

This software package permits direct program loading and DSP control from within the Matlab environment. Data transfer between Matlab variables and variables in the DSP code is supported using both absolute address and symbol names. Both 32-bit and 64-bit versions of Matlab are supported.

To use this interface, the C8X_DEBUG.mexw32/C8X_DEBUG.mexw64 files, the C8X_CONTROL_DLL32.dll/C8X_CONTROL_DLL64.dll files, and the desired *.out file must be in a directory in the Matlab path, and the commands must be executed within that directory. **The 'init' command must be executed before any other commands are used.** Sample m-files and DSP software projects are provided for the OMAP-L138 and TMS320C6713 DSK.

This application supports the following hardware

- OMAP-L138 with the winDSK8 kernel in flash
- TMS320C6713 DSK with Educational DSP DSK6XXHPI daughtercard (DSK SW3-3 must be ON)
 - USB and serial interfaces only (no parallel port support)
- TMS320C6416T DSK with Educational DSP DSK6XXHPI daughtercard with software version 1.0.1.0 or later (DSK SW3-2 and SW3-3 must be ON)
 - USB and serial interfaces only (no parallel port support)

This software requires that the host machine have the Microsoft Visual C++ 2010 SP1 Redistributable Package support software be installed. This can be downloaded from Microsoft if it is not already installed on your machine.

Host Port Interface Issues:

This software package is intended to transfer data between the Matlab environment and 32-bit DSP variables that are of type *float*, *int*, or *unsigned int* only. All DSP variable addresses must be aligned on a 4-byte boundary.

Command Specifications:

Numeric addresses are entered as strings, and can be specified in hexadecimal (leading 0x), octal (leading 0) or decimal per the standard C *scanf* conversions. Do not put any spaces before the symbol name. Symbolic addresses defined in a C program must have a leading underscore appended, per standard C compiler name translation conventions. Symbolic addresses must be present in the symbol table of the executable file (*.out) in order to be accessible. Symbolic addresses can include an optional offset, such as *_MyVar+10*. The offset value is multiplied by 4 and then added to the symbolic address, providing the equivalent of array element access.

All Matlab-variables used as data transfer arguments to the commands (**X** in the descriptions below) must be double-precision. Data is transferred to and from the variables in column-row order. Data being transferred to integer DSK variables will be rounded towards 0.

Command argument types are denoted as follows:

- COUNT numeric argument
- 'arg' text argument
- X** Matlab double-precision variable

Command	Syntax	Description
Initialize DSK	C8X_DEBUG('init', PORT_NUMBER, BAUDRATE_INDEX)	Initializes the C8X_DEBUG.mexwXX software and the interface to the DSP board. The PORT_NUMBER argument is used to specify which COM port will be used for the connection (i.e. 1-256). The BAUDRATE_INDEX ARGUMENT must be set to match the baud rate of the DSP board. The possible values are listed at the end of this document. This command must be executed before any other commands.
Get Version	C8X_DEBUG('version')	Displays the C8X_DEBUG.mexwXX version information in the Matlab workspace.

Command	Syntax	Description
Reset DSK	C8X_DEBUG('reset')	Resets the DSP.
Enter Debug Mode	C8X_DEBUG('debug')	As each command is executed the software sends messages to the Matlab workspace for debugging purposes. This slows down the performance of the commands, so it should only be used when necessary.
Leave Debug Mode	C8X_DEBUG('silent')	Turns off the debug messages.
Load/Run COFF File	C8X_DEBUG('run', 'filename')	Loads the specified *.out file to the DSP and starts it running.
Read Unsigned Integers from Numeric Address	C8X_DEBUG('rua', X, 'address', COUNT)	Reads COUNT words of unsigned integer data from the given address on the DSP, and returns it in the double-precision matrix X. The variable X must be the correct size for the block size selected.
Read Unsigned Integers from Symbolic Address	C8X_DEBUG('rus', X, 'symbol_name<+offset>', COUNT)	Reads COUNT words of unsigned integer data from the address of the given symbol name on the DSP, and returns it in the double-precision matrix X. The variable X must be the correct size for the block size selected. The *.out file must have symbolic information included and the symbol defined for this command to succeed.
Read Signed Integers from Numeric Address	C8X_DEBUG('ria', X, 'address', COUNT)	Reads COUNT words of signed integer data from the given address on the DSP, and returns it in the double-precision matrix X. The variable X must be the correct size for the block size selected.
Read Signed Integers from Symbolic Address	C8X_DEBUG('ris', X, 'symbol_name<+offset>', COUNT)	Reads COUNT words of signed integer data from the address of the given symbol name on the DSP, and returns it in the double-precision matrix X. The variable X must be the correct size for the block size selected. The *.out file must have symbolic information included and the symbol defined for this command to succeed.
Read Floating-Point Values from Numeric Address	C8X_DEBUG('rfa', X, 'address', COUNT)	Reads COUNT words of floating-point data from the given address on the DSP, and returns it in the double-precision matrix X. The variable X must be the correct size for the block size selected.
Read Floating-Point Values from Symbolic Address	C8X_DEBUG('rfs', X, 'symbol_name<+offset>', COUNT)	Reads COUNT words of floating-point data from the address of the given symbol name on the DSP, and returns it in the double-precision matrix X. The variable X must be the correct size for the block size selected. The *.out file must have symbolic information included and the symbol defined for this command to succeed.
Write Unsigned Integers to Numeric Address	C8X_DEBUG('wua', X, 'address', COUNT)	Writes COUNT words of unsigned integer data to the given address on the DSP. The data is obtained from the double-precision matrix X. The variable X must be the correct size for the block size selected.
Write Unsigned Integers to Symbolic Address	C8X_DEBUG('wus', X, 'symbol_name<+offset>', COUNT)	Writes COUNT words of unsigned integer data to the address of the given symbol name on the DSP. The data is obtained from the double-precision matrix X. The variable X must be the correct size for the block size selected. The *.out file must have symbolic information included and the symbol defined for this command to succeed.

Command	Syntax	Description
Write Signed Integers to Numeric Address	C8X_DEBUG('wia', X , 'address', COUNT)	Writes COUNT words of signed integer data to the given address on the DSP. The data is obtained from the double-precision matrix X. The variable X must be the correct size for the block size selected.
Write Signed Integers to Symbolic Address	C8X_DEBUG('wis', X , 'symbol_name<+offset>', COUNT)	Writes COUNT words of signed integer data to the address of the given symbol name on the DSP. The data is obtained from the double-precision matrix X. The variable X must be the correct size for the block size selected. The *.out file must have symbolic information included and the symbol defined for this command to succeed.
Write Floating-Point Values to Numeric Address	C8X_DEBUG('wfa', X , 'address', COUNT)	Writes COUNT words of floating-point data to the given address on the DSP. The data is obtained from the double-precision matrix X. The variable X must be the correct size for the block size selected.
Write Floating-Point Values to Symbolic Address	C8X_DEBUG('wfs', X , 'symbol_name<+offset>', COUNT)	Writes COUNT words of floating-point data to the address of the given symbol name on the DSP. The data is obtained from the double-precision matrix X. The variable X must be the correct size for the block size selected. The *.out file must have symbolic information included and the symbol defined for this command to succeed.

Supported *BAUDRATE_INDEX* values:

- 0 OMAP-L138 at 115200 baud or DSK6XXXHPI daughtercard serial interface
- 1 OMAP-L138 at 230400 baud
- 2 OMAP-L138 at 460800 baud
- 3 OMAP-L138 at 921600 baud or DSK6XXXHPI daughtercard USB interface