

MC33816 Developer Studio User's Guide

The screenshot shows the MC33816 Developer Studio interface. The main window displays a C code file for 'MC33816 example - 4 injectors control with Peak & Hold'. The code includes comments about Freescale's rights and responsibilities, and defines variables for boost phase current values and peak duration times. The right side of the interface shows the assembly output for the code, generated by SPIGen. The assembly output includes comments for opamp gain settings and various memory addresses for loading configuration data.

```

MC33816 Developer Studio - KIT33816AEEVM_Demo_Hands_On_Training

File Settings Tools Help
File MC33816 Developer Studio - KIT33816AEEVM_Demo_Hands_On_Training

Channel 1 Channel 2
Copyright (c) Freescale 2013
* File Name: MC33816.ch1.dfi
* Current Revision: 1.0
* Purpose: MC33816 example - 4 injectors control with Peak & Hold
* Description: MC33816 Channel 1 main function provide peak and hold current
* profile for UcOChi and UcIChi
* File created by: Freescale Analog, Tempe
* Freescale Analog, Toulouse
* UPDATE HISTORY
* REV AUTHOR DATE DESCRIPTION OF CHANGE
* --- -----
* 0.1 b39463 2012/05/16 - initial coding
* 1.0 thoi01 Not released - reworked/optimized by same functionality
* Freescale reserves the right to make changes without further notice to any product herein to improve reliability, function, or design. Freescale does not assume any liability arising out of the application or use of any product, circuitry and its patent rights, neither does it convey any license under its patent rights to others. Freescale products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support life, or for any other application in which the failure of the Freescale product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale products for any such intended or unauthorized application, Buyer shall indemnify and hold Freescale and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees, arising out of or directly or indirectly, any claim of personal injury or death associated with such unanticipated or unauthorized use, even if such claim alleges that Freescale was negligent regarding the design or manufacture of the part. Freescale and the Freescale logo are registered trademarks of Freescale Ltd.
* ## Channel 1 - uCore0 controls the injectors 1 and 2 ##
* ## Variables declaration ##
* Note: The Ihold_tot variable defines the current profile time out. The active STARTX pin is expected to
* ## Initialization phase ##
init0:    stgn gain12.6 ssac;
          ldjrl eoinj0;
          ldjrl idle0;
          owei jrl_start row1;
          !!

stgn : set opamp gain
Description : set the gain of an operational amplifier used to measure the current flowing through the actuator
- value define the desired gain
- otarget identifies the sensitive opamp
the operation is successful only if the sequencer has the right to control the related current measure block (curAccess register)
Syntax : stgn Value Otarget;
Value
gain5.8
gain8.7
gain11.6
gain19.2
gain19.3
gain19.4
gain5.79
gain8.68
gain12.53
gain19.25
Otarget
ssac
osac
ssoc
osoc
Build
Build Output
Starting build...
Deleting previous output files
launching assembler for FREESCALE MC33PT2000
Using default Syntax file: config syntax.xml Version: 1.71 for Device: 33816
compiling source from source file: C:\Users\B39463\Documents\Devices\MC33816\Projects\KIT33816AEEVM_Demo_Hands_On_Training\source\MC33816.ch1.dfi
Cipher with NFSR: B6D552EFC950936FABA, LFSR : 885B4CC5BEE23AF88705
Binary CRC MSB : 011010000111000, LSB : 1011000101101010
Hex CRC MSB : 0x6338, LSB : 0xB16A
Assembly completed with success.

launching assembler for FREESCALE MC33PT2000
Using default Syntax file: config syntax.xml Version: 1.71 for Device: 33816
compiling source from source file: C:\Users\B39463\Documents\Devices\MC33816\Projects\KIT33816AEEVM_Demo_Hands_On_Training\source\MC33816.ch2.dfi
Cipher with NFSR: B6D552EFC950936FABA, LFSR : 885B4CC5BEE23AF88705
Binary CRC MSB : 000011000111010, LSB : 11011001110001
Hex CRC MSB : 0xC07A, LSB : 0xEEE7
Assembly completed with success.

Successfully created SPIGen load file.
C:\Users\B39463\Documents\Devices\MC33816\Projects\KIT33816AEEVM_Demo_Hands_On_Training\SPIGenMC33816Files.txt
Checksum and code with set for ch1 and ch2.
Channel 1 uc0 entry point register was set.
Channel 1 uc1 entry point register was set.
Channel 2 uc0 entry point register was set.

Launch SPIGen

```

Contents

1 Introduction	3
2 Editing Microcode with MC33816 Developer Studio	4
3 Using MC33816 Developer Studio.....	6
4 Revision History.....	14

1 Introduction

MC33816 Developer Studio is a Windows application created to streamline the development of microcode for the MC33816 Programmable Solenoid Controller.

MC33816 Developer Studio allows the user to create, edit and compile source code for the MC33816. The built-in source code editor features syntax highlighting, auto-complete functionality for instructions and parameters, and context sensitive help. Register and data RAM configurations can also be created and saved.

MC33816 Developer Studio allows the creation of projects for different MC33816 applications. All Register, DRAM, source code, and configuration settings can be saved to a project file and retrieved for later use. After an initial project set up, your settings can be loaded and you can start working immediately. This can be done for multiple MC33816 projects allowing easy switching between projects.

1.1 Installing MC33816 Developer Studio on Your Computer

MC33816 Developer Studio is compatible with Microsoft Windows 7 and later operating systems. It is compatible with both 32-bit and 64-bit versions of Windows. Download the installer from the MC33816 web site and run it on the computer to install the program.

1.2 Uninstalling MC33816 Developer Studio

MC33816 Developer Studio can be uninstalled in Windows in the Control Panel. To access the Control Panel, click the Start button, select "Settings", and select "Control Panel".

Click on "Programs and Features". Find "33816 Developer Studio" in the list of programs, then click "Uninstall". This invokes an uninstall wizard to begin the uninstall process.

If the uninstall wizard was not able to delete some part of the program, there are other files in the program's directory. If any configuration files were saved, this could prevent the uninstall wizard from completely removing the software.

To complete the uninstall:

1. Wait for completion of the uninstall wizard.
2. Locate where MC33816 Developer Studio was installed.
3. Delete the MC33816 Developer Studio folder and all of its contents.

2 Editing Microcode with MC33816 Developer Studio

2.1 Microcode Editor



Figure 1. Microcode Editor Window

2.2 Using the Microcode Editor

The microcode editor window allows code entry, with basic features (cut, paste, find, etc.). There is one tab for each channel on the MC33816. The editor also provides the following advanced features:

- Custom Syntax Highlighting: Instructions are highlighted in blue, comments in green and numbers in violet.
- Content-aware auto-completion: This is an advanced auto-completion feature that proposes values for instructions and parameters by analyzing the source code.
- Column editing: By pressing the Alt key, it is possible to select multiple columns, and to cut, paste, and edit in column mode.
- Row Numbering: For each row, the text line number and the code line number are shown. The code line number is the address of the specific instruction in the Code RAM

2.3 Context Sensitive Help

When the mouse cursor is placed on an instruction in the microcode window, a description of the command and a list of parameters displays in the help window as shown in [Figure 2](#). A list of acceptable values for each parameter is also shown.

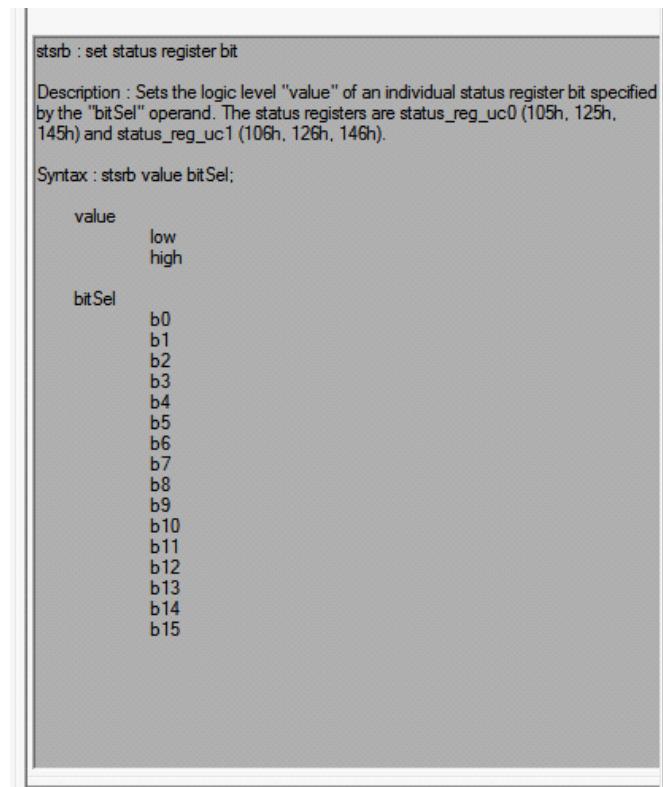


Figure 2. Context Sensitive Help Window

3 Using MC33816 Developer Studio

3.1 Device Configuration Window

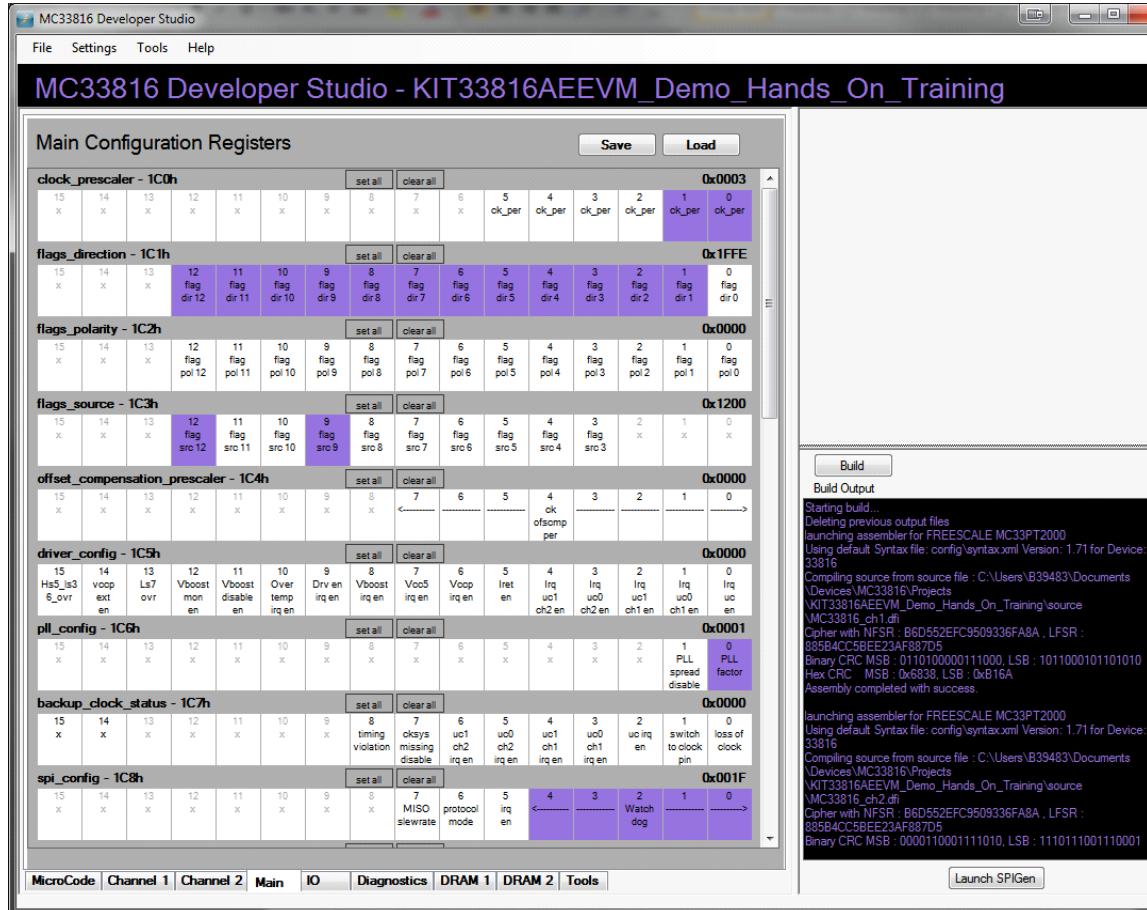


Figure 3. MC33816 Developer Studio Main Window

3.1.1 Register Configuration Window

The main application window has three sections. The first section contains the register configuration section. It contains five register tabs, one tab for each of the register groups on the MC33816. The register groups are Main, Channel 1, Channel 2, IO, and Diagnostic. Click on the desired tab to display the register group. Functional details on the registers are in the MC33816 data sheet available at the Freescale website (www.freescale.com/psc).

Clicking on a bit toggles the current state of the bit. When the bit appears in a non-white color, it is set (1), and when the bit appears white, it is clear (0). When the mouse pointer hovers over a bit, a tool tip window displays the function of that bit. If there is a group of bits for a certain function, only the rightmost bit in the group has the explanation tool tip window. The register configuration can be saved to, or loaded from, a file using the Save and Load buttons in the top right corner of the register window. These buttons are only active when a valid project is loaded.

3.1.2 DRAM Configuration Window

The DRAM configuration window allows the contents of the Data RAM for channels 1 and 2 to be edited and saved to a file. Click on the DRAM 1 tab to modify and save DRAM values for channel 1. Click on the DRAM 2 tab to modify and save DRAM values for channel 2. All 64 DRAM locations for each channel can be accessed.

The data may be entered in binary, hexadecimal or decimal format. Entering data in any one of these three fields causes the other two fields to automatically update to reflect the equivalent value.

A label can be specified which is an alternate name of the parameter contained in the Data RAM cell. It can be used in the code when reading/writing this cell in place of its address.

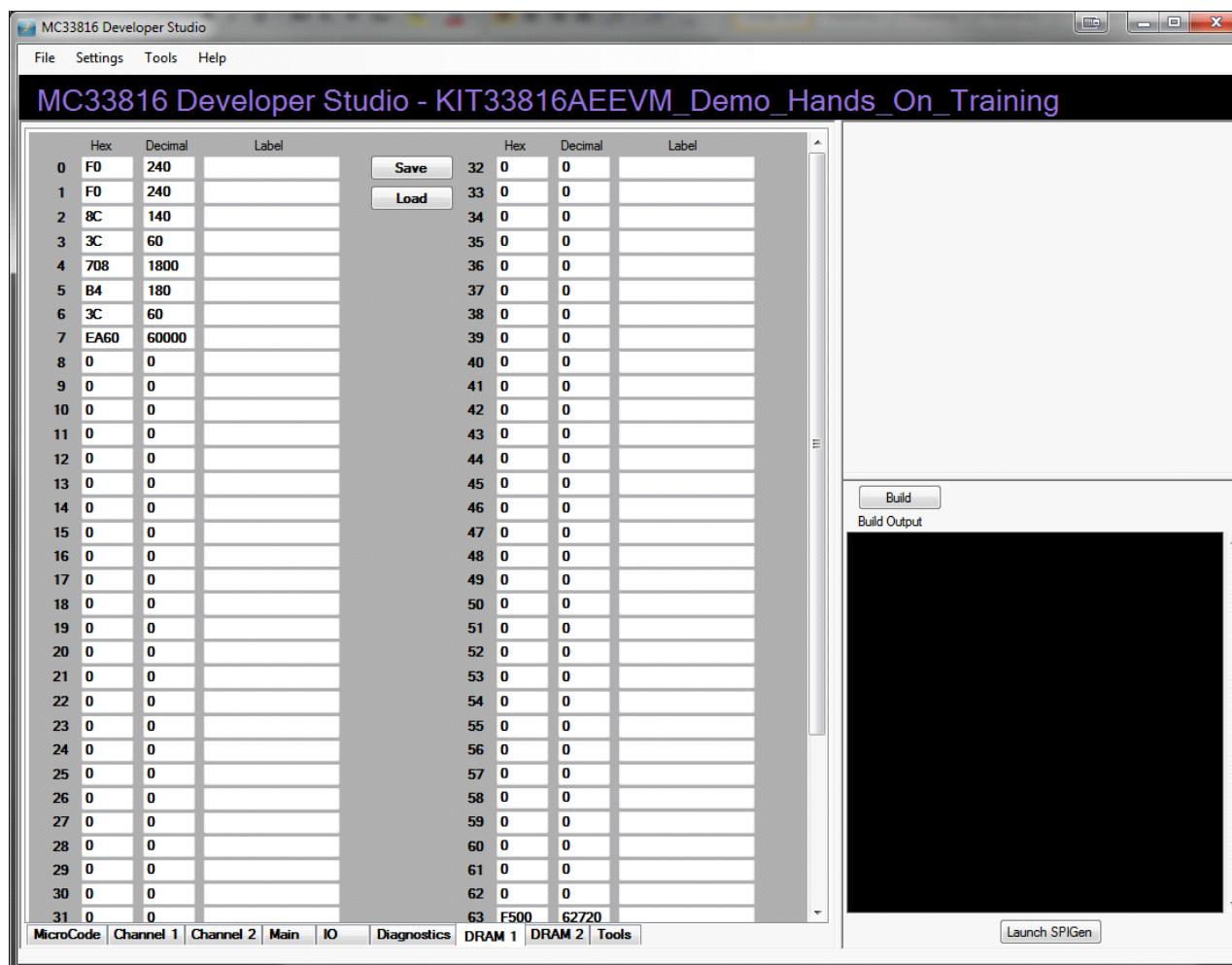


Figure 4. DRAM Window

3.1.3 Tools Window

The Tools window allows easy configuration of the various parameters required to make the MC33816 operational. Parameters can be configured for:

- Current sense threshold of the load
- V_{boost} over current threshold
- Clock/Timer settings
- V_{boost} voltage monitoring and filters

Calculations are performed automatically and their values are stored in the appropriate registers and/or DRAM locations by clicking the Apply button for the specific setting. The Tools window is shown in Figure 5.

3.1.3.1 Current Sense Threshold

This tool calculates the current sense threshold for a load, and stores the DAC value in the selected DRAM location. R_{shunt} is the current sense resistor value that is in series with the load. Gain is the current sense operational amplifier gain value. The gain value used in this tool must match the gain set by the "stgn" instruction in the microcode. After entering the R_{shunt} and gain values, set the DAC value to achieve the desired current threshold for the load. Select the DAC number to be used and click on the "Apply" button next to the DAC number selection. This will set the threshold in the `dac_value` register for the selected DAC. Next, select the DRAM bank and line number and click on the lower "Apply" button to save the DAC value to DRAM where it can be accessed by the microcode.

3.1.3.2 V_{boost} Overcurrent Threshold

This tool calculates the V_{boost} overcurrent threshold. R_{shunt} is the V_{boost} current sense resistor value. Gain is fixed at -2.0. After entering the R_{shunt} value, set the DAC value to achieve the desired overcurrent threshold for V_{boost} . Select the DAC number to be used and click on the "Apply" button next to the DAC number selection. This sets the threshold in the `dac_value` register for the selected DAC. Next, select the DRAM bank and line number and click on the lower "Apply" button to save the DAC value to DRAM where it can be accessed by the microcode.

3.1.3.3 CLK/Timer Calculator

This tool calculates a timer value that can be used in the microcode based on the PLL factor and `ck_per` values. After setting the PLL factor and `ck_per` values, enter the required period in microseconds or milliseconds, and the timer value will be calculated. Click on "Apply" to save the timer value to the selected DRAM bank and line number. This timer value can be accessed by the microcode as needed.

3.1.3.4 V_{boost} Voltage Monitor

This tool calculates the V_{boost} voltage threshold based on the DAC value entered. Clicking on "Apply" writes the threshold value to the `boost_dac` register on the IO tab. The DAC value must be set to a value of 0 to 225.

3.1.3.5 Current/ V_{boost} Filter

This tool calculates the current filter length based on the time and clock values entered. The filter type (reset or decrease) can also be set. Clicking on "Apply" writes the filter length and type to the selected `current_filter` register on the IO tab. The "Time" value must be set to a value from 334.0 ns to 5334.0 ns.

3.1.3.6 V_{boost} Filter

This tool calculates the V_{boost} filter length based on the time and clock values entered. The filter type (reset or decrease) can also be set. Clicking on "Apply" writes the filter length and type to the `boost_filter` register on the IO tab. The "Time" value must be set to a value from 0.334 us to 682.6 μ s.

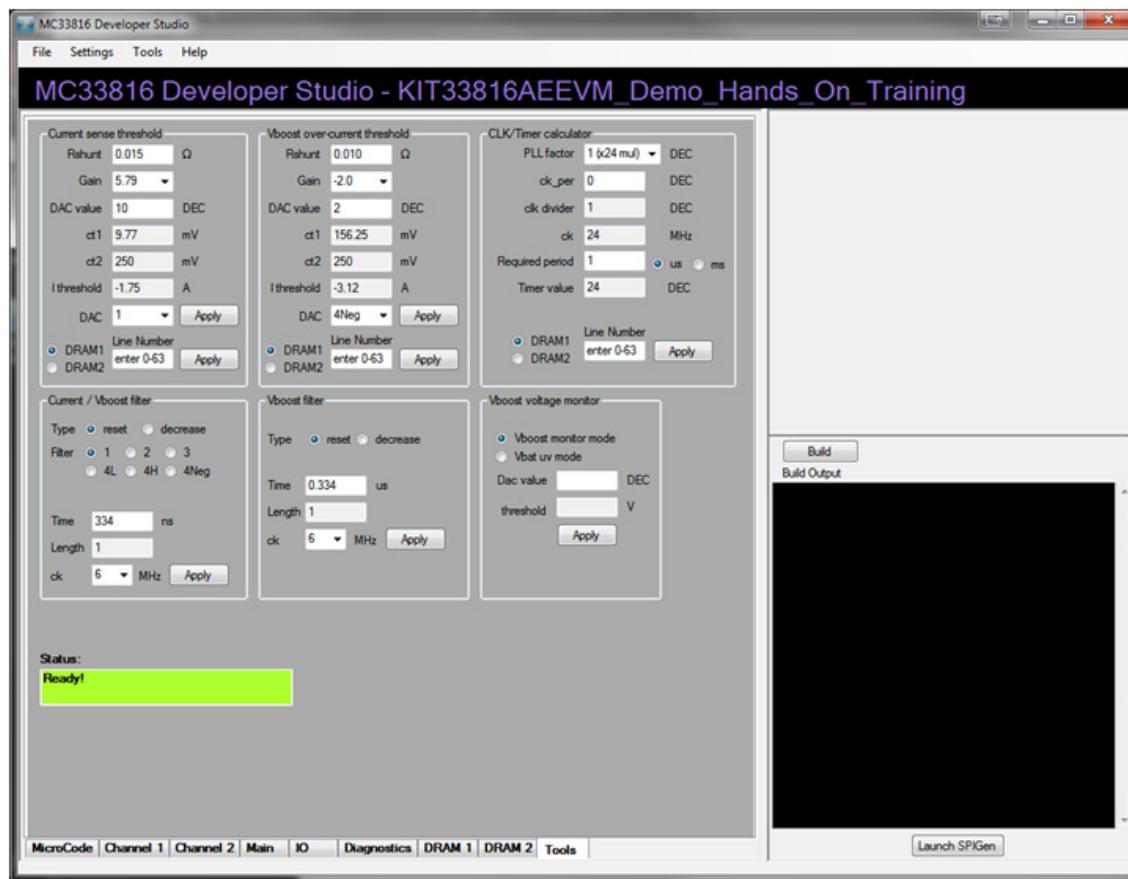


Figure 5. Tools Window

3.2 Projects

3.2.1 Creating a New Project

To create a new project, select "New Project" from the "File" menu. The Create a New MC33816 Project window appears (see [Figure 6](#)). Enter the name of the project and the location where the project is created. Existing source code files can be selected using the Browse buttons. If existing files are selected, they are copied to the new project location and the original files remain intact. To create new source code files, check the "Generate New Source Files" check box. New source code files contain descriptive file headers and an entry point label where microcode for the MC33816 can be written.

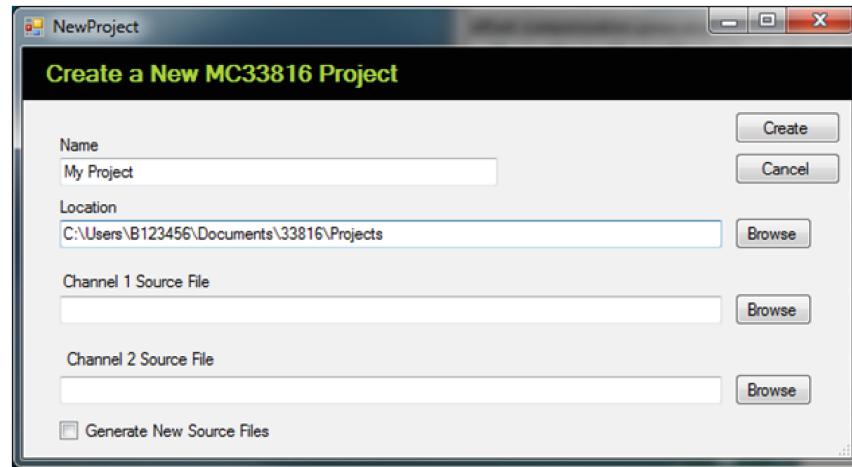


Figure 6. MC33816 Developer Studio New Project Dialog

3.2.2 Opening an Existing Project

To open an existing project, select "Open Project" from the "File" menu. An open project dialog box appears. Browse to the location of the project and select the project file name. Click on the Open button and the project loads into the MC33816 Developer Studio IDE.

Recent projects can be opened quickly by selecting "Recent Projects" from the "File" menu. A list of recently opened projects appears. Click on the project name to load it into the MC33816 Developer Studio IDE.

3.3 Building the Code

After the register, DRAM, and source code files are complete, compile the source code to create the encrypted binary files for loading into the MC33816. To start the build, click on the Build button located below the Project Files window. The code is assembled and the output from the assembler is displayed in the Build Output window (see [Figure 7](#)).

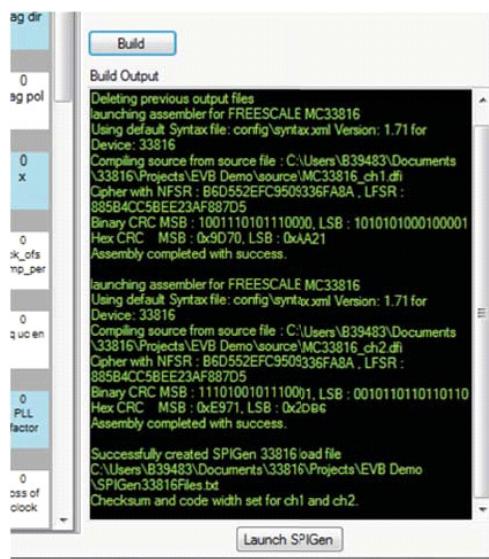


Figure 7. Build Button and Build Output Window

After a successful build, the MC33816 Developer Studio automatically sets the correct values for code width and checksum values in the channel 1 and channel 2 register files.

MC33816 Developer Studio also generates a file used to load the register, DRAM, and microcode file names into the SPIGen application Microcode page (see [Figure 8](#)). This Microcode page can then be used to load the MC33816 device using the SPI bus interface.

The SPIGen application and associated documentation can be found at www.freescale.com/spigen.

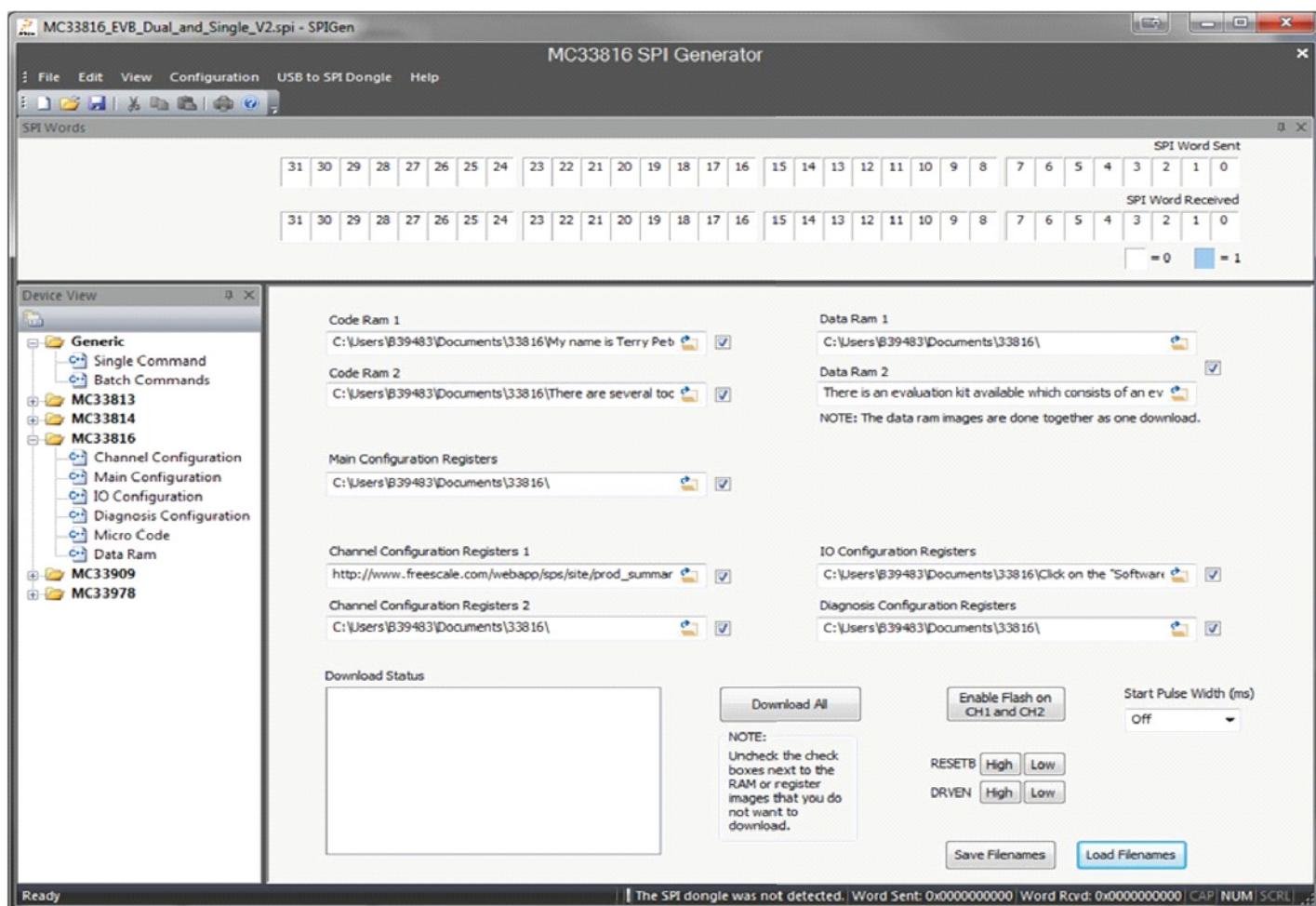


Figure 8. SPIGen MC33816 Microcode Page

3.4 Generating Embedded Source Code Files

The MC33816 Developer Studio generates C language source code files used in an embedded microcontroller design to load the code RAM, data RAM, and register data into the MC33816 device.

The C language files it generates are:

- A file of data arrays for the Code RAM, Data RAM, and register data
- A header file for the data array file
- A file with code to load the MC33816 device using the SPI protocol

These files are generated by selecting "Generate MC33816 Startup Code" from the "Tools" menu. The generated files are located in the "sample_code" directory under the project currently open in MC33816 Developer Studio.

3.5 Exporting a Project to the PSC Simulator

Once project development is complete, the MC33816 Developer Studio can export the project files to a project used by the PSC Simulator. To export the project files, load a project and make sure it builds successfully. Select "Export Project Files to PSC Simulator" from the "Tools" menu. The following message window appears:

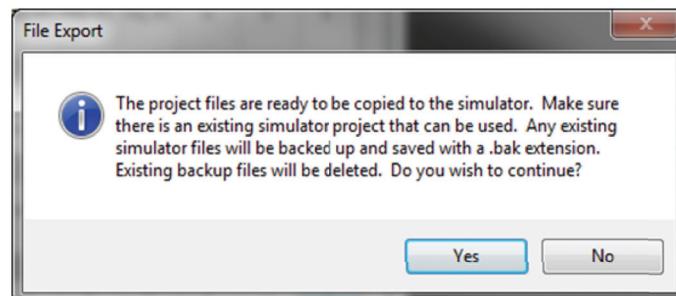


Figure 9. File Duplication Verification Window

A PSC Simulator project must have already been created to export the project. The PSC Simulator software and user's guide can be downloaded from www.freescale.com/psc.

Click on "Yes" to export the files or "No" to cancel. If "Yes" was selected, you are prompted to select a simulator project folder. Select the desired folder and click on "OK". First, the existing simulator files are renamed with a ".bak" extension for backup purposes. If a set of backup files already exists, they are deleted before the current files are backed up. Next the project files from MC33816 Developer Studio are copied to the simulator project folder in the correct format. Open and run the project using the PSC Simulator application.

3.6 Setting Entry Point Labels

Entry point and interrupt address values can be set automatically in the appropriate registers if address labels are defined prior to building the project. Labels are defined by selecting "Set Microcode Address Labels" from the "Tools" menu. When selected, the following dialog box appears:

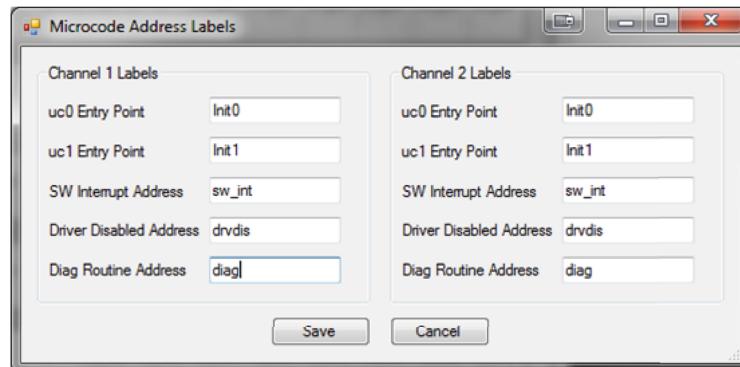


Figure 10. Microcode Address Labels Dialog

The five entry point and address labels must be unique for a given channel. The channels run in separate code spaces allowing the same set of labels to be used on both channels if desired, as shown in Figure 10.

The labels entered require a corresponding label in the microcode for the address register to set correctly. If the label value is left blank, or does not have a corresponding label in the microcode, the register is not set.

3.7 Generating CSV Files

MC33816 Developer Studio generates comma separated value (CSV) files containing register and DRAM data. These files can be used in spreadsheets, incorporated into reports, or used to document project configurations. Select "Generate CSV Files for Registers and DRAM" from the "Tools" menu to generate a set of CSV files for a project. A valid project must be loaded to use this function. A total of seven files are generated, one file for each of the five register groups, and two files for the DRAM values. Each register file contains the register name, the address, and the register value in binary format. Each DRAM file contains the DRAM address, and the value in both hexadecimal and decimal formats. All files include a header with a label for each column.

3.8 Setting the Color Scheme

To configure the color scheme settings in MC33816 Developer Studio, click "Color Scheme..." in the "Settings" menu.

4 Revision History

Revision	Date	Description
1.0	1/2015	<ul style="list-style-type: none">Initial release
2.0	10/2015	<ul style="list-style-type: none">Added a section to describe the new micro code editor that was added to the Dev. Studio software

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off.

© 2015 Freescale Semiconductor, Inc.