

DeviceNet Group 2 Slave Library Module

1. Introduction	2
2. Module Features.....	2
3. List of Component Modules.....	3
4. Using the Library Module in a Project.....	5
5. List of Shared Parameters	6
6. Functions.....	6
7. Macros	6
8. Error and Status Flags.....	6

1. Introduction

DeviceNet is an open network standard built on Controller Area Network (CAN) designed to reduce the cost and time to install industrial devices while providing compatibility with multiple vendors. The specification is available from the Open DeviceNet Vendor Association, Inc. (ODVA). Example DeviceNet devices might include motor starters, valves, sensors, displays, and more.

The specification covers multiple layers from the wiring and protection circuits up to the software protocol and application definition. This module provides a specific subset of the specification known as the Predefined Master/Slave Connection Set. To be even more accurate this module only provides low-level firmware to support a slave node within the Predefined Connection Set also referred to as a Group 2 Slave.

The intent of this module is to accelerate Group 2 Slave device development by providing most of the low-level network management. Please refer to the associated application note (AN877) for more details on this firmware.

2. Module Features

- Firmware developed in C
- Supports Polling Messaging
- Supports Multicast Polling Messaging
- Supports Change of State / Cyclic Messaging
- Supports Bit-strobe Messaging
- Supports I/O and Explicit Message Fragmentation

3. List of Component Modules

Conn.c Conn.h	Connection Object managing functions, services, and macros. Include these files in your project. And include the header file in your application object.
Conn1.c Conn1.h	Explicit Messaging event functions. Include these files in your project.
Conn2.c Conn2.h	Polled Messaging event functions. Include these files in your project.
Conn3.c Conn3.h	Bit-Strobe Messaging event functions. Include these files in your project.
Conn4.c Conn4.h	Cyclic/COS Messaging event functions. Include these files in your project.
Conn5.c Conn5.h	Multicast Polled Messaging event functions. Include these files in your project.
Conn6.c Conn6.h	Unconnected Explicit Messaging event functions. Include these files in your project.
Conn7.c Conn7.h	Duplicate MAC ID Messaging event functions. Include these files in your project.
Frag.c Frag.h	Fragmentation support for I/O messages. Include these files in your project.
Can.c Can.h	CAN driver function for the PIC18FXX8 devices. Include these files in your project.
Emm.c Emm.h	Explicit Messaging Management. Include these files in your project.
Uemm.c Uemm.h	Unconnected Explicit Messaging Management. Include these files in your project.
Nasm.c Nasm.h	Network Access State Machine Functions. Include these files in your project.
Route.c Route.h	Routing management and internal Explicit Messaging services. Include these files in your project.
Usrconn.c Usrconn.h	User definable Connection Events. Include these files in your project. <u>Code must be developed in this object.</u>
Dnet.c Dnet.h	DeviceNet Object support and internal device network information. Include these files in your project.
Usrdnet.c Usrdnet.h	User definable DeviceNet Object Events. Include these files in your project. <u>Code must be developed in this object.</u>
Ident.c Ident.h	Identity Object support and internal device status information. Include these files in your project.
Usrident.c Usrident.h	User definable Identity Object Events. Include these files in your project. <u>Code must be developed in this object.</u>
GoDnet.c GoDnet.h	Main process and dispatch routines. Include these files in your project.
Class.h	DeviceNet class codes. Include this file in your project.
Error.h	DeviceNet error codes. Include this file in your project.
Services.h	DeviceNet error codes. Include this file in your project.
Typedefs.h	Internal type definitions. Include this file in your project.
Timer.c Timer.h	Demonstrations timer routines.
App.c App.h	Demonstration application object.
Main.c	Demonstration main.
DeviceNet.mcw	Example workspace.
DeviceNet.mcp	Example project.

4. Using the Library Module in a Project

Please follow below steps to use this library module in your project.

1. Use Application Maestro to configure your code as required.
2. At the Generate Files step, save the output to the directory where your code project resides.
3. Launch MPLAB, and open the project's workspace.
4. Verify that the Microchip language tool suite is selected (*Project>Select Language Toolsuite*).
5. In the Workspace view, right-click on the "Source Files" node. Select the "Add Files" option. Select the source files specified in section 3 and click **OK**.
6. Now right-click on the "Linker Scripts" node and select "Add Files". Add the appropriate linker file (.lkr) for the project's target microcontroller.
7. Add any other files that the project may require. Save and close the project.
8. To use the module in your application, invoke the functions or macros as needed.

An example project is included. To view the project, load the example workspace `DeviceNet.mcw`.

This module has several compile-time parameters provided by the Microchip Application Maestro. The following are some recommended settings; note not all setting are shown here.

125k SJW	3 Tq	These values are set based on a 10MHz clock source (Fosc).
125k BRP	3	
125k SEG2PHTS	Programmable	
125k WAKFIL	Enabled	
125k SEG1PH	3 Tq	
125k SEG2PH	3 Tq	
125k PRSEG	3 Tq	
125k SAM	1x Sample	
250k SJW	3 Tq	
250k BRP	1	
250k SEG2PHTS	Programmable	
250k WAKFIL	Enabled	
250k SEG1PH	3 Tq	
250k SEG2PH	3 Tq	
250k PRSEG	3 Tq	
250k SAM	1x Sample	
500k SJW	3 Tq	
500k BRP	0	
500k SEG2PHTS	Programmable	
500k WAKFIL	Enabled	
500k SEG1PH	3 Tq	
500k SEG2PH	3 Tq	
500k PRSEG	3 Tq	
500k SAM	1x Sample	
Tick resolution (ms)	8ms	Time event provided by the application. Timer.c and main.c demonstrates.
Acknowledge Time-out	1024	This value must be evenly divisible by the tick resolution.
Use ACCESS memory	yes	Forces the compiler to use ACCESS memory for specific sections of the DeviceNet stack.

5. List of Shared Parameters

The application note associated with this module describes all shared parameters. Refer to the application note (AN877) for more details.

6. Functions

The application note associated with this module describes all functions. Refer to the application note (AN877) for more details.

7. Macros

The application note associated with this module describes all macros. Refer to the application note (AN877) for more details.

8. Error and Status Flags

The application note associated with this module describes all flags. Refer to the application note (AN877) for more details.