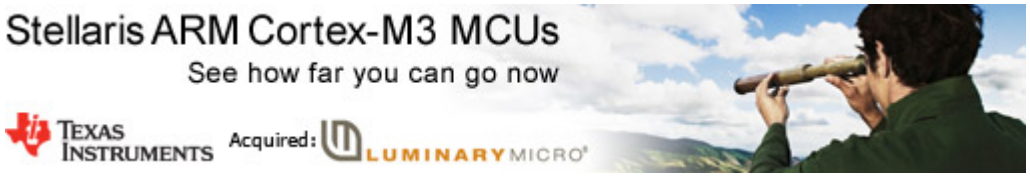


Stellaris Cortex M3 32-bit MCU Family

Stellaris MCUs and ARM Cortex-M3 offers a direct path to the strongest ecosystem of development tools, software and knowledge in the industry

TechDay 2010

Agenda

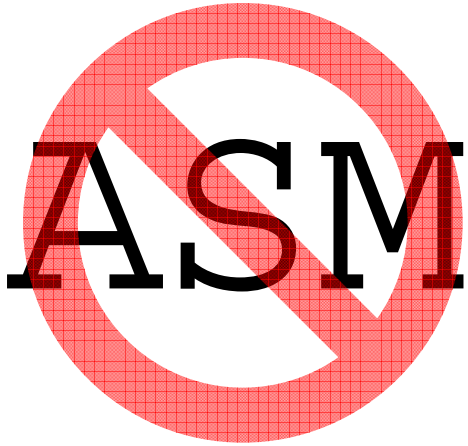
- **TI Embedded Processor Overview**
- **ARM7 vs. Cortex M3**
- **Stellaris Roadmap and Device Overview**
- **Architectural Details**
- **Key Peripherals**
  - ✓ Ethernet, USB, Motion Control, EPI
- **Getting Started**
  - ✓ StellarisWare Software
  - ✓ Hardware Tools
- **Summary**

TI Embedded Processing Portfolio

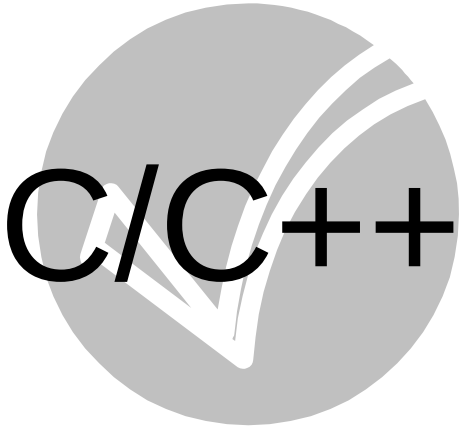
Microcontrollers			ARM-Based		DSP
16-bit	**32-bit Real-time**	**32-bit ARM**	**ARM+**	**ARM + DSP**	**DSP**
MSP430	**C2000™**	**ARM**	**ARM9 Cortex A-8**	**C64x+ plus ARM9/Cortex A-8**	**C647x, C64x+, C55x**
Ultra-Low Power	Fixed & Floating Point	Industry Std Low Power	Industry-Std Core, High-Perf GPP	Industry-Std Core + DSP for Signal Proc.	Leadership DSP Performance
Up to 25 MHz	Up to 300 MHz	<100 MHz	Accelerators	4800 MMACs/ 1.07 DMIPS/MHz	24,000 MMACS
Flash 1 KB to 256 KB	Flash 32 KB to 512 KB	Flash 64 KB to 1 MB	MMU	MMU, Cache	Up to 3 MB L2 Cache
Analog I/O, ADC LCD, USB, RF	PWM, ADC, CAN, SPI, I²C	USB, ENET, ADC, PWM, SPI	USB, LCD, MMC, EMAC	VPSS, USB, EMAC, MMC	1G EMAC, SRIO, DDR2, PCI-66
Measurement, Sensing, General Purpose	Motor Control, Digital Power, Lighting, Sensing	Host Control	Linux/WinCE User Apps	Linux/Win + Video, Imaging, Multimedia	Comm, WiMAX, Industrial/ Medical Imaging
$0.49 to $9.00	$1.50 to $20.00	$2.00 to $8.00	$8.00 to $35.00	$12.00 to $65.00	$4.00 to $99.00+

Software & Dev. Tools

No assembly required!

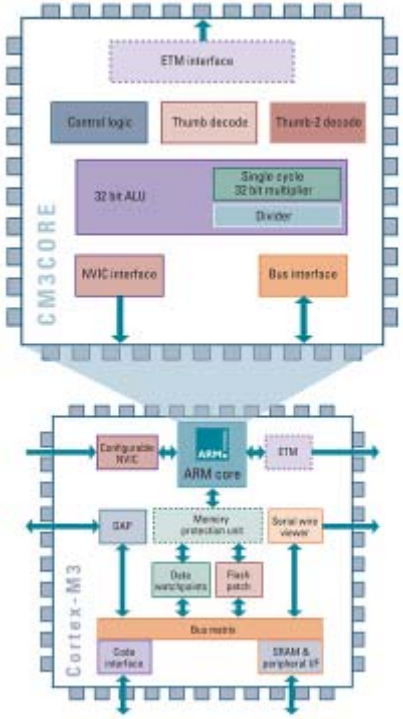


- **Cortex-M3 has complete hardware support for interrupts**
  - ✓ Interrupt Service Routines (ISRs) are purely written in C/C++
  - ✓ Interrupt setup is easily done in C/C++
    - C/C++ array which contains the vectors (pointers to the C/C++ functions)
    - Pointer to the stack (a C/C++ array)
- **No boot code ASM, no system configuration ASM**
  - ✓ ARM7 compilers normally comes with a ASM boot routine (in object form) that does setup.
  - ✓ For Cortex-M3, no boot routine is needed
    - Cortex-M3 hardware loads the stack pointer from memory and the initial PC from memory and enters as a normal C function.
  - ✓ User C/C++ code is all that is required.
- **Entire software code base can be written in C/C++**
  - ✓ ISRs
  - ✓ RTOS
  - ✓ Application code



Cortex™-M3 benefits

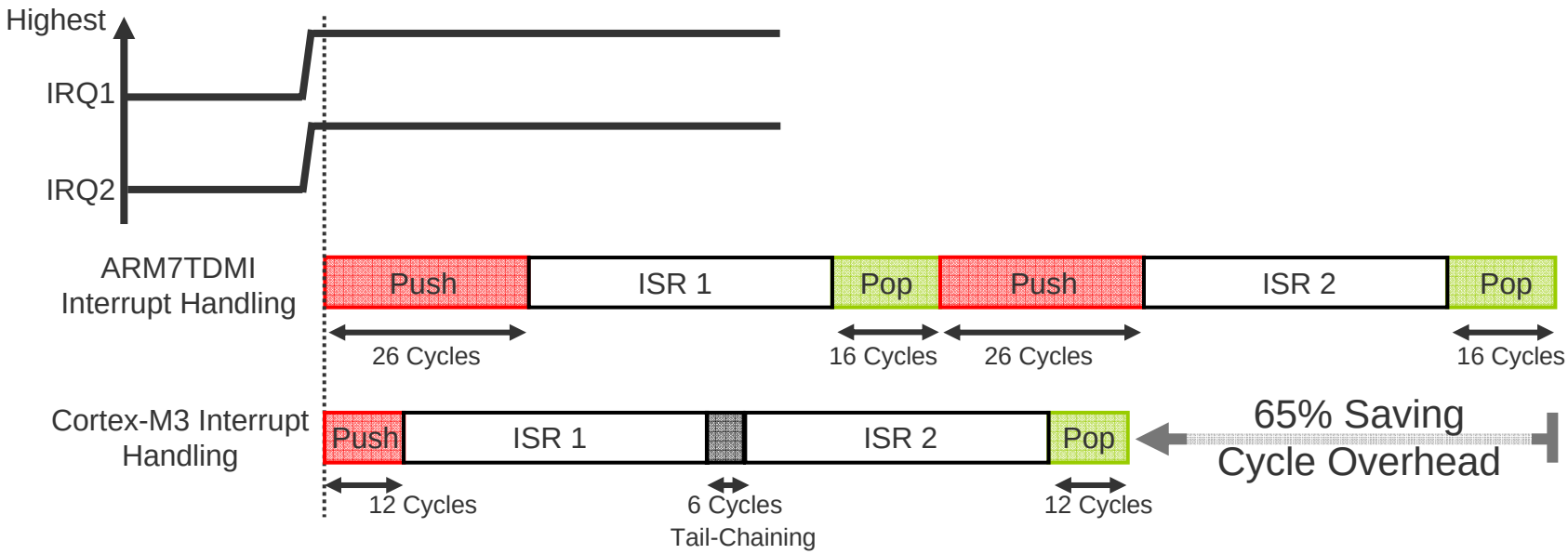
- **Capabilities beyond ARM7 for the MCU market:**
  - ✓ No Assembly Required!
  - ✓ Cortex-M3 requires approximately ½ the flash of ARM7 implementations
  - ✓ 2-4 times faster on MCU control applications
    - Raw interrupt performance: we're 85% faster
    - PID (process control) main loop: we're 217% faster
    - Multiply-intensive code: we're 294% faster
    - Divide-intensive code: we're 726% faster



Features	ARM7TDMI	ARM Cortex-M3
Architecture	ARMv4T (von Neumann	ARMv7-M (Harvard)
ISA Support	Thumb / ARM	Thumb / Thumb-2
Pipeline	3-stage	3-stage + branch speculation
Interrupts	FIQ / IRQ	NMI +1 to 240 physical interrupts
Interrupt Latency	24 - 42 cycles	12 cycles
Inter-Interrupt Latency	24 cycles	6 cycles
Sleep Modes	None	Integrated
Memory Protection	None	8 region MPU
Dhrystone	0.95 DMIPS/MHz (ARM) 0.74 DMIPS/MHz (Thumb)	1.25 DMIPS/MHz

Source: http://www.arm.com/products/CPUs/ARM_Cortex-M3.html

Interrupt response – tail chaining



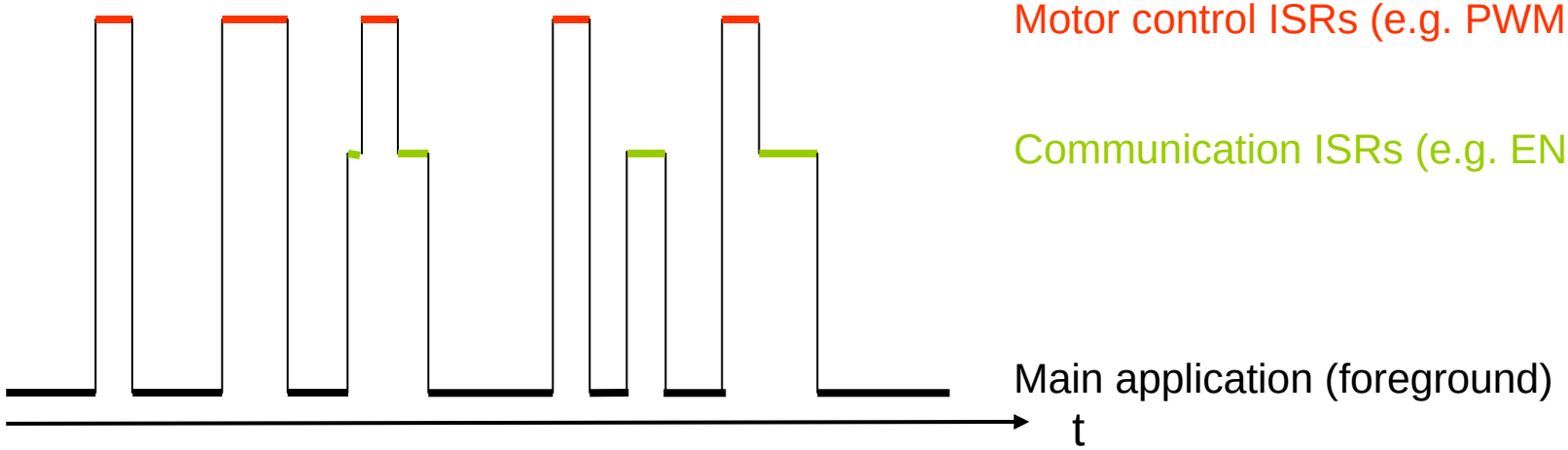
ARM7TDMI

- 26 cycles from IRQ1 to ISR1 (up to 42 cycles if in LSM)
- 42 cycles from ISR1 exit to ISR2 entry
- 16 cycles to return from ISR2

Cortex-M3

- 12 cycles from IRQ1 to ISR1 (Interruptible/Continual LSM)
- 6 cycles from ISR1 exit to ISR2 entry
- 12 cycles to return from ISR2

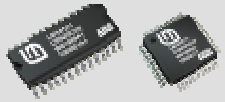
How it works on Cortex-M3



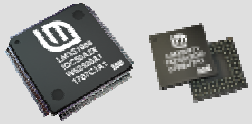
- **Main application runs as foreground (base level)**
  - ✓ Easy to write since no “factoring” – just normal application or RTOS based
  - ✓ Can use PLC style state-machine poll loop safely: ISRs keep data available
- **ISRs for Motor control are highest priority(ies)**
  - ✓ PWM, ADCs, Timer(s), Fault (may be highest), Temp sensor, etc
- **ISRs for communications below that**
  - ✓ Ethernet, CAN, and/or serial
- **May use other priorities as needed**
  - ✓ Very fast interrupt response time, true nested interrupts, priority masking, easy ISR setup all contribute to making an easy solution
  - ✓ Application uses priority masking vs. interrupt-disable if needs critical region

Roadmap and Device Overview

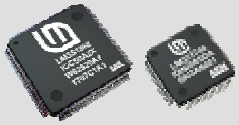
Stellaris Roadmap



Sandstorm
- 64K flash
- 8K SRAM
- Motion control
- 1MSPS ADC



Fury class
- 256K flash
- 64K SRAM
- Motion control
- ETH MAC+PHY
- CAN 2.0
- Ethernet+CAN



DustDevil class
- 128K flash
- 64K SRAM
- Motion control enhancements
- USB 2.0 O/H/D
- 32ch DMA

Tempest class
- Higher performance
- Low power
- ETH+CAN+USB OTG
- External bus capability
- I2S
- Precision OSC

512K Firestorm
- Advanced analog
- 512KB Flash
- Larger pin count
- Expanded serial connectivity

Blizzard class
- Small form factor
- Low pin count
- Expanded serial connectivity

Whiteout class
- Small form factor
- Low pin count

2006 2007 2008 2009 2010 2011

Stellaris product lines

- LM3S1nnn non-CAN, non-Ethernet

- LM3S2nnn CAN
- LM3S3nnn USB
- LM3S5nnn CAN + USB

- LM3S6nnn Ethernet
- LM3S8nnn Ethernet + CAN
- LM3S9nnn Ethernet + CAN + USB

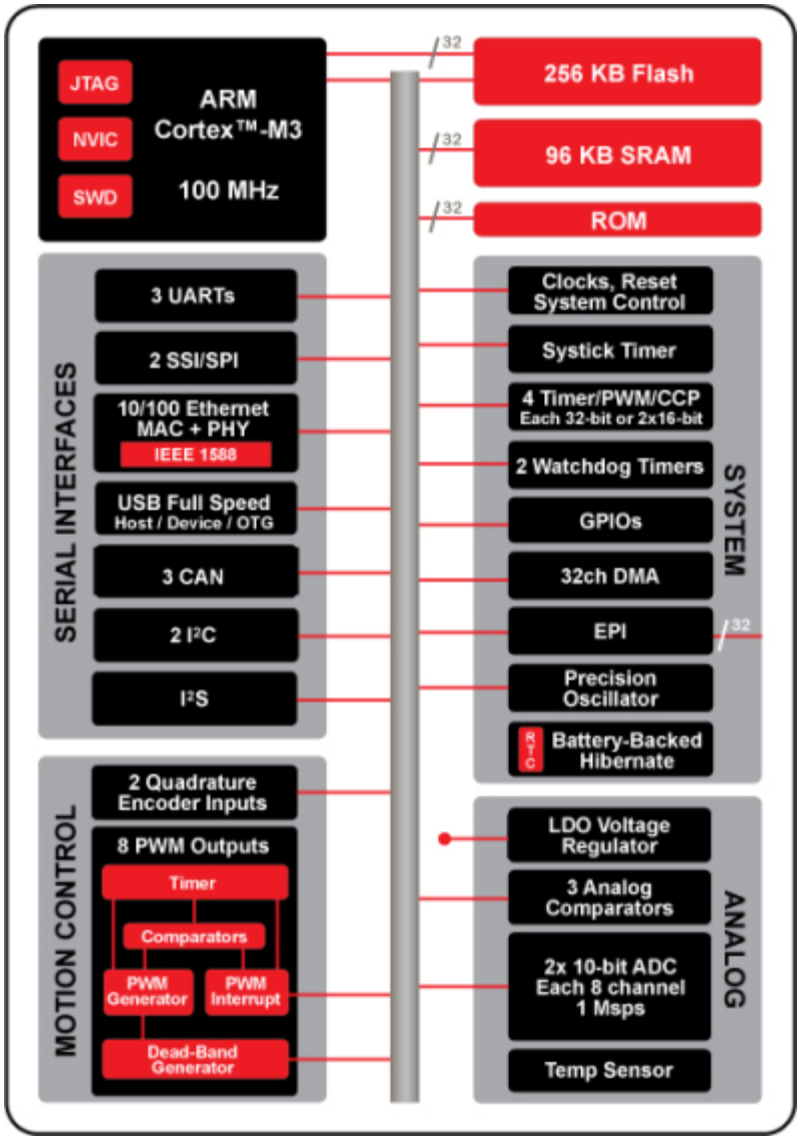
Stellaris product lines

Part number	Flash size	SRAM size
LM3Sx1xx	64K Flash	16K SRAM
LM3Sx2xx		32K SRAM
LM3Sx3xx	96K Flash	16K SRAM
LM3Sx4xx		32K SRAM
LM3Sx5xx		64K SRAM
LM3Sx6xx	128K Flash	32K SRAM
LM3Sx7xx		64K SRAM
LM3Sx8xx	256K Flash	32K SRAM
LM3Sx9xx		64K SRAM
LM3SxBxx		96K SRAM
LM3SxDxx	512K Flash	96K SRAM

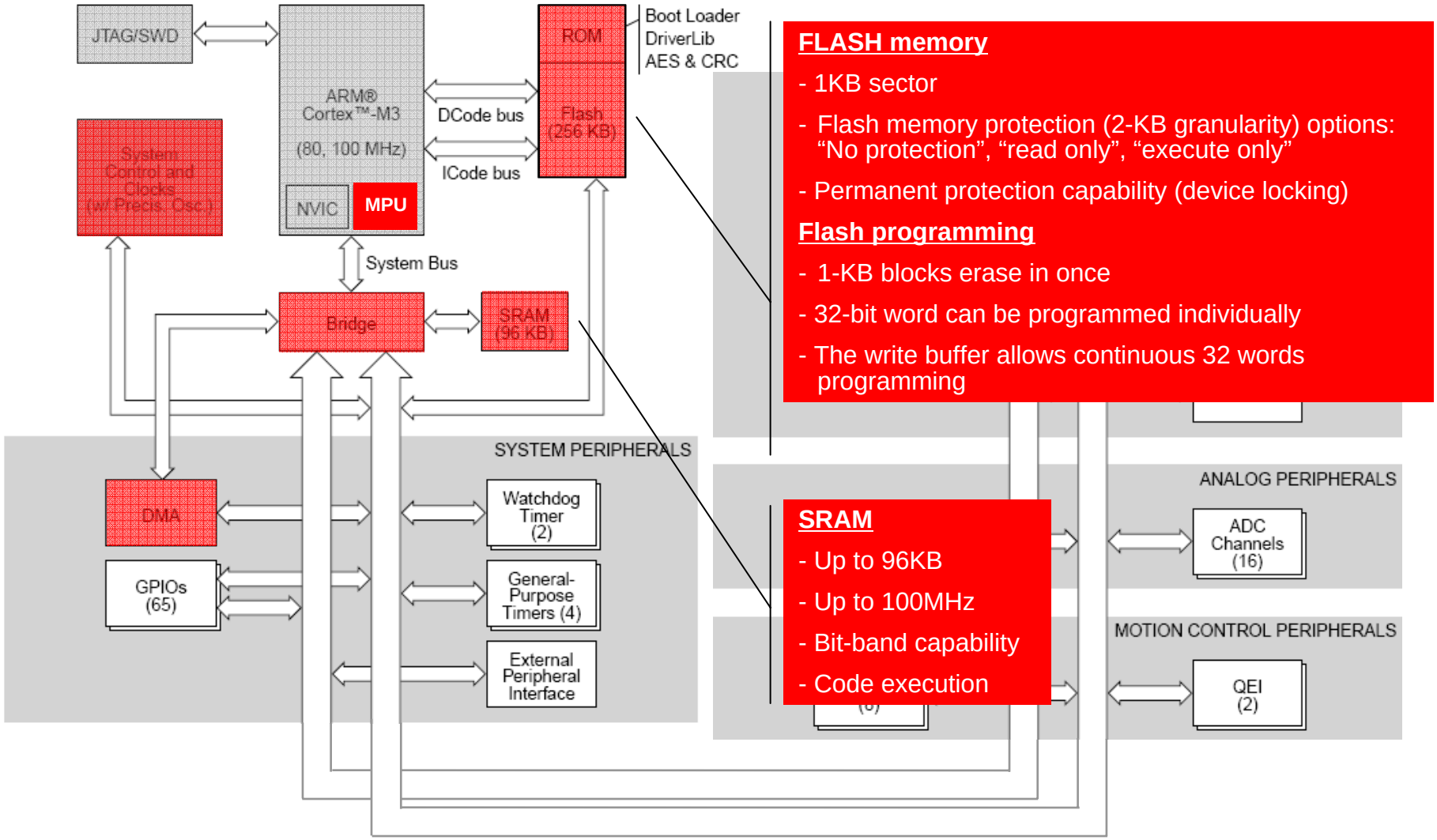
Stellaris Architectural Details & Key Peripherals

Stellaris® modules

- **ARM® Cortex™-M3 Processor Core**
  - ✓ Up to 100 MHz
- **On-chip Memory**
  - ✓ 256 KB Flash, 96 KB SRAM
  - ✓ **ROM** loaded with Stellaris DriverLib, BootLoader, AES tables, and CRC
- **External Peripheral Interface (EPI)**
  - ✓ 32-bit dedicated parallel bus for external peripherals
  - ✓ Supports SDRAM, SRAM/Flash, M2M
- **Advanced Serial Integration**
  - ✓ 10/100 **Ethernet MAC and PHY**
  - ✓ 3 CAN 2.0 A/B Controllers
  - ✓ USB (full speed) OTG / Host / Device
  - ✓ 3 UARTs with IrDA and ISO 7816 support*
  - ✓ 2 I2Cs
  - ✓ 2 Synchronous Serial Interfaces (SSI)
  - ✓ Integrated Interchip Sound (I2S)
- **System Integration**
  - ✓ 32-channel DMA Controller
  - ✓ Internal Precision 16MHz Oscillator
  - ✓ Two watchdog timers with separate clock domains
  - ✓ ARM Cortex Systick Timer
  - ✓ 4 **32-bit timers** (up to 8 16-bit) with RTC capability
  - ✓ Lower-power battery-backed hibernation module
- **Flexible pin-muxing capability**
- **Advanced Motion Control**
  - ✓ 8 advanced PWM outputs for motion and energy applications
  - ✓ **2 Quadrature Encoder** Inputs (QEI)
- **Analog**
  - ✓ 2x 8-ch 10-bit ADC (for a total of 16 channels)
  - ✓ 3 **analog comparators**
  - ✓ **On-chip voltage regulator** (1.2V internal operation)



Stellaris architecture



FLASH memory

- 1KB sector
- Flash memory protection (2-KB granularity) options: "No protection", "read only", "execute only"
- Permanent protection capability (device locking)

Flash programming

- 1-KB blocks erase in once
- 32-bit word can be programmed individually
- The write buffer allows continuous 32 words programming

SRAM

- Up to 96KB
- Up to 100MHz
- Bit-band capability
- Code execution

Low Flash access cycle

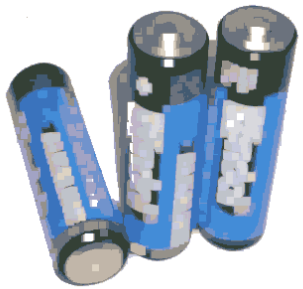
- **Single-Cycle Flash memory up to 50MHz!**
  - ✓ Core speeds does not count if the flash is not single-cycle

Vendor	MCU Line	Flash Access Time 20MHz CPU	Flash Access Time 25MHz CPU	Flash Access Time 50MHz CPU	Unit of Measure
TI	**Stellaris**	**1**	**1**	**1**	Cycle

- **Prefetch buffer enabled when CPU > 50 MHz.**
  - ✓ Prefetches two 32-bit words per clock allowing Flash memory to be read with no wait states while code is executing linearly

Flash access specifications from published datasheets

Stellaris means:



- **Battery-backed Hibernation Module (Standby current as low as 10µA*)**
  - ✓ 32-bit real-time counter (RTC)
    - Programmable 32.768-kHz external oscillator or a 4.194304-MHz crystal
    - RTC software trim for making fine adjustments to the clock rate
  - ✓ 256 bytes (sixty-four 32-bit words) of non-volatile battery-backed memory
  - ✓ Power-switching logic to discrete external regulator (switch to battery)
  - ✓ Low-battery detection, signaling, and interrupt generation
  - ✓ Wake on RTC match and / or external pin
- **On-chip Low Drop-Out (LDO) voltage regulator**
- **Low-power options on controller: Sleep and Deep-sleep modes**
- **Low-power options for peripherals: software controls shutdown of individual peripherals**
- **3.3-V supply brownout detection and reporting via interrupt or reset**

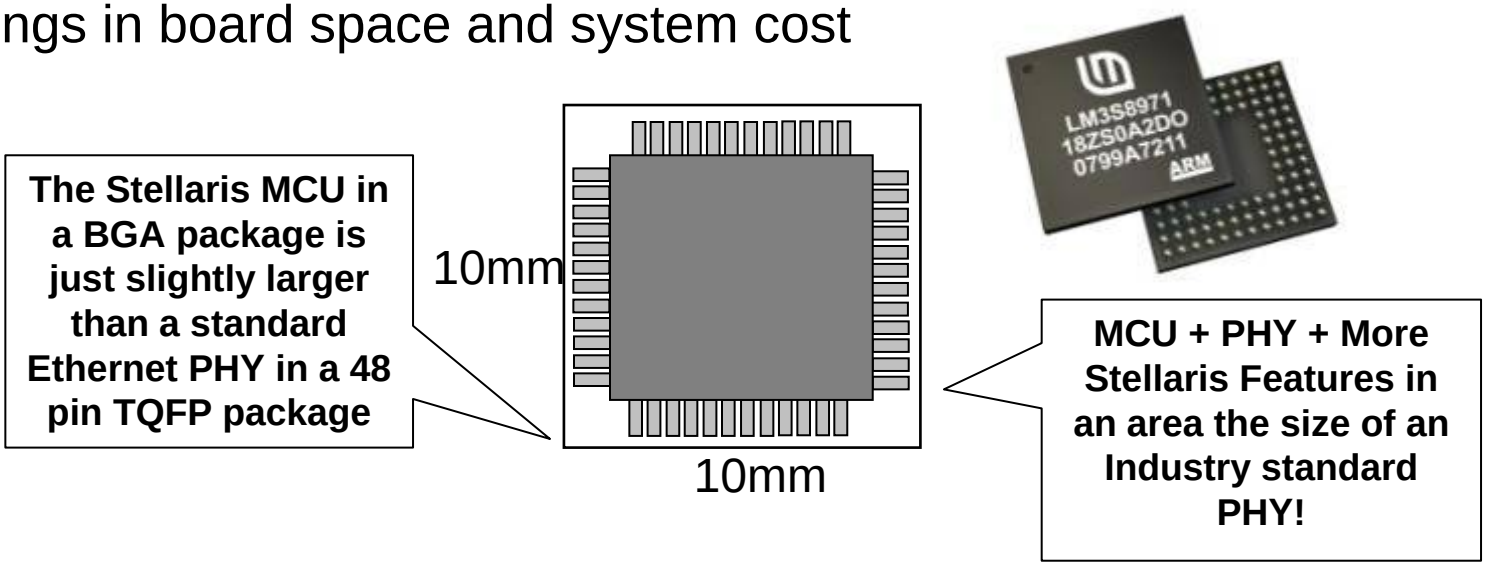
Operating Mode	Sandstorm Class	Fury Class	Dust Devil Class	Tempest Class*
Run	< 120 mA	160 mA (w/ETH)	120 mA	60 mA (w/o ETH) 80 mA (w/ETH)
Sleep	20 mA	20 mA (w/ETH)	20 mA	8 mA
Deep Sleep	700 µA	5 mA (w/ETH)	350uA	600 µA
Hibernate	—	10 to 18 µA	10 to 18 µA	10 to 18 µA

* Preliminary

Ethernet MAC+PHY

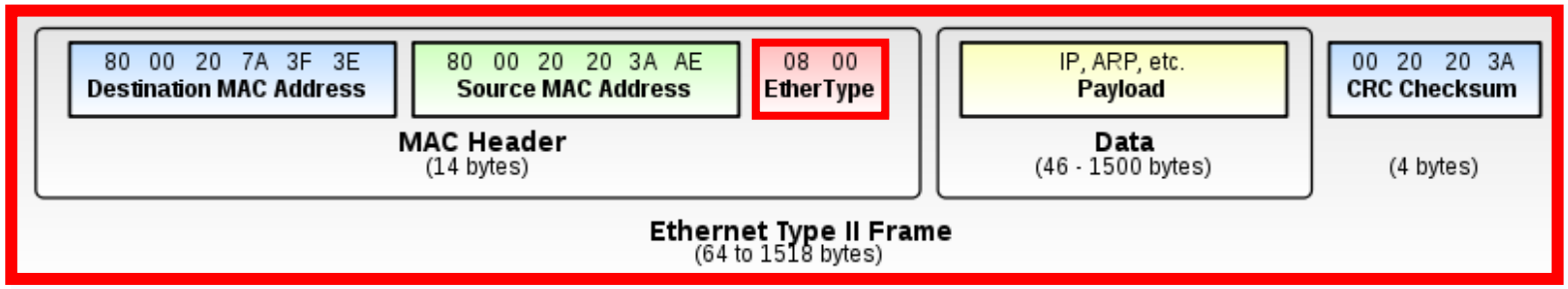
- **The only ARM MCU with 10/100 Ethernet MAC / PHY**

  - ✓ Enables network connectivity and embedded web servers
  - ✓ Lower external power budget requirements than solutions using an external PHY
  - ✓ Savings in board space and system cost



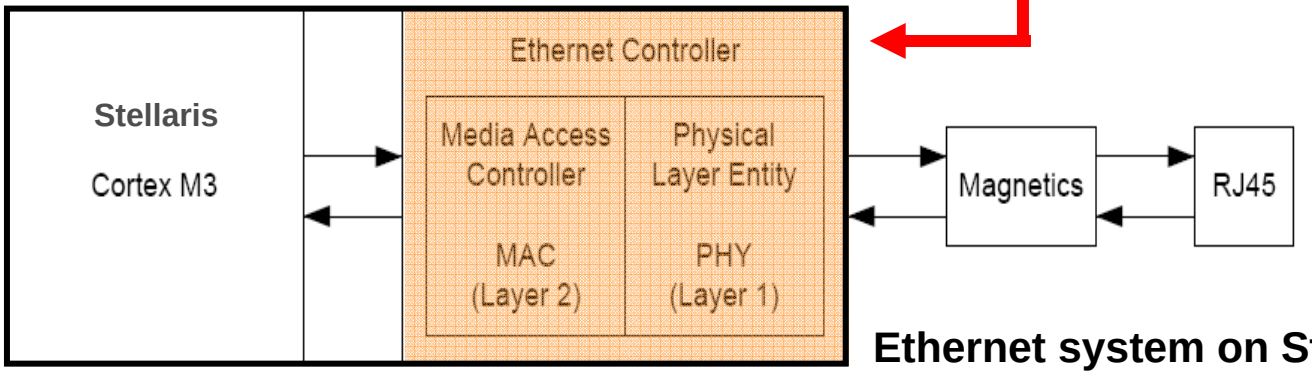
  - ✓ Hardware support for Precision Time Protocol (IEEE 1588 PTP)

Stellaris Ethernet MAC + PHY



Application layer	DHCP, DNS, FTP, HTTP, …
Transport layer	TCP, …
Network/Internet layer	IP (IPv4, IPv6), …
Data link layer	Ethernet, 802.11 (WLAN), …
Physical layer	Ethernet physical layer, …

Ethernet & network layers



Ethernet system on Stellaris

Stellaris Ethernet MAC + PHY

MAC + PHY module

- **Integrated 10/100 Mbps Transceiver (PHY)**
- **IEEE 1588 PTP Hardware-Assisted Support**
- **10BASE-T and 100BASE-TX/RX IEEE 802.3 Full/Half-Duplex support**
- **Automatic MDI/MDI-X cross-over correction**
- **Programmable MAC address**
- **Promiscuous mode support**
- **2KB Transmit FIFO / 2KB Receive FIFO**

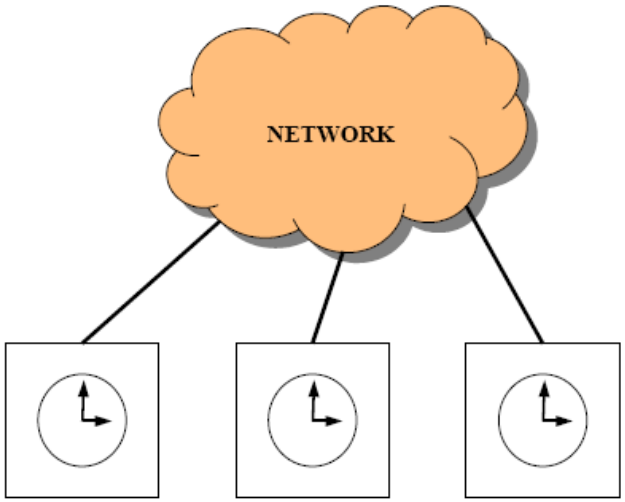
Collaterals

- **Evaluation kits:** EK-LM3S6965, EK-LM3S8962 (+ CAN), EK-LM3S9B92 (+ CAN + USB) with code examples
- Ethernet MAC layer access through drivers lib (Rom Coded for some devices)
- **Communication stack avalibilities**
  - ✓ Ported open-source lwIP stack and µIP stacks
  - ✓ Web Server applications (from Luminary Micro and FreeRTOS.org™)
  - ✓ Third parties

µC/TCP-IP Protocol Stack expresslogic CMX SYSTEMS freeRTOS µIP lwIP IXXAT interniche technologies, inc. SEVENSTAX EMBEDDED TECHNOLOGIES

Ethernet - IEEE1588 PTP

- **What is IEEE1588?**
  - ✓ IEEE 1588 is “Precision Clock Synchronization Protocol for Network and Control Systems” or Precision Time Protocol (PTP)
  - ✓ IEEE 1588 is a protocol designed to synchronize real-time clocks in the nodes of a distributed system that communicate using a network (Ethernet) at a high degree of accuracy



- **Stellaris PTP HW support**
  - ✓ Microsecond accuracy is easily achievable using low cost, small footprint implementations such as Stellaris

PTP in Industrial Applications

- **Industry synchronization requirements for PTPA**

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/IPB
  - ✓ Servo-controlled systems require 100s of nanoseconds
    - Requires significant hardware assist
    - 36% of applications are networked, 56% use Ethernet TCP/IPC

- **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.

Stellaris USB

Integrated controller and PHY

- **USB 2.0 Full Speed (12 Mbps) operation**
- **Devices with OTG/Host/Device or Host/Device**
- **Transfer: Control, Interrupt, Bulk and Isochronous**
- **Up to 32 Endpoints**
  - ✓ 1 dedicated control IN endpoint and 1 dedicated control OUT endpoint
  - ✓ Up to 15 configurable IN endpoints and 15 configurable OUT endpoints
  - ✓ 4 KB Dedicated Endpoint Memory
  - ✓ DMA capability (up to three IN Endpoints and three OUT Endpoints)
  - ✓ One endpoint may be defined for double-buffered 1023-bytes isochronous packet size

Stellaris collaterals

- **Luminary Micro is a member of the USB Implementers Forum.**
- **Stellaris pUSB device and host compliance testing and therefore listed on the USB-IF Integrators List and approved to use the USB logo**
- **Luminary Micro sublicenses VID & PIDs for customer use**
- **StellarisWare USB lib including USB low layer drivers, USB Host & device classes support**
- **Evaluation kits like EK-LM3S3748 with USB class implementations examples**

Controller Area Network (CAN)

Stellaris Integrates Controller Area Network

- Up to 3 Bosch-licensed CAN controllers
- Each supports CAN protocol version 2.0 part A/B
- Bit rates up to 1Mb/s
- 32 message objects, each with own identifier mask
- Maskable interrupt
- Disable automatic retransmission mode for TTCAN
- Programmable loop-back mode for self test operation

Stellaris collaterals

- Over 50 CAN-enabled Stellaris® ARM® Cortex™-M3 microcontrollers
- The EK-LM3S2965 CAN-network-in-a-CAN evaluation kit
- The EK-LM3S8962 CAN-network-plus-Ethernet evaluation kit
- Access to CAN quickstart applications and software examples from renowned CAN stack providers.

Motion control

Meticulous Motion Control…

- Up to 8 general-purpose PWMs **AND**
- Up to 8 channels of motion control PWMs.
- General-purpose PWMs
  - ✓ Stellaris 16-bit timer simple PWM mode with programmable output negation.
- Motion-control PWM Module
  - ✓ Can generate simple PWM signals for a simple charge pump.
  - ✓ Can generate paired PWM signals with dead-band delays for a half-H bridge driver.
  - ✓ Can generate the full six channels of gate controls for a 3-Phase inverter bridge.
  - ✓ Dead-band generator providing shoot-through protection.
  - ✓ Synchronization of timers enables precise alignment of ***all*** edges.
- Up to 4 fault-condition handling inputs in hardware quickly provide low-latency shutdown.
- Up to 2 Quadrature Encoder Inputs provide accurate positioning for closed-feedback control

Stellaris PWM

- **One 16-bit counter**
  - ✓ Runs in down or up/down mode
  - ✓ Output frequency controlled by a 16-bit load value
  - ✓ Load value updates can be synchronized
  - ✓ Produces output signals at zero and load value
- **Two comparators**
  - ✓ Comparator value updates can be synchronized
  - ✓ Produces output signals on match
- **PWM generator**
  - ✓ Output constructed based on actions taken as a result of the counter and comparator output signals
  - ✓ Produces two independent PWM signals
- **Dead band generator**
  - ✓ Produces two PWM signals with programmable dead band delays suitable for driving a half-H bridge
  - ✓ Can be bypassed, leaving input PWM signals unmodified
- **Output control block**
  - ✓ PWM output enable of each PWM signal
  - ✓ Optional output inversion of each PWM signal
  - ✓ Optional fault handling for each PWM signal
  - ✓ Synchronization of timers in the PWM generator blocks
  - ✓ Synchronization of timer/comparator updates across the PWM generator blocks
  - ✓ Interrupt status summary of the PWM generator blocks

Stellaris ADCs

- **Flexible control for up to 16 channels of ADC**
- **Dual sampling available on selected Stellaris MCUs**
- **4 sequencers available of sample length 8, 4, 4, and 1**

0	1	0	2	0	3	0	4

ADC Channel Sequencer

- **Triggers: Software, Timers, GPIO, Analog comparator, PWM**
- **Single-ended and differential-input configurations**
- **On-chip internal temperature sensor**
- **Sample rate up to 1MSPS**
- **Hardware averaging of up to 64 samples for improved accuracy**
- **Converter uses an internal 3-V reference or an external reference**

Differential Pair	Analog Inputs
0	0 and 1
1	2 and 3
2	4 and 5
3	6 and 7
4	8 and 9
5	10 and 11
6	12 and 13
7	14 and 15

Differential inputs settings

Benefits

- ✓ Easily program different sample rates for different analog inputs
- ✓ Multiple channels can be selected for automatic read on a trigger
  - No SW / CPU overhead
  - Leaves CPU for data processing—not data collection

Stellaris EPI

- **Multiple device types supported**
  - ✓ **SDRAM:** Supports x16 (Single Data Rate) at up to **50MHz**
    - Supports low-cost SDRAMS up to 64 MB
    - Includes automatic refresh and access to all banks/rows
    - Includes a sleep/standby mode to keep contents alive with minimal power draw
  - ✓ **Host-Bus Interface:** Traditional x8 MCU bus interface capabilities
    - Support of both muxed and de-muxed address and data
    - Access to SRAM, NOR Flash, and other devices, with up to 1MB of addressing (256 MB muxed)
    - Access to a range of devices supporting the non-address FIFO x 8 interface variant, with support for TXempty and RXfull for FIFO empty and FIFO full flags
    - Speed controlled, with read and write data wait-state counters
  - ✓ **Machine-to-Machine:** Wide parallel interfaces for fast communications
    - For instance, CPLDs and FPGAs
    - Data widths up to 32-bits, data rates up to 150 Mbytes/second
    - Optional “address” sizes from 4-bits to 16-bits
    - Optional **clock output**, **read/write strobes**, **framing** (with counter-based size), and **clock-enable input**
- **Other features**
  - ✓ General parallel GPIO, FIFOed with speed control – for custom peripherals or digital controls
  - ✓ Blocking and non-blocking reads
  - ✓ FIFOed writes separate the processor from timing details
  - ✓ Direct memory access (DMA)

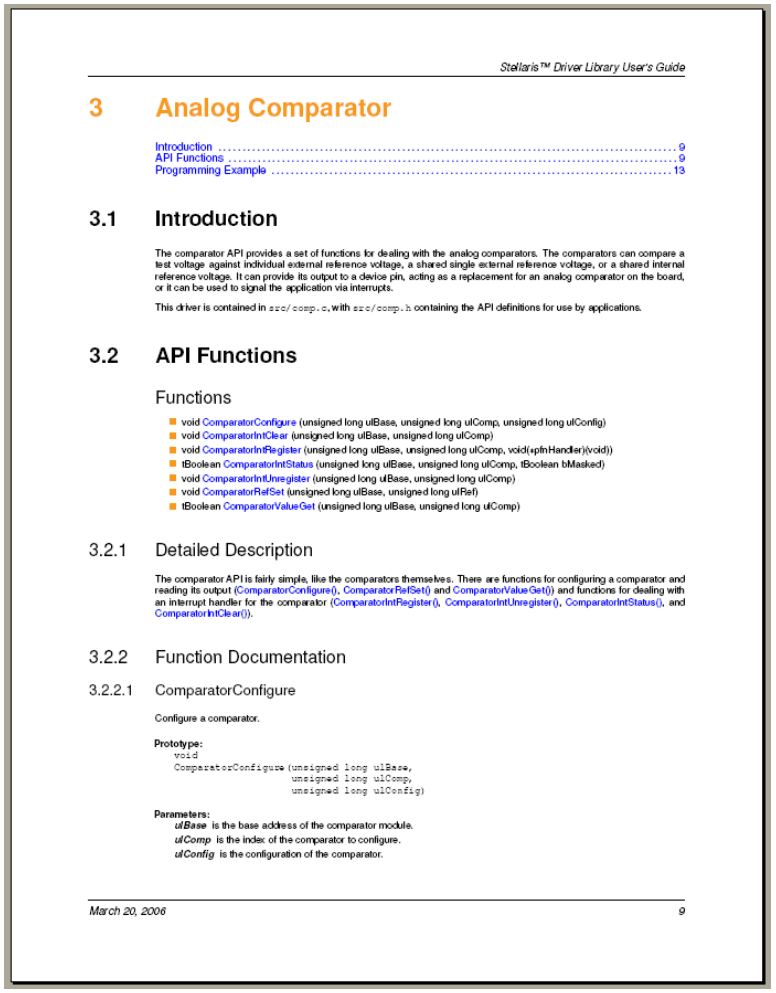
Software

Introducing StellarisWare®

Peripheral driver library

StellarisWare®

- **High-level API interface to complete peripheral set**
- **Free license and royalty-free use**
- **Simplifies and speeds development of applications**
- **Can be used for application development or as programming example**
- **Available as object library and as source code**
- **Compiles on ARM/Keil, IAR, Code Red, and GNU tools**
- **Includes Stellaris Graphics Library and Stellaris USB Library**
- **Peripheral driver library functions are preprogrammed in ROM on select Stellaris MCUs**



Stellaris™ Driver Library User's Guide

3 Analog Comparator

Introduction 9
API Functions 9
Programming Example 13

3.1 Introduction

The comparator API provides a set of functions for dealing with the analog comparators. The comparators can compare a test voltage against individual external reference voltage, a shared single external reference voltage, or a shared internal reference voltage. It can provide its output to a device pin, acting as a replacement for an analog comparator on the board, or it can be used to signal the application via interrupts.

This driver is contained in `src/comp.c`, with `src/comp.h` containing the API definitions for use by applications.

3.2 API Functions

Functions

- void ComparatorConfigure (unsigned long ulBase, unsigned long ulComp, unsigned long ulConfig)
- void ComparatorIntClear (unsigned long ulBase, unsigned long ulComp)
- void ComparatorIntRegister (unsigned long ulBase, unsigned long ulComp, void(*pfnHandler)(void))
- tBoolean ComparatorIntStatus (unsigned long ulBase, unsigned long ulComp, tBoolean bMasked)
- void ComparatorIntUnregister (unsigned long ulBase, unsigned long ulComp)
- void ComparatorRefSet (unsigned long ulBase, unsigned long ulRef)
- tBoolean ComparatorValueGet (unsigned long ulBase, unsigned long ulComp)

3.2.1 Detailed Description

The comparator API is fairly simple, like the comparators themselves. There are functions for configuring a comparator and reading its output (ComparatorConfigure(), ComparatorRefSet() and ComparatorValueGet()) and functions for dealing with an interrupt handler for the comparator (ComparatorIntRegister(), ComparatorIntUnregister(), ComparatorIntStatus(), and ComparatorIntClear()).

3.2.2 Function Documentation

3.2.2.1 ComparatorConfigure

Configure a comparator.

Prototype:

```
void
ComparatorConfigure (unsigned long ulBase,
                     unsigned long ulComp,
                     unsigned long ulConfig)
```

Parameters:

ulBase is the base address of the comparator module.
ulComp is the index of the comparator to configure.
ulConfig is the configuration of the comparator.

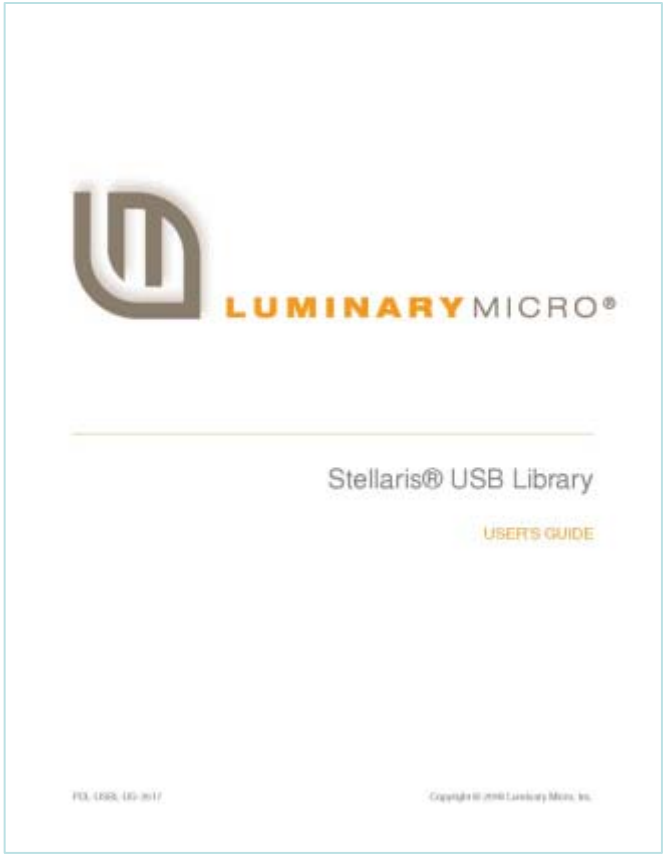
March 20, 2006 9

USB library stacks and examples

StellarisWare®

- **USB-IF Compliance**
  - ✓ Stellaris has passed USB Device and Embedded Host compliance testing

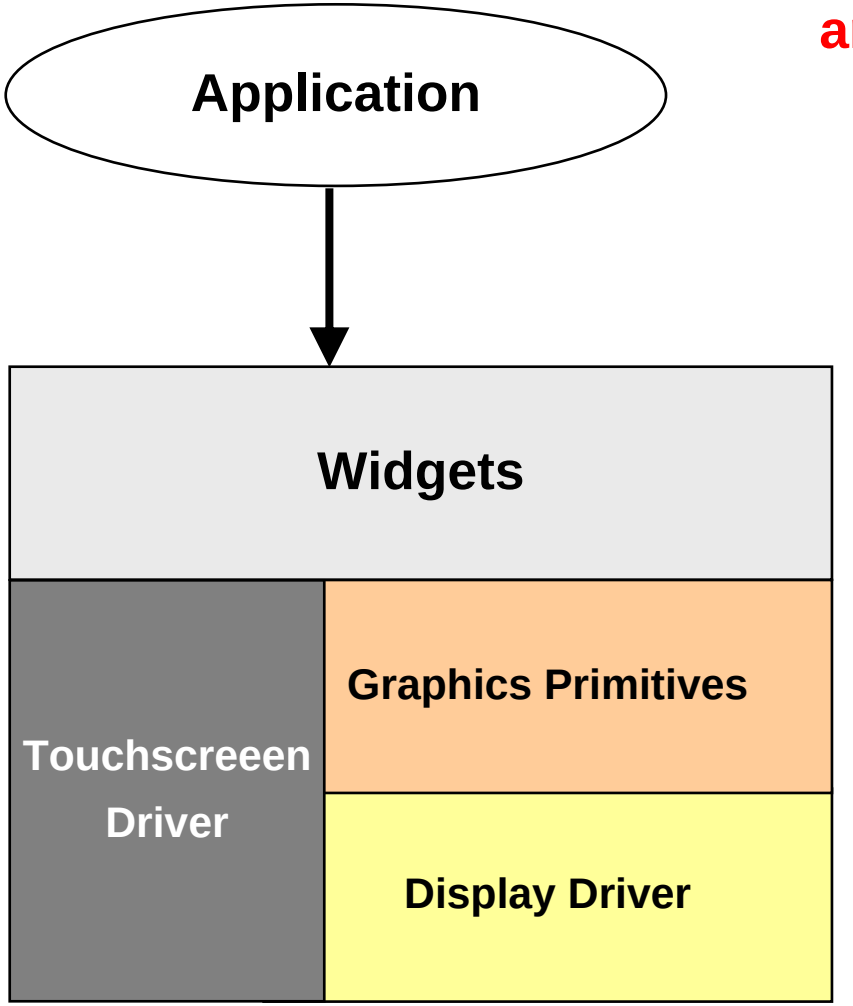
- **Examples available:**
  - ✓ Device Examples:
    - HID Keyboard
    - HID Mouse
    - CDC Serial
    - Generic Bulk
    - Device Firmware Upgrade
    - Oscilloscope
  - ✓ Host Examples:
    - Mass Storage
    - HID Keyboard
    - HID Mouse
  - ✓ Windows INF for supported classes
    - Points to base Windows drivers
    - Sets config string
    - Sets PID/VID
    - Precompiled DLL saves development time
  - ✓ Device framework integrated into USBLib



Stellaris Graphics Library

StellarisWare®

→ **Set of graphics primitives and widgets**



Canvas, Checkbox, Container, Push Button, Radio Button, Slider, ListBox

Lines, Circles, Text, Point, Rectangle, Circle, Image, …

Bitmap/Pixel Control

Safe at home with IEC 60730

StellarisWare®

The International Electrotechnical Commission (IEC)

- IEC: World's authority in international standards for household appliances
- StellarisWare extension provides support for IEC 60730 Class B safety requirements
- Class B covers most home appliances, such as washers/dryers, refrigerators, freezers, and cookers/stoves
- Free license and royalty-free use for use on Stellaris MCUs
- Library supports both startup and periodic testing requirements of IEC 60730

http://www.iec.ch/index.html

	Module	Description
StellarisWare™ Software	Reset Handler	Performs basic register and memory test out of reset.
	CPU Test	Performs stuck bit testing on the CPU PC and registers.
	SRAM Test	Performs stuck bit testing on the SRAM.
	Flash Test	Performs a CRC test on the Flash.
	ADC Test	Performs a conversion test on an ADC channel connected to a known voltage reference.
		Performs ADC temperature sensor test.
	GPIO Test	Performs GPIO input/output plausibility test.
	Clock/Interrupt Test	Performs tests to check the clock frequency, interrupt handling, and execution.
Stellaris® Hardware	Nested Vector Interrupt Controller	Deterministic, fast interrupt processing for execution certainty.
	Automotive-grade Flash Memory	High reliability non-volatile memory for robust environments.
	Cyclical Redundancy Check in ROM	Especially useful in verifying the contents of memory in a Stellaris microcontroller.
	2 Watchdog Timers	Clocked with precision oscillator, a second WDT takes advantage of the non-maskable interrupt (NMI) handler safety feature of the ARM Cortex-M3 processor.
	Precision Oscillator	Supplies an accurate, independent time base when periodic safety tests are executed.
	Advanced Motion Control with Multiple Fault Conditioning Inputs	Provides quick motor shutdown in low latency situations.
	Quadrature Encoder Inputs	Provides precise, closed loop control of motors.
	Integrated Analog Comparators	Used to trigger Stellaris' accurate ADC and to trigger an interrupt when needed, which is useful for infrequent out-of-range events such as a current or voltage spike.
		Eliminates the performance-wasting requirement of constant CPU polling.
	Internal Temperature Sensor	Used to monitor and shut down an appliance if the appliance overheats.
	10/100 Ethernet MAC/PHY with IEEE 1588 PTP	Offers highly synchronized connectivity features for precision internetworking.
	Controller Area Network (CAN) 2.0 MACs	

On-chip software enhancements (ROM)

StellarisWare®

StellarisWare® DriverLib

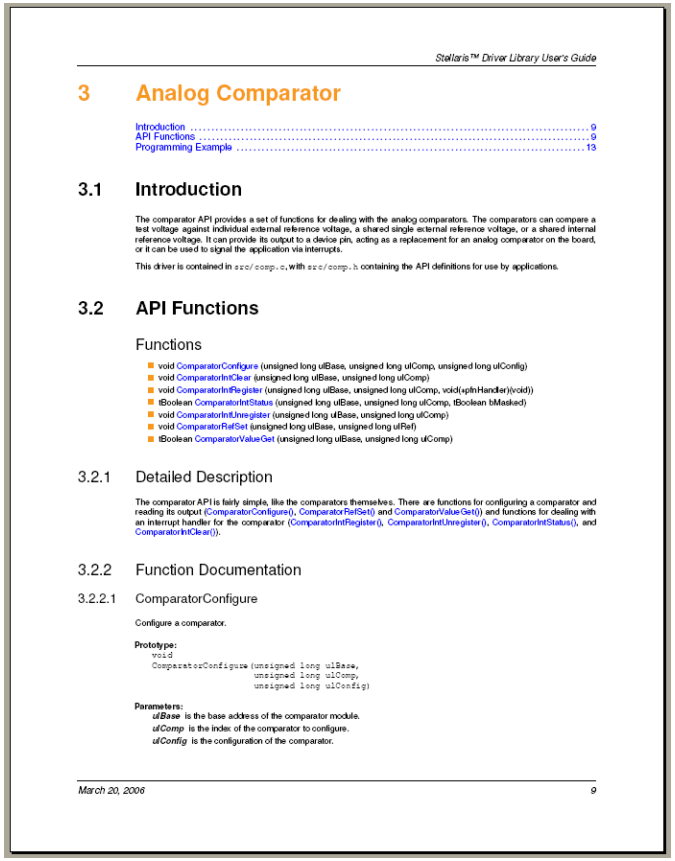
- **High-level API interface to complete peripheral set.**
- **Simplifies and speeds development of applications.**
- **Saves user flash by storing peripheral setup and configuration code**
- **Allows programmer focus to be on the application—not setup**

StellarisWare® Bootloader

- **Download code to flash memory for firmware updates**
- **Interface options include UART (default), I2C, SSI, Ethernet**

Other flash memory-saving options

- **Advanced Encryption Standard (AES) tables – for cryptography**
  - ✓ Supported by the current AES example application
  - ✓ Covers all three sizes: 128, 192, 256
- **Cyclic Redundancy Check (CRC) functionality – for error detection**



Stellaris™ Driver Library User's Guide

3 Analog Comparator

Introduction 9
API Functions 9
Programming Example 13

3.1 Introduction

The comparator API provides a set of functions for dealing with the analog comparators. The comparators can compare a test voltage against individual external reference voltage, a shared single external reference voltage, or a shared internal reference voltage. It can provide its output to a device pin, acting as a replacement for an analog comparator on the board, or it can be used to signal the application via interrupts.

This driver is contained in `src/comp.c`, with `src/comp.h` containing the API definitions for use by applications.

3.2 API Functions

Functions

- void ComparatorConfigure (unsigned long ulBase, unsigned long ulComp, unsigned long ulConfig)
- void ComparatorIntClear (unsigned long ulBase, unsigned long ulComp)
- void ComparatorIntRegister (unsigned long ulBase, unsigned long ulComp, void(*pfnHandler)(void))
- tBoolean ComparatorIntStatus (unsigned long ulBase, unsigned long ulComp, tBoolean bMasked)
- void ComparatorIntUnregister (unsigned long ulBase, unsigned long ulComp)
- void ComparatorRefSet (unsigned long ulBase, unsigned long ulRef)
- tBoolean ComparatorValueGet (unsigned long ulBase, unsigned long ulComp)

3.2.1 Detailed Description

The comparator API is fairly simple, like the comparators themselves. There are functions for configuring a comparator and reading its output (ComparatorConfigure(), ComparatorRefSet() and ComparatorValueGet()) and functions for dealing with an interrupt handler for the comparator (ComparatorIntRegister(), ComparatorIntUnregister(), ComparatorIntStatus(), and ComparatorIntClear()).

3.2.2 Function Documentation

3.2.2.1 ComparatorConfigure

Configure a comparator.

Prototype:

```
void
ComparatorConfigure(unsigned long ulBase,
                    unsigned long ulComp,
                    unsigned long ulConfig)
```

Parameters:

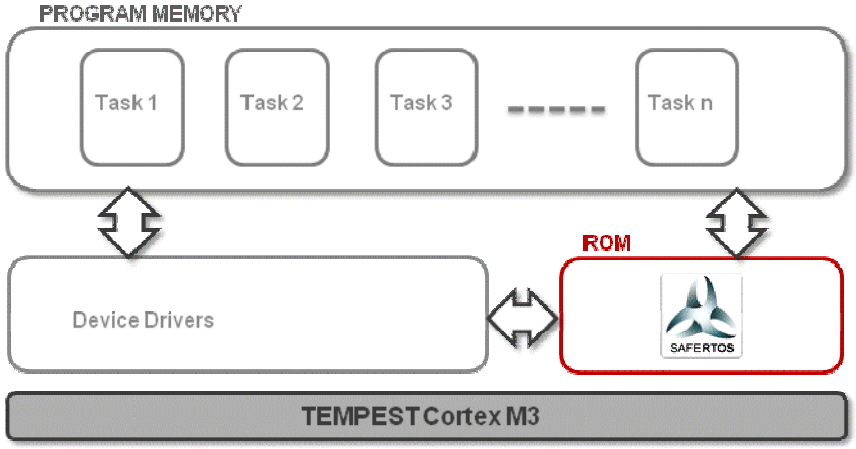
ulBase is the base address of the comparator module.
ulComp is the index of the comparator to configure.
ulConfig is the configuration of the comparator.

March 20, 2006 9

Stored in exclusive ROM on select Stellaris MCUs

SAFERTOS included in the LM3S9B96

- High-integrity RTOS in ROM
- Can be used as a standard operating system *OR* as part of a high integrity application which requires certification to **IEC61508** or **FDA510(k)**

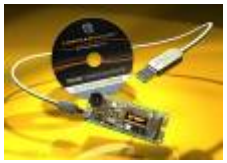


- RTOS **value $65k free** with Tempest LM3S9B96
- Integrated hardware/software solution shortens the time to market and significantly reduces cost for **Industrial** and **Medical** Applications
- Innovative *Design Assurance Pack* available separately from WITTENSTEIN provides **complete turnkey evidence** and process documentation

Tools

Stellaris evaluation kits: “Zero-to-32bits” in 10 minutes

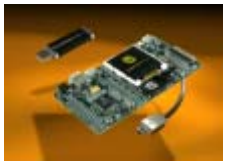
- Everything a developer needs to get up and running in 10 minutes or less
  - evaluation board(s), all required cables, a choice of evaluation tools suites for popular development tools, documentation, StellarisWare software, and applications notes
- Spans the design spectrum from evaluation to prototyping to application-specific design by functioning both as an evaluation platform and as a serial in-circuit debug interface for any Stellaris microcontroller-based target board



EK-LM3S811
Low pin count
$49

EK-LM3S1968
High pin count
$59

EK-LM3S2965
CAN Functionality
$79



EK-LM3S3748
USB Host/Device
$109

EK-LM3S6965
Ethernet MAC+PHY
$69

EK-LM3S8962
Ethernet+CAN
$89

EK-LM3S9B90
Ethernet+USB OTG
$99

EK-LM3S9B92
Ethernet+OTG+MC
$99

Four 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 full evaluation license locked to board

Stellaris RDK - Open-Tool Motor

RDK-ACIM
$379

AC Induction Motor Controller Design

Example applications:

- White goods
- Residential and light commercial HVAC
- 3-ph Industrial Motor Drives

RDK-STEPPER
$199

Stepper Motor Controller Design

Example applications:

- 2 and 3 axis CNC equipment
- Sorting and grading equipment
- Specialized printers and scanners

RDK-BLDC
$219

Brushless DC Motor Controller with CAN/Ethernet

Example applications:

- Small appliances
- Electric wheelchairs and mobility devices
- Pumping and ventilation systems

RDK-BDC
$219

Brush DC Motor Controller with CAN

Example applications:

- Small appliances
- Electric wheelchairs and mobility devices
- Pumping and ventilation systems

Official FIRST KoP Speed Controller – FRC 2009

Stellaris RDK - Open-Tool

RDK-IDM
$219

Touch-screen Intelligent Display Module with PoE

Example applications:

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

RDK-IDM-L35
$219

Landscape-oriented Touch-screen Intelligent Display Module

Example applications:

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

RDK-IDM-SBC
$299

Stellaris 3.5" Landscape IDM Single Board Computer

Example applications:

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

RDK-S2E
$139

Tiny Footprint Serial-to-Ethernet Module

Example applications:

- SCADA Remote Terminal Units (RTUs)
- Electronic Flow Meters (EFMs)
- CCTV RS-232 Recorders

Stellaris Development Kits

Only $249!

Only $425!

- **DK-LM3S1xx/3xx/8xx Development Kits**
  - ✓ Development Board with a choice of daughtercard
  - ✓ UART transceiver and DB9 male connector
  - ✓ All I/O available on headers
  - ✓ One potentiometer and one photocell for driving the comparator inputs
  - ✓ Eight user LEDs and one pushbutton for use with the Stellaris™ GPIOs
  - ✓ Standard ARM® 20-pin JTAG debug connector
  - ✓ USB 2.0 full-speed interface allows JTAG/SWD debug
  - ✓ 1 Mbit SPI-based flash memory
  - ✓ One buzzer for PWM use
  - ✓ User-prototype area

- **DK-LM3S9B96 Development Kit**
  - ✓ 80 MHz Stellaris LM3S9B96 MCU with fully-integrated Ethernet, CAN, and USB OTG/Host/Device
  - ✓ Bright 3.5” QVGA LCD touch-screen display
  - ✓ Navigation POT switch and select pushbuttons
  - ✓ Integrated Interchip Sound (I2S) Audio Interface
  - ✓ EPI cards: I/O break-out board and 8 MB SDR SDRAM module
  - ✓ MicroSD card interface
  - ✓ LM3S9B96 I/O available on labeled break-out pads
  - ✓ ARM® 10-pin JTAG debug connector with input and output modes

KEIL An ARM® Company · code_red · IAR SYSTEMS · CODESOURCERY

Summary

Stellaris® value proposition

High performance Features

20-100 MHz ARM-M3 CPU

- Optimized for single-cycle flash usage
- Integrated 32-ch DMA for ease of use & high data rate without CPU overhead
- Thumb-2 ISA with high code density
- Flexible clock system sources up to 8 timers
- Single-cycle multiply and hardware divide
- Three power modes and battery-backed hibernation with non-volatile memory

Connectivity

- Ethernet MAC & PHY with 1588 PTP support
- USB Host, Device, or On-The-Go
- CAN 2.0 A/B with 32 mailboxes
- External Peripheral Interface supporting SRAM, SDRAM, M2M, FPGA, CPLD
- Integrated UART, I2C, SSI module
- Integrated I2S master or slave

Broad Portfolio

- ***Largest ARM MCU portfolio in the world with 138 devices***
- 8KB to 256KB Flash and up to 96KB RAM
- Up to 8 advanced PWM modules
- RTC, and integrated LDO
- Analog comparators and temp sensor
- 28 to 108 pin from SOIC to BGA
- 10-bit, 8ch ADCs from 250ksps-1MSPS

Speed to Market

- StellarisWare on ROM includes driver and peripheral libraries to ease development
- C friendly IDE and compilers from industry leaders
- Low cost development tools
- Application specific and advanced development kits
- Production-ready application modules