

Application Note: Creating a Stand-Alone Host Using the SSA-485 Motion Control Library

- Demonstrates stand-alone host development using the **SSA-485** Smart Serial Adapter.
- Stand-alone host developed in C using the **SSA-485 Motion Control Library**.
- Uses Microchip's MPLAB IDE Development Environment, MPLAB C18 compiler, and MPLAB ICD 2 In-Circuit Debugger.
- Includes **SSA-485** stand-alone host sample application to control the **PIC-SERVO SC**.

1. Overview

The **PIC-SERVO**, **PIC-STEP** and **PIC-I/O** control modules are designed to be used with a host controller such as a PC or embedded control computer. This application note describes how to develop a stand-alone host using the **SSA-485** Smart Serial Adapter with a PIC18F2620 microcontroller, and programmed using the **SSA-485 Motion Control Library**. It is developed using the following tools from Microchip: the MPLAB IDE Integrated Development Environment, the MPLAB C18 C compiler, and the MPLAB ICD 2 In-Circuit Debugger. This application note first describes the setup of the Microchip MPLAB development environment and the creation/debugging/programming of a **SSA-485** demo project. It then proceeds to modify this project for a simple application using the **PIC-SERVO SC** controller.

1.1 SSA-485 Smart Serial Adapter

The **SSA-485** Smart Serial Adapter is a USB/RS232 to RS485 converter, which allows you to communicate with NMC control modules such as the **PIC-SERVO**, **PIC-STEP**, and **PIC-I/O** through your host computer's USB or RS232 COM port. The **SSA-485** may also be used as a stand-alone host adding either our **Simple Sequencer** chip or a PIC18FXXXX microcontroller. The **SSA-485** comes with an on-board connector for the Microchip MPLAB ICD 2 In-Circuit Debugger. Please download the data sheet SSA485.PDF from jrkerr.com/docs.html for complete details on the functionality and use of the **SSA-485** Smart Serial Adapter.

1.2 SSA-485 Motion Control Library

The **SSA-485 Motion Control Library** is a library of C language functions for communicating with the **PIC-SERVO**, **PIC-STEP**, and **PIC-I/O** modules. This library is for use with Microchip's C18 C compiler. Included in the library are high level routines for initializing, controlling, and managing **PIC-SERVO**, **PIC-STEP**, and **PIC-I/O** modules. Please download the data sheet SSA485MCL.ZIP from jrkerr.com/software.html for the software library and documentation.

... CAUTION ...

This application note is supplied by **JEFFREY KERR, LLC** solely for the purpose of illustrating the use of its products. It is intended for use only by experienced engineers. **JEFFREY KERR, LLC** makes no claims or warranties regarding the safety or functioning of the circuitry described herein. All risks associated with the use of this application note are assumed by the user.

1.3 MPLAB IDE Integrated Development Environment

MPLAB Integrated Development Environment (IDE) is a free MS Windows application from Microchip. It is an integrated tool set for developing embedded applications employing Microchip's microcontrollers, and serves as a unified graphical user interface for Microchip and third party software and hardware development tools. It includes a free simulator. For more information about MPLAB IDE, please see Microchip's web page at *microchip.com*.

1.5 MPLAB ICD 2 In-Circuit Debugger

The Microchip MPLAB ICD 2 In-Circuit Debugger is a real-time debugger and programmer for selected Microchip MCUs. The debugger allows you to develop, download, execute and debug programs on Microchip microcontrollers. The debugger comes with the MPLAB IDE Integrated Development Environment, an assembler, and a linker. The MPLAB ICD 2 connects to your PC's USB or RS232 port, and attaches to a modular jack on your target circuit using an ICD interface cable supplied with the debugger. For a complete description of the MPLAB ICD 2 In-Circuit Debugger, please see Microchip's web page at *microchip.com*.

1.4 MPLAB C18 Compiler

The MPLAB C18 compiler is a ANSI compliant C compiler for the PIC18 family of PICMicro 8-bit MCUs. It works as a fully integrated component of Microchip's MPLAB IDE allowing source level debugging with the MPLAB ICD 2 In-Circuit Debugger. The MPLAB C18 compiler is available for purchase at the Microchip web site *microchip.com*. A free Student Edition/Demo version is also available from Microchip. This version comes with all the features of the full compiler and libraries for 60 days. After 60 days the full compiler expires, but continues to work with certain optimizations disabled. For complete information on the MPLAB C18 compiler, see the Microchip web site at *microchip.com*.

2. Setting up the SSA-485 Development Environment

You will need the following items to set up a development environment for the **SSA-485** stand-alone host:

- 1 **SSA-485** Smart Serial Adapter
- SSA-485 Motion Control Library** Software
- HEXTERM Terminal Software
- 1 Microchip PIC18F2620 microcontroller
- 1 Microchip MPLAB ICD 2 In-Circuit Debugger
- Microchip MPLAB IDE software (Full or Student/Demo version)
- Microchip MPLAB C18 compiler
- PC running Windows XP or Windows 2000

Install Software

Install the software in the order listed below. To prevent software compatibility problems, use the latest versions of the MPLAB IDE and MPLAB C18 Compiler software. (Note: Do not connect the RS232 or USB cable to the MPLAB ICD 2 until after the MPLAB IDE software is installed).

1. Install the MPLAB IDE software. Download and install the latest MPLAB IDE software from the Development Tools Page of the Microchip Web Site at *microchip.com*. The software may also be installed from the MPLAB IDE CD-ROM that comes with the MPLAB ICD 2, but verify that it is the latest version before installing. Install using the default settings.
2. Install the Microchip MPLAB C18 compiler. If you purchased the MPLAB C18 compiler, follow the installation instructions included with the software. To use the MPLAB C18 Student Edition/Demo version, download the demo version from the Development Tools Page of the Microchip Web Site at *microchip.com*, and click on the downloaded file to start the installation wizard. Install in the default directory (“\mcc18”) and use the default settings. When prompted, check the box “Update MPLAB IDE to use this MPLAB C18”, and check the box “Update MPLAB IDE to use this MPLINK Linker”.
3. Install the **SSA-485 Motion Control Library** software. Create a directory \mcc18\ssa485lib. Download the file SSA485MCL.ZIP from *jrkerr.com/software.html*. From this zip file, copy the files from the “ssa485lib” subdirectory into \mcc18\ssa485lib, and copy the files from the “h” subdirectory into \mcc18\h.
4. Install HEXTERM – the Hexadecimal Format Terminal Program. Download the file HEXTERM.ZIP from *jrkerr.com/software.html*, and unzip it into a single directory.

Configure PC Communications

Follow the MPLAB ICD 2 instructions to configure PC communications with the MPLAB ICD2 using either RS-232 or USB.

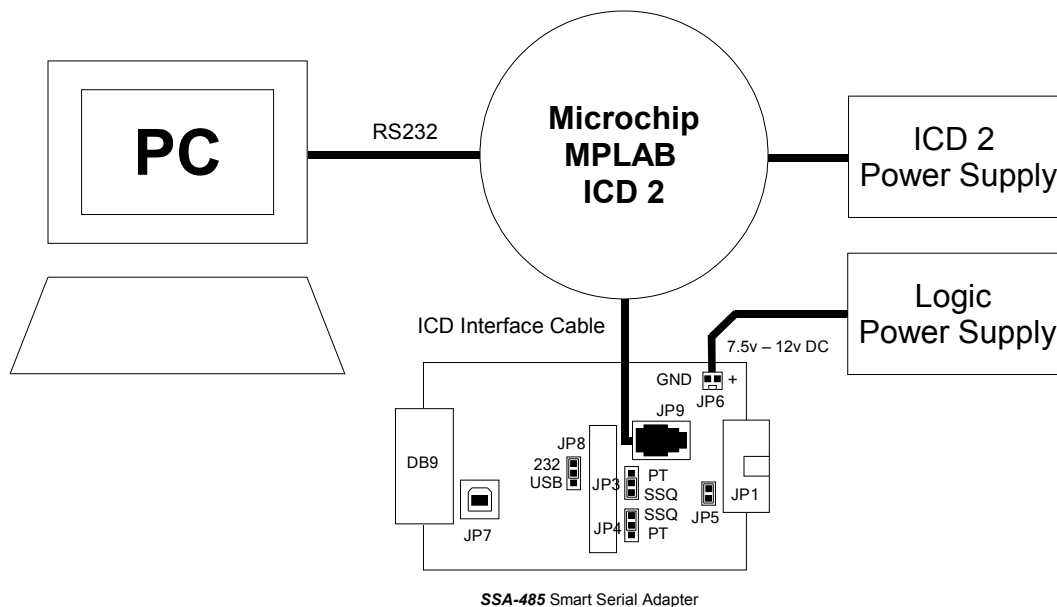


Figure 1: **SSA-485** Host Development using RS232

Connect SSA-485 and Power

There are two configurations for powering the MPLAB ICD 2 and the **SSA-485** target. In both configurations, the **SSA-485** must have it's own supply. In *Figure 1*, the MPLAB ICD 2 connects to the PC via a RS232 cable, and the ICD 2 must have a dedicated power supply (available from

Microchip). In *Figure 2*, the MPLAB ICD 2 connects to the PC via a USB cable, and the MPLAB ICD 2 is powered by the PC over the USB cable.

Connect the **SSA-485** to the ICD 2, and the ICD 2 to your PC as follows:

1. Install the PIC18F2620 chip into socket U4 on the **SSA-485** board.
2. Set JP3 and JP4 to the 'SSQ' position (this sets the board as a stand-alone host) and make sure jumper JP5 is installed (this will supply logic power to the NMC controller boards from connector JP1).
3. Connect the RS232 or USB cable from the PC to the MPLAB ICD 2. If using RS232, connect power supply to MPLAB ICD 2
4. Connect the ICD 2 to modular jack JP9 on the **SSA-485** board using the ICD interface cable that is supplied with the MPLAB ICD 2.
5. Connect a Logic Power Supply (7.5 – 12 vdc) to jumper JP6 of the **SSA-485** board. **Do not apply Logic Power supply at this time.**

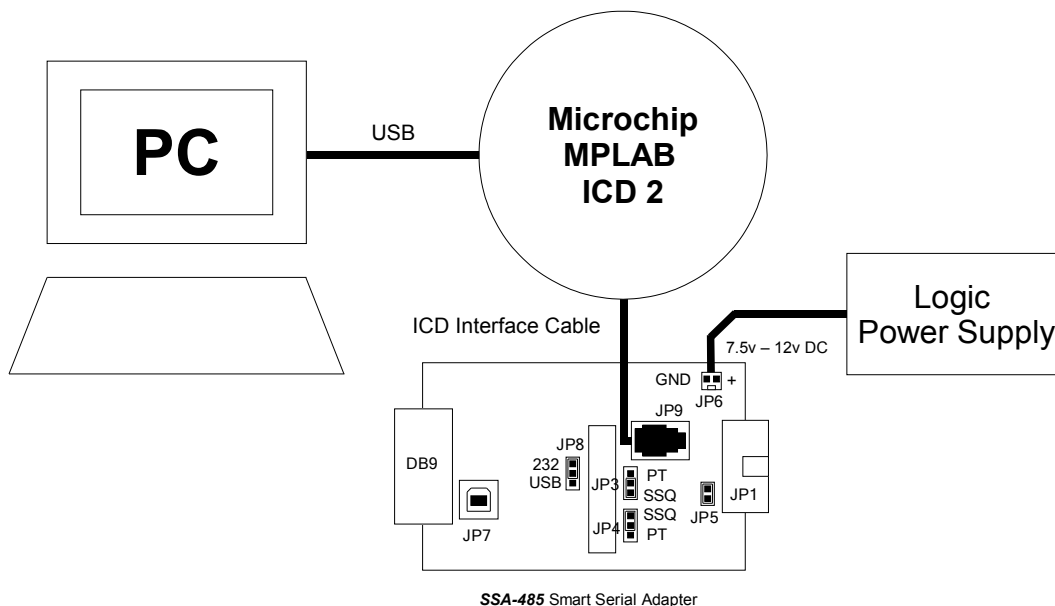


Figure 2: SSA-485 Host Development using USB

Configure MPLAB IDE

Start the MPLAB IDE program and configure to run with the ICD 2 and **SSA-485** as follows:

1. Start MPLAB IDE. The MPLAB IDE desktop will appear.
2. Select *Debugger>Select Tool>MPLAB ICD 2* to put the MPLAB IDE in debugging mode with the ICD 2.
3. Select the *Debugger>Settings>Communication* tab, and choose the communications port (USB, COM1, COM2, etc) to use with the ICD 2. Click OK. Make sure you install any USB drivers according to the MPLAB ICD 2 instructions.
4. Select the *Debugger>Settings>Power* tab, and verify that the box “Power target circuit from MPLAB ICD 2” is not checked. If checked, uncheck box and click OK.
5. Select *Configure>Select Device* and the Device dialog box will appear. Select PIC18F2620 from the device list and click OK.

3. Creating an SSA-485 Project with MPLAB IDE

This section describes creating a **SSA-485** stand-alone host project using the MPLAB IDE.

1. Create a directory for the **SSA-485** stand-alone host project named “\ssa485demo”. From the SSA485MCL.ZIP file, copy the files in the “demo” subdirectory into \ssa485demo
2. Start the MPLAB IDE software. Create a MPLAB IDE project using the Project Wizard by selecting *Project>Project Wizard*. When the Welcome dialog appears, click on Next to advance.
3. Select a device. From the Device dialog, select and verify that the device says PIC18F2620, then click on Next.
4. The “Step Two: Select language toolsuite” dialog should be displayed. Select “Microchip C18 Toolsuite” from the Active Toolsuite list box. The Toolsuite Contents box should now list “MPASM Assembler”, “MPLINK Object Linker”, “MPLAB C18 C Compiler”, and “MPLIB Librarian”. Click on each of the tools and verify that their location matches the locations listed below (this assumes MPLAB IDE and MPLAB C18 were installed in their default location):

MPASM Assembler :	\mcc18\mpasm\mpasmwin.exe
MPLINK Object Linker:	\mcc18\bin\mplink.exe
MPLAB C18 C Compiler:	\mcc18\bin\mcc18.exe
MPLIB Librarian:	\mcc18\bin\mplib.exe

If the locations are not correct, use the browse button to select the correct locations, then click on the Next button.

5. The “Step Three: Name your project” dialog box should be displayed. Enter “nmcdemo” in the project name box, and use the Browse button and select “\ssa485demo” for the project directory box, then click on the Next button.
6. The “Step Four: Add any existing files to your project” dialog box should be displayed. Use the Add Button to add the following files to the project list:

```
\mcc18\lkr\18f2620.lkr
\mcc18\lib\p18f2620.lib
\mcc18\ssa485lib\nmccom.c
\mcc18\ssa485lib\picio.c
\mcc18\ssa485lib\picservo.c
\mcc18\ssa485lib\picstep.c
\mcc18\ssa485lib\openuart.asm
\mcc18\ssa485lib\readuart.asm
\mcc18\ssa485lib\writuart.asm
\ssa485demo\nmcdemo.c
\ssa485demo\nmcdemo.h
```

Once files are added to the project list, do *not* need to click on any of the check boxes to move files into the project directory – all files should already be in the proper locations. Click on the Next button.

7. The project summary dialog box should be displayed. Verify the device (“PIC18F2620”), tool suite (“Microchip C18 Toolsuite”), and project name (“c:\ssa485demo\nmcdemo.mcp”). If everything looks correct, click on the Finish button.

8. The nmcdemo Project Window should now be displayed on the MPLAB IDE desktop (if not, open the project window from *View>project*). Verify that the following files are displayed:

<u>Source Files</u>	<u>Header Files</u>	<u>Library Files</u>	<u>Linker Scripts</u>
nmcocom.c	nmcdemo.h	p18f2620.lib	18f2620.lkr
nmcdemo.c			
picio.c			
picservo.c			
picstep.c			
openuart.asm			
readuart.asm			
writuart.asm			

9. Build the project. From the Project Menu select *Project>Build All*. The project should start building with build status displayed in the Output window. At the end of a successful build, the Output window will display the line “BUILD SUCCEEDED” followed by the date.

4. Debugging the SSA-485 Project with the MPLAB IDE ICD 2

After your project is successfully built, you can download it in “debug” mode to the PIC18F2620 processor and then debug it using the MPLAB IDE and the MPLAB ICD 2. This section shows how to start the debugger, download the code, set a breakpoint, put a variable in the watch window, and step through the program. This section assumes that all the steps outlined in *Section 3* have been completed and that the project file “nmcdemo.mcp” is loaded.

1. Turn on the logic power supply connected to the **SSA-485** board.
2. In the MPLAB IDE, select *Debugger>MPLAB ICD 2* to put MPLAB in debugging mode with the MPLAB ICD 2.
3. Select *Debugger>Connect* to connect to the MPLAB ICD 2. If the warning “ICDWarn0030: MPLAB ICD2 is about to download a new operating system...” appears, press OK to continue.
4. Select *Debugger>Program* to download your code and debug executive to the PIC18F2620 on the **SSA-485** target board.
5. Open the file nmcdemo.c by double-clicking on it in the Project Window titled nmcdemo.mcp. A window will open displaying nmcdemo.c.
6. Scroll down to main() in nmcdemo.c. main() contains demo application code that initializes the hardware UART, initializes the NMC network with a call to NmcInit(), then enters an infinite while loop that sends the contents of the global “nummod” and the local variable “count” out the hardware UART. The global “nummod” contains the number of NMC modules detected, and “count” is a counter that increments after each pass through the while loop.
7. Scroll down in main() and double-click on the first line within the while (1) loop that says “putcUSART(count++)” to set a breakpoint at that line.
8. Select *Debugger>Reset>Processor Reset* to reset the processor. A window displaying the startup code should be displayed. You can go ahead and close this window.
9. Select *Debugger>Run* (or F9) to start the debugger. The program should run then stop at breakpoint on main().
10. Select *Debugger>Run* (or F9) again. The program should run and stop at the

breakpoint in the while loop.

11. Select *Debugger>Step Over* (or F8) a couple of times to step through individual lines in `main()`.
12. Select *View>Watch* to open the Watch Window. Add the symbols “count” and “nummod” to the Watch Window by selecting the symbols from the right pop-down window then pressing the Add Symbol button. The symbol “nummod” should have a value of 0 because no NMC modules are connected to the **SSA-485**, and the symbol “count” should have the value of 1.
13. Select *Debugger>Run* several times. With each iteration of the while loop the value of “count” displayed in the Watch Window should increment.
14. Select *Debugger>Reset>Reset Processor* to reset the processor.

5. Programming the SSA-485 Project for Stand-alone Operation (*no debug*)

You are now ready to program the demo application into the PIC18F2620 for stand-alone operation with no debugging:

1. Select *Programmer>Select Programmer>MPLAB ICD 2* to load the programmer. Press OK if the warning window “ICDWarn0037: MPLAB IDE No longer allows MPLAB ICD 2 to be loaded as a programmer and debugger simultaneously...” appears.
2. Select *Programmer>Connect* to connect the MPLAB ICD 2 to the **SSA-485** board.
3. Select *Programmer>Program* to program the PIC18F2620.
4. Turn off power to the **SSA-485** board and disconnect the MPLAB ICD 2 interface cable.
5. Connect a USB (or RS232) cable from the **SSA485** USB socket JP7 (or RS232 socket P1) to your PC, and set JP8 of the **SSA-485** to the corresponding “USB” or “232” setting.
6. Start HEXTERM. From the HEXTERM Window, set the HEXTERM Baud Rate to 19,200, and the Com Port to the port that is connected to the **SSA-485**.
7. Turn on the power to the target.
8. The HEXTERM output should start scrolling characters such as “00 00 01 00 02 00 03 00”. The incrementing characters are the value of variable “count” and the repeating 00 character is the value of variable “nummod”.

6. Modify the SSA-485 Project for use with a PIC-SERVO-SC Controller

We can now modify the demo project for use with the **SSA-485** connected to a **PIC-SERVO SC** motor controller. The host will initialize NMC communications with the **PIC-SERVO SC**, set the servo PID operating parameters, and send simple motion control commands. In addition to your current setup, you will need to add the following items:

- 1 **PIC-SERVO SC** Motion Control Board (p/n KAE-T0V10-BDV1)
- 1 Motor power supply: 12 – 48vdc, with enough power for motor
- 1 DC brush-type motor with quadrature encoder
- 1 NMC communications cable (p/n KAE-CC20-AC)
- 1 Motor/encoder cable (DB15 male connects to your motor)

Cabling

Most of the cables are available from *jrkerr.com* or from computer or electronic stores. However, you will probably have to make your own motor/encoder cable to connect to your particular motor. Refer to *Section 2.1* of the **PIC-SERVO SC Motion Control Board** data sheet for the connector pin definitions. To start off, you only need to connect M+, M-, Encoder A, Encoder B, Encoder +5v and Encoder GND. Other connections can be made as needed. Note that when testing, you may have to swap the M+ and M- leads to correct for the polarity of your motor.

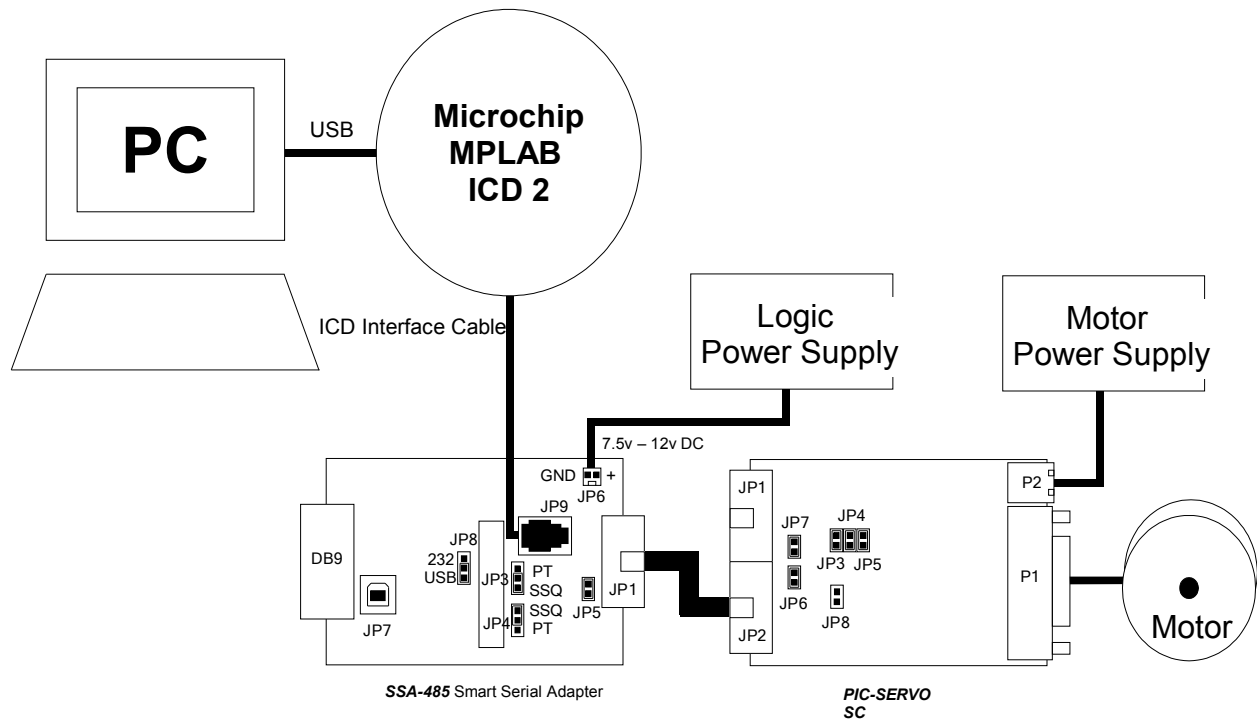


Figure 3: SSA-485 Stand-Alone Host for **PIC-SERVO SC**

Setup

This sample application will use the MPLAB IDE Project “nmcdemo” created earlier.

1. Set up the sample application components as shown in *Figure 3*. Do not apply power to the target or motor at this time.
2. Start MPLAB IDE. Select *Project>Open* to open the project `\ssa485demo\nmcdemo.mcp`. Close all windows except the Project window and the Output window.
3. Open the file `nmcdemo.c` by double-clicking on it in the Project Window.
4. Conditionally compile out all of the `main()` commented “`//demo main`” using preprocessor statements “`#if 0`” and “`#endif`” as follows:

```
#if 0
void main(void) //demo main
{
...
}
#endif
```

5. Scroll down to the `main()` commented “`//application main`” and make it the new

- main() for the build by changing the “#if 0” above it to “#if 1”.
6. Select *Project>Build All* to build the sample application.
 7. Select *Debugger>Select Tool>MPLAB ICD 2* to load the debugger. Press OK if the warning window “ICDWarn0036: MPLAB IDE No longer allows MPLAB ICD 2 to be loaded as a programmer and debugger simultaneously...” appears.
 8. Turn on the logic power to the **SSA-485** target.
 9. Select *Debugger>Connect* to connect to the MPLAB ICD 2.
 10. Select *Debugger>Program* to download your code and debug executive to the PIC18F2620 on the **SSA-485** target board.
 11. Select *Debugger>Reset>Reset Processor* to reset the processor.
 12. Select *Debugger>Run* (or F9) to run the sample application. The application will initialize NMC communications with the **PIC-SERVO SC**, then wait for motor power to be turned on.
 13. Turn on power to the the motor.
 14. The motor should turn forward then backward 4 times, then stop.
 15. Select *Debugger>Halt* to stop the application.
 16. To program the sample application on the **SSA-485** board for stand-alone use, follow the steps in *Section 5*.