

**Innomatix eMbed CAN Interface**

**API Specification**

|  |  |
| --- | --- |
| Revision | 1.1 |
| Revision Date | 6-Sep-2017 |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Change** |
| 1.0 | 8-Jun-2015 | JLASH | Initial Creation |
| 1.1 | 6-Sep-17 | JLASH | Add CanOpen() |
|  |  |  |  |

Table of Contents

1.0 Overview 2

2.0 Innomatix eMbed CAN API 2

2.1. Init 2

2.2. Open 2

2.3. Close 3

2.4. Send Message 3

2.5. Receive Message 3

2.6. Get Statistics 3

2.7. API Return Values 4

2.1. CAN Channel Enum 4

2.2. CAN Speed Enum 4

2.3. CAN Format Enum 5

1. Overview

This document defines the Innomatix Coprocessor CAN API. The purpose of this functionality is to provide a robust, fast and simple interface for using the CAN networks of a DAP Coprocessor.

Other CAN API’s may exist for a Coprocessor – either provided by the development tools or by third-parties – and can be used if more flexibility or control is desired.

1. Innomatix eMbed CAN API

The API consists of the following public functions:

* 1. Init

Initializes the CAN library and provides two CAN objects created by the application.

results\_e CanInit( CAN \*can0, CAN \*can1 )

Return Values:

* Success – the API was initialized and objects were saved

The ‘can0’ object is associated with the *eCanChDAP3* channel enumeration. The ‘can1’ object is associated with the *eCanChDAP4* channel enumeration.

* 1. Open

This function opens the designated CAN channel at the specified baud rate.

results\_e CanInit( CanChannel\_e Channel, int Speed )

Return Values:

* Success – the API was initialized and the channel is ready to use.
* InvalidChannel – the specified channel is not supported
* InvalidSpeed – the specified network speed is not supported
* GeneralError – an unknown or unspecified error has occurred

The Speed parameter is the desired bits per second. Common values are:

* 125000 – low speed bus
* 250000 – medium speed bus
* 500000 – high speed bus
  1. Close

This function disables the designated CAN interface.

void CanClose(CanChannel \_e Channel )

* 1. Send Message

Allow the client to send a message on the CAN bus.

results\_e CanSend(CanChannel \_e Channel, int Id,

CanFormat\_e Format, unsigned char \*Data, int DLen )

Return Values:

* Success – the message was sent
* InvalidChannel – the specified channel is not supported
* InvalidIdentifier – the provided message id is not a valid for the specified format
* InvalidLength – more than 8 bytes of data were indicated
* GeneralError – an unknown or unspecified error has occurred
  1. Receive Message

Allow the client to get a message from the CAN bus.

results\_e CanReceive(CanChannel \_e Channel, int \*Id,

CanFormat\_e \*Format, unsigned char \*Data, int \*DLen )

Return Values:

* Success – a message was retrieved, the parameters hold the message values.
* InvalidChannel – the specified channel is not supported
* NoData – no message is available on the requested channel
* GeneralError – an unknown or unspecified error has occurred

The client should always check the results before using the returned values.

The callers values are unchanged if the result is not Success.

Callers buffer is assumed to be at least 8 bytes

* 1. Get Statistics

Allow the client to retrieve statistics about the CAN interface.

results\_e CanStatistics(CanChannel \_e Channel,

StatisticsStruct\_t \*Stats )

Return Values:

* Success – statistics were retrieved
* InvalidChannel – the specified channel is not supported
* GeneralError – an unknown or unspecified error has occurred

The Statistics structure contains the following fields:

typedef struct

{

int TotalTxMessages;

int TotalRxMessages;

int OverflowCount;

}StatisticsStruct\_t;

* 1. API Return Values

|  |
| --- |
| Name |
| eSuccess |
| eInvalidChannel |
| eInvalidSpeed |
| eInvalidIdentifier |
| eInvalidLength |
| eNoData |
| eGeneralError |

* 1. CAN Channel Enum

|  |  |
| --- | --- |
| Name | Purpose |
| eCanChDAP3 | CAN 3 interface on the DAP connector or wiring harness |
| eCanChDAP4 | CAN 4 interface on the DAP connector or wiring harness |

* 1. CAN Speed Enum

|  |  |
| --- | --- |
| Name | Purpose |
| eCANSpeed125k | 125k bits |
| eCANSpeed250k | 250k bits |
| eCANSpeed500k | 500k bits |

* 1. CAN Format Enum

|  |  |
| --- | --- |
| Name | Purpose |
| eCanFormatStandard | A standard, 11-bit identifier |
| eCanFormatExtended | An extended, 29-bit identifier |