User's Guide

SLAU590J–March 2015–Revised November 2018

Arm® Keil® MDK Version 5 for SimpleLink™ MSP432™ Microcontrollers

This user's guide describes the use of the Arm® Keil® MDK version 5 with the SimpleLink™ MSP432™ low-power microcontrollers (MCUs).

Contents

List of Figures

Trademarks

SimpleLink, MSP432, E2E, LaunchPad are trademarks of Texas Instruments.
Arm, Keil, µVision, CoreSight are registered trademarks of Arm Limited.
All other trademarks are the property of their respective owners.

Preface: Read This First

How to Use This User's Guide

This user's guide describes only the features of the Keil MDK 5 that are specific to the MSP432 low-power microcontrollers. It does not fully describe the MSP432 microcontrollers or the complete development software and hardware systems. For details on these items, see the appropriate TI documents listed in Important MSP432 MCU Documents on the Web.

Important MSP432 MCU Documents on the Web

The primary sources of information on the MSP432 MCU family are the device-specific data sheets and user's guides. The MSP432 website contains the most recent version of these documents.

Documents that describe the Keil MDK 5 IDE can be found on www.keil.com. The Texas Instruments E2E™ Community support forums at e2e.ti.com can provide additional help.

Documentation for third-party tools, such as the Keil ULINK2 and ULINK Pro debug probes or the Segger J-Link debug probe, can usually be found on the respective third-party website.

If You Need Assistance

Support for the MSP432 microcontrollers and the hardware development tools is provided by the Texas Instruments Product Information Center (PIC). Contact information for the PIC can be found on the TI website at www.ti.com/support. The TI E2E Community support forums for the MSP432 MCUs provide open interaction with peer engineers, TI engineers, and other experts. Additional device-specific information can be found on the MSP432 website.

1 Installing Keil µVision 5 IDE

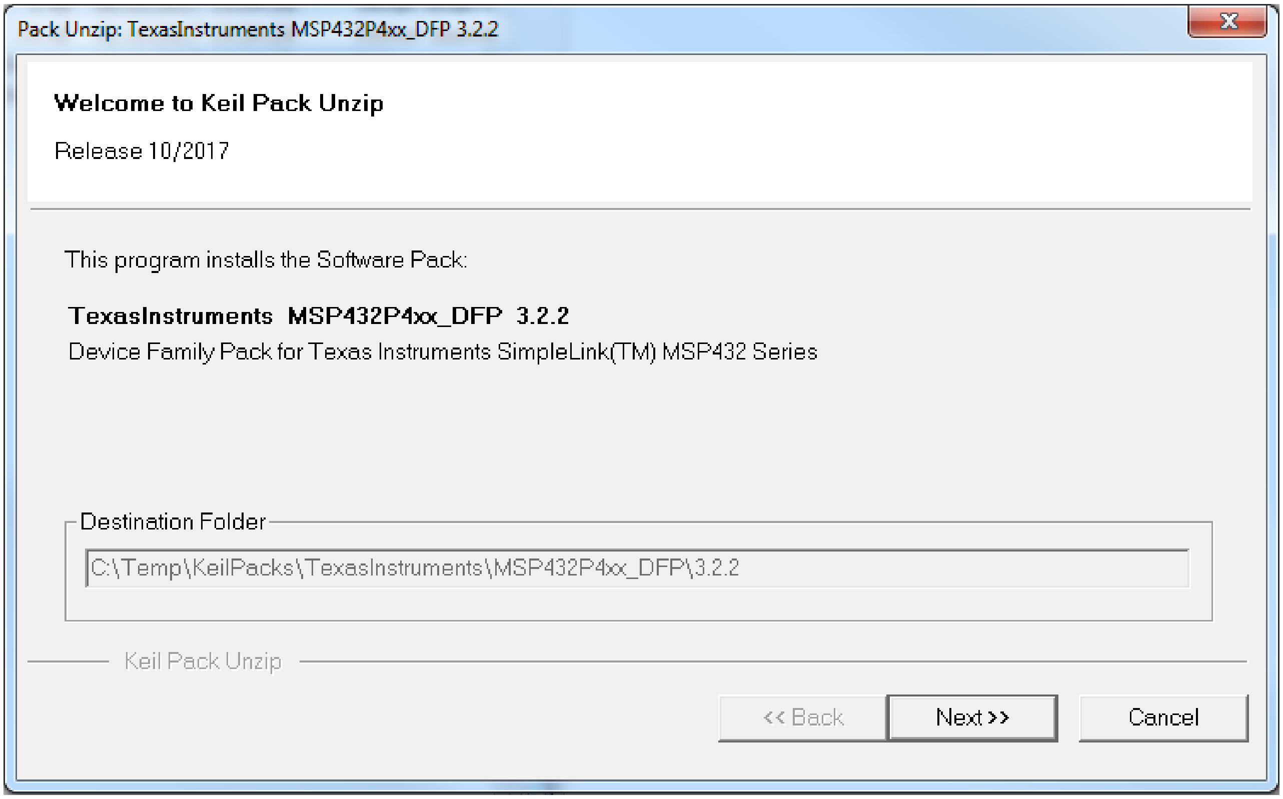
The Keil µVision® 5 IDE can be obtained from the Keil website. MSP432 low-power microcontrollers are supported by Keil µVision 5 and higher versions. Previous versions do not support MSP432 MCUs and might not be compatible with MSP432 Device Family Packs.

2 Installing Device Family Packs

The Keil µVision 5 IDE does not come with preinstalled device support. To add support for MSP432 MCUs to the IDE, either download an MSP432 Device Family Pack from the TI website (see Section 2.1), or install the pack using the Keil Pack Installer (see Section 2.2).

2.1 Installing Using a Downloaded SimpleLink MSP432P4xx Device Family Pack

To download this pack, visit the MSP432 Device Family Pack page and download the file named TexasInstruments.MSP432P4xx_DFP.<version>.pack, where <version> is the revision of the device family pack. When downloaded completes, double click on the file. The Keil Pack Unzip tool starts and guides you through the installation process.



Pack Unzip: TexasInstruments MSP432P4xx_DFP 3.2.2

Welcome to Keil Pack Unzip

Release 10/2017

This program installs the Software Pack:

TexasInstruments MSP432P4xx_DFP 3.2.2

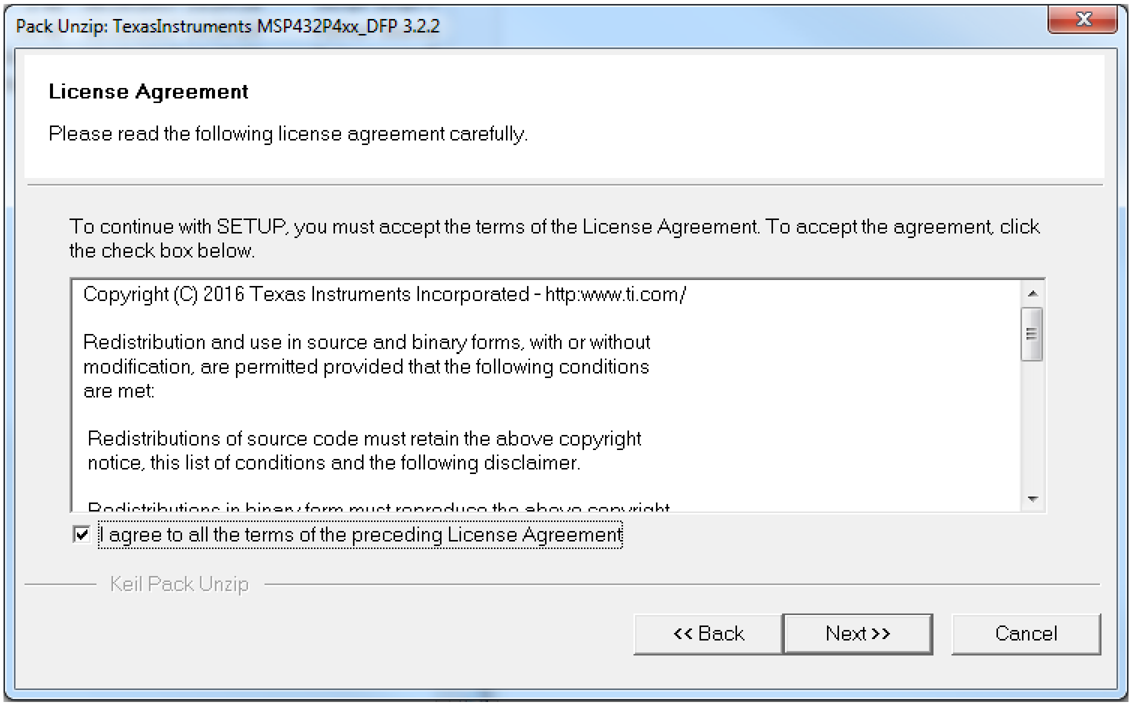
Device Family Pack for Texas Instruments SimpleLink(TM) MSP432 Series

Destination Folder: C:\Temp\KeilPacks\TexasInstruments\MSP432P4xx_DFP\3.2.2

Figure 1. Keil Pack Unzip Tool

Click **Next** to display the TI License Agreement (see Figure 2).

Take the time to read the agreement before accepting it, then click **Next**.



Pack Unzip: TexasInstruments MSP432P4xx_DFP 3.2.2

License Agreement

Please read the following license agreement carefully.

To continue with SETUP, you must accept the terms of the License Agreement. To accept the agreement, click the check box below.

Copyright (C) 2016 Texas Instruments Incorporated - http:www.ti.com/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

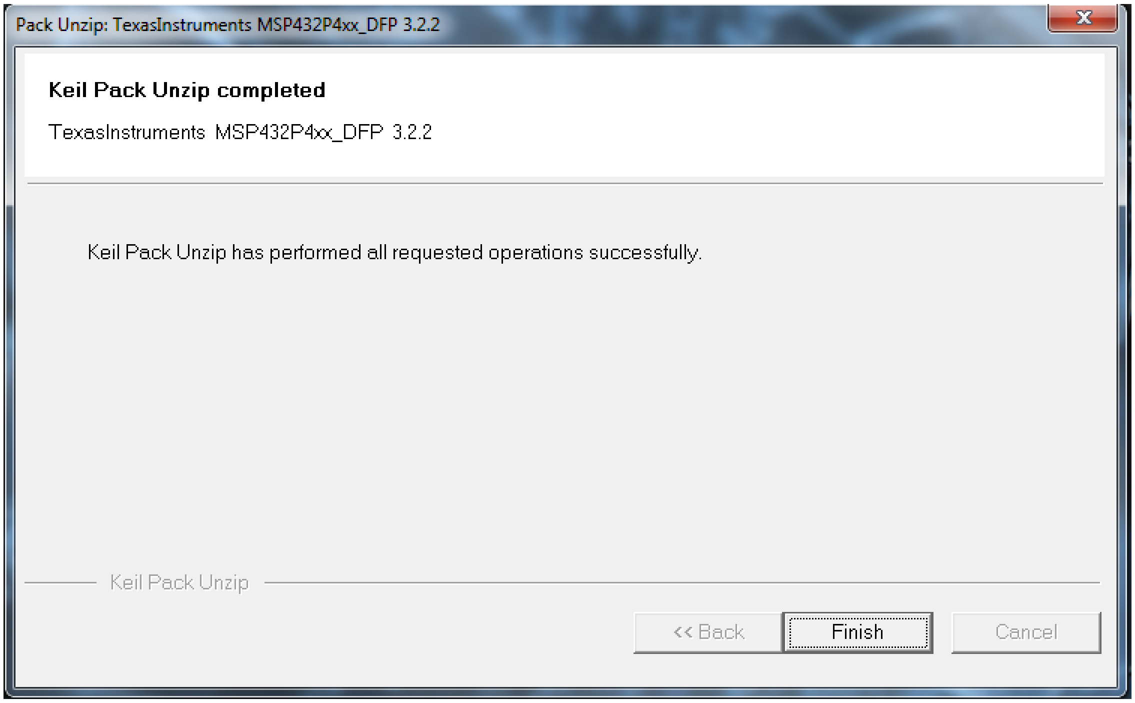
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright

☑ I agree to all the terms of the preceding License Agreement

Figure 2. License Agreement

After the Pack Unzip tool has copied the device family files into the µVision IDE installation directory, the Pack Unzip tool finishes (see Figure 3).



Pack Unzip: TexasInstruments MSP432P4xx_DFP 3.2.2

Keil Pack Unzip completed

TexasInstruments MSP432P4xx_DFP 3.2.2

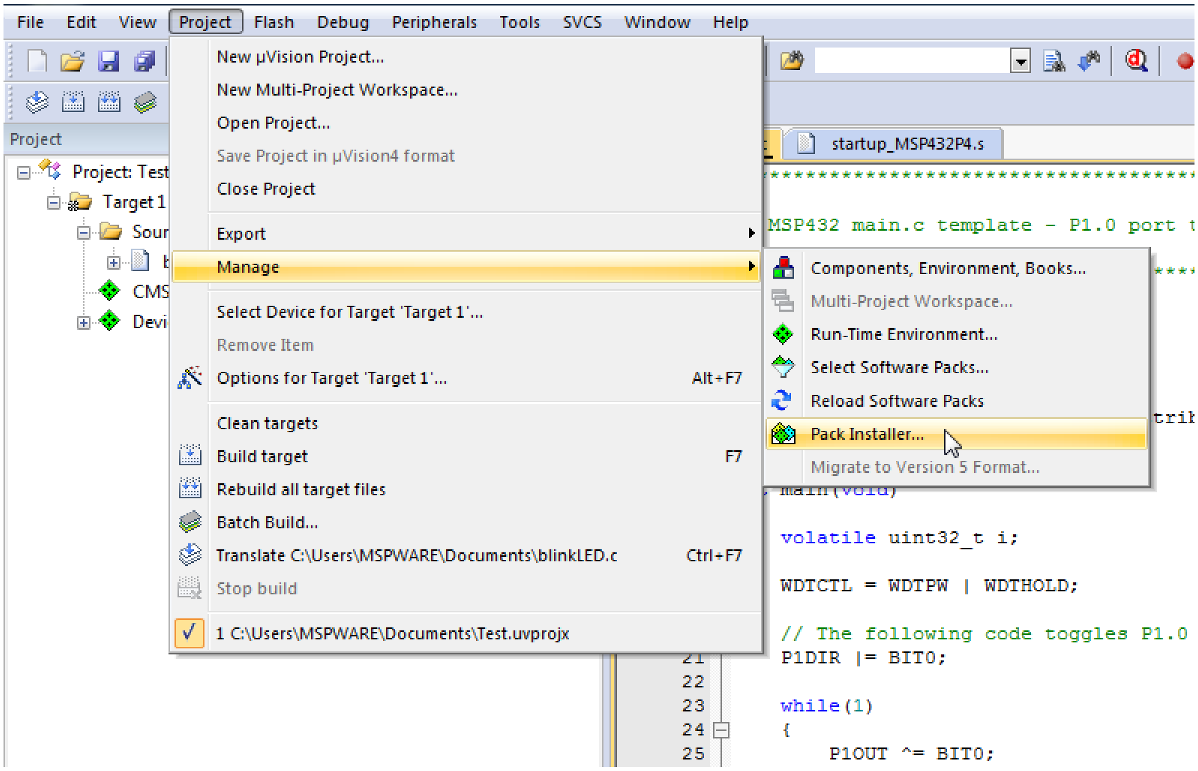
Keil Pack Unzip has performed all requested operations successfully.

Figure 3. Device Family Pack Has Been Installed Successfully

If the tool reports an error, check whether or not the Device Family Pack has been downloaded correctly, and whether or not the µVision IDE directory is writeable.

2.2 Installing Using the Keil Pack Installer

To install using the Keil Pack Installer, open the Keil µVision IDE. To launch the Keil Pack Installer, go to **Project** → **Manage** → **Pack Installer** (see Figure 4).

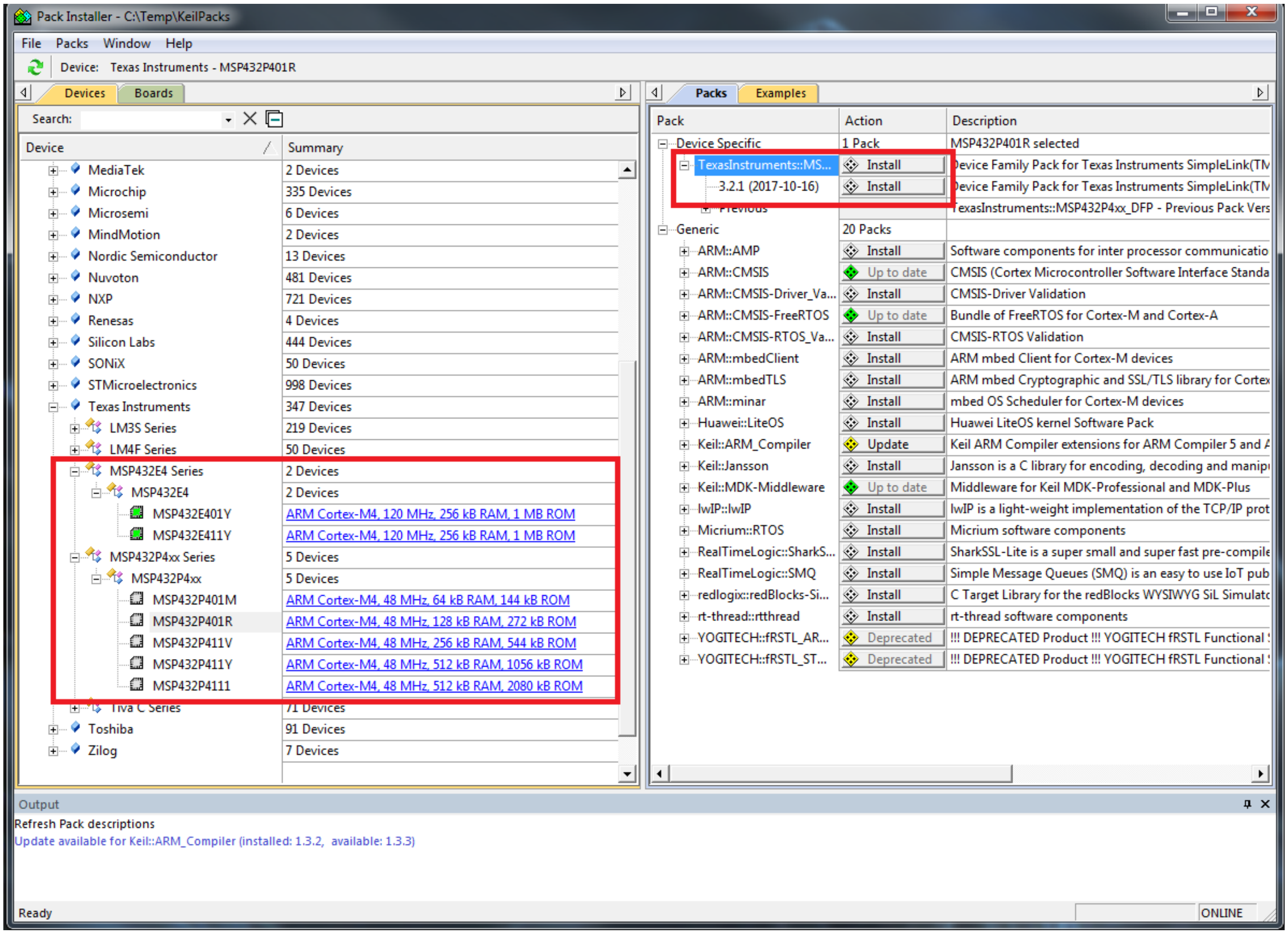


```
MSP432 main.c template - P1.0 port t

    volatile uint32_t i;

    WDTCTL = WDTPW | WDTHOLD;

    // The following code toggles P1.0
    P1DIR |= BIT0;

    while(1)
    {
        P1OUT ^= BIT0;
```

Figure 4. Launching the Pack Installer

When the pack installer has been launched, locate Texas Instruments::MSP432P4xx Series in the devices listing and select the newest version of the associated Texas Instruments::MSP432P4xx_DFP pack (see Figure 5).

Click **Install** to let the IDE download and install the MSP432P4xx Device Family Pack into the IDE.



Device: Texas Instruments - MSP432P401R

Device	Summary
MediaTek	2 Devices
Microchip	335 Devices
Microsemi	6 Devices
MindMotion	2 Devices
Nordic Semiconductor	13 Devices
Nuvoton	481 Devices
NXP	721 Devices
Renesas	4 Devices
Silicon Labs	444 Devices
SONiX	50 Devices
STMicroelectronics	998 Devices
Texas Instruments	347 Devices
LM3S Series	219 Devices
LM4F Series	50 Devices
MSP432E4 Series	2 Devices
MSP432E4	2 Devices
MSP432E401Y	ARM Cortex-M4, 120 MHz, 256 kB RAM, 1 MB ROM
MSP432E411Y	ARM Cortex-M4, 120 MHz, 256 kB RAM, 1 MB ROM
MSP432P4xx Series	5 Devices
MSP432P4xx	5 Devices
MSP432P401M	ARM Cortex-M4, 48 MHz, 64 kB RAM, 144 kB ROM
MSP432P401R	ARM Cortex-M4, 48 MHz, 128 kB RAM, 272 kB ROM
MSP432P411V	ARM Cortex-M4, 48 MHz, 256 kB RAM, 544 kB ROM
MSP432P411Y	ARM Cortex-M4, 48 MHz, 512 kB RAM, 1056 kB ROM
MSP432P4111	ARM Cortex-M4, 48 MHz, 512 kB RAM, 2080 kB ROM
Tiva C Series	71 Devices
Toshiba	91 Devices
Zilog	7 Devices

Pack	Action	Description
Device Specific	1 Pack	MSP432P401R selected
TexasInstruments::MS...	Install	Device Family Pack for Texas Instruments SimpleLink(TM
3.2.1 (2017-10-16)	Install	Device Family Pack for Texas Instruments SimpleLink(TM
Previous		TexasInstruments::MSP432P4xx_DFP - Previous Pack Vers
Generic	20 Packs	
ARM::AMP	Install	Software components for inter processor communicatio
ARM::CMSIS	Up to date	CMSIS (Cortex Microcontroller Software Interface Standa
ARM::CMSIS-Driver_Va...	Install	CMSIS-Driver Validation
ARM::CMSIS-FreeRTOS	Up to date	Bundle of FreeRTOS for Cortex-M and Cortex-A
ARM::CMSIS-RTOS_Va...	Install	CMSIS-RTOS Validation
ARM::mbedClient	Install	ARM mbed Client for Cortex-M devices
ARM::mbedTLS	Install	ARM mbed Cryptographic and SSL/TLS library for Cortex
ARM::minar	Install	mbed OS Scheduler for Cortex-M devices
Huawei::LiteOS	Install	Huawei LiteOS kernel Software Pack
Keil::ARM_Compiler	Update	Keil ARM Compiler extensions for ARM Compiler 5 and A
Keil::Jansson	Install	Jansson is a C library for encoding, decoding and manipu
Keil::MDK-Middleware	Up to date	Middleware for Keil MDK-Professional and MDK-Plus
lwIP::lwIP	Install	lwIP is a light-weight implementation of the TCP/IP prot
Micrium::RTOS	Install	Micrium software components
RealTimeLogic::SharkS...	Install	SharkSSL-Lite is a super small and super fast pre-compile
RealTimeLogic::SMQ	Install	Simple Message Queues (SMQ) is an easy to use IoT pub
redlogix::redBlocks-Si...	Install	C Target Library for the redBlocks WYSIWYG SiL Simulato
rt-thread::rtthread	Install	rt-thread software components
YOGITECH::fRSTL_AR...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional !
YOGITECH::fRSTL_ST...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional !

```
Refresh Pack descriptions
Update available for Keil::ARM_Compiler (installed: 1.3.2,  available: 1.3.3)
```

Ready ONLINE

Figure 5. Installing a MSP432 Device Family Pack

If the tool reports an error, check whether the µVision IDE directory is writeable.

3 Creating a SimpleLink MSP432 MCU Project From Scratch

When starting µVision for the first time, create a project for the MSP432 MCU. Go to **Project → New µVision Project** and click on the selection (see Figure 6).

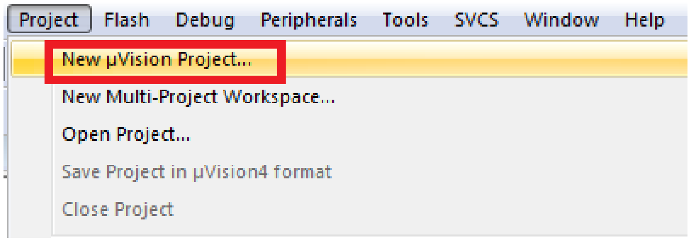
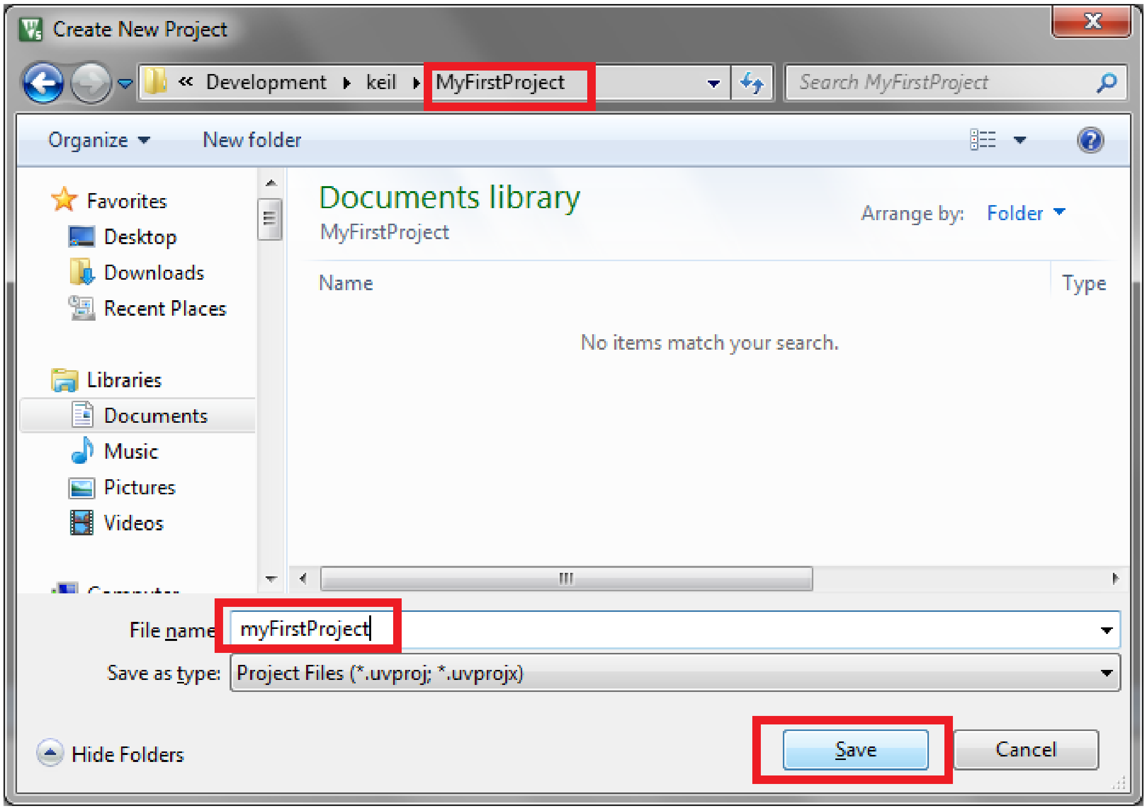

Project Flash Debug Peripherals Tools SVCS Window Help
New µVision Project...
New Multi-Project Workspace...
Open Project...
Save Project in µVision4 format
Close Project

Figure 6. Create New Project

A file dialog (see Figure 7) is displayed to prompt for the project name and location. Navigate to an appropriate directory on the computer, and then enter the project name (for example, myFirstProject). Click the **Save** button to create the project files.

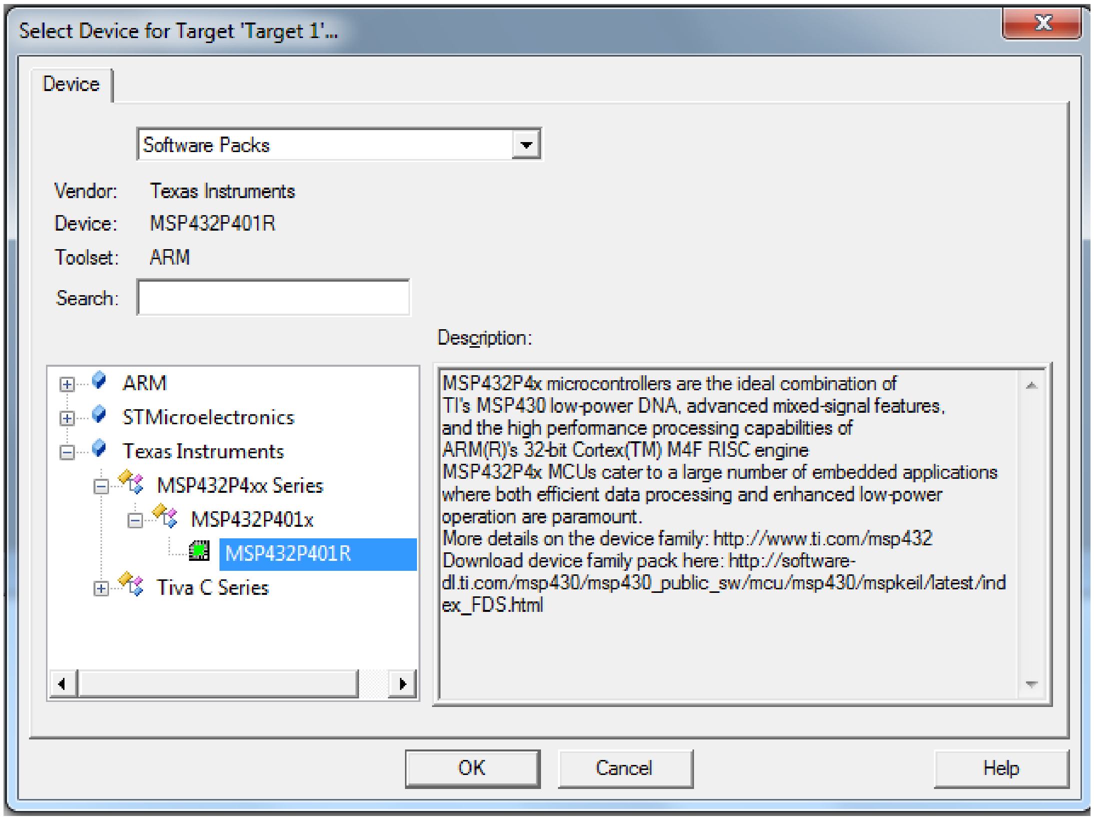


File name: myFirstProject

Save as type: Project Files (*.uvproj; *.uvprojx)

Figure 7. Assign Project Name

After the project has been created, µVision IDE prompts you to select the device to use, so that it can apply device-specific settings to the project. Go to **Texas Instruments** → **MSP432xx Series** → **MSP432P401x** and select the appropriate device from the list of available devices (see Figure 8).



Software Packs

Vendor: Texas Instruments
Device: MSP432P401R
Toolset: ARM
Search:

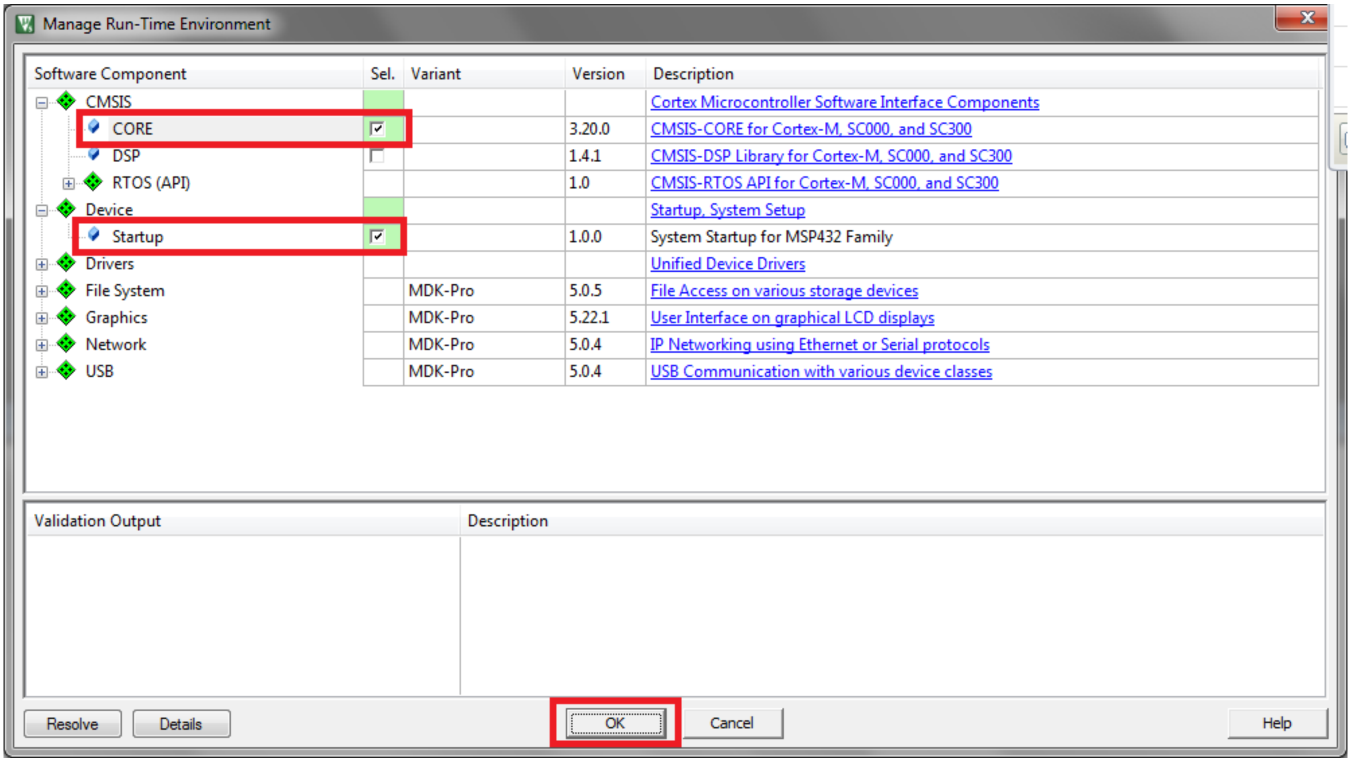
Description:
MSP432P4x microcontrollers are the ideal combination of
TI's MSP430 low-power DNA, advanced mixed-signal features,
and the high performance processing capabilities of
ARM(R)'s 32-bit Cortex(TM) M4F RISC engine
MSP432P4x MCUs cater to a large number of embedded applications
where both efficient data processing and enhanced low-power
operation are paramount.
More details on the device family: http://www.ti.com/msp432
Download device family pack here: http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/mspkeil/latest/index_FDS.html

Figure 8. Select MSP432P401R Device

After the device has been selected, the run-time environment components can be added to the project (see Figure 9). If this step is skipped, the components can be added at a later point, or when additional components have been added through specific Software Packs, through the **Manage Run-Time Environment** button in the µVision IDE.

Run-Time Environment components include device startup and header files, and Cortex-M generic functions and definitions that are part of the Arm CMSIS package.

To develop code for the MSP432 MCUs, the selections for **CMSIS** → **CORE** and **Device** → **Startup** need to be enabled. Others are optional.



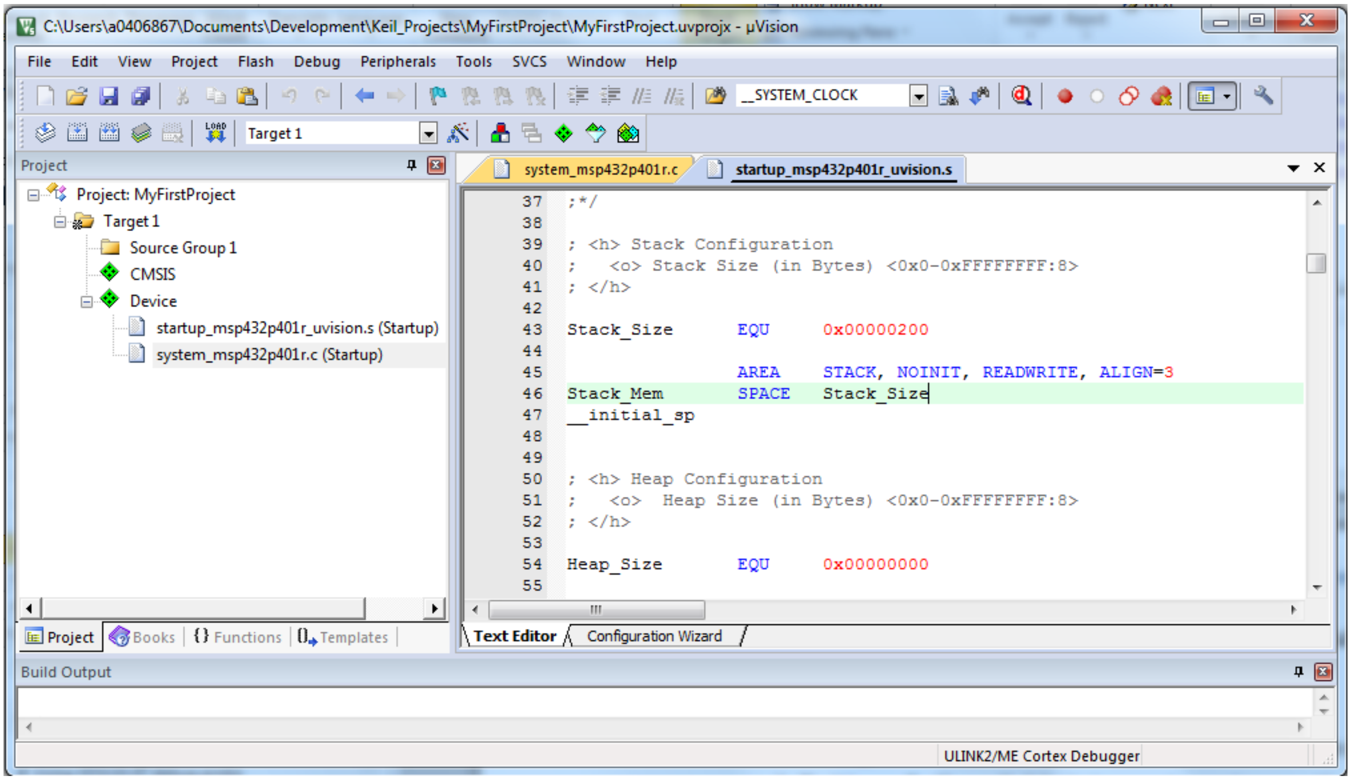
Software Component	Sel.	Variant	Version	Description
CMSIS				Cortex Microcontroller Software Interface Components
CORE	☑		3.20.0	CMSIS-CORE for Cortex-M, SC000, and SC300
DSP	□		1.4.1	CMSIS-DSP Library for Cortex-M, SC000, and SC300
RTOS (API)			1.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
Device				Startup, System Setup
Startup	☑		1.0.0	System Startup for MSP432 Family
Drivers				Unified Device Drivers
File System		MDK-Pro	5.0.5	File Access on various storage devices
Graphics		MDK-Pro	5.22.1	User Interface on graphical LCD displays
Network		MDK-Pro	5.0.4	IP Networking using Ethernet or Serial protocols
USB		MDK-Pro	5.0.4	USB Communication with various device classes

Validation Output	Description

Figure 9. Add Run-Time Environment Components to the Project

When you have finished the wizard by clicking the **OK** button, µVision will show the newly created project containing two source code files:

- startup_MSP432p401r_uvision.s: Assembler file containing the interrupt vector table and a definition of all available interrupt service routines, including the reset vector
- system_ MSP432p401r_uvision.c: C file containing low-level device initialization functions, including a watchdog timer halt command


C:\Users\a0406867\Documents\Development\Keil_Projects\MyFirstProject\MyFirstProject.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
__SYSTEM_CLOCK
Target 1
Project
Project: MyFirstProject
Target 1
Source Group 1
CMSIS
Device
startup_msp432p401r_uvision.s (Startup)
system_msp432p401r.c (Startup)
system_msp432p401r.c
startup_msp432p401r_uvision.s
Project
Books
{} Functions
0→ Templates
Text Editor
Configuration Wizard
Build Output
ULINK2/ME Cortex Debugger

```
;*/

; <h> Stack Configuration
;   <o> Stack Size (in Bytes) <0x0-0xFFFFFFFF:8>
; </h>

Stack_Size      EQU     0x00000200

                AREA    STACK, NOINIT, READWRITE, ALIGN=3
Stack_Mem       SPACE   Stack_Size
__initial_sp


; <h> Heap Configuration
;   <o>  Heap Size (in Bytes) <0x0-0xFFFFFFFF:8>
; </h>

Heap_Size       EQU     0x00000000
```

Figure 10. Empty Project

To start code development, a main() function is required. To create a source code file, right-click on **Source Group 1**, then select **Add New Item to Group** (see Figure 11).

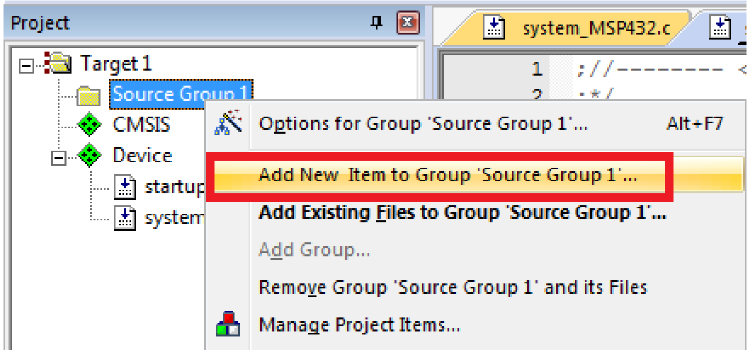
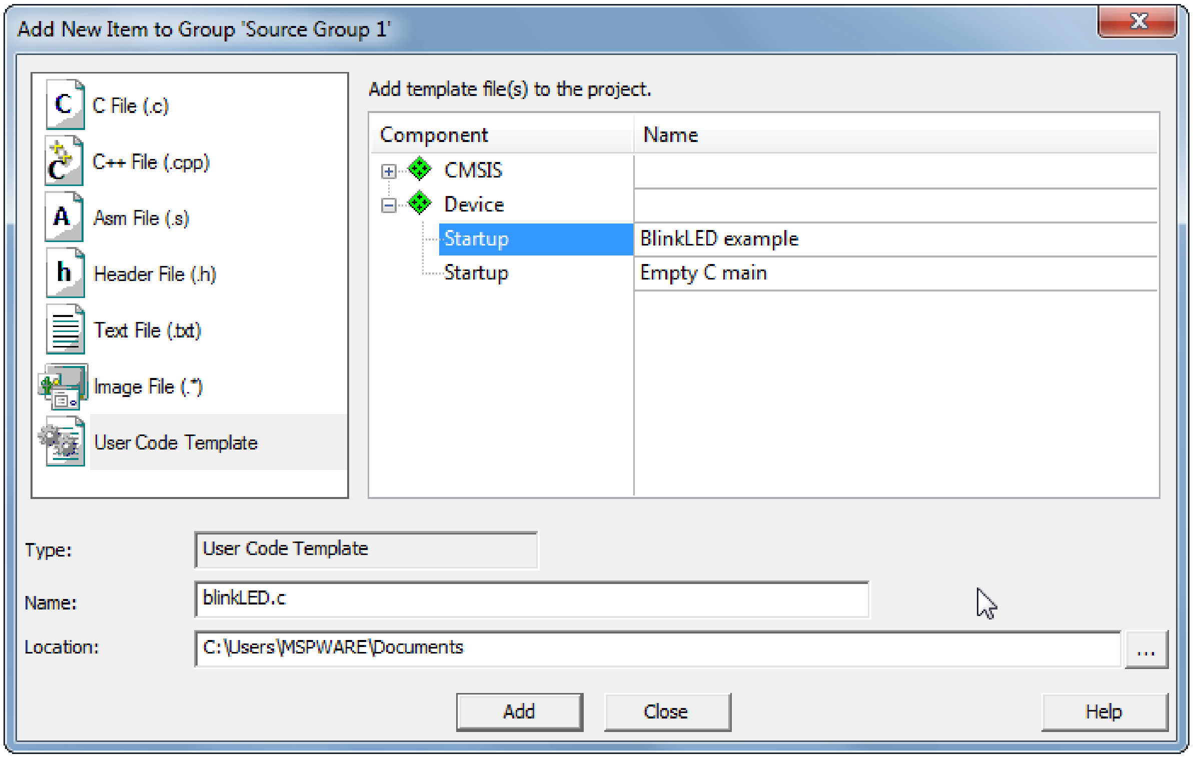

Project
Target 1
Source Group 1
CMSIS
Device
startup
system
system_MSP432.c
1 ;//---------- <
2 *-/
Options for Group 'Source Group 1'... Alt+F7
Add New Item to Group 'Source Group 1'...
Add Existing Files to Group 'Source Group 1'...
Add Group...
Remove Group 'Source Group 1' and its Files
Manage Project Items...

Figure 11. Add New Item to Project

A wizard opens in which several types of files can be created. Along with the MSP432 Device Family Pack, User Code Templates have been provided to select between the BlinkLED code and empty main code. The templates include the msp.h family header file and use C header files that provide MSP430 backward compatibility.



Add template file(s) to the project.

Component	Name
CMSIS	
Device	
Startup	BlinkLED example
Startup	Empty C main

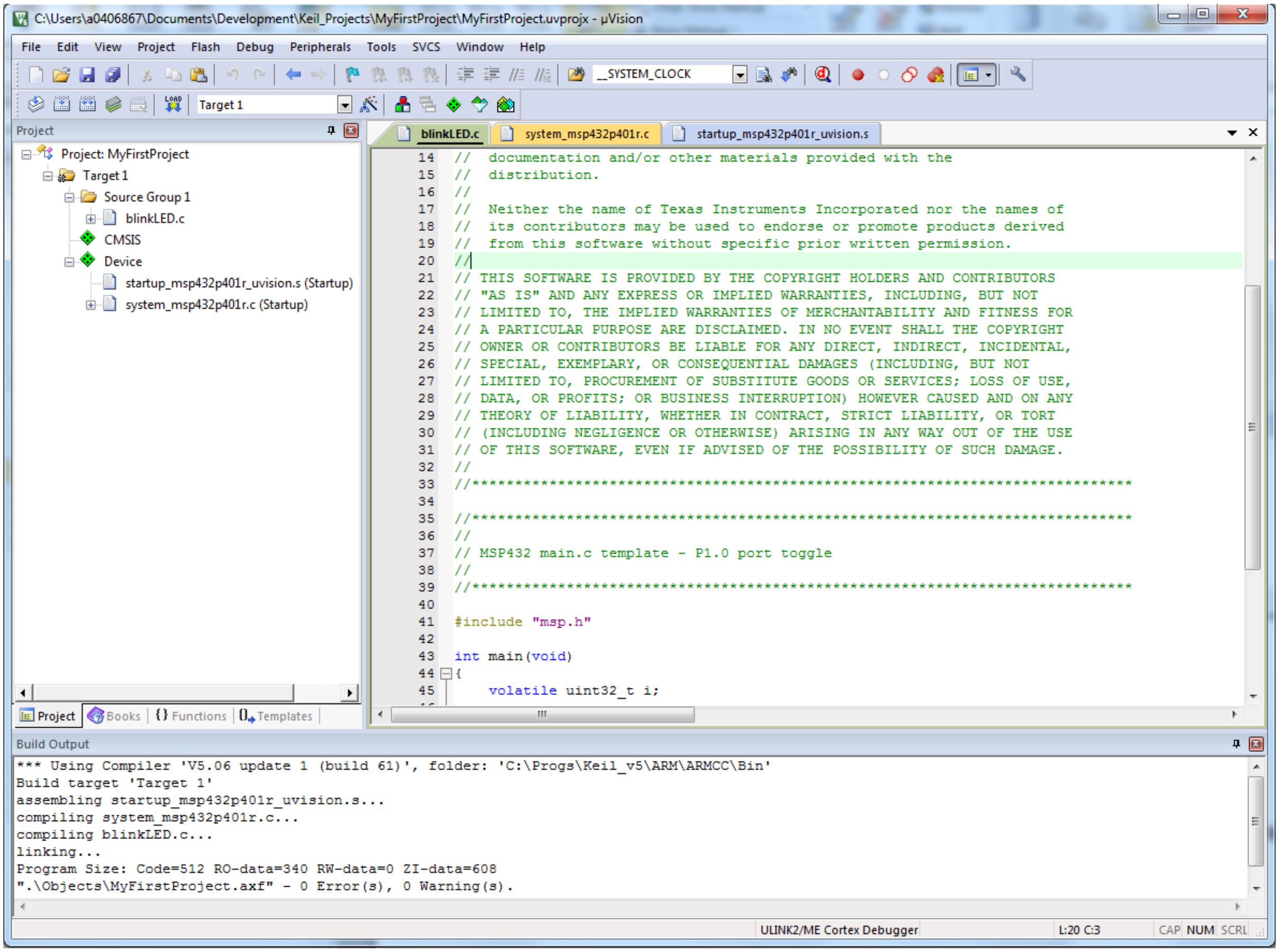
Type: User Code Template

Name: blinkLED.c

Location: C:\Users\MSPWARE\Documents

Figure 12. Add User Code Template

After adding a user code template to the code, the project is ready for compilation. Right-click on the project and select **Rebuild all target files** from the context menu, or press **F7** to rebuild all target files. If the project has been set up correctly, a compiler and linker output similar to the one shown in Figure 13 is displayed.


C:\Users\a0406867\Documents\Development\Keil_Projects\MyFirstProject\MyFirstProject.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
__SYSTEM_CLOCK
Target 1
Project
Project: MyFirstProject
Target 1
Source Group 1
blinkLED.c
CMSIS
Device
startup_msp432p401r_uvision.s (Startup)
system_msp432p401r.c (Startup)
Project
Books
{} Functions
0→ Templates
blinkLED.c
system_msp432p401r.c
startup_msp432p401r_uvision.s
Build Output
ULINK2/ME Cortex Debugger
L:20 C:3
CAP NUM SCRL

```
//  documentation and/or other materials provided with the
//  distribution.
//
//  Neither the name of Texas Instruments Incorporated nor the names of
//  its contributors may be used to endorse or promote products derived
//  from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
//
//****************************************************************************

//****************************************************************************
//
// MSP432 main.c template - P1.0 port toggle
//
//****************************************************************************

#include "msp.h"

int main(void)
{
    volatile uint32_t i;
```

```
*** Using Compiler 'V5.06 update 1 (build 61)', folder: 'C:\Progs\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
assembling startup_msp432p401r_uvision.s...
compiling system_msp432p401r.c...
compiling blinkLED.c...
linking...
Program Size: Code=512 RO-data=340 RW-data=0 ZI-data=608
".\Objects\MyFirstProject.axf" - 0 Error(s), 0 Warning(s).
```

Figure 13. Compiling the Project

The project is now ready for upload to the device.

4 Opening an Existing SimpleLink MSP432 MCU Project

To open a project first launch the µVision IDE. After the IDE launches click **Project** → **Open Project** (see Figure 14).

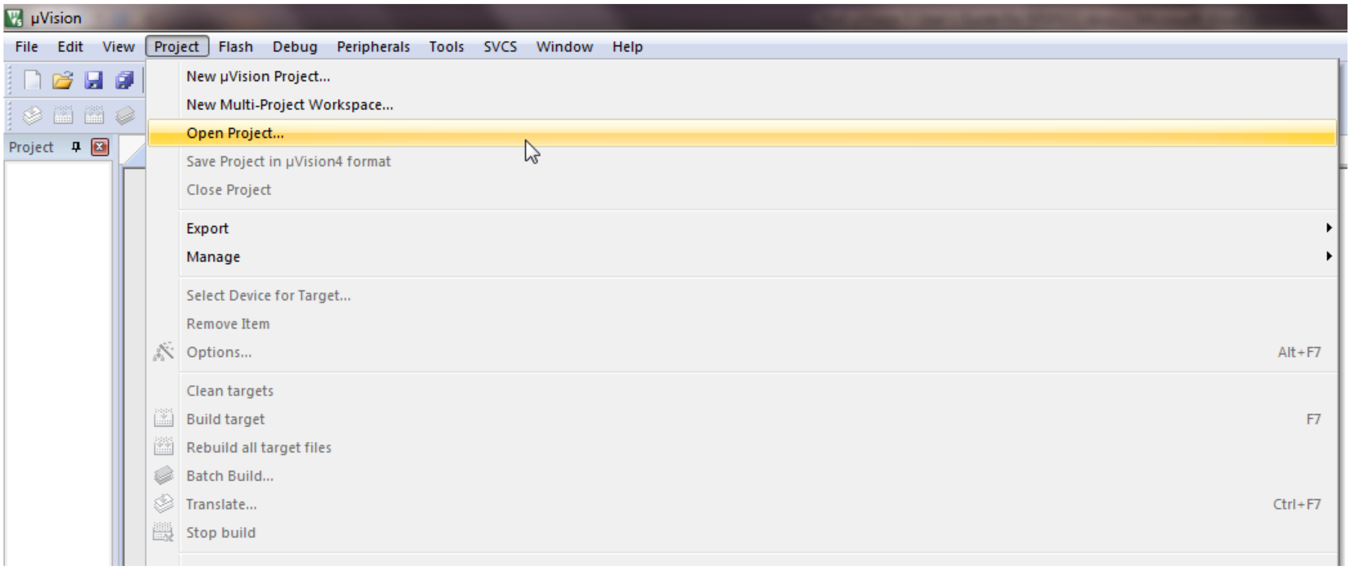
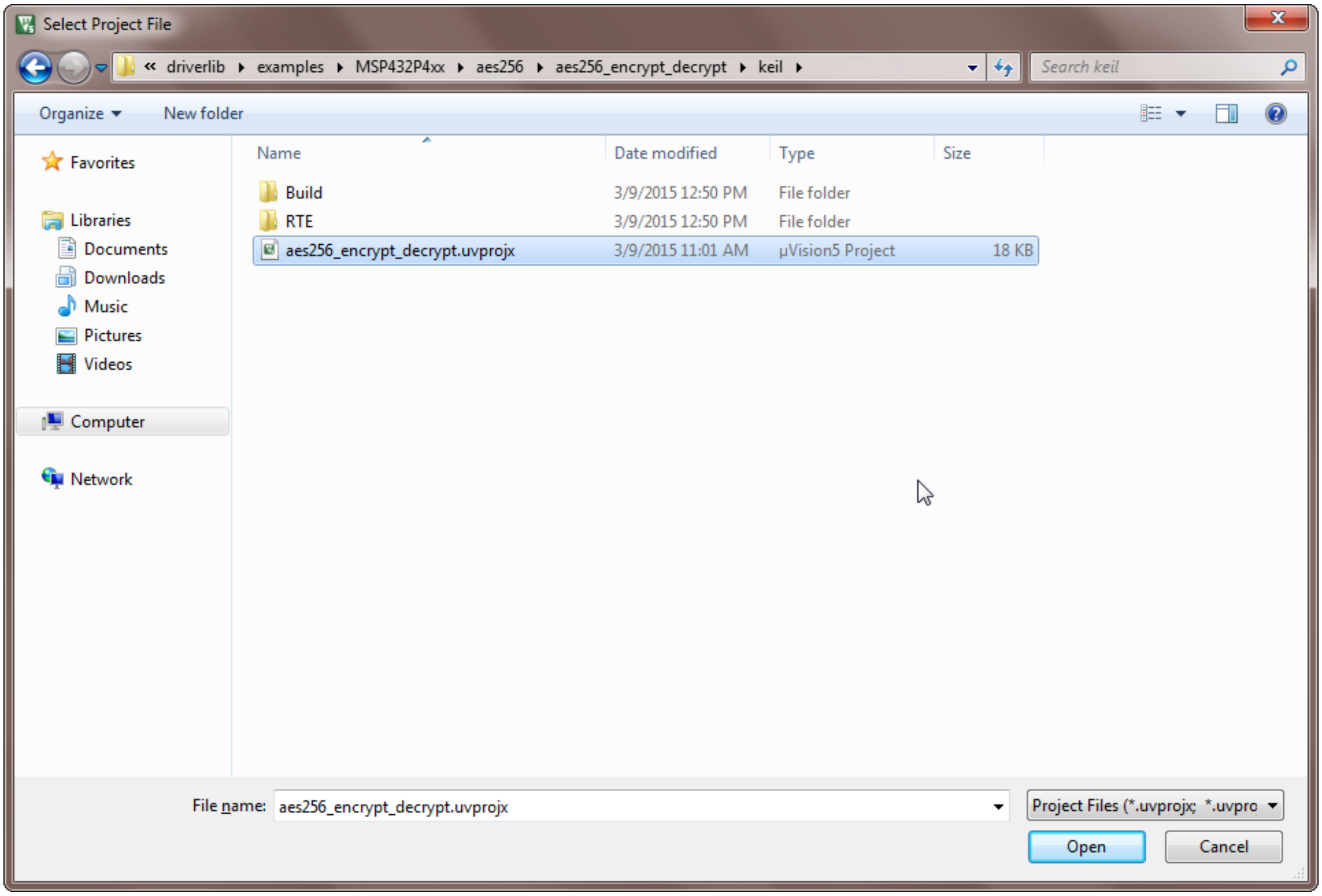


Figure 14. Opening a Project

After clicking "Open Project…", navigate to where the project is stored and double click the .uvprojx file to open the project in the IDE (see Figure 15).



Name	Date modified	Type	Size
Build	3/9/2015 12:50 PM	File folder	
RTE	3/9/2015 12:50 PM	File folder	
aes256_encrypt_decrypt.uvprojx	3/9/2015 11:01 AM	µVision5 Project	18 KB

File name: aes256_encrypt_decrypt.uvprojx

Project Files (*.uvprojx; *.uvpro

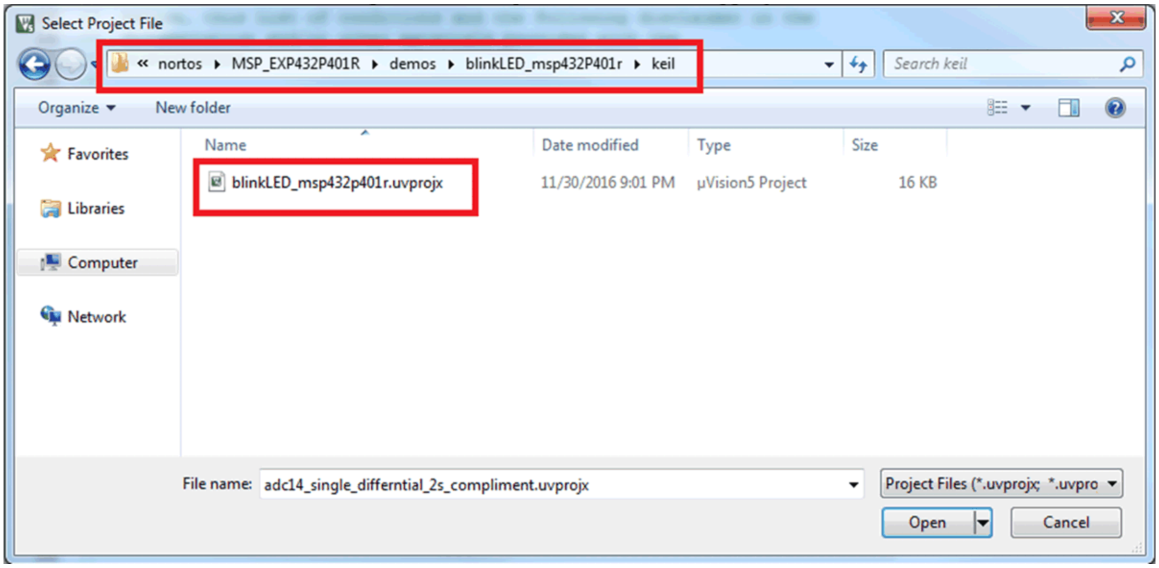
Figure 15. Choosing a Project to Import

5 Using SimpleLink MSP432 MCU Examples From the Software Development Kit

The SimpleLink MSP432 software development kit (SDK) contains many software examples, projects, documentation, application notes, and training for all MSP432 devices. This includes example projects for Keil µVision that work with MSP432 MCUs. For more information visit the SimpleLink MSP432 SDK page.

All documentation for the SDK can be found in the SDK installation path in the docs folder. Open Documention_Overview.html from that folder and then navigate to the Quick Start Guide for the IDE.

Each example in the SDK includes a folder named "Keil" that contains the *.uvprojx file with the project description. To import the project into the Keil MDK, copy it to the working location and open the project file (see Figure 16).



« nortos ▸ MSP_EXP432P401R ▸ demos ▸ blinkLED_msp432P401r ▸ keil

Name	Date modified	Type	Size
blinkLED_msp432p401r.uvprojx	11/30/2016 9:01 PM	µVision5 Project	16 KB

File name: adc14_single_differntial_2s_compliment.uvprojx

Project Files (*.uvprojx; *.uvpro

Figure 16. Select Project From SDK

You might need to change the device settings and debugger to fit the environment.

6 Debugging the Application

The following debug probes have been tested successfully with MSP432 MCUs and µVision IDE.

- Segger J-Link
- Keil ULINK2 and ULINK Pro
- XDS110-ET (using CMSIS-DAP or native support with MDK 5.20 and higher)

To use a debug probe that is not listed here, check with the vendor of the debug probe or with Arm if you experience problems.

6.1 Using Segger J-Link Debug Probe

To use the Segger J-Link debug probe, right-click on the active project, then select **Options for Target** (see Figure 17).

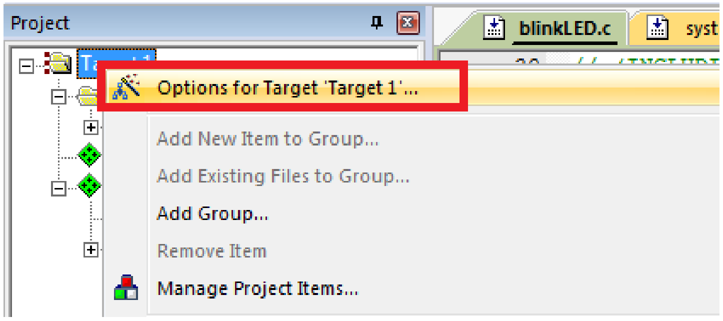
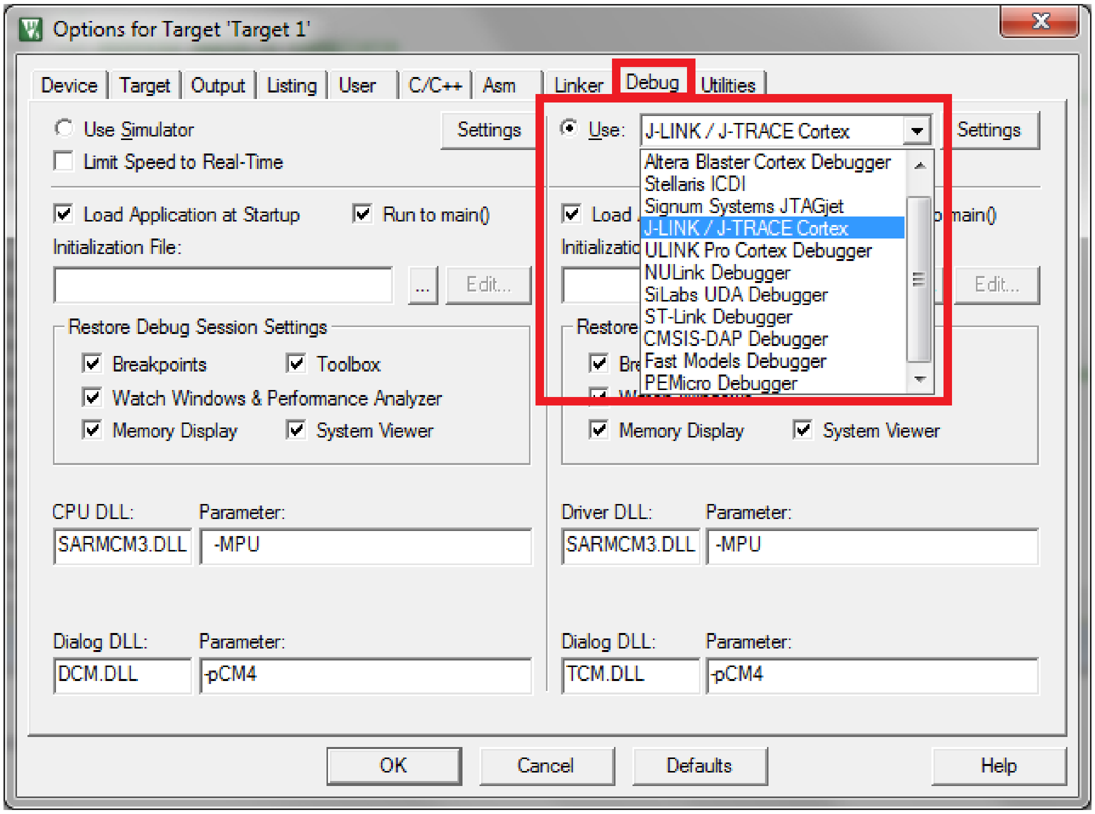


Figure 17. Project Options

In the **Target Options** window, select the **Debug** pane (see Figure 18). From the pulldown menu on the right-hand side, select the J-Link debug probe.



○ Use Simulator
□ Limit Speed to Real-Time
☑ Load Application at Startup ☑ Run to main()
Initialization File:
Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows & Performance Analyzer
☑ Memory Display ☑ System Viewer
CPU DLL: SARMCM3.DLL Parameter: -MPU
Dialog DLL: DCM.DLL Parameter: -pCM4

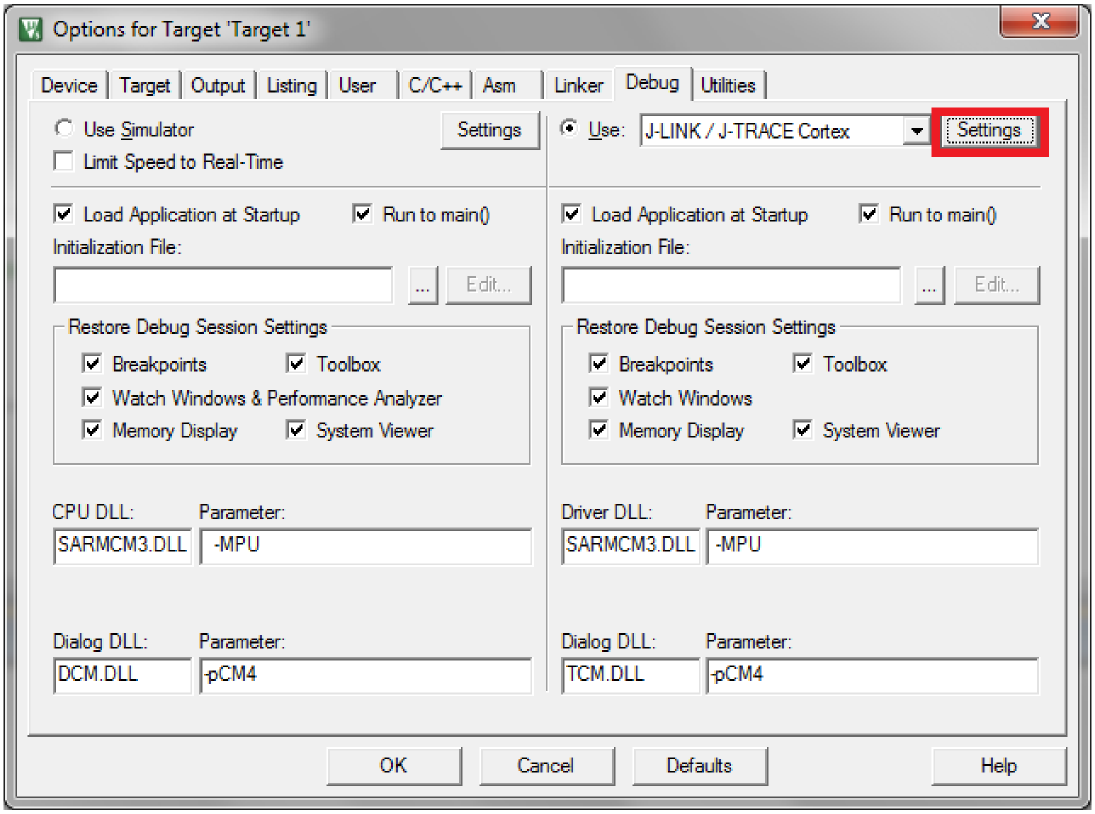
● Use: J-LINK / J-TRACE Cortex

- Altera Blaster Cortex Debugger
- Stellaris ICDI
- Signum Systems JTAGjet
- J-LINK / J-TRACE Cortex
- ULINK Pro Cortex Debugger
- NULink Debugger
- SiLabs UDA Debugger
- ST-Link Debugger
- CMSIS-DAP Debugger
- Fast Models Debugger
- PEMicro Debugger

☑ Load ... main()
Initializatio
Restore
☑ Bre
☑ Memory Display ☑ System Viewer
Driver DLL: SARMCM3.DLL Parameter: -MPU
Dialog DLL: TCM.DLL Parameter: -pCM4

Figure 18. Select J-Link Debug Probe

Next, go to the debug probe specific settings by clicking on the **Settings** button next to the debug probe (see Figure 19).



○ Use Simulator

□ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows & Performance Analyzer

☑ Memory Display ☑ System Viewer

CPU DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: DCM.DLL Parameter: -pCM4

● Use: J-LINK / J-TRACE Cortex

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows

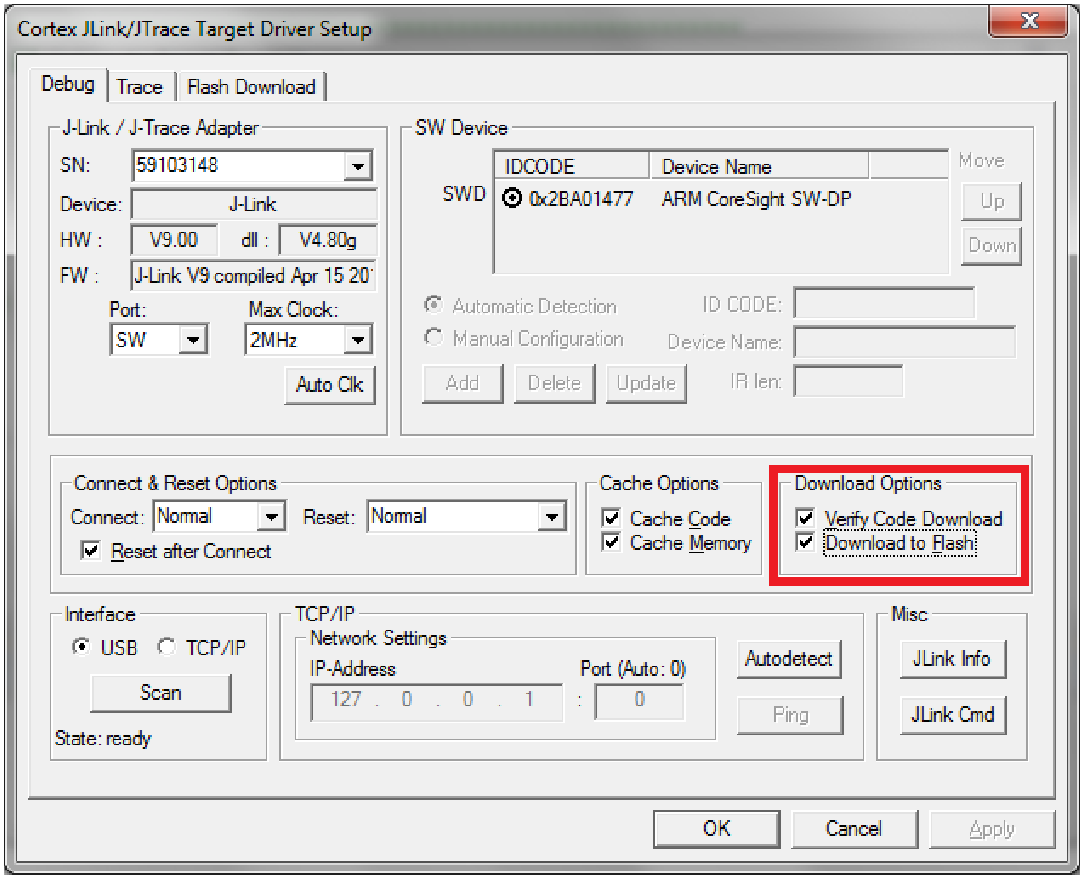
☑ Memory Display ☑ System Viewer

Driver DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: TCM.DLL Parameter: -pCM4

Figure 19. Go to J-Link Debug Probe Settings Window

In the Debug Probe Settings window (see Figure 20), make sure that both check boxes in the **Download Options** box are checked.



Cortex JLink/JTrace Target Driver Setup

Debug | Trace | Flash Download

J-Link / J-Trace Adapter

SN: 59103148

Device: J-Link

HW : V9.00 dll : V4.80g

FW : J-Link V9 compiled Apr 15 20

Port: SW Max Clock: 2MHz

SW Device

	IDCODE	Device Name
SWD	● 0x2BA01477	ARM CoreSight SW-DP

○ Automatic Detection ID CODE:

○ Manual Configuration Device Name:

IR len:

Connect & Reset Options

Connect: Normal Reset: Normal

☑ Reset after Connect

Cache Options

☑ Cache Code

☑ Cache Memory

Download Options

☑ Verify Code Download

☑ Download to Flash

Interface

● USB ○ TCP/IP

State: ready

TCP/IP

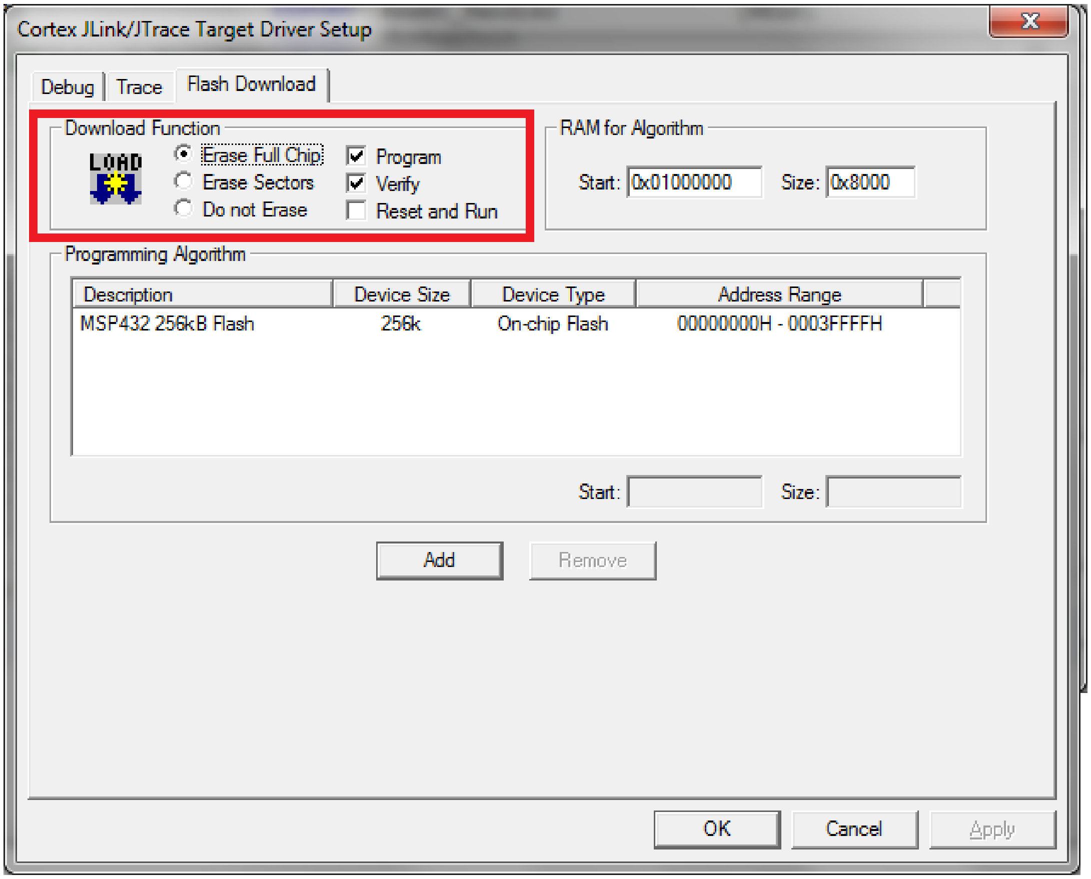
Network Settings

IP-Address 127 . 0 . 0 . 1 Port (Auto: 0) : 0

Misc

Figure 20. Enable J-Link Download Options

Go to the **Flash Download** pane to configure how the Flash memory should be erased, programmed, and/or verified. The settings shown in Figure 21 issue a Flash mass erase prior to Flash programming and verify the programmed image before enabling debug. Even though the "Erase Full Chip" is selected, the bootstrap loader (BSL) is not erased. To replace the BSL, choose "Erase Sectors" and place code in the BSL section of memory.



Download Function

- ● Erase Full Chip
- ○ Erase Sectors
- ○ Do not Erase
- ☑ Program
- ☑ Verify
- □ Reset and Run

RAM for Algorithm

Start: 0x01000000 Size: 0x8000

Programming Algorithm

Description	Device Size	Device Type	Address Range
MSP432 256kB Flash	256k	On-chip Flash	00000000H - 0003FFFFH

Start: Size:

Figure 21. J-Link Download Function Configuration

After the Flash download has been configured, the Debug Session can be started. Go to **Debug** → **Start/Stop Debug Session**, or alternatively press **Ctrl+F5**.

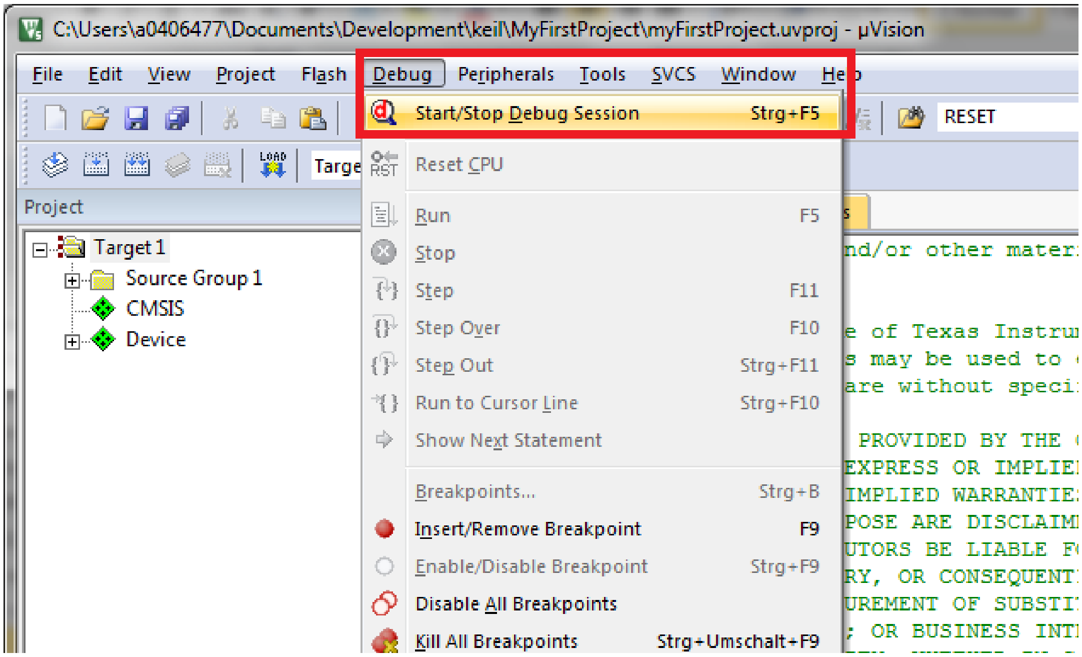
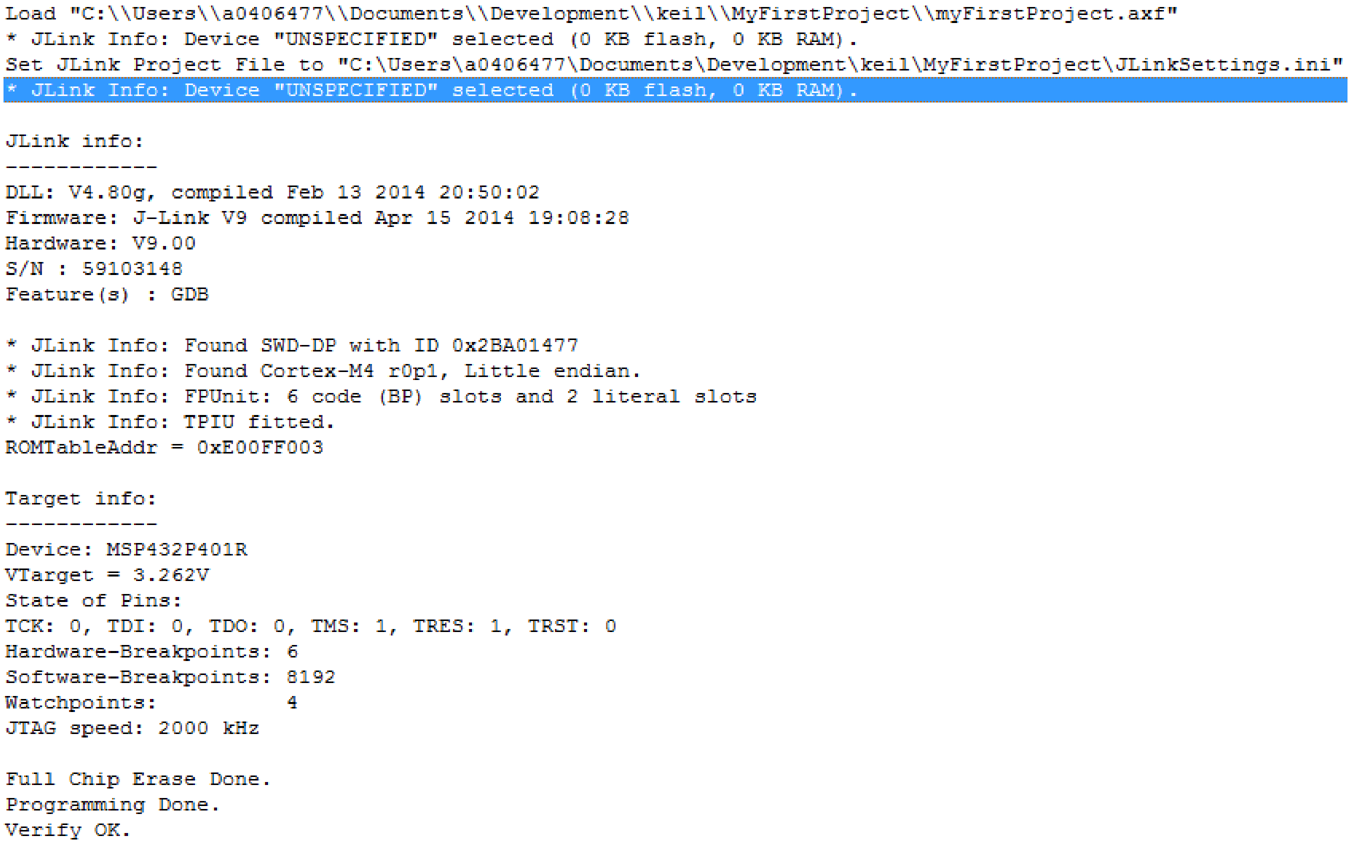


Figure 22. Start Debug Session

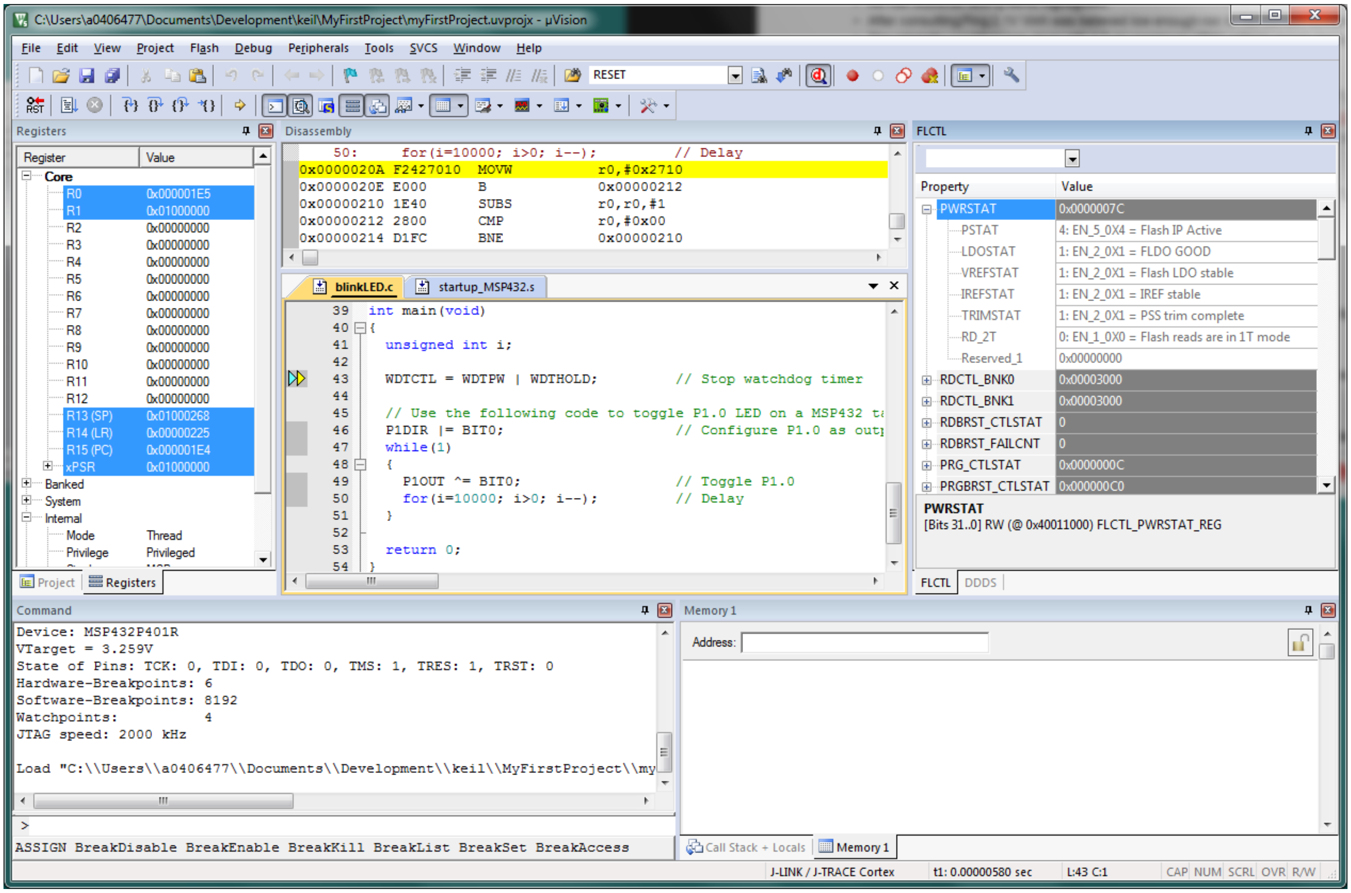
If using an older version of the Segger J-Link software, a popup might show once and indicate that the MSP432 device is not yet in the Segger J-Link database. It is safe to answer the manual device selection question with "No", but consider checking with Segger for an updated version of the J-Link software. After that, the debug session is started. Figure 23 shows the Command window output during the start of the debug session.



```
Load "C:\\Users\\a0406477\\Documents\\Development\\keil\\MyFirstProject\\myFirstProject.axf"
* JLink Info: Device "UNSPECIFIED" selected (0 KB flash, 0 KB RAM).
Set JLink Project File to "C:\Users\a0406477\Documents\Development\keil\MyFirstProject\JLinkSettings.ini"
* JLink Info: Device "UNSPECIFIED" selected (0 KB flash, 0 KB RAM).

JLink info:
------------
DLL: V4.80g, compiled Feb 13 2014 20:50:02
Firmware: J-Link V9 compiled Apr 15 2014 19:08:28
Hardware: V9.00
S/N : 59103148
Feature(s) : GDB

* JLink Info: Found SWD-DP with ID 0x2BA01477
* JLink Info: Found Cortex-M4 r0p1, Little endian.
* JLink Info: FPUnit: 6 code (BP) slots and 2 literal slots
* JLink Info: TPIU fitted.
ROMTableAddr = 0xE00FF003

Target info:
------------
Device: MSP432P401R
VTarget = 3.262V
State of Pins:
TCK: 0, TDI: 0, TDO: 0, TMS: 1, TRES: 1, TRST: 0
Hardware-Breakpoints: 6
Software-Breakpoints: 8192
Watchpoints:          4
JTAG speed: 2000 kHz

Full Chip Erase Done.
Programming Done.
Verify OK.
```

Figure 23. Messages During Debug Session Startup

After the program image has been written to flash memory, the debug session starts and the debugger halts at the main function (see Figure 24).



Register	Value
Core	
R0	0x000001E5
R1	0x01000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x01000268
R14 (LR)	0x00000225
R15 (PC)	0x000001E4
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged

```
    50:      for(i=10000; i>0; i--);          // Delay
0x0000020A F2427010  MOVW          r0,#0x2710
0x0000020E E000      B             0x00000212
0x00000210 1E40      SUBS          r0,r0,#1
0x00000212 2800      CMP           r0,#0x00
0x00000214 D1FC      BNE           0x00000210
```

```
int main(void)
{
  unsigned int i;

  WDTCTL = WDTPW | WDTHOLD;         // Stop watchdog timer

  // Use the following code to toggle P1.0 LED on a MSP432 ta
  P1DIR |= BIT0;                    // Configure P1.0 as outp
  while(1)
  {
    P1OUT ^= BIT0;                  // Toggle P1.0
    for(i=10000; i>0; i--);         // Delay
  }

  return 0;
}
```

Property	Value
PWRSTAT	0x0000007C
PSTAT	4: EN_5_0X4 = Flash IP Active
LDOSTAT	1: EN_2_0X1 = FLDO GOOD
VREFSTAT	1: EN_2_0X1 = Flash LDO stable
IREFSTAT	1: EN_2_0X1 = IREF stable
TRIMSTAT	1: EN_2_0X1 = PSS trim complete
RD_2T	0: EN_1_0X0 = Flash reads are in 1T mode
Reserved_1	0x00000000
RDCTL_BNK0	0x00003000
RDCTL_BNK1	0x00003000
RDBRST_CTLSTAT	0
RDBRST_FAILCNT	0
PRG_CTLSTAT	0x0000000C
PRGBRST_CTLSTAT	0x000000C0

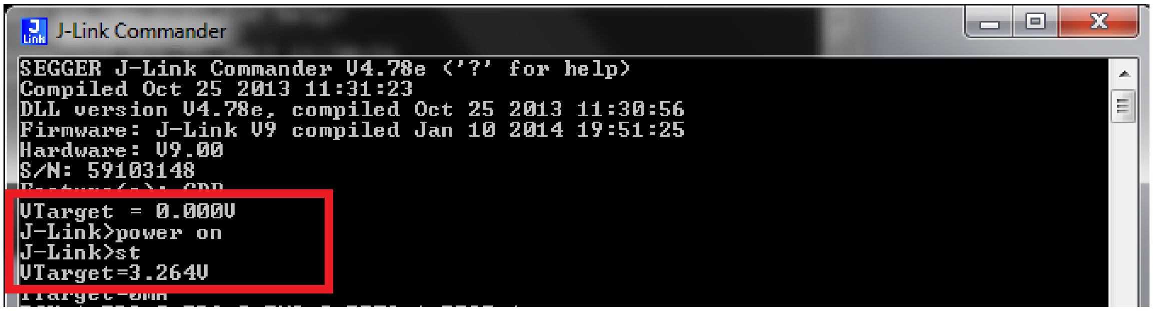
PWRSTAT
[Bits 31..0] RW (@ 0x40011000) FLCTL_PWRSTAT_REG

```
Device: MSP432P401R
VTarget = 3.259V
State of Pins: TCK: 0, TDI: 0, TDO: 0, TMS: 1, TRES: 1, TRST: 0
Hardware-Breakpoints: 6
Software-Breakpoints: 8192
Watchpoints:          4
JTAG speed: 2000 kHz

Load "C:\\Users\\a0406477\\Documents\\Development\\keil\\MyFirstProject\\my
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Figure 24. Debug Session Halted at main()

When using an MSP432 target socket board, you can benefit from the 5-V voltage output the J-Link provides on pin 19 of its Cortex-M debug connector to power the MSP432 device. This option needs to be enabled through the J-Link Commander, a console application available from Segger, since it is not available from the IDE. After it is enabled, the debug probe provides 5 V to the target system. See the *MSP432™ SimpleLink™ Microcontrollers Hardware Tools User's Guide* for instructions on how to configure the target socket board to use the 5-V power supply to generate 3.3-V device voltage, and see Segger's documentation for instructions on how to enable the voltage output.

Figure 25 shows the effect of the **power on** command, when applied in the J-Link Commander. Before executing the command, the measured target voltage is 0 V, and right after applying target power, 3.3 V is available as target voltage.



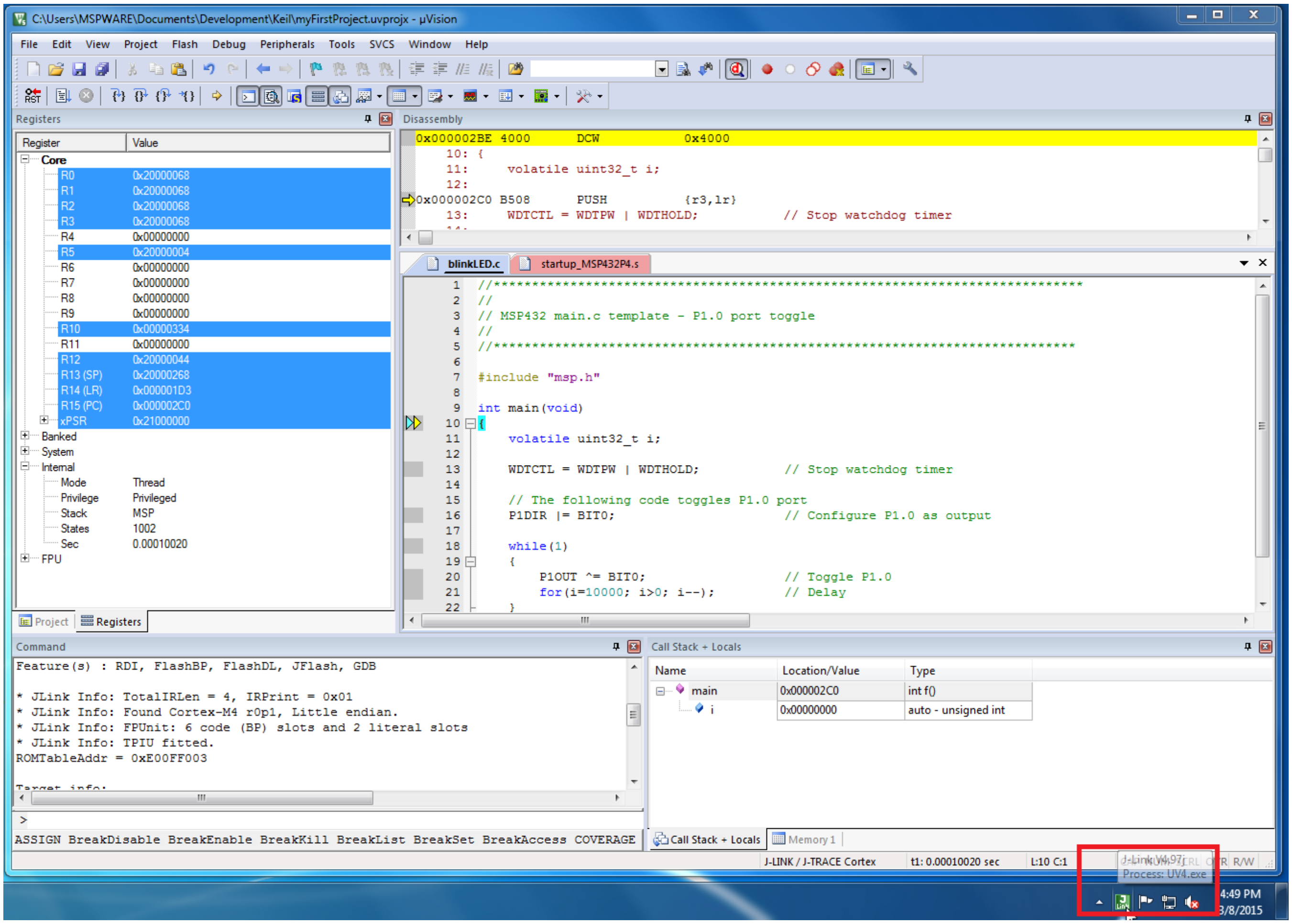
```
SEGGER J-Link Commander V4.78e ('?' for help)
Compiled Oct 25 2013 11:31:23
DLL version V4.78e, compiled Oct 25 2013 11:30:56
Firmware: J-Link V9 compiled Jan 10 2014 19:51:25
Hardware: V9.00
S/N: 59103148
Feature(s): GDB
VTarget = 0.000V
J-Link>power on
J-Link>st
VTarget=3.264V
ITarget=0mA
```

Figure 25. Using J-Link Commander to Enable Power Output to Target System

Now, download the program normally and debug using the Segger J-Link debug probe with µVision.

6.1.1 Working With Device Security (MSP432P4xx Devices Only)

If JTAG access is disabled on the device or the application needs to unlock a secure IP zone, a J-Link Script must be added to the debug session to enable a factory reset. During a debug session launch the J-Link control panel by clicking on the J-Link icon in the status bar (see Figure 26).


C:\Users\MSPWARE\Documents\Development\Keil\myFirstProject.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
RST
Registers
Disassembly
blinkLED.c
startup_MSP432P4.s
Project
Registers
Command
Call Stack + Locals
Memory 1
J-Link V4.97j
Process: UV4.exe
4:49 PM
3/8/2015

Register	Value
Core	
R0	0x20000068
R1	0x20000068
R2	0x20000068
R3	0x20000068
R4	0x00000000
R5	0x20000004
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000334
R11	0x00000000
R12	0x20000044
R13 (SP)	0x20000268
R14 (LR)	0x000001D3
R15 (PC)	0x000002C0
xPSR	0x21000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	1002
Sec	0.00010020
FPU	

```
0x000002BE 4000      DCW          0x4000
    10: {
    11:     volatile uint32_t i;
    12:
0x000002C0 B508      PUSH         {r3,lr}
    13:     WDTCTL = WDTPW | WDTHOLD;          // Stop watchdog timer
```

```
//*****************************************************************************
//
// MSP432 main.c template - P1.0 port toggle
//
//****************************************************************************

#include "msp.h"

int main(void)
{
    volatile uint32_t i;

    WDTCTL = WDTPW | WDTHOLD;           // Stop watchdog timer

    // The following code toggles P1.0 port
    P1DIR |= BIT0;                      // Configure P1.0 as output

    while(1)
    {
        P1OUT ^= BIT0;                  // Toggle P1.0
        for(i=10000; i>0; i--);         // Delay
    }
```

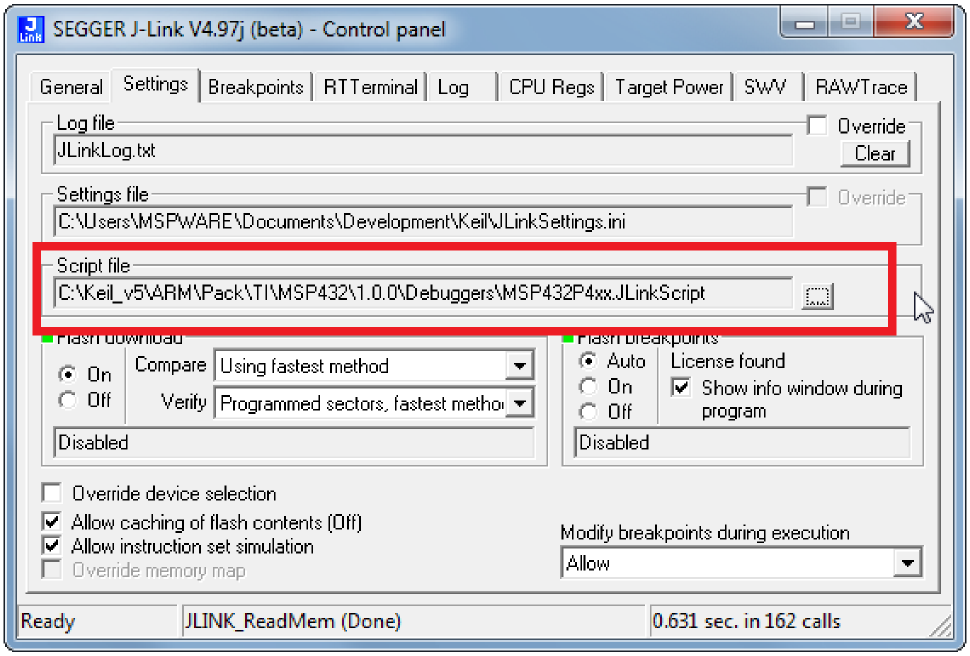
```
Feature(s) : RDI, FlashBP, FlashDL, JFlash, GDB

* JLink Info: TotalIRLen = 4, IRPrint = 0x01
* JLink Info: Found Cortex-M4 r0p1, Little endian.
* JLink Info: FPUnit: 6 code (BP) slots and 2 literal slots
* JLink Info: TPIU fitted.
ROMTableAddr = 0xE00FF003

Target info:
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
```

Name	Location/Value	Type
main	0x000002C0	int f()
i	0x00000000	auto - unsigned int

J-LINK / J-TRACE Cortex   t1: 0.00010020 sec   L:10 C:1   R/W

Figure 26. Launch J-Link Control Panel

After launching the J-Link control panel, add the J-Link script for MSP432 MCUs provided in the Keil pack. Figure 27 shows the location of the script file.



SEGGER J-Link V4.97j (beta) - Control panel

Log file: JLinkLog.txt □ Override

Settings file: C:\Users\MSPWARE\Documents\Development\Keil\JLinkSettings.ini □ Override

Script file: C:\Keil_v5\ARM\Pack\TI\MSP432\1.0.0\Debuggers\MSP432P4xx.JLinkScript

Flash download: ● On ○ Off
Compare: Using fastest method
Verify: Programmed sectors, fastest metho
Disabled

Flash breakpoints: ● Auto ○ On ○ Off
License found
☑ Show info window during program
Disabled

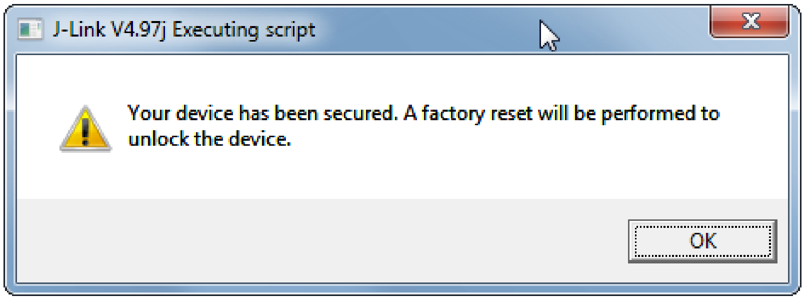
□ Override device selection
☑ Allow caching of flash contents (Off)
☑ Allow instruction set simulation
□ Override memory map

Modify breakpoints during execution: Allow

Ready | JLINK_ReadMem (Done) | 0.631 sec. in 162 calls

Figure 27. J-Link Control Panel

The J-Link script now runs at the launch of very debug session and every time code is downloaded to the device. If the device has been secured when trying to download code, a dialog box reports that the device is secured and will be erased (see Figure 28).

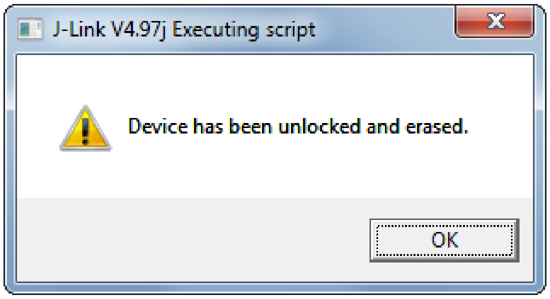


J-Link V4.97j Executing script

Your device has been secured. A factory reset will be performed to unlock the device.

Figure 28. J-Link Script Detecting That the Device Has Been Secured

Click OK on the dialog box to issue a factory reset, which erases any code present on the device, and to start to download the compiled code. After the factory reset is a complete, a confirmation dialog box is displayed (see Figure 29).



J-Link V4.97j Executing script

Device has been unlocked and erased.

Figure 29. Device Has Been Unlocked and Erased

6.2 Using Keil ULINK2 and ULINK Pro debug probe

To use Keil ULINK2 or ULINK Pro debug probes, right-click on the active project, then select **Options for Target**.

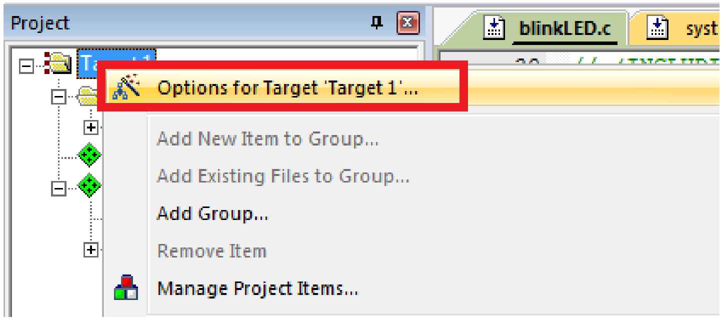
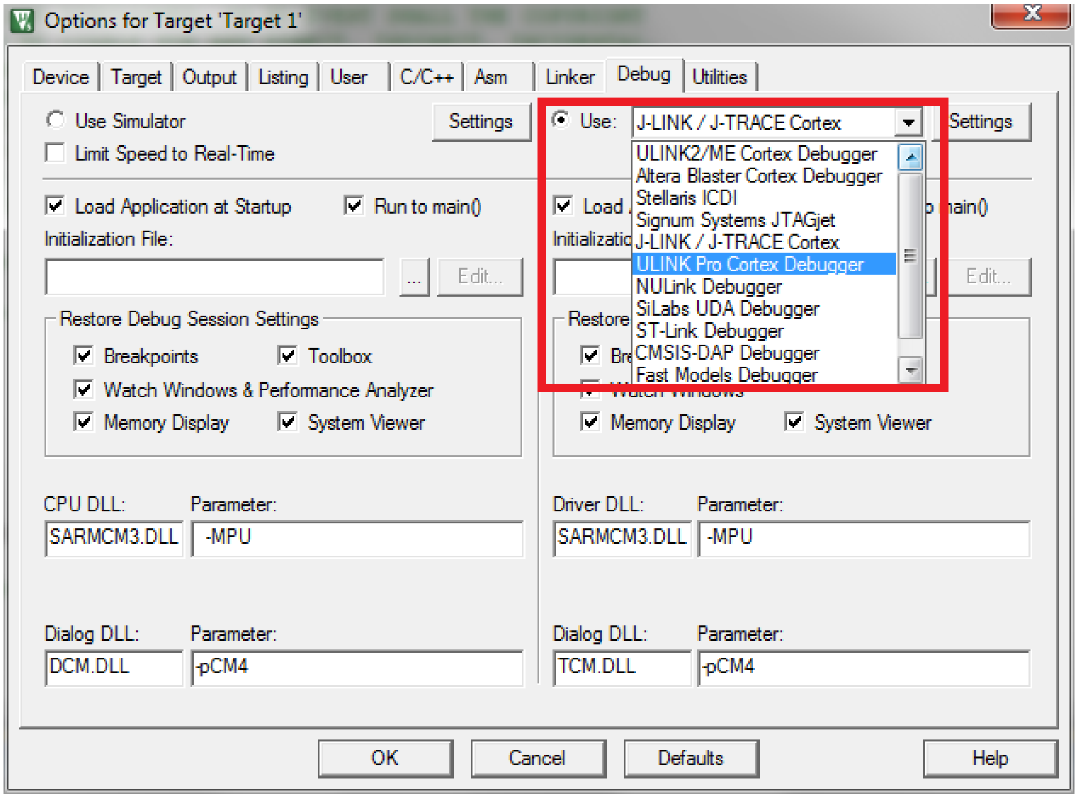


Figure 30. Project Options

In the Target Options window, select the **Debug** pane (see Figure 31). From the pulldown menu, select the ULINK2 or ULINK Pro debug probe. Be careful to select the right debug probe, as the driver is different.

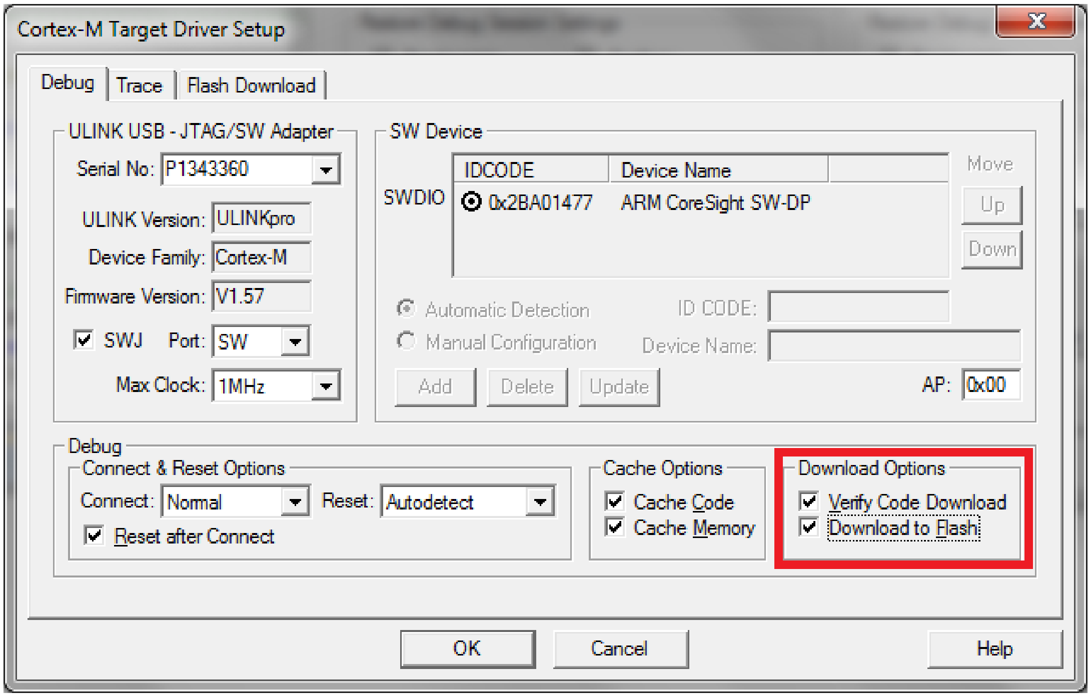


○ Use Simulator
□ Limit Speed to Real-Time
☑ Load Application at Startup ☑ Run to main()
Initialization File:
Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows & Performance Analyzer
☑ Memory Display ☑ System Viewer
CPU DLL: SARMCM3.DLL Parameter: -MPU
Dialog DLL: DCM.DLL Parameter: -pCM4

● Use: J-LINK / J-TRACE Cortex
☑ Load ... ain()
Initializatio...
Restore ...
☑ Bre...
☑ Memory Display ☑ System Viewer
Driver DLL: SARMCM3.DLL Parameter: -MPU
Dialog DLL: TCM.DLL Parameter: -pCM4

Figure 31. Selecting the ULINK Pro Debug Probe

In the Debug Probe Settings window (see Figure 32), make sure that both check boxes in the **Download Options** box are checked.



Cortex-M Target Driver Setup

ULINK USB - JTAG/SW Adapter

Serial No: P1343360

ULINK Version: ULINKpro

Device Family: Cortex-M

Firmware Version: V1.57

☑ SWJ Port: SW

Max Clock: 1MHz

SW Device

	IDCODE	Device Name
SWDIO	● 0x2BA01477	ARM CoreSight SW-DP

● Automatic Detection ID CODE:

○ Manual Configuration Device Name:

AP: 0x00

Debug

Connect & Reset Options

Connect: Normal Reset: Autodetect

☑ Reset after Connect

Cache Options

☑ Cache Code

☑ Cache Memory

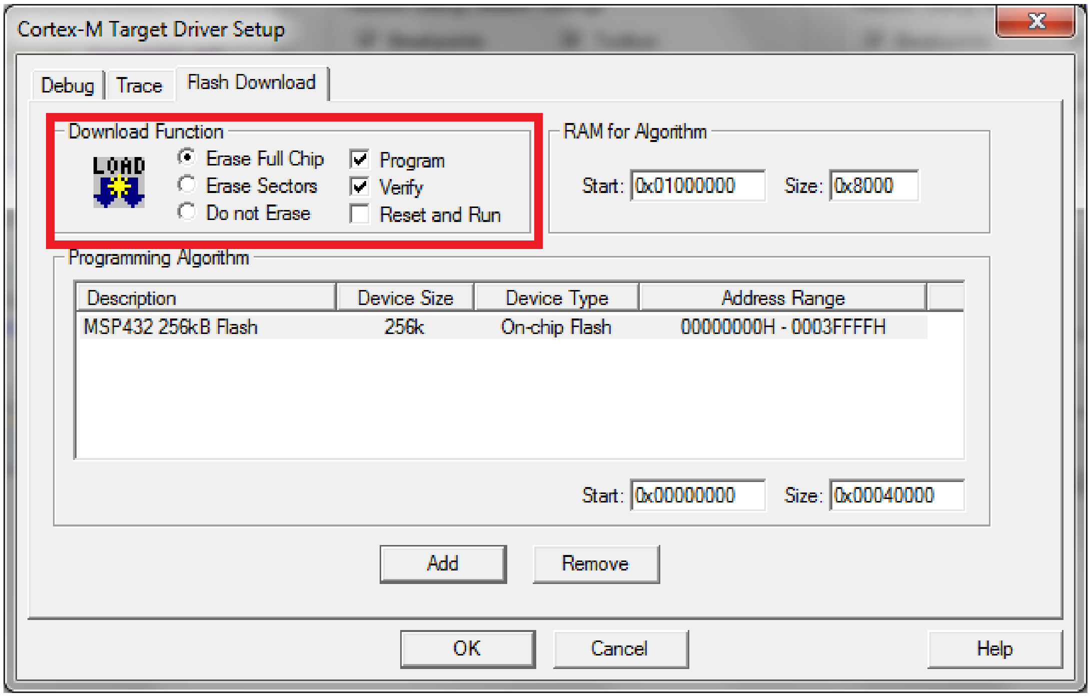
Download Options

☑ Verify Code Download

☑ Download to Flash

Figure 32. Selecting ULINK Pro Download Options

Now move to **Flash Download** pane to configure the way the Flash memory should be erased, programmed, and/or verified. Figure 33 shows the settings to issue a flash mass erase prior to flash programming and verify the programmed image before enabling debug. Even though the "Erase Full Chip" is selected, the BSL is not erased. To replace the BSL, choose "Erase Sectors" and place code in the BSL section of memory.



Cortex-M Target Driver Setup

Download Function

● Erase Full Chip ☑ Program

○ Erase Sectors ☑ Verify

○ Do not Erase ☐ Reset and Run

RAM for Algorithm

Start: 0x01000000 Size: 0x8000

Programming Algorithm

Description	Device Size	Device Type	Address Range
MSP432 256kB Flash	256k	On-chip Flash	00000000H - 0003FFFFH

Start: 0x00000000 Size: 0x00040000

Figure 33. Select ULINK Pro Download Function

After the flash download has been configured, the debug session can be started. Go to **Debug** → **Start/Stop Debug Session**, or alternatively press **Ctrl+F5**.

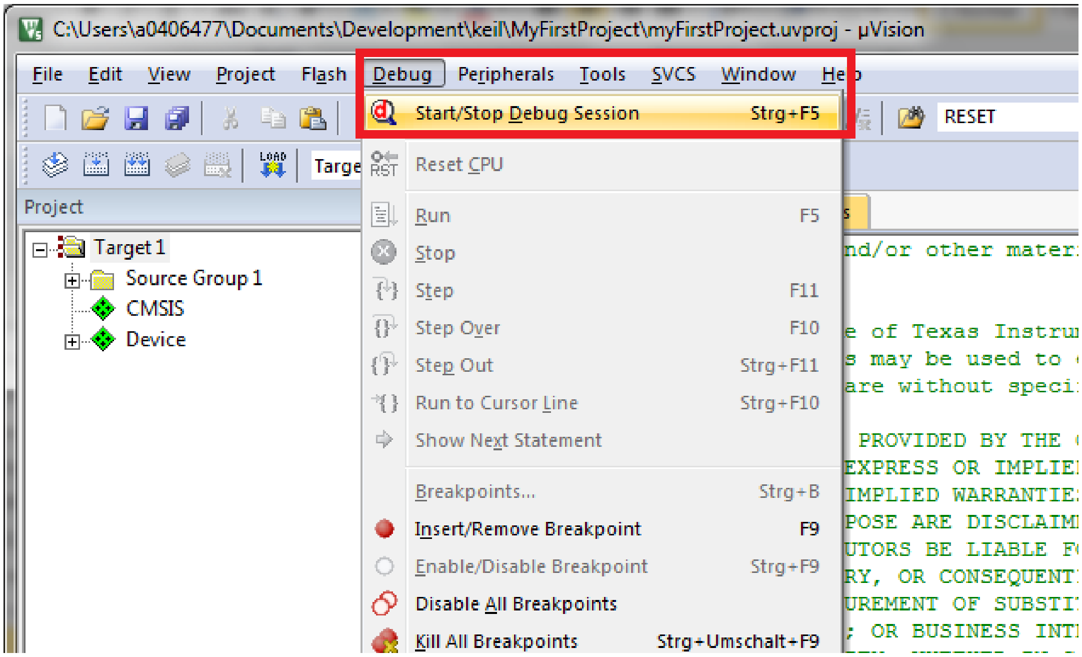
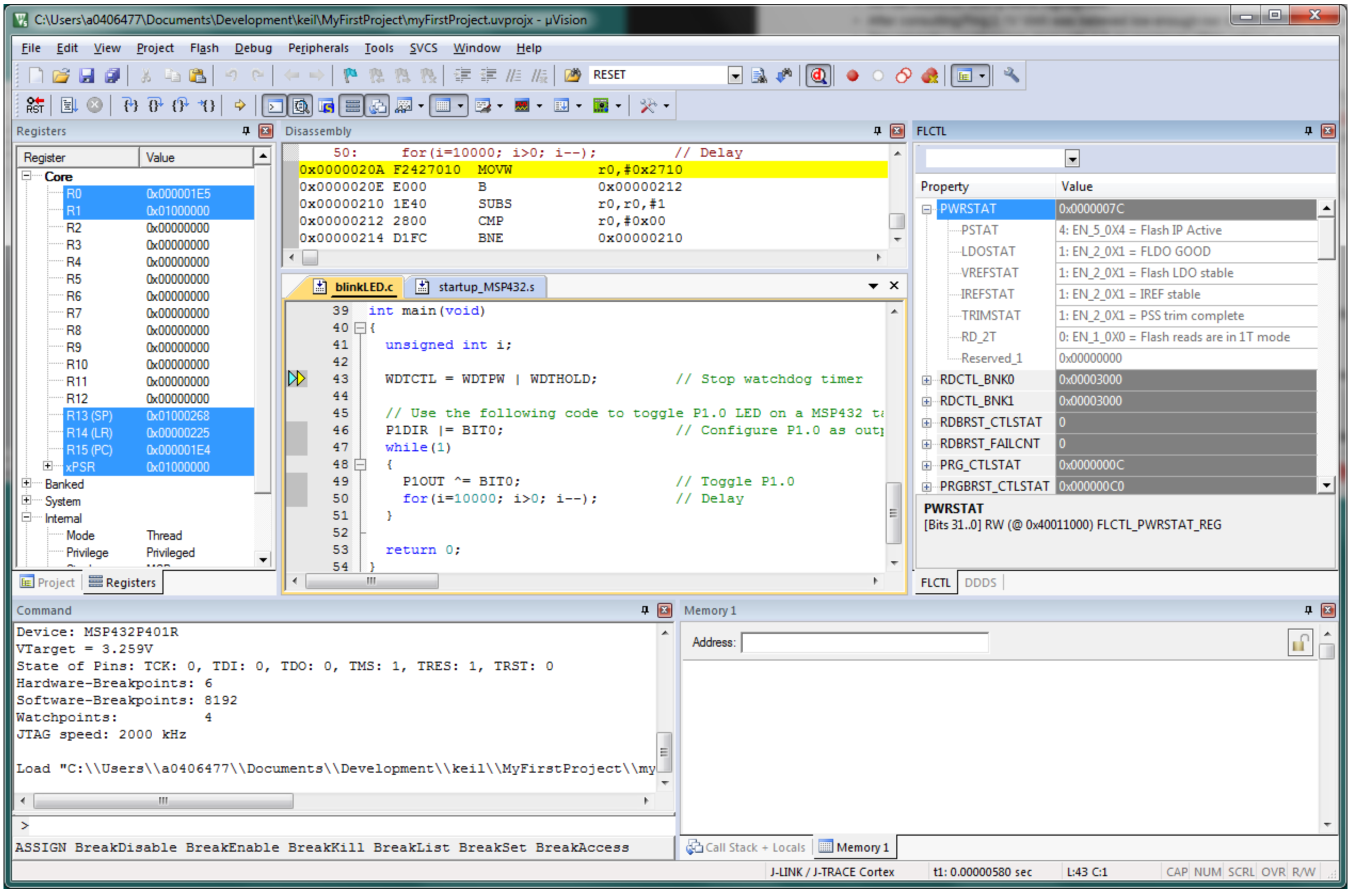


Figure 34. Start Debug Session

After the program image has been written to flash memory, the debug session starts and the debugger halts at the main function (see Figure 35).



Register	Value
Core	
R0	0x000001E5
R1	0x01000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x01000268
R14 (LR)	0x00000225
R15 (PC)	0x000001E4
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged

```
    50:      for(i=10000; i>0; i--);          // Delay
0x0000020A F2427010  MOVW           r0,#0x2710
0x0000020E E000      B              0x00000212
0x00000210 1E40      SUBS           r0,r0,#1
0x00000212 2800      CMP            r0,#0x00
0x00000214 D1FC      BNE            0x00000210
```

```
int main(void)
{
  unsigned int i;

  WDTCTL = WDTPW | WDTHOLD;          // Stop watchdog timer

  // Use the following code to toggle P1.0 LED on a MSP432 ta
  P1DIR |= BIT0;                     // Configure P1.0 as outp
  while(1)
  {
    P1OUT ^= BIT0;                   // Toggle P1.0
    for(i=10000; i>0; i--);          // Delay
  }

  return 0;
}
```

Property	Value
PWRSTAT	0x0000007C
PSTAT	4: EN_5_0X4 = Flash IP Active
LDOSTAT	1: EN_2_0X1 = FLDO GOOD
VREFSTAT	1: EN_2_0X1 = Flash LDO stable
IREFSTAT	1: EN_2_0X1 = IREF stable
TRIMSTAT	1: EN_2_0X1 = PSS trim complete
RD_2T	0: EN_1_0X0 = Flash reads are in 1T mode
Reserved_1	0x00000000
RDCTL_BNK0	0x00003000
RDCTL_BNK1	0x00003000
RDBRST_CTLSTAT	0
RDBRST_FAILCNT	0
PRG_CTLSTAT	0x0000000C
PRGBRST_CTLSTAT	0x000000C0

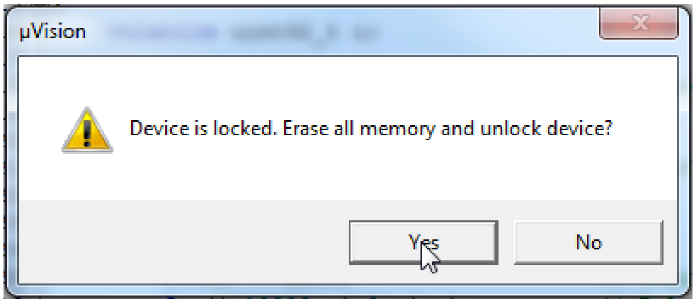
PWRSTAT
[Bits 31..0] RW (@ 0x40011000) FLCTL_PWRSTAT_REG

```
Device: MSP432P401R
VTarget = 3.259V
State of Pins: TCK: 0, TDI: 0, TDO: 0, TMS: 1, TRES: 1, TRST: 0
Hardware-Breakpoints: 6
Software-Breakpoints: 8192
Watchpoints:          4
JTAG speed: 2000 kHz

Load "C:\\Users\\a0406477\\Documents\\Development\\keil\\MyFirstProject\\my
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Figure 35. Debug Session Successfully Started and Halted at main()

6.2.1 Working With Device Security (MSP432P4xx Devices Only)

If JTAG access is disabled on the device or the application needs to unlock a secure IP zone, Keil automatically runs a check on the device before downloading code. If Keil finds that the device has been secured, a dialog box is displayed (see Figure 36).



Device is locked. Erase all memory and unlock device?

Figure 36. Dialog Box Asking to Perform a Factory Reset

Click **Yes** to perform a factory reset, which unlocks the device so that code can be downloaded. After the code has been downloaded, the debug session starts.

Click No to end the debug session and leave the device locked.

6.3 Using XDS110-ET Debug Probe

The XDS110-ET is the debugger featured on the MSP-EXP432P401R LaunchPad™ development kit and allows developers to quickly get started developing and debugging applications on MSP432 MCUs. In addition, the XDS110 is available as a stand-alone probe (see XDS110 JTAG Debug Probe).

To use XDS110-ET debug probe, right-click on the active project, then select **Options for Target**.

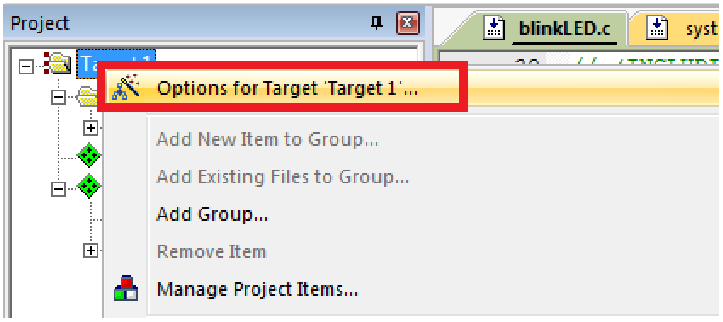


Figure 37. Project Options

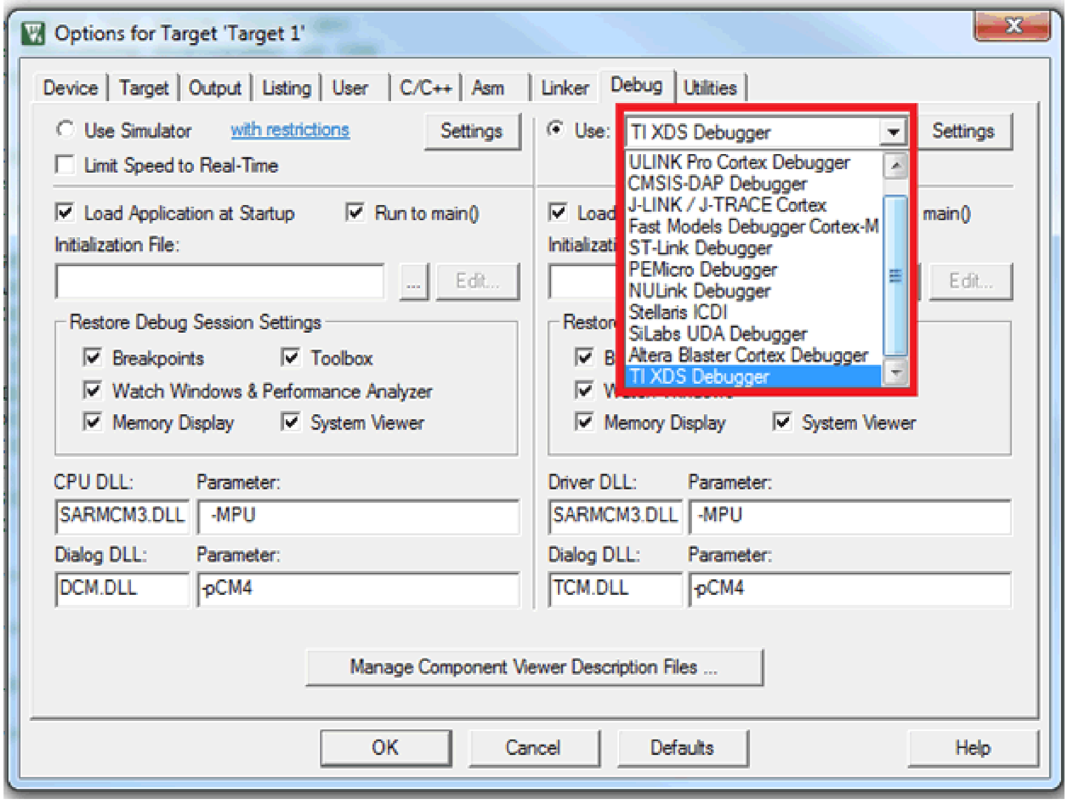
In the Target Options window, select the **Debug** pane. From the pulldown menu, select either CMSIS-DAP Debugger or TI XDS Debugger. TI XDS gives higher performance and is available in MDK 5.20 or higher.

NOTE: TI XDS110 debuggers are supported natively or through CMSIS DAP. TI XDS native support in MDK 5.20 needs a separate installation of the TI emulation pack software in the default folder C:\TI\ccs-base.

MDK 5.20 needs version 6.0.83.1 of the emulation pack and the associated firmware 2.2.5.1 on the XDS110 debug probe. This can be checked and changed with the xdsdfu tool provided in the emulation pack. See the readme in the installation folder (C:\TI\ccs_base\common\uscif\xds110) for details.

All versions of the emulation pack are available in the Texas Instruments XDS Emulation Software.

Higher versions of MDK update the XDS firmware accordingly.


Options for Target 'Target 1'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

○ Use Simulator with restrictions Settings

□ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows & Performance Analyzer

☑ Memory Display ☑ System Viewer

CPU DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: DCM.DLL Parameter: -pCM4

● Use: TI XDS Debugger Settings

ULINK Pro Cortex Debugger
CMSIS-DAP Debugger
J-LINK / J-TRACE Cortex
Fast Models Debugger Cortex-M
ST-Link Debugger
PEMicro Debugger
NULink Debugger
Stellaris ICDI
SiLabs UDA Debugger
Altera Blaster Cortex Debugger
TI XDS Debugger

☑ Load ... ☑ ... main()

Initializati...

Restor...

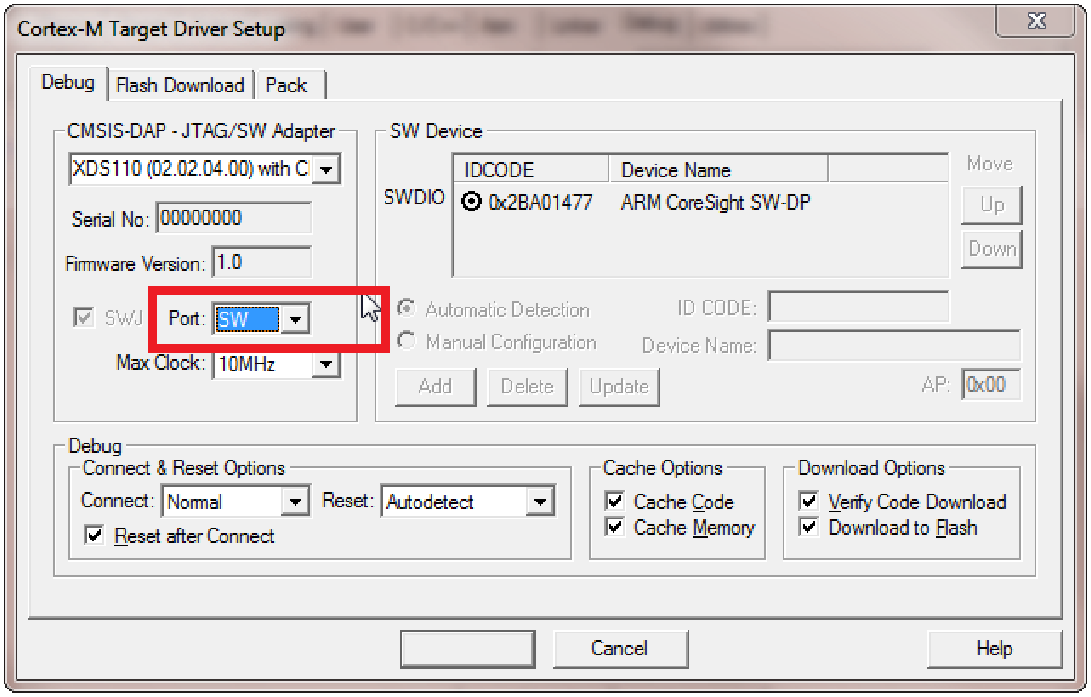
☑ Memory Display ☑ System Viewer

Driver DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: TCM.DLL Parameter: -pCM4

Figure 38. Selecting the XDS110 Debug Probe

In the Debug Probe Settings window (see Figure 39), make sure that both check boxes in the **Download Options** box are checked.


Cortex-M Target Driver Setup

Debug | Flash Download | Pack

CMSIS-DAP - JTAG/SW Adapter

XDS110 (02.02.04.00) with C

Serial No: 00000000

Firmware Version: 1.0

☑ SWJ Port: SW

Max Clock: 10MHz

SW Device

	IDCODE	Device Name
SWDIO	● 0x2BA01477	ARM CoreSight SW-DP

● Automatic Detection ID CODE:

○ Manual Configuration Device Name:

AP: 0x00

Debug

Connect & Reset Options

Connect: Normal Reset: Autodetect

☑ Reset after Connect

Cache Options

☑ Cache Code

☑ Cache Memory

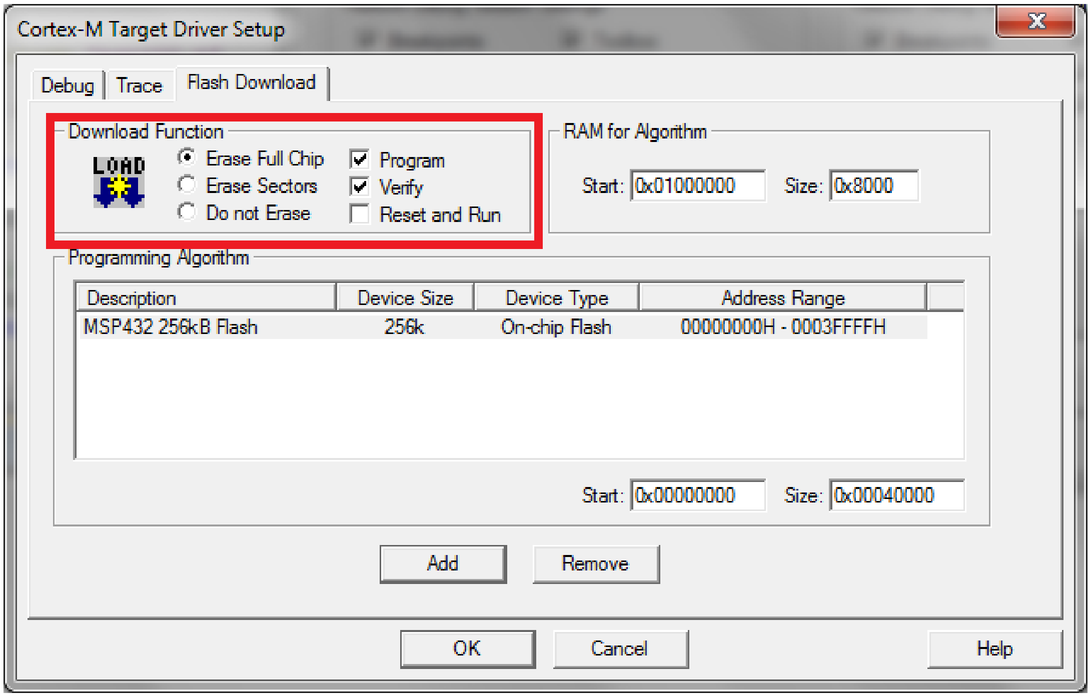
Download Options

☑ Verify Code Download

☑ Download to Flash

Figure 39. Selecting CMSIS-DAP Settings

Now move to **Flash Download** pane to configure the way the Flash memory should be erased, programmed and/or verified. The settings shown in Figure 40 issue a Flash mass erase prior to Flash programming and verify the programmed image before enabling debug. Even though the "Erase Full Chip" is selected the Bootstrap Loader (BSL) will not be erased. To replace the BSL choose "Erase Sectors" and place code in the BSL section of memory.



Download Function

- ● Erase Full Chip
- ○ Erase Sectors
- ○ Do not Erase
- ☑ Program
- ☑ Verify
- ☐ Reset and Run

RAM for Algorithm

Start: 0x01000000 Size: 0x8000

Programming Algorithm

Description	Device Size	Device Type	Address Range
MSP432 256kB Flash	256k	On-chip Flash	00000000H - 0003FFFFH

Start: 0x00000000 Size: 0x00040000

Figure 40. Select ULINK Pro Download Function

After the Flash download has been configured, the Debug Session can be started. Go to **Debug** → **Start/Stop Debug Session**, or alternatively press **Ctrl+F5**.

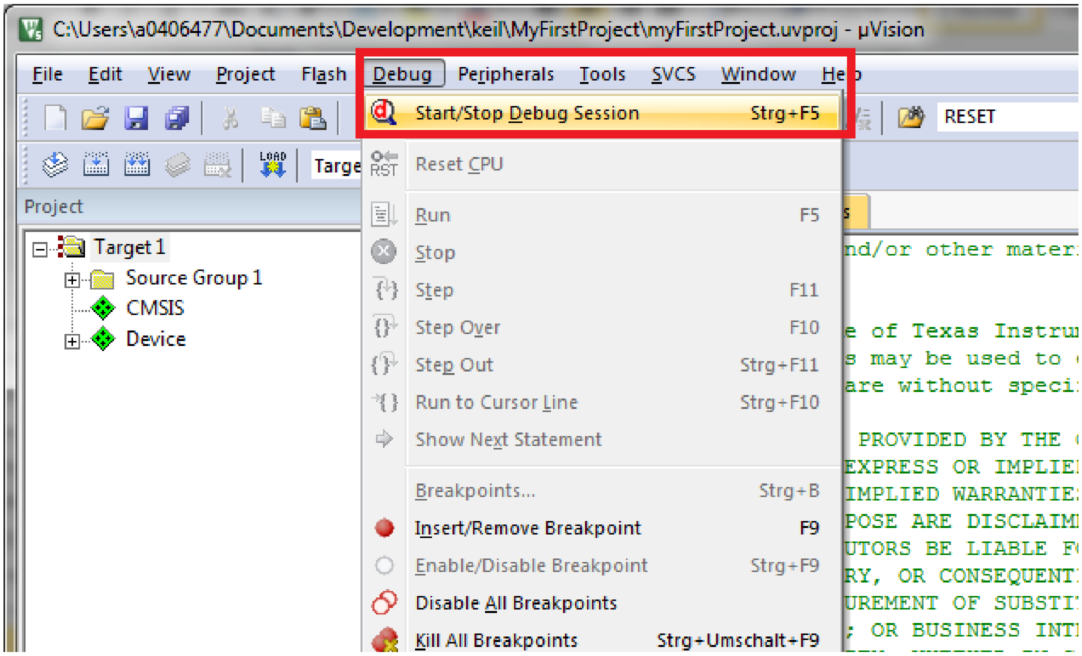
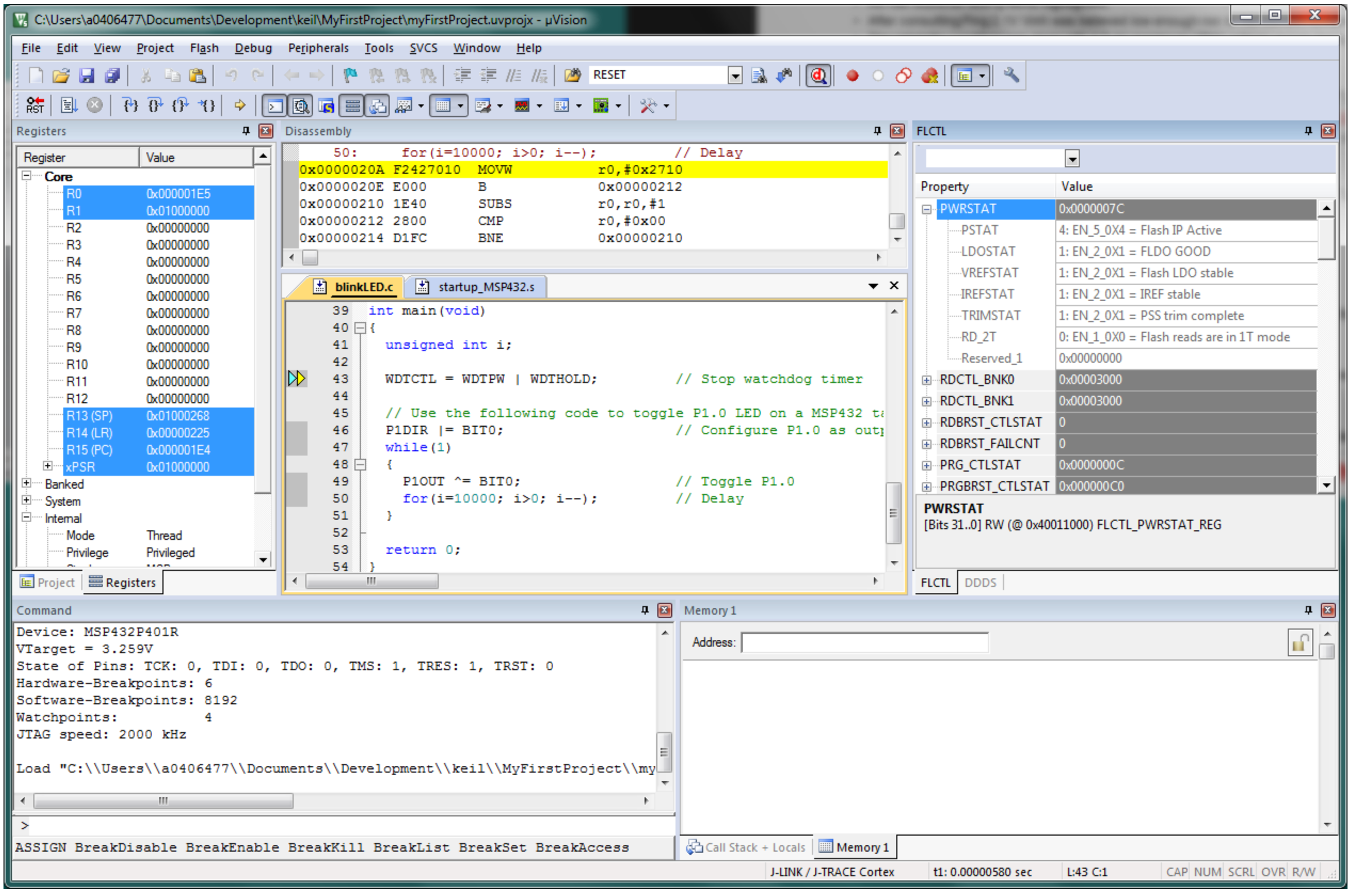


Figure 41. Start Debug Session

After the program image has been written to flash memory, the debug session starts and the debugger halts at the main function (see Figure 42).



C:\Users\a0406477\Documents\Development\keil\MyFirstProject\myFirstProject.uvprojx - µVision

Register	Value
Core	
R0	0x000001E5
R1	0x01000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x01000268
R14 (LR)	0x00000225
R15 (PC)	0x000001E4
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged

```
    50:      for(i=10000; i>0; i--);           // Delay
0x0000020A F2427010  MOVW         r0,#0x2710
0x0000020E E000      B            0x00000212
0x00000210 1E40      SUBS         r0,r0,#1
0x00000212 2800      CMP          r0,#0x00
0x00000214 D1FC      BNE          0x00000210
```

blinkLED.c   startup_MSP432.s

```
int main(void)
{
  unsigned int i;

  WDTCTL = WDTPW | WDTHOLD;          // Stop watchdog timer

  // Use the following code to toggle P1.0 LED on a MSP432 ta
  P1DIR |= BIT0;                     // Configure P1.0 as outp
  while(1)
  {
    P1OUT ^= BIT0;                   // Toggle P1.0
    for(i=10000; i>0; i--);          // Delay
  }

  return 0;
}
```

Property	Value
PWRSTAT	0x0000007C
PSTAT	4: EN_5_0X4 = Flash IP Active
LDOSTAT	1: EN_2_0X1 = FLDO GOOD
VREFSTAT	1: EN_2_0X1 = Flash LDO stable
IREFSTAT	1: EN_2_0X1 = IREF stable
TRIMSTAT	1: EN_2_0X1 = PSS trim complete
RD_2T	0: EN_1_0X0 = Flash reads are in 1T mode
Reserved_1	0x00000000
RDCTL_BNK0	0x00003000
RDCTL_BNK1	0x00003000
RDBRST_CTLSTAT	0
RDBRST_FAILCNT	0
PRG_CTLSTAT	0x0000000C
PRGBRST_CTLSTAT	0x000000C0

PWRSTAT
[Bits 31..0] RW (@ 0x40011000) FLCTL_PWRSTAT_REG

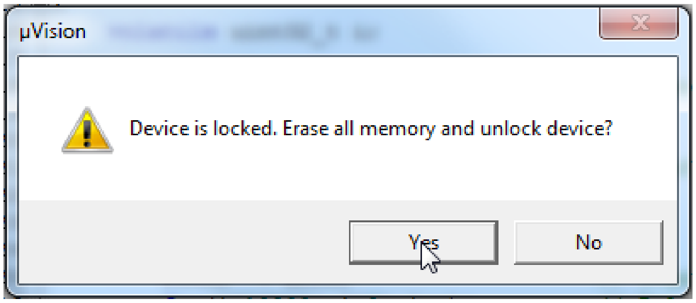
```
Device: MSP432P401R
VTarget = 3.259V
State of Pins: TCK: 0, TDI: 0, TDO: 0, TMS: 1, TRES: 1, TRST: 0
Hardware-Breakpoints: 6
Software-Breakpoints: 8192
Watchpoints:          4
JTAG speed: 2000 kHz

Load "C:\\Users\\a0406477\\Documents\\Development\\keil\\MyFirstProject\\my
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

J-LINK / J-TRACE Cortex   t1: 0.00000580 sec   L:43 C:1

Figure 42. Debug Session Successfully Started and Halted at main()

6.3.1 Working With Device Security (MSP432P4xx Devices Only)

If JTAG access is disabled on the device or the application needs to unlock a secure IP zone, Keil automatically runs a check on the device before downloading code. If Keil finds that the device has been secured, a dialog box is displayed (see Figure 43).



Device is locked. Erase all memory and unlock device?

Figure 43. Dialog Box Asking to Perform a Factory Reset

Click "Yes" to perform a factory reset. The device is unlocked so that code can be downloaded. Click "No" to quit the debug session. The device remains locked. After the code has been downloaded, the debug session starts.

6.4 Debugging Driver Lib in ROM

The MSP432P4xx family includes a complete peripheral driver library (DriverLib) fully integrated into the ROM memory. Developers can leverage the ROM DriverLib for multiple benefits including access to highly robust and tested APIs, single-cycle ROM execution speed at lower power consumption, and freeing up memory space for additional application code. Developers can gain access to ROM APIs by adding DriverLib header file to projects and linking to a prebuilt library.

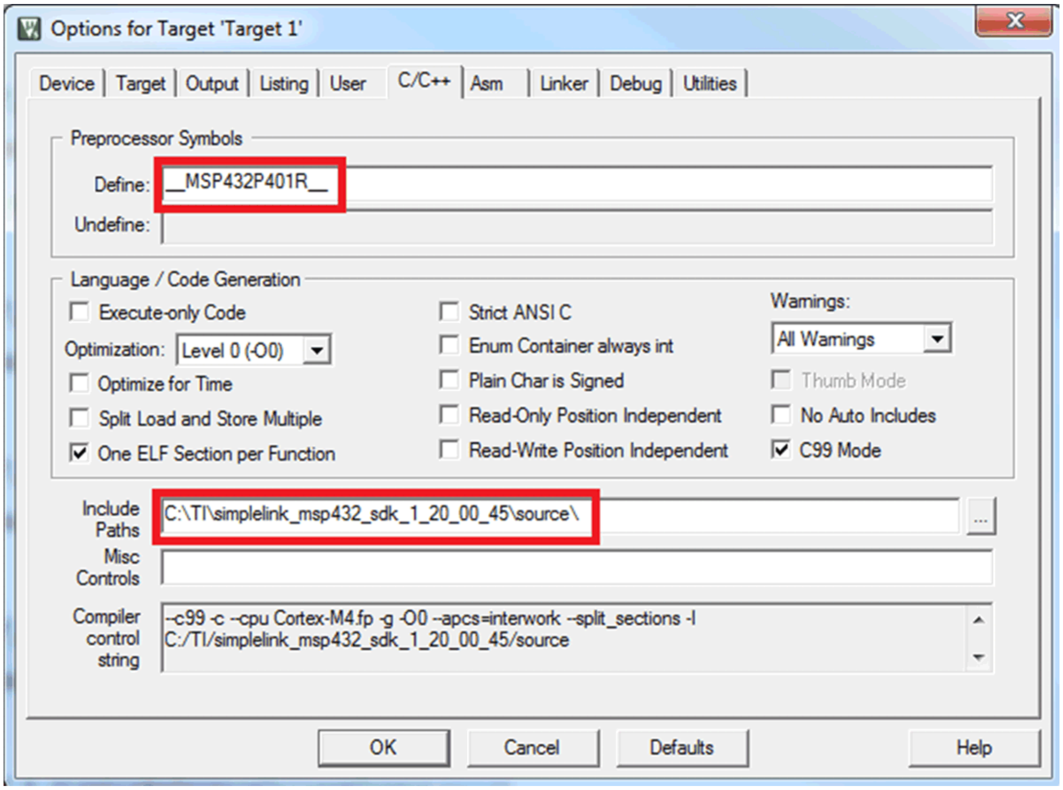
The driver library source code is now part of the SimpleLink MSP432 SDK. The driver library is a low-level software layer below the TI drivers, which are also part of the SDK. In the SDK, the DriverLib source code can be found in <SDK_InstallationPath>\source\ti\devices\msp432p4xx\driverlib.

For more information on MSP432P4xx Driver Library and what is provided in ROM DriverLib, see the documentation in the SimpleLink MSP432 SDK.

6.4.1 Add ROM Source and Symbol to an Existing SimpleLink MSP432 MCU Project

All DriverLib example projects from the SDK already have ROM debugging enabled. To add ROM debugging to an existing project, perform the following steps:

1. Include the driver library header in the source code:
   #include <ti/devices/msp432p4xx/driverlib/driverlib.h>
2. If the path to the driver library headers and sources is not yet included in the project, add it now. Right-click on the active µVision project, then select Options for Target. Add the include path to the driver library source folder in the C/C++ options (see Figure 44). For example, add "C:\TI\simplelink_msp432_sdk_1_20_00_45\source\".
3. Make sure that the correct preprocessor symbol for the device is defined (for example, __MSP432P401R__ for MSP432P401R devices).



Options for Target 'Target 1'

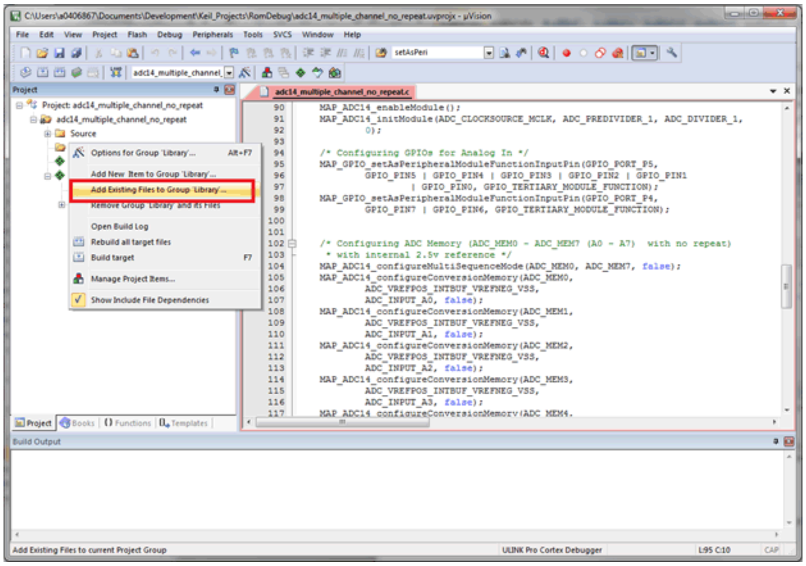
Preprocessor Symbols
Define: __MSP432P401R__
Undefine:

Language / Code Generation
□ Execute-only Code
Optimization: Level 0 (-O0)
□ Optimize for Time
□ Split Load and Store Multiple
☑ One ELF Section per Function
□ Strict ANSI C
□ Enum Container always int
□ Plain Char is Signed
□ Read-Only Position Independent
□ Read-Write Position Independent
Warnings: All Warnings
□ Thumb Mode
□ No Auto Includes
☑ C99 Mode

Include Paths: C:\TI\simplelink_msp432_sdk_1_20_00_45\source\
Misc Controls:
Compiler control string: --c99 -c --cpu Cortex-M4.fp -g -O0 --apcs=interwork --split_sections -I C:/TI/simplelink_msp432_sdk_1_20_00_45/source

Figure 44. Adding Path to the Driver Library Folder

4. To add the precompiled driver library to the project, right-click on the Library folder (create one if it does not exist), then select Add Existing Files to Group 'Library' and add the file msp432p4xx_driverlib.lib from the subfolder "\source\ti\devices\msp432p4xx\rom\keil" in the SDK directory. Make sure to select the file type Library file (*.lib) in the dialog.


C:\Users\a0406867\Documents\Development\Keil_Projects\RomDebug\adc14_multiple_channel_no_repeat.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
setAsPeri
adc14_multiple_channel_
Project
Project: adc14_multiple_channel_no_repeat
adc14_multiple_channel_no_repeat
Source
Options for Group 'Library'... Alt+F7
Add New Item to Group 'Library'...
Add Existing Files to Group 'Library'...
Remove Group 'Library' and its Files
Open Build Log
Rebuild all target files
Build target F7
Manage Project Items...
Show Include File Dependencies
Project
Books
Functions
Templates
adc14_multiple_channel_no_repeat.c
Build Output
Add Existing Files to current Project Group
ULINK Pro Cortex Debugger
L:95 C:10
CAP

```
    MAP_ADC14_enableModule();
    MAP_ADC14_initModule(ADC_CLOCKSOURCE_MCLK, ADC_PREDIVIDER_1, ADC_DIVIDER_1,
            0);

    /* Configuring GPIOs for Analog In */
    MAP_GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P5,
            GPIO_PIN5 | GPIO_PIN4 | GPIO_PIN3 | GPIO_PIN2 | GPIO_PIN1
                    | GPIO_PIN0, GPIO_TERTIARY_MODULE_FUNCTION);
    MAP_GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P4,
            GPIO_PIN7 | GPIO_PIN6, GPIO_TERTIARY_MODULE_FUNCTION);


    /* Configuring ADC Memory (ADC_MEM0 - ADC_MEM7 (A0 - A7)  with no repeat)
     * with internal 2.5v reference */
    MAP_ADC14_configureMultiSequenceMode(ADC_MEM0, ADC_MEM7, false);
    MAP_ADC14_configureConversionMemory(ADC_MEM0,
            ADC_VREFPOS_INTBUF_VREFNEG_VSS,
            ADC_INPUT_A0, false);
    MAP_ADC14_configureConversionMemory(ADC_MEM1,
            ADC_VREFPOS_INTBUF_VREFNEG_VSS,
            ADC_INPUT_A1, false);
    MAP_ADC14_configureConversionMemory(ADC_MEM2,
            ADC_VREFPOS_INTBUF_VREFNEG_VSS,
            ADC_INPUT_A2, false);
    MAP_ADC14_configureConversionMemory(ADC_MEM3,
            ADC_VREFPOS_INTBUF_VREFNEG_VSS,
            ADC_INPUT_A3, false);
    MAP_ADC14_configureConversionMemory(ADC_MEM4,
```

Figure 45. Adding Driver Library to a Project

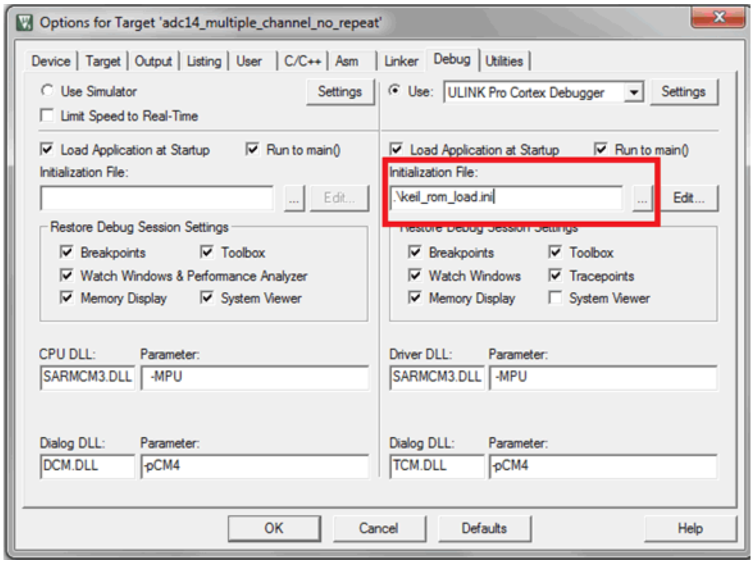
6.4.2 Load ROM Symbol at the Beginning of a Debug Session

The loading of the ROM debugging symbols is done through an initialization file. Copy and adapt the "keil_rom_load.ini" found in the "\source\ti\devices\msp432p4xx\rom\keil" directory in the SDK folder, or create a new file for the project.

In either case, the file should contain a line with loading instructions and the absolute or relative path to the driver library rom image. This example assumes that the SimpleLink MSP432 SDK has been installed in "C:\TI" and, therefore, the initialization file would look like the following:

LOAD
"C:\\TI\\simplelink_msp432_sdk_1_20_00_45\\source\\ti\\devices\\msp432p4xx\\rom\\
msp432_driverlib_rom_image.out" INCREMENTAL NOCODE

Right-click on the active µVision project and select Options for Target. Add the initialization file in the Debug section as shown in Figure 46.

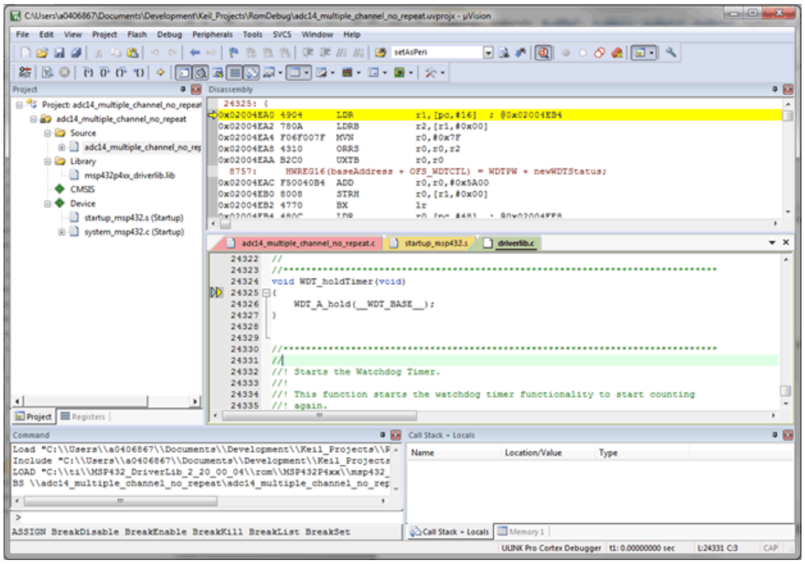

Options for Target 'adc14_multiple_channel_no_repeat'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
Settings
Settings
Edit...
Edit...
Restore Debug Session Settings
Restore Debug Session Settings
OK
Cancel
Defaults
Help

○ Use Simulator
☐ Limit Speed to Real-Time
☑ Load Application at Startup ☑ Run to main()
Initialization File:
Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows & Performance Analyzer
☑ Memory Display ☑ System Viewer
CPU DLL: SARMCM3.DLL Parameter: -MPU
Dialog DLL: DCM.DLL Parameter: -pCM4

● Use: ULINK Pro Cortex Debugger
☑ Load Application at Startup ☑ Run to main()
Initialization File: .\keil_rom_load.ini
Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows ☑ Tracepoints
☑ Memory Display ☐ System Viewer
Driver DLL: SARMCM3.DLL Parameter: -MPU
Dialog DLL: TCM.DLL Parameter: -pCM4

Figure 46. Adding Initialization File for Debugging

When the next debug session is started, you can step through the ROM API. The IDE automatically opens the corresponding source file (see Figure 47).


C:\Users\a0406867\Documents\Development\Keil_Projects\RomDebug\adc14_multiple_channel_no_repeat.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
setAsPeri
Project
Project: adc14_multiple_channel_no_repeat
adc14_multiple_channel_no_repeat
Source
adc14_multiple_channel_no_rep
Library
msp432p4xx_driverlib.lib
CMSIS
Device
startup_msp432.s (Startup)
system_msp432.c (Startup)
Project
Registers
Disassembly
adc14_multiple_channel_no_repeat.c
startup_msp432.s
driverlib.c
Command
Call Stack + Locals
Name
Location/Value
Type
Call Stack + Locals
Memory 1
ULINK Pro Cortex Debugger
t1: 0.00000000 sec
L:24331 C:3
CAP

```
  24325: {
0x02004EA0 4904      LDR        r1,[pc,#16]  ; @0x02004EB4
0x02004EA2 780A      LDRB       r2,[r1,#0x00]
0x02004EA4 F06F007F  MVN        r0,#0x7F
0x02004EA8 4310      ORRS       r0,r0,r2
0x02004EAA B2C0      UXTB       r0,r0
  8757:     HWREG16(baseAddress + OFS_WDTCTL) = WDTPW + newWDTStatus;
0x02004EAC F50040B4  ADD        r0,r0,#0x5A00
0x02004EB0 8008      STRH       r0,[r1,#0x00]
0x02004EB2 4770      BX         lr
0x02004EB4 480C      LDR        r0,[pc,#48]  ; @0x02004EE8
```

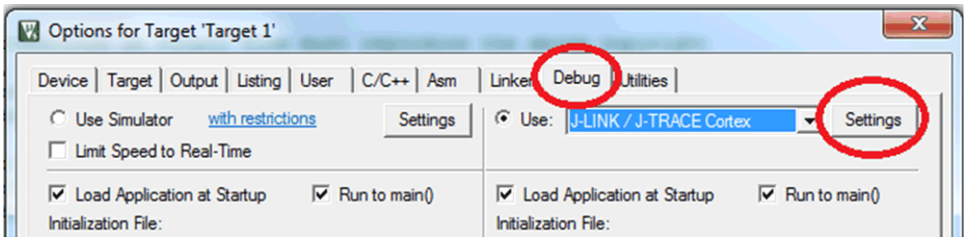
```
//
//*****************************************************************************
void WDT_holdTimer(void)
{
    WDT_A_hold(__WDT_BASE__);
}


//*****************************************************************************
//
//! Starts the Watchdog Timer.
//!
//! This function starts the watchdog timer functionality to start counting
//! again.
```

```
Load "C:\\Users\\a0406867\\Documents\\Development\\Keil_Projects\\R
Include "C:\\Users\\a0406867\\Documents\\Development\\Keil_Projects
LOAD "C:\\ti\\MSP432_DriverLib_2_20_00_04\\rom\\MSP432P4xx\\msp432_
BS \\adc14_multiple_channel_no_repeat\adc14_multiple_channel_no_rep
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Figure 47. Stepping Through ROM Driver Lib Source Code

6.5 Using Serial Wire Output (SWO) Hardware Trace Analyzer

Keil MDK supports the Arm CoreSight® components, including the Instrumentation Trace Macrocell (ITM) and Data Watchpoint and Trace Unit (DWT) .This user's guide concentrates on enabling the SWO trace in the µVision MDK. A more complete description of the MDK trace features can be found in the Arm® Keil® MDK 5 Getting Started Guide. Be aware that not all debug probes support the same features. To enable SWO, click on the options for the target and select the debug tab, then select Settings (see Figure 48).



○ Use Simulator

☐ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

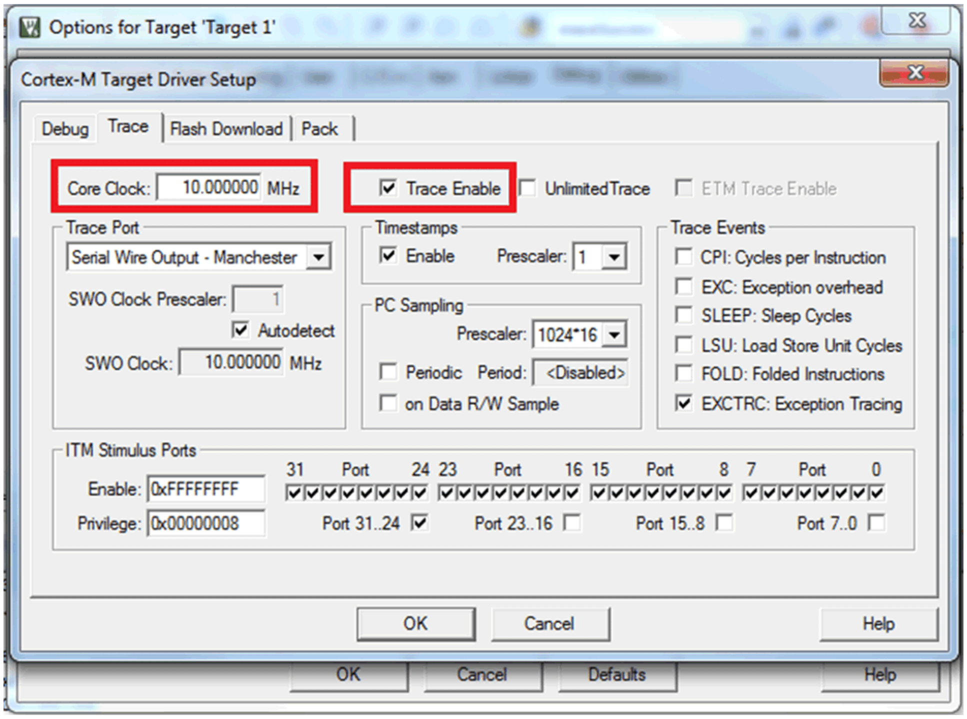
● Use: J-LINK / J-TRACE Cortex

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Figure 48. Debug Settings

In the Debug tab, select "SW" as port. In the Trace tab, set the flag for "Trace Enable", and make sure that the core clock is set to the correct value (see Figure 49).



Core Clock: 10.000000 MHz

☑ Trace Enable ☐ Unlimited Trace ☐ ETM Trace Enable

Trace Port

Serial Wire Output - Manchester

SWO Clock Prescaler: 1

☑ Autodetect

SWO Clock: 10.000000 MHz

Timestamps

☑ Enable Prescaler: 1

PC Sampling

Prescaler: 1024*16

☐ Periodic Period: <Disabled>

☐ on Data R/W Sample

Trace Events

☐ CPI: Cycles per Instruction

☐ EXC: Exception overhead

☐ SLEEP: Sleep Cycles

☐ LSU: Load Store Unit Cycles

☐ FOLD: Folded Instructions

☑ EXCTRC: Exception Tracing

ITM Stimulus Ports

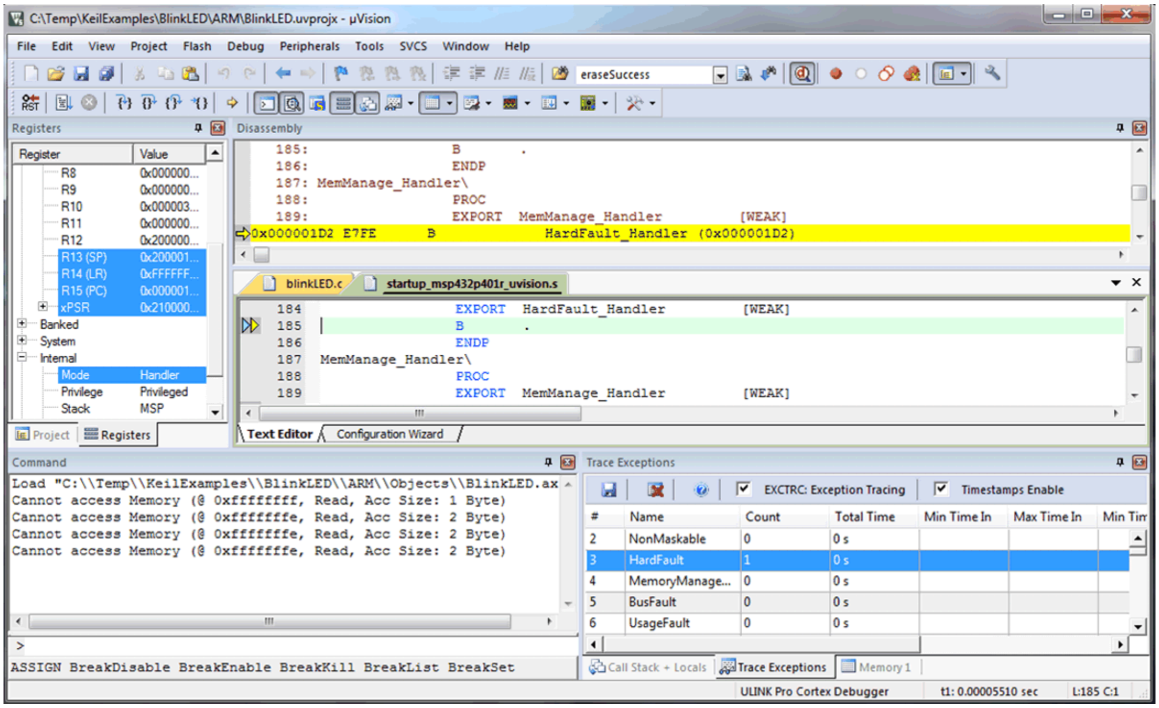
	31 Port 24	23 Port 16	15 Port 8	7 Port 0
Enable: 0xFFFFFFFF	☑☑☑☑☑☑☑☑	☑☑☑☑☑☑☑☑	☑☑☑☑☑☑☑☑	☑☑☑☑☑☑☑☑
Privilege: 0x00000008	Port 31..24 ☑	Port 23..16 ☐	Port 15..8 ☐	Port 7..0 ☐

Figure 49. Enabling Trace in Project Options

Now different trace options are available:

- Exception Trace
- Event Viewer
- Logic analyzer for tracing variables over time
- Debug (printf) viewer
- Event counters

For details on the µVision trace features, see the Arm® Keil® MDK 5 Getting Started Guide. Figure 50 shows an example exception trace for an MSP432 MCUs.


C:\Temp\KeilExamples\BlinkLED\ARM\BlinkLED.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
eraseSuccess
Registers
Disassembly
blinkLED.c
startup_msp432p401r_uvision.s
Project
Registers
Text Editor
Configuration Wizard
Command
Trace Exceptions
Call Stack + Locals
Trace Exceptions
Memory 1
ULINK Pro Cortex Debugger
t1: 0.00005510 sec
L:185 C:1

Register	Value
R8	0x000000...
R9	0x000000...
R10	0x000003...
R11	0x000000...
R12	0x200000...
R13 (SP)	0x200001...
R14 (LR)	0xFFFFFF...
R15 (PC)	0x000001...
xPSR	0x210000...
Banked	
System	
Internal	
Mode	Handler
Privilege	Privileged
Stack	MSP

```
   185:                    B        .
   186:                    ENDP
   187: MemManage_Handler\
   188:                    PROC
   189:                    EXPORT  MemManage_Handler          [WEAK]
0x000001D2 E7FE       B            HardFault_Handler (0x000001D2)
```

```
                EXPORT  HardFault_Handler          [WEAK]
                B       .
                ENDP
MemManage_Handler\
                PROC
                EXPORT  MemManage_Handler          [WEAK]
```

```
Load "C:\\Temp\\KeilExamples\\BlinkLED\\ARM\\Objects\\BlinkLED.ax
Cannot access Memory (@ 0xffffffff, Read, Acc Size: 1 Byte)
Cannot access Memory (@ 0xfffffffe, Read, Acc Size: 2 Byte)
Cannot access Memory (@ 0xfffffffe, Read, Acc Size: 2 Byte)
Cannot access Memory (@ 0xfffffffe, Read, Acc Size: 2 Byte)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

☑ EXCTRC: Exception Tracing ☑ Timestamps Enable

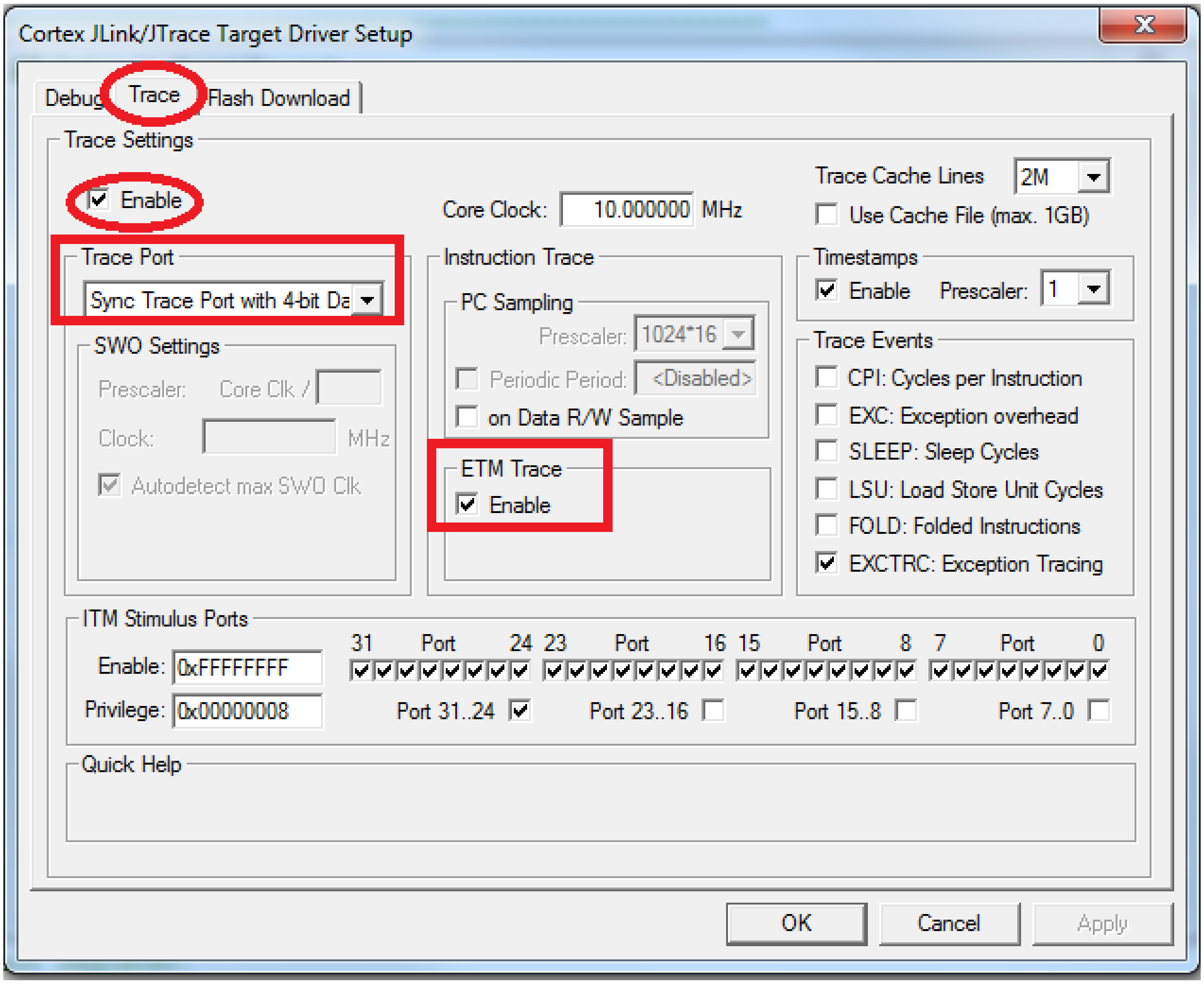
#	Name	Count	Total Time	Min Time In	Max Time In	Min Tim
2	NonMaskable	0	0 s			
3	HardFault	1	0 s			
4	MemoryManage...	0	0 s			
5	BusFault	0	0 s			
6	UsageFault	0	0 s			

Figure 50. Exception Trace

6.6 Using ETM Trace (MSP432E4 Devices Only)

The Embedded Trace Macrocell (ETM) on some TI devices in combination with a ETM trace enabled probe like µLink Pro and Segger J-Trace can perform nonintrusive tracing of every instruction in the MCU.

To activate ETM, open the settings of the debug probe (see Figure 48). In the Cortex-M target driver setup, select the "Trace" tab and set the checkmark for "Trace Enable". In the Trace Port pulldown menu, select "Sync Trace-Port with 4 bit data", and set the check mark for "ETM Trace Enable" (see Figure 51).



Trace Settings

☑ Enable

Core Clock: 10.000000 MHz

Trace Cache Lines 2M

☐ Use Cache File (max. 1GB)

Trace Port

Sync Trace Port with 4-bit Da

SWO Settings

Prescaler: Core Clk /

Clock: MHz

☑ Autodetect max SWO Clk

Instruction Trace

PC Sampling

Prescaler: 1024*16

☐ Periodic Period: <Disabled>

☐ on Data R/W Sample

ETM Trace

☑ Enable

Timestamps

☑ Enable Prescaler: 1

Trace Events

☐ CPI: Cycles per Instruction

☐ EXC: Exception overhead

☐ SLEEP: Sleep Cycles

☐ LSU: Load Store Unit Cycles

☐ FOLD: Folded Instructions

☑ EXCTRC: Exception Tracing

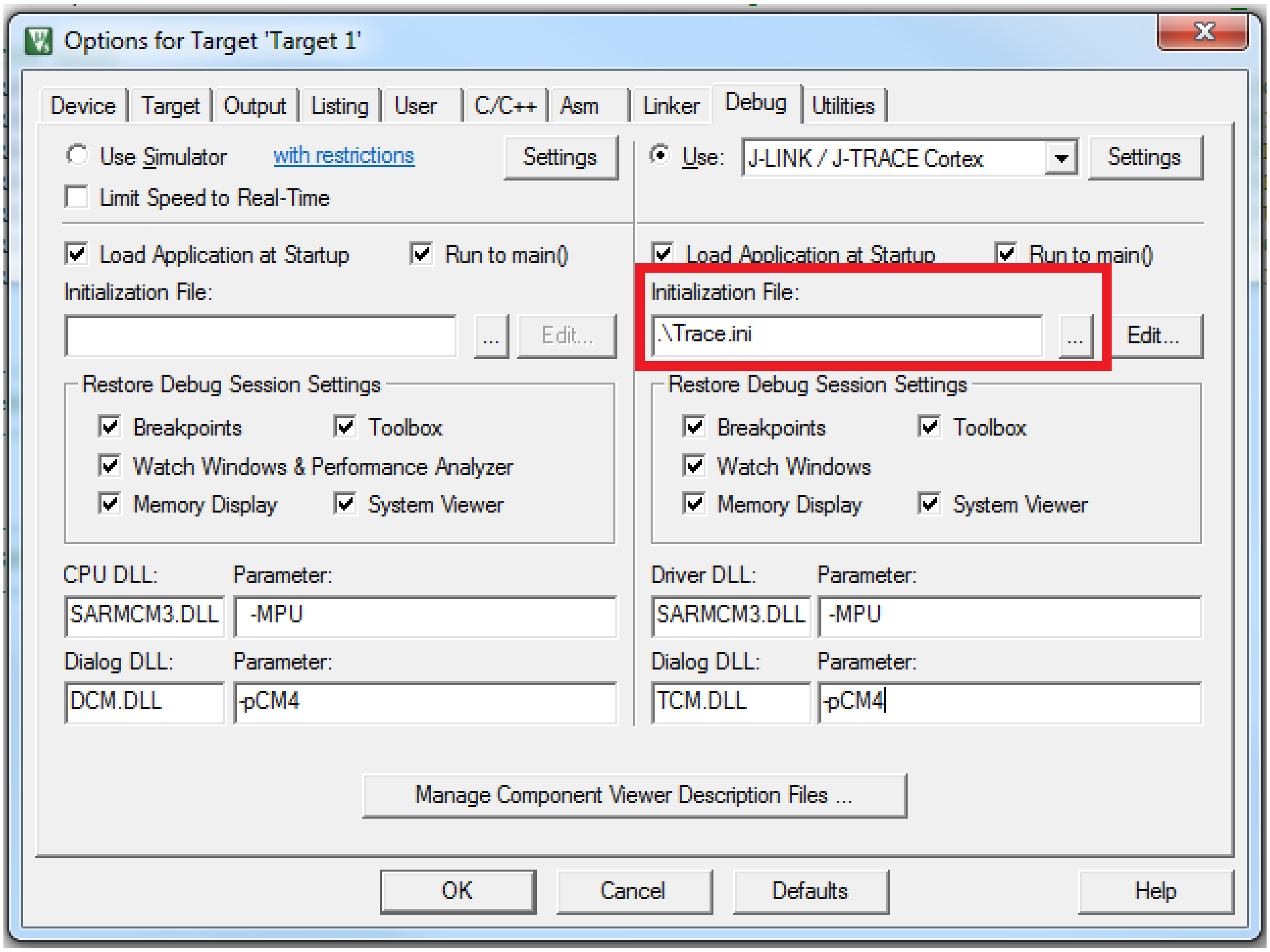
ITM Stimulus Ports

	31 Port 24	23 Port 16	15 Port 8	7 Port 0
Enable: 0xFFFFFFFF	☑☑☑☑☑☑☑☑	☑☑☑☑☑☑☑☑	☑☑☑☑☑☑☑☑	☑☑☑☑☑☑☑☑
Privilege: 0x00000008	Port 31..24 ☑	Port 23..16 ☐	Port 15..8 ☐	Port 7..0 ☐

Quick Help

Figure 51. ETM Debug Probe Setup

Adjust any other settings as needed and go back to the project options dialog debug tab. ETM trace needs an initialization file which is provided in the CMSIS DFP examples. If you are not working with a ARM trace probe, add Trace.ini to the project (see Figure 52). The Trace.ini file is located in the top level of each DFP example. Should you have created a project from scratch, copy the file to the project first.


Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities
Settings
Settings
...
Edit...
...
Edit...
Manage Component Viewer Description Files ...
OK
Cancel
Defaults
Help

○ Use Simulator with restrictions

☐ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows & Performance Analyzer

☑ Memory Display ☑ System Viewer

CPU DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: DCM.DLL Parameter: -pCM4

● Use: J-LINK / J-TRACE Cortex

☑ Load Application at Startup ☑ Run to main()

Initialization File: .\Trace.ini

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows

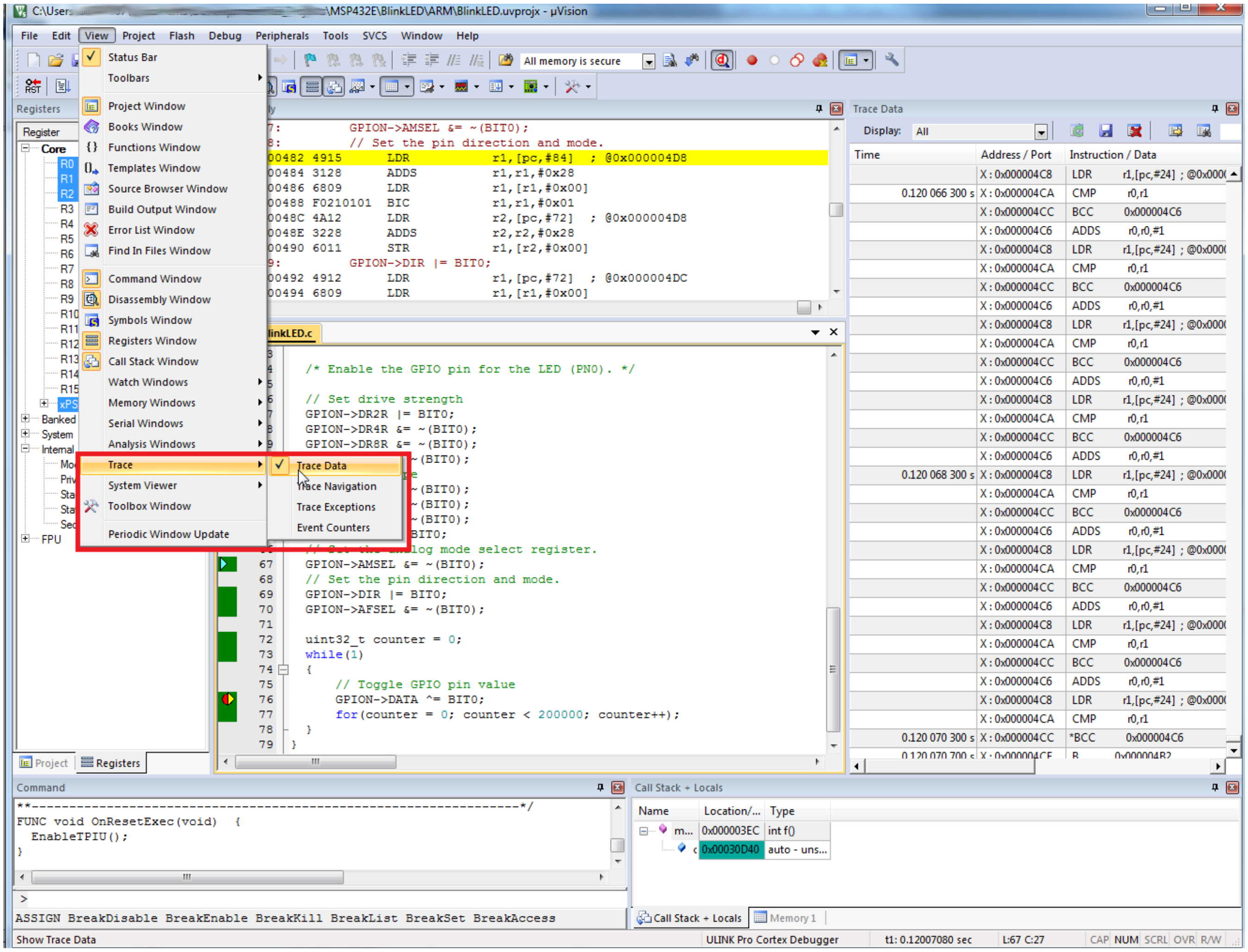
☑ Memory Display ☑ System Viewer

Driver DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: TCM.DLL Parameter: -pCM4

Figure 52. Adding Trace Initialization File

Trace data width is pre configured to 4 bit and ETM trace should now work out of the box. Start a debug session and open the Trace Data / Instruction Window from the View dialog (see Figure 53).



```
7:          GPION->AMSEL &= ~(BIT0);
8:          // Set the pin direction and mode.
00482 4915      LDR           r1,[pc,#84]  ; @0x000004D8
00484 3128      ADDS          r1,r1,#0x28
00486 6809      LDR           r1,[r1,#0x00]
00488 F0210101  BIC           r1,r1,#0x01
0048C 4A12      LDR           r2,[pc,#72]  ; @0x000004D8
0048E 3228      ADDS          r2,r2,#0x28
00490 6011      STR           r1,[r2,#0x00]
9:          GPION->DIR |= BIT0;
00492 4912      LDR           r1,[pc,#72]  ; @0x000004DC
00494 6809      LDR           r1,[r1,#0x00]
```

```

    /* Enable the GPIO pin for the LED (PN0). */

    // Set drive strength
    GPION->DR2R |= BIT0;
    GPION->DR4R &= ~(BIT0);
    GPION->DR8R &= ~(BIT0);
                   ~(BIT0);
                 e
                   ~(BIT0);
                   ~(BIT0);
                   ~(BIT0);
                  BIT0;
    //         log mode select register.
    GPION->AMSEL &= ~(BIT0);
    // Set the pin direction and mode.
    GPION->DIR |= BIT0;
    GPION->AFSEL &= ~(BIT0);

    uint32_t counter = 0;
    while(1)
    {
        // Toggle GPIO pin value
        GPION->DATA ^= BIT0;
        for(counter = 0; counter < 200000; counter++);
    }
}
```

Time	Address / Port	Instruction / Data	
	X : 0x000004C8	LDR	r1,[pc,#24] ; @0x0000
0.120 066 300 s	X : 0x000004CA	CMP	r0,r1
	X : 0x000004CC	BCC	0x000004C6
	X : 0x000004C6	ADDS	r0,r0,#1
	X : 0x000004C8	LDR	r1,[pc,#24] ; @0x0000
	X : 0x000004CA	CMP	r0,r1
	X : 0x000004CC	BCC	0x000004C6
	X : 0x000004C6	ADDS	r0,r0,#1
	X : 0x000004C8	LDR	r1,[pc,#24] ; @0x0000
	X : 0x000004CA	CMP	r0,r1
	X : 0x000004CC	BCC	0x000004C6
	X : 0x000004C6	ADDS	r0,r0,#1
	X : 0x000004C8	LDR	r1,[pc,#24] ; @0x0000
	X : 0x000004CA	CMP	r0,r1
	X : 0x000004CC	BCC	0x000004C6
	X : 0x000004C6	ADDS	r0,r0,#1
0.120 068 300 s	X : 0x000004C8	LDR	r1,[pc,#24] ; @0x0000
	X : 0x000004CA	CMP	r0,r1
	X : 0x000004CC	BCC	0x000004C6
	X : 0x000004C6	ADDS	r0,r0,#1
	X : 0x000004C8	LDR	r1,[pc,#24] ; @0x0000
	X : 0x000004CA	CMP	r0,r1
	X : 0x000004CC	BCC	0x000004C6
	X : 0x000004C6	ADDS	r0,r0,#1
	X : 0x000004C8	LDR	r1,[pc,#24] ; @0x0000
	X : 0x000004CA	CMP	r0,r1
	X : 0x000004CC	BCC	0x000004C6
	X : 0x000004C6	ADDS	r0,r0,#1
	X : 0x000004C8	LDR	r1,[pc,#24] ; @0x0000
	X : 0x000004CA	CMP	r0,r1
0.120 070 300 s	X : 0x000004CC	*BCC	0x000004C6
0.120 070 700 s	X : 0x000004CE	B	0x000004B2

```
**-----------------------------------------------------------------*/
FUNC void OnResetExec(void)  {
  EnableTPIU();
}
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Name	Location/...	Type
m...	0x000003EC	int f()
c	0x00030D40	auto - uns...

Show Trace Data   ULINK Pro Cortex Debugger   t1: 0.12007080 sec   L:67 C:27   CAP NUM SCRL OVR R/W

Figure 53. Show Data Trace Window

For more details on ETM trace, see the debug probe documentation.

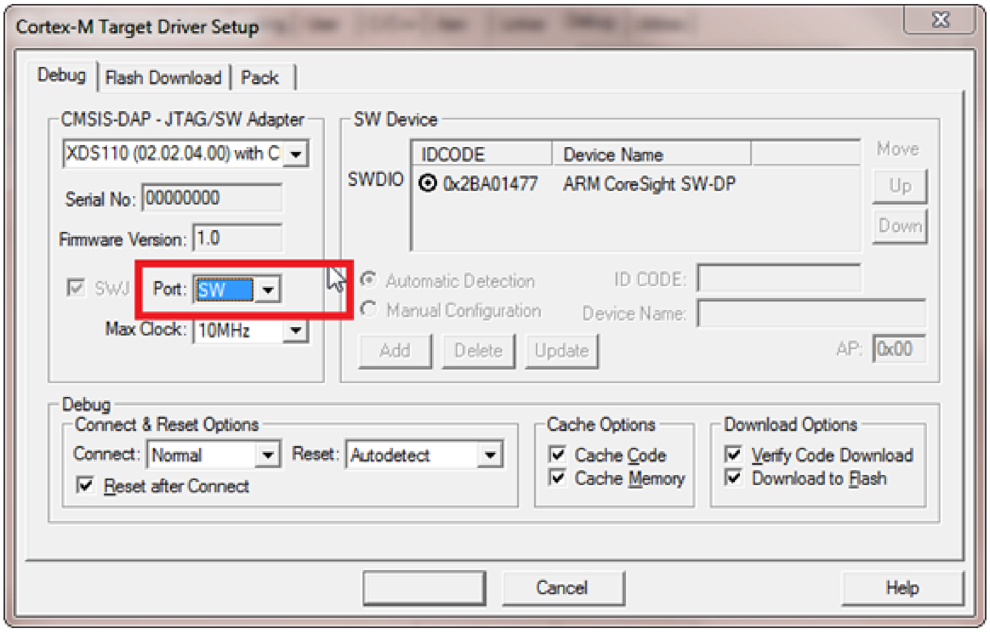
7 Frequently Asked Questions

Q: I cannot program my LaunchPad development kit; the IDE cannot connect to target. What's wrong?

A: Check the following:

- Check the debugger settings and change to Serial Wire Debug (SWD) without SWO. When the settings of Port J (PJSEL0 and PJSEL1 bits) are changed, full JTAG access is prevented on these pins. Changing to use SWD allows access through the dedicated debug pins only.

  Figure 54 shows how to configure the debugger to use SWD instead of JTAG by opening the debugger settings window.



Cortex-M Target Driver Setup

CMSIS-DAP - JTAG/SW Adapter: XDS110 (02.02.04.00) with C
Serial No: 00000000
Firmware Version: 1.0
☑ SWJ Port: SW
Max Clock: 10MHz

SW Device

	IDCODE	Device Name
SWDIO	● 0x2BA01477	ARM CoreSight SW-DP

○ Automatic Detection
○ Manual Configuration
ID CODE:
Device Name:
AP: 0x00

Debug
Connect & Reset Options: Connect: Normal Reset: Autodetect
☑ Reset after Connect
Cache Options: ☑ Cache Code ☑ Cache Memory
Download Options: ☑ Verify Code Download ☑ Download to Flash

Figure 54. Changing Debugger Setting to SWD

- If even this cannot connect, reset the device to factory settings. Review the *Device Security* section of the *Code Composer Studio™ IDE 7.1+ for SimpleLink™ MSP432™ Microcontrollers User's Guide* for information on how to perform a factory reset on the device.

Q: Why doesn't the back-channel UART on the MSP432 LaunchPad kit work with my serial terminal program at speeds faster than 56000 baud?

A: Certain serial terminal programs such as HTerm or the CCS built-in terminal might not work with the MSP432 LaunchPad kit at specific baud rates, resulting in the software not being able to open the virtual COM port or in the baud rate getting configured incorrectly. An issue with the LaunchPad kit's emulator firmware has been identified and will be fixed in the next release. Until the update is available, use Tera Term, ClearConnex, or HyperTerminal instead or reduce the baud rate to speeds of 38400 baud or lower.

Q: Problems plugging the MSP432 LaunchPad kit into a USB 3.0 Port

A: It has been observed that when the MSP432 LaunchPad kit is connected to USB 3.0 ports provided by a certain combination of USB 3.0 host controller hardware and associated device drivers that the IDE is unable to establish a debug session with the LaunchPad kit, resulting in an error message like "CS_DAP_0: Error connecting to the target: (Error -260 @ 0x0) An attempt to connect to the XDS110 failed." in the case of Code Composer Studio. In this case the CCS-provided low-level command line utility 'xdsdfu' will also not be able to establish a connection with the LaunchPad kit.

Specifically, this issue was observed on PCs running Windows 7 that show the "Renesas Electronics USB 3.0 Host Controller" and the associated "Renesas Electronics USB 3.0 Root Hub" in the device manager. After updating the associated Windows USB drivers to more recent versions obtained from the hardware vendor the issue went away. There might be other USB 3.0 hardware and device driver combinations that will lead to the same issue. If you think you might be affected try contacting the PC vendor or try locating and installing more recent versions of the USB 3.0 device drivers. Alternatively, connect the LaunchPad kit to an USB2.0 port on the PC if available.

Q: I cannot get the backchannel UART to connect. What's wrong?

A: Check the following:

- Do the baud rate in the host's terminal application and the eUSCI settings match?
- Are the appropriate jumpers in place on the isolation jumper block?
- Probe on RXD and send data from the host. If you don't see data, it might be a problem on the host side.
- Probe on TXD while sending data from the MSP432 MCU. If you don't see data, it might be a configuration problem with the eUSCI module.
- Consider the use of the hardware flow control lines (especially for higher baud rates).

Q: After updating the MSP432 device family pack, an existing project is not compiling.

A: Try to update the device configuration file (startup_msp432.c) by right clicking on it in the project folder and selecting "Update Config File".

Q: My MSP432E device has been locked. What can I do?

A: MSP432P4xx and MSP432E devices behave differently when locked, and the unlock process also differs. See the corresponding sections in the device-specific technical reference manual or data sheet.

For MSP432E devices, The XDS debug probe supports a command line option to remove the JTAG lock. In contrast to MSP432P4xx devices, this is not assisted by the IDE.

```
C:\ti\ccs_base\common\uscif>dbgjtag.exe -f @<XDS debug probe> -Y unlock, mode=msp432e4
```

The <XDS debug probe> option in this command must be changed to specify the XDS probe in use, for example, *xds110*.

Q: Watch, Memory, and Serial Wire Viewer windows update only when the program is stopped. What can I do?

A: Set the check mark for *Periodic Window Update* in the View dialog after starting a debug session. This setting enables continuous updates.

8 Additional Keil MDK-Arm Information

For more information about Keil MDK-Arm, see the following links:

- Keil Product Support
- Keil Microcontroller Development Kit Getting Started Guide
- Keil Microcontroller Development Kit Version 5
- Keil Microcontroller Development Kit Software Packs
- CMSIS-Pack Documentation

9 References

1. SimpleLink MSP432 SDK
2. Code Composer Studio™ IDE 7.1+ for SimpleLink™ MSP432™ Microcontrollers User's Guide
3. Texas Instruments XDS Emulation Software
4. Arm Keil MDK Getting Started Guide
5. Uniflash Standalone Flash Tool with additional features for TI MCUs and TI debug probes

Revision History

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

Changes from December 13, 2017 to November 26, 2018	Page
• Updated the paragraph that begins "Adjust any other settings as needed and..." in Section 6.6, *Using ETM Trace (MSP432E4 Devices Only)*	38

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265