

CanLine-1

.NET Library

Programmer's Manual

Edition	Publication date	Revision history
1st edition	October 2026	Initial English release

Contents

1. Overview	2
2. Adding a DLL Reference.....	3
3. Properties.....	5
4. Function List.....	7
5. Function Reference	8
5.1. Open: Open.....	8
5.2. Close: Close	9
5.3. GetFirmVersion: Read Firmware Version	10
5.4. OpenEx: Open by Product ID.....	11
5.5. GetProductIdList: Get Product ID List.....	12
5.6. CanSetConfig: Configure CAN Communication	13
5.7. CanSetIdFilter: Configure CAN ID Filters	15
5.8. CanStart: Start CAN Communication.....	17
5.9. CanStop: Stop CAN Communication	18
5.10. CanSingleSend: Send a Single CAN Message.....	19
5.11. CanSetSendBox: Update CAN Transmit Box	20
5.12. CanStartIntervalSend: Start Periodic CAN Transmission	22
5.13. CanStartScheduleSend: Start Scheduled CAN Transmission	23
5.14. CanStopSend: Stop Periodic or Scheduled CAN Transmission	24
5.15. CanGetMessage: Get CAN Notification Messages.....	25
5.16. LinSetConfig: Configure LIN Communication	27
5.17. LinStart: Start LIN Communication.....	28
5.18. LinStop: End LIN Communication	29
5.19. LinSingleSend: Send a Single LIN Message	30
5.20. LinSetMasterSendBox: Update LIN Master Simulation Transmit Box	31
5.21. LinStartMasterSimulation1: Start Periodic LIN Master Simulation	33
5.22. LinStartMasterSimulation2: Start Scheduled LIN Master Simulation.....	34
5.23. LinSetSlaveSendBox: Update LIN Slave Simulation Transmit Box	35
5.24. LinStartSlaveSimulation: Start LIN Slave Simulation	37
5.25. LinStopSimulation: Stop LIN Simulation	38
5.26. LinGetMessage: Get LIN Notification Messages	39

1. Overview

The CanLine-1 .NET library is a dynamic link library (DLL) for .NET that controls the CanLine-1 device. It enables you to create custom applications using CanLine-1 in Visual Studio with languages such as Visual Basic and C#.

2. Adding a DLL Reference

Prerequisites

Download the following files from the product page below.

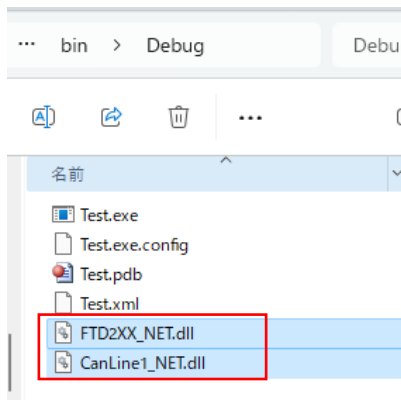
<https://hsdev.co.jp/canline-1-en/>

- [.NET Library Version *.*]: The .NET library
- [Application Version *.*]: A ZIP file containing the USB driver

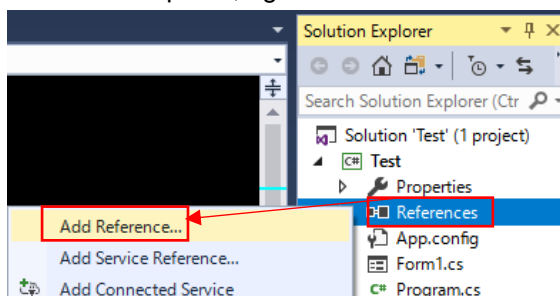
Extract the application ZIP file, then run the installer in the USB_driver folder to install the USB driver.

① Place the following two DLLs in the same folder as the executable generated by Visual Studio.

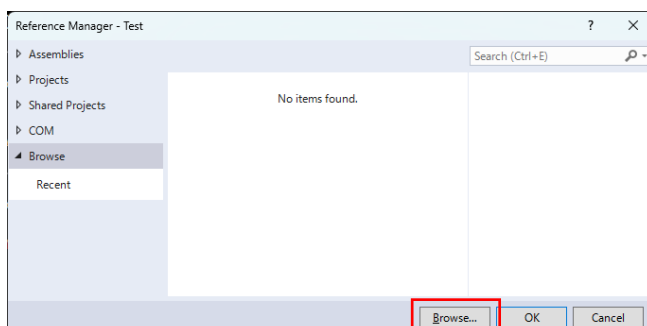
- FTD2XX_NET.dll (USB library)
- CanLine1_NET.dll (CanLine-1 library)



② In Solution Explorer, right-click References and select Add Reference.

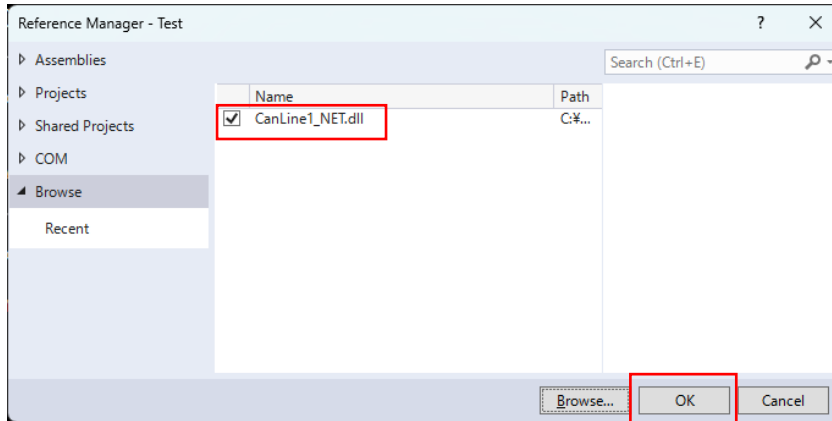


③ Click Browse.

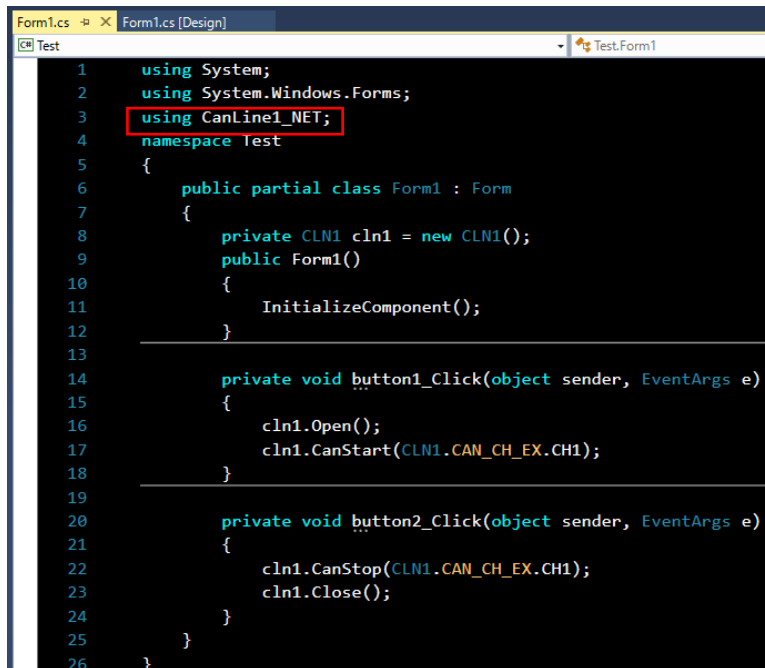


④ Select the CanLine1_NET.dll file placed in the folder above.

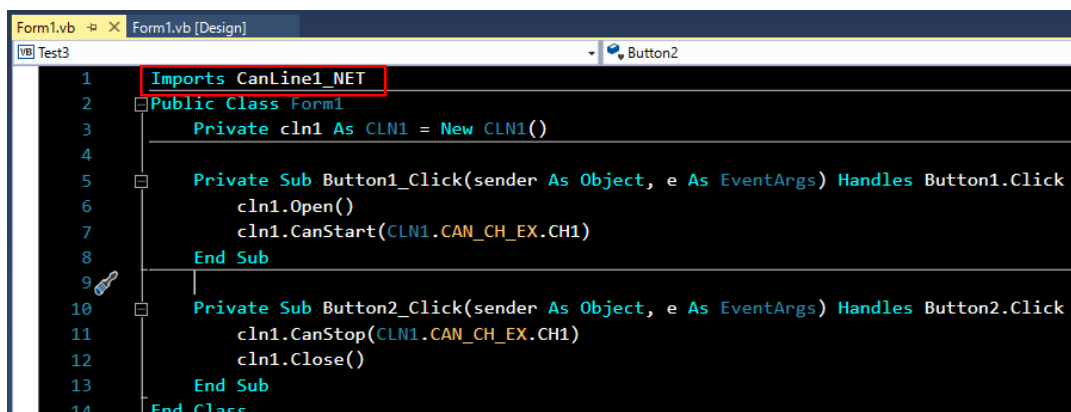
⑤ Select the checkbox for CanLine1_NET.dll and click OK.



⑥ For C#, add "using CanLine1_NET;" at the beginning of the source code.



For Visual Basic, add "Imports CanLine1_NET".



3. Properties

[Syntax]

```
public bool IsOpen { get; }
public CAN_STATUS Can1Status { get; }
public CAN_STATUS Can2Status { get; }
public CAN_MODE Can1Mode { get; }
public CAN_MODE Can2Mode { get; }
public int CanRxMsgCount { get; }
public int CanRxMsgBufferSize { get; set; }
public LIN_STATUS LinStatus { get; }
public int LinRxMsgCount { get; }
public int LinRxMsgBufferSize { get; set; }
public string OtherErrorMsg { get; }
public bool Can1ScheduleEnd{ get; }
public bool Can2ScheduleEnd{ get; }
public bool LinScheduleEnd{ get; }
```

[Details]

Property name	Read/ Write	Description	Value
IsOpen	Read	USB driver Open state	false: Closed true: Open
Can1Status	Read	CAN1 bus Status	• CAN_STATUS.BUSOFF: Bus off • CAN_STATUS.BUSON: Bus on • CAN_STATUS.ERR_PASIVE: Error passive • CAN_STATUS.ERR_BUSOFF: Bus off due to an error
Can2Status	Read	CAN2 bus Status	• CAN_STATUS.BUSOFF: Bus off • CAN_STATUS.BUSON: Bus on • CAN_STATUS.ERR_PASIVE: Error passive • CAN_STATUS.ERR_BUSOFF: Bus off due to an error
Can1Mode	Read	CAN1 mode	• CAN_MODE.SILENT: Silent mode • CAN_MODE.NORMAL: Normal mode

Can2Mode	Read	CAN2 mode	• CAN_MODE.SILENT: Silent mode • CAN_MODE.NORMAL: Normal mode
CanRxMsgCount	Read	CAN notification Message count	0 to CAN notification message buffer size CanGetMessage can retrieve the number of messages indicated by this property.
CanRxMsgBufferSize	Read/ Write	CAN notification Message Buffer size	Default: 50000 When the limit is exceeded, the oldest messages are deleted first.
LinStatus	Read	LIN bus Status	• LIN_STATUS.BUSOFF: Bus off • LIN_STATUS.BUSON: Bus on • LIN_STATUS.TX_ERROR: Transmission error
LinRxMsgCount	Read	LIN notification Message count	0 to LIN notification message buffer size LinGetMessage can retrieve the number of messages indicated by this property.
LinRxMsgBufferSize	Read/ Write	LIN notification Message Buffer size	Default: 50000 When the limit is exceeded, the oldest messages are deleted first.
OtherErrorMsg	Read	Error Message	Detailed error message when a function returns CL_RETURN.OTHER_ERROR.
Can1ScheduleEnd	Read	CAN1 scheduled transmission complete	true: CAN1 scheduled transmission has completed. Note: This does not become true when the repeat count is 0 (infinite repetition). false: Scheduled transmission has not completed.
Can2ScheduleEnd	Read	CAN2 scheduled transmission complete	true: CAN2 scheduled transmission has completed. Note: This does not become true when the repeat count is 0 (infinite repetition). false: Scheduled transmission has not completed.
LinScheduleEnd	Read	LIN scheduled transmission complete	true: LIN scheduled transmission has completed. Note: This does not become true when the repeat count is 0 (infinite repetition). false: Scheduled transmission has not completed.

4. Function List

Common CAN/LIN Functions

Function name	Operation	Description
Open	Open	Opens the USB driver.
Close	Close	Closes the USB driver.
GetFirmVersion	Read firmware version	Reads the firmware version of the CanLine-1 device.
OpenEx	Open by product ID	Opens the USB driver for the specified product ID. Supports multiple devices connected to one PC.
GetProductIdList	Get product ID list	Retrieves the list of product IDs connected to the PC.

CAN Functions

Function name	Operation	Description
CanSetConfig	Configure CAN communication	Updates the CAN communication settings.
CanSetIdFilter	Configure CAN ID filters	Filters CAN IDs.
CanStart	Start CAN communication	Starts CAN communication.
CanStop	Stop CAN communication	Stops CAN communication.
CanSingleSend	Send a single CAN message	Sends a single CAN message.
CanSetSendBox	Update CAN transmit box	Updates a transmit box for periodic or scheduled CAN transmission.
CanStartIntervalSend	Start periodic CAN transmission	Starts periodic CAN transmission.
CanStartScheduleSend	Start scheduled CAN transmission	Starts scheduled CAN transmission.
CanStopSend	Stop periodic CAN transmission	Stops periodic or scheduled CAN transmission.
CanGetMessage	Get CAN notification messages	Retrieves CAN notification messages.

LIN Functions

Function name	Operation	Description
LinSetConfig	Configure LIN communication	Updates the LIN communication settings.
LinStart	Start LIN communication	Starts LIN communication.
LinStop	Stop LIN communication	Ends LIN communication.
LinSingleSend	Send a single LIN message	Sends a single LIN message.
LinSetMasterSendBox	Update LIN master simulation transmit box	Updates a transmit box for LIN master simulation (periodic or scheduled transmission).
LinStartMasterSimulation1	Start LIN master simulation (periodic transmission)	Starts LIN master simulation (periodic transmission).
LinStartMasterSimulation2	Start LIN master simulation (scheduled transmission)	Starts LIN master simulation (scheduled transmission).
LinSetSlaveSendBox	Update LIN slave simulation transmit box	Updates a transmit box for LIN slave simulation (response).
LinStartSlaveSimulation	Start LIN slave simulation	Starts LIN slave simulation.
LinStopSimulation	Stop LIN simulation	Stops LIN simulation (master or slave).
LinGetMessage	Get LIN notification messages	Retrieves LIN notification messages.

5. Function Reference

5.1. Open: Open

[Description]

Opens the USB driver. Call this function first to start controlling CanLine-1. Once this function completes successfully, the other functions can be called.

[Syntax]

CL_RETURN Open()

[Parameters]

None

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_FOUND	Device (CanLine-1) not found.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.2. Close: Close

[Description]

Closes the USB driver.

Call this function to stop controlling CanLine-1.

[Syntax]

CL_RETURN Close()

[Parameters]

None

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.OTHER_ERROR	Other error Note: See the OtherErrorMsg property for error details.
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.

5.3. GetFirmVersion: Read Firmware Version

[Description]

Reads the firmware version of the CanLine-1 device.

[Syntax]

CL_RETURN GetFirmVersion(ref string version)

[Parameters]

Variable	I/O	Description	Value
version	OUT	Firmware version	Example: 1.0.1

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.DEVICE_RESPONSE_ERROR	Invalid response from the CanLine-1 device.
CL_RETURN.DEVICE_NO_RESPONSE	No response from the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.

5.4. OpenEx: Open by Product ID

[Description]

Opens the USB driver for the specified product ID. Call this function first to start controlling CanLine-1. Once this function completes successfully, the other functions can be called.

Note: Use this function when using multiple CanLine-1 devices simultaneously on one PC. You can find the product ID under Settings -> System Settings in the standard CanLine-1 application.

[Syntax]

CL_RETURN OpenEx(string productID)

[Parameters]

Variable	I/O	Description	Value
productID	IN	Product ID	Example: AB0NZ3P8

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_FOUND	Device (CanLine-1) with the specified product ID not found.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.5. GetProductIdList: Get Product ID List

[Description]

Retrieves the list of product IDs of devices connected to the PC. Call this function before calling Open or OpenEx. Devices that have already been opened using Open or OpenEx are not included in the product ID list.

[Syntax]

CL_RETURN GetProductIdList (ref List<string> lstProductId)

[Parameters]

Variable	I/O	Description	Value
lstProductId	OUT	Product ID list	Example: AB0NZ3P8, AB0NZ3P7...

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_FOUND	Device (CanLine-1) not found.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.6. CanSetConfig: Configure CAN Communication

[Description]

Updates the CAN communication settings. Call before starting CAN communication. CanStop (stop CAN communication) resets these settings, so call this function each time before CanStart (start CAN communication).

[Syntax]

CL_RETURN CanSetConfig(CAN_CH chNo, CAN_CONFIG config)

[Parameters]

Variable name	I/O	Description	Value
chNo	IN	Channel No.	• CAN_CH.CH1: CAN1 • CAN_CH.CH2: CAN2
config	IN	CAN communication settings	See the <u>CAN CONFIG members</u> below.

CAN CONFIG Members

```
public class CAN_CONFIG{  
    public CAN_TERM Terminator;  
    public CAN_MODE Mode;  
    public CAN_RATE BaudRate;  
    public CAN_SP SamplePoint;  
}
```

Variable	Description	Value
Terminator	Termination resistor (120 Ω)	• CAN_TERM.OFF: Termination resistor disabled • CAN_TERM.ON: Termination resistor enabled
Mode	CAN mode	• CAN_MODE.SILENT: Silent mode • CAN_MODE.NORMAL: Normal mode
BaudRate	Bit rate [bps]	• CAN_RATE._125K: 125 Kbps • CAN_RATE._250K: 250 Kbps • CAN_RATE._500K: 500 Kbps • CAN_RATE._1000K: 1000 Kbps
SamplePoint	Sample point	• CAN_SP._60Per: 60% • CAN_SP._70Per: 70% • CAN_SP._80per: 80%

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.CAN_STARTED	The CAN communication start function has already been called. Call this function before starting CAN communication.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.DEVICE_RESPONSE_ERROR	Invalid response from the CanLine-1 device.
CL_RETURN.DEVICE_NO_RESPONSE	No response from the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.7. CanSetIdFilter: Configure CAN ID Filters

[Description]

Filters IDs by specifying the ranges of CAN IDs to receive. Up to 10 ranges can be specified. Call before starting CAN communication. CanStop (stop CAN communication) resets the filters, so to enable filtering, call this function each time before CanStart (start CAN communication).

[Syntax]

CL_RETURN CanSetIdFilter(CAN_CH chNo, CAN_ID_FILTER filter)

[Parameters]

Variable	I/O	Description	Value
chNo	IN	Channel No.	• CAN_CH.CH1: CAN1 • CAN_CH.CH2: CAN2
filter	IN	CAN ID filter settings	See the <u>CAN ID FILTER members</u> below.

CAN ID FILTER Members

```
public class CAN_ID_FILTER{  
    public int FilterNo;  
    public CAN_ID_TYPE IdType;  
    public UInt32 FromID;  
    public UInt32 ToID;  
}
```

Variable	Description	Value
FilterNo	Filter No.	0~9 Up to 10 filters can be configured.
IdType	CAN ID type	• CAN_ID_TYPE.STD: Standard ID (11 bits) • CAN_ID_TYPE.EXT: Extended ID (29 bits)
FromID	Start ID of receive range	For standard CAN IDs: 0 to 0x7FF; for extended CAN IDs: 0 to 0x1FFFFFFF.
ToID	End ID of receive range	For standard CAN IDs: 0 to 0x7FF; for extended CAN IDs: 0 to 0x1FFFFFFF.

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.CAN_STARTED	The CAN communication start function has already been called. Call this function before starting CAN communication.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.DEVICE_RESPONSE_ERROR	Invalid response from the CanLine-1 device.
CL_RETURN.DEVICE_NO_RESPONSE	No response from the CanLine-1 device.
CL_RETURN.PARAMETER_ERROR	Parameter out of range.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.8. CanStart: Start CAN Communication

[Description]

Starts CAN communication. CAN messages can be transmitted and received after calling this function.

[Syntax]

CL_RETURN CanStart(CAN_CH_EX chNo)

[Parameters]

Variable	I/O	Description	Value
chNo	IN	Channel No.	• CAN_CH_EX.CH1: CAN1 • CAN_CH_EX.CH2: CAN2 • CAN_CH_EX.ALL: Both CAN1 and CAN2

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.9. CanStop: Stop CAN Communication

[Description]

Stops CAN communication.

[Syntax]

CL_RETURN CanStop(CAN_CH_EX chNo)

[Parameters]

Variable	I/O	Description	Value
chNo	IN	Channel No.	- CAN_CH_EX.CH1: CAN1 - CAN_CH_EX.CH2: CAN2 - CAN_CH_EX.ALL: Both CAN1 and CAN2

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.10. CanSingleSend: Send a Single CAN Message

[Description]

Sends a single CAN message.

[Syntax]

CL_RETURN CanSingleSend(CAN_CH chNo, CAN_TX_MSG txMsg)

[Parameters]

Variable	I/O	Description	Value
chNo	IN	Channel No.	• CAN_CH.CH1: CAN1 • CAN_CH.CH2: CAN2
txMsg	IN	CAN transmit message information	See the <u>CAN TX MSG members</u> below.

CAN TX MSG Members

```
public class CAN_TX_MSG{  
    public CAN_ID_TYPE IdType;  
    public UInt32 Id;  
    public byte Dlc;  
    public byte[] Data = new byte[8];  
}
```

Variable	Description	Value
IdType	CAN ID type	• CAN_ID_TYPE.STD: Standard ID (11 bits) • CAN_ID_TYPE.EXT: Extended ID (29 bits)
Id	CAN ID	For standard CAN IDs: 0 to 0x7FF; for extended CAN IDs: 0 to 0x1FFFFFFF.
Dlc	Data length	1~8
Data[0]~[7]	Data	0~0xFF

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.CAN_NOT_STARTED	The CAN communication start function has not been called.
CL_RETURN.CAN_MODE_SILENT	CAN is in silent mode. Call this function in normal mode.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.11. CanSetSendBox: Update CAN Transmit Box

[Description]

Updates a CAN transmit box. Up to 20 transmit boxes can be configured.

After calling this function, call CanStartIntervalSend to start periodic CAN transmission or

CanStartScheduleSend to start scheduled CAN transmission. This function can also be called during periodic or scheduled transmission to update the data being transmitted.

[Syntax]

CL_RETURN CanSetSendBox(CAN_CH chNo, int txBoxNo, CAN_TX_MSG txMsg, UInt32 time)

[Parameters]

Variable	I/O	Description	Value
chNo	IN	CAN channel No.	• CAN_CH.CH1: CAN1 • CAN_CH.CH2: CAN2
txBoxNo	IN	Transmit box No.	0 to 19 (0 = transmit box 1 ... 19 = transmit box 20)
txMsg	IN	CAN transmit message information	See the <u>CAN TX MSG members</u> below.
time	IN	Periodic transmission interval [ms] / scheduled transmission time [ms]	1~99999

CAN TX MSG Members

```
public class CAN_TX_MSG{  
    public CAN_ID_TYPE IdType;  
    public UInt32 Id;  
    public byte Dlc;  
    public byte[] Data = new byte[8];  
}
```

Variable	Description	Value
IdType	CAN ID type	• CAN_ID_TYPE. STD: Standard ID (11 bits) • CAN_ID_TYPE. EXT: Extended ID (29 bits)
Id	CAN ID	For standard CAN IDs: 0 to 0x7FF; for extended CAN IDs: 0 to 0x1FFFFFFF.
Dlc	Data length	1~8
Data[0]~[7]	Data	0~0xFF

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.PARAMETER_ERROR	Parameter out of range.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.12. CanStartIntervalSend: Start Periodic CAN Transmission

[Description]

Call this function after CanSetSendBox (update CAN transmit box) to start periodic CAN transmission. Specify whether transmit boxes 1 to 20 are enabled or disabled.

[Syntax]

CL_RETURN CanStartIntervalSend(CAN_CH chNo, bool[] txBoxNoEnable)

[Parameters]

Variable	I/O	Description	Value
chNo	IN	CAN channel No.	- CAN_CH.CH1: CAN1 - CAN_CH.CH2: CAN2
txBoxNoEnable [0]~[19]	IN	txBoxNoEnable[0]: Enable/disable transmit box 1 txBoxNoEnable[1]: Enable/disable transmit box 2 ... txBoxNoEnable[19]: Enable/disable transmit box 20	- false: Disabled - true: Enabled

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.CAN_NOT_STARTED	The CAN communication start function has not been called.
CL_RETURN.CAN_MODE_SILENT	CAN is in silent mode. Call this function in normal mode.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

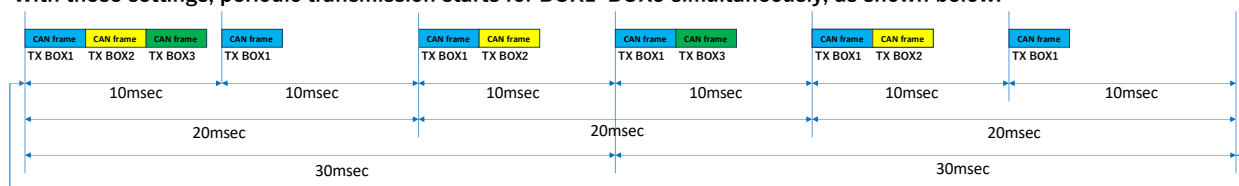
[Periodic transmission]

BOX1=10msec

BOX2=20msec

BOX3=30msec

With these settings, periodic transmission starts for BOX1-BOX3 simultaneously, as shown below.



5.13. CanStartScheduleSend: Start Scheduled CAN Transmission

[Description]

Call this function after CanSetSendBox (update CAN transmit box) to start scheduled CAN transmission. Specify whether transmit boxes 1 to 20 are enabled or disabled. A repeat count of 0 repeats transmission until CanStopSend is called. A value of 1 or greater repeats transmission the specified number of times. Completion can be checked using the Can1ScheduleEnd and Can2ScheduleEnd properties.

[Syntax]

CL_RETURN CanStartScheduleSend(CAN_CH chNo, bool[] txBoxNoEnable, UInt16 repeatCount)

[Parameters]

Variable	I/O	Description	Value
chNo	IN	CAN channel No.	- CAN_CH.CH1: CAN1 - CAN_CH.CH2: CAN2
txBoxNoEnable [0]~[19]	IN	txBoxNoEnable[0]: Enable/disable transmit box 1 txBoxNoEnable[1]: Enable/disable transmit box 2 ... txBoxNoEnable[19]: Enable/disable transmit box 20	- false: Disabled - true: Enabled
repeatCount	IN	Repeat count	0 = Infinite repetition 1 to 999 repetitions

[Return Value]

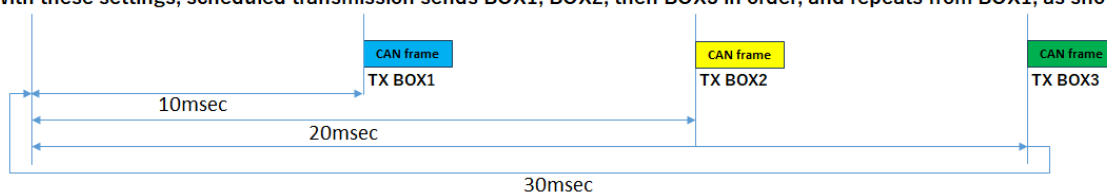
Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.CAN_NOT_STARTED	The CAN communication start function has not been called.
CL_RETURN.CAN_MODE_SILENT	CAN is in silent mode. Call this function in normal mode.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

BOX1=10msec

BOX2=20msec

BOX3=30msec

With these settings, scheduled transmission sends BOX1, BOX2, then BOX3 in order, and repeats from BOX1, as shown below



5.14. CanStopSend: Stop Periodic or Scheduled CAN Transmission

[Description]

Stops periodic or scheduled CAN transmission.

[Syntax]

CL_RETURN CanStopSend(CAN_CH chNo)

[Parameters]

Variable	I/O	Description	Value
chNo	IN	CAN channel No.	• CAN_CH.CH1: CAN1 • CAN_CH.CH2: CAN2

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.15. CanGetMessage: Get CAN Notification Messages

[Description]

Retrieves notification messages for completed CAN transmissions and received CAN messages.

[Syntax]

```
int CanGetMessage(ref List<CAN_RX_MSG> lstRxMsg, int count)
```

[Parameters]

Variable	I/O	Description	Value
lstRxMsg	OUT	List of CAN notification message information	See the <u>CAN_RX_MSG members</u> below.
count	IN	Number of messages to retrieve	Read the received message count using the CanRxMsgCount property and specify a value no greater than that count.

CAN_RX_MSG Members

```
public class CAN_RX_MSG{  
    public UInt32 TimeStamp;  
    public CAN_RX_MSG_TYPE Type;  
    public CAN_ID_TYPE IdType;  
    public UInt32 Id;  
    public byte Dlc;  
    public byte[] Data = new byte[8];  
}
```

Variable	Description	Value
TimeStamp	Time stamp	0~4294967295 Resolution: 0.1 ms. Example: 123 = 12.3 ms
Type	CAN notification Message Type	• CAN_RX_MSG_TYPE.CAN1_RECIEVD: CAN1 message received • CAN_RX_MSG_TYPE.CAN1_SENT: CAN1 transmission completed • CAN_RX_MSG_TYPE.CAN2_RECIEVD: CAN2 message received • CAN_RX_MSG_TYPE.CAN2_SENT: CAN2 transmission completed
IdType	CAN ID Type	• CAN_ID_TYPE.STD: Standard ID (11 bits) • CAN_ID_TYPE.EXT: Extended ID (29 bits)
Id	CAN ID	For standard CAN IDs: 0 to 0x7FF; for extended CAN IDs: 0 to 0x1FFFFFFF.
Dlc	Data length	1~8
Data[0]~[7]	Data	0~0xFF

[Return Value]

Value	Description
int	Number of messages retrieved

5.16. LinSetConfig: Configure LIN Communication

[Description]

Updates the LIN communication settings.

[Syntax]

CL_RETURN LinSetConfig(LIN_CONFIG config)

[Parameters]

Variable	I/O	Description	Value
config	IN	LIN communication settings	See the <u>LIN_CONFIG members</u> below.

LIN_CONFIG Members

```
public class LIN_CONFIG{  
    public LIN_RATE BaudRate;  
    public LIN_PULLUP PullUp;  
}
```

Variable	Description	Value
BaudRate	LIN bit rate [bps]	• LIN_RATE._2400: 2400 bps • LIN_RATE._4800: 4800 bps • LIN_RATE._9600: 9600 bps • LIN_RATE._19200: 19200 bps
PullUp	Master pull-up resistor (1kΩ)	• LIN_PULLUP.OFF: Pull-up resistor disabled • LIN_PULLUP.ON: Pull-up resistor enabled

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.LIN_STARTED	The LIN communication start function has already been called. Call this function before starting LIN communication.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.DEVICE_RESPONSE_ERROR	Invalid response from the CanLine-1 device.
CL_RETURN.DEVICE_NO_RESPONSE	No response from the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.17. LinStart: Start LIN Communication

[Description]

Starts LIN communication. LIN messages can be transmitted and received after calling this function.

[Syntax]

CL_RETURN LinStart()

[Parameters]

None

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.18. LinStop: End LIN Communication

[Description]

Ends LIN communication.

[Syntax]

CL_RETURN LinStop()

[Parameters]

None

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.19. LinSingleSend: Send a Single LIN Message

[Description]

Sends a single LIN message.

[Syntax]

CL_RETURN LinSingleSend(LIN_TX_MSG txMsg)

[Parameters]

Variable	I/O	Description	Value
txMsg	IN	LIN transmit message information	See the <u>LIN_TX_MSG members</u> below.

LIN_TX_MSG Members

```
public class LIN_TX_MSG{  
    public byte Id;  
    public byte Dlc;  
    public byte[] Data = new byte[8];  
    public LIN_SUM_TYPE SumType;  
}
```

Variable	Description	
Id	LIN ID	0~0x3F
Dlc	Data length	0~8
Data[0]~[7]	Data	0~0xFF
SumType	Checksum Type	• LIN_SUM_TYPE.CLASSIC: Classic checksum • LIN_SUM_TYPE.ENHANCED: Enhanced checksum

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.LIN_NOT_STARTED	The LIN communication start function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.20. LinSetMasterSendBox: Update LIN Master Simulation Transmit Box

[Description]

Updates a LIN transmit box. Up to 20 transmit boxes can be configured.

After calling this function, call `LinStartMasterSimulation1` to start LIN master simulation with periodic transmission or `LinStartMasterSimulation2` to start LIN master simulation with scheduled transmission. This function can also be called during periodic or scheduled transmission to update the data being transmitted.

[Syntax]

CL_RETURN `LinSetMasterSendBox`(int txBoxNo, LIN_TX_MSG txMsg, UInt32 time)

[Parameters]

Variable	I/O	Description	Value
txBoxNo	IN	Transmit box No.	0 to 19 (0 = transmit box 1 ... 19 = transmit box 20)
txMsg	IN	LIN transmit message information	See the <u>LIN TX MSG members</u> below.
time	IN	Periodic transmission interval [ms] / scheduled transmission time [ms]	1~99999

LIN TX MSG Members

```
public class LIN_TX_MSG{  
    public byte Id;  
    public byte Dlc;  
    public byte[] Data = new byte[8];  
    public LIN_SUM_TYPE SumType;  
}
```

Variable	Description	
Id	LIN ID	0~0x3F
Dlc	Data length	0~8 Note: Selecting 0 allows a response from a slave to be received.
Data[0]~[7]	Data	0~0xFF
SumType	Checksum type	- LIN_SUM_TYPE.CLASSIC: Classic checksum - LIN_SUM_TYPE.ENHANCED: Enhanced checksum

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.PARAMETER_ERROR	Parameter out of range.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.21. LinStartMasterSimulation1: Start Periodic LIN Master Simulation

[Description]

Call this function after LinSetMasterSendBox (update LIN master simulation transmit box) to start LIN master simulation with periodic transmission. Specify whether transmit boxes 1 to 20 are enabled or disabled.

[Syntax]

CL_RETURN LinStartMasterSimulation1(bool[] txBoxNoEnable)

[Parameters]

Variable	I/O	Description	Value
txBoxNoEnable[0]~ [19]	IN	txBoxNoEnable[0]: Enable/disable transmit box 1 txBoxNoEnable[1]: Enable/disable transmit box 2 ... txBoxNoEnable[19]: Enable/disable transmit box 20	- false: Disabled - true: Enabled

[Return Value]

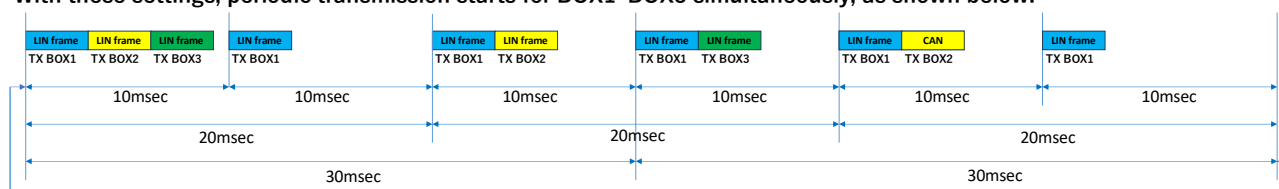
Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.LIN_NOT_STARTED	The LIN communication start function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

BOX1=10msec

BOX2=20msec

BOX3=30msec

With these settings, periodic transmission starts for BOX1–BOX3 simultaneously, as shown below.



5.22. LinStartMasterSimulation2: Start Scheduled LIN Master Simulation

[Description]

Call this function after LinSetMasterSendBox (update LIN master simulation transmit box) to start LIN master simulation with scheduled transmission. Specify whether transmit boxes 1 to 20 are enabled or disabled. A repeat count of 0 repeats transmission indefinitely until LinStopSimulation is called. A value of 1 or greater repeats transmission the specified number of times. Completion can be checked using the LinScheduleEnd property.

[Syntax]

CL_RETURN LinStartMasterSimulation2(bool[] txBoxNoEnable, UInt16 repeatCount)

[Parameters]

Variable	I/O	Description	Value
txBoxNoEnable[0]~ [19]	IN	txBoxNoEnable[0]: Enable/disable transmit box 1 txBoxNoEnable[1]: Enable/disable transmit box 2 ... txBoxNoEnable[19]: Enable/disable transmit box 20	- false: Disabled - true: Enabled
repeatCount	IN	Repeat count	0 = Infinite repetition 1 to 999 repetitions

[Return Value]

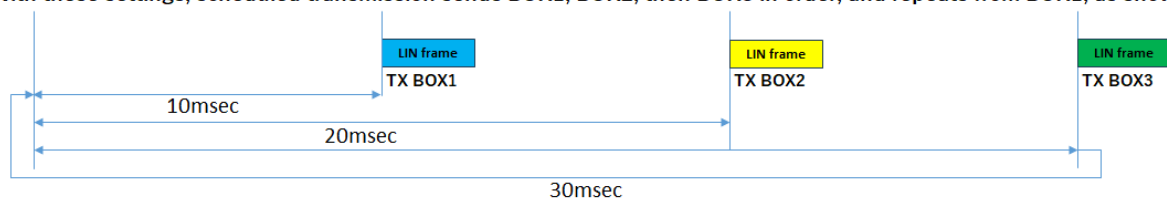
Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.LIN_NOT_STARTED	The LIN communication start function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

BOX1=10msec

BOX2=20msec

BOX3=30msec

With these settings, scheduled transmission sends BOX1, BOX2, then BOX3 in order, and repeats from BOX1, as shown below.



5.23. LinSetSlaveSendBox: Update LIN Slave Simulation Transmit Box

[Description]

Updates a transmit box for LIN slave simulation (response). Up to 20 types of slave responses are supported. After calling this function, call `LinStartSlaveSimulation` (start LIN slave simulation) to start slave responses. Calling this function after slave simulation has started updates the slave response data.

[Syntax]

CL_RETURN `LinSetSlaveSendBox`(int txBoxNo, LIN_TX_MSG txMsg)

[Parameters]

Variable	I/O	Description	Value
txBoxNo	IN	Transmit box No.	0~19 Up to 20 message types can be returned as responses.
txMsg	IN	LIN transmit message information	See the <u>LIN TX MSG members</u> below.

LIN TX MSG Members

```
public class LIN_TX_MSG{  
    public byte Id;  
    public byte Dlc;  
    public byte[] Data = new byte[8];  
    public LIN_SUM_TYPE SumType;  
}
```

Variable	Description	
Id	LIN ID	0~0x3F
Dlc	Data length	1~8
Data[0]~[7]	Data	0~0xFF
SumType	Checksum Type	- LIN_SUM_TYPE.CLASSIC: Classic checksum - LIN_SUM_TYPE.ENHANCED: Enhanced checksum

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.PARAMETER_ERROR	Parameter out of range.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.24. LinStartSlaveSimulation: Start LIN Slave Simulation

[Description]

Call this function after LinSetSlaveSendBox (update LIN slave simulation transmit box) to start LIN slave simulation (response). Specify whether transmit boxes 1 to 20 are enabled or disabled.

[Syntax]

CL_RETURN LinStartSlaveSimulation(bool[] txBoxNoEnable)

[Parameters]

Variable	I/O	Description	Value
txBoxNoEnable[0]~ [19]	IN	txBoxNoEnable[0]: Enable/disable transmit box 1 txBoxNoEnable[1]: Enable/disable transmit box 2 ... txBoxNoEnable[19]: Enable/disable transmit box 20	• false: Disabled • true: Enabled

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.LIN_NOT_STARTED	The LIN communication start function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.25. LinStopSimulation: Stop LIN Simulation

[Description]

Stops LIN simulation (master or slave).

[Syntax]

CL_RETURN LinStopSimulation()

[Parameters]

None

[Return Value]

Value	Description
CL_RETURN.OK	Success
CL_RETURN.DEVICE_NOT_OPENED	The open function has not been called.
CL_RETURN.DEVICE_WRITE_ERROR	Error sending a command to the CanLine-1 device.
CL_RETURN.OTHER_ERROR	Other error Note: Check the OtherErrorMsg property for the error message.

5.26. LinGetMessage: Get LIN Notification Messages

[Description]

Retrieves notification messages for completed LIN transmissions and received LIN messages.

[Syntax]

```
int LinGetMessage(ref List<LIN_RX_MSG> lstRxMsg, int count)
```

[Parameters]

Variable	I/O	Description	Value
lstRxMsg	OUT	List of LIN notification message information	See the <u>LIN_RX_MSG members</u> below.
count	IN	Number of messages to retrieve	Read the received message count using the <code>LinRxMsgCount</code> property and specify a value no greater than that count.

LIN_RX_MSG Members

```
public class LIN_RX_MSG{  
    public UInt32 TimeStamp;  
    public LIN_RX_MSG_TYPE Type;  
    public byte Id;  
    public byte Dlc;  
    public byte[] Data = new byte[8];  
    public LIN_SUM_TYPE SumType;  
    public LIN_RX_MSG_STATUS Status;  
}
```

Variable	Description	Value
TimeStamp	Time stamp	0~4294967295 Resolution: 0.1 ms Example: 123 = 12.3 ms
Type	Message Type	- LIN_RX_MSG_TYPE.RECIEVD: LIN message received - LIN_RX_MSG_TYPE.SENT: LIN transmission completed
Id	LIN ID	0~0x3F
Dlc	Data length	0~8
Data[0]~[7]	Data	0~0xFF
SumType	Checksum Type	- LIN_SUM_TYPE.CLASSIC: Classic checksum - LIN_SUM_TYPE.ENHANCED: Enhanced checksum

Status	Status	• LIN_RX_MSG_STATUS.SUCCESS: Success • LIN_RX_MSG_STATUS.TIME_OUT: Timeout error • LIN_RX_MSG_STATUS.SUM_ERROR: Checksum error • LIN_RX_MSG_STATUS.PARITY_ERROR: Parity error
--------	--------	--

[Return Value]

Value	Description
int	Number of messages retrieved