AN13437

Using KE17Z LPIT to Trigger ADC Multi-Channel Conversion

Rev. 1 — 13 July 2023

Application note

Document Information

Information	Content
Keywords	AN13437, FRDM-KE17Z board, FTM, LPIT, ADC, TRGMUX, ADC sampling, PWM period, BLDC motor control applications
Abstract	This application note explains enabling ADC sampling in sequence in one PWM period on FRDM-KE17Z board by configuring the peripherals, FTM, LPIT, ADC, and TRGMUX.

1 Introduction

Kinetis KE1xZ256 MCUs are the leading parts for the KE1xZ family based on Arm Cortex-M0+ core. KE1xZ extends Kinetis E family to higher performance and broader scalability. It provides up to 256 KB flash, up to 48 KB RAM, and a complete set of analog or digital features. Robust and enhanced TSIs provide high-level stability and accuracy to HMI system of customers. 1 Msps ADC and FlexTimer help build a perfect solution for BLDC motor control systems.

In the BLDC motor control applications, users always use the case where the FTM triggers the ADC to achieve alternate conversion of multiple ADC channels. However, the ADC of KE17Z does not support sampling in sequence, so it needs LPIT to provide an alternate ADC hardware trigger source. TRGMUX implementation enables the realization of ADC sampling in sequence in one PWM period. LPIT can also provide precise timing interval as the trigger delay between the conversion of ADC multi-channels.

This application note explains how to enable ADC sampling in sequence on the FRDM-KE17Z board by using and interconnecting the peripherals, FTM, LPIT, ADC, and TRGMUX. It also provides an example use case to help users understand how to configure the peripherals.

The example code is developed on IAR-embedded workbench 9.10.2, based on SDK_2_10_1_FRDM-KE17Z. NXP FRDM-KE17Z board is used as the hardware.

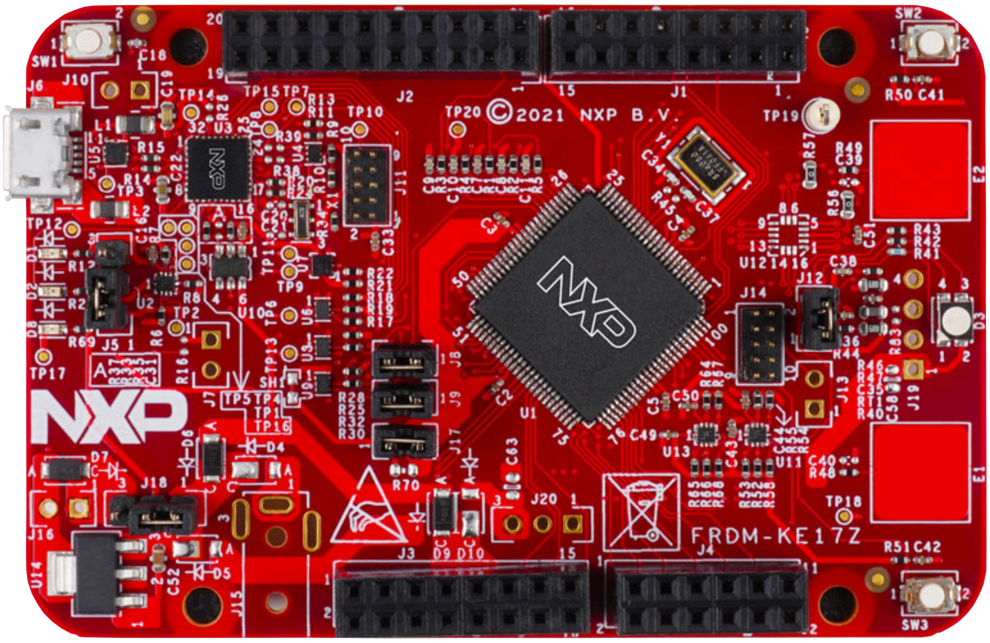

Figure 1. FRDM-KE17Z board

2 Peripheral features

Here is the detailed description for these peripherals FTM0, LPIT0, ADC0, and TRGMUX0.

2.1 Introduction of peripherals

The use case described in this application note utilizes the below listed peripherals (TRGMUX0, FTM0, LPIT0, ADC0), as described below:

- **TRGMUX0**

Trigger multiplexer (TRGMUX) is used for the trigger interconnection between different modules in KE17Z MCU. The TRGMUX module allows the software to select the different trigger sources for target peripherals, such as ADC and low-power period interrupt timer (LPIT). For TRGMUX0 of KE17Z, each TRGMUX control register supports up to 32 trigger inputs and up to 4 trigger outputs.

The control register `TRGMUX_LPIT0` is used to configure the trigger source for LPIT0 channel 0 ~ 3. The `TRGMUX_ADC0` is configured to select the trigger and pre-trigger sources for ADC0 to trigger four ADC0 channels.

- **FTM0**

  The FlexTimer module (FTM) is a multi-channel timer supporting input capture, output comparison, and generation of PWM signals. It can be used to control electric motor and power management applications. FTM can generate two kinds of trigger outputs, initialization trigger (`init_trig`) and channel match trigger (`ext_trig`). The trigger outputs are used as trigger source by other modules.

  FTM0 generates the continuous PWM signal. The initialization trigger of FTM0 is used as the trigger source of LPIT0 channel 0 via `TRGMUX_LPIT0` control register.
- **LPIT0**

  LPIT is a timer that supports up to 4 channels, with each channel having an independent pre-trigger and trigger. The trigger and the pre-trigger are paired up one-to-one. They are both selected by the same trigger control register. The trigger output assert is 1 clock cycle delayed of the pre-trigger pulse. The trigger output and the pre-trigger output de-assert at the same time.

  LPIT0 can define controllable delay time to determine a precise conversion time interval between different ADC channels. When LPIT0 channel times out, the trigger and pre-trigger assert to trigger ADC0 channel conversion. The implementation is through TRGMUX. The pre-trigger output is used to specify the selected ADC0 channel prior to ADC0 conversion.

  LPIT0 is configured to use an external trigger input source. FTM0 initialization and ADC0 conversion completion triggers are trigger sources for the four LPIT0 channels, shown as in Figure 2 and Figure 3.

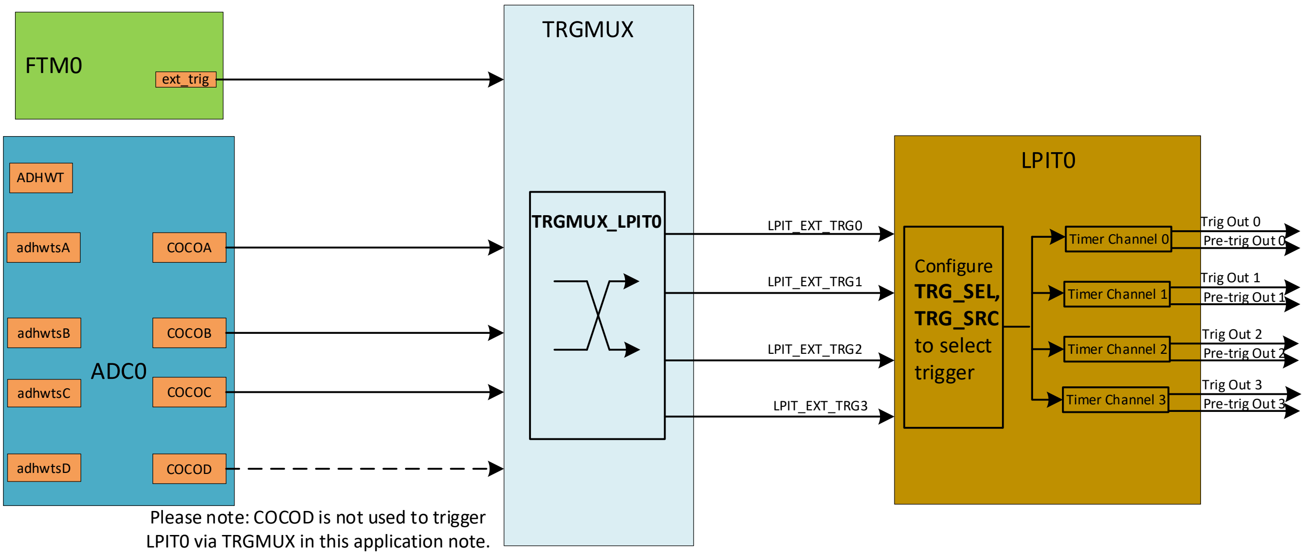

Figure 2. Hardware triggers and pre-triggers for LPIT0 via TRGMUX

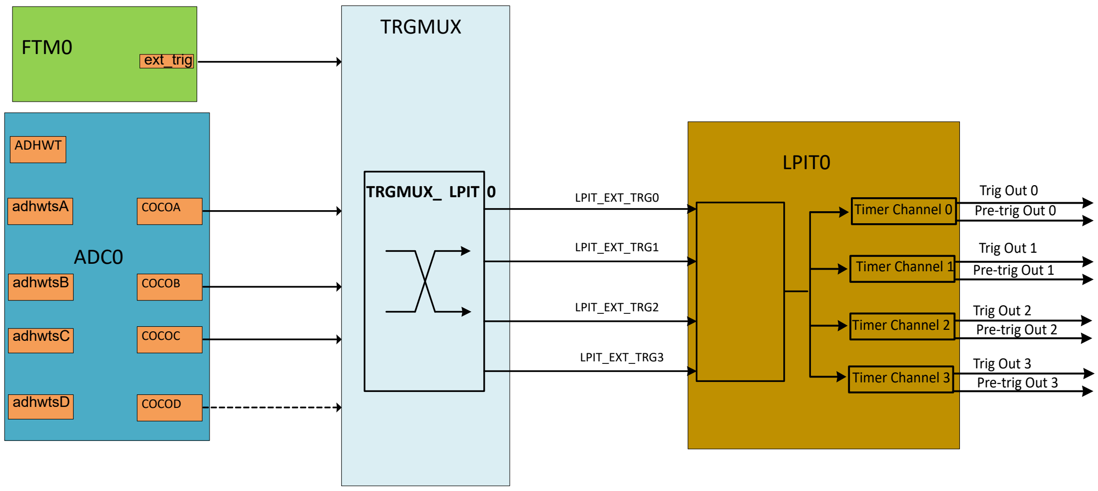

Figure 3. Hardware triggers and pre-triggers for LPIT0 via TRGMUX

Note:

– In this application note, COCOD is not used to trigger LPIT0 via TRGMUX.

– Configure `TRG_SEL` and `TRG_SRC` to select the trigger source for LPIT0 channels.

- **ADC0**

  The ADC module supports up to four single-ended inputs. It has four groups of configuration and result registers. For example:

  – *Group A: ADC_SC1A* and *ADC_RA*

  …

  …

  – *Group D: ADC_SC1D* and *ADC_RD*

  The four groups of configuration and result registers enable the ADC to configure inputs of four groups at the same time.

  The ADC0 supports two kinds of trigger inputs: pre-trigger and trigger. The pre-triggers are used to precondition the ADC0 block before the actual trigger occurs. When the trigger of ADC0 is asserted, the ADC0 starts the conversion according to the precondition determined by the pre-triggers.

  TRGMUX is used to select LPIT triggers and pre-triggers output as ADC trigger and pre-trigger sources. Set `SIM_ADCOPT[ADCxTRGSEL]` to `0x1` to select the TRGMUC output as the ADC trigger source. LPIT0 trigger output can be used as the trigger source to ADC0 via TRGMUX. Set `SIM_ADCOPT[ADCxPRETRGSEL]` to `0x1` to select the TRGMUX output as the ADC pre-trigger source. The four LPIT pre-triggers connect directly to ADC0 ADHWTSA ~ ADHWTSD ports to control the ADC channels via TRGMUX.

  Figure 4 shows the hardware trigger and pre-trigger connection between LPIT0 and ADC0 via TRGMUX. The LPIT0 is used as ADC0 hardware trigger source, whose implementation is via TRGMUX. Each LPIT0 channel supports one pre-trigger and one trigger. The LPIT0 channels are implemented based on independent counters. When used as an ADC trigger source, the channel outputs are connected together via an OR gate to generate the ADC hardware trigger.

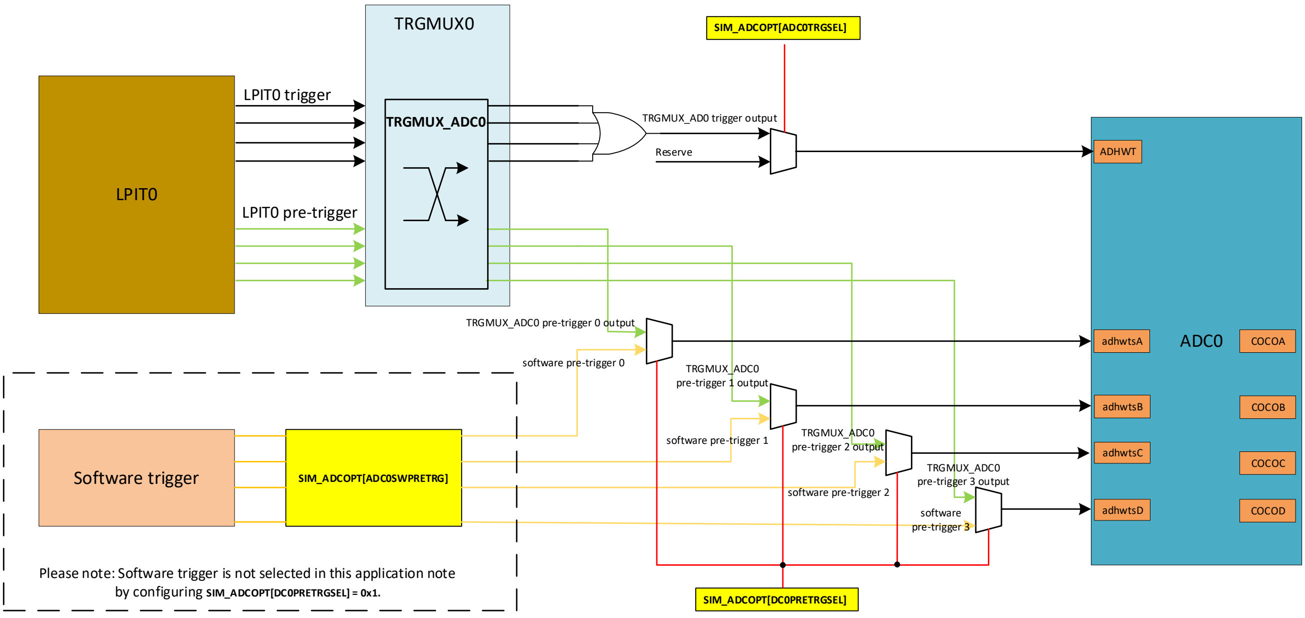

Figure 4. Hardware trigger and pre-trigger connection between LPIT0 and ADC0 via TRGMUX

2.2 Interconnection of peripherals

Figure 5 shows the process of using LPIT to achieve FTM triggering the alternate conversion of four ADC channels in one PWM cycle.

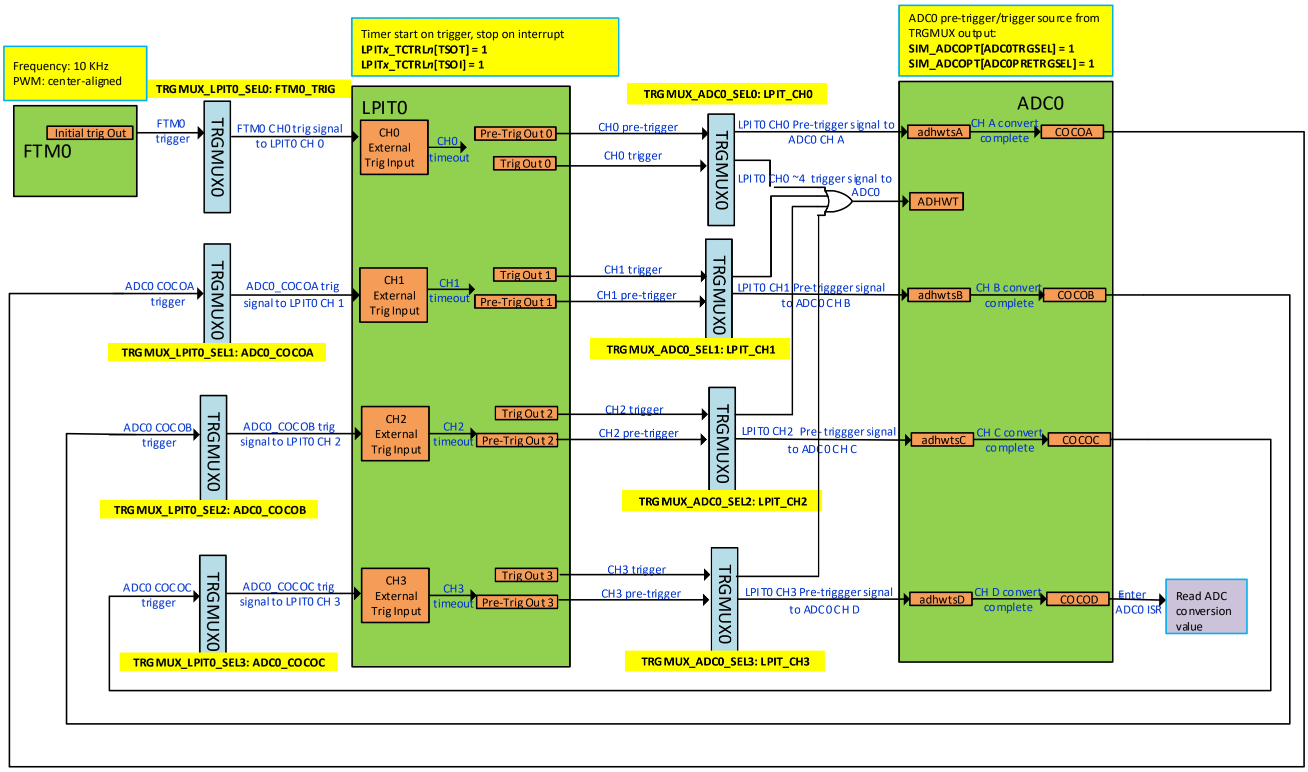

Figure 5. Interconnection of FTM0, LPIT0, ADC0, and TRGMUX0

In one PWM period:

- FTM0 channel 0 initial trigger output is the hardware trigger input source of LPIT0 channel 0, and the implementation is through TRGMUX0. The LPIT0 channel 0 timeout overflow trigger can be used as ADC0 hardware trigger source, whose implementation is via TRGMUX0. ADC0 conversion completion trigger of data result A is the trigger source for LPIT0 channel 1, whose implementation is through TRGMUX0.
- The LPIT0 channel 1 timeout overflow trigger can be used as ADC0 hardware trigger source, whose implementation is via TRGMUX0. ADC conversion completion trigger of data result B is the trigger source for LPIT0 channel 2 and its implementation is through TRGMUX0. This step ensures that the ADC0 conversion is completed before starting the next channel conversion.
- The LPIT0 channel 2 timeout overflow trigger can be used as ADC0 hardware trigger source, whose implementation is via TRGMUX0. ADC conversion completion trigger of data result C is the trigger source for LPIT0 channel 3, and is implemented through TRGMUX0.
- The LPIT0 channel 3 timeout overflow trigger can be used as ADC0 hardware trigger source, whose implementation is via TRGMUX0. Setting ADC conversion completion trigger of data result D enables the code to enter the ADC0 ISR handler. In addition, the conversion values of the four ADC0 channels are read.

2.3 Timings of LPIT0 trigger (pre-trigger to ADC0)

Figure 6 shows the trigger outputs of LPIT0 and pre-trigger outputs to ADC0 via TRGMUX.

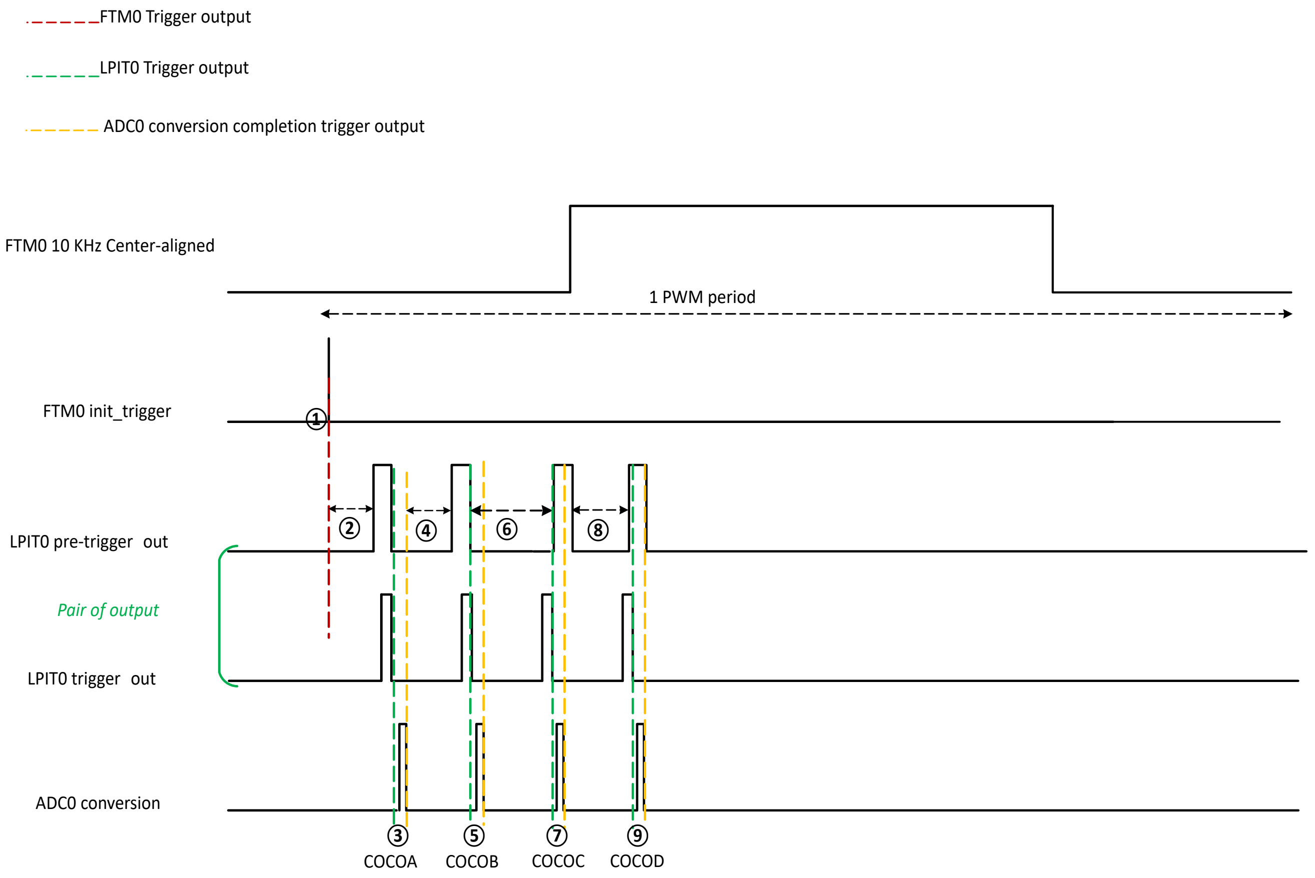

Figure 6. The timing of LPIT0 output trigger/pre-trigger to ADC0

The steps marked in Figure 6 are described below. These occur in the process of LPTI0 trigger of four ADC0 alternative conversion.

1. **FTM0 initialization trigger output**: This signal begins the whole process to trigger LPIT0 channel 0 to start counting. The LPIT0 channel 0 defines the delay time of this signal.
2. After LPIT0 channel 0 defines the delay time (marked as 2 in Figure 6), one pair of pre-trigger and trigger outputs to ADC0, ADHWTSA, and ADHWT are generated, when LPIT0 channel 0 times out.
3. Now, ADC0 channel configured by **ADC0 Group A** starts the conversion. When the conversion is complete, ADC0 channel 1 completion triggers, and COCOA is generated to trigger LPIT0 channel 1 to begin counting.
4. After LPIT0 channel 1 defines the delay time (marked as 4), a pair of pre-trigger and trigger outputs to ADC0, ADHWTSB, and ADHWT are generated when LPIT0 channel 2 times out.
5. The ADC0 channel configured by **ADC0 Group B** starts conversion. When the conversion is complete, ADC0 channel 2 completion triggers, and COCOB is generated to trigger LPIT0 channel 2 to begin counting.
6. After LPIT0 channel 2 defines the delay time (marked as 6), one pair of pre-trigger and trigger outputs to ADC0, ADHWTSC, and ADHWT are generated when LPIT0 channel 2 times out.
7. The ADC0 channel configured by **ADC0 Group C** starts conversion. When the conversion is complete, ADC0 channel C completion triggers and COCOC is generated to trigger LPIT0 channel 3 to begin counting.
8. After LPIT0 channel 3 defines the delay time (marked as 8), one pair of pre-trigger and trigger outputs to ADC0, ADHWTSD, and ADHWT are generated when LPIT0 channel 3 times out.
9. The ADC0 channel configured by **ADC0 Group D** starts conversion. When the conversion is complete, this round of ADC four channel conversion is completed, and the conversion values can be read. After FTM0 time period, the initialization trigger generated in each FTM0 channel 0 time period restarts a new round of conversion for the ADC0 channels.

3 Implementation details

This section shows a use case for implementation details of LPIT0 trigger ADC0 four channels alternate conversion. In the use case, the peripherals FTM0, LPIT0, ADC0, TRGMUX0 must be configured.

Figure 7 shows the flowchart of the attached example code.

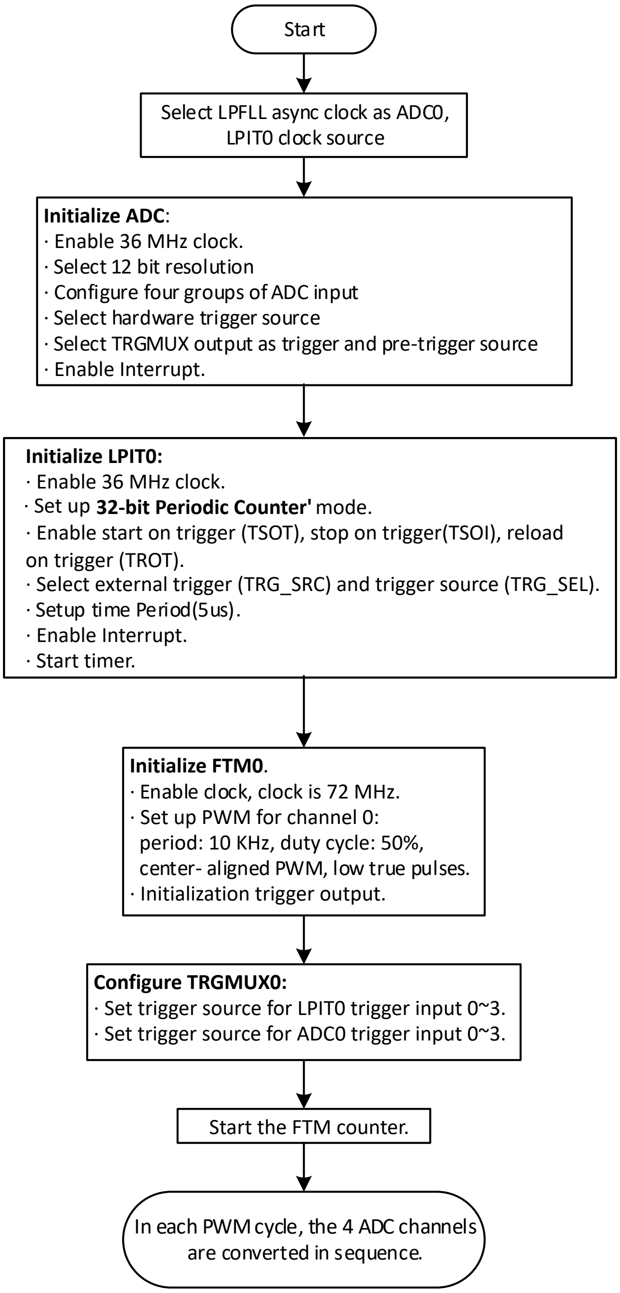
Start
Select LPFLL async clock as ADC0, LPIT0 clock source
Initialize ADC:
· Enable 36 MHz clock.
· Select 12 bit resolution
· Configure four groups of ADC input
· Select hardware trigger source
· Select TRGMUX output as trigger and pre-trigger source
· Enable Interrupt.
Initialize LPIT0:
· Enable 36 MHz clock.
· Set up 32-bit Periodic Counter' mode.
· Enable start on trigger (TSOT), stop on trigger(TSOI), reload on trigger (TROT).
· Select external trigger (TRG_SRC) and trigger source (TRG_SEL).
· Setup time Period(5us).
· Enable Interrupt.
· Start timer.
Initialize FTM0.
· Enable clock, clock is 72 MHz.
· Set up PWM for channel 0:
period: 10 KHz, duty cycle: 50%,
center- aligned PWM, low true pulses.
· Initialization trigger output.
Configure TRGMUX0:
· Set trigger source for LPIT0 trigger input 0~3.
· Set trigger source for ADC0 trigger input 0~3.
Start the FTM counter.
In each PWM cycle, the 4 ADC channels are converted in sequence.

Figure 7. Flowchart of software configuration

The initialization sequence for the four modules is shown in Figure 7.

1. **Initialization of peripherals**
   Initializing ADC first can prevent the COCO flag generated during ADC arbitration from accidentally triggering LPIT0 channel counting. Therefore, the initialization sequence in the above flowchart is recommended. This sequence ensures that no unexpected signals are generated and the ADC0 channels are sampled in order.
2. **Clock configuration**
   In this application, FTM0 input clock is 72 MHz system clock. ADC0 and LPIT0 select LPFLL peripheral clock source to ensure that they can capture the input of the trigger signal every time.

3.1 TRGMUX configuration

In this application, TRGMUX implements the connection of all trigger signals.

TRGMUX is configured such that the FTM0 hardware trigger is selected as the LPIT0 external trigger 0 input, as shown in the below sample code.

```
/* LPIT external trigger 0 trigger input source: FTM0 */
TRGMUX_SetTriggerSource(TRGMUX0, kTRGMUX_Lpit, kTRGMUX_TriggerInput0,
 kTRGMUX_SourceFtm0);
```

ADC0 conversion completion triggers (COCOA, COCOB, and COCOC) are selected as the LPIT0 other external trigger inputs (external trigger 1, external trigger 2, and external trigger 3) by configuring the TRGMUX.

```
/* LPIT external trigger 1 input source: ADC0_COCOA */
TRGMUX_SetTriggerSource(TRGMUX0, kTRGMUX_Lpit, kTRGMUX_TriggerInput1,
 kTRGMUX_SourceAdc0CocoA);
/* LPIT external trigger 2 input source: ADC0_COCOB */
TRGMUX_SetTriggerSource(TRGMUX0, kTRGMUX_Lpit, kTRGMUX_TriggerInput2,
 kTRGMUX_SourceAdc0CocoB);
/* LPIT external trigger 3 input source: ADC0_COCOC */
TRGMUX_SetTriggerSource(TRGMUX0, kTRGMUX_Lpit, kTRGMUX_TriggerInput3,
 kTRGMUX_SourceAdc0CocoC);
```

Similarly, the TRGMUX is configured to set LPIT0 channel timeout pre-triggers and triggers to be selected as ADC0 trigger input of four groups of channels. The sample code is shown below:

```
/* ADC0 Channel A trigger input source: LPIT0CH0 */
TRGMUX_SetTriggerSource(TRGMUX0, kTRGMUX_Adc0, kTRGMUX_TriggerInput0,
 kTRGMUX_SourceLpit0Ch0);
/* ADC0 Channel B trigger input source: LPIT0CH1 */
TRGMUX_SetTriggerSource(TRGMUX0, kTRGMUX_Adc0, kTRGMUX_TriggerInput1,
 kTRGMUX_SourceLpit0Ch1);
/* ADC0 Channel C trigger input source: LPIT0CH2 */
TRGMUX_SetTriggerSource(TRGMUX0, kTRGMUX_Adc0, kTRGMUX_TriggerInput2,
 kTRGMUX_SourceLpit0Ch2);
/* ADC0 Channel D trigger input source: LPIT0CH3 */
TRGMUX_SetTriggerSource(TRGMUX0, kTRGMUX_Adc0, kTRGMUX_TriggerInput3,
 kTRGMUX_SourceLpit0Ch3);
```

3.2 FTM configuration

The configuration described here is used in BLDC motor control applications. FTM0 is initialized to the output center-aligned PWM signal, and the frequency of PWM is configured as 10 kHz. Select FTM0 channel 0 to generate the external initialization trigger by setting `FTM0_EXTTRIG_ INITTRIGEN` to `1`.

The initialization code to be used for reference is as follows:

```
/* Initialize FTM module. */
FTM_GetDefaultConfig(&ftmConfigStruct);
ftmConfigStruct.extTriggers =kFTM_InitTrigger; /* Enable to output the
 initialization trigger. */
FTM_Init(DEMO_FTM_BASE, &ftmConfigStruct);
/* Configure FTM params with frequency 10 KHz */
pwmParam.chnlNumber = kFTM_Chnl_0;
pwmParam.level = pwmLevel;
pwmParam.dutyCyclePercent = 50U;              /* Percent: 0 - 100. */
pwmParam.firstEdgeDelayPercent = 0U;
FTM_SetupPwm(DEMO_FTM_BASE, &pwmParam, 1U, kFTM_CenterAlignedPwm,
 DEMO_FTM_PWM_HZ, DEMO_FTM_COUNTER_CLOCK_HZ);
```

3.3 ADC configuration

ADC0 contains four groups of configuration and result registers: A, B, C, and D. Four inputs can be configured at the same time, but the ADC itself does not support the sequential conversion of four groups of channels.

- Set ADC0_SC1m_ADCH to the channel number to select the external channels as ADC0 input.

Initialize ADC0, in addition to the clock select, set the hardware trigger, arbitration, and some basic initialization configurations. TRGMUX outputs should be selected as pre-trigger sources and trigger sources of ADC0.

To perform this:

- Set SIM_ADCOPT[ADC0TRGSEL] to 0x1, and
- Set SIM_ADCOPT[ADC0PRETRGSEL] to 0x1

The above register configurations enables the trigger signals between ADC0 and LPIT0 to be connected via the TRGMUX_LPIT0 control register.

```
/* Configure SIM for ADC hardware trigger source selection */
/* Use TRGMUX output as ADC pre-trigger trigger source */
 SIM->ADCOPT |= SIM_ADCOPT_ADC0TRGSEL(1U);
 SIM->ADCOPT |= SIM_ADCOPT_ADC0PRETRGSEL(1U);
```

3.4 LPIT0 configuration

The four LPIT0 channels should be configured to trigger ADC four channels conversion. The below points should be noted:

- The time period of the LPIT0 channel is used to control the time interval between two ADC channels conversion.
- LPIT0_TCTRLn_TSOT and LPIT0_TCTRLn_TSOI must be set to realize that the LPIT0 timer starts on trigger and stops after timeout.
- The four LPIT0 channels are independent. Each channel (LPIT0_TVALn, TRG_SRC, TRG_SEL) should be configured independently to define the delay time and trigger source.
- Follow the steps listed below to implement ADC channel conversion one by one, in the sequence below: (**ADC0 CHA > ADC0 CHB > ADC0 CHC > ADC0 CHD**)
  LPTI0 must configure TRG_SRC and TRG_SEL for each LPIT channel to select:
  - ADC0 COC0A as the trigger source of LPIT0 channel 1
  - ADC0 COC0B as the trigger source of LPIT0 channel 2
  - ADC0 COC0C as the trigger source of LPIT0 channel 3

  The above configuration ensures that the ADC0 is triggered again only after the conversion is completed. The following sample code shows the configuration of one LPIT channel.

```
/* Init lpit module */
 LPIT_Init(DEMO_LPIT_BASE, &lpitConfig);
 lpitChannelConfig.chainChannel           = false;
 lpitChannelConfig.enableReloadOnTrigger  = true;
/* Timer starts to decrement when rising edge on selected trigger is detected
 */
 lpitChannelConfig.enableStartOnTrigger   = true;
 lpitChannelConfig.enableStopOnTimeout    = true;
 lpitChannelConfig.timerMode              = kLPIT_PeriodicCounter;
```

```
/* Set default values for the trigger source */
 lpitChannelConfig.triggerSource          = kLPIT_TriggerSource_External;
/* Init lpit channel 0 */
 lpitChannelConfig.triggerSelect          = kLPIT_Trigger_TimerChn0;
 LPIT_SetupChannel(DEMO_LPIT_BASE, kLPIT_Chnl_0, &lpitChannelConfig);
```

3.5 Timing

Figure 8 shows the timing of LPIT0 trigger for four ADC channels alternate conversion in one PWM period.

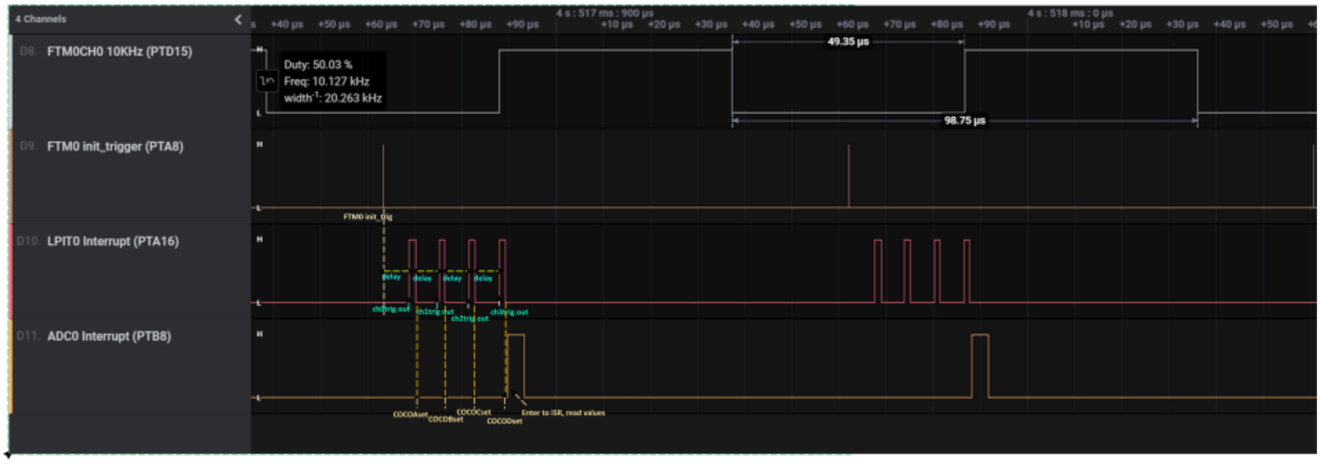

Figure 8. Timing of LPIT0 trigger for four ADC channels in this use case

4 Conclusion

This application note describes how to use LPIT0 to implement periodic triggering of four ADC channel conversions. It explains the inter-connections of FTM, LPIT, ADC, and TRGMUX. Users can easily implement the function by following the details of configuration described in this document.

5 References

Following references are available on NXP website:

- *Kinetis KE17Z/13Z/12Z with up to 256 KB Flash Reference Manual* (Document KE1xZP100M72SF1RM)
- For other documentation, refer to 72MHz, 5V Main Stream CM0+ MCU with NXP Touch (TSI) and CAN Control

6 Revision history

Table 1 summarizes the revisions to this document.

Revision history

Revision number	Date	Substantive changes
1	13 July 2023	• Added Section 7 • Made some editorial changes • Attached AN13437SW with this application note
0	06 December 2021	Initial release

7 Legal information

7.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

7.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. - NXP B.V. is not an operating company and it does not distribute or sell products.

7.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Kinetis — is a trademark of NXP B.V.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com

Date of release: 13 July 2023
Document identifier: AN13437