_active_vnet_channel()` method.

---

`ics.icsneoSetBackupPowerEnabled()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_backup_power_enabled()` method.

---

`ics.icsneoSetBitRate()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_bit_rate()` method.

---

`ics.icsneoSetBitRateEx()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_bit_rate_ex()` method.

---

`ics.icsneoSetContext()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_context()` method.

---

`ics.icsneoSetFDBitRate()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_fd_bit_rate()` method.

---

`ics.icsneoSetFireSettings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_device_settings()` method.

---

`ics.icsneoSetRTC()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_rtc()` method.

---

`ics.icsneoSetReflashDisplayCallbacks()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_reflash_callback()` method.

---

`ics.icsneoSetVCAN3Settings()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.set_device_settings()` method.

---

`ics.icsneoTxMessages()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.transmit_messages()` method.

---

`ics.icsneoValidateHObject()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.validate_hobject()` method.

---

`ics.icsneoWriteSDCard()`

---

**Note:** Compatibility Function Identical to PEP8 compliant `ics.write_sdcard()` method.

---

`ics.iso15765_disable_networks(device)`

Disables ISO15765 networks.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** None

**ics.iso15765\_enable\_networks** (*device, networks*)  
Enables ISO15765 networks.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** None

**ics.iso15765\_receive\_message** (*device, netid, rx\_msg*)  
Setup rx ISO15765 Message.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

prx\_msg (*ics.CmISO157652RxMessage*): *ics.CmISO157652RxMessage*

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True on success, False on failure.

**ics.iso15765\_transmit\_message** (*device, ulNetworkID, pMsg, ulBlockingTimeout*)  
Transmits an ISO15765 Message.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

pMsg (*ics.CmISO157652TxMessage*): *ics.CmISO157652TxMessage*

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True on success, False on failure.

**ics.load\_default\_settings** (*device*)  
Load the default settings in the device.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** None.

```
>>> device = ics.open_device()
>>> settings = ics.load_default_settings(device)
>>>
```

**ics.open\_device** (*device*)  
Opens the device. *device* can be omitted to return a *ics.NeoDevice* of the first free available device, a *ics.NeoDevice*, or a serial number of the device.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

device (int): Serial Number of the device

bNetworkIDs (int): Network Enables

bConfigRead (int): Config Read

iOptions (int): DEVICE\_OPTION\_\* defines

stOptionsOpenNeoEx (int): Usually ics.NETID\_CAN, if needed

**Raises:** *ics.RuntimeError*

**Returns:** If *ics.NeoDevice* is passed as a parameter, None. If serial number is passed as a parameter, a *ics.NeoDevice* will be returned. If *device* parameter is omitted, a *ics.NeoDevice* will be returned with the first available free device.



```
>>> for device in ics.find_devices():
...     ics.open_device(device)
... 
```

---

**Note:** `ics.NeoDevice` will automatically close the device when it goes out of scope.

---

`ics.override_library_name(new_name)`

Sets active vnet channel for the device.

**Args:** name: Absolute path or relative path including filename.

**Raises:** `ics.RuntimeError`

**Returns:** None

```
>>> import ics
>>> ics.find_devices()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ics.RuntimeError: Error: find_devices(): Failed to open library: 'icsneo40.dll
↳' with error code: #126
>>> ics.override_library_name(r"C:\Windows\SysWOW64\icsneo40-different.dll")
>>> ics.find_devices()
(<ics.NeoDevice object at 0x00284C50>, <ics.NeoDevice object at 0x007C9A10>)
```

`ics.read_sdcard()`

`icsneoReadSDCard()`, Accepts a `ics.NeoDevice` and sector index. Returns a bytearray of 512 bytes max. Exception on error.

`ics.request_enter_sleep_mode(device, timeout_ms, mode, reserved_zero)`

Signal neoVI to immediate go to sleep. Currently only supported by FIREVNET/PLASMA. If using over USB this will likely return true but never cause PLASMA to sleep since USB insertion keeps it alive. This API allows Android/Linux applications to invoke power management.

**Args:** device (`ics.NeoDevice`): `ics.NeoDevice`

timeout\_ms (int): 16bit word for how long to wait on idle bus before going to sleep. If caller does not want to change it pass in 65535 (0xFFFF) and it will stay whatever it was set to in explorer/coremini.

mode (int): 16bit word for power mode to enter. If caller does not want to change it pass in 65535 (0xFFFF) and it will stay whatever it was set to in explorer/coremini. If it is zero then neoVI will do 'normal sleep'. 0 - power mode off but calling this function will do 'normal sleep'. 1 - normal sleep. 2 - instant sleep. 3 - comatose sleep.

reserved\_zero (int): Reserved, Keep as zero.

**Raises:** `ics.RuntimeError`

**Returns:** Boolean: True on success, False on failure.

```
>>> ics.request_enter_sleep_mode(device, 1, 0)
True
```

`ics.set_active_vnet_channel(device, channel)`

Sets active vnet channel for the device.

**Args:** device (`ics.NeoDevice`): `ics.NeoDevice`

**Raises:** `ics.RuntimeError`

**Returns:** Boolean: True on success, False on failure.

`ics.set_backup_power_enabled(device, enable)`  
Sets the device backup power enabled for the device.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True on success, False on failure.

`ics.set_bit_rate(device, BitRate, NetworkID)`  
Sets the bitrate for a given Network ID on the device..

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Int: None.

`ics.set_bit_rate_ex()`  
`ics.set_fd_bit_rate_ex(device, BitRate, NetworkID, iOptions)`  
Sets the bitrate for a given Network ID on the device with extended options.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Int: None.

`ics.set_context(device)`  
Sets the “context” of how `icsneoFindNeoDevices(Ex)` and `icsneoOpenNeoDevice(Ex)` function. If the context is 0 (null) than `icsneoFindNeoDevices(Ex)` will be system wide, searching USB and other supported computer interfaces. `icsneoFindNeoDevices` can then be used to connect to devices found in this manner. If the context is a handle to connected CAN tool than `icsneoFindNeoDevices(Ex)` will search a specific CAN bus for supported IntrepidCS CAN Nodes. Again `icsneoOpenNeoDevice(Ex)` would be used create logical connections to found CAN Nodes.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True on success, False on failure.

```
>>> ics.set_context(device)
True
```

`ics.set_device_settings(device, settings, save_to_eeprom, vnet_slot)`  
Sets the settings in the device. `vnet_slot` defaults to `ics.PlasmaIonVnetChannelMain`

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

`settings` (*ics.DeviceSettings*): *ics.DeviceSettings*

**Raises:** *ics.RuntimeError*

**Returns:** None.

```
>>> d = ics.open_device()
>>> d.Name
'neoVI ION'
>>> d.SerialNumber
404444
>>> s = ics.get_device_settings(d, ics.PlasmaIonVnetChannelA) # Get Slave_
↪ settings, channel selection not needed if not a Plasma/Ion (continues on next page)
```

(continued from previous page)

```

>>> s.DeviceSettingType
2
>>> s.cyan.can_switch_mode
1
>>> s.cyan.can_switch_mode = 2
>>> ics.set_device_settings(d, s, True, ics.PlasmaIonVnetChannelA)
>>>

```

**ics.set\_fd\_bit\_rate** (*device*, *BitRate*, *NetworkID*)  
Sets the FD bitrate for a given Network ID on the device..

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

**Raises:** *ics.RuntimeError*

**Returns:** Int: None.

**ics.set\_reflash\_callback** (*callback*)  
Sets the reflash display callback.

**Args:** *callback* (function): Must be a callable Python function (*def callback(msg, progress)*)

**Raises:** *ics.RuntimeError*

**Returns:** None.

```

>>> def callback(msg, progress):
...     print(msg, progress)
...
>>> ics.set_reflash_callback(callback)
>>>

```

**ics.set\_rtc** (*device*[, *time* ])  
Sets the Real-Time Clock of the device.

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*ime* (*datetime.datetime*): Optional. Sets to current time, if omitted.

**Raises:** *ics.RuntimeError*

**Returns:** None.

```

>>> device = ics.open_device()
>>> ics.set_rtc(device)

```

**ics.transmit\_messages** (*device*, *messages*)  
Transmits message(s) on the device. *messages* can be a tuple of *ics.SpyMessage*

**Args:** *device* (*ics.NeoDevice*): *ics.NeoDevice*

*messages* (*ics.SpyMessage*): *ics.SpyMessage*

**Raises:** *ics.RuntimeError*

**Returns:** None.

```

>>> device = ics.open_device()
>>> msg = ics.SpyMessage()
>>> msg.ArbIDOrHeader = 0xFF
>>> msg.NetworkID = ics.NETID_HSCAN

```

(continues on next page)

(continued from previous page)

```
>>> msg.Data = (0,1,2,3,4,5,6,7)
>>> ics.transmit_messages(device, msg)
>>>
```

**ics.validate\_hobject** (*device*)

Validates the handle is valid for a *device*. Handles are only valid when the device is open.

**Args:** device (*ics.NeoDevice*): *ics.NeoDevice*

or:

device (int): c style integer handle to the device.

**Raises:** *ics.RuntimeError*

**Returns:** Boolean: True if valid, false otherwise.

```
>>> device = ics.open_device()
>>> ics.validate_hobject(device)
1
>>> ics.validate_hobject(device._Handle)
1
```

**ics.write\_sdcard** ()

icsneoReadSDCard(), Accepts a ics.NeoDevice, sector index, and a bytearray of 512 bytes. Exception on error.

## CHAPTER 9

---

### Module Variables

---

```
ics.AUTO = 0
ics.BPS100 = 5
ics.BPS1000 = 10
ics.BPS100000 = 7
ics.BPS10400 = 1
ics.BPS117647 = 8
ics.BPS125 = 6
ics.BPS20 = 0
ics.BPS2000 = 12
ics.BPS250 = 7
ics.BPS33 = 1
ics.BPS33333 = 2
ics.BPS4000 = 13
ics.BPS50 = 2
ics.BPS500 = 8
ics.BPS5000 = 0
ics.BPS50000 = 3
ics.BPS62 = 3
ics.BPS62500 = 4
ics.BPS666 = 11
ics.BPS71429 = 5
ics.BPS800 = 9
```

```
ics.BPS83 = 4
ics.BPS83333 = 6
ics.BUILD_DATETIME = May 29 2018 16:04:26
ics.CANFD_BRS_ENABLED = 2
ics.CANFD_BRS_ENABLED_ISO = 4
ics.CANFD_ENABLED = 1
ics.CANFD_ENABLED_ISO = 3
ics.CANFD_SETTINGS_SIZE = 10
ics.CANTERM_SETTINGS_SIZE = 6
ics.CAN_BPS10000 = 17
ics.CAN_BPS5000 = 14
ics.CAN_BPS6667 = 15
ics.CAN_BPS8000 = 16
ics.CAN_SETTINGS_SIZE = 12
ics.DISABLE = 1
ics.ETHERNET_SETTINGS_SIZE = 8
ics.FAST_MODE = 3
ics.GLOBAL_SETTINGS_SIZE = 908
ics.GS_VERSION = 5
ics.ISO15765_2_NETWORK_HSCAN = 1
ics.ISO15765_2_NETWORK_HSCAN2 = 4
ics.ISO15765_2_NETWORK_HSCAN3 = 8
ics.ISO15765_2_NETWORK_HSCAN4 = 20
ics.ISO15765_2_NETWORK_HSCAN5 = 24
ics.ISO15765_2_NETWORK_HSCAN6 = 28
ics.ISO15765_2_NETWORK_HSCAN7 = 32
ics.ISO15765_2_NETWORK_MSCAN = 2
ics.ISO15765_2_NETWORK_SWCAN = 16
ics.ISO15765_2_NETWORK_SWCAN2 = 36
ics.ISO9141_KEYWORD2000_SETTINGS_SIZE = 114
ics.ISO9141_KEYWORD2000__INIT_STEP_SIZE = 6
ics.J1708_SETTINGS_SIZE = 2
ics.LIN_SETTINGS_SIZE = 10
ics.LISTEN_ALL = 7
ics.LISTEN_ONLY = 3
ics.LOOPBACK = 2
```

```
ics.NEODEVICE_ALL = -16385
ics.NEODEVICE_ANY_ION = 1310720
ics.NEODEVICE_ANY_PLASMA = 208896
ics.NEODEVICE_BLUE = 1
ics.NEODEVICE_CMPROBE = 8388608
ics.NEODEVICE_CT_OBD = 32768
ics.NEODEVICE_DW_VCAN = 4
ics.NEODEVICE_ECU = 128
ics.NEODEVICE_ECUCHIP_UART = 2048
ics.NEODEVICE_EEVB = 16777216
ics.NEODEVICE_FIRE = 8
ics.NEODEVICE_FIRE2 = 67108864
ics.NEODEVICE_FIRE_VNET = 8192
ics.NEODEVICE_FLEX = 134217728
ics.NEODEVICE_IEVB = 256
ics.NEODEVICE_ION_2 = 262144
ics.NEODEVICE_ION_3 = 1048576
ics.NEODEVICE_NEOANALOG = 16384
ics.NEODEVICE_NEOECUCHIP = 256
ics.NEODEVICE_OBD2_PRO = 1024
ics.NEODEVICE_OBD2_SIM = -2147483648
ics.NEODEVICE_PENDANT = 512
ics.NEODEVICE_PLASMA_1_11 = 4096
ics.NEODEVICE_PLASMA_1_12 = 65536
ics.NEODEVICE_PLASMA_1_13 = 131072
ics.NEODEVICE_RADGALAXY = 268435456
ics.NEODEVICE_RADSTAR = 524288
ics.NEODEVICE_RADSTAR2 = 536870912
ics.NEODEVICE_RED = 64
ics.NEODEVICE_SW_VCAN = 2
ics.NEODEVICE_UNKNOWN = 0
ics.NEODEVICE_VCAN3 = 16
ics.NEODEVICE_VCAN4 = 2097152
ics.NEODEVICE_VCAN4_12 = 4194304
ics.NEODEVICE_VCANRF = 33554432
ics.NEODEVICE_VIVIDCAN = 1073741824
```

```
ics.NEOVI6_VCAN_TIMESTAMP_1 = 1e-06
ics.NEOVI6_VCAN_TIMESTAMP_2 = 0.065536
ics.NEOVIPRO_VCAN_TIMESTAMP_1 = 1e-06
ics.NEOVIPRO_VCAN_TIMESTAMP_2 = 0.065536
ics.NEOVI_3G_MAX_SETTINGS_SIZE = 908
ics.NEOVI_COMMTYPE_FIRE_USB = 5
ics.NEOVI_COMMTYPE_RS232 = 0
ics.NEOVI_COMMTYPE_TCPIP = 3
ics.NEOVI_COMMTYPE_USB_BULK = 1
ics.NEOVI_RED_TIMESTAMP_1_10NS = 1e-08
ics.NEOVI_RED_TIMESTAMP_1_25NS = 2.5e-08
ics.NEOVI_RED_TIMESTAMP_2_10NS = 429.4967296
ics.NEOVI_RED_TIMESTAMP_2_25NS = 107.3741824
ics.NEOVI_TIMESTAMP_1 = 1.6e-06
ics.NEOVI_TIMESTAMP_2 = 0.1048576
ics.NEO_CFG_MPIC_HS_CAN_CNF1 = 522
ics.NEO_CFG_MPIC_HS_CAN_CNF2 = 521
ics.NEO_CFG_MPIC_HS_CAN_CNF3 = 520
ics.NEO_CFG_MPIC_HS_CAN_MODE = 566
ics.NEO_CFG_MPIC_LSFT_CAN_CNF1 = 558
ics.NEO_CFG_MPIC_LSFT_CAN_CNF2 = 557
ics.NEO_CFG_MPIC_LSFT_CAN_CNF3 = 556
ics.NEO_CFG_MPIC_MS_CAN_CNF1 = 534
ics.NEO_CFG_MPIC_MS_CAN_CNF2 = 533
ics.NEO_CFG_MPIC_MS_CAN_CNF3 = 532
ics.NEO_CFG_MPIC_SW_CAN_CNF1 = 546
ics.NEO_CFG_MPIC_SW_CAN_CNF2 = 545
ics.NEO_CFG_MPIC_SW_CAN_CNF3 = 544
ics.NETID_3G_APP_SIGNAL_STATUS = 56
ics.NETID_3G_FB_STATUS = 55
ics.NETID_3G_LOGGING_OVERFLOW = 59
ics.NETID_3G_READ_DATA LINK_CM_RX_MSG = 58
ics.NETID_3G_READ_DATA LINK_CM_TX_MSG = 57
ics.NETID_3G_READ_SETTINGS_EX = 60
ics.NETID_3G_RESET_STATUS = 54
ics.NETID_AUX = 7
```



```
ics.NETID_CGI = 53
ics.NETID_DATA_TO_HOST = 70
ics.NETID_DEVICE = 0
ics.NETID_DEVICE_STATUS = 513
ics.NETID_ETHERNET = 93
ics.NETID_ETHERNET_DAQ = 69
ics.NETID_FLEXRAY = 85
ics.NETID_FLEXRAY1A = 80
ics.NETID_FLEXRAY1B = 81
ics.NETID_FLEXRAY2 = 86
ics.NETID_FLEXRAY2A = 82
ics.NETID_FLEXRAY2B = 83
ics.NETID_FORDSCP = 5
ics.NETID_GMFSA = 94
ics.NETID_HSCAN = 1
ics.NETID_HSCAN2 = 42
ics.NETID_HSCAN3 = 44
ics.NETID_HSCAN4 = 61
ics.NETID_HSCAN5 = 62
ics.NETID_HSCAN6 = 96
ics.NETID_HSCAN7 = 97
ics.NETID_HW_COM_LATENCY_TEST = 512
ics.NETID_I2C1 = 71
ics.NETID_INVALID = 65535
ics.NETID_ISO = 9
ics.NETID_ISO14230 = 15
ics.NETID_ISO2 = 14
ics.NETID_ISO3 = 41
ics.NETID_ISO4 = 47
ics.NETID_ISOPIC = 10
ics.NETID_J1708 = 6
ics.NETID_JVPW = 8
ics.NETID_LIN = 16
ics.NETID_LIN2 = 48
ics.NETID_LIN3 = 49
ics.NETID_LIN4 = 50
```

```
ics.NETID_LIN5 = 84
ics.NETID_LIN6 = 98
ics.NETID_LSFTCAN = 4
ics.NETID_LSFTCAN2 = 99
ics.NETID_MAIN51 = 11
ics.NETID_MAX = 100
ics.NETID_MOST = 51
ics.NETID_MOST150 = 92
ics.NETID_MOST25 = 90
ics.NETID_MOST50 = 91
ics.NETID_MSCAN = 2
ics.NETID_OP_ETHERNET1 = 17
ics.NETID_OP_ETHERNET10 = 78
ics.NETID_OP_ETHERNET11 = 79
ics.NETID_OP_ETHERNET12 = 87
ics.NETID_OP_ETHERNET2 = 18
ics.NETID_OP_ETHERNET3 = 19
ics.NETID_OP_ETHERNET4 = 45
ics.NETID_OP_ETHERNET5 = 46
ics.NETID_OP_ETHERNET6 = 73
ics.NETID_OP_ETHERNET7 = 75
ics.NETID_OP_ETHERNET8 = 76
ics.NETID_OP_ETHERNET9 = 77
ics.NETID_RED = 12
ics.NETID_RED_APP_ERROR = 52
ics.NETID_RED_VBAT = 74
ics.NETID_RS232 = 63
ics.NETID_SCI = 13
ics.NETID_SPI1 = 72
ics.NETID_SWCAN = 3
ics.NETID_SWCAN2 = 68
ics.NETID_TCP = 95
ics.NETID_TEXTAPI_TO_HOST = 71
ics.NETID_UART = 64
ics.NETID_UART2 = 65
ics.NETID_UART3 = 66
```

```
ics.NETID_UART4 = 67
ics.NORMAL = 0
ics.NORMAL_MODE = 2
ics.NO_CANFD = 0
ics.OPETH_FUNC_MEDIACONVERTER = 1
ics.OPETH_FUNC_TAP = 0
ics.OPETH_FUNC_TAP_LOW_LATENCY = 2
ics.OPETH_LINK_AUTO = 0
ics.OPETH_LINK_MASTER = 1
ics.OPETH_LINK_SLAVE = 2
ics.OPETH_MAC_SPOOF_DST_ADDR = 0
ics.OPETH_MAC_SPOOF_SRC_ADDR = 1
ics.OP_ETH_GENERAL_SETTINGS_SIZE = 20
ics.OP_ETH_SETTINGS_SIZE = 16
ics.PLASMA_SLAVE1_OFFSET = 100
ics.PLASMA_SLAVE1_OFFSET_RANGE2 = 4608
ics.PLASMA_SLAVE2_OFFSET = 200
ics.PLASMA_SLAVE2_OFFSET_RANGE2 = 8704
ics.PLASMA_SLAVE3_OFFSET_RANGE2 = 12800
ics.PLASMA_SLAVE_NUM = 51
ics.REPORT_ON_GPS = 15
ics.REPORT_ON_KLINE = 9
ics.REPORT_ON_LED1 = 7
ics.REPORT_ON_LED2 = 8
ics.REPORT_ON_MISC1 = 1
ics.REPORT_ON_MISC2 = 2
ics.REPORT_ON_MISC3 = 3
ics.REPORT_ON_MISC3_AIN = 10
ics.REPORT_ON_MISC4 = 4
ics.REPORT_ON_MISC4_AIN = 11
ics.REPORT_ON_MISC5 = 5
ics.REPORT_ON_MISC5_AIN = 12
ics.REPORT_ON_MISC6 = 6
ics.REPORT_ON_MISC6_AIN = 13
ics.REPORT_ON_PERIODIC = 0
ics.REPORT_ON_PWM_IN1 = 14
```

```
ics.RESISTOR_OFF = 1
ics.RESISTOR_ON = 0
ics.SCRIPT_LOCATION_FLASH_MEM = 0
ics.SCRIPT_LOCATION_INTERNAL_FLASH = 2
ics.SCRIPT_LOCATION_SDCARD = 1
ics.SCRIPT_LOCATION_VCAN3_MEM = 4
ics.SCRIPT_STATUS_RUNNING = 1
ics.SCRIPT_STATUS_STOPPED = 0
ics.SLEEP_MODE = 0
ics.SLOW_MODE = 1
ics.SPY_PROTOCOL_BEAN = 11
ics.SPY_PROTOCOL_CAN = 1
ics.SPY_PROTOCOL_CANFD = 30
ics.SPY_PROTOCOL_CGI = 18
ics.SPY_PROTOCOL_CHRYSLER_CCD = 8
ics.SPY_PROTOCOL_CHRYSLER_JVPW = 14
ics.SPY_PROTOCOL_CHRYSLER_SCI = 9
ics.SPY_PROTOCOL_CUSTOM = 0
ics.SPY_PROTOCOL_DALLAS_1WIRE = 25
ics.SPY_PROTOCOL_ETHERNET = 29
ics.SPY_PROTOCOL_FLEXRAY = 16
ics.SPY_PROTOCOL_FORD_UBP = 10
ics.SPY_PROTOCOL_GENERIC_MANCHSESTER = 26
ics.SPY_PROTOCOL_GENERIC_UART = 22
ics.SPY_PROTOCOL_GME_CIM_SCL_KLINE = 19
ics.SPY_PROTOCOL_GMFSA = 31
ics.SPY_PROTOCOL_GMLAN = 2
ics.SPY_PROTOCOL_GM_ALDL_UART = 7
ics.SPY_PROTOCOL_I2C = 21
ics.SPY_PROTOCOL_ISO9141 = 5
ics.SPY_PROTOCOL_J1708 = 13
ics.SPY_PROTOCOL_J1850PWM = 4
ics.SPY_PROTOCOL_J1850VPW = 3
ics.SPY_PROTOCOL_J1939 = 15
ics.SPY_PROTOCOL_JTAG = 23
ics.SPY_PROTOCOL_LIN = 12
```

```
ics.SPY_PROTOCOL_MOST = 17
ics.SPY_PROTOCOL_SENT_PROTOCOL = 27
ics.SPY_PROTOCOL_SPI = 20
ics.SPY_PROTOCOL_TCP = 32
ics.SPY_PROTOCOL_UART = 28
ics.SPY_PROTOCOL_UNIO = 24
ics.SPY_STATUS2_CAN_HAVE_LINK_DATA = 4194304
ics.SPY_STATUS2_CAN_ISO15765_LOGICAL_FRAME = 2097152
ics.SPY_STATUS2_END_OF_LONG_MESSAGE = 1048576
ics.SPY_STATUS2_ERROR_FRAME = 131072
ics.SPY_STATUS2_ETHERNET_CRC_ERROR = 2097152
ics.SPY_STATUS2_ETHERNET_FCS_AVAILABLE = 8388608
ics.SPY_STATUS2_ETHERNET_FRAME_TOO_SHORT = 4194304
ics.SPY_STATUS2_ETHERNET_NO_PADDING = 16777216
ics.SPY_STATUS2_ETHERNET_PREEMPTION_ENABLED = 33554432
ics.SPY_STATUS2_FLEXRAY_NO_CRC = 33554432
ics.SPY_STATUS2_FLEXRAY_NO_HEADERCRC = 67108864
ics.SPY_STATUS2_FLEXRAY_TX_AB = 2097152
ics.SPY_STATUS2_FLEXRAY_TX_AB_NO_A = 4194304
ics.SPY_STATUS2_FLEXRAY_TX_AB_NO_B = 8388608
ics.SPY_STATUS2_FLEXRAY_TX_AB_NO_MATCH = 16777216
ics.SPY_STATUS2_GLOBAL_CHANGE = 65536
ics.SPY_STATUS2_HAS_VALUE = 1
ics.SPY_STATUS2_HIGH_VOLTAGE = 4
ics.SPY_STATUS2_ISO_FRAME_ERROR = 134217728
ics.SPY_STATUS2_ISO_OVERFLOW_ERROR = 268435456
ics.SPY_STATUS2_ISO_PARITY_ERROR = 536870912
ics.SPY_STATUS2_LIN_ERR_MSG_ID_PARITY = 67108864
ics.SPY_STATUS2_LIN_ERR_RX_BREAK_NOT_0 = 2097152
ics.SPY_STATUS2_LIN_ERR_RX_BREAK_TOO_SHORT = 4194304
ics.SPY_STATUS2_LIN_ERR_RX_DATA_GREATER_8 = 16777216
ics.SPY_STATUS2_LIN_ERR_RX_SYNC_NOT_55 = 8388608
ics.SPY_STATUS2_LIN_ERR_TX_RX_MISMATCH = 33554432
ics.SPY_STATUS2_LIN_ID_FRAME_ERROR = 268435456
ics.SPY_STATUS2_LIN_NO_SLAVE_DATA = -2147483648
ics.SPY_STATUS2_LIN_SLAVE_BYTE_ERROR = 536870912
```

```
ics.SPY_STATUS2_LIN_SYNC_FRAME_ERROR = 134217728
ics.SPY_STATUS2_LONG_MESSAGE = 8
ics.SPY_STATUS2_MOST_CHANGED_PAR = -2147483648
ics.SPY_STATUS2_MOST_CONTROL_DATA = 16777216
ics.SPY_STATUS2_MOST_I2S_DUMP = 134217728
ics.SPY_STATUS2_MOST_LOW_LEVEL = 8388608
ics.SPY_STATUS2_MOST_MHP_CONTROL_DATA = 67108864
ics.SPY_STATUS2_MOST_MHP_USER_DATA = 33554432
ics.SPY_STATUS2_MOST_MOST150 = 1073741824
ics.SPY_STATUS2_MOST_MOST50 = 536870912
ics.SPY_STATUS2_MOST_PACKET_DATA = 2097152
ics.SPY_STATUS2_MOST_TOO_SHORT = 268435456
ics.SPY_STATUS2_RX_TIMEOUT_ERROR = 1073741824
ics.SPY_STATUS2_VALUE_IS_BOOLEAN = 2
ics.SPY_STATUS3_CANFD_BRS = 16
ics.SPY_STATUS3_CANFD_ESI = 1
ics.SPY_STATUS3_CANFD_FDF = 8
ics.SPY_STATUS3_CANFD_IDE = 2
ics.SPY_STATUS3_CANFD_RTR = 4
ics.SPY_STATUS3_LIN_JUST_BREAK_SYNC = 1
ics.SPY_STATUS3_LIN_ONLY_UPDATE_SLAVE_TABLE_ONCE = 4
ics.SPY_STATUS3_LIN_SLAVE_DATA_TOO_SHORT = 2
ics.SPY_STATUS_ANALOG_DIGITAL_INPUT = 16777216
ics.SPY_STATUS_AUDIO_COMMENT = 4194304
ics.SPY_STATUS_AVSI_REC_OVERFLOW = 1048576
ics.SPY_STATUS_BAD_MESSAGE_BIT_TIME_ERROR = 16384
ics.SPY_STATUS_BREAK = 524288
ics.SPY_STATUS_BUS_RECOVERED = 1024
ics.SPY_STATUS_BUS_SHORTED_GND = 4096
ics.SPY_STATUS_BUS_SHORTED_PLUS = 2048
ics.SPY_STATUS_CANFD = 536870912
ics.SPY_STATUS_CAN_BUS_OFF = 512
ics.SPY_STATUS_CAN_ERROR_PASSIVE = 32
ics.SPY_STATUS_CHECKSUM_ERROR = 8192
ics.SPY_STATUS_COMM_IN_OVERFLOW = 65536
ics.SPY_STATUS_CRC_ERROR = 16
```

```
ics.SPY_STATUS_EXPECTED_LEN_MISMATCH = 131072
ics.SPY_STATUS_EXTENDED = -2147483648
ics.SPY_STATUS_FLEXRAY_PDU = 536870912
ics.SPY_STATUS_FLEXRAY_PDU_NO_UPDATE_BIT = 8
ics.SPY_STATUS_FLEXRAY_PDU_UPDATE_BIT_SET = 1073741824
ics.SPY_STATUS_GLOBAL_ERR = 1
ics.SPY_STATUS_GPS_DATA = 8388608
ics.SPY_STATUS_HEADERCRC_ERROR = 32
ics.SPY_STATUS_HIGH_SPEED = 1073741824
ics.SPY_STATUS_INCOMPLETE_FRAME = 64
ics.SPY_STATUS_INIT_MESSAGE = 536870912
ics.SPY_STATUS_LIN_MASTER = 536870912
ics.SPY_STATUS_LOST_ARBITRATION = 128
ics.SPY_STATUS_MSG_NO_MATCH = 262144
ics.SPY_STATUS_NETWORK_MESSAGE_TYPE = 67108864
ics.SPY_STATUS_PDU = 536870912
ics.SPY_STATUS_REMOTE_FRAME = 8
ics.SPY_STATUS_TEST_TRIGGER = 2097152
ics.SPY_STATUS_TEXT_COMMENT = 33554432
ics.SPY_STATUS_TX_MSG = 2
ics.SPY_STATUS_TX_NOMATCH = 32768
ics.SPY_STATUS_UNDEFINED_ERROR = 256
ics.SPY_STATUS_VSI_IFR_CRC_BIT = 268435456
ics.SPY_STATUS_VSI_TX_UNDERRUN = 134217728
ics.SPY_STATUS_XTD_FRAME = 4
ics.SWCAN_AUTOSWITCH_DISABLED = 0
ics.SWCAN_AUTOSWITCH_DISABLED_RESISTOR_ENABLED = 3
ics.SWCAN_AUTOSWITCH_NO_RESISTOR = 1
ics.SWCAN_AUTOSWITCH_WITH_RESISTOR = 2
ics.SWCAN_SETTINGS_SIZE = 14
ics.UART_SETTINGS_SIZE = 16
ics.USE_TQ = 1
ics.VNETBITS_FEATURE_ANDROID_MSGS = 1
ics.VNETBITS_FEATURE_DISABLE_USB_CHECK = 2
```





**i**

`ics`, [23](#)



## A

AckBytes (*ics.SpyMessage* attribute), 38  
 AckBytes (*ics.SpyMessageJ1850* attribute), 39  
 ain\_sample\_period (*ics.CyanSettings* attribute), 27  
 ain\_sample\_period (*ics.FireSettings* attribute), 30  
 ain\_sample\_period (*ics.RadGalaxySettings* attribute), 34  
 ain\_sample\_period (*ics.RadStar2Settings* attribute), 36  
 ain\_threshold (*ics.CyanSettings* attribute), 27  
 ain\_threshold (*ics.FireSettings* attribute), 30  
 ain\_threshold (*ics.RadGalaxySettings* attribute), 34  
 ain\_threshold (*ics.RadStar2Settings* attribute), 36  
 ApiFirmwareInfo (class in *ics*), 24  
 ArbIDOrHeader (*ics.SpyMessage* attribute), 38  
 ArgumentError, 23  
 AUTO (in module *ics*), 73  
 auto\_baud (*ics.CanSettings* attribute), 25  
 auto\_baud (*ics.SWCanSettings* attribute), 38  
 auto\_neg (*ics.EthernetSettings* attribute), 30  
 AutoHandleClose (*ics.NeoDevice* attribute), 33

## B

backupPowerEnabled (*ics.Fire2DeviceStatus* attribute), 30  
 backupPowerGood (*ics.Fire2DeviceStatus* attribute), 30  
 base36enc () (in module *ics*), 52  
 Baudrate (*ics.CanSettings* attribute), 25  
 Baudrate (*ics.Iso9141Keyword2000Settings* attribute), 32  
 Baudrate (*ics.LinSettings* attribute), 33  
 Baudrate (*ics.SWCanSettings* attribute), 37  
 Baudrate (*ics.UartSettings* attribute), 41  
 bEnReportLinkQuality (*ics.OpEthGeneralSettings* attribute), 33  
 blockSize (*ics.CmISO157652RxMessage* attribute), 25

blockSize (*ics.CmISO157652TxMessage* attribute), 26  
 bOptions (*ics.UartSettings* attribute), 41  
 BPS100 (in module *ics*), 73  
 BPS1000 (in module *ics*), 73  
 BPS100000 (in module *ics*), 73  
 BPS10400 (in module *ics*), 73  
 BPS117647 (in module *ics*), 73  
 BPS125 (in module *ics*), 73  
 BPS20 (in module *ics*), 73  
 BPS2000 (in module *ics*), 73  
 BPS250 (in module *ics*), 73  
 BPS33 (in module *ics*), 73  
 BPS33333 (in module *ics*), 73  
 BPS4000 (in module *ics*), 73  
 BPS50 (in module *ics*), 73  
 BPS500 (in module *ics*), 73  
 BPS5000 (in module *ics*), 73  
 BPS50000 (in module *ics*), 73  
 BPS62 (in module *ics*), 73  
 BPS62500 (in module *ics*), 73  
 BPS666 (in module *ics*), 73  
 BPS71429 (in module *ics*), 73  
 BPS800 (in module *ics*), 73  
 BPS83 (in module *ics*), 73  
 BPS83333 (in module *ics*), 74  
 brgh (*ics.Iso9141Keyword2000Settings* attribute), 32  
 brgh (*ics.LinSettings* attribute), 33  
 brgh (*ics.UartSettings* attribute), 41  
 BRP (*ics.CanSettings* attribute), 25  
 BRP (*ics.SWCanSettings* attribute), 37  
 bTapEnPtp (*ics.OpEthGeneralSettings* attribute), 33  
 bTapEnSwitch (*ics.OpEthGeneralSettings* attribute), 33  
 BUILD\_DATETIME (in module *ics*), 74

## C

can1 (*ics.CyanSettings* attribute), 27  
 can1 (*ics.FireSettings* attribute), 30  
 can1 (*ics.RadGalaxySettings* attribute), 34

can1 (*ics.RadStar2Settings* attribute), 36  
can1 (*ics.Vcan3Settings* attribute), 41  
can1 (*ics.Vcan412Settings* attribute), 42  
can1 (*ics.Vcan4Settings* attribute), 42  
can1 (*ics.VcanRFSettings* attribute), 43  
can1 (*ics.VividCANSettings* attribute), 44  
can1\_options (*ics.TextApiSettings* attribute), 40  
can1\_rx\_id (*ics.TextApiSettings* attribute), 40  
can1\_tx\_id (*ics.TextApiSettings* attribute), 40  
can2 (*ics.CyanSettings* attribute), 27  
can2 (*ics.FireSettings* attribute), 30  
can2 (*ics.RadGalaxySettings* attribute), 34  
can2 (*ics.RadStar2Settings* attribute), 36  
can2 (*ics.Vcan3Settings* attribute), 41  
can2 (*ics.Vcan412Settings* attribute), 42  
can2 (*ics.Vcan4Settings* attribute), 42  
can2 (*ics.VcanRFSettings* attribute), 43  
can2\_options (*ics.TextApiSettings* attribute), 40  
can2\_rx\_id (*ics.TextApiSettings* attribute), 40  
can2\_tx\_id (*ics.TextApiSettings* attribute), 40  
can3 (*ics.CyanSettings* attribute), 27  
can3 (*ics.FireSettings* attribute), 30  
can3 (*ics.RadGalaxySettings* attribute), 34  
can3 (*ics.Vcan4Settings* attribute), 42  
can3 (*ics.VcanRFSettings* attribute), 43  
can3\_options (*ics.TextApiSettings* attribute), 40  
can3\_rx\_id (*ics.TextApiSettings* attribute), 40  
can3\_tx\_id (*ics.TextApiSettings* attribute), 40  
can4 (*ics.CyanSettings* attribute), 27  
can4 (*ics.FireSettings* attribute), 30  
can4 (*ics.RadGalaxySettings* attribute), 34  
can4 (*ics.Vcan4Settings* attribute), 42  
can4 (*ics.VcanRFSettings* attribute), 44  
can4\_options (*ics.TextApiSettings* attribute), 40  
can4\_rx\_id (*ics.TextApiSettings* attribute), 40  
can4\_tx\_id (*ics.TextApiSettings* attribute), 40  
can5 (*ics.CyanSettings* attribute), 27  
can5 (*ics.RadGalaxySettings* attribute), 34  
can6 (*ics.CyanSettings* attribute), 27  
can6 (*ics.RadGalaxySettings* attribute), 34  
can7 (*ics.CyanSettings* attribute), 27  
can7 (*ics.RadGalaxySettings* attribute), 34  
can8 (*ics.CyanSettings* attribute), 27  
can8 (*ics.RadGalaxySettings* attribute), 34  
CAN\_BPS10000 (*in module ics*), 74  
CAN\_BPS5000 (*in module ics*), 74  
CAN\_BPS6667 (*in module ics*), 74  
CAN\_BPS8000 (*in module ics*), 74  
CAN\_SETTINGS\_SIZE (*in module ics*), 74  
can\_switch\_mode (*ics.CyanSettings* attribute), 27  
can\_switch\_mode (*ics.RadGalaxySettings* attribute), 34  
can\_switch\_mode (*ics.RadStar2Settings* attribute), 36  
can\_switch\_mode (*ics.VividCANSettings* attribute), 45  
canfd1 (*ics.CyanSettings* attribute), 27  
canfd1 (*ics.RadGalaxySettings* attribute), 34  
canfd1 (*ics.RadStar2Settings* attribute), 36  
canfd1 (*ics.Vcan412Settings* attribute), 42  
canfd1 (*ics.Vcan4Settings* attribute), 42  
canfd2 (*ics.CyanSettings* attribute), 27  
canfd2 (*ics.RadGalaxySettings* attribute), 34  
canfd2 (*ics.RadStar2Settings* attribute), 36  
canfd2 (*ics.Vcan412Settings* attribute), 42  
canfd2 (*ics.Vcan4Settings* attribute), 43  
canfd3 (*ics.CyanSettings* attribute), 27  
canfd3 (*ics.RadGalaxySettings* attribute), 34  
canfd3 (*ics.Vcan4Settings* attribute), 43  
canfd4 (*ics.CyanSettings* attribute), 27  
canfd4 (*ics.RadGalaxySettings* attribute), 34  
canfd4 (*ics.Vcan4Settings* attribute), 43  
canfd5 (*ics.CyanSettings* attribute), 27  
canfd5 (*ics.RadGalaxySettings* attribute), 34  
canfd6 (*ics.CyanSettings* attribute), 27  
canfd6 (*ics.RadGalaxySettings* attribute), 34  
canfd7 (*ics.CyanSettings* attribute), 27  
canfd7 (*ics.RadGalaxySettings* attribute), 34  
canfd8 (*ics.CyanSettings* attribute), 27  
canfd8 (*ics.RadGalaxySettings* attribute), 34  
CANFD\_BRS\_ENABLED (*in module ics*), 74  
CANFD\_BRS\_ENABLED\_ISO (*in module ics*), 74  
CANFD\_ENABLED (*in module ics*), 74  
CANFD\_ENABLED\_ISO (*in module ics*), 74  
CANFD\_SETTINGS\_SIZE (*in module ics*), 74  
CanFdSettings (*class in ics*), 24  
CanSettings (*class in ics*), 24  
CANTERM\_SETTINGS\_SIZE (*in module ics*), 74  
cf\_timeout (*ics.CmISO157652RxMessage* attribute), 25  
cgi\_baud (*ics.FireSettings* attribute), 30  
cgi\_chksum\_enable (*ics.FireSettings* attribute), 30  
cgi\_enable\_reserved (*ics.FireSettings* attribute), 30  
cgi\_rx\_ifs\_bit\_times (*ics.FireSettings* attribute), 30  
cgi\_tx\_ifs\_bit\_times (*ics.FireSettings* attribute), 30  
chksum\_enabled (*ics.Iso9141Keyword2000Settings* attribute), 32  
close\_device() (*in module ics*), 52  
ClosePort() (*in module ics*), 45  
CmISO157652RxMessage (*class in ics*), 25  
CmISO157652TxMessage (*class in ics*), 26  
coremini\_clear() (*in module ics*), 52  
coremini\_get\_fblock\_status() (*in module ics*), 53  
coremini\_get\_status() (*in module ics*), 53

coremini\_load() (in module ics), 53  
 coremini\_read\_app\_signal() (in module ics), 53  
 coremini\_read\_rx\_message() (in module ics), 53  
 coremini\_read\_tx\_message() (in module ics), 54  
 coremini\_start() (in module ics), 54  
 coremini\_start\_fblock() (in module ics), 54  
 coremini\_stop() (in module ics), 54  
 coremini\_stop\_fblock() (in module ics), 55  
 coremini\_write\_app\_signal() (in module ics), 55  
 coremini\_write\_rx\_message() (in module ics), 55  
 coremini\_write\_tx\_message() (in module ics), 55  
 create\_neovi\_radio\_message() (in module ics), 55  
 cyan (ics.DeviceSettings attribute), 29  
 CyanSettings (class in ics), 26

## D

data (ics.CmISO157652TxMessage attribute), 26  
 Data (ics.SpyMessage attribute), 38  
 Data (ics.SpyMessageJ1850 attribute), 39  
 DescriptionID (ics.SpyMessage attribute), 38  
 DescriptionID (ics.SpyMessageJ1850 attribute), 39  
 DeviceSettings (class in ics), 29  
 DeviceSettingType (ics.DeviceSettings attribute), 29  
 DeviceType (ics.NeoDevice attribute), 33  
 digitalIoThresholdEnable (ics.CyanSettings attribute), 27  
 digitalIoThresholdTicks (ics.CyanSettings attribute), 27  
 DISABLE (in module ics), 74  
 disableUsbCheckOnBoot (ics.CyanSettings attribute), 27  
 disableUsbCheckOnBoot (ics.Vcan412Settings attribute), 42  
 disableUsbCheckOnBoot (ics.VividCANSettings attribute), 45  
 duplex (ics.EthernetSettings attribute), 30

## E

ecu\_id (ics.VividCANSettings attribute), 45  
 enable\_network\_com() (in module ics), 56  
 enableLatencyTest (ics.CyanSettings attribute), 27  
 enableLatencyTest (ics.Vcan412Settings attribute), 42  
 enableLatencyTest (ics.Vcan4Settings attribute), 43

enableLatencyTest (ics.VividCANSettings attribute), 45  
 EnableNetworkCom() (in module ics), 45  
 enablePcEthernetComm (ics.Vcan4Settings attribute), 43  
 ethernet (ics.CyanSettings attribute), 27  
 ethernet (ics.Vcan4Settings attribute), 43  
 ETHERNET\_SETTINGS\_SIZE (in module ics), 74  
 ethernetActivationLineEnabled (ics.Fire2DeviceStatus attribute), 30  
 ethernetActivationLineEnabled (ics.Vcan4DeviceStatus attribute), 42  
 EthernetSettings (class in ics), 30  
 extendedAddress (ics.CmISO157652RxMessage attribute), 25  
 extendedAddress (ics.CmISO157652TxMessage attribute), 26  
 ExtraDataPtr (ics.SpyMessage attribute), 38  
 ExtraDataPtr (ics.SpyMessageJ1850 attribute), 39  
 ExtraDataPtrEnabled (ics.SpyMessage attribute), 38  
 ExtraDataPtrEnabled (ics.SpyMessageJ1850 attribute), 39

## F

fast\_init\_network\_enables\_1 (ics.FireSettings attribute), 30  
 fast\_init\_network\_enables\_2 (ics.FireSettings attribute), 30  
 FAST\_MODE (in module ics), 74  
 fc\_id (ics.CmISO157652RxMessage attribute), 25  
 fc\_id (ics.CmISO157652TxMessage attribute), 26  
 fc\_id\_mask (ics.CmISO157652TxMessage attribute), 26  
 FDBaudrate (ics.CanFdSettings attribute), 24  
 FDBRP (ics.CanFdSettings attribute), 24  
 FDMODE (ics.CanFdSettings attribute), 24  
 FDTqProp (ics.CanFdSettings attribute), 24  
 FDTqSeg1 (ics.CanFdSettings attribute), 24  
 FDTqSeg2 (ics.CanFdSettings attribute), 24  
 FDTqSync (ics.CanFdSettings attribute), 24  
 find\_devices() (in module ics), 56  
 FindNeoDevices() (in module ics), 45  
 fire (ics.DeviceSettings attribute), 29  
 Fire2DeviceStatus (class in ics), 30  
 fire2Status (ics.IcsDeviceStatus attribute), 32  
 FireSettings (class in ics), 30  
 firmware\_update\_required() (in module ics), 56  
 FirmwareUpdateRequired() (in module ics), 45  
 flags (ics.CmISO157652RxMessage attribute), 25  
 flags (ics.CmISO157652TxMessage attribute), 26  
 flow\_control (ics.UartSettings attribute), 41

flowControlExtendedAddress  
(*ics.CmISO157652RxMessage* attribute),  
25

flowControlExtendedAddress  
(*ics.CmISO157652TxMessage* attribute),  
26

force\_firmware\_update() (in module *ics*), 57

ForceFirmwareUpdate() (in module *ics*), 45

fs\_timeout (*ics.CmISO157652TxMessage* attribute),  
26

fs\_wait (*ics.CmISO157652TxMessage* attribute), 26

## G

get\_active\_vnet\_channel() (in module *ics*), 57

get\_backup\_power\_enabled() (in module *ics*),  
57

get\_backup\_power\_ready() (in module *ics*), 57

get\_device\_settings() (in module *ics*), 57

get\_device\_status() (in module *ics*), 58

get\_dll\_firmware\_info() (in module *ics*), 58

get\_dll\_version() (in module *ics*), 58

get\_error\_messages() (in module *ics*), 58

get\_hw\_firmware\_info() (in module *ics*), 59

get\_last\_api\_error() (in module *ics*), 59

get\_library\_path() (in module *ics*), 59

get\_messages() (in module *ics*), 59

get\_performance\_parameters() (in module  
*ics*), 60

get\_rtc() (in module *ics*), 60

get\_script\_status() (in module *ics*), 60

get\_serial\_number() (in module *ics*), 60

get\_timestamp\_for\_msg() (in module *ics*), 60

GetActiveVNETChannel() (in module *ics*), 46

GetBackupPowerEnabled() (in module *ics*), 46

GetBackupPowerReady() (in module *ics*), 46

GetDeviceStatus() (in module *ics*), 46

GetDLLFirmwareInfo() (in module *ics*), 46

GetDLLVersion() (in module *ics*), 46

GetErrorMessages() (in module *ics*), 46

GetFireSettings() (in module *ics*), 46

GetHWFirmwareInfo() (in module *ics*), 47

GetLastAPIError() (in module *ics*), 47

GetMessages() (in module *ics*), 47

GetPerformanceParameters() (in module *ics*),  
47

GetRTC() (in module *ics*), 47

GetSerialNumber() (in module *ics*), 47

GetTimeStampForMsg() (in module *ics*), 47

GetVCAN3Settings() (in module *ics*), 47

GLOBAL\_SETTINGS\_SIZE (in module *ics*), 74

GS\_VERSION (in module *ics*), 74

## H

Handle (*ics.NeoDevice* attribute), 33

Header (*ics.SpyMessageJ1850* attribute), 39

high\_speed\_auto\_switch (*ics.SWCanSettings* at-  
tribute), 38

hwComLatencyTestEn (*ics.RadStar2Settings* at-  
tribute), 36

## I

iAppMajor (*ics.ApiFirmwareInfo* attribute), 24

iAppMinor (*ics.ApiFirmwareInfo* attribute), 24

iBoardRevMajor (*ics.ApiFirmwareInfo* attribute), 24

iBoardRevMinor (*ics.ApiFirmwareInfo* attribute), 24

iBootLoaderVersionMajor (*ics.ApiFirmwareInfo*  
attribute), 24

iBootLoaderVersionMinor (*ics.ApiFirmwareInfo*  
attribute), 24

ics (module), 23

IcsDeviceStatus (class in *ics*), 32

icsneoClosePort() (in module *ics*), 61

icsneoEnableNetworkCom() (in module *ics*), 61

icsneoFindNeoDevices() (in module *ics*), 61

icsneoFirmwareUpdateRequired() (in module  
*ics*), 61

icsneoForceFirmwareUpdate() (in module *ics*),  
61

icsneoGetActiveVNETChannel() (in module  
*ics*), 61

icsneoGetBackupPowerEnabled() (in module  
*ics*), 61

icsneoGetBackupPowerReady() (in module *ics*),  
61

icsneoGetDeviceStatus() (in module *ics*), 62

icsneoGetDLLFirmwareInfo() (in module *ics*),  
62

icsneoGetDLLVersion() (in module *ics*), 62

icsneoGetErrorMessages() (in module *ics*), 62

icsneoGetFireSettings() (in module *ics*), 62

icsneoGetHWFirmwareInfo() (in module *ics*), 62

icsneoGetLastAPIError() (in module *ics*), 62

icsneoGetMessages() (in module *ics*), 62

icsneoGetPerformanceParameters() (in mod-  
ule *ics*), 62

icsneoGetRTC() (in module *ics*), 63

icsneoGetSerialNumber() (in module *ics*), 63

icsneoGetTimeStampForMsg() (in module *ics*),  
63

icsneoGetVCAN3Settings() (in module *ics*), 63

icsneoISO15765\_DisableNetworks() (in mod-  
ule *ics*), 63

icsneoISO15765\_EnableNetworks() (in mod-  
ule *ics*), 63

icsneoISO15765\_ReceiveMessage() (in mod-  
ule *ics*), 63

icsneoISO15765\_TransmitMessage() (in mod-  
ule *ics*), 63



icsneoLoadDefaultSettings() (in module ics), 64  
 icsneoOpenNeoDevice() (in module ics), 64  
 icsneoReadSDCard() (in module ics), 64  
 icsneoRequestEnterSleepMode() (in module ics), 64  
 icsneoScriptClear() (in module ics), 64  
 icsneoScriptGetFBlockStatus() (in module ics), 64  
 icsneoScriptGetScriptStatus() (in module ics), 64  
 icsneoScriptGetScriptStatusEx() (in module ics), 64  
 icsneoScriptLoad() (in module ics), 64  
 icsneoScriptReadAppSignal() (in module ics), 65  
 icsneoScriptReadRxMessage() (in module ics), 65  
 icsneoScriptReadTxMessage() (in module ics), 65  
 icsneoScriptStart() (in module ics), 65  
 icsneoScriptStartFBlock() (in module ics), 65  
 icsneoScriptStop() (in module ics), 65  
 icsneoScriptStopFBlock() (in module ics), 65  
 icsneoScriptWriteAppSignal() (in module ics), 65  
 icsneoScriptWriteRxMessage() (in module ics), 66  
 icsneoScriptWriteTxMessage() (in module ics), 66  
 icsneoSetActiveVNETChannel() (in module ics), 66  
 icsneoSetBackupPowerEnabled() (in module ics), 66  
 icsneoSetBitRate() (in module ics), 66  
 icsneoSetBitRateEx() (in module ics), 66  
 icsneoSetContext() (in module ics), 66  
 icsneoSetFDBitRate() (in module ics), 66  
 icsneoSetFireSettings() (in module ics), 67  
 icsneoSetReflashDisplayCallbacks() (in module ics), 67  
 icsneoSetRTC() (in module ics), 67  
 icsneoSetVCAN3Settings() (in module ics), 67  
 icsneoTxMessages() (in module ics), 67  
 icsneoValidateHObject() (in module ics), 67  
 icsneoWriteSDCard() (in module ics), 67  
 id (ics.CmlISO157652RxMessage attribute), 25  
 id (ics.CmlISO157652TxMessage attribute), 26  
 id\_mask (ics.CmlISO157652RxMessage attribute), 25  
 idle\_wakeup\_network\_enables\_1 (ics.RadGalaxySettings attribute), 35  
 idle\_wakeup\_network\_enables\_1 (ics.RadStar2Settings attribute), 36  
 idle\_wakeup\_network\_enables\_1 (ics.VcanRFSettings attribute), 44  
 idle\_wakeup\_network\_enables\_2 (ics.RadGalaxySettings attribute), 35  
 idle\_wakeup\_network\_enables\_2 (ics.RadStar2Settings attribute), 36  
 idle\_wakeup\_network\_enables\_2 (ics.VcanRFSettings attribute), 44  
 idle\_wakeup\_network\_enables\_3 (ics.CyanSettings attribute), 28  
 idle\_wakeup\_network\_enables\_3 (ics.RadGalaxySettings attribute), 35  
 idle\_wakeup\_network\_enables\_3 (ics.RadStar2Settings attribute), 36  
 iMainFirmChkSum (ics.ApiFirmwareInfo attribute), 24  
 iMainFirmDateDay (ics.ApiFirmwareInfo attribute), 24  
 iMainFirmDateHour (ics.ApiFirmwareInfo attribute), 24  
 iMainFirmDateMin (ics.ApiFirmwareInfo attribute), 24  
 iMainFirmDateMonth (ics.ApiFirmwareInfo attribute), 24  
 iMainFirmDateSecond (ics.ApiFirmwareInfo attribute), 24  
 iMainFirmDateYear (ics.ApiFirmwareInfo attribute), 24  
 iMainVnetHWrevMajor (ics.ApiFirmwareInfo attribute), 24  
 iMainVnetHWrevMinor (ics.ApiFirmwareInfo attribute), 24  
 iMainVnetSRAMSize (ics.ApiFirmwareInfo attribute), 24  
 iManufactureDay (ics.ApiFirmwareInfo attribute), 24  
 iManufactureMonth (ics.ApiFirmwareInfo attribute), 24  
 iManufactureYear (ics.ApiFirmwareInfo attribute), 24  
 init\_steps (ics.Iso9141Keyword2000Settings attribute), 32  
 innerFrameDelay25us (ics.CanSettings attribute), 25  
 ISO15765\_2\_NETWORK\_HSCAN (in module ics), 74  
 ISO15765\_2\_NETWORK\_HSCAN2 (in module ics), 74  
 ISO15765\_2\_NETWORK\_HSCAN3 (in module ics), 74  
 ISO15765\_2\_NETWORK\_HSCAN4 (in module ics), 74  
 ISO15765\_2\_NETWORK\_HSCAN5 (in module ics), 74  
 ISO15765\_2\_NETWORK\_HSCAN6 (in module ics), 74  
 ISO15765\_2\_NETWORK\_HSCAN7 (in module ics), 74  
 ISO15765\_2\_NETWORK\_MSCAN (in module ics), 74  
 ISO15765\_2\_NETWORK\_SWCAN (in module ics), 74  
 ISO15765\_2\_NETWORK\_SWCAN2 (in module ics), 74  
 iso15765\_disable\_networks() (in module ics),

67  
ISO15765\_DisableNetworks() (in module ics),  
47  
iso15765\_enable\_networks() (in module ics),  
67  
ISO15765\_EnableNetworks() (in module ics), 48  
iso15765\_receive\_message() (in module ics),  
68  
ISO15765\_ReceiveMessage() (in module ics), 48  
iso15765\_separation\_time\_offset  
(ics.CyanSettings attribute), 28  
iso15765\_separation\_time\_offset  
(ics.FireSettings attribute), 30  
iso15765\_separation\_time\_offset  
(ics.RadGalaxySettings attribute), 35  
iso15765\_separation\_time\_offset  
(ics.RadStar2Settings attribute), 36  
iso15765\_separation\_time\_offset  
(ics.Vcan3Settings attribute), 41  
iso15765\_separation\_time\_offset  
(ics.Vcan4I2Settings attribute), 42  
iso15765\_separation\_time\_offset  
(ics.Vcan4Settings attribute), 43  
iso15765\_separation\_time\_offset  
(ics.VcanRFSettings attribute), 44  
iso15765\_separation\_time\_offset  
(ics.VividCANSettings attribute), 45  
iso15765\_transmit\_message() (in module ics),  
68  
ISO15765\_TransmitMessage() (in module ics),  
48  
ISO9141\_KEYWORD2000\_\_INIT\_STEP\_SIZE (in  
module ics), 74  
ISO9141\_KEYWORD2000\_SETTINGS\_SIZE (in  
module ics), 74  
iso9141\_kwp\_enable\_reserved (ics.FireSettings  
attribute), 31  
iso9141\_kwp\_enable\_reserved  
(ics.VcanRFSettings attribute), 44  
iso9141\_kwp\_settings (ics.FireSettings attribute),  
31  
iso9141\_kwp\_settings (ics.VcanRFSettings at-  
tribute), 44  
iso9141\_kwp\_settings\_1 (ics.CyanSettings at-  
tribute), 28  
iso9141\_kwp\_settings\_1 (ics.RadGalaxySettings  
attribute), 35  
iso9141\_kwp\_settings\_1 (ics.RadStar2Settings  
attribute), 36  
iso9141\_kwp\_settings\_1 (ics.Vcan4Settings at-  
tribute), 43  
iso9141\_kwp\_settings\_2 (ics.CyanSettings at-  
tribute), 28  
iso9141\_kwp\_settings\_2 (ics.FireSettings at-  
tribute), 31  
iso9141\_kwp\_settings\_2 (ics.VcanRFSettings at-  
tribute), 44  
iso9141\_kwp\_settings\_3 (ics.CyanSettings at-  
tribute), 28  
iso9141\_kwp\_settings\_3 (ics.FireSettings at-  
tribute), 31  
iso9141\_kwp\_settings\_4 (ics.CyanSettings at-  
tribute), 28  
iso9141\_kwp\_settings\_4 (ics.FireSettings at-  
tribute), 31  
Iso9141Keyword2000InitSteps (class in ics), 32  
Iso9141Keyword2000Settings (class in ics), 32  
iso\_9141\_kwp\_enable\_reserved  
(ics.RadStar2Settings attribute), 36  
iso\_9141\_kwp\_enable\_reserved  
(ics.Vcan4Settings attribute), 43  
iso\_msg\_termination (ics.FireSettings attribute),  
31  
iso\_msg\_termination (ics.VcanRFSettings at-  
tribute), 44  
iso\_msg\_termination\_1 (ics.CyanSettings at-  
tribute), 28  
iso\_msg\_termination\_1 (ics.RadGalaxySettings  
attribute), 35  
iso\_msg\_termination\_1 (ics.RadStar2Settings at-  
tribute), 36  
iso\_msg\_termination\_1 (ics.Vcan4Settings  
attribute), 43  
iso\_msg\_termination\_2 (ics.CyanSettings at-  
tribute), 28  
iso\_msg\_termination\_2 (ics.FireSettings at-  
tribute), 31  
iso\_msg\_termination\_2 (ics.VcanRFSettings at-  
tribute), 44  
iso\_msg\_termination\_3 (ics.CyanSettings at-  
tribute), 28  
iso\_msg\_termination\_3 (ics.FireSettings at-  
tribute), 31  
iso\_msg\_termination\_4 (ics.CyanSettings at-  
tribute), 28  
iso\_msg\_termination\_4 (ics.FireSettings at-  
tribute), 31  
iso\_parity (ics.FireSettings attribute), 31  
iso\_parity (ics.VcanRFSettings attribute), 44  
iso\_parity\_1 (ics.CyanSettings attribute), 28  
iso\_parity\_1 (ics.RadGalaxySettings attribute), 35  
iso\_parity\_1 (ics.RadStar2Settings attribute), 36  
iso\_parity\_1 (ics.Vcan4Settings attribute), 43  
iso\_parity\_2 (ics.CyanSettings attribute), 28  
iso\_parity\_2 (ics.FireSettings attribute), 31  
iso\_parity\_2 (ics.VcanRFSettings attribute), 44  
iso\_parity\_3 (ics.CyanSettings attribute), 28  
iso\_parity\_3 (ics.FireSettings attribute), 31



iso\_parity\_4 (*ics.CyanSettings* attribute), 28  
 iso\_parity\_4 (*ics.FireSettings* attribute), 31  
 iso\_tester\_pullup\_enable (*ics.FireSettings* attribute), 31  
 iso\_tester\_pullup\_enable (*ics.VcanRFSettings* attribute), 44  
 IsOpen (*ics.NeoDevice* attribute), 33  
 iType (*ics.ApiFirmwareInfo* attribute), 24

## J

J1708\_SETTINGS\_SIZE (in module *ics*), 74

## K

k (*ics.Iso9141Keyword2000InitSteps* attribute), 32

## L

l (*ics.Iso9141Keyword2000InitSteps* attribute), 32  
 led\_mode (*ics.EthernetSettings* attribute), 30  
 lin1 (*ics.CyanSettings* attribute), 28  
 lin1 (*ics.FireSettings* attribute), 31  
 lin1 (*ics.RadGalaxySettings* attribute), 35  
 lin1 (*ics.RadStar2Settings* attribute), 37  
 lin1 (*ics.Vcan4Settings* attribute), 43  
 lin1 (*ics.VcanRFSettings* attribute), 44  
 lin2 (*ics.CyanSettings* attribute), 28  
 lin2 (*ics.FireSettings* attribute), 31  
 lin2 (*ics.VcanRFSettings* attribute), 44  
 lin3 (*ics.CyanSettings* attribute), 28  
 lin3 (*ics.FireSettings* attribute), 31  
 lin4 (*ics.CyanSettings* attribute), 28  
 lin4 (*ics.FireSettings* attribute), 31  
 lin5 (*ics.CyanSettings* attribute), 28  
 lin6 (*ics.CyanSettings* attribute), 28  
 LIN\_SETTINGS\_SIZE (in module *ics*), 74  
 link\_speed (*ics.EthernetSettings* attribute), 30  
 LinSettings (class in *ics*), 33  
 LISTEN\_ALL (in module *ics*), 74  
 LISTEN\_ONLY (in module *ics*), 74  
 load\_default\_settings() (in module *ics*), 68  
 LoadDefaultSettings() (in module *ics*), 48  
 LOOPBACK (in module *ics*), 74  
 lsft (*ics.FireSettings* attribute), 31  
 lsft1 (*ics.CyanSettings* attribute), 28  
 lsft2 (*ics.CyanSettings* attribute), 28  
 lsftcan1 (*ics.VividCANSettings* attribute), 45

## M

MasterEnable (*ics.TimesyncSettings* attribute), 41  
 MasterNetwork (*ics.TimesyncSettings* attribute), 41  
 MasterResistor (*ics.LinSettings* attribute), 33  
 MaxAllowedClients (*ics.NeoDevice* attribute), 33  
 MessagePieceID (*ics.SpyMessage* attribute), 38  
 MessagePieceID (*ics.SpyMessageJ1850* attribute), 39

misc\_io\_analog\_enable (*ics.CyanSettings* attribute), 29  
 misc\_io\_analog\_enable (*ics.FireSettings* attribute), 31  
 misc\_io\_analog\_enable (*ics.RadGalaxySettings* attribute), 35  
 misc\_io\_analog\_enable (*ics.RadStar2Settings* attribute), 37  
 misc\_io\_analog\_enable (*ics.VcanRFSettings* attribute), 44  
 misc\_io\_initial\_ddr (*ics.CyanSettings* attribute), 29  
 misc\_io\_initial\_ddr (*ics.FireSettings* attribute), 31  
 misc\_io\_initial\_ddr (*ics.RadGalaxySettings* attribute), 35  
 misc\_io\_initial\_ddr (*ics.RadStar2Settings* attribute), 37  
 misc\_io\_initial\_ddr (*ics.Vcan3Settings* attribute), 41  
 misc\_io\_initial\_ddr (*ics.VcanRFSettings* attribute), 44  
 misc\_io\_initial\_latch (*ics.CyanSettings* attribute), 29  
 misc\_io\_initial\_latch (*ics.FireSettings* attribute), 31  
 misc\_io\_initial\_latch (*ics.RadGalaxySettings* attribute), 35  
 misc\_io\_initial\_latch (*ics.RadStar2Settings* attribute), 37  
 misc\_io\_initial\_latch (*ics.Vcan3Settings* attribute), 41  
 misc\_io\_initial\_latch (*ics.VcanRFSettings* attribute), 44  
 misc\_io\_on\_report\_events (*ics.CyanSettings* attribute), 29  
 misc\_io\_on\_report\_events (*ics.FireSettings* attribute), 31  
 misc\_io\_on\_report\_events (*ics.RadGalaxySettings* attribute), 35  
 misc\_io\_on\_report\_events (*ics.RadStar2Settings* attribute), 37  
 misc\_io\_on\_report\_events (*ics.Vcan3Settings* attribute), 41  
 misc\_io\_on\_report\_events (*ics.VcanRFSettings* attribute), 44  
 misc\_io\_report\_period (*ics.CyanSettings* attribute), 29  
 misc\_io\_report\_period (*ics.FireSettings* attribute), 31  
 misc\_io\_report\_period (*ics.RadGalaxySettings* attribute), 35  
 misc\_io\_report\_period (*ics.RadStar2Settings* attribute), 37

`misc_io_report_period` (*ics.Vcan3Settings attribute*), 41  
`misc_io_report_period` (*ics.VcanRFSettings attribute*), 44  
`MiscData` (*ics.SpyMessage attribute*), 38  
`MiscData` (*ics.SpyMessageJ1850 attribute*), 39  
`Mode` (*ics.CanSettings attribute*), 25  
`Mode` (*ics.LinSettings attribute*), 33  
`Mode` (*ics.SWCanSettings attribute*), 37

## N

`Name` (*ics.NeoDevice attribute*), 33  
`NEO_CFG_MPIC_HS_CAN_CNF1` (*in module ics*), 76  
`NEO_CFG_MPIC_HS_CAN_CNF2` (*in module ics*), 76  
`NEO_CFG_MPIC_HS_CAN_CNF3` (*in module ics*), 76  
`NEO_CFG_MPIC_HS_CAN_MODE` (*in module ics*), 76  
`NEO_CFG_MPIC_LSFT_CAN_CNF1` (*in module ics*), 76  
`NEO_CFG_MPIC_LSFT_CAN_CNF2` (*in module ics*), 76  
`NEO_CFG_MPIC_LSFT_CAN_CNF3` (*in module ics*), 76  
`NEO_CFG_MPIC_MS_CAN_CNF1` (*in module ics*), 76  
`NEO_CFG_MPIC_MS_CAN_CNF2` (*in module ics*), 76  
`NEO_CFG_MPIC_MS_CAN_CNF3` (*in module ics*), 76  
`NEO_CFG_MPIC_SW_CAN_CNF1` (*in module ics*), 76  
`NEO_CFG_MPIC_SW_CAN_CNF2` (*in module ics*), 76  
`NEO_CFG_MPIC_SW_CAN_CNF3` (*in module ics*), 76  
`NeoDevice` (*class in ics*), 33  
`NEODEVICE_ALL` (*in module ics*), 74  
`NEODEVICE_ANY_ION` (*in module ics*), 75  
`NEODEVICE_ANY_PLASMA` (*in module ics*), 75  
`NEODEVICE_BLUE` (*in module ics*), 75  
`NEODEVICE_CMPROBE` (*in module ics*), 75  
`NEODEVICE_CT_OBD` (*in module ics*), 75  
`NEODEVICE_DW_VCAN` (*in module ics*), 75  
`NEODEVICE_ECU` (*in module ics*), 75  
`NEODEVICE_ECUCHIP_UART` (*in module ics*), 75  
`NEODEVICE_EEVB` (*in module ics*), 75  
`NEODEVICE_FIRE` (*in module ics*), 75  
`NEODEVICE_FIRE2` (*in module ics*), 75  
`NEODEVICE_FIRE_VNET` (*in module ics*), 75  
`NEODEVICE_FLEX` (*in module ics*), 75  
`NEODEVICE_IEVB` (*in module ics*), 75  
`NEODEVICE_ION_2` (*in module ics*), 75  
`NEODEVICE_ION_3` (*in module ics*), 75  
`NEODEVICE_NEOANALOG` (*in module ics*), 75  
`NEODEVICE_NEOECUCHIP` (*in module ics*), 75  
`NEODEVICE_OBD2_PRO` (*in module ics*), 75  
`NEODEVICE_OBD2_SIM` (*in module ics*), 75  
`NEODEVICE_PENDANT` (*in module ics*), 75  
`NEODEVICE_PLASMA_1_11` (*in module ics*), 75  
`NEODEVICE_PLASMA_1_12` (*in module ics*), 75  
`NEODEVICE_PLASMA_1_13` (*in module ics*), 75

`NEODEVICE_RADGALAXY` (*in module ics*), 75  
`NEODEVICE_RADSTAR` (*in module ics*), 75  
`NEODEVICE_RADSTAR2` (*in module ics*), 75  
`NEODEVICE_RED` (*in module ics*), 75  
`NEODEVICE_SW_VCAN` (*in module ics*), 75  
`NEODEVICE_UNKNOWN` (*in module ics*), 75  
`NEODEVICE_VCAN3` (*in module ics*), 75  
`NEODEVICE_VCAN4` (*in module ics*), 75  
`NEODEVICE_VCAN4_12` (*in module ics*), 75  
`NEODEVICE_VCANRF` (*in module ics*), 75  
`NEODEVICE_VIVIDCAN` (*in module ics*), 75  
`NEOVI6_VCAN_TIMESTAMP_1` (*in module ics*), 75  
`NEOVI6_VCAN_TIMESTAMP_2` (*in module ics*), 76  
`NEOVI_3G_MAX_SETTINGS_SIZE` (*in module ics*), 76  
`NEOVI_COMMTYPE_FIRE_USB` (*in module ics*), 76  
`NEOVI_COMMTYPE_RS232` (*in module ics*), 76  
`NEOVI_COMMTYPE_TCPIP` (*in module ics*), 76  
`NEOVI_COMMTYPE_USB_BULK` (*in module ics*), 76  
`NEOVI_RED_TIMESTAMP_1_10NS` (*in module ics*), 76  
`NEOVI_RED_TIMESTAMP_1_25NS` (*in module ics*), 76  
`NEOVI_RED_TIMESTAMP_2_10NS` (*in module ics*), 76  
`NEOVI_RED_TIMESTAMP_2_25NS` (*in module ics*), 76  
`NEOVI_TIMESTAMP_1` (*in module ics*), 76  
`NEOVI_TIMESTAMP_2` (*in module ics*), 76  
`NEOVI_PRO_VCAN_TIMESTAMP_1` (*in module ics*), 76  
`NEOVI_PRO_VCAN_TIMESTAMP_2` (*in module ics*), 76  
`NETID_3G_APP_SIGNAL_STATUS` (*in module ics*), 76  
`NETID_3G_FB_STATUS` (*in module ics*), 76  
`NETID_3G_LOGGING_OVERFLOW` (*in module ics*), 76  
`NETID_3G_READ_DATALINK_CM_RX_MSG` (*in module ics*), 76  
`NETID_3G_READ_DATALINK_CM_TX_MSG` (*in module ics*), 76  
`NETID_3G_READ_SETTINGS_EX` (*in module ics*), 76  
`NETID_3G_RESET_STATUS` (*in module ics*), 76  
`NETID_AUX` (*in module ics*), 76  
`NETID_CGI` (*in module ics*), 76  
`NETID_DATA_TO_HOST` (*in module ics*), 77  
`NETID_DEVICE` (*in module ics*), 77  
`NETID_DEVICE_STATUS` (*in module ics*), 77  
`NETID_ETHERNET` (*in module ics*), 77  
`NETID_ETHERNET_DAQ` (*in module ics*), 77  
`NETID_FLEXRAY` (*in module ics*), 77  
`NETID_FLEXRAY1A` (*in module ics*), 77  
`NETID_FLEXRAY1B` (*in module ics*), 77  
`NETID_FLEXRAY2` (*in module ics*), 77  
`NETID_FLEXRAY2A` (*in module ics*), 77  
`NETID_FLEXRAY2B` (*in module ics*), 77

- NETID\_FORDSCP (in module ics), 77
- NETID\_GMFSA (in module ics), 77
- NETID\_HSCAN (in module ics), 77
- NETID\_HSCAN2 (in module ics), 77
- NETID\_HSCAN3 (in module ics), 77
- NETID\_HSCAN4 (in module ics), 77
- NETID\_HSCAN5 (in module ics), 77
- NETID\_HSCAN6 (in module ics), 77
- NETID\_HSCAN7 (in module ics), 77
- NETID\_HW\_COM\_LATENCY\_TEST (in module ics), 77
- NETID\_I2C1 (in module ics), 77
- NETID\_INVALID (in module ics), 77
- NETID\_ISO (in module ics), 77
- NETID\_ISO14230 (in module ics), 77
- NETID\_ISO2 (in module ics), 77
- NETID\_ISO3 (in module ics), 77
- NETID\_ISO4 (in module ics), 77
- NETID\_ISOPIC (in module ics), 77
- NETID\_J1708 (in module ics), 77
- NETID\_JVPW (in module ics), 77
- NETID\_LIN (in module ics), 77
- NETID\_LIN2 (in module ics), 77
- NETID\_LIN3 (in module ics), 77
- NETID\_LIN4 (in module ics), 77
- NETID\_LIN5 (in module ics), 77
- NETID\_LIN6 (in module ics), 78
- NETID\_LSFTCAN (in module ics), 78
- NETID\_LSFTCAN2 (in module ics), 78
- NETID\_MAIN51 (in module ics), 78
- NETID\_MAX (in module ics), 78
- NETID\_MOST (in module ics), 78
- NETID\_MOST150 (in module ics), 78
- NETID\_MOST25 (in module ics), 78
- NETID\_MOST50 (in module ics), 78
- NETID\_MSCAN (in module ics), 78
- NETID\_OP\_ETHERNET1 (in module ics), 78
- NETID\_OP\_ETHERNET10 (in module ics), 78
- NETID\_OP\_ETHERNET11 (in module ics), 78
- NETID\_OP\_ETHERNET12 (in module ics), 78
- NETID\_OP\_ETHERNET2 (in module ics), 78
- NETID\_OP\_ETHERNET3 (in module ics), 78
- NETID\_OP\_ETHERNET4 (in module ics), 78
- NETID\_OP\_ETHERNET5 (in module ics), 78
- NETID\_OP\_ETHERNET6 (in module ics), 78
- NETID\_OP\_ETHERNET7 (in module ics), 78
- NETID\_OP\_ETHERNET8 (in module ics), 78
- NETID\_OP\_ETHERNET9 (in module ics), 78
- NETID\_RED (in module ics), 78
- NETID\_RED\_APP\_ERROR (in module ics), 78
- NETID\_RED\_VBAT (in module ics), 78
- NETID\_RS232 (in module ics), 78
- NETID\_SCI (in module ics), 78
- NETID\_SPI1 (in module ics), 78
- NETID\_SWCAN (in module ics), 78
- NETID\_SWCAN2 (in module ics), 78
- NETID\_TCP (in module ics), 78
- NETID\_TEXTAPI\_TO\_HOST (in module ics), 78
- NETID\_UART (in module ics), 78
- NETID\_UART2 (in module ics), 78
- NETID\_UART3 (in module ics), 78
- NETID\_UART4 (in module ics), 78
- network\_enabled\_on\_boot (ics.CyanSettings attribute), 29
- network\_enabled\_on\_boot (ics.FireSettings attribute), 31
- network\_enabled\_on\_boot (ics.RadGalaxySettings attribute), 35
- network\_enabled\_on\_boot (ics.RadStar2Settings attribute), 37
- network\_enabled\_on\_boot (ics.Vcan3Settings attribute), 41
- network\_enabled\_on\_boot (ics.Vcan412Settings attribute), 42
- network\_enabled\_on\_boot (ics.Vcan4Settings attribute), 43
- network\_enabled\_on\_boot (ics.VcanRFSettings attribute), 44
- network\_enabled\_on\_boot (ics.VividCANSettings attribute), 45
- network\_enables (ics.CyanSettings attribute), 29
- network\_enables (ics.FireSettings attribute), 32
- network\_enables (ics.RadGalaxySettings attribute), 35
- network\_enables (ics.RadStar2Settings attribute), 37
- network\_enables (ics.TextApiSettings attribute), 40
- network\_enables (ics.Vcan3Settings attribute), 41
- network\_enables (ics.Vcan412Settings attribute), 42
- network\_enables (ics.Vcan4Settings attribute), 43
- network\_enables (ics.VcanRFSettings attribute), 44
- network\_enables (ics.VividCANSettings attribute), 45
- network\_enables\_2 (ics.CyanSettings attribute), 29
- network\_enables\_2 (ics.FireSettings attribute), 32
- network\_enables\_2 (ics.RadGalaxySettings attribute), 35
- network\_enables\_2 (ics.RadStar2Settings attribute), 37
- network\_enables\_2 (ics.Vcan4Settings attribute), 43
- network\_enables\_2 (ics.VcanRFSettings attribute), 44
- network\_enables\_3 (ics.CyanSettings attribute), 29
- network\_enables\_3 (ics.RadGalaxySettings attribute), 35
- network\_enables\_3 (ics.RadStar2Settings attribute), 37

network\_enables\_3 (*ics.Vcan4Settings* attribute), 43

NetworkID (*ics.SpyMessage* attribute), 38

NetworkID (*ics.SpyMessageJ1850* attribute), 39

NetworkID2 (*ics.SpyMessage* attribute), 38

NetworkID2 (*ics.SpyMessageJ1850* attribute), 39

NO\_CANFD (in module *ics*), 79

NodeID (*ics.SpyMessage* attribute), 38

NodeID (*ics.SpyMessageJ1850* attribute), 39

noExtraDataPtrCleanup (*ics.SpyMessage* attribute), 39

noExtraDataPtrCleanup (*ics.SpyMessageJ1850* attribute), 40

NORMAL (in module *ics*), 79

NORMAL\_MODE (in module *ics*), 79

num\_bytes (*ics.CmISO157652TxMessage* attribute), 26

NumberBytesData (*ics.SpyMessage* attribute), 38

NumberBytesData (*ics.SpyMessageJ1850* attribute), 39

NumberBytesHeader (*ics.SpyMessage* attribute), 38

NumberBytesHeader (*ics.SpyMessageJ1850* attribute), 39

NumberOfClients (*ics.NeoDevice* attribute), 33

## O

OP\_ETH\_GENERAL\_SETTINGS\_SIZE (in module *ics*), 79

OP\_ETH\_SETTINGS\_SIZE (in module *ics*), 79

open\_device() (in module *ics*), 68

OpenNeoDevice() (in module *ics*), 48

opEth1 (*ics.RadGalaxySettings* attribute), 35

opEth1 (*ics.RadStar2Settings* attribute), 37

opEth10 (*ics.RadGalaxySettings* attribute), 35

opEth11 (*ics.RadGalaxySettings* attribute), 35

opEth12 (*ics.RadGalaxySettings* attribute), 35

opEth2 (*ics.RadGalaxySettings* attribute), 35

opEth2 (*ics.RadStar2Settings* attribute), 37

opEth3 (*ics.RadGalaxySettings* attribute), 35

opEth4 (*ics.RadGalaxySettings* attribute), 35

opEth5 (*ics.RadGalaxySettings* attribute), 35

opEth6 (*ics.RadGalaxySettings* attribute), 35

opEth7 (*ics.RadGalaxySettings* attribute), 35

opEth8 (*ics.RadGalaxySettings* attribute), 36

opEth9 (*ics.RadGalaxySettings* attribute), 36

OPETH\_FUNC\_MEDIACONVERTER (in module *ics*), 79

OPETH\_FUNC\_TAP (in module *ics*), 79

OPETH\_FUNC\_TAP\_LOW\_LATENCY (in module *ics*), 79

OPETH\_LINK\_AUTO (in module *ics*), 79

OPETH\_LINK\_MASTER (in module *ics*), 79

OPETH\_LINK\_SLAVE (in module *ics*), 79

OPETH\_MAC\_SPOOF\_DST\_ADDR (in module *ics*), 79

OPETH\_MAC\_SPOOF\_SRC\_ADDR (in module *ics*), 79

opEthGen (*ics.RadGalaxySettings* attribute), 36

opEthGen (*ics.RadStar2Settings* attribute), 37

OpEthGeneralSettings (class in *ics*), 33

OpEthSettings (class in *ics*), 33

override\_library\_name() (in module *ics*), 69

## P

p2\_500us (*ics.Iso9141Keyword2000Settings* attribute), 32

p3\_500us (*ics.Iso9141Keyword2000Settings* attribute), 32

p4\_500us (*ics.Iso9141Keyword2000Settings* attribute), 32

padding (*ics.CmISO157652RxMessage* attribute), 26

padding (*ics.CmISO157652TxMessage* attribute), 26

parity (*ics.UartSettings* attribute), 41

pc\_com\_mode (*ics.RadStar2Settings* attribute), 37

perf\_en (*ics.CyanSettings* attribute), 29

perf\_en (*ics.FireSettings* attribute), 32

perf\_en (*ics.RadGalaxySettings* attribute), 36

perf\_en (*ics.RadStar2Settings* attribute), 37

perf\_en (*ics.Vcan3Settings* attribute), 41

perf\_en (*ics.Vcan412Settings* attribute), 42

perf\_en (*ics.Vcan4Settings* attribute), 43

perf\_en (*ics.VcanRFSettings* attribute), 44

perf\_en (*ics.VividCANSettings* attribute), 45

PLASMA\_SLAVE1\_OFFSET (in module *ics*), 79

PLASMA\_SLAVE1\_OFFSET\_RANGE2 (in module *ics*), 79

PLASMA\_SLAVE2\_OFFSET (in module *ics*), 79

PLASMA\_SLAVE2\_OFFSET\_RANGE2 (in module *ics*), 79

PLASMA\_SLAVE3\_OFFSET\_RANGE2 (in module *ics*), 79

PLASMA\_SLAVE\_NUM (in module *ics*), 79

preemption\_en (*ics.OpEthSettings* attribute), 34

Protocol (*ics.SpyMessage* attribute), 38

Protocol (*ics.SpyMessageJ1850* attribute), 39

pwm\_man\_timeout (*ics.FireSettings* attribute), 32

pwr\_man\_enable (*ics.CyanSettings* attribute), 29

pwr\_man\_enable (*ics.FireSettings* attribute), 32

pwr\_man\_enable (*ics.RadGalaxySettings* attribute), 36

pwr\_man\_enable (*ics.RadStar2Settings* attribute), 37

pwr\_man\_enable (*ics.Vcan412Settings* attribute), 42

pwr\_man\_enable (*ics.Vcan4Settings* attribute), 43

pwr\_man\_enable (*ics.VcanRFSettings* attribute), 44

pwr\_man\_enable (*ics.VividCANSettings* attribute), 45

pwr\_man\_timeout (*ics.CyanSettings* attribute), 29

pwr\_man\_timeout (*ics.RadGalaxySettings* attribute), 36

pwr\_man\_timeout (*ics.RadStar2Settings* attribute), 37



pwr\_man\_timeout (*ics.Vcan412Settings* attribute), 42  
 pwr\_man\_timeout (*ics.Vcan4Settings* attribute), 43  
 pwr\_man\_timeout (*ics.VcanRFSettings* attribute), 44  
 pwr\_man\_timeout (*ics.VividCANSettings* attribute), 45

## R

radgalaxy (*ics.DeviceSettings* attribute), 29  
 RadGalaxySettings (class in *ics*), 34  
 radstar2 (*ics.DeviceSettings* attribute), 29  
 RadStar2Settings (class in *ics*), 36  
 read\_sdcard() (in module *ics*), 69  
 ReadSDCard() (in module *ics*), 48  
 REPORT\_ON\_GPS (in module *ics*), 79  
 REPORT\_ON\_KLINE (in module *ics*), 79  
 REPORT\_ON\_LED1 (in module *ics*), 79  
 REPORT\_ON\_LED2 (in module *ics*), 79  
 REPORT\_ON\_MISC1 (in module *ics*), 79  
 REPORT\_ON\_MISC2 (in module *ics*), 79  
 REPORT\_ON\_MISC3 (in module *ics*), 79  
 REPORT\_ON\_MISC3\_AIN (in module *ics*), 79  
 REPORT\_ON\_MISC4 (in module *ics*), 79  
 REPORT\_ON\_MISC4\_AIN (in module *ics*), 79  
 REPORT\_ON\_MISC5 (in module *ics*), 79  
 REPORT\_ON\_MISC5\_AIN (in module *ics*), 79  
 REPORT\_ON\_MISC6 (in module *ics*), 79  
 REPORT\_ON\_MISC6\_AIN (in module *ics*), 79  
 REPORT\_ON\_PERIODIC (in module *ics*), 79  
 REPORT\_ON\_PWM\_IN1 (in module *ics*), 79  
 request\_enter\_sleep\_mode() (in module *ics*), 69  
 RequestEnterSleepMode() (in module *ics*), 48  
 reserved (*ics.CmISO157652RxMessage* attribute), 26  
 reserved (*ics.CyanSettings* attribute), 29  
 RESERVED (*ics.SWCanSettings* attribute), 37  
 reserved (*ics.Vcan412Settings* attribute), 42  
 reserved (*ics.Vcan4Settings* attribute), 43  
 reserved (*ics.VividCANSettings* attribute), 45  
 reserved0 (*ics.OpEthGeneralSettings* attribute), 33  
 reserved0 (*ics.OpEthSettings* attribute), 34  
 reserved\_1 (*ics.UartSettings* attribute), 41  
 RESISTOR\_OFF (in module *ics*), 79  
 RESISTOR\_ON (in module *ics*), 80  
 rsvd (*ics.EthernetSettings* attribute), 30  
 RuntimeError, 24  
 SCRIPT\_STATUS\_STOPPED (in module *ics*), 80  
 ScriptClear() (in module *ics*), 48  
 ScriptGetFBlockStatus() (in module *ics*), 49  
 ScriptGetScriptStatus() (in module *ics*), 49  
 ScriptGetScriptStatusEx() (in module *ics*), 49  
 ScriptLoad() (in module *ics*), 49  
 ScriptReadAppSignal() (in module *ics*), 49  
 ScriptReadRxMessage() (in module *ics*), 49  
 ScriptReadTxMessage() (in module *ics*), 49  
 ScriptStart() (in module *ics*), 49  
 ScriptStartFBlock() (in module *ics*), 50  
 ScriptStop() (in module *ics*), 50  
 ScriptStopFBlock() (in module *ics*), 50  
 ScriptWriteAppSignal() (in module *ics*), 50  
 ScriptWriteRxMessage() (in module *ics*), 50  
 ScriptWriteTxMessage() (in module *ics*), 50  
 SerialNumber (*ics.NeoDevice* attribute), 33  
 set\_active\_vnet\_channel() (in module *ics*), 69  
 set\_backup\_power\_enabled() (in module *ics*), 70  
 set\_bit\_rate() (in module *ics*), 70  
 set\_bit\_rate\_ex() (in module *ics*), 70  
 set\_context() (in module *ics*), 70  
 set\_device\_settings() (in module *ics*), 70  
 set\_fd\_bit\_rate() (in module *ics*), 71  
 set\_reflash\_callback() (in module *ics*), 71  
 set\_rtc() (in module *ics*), 71  
 SetActiveVNETChannel() (in module *ics*), 50  
 SetBackupPowerEnabled() (in module *ics*), 50  
 SetBaudrate (*ics.CanSettings* attribute), 25  
 SetBaudrate (*ics.SWCanSettings* attribute), 37  
 SetBitRate() (in module *ics*), 51  
 SetBitRateEx() (in module *ics*), 51  
 SetContext() (in module *ics*), 51  
 SetFDBitRate() (in module *ics*), 51  
 SetFireSettings() (in module *ics*), 51  
 SetReflashDisplayCallback() (in module *ics*), 51  
 SetRTC() (in module *ics*), 51  
 SetVCAN3Settings() (in module *ics*), 51  
 SlaveEnable (*ics.TimesyncSettings* attribute), 41  
 SlaveNetwork (*ics.TimesyncSettings* attribute), 41  
 slaveVnetA (*ics.CyanSettings* attribute), 29  
 slaveVnetB (*ics.CyanSettings* attribute), 29  
 SLEEP\_MODE (in module *ics*), 80  
 SLOW\_MODE (in module *ics*), 80  
 spbrg (*ics.Iso9141Keyword2000Settings* attribute), 32  
 spbrg (*ics.LinSettings* attribute), 33  
 spbrg (*ics.UartSettings* attribute), 41  
 SPY\_PROTOCOL\_BEAN (in module *ics*), 80  
 SPY\_PROTOCOL\_CAN (in module *ics*), 80  
 SPY\_PROTOCOL\_CANFD (in module *ics*), 80  
 SPY\_PROTOCOL\_CGI (in module *ics*), 80  
 SPY\_PROTOCOL\_CHRYSLER\_CCD (in module *ics*), 80

## S

SCRIPT\_LOCATION\_FLASH\_MEM (in module *ics*), 80  
 SCRIPT\_LOCATION\_INTERNAL\_FLASH (in module *ics*), 80  
 SCRIPT\_LOCATION\_SDCARD (in module *ics*), 80  
 SCRIPT\_LOCATION\_VCAN3\_MEM (in module *ics*), 80  
 SCRIPT\_STATUS\_RUNNING (in module *ics*), 80

SPY\_PROTOCOL\_CHRYSLER\_JVPW (in module ics), 80

SPY\_PROTOCOL\_CHRYSLER\_SCI (in module ics), 80

SPY\_PROTOCOL\_CUSTOM (in module ics), 80

SPY\_PROTOCOL\_DALLAS\_1WIRE (in module ics), 80

SPY\_PROTOCOL\_ETHERNET (in module ics), 80

SPY\_PROTOCOL\_FLEXRAY (in module ics), 80

SPY\_PROTOCOL\_FORD\_UBP (in module ics), 80

SPY\_PROTOCOL\_GENERIC\_MANCHSESTER (in module ics), 80

SPY\_PROTOCOL\_GENERIC\_UART (in module ics), 80

SPY\_PROTOCOL\_GM\_ALDL\_UART (in module ics), 80

SPY\_PROTOCOL\_GME\_CIM\_SCL\_KLINE (in module ics), 80

SPY\_PROTOCOL\_GMFSA (in module ics), 80

SPY\_PROTOCOL\_GMLAN (in module ics), 80

SPY\_PROTOCOL\_I2C (in module ics), 80

SPY\_PROTOCOL\_ISO9141 (in module ics), 80

SPY\_PROTOCOL\_J1708 (in module ics), 80

SPY\_PROTOCOL\_J1850PWM (in module ics), 80

SPY\_PROTOCOL\_J1850VPW (in module ics), 80

SPY\_PROTOCOL\_J1939 (in module ics), 80

SPY\_PROTOCOL\_JTAG (in module ics), 80

SPY\_PROTOCOL\_LIN (in module ics), 80

SPY\_PROTOCOL\_MOST (in module ics), 80

SPY\_PROTOCOL\_SENT\_PROTOCOL (in module ics), 81

SPY\_PROTOCOL\_SPI (in module ics), 81

SPY\_PROTOCOL\_TCP (in module ics), 81

SPY\_PROTOCOL\_UART (in module ics), 81

SPY\_PROTOCOL\_UNIO (in module ics), 81

SPY\_STATUS2\_CAN\_HAVE\_LINK\_DATA (in module ics), 81

SPY\_STATUS2\_CAN\_ISO15765\_LOGICAL\_FRAME (in module ics), 81

SPY\_STATUS2\_END\_OF\_LONG\_MESSAGE (in module ics), 81

SPY\_STATUS2\_ERROR\_FRAME (in module ics), 81

SPY\_STATUS2\_ETHERNET\_CRC\_ERROR (in module ics), 81

SPY\_STATUS2\_ETHERNET\_FCS\_AVAILABLE (in module ics), 81

SPY\_STATUS2\_ETHERNET\_FRAME\_TOO\_SHORT (in module ics), 81

SPY\_STATUS2\_ETHERNET\_NO\_PADDING (in module ics), 81

SPY\_STATUS2\_ETHERNET\_PREEMPTION\_ENABLED (in module ics), 81

SPY\_STATUS2\_FLEXRAY\_NO\_CRC (in module ics), 81

SPY\_STATUS2\_FLEXRAY\_NO\_HEADERCRC (in module ics), 81

SPY\_STATUS2\_FLEXRAY\_TX\_AB (in module ics), 81

SPY\_STATUS2\_FLEXRAY\_TX\_AB\_NO\_A (in module ics), 81

SPY\_STATUS2\_FLEXRAY\_TX\_AB\_NO\_B (in module ics), 81

SPY\_STATUS2\_FLEXRAY\_TX\_AB\_NO\_MATCH (in module ics), 81

SPY\_STATUS2\_GLOBAL\_CHANGE (in module ics), 81

SPY\_STATUS2\_HAS\_VALUE (in module ics), 81

SPY\_STATUS2\_HIGH\_VOLTAGE (in module ics), 81

SPY\_STATUS2\_ISO\_FRAME\_ERROR (in module ics), 81

SPY\_STATUS2\_ISO\_OVERFLOW\_ERROR (in module ics), 81

SPY\_STATUS2\_ISO\_PARITY\_ERROR (in module ics), 81

SPY\_STATUS2\_LIN\_ERR\_MSG\_ID\_PARITY (in module ics), 81

SPY\_STATUS2\_LIN\_ERR\_RX\_BREAK\_NOT\_0 (in module ics), 81

SPY\_STATUS2\_LIN\_ERR\_RX\_BREAK\_TOO\_SHORT (in module ics), 81

SPY\_STATUS2\_LIN\_ERR\_RX\_DATA\_GREATER\_8 (in module ics), 81

SPY\_STATUS2\_LIN\_ERR\_RX\_SYNC\_NOT\_55 (in module ics), 81

SPY\_STATUS2\_LIN\_ERR\_TX\_RX\_MISMATCH (in module ics), 81

SPY\_STATUS2\_LIN\_ID\_FRAME\_ERROR (in module ics), 81

SPY\_STATUS2\_LIN\_NO\_SLAVE\_DATA (in module ics), 81

SPY\_STATUS2\_LIN\_SLAVE\_BYTE\_ERROR (in module ics), 81

SPY\_STATUS2\_LIN\_SYNC\_FRAME\_ERROR (in module ics), 81

SPY\_STATUS2\_LONG\_MESSAGE (in module ics), 82

SPY\_STATUS2\_MOST\_CHANGED\_PAR (in module ics), 82

SPY\_STATUS2\_MOST\_CONTROL\_DATA (in module ics), 82

SPY\_STATUS2\_MOST\_I2S\_DUMP (in module ics), 82

SPY\_STATUS2\_MOST\_LOW\_LEVEL (in module ics), 82

SPY\_STATUS2\_MOST\_MHP\_CONTROL\_DATA (in module ics), 82

SPY\_STATUS2\_MOST\_MHP\_USER\_DATA (in module ics), 82

SPY\_STATUS2\_MOST\_MOST150 (in module ics), 82

SPY\_STATUS2\_MOST\_MOST50 (in module ics), 82

SPY\_STATUS2\_MOST\_PACKET\_DATA (in module ics), 82

SPY\_STATUS2\_MOST\_TOO\_SHORT (in module ics), 82

SPY\_STATUS2\_RX\_TIMEOUT\_ERROR (in module

- ics*), 82
  - SPY\_STATUS2\_VALUE\_IS\_BOOLEAN (*in module ics*), 82
  - SPY\_STATUS3\_CANFD\_BRS (*in module ics*), 82
  - SPY\_STATUS3\_CANFD\_ESI (*in module ics*), 82
  - SPY\_STATUS3\_CANFD\_FDF (*in module ics*), 82
  - SPY\_STATUS3\_CANFD\_IDE (*in module ics*), 82
  - SPY\_STATUS3\_CANFD\_RTR (*in module ics*), 82
  - SPY\_STATUS3\_LIN\_JUST\_BREAK\_SYNC (*in module ics*), 82
  - SPY\_STATUS3\_LIN\_ONLY\_UPDATE\_SLAVE\_TABLE (*in module ics*), 82
  - SPY\_STATUS3\_LIN\_SLAVE\_DATA\_TOO\_SHORT (*in module ics*), 82
  - SPY\_STATUS\_ANALOG\_DIGITAL\_INPUT (*in module ics*), 82
  - SPY\_STATUS\_AUDIO\_COMMENT (*in module ics*), 82
  - SPY\_STATUS\_AVSI\_REC\_OVERFLOW (*in module ics*), 82
  - SPY\_STATUS\_BAD\_MESSAGE\_BIT\_TIME\_ERROR (*in module ics*), 82
  - SPY\_STATUS\_BREAK (*in module ics*), 82
  - SPY\_STATUS\_BUS\_RECOVERED (*in module ics*), 82
  - SPY\_STATUS\_BUS\_SHORTED\_GND (*in module ics*), 82
  - SPY\_STATUS\_BUS\_SHORTED\_PLUS (*in module ics*), 82
  - SPY\_STATUS\_CAN\_BUS\_OFF (*in module ics*), 82
  - SPY\_STATUS\_CAN\_ERROR\_PASSIVE (*in module ics*), 82
  - SPY\_STATUS\_CANFD (*in module ics*), 82
  - SPY\_STATUS\_CHECKSUM\_ERROR (*in module ics*), 82
  - SPY\_STATUS\_COMM\_IN\_OVERFLOW (*in module ics*), 82
  - SPY\_STATUS\_CRC\_ERROR (*in module ics*), 82
  - SPY\_STATUS\_EXPECTED\_LEN\_MISMATCH (*in module ics*), 82
  - SPY\_STATUS\_EXTENDED (*in module ics*), 83
  - SPY\_STATUS\_FLEXRAY\_PDU (*in module ics*), 83
  - SPY\_STATUS\_FLEXRAY\_PDU\_NO\_UPDATE\_BIT (*in module ics*), 83
  - SPY\_STATUS\_FLEXRAY\_PDU\_UPDATE\_BIT\_SET (*in module ics*), 83
  - SPY\_STATUS\_GLOBAL\_ERR (*in module ics*), 83
  - SPY\_STATUS\_GPS\_DATA (*in module ics*), 83
  - SPY\_STATUS\_HEADERCRC\_ERROR (*in module ics*), 83
  - SPY\_STATUS\_HIGH\_SPEED (*in module ics*), 83
  - SPY\_STATUS\_INCOMPLETE\_FRAME (*in module ics*), 83
  - SPY\_STATUS\_INIT\_MESSAGE (*in module ics*), 83
  - SPY\_STATUS\_LIN\_MASTER (*in module ics*), 83
  - SPY\_STATUS\_LOST\_ARBITRATION (*in module ics*), 83
  - SPY\_STATUS\_MSG\_NO\_MATCH (*in module ics*), 83
  - SPY\_STATUS\_NETWORK\_MESSAGE\_TYPE (*in module ics*), 83
  - SPY\_STATUS\_PDU (*in module ics*), 83
  - SPY\_STATUS\_REMOTE\_FRAME (*in module ics*), 83
  - SPY\_STATUS\_TEST\_TRIGGER (*in module ics*), 83
  - SPY\_STATUS\_TEXT\_COMMENT (*in module ics*), 83
  - SPY\_STATUS\_TX\_MSG (*in module ics*), 83
  - SPY\_STATUS\_TX\_NOMATCH (*in module ics*), 83
  - SPY\_STATUS\_UNDEFINED\_ERROR (*in module ics*), 83
  - SPY\_STATUS\_VSI\_IFR\_CRC\_BIT (*in module ics*), 83
  - SPY\_STATUS\_VSI\_TX\_UNDERRUN (*in module ics*), 83
  - SPY\_STATUS\_XTD\_FRAME (*in module ics*), 83
  - SpyMessage (*class in ics*), 38
  - SpyMessageJ1850 (*class in ics*), 39
  - StatusBitField (*ics.SpyMessage attribute*), 38
  - StatusBitField (*ics.SpyMessageJ1850 attribute*), 40
  - StatusBitField2 (*ics.SpyMessage attribute*), 38
  - StatusBitField2 (*ics.SpyMessageJ1850 attribute*), 40
  - StatusBitField3 (*ics.SpyMessage attribute*), 38
  - StatusBitField3 (*ics.SpyMessageJ1850 attribute*), 40
  - StatusBitField4 (*ics.SpyMessage attribute*), 39
  - StatusBitField4 (*ics.SpyMessageJ1850 attribute*), 40
  - stMin (*ics.CmlISO157652RxMessage attribute*), 26
  - stMin (*ics.CmlISO157652TxMessage attribute*), 26
  - stop\_bits (*ics.UartSettings attribute*), 41
  - swcan (*ics.FireSettings attribute*), 32
  - swcan1 (*ics.CyanSettings attribute*), 29
  - swcan1 (*ics.RadGalaxySettings attribute*), 36
  - swcan1 (*ics.VividCANSettings attribute*), 45
  - swcan2 (*ics.CyanSettings attribute*), 29
  - swcan2 (*ics.RadGalaxySettings attribute*), 36
  - SWCAN\_AUTOSWITCH\_DISABLED (*in module ics*), 83
  - SWCAN\_AUTOSWITCH\_DISABLED\_RESISTOR\_ENABLED (*in module ics*), 83
  - SWCAN\_AUTOSWITCH\_NO\_RESISTOR (*in module ics*), 83
  - SWCAN\_AUTOSWITCH\_WITH\_RESISTOR (*in module ics*), 83
  - SWCAN\_SETTINGS\_SIZE (*in module ics*), 83
  - SWCanSettings (*class in ics*), 37
- ## T
- tapPair0 (*ics.OpEthGeneralSettings attribute*), 33
  - tapPair1 (*ics.OpEthGeneralSettings attribute*), 33
  - tapPair2 (*ics.OpEthGeneralSettings attribute*), 33
  - tapPair3 (*ics.OpEthGeneralSettings attribute*), 33

tapPair4 (*ics.OpEthGeneralSettings* attribute), 33  
tapPair5 (*ics.OpEthGeneralSettings* attribute), 33  
termination\_enables (*ics.CyanSettings* attribute), 29  
termination\_enables (*ics.Vcan4I2Settings* attribute), 42  
termination\_enables (*ics.Vcan4Settings* attribute), 43  
termination\_enables (*ics.VividCANSettings* attribute), 45  
text\_api (*ics.CyanSettings* attribute), 29  
text\_api (*ics.FireSettings* attribute), 32  
text\_api (*ics.RadGalaxySettings* attribute), 36  
text\_api (*ics.RadStar2Settings* attribute), 37  
text\_api (*ics.Vcan4I2Settings* attribute), 42  
text\_api (*ics.Vcan4Settings* attribute), 43  
TextApiSettings (class in *ics*), 40  
time\_500us (*ics.Iso9141Keyword2000InitSteps* attribute), 32  
TimeHardware (*ics.SpyMessage* attribute), 39  
TimeHardware (*ics.SpyMessageJ1850* attribute), 40  
TimeHardware2 (*ics.SpyMessage* attribute), 39  
TimeHardware2 (*ics.SpyMessageJ1850* attribute), 40  
TimeStampHardwareID (*ics.SpyMessage* attribute), 39  
TimeStampHardwareID (*ics.SpyMessageJ1850* attribute), 40  
TimeStampSystemID (*ics.SpyMessage* attribute), 39  
TimeStampSystemID (*ics.SpyMessageJ1850* attribute), 40  
TimesyncSettings (class in *ics*), 41  
timeSyncSettings (*ics.RadStar2Settings* attribute), 37  
TimeSystem (*ics.SpyMessage* attribute), 39  
TimeSystem (*ics.SpyMessageJ1850* attribute), 40  
TimeSystem2 (*ics.SpyMessage* attribute), 39  
TimeSystem2 (*ics.SpyMessageJ1850* attribute), 40  
TqProp (*ics.CanSettings* attribute), 25  
TqProp (*ics.SWCanSettings* attribute), 37  
TqSeg1 (*ics.CanSettings* attribute), 25  
TqSeg1 (*ics.SWCanSettings* attribute), 38  
TqSeg2 (*ics.CanSettings* attribute), 25  
TqSeg2 (*ics.SWCanSettings* attribute), 38  
TqSync (*ics.CanSettings* attribute), 25  
TqSync (*ics.SWCanSettings* attribute), 38  
transceiver\_mode (*ics.CanSettings* attribute), 25  
transceiver\_mode (*ics.SWCanSettings* attribute), 38  
transmit\_messages () (in module *ics*), 71  
tx\_index (*ics.CmISO157652TxMessage* attribute), 26  
TxMessages () (in module *ics*), 51  
uart2 (*ics.FireSettings* attribute), 32  
UART\_SETTINGS\_SIZE (in module *ics*), 83  
UartSettings (class in *ics*), 41  
ucConfigMode (*ics.OpEthSettings* attribute), 34  
ucInterfaceType (*ics.OpEthGeneralSettings* attribute), 33  
usbHostPowerEnabled (*ics.Fire2DeviceStatus* attribute), 30  
USE\_TQ (in module *ics*), 83

## V

validate\_hobject () (in module *ics*), 72  
ValidateHObject () (in module *ics*), 52  
vcan3 (*ics.DeviceSettings* attribute), 29  
Vcan3Settings (class in *ics*), 41  
vcan4 (*ics.DeviceSettings* attribute), 29  
Vcan4I2Settings (class in *ics*), 41  
vcan4\_12 (*ics.DeviceSettings* attribute), 29  
Vcan4DeviceStatus (class in *ics*), 42  
Vcan4Settings (class in *ics*), 42  
vcan4Status (*ics.IcsDeviceStatus* attribute), 32  
VcanRFSettings (class in *ics*), 43  
vividcan (*ics.DeviceSettings* attribute), 30  
VividCANSettings (class in *ics*), 44  
vnetBits (*ics.FireSettings* attribute), 32  
VNETBITS\_FEATURE\_ANDROID\_MSGS (in module *ics*), 83  
VNETBITS\_FEATURE\_DISABLE\_USB\_CHECK (in module *ics*), 83  
vs\_netid (*ics.CmISO157652RxMessage* attribute), 26  
vs\_netid (*ics.CmISO157652TxMessage* attribute), 26

## W

write\_sdcard () (in module *ics*), 72  
WriteSDCard () (in module *ics*), 52

## U

uart (*ics.FireSettings* attribute), 32