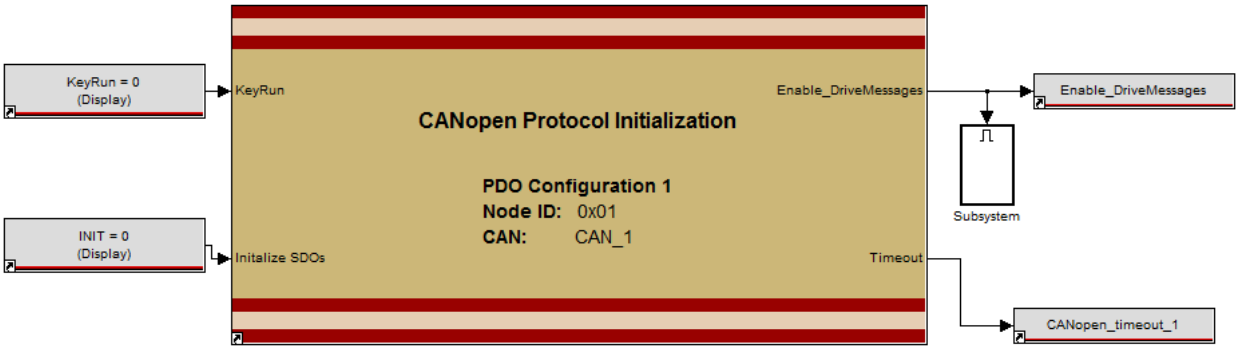
New Eagle™ CANopen MotoHawk™ Library

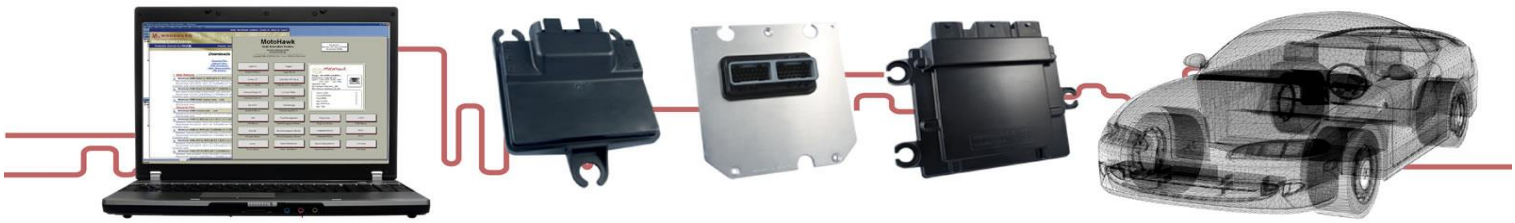
NE-MH-CANOPEN-LIB-001

User Guide

Revision 001

7/1/2015





New Eagle 3588 Plymouth Rd. #272 Ann Arbor, MI 48105-2603 877.234.1410 (voice) 270.964.8706 (fax)

CONTENTS

PRODUCT DESCRIPTION

OVERVIEW

The New Eagle™ CANopen Library is the perfect toolbox to add the CANopen protocol to any Motohawk project. This library interface allows for complete customization of the CANopen initialization sequence and control of up to 64 slave nodes. The library enables you to quickly establish the communication protocol for a multi-node CANopen network in a familiar Motohawk environment, reducing development and complexity.

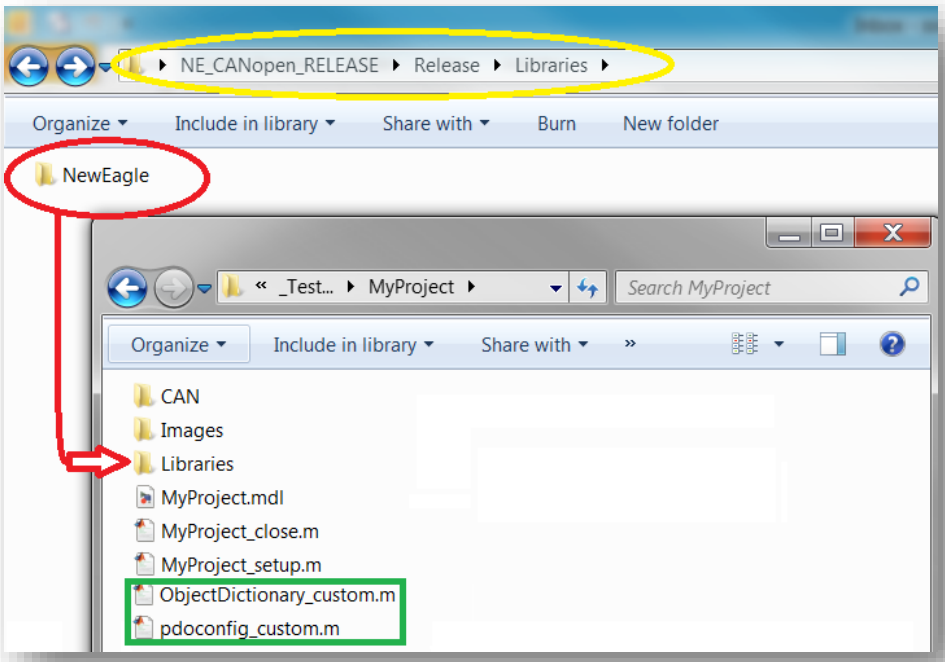
PRIMARY SOFTWARE FUNCTIONS

Configuration of the library is narrowed to a PDO data definition file. From that file the library internalizes and manages the following protocols:

- CANopen Protocol Initialization
  - Network Management (NMT) protocol (*master*)
    - Module control protocol
    - Heartbeat protocol
  - Service Data Object (SDO) Protocol
    - Custom object dictionary definitions
- Process Data Object (PDO) Protocol
  - TPDOs 1-4 (Transmit PDOs) Customization
  - RPDOs 1-4 (Receive PDOs) Customization
- Synchronization Object (SYNC) Protocol
  - Configurable Sync-Producer

INSTALLATION

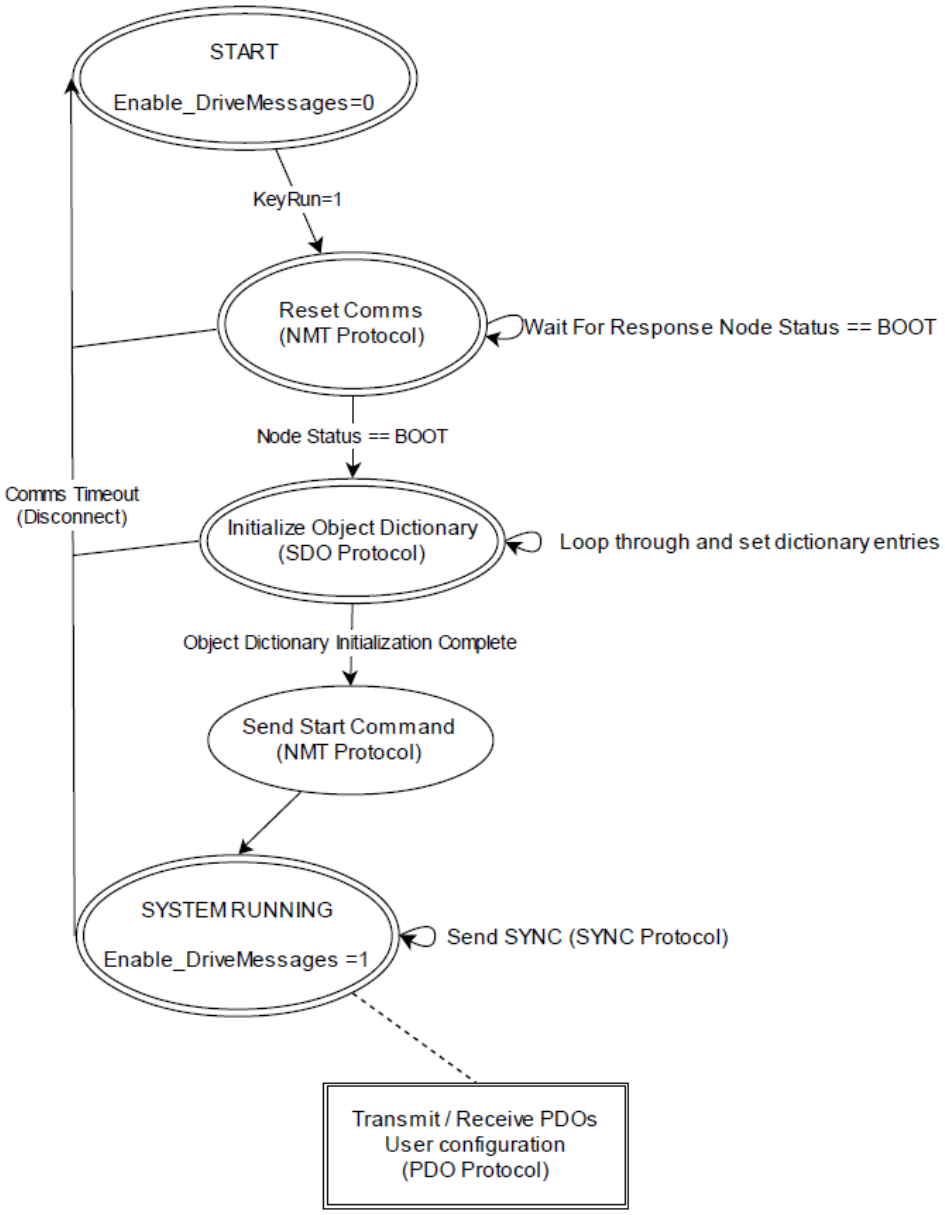
To install the CANopen Library, copy the CANopenLib directory into your Libraries directory under the directory in which your project resides. In order for MATLAB to find the CANopen libraries, make sure your working path is set to your model's directory. You can use the Initialization block and the CANopen protocol blocks the library provides by browsing to 'CANopen Library' in the Simulink Library Browser.



USAGE

CANOPEN PROTOCOL INITIALIZATION FLOWCHART

Understanding the entire initialization process isn't necessary to use this library, but understanding the general flow may be beneficial if issues arise.

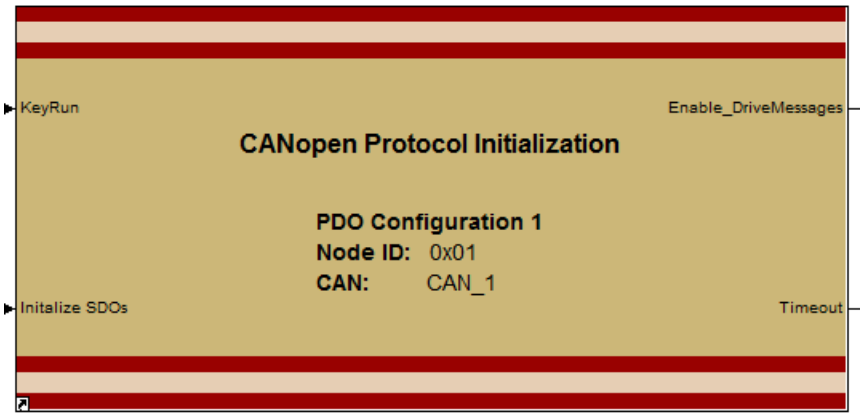


Above, a simplified, but accurate, flowchart depicts the states of the initialization block sequence. It begins in the START state, and after the initialization process is complete, it enters the SYSTEM_RUNNING state. This is indicated by the output state Enable_DriveMessages == 1, and at this point the user process should take control to send and react to PDO messages.

PDO Configuration

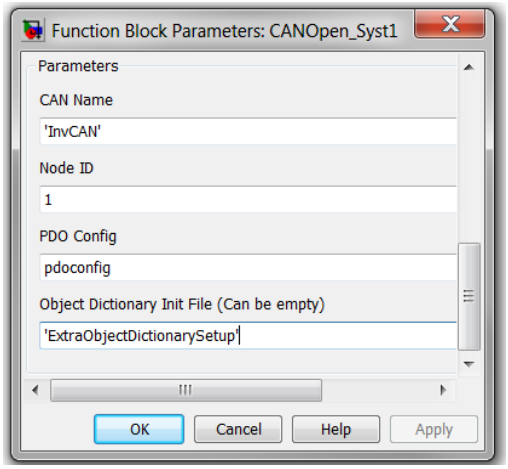
a. The PDOs need to be changed from the default PDO configuration, you will need to update the corresponding PDO file. See 'pdoconfig_example.m' for details on customizing the PDO configuration.
b. Note: If there are multiple CANopen nodes with different PDO configurations, you will need a PDO m-file for each unique configuration.

Initialization Block

a. This block is critical to the CANopen Protocol. It handles the startup sequence to initialize the CANopen node, as well as logic to re-initialize should the node disconnect or restart.
b. 

c. Block Inputs/Outputs:
    i. Input - Key Run
        1. The first time this is enabled, it will attempt to connect and initialize the drive. After the first initialization, enabling/disabling this will "pause" the system.
    ii. Input - Initialize SDOs:
        1. While the system is running it may be necessary to re-initialize a node. A rising edge on this input will trigger a re-initialization.
    iii. Output - Enable_DriveMessages:
        1. When this output is high, the node is capable of receiving and transmitting messages.
    iv. Output – Timeout
        1. Indicates the initialization sequence has timed out.

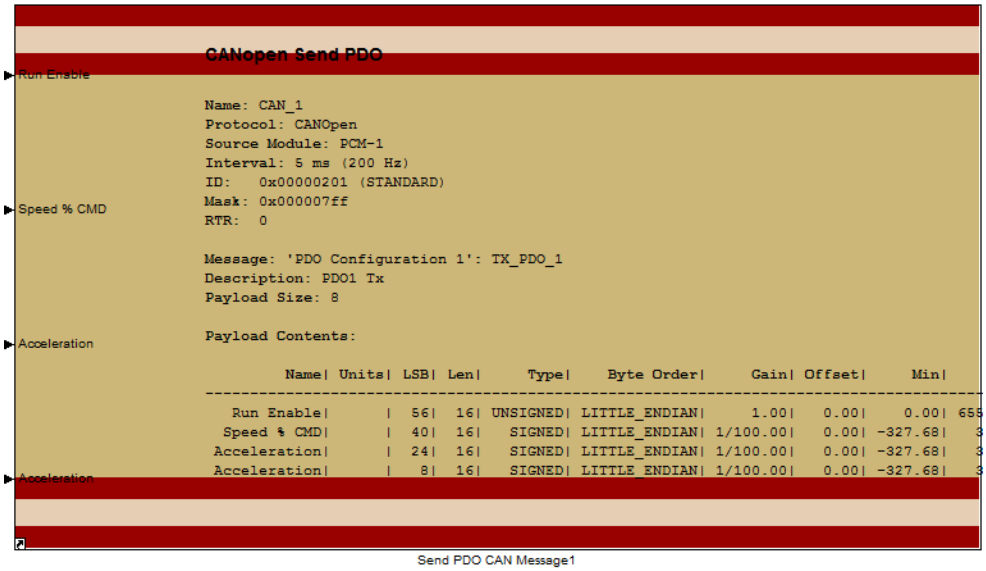
d. Block Configuration/Parameters:

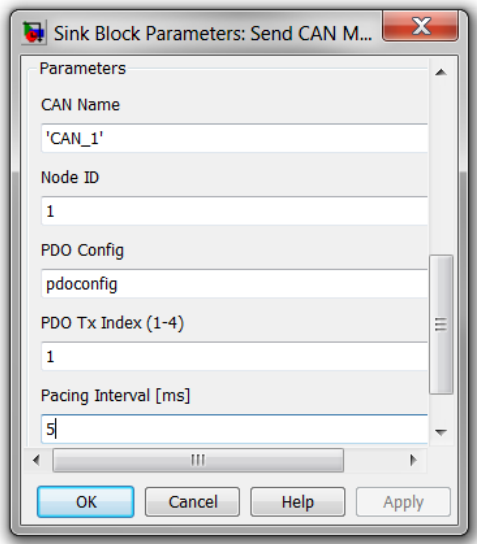
e. 

v. **CAN Name**: Specify the Motohawk CAN name, default is likely to be 'CAN_1'
vi. **Node ID**: The ID of the node you would like to communicate with (1-64).
vii. **PDO Config**: The m-file that will be used during the PDO initialization sequence.
viii. **Object Dictionary Init File**:
   1. This is an <u>optional</u> parameter.
   2. Use this if you need custom Object Dictionary entries (do not specify PDO configurations here). <u>Note</u>: The parameter value is expecting the name of an m-file without the extension (.m) surrounded by single quotes, e.g. 'ObjectDictionarySetup'.

TRANSMIT PDO

a. Based off of the MotoHawk CAN Send block, this block transmits PDO messages.

b. 

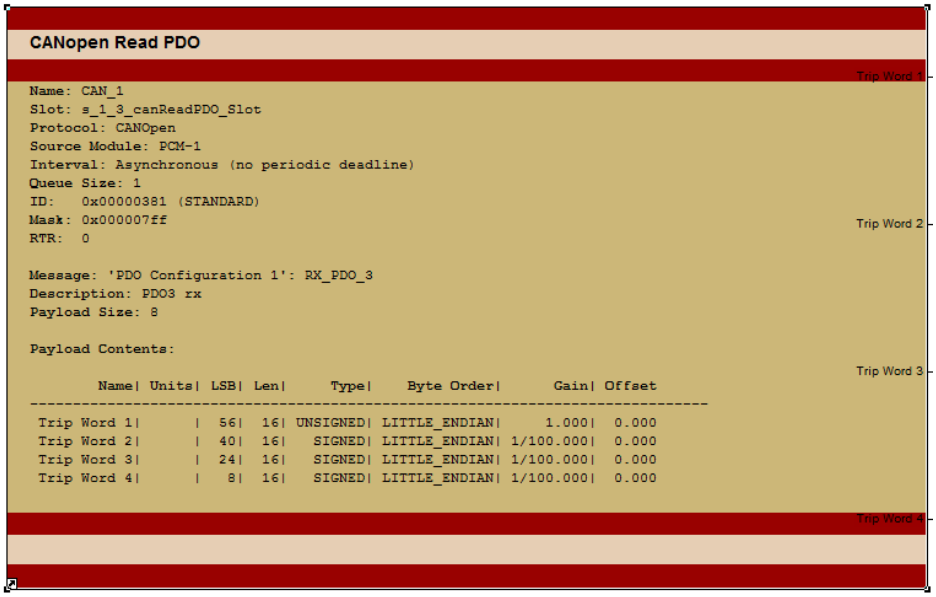
c. 

d. Block Parameter Configuration:

   ix. **CAN Name**: The MotoHawk CAN bus on which to transmit the PDO message.
   x. **Node ID**: The node ID of which the PDO message will be addressed (1-64).
   xi. **PDO Config**: The m-file that holds all of the PDO definitions.
   xii. **PDO Tx Index** (1-4): The index of the TX PDO that will be transmitted, as it corresponds to the PDO Config file.
   xiii. **Pacing**: Rate at which to send the PDO message in ms.

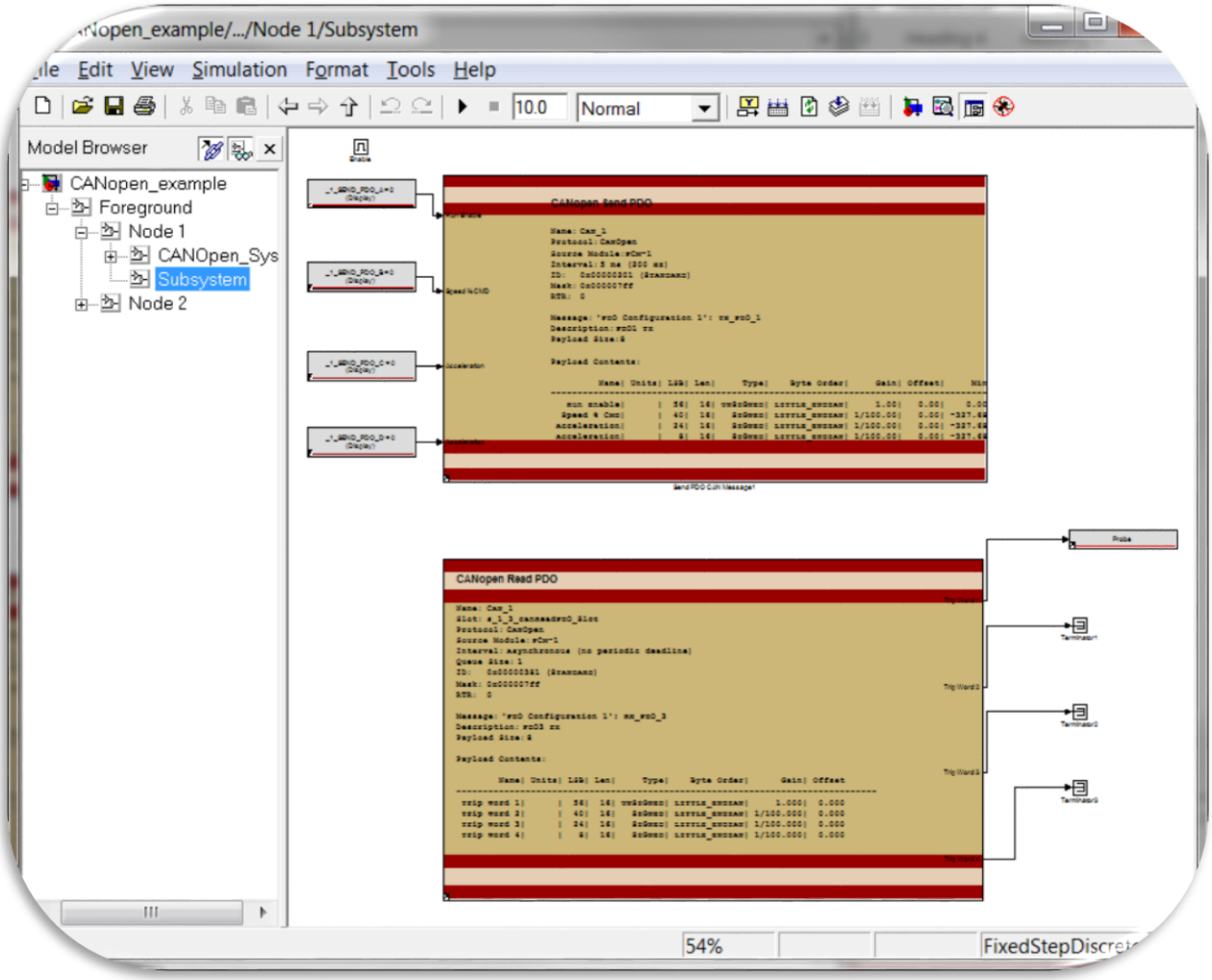
RECEIVE PDO

a. This receives PDO messages from another CANopen node (references config.rx in pdo config file)

b. 

EXAMPLE

An example model is included with the CANopen Library, 'CANopen_example.mdl'. This model should be sufficient to get you started. The included pdo configuration m-file contains additional help text and usage recommendations.



SUPPORT

INCLUDED SUPPORT

Eight (8) hours of general support is included with the purchase of the CANopen Library. Submit your inquiries to support@neweagle.net.

POTENTIAL FUNCTIONALITY OPTIONS

With an additional contract, New Eagle can create custom software to meet your CANopen application's specific needs. This includes (but is not limited to):

- Customization of the CANopen library within your Motohawk project to meet a specific CANopen implementation
- Functioning as a slave node within the Network Management (NMT) protocol
- Compatibility with non-traditional CANopen implementations

APPENDIX A: ADDITIONAL DOCUMENTATION AND REFERENCES

CANopen protocol specification via CAN-in-Automation: http://www.can-cia.org/?canopen

New Eagle's Wiki article on this CANopen library: link

pdo_config_example.m: Included in the library release, it contains additional information on how to customize the PDO configuration for your application.

PRODUCT DISCLAIMER

This product is intended to be used by controls engineers familiar with the MotoHawk® and Simulink® model-based development environment. Please contact New Eagle for any necessary application assistance or product training.

PRODUCT COMPATIBILITY

The library is compatible with recent versions of Motohawk and MATLAB 32/64bit. This library has been tested with the following versions: MATLAB 2009b 32-bit, MATLAB 2010a 64-bit, and MATLAB 2012a 64-bit.

If your version is not listed please contact New Eagle support (support@neweagle.net) to check for compatibility.