Embedded Web Server-Enabled Design Made Easy with Stellaris® MCUs

Miguel Emilio Oznaya Angeles
Texas Instruments
soporte@ti.com

INTRODUCTION TO STELLARIS ETHERNET HARDWARE AND SOFTWARE

Texas Instruments

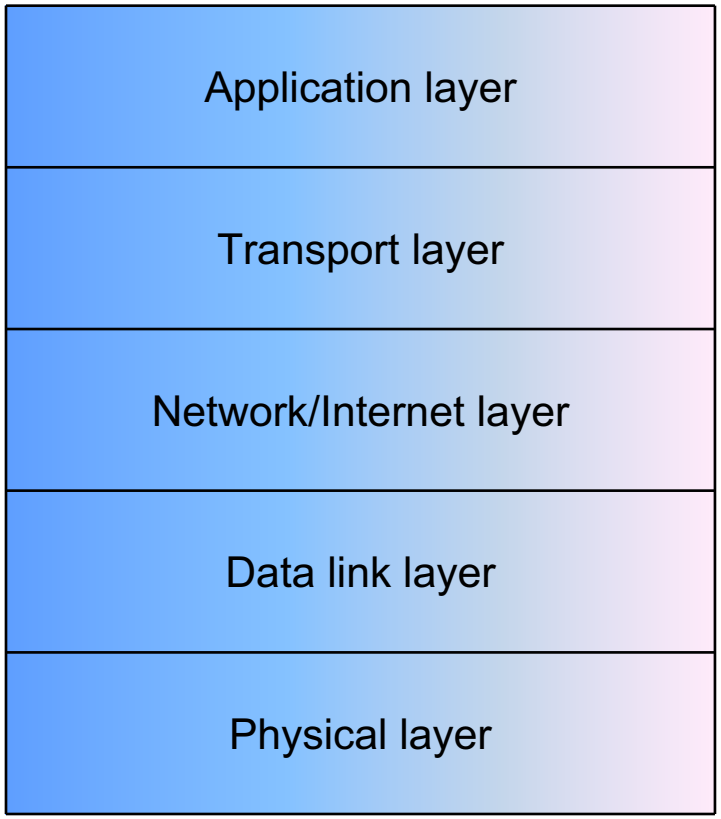
What is Ethernet?

- Ethernet refers to the family of local-area network (LAN) products covered by the IEEE 802.3 standard that defines what is commonly known as the CSMA/CD protocol

- Five data rates are currently defined for operation over optical fiber and twisted-pair cables:
  - 10Base-T Ethernet (1 Mbps/10 Mbps)
  - Fast Ethernet (100 Mbps)
  - Gigabit Ethernet (1000 Mbps)
  - 10-Gigabit Ethernet (10000 Mbps)
  - 100-Gigabit Ethernet (100000 Mbps)

TCP/IP Model

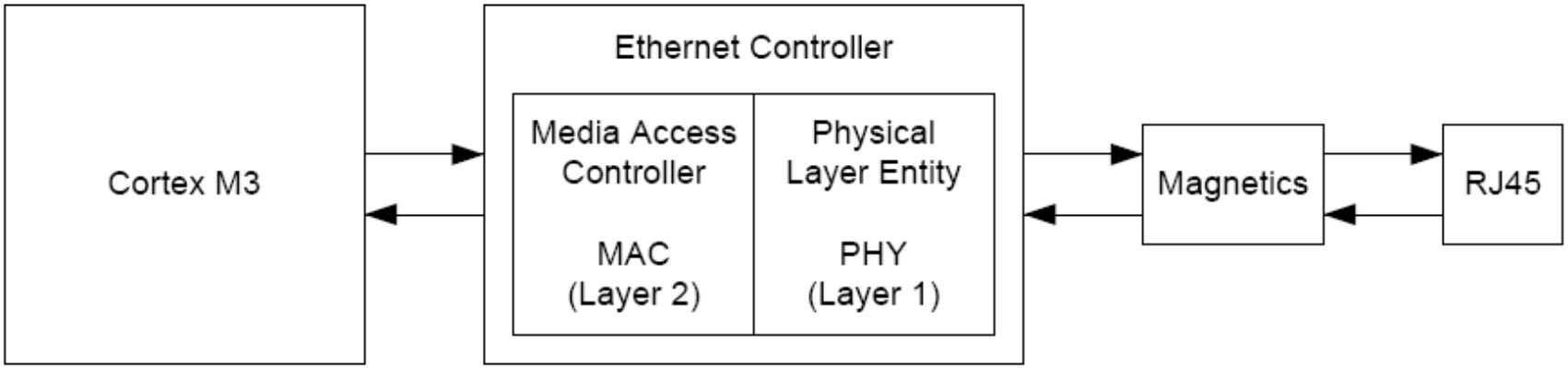
TCP/IP defines a set of rules to enable computers to communicate over a network, specifying how data should be packaged, addressed, shipped, routed and delivered to the right destination.



Five-Layer TCP/IP Model

- Application Layer: DHCP, DNS, FTP, HTTP, IMAP4, IRC, NNTP, XMPP, POP3, RTP, SIP, SMTP, SNMP, SSH, TELNET, RPC, RTCP, RTSP, TLS (and SSL), SDP, SOAP, GTP, STUN, NTP, etc...
- Transport Layer: TCP, UDP, DCCP, SCTP, RSVP, ECN, etc...
- Network/Internet Layer: IP (IPv4, IPv6), OSPF, IS-IS, BGP, IPsec, ARP, RARP, RIP, ICMP, ICMPv6, IGMP, etc...
- Data Link Layer: Ethernet, 802.11 (WLAN), 802.16, Wi-Fi, WiMAX, ATM, DTM, Token ring, FDDI, Frame Relay, GPRS, EVDO, HSPA, HDLC, PPP, PPTP, L2TP, ISDN, ARCnet, LLTD, etc...
- Physical Layer: Ethernet physical layer, Twisted pair, Modems, PLC, SONET/SDH, G.709, Optical fiber, Coaxial cable, etc...

Ethernet in Embedded Systems



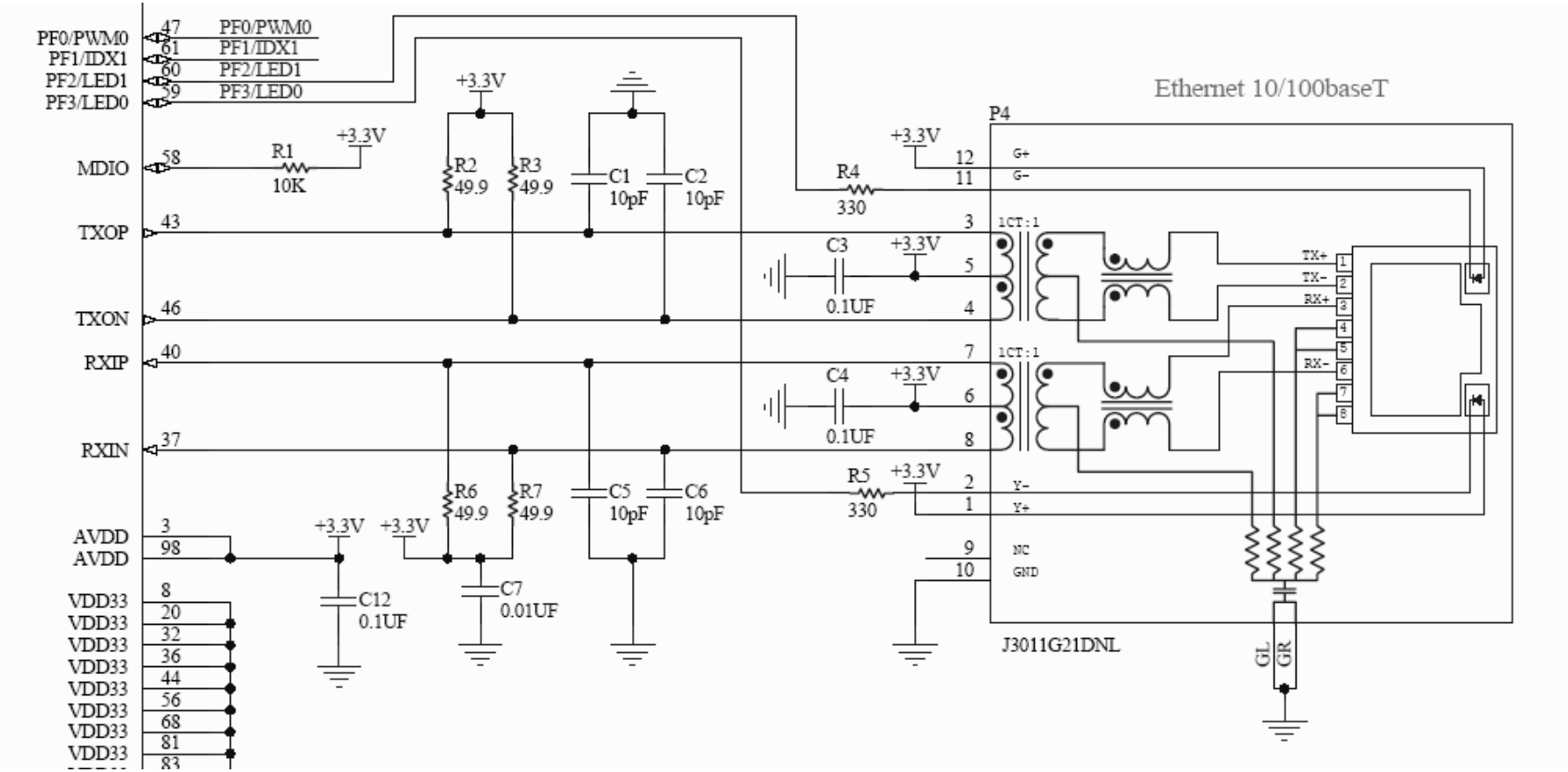
Media Access Controller (MAC) – Part of the Data Link Layer. The MAC provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multipoint network.

Physical Layer (PHY) - The most basic network layer, providing only the means of transmitting raw bits rather than packets over a physical data link connecting network nodes.

Magnetics (Isolation Transformer) - Part of the Physical layer used to decouple PHY from the physical Ethernet cable.

RJ45 - That “Ethernet” Connector.

Stellaris Ethernet Hardware: Simple Hardware Design



Third-Party Communications Stacks for Stellaris MCUs

µC/TCP-IP Protocol Stack	Micriµm µC/TCP-IP
expresslogic	Express Logic NetX™ TCP/IP protocol stack
CMX SYSTEMS	CMX-MicroNet™ protocol stacks
interniche technologies, inc.	InterNiche TCP/IP NicheStack™, NicheLITE™, and add-on modules such as HTTP, SNMP, and security protocols
REAL TIME AUTOMATION	EtherNet/IP™ protocol stacks
freeRTOS	FreeRTOS.org Open-Source µIP Embedded web server
µIP Embedded TCP/IP Stack	Open source TCP/IP stack for small footprint embedded systems
lwIP TCP/IP Stack	Open source light-weight implementation of the TCP/IP stack for small RAM embedded systems
IXXAT	IEEE 1588 PTP (Precision Time Protocol)
SEVENSTAX EMBEDDED TECHNOLOGIES	SEVENSTAX TCP/IP Protocol Stack

Open Source TCP/IP Stacks for Stellaris

- lwip
  - Protocols supported
    - Internet Protocol (IP) including packet forwarding over multiple network interfaces
    - Internet Control Message Protocol (ICMP) for network maintenance and debugging
    - User Datagram Protocol (UDP) including experimental UDP-lite extensions
    - Transmission Control Protocol (TCP) with congestion control, RTT estimations, and fast recovery/transmit
    - Dynamic Host Configuration Protocol (DHCP)
    - Point-to-Point Protocol (PPP)
    - Address Resolution Protocol (ARP) for Ethernet
    - Specialized raw API for enhanced performance
    - Optional Berkeley-like socket API
  - Memory Requirements
    - Typical code size is on the order of 25 to 40 kilobytes
    - RAM requirements are approximately 15 to a few tens of kilobytes
  - Website
    - http://www.sics.se/~adam/lwip
    - http://savannah.nongnu.org/projects/lwip

- uip
  - Protocols supported
    - Transmission Control Protocol (TCP)
    - User Datagram Protocol (UDP)
    - Internet Protocol (IP)
    - Internet Control Message Protocol (ICMP)
    - Address Resolution Protocol (ARP)
  - Memory requirements
    - Typical code size on the order of a few kilobytes
    - RAM usage can be as low as a few hundred bytes.
    - Memory conserved by limiting to one outstanding transmit packet
  - Website
    - http://www.sics.se/~adam/uip

- uip and lwip licenses
  - No restriction in shipping in real products
  - Redistribution of stack source or binaries (such as in our kit) must carry copyright

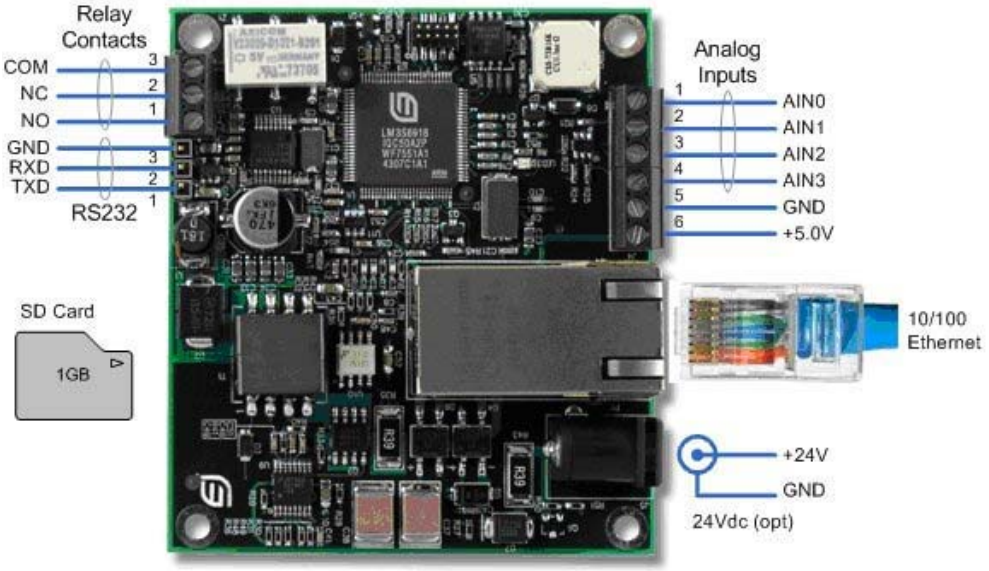
ETHERNET EXTRAS

Ethernet Extras: Power over Ethernet (PoE) – Stellaris RDK-IDM

Example applications:

- Security Systems & Building Access Controllers
- White Goods and other Home Appliances
- Factory Automation
- System Status and Configuration

- Bright QVGA LCD touch-screen display
  - 2.8” QVGA 240 x 320 pixels
  - 16-bit color
  - White LED backlight
  - Resistive touch panel
- Ethernet and Serial connectivity options
  - 10/100 Ethernet with Auto MDI/MDIX and Traffic /Link indicator LED
  - Header provides TXD and RXD signals
  - RS232 signal levels
  - Default 115.2k,8,n,1 operation
- High performance and memory
  - 32-bit ARM Cortex-M3 core
  - 256KB Main Flash memory, 64KB SRAM
  - 168K Image RAM
  - microSD slot (typically 1GB storage)
- Flexible power supply options
  - Power over Ethernet (IEEE 802.3af compliant)
  - 24V DC power jack
  - 5V DC terminals

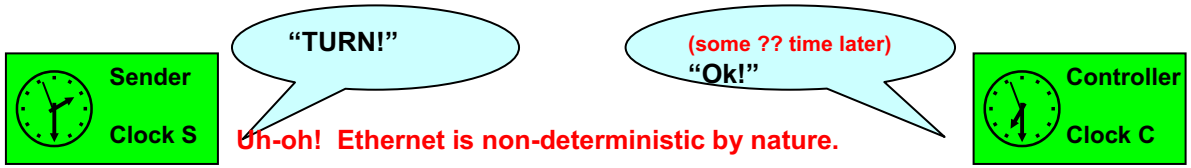


Ethernet Extras: IEEE 1588™ System

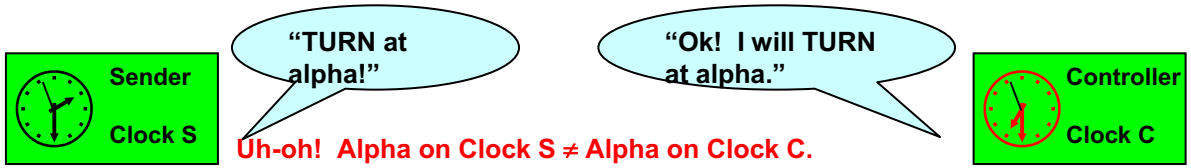
- An IEEE 1588 system is a collection of IEEE 1588 Clocks configured in such a way that all of the clocks synchronize with each other to maintain a consistent timescale.
- IEEE 1588 allows the clocks in the system components to synchronize to a high degree of accuracy.
- Microsecond accuracy is easily achievable using low cost, small footprint implementations such as Stellaris.
- How are the synchronized clocks used? The clocks in an IEEE 1588 system are typically used to coordinate the activities of the primary applications executing on the system.
- For example, sensor data may be time stamped at the source. Since all clocks are synchronized, the time stamped data may be correlated in post acquisition operations.
- The clocks may also be used in initiating actions in one or more components based on the times of the local clocks. For example an actuator could be commanded to change its value at time T and a sensor to measure a value at time T+delta.
- Since the clocks are synchronized, the resulting actions are coordinated in time.

Ethernet Extras: IEEE 1588™

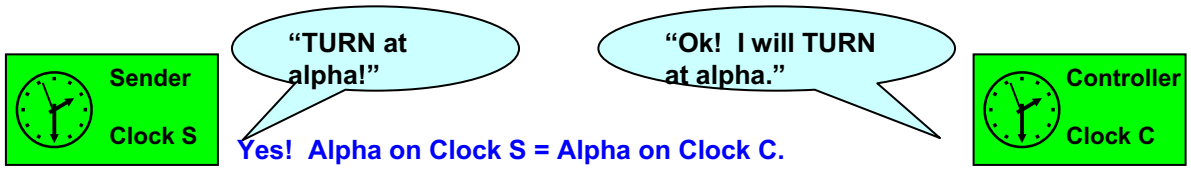
- Before IEEE 1588, Ethernet communication in control applications occurred without absolute determinism:
  - Assume *Sender* sends a control instruction *Turn* to *Controller*
  - Assume also that *Clock S* and *Clock C* are not synchronized
  - If *Sender* asks *Controller* to *Turn* upon receipt of the instruction, then there is no telling when *Controller* will receive *Turn*.



  - Even if *Sender* asks *Controller* to *Turn* at a given time *alpha*, there is still the problem of unsynchronized clocks.



  - But if *Sender* asks *Controller* to *Turn* at a given time *alpha*, and the clocks are synchronized to a master, then determinism is achieved.



Ethernet Extras: PTP in Industrial Applications

- Industry synchronization requirements for PTP[A]

Application area	Required synchronization accuracy
Low speed sensors (e.g. pressure, temperature)	Milliseconds
Common electro-mechanical devices (e.g. relays, breakers, solenoids, valves)	Milliseconds
General automation (e.g. materials handling, chemical processing)	Milliseconds
Precise motion control (e.g. high speed packaging, printing, robotics)	A few microseconds
High speed electrical devices (e.g. synchrophasor measurements)	Microseconds
Electronic ranging (e.g. fault detection, triangulation)	Sub microsecond

- PTP and motion control
  - Variable frequency drives require few 10s of microseconds
    - Software generally 5uS
    - 44% of applications are networked, 63% use Ethernet TCP/IP[B]
  - Servo-controlled systems require 100s of nanoseconds
    - Requires significant hardware assist
    - 36% of applications are networked, 56% use Ethernet TCP/IP[C]
- Stellaris implementation
  - Open source lwIP + PTPd : within 500nS of master clock, jitter +/- 500nS
  - This represents a greater than ten fold improvement over typical SW-only implementations.

[A] 2002, http://ieee1588.nist.gov/PTTI_draft_final.pdf
[B] 2003, http://www.controleng.com/article/CA315281.html
[C] 2004, http://controleng.com/article/CA407419.html?text=1588

INTRODUCTION TO EK-LM3S6965

Texas Instruments

Stellaris Evaluation Kits

- **Everything a developer needs to get up and running in 10 minutes or less**
  - Each kit includes: evaluation board(s), all required cables, a choice of evaluation tools suites for popular development tools, documentation, StellarisWare® software, and applications notes

EK-LM3S811
Low pin count
49 USD

EK-LM3S1968
High pin count
59 USD

EK-LM3S2965
CAN Functionality
79 USD

EK-LM3S3748
USB Host/Device
109 USD

EK-LM3S6965
Ethernet MAC+PHY
69 USD

EK-LM3S8962
Ethernet+CAN
89 USD

EK-LM3S9B90
Ethernet+USB OTG
99 USD

EK-LM3S9B92
Ethernet+OTG+MC
99 USD

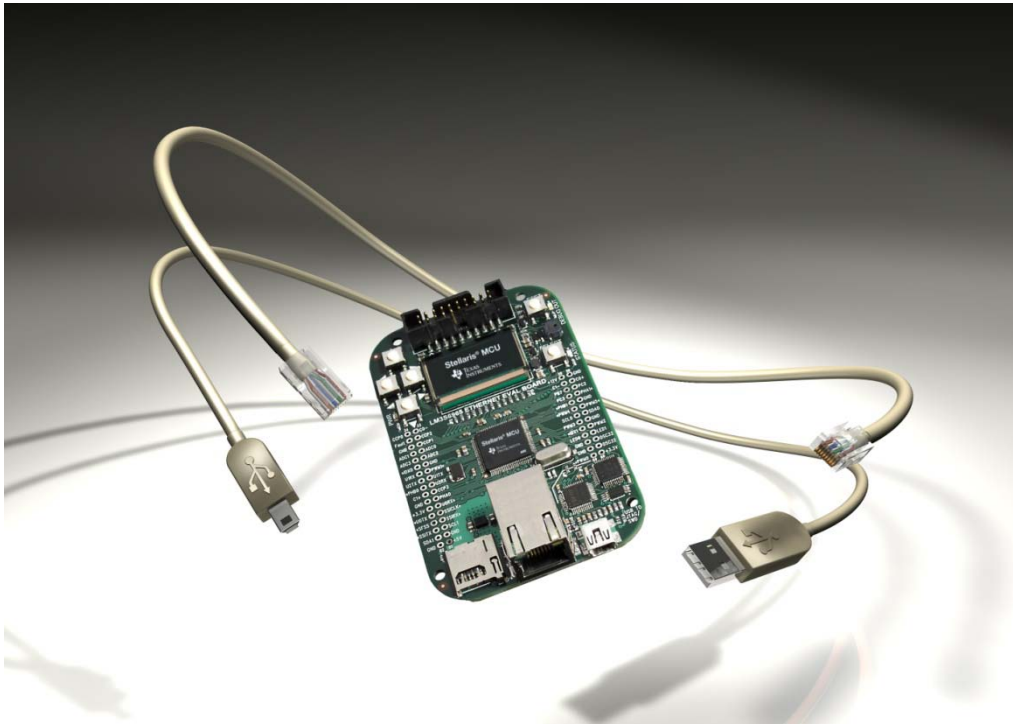
- **Evaluate • Prototype • Debug**: Each kit spans the design spectrum by functioning both as an evaluation platform and as a serial in-circuit debug interface for any Stellaris microcontroller-based target board:

- **Five versions of each kit:**
  - EKK-LM3Sx
    - ARM RealView Microcontroller Development Kit tools with 32KB address Limit
  - EKI-LM3Sx
    - IAR Embedded Workbench KickStart with 32KB address limit
  - EKC-LM3Sx
    - CodeSourcerySourcery G++ GNU with 30-day evaluation license
  - EKT-LM3Sx
    - Code Red Technologies Red Suite with 90-day evaluation license
  - EKS-LM3Sx
    - TI Code Composer Studio with full evaluation license locked to board

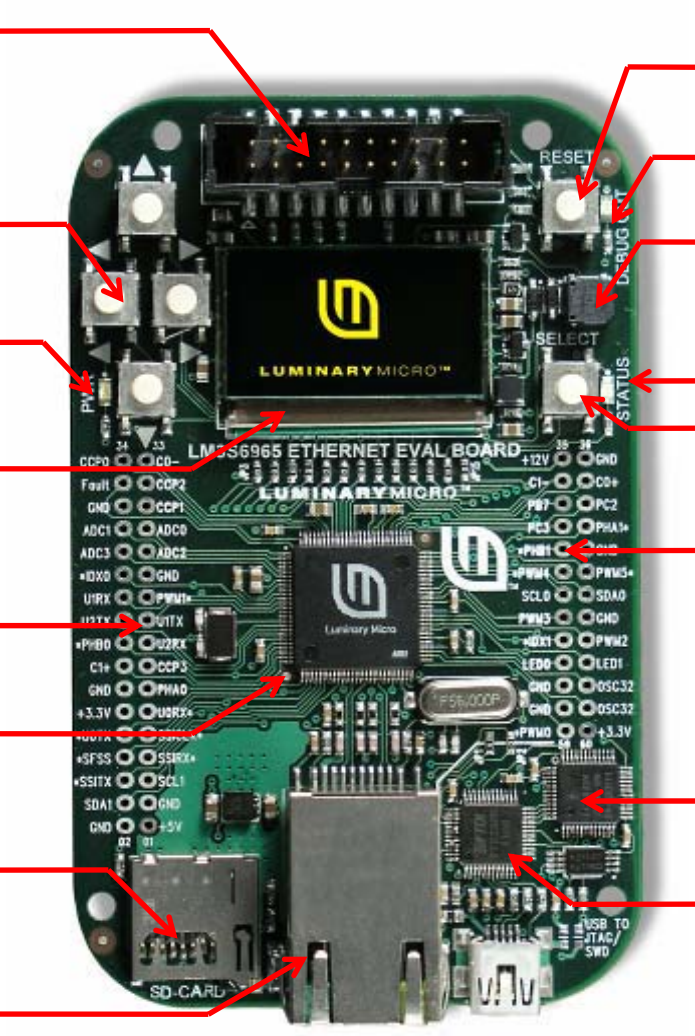
EK-LM3S6965: Evaluation Kit Overview

Stellaris LM3S6965 Evaluation Kit:

- LM3S6965 Evaluation Board
  - Stellaris LM3S6965 microcontroller with fully-integrated 10/100 Ethernet controller
  - Simple setup
  - OLED graphics display with 128 x 64 pixel resolution
  - User LED, navigation switches, and select pushbuttons
  - Magnetic speaker
  - LM3S6965 I/O available on labeled break-out pads
  - Standard ARM® 20-pin JTAG debug connector with input and output modes
  - MicroSD card slot
- Included µIP and lwIP IP stacks with Web Servers
- Retractable Ethernet Cable, USB cable, and JTAG cable
- CD containing:
  - Evaluation software tools
  - Device documentation
  - quickstart guide,
  - Stellaris Peripheral Driver Library
  - Example source code



EK-LM3S6965: Hardware Features



JTAG/SWD Input and Output

Navigation Buttons

Power LED

OLED Graphics Display

34 Pin I/O Breakout Header

Stellaris® LM3S6965

microSD Memory Slot

10/100BaseT Ethernet Jack

Reset Switch

Debug Out LED

Speaker

User Status LED

User Push Button

26 pin I/O Breakout Header

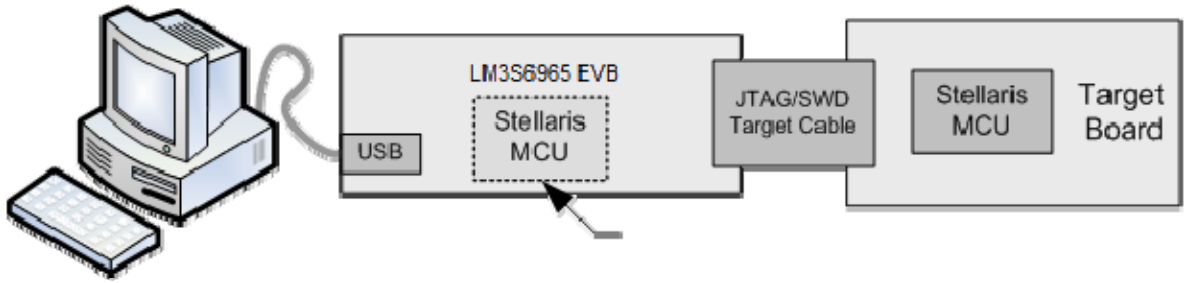
In-circuit Debug Interface

USB Interface

EK-LM3Snnn: Debug Interface Mode

- All Stellaris evaluation boards can be used as an In-Circuit Debugger Interface (ICDI).

- The board can be used to debug other Stellaris hardware such as a custom board. This feature is supported by CCS, LM Flash Programmer, Keil, IAR, CodeSourcery, and Code Red tools.



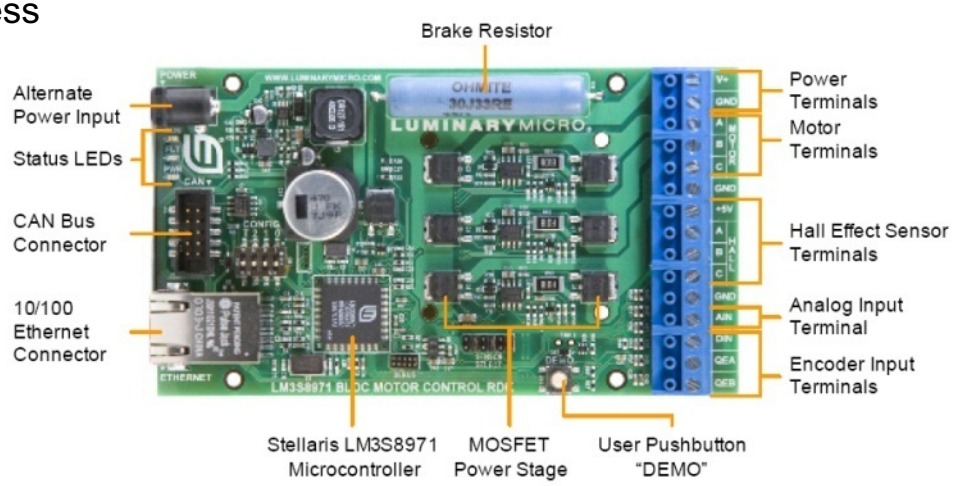
EMBEDDED ETHERNET CONTROL EXAMPLES

Embedded Ethernet Control: Motor Control

Example applications:

- **Factory automation**
- **Small appliances**
- **Electric wheelchairs and mobility devices**
- **Pumping and ventilation systems**

- Advanced motor control for three-phase brushless DC motors up to 36 V 500 W
- Flexible platform accelerates integration process
- Uses a Stellaris LM3S8971 microcontroller
- 10/100 Ethernet and CAN interfaces
- Four quadrant operation for precise control
- Hall Effect, Quadrature, and Sensorless operation modes
- On-board braking circuit
- Incremental quadrature encoder input
- Analog and digital control inputs
- Status LEDs indicate Power, Run, and Fault conditions
- Optional power-managed fan for forced-air cooling
- JTAG/SWD port for software debugging



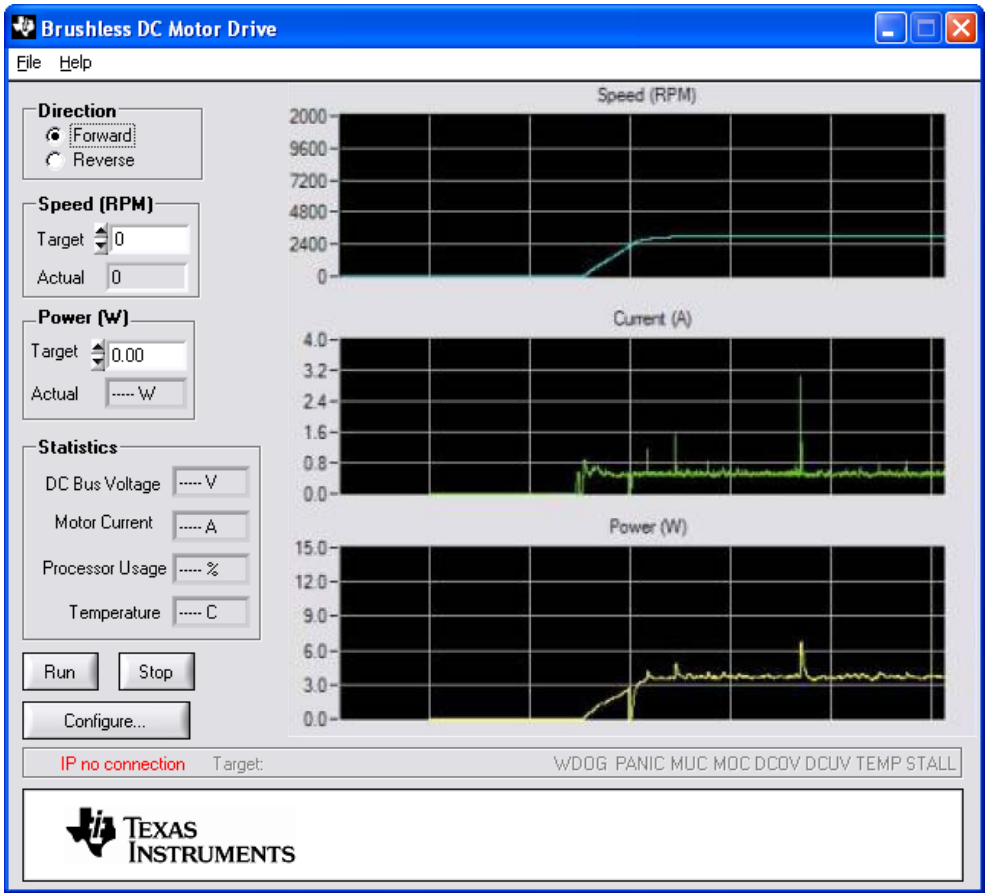
RDK-BLDC resale: 219 USD
MDL-BLDC single unit resale: 149 USD

Any Stellaris evaluation kit can function as an ARM Cortex-M3 USB-to-JTAG emulator.

Texas Instruments

Embedded Ethernet Control: Motor Control GUI

- Optional PC-side GUI based on LabWindows
- Configure motor capabilities and safety parameters
- Test controls and effects
- Understand tradeoffs in end motor system design
- View system statistics



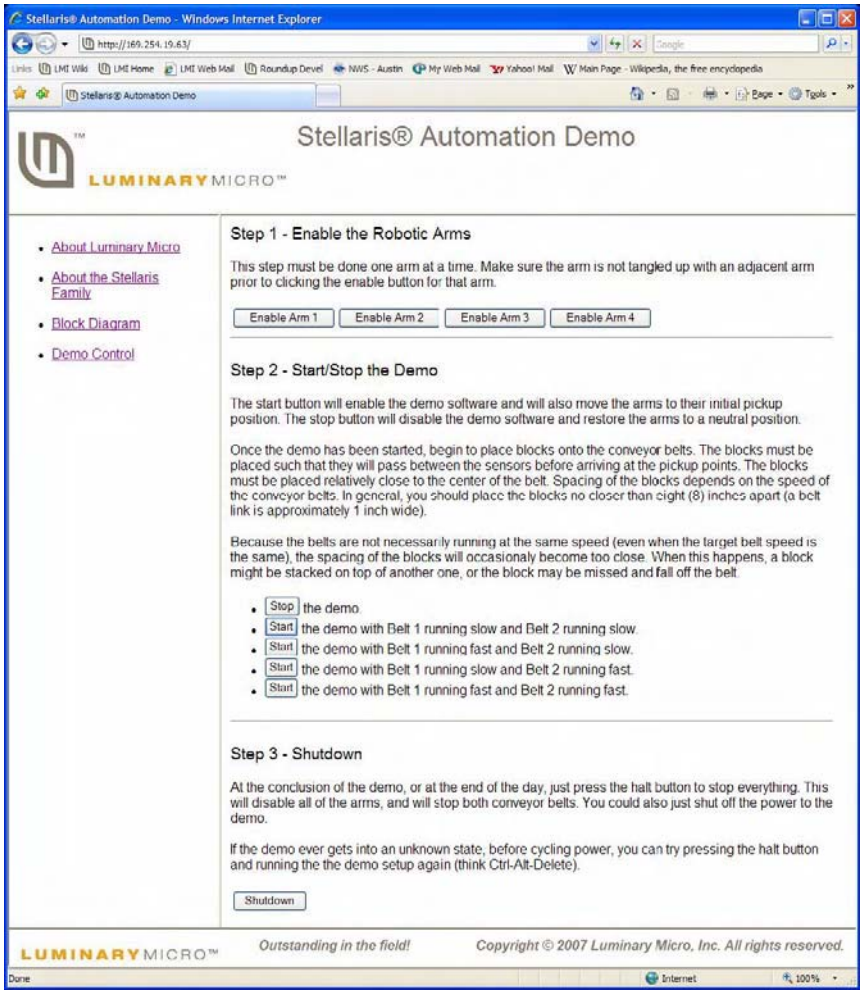
Embedded Ethernet Control: Serial-to-Ethernet

Example applications:

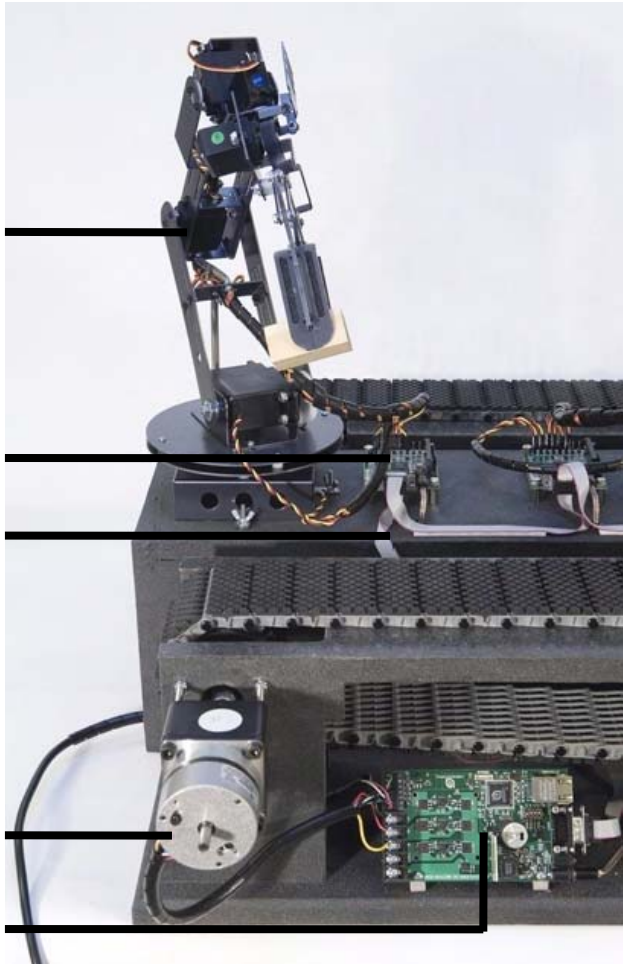
- **SCADA Remote Terminal Units (RTUs)**
- **Electronic Flow Meters (EFMs)**
- **Medical Point-of-Care and Retail Point-of-Sales Machines**
- **CCTV RS-232 Recorders**
- **RS-232 Stepper Motor Controller Systems**

- LM3S6432 in a 10 x 10 mm BGA package for reduced board size
- 10/100 Mbit Ethernet port
  - Auto MDI/MDIX cross-over correction
  - Traffic and link indicators
- Serial ports
  - UART0 has RS232 levels, transceiver runs at up to 250 Kbits/sec
  - UART1 has CMOS/TTL levels, can run at 1.5 Mbits/sec
  - UART ports include RTS/CTS for flow control
- Software
  - IP configuration with static IP address or DHCP
  - Telnet server for access to serial port
  - Web server for module configuration
  - UDP responder for device discovery
  - Telnet client for Ethernet-based serial port extender
- Module supports 5 V and 3.3 V supplies
- JTAG port pads for factory programming

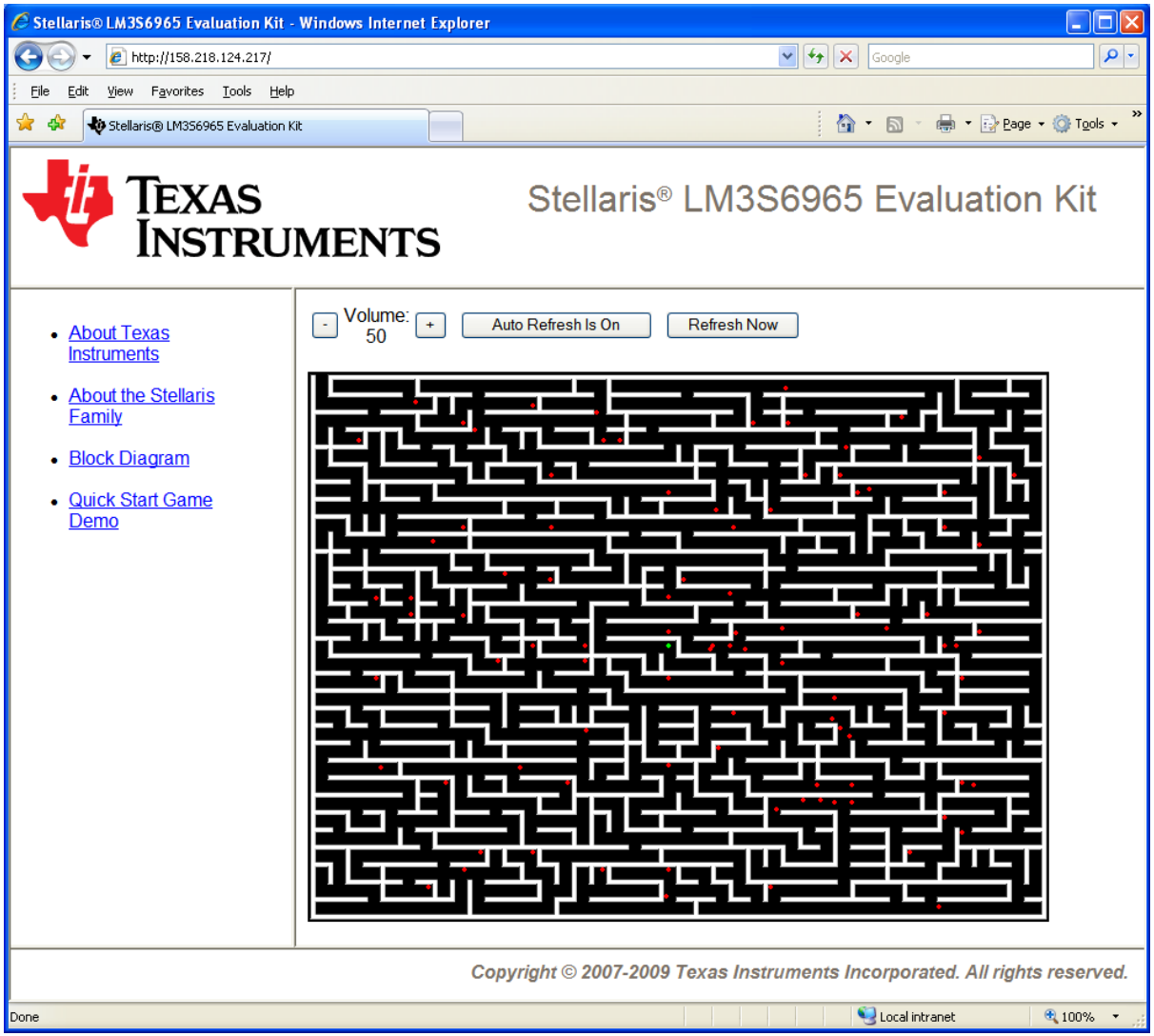
Embedded Ethernet Control: Automation Demo



Web Browser Console for Automation System Demo



Quickstart Web Server – Game Mode



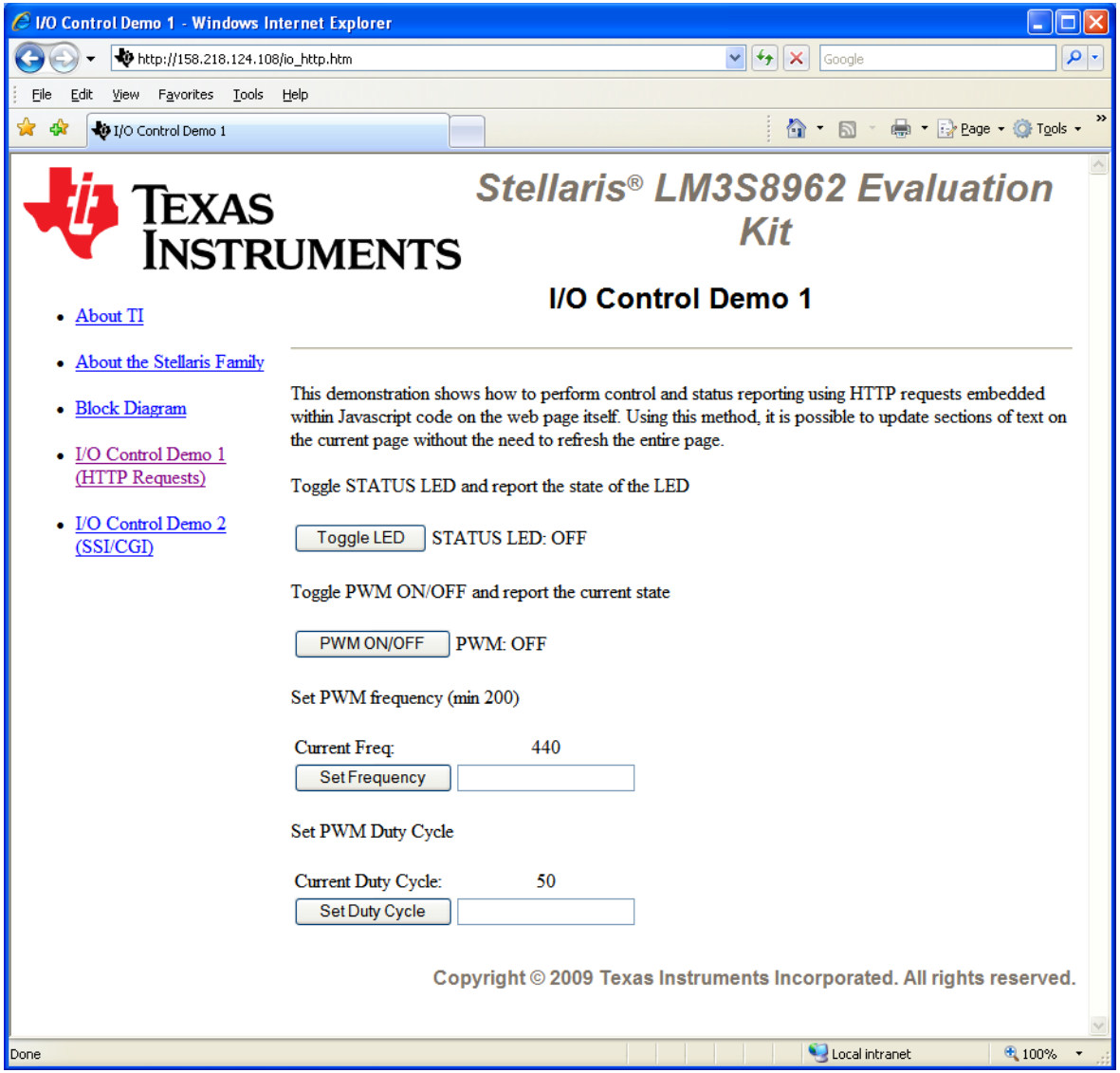
Embedded Control Web Page

- enet_io example
- Turns on LED and PWM using web GUI interface
- Main page is shown here

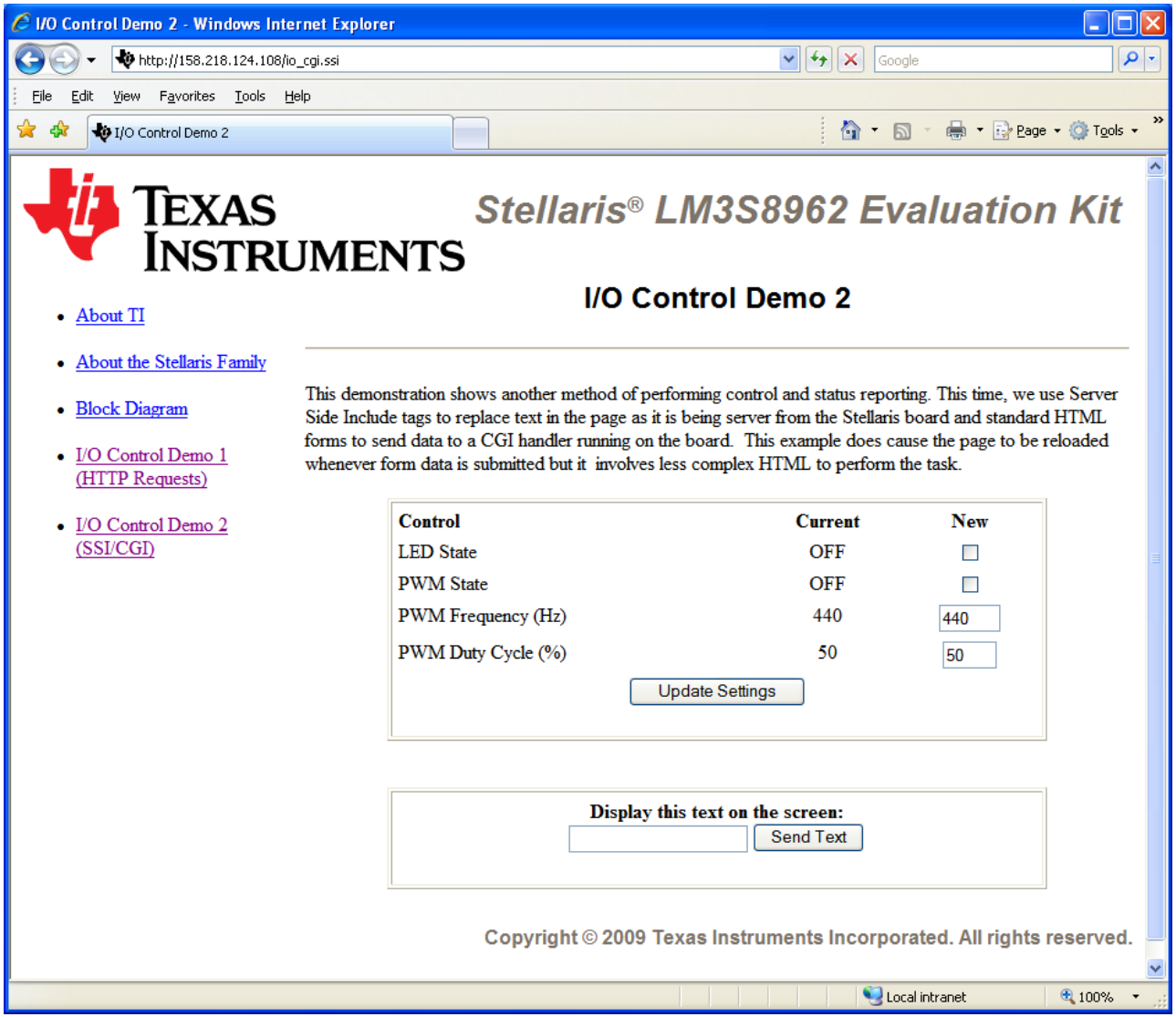
I/O Control Demonstrations

- The enet_io application in the StellarisWare board examples illustrates two methods of controlling board operations from the web browser:
  - I/O Control Demo 1 shows direct HTTP requests generated via JavaScript code in the web page (io_http.html).
  - I/O Control Demo 2 shows the use of Server Side Includes (SSI) and Common Gateway Interface (CGI) to perform the same operations (io_cgi.shtml)
- All web site files are stored as a file system image (lmi_fs.c) linked into the application image in flash.

I/O Control Demo 1



Hands On: Demo 2 – SSI/CGI



THANK YOU.
QUESTIONS?