TI Technology Days 2010

Stellaris® ARM® Cortex™-M3 based MCUs

Yunus Karaborek

Stellaris® ARM® Cortex™-M3 based MCUs

1 Introduction

2 Stellaris solutions overview

3 Platform overview & architecture

4 Motion control

5 Communication features

6 Stellarisware & Tools

Introduction

- **Learning objectives**
  - ✓ Introduction to Cortex M3
  - ✓ Stellaris portfolio overview
  - ✓ Stellaris architecture and HW features
  - ✓ StellarisWare libraries use

Stellaris® in TI Processer Portfolio

Microcontrollers				Applications Processors / DSP		
16-bit	32-bit Real-time	32-bit M3 ARM	32-bit R4F ARM`	32-bit ARM+	32-bit ARM+DSP	DSP
ARM Core Offerings						
MSP430	C2000	Stellaris-M3	TMS570	ARM9 ARM Cortex-A8	ARM9/Cortex-A8 plus C64x+	C55x, C64x+ C647x
Ultra-low Power	Fixed & Floating Point	Industry Std Low Power	Floating Point	Industry-Std Core, High-Perf GPP	Industry-Std Core + DSP for Signal Proc.	Leadership DSP Performance
Up to 25 MHz	Up to 300 MHz	Up to 80 MHz	Over 250 DMIPS	Accelerators	4800 MMACs/ 1.07 DMIPS/MHz	24,000 MMACS
Flash 1 KB to 256 KB	Flash 32 KB to 512 KB	Flash Up to 512 KB	Flash Up to 3 MB	MMU	MMU, Cache	Up to 3 MB L2 Cache
Analog I/O, ADC LCD, USB, RF	PWM, ADC, CAN, SPI, I^2C	USB, ENET, ADC, PWM, CAN	Timer co-proc ADC, CAN	USB, LCD, MMC, EMAC	VPSS, USB, EMAC, MMC	1G EMAC, SRIO, DDR2, PCI-66
Measurement, Sensing, General Purpose	Motor Control, Digital Power, Lighting	Host Control	Safety SIL3 Control	Linux/WinCE User Apps	Lin/Win O/S + Video, Imag, MM	Comm, WiMAX, Industrial/ Medical Imaging
$0.49 to $9.00	$1.50 to $20.00	$1.00 to $7.00	$7.00 to $18.00	$8.00 to $35.00	$12.00 to $65.00	$4.00 to $99+

Software, Tools & BSPs

Texas Instruments Developer Network

Texas Instruments eXpressDSP™

Instruments

Stellaris® ARM® Cortex™-M3 based MCUs

1 Introduction

2 Stellaris solutions overview

3 Platform overview & architecture

4 Motion Control

5 Communication features

6 Stellarisware & Tools

Stellaris® value proposition

High performance Features

20-80 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

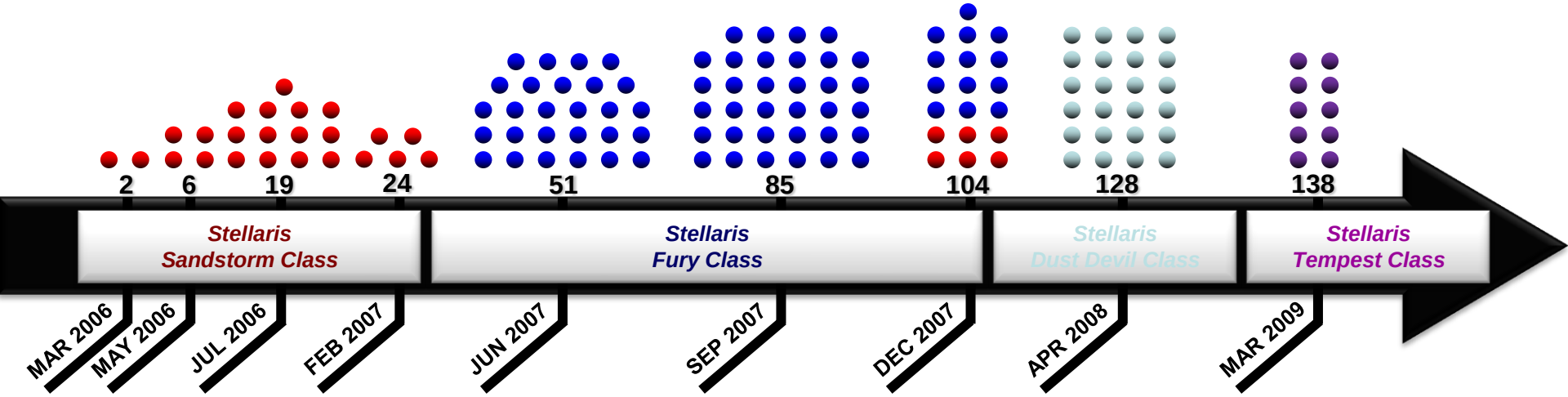
- StellarisWare on ROM includes driver and peripheral libraries to ease development
- C friendly IDE and compilers from industry leaders
- Low cost development tools
- Application specific and advanced development kits
- Production-ready application modules

Stellaris® broad portfolio

First in ARM Cortex-M3 uC

- **TI AEC Austin: Stellaris MCU Design, Systems, Software, Product, and Applications Engineering**

- **Stellaris® family has over 160 microcontrollers!**
  - ✓ Broad line card of mixed-signal microcontrollers focused on applications in energy, security, and connectivity markets
  - ✓ Unique IP for motion control applications, real time connectivity (Ethernet, Controller Area Network, and USB), intelligent analog functionality, and power conservation
  - ✓ Experience fastest time-to-market for the most cost effective, standardized, market-leading solutions through extensive Stellaris hardware tools, StellarisWare® software, documentation, technical support, and ARM's vast 3rd party ecosystem



Stellaris® Family Overview

Over 160 family members

20, 25, 50 and 80MHz

8K to 256K Flash

2K to 96K SRAM

In 48-QFN

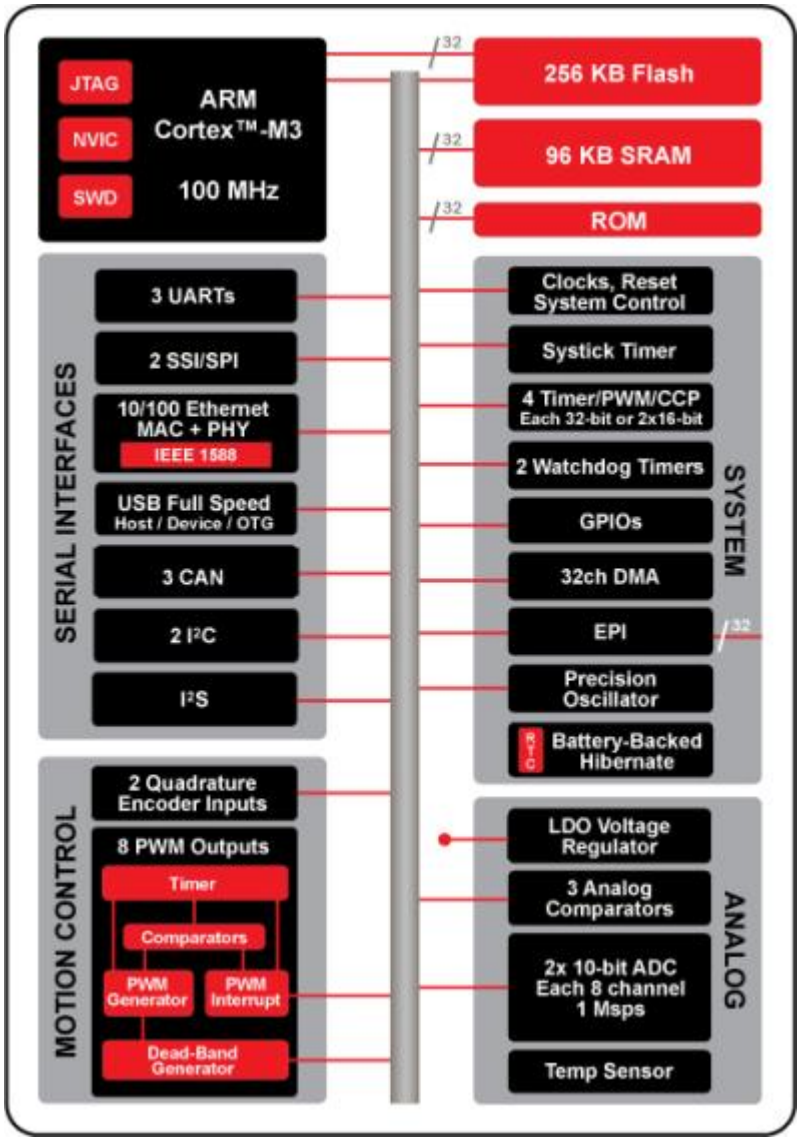
48-LQFP

64-LQFP

100-LQFP

and 108-BGA

Industrial & Extended Temperature



Pb

RoHS COMPLIANT 2002/95/EC

Stellaris Family Portfolio

Production

Sampling

Features

Ethernet Series

6000 Series
- up to 50MHz
- up to 256kB Flash
- 10/100 Ethernet (MAC+PHY) + IEEE1588

8000 Series
- up to 50MHz
- up to 256kB Flash
- 10/100 Ethernet (MAC+PHY) + IEEE1588
- up to 3x CAN 2.0

9000 Series
- up to 80MHz
- up to 256kB Flash
- 10/100 Ethernet (MAC+PHY) + IEEE1588
- USB Full Speed Host/Device/OTG
- up to 2x CAN 2.0

CAN & USB Series

2000 Series
- up to 80MHz
- up to 256kB Flash
- up to 2x CAN 2.0

3000 Series
- up to 50MHz
- up to 256kB Flash
- USB Full Speed Host/Device/OTG

5000 Series
- up to 80MHz
- up to 256kB Flash
- USB Full Speed Host/Device/OTG
- up to 2x CAN 2.0

General Purpose Series

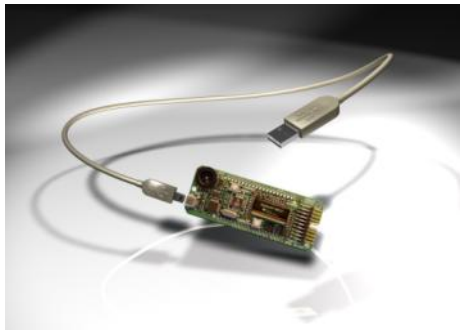
x00 Series
- up to 50MHz
- up to 64kB Flash
- std. Peripherals
- low pin-count

1000 Series
- up to 50MHz
- up to 256kB Flash
- std. Peripherals
- high pin-count

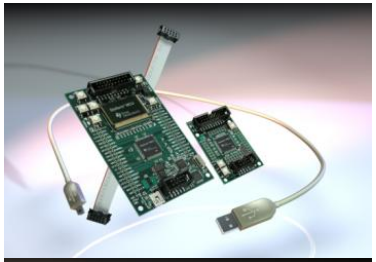
160+ Code Compatible Devices

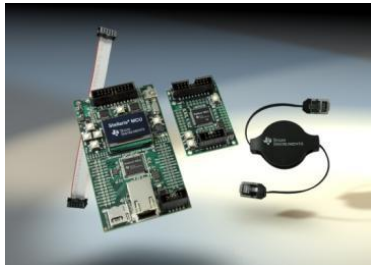
Portfolio - General Purpose Series

- LM3S**x00** – 48LQFP/48QFN
  - 20MHz – 50MHz
  - 8K – 64K Flash, 2K – 8K SRAM
- LM3S**1**xxx - 64/100LQFP, 108-BGA
  - 25MHz – 50MHz
  - 16K – 256K Flash, 6K – 64K SRAM
- Peripherals include:
  - ADC, ACOMP
  - SPI, UART, I2C
  - PWM, QEI, MPU
  - WDT, GP Timers, Capture/Compare
- **Evaluation Kits**
  - EK**x**-LM3S811 (48LQFP)
  - EX**x**-LM3S1968 (100LQFP)
    - '**x**'= 'K'-Keil,'I'-IAR, 'C'-Codesourcery, 'T'-CodeRed, 'S'-Code Composer Studio



Portfolio - Advanced Communications Series (1 of 2)

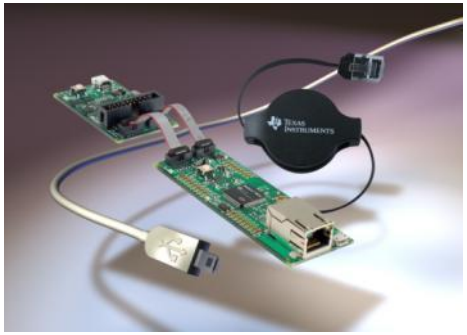




- LM3S**2**xxx - CAN (up to 2)
- LM3S**3**xxx - USB Host/Device/OTG
- LM3S**5**xxx - USB Host/Device/OTG + CAN (up to 2)
- LM3S**6**xxx - Ethernet MAC & PHY
- LM3S**8**xxx - Ethernet MAC & PHY + CAN (up to 3)
  - 25MHz – 80MHz
  - 64K – 256K Flash, 16K – 96K SRAM
  - 64/100LQFP, 108-BGA
  - Peripherals include:
    - SPI, UART, I2C
    - ADC, ACOMP
    - PWM, QEI, MPU, EPI
    - WDT, GP Timers, Capture/Compare
- **Evaluation Kits**
  - EX**x**-LM3S2965 (100LQFP)
  - EX**x**-LM3S3748 (100LQFP)
  - EX**x**-LM3S6965 (100LQFP)
  - EX**x**-LM3S8962 (100LQFP)
    - '**x**'= 'K'-Keil,'I'-IAR, 'C'-Codesourcery, 'T'-CodeRed , 'S'-Code Composer Studio

Portfolio - Advanced Communications Series (2 of 2)

- LM3S**9**xxx - Ethernet MAC & PHY + USB(OTG) + CAN (2)
  - 80MHz
  - 64K – 256K Flash, 64K – 96K SRAM
  - 100LQFP
  - Peripherals include:
    - SPI(2), UART(3), I2C(2)
    - ADC(16), ACOMP(3), DCOMP(16)
    - PWM, QEI, MPU, EPI
    - WDT(2), GP Timers, Capture/Compare



  - **Evaluation Kits**
    - EX**x**-LM3S9B90 (100LQFP)
    - EX**x**-LM3S9B92 (100LQFP)
      - '**x**'= 'K'-Keil,'I'-IAR, 'C'-Codesourcery, 'T'-CodeRed , 'S'-Code Composer Studio
  - **Development Kit**
    - DK-LM3S9B96

Stellaris® product line summary

- LM3Sx00 General purpose, 48pin, no CAN, no ETH, no USB
- LM3S1xxx General purpose, 64/100/108pin, no CAN, no ETH, no USB

- LM3S2xxx CAN, 64/100/108 pin
- LM3S3xxx USB, 64/100/108 pin
- LM3S5xxx CAN + USB, 64/100/108 pin

- LM3S6xxx Ethernet , 64/100/108 pin
- LM3S8xxx Ethernet + CAN , 64/100/108 pin
- LM3S9xxx Ethernet + CAN + USB , 64/100/108 pin

Stellaris ® Product Selector Guide (PSG)

- **Important tool for choosing Stellaris M3**
- **Contains information on**
  - ✓ All Devices
    - In production
    - Sampling
  - ✓ Evaluation kits
  - ✓ Development Kit
  - ✓ Reference Design kits

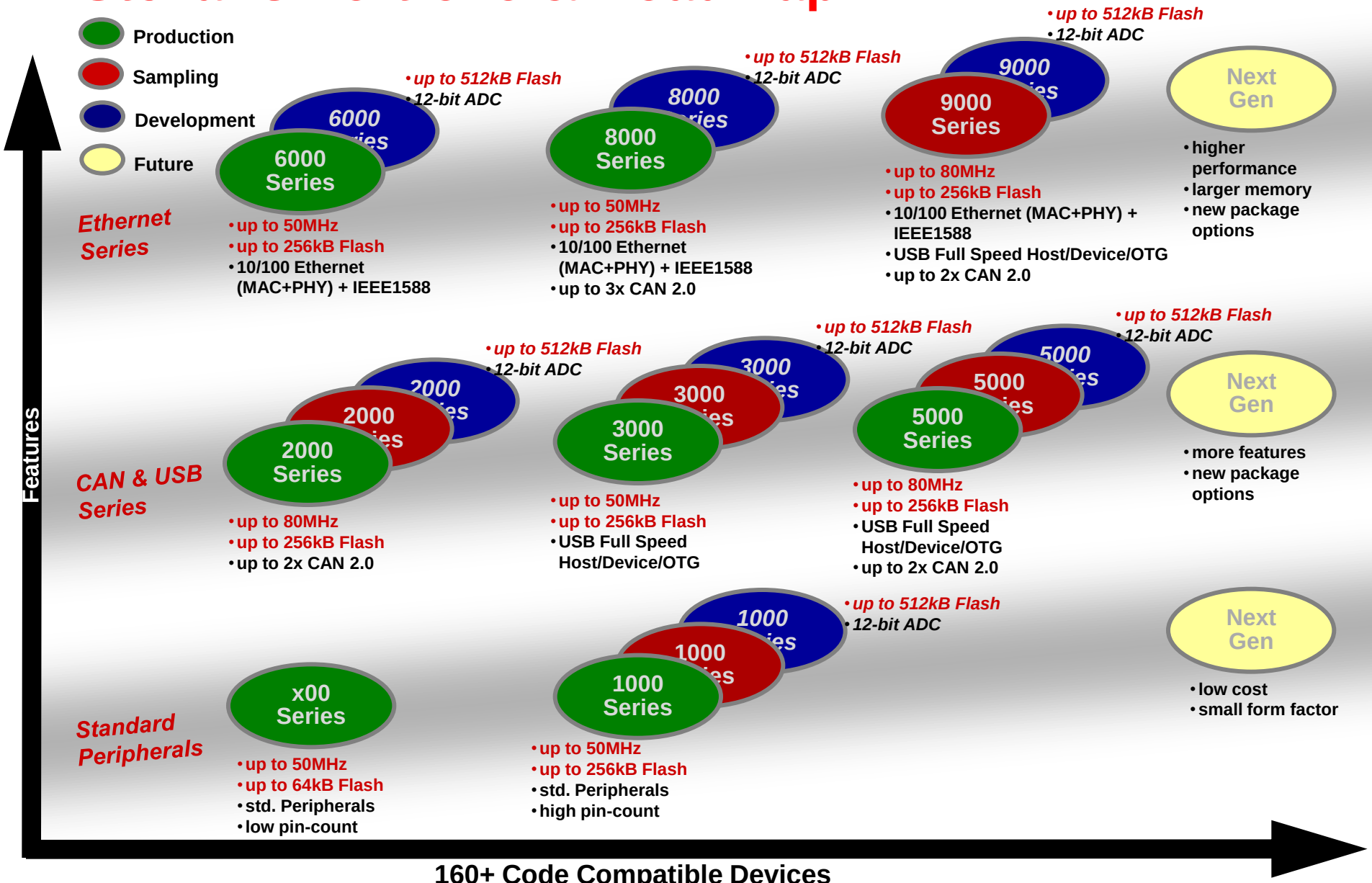
Interactive Product Selection Guide

Flash (KB)	>=	8	16	32	64	96	128	256																					
SRAM (KB)	>=	2	4	8	16	32	64	96																					
Max Speed (MHz)	>=			20	25	50	80	100																					
32-bit General Purpose Timers	>=					2	3	4																					
Watchdog Timers							1	2																					
Motion control PWM	>=			0	2	4	6	8																					
Capture & Compare Pins (CCP)	>=	1	2	4	5	6	7	8																					
10-bit ADC Channels	>=	0	2	3	4	6	8	16																					
USB			NONE		Host/Device		OTG/Host/Device																						
Ethernet MAC/PHY	✓																												
IEEE1588 support	✓																												
StellarisWare in ROM	✓																												
CAN	>=					0	1	2	3																				
UART	>=						1	2	3																				
I2C	>=						0	1	2																				
SSI (SPI)								1	2																				
Analog Comparators	>=					0	1	2	3																				
Quad Encoder Input	>=						0	1	2																				
Max GPIO						≤30	>30	>40	>50	>60																			
ADC Sample Rate	>=					0	250K	500K	1M																				
Package						28 SOIC	48 LQFP	64 LQFP	100 LQFP	108 BGA																			
Direct Memory Access	✓																												
Memory Protection Unit	✓																												
Precision Oscillator	✓																												
External Peripheral Interface	✓																												
Battery Backed Hibernate	✓																												
I2S	✓																												
Temperature Sensor	✓																												
Digital Comparator	✓																												
RESET ALL CRITERIA																													

1 to 18 of 138 results

	Flash (KB)	SRAM (KB)	Max Speed (MHz)	MPU	Systick	32-bit Timers	Motion control PWM	CCP	USB	Ethernet MAC/PHY	IEEE1588 support	CAN	UART	I2C	SSI (SPI)	Quad Encoder Input	10-bit ADC Channels	ADC Sample Rate	Temperature Sensor	Analog Comparators	MAX GPIO	Battery Hibernate	Package	Watchdog timers	StellarisWare in ROM	Direct Memory Access	Precision Oscillator	Ext.Peripheral Interface	I2S	Digital Comparator
LM3S2616	128	16	50	✓	✓	4	6	-	NA	-	-	1	1	1	-	1	6	1M	✓	2	1-33	✓	64 LQFP	1	✓	✓	-	-	-	-
LM3S102	8	2	20	-	✓	2	-	2	NA	-	-	-	1	1	1	-	-	NA	-	1	0-18	-	28 SOIC	1	-	-	-	-	-	-
LM3S101	8	2	20	-	✓	2	-	1	NA	-	-	-	1	-	1	-	-	NA	-	2	2-18	-	28 SOIC	1	-	-	-	-	-	-
LM3S300	16	4	25	✓	✓	3	-	6	NA	-	-	-	2	1	1	-	-	NA	-	3	8-36	-	48 LQFP	1	-	-	-	-	-	-
LM3S308	16	4	25	✓	✓	3	-	6	NA	-	-	-	2	1	1	-	8	500K	✓	1	5-28	-	48 LQFP	1	-	-	-	-	-	-
LM3S328	16	4	25	✓	✓	3	-	6	NA	-	-	-	2	1	1	-	8	500K	✓	-	7-28	-	48 LQFP	1	-	-	-	-	-	-
LM3S6100	64	16	25	✓	✓	3	-	4	NA	✓	-	-	1	-	1	-	-	NA	-	1	10-30	-	100 LQFP	1	-	-	-	-	-	-
LM3S2139	64	16	25	✓	✓	3	-	6	NA	-	-	1	2	1	1	-	4	250K	✓	3	26-56	-	100 LQFP	1	-	-	-	-	-	-
LM3S1110	64	16	25	✓	✓	3	-	2	NA	-	-	-	2	-	1	-	-	NA	-	2	20-41	✓	100 LQFP	1	-	-	-	-	-	-
LM3S6422	96	32	25	✓	✓	3	-	4	NA	✓	-	-	1	-	1	-	2	250K	✓	2	12-34	-	100 LQFP	1	-	-	-	-	-	-
LM3S6420	96	32	25	✓	✓	3	-	4	NA	✓	-	-	1	-	1	-	-	NA	-	2	23-46	-	100 LQFP	1	-	-	-	-	-	-
LM3S2410	96	32	25	✓	✓	3	-	4	NA	-	-	1	1	-	1	-	-	NA	-	2	37-60	-	100 LQFP	1	-	-	-	-	-	-
LM3S628	32	8	50	✓	✓	3	-	4	NA	-	-	-	2	1	1	-	8	1M	✓	-	9-28	-	48 LQFP	1	-	-	-	-	-	-
LM3S608	32	8	50	✓	✓	3	-	6	NA	-	-	-	2	1	1	-	8	500K	✓	1	5-28	-	48 LQFP	1	-	-	-	-	-	-
LM3S600	32	8	50	✓	✓	3	-	6	NA	-	-	-	2	1	1	-	-	NA	-	3	8-36	-	48 LQFP	1	-	-	-	-	-	-
LM3S828	64	8	50	✓	✓	3	-	6	NA	-	-	-	2	1	1	-	8	1M	✓	-	7-28	-	48 LQFP	1	-	-	-	-	-	-
LM3S800	64	8	50	✓	✓	3	-	6	NA	-	-	-	2	1	1	-	-	NA	-	3	8-36	-	48 LQFP	1	-	-	-	-	-	-
LM3S808	64	8	50	✓	✓	3	-	6	NA	-	-	-	2	1	1	-	8	500K	✓	1	5-28	-	48 LQFP	1	-	-	-	-	-	-

Stellaris Portfolio & Roadmap



Texas Instruments

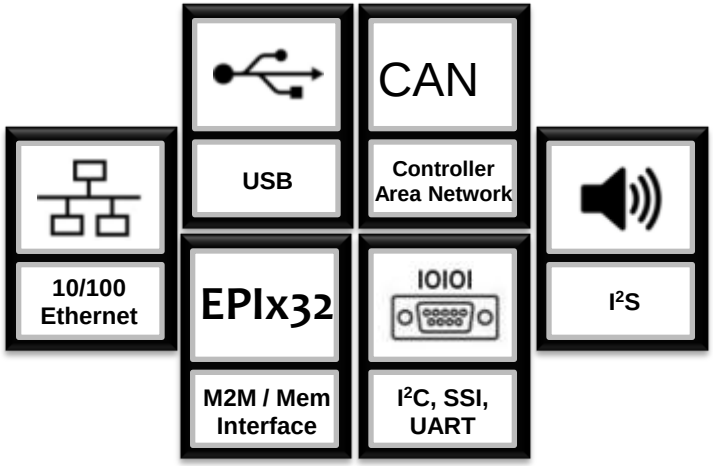
Stellaris® value proposition

Stellaris® Means:

Serious Connectivity

The only ARM MCU with 10/100 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 (Stellaris BGA with Ethernet MAC/PHY just slightly larger than a standard Ethernet PHY)



USB 2.0 Full Speed:

- Enables USB connectivity supporting DEVICE, HOST and OTG

CAN Networking

Serial Communications

- UART, SPI, I2C, I2S

Stellaris® Means:

Texas Instruments

High Speed Memory Performance

Single-Cycle Flash memory up to a maximum 50MHz!

- Some competitors claim faster core speeds with ARM7 and Cortex-M3, but the flash is not single-cycle!
- 1-KB Sectors typically
- 100K cycles typically
- **>50MHz** use of an optimised, advanced pre-fetch buffer

SRAM

- Up to 80MHz
- Bit-band capability
- Code execution

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

Flash access specifications from published datasheets

Texas Instruments

Powerful External Peripheral Interface (EPI)

- 8/16/32-bit dedicated parallel bus for external peripherals and memory
- 3 functional modes and supported devices
  - **SDRAM**: Supports x16 (Single Data Rate) at up to 50MHz
  - Host-Bus Interface: Traditional x8 MCU bus (**Flash / SRAM** support)
  - Machine-to-Machine: Wide parallel interfaces for fast communications
- Memory interface supports contiguous memory access independent of data bus width, thus enabling code execution directly from SDRAM, SRAM and Flash memory
- Blocking and non-blocking reads
- Separates processor from timing details through use of an internal write FIFO

Pin compatibility between 64 pin devices

Horizontal columns show pin compatible devices

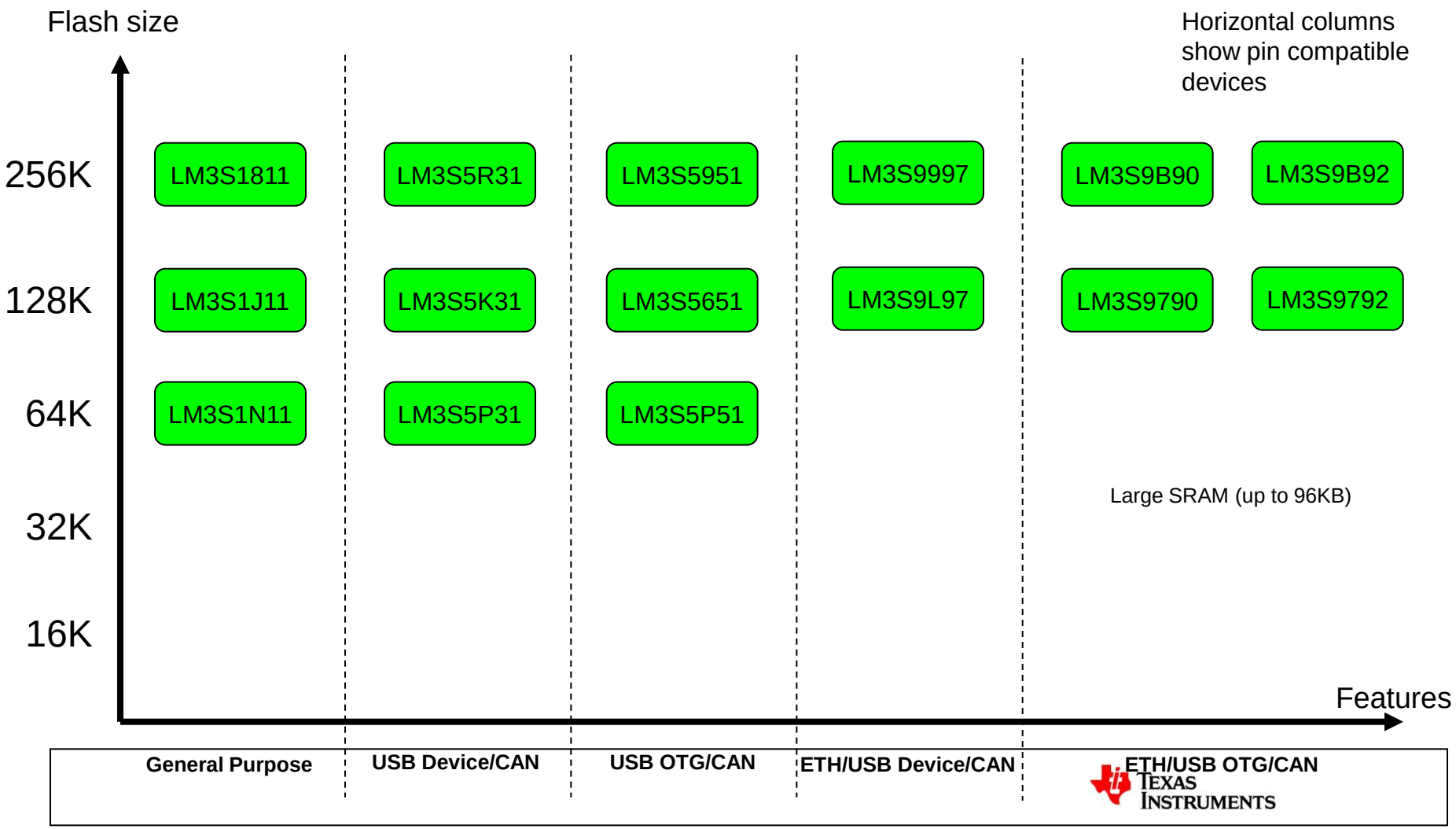
Flash size	General Purpose	USB Device	USB Device/CAN	USB OTG/CAN
256K	LM3S1816	LM3S3826	LM3S5R36	LM3S5956
128K	LM3S1J16	LM3S3J26	LM3S5K36	LM3S5656
64K	LM3S1N16	LM3S3N26	LM3S5P36	LM3S5P56
32K	LM3S1W16	LM3S3W26	LM3S5T36	
16K	LM3S1Z16	LM3S3Z26	LM3S5Y36	

Features

Stellaris® Means:

Texas Instruments

Pin compatibility between 100 pin devices



Stellaris® Means:

Meticulous Motion Control…

- Stellaris supports 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.
- Most competitors do not even have motion-control PWMs.
- **Stellaris Exclusive!** Up to 4 fault-condition handling inputs in hardware quickly provide low-latency shutdown.
- **Stellaris Exclusive!** Up to 2 Quadrature Encoder Inputs provide accurate positioning for closed-feedback control.
- **Remember:** Stellaris integrated connectivity features: motion + connectivity is possible only with a deterministic architecture like ARM Cortex-M3!

StellarisWare® - Minimizes software time to market

- With Stellaris microcontrollers, you can choose to keep all your programming in C/C++, even interrupt service routines and startup code
- TI makes it even easier by providing StellarisWare® software that **includes source code and royalty-free** libraries for applications support
- StellarisWare® Software compiles on:
  - **ARM/Keil Microcontroller Development Toolkit for ARM**
  - **IAR Embedded Workbench**
  - **Code Composer Studio**
  - **Code Red Technologies' RedSuite**
  - **Code Sourcery SourceryG++**
  - **Generic GNU development tools**.
- The key functional areas are:
  - **Stellaris Peripheral Driver Library**
  - **Stellaris Graphics Library**
  - **Stellaris USB Library**
  - **Stellaris IEC 60730 Library**
  - **Code Samples (USB, ETH, Graphics, etc)**
  - **Stellaris In-System Programming Support**

- Download/Install the whole thing (all boards/all packages), or download/install a specific StellarisWare Firmware Development Package built for your specific Stellaris hardware tool.
- StellarisWare is **preprogrammed in ROM** on most Stellaris MCUs
- Learn More: http://www.ti.com/stellaris

Stellaris® Means:

Texas Instruments

World Class Support

Compilers, Debuggers	KEIL An ARM® Company; IAR Systems; code_red; Rowley Associates; hitex Development Tools; Code Composer Studio; CodeSourcery; iSYSTEM
RTOS	KEIL An ARM® Company; IAR Systems; Micriμm; μTasker; CMX Systems; PUMPKIN Incorporated; freeRTOS; expresslogic; SEGGER; SCIOPTA; Quadros Systems Inc.
Stacks, Specialty	SEGGER; PCB Matrix; IXXAT; Willert Software Tools – Pioneers in Embedded Software Engineering; Quadros Systems Inc.; interniche technologies, inc.; WhizNets; SEVENSTAX Embedded Technologies; Micriμm; port; expresslogic; National Instruments; Micromint USA; RoweBots; CMX Systems
Training Partners	FEASER; hitex; DOULOS; mbest ARM工具关键合作伙伴
Programmers	BPM Microsystems; System General; XELTEK; Zhiyuan Electronics

Stellaris Speed to market

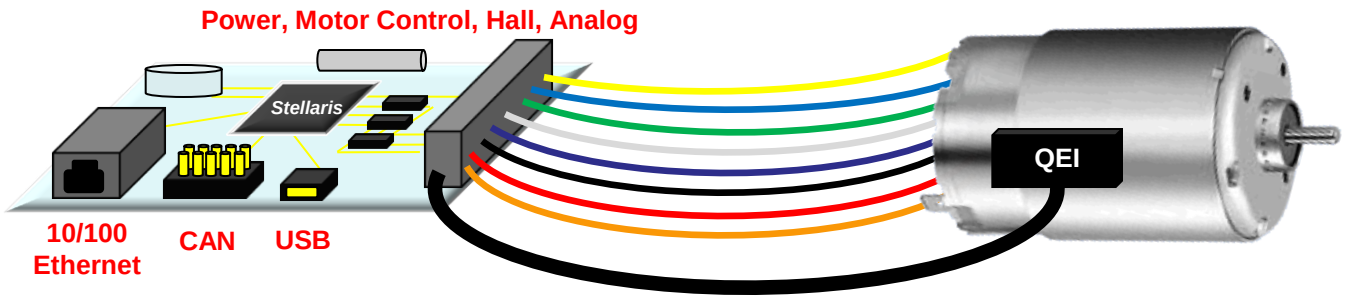
The Full-Solution Approach

Fully Integrated Stellaris MCUs

- ARM Cortex-M3 core with single-cycle Flash
- Advanced Motion Control
- Integrated Deterministic Connectivity
- Easy adoption / learning curve through 10-min Out-of-the-Box Evaluation Kits

Production-ready Modules

- Customizable modules for drop-in implementation
- Multiple motors supported
- Multiple connectivity options
- Copy-exactly with Open-tooled HW and SW



Complete Open-tooled RDKs

- Open-tooled HW/SW Reference Design Kits
- Motor included for out-of-the-box demonstration
- Fully documented, available for download, and in stock

Proof-of-Concept

- Stellaris MCUs / Modules
- Putting our Motion Control to the test before you do.

End-to-End Solution Source Files

Royalty-Free

Schematics

Placement

Bill of Materials

Gerbers

Motor App and StellarisWare® Source

Control / Config GUI

Stellaris® ARM® Cortex™-M3 based MCUs

1 Introduction

2 Stellaris solutions overview

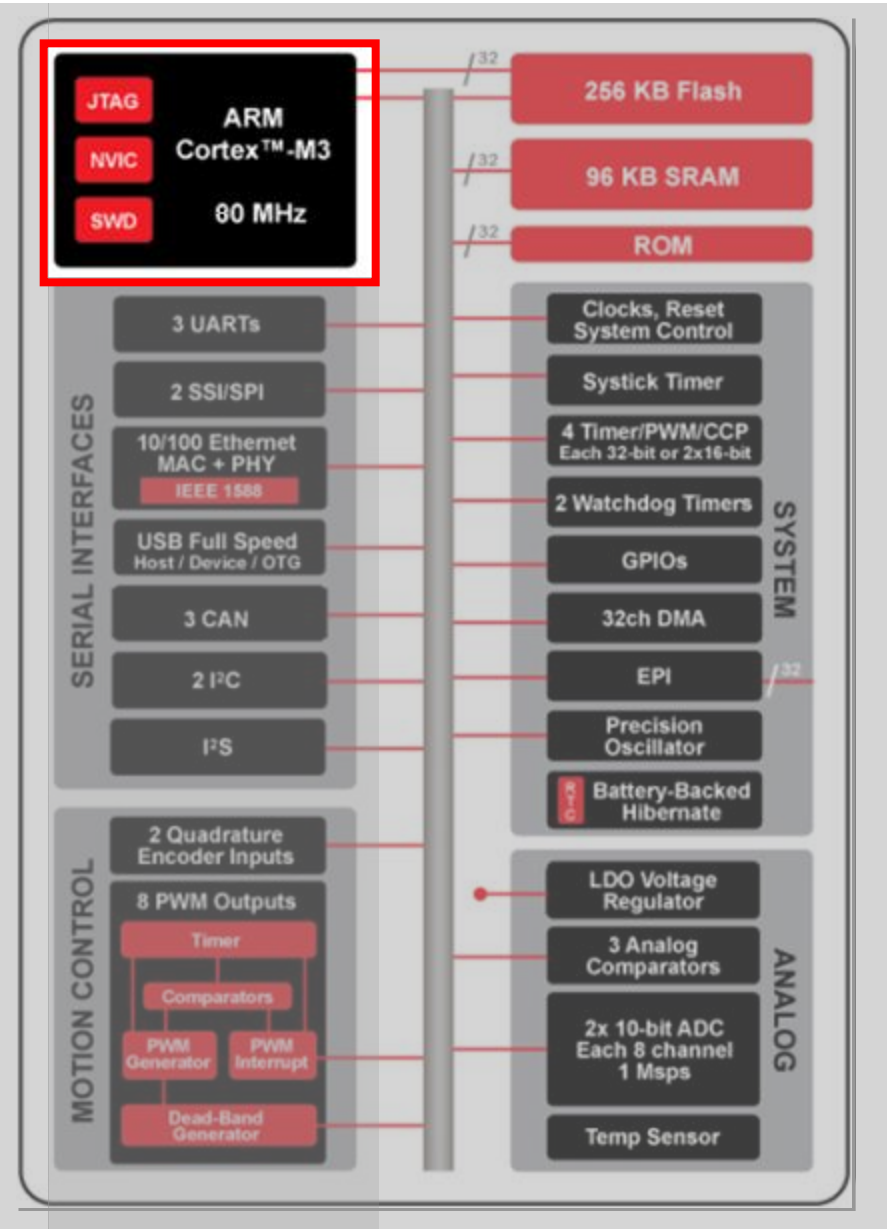
3 Platform overview & architecture

4 Motion control

5 Stellarisware & Tools

6 Communication features

Cortex M3 Core



Cortex M3 core: ARM series

- **ARM Cortex provides a range of solutions**
- **Different series aligning technology around specific applications**
- **3 series implementing the Thumb-2 instruction set:**
  - ✓ ARM Cortex-A Series
    - Applications processors for complex OS and user applications.
    - Supports the ARM, Thumb and Thumb-2 instruction sets
  - ✓ ARM Cortex-R Series
    - Embedded processors for real-time systems.
    - Supports the ARM, Thumb, and Thumb-2 instruction sets
  - ✓ ARM Cortex-M Series
    - Deeply embedded processors
    - optimized for cost sensitive applications.
    - Supports the Thumb-2 instruction set only

Note:

ARM Code 32-bit

Thumb Code 16-bit

Thumb-2 Code mostly 16-bit & some 32-bit (25% Faster, 26% Smaller)

Cortex™
Intelligent Processors by ARM®

Cortex M3 core: Features

- Optimized for single-cycle flash usage
- Deterministic, fast interrupt processing: as low as six cycles, no more than twelve
- Single-cycle multiply instruction and hardware divide
- Native Thumb2 mixed 16-/32-bit instruction set - no mode switching
- Three sleep modes with clock gating for low power
- Superior debug features including data breakpoints and flash patching
- Atomic operations - read/modify/write in single instruction
- 1.25 DMIPS/MHz - better than ARM7 and ARM9

Cortex™
Intelligent Processors by ARM®

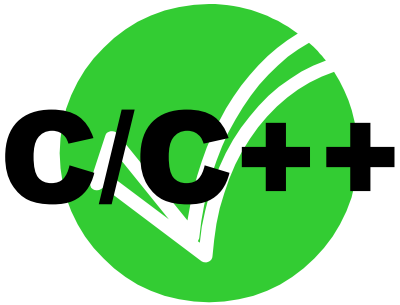
Cortex M3 core: Benefits

- **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: 85% faster
    - PID (process control) main loop: 217% faster
    - Multiply-intensive code: 294% faster
    - Divide-intensive code: 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

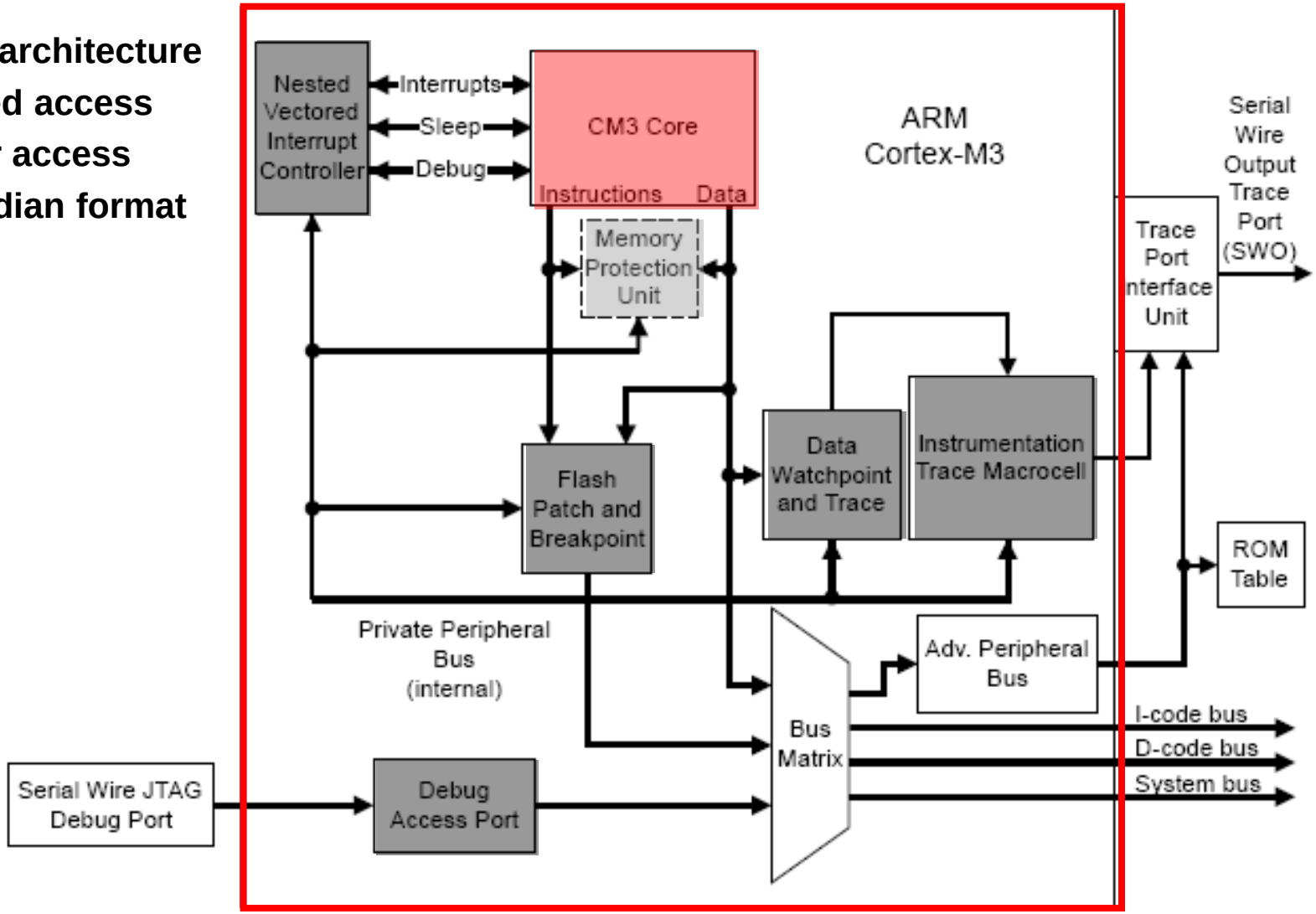
Cortex M3 core: no assembly!

- **Cortex-M3 has complete HW 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 core: architecture

- Harvard architecture
- Privileged access and user access
- Little-endian format



Cortex M3 System Timer (SysTick)

→ 24-bit clear-on-write, decrementing, wrap-on-zero counter

Could be set as:

- RTOS tick timer which fires at a programmable rate and invokes a SysTick routine
- High-speed alarm timer using the system clock
- A variable rate alarm or signal timer
- A simple counter used to measure time to completion and time used
- An internal clock source control based on missing/meeting durations. The COUNTFLAG bit-field

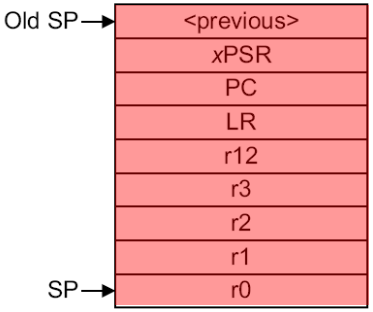
Cortex M3 System debug

- **JTAG**
  - ✓ Industry standard boundary scan for in-circuit testing
  - ✓ In-circuit flash programming
- **Parallel JTAG TAP**
  - ✓ Allows access to chip JTAG for boundary scan or Cortex-M3 JTAG for debug support
- **Serial Wire Debug / Serial Wire Trace (SWD/SWT)**
  - ✓ New technology to provide debug access and control in two pins, with an optional pin for trace information

Note: Stellaris implements the possibility to disable the debug permanently

Cortex M3 Interrupts

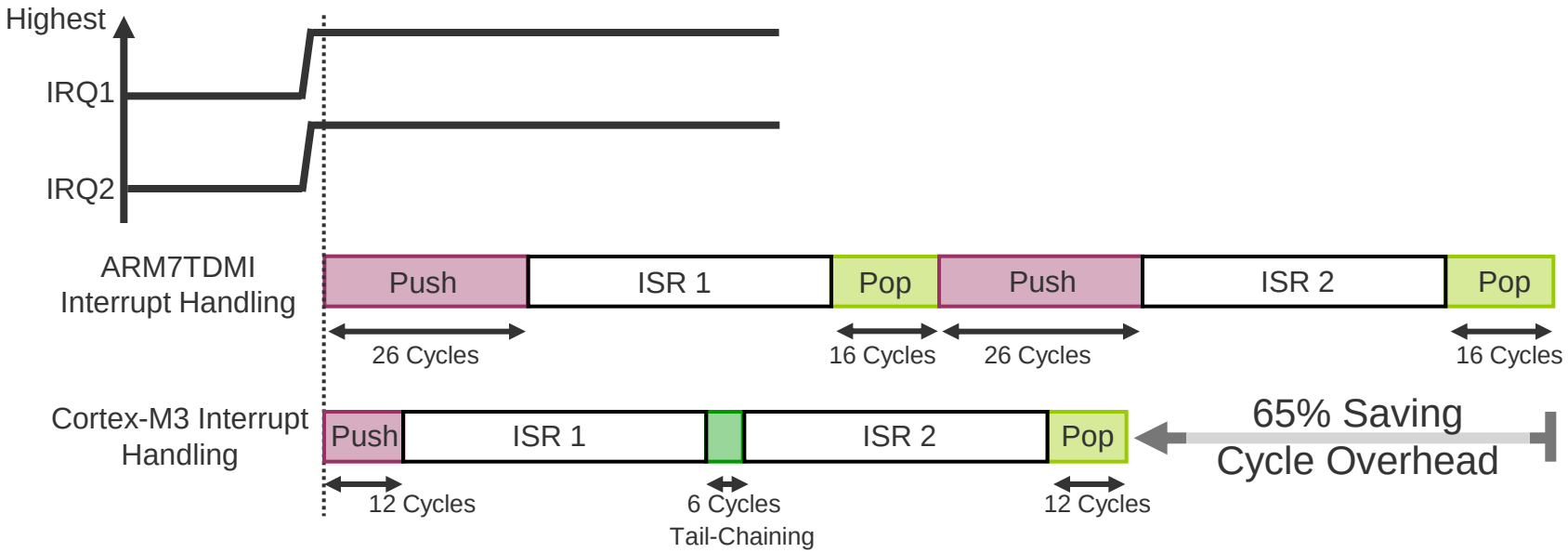
- *Nested Vectored Interrupt Controller* (NVIC) prioritize and handle all exceptions
- Automatic state saving and restoring
- Automatic reading of the vector table entry
- Pre-emptive/Nested interrupts implementation



- 10 Cortex M3 core exceptions types with programmable priority except Reset, NMI, Hard fault (highest priorities)
- Up to 53 peripherals interrupts (GPIOs, PWMs, ADCs, .)
- Tail-chaining support
- Priority grouping
- Process and Main (Handler) stacks
- only fully access the NVIC from privileged mode

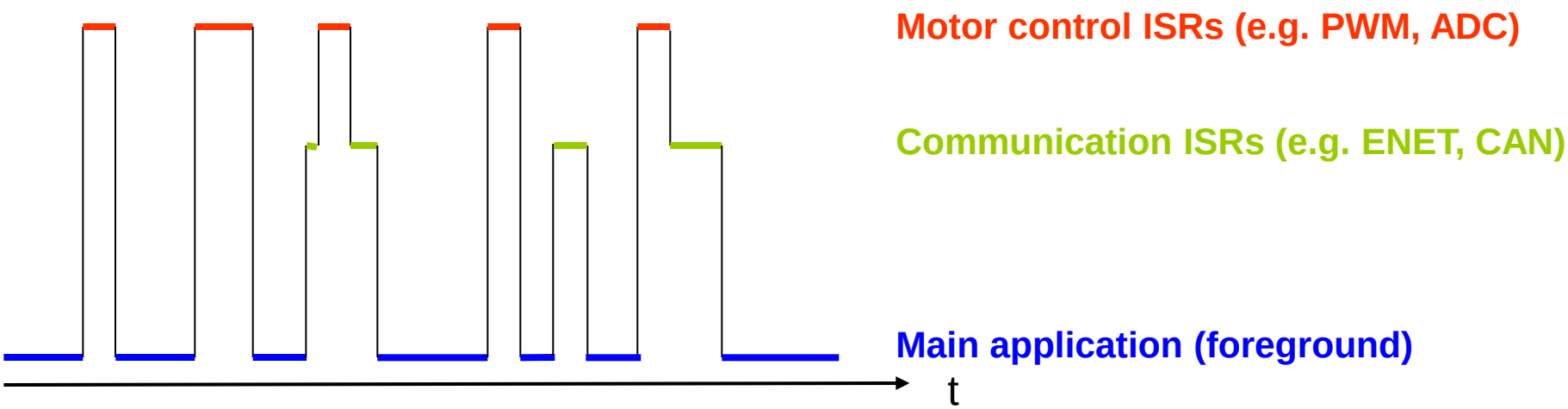
Note: Clear the interrupt source at the beginning of the interrupt handler

Cortex M3 interrupts: Tail Chaining



ARM7TDMI	Cortex-M3
▪ 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	▪ 12 cycles from IRQ1 to ISR1 (Interruptible/Continual LSM) ▪ 6 cycles from ISR1 exit to ISR2 entry ▪ 12 cycles to return from ISR2

Cortex M3 Interrupts: tasks&priorities



- **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

Cortex M3 Memory Protection Unit

→ **Silicon vendor option**

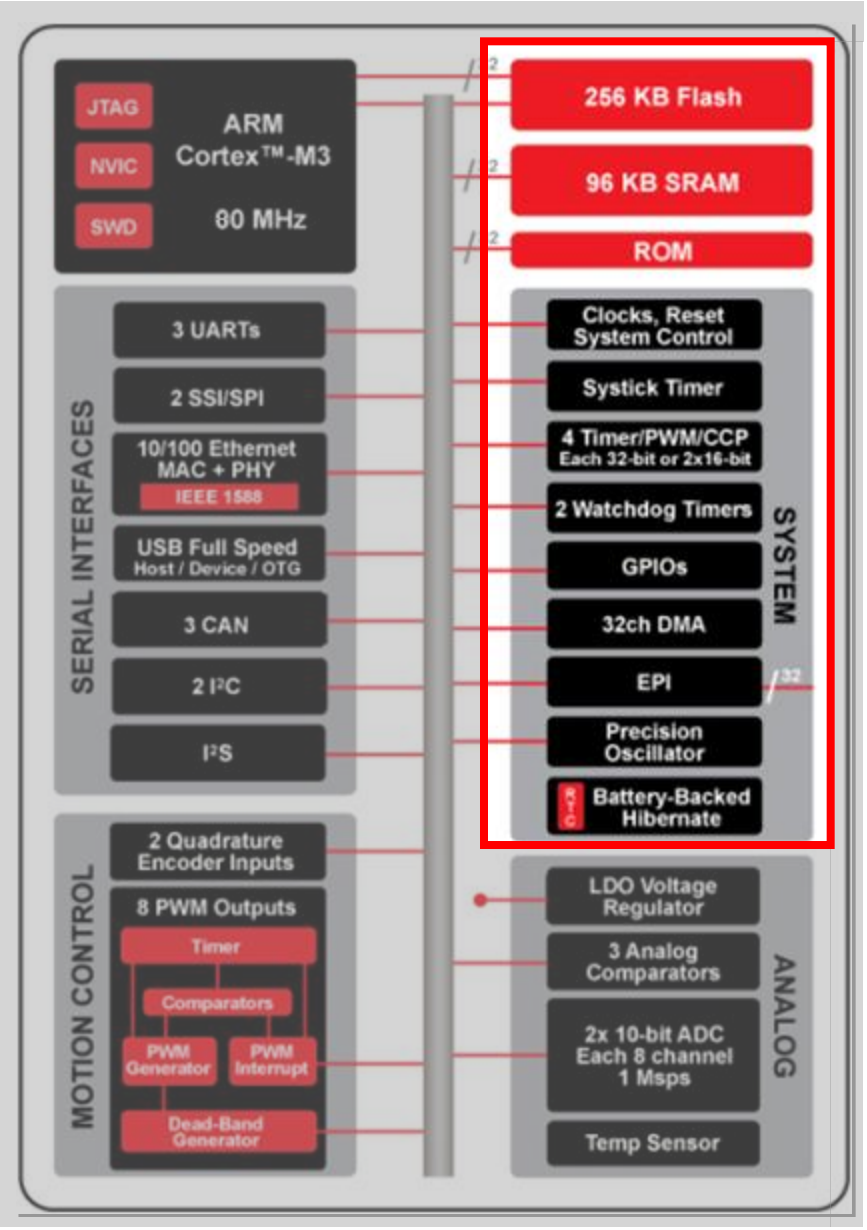
Benefits:

- Enforce privilege rules
- Separate processes
- Enforce access rules.

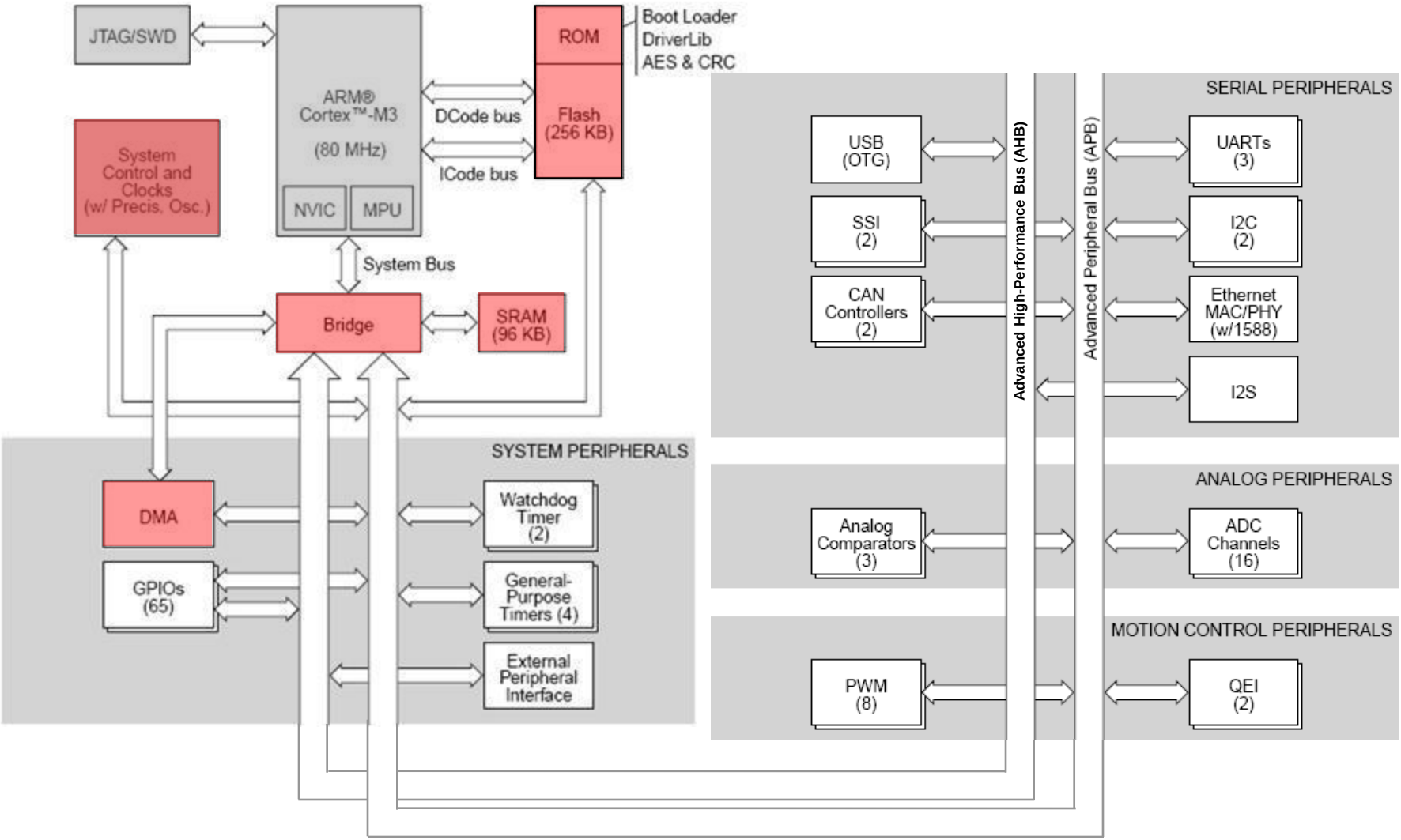
Features

- 8 Protection regions (no access, R/W, Read only) from 32B to 4GB range
- Access permissions (privileged/user)
- Overlapping protection regions with region priority
- MPU mismatches and permission violations invoke MemManage fault handler

Stellaris system



Stellaris® architecture



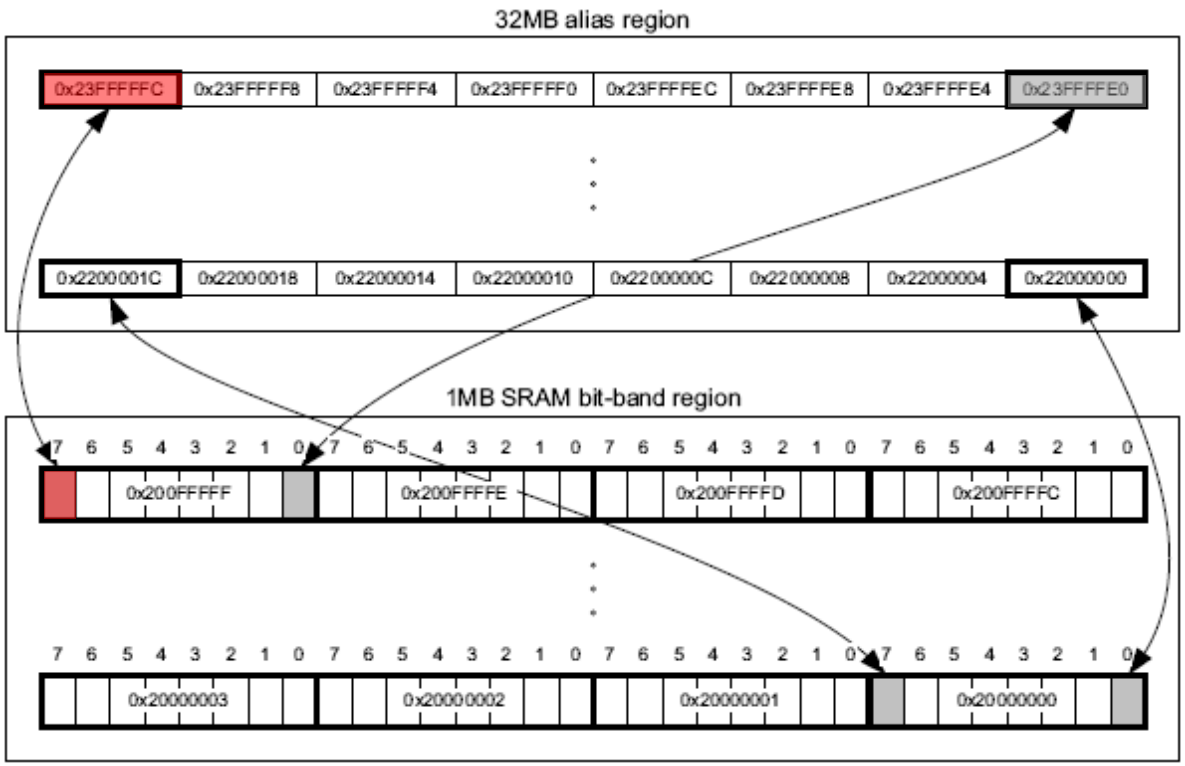
Stellaris® Memory map

0x0000 0000	Internal FLASH
0x0100 0000	On-chip ROM
	Boot Loader, DriverLib, AES+CRC software
0x2000 0000	Bit-banded on-chip SRAM
0x2200 0000	Bit-band alias of 0x2000.0000
0x4000.0000	Peripherals
0x4200 0000	Bit-band alias of 0x4000.0000
0x6000 0000	EPI0 mapped peripheral and RAM
0xE000 0000	Private peripherals bus
	Instrumentation Trace Macrocell (ITM)
	Data Watchpoint and Trace (DWT)
	Flash Patch and Breakpoint (FPB)
	Nested Vectored Interrupt Controller (NVIC)
	Trace Port Interface Unit (TPIU)

Bit-band

- Bit_word_addr = bit_band_base + (byte_offset x 32) + (bit_number 4)
- Example:

  The alias word at 0x23FFFFE0 maps to bit [0] of the bit-band byte at 0x200FFFFF: 0x23FFFFE0 = 0x22000000 + (0xFFFFF*32) + 0*4.



Stellaris® Internal memories

- **FLASH**
  - ✓ Flash memory protection
    - 2-KB Flash memory block granularity
    - “No protection”, “read only”, “execute only” options
    - Possibility to make protection permanent (sensitive data protection)
  - ✓ Flash programming
    - 1-KB blocks erase in once
    - 32-bit word can be programmed individually
    - The write buffer allows continuous 32 words programming

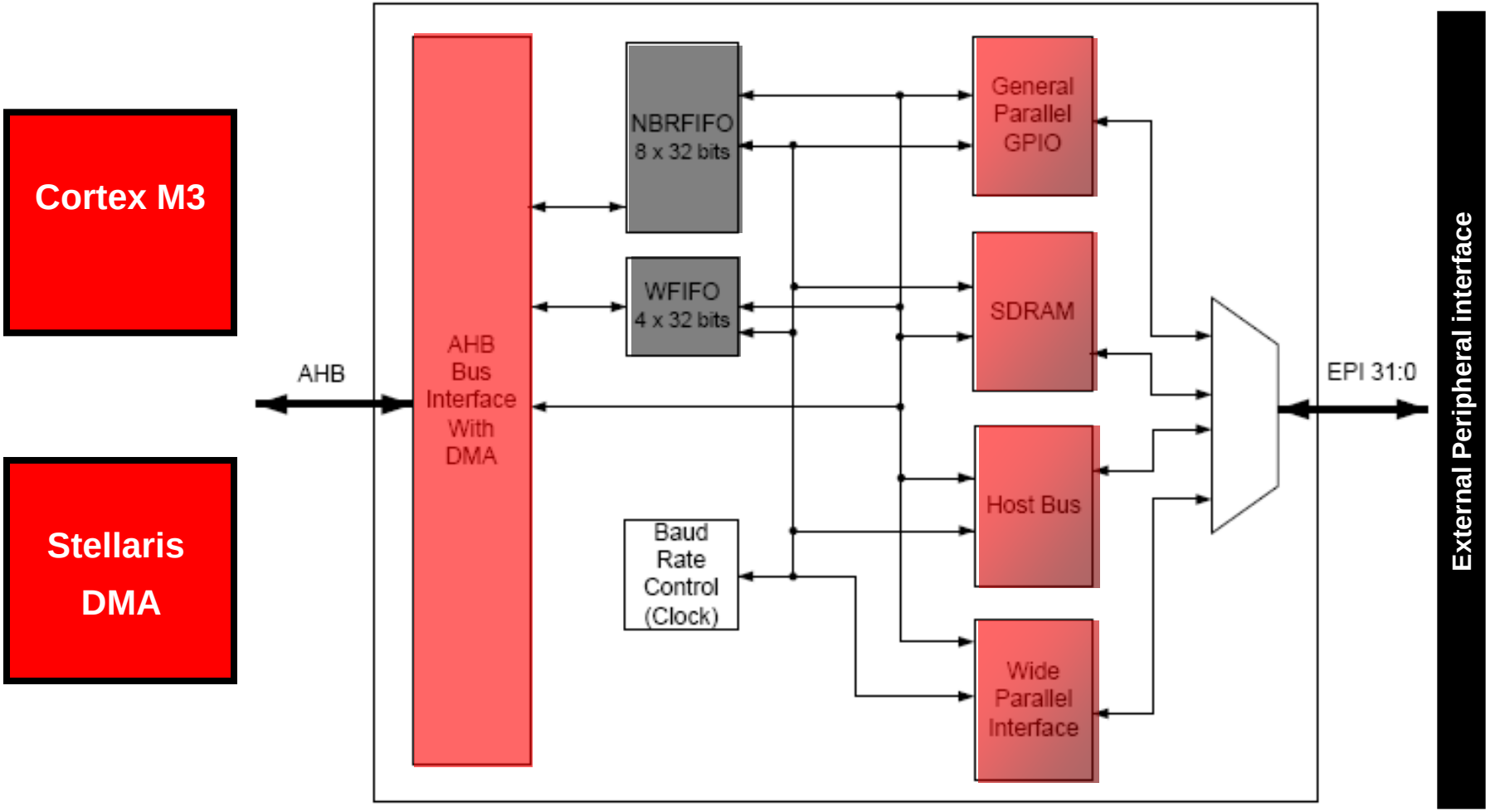
- **SRAM**
  - ✓ Bit-band capability

- **ROM**
  - ✓ Stellaris® Boot Loader and vector table
  - ✓ Stellaris® Peripheral Driver Library (DriverLib) release for product-specific peripherals and interfaces
  - ✓ Advanced Encryption Standard (AES) cryptography tables
  - ✓ Cyclic Redundancy Check (CRC) error detection functionality

Stellaris® External Peripheral Interface (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)

Stellaris® EPI block diagram



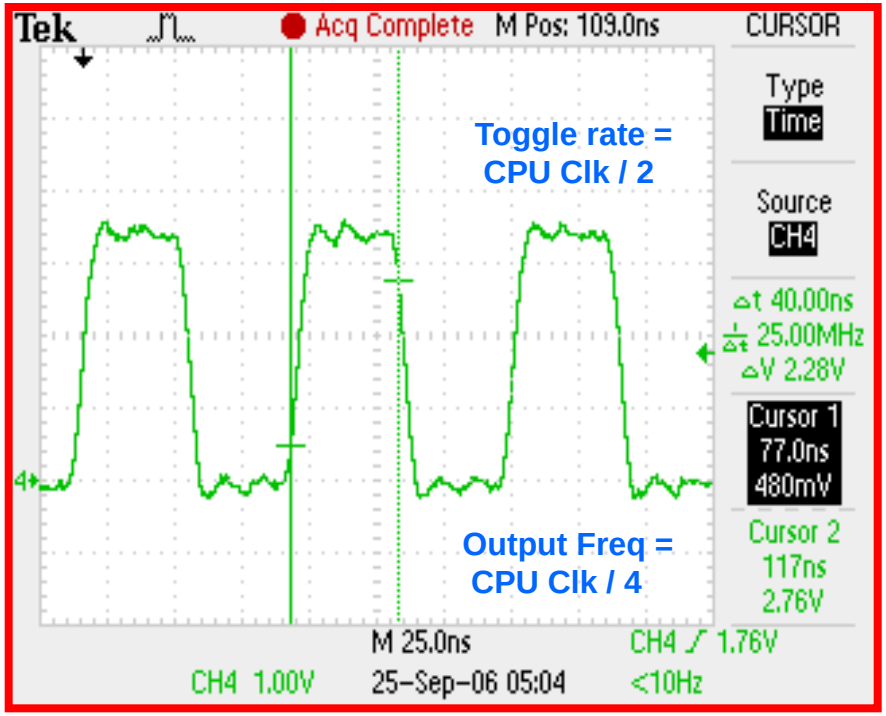
Stellaris® DMA

- **32-channel configurable µDMA controller**
- **Dedicated channels for supported peripherals**
  - ✓ One channel each for receive and transmit path for bidirectional peripherals
  - ✓ Multiple data sizes, Two levels of priority, Maskable device requests
- **Interrupt on transfer completion with a separate interrupt per channel**
- **Support for multiple transfer modes:**
  - ✓ Basic, for simple transfer scenarios
  - ✓ Ping-pong, for continuous data flow to/from peripherals
  - ✓ Scatter-gather, from a programmable list of arbitrary transfers initiated from a single request
- **DMA requests**
  - ✓ UART, Timer, USB, Ethernet, ADC, SSI, External Peripherals I/F

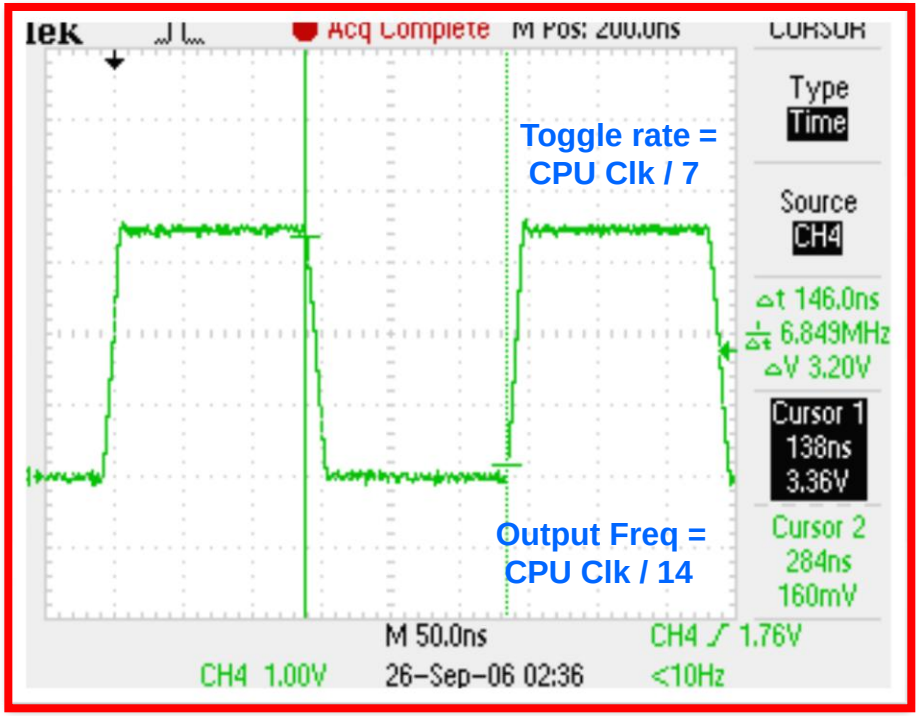
Stellaris® GPIOs

- **Programmable pad configuration through GPIO module**
- **Any GPIO can be full featured external interrupt**
- **Bit addressable pins, atomic operation**
- **Fast output toggling: Toggle rate up to ½ the CPU clock**
- **5-V-tolerant input/outputs**
- **Up to 61 GPIO available on a single device**
- **Programmable Drive Strength**
  - ✓ 2, 4, 8 mA or 8 mA with slew rate control
- **Programmable weak pull-up, pull-down**
- **Open drain enables**
- **Digital input enables**

Stellaris® GPIOs - Turn on



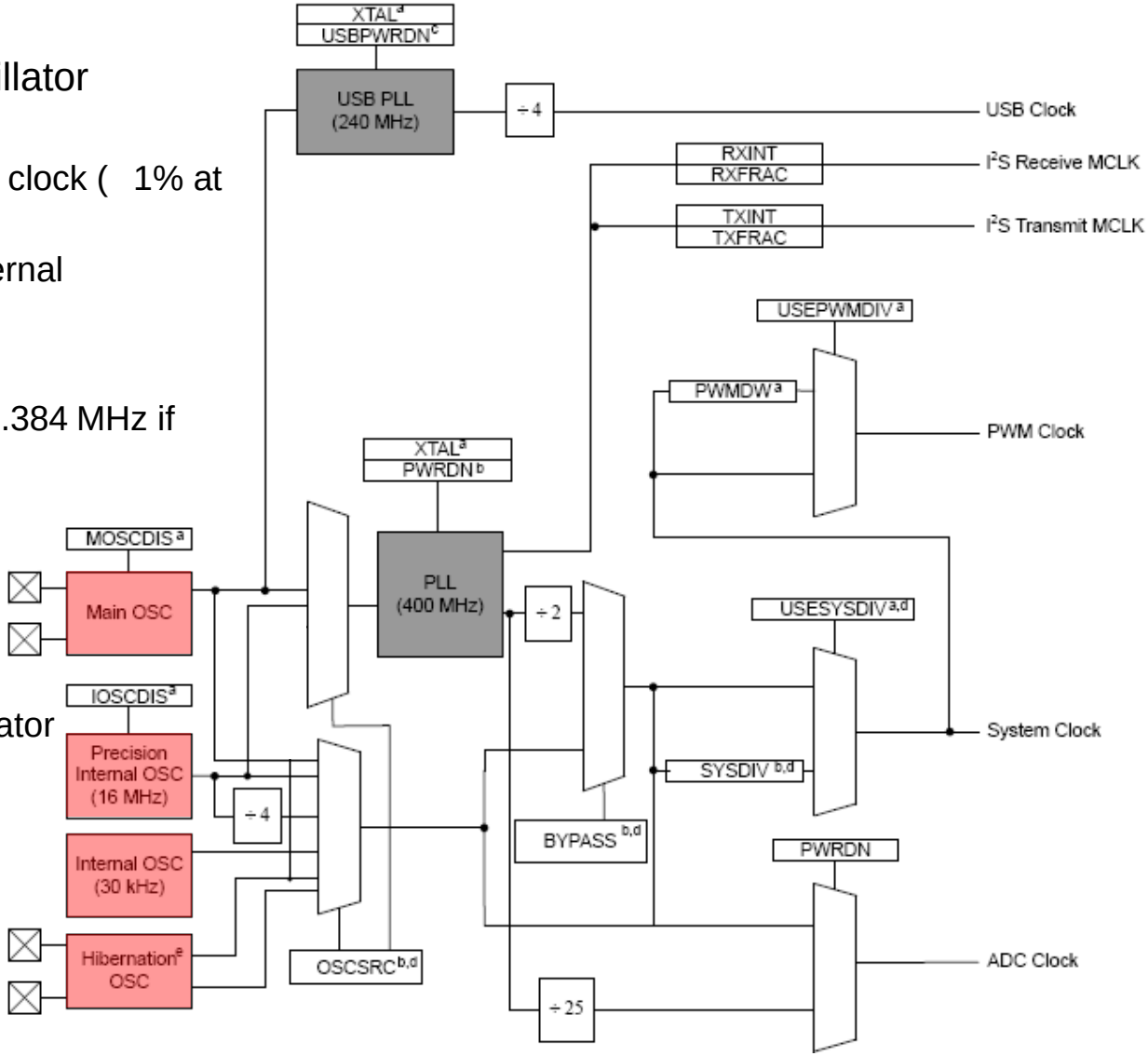
Stellaris MCUs



Arm 7

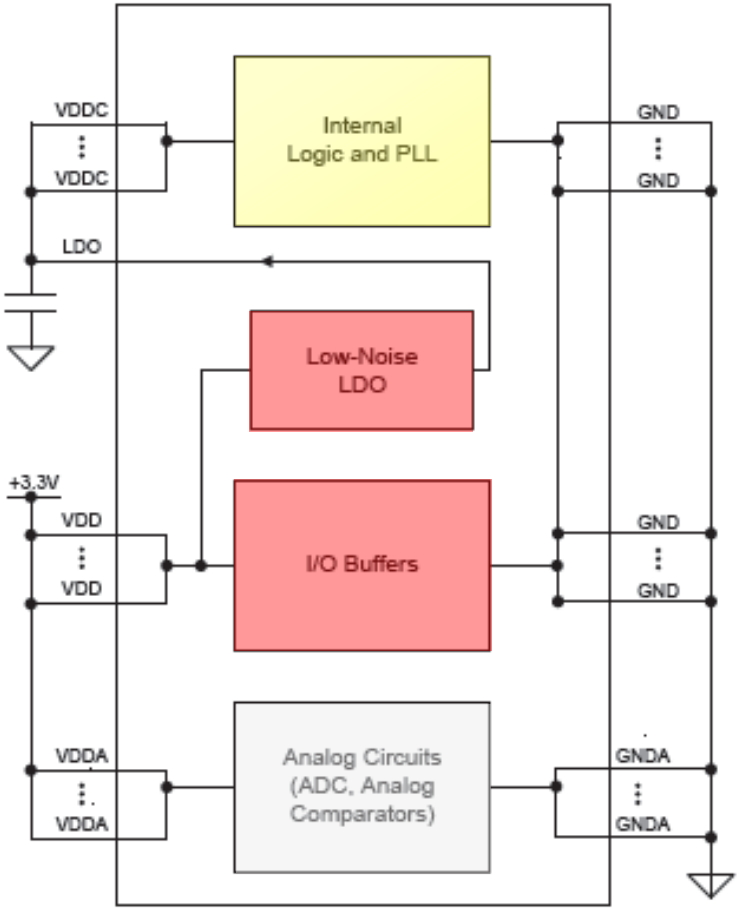
Stellaris® Clock architecture

- The precision internal oscillator (PIOSC)
  - ✓ On-chip precision 16MHz clock (1% at room temperature)
  - ✓ Does not require any external components
- Main OSC
  - ✓ 3.579545 MHz through16.384 MHz if PLL used
- Internal OSC
  - ✓ 30KHz (30%)
- Hibernation Oscillator
  - ✓ 4.194304MHz crystal
  - ✓ 32.768kHz external oscillator



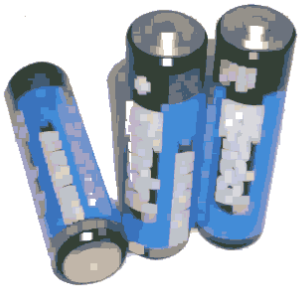
Stellaris® Power architecture

- **3 power domains**
  - ✓VDDC
  - ✓VDD
  - ✓VDDA



Stellaris® Power consumption

- **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

Stellaris® Watchdog Timers

- **Allows software to regain control when there is a system failure caused by software error or an external device**
- **Generates an interrupt or system reset on time-out**
- **Configuration can be locked (key) to prevent inadvertent changes**
- **User-enabled stalling for software debugging**
- **Some Stellaris MCUs feature two Watchdog Timer Modules, where :**
  - ✓ One module is clocked by the system clock (Watchdog Timer 0)
  - ✓ One module is clocked by the PIOSC (Watchdog Timer 1).

Stellaris® Timer Modules

- **Up to 4 timer modules**
  - ✓ Each configurable as single 32-bit timer, or two 16-bit timers
- **32-bit Timer Modes**
  - ✓ 32-bit programmable one-shot timer with selectable clock
  - ✓ 32-bit programmable periodic timer with selectable clock
  - ✓ RTC using a 32.768-kHz input clock
  - ✓ Software controlled debug event stalling (except for the RTC function)
- **16-bit Timer Modes**
  - ✓ 16-bit general-purpose timer with 8-bit prescaler and input clock selection
  - ✓ General-purpose free running timer modes: one-shot or periodic
  - ✓ Selectable debug event stalling
- **16-bit Capture Modes (CCP)**
  - ✓ Input edge count capture
  - ✓ Input edge time capture
- **PWM mode**
  - ✓ Simple PWM mode with programmable output negation

Stellaris® ARM® Cortex™-M3 based MCUs

1 Introduction

2 Stellaris solutions overview

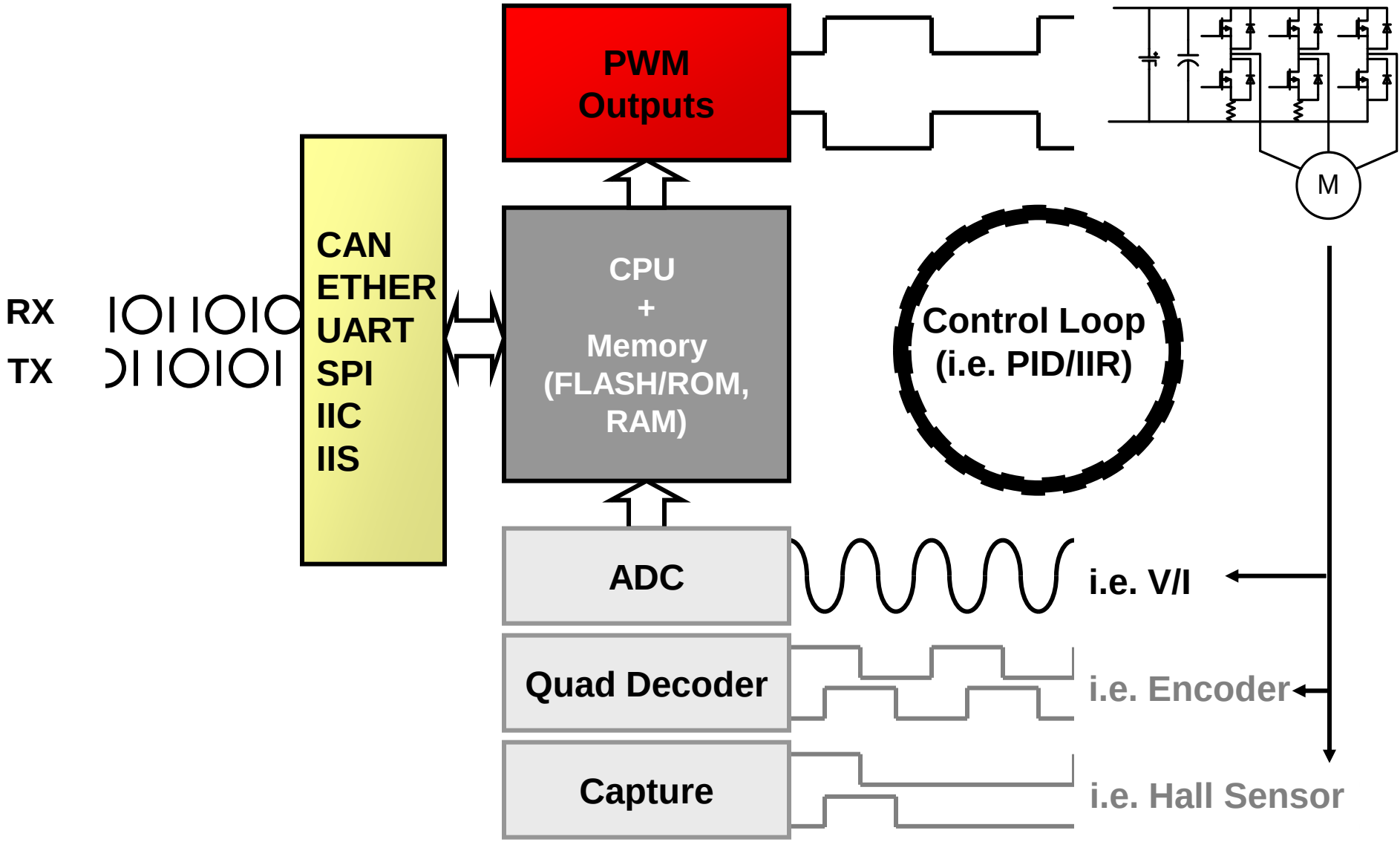
3 Platform overview & architecture

4 Motion control

5 Communication features

6 Stellarisware & Tools

Digital Motor Control concept



Stellaris® ADCs

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

0	1	0	2	0	3	0	4

ADC Channel Sequencer

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

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

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® analog comparators

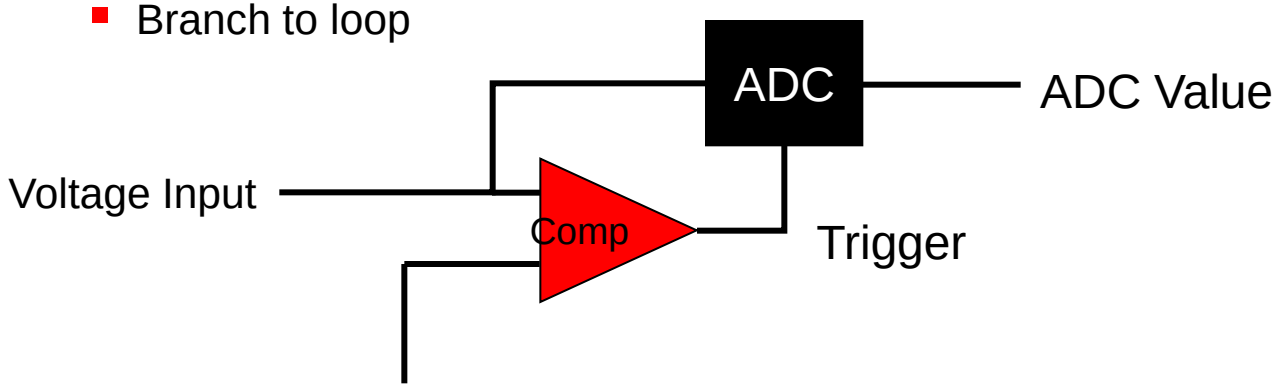
Situation: monitor a voltage for a drop below a threshold value

- **Without Analog Comparator**

  Poll the ADC
  - Loop: Read ADC value
  - If below threshold, call handler
  - Wait
  - Branch to loop

- **With Analog Comparator**

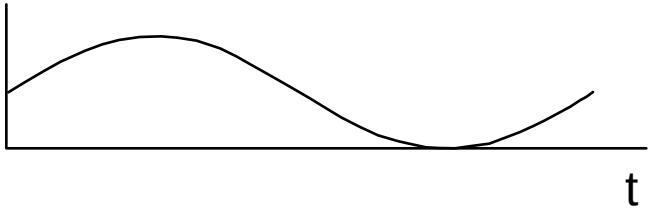
  Use the comparator to trigger the ADC when needed, then interrupt the CPU when value read



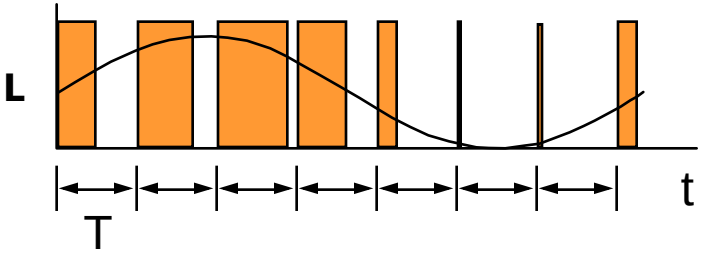
Benefits
- No software / CPU overhead
- Especially useful for infrequent out of range events

What is Pulse Width Modulation?

- **PWM is a scheme to represent a signal as a sequence of pulses**
  - ✓ fixed carrier frequency
  - ✓ fixed pulse amplitude
  - ✓ pulse width proportional to instantaneous signal amplitude
  - ✓ PWM energy ⇒ original signal energy



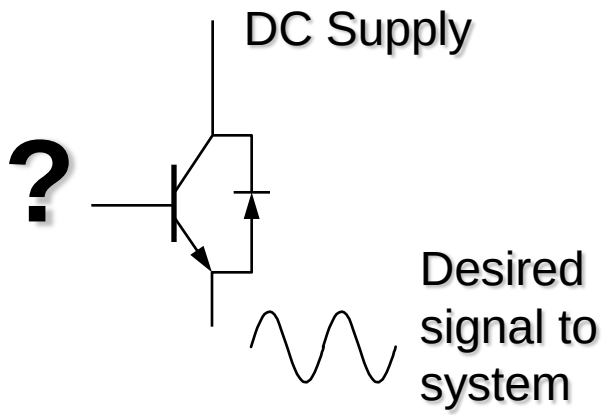
Original Signal



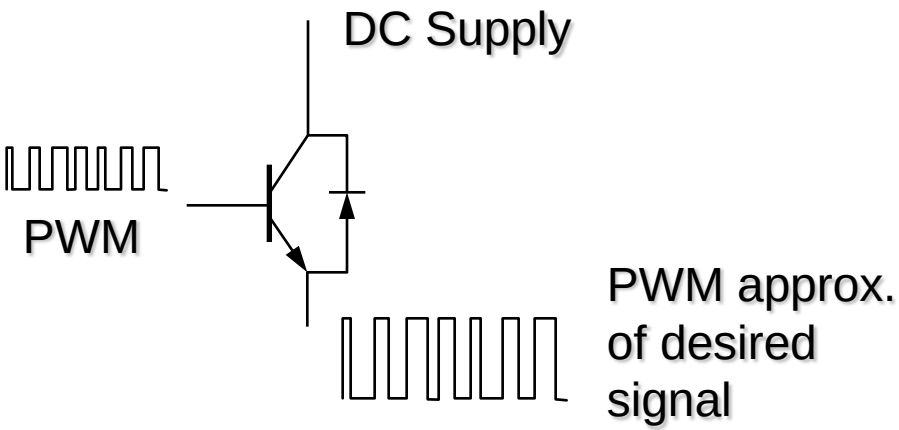
PWM representation

PWM with Power Switching Devices

- **Desired output currents or voltages are known**
- **Power switching devices are transistors**
  - ✓ Difficult to control in proportional region
  - ✓ Easy to control in saturated region
- **PWM is a digital signal ⇒ easy for uC to output**

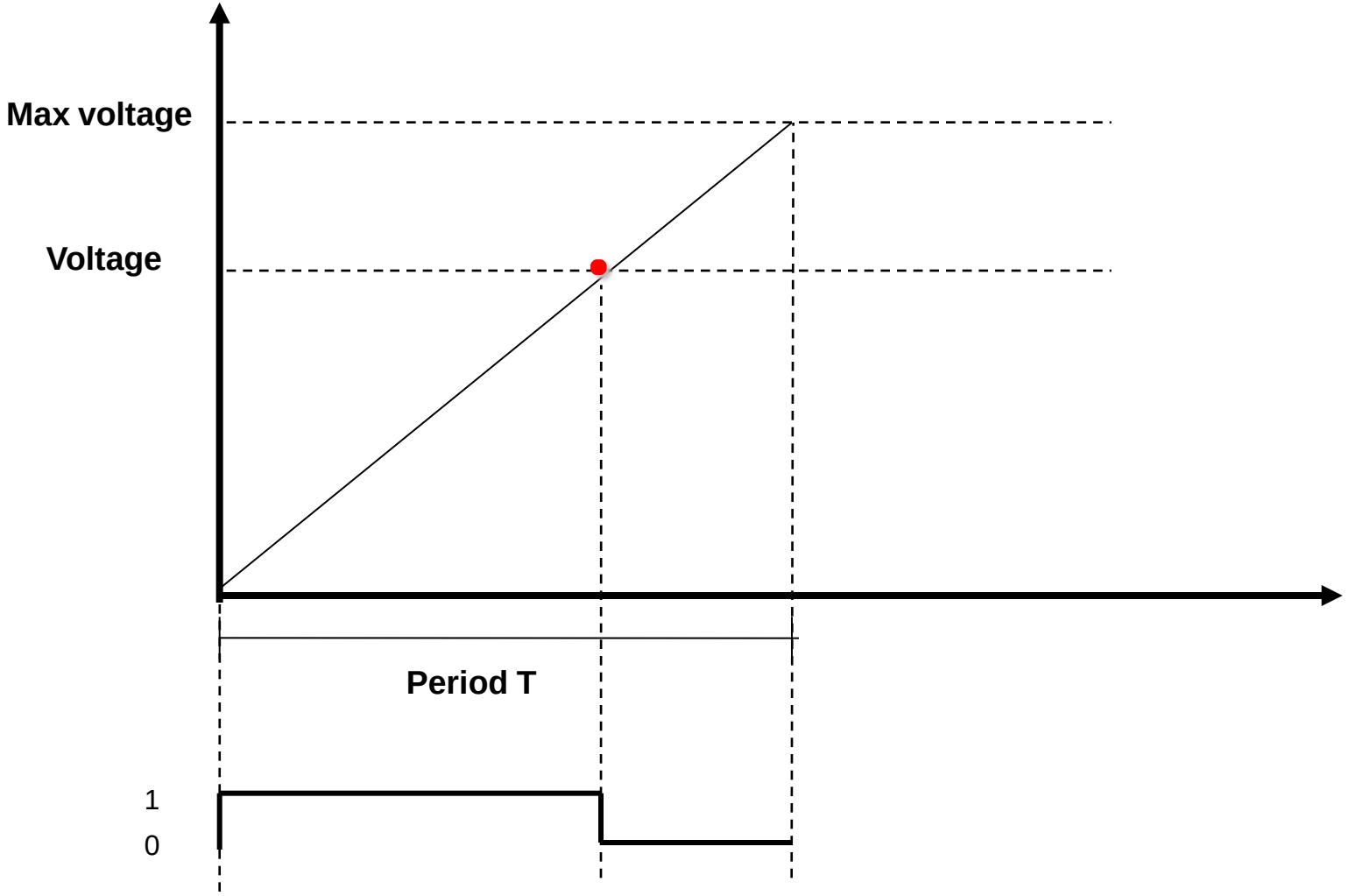


Unknown Gate Signal

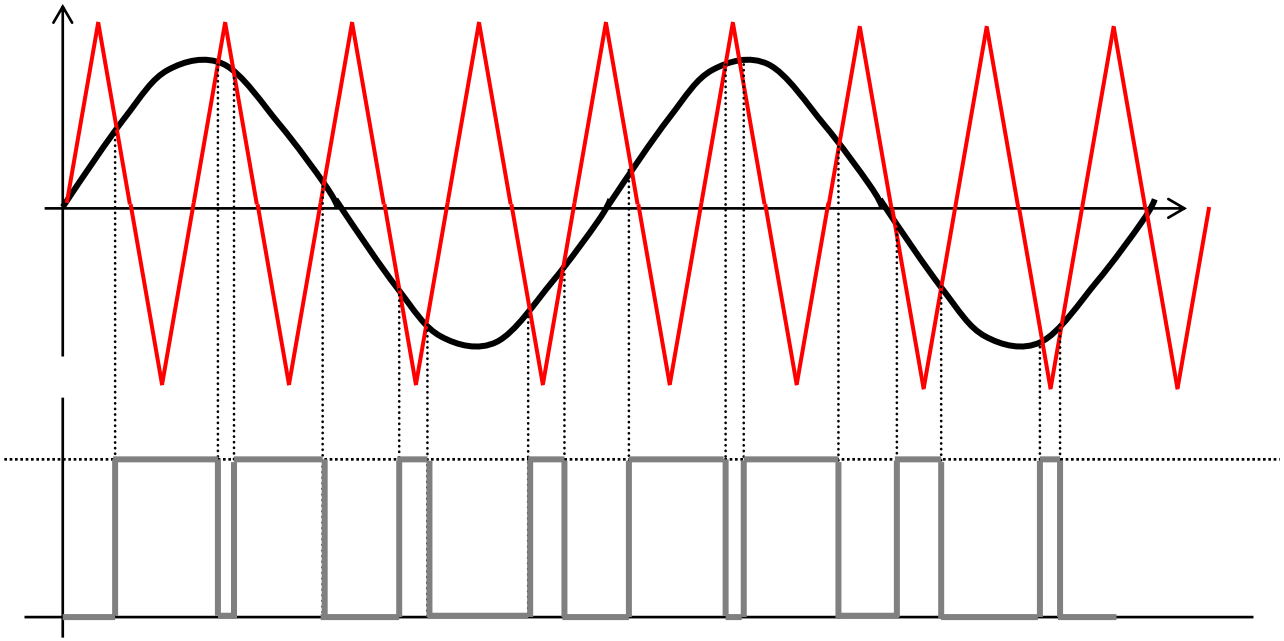


Gate Signal Known with PWM

PWM Signal Generation

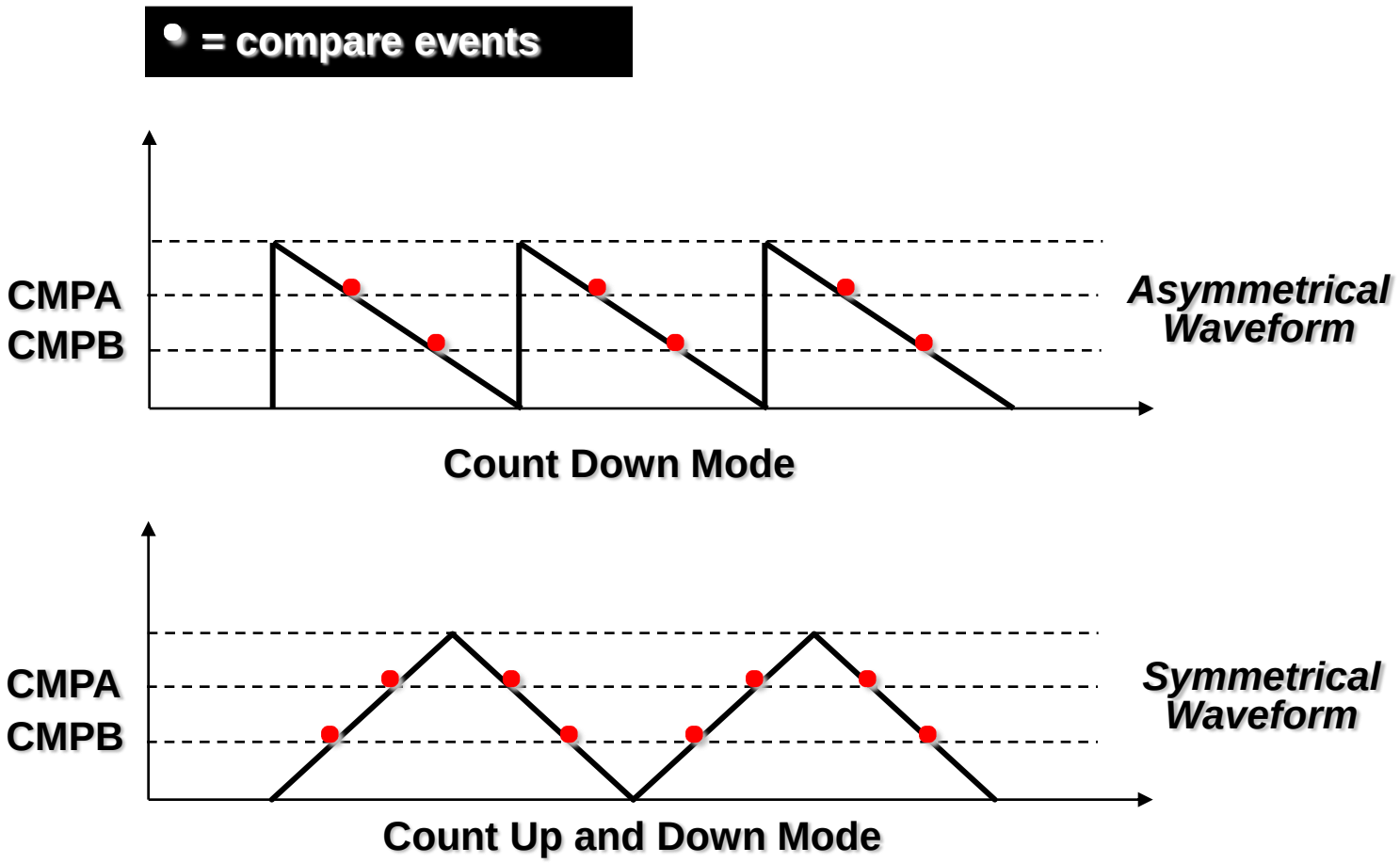


PWM Signal Generation



- **Traditional way: comparing three-phase sinusoidal waveforms with a triangular carrier**
- **Van = V.sin(ωt) (Van phase-neutral voltage)**

Stellaris® PWM Compare Events



Stellaris® Pulse Width Modulation

- **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

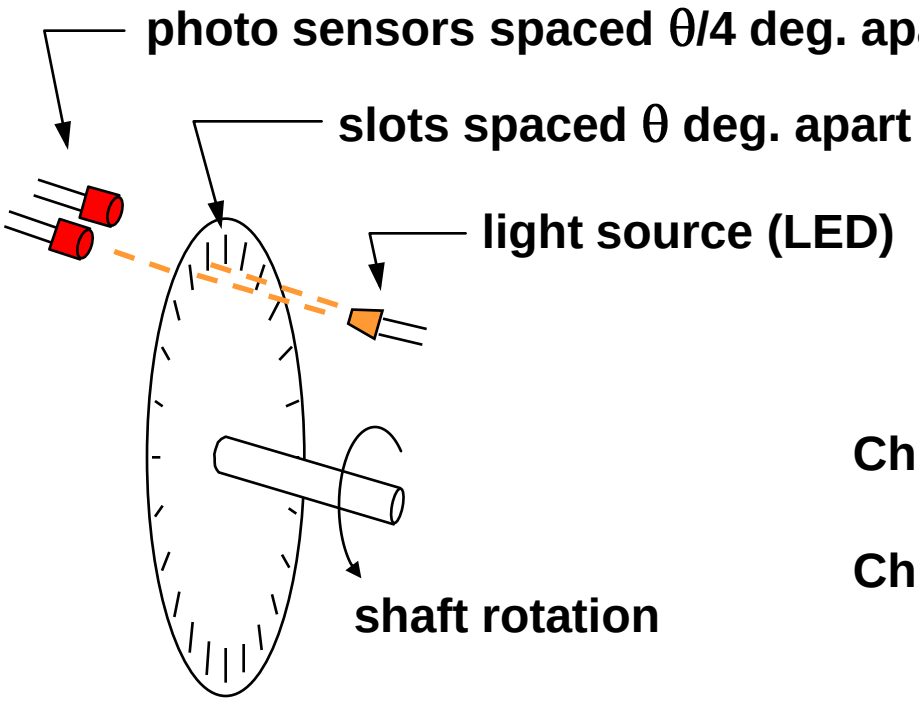
Stellaris® Fault-condition

- **A fault condition is one in which the controller must be signaled to stop normal PWM function and then sets the outputs to a safe state.**

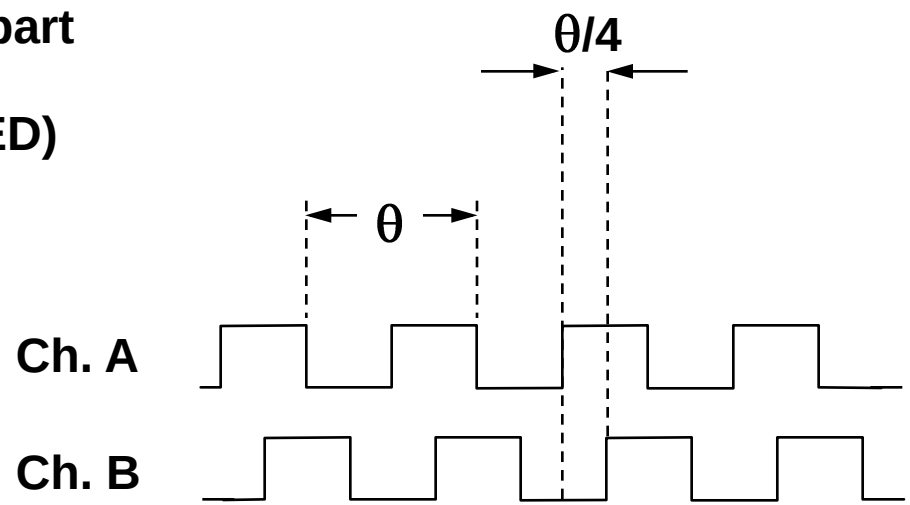
- **There are basic situations where this becomes necessary:**
  - ✓ The controller is stalled and cannot perform the necessary computation in the time required for motion control (Stall signal from internal debugger)
  - ✓ An external error or event is detected (FAULTn pin)
  - ✓ Trigger of an ADC digital comparator

Stellaris® Quadrature Encoder

- Also known as a 2-channel incremental encoder
- Converts linear displacement into a pulse signal
- Can track the position, direction of rotation, and speed of a motor
- A third channel, or index, signal, can be used to reset the position counter.



Incremental Optical Encoder



Quadrature Output from Photo Sensors

Stellaris® Stellaris QEI

- **Includes up to two quadrature encoder interface (QEI) modules**
- **Position integrator that tracks the encoder position**
- **Programmable noise filter on the inputs**
- **Velocity capture using built-in timer**
- **Interrupt generation on:**
  - ✓ Index pulse
  - ✓ Velocity-timer expiration
  - ✓ Direction change
  - ✓ Quadrature error detection

Stellaris® ARM® Cortex™-M3 based MCUs

1 Introduction

2 Stellaris solutions overview

3 Platform overview & architecture

4 Motion control

5 Communication features

6 Stellarisware & Tools

Stellaris® Ethernet MAC + PHY

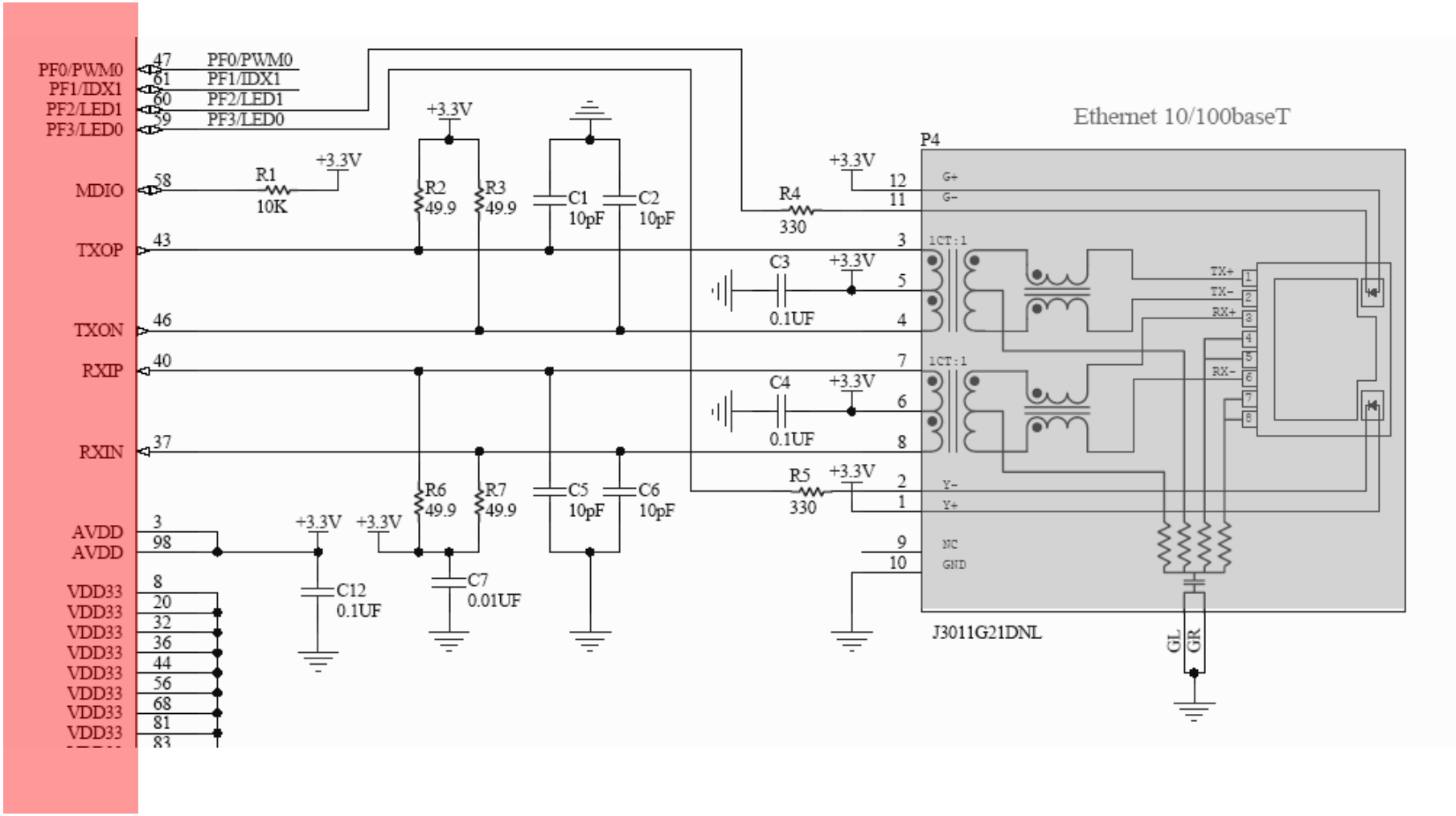
MAC + PHY module

- **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:** K-LM3S6965 Ethernet, EK-LM3S8962 Ethernet + CAN
- **Communication stack availibility**
  - ✓ Ported open-source lwIP stack and µIP stacks
  - ✓ Web Server applications (from TI and FreeRTOS.org™)
  - ✓ Third parties

Stellaris® Ethernet Hardware design



Communications Stacks for Stellaris®

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
RTA REAL TIME AUTOMATION INC.	EtherNet/IP™ protocol stacks
freeRTOS	FreeRTOS.org Open-Source µIP Embedded web server
µIP	**Open source** TCP/IP stack for small footprint embedded systems
lwIP	**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

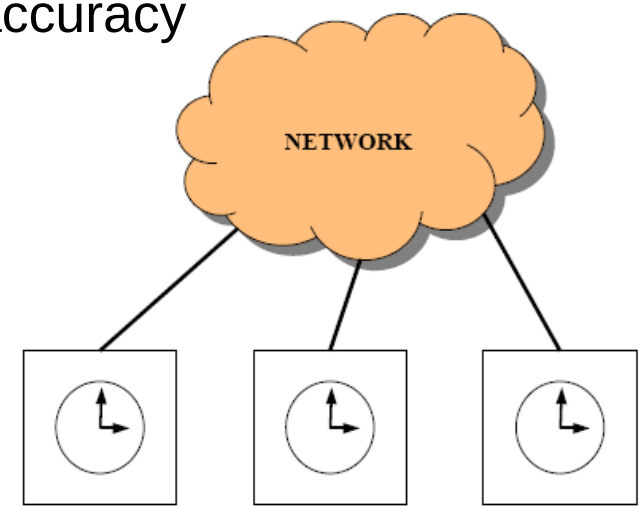
Communications Stacks for Stellaris®

TPV	Product	Stack	ARP	AutoIP	BOOTP	BSD	DHCP	DNS	FTP	HTTP	ICMP	IGMP	IKE	IP	IPSec	NAT	POP3	PPP	PTP	RARP	RIP	RTP	SLIP	SMTP	SNMP	SNTP	SSL	TCP	Telnet	TFTP	UDP	802.11
CMX Systems	CMX-MicroNET	TCP/IP	●		●						●	●		●				●					●					●			●	
CMX Systems	CMX Add Ons	Networking SW Options					●	●	●	●							●	●						●	●	●				●		●
ExpressLogic	NetX	TCP/IP	●								●	●		●						●								●			●	
ExpressLogic	NetX Add Ons	Networking SW Options		●		●	●	●	●	●						●	●	●						●	●	●			●	●		
Interniche	NicheLITE	TCP/IP	●		●		●	●			●			●														●		●	●	
Interniche	NicheStack	TCP/IP	●		●	●	●	●	●		●	●		●														●	●	●	●	
Interniche	Interniche Add Ons	Networking SW Options					●	●	●	●			●		●	●	●	●			●	●		●	●	●	●		●			
Micriμm	μC/UDP-IP	UDP/IP	●			●					●			●																	●	
Micriμm	μC/TCP-IP	TCP/IP	●			●					●			●														●			●	
Micriμm	Micriμm Add Ons	Networking SW Options					●	●	●	●							●							●		●			●	●		
SEVENSTAX	SEVENSTAX TCP/IP	TCP/IP									●															●		●			●	
SEVENSTAX	SEVENSTAX Add Ons	Networking SW Options	●		●		●	●		●				●			●	●						●								
SEGGER	embOS/IP	TCP/IP	●		●		●	●	●	●	●	●		●			●			●				●			●	●	●		●	
uIP	open source	TCP/IP	●									●		●														●			●	
lwIP	open source	TCP/IP	●				●					●		●				●										●			●	

∇ List is subject to change

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

Stellaris® Controller Area Network

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.

Stellaris® UART

- **The Stellaris® Family features up to 3 UARTs**
- **Provides fully programmable, 16C550-type serial interface characteristics.**
- **Each UART has the following features:**
  - ✓ Separate transmit and receive FIFOs
  - ✓ Programmable FIFO length
  - ✓ FIFO trigger levels of 1/8, 1/4, 1/2, 3/4, and 7/8
  - ✓ Programmable baud-rate generator allowing rates up to speeds up to 12.5 Mbps
  - ✓ Standard asynchronous communication bits for start, stop and parity
  - ✓ False start bit detection
  - ✓ Line-break generation and detection
- **Fully programmable serial interface characteristics:**
  - ✓ 5, 6, 7, or 8 data bits
  - ✓ Even, odd, stick, or no-parity bit generation/detection
  - ✓ 1 or 2 stop bit generation
- **IrDA serial-IR (SIR) encoder/decoder providing:**
  - ✓ Programmable use of IrDA Serial InfraRed (SIR) or UART input/output
  - ✓ Support of IrDA SIR encoder/decoder functions for data rates up to 115.2 Kbps half-duplex
  - ✓ Support of normal 3/16 and low-power (1.41-2.23 μs) bit durations
  - ✓ Programmable internal clock generator enabling division of reference clock by 1 to 256 for low-power mode bit duration
- **ISO 7816 Support (SmartCard communication) on tempest devices**

Stellaris® I2S

- **Four modes: Stereo, Mono, Compact 16-bit Stereo and Compact 8-Bit Stereo.**
- **I2S, Left-justification, and Right-justification**
- **Sample size from 8 to 32 bits**
- **Mono and Stereo support**
- **8-, 16-, and 32-bit FIFO I/F for packing memory**
- **Independent transmit and receive 8-entry FIFOs**
- **Configurable FIFO-level interrupt and µDMA requests**
- **Independent transmit and receive MCLK direction control**
- **Transmit and receive internal MCLK sources**
- **Independent transmit and receive control for serial clock and word select**
- **MCLK and SCLK can be independently set to master or slave**
- **Configurable transmit zero or last sample when FIFO empty**
- **Efficient transfers using Micro Direct Memory Access Controller (µDMA)**
  - ✓ Separate channels for transmit and receive
  - ✓ Burst requests
  - ✓ Channel requests asserted when FIFO contains required amount of data

Stellaris® I2C

- **Master or Slave**
- **Simultaneous operation as both a master and a slave**
- **There are a total of four I2C modes:**
  - ✓ Master Transmit, Master Receive
  - ✓ Slave Transmit, Slave Receive
- **Standard (100 Kbps) and Fast (400 Kbps) speed support**
- **Master and slave interrupts support**
  - ✓ I2C master generates interrupts when a transmit or receive operation completes (or aborts).
  - ✓ I2C slave generates interrupts when data has been sent or requested by a master.

Stellaris® SSI

- Master or slave
- Master runs up to sys_clk/2 (25Mb @ 50MHz)
- Slave runs up to sys_clk/12 (4.1667Mb @ 50MHz)
- Programmable clock bit rate and prescale
- Separate transmit and receive FIFOs, 16 bits wide, 8 locations deep
- Programmable interface operation for usual synchronous serial interfaces
- Programmable data frame size from 4 to 16 bits
- Internal loopback test mode for diagnostic/debug testing

Stellaris® ARM® Cortex™-M3 based MCUs

1 Introduction

2 Stellaris solutions overview

3 Platform overview & architecture

4 Motion control

5 Communication features

6 Stellarisware & Tools

Stellaris® Evaluation Kits

- **Start in 10 minutes or less**
- **Evaluation board packages includes:**
  - ✓ Cables
  - ✓ A choice of evaluation tools suites for popular development tools
  - ✓ Documentation (QuickStart guide, User's guide, …)
  - ✓ StellarisWare® software
  - ✓ 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

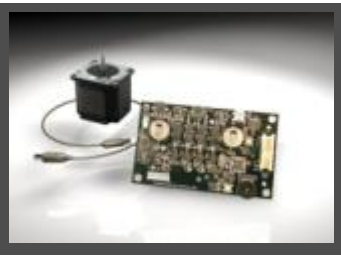
- **Function both as an evaluation platform and as a serial in-circuit debug interface for any Stellaris microcontroller-based target board**

Stellaris® RDK - Open-Tool Motor

AC Induction Motor Controller Design

Example applications:

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



Stepper Motor Controller Design

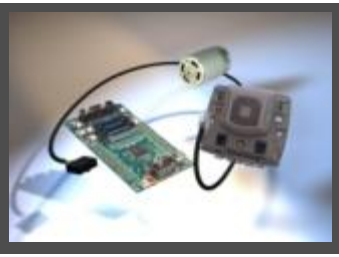
Example applications:

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

Brushless DC Motor Controller with CAN/Ethernet

Example applications:

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



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

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)

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)

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)

Tiny Footprint Serial-to-Ethernet Module

Example applications:

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

Stellaris® Development Kit

- **LM3S9B96 Full-Featured Development Board**
  - ✓ 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 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® 20-pin JTAG debug connector with input and output modes
- **1 GB MicroSD Card, 128 MB USB Flash Drive**
- **Ethernet cable, CAN ribbon cable, USB and JTAG cables**
- **CDs containing evaluation software tools, documentation, quickstart guide, StellarisWare® Graphics, USB, and Peripheral Driver Libraries, and source code**

Evaluation version software tools included in the kit: KEIL, IAR SYSTEMS, code_red, CODESOURCERY, **Code Composer Studio**

Included

EPI – Breakout Board

EPI – 8MB SDRAM Board

Available now

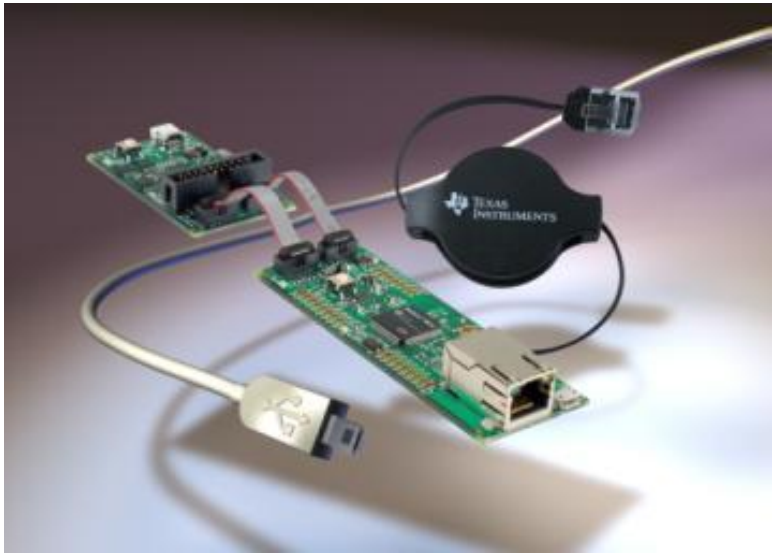
EPI – Flash/SRAM/LCD Board

Available now

EPI – FPGA and Camera Board

Stellaris Evaluation Kits: Getting started

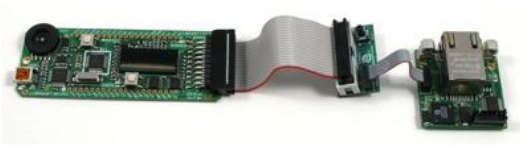
- **Obtain EVM**
- **Plug in via USB cable**
- **From the CD install:**
  - ✓USB Drivers
  - ✓Development IDE
  - ✓Software example
- **Begin developing!**



IDE Stellaris® MCUs: Third Party

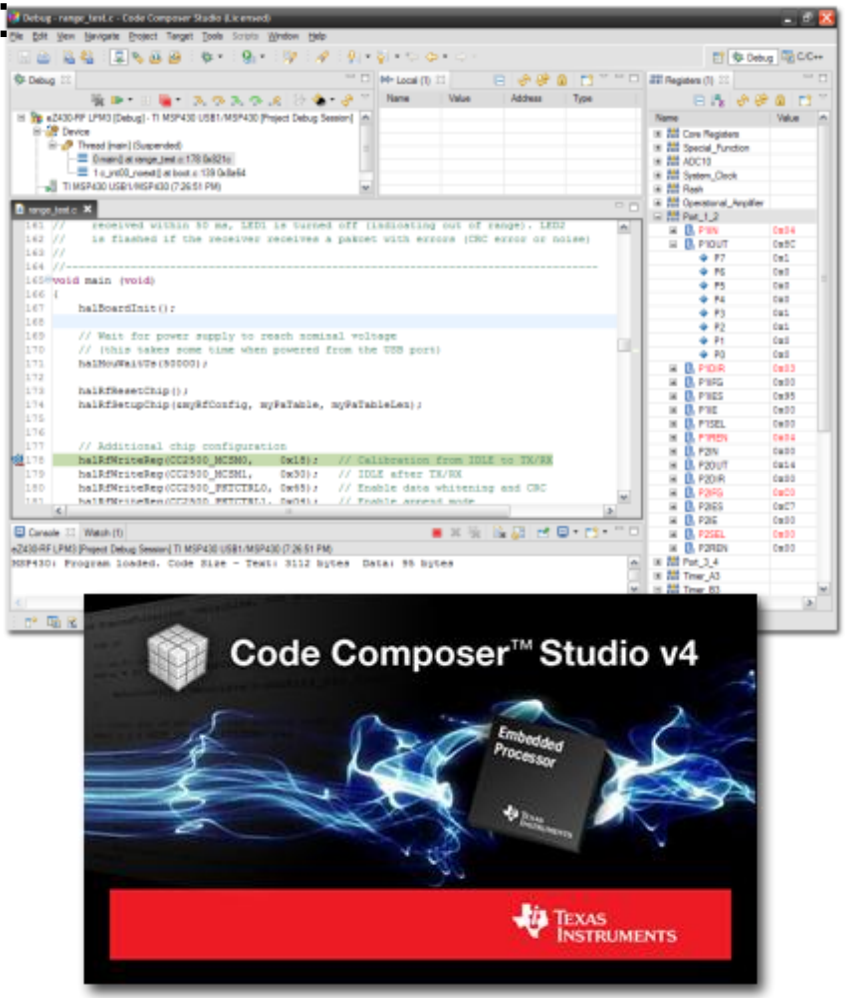
	CodeSourcery	IAR Systems	ARM KEIL An ARM® Company	code_red	Code Composer Studio
Eval Kit License	30-day full function. Upgradeable	32KB address-limited. Upgradeable	32KB address-limited. Upgradeable	90-day full function. Upgradeable	Full function. Onboard emulation limited
Compiler	GNU C/C++	IAR C/C++	RealView C/C++	GNU C/C++	TI C/C++
Debugger / IDE	gdb / Eclipse	C-SPY / Embedded Workbench	µVision	code_probe / Eclipse-based tool suite	CCS/Eclipse-based suite
Full Upgrade	199 USD personal edition / 3000 USD full support	2700 USD	MDK-Basic (256 KB) = €2000 (2895 USD)	999 USD (upgrade to run on customer platform)	495 USD
JTAG Debugger		J-Link, ~299 USD	U-Link, ~199 USD	Red Probe, 150 USD	XDS510 /XDS560

Note: In addition to its original use as an evaluation kit, each Stellaris evaluation kit has the built-in capability for use as a simple USB-to-20-pin JTAG debugger.



IDE Stellaris® MCUs: Code Composer *Studio* v4

- Easy to use, Eclipse based IDE: Compiler, linker, more
- Supports *all* Stellaris MCUs
- Enhancements since CCE v3:
  - Speed
  - Code size improvements
  - Auto-updating
- $495 for CCS v4 MCU Edition
- Board locked for Stellaris EVMs
- Identical look and feel as Code Composer *Essentials*



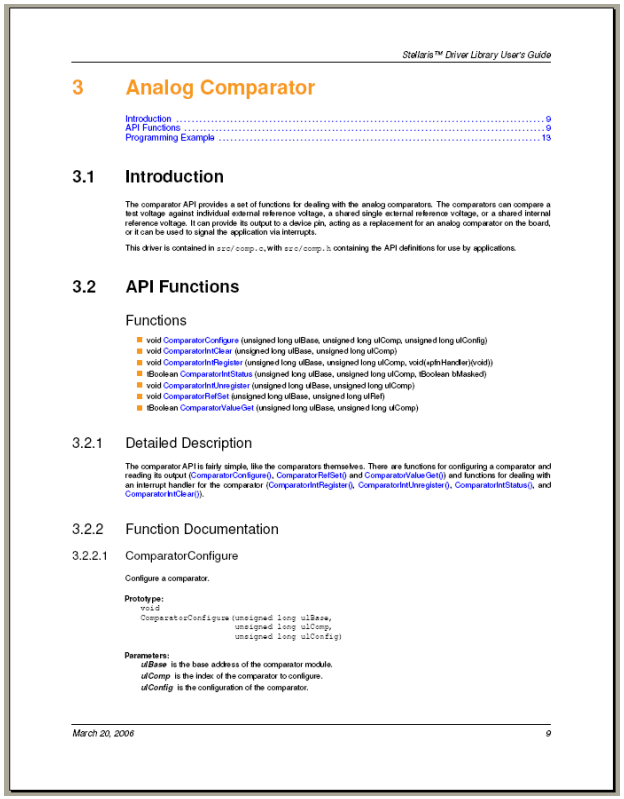
http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html

Stellarisware introduction

- **100% C/C++ progamming possibility even ISR and initialization**
- **Includes source code and royalty-free libraries for applications support**
- **Stellaris Software compiles on:**
  - ✓ ARM/Keil Microcontroller Development Toolkit for ARM
  - ✓ IAR Embedded Workbench
  - ✓ Code Red Technologies' RedSuite
  - ✓ Code Sourcery SourceryG++
  - ✓ Generic GNU development tools.
- **Key functional areas (Free license & royalty-free source code):**
  - ✓ Stellaris Peripheral Driver Library
  - ✓ Stellaris Graphics Library
  - ✓ Stellaris USB Library
  - ✓ Stellaris IEC 60730 Library
  - ✓ Code Samples
  - ✓ Stellaris In-System Programming Support

Stellaris® Peripheral Driver Lib

- 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
- DriverLib 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

StellarisWare® Peripheral Driver Lib

- **Collection of ‘c’ source and header files**
- **Base root directory structure with individual compiler specific library output directory.**
- **“.\StellarisWare” contains Hardware specific header files**
  - Include peripheral specific definition
  - Required ‘Type’ definitions
  - Macros
- **‘.\StellarisWare\driverlib’ contains**
  - ✓ ‘c’ source and header files peripheral specific functionality
  - ✓ Compiler specific project files for building the driver library ‘libraries’
  - ✓ ‘Compiler Specific’ output directories and files, ie the actual ‘library’ file used by each compiler.
    - C:\StellarisWare\driverlib\ewarm - IAR
    - C:\StellarisWare\driverlib\gcc - CodeRed
    - C:\StellarisWare\driverlib\rvmdk - Keil
    - C:\StellarisWare\driverlib\sourcerygxx - CodeSourcery

Peripheral Driver Lib: UART example

```
Int
main(void)
{
   // Set the clocking to run directly from the crystal.
   SysCtlClockSet(SYSCTL_SYSDIV_1 | SYSCTL_USE_OSC |
                  SYSCTL_OSC_MAIN | SYSCTL_XTAL_8MHZ);

   // Enable the peripherals used by this example.
   SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
   SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);

   // Enable processor interrupts.
   IntMasterEnable();

   // Set GPIO A0 and A1 as UART pins.
   GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);

   // Configure the UART for 115,200, 8-N-1 operation.
   UARTConfigSet(UART0_BASE, 115200, (UART_CONFIG_WLEN_8 |
                 UART_CONFIG_STOP_ONE |
                 UART_CONFIG_PAR_NONE));

   // Enable the UART interrupt.
   IntEnable(INT_UART0);
   UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT);

   // Loop forever echoing data through the UART.
   while(1)
   {
   }
}
```

```
void
UARTIntHandler(void)
{
   unsigned long ulStatus;

   // Get the interrrupt status.
   ulStatus = ROM_UARTIntStatus(UART0_BASE, true);

   // Clear the asserted interrupts.
   ROM_UARTIntClear(UART0_BASE, ulStatus);

   // Loop while there are characters in the receive FIFO.
   while(ROM_UARTCharsAvail(UART0_BASE))
   {
      // Read the next character from the UART and write it back to the UART.
      ROM_UARTCharNonBlockingPut(UART0_BASE,
                 ROM_UARTCharNonBlockingGet(UART0_BASE));
   }
}
```

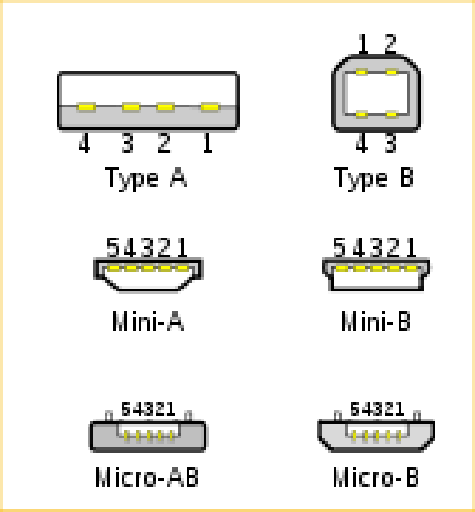
StellarisWare® Graphics Library

- Set of graphics primitives and widgets for use on Stellaris MCUs.
- Three subsequent layers of functionality:
  - ✓ Display Driver Layer
  - ✓ Graphics Primitives Layer
  - ✓ Widget Layer
  - ✓ Each API in each layer is directly callable
- Written entirely in C (except where not possible), self-contained, easy-to-understand, efficient.
- Compiles on ARM/Keil, IAR, Code Red, CCS and GNU tools.
- Computations that can be performed at compile time whenever possible.
- Graphics Primitives:
  - ✓ Point, Line, Rectangle, Circle, Font, Image, Context, Buffer
  - ✓ 134 Computer Modern predefined fonts available
  - ✓ Up to 24-bit color (~150 common colors conveniently referenced in GraphicsLib)
- Widgets:
  - ✓ Canvas, Checkbox, Container, Push Button, Radio Button, Slider, ListBox
- Special Utilities
  - ✓ *ftrasterize*: render your own font to be recognized by GraphicsLib
  - ✓ *lmi-button*: predefined button shape with shadow and 3-D
  - ✓ *pnmtoc*: Convert a NetPBM image file into a format recognized by GraphicsLib

StellarisWare® USB Library

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

- **Device Examples:**
  - ✓ HID Keyboard
  - ✓ HID Mouse
  - ✓ CDC Serial
  - ✓ Generic Bulk
  - ✓ Audio class
  - ✓ 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**



VID
Request for embedded USB products

FREE
Vendor ID/
Product ID
sharing
program

StellarisWare® IEC 60730 support

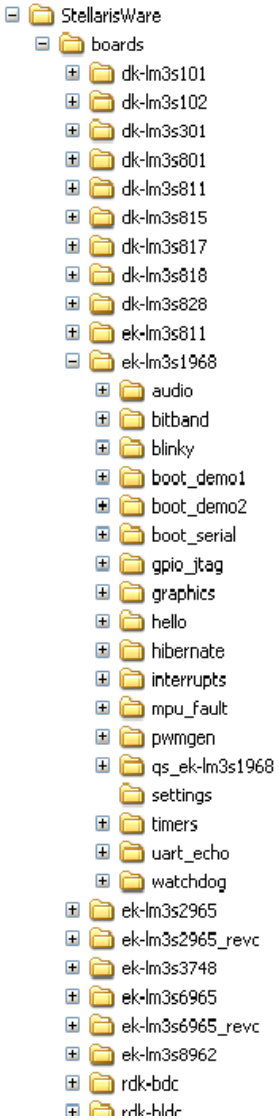
The International Electrotechnical Commission (IEC)

- 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

	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 seoncd 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	

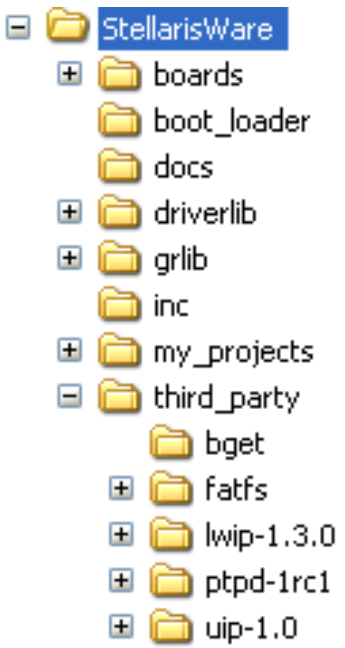
StellarisWare® code examples

- **'.\StellarisWare\boards 'contains all development, evaluation and reference design kit project**
- **Each kit has it's own directory**
- **There are multi-project workspace files for all available projects for all supported compilers**
- **Each individual project directory contains**
  - ✓ 'Readme' giving details of the project
  - ✓ Project files supported compilers
  - ✓ Project source code
  - ✓ Compiler specific 'library' output directories
    - C:\StellarisWare\boards\kit\project\ewarm – IAR
    - C:\StellarisWare\kit\project\gcc – CodeRed
    - C:\StellarisWare\kit\project\\rvmdk – Keil
    - C:\StellarisWare\kit\project\\sourcerygxx – CodeSourcery
- **All project that use the driver library reference**
  - ✓ Driver library 'library' file from 'C:\StellarisWare\driverlib\complier'
  - ✓ Driver Library header '*.h' file from 'C:\StellarisWare\driverlib'
- **Other related files**
  - ✓ 'c:\StellarisWare\inc' – Device specific header 'lm3sxxxx.h' files
  - ✓ 'c:\StellarisWare' – Peripheral hardware header 'hw_xxxx.h' files



StellarisWare® 3rd Party File Organization

- **'.\StellarisWare\third_party 'contains any required third party source code files and information**
- **Each 'third_party' directory contains all required information**
  - ✓ Documentation, source code files, etc
- **Example projects using any 'third_party' directly links the required source code '*.c' and header '*.h' from the relevant directory**



Available in the Stellaris Peripheral Driver Library (DriverLib) Download

On-chip SW Enhancements (ROM)

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**

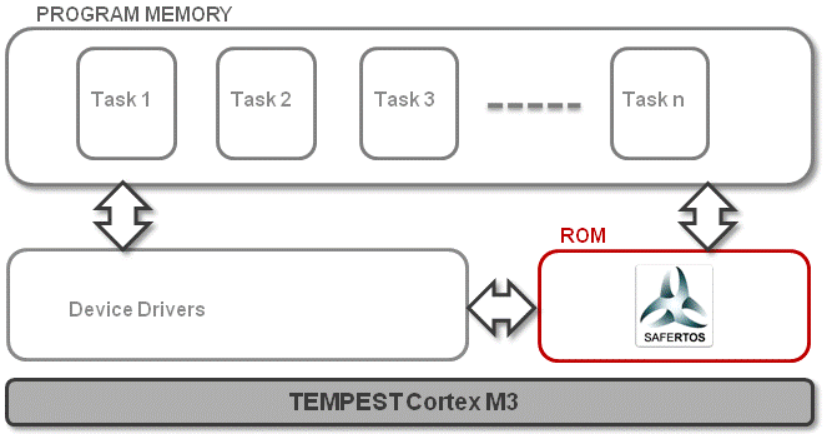
Other flash memory-saving options

- **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**

Stored in exclusive ROM on select Stellaris MCUs

On-chip SW Enhancements (ROM)

SAFERTOS for Tempest



- High-integrity RTOS in ROM
- RTOS value $65k free with Tempest LM3S9B96
- Can be used as a standard operating system OR as part of a high integrity application which requires certification to IEC61508 or FDA510(k)
- 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