
PyOpenLCB Documentation

Release 0.2

Timothy C. Hatch, Dustin C. Hatch

February 01, 2015

1	API Reference	3
1.1	OpenLCB over CAN	3
1.2	Node Communication	9
2	Indices and tables	11
	Python Module Index	13

Contents:

API Reference

1.1 OpenLCB over CAN

1.1.1 OpenLCB Message Representations

```
class openlcb.can.messages.AliasMapDefinition (**keywords)
```

```
    MTI = 67329
```

```
class openlcb.can.messages.AliasMapEquiry (**keywords)
```

```
    MTI = 67330
```

```
class openlcb.can.messages.AliasMapRequest (**keywords)
```

```
    MTI = 67331
```

```
class openlcb.can.messages.CidMessage1 (**keywords)
```

```
    MTI = 23
```

```
class openlcb.can.messages.CidMessage2 (**keywords)
```

```
    MTI = 22
```

```
class openlcb.can.messages.CidMessage3 (**keywords)
```

```
    MTI = 21
```

```
class openlcb.can.messages.CidMessage4 (**keywords)
```

```
    MTI = 20
```

```
class openlcb.can.messages.ConsumerIdentifiedInvalid (**keywords)
```

```
    MTI = 103621
```

```
class openlcb.can.messages.ConsumerIdentifiedReserved (**keywords)
```

MTI = 103622

```
class openlcb.can.messages.ConsumerIdentifiedUnknown (**keywords)
```

MTI = 103623

```
class openlcb.can.messages.ConsumerIdentifiedValid (**keywords)
```

MTI = 103620

```
class openlcb.can.messages.ConsumerIdentifyRange (**keywords)
```

MTI = 103588

```
class openlcb.can.messages.DatagramConfiguration (**keywords)
```

BODY_DATA = 32

MTI = 26

```
class openlcb.can.messages.DatagramGeneral_1 (**keywords)
```

BODY_DATA = 542179328

MTI = 27

```
class openlcb.can.messages.DatagramGeneral_2 (**keywords)
```

BODY_DATA = 1592066112

MTI = 29

```
class openlcb.can.messages.DatagramIntermediate (**keywords)
```

MTI = 28

```
class openlcb.can.messages.DatagramLast (**keywords)
```

MTI = 29

```
class openlcb.can.messages.DatagramOutOfOrder (**keywords)
```

BODY_DATA = 5070848

MTI = 26

```
class openlcb.can.messages.DatagramReceived (**keywords)
```

MTI = 105000

```
class openlcb.can.messages.DatagramRejected (**keywords)
```

BODY_DATA = 77

MTI = 105032


```
class openlcb.can.messages.DatagramRejectedBufferFull (**keywords)
```

```
    BODY_DATA = 5054464
```

```
    MTI = 26
```

```
class openlcb.can.messages.DatagramTypeNotAccepted (**keywords)
```

```
    BODY_DATA = 178917440
```

```
    MTI = 105032
```

```
class openlcb.can.messages.GeneralDatagram (**keywords)
```

```
    MTI = 26
```

```
class openlcb.can.messages.IdentifyConsumers (**keywords)
```

```
    MTI = 104692
```

```
class openlcb.can.messages.IdentifyEventsAddressed (**keywords)
```

```
    MTI = 104808
```

```
class openlcb.can.messages.IdentifyEventsGlobal (**keywords)
```

```
    MTI = 104816
```

```
class openlcb.can.messages.IdentifyProducers (**keywords)
```

```
    MTI = 104724
```

```
class openlcb.can.messages.IdentifyProducersInvalid (**keywords)
```

```
    MTI = 103749
```

```
class openlcb.can.messages.IdentifyProducersUnknown (**keywords)
```

```
    MTI = 103751
```

```
class openlcb.can.messages.InitializationComplete (**keywords)
```

```
    MTI = 102656
```

```
class openlcb.can.messages.InteractionRejected (**keywords)
```

```
    MTI = 102504
```

```
class openlcb.can.messages.PIPMessageRecv (**keywords)
```

```
    MTI = 104040
```

```
class openlcb.can.messages.PIPMessageSend (**keywords)
```

MTI = 104488

```
class openlcb.can.messages.ProducerIdentified (**keywords)
```

MTI = 103748

```
class openlcb.can.messages.ProducerIdentifiedReserved (**keywords)
```

MTI = 103750

```
class openlcb.can.messages.ProducerIdentifyRange (**keywords)
```

MTI = 103716

```
class openlcb.can.messages.ReserveID (**keywords)
```

MTI = 67328

```
class openlcb.can.messages.ResetDatagram (**keywords)
```

BODY_DATA = 8361

MTI = 26

```
class openlcb.can.messages.SendProducerConsumerEvent (**keywords)
```

MTI = 103860

```
class openlcb.can.messages.SimpleNodeIdentInfo (**keywords)
```

MTI = 105960

```
class openlcb.can.messages.SimpleNodeIdentInfoAck (**keywords)
```

MTI = 104968

```
class openlcb.can.messages.StartDatagramFrame (**keywords)
```

MTI = 27

```
class openlcb.can.messages.VerifiedNodeAddressed (**keywords)
```

MTI = 103560

```
class openlcb.can.messages.VerifiedNodeIDNumber (**keywords)
```

MTI = 102768

```
class openlcb.can.messages.VerifyNodeIDNumberSimple (**keywords)
```

MTI = 103568

OpenLCB over CAN

Classes for creating, manipulating, and parsing CAN frames as OpenLCB messages.

author Dustin C. Hatch

author Timothy C. Hatch

class `openlcb.can.AddressedMessage` (***keywords*)

Represents a message sent to a single node

Parameters

- **src_alias** (*int*) – The alias of the node from which messages originate
- **dst_alias** (*int*) – The alias of the node to which the message is sent

src_alias

The node alias of the node sending the message

dst_alias

The alias of the node for which the message is intended

exception `openlcb.can.AlreadyRegistered` (*mti, body_data=None*)

Raised when more than one class attempt to use the same MTI

class `openlcb.can.CANMessage` (*header='', body=''*)

Base class for Controller Area Network messages

Calling `str()` on `CANMessage` instances returns a string containing the CAN frame.

header

The value of the header that would be sent in the frame

body

The value of the body that would be sent with the frame

classmethod `from_sequence` (*seq*)

Convert a sequence (list, etc.) of strings to Message objects

Parameters `seq` (*sequence*) – A sequence of OpenLCB frame strings

Yields Instances of the `CANMessage` subclass

classmethod `from_string` (*frame*)

Create a `CANMessage` instance from a frame string

Parameters `frame` (*str*) – The complete CAN frame, including control characters

Returns A new instance of the `CANMessage` subclass

Subclasses of `CANMessage` should *not* override this method, but rather `parse_frame()` instead.

classmethod `parse_frame` (*frame*)

Parse a string containing a CAN frame into its parts

Returns dict A dictionary containing the valuable message parts

Subclasses of `CANMessage` should override this method and provide their own unique logic for parsing the header and body into usable properties.

class `openlcb.can.DatagramData` (***keywords*)

Represents a message that contains a full node ID

node_id

The full of the ID of the node identified in the message

class `openlcb.can.EventMessage` (***keywords*)

Represents a message containing an event ID

event_id

The ID of the event identified in the message

According to the [Event Identifiers Technical Note](#), a node ID is part of the event ID, however it does not have to have any relationship to the event (i.e. it does not have to refer to the node sending the message).

class `openlcb.can.GlobalMessage` (***keywords*)

Represents a message sent to the entire bus

exception `openlcb.can.IncorrectMTI` (*mti, cls*)

Raised when creating a message from a string with the wrong MTI

exception `openlcb.can.InvalidMessage` (*frame*)

Raised when attempting to parse an improperly-formatted frame

class `openlcb.can.NodeAlias` (*alias*)

Utility class for handling node alias

- Calling `int` on an instance of `NodeAlias` will return the integer value.

- Calling `str` on an instance of `NodeAlias` will return a hexadecimal number.

class `openlcb.can.NodeIDMessage` (***keywords*)

Represents a message that contains a full node ID

node_id

The full of the ID of the node identified in the message

class `openlcb.can.OpenLCBMessage` (***keywords*)

Base class for all OpenLCB CAN messages

Parameters `src_alias` (*int*) – The alias of the node from which messages originate

src_alias

The node alias of the node sending the message

MTI

The message type indicator, as an integer. This value should be set by subclasses of `CANMessage`. See <http://www.openlcb.org/trunk/specs/MtiAllocations.pdf> for a list of MTI allocations.

class `openlcb.can.RegisteredMessage`

Metaclass for OpenLCB message classes

Classes using `RegisteredMessage` as their metaclass will automatically have their MTIs registered, which creates a reverse mapping from MTI to class. This registration is required for `parse_frame()` to discover a message class based on the MTI and return an instance of it.

`openlcb.can.parse_frame` (*frame*)

Parse an OpenLCB Message from a CAN frame

Parameters `frame` (*str*) – CAN frame as a string, including control characters

Returns An instance of the class registered for the MTI specified in the frame

The message type lookup proceeds as follows:

- 1.The MTI of the message is determined by shifting the header to the left by 12 bits
- 2.If the message contains a body, the first byte (2 characters) is taken as the body MTI
- 3.The list of registered message classes is searched for a class with the same MTI and no body MTI
- 4.If no class is found, the list of registered message classes is searched again for the same MTI, this time including the body MTI
- 5.If no class is found, a new MTI is calculated by bit shifting the MTI 12 places to the right

6. The list of registered message classes is searched for a class with the recalculated MTI and the body MTI

7. If a class is still not found, a new one is created. The new class is called `UnknownMessage` and inherits directly from `OpenLCBMessage`. Its MTI is set to the original MTI (i.e. before the second bit shift), and its body MTI is set, if it exists.

Once a suitable message class has been found (or created), it will be instantiated by calling its `from_string` classmethod, passing in the original message frame string.

1.2 Node Communication

exception `openlcb.communication.CommunicationException`

Raised if an error occurs while communicating with a node

class `openlcb.communication.EthernetConnection(hostname, port)`

Class for communicating with nodes via Eth2CAN

Parameters

- **hostname** (*str*) – Host name or IP address of the Eth2CAN device
- **port** (*int*) – TCP port of the Eth2CAN device

close()

Close the TCP/IP communication socket

connect()

Connect to the Eth2CAN device over TCP/IP

receive_multi(timeout=None)

Retrieve multiple responses from the OpenLCB bus

send(message)

Send an OpenLCB message over TCP/IP

class `openlcb.communication.OpenLCBConnection`

Base class for OpenLCB communication mechanisms

Subclasses should define the following methods for communicating with an OpenLCB bus over a given transport:

- `connect()`
- `send()`
- `receive_multi()`
- `close()`

Instances are also *context managers*. This facilitates opening and closing of connections using the `with` statement:

```
conn = EthernetConnection(hostname)
with conn:
    conn.send(message)
```

The above example will automatically open and close the connection upon entering and leaving the context.

close()

Close the connection to the OpenLCB bus

connect()

Connect to the OpenLCB bus

receive()

Retreive a response from the OpenLCB bus

Returns str A string containing the CAN frame received from the bus, suitable for creating a new instance of a `CANMessage` subclass

Deprecated since version 0.1: This method is ambiguous; use `receive_one()` instead.

receive_multi()

Retreive multiple responses from the OpenLCB bus

Returns list A list of strings containing CAN frames, suitable for creating new instances of a subclass of `CANMessage`. If no response was returned within the timeout period, an empty list (i.e. `[]`) is returned.

receive_one()

Retreive a single response message from the node

Returns str A string containing the first message received, suitable for creating a new instance of a `CANMessage` subclass. If no response was received within the timeout period, `None` is returned.

Note: Calling this method will actually empty the buffer, meaning that if more than one message was received, any subsequent messages will be discarded. If multiple response messages are expected, use `receive_multi()` instead.

send(message)

Send a message to the OpenLCB bus

Parameters message (str) – The CAN frame to send

Any object that can be converted to a string can be passed as the `message` parameter, such as `openlcb.can.CANMessage` subclasses.

class openlcb.communication.SerialConnection(com_port, speed)

Class for communicating with nodes via Serial Connection

Parameters

- **com_port (str)** – Serial port device name or port number
- **speed (int)** – Serial port Baud rate (defaults to 500000)

close()

Close the serial port and terminate the OpenLCB connection

connect()

Open the serial port and set RTS

receive_multi(timeout=None)

Retreive multiple responses from the OpenLCB bus

send(frame)

Send an OpenLCB message through the serial port

Indices and tables

- *genindex*
- *modindex*
- *search*

O

`openlcb.can`, 6

`openlcb.can.messages`, 3

`openlcb.communication`, 9

A

AddressedMessage (class in openlcb.can), 7
 AliasMapDefinition (class in openlcb.can.messages), 3
 AliasMapEquiry (class in openlcb.can.messages), 3
 AliasMapRequest (class in openlcb.can.messages), 3
 AlreadyRegistered, 7

B

body (openlcb.can.CANMessage attribute), 7
 BODY_DATA (openlcb.can.messages.DatagramConfiguration attribute), 4
 BODY_DATA (openlcb.can.messages.DatagramGeneral_1 attribute), 4
 BODY_DATA (openlcb.can.messages.DatagramGeneral_2 attribute), 4
 BODY_DATA (openlcb.can.messages.DatagramOutOfOrder attribute), 4
 BODY_DATA (openlcb.can.messages.DatagramRejected attribute), 4
 BODY_DATA (openlcb.can.messages.DatagramRejectedBufferFull attribute), 5
 BODY_DATA (openlcb.can.messages.DatagramTypeNotAccepted attribute), 5
 BODY_DATA (openlcb.can.messages.ResetDatagram attribute), 6

C

CANMessage (class in openlcb.can), 7
 CidMessage1 (class in openlcb.can.messages), 3
 CidMessage2 (class in openlcb.can.messages), 3
 CidMessage3 (class in openlcb.can.messages), 3
 CidMessage4 (class in openlcb.can.messages), 3
 close() (openlcb.communication.EthernetConnection method), 9
 close() (openlcb.communication.OpenLCBConnection method), 9
 close() (openlcb.communication.SerialConnection method), 10
 CommunicationException, 9

connect() (openlcb.communication.EthernetConnection method), 9
 connect() (openlcb.communication.OpenLCBConnection method), 9
 connect() (openlcb.communication.SerialConnection method), 10
 ConsumerIdentifiedInvalid (class in openlcb.can.messages), 3
 ConsumerIdentifiedReserved (class in openlcb.can.messages), 3
 ConsumerIdentifiedUnknown (class in openlcb.can.messages), 4
 ConsumerIdentifiedValid (class in openlcb.can.messages), 4
 ConsumerIdentifyRange (class in openlcb.can.messages), 4

D

DatagramConfiguration (class in openlcb.can.messages), 4
 DatagramData (class in openlcb.can), 7
 DatagramGeneral_1 (class in openlcb.can.messages), 4
 DatagramGeneral_2 (class in openlcb.can.messages), 4
 DatagramIntermediate (class in openlcb.can.messages), 4
 DatagramLast (class in openlcb.can.messages), 4
 DatagramOutOfOrder (class in openlcb.can.messages), 4
 DatagramReceived (class in openlcb.can.messages), 4
 DatagramRejected (class in openlcb.can.messages), 4
 DatagramRejectedBufferFull (class in openlcb.can.messages), 4
 DatagramTypeNotAccepted (class in openlcb.can.messages), 5
 dst_alias (openlcb.can.AddressedMessage attribute), 7

E

EthernetConnection (class in openlcb.communication), 9
 event_id (openlcb.can.EventMessage attribute), 7
 EventMessage (class in openlcb.can), 7

F

from_sequence() (openlcb.can.CANMessage class

method), 7
from_string() (openlcb.can.CANMessage class method), 7

G

GeneralDatagram (class in openlcb.can.messages), 5
GlobalMessage (class in openlcb.can), 8

H

header (openlcb.can.CANMessage attribute), 7

I

IdentifyConsumers (class in openlcb.can.messages), 5
IdentifyEventsAddressed (class in openlcb.can.messages), 5
IdentifyEventsGlobal (class in openlcb.can.messages), 5
IdentifyProducers (class in openlcb.can.messages), 5
IdentifyProducersInvalid (class in openlcb.can.messages), 5
IdentifyProducersUnknown (class in openlcb.can.messages), 5
IncorrectMTI, 8
InitializationComplete (class in openlcb.can.messages), 5
InteractionRejected (class in openlcb.can.messages), 5
InvalidMessage, 8

M

MTI (openlcb.can.messages.AliasMapDefinition attribute), 3
MTI (openlcb.can.messages.AliasMapEquiry attribute), 3
MTI (openlcb.can.messages.AliasMapRequest attribute), 3
MTI (openlcb.can.messages.CidMessage1 attribute), 3
MTI (openlcb.can.messages.CidMessage2 attribute), 3
MTI (openlcb.can.messages.CidMessage3 attribute), 3
MTI (openlcb.can.messages.CidMessage4 attribute), 3
MTI (openlcb.can.messages.ConsumerIdentifiedInvalid attribute), 3
MTI (openlcb.can.messages.ConsumerIdentifiedReserved attribute), 3
MTI (openlcb.can.messages.ConsumerIdentifiedUnknown attribute), 4
MTI (openlcb.can.messages.ConsumerIdentifiedValid attribute), 4
MTI (openlcb.can.messages.ConsumerIdentifyRange attribute), 4
MTI (openlcb.can.messages.DatagramConfiguration attribute), 4
MTI (openlcb.can.messages.DatagramGeneral_1 attribute), 4
MTI (openlcb.can.messages.DatagramGeneral_2 attribute), 4
MTI (openlcb.can.messages.DatagramIntermediate attribute), 4

MTI (openlcb.can.messages.DatagramLast attribute), 4
MTI (openlcb.can.messages.DatagramOutOfOrder attribute), 4
MTI (openlcb.can.messages.DatagramReceived attribute), 4
MTI (openlcb.can.messages.DatagramRejected attribute), 4
MTI (openlcb.can.messages.DatagramRejectedBufferFull attribute), 5
MTI (openlcb.can.messages.DatagramTypeNotAccepted attribute), 5
MTI (openlcb.can.messages.GeneralDatagram attribute), 5
MTI (openlcb.can.messages.IdentifyConsumers attribute), 5
MTI (openlcb.can.messages.IdentifyEventsAddressed attribute), 5
MTI (openlcb.can.messages.IdentifyEventsGlobal attribute), 5
MTI (openlcb.can.messages.IdentifyProducers attribute), 5
MTI (openlcb.can.messages.IdentifyProducersInvalid attribute), 5
MTI (openlcb.can.messages.IdentifyProducersUnknown attribute), 5
MTI (openlcb.can.messages.InitializationComplete attribute), 5
MTI (openlcb.can.messages.InteractionRejected attribute), 5
MTI (openlcb.can.messages.PIPMessageRecv attribute), 5
MTI (openlcb.can.messages.PIPMessageSend attribute), 5
MTI (openlcb.can.messages.ProducerIdentified attribute), 6
MTI (openlcb.can.messages.ProducerIdentifiedReserved attribute), 6
MTI (openlcb.can.messages.ProducerIdentifyRange attribute), 6
MTI (openlcb.can.messages.ReserveID attribute), 6
MTI (openlcb.can.messages.ResetDatagram attribute), 6
MTI (openlcb.can.messages.SendProducerConsumerEvent attribute), 6
MTI (openlcb.can.messages.SimpleNodeIdentInfo attribute), 6
MTI (openlcb.can.messages.SimpleNodeIdentInfoAck attribute), 6
MTI (openlcb.can.messages.StartDatagramFrame attribute), 6
MTI (openlcb.can.messages.VerifiedNodeAddressed attribute), 6
MTI (openlcb.can.messages.VerifiedNodeIDNumber attribute), 6
MTI (openlcb.can.messages.VerifyNodeIDNumberSimple

attribute), 6
 MTI (openlcb.can.OpenLCBMessage attribute), 8

N

node_id (openlcb.can.DatagramData attribute), 7
 node_id (openlcb.can.NodeIDMessage attribute), 8
 NodeAlias (class in openlcb.can), 8
 NodeIDMessage (class in openlcb.can), 8

O

openlcb.can (module), 6
 openlcb.can.messages (module), 3
 openlcb.communication (module), 9
 OpenLCBConnection (class in openlcb.communication), 9
 OpenLCBMessage (class in openlcb.can), 8

P

parse_frame() (in module openlcb.can), 8
 parse_frame() (openlcb.can.CANMessage class method), 7
 PIPMessageRecv (class in openlcb.can.messages), 5
 PIPMessageSend (class in openlcb.can.messages), 5
 ProducerIdentified (class in openlcb.can.messages), 6
 ProducerIdentifiedReserved (class in openlcb.can.messages), 6
 ProducerIdentifyRange (class in openlcb.can.messages), 6

R

receive() (openlcb.communication.OpenLCBConnection method), 10
 receive_multi() (openlcb.communication.EthernetConnection method), 9
 receive_multi() (openlcb.communication.OpenLCBConnection method), 10
 receive_multi() (openlcb.communication.SerialConnection method), 10
 receive_one() (openlcb.communication.OpenLCBConnection method), 10
 RegisteredMessage (class in openlcb.can), 8
 ReserveID (class in openlcb.can.messages), 6
 ResetDatagram (class in openlcb.can.messages), 6

S

send() (openlcb.communication.EthernetConnection method), 9
 send() (openlcb.communication.OpenLCBConnection method), 10
 send() (openlcb.communication.SerialConnection method), 10
 SendProducerConsumerEvent (class in openlcb.can.messages), 6

SerialConnection (class in openlcb.communication), 10
 SimpleNodeIdentInfo (class in openlcb.can.messages), 6
 SimpleNodeIdentInfoAck (class in openlcb.can.messages), 6
 src_alias (openlcb.can.AddressedMessage attribute), 7
 src_alias (openlcb.can.OpenLCBMessage attribute), 8
 StartDatagramFrame (class in openlcb.can.messages), 6

V

VerifiedNodeAddressed (class in openlcb.can.messages), 6
 VerifiedNodeIDNumber (class in openlcb.can.messages), 6
 VerifyNodeIDNumberSimple (class in openlcb.can.messages), 6