LUMINARY MICRO

Integrating Web Servers in Embedded Applications

by Kyle Norman, Luminary Micro Applications Engineer

Answering the call for high performance, easy-to-use, and cost-effective technology for web server-capable embedded applications, Luminary Micro is leading the charge with Ethernet-enabled Stellaris microcontrollers. Based on the ARM® Cortex™-M3 core, the award-winning Stellaris® family provides exceptional processor horsepower, ample single-cycle memory for efficient Ethernet traffic handling, high integration to connect to virtually any embedded interface, and optimization for low power.

Overview

The web browser has had a profound positive effect on consumers–raising consumer expectations in terms of how quickly and easily information can be made available from virtually anywhere in the world. While web browser technology and usage models have literally saturated the consumer market segments, the fundamental values of what caused worldwide adoption of web browsers, such as simple user interface, accessibility to remote content, and speed, are actively advancing applications far from consumer segments, such as factory automation, building control, data transaction, and the medical field. In the same way that a consumer can securely access a bank to check on an account balance, a petroleum engineer in Lafayette, Louisiana, can monitor the pressure of a crude oil valve on a remote oil rig in the middle of the Gulf of Mexico. And just as easily as a child can play learning games on her favorite children's web site, the petroleum engineer is able to control the valve settings and adjust pressure with the click of a mouse.

The idea of a web browser to monitor and control embedded applications is not a new concept, but the traditional high cost to overhaul a legacy system with capable technology has stunted embedded adoption. Fortunately, as Moore's Law yields yet again, lithography generation and transistor realty has finally reached a point where a given technology is affordable. Now, with a fundamental understanding of how a web server is expected to perform in a given application, along with knowledge of integral hardware and software pieces required to complete a cost- and feature-effective web server system, high-performance web server capability is easily within reach for embedded designers.

The Ethernet-enabled Stellaris microcontrollers integrate peripherals that are designed for meticulous motion control, meaning that a Stellaris Ethernet-enabled microcontroller is concurrently capable of serving a web browser while driving a mission-critical motor. To simplify web server implementation on Stellaris microcontrollers, Luminary Micro offers three versions of an Ethernet evaluation kit, each including complete web server demonstrations using both the uIP and lwIP TCP/IP open source Ethernet protocol stacks, as well as all the hardware, software, peripheral drivers, development tools, and documentation needed. Utilizing the benefits of the Stellaris Ethernet evaluation kit and realizing the unique advantages that Stellaris technology offers to a web server application enables embedded designers to provide effective web server products with fast time-to-market.

Definitions: Web Server Infrastructure Components

Before discussing the implementation options for creating a web server in an embedded application, we need to first define the software infrastructure components of a web server system. This includes the encoding of a web page being served and the protocol that the encoded page will sit upon to transmit/receive information to and from a client.

Server	The source of a web page to be served to the network.
Client	The remote requestor of the web page offered by the server.
TCP	Transmission Control Protocol. TCP is a protocol used to create connections between applications running on endpoints with each other. TCP distinguishes data from multiple connections to multiple applications running on the same endpoint; for example distinguishing data exchanged between a web server and client from that of an email server and client. TCP provides reliable, in-order delivery of data between endpoints.
UDP	User Datagram Protocol. Using UDP, programs on networked computers can send short messages, sometimes known as datagrams, to one another. UDP is faster, more efficient, and is a smaller protocol than TCP. The DNS system uses UDP to obtain the IP address of a prescribed Uniform Resource Locator (URL), which is the web site’s human-readable address. While UDP is more efficient than TCP, UDP does not acknowledge receipt of data being transmitted as with TCP. In other words, when packets are not received (by the client/from the server) as requested, the TCP protocol requests re-transmission of the missing packets. However, with UDP, the client never asks for re-transmission of missing packets since the protocol is not intelligent enough to know that packets are missing in the first place.
IP	Internet Protocol. IP is used to manage the communication between endpoints across a packet-switched network, such as Ethernet. IP provides no error checking beyond the packet header checksum. The primary function of IP is providing addressing and routing. If provided more data than can be transmitted in a single datagram, IP fragments the data into multiple datagrams for transmission.

TCP/IP	Transmission Control Protocol/Internet Protocol. TCP/IP is a term often used to describe a collection of internet core protocols that are used to communicate between endpoints (computers) on a packet switched network. Examples of protocols in the TCP/IP suite (or stack) include TCP, IP, UDP, ARP, RARP, ICMP, and others.
HTML	Hypertext Markup Language. HTML is the simplest encoding of a web page to be served
HTTP	Hypertext Transfer Protocol. HTTP, from a software prospective, sits on top of the TCP/IP stack and uses the capabilities of the stack to transmit and receive information from remote clients.
DNS	Domain Name System. To start the entire process of requesting a web page, DNS is used to translate hostnames to IP addresses. For example, if you want to know the internet address for www.luminarymicro.com, DNS can be used to tell you it is 74.52.63.50.
URL	Uniform Resource Locator. The URL is the address designated on a given network for a specific web page that can be reached by clients within the network.

Implementation

A web server consists of three components: hardware, software, and content. The hardware component of a web server can be implemented with a solution as simple as a Stellaris Ethernet-enabled microcontroller and a RJ45 connector with integrated magnetics. The microcontroller provides a glueless interface to the most popular implementation–twisted pair wiring. The processor integrated into the microcontroller provides the necessary processing performance to provide web services while also processing the application and interfaced devices.

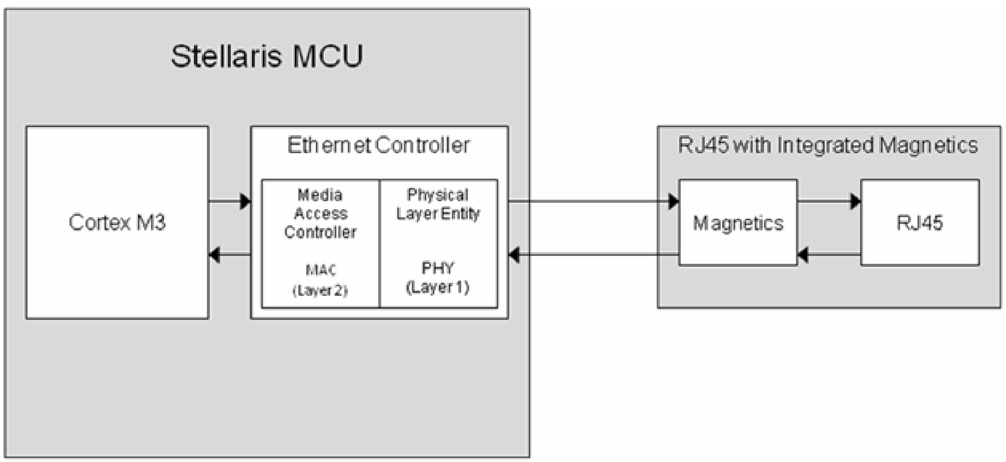
The necessary software to implement a web server is available from many sources. Two popular solutions are the freely available open source lwIP and uIP TCP/IP stacks. Fitting into small memory footprints, these stacks enable network connectivity for the embedded world. Available embedded web server applications sit on top of these stacks and are used to send and receive the appropriate data.

The simplest web server stores the web page content in local memory as static pages. In this case, any request from a client will always result in the client receiving the page as it is stored at the server. To make the web server less static in terms of what the server provides the client, an enhancement called common gateway interface (CGI) is often used. CGI adds the capability of dynamic web page content, which allows alteration of information provided by the web server, inserting new timely information before being sent to the remote client. Dynamic web page creation can be implemented by the addition of a function call to a compiled program or script running on the server. This called program or script generates an image of any current data and passes that information into the stream of HTML data to be assembled as a web page by the client.

Design Considerations for a Hardwired Network Interface

In order to connect to the internet, a web server design requires four main pieces of hardware: the media access controller (MAC), the network physical interface device (PHY), isolating magnetics, and the physical jack that connects to the network system (RJ45) over copper or fiber media. The Stellaris microcontroller solution integrates both MAC and PHY as shown in Figure 1 which also displays the high-level hardware diagram of a hardwired Ethernet interface. The MAC layer provides transmit and receive processing for Ethernet frames. The MAC layer also provides the interface to the PHY module via an internal Media Independent Interface (MII). The PHY is the interface to the network, including the ability to scramble/descramble incoming and outgoing Ethernet packets. For an effective Ethernet-connected embedded system, the PHY should ideally interface to Category-5 unshielded twisted pair (Cat-5 UTP) cabling for 100BASE-TX applications and Category-3 unshielded twisted pair (Cat-3 UTP) for 10BASE-T applications.

Figure 1. Ethernet Implementation with an ARM Cortex-M3 Core



Consideration 1: Find a microcontroller that integrates both MAC and PHY on-chip.

Although there are numerous microcontroller options that offer an on-chip Ethernet MAC, there are few that offer the combination of both MAC and PHY into a single device. However, the integration of both MAC and PHY works to the advantage of the embedded platform designer, since the on-chip combination has the potential to save the designer significant BOM cost through consolidation of functionality and reduction in footprint, not to mention manufacturing savings in reduced stocking charges and reduced waste from multi-chip losses. In addition, microcontrollers that integrate both MAC and PHY ensure a reliable Ethernet system. For instance, designers may encounter difficulty with an implementation of a stand-alone PHY. A lack of adequate design guidance by the supplier of the PHY can result in a non-functional or impaired design. Tested and qualified components that are known to interoperate with the PHY are critical. Therefore, using a microcontroller with integrated MAC and PHY already accounts for these concerns.

Consideration 2: Find a microcontroller with ample on-chip memory.

There are many documented instances of web server technology downsized for use on platforms with tiny memory complements. Single-chip microcontrollers typically provide only minimal non-volatile storage for use by communication programming. Designs using microcontrollers tend to implement serial interface memories, making available additional bulk memory needed for storage of the static data making up a web page (HTML). Besides increasing system cost with the serial memory, the delays caused by serial transfer of data from the serial memory make delivery of any web page lethargic. The overhead of serial memory transfers introduces processing delays into the platform's main application program, thus slowing overall system performance. Solutions using serial memories tend to be static web page displays or pages that present extremely limited dynamic information. Therefore, a microcontroller should be identified that features enough fast on-chip flash memory to store and serve the web content to be served.

Consideration 3: Find a microcontroller with sufficient performance.

Many embedded Ethernet-enabled systems need to not only handle network communication, but also be able to run a complex primary application at the same time. The microcontroller needs to have the necessary overall performance to handle interleaving the network communication and application. Choosing a 32-bit microcontroller is critical to achieving the performance needed to handle these applications. Other performance considerations include fast memory and interrupt processing. On-chip single-cycle flash and SRAM increase performance by minimizing the processor overhead due to memory accesses. Microcontrollers with a low latency interrupt controller reduce the overhead incurred each time an interrupt is received which is critical for a system that will be multitasking.

Consideration 4: Find a microcontroller vendor that offers a complete web server solution.

To quickly get a final product to market, embedded engineers need to work with a microcontroller vendor that offers a complete example web server solution, including all the hardware and software necessary to begin development. A complete example web server solution should include cost-effective hardware, a web server demonstration that runs out-of-the-box, necessary software tools, documentation, a quickstart guide, a working driver library, and example source code. In addition, the vendor should offer application notes and excellent applications support. This way, embedded engineers can begin designs with the confidence in knowing that they are starting from a robust web server example.

Highly Integrated Microcontroller Supporting an Embedded Web Server

To bring network connectivity and web server capability to an embedded platform, ideally the design should include a high-performance processor with on-chip MAC and PHY, a core capable of efficient data movement, and a large memory complement. Additionally, the system should incorporate a software support suite which implements TCP/IP, UDP, an HTTP server, and a CGI or equivalent interface.

Processor performance is critical to an application's ability to offer execution of the application program while at the same time delivering dynamic data over a network in a web page format. The 8-bit and 16-bit microcontrollers available in the marketplace can generate mostly static web pages, but at a high cost in memory use and application performance. Attempting dynamic pages with all but the simplest content may be unobtainable. Therefore, it is highly recommended to incorporate a 32-bit microcontroller, such as a microcontroller featuring an ARM Cortex-M3 processor core.

As mentioned before, designs that attempt to use individual MAC and PHY solutions compromise cost, PCB space/footprint, and manufacturing efficiency. The component sum results in a larger bill of materials to manage and procure, as well as jeopardizes the functionality of the design. A microcontroller that integrates both the MAC and PHY components on-chip with the 32-bit processor core and memory eliminates availability and design issues such as EMI/RFI, signal quality, and timing mismatches.

Memory is a key ingredient in successful embedded network applications. Storage of HTML pages with richer content level is only achievable with sufficient space in which to store the content. The ideal solution is to have large non-volatile FLASH memory attached to the processor over a fast parallel bus. Optimization of this memory system can be best achieved by having the memory array on the semiconductor die with the processor itself. Ideally, this memory should be single-cycle, meaning that the speed that memory can be read is exactly the same speed as the host processor without a single wait state. Ample single-cycle memory ensures optimal performance of the Ethernet system.

Oftentimes, an embedded designer desires more functionality from a microcontroller than simply to connect to the network. For instance, the microcontroller might be expected to interpret printed barcodes on boxes moving down a conveyor belt, then serve a web page updated with the bar code information to the network. In these cases, the microcontroller must feature sufficient processor performance, memory space, peripherals, and I/O connections to achieve its intended uses.

A complete web-enabled platform design can require a daunting bit of software engineering. An organic attempt to develop network protocol stacks, real-time operating systems, and hardware interfacing driver code can extend a design program well beyond even the most conservative project estimates. Using open source software can shorten the software critical path from months to just days. Small, easily ported network stacks such as uIP and lwIP provide almost instant network access. Embedded web servers are available with most TCP/IP stacks, including both uIP and lwIP.

HTML files can be generated easily with today's most common PC software packages. Microsoft's office suite (Word, Excel, Power Point, Publisher, Internet Explorer) supports saving files in HTML directly. Several programs, such as Microsoft Front Page and Adobe Dreamweaver, are designed specifically for web page creation. Even certain photo editors allow users to save files in HTML. For simple implementations and those familiar with HTML code, a designer can even create a simple *.txt file to write out the encoded page. The PC space is replete with programs containing HTML capability.

Including the HTML information in an embedded platform can be done by manually or programmatically inserting each line of HTML into an appropriate 'C' language (or other high-level language) data structure. After compilation and linking, the HTML is programmed in the on-chip FLASH memory as part of the final code module. One could also create a simple file system in FLASH memory containing the HTML text as a file and sequentially read from it.

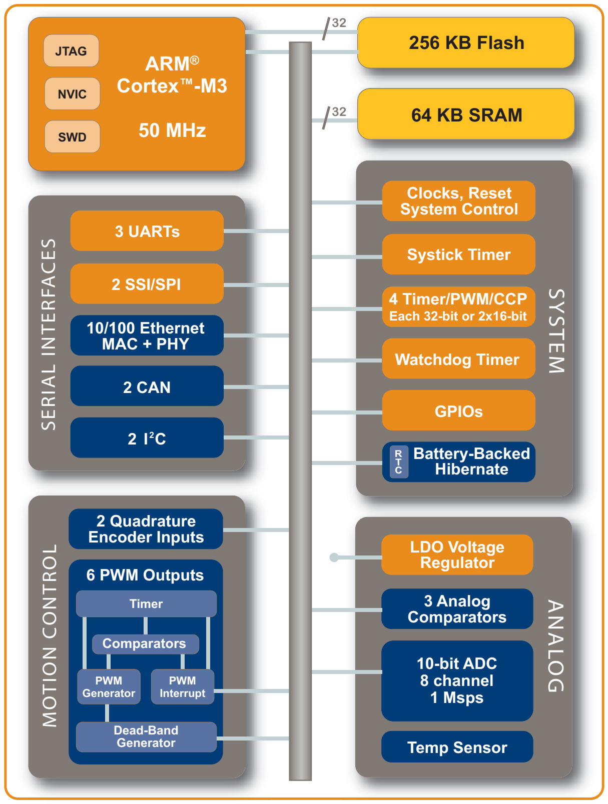
Thanks to the HTML “link” capability, multiple servers can cooperate in the delivery of web content. Companies wanting to include rich content to the client can deliver large amounts of data to the client from corporate web servers upon request from the embedded platform. Items such as backgrounds and pictures can be stored and supplied by the traditional web servers, therefore creating a template that does not have to be stored on the embedded server. The embedded server can then simply plug data into the template already on the network, resulting in a much more efficient system. Larger servers are also able to coalesce data from multiple external embedded servers into a single dynamic page for clients requesting the data from anywhere on the network.

Stellaris – The Cost-Effective Remote Management Solution

The Luminary Micro Stellaris family of microcontrollers offers the most ideal solution to fulfill the requirements of a cost-effective embedded web server. The Stellaris family is based on the ARM Cortex-M3 processor which is the microcontroller member of the ARM Cortex processor family. Designed for serious microcontroller applications, the Stellaris family provides entry into the industry's strongest ecosystem, with code compatibility ranging from $1 to 1 GHz.

The Stellaris microcontroller family offers the advantages of ARM's widely available development tools, System-on-Chip (SoC) infrastructure IP applications, and a large user community. Additionally, the microcontroller uses ARM's Thumb-2 instruction set to reduce memory requirements and, therefore, reduce cost. Finally, the extensive Stellaris family is code-compatible with all other family members, providing flexibility to fit precise needs for embedded applications. See Figure 2 for a block diagram of the rich feature set of the Stellaris Family of microcontrollers.

Figure 2. Stellaris Family Block Diagram



The Stellaris family offers efficient performance and extensive integration, favorably positioning the device into cost-conscious applications requiring significant control processing and connectivity capabilities such as test and measurement equipment, motion control, medical instrumentation, HVAC and building control, transportation, remote monitoring, electronic point-of-sale machines, network appliances and switches, and gaming equipment. For specific features of Stellaris Ethernet-enabled devices, see the Stellaris Family Product Selector Guide which can be found on Luminary Micro's web site at wwww.luminarymicro.com.

Superior performance through single-cycle flash coupled with inherently better code efficiency on the ARM Cortex-M3 core

The Stellaris Family of ARM Cortex-M3 microcontrollers feature up to 256 Kbytes of single-cycle industrial-grade flash memory, coupled with up to 64 Kbytes of on-chip single-cycle SRAM. Single-cycle memory means that the R/W of memory speed is the same as the Stellaris MCU core frequency–up to 50 MHz. In addition, the inherently better code efficiency of the Cortex-M3 core translates to cost savings to Stellaris customers, as equivalent code stored in Stellaris flash can typically fit in half of the flash needed for a traditional ARM7 MCU.

Stellaris—the first microcontroller to bring serious industrial connectivity capability to the ARM architecture

Several members in the Stellaris Family of ARM Cortex-M3 microcontrollers feature either a fully integrated 10/100 Ethernet MAC and PHY or integrated Bosch CAN networking technology, the golden standard in short-haul industrial networks. In addition, Stellaris MCUs feature up to 3 UARTs, 2 I^2Cs, and up to 2 SSI/SPI serial interfaces. Together with larger on-chip memories, enhanced power management, and expanded I/O and control capabilities, these Stellaris family members are optimized for industrial applications requiring reliable connectivity, including remote monitoring, electronic point-of-sale machines, test and measurement equipment, network appliances and switches, factory automation, HVAC and building control, gaming equipment, motion control, medical instrumentation, and fire and security.

Sophisticated Stellaris motion control technology—accelerating motor design

The Stellaris Family of ARM Cortex-M3 microcontrollers features functionality especially designed for meticulous motion control and typically found only on costly specialty devices, including up to six full channels of pulse width modulation waveform generators with dead-band timers for applications such as 3-phase inverter bridges. Stellaris motion functionality also includes shoot-through protection, fault-condition handling in hardware to quickly provide low-latency shutdown, synchronization of timers to enable precise alignment of all PWM edges, and hardware quadrature encoders to enable precise positioning sensing.

Fast, Flexible, and Smart Analog Capability

The Stellaris Family of ARM Cortex-M3 microcontrollers features intelligent analog capability, including up to 8 channels of ADC operating at up to 1 Msps, up to 3 analog comparators, and an on-chip temperature sensor. The ADC features sophisticated event sequencers to minimize CPU utilization, ensuring that the CPU is used for data processing—not data collection. In addition, every Stellaris device offers an on-chip LDO voltage regulator to provide the correct voltages to power the device from a 3.3 V source.

Complete Software Support

Luminary Micro offers a complete solution to get to market quickly, with a customer development board, white papers and application notes, an easy-to-use peripheral driver library, and a strong support, sales, and distributor network.

Luminary Micro provides software support for engineers wanting to implement an embedded web server in their design. Both uIP and lwIP are open-source embedded TCP/IP stacks with supporting embedded web servers. Both the uIP and lwIP stacks have been ported and are offered as part of Stellaris evaluation kits featuring Ethernet connectivity (EKx-LM3S6965).

The primary differences between uIP and lwIP are the memory requirements and capabilities. The uIP package tends to be smaller and implements the IP, ICMP, UDP, and TCP protocols. The lwIP package is larger and implements the same IP, ICMP, UDP, and TCP protocols as well as provides an operating system emulation layer, buffer and memory management subsystems, and network interface functions. Both packages provide support for dynamic web page delivery. The designer can choose which implementation is best for his design based on size and needed networking functionality. Both of these capable packages fit neatly into the Stellaris memory space.

Luminary provides software examples for both the uIP and lwIP TCP/IP stacks with their evaluation kits. The uIP example uses a static IP address to serve up a basic web page. For this example, the uIP stack is configured so that it requires 4.5 Kbytes of flash memory for code and 2.5 Kbytes of SRAM for data. The lwIP example uses DHCP to acquire an IP address and serves up sample web pages. The lwIP stack in this example consumes 25 Kbytes of flash memory for code and 15 Kbytes of SRAM for data. A technical explanation of these examples can be found in Luminary Micro's application notes. In addition to these two examples, the Stellaris Ethernet evaluation kits ship with an interactive game and a fully functional web server programmed into the flash. The source code is provided as well.

A comprehensive library of drivers (the Stellaris Peripheral Driver Library, or DriverLib) for the Stellaris' on-chip peripherals is available and supplied in each evaluation kit. The already mentioned examples, as well as others, are included in this library. The library's 200+ page user's guide provides ample explanation of the application program interface (API) functions and detailed programming examples enabling easy access to the powerful on-chip peripherals without requiring extensive study of the data sheet. DriverLib is coded in the 'C' programming language, allowing software designers to avoid assembly language during the entire coding and debugging process. Quick software development is enabled by the Stellaris peripheral driver library's pre-tested support routines.

An array of support tools is offered for Stellaris designers. Compilers from Keil, IAR, and Code Sourcery are fully compatible with the Stellaris Peripheral Driver Library and have additional example programs. In addition to uIP and lwIP, commercial TCP/IP stacks are available for Stellaris microcontrollers from Micrium, expresslogic, CMX Systems, and Interniche. RTOS support is offered by Keil, IAR, FreeRTOS.org, expresslogic, Micrium, SEGGER, and Pumpkin, which can add multi-tasking support to a networked design.

Summary

Luminary Micro's Stellaris family of Ethernet-enabled microcontrollers are ideal for designs seeking to support high performance and cost-effective embedded web server capability. Adding the benefits of complete evaluation platforms, example applications, the Stellaris Peripheral Driver Library, available TCP/IP stacks, RTOS support, broad third-party tool support, and comprehensive documentation enables fast time-to-market.

Luminary Micro offers a complete solution to get to market quickly, with evaluation and development platforms, white papers and application notes, an easy-to-use peripheral driver library, and a strong support, sales, and distributor network. For more information on Stellaris Ethernet-enabled microcontrollers, go to http://www.luminarymicro.com/embeddedwebserver. To purchase an evaluation kit for Stellaris Ethernet microcontrollers including the embedded web server application described in this white paper, go to http://www.luminarymicro.com/products/lm3s6965_ethernet_evaluation_kit.html to purchase online or to http://www.luminarymicro.com/sales to find your local sales partner.

LUMINARY MICRO

References

The following documents are available for download at www.luminarymicro.com:

- Stellaris microcontroller data sheets for Ethernet, Publication Number DS-LM3S*nnnn* (where *nnnn* is the part number for that specific Stellaris family device)
- *Using the Stellaris® Ethernet Controller with Micro IP (uIP) Application Note*, Publication Number AN01260
- *Using the Stellaris® Ethernet Controller with Lightweight IP (lwIP) Application Note*, Publication Number AN01261
- *Stellaris® Peripheral Driver Library User's Guide*, Publication Number PDL-LM3Snnnn
- Stellaris Peripheral Driver Library
- *Stellaris® Family Product Selector Guide*, Publication Number PSG-LM3SFAM-05

Company Information

Luminary Micro, Inc. designs, markets, and sells ARM Cortex-M3-based microcontrollers (MCUs). Austin, Texas-based Luminary Micro is the lead partner for the Cortex-M3 processor, delivering the world's first silicon implementation of the Cortex-M3 processor. Luminary Micro's introduction of the Stellaris® family of products provides 32-bit performance for the same price as current 8- and 16-bit microcontroller designs. With entry-level pricing at $1.00 for an ARM technology-based MCU, Luminary Micro's Stellaris product line allows for standardization that eliminates future architectural upgrades or software tool changes.

Luminary Micro, Inc.
108 Wild Basin, Suite 350
Austin, TX 78746
Main: +1-512-279-8800
Fax: +1-512-279-8879
http://www.luminarymicro.com

LUMINARY MICRO™ ARM POWERED® Cortex Intelligent Processors by ARM®