Application Report
SLAA735A – June 2017 – Revised September 2017

Leveraging Precision ADC on SimpleLink™ MSP432™ Microcontrollers for Predictive Maintenance

ABSTRACT

This application report describes a high-performance method for predictive maintenance of motors by monitoring their frequency response with the SimpleLink™ MSP432P401R microcontroller using precision ADC with up to 16-bit ENOB and FFT analysis.

The software described in this application report is available as part of SimpleLink™ MSP432™ Software Development Kit (SDK).

Contents

List of Figures

List of Tables

Trademarks

SimpleLink, MSP432, Code Composer Studio, LaunchPad are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Introduction

Predictive maintenance is a growing trend in a host of industrial applications. Predictive maintenance allows users and operators of industrial equipment insight into failures before they happen. This enables early preventive and corrective actions to be taken to reduce the cost of the failure incurred due to costly repairs or potential replacement of industrial equipment. One specific application is in the field of motor vibration analysis.

While electronic faults may be easy to identify and repair, mechanical faults in moving parts of a motor may take time to be observable. A motor construction consists of four major elements (see Figure 1).

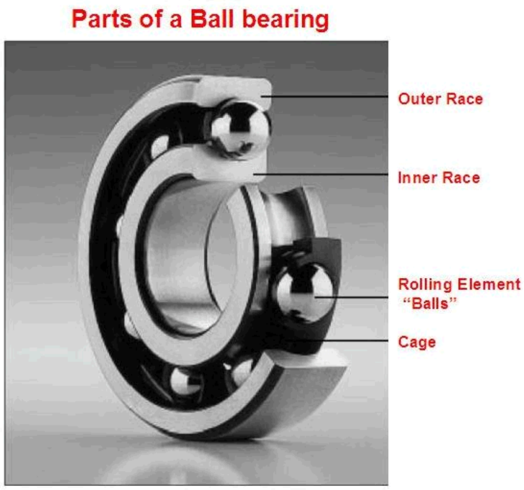

Figure 1. Motor Construction

1. Inner race: The inner surface of the moving element, which is typically mounted on the shaft.
2. Outer race: The outer surface of the motor, which is a static component.
3. Cage: The housing that holds the ball bearings.
4. Balls: The moving element between the inner and the outer races

The mechanical faults are primarily related to the moving elements of a motor. These faults are also referred to as defect frequencies and occur due to repeated and consistent impact between two surfaces. The defect frequencies are determined by the geometry of the bearings, number of bearings, the load on the shaft, and the rotational speed of the shaft. These defect frequencies are well documented and are categorized as:

1. FTF (Fundamental Train Frequency): The rotational speed of the bearing cage.
2. BPFI (Ball Pass Frequency of the Inner race): The frequency at which the balls contact a single point on the inner race during one revolution of the inner ring.
3. BPFO (Ball Pass Frequency of the Outer race): The frequency at which the balls contact a single point on the outer race during one revolution of the inner ring.
4. BSF (Ball Spin Frequency): The rotational speed of the balls on their own axes.

Without getting into the details of the equations used to compute these parameters, the parameters must be documented by the bearing manufacturer (called the Bearing ID) and are categorized in terms of the shaft frequency. For example, if the BPFO of a particular bearing is 9.740 and the shaft rotation is 1000 rpm (rotations per minute), then the FTF appears at 9740 cycles per minute (cpm), or 162.33 Hz in the frequency spectrum. Figure 2 shows a representation of BPFO and BPFI as seen in the frequency domain. The figure also shows the FFT of a normally functioning bearing compared to a faulty bearing.

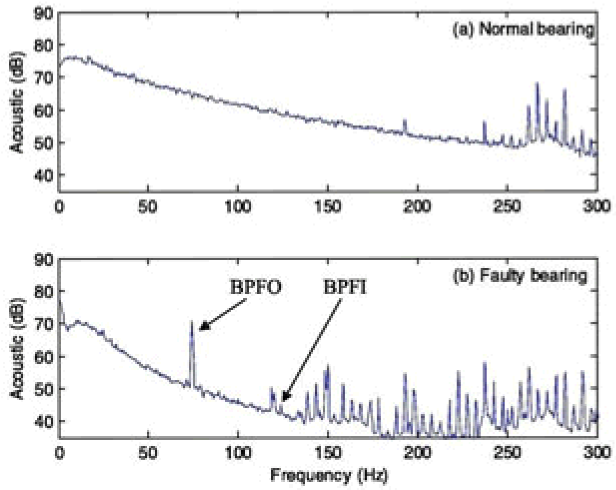

Figure 2. BPFO and BPFI of a Normal Bearing Compared to a Faulty Bearing

Another important aspect of predictive maintenance is to look at trends of shock pulse, which increase within the ultrasonic frequency range. Over a period of time, as the bearings gradually wear, the energy distribution begins to show at the defect frequency. So, for critical equipments, monitoring the ultrasonic region is also an important aspect of predictive maintenance of motors.

2 System Description

This application report uses the low-power MSP432 MCU, which features a 14-bit ADC to perform high-accuracy data acquisition and a configurable DMA to not only read the data at high rates but also to reconfigure the system dynamically to perform data acquisition at different data rates. With the 14-bit ADC, it is possible to design an extremely low-power yet sensitive and high-performance system that can perform a wide range of vibration data analysis over the operational time of a motor without having to invest in expensive maintenance infrastructure.

2.1 Hardware Architecture

As the distribution of the defect frequency can be in different parts of the frequency spectrum, it is important to process the signal in the appropriate parts of the frequency spectrum. Figure 3 shows the system configuration used for this application report. The input may be one or more sensors or transducers; for example, a microphone for audio band or a piezoelectric vibration sensor. The output of these devices must be then passed through different analog filters so that the correct range of frequencies is then captured by the precision ADC in the MSP432P401R microcontroller.

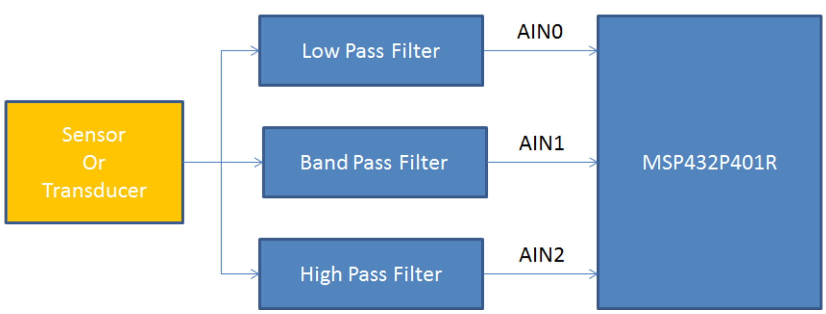

Figure 3. Hardware Architecture

For illustration purpose, the following filters are suggested.

1. Low-pass filter: The low-pass filter is used to sample signals less than 256 Hz, which is typically for the FTF. Hence, a 3rd-order Chebyshev low-pass filter with a cutoff frequency of 250 Hz and a ripple of 0.1 dB is used. Figure 4 shows the magnitude response of the filter.

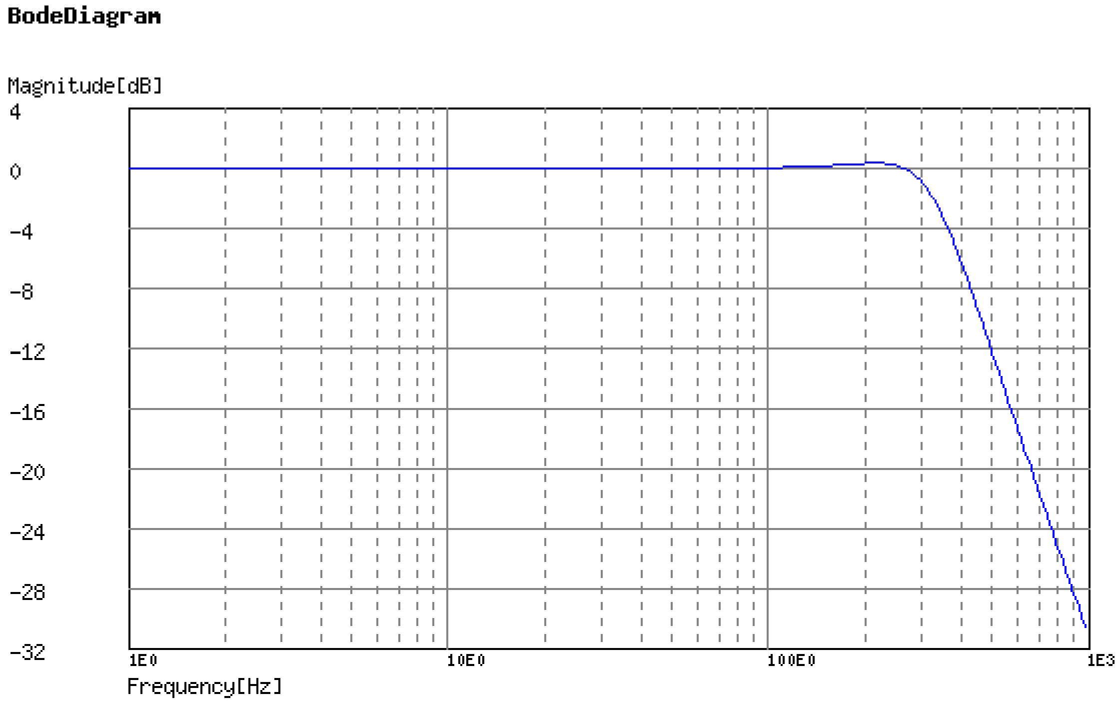

Figure 4. Magnitude Response of Low-Pass Filter

2. Band-pass filter: The band-pass filter is used to sample signals more than 1 kHz and less than 4 kHz. Hence, a multiple feedback band-pass filter with a resonant frequency of 2.2 kHz, a gain of –1 at resonant frequency, and a quality factor of 4 is used. Figure 5 shows the magnitude response of the filter.

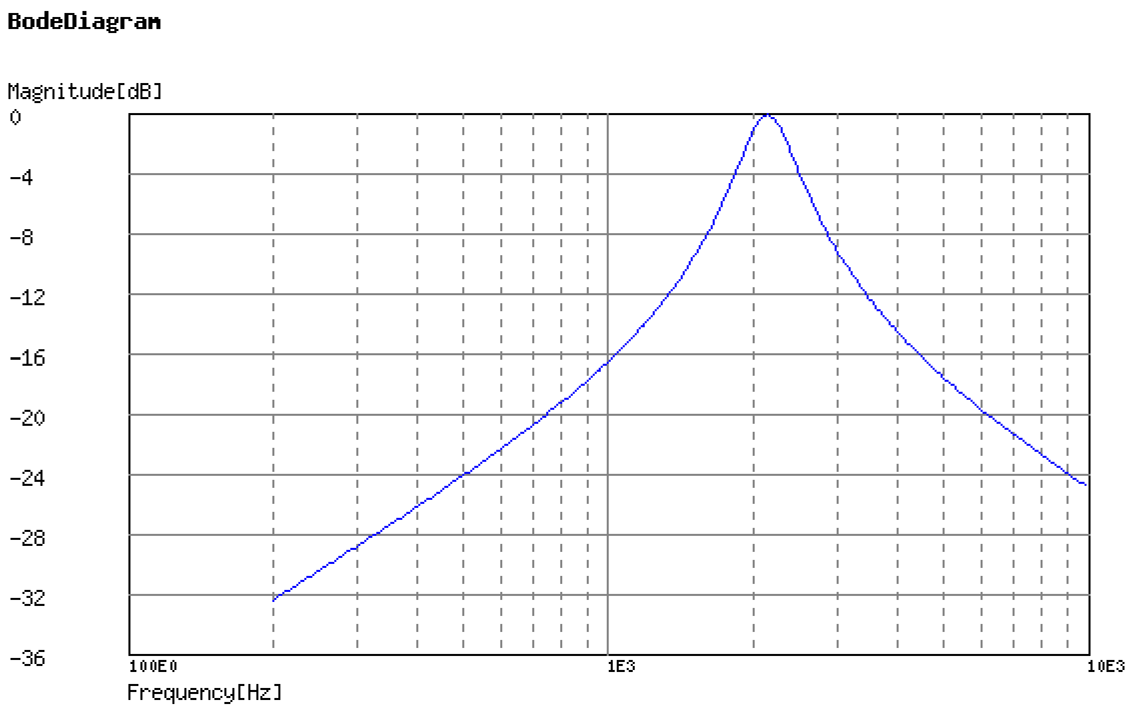

Figure 5. Magnitude Response of Band-Pass Filter

3. High-pass filter: The high-pass filter is used to sample signals more than 20 kHz. Hence a 3rd-order Chebyshev high-pass filter with a cutoff frequency of 20 kHz and a ripple of 0.1 dB is used. Figure 6 shows the magnitude response of the filter.

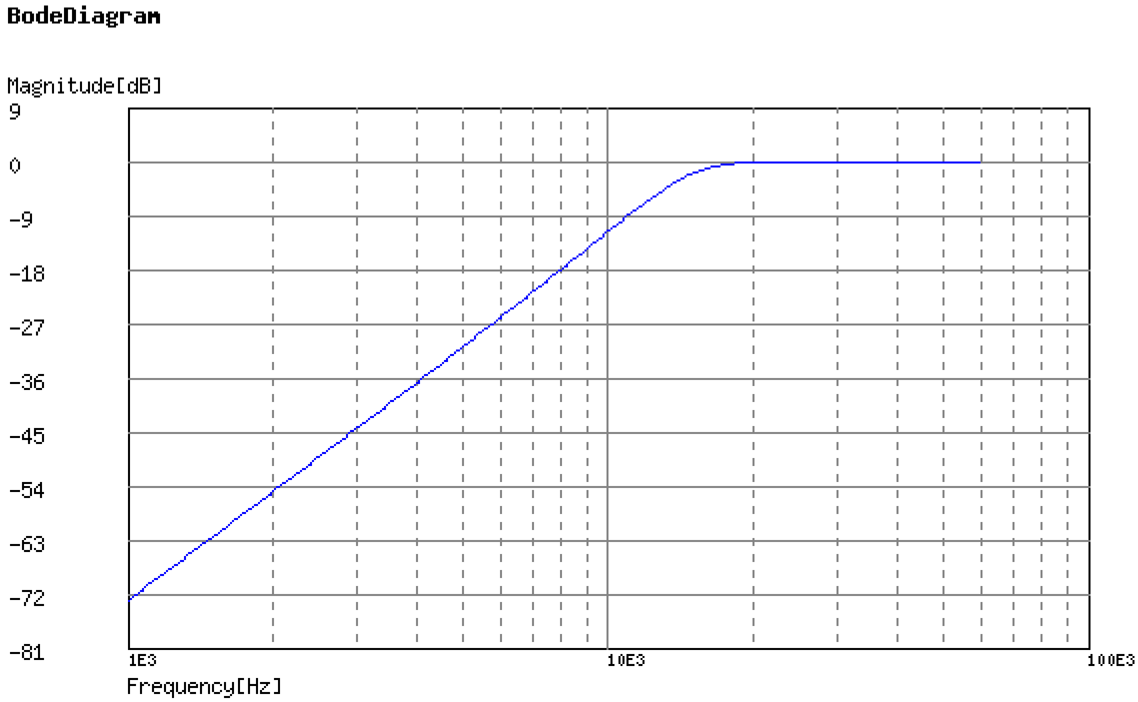

Figure 6. Magnitude Response of High-Pass Filter

2.2 Software Architecture

Figure 7 shows the software architecture of the application. The software can be downloaded from http://www.ti.com/lit/zip/slaa735.

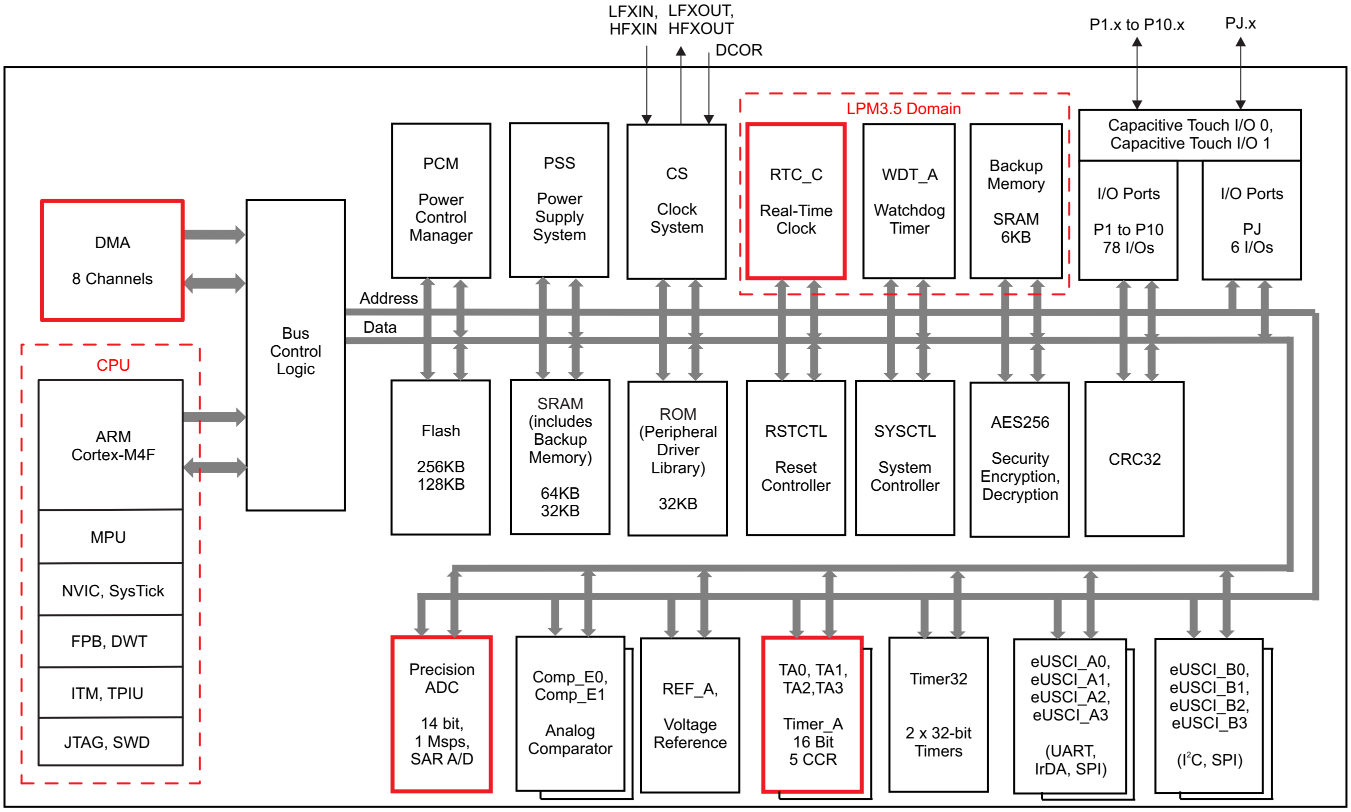

Figure 7. Software Architecture

The four main components of the software are:

- Application component
- RTC component
- ADC, DMA, and TimerA component
- DSP component

2.2.1 Application Component

The main application routine configures the device using DriverLib API calls for the different peripherals and manages the low-power transition of the device during idle period to reduce power consumption. This routine also performs the three FFT conversions.

1. The first FFT is the low-frequency 512-point FFT. This FFT is performed on the data being sampled at a sampling rate of 512 Hz on ADC channel 0. As a result, it can be used to capture and compute the energy distribution of the signals up to 256 Hz. With a sampling frequency of 512 Hz and 512 samples, each bin after FFT conversion has a resolution of 1 Hz.
2. The second FFT is the audio-band frequency 1024-point FFT. This FFT is performed on the data being sampled at a sampling rate of 40 kHz on ADC channel 1. As a result, it can be used to capture and compute the energy distribution of the signals up to 20 kHz. With a sampling frequency of 40 kHz and 1024 samples, each bin after FFT conversion has a resolution of 39.0625 Hz.

3. The third FFT is the ultrasonic frequency 1024-point FFT. This FFT is performed on the data being sampled at a sampling rate of 400 kHz on ADC channel 1. As a result, it can be used to capture and compute the energy distribution of the signals up to 200 kHz. With a sampling frequency of 400 kHz and 1024 samples, each bin after FFT conversion has a resolution of 390.625 Hz.

The total time needed to perform the data sampling on the three channels and the processing the FFT is fixed and can be computed as follows.

$$\text{Total Time for Sampling and Acquisition } T_{SA} = 512 \times T_{512Hz} + 1024 \times T_{40kHz} + 1024 \times T_{400kHz} + T_{FFT512} + 2 \times T_{1024FFT}$$

where

$$T_{512Hz} = 1.953 \text{ ms}$$

$$T_{40kHz} = 25\ \mu s$$

$$T_{400kHz} = 2.5\ \mu s$$

$$T_{FFT512} = 1.736 \text{ ms}$$

$$T_{FFT1024} = 3.279 \text{ ms} \quad (1)$$

This equation results in the total time for running the process T_{SA} = 1.037 seconds.

The application performs the three FFT conversions once every 5 minutes. Because the FFT energy distribution due to size of bin may cause the bin energy distribution to change from one set of sampling to another, an averaging algorithm has been applied in the application. To achieve the averaging, when the sampling and FFT conversions is completed, the application restarts the process of sampling and FFT conversion. The define AVG_WINDOW in the application is 10, but may be modified by the user. The data collected from each of the frequency bins is then stored in a structure. Because both 512-point and 1024-point FFTs are used, two structures have been declared in the application. The structures are identical except for the number of energy bins based on the FFT (see Figure 8).

uint8_t	iteration
uint16_t	lowBin
uint16_t	highBin
uint16_t	monitorValue
uint16_t	alaramValue
uint32_t	energyValue[FFT POINT]

Figure 8. Structure for Storing FFT Energy Data

1. The *iteration* field is an unsigned 8-bit value to store the number of FFT iterations performed.
2. The *lowBin* field is an unsigned 16-bit value to store the index of the lower-energy bin where the application needs to monitor the energy.
3. The *highBin* field is an unsigned 16-bit value to store the index of the higher-energy bin in the FFT where the application needs to monitor the energy.
4. The *monitorValue* field is an unsigned 16-bit value that is used to check if the energy in the range of *lowBin* to *highBin* index has crossed a threshold that may require the user to be notified.
5. The *alarmValue* field is an unsigned 16-bit value that is used to check if the energy in the entire range of the FFT has crossed a threshold that may require the user to be notified of an impending failure.
6. The *energyValue* field is an unsigned 32-bit array that contains the sum of the energy calculation at every bin over the *iteration* count. This allows the application to average out the calculation to account for changes over the iterations of FFT computation.

2.2.2 RTC Component

As described in Section 2.2.1, the most prominent component of the T_{SA} is the time taken to acquire the 512 samples of the data at 512 Hz. To manage the power consumption during this time window, the first function of the RTC is to generate a wake-up signal using the RTC interval interrupt. The RTC wakes the CPU every 1.953 ms, and the CPU acquires the data from ADC channel 0 and returns to the LPM3.5 state. This allows the application to conserve power when performing the low-frequency data acquisition. When the low-frequency FFT data acquisition is completed, the interrupt source is disabled to allow the device to remain in LPM3.0 for a longer duration before the next set of sample data is to be collected.

The second function of the RTC is to manage the interval between these data acquisitions using the calendar mode and match feature. Under ideal conditions, when the motor vibrations are minimal during its early life, the device needs to wake up once every 5 minutes. However, when the energy in the bin of interest exceeds the programmed *monitorValue* or, if across its spectrum, there is a spike above the *alarmValue*, then the device needs to wake even more quickly. This is achieved by setting the RTC match value based on the data collected in the previous FFT to a 2-minute interval.

2.2.3 ADC, DMA, and TimerA Components

Although the application is now aware of the low-frequency energy component, it is also necessary to detect the audio range and ultrasonic range frequency energy component. To achieve this detection and yet be able to maintain a low system power, the application uses the flexible ADC, DMA, and TimerA of the MSP432 device. The TimerA is used to trigger the ADC data acquisition on analog channels 1 and 2, while the DMA is used to move the data to a buffer in SRAM. To further improve the system performance without keeping the device in active power state for a long time, the TimerA is programmed to generate triggers to the precision ADC at 400 kHz, and the DMA is programmed in scatter gather mode to:

- Copy the data from precision ADC when conversion is complete, based on the TimerA trigger.
- Disable the TimerA and precision ADC when the 1024 samples for the ultrasonic FFT analysis are captured.
- Reprogram the TimerA for 40-kHz trigger pulses for precision ADC.
- Reprogram the precision ADC to capture audio-band data on channel 1.
- Restart the precision ADC and TimerA.
- Copy the data from precision ADC when conversion is complete, based on the new TimerA trigger.
- When the 1024 samples for the audio-band FFT analysis is captured, disable the TimerA and precision ADC.
- Reprogram the precision ADC for low-frequency data capture and CPU trigger.

This sequence allows the DMA to offload the CPU, which in the same duration performs the low-frequency FFT conversion and data structure processing and enters LPM0 to reduce device current consumption.

2.2.4 CMSIS DSP

As the application requires the conversion of the time-domain samples to frequency domain, an FFT algorithm must be used. For this specific purpose, the CMSIS-DSP library was chosen as it provides the customer an existing and proven DSP library function.

2.2.5 Application Software Flow

Figure 9 show the overall application software flow which integrates the above mentioned components.

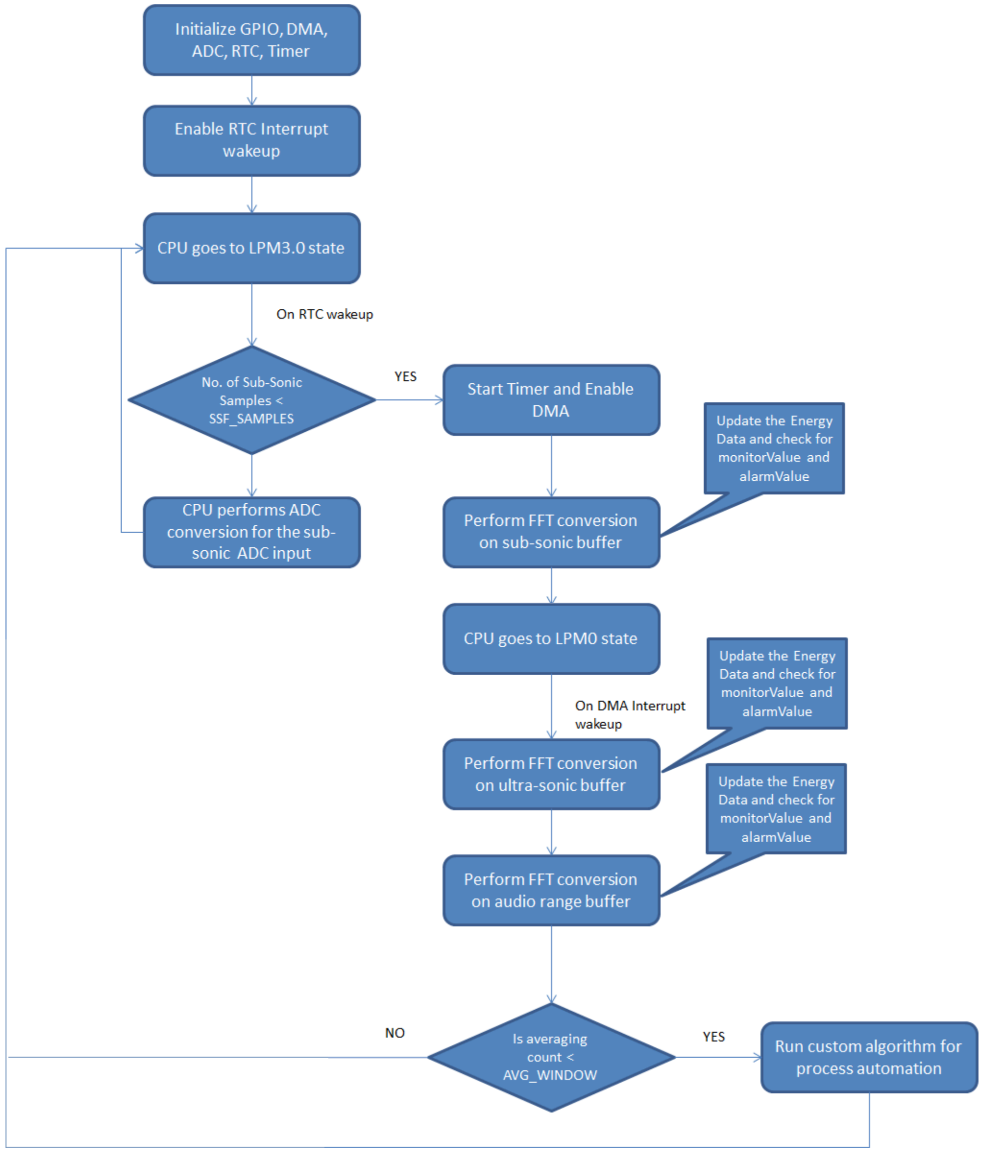

Figure 9. Application Software Flowchart

3 Test Setup and Results

To run this application report, import the example from SimpleLink MSP432 SDK version 1.50.00.12. Import the project **cmsis_dsplib_vibration_msp432p401r** from the path **examples\nortos\MSP_EXP432P401R\demos** under the SimpleLink MSP432 SDK into Code Composer Studio™ IDE version 7. Compile the example and download the firmware output file to the MSP-EXP432P401R LaunchPad.

This application report uses the MSP-EXP432P401R LaunchPad™ development kit along with an Arbitrary Waveform Generator (AWG). the AWG provides the input signal that is used to test the FFT result from the different frequency bands. When the energy detection in the different frequency spectrums either exceeds the *monitorValue* or *alarmValue* parameters then a corresponding LED on the board is lighted (see Table 1).

Table 1. Frequency Component LEDs

Frequency Component Detected	LED
Low-frequency component	Red
Audio-band frequency component	Green
Ultrasonic-frequency component	Blue

The following figures show the current profile for the application. Figure 10 shows the current for data capture and processing during the averaging window corresponding to T_{SA}.

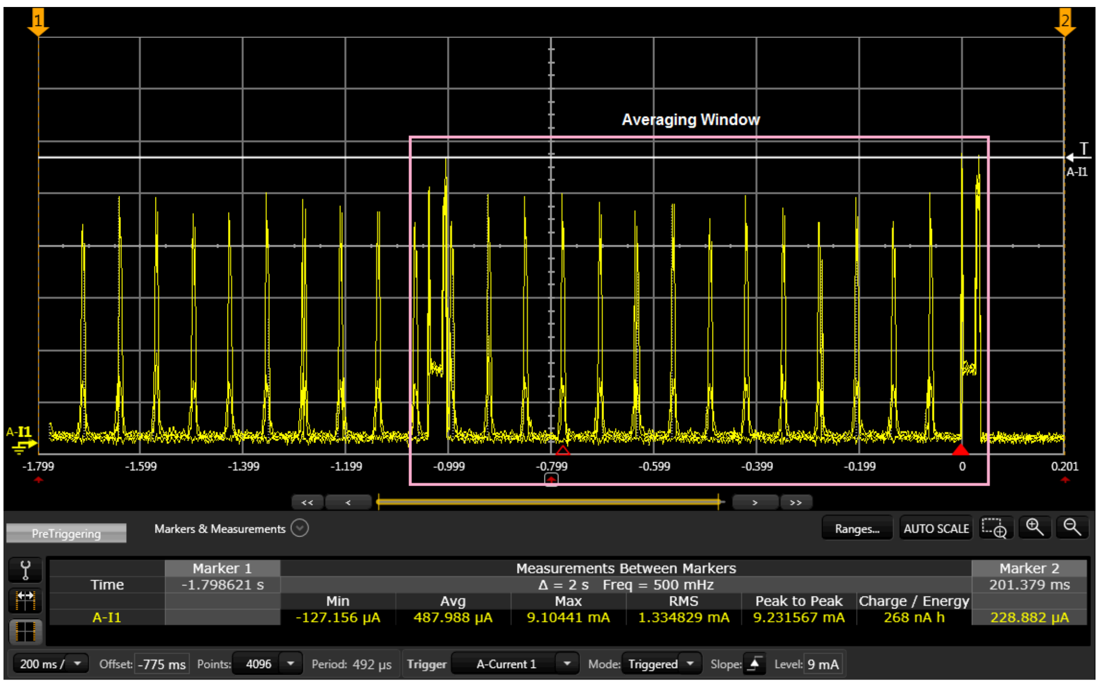

Figure 10. Typical Averaging Window

Figure 11 shows a zoomed in version of the processing window, which corresponds to the time frame $T_{SA} - 512 \times T_{512Hz}$. In this figure, the first set of peaks correspond to the FFT duration for the low-frequency sampler. The second flat current consumption range is when the device is acquiring the ultrasonic and audio-band samples with the device in LPM0 state to conserve power. The last set of peaks is when the ultrasonic and audio-band FFT is being performed. Subsequently, the device enters LPM3.0 state. The RMS current consumption during this window is 3.312 mA with a peak of 12.079 mA. In the LPM3.0 state, the device current consumption drops to the range of a few 10s of µA.

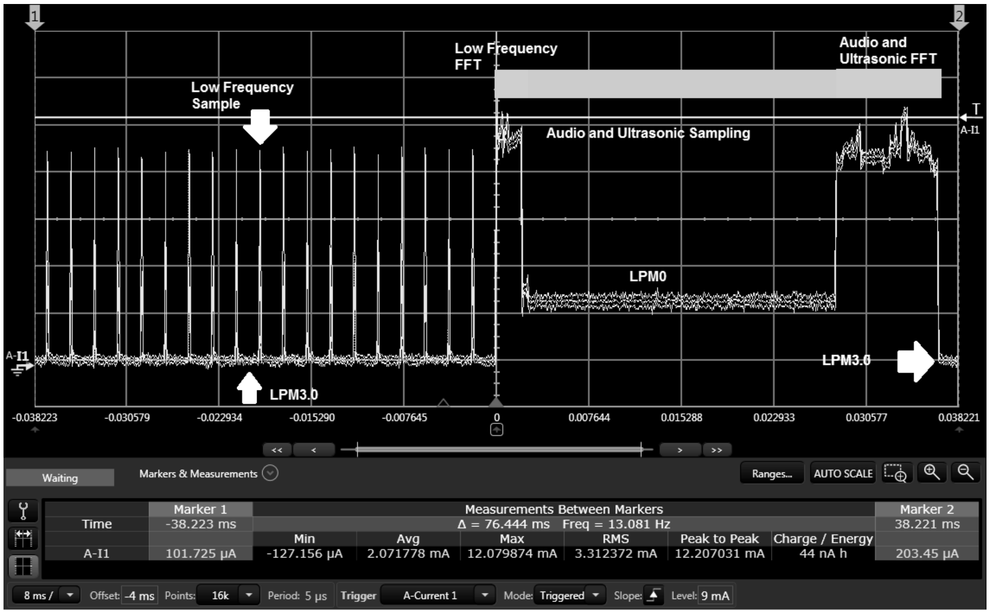

Figure 11. Compute Window

Figure 12 shows a further zoom in version of the same window in Figure 11 to compute the current consumption of the device in LPM0 state. During this time, the DMA, ADC, and TimerA module are active, resulting in 3.312-mA RMS current.

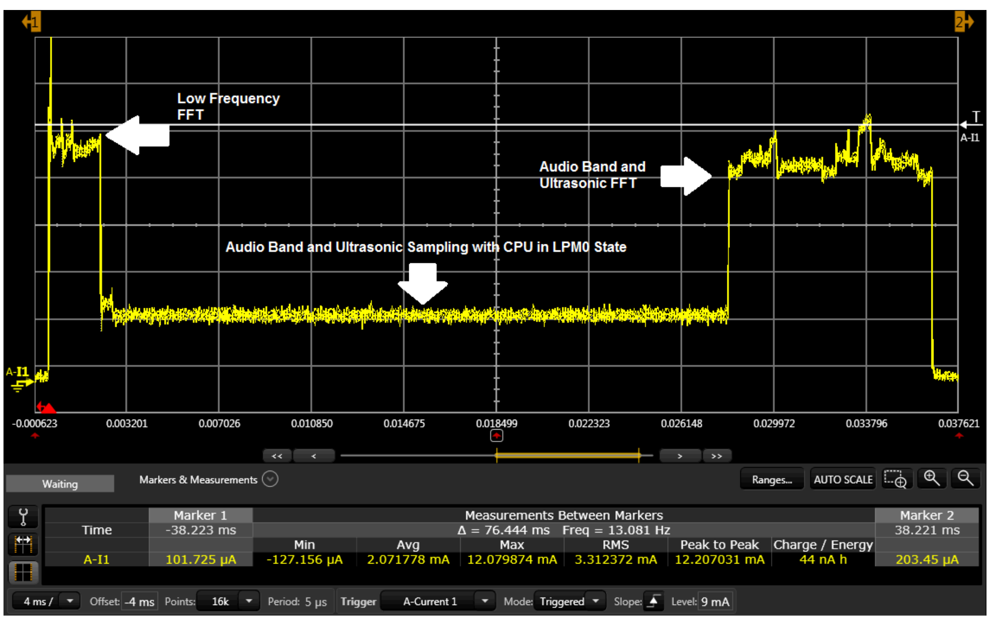

Figure 12. LPM0 State Current Consumption

The AWG is configured to generate an output sine wave of 1 V peak-to-peak with an offset of 600 mV with three different sine outputs. The subsonic band is tested with a sine wave of 100.0 Hz, the audio band is tested with a sine wave of 1.0 kHz, and the ultrasonic band is tested with a sine wave of 43.0 kHz. Figure 13, Figure 14, and Figure 15 show the FFT computation on the three components.

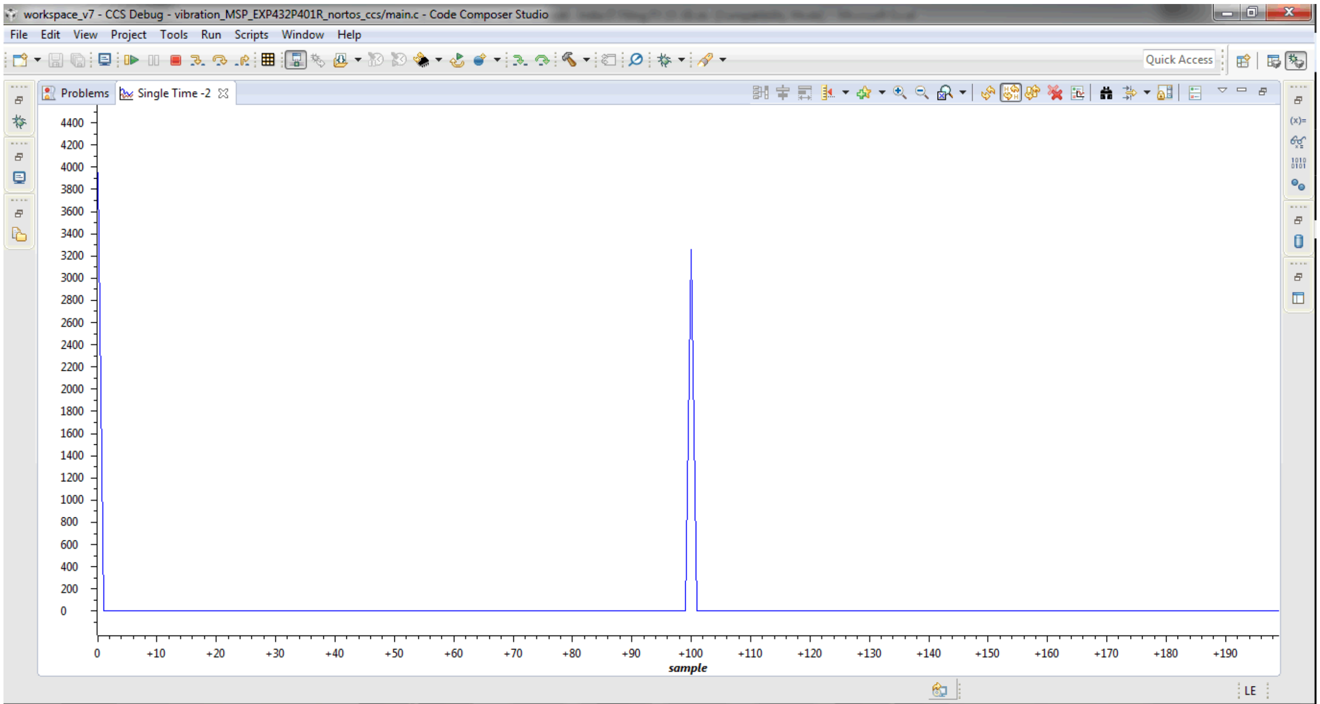

Figure 13. Subsonic FFT Profile for 100.0-Hz Sine Wave

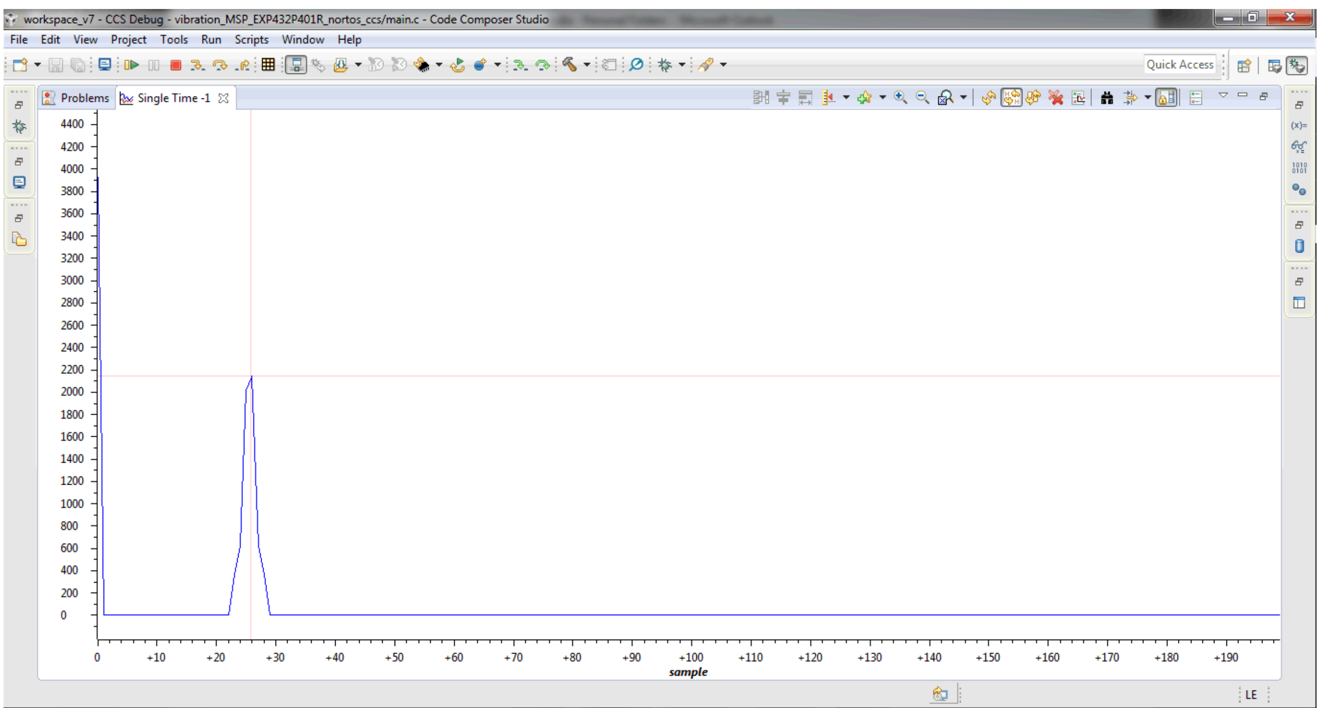

Figure 14. Audio-Band FFT Profile for 1.0-kHz Sine Wave

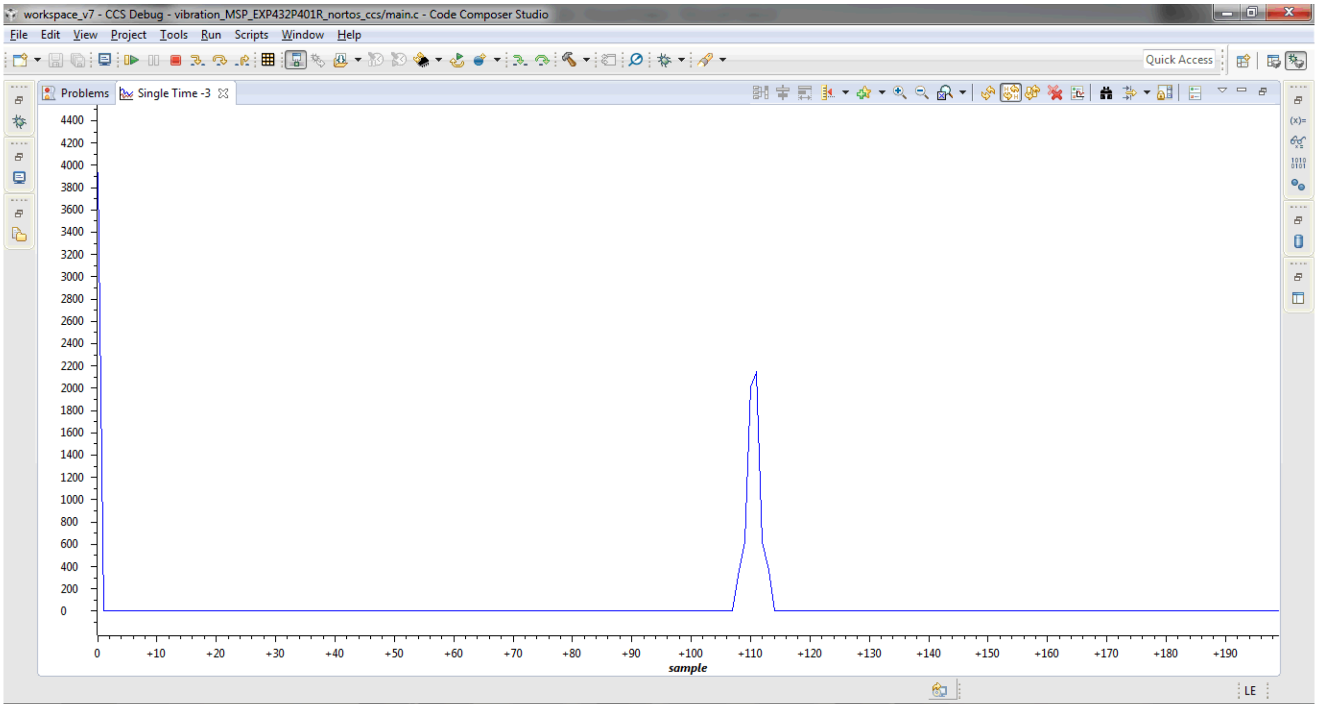

Figure 15. Ultrasonic FFT Profile for 43.0-kHz Sine Wave

4 Summary

This application report describes a predictive maintenance monitor based on the MSP432P401R MCU that consumes less than 25 µA during long periods of inactivity and yet is able to perform high-resolution frequency analysis of different defect frequencies generated by a motor over extended period. This system could be connected to a wireless solutions from the Texas Instruments SimpleLink ecosystem to achieve the goal of long-term remote monitoring of a motor, allowing operators to identify early wear and tear of the moving components and perform maintenance before a failure increases the cost of down time.

5 References

The following related documents have been used in this application report.

1. *MSP432P401R, MSP432P401M SimpleLink™ Mixed-Signal Microcontrollers*
2. SimpleLink™ MSP432™ Software Development Kit (SDK)

Revision History

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