This manual describes the use of IAR Embedded Workbench® for Arm® (EWARM) with the SimpleLink™ MSP432™ low-power microcontrollers.

This guide is for EWARM releases 8.10.1 and later. Some descriptions in this guide, like debugging, are also valid for EWARM versions before 8.10.1.

Contents
1 Installing Embedded Workbench for Arm................................................................. 4
2 Creating a SimpleLink Project From the MSP432 SDK ........................................ 4
3 Creating a Project From MSP432 CMSIS Device Family Pack (DFP) .................... 7
4 Debugging the Application....................................................................................... 8
5 Using Serial Wire Output (SWO) Hardware Trace Analyzer ................................ 24
6 Using ETM Trace (MSP432E4 Devices Only)...................................................... 26
7 Erasing the Bootloader (BSL) (MSP432P4xx Devices Only)................................. 27
8 EnergyTrace™ Technology.................................................................................... 29
9 Frequently Asked Questions.................................................................................. 35
10 Additional IAR EWARM Information .................................................................. 37
11 References............................................................................................................. 37

List of Figures
1 TI Resource Explorer............................................................................................... 5
2 SDK Quick Start Guide.......................................................................................... 6
3 Select Devices......................................................................................................... 7
4 Select Example....................................................................................................... 7
5 Change Device....................................................................................................... 8
6 Select Project Options........................................................................................... 9
7 Selecting the TI XDS Debug Probe....................................................................... 10
8 Debug Download Options...................................................................................... 11
9 Dialog Box Asking to Perform a Factory Reset.................................................... 12
10 Selecting MSP-FET as Debug Probe.................................................................... 12
11 Securing Device With MSP-FET......................................................................... 13
12 MSP-FET Release JTAG on Go........................................................................... 14
13 Select the Segger J-Link Debug Probe................................................................. 15
14 Enabling Usage of Flash Loaders......................................................................... 16
15 Using J-Link Commander to Enable Power Output to Target System.................. 16
16 Launch J-Link Control Panel................................................................................ 17
17 J-Link Control Panel............................................................................................ 18
18 J-Link Script Detecting That the Device Has Been Secured............................... 18
19 Device Has Been Unlocked and Erased.............................................................. 18
20 Selecting the IAR I-jet Debug Probe.................................................................... 19
21 Enabling Use of Flash Loaders............................................................................. 20
22 Debug Probe Setup.............................................................................................. 21
23 Dialog Box Asking to Perform a Factory Reset .......................................................... 21
24 Adding Driver Library to a Project ........................................................................... 22
25 Adding ROM Symbol to Debugger ........................................................................ 23
26 Stepping Through ROM DriverLib Source Code .................................................. 24
27 Trace Settings Configured From Running Debug Session Using Debugger Menu .... 25
28 SWO Trace Window Settings  .................................................................................. 25
29 SWO Configuration ................................................................................................. 25
30 ETM Trace With I-Jet Trace .................................................................................... 26
31 Edit Flash Loader Settings ..................................................................................... 27
32 Edit Info Memory Settings ..................................................................................... 27
33 Add --bsl_erase Option ......................................................................................... 28
34 Pulse Density and Current Flow .............................................................................. 29
35 Debug Session With EnergyTrace+ Windows .......................................................... 30
36 TI MSP-FET Emulator Pulldown Menu With EnergyTrace+ Related Functions ........ 31
37 Enabling the State Log Window ............................................................................ 31
38 State Log Window With EnergyTrace+ Data ......................................................... 32
39 State Log Summary With EnergyTrace+ Data ....................................................... 32
40 Power Log Setup Window ....................................................................................... 33
41 Power Log Window With EnergyTrace+ Data ....................................................... 33
42 Timeline With Power Log and State Graphs Disabled ............................................ 34
43 Timeline With EnergyTrace+ Data ......................................................................... 34
44 Choosing SWD in Debugger Settings ..................................................................... 36

Trademarks
SimpleLink, MSP432, E2E, LaunchPad, EnergyTrace are trademarks of Texas Instruments.
CoreSight is a trademark of Arm Limited.
Arm is a registered trademark of Arm Limited.
IAR Embedded Workbench is a registered trademark of IAR Systems.
All other trademarks are the property of their respective owners.
Preface: Read This First

How to Use This User's Guide

This manual describes only EWARM features 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 Documents on the Web.

Important MSP432 Documents on the Web

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

Documents that describe the IAR Embedded Workbench for Arm can be found at www.iar.com. The TI E2E™ Community support forums can provide additional help.

Information about the TI XDS100 and XDS200 debug probes is not included in this document and can be found at www.ti.com/tool/xds100 and www.ti.com/tool/xds200.

Documentation for third-party tools, such as the SEGGER J-Link debug probe, can be found on the respective third-party website.

If You Need Assistance

Support for the MSP432 devices and the hardware development tools is provided by the TI 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 Embedded Workbench for Arm

IAR Embedded Workbench for Arm (EWARM) IDE is available from the IAR website or the TI IAR kickstart page. MSP432 low-power microcontrollers require EWARM 7.40.2 or higher.

The TI MSP432 CMSIS device family packs require EWARM 8.10.1 or higher.

The SimpleLink MSP432 SDK requires EWARM 7.80.3 or higher. See the SimpleLink MSP432 SDK documentation for finding the right SDK for your IDE version.

CMSIS device packs (DFP) are an alternative way of adding device support to EWARM and support is available in EWARM 8.10.1 and higher. CMSIS DFPs are available for all SimpleLink MSP432 MCUs through the integrated pack manager.

2 Creating a SimpleLink Project From the MSP432 SDK

The SimpleLink MSP432 software development kit (SDK) contains software examples, projects, documentation, application notes, and training for all MSP432 devices. This includes example projects for IAR EWARM that work with MSP432 MCUs. For more information, visit the following TI SimpleLink SDK pages.

NOTE: The SimpleLink MSP432 SDK supports the MSP432P4xx devices, and the SimpleLink MSP432E4 SDK supports the MSP432E4 devices.

The SDK for your device and the examples can be either downloaded in form of an installer directly from the above links or, in the newer IAR EWARM 8.11.3+ versions, within the integrated TI resource explorer inside the IDE.

For accessing the online TI resource explorer open the IAR Information Center for Arm and click Example projects. In the section "Links to silicon partner example applications", click Texas Instruments SimpleLink MCUs and then TI SimpleLink™ MSP432™ Software Development Kit (SDK). This accesses the online TI resource explorer (see Figure 1).
In the Resource explorer, use the search dialog to limit the examples to the correct ones for your device. Any example you select starts the download of a zipped SDK for your device. Unpack this zipped file to C: TI.

**NOTE:** The TI example package in the "Example projects that can be downloaded" does not contain any examples for SimpleLink MCUs.

Regardless of how you obtained the SDK, all documentation for it can be found inside the SDK installation path in the docs folder. Open Documention_Overview.html from that folder and then navigate to the Quick Start Guide for your IDE (see Figure 2). Follow the instructions in this guide for required configuration before importing SDK examples.
Quick Start Guide for SimpleLink MSP432 SDK

Table of Contents

- 1. Introduction
- 2. Prerequisites
- 3 SDK - Download and Installation
  - 3.1 Using the SDK Online
  - 3.2 Installing the SDK
- 4. Quick Start for CCS IDE (CCS or GCC Projects)
  - 4.1 Download and Installation
  - 4.2 Running Your First Example
  - 4.3 Create a new SDK based User Application
  - 4.4 Migrate an existing MSP432 Project to an SDK-based Project
- 5. Quick Start for IAR IDE
  - 5.1 Download and Installation
  - 5.2 Running Your First Example
  - 5.3 Create a New SDK Based User Application
  - 5.4 Migrate an existing MSP432 Project to an SDK-based Project
- 6. Quick Start for Keil
  - 6.1 Download and Installation
  - 6.2 Running Your First Example
  - 6.3 Create a New SDK Based User Application
  - 6.4 Migrate an existing MSP432 Project to an SDK-based Project
- 7. Quick Start for Makefile Users
  - 7.1 Update imports.mk
  - 7.2 Build RTOS config
  - 7.3 Build the makefile

Figure 2. SDK Quick Start Guide

In particular, make sure to execute the following steps:

- Configuring custom argument variables
- RTOS configuration, if you intend to use an RTOS

A list of examples for IAR can be found under `<SDK installation path>/ tools/iar/Examples.html`. Follow the instructions in the quick start guide on how to import these examples to the EWARM IDE.
3 Creating a Project From MSP432 CMSIS Device Family Pack (DFP)

IAR EWARM 8.10 and higher supports CMSIS packs (see the CMSIS Pack Documentation for a description of CMSIS packs). The latest Texas Instruments MSP432P4xx and MSP432E4 Device Family Packs are supported by IAR EWARM IDE 8.10 or higher. The packs can be downloaded with the pack manager in the IDE or directly from the MSP432 CMSIS device family pack page. A device family pack adds device support to the IDE and features some basic examples.

**NOTE:** Examples from the SimpleLink MSP432 SDK do not build on CMSIS device family packs in IAR EWARM. These rely on the native device support through the IDE.

**NOTE:** The following description is for the CMSIS Manager in IAR EWARM 8.30. The procedure might be different in other versions of IAR EWARM.

To create an example from the device family pack:

1. In a new workspace, click to start the CMSIS manager, go to the Devices tab, select the desired device family, and go to the examples tab (see Figure 3).

![Figure 3. Select Devices](image)

2. In the examples tab, select the desired example and click **Copy** to add the example to your workspace (see Figure 4).

![Figure 4. Select Example](image)
3. To change the device, if needed, click the device tab in the lower left and then click the **Change** button (see Figure 5).

![Figure 5. Change Device](image.jpg)

For projects based on CMSIS packs, most setting must be done in the CMSIS manager. See the IAR EWARM help for more information.

4. Save the settings and return to your workspace. The example project is ready to be compiled.

### Debugging the Application

The following debug probes can be used with MSP432 MCUs and EWARM.

- Texas Instruments XDS100v2, XDS100v3, XDS200, XDS110 (including the **XDS110 stand-alone probe**)
- TI MSP-FET (MSP432P4xx Devices Only)
- IAR I-jet
- Segger J-Link

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

Be aware that for some debug probes power has to be supplied externally to the device. Check the probes user guides for details. For the TI XDS110 stand-alone probe, see the **XDS110 Debug Probe User's Guide**.
4.1 Using TI XDS100, XDS110, and XDS200 Debug Probes

TI offers a range of debug probes for Arm-based devices, including the XDS100v2, XDS100v3, XDS110, and XDS200 debug probes.

**NOTE:** TI XDS110 debug probes are enabled for use through the CMSIS-DAP protocol. However, the TI XDS native drivers currently give higher performance.

To use them with EWARM, installation of the XDS emulation package is required. A copy of the emulation package is located in the EWARM installation under `arm\drivers\ti-xds`. See the Readme.txt document that is also located in this folder. TI recommends installing the emulation package in `c:\ti\xds\ewarm_version`. Do not use XDS emulation packages from other EWARM versions, as they might not be compatible and can result in errors when debugging.

When the emulation package has been installed, XDS debug probes can be selected in the Project Options menu. Right click the active project, then select **Options** (see Figure 6).

![Figure 6. Select Project Options](image)
Click **Debugger** in the left pane and select **TI XDS**. Then click XDS debuggers in the left pane and select the correct XDS debug probe from the list. (see **Figure 7**).

![Figure 7. Selecting the TI XDS Debug Probe](image-url)
Make sure that the **Use flash loader(s)** option is selected in the Download tab (see Figure 8).

![Figure 8. Debug Download Options](image-url)
4.1.1 Working With Device Security (MSP432P4xx Devices Only)

If you have disabled JTAG access on the device or are working on an application where you need to unlock a secure IP zone, IAR Embedded Workbench automatically runs a check on the device before downloading code. If IAR finds that the device has been secured, a dialog box opens as shown in Figure 9.

Figure 9. Dialog Box Asking to Perform a Factory Reset

Click Yes to perform a factory reset and unlock the device so that code can be downloaded. Click No to end the debug session and leave the device locked. After the code has been downloaded, the debug session starts.

4.2 Using TI MSP-FET (MSP432P4xx Devices Only)

To use the MSP-FET for debugging MSP432x devices, IAR 7.60 or higher is required. No additional driver installation is required.

TI MSP-FET debug probes can be selected in the Project Options menu. Right click the active project, then select Options (see Figure 6).

Click Debugger in the left pane, and select TI MSP-FET (see Figure 10).

Figure 10. Selecting MSP-FET as Debug Probe
4.2.1 Working With Device Security (MSP432P4xx Devices Only)

Similar to the process when using TI XDS debug probes, you can unlock a protected device. If you have disabled JTAG access on the device or are working on an application where you need to unlock a secure IP zone, IAR Embedded Workbench automatically runs a check on the device before downloading code. If IAR finds that the device has been secured, a dialog box opens as shown in Figure 9.

Click **Yes** to perform a factory reset and unlock the device so that code can be downloaded. Click **No** to end the debug session and leave the device locked. After the code has been downloaded, the debug session starts.

The easy-to-use TI MSP-FET offers to secure the device by disabling JTAG access. Click **Secure Device** in the **TI MSP-FET** menu when a debugging session is running (see Figure 11).

![Figure 11. Securing Device With MSP-FET](image)

If only parts of the device are IP protected, MSP-FET displays a console message during connect: "IP protection is enabled on the device. Not all flash memory locations may be readable or writable". See the online documentation on [www.ti.com/msp432](http://www.ti.com/msp432) for details and tools for handling device security.
### 4.2.2 Release JTAG on Go Option

When debugging low-power modes the **Release JTAG on Go** option for the MSP-FET (see Figure 12) should be used to make sure that the debugger does not lose communication with the device. However, with this option selected, the debugger cannot identify breakpoints.

![Figure 12. MSP-FET Release JTAG on Go](image)

After enabling the **Release JTAG on Go** option, execute the **RUN** command.

### 4.2.3 Further Advice for MSP-FET in IAR EWARM

If the $V_{CC}$ voltage is not high enough when trying to erase or write flash memory, the following message is displayed in the console:

"Target device supply voltage is too low for Flash erase/programming".

Raise the supply voltage to correct this error.

Do not connect through a USB hub when performing a firmware update on the MSP-FET, the MSP-FET430UIF, or a LaunchPad™ development kit.
4.3 Using Segger J-Link Debug Probe

To use the Segger J-Link debug probe, right click the active project, then select Options. From the pulldown menu select J-Link/J-Trace (see Figure 13).

![Figure 13. Select the Segger J-Link Debug Probe](image)

**Figure 13. Select the Segger J-Link Debug Probe**
Next, switch to the Download pane and enable the **Use flash loader(s)** option (see Figure 14).

![Figure 14. Enabling Usage of Flash Loaders](image)

When using a target socket board for the MSP432 MCUs, you can benefit from the 5-V voltage output the J-Link provides on pin 19 of its Cortex-M debug connector. This option needs to be enabled through the J-Link Commander, a console application available from Segger. When enabled, the debug probe provide 5 V to the target system. See the *MSP432™ SimpleLink™ Microcontrollers Hardware Tools User's Guide* how to configure the target socket board to use the 5-V power supply to generate 3.3-V device voltage and to Segger's documentation how to enable the voltage output.

Figure 15 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.

![Figure 15. Using J-Link Commander to Enable Power Output to Target System](image)

Now you can download the program and debug using the Segger J-Link debug probe with EWARM.
4.3.1 Working With Device Security (MSP432P4xx Devices Only)

If you have disabled JTAG access on the device or are working on an application where you need to unlock a secure IP zone, a J-Link Script needs to 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 16).

Figure 16. Launch J-Link Control Panel
After launching the J-Link control panel, add the MSP432 J-Link script provided in IAR. Figure 17 shows the location of the script file.

The J-Link script now runs at the launch of every 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 can be erased (see Figure 18).

Click OK on the dialog box to issue a factory reset, which erases any code present on the device and then starts to download the compiled code. When the factory reset is a complete, a confirmation dialog box reports that the process is complete (see Figure 19).
4.4 Using IAR I-jet Debug Probe

To use the IAR I-jet debug probe, right click the active project, then click **Options**. From the pulldown **Driver** menu, select **I-jet / JTAGjet** (see Figure 20).

![Figure 20. Selecting the IAR I-jet Debug Probe](image-url)
Next, switch to the Download pane and enable the **Use flash loader(s)** option (see Figure 21).

![Figure 21. Enabling Use of Flash Loaders](image-url)
When using a target socket board for the MSP432 MCUs, you can benefit from the 5-V voltage output the I-jet provides on pin 19 of its Cortex-M debug connector. Go to the I-jet/JTAGjet specific menu and enable the **Target Power** option (see Figure 22). See the [*MSP432™ SimpleLink™ Microcontrollers Hardware Tools User’s Guide*](https://www.ti.com) how to configure the target socket board to use the 5-V power supply to generate 3.3-V device voltage.

![Figure 22. Debug Probe Setup](image)

Now you can download the program and debug using the IAR I-jet debug probe with EWARM.

### 4.4.1 Working With Device Security (MSP432P4xx Devices Only)

If you have disabled JTAG access on the device or are working on an application where you need to unlock a secure IP zone, IAR Embedded Workbench automatically runs a check on the device before downloading code if you are using an IAR I-Jet. If IAR finds that the device has been secured, a dialog box opens as shown in Figure 23.

![Figure 23. Dialog Box Asking to Perform a Factory Reset](image)

Click **Yes** to perform a factory reset and unlock the device so that code can be downloaded. Click **No** to end the debug session and leave the device locked. After the code has been downloaded, the debug session starts.
4.5  **Debugging Driver Lib in ROM**

The MSP432P4xx family includes a complete peripheral driver library (DriverLib) that is 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.

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

4.5.1 **Enable Use of Software in ROM in MSP432 Project (MSP432P4xx Devices Only)**

If the path to the driver library headers and sources is not yet included in the project, add it. Click **Project Options** (Alt+F7), then select **C/C++ Compiler**, and select the **Preprocessor** tab. Add the include path to the MSP432P4xx driver library source folder in the **Additional include directories** field (see Figure 24). For example, add `C:\ti\MSP432_DriverLib_2_20_00_08\driverlib\MSP432P4xx`, assuming that the driver library was extracted to `C:\ti\MSP432_DriverLib_2_20_00_08`.

Then in the **Defined Symbols** field, add `TARGET_IS_MSP432P4XX` to enable the Software in ROM use. (see Figure 24).

![Figure 24. Adding Driver Library to a Project](image-url)
4.5.2 Load ROM Symbol Into Debugger

To enable debug of the software in ROM, the corresponding symbol should be loaded. Click Project Options (ALT+F7) and select Debugger. Then select the Images tab and proceed as follows (see Figure 25):

1. Check the box Download extra image
2. In the Path field, click the browse button and locate the ROM debugging symbols (*.out). The MSP432P4xx Driver Library provides this file msp432_driverlib_rom_image.out available in C:\ti\MSP432_DriverLib_2_20_00_08\rom\MSP432P4xx\msp432_driverlib_rom_image.out. It is important to keep the driverlib.c file on the same path as the symbol file in the same directory.
3. In the Offset field, type 0x0.
4. Click OK and start debugging.

![Figure 25. Adding ROM Symbol to Debugger](image)

When the next debug session is started, you can step through the ROM API. The IDE automatically opens the corresponding source file (see Figure 26). MSPWare also provides the software in ROM symbol available in C:\ti\MSPWare_2_20_00_19\driverlib\rom\MSP432P4xx.
The SWO Trace tools for MSP432 MCUs are implemented using the features of the Arm CoreSight™ components, especially the Instrumentation Trace Macrocell (ITM) and Data Watchpoint and Trace Unit (DWT) (ETM is not present in MSP432P4xx MCUs).

To use SWO trace, an SWO-enabled debug probe must be used. In IAR EWARM, this is currently possible with IAR I-JET, Segger J-Link, TI XDS200, and TI XDS110 debug probes. Change the interface of the debug probe to SWD and start a debug session. Trace itself is configured from a running debug session.

Adjust the trace settings to the project settings and frequencies, and start with a lightweight setting. For example, to enable SWO trace for a bare metal blinky example, use SWO trace settings as shown in the following figures. Figure 27 shows the menu to open the configuration dialogs.

5 Using Serial Wire Output (SWO) Hardware Trace Analyzer

The SWO Trace tools for MSP432 MCUs are implemented using the features of the Arm CoreSight™ components, especially the Instrumentation Trace Macrocell (ITM) and Data Watchpoint and Trace Unit (DWT) (ETM is not present in MSP432P4xx MCUs).

To use SWO trace, an SWO-enabled debug probe must be used. In IAR EWARM, this is currently possible with IAR I-JET, Segger J-Link, TI XDS200, and TI XDS110 debug probes. Change the interface of the debug probe to SWD and start a debug session. Trace itself is configured from a running debug session.

Adjust the trace settings to the project settings and frequencies, and start with a lightweight setting. For example, to enable SWO trace for a bare metal blinky example, use SWO trace settings as shown in the following figures. Figure 27 shows the menu to open the configuration dialogs.
Figure 27. Trace Settings Configured From Running Debug Session Using Debugger Menu

Force **Time Stamps** and **PC Samples** in the SWO Trace Window Settings (see Figure 28).

Figure 28. SWO Trace Window Settings

Override **Clock Setup** and set the sample rate to a low value in SWO Configuration (see Figure 29).

Figure 29. SWO Configuration
After that, open the trace windows of interest (for example, function profiling and interrupt profiling) and make sure that all the traces are enabled in the corresponding windows. Optionally, instruct the code to output custom events through the Terminal I/O window by adding the following lines:

```c
#include "arm_itm.h" ITM_EVENT8(1, 'a');
```

The prerequisite for this is to have the corresponding ITM stimulus ports enabled in the SWO configuration (see Figure 29).

For more information on SWO trace in IAR EWARM, see the user guides on www.iar.com.

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 IAR I-Jet Trace and Segger J-Trace entitles you to perform non-intrusive tracing of every single instruction in your MCU.

For activating ETM just start a debug session and select **ETM Trace** from the debugger menu (see Figure 30). Trace data width is preconfigured to 4 bit and ETM trace should work out of the box.

![Figure 30. ETM Trace With I-Jet Trace](image)

**NOTE:** ETM trace configuration settings for MSP432E devices has been added recently only. In order to have the best user experience make sure you have the latest IAR EWARM version or the latest MSP432E CMSIS DFP installed.

For more details on ETM trace refer to your debug probe documentation.
Erasing the Bootloader (BSL) (MSP432P4xx Devices Only)

The BSL is a program built into an MSP432 microcontroller designed to communicate with the device, primarily for the purpose of reading and writing to memory. The BSL can be erased and rewritten but, by default, the IAR Embedded Workbench tools protect it from accidental deletion.

To erase the BSL right click the active project, then select Options. Click the Debugger category on the left side and the Download tab. In the Download tab make sure Use flash loaders(s) and Override default .board file are checked. Then click the Edit button (see Figure 31).

After you click the edit button, a dialog window pops up that allows to change the flash loader settings. Select the section of memory where the BSL resides (0x200000 to 0x203FFF), and then click Edit (Figure 32).
You can now specify extra parameters to pass to the RAM loader. There is also a list of parameters in the Parameter descriptions info group. To erase the BSL enter "--bsl_erase" in the Extra parameters field as shown in Figure 33.

![Figure 33. Add --bsl_erase Option](image)

The BSL memory is now unprotected during code download, and the user can erase and overwrite the current BSL. For more information regarding the BSL, see the MSP432™ SimpleLink™ Microcontrollers Bootloader (BSL) User's Guide.
8 EnergyTrace™ Technology

EnergyTrace™ Technology is an energy-based code analysis tool that measures and displays the application's energy profile and helps to optimize the application for ultra-low power consumption.

MSP432 MCUs with built-in EnergyTrace+[CPU State] (or in short EnergyTrace+) technology allow real-time monitoring of many internal device states while user program code executes. EnergyTrace+ technology is supported on selected MSP432 devices and debuggers (for example, MSP432P4xx devices).

EnergyTrace mode (without the "+") is the base of EnergyTrace Technology and enables analog energy measurement to determine the energy consumption of an application but does not correlate it to internal device information. The EnergyTrace mode is available for all MSP432 devices with selected debuggers, including IAR EWARM.

Devices that support EnergyTrace technology also benefit from the XDS110 EnergyTrace™ High Dynamic Range (ETHDR) debug probe add-on.

8.1 Energy Measurement

Debuggers with EnergyTrace technology support include a new and unique way of continuously measuring the energy supplied to a target microcontroller that differs considerably from the well-known method of amplifying and sampling the voltage drop over a shunt resistor at discrete times. A software controlled DC-DC converter is used to generate the target power supply. The time density of the DC-DC converter charge pulses equals the energy consumption of the target microcontroller. A built-in on-the-fly calibration circuit defines the energy equivalent of a single DC-DC charge pulse.

Figure 34 shows the energy measurement principle. Periods with a small number of charge pulses per time unit indicate low energy consumption and thus low current flow. Periods with a high number of charge pulses per time unit indicate high energy consumption and also a high current consumption. Each charge pulse leads to a rise of the output voltage VOUT, which results in an unavoidable voltage ripple common to all DC-DC converters.

![Figure 34. Pulse Density and Current Flow](image)

The benefit of sampling continuously is evident: even the shortest device activity that consumes energy contributes to the overall recorded energy. No shunt-based measurement system can achieve this.

8.2 IAR Embedded Workbench for Arm Integration

EnergyTrace Technology is available as part of IAR Embedded Workbench for Arm microcontrollers version 7.80 or higher exclusively for MSP432 MCUs. During debugging of an application, additional windows are available if the debug probe and the target device support EnergyTrace Technology.

The EnergyTrace+ Technology is available only when using the TI MSP-FET debug probe (MSP432P4xx Devices Only).

Devices that support EnergyTrace+ Technology allow sampling of internal device states while an application is executing (see Figure 35).
Figure 35. Debug Session With EnergyTrace+ Windows
Click the TI MSP-FET Emulator menu for EnergyTrace+ related entries (see Figure 36).

- State Log
- Power Log
- Timeline

![Figure 36. TI MSP-FET Emulator Pulldown Menu With EnergyTrace+ Related Functions](image)

To enable the functions that are related to EnergyTrace technology, right click in the respective window and select **Enable** (see Figure 37).

![Figure 37. Enabling the State Log Window](image)
8.2.1 State Log

CPU activity is grouped under States. When a program executes, digital data is collected from the target device and displayed in list format (see Figure 38). The State Log shows at what point in time the CPU has been activated or deactivated and gives a reference to the program counter location where this happened.

<table>
<thead>
<tr>
<th>Time</th>
<th>Source</th>
<th>Status</th>
<th>Program Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>12s</td>
<td>CPU LPMx</td>
<td>Off</td>
<td>0x2005C56</td>
</tr>
<tr>
<td>12s</td>
<td>CPU Active Mode</td>
<td>On</td>
<td>0x2005C56</td>
</tr>
<tr>
<td>13s</td>
<td>CPU Active Mode</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>13s</td>
<td>CPU LPMx</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>13s</td>
<td>CPU Active Mode</td>
<td>Off</td>
<td>0x2002BC0</td>
</tr>
<tr>
<td>13s</td>
<td>CPU Active Mode</td>
<td>On</td>
<td>0x2002BC0</td>
</tr>
<tr>
<td>14s</td>
<td>CPU Active Mode</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>14s</td>
<td>CPU LPMx</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>14s</td>
<td>CPU LPMx</td>
<td>Off</td>
<td>0x20042C8</td>
</tr>
<tr>
<td>14s</td>
<td>CPU Active Mode</td>
<td>On</td>
<td>0x20042C8</td>
</tr>
<tr>
<td>15s</td>
<td>CPU Active Mode</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>15s</td>
<td>CPU LPMx</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>15s</td>
<td>CPU LPMx</td>
<td>Off</td>
<td>0x240</td>
</tr>
<tr>
<td>15s</td>
<td>CPU Active Mode</td>
<td>On</td>
<td>0x240</td>
</tr>
</tbody>
</table>

Figure 38. State Log Window With EnergyTrace+ Data

8.2.2 State Log Summary

The State Log Summary window shows a condensed view of the CPU activity of a profiled program (see Figure 39). Click the column headers to sort the data.

<table>
<thead>
<tr>
<th>Source</th>
<th>Count</th>
<th>First Time</th>
<th>Total (Time)</th>
<th>Total (%)</th>
<th>Shortest</th>
<th>Longest</th>
<th>Min Interval</th>
<th>Max Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU LPMx</td>
<td>7</td>
<td>213844.00</td>
<td>67520.00</td>
<td>12.95</td>
<td>76520.00</td>
<td>958864.00</td>
<td>1530801.00</td>
<td>718474.00</td>
</tr>
<tr>
<td>CPU Active Mode</td>
<td>8</td>
<td>0.00</td>
<td>75947.00</td>
<td>0.07</td>
<td>210344.00</td>
<td>933904.00</td>
<td>933904.00</td>
<td>933904.00</td>
</tr>
</tbody>
</table>

Current time: 20s 522134900 us

Figure 39. State Log Summary With EnergyTrace+ Data
8.2.3 Power Log Setup

The Power Log Setup can be used to control the analog measurement (see Figure 40). Check each parameter to enable data collection.

![Figure 40. Power Log Setup Window](image)

8.2.4 Power Log Summary

Similar to the State Log window, the Power Log window shows the current, voltage, and energy profile over time, with reference to the program counter that was sampled at the given time stamp (see Figure 41).

![Figure 41. Power Log Window With EnergyTrace+ Data](image)
8.2.5 Timeline

When invoking the Timeline for the first time, both Power Log and State graphs are disabled. Right click each section to enable it, and use the mouse wheel to zoom in and out (see Figure 42 and Figure 43).

Figure 42. Timeline With Power Log and State Graphs Disabled

Figure 43. Timeline With EnergyTrace+ Data

8.3 Measuring Low-Power Currents

During the capture of the internal states or even when simply executing until breakpoint halt, the target microcontroller is constantly accessed by the JTAG or SWD debug logic. These debug accesses consume energy that is included in the numbers shown in the Power Log window and graph. To measure the absolute power consumption of the application, TI recommends using the EnergyTrace mode in combination with the Release JTAG on Go option. This combination makes sure that the debug logic of the target microcontroller is not accessed while measuring energy consumption.

See Section 4.2.2 for more details.
9 Frequently Asked Questions

Q: My project does not build or reports errors about missing files. What could be wrong?

A: Make sure you have selected the correct device and that your project options are correct. Open the “C/C++ Compiler options”. Verify in the preprocessor tab that the following additional include directories are added:

$TOOLKIT_DIR$\inc\TexasInstruments
$TOOLKIT_DIR$\CMSIS\Include

Also in Preprocessor tab, verify that a symbol is defined for the device and ewarm, for example:

__MSP432P4111__

ewarm

The main.c file should include msp.h:

#include “msp.h”

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

A: Check the following:

• Is the JTAG switch (S101) in the correct orientation?
  Switch to left for XDS110-ET onboard debugger
  Switch to the right for external debugger connection

• 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 44 shows how to configure the debugger to use SWD instead of JTAG by opening the debugger settings window.
• 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 backchannel UART on the MSP432 LaunchPad development 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 development kit at specific baud rates, resulting in the software not being able to open the virtual COM port or in the baud rate being configured incorrectly. An issue with the LaunchPad 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 development kit into a USB3.0 Port**

A: When the MSP432 LaunchPad development kit is connected to USB3.0 ports provided by a certain combination of USB3.0 host controller hardware and device drivers, the IDE cannot establish a debug session with the LaunchPad development kit, resulting in an error message similar to “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` also cannot establish a connection with the LaunchPad development kit.

This issue has been 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 was resolved. Other USB3.0 hardware and device driver combinations might lead to the same issue. If you might be affected, contact the PC vendor or locate and install more recent versions of the USB3.0 device drivers. Alternatively, connect the LaunchPad development kit to an USB2.0 port on the PC, if one is available.

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

A: Check the following:

- Do the baud rate in the host 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: 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>dbgtag.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`.

---

**10 Additional IAR EWARM Information**

For more information about IAR Embedded Workbench, visit the following links:

- IAR Support
- IAR User's Guides for IAR Embedded Workbench for Arm
- IAR Embedded Workbench
- IAR Embedded Workbench Product News

**11 References**

2. SimpleLink MSP432 SDK
3. Debuggers for MSP432 Microcontrollers
4. Migration Guide for SimpleLink MSP432 SDK
5. CMSIS Pack Documentation
6. MSP432P4xx CMSIS Device Family Pack
7. XDS110 EnergyTrace™ High Dynamic Range (ETHDR) debug probe add-on
8. 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.

<table>
<thead>
<tr>
<th>Changes from February 15, 2018 to June 25, 2018</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Added the note that begins &quot;The following description is for the CMSIS Manager in IAR EWARM 8.30&quot; in Section 3, Creating a Project From MSP432 CMSIS Device Family Pack (DFP)</td>
<td>7</td>
</tr>
<tr>
<td>• Changed the steps to create an example in Section 3, Creating a Project From MSP432 CMSIS Device Family Pack (DFP)</td>
<td>7</td>
</tr>
<tr>
<td>• Added the note that begins &quot;TI XDS110 debug probes are enabled for use...&quot; in Section 4.1, Using TI XDS100, XDS110, and XDS200 Debug Probes</td>
<td>9</td>
</tr>
</tbody>
</table>