

630 Komas Drive | Suite 200

Salt Lake City | UT 84108 | USA

P +1 (801) 582-5533 | F +1 (801) 582-1509

www.blackrockmicro.com



NSP Extension Code

Quick Start Guide

Table of Contents

Introduction	3
Hardware Setup	3
<i>The Neural Signal Processor</i>	3
<i>The Host PC</i>	3
Software Setup.....	4
Loading and Building an Extension.....	4
<i>Loading the Common Files</i>	5
<i>Creating the Project Folder</i>	6
<i>Building the Extension</i>	7
<i>Running the Extension</i>	7
Troubleshooting	8

Introduction

The Neural Signal Processor (NSP) of the Cerebus System maintains a free CPU core to run user-defined programs that may access sampled front-end amplifier, analog input, and digital input data in real time. The extension code may print messages to the NSP monitor, send log event packets to the Host PC to be included in the Neural Event file, and control analog and digital output ports.

This guide includes all the information necessary to upload and compile sample Extension code on the NSP

Hardware Setup

The Neural Signal Processor

Extension code can run on all Cerebus NSPs with part numbers 7530, 9650, and 10411. The NSP must be connected to power and a host PC via an ethernet cable. A monitor should be connected to the DVI port on the back of the NSP for debugging. This monitor connection should be plugged in before the NSP is powered on, otherwise the monitor may not be detected properly.

The Host PC

The Host PC should be powered on and connected to the NSP via an ethernet cable. Central Software Suite must be installed on the Host PC. There are no hardware requirements for the host PC beyond the ability to write C source code and run the Extension Loader utility.

Software Setup

The Extension Loader utility provides a graphical interface to connect to the NSP, load common files and source code, compile the extension and communicate with the NSP. The Extension Loader executable file can be found in the Cerebus Windows Suite folder within the Blackrock Microsystems program files folder. A shortcut to the executable can be found in the Start menu under Blackrock Microsystems. As well Extension Loader will attempt to connect to only one instance of the NSP at a time, selectable in the drop down on the bottom left.



Loading and Building an Extension

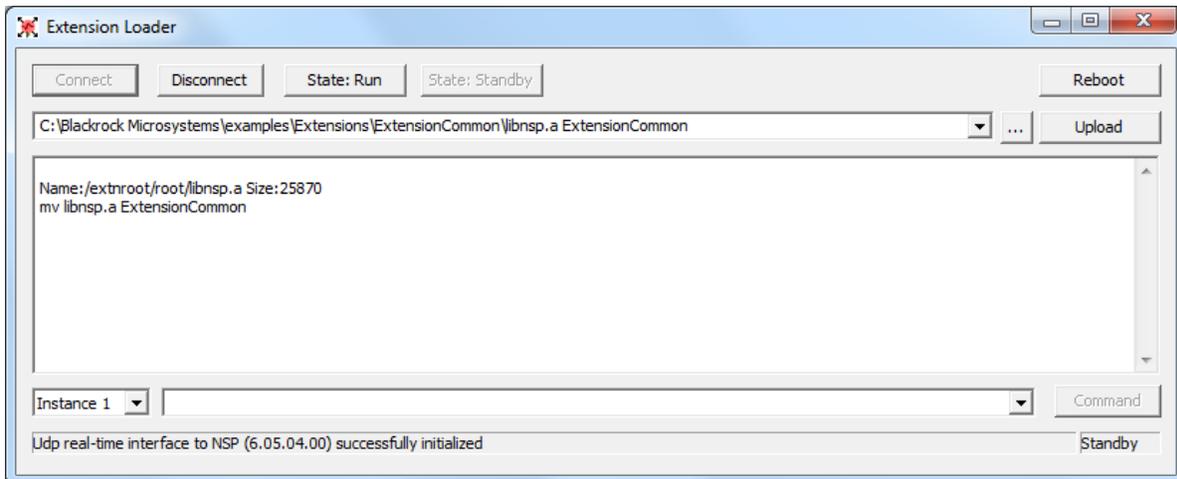
Extension code is compiled, or built, on the NSP. Example source code can be found on blackrockmicro.com (Support -> Software Downloads) or by contacting Blackrock Microsystems Support at support@blackrockmicro.com. The example code includes verbose commenting to describe what each line is doing and why. The source code references code found in common resource files that are part of the Central Software Suite installation. These files must be loaded onto the NSP before an extension can be compiled. Source code, common resource files, and a makefile must all reside on the NSP for a successful build.

Loading the Common Files

The common files are specific to the version of Central currently installed and must be updated to match. To load the common files, start the Extension Loader application from the Windows start menu or from the Blackrock Microsystems program files folder.

Perform the following steps:

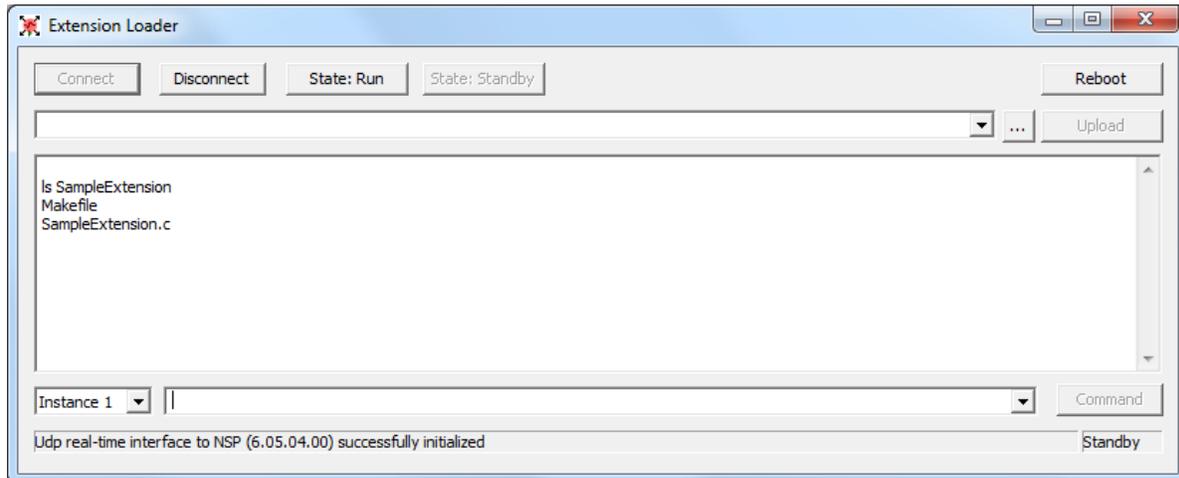
1. Click 'Connect' to initialize communication with the NSP
2. Click 'Standby' to ready the NSP for incoming file transfers
3. Make a new directory called "ExtensionCommon" to hold the resource files by typing 'mkdir ExtensionCommon' in the command line and pressing 'Enter'
4. To select files to upload, type the file path in the top address bar, or click the '...' button
5. Click the ... button and browse to libnsp.a
 - C:\Blackrock Microsystems\examples\Extensions\ExtensionCommon
6. To upload the file to a specific directory, add ' ExtensionCommon,' making sure the preceding space is included, to the line containing the file path
7. Click Upload
8. Repeat steps 4-6 for nspChanTrigPlugin.h and nspPlugin.h
9. Repeat steps 4-6 for cbhwlib.h,
 - C:\Program Files (x86)\Blackrock Microsystems\Cerebus Windows Suite\cbsdk\include
10. Verify the contents of the ExtensionCommon folder are correct by typing 'ls SampleExtension' in the command line



Creating the Project Folder

Each custom Extension will have a unique project folder containing a makefile and source code. To create and populate the project folder perform the following steps:

1. Open Extension Loader and connect as before
2. Click 'Standby' to ready the NSP for incoming file transfers
3. Make the project directory by typing 'mkdir SampleExtension' and press Enter. Any name can be used for new extensions
4. Click the ... button and browse to the location of the makefile
5. A makefile is available with the latest sample extension distributed by Blackrock Microsystems Support
6. Upload the makefile to the project directory by typing ' SampleExtension' after the file path, making sure the preceding space is included
7. Click upload
8. Click the ... button and browse to SampleExtension.c
9. SampleExtension source code is available with the latest sample extension distributed by Blackrock Microsystems Support
10. Add ' SampleExtension' making sure the preceding space is included, to the line containing the file path
11. Verify the contents of the SampleExtension folder are correct by typing 'ls SampleExtension' in the command line



Building the Extension

Once the ExtensionCommon folder contains the necessary resource files, and a sample extension project folder contains source code and a makefile, the extension can be compiled. The following steps describe how to run the build procedure.

1. Type 'make -C SampleExtension' in the command window. The built extension will be named 'nspext_default.so' and will be placed into the proper folder to run next time the NSP is rebooted.
2. When the Extension is no longer needed type 'make -C SampleExtension clean' in the command window. This will remove the built extension and any temporary object files.

If a monitor is attached to the NSP directly, any warnings or errors encountered during the build process will be printed. If no errors occur, then the extension is ready to run the next time the NSP starts.

Running the Extension

The NSP will run newly compiled extension code when it is restarted. The extension code will be loaded on every subsequent restart as long as the compiled extension is not removed from the the NSP. Specific functionality of the Extension code may be tested now by following documented testing functionality in the Sample Extension source code made available by Blackrock Support.

Troubleshooting

1. Extension Loader will not connect

If Extension Loader will not connect to the NSP, verify that the NSP is powered on and connected to the host PC with an Ethernet cable. Run Central Suite and verify that Central can connect to the NSP. If so, try running Extension Loader while Central is running. If not, verify that the Ethernet port is configured to connect to the NSP as specified in the Central Software Suite IFU found on Blackrockmicro.com.

2. An error message says 'Unable to allocate shared memory'

This error occurs when the NSP is already connected to the host PC in some other way. Verify that other connections to the NSP, such as cbmex, are closed. Run Central first, then connect with Extension Loader.

3. The 'Restart' button in Extension Loader does not work

Ensure Central is running in the Background to enable the 'Restart' Button

4. I've updated my Central Software Suite version and now my extension code won't compile or run

When Central Suite is updated the extension common files will need to be updated as well. Use the recursive remove command `'rm -r ExtensionCommon'` to delete the ExtensionCommon folder and all files contained within it. Re-create and repopulate the folder following the instructions in this guide.

5. I've changed the name of my extension, now it won't compile

If the source code file name is changed, this must be reflected in the makefile. Open the makefile in an editor and search for `'PLUGIN_SRC :='` and update the source file name on that line.

6. My extension crashes and I don't know why

Add print statements before and after lines of interest within your source code to see if code is running as expected. Print variables to confirm expected values and utilize the `'res ='` functionality built into most Extension functions.

Support

Blackrock Microsystems is proud to offer comprehensive remote support and troubleshooting. Contact our support team for additional information or help with any of our products.

www.blackrockmicro.com/technical-support

support@blackrockmicro.com

+1 (801) 582-5533

License Agreement

This License Agreement (the “Agreement”) is an agreement concluded between You, Your place of business and Blackrock. Please read the terms of this Agreement carefully following the link below. By downloading, accessing, saving or using Blackrock’s Software or any part of it, You agree with this Agreement. If You do not agree to any term hereof, quit downloading, accessing, saving or using the Software and all its components already downloaded or saved and note that You do not have any right to Use the Software.

License Agreement: <http://blackrockmicro.com/central-software-license-agreement/>