Signal Processing With a SimpleLink™ MSP432™ Microcontroller and the CMSIS-DSP Library

ABSTRACT

This application report describes the performance of the digital signal processing (DSP) CPU extension of the SimpleLink™ MSP432P401R microcontroller (MCU) leveraging the standard CMSIS DSP library in FFT operations. This is a software implementation that demonstrates several high-performance analog and digital integrations through various operations including CPU and DSP (visualization, vector processing), DMA, Precision ADC, and high-speed SPI (color LCD control) with a high-degree of parallel operations.

The example project boostxl_edumkii_microphonefft is in TI Resource Explorer under Software → SimpleLink MSP432P4 SDK → Examples → Development Tools → MSP432P401R LaunchPad → Demos → boostxl_edumkii_microphonefft_msp432p401r. You can import the project directly from this location.

Contents

List of Figures

Trademarks

SimpleLink, MSP432, LaunchPad, BoosterPack, MSP430, Code Composer Studio are trademarks of Texas Instruments.
Arm, Cortex, Thumb are registered trademarks of Arm Limited.
IAR Embedded Workbench is a registered trademark of IAR Systems.

1 Introduction

Features of this application include:

- Real-time FFT using accelerated hardware
- Graphical visualization on a color LCD with high-speed serial communication

2 System Description

The Fast Fourier Transform (FFT) algorithm converts a time-domain signal to frequency-domain (or the reverse) when FFT is manually used in many applications in signal processing. The Arm® Cortex®-M4F DSP extension, in conjunction with the CMSIS DSP Library, can efficiently perform vector-based operations such as FFT, FIR, matrix multiplications, and more. The software example in this application report demonstrates the clock cycles required to execute a real-time 512-point FFT using CMSIS DSP Library. This application report also demonstrates the high-performance features of the MSP432™ MCU including the 48-MHz Cortex-M4F CPU and the high-speed SPI communication through graphical data manipulation and FFT data visualization.

Figure 1 shows a high-level overview of the system function.

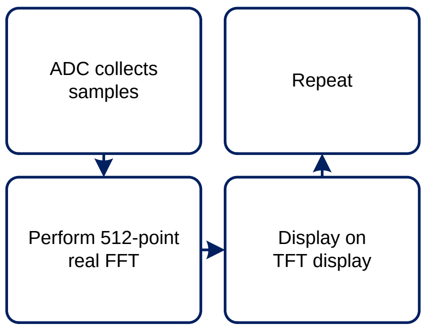


Figure 1. High-Level Overview

To get started with this application, the user is required to use the MSP-EXP432P401R LaunchPad™ development kit (see Section 5.1) and the BOOSTXL-EDUMKII BoosterPack™ plug-in module (see Section 5.2). Section 5 describes how to get started with the Educational BoosterPack MKII with the EXP-MSP432P401R LaunchPad development kit.

2.1 MSP432P401R

The MSP432P401R MCU is a 48-MHz 32-bit Cortex-M4F ultra-low-power MCU with 256KB of flash and 64KB of SRAM. This MCU integrates various digital and analog peripherals, many of which are used by this application, including the precision 1-Msps 14-bit ADC, the 48-MHz performance, the DSP extension of the Cortex-M4F CPU, the high-performance DMA, and the high-speed SPI serial communication.

Overall, the MSP432P401x is an ideal combination of the TI MSP430™ low-power DNA, advance mixed-signal features, and the processing capabilities of the 32-bit Cortex-M4 RISC engine. The devices ship with bundled driver libraries and are compatible with standardized components of the Arm ecosystem.

To find out more about this MCU, visit the MSP432P401R product folder.

2.2 Cortex-M4F With DSP Extension

The MSP432 Cortex-M4F processor features a single-cycle multiply-accumulate (MAC) unit, optimized single instruction multiple data (SIMD) instructions, saturating arithmetic instructions, and a single-precision floating-point unit (FPU). These digital signal control features build on the innovative technology that characterizes the Cortex-M family of processors. These features include a 32-bit core capable of 1.25 DMIPS/MHz for high performance, Thumb®-2 instructions for optimum code density and a Nested Vector Interrupt Controller (NVIC) for outstanding interrupt handling.

The CMSIS DSP library provides a suite of common signal processing functions for use on devices based on the Cortex-M processor. The implementation on Cortex-M4F fully leverages the integrated DSP instructions to provide robust vector math operations including:

- Basic math functions
- Fast math functions
- Complex math functions
- Filters
- Matrix functions
- Transforms
- Motor control functions
- Statistical functions
- Support functions
- Interpolation functions

For more information, see the CMSIS DSP Software Library at http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html.

2.3 Precision ADC

The MSP432P401R microcontroller has an on-chip precision 1-Msps 14-bit SAR ADC that is used to continuously sample the microphone output at 8 kHz. The microphone has a front-end operational amplifier. The output of the op amp is connected to ADC channel 10 (P4.3). To ensure a deterministic sampling rate, the ADC is automatically triggered by Timer_A0 channel CCR1.

2.4 µDMA

When the ADC sample is completed, the data is automatically moved to one of the two SRAM buffers using DMA channel 7. The µDMA module on the MSP432 MCUs acts as a separate master on the bus and does not interfere or stall the CPU access. The µDMA modules as a master enables parallel operations with the CPU executing code while the ADC sampling and DMA transfers occur.

3 Block Diagram

Figure 2 shows the block diagram of the application.

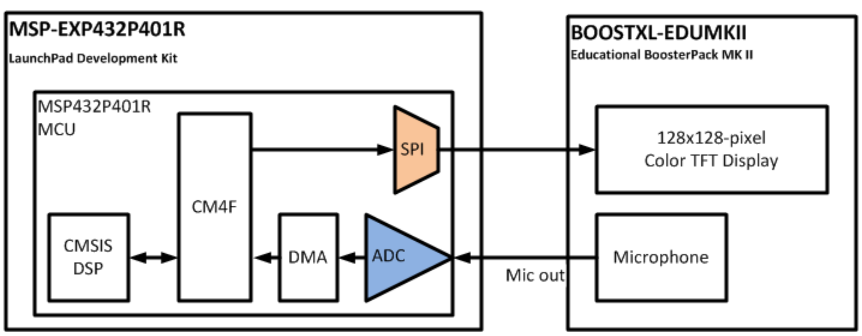


Figure 2. Block Diagram

4 System Design

This application uses the DSP extension of the Cortex-M4F CPU and the CMSIS-DSP library to perform real-time FFT with audio sampled using the onboard Precision ADC from microphone on the Educational BoosterPack MK II.

To ensure a real-time performance with no sample loss during processing, this application implements dual-buffered memory to store the ADC samples. When a buffer is full, the application starts processing the buffer while the application continues to sample the ADC in the background into the secondary buffer. A ping-pong scheme is set up for the µDMA, where the primary and alternative channel controls are configured to fill their respective 512-sample-deep data buffer. When the primary data buffer is filled, the DMA switches to the alternative data buffer while FFT processing on the primary data buffer begins in parallel.

Figure 3 shows the application flow chart.

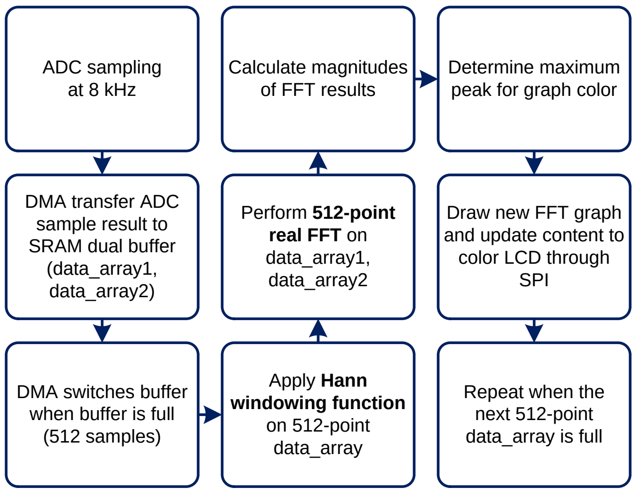


Figure 3. Application Flow Chart

4.1 *Real-Time 512-Point Real FFT*

The MSP432 MCU combined with the CMSIS-DSP Library supports up to 8192-point real and complex FFTs. This application demonstrates the performance of a 512-point real FFT. This application collects up to 512 ADC samples from the onboard microphone before starting the FFT routine. To ensure real-time FFT performance, this application implements a dual-buffered array where one buffer is always being filled, while the other is processed. The audio is sampled at a rate of 8 kHz using onboard intelligent peripherals such as the timer that automatically triggers the ADC to sample. The advanced µDMA moves the data to memory completely independent of CPU operations. A Hann window function is applied on the sampled data to further reduce the noise from the microphone input. The CMSIS-DSP API arm_rfft_q15() then performs the real FFT where the FFT output contains real values. The magnitude is then calculated and contains 256-point positive and negative frequencies. Because the display is only 128 pixels wide, each two adjacent frequency bins are combined, averaged, and displayed as one bar on the output graph.

Figure 4 shows the timing diagram.

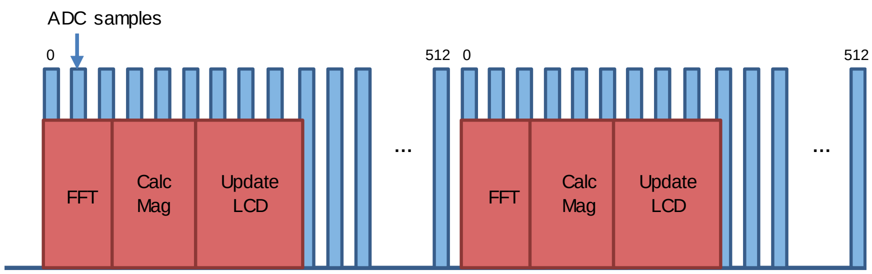


Figure 4. Timing Diagram

5 Getting Started Hardware

Figure 5 shows that this application uses the MSP-EXP432P401R LaunchPad development kit with the BOOSTXL-EDUMKII Educational BoosterPack plug-in module.

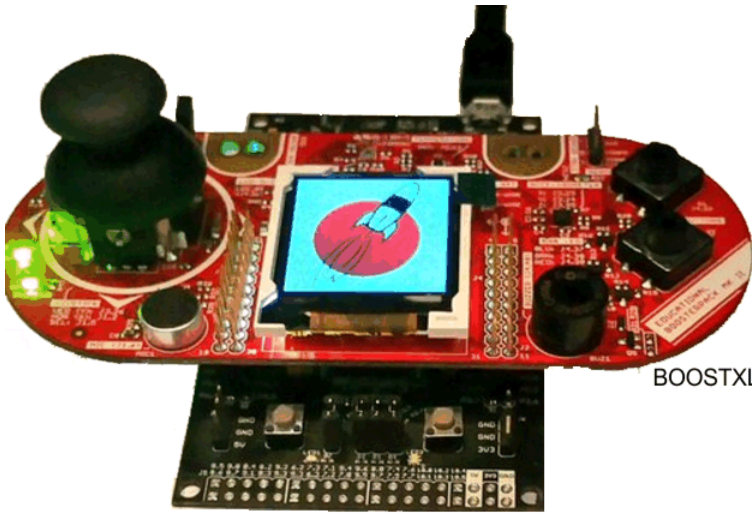


Figure 5. Stacked LaunchPad Kit and BoosterPack Module

To power the LaunchPad development kit, connect the Micro-USB connector.

5.1 MSP-EXP432P401R LaunchPad Development Kit

The MSP-EXP432P401R LaunchPad development kit is based on the MSP432P401R microcontroller featuring a 48-MHz Cortex-M4F CPU and a wide variety of MSP analog and digital peripherals including the Precision ADC, high-speed communication, and enhanced security features. The MSP-EXP432P401R LaunchPad development kit integrates an onboard XDS110 debugger capable of programming and debugging the MSP432P401R MCU through JTAG, SWD, and BSL. The MSP-EXP432P401R LaunchPad development kit includes a standardized 40-pin header to enable interfacing with a wide variety of BoosterPack plug-in modules.

To learn more about this LaunchPad development kit, visit the MSP-EXP432P401R tool folder.

5.2 BOOSTXL-EDUMKII Educational BoosterPack MK II

The BOOSTXL-EDUMKII Educational BoosterPack MK II offers a high level of integration for developers to quickly prototype complete solutions. Various analog and digital inputs and outputs are at your disposal including an analog joystick, environmental and motion sensors, RGB LED, microphone, buzzer, color LCD display, and more.

Figure 6 shows the modifications that this application requires to the BOOSTXL-AUDIO BoosterPack plug-in module.

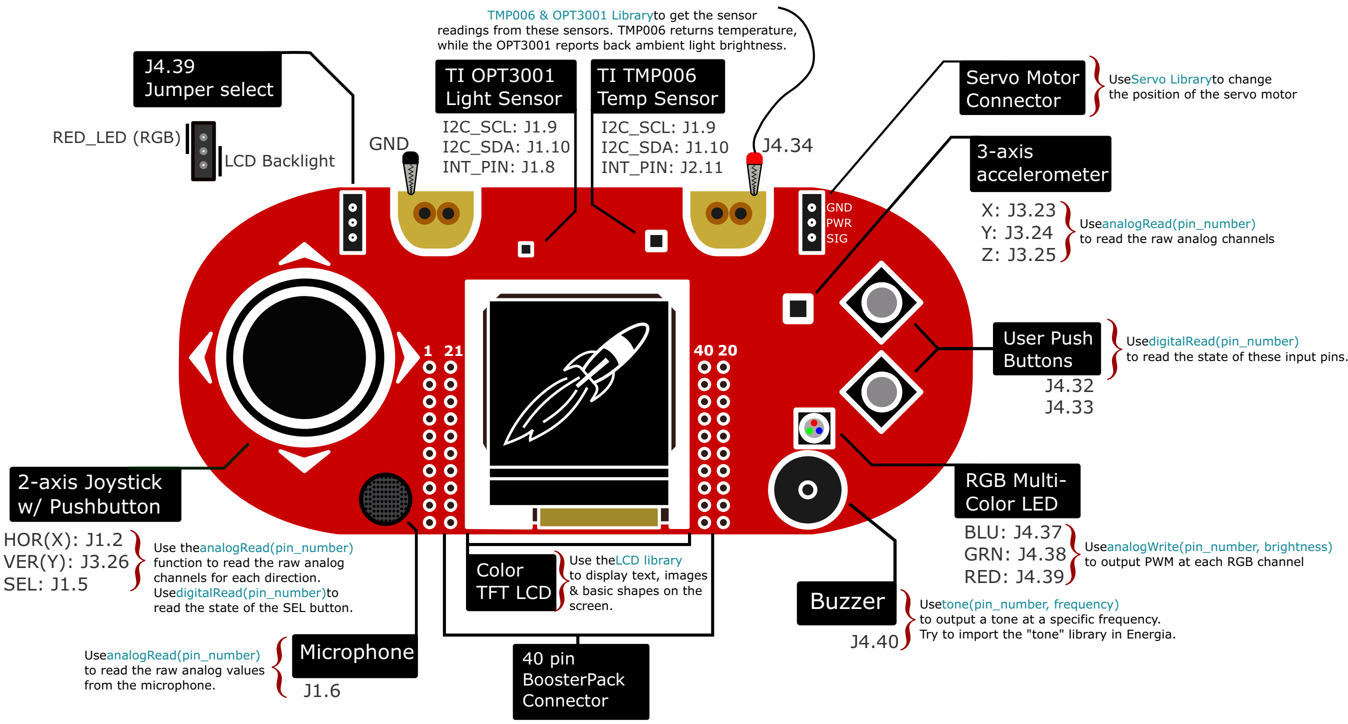


Figure 6. BOOSTXL-AUDIO BoosterPack Required Modification

For more information, visit the Educational BoosterPack MK II tool folder.

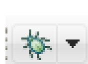
6 Getting Started Firmware

To get started, download the SimpleLink MSP432 SDK from http://www.ti.com/tool/download/SIMPLELINK-MSP432-SDK. After the SDK is installed, perform the follow steps to import the projects into Code Composer Studio™ IDE from the TI Cloud Resource Explorer.

6.1 Code Composer Studio™ (CCS) IDE

To get started with CCS:

1. Launch CCS.
2. Navigate to Project → Import CCS Projects.
3. Click Browse.
4. Navigate to the folder <SimpleLink SDK path>/boostxl_edumkii_microphonefft_msp432p401r/No RTOS/CCS Compiler.

   The default location for the current SimpleLink MSP432P4 SDK path is: c:\ti\simplelink_msp432p4_sdk_<version#>
5. Click Finish.
6. Click the debug button  to download and debug the project.

You can also import the example project boostxl_edumkii_microphonefft from the CCS TI Explorer view. Navigate to Software → SimpleLink MSP432P4 SDK → Examples → Development Tools → MSP432P401R LaunchPad → Demos → boostxl_edumkii_microphonefft_msp432p401r (see Figure 7) and directly import the project from this location.

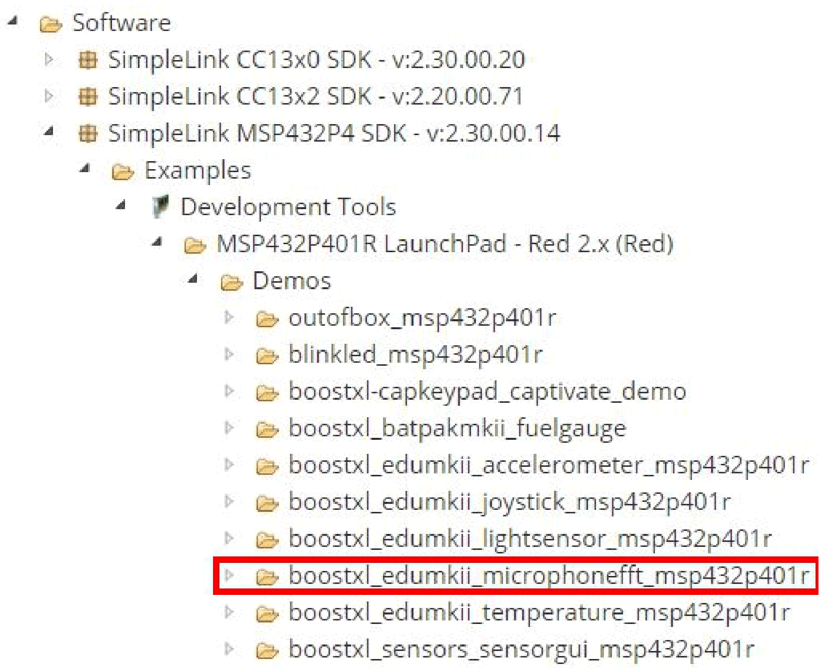


Figure 7. Importing Project From the SimpleLink MSP432 SDK in TI Resource Explorer

6.2 IAR Embedded Workbench® (IAR) IDE

To get started with IAR:

1. Select File → Open Workspace, and navigate to the project folder for the BOOSTXL-EDUMKII Microphone FFT project in the folder <SimpleLink MSP432P4 path>\examples\nortos\MSP_EXP432P401R\demos\boostxl_edumkii_microphonefft_msp432p401r\iar.

   The default location for the current SimpleLink MSP432P4 SDK path is: c:\ti\simplelink_msp432p4_sdk_<version#>
2. Double-click on BOOSTXL-EDUMKII_MicrophoneFFT_MSP432P401R.eww to open the IAR workspace project.
3. Click Project → Make to build the project.
4. Make sure that the Micro-USB is plugged in first to download and debug the application.
5. Click Project → Download and Debug.

7 Application Execution Performance

The signal-processing performance includes the clock cycles required to execute each FFT routine. The number of cycles required to perform a 512-point real FFT is approximately 25392 cycles, and approximately 530 µs at 48 MHz (IAR).

The system was designed to minimize the real-time impact due to multiple concurrent resources by leveraging CPU and DMA operations in parallel. DMA and CPU requests and activities can occur concurrently with minimal stalling or wait-states incurred.

The number of cycles required to perform a 512-point complex FFT is approximately 30489 cycles, and takes approximately 635.2 µs at 48 MHz with flash configured with one read wait state (using CCS). With further optimization and benchmarked at zero flash wait states, the 512-point real FFT function alone only takes 25392 cycles (IAR, maximum nondebug optimization settings). Figure 8 shows detailed cycle duration for the entire application. Major operations include: Hann window function on sampled data, FFT calculation, magnitude calculation, maximum peak identification, graph creation, and color LCD screen update.

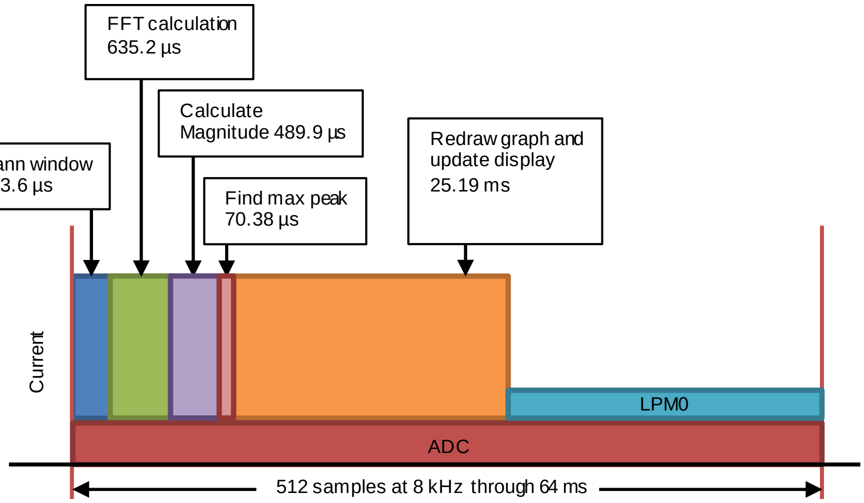


Figure 8. FFT and LCD Operation Per Update Execution Timeline

Table 1 lists the main application operations and their durations.

Table 1. Application Operations and Duration (48 MHz With One Wait State)

FUNCTION	DURATION (cycles)	DURATION (ms)
Hann window initialization	323952	6.749
Hann window function	10253	0.2136
FFT initialization and processing	30490	0.6352
Magnitude calculation	23515	0.4899
Maximum peak identification	3378	0.07038
Redraw graph and update LCD	1209120	25.19

When the current is measured, the educational BoosterPack module is removed from the LaunchPad kit, because the backlight from the LCD on the BoosterPack module significantly contributes to the current on the system and dominates the system current. Figure 9 shows the current periodic interval of the system measuring the ADC while DMA moves the data and calculates the FFT.

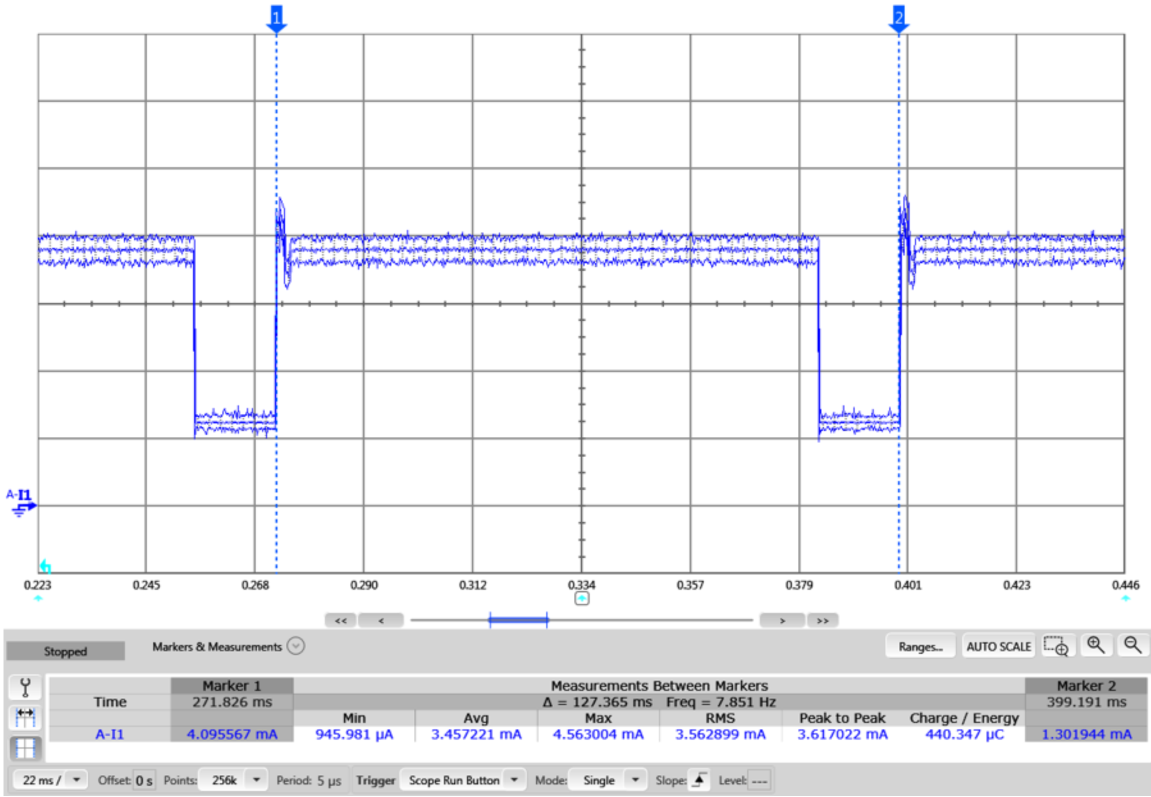


Figure 9. FFT Application Current Profile

8 Software Files

The example project boostxl_edumkii_microphonefft is in TI Resource Explorer under Software → SimpleLink MSP432P4 SDK → Examples → Development Tools → MSP432P401R LaunchPad → Demos → boostxl_edumkii_microphonefft_msp432p401r. You can import the project directly from this location.

9 References

1. TI Resource Explorer
2. CMSIS DSP Software Library
3. *MSP432P401R, MSP432P401M SimpleLink™ Mixed-Signal Microcontrollers*
4. *MSP432P4xx SimpleLink™ Microcontrollers Technical Reference Manual*
5. SimpleLink™ MSP432P401R High-Precision ADC LaunchPad™ Development Kit tool page

Revision History

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

Changes from May 31, 2016 to October 30, 2018 **Page**

- Changed the document title to add "SimpleLink" 1
- Changed the location of the example project in the abstract 1
- Changed "14-Bit ADC" to "Precision ADC" for consistency with other documents 4
- Changed the description for downloading the example project in Section 6, *Getting Started Firmware* 7
- Changed the location of the example project in Section 6.1, *Code Composer Studio™ (CCS) IDE* 7
- Changed the paragraph that begins "You can also import the example project..." in Section 6.1, *Code Composer Studio™ (CCS) IDE* 7
- Changed the location of the example project in Section 6.2, *IAR Embedded Workbench® (IAR) IDE* 8
- Changed the location of the example project in Section 8, *Software Files* 10
- Updated Section 9, *References* 10