

MSP430L092 Development Guide

Dung Dang

MSP430

ABSTRACT

This application report provides in-depth information to develop an application using the MSP430L092 device. The document begins with a quick-start guide that is useful for working with the MSP430L092 target board kit. It offers further guidance on developing your own prototype with the MSP430L092, in hardware, firmware, and IDE perspectives. The report concludes with some tips and tricks that might be helpful during debug and development with the MSP430L092.

Contents

1	MSP430L092 Target Board Quick-Start Guide	2
2	MSP430L092 Hardware and Software Development	8
3	MSP430L092 IDE, Development and Debug Platform	11
4	FAQs / Tips and Trick	19
5	References	20
Appendix A	MSP430L092 Target Board and Active Cable Schematics and Layouts	21

List of Figures

1	MSP430L092 Target Board	3
2	MSP430L092 Target Board Connections.....	4
3	MSP430L092 Target Board Schematic	5
4	MSP430L092 Active Cable Schematic.....	6
5	MSP430L092 Active Cable	6
6	MSP430L092 Active Cable Connections	7
7	A Typical External SPI-Memory Interface for MSP430L092	9
8	External SPI-Memory Interface Implementation on MSP430L092 Target Board	10
9	SUC and Loader Code Timeline	11
10	MSP430L092 Project Selection in IAR	13
11	L092 and C092 Emulation Modes in IAR	14
12	MSP430L092 Firmware Download Options [RAM/EEPROM] – IAR	15
13	MSP430L092 EEPROM Download - Fill Memory Option in IAR.....	16
14	MSP430L092 Project Selection in Code Composer Studio.....	17
15	MSP430L092.ccxml in a Code Composer Studio MSP430L092 Project	17
16	Advanced Setup → Target Configuration in Code Composer Studio	18
17	L092/C092 Emulation Type Selection in Code Composer Studio	18
18	MSP430L092 Firmware Download Options [RAM/EEPROM] in Code Composer Studio	19
19	MSP430L092 Target Board Schematic.....	21
20	MSP430L092 Target Board Top Layer	22
21	MSP430L092 Target Board Bottom Layer	23
22	MSP430L092 Target Board Components and Through Holes	24
23	L092 Active Cable Schematic.....	25
24	L092 Active Cable Top Layer	25
25	L092 Active Cable Bottom Layer	26

26	L092 Active Cable Components and Through Hole	26
----	---	----

List of Tables

1	Active Cable Supply Power Jumper Configuration	7
2	Memory Map - MSP430L092 in C092 Emulation Mode	12

1 MSP430L092 Target Board Quick-Start Guide

The MSP430L092 development kit consists of three hardware components: the traditional MSP-FET430UIF, the MSP430L092 target board and the L092 active cable board.

The MSP-FET430UIF is the most popular IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture (JTAG) programming and debugging tool that can be used for most of MSP430 devices across different family devices. It is used to program and debug the MSP430L092 platform in JTAG mode. The target board is the main evaluation platform for the MSP430L092, and the L092 active cable board is used to connect the MSP-FET430UIF and the target board.

1.1 MSP430L092 Target Board

The MSP430L092 target board is the standard development platform for the MSP430L092 devices. The target board provides a 14-pin PW socket housing the device, as well as the pin-out connectors for the 14 pins of the device. The target board is supplied by a voltage source of up to 1.5 V, either from the JTAG connection or externally from a bench power supply or a single-cell battery.

However, due to the nature of the MSP430L092 device containing only RAM memory and no non-volatile memory, the board also provides an external electrically erasable programmable read-only memory (EEPROM) IC to store the MSP430 program code. This allows for you to program the code into the EEPROM and restore the code onto the MSP430 upon start up, enabling a stand-alone operation mode.

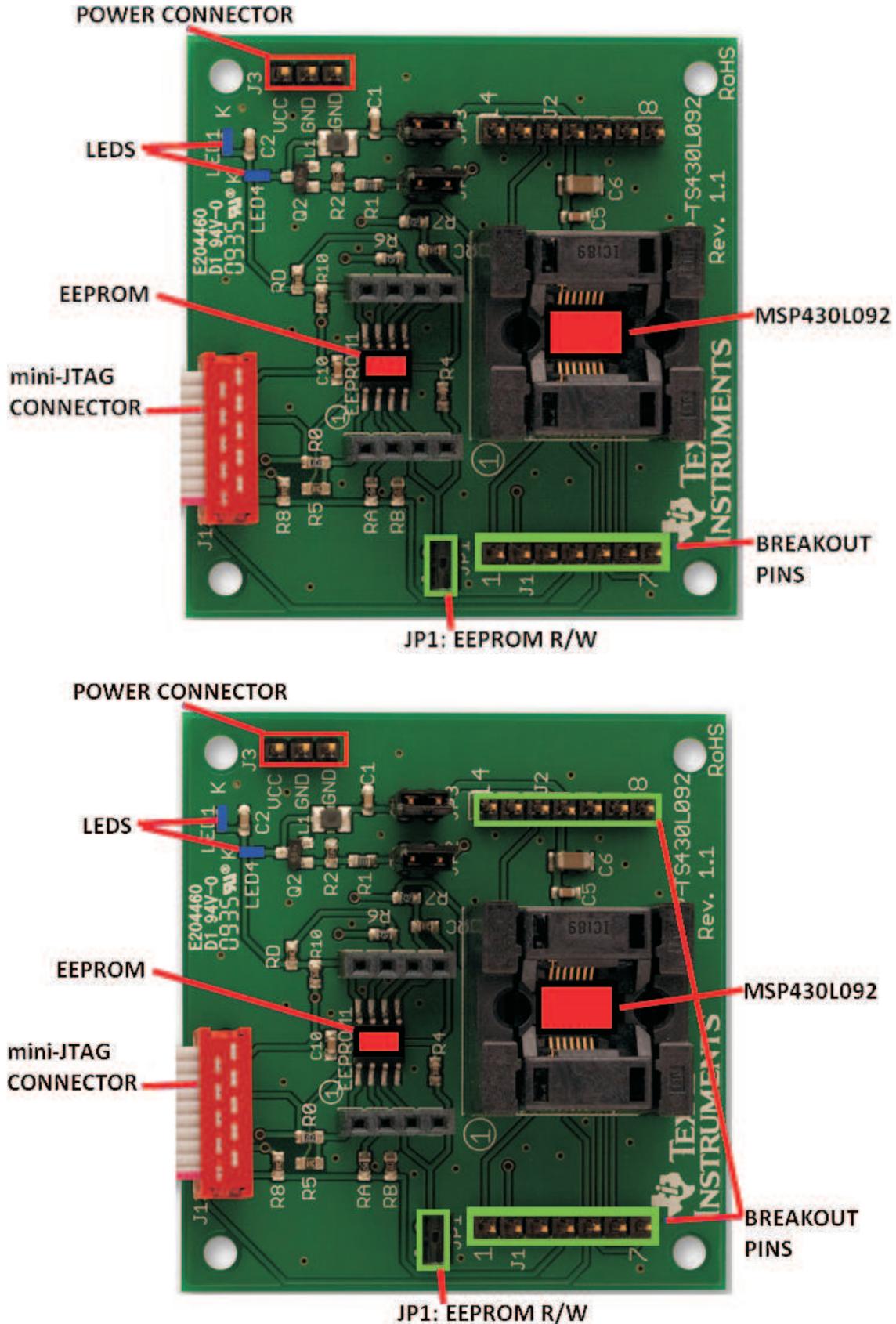


Figure 1. MSP430L092 Target Board

The MSP430L092 target board image and connections can be found in [Figure 1](#) and [Figure 2](#).

All 14 pins of the MSP430L092 are broken out to the two single-row connectors J1 and J2. This allows for easy debugging with oscilloscopes and power supply, or prototyping with external connectors.

J13 is the mini-JTAG connector that can be connected to the MSP430L092 active cable via the mini-JTAG cable.

J3 is the power connector, V_{CC} and two GNDs, that allows for external power supply, such as from a single-cell battery. Stand-alone prototype or demo can be realized using this power option.

JP1 controls the read-only/writable status of the EEPROM. When set, EEPROM is writable, otherwise EEPROM is read-only.

JP2 and JP3 connect device supplies with boost converters. They can be opened to measure device current consumption, but close connection is required to enable the boost converter for the EEPROM.

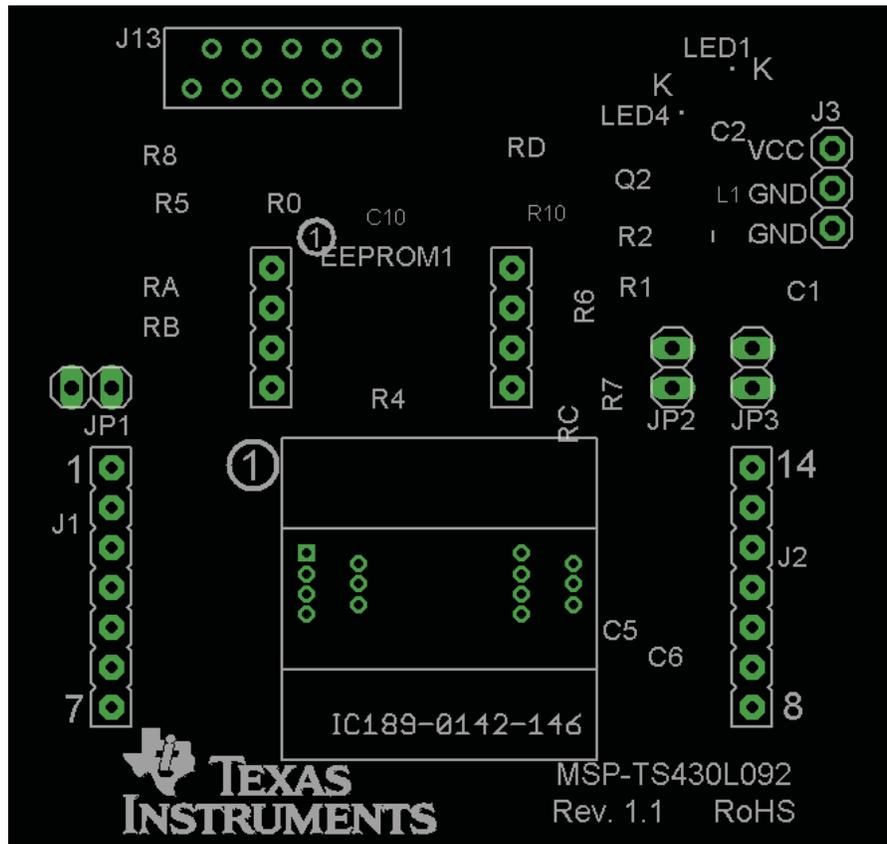


Figure 2. MSP430L092 Target Board Connections

Since both the target board and the MSP430L092 operate at the 0.9 V-1.5 V range, but the EEPROM IC operates at 3-V logic level, a 3 V-level voltage source is required to power the EEPROM, and a logic level translation is required to accommodate the communication between these two devices. The power source is effectively enabled by the use of the MSP430L092-driven pulse width modulator (PWM) in conjunction with a passive boost converter. The logic level translation is enabled with an adaptive network (passive resistor divider). [Figure 3](#) shows the schematic of the MSP430L092 target board, including the boost converter and serial peripheral interface (SPI).

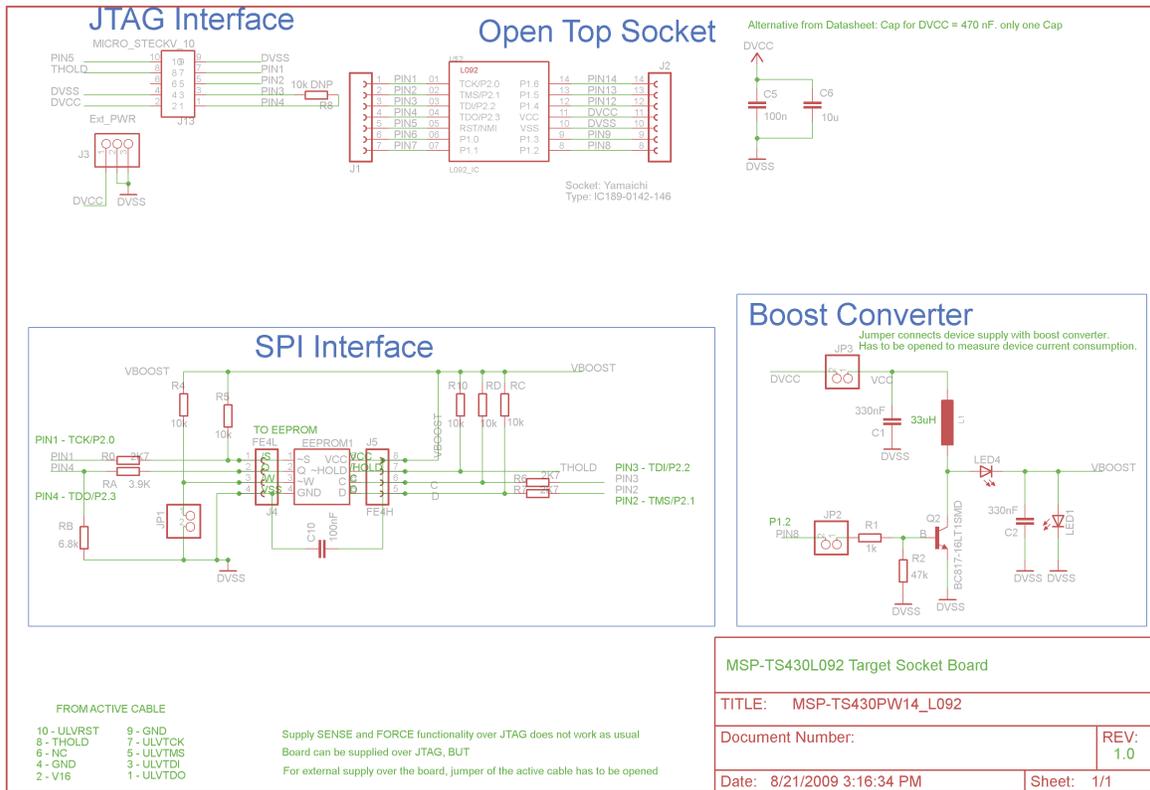


Figure 3. MSP430L092 Target Board Schematic

The boost converter circuit is also used to drive two LEDs, LED1 and LED4. The default usage of the LEDs is to indicate the boost converter status and EEPROM firmware upload/download progress. They can also be used for further in-application debugging purposes.

1.2 Active Cable – JTAG Level Shifter

The MSP430L092 active cable is a bi-directional voltage level shifter bridge that translates the power and JTAG signal at 3.3 V from the MSP-FET430UIF down to 1.5 V level to the MSP430L092 target board and vice versa. The MSP430L092 active cable has two sets of jumpers, allowing for power supply configuration.

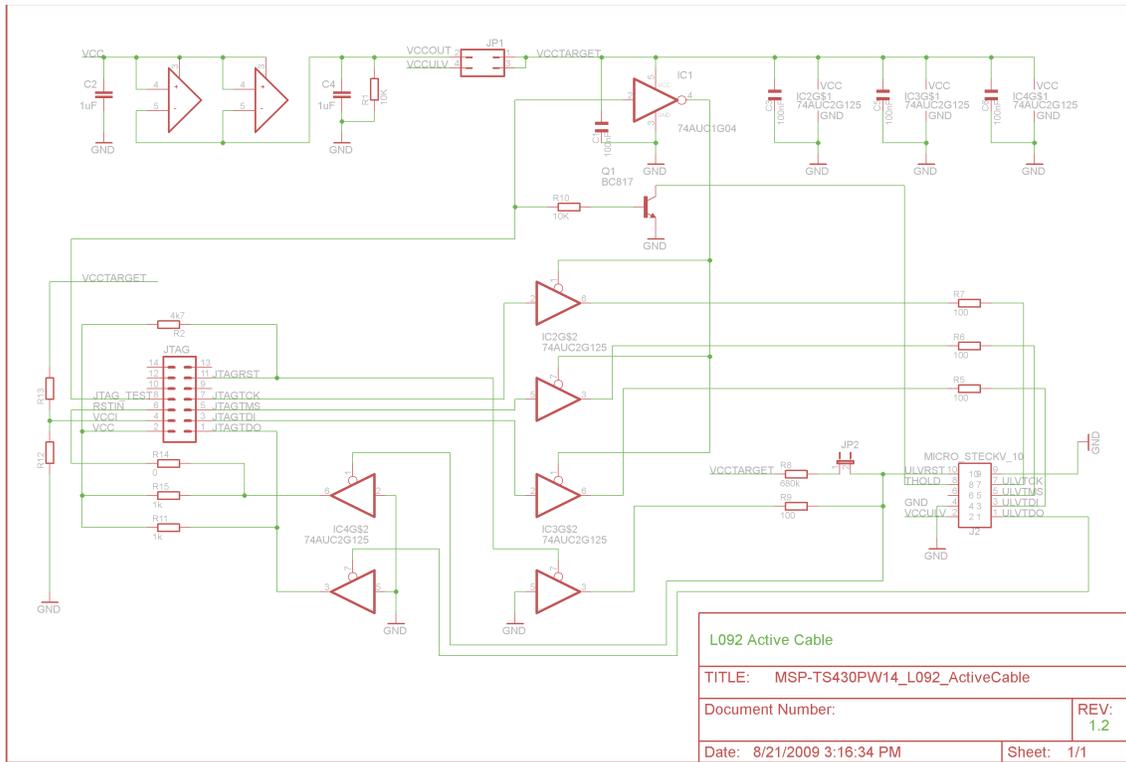


Figure 4. MSP430L092 Active Cable Schematic

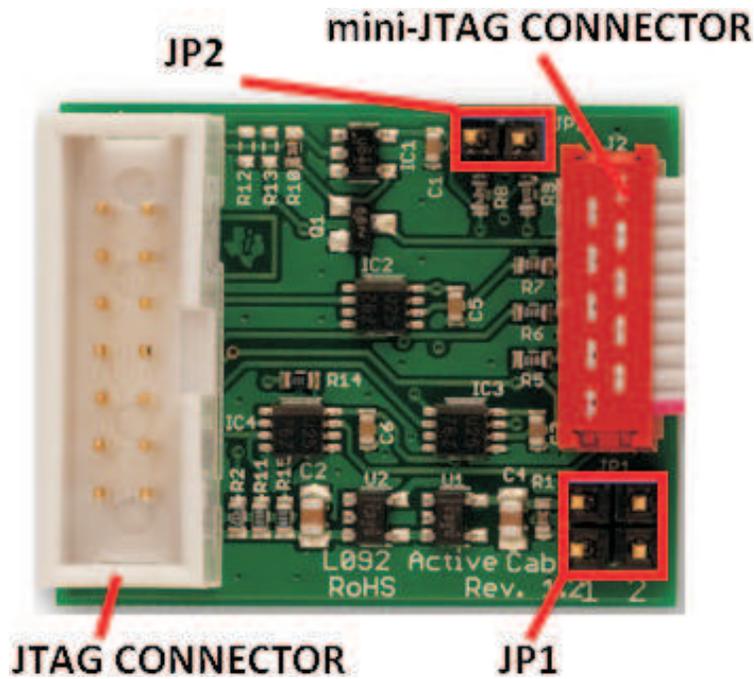


Figure 5. MSP430L092 Active Cable

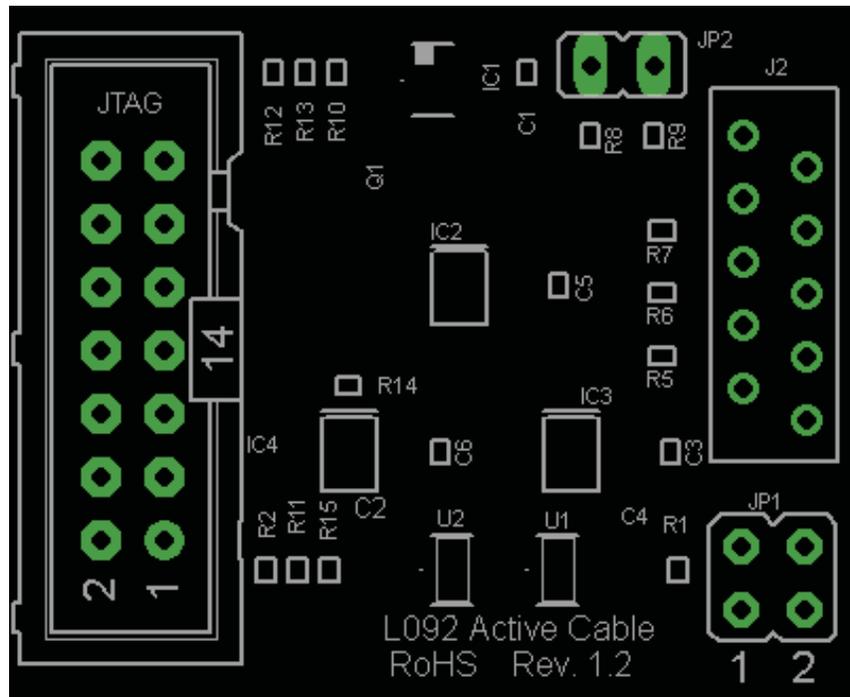


Figure 6. MSP430L092 Active Cable Connections

The main JTAG connector provides regular JTAG connection with the MSP-FET430UIF. The mini-JTAG connector, J2, is to be connected to the MSP430L092 target board via a mini-JTAG cable.

Table 1. Active Cable Supply Power Jumper Configuration

JP1	JP2	Description
Off	Off	Active cable has no power and is not working.
Off	On	Active cable gets power from target socket. For this, the target socket needs to have its own power supply.
On	Off	Active cable gets power from target socket. For this the target socket needs to have its own power supply.
On	On	The JTAG connector is powering the active cable and the target socket. The target socket must not have its own power source; it would cause an undefined state.

1.3 Default Project

Out of the box, the EEPROM on the MSP430L092 target board is pre-programmed with the firmware that allows the MSP430L092 to blink the LEDs once loaded onto the device. Upon providing power to the target board, the MSP430L092 starts up, executes the boot code (ROM), executes the loader code (ROM) that downloads the application code from the EEPROM into RAM memory and then executes the loaded application code from RAM.

2 MSP430L092 Hardware and Software Development

In the first phase of evaluation and prototyping, MSP430L092 developers are presented with two practical choices of hardware development platform: spinning a PCB specifically for the MSP430L092 or simply reusing the MSP430L092 target board and building the prototype on top.

In the first option, further hardware decisions must be made depending on the firmware retention requirement. If the prototype is to only be tested with JTAG connections or with constant power supply, the design only needs to involve the MSP430L092. In the second option, if a stand-alone prototype is intended to resume operation after each power off, external non-volatile memory IC should be designed in the prototype as well. TI provides several reference designs for both the boost converter and SPI interface to work with numerous types of memory ICs. These references can be found in the *MSP430x09x Family User's Guide* ([SLAU321](#)).

With the use of the provided reference designs, the developer prototype requires very little design effort. In contrast, the second option, requires no hardware design or reinvestment with regard to the non-volatile memory and the supporting circuitry.

[Section 2.1](#) provides a detailed description of the MSP430L092 target board and also points to the reference designs for more custom development. As a by-product of the SPI memory hardware, the boost converter can also be used to drive other components or applications that require a higher voltage.

[Section 2.3](#) describes the MSP430L092 Loader Code, resident in the ROM space of the MSP430L092, which provides function calls to power up and communicate to the external SPI-memory. These calls are made available to the application and can also be of benefit to you during software development.

2.1 EEPROM, Boost Converter + PWM, SPI Level Shifter

During the development cycle, if an L092 prototype requires operation without JTAG or constant power connection, an external non-volatile memory device is required for external firmware storage. Since most of the EEPROM-type ICs operate at the 3-V logic level and use serial communication, booster converters, as well as a voltage-level shifter for serial communication are required to enable the interface between the MSP430L092 and the external EEPROM.

This section provides some initial information on the requirements as well as hardware recommendation for such system.

Figure 7 illustrates a typical interface between the MSP430L092 and the SPI memory IC. The MSP430L092 is responsible for both driving the power and communicating to the SPI memory IC. These two functions are aided by two circuits: one for generating the 3 V power from a 0.9 V source and the other for adapting the logic signals between these two voltage nodes.

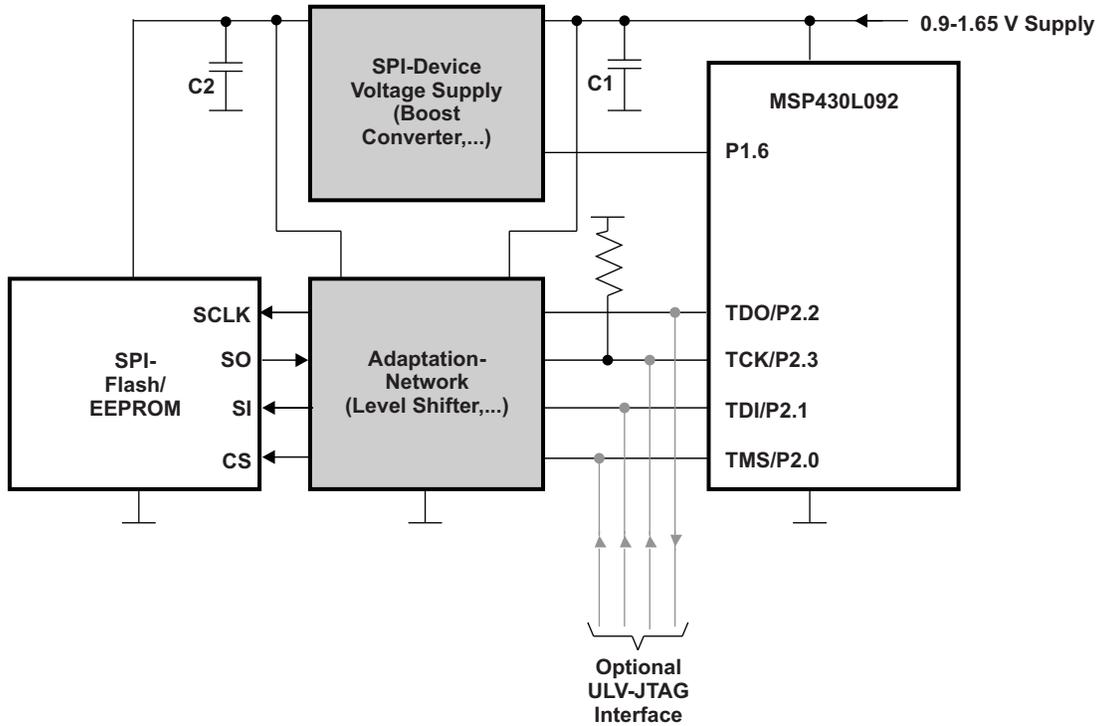


Figure 7. A Typical External SPI-Memory Interface for MSP430L092

Figure 8 shows the specific implementation of the aforementioned circuits on the MSP430L092 target board. This implementation also serves as a reference design for any 0.9 V-3.0 V boost or logic system. Other recommendations can be found in the Loader Code section of the *MSP430L092 Loader Code User's Guide* ([SLAU324](#)).

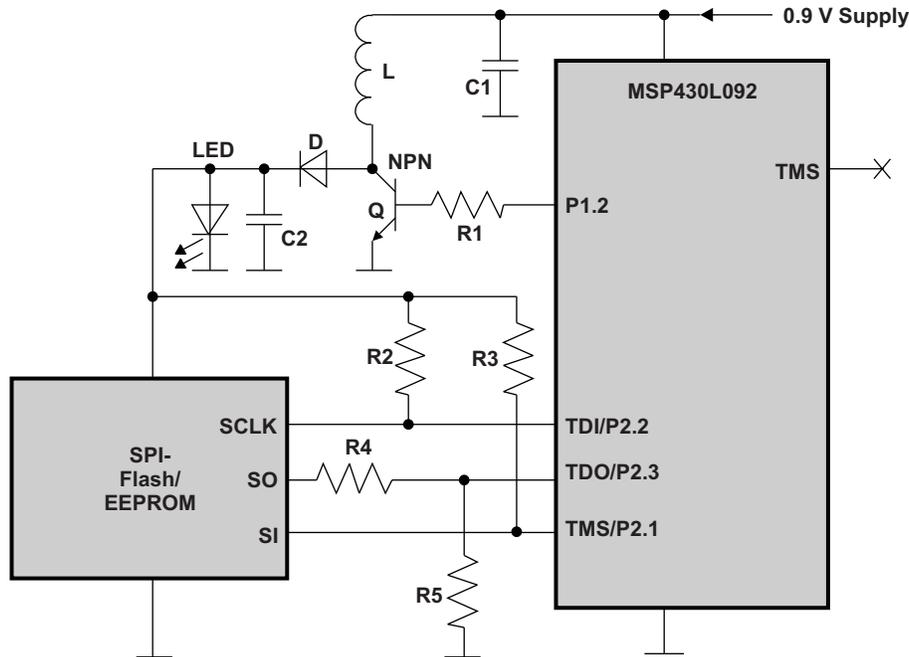


Figure 8. External SPI-Memory Interface Implementation on MSP430L092 Target Board

2.2 Software Development

The MSP430L092 inherits all the traditional features of an MSP430 device allowing for similar software development cycles on the new platform.

- Similar compiler, header files, support libraries, etc., code development in MSP430L092 will be 100% compatible with previous families.
- Code reuse for all traditional or legacy modules such as core, WDT_A, or Timers.
- Code examples package available for immediate testing and evaluation.
- Application reports and training to provide technical information.

Furthermore, internal software support for this system is already implemented in the ROM memory of the MSP430L092. When interfaced with compatible circuitry and external Flash/EEPROM memory IC per our guidance, the MSP430L092 firmware can take advantage of the ROM bootloader APIs to drive the boost converter and communicate with the non-volatile memory device.

For further details on design guidelines, see the L092 Loader Code [7].

2.3 Loader Code

The Loader Code is part of the MSP430L092 boot code stored in ROM of the device, which controls the supply power and communication with the external SPI memory and handles the firmware download into the RAM space of the MSP430L092.

Being part of the device boot code, the Loader Code is executed during the start-up sequence of the MSP430L092. After V_{CC} ramp-up or reset release, the control is given to the start-up code (SUC). The SUC initializes the device and verifies its integrity, then passes control to the Loader Code. Figure 9 shows the detailed timeline of the Loader Code activities, including driving a 250 kHz PWM signal from P1.2 to boost power for the external memory and downloading the firmware from the external memory into RAM. After the checksum of the downloaded firmware is correctly verified, the control is passed onto the firmware application code in RAM.

The timeline also helps you understand when the SPI and boost connections are necessary. During debug, these pins can be reused for application purposes after the loader code has finished.

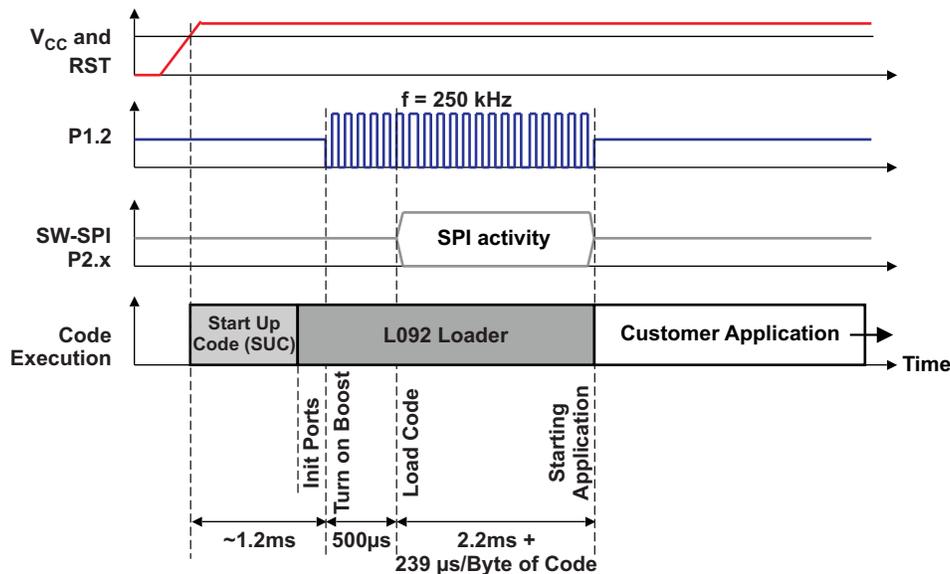


Figure 9. SUC and Loader Code Timeline

The Loader Code supports various types of SPI memory devices to interface and work with the MSP430L092. For more detailed information, see the *MSP430L092 Loader Code User's Guide* (SLAU324). Some of the common SPI memory types include EEPROM, Flash, static random access memory (SRAM) or ferroelectric RAM (FRAM) memory ICs.

The Loader Code is architected to provide a set of API calls that can then be further used by the MSP430L092 application itself.

For more details on how to take advantage of these API calls, see the L092 Loader Code .

3 MSP430L092 IDE, Development and Debug Platform

Texas Instruments provides MSP430L092 code development support for the RAM-only device in two Integrated Development Platforms (IDE): Code Composer Studio™ and IAR. This chapter details how to enable MSP430 code development in each IDE.

There are two methods to debug an MSP430L092 system: the IDE can directly download the firmware onto the MSP430L092 RAM or interface with the MSP430L092 Bootloader APIs in ROM to download the firmware into the external EEPROM. Upon the completion of firmware download, the IDE can start debugging the application via JTAG control.

3.1 C092 Emulation Mode in MSP430L092 (Memory Swap)

If the MSP430L092 device is selected, two debugging modes are available for the project depending on the intention of the final application. If the application is implemented with the MSP430L092, the L092 mode should be selected. If the final firmware is on the MSP430C09x (ROM mask) platform, the C092 emulation mode should be selected to emulate the ROM addressing space with the RAM memory.

Table 2. Memory Map - MSP430L092 in C092 Emulation Mode

	Type	MSP430C091	MSP430C092	MSP430L092	MSP430L092 (EMU)
Primary interrupt vectors:	ROM	32 B 0x0FFE0-0x0FFF	32 B 0x0FFE0-0x0FFF	32 B 0x0FFE0-0x0FFF	32 B 0x0FFE0-0x0FFF
Secondary interrupt vectors:	RAM (lockable)	-	-	0x01C60-0x01C7F	
Application ROM memory	ROM	864 B 0x0FC80-0x0FFDF	1888 B 0x0F880-0x0FFDF	-	ROM not available
Boot Code (BC)/Loader Code	ROM (by TI)	128 B (BC) 0x0F800-0x0F87F	128 B (BC) 0x0F800-0x0F87F	2016 B (Loader) 0x0F800-0x0FFDF	Config/Loading by tool 0x0F800-0x0F87F
RAM memory	RAM	128 B 0x02380-0x023FF	128 B 0x02380-0x023FF	128 B 0x02380-0x023FF	128 B 0x02380-0x023FF
LRAM memory (lockable)	RAM	-	-	1792B 0x01C80-0x0237F	1760B 0xF900-0xFFDF
CRAM memory (lockable)	RAM	96B 0x01C00-0x01C5F	96B 0x01C00-0x01C5F	96B 0x01C00-0x01C5F	128B 0x0F880-0x0F8FF
Peripherals	Size	4kB 0x00000-0x00FFF	4kB 0x00000-0x00FFF	4kB 0x00000-0x00FFF	4kB 0x00000-0x00FFF

In the MSP430L092, the RAM memory dedicated for application firmware resides at 0x1C80-0x237F. Its ROM Loader Code is located at 0xF800-0xF87F.

However, in MSP430C09x devices, this 0xF800-0xF87F memory space is reserved for the firmware ROM memory, specifically 0xFC80-0xFFDF for C091 and 0xF880-0xFFDF for C092.

To simulate a C09x device's memory map within an MSP430L092, the application RAM space must be swapped with the boot Loader Code space. The memory address swapping mechanism is enabled by a combination of address decoding translation at device level, as well as from the IDE during compilation time for firmware and code address allocation and debugging time with JTAG working in conjunction with the device address decoding.

3.2 Download Firmware to RAM or to External SPI Memory

There are two methods to debug an MSP430L092 system: directly from MSP430L092 application RAM or via the aid of the external SPI memory.

The IDE can directly download the firmware onto the MSP430L092 RAM or interface with the MSP430L092 Boot Loader APIs in ROM to download the firmware into the external EEPROM.

In the first method, the IDE can directly download the firmware into the MSP430L092 RAM and immediately initiate the debug session.

The second method can be used to either store the firmware into the external SPI memory or to debug the application after such download. First, the IDE interfaces with the MSP430L092 Loader Code, resident in MSP430L092 ROM, to power up and transmit the firmware to the external SPI memory IC. If immediate debug is required after firmware transfer completion, the IDE can use the same Loader Code to retrieve the firmware from the external SPI memory and load it into the RAM of the MSP430L092. The IDE can then start the debug session.

3.3 MSP430L092 Development in IAR

This section details the steps to create an MSP430L092 project in IAR. MSP430L092 family device support is available in IAR version 5.2 or later.

1. Create a project using the New Project option in IAR.
2. Select the MSP430L092 as the project device from the drop-down menu under Project → Options → General Options.

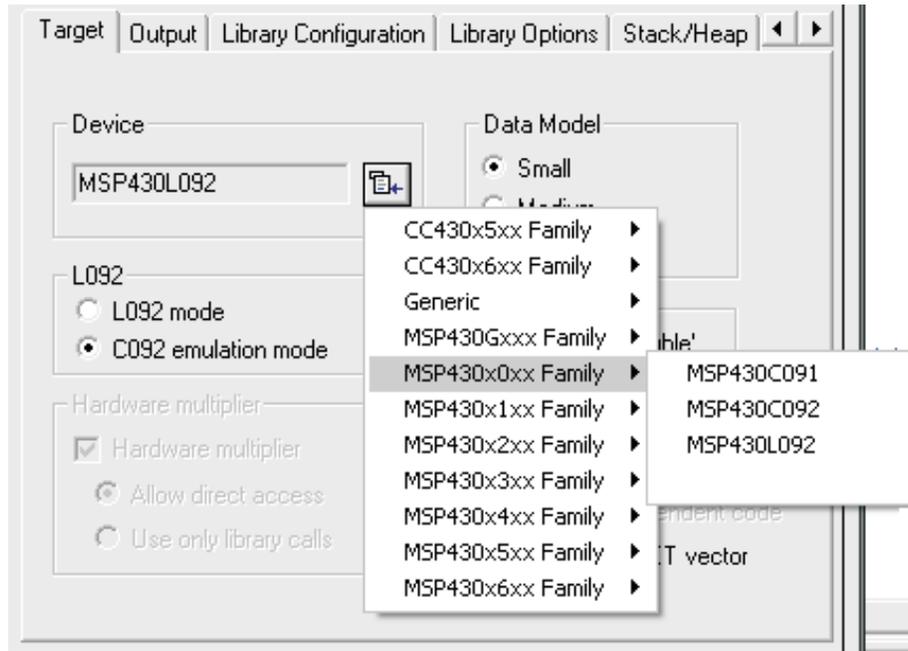


Figure 10. MSP430L092 Project Selection in IAR

3.3.1 C092 Emulation Mode in IAR

The project can be selected to be a strict L092 project or managed in C092 emulation mode. This selection is available with the radio buttons in the L092-specific option box (see [Figure 11](#)).

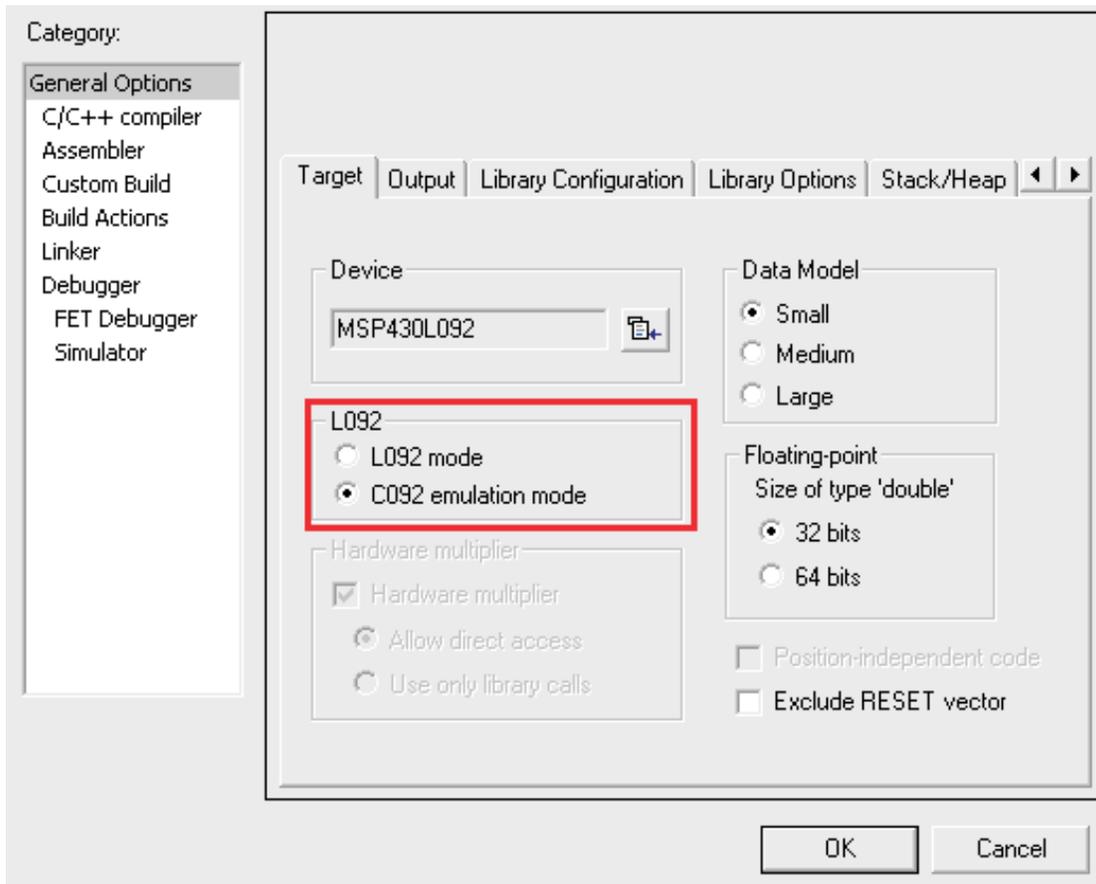


Figure 11. L092 and C092 Emulation Modes in IAR

3.3.2 RAM/External SPI Memory Download Option in IA

During development, the MSP430L092 project can utilize both debugging modes: from RAM or from external SPI memory. This firmware storage option is available under Project → Options → FET Debugger → Download as shown in [Figure 12](#).

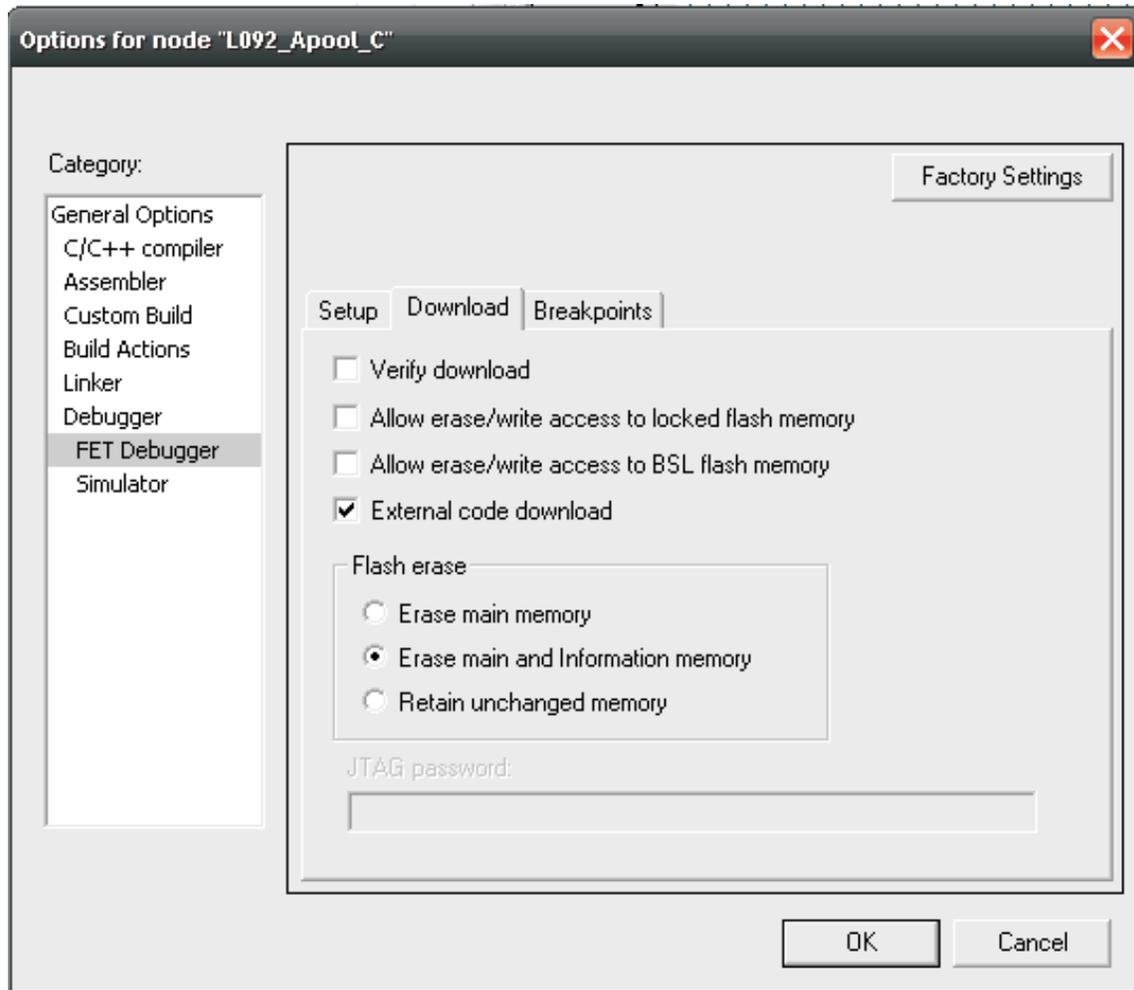


Figure 12. MSP430L092 Firmware Download Options [RAM/EEPROM] – IAR

For download and debug from RAM, uncheck the *External code download* checkbox.

For download and debug from EEPROM, check the *External code download* checkbox.

Additionally, when downloading to the EEPROM, the entire EEPROM memory must be filled. Therefore, one additional setting must be configured for the EEPROM option under Project → Options → Linker → Checksum. Check the *Fill unused code memory* checkbox and specify the fill pattern to 0xFF as shown in [Figure 13](#). This ensures proper firmware download to the EEPROM and subsequent read-back into the RAM memory of the MSP430L092.

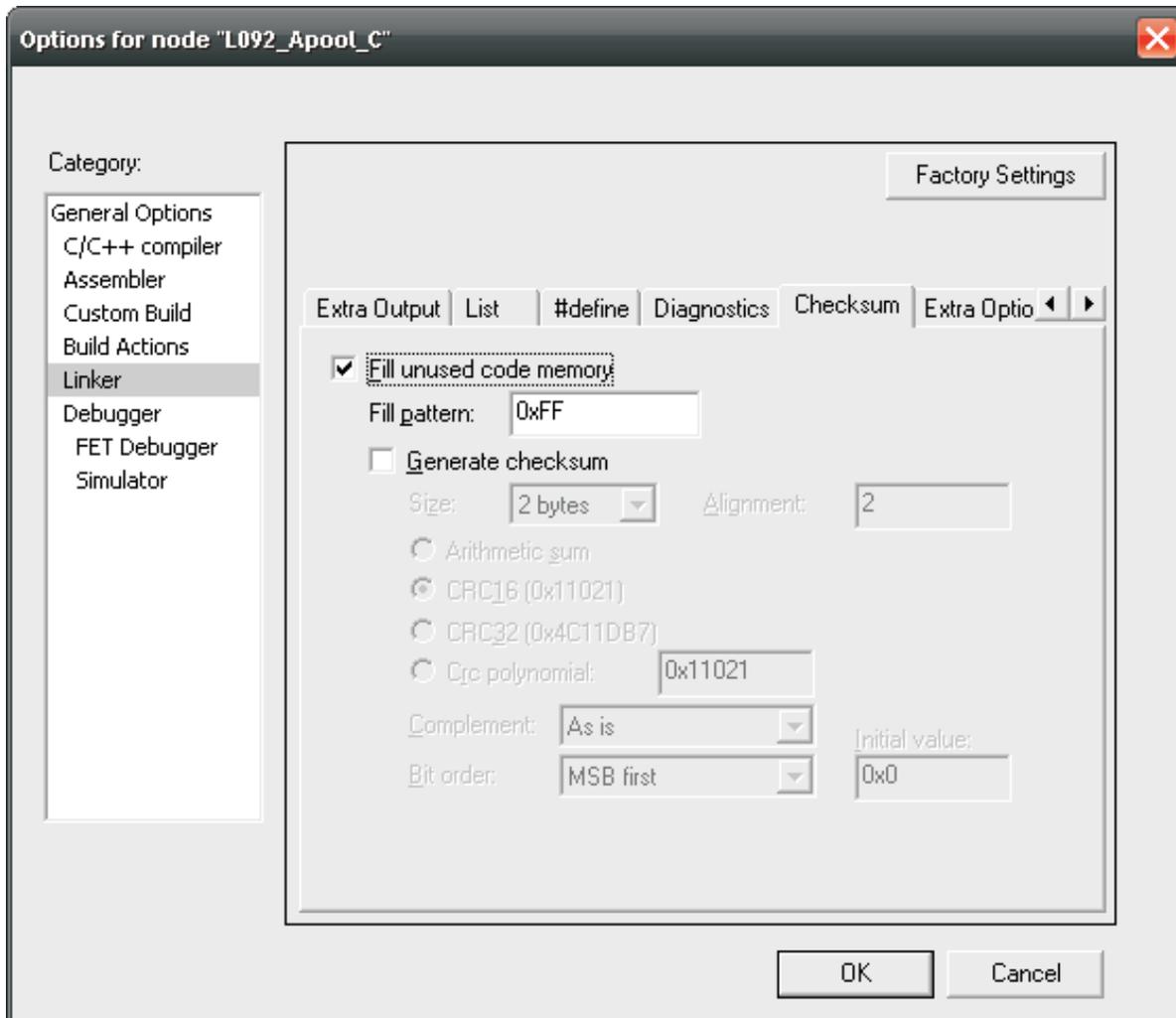


Figure 13. MSP430L092 EEPROM Download - Fill Memory Option in IAR

3.4 MSP430L092 Development in Code Composer Studio

MSP430L092 family device support is available in Code Composer Studio version 4.2 or later.

1. Go to Project → New Project in Code Composer Studio.
2. Select MSP430L092 from the drop-down menu in the MSP430 device selection screen. The device family filter can be used to shorten the list by selecting MSP430L0XX in the filter drop-down menu as shown in Figure 14.

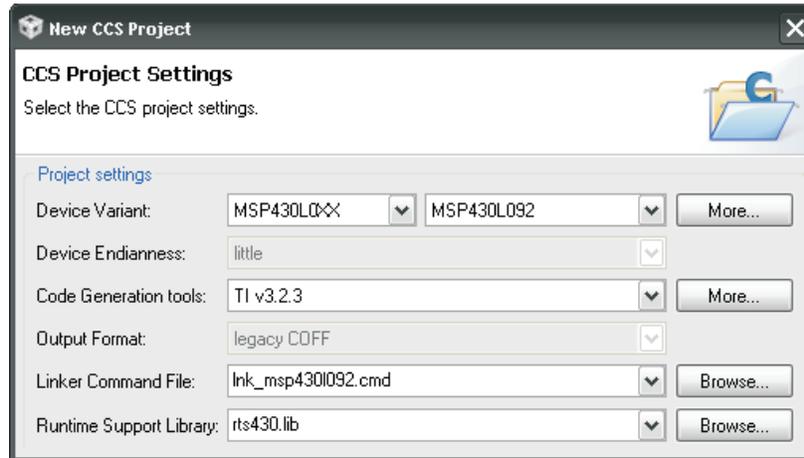


Figure 14. MSP430L092 Project Selection in Code Composer Studio

3.4.1 L092 and C092 Emulation Modes in Code Composer Studio

To specify between strict L092 and C092 emulation modes for an MSP430L092, open the active MSP430L092.ccxml file (see Figure 15) in Code Composer Studio (available inside the project folder and provided with Code Composer Studio after creating an MSP430L092 project).

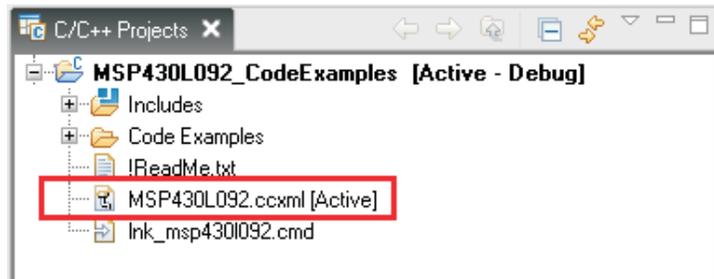


Figure 15. MSP430L092.ccxml in a Code Composer Studio MSP430L092 Project

Select *Target Configuration* under *Advanced Setup* as shown in [Figure 16](#).

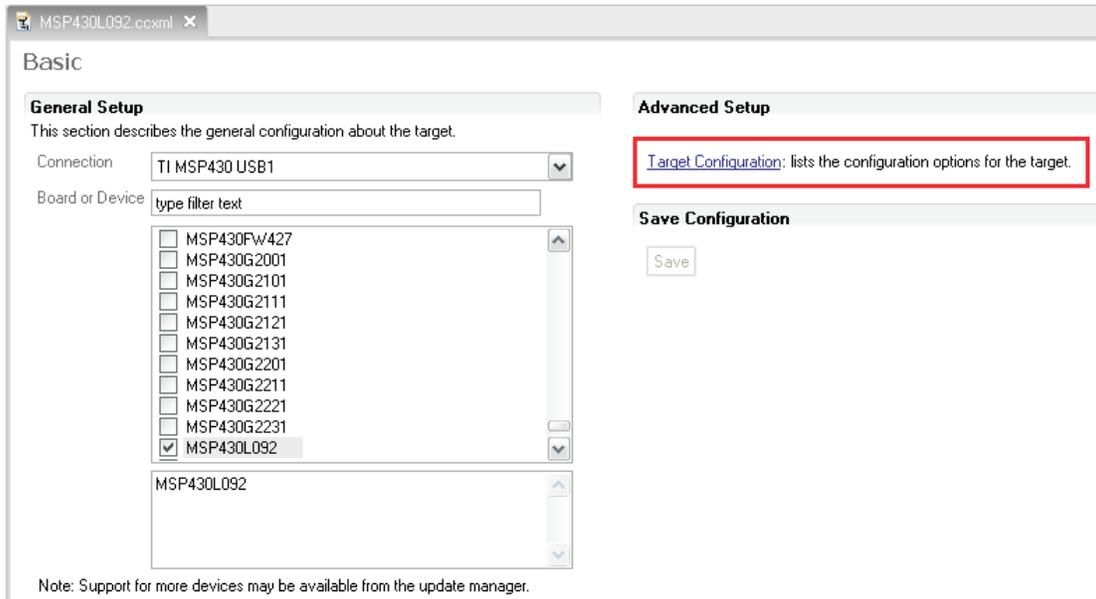


Figure 16. Advanced Setup → Target Configuration in Code Composer Studio

Once MSP430 is selected, [Figure 17](#) shows that the CPU properties section becomes available with the C092/L092 emulation options available under the emulation type drop-down menu.

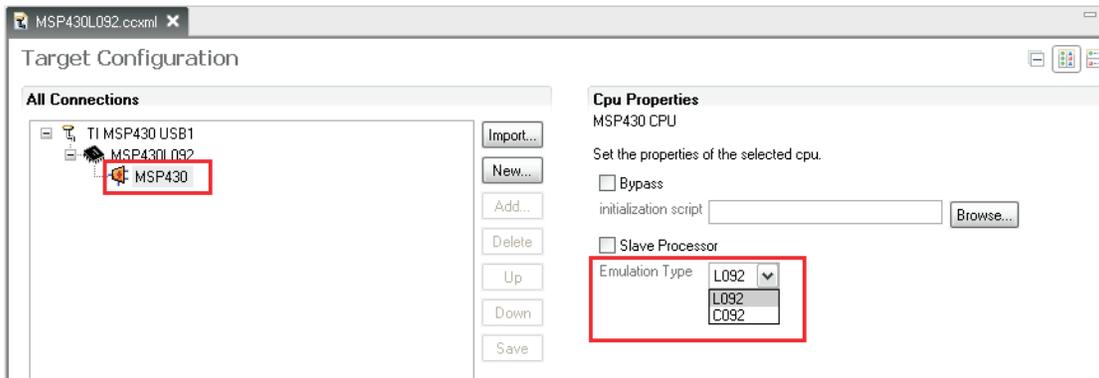


Figure 17. L092/C092 Emulation Type Selection in Code Composer Studio

- Single-cell application

3.4.2 RAM/External SPI Memory Download Option in Code Composer Studio

For download and debug from RAM, uncheck the *Copy application to external SPI memory after program load* checkbox as shown in Figure 18.

For download and debug from RAM, uncheck the *Copy application to external SPI memory after program load* checkbox as shown in Figure 18.

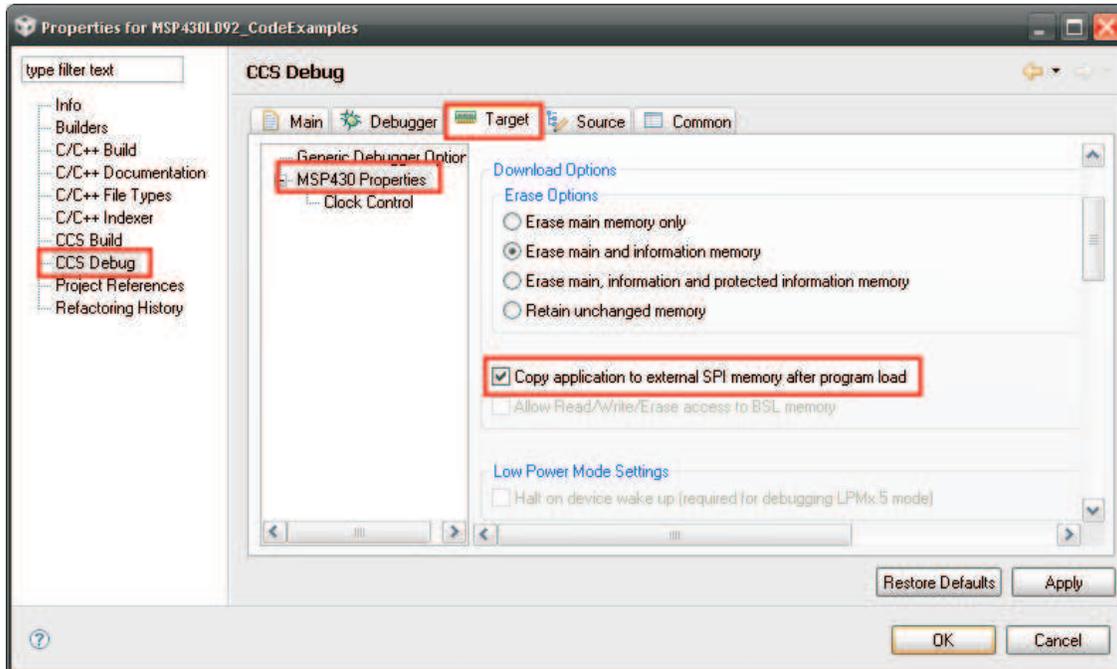


Figure 18. MSP430L092 Firmware Download Options [RAM/EEPROM] in Code Composer Studio

4 FAQs / Tips and Trick

- SYSCNF must be accessed bit-wise to prevent unwanted modification of MEMSWAP bit, which controls the memory mirroring from the MSP430L092 RAM to the MSP430C092 ROM address.
- Upon startup, all clocks (MCLK, SMCLK, ACLK) are configured to be sourced by the LF-OSC. This is due to the internal boot code and different potential startup combinations (BOR, reset, POR, etc.) Reconfiguration of the clocks to the desired setting for the application is recommended at the beginning of the application code.
- During debugging with JTAG, if writing to external EEPROM is not critical (i.e., the application does not need to run after power out) use download to RAM to minimize download time and significantly speed up debug time.
- On the MSP430L092 target board, some of the MSP430L092 pins are not available for application functions due to their dedicated functions for JTAG/SPI communication and PWM to drive the boost converter. These pins are P2.0-3 (JTAG for MSP-FET430UIF and software SPI for EEPROM) and P1.2 (PWM Boost).
However, P1.2 can be utilized for application purposes by removing JP2. Specifically, when downloading or debugging in RAM, JP2 can be removed completely. When downloading or debugging from EEPROM, JP2 can be removed after the firmware has been read back from the EEPROM into the MSP430L092. The blinking LEDs help indicate the status of the firmware download.

5 References

1. *MSP430x09x Family User's Guide* ([SLAU321](#))
2. *MSP430L092, MSP430C09x Mixed Signal Microcontroller Data Sheet* ([SLAS673](#))
3. MSP430L092 Target Board MSP-TS430L092:
<http://focus.ti.com/docs/toolsw/folders/print/msp-ts430l092.html>
4. Code Composer Studio (CCStudio) Integrated Development Environment (IDE) v4.x:
<http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html>
5. IAR Embedded Workbench Kickstart - Free 4KB IDE:
<http://focus.ti.com/docs/toolsw/folders/print/iar-kickstart.html>
6. Low Voltage Microcontrollers from MSP430: www.ti.com/lowvoltage/wiki
7. *MSP430L092 Loader Code User's Guide* ([SLAU324](#))
8. *Code Composer Studio v4.2 User's Guide for MSP430 User's Guide* ([SLAU157](#))

Appendix A MSP430L092 Target Board and Active Cable Schematics and Layouts

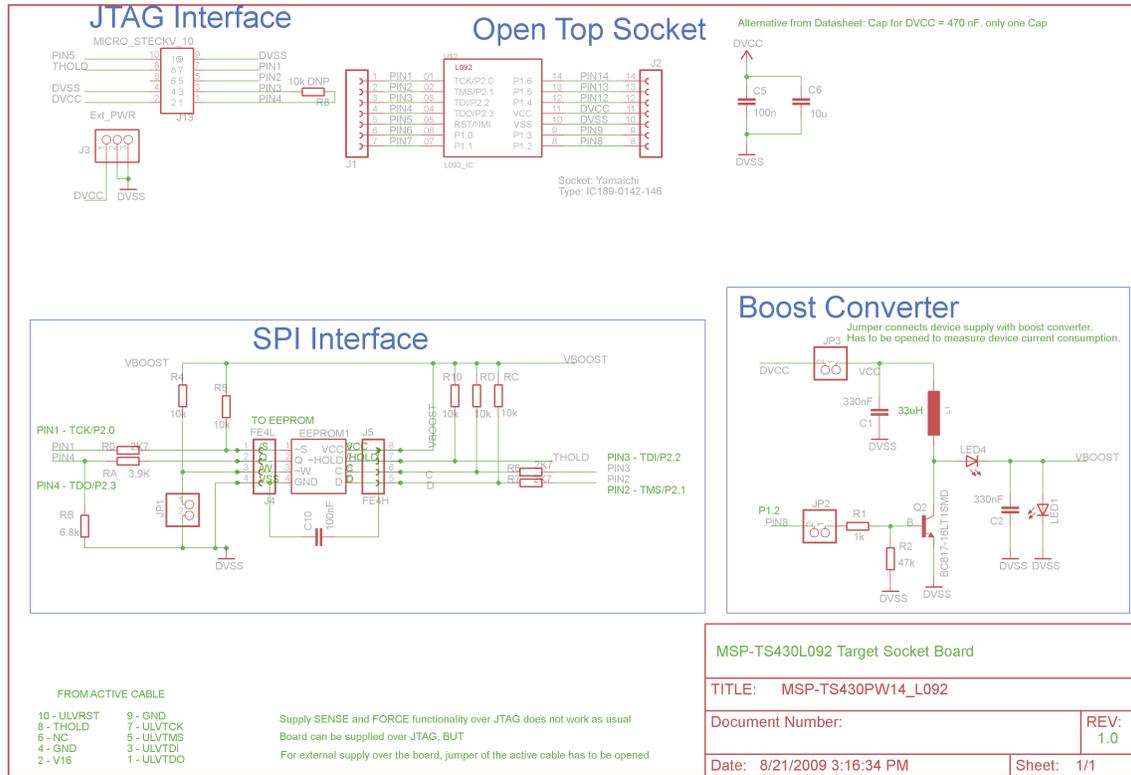


Figure 19. MSP430L092 Target Board Schematic

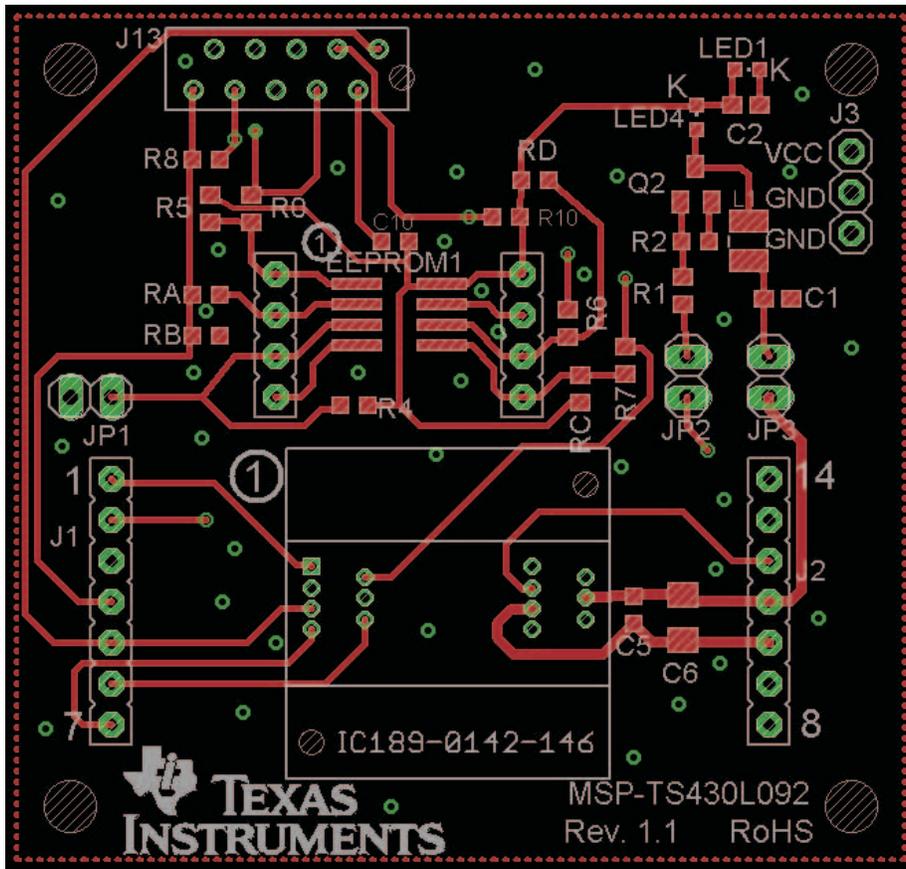


Figure 20. MSP430L092 Target Board Top Layer

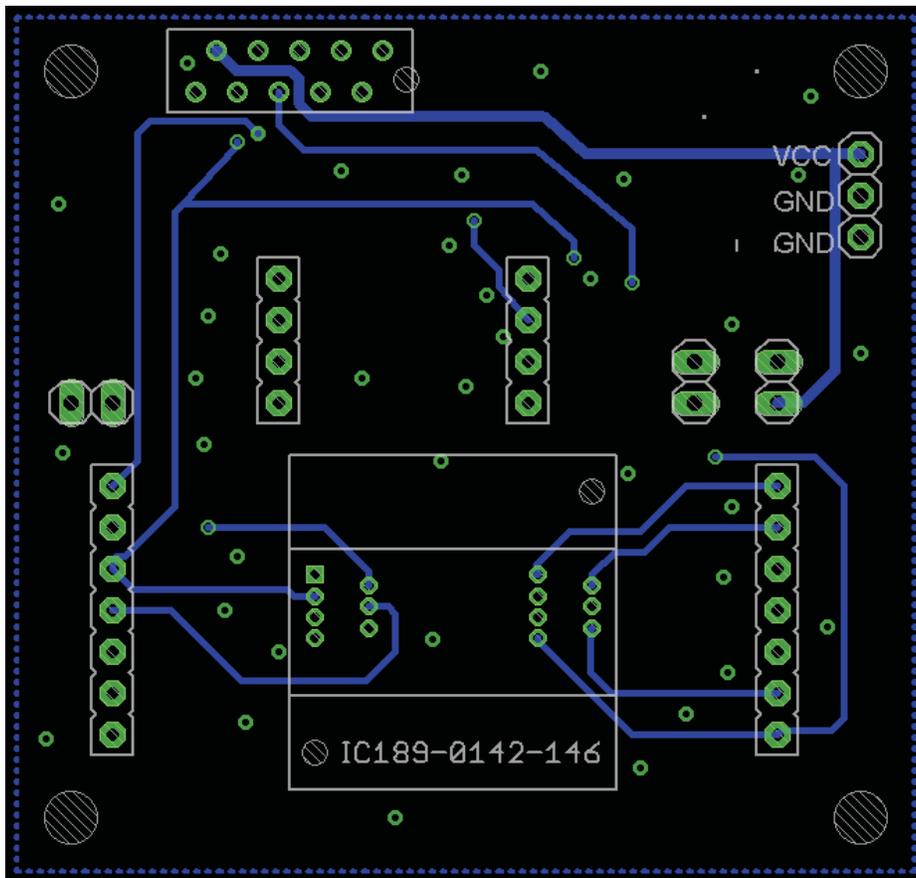


Figure 21. MSP430L092 Target Board Bottom Layer

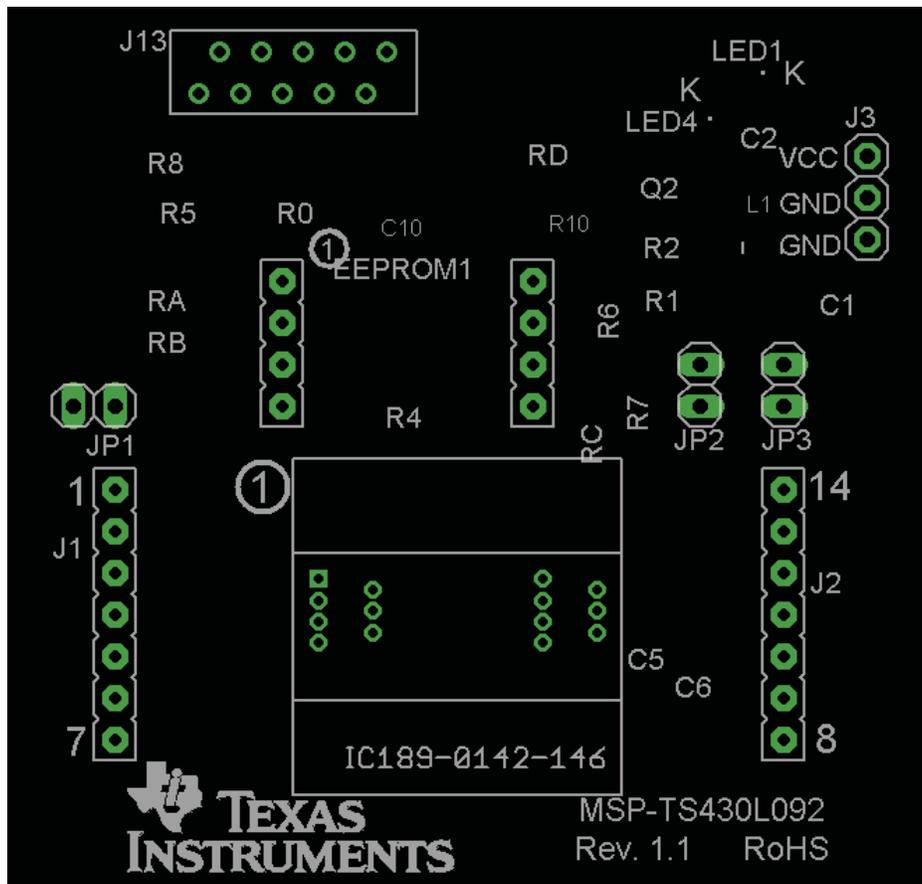


Figure 22. MSP430L092 Target Board Components and Through Holes

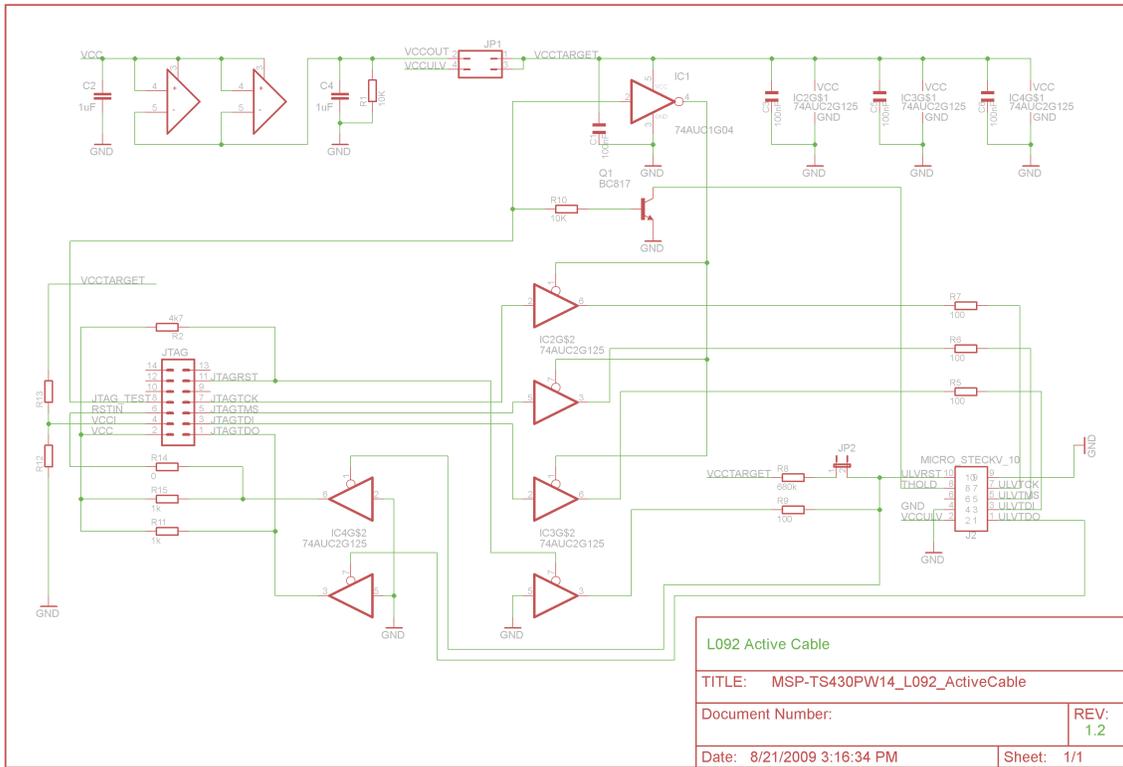


Figure 23. L092 Active Cable Schematic

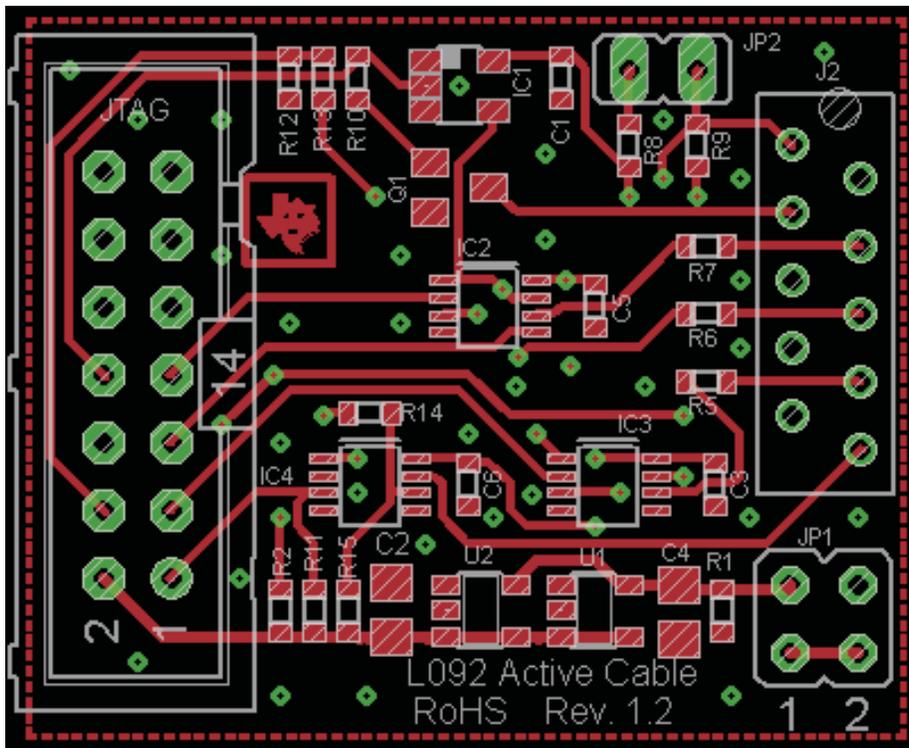


Figure 24. L092 Active Cable Top Layer

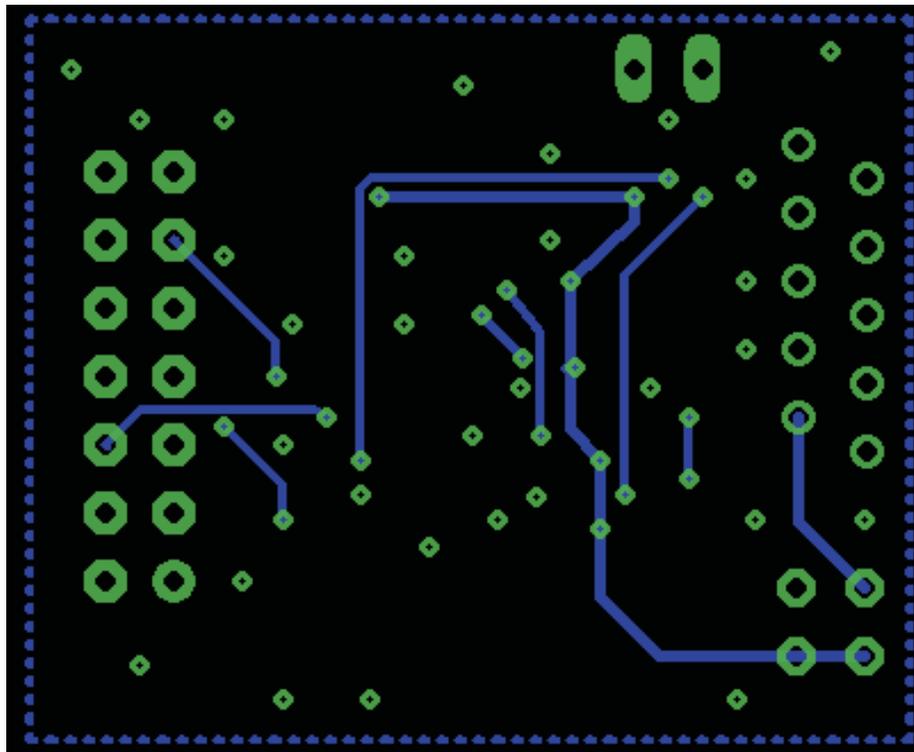


Figure 25. L092 Active Cable Bottom Layer

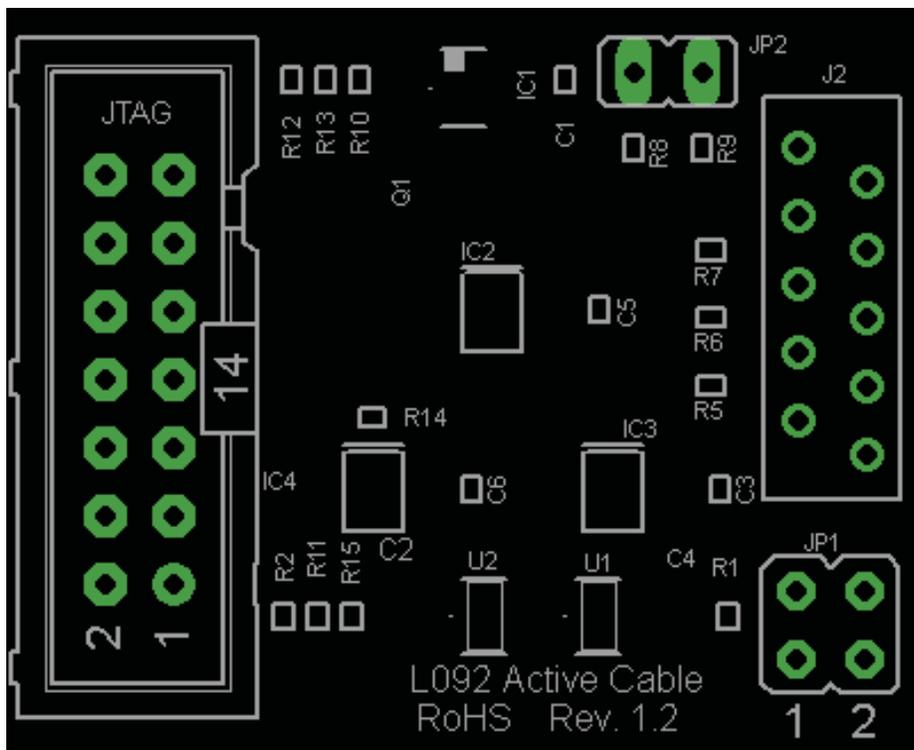


Figure 26. L092 Active Cable Components and Through Hole

Revision History

Changes from Original (December 2010) to A Revision Page

- Deleted Section 4, Finalize L092 Application Development, which contained no information [19](#)
-

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
RF/IF and ZigBee® Solutions	www.ti.com/lprf

Applications

Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Transportation and Automotive	www.ti.com/automotive
Video and Imaging	www.ti.com/video
Wireless	www.ti.com/wireless-apps

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated