



# Stellaris<sup>®</sup> One Day Workshop 2010

---

*Student Guide*



*Revision 1.4  
July 2010*



## Important Notice

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Copyright © 2005, 2010 Texas Instruments Incorporated

## Revision History

|              |               |                                  |
|--------------|---------------|----------------------------------|
| Version 1.0  | November 2009 | TTO release of workshop          |
| Version 1.1  | January 2010  | Errata                           |
| Version 1.2  | January 2010  | Errata                           |
| Version 1.3  | February 2010 | Lab step errata                  |
| Version 1.31 | February 2010 | Lab step errata                  |
| Version 1.4  | July 2010     | New roadmap and portfolio slides |

### Mailing Address

Texas Instruments  
Training Technical Organization  
7839 Churchill Way, M/S 3984  
Dallas, Texas 75251-1903

# Stellaris<sup>®</sup> One-Day Workshop

---

## Introduction

Welcome to the Texas Instruments Stellaris one-day workshop. This workshop has been segmented in order for it to be presented in a variety of formats: lunch and learn style, half day and full day.

If you are attending the lunch and learn format, this introductory presentation will be presented.

If you are attending a half-day format, you will also have the chance to get hands-on with Code Composer Studio, and the LM3S3748 and LM3S8962 evaluation kits.

If you are attending the full-day presentation, your afternoon will be a series of in-depth presentations and labs covering the USB, CAN and Ethernet peripherals.

Whichever format that you're here for, welcome to the class ... let's get started!

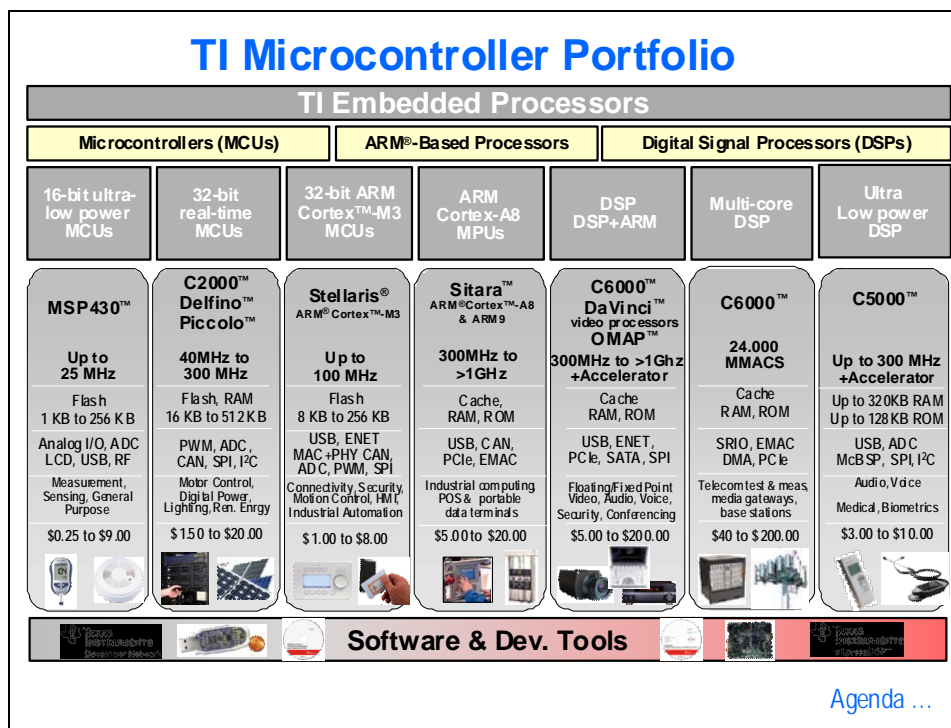
## Objectives

- **Stellaris Microcontrollers Overview**
- **Stellaris Key Advantages**
- **Evaluation and Reference Design Kits**
- **Development Tools and Software Support**
- **Product Demonstrations**
- **Summary**

# Module Topics

|  |             |
|--|-------------|
| <b>Stellaris® One-Day Workshop.....</b>            | <b>1-1</b>  |
| <i>Module Topics.....</i>                          | <i>1-2</i>  |
| <i>TI Microcontroller Portfolio.....</i>           | <i>1-3</i>  |
| <i>Stellaris ARM® Cortex™- M3 Overview .....</i>   | <i>1-4</i>  |
| ARM® Cortex™-M3 Benefits .....                     | 1-4         |
| No Assembly Required.....                          | 1-5         |
| First in ARM® Cortex™-M3 Microcontrollers .....    | 1-5         |
| Roadmap.....                                       | 1-6         |
| Family Technology.....                             | 1-6         |
| Internal Memories .....                            | 1-7         |
| Memory Protection Unit.....                        | 1-7         |
| Integrated MAC+PHY.....                            | 1-8         |
| NVIC .....   | 1-9         |
| External Peripheral Interface .....                | 1-10        |
| Battery-Backed Hibernation .....                   | 1-10        |
| Motor Control.....                                 | 1-11        |
| <i>Evaluation and Reference Design Kits .....</i>  | <i>1-12</i> |
| Motor Control Reference Design Kits.....           | 1-12        |
| Reference Design Kits.....                         | 1-13        |
| Modules.....                                       | 1-13        |
| Product Development Flexibility.....               | 1-14        |
| <i>Development Tools and Support Software.....</i> | <i>1-15</i> |
| Stellaris Partners .....                           | 1-15        |
| StellarisWare .....                                | 1-16        |
| Available On-Line .....                            | 1-16        |
| Peripheral Driver Library .....                    | 1-17        |
| Graphics Library.....                              | 1-17        |
| USB Library .....                                  | 1-18        |
| IEC 60730.....                                     | 1-18        |
| In System Programming .....                        | 1-19        |
| Flash Programming GUI .....                        | 1-19        |
| ROM Enhancements.....                              | 1-20        |
| SAFERTOS .....                                     | 1-20        |
| <i>Product Demonstrations .....</i>                | <i>1-21</i> |
| CAN/Ethernet Demo .....                            | 1-21        |
| CNC Machine Demo .....                             | 1-22        |
| Autonomous Car.....                                | 1-22        |
| FIRST Robotics.....                                | 1-23        |
| <i>Summary .....</i>                               | <i>1-24</i> |

# TI Microcontroller Portfolio



# Stellaris ARM® Cortex™- M3 Overview

## What Is The ARM® Cortex™-M3?

The ARM Cortex family is comprised of three series

- **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®

For more information on the ARM Cortex-M3, see:

**The Definitive Guide to the ARM Cortex-M3** by Joseph Yiu

ISBN: 978-0-7506-8534-4

Benefits ...

## ARM® Cortex™-M3 Benefits

### Cortex™-M3 Benefits

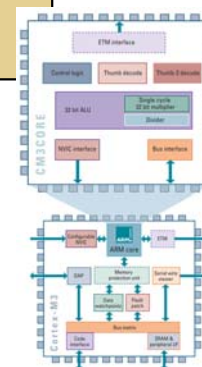
◆ **Capabilities beyond ARM7 for the MCU market:**

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



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

Source: [http://www.arm.com/products/CPUs/ARM\\_CortexM3.html](http://www.arm.com/products/CPUs/ARM_CortexM3.html)


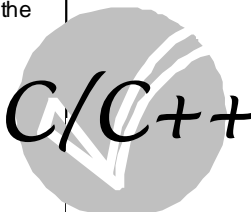


No ASM Required ...

## No Assembly Required

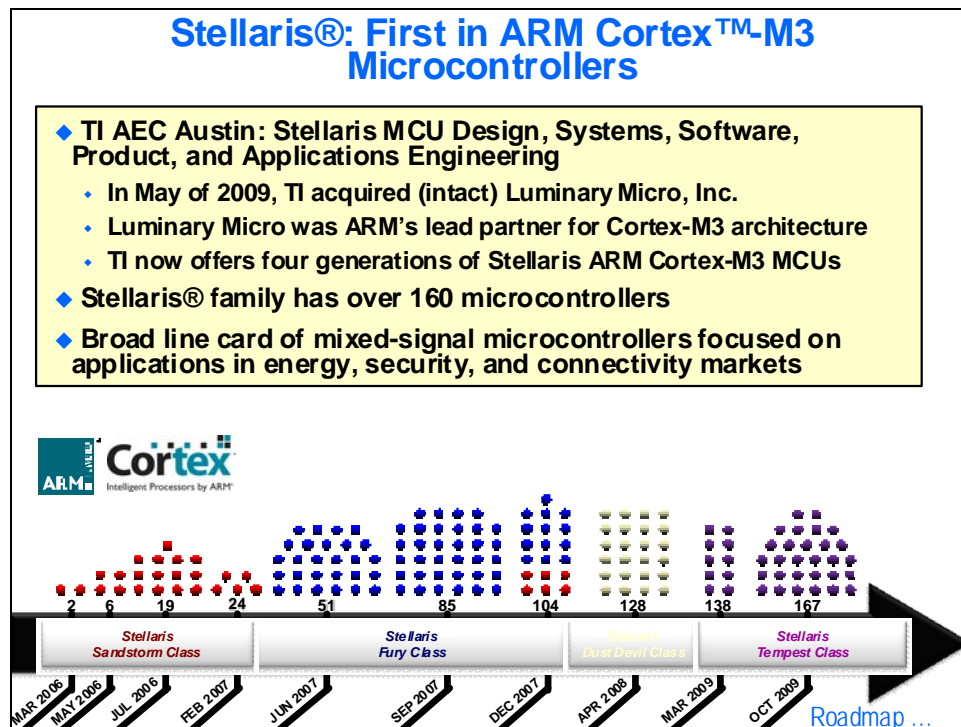
### No Assembly Required

- ◆ **Cortex-M3 has complete hardware support for interrupts**
  - Interrupt Service Routines (ISRs) are 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 come with an ASM boot routine (in object form) that does the 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

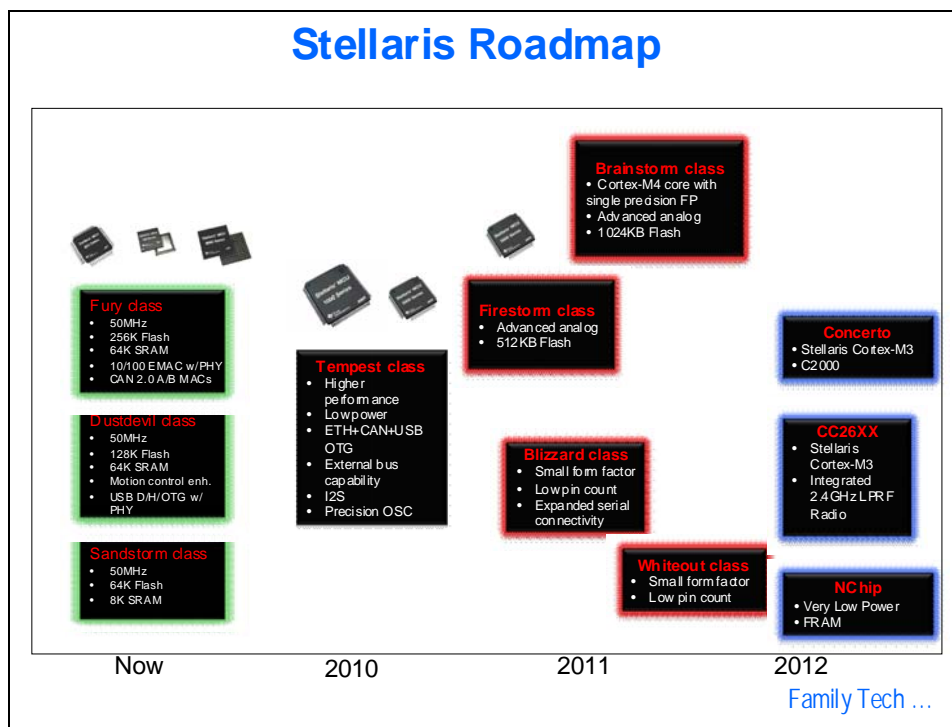



First in ARM ...

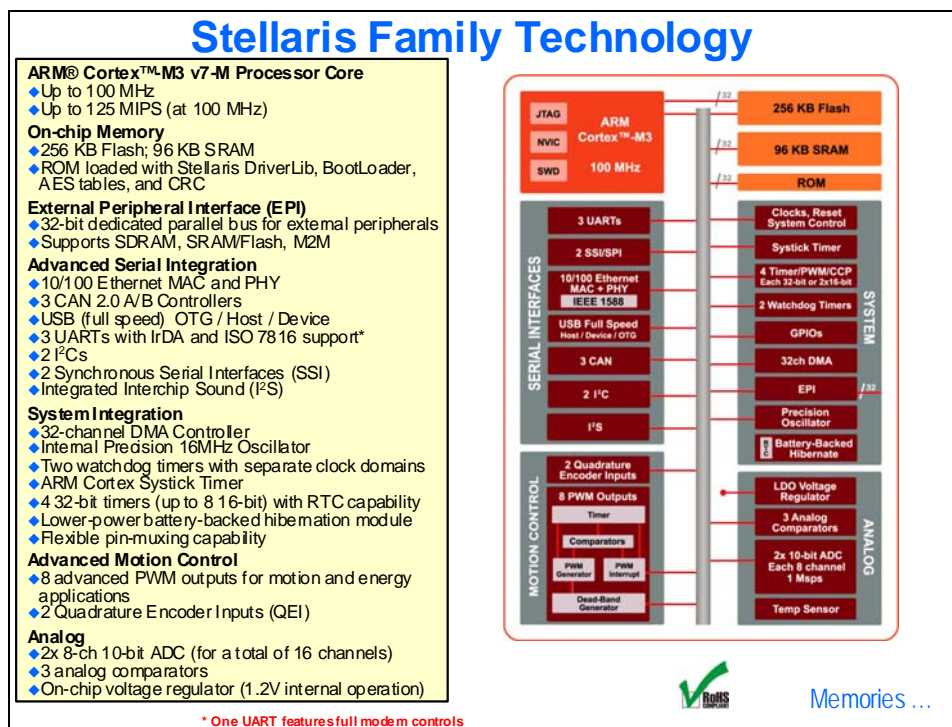
## First in ARM® Cortex™-M3 Microcontrollers



## Roadmap

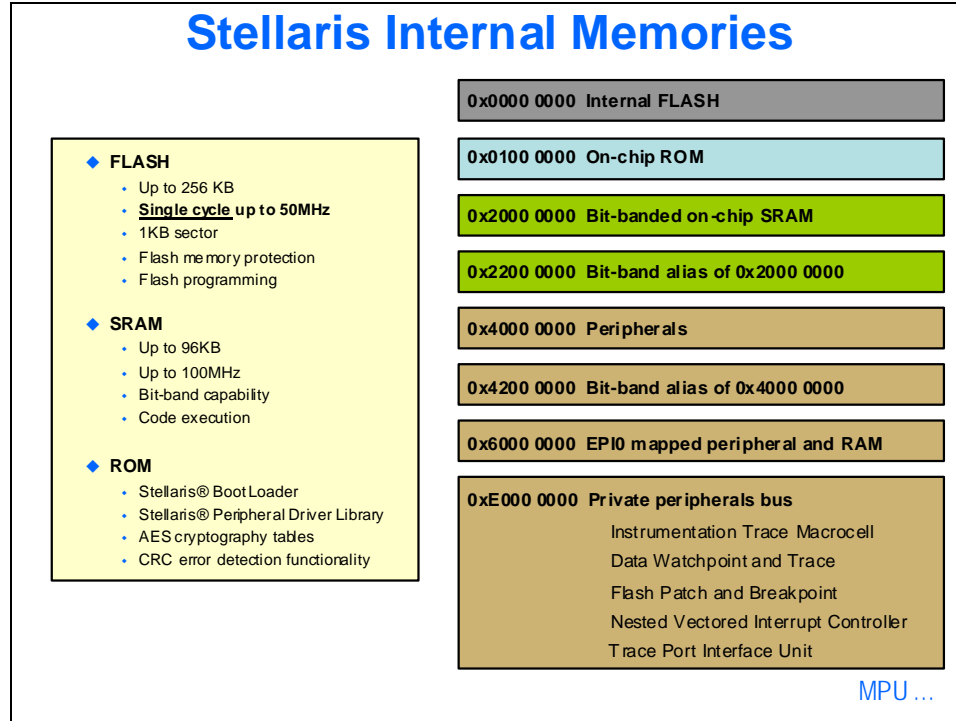


## Family Technology

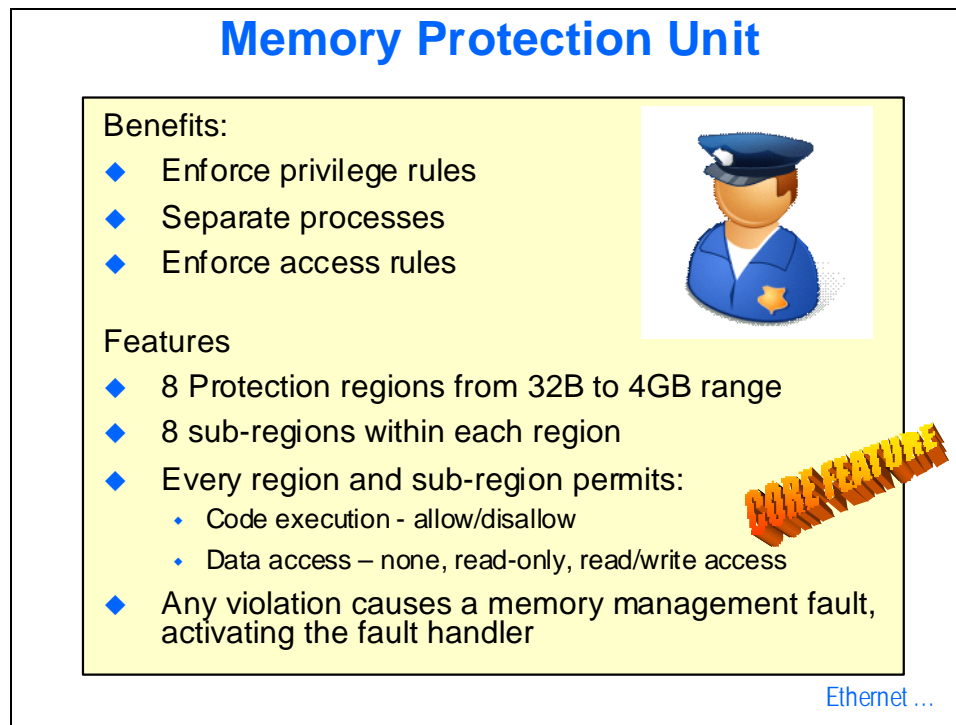




## Internal Memories



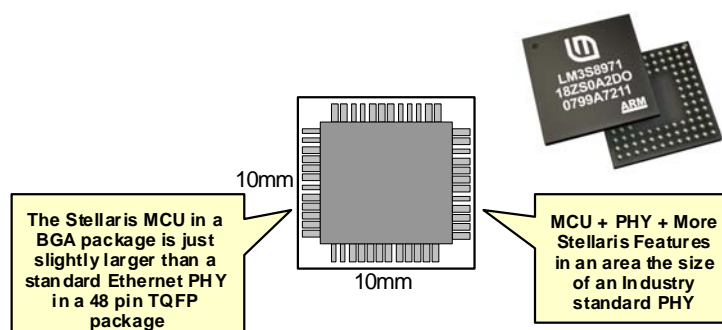
## Memory Protection Unit



## Integrated MAC+PHY

### The Only ARM MCU w/ Integrated 10/100 Ethernet MAC+PHY

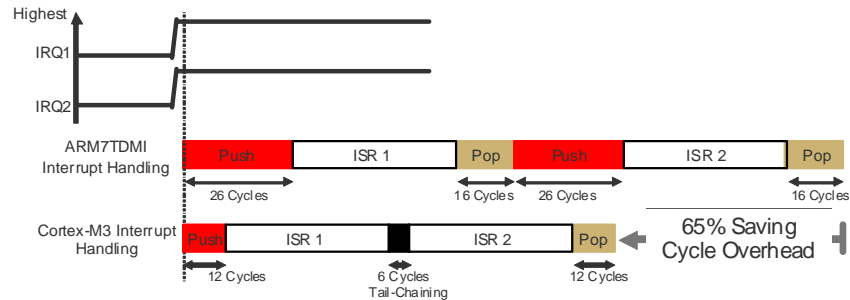
- ◆ Enables network connectivity and embedded web servers
- ◆ Lower external power budget requirements than solutions using an external PHY
- ◆ Savings in board space and system cost
- ◆ Hardware support for Precision Time Protocol (IEEE 1588 PTP)



NVIC ...

# NVIC

## Nested Vectored Interrupt Controller – Tail Chaining



### ARM7TDMI

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

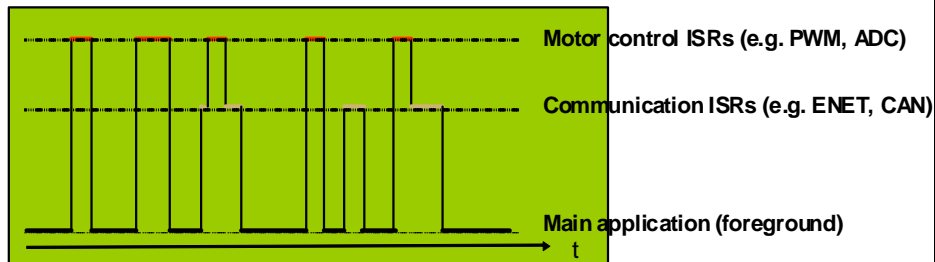
### Cortex-M3

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

LSM = Load/Store Multiple instruction

**CORE FEATURE**  
NVIC ...

## NVIC Interrupt Priorities Example



- ◆ Main application runs as foreground (base level)
  - Easy to write since no “factoring” – 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 motor controls
  - 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

**CORE FEATURE**

EPI ...

## External Peripheral Interface

### 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
  - Similar device compatibility options as PIC, ATmega, 8051, and others
  - Access to SRAM, NOR Flash, and other devices, with up to 24MB of addressing
  - Support of both muxed and de-muxed address and data
  - Access to a range of devices supporting the non-address FIFO x8 interface variant, with support for TXempty and RXfull
  - Speed controlled, with read and write data wait-state counters
  - Manual chip-enable (or use extra address pins)
- ◆ **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)

Hibernation ...

## Battery-Backed Hibernation

### Battery-Backed Hibernation

#### ◆ 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)<br>80 mA (w/ETH) |
| Sleep          | 20 mA           | 20 mA (w/ETH)  | 20 mA            | 8 mA                             |
| Deep Sleep     | 700 µA          | 5 mA (w/ETH)   | 350 µA           | 600 µA                           |
| Hibernate      | —               | 10 to 18 µA    | 10 to 18 µA      | 10 to 18 µA                      |

\* Preliminary

Motor Control ...

## Motor Control

### Motor 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.

- ◆ Up to 4 fault-condition handling inputs in hardware quickly provide low-latency shutdown.

- ◆ Up to 2 Quadrature Encoder Inputs provide accurate positioning for closed-feedback control.



## Evaluation and Reference Design Kits

### Evaluation Kits: “Zero-to-32bits” In 10 Minutes

- Everything a developer needs to get up and running in 10 minutes or less
  - Evaluation board(s)
  - All required cables
  - A choice of evaluation tools suites for popular development tools
  - Documentation
  - StellarisWare software
  - Applications notes
- Each kit functions both as an evaluation platform and as a serial in-circuit debug interface for any Stellaris microcontroller-based target board

|   |   |   |   |   |  |   |   |
|---|---|---|---|---|--|---|---|
|  |  |  |  |  |  |  |  |
| E-K-LM3S811<br>Low pin count<br>\$49  | E-K-LM3S1968<br>High pin count<br>\$59  | E-K-LM3S2965<br>CAN Functionality<br>\$79   | E-K-LM3S3748<br>USB Host/Device<br>\$109  | E-K-LM3S6965<br>Ethernet MAC+PHY<br>\$69  | E-K-LM3S8962<br>Ethernet+CAN<br>\$89   | E-K-LM3S9B90<br>Ethernet+USB OTG<br>\$99  | E-K-LM3S9B92<br>Ethernet+OTG+MC<br>\$99   |

Each kit comes in four versions:



And now ...



Order the kit you like, the other IDEs can be downloaded from  
[www.ti.com/Stellaris](http://www.ti.com/Stellaris)

Ref Des Kits ...

## Motor Control Reference Design Kits

### Open-Tool Motor Control Reference Design Kits



RDK-ACIM  
\$379

#### AC Induction Motor Controller Design

Example applications:

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



RDK-STEPPER  
\$199

#### Stepper Motor Controller Design

Example applications:

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



RDK-BLDC  
\$219

#### Brushless DC Motor Controller with CAN/Ethernet

Example applications:

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



RDK-BDC  
\$219

#### Brush DC Motor Controller with CAN

Example applications:

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







Official FRST KoP Speed  
Controller – FRC 2009

Ref Des Kits ...

## Reference Design Kits







### Open-Tool Reference Design Kits

|  |  |
|--|--|
|  <p><b>RDK-DM</b><br/>\$219</p>     | <p><b>Touch-screen Intelligent Display Module with PoE</b></p> <p><b>Example applications:</b></p> <ul style="list-style-type: none"> <li>• Security Systems &amp; Building Access Controllers</li> <li>• White Goods and other Home Appliances</li> <li>• Factory Automation (System Status and Configuration)</li> </ul>           |
|  <p><b>RDK-DM-L35</b><br/>\$219</p> | <p><b>Landscape-oriented Touch-screen Intelligent Display Module</b></p> <p><b>Example applications:</b></p> <ul style="list-style-type: none"> <li>• Security Systems &amp; Building Access Controllers</li> <li>• White Goods and other Home Appliances</li> <li>• Factory Automation (System Status and Configuration)</li> </ul> |
|  <p><b>RDK-DM-SBC</b><br/>\$299</p> | <p><b>Stellaris 3.5" Landscape IDM Single Board Computer</b></p> <p><b>Example applications:</b></p> <ul style="list-style-type: none"> <li>• Security Systems &amp; Building Access Controllers</li> <li>• White Goods and other Home Appliances</li> <li>• Factory Automation (System Status and Configuration)</li> </ul>         |
|  <p><b>RDK-S2E</b><br/>\$139</p>    | <p><b>Tiny Footprint Serial-to-Ethernet Module</b></p> <p><b>Example applications:</b></p> <ul style="list-style-type: none"> <li>• SCADA Remote Terminal Units (RTUs)</li> <li>• Electronic Flow Meters (EFMs)</li> <li>• CCTV RS-232 Recorders</li> </ul>  |

Modules ...

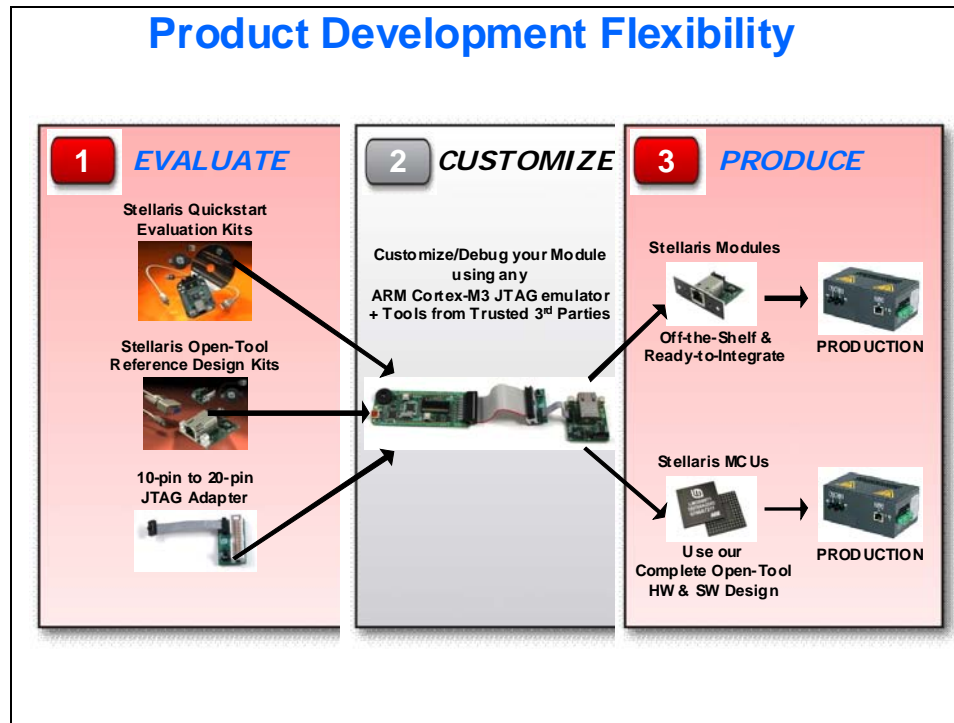
## Modules

### Open-Tool Modules Speed Time-to-Market

|   |   |   |
|---|---|---|
|  <p><b>Brush DC Motor Control</b><br/><i>MDL-BDC</i><br/>Single unit: 109 USD</p>    |  <p><b>Intelligent Display Module</b><br/><i>MDL-IDM-L35</i><br/>Single unit: 185 USD<br/>...with Ethernet<br/><i>MDL-IDM28</i><br/>Single unit: 185 USD<br/>...with PoE<br/><i>MDL-IDM</i><br/>Single unit: 199 USD</p> |  <p><b>Ethernet+CAN BLDC Motor Controller</b><br/><i>MDL-BLDC</i><br/>Single unit: 149 USD</p> |
|  <p><b>STEPPER Motor Control</b><br/><i>MDL-STEPPER</i><br/>Single unit: 169 USD</p> |  <p><b>Serial-to-Ethernet</b><br/><i>MDL-S2E</i><br/>Single unit: 49 USD</p>   |  <p><b>AC Induction Motor Control</b><br/><i>MDL-ACIM</i><br/>Single unit: 239 USD</p>         |

Flexibility ...






## Product Development Flexibility





# Development Tools and Support Software

## Development Tools for Stellaris MCUs

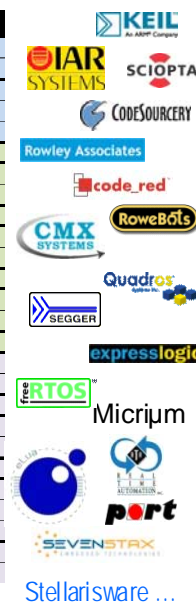
|                  |  |  |  |  |  |
|------------------|---|---|---|---|---|
| Eval Kit License | 30-day full function. Upgradeable   | 32KB address-limited. Upgradeable   | 32KB address-limited. Upgradeable   | 90-day full function. Upgradeable   | Full function. Onboard emulation limited  |
| Compiler         | 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) = €000 (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   |

Partners ...

## Stellaris Partners

## 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++ Compiler, 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)         |
|                     | Quadraxis      | 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                        |
|                     | Interneche     | NicheLite and ARM Network Evaluation Kits                                    |
|                     | Micrium        | µC/USB Device, µC/USB Host, µC/TCP-IP, µC/Modbus, µC/CAN protocol stacks     |
|                     | MicroDigital   | smxUSB Device, smxUSB Host, and smxUSB On-The-Go (OTG) Stacks                |
|                     | port GmbH      | CANopen Library for Stellaris Microcontrollers                               |
|                     | Quadraxis      | 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                               |



Stellarisware ...

## StellarisWare

# StellarisWare®

**License-free and Royalty-free source code for TI Cortex-M3 devices:**

- Peripheral Driver Library
- Graphics Library
- USB Library
- Boot Loader
- IEC 60730 Library
- Flash Programming
- On-Chip ROM Enhancements

On-line ...

## Available On-Line

# StellarisWare®

## Available On-Line

The screenshot shows the StellarisWare website interface. The main navigation bar includes links for Products, Applications, Design Support, and Sample & Buy. The 'StellarisWare & Code Examples' link is circled in red. Below the navigation bar, the 'Microcontrollers (MCU)' section is visible, with a 'Stellaris ARM Cortex-M3-based MCUs' link also circled in red. The 'Complete Listing of StellarisWare Software' section is shown, with a 'StellarisWare Software' link circled in red. The 'StellarisWare Complete (all boards, all components)' status is shown as 'ACTIVE'.

Driver Lib ...

## Peripheral Driver Library

### StellarisWare® Peripheral Driver Library

- ◆ 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, CCS and GNU tools
- ◆ Peripheral driver library functions are preprogrammed in ROM on select Stellaris MCUs



GrLib ...

## Graphics Library

### 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
  - *frasterize*: render your own font to be recognized by GraphicsLib
  - *lmi-button*: predefined button shape with shadow and 3-D
  - *pnmto*: Convert a NetPBM image file into a format recognized by GraphicsLib



USLib ...

## USB Library

### StellarisWare® USB Library Stacks and Examples

◆ **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

IEC60730 ...

## IEC 60730

### StellarisWare® Safe At Home With IEC 60730

**IEC** 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   |
|--|--|---|
| <b>StellarisWare™ Software</b>         | 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.  |
|  | GPIO Test  | Performs ADC temperature sensor test.   |
|  | Clock/Interrupt Test   | Performs GPIO input/output plausibility test.   |
| <b>Stellaris® Hardware</b>             | Nested Vector Interrupt Controller                                     | Performs tests to check the clock frequency, interrupt handling, and execution.   |
|  | Automotive-grade Flash Memory  | Deterministic, fast interrupt processing for execution certainty.   |
|  | Cyclical Redundancy Check in ROM                                       | High reliability non-volatile memory for robust environments.   |
|  | 2 Watchdog Timers  | Especially useful in verifying the contents of memory in a Stellaris microcontroller.   |
|  | Precision Oscillator   | Clocked with precision oscillator, a second WDT takes advantage of the non-maskable interrupt (NMI) handler safety feature of the ARM Cortex-M3 processor.              |
|  | Advanced Motion Control with Multiple Fault Conditioning Inputs        | Supplies an accurate, independent time base when periodic safety tests are executed.  |
|  | Quadrature Encoder Inputs  | Provides quick motor shutdown in low latency situations.  |
|  | Integrated Analog Comparators  | Provides precise, closed loop control of motors.  |
|  | Internal Temperature Sensor  | 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. |
|  | 10/100 Ethernet MAC/PHY with IEEE 1588 PTP                             | Eliminates the performance-wasting requirement of constant CPU polling.   |
| Controller Area Network (CAN) 2.0 MACs | Used to monitor and shut down an appliance if the appliance overheats. |   |

**Note:** Watchdog timers are completely independent hardware timers In System Programming ...

## In System Programming

### 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)
- ◆ See application note [AN01242](#)

#### 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<sup>2</sup>C, SSI, Ethernet, USB
- ◆ Included in the Stellaris Peripheral Driver Library with full applications examples
- ◆ Preloaded in ROM on select Stellaris Microcontrollers

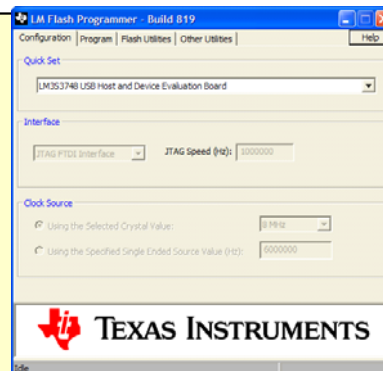
Flash GUI ...

## Flash Programming GUI

### StellarisWare® 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 online
  - <http://focus.ti.com/mcu/docs/mcuorphan.tsp?contentId=87903>



ROM ...

## ROM Enhancements


### StellarisWare® On-Chip Software 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

**Other flash memory-saving options**

- ◆ Advanced Encryption Standard (AES) cryptographic tables
  - Supported by the current AES example application
  - 128, 192 and 256-bits
- ◆ Cyclic Redundancy Check (CRC) functionality – for error detection



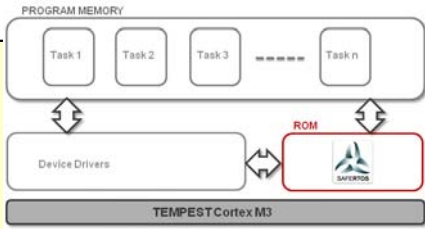
Stored in ROM on select Stellaris MCUs

SAFERTOS ...

## SAFERTOS

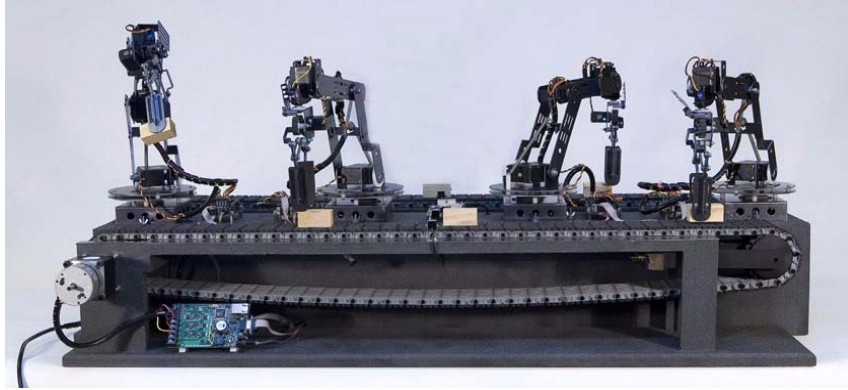
### SAFERTOS Included On The LM3S9B96

- ◆ 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



# Product Demonstrations

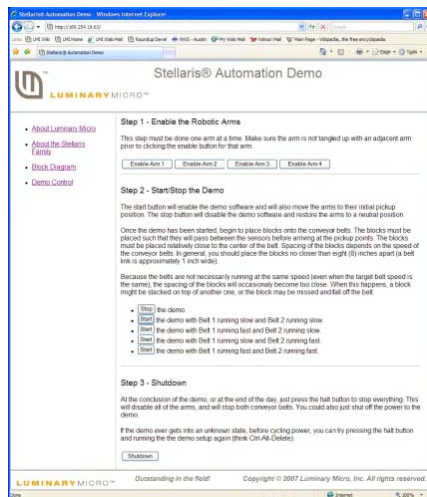
## Stellaris CAN/Ethernet Automation Demo



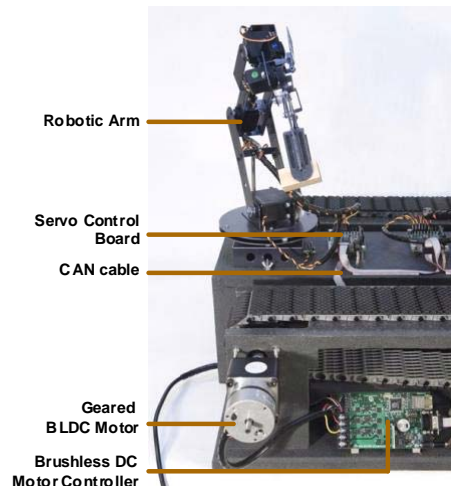
<http://www.youtube.com/watch?v=RyeUMx5cwSM>

## CAN/Ethernet Demo

## Stellaris CAN/Ethernet Automation Demo



Web Browser Console for Automation System Demo



Robotic Arm

Servo Control Board

CAN cable

Geared BLDC Motor

Brushless DC Motor Controller

<http://www.youtube.com/watch?v=RyeUMx5cwSM>

CNC Demo ...



## CNC Machine Demo

### Stellaris 3-axis CNC Machine (AN01246)



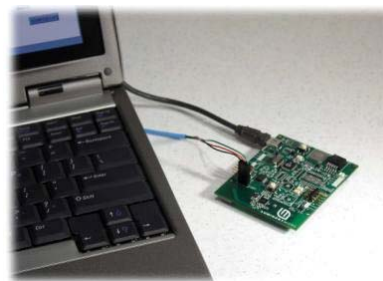
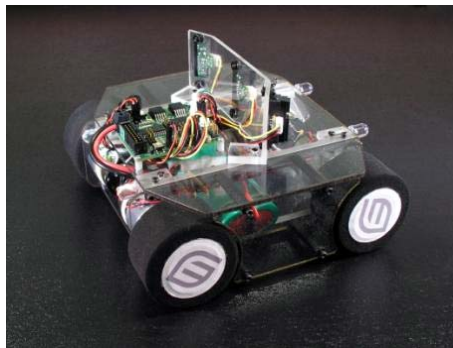
- ◆ **LM3S615 controls all three axes of stepper motion**
  - 6 advanced motion-control PWMs
  - Current sensing
  - Six limit switches
  - Active Indicator
  - Driver for tool control signals
  - Connectivity
  - CNC = Computer Numerical Control
- ◆ **LM3S316 controls QVGA LCD Touch Panel**

<http://www.youtube.com/watch?v=W8FpEJ5ZIY>

Car Demo ...

## Autonomous Car

### Stellaris Autonomous Car (AN01245)



- ◆ **Robot Uses one LM3S316 Stellaris MCU**
  - Four advanced motion-control PWMs drive four brushed motors
  - Four ADC channels for three infrared sensors and a bridge current monitor
  - Analog Comparator for photocell "nighttime" sensor
  - GPIOs for LED headlights
  - SPI for connection to 802.15.4 radio connection


<http://www.youtube.com/watch?v=M-7C7TIYJ8I>


FIRST ...



## FIRST Robotics

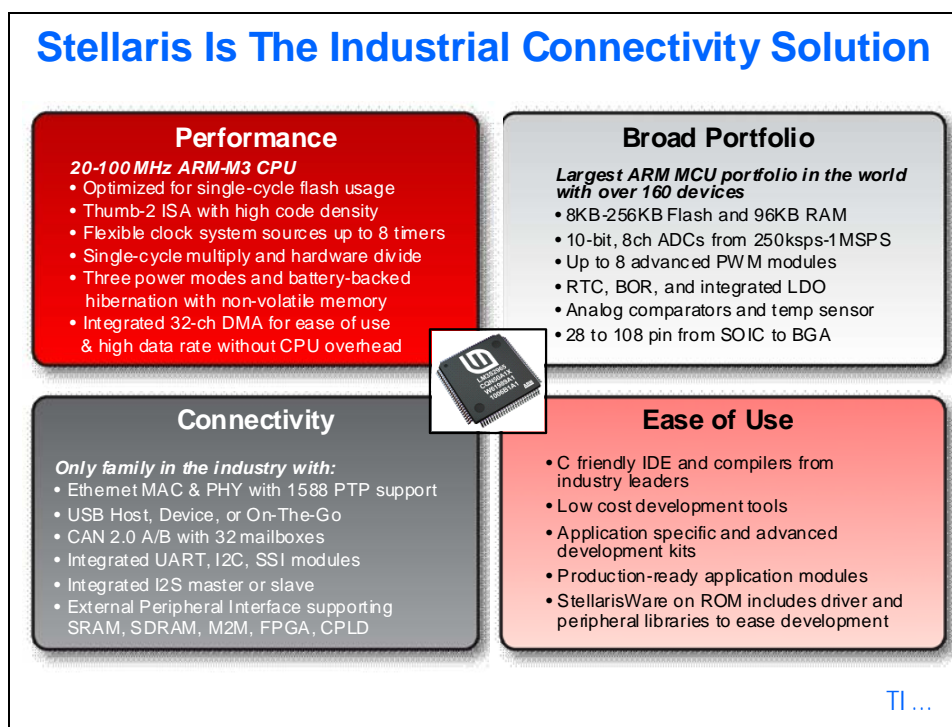
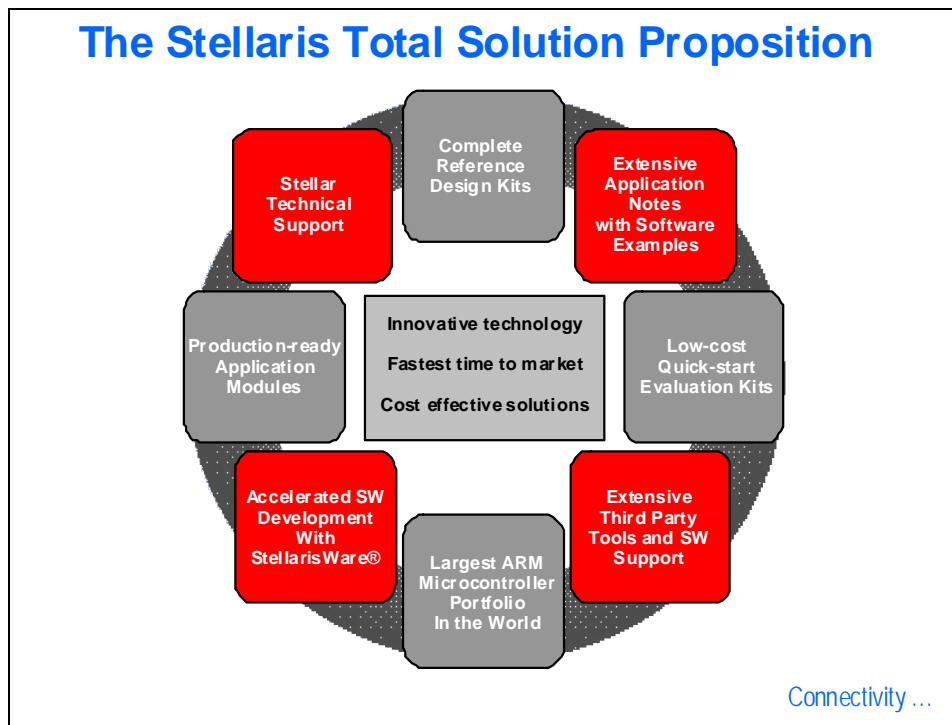
### FIRST Robotics Competition





- ◆ **FRC** is a worldwide robotics competition for high school students.
- ◆ FRC 2009 featured 1700 international teams.
- ◆ Stellaris MDL-BDC “Jaguar” selected as the Official Supplier of the speed controller in the FRC 2009 Kit-of-Parts delivered to each FRC 2009 team.

## Summary



## Introduction

In this section we'll have a chance to get hands-on with the LM3S3748 and LM3S8962 evaluation kits. So let's take a quick look at how these boards operate.

## Objectives

- **Code Composer IDE QuickStart**
- **Flash Programmer**
- **LM3S8962 README FIRST**
- **LM3S3748 README FIRST**

## Module Topics

|  |            |
|--|------------|
| <b>Quickstart Labs .....</b>           | <b>2-1</b> |
| <i>Module Topics.....</i>              | <i>2-2</i> |
| <i>LM3S3748 Evaluation Kit.....</i>    | <i>2-3</i> |
| Oscilloscope Application.....          | 2-3        |
| <i>LM3S8962 Evaluation Kit.....</i>    | <i>2-4</i> |
| Web Server Maze Game Application ..... | 2-4        |
| <i>Code Composer Studio 4.1.....</i>   | <i>2-5</i> |
| Lab Procedure.....                     | 2-5        |

# LM3S3748 Evaluation Kit

## LM3S3748 Evaluation Kit

- ◆ 50MHz LM3S3748 w/ 128K Flash & 64K SRAM
- ◆ Host and Device USB Connectors
- ◆ Bus or Self-powered USB Device support
- ◆ 128x128 Color LCD display
- ◆ microSD card slot
- ◆ Speaker with amplifier
- ◆ USB Flash memory stick
- ◆ USB Debugger interface
- ◆ Oscilloscope QuickStart application



[OS App ...](#)

## Oscilloscope Application

### LM3S3748 Evaluation Kit QuickStart App - Oscilloscope Demonstration -



Oscilloscope Wiring

Oscilloscope Demo

Oscilloscope Options



USB Host Mode  
Data dump to the memory stick



USB Device Mode  
Control the demo via a PC

[8962 ...](#)

## LM3S8962 Evaluation Kit

### LM3S8962 Ethernet+CAN Evaluation Kit

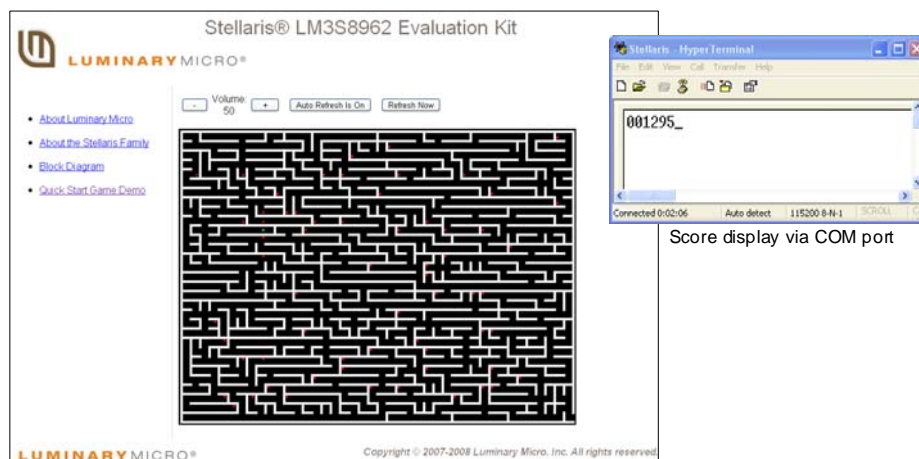
- ◆ LM3S8962 w/ Integrated 10/100 Ethernet controller and CAN MAC
- ◆ OLED graphics display
- ◆ Speaker
- ◆ microSD card slot
- ◆ USB Debugger interface
- ◆ Maze Game QuickStart application



OS App ...

## Web Server Maze Game Application

### QuickStart App for LM3S8962 Evaluation Kit – Web Server Maze Game -



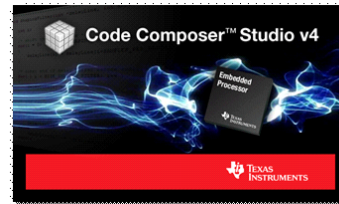
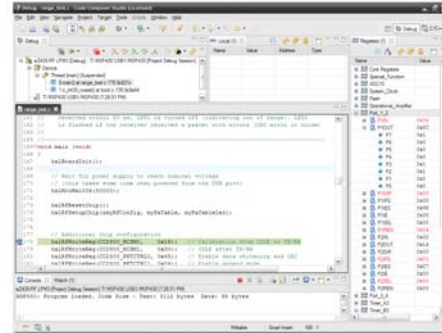
Game display on OLED and web browser

CCS ...

# Code Composer Studio 4.1

## Code Composer Studio v4.1

- ◆ **Code Composer Studio v4.1**
  - Platinum version supports Stellaris
  - Purchase MCU license for \$495
- ◆ **Can be used FREE with EVMs**
- ◆ **120 day full version eval period**
- ◆ **Eclipse based**
- ◆ **Includes**
  - Debugger
  - Profiler
  - Scripting
  - Image analysis and Visualization
  - C/C++ Compiler
  - Simulator



Procedure ...

## Lab Procedure

### Workshop Lab Procedure

- ◆ **If you have a 3748 board:**
  - Run the CCS QuickStart lab (chapter 3)
  - Reprogram the QuickStart App onto your board using the Flash Programmer procedure (chapter 4)
  - Run the 3748 README lab (chapter 5)
- ◆ **If you have a 8962 board:**
  - Run the CCS QuickStart lab (chapter 3)
  - Reprogram the QuickStart App onto your board using the Flash Programmer procedure (chapter 4)
  - Run the 8962 README lab (chapter 6)
- ◆ **Best case ... pair up with a partner who has the other board and see all the labs**

TI ...

\*\*\* Doodle Here \*\*\*



# 03 - QUICKSTART – CODE COMPOSER™ STUDIO

---

## Stellaris® Development and Evaluation Kits for Code Composer™ Studio

The Stellaris Development and Evaluation Kits provide a low-cost way to start designing with Stellaris microcontrollers using Texas Instruments' Code Composer Studio development tools. The evaluation boards can function as either a complete evaluation target or as a debugger interface to any external Stellaris device.

### Requirements

- You have a PC with a USB interface, running Microsoft® Windows XP (SP2 or greater) or Vista
- You have the Workshop Installation Software Flash Drive



**CAUTION:** There is a known electrical issue with the FT2232 device that is used in the on-board In Circuit Debug Interface (ICDI). Some USB hubs can cause the device to misbehave, with symptoms ranging from failed enumeration to corrupt data transfers. If you experience trouble when using the on-board ICDI, try connecting the USB cable directly to one of the USB ports on your PC or laptop.

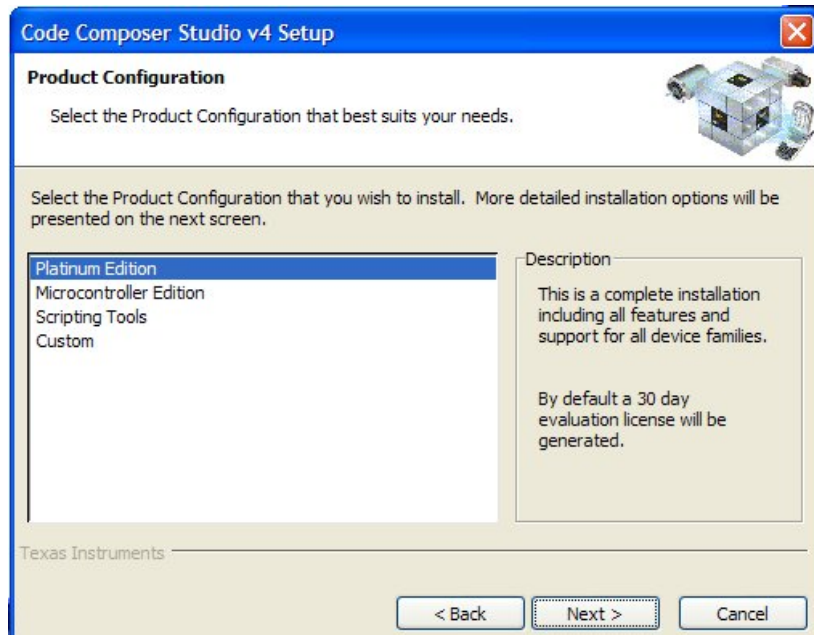
## Code Composer Studio

This quickstart shows you how to install the Code Composer Studio development tool and how to use it to build and run an example application on your Stellaris Evaluation or Development Board.

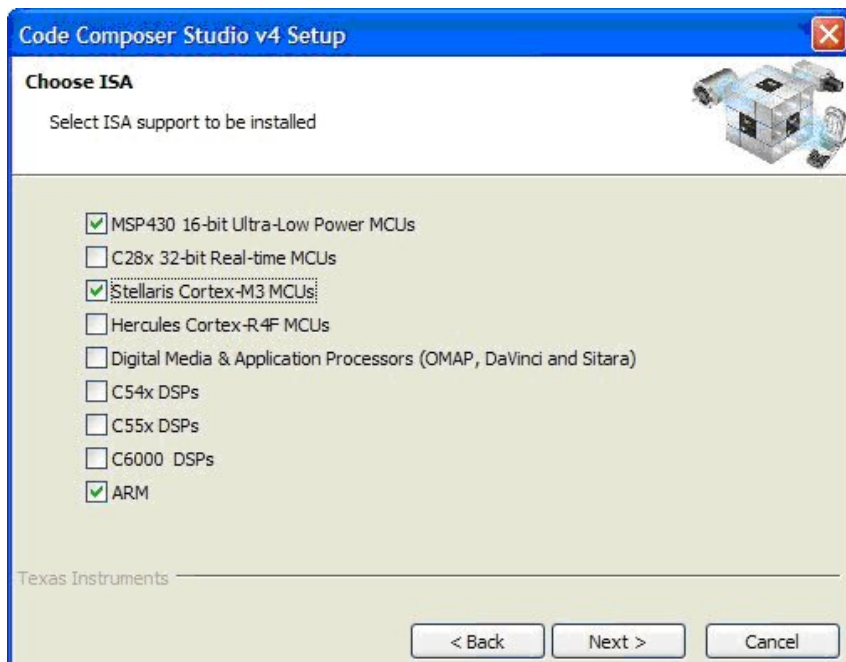
### ***Step 1: Install Code Composer Studio***

1. **Disconnect** any evaluation board that you have connected to your PC's USB port(s). **Insert** the Workshop Installation Flash Drive into a free USB port.
2. Using **Windows Explorer**, find the **setup\_CCS\_4.1** folder on the Flash drive and double-click on the file named **setup\_CCS\_n.n.n.n.exe**.
3. Follow the instructions in the Code Composer Studio installation program. Select the **Platinum Edition** for installation when the **Product Configuration** dialog window appears. Click **Next**.

# QUICKSTART – CODE COMPOSER STUDIO



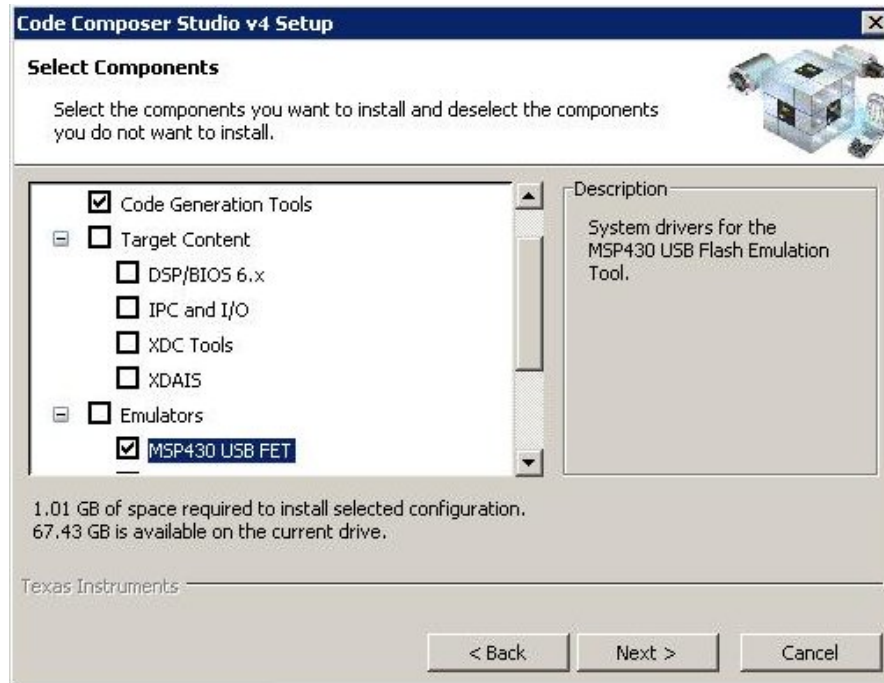
4. In the **Choose ISA** dialog, if you are attending a Stellaris only workshop, make sure that only the **Stellaris Cortex-M3 MCU** and **ARM** checkboxes are selected. If you are also attending an **MSP430** workshop, check that checkbox too. Click **Next**.



5. In the **Select Components** dialog, **uncheck** the **Target Content** and **Emulators** checkboxes. If you are attending a **Stellaris only** workshop, click **Next**. If you are attending a **MSP430** workshop too, check the **MSP430 USB FET** checkbox and click **Next**. The installation should take less than 10 minutes to complete.

# QUICKSTART – CODE COMPOSER STUDIO

---



If you've been tasked with installing Code Composer only, please stop here and ask your instructor for further directions.

# QUICKSTART – CODE COMPOSER STUDIO

---

## **Step 2: Install the StellarisWare® Package**

A full set of C-based peripheral drivers is provided, covering all peripherals and functionality of the Stellaris devices. The StellarisWare package includes various example applications with project files for all major tool vendors that support Stellaris, including Code Composer Studio. To install StellarisWare components, follow these steps:

1. Make sure that the **Workshop Installation Flash Drive** is inserted into one of your PC's USB ports.
2. Using **Windows Explorer**, **open** the Flash drive and find the StellarisWare installation file that matches your board.
  - **LM3S3748 board**    **SW-EK-LM3S3748-xxxx.exe**
  - **LM3S8962 board**    **SW-EK-LM3S8962-xxxx.exe**

**Double-click** on the file for your board and select the **default** installation location when prompted. If you intend to run the labs for **both** boards, you will need the drivers and StellarisWare for **both** boards installed.

If you run a second (or more) StellarisWare installation, use the **default** installation directory. The board files will be installed in separate folders for each board. When you are warned about overwriting files, click **Yes to all**. All the overwritten files are the same.

**NOTE:** Check the [www.ti.com/Stellaris](http://www.ti.com/Stellaris) web site for the latest software updates.

# QUICKSTART – CODE COMPOSER STUDIO

---

## **Step 3: Start Code Composer Studio and Open a Workspace**

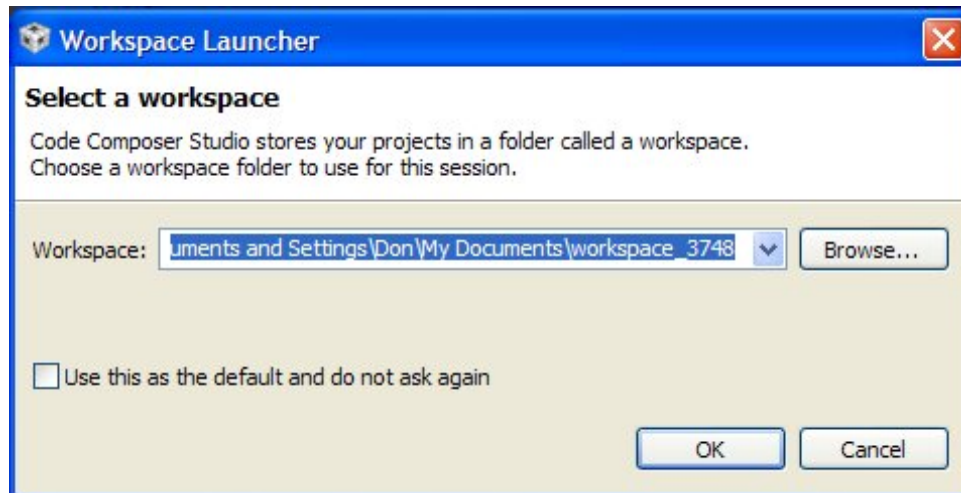
1. **Start** the **Code Composer Studio** IDE by selecting it from the Windows Start menu or double-clicking the icon installed on your desktop.

When the IDE loads, it asks you where to open the workspace folder. To keep your projects separated, you should use a workspace for each board.

If you are using the **LM3S3748** board, name your workspace **workspace\_3748**.

If you are using the **LM3S8962** board, name your workspace **workspace\_8962**.

Do **not** check the **Use this as the default and do not ask again** checkbox. While it's possible to switch back and forth in the IDE, it's best to select your workspace at the start to prevent confusion. Click **OK**.

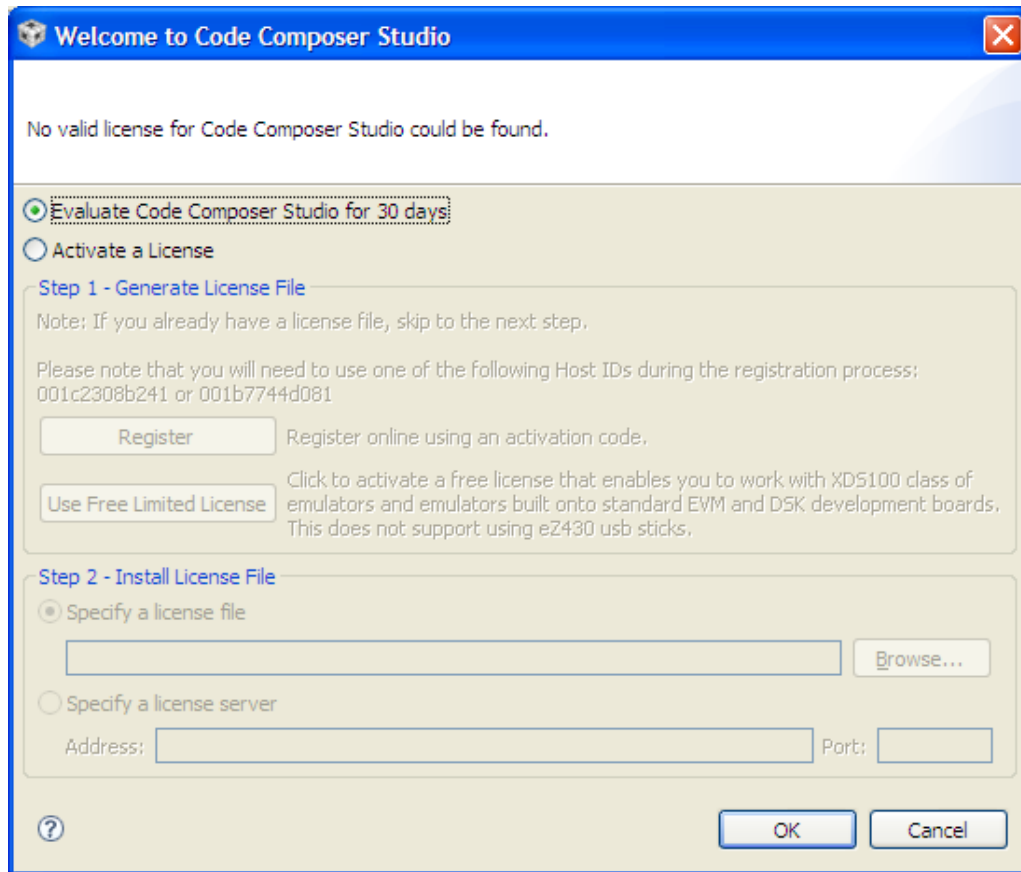


# QUICKSTART – CODE COMPOSER STUDIO

---

2. If this is the first time you have run **Code Composer Studio IDE**, a dialog box may appear like the one shown below. If the dialog appears, select **Evaluate Code Composer Studio for 30 days** unless you already have a license that you wish to activate. Click **OK** to continue.

NOTE: If you have previously installed a Code Composer evaluation disk, you may not be able to move beyond this screen. Ask your instructor for assistance.



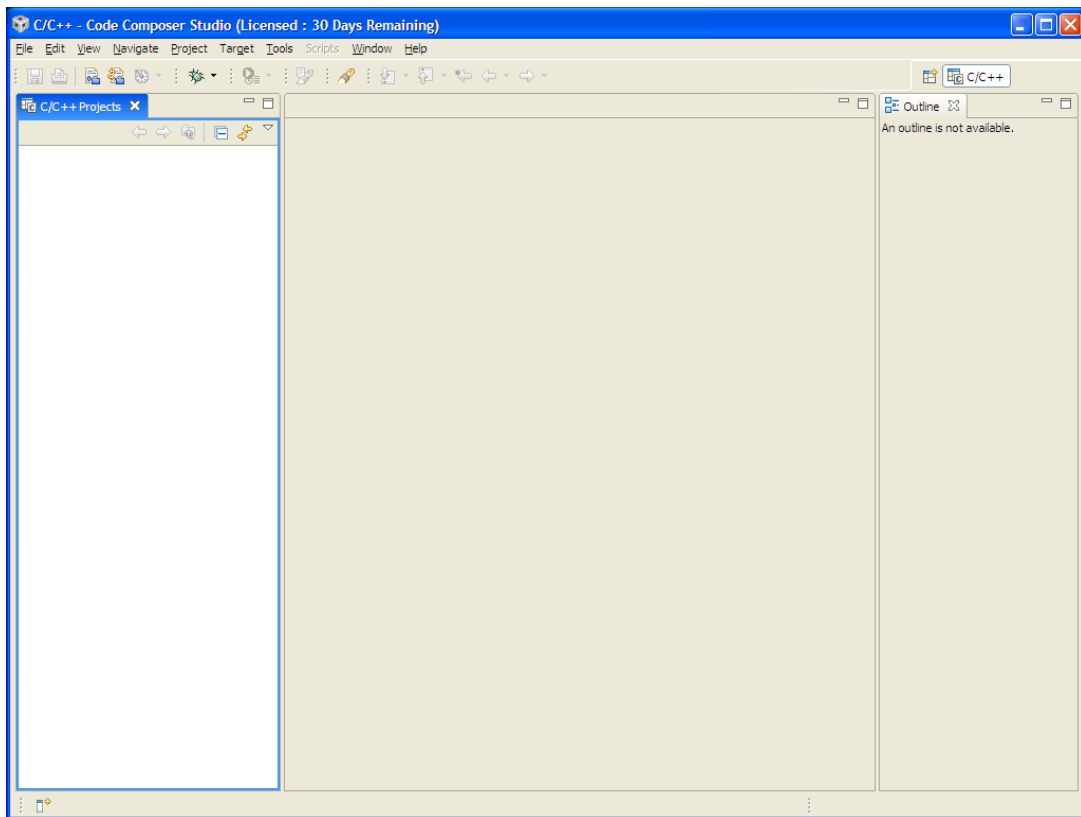
# QUICKSTART – CODE COMPOSER STUDIO

---

3. **Code Composer Studio** may now open with the welcome page. If so, close out the welcome page by clicking the link in the upper right



or by clicking the **X** on the **tab**. You should now have an empty workspace like that shown below. **Maximize** the window.

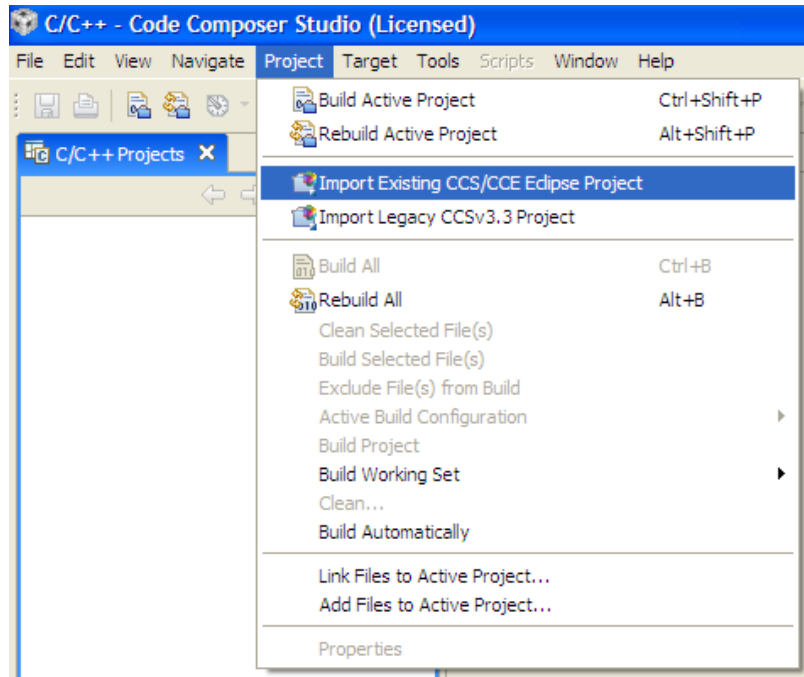


# QUICKSTART – CODE COMPOSER STUDIO

---

## ***Step 4: Import Libraries***

1. From the menu bar, click on **Project**, and then select **Import Existing CCS/CCE Eclipse Project**.

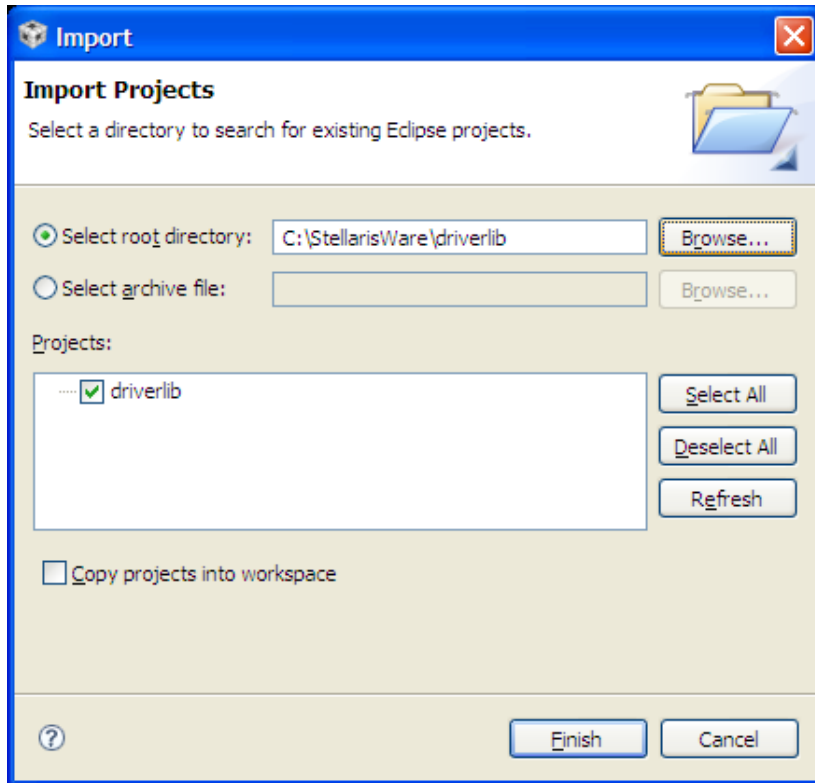




# QUICKSTART – CODE COMPOSER STUDIO

---

2. The **Import** dialog appears. Browse to the root directory of the driver library (C:\StellarisWare\driverlib) and click OK. Be sure that the checkbox next to **driverlib** in the Project pane is **checked** and that **Copy projects into workspace** is **unchecked**. Click **Finish**.

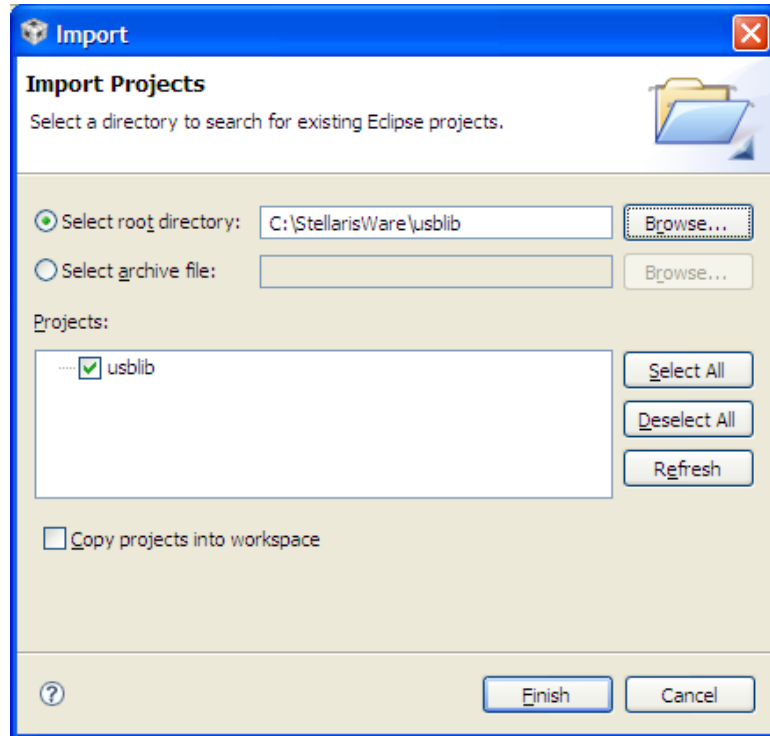


# QUICKSTART – CODE COMPOSER STUDIO

---

## 3. Skip this step if you are using the LM3S8962 board.

Select **Import Existing CCS/CCE Eclipse Project** from the **Project** menu again. Browse to the root directory of the USB library (**C:\StellarisWare\usblib**) and click **OK**. Be sure that the checkbox next to **usblib** in the Project pane is **checked** and that **Copy projects into workspace** is **unchecked**. Click **Finish**.

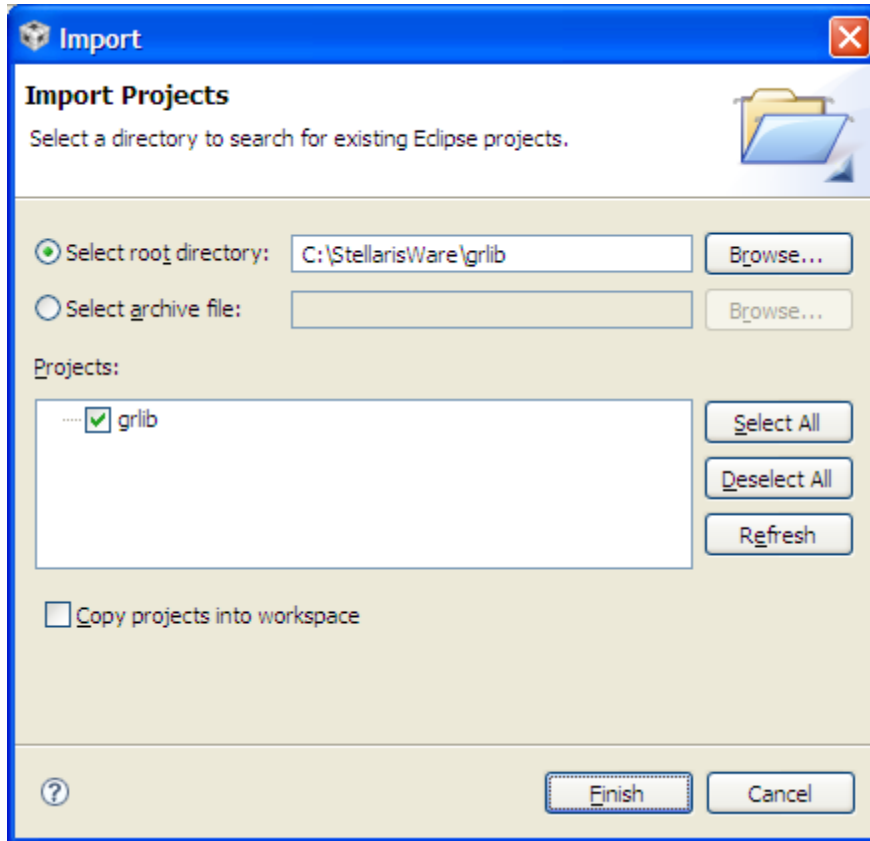


# QUICKSTART – CODE COMPOSER STUDIO

---

## 4. Skip this step if you are using the LM3S8962 board.

Select **Import Existing CCS/CCE Eclipse Project** from the **Project** menu again. Browse to the root directory of the graphics library (C:\StellarisWare\gplib) and click **OK**. Be sure that the checkbox next to **gplib** in the Project pane is **checked** and that **Copy projects into workspace** is **unchecked**. Click **Finish**.



# QUICKSTART – CODE COMPOSER STUDIO

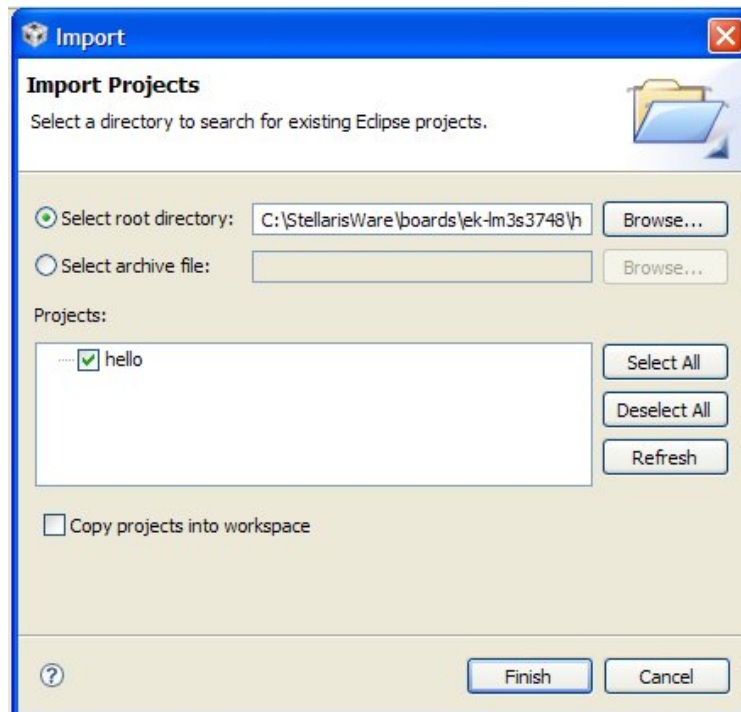
---

## **Step 4: Import Board Example**

5. Select **Import Existing CCS/CCE Eclipse Project** from the **Project** menu again. Browse to the root directory for your chosen board ...
  - **C:\StellarisWare\boards\ek-lm3s3748** for the **3748** board
  - **C:\StellarisWare\boards\ek-lm3s8962** for the **8962** board

Browse to the root directory of the **hello** project and click **OK**.

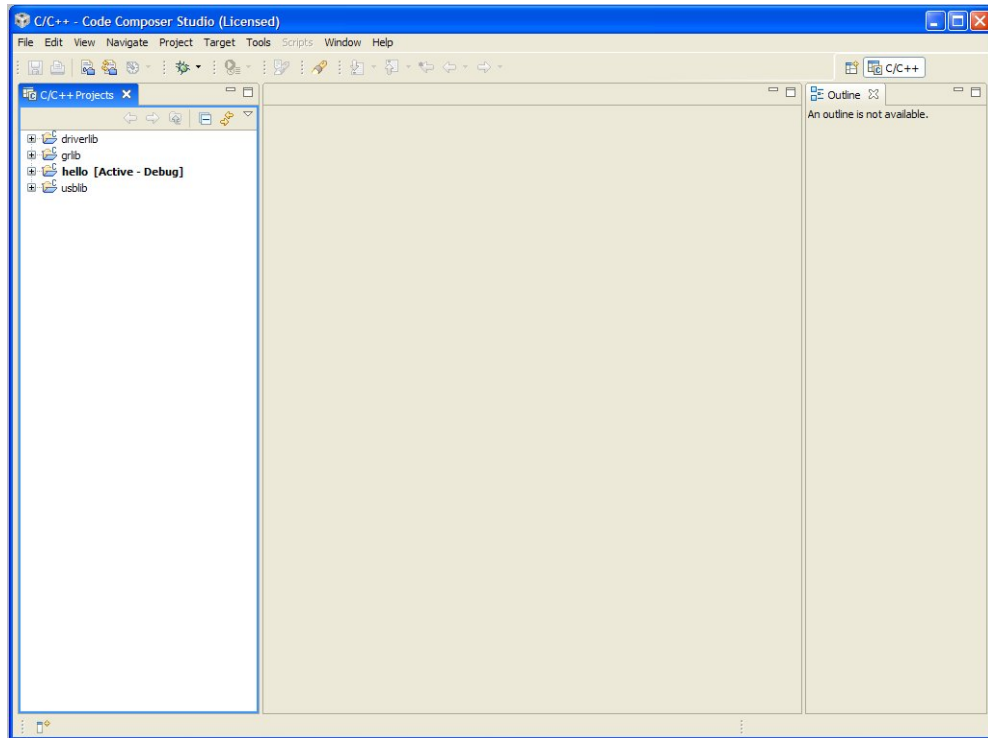
The example screen shot below has the **EK-LM3S3748** board as the chosen board. (**C:\StellarisWare\boards\ek-lm3s3748**). Be sure that the checkbox next to **hello** in the Project pane is **checked** and that **Copy projects into workspace** is **unchecked**. Click **Finish**.



# QUICKSTART – CODE COMPOSER STUDIO

---

6. All of the imported projects now appear in the **Projects Explorer** pane.



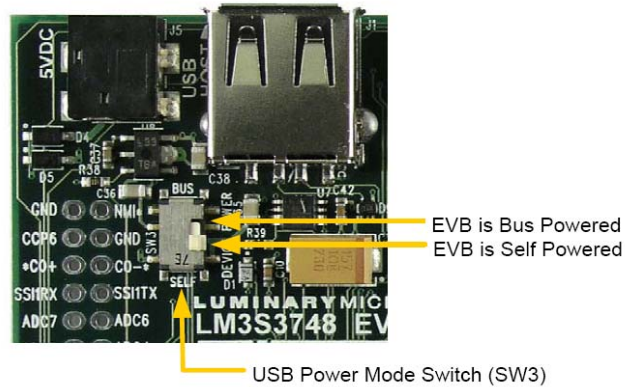
Since the **hello** project was the last one imported, it becomes the **Active Project**. Otherwise, you can right-click on it and set it as the **Active Project**.

# QUICKSTART – CODE COMPOSER STUDIO

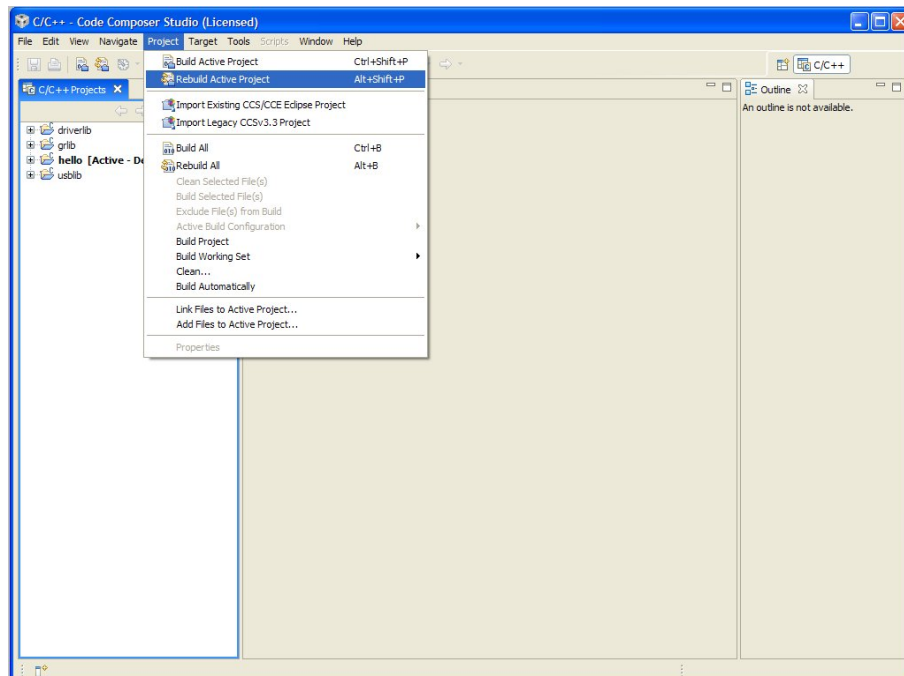
---

## Step 5: Building and Debugging a Project


1. Make sure that the **power switch** on your development board is in the **SELF** position, if your board has a switch. Then connect your development boards **DEBUG USB** port to your laptops USB port. Don't change the position of this switch while the board is powered as you may damage the switch's internal contacts. The **3748** power switch is shown below:

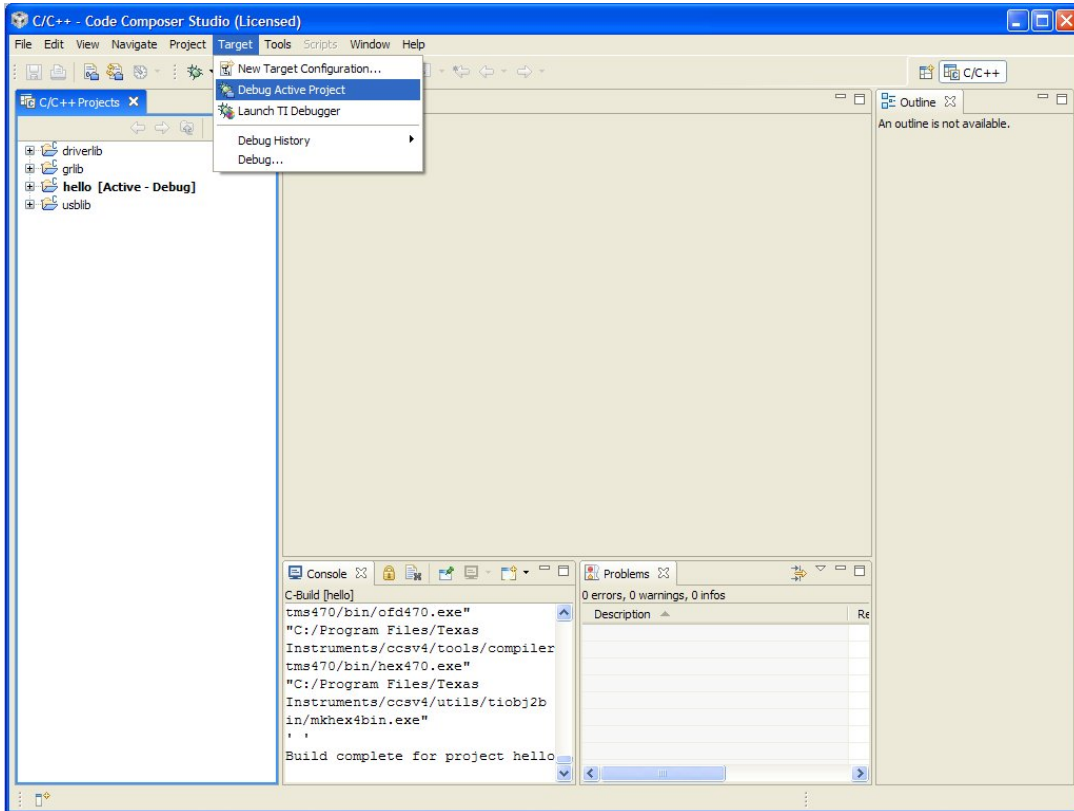



2. On the menu bar, click on **Project** and select **Rebuild Active Project**. The build may take a few moments. As the project builds, messages scroll by in the console window. When the build is complete, the words **Build complete for project hello** will appear in the console window.



# QUICKSTART – CODE COMPOSER STUDIO

3. On the menu bar, click on **Target** and select **Debug Active Project**. Alternatively, you can simply click the **Debug Launch** icon  on the menu bar.




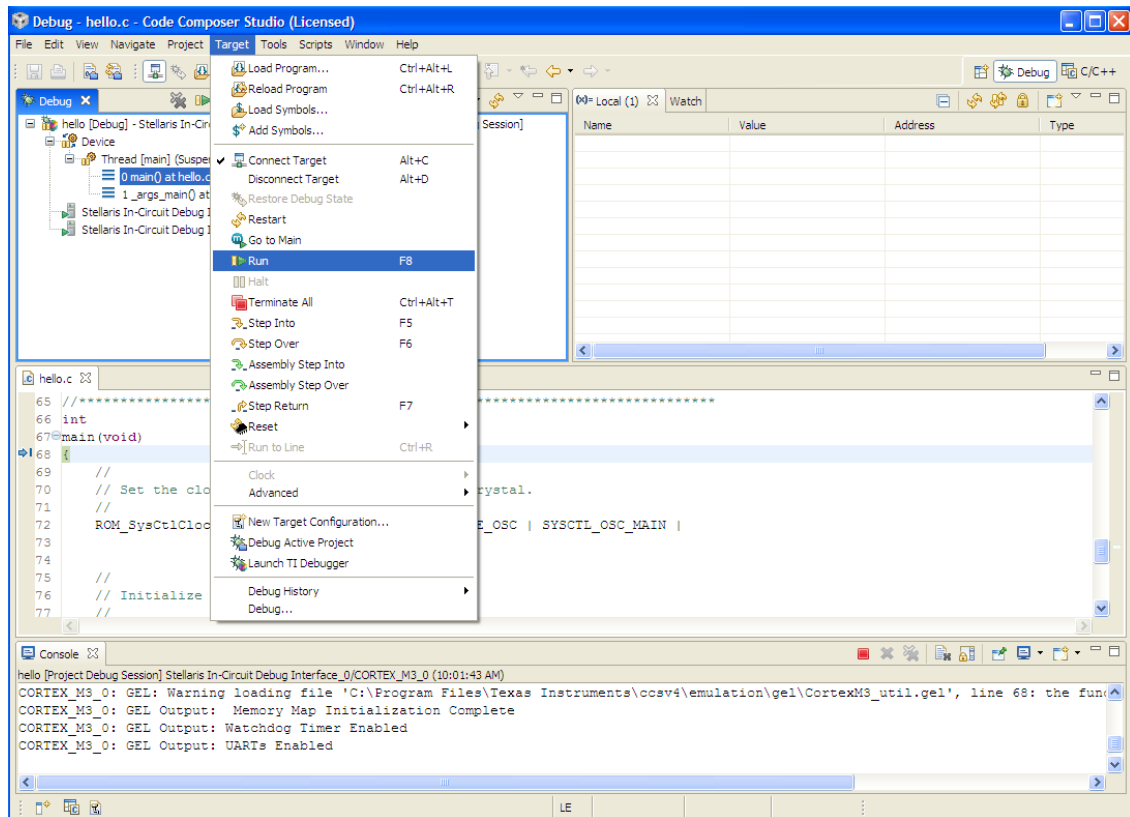
Code Composer Studio should switch to the **Debug Perspective**. If it does not, click the **Debug Perspective** button  in the upper right.

# QUICKSTART – CODE COMPOSER STUDIO

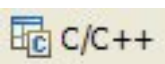
4. The **Code Composer Studio** debugger automatically connects to your evaluation board, programs the Flash memory, and runs to the beginning of the `main()` function. From here, you can examine and modify memory, program variables and processor registers, set breakpoints, step, and perform other typical debugging activities.

To run the program, select **Run** from the **Target** pull-down menu or click the

**Run**  button on the menu bar.



Note the display on your board. If everything has worked properly, you should see **Hello World** on the display. Don't worry about the preprogrammed application that came with the board, we'll re-flash it in the **Flash Programmer** section later.

Click the  **C/C++** **Perspective** button in the upper right of your display to return to the editor perspective.

**Right-click** on **hello** in the **Project** pane and select **Close Project** from the list.

If you are not going to run the next lab section, **Close** Code Composer Studio now.



# QUICKSTART – CODE COMPOSER STUDIO

---

## Optional!

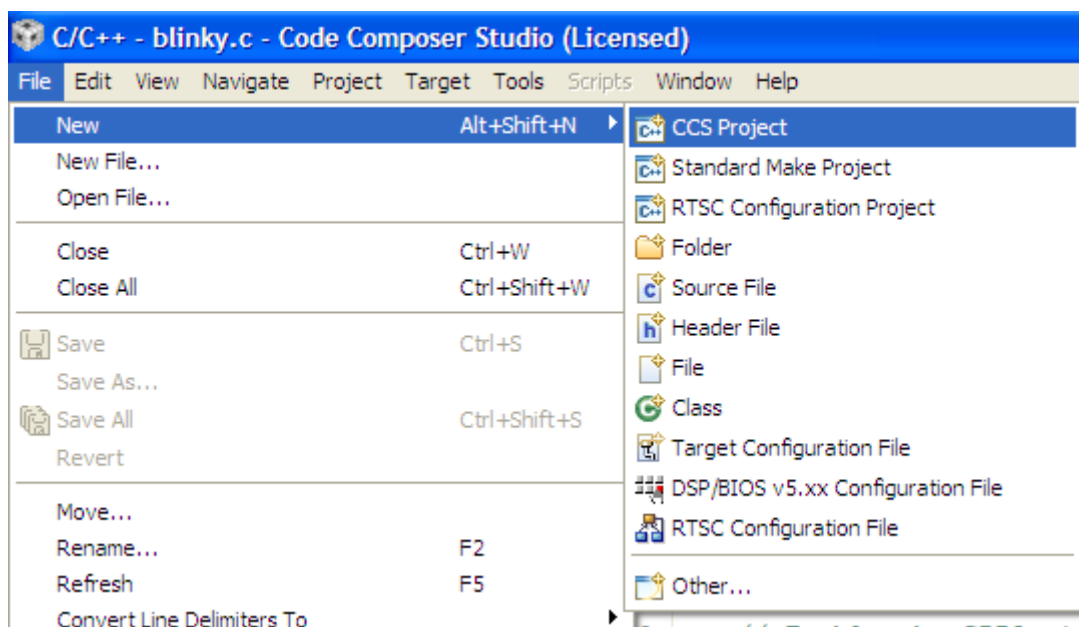
The following section should only be attempted if you have sufficient time to complete it. Let your instructor know that you're ready to start this section. Otherwise, you can complete it at home.

## Creating a New Project

Once you have gone through the StellarisWare example applications, you may want to create your own project to begin development. While you can always start with an existing, simple project, sometimes you may want to start fresh.

The example provided below creates a fresh project, copies code from an existing project, and builds the new project.

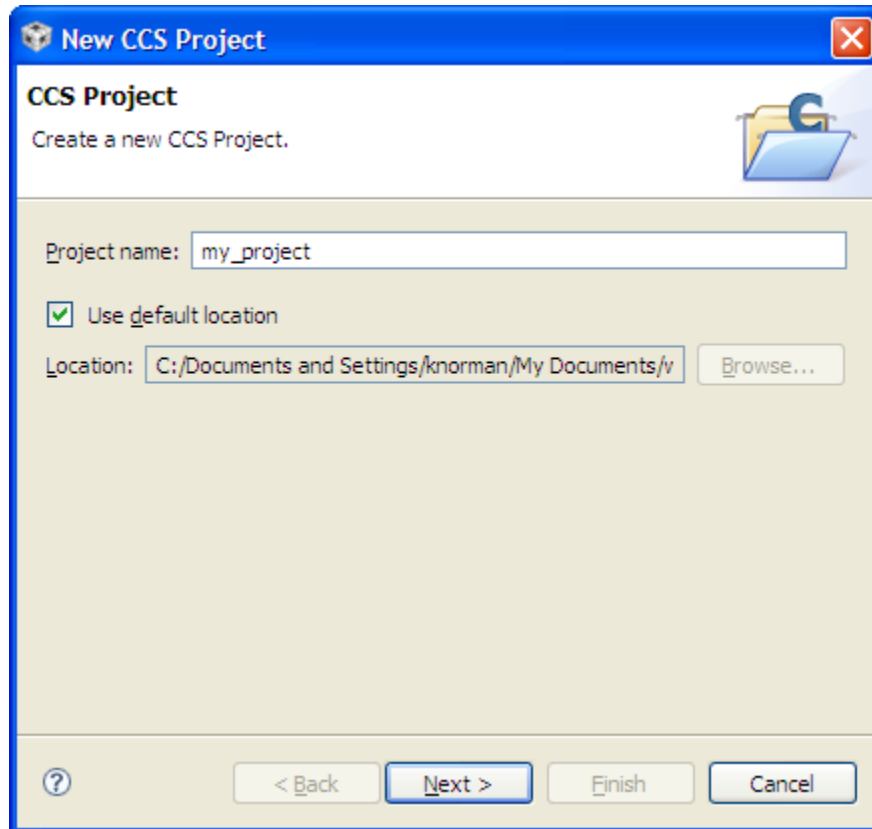
1. To add a new project to your workspace, go to **File → New → CCS Project**.



# QUICKSTART – CODE COMPOSER STUDIO

---

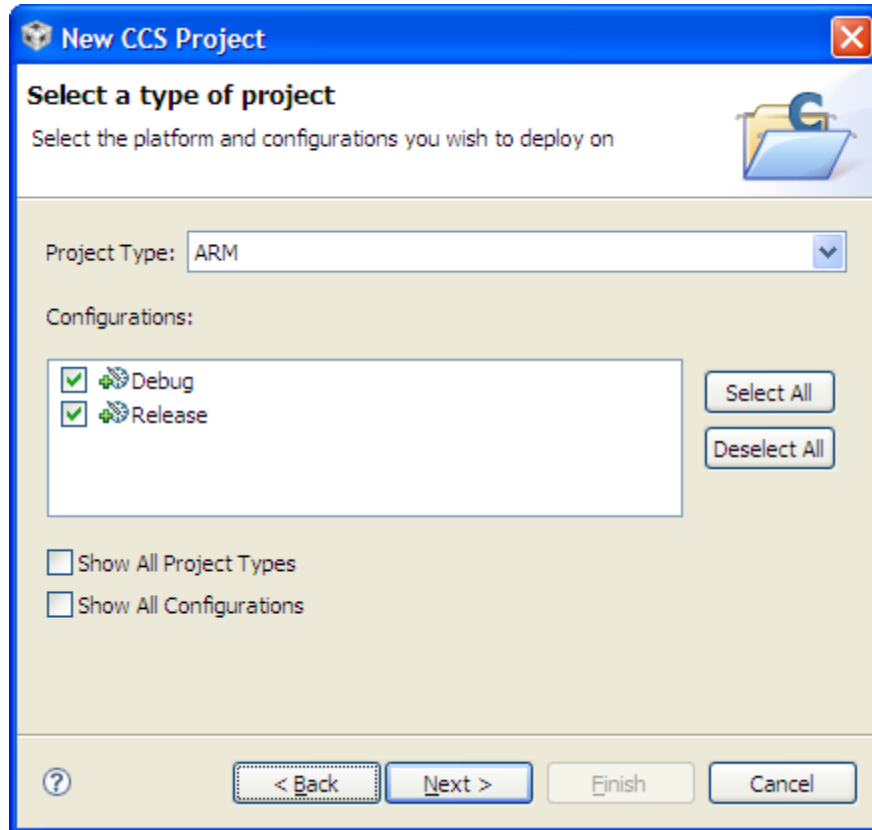
2. **Code Composer Studio** prompts you with a dialog asking you to name the project (how about **my\_project**) and specify the location of the project. Name your project, specify the location (your default workspace will do), and click **Next**.



# QUICKSTART – CODE COMPOSER STUDIO

---

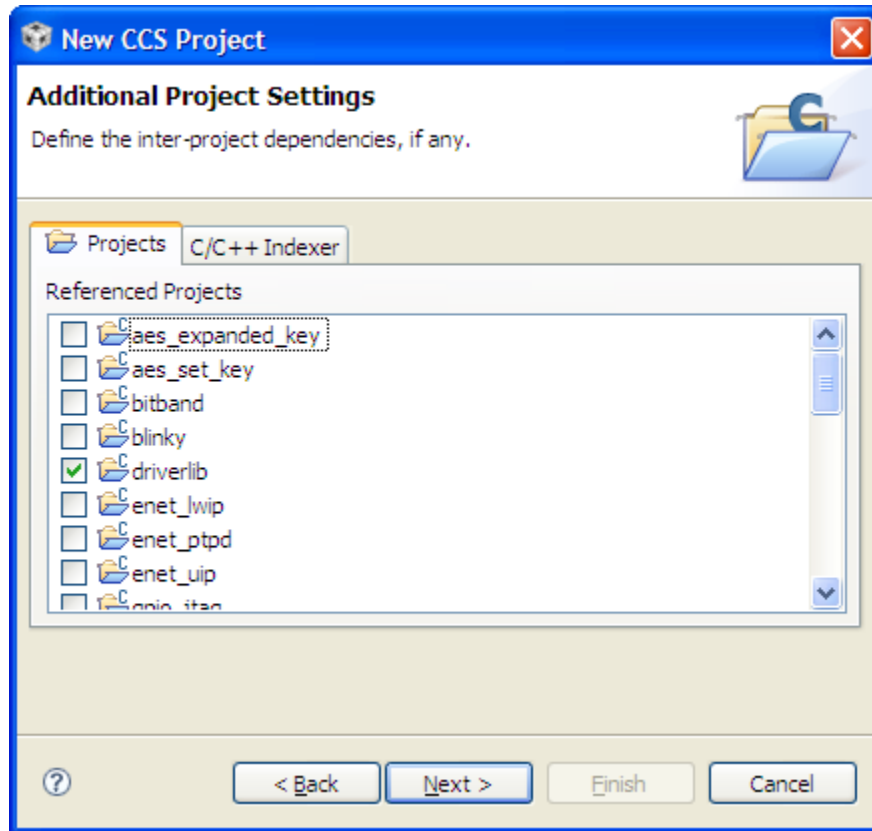
3. The next dialog asks for the project type and configurations. Select **ARM** as the project type, check both the **Debug** and **Release** configurations, and click **Next**.



# QUICKSTART – CODE COMPOSER STUDIO

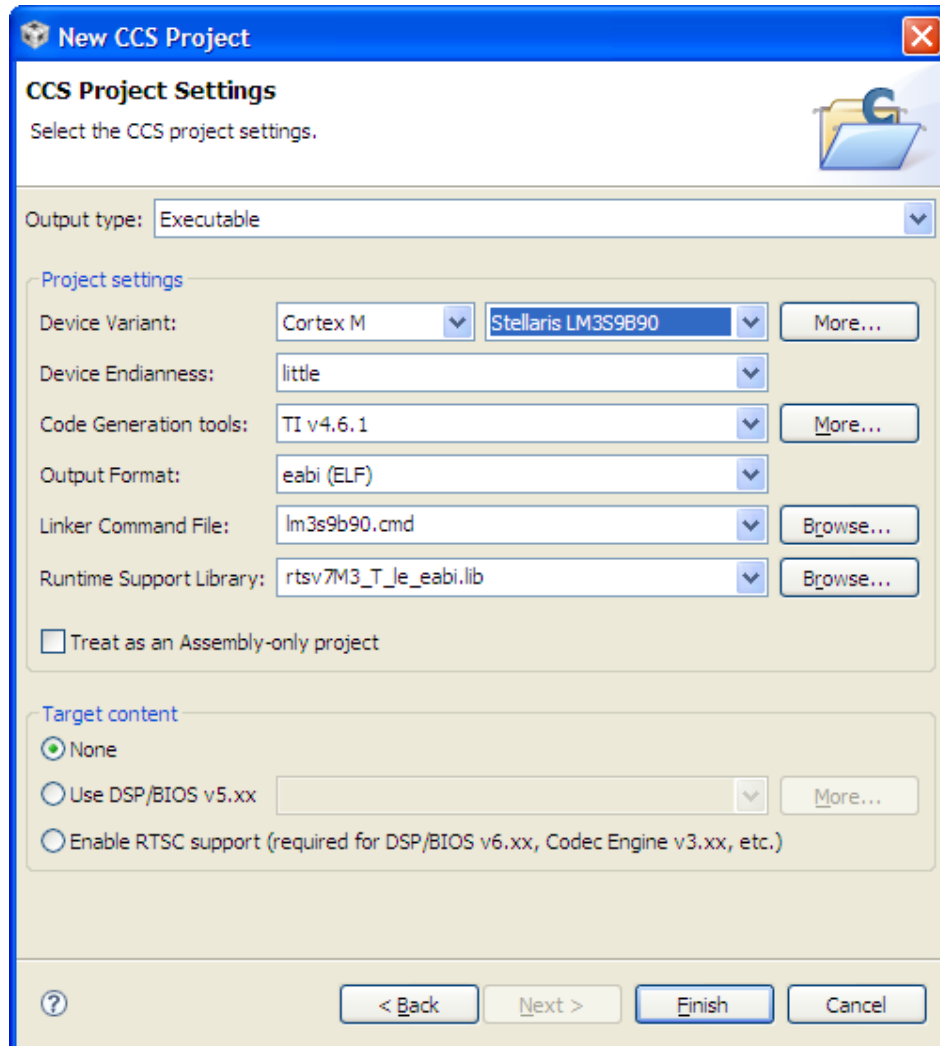
---

4. The next dialog allows you to define any inter-project dependencies. If your project will be using **driverlib**, **usblib**, or **gplib**, now is a good time to define that dependency. **Select** those libraries and click **Next**.



# QUICKSTART – CODE COMPOSER STUDIO

5. On the next dialog, select the appropriate **Stellaris device**, **little endian**, the **TI code generation tool**, **ELF output format**, and the **rtsv7M3\_T\_le\_eabi.lib** runtime library. Click **Finish**.

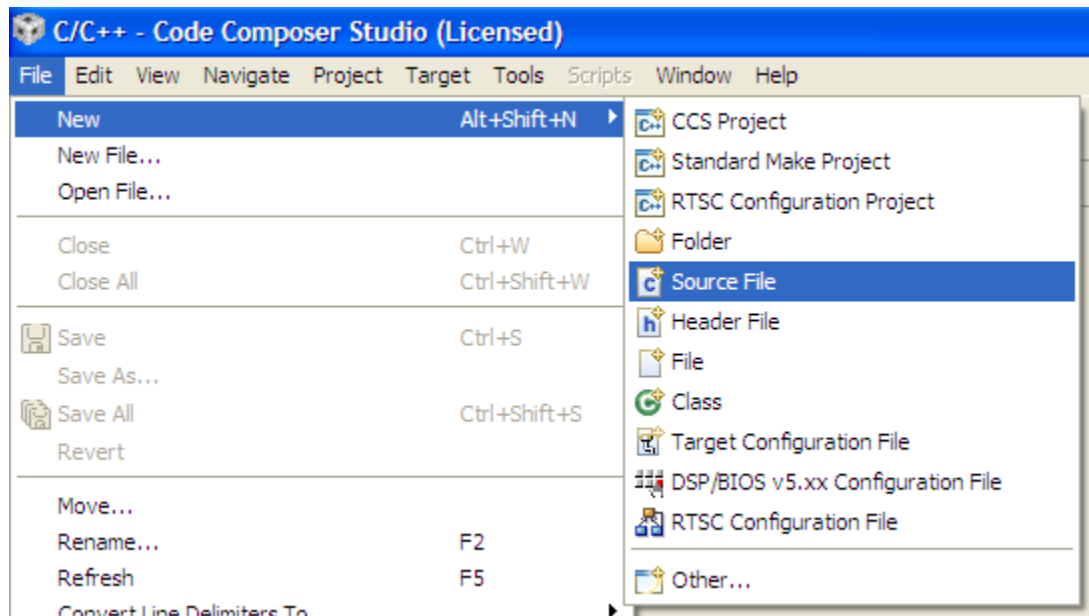


6. A new project has now been created, but the project does not include any source code. The next step is to add some startup code to the project. Using Windows Explorer, copy the **startup\_ccs.c** file from an existing example directory in StellarisWare (such as **C:\StellarisWare\boards\<board>\blinky**) to your new project directory created in step 2 above (in your **My Documents** folder).

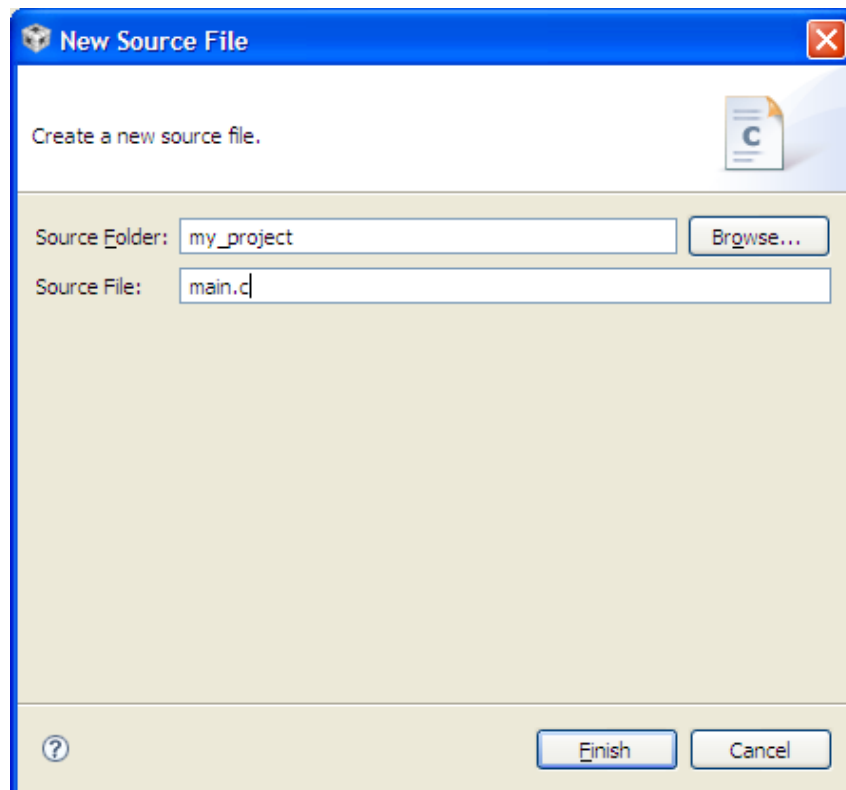
# QUICKSTART – CODE COMPOSER STUDIO

---

7. Create a new C source file by going to **File → New → Source File**.




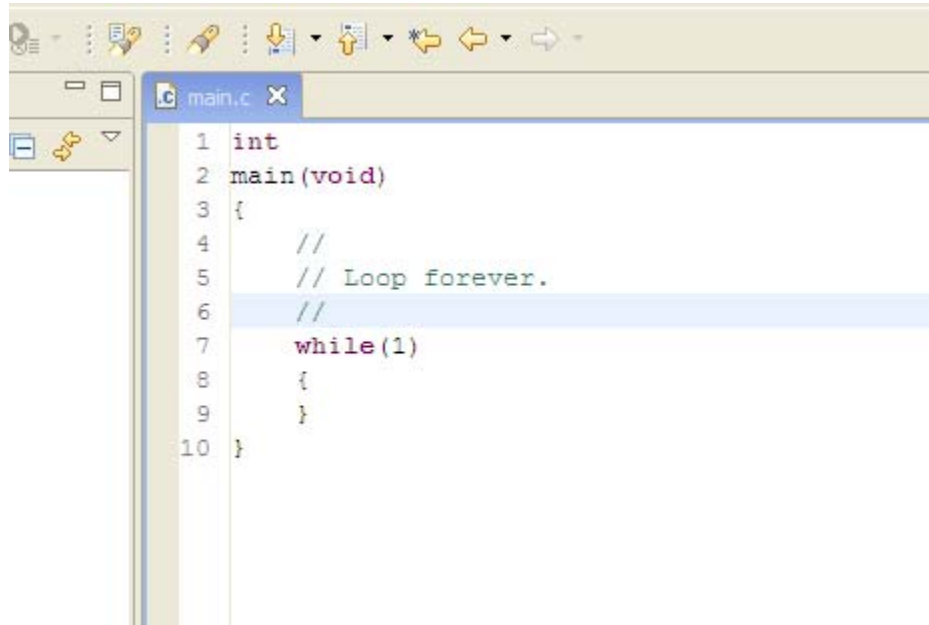
8. In the dialog box that pops up, name the file **main.c** and click **Finish**.



# QUICKSTART – CODE COMPOSER STUDIO

---

9. **Code Composer Studio** will open the **main.c** file for editing. Add a **main()** function to **main.c** as shown below. **Save** the file by clicking the  **Save** button.



10. Both files, **main.c** and **startup\_ccs.c**, should have automatically been added to the project. If not, **right-click** on **my\_project** in the Project pane, select **Add Files to Project** and browse to the files.

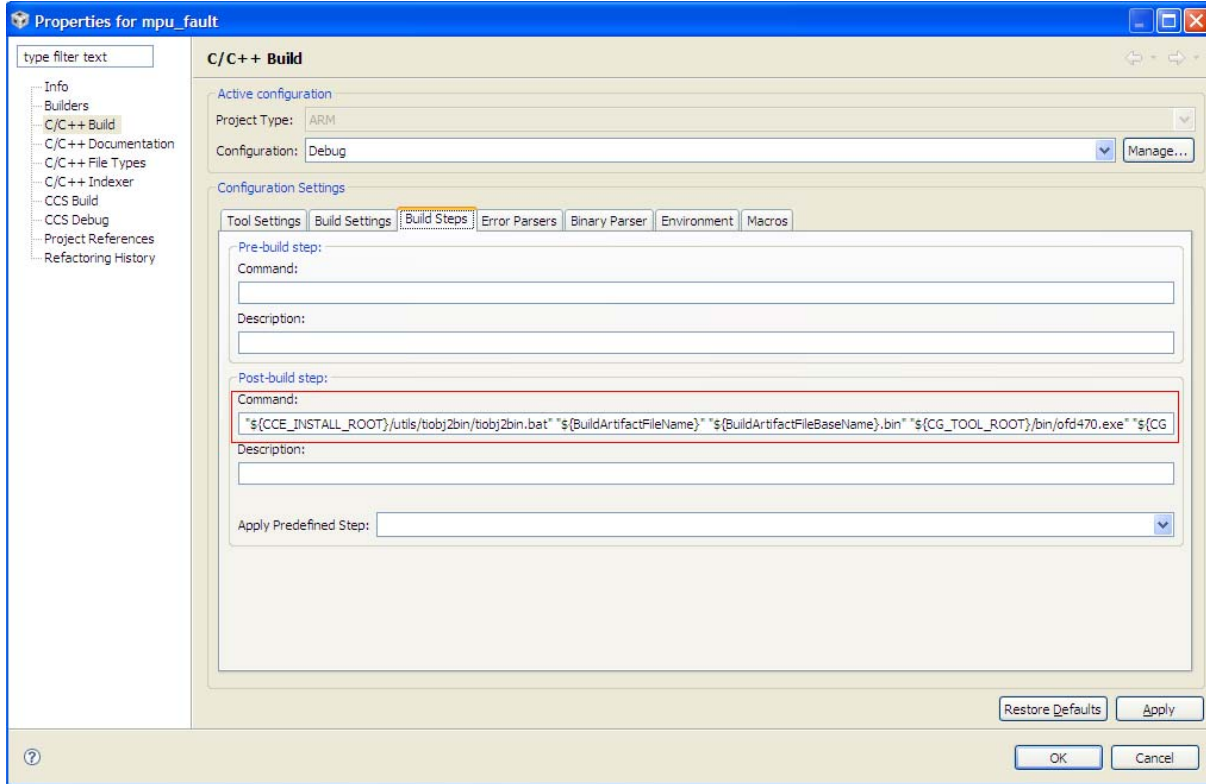


11. **Build** the new project by selecting **Project → Rebuild Active Project**.

With your project created, all you really need to do is add your own code. Use the existing StellarisWare board examples as a reference.

# QUICKSTART – CODE COMPOSER STUDIO

**NOTE:** To set up your project to output a binary file (**.bin**) to be used with tools such as **LM Flash Programmer**, copy the post-build step from an existing StellarisWare example project.



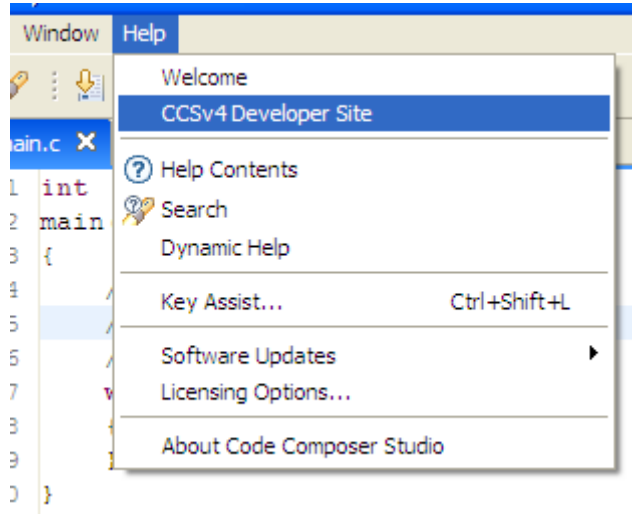


# QUICKSTART – CODE COMPOSER STUDIO

---

## Conclusion

You have now installed the Code Composer Studio development tools and used them to build and load an example application on your Stellaris Evaluation Board. From here, you can experiment with the debugger or start creating your own application using the example projects as a reference. For further information on Code Composer Studio, go to the CCS Developer Site.



**Close** Code Composer Studio now.

## References

The following references are included on the Stellaris Evaluation Kit Documentation and Software DVD and are also available for download at [www.ti.com/Stellaris](http://www.ti.com/Stellaris):

- *Stellaris Evaluation Kit User's Manual*
- *StellarisWare Software*, Order Number SW-LM3S
- *StellarisWare Peripheral Driver Library User's Guide*, Order Number SW-DRL-UG

In addition, the following website may be useful:

- Code Composer Studio website at <http://www.ti.com/ccstudio>

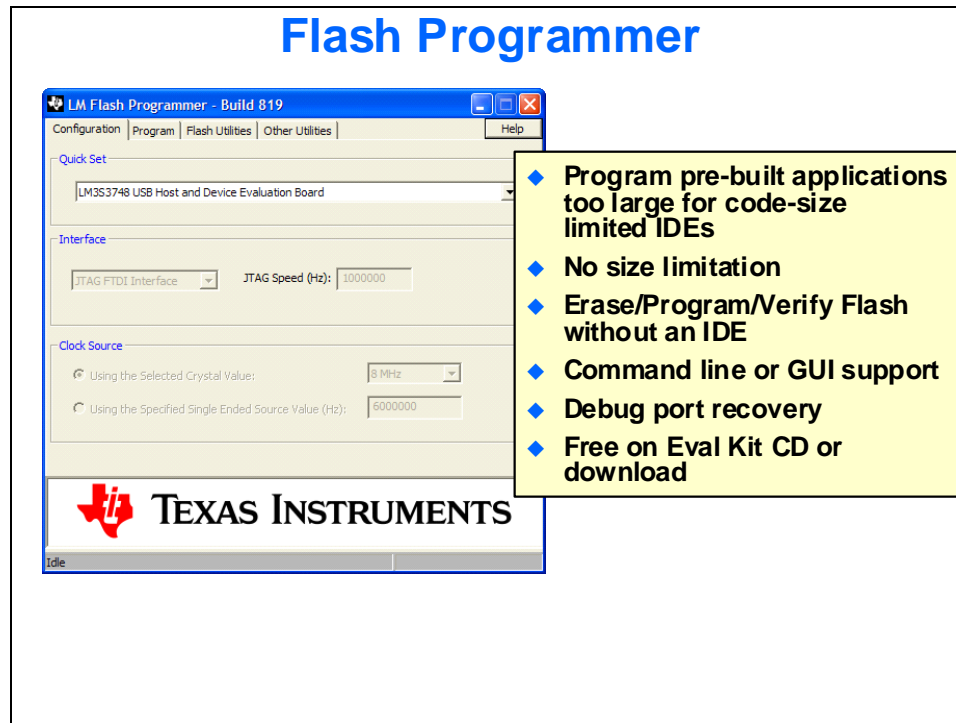
# QUICKSTART – CODE COMPOSER STUDIO

---

# LM Flash Programmer – Installation and Use

Before we run the next lab(s), you'll need to restore the quickstart application to your board.

This lab will step through the installation and use of the LM Flash Programmer. Steps are included for the QuickStart applications on both the LM3S3748 and LM3S8962 evaluation kits.



## Hardware list:

- PC
- USB cable
- LM3S3748 or LM3S8962 evaluation kit

## Software list:

- Workshop Installation Flash Drive

# LM Flash Programmer Procedure

## Drivers

### 1. StellarisWare Package

If you have not done so already, follow the steps in the **QUICKSTART** procedure to install the StellarisWare package for your board(s). This package includes the **bin** files required to program the quickstart application into the Flash memory of the microcontroller on your evaluation board.

### 2. Install the LM Flash Programmer onto your PC

**Insert** the **Workshop Installation Flash Drive** into a free USB port on your computer. Using Windows Explorer, open the Flash drive and double-click on **LMFlashProgrammer.msi**.

Follow the steps in the **Setup Wizard**, agreeing to the license and selecting the default installation folder. When the installation is complete, click **Close**.

### 3. Run the Flash Programmer

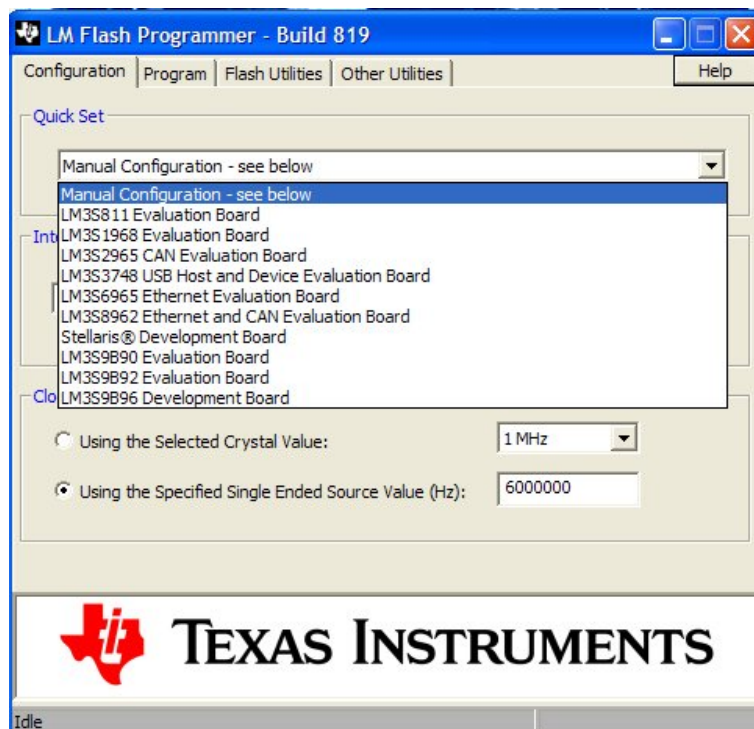
Make sure that Code Composer Studio is **closed** ... otherwise CCS and the Flash Programmer will “fight” for control of the board.



There should be a shortcut to the **LM Flash Programmer** on your desktop **double-click** it to open the tool. If the shortcut does not appear, go to **Start → All Programs → Texas Instruments → LM Flash Programmer** and click on **LM Flash Programmer**.

#### 4. Select Your Board

Select your evaluation board, either the **LM3S3748 USB Host and Device Evaluation Board** or the **LM3S8962 Ethernet and CAN Evaluation Board** from the **Quick Set** pull-down menu under the **Configuration** tab.



#### 5. Programming Setup

Click on the **Program** tab. Click the **Browse** button and navigate to the **bin** file for your board:

**LM3S3748 QuickStart:**

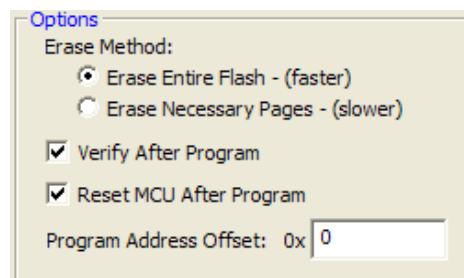
**C:\StellarisWare\boards\ek-lm3s3748\qs-scope\ccs\Debug\qs-scope.bin**

**LM3S8962 QuickStart:**

**C:\StellarisWare\boards\ek-lm3s8962\qs\_ek-lm3s8962\ccs\Debug\qs\_ek-lm3s8962.bin**

**FYI:** There are folders here containing applications which have been built with each supported IDE.

Make sure that the following checkboxes are selected:



## 6. Program the Flash

Assure that your board is properly connected to your computer's USB port and the evaluation boards **Debug** port. **Click** the **Program button**. You should see the programming and verification status at the bottom of the window. After these steps are complete, the application should be running on your evaluation kit.

**Close the LM Flash Programmer.**



You're done.

\*\*\* eee's uv yous \*\*\*



# Stellaris<sup>®</sup> LM3S3748 Evaluation Kit

## README FIRST

The Stellaris LM3S3748 Evaluation Kit provides a low-cost way to start designing applications with Stellaris microcontrollers on a compact and versatile evaluation platform. The evaluation kit design highlights the LM3S3748 microcontroller's key features including USB 2.0 full-speed (12 Mbps) controller, Analog-to-Digital Converter (ADC), and serial interfaces. The LM3S3748 Evaluation Board (EVB) includes connectors for both embedded USB Host and USB Device operation.

The LM3S3748 EVB can be used either as an evaluation platform or as a low-cost in-circuit debug interface (ICDI). In ICDI mode, the on-board microcontroller is bypassed, allowing programming or debugging of an external target.

Power to the EVB can be supplied through the DC jack, the USB Device connector, or the USB debug interface connector. A small switch controls whether the board is bus-powered using the USB Device connector or self-powered using the DC jack or USB debug interface connector.

**WARNING: Do not change the power mode switch while power is applied. Doing so may damage the switch contacts.**

## **Hardware list:**

- PC with two USB ports, running Microsoft® Windows 2000, XP, or Vista
- USB cable
- LM3S3748 evaluation kit

## **Software list:**

- Workshop Installation Flash Drive

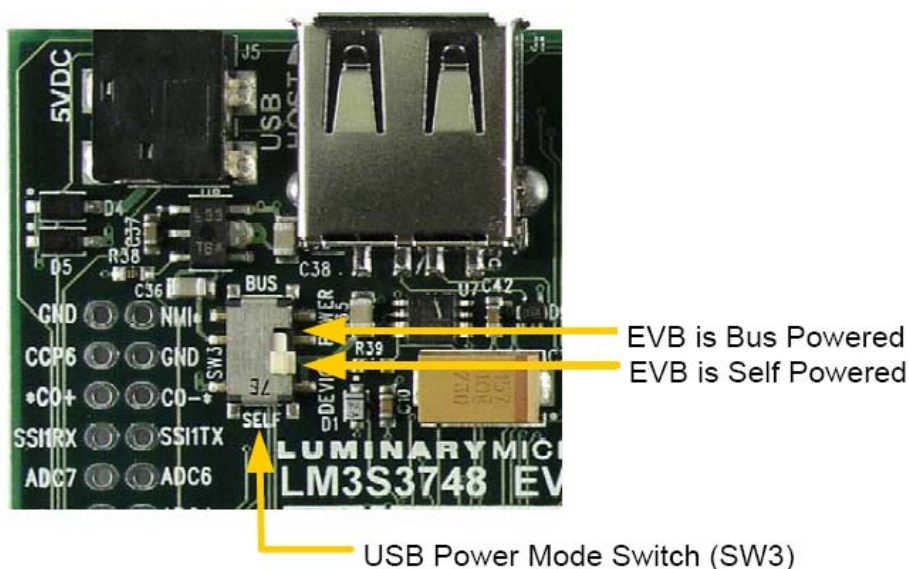
## Procedure

### Initial Board Set-Up

#### 1. Power the EVB

Move the USB power mode switch (**SW3**) to the **SELF** position as shown below.

**Note:** The switch must be placed in the **SELF** position for the initial board setup to work correctly.



#### 2. Connect the USB cable to the board

Using one of the USB cables provided in the kit, connect the mini-b (smaller) end of the USB cable to the USB debug interface connector labeled **DEBUG USB** on the EVB.

**Note:** If the USB cable is not plugged into the **DEBUG USB** connector, the board will not be powered and you will not be able to install the FTDI drivers in the next section.

#### 3. Connect the USB cable to your PC

Connect the other end (Type A) of the USB cable to a free USB port on your host PC.

The PC's USB port is capable of sourcing up to 500 mA for each attached device, which is sufficient for the evaluation board. If connecting the board through a USB hub, it **must** be a powered hub.

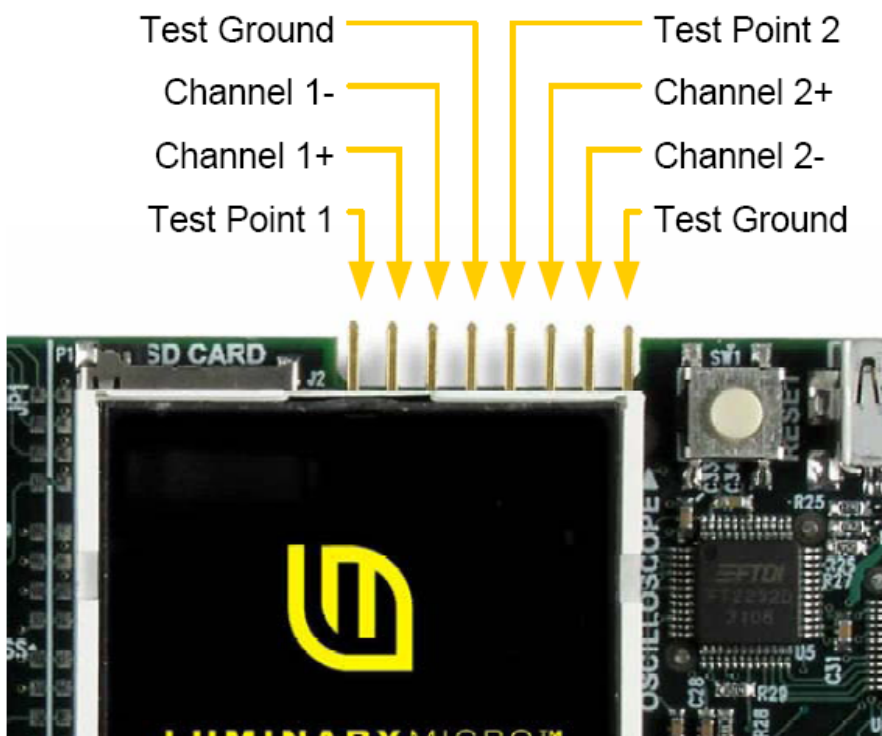
## Quickstart Application

The LM3S3748 Evaluation Board comes preprogrammed with a quickstart application. Once you have powered the board, this application runs automatically. You have probably already noticed this running as you installed the drivers. A splash screen appears on the LCD for a few seconds before the application begins.

The quickstart application provides a simple two channel oscilloscope sampling at up to 1M samples per second. The two oscilloscope channels are differential measurement channels which provide waveform acquisition using the LM3S3748 microcontroller's Analog-to-Digital Converter (ADC). The evaluation board includes an oscilloscope header that contains the two channel differential inputs, two test point pins, and two test ground pins.

**Test Point 1** is connected to the speaker input on the EVB and allows the signal for the keyboard click to be viewed. Note that waveform capture is typically not taking place while the keyboard is being serviced so the click may not be seen on the waveform display for every keypress.

**Test Point 2** is connected to the output of a PWM generator set to drive a 1KHz square wave.



The EVB has a four-way navigation switch with press-to-select functionality that is used to configure the oscilloscope. The navigation switch is labeled **NAVIGATE** on the board. Rocking the control in the desired direction sends **up**, **down**, **left**, or **right** messages to the application and pressing on the center sends the **select** message.

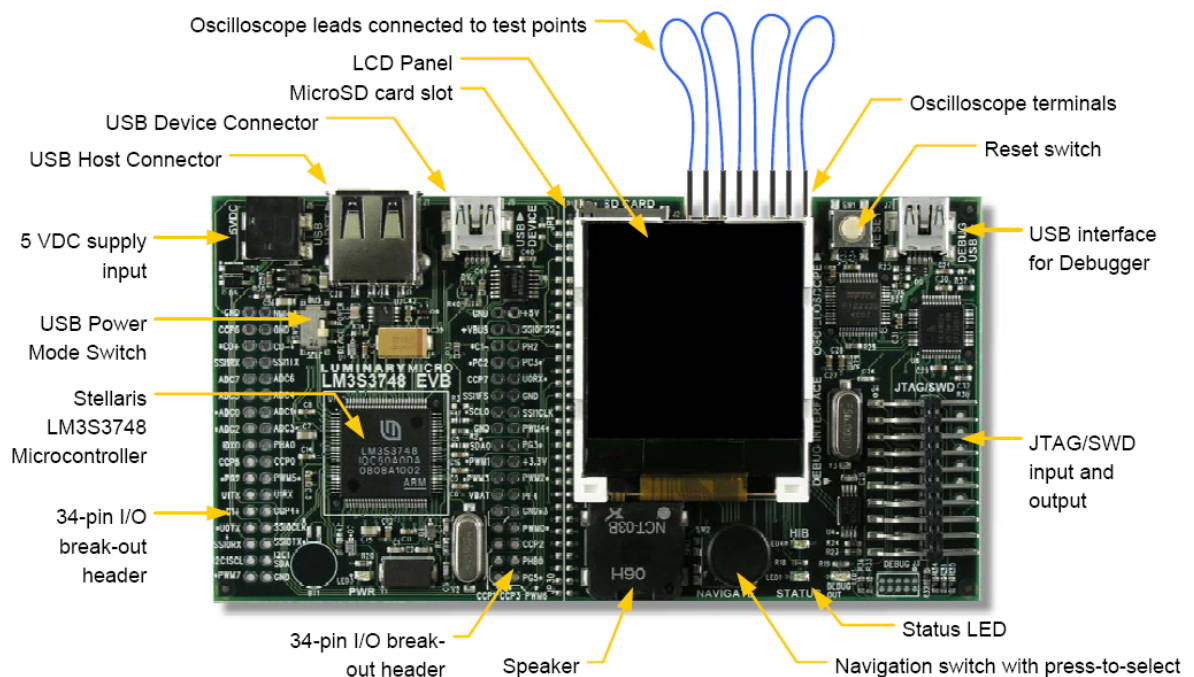
Controls and settings are arranged into groups by function such as display settings, trigger settings, file operations, and setup choices. These groups are accessed by pressing **select** to display the main menu. With the menu displayed, use **up** and **down** to navigate between the available groups. When the desired group is highlighted, press **select** once again to dismiss the menu.

Controls from the currently selected group are shown in the bottom portion of the LCD. Use **up** and **down** to cycle through the controls in the group and **left** and **right** to change the value of, or select the action associated with, the control which is currently displayed.

Using the oscilloscope to view Test Point 1 and Test Point 2, make the following connections using the included jumpers:

1. Connect Test Point 1 to Channel 1+.
2. Connect Test Ground to Channel 1-.
3. Connect Test Point 2 to Channel 2+.
4. Connect Test Ground to Channel 2-.

The connections should look like the graphic below.



5. The test point signals should now be visible on the LCD. To modify the volts per division and time per division options, follow these additional steps. Press **select** to bring up the main menu, navigate to highlight the **Display** group, and press **select** again. The **Display** group will be displayed at the bottom of the LCD.

**Note:** The **Display** group is the default active group after power-up.

6. Press **up** or **down** until the **Timebase** option is selected and then press **left** or **right** to modify the amount of time per division.

7. Press **up** or **down** until the **Ch1 Scale** or **Ch2 Scale** option is selected and then press **left** or **right** to modify the number of volts per division for that channel.

## USB Device Mode

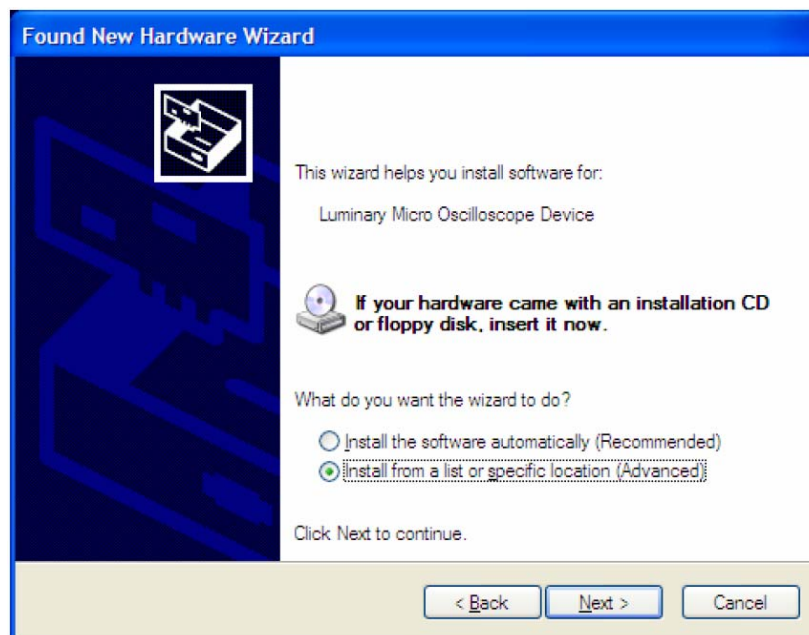
The quickstart oscilloscope application can connect to a Windows host machine via USB where a companion application running on the PC can be used to control the oscilloscope and display and save the waveforms. In this mode, the LM3S3748 EVB will be operating in USB device mode.

The first step is to install the necessary USB drivers on the PC.

1. Disconnect the USB cable from the USB connector labeled **DEBUG USB** if connected.
2. Move the USB power mode switch (SW3) to the **BUS** position.
3. Connect the mini-b (smaller) end of the USB cable to the USB device connector labeled **USB DEVICE** on the EVB. Connect the other end (Type A) to a free USB port on your host PC.
4. Windows starts the **Found New Hardware Wizard** and asks if it can connect to Windows Update to search for software. Select **No, not this time** and then click **Next**.



- Next, the **Found New Hardware Wizard** asks you from where to find the installation software. Select **Install from a list or specific location (Advanced)** and click **Next**.



- Make sure that the **Workshop Installation Flash Drive** is installed and ready in one of your laptops USB ports. Select **Search for the best driver in these locations**, and check **Include this location in the search**. Browse on the Flash Drive to the folder named **F:\windows\_drivers** (the drive letter may be different) and click **Next**



- When the driver installation is finished, click **Finish** to close the dialog box.



**8.** The next step is to install the Windows Oscilloscope application from the **Workshop Installation Flash Drive**.

**Note:** The Windows oscilloscope application only supports WindowsXP and Vista.

Make sure that the **Workshop Installation Flash Drive** is installed and ready in one of your laptops USB ports.

**9.** Using **Windows Explorer**, open the **Workshop Installation Flash Drive**, find the file named **SW-USB-win-xxxx.msi** and **double-click** on it.

When the **Setup Wizard** appears, click **Next** and **Next** again to select the **default installation folder**. **Agree** to the license and click **Next**. Finally click **Next** to start the installation. When the installation completes, click **Close**.

**10.** After the USB examples have been installed, you can run the Oscilloscope application by clicking **Start → All Programs → Texas Instruments → Stellaris → USB Example → LM Oscilloscope**.

You should be able to see and control the application on the board using the Windows application. If you do not see the waveform in the display, **disconnect** and **reconnect** the USB cable on the evaluation board.

**Close** the Windows application when you are finished.

## USB Host Mode

The quickstart oscilloscope application can also run in a USB Host mode. To run the application in USB Host mode, do the following:

1. Disconnect the USB cable from the evaluation board.
2. Move the USB power mode switch (SW3) to the **SELF** position.
3. Connect the mini-b (smaller) end of the USB cable to the USB debug interface connector labeled **DEBUG USB** on the EVB. Connect the other end (Type A) to a free USB port on your host PC.
4. Press **select** to bring up the main menu, navigate to highlight the **Setup** group, and press **select** again. The **Setup** group will be displayed at the bottom of the LCD. Press **up** or **down** until the **USB Mode** option is selected and then press **left** or **right** to switch to **Host** mode.

**Note:** We have had some issues with a recent revision of the quickstart application when switching to host mode. If the application appears to freeze at this point, you will need to re-flash the application. The flash programmer lab in chapter 6 will step you through this process.

In USB Host mode, the quickstart application can save waveforms to a USB flash memory stick or a Micro-SD card. The waveform files can be saved in bitmap (.bmp) or as comma separated value (.csv) formats. The LM3S3748 Evaluation Kit includes a USB flash memory stick, but a Micro-SD card is not included.

To save waveforms to the USB flash memory stick, do the following:

5. Plug the USB memory stick into the **USB HOST** connector. You should see a **USB drive detected** message on the LCD.
6. Press **select** to bring up the main menu, navigate to highlight the **File** group, and press **select** again. The **File** group will be displayed at the bottom of the LCD.
7. Press **up** or **down** until the **BMP on USB** option is selected, and then press **left** or **right** to save the bitmap waveform file to the USB memory stick.
8. Press **up** or **down** until the **CSV on USB** option is selected, and then press **left** or **right** to save the comma separated value waveform file to the USB memory stick.

## Where to Find More Information

For more information on the LM3S3748 Evaluation Kit, see the *Stellaris LM3S3748 Evaluation Kit User's Manual*.

For more information on the LM3S3748 Evaluation Kit Quickstart Oscilloscope Application, see the *StellarisWare® Driver Library User's Guide* in the LM3S3748 Evaluation Kit's Example Applications section.

The above mentioned documents can be found on the Stellaris LM3S3748 Evaluation Kit CD or at the [www.ti.com/Stellaris](http://www.ti.com/Stellaris) web site.

## Software Development Tools

The next step is to install and run the software development tools included in the evaluation kit. For more information, see the quickstart guides included on the Stellaris LM3S3748 Evaluation Kit CD. Additional tools may be available through the [www.ti.com/Stellaris](http://www.ti.com/Stellaris) web site.

## References

The following references are included on the Stellaris LM3S3748 Evaluation Kit Documentation and Software CD and are also available for download at [www.ti.com/Stellaris](http://www.ti.com/Stellaris):

- *Stellaris LM3S3748 Evaluation Kit User's Manual*
- StellarisWare Driver Library
- *StellarisWare Driver Library User's Guide*
- *Stellaris LM3S3748 Microcontroller Data Sheet*



You're done.

\*\*\* one more blank page \*\*\*

# **Stellaris® LM3S8962 Evaluation Kit**

## README FIRST

The Stellaris LM3S8962 Evaluation Kit for Ethernet and CAN provides a low-cost way to start designing Ethernet and controller area network (CAN) applications with Stellaris microcontrollers. The LM3S8962 Evaluation Board (EVB) can function as either a complete evaluation target, or as a debugger interface to any external Stellaris device. The included USB cable is all that is needed to provide power and communication to the host PC.

## **Hardware list:**

- PC with a USB port, running Microsoft® Windows 2000, XP, or Vista
- USB cable
- LM3S8962 evaluation kit

## **Software list:**

- Workshop Installation Flash Drive

# Procedure

## Initial Board Set-Up

### 1. Connect and power the EVB

Connect the LM3S8962 Evaluation Board to the LM3S2110 CAN Device Board using the 10-way ribbon cable included in the kit. The cable inserts into the header labeled CAN on each board.

**Note:** Do not connect the larger 20-pin ribbon cable between the two boards. This will enable JTAG debugging of the LM3S2110. You could inadvertently overwrite the code in the devices flash memory.

To power the EVB, use the USB cable supplied in the kit. Connect the mini-b (smaller) end of the USB cable to the connector labeled **P4** on the EVB. Connect the other end (Type A) to a free USB port on your host PC. A PC's USB is capable of sourcing up to 500 mA for each attached device, which is sufficient for the evaluation board. If connecting the board through a USB hub, it must be a powered hub.

## Quickstart Application

The LM3S8962 Evaluation Board comes preprogrammed with a quickstart application. Once you have powered the board, this application runs automatically. You have probably already noticed this running as you installed the drivers. A splash screen appears on the OLED display for a few seconds before the application begins.

The quickstart application is a game in which you navigate a character through a maze. Use the directional push buttons to move the character, and the user pushbutton (**SELECT**) to fire a missile to destroy the monsters. Score accumulates for maze progress and the number of monsters destroyed. The game lasts for only one character “life,” the score displays at the end of the game.

Since the OLED display on the evaluation board has burn-in characteristics similar to a CRT, the application also contains a screen saver. The screen saver only becomes active if two minutes have passed without the user pushbutton being pressed while waiting to start the game (that is, the screen saver never appears during game play).

After two minutes of running the screen saver, the display turns off and the user LED blinks. Exit either mode of screen saver by pressing the user pushbutton (**SELECT**). Press the button again to start the game.

The LM3S8962 Evaluation Board uses the CAN module on the LM3S8962 to communicate with the included CAN device board. During game play, the volume is adjusted by using the **up** and **down** buttons on the CAN device board. The Status LED on the CAN device board will blink during various portions of the quick start application.

The LM3S8962 microcontroller contains an integrated Ethernet controller, which is also used by the game. During the game, a user can attach the board to a LAN. If there is a DHCP server present on the LAN, the board attempts to obtain an IP address from the DHCP server. After about a minute, if no DHCP server is found, the board defaults to an IP address displayed at the bottom of the main game screen. To view the web page you must configure your host machine to be on the same subnet as the board. If connecting to a LAN, your PC's configuration should match the board's, except for the last 3 digits of the address. In most cases, if you connect the board directly to your PC, your machine automatically detects the correct IP address and subnet settings after several seconds.



## 1. PC Networking Changes

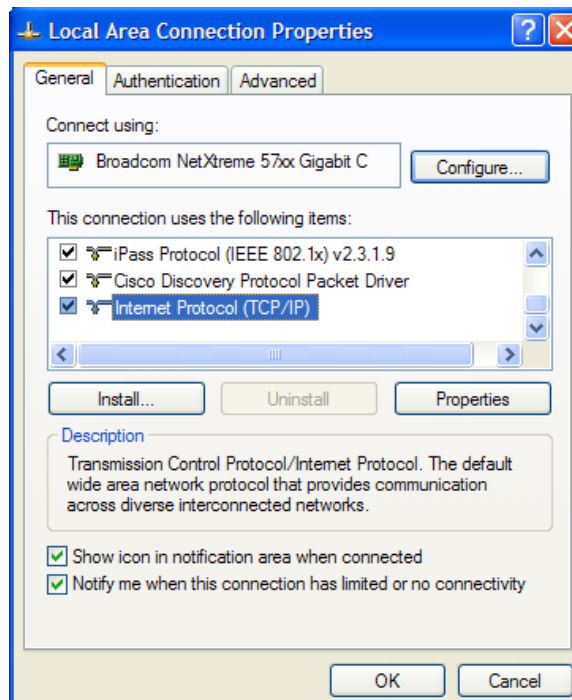
After 30 seconds or so, you should see an IP address at the bottom of the OLED display. This is the IP address of the LM3S8962 Ethernet interface.

If you have a 802.11, AirCard™, 3G or other wireless Internet connection, **disable** it at this time. We want to guarantee that your browser makes its connection attempt through the Ethernet cable. **Connect** the provided Ethernet cable to your PC and the 8962 board.

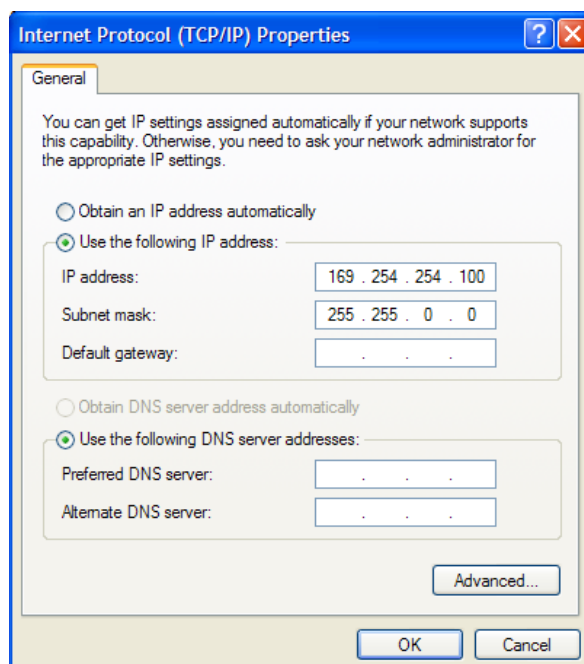
If you have problems connecting to the board in later steps, you may need to disable your firewall software.

Since there is no DHCP server present to allocate an address to the Ethernet port on the LM3S8962 board, the **lwIP** stack has defaulted to a preprogrammed IP address. In order to communicate, we need to provide the PCs Ethernet port a suitable address. Depending on the OS that you're using, the following procedure may be slightly different.

Go to **Start → Control Panel → Network Connections → Local Area Connection**. Click the **Properties** button. When the **Local Area Connection Properties** window appears, scroll down until you see **Internet Protocol (TCP/IP)** like shown below:



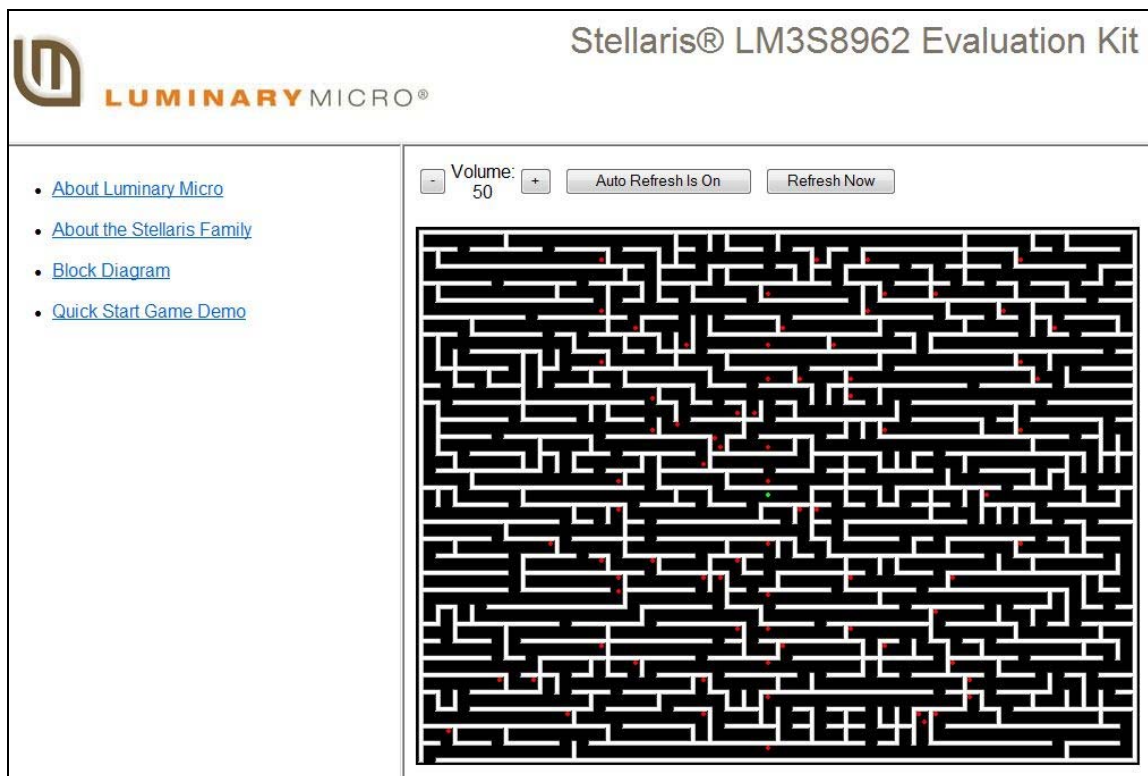
Click the **Properties** button. When the **Internet Protocol (TCP/IP) Properties** window appears, make a mental (or other) note of the settings you see, then make the selections shown below (if your boards' IP address was different, simply make sure that the first three fields are the same as shown on the OLED, and the fourth is different):



Click **OK**. Close the **Local Area Connection Properties** and **Local Area Connection Status** windows.

## 2. Start your browser

Start your web browser, and **enter** the address shown on your OLED display and press **Enter**. A web page, like the one below, served from the LM3S8962 should appear.



If the maze image above appears as a gray box on your browser, you likely do not have Java installed on your PC. Look on the flash installation drive and run the **jre-6u17-windows-i586-s.exe** Java offline installation file.

### 3. UART Connection

While the game is being played, a running tally of the score is output through UART0 of the LM3S8962 microcontroller. UART0 is connected to the FTDI's second serial channel. This serial channel is available to Windows as a Virtual COM port. To determine which COM# Windows has assigned to the Virtual COM port on the LM3S8962 board, follow these steps:

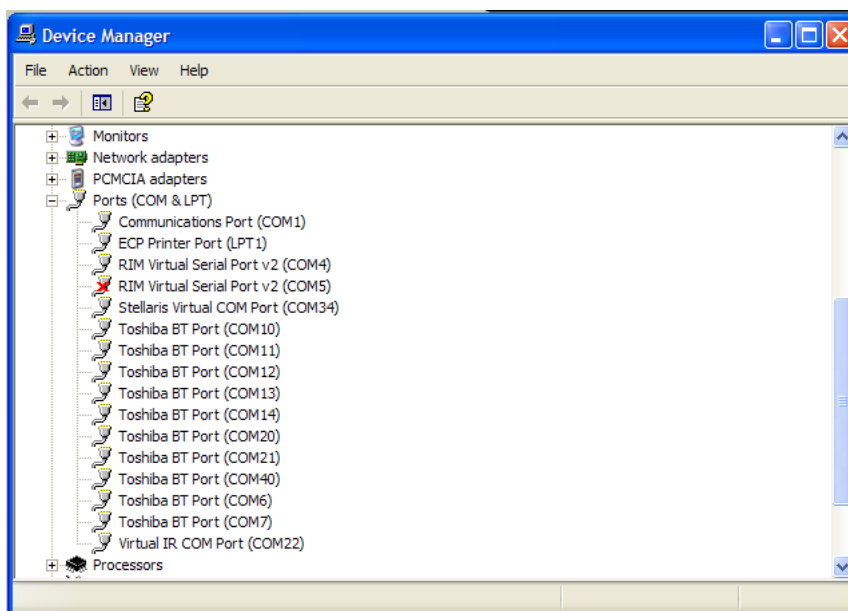
From the **Start Menu**, select **Control Panel**, then double-click the **System** icon.

Select the **Hardware** tab.

Click on the **Device Manager** button.

Click on the + symbol to expand the **Ports (COM & LPT)** group.

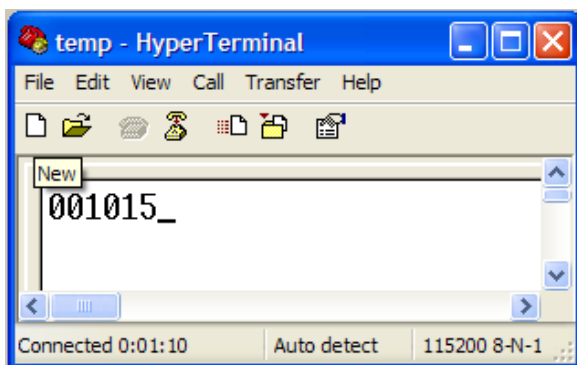
**Stellaris Virtual COM Port (COM#)** is listed as shown in the figure below. This COM# is the device you connect to using your terminal application. In this example, the COM port is **COM34**.




To view the score, open up a terminal application on your PC such as **HyperTerminal** (usually found under **Start → All Programs → Accessories → Communications** ).

Give the connection a name like **Temp**, then connect using COM#, where # is the number Windows has assigned the Virtual COM port. Set the serial connection to a baud rate of **115200**, **8** data bits, **no** parity, **1** stop bit, and **no** flow control.

Your HyperTerminal window should look something like this:



If you have problems running HyperTerminal, open **Code Composer Studio** and, from the Menu bar, click **Window → Show view → Other...**. Click the + next to the **Terminal** folder and select **Terminal**. Click **OK**. At the top of the Terminal pane, click **Settings** . Change the **Connection Type** to **Serial** and use the settings shown above.

## References

The following references are included on the Stellaris LM3S8962 Evaluation Kit Documentation and Software CD and are also available for download from [www.ti.com/Stellaris](http://www.ti.com/Stellaris) :

- *Stellaris LM3S8962 Evaluation Kit User's Manual*
- StellarisWare® Driver Library
- *StellarisWare® Driver Library User's Manual*
- *Stellaris LM3S8962 Microcontroller Data Sheet*



You're done.

\*\*\* another blank page \*\*\*

## Introduction

Thanks for attending the workshop. Make sure to take all your belongings and handouts with you. Good luck with your project!

### Don't Forget!

- ◆ Take your workbook and purchased board(s) home with you
- ◆ Please leave the installation flash drives on the table
- ◆ Fill out the evaluation form on line if possible:  
[www.tiworkshop.com](http://www.tiworkshop.com) and click Online feedback form  
(use paper forms otherwise)
- ◆ Visit the TI Wiki for this workshop:  
[http://wiki.davincidsp.com/index.php/Stellaris\\_One\\_Day\\_Workshop](http://wiki.davincidsp.com/index.php/Stellaris_One_Day_Workshop)

*Thank you for attending*

*Have a safe trip home*

2

