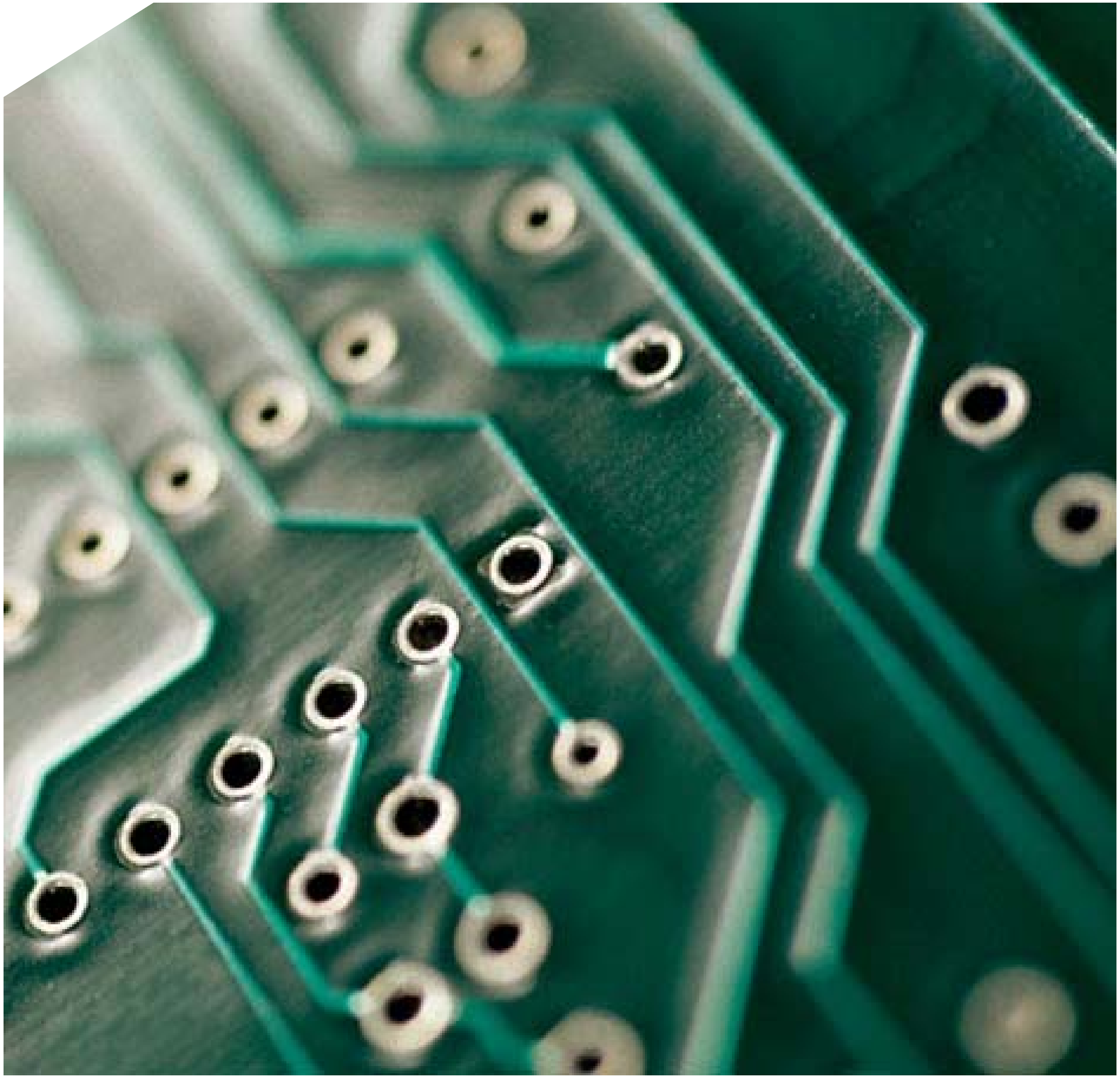
Hands-on Workshop: Dynamic Web Page Server with the MCF5223X Family (AZ131)

Freescale Technological Forum
July 2006

Juan Morales / Eric Gregori
Product Marketer / Software Field Application Engineer

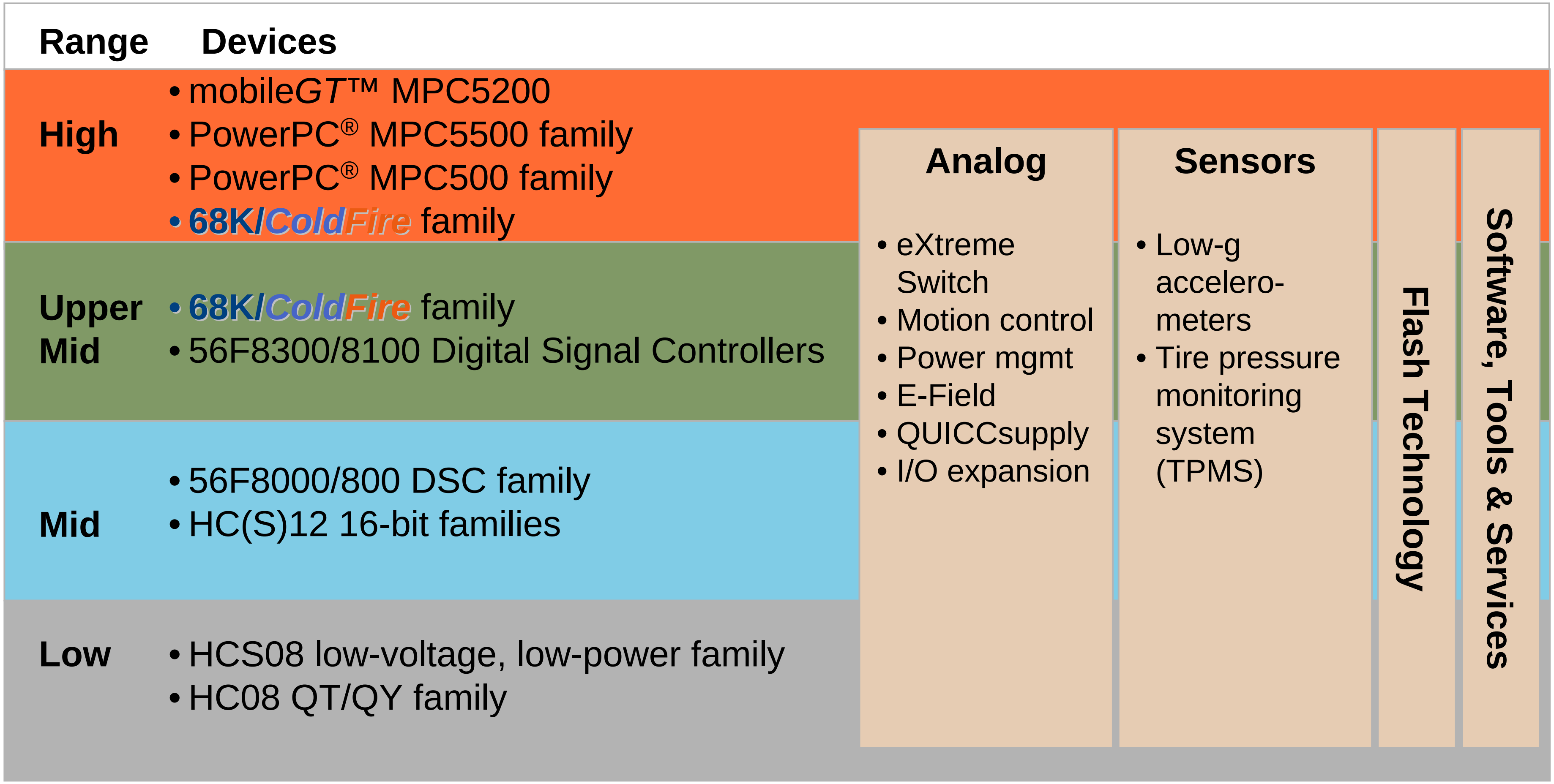


MCF522xx Family Overview

MCU Ethernet/USB Connectivity

freescale™ semiconductor

Controller Continuum
A full range of products, technology, services, and tools



Range	Devices
High	• mobile*GT*™ MPC5200 • PowerPC® MPC5500 family • PowerPC® MPC500 family • 68K/*ColdFire* family
Upper Mid	• 68K/*ColdFire* family • 56F8300/8100 Digital Signal Controllers
Mid	• 56F8000/800 DSC family • HC(S)12 16-bit families
Low	• HCS08 low-voltage, low-power family • HC08 QT/QY family

Analog

- eXtreme Switch
- Motion control
- Power mgmt
- E-Field
- QUICCsupply
- I/O expansion

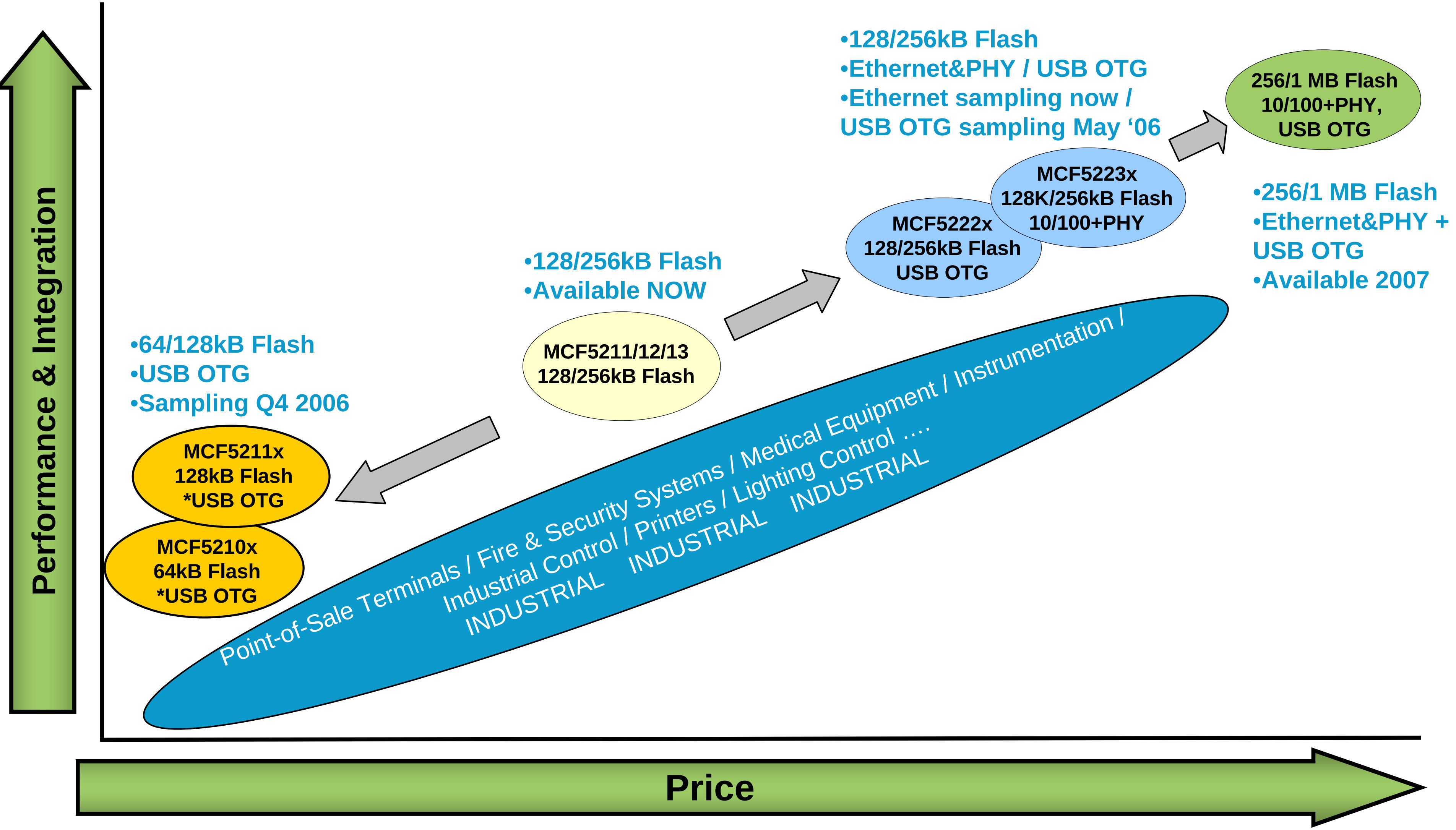
Sensors

- Low-g accelero-meters
- Tire pressure monitoring system (TPMS)

Flash Technology

Software, Tools & Services

68K/*ColdFire* Low Cost MCU Roadmap



Performance & Integration

Price

Point-of-Sale Terminals / Fire & Security Systems / Medical Equipment / Instrumentation / Industrial Control / Printers / Lighting Control …. INDUSTRIAL INDUSTRIAL INDUSTRIAL

MCF5210x
64kB Flash
*USB OTG

MCF5211x
128kB Flash
*USB OTG

- •64/128kB Flash
- •USB OTG
- •Sampling Q4 2006

MCF5211/12/13
128/256kB Flash

- •128/256kB Flash
- •Available NOW

MCF5222x
128/256kB Flash
USB OTG

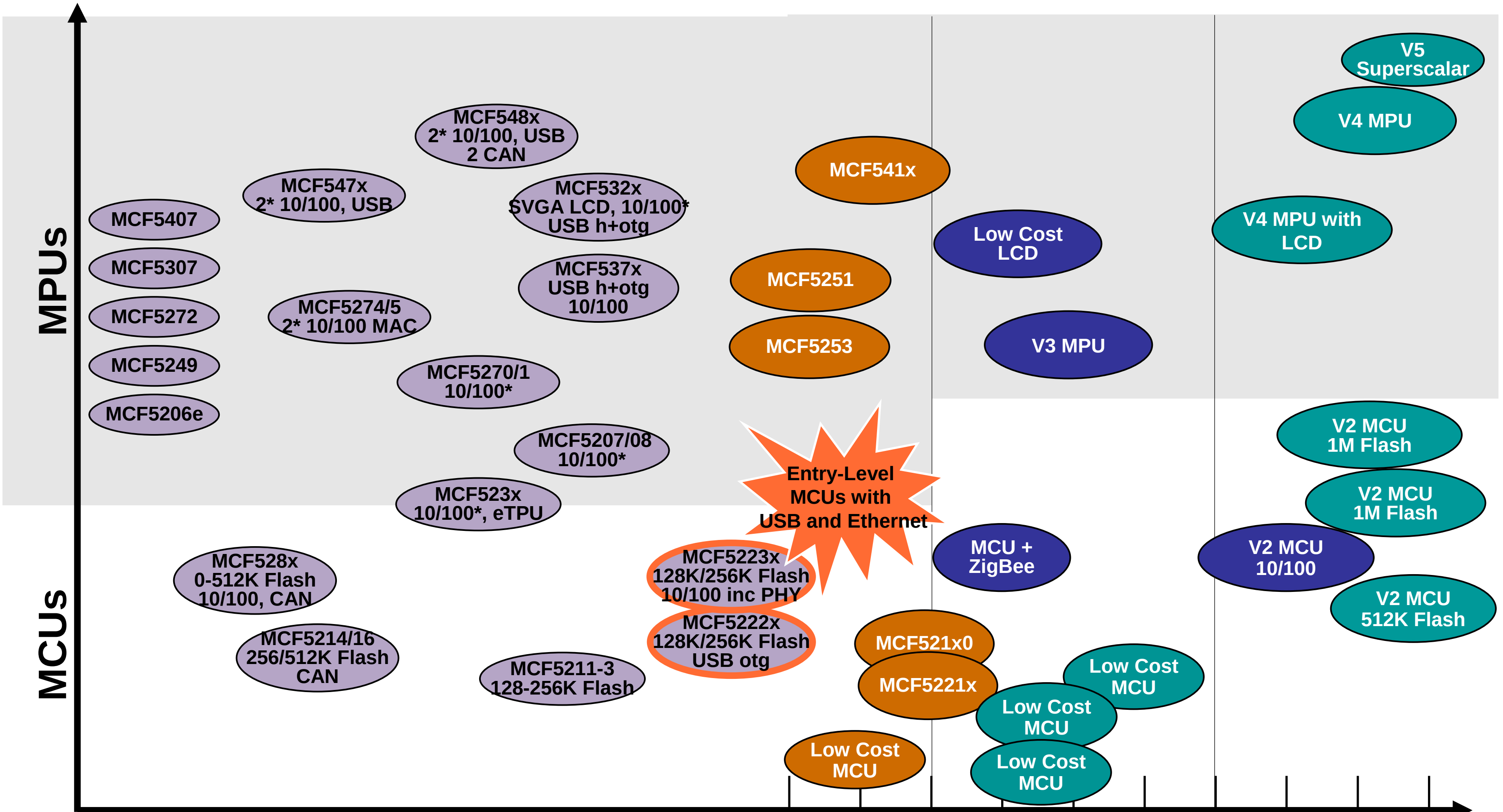
MCF5223x
128K/256kB Flash
10/100+PHY

- •128/256kB Flash
- •Ethernet&PHY / USB OTG
- •Ethernet sampling now / USB OTG sampling May ‘06

256/1 MB Flash
10/100+PHY,
USB OTG

- •256/1 MB Flash
- •Ethernet&PHY + USB OTG
- •Available 2007

68K/*ColdFire* Products Roadmap



MPUs

MCF5407

MCF5307

MCF5272

MCF5249

MCF5206e

MCF547x
2* 10/100, USB

MCF548x
2* 10/100, USB
2 CAN

MCF532x
SVGA LCD, 10/100*
USB h+otg

MCF537x
USB h+otg
10/100

MCF5274/5
2* 10/100 MAC

MCF5270/1
10/100*

MCF5207/08
10/100*

MCF523x
10/100*, eTPU

MCF541x

MCF5251

MCF5253

Low Cost
LCD

V3 MPU

V5
Superscalar

V4 MPU

V4 MPU with
LCD

MCUs

MCF528x
0-512K Flash
10/100, CAN

MCF5214/16
256/512K Flash
CAN

MCF5211-3
128-256K Flash

Entry-Level
MCUs with
USB and Ethernet

MCF5223x
128K/256K Flash
10/100 inc PHY

MCF5222x
128K/256K Flash
USB otg

MCF521x0

MCF5221x

Low Cost
MCU

MCU +
ZigBee

Low Cost
MCU

Low Cost
MCU

Low Cost
MCU

V2 MCU
1M Flash

V2 MCU
1M Flash

V2 MCU
10/100

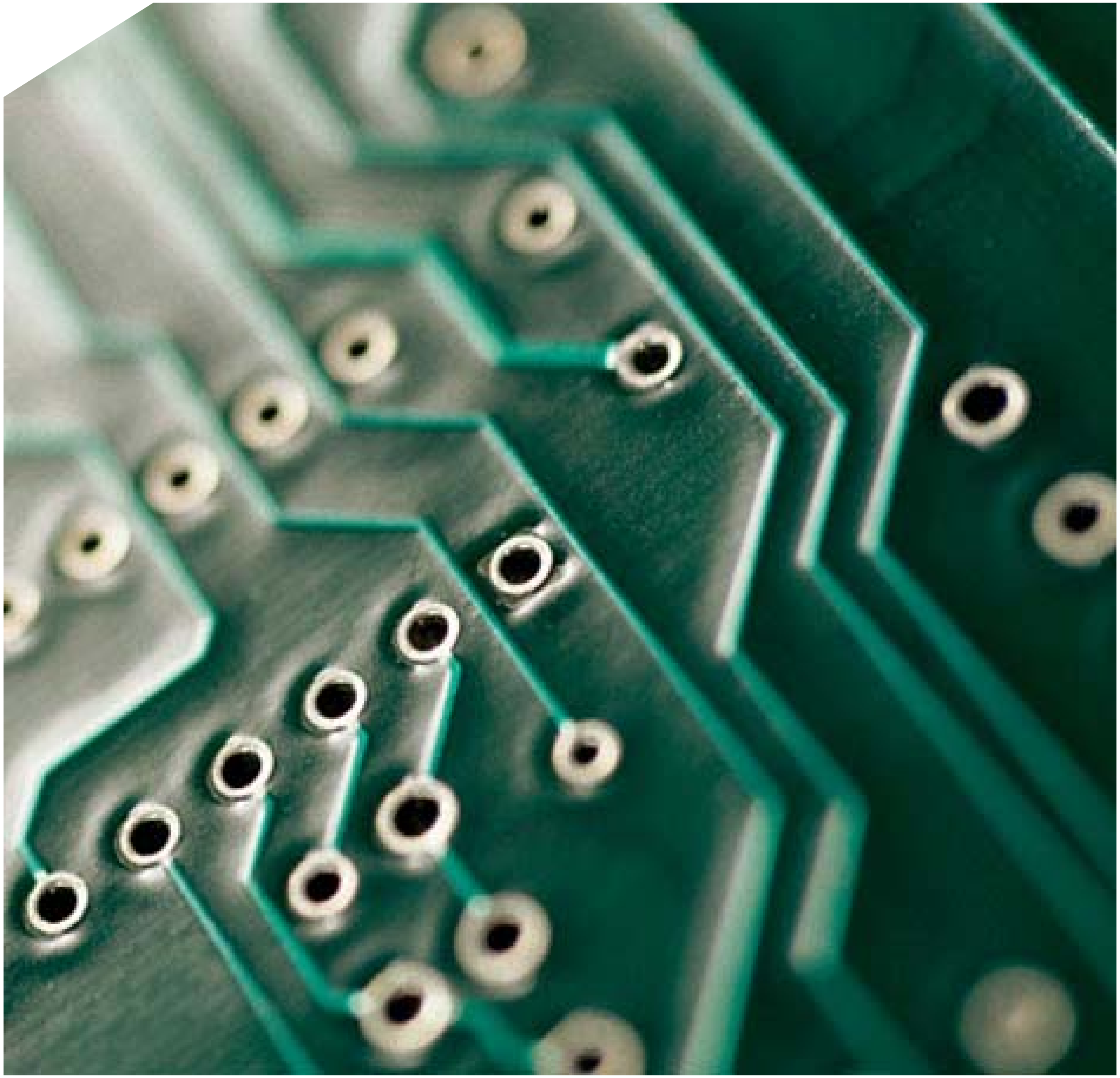
V2 MCU
512K Flash

2006 2007 2008

* = Optional

Production - *Available NOW*
Execution - *Specification Frozen, High Confidence Schedule*
Planning - *Specification Subject to Change, Tentative Schedule*
Proposal - *Project Subject to Change, Open to Market Feedback*

freescale™
semiconductor



MCF5222x USB enabled solutions

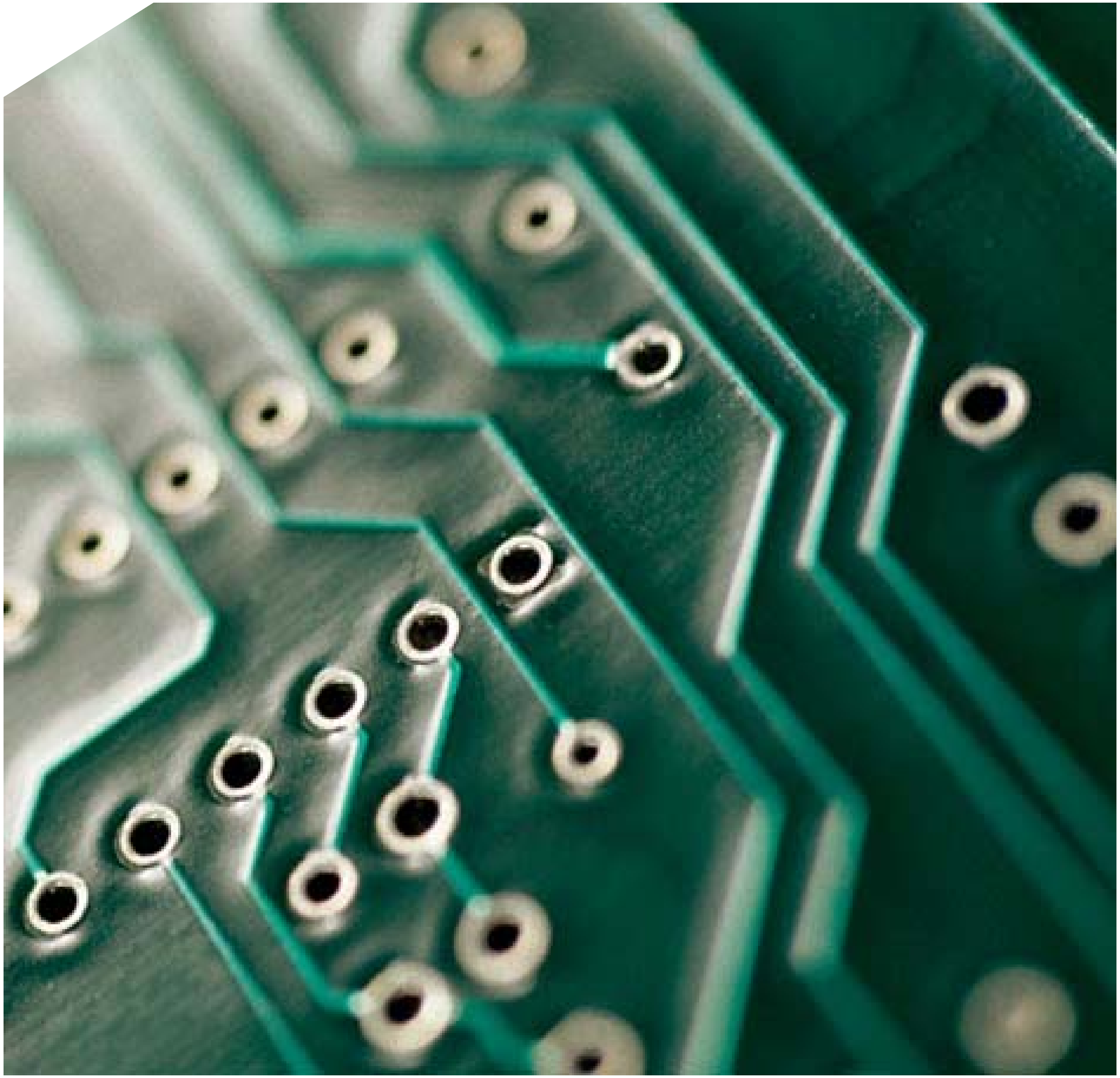
Software Collateral for USB support

- The MCF5222x will be rolled out with a robust software offering for USB support.
- The firmware will be provided license free with source from Freescale.

- Example Firmware (all source included):

  - A virtual COM port demo – The ColdFire will act as a USB to serial port dongle. API hooks will be available on the ColdFire side to provide putchar and getchar functionality for a users application.

  - HID (host and device) – The ColdFire will emmulate a keyboard or mouse, connected to the PC. One the host side, a user will be able to connect a keyboard or mouse to the ColdFire. API's will be provided on the ColdFire side to allow a users application direct access to the PC via the keyboard HID interface, same applies for the mouse.



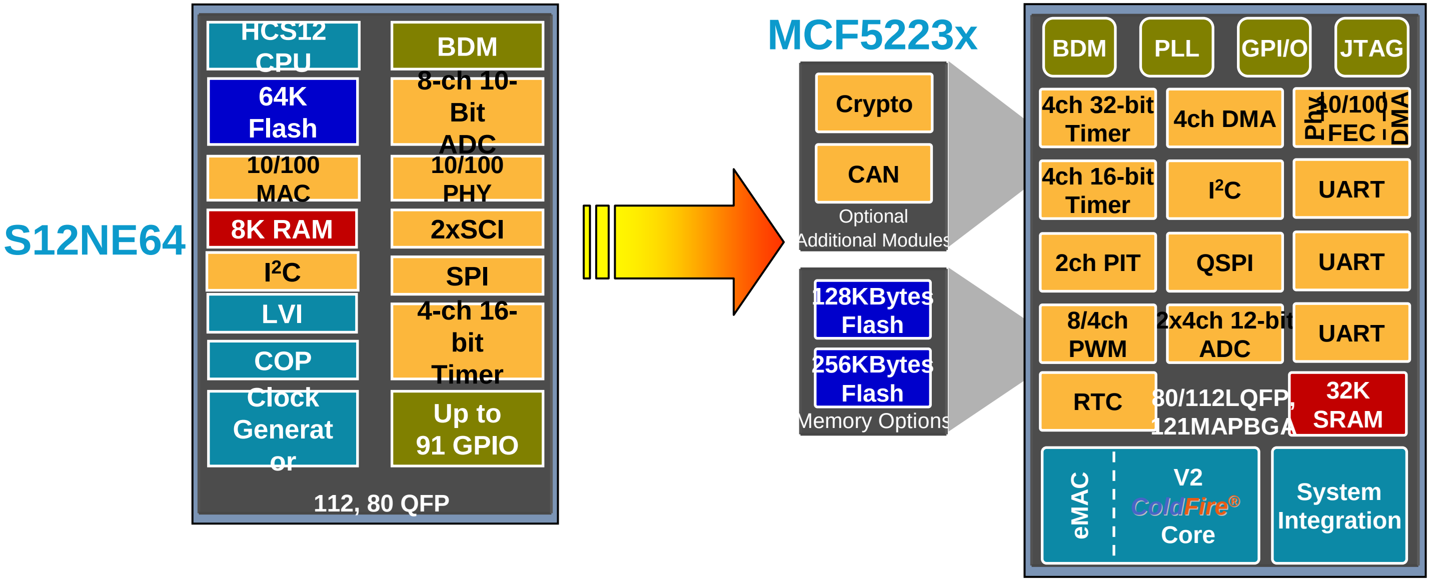
MCF5223x Ethernet enabled solutions

Ethernet Enabled MCUs

Based on customer requirements for higher performance, more RAM and a hardware encryption option, future Ethernet enabled MCUs with >64KB will be based on the ColdFire architecture.
Freescale will **continue to support the S12NE64** for production and new designs

For customers requiring >64kB, migration to the ColdFire architecture offers a number of benefits including -

- Higher performance at the same price points (going from 25MHz to 60MHz)
- Up to 256kB of Flash, up to 32kB of RAM
- Enhanced peripherals: SCI → UART, 16 bit Timer→ 32 bit Timer, 4 ch DMA, Optional Hardware Encryption module, CAN
- Large existing portfolio of Ethernet enabled ColdFire devices with several new products in development



MCF5223x *ColdFire* Family

Targeted at Industrial Control Applications

- Environmental Monitoring
- Remote Data Collection
- Medical Pumps and Monitors
- Power-over-Ethernet
- Security/Access Panels
- Lighting Control Nodes
- Vending Machines

Key Features

- 57 MIPS V2 Core with Enhanced Multiply and Accumulate for DSP-like functionality!
- Integrated Connectivity including:
  - **10/100 Ethernet Controller**
  - **10/100 Ethernet Physical Layer**
  - **CAN 2.0B Controller**
  - **Cryptographic Acceleration Unit**
- Additional control features include:
  - Up to 73 General Purpose I/O
  - 4ch. 32-bit timers with DMA support
- Starting from $7.99 suggested resale price

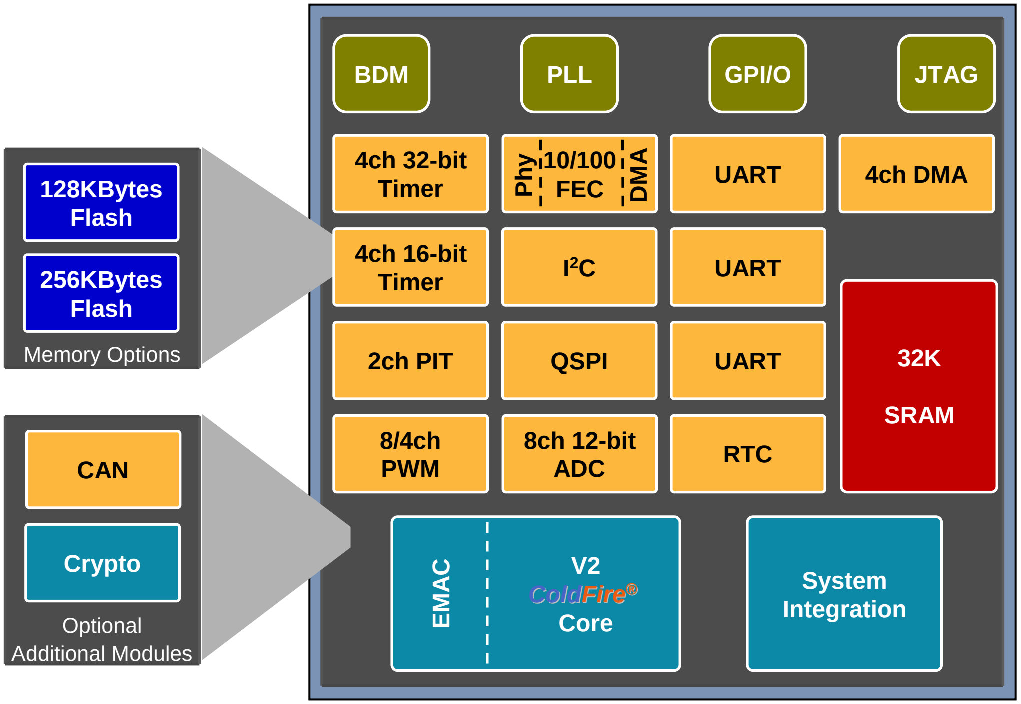
68K/*ColdFire*: MCF5223x

68K/*ColdFire* V2 Core

- Up to 56 Dhrystone 2.1 MIPS @ 60 MHz
- EMAC Module and HW Divide
- Optional Cryptography Accelerator with Random Number Generator

No external bus

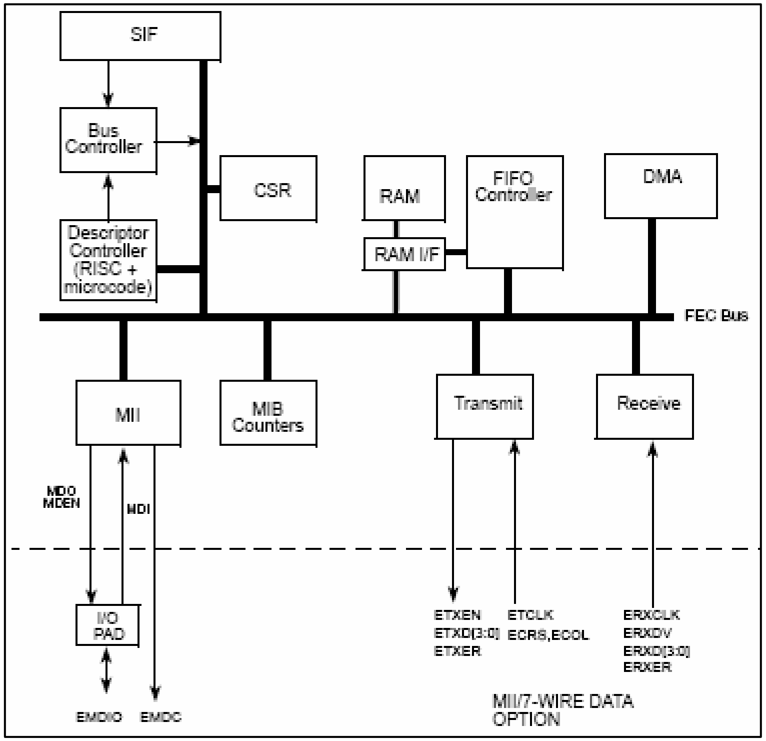
- 32K bytes SRAM
- Up to 256K bytes Flash
  - 100K W/E cycles, 10 years data retention
- 10/100 Ethernet MAC with PHY
- Optional CAN 2.0B Controller
- 3 UARTs
- Queued Serial Peripheral Interface (QSPI)
- I²C bus interface
- 4 ch. 32-bit timers with DMA support
- 4 ch. 16-Bit Capture/Compare/PWM timers
- 2 ch. Periodic Interrupt Timer
- 8/4 ch. 8/16-bit PWM timer
- 8 ch. 12-bit A-to-D converter with Simultaneous Sampling
- Real Time Clock
- 4 ch. DMA controller
- Up to 63 General-Purpose I/O
- System Integration (PLL, SW Watchdog)
- Single 3.3V supply
- Temperature Range: -40°C to +85°C
- Available Speeds: 60MHz
- From $7.99 @ 10k qty



Part Number	Flash K bytes	CAN	Crypto	Packages	Target 10K Resale
MCF52230	128	No	No	121 MAPBGA 80 LQFP	$8.22 $7.99
MCF52231	128	Yes	No	112 LQFP 80 LQFP	$8.22 $8.79
MCF52233	256	No	No	112 LQFP 80 LQFP	$8.92 $8.69
MCF52234	256	Yes	No	121 MAPBGA, 112 LQFP 80 LQFP	$9.62 $9.39
MCF52235	256	Yes	Yes	121 MAPBGA, 112 LQFP	$11.32

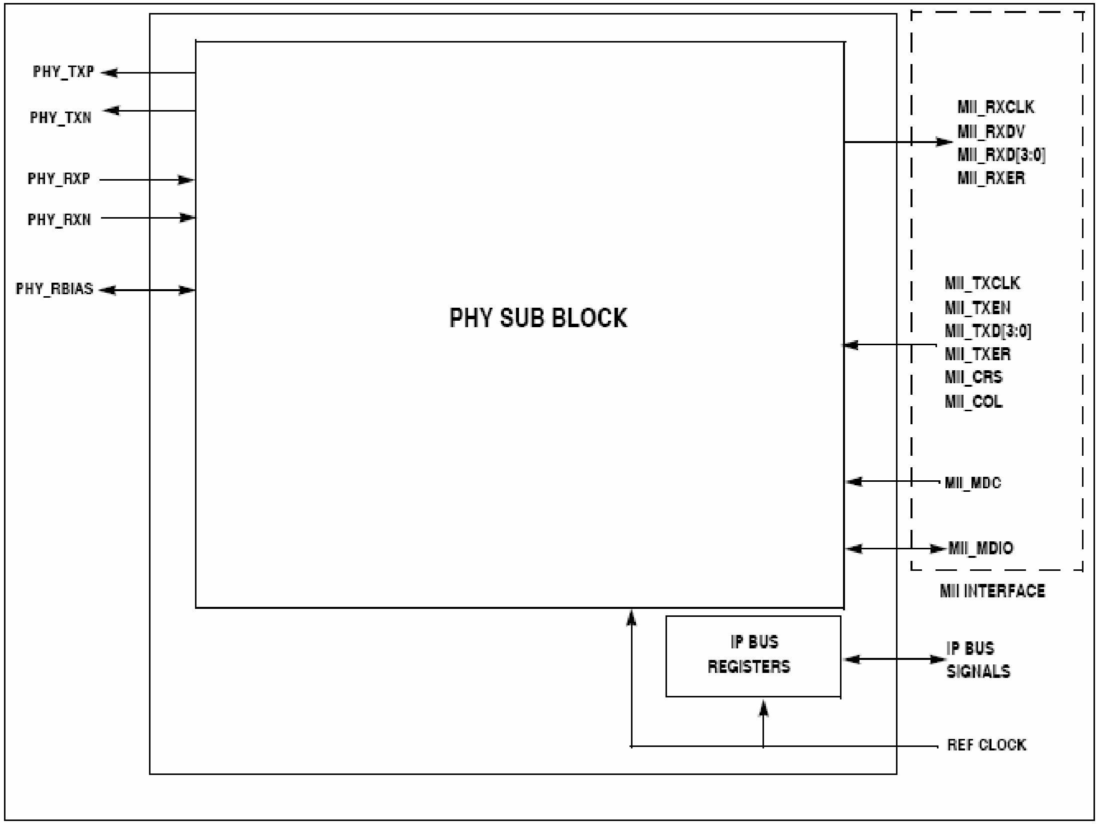
MCF5223x – Ethernet Media Access Controller (MAC)

- The Ethernet MAC supports 10/100 Mbps Ethernet/IEEE 802.3 networks
- IEEE 802.3 full duplex flow control
- Support for full-duplex operation (40Mbps throughput) with a minimum system clock rate of 50MHz
- Support for half-duplex operation (20Mbps throughput) with a minimum system clock rate of 25MHz



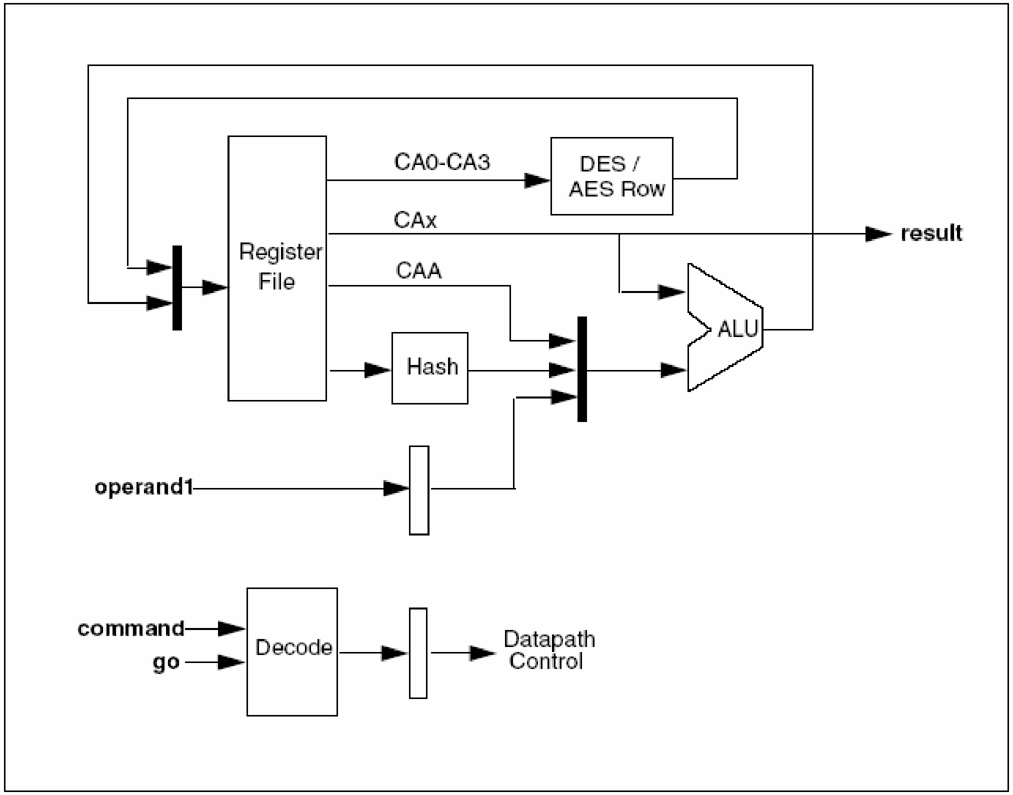
MCF5223x - ePHY

- The ePHY (embedded PHYsical layer interface) is IEEE 802.3 compliant
- Supports both the media-independent interface (MII) and the MII management interface
- Full-/half-duplex support in all modes
- Requires a 25-MHz crystal for its basic operation
- Supports Loopback modes



MCF5223x - Cryptographic Acceleration Unit (CAU)

- Uses standard ***ColdFire®*** coprocessor interface and instructions
- Simple, flexible programming model
- Supports DES, 3DES, AES, MD5 and SHA-1.
- Architecture allows for future enhancements
- Supports all ***ColdFire®*** cores



FlexCAN – Controller Area Network

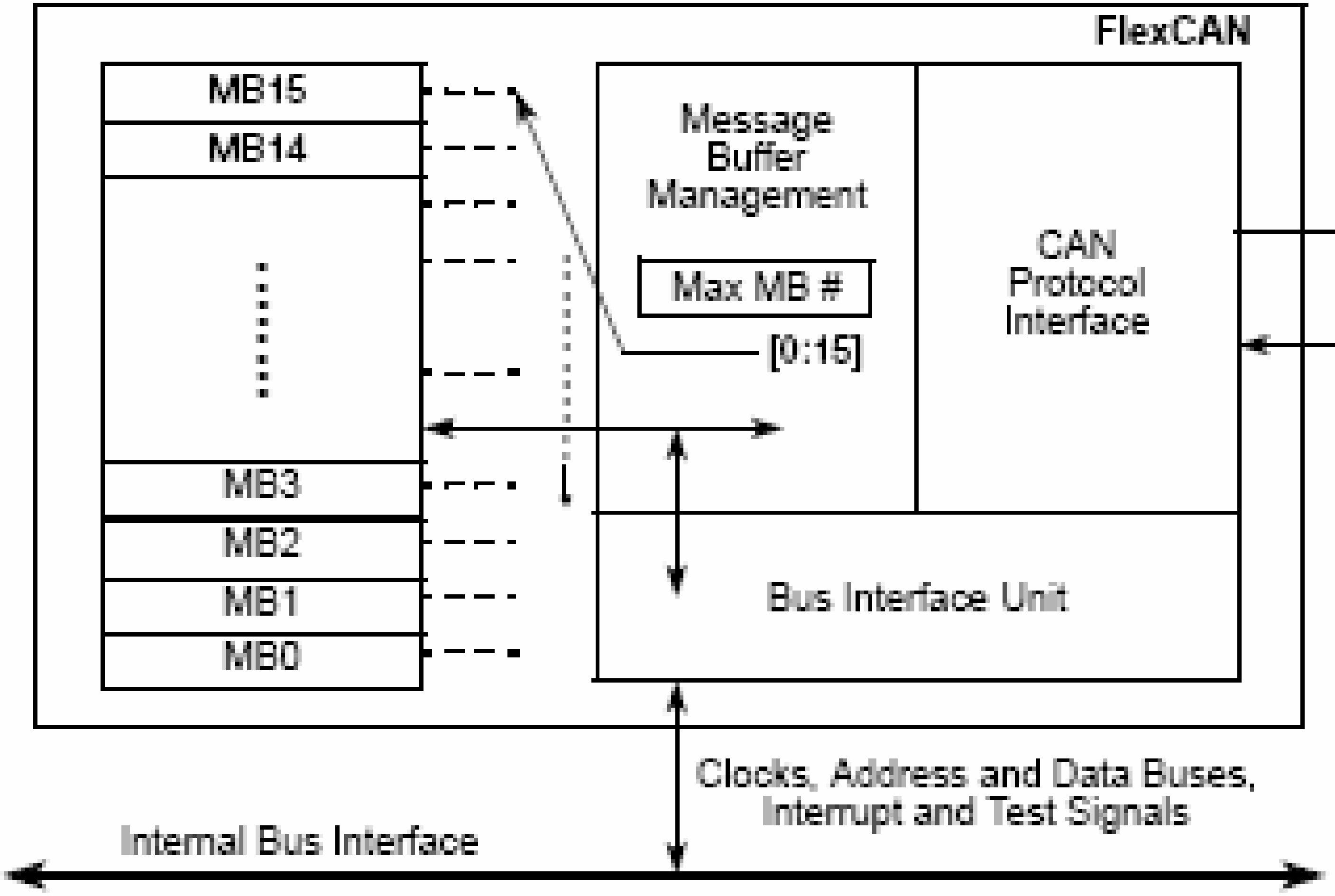
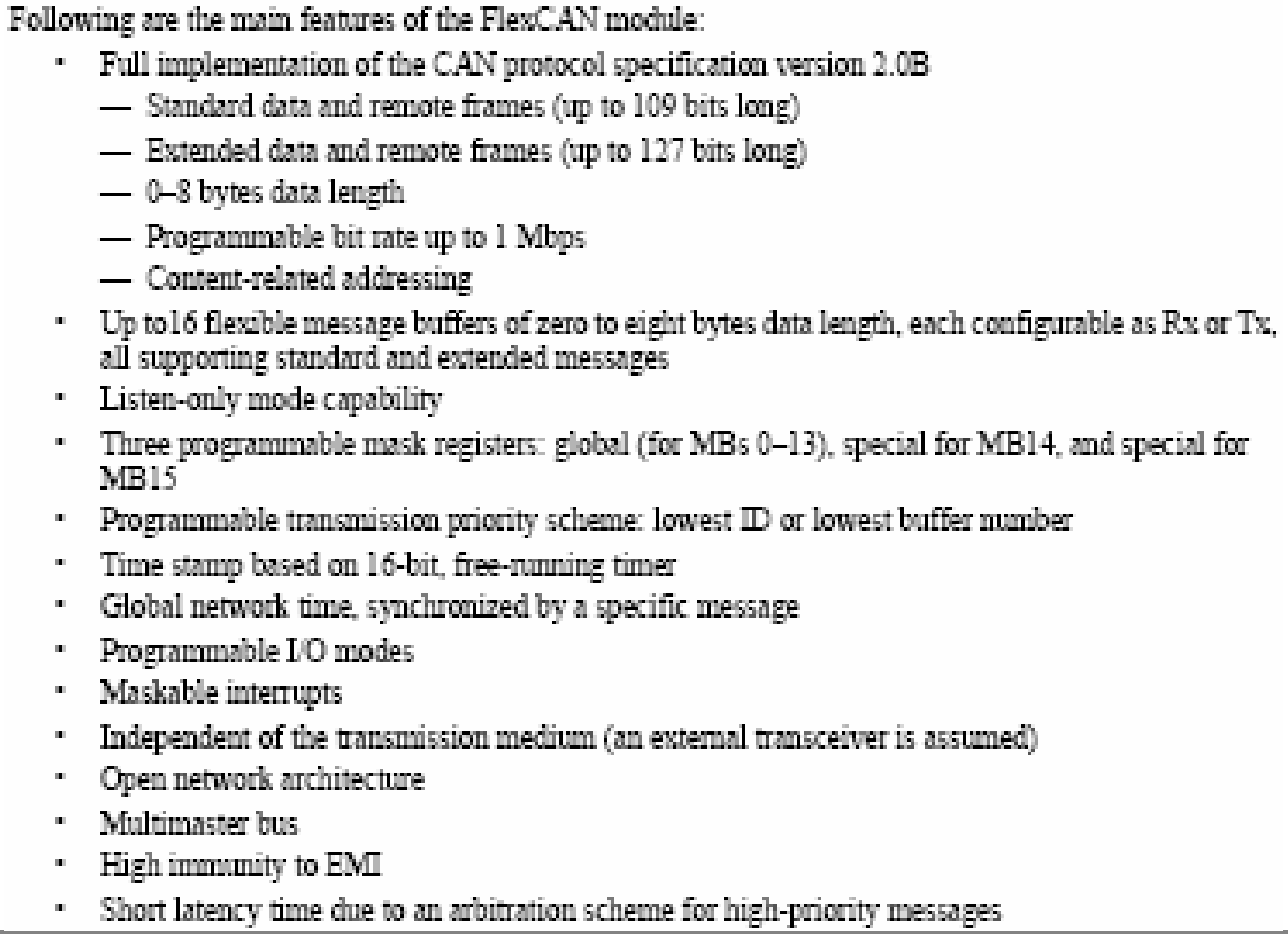


Figure 30-1. FlexCAN Block Diagram and Pinout

FlexCAN – Controller Area Network



Following are the main features of the FlexCAN module:

- Full implementation of the CAN protocol specification version 2.0B
  - — Standard data and remote frames (up to 109 bits long)
  - — Extended data and remote frames (up to 127 bits long)
  - — 0–8 bytes data length
  - — Programmable bit rate up to 1 Mbps
  - — Content-related addressing
- Up to16 flexible message buffers of zero to eight bytes data length, each configurable as Rx or Tx, all supporting standard and extended messages
- Listen-only mode capability
- Three programmable mask registers: global (for MBs 0–13), special for MB14, and special for MB15
- Programmable transmission priority scheme: lowest ID or lowest buffer number
- Time stamp based on 16-bit, free-running timer
- Global network time, synchronized by a specific message
- Programmable I/O modes
- Maskable interrupts
- Independent of the transmission medium (an external transceiver is assumed)
- Open network architecture
- Multimaster bus
- High immunity to EMI
- Short latency time due to an arbitration scheme for high-priority messages

MCF5223x Family Device Matrix

Part Number	Flash/SRAM	Key Features	Package	Speed MHz	*10k Sugg. Resale Pricing
MCF52230	128KB/32KB	FEC, EPHY, 3 UARTs, I2C, QSPI, A/D, 16-bit, 32-bit, PWM timers, DMA	80/112 LQFP	60	From $7.99
MCF52231	128KB/32KB	FEC, EPHY, 3 UARTs, I2C, QSPI, A/D, 16-bit, 32-bit, PWM timers, DMA, CAN	80/112 LQFP	60	From $8.79
MCF52233	256KB/32KB	FEC, EPHY, 3 UARTs, I2C, QSPI, A/D, 16-bit, 32-bit, PWM timers, DMA	80/112 LQFP	60	From $8.69
MCF52234	256KB/32KB	FEC, EPHY, 3 UARTs, I2C, QSPI, A/D, 16-bit, 32-bit, PWM timers, DMA, CAN	112 LQFP 121 MAPBGA,	60	From $9.42
MCF52235	256KB/32KB	FEC, EPHY, 3 UARTs, I2C, QSPI, A/D, 16-bit, 32-bit, PWM timers, DMA, CAN, CAU	112 LQFP 121 MAPBGA,	60	From $11.32

*Freescale Suggested 10K Resale Pricing

Available April 2006

M52235EVB Development Kit	$299

Available May 2006

M52233DEMO Low Cost Board	$99

M52235EVB Evaluation Board

M52235EVB Evaluation Board and Development System

- Evaluation board with fully functional Power over Ethernet circuitry. Supports plug-in Zigbee daughter card
- Kit to include CD ROM, Power Supply, P&E BDM Cable, and Ethernet Crossover Cable
- Target Suggested Resale Price: $299

M52235EVB Software Support

- Free ColdFire_TCP/IP_Lite stack
- Free CodeWarrior® SPECIAL EDITION Included in Each Development Kit
- ColdFire Init – Graphical Initialization Tool
- Professional Tools and Systems demos scheduled from:
  - CodeWarrior® IDE
  - Accelerated Technology compiler debugger
  - MQX™ Embedded Precise RTOS
  - Green Hills Software IDE RTOS
  - Wind River – Wind River Compiler™ and Hardware Assisted Debugger
  - TCP/IP Stacks: ColdFire_TCP/IP_Lite Stack by Interniche ($0)

http://www.iniche.com/
http://www.cmx.com/
http://www.treck.com/
http://www.ghs.com

M52233DEMO Low cost demo board

M52233DEMO Low Cost Board

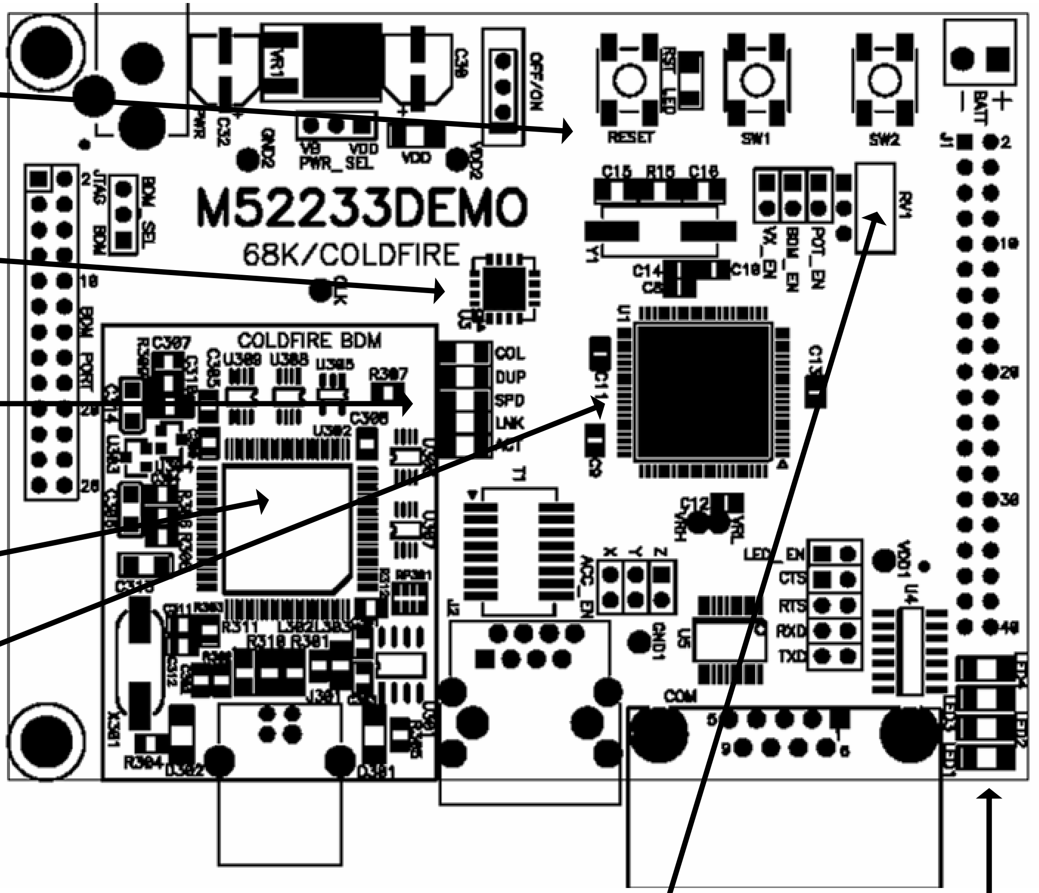
- Evaluation board with Plug-in Zigbee daughter card
- Kit to include CD ROM, Power Supply, and Ethernet Crossover Cable
- Target Suggested Resale Price: $99
- Available: May 2006

M52233DEMO Software Support

- Free ColdFire_TCP/IP_Lite stack
- Free CodeWarrior® SPECIAL EDITION Included in Each Development Kit
- ColdFire Init – Graphical Initialization Tool
- Professional Tools and Systems demos scheduled from:
  - CodeWarrior® IDE
  - Accelerated Technology compiler debugger
  - MQX™ Embedded Precise RTOS
  - Green Hills Software IDE RTOS
  - Wind River – Wind River Compiler™ and Hardware Assisted Debugger
  - TCP/IP Stacks: ColdFire_TCP/IP_Lite Stack by Interniche ($0)

http://www.iniche.com/
http://www.cmx.com/
http://www.ghs.com
http://www.treck.com

Demo Board - Major Components



RESET and User Switches

Tri- Axis Accelerometer

ePHY status LEDs

Integrated BDM using 9S12UF32

80 pin MCF52233

Potentiometer

User LEDs

EVB and DEMO Comparison

- MCF52235
  - 32K RAM 256K Flash, Ethernet with PHY, CAN, Crypto
  - 112 LQFP pin
- Light Sensor
- PoE capabilities
- 3 UARTs
- Supports plug-in Zigbee daughter card

- MCF52233
  - 32K RAM 256K Flash, Ethernet with PHY
  - 80 LQFP pin
- Accelerometer (3 axis g sensor)
- 1 UART
- Supports plug-in Zigbee daughter card

Segmented *ColdFire*® Development Tools v6.3

Special Edition “Free” to customer

- Assembly and C language support
- P&E parallel/USB support (CodeWarrior® -USBTAP™ when available)
- CF Flasher Included
- Node locked only
- Fully Optimizing compiler included
- Support for entire range of Freescale ColdFire silicon
- Code size restricted to 128K
- 1 yr tech support included

Standard Edition ($2,495)

- Assembly and C language support
- Full Floating Point libraries (download extended libraries) and support for FPU hardware instructions
- V2 and V4e instruction set simulator
- P&E parallel and USB (CodeWarrior® - USBTAP™ when available)
- Integrated CodeWarrior® Flash programmer and Freescale CF Flasher
- Support for entire range of Freescale ColdFire silicon
- 1 yr tech support included

Professional Edition ($5,995)

Everything in the Standard Edition plus these advanced professional features:

- C++language support
- Abatron BDI and CodeWarrior® EthernetTAP™ run control solutions (when available)
- CodeWarrior® extensions enabled (eg version control)
- RTOS aware debugger (for use with 3rd party RTOS like ARC, ThreadX, Quadros and more…)
- 1 yr tech support included

*ColdFire*_TCP/IP_Lite Stack

Available from Freescale:
*InterNiche Technologies and Freescale have collaborated to provide an OEM version of InterNiche's NicheLite™, ColdFire*_TCP/IP_Lite

Features

- Address Resolution Protocol (ARP)
- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
- Dynamic Host Configuration Protocol
- (DHCP) Client
- Bootstrap Protocol (BOOTP)
- Trivial File Transfer Protocol (TFTP)

Freescale Provided additional free software:

- Web Server with Flash File System
- Mail Server

MCF5223x Roll-out Schedule

Deliverable	Availability
Samples	80QFP: now (PCF52233CAF60) 112 LQFP: now (PCF52235CAL60) 121 MAPBGA: July ’06 (PCF52235CVM60)
EVBs	Now (M52235EVB)
DEMO Boards	August 06 (M52233DEMO)
Market Launch	4 April 2006
CodeWarrior	Beta Version available in EVB and DEMO boards Full release in July, 2006
TCP/IP Stacks	*ColdFire*_TCP/IP_Lite Stack April ‘06 Interniche: May ’06 CMX: now Treck: Now
16 to 32-bit migration Application Note	August ’06
Product Qualification	August ‘06

68K/*ColdFire* : Web Resources

68K/*ColdFire* Home Page

http://www.Freescale.com/ColdFire

- Latest documentation
- Application notes
- Reference Designs
- Evaluation board schematics
- Links of interest
- Sample code

68K/*ColdFire* Discussion Groups

http://forums.freescale.com

Expert advice from the developer community moderated by Freescale 68K/*ColdFire* application engineers

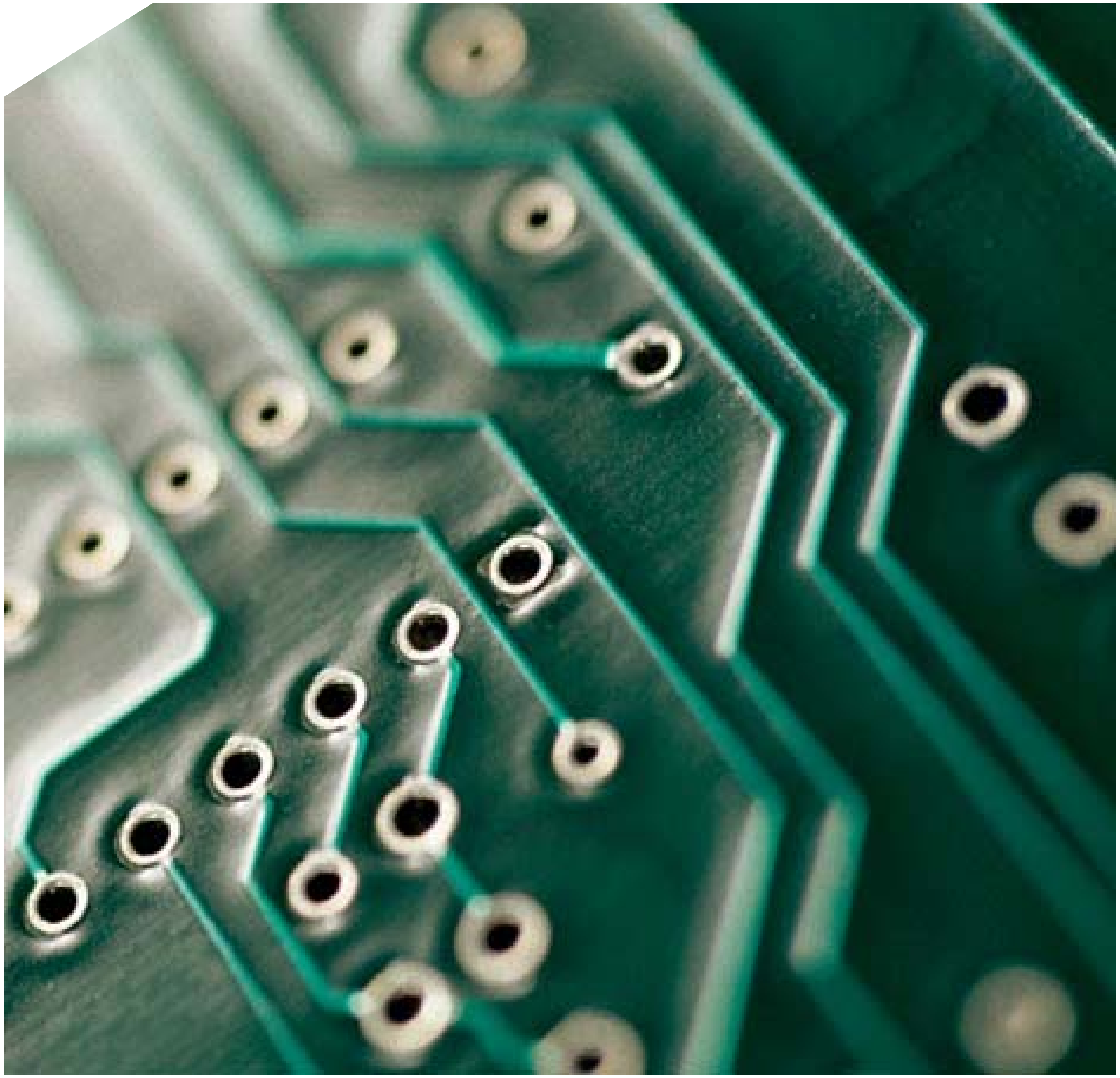
http://www.wildrice.com/ColdFire

Historical 68K/*ColdFire* discussion group not affiliated with Freescale

MCF522xx Tools and Software

Tools	Vendors
Eval Boards Ref Designs	freescale™ semiconductor InterNiche technologies, inc. AXIOM manufacturing NetBurner Networking in 1 Day!
RTOS	Green Hills® SOFTWARE, INC. Accelerated Technology® Embedded Systems Division of Mentor Graphics Quadros™ Systems Inc. ARC
Compiler Simulator Debugger	freescale™ semiconductor Green Hills® SOFTWARE, INC. WIND RIVER GNU Accelerated Technology® Embedded Systems Division of Mentor Graphics
Stacks Drivers Translators	freescale™ semiconductor InterNiche technologies, inc. NetBurner Networking in 1 Day! MICRO APL CMX SYSTEMS Quadros™ Systems Inc. ARC Green Hills® SOFTWARE, INC. Accelerated Technology® Embedded Systems Division of Mentor Graphics

freescale™ semiconductor



M52233DEMO Development Kit Set Up

freescale™ semiconductor

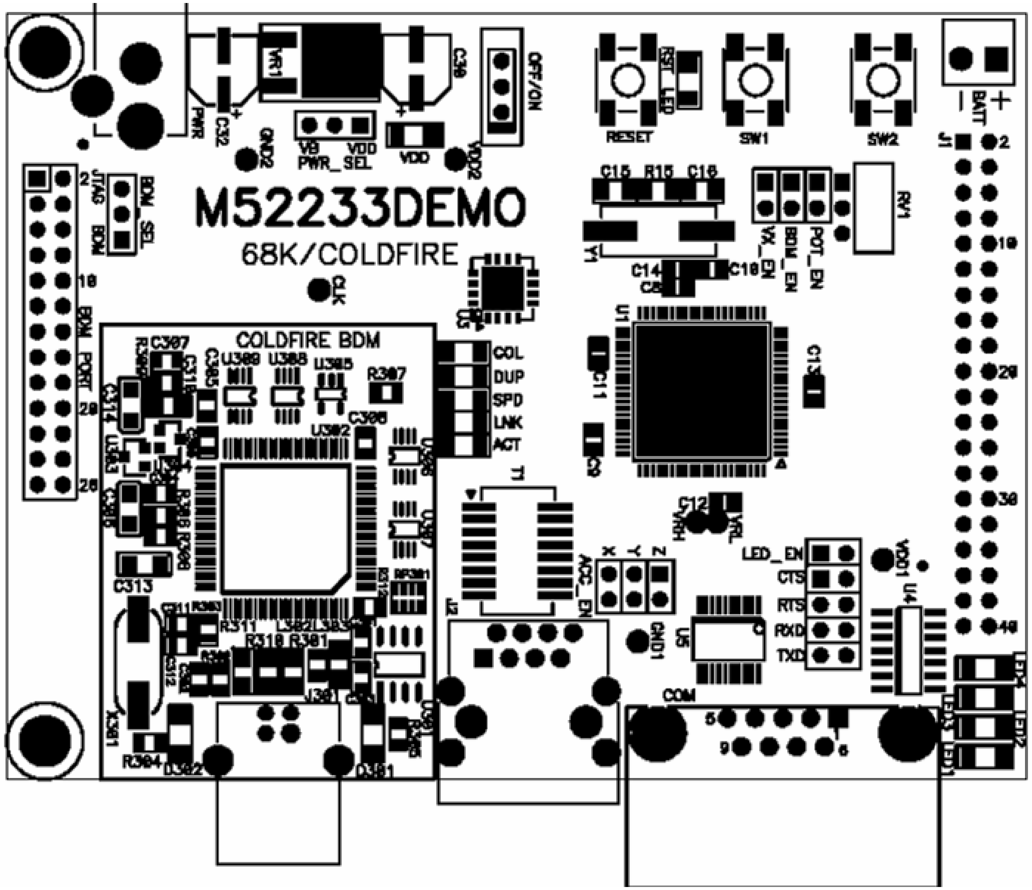
Let’s Start

- Open Kit – plug Ethernet and USB cables
- Turn on Power switch
- Should have power and USB LEDs

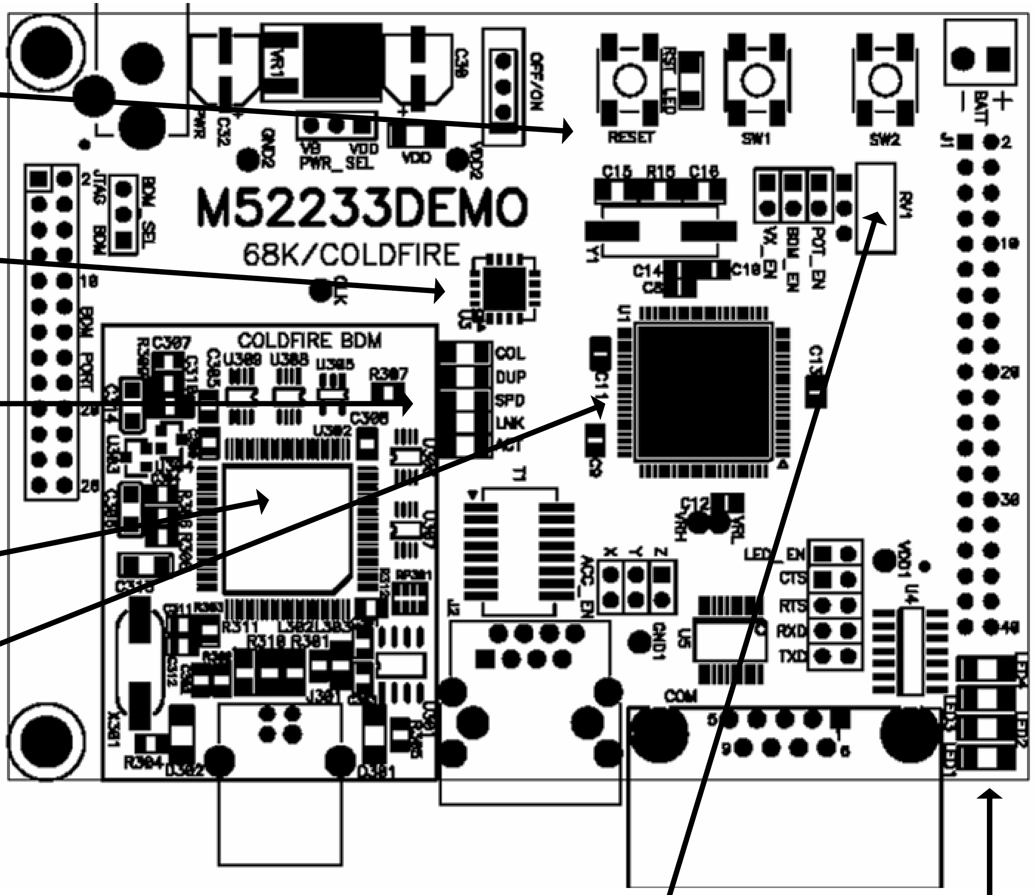
Demo Board

Contents - DB9 Serial Cable, USB cable, Ethernet Cable, Support CD, and CodeWarrior® Development Studio CD

USB powered! No need for external power supply.



Demo Board - Major Components



RESET and User Switches

Tri- Axis Accelerometer

ePHY status LEDs

Integrated BDM using 9S12UF32

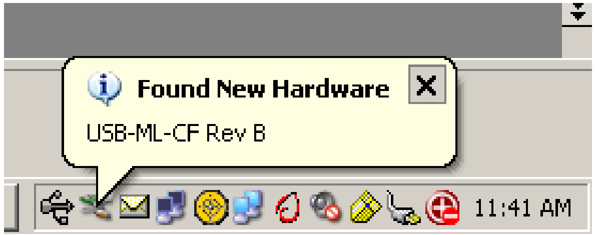
80 pin MCF52233

Potentiometer

User LEDs

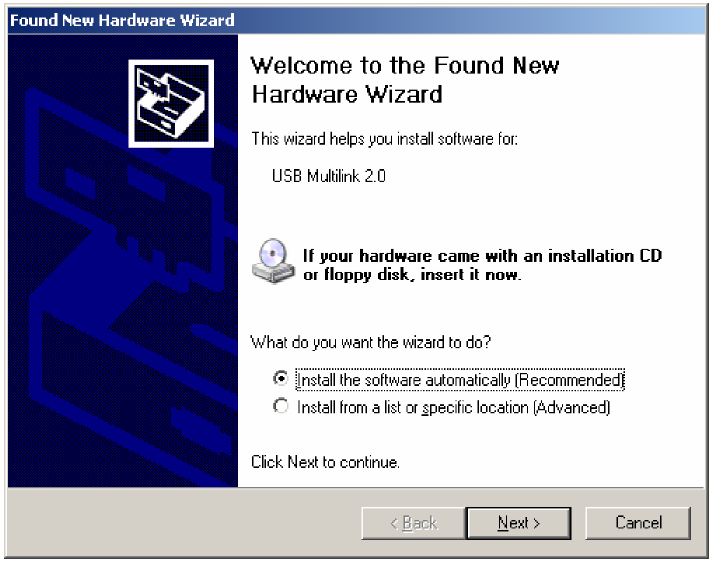
Plug in the USB Cable

Plug in the Supplied USB cable and Windows will detect and install driver



Found New Hardware

USB-ML-CF Rev B



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

USB Multilink 2.0

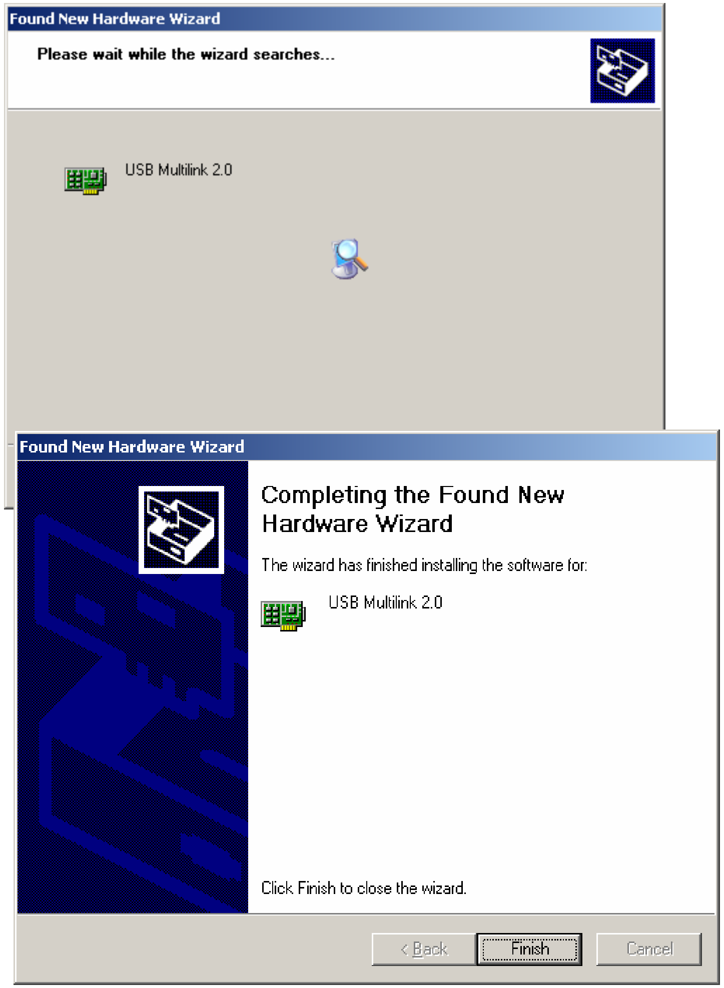
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

● Install the software automatically (Recommended)

○ Install from a list or specific location (Advanced)

Click Next to continue.



Found New Hardware Wizard

Please wait while the wizard searches...

USB Multilink 2.0

Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

USB Multilink 2.0

Click Finish to close the wizard.

Serial Port

Should see something close to this on serial port

Once the USB is configured, cycle the USB cable and turn DEMO on.

You should see some like on the left on your terminal program.

(115200, 8, N)

- ePHY enabled

- ePHY delay ready

- Running Open Source Network Stack
- Built on Apr 19 2006 15:58:09
- Software Ver: 01.00.312

- Main Entered

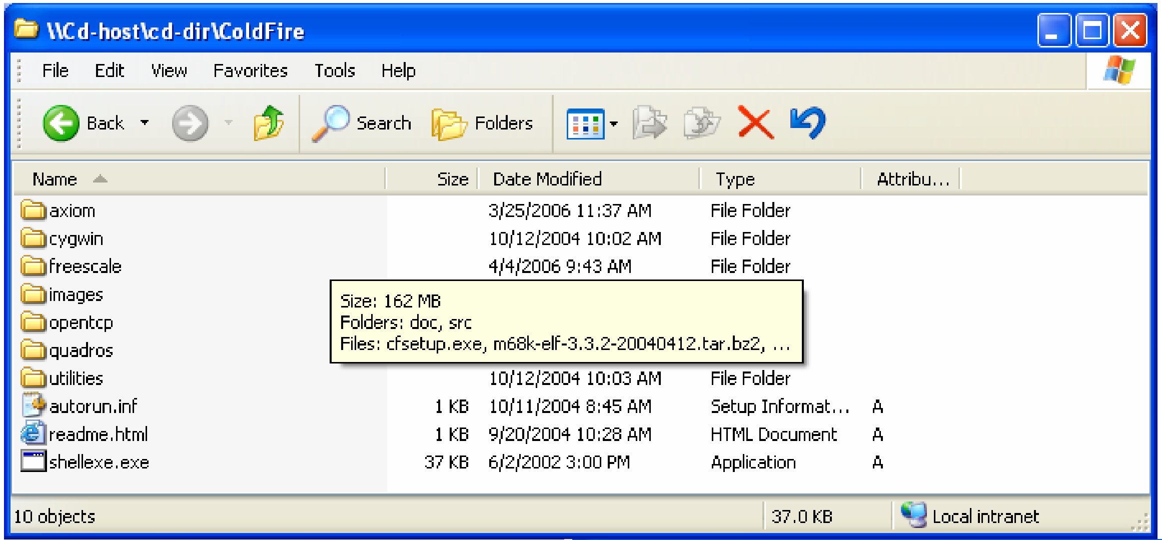
- External Reset
- MCF5223 Rev. 1 Core Initialization Complete!

- Chip ID: 4C
- Single-chip Mode, Default Drive

- DHCP Failed - Reverting to local IP

- MAC Address: 00:0B:06:E3:40:7B
- IP Address: 192.168.001.004
- Gateway: 192.168.001.001
- Subnet Mask: 255.255.255.000

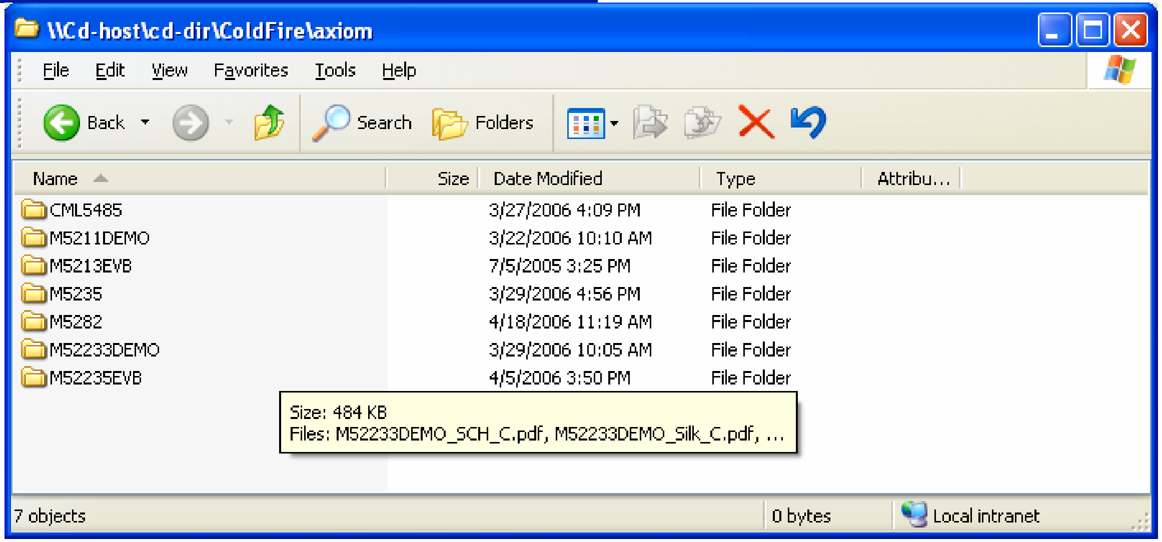
Contents of Axiom CD


\\Cd-host\cd-dir\ColdFire
File Edit View Favorites Tools Help
Back
Search
Folders

Name	Size	Date Modified	Type	Attribu...
axiom		3/25/2006 11:37 AM	File Folder	
cygwin		10/12/2004 10:02 AM	File Folder	
freescale		4/4/2006 9:43 AM	File Folder	
images				
opentcp				
quadros				
utilities		10/12/2004 10:03 AM	File Folder	
autorun.inf	1 KB	10/11/2004 8:45 AM	Setup Informat...	A
readme.html	1 KB	9/20/2004 10:28 AM	HTML Document	A
shellexe.exe	37 KB	6/2/2002 3:00 PM	Application	A

Size: 162 MB
Folders: doc, src
Files: cfsetup.exe, m68k-elf-3.3.2-20040412.tar.bz2, ...

10 objects | 37.0 KB | Local intranet


\\Cd-host\cd-dir\ColdFire\axiom
File Edit View Favorites Tools Help
Back
Search
Folders

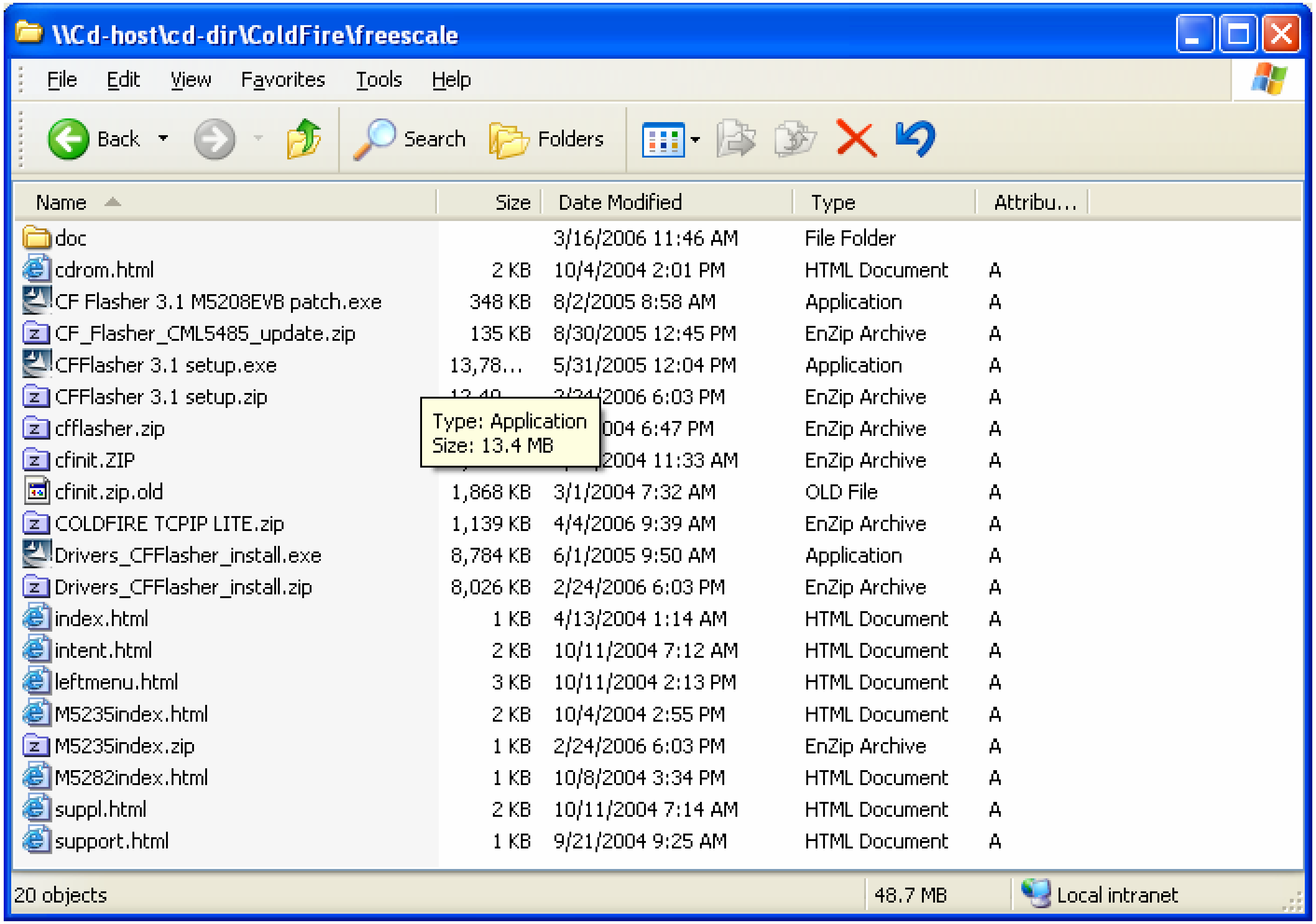
Name	Size	Date Modified	Type	Attribu...
CML5485		3/27/2006 4:09 PM	File Folder	
M5211DEMO		3/22/2006 10:10 AM	File Folder	
M5213EVB		7/5/2005 3:25 PM	File Folder	
M5235		3/29/2006 4:56 PM	File Folder	
M5282		4/18/2006 11:19 AM	File Folder	
M52233DEMO		3/29/2006 10:05 AM	File Folder	
M52235EVB		4/5/2006 3:50 PM	File Folder	

Size: 484 KB
Files: M52233DEMO_SCH_C.pdf, M52233DEMO_Silk_C.pdf, ...

7 objects | 0 bytes | Local intranet

freescale™ semiconductor

Contents of Axiom CD - cont.



\\Cd-host\cd-dir\ColdFire\freescale

Name	Size	Date Modified	Type	Attribu...
doc		3/16/2006 11:46 AM	File Folder	
cdrom.html	2 KB	10/4/2004 2:01 PM	HTML Document	A
CF Flasher 3.1 M5208EVB patch.exe	348 KB	8/2/2005 8:58 AM	Application	A
CF_Flasher_CML5485_update.zip	135 KB	8/30/2005 12:45 PM	EnZip Archive	A
CFFlasher 3.1 setup.exe	13,78...	5/31/2005 12:04 PM	Application	A
CFFlasher 3.1 setup.zip	13,40...	2/24/2006 6:03 PM	EnZip Archive	A
cfflasher.zip	[illegible]	[illegible]004 6:47 PM	EnZip Archive	A
cfinit.ZIP	[illegible]	[illegible]2004 11:33 AM	EnZip Archive	A
cfinit.zip.old	1,868 KB	3/1/2004 7:32 AM	OLD File	A
COLDFIRE TCPIP LITE.zip	1,139 KB	4/4/2006 9:39 AM	EnZip Archive	A
Drivers_CFFlasher_install.exe	8,784 KB	6/1/2005 9:50 AM	Application	A
Drivers_CFFlasher_install.zip	8,026 KB	2/24/2006 6:03 PM	EnZip Archive	A
index.html	1 KB	4/13/2004 1:14 AM	HTML Document	A
intent.html	2 KB	10/11/2004 7:12 AM	HTML Document	A
leftmenu.html	3 KB	10/11/2004 2:13 PM	HTML Document	A
M5235index.html	2 KB	10/4/2004 2:55 PM	HTML Document	A
M5235index.zip	1 KB	2/24/2006 6:03 PM	EnZip Archive	A
M5282index.html	1 KB	10/8/2004 3:34 PM	HTML Document	A
suppl.html	2 KB	10/11/2004 7:14 AM	HTML Document	A
support.html	1 KB	9/21/2004 9:25 AM	HTML Document	A

Type: Application
Size: 13.4 MB

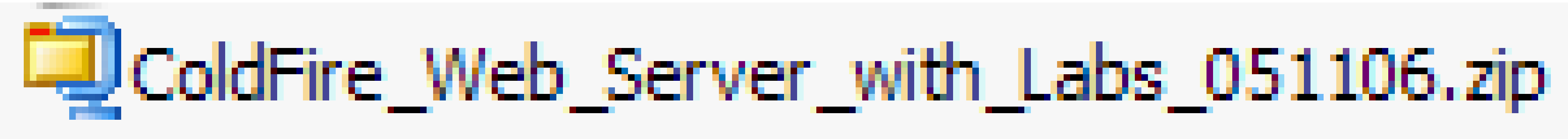
20 objects 48.7 MB Local intranet

Installing the ColdFireLite project and Labs

1. Find the ColdFire_Web_Server_with_Labs_?????.zip file

   on the CD ROM.

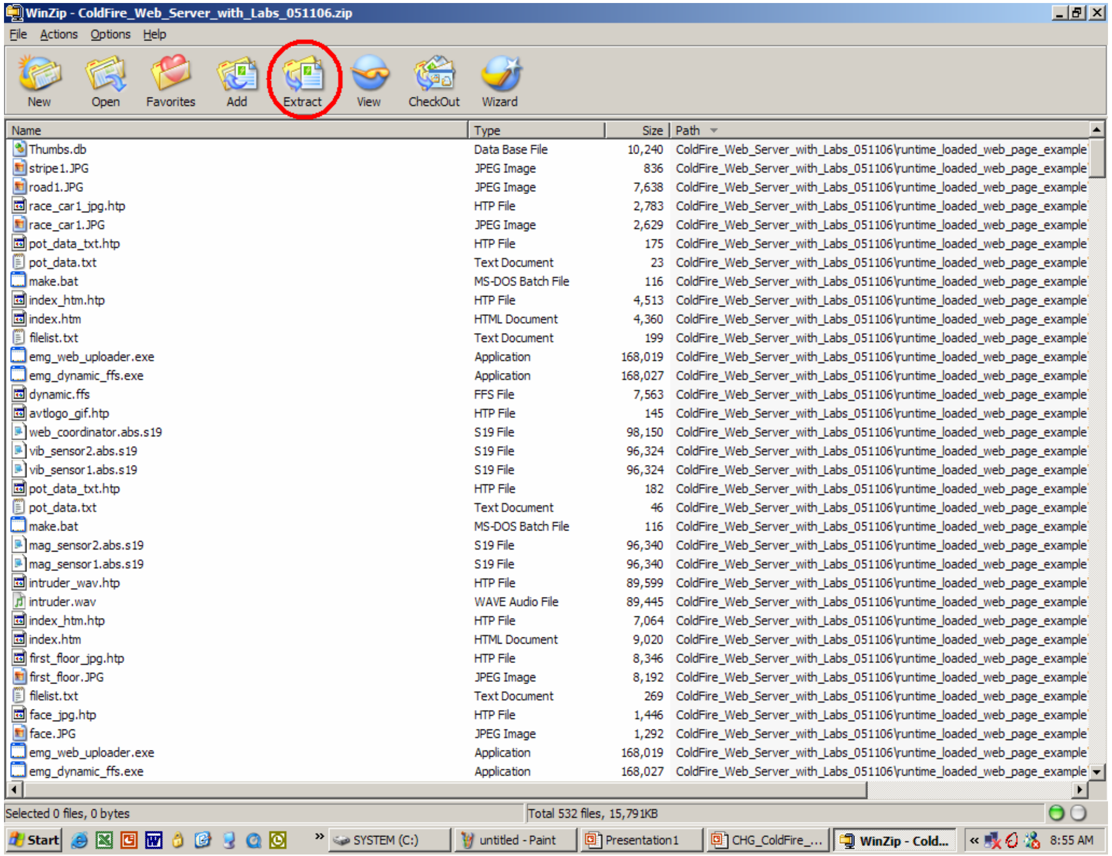
   (The ????? Is the date revision of the project, just select the latest and greatest if there are more then 1)



ColdFire_Web_Server_with_Labs_051106.zip

Installing the ColdFire Lite project and Labs

2. Un-Zip the ColdFire_Web_Server_with_Labs_?????.zip by double clicking on it.
   Select Extract button circled to open the unzip dialog.



WinZip - ColdFire_Web_Server_with_Labs_051106.zip
File Actions Options Help
New
Open
Favorites
Add
Extract
View
CheckOut
Wizard

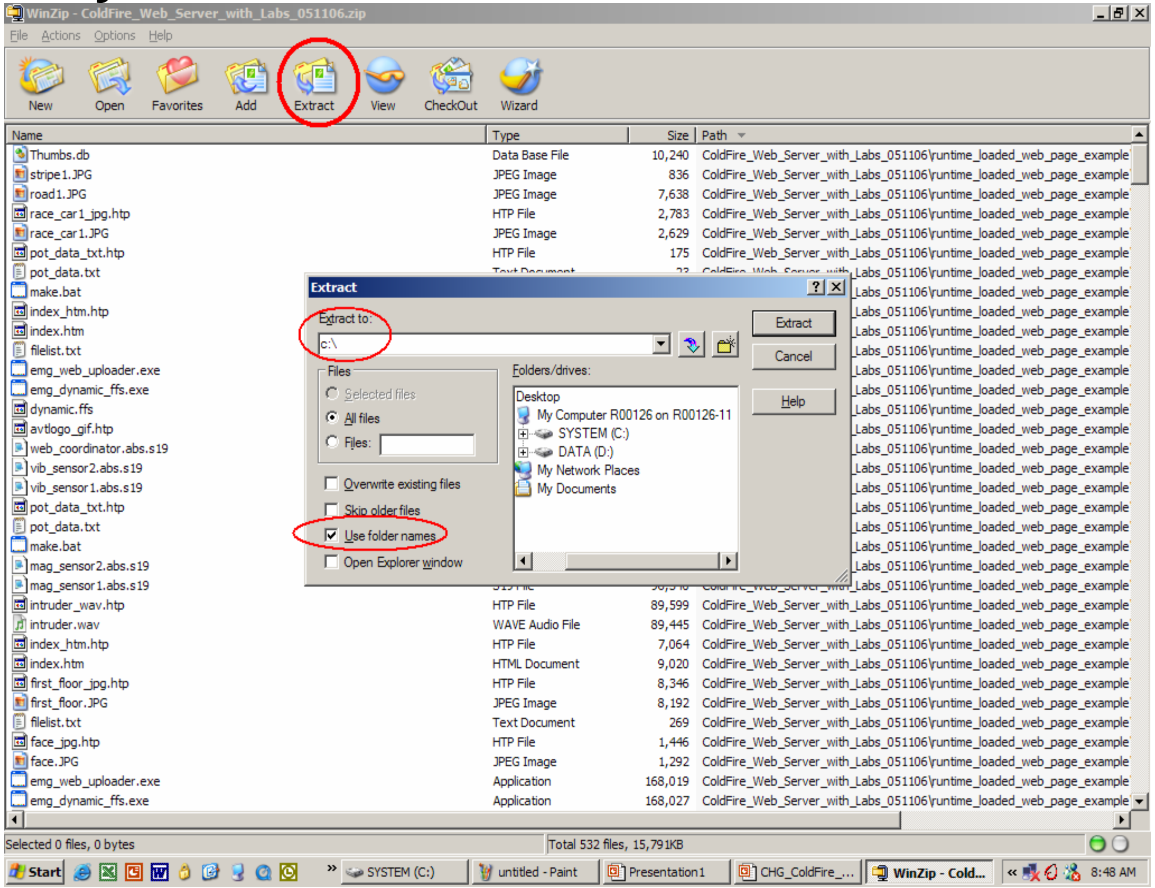
Name	Type	Size	Path
Thumbs.db	Data Base File	10,240	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
stripe1.JPG	JPEG Image	836	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
road1.JPG	JPEG Image	7,638	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
race_car1_jpg.htp	HTP File	2,783	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
race_car1.JPG	JPEG Image	2,629	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
pot_data_txt.htp	HTP File	175	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
pot_data.txt	Text Document	23	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
make.bat	MS-DOS Batch File	116	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
index_htm.htp	HTP File	4,513	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
index.htm	HTML Document	4,360	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
filelist.txt	Text Document	199	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
emg_web_uploader.exe	Application	168,019	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
emg_dynamic_ffs.exe	Application	168,027	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
dynamic.ffs	FFS File	7,563	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
avtlogo_gif.htp	HTP File	145	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
web_coordinator.abs.s19	S19 File	98,150	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
vib_sensor2.abs.s19	S19 File	96,324	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
vib_sensor1.abs.s19	S19 File	96,324	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
pot_data_txt.htp	HTP File	182	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
pot_data.txt	Text Document	46	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
make.bat	MS-DOS Batch File	116	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
mag_sensor2.abs.s19	S19 File	96,340	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
mag_sensor1.abs.s19	S19 File	96,340	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
intruder_wav.htp	HTP File	89,599	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
intruder.wav	WAVE Audio File	89,445	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
index_htm.htp	HTP File	7,064	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
index.htm	HTML Document	9,020	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
first_floor_jpg.htp	HTP File	8,346	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
first_floor.JPG	JPEG Image	8,192	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
filelist.txt	Text Document	269	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
face_jpg.htp	HTP File	1,446	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
face.JPG	JPEG Image	1,292	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
emg_web_uploader.exe	Application	168,019	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
emg_dynamic_ffs.exe	Application	168,027	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example

Selected 0 files, 0 bytes

Total 532 files, 15,791KB

Installing the ColdFire Lite project and Labs

3. In the extract dialog box, make sure the “use folder names” box is checked.
It is recommended that you extract to c root “C:\”



WinZip - ColdFire_Web_Server_with_Labs_051106.zip

Name	Type	Size	Path
Thumbs.db	Data Base File	10,240	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
stripe1.JPG	JPEG Image	836	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
road1.JPG	JPEG Image	7,638	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
race_car1_jpg.htp	HTP File	2,783	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
race_car1.JPG	JPEG Image	2,629	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
pot_data_txt.htp	HTP File	175	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
pot_data.txt	Text Document	23	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
make.bat			Labs_051106\runtime_loaded_web_page_example
index_htm.htp			Labs_051106\runtime_loaded_web_page_example
index.htm			Labs_051106\runtime_loaded_web_page_example
filelist.txt			Labs_051106\runtime_loaded_web_page_example
emg_web_uploader.exe			Labs_051106\runtime_loaded_web_page_example
emg_dynamic_ffs.exe			Labs_051106\runtime_loaded_web_page_example
dynamic.ffs			Labs_051106\runtime_loaded_web_page_example
avtlogo_gif.htp			Labs_051106\runtime_loaded_web_page_example
web_coordinator.abs.s19			Labs_051106\runtime_loaded_web_page_example
vib_sensor2.abs.s19			Labs_051106\runtime_loaded_web_page_example
vib_sensor1.abs.s19			Labs_051106\runtime_loaded_web_page_example
pot_data_txt.htp			Labs_051106\runtime_loaded_web_page_example
pot_data.txt			Labs_051106\runtime_loaded_web_page_example
make.bat			Labs_051106\runtime_loaded_web_page_example
mag_sensor2.abs.s19			Labs_051106\runtime_loaded_web_page_example
mag_sensor1.abs.s19	S19 File	[illegible]	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
intruder_wav.htp	HTP File	89,599	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
intruder.wav	WAVE Audio File	89,445	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
index_htm.htp	HTP File	7,064	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
index.htm	HTML Document	9,020	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
first_floor_jpg.htp	HTP File	8,346	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
first_floor.JPG	JPEG Image	8,192	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
filelist.txt	Text Document	269	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
face_jpg.htp	HTP File	1,446	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
face.JPG	JPEG Image	1,292	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
emg_web_uploader.exe	Application	168,019	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example
emg_dynamic_ffs.exe	Application	168,027	ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example

Extract

Extract to: c:\

Files
- ○ Selected files
- ● All files
- ○ Files:

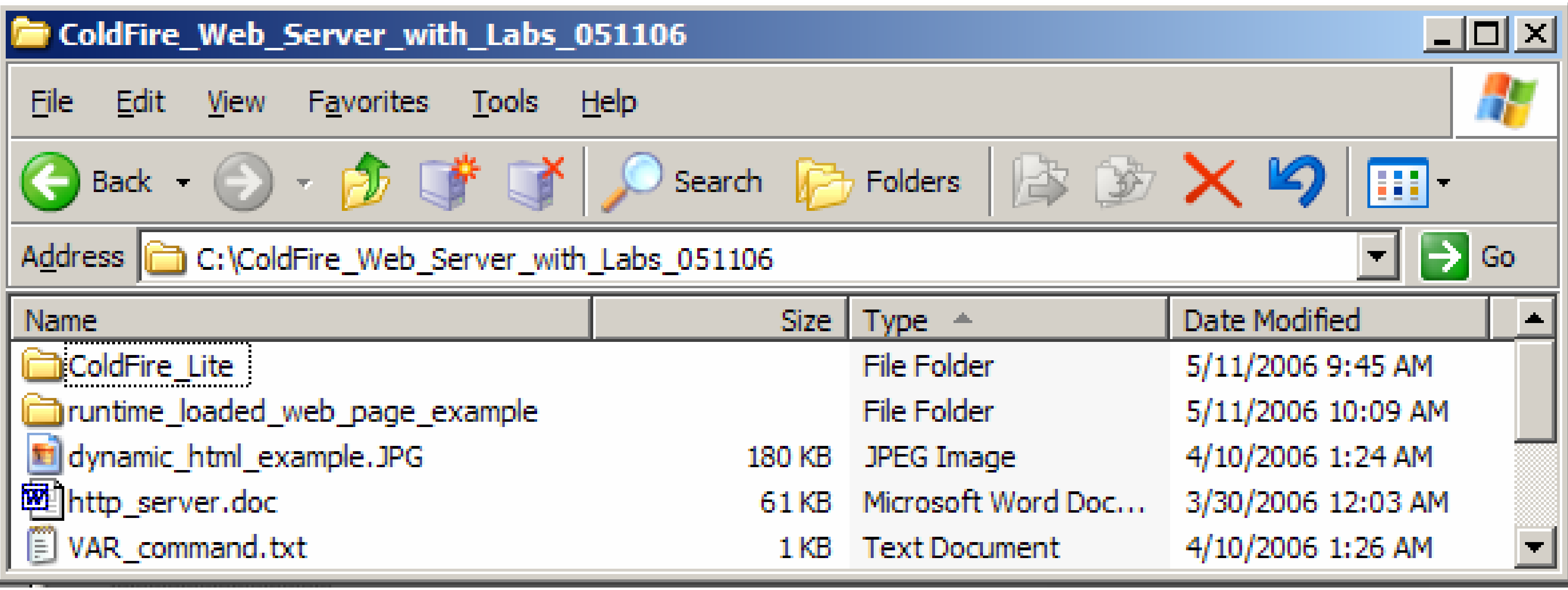
- □ Overwrite existing files
- □ Skip older files
- ☑ Use folder names
- □ Open Explorer window

Folders/drives:
Desktop
My Computer R00126 on R00126-11
SYSTEM (C:)
DATA (D:)
My Network Places
My Documents

Selected 0 files, 0 bytes

Total 532 files, 15,791KB

The Directory Structure



C:\ColdFire_Web_Server_with_Labs_051106

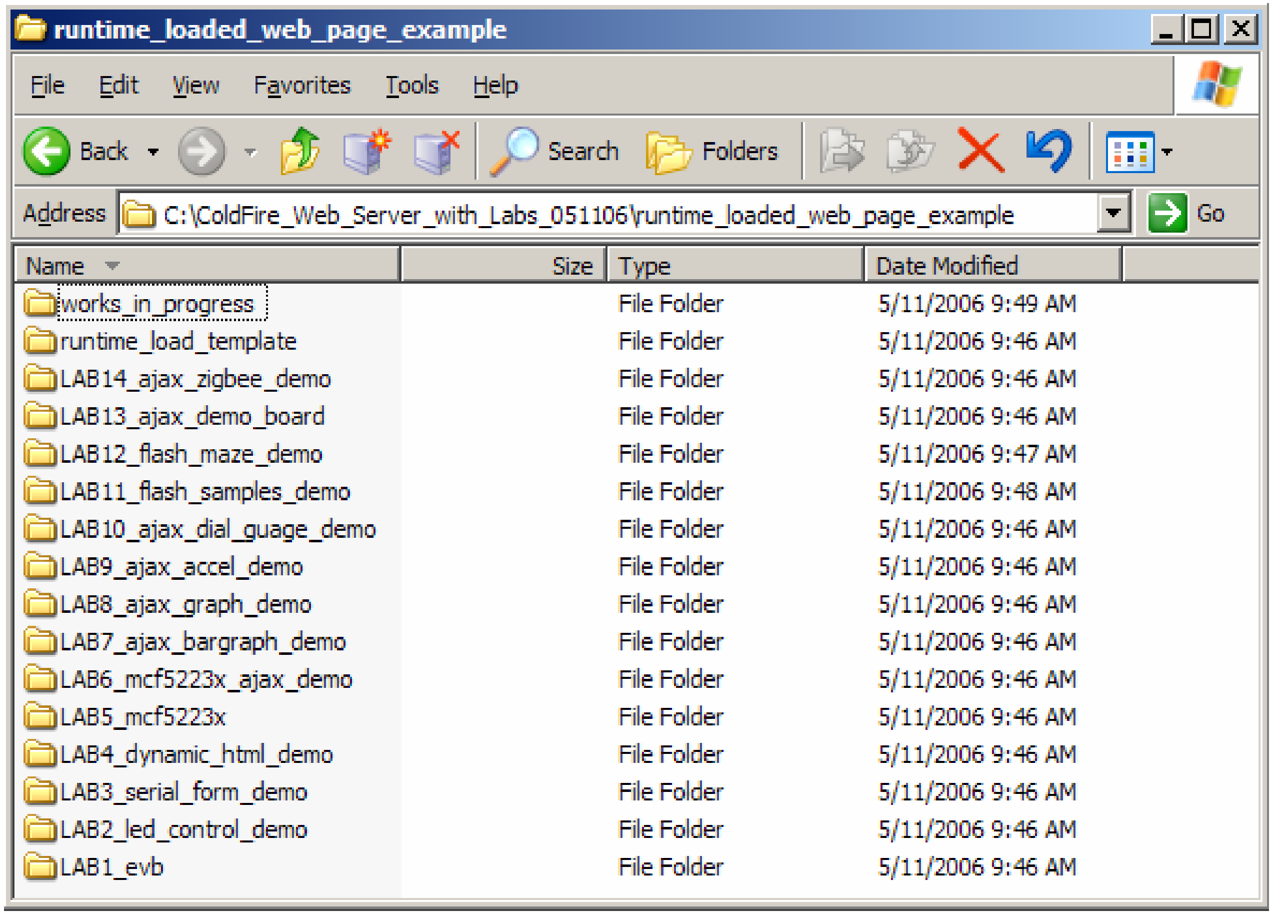
Name	Size	Type	Date Modified
ColdFire_Lite		File Folder	5/11/2006 9:45 AM
runtime_loaded_web_page_example		File Folder	5/11/2006 10:09 AM
dynamic_html_example.JPG	180 KB	JPEG Image	4/10/2006 1:24 AM
http_server.doc	61 KB	Microsoft Word Doc...	3/30/2006 12:03 AM
VAR_command.txt	1 KB	Text Document	4/10/2006 1:26 AM

Directory Details

• VAR_command.txt	Documentation on the VAR command.
• http_server.doc	A overview of the web server.
• Dynamic_html_example.jpg	A picture speaks a 1000 words.

Directory Details – Runtime Loadable Demos/Labs

Runtime_loaded_web_page_example directory
This directory contains the runtime loadable demos/labs.



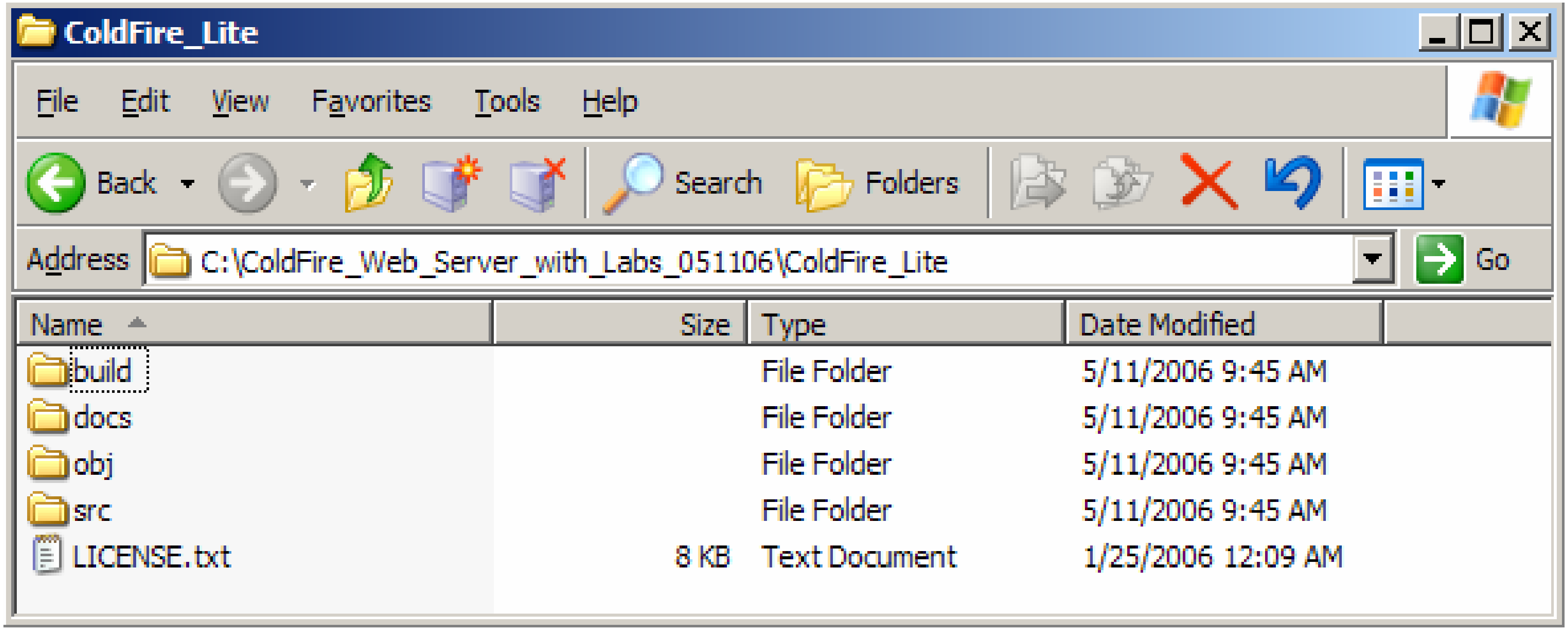
C:\ColdFire_Web_Server_with_Labs_051106\runtime_loaded_web_page_example

Name	Size	Type	Date Modified
works_in_progress		File Folder	5/11/2006 9:49 AM
runtime_load_template		File Folder	5/11/2006 9:46 AM
LAB14_ajax_zigbee_demo		File Folder	5/11/2006 9:46 AM
LAB13_ajax_demo_board		File Folder	5/11/2006 9:46 AM
LAB12_flash_maze_demo		File Folder	5/11/2006 9:47 AM
LAB11_flash_samples_demo		File Folder	5/11/2006 9:48 AM
LAB10_ajax_dial_guage_demo		File Folder	5/11/2006 9:46 AM
LAB9_ajax_accel_demo		File Folder	5/11/2006 9:46 AM
LAB8_ajax_graph_demo		File Folder	5/11/2006 9:46 AM
LAB7_ajax_bargraph_demo		File Folder	5/11/2006 9:46 AM
LAB6_mcf5223x_ajax_demo		File Folder	5/11/2006 9:46 AM
LAB5_mcf5223x		File Folder	5/11/2006 9:46 AM
LAB4_dynamic_html_demo		File Folder	5/11/2006 9:46 AM
LAB3_serial_form_demo		File Folder	5/11/2006 9:46 AM
LAB2_led_control_demo		File Folder	5/11/2006 9:46 AM
LAB1_evb		File Folder	5/11/2006 9:46 AM

Directory Details – ColdFire_Lite

ColdFire_Lite directory

The ColdFire_Lite directory contains the TCP/IP stack and Web Server Firmware.

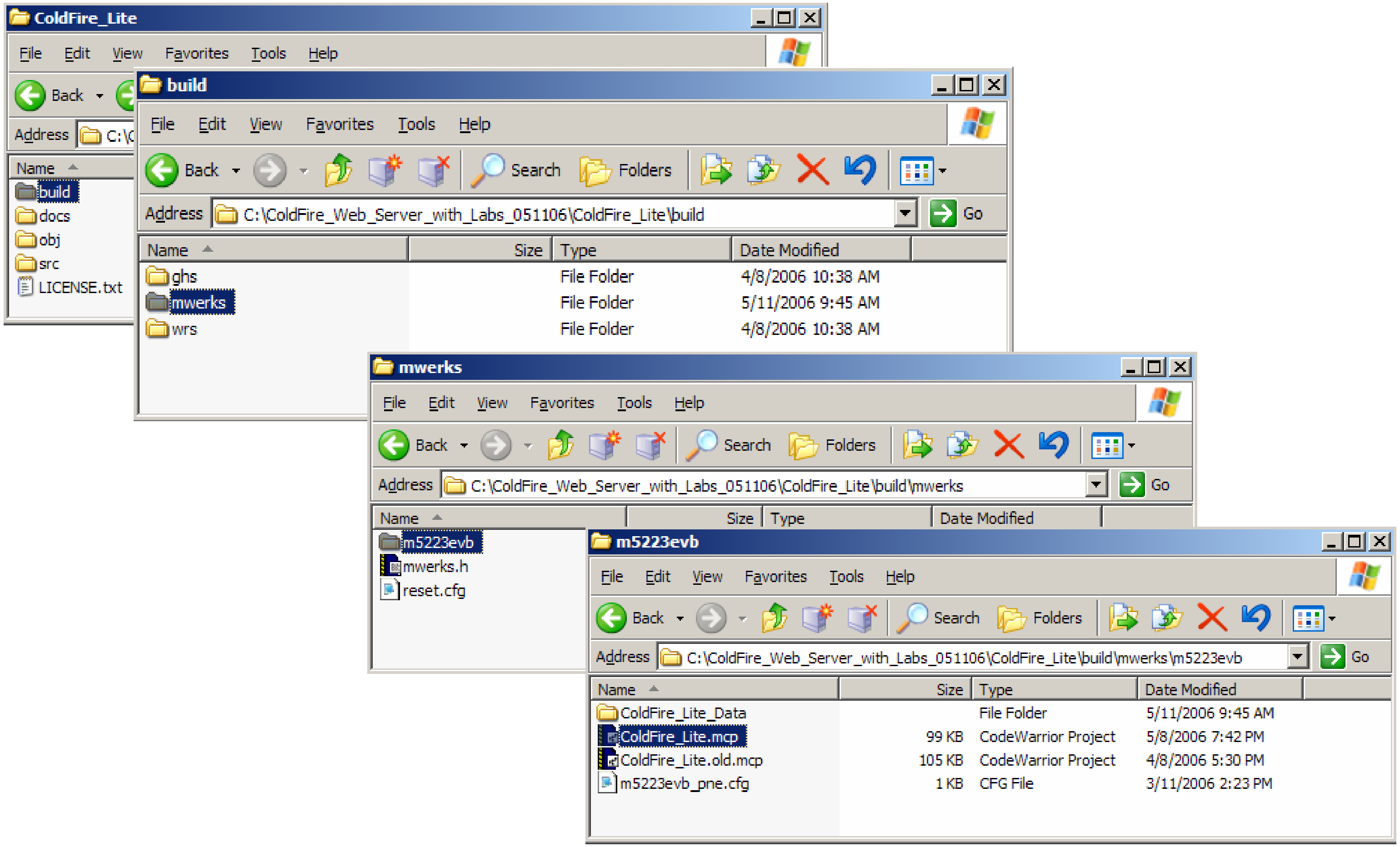


C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite

Name	Size	Type	Date Modified
build		File Folder	5/11/2006 9:45 AM
docs		File Folder	5/11/2006 9:45 AM
obj		File Folder	5/11/2006 9:45 AM
src		File Folder	5/11/2006 9:45 AM
LICENSE.txt	8 KB	Text Document	1/25/2006 12:09 AM

ColdFire_Lite Project File

- The project File is used to open the project in CodeWarrior®.



ColdFire_Lite

Address C:\C

Name
build
docs
obj
src
LICENSE.txt

build

Address C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\build

Name	Size	Type	Date Modified
ghs		File Folder	4/8/2006 10:38 AM
mwerks		File Folder	5/11/2006 9:45 AM
wrs		File Folder	4/8/2006 10:38 AM

mwerks

Address C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\build\mwerks

Name	Size	Type	Date Modified
m5223evb			
mwerks.h			
reset.cfg			

m5223evb

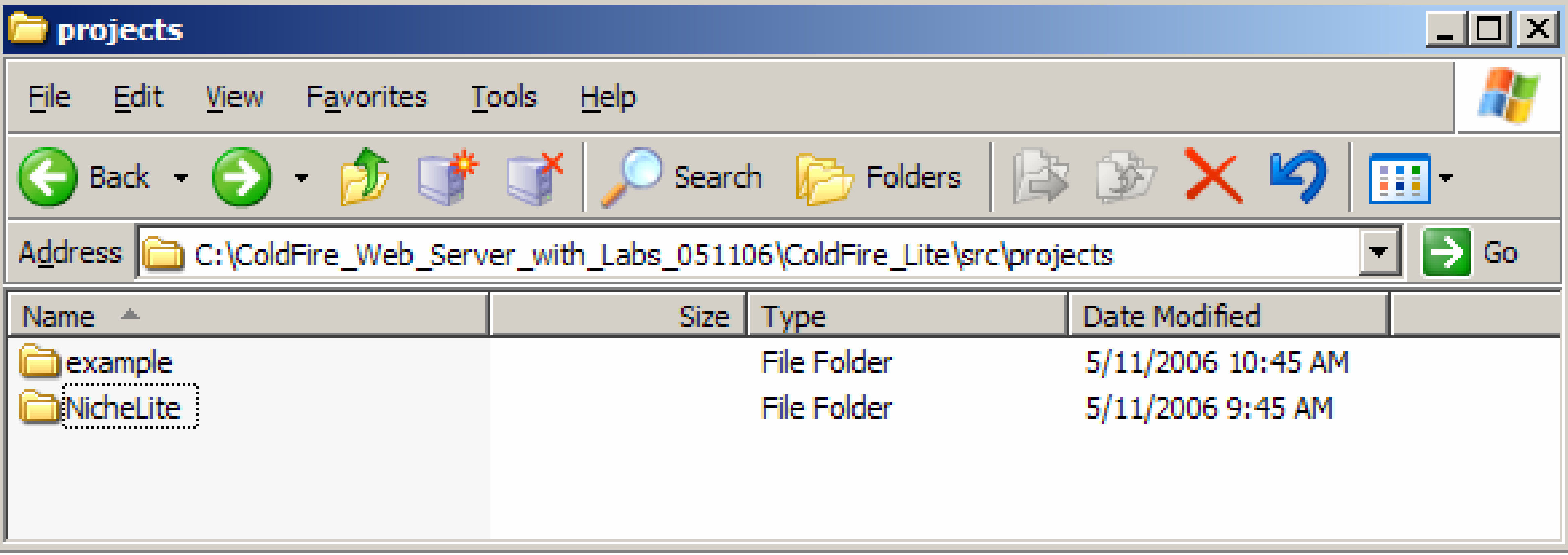
Address C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\build\mwerks\m5223evb

Name	Size	Type	Date Modified
ColdFire_Lite_Data		File Folder	5/11/2006 9:45 AM
ColdFire_Lite.mcp	99 KB	CodeWarrior Project	5/8/2006 7:42 PM
ColdFire_Lite.old.mcp	105 KB	CodeWarrior Project	4/8/2006 5:30 PM
m5223evb_pne.cfg	1 KB	CFG File	3/11/2006 2:23 PM

The Niche Lite directory

The NicheLite directory contains the source to the TCP/IP stack.

ColdFire_Lite\src\projects



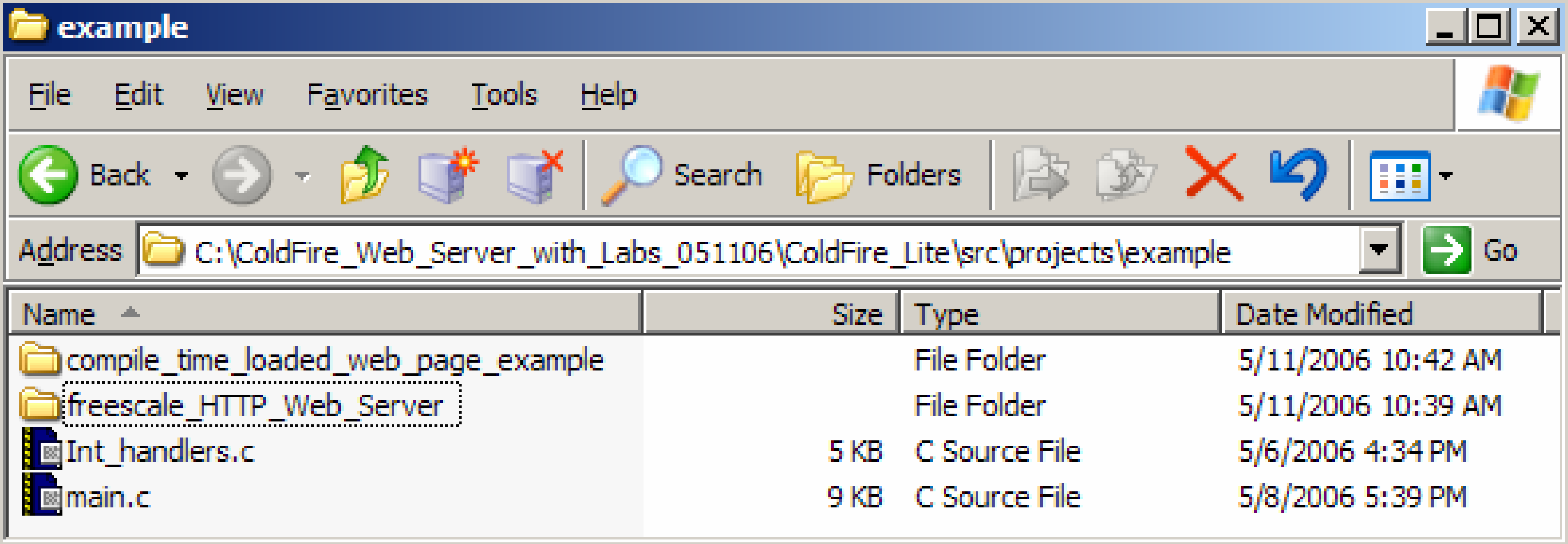
C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\src\projects

Name	Size	Type	Date Modified
example		File Folder	5/11/2006 10:45 AM
NicheLite		File Folder	5/11/2006 9:45 AM

Freescale_HTTP_Web_Server directory

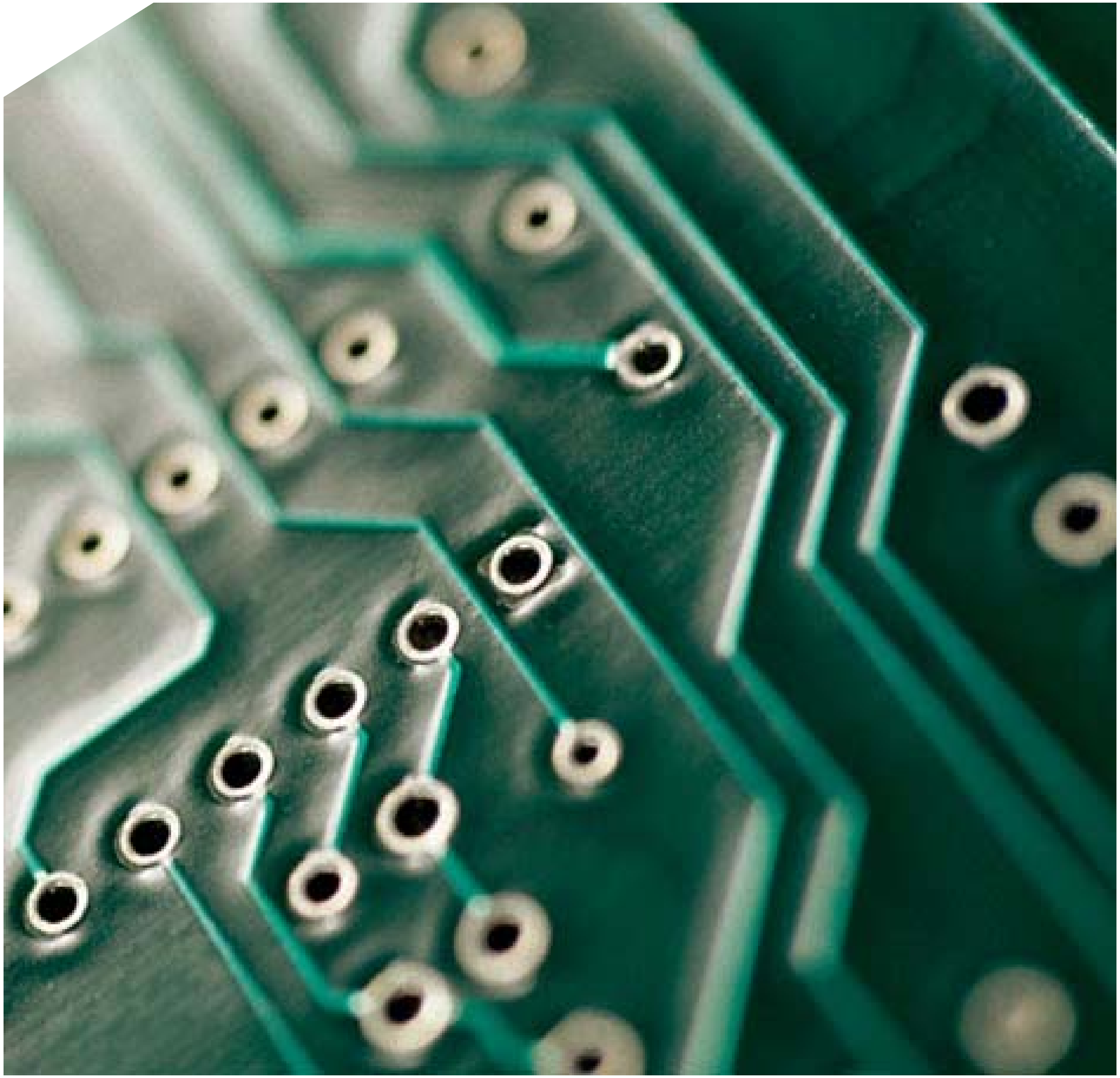
The Freescale_HTTP_Web_Server directory contains the source code for the Freescale Web Server.

ColdFire_Lite\src\projects\example



C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\src\projects\example

Name	Size	Type	Date Modified
compile_time_loaded_web_page_example		File Folder	5/11/2006 10:42 AM
freescale_HTTP_Web_Server		File Folder	5/11/2006 10:39 AM
Int_handlers.c	5 KB	C Source File	5/6/2006 4:34 PM
main.c	9 KB	C Source File	5/8/2006 5:39 PM



Getting Started with CodeWarrior® (for ColdFire®)

A Hands On Lab based on the M52233DEMO

freescale™ semiconductor

Lab 2: Using the Source Code Browser

- In this exercise you will become familiar with the IDE’s (Integrated Development Environment) code navigation features. These include features from both the editor and the source code browser.
- This exercise concentrates on navigating between files that are related to each other and, on how to jump to interesting places in your code.
- For this LAB we will debug the actual TCP/IP stack and Web Server project.

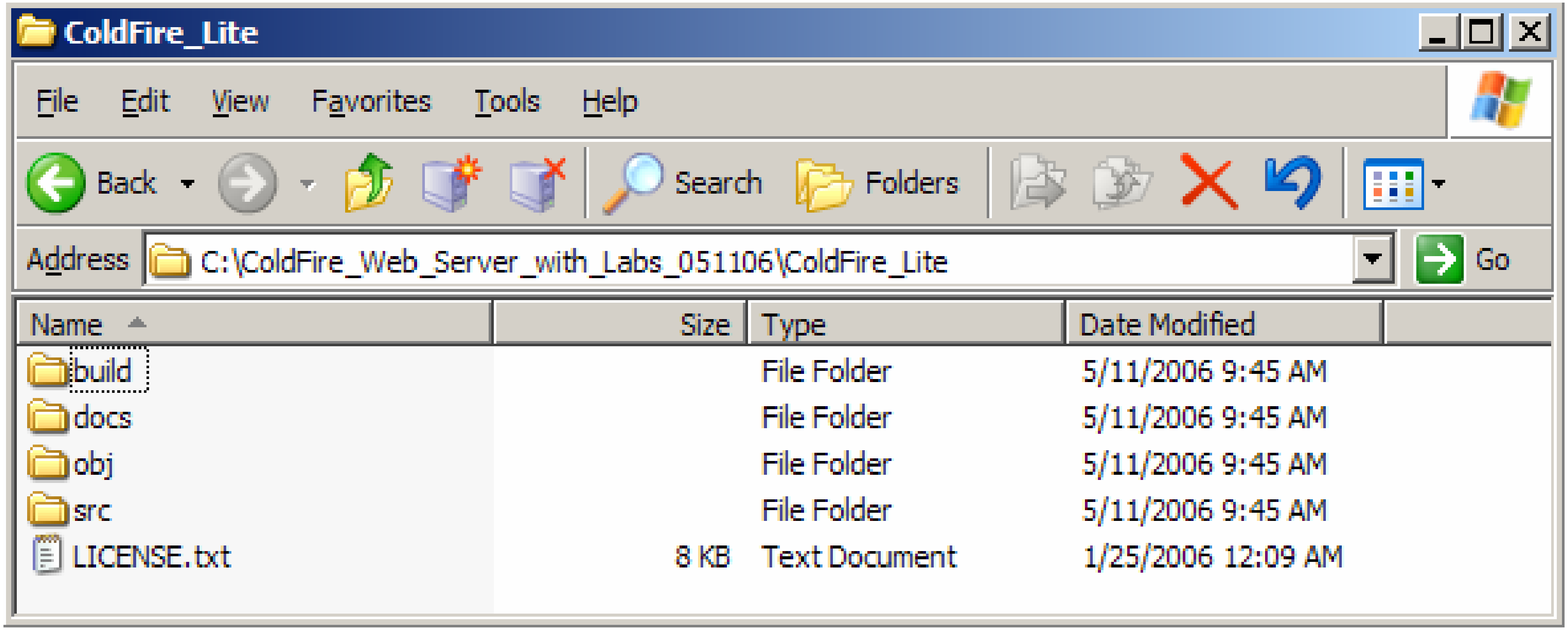
Locate and Open the TCP/IP/Web Server Project

- Close all open CodeWarrior® Project Windows.
- Choose File > Open
- Browse to the ColdFire Lite Directory.
  - This will be located where you unzipped the ColdFire Lite project, or if you are using a Freescale laptop

Directory Details – ColdFire_Lite

ColdFire_Lite directory

The ColdFire_Lite directory contains the TCP/IP stack and Web Server Firmware.

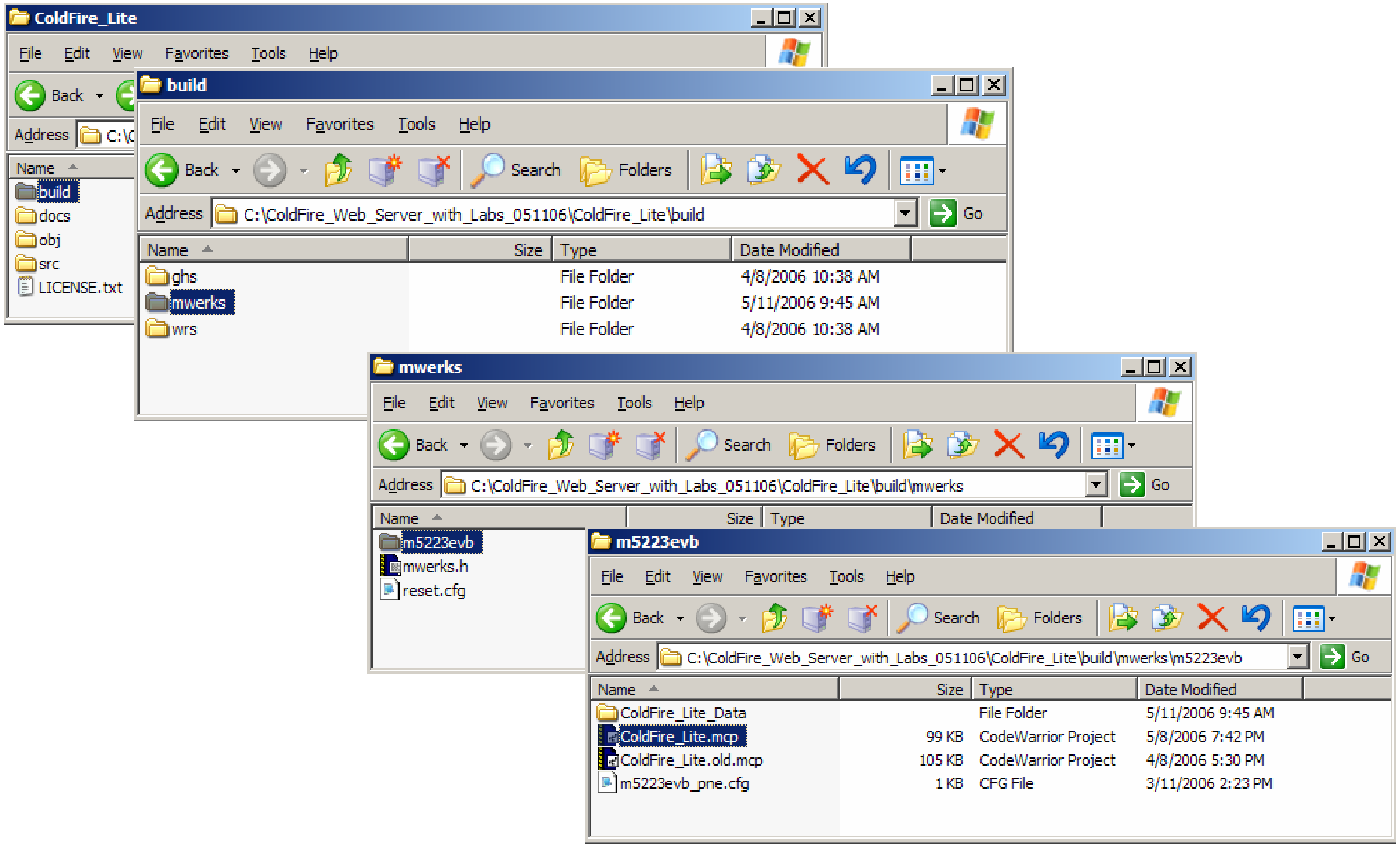


C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite

Name	Size	Type	Date Modified
build		File Folder	5/11/2006 9:45 AM
docs		File Folder	5/11/2006 9:45 AM
obj		File Folder	5/11/2006 9:45 AM
src		File Folder	5/11/2006 9:45 AM
LICENSE.txt	8 KB	Text Document	1/25/2006 12:09 AM

ColdFire_Lite Project File

- The project File is used to open the project in CodeWarrior®.


ColdFire_Lite
File Edit View Favorites Tools Help
Back
build
File Edit View Favorites Tools Help
Back Search Folders
Go
mwerks
File Edit View Favorites Tools Help
Back Search Folders
Go
m5223evb
File Edit View Favorites Tools Help
Back Search Folders
Go

Address C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\build

Name	Size	Type	Date Modified
ghs		File Folder	4/8/2006 10:38 AM
mwerks		File Folder	5/11/2006 9:45 AM
wrs		File Folder	4/8/2006 10:38 AM

Name
build
docs
obj
src
LICENSE.txt

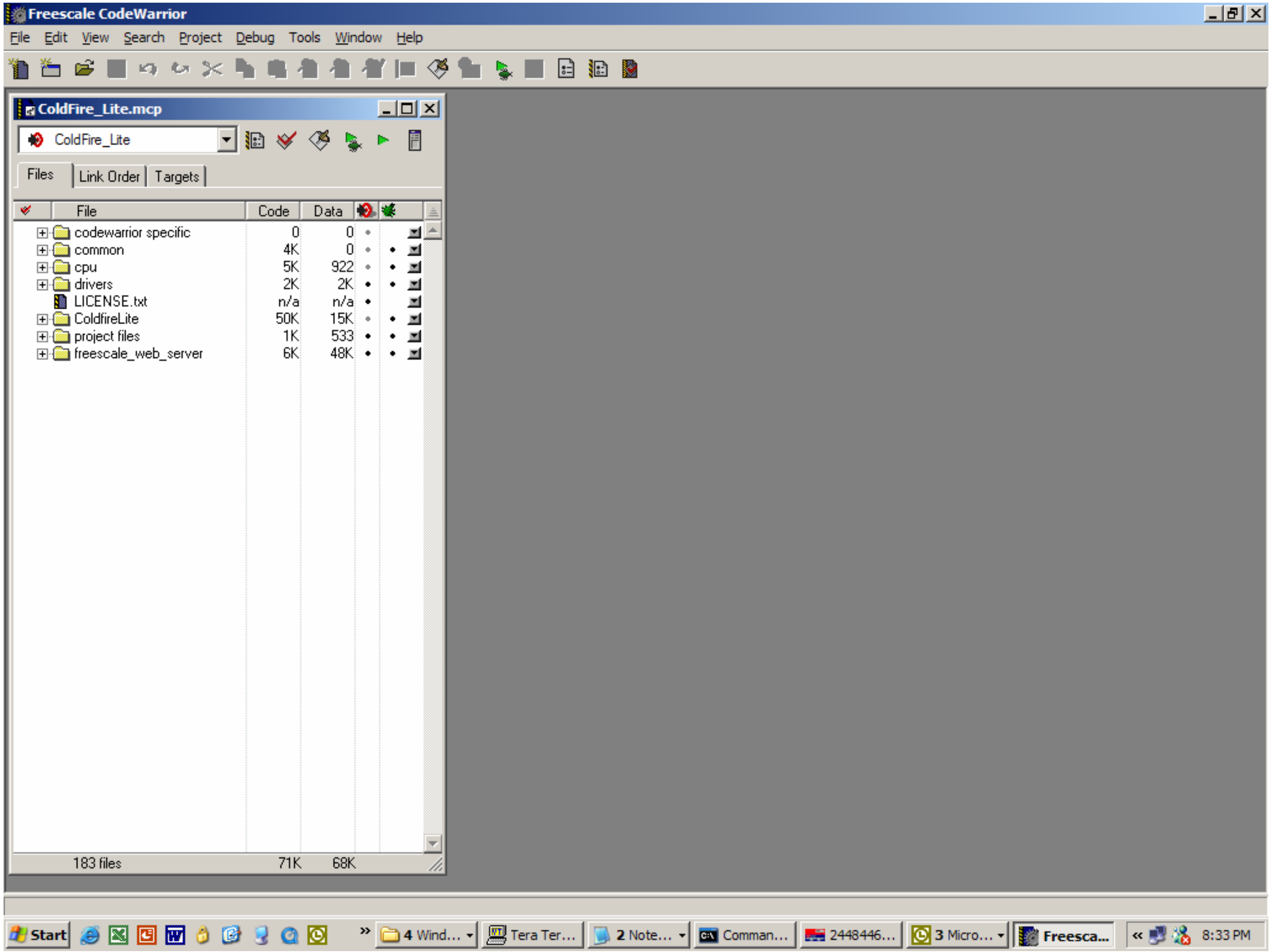
Address C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\build\mwerks

Name	Size	Type	Date Modified
m5223evb			
mwerks.h			
reset.cfg			

Address C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\build\mwerks\m5223evb

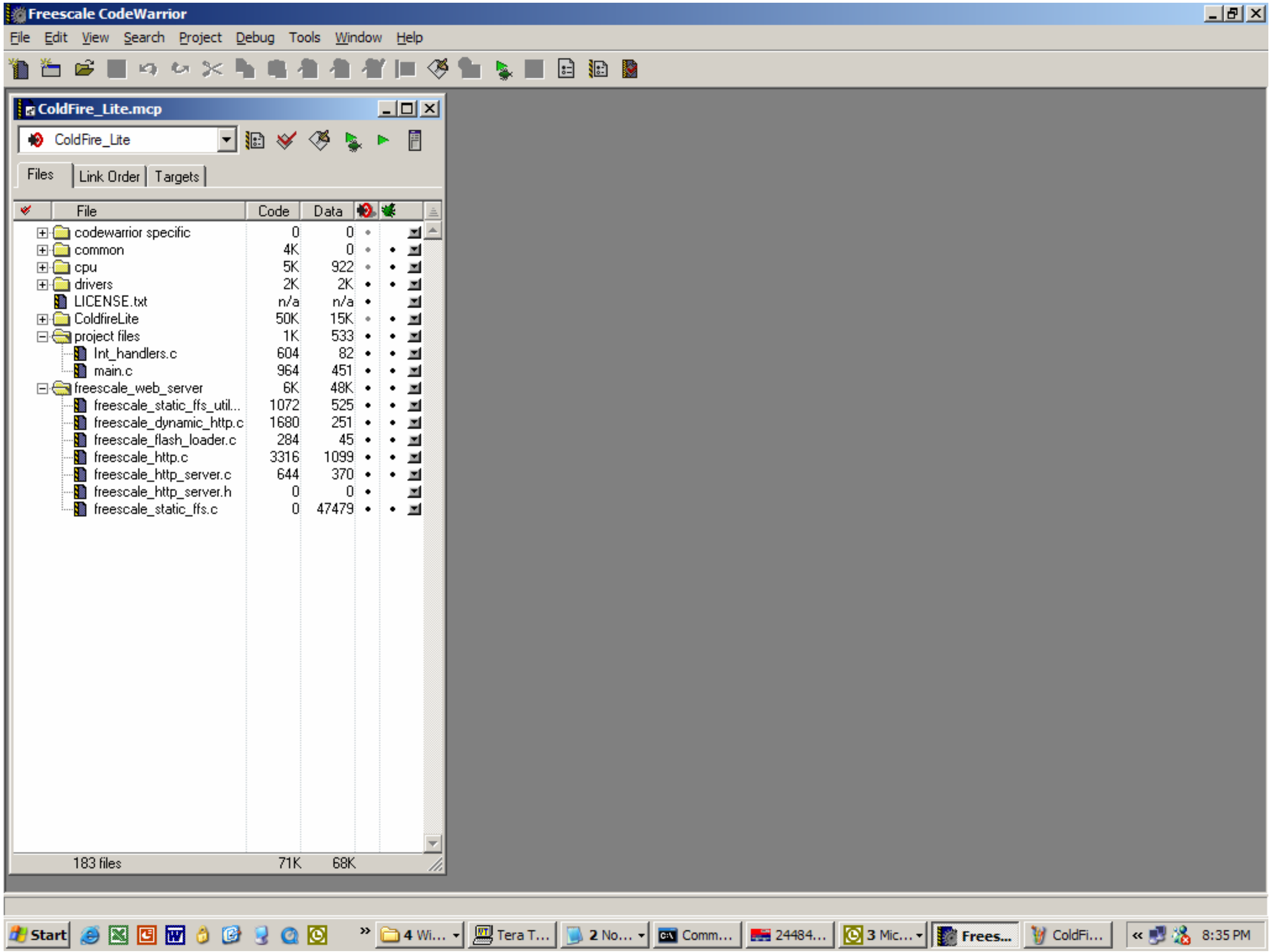
Name	Size	Type	Date Modified
ColdFire_Lite_Data		File Folder	5/11/2006 9:45 AM
ColdFire_Lite.mcp	99 KB	CodeWarrior Project	5/8/2006 7:42 PM
ColdFire_Lite.old.mcp	105 KB	CodeWarrior Project	4/8/2006 5:30 PM
m5223evb_pne.cfg	1 KB	CFG File	3/11/2006 2:23 PM

The ColdFire Lite project



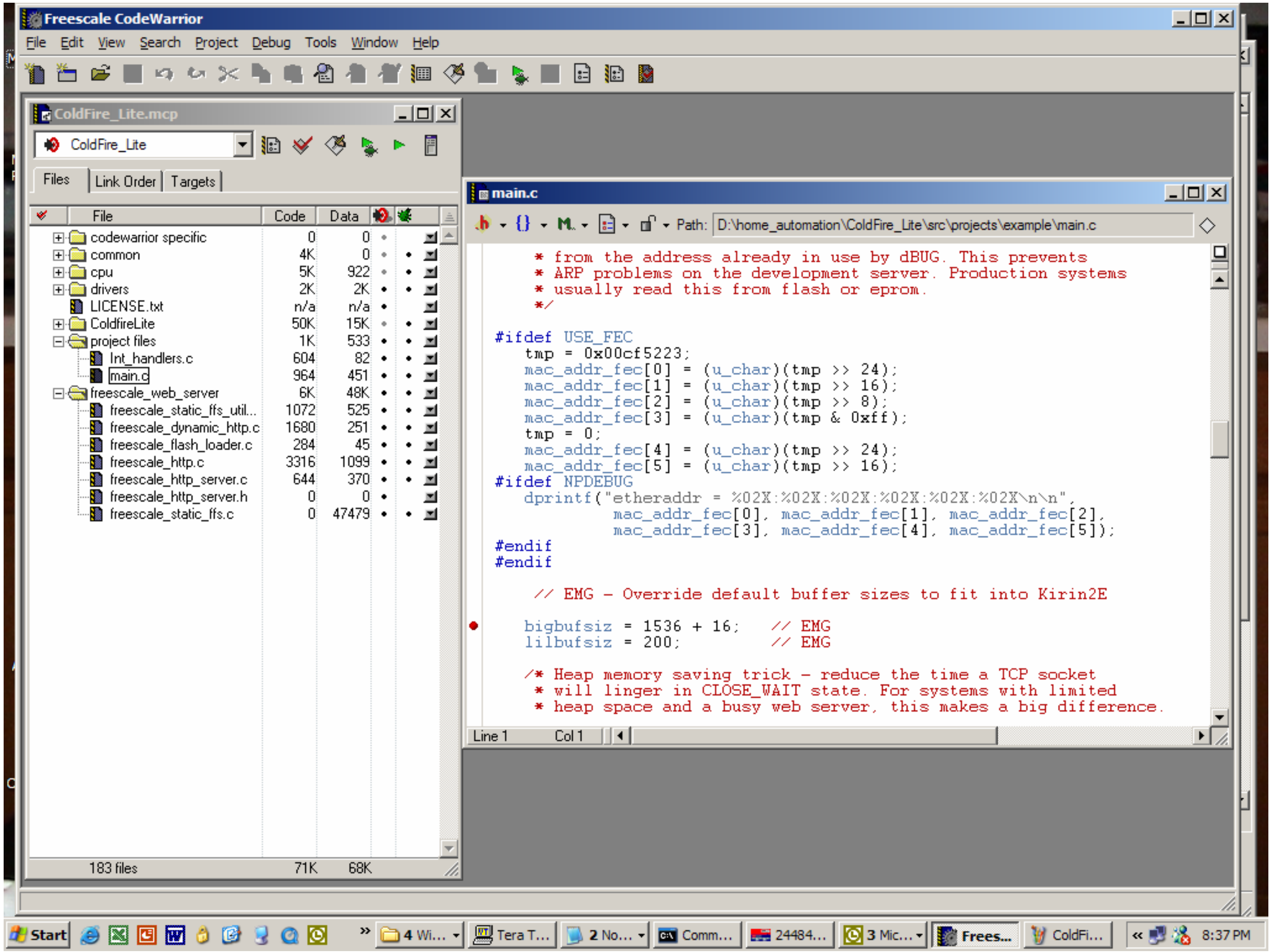
File	Code	Data		
codewarrior specific	0	0	•	
common	4K	0	•	•
cpu	5K	922	•	•
drivers	2K	2K	•	•
LICENSE.txt	n/a	n/a	•	
ColdfireLite	50K	15K	•	•
project files	1K	533	•	•
freescale_web_server	6K	48K	•	•
183 files	71K	68K		

Open The directory Containing MAIN.C



✓	File	Code	Data			
	⊞ codewarrior specific	0	0	·		
	⊞ common	4K	0	·	•	
	⊞ cpu	5K	922	·	•	
	⊞ drivers	2K	2K	•	•	
	LICENSE.txt	n/a	n/a	•		
	⊞ ColdfireLite	50K	15K	·	•	
	⊟ project files	1K	533	•	•	
	Int_handlers.c	604	82	•	•	
	main.c	964	451	•	•	
	⊟ freescale_web_server	6K	48K	•	•	
	freescale_static_ffs_util...	1072	525	•	•	
	freescale_dynamic_http.c	1680	251	•	•	
	freescale_flash_loader.c	284	45	•	•	
	freescale_http.c	3316	1099	•	•	
	freescale_http_server.c	644	370	•	•	
	freescale_http_server.h	0	0	•		
	freescale_static_ffs.c	0	47479	•	•	
	183 files	71K	68K			

Open main.c

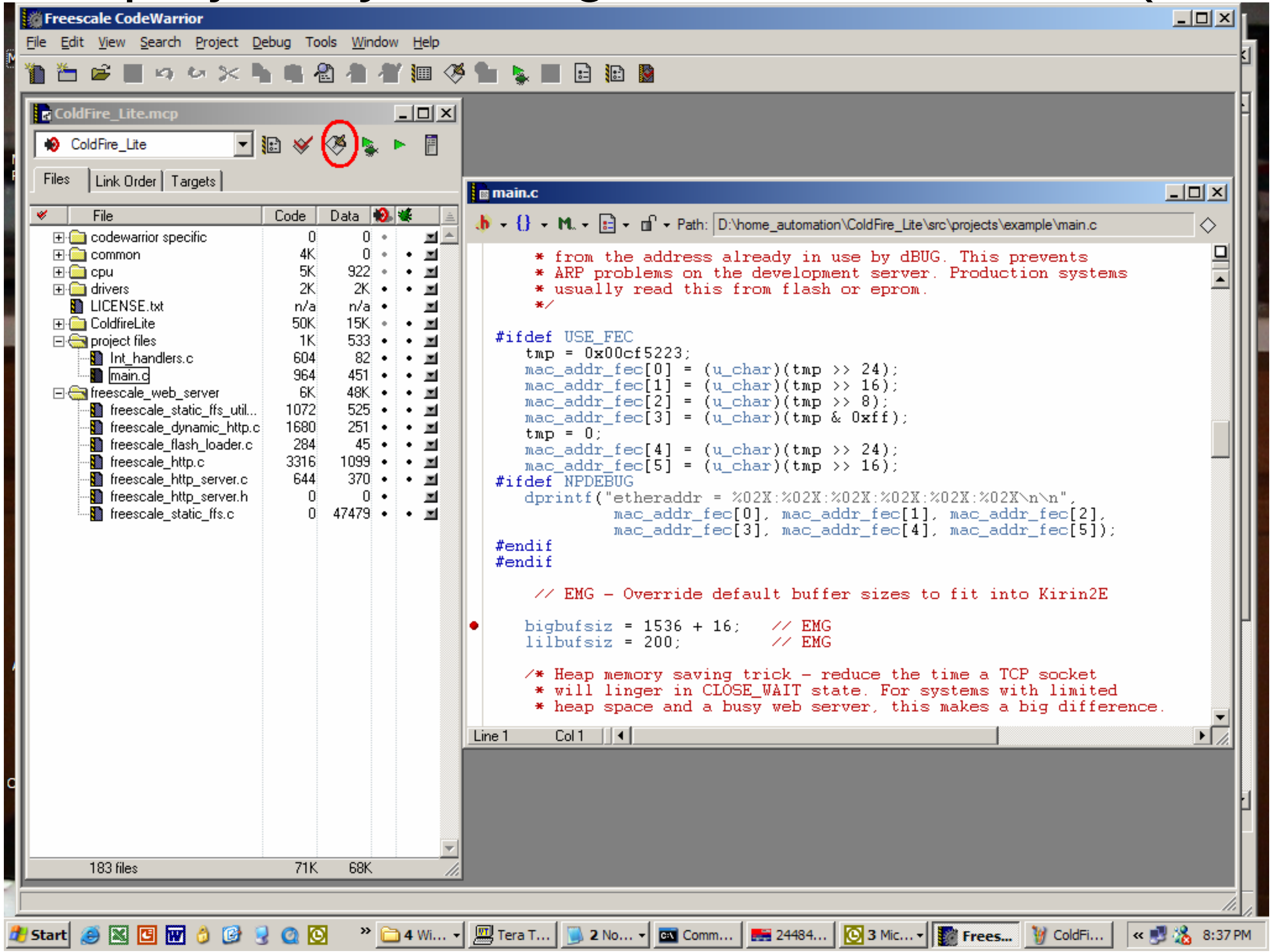


File	Code	Data		
codewarrior specific	0	0	•	
common	4K	0	•	•
cpu	5K	922	•	•
drivers	2K	2K	•	•
LICENSE.txt	n/a	n/a	•	
ColdfireLite	50K	15K	•	•
project files	1K	533	•	•
Int_handlers.c	604	82	•	•
main.c	964	451	•	•
freescale_web_server	6K	48K	•	•
freescale_static_ffs_util...	1072	525	•	•
freescale_dynamic_http.c	1680	251	•	•
freescale_flash_loader.c	284	45	•	•
freescale_http.c	3316	1099	•	•
freescale_http_server.c	644	370	•	•
freescale_http_server.h	0	0	•	
freescale_static_ffs.c	0	47479	•	•
183 files	71K	68K		

```
     * from the address already in use by dBUG. This prevents
     * ARP problems on the development server. Production systems
     * usually read this from flash or eprom.
     */

#ifdef USE_FEC
   tmp = 0x00cf5223;
   mac_addr_fec[0] = (u_char)(tmp >> 24);
   mac_addr_fec[1] = (u_char)(tmp >> 16);
   mac_addr_fec[2] = (u_char)(tmp >> 8);
   mac_addr_fec[3] = (u_char)(tmp & 0xff);
   tmp = 0;
   mac_addr_fec[4] = (u_char)(tmp >> 24);
   mac_addr_fec[5] = (u_char)(tmp >> 16);
#ifdef NPDEBUG
   dprintf("etheraddr = %02X:%02X:%02X:%02X:%02X:%02X\n\n",
            mac_addr_fec[0], mac_addr_fec[1], mac_addr_fec[2],
            mac_addr_fec[3], mac_addr_fec[4], mac_addr_fec[5]);
#endif
#endif

    // EMG - Override default buffer sizes to fit into Kirin2E

   bigbufsiz = 1536 + 16;   // EMG
   lilbufsiz = 200;         // EMG

   /* Heap memory saving trick - reduce the time a TCP socket
    * will linger in CLOSE_WAIT state. For systems with limited
    * heap space and a busy web server, this makes a big difference.
```

Building the Project

Build the project by clicking on the MAKE icon (circled in **RED**)



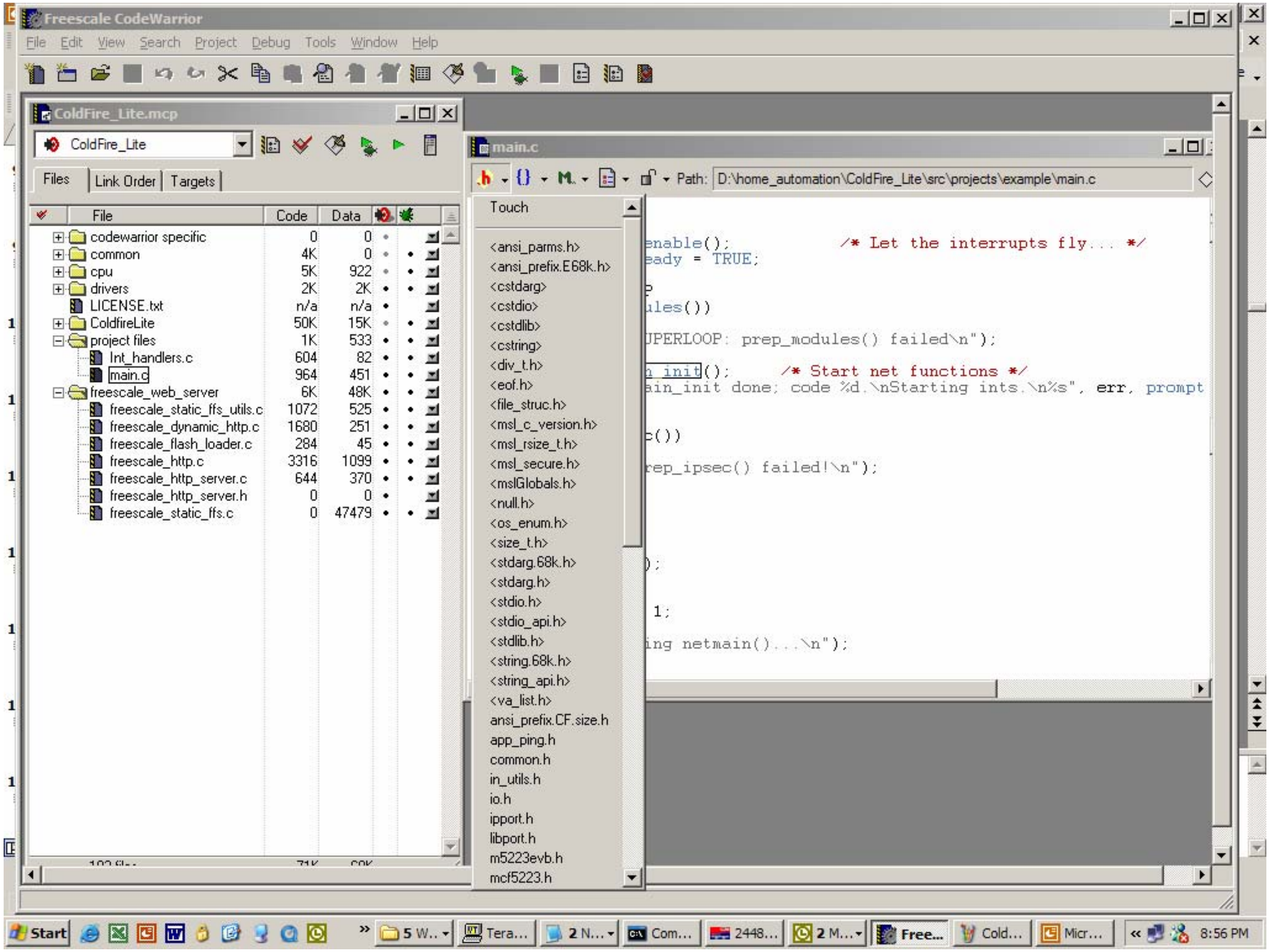
File	Code	Data			
codewarrior specific	0	0	●		
common	4K	0	●	●	
cpu	5K	922	●	●	
drivers	2K	2K	●	●	
LICENSE.txt	n/a	n/a	●		
ColdfireLite	50K	15K	●	●	
project files	1K	533	●	●	
Int_handlers.c	604	82	●	●	
main.c	964	451	●	●	
freescale_web_server	6K	48K	●	●	
freescale_static_ffs_util...	1072	525	●	●	
freescale_dynamic_http.c	1680	251	●	●	
freescale_flash_loader.c	284	45	●	●	
freescale_http.c	3316	1099	●	●	
freescale_http_server.c	644	370	●	●	
freescale_http_server.h	0	0	●		
freescale_static_ffs.c	0	47479	●	●	
183 files	71K	68K			

```
     * from the address already in use by dBUG. This prevents
     * ARP problems on the development server. Production systems
     * usually read this from flash or eprom.
     */

#ifdef USE_FEC
   tmp = 0x00cf5223;
   mac_addr_fec[0] = (u_char)(tmp >> 24);
   mac_addr_fec[1] = (u_char)(tmp >> 16);
   mac_addr_fec[2] = (u_char)(tmp >> 8);
   mac_addr_fec[3] = (u_char)(tmp & 0xff);
   tmp = 0;
   mac_addr_fec[4] = (u_char)(tmp >> 24);
   mac_addr_fec[5] = (u_char)(tmp >> 16);
#ifdef NPDEBUG
   dprintf("etheraddr = %02X:%02X:%02X:%02X:%02X:%02X\n\n",
            mac_addr_fec[0], mac_addr_fec[1], mac_addr_fec[2],
            mac_addr_fec[3], mac_addr_fec[4], mac_addr_fec[5]);
#endif
#endif

    // EMG - Override default buffer sizes to fit into Kirin2E

   bigbufsiz = 1536 + 16;   // EMG
   lilbufsiz = 200;         // EMG

   /* Heap memory saving trick - reduce the time a TCP socket
    * will linger in CLOSE_WAIT state. For systems with limited
    * heap space and a busy web server, this makes a big difference.
```

Open Related Header Files

- The Edit windows provides a quick access to all files included (directly or indirectly in a project).
- Open the file **main.c** in the editor. Double click on the file name in the project window.
- Click on the arrow next to the .h icon. You see a list of all header files directly or indirectly included in the source file.

File Dependencies



File	Code	Data		
⊞ codewarrior specific	0	0	•	
⊞ common	4K	0	•	•
⊞ cpu	5K	922	•	•
⊞ drivers	2K	2K	•	•
LICENSE.txt	n/a	n/a	•	
⊞ ColdfireLite	50K	15K	•	•
⊟ project files	1K	533	•	•
Int_handlers.c	604	82	•	•
main.c	964	451	•	•
⊟ freescale_web_server	6K	48K	•	•
freescale_static_ffs_utils.c	1072	525	•	•
freescale_dynamic_http.c	1680	251	•	•
freescale_flash_loader.c	284	45	•	•
freescale_http.c	3316	1099	•	•
freescale_http_server.c	644	370	•	•
freescale_http_server.h	0	0	•	
freescale_static_ffs.c	0	47479	•	•

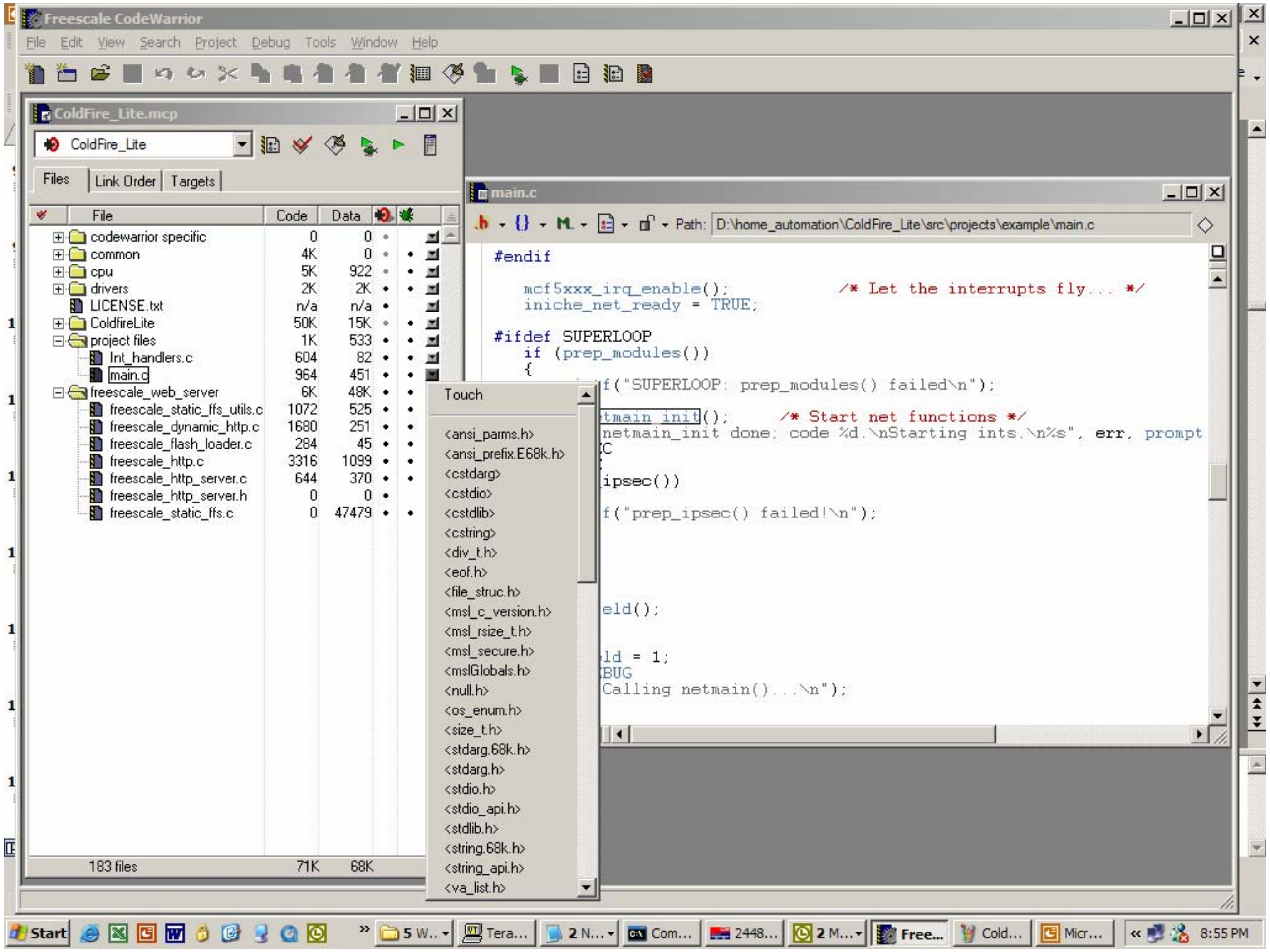
Touch

<ansi_parms.h>
<ansi_prefix.E68k.h>
<cstdarg>
<cstdio>
<cstdlib>
<cstring>
<div_t.h>
<eof.h>
<file_struc.h>
<msl_c_version.h>
<msl_rsize_t.h>
<msl_secure.h>
<mslGlobals.h>
<null.h>
<os_enum.h>
<size_t.h>
<stdarg.68k.h>
<stdarg.h>
<stdio.h>
<stdio_api.h>
<stdlib.h>
<string.68k.h>
<string_api.h>
<va_list.h>
ansi_prefix.CF.size.h
app_ping.h
common.h
in_utils.h
io.h
ipport.h
libport.h
m5223evb.h
mcf5223.h

```
enable();           /* Let the interrupts fly... */
eady = TRUE;

p
ules())

UPERLOOP: prep_modules() failed\n");

n_init();      /* Start net functions */
ain_init done; code %d.\nStarting ints.\n%s", err, prompt

c())

rep_ipsec() failed!\n");



);


 1;

ing netmain()...\n");
```

Slide 54

File Dependencies



File	Code	Data		
⊞ codewarrior specific	0	0	•	
⊞ common	4K	0	•	•
⊞ cpu	5K	922	•	•
⊞ drivers	2K	2K	•	•
LICENSE.txt	n/a	n/a	•	
⊞ ColdfireLite	50K	15K	•	•
⊟ project files	1K	533	•	•
Int_handlers.c	604	82	•	•
main.c	964	451	•	•
⊟ freescale_web_server	6K	48K	•	•
freescale_static_ffs_utils.c	1072	525	•	•
freescale_dynamic_http.c	1680	251	•	•
freescale_flash_loader.c	284	45	•	•
freescale_http.c	3316	1099	•	•
freescale_http_server.c	644	370	•	•
freescale_http_server.h	0	0	•	
freescale_static_ffs.c	0	47479	•	•
183 files	71K	68K		

Path: D:\home_automation\ColdFire_Lite\src\projects\example\main.c

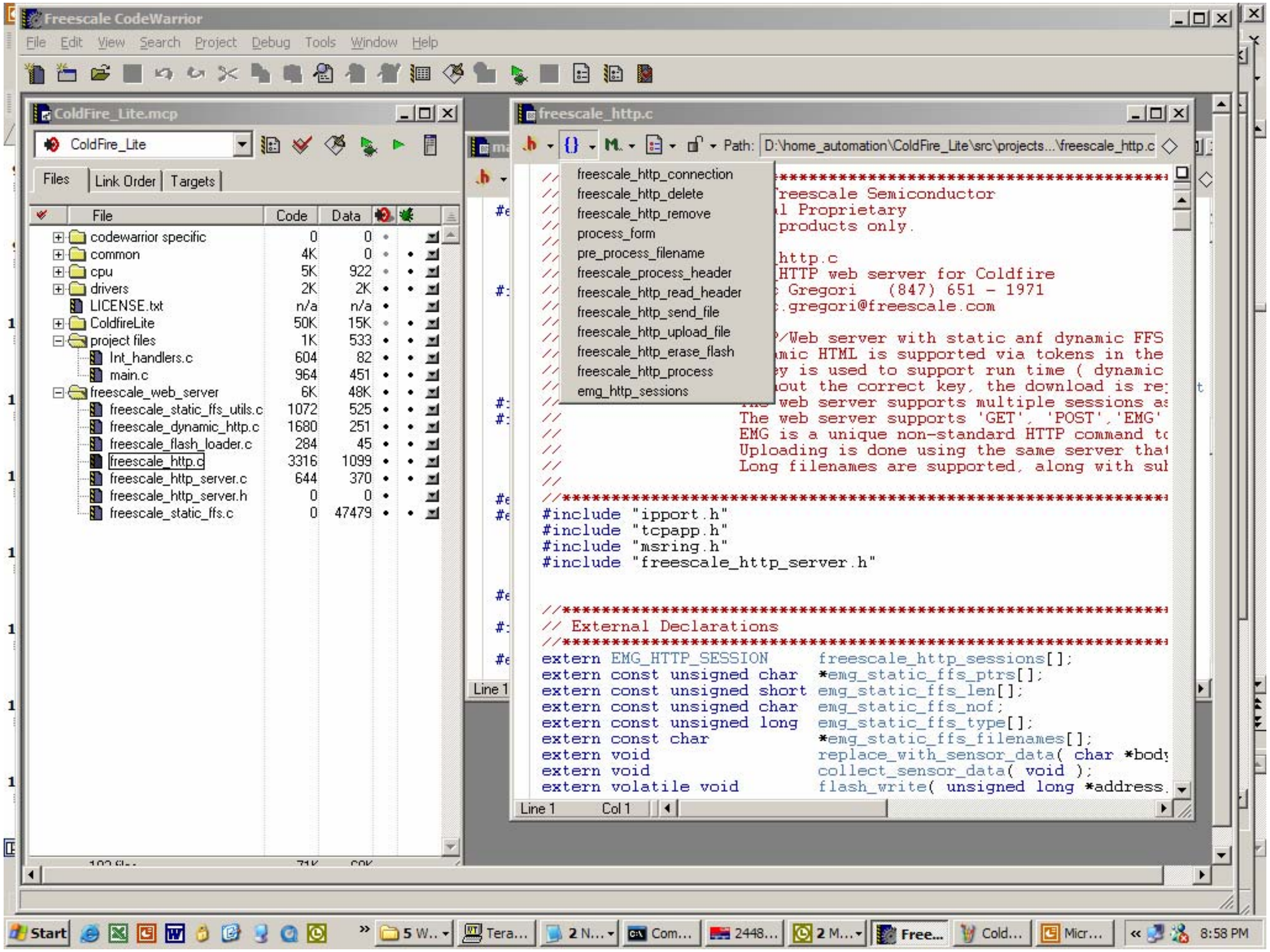
```
#endif

   mcf5xxx_irq_enable();          /* Let the interrupts fly... */
   iniche_net_ready = TRUE;

#ifdef SUPERLOOP
   if (prep_modules())
   {
          f("SUPERLOOP: prep_modules() failed\n");

          tmain_init();     /* Start net functions */
          netmain_init done: code %d.\nStarting ints.\n%s", err, prompt
          C

          ipsec())

          f("prep_ipsec() failed!\n");




          eld();


          ld = 1;
          BUG
          Calling netmain()...\n");
```

Find a Function in a File

- The Edit window also provides a quick access to all functions implemented inside a source file.
- Open the file **main.c** in the editor. Double click on the file name in the mcp window.
- Click on the arrow next to the {} icon. You see a list of all functions implemented in the current source file.
- Select **main** from the drop down list. The editor window scrolls down to the implementation of the function.

main

Functions in a file

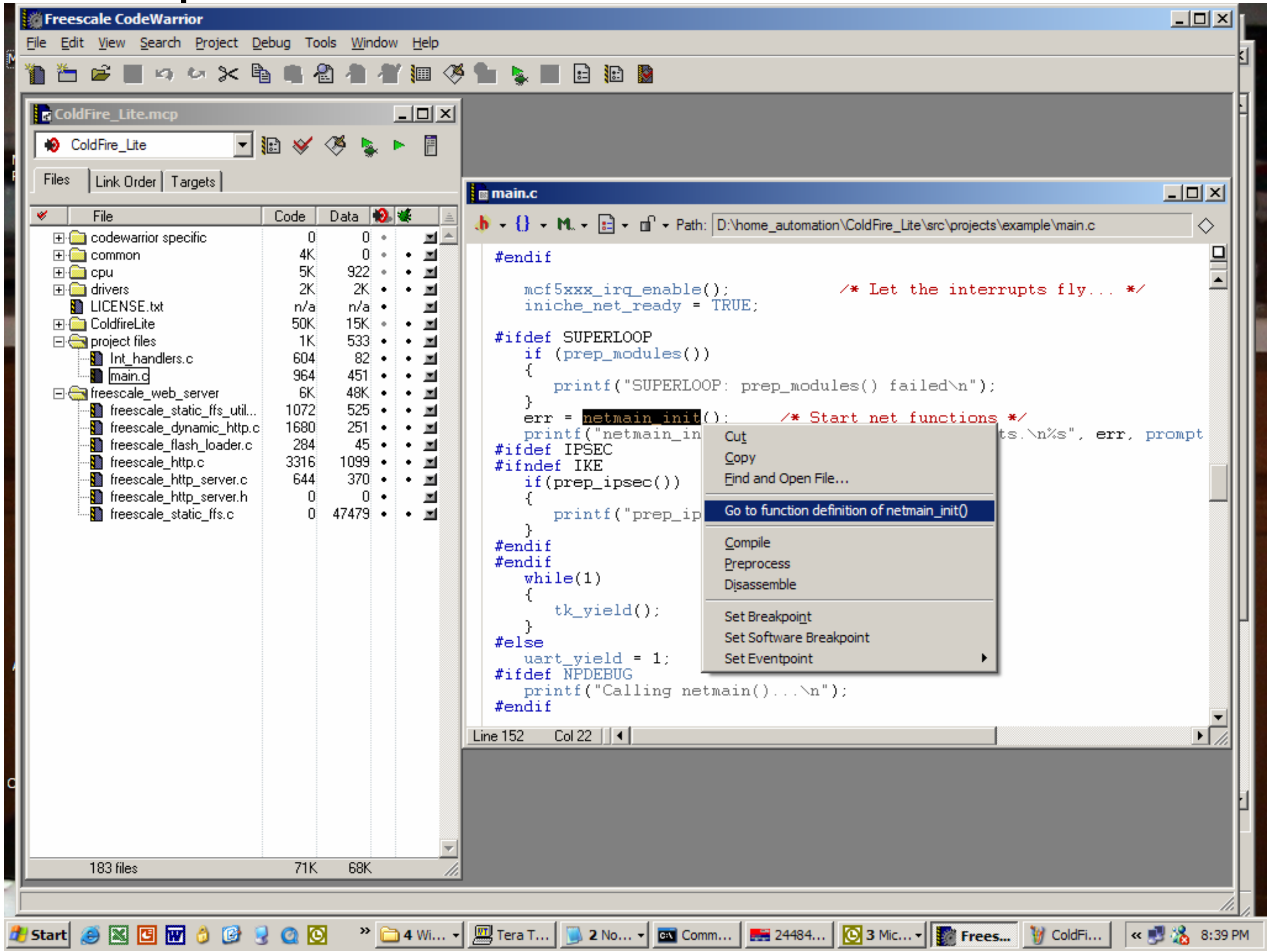


	File	Code	Data		
	codewarrior specific	0	0	●	
	common	4K	0	●	●
	cpu	5K	922	●	●
	drivers	2K	2K	●	●
	LICENSE.txt	n/a	n/a	●	
	ColdfireLite	50K	15K	●	●
	project files	1K	533	●	●
	Int_handlers.c	604	82	●	●
	main.c	964	451	●	●
	freescale_web_server	6K	48K	●	●
	freescale_static_ffs_utils.c	1072	525	●	●
	freescale_dynamic_http.c	1680	251	●	●
	freescale_flash_loader.c	284	45	●	●
	freescale_http.c	3316	1099	●	●
	freescale_http_server.c	644	370	●	●
	freescale_http_server.h	0	0	●	
	freescale_static_ffs.c	0	47479	●	●

```
//                       *****************************************
//                       reescale Semiconductor
//                       l Proprietary
//                       products only.
//
//                       http.c
//                       HTTP web server for Coldfire
//                        Gregori    (847) 651 - 1971
//                       .gregori@freescale.com
//
//                       /Web server with static anf dynamic FFS
//                       mic HTML is supported via tokens in the
//                       y is used to support run time ( dynamic
//                       out the correct key, the download is re
//                       web server supports multiple sessions a
//                       The web server supports 'GET', 'POST','EMG'
//                       EMG is a unique non-standard HTTP command t
//                       Uploading is done using the same server tha
//                       Long filenames are supported, along with su
//
//*******************************************************************
#include "ipport.h"
#include "tcpapp.h"
#include "msring.h"
#include "freescale_http_server.h"


//*******************************************************************
// External Declarations
//*******************************************************************
extern EMG_HTTP_SESSION      freescale_http_sessions[];
extern const unsigned char   *emg_static_ffs_ptrs[];
extern const unsigned short  emg_static_ffs_len[];
extern const unsigned char   emg_static_ffs_nof;
extern const unsigned long   emg_static_ffs_type[];
extern const char            *emg_static_ffs_filenames[];
extern void                  replace_with_sensor_data( char *bod
extern void                  collect_sensor_data( void );
extern volatile void         flash_write( unsigned long *address,
```

Source Code Browser

- Left clicking on a macro or function will highlight it.
- Right clicking will open a option box.



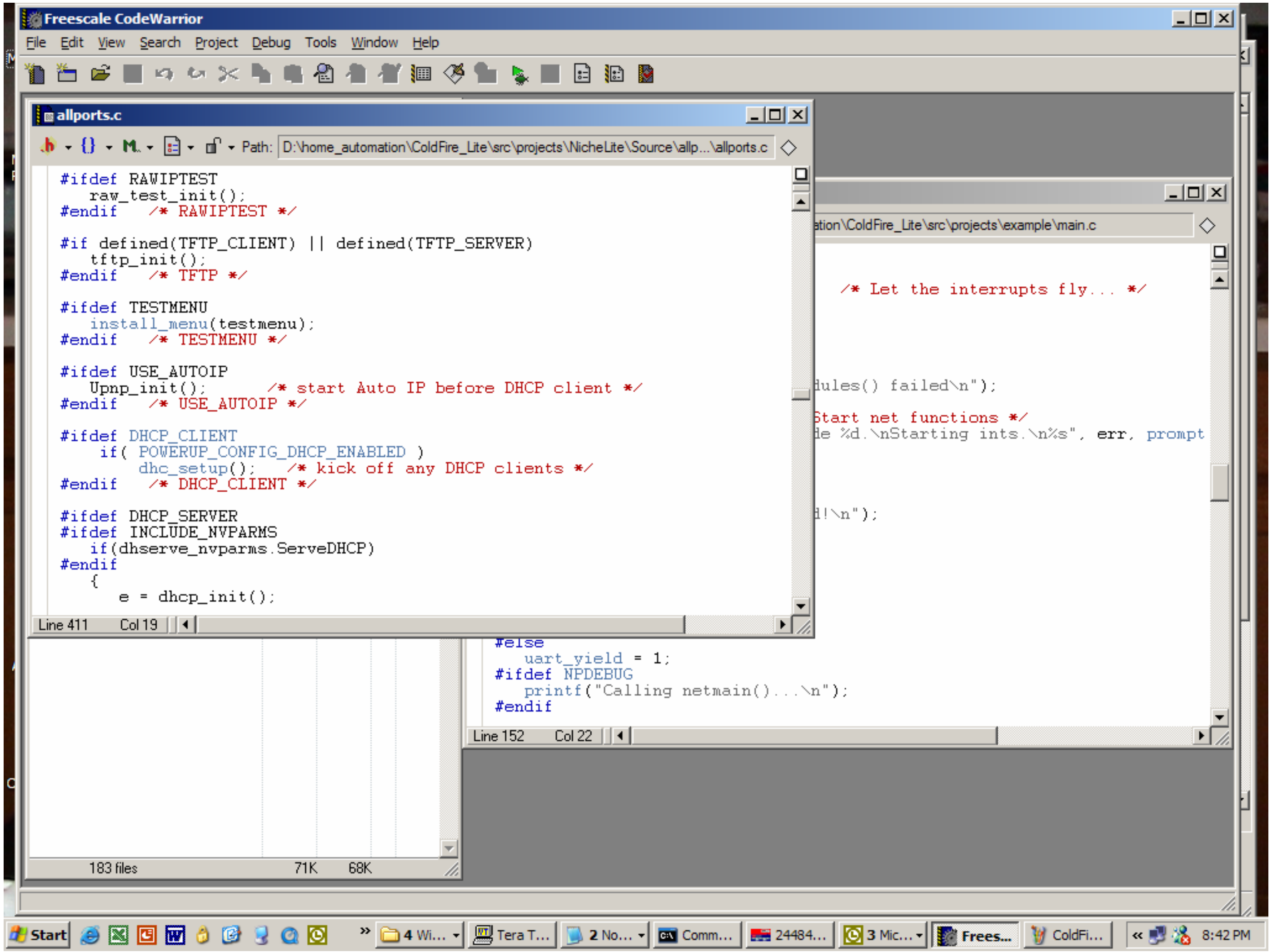
File	Code	Data
codewarrior specific	0	0
common	4K	0
cpu	5K	922
drivers	2K	2K
LICENSE.txt	n/a	n/a
ColdfireLite	50K	15K
project files	1K	533
Int_handlers.c	604	82
main.c	964	451
freescale_web_server	6K	48K
freescale_static_ffs_util...	1072	525
freescale_dynamic_http.c	1680	251
freescale_flash_loader.c	284	45
freescale_http.c	3316	1099
freescale_http_server.c	644	370
freescale_http_server.h	0	0
freescale_static_ffs.c	0	47479
183 files	71K	68K

Path: D:\home_automation\ColdFire_Lite\src\projects\example\main.c

```
#endif

   mcf5xxx_irq_enable();            /* Let the interrupts fly... */
   iniche_net_ready = TRUE;

#ifdef SUPERLOOP
   if (prep_modules())
   {
      printf("SUPERLOOP: prep_modules() failed\n");
   }
   err = netmain_init();       /* Start net functions */
   printf("netmain_in                          ts.\n%s", err, prompt
#ifdef IPSEC
#ifndef IKE
   if(prep_ipsec())
   {
      printf("prep_ip
   }
#endif
#endif
   while(1)
   {
      tk_yield();
   }
#else
   uart_yield = 1;
#ifdef NPDEBUG
   printf("Calling netmain()...\n");
#endif
```

Line 152 Col 22

Notice DHCP_CLIENT is blue

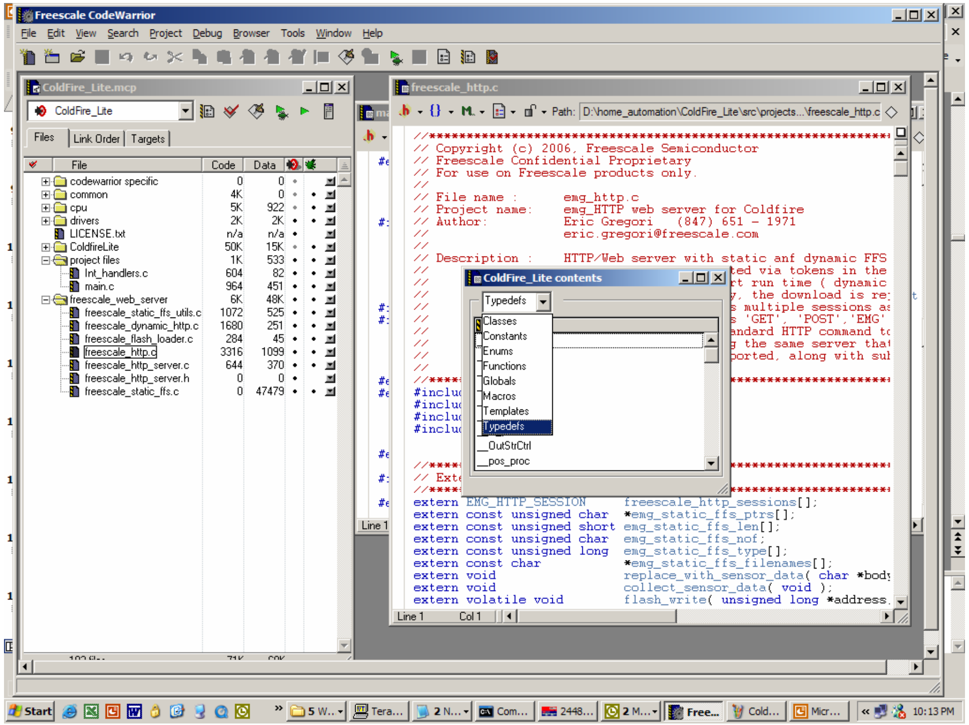

Freescale CodeWarrior
File Edit View Search Project Debug Tools Window Help
allports.c
Path: D:\home_automation\ColdFire_Lite\src\projects\NicheLite\Source\allp...\allports.c
Line 411 Col 19
ation\ColdFire_Lite\src\projects\example\main.c
Line 152 Col 22
183 files 71K 68K
Start
4 Wi...
Tera T...
2 No...
Comm...
24484...
3 Mic...
Frees...
ColdFi...
8:42 PM

```
#ifdef RAWIPTEST
   raw_test_init();
#endif   /* RAWIPTEST */

#if defined(TFTP_CLIENT) || defined(TFTP_SERVER)
   tftp_init();
#endif   /* TFTP */

#ifdef TESTMENU
   install_menu(testmenu);
#endif   /* TESTMENU */

#ifdef USE_AUTOIP
   Upnp_init();      /* start Auto IP before DHCP client */
#endif   /* USE_AUTOIP */

#ifdef DHCP_CLIENT
    if( POWERUP_CONFIG_DHCP_ENABLED )
        dhc_setup();   /* kick off any DHCP clients */
#endif   /* DHCP_CLIENT */

#ifdef DHCP_SERVER
#ifdef INCLUDE_NVPARMS
   if(dhserve_nvparms.ServeDHCP)
#endif
   {
      e = dhcp_init();
```

```
/* Let the interrupts fly... */

dules() failed\n");

Start net functions */
de %d.\nStarting ints.\n%s", err, prompt

d!\n");

#else
   uart_yield = 1;
#ifdef NPDEBUG
   printf("Calling netmain()...\n");
#endif
```

Browser Contents

- Choose **View > Browser** Contents.
- A window appears that displays the debug contents of the project.
- Play with the drop down list to see things like functions, globals, enums etc.
- Double clicking an item takes you to its definition.

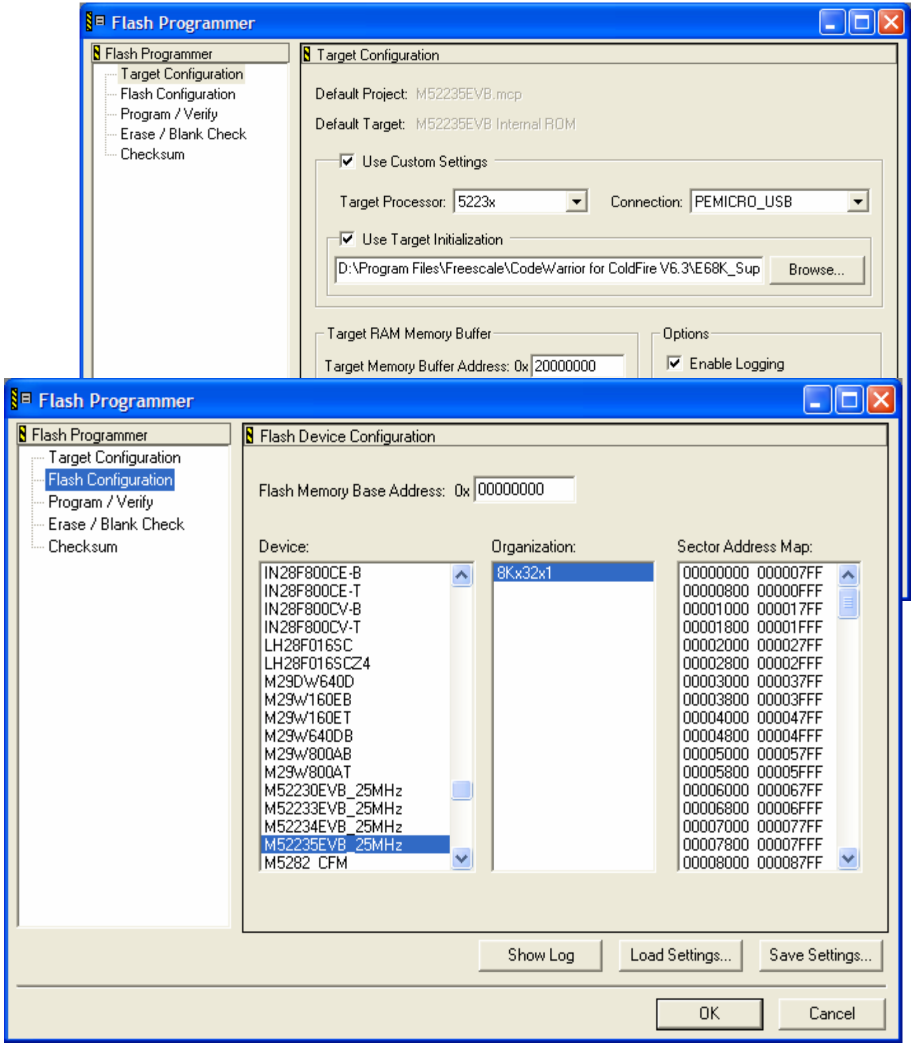


File	Code	Data		
codewarrior specific	0	0	•	
common	4K	0	•	•
cpu	5K	922	•	•
drivers	2K	2K	•	•
LICENSE.txt	n/a	n/a	•	
ColdfireLite	50K	15K	•	•
project files	1K	533	•	•
Int_handlers.c	604	82	•	•
main.c	964	451	•	•
freescale_web_server	6K	48K	•	•
freescale_static_ffs_utils.c	1072	525	•	•
freescale_dynamic_http.c	1680	251	•	•
freescale_flash_loader.c	284	45	•	•
freescale_http.c	3316	1099	•	•
freescale_http_server.c	644	370	•	•
freescale_http_server.h	0	0	•	
freescale_static_ffs.c	0	47479	•	•

```
//****************************************************************************
// Copyright (c) 2006, Freescale Semiconductor
// Freescale Confidential Proprietary
// For use on Freescale products only.
//
// File name :       emg_http.c
// Project name:     emg_HTTP web server for Coldfire
// Author:           Eric Gregori   (847) 651 - 1971
//                   eric.gregori@freescale.com
//
// Description :     HTTP/Web server with static anf dynamic FFS
//                                         ted via tokens in the
//                                         rt run time ( dynamic
//                                         y, the download is re
//                                         s multiple sessions a
//                                         s 'GET', 'POST','EMG'
//                                         andard HTTP command t
//                                         g the same server tha
//                                         ported, along with su
//
//****                                     ******************************
#incluc
#incluc
#incluc
#incluc


//****                                     ******************************
// Exte
//****                                     ******************************
extern EMG_HTTP_SESSION        freescale_http_sessions[];
extern const unsigned char     *emg_static_ffs_ptrs[];
extern const unsigned short    emg_static_ffs_len[];
extern const unsigned char     emg_static_ffs_nof;
extern const unsigned long     emg_static_ffs_type[];
extern const char              *emg_static_ffs_filenames[];
extern void                    replace_with_sensor_data( char *body
extern void                    collect_sensor_data( void );
extern volatile void           flash_write( unsigned long *address
```

Flash Programming Utility

- Open the Flash Programming Utility Tools > Flash Programmer.
- Click **Load Settings**
- Select the flash device configuration file **M52235EVB_25MHz.xml.**
- This script configures the Flash Programmer for the internal flash of the MCF52235 as can be seen on the **Flash Device Configuration** window.



Default Project: M52235EVB.mcp

Default Target: M52235EVB Internal ROM

☑ Use Custom Settings

Target Processor: 5223x

Connection: PEMICRO_USB

☑ Use Target Initialization

D:\Program Files\Freescale\CodeWarrior for ColdFire V6.3\E68K_Sup

Target RAM Memory Buffer

Target Memory Buffer Address: 0x 20000000

Options

☑ Enable Logging

Flash Memory Base Address: 0x 00000000

Device:

IN28F800CE-B
IN28F800CE-T
IN28F800CV-B
IN28F800CV-T
LH28F016SC
LH28F016SCZ4
M29DW640D
M29W160EB
M29W160ET
M29W640DB
M29W800AB
M29W800AT
M52230EVB_25MHz
M52233EVB_25MHz
M52234EVB_25MHz
M52235EVB_25MHz
M5282_CFM

Organization:

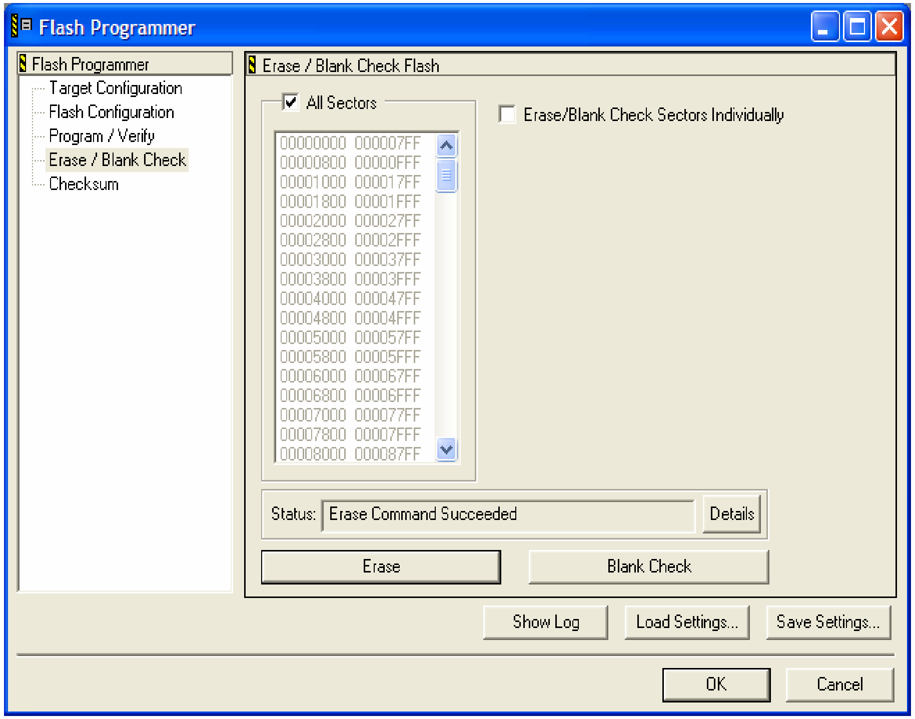
8Kx32x1

Sector Address Map:

```
00000000  000007FF
00000800  00000FFF
00001000  000017FF
00001800  00001FFF
00002000  000027FF
00002800  00002FFF
00003000  000037FF
00003800  00003FFF
00004000  000047FF
00004800  00004FFF
00005000  000057FF
00005800  00005FFF
00006000  000067FF
00006800  00006FFF
00007000  000077FF
00007800  00007FFF
00008000  000087FF
```

Erasing Flash

- Select **Erase / Blank Check.**
- Select All Sectors and uncheck **Erase/Blank Check Sectors Individually.**
- Hit **Erase** button.
- Status shows **Erase Command Succeeded.**
- Hit **Blank Check** Button.
- Status shows **Blank Check Completed Successfully.**



☑ All Sectors

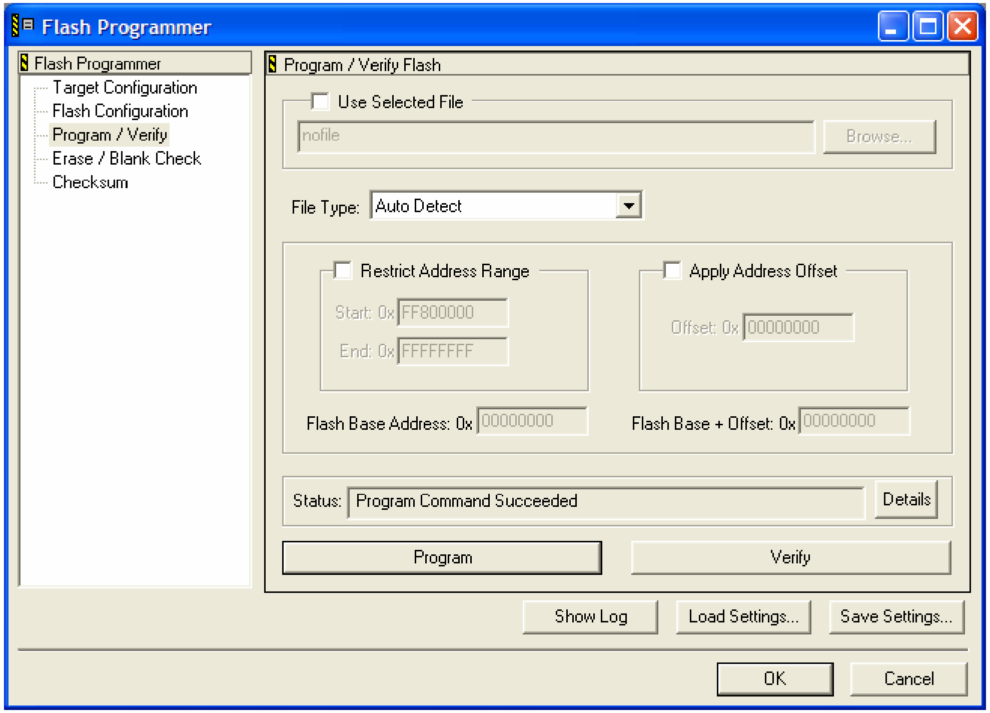
```
00000000  000007FF
00000800  00000FFF
00001000  000017FF
00001800  00001FFF
00002000  000027FF
00002800  00002FFF
00003000  000037FF
00003800  00003FFF
00004000  000047FF
00004800  00004FFF
00005000  000057FF
00005800  00005FFF
00006000  000067FF
00006800  00006FFF
00007000  000077FF
00007800  00007FFF
00008000  000087FF
```

☐ Erase/Blank Check Sectors Individually

Status: Erase Command Succeeded

Program Flash – Auto Detect File Type

- Select **Program / Verify.**
- Select **Program.**
- File type **Auto Detect** will automatically load correct file.
- Select **Verify.**
- Status line indicates **Verify Command Succeeded**.
- Power down the board with ON/OFF switch.
- Power up the board with ON/OFF switch.
- LEDS 1-4 will cycle.



□ Use Selected File

nofile

File Type: Auto Detect

□ Restrict Address Range

Start: 0x FF800000

End: 0x FFFFFFFF

□ Apply Address Offset

Offset: 0x 00000000

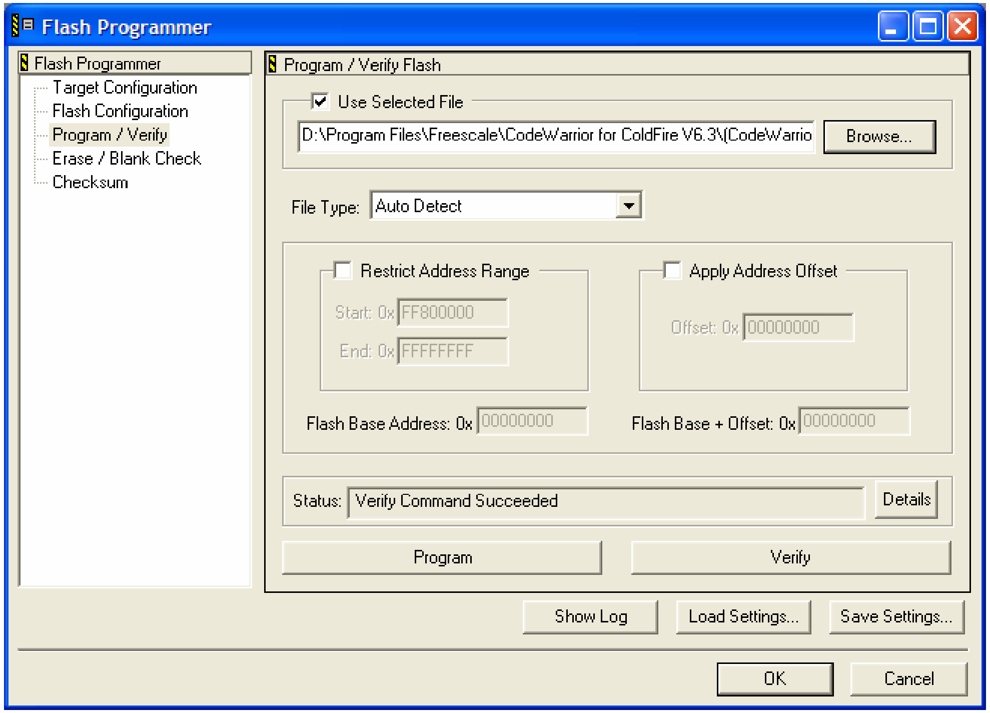
Flash Base Address: 0x 00000000

Flash Base + Offset: 0x 00000000

Status: Program Command Succeeded

Program Flash – Use Selected File

- Flash programming file can also be manually selected by checking the **Use Selected File** box.
- You can load symbolic debug information using the .elf file or S-record using .s19 file.



☑ Use Selected File

D:\Program Files\Freescale\CodeWarrior for ColdFire V6.3\(CodeWarrio

File Type: Auto Detect

☐ Restrict Address Range

Start: 0x FF800000

End: 0x FFFFFFFF

☐ Apply Address Offset

Offset: 0x 00000000

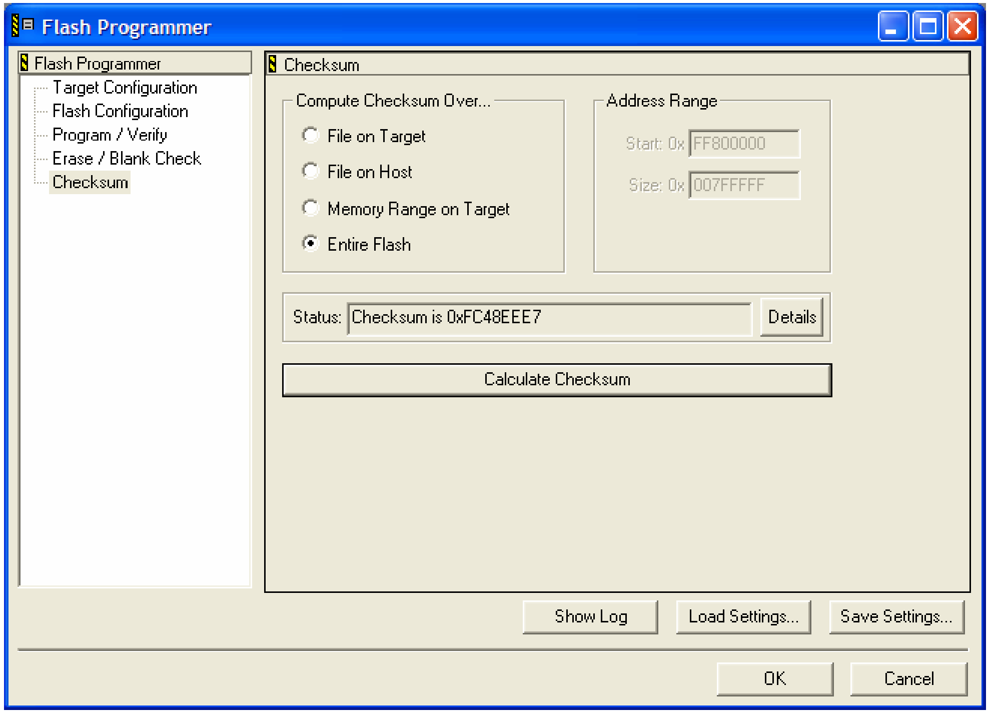
Flash Base Address: 0x 00000000

Flash Base + Offset: 0x 00000000

Status: Verify Command Succeeded

Calculate Checksum

- Select the **Checksum** panel.
- Select **Entire Flash.**
- Select **Calculate Checksum**
- Close the Flash programming utility.



Flash Programmer

Checksum

Compute Checksum Over...

○ File on Target
○ File on Host
○ Memory Range on Target
● Entire Flash

Address Range

Start: 0x FF800000

Size: 0x 007FFFFF

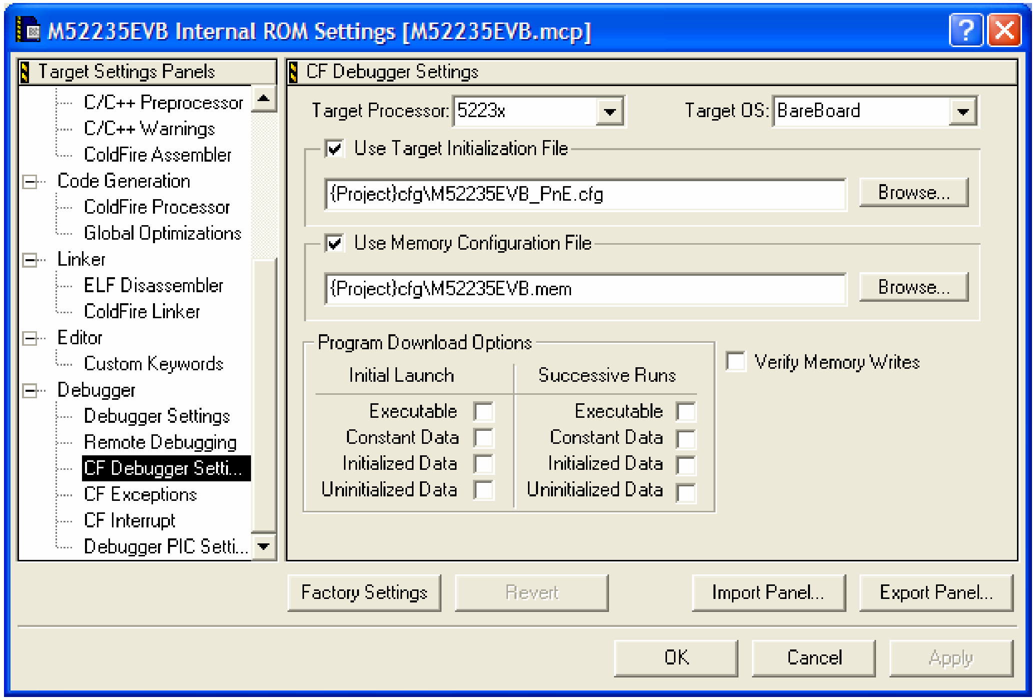
Status: Checksum is 0xFC48EEE7

Flash Programming Caution

- CodeWarrior® for ColdFire **DOES NOT** behave like the HC08 and HC12 tools when downloading code to internal FLASH.
- Code **MUST** be downloaded by the Flash programmer to internal Flash of the MCF52235 as described in the previous slides.
- Once code is programmed in Flash it can be debugged using the procedure in the following slides.

Debugging Flash Code

- In the left hand window select **CF Debugger Settings.**
- Note that **ALL** Check boxes for **Initial Launch** and **Successive Runs** are unchecked.
- This means that the debugger will NOT try to load code to Flash (because its already there) but will simply open the connection and make itself symbolically aware of the code.
- Close the window.
- Hit **Debug** **(F5)**
- Play around debugging flash code.
- Close the debugger.



CF Debugger Settings

Target Processor: 5223x

Target OS: BareBoard

☑ Use Target Initialization File

{Project}cfg\M52235EVB_PnE.cfg

☑ Use Memory Configuration File

{Project}cfg\M52235EVB.mem

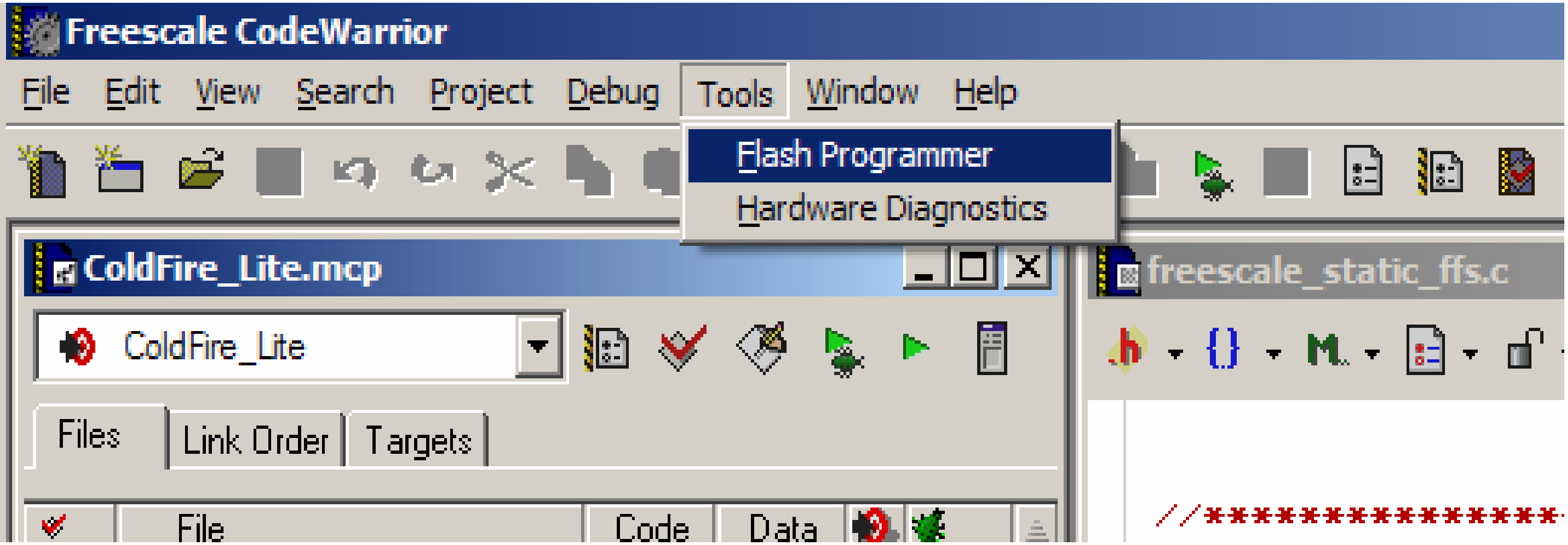
Program Download Options

Initial Launch		Successive Runs	
Executable	☐	Executable	☐
Constant Data	☐	Constant Data	☐
Initialized Data	☐	Initialized Data	☐
Uninitialized Data	☐	Uninitialized Data	☐

☐ Verify Memory Writes

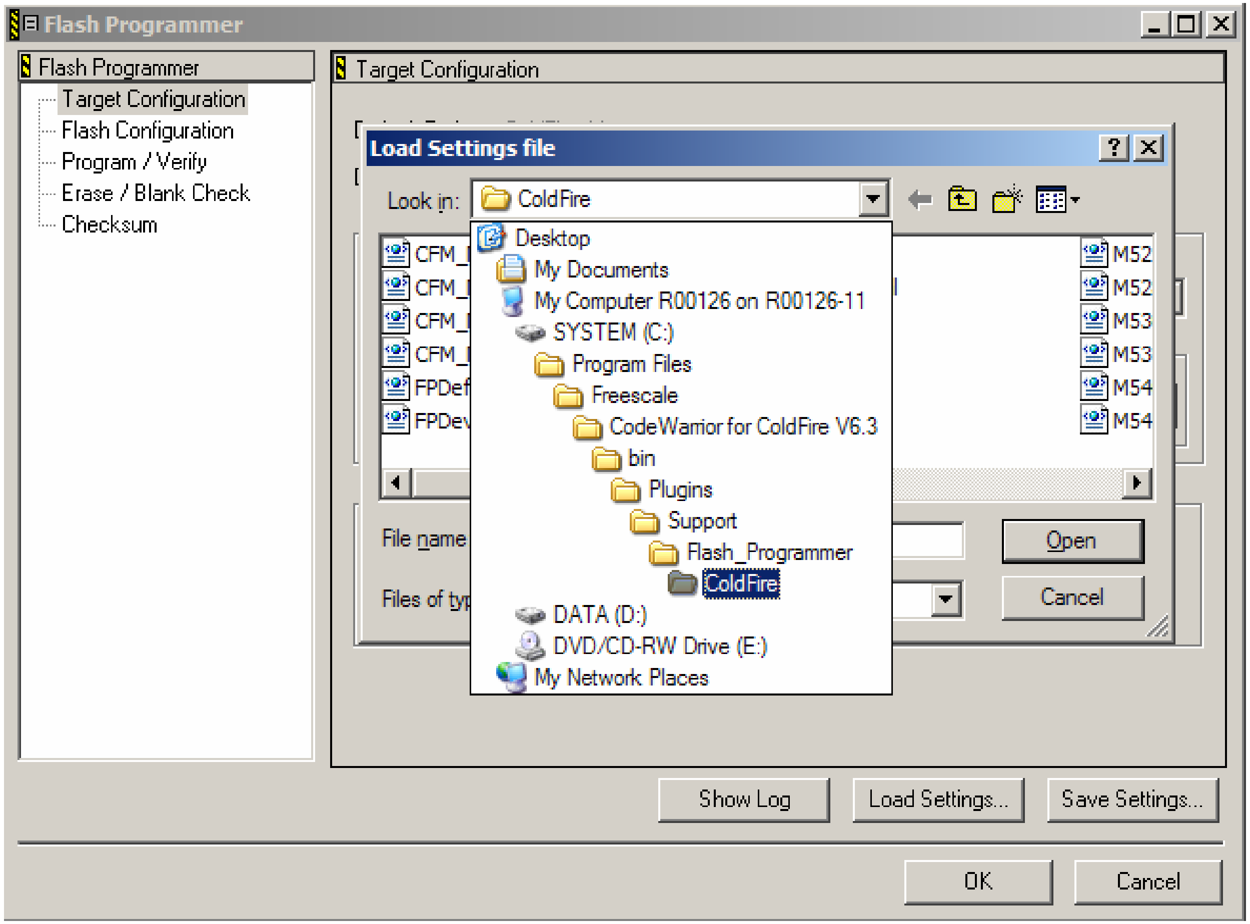
Starting the Flash Programmer

Start the Flash Programmer by selecting the tools Flash Programmer Pull Down



Browse the Directory

Click on the ‘Load Settings’ button, and browse for the directory shown below.

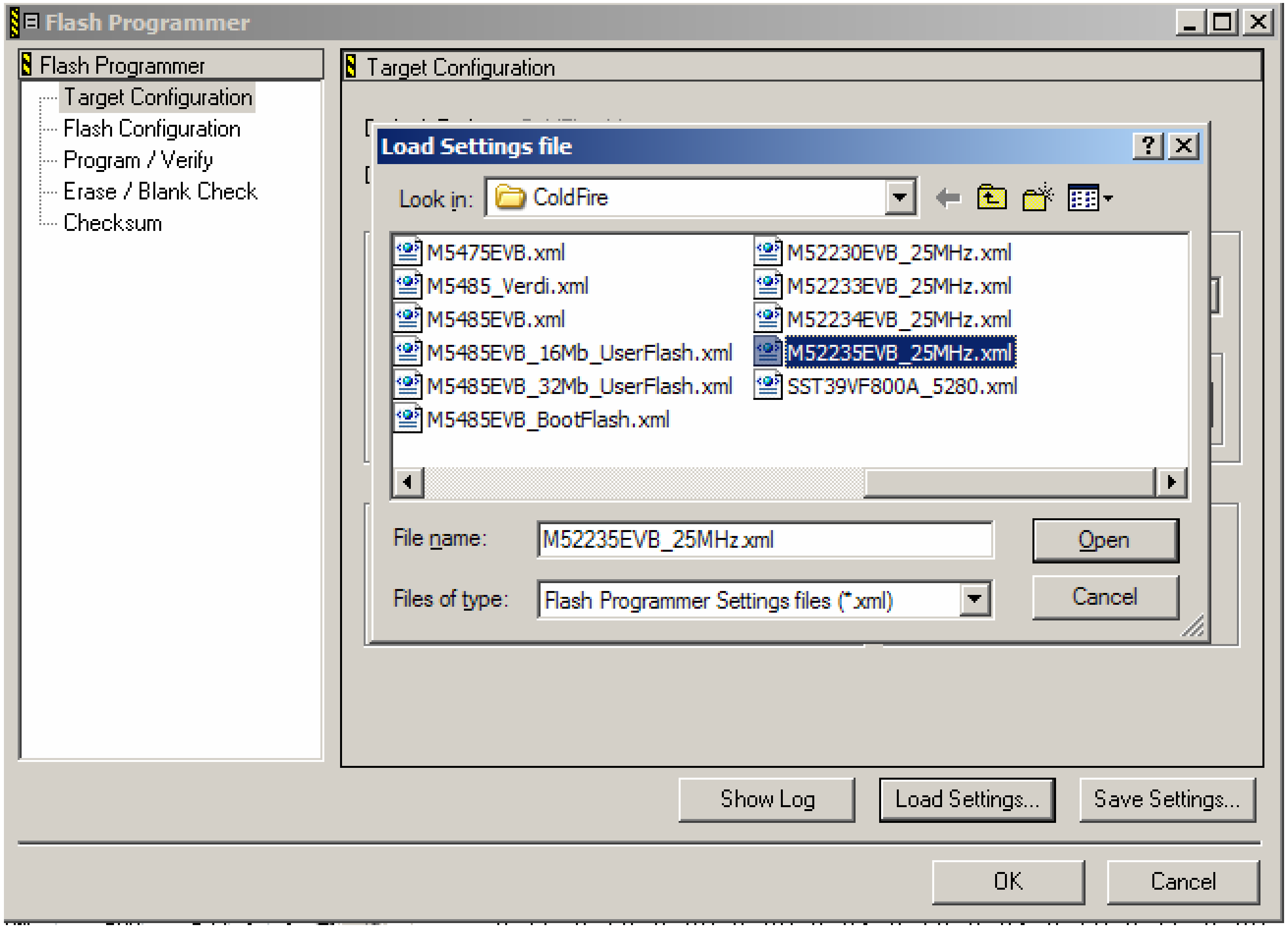


Look in: ColdFire

Desktop
My Documents
My Computer R00126 on R00126-11
SYSTEM (C:)
Program Files
Freescale
CodeWarrior for ColdFire V6.3
bin
Plugins
Support
Flash_Programmer
ColdFire
DATA (D:)
DVD/CD-RW Drive (E:)
My Network Places

Selecting the XML File

Select the M5223EVB-25MHZ xml file.



Load Settings file

Look in: ColdFire

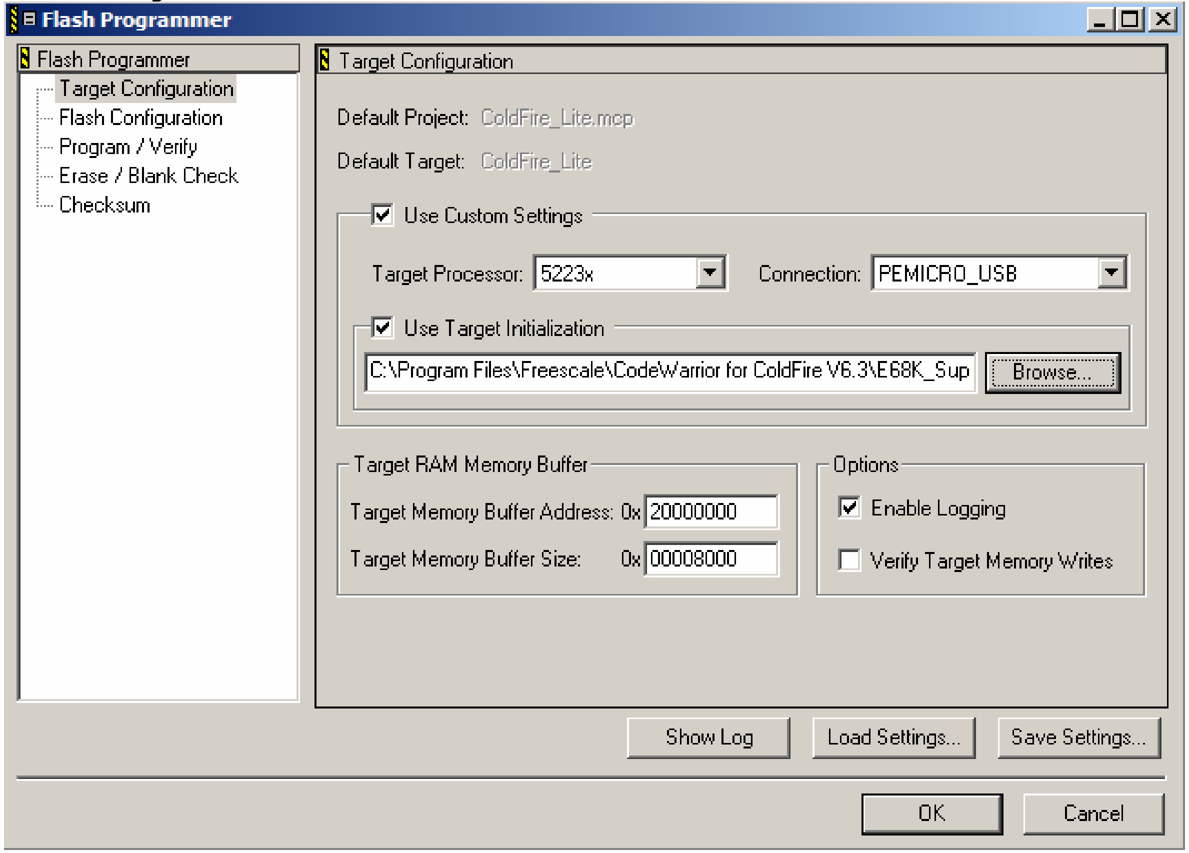
M5475EVB.xml
M5485_Verdi.xml
M5485EVB.xml
M5485EVB_16Mb_UserFlash.xml
M5485EVB_32Mb_UserFlash.xml
M5485EVB_BootFlash.xml
M52230EVB_25MHz.xml
M52233EVB_25MHz.xml
M52234EVB_25MHz.xml
M52235EVB_25MHz.xml
SST39VF800A_5280.xml

File name: M52235EVB_25MHz.xml

Files of type: Flash Programmer Settings files (*.xml)

Flash Programmer Screen

After Loading the XML file, the Flash Programmer will show following screen. Note the Target Processor, and RAM memory buffers are setup automatically from the XML file.



Default Project: ColdFire_Lite.mcp

Default Target: ColdFire_Lite

☑ Use Custom Settings

Target Processor: 5223x

Connection: PEMICRO_USB

☑ Use Target Initialization

C:\Program Files\Freescale\CodeWarrior for ColdFire V6.3\E68K_Sup

Target RAM Memory Buffer

Target Memory Buffer Address: 0x 20000000

Target Memory Buffer Size: 0x 00008000

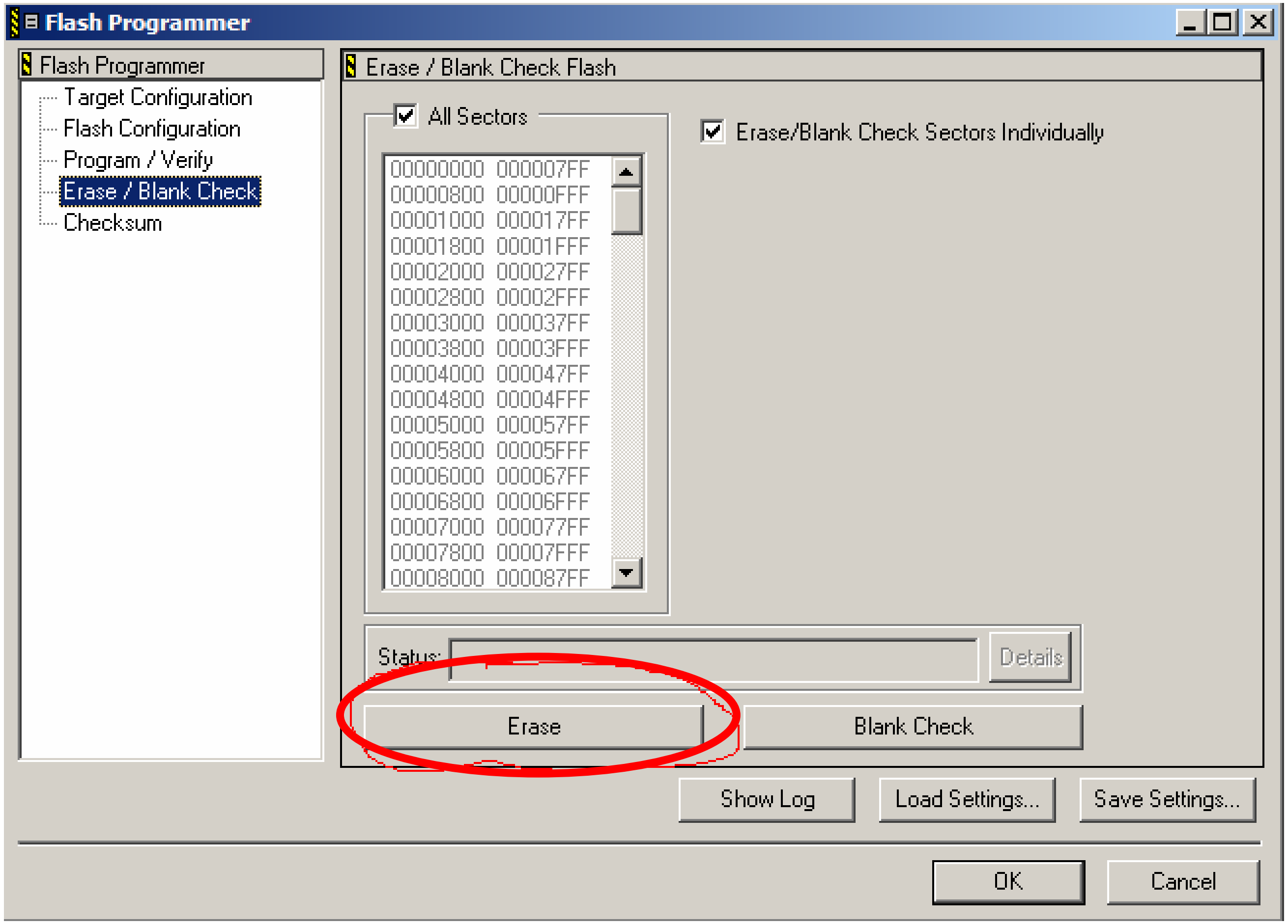
Options

☑ Enable Logging

□ Verify Target Memory Writes

Erase Flash

Erase the Flash by selecting Erase/Blank Check, and clicking the Erase button. Watch the Status window for errors.



☑ All Sectors

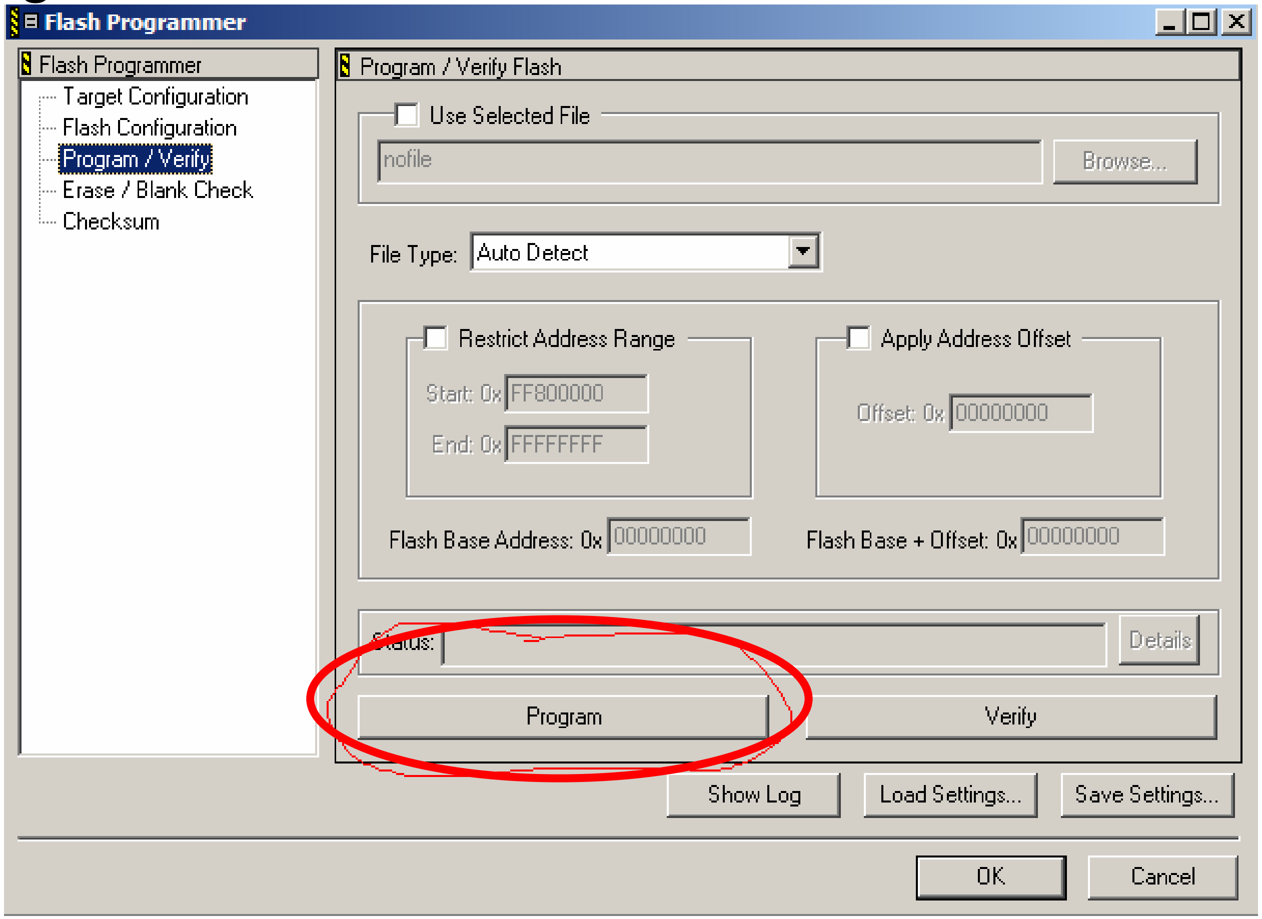
☑ Erase/Blank Check Sectors Individually

00000000	000007FF
00000800	00000FFF
00001000	000017FF
00001800	00001FFF
00002000	000027FF
00002800	00002FFF
00003000	000037FF
00003800	00003FFF
00004000	000047FF
00004800	00004FFF
00005000	000057FF
00005800	00005FFF
00006000	000067FF
00006800	00006FFF
00007000	000077FF
00007800	00007FFF
00008000	000087FF

Status:

Programming the Flash

After the Erase is Complete, go to the Program/Verify window and click on the Program button.



Flash Programmer

Program / Verify Flash

☐ Use Selected File

nofile

File Type: Auto Detect

☐ Restrict Address Range

Start: 0x FF800000

End: 0x FFFFFFFF

☐ Apply Address Offset

Offset: 0x 00000000

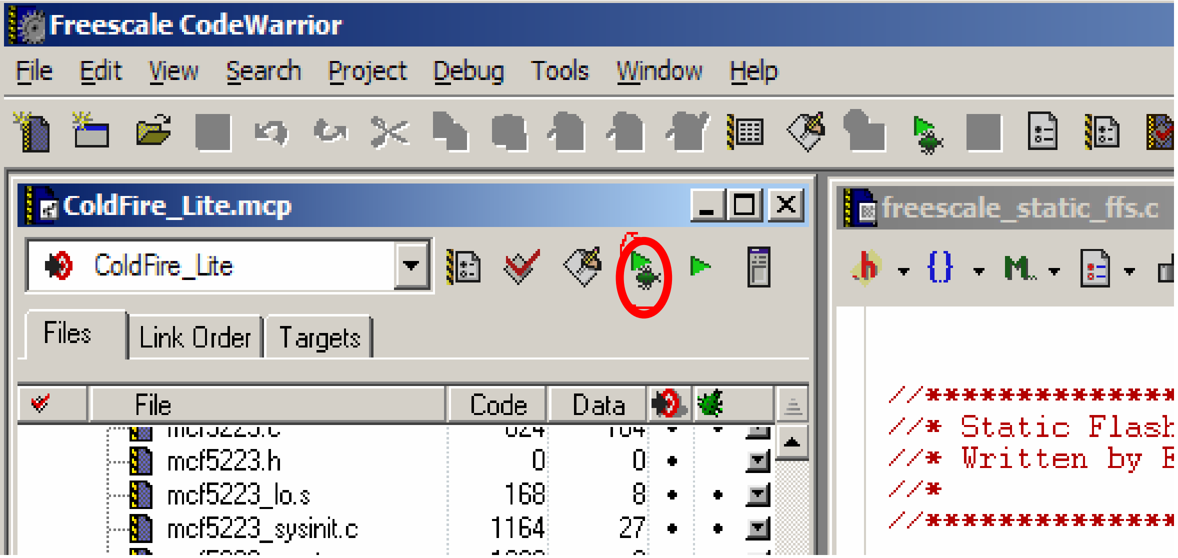
Flash Base Address: 0x 00000000

Flash Base + Offset: 0x 00000000

Status:

RUN

Click on the Run icon, circled in RED below. This will execute the code in flash. If you have an external power supply, you could also disconnect the USB from the board and hit reset.

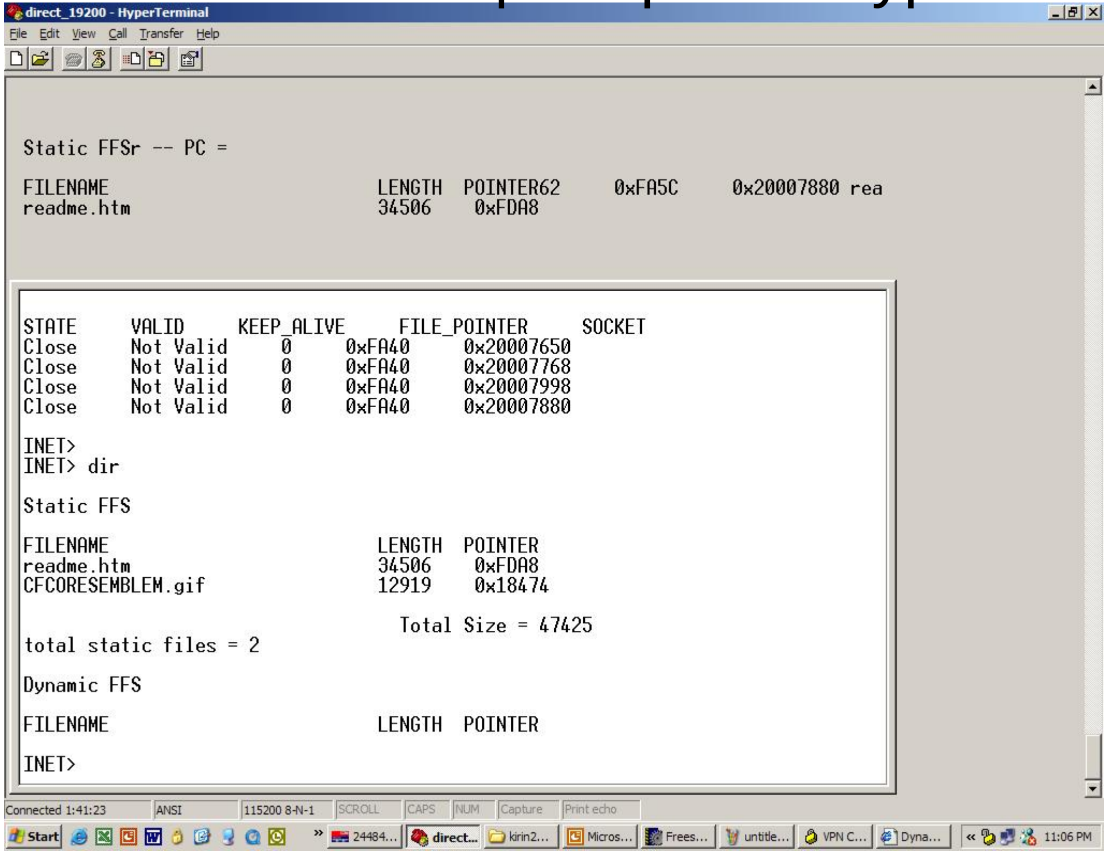


	File	Code	Data
	mcf5223.h	0	0
	mcf5223_lo.s	168	8
	mcf5223_sysinit.c	1164	27

```
//***************
//* Static Flash
//* Written by E
//*
//***************
```

Connecting the Serial Port

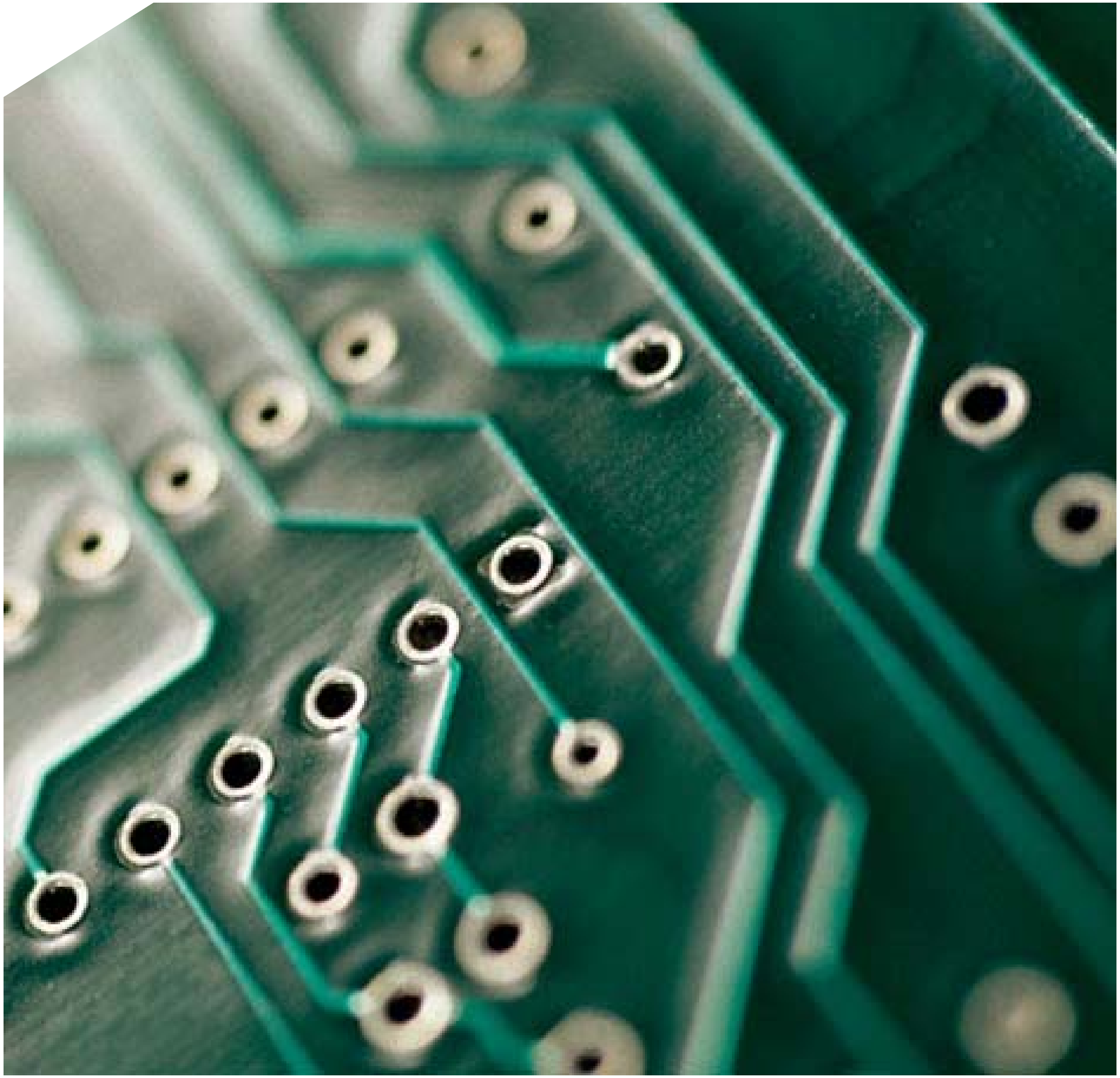
Connect the serial port on the demo board to the PC. Then open hyperterminal and configure for 115Kbaud, 8, n, 1, no flow control. Hit enter until you see the ‘INET>’ prompt then type ‘dir’.



direct_19200 - HyperTerminal
File Edit View Call Transfer Help

```
Static FFSr -- PC =

FILENAME                          LENGTH  POINTER62     0xFA5C     0x20007880 rea
readme.htm                        34506    0xFDA8
```

```
STATE     VALID     KEEP_ALIVE     FILE_POINTER     SOCKET
Close     Not Valid     0      0xFA40     0x20007650
Close     Not Valid     0      0xFA40     0x20007768
Close     Not Valid     0      0xFA40     0x20007998
Close     Not Valid     0      0xFA40     0x20007880

INET>
INET> dir

Static FFS

FILENAME                          LENGTH  POINTER
readme.htm                        34506    0xFDA8
CFCORESEMBLEM.gif                 12919    0x18474

                                    Total Size = 47425
total static files = 2

Dynamic FFS

FILENAME                          LENGTH  POINTER

INET>
```



HTTP/HTML/AJAX Overview

(and the *ColdFire*_TCP/IP_Lite)

ColdFire_TCP/IP_Lite Stack

The ***ColdFire***_TCP/IP_Lite stack includes:

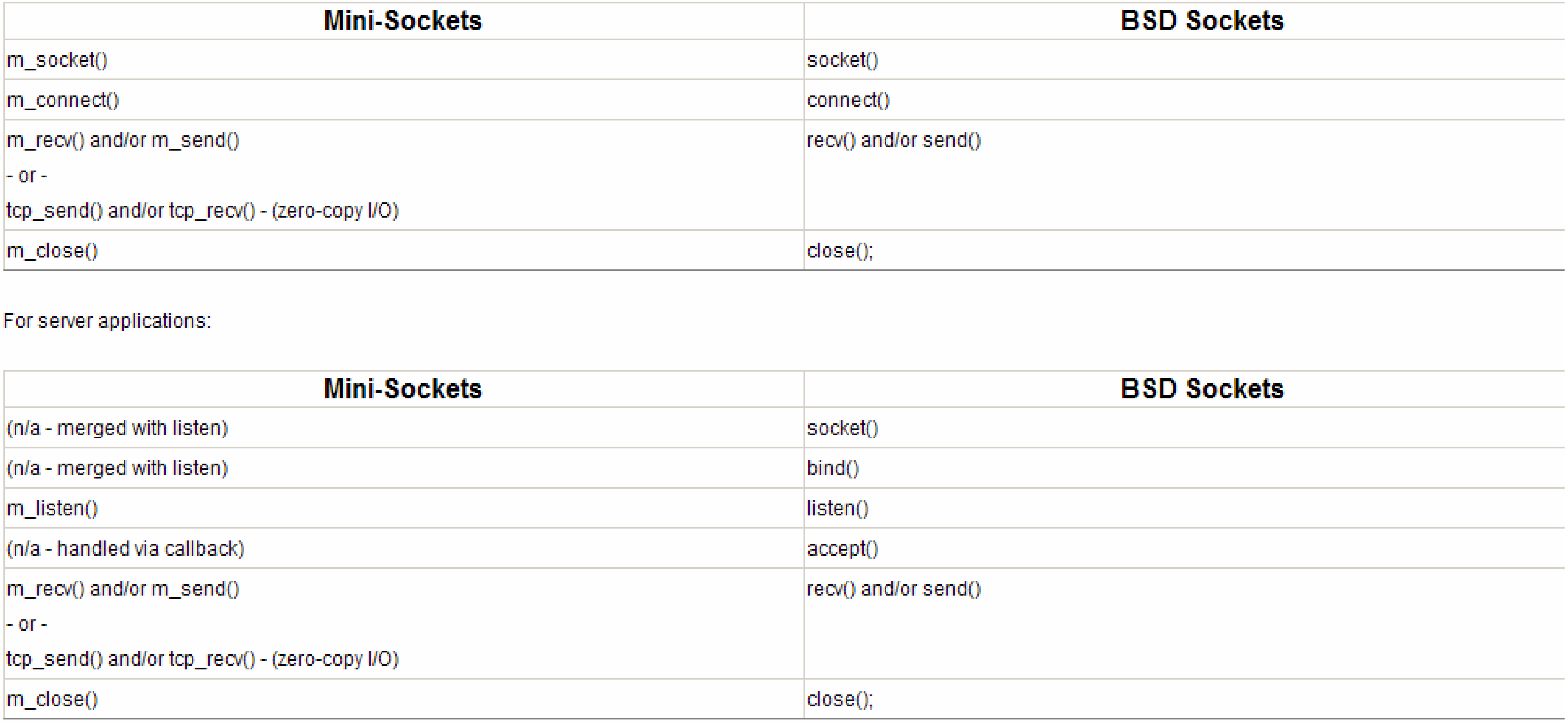
- A Mini-Sockets TCP API.
- A TFTP (Trivial File Transfer protocol) server.
- A DHCP (Dynamic Host Configuration protocol) client.
- Zero-copy sockets for performance.
- Less then 40K of program space.

Mini-Sockets TCP API

The mini-Sockets API is designed to be as close as possible to the BSD Sockets API and still allow a small footprint. The primary differences are that passive connections are accomplished with a single call, m_listen(), rather than the BSD bind()-listen()-accept() sequence, and the BSD select() call is replaced with a callback mechanism.

BSD = **B**erkeley **S**oftware **D**istribution

Mini-Socket Interface Compared to BSD Sockets



Mini-Sockets	BSD Sockets
m_socket()	socket()
m_connect()	connect()

m_recv() and/or m_send()
- or -
tcp_send() and/or tcp_recv() - (zero-copy I/O)
	recv() and/or send()

m_close()	close();

For server applications:

Mini-Sockets	BSD Sockets
(n/a - merged with listen)	socket()
(n/a - merged with listen)	bind()
m_listen()	listen()
(n/a - handled via callback)	accept()

m_recv() and/or m_send()
- or -
tcp_send() and/or tcp_recv() - (zero-copy I/O)
	recv() and/or send()

m_close()	close();

A Simple Server Using Mini-Sockets

- **Creating a Listening Socket**

```
// Init a socket structure with our Port Number
emg_http_sin.sin_addr.s_addr                = (INADDR_ANY);
emg_http_sin.sin_port                       = (PORT_NUMBER);

 emg_http_server_socket   = m_listen(&emg_http_sin, freescale_http_cmdcb, &e);
```

- **Accepting a Connection**

```
switch(code)
{
        // socket open complete
        case M_OPENOK:
                msring_add(&emg_http_msring, so);
                break;
}
```

- **Receiving TCP data**

```
length = m_recv( freescale_http_sessions[session].socket, (char *)buffer, RECV_BUFFER_SIZE );
```

- **Sending TCP data**

```
bytes_sent = m_send( freescale_http_sessions[session].socket, data, length );
```

- **Closing the Socket**

```
j = m_close( so );
```

A Simple Client Using Mini-Sockets

- **Creating a Socket**

M_SOCK Socket = m_socket();

- **Connecting to a Server**

int m_connect(M_SOCK socket, struct sockaddr_in * sin, M_CALLBACK(name));
// m_connect is blocking until a connection completes.
// If the socket is configured for non-blocking, then the callback funtion is used to indicate when the connection is established.

- **Receiving TCP data**

length = m_recv(freescale_http_sessions[session].socket, (char *)buffer, RECV_BUFFER_SIZE);

- **Sending TCP data**

bytes_sent = m_send(freescale_http_sessions[session].socket, data, length);

- **Closing the Socket**

j = m_close(so);

The Included RTOS

*ColdFire*_TCP/IP_Lite includes a very nice RTOS

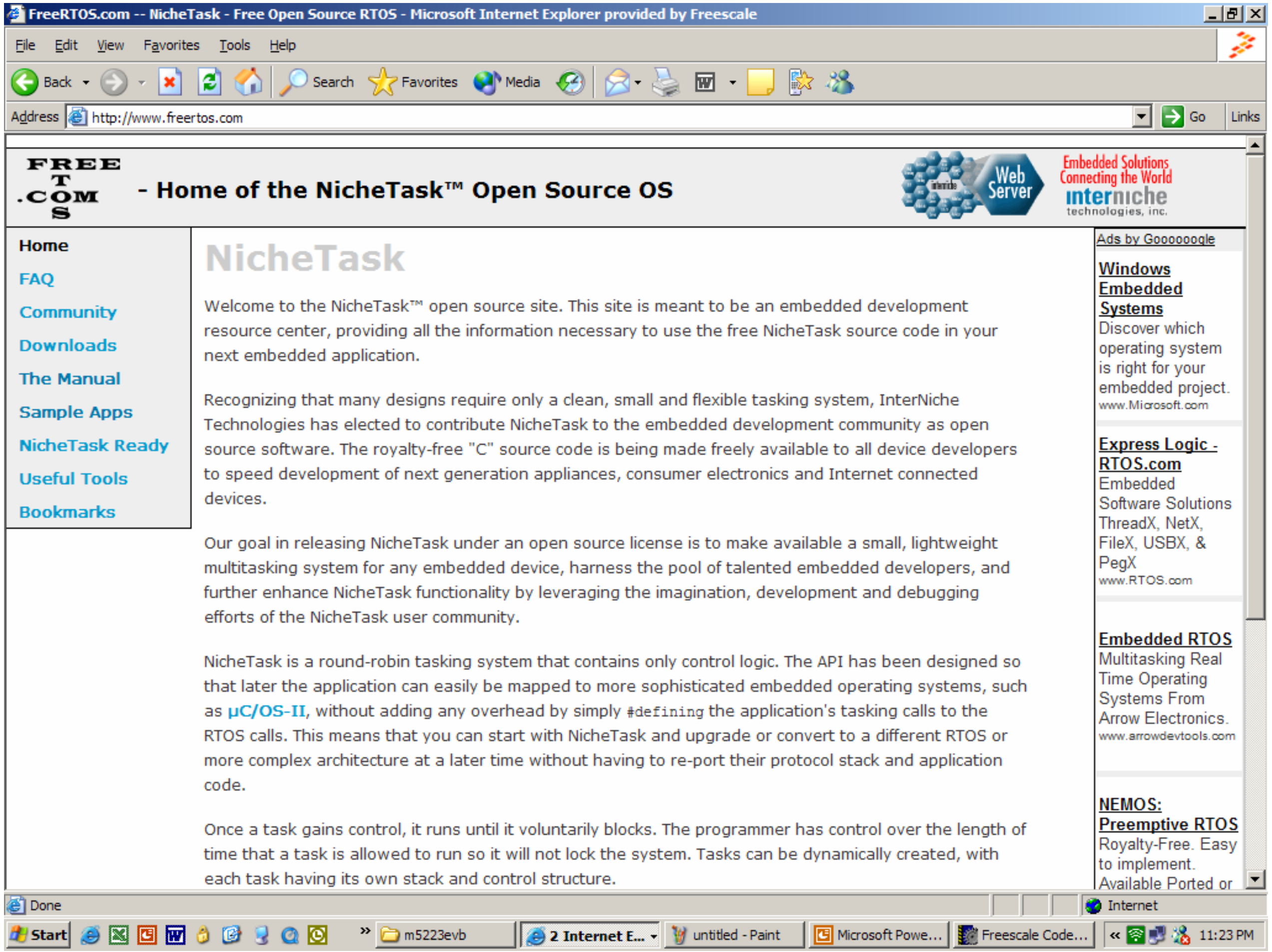
- Tasks are supported with message rings, and separate stacks.
- Priorities are supported.
- Task Sleeping support.
- Tasks can sleep waiting on an event.

RTOS Functions

```
// entry points to tasking system
task *   tk_init(stack_t * base, int st_size);          // Init the RTOS
task *   tk_new(task*, int(*)(int), int, char*, int);   // fork a new task
void     tk_block(void);                                // switch to next runnable task
void     tk_exit(void);                                 // kill & delete current task
void     tk_kill(task * tk_to_die);                     // mark any task for death
void     tk_wake(task * tk);                            // mark a task to run
void     tk_sleep(long ticks);                          // sleep for number of ticks
void     tk_ev_block(void * event);                     // block until event occurs
void     tk_ev_wake(void * event);                      // wake tasks waiting for event
```

For More Information on the RTOS

http://www.freertos.com



FREE RTOS .COM - Home of the NicheTask™ Open Source OS

Web Server

Embedded Solutions Connecting the World
interniche technologies, inc.

Home
FAQ
Community
Downloads
The Manual
Sample Apps
NicheTask Ready
Useful Tools
Bookmarks

NicheTask

Welcome to the NicheTask™ open source site. This site is meant to be an embedded development resource center, providing all the information necessary to use the free NicheTask source code in your next embedded application.

Recognizing that many designs require only a clean, small and flexible tasking system, InterNiche Technologies has elected to contribute NicheTask to the embedded development community as open source software. The royalty-free "C" source code is being made freely available to all device developers to speed development of next generation appliances, consumer electronics and Internet connected devices.

Our goal in releasing NicheTask under an open source license is to make available a small, lightweight multitasking system for any embedded device, harness the pool of talented embedded developers, and further enhance NicheTask functionality by leveraging the imagination, development and debugging efforts of the NicheTask user community.

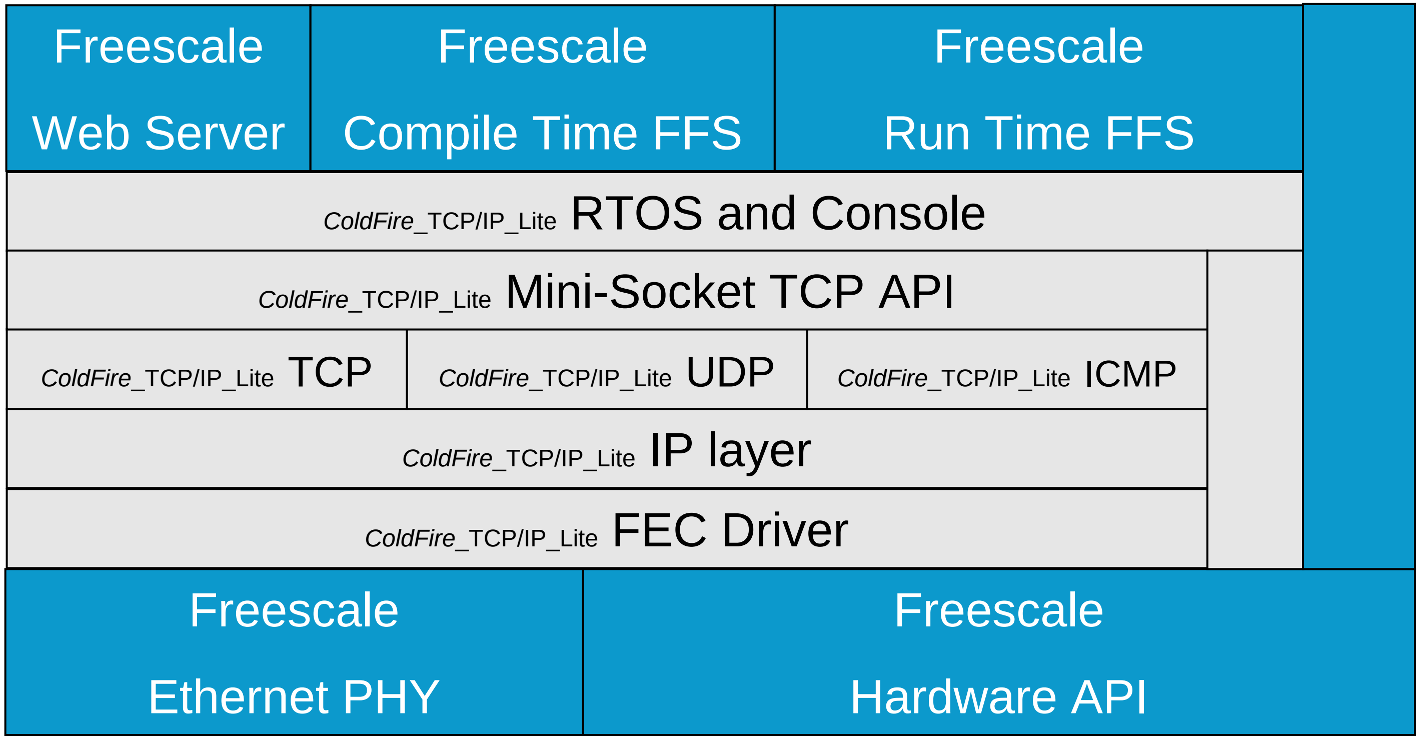
NicheTask is a round-robin tasking system that contains only control logic. The API has been designed so that later the application can easily be mapped to more sophisticated embedded operating systems, such as μC/OS-II, without adding any overhead by simply `#defining` the application's tasking calls to the RTOS calls. This means that you can start with NicheTask and upgrade or convert to a different RTOS or more complex architecture at a later time without having to re-port their protocol stack and application code.

Once a task gains control, it runs until it voluntarily blocks. The programmer has control over the length of time that a task is allowed to run so it will not lock the system. Tasks can be dynamically created, with each task having its own stack and control structure.

Freescale Web Server

- HTTP1.0 compliant server with connection persistance and multiple sessions (HTTP1.1 will be available in future revisions).
- GET and POST elements supported.
- Dynamic HTML support with replace and conditional tokens.
- Serial interface support for Dynamic HTML variables.
- Provides run time and compile time flash file systems.
- Long file name support with subdirectories.
- ‘DIR’ command supported on serial interface.
- PC utilities for compressing compile time and run time downloadable images of multi-page web pages.
- PC utility for downloading run time downloadable web page image through port 80 (to get through firewalls).
- 32 byte ascii key for web page download security.
- It’s Free for use on *ColdFire*® processors!!!

Software Model



Freescale Web Server

Freescale Compile Time FFS

Freescale Run Time FFS

*ColdFire*_TCP/IP_Lite RTOS and Console

*ColdFire*_TCP/IP_Lite Mini-Socket TCP API

*ColdFire*_TCP/IP_Lite TCP

*ColdFire*_TCP/IP_Lite UDP

*ColdFire*_TCP/IP_Lite ICMP

*ColdFire*_TCP/IP_Lite IP layer

*ColdFire*_TCP/IP_Lite FEC Driver

Freescale Ethernet PHY

Freescale Hardware API

FFS = Flash File System

Web Servers

- Web Servers implement the HyperText Transfer Protocol (HTTP) to send web pages from a server to a client.
- The Web Server contains the content, the Web Browser Displays the content.
- For these labs, the Web Browser used will be the Internet Explorer.

HTTP - An Overview

- HTTP – HyperText Transport Protocol.
- HTTP – Is used to transfer HTML/Web Pages on the web.
- From RFC1945:

  The HTTP protocol is based on a request/response paradigm. A client establishes a connection with a server and sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content. The server responds with a status line, including the messages protocol version and a success or error code, followed by a MIME-like message containing server information, entity metainformation, and possible body content.

- Generally HTTP uses TCP/IP port 80.

- There are two versions of HTTP, 1.0 and 1.1.

- HTTP1.0 is defined by RFC1945.

HTTP Protocol Exchange

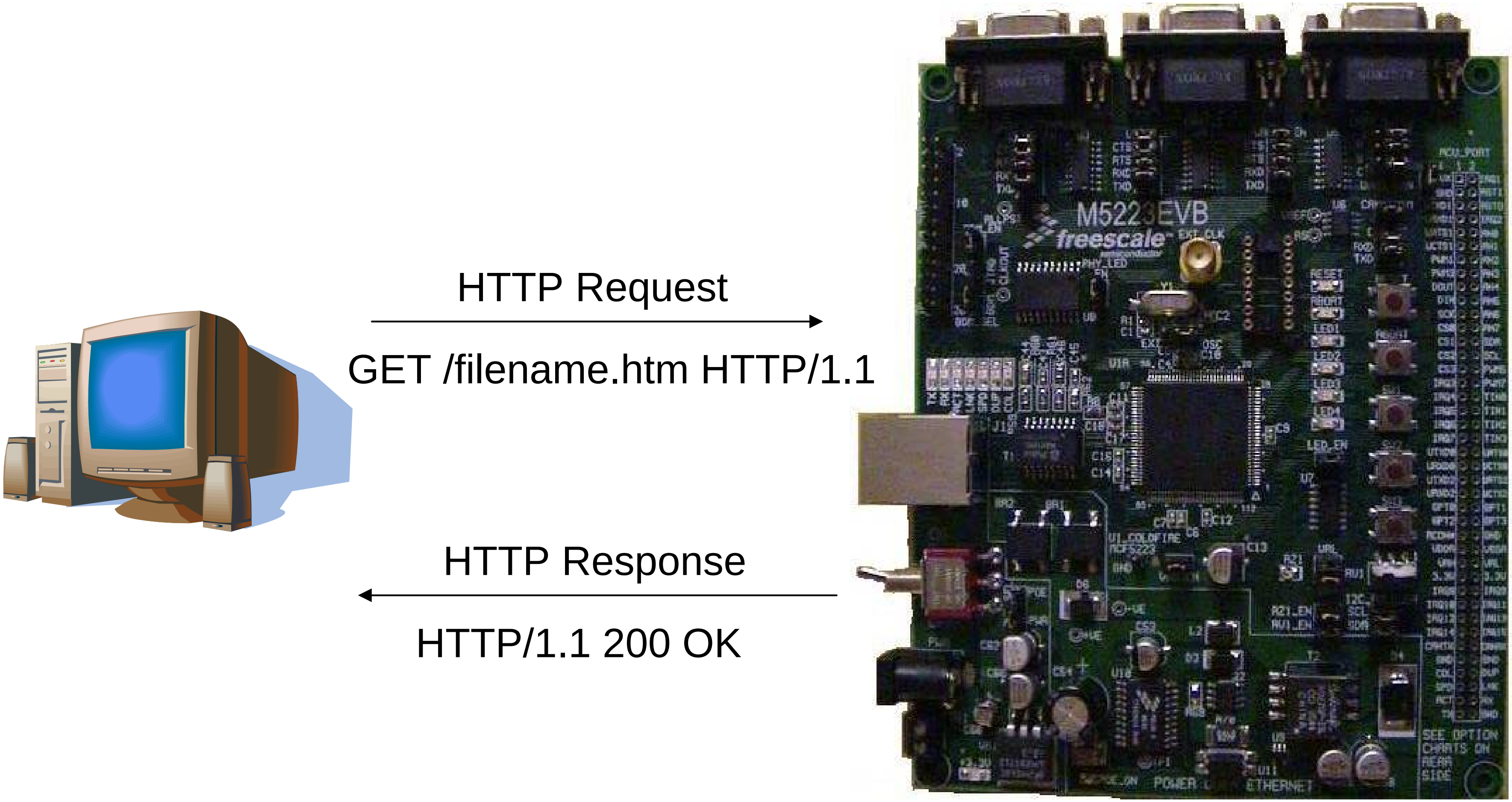
The client starts an exchange using one of two Methods:

- GET method – Request the server to send a file
- POST method – Sends a file to the server
  - The method is followed by a list of Request Header Fields

The server responds with a response message:

- The first line of the message is the status line.
  - Sample Status line HTTP/1.0 200 OK
    - > Status code 2xx means success
    - > Status code 4xx means error
- The status line is followed by a series of entity header fields separated by carriage return/line feeds.

HTTP Request / Response



HTTP Request

GET /filename.htm HTTP/1.1

HTTP Response

HTTP/1.1 200 OK

The Client (Browser) HTTP Request

```
GET /filename.htm HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/msword
Accept-language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozzilla/4.0 (compatable; MSIE 6.0; Windows NT 5.1 )
Host: www.msn.com
Connection: Keep-Alive
```

The above text is sent to the server on TCP/IP port 80

- It asks the server to respond with the contents of filename.htm
- It tells the server that it supports the HTTP1.1 standard
- It tells the server that the client supports: gif, x-xbitmaps, jpeg, and pjpeg images
- It tells the server that it supports msword documents
- It tells the server that the language is English, and that the gzip and deflate decompression algorithm’s are available
- It tells the server that the browser is running IE6.0 on a Windows machine
- Finally it tells the server NOT to close the connection after the file is sent

Keep-Alive

- By default, after the server sends the file to the client, it closes the TCP/IP connection.
- The Keep-Alive request header field tells the server NOT to close the TCP/IP connection after the file contents are sent.
- This decreases the packet overhead for future connections.

The Server Response Header

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/6.0
Cache-Control: no-cache
Content-Type: text/html
Content-Encoding: gzip
Content-Length: 9062
```

Followed by data from file, in this case encoded using gzip

The above data is returned by the server, to the client:

- The HTTP/1.1 200 OK line tells the client/browser that HTTP1.1 is supported, and the 200 tells the client that the file was found
- The Server line informs the client of the Web Server type and version
- The Cache-Control line tells the client to disable cache
- The Content-Type line tells the client the type of data that will follow
- The Content-Encoding line tells the client that the following data is encrypted using gzip
- The Content-Length line tells the client how many bytes are to follow

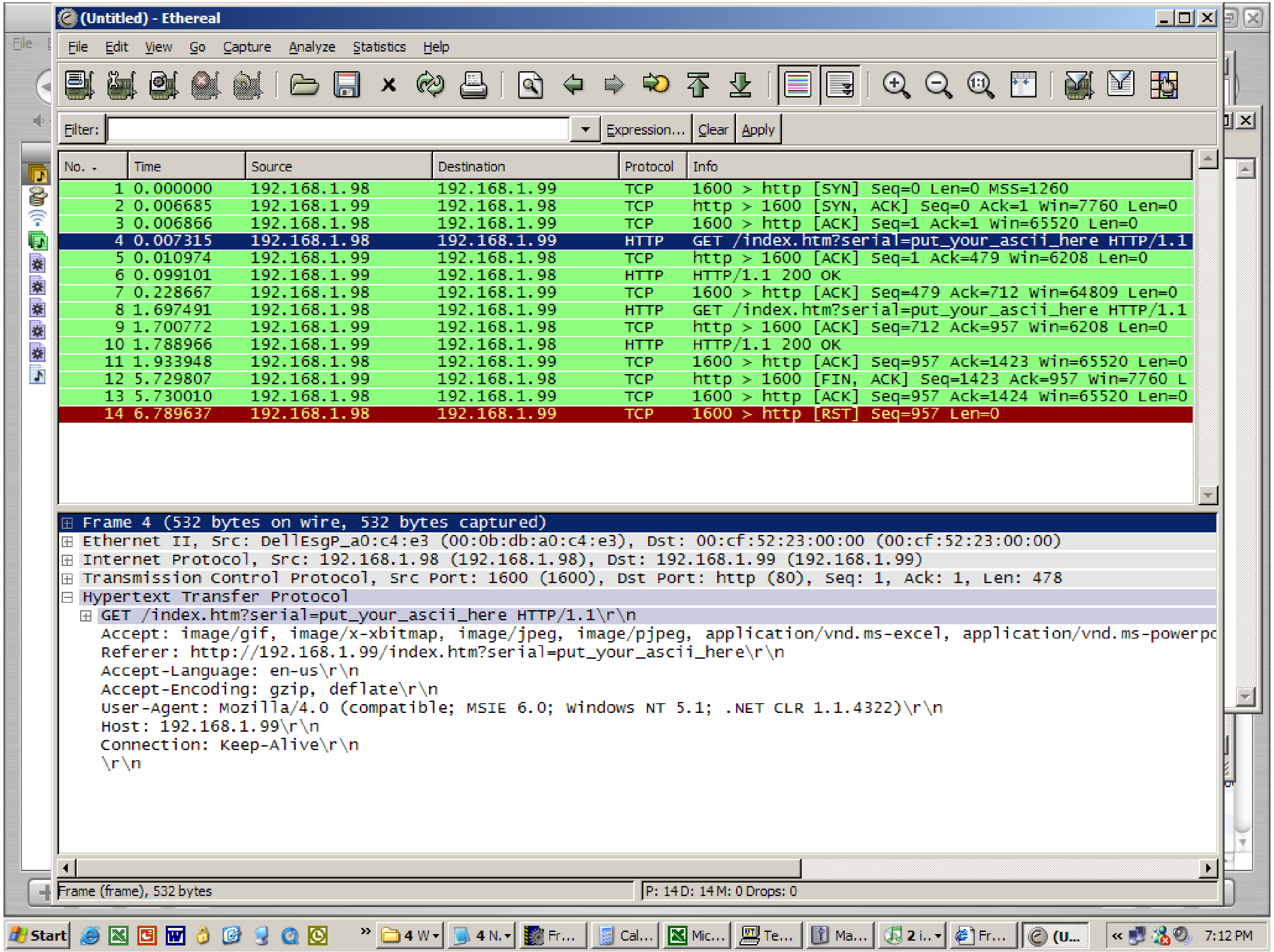
HTTP 1.1

HTTP 1.1 is defined by RFC2616

Additions to HTTP 1.1:

- Faster response, by allowing multiple transactions to take place over a single *persistent connection*.
- Faster response and great bandwidth savings, by adding cache support.
- Faster response for dynamically-generated pages, by supporting *chunked encoding*, which allows a response to be sent before its total length is known.
- Efficient use of IP addresses, by allowing multiple domains to be served from a single IP address.

Ethereal HTTP demo



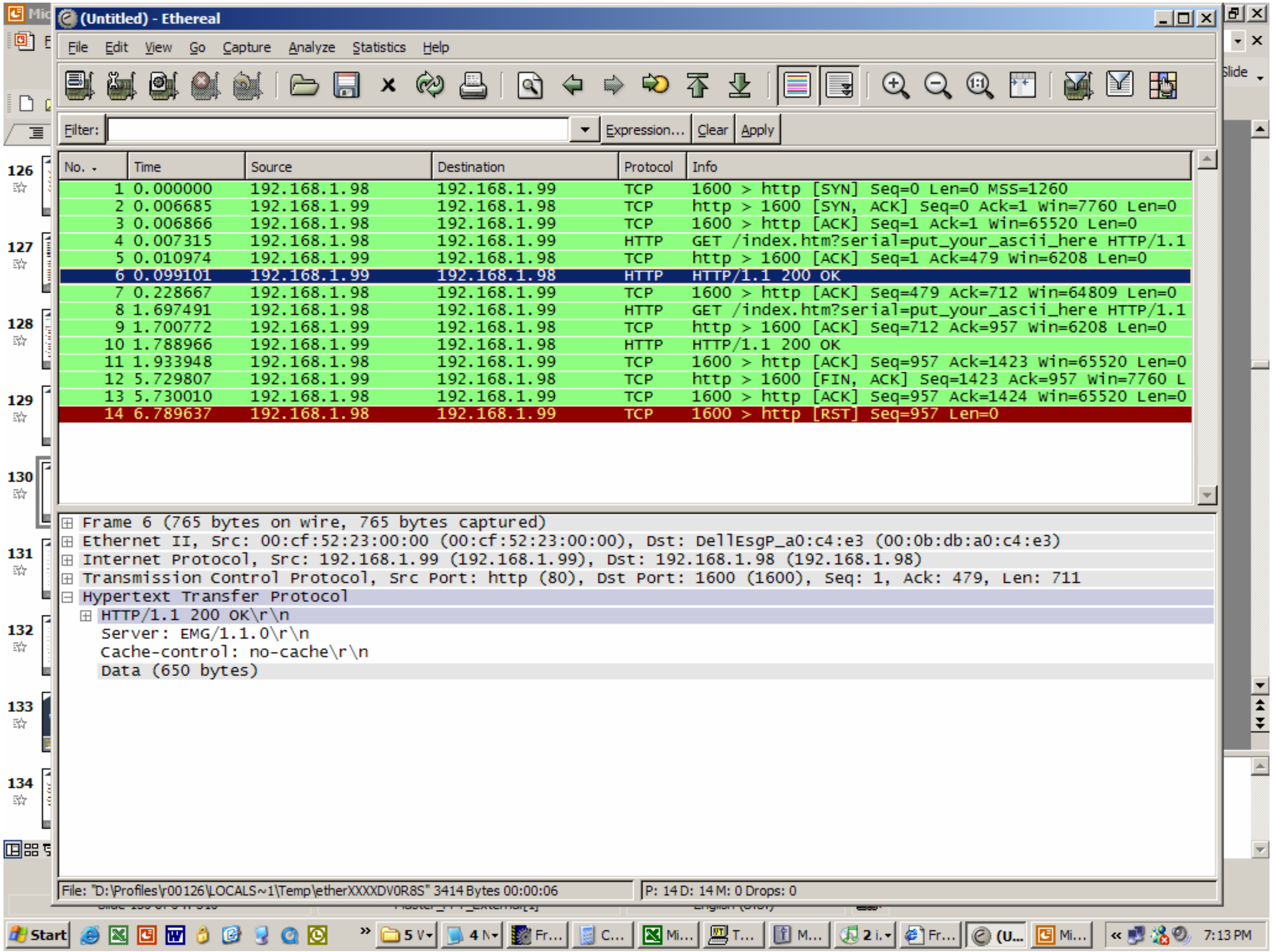
No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.98	192.168.1.99	TCP	1600 > http [SYN] Seq=0 Len=0 MSS=1260
2	0.006685	192.168.1.99	192.168.1.98	TCP	http > 1600 [SYN, ACK] Seq=0 Ack=1 Win=7760 Len=0
3	0.006866	192.168.1.98	192.168.1.99	TCP	1600 > http [ACK] Seq=1 Ack=1 Win=65520 Len=0
4	0.007315	192.168.1.98	192.168.1.99	HTTP	GET /index.htm?serial=put_your_ascii_here HTTP/1.1
5	0.010974	192.168.1.99	192.168.1.98	TCP	http > 1600 [ACK] Seq=1 Ack=479 Win=6208 Len=0
6	0.099101	192.168.1.99	192.168.1.98	HTTP	HTTP/1.1 200 OK
7	0.228667	192.168.1.98	192.168.1.99	TCP	1600 > http [ACK] Seq=479 Ack=712 Win=64809 Len=0
8	1.697491	192.168.1.98	192.168.1.99	HTTP	GET /index.htm?serial=put_your_ascii_here HTTP/1.1
9	1.700772	192.168.1.99	192.168.1.98	TCP	http > 1600 [ACK] Seq=712 Ack=957 Win=6208 Len=0
10	1.788966	192.168.1.99	192.168.1.98	HTTP	HTTP/1.1 200 OK
11	1.933948	192.168.1.98	192.168.1.99	TCP	1600 > http [ACK] Seq=957 Ack=1423 Win=65520 Len=0
12	5.729807	192.168.1.99	192.168.1.98	TCP	http > 1600 [FIN, ACK] Seq=1423 Ack=957 Win=7760 L
13	5.730010	192.168.1.98	192.168.1.99	TCP	1600 > http [ACK] Seq=957 Ack=1424 Win=65520 Len=0
14	6.789637	192.168.1.98	192.168.1.99	TCP	1600 > http [RST] Seq=957 Len=0

```
⊞ Frame 4 (532 bytes on wire, 532 bytes captured)
⊞ Ethernet II, Src: DellEsgP_a0:c4:e3 (00:0b:db:a0:c4:e3), Dst: 00:cf:52:23:00:00 (00:cf:52:23:00:00)
⊞ Internet Protocol, Src: 192.168.1.98 (192.168.1.98), Dst: 192.168.1.99 (192.168.1.99)
⊞ Transmission Control Protocol, Src Port: 1600 (1600), Dst Port: http (80), Seq: 1, Ack: 1, Len: 478
⊟ Hypertext Transfer Protocol
  ⊞ GET /index.htm?serial=put_your_ascii_here HTTP/1.1\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpo
    Referer: http://192.168.1.99/index.htm?serial=put_your_ascii_here\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)\r\n
    Host: 192.168.1.99\r\n
    Connection: Keep-Alive\r\n
    \r\n
```

Frame (frame), 532 bytes

P: 14 D: 14 M: 0 Drops: 0

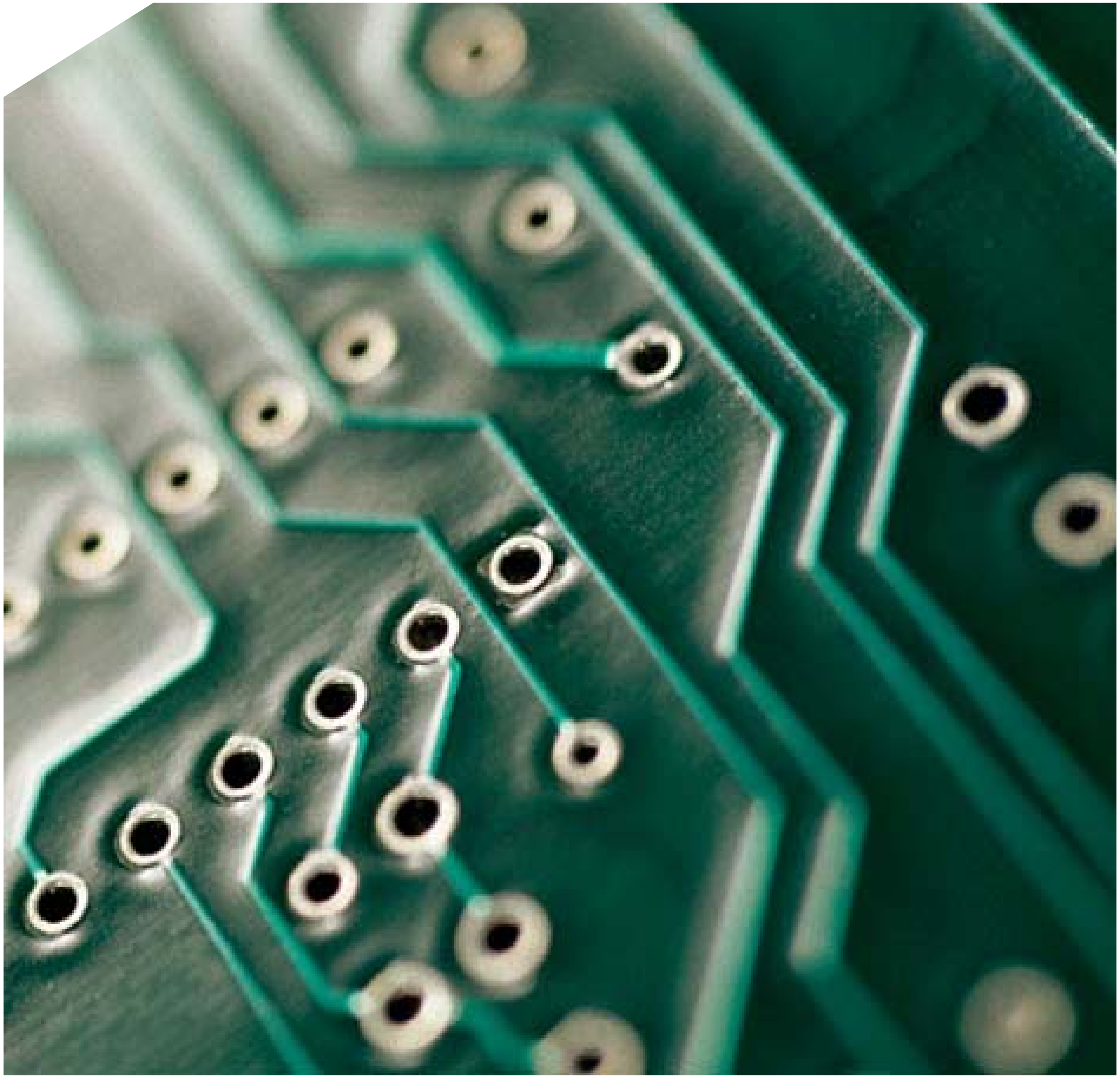
Ethereal HTTP demo



```
No. -  Time      Source          Destination     Protocol  Info
  1 0.000000    192.168.1.98    192.168.1.99    TCP       1600 > http [SYN] Seq=0 Len=0 MSS=1260
  2 0.006685    192.168.1.99    192.168.1.98    TCP       http > 1600 [SYN, ACK] Seq=0 Ack=1 Win=7760 Len=0
  3 0.006866    192.168.1.98    192.168.1.99    TCP       1600 > http [ACK] Seq=1 Ack=1 Win=65520 Len=0
  4 0.007315    192.168.1.98    192.168.1.99    HTTP      GET /index.htm?serial=put_your_ascii_here HTTP/1.1
  5 0.010974    192.168.1.99    192.168.1.98    TCP       http > 1600 [ACK] Seq=1 Ack=479 Win=6208 Len=0
  6 0.099101    192.168.1.99    192.168.1.98    HTTP      HTTP/1.1 200 OK
  7 0.228667    192.168.1.98    192.168.1.99    TCP       1600 > http [ACK] Seq=479 Ack=712 Win=64809 Len=0
  8 1.697491    192.168.1.98    192.168.1.99    HTTP      GET /index.htm?serial=put_your_ascii_here HTTP/1.1
  9 1.700772    192.168.1.99    192.168.1.98    TCP       http > 1600 [ACK] Seq=712 Ack=957 Win=6208 Len=0
 10 1.788966    192.168.1.99    192.168.1.98    HTTP      HTTP/1.1 200 OK
 11 1.933948    192.168.1.98    192.168.1.99    TCP       1600 > http [ACK] Seq=957 Ack=1423 Win=65520 Len=0
 12 5.729807    192.168.1.99    192.168.1.98    TCP       http > 1600 [FIN, ACK] Seq=1423 Ack=957 Win=7760 L
 13 5.730010    192.168.1.98    192.168.1.99    TCP       1600 > http [ACK] Seq=957 Ack=1424 Win=65520 Len=0
 14 6.789637    192.168.1.98    192.168.1.99    TCP       1600 > http [RST] Seq=957 Len=0
```

```
⊞ Frame 6 (765 bytes on wire, 765 bytes captured)
⊞ Ethernet II, Src: 00:cf:52:23:00:00 (00:cf:52:23:00:00), Dst: DellEsgP_a0:c4:e3 (00:0b:db:a0:c4:e3)
⊞ Internet Protocol, Src: 192.168.1.99 (192.168.1.99), Dst: 192.168.1.98 (192.168.1.98)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 1600 (1600), Seq: 1, Ack: 479, Len: 711
⊟ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 200 OK\r\n
    Server: EMG/1.1.0\r\n
    Cache-control: no-cache\r\n
    Data (650 bytes)
```

```
File: "D:\Profiles\r00126\LOCALS~1\Temp\etherXXXXDV0R8S" 3414 Bytes 00:00:06    P: 14 D: 14 M: 0 Drops: 0
```



Web Server/Customer Web Server Labs

Loading the Default Static Web Page

- The purpose of this lab is to use CodeWarrior® to build and load the stack and default web page.
- The static file system utility will be used to change the default static web page.
- We will also learn how to configure the static IP address in both the demo board and the PC.

Using CodeWarrior® to Build the Default Web Page

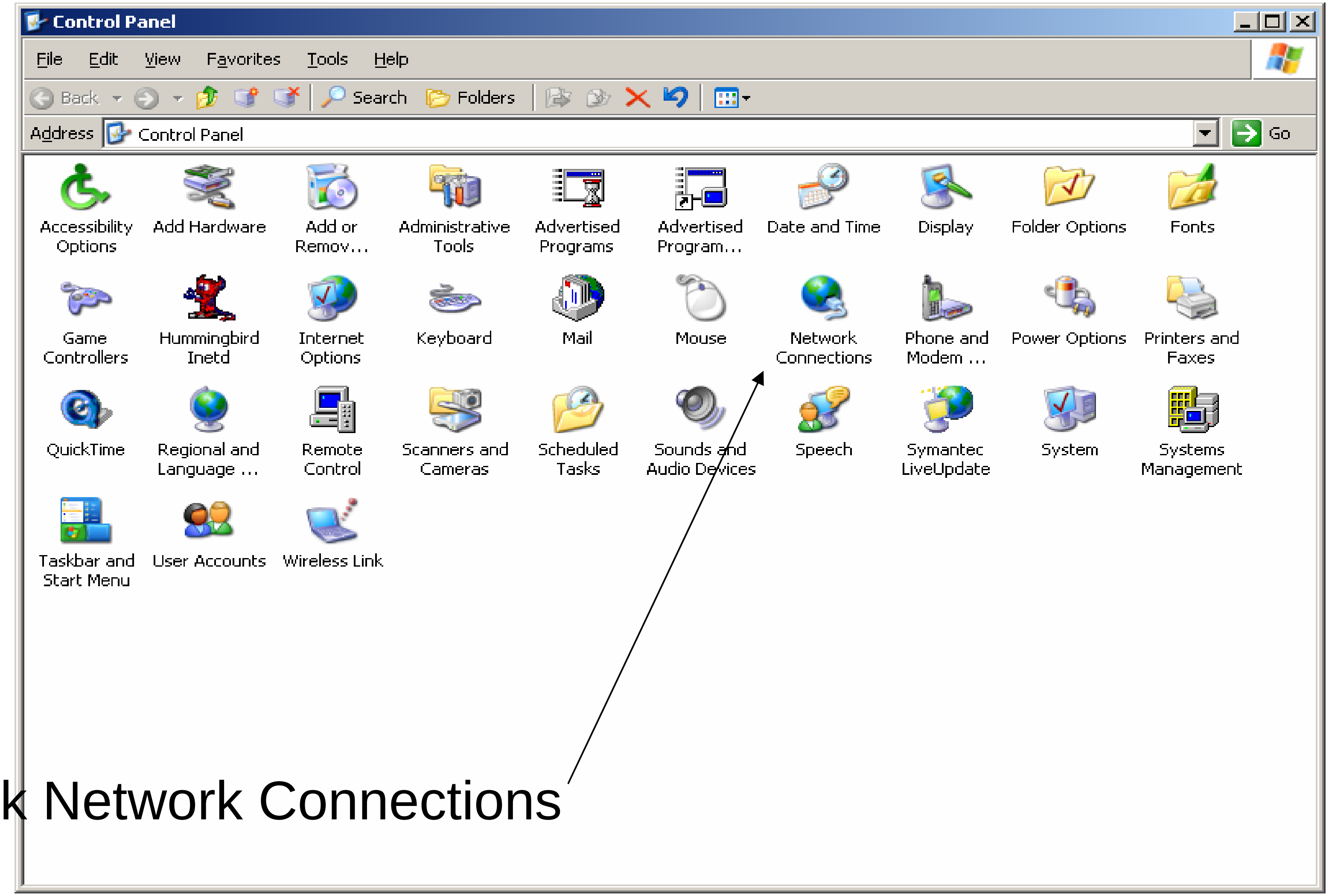
- Follow the instructions from the CodeWarrior lab to configure CodeWarrior and the flash programmer for the MCF5223x.
- Load the MCP file

Set up PC Network Connection

Follow one of the following two methods:

- From Control Panel install new connect.
- Use existing connection.

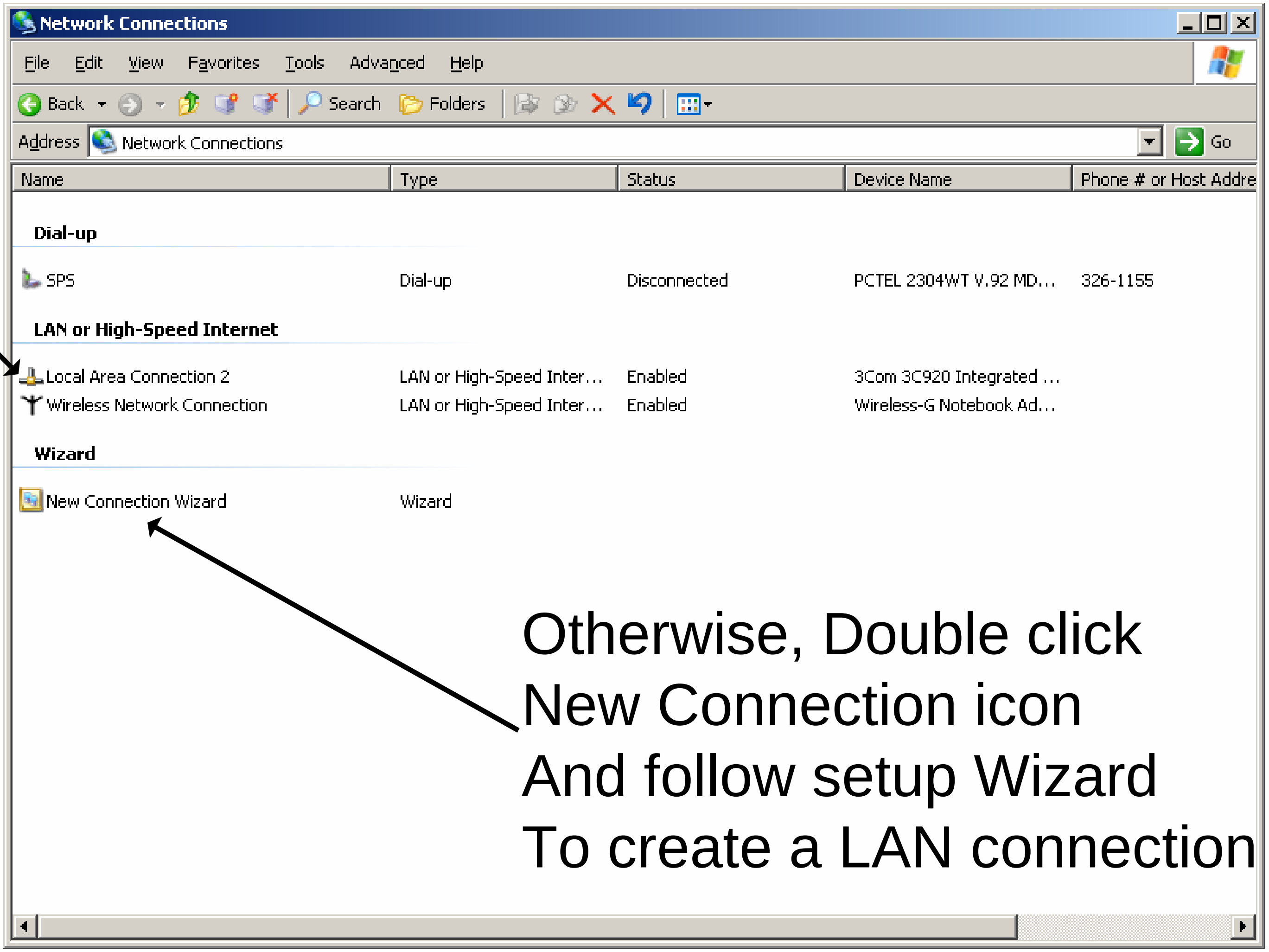
Set up PC Network Connection



Double click Network Connections

Set up PC Network Connection

If available, Double click icon.



Name	Type	Status	Device Name	Phone # or Host Addre
Dial-up				
SPS	Dial-up	Disconnected	PCTEL 2304WT V.92 MD...	326-1155
LAN or High-Speed Internet				
Local Area Connection 2	LAN or High-Speed Inter...	Enabled	3Com 3C920 Integrated ...	
Wireless Network Connection	LAN or High-Speed Inter...	Enabled	Wireless-G Notebook Ad...	
Wizard				
New Connection Wizard	Wizard			

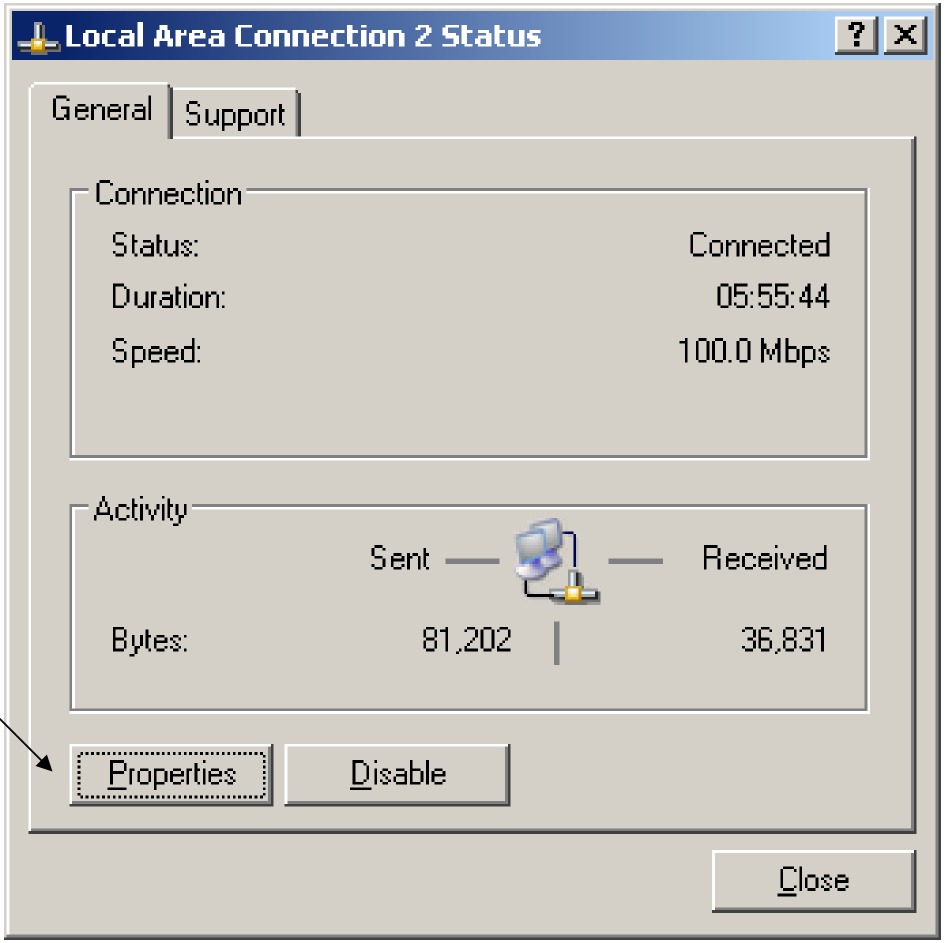
Otherwise, Double click
New Connection icon
And follow setup Wizard
To create a LAN connection

Set up PC Network Connection

Now that a LAN connection is available

Let’s set it up for our needs

Click on Properties Tab



Local Area Connection 2 Status

Connection

Status: Connected

Duration: 05:55:44

Speed: 100.0 Mbps

Activity

Sent — — Received

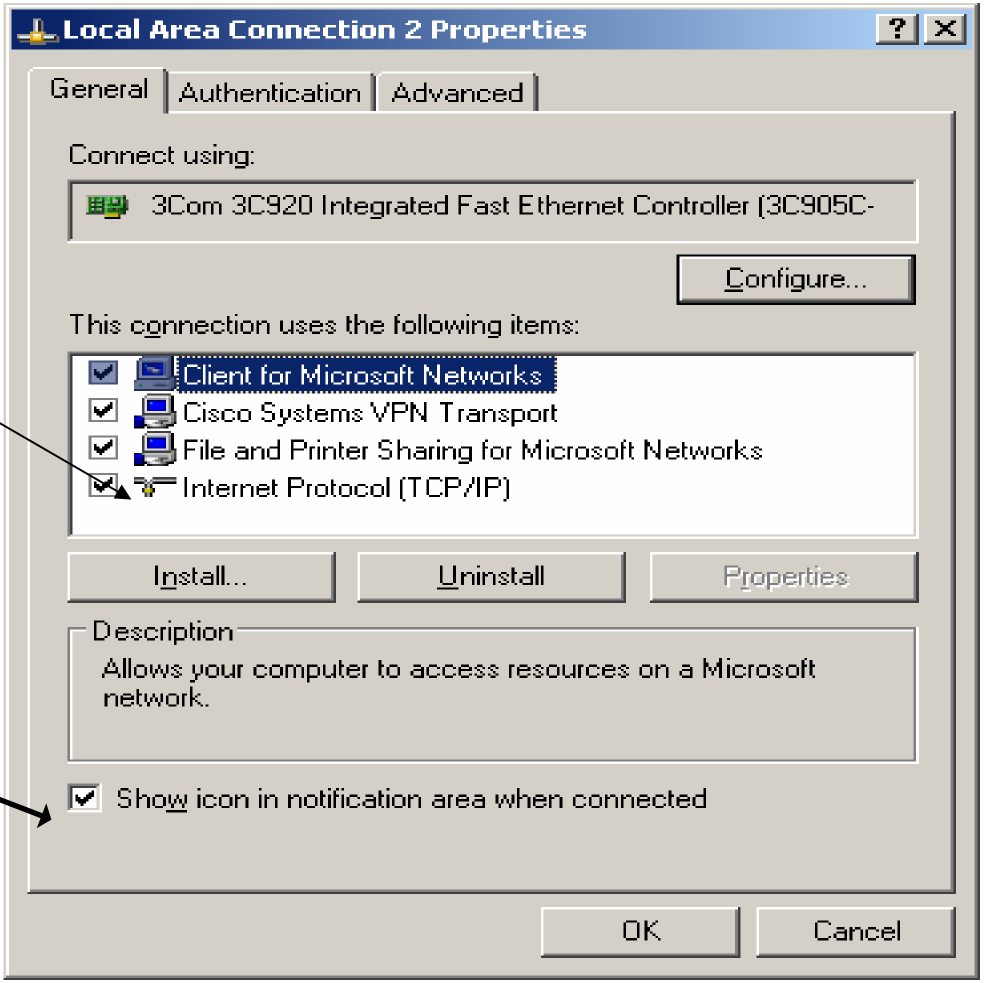
Bytes: 81,202 | 36,831

Set up PC Network Connection

The following properties dialog will open

Double Click on the Internet Protocol (TCP/IP) Icon

Checking this will aid config changes later



Local Area Connection 2 Properties

General | Authentication | Advanced

Connect using:

3Com 3C920 Integrated Fast Ethernet Controller (3C905C-

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ Cisco Systems VPN Transport
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol (TCP/IP)

Install... | Uninstall | Properties

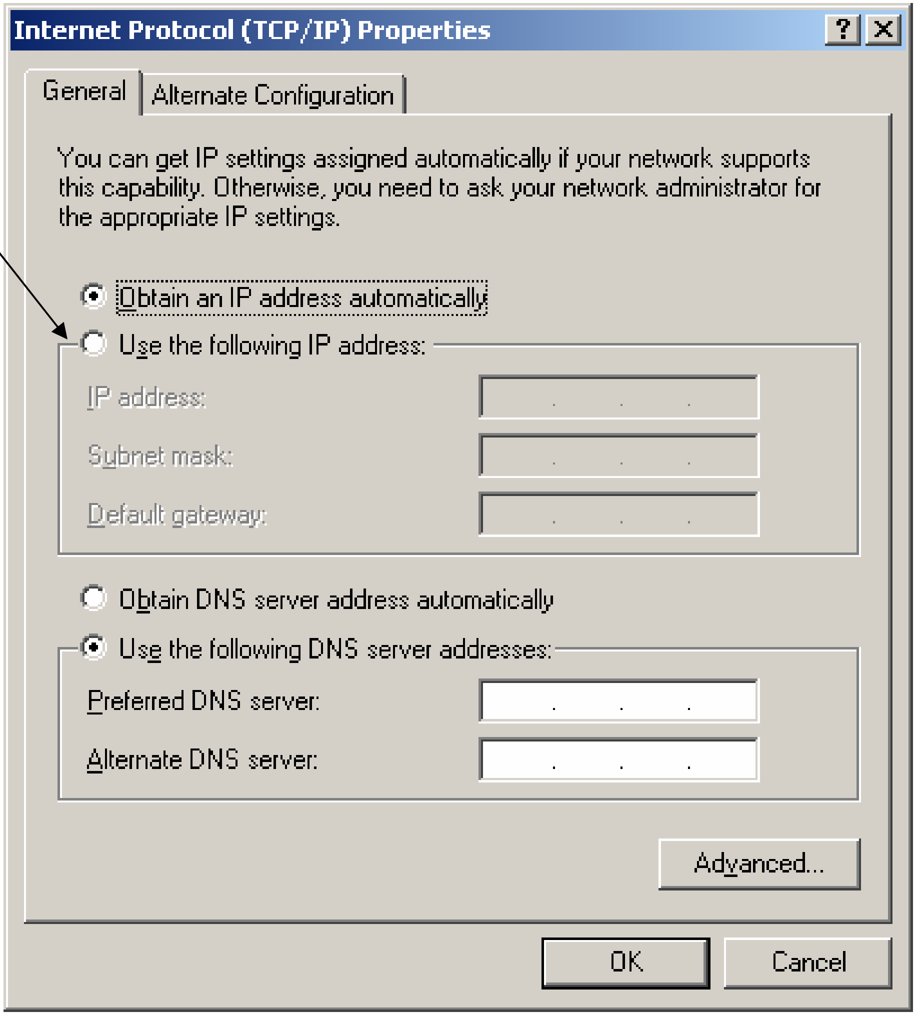
Description

Allows your computer to access resources on a Microsoft network.

☑ Show icon in notification area when connected

Set up PC Network Connection

Select
Use the following IP address



Internet Protocol (TCP/IP) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically
○ Use the following IP address:

IP address: . . .
Subnet mask: . . .
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

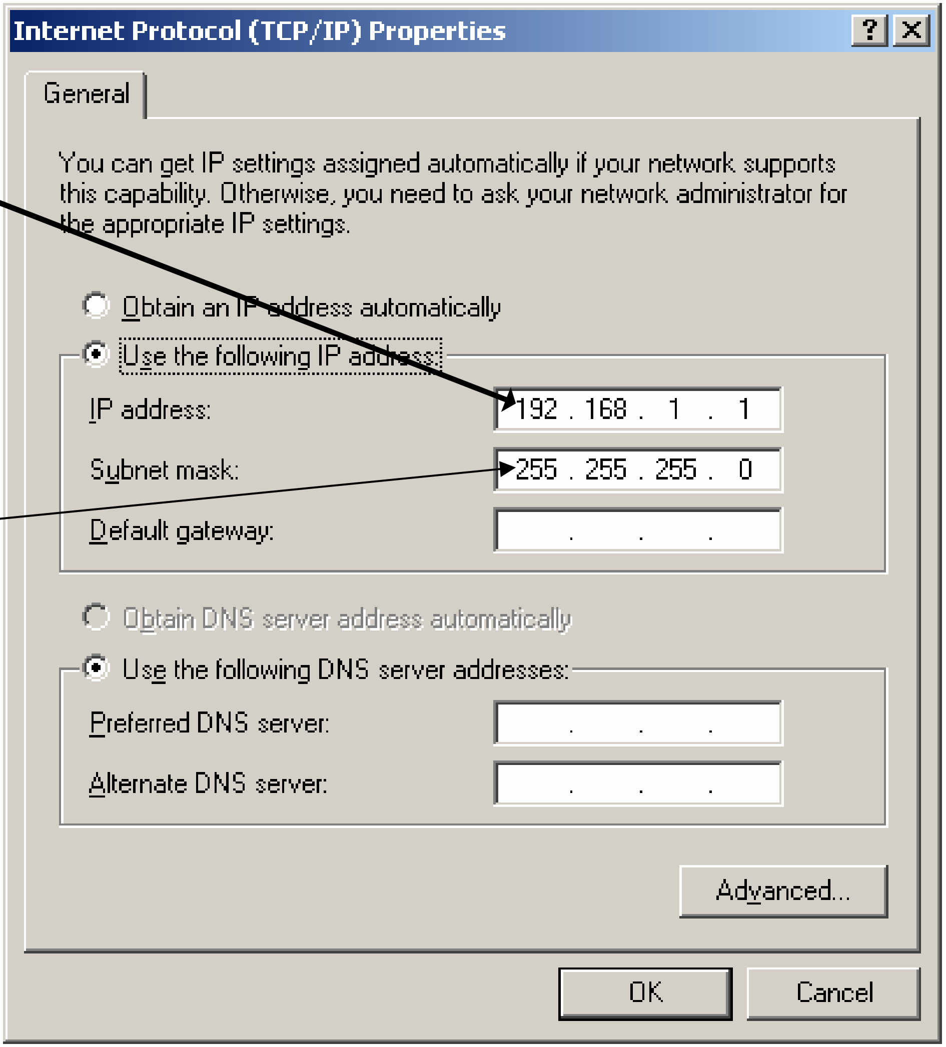
Preferred DNS server: . . .
Alternate DNS server: . . .

Set up PC Network Connection

Enter 192.168.1.1 for the IP address

Click in the Subnet Mask Field and it will auto-fill with 255.255.255.0

Click OK on all LAN setup dialog boxes and close them



Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 1

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

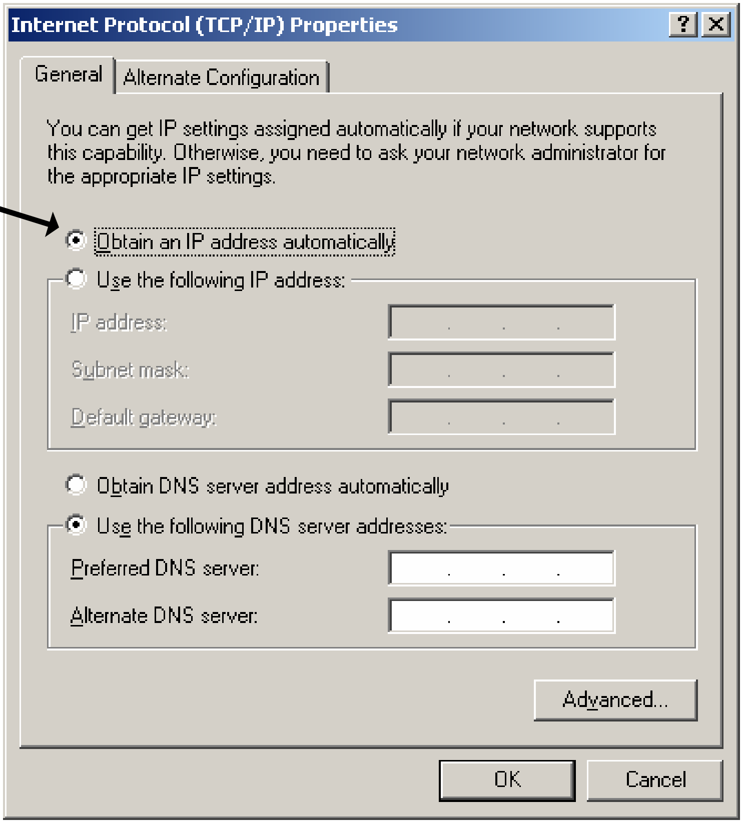
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Set up PC Network Connection

To return your LAN setting for normal Operation reopen the Internet Properties Dialog box and select Auto IP address



Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

○ Obtain DNS server address automatically

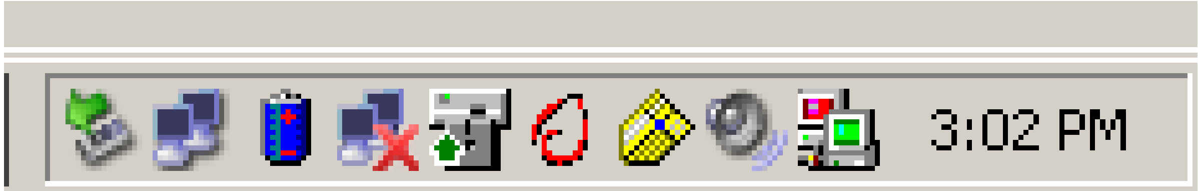
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

The Taskbar

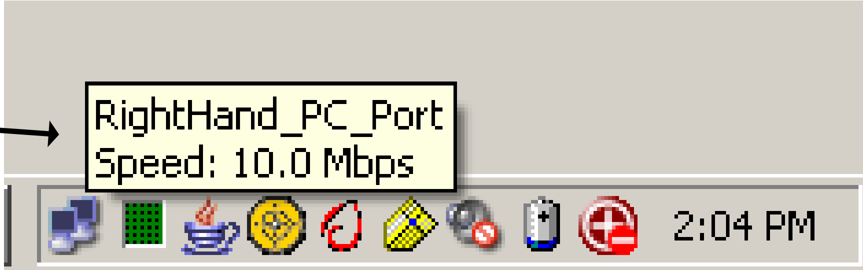
If there was a connection previously, an icon may be on the taskbar



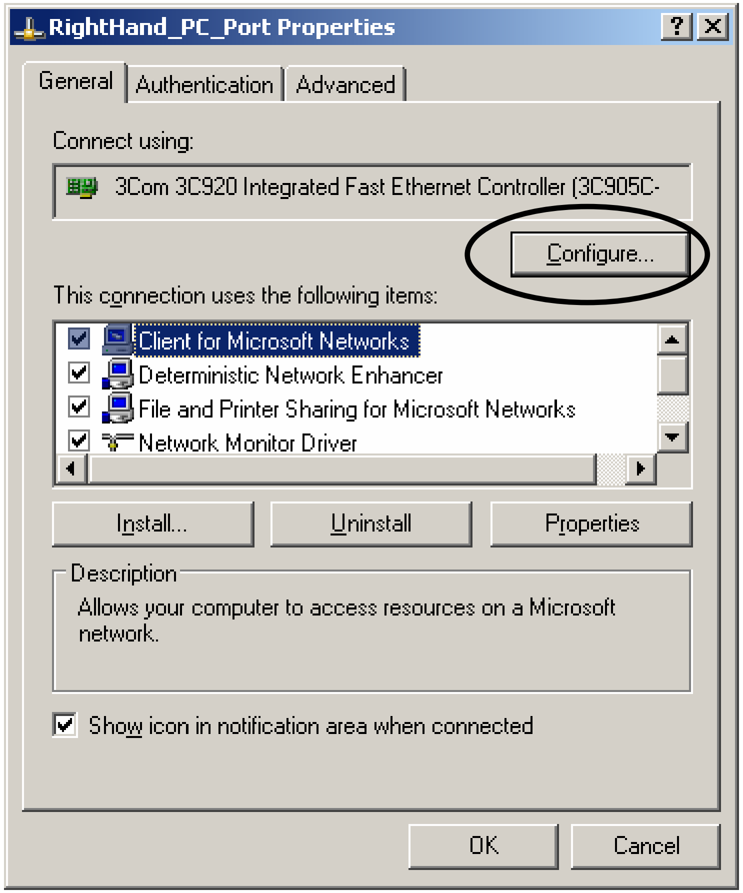
Configured LAN connections

If a connection bubble like shown at left is not shown (100bT is OK too)

And you have a 1gbit card reconfigure your card as shown in the next slides


RightHand_PC_Port
Speed: 10.0 Mbps

Setting Speed to 100Mb, Half for 1Gbit Cards



RightHand_PC_Port Properties

Connect using:

3Com 3C920 Integrated Fast Ethernet Controller (3C905C-

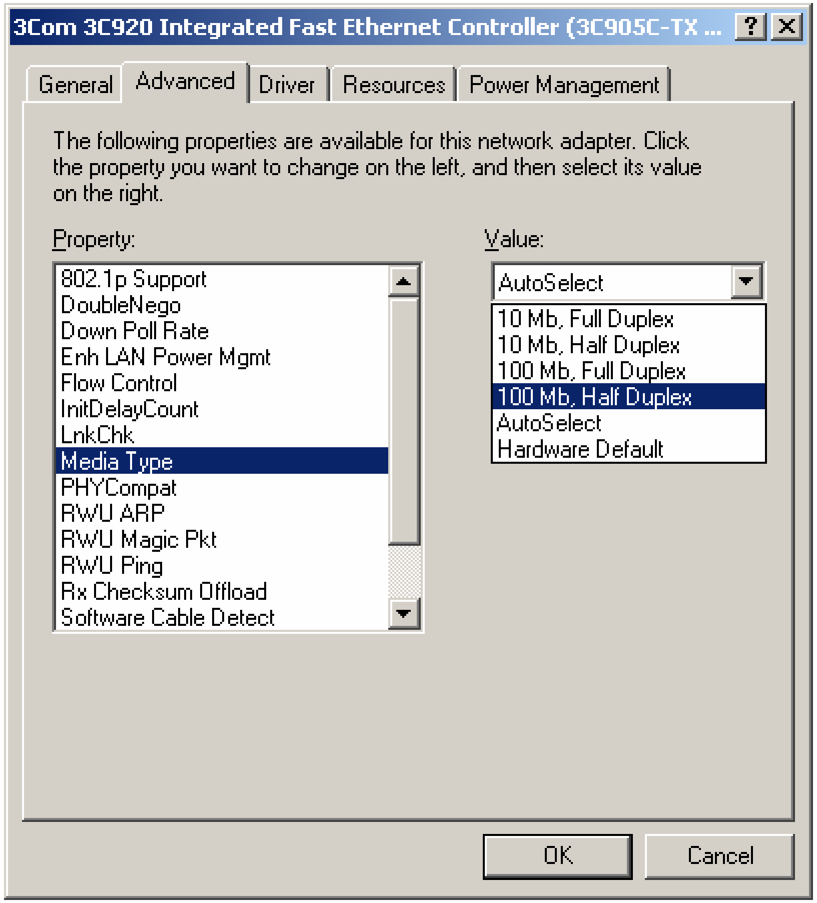
This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ Deterministic Network Enhancer
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Network Monitor Driver

Description

Allows your computer to access resources on a Microsoft network.

☑ Show icon in notification area when connected



3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX ...

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

- 802.1p Support
- DoubleNego
- Down Poll Rate
- Enh LAN Power Mgmt
- Flow Control
- InitDelayCount
- LnkChk
- Media Type
- PHYCompat
- RWU ARP
- RWU Magic Pkt
- RWU Ping
- Rx Checksum Offload
- Software Cable Detect

Value:

AutoSelect

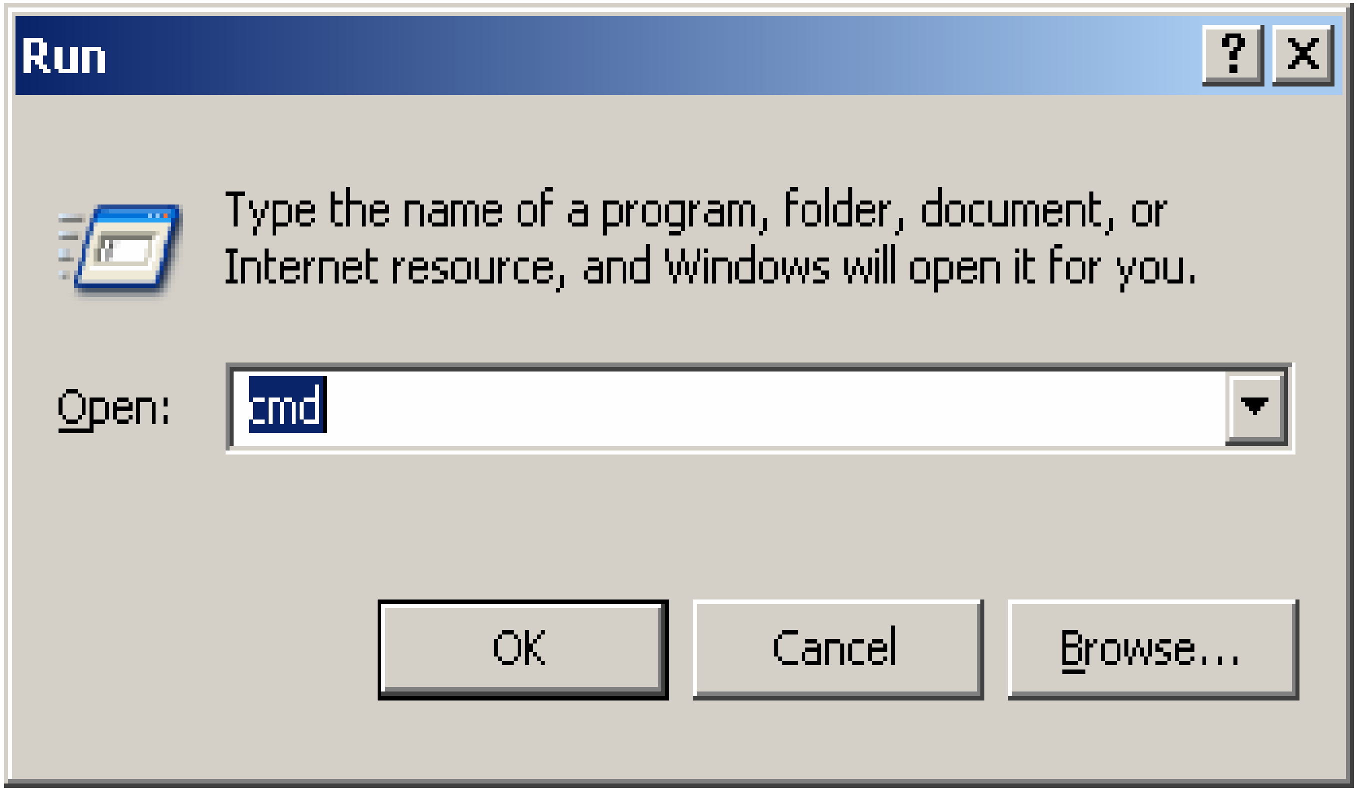
- 10 Mb, Full Duplex
- 10 Mb, Half Duplex
- 100 Mb, Full Duplex
- 100 Mb, Half Duplex
- AutoSelect
- Hardware Default

*****Only needed if communications issues with 1 Gbit card

RUN

From the Start menu select RUN

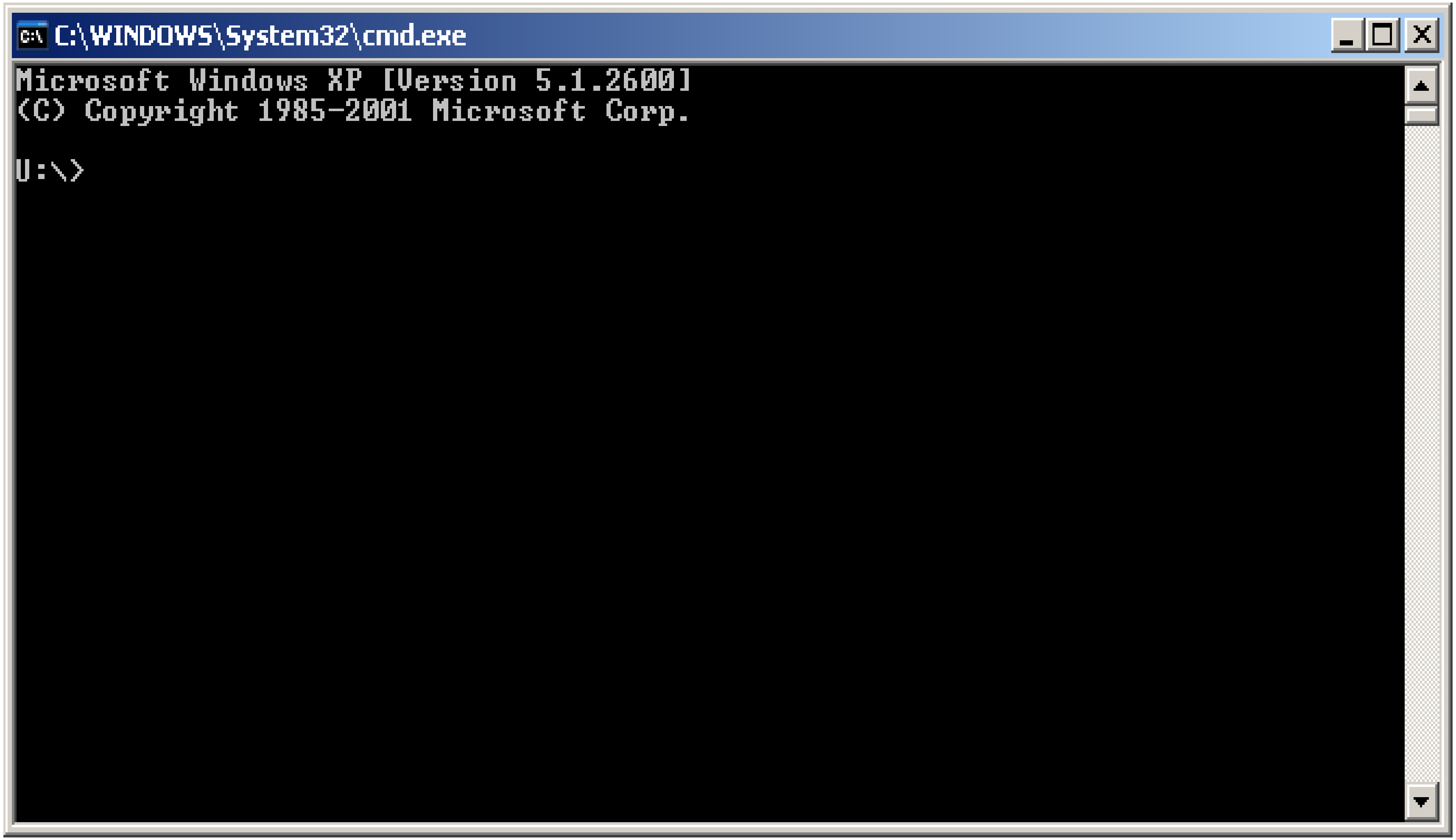
Enter “CMD”, click OK



Run
?
X
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
cmd
OK
Cancel
Browse...

DOS Window

A DOS window should open.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

U:\>
```

Enter -> ping 192.168.1.99

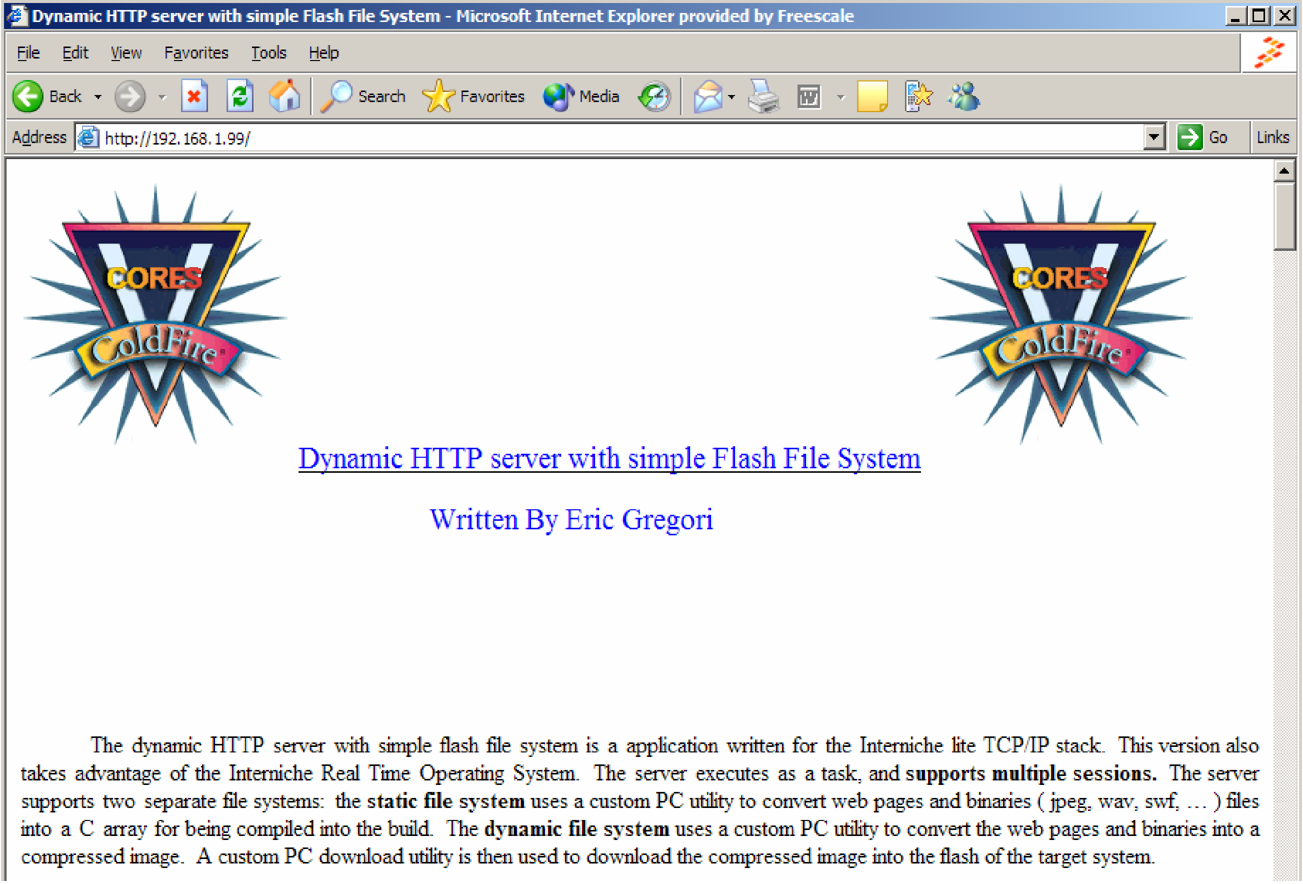
At the DOS prompt type ping 192.168.1.99 then hit enter

If Ping Does not work

- Go to your hyperterminal window, hit enter a few times.
- Verify a INET> prompt appears.
- Verify that you have a cross connect cable.
- Verify that you have disabled VPN (on your personal machine)
- Type   iface soft   at the INET> prompt.
- Try Ping again after 2 seconds.

The Default Web Page

Open Internet Explorer, and type 192.168.1.99 (the IP address of the demo board) into the address bar. This is the default compile time web page you just loaded with the TCP/IP stack and Web Server.



Dynamic HTTP server with simple Flash File System - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
Go
Links
CORES
ColdFire
CORES
ColdFire

http://192.168.1.99/

Dynamic HTTP server with simple Flash File System

Written By Eric Gregori

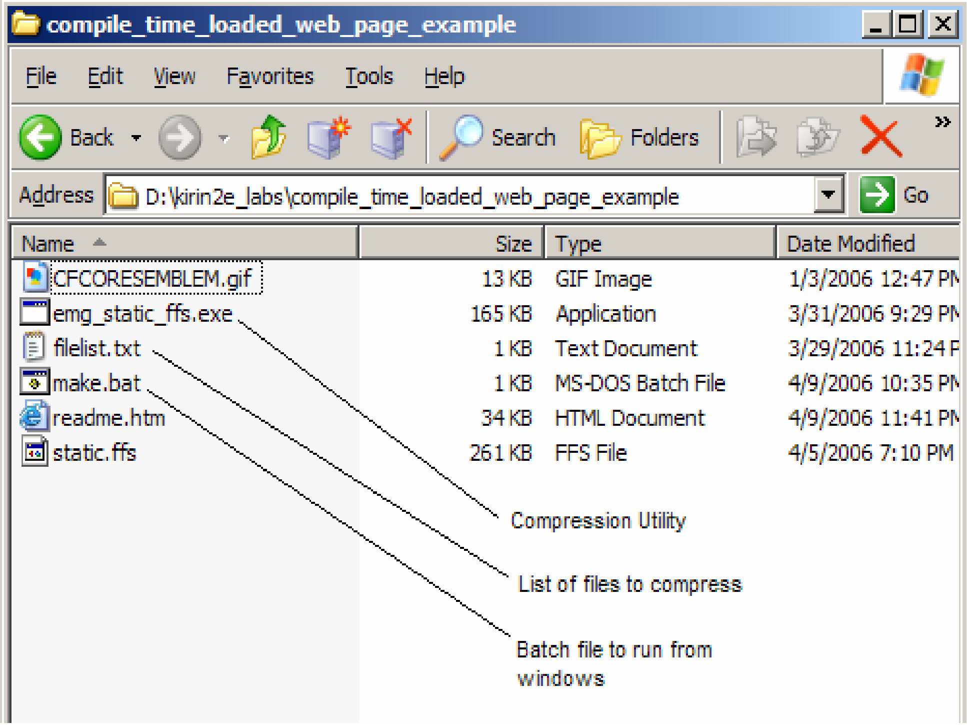
The dynamic HTTP server with simple flash file system is a application written for the Interniche lite TCP/IP stack. This version also takes advantage of the Interniche Real Time Operating System. The server executes as a task, and **supports multiple sessions.** The server supports two separate file systems: the **static file system** uses a custom PC utility to convert web pages and binaries (jpeg, wav, swf, …) files into a C array for being compiled into the build. The **dynamic file system** uses a custom PC utility to convert the web pages and binaries into a compressed image. A custom PC download utility is then used to download the compressed image into the flash of the target system.

The Static/Compile Time Flash File System

- The Static/Compile Time Flash File System allows the user to embed web pages consisting of one or multiple files into a target build.
- The system has two parts: The firmware running in the ColdFire® processor as part of the Web Server, and the compression utility which is executed on the PC.
- The Compression utility takes a list of files, and compresses them into a single ‘C’ file. The ‘C’ file is then compiled and linked into the final target build with the TCP/IP stack and the Web Server.

The Static/Compile Time Flash File System Compression Utility

The compression utility: emg_static_ffs.exe is a DOS command utility that can be executed from windows using a BATCH file.


compile_time_loaded_web_page_example
File
Edit
View
Favorites
Tools
Help
Back
Search
Folders
Address
D:\kirin2e_labs\compile_time_loaded_web_page_example
Go

Name	Size	Type	Date Modified
CFCORESEMBLEM.gif	13 KB	GIF Image	1/3/2006 12:47 P
emg_static_ffs.exe	165 KB	Application	3/31/2006 9:29 P
filelist.txt	1 KB	Text Document	3/29/2006 11:24 P
make.bat	1 KB	MS-DOS Batch File	4/9/2006 10:35 P
readme.htm	34 KB	HTML Document	4/9/2006 11:41 P
static.ffs	261 KB	FFS File	4/5/2006 7:10 PM

Compression Utility

List of files to compress

Batch file to run from windows

Compression Utility Command Line Arguments

Emg_static_ffs filelist.txt output_file.c

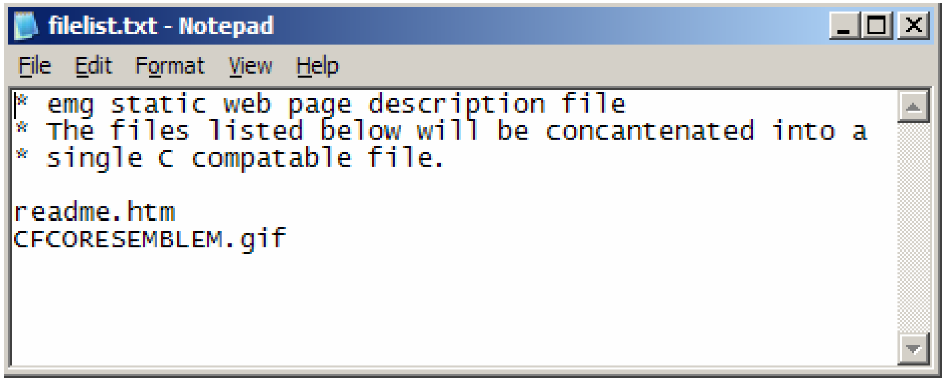
Where:

- Filelist.txt is a text file containing the list of files to compress. Each file should be on its own line, and the first file is the default.
- Comments can be added using a ‘*’ as the first character in a line.
- Output_file.c is the file generated containing all the files in the filelist compressed together, along with data structures used to reference the files from the Web Server.

Sample filelist.txt

* emg static web page description file
* The files listed below will be concantenated into a
* single C compatable file.

readme.htm
CFCORESEMBLEM.gif



```
* emg static web page description file
* The files listed below will be concantenated into a
* single C compatable file.

readme.htm
CFCORESEMBLEM.gif
```

The last line must be a blank line with just a CRLF (just hit enter in the last blank line).

The output_file.c

```
//***********************************************************************
//* Static Flash File System Generator                                  *
//* Written by Eric Gregori - Chicago FAE                                *
//*                                                     *
//***********************************************************************


  const unsigned char readme_htm[] = {
  0x48,0x54,0x54,0x50,0x2F,0x31,0x2E,0x31,0x20,0x32,

  Data removed for space in presentation

  0x0D,0x0A,0x00 };

  const unsigned char CFCORESEMBLEM_gif[] = {
  0x48,0x54,0x54,0x50,0x2F,0x31,0x2E,0x31,0x20,0x32,

  Data removed for space in presentation

  0xA8,0xC7,0x3D,0xF2,0xB1,0x8F,0x7E,0xFC,0x23,0x20,
  0x6F,0x17,0x10,0x00,0x3B,0x00 };

  const char *emg_static_ffs_filenames[] =   {
                                        "readme.htm",
                                        "CFCORESEMBLEM.gif"
                                       };

  const unsigned char *emg_static_ffs_ptrs[] =   {
                                        readme_htm,
                                        CFCORESEMBLEM_gif
                                       };

  const unsigned short emg_static_ffs_len[] =     {
                                        34506,
                                        12919
                                       };

  const unsigned char emg_static_ffs_type[] =     {
                         0x68746d6c,
                             0x67696666
                                       };

  const unsigned char emg_static_ffs_nof = 2;
```

The output file contains the contents of each file stored as a ‘C’ array. The files inserted are from the filelist.txt file (see previous slides).

Array containing list of filenames

Array containing list of pointers to files.

Array containing file sizes

Array containing file type

Number of files

Other Uses for the Static/Compile Time Flash File System

- User Data can also be stored in the static system. The data can be binary or text, but name the file *.txt. The utility actually treats all files as binary files.
- The user can access the data from the firmware using examples in the firmware.
- This feature can be usefull in the static/Compile Time System, but is considerably more usefull in the run time loadable system.

Static/Compile Time Web Page LAB 1

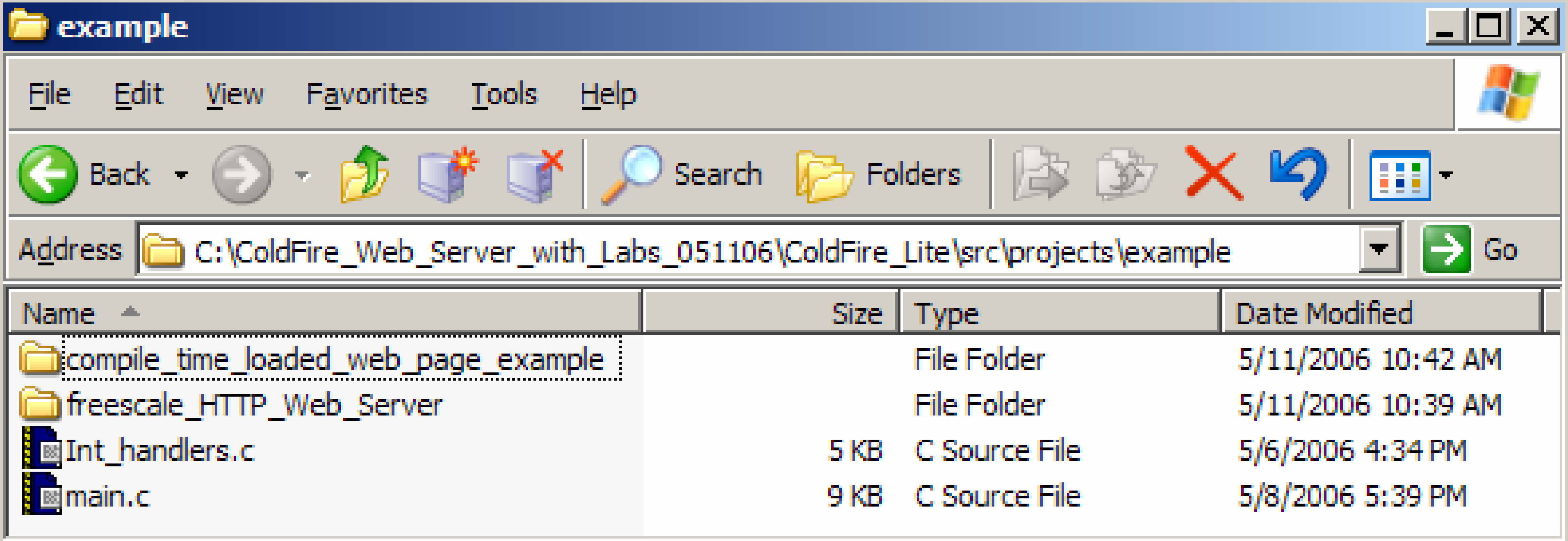
- We are going to edit a HTML file.
- Build a Compressed ‘C’ image.
- Copy the Image to our project.
- Re-build the project.
- Load the new image in flash.

ColdFire_Lite Compile_Time_Loaded_Web_Page_Example

The Compile_Time_Loaded_Web_Page_Example

This is the directory for the static web page demo/lab.

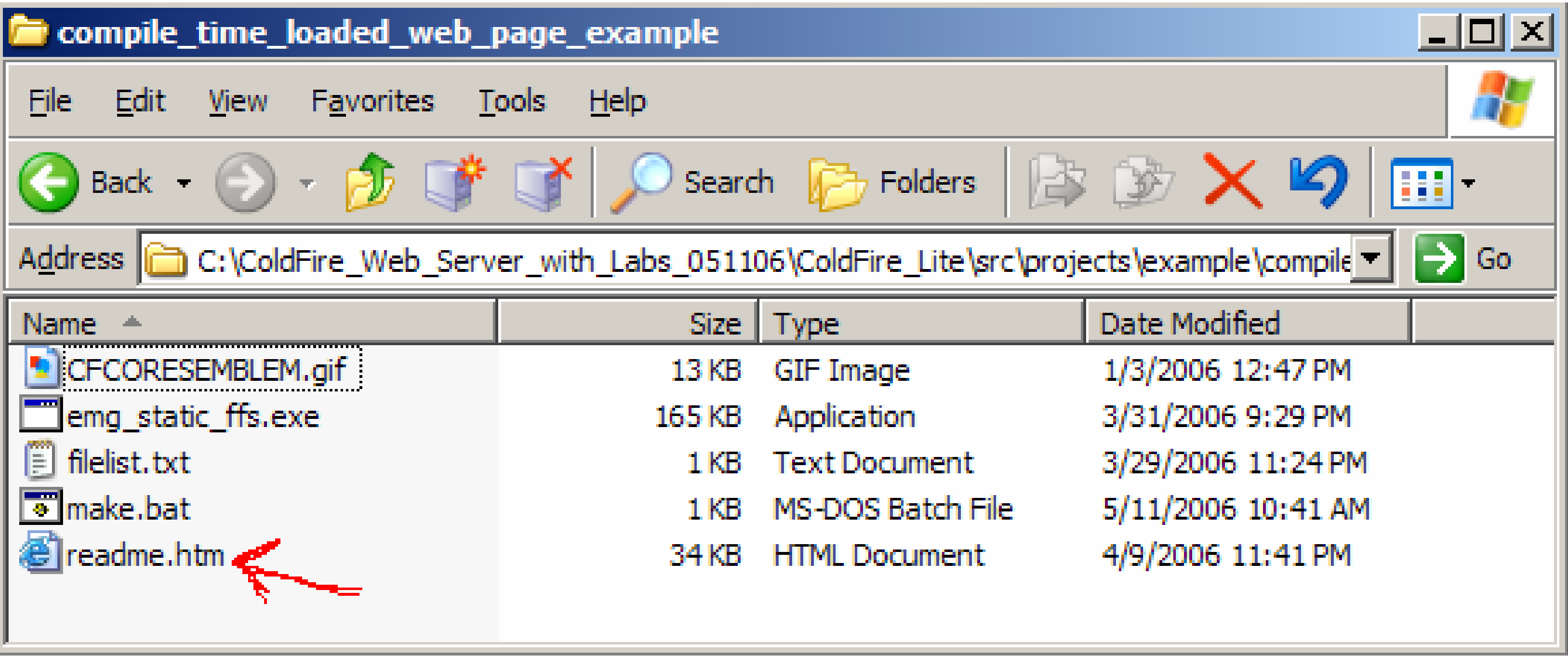
ColdFire_Lite\src\projects\example



C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\src\projects\example

Name	Size	Type	Date Modified
compile_time_loaded_web_page_example		File Folder	5/11/2006 10:42 AM
freescale_HTTP_Web_Server		File Folder	5/11/2006 10:39 AM
Int_handlers.c	5 KB	C Source File	5/6/2006 4:34 PM
main.c	9 KB	C Source File	5/8/2006 5:39 PM

Opening the HTML File



compile_time_loaded_web_page_example

Address C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\src\projects\example\compile

Name	Size	Type	Date Modified
CFCORESEMBLEM.gif	13 KB	GIF Image	1/3/2006 12:47 PM
emg_static_ffs.exe	165 KB	Application	3/31/2006 9:29 PM
filelist.txt	1 KB	Text Document	3/29/2006 11:24 PM
make.bat	1 KB	MS-DOS Batch File	5/11/2006 10:41 AM
readme.htm	34 KB	HTML Document	4/9/2006 11:41 PM

HTML- An Overview

- HTML or HyperText Markup Language is the language used to describe web pages.
- HTML is a ascii text based language that defines how text and images are placed on a page.
- HTML is a ascii text based language that uses “tags” to instruct a web browser how text and images are placed on a page.

HTML Tags

- Tags start with a ‘<‘ and end with a ‘>’.
- Most tags have a open and close form.
- The open form <HTML>
- The close form </HTML>
- Tag form: <TAG ATTRIBUTE=value>
- Tags/attributes are used to define placement, color, style, and fonts for text.
- Tags are also used to define position and size for a image.

A Simple Web Page

```
<HTML>
<HEAD>
<TITLE>This text will appear at the top of the web browser, the navigation bar</TITLE>
</HEAD>
<BODY>
<CENTER>Hello World</CENTER>
</BODY>
</HTML>
```

The HTML element is used to tell the web browser that we are using HTML instead of JavaScript, or some other language.

The HEAD element contains meta-information. Meta-information is not part of the body of the document but defines the document in a general sense. The Title of the web page is a good example. It is not displayed in the body of the web page, but on the navigation bar of the web browser.

The BODY element defines the displayed portion of the web page.

Some Interesting HTML Tags

<CENTER>	Centers the object on the page.
<Hx>	Heading Size x, where x is from 1-6.
<P>	Start or paragraph.
<FONT COLOR=RED>	Sets font color to red.
<FONT SIZE=x>	Sets font size, where x is from 1-?.
<A HREF=“freescale.com”>	Makes text a URL pointing to freescale.com.
<IMG SRC=“filename.jpg”>	Puts the image filename.jpg into the web page.
<IMG SRC=filename.jpg” ALIGN=center>	Loads the image filename.jpg and centers it in the page.
<TABLE>	Creates a table with the help of <TR> table row and <TD> table data.

Creating Web Pages

- Using notepad, you can start writing HTML immediately, and build your own Web Page.
- Or, you can use an HTML generator.
  - These programs allow you to design a web page, and generate the HTML for you.
  - Just search for “HTML generator” on the web.
  - There are dozens of them, some free.

Using Microsoft Word

Microsoft Word can also be used to generate a Web Page

- By saving a document as *.htm in Microsoft Word, Word will create a web page.
  - The web pages created by Word tend to be very large.
  - Also, Word creates a subdirectory for images.
  - Be sure to change the image reference paths to remove the directories.
- The web page for this lab (readme.htm) was generated in Word.

Edit the HTML

To Edit the HTML open the readme.HTM file in Notepad

- The first few lines of the readme.htm file

```
<html>
<head>
<title>Dynamic HTTP server with simple Flash File System</title>
</head>
```

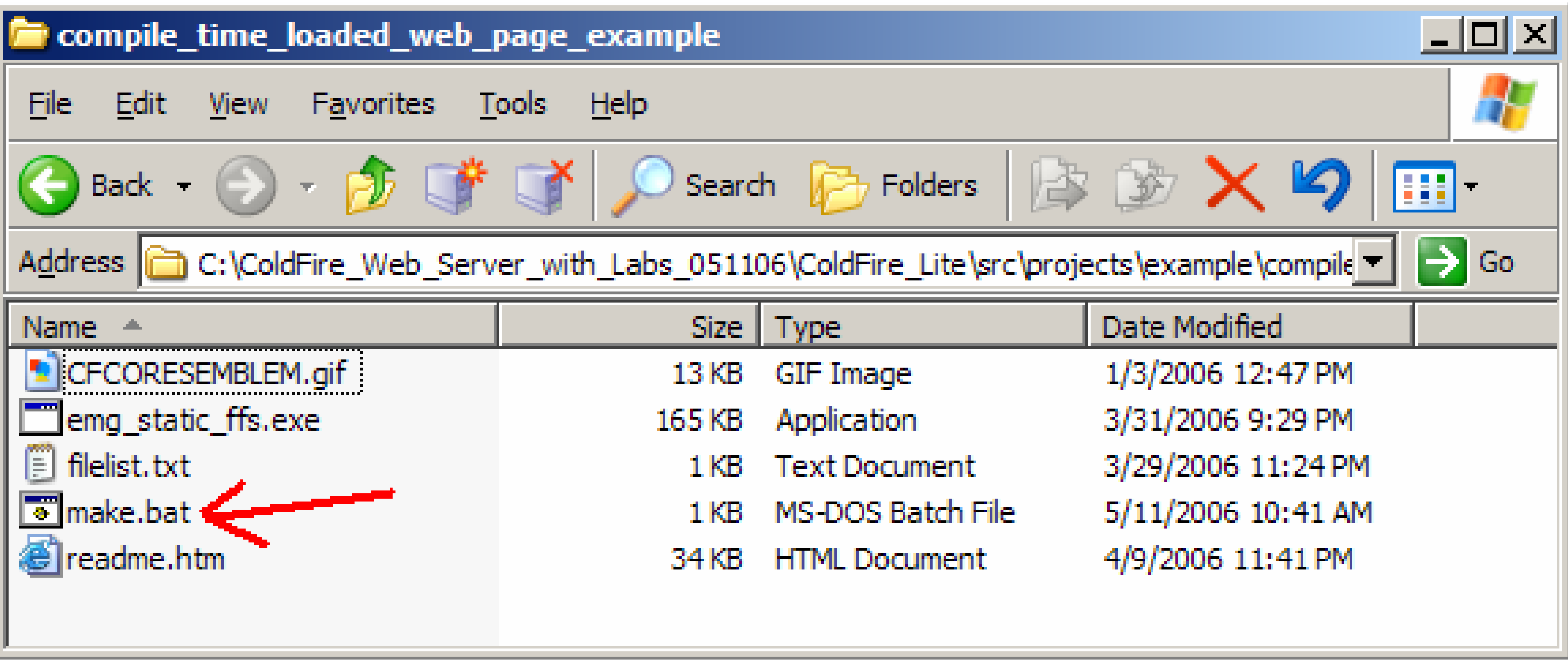
- Modify the Dynamic HTTP server …. String with something else

```
<html>
<head>
<title>This is really cool</title>
</head>
```

- Save the new file

Build a New Output File

- First Double click the batch file make.bat to build the image.

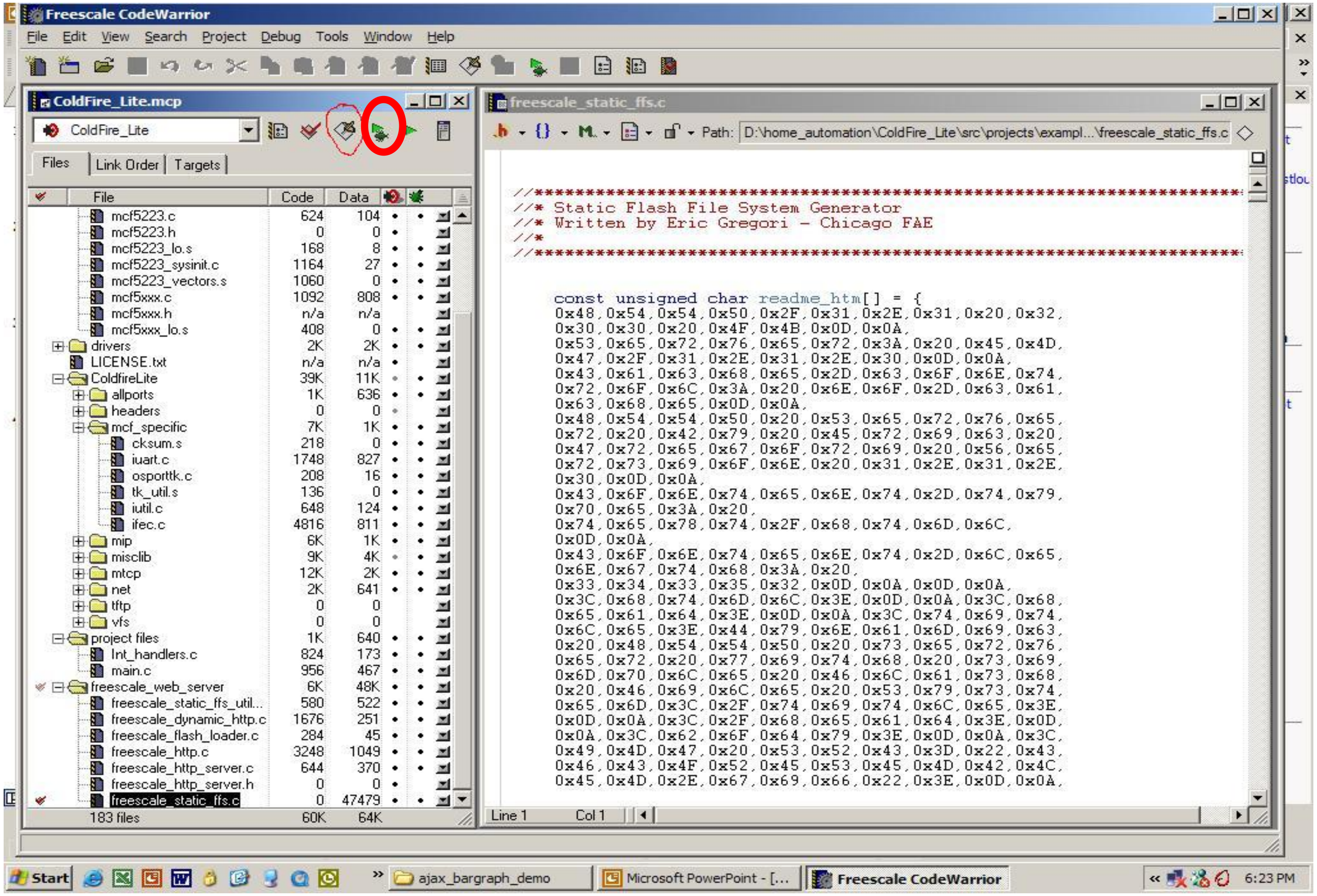


compile_time_loaded_web_page_example
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\ColdFire_Web_Server_with_Labs_051106\ColdFire_Lite\src\projects\example\compile
Go

Name	Size	Type	Date Modified
CFCORESEMBLEM.gif	13 KB	GIF Image	1/3/2006 12:47 PM
emg_static_ffs.exe	165 KB	Application	3/31/2006 9:29 PM
filelist.txt	1 KB	Text Document	3/29/2006 11:24 PM
make.bat	1 KB	MS-DOS Batch File	5/11/2006 10:41 AM
readme.htm	34 KB	HTML Document	4/9/2006 11:41 PM

Build the Project

Build the project by clicking on the MAKE icon (circled in **RED**)

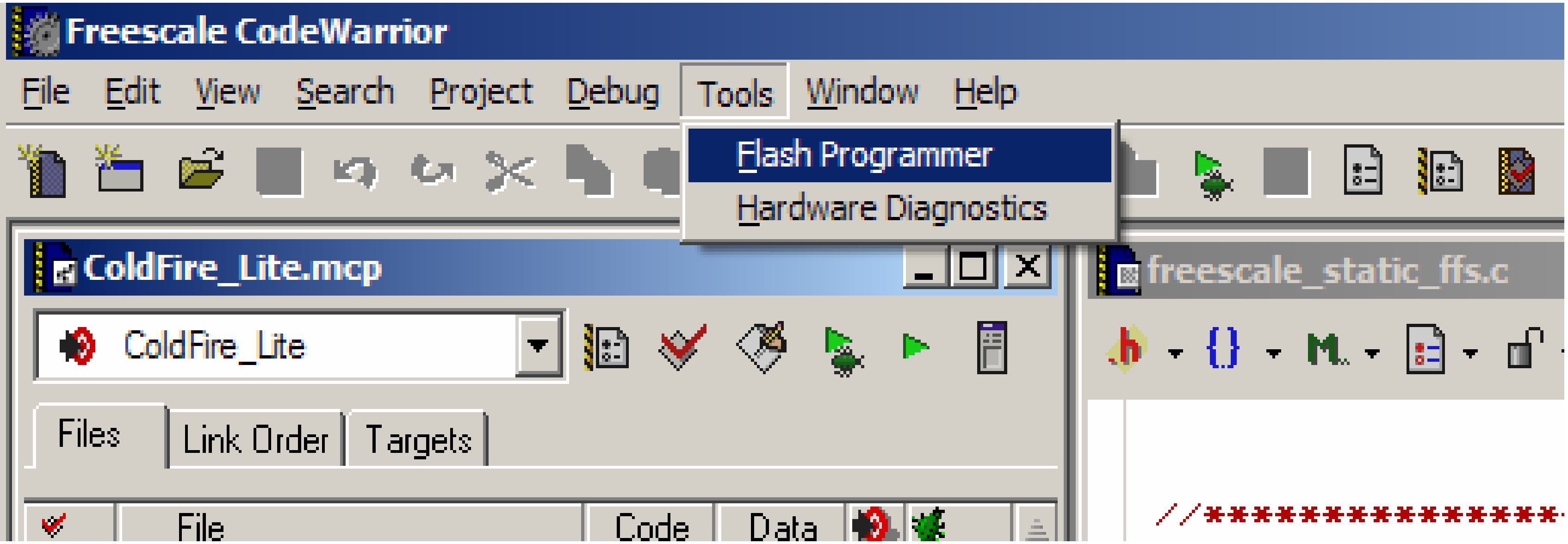

Freescale CodeWarrior
File Edit View Search Project Debug Tools Window Help
ColdFire_Lite.mcp
ColdFire_Lite
Files
Link Order
Targets
freescale_static_ffs.c
Path: D:\home_automation\ColdFire_Lite\src\projects\exampl...\freescale_static_ffs.c
Line 1
Col 1
Start
ajax_bargraph_demo
Microsoft PowerPoint - [...
Freescale CodeWarrior
6:23 PM

File	Code	Data		
mcf5223.c	624	104	●	●
mcf5223.h	0	0	●	
mcf5223_lo.s	168	8	●	●
mcf5223_sysinit.c	1164	27	●	●
mcf5223_vectors.s	1060	0	●	●
mcf5xxx.c	1092	808	●	●
mcf5xxx.h	n/a	n/a		
mcf5xxx_lo.s	408	0	●	●
drivers	2K	2K	●	●
LICENSE.txt	n/a	n/a	●	
ColdfireLite	39K	11K	●	●
allports	1K	636	●	●
headers	0	0	●	
mcf_specific	7K	1K	●	●
cksum.s	218	0	●	●
iuart.c	1748	827	●	●
osporttk.c	208	16	●	●
tk_util.s	136	0	●	●
iutil.c	648	124	●	●
ifec.c	4816	811	●	●
mip	6K	1K	●	●
misclib	9K	4K	●	●
mtcp	12K	2K	●	●
net	2K	641	●	●
tftp	0	0		
vfs	0	0		
project files	1K	640	●	●
Int_handlers.c	824	173	●	●
main.c	956	467	●	●
freescale_web_server	6K	48K	●	●
freescale_static_ffs_util...	580	522	●	●
freescale_dynamic_http.c	1676	251	●	●
freescale_flash_loader.c	284	45	●	●
freescale_http.c	3248	1049	●	●
freescale_http_server.c	644	370	●	●
freescale_http_server.h	0	0	●	
freescale_static_ffs.c	0	47479	●	●
183 files	60K	64K		

```
//***********************************************************************
//* Static Flash File System Generator
//* Written by Eric Gregori - Chicago FAE
//*
//***********************************************************************


    const unsigned char readme_htm[] = {
    0x48,0x54,0x54,0x50,0x2F,0x31,0x2E,0x31,0x20,0x32,
    0x30,0x30,0x20,0x4F,0x4B,0x0D,0x0A,
    0x53,0x65,0x72,0x76,0x65,0x72,0x3A,0x20,0x45,0x4D,
    0x47,0x2F,0x31,0x2E,0x31,0x2E,0x30,0x0D,0x0A,
    0x43,0x61,0x63,0x68,0x65,0x2D,0x63,0x6F,0x6E,0x74,
    0x72,0x6F,0x6C,0x3A,0x20,0x6E,0x6F,0x2D,0x63,0x61,
    0x63,0x68,0x65,0x0D,0x0A,
    0x48,0x54,0x54,0x50,0x20,0x53,0x65,0x72,0x76,0x65,
    0x72,0x20,0x42,0x79,0x20,0x45,0x72,0x69,0x63,0x20,
    0x47,0x72,0x65,0x67,0x6F,0x72,0x69,0x20,0x56,0x65,
    0x72,0x73,0x69,0x6F,0x6E,0x20,0x31,0x2E,0x31,0x2E,
    0x30,0x0D,0x0A,
    0x43,0x6F,0x6E,0x74,0x65,0x6E,0x74,0x2D,0x74,0x79,
    0x70,0x65,0x3A,0x20,
    0x74,0x65,0x78,0x74,0x2F,0x68,0x74,0x6D,0x6C,
    0x0D,0x0A,
    0x43,0x6F,0x6E,0x74,0x65,0x6E,0x74,0x2D,0x6C,0x65,
    0x6E,0x67,0x74,0x68,0x3A,0x20,
    0x33,0x34,0x33,0x35,0x32,0x0D,0x0A,0x0D,0x0A,
    0x3C,0x68,0x74,0x6D,0x6C,0x3E,0x0D,0x0A,0x3C,0x68,
    0x65,0x61,0x64,0x3E,0x0D,0x0A,0x3C,0x74,0x69,0x74,
    0x6C,0x65,0x3E,0x44,0x79,0x6E,0x61,0x6D,0x69,0x63,
    0x20,0x48,0x54,0x54,0x50,0x20,0x73,0x65,0x72,0x76,
    0x65,0x72,0x20,0x77,0x69,0x74,0x68,0x20,0x73,0x69,
    0x6D,0x70,0x6C,0x65,0x20,0x46,0x6C,0x61,0x73,0x68,
    0x20,0x46,0x69,0x6C,0x65,0x20,0x53,0x79,0x73,0x74,
    0x65,0x6D,0x3C,0x2F,0x74,0x69,0x74,0x6C,0x65,0x3E,
    0x0D,0x0A,0x3C,0x2F,0x68,0x65,0x61,0x64,0x3E,0x0D,
    0x0A,0x3C,0x62,0x6F,0x64,0x79,0x3E,0x0D,0x0A,0x3C,
    0x49,0x4D,0x47,0x20,0x53,0x52,0x43,0x3D,0x22,0x43,
    0x46,0x43,0x4F,0x52,0x45,0x53,0x45,0x4D,0x42,0x4C,
    0x45,0x4D,0x2E,0x67,0x69,0x66,0x22,0x3E,0x0D,0x0A,
```

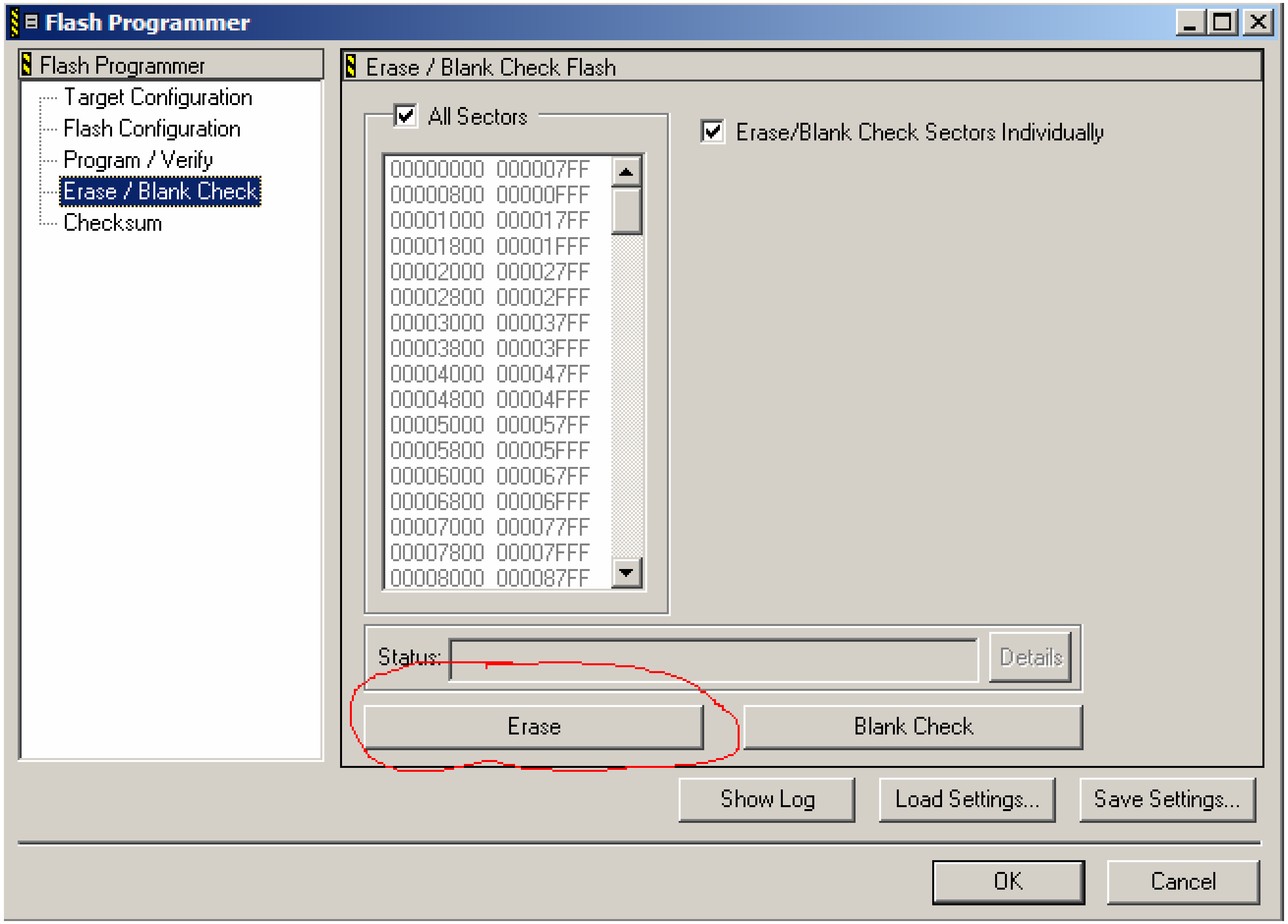
Start the Flash Programmer

Start the Flash Programmer by selecting the tools Flash Programmer Pull Down



Erase Flash

Erase Flash by selecting Erase/Blank Check, and clicking the Erase button. Watch the Status window for errors.



Flash Programmer

- Flash Programmer
  - Target Configuration
  - Flash Configuration
  - Program / Verify
  - Erase / Blank Check
  - Checksum

Erase / Blank Check Flash

☑ All Sectors

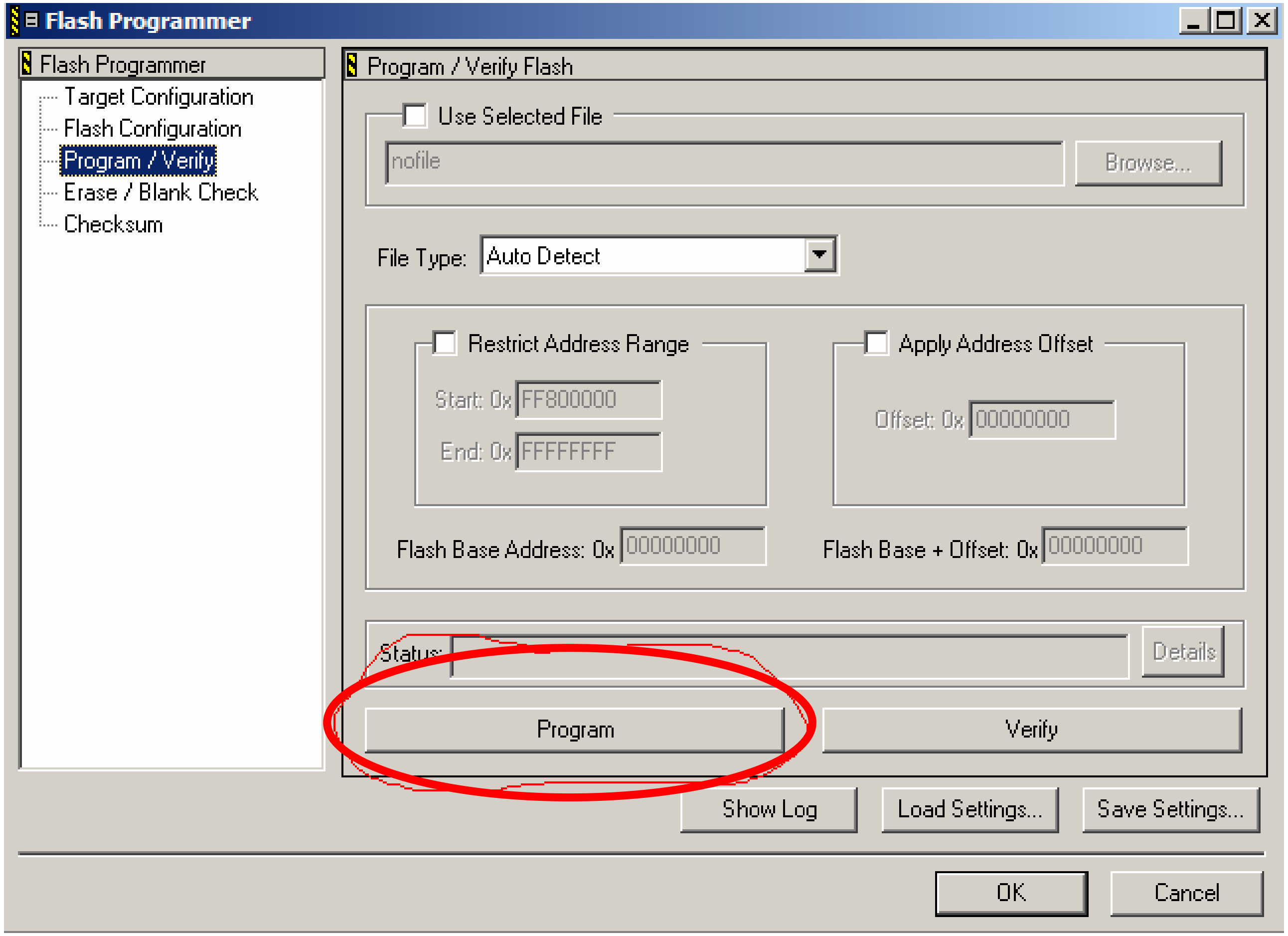
```
00000000  000007FF
00000800  00000FFF
00001000  000017FF
00001800  00001FFF
00002000  000027FF
00002800  00002FFF
00003000  000037FF
00003800  00003FFF
00004000  000047FF
00004800  00004FFF
00005000  000057FF
00005800  00005FFF
00006000  000067FF
00006800  00006FFF
00007000  000077FF
00007800  00007FFF
00008000  000087FF
```

☑ Erase/Blank Check Sectors Individually

Status:

Program

After the Erase is Complete, go to the Program/Verify window and click on the Program button.



Flash Programmer

Flash Programmer
- Target Configuration
- Flash Configuration
- Program / Verify
- Erase / Blank Check
- Checksum

Program / Verify Flash

□ Use Selected File

nofile

File Type: Auto Detect

□ Restrict Address Range

Start: 0x FF800000

End: 0x FFFFFFFF

□ Apply Address Offset

Offset: 0x 00000000

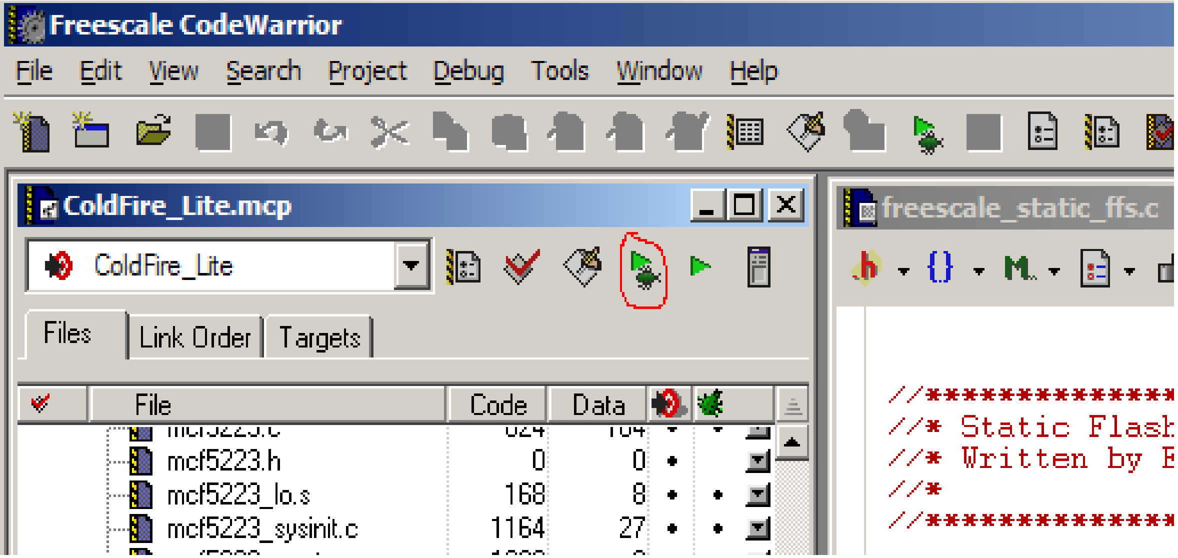
Flash Base Address: 0x 00000000

Flash Base + Offset: 0x 00000000

Status:

Run

Click on the Run icon, circled in RED below. This will execute the code in flash. If you have an external power supply, you could also disconnect the USB from the board and hit reset.

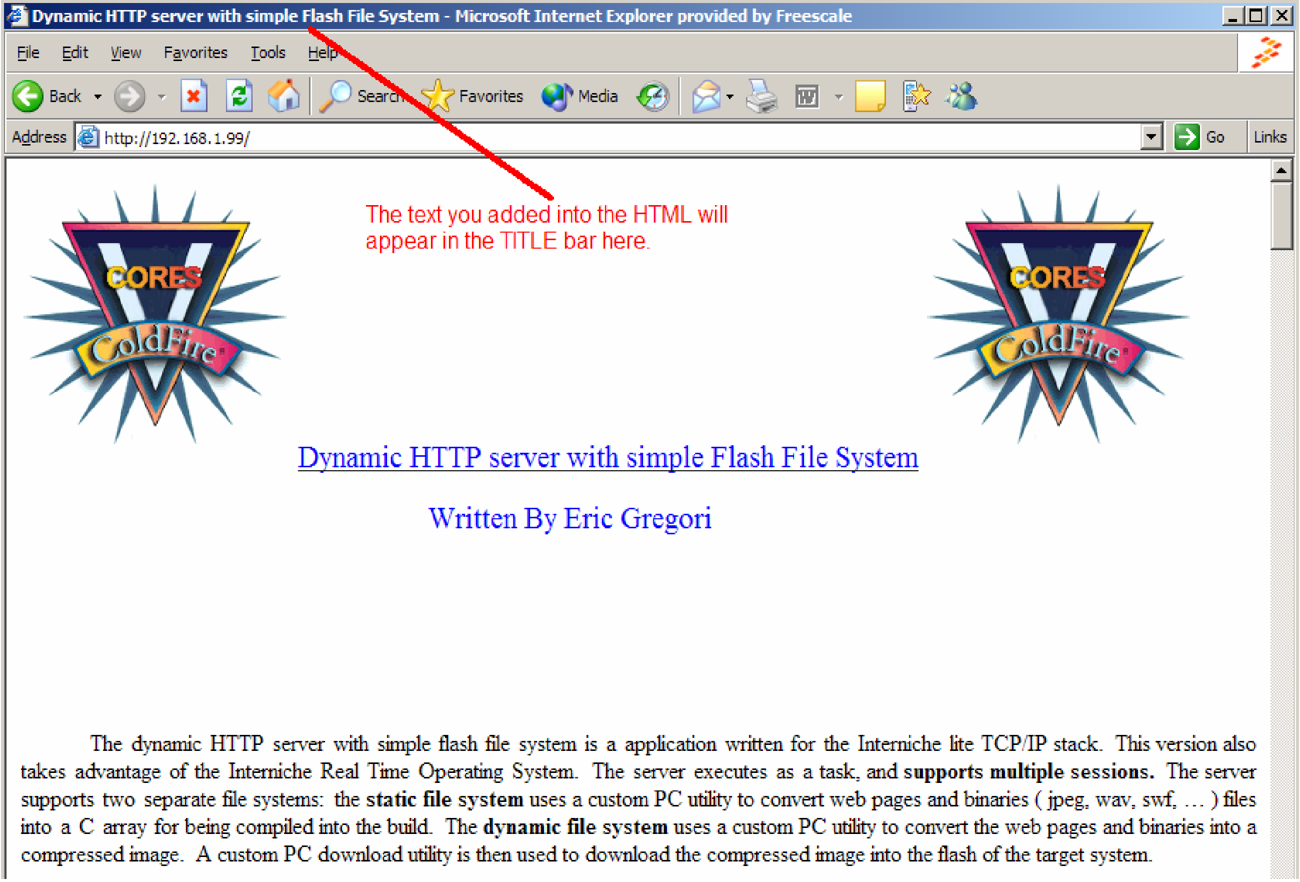


File	Code	Data
mcf5223.h	0	0
mcf5223_lo.s	168	8
mcf5223_sysinit.c	1164	27

```
//***************
//* Static Flash
//* Written by F
//*
//***************
```

Default Web Page

Open Internet Explorer, and type 192.168.1.99 (the IP address of the demo board) into the address bar. This is the default compile time web page you just loaded with the TCP/IP stack and Web Server.



Dynamic HTTP server with simple Flash File System - Microsoft Internet Explorer provided by Freescale
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Address
http://192.168.1.99/
Go
Links
CORES
ColdFire

The text you added into the HTML will appear in the TITLE bar here.

Dynamic HTTP server with simple Flash File System

Written By Eric Gregori

The dynamic HTTP server with simple flash file system is a application written for the Interniche lite TCP/IP stack. This version also takes advantage of the Interniche Real Time Operating System. The server executes as a task, and **supports multiple sessions.** The server supports two separate file systems: the **static file system** uses a custom PC utility to convert web pages and binaries (jpeg, wav, swf, …) files into a C array for being compiled into the build. The **dynamic file system** uses a custom PC utility to convert the web pages and binaries into a compressed image. A custom PC download utility is then used to download the compressed image into the flash of the target system.

Run Time Loadable Web Pages

- Web Pages can be uploaded via Ethernet at run time.
- Web Pages can be loaded over and over again. # of re-loads only limited by # of writes to flash.
- Loaded Web Pages take priority over default or Compile Time Web Pages.
- Loaded Web Pages are protected with a 32 character password string.

Build and Loading a Run Time Loadable Web Page

- A single Batch file is used to both build and load the Web Page.
- Within the Batch file are calls to two executable.
- The first executable: emg_dynamic_ffs.exe

  Compresses the Web Pages into a binary, and adds a File Allocation Table (FAT) to the top of the file. The firmware in the Web Server uses the FAT to reference the data in the file from within the binary image.

Emg_dynamic_FFS.exe

Emg_dynamic_ffs filelist.txt output_file.ffs

Where:

- Filelist.txt is a text file containing the list of files to compress. Each file should be on its own line, and the first file is the default.
- Comments can be added using a ‘*’ as the first character in a line.

- Output_file.ffs is the file generated containing all the files in the filelist compressed together, along with File Allocation Table used to reference the files from the Web Server.

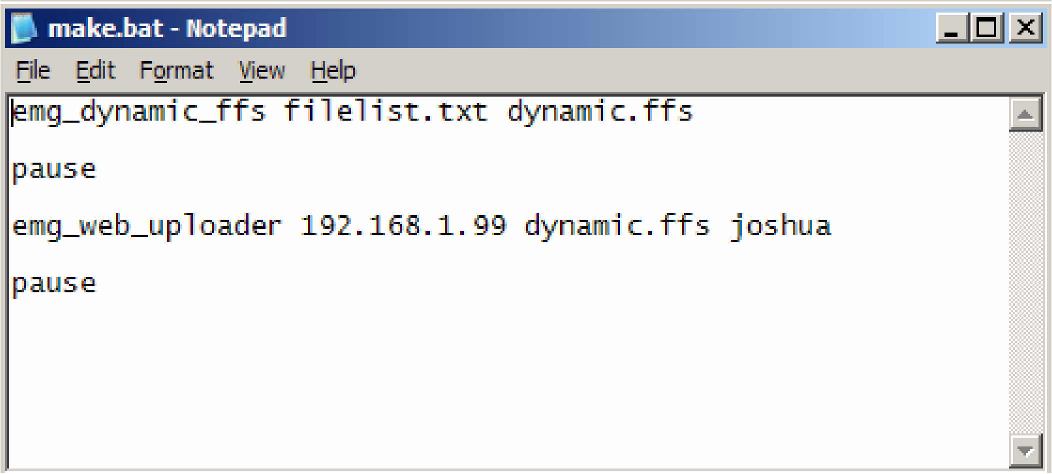
Emg_web_uploader.exe

Emg_web_uploader ip_address filename.ffs key_string

Where:

- Ip_address is the ip address of the hardware (192.168.1.99) in examples.
- Filename.ffs is the file generated by the emg_dynamic_ffs utility.
- Key_string is the 32 character key used to unlock the flash file system (joshua) in examples.

The make.bat File



```
emg_dynamic_ffs filelist.txt dynamic.ffs

pause

emg_web_uploader 192.168.1.99 dynamic.ffs joshua

pause
```

The filelist.txt file lists the files that will be included in the FFS.

Dynamic.ffs is the binary image containing all the files and the FAT.

Pause is a DOS command to prompt the user to hit any key.

192.168.1.99 is the IP address of the hardware for these examples.

Joshua is the key string for these examples.

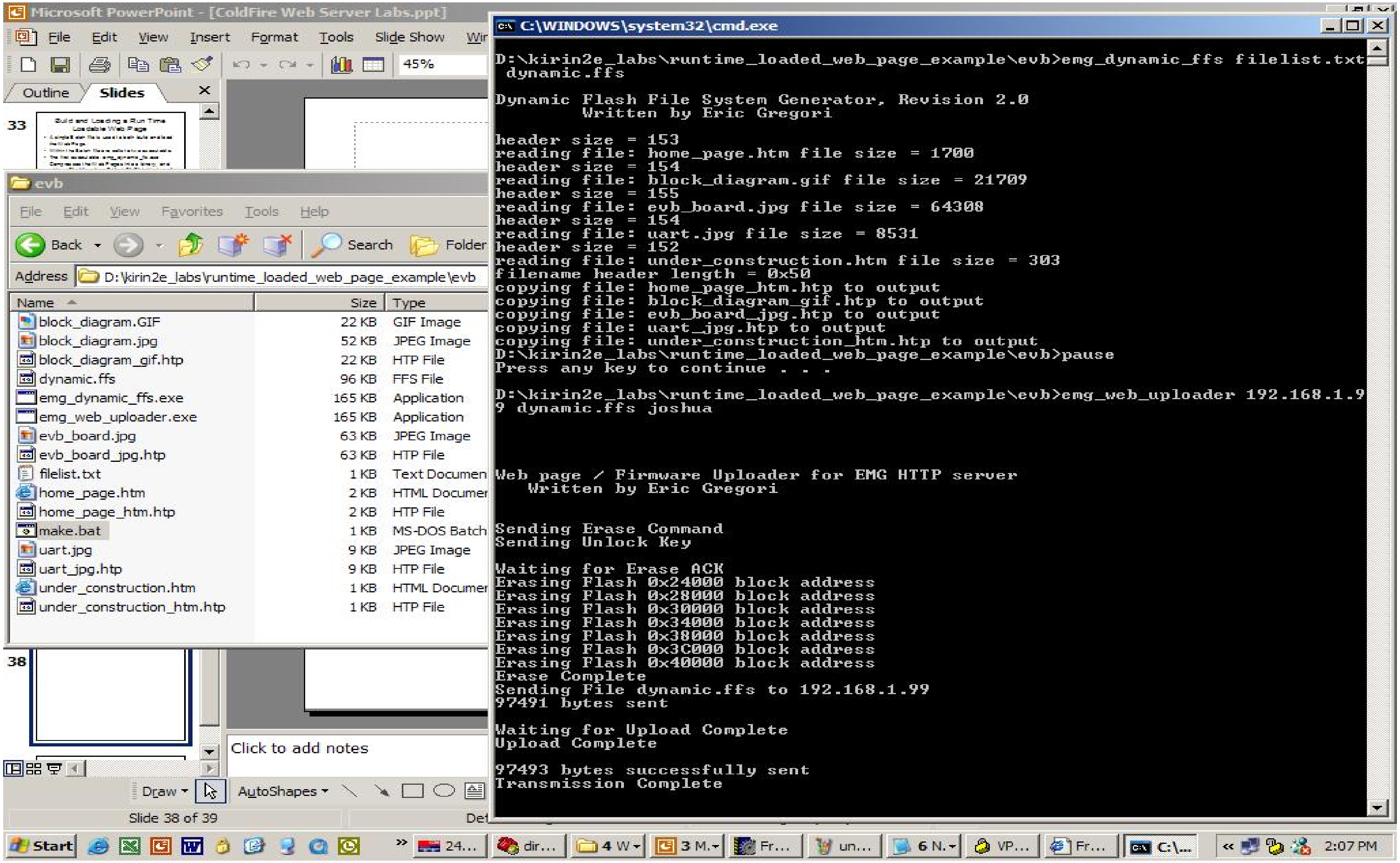
LAB 1: Building and Loading a Run Time Web Page Image

Go to your project directory and find:

runtime_loaded_web_page_example\LAB1_?????

- Be sure the demo board is powered up and plugged into Ethernet.
- Double click the batch file.
- Hit any key when prompted.
- Wait for download to complete.
- Go to browser and load page.

LAB 1: Building and Loading a Run Time Web Page Image

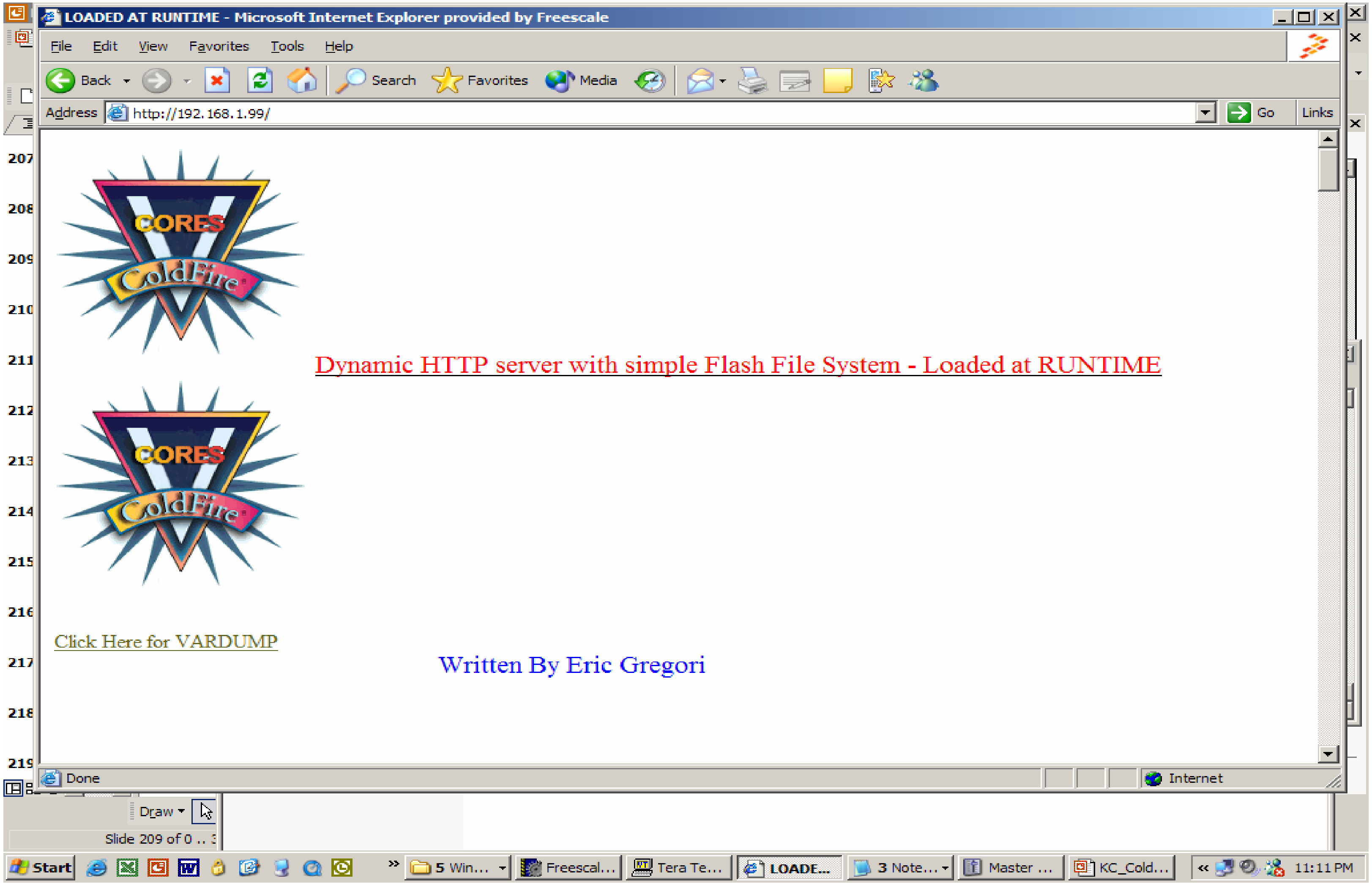


Microsoft PowerPoint - [ColdFire Web Server Labs.ppt]
File Edit View Insert Format Tools Slide Show
45%
Outline Slides
33
38
Click to add notes
Draw AutoShapes
Slide 38 of 39
evb
File Edit View Favorites Tools Help
Back Search Folder
Address D:\kirin2e_labs\runtime_loaded_web_page_example\evb
C:\WINDOWS\system32\cmd.exe
Start
2:07 PM

Name	Size	Type
block_diagram.GIF	22 KB	GIF Image
block_diagram.jpg	52 KB	JPEG Image
block_diagram_gif.htp	22 KB	HTP File
dynamic.ffs	96 KB	FFS File
emg_dynamic_ffs.exe	165 KB	Application
emg_web_uploader.exe	165 KB	Application
evb_board.jpg	63 KB	JPEG Image
evb_board_jpg.htp	63 KB	HTP File
filelist.txt	1 KB	Text Documen
home_page.htm	2 KB	HTML Documer
home_page_htm.htp	2 KB	HTP File
make.bat	1 KB	MS-DOS Batch
uart.jpg	9 KB	JPEG Image
uart_jpg.htp	9 KB	HTP File
under_construction.htm	1 KB	HTML Documer
under_construction_htm.htp	1 KB	HTP File

```
D:\kirin2e_labs\runtime_loaded_web_page_example\evb>emg_dynamic_ffs filelist.txt
 dynamic.ffs

Dynamic Flash File System Generator, Revision 2.0
        Written by Eric Gregori

header size = 153
reading file: home_page.htm file size = 1700
header size = 154
reading file: block_diagram.gif file size = 21709
header size = 155
reading file: evb_board.jpg file size = 64308
header size = 154
reading file: uart.jpg file size = 8531
header size = 152
reading file: under_construction.htm file size = 303
filename header length = 0x50
copying file: home_page_htm.htp to output
copying file: block_diagram_gif.htp to output
copying file: evb_board_jpg.htp to output
copying file: uart_jpg.htp to output
copying file: under_construction_htm.htp to output
D:\kirin2e_labs\runtime_loaded_web_page_example\evb>pause
Press any key to continue . . .

D:\kirin2e_labs\runtime_loaded_web_page_example\evb>emg_web_uploader 192.168.1.9
9 dynamic.ffs joshua



Web page / Firmware Uploader for EMG HTTP server
   Written by Eric Gregori


Sending Erase Command
Sending Unlock Key

Waiting for Erase ACK
Erasing Flash 0x24000 block address
Erasing Flash 0x28000 block address
Erasing Flash 0x30000 block address
Erasing Flash 0x34000 block address
Erasing Flash 0x38000 block address
Erasing Flash 0x3C000 block address
Erasing Flash 0x40000 block address
Erase Complete
Sending File dynamic.ffs to 192.168.1.99
97491 bytes sent

Waiting for Upload Complete
Upload Complete

97493 bytes successfully sent
Transmission Complete
```

LAB 1: Building and Loading a Run Time Web Page Image



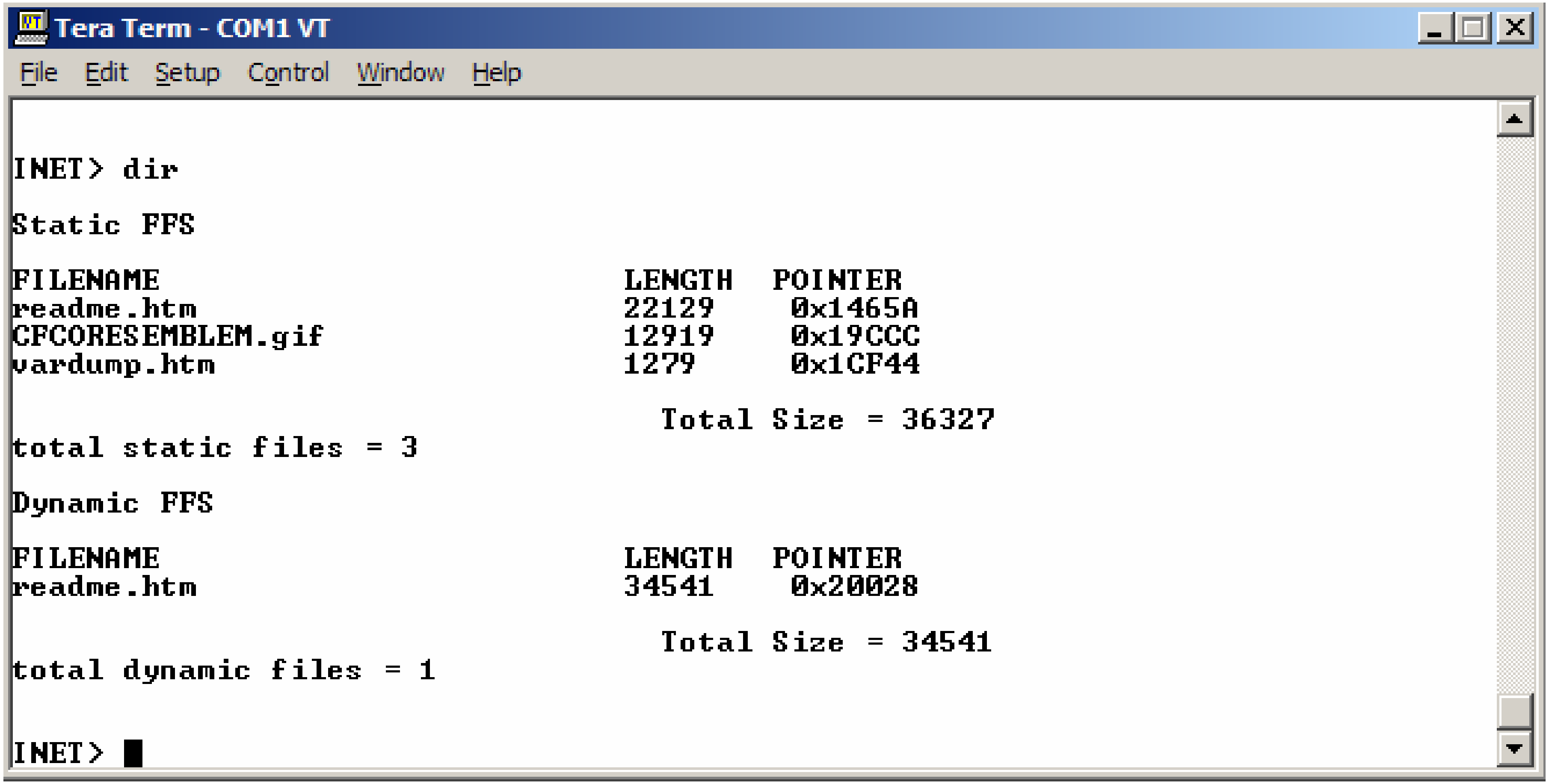
Dynamic HTTP server with simple Flash File System - Loaded at RUNTIME

Click Here for VARDUMP

Written By Eric Gregori

LAB 1: Building and Loading a Run Time Web Page Image

At the Serial INET> prompt type ‘dir’



```
INET> dir

Static FFS

FILENAME                         LENGTH   POINTER
readme.htm                       22129     0x1465A
CFCORESEMBLEM.gif                12919     0x19CCC
vardump.htm                      1279      0x1CF44

                                   Total Size = 36327
total static files = 3

Dynamic FFS

FILENAME                         LENGTH   POINTER
readme.htm                       34541     0x20028

                                   Total Size = 34541
total dynamic files = 1


INET> █
```

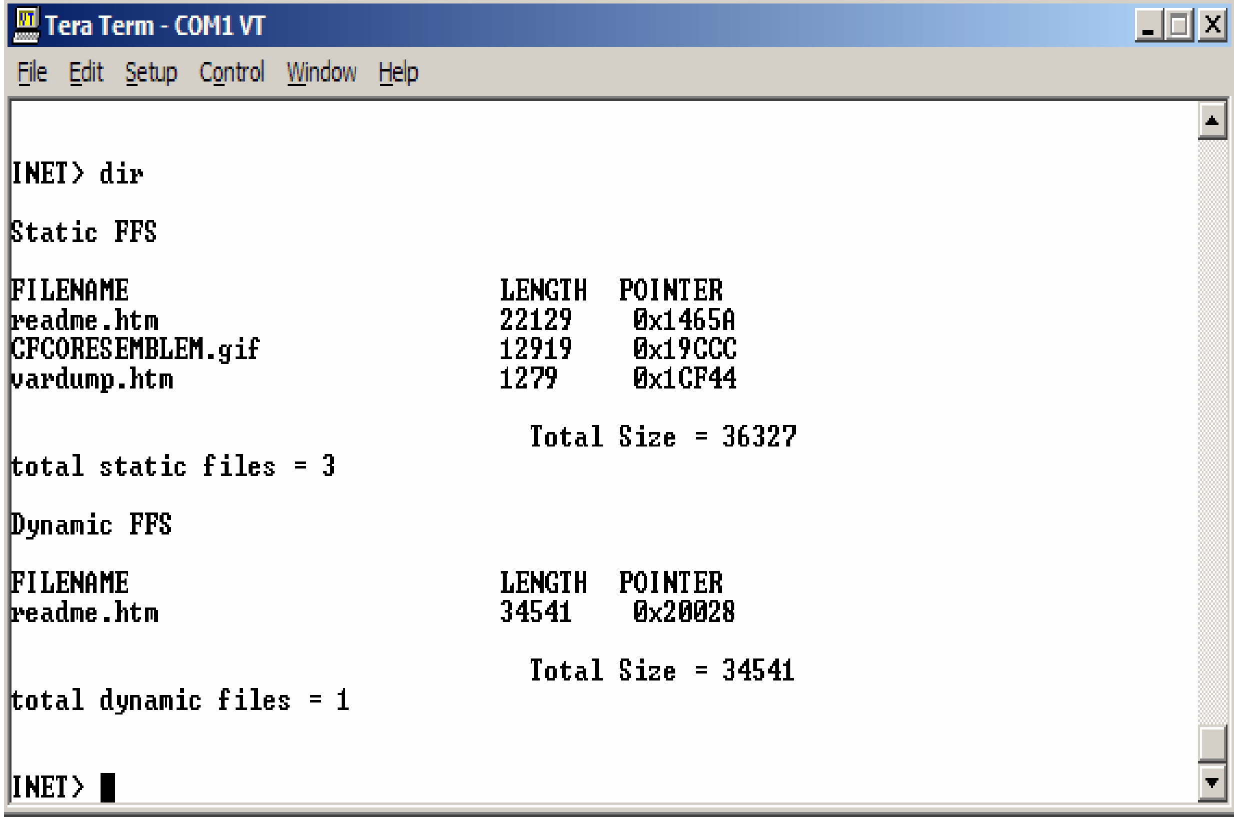
Demo Board Directory

Let take a look at the contents in the directory of the demo board

Notice, the static file system (compile time) still contains files.

When the dynamic (run time) file system is loaded with a binary image, it takes priority over the static file system.

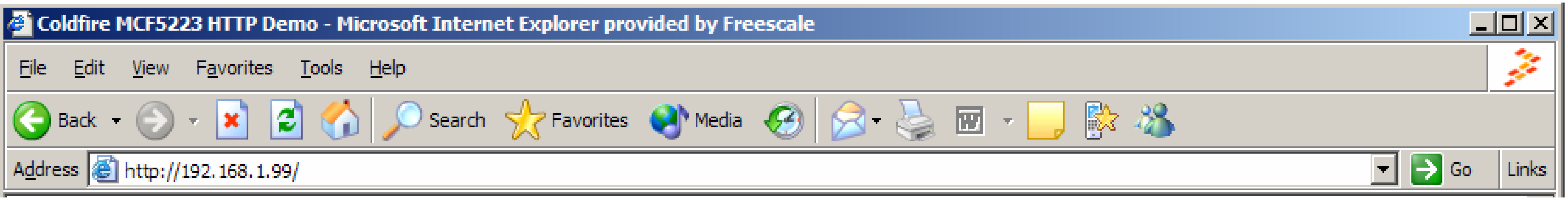
Other files in the static FFS are still available.



```
INET> dir

Static FFS

FILENAME                         LENGTH  POINTER
readme.htm                       22129    0x1465A
CFCORESEMBLEM.gif                12919    0x19CCC
vardump.htm                      1279     0x1CF44

                                   Total Size = 36327
total static files = 3

Dynamic FFS

FILENAME                         LENGTH  POINTER
readme.htm                       34541    0x20028

                                   Total Size = 34541
total dynamic files = 1


INET> 
```

Web Server Defaults

Notice what we entered at the address bar. No filename is specified. When no filename is specified the Web Server defaults to the first file listed in the file system.


Coldfire MCF5223 HTTP Demo - Microsoft Internet Explorer provided by Freescale
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Go
Links

Address http://192.168.1.99/

* emg dynamic web page description file
* The files listed below will be concantenated into a
* single compressed downloadable image.

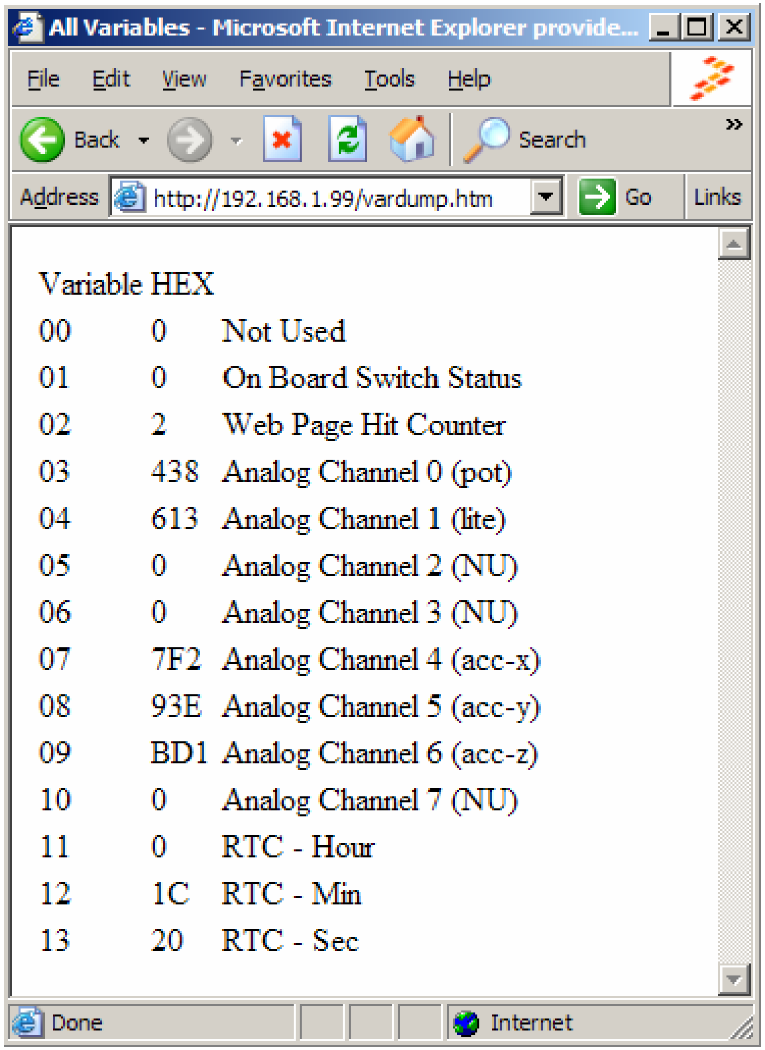
* The first file in the list is the default file

Readme.htm.htm ← This is the file that is loaded by default.
CFCORESEMBLEM.gif
vardump.htm

Going Direct to a File Using the Browser

To go directly to a file in the FFS from the browser, just include the name of the file after the '/' in the IP address.

Notice Vardump.htm is in the static file system, but is still available after loading a dynamic FFS.


All Variables - Microsoft Internet Explorer provide...
File Edit View Favorites Tools Help
Back
Search
Address
Go
Links
Done
Internet

http://192.168.1.99/vardump.htm

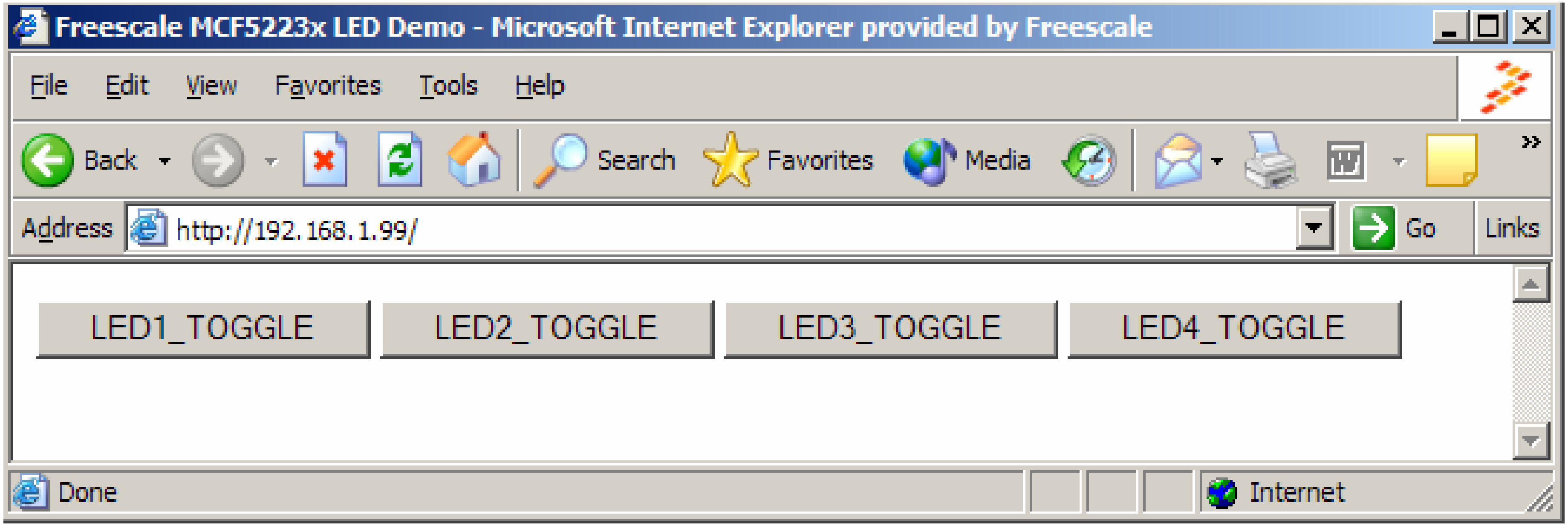
Variable	HEX	
00	0	Not Used
01	0	On Board Switch Status
02	2	Web Page Hit Counter
03	438	Analog Channel 0 (pot)
04	613	Analog Channel 1 (lite)
05	0	Analog Channel 2 (NU)
06	0	Analog Channel 3 (NU)
07	7F2	Analog Channel 4 (acc-x)
08	93E	Analog Channel 5 (acc-y)
09	BD1	Analog Channel 6 (acc-z)
10	0	Analog Channel 7 (NU)
11	0	RTC - Hour
12	1C	RTC - Min
13	20	RTC - Sec

LAB 2: Sending Commands to the Server Using HTML Forms

- HTML INPUT methods and BUTTON methods use a form to request data from the server.
- The form adds assignment information to the filename in the GET request.
- Form data is pre-pended with a ‘?’.
- The form data can be used to control hardware, or change parameters on the server.

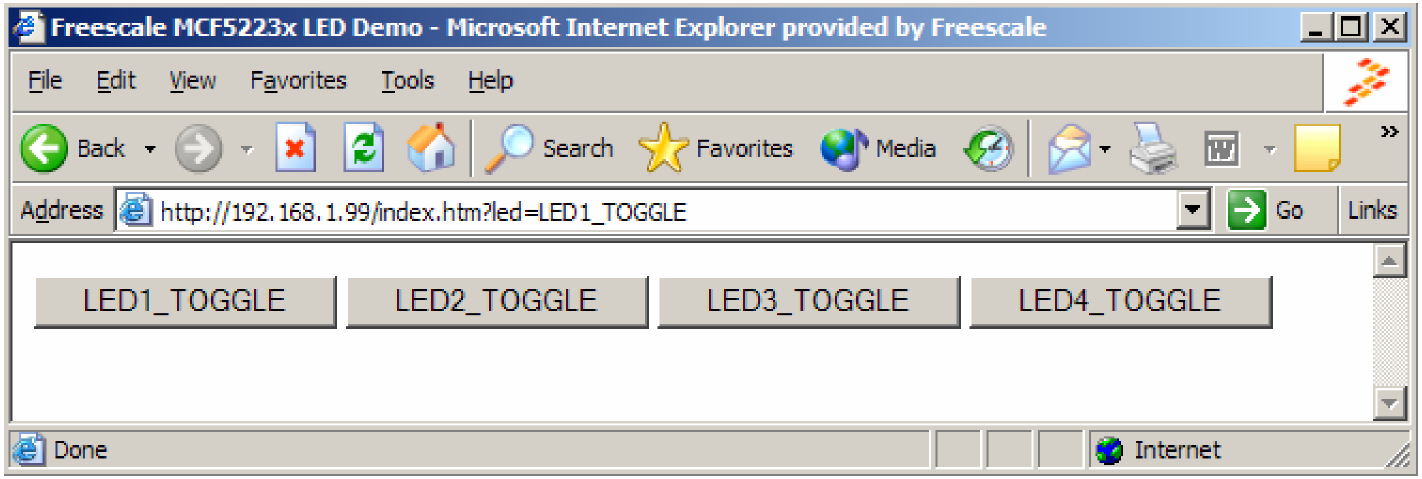
Loading the LED Control Demo Run Time Lab

- Load the demo in the board by double clicking on the make.bat file in the led_control_demo directory.
- Load the web page by typing 192.168.1.99 in the web browser address bar.
- Clicking on the buttons will toggle led's on the demo board.


Freescale MCF5223x LED Demo - Microsoft Internet Explorer provided by Freescale

Address http://192.168.1.99/

How Form Data is Sent to the Server


Freescale MCF5223x LED Demo - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Links
LED1_TOGGLE
LED2_TOGGLE
LED3_TOGGLE
LED4_TOGGLE
Done
Internet

http://192.168.1.99/index.htm?led=LED1_TOGGLE

Notice what happens when the LED1_TOGGLE button is clicked.

The browser sends a request to the server for the filename:

Index.htm?led=LED1_TOGGLE

The ‘?’ tells the server that the string following is a form assignment.

Multiple form assignments can reside one after another seperated by ‘?’.

The web server includes a form assignment parser with functions to handle the led variable.

The LED Demo HTML

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Freescale MCF5223x LED Demo</title>
</head>

<body>

<form action="/index.htm">
<TABEL BORDER=0 CELLSPACING=0 CELLPADDING=0 width=300>
<tr>
<td><input type=submit name=led value=LED1_TOGGLE></td>
<td><input type=submit name=led value=LED2_TOGGLE></td>
<td><input type=submit name=led value=LED3_TOGGLE></td>
<td><input type=submit name=led value=LED4_TOGGLE></td>
</tr>
>/form>
</body>
</html>
```

Notice the <FORM ACTION=“/index.htm”> command. This tells the web browser to load the file Index.htm anytime a submit occurs within the FORM command.

The INPUT methods tell the web browser to send form commands led=LED1_TOGGLE when the LED1_TOGGLE button is clicked.

The WEB Server Firmware - Processing a FORM Submit

- The Web Server detects the form by the ‘?’ in the filename.
- The FORM is then parsed into the two parts, the NAME and the VALUE.
- The NAME is on the left of the ‘=‘ sign, the VALUE on the right.
- The Name is used to call the function “LED”, and pass it the VALUE.*

LAB 3:Serial Forms

- What if you wanted to control a device connected to the serial port of the demo board.
- Serial forms allows you to do just that.
- You can pass a serial string from a web page to the serial port on the ColdFire.

- Serial forms use a internal form called serial.

The HTML to use serial forms

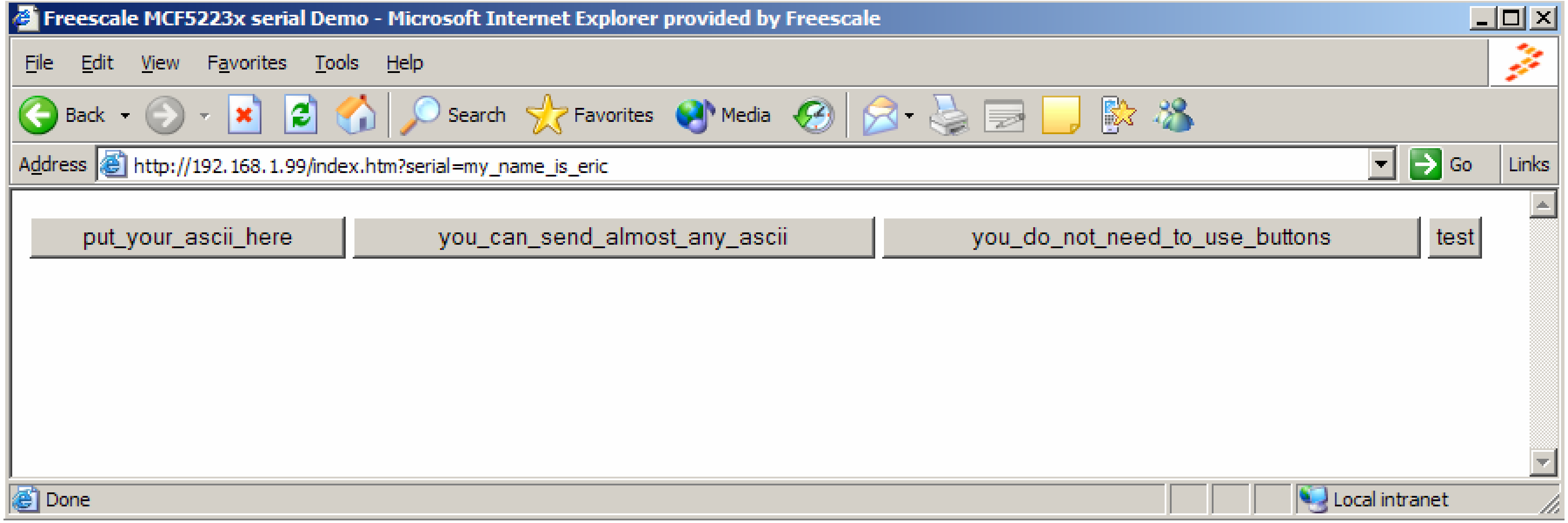
The serial string is embedded as the value of the form.
The form name must be “serial”.

```
<form action="/index.htm">
<TABEL BORDER=0 CELLSPACING=0 CELLPADDING=0 width=300>
<tr>
<td><input type=submit name=serial value=put_your_ascii_here></td>
<td><input type=submit name=serial
   value=you_can_send_almost_any_ascii></td>
<td><input type=submit name=serial
   value=you_do_not_need_to_use_buttons></td>
<td><input type=submit name=serial value=test></td>
</tr>
</body>
</html>
```

Direct access to serial port

- You can also send commands to the serial port by simply typing the serial form directly into the address bar, or from a href in HTML.

- The following will send “my_name_is_eric” to the serial port.



Freescale MCF5223x serial Demo - Microsoft Internet Explorer provided by Freescale

Address http://192.168.1.99/index.htm?serial=my_name_is_eric

Dynamic HTML Tokens

- Dynamic HTML Tokens allow variable content like sensor data to be inserted into web pages, no programming required.
- Just insert the token ~IIF; into your HTML, and the token will be replaced with the data referenced by II.
- Conditional tokens take the idea one step further, by allowing whole HTML strings to be replaced based a data comparison to a constant.

The REPLACE Token ~IIF;

Where:

- II = The decimal variable index to read the data.
  - The variable array contains 32 longwords (can be as high as 99).

- F = The format to display the data (D = Decimal, H = Hex).

Example:

```
<HTML>
<HEAD>
<TITLE>This text will appear at the top of the web browser, the navigation bar</TITLE>
</HEAD>
<BODY>
<CENTER>You have opened this page ~02D; times</CENTER>
</BODY>
</HTML>
```

The Variable index 02 is the web page hit counter.

The CONDITIONAL Token ^II>C|true|false|;

Where:

- II = The decimal variable index to read the data.
  The variable array contains 32 longwords (can be as high as 99)
- C = Hex value for comparison.
- > = Variable value greater then C
- = = Variable value equal to C
- & = Variable value and C
- ! = Variable not equal to C

- “true” = ascii string to replace if condition is true
- “false” = ascii string to replace if condition is false

LAB 4: Using Dynamic HTML with Tokens

- Go to the dynamic_html_with_tokens directory, and double click on the make.bat file. This loads the firmware into the eval board.
- Adjust the POT on the demo board. Notice the data in variable index 03 is changing. Notice that the font turns red when the variable goes above 0x0800.
- Push SW1 and SW2. Notice the status of the switches appears next variable index 03.

The Variable Array

Index	Parameter
00	Available to user
01	On Board Switch Status
02	Web Page Hit Counter
03	Analog Channel 0 (pot)
04	Analog Channel 1 (lite)
05	Analog Channel 2 (NU)
06	Analog Channel 3 (NU)
07	Analog Channel 4 (acc-x)
08	Analog Channel 5 (acc-y)
09	Analog Channel 6 (acc-z)
10	Analog Channel 7 (NU)
11	RTC - Hour
12	RTC - Min
13	RTC - Sec
14	Available to user
15	Available to user
16	Available to user
17	Available to user
18	Available to user
19	Available to user
20	Available to user
21	Available to user
22	Available to user
23	Available to user
24	Available to user
25	Available to user
26	Available to user
27	Available to user
28	Available to user
29	Available to user
30	Available to user
31	Available to user

The Serial Interface

- Notice the “Available To User” entries in the variable array.
- You can modify the ‘C’ code for the Web Server to assign any 32 bit value you want to a available position in the variable array.
- Or, you can use the serial interface to modify the variable in the array.
- The serial interface method is designed for interfacing to other embedded systems.
- The serial port supports autobaud, so it will automatically sync to the baud of your embedded device.

Using the Serial Interface- The ‘VAR’ command

```
INET> var

Dynamic HTML variable dump
Variable 0 = 12345678   BC614E
Variable 1 = 0   0
Variable 2 = 1035   40B
Variable 3 = 2202   89A
Variable 4 = 2205   89D
Variable 5 = 0   0
Variable 6 = 0   0
Variable 7 = 2435   983
Variable 8 = 387   183
Variable 9 = 3125   C35
Variable 10 = 0   0
Variable 11 = 23   17
Variable 12 = 26   1A
Variable 13 = 56   38
Variable 14 = 99   63

INET>
```

VAR Command Parameters

- var – Dumps the contents of the array to the serial port.
- Var 14 – Dumps the contents of variable index 14.
- Var 14, 12345678 – Assigns 12345678 decimal to variable index 14.

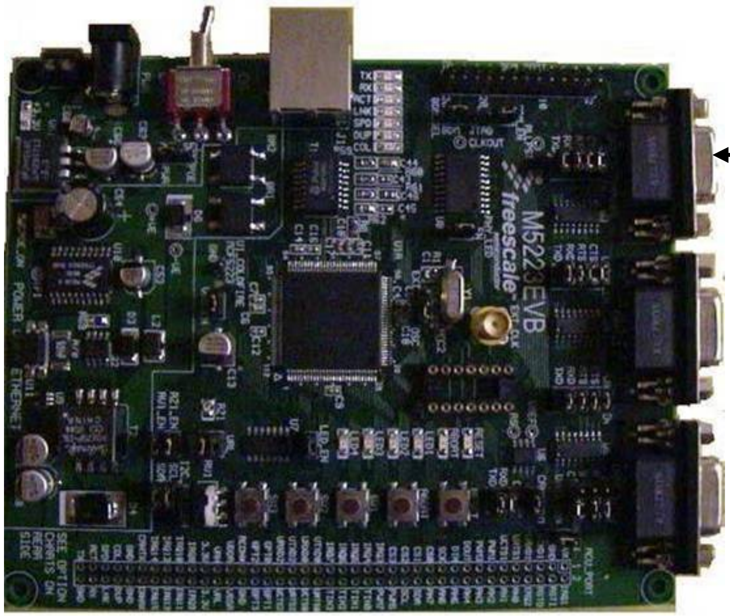
The ‘var x’ Command

```
INET> var 0
Variable 0 = 12345678   BC614E
INET> var 2
Variable 2 = 1195   4AB
INET> var 3
Variable 3 = 2202   89A
INET> var 4
Variable 4 = 2275   8E3
INET>
```

Assigning a Variable with the ‘var’ Command var II, decimal_data

```
INET> var 14, 100


INET> var 14

Variable 14 = 100   64

INET> var 14,250


INET> var 14

Variable 14 = 250   FA

INET> var 14, 900


INET> var 14

Variable 14 = 900   384

INET>
```

How to Use the VAR Command



115Kbaud, 8,n,1

Zigbee Coordinator

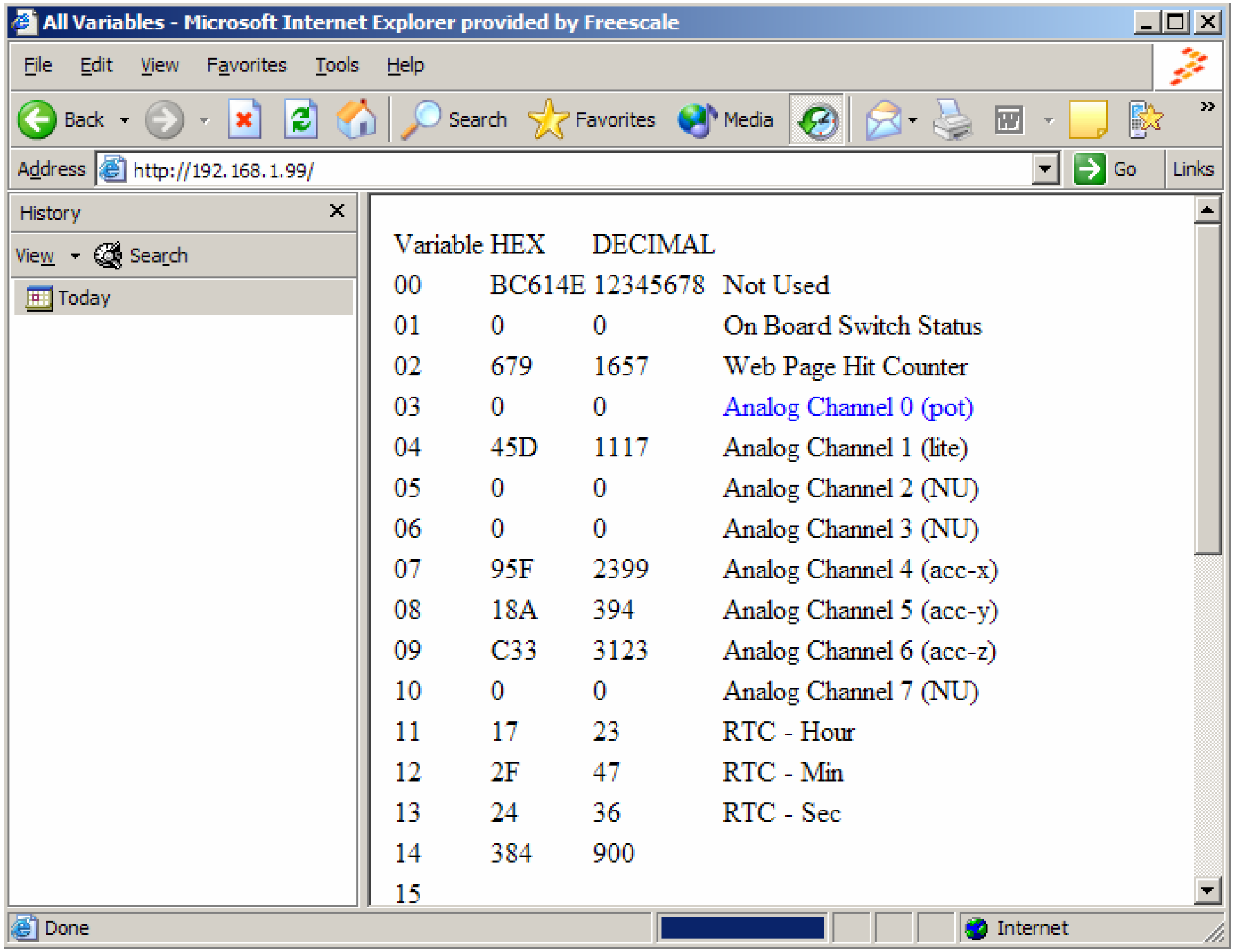
The Zigbee Coordinator collects data from its sensors, then converts it into ‘VAR’ commands. Each sensor is given a separate variable index.

The ‘VAR’ command is terminated with a CR, the INET> prompt provides software handshaking.

The HTML Code

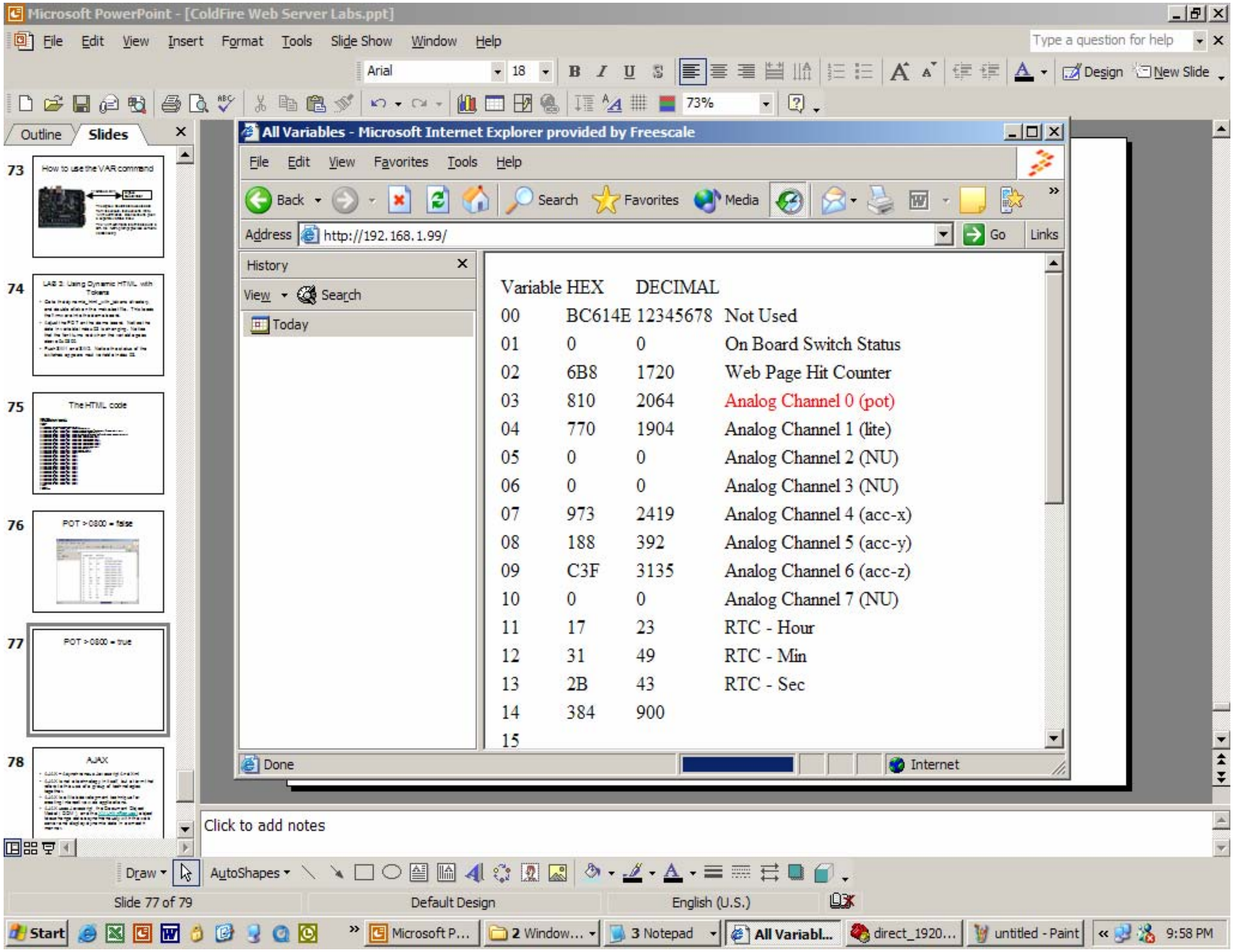
```
<html><head>
<meta http-equiv="refresh" content="1">
<title>All Variables</title></head><body>
<TABLE>
<tbody>
<tr><td>Variable</td><td>HEX</td><td>DECIMAL</td></tr>
<tr><td>00</td><td>~00H    ;</td><td>~00D    ;</td><td>Not Used</td></tr>
<tr><td>01</td><td>~01H    ;</td><td>~01D    ;</td><td>On Board Switch Status ^01&0001|SW1||;^01&0008|SW2||;</td></tr>
<tr><td>02</td><td>~02H    ;</td><td>~02D    ;</td><td>Web Page Hit Counter</td></tr>
<tr><td>03</td><td>~03H    ;</td><td>~03D    ;</td><td><FONT COLOR=^03>0800|"RED"|"BLUE"|;>Analog Channel 0 (pot)</td></tr>
<tr><td>04</td><td>~04H    ;</td><td>~04D    ;</td><td>Analog Channel 1 (lite)</td></tr>
<tr><td>05</td><td>~05H    ;</td><td>~05D    ;</td><td>Analog Channel 2 (NU)</td></tr>
<tr><td>06</td><td>~06H    ;</td><td>~06D    ;</td><td>Analog Channel 3 (NU)</td></tr>
<tr><td>07</td><td>~07H    ;</td><td>~07D    ;</td><td>Analog Channel 4 (acc-x)</td></tr>
<tr><td>08</td><td>~08H    ;</td><td>~08D    ;</td><td>Analog Channel 5 (acc-y)</td></tr>
<tr><td>09</td><td>~09H    ;</td><td>~09D    ;</td><td>Analog Channel 6 (acc-z)</td></tr>
<tr><td>10</td><td>~10H    ;</td><td>~10D    ;</td><td>Analog Channel 7 (NU)</td></tr>
<tr><td>11</td><td>~11H    ;</td><td>~11D    ;</td><td>RTC - Hour</td></tr>
<tr><td>12</td><td>~12H    ;</td><td>~12D    ;</td><td>RTC - Min </td></tr>
<tr><td>13</td><td>~13H    ;</td><td>~13D    ;</td><td>RTC - Sec </td></tr>
<tr><td>14</td><td>~14H    ;</td><td>~14D    ;</td></tr>
<tr><td>15</td><td>~15H    ;</td><td>~15D    ;</td></tr>
<tr><td>16</td><td>~16H    ;</td><td>~16D    ;</td></tr>
<tr><td>17</td><td>~17H    ;</td><td>~17D    ;</td></tr>
<tr><td>18</td><td>~18H    ;</td><td>~18D    ;</td></tr>
<tr><td>19</td><td>~19H    ;</td><td>~19D    ;</td></tr>
<tr><td>20</td><td>~20H    ;</td><td>~20D    ;</td></tr>
<tr><td>21</td><td>~21H    ;</td><td>~21D    ;</td></tr>
<tr><td>22</td><td>~22H    ;</td><td>~22D    ;</td></tr>
<tr><td>23</td><td>~23H    ;</td><td>~23D    ;</td></tr>
<tr><td>24</td><td>~24H    ;</td><td>~24D    ;</td></tr>
<tr><td>25</td><td>~25H    ;</td><td>~25D    ;</td></tr>
<tr><td>26</td><td>~26H    ;</td><td>~26D    ;</td></tr>
<tr><td>27</td><td>~27H    ;</td><td>~27D    ;</td></tr>
<tr><td>28</td><td>~28H    ;</td><td>~28D    ;</td></tr>
<tr><td>29</td><td>~29H    ;</td><td>~29D    ;</td></tr>
<tr><td>30</td><td>~30H    ;</td><td>~30D    ;</td></tr>
<tr><td>31</td><td>~31H    ;</td><td>~31D    ;</td></tr>
</tbody>
</TABLE>
</body></html>
```

POT > 0800 = false



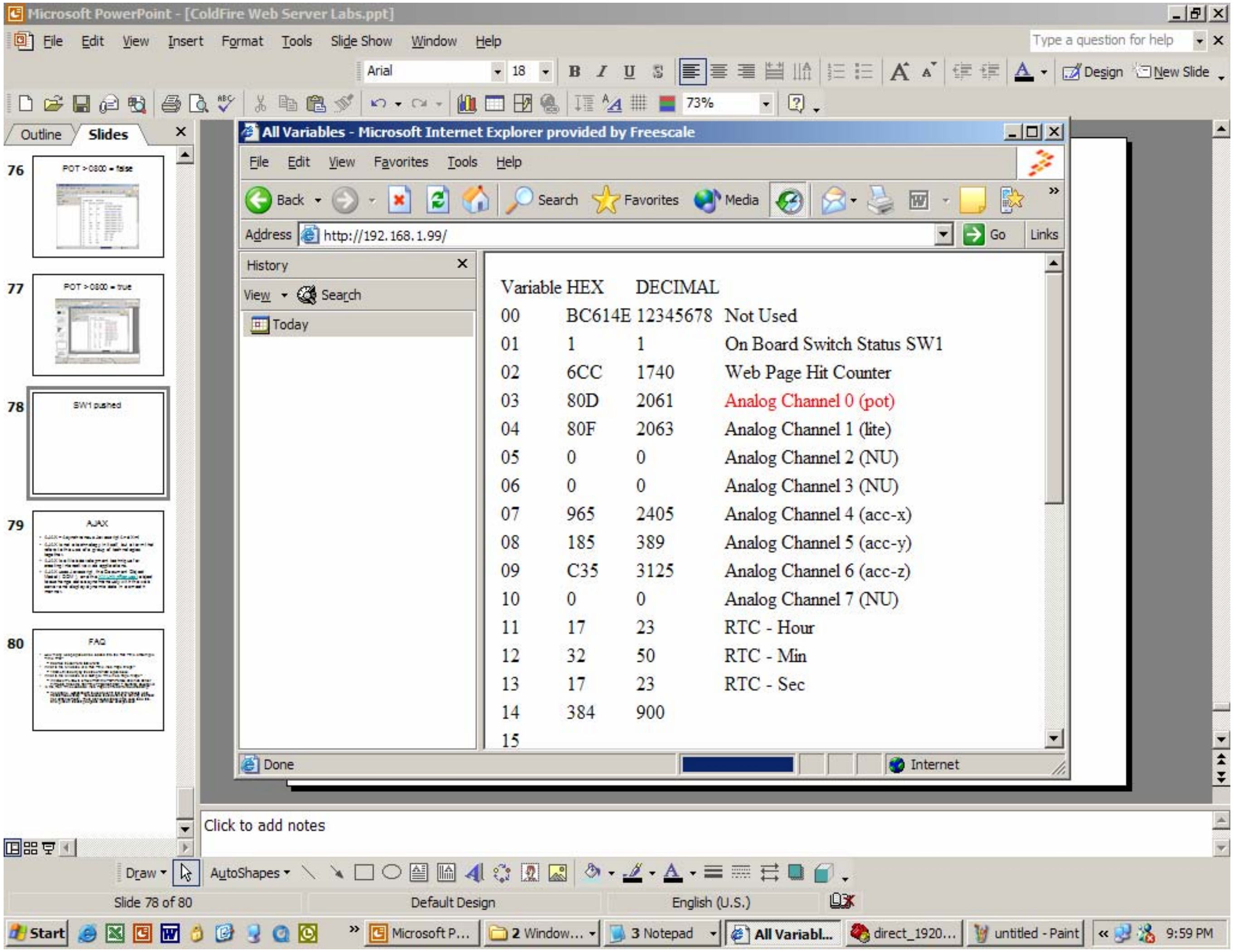
Variable	HEX	DECIMAL	
00	BC614E	12345678	Not Used
01	0	0	On Board Switch Status
02	679	1657	Web Page Hit Counter
03	0	0	Analog Channel 0 (pot)
04	45D	1117	Analog Channel 1 (lite)
05	0	0	Analog Channel 2 (NU)
06	0	0	Analog Channel 3 (NU)
07	95F	2399	Analog Channel 4 (acc-x)
08	18A	394	Analog Channel 5 (acc-y)
09	C33	3123	Analog Channel 6 (acc-z)
10	0	0	Analog Channel 7 (NU)
11	17	23	RTC - Hour
12	2F	47	RTC - Min
13	24	36	RTC - Sec
14	384	900	
15			

POT > 0800 = true


Microsoft PowerPoint - [ColdFire Web Server Labs.ppt]
All Variables - Microsoft Internet Explorer provided by Freescale
Address http://192.168.1.99/
History
Today
Done
Internet
Click to add notes
Slide 77 of 79
Default Design
English (U.S.)
9:58 PM

Variable	HEX	DECIMAL	
00	BC614E	12345678	Not Used
01	0	0	On Board Switch Status
02	6B8	1720	Web Page Hit Counter
03	810	2064	Analog Channel 0 (pot)
04	770	1904	Analog Channel 1 (lite)
05	0	0	Analog Channel 2 (NU)
06	0	0	Analog Channel 3 (NU)
07	973	2419	Analog Channel 4 (acc-x)
08	188	392	Analog Channel 5 (acc-y)
09	C3F	3135	Analog Channel 6 (acc-z)
10	0	0	Analog Channel 7 (NU)
11	17	23	RTC - Hour
12	31	49	RTC - Min
13	2B	43	RTC - Sec
14	384	900	
15			

SW1 Pushed


Microsoft PowerPoint - [ColdFire Web Server Labs.ppt]
All Variables - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.1.99/ Go Links
History
Today
Done
Internet
Click to add notes
Slide 78 of 80
Default Design
English (U.S.)
9:59 PM

Variable	HEX	DECIMAL	
00	BC614E	12345678	Not Used
01	1	1	On Board Switch Status SW1
02	6CC	1740	Web Page Hit Counter
03	80D	2061	Analog Channel 0 (pot)
04	80F	2063	Analog Channel 1 (lite)
05	0	0	Analog Channel 2 (NU)
06	0	0	Analog Channel 3 (NU)
07	965	2405	Analog Channel 4 (acc-x)
08	185	389	Analog Channel 5 (acc-y)
09	C35	3125	Analog Channel 6 (acc-z)
10	0	0	Analog Channel 7 (NU)
11	17	23	RTC - Hour
12	32	50	RTC - Min
13	17	23	RTC - Sec
14	384	900	
15			

Browser Update

Notice how the last lab updated itself in the browser

- The <meta http-equiv="refresh" content="1"> HTML tag causes the page to automatically reload.
- The “1” is the number of seconds to wait before reloading the page.

- This is the old method of automatically updating a web page.
- Notice its not very efficient, the whole page is reloaded even though only a few values change.
- Notice the page flickers.

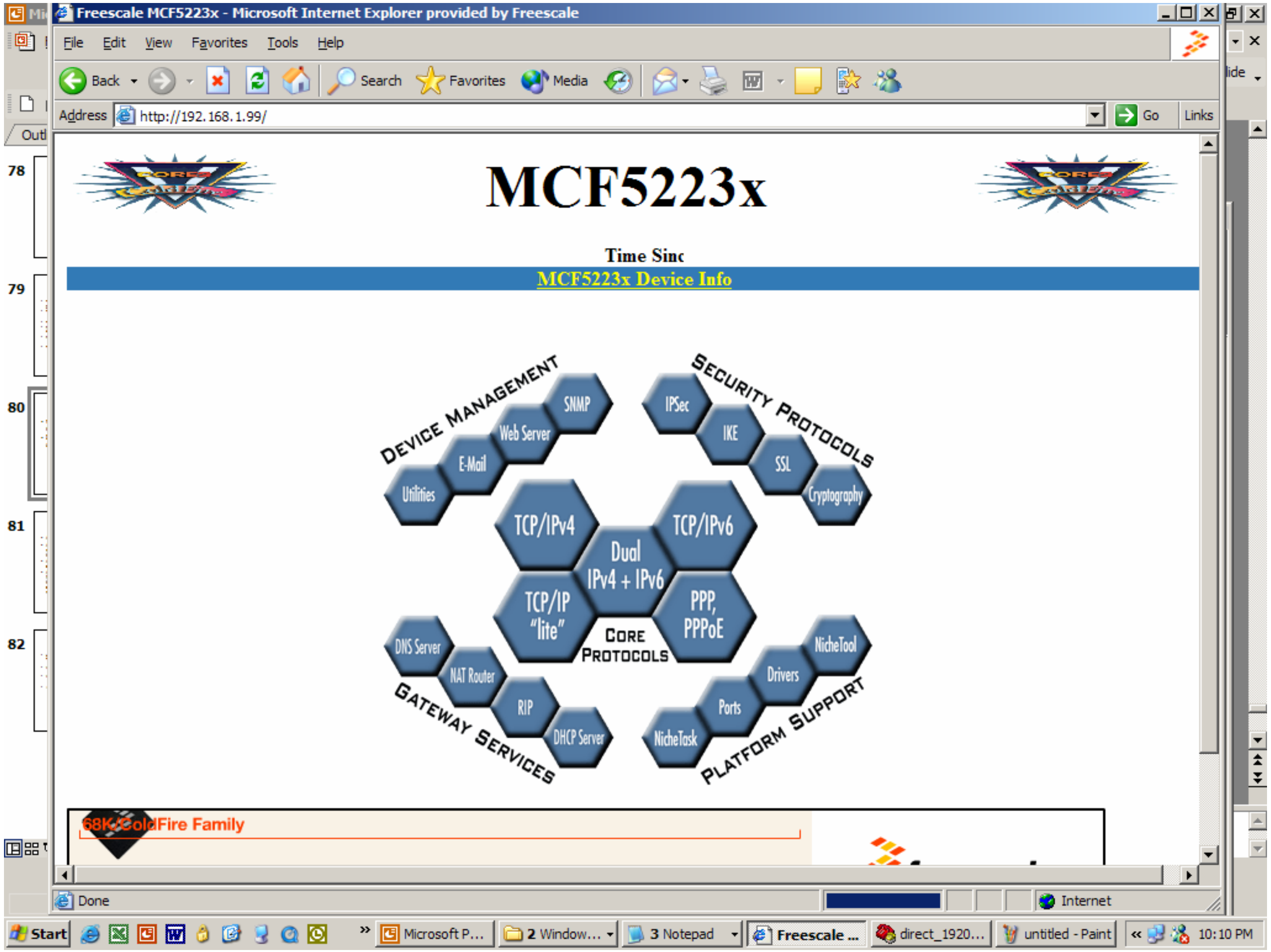
- These limitations are addressed in WEB2.0.

LAB 5: Another Reload Example with a Twist

- Go to the mcf5223x directory in the runtime directory.
- Double click the make.bat file to load the demo into the evaluation board.

Web Page Reloading

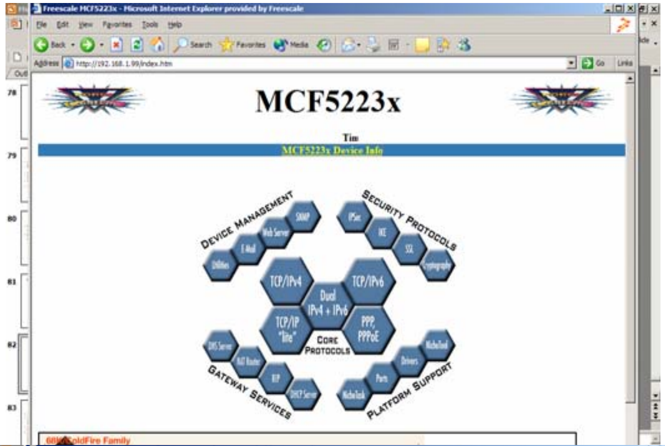
Notice the web page is reloading every second. Take a look at the marquee. It shows time since board reset


Freescale MCF5223x - Microsoft Internet Explorer provided by Freescale

Address http://192.168.1.99/

A Peek at Web2.0

Click on the mcf5223x_device_info URL



Freescale MCF5223x - Microsoft Internet Explorer provided by Freescale

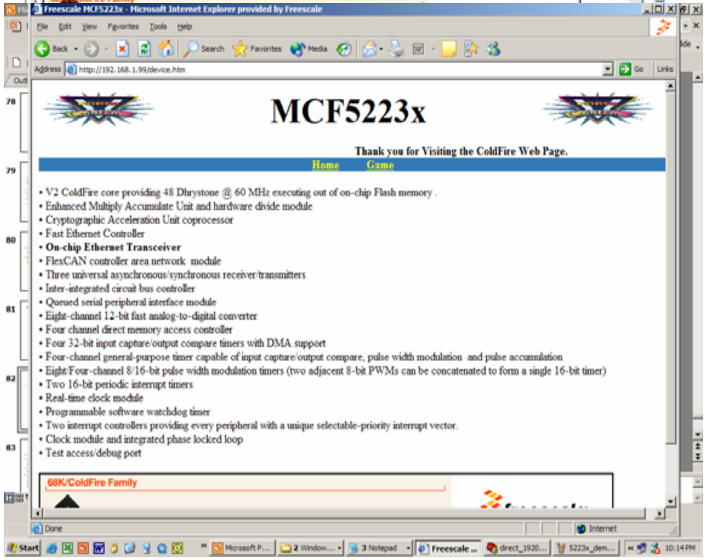
Address http://192.168.1.99/index.htm

MCF5223x

Tim

MCF5223x Device Info

Click on the game URL



Freescale MCF5223x - Microsoft Internet Explorer provided by Freescale

Address http://192.168.1.99/device.htm

MCF5223x

Thank you for Visiting the ColdFire Web Page.

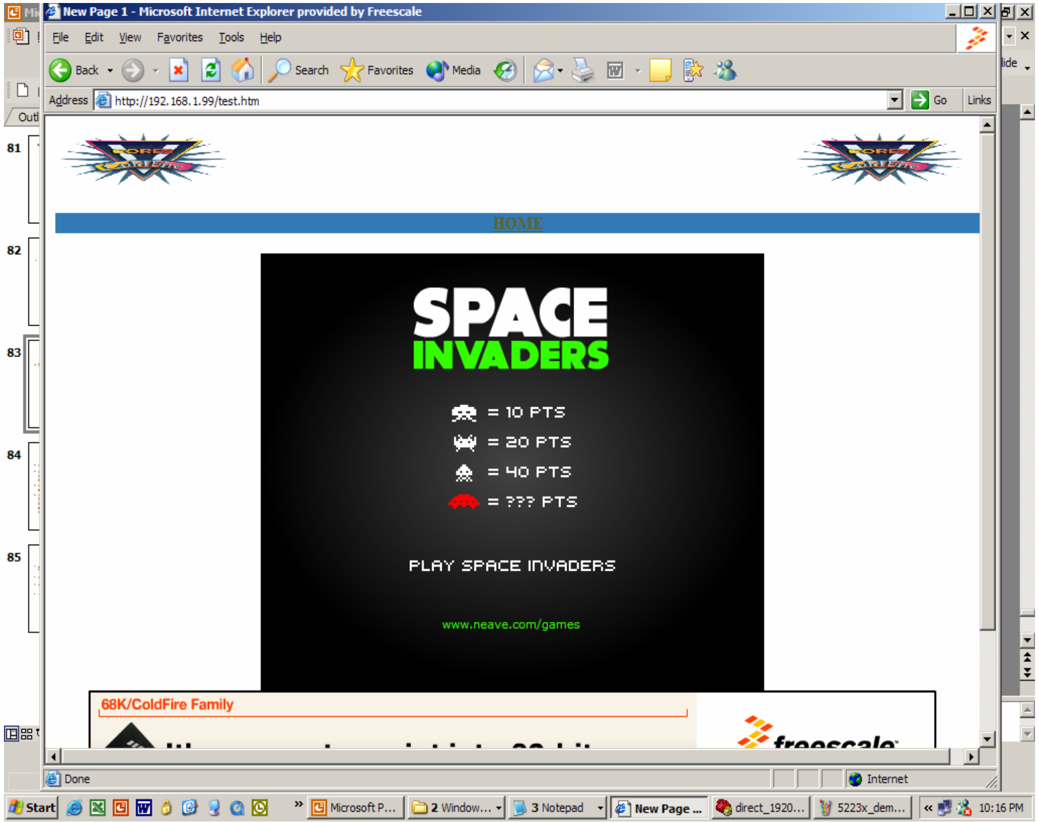
Home Game

- V2 ColdFire core providing 48 Dhrystone @ 60 MHz executing out of on-chip Flash memory .
- Enhanced Multiply Accumulate Unit and hardware divide module
- Cryptographic Acceleration Unit coprocessor
- Fast Ethernet Controller
- **On-chip Ethernet Transceiver**
- FlexCAN controller area network module
- Three universal asynchronous/synchronous receiver/transmitters
- Inter-integrated circuit bus controller
- Queued serial peripheral interface module
- Eight-channel 12-bit fast analog-to-digital converter
- Four channel direct memory access controller
- Four 32-bit input capture/output compare timers with DMA support
- Four-channel general-purpose timer capable of input capture/output compare, pulse width modulation and pulse accumulation
- Eight/Four-channel 8/16-bit pulse width modulation timers (two adjacent 8-bit PWMs can be concatenated to form a single 16-bit timer)
- Two 16-bit periodic interrupt timers
- Real-time clock module
- Programmable software watchdog timer
- Two interrupt controllers providing every peripheral with a unique selectable-priority interrupt vector.
- Clock module and integrated phase locked loop
- Test access/debug port

68K/ColdFire Family

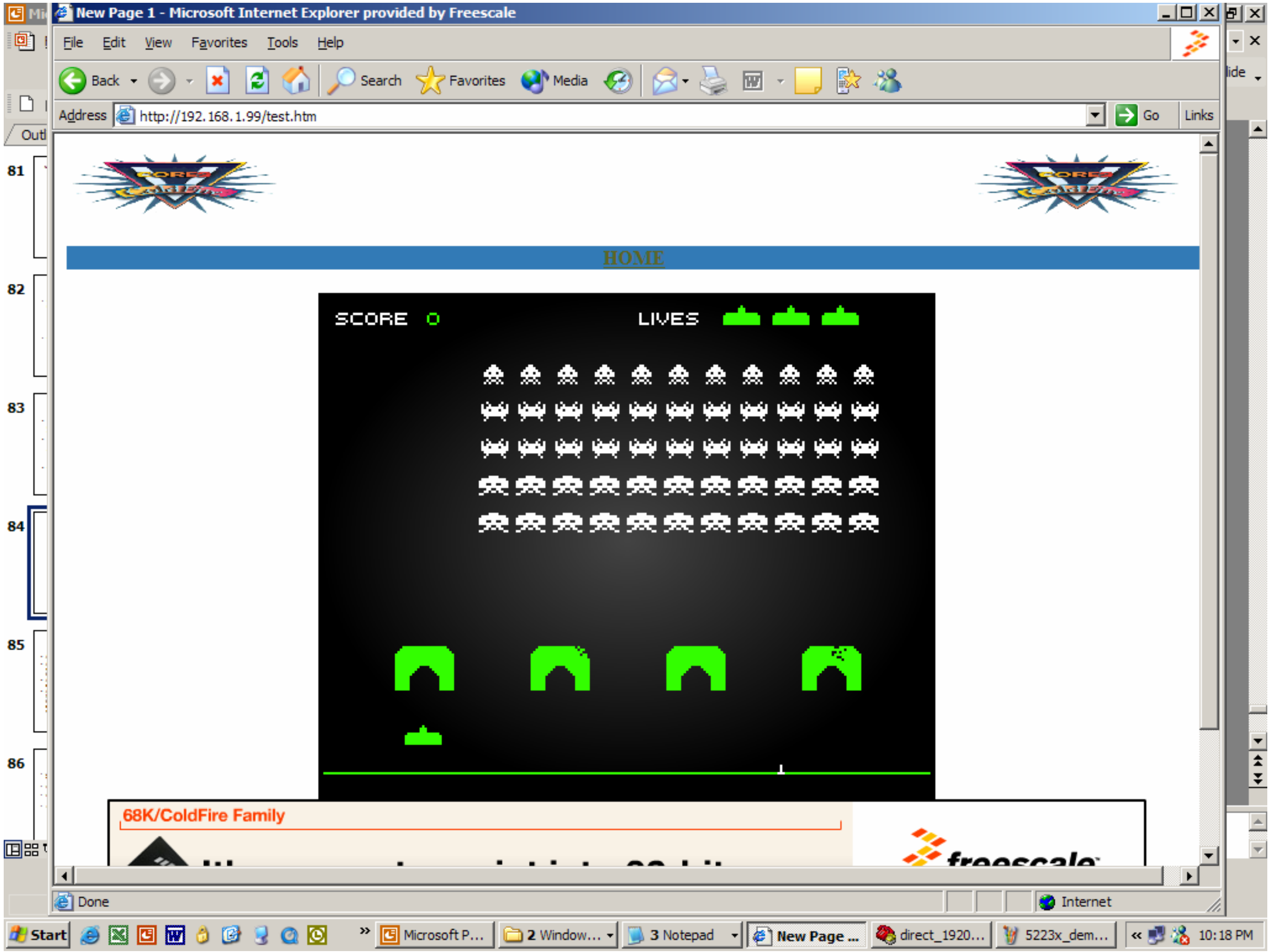
A Peek at Web2.0

- The space invaders game is a Shockwave file.
- Shockwave is a proprietary (reader is free, writer must be purchased) web plug-in.
- Click on ‘Play space invaders’



New Page 1 - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.1.99/test.htm
Go Links
HOME
SPACE INVADERS
= 10 PTS
= 20 PTS
= 40 PTS
= ??? PTS
PLAY SPACE INVADERS
www.neave.com/games
68K/ColdFire Family
freescale
Done
Internet
Start Microsoft P... 2 Window... 3 Notepad New Page ... direct_1920... 5223x_dem...
10:16 PM

The Power of Web2.0


New Page 1 - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.1.99/test.htm
Go Links
HOME
SCORE 0
LIVES
68K/ColdFire Family
Done
Internet
Start
Microsoft P...
2 Window...
3 Notepad
New Page ...
direct_1920...
5223x_dem...
10:18 PM

WEB 2.0

- Web 2.0 generally refers to a second generation of services available on the World Wide Web that gives users an experience closer to a desktop application than the traditional static web pages.
- The traditional world wide web was designed to present static information.
- Web 2.0 is designed to be interactive.

AJAX - A Key Component of Web 2.0

- AJAX – Asynchronous Javascript And XML
- AJAX is not a technology in itself, but a term that refers to the use of a group of technologies together.
- AJAX is a Web development technique for creating interactive web applications.
- AJAX uses Javascript, the Document Object Model (DOM), and the XMLHttpRequest object to exchange data asynchronously with the web server and display dynamic data in a smooth manner.

Javascript

- Javascript is a prototype-based scripting language with a syntax loosely based on ‘C’.
- Javascript is embedded as ascii source in web pages.
- The web browser interprets the Javascript within the <HTML> tags.
- Since the browser actually runs the Javascript, all the web server has to do is serve it up.
- Including Javascript in your we pages is easy.

Simple “Hello World” in Javascript

```
<html>
<head>
<title>Simple Javascript</title>
</head>
<script language="JavaScript">
document.write(“Hello World”);
</script>
</html>
```

Document Object Model (DOM)

- Javascript would be relatively useless if it could not alter the web page.
- Of course, Javascript can alter the web page using the DOM.
- The DOM makes everything on a web page a object accessible by Javascript.
- Javascript accesses the object using the object ID.

Javascript + DOM

Remember the marquee in the web page from the last lab

- <marquee width="800" scrollamount=8>Time Since Last Reset: ~11D;~12D;~13D;</marquee>

We modify it slightly by adding the id element

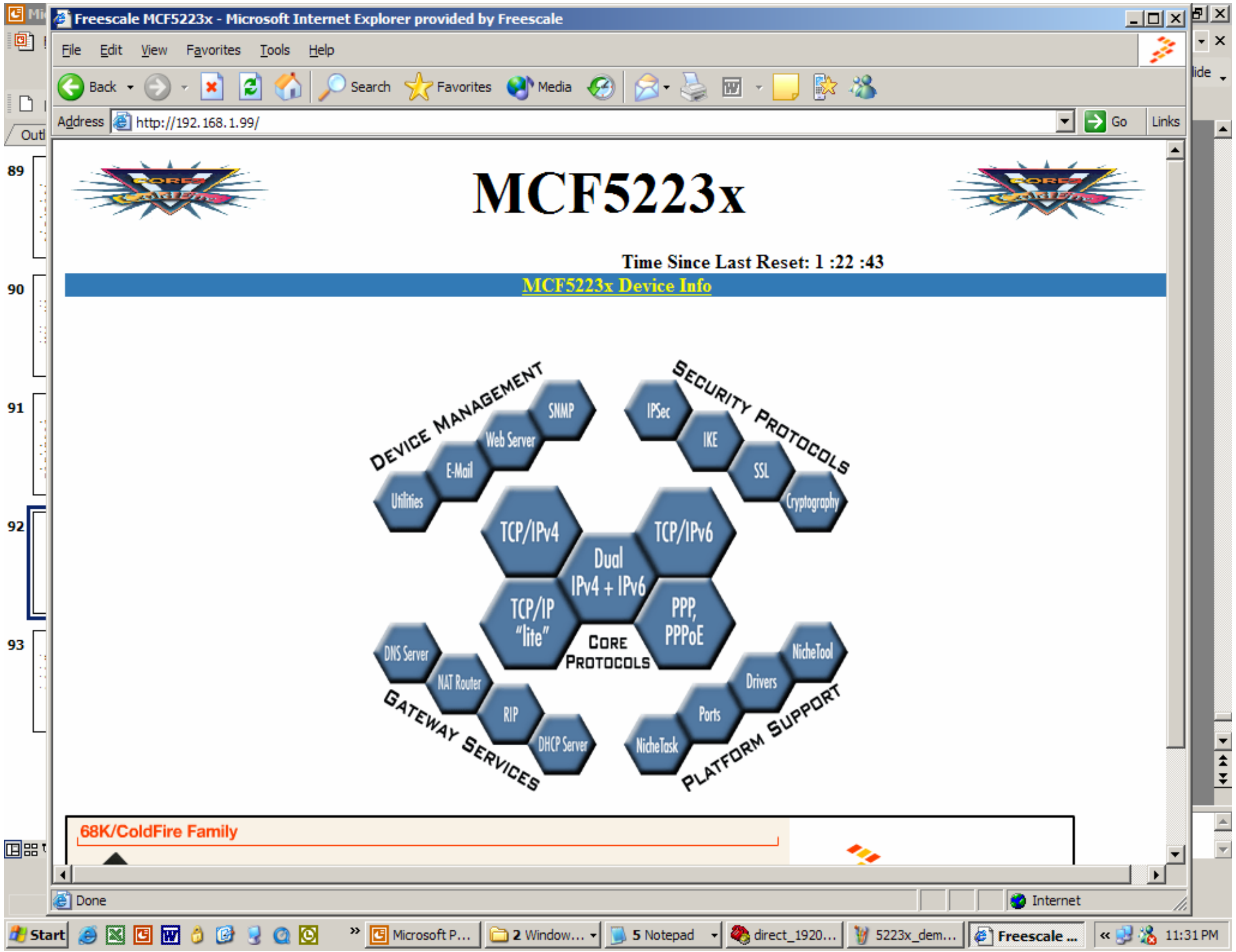
- <marquee id="scroller" width="800" scrollamount=8>Time Since Last Reset: ~11D;~12D;~13D;</marquee>
- Now, we can alter the marquee from Javascript.

LAB 6: Improving an Old Friend

- Go to the mcf5223x_ajax_demo in the runtime_loaded_web_page_example directory.
- Double click on the make.bat file.
- This loads the web page onto the eval board.
- Open your browser and type 192.168.1.99 in the address bar.

Notice - No Flicker

Check out the marquee time



Freescale MCF5223x - Microsoft Internet Explorer provided by Freescale

Address http://192.168.1.99/

MCF5223x

Time Since Last Reset: 1 :22 :43

MCF5223x Device Info

Javascript Runs in the Background

- The time in the web page automatically updates.
- The time is actually being read from the ***ColdFire***® evaluation board Real Time Clock.
- Javascript uses the XMLHttpRequest function to request data from the web server, without effecting the viewable page.

Javascript and Internet Explorer

- Internet Explorer has an issue terminating Javascript.
- Between the Javascript labs, you should close and re-open Internet Explorer.

LAB 7: ShockWave Example

- Goto the LAB7_?????? Directory.
- Double Click the make.bat to load the LAB into the ColdFire.
- At the serial prompt, type dir
  - Notice the Flash File System supports subdirectories.

LAB 8: ShockWave Example

- Goto the LAB8_?????? Directory.
- Double Click the make.bat to load the LAB into the ColdFire.

LAB 9: Accelerometer Example

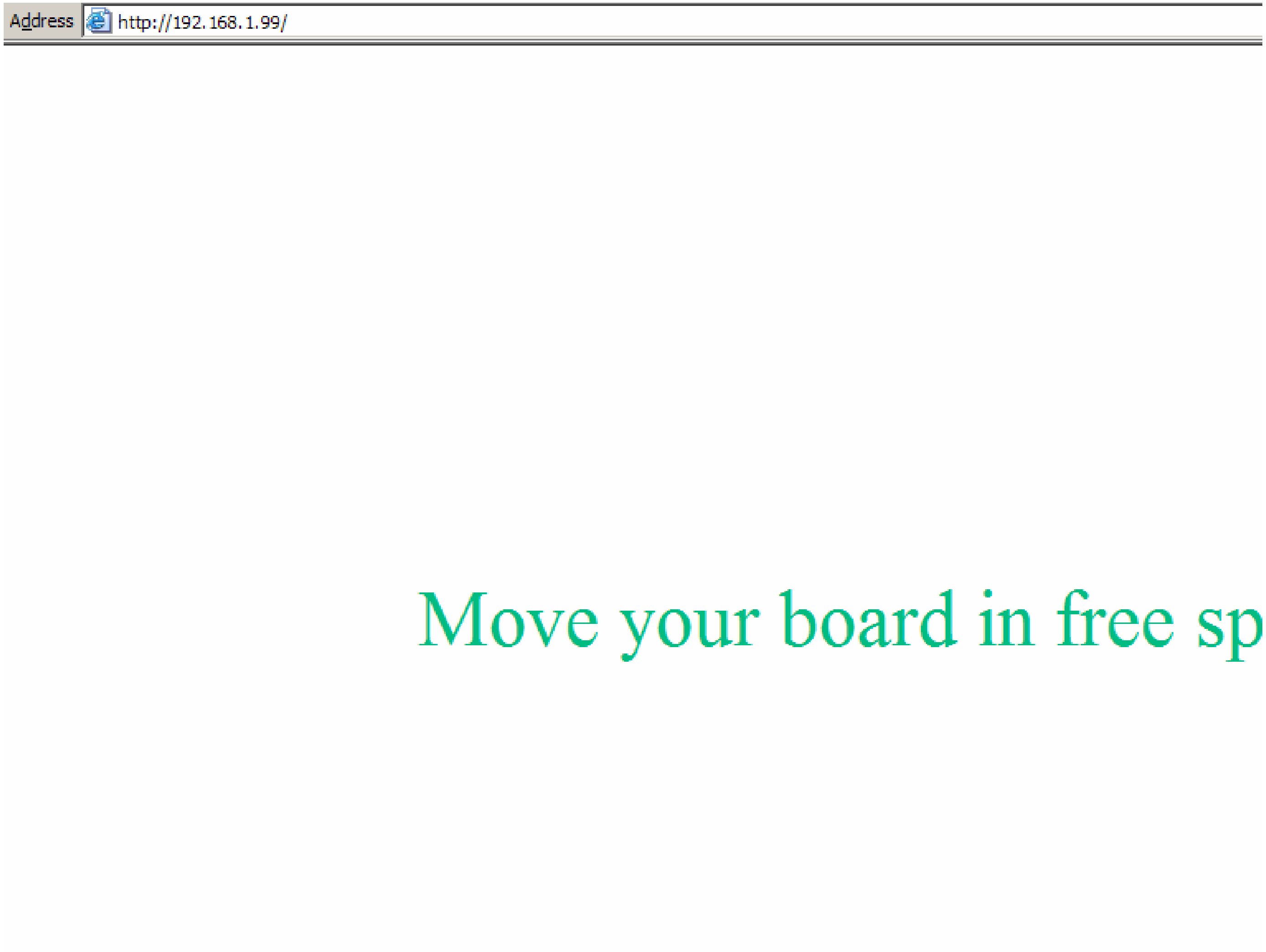
- Goto the LAB9_?????? Directory.
- Double Click the make.bat to load the LAB into the ColdFire.

The 52233DEMO board has a 3-axis accelerameter. This device outputs 3 analog voltages representing the x, y, and z planes.

The ColdFire has 2 separate 4 channel 12 bit A/D converters.

3 channels are used here to read the X, y, and z planes, then the A/D values are stored in VAR array locations 7, 8, and 9.

LAB 9: Accelerometer Example



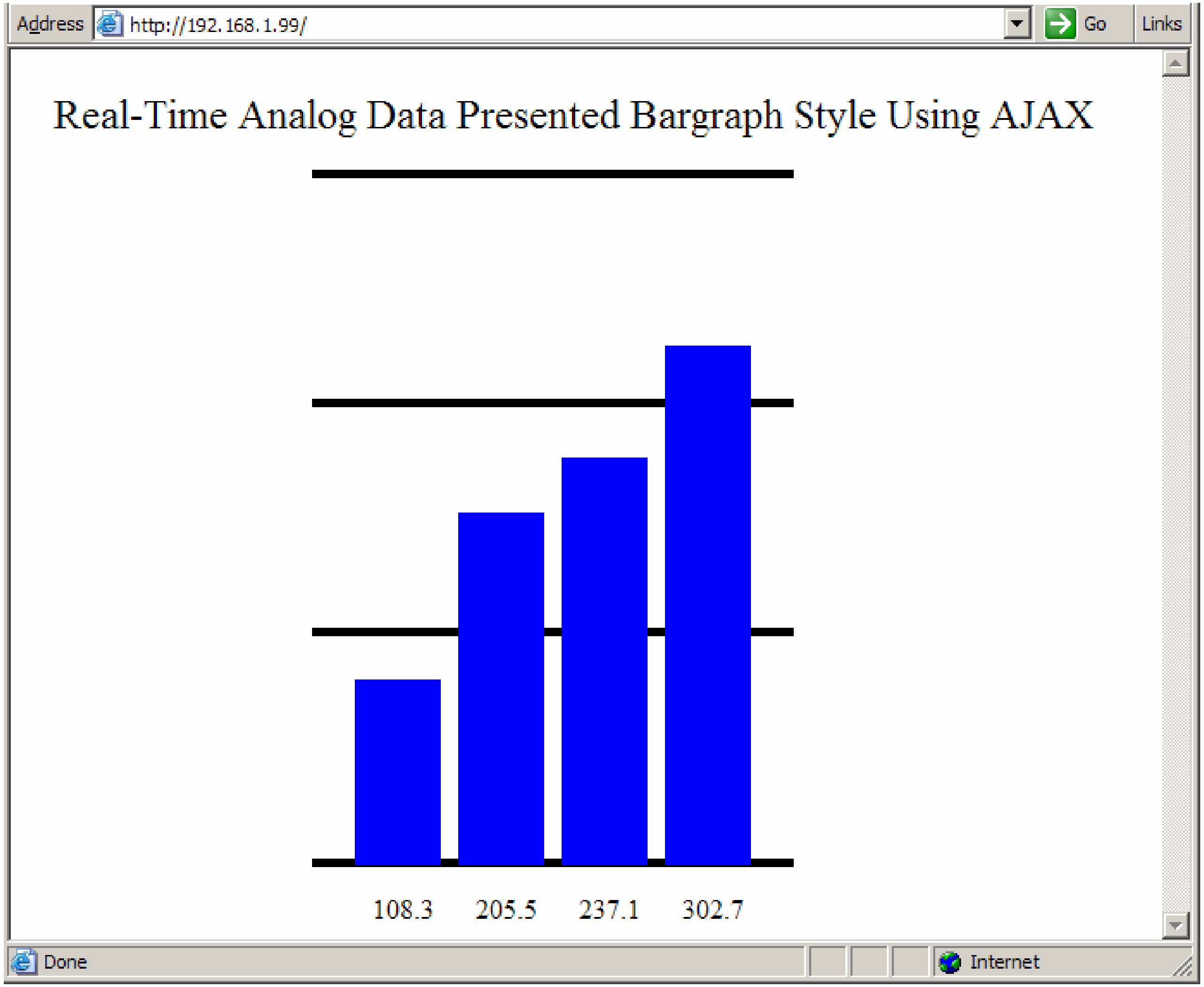
http://192.168.1.99/

Move your board in free sp

LAB 10: Monitoring Analog Data

- Goto the LAB10_?????? Directory.
- Double Click the make.bat to load the LAB into the ColdFire.

LAB 10: Monitoring Analog Data



Real-Time Analog Data Presented Bargraph Style Using AJAX

HTML and Javascript

Notice the image has been given an id of bargraph

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Freescale MCF5223x</title>
</head>

<body>

<IMG SRC="avtlogo.gif" id="bargraph" BORDER=0 WIDTH=549 HEIGHT=470 >

</body>
```

HTML and Javascript

The Javascript assigns the height of the bargraph object to the pot_value/10

```
<script language="JavaScript">
//////////////////////////////////////////////////////////////////////////////////////////
// Javascript for 5223EVB demo Written by Eric Gregori
//
// The script communicates with the board using a AJAX technique.
/////////////////////////////////////////////////////////////////////////////////////////


/////////////////////////////////////////////////////////////////////////////////////////
// Variables global to script
/////////////////////////////////////////////////////////////////////////////////////////
var pot_value


/////////////////////////////////////////////////////////////////////////////////////////
// Parse input file
/////////////////////////////////////////////////////////////////////////////////////////
function parse_vars( data )
{
                var parsed = data.split( "\n" );

                pot_value                = parsed[0]

                bargraph.height = pot_value/10
}
```

HTML and Javascript

The Javascript request the data from the server using http_request.open('GET', url, true);

```
////////////////////////////////////////////////////////////////////////////////
// Request input file
////////////////////////////////////////////////////////////////////////////////
function makeRequest(url)
{
        var http_request = false;

  if (window.XMLHttpRequest)
        { // Mozilla, Safari,...
        http_request = new XMLHttpRequest();
        if (http_request.overrideMimeType)
                {
        http_request.overrideMimeType('text/xml');
        }
  }
        else if (window.ActiveXObject)
        { // IE
                try
                {
        http_request = new ActiveXObject("Msxml2.XMLHTTP");
        }
                catch (e)
                {
        try
                        {
                http_request = new ActiveXObject("Microsoft.XMLHTTP");
        }
                        catch (e) {}

        }
  }

  if (!http_request)
        {
        alert('Giving up :( Cannot create an XMLHTTP instance');
        return false;
  }

  http_request.onreadystatechange = function() { alertContents(http_request); };
  http_request.open('GET', url, true);
  http_request.send(null);
}
```

HTML and Javascript

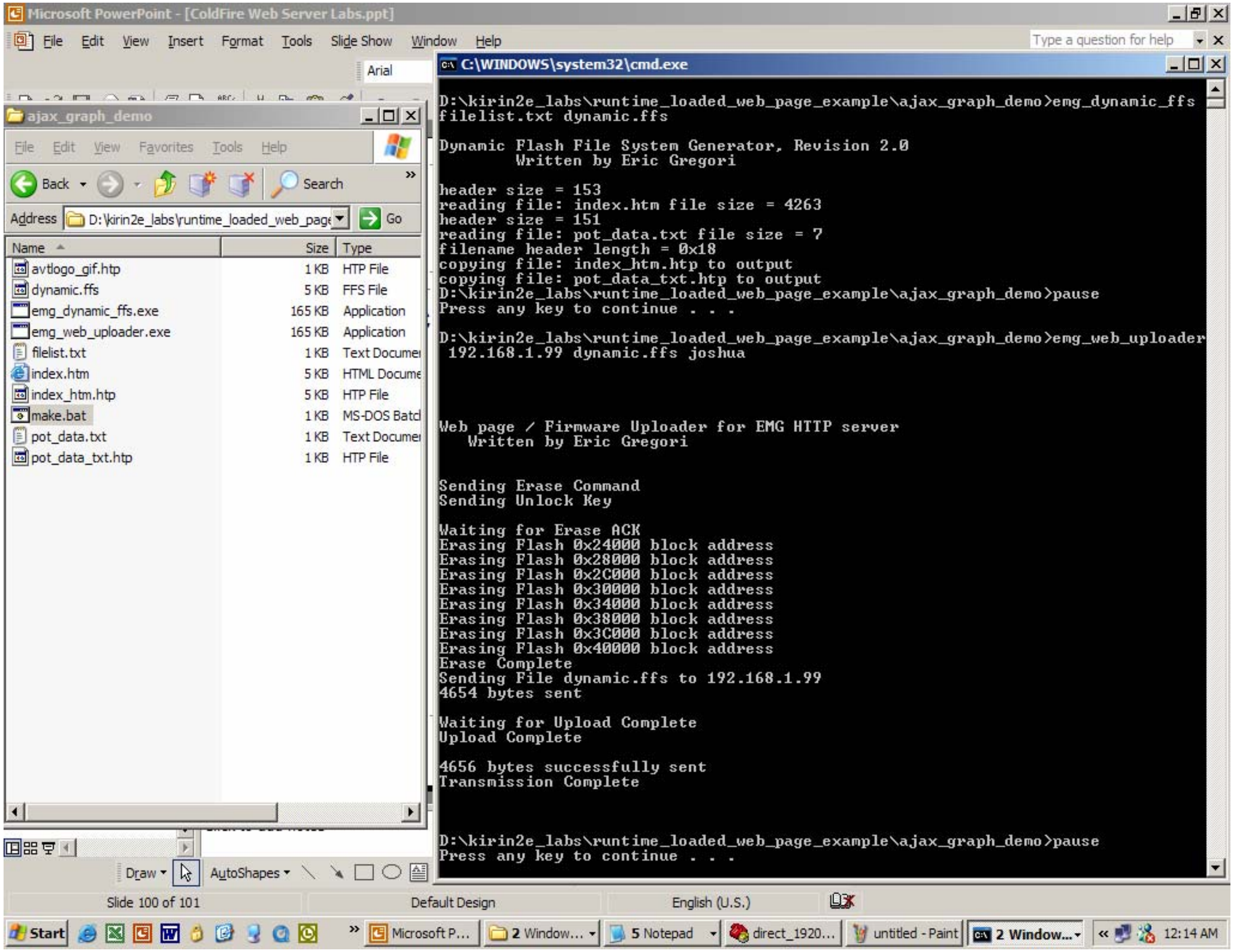
The javascript request the data from the server by requesting the file pot_data.txt
This request is done every 200ms (setTimeout).

```
////////////////////////////////////////////////////////////////////////////////////////////
// Handle file request response
////////////////////////////////////////////////////////////////////////////////////////////
function alertContents(http_request)
{
        if (http_request.readyState == 4)
        {
                if (http_request.status == 200)
                {
        parse_vars(http_request.responseText);
        }
                else
                {
//      alert('There was a problem with the request.');
//                      alert( http_request.status );
        }
  }
}

////////////////////////////////////////////////////////////////////////////////////////////
// Infinite loop with delay
////////////////////////////////////////////////////////////////////////////////////////////
function loop()
{
        makeRequest("pot_data.txt");
        setTimeout("loop()",200);
}

////////////////////////////////////////////////////////////////////////////////////////////
// Run
////////////////////////////////////////////////////////////////////////////////////////////
window.onload=loop;


</script>
</html>
```

LAB 11: Back to Real Work

- AJAX can be used for more than fun and games.
- In an embedded environment sometimes it would be nice to present real-time changing data in a graphic manner.
- Go to the ajax_graph_demo directory.
- Close the web browser (internet explorer).
- Double click the make.bat file.
- Open Internet Explorer, and type 192.168.1.99 in the address bar.

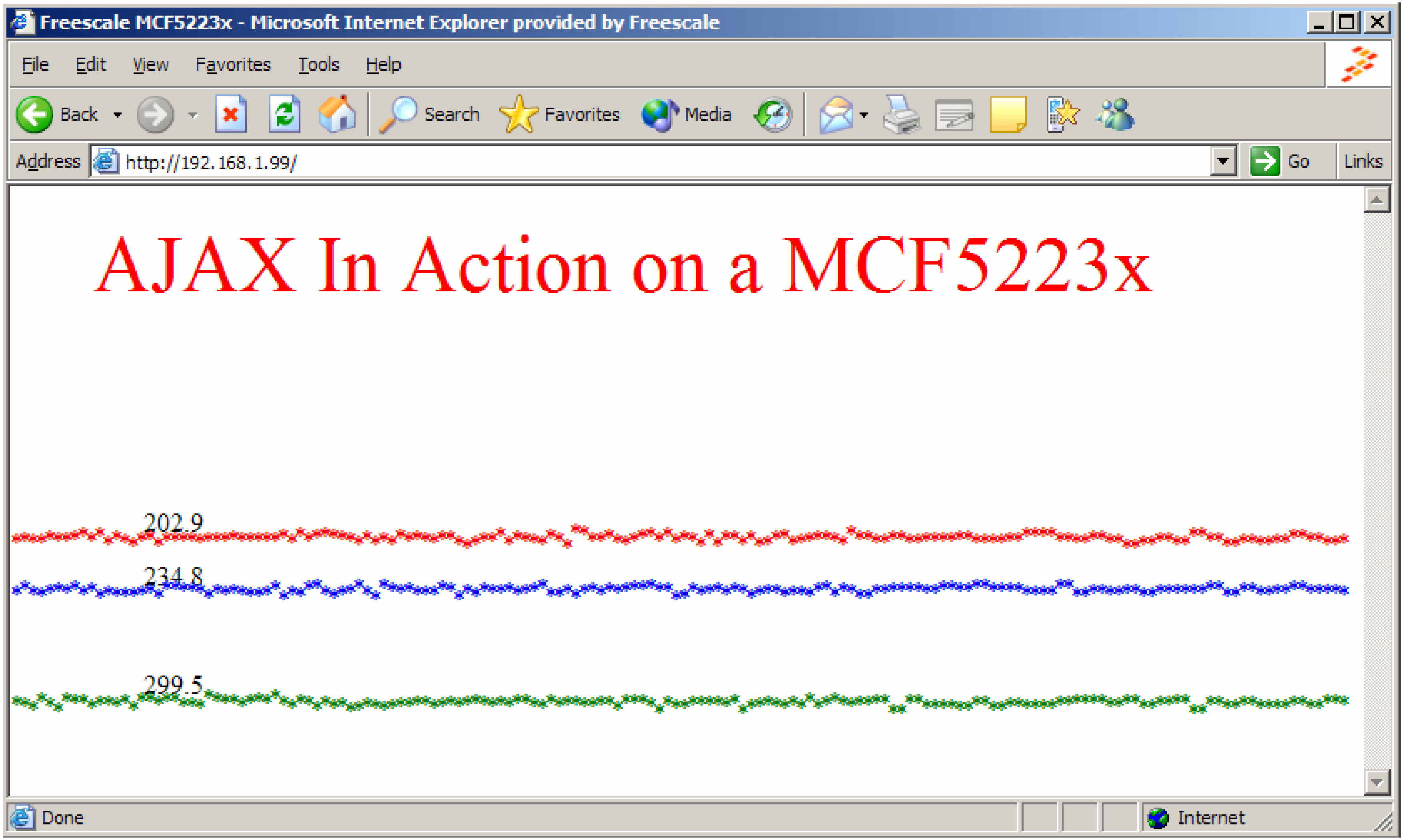
Build and Load ajax_graph_demo


Microsoft PowerPoint - [ColdFire Web Server Labs.ppt]
File Edit View Insert Format Tools Slide Show Window Help
Type a question for help
Arial
ajax_graph_demo
File Edit View Favorites Tools Help
Back
Search
Address D:\kirin2e_labs\runtime_loaded_web_page
Go
C:\WINDOWS\system32\cmd.exe
Draw
AutoShapes
Slide 100 of 101
Default Design
English (U.S.)
Start
Microsoft P...
2 Window...
5 Notepad
direct_1920...
untitled - Paint
2 Window...
12:14 AM

Name	Size	Type
avtlogo_gif.htp	1 KB	HTP File
dynamic.ffs	5 KB	FFS File
emg_dynamic_ffs.exe	165 KB	Application
emg_web_uploader.exe	165 KB	Application
filelist.txt	1 KB	Text Docume
index.htm	5 KB	HTML Docume
index_htm.htp	5 KB	HTP File
make.bat	1 KB	MS-DOS Batc
pot_data.txt	1 KB	Text Docume
pot_data_txt.htp	1 KB	HTP File

```
D:\kirin2e_labs\runtime_loaded_web_page_example\ajax_graph_demo>emg_dynamic_ffs
filelist.txt dynamic.ffs

Dynamic Flash File System Generator, Revision 2.0
        Written by Eric Gregori

header size = 153
reading file: index.htm file size = 4263
header size = 151
reading file: pot_data.txt file size = 7
filename header length = 0x18
copying file: index_htm.htp to output
copying file: pot_data_txt.htp to output
D:\kirin2e_labs\runtime_loaded_web_page_example\ajax_graph_demo>pause
Press any key to continue . . .

D:\kirin2e_labs\runtime_loaded_web_page_example\ajax_graph_demo>emg_web_uploader
 192.168.1.99 dynamic.ffs joshua



Web page / Firmware Uploader for EMG HTTP server
   Written by Eric Gregori


Sending Erase Command
Sending Unlock Key

Waiting for Erase ACK
Erasing Flash 0x24000 block address
Erasing Flash 0x28000 block address
Erasing Flash 0x2C000 block address
Erasing Flash 0x30000 block address
Erasing Flash 0x34000 block address
Erasing Flash 0x38000 block address
Erasing Flash 0x3C000 block address
Erasing Flash 0x40000 block address
Erase Complete
Sending File dynamic.ffs to 192.168.1.99
4654 bytes sent

Waiting for Upload Complete
Upload Complete

4656 bytes successfully sent
Transmission Complete



D:\kirin2e_labs\runtime_loaded_web_page_example\ajax_graph_demo>pause
Press any key to continue . . .
```

Turn the POT

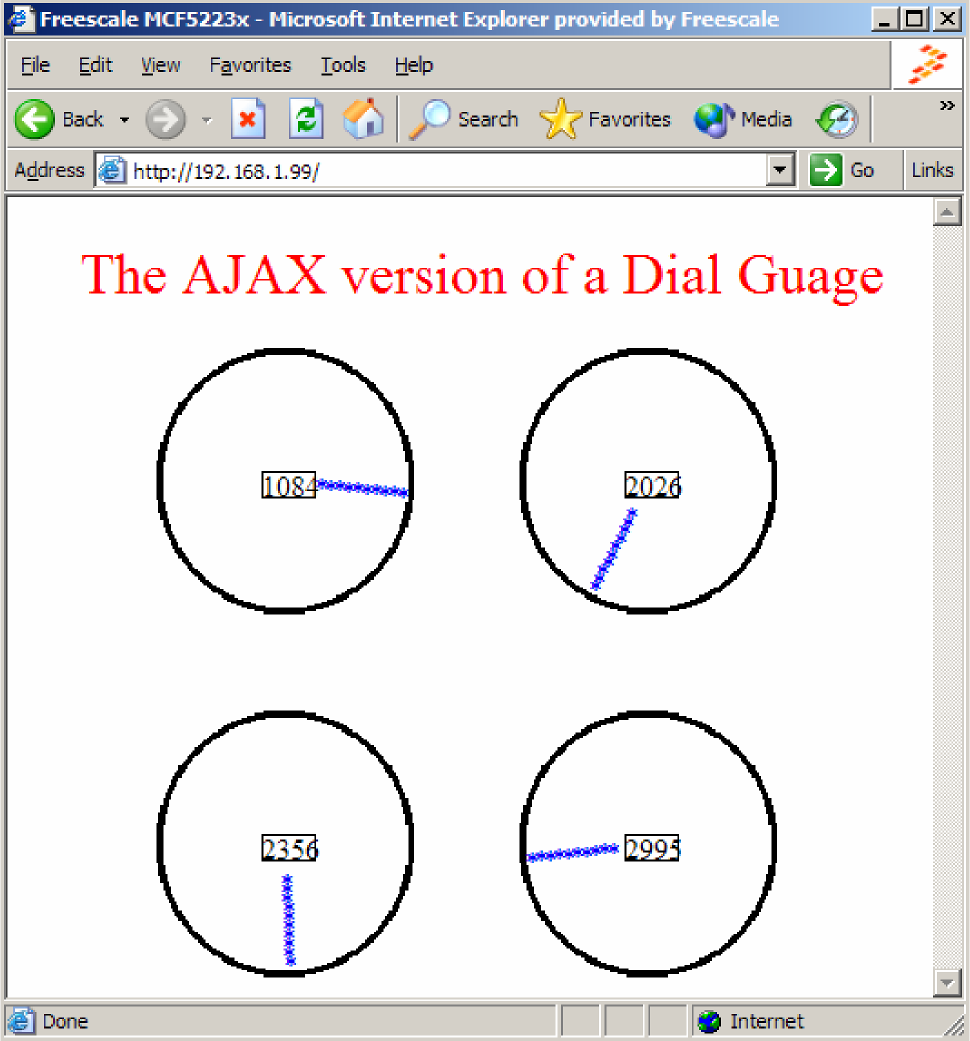

Freescale MCF5223x - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://192.168.1.99/
Go
Links
Done
Internet

AJAX In Action on a MCF5223x

LAB 12: Monitoring Analog Data with a dial guage

- Goto the LAB12_?????? Directory.
- Double Click the make.bat to load the LAB into the ColdFire.

Turn the POT, and move the board around



Freescale MCF5223x - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.1.99/ Go Links

The AJAX version of a Dial Guage

1084
2026
2356
2995
Done
Internet

LAB 13: Accessing files in the FFS

- Goto the LAB13_?????? Directory.
- Double Click the make.bat to load the LAB into the ColdFire.
- Go through the presentation

The Powerpoint presentation has been converted to HTML and Javascript. The presentation is being served up by the ColdFire.

FFS User API

- The FFS has a User API for user applications to access the flash file system.
- The FFS can be used to store any type of data, binary or ascii.
- The user can store accel tables, nv parameters, configuration info, …
- The information can be accessed by the firmware with a simple open call.
- The user can update the information by doing a runtime file load.

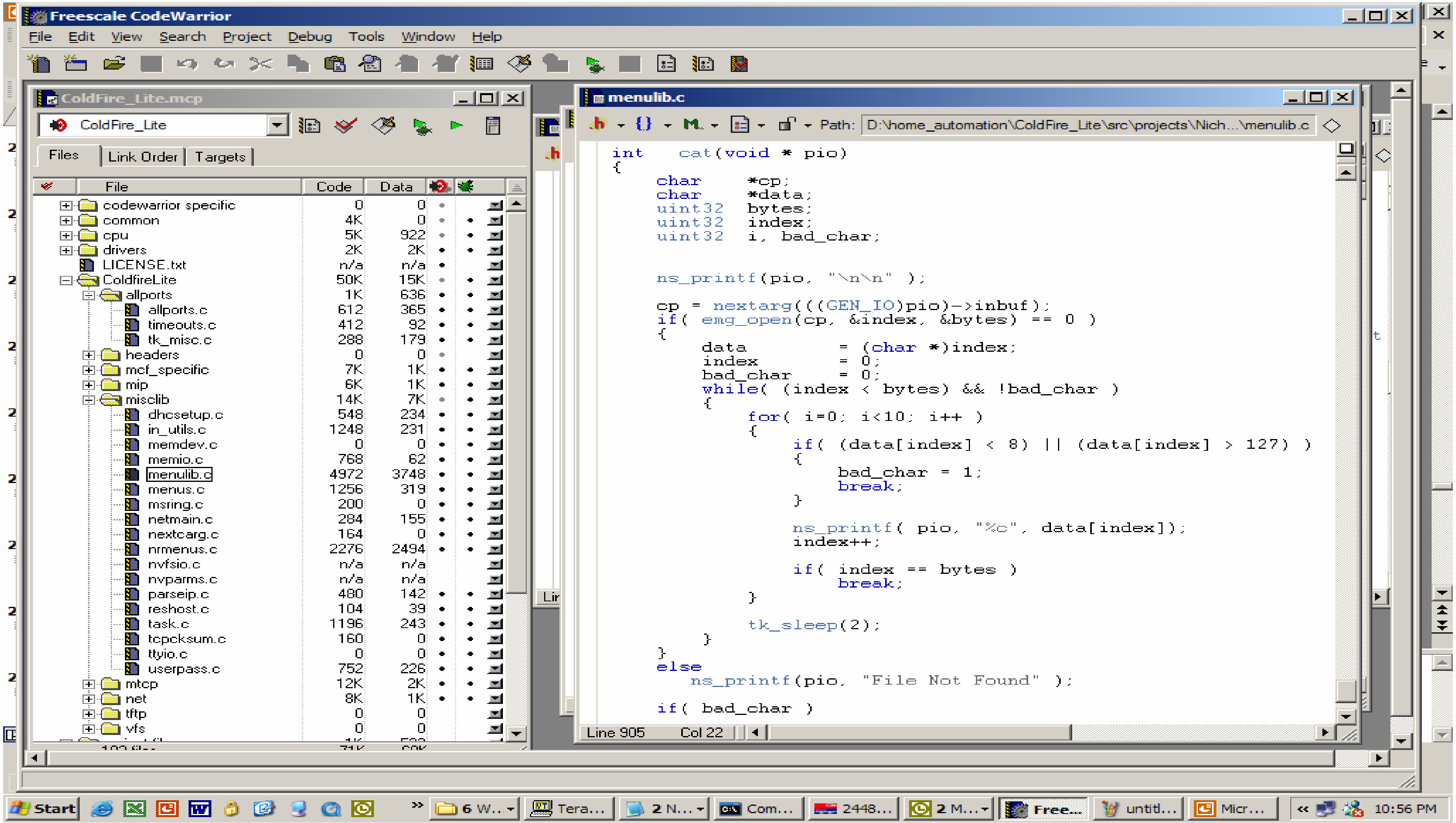
Emg_open

- //**
- // int emg_open(char *filename, uint32 *data_pointer, uint32 *file_size)
- //
- // User API to dynamic flash file system
- //
- // Finds the file descriptor in the FAT.
- // Sets data_pointer to start of data.
- // Sets file_size to size of file in bytes.
- // returns a < 0 if error, 0 = success
- //
- // for an example of using emg_open(), see cat command in menulib.c
- //
- //
- // Author: Eric Gregori (847) 651 - 1971
- //   eric.gregori@freescale.com
- //**

The CAT command

- The CAT command is an example of how to use the emg_open() function.
- The CAT command will dump the contents of a file to the console.

The CAT command code



File	Code	Data		
codewarrior specific	0	0	●	
common	4K	0	●	●
cpu	5K	922	●	●
drivers	2K	2K	●	●
LICENSE.txt	n/a	n/a	●	
ColdfireLite	50K	15K	●	●
allports	1K	636	●	●
allports.c	612	365	●	●
timeouts.c	412	92	●	●
tk_misc.c	288	179	●	●
headers	0	0	●	
mcf_specific	7K	1K	●	●
mip	6K	1K	●	●
misclib	14K	7K	●	●
dhcsetup.c	548	234	●	●
in_utils.c	1248	231	●	●
memdev.c	0	0	●	●
memio.c	768	62	●	●
menulib.c	4972	3748	●	●
menus.c	1256	319	●	●
msring.c	200	0	●	●
netmain.c	284	155	●	●
nextcarg.c	164	0	●	●
nrmenus.c	2276	2494	●	●
nvfsio.c	n/a	n/a		
nvparms.c	n/a	n/a		
parseip.c	480	142	●	●
reshost.c	104	39	●	●
task.c	1196	243	●	●
tcpcksum.c	160	0	●	●
ttyio.c	0	0	●	●
userpass.c	752	226	●	●
mtcp	12K	2K	●	●
net	8K	1K	●	●
tftp	0	0		
vfs	0	0		

```
int    cat(void * pio)
{
        char     *cp;
        char     *data;
        uint32   bytes;
        uint32   index;
        uint32   i, bad_char;


        ns_printf(pio, "\n\n" );

        cp = nextarg(((GEN_IO)pio)->inbuf);
        if( emg_open(cp, &index, &bytes) == 0 )
        {
                data          = (char *)index;
                index         = 0;
                bad_char      = 0;
                while( (index < bytes) && !bad_char )
                {
                        for( i=0; i<10; i++ )
                        {
                                if( (data[index] < 8) || (data[index] > 127) )
                                {
                                        bad_char = 1;
                                        break;
                                }

                                ns_printf( pio, "%c", data[index]);
                                index++;

                                if( index == bytes )
                                        break;
                        }

                        tk_sleep(2);
                }
        }
        else
           ns_printf(pio, "File Not Found" );

        if( bad_char )
```

LAB 14: Try to load a image > 128K

- Goto the LAB14_?????? Directory.
- Double Click the make.bat to load the LAB into the ColdFire.

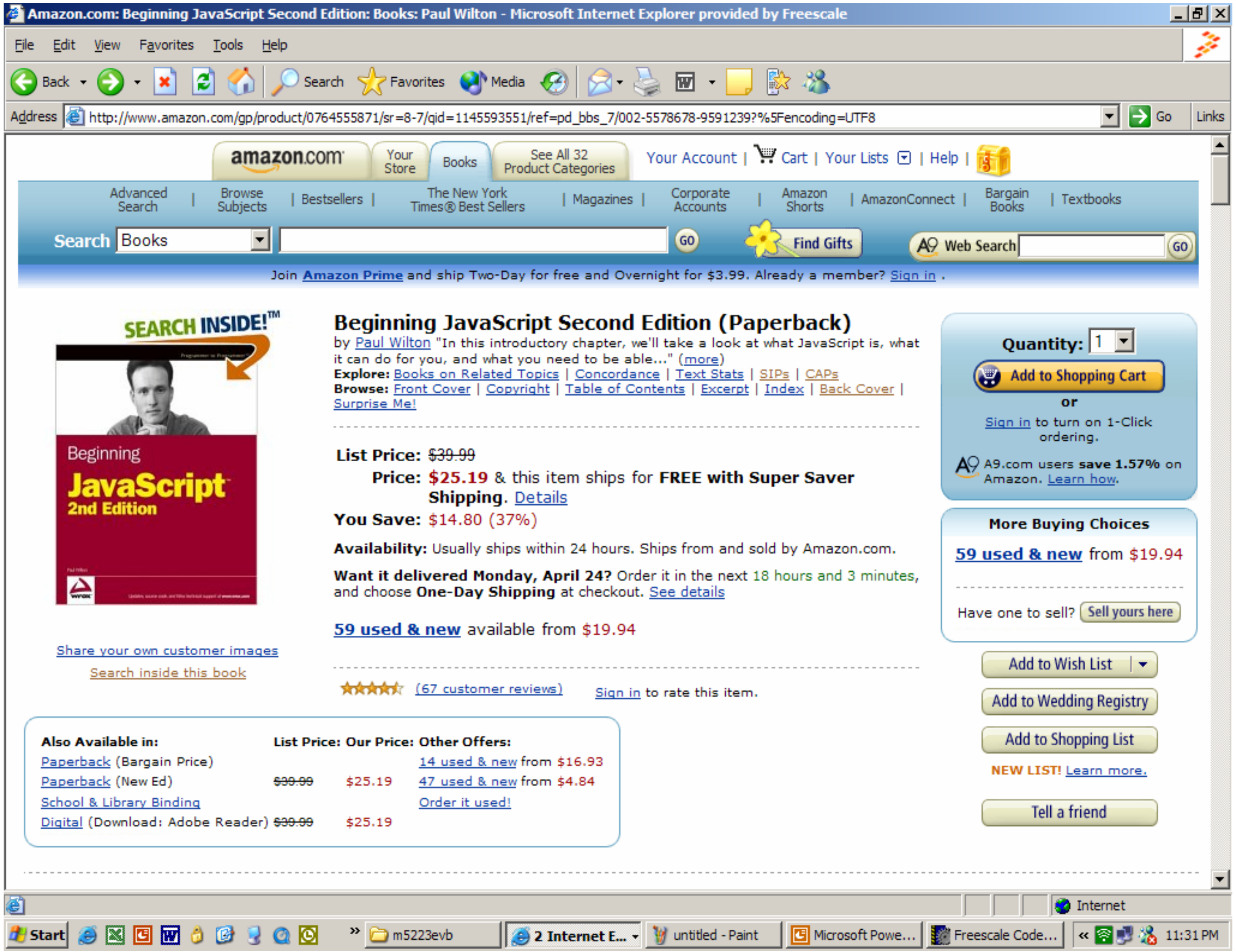
The load will fail, because the image is too big.

Verify that the original dynamic FFS contents have not been corrupted.

FAQs

- How many web pages can be loaded into the Run Time or Compile Time FFS?
  - 255 files in each for a total of 510
- What is the MAX size of a Run Time Web Page image?
  - 128K, Limited only by the size of a flash logical block.
- What is the MAX size of a Compile Time Web Page Image?
  - Whatever FLASH is left over from the TCP/IP stack and Web Server Firmware minus the Run Time FFS area(128K) = Currently about 64K.
- Is the Run Time Loadable Web Page verified after downloading?
  - Yes and no. Handshaking is used to verify that all the pakets were transferred correctly. No, because there currently is no verify that flash got written correctly. There are hooks already in the code to do this, and I plan on releasing a update with these changes soon.
- How quickly can AJAX poll the server for information?
  - That depends on the connection, and the web browser. With a small closed network, and Internet Explorer 6.0, the update rate can be as high as 100ms.

Reference Material


Amazon.com: Beginning JavaScript Second Edition: Books: Paul Wilton - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://www.amazon.com/gp/product/0764555871/sr=8-7/qid=1145593551/ref=pd_bbs_7/002-5578678-9591239?%5Fencoding=UTF8
Go
Links
amazon.com
Your Store
Books
See All 32 Product Categories
Your Account | Cart | Your Lists | Help |
Advanced Search | Browse Subjects | Bestsellers | The New York Times® Best Sellers | Magazines | Corporate Accounts | Amazon Shorts | AmazonConnect | Bargain Books | Textbooks
Search Books
GO
Find Gifts
A9 Web Search
GO
Internet
Start
m5223evb
2 Internet E...
untitled - Paint
Microsoft Powe...
Freescale Code...
11:31 PM

Join **Amazon Prime** and ship Two-Day for free and Overnight for $3.99. Already a member? Sign in .

SEARCH INSIDE!™

Beginning JavaScript 2nd Edition

Share your own customer images

Search inside this book

Beginning JavaScript Second Edition (Paperback)

by Paul Wilton "In this introductory chapter, we'll take a look at what JavaScript is, what it can do for you, and what you need to be able..." (more)

Explore: Books on Related Topics | Concordance | Text Stats | SIPs | CAPs

Browse: Front Cover | Copyright | Table of Contents | Excerpt | Index | Back Cover | Surprise Me!

List Price: ~~$39.99~~

Price: $25.19 & this item ships for FREE with Super Saver Shipping. Details

You Save: $14.80 (37%)

Availability: Usually ships within 24 hours. Ships from and sold by Amazon.com.

Want it delivered Monday, April 24? Order it in the next 18 hours and 3 minutes, and choose **One-Day Shipping** at checkout. See details

59 used & new available from $19.94

(67 customer reviews) Sign in to rate this item.

Quantity: 1

Add to Shopping Cart

or

Sign in to turn on 1-Click ordering.

A9.com users **save 1.57%** on Amazon. Learn how.

More Buying Choices

59 used & new from $19.94

Have one to sell? Sell yours here

Add to Wish List

Add to Wedding Registry

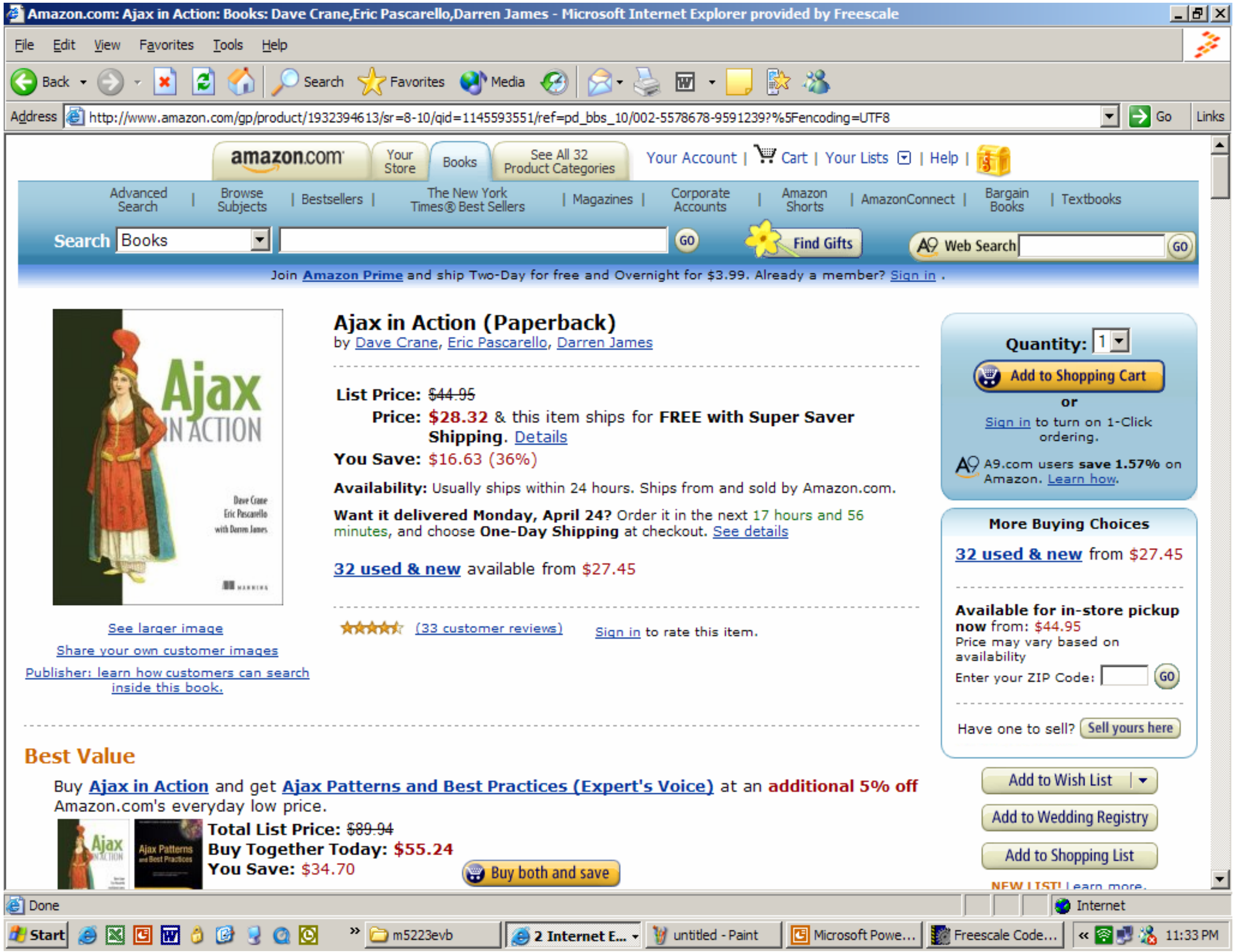
Add to Shopping List

NEW LIST! Learn more.

Tell a friend

Also Available in:	List Price:	Our Price:	Other Offers:
Paperback (Bargain Price)			14 used & new from $16.93
Paperback (New Ed)	~~$39.99~~	$25.19	47 used & new from $4.84
School & Library Binding			Order it used!
Digital (Download: Adobe Reader)	~~$39.99~~	$25.19	

Reference Material


Amazon.com: Ajax in Action: Books: Dave Crane,Eric Pascarello,Darren James - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.amazon.com/gp/product/1932394613/sr=8-10/qid=1145593551/ref=pd_bbs_10/002-5578678-9591239?%5Fencoding=UTF8
Go Links
amazon.com
Your Store
Books
See All 32 Product Categories
Your Account | Cart | Your Lists | Help |
Advanced Search | Browse Subjects | Bestsellers | The New York Times® Best Sellers | Magazines | Corporate Accounts | Amazon Shorts | AmazonConnect | Bargain Books | Textbooks
Search Books GO
Find Gifts
A9 Web Search GO
Done
Internet
Start
m5223evb
2 Internet E...
untitled - Paint
Microsoft Powe...
Freescale Code...
11:33 PM

Join **Amazon Prime** and ship Two-Day for free and Overnight for $3.99. Already a member? Sign in .

Ajax in Action (Paperback)

by Dave Crane, Eric Pascarello, Darren James

List Price: ~~$44.95~~

Price: $28.32 & this item ships for **FREE with Super Saver Shipping**. Details

You Save: $16.63 (36%)

Availability: Usually ships within 24 hours. Ships from and sold by Amazon.com.

Want it delivered Monday, April 24? Order it in the next 17 hours and 56 minutes, and choose **One-Day Shipping** at checkout. See details

32 used & new available from $27.45

(33 customer reviews) Sign in to rate this item.

See larger image

Share your own customer images

Publisher: learn how customers can search inside this book.

Quantity: 1

Add to Shopping Cart

or

Sign in to turn on 1-Click ordering.

A9.com users **save 1.57%** on Amazon. Learn how.

More Buying Choices

32 used & new from $27.45

Available for in-store pickup now from: $44.95

Price may vary based on availability

Enter your ZIP Code: GO

Have one to sell? Sell yours here

Add to Wish List

Add to Wedding Registry

Add to Shopping List

NEW LIST! Learn more.

Best Value

Buy **Ajax in Action** and get **Ajax Patterns and Best Practices (Expert's Voice)** at an **additional 5% off** Amazon.com's everyday low price.

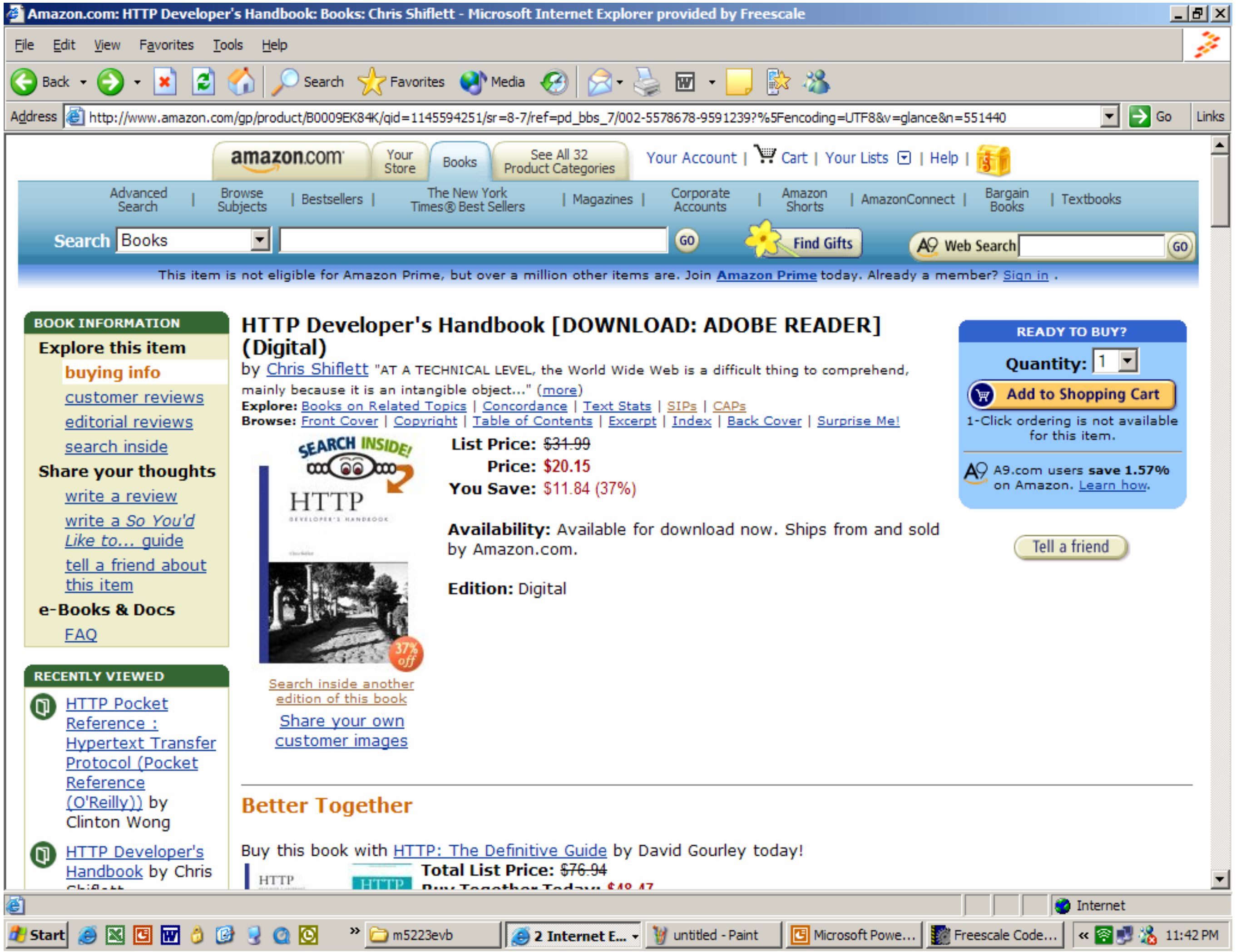
Total List Price: ~~$89.94~~

Buy Together Today: $55.24

You Save: $34.70

Buy both and save

Reference Material


Amazon.com: HTTP Developer's Handbook: Books: Chris Shiflett - Microsoft Internet Explorer provided by Freescale

File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Address
Go
Links

http://www.amazon.com/gp/product/B0009EK84K/qid=1145594251/sr=8-7/ref=pd_bbs_7/002-5578678-9591239?%5Fencoding=UTF8&v=glance&n=551440

amazon.com
Your Store
Books
See All 32 Product Categories
Your Account | Cart | Your Lists | Help |
Advanced Search | Browse Subjects | Bestsellers | The New York Times® Best Sellers | Magazines | Corporate Accounts | Amazon Shorts | AmazonConnect | Bargain Books | Textbooks
Search Books
GO
Find Gifts
A9 Web Search
GO

This item is not eligible for Amazon Prime, but over a million other items are. Join Amazon Prime today. Already a member? Sign in .

BOOK INFORMATION

Explore this item

- buying info
- customer reviews
- editorial reviews
- search inside

Share your thoughts

- write a review
- write a *So You'd Like to...* guide
- tell a friend about this item

e-Books & Docs

- FAQ

RECENTLY VIEWED

- HTTP Pocket Reference : Hypertext Transfer Protocol (Pocket Reference (O'Reilly)) by Clinton Wong
- HTTP Developer's Handbook by Chris Shiflett

HTTP Developer's Handbook [DOWNLOAD: ADOBE READER] (Digital)

by Chris Shiflett "AT A TECHNICAL LEVEL, the World Wide Web is a difficult thing to comprehend, mainly because it is an intangible object..." (more)

Explore: Books on Related Topics | Concordance | Text Stats | SIPs | CAPs

Browse: Front Cover | Copyright | Table of Contents | Excerpt | Index | Back Cover | Surprise Me!

Search inside another edition of this book

Share your own customer images

List Price: ~~$31.99~~

Price: $20.15

You Save: $11.84 (37%)

Availability: Available for download now. Ships from and sold by Amazon.com.

Edition: Digital

READY TO BUY?

Quantity: 1

Add to Shopping Cart

1-Click ordering is not available for this item.

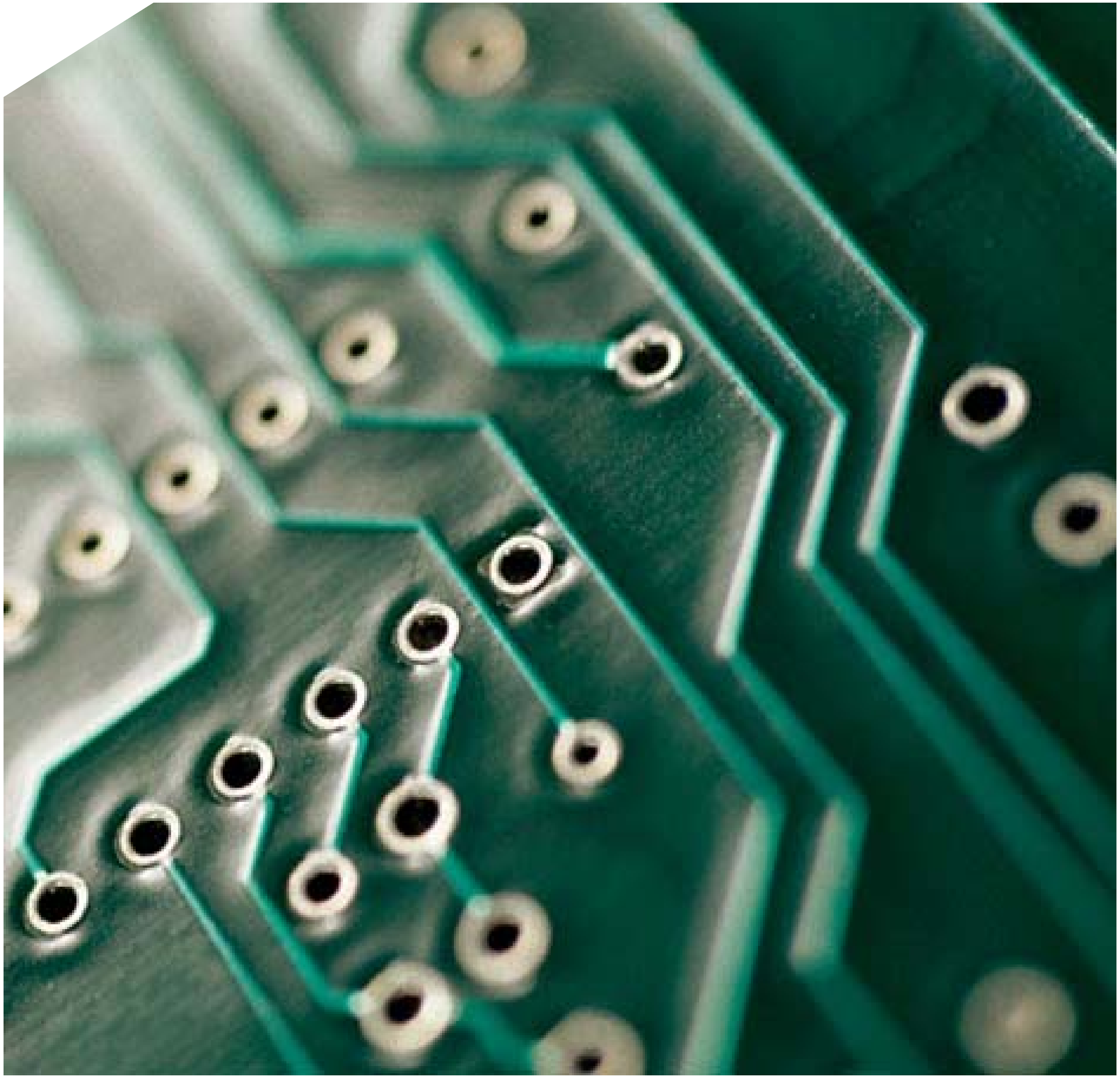
A9.com users **save 1.57%** on Amazon. Learn how.

Tell a friend

Better Together

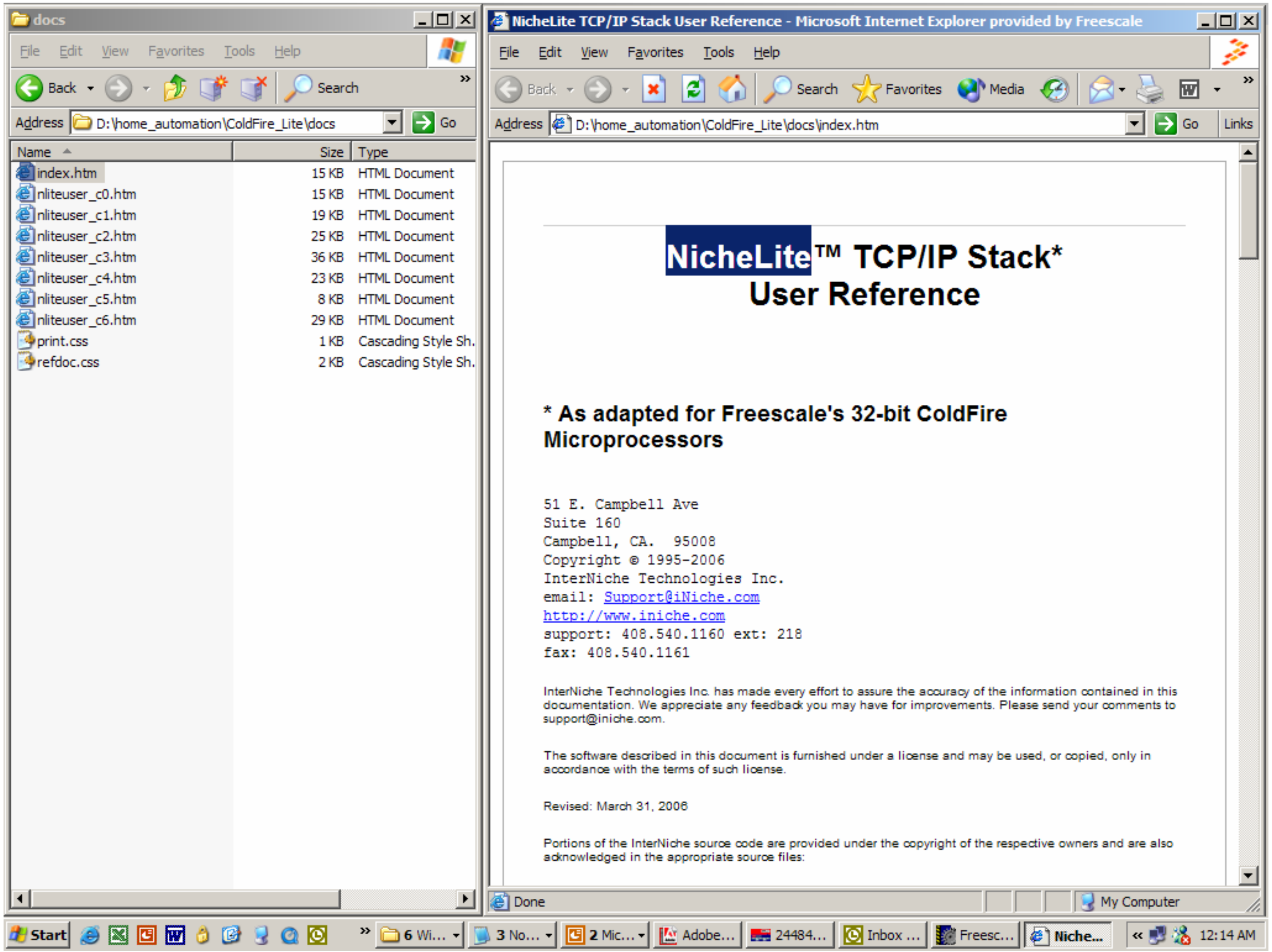
Buy this book with HTTP: The Definitive Guide by David Gourley today!

Total List Price: ~~$76.94~~



Firmware Overview

NicheLite Documentation can be found in the project


docs
File Edit View Favorites Tools Help
Back Search
Address D:\home_automation\ColdFire_Lite\docs
Go
NicheLite TCP/IP Stack User Reference - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Back Search Favorites Media
Address D:\home_automation\ColdFire_Lite\docs\index.htm
Go Links
Done
My Computer
Start
6 Wi... 3 No... 2 Mic... Adobe... 24484... Inbox ... Freesc... Niche...
12:14 AM

Name	Size	Type
index.htm	15 KB	HTML Document
nliteuser_c0.htm	15 KB	HTML Document
nliteuser_c1.htm	19 KB	HTML Document
nliteuser_c2.htm	25 KB	HTML Document
nliteuser_c3.htm	36 KB	HTML Document
nliteuser_c4.htm	23 KB	HTML Document
nliteuser_c5.htm	8 KB	HTML Document
nliteuser_c6.htm	29 KB	HTML Document
print.css	1 KB	Cascading Style Sh.
refdoc.css	2 KB	Cascading Style Sh.

NicheLite™ TCP/IP Stack*
User Reference

*** As adapted for Freescale's 32-bit ColdFire Microprocessors**

```
51 E. Campbell Ave
Suite 160
Campbell, CA.  95008
Copyright © 1995-2006
InterNiche Technologies Inc.
email: Support@iNiche.com
http://www.iniche.com
support: 408.540.1160 ext: 218
fax: 408.540.1161
```

InterNiche Technologies Inc. has made every effort to assure the accuracy of the information contained in this documentation. We appreciate any feedback you may have for improvements. Please send your comments to support@iniche.com.

The software described in this document is furnished under a license and may be used, or copied, only in accordance with the terms of such license.

Revised: March 31, 2006

Portions of the InterNiche source code are provided under the copyright of the respective owners and are also acknowledged in the appropriate source files:

The Serial Console Interface – type help at the INET> prompt

```
INET> help
SNMP Station: general commands:
   help     - help with menus
   state    - show current station setup
   delay     - set milliseconds to wait between pings
   host     - set default active IP host
   length   - set default ping packet length
   quit     - quit station program
   ping     - send a ping
   baud      - set serial console BAUD
   setip    - set interface IP address
   version  - display version information
   !command - pass command to OS shell
Also try 'help [general|diagnostic|EMG HTTP]'
INET>
```

Type help diag at the INET> prompt

```
INET> help diag
SNMP Station: diagnostic commands:
   arps     - display ARP stats and table
   buffers  - display free q buffer stats
   queues   - dump packet buffer queues
   dbytes   - dump block of memory
   debug    - set IP stack debug tracing
   dtrap    - try to hook debugger
   iface    - display net interface stats
   linkstats - display link layer specific stats
   tcp      - display TCP stats
   sockets  - display socket list
   tbconn   - tcp BSD connection stats
   tbsend   - tcp BSD send stats
   tbrcv    - tcp BSD receive stats
   allocsize - set size for alloc() breakpoint
   ipstat   - display IP layer stats
   icmpstat - display ICMP layer stats
   udp      - display UDP layer stats
   upcall   - trace received packets
   tkstats  - tasking system status
   users    - list all users
   adduser  - add a new user
INET>
```

HTTP Server Info

```
INET> help EMG
SNMP Station: EMG HTTP commands:
   dir        - Dir of EMG FFS
   flash_erase - Erase the dynamic FLASH area
   var         - Dynamic HTML variable
   http        - Dump HTTP sessions array
INET> http


HTTP sessions array Dump

STATE     VALID     KEEP_ALIVE     FILE_POINTER     SOCKET
Wait for header     Not Valid     0     0x0     0x0
Wait for header     Not Valid     0     0x0     0x0
Wait for header     Not Valid     0     0x0     0x0
Wait for header     Not Valid     0     0x0     0x0

INET>
```

Insight into the RTOS

```
INET> tkstats
tasking status:task wakeups: D
   name                  state       stack   used    wakes
console                  running      2048    536     1216676
EMG HTTP server          ready        2048    192     51859563
clock tick               sleeping     2048    104     42047
Main                     blocked      4096    392     0
INET>
```

Ethernet info – the iface command

```
INET> iface
Interface  - Fast Ethernet
Status; Admin:up Oper:up for: 8 minutes, 45 sec.
rcvd: errors:0   dropped:0   station:0   bcast:0   bytes:0
sent: errors:0   dropped:0   station:0   bcast:0   bytes:0
MAC address: 00 CF 52 23 00 00 ..R#..



Control Register                 = 3000

DATARATE = 100Mbps
ANE = Autonegotiation Enabled
DPLX = Half Duplexe

This register advertises the capabilities of the port to the MII
Status Register                   = 7849

Indicates the PHY supports 100BASE-TX full-duplex mode
Indicates the PHY supports 100BASE-TX half-duplex mode
Indicates the PHY supports 10BASE-T full-duplex mode
Indicates the PHY supports 10BASE-T half-duplex mode
No fault detected
PHY has auto-negotiation ability
valid link has NOT been established
AutoNegotiation NOT complete - Data is NOT Valid

Auto-Neg. Advertisement Register = 81E1

100BASE-TX full -duplex capable
100BASE-TX half-duplex capable
10BASE-T full-duplex capable
10BASE-T half-duplex capable

INET>
```

Adding a Your Own Command

```
//******************************************************************************
// Fill out structure for EMG FFS DIRectory menu command
//******************************************************************************
struct menu_op emg_ffs_dir_menu[] =            {

                "EMG HTTP",                     stooges,                        "EMG HTTP menu",

                "dir",                          emg_ffs_dir,                    "Dir of EMG FFS",

                "flash_erase",                  flash_erase,                    "Erase the dynamic FLASH area",

                "var",                          emg_http_var,                   "Dynamic HTML variable",

                "http",                         emg_http_sessions,              "Dump HTTP sessions array",

                NULL,
                                                };
```

```
    // Install Menu item 'DIR' for EMG FFS
    if( install_menu( emg_ffs_dir_menu ) )
                printf( "\nCould not install DIR menu item for EMG FFS" );
```

Commands are passed arguments

```
//**********************************************************************************
// int SoftEthernetNegotiation( int seconds )   Written By Eric Gregori
//
// Work-around for bug in hardware autonegotiation.
// Attempt to connect at 100Mbps - Half Duplexe
// Wait for seconds
// Attempt to connect at 10Mbps - Half Duplexe
//
// Returns 10, or 100 on success, 0 on failure
//**********************************************************************************
int   set_baud(void * pio)
{
   char    *cp;


   cp = nextarg(((GEN_IO)pio)->inbuf);
   iuart_set_baud( 0, atoi(cp) );

   return(0);
}
```

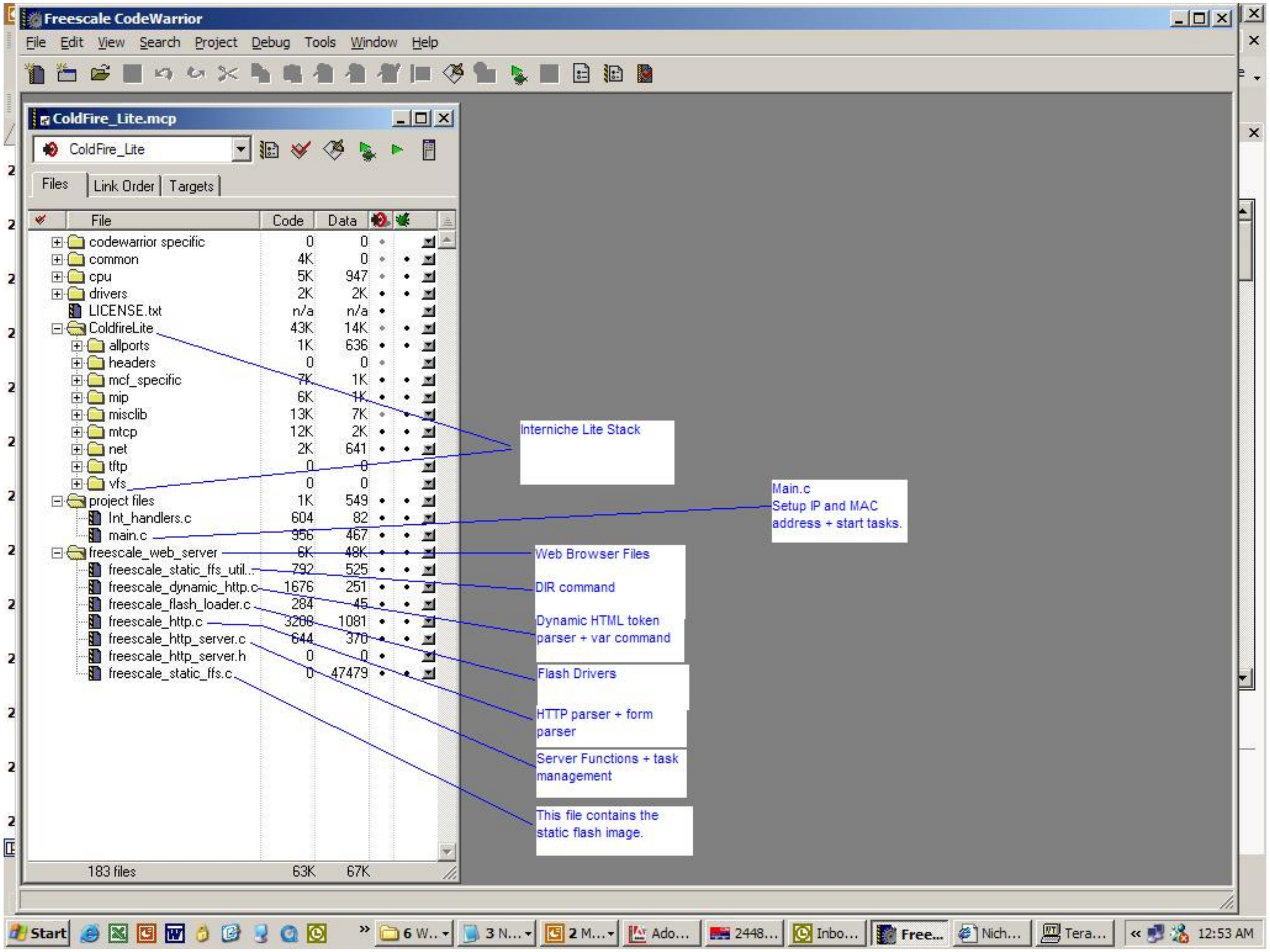
Printf is supported with formatting

```
//*******************************************************************************
// Print Directory of Static and Dynamic Flash File Systems.
//
// Author: Eric Gregori  (847) 651 - 1971
//*******************************************************************************
int emg_ffs_dir(void * pio)
{
    int                                         file_count, total_file_size, k, j;
    volatile unsigned long                      *fat_file_sys;
    volatile unsigned char                      *fat_file_names;


    ns_printf( pio, "\nStatic FFS" );
    ns_printf( pio, "\n\n%-32s %-6s %-8s",
                                                        "FILENAME",
                                                        "LENGTH",
                                                        " POINTER" );

    total_file_size = 0;

    // Loop through each file printing the info
    for( file_count=0; file_count<emg_static_ffs_nof; file_count++ )
    {
            ns_printf( pio, "\n%-33s", emg_static_ffs_filenames[file_count] );
            ns_printf( pio, "%-9d", emg_static_ffs_len[file_count] );
            ns_printf( pio, "0x%-8x", (unsigned long)emg_static_ffs_ptrs[file_count] );
            total_file_size += emg_static_ffs_len[file_count];
    }

    ns_printf(pio,"\n\n                     Total Size = %d",total_file_size);
    ns_printf(pio,"\ntotal static files = %d\n",file_count);

    ns_printf( pio, "\nDynamic FFS" );
    ns_printf( pio, "\n\n%-32s %-6s %-8s",
                                                        "FILENAME",
                                                        "LENGTH",
                                                        " POINTER" );
```

This package is ideal for remote testing

- Imagine this, you need a method to instrument a device you are testing.
- Just write your own command, or better yet put your data in a VAR, and you can access that data from anywhere in the world.
- This is a ideal platform for engineers to write small test programs, or build quick prototypes.
- The MCF5223 has:
  - 2 independent 4 channel 12 bit A/D converters
  - 8 PWM modules
  - 4   24 bit timers (can be used as pulse accumulators)
  - 1 16 bit timer
  - IIC, SPI, 3 UARTS, …..

Project Files



File	Code	Data		
codewarrior specific	0	0	•	
common	4K	0	•	•
cpu	5K	947	•	•
drivers	2K	2K	•	•
LICENSE.txt	n/a	n/a	•	
ColdfireLite	43K	14K	•	•
allports	1K	636	•	•
headers	0	0	•	
mcf_specific	7K	1K	•	•
mip	6K	1K	•	•
misclib	13K	7K	•	•
mtcp	12K	2K	•	•
net	2K	641	•	•
tftp	0	0		
vfs	0	0		
project files	1K	549	•	•
Int_handlers.c	604	82	•	•
main.c	956	467	•	•
freescale_web_server	6K	48K	•	•
freescale_static_ffs_util...	792	525	•	•
freescale_dynamic_http.c	1676	251	•	•
freescale_flash_loader.c	284	45	•	•
freescale_http.c	3208	1081	•	•
freescale_http_server.c	644	370	•	•
freescale_http_server.h	0	0	•	
freescale_static_ffs.c	0	47479	•	•
183 files	63K	67K		

Interniche Lite Stack

Main.c
Setup IP and MAC address + start tasks.

Web Browser Files

DIR command

Dynamic HTML token parser + var command

Flash Drivers

HTTP parser + form parser

Server Functions + task management

This file contains the static flash image.

Main.c

```
 /* hardcode FEC IP address for now. We set it in netstatic, and
    * Ip startup code will initialize net[] from it.
    */
#if 1  // EMG 192.168.1.99
   netstatic[0].n_ipaddr = (0xC0A80163);
   netstatic[0].n_defgw  = (0x00000000);
   netstatic[0].snmask   = (0xffffff00);
#else  //jpw  192.168.2.3
   netstatic[0].n_ipaddr = (0xC0A80203);
   netstatic[0].n_defgw  = (0xC0A80201);
   netstatic[0].snmask   = (0xffffff00);
#endif

   netstatic[0].mib.ifDescr = (u_char *)"Fast Ethernet Controller";

   /* We set the station's Ethernet physical (MAC) address
    * from the address already in use by dBUG. This prevents
    * ARP problems on the development server. Production systems
    * usually read this from flash or eprom.
    */

#ifdef USE_FEC
   tmp = 0x00cf5223;
   mac_addr_fec[0] = (u_char)(tmp >> 24);
   mac_addr_fec[1] = (u_char)(tmp >> 16);
   mac_addr_fec[2] = (u_char)(tmp >> 8);
   mac_addr_fec[3] = (u_char)(tmp & 0xff);
   tmp = 0;
   mac_addr_fec[4] = (u_char)(tmp >> 24);
   mac_addr_fec[5] = (u_char)(tmp >> 16);
#ifdef NPDEBUG
   dprintf("etheraddr = %02X:%02X:%02X:%02X:%02X:%02X\n\n",
            mac_addr_fec[0], mac_addr_fec[1], mac_addr_fec[2],
            mac_addr_fec[3], mac_addr_fec[4], mac_addr_fec[5]);
#endif
#endif
```

The HTTP Server Task

```
//******************************************************************************
// Declare Task Object
//******************************************************************************
TK_OBJECT(to_emghttpsrv);
TK_ENTRY(tk_emghttpsrv);
struct inet_taskinfo emg_http_task = {
                                        &to_emghttpsrv,
                                        "EMG HTTP server",
                                        tk_emghttpsrv,
                                        NET_PRIORITY,
                                        APP_STACK_SIZE
                                        };
long emghttpsrv_wakes = 0;



TK_ENTRY(tk_emghttpsrv)
{
   int err;

   while (!iniche_net_ready)
      TK_SLEEP(1);

   err = freescale_http_init();
   if( err == SUCCESS )
   {
      exit_hook(freescale_http_cleanup);
   }
   else
   {
      dtrap();                                                                  // emghttp_init()  shouldn't ever fail
   }

   for (;;)
   {
      freescale_http_check();                                                   // will block on select
      tk_yield();                                                               // give up CPU in case it didn't block

     emghttpsrv_wakes++;                                                                           //

      if (net_system_exit)
         break;
   }
   TK_RETURN_OK();
}
```

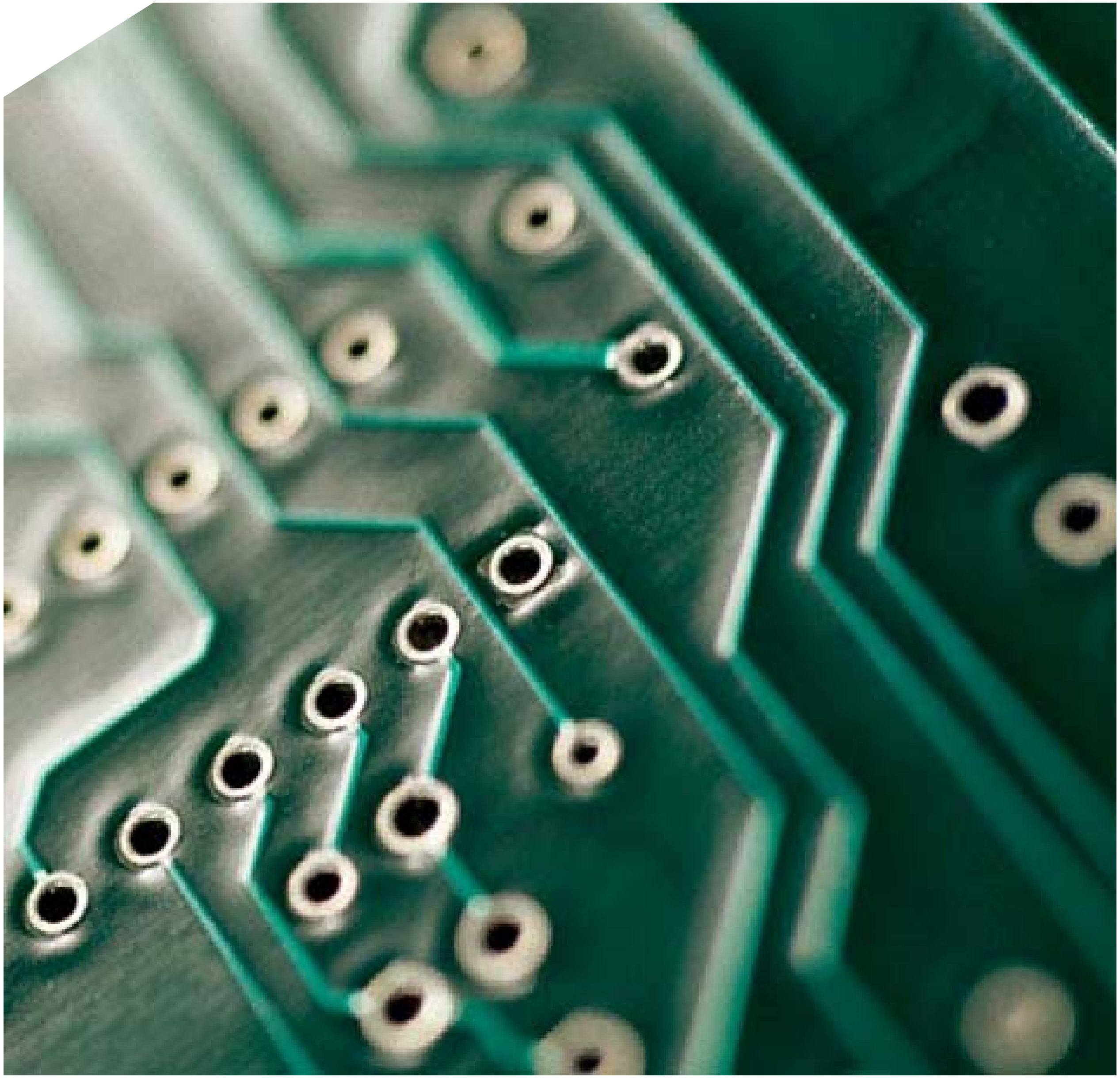
Enabling the DHCP client

In the file ipport.h you will find the following.

```
#define INCLUDE_ARP     1  /* use Ethernet ARP */
#define FULL_ICMP       1  /* use all ICMP || ping only */
#define OMIT_IPV4       1  /* not IPV4, use with MINI_IP */
#define MINI_IP         1   /* Use Nichelite mini-IP layer */
#define MINI_TCP        1   /* Use Nichelite mini-TCP layer */
#define MINI_PING       1   /* Build Light Weight Ping App for Niche Lite */
#define BSDISH_RECV     1   /* Include a BSD recv()-like routine with mini_tcp */
#define BSDISH_SEND     1   /* Include a BSD send()-like routine with mini_tcp */
#define NB_CONNECT      1  /* support Non-Blocking connects (TCP, PPP, et al) */
#define MUTE_WARNS      1  /* gen extra code to suppress compiler warnings */
#define IN_MENUS        1  /* support for InterNiche menu system */
#define NET_STATS       1  /* include statistics printfs */
#define QUEUE_CHECKING  1  /* include code to check critical queues */
#define INICHE_TASKS    1  /* InterNiche multitasking system */
#define MEM_BLOCKS      1  /* list memory heap stats */
// EMG #define TFTP_CLIENT     1  /* include TFTP client code */
// EMG #define TFTP_SERVER     1  /* include TFTP server code */
// EMG #define DNS_CLIENT      1  /* include DNS client code */
#define INICHE_TIMERS   1  /* Provide Interval timers */


// EMG - To enable DHCP, uncomment the line below
//#define DHCP_CLIENT     1  /* include DHCP client code */


// EMG #define INCLUDE_NVPARMS 1  /* non-volatile (NV) parameters logic */
#define NPDEBUG         1  /* turn on debugging dprintf()s */
// EMG #define VFS_FILES       1  /* include Virtual File System */
// EMG #define USE_MEMDEV      1   /* Psuedo VFS files mem and null */
#define NATIVE_PRINTF   1   /* use target build environment's printf function */
#define NATIVE_SPRINTF  1   /* use target build environment's printf function */
#define PRINTF_STDARG   1   /* build ...printf() using stdarg.h */
#define TK_STDIN_DEVICE 1   /* Include stdin (uart) console code */
#define BLOCKING_APPS   1   /* applications block rather than poll */
#define INCLUDE_TCP     1  /* this link will include NetPort TCP w/MIB */

/**** end of option list ***/
```

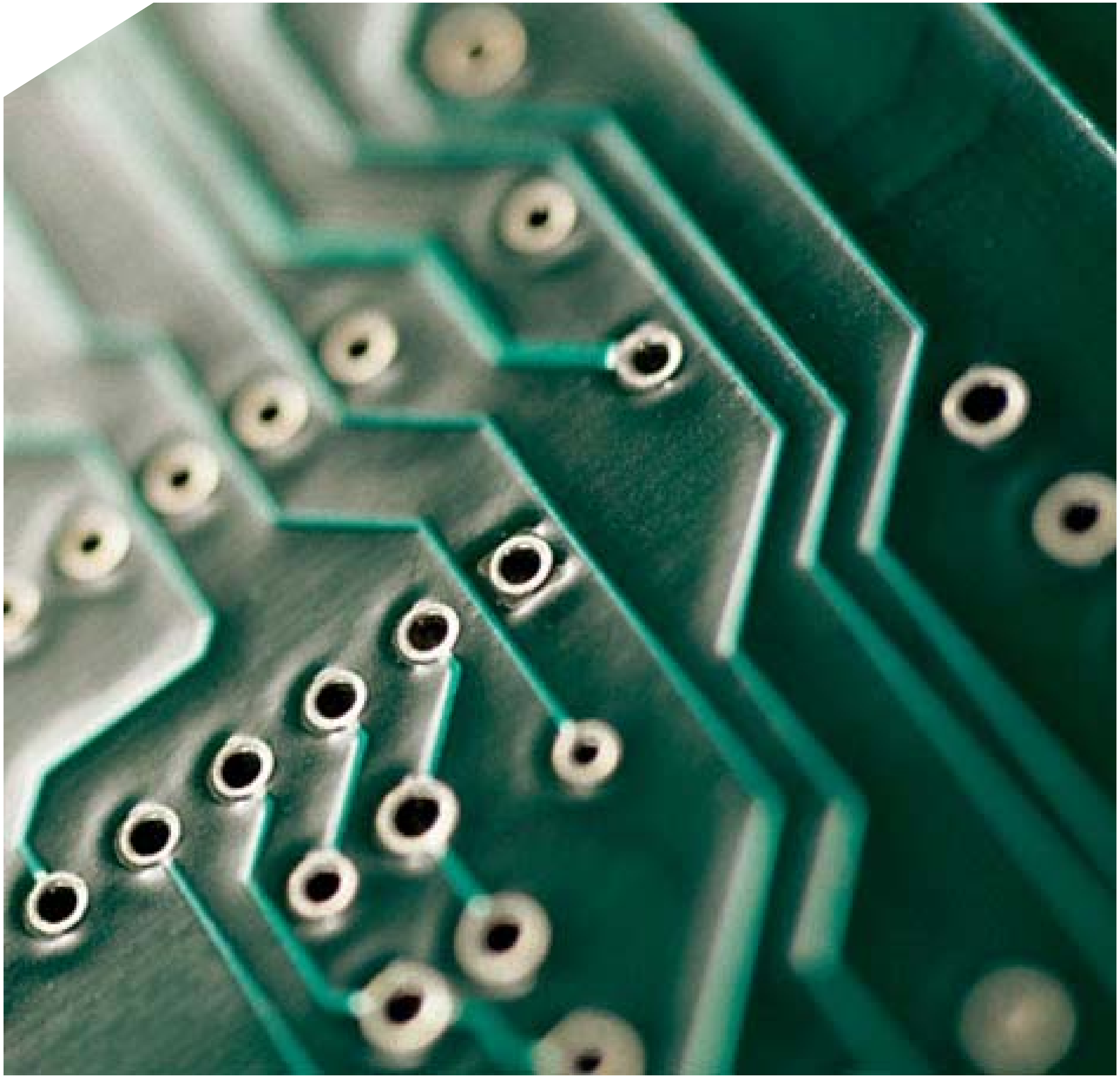
DHCP Demo

- Pushing SW1 at power-up will enable DHCP.



ZigBee/802.15.4 + ColdFire® Ethernet = A Winning Combination

freescale™ semiconductor



The Home Automation Demonstration

freescale™ semiconductor

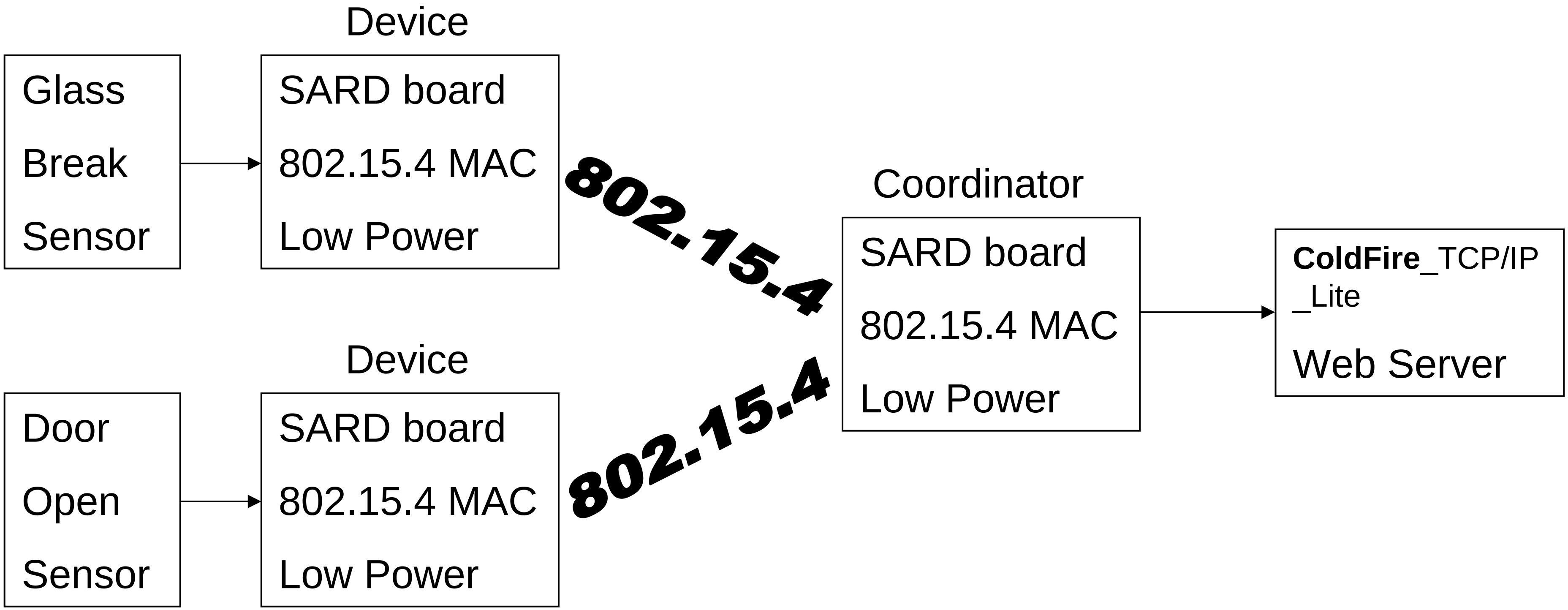
802.15.4 Sensor Network

- The firmware provided is a low power demonstration based on the Freescale 802.15.4 MAC.
- The coordinator is connected to the Web Server via a NULL modem adapter.
- The coordinator is actually a dumb serial passthrough.
- The device actually generates the var command based on the data from the sensor, and sends it to the coordinator. The coordinator simply takes any data received from any device and sends it through the serial port at 38400 baud.

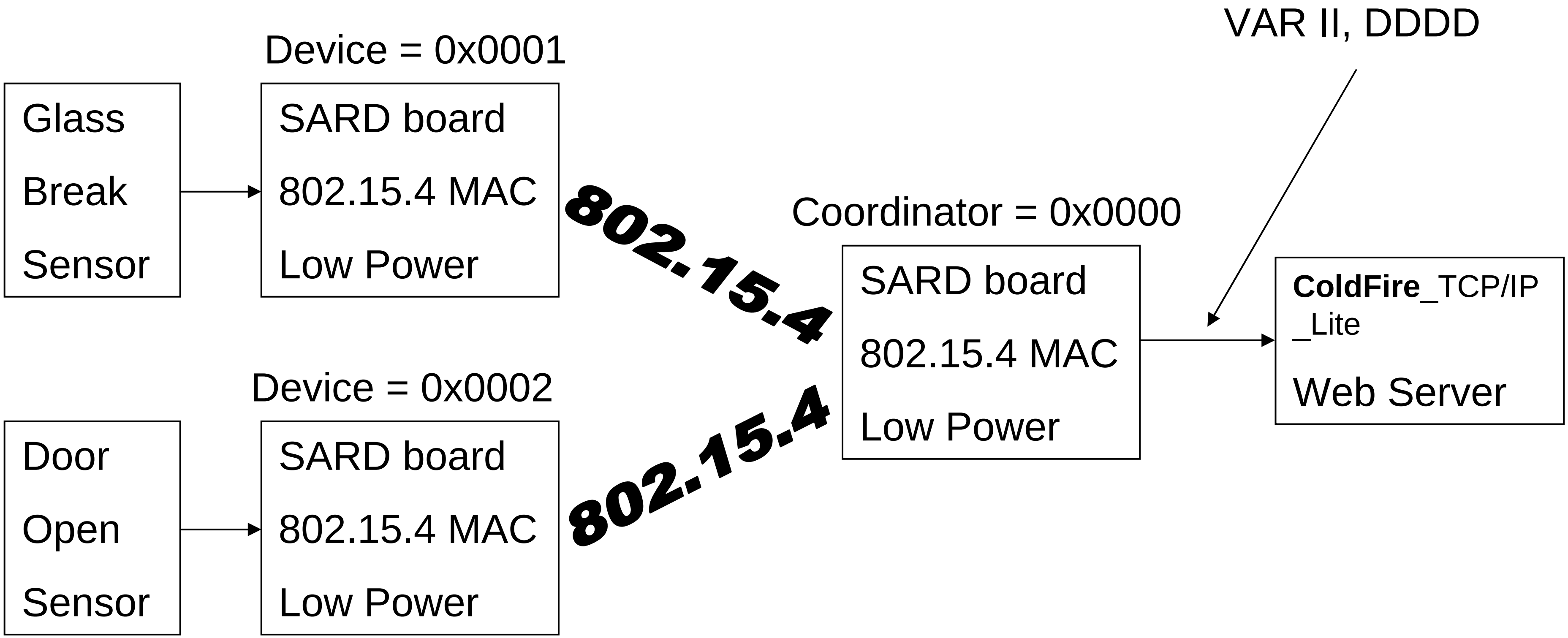
Two Sensor Types

- There is a rising and falling edge sensor type for the door and glass breakage detectors.
- The firmware is built for one or the other.
- The different sensor types send distinct codes to the web server through the coordinator.
- The web server can detect sensor type using JavaScript.
- This is demonstrated by the glass breakage sensor indicating a fault by the window, and the door edge sensor indicationg a fault by the door.
- This configuration is auto-detected.

System Diagram



Each Sensor is Assigned an Address at Power Up



Both Devices and coordinator share the same PAN ID.

This network is configured as a direct network with ACKS.

It can also be configured as a polled network.

Idd

- The sensors spend most of their time in hybernate mode.
- In hybernate mode, each sensor only draws 4µA.
- Each sensor wakes up every 5 seconds as a heartbeat, using the RTI.
- If the sensor detects a trigger, it wakes up immediately to send its data.
- Assuming less then one trigger every 5 seconds, each sensor should get a battery life of over 3 years using 2 AA's.
- The coordinator is always powered up.

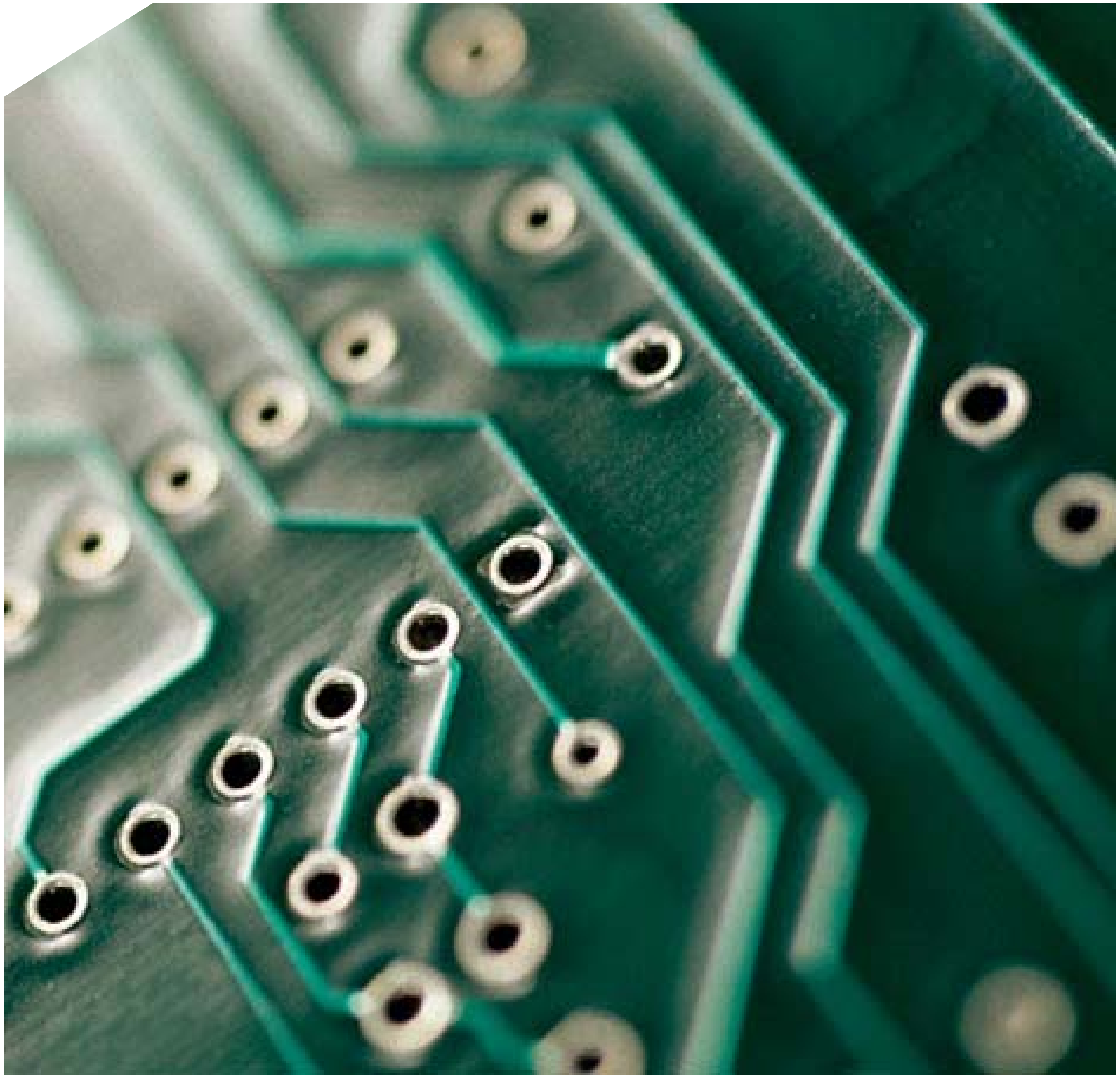
The Web Interface

- The web server provides a easy method of connecting external embedded systems over serial.
- The external embedded system can send data to the web server using the VAR command.
- The web server can send data over serial to the embedded system using forms.
- This provides a simple mechanism for getting your embedded system on the web.

Serial Autobaud

- The serial port on the Zigbee board is 38400, the ColdFire will automatically switch from 115200 to 38400.

- Show Coordinator output on serial - 38400
- Show ColdFire at 115200
- Connect



Questions, Answers and Consultations

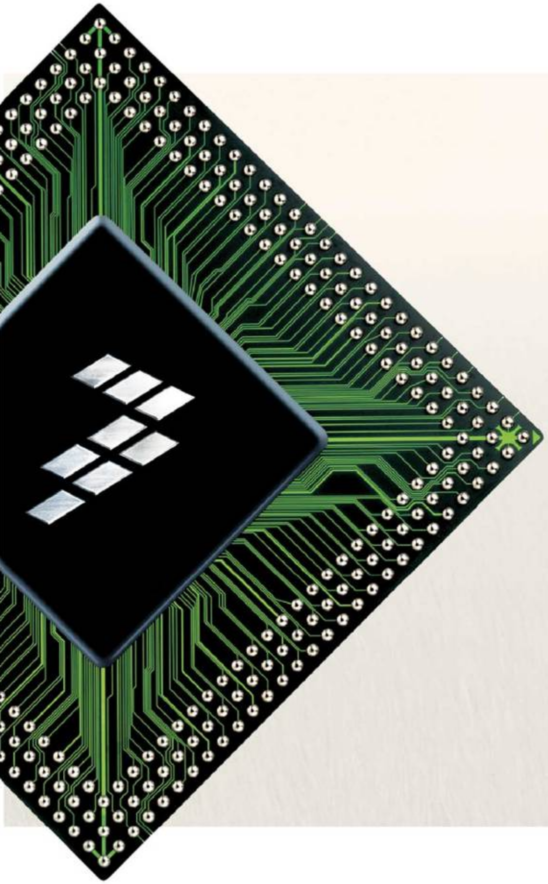
Continuum Shop Talk

To ask questions, discuss our topic further, or chat about the newest microcontroller technology…

Join me in Del Lago Room 1 for the

Controller Continuum Shop Talk

Wednesday, 9:30-10:30 am



Freescale Technology Forum

Design Freedom.