openCONFIGURATOR

Quick Start Guide

1.4.1

Prepared by
Kalycito Infotech Pvt. Ltd.,
India

License

Registered Office

Kalycito Infotech Private Limited,
E-LAB, Science and Technology Entrepreneurial Park I,
PSG College of Technology, Avinashi Road,
Coimbatore – 641004, Tamil Nadu, INDIA
Phone: 00 91 422 4518454
Email: info@kalycito.com

Abbreviations

API	Application Process Interface
CAN	Controller Area Network
CDC	Concise Device Configuration
CiA	CAN in Automation
CN	POWERLINK Controlled Node (slave)
DLL	Dynamic Link Library
EPL	Ethernet POWERLINK
EPSG	Ethernet POWERLINK Standardization Group
GUI	Graphical User Interface
ID	Identifier
IEC	International Electro-technical Commission
MN	Managing node
MNOBD	Object Dictionary of the Managing Node
NMT	Network Management
PDO	Process Data Objects
PReq	Poll Request (POWERLINK frame type)
PRes	Poll Response (POWERLINK frame type)
RPDO	Receive Process Data Object
SWIG	Simplified Wrapper and Interface Generator
TCL	Tool Command Language
TPDO	Transmit Process Data Object
XAP	Extend Application Process variables
XDC	XML Device Configuration file
XDD	XML Device Description file
XML	Extensible Markup Language

Table of Contents

Table of Figures

1. Introduction

1.1. Context

The purpose of this document is to help users to setup openCONFIGURATOR, quickly create and edit a sample network configuration project with a CN having one RPDO and one TPDO.

1.2. Intended audience and reading suggestions

A common knowledge of POWERLINK and/or CANopen technology is assumed throughout this document.

1.3. Scope

This document limits its scope with explaining how to create the demo project using openCONFIGURATOR.

1.4. References

- openCONFIGURATOR user manual v1.4.1
- Ethernet POWERLINK Communication Profile Specification 301 Version 1.2.0

1.5. Terminology used in this document

To make reading the docs easier, the names of all the screens and Menus from openCONFIGURATOR are marked up in a different font. The Build Project for instance.
A menu choice is indicated with an arrow. View → Advanced view means: select Advanced view from the openCONFIGURATOR View menu.

User Interface buttons are indicated like this: Press Ok to continue.

Important:
Important notes are indicated with this icon.

Warning:
Very important warnings are indicated with this icon. If such warnings are ignored, it might lead to data corruption or unpredictable behavior in the application.

2. Setup

Download latest version of openCONFIGURATOR from http://www.sourceforge.net/projects/openconf and refer openCONFIGURATOR user manual v1.07 for installation instructions.

3. What is openCONFIGURATOR?

openCONFIGURATOR is an open-source configuration tool for easy setup, configuration and maintenance of any POWERLINK network. It ideally complements openPOWERLINK, the open source POWERLINK protocol stack for master and slave.

The application and the source code for the openCONFIGURATOR are licensed under BSD license. Please refer to the License section for more information.

4. Sample project

This sample project will be used to create a network configuration having one CN with one byte TPDO & one byte RPDO mapped to the MN.

4.1. Create a sample project

The user can use the wizard to create a sample project.

To create a project, choose Create New Project option from the Wizard and click Ok as shown in Figure 1: Project wizard

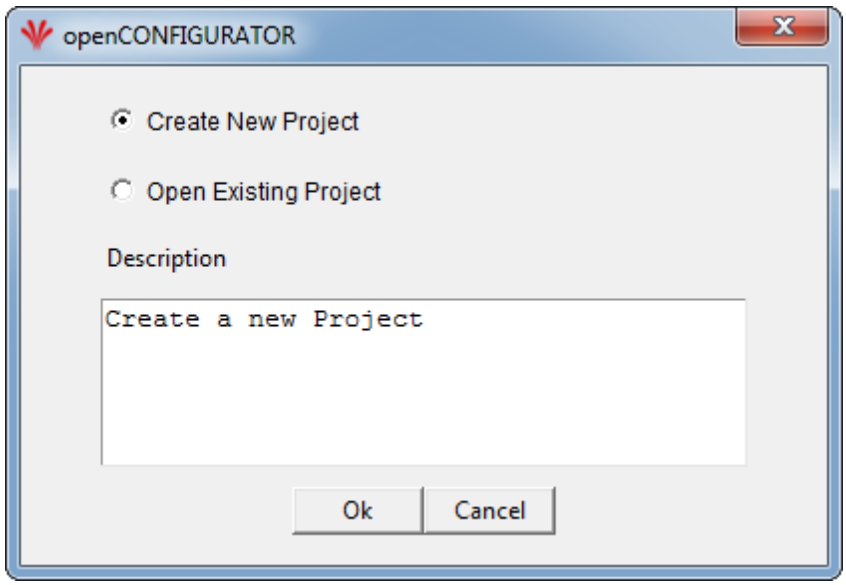

Figure 1: Project wizard

Alternatively, the user can create a new project by selecting File → New Project or by using the keyboard shortcut CTRL + N.

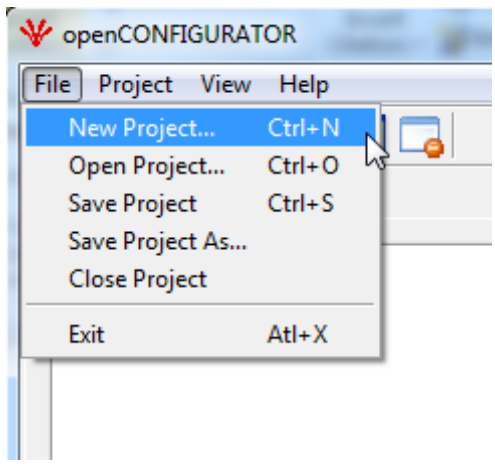

Figure 2: New project menu

After clicking on the Ok button the tool will guide you through a project creation wizard:

- Set the ‘Project Name’ for the project.
- Choose the project location by clicking on Browse.
- Choose the ‘Save’ option as required.

Options	Description
Auto Save	Saves the configuration automatically without prompting the user
Prompt (Default)	Prompt the user with the option to save before exiting from the project
Discard	Requires the user to manually save the configuration by clicking save button

Table 1: Save options

In this sample project, enter 'Sample' as the 'Project Name' and select 'Prompt' as the save option and click Next button as shown in Figure 3: Project wizard - settings

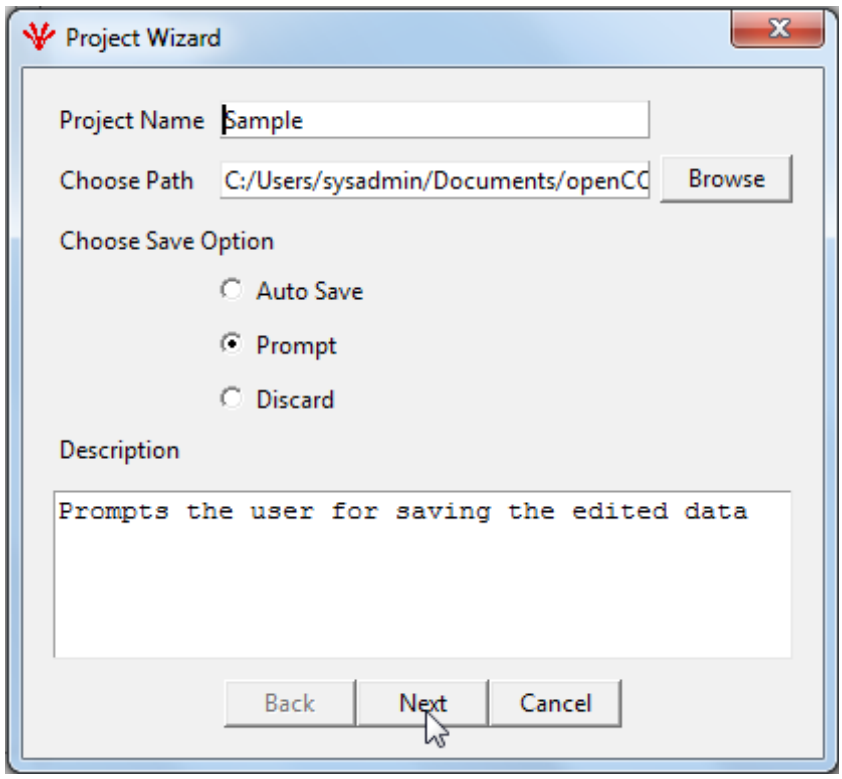

Figure 3: Project wizard - settings

After clicking on the Next button the tool will guide you through the next step of MN configuration.

Configuration option	Description
Default	Default MN XDD which will be available with the installation package.
Import XDD/XDC	User defined MN configuration.

Select the relevant 'Auto Generate' option.

Auto generate option	Description
Yes	The MN configuration will be auto generated with the available CN's configuration.
No	The MN configuration will have to be manually generated/updated by the user.

In this sample project, select the 'default' MN XDD and 'Yes' for Auto Generate options and then click Ok as shown in

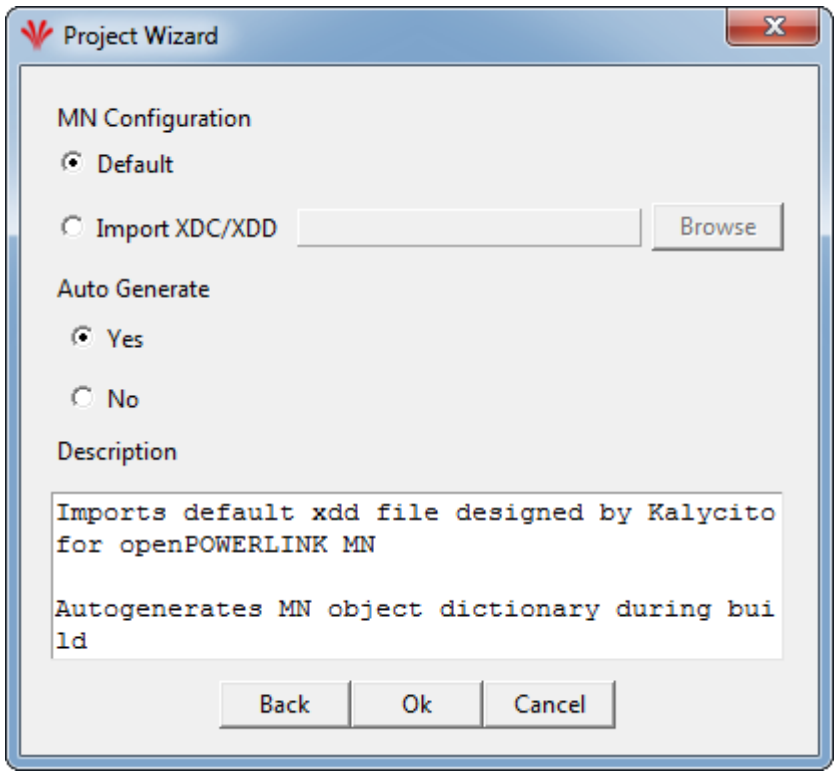

Figure 4: Project wizard - MN configuration

Now the MN node will be created and the project window will look similar to Figure 5: Initial view of the tree

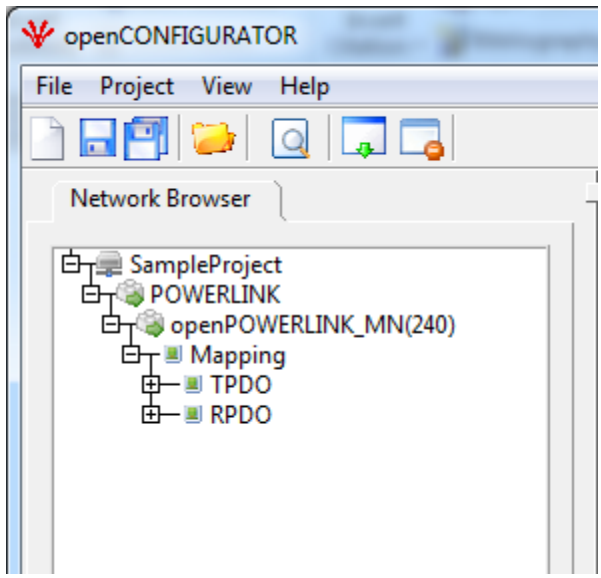

Figure 5: Initial view of the tree

4.2. Add a CN

To add a Controlled Node, right click on POWERLINK in the tree browser and choose Add CN option as shown in Figure 6: Add a CN

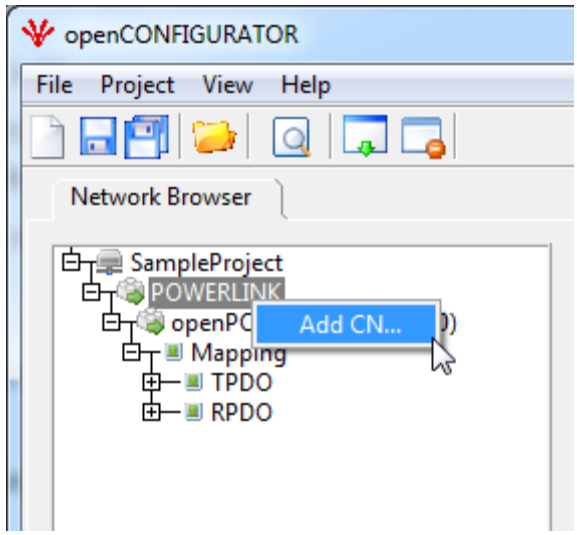

Figure 6: Add a CN

Now the 'Add New Node' dialog box will pop prompting the user to give the required configuration for the CN.

In this sample project, enter Name as 'CN_1' and NodeID as '1' and select 'Import XDC/XDD' to import the XDD from the openPOWERLINK source code.

Click Browse button and import the 00000000_POWERLINK_CiA401_CN.xdd as the configuration as shown in Figure 7: Add CN window

Important:
The sample XDD 00000000_POWERLINK_CiA401_CN.xdd is present in openPOWERLINK source directory.
Path: openPOWERLINK2/objdicts/CiA401_CN

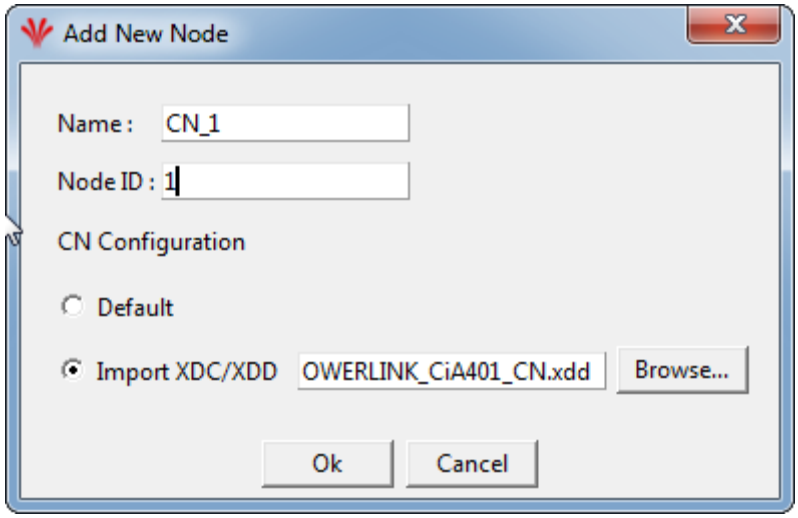

Figure 7: Add CN window

After adding a CN the project window will look similar to Figure 8: CN properties

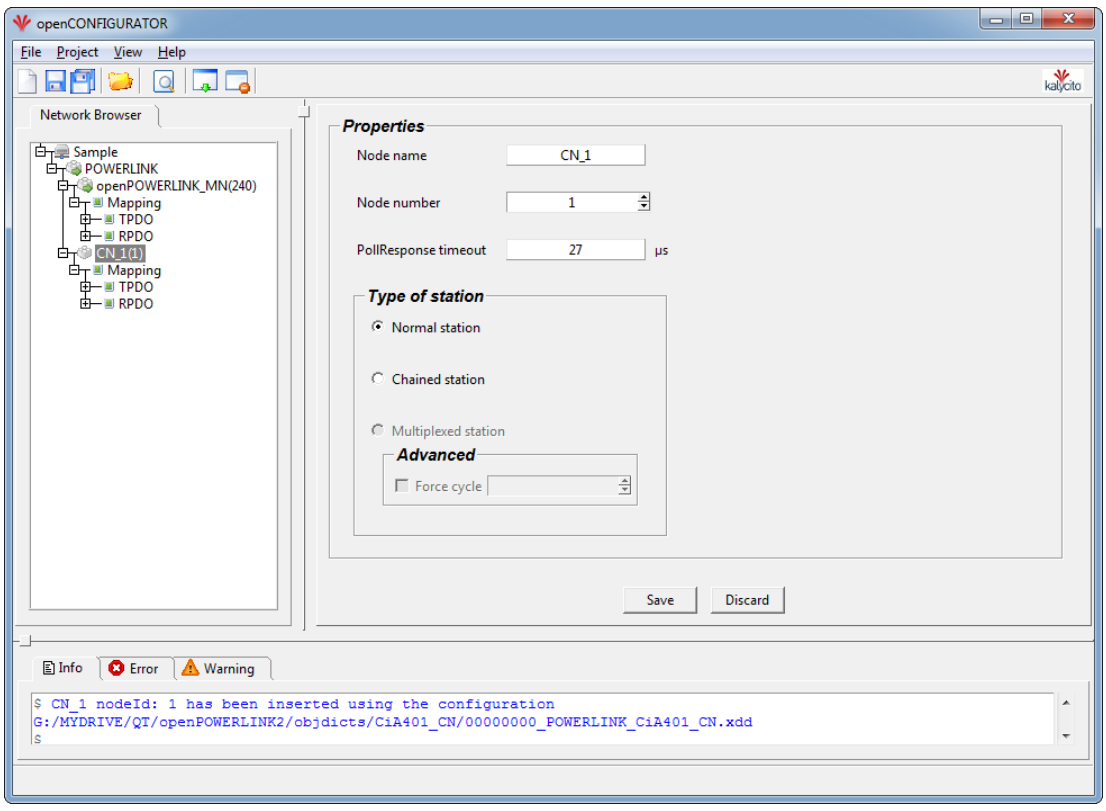

Figure 8: CN properties

4.3. PDO Mapping for the CN process variables

A PDO (Process Data Object) is used to exchange process variables between the managing node (MN) and the controlled nodes (CN) of the POWERLINK network.

The two types of PDO are Receive PDO (RPDO) and Transmit PDO (TPDO). (For more information, refer section 6.4 of Ethernet POWERLINK Communication Profile Specification Version 1.2.0).

Click on TPDO-00 from the "Tree Browser". A PDO mapping table will be loaded on the right pane as shown in Figure 9: PDO mapping table

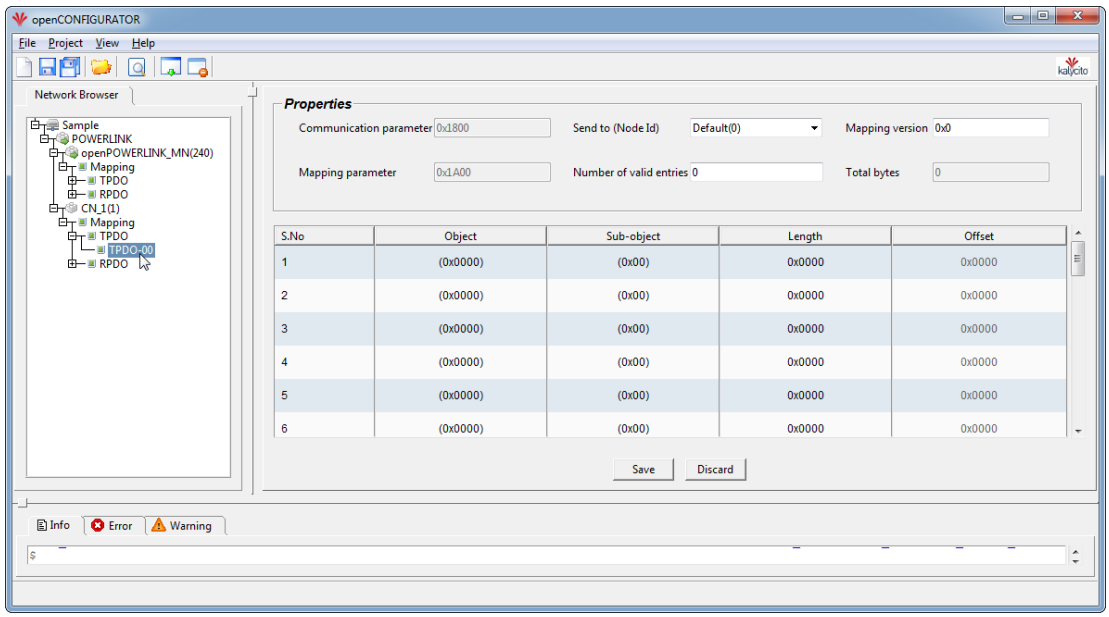

Figure 9: PDO mapping table

Select the appropriate 'Send to (Node ID)' as shown in Figure 10: Select target node ID

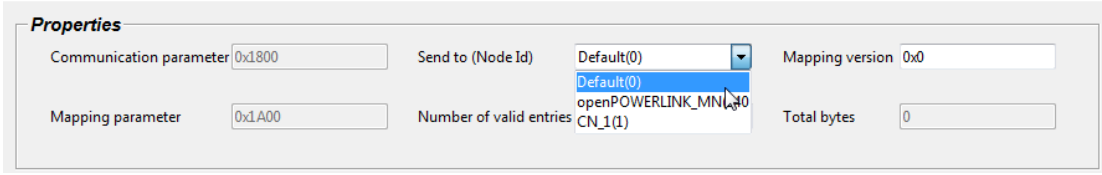

Figure 10: Select target node ID

Select an ‘Object’ value available in that list as shown in Figure 11: PDO mapping table - Select object

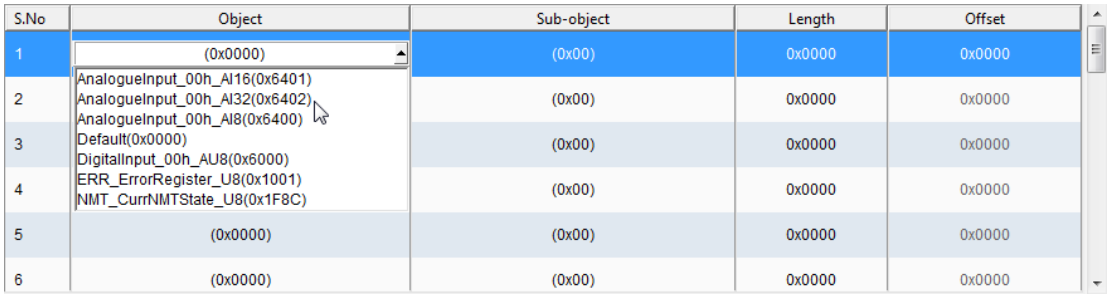

Figure 11: PDO mapping table - Select object

Select a ‘Sub-object’ value available in the list as show in Figure 12: PDO mapping table - Select sub-object

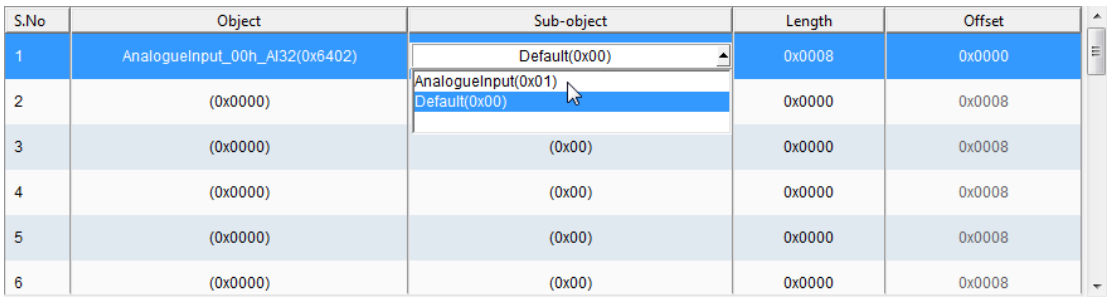

Figure 12: PDO mapping table - Select sub-object

The ‘Length’ and ‘Offset’ values will be updated automatically after the ‘Object’ or ‘Sub-object’ is selected as shown in Figure 13: PDO mapping table - Overview

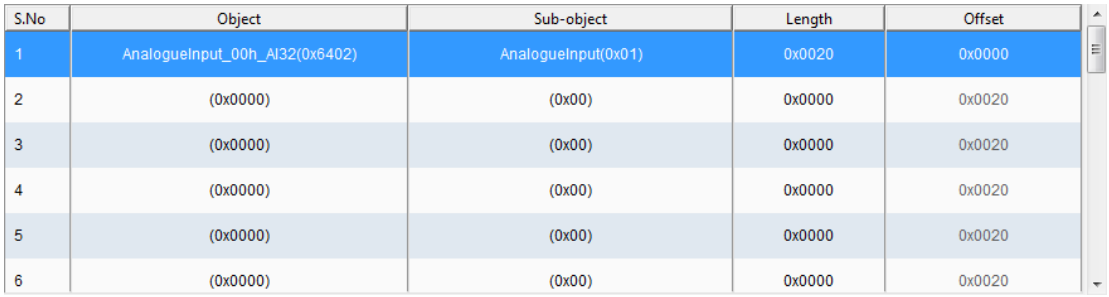

Figure 13: PDO mapping table - Overview

Save the changes by clicking on the Save button or discard the changes by clicking on the Discard button.

Similarly, create the PDO mapping for the input process variable with the Object (0x6200 - Digital_Output), Sub-object (0x01 - Digital_Output_01) as shown in Figure 14: PDO mapping table - RPDO mapping values

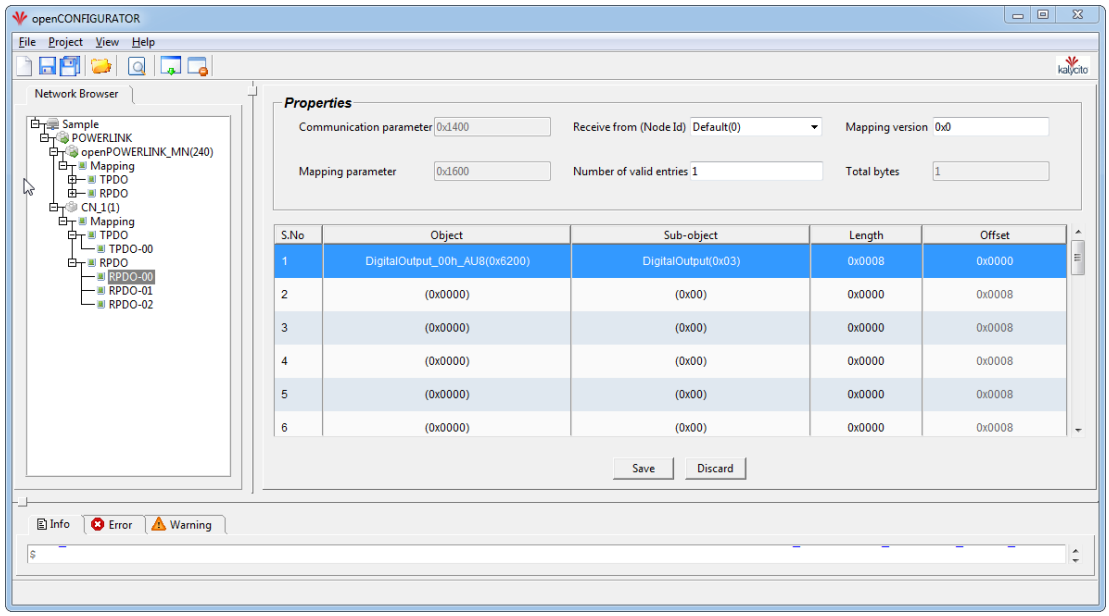

Figure 14: PDO mapping table - RPDO mapping values

4.4. Build the sample project

The user can build the project by selecting Project → Build Project as shown in Figure 15: Build project or by using the keyboard shortcut F7 or by clicking on build project icon.

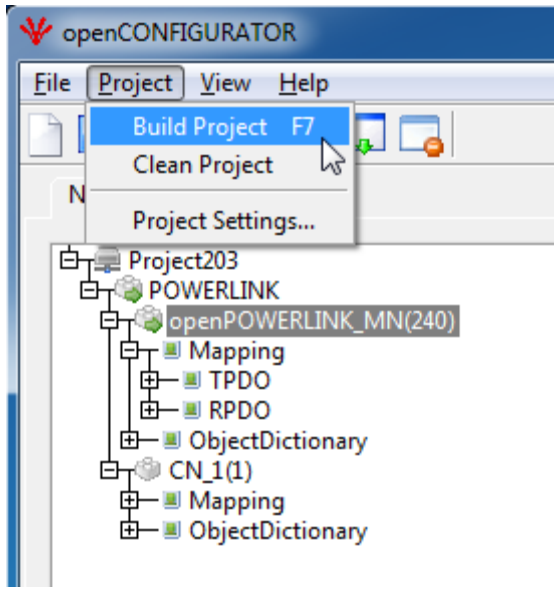

Figure 15: Build project

After selecting the 'Build Project' option, a message pop-up will indicate that the user edited values for MN will be lost since PDO mapping will be calculated automatically. Click Yes as shown in Figure 16: Build project - Auto generate MNOBD since Auto Generate MN OBD is enabled. This will automatically generate MN PDO mapping.

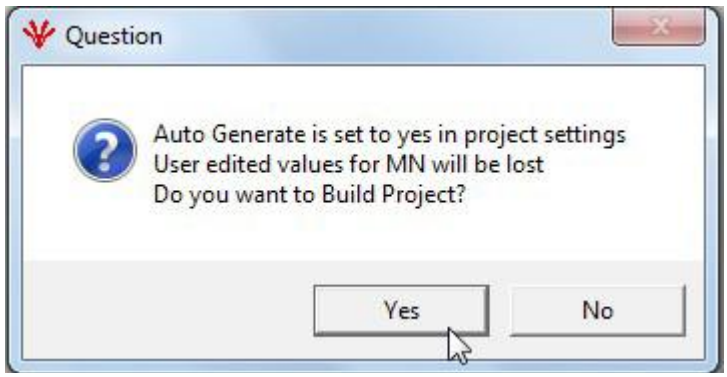

Figure 16: Build project - Auto generate MNOBD

The files listed in Figure 17: Output files will be created after a successful build of the project. These files will be present in <Project location >/<Project Name>/output directory.

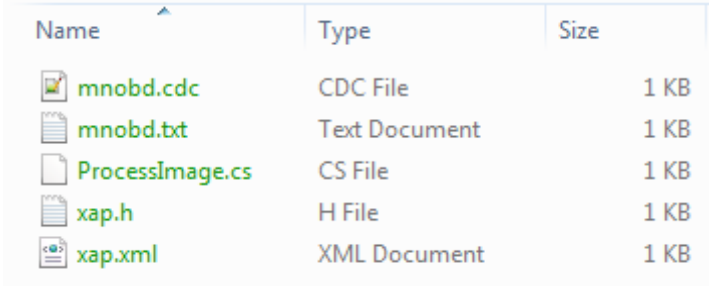

Figure 17: Output files

Output files		Description
	mnobd.cdc	Concise Device Configuration - binary file used with the openPOWERLINK stack
	mnobd.txt	Text version of the binary CDC file
	XAP.h	ANSI C header file that describes the process image for C projects
	ProcessImage.cs	C# class that describes the process image for .NET projects
	XAP.xml	XML file that describes the process image generally

Table 2: openCONFIGURATOR output files

mnobd.txt will be generated as below

```
00000011
//// NodeId Assignment
1F81    01  00000004    00000007

1600    00  00000001    00
1A00    00  00000001    00
1C02    01  00000004    00000028
1C02    03  00000004    00000028
1C09    01  00000004    00000028
1F26    01  00000004    00002bfa
1F27    01  00000004    034be328
1F92    01  00000004    00006978
1400    01  00000001    01
1600    01  00000008    002000000001A640
1800    01  00000001    01
1A00    01  00000008    000800000001A040
1600    00  00000001    01
1A00    00  00000001    01
////Configuration Data for CN-1
1F22    01  00000079
0000000C
1600    00  00000001    00
1A00    00  00000001    00
1006    00  00000004    00000190
1020    01  00000004    00002bfa
1020    02  00000004    034be328
1C0B    03  00000004    00000050
1C0D    03  00000004    00000050
1600    01  00000008    0008000000036200
1A00    01  00000008    0020000000016402
1600    00  00000001    01
1A00    00  00000001    01

//// NodeId Reassignment
1F81    01  00000004    80000007
```

XAP.h will be generated as below

```
/* This file was autogenerated by openCONFIGURATOR-1.4.0 */
#ifndef XAP_h
#define XAP_h

# define COMPUTED_PI_OUT_SIZE 4
typedef struct
{
    unsigned CN1_M02_AnalogueInput_00h_AI32_AnalogueInput:32;
} PI_OUT;

# define COMPUTED_PI_IN_SIZE 4
typedef struct
{
    unsigned CN1_M00_DigitalOutput_00h_AU8_DigitalOutput:8;
    unsigned PADDING_VAR_1:24;
} PI_IN;

#endif
```

XAP.xml will be generated as below

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This file was autogenerated by openCONFIGURATOR-1.4.0 -->
<ApplicationProcess>
  <ProcessImage type="output" size="4">
    <Channel Name="CN1.M02.AnalogueInput_00h_AI32.AnalogueInput"
dataType="Integer32" dataSize="32" PIOffset="0x0000"/>
  </ProcessImage>
  <ProcessImage type="input" size="1">
    <Channel Name="CN1.M00.DigitalOutput_00h_AU8.DigitalOutput"
dataType="Unsigned8" dataSize="8" PIOffset="0x0000"/>
  </ProcessImage>
</ApplicationProcess>
```

ProcessImage.cs will be generated as below

```
using System;
using System.Runtime.InteropServices;
/// <summary>
/// This file was autogenerated by openCONFIGURATOR-1.4.0
/// </summary>

namespace openPOWERLINK
{
    /// <summary>
    /// Struct : ProcessImage Out
    /// </summary>
    [StructLayout(LayoutKind.Explicit, Pack = 1, Size = 4)]
    public struct AppProcessImageOut
    {
        [FieldOffset(0)]
        public Int32 CN1_M02_AnalogueInput_00h_AI32_AnalogueInput;
    }

    /// <summary>
    /// Struct : ProcessImage In
    /// </summary>
    [StructLayout(LayoutKind.Explicit, Pack = 1, Size = 4)]
    public struct AppProcessImageIn
    {
        [FieldOffset(0)]
        public byte CN1_M00_DigitalOutput_00h_AU8_DigitalOutput;
        [FieldOffset(1)]
        public byte PADDING_VAR_1;
        [FieldOffset(2)]
        public byte PADDING_VAR_2;
        [FieldOffset(3)]
        public byte PADDING_VAR_3;
    }
}
```

5. Conclusion

The project creation, configuration setup and generation of CDC, XAP & C# namespace ouputs has been demonstrated for a project having a CN with one RPDO and one TPDO. These outputs can be used as an input to openPOWERLINK MN for running the application.

6. References

- EPSG Draft Standard 301 v1.1.0_01 available at http://www.ethernet-powerlink.org
- XML Device Description Implementation Guidelines v1.0.0 available at http://www.ethernetpowerlink.org
- openCONFIGURATOR High level design document v1.3 available at http://www.sourceforge.net/projects/openconf
- openCONFIGURATOR User quick start guide v1.3 available at http://www.sourceforge.net/projects/openconf
- openPOWERLINK wiki pages available at http://sourceforge.net/p/openpowerlink/wiki/
- Complete openPOWERLINK guide is available at http://openpowerlink.sourceforge.net/

7. Support

7.1. Sourceforge forum

For more information on using openCONFIGURATOR, please post on the help forum at http://sourceforge.net/p/openconf/discussion/help/

7.2. Readme

The Readme.txt present in the openCONFIGURATOR installation directory lists the feature additions, bug fixes and known issues for that version.