Stellaris® ARM® Cortex™-M Microcontroller Solutions

StellarisWare® Drivers, Libraries

Simplicity	… in Adoption and Implementation.
Robustness	… in Design.
Efficiency	… in System Architecture.
Determinism	… in Control and Connectivity.
Flexibility	… in Production.

Introducing StellarisWare®

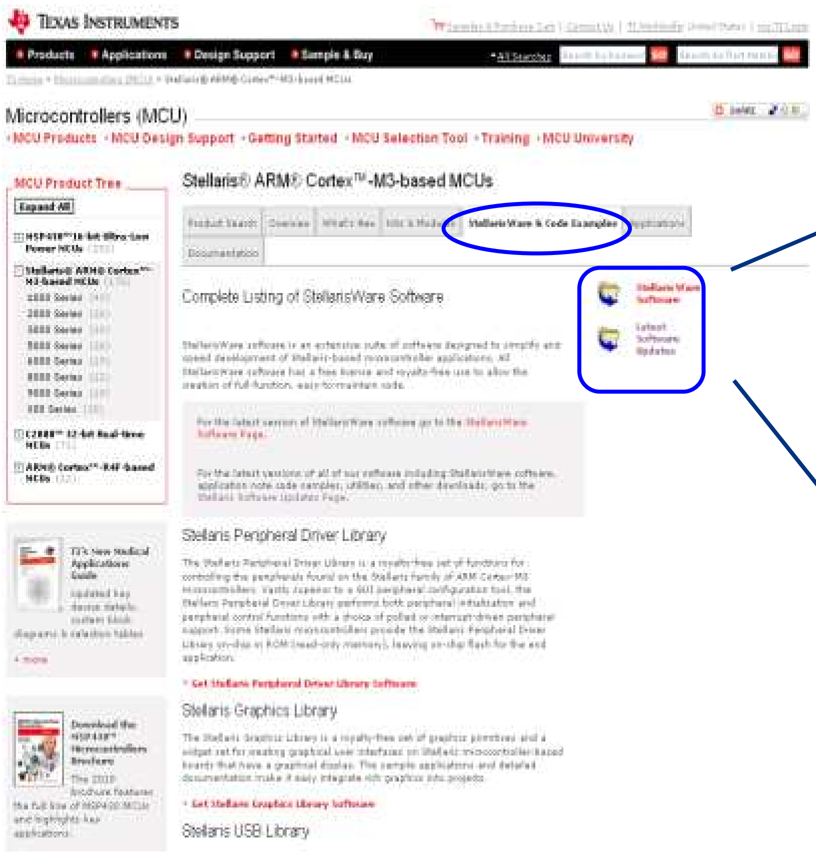
- StellarisWare software includes source-code and royalty-free libraries
- Keep all your programming in C/C++, including ISR's and startup code
- The key functional areas are:
  - Stellaris Peripheral Driver Library
  - Stellaris Graphics Library
  - Stellaris USB Library
  - Stellaris IEC 60730 Library
- Includes reference application software
- Stellaris In-System Programming support
- StellarisWare® Software is supported by all the most popular tools vendors
- **Robust**: StellarisWare is preprogrammed into ROM on most Stellaris MCUs

EK-LM3S8962 Firmware Development Package Setup
Stellaris® Microcontrollers
Texas Instruments
Version 5228
< Back
Next >
Cancel

Welcome to the Texas Instruments Stellaris® EK-LM3S8962 Firmware Development Package Setup Wizard

The Setup Wizard will install the EK-LM3S8962 Firmware Development Package on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Introducing StellarisWare



Microcontrollers (MCU)

- MCU Products · MCU Design Support · Getting Started · MCU Selection Tool · Training · MCU University

MCU Product Tree

Stellaris® ARM® Cortex™-M3-based MCUs

Complete Listing of StellarisWare Software

StellarisWare software is an extensive suite of software designed to simplify and speed development of Stellaris-based microcontroller applications. All StellarisWare software has a free license and royalty-free use to allow the creation of full-function, easy-to-maintain code.

For the latest version of StellarisWare software go to the StellarisWare Software Page.

For the latest versions of all of our software including StellarisWare software, application note code samples, utilities, and other downloads, go to the Stellaris Software Updates Page.

Stellaris Peripheral Driver Library

The Stellaris Peripheral Driver Library is a royalty-free set of functions for controlling the peripherals found on the Stellaris family of ARM Cortex-M3 microcontrollers. Vastly superior to a GUI peripheral configuration tool, the Stellaris Peripheral Driver Library performs both peripheral initialization and peripheral control functions with a choice of polled or interrupt-driven peripheral support. Some Stellaris microcontrollers provide the Stellaris Peripheral Driver Library on-chip in ROM (read-only memory), leaving on-chip flash for the end application.

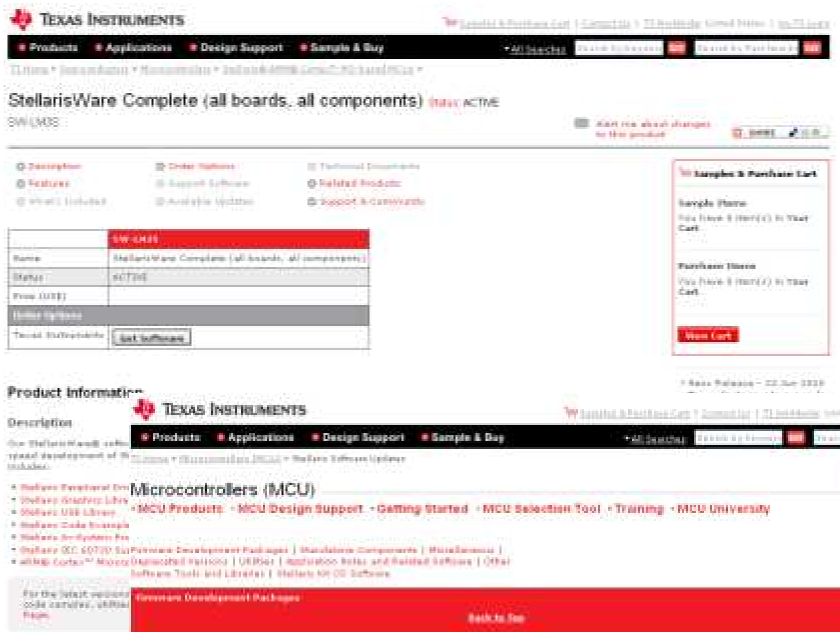
- Get Stellaris Peripheral Driver Library Software

Stellaris Graphics Library

The Stellaris Graphics Library is a royalty-free set of graphics primitives and a widget set for creating graphical user interfaces on Stellaris microcontroller-based boards that have a graphical display. The sample applications and detailed documentation make it easy integrate rich graphics into projects.

- Get Stellaris Graphics Library Software

Stellaris USB Library



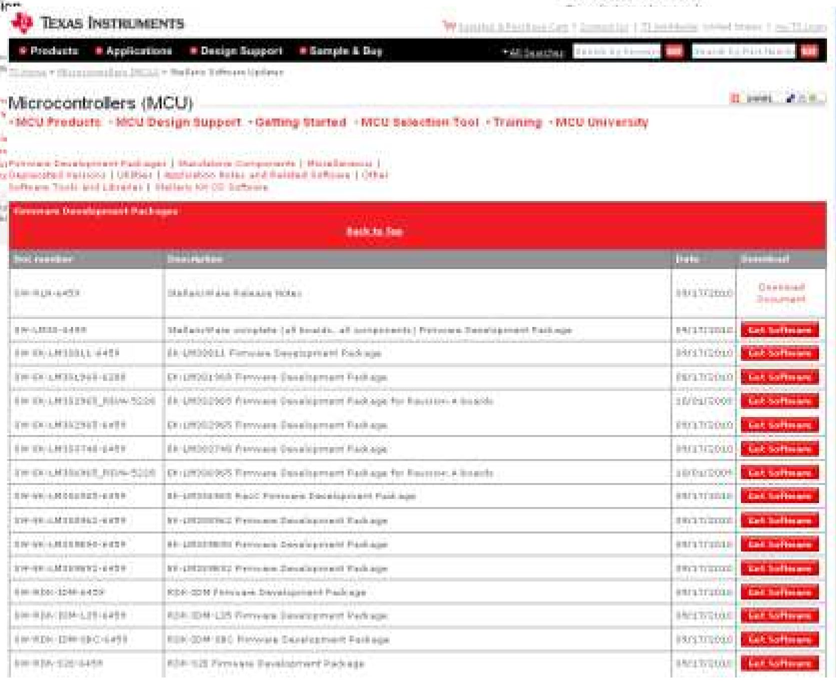
StellarisWare Complete (all boards, all components) Status: ACTIVE

SW-LM3S

	SW-LM3S
Name	StellarisWare Complete (all boards, all components)
Status	ACTIVE
Price (USD)	
Order Options	
Texas Instruments	Get Software

Product Information

Description



Microcontrollers (MCU)

- MCU Products · MCU Design Support · Getting Started · MCU Selection Tool · Training · MCU University

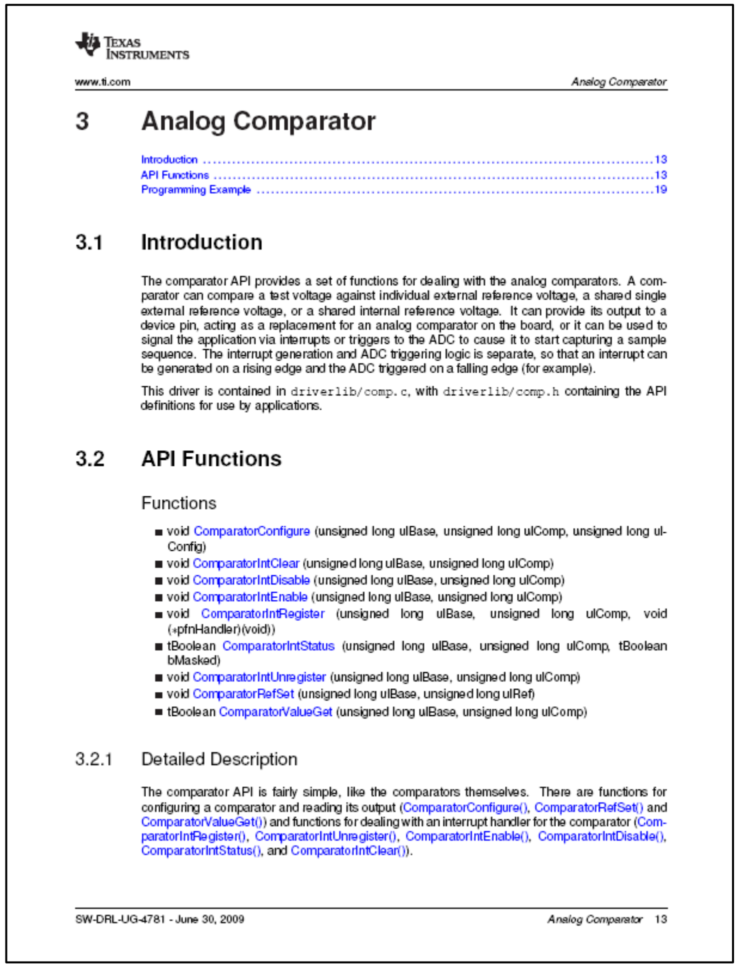
Firmware Development Packages

Doc number	Description	Date	Download
SW-RLN-6459	StellarisWare Release Notes	09/17/2010	Download Document
SW-LM3S-6459	StellarisWare complete (all boards, all components) Firmware Development Package	09/17/2010	Get Software
SW-EK-LM3S811-6459	EK-LM3S811 Firmware Development Package	09/17/2010	Get Software
SW-EK-LM3S1968-6288	EK-LM3S1968 Firmware Development Package	09/17/2010	Get Software
SW-EK-LM3S2965_REVA-5228	EK-LM3S2965 Firmware Development Package for Revision A boards	10/01/2009	Get Software
SW-EK-LM3S2965-6459	EK-LM3S2965 Firmware Development Package	09/17/2010	Get Software
SW-EK-LM3S3748-6459	EK-LM3S3748 Firmware Development Package	09/17/2010	Get Software
SW-EK-LM3S6965_REVA-5228	EK-LM3S6965 Firmware Development Package for Revision A boards	10/01/2009	Get Software
SW-EK-LM3S6965-6459	EK-LM3S6965 [illegible] Firmware Development Package	09/17/2010	Get Software
SW-EK-LM3S8962-6459	EK-LM3S8962 Firmware Development Package	09/17/2010	Get Software
SW-EK-LM3S9B90-6459	EK-LM3S9B90 Firmware Development Package	09/17/2010	Get Software
SW-EK-LM3S9B92-6459	EK-LM3S9B92 Firmware Development Package	09/17/2010	Get Software
SW-RDK-IDM-6459	RDK-IDM Firmware Development Package	09/17/2010	Get Software
SW-RDK-IDM-L35-6459	RDK-IDM-L35 Firmware Development Package	09/17/2010	Get Software
SW-RDK-IDM-SBC-6459	RDK-IDM-SBC Firmware Development Package	09/17/2010	Get Software
SW-RDK-S2E-6459	RDK-S2E Firmware Development Package	09/17/2010	Get Software

Firmware Development Packages | Standalone Components | Miscellaneous | Deprecated Versions | Utilities | Application Notes and Related Software | Other Software Tools and Libraries | Stellaris Kit CD Software
Back to Top

Peripheral Driver Library (DriverLib)

- 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**
- **StellarisWare is preprogrammed in ROM on most Stellaris MCUs**



TEXAS INSTRUMENTS

www.ti.com *Analog Comparator*

3 Analog Comparator

3.1 Introduction

The comparator API provides a set of functions for dealing with the analog comparators. A comparator 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 or triggers to the ADC to cause it to start capturing a sample sequence. The interrupt generation and ADC triggering logic is separate, so that an interrupt can be generated on a rising edge and the ADC triggered on a falling edge (for example).

This driver is contained in `driverlib/comp.c`, with `driverlib/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 ComparatorIntDisable (unsigned long ulBase, unsigned long ulComp)
- void ComparatorIntEnable (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(), ComparatorIntEnable(), ComparatorIntDisable(), ComparatorIntStatus(), and ComparatorIntClear()).

SW-DRL-UG-4781 - June 30, 2009 *Analog Comparator* 13

TEXAS INSTRUMENTS

Peripheral Driver Library Organization

- Collection of ‘.c’ source and ‘.h’ header files
- Base root directory structure with individual compiler specific library output directory.
- “.\StellarisWare” contains Hardware specific header files (default)
    - 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
C:\StellarisWare\driverlib\ccs	- Code Composer Studio

Peripheral Driver Library: 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)
    {
    }
}
```

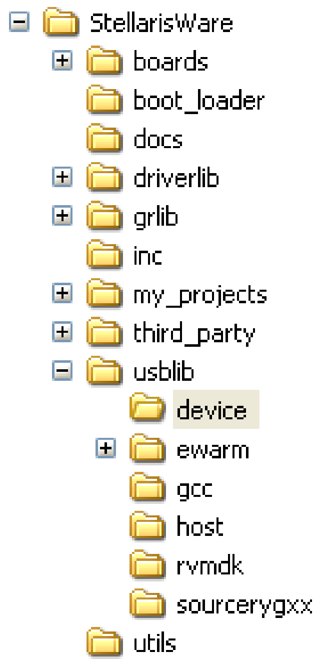
```
void
UARTIntHandler(void)
{
    unsigned long ulStatus;

    // 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 USB Library

- Royalty-free sample applications accelerate USB implementation on Stellaris MCUs
- Examples available:
  - **Device Examples:**
    - HID Keyboard
    - HID Mouse
    - CDC Serial
    - Mass Storage
    - Generic Bulk
    - Audio
    - Device Firmware Upgrade
    - Oscilloscope
  - **Host Examples:**
    - Mass Storage
    - HID Keyboard
    - HID Mouse
    - Isochronous Audio Input
  - **OTG Examples:**
    - SRP (Session Request Protocol)
    - HNP (Host Negotiation Protocol)*
  - **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**
- USB-IF Compliance
  - Stellaris MCUs pass USB Device and Embedded Host compliance testing
- TI sub-licenses Stellaris VID & PIDs for customer use!

StellarisWare USB Library File Organization

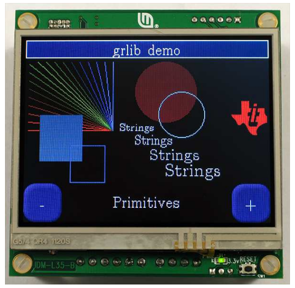
- '.\StellarisWare\usblib' contains all USB Library related files
  - Compiler specific project information
  - USB Class specific '.c' source and '.h'header files
    - CDC
    - HID
    - MSD
  - Other generic USB related '.c' source and header files
  - Compiler specific 'library' output directories
    - C:\StellarisWare\usblib\ewarm – IAR
    - C:\StellarisWare\usblib\gcc – CodeRed
    - C:\StellarisWare\usblib\rvmdk – Keil
    - C:\StellarisWare\usblib\sourcerygxx – CodeSourcery
- '.\StellarisWare\usblib\device' contains USB 'Device' specific files
- '.\StellarisWare\usblib\host' contains USB 'Host' specific files



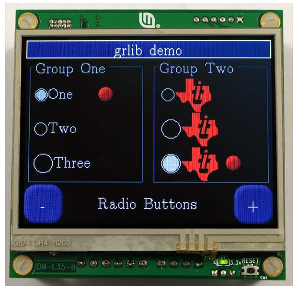
```
StellarisWare
  boards
  boot_loader
  docs
  driverlib
  grlib
  inc
  my_projects
  third_party
  usblib
    device
    ewarm
    gcc
    host
    rvmdk
    sourcerygxx
  utils
```

StellarisWare® Graphics Library

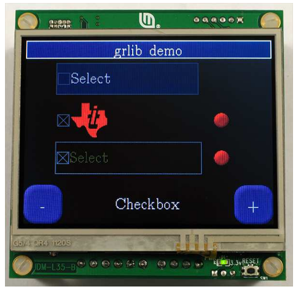
- Written entirely in C, easy-to-use, efficient.
- Three layers of functionality, each directly callable:
  - Display Driver Layer (Lowest Level)
  - Graphics Primitives Layer …
  - Widget Layer (Highest Level)
- Graphics Primitives:
  - Point, Line, Rectangle, Circle, Font, Image, Context, Buffer
  - 134 Computer Modern predefined fonts
  - Western European and Asian fonts
  - Support for 24-bit color
- Widgets:
  - Canvas, Checkbox, Container, Push Button, Radio Button, Slider, ListBox



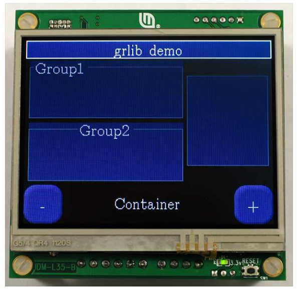
Primitives



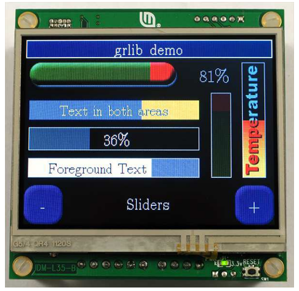
Radio Buttons



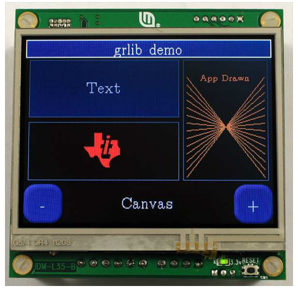
Checkbox



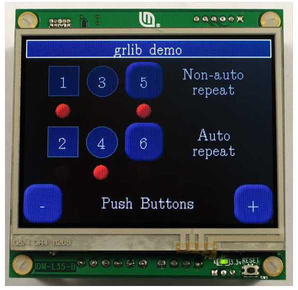
Container



Sliders

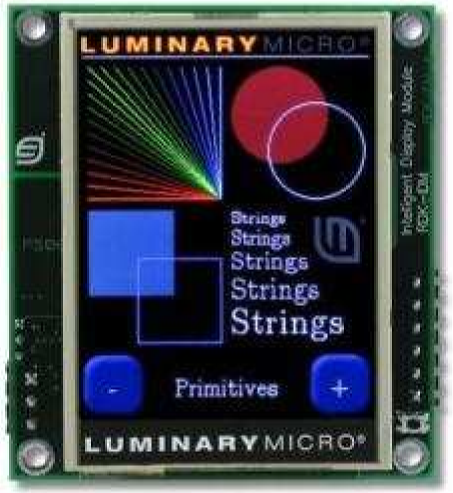


Canvas

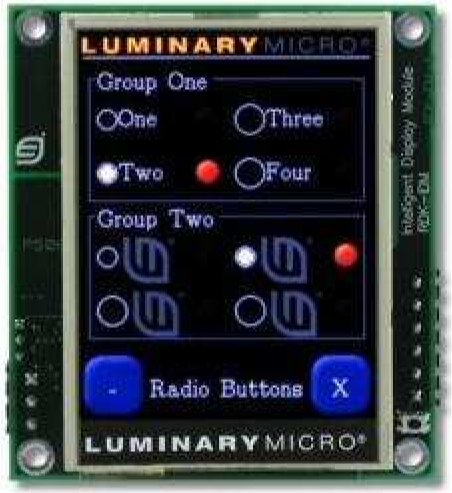


Push Buttons

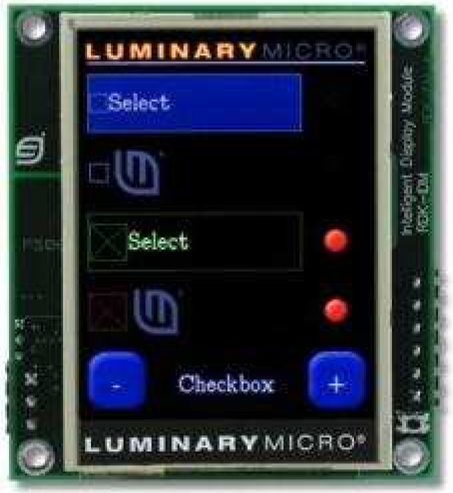
StellarisWare Graphics Library Examples



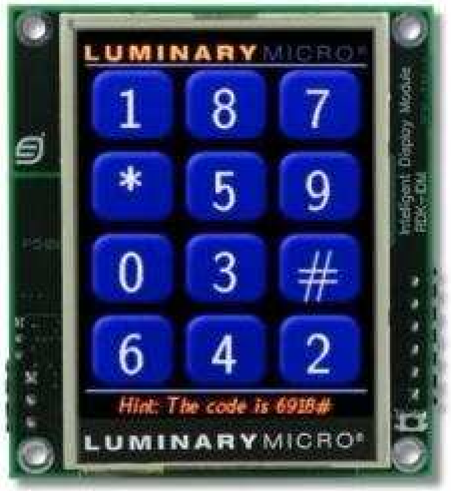
Primitives



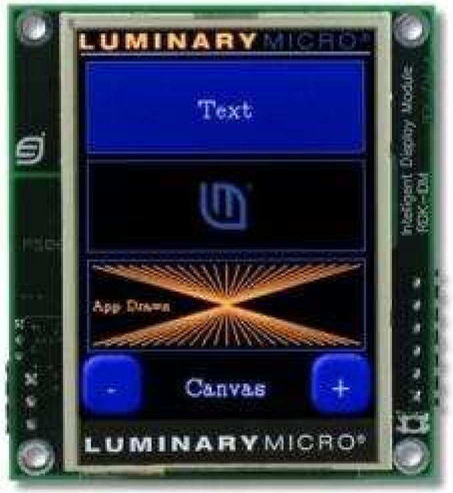
Radio Buttons



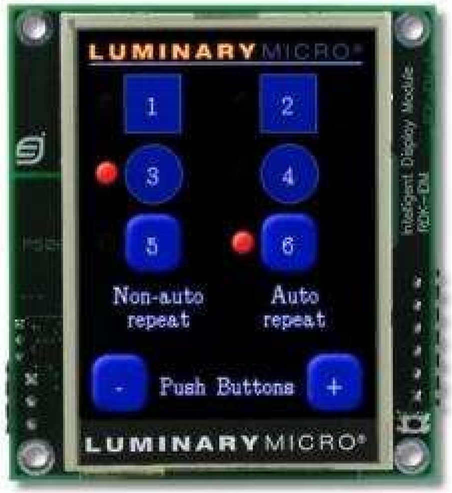
Checkbox



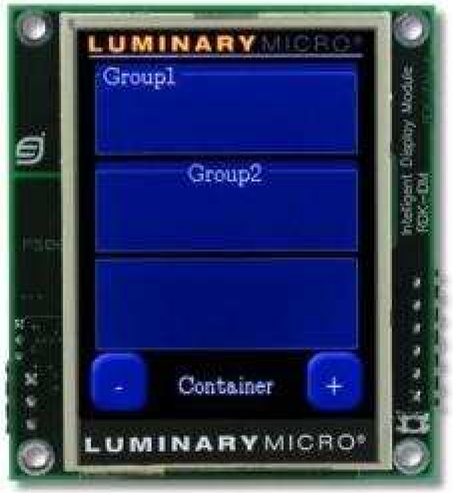
Security Keypad



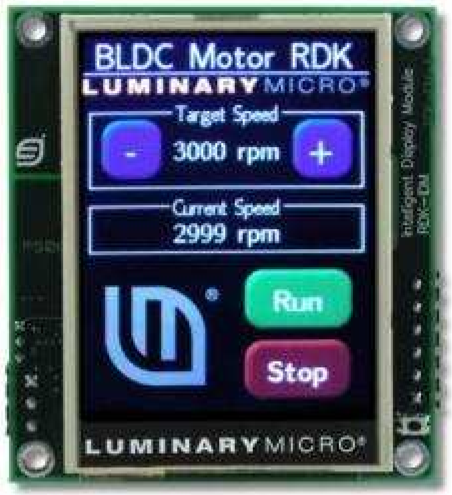
Canvas



Push Buttons



Container



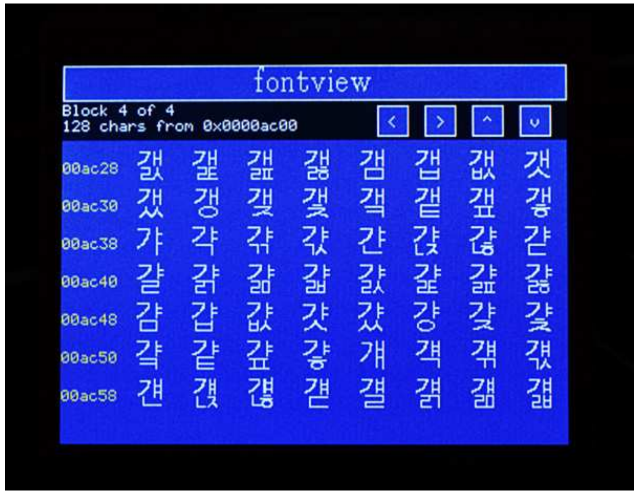
BLDC Touchscreen Motor Controller

StellarisWare® Graphics Library

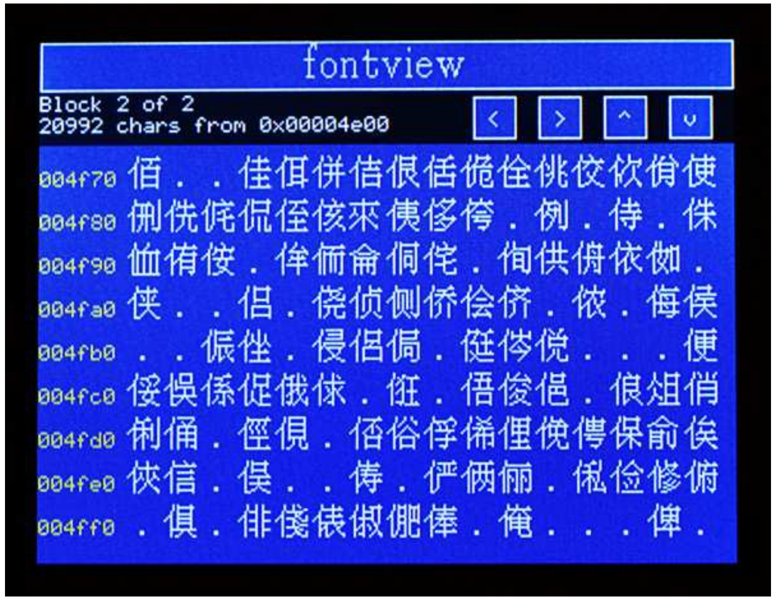
Special Utilities

- *ftrasterize*: render your own font
- *mkstringtable*: build multi-language string tables
- *makefsfile*: dump binary files as C arrays
- *lmi-button*: predefined button with shadow and 3-D
- *pnmtoc*: Convert image file to GraphicsLib format

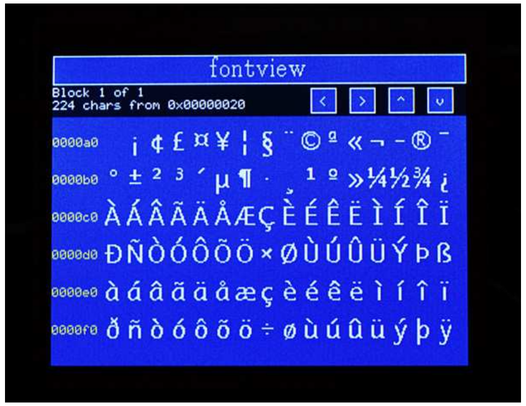
- **International Fonts;** program HMI in more languages
  - e.g. Chinese, Korean, Japanese ideographs, accented western European characters, etc.
  - Supports UTF8, ISO8859 and Unicode.



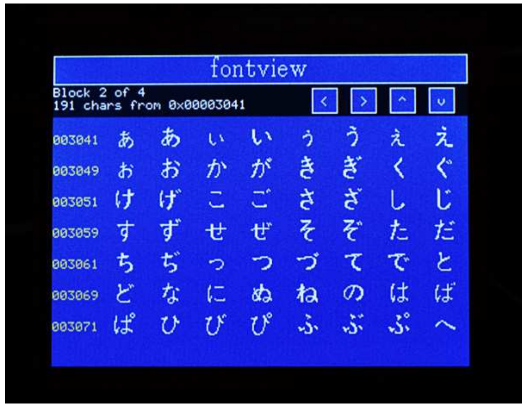
```
Block 4 of 4
128 chars from 0x0000ac00
00ac28 갨 갩 갪 갫 갬 갭 갮 갯
00ac30 갰 갱 갲 갳 갴 갵 갶 갷
00ac38 갸 갹 갺 갻 갼 갽 갾 갿
00ac40 걀 걁 걂 걃 걄 걅 걆 걇
00ac48 걈 걉 걊 걋 걌 걍 걎 걏
00ac50 걐 걑 걒 걓 걔 걕 걖 걗
00ac58 걘 걙 걚 걛 걜 걝 걞 걟
```



```
Block 2 of 2
20992 chars from 0x00004e00
004f70 佰..佳佴併佶佷佸佹佺佻佼佽佾使
004f80 侀侁侂侃侄侅來侇侈侉.例.侍.侏
004f90 侐侑侒.侔侕侖侗侘.侚供侜依侞.
004fa0 侠..侣.侥侦侧侨侩侪.侬.侮侯
004fb0 ..侲侳.侵侶侷.侹侺侻...便
004fc0 俀俁係促俄俅.俇.俉俊俋.俍俎俏
004fd0 俐俑.俓俔.俖俗俘俙俚俛俜保俞俟
004fe0 俠信.俣..俦.俨俩俪.俬俭修俯
004ff0 .俱.俳俴俵俶俷俸.俺...俾.
```



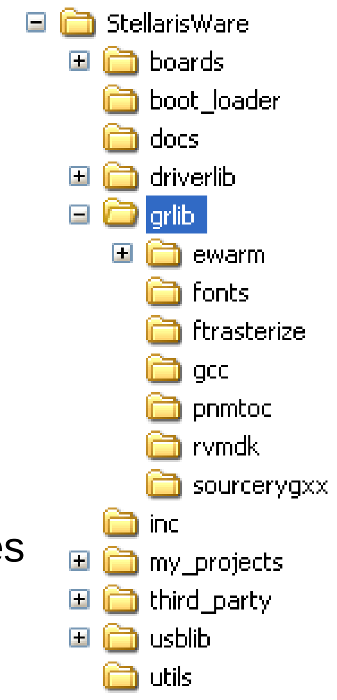
```
Block 1 of 1
224 chars from 0x00000020
0000a0   ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ - ® ¯
0000b0 ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
0000c0 À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
0000d0 Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
0000e0 à á â ã ä å æ ç è é ê ë ì í î ï
0000f0 ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```



```
Block 2 of 4
191 chars from 0x00003041
003041 ぁ あ ぃ い ぅ う ぇ え
003049 ぉ お か が き ぎ く ぐ
003051 け げ こ ご さ ざ し じ
003059 す ず せ ぜ そ ぞ た だ
003061 ち ぢ っ つ づ て で と
003069 ど な に ぬ ね の は ば
003071 ぱ ひ び ぴ ふ ぶ ぷ へ
```

StellarisWare Graphics Library File Organization

- '.\StellarisWare\grlib' contains all Graphics Library related files
  - Compiler specific project information
  - Graphics Library specific '.c' source and '.h' header files for all graphics objects
    - Canvas, checkbox, circle, container, context
    - Image, line, listbox, pushbutton, radiobutton
    - Rectangle, string, slider, widget
  - Other generic Graphics Library related 'c' source and header files
  - Compiler specific 'library' output directories
    - .\StellarisWare\grlib\ewarm – IAR
    - .\StellarisWare\grlib\gcc – CodeRed
    - .\StellarisWare\grlib\rvmdk – Keil
    - .\StellarisWare\grlib\sourcerygxx – CodeSourcery
  - '.\StellarisWare\grlib\fonts' contains Graphics Library font specific files
- Graphics Library utilities
  - '.\StellarisWare\grlib\ftrasterise' for creating fonts
  - '.\StellarisWare\grlib\pnmtoc' for creating images



StellarisWare® - Safe at Home with IEC 60730

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

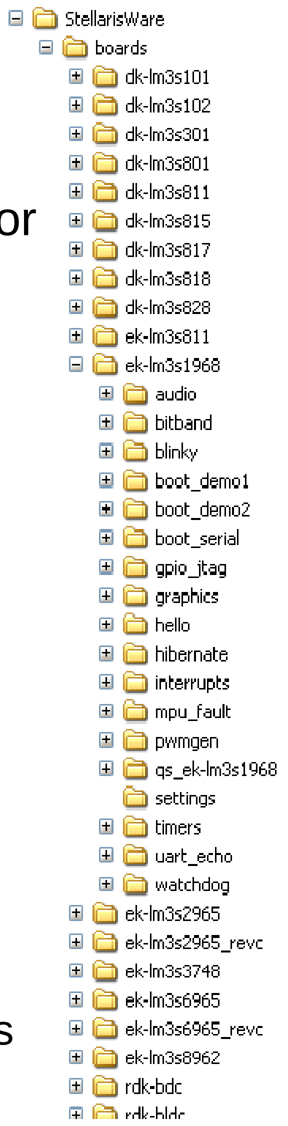
	Module	Description
StellarisWare® Software	Reset Handler	Performs basic register and memory test out of reset.
	CPU Test	Performs stuck bit testing on the CPU PC and registers.
	SRAM Test	Performs stuck bit testing on the SRAM.
	Flash Test	Performs a CRC test on the Flash.
	ADC Test	Performs a conversion 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	

See App Note AN01272: Using the IEC 60730 Standard for Safe and Reliable Operation of Stellaris® Microcontrollers

Texas Instruments

StellarisWare Code Examples

- '.\StellarisWare\boards' contains all development, evaluation and reference design kit projects
- Each EVM / DK /RDK 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 for supported compilers
  - Project source code files
  - Compiler specific 'library' output directories
    - .\StellarisWare\boards\EVM\Project\ewarm – IAR
    - .\StellarisWare\boards\EVM\Project\gcc – CodeRed
    - .\StellarisWare\boards\EVM\Project\rvmdk – Keil
    - .\StellarisWare\boards\EVM\Project\sourcerygxx – CodeSourcery
- All projects that use the driver library reference
  - Driver '*library*' file from '.\StellarisWare\driverlib\complier'
  - Driver Library header '*.h' file from '.\StellarisWare\driverlib'
- Other related files
  - '.\StellarisWare\inc' – Device specific header 'lm3sxxxx.h' files
  - '.\StellarisWare' – Peripheral hardware header 'hw_xxxx.h' files

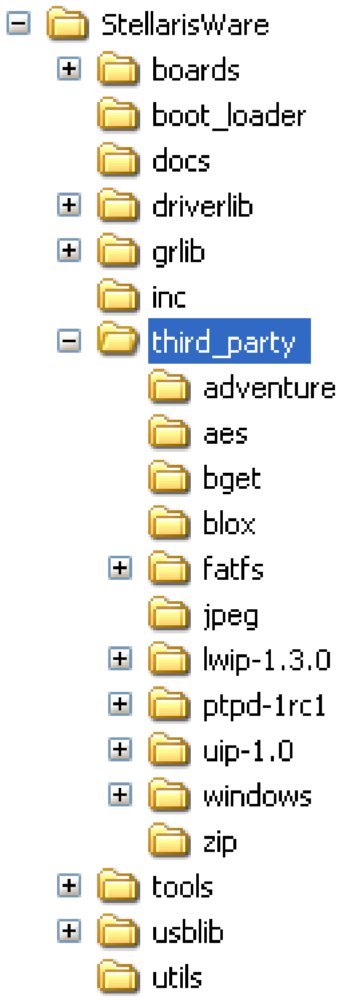


```
StellarisWare
  boards
    dk-lm3s101
    dk-lm3s102
    dk-lm3s301
    dk-lm3s801
    dk-lm3s811
    dk-lm3s815
    dk-lm3s817
    dk-lm3s818
    dk-lm3s828
    ek-lm3s811
    ek-lm3s1968
      audio
      bitband
      blinky
      boot_demo1
      boot_demo2
      boot_serial
      gpio_jtag
      graphics
      hello
      hibernate
      interrupts
      mpu_fault
      pwmgen
      qs_ek-lm3s1968
      settings
      timers
      uart_echo
      watchdog
    ek-lm3s2965
    ek-lm3s2965_revc
    ek-lm3s3748
    ek-lm3s6965
    ek-lm3s6965_revc
    ek-lm3s8962
    rdk-bdc
    rdk-bldc
```

- Most Stellaris peripherals are by-and-large the exact same on each MCU / board.
- Code examples are usually distinct for each Stellaris HW kit.
- Therefore, download all of StellarisWare and check out the code examples for each kit !
  e.g. Distinct Ethernet code examples: -6965, -9B92, -9B96, -S2E, -BLDC, -IDM, etc.

StellarisWare Third Party File Organization

- Many application examples use third party code
  - ‘.\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 th required source code ‘*.c’ and header ‘*.h’ from the relevant ‘*third_party*’ directory.



StellarisWare Additional Features

- StellarisWare Bootloader
  - Download code to flash memory for firmware updates
  - Interface options include USB, UART, CAN, I2C, SPI, ENET
- Other flash memory-saving options
  - Peripheral DriverLib, USBLib, GraphicsLib offered in ROM
  - Advanced Encryption Standared (AES) tables for AES-128, -192, - 256
    - Comes with AES reference application software
  - Cyclic Redundancy Check (CRC) functionality for error detection
- StellarisWare allows programmers to focus on the application – not the setup!

StellarisWare®

StellarisWare In-system Programming Options

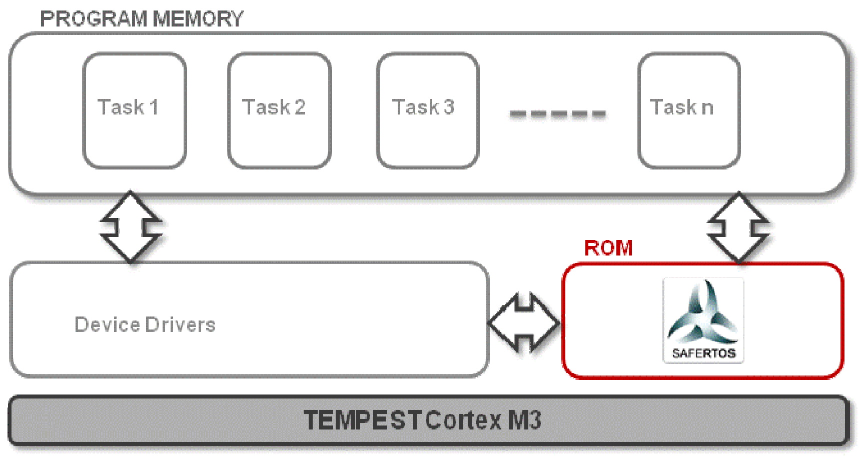
Stellaris Serial Flash Loader

- Small piece of code that allows programming of the flash without the need for a debugger interface.
- All Stellaris MCUs ship with this pre-loaded in flash
- Interface options include UART or SSI
- TI supplies a Windows™ application (GUI or command line) that makes full use of all commands supported by the serial flash loader (LMflash.exe)

Stellaris Boot Loader

- Small piece of code that can be programmed at the beginning of flash to act as an application loader
- Also used as an update mechanism for an application running on a Stellaris microcontroller.
- Interface options include UART (default), I^{2}C, SSI, Ethernet, USB
- Included in the Stellaris Peripheral Driver Library with full applications examples
- Preloaded in ROM on select Stellaris Microcontrollers

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

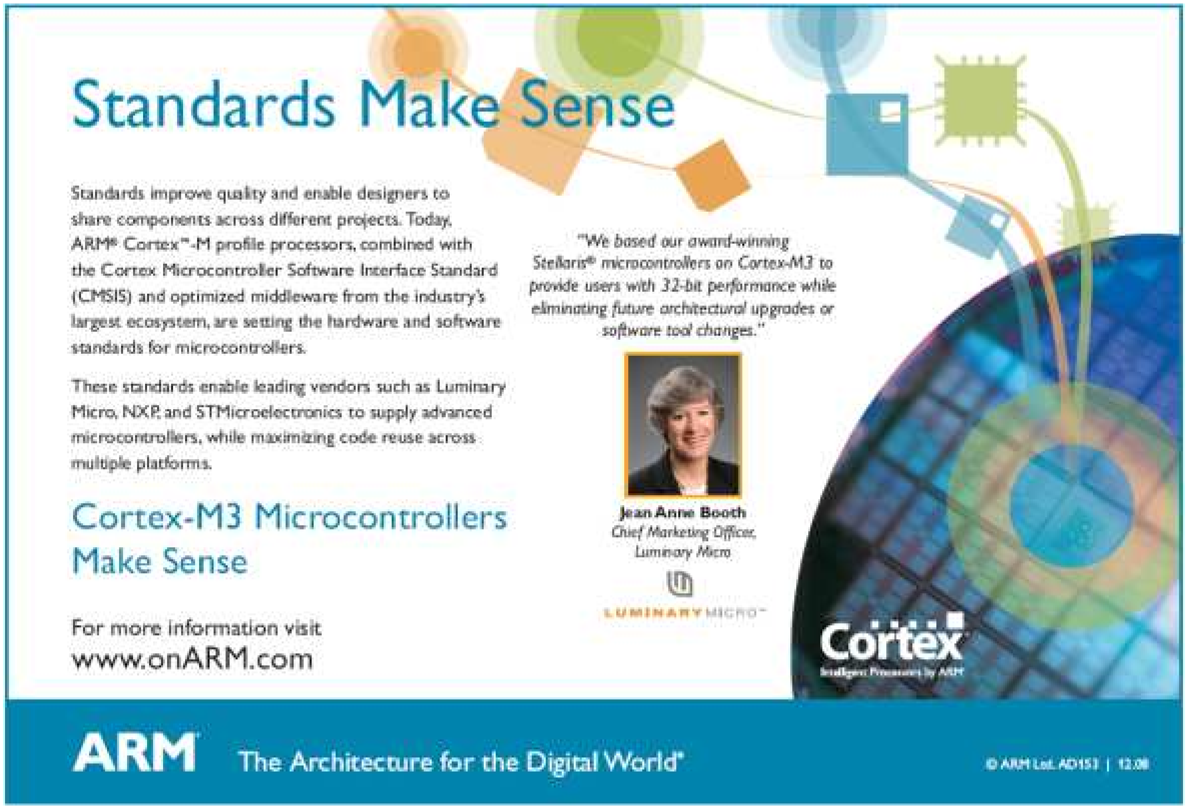
StellarisWare Application Notes

- Programming the On-Chip Flash Memory in a Stellaris Microcontroller
- Driving a Brushless DC Motor with a Stellaris Microcontroller
- ADC Oversampling Techniques
- Clocking options for Stellaris Family Microcontrollers
- Using a Stellaris Microcontroller as an I/O Processor
- Using the Stellaris Serial Flash Loader
- Adding 32KB of Serial SRAM to a Stellaris Microcontroller
- Evaluating PeerSec Networks' MatrixSSL on a Stellaris Microcontroller
- Creating an Autonomous Car with the Stellaris LM3S316 Microcontroller
- Using the Stellaris® LM3S615 and LM3S316 Microcontrollers to Control a CNC Machine
- Using the Stellaris Microcontroller Analog-to-Digital Converter
- Using the Stellaris Boot Loader
- Upgrading to TI's Stellaris Microcontrollers from Microchip's PIC Microcontrollers
- Migrating to the New Members of the Stellaris® Family of Microcontrollers
- Implementing RS-232 Flow Control on a Stellaris® Microcontroller
- Using Stellaris® Microcontrollers' PCB Libraries
- Flash Protection for Stellaris® Microcontrollers
- Using the Stellaris® Ethernet Controller with Micro IP (uIP)
- Using the Stellaris® Ethernet Controller with Lightweight IP (lwIP)
- Optimizing Code Performance and Size for Stellaris Microcontrollers
- Serial-to-Ethernet Converter for Stellaris Microcontrollers
- Software UART for Stellaris® Microcontrollers
- USB Certification for Stellaris® Microcontroller-based USB Peripherals and Embedded Host Systems
- Using the IEC 60730 Standard for Safe and Reliable Operation of Stellaris® Microcontrollers
- Application Update Using the USB Device Firmware Upgrade Class
- Configuring Stellaris® Microcontrollers with Pin Multiplexing

http://www.luminarymicro.com/home/app_notes.html

Now Available: CMSIS Support for Stellaris

- Introducing CMSIS Support for Stellaris Microcontrollers
  - Cortex Microcontroller Software Interface Standard
  - See http://www.keil.com/support/docs/3440.htm
  - Available at http://www.ti.com/software_updates



Standards Make Sense

Standards improve quality and enable designers to share components across different projects. Today, ARM® Cortex™-M profile processors, combined with the Cortex Microcontroller Software Interface Standard (CMSIS) and optimized middleware from the industry's largest ecosystem, are setting the hardware and software standards for microcontrollers.

These standards enable leading vendors such as Luminary Micro, NXP, and STMicroelectronics to supply advanced microcontrollers, while maximizing code reuse across multiple platforms.

Cortex-M3 Microcontrollers Make Sense

For more information visit
www.onARM.com

"We based our award-winning Stellaris® microcontrollers on Cortex-M3 to provide users with 32-bit performance while eliminating future architectural upgrades or software tool changes."

Jean Anne Booth
Chief Marketing Officer, Luminary Micro

LUMINARY MICRO™

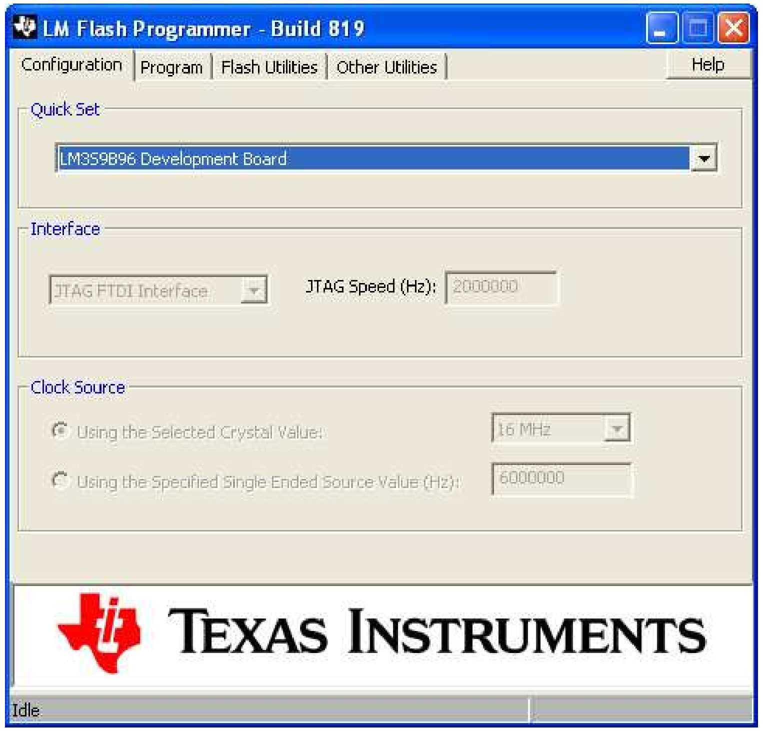
Cortex
Intelligent Processors by ARM

ARM The Architecture for the Digital World®

© ARM Ltd. AD153 | 12.08

StellarisWare Serial Flash Programming GUI

- **LM Flash Programming GUI**
  - Simple graphical user interface
  - Support for all Evaluation Kits
  - Key features include:
    - Program
    - Verify
    - Erase
    - Read memory
  - Available now
    - http://www.ti.com/software_updates



Quick Set: LM3S9B96 Development Board

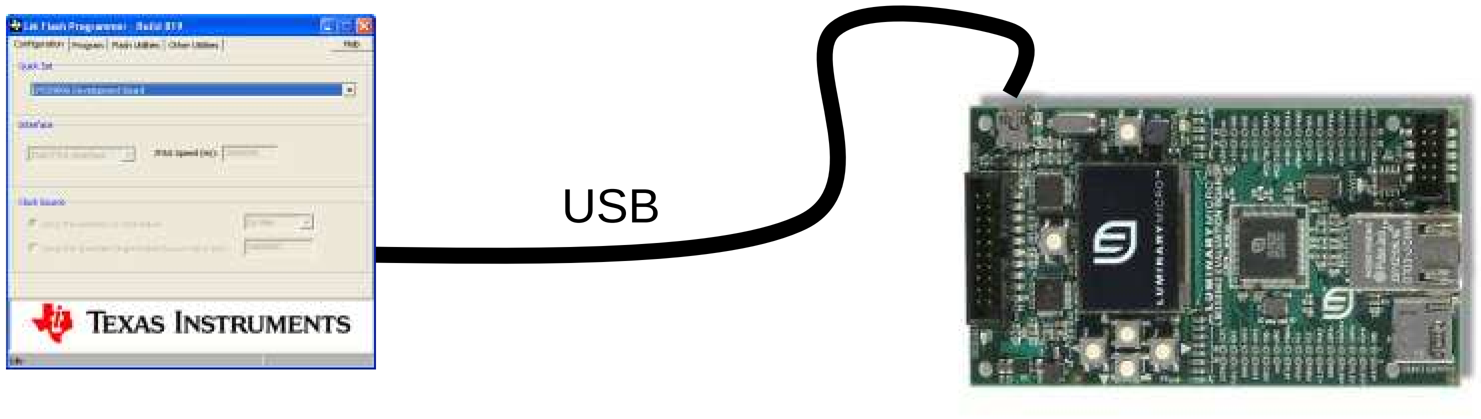
Interface: JTAG FTDI Interface — JTAG Speed (Hz): 2000000

Clock Source:
- Using the Selected Crystal Value: 16 MHz
- Using the Specified Single Ended Source Value (Hz): 6000000

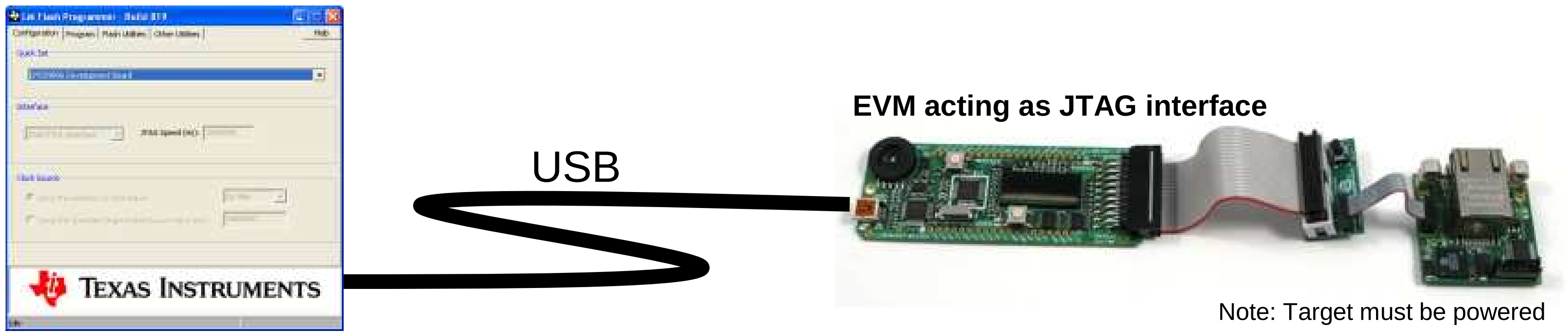
Idle

Flash Programming GUI Supports:

- Programming evaluation kits (EVM) directly



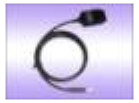
- Programming target HW indirectly via EVM



Stellaris Third Party Programmer Solutions

- Desktop – BPM Micro, System General [SGI], Xeltek
  - *BPM Micro*: 28-SOIC, 48-LQFP, 100-LQFP; 64-LQFP; 108-BGA.
  - *Programtec*: 28-SOIC, 48-LQFP, 100-LQFP; 64-LQFP; 108-BGA.
  - *SGI*: 28-SOIC, 48-LQFP, 100-LQFP; 64-LQFP; 108-BGA.
  - *Xeltek*: 28-SOIC, 48-LQFP; 100-LQFP, 64-LQFP, 108-BGA.

- ISP – Texas Instruments, Keil, IAR, ZLG, SEGGER, Abatron
  - ISP = In-System Programmer
  - *Texas Instruments:* Any Stellaris evaluation kit can be used as a USB-to-JTAG ISP.
  - *Keil*: ULINK2 ARM USB-JTAG ISP.
  - *IAR*: J-Link ARM USB-JTAG ISP.
  - *ZLG*: LM Link USB-JTAG ISP.
  - *SEGGER*: J-Link ARM USB JTAG ISP and Flasher Standalone ISP Programmer.
  - *FTDI*: USB-to-TTL Serial Converter cable (used with LMFlash GUI and Stellaris Serial Flash Loader)
  - *Abatron*: BDI3000 JTAG ISP.
  - *CooCox:* Colink USB-JTAG ISP.
  - *Code Red:* Red Probe USB-JTAG ISP.



- Service – Source USA
- Production – BPM Micro

Stellaris Partners in Excellence

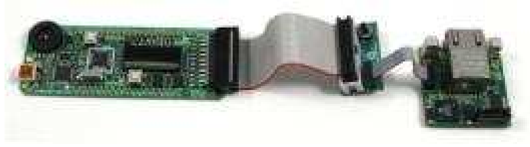
Product	Third Party	Description
Compiler / Debugger	Code Red	Red Suite (GNU C/C++ Compiler, code_probe / Eclipse Debugger / IDE)
	CodeSourcery	CodeSourcery G++ (C/C++ Compiler), GDB / Eclipse Debugger / IDE
	IAR	IAR C/C++ Compliler, C-SPY / Embedded Workbench Debugger / IDE
	Keil	RealView C/C++ Compiler, µVision Debugger / IDE
	Rowley	CrossWorks for ARM (C/C++ Compiler, CrossStudio Debugger / IDE)
RTOS	CMX	CMX-RTX™ RTOS offering small footprint, fast context switch times
	ExpressLogic	ThreadX advanced RTOS designed specifically for deeply embedded applications
	FreeRTOS.org	FreeRTOS.org™ Open-Source mini real time kernel
	IAR	PowerPac™ fully featured RTOS combined with a high performance file system
	Keil	RTX flexible royalty-free RTOS with source code
	Micrium	Portable, scalable, preemptive real-time, multitasking kernel (RTOS)
	Quadros	RTXC for embedded applications
	RoweBots	Unison Ultra Tiny Embedded Linux and POSIX Compatible RTOS
	SCIOPTA	SCIOPTA real-time operating system for safety-critical applications
	SEGGER	embOS RTOS for embedded applications designed
Stacks / Specialty	CMX	CMX-USB Device, CMX-CANopen™, CMX MicroNet, and TCP/IP protocol stacks
	eLua	Embedded Lua Programming Language for Stellaris
	ExpressLogic	NetX™ TCP/IP and USBX™ supporting USB Host and Device
	Interniche	NicheLite and ARM Network Evaluation Kits
	Micrium	µC/USB Device, µC/USB Host, µC/TCP-IP, µC/Modbus, µC/CAN protocol stacks
	MicroDigital	smxUSBD Device, smxUSBH Host, and smxUSBO On-The-Go (OTG) Stacks
	port GmbH	CANopen Library for Stellaris Microcontrollers
	Quadros	RTXCusb Host and Device stacks, CANopenRT CAN stack, and QuadNet TCP/IP
	RTA Automation	RTA Automation DeviceNet™ protocol stacks
	SEGGER	embOS/IP TCP/IP and emUSB Device Stack
	SEVENSTAX	SEVENSTAX TCP/IP-Stack and Embedded Web Server

KEIL An ARM® Company
IAR SYSTEMS
SCIOPTA
CODESOURCERY
Rowley Associates
code_red
RoweBots
CMX SYSTEMS
Micro Digital
SEGGER
Quadros
expresslogic
free RTOS
Micrium
REAL TIME AUTOMATION
port
SEVENSTAX EMBEDDED TECHNOLOGIES

Development Tools for Stellaris MCUs

	CodeSourcery	IAR Systems	ARM Keil	code_red	Texas Instruments
Eval Kit License	30-day full function. Upgradeable.	32KB address-limited. Upgradeable.	32KB address-limited. Upgradeable.	90-day full function. Upgradeable.	Full functional; locked to board. Upgradeable.
Compiler	GNU C/C++	IAR C/C++	RealView C/C++	GNU C/C++	CCStudio
Debugger / IDE	gdb / Eclipse	C-SPY / Embedded Workbench	µVision	code_probe / Eclipse-based tool suite	CCStudio / Eclipse
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)	445 USD (includes full license for TI Stellaris, C2000, and MSP430 MCUs)
JTAG Debugger	Stellaris ICDI (on Stellaris EVK)	J-Link, ~299 USD	U-Link, ~199 USD	Red Probe, 150 USD	Stellaris ICDI (on Stellaris EVK)

Remember: 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.



RTOS Support for Stellaris:

Industry RTOS Basics	FreeRTOS
Scheduler (policy, threads)	Pre-emptive (3 thread types)
Small footprint, fast execution	Very small (<5KB), fast (uC-specific)
Dynamic/static declarations	Threads (dyn), data (static/dynamic)
Object based	Yes
User APIs (via library or src)	3 C source files provide entire kernel
File Mgmt (nice-to-have)	yes, but not built in
Cost – $0+	$Free$

OPENRTOS

- Can upgrade from freeRTOS
- <5KB (ROMable)
- Supports MSP430 and LM3
- Commercial license ($2500+)

SAFERTOS

- Can upgrade from freeRTOS
- Certified IEC 61508 (TUV SUD) SIL3 (Safety Integrity Level)
- Areospace/medical apps
- $65K license
- ROM'd (LM3S9B96)

RTOS support for Stellaris

KEIL™ An ARM® Company	RTX flexible royalty-free RTOS with source code
IAR SYSTEMS	PowerPac™ fully featured RTOS combined with a high performance file system
CMX SYSTEMS	CMX-RTX™ RTOS offering small footprint, fast context switch times
SEGGER	embOS RTOS for embedded applications designed
Quadros Systems Inc.	RTXC for embedded applications
SCIOPTA	SCIOPTA real-time operating system for safety-critical applications
RoweBots	Unison Ultra Tiny Embedded Linux and POSIX Compatible RTOS
Micrium	Portable, scalable, preemptive real-time, multitasking kernel (RTOS)
expresslogic	ThreadX advanced RTOS designed specifically for deeply embedded applications
free RTOS™	FreeRTOS.org™ Open-Source mini real time kernel

TCP/IP Communications Stacks for Stellaris

µC/TCP-IP Protocol Stack	Micri µ m µC/TCP-IP
expresslogic	Express Logic NetX™ TCP/IP protocol stack
CMX SYSTEMS	CMX-MicroNet™ protocol stacks
interniche technologies, inc.	InterNiche TCP/IP NicheStack™, NicheLITE™, and add-on modules such as HTTP, SNMP, and security protocols
REAL TIME AUTOMATION	EtherNet/IP™ protocol stacks
freeRTOS	FreeRTOS.org Open-Source µIP Embedded web server
µIP	**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

Networking Stacks Supporting 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					●	●	●	●							●	●						●	●	●				●		●
Express Logic	NetX	TCP/IP	●								●	●		●						●								●			●	
Express Logic	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

Typical Stacks Options

Acronym	Translation	Wikipedia Link	High-level Purpose
TCP	- Transmission Control Protocol	wikipedia Link	(guarantee delivery)
IP	- Internet Protocol	wikipedia Link	(data oriented)
UDP	- User Datagram Protocol	wikipedia Link	(fire-and-forget)
ARP	- Address Resolution Protocol	wikipedia Link	(finding a address)
RARP	- Reverse Address Resolution Protocol	wikipedia Link	(finding a address)
BOOTP	- Bootstrap Protocol	wikipedia Link	(finding a address)
DHCP	- Dynamic Host Configuration Protocol	wikipedia Link	(adding devices to a network)
BSD	- Berkeley Socket	wikipedia Link	(connecting to the internet)
ICMP	- Internet Control Message Protocol	wikipedia Link	(error message generation)
IGMP	- Internet Group Management Protocol	wikipedia Link	(manage IP multicast groups)
PPP	- Point-To-Point Protocol	wikipedia Link	(direct point-to-point connection)
SLIP	- Serial Line Internet Protocol	wikipedia Link	(direct point-to-point connection)
DNS	- Domain Name System	wikipedia Link	(translate host name to address)
FTP	- File Transfer Protocol	wikipedia Link	(transfer files point-to-point)
TFTP	- Trivial File Transfer Protocol	wikipedia Link	(FTP, but for smaller files)
RIP	- Routing Information Protocol	wikipedia Link	(routing internal networks)
RTP/RTCP	- Real-time Transport (Control) Protocol	wikipedia Link	(send audio/video over internet)
Telnet	- Terminal Emulation	wikipedia Link	(remote access)
HTTP	- Hypertext transfer Protocol Server	wikipedia Link	(publish/retrieve web pages)
SNMP	- Simple Network Management Protocol	wikipedia Link	(manage/monitor client status)
SMTP	- Simple Mail Transport Protocol	wikipedia Link	(send email over internet)
POP3	- Post Office Protocol-3	wikipedia Link	(retrieve email over internet)
SNTP	- Synchronized Network Time Protocol	wikipedia Link	(network clock synchronization)
PTP*	- Precision Time Protocol (also called IEEE1588)	wikipedia Link	(deterministic synchronization)
NAT	- Network Address Translation	wikipedia Link	(network privacy)
SSL	- Secure Sockets Layer	wikipedia Link	(secure communication)
IPSec	- Internet Protocol Security	wikipedia Link	(virtual private network)
IKE	- Internet Key Exchange	wikipedia Link	(security key/certificate sharing)

***Several Stellaris MCUs integrate hardware assistance for IEEE1588 PTP.**

USB Communications Stacks for Stellaris

CMX SYSTEMS	CMX-USB Device
SEGGER	emUSB Device stack
expresslogic	USBX supporting USB host and device
Micriμm	uC/USB Host, uC/USB Device
Quadros Systems Inc.	RTXCusb Host and Device stacks
Micro Digital	smxUSBD Device stack, smxUSBH Host stack, smxO On-The-Go (OTG) stacks

CAN Communication Stacks for Stellaris

CMX SYSTEMS	CMX – CANopen
Quadros	CANopenRT CAN stack
Micriμm	uC/CAN Protocol stacks
REAL TIME AUTOMATION	RTA Automation DeviceNet protocol stack
port	CANopen library