



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



EMBEDDED ETHERNET DEVELOPMENT KIT USER'S GUIDE

1. Overview

The Embedded Ethernet Development Kit (Ethernet-DK) provides all the hardware and software required to develop real-world embedded Ethernet applications using the industry proven CMX Micronet™ protocol stack and high-performance Silicon Laboratories microcontrollers. The TCP/IP protocol stack is developed specifically for embedded processors and is freely distributed in an easy-to-use library. A TCP/IP Configuration Wizard is provided to generate a highly customized library optimized for user selected protocols. It also generates the framework code required to use the library and a project file that can be managed in the Silicon Laboratories Integrated Development Environment (IDE). Additional information about the TCP/IP library and Configuration Wizard, including a detailed API description, can be found in application note “AN237: TCP/IP Library Programmer’s Guide”.

The Embedded Ethernet Development Kit hardware includes a C8051F120 Target Board, AB4 Ethernet Development Board, USB Debug Adapter, and all necessary cables to debug the MCU and connect it to an Ethernet network. The C8051F120 Target Board features the 100 MIPS, 128 kB Flash, 8 kB RAM, 8051-based C8051F120 MCU. The MCU network interface provided by the AB4 board consists of the CP2200 Ethernet controller and an RJ-45 connector with integrated magnetics and LEDs. The TCP/IP Library includes a built-in driver for the CP2200 and has a custom driver interface with auto-generated templates to allow drivers for any Ethernet controller to be written.

The Silicon Labs IDE supports full-speed, non-intrusive MCU debugging and is bundled with an evaluation version of the Keil C51 Toolchain allowing immediate application code evaluation in C. Projects with up to 4 kB of object code and unlimited library code can be developed using the included toolset. Numerous application code examples are included in the development kit and a walkthrough of an embedded web server demo is included in Section 6 on page 9.

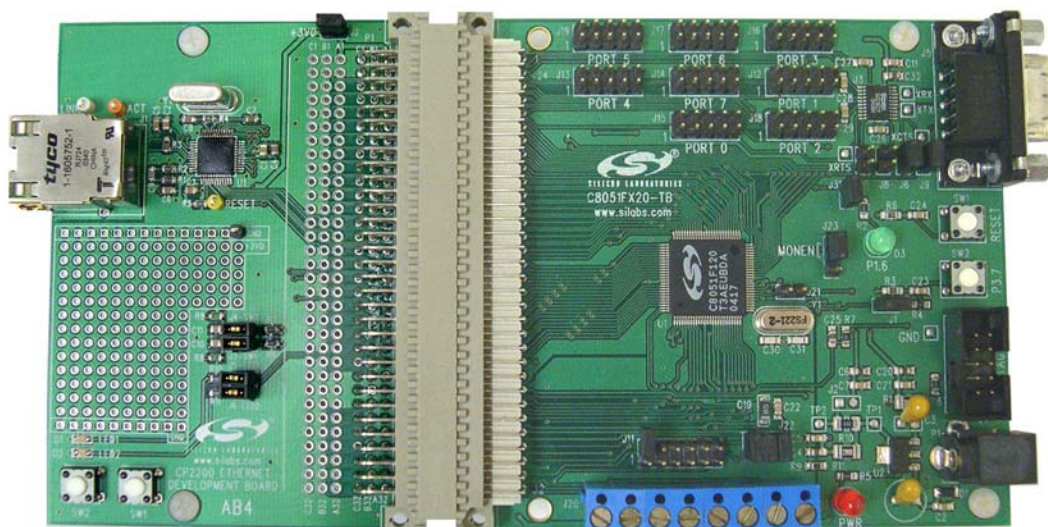


Figure 1. C8051F120TB Target Board and AB4 Ethernet Development Board

Ethernet-DK

2. Kit Contents

The Embedded Ethernet Development Kit contains the following items:

- C8051F120 Target Board
- AB4 Ethernet Development Board
- CAT5e Ethernet Cable
- Silicon Laboratories Evaluation Kit IDE and Product Information CD-ROM. CD content includes the following:
 - Silicon Laboratories Integrated Development Environment (IDE)
 - Keil C51 Evaluation Toolchain (assembler, linker and C compiler with 4 kB object code generation and unlimited library code usage)
 - Source code examples and register definition files
 - Documentation
 - TCP/IP Configuration Wizard
 - Embedded Ethernet Development Kit User's Guide (this document)
- Universal (100–240 V, 50–60 Hz) AC to DC Power Adapter
- USB Debug Adapter (USB to JTAG/C2 Debug Interface)
- USB Cable
- Serial Cable

3. Hardware Setup

The following instructions illustrate how to setup the hardware included with the kit.

1. Connect the **AB4 Ethernet Development Board** to the **C8051F120 Target Board** at J24 (Figure 2). Apply slight pressure to ensure the mating 96-pin connectors are firmly connected.
2. Using the ribbon cable, connect the **USB Debug Adapter** to the **C8051F120 Target Board** at the JTAG header as shown in Figure 3.
3. Connect one end of the provided USB cable to any available USB port on the **PC** (Figure 3).
4. Connect the other end of the USB cable into the USB connector on the **USB Debug Adapter** (Figure 3).
5. Connect the **ac/dc Power Adapter** to the **C8051F120 Target Board** at P1 (Figure 3). This connection should power both boards. Power is indicated by the "PWR" LED on the C8051F120 TB.

Note: The 'F120TB places a $2\ \Omega$ resistor in series with the power supply net powering the AB4 Board. We recommend replacing this resistor with a $0\ \Omega$ resistor to prevent large voltage drops, possibly triggering a V_{DD} Monitor reset, when current demand increases.

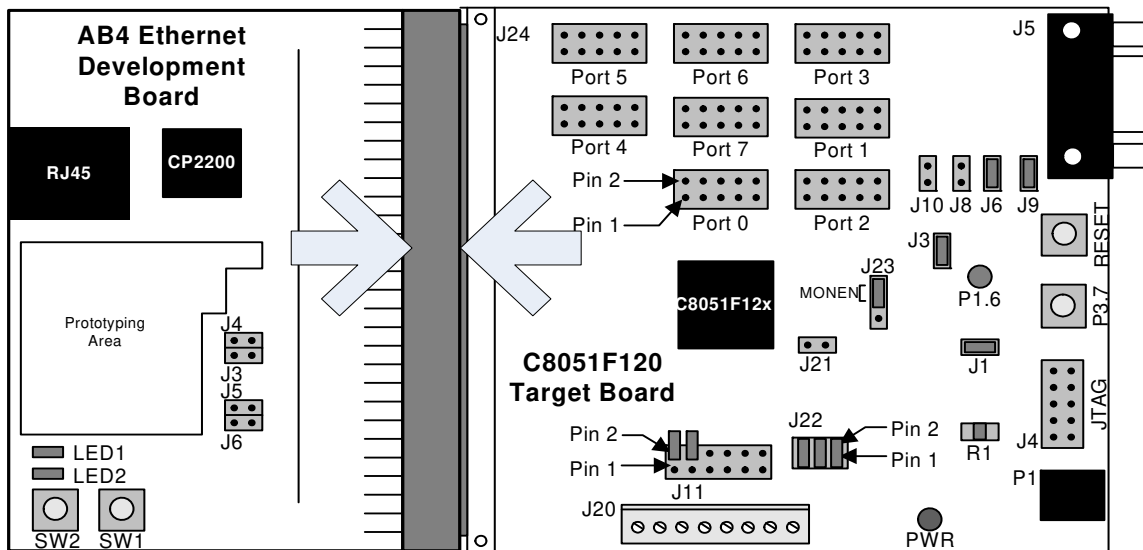


Figure 2. Embedded Ethernet Development Board Attachment

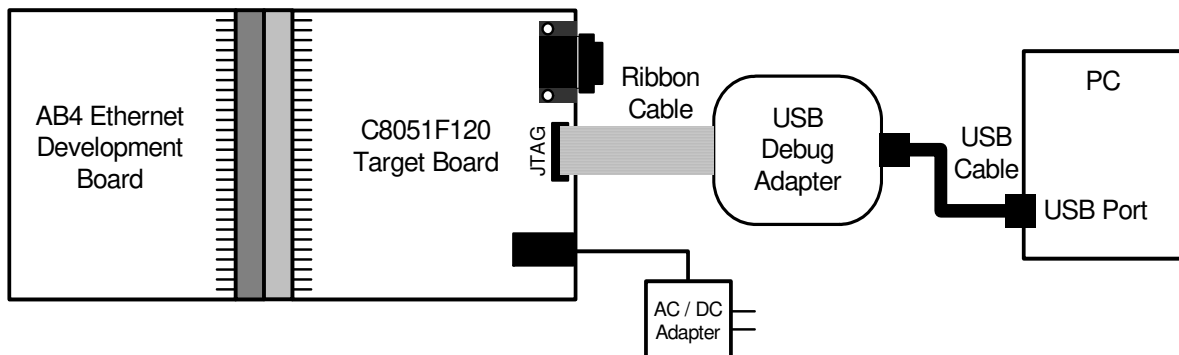


Figure 3. Embedded Ethernet Development Kit Debug Connections

Ethernet-DK

4. Network Setup

The Embedded Ethernet Development Kit can be connected to an Ethernet network using a standard Ethernet cable (see Figure 4) or directly to a PC using a crossover cable (see Figure 5). Table 1 describes the benefits of using each of the connection methods. A standard Ethernet cable is included in the kit and crossover cables are available for order from the Silicon Laboratories website, at www.silabs.com.

Note: Firewalls or VPN policies may need to be disabled in order to communicate with the development kit.

Table 1. Ethernet Cable Comparison

Standard Cable	Crossover Cable
Remote access to embedded system.	Distance from PC to embedded system is limited by the length of the cable.
Multiple embedded systems may be networked and accessed from the same (or multiple) PC(s).	Only a single embedded system may be accessed from a single PC.
Shared communication medium. System may experience packet delays under heavy network traffic. This effect is greatly reduced if using a switched network.	Dedicated channel bandwidth. Good for achieving consistent data throughput measurements.
An existing network with wall outlet or a router/switch is required.	No additional hardware required.
PC does not require any additional IP address configuration.	PC must be configured with a static IP address in order to recognize embedded system.
Standard cable is included in development kit.	Crossover cable must be purchased separately.

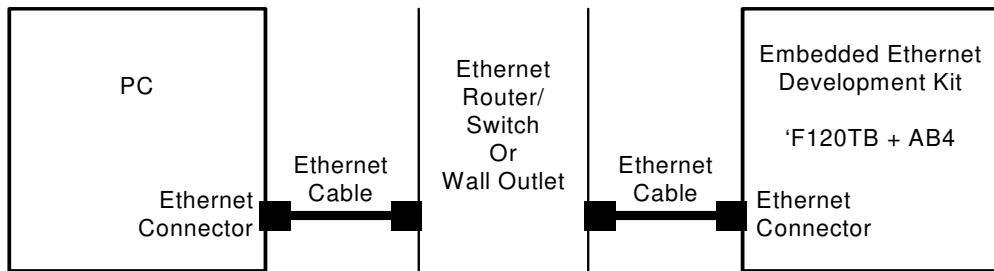


Figure 4. Embedded Ethernet Network Connection (Standard Cable)

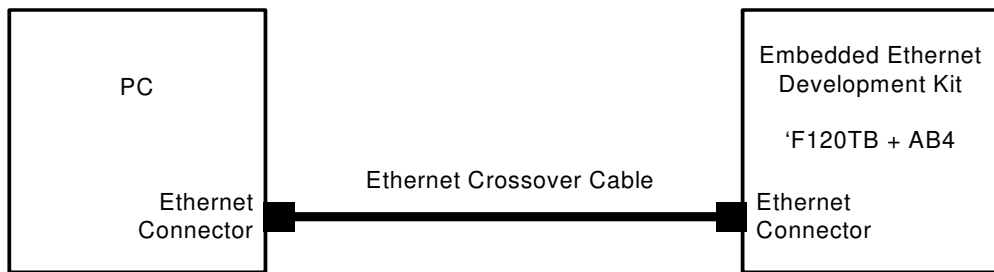


Figure 5. Embedded Ethernet Network Connection (Crossover Cable)

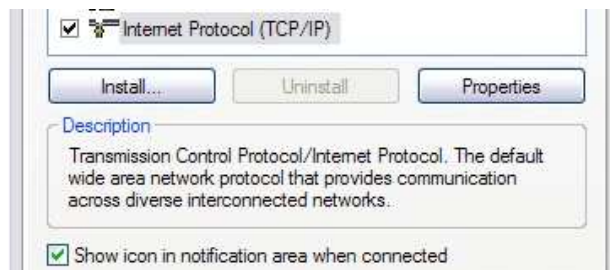
4.1. Network Setup Procedure

If using a standard cable:

1. Connect the **AB4 Ethernet Development Board** to an Ethernet wall outlet or to a router/switch using a standard Ethernet cable.
2. Connect a **PC** to the same Ethernet network using a standard Ethernet cable.

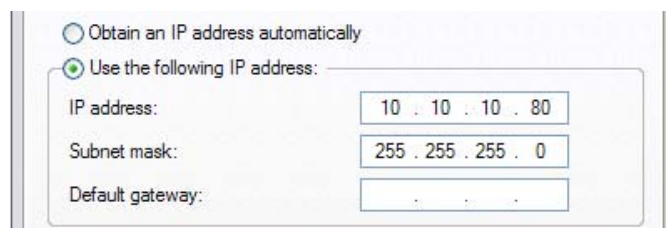
If using a crossover cable:

1. Connect the **AB4 Ethernet Development Board** directly to a **PC** using an Ethernet crossover cable.
2. Configure the **PC** to have a static IP address. The steps below show how to configure a Windows PC to use a static IP address.
 - a. Open the *Network Connections* Folder. This is accessible from the control panel or from the start menu by right-clicking on *My Network Places* or *Network* and selecting *Properties*.
 - b. Right-click on the *Local Area Connection* and select *Properties*.
 - c. Select *Internet Protocol (TCP/IP)* and click *Properties*. Optionally, if the *Show icon in notification area when connected* is checked, right-clicking the system tray icon will provide a quick shortcut to the *LAN properties* dialog and can be used as an indicator that the network is functioning properly.



- d. Select *Use the following IP address* and specify an IP address and subnet mask for the PC. For a subnet mask of 255.255.255.0, the embedded system's IP address must match the first three octets of the PC's IP address for the PC to recognize the embedded system. See Section 4.2 for additional information about choosing the embedded system's IP address.

Note: When entering an IP or subnet address into the dialog box, the cursor will automatically advance to the next field for a three digit octet. If entering a one or two digit octet, the spacebar, right arrow key, or '.' can be pressed to advance to the next field.



- e. Click *OK* after the static IP address and subnet mask have been configured. The PC will now be able to access the embedded system using a crossover cable.

Ethernet-DK

4.2. Selecting an IP Address for the Embedded System

For a PC to recognize an embedded system on a network, its IP address and subnet mask need to be configured. Below are a few guidelines to follow when choosing an IP address for the embedded system. Figure 6 shows an example of a compatible PC and embedded system IP address combination.

1. Find the IP address and Subnet mask for the PC. If a Default Gateway is specified, then save this address for later use. If you are using a crossover cable, you may choose any IP address for your PC as long as the Subnet mask allows it to recognize the embedded system.
2. The IP address chosen for the embedded system must match the PC's IP address in all bit locations where the Subnet mask is a 1 in order for the PC to recognize the embedded system. Otherwise, the PC will send it's request outside the local network.
3. Do not duplicate IP addresses or select a broadcast address. An IP address is considered a broadcast address if all bits which are 0 in the Subnet mask are 1 in the IP address. Broadcast addresses with additional 1s such as 10.10.255.255 (Figure 6) can be broadcast to nodes outside the local network.
4. The address 255.255.255.255 is known as the Ethernet broadcast address and is used when the Subnet mask for the network is not known. Any packet transmitted to this address will reach all nodes on the local network but cannot go further than the nearest router.

4.3. IP Address Selection Example

The example in Figure 6 shows the IP address and subnet mask of the PC we want to connect to the embedded system. Since the first 24 bits of the subnet mask are 1, the first 24 bits of the embedded web server's IP address (shown in bold) must match the PC's IP address. Only the least significant 8 bits may vary making the valid range of IP addresses for the embedded web server 10.10.10.0 to 10.10.10.254 with the exception of 10.10.10.80 since this address is already taken by the PC. 10.10.10.255 is reserved because it is the broadcast address for this network.

PC IP Address

10	10	10	80	(decimal)
0000 1010	0000 1010	0000 1010	0101 0000	(binary)

PC Subnet Mask

255	255	255	0	(decimal)
1111 1111	1111 1111	1111 1111	0000 0000	(binary)

Embedded Web Server IP Address

10	10	10	163	(decimal)
0000 1010	0000 1010	0000 1010	1010 0011	(binary)

Figure 6. IP Address Selection Example

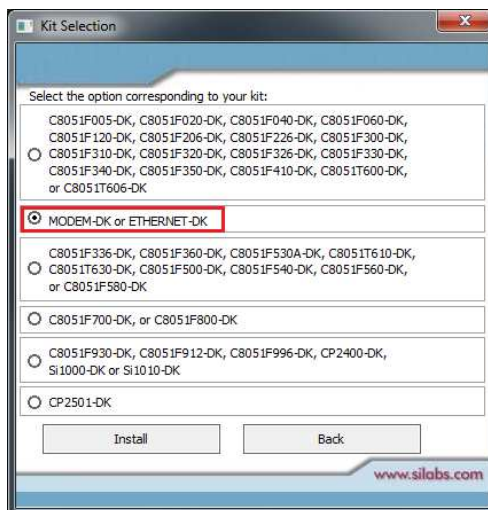
5. Software Setup

The included CD-ROM contains the Silicon Laboratories IDE, Keil C51 toolset, and documentation including datasheets, application notes, and an electronic version of this user's guide. The instructions below describe how to install the Embedded Ethernet Development Kit software. Refer to the *readme.txt* file on the CD-ROM for the IDE release notes containing the latest information regarding supported devices, revision history, and known issues.

1. Place the Embedded Ethernet Development Kit CD-ROM into the PC's CD-ROM drive. An installer will automatically launch, allowing you to install the IDE software or read documentation by clicking buttons on the Installation Panel. If the installer does not automatically start when you insert the CD-ROM, run *autorun.exe*, found in the root directory of the CD-ROM.
2. After the installation dialogue box appears, click the **Install Development Tools** button.

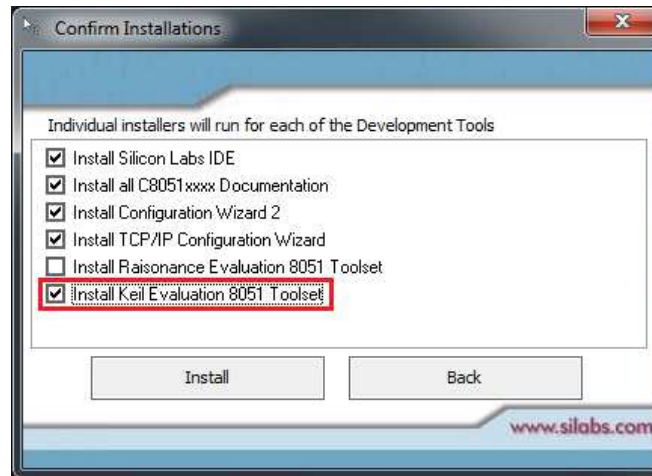


3. Select the ETHERNET-DK entry and press the **Install** button.



Ethernet-DK

4. Unselect the “Install Raisonance Evaluation 8051 Toolset” option and select “Install Keil Evaluation 8051 Toolset”. Then press the **Install** button.



5. Follow the installation prompts to install the development tools. The following applications will be installed:
- The Silicon Laboratories IDE will be installed by default in the *C:\SiLabs\MCU* directory.
 - The Embedded Ethernet examples will be installed by default in the *C:\SiLabs\MCU\Examples\C8051F12x\Ethernet* directory.
 - The TCP/IP Configuration Wizard will be installed by default in the *C:\SiLabs\MCU\TCP-IP Config* directory.
 - Shortcuts to the applications will be placed in the **Start**→**Programs** menu.
6. To compile the projects discussed in this tutorial, the Keil tools must be enabled to the 4k-code limited evaluation version. See “AN104: Integrating Keil 8051 Tools into the Silicon Labs IDE” for more information. Application notes can be found on the Silicon Labs Application Notes webpage: <http://www.silabs.com/products/mcu/Pages/ApplicationNotes.aspx>.

6. Embedded Ethernet Tutorial

Now that the Embedded Ethernet Development Kit hardware has been set up, the software installed, and the embedded system connected to a network, it is time to download firmware into the MCU and test its network connectivity. The Embedded Ethernet Tutorial consists of four stages of increasing functionality and will allow the user to quickly learn how to use the development kit hardware and software.

6.1. Stage 1—Basic Web Server

In the first demo stage, we will generate a “Hello World” web server with a single static web page using the TCP/IP Configuration Wizard. We will learn how to PING the embedded web server and view the page in a web browser.

6.1.1. Generating a Project using the TCP/IP Configuration Wizard

The instructions below show how to generate a new project using the TCP/IP Configuration Wizard.

1. From the Windows Start menu, start the **TCP/IP Configuration Wizard**. The shortcut will be under the **Start→Silicon Laboratories** menu. Alternatively, the Configuration Wizard can be started by double-clicking *TCPIP_Config.exe* located by default in the “*C:\SiLabs\MCU\TCP-IP Config*” directory.
2. In the **Communications Adapter** section of the left window, check **CP2200**.
3. In the **Device** section of the left window, verify that **C8051F12x** is selected.
4. In the **Protocol Settings** section of the left window, make sure **Ethernet** is selected then check **HTTP**. The lower level protocols required to run HTTP (e.g., TCP) will automatically be enabled (see Figure 7).

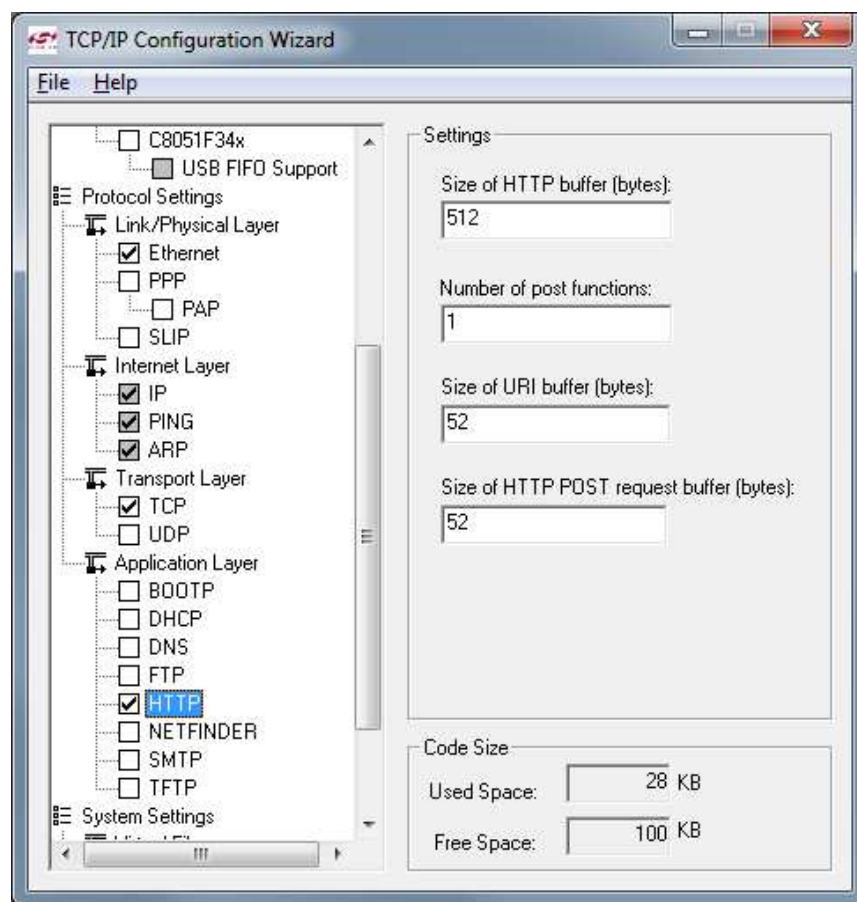


Figure 7. TCP/IP Configuration Wizard Protocol Selection

Ethernet-DK

5. In the **System Settings** section of the left window, select **IP Addresses**.
6. In the right window, set the **Source IP Address** to the IP address of the embedded system.
7. If the **Gateway IP Address** and **Subnet mask** of the network are known, then fill in the addresses on the right window (see Figure 8).
8. If using the CP2200, the MAC address field is ignored. The CP2200 Ethernet Controller contains a unique factory-programmed MAC address stored in Flash memory.

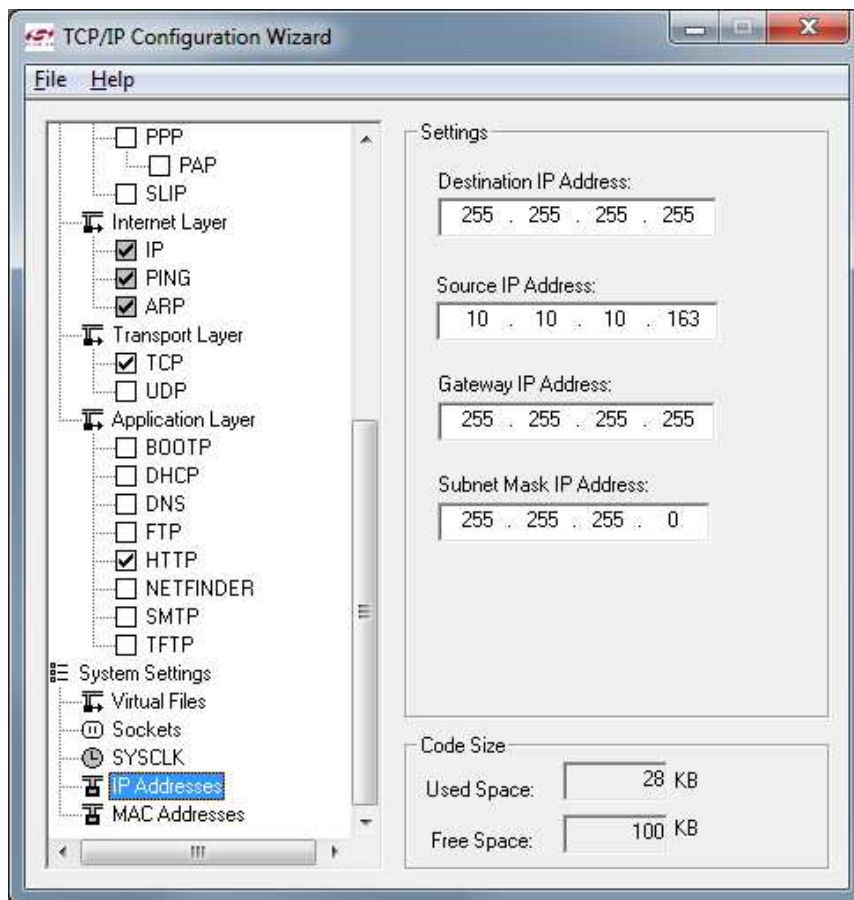


Figure 8. TCP/IP Configuration Wizard IP Address Specification

9. Save the selected configuration using the **File**→**Save As** menu.
10. Generate a new project with supporting firmware by selecting **Generate Project** from the **File** menu (see Figure 9).
11. When prompted for a folder to save the project, browse to an empty directory or create a new directory.
Note: Any files in the selected directory will be overwritten. Click the **OK** button.
12. Click the **OK** button when the “Project generated successfully” message appears.
13. Close the TCP/IP Configuration Wizard.

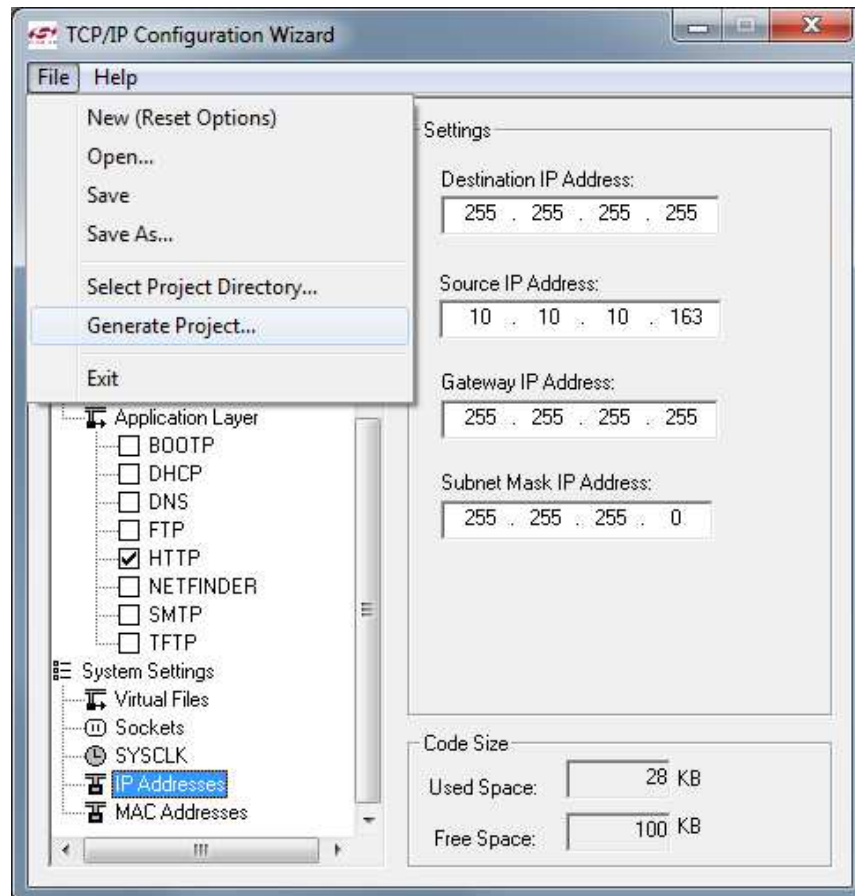


Figure 9. TCP/IP Configuration Wizard Project Generation

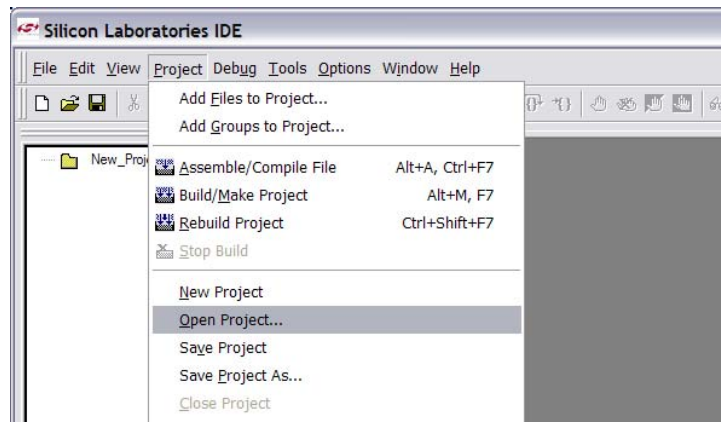
Ethernet-DK

6.1.2. Programming the MCU

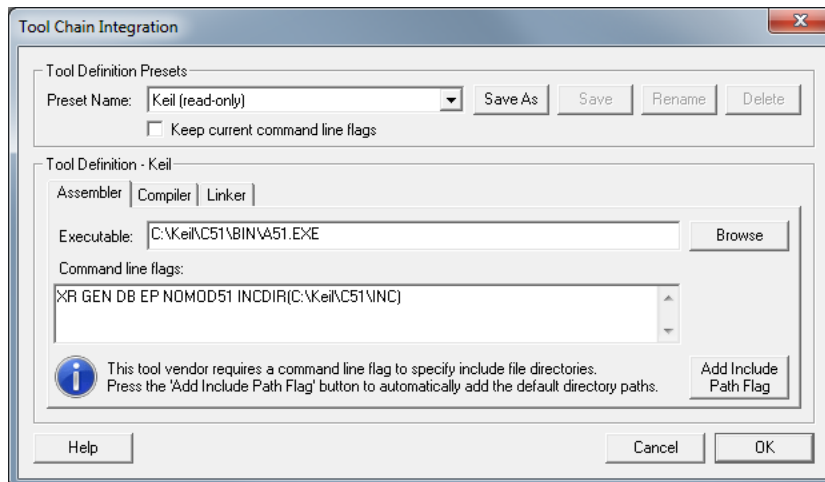
We will now build the project created by the TCP/IP Configuration Wizard and download the firmware to the MCU using the Silicon Laboratories IDE.

1. From the Windows Start menu, start the **Silicon Laboratories IDE**. The shortcut will be under the **Start**→**Silicon Laboratories** menu. Alternatively, the Silicon Labs IDE can be started by double-clicking *IDE.exe* located by default in the “C:\SiLabs\MCU” directory.
2. After the splash screen disappears, open the generated project using the **Project**→**Open Project** menu command. Browse to the directory where the TCP/IP project was generated. Select *TCPIP_Project.wsp*, and click the **Open** button.

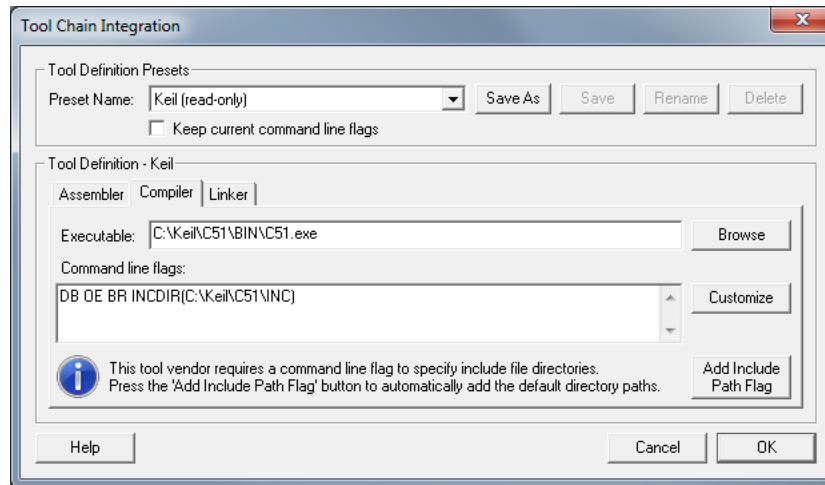
Note: An example project is available by default in the *C:\SiLabs\MCU\Examples\C8051F12x_13x\Ethernet\HTTP\web-server-1* directory. This project is identical to the one generated in the previous steps.



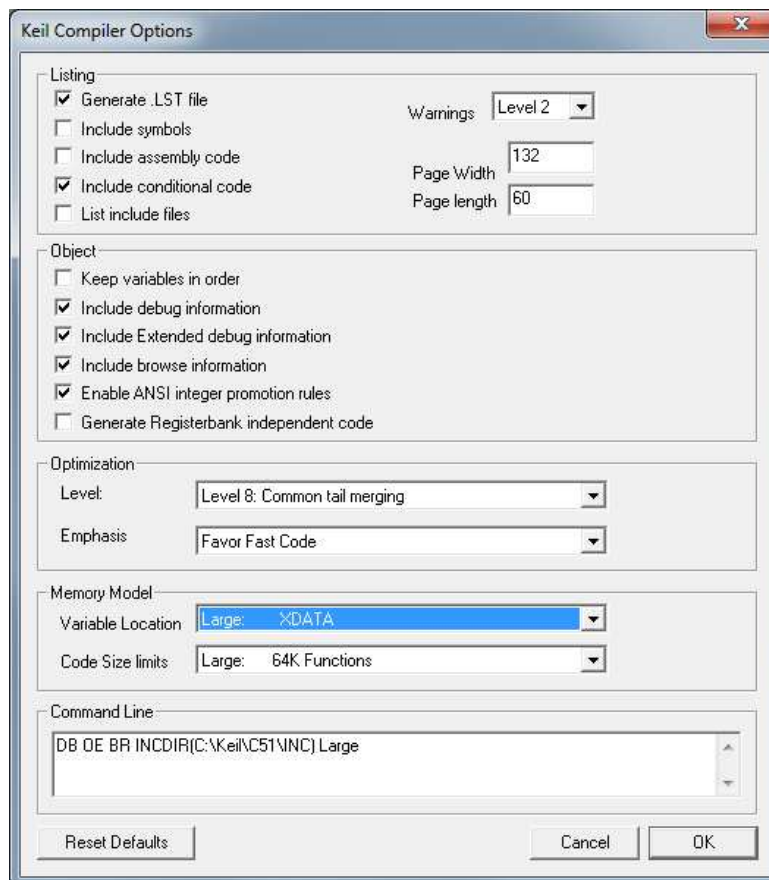
3. If you would like to view the source code, double-click on *main.c* on the left hand side of the screen in the Project Window. Source code modification is not required to run this demo.
4. Select the Keil tools by going to **Project**→**Tool Chain Integration** and selecting the Keil preset under the **Preset Name** drop down menu. Press No when the **Save Preset?** dialog appears.



5. Verify the Compiler and Linker tabs are pointing to the C51.exe (compiler) and BL51.exe (linker) files in *C:\Keil\C51\BIN* in the **Executable** field.




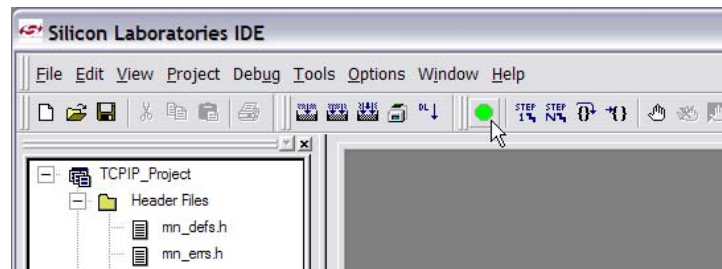
6. Press the Compiler tab's **Customize** button. Under the **Variable Location** drop down box, select **Large: XDATA** and press **OK**. Then press **OK** to close the Tool Chain Integration dialog.



7. Build the project. This can be done by clicking on the **Build/Make Project** button in the toolbar, selecting **Project**→**Build/Make Project** from the menu, or pressing the **F7** shortcut key.

Ethernet-DK

8. Click the **Connect** button in the toolbar or select **Debug**→**Connect** from the menu.
Note: If you receive the error message “Communication could not be established with the specified serial adapter”, open the **Connection Options Dialog** by selecting **Options**→**Connection Options** from the menu. Verify that the **USB Debug Adapter** is selected and that the debug interface is set to **JTAG**.
9. Download the project to the target by clicking the **Download Code** button  in the toolbar.
10. Run the demo by clicking the **Go** button in the toolbar, selecting **Debug**→**Go** from the menu, or pressing the shortcut key **F5**.



6.1.3. Testing Connectivity Using PING

After pressing **GO**, the MCU will be executing the TCP/IP stack firmware. The LEDs on the **RJ-45 Ethernet connector** should light up if the MCU is connected to a network. Once the network router detects the presence of the embedded system, it will be accessible from any PC connected to the same network. This usually takes a few seconds, but can take over 30 seconds depending on the size of the network. The delay is minimized if the embedded system is connected directly to a PC using a crossover cable.

The PING utility is available on all PCs and allows the user to check if the MCU is responding to network requests. The instructions below show how to ping the embedded system.

1. Using the Windows Start menu, select **Start→Run** and type **cmd** when the **Run** dialog appears. Alternatively, type **cmd** in the **Search programs and files** field. A black terminal window should appear.
2. At the prompt, type **ping** followed by a **space** and the **IP address** of the embedded system as shown in Figure 10. Press the **Enter** key when finished.
3. If the PC is able to communicate with the embedded system, then ping will report the number of milliseconds it took to receive a reply from the embedded system. Otherwise, it will report "Request timed out".

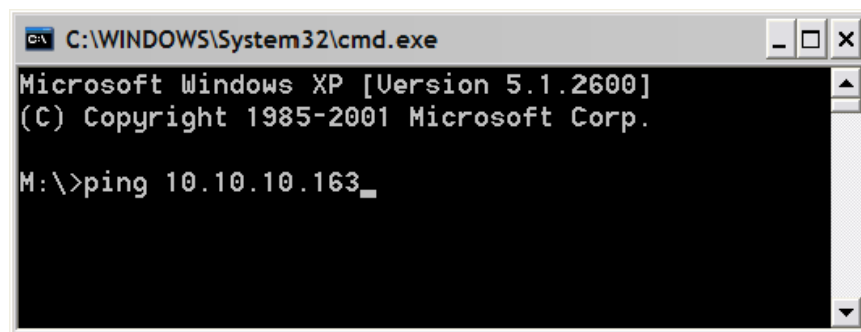


Figure 10. Pinging the Embedded System

6.1.4. Accessing the Embedded System using a Web Browser

Once the ping utility is able to receive a reply from the embedded system, the embedded system is now ready to be accessed from a web browser. The instructions below show how to access the embedded HTTP web server.

1. Open an instance of your favorite web browser.
2. Type in the IP address of the embedded system and press Enter. A web page stored inside the MCU should be displayed on the screen as shown in Figure 11.

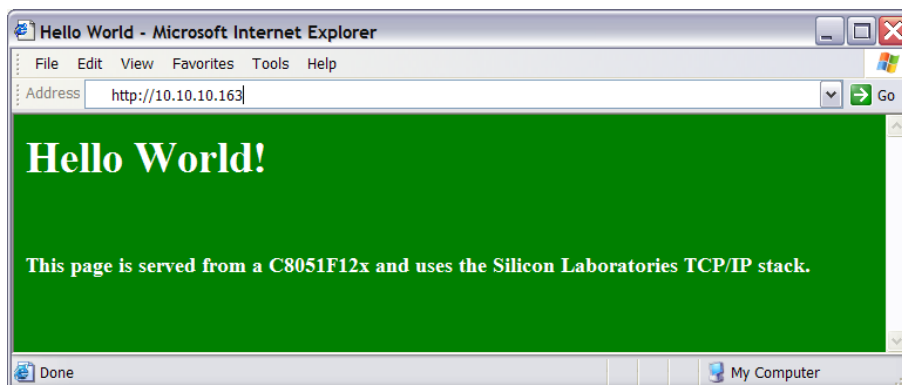


Figure 11. Embedded Web Server

6.2. Stage 2—Basic Web Server with DHCP and Netfinder Capability

In the first stage of the demo, we learned how to generate firmware for an embedded web server using the TCP/IP Configuration Wizard and access the embedded server from a web browser. In this stage, we will learn how to generate a web server that uses automatic network configuration (DHCP) to automatically obtain an IP address from the network. We will also find the embedded web server on the network using the Netfinder utility.

Note: The Netfinder utility v1.0 is not compatible with Windows Vista or Windows 7.

6.2.1. Generating the Embedded Web Server Firmware

We will now use the TCP/IP Configuration Wizard to generate a project with DHCP and Netfinder capability.

1. From the Windows Start menu, start the **TCP/IP Configuration Wizard**. The shortcut will be under the **Start→Silicon Laboratories** menu. Alternatively, the Configuration Wizard can be started by double-clicking *TCPIP_Config.exe* located by default in the “C:\SiLabs\MCU\TCP-IP Config” directory.
2. In the **Communications Adapter** section of the left window, check **CP2200**.
3. In the **Device** section of the left window, verify that **C8051F12x** is selected.
4. In the **Protocol Settings** section of the left window, make sure **Ethernet** is selected then check **DHCP**, **HTTP**, and **NETFINDER**. The lower level protocols (TCP and UDP) will automatically be enabled as shown in Figure 12.

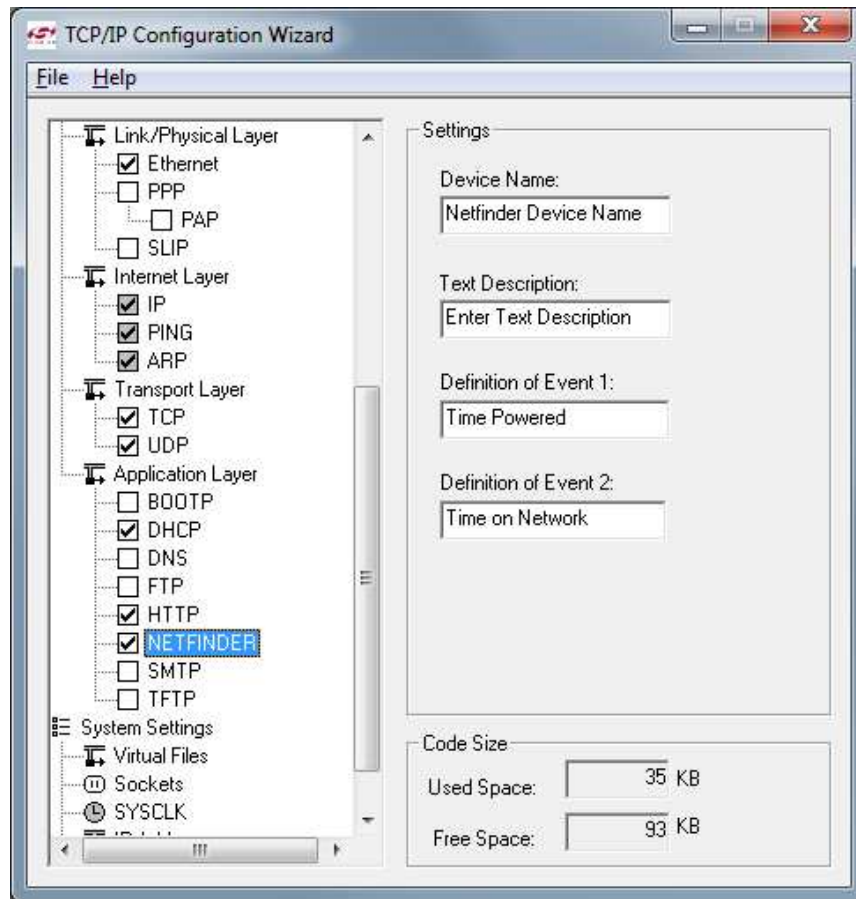


Figure 12. TCP/IP Configuration Wizard Protocol Selection

5. Save the selected configuration using the **File**→**Save As** menu.
6. Generate a new project with supporting firmware by selecting **Generate Project** from the **File** menu (see Figure 13).
7. When prompted for a folder to save the project, browse to an empty directory or create a new directory.
Note: Any files in the selected directory will be overwritten. Click the **OK** button.
8. Click the **OK** button when the “Project generated successfully” message appears.
9. Close the TCP/IP Configuration Wizard.

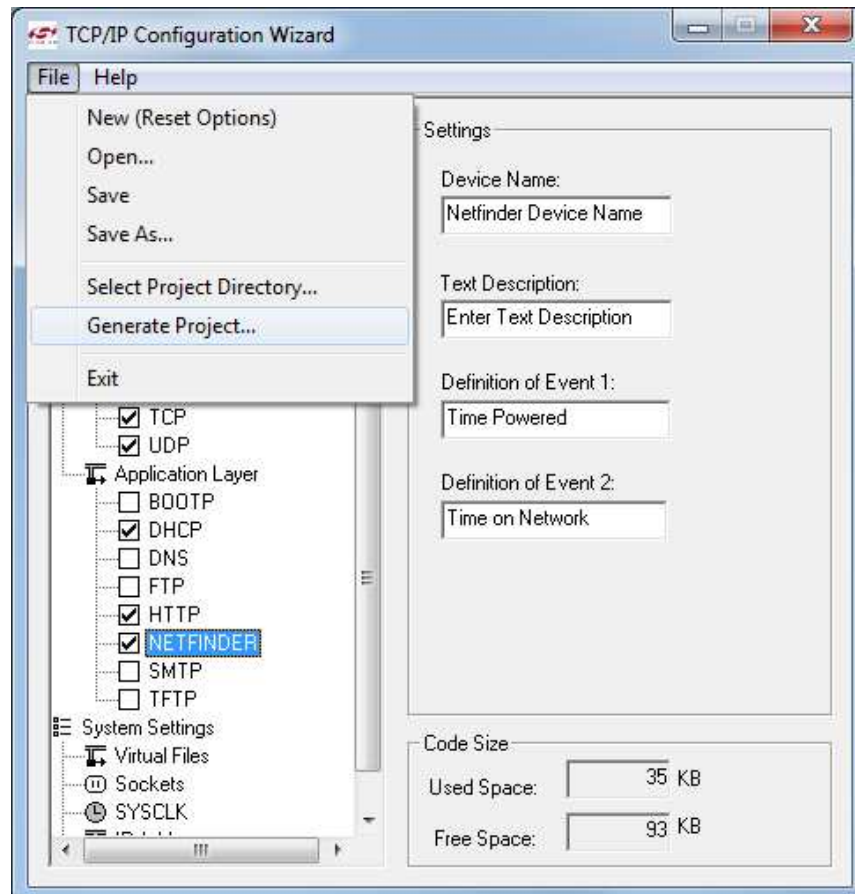


Figure 13. TCP/IP Configuration Wizard Project Generation

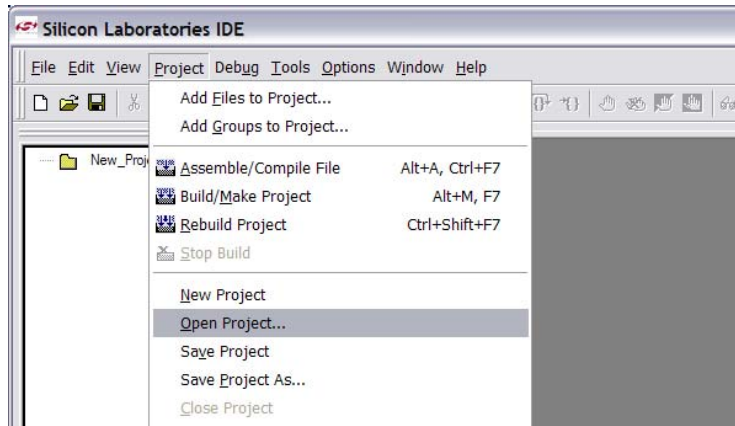
Ethernet-DK

6.2.2. Programming the MCU

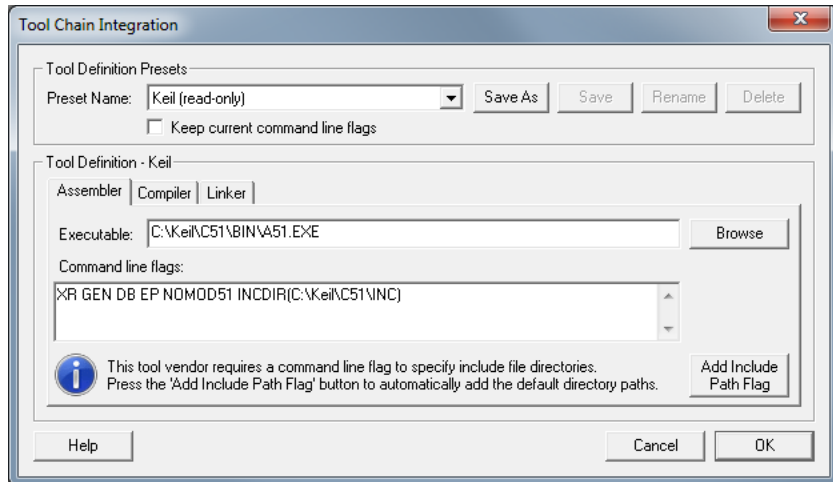
We will now build the project created by the TCP/IP Configuration Wizard and download the firmware to the MCU using the Silicon Laboratories IDE.

1. From the Windows Start menu, start the **Silicon Laboratories IDE**. The shortcut will be under the **Start**→**Silicon Laboratories** menu. Alternatively, the Silicon Labs IDE can be started by double-clicking *IDE.exe* located by default in the “*C:\SiLabs\MCU*” directory.
2. After the splash screen disappears, open the generated project using the **Project**→**Open Project** menu command. Browse to the directory where the TCP/IP project was generated. Select *TCPIP_Project.wsp*, and click the **Open** button.

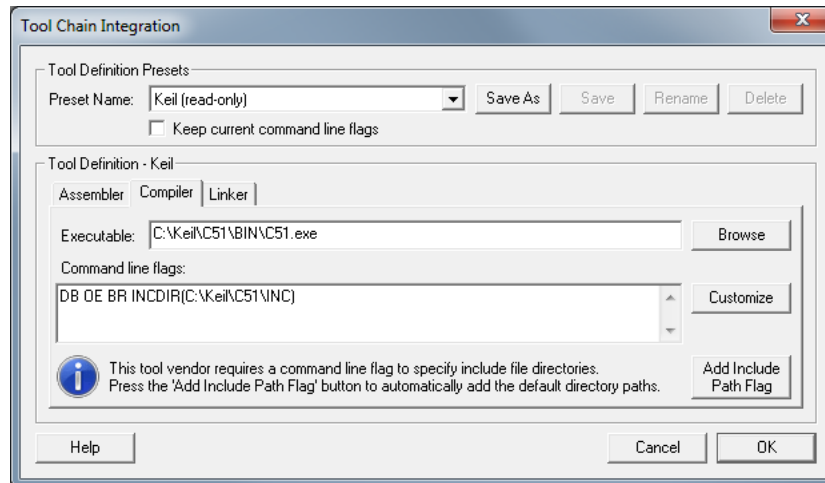
Note: An example project is available by default in the *C:\SiLabs\MCU\Examples\C8051F12x_13x\Ethernet\HTTP\web-server-2* directory. This project is identical to the one generated in the previous steps.



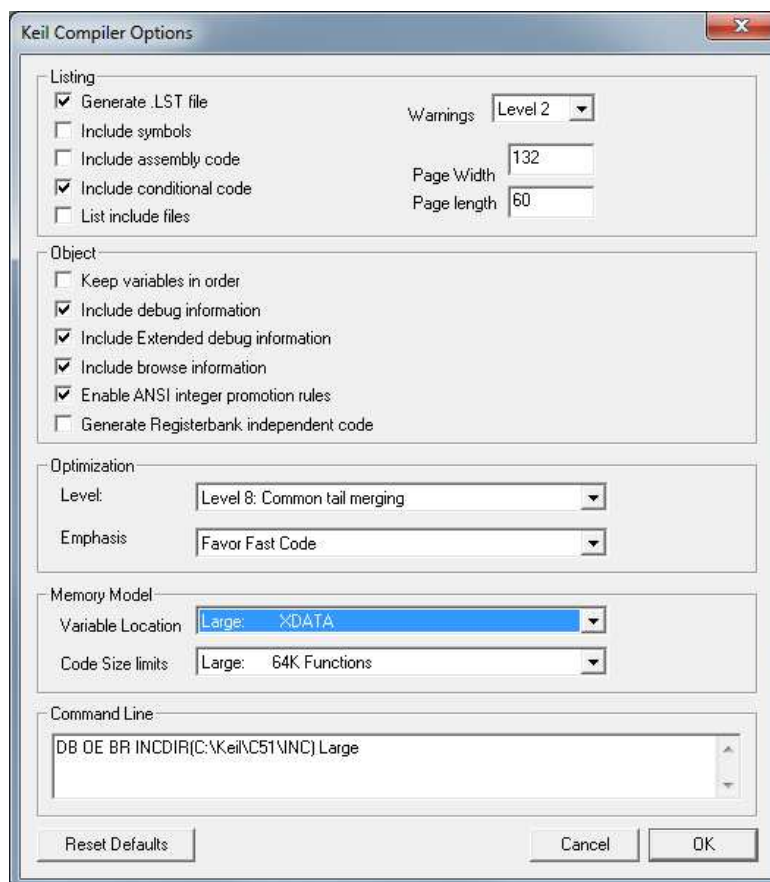
3. If you would like to view the source code, double-click on *main.c* on the left hand side of the screen. Source code modification is not required to run this demo.
4. Select the Keil tools by going to **Project**→**Tool Chain Integration** and selecting the Keil preset under the **Preset Name** drop down menu. Press No when the **Save Preset?** dialog appears.



5. Verify the Compiler and Linker tabs are pointing to the C51.exe (compiler) and BL51.exe (linker) files in `C:\Keil\C51\BIN` in the **Executable** field.




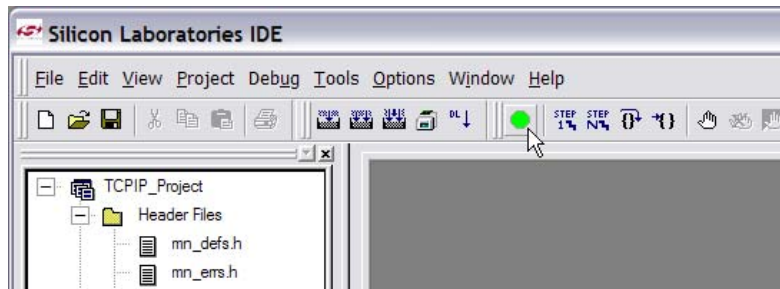
6. Press the Compiler tab's **Customize** button. Under the **Variable Location** drop down box, select **Large: XDATA** and press **OK**. Then press **OK** to close the Tool Chain Integration dialog.



7. Build the project. This can be done by clicking on the **Build/Make Project** button in the toolbar, selecting **Project**→**Build/Make Project** from the menu, or pressing the **F7** shortcut key.

Ethernet-DK

8. Click the **Connect** button in the toolbar or select **Debug**→**Connect** from the menu.
Note: If you receive the error message “Communication could not be established with the specified serial adapter”, open the **Connection Options Dialog** by selecting **Options**→**Connection Options** from the menu. Verify that the **USB Debug Adapter** is selected and that the debug interface is set to **JTAG**.
9. Download the project to the target by clicking the **Download Code** button  in the toolbar.
10. Run the demo by clicking the **Go** button in the toolbar, selecting **Debug**→**Go** from the menu, or pressing the shortcut key **F5**.



6.2.3. Finding the Embedded System

We will now search for embedded systems on the network.

Note: The Netfinder utility v1.0 is not compatible with Windows Vista or Windows 7.

1. Open the **Netfinder** utility found in *C:\SiLabs\MCU\Utilities\Netfinder*.
2. Press the **Search** button as shown in Figure 14 to begin searching for devices.

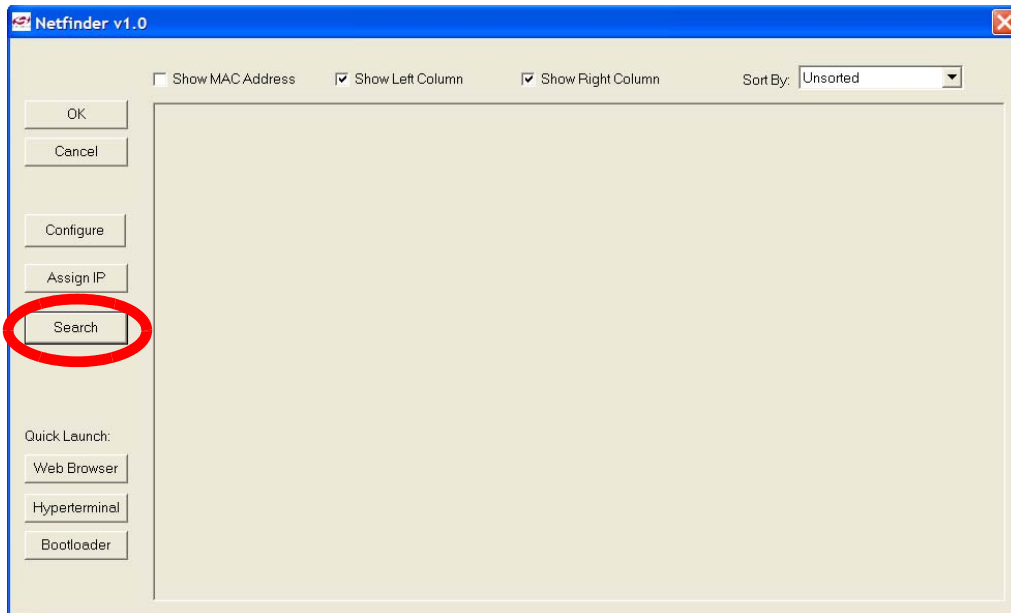


Figure 14. Search Button

3. The Embedded System should appear inside a box with a green background. If it appears inside a box with a yellow background, then a DHCP server was not found. Please skip directly to step 5.
4. Select the embedded system by clicking anywhere inside the green area. After the black border appears, press the **Web Browser** Quick Launch Button.

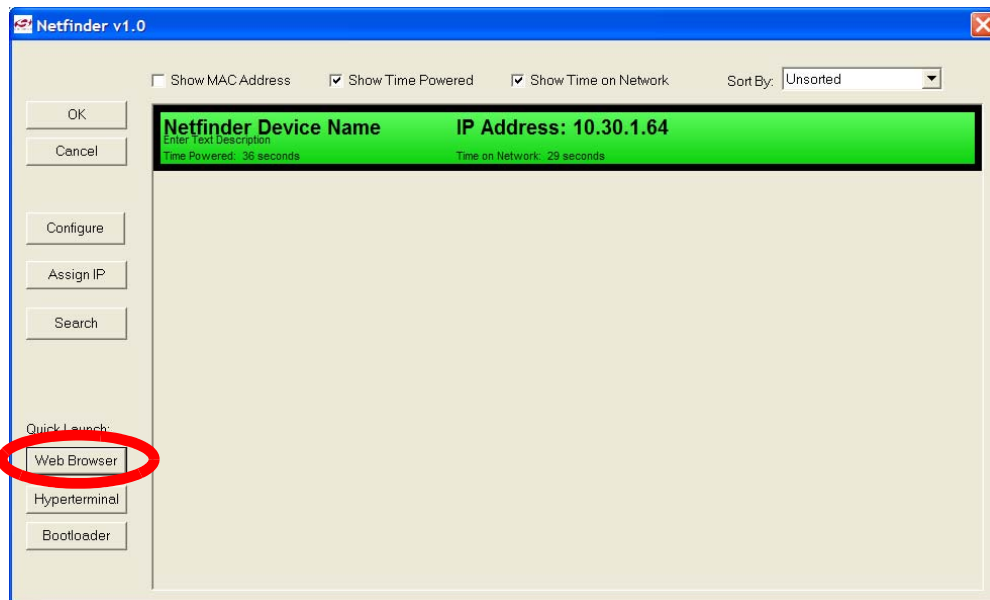


Figure 15. Web Browser Quick Launch Button

Ethernet-DK

5. The default web browser should launch and display the home page served by the embedded web server. The same result can be achieved by typing the IP address of the embedded web server into the address bar of a web browser. Figure 16 shows a screenshot of the home page.
- Now go directly to Section "6.3. Stage 3—Configuring MAC and IP Addresses over the Serial Port" on page 25.**

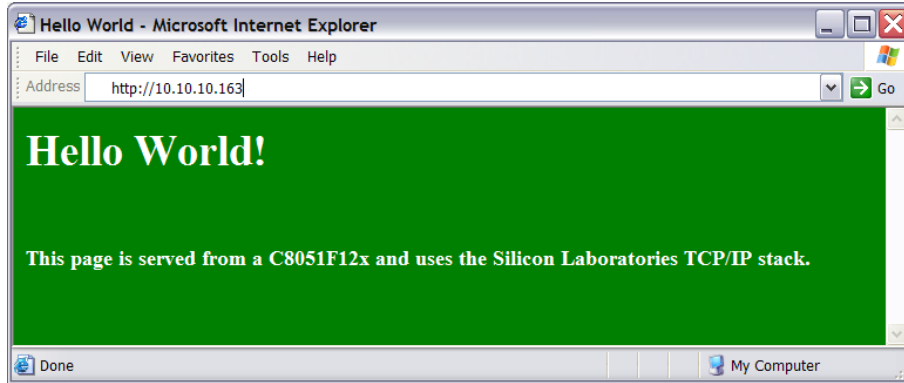


Figure 16. Hello World Web Page

6. If the network does not have a DHCP server, then the embedded system will search for approximately 35 seconds before going into static IP address mode. After going into static IP address mode, Netfinder will display information about the device in a window with a yellow background. Select the device by **clicking inside the yellow area** and **pressing the Assign IP button** as shown in Figure 17.

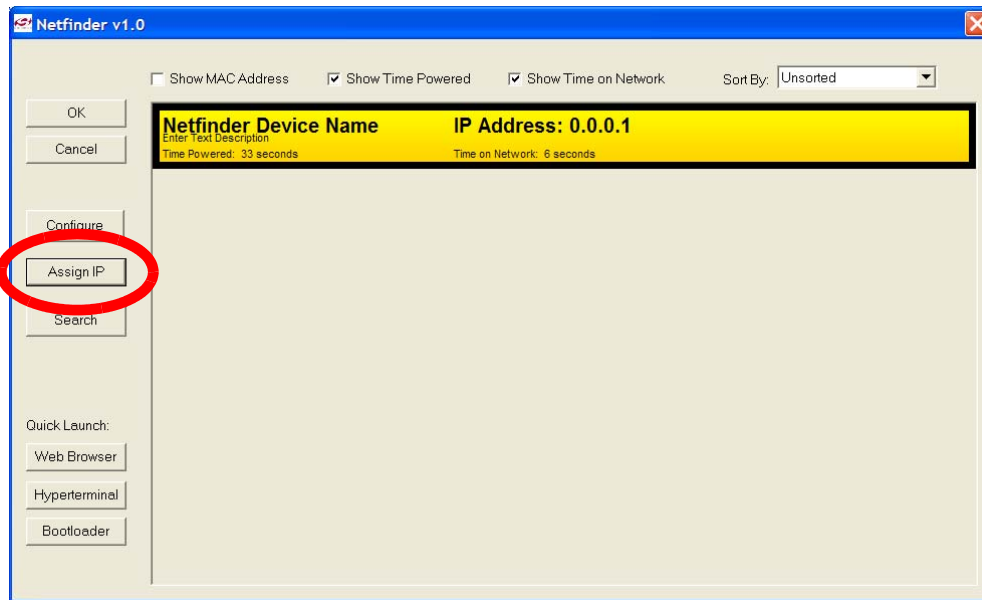


Figure 17. Static IP Address Mode

7. Enter the IP address, subnet mask, and default gateway (if known) into the 'Assign IP Address' dialog and press **OK**. See Appendix A for information on how to configure the PC with a static IP address and how to select a static IP address for the CP2201EB. A windows command line window should appear, ping the device at its new address, then close. Programming is successful when the progress bar stops and gives the message "Success - Device Programmed" as shown in Figure 18.

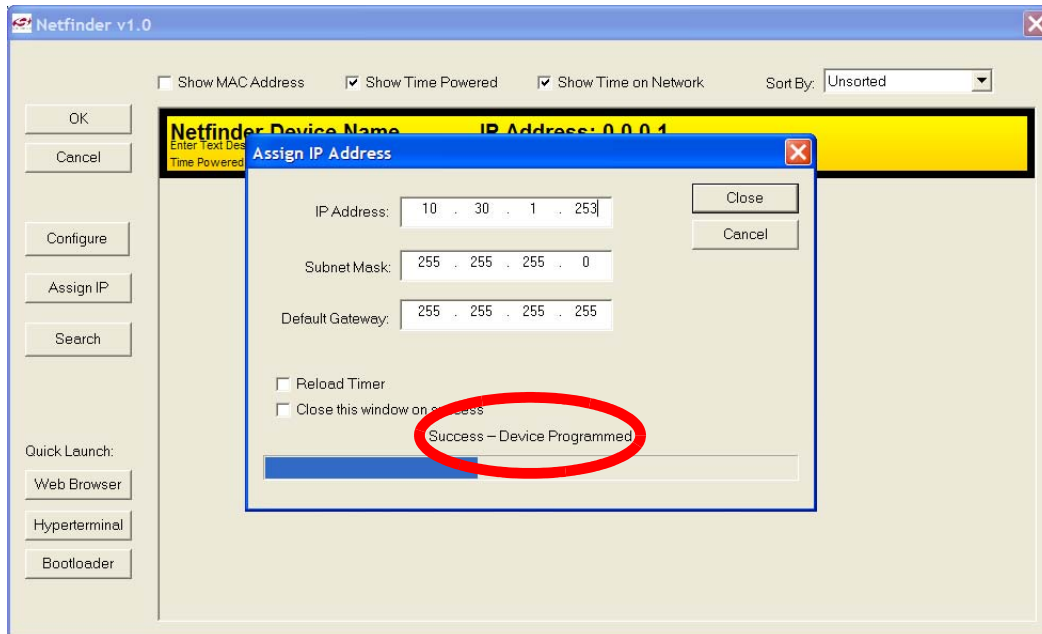


Figure 18. Assign IP Address Dialog

8. Close the 'Assign IP Address' dialog then press **Search** to refresh the Netfinder screen.
9. Select the device by **clicking inside the yellow area** and **pressing the Web Browser quick launch button** as shown in Figure 19.

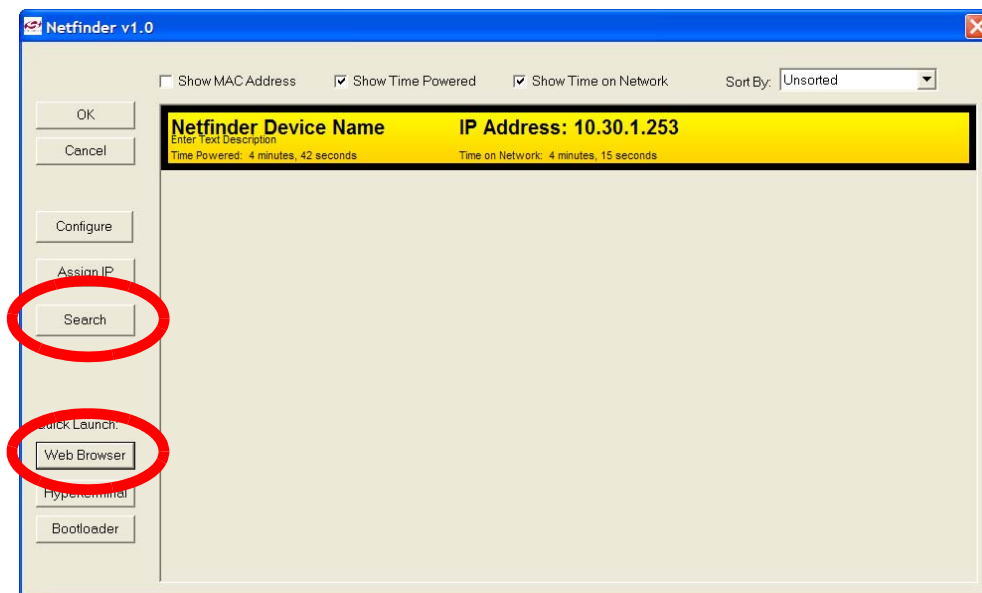


Figure 19. Web Browser Quick Launch Button In Static IP Address Mode

Ethernet-DK

10. The default web browser should launch and display the home page served by the embedded web server. The same result can be achieved by typing the IP address of the embedded web server into the address bar of a web browser. Figure 20 shows a screenshot of the uWeb-v2.0 home page.



Figure 20. Hello World Web Page

6.3. Stage 3—Configuring MAC and IP Addresses over the Serial Port

In the first two stages of the demo, we learned how to generate firmware for an embedded web server using the TCP/IP Configuration Wizard and access the embedded server from a web browser. In this stage, we will learn how to dynamically configure the IP address over the serial port.

6.3.1. Configuring the Target Board Jumpers for UART communication.


To enable UART communication on the C8051F120 Target Board, jumpers J6 and J9 must be shorted. See Figure 27 for the location of J6 and J9.

6.3.2. Modifying the Firmware to Enable the Serial Port

The simple web server firmware generated by the TCP/IP Configuration Wizard does not support configuring the IP address over the serial port. In order to configure the MCU using the serial port, the firmware must be modified. The *web-server-3* project in the examples folder has been created to demonstrate an example of configuration using the serial port. The following list outlines the functionality that has been added to *web-server-3*:

- VDD Monitor is enabled as a reset source because firmware needs to write/erase Flash memory.
- UART1 Initialization routine uses Timer 1 for baud rate generation.
- UART1 is assigned to P0.0 and P0.1 in the Crossbar.
- The *stdio.h* header file is included to allow use of string formatting functions such as *printf()*.
- The *ipconfig()* routine is used to configure the MAC address.
- The global *ip_address[4]* variables is added in the Scratchpad area of Flash to store the programmed MAC Address. The Scratchpad area is used to store the variable because it can be erased and re-written without interfering with program code.
- The global *first_time* variable is stored in Flash to keep track of whether the IP address has been programmed.

The following instructions show how to load the *web-server-3* project:

1. Halt the MCU by clicking the **Stop** button in the toolbar, selecting **Debug**→**Stop** from the menu, or pressing the shortcut key **F4**.
2. Close the current open project using the **Project**→**Close Project** menu command.
3. Open the *web-server-3* project using the **Project**→**Open Project** menu command. The *web-server-3* project workspace is located by default in *C:\SiLabs\MCU\Examples\C8051F12x_13x\Ethernet\HTTP\web-server-3* directory.
4. Build the project. This can be done by clicking on the **Build/Make Project** button in the toolbar, selecting **Project**→**Build/Make Project** from the menu, or pressing the **F7** shortcut key.
5. If not already connected, click the **Connect** button in the toolbar or select **Debug**→**Connect** from the menu.
Note: If you receive the error message “Communication could not be established with the specified serial adapter”, open the **Connection Options Dialog** by selecting **Options**→**Connection Options** from the menu. Verify that the **USB Debug Adapter** is selected and that the debug interface is set to **JTAG**.
6. Download the project to the target by clicking the **Download Code** button  in the toolbar.
7. Run the demo by clicking the **Go** button in the toolbar, selecting **Debug**→**Go** from the menu, or pressing the shortcut key **F5**.