Inspiring Motion
Since 1988

Release Notes for Maestro .NET Library 2.1.1.5

Version 2.1.1.5

February 2016 (Ver. 1.000) www.elmomc.com

Elmo
Motion Control

Notice

This guide is delivered subject to the following conditions and restrictions:

- This guide contains proprietary information belonging to Elmo Motion Control Ltd. Such information is supplied solely for the purpose of assisting users of the Gold Line technology.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Information in this document is subject to change without notice.

Elmo Motion Control and the Elmo Motion Control logo are registered trademarks of Elmo Motion Control Ltd.

EtherCAT Conformance Tested. EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

CANopen compliant. CANopen® is a registered trademark and patented technology, licensed by CAN in Automation (CiA) GmbH, Kontumazgarten 3, DE-90429 Nuremberg, Germany.

Document no. MAN-EASII-RN (Ver. 1.000)

Revision History

Version	Date	Details
Ver. 1.000	February 2016	Initial version

Chapter 1: General

The following release notes highlight the features of the current Maestro .NET library release (version 2.1.1.5).

This library provides a C# based API for connecting with the Maestro devices and activating those devices.

This document contains general information and it doesn't replace the library documentation and the example applications also available with the library or on the Elmo web site.

This release is for the use of all customers.

Chapter 2: Communicating with the Maestro

Means of Communication to the Maestro

The Maestro .NET library allows you to communicate with the Maestro devices using a single communication methods supported by the Maestro, which is TCP/IP.

Once connected to the Maestro, you activate it by calling the various Function Block (FB) methods on various communication objects using a connection handle.

Each FB methods returns a success / failure status as well as an error code used to analyze the error received.

The Maestro also sends events using the UDP protocol. The library can receive those events from the Maestro and pass them to the client application.

Communication Object

The class used for connecting to the Maestro is called `MMCConnection`. The connection is established by calling a static method on this class called `ConnectRPC`, which returns a numeric number used as the connection handle for all future calls to this Maestro.

Library Object Hierarchy

The FBs calling the Maestro can affect the Maestro itself, and / or the devices located under the Maestro network.

The FBs are located in a class hierarchy that represent the affected device/s. This FB and class hierarchy is built accordingly:

- General FBs – affect the Maestro itself and/or all the devices in its network. Located in the `MMCConnection` class
- FBs affecting a single axis (drive) – located in the `MMCSingleAxis` class
- FBs affecting a group (vector) of axes in the Maestro – located in the `MMCGroupAxis` class
- FBs affecting a single node (drive or a general device type such as I/O etc.) – located in the `MMCNode` class
- FBs affecting either a single axis or a group of axes – located in the `MMCAxis` class

The `MMCAxis` class is inherited by the classes `MMCNode` and `MMCGroupAxis`. `MMCNode` is inherited by the class `MMCSingleAxis`.

Apart from the above classes, there are specific classes which hold the API to specific subjects such as the `MMCErrorCorr` class which holds the error correction API.

Receiving Events from the Maestro

The `ConnectRPC` method contains a parameter called `callbackFunc`, and another parameter called `eventMask`.

The `eventMask` parameter is a mask defining which events the Maestro will send to this connection. If you wish to receive all the possible events, set this mask to the value `0xEFFFFFFF`.

The `callbackFunc` parameter is a delegate method which is called whenever an event is received from the Maestro.

When an event is arrived from the Maestro, the library will receive this event and call the `callbackFunc` method with a struct named `MMC_CAN_REPLY_DATA_OUT`. This struct contains the entire event data, including the event type which is described by the `eASYNC_EVENT` enum.

Chapter 3: Changes from Version 2.1.1.4

General

The Maestro .NET library version 2.1.1.4 was released in January 2015.

Since then the Maestro wasn't changed dramatically.

Function Block Notification Event

The library now include a specific event for function block notification.

When the user uses the "InsertNotificationFb" API function a notification function block will be added to the queue, when the queue reach the notification function block the event will be called.