CP3UB17/CP3CN17 Evaluation Kit

User's Guide

Revision 1.5

National Semiconductor
The Sight & Sound of Information

420012131-101A

Preface

Scope of This Document

This document is a quick introduction to the Software Evaluation Kit (SEK) and CP3UB17/CP3CN17 Evaluation Board (CP3-DB-xx1x) for the CP3UB17/CP3CN17 connectivity processors.

This document is written for the Integrated Development Environment (IDE) based on the winIDEA project manager/file editor/debugger from iSYSTEM. winIDEA is used to call a C compiler tool chain developed by National Semiconductor from the GNU tools.

Future software releases may operate differently than the software described in this book.

Related Documentation

CompactRISC CR16C Programmer's Reference Manual—This is the authoritative reference for the architecture of the CR16C CPU. Compiler writers and assembly-language programmers should consult this document for detailed information about the instruction set. After installation of the software on the evaluation kit CD-ROM, the default location for the Programmer's Reference Manual is **C:\National_SEK_X_Y\docs\Prog_16C.pdf**, in which X.Y is the version number of the release.

Preface

CP3UB17/CP3CN17 Data Sheets—These are the full data sheets for these devices in the CP3000 family of connectivity processors. Refer to these documents for information about the on-chip peripheral devices, signal descriptions, package pinout, and electrical specifications. After installation of the software on the CD-ROM, the default location for the folder containing the data sheets is **C:\National_SEK_X_Y\docs**, in which X.Y is the version number of the release.

CP3UB17/CP3CN17 ISP and CRISP User's Guide—A comprehensive guide to implementing the In-System Programming (ISP) capability of the CP3UB17/CP3CN17 devices. This is a facility for programming the flash memory over a serial interface. CRISP is a utility for controlling the ISP interface from a host PC.

winIDEA Software Manual—This is the manual for the winIDEA IDE and debugger. It can be downloaded as a PDF file from the iSYSTEM web site at http://www.isystem.com. Select Support -> Downloads, select Documents, then click on winIDEA Software Manual.

Notational Conventions

Commands selected from menus are shown as "File -> New", which represents the "New" command selected from the "File" menu.

High-level language and assembly code, command lines, and macro file statements are shown in the Courier font, for example:

```
add r2,r3;
```

When a single command or statement is too long to fit on one line, a backslash (\) is used to indicate continuation on the following line, for example:

```
echo "ERROR 137 -- square root domain error, \
try using a positive number"
```

Revision History

Revision	Release Date	Summary of Changes
1.5	9/17/03	Original release.

Preface

Table of Contents

Table of Contents

Table of Contents

List of Figures

List of Figures

List of Tables

List of Tables

Overview

1

The CP3UB17/CP3CN17 Evaluation Kit provides a complete hardware and software foundation for developing applications on CP3UB17/CP3CN17 devices. The Integrated Development Environment (IDE) is winIDEA, from iSYSTEM. The C compiler tool chain is a port of the GNU tools developed by National Semiconductor. The IDE and software tools are provided with peripheral software and example applications on the CP3UB17/CP3CN17 Evaluation Kit CD-ROM. The following chapters provide concise instructions to help you get started as quickly as possible.

Chapter 2—Evaluation Kit Components. Lists the parts of the evaluation kit, so you can make sure there are no missing parts.

Chapter 3—Developer Registration. National Semiconductor provides a web site for CP3UB17/CP3CN17 developer's that provides the latest information and software releases.

Chapter 4—Software Installation. Refers to the vendor documentation for installing the winIDEA project manager and debugger and the National Semiconductor software distribution.

Chapter 5—Hardware Installation. Describes the hardware setup procedure for the CP3UB17/CP3CN17 Evaluation Board and the interface to the host PC.

Chapter 6—Configuration Settings. Use the Configurator utility to describe the target hardware configuration.

Chapter 7—Example Files. Describes the files loaded by the CP3UB17/CP3CN17 Evaluation Kit CD-ROM for the programming example used in this book.

Chapter 8—Getting Started with winIDEA. Describes the winIDEA user interface and how to create a workspace.

Chapter 9—Building an Application. Describes how the winIDEA IDE calls an external makefile to build an application.

Chapter 10—Downloading Applications. Describes how to download application code to the evaluation board using winIDEA.

Chapter 11—Running and Debugging. Describes how to run the **CPDemo** example, install the USB driver for the example, and run the host-side interface for the example. Briefly introduces some of the debugging features of winIDEA.

Chapter 12—Evaluation Board. Detailed description of the CP3UB17/CP3CN17 Evaluation Board.

The Integrated Development Environment (IDE) for the CP3UB17/CP3CN17 devices is shown in Figure 1-1. winIDEA provides the project manager, editor, and debugger. The tool chain consists of the C compiler, loader, and linker. The evaluation board interfaces to the host PC through its parallel port.

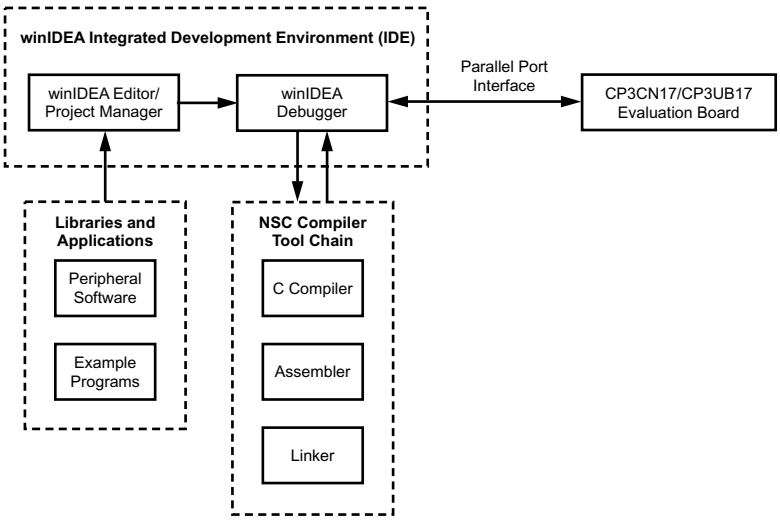


Figure 1-1. Integrated Development Environment

1.1 Minimum System Requirements

The minimum system requirements for running the tool chain are listed in Table 1-1.

Table 1-1. System Requirements

Resource	Requirement
CPU	Intel Pentium or better
RAM	64M
Free disk space	100M
Operating system	Windows 98/NT/2000/XP

Evaluation Kit Components

2

When you unpack the evaluation kit, check to make sure there are no missing pieces. The kit contains the components listed in Table 2-1.

Table 2-1. CP3UB17/CP3CN17 Evaluation Kit Components

Component	Description
CP3UB17/CP3CN17 Evaluation Board	CP3UB17/CP3CN17 system providing CPU, 2M external flash memory, 1M external SRAM memory, USB interface, CAN interface, serial port, and external CODEC.
Power Supplies	Two wall-mount power supplies, one for European use and one for U.S. use.
Parallel Port Cable	Used to connect the host PC to the evaluation board.
CP3UB17/CP3CN17 Evaluation Kit CD-ROM	Software distribution disk for CP3UB17/CP3CN17 documentation, compiler tool chain, peripheral drivers, embedded operating system, and example applications.
CP3UB17/CP3CN17 Evaluation Kit User's Guide	This manual.

Evaluation Kit Components

Registration and Support

3

To receive the best product support, it is recommended that each owner of the CP3UB17/CP3CN17 evaluation kit register with National Semiconductor. The registration process is easy:

- **Open an Account on National Semiconductor's Web Site.** A password will be sent to you by e-mail. Your user name is your e-mail address.
- **Sign-On to the Web Site.** Use your e-mail address and password to enter the National web site.

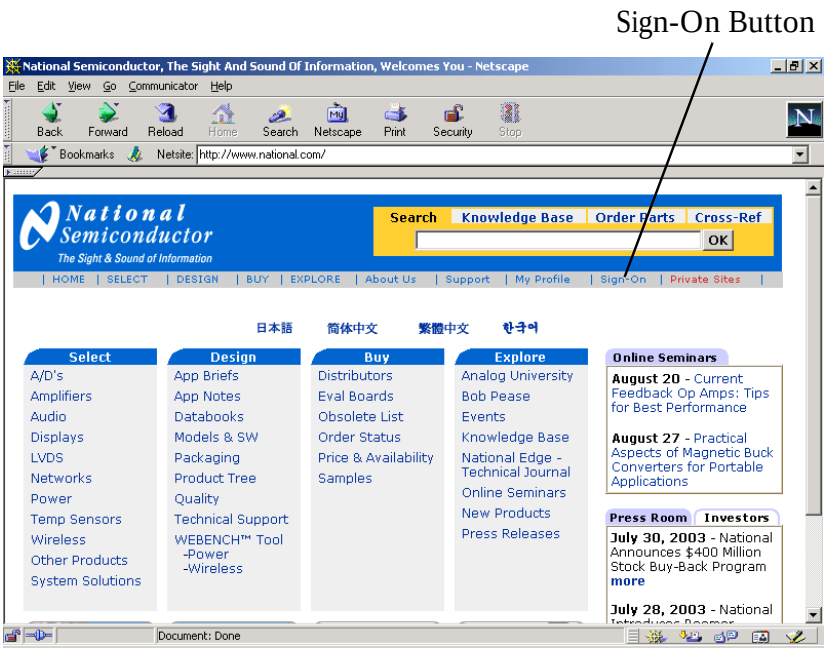
After registration, you can access the following services:

- **Software Updates.** Browse the **www.national.com/appinfo/cp3000** web page to download updates to your software release. You will need the serial number supplied with the evaluation kit to access these updates, so **do not lose this serial number**.
- **Product Support.** On the **www.national.com** web page, click the Support button for additional product support.

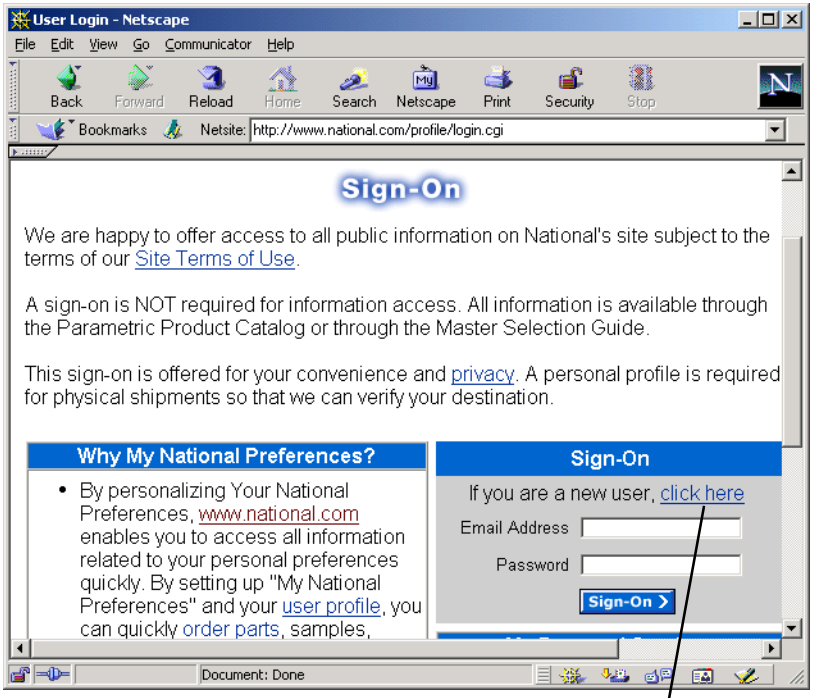
3.1 Registration and Sign On

The following procedure opens an account on the National web site and obtains access to the CP3000 Developer's area:

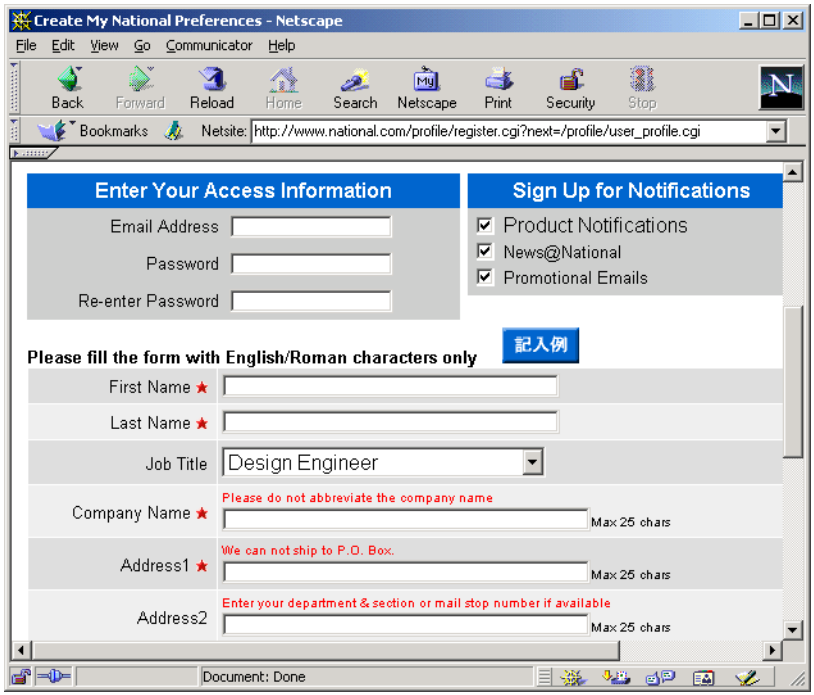
1. **Browse to the Main National Web Page.** Go to the page at **http://www.national.com**. Click on the Sign-On button.



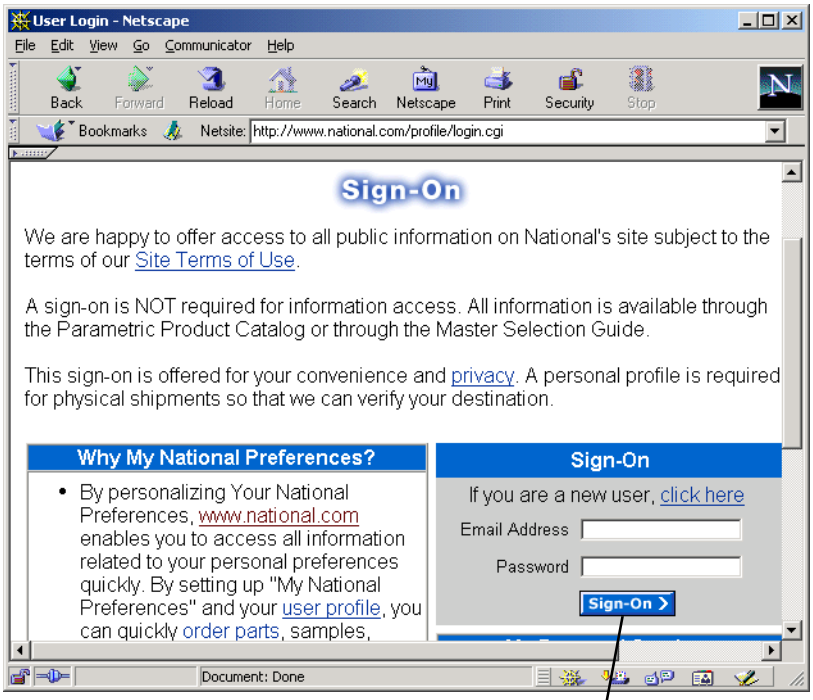
2. **Create a New User Account.** Click on the Click Here button.



3. **Fill Out Account Request Form.** Enter all required information, then click on the Create button at the bottom of the page.



4. **Activate Your Account and Sign On.** An e-mail will be sent to you with instructions for activating your account. After activating your account, browse to the National sign-on page and enter your user name and password, then click the Sign-On button



Registration and Support

Registration and Sign On

Software Installation

4

Check the CP3000 area on the National Semiconductor web site (**www.national.com / appinfo / cp3000**) and the Release Notes supplied with the evaluation kit to be sure you are using the latest versions of the software tools.

To install the software, load the evaluation kit CD-ROM, click through the license agreement (if you agree to its terms), then click on the Software Installation button in the Main Navigation Screen. This brings up the view shown below.

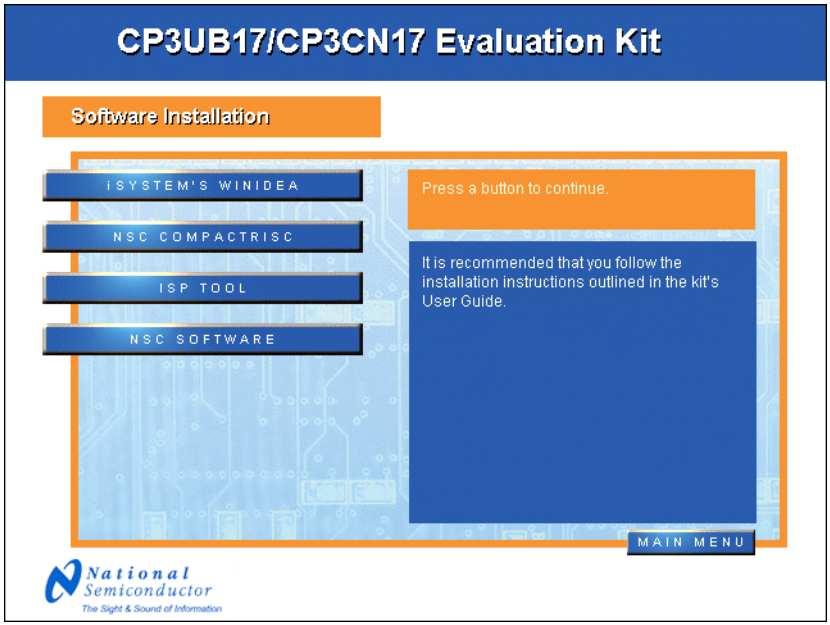


Figure 4-1. Software Installation Screen

Software Installation

Software installation has four main parts, which correspond to the four buttons on the screen shown in Figure 4-1:

- **winIDEA Project Manager and Debugger**—the integrated development environment (IDE) for compiling, downloading, and debugging application programs.
- **NSC CompactRISC**—the C compiler tool chain developed by National Semiconductor from the GNU tools. winIDEA can be configured to call this tool chain for compiling application programs. (Alternatively, winIDEA support is available for a C compiler tool chain from IAR Systems.) A license key is provided in the evaluation kit which is needed to install the NSC tools.
- **ISP Tool**—the CRISP tool for downloading application code to the evaluation board over a serial connection. See the CP3UB17/CP3CN17 ISP and CRISP User's Guide for more information.
- **NSC Software**—application examples and template files for creating new applications. The template files configure winIDEA to call the NSC tool chain. The top-level structure of the NSC software installation is shown in Figure 4-2.

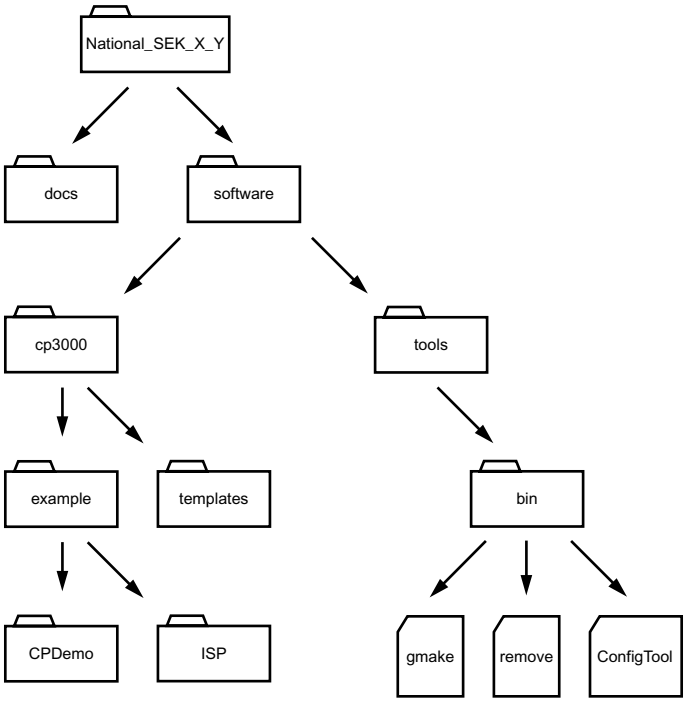


Figure 4-2. Organization of Files Loaded by the NSC Software Installation

The **bin** folder contains these executable files:

- **gmake**—the GNU make utility.
- **remove**—a file deletion utility called by makefiles.
- **ConfigTool**—a utility which defines target configuration settings used during compilation (CPU type, board type, etc.).

The **templates** folder contains winIDEA files which are replicated to create a new project. There are two sets of template files which configure winIDEA to call the NSC tool chain:

- **template_NS.jrf, template_NS.QRF**—template files to compile applications for conventional download using winIDEA.
- **template_NS_ISP.jrf, template_NS_ISP.QRF**—template files to compile applications for download through the In-System Programming (ISP) interface. See the **CP3UB17/CP3CN17 ISP and CRISP User's Guide** for more information about the ISP.

The **example** folder contains the files for the **CPDemo** example program used in this manual and the ISP software described in the **CP3UB17/CP3CN17 ISP and CRISP User's Guide**.

4.1 Program Menu Shortcuts

The software installation adds shortcuts to the Start -> Programs menu. The shortcuts you are likely to use most often are:

- **winIDEA 9.4 -> winIDEA NSC EVB**—enters the winIDEA IDE.
- **CRISP -> CRISP**—enters the CRISP tool.
- **National SEK 1.5 -> Configuration Tool**—initializes macros which control compilation for a specific hardware target configuration. This utility **must** be run before you can compile any application programs.
- **National SEK 1.5 -> CPDemo**—user interface utility for controlling the **CPDemo** application example. This utility runs on the host PC and communicates over a serial connection to the **CPDemo** executable code running on the evaluation board.

Software Installation

Program Menu Shortcuts

Hardware Installation

5

Figure 5-1 shows the hardware connections between the host PC and the CP3UB17/CP3CN17 Evaluation Board.

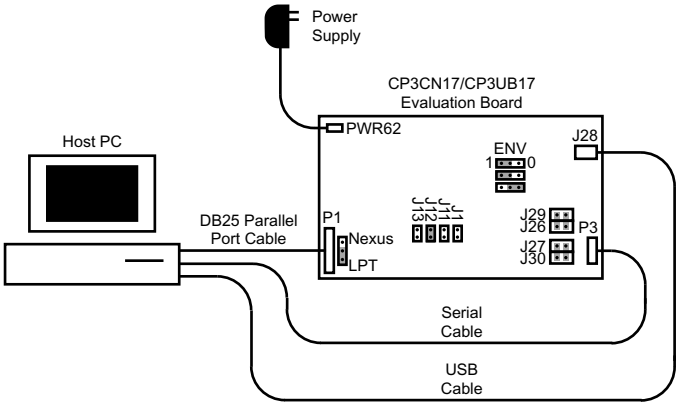


Figure 5-1. Hardware Connections

The following procedure sets up the hardware:

1. **Connect the Parallel Port on the Host PC to the Evaluation Board.** The cable is plugged into the connector for the LPT1 parallel port on the host PC and connector P1 on the evaluation board.
2. **Connect a Serial Port on the Host PC to the Evaluation Board.** The cable is plugged into a connector for the COM1 or COM2 serial port on the host PC and connector P3 on the evaluation board.

3. **Connect a USB Port on the Host PC to the Evaluation Board.** The cable is plugged into a connector for a USB port on the host PC and connector J28 on the evaluation board.
4. **Configure the Clock Source Jumpers.** The clock source is controlled by jumpers J1, J11, J12, and J13, as described in Table 12-7. Select the oscillator module by installing jumper J12. Leave jumper positions J1, J11, and J13 empty.
5. **Configure the ENV Jumpers.** Configure the board for ERE mode, in which the on-chip flash memory and external flash memory are enabled. Jumpers J15 (ENV0) and J16 (ENV1) are installed on the "1" side of the jumper block. Jumper J17 (ENV2) is installed on the "0" side.
6. **Enable the Parallel Port Interface.** The connection used for downloading applications to the evaluation board can be the parallel port interface on connector P1 or the Nexus/JTAG interface on connector JP1. Enable the parallel port interface by installing jumper J25 on the LPT side of the jumper block.
7. **Apply Power to the Evaluation Board.** Plug the power supply into an AC outlet, and plug the power cable into connector PWR62 on the evaluation board.

Configuration Settings

6

There are configuration settings which must be created before the software development environment can be used. These settings define features of the target configuration, such as the CPU type and board type. Once the settings are created, they are used by the compiler tool chain until they are redefined. The settings are global to all projects, not project-specific.

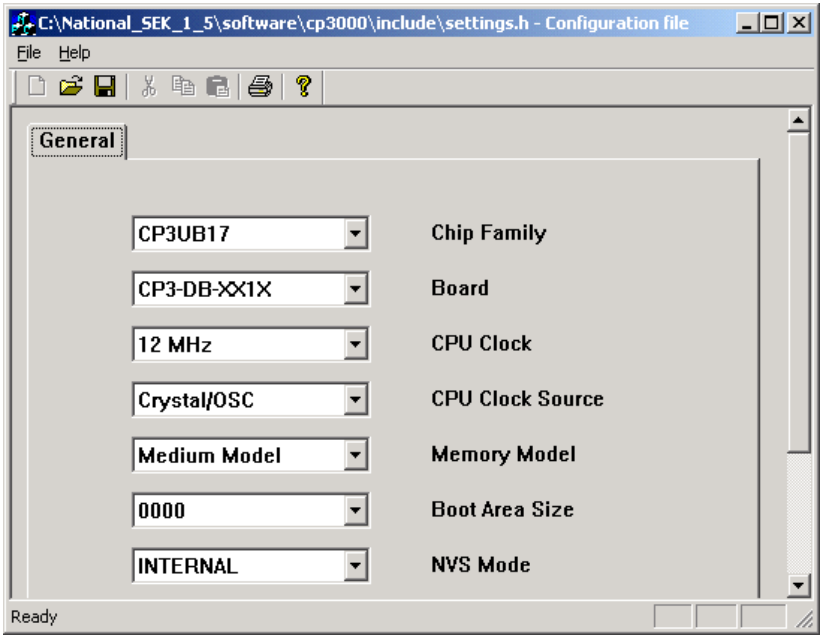
The Configurator program is a simple utility used to define these settings. This program is located in the executable file at **C:\National_SEK_X_Y\software\tools\bin\ConfigTool.exe**, in which X.Y is the version number of the software release.

After the Configurator program is run, two files are created in **C:\National_SEK_X_Y\software\cp3000\include**, called **settings.h** and **settings.mak**. These files must be created before attempting to compile the example programs.

6.1 Define Configuration Settings

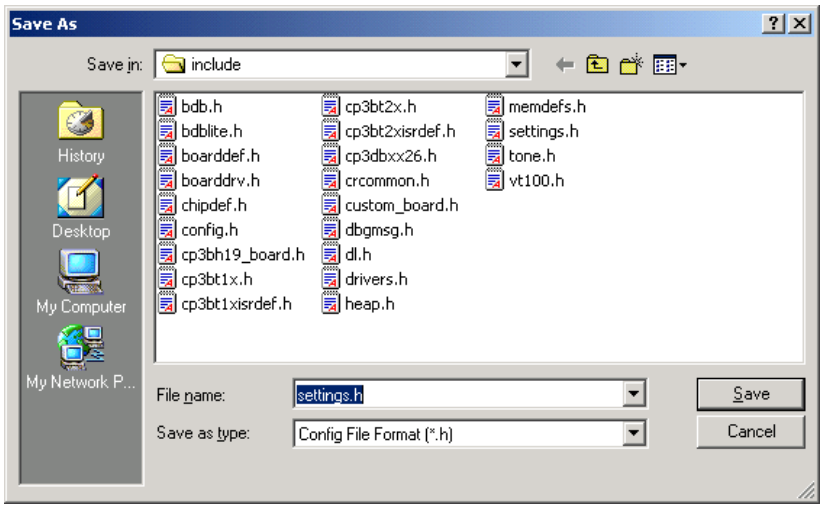
The following procedure selects the configuration settings and saves them in the **settings.h** and **settings.mak** files:

1. **Enter the Config Utility.** Double-click on the file **C:\National_SEK_X_Y\software\tools\bin\Config-Tool.exe**, in which X.Y is the version number of the software release. This opens a new window with the view shown below.



Change any default settings that are different from those shown above.

2. **Select Destination Folder for Saving the settings.h and settings.mak Files.** Select the File -> Save As command, then browse for the **C:\National_SEK_X_Y\software\cp3000\include** folder.



3. **Save the settings.h and settings.mak Files.** Click the Save button.
4. **Exit from ConfigTool.** Select the File -> Exit command.

Configuration Settings

Define Configuration Settings

7 Example Files

To demonstrate using the software tools, an example program **CPDemo** is provided on the CP3UB17/CP3CN17 Evaluation Kit CD-ROM. This program sets up a 1-second periodic timer interrupt which flashes two LEDs on the evaluation board. The program also provides a set of demonstrations for evaluation board peripherals such as the USB port and the audio interface. To compile the **CPDemo.c** example, you will need the files shown in Table 7-1. After installation of the CD-ROM software, these files are located in two sub-folders of the folder at the default location **C:\National_SEK_X_Y\software\cp3000\example\CPDemo**, in which X.Y is the version number of the software release. The source code and winIDEA project files for the example are in the sub-folder **fw**. Files that execute on the host PC are in the sub-folder **gui**.

Example Files

Table 7-1. Files Required for the CPDemo.c Example

Name	File Type	Description
fw/AudioDemo.c	C source	Program source file.
fw/AudioDemo.h	C source	Header file.
fw/build.bat	C source	Called by **winIDEA**. Calls **gmake.exe** to execute make files.
fw/CommIntp.c	C source	Program source file.
fw/commintp.h	C source	Header file.
fw/CPDemo.c	C source	Program source file.
fw/depend.mak	C source	Defines paths to header files used by makefile.
fw/descrpt.h	C source	Header file.
fw/devprop.h	C source	Header file.
fw/makefile	make file	Top-level make file.
fw/makefile.iar	make file	makefile containing IAR tool chain **make** rules. This tool chain is not used in this manual.
fw/makefile.ns	make file	makefile containing NSC tool chain **make** rules.
fw/protocol.h		Header file.
fw/USB_Demo.c		Program source file.
gui/CPDemo.exe		Host-side user interface application that communicates with the CPDemo program over the serial and USB connections.
gui/usbinstall		Folder that contains host-side USB software for communication with the evaluation board.

Getting Started with winIDEA

8

This chapter describes the winIDEA user interface and how to create a workspace. For detailed information on the winIDEA environment, see the *winIDEA User's Manual.*

8.1 winIDEA User Interface

A typical view of the winIDEA user interface is shown in Figure 8-1.

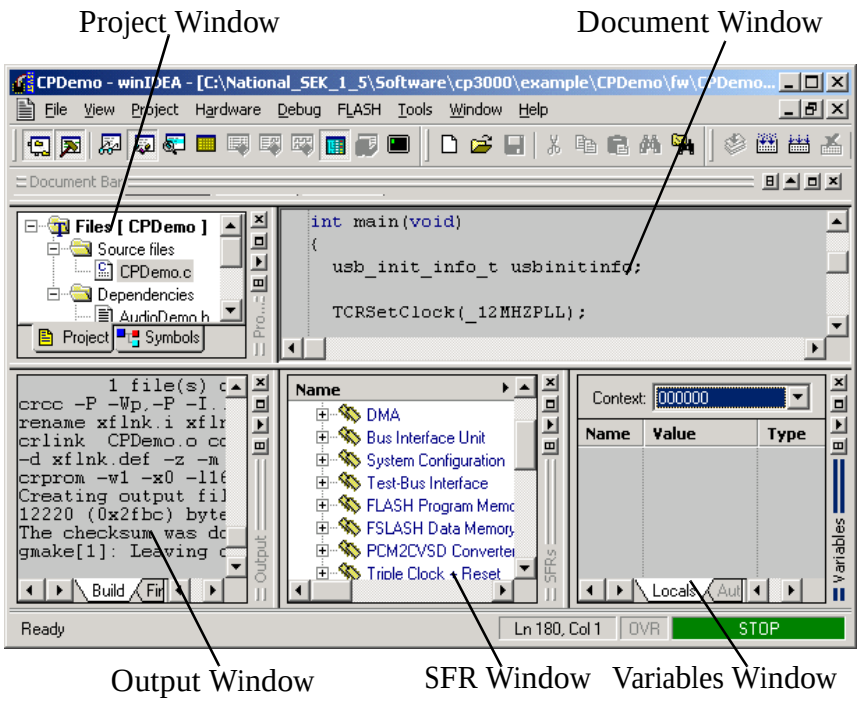


Figure 8-1. winIDEA User Interface

The windows shown in Figure 8-1 are:

- **Project Window**—lists the files which comprise the project.
- **Document Window**—used to view and edit source files, to indicate where program execution stopped, and to set breakpoints.
- **Output Window**—displays error and warning messages during compilation.
- **SFR Window**—shows the contents of Special Function Registers (SFR) for the CPU and peripherals.
- **Variables Window**—shows the values and types of variables in the current context.

8.2 Invoking C Compiler Tool Chain From winIDEA

An external make utility is used to invoke the C compiler tool chain from winIDEA. The external make utility is the **gmake** program from the GNU tool set. A batch file (**build.bat**) is invoked from winIDEA to call the **gmake** utility, as shown in Figure 8-2.

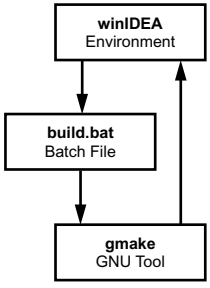


Figure 8-2. Calling gmake from winIDEA

Because winIDEA has only a limited understanding of the messages produced by **gmake**, the Tools tab in the Output window may be used to view more detailed information about the activity of the tools during a make or build process.

8.3 Creating a New Project

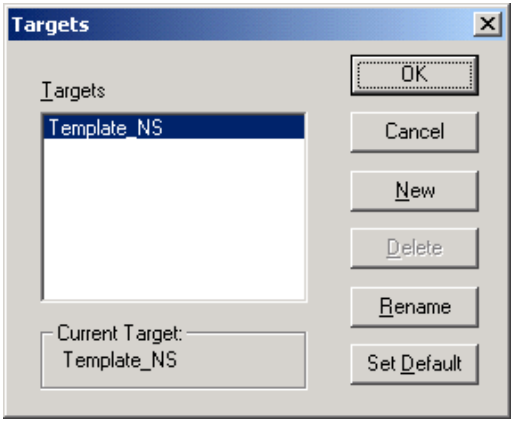
A set of template files is provided in the **C:\National_SEK_X_Y\software\cp3000\templates** folder (in which X.Y is the version number of the software release) for creating new projects. These two files are used to compile the **CPDemo** example:

- **template_NS.jrf**—this is a workspace file, which holds information about the project files, arrangement of windows in the winIDEA user interface, and other data defining a session in the winIDEA environment. To enter the winIDEA IDE, there must be an open workspace file.
- **template_NS.QRF**—this is a project configuration file, which holds the project settings accessed by selecting the Project -> Settings command in winIDEA.

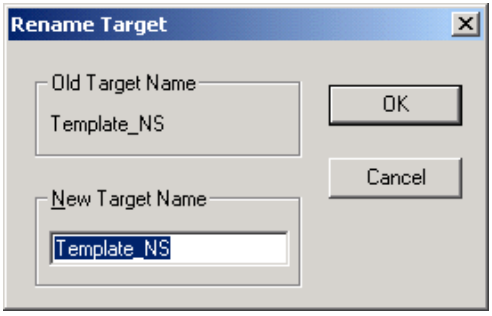
The following procedure creates a new project:

1. **Copy the Template Files.** Copy the **template_NS.jrf** and **template_NS.QRF** files, and paste them into the **C:\National_SEK_X_Y\software\cp3000\example\CPDemo\fw** folder (in which X.Y is the version number of the software release). Rename these files **CPDemo.jrf** and **CPDemo.QRF**.

2. **Enter winIDEA.** Double-click on the **CPDemo.jrf** file to enter winIDEA.
3. **Change the Project Target.** Select the Project -> Targets command. This brings up the Targets dialog box.

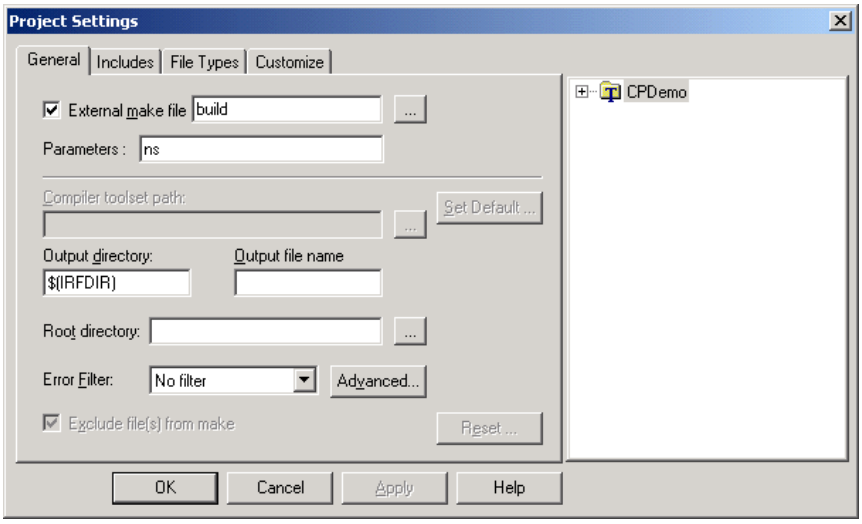


Click the Rename button. This opens the Rename Target dialog box.



Name the project **CPDemo,** and click the OK button to close the Rename Target dialog box, then click another OK button to close the Targets dialog box.

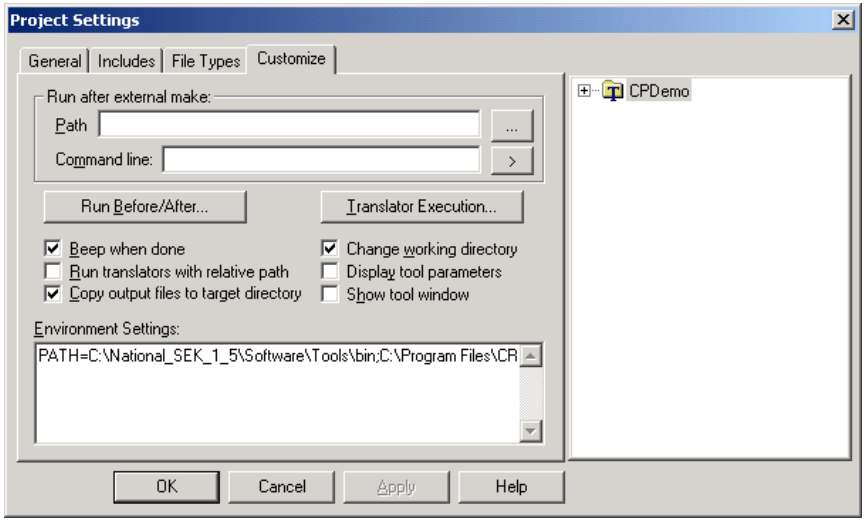
4. **Change the Project Settings.** Select the Project -> Settings command. This brings up a dialog box.



The Parameters box is used to pass an argument to the makefile. The **ns** argument generates output files in the National COFF format (**.x** extension. Other defined arguments are listed below:

- **nshex**—Generate object files in Intel Hex format (**.hex**). This format does not include debug information.
- **clean**—Removes object files from the current directory.
- **clobber**—Removes objects and executables from all project directories.

5. **Edit the Search Path.** Click the Customize tab. In the Environment Settings text box, change **National_SEK_X_Y** so that X.Y corresponds to the release of the software. For example, for release 1.5, this would be **National_SEK_1_5**.



Click the OK button when finished.

6. **Save Files.** Select the File -> Save All command to save the current workspace with its new settings.

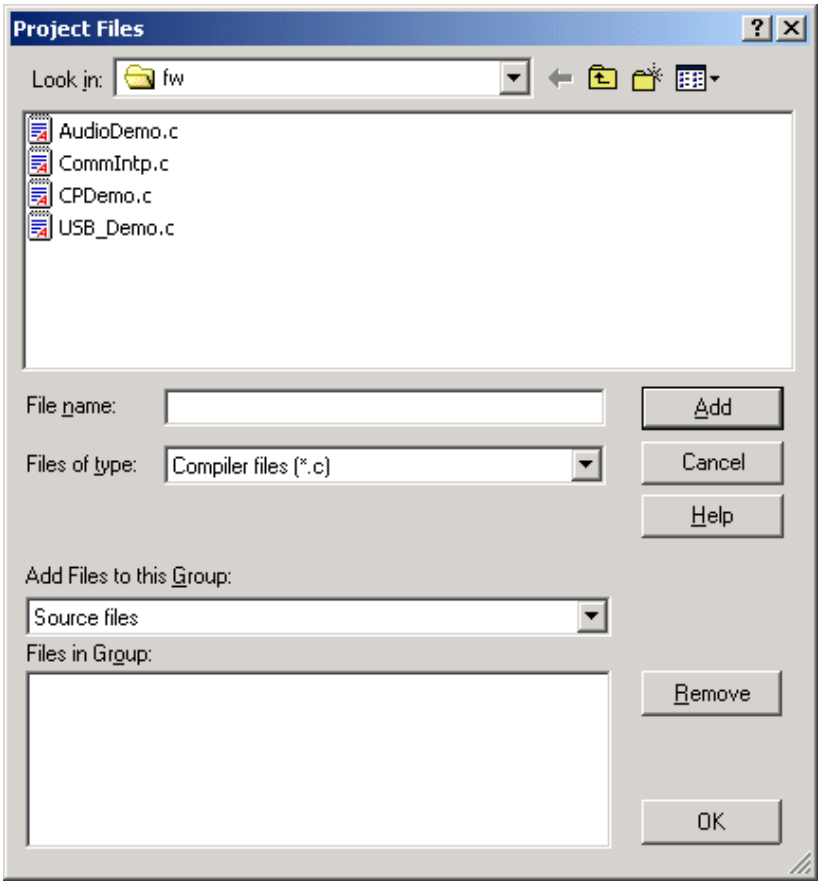
Building an Application

9

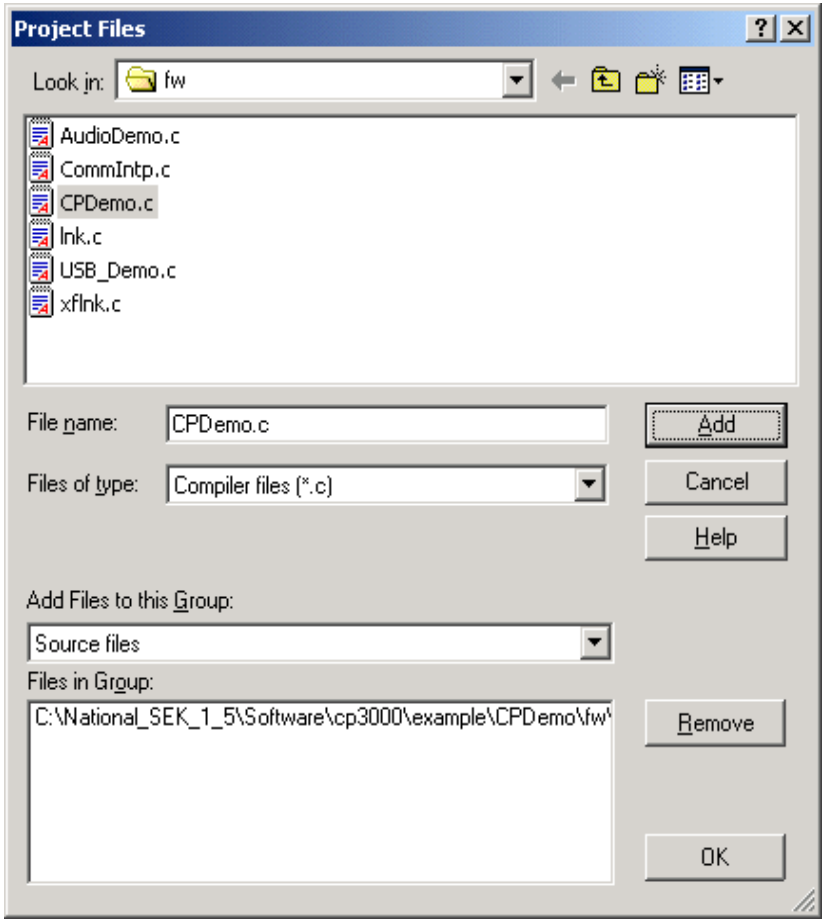
9.1 Add Project Files

The following procedure adds files to a project:

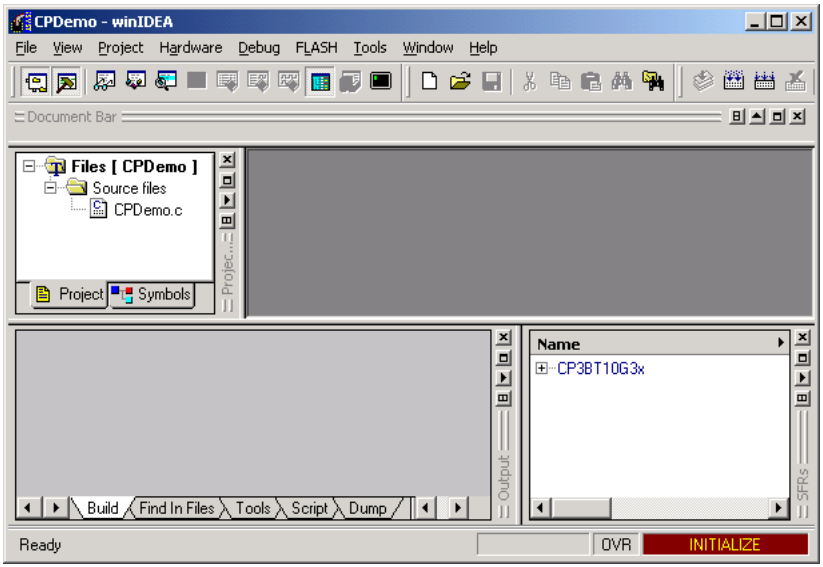
1. **Bring Up the Project Files Dialog Box.** Select the Project -> Project Files command to bring up the Project Files dialog box.



2. **Add the CPDemo.c Source File.** Click on **CPDemo.c**, click on the Add button, and then click on the OK button.



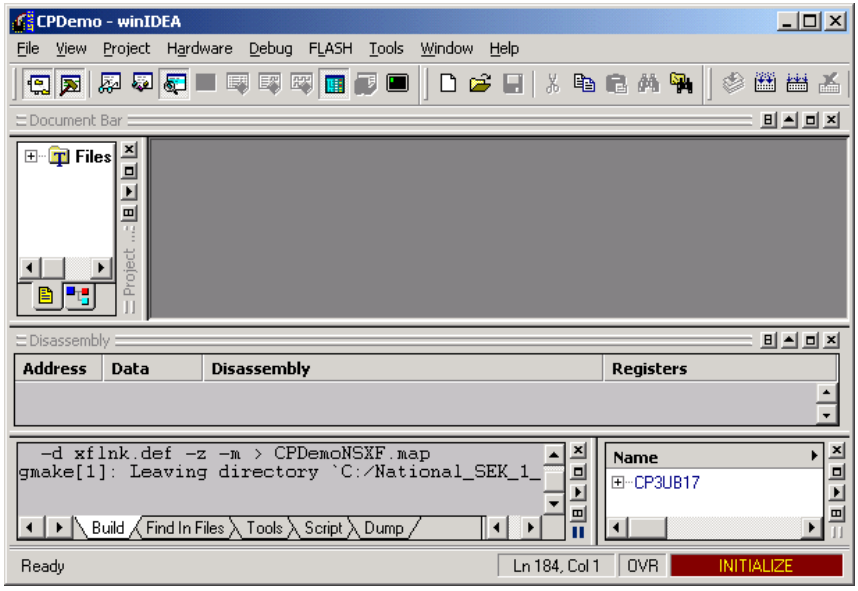
3. **Review the Project Source Files.** After returning to the Project window, double-click on Files, then double-click on Source Files to view the source files in the project. The **CPDemo.c** file will appear as a source file.



4. **Save Files.** Select the File -> Save All command to save the current workspace with the new project file information.

9.2 Build the Application

Select the Project -> Make command to compile and link the project.



Downloading Applications 10

This chapter describes how to download the **CPDemoXF.x** file to the evaluation board.

10.1 Configuration for Downloading to Flash Memory

To download a program to flash memory, jumpers J15 and J16 must be in the position marked as 1 on the board. Jumper J17 is in the 1 position for IRE mode or the 0 position for ERE mode. The on-chip flash memory can be programmed and executed in either IRE or ERE mode. The external (off-chip) flash memory can only be used in ERE mode. Because the **CPDemo** example uses the external flash memory, the evaluation board must be in ERE mode for this example.

Acceptable file formats for download in IRE or ERE mode include National COFF format (**.x** extension), Intel Extended Hex (**.a45** or **.hex** extension), and UBROF8 (**.d45** or **.dbg** extension). Loading an NSC COFF or UBROF8 file will take up more space in the internal flash memory due to some debug overhead, while loading an Intel Extended Hex file into the internal flash memory will occupy a smaller amount of memory because no debug information is included. The winIDEA debugger checks the CP3UB17/CP3CN17 operating environment (IRE/ERE or DEV mode) and downloads to either the internal flash or the external SRAM, respectively.

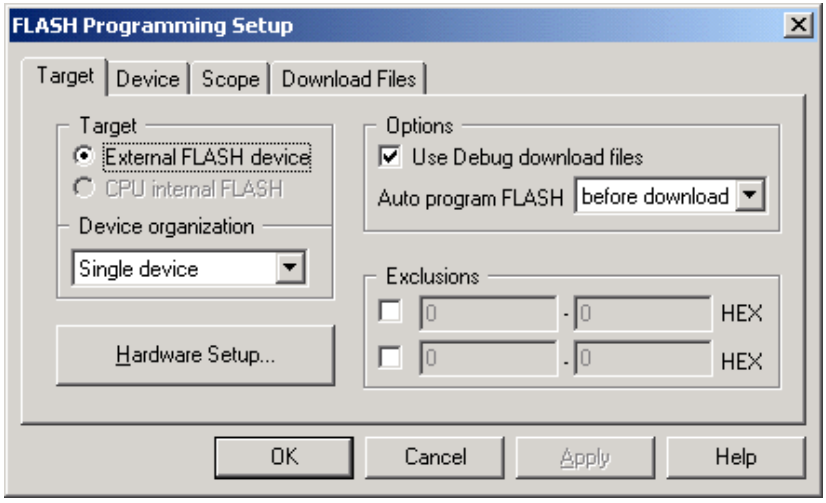
10.2 Configuration for Downloading to External SRAM

The **CPDemo** program does not support operation from SRAM. This section is provided only to inform you about other methods for downloading application code.

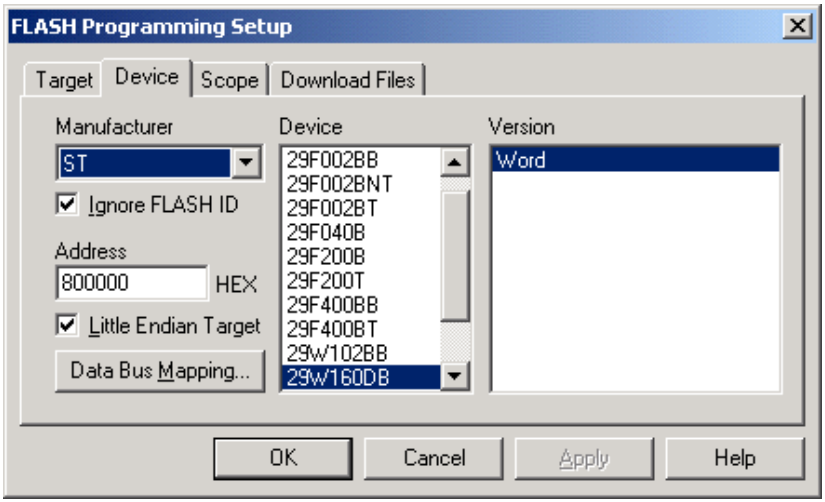
To download a program to the external SRAM on the evaluation board, jumpers J15, J16, and J17 must be in the position marked as 0 on the board. This configures the CP3UB17/CP3CN17 to operate in DEV (Development) mode. Acceptable file formats for download in DEV mode include National COFF format (**.x** extension), Intel Extended Hex (**.a45** or **.hex** extension), and UBROF8 (**.d45** or **.dbg** extension). Downloading an NSC COFF or UBROF8 file allows debugging features (such as breakpoints) to be used from winIDEA, while loading an Intel Extended Hex file only allows running the program without debugging features. The winIDEA debugger checks the CP3UB17/CP3CN17 operating environment (IRE/ERE or DEV mode) and downloads to either the internal flash or the external SRAM, respectively.

10.3 Downloading CPDemoXF.x

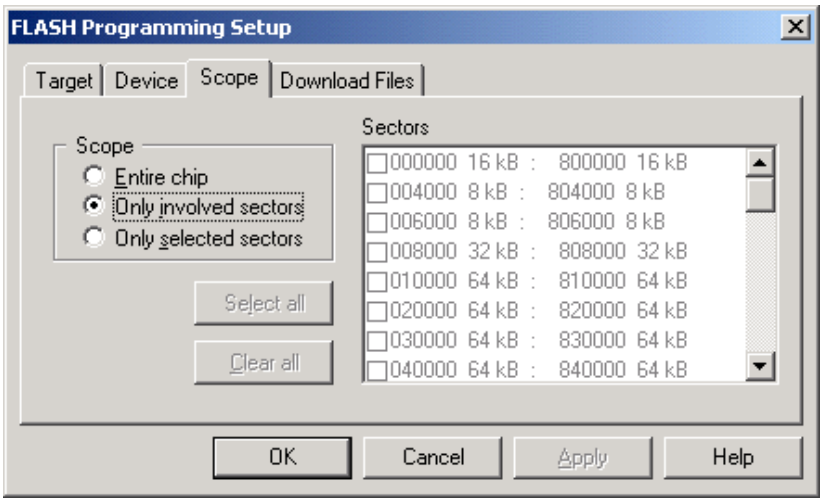
1. **Set Up the External Flash Memory Download.** The **CPDemoXF.x** file contains code for programming to both the on-chip flash memory and the external flash memory. winIDEA allows separate files to be specified for both the on-chip flash memory and the external flash memory, but in this example **CPDemoXF.x** will be specified for both download operations. First, the external flash memory download file is specified. Select the FLASH -> Setup command. This brings up a dialog box for specifying the target. The template file should already have the default settings shown below.



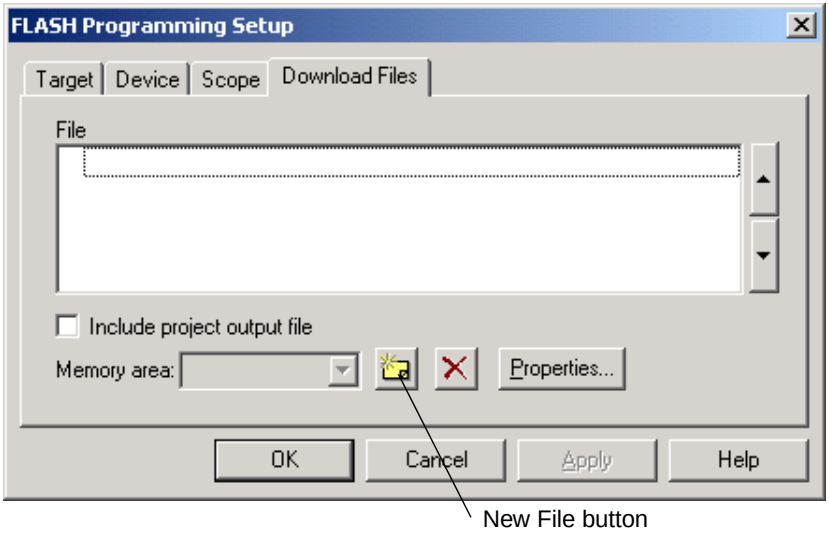
2. **Review the Device Settings.** Click on the Device tab. The template file should already have the default settings shown below.



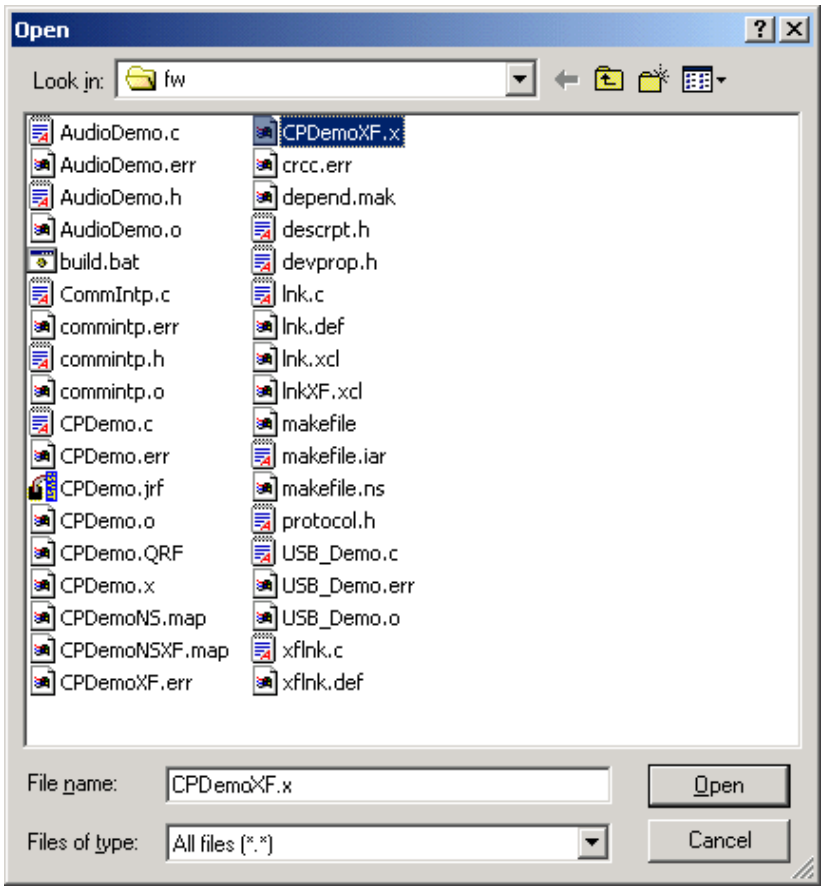
3. **Review the Scope Settings.** Click on the Scope tab. The template file should already have the default settings shown below.



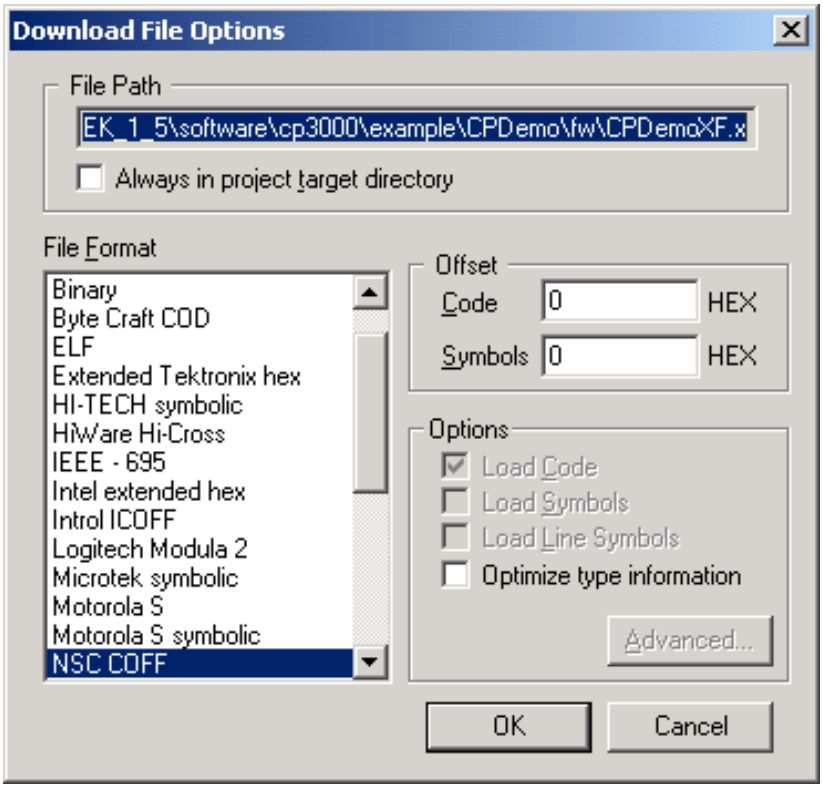
4. **Specify a New Download File.** Click on the Download Files tab. The New File button is used to specify a new download file. Click on the New File button.



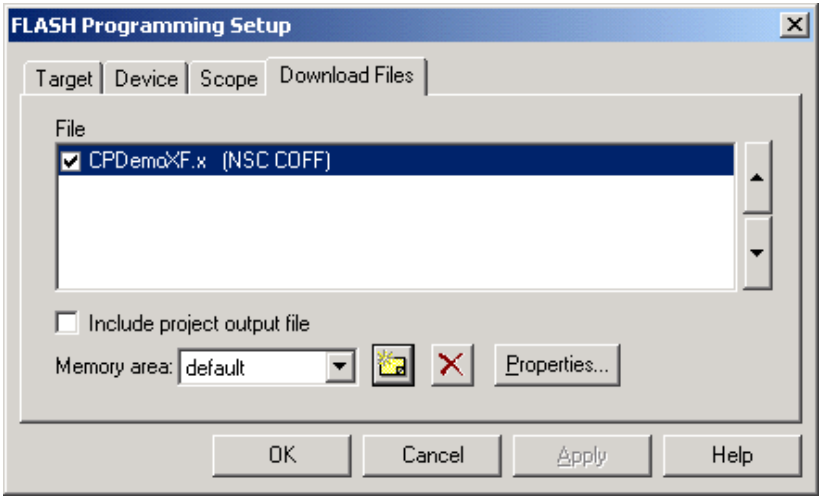
5. **Browse for the Download File.** Select All files from the Files of type drop-down menu. Then, select the **CPDemoXF.x** file, and click the Open button to proceed.



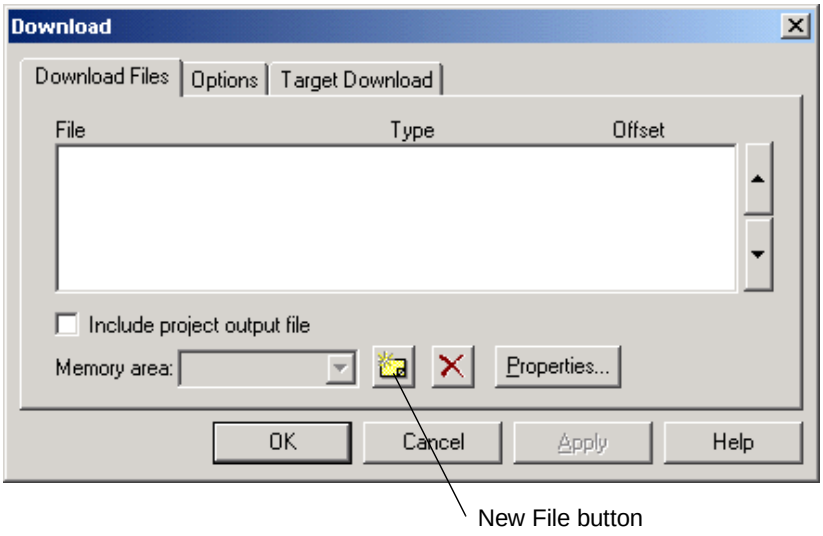
6. **Confirm the Download File Options.** Click the OK button to proceed.



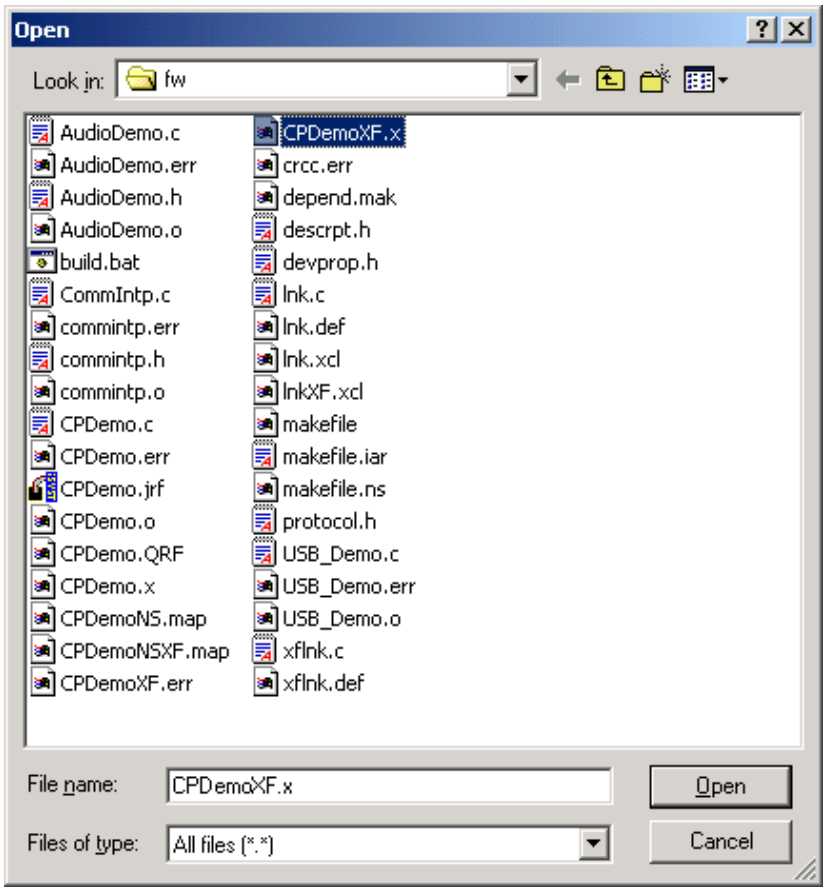
7. **Confirm the Download File.** At this point, the Download dialog box should display the file **CPDemoXF.x**. Click the OK button to proceed.



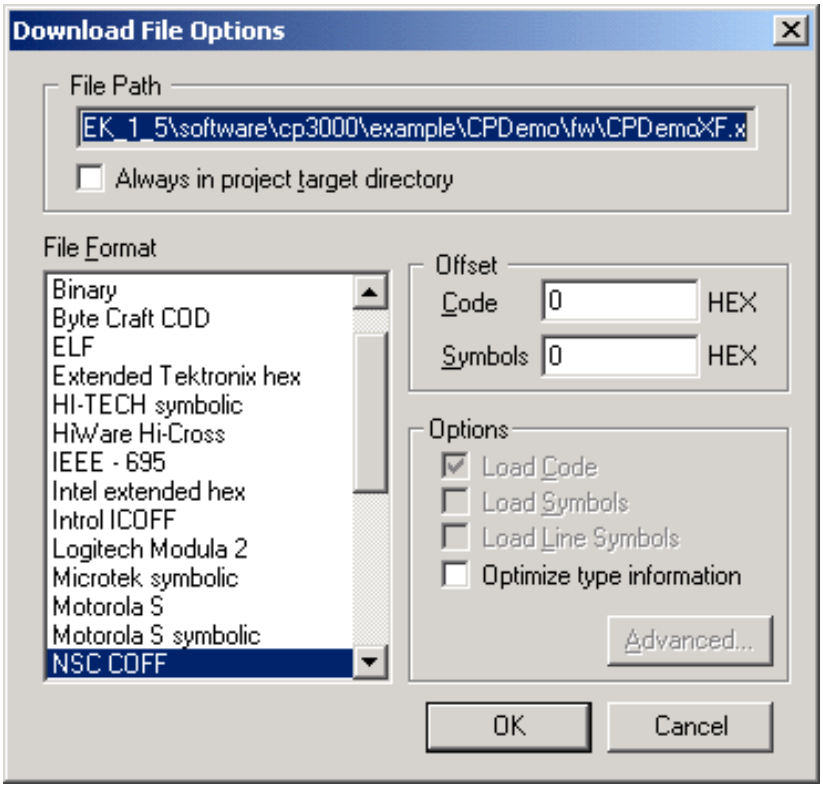
8. **Set Up the External Flash Memory Download.** Now, the file for downloading to the on-chip flash memory will be specified. Select the Debug -> Files For Download command. This brings up a dialog box for specifying the files. Click on the New File button.



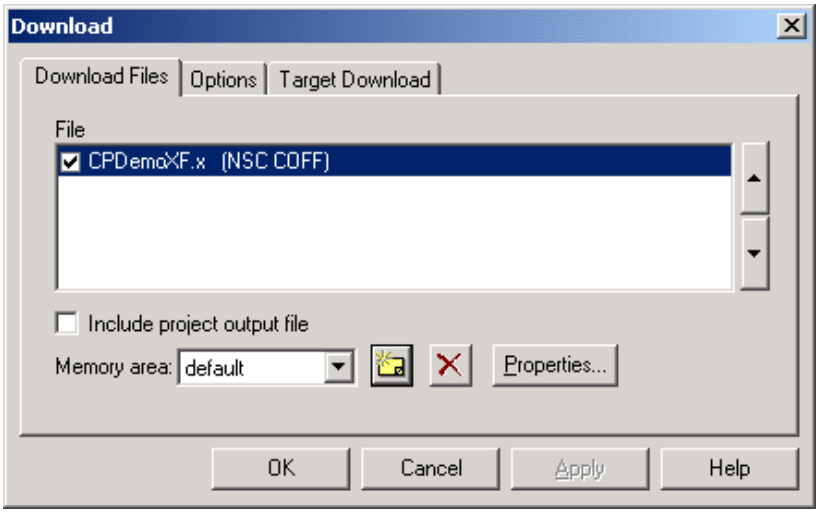
9. **Browse for the Executable File.** Select All files from the Files of type drop-down menu. Then, select the **CPDemoXF.x** file, and click the Open button to proceed.



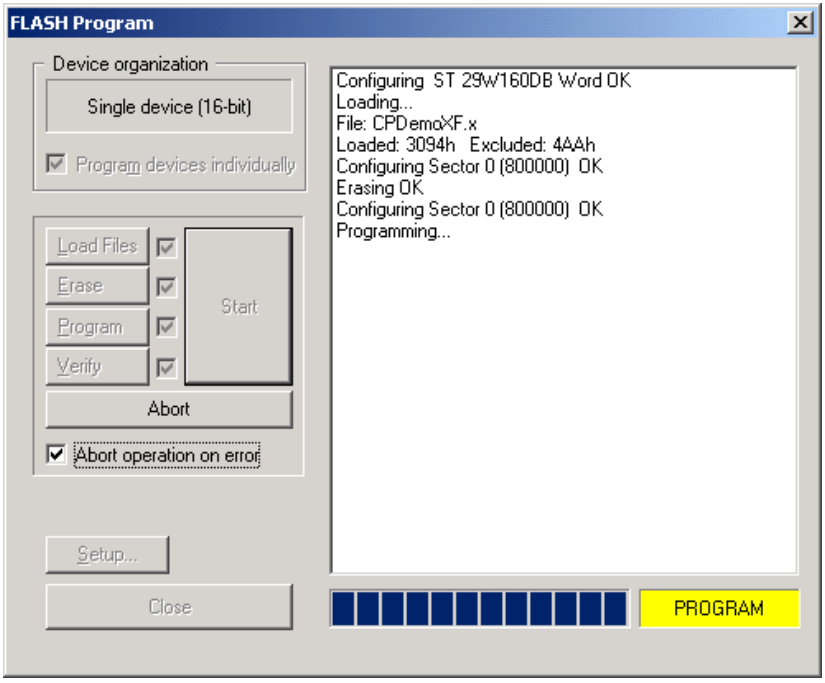
10. **Confirm the Download File Options.** Click the OK button to proceed.



11. **Confirm the Download File.** At this point, the Download dialog box should display the file **CPDemoXF.x**. Click the OK button to proceed.



12. **Download to the On-Chip and External Flash Memories.** Select the Debug -> Download command. During the programming operation, a window will appear to report the progress as shown below.



13. **Run the Program on the Target.** Select the Debug -> Reset And Run command. At this point, two LEDs (LD1 and LD2) on the evaluation board should start flashing.

10.4 Changing the Protection Word

The Protection Word holds critical device protection and configuration bits. The Protection Word usually does not need to be changed during program development.

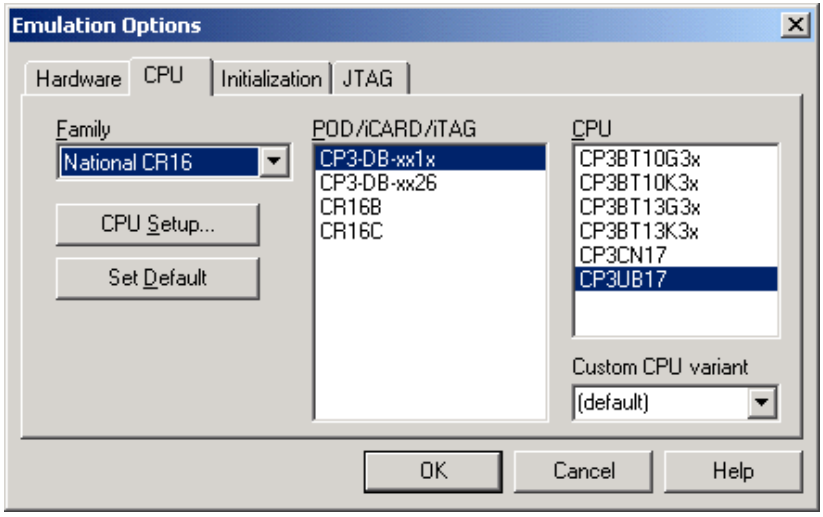
Before programming the Protection Word, read the device data sheet about the features controlled by the Protection Word. You MUST understand these features before programming the Protection Word, because some settings are irreversible and may make the device unusable. These features are fully discussed in the device data sheet, but not in this document.

The Protection Word is located in Information Block 1, which is a section of flash memory outside of the CPU address space. At reset, the contents of the Protection Word are read into a register, which then controls the functions assigned to the Protection Word. Therefore, changes made to the Protection Word do not take effect until the next device reset.

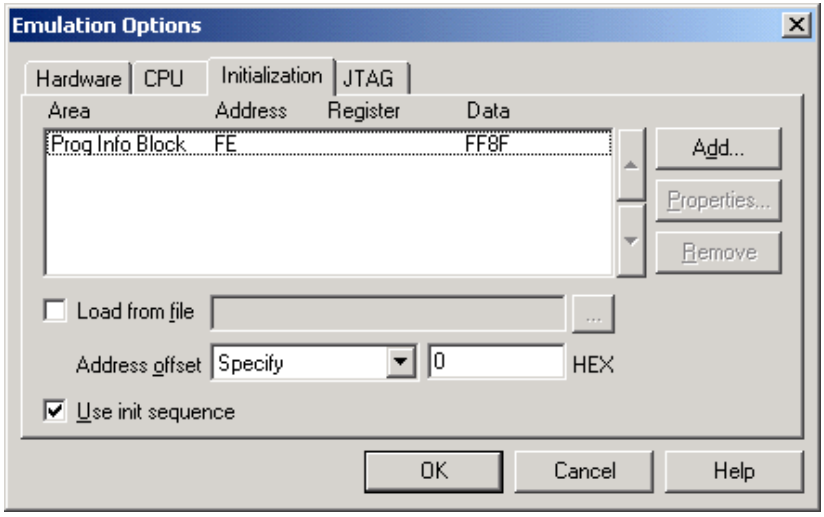
10.4.1 Setting Up Protection Word Programming

The following procedure sets up winIDEA to program the Protection Word during download:

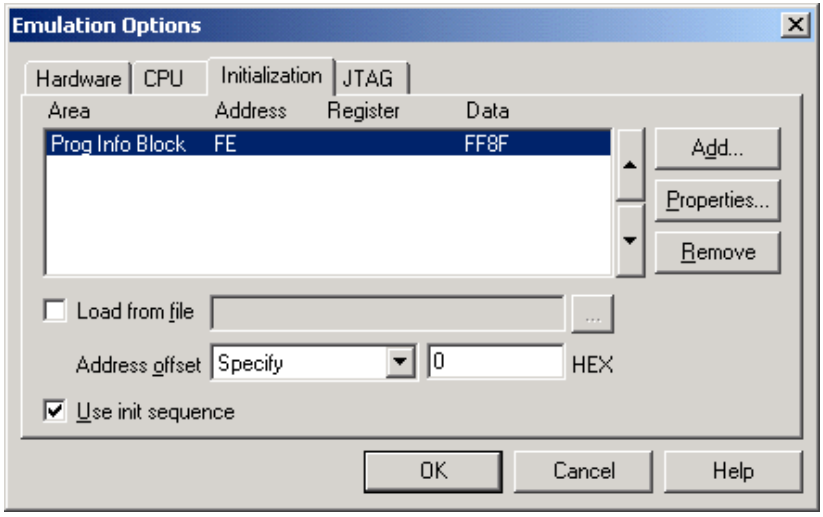
1. **Enter the Emulation Options Dialog Box.** Select the Hardware -> Emulation Options command.



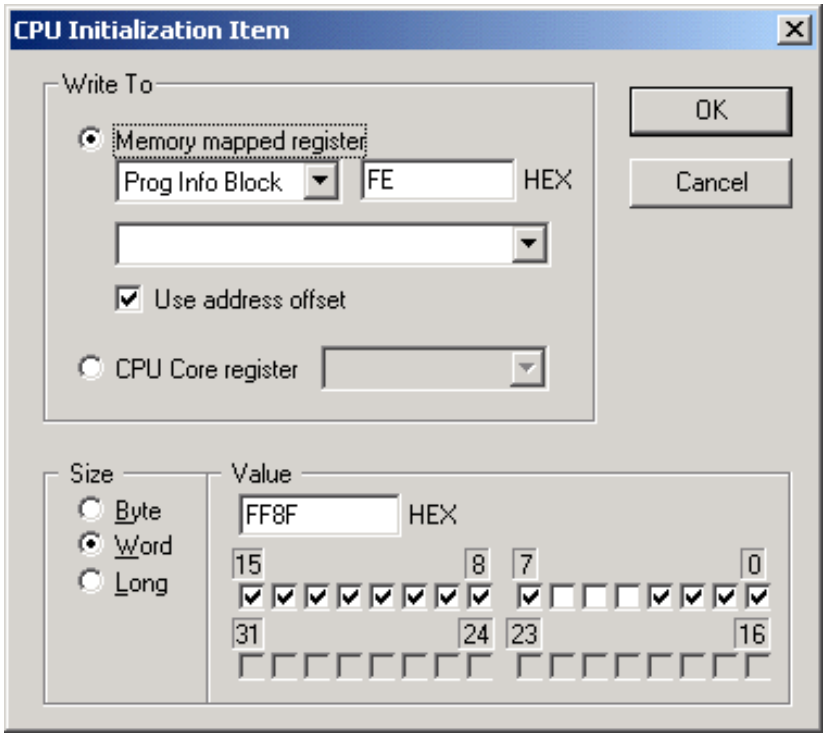
2. **Click the Initialization Tab.** This changes the display to show memory locations that are initialized on download. The project template files are set up to initialize the Protection Word (FEh) to a safe value (FF8Fh).



3. **Select the Protection Word.** This enables the Properties button, which is used to change the definition of a memory location to be initialized. For specifying a new definition, the Add button is used.



4. **Click the Properties Button.** This brings up the CPU Initialization Item dialog box for the Protection Word.

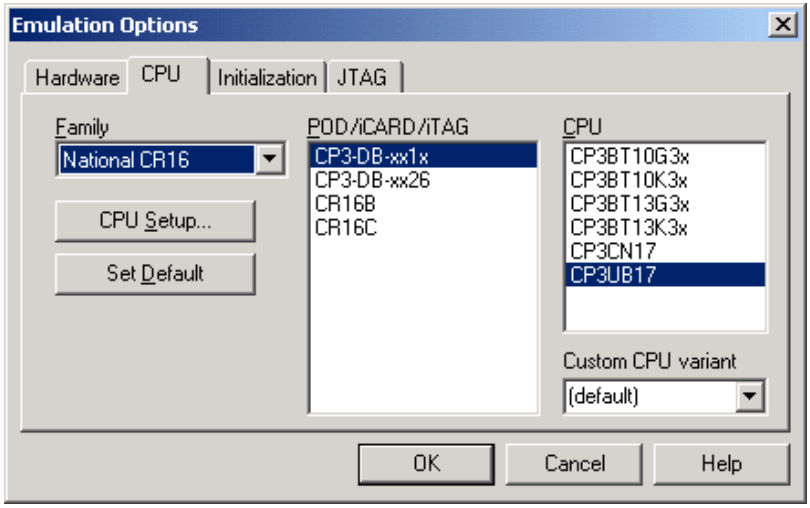


A safe value to program into the Protection Word is FF8F (global read protection disabled, global write protection disabled, no ISP routines, code starts at address 0). Click the OK button to close the CPU Initialization Item dialog box. Then, click the OK button to return to the main winIDEA display.

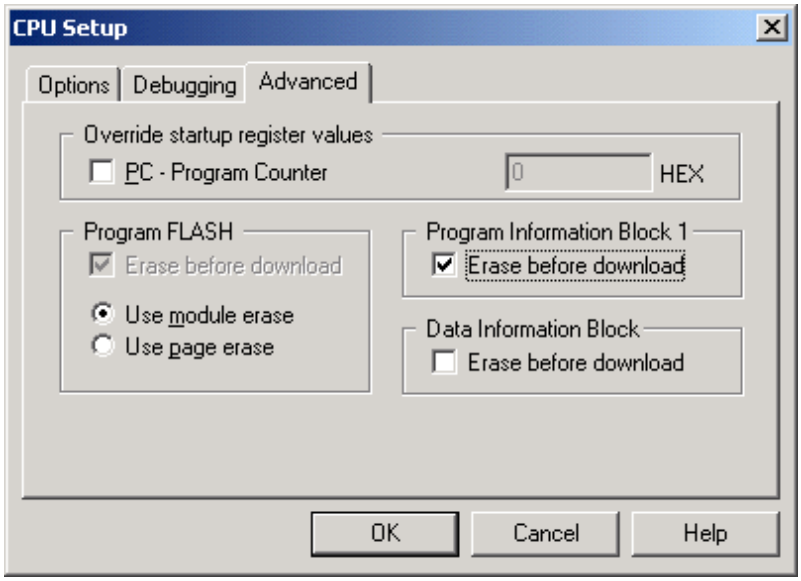
10.4.2 Setting Up Information Block Erase

The procedure described in Figure 10.4.1 for programming the Protection Word can only change bits from 1 to 0. If any of the changes to the Protection Word require changing a bit from 0 to 1, Information Block 1 (which contains the Protection Word) must be erased before programming. Erasing changes all of the bits to 1. The following procedure sets up winIDEA to erase Information Block 1 before programming.

1. **Enter the CPU Setup Dialog Box.** Select the Hardware -> Emulation Options command. Then click the CPU Setup button.



2. **Enable Information Block Erase.** Click on the Advanced tab. Click on the Erase before download check box. Click the OK button to return to the Emulation Options dialog box, then click the OK button to return to the main winIDEA display.



Downloading Applications

Changing the Protection Word

11 Running and Debugging

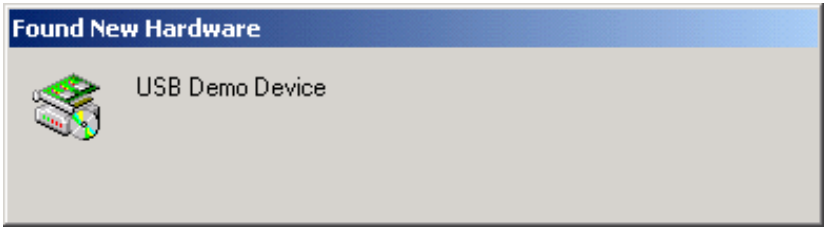
This chapter describes how to run the **CPDemo** program, and it provides a brief introduction to the debugging features of winIDEA.

When the program is run for the first time, Windows will need to be guided to the USB driver used to support the **CPDemo** program. Then, the host-side application program **CPDemo.exe** is executed on the host PC. This program provides a user-friendly interface for running the **CPDemo** example.

11.1 Installing the USB Driver

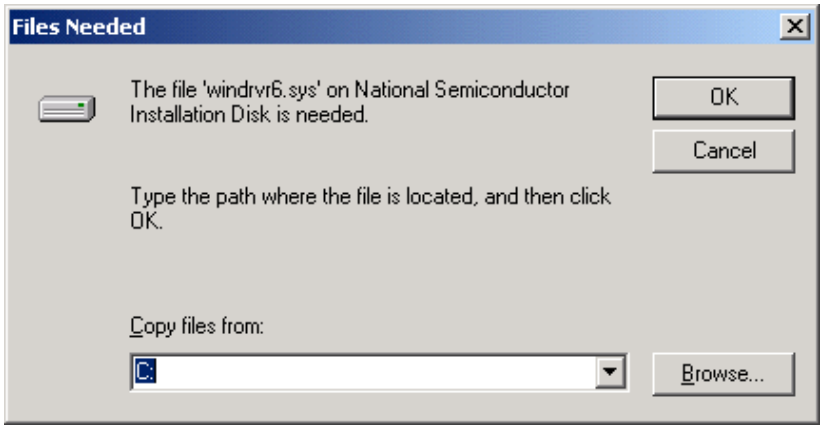
The first time the **CPDemo** program is executed with the evaluation board connected to the host PC, the USB driver will need to be installed.

1. **Run the CPDemo Program.** When the Debug -> Reset and Run command is selected (after successfully downloading the **CPDemo** application code), the USB connection between the evaluation board and the host PC becomes active. The first time this happens, Windows will hunt for the driver.

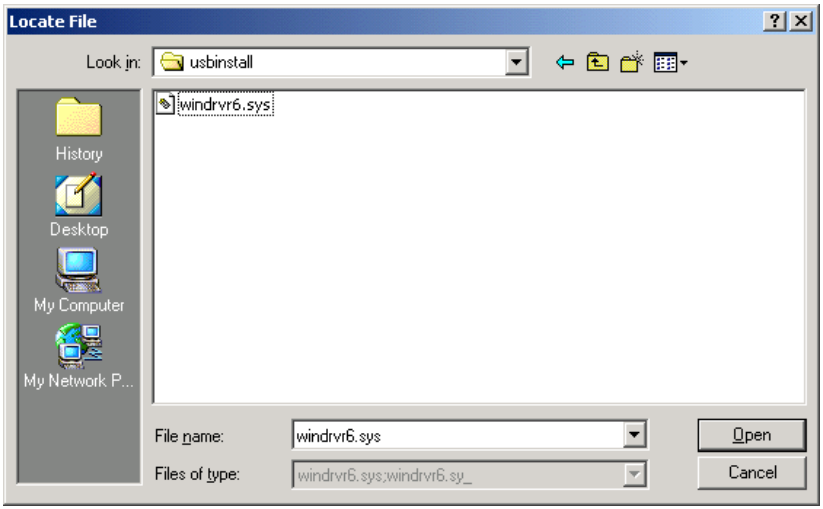


Wait a few seconds for the next message to appear.

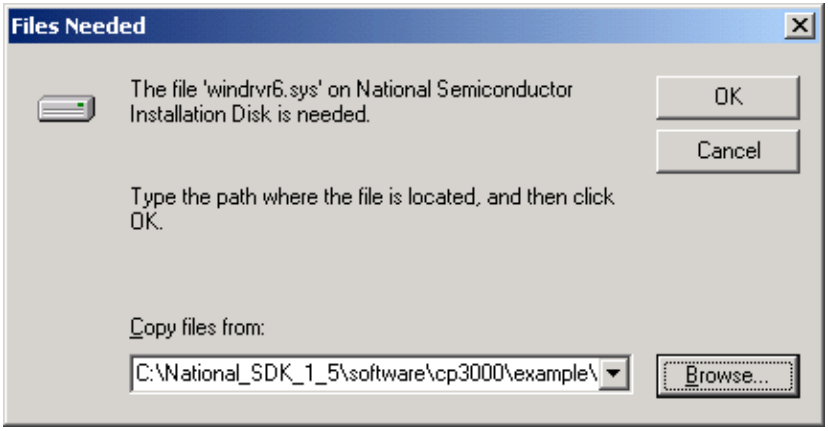
2. **Browse for the USB Driver.** Windows will hunt for a driver called **windrvr6.sys**. Click on the Browse button.



3. **Go to the Folder Holding the Driver.** Go to the **C:\National_SEK_X_Y\software\cp3000\example\CPDemo\gui\usbinstall** folder (in which X.Y is the version number of the software release). Click the Open button to proceed.

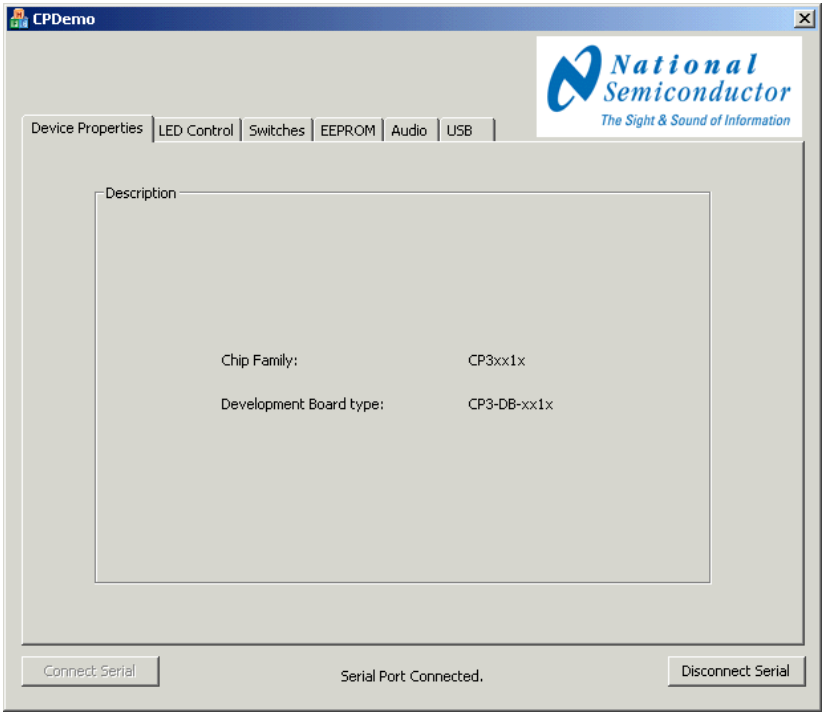


4. **Confirm the USB Driver.** After selecting the **usbinstall** folder, click on the OK button. This will install the USB driver needed by the **CPDemo** program.

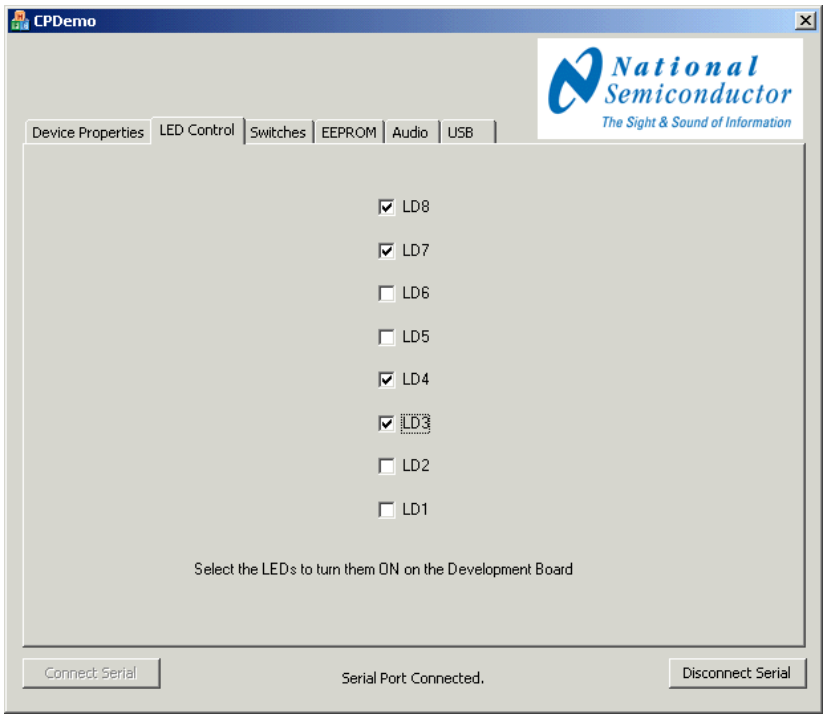


11.2 Running the Host-Side Application

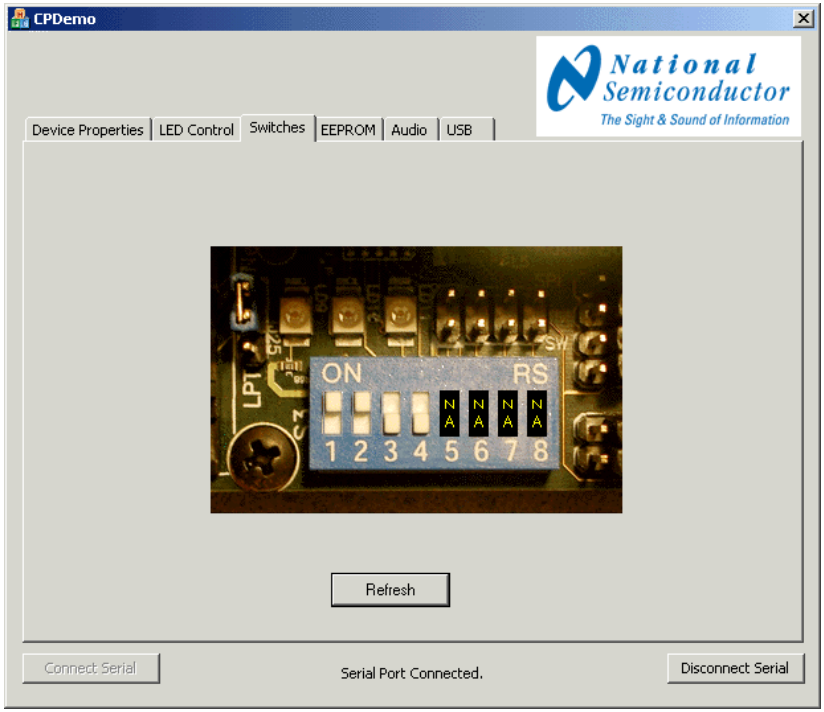
1. **Execute CPDemo.exe.** On the host PC, from the Start menu, select Programs -> National SEK 1.5 -> CPDemo. When the program starts up, it automatically attempts to connect with the evaluation board through the lower COM ports, such as COM1 or COM2. The first view should look like the one shown below, with the Device Properties tab in front.



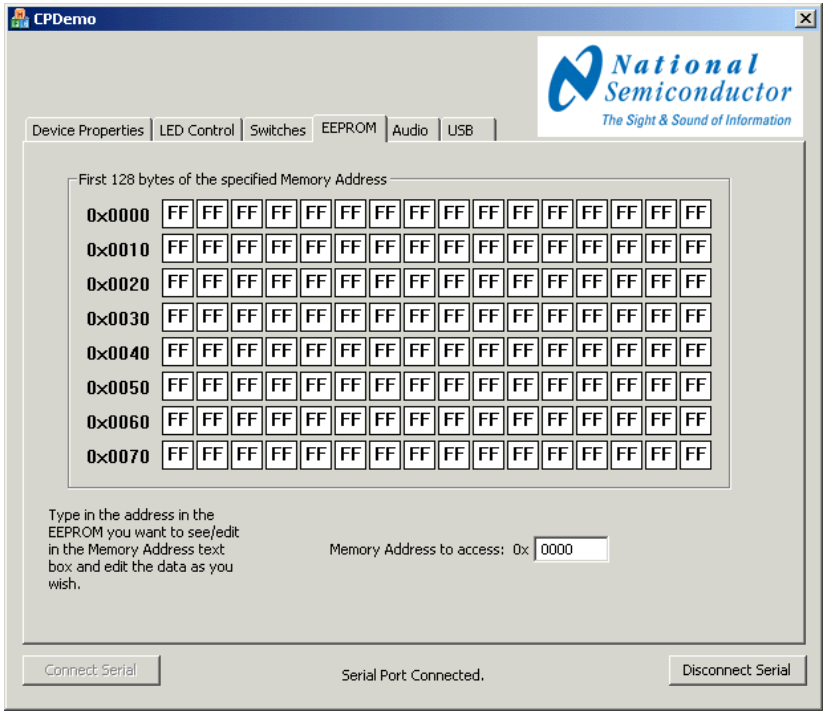
2. **Select the LED Control Tab.** Click on the LED Control tab to bring it to the front. Checking any box in this view will light up the corresponding LED on the evaluation board.



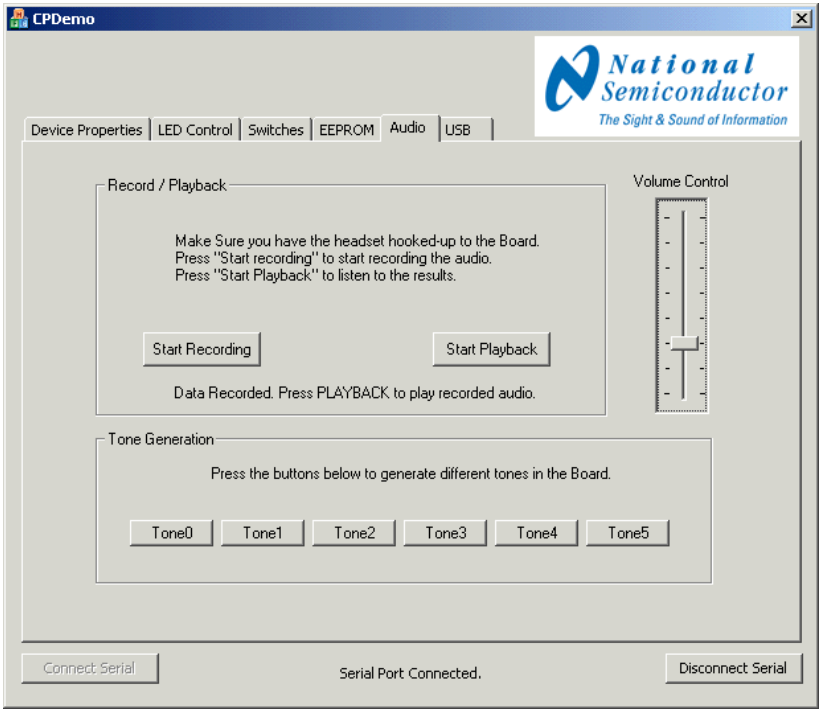
3. **Select the Switches Tab.** Click on the Switches tab to bring it to the front. Change DIP switches 1 through4, then click the Refresh button to observe the change. If the switches are stuck in the ON position, check that jumpers J42 through J45 are installed.



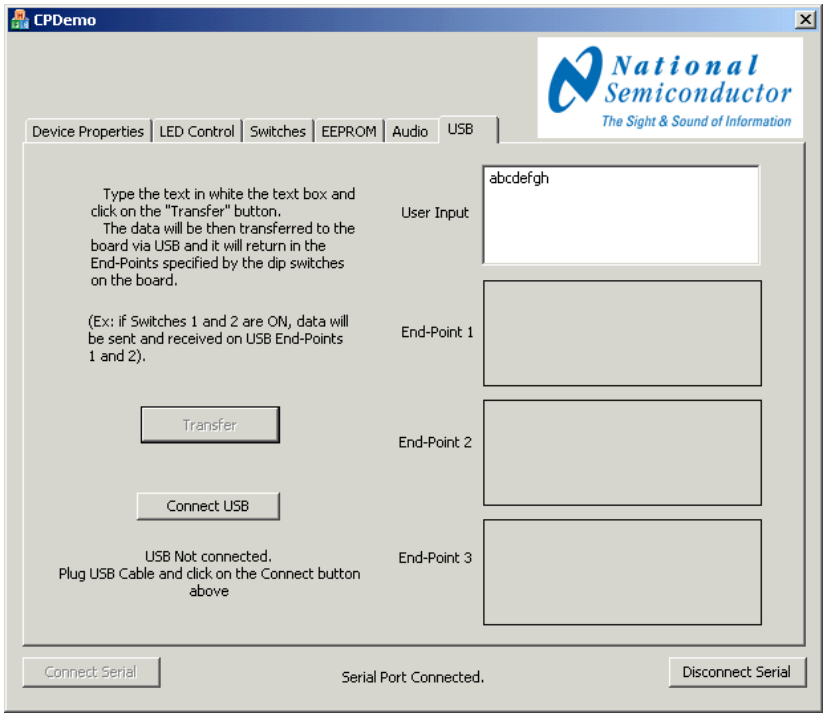
4. **Select the EEPROM Tab.** Click on the EEPROM tab to bring it to the front. Click on any byte in this view to edit its value. An address can be entered into the Memory Address box, however the memory space wraps around starting at address 800h.



5. **Select the Audio Tab.** Click on the Audio tab to bring it to the front. A microphone/speaker headset with a 2.5 mm plug must be connected to J34 for the demonstrations accessed through this view. Click on a Tone button to hear a short tone. Click on the Start Recording button to record about 6 seconds of audio, and click on the Start Playback button to play it back.

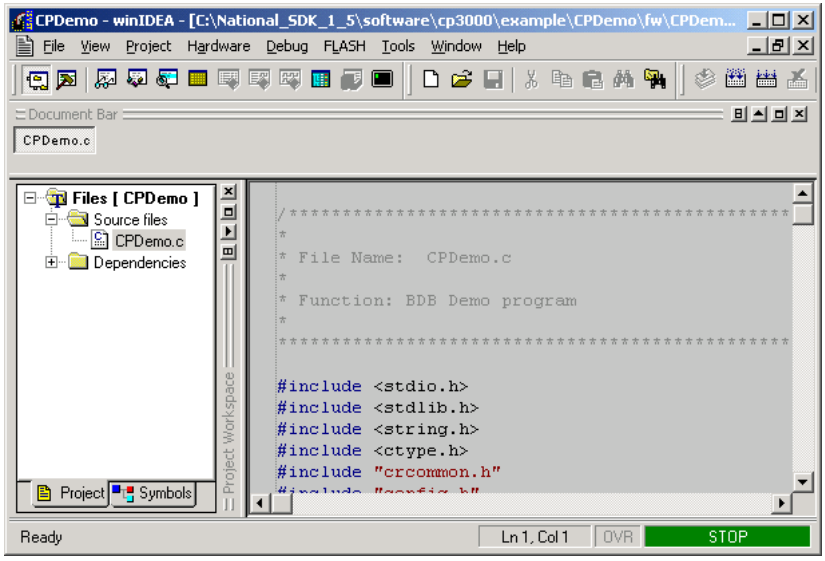


6. **Select the USB Tab.** Click on the USB tab to bring it to the front. Click the Connect USB button to connect to the evaluation board. Enter text into the User Input box, then click on the Transfer button to send the text to the evaluation board over the USB connection. It will be echoed back with uppercase/lowercase inverted to the endpoints selected by the DIP switches.

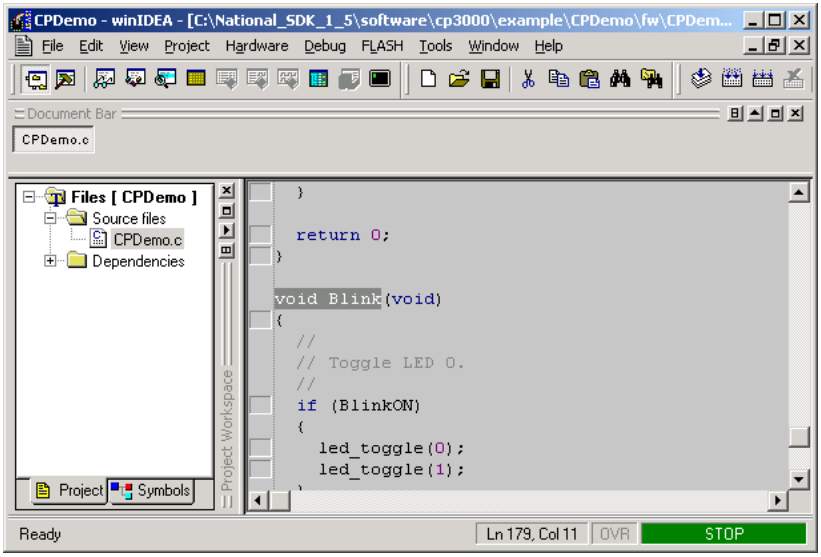


11.3 Viewing Program Execution

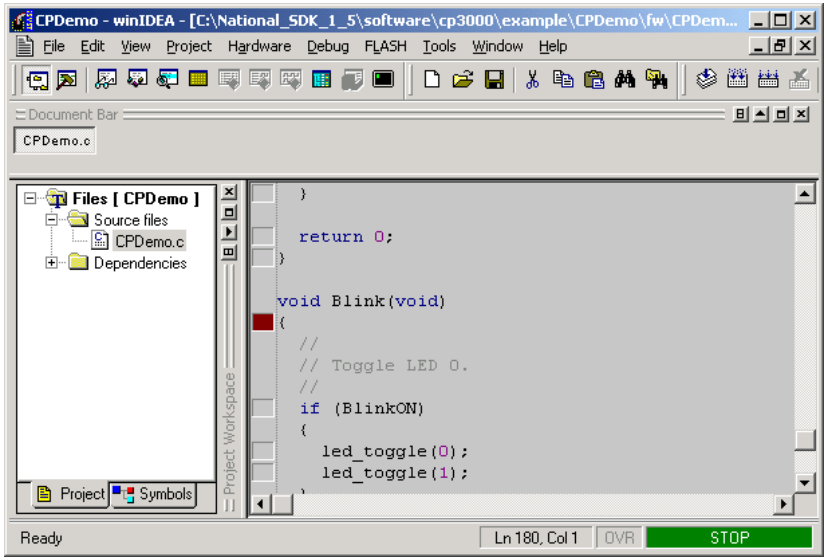
1. **Stop Program Execution.** Select the Debug -> Stop command.
2. **Open CPDemo.c.** Double-click on **CPDemo.c** in the Project window to open a window for the source file. In the view shown below, other winIDEA windows have been closed.



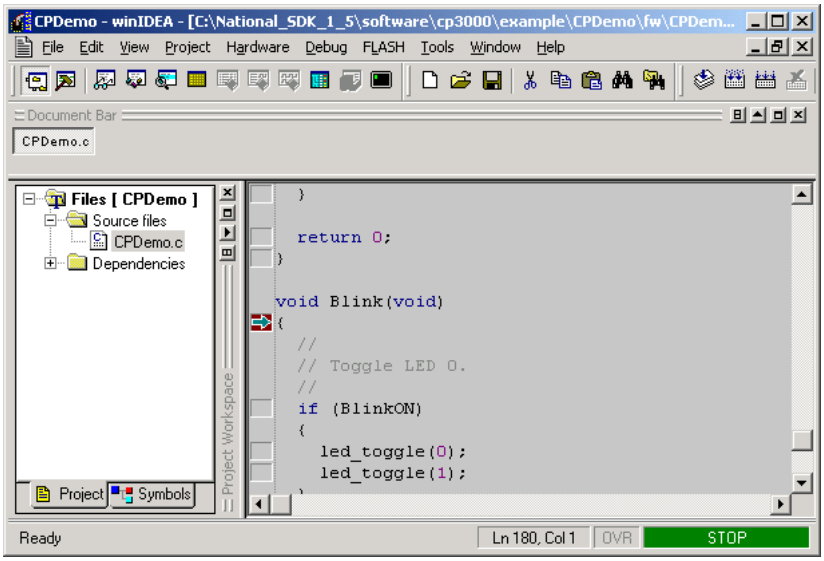
3. **Find the Blink Function.** Click in the **CPDemo.c** window. Then, enter Control-F or select the Edit -> Find command. Enter "void Blink" into the search field, and click the Find Next button. Repeat to search for the second occurrence.



4. **Set a Breakpoint.** Hit the F9 function key (or select the Debug -> Toggle Breakpoint command) to toggle the breakpoint status of the selected line. A box to the left of the line is highlighted in color, to indicate that a breakpoint has been set.



5. **Run Until the Breakpoint is Reached.** Hit the F5 function key or select the Debug -> Run command, so the program runs until reaching the breakpoint. An arrow appears to the left of the program line to indicate where execution stopped. Repeat this keystroke or command, so that the program continues running until it hits the breakpoint again. Each cycle through the program toggles the two LEDs that flash when the program is free-running.



11.3.1 Function Keys

Some of the function keys used to control program execution are listed in Table 11-1. These keys are equivalent to commands on the Debug menu.

Table 11-1. Program Execution Control Function Keys

Function Key	Debug Menu Command
F5	Run
Shift + F5	Reset
F9	Toggle Breakpoint
F11	Step

12 Evaluation Board

The CP3UB17/CP3CN17 Evaluation Board (CP3-DB-xx1x) is a platform for creating applications based on the CP3UB17/CP3CN17 connectivity processors. These applications typically use the USB or CAN interface.

The evaluation board provides a socket for the CP3CN17 or CP3UB17. Jumpers configure the evaluation board for the specific processor and operating environment.

The evaluation board supports four operating environments:

- **DEV (Development)**—execution from external SRAM.
- **IRE (Internal ROM Enabled)**—execution from on-chip flash memory.
- **ERE (External ROM Enabled)**—execution from on-chip and external flash memory.
- **ISP (In-System Programming)**—execution from boot code area of on-chip flash memory.

Evaluation Board

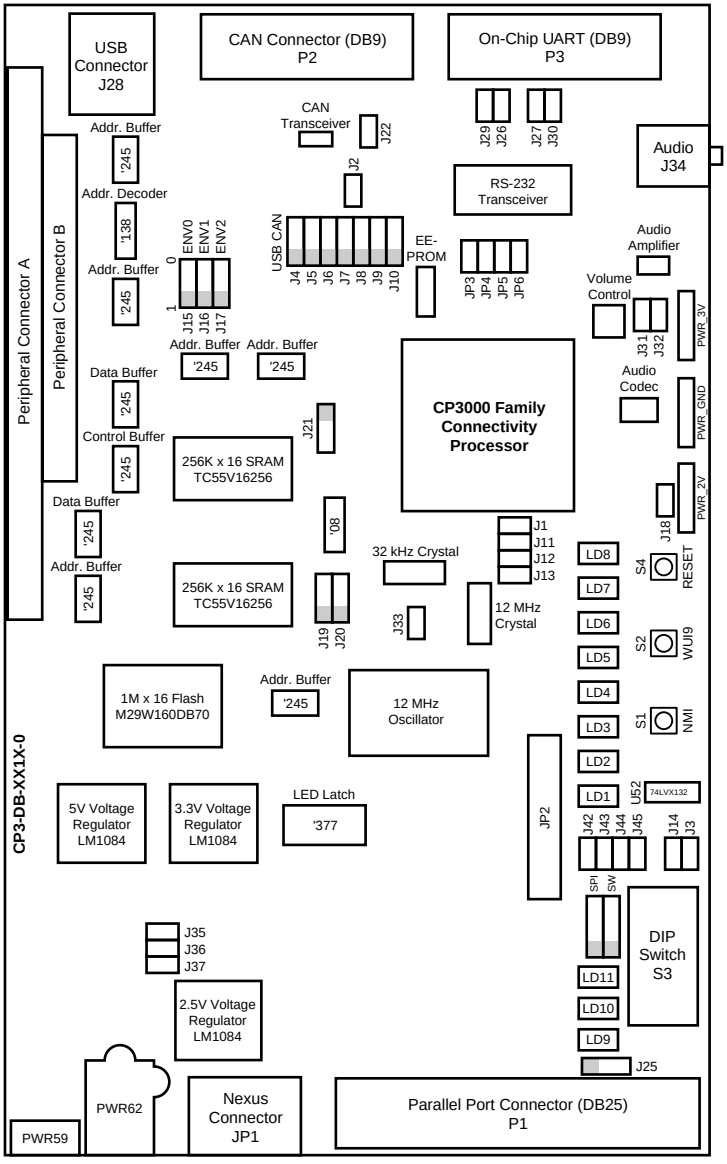
The evaluation board provides several types of memory, to support a range of applications:

- **SRAM**—256K × 16 for DEV mode
- **SRAM**—256K × 16 for ERE mode
- **Flash**—1M × 16 for ERE mode
- **ACCESS.bus Serial EEPROM**—2K bytes for all modes

A variety of peripherals for debugging and I/O are on the evaluation board:

- USB interface
- CAN interface
- UART connector for on-chip UART
- On-board codec for voice/audio applications
- DIP switches
- Pushbutton switches
- LEDs
- Peripheral board connector for addition of application specific interfaces
- JTAG-based Nexus interface for development tools
- Parallel port connector for low-cost development tools

12.1 Board Layout



12.2 Memory Map

The following tables show the memory map in the DEV, IRE, and ERE operating environments. The memory map in the ISP environment is a subset of the IRE and ERE environments, with the start address moved to the start of the boot code area.

Table 12-1. Memory Map in DEV Environment

Address Range	Size (bytes)	Contents	Description
0–08 0000h	512K	External SRAM	
0E C000h–0E C7FFh	10K	Internal SRAM	
FF FB00h–FF FBFFh	256	I/O Expansion	Off-chip peripherals. See I/O expansion table.
FF FC00h–FF FFFFh	1K	Internal I/O	On-chip peripherals

Table 12-2. Memory Map in IRE Environment

Address Range	Size (bytes)	Contents	Description
0h–03 FFFFh	256K	Internal Program Flash	
0E 0000h–0E 1FFFh	8K	Internal Data Flash	
0E C000h–0E C7FFh	10K	Internal SRAM	
FF FB00h–FF FBFFh	256	I/O Expansion	Off-chip peripherals. Must be activated by software. See I/O expansion table.
FF FC00h–FF FFFFh	1K	Internal I/O	On-chip peripherals

Table 12-3. Memory Map in ERE Environment

Address Range	Size (bytes)	Contents	Description
0h–03 FFFFh	256K	Internal Program Flash	
0E 0000h–0E 1FFFh	8K	Internal Data Flash	
0E C000h–0E C7FFh	10K	Internal SRAM	
40 0000h–47 FFFFh	512K	External SRAM	External memory zone 1. Timing values set by software.
80 0000h–9F FFFFh	2M	External Flash	External memory zone 2. Timing values set by software.
FF FB00h–FF FBFFh	256	I/O Expansion	Off-chip peripherals. Must be activated by software. See I/O expansion table.
FF FC00h–FF FFFFh	1K	Internal I/O	On-chip peripherals

12.3 I/O Expansion Decoding

A 64-byte region of the address space between FF FB00h and FF FBBFh is further decoded to provide eight select lines. Each select line is asserted when its corresponding 8-byte region is addressed, as described in Table 12-4. One select line is used to control the 8-bit latch that drives the LEDs. All select lines are available on peripheral connector B.

Table 12-4. I/O Expansion Decoding Map

Address Range	Size (bytes)	Contents	Comments
FF FBB8h–FF FBBFh	8	LED Latch	Any write to this region loads the 8-bit LED latch from bits 7:0 of the data bus. LD1 corresponds to data bus bit 0, LD2 corresponds to bit 1, LD3 to bit 2, etc.
FF FBB0h–FF FBB7h	8	Select Line Q6	Available on peripheral connector B.
FF FBA8h–FF FBAFh	8	Select Line Q5	
FF FBA0h–FF FBA7h	8	Select Line Q4	
FF FB98h–FF FB9Fh	8	Select Line Q3	
FF FB90h–FF FB97h	8	Select Line Q2	
FF FB88h–FF FB8Fh	8	Select Line Q1	
FF FB80h–FF FB87h	8	Select Line Q0	

12.4 Jumpers

Table 12-5 lists the jumper-selected configuration options.

Table 12-5. Jumper Configuration Options

Jumper	Description
J1	Installed – X1CKO is connected to external crystal network. Uninstalled – X1CKO is floating.
J2	Installed – UVCC is connected to 3.3V supply rail. Uninstalled – UVCC is isolated from 3.3V supply rail.
J3	Installed – Pushbutton S1 is connected to NMI. Uninstalled – NMI is floating (may be driven from peripheral connector B).
J5	Installed on USB side – required for CP3BT10 and CP3UB17. Installed on CAN side – required for CP3BT13 and CP3CN17.
J6	Installed on USB side – required for CP3BT10 and CP3UB17. Installed on CAN side – required for CP3BT13 and CP3CN17.
J7	Installed on USB side – required for CP3BT10 and CP3UB17. Installed on CAN side – required for CP3BT13 and CP3CN17.
J8	Installed on USB side – required for CP3BT10 and CP3UB17. Installed on CAN side – required for CP3BT13 and CP3CN17.
J9	Installed on USB side – required for CP3BT10 and CP3UB17. Installed on CAN side – required for CP3BT13 and CP3CN17.
J10	Installed on USB side – required for CP3BT10 and CP3UB17. Installed on CAN side – required for CP3BT13 and CP3CN17.
J11	Installed – X1CKI is connected to external crystal network. Uninstalled – X1CKI is not connected to external crystal network (may be connected to oscillator module).
J12	Installed – X1CKI is connected to oscillator module. Uninstalled – X1CKI is not connected to oscillator module (may be connected to external crystal network).
J13	Do not install.
J14	Installed – Pushbutton S2 is connected to WUI9. Uninstalled – WUI9 is floating (may be driven from peripheral connector A).

Table 12-5. Jumper Configuration Options (Continued)

Jumper	Description
J15	Installed on 0 side – ENV0 pin is pulled low (0). Installed on 1 side – ENV0 pin is pulled high (1).
J16	Installed on 0 side – ENV1 pin is pulled low (0). Installed on 1 side – ENV1 pin is pulled high (1).
J17	Installed on 0 side – ENV2 pin is pulled low (0). Installed on 1 side – ENV2 pin is pulled high (1).
J19	Installed toward oscillator module – Zone 2 mapped to peripheral connector. Installed away from oscillator module – Zone 2 mapped to external Flash.
J20	Installed toward oscillator module – Zone1 mapped to peripheral connector. Installed away from oscillator module – Zone 1 mapped to external SRAM.
J21	Installed toward oscillator module – Zone 1 or Zone 2 mapped to peripheral connector. Installed away from oscillator module – I/O Zone mapped to peripheral connector.
J22	Installed – CAN termination resistor is in the circuit. Uninstalled – CAN termination resistor is out of the circuit.
J25	Installed on LPT side – LPT debugging interface enabled. Installed on Nexus side – Nexus debugging interface enabled.
J26	UART configuration jumpers. See Section 12.5.6 for details.
J27	
J29	
J30	
J31	Installed – Audio codec is enabled. Uninstalled – Audio codec is in power-down mode.
J32	Installed – Audio amplifier is enabled. Uninstalled – Audio amplifier is in shutdown mode.
J33	Do not install.
J35	Installed – 5V regulator output connected to 5V supply rail. Uninstalled – 5V supply rail isolated.
J36	Installed – 3.3V regulator output connected to 3.3V supply rail. Uninstalled – 3.3V supply rail isolated.

Table 12-5. Jumper Configuration Options (Continued)

Jumper	Description
J37	Installed – 2.5V regulator output connected to 2.5V supply rail. Uninstalled – 2.5V supply rail isolated.
J4	Installed on USB side – required for CP3BT10 and CP3UB17. Installed on CAN side – required for CP3BT13 and CP3CN17.
J42	Installed – DIP switch S3 bit 1 is connected to PH0. Uninstalled – PH0 is not connected to DIP switch.
J43	Installed – DIP switch S3 bit 2 is connected to PH1. Uninstalled – PH1 is not connected to DIP switch.
J44	Installed – DIP switch S3 bit 3 is connected to PH2. Uninstalled – PH2 is not connected to DIP switch.
J45	Installed – DIP switch S3 bit 4 is connected to PH3. Uninstalled – PH3 is not connected to DIP switch.
JP3	Installed – EEPROM address bit 2 is 0. Uninstalled – EEPROM address bit 2 is 1.
JP4	Installed – EEPROM address bit 1 is 0. Uninstalled – EEPROM address bit 1 is 1.
JP5	Installed – EEPROM address bit 0 is 0. Uninstalled – EEPROM address bit 0 is 1.
JP6	Installed – EEPROM is in write-protect mode. Uninstalled – EEPROM writes are allowed.

12.5 System Configuration

Jumpers configure the following interfaces, peripherals, memory, and logic:

- DIP Switches
- Pushbutton Switches
- 8 Programmable LEDs (3 additional LEDs indicate power status)
- Clock Source
- External Audio Codec and Audio Amplifier
- On-chip UART
- On-chip USB (CP3UB17 only)
- On-chip CAN (CP3CN17 only)
- External Flash and SRAM
- Host PC Interface

12.5.1 DIP Switches

Four positions of DIP switch S3 are connected directly to Port H on the CP3UB17/CP3CN17 device through jumpers. Table 12-6 shows the jumpers which configure these switches. Installing these jumpers connects the switches to the corresponding port pins.

Table 12-6. DIP Switch Configuration Jumpers

Switch	CP3000 Pin	Configuration Jumper
S3 Position 1	PH0	J42
S3 Position 2	PH1	J43
S3 Position 3	PH2	J44
S3 Position 4	PH3	J45

The remaining four positions are available on header SW above DIP switch S3. The signals on the header are pulled to ground when the corresponding switches are closed, and 10k-ohm pullup resistors pull the signals to the 3.3V supply rail when the switches are open.

12.5.2 Pushbutton Switches

Two pushbutton switches are connected to CP3UB17/CP3CN17 device pins through jumpers. Jumper J14 is used to isolate pushbutton S1 from the NMI pin. Jumper J3 is used to isolate pushbutton S2 from the WUI9 pin.

12.5.3 Programmable LEDs

Eight programmable LEDs are controlled by the CP3UB17/CP3CN17 device through a latch. A write to any address in the range from FF FBB8h to FF FBBFh loads the latch from the low 8 bits of the data bus. Software must configure the I/O Zone before accessing it.

12.5.4 Clock Source

The clock to the CP3UB17/CP3CN17 device may come from either of the sources shown in Table 12-7

Table 12-7. Clock Source Jumper Configurations

Clock Source	Installed Jumpers	Uninstalled Jumpers
Crystal Network	J1, J11	J12, J13
Oscillator Module	J12	J1, J11, J13

12.5.5 Operating Environment

The operating environment of the CP3UB17/CP3CN17 device is selected by jumpers J15, J16, and J17. Each of these jumpers has two positions, marked on the evaluation board as 0 and 1. Table 12-8 shows the positions of the jumpers used to select the operating environment.

Table 12-8. Operating Environment Jumper Configurations

Operating Environment	Jumper J17 (ENV2)	Jumper J16 (ENV1)	Jumper J15 (ENV0)
DEV	0	0	0
IRE	1	1	1
ERE	0	1	1
ISP	0	1	0

12.5.6 UART Configuration

Jumpers are used to configure the signals between the UART and connector P3 for a crossover or straight-through configuration. The serial cable provided in the evaluation kit requires a crossover configuration.

The UART configuration jumpers are in two sets of adjacent two-pin headers, as shown in Figure 12-1.

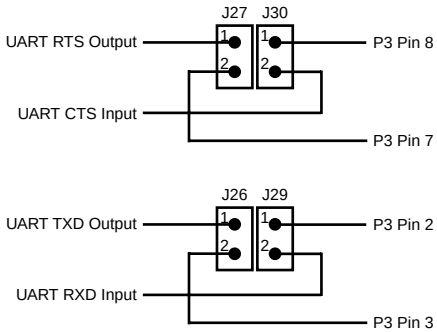


Figure 12-1. UART Configuration Jumpers

Jumpers are installed either between pins on the same header or between the same pin positions on adjacent headers. Table 12-9 shows the jumper configurations.

Table 12-9. UART Jumper Configurations

Crossover	Straight-Through
J26 Pin 1 - J29 Pin 1 J26 Pin 2 - J29 Pin 2 J27 Pin 1 - J30 Pin 1 J27 Pin 2 - J30 Pin 2	J26 Pin 1 - J26 Pin 2 J27 Pin 1 - J27 Pin 2 J29 Pin 1 - J29 Pin 2 J30 Pin 1 - J30 Pin 2

12.5.7 Host PC Interface

The host PC interfaces to the evaluation board through a DB25 parallel port connector (P1). To enable this interface, jumper J25 must be installed on the LPT side.

12.6 Connectors

The evaluation board has several connectors for development tools, power, peripheral I/O, and communication interfaces, as listed in Table 12-10.

Table 12-10. Connectors

Connector	Description
PWR59	DC power connector (screw terminals)
PWR62	DC power jack for modular power supply
JP1	JTAG/Nexus connector
P1	Parallel port connector for interface to the host PC
J34	2.5 mm audio jack
P3	DB9 (female) for CAN interface
P2	DB9 (male) for UART interface
T1	USB connector

12.7 USB Configuration

The evaluation board can be configured for USB development with the following steps:

- Install a CP3UB17 processor into the CP3000 device socket. **To remove the socket, gently pry the metal side tabs on each side of the socket. Be careful not to damage surface mount parts around the socket. Note the orientation of the CPU before removing it from the holder.**
- Install jumpers J4, J5, J6, J7, J8, J9, and J10 on the USB side of the jumper block.
- Install jumper J2.
- Configure the evaluation board to use a crystal network or oscillator module clock source.

12.8 CAN Configuration

The evaluation board can be configured for CAN development with the following steps:

- Install a CP3CN17 processor into the CP3000 device socket. **To remove the socket, gently pry the metal side tabs on each side of the socket. Be careful not to damage surface mount parts around the socket. Note the orientation of the CPU before removing it from the holder.**
- Install jumpers J4, J5, J6, J7, J8, J9, and J10 on the CAN side of the jumper block.
- If it is desired to take the 120-ohm termination resistor out of the network interface, remove jumper J22.
- Configure the evaluation board to use a crystal network or oscillator module clock source.

12.9 Audio Circuit

The Advanced Audio Interface (AAI) on the CP3UB17/CP3CN17 device is connected to an on-board external codec. Install jumper J31 to enable the codec. The audio jack J34 is driven by the codec through an audio amplifier. Install jumper J32 to enable the amplifier.

12.10 External Memory Bus

The CP3UB17/CP3CN17 device implements two static zones that can be mapped to an external memory bus or the peripheral connector. Zone 1 can be mapped to 512K bytes of SRAM. Zone 2 can be mapped to 2M bytes of flash memory. Either of these zones (but not both simultaneously) can also be mapped to the peripheral connector for off-board memory expansion. The mapping of the static zones is controlled by jumpers. Each jumper has two positions, as described in Table 12-11.

Table 12-11. Memory Configuration Jumpers

Memory Configuration	Jumper J19 Position	Jumper J20 Position	Jumper J21 Position
Zone 1 = SRAM Zone 2 = Flash	Away from oscillator module	Away from oscillator module	Away from oscillator module
Zone 1 = SRAM Zone 2 = Peripheral Connector	Toward oscillator module	Away from oscillator module	Toward oscillator module
Zone 1 = Peripheral Connector Zone 2 = Flash	Away from oscillator module	Toward oscillator module	Toward oscillator module

Evaluation Board

External Memory Bus

Software Vendor Contacts

iSYSTEM GmbH
Carl-Zeiss-Strasse 1
D-85247 Schwabhausen
Germany

+49(8138)6971-0
+49(8138)6971-46 (fax)

iSYSTEM USA
16776 Bernado Center Drive
Suite 204A
San Diego, CA 92128, USA

+1(858)385-9100
+1(888)543-5300
+1(858)385-9119 (fax)

iSYSTEM Asia-Pacific Ltd.
7F Uniana Building 907-18
Mok5-dong, Yangchon-gu
158-055 Seoul Korea

+82(2)2645-0386
+82(2)2649-8290 (fax)

National Semiconductor
2900 Semiconductor Drive
PO Box 58090
Santa Clara, CA 95052

Tel: 1-800-272-9959
Fax: 1-800-737-7018

Visit our Web site at::
www.national.com

For more information, send Email to:
support@nsc.com

National Semiconductor Europe

Fax:	+49 (0) 180-530 85 86
Email:	europe.support@nsc.com
Deutsch Tel:	+49 (0) 69 9508 6208
English Tel:	+44 (0) 870 24 0 2171
Francais Tel:	+33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-254-4466
Fax: 65-250-4466
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507