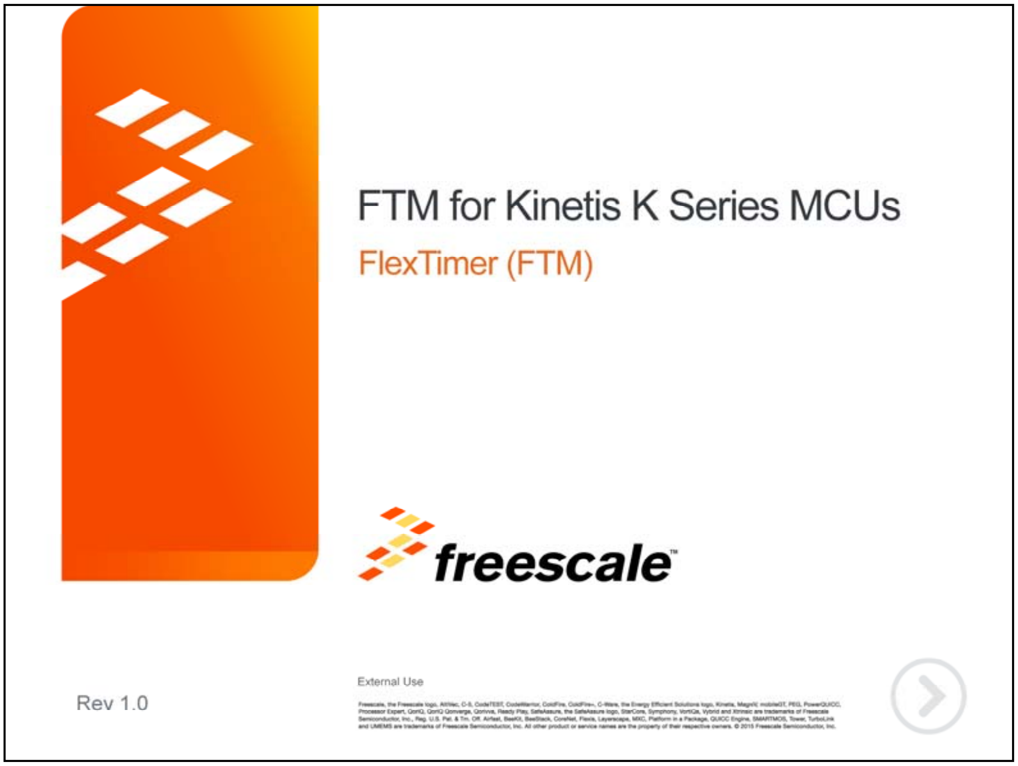


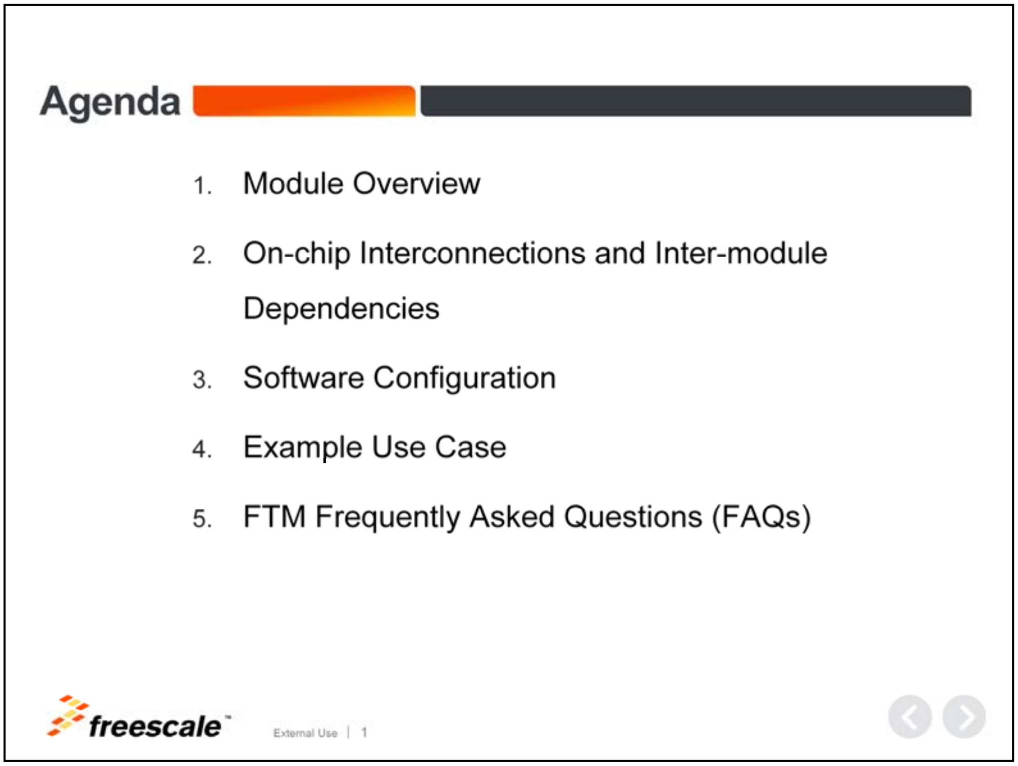
FTM for Kinetis K Series MCUs

FlexTimer (FTM)

Rev 1.0

External Use

Hello, and welcome to this presentation of the FlexTimer – or FTM – module for Kinetis K series MCUs. In this session, you'll learn about the FTM, its main features and the application benefits of leveraging this function.



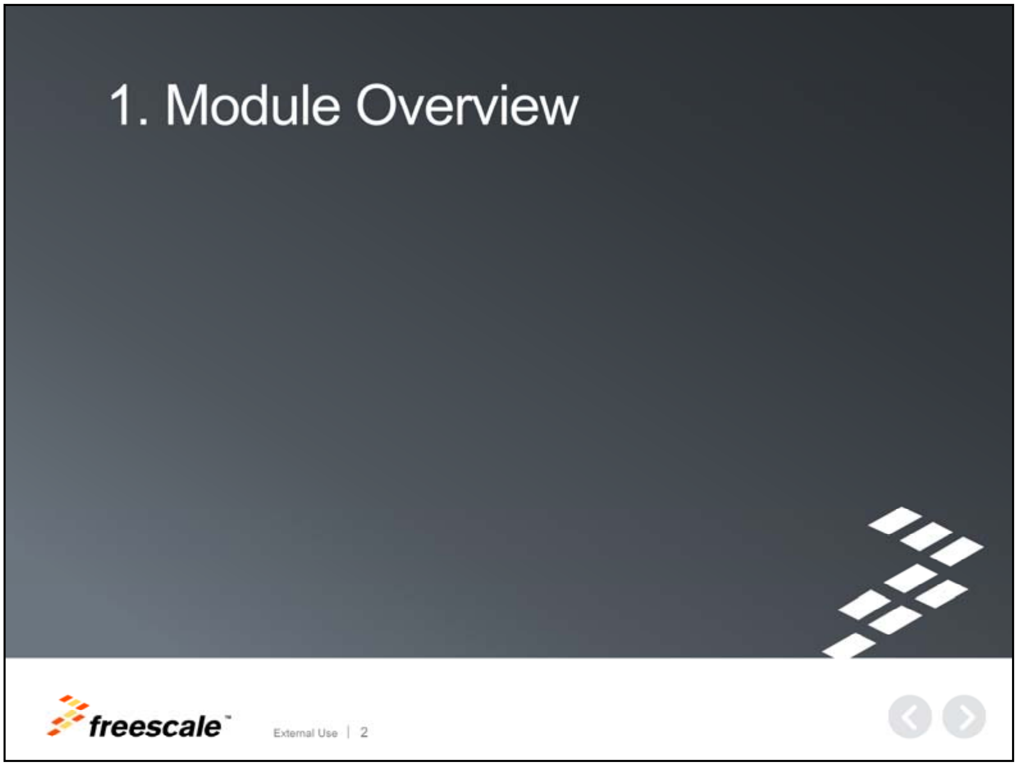
Agenda

1. Module Overview
2. On-chip Interconnections and Inter-module Dependencies
3. Software Configuration
4. Example Use Case
5. FTM Frequently Asked Questions (FAQs)

freescale External Use | 1

In this presentation we’ll cover:

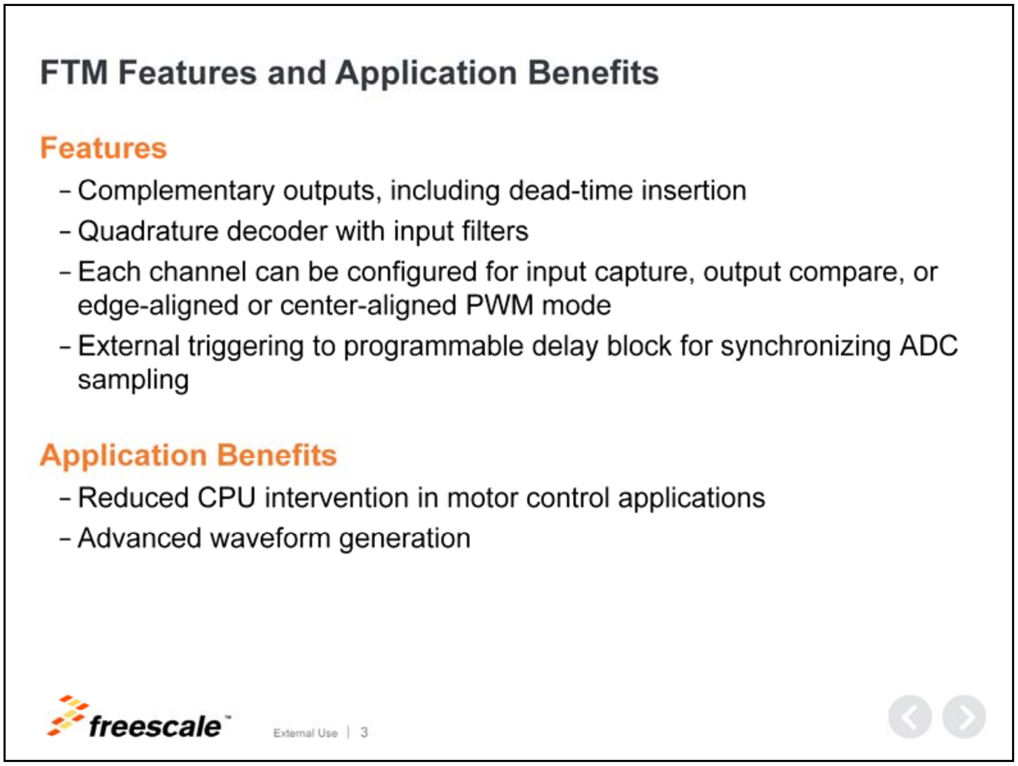
- An overview of the FTM module itself
- The on-chip interconnections and inter-module dependencies
- Software configurations
- An example use case
- And some frequently asked questions



1. Module Overview

External Use | 2

First, let’s start with an overview of the module.



FTM Features and Application Benefits

Features

- Complementary outputs, including dead-time insertion
- Quadrature decoder with input filters
- Each channel can be configured for input capture, output compare, or edge-aligned or center-aligned PWM mode
- External triggering to programmable delay block for synchronizing ADC sampling

Application Benefits

- Reduced CPU intervention in motor control applications
- Advanced waveform generation

External Use | 3

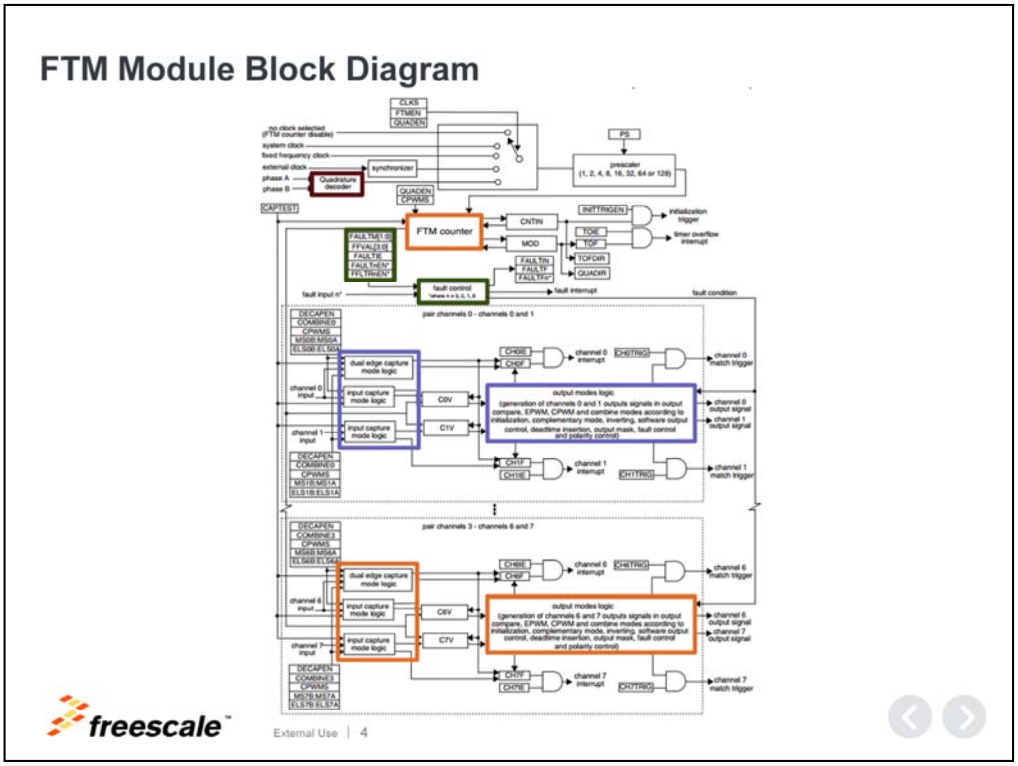
FTM Features and Application Benefits

The FTM module features include:

- Complementary pulse width modulation (or PWM) outputs with dead-time insertion
- Quadrature decoder (Please refer to the device reference manual for the decoder availability)
- Multiple channels with input capture, output compare and PWM capabilities
- And the FTM module provides external triggering to some peripherals, like the programmable delay block or PDB to synchronize ADC sampling

Application benefits include:

- Reduced CPU intervention in handling PWM for motor control applications
- Advanced waveform generation for motor control and external sensors



FTM Module Block Diagram

External Use | 4

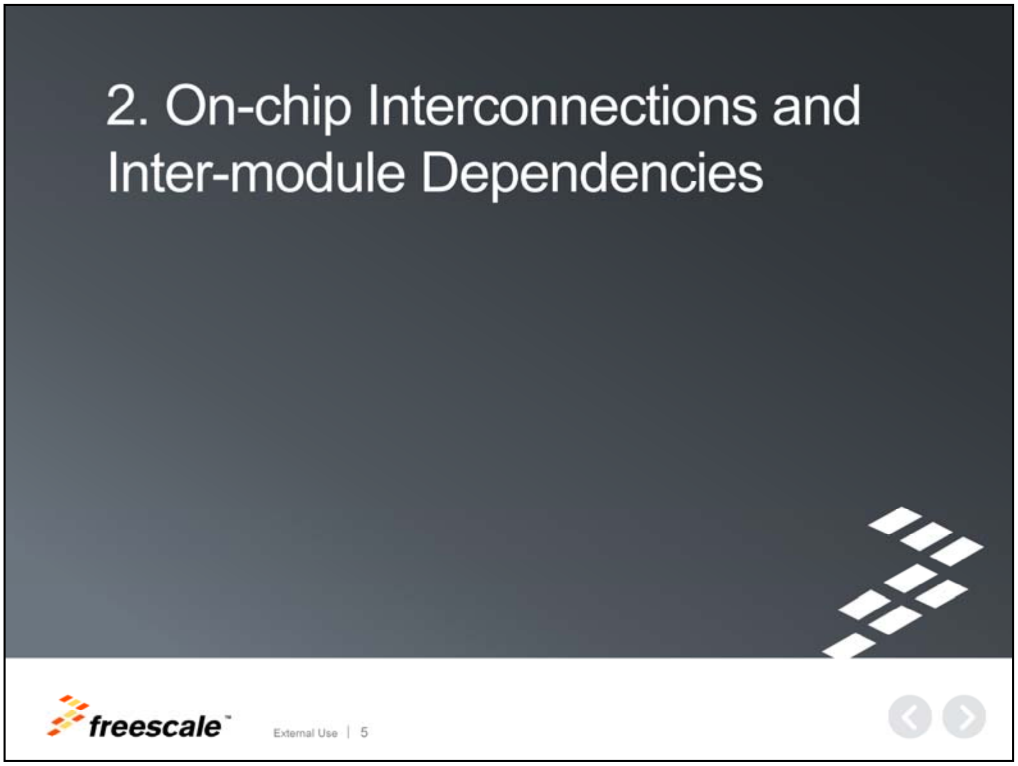
FTM Module Block Diagram

The 16-bit FTM counter outlined at the top of the block diagram in orange has multiple clock sources. The FTM module can be nested with other FTM modules to create a 32-bit counter.

Outlined in green are the fault inputs and control with programmable polarity that allow the halting of PWM outputs whenever a fault is detected.

Outlined in brown is the quadrature decoder, which is present for speed and position measurement in motor control applications.

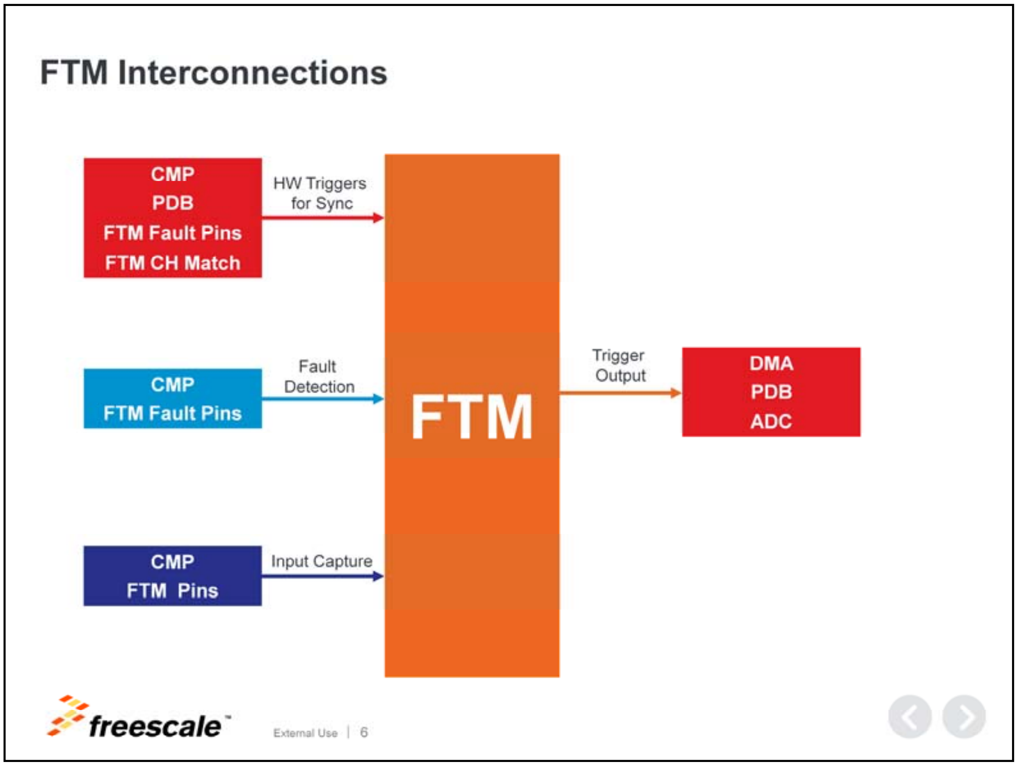
Outlined in purple are all PWM channels that are input capture or output compare capable. Please refer to the device reference manual for the exact number of channels.



2. On-chip Interconnections and Inter-module Dependencies

External Use | 5

Now, let’s talk about on-chip interconnection and inter-module dependencies.



FTM Interconnections

CMP
PDB
FTM Fault Pins
FTM CH Match

HW Triggers for Sync

CMP
FTM Fault Pins

Fault Detection

CMP
FTM Pins

Input Capture

FTM

Trigger Output

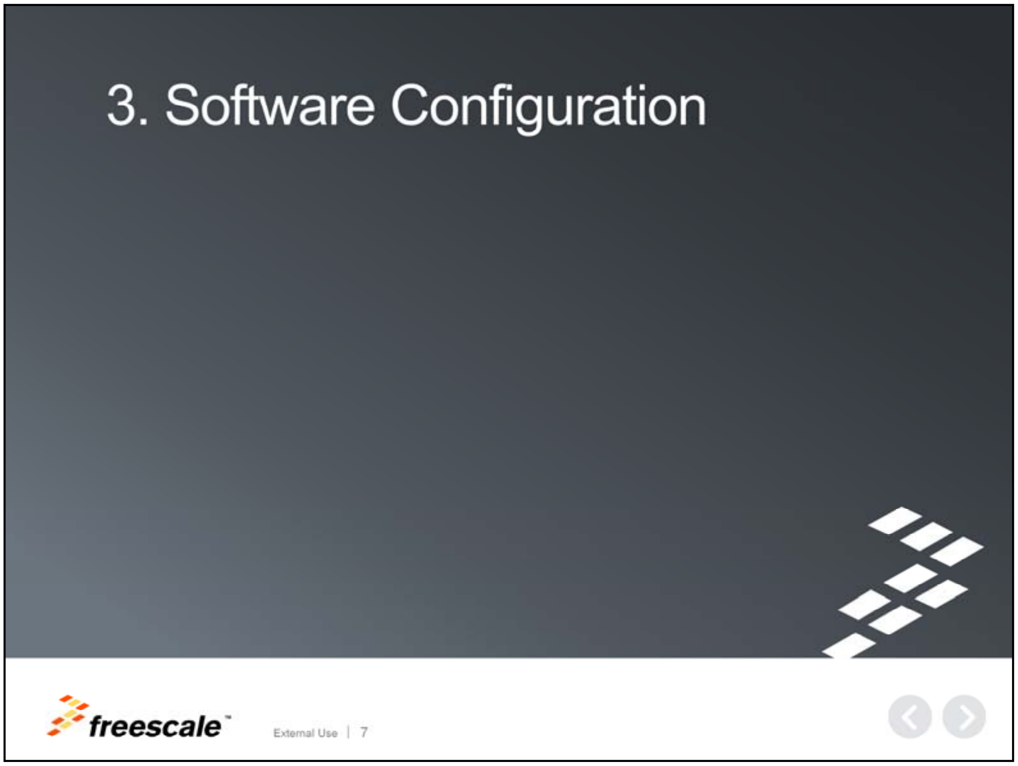
DMA
PDB
ADC

freescale External Use | 6

FTM Interconnections

The following diagram displays the FTM and it’s relationship with other peripherals:

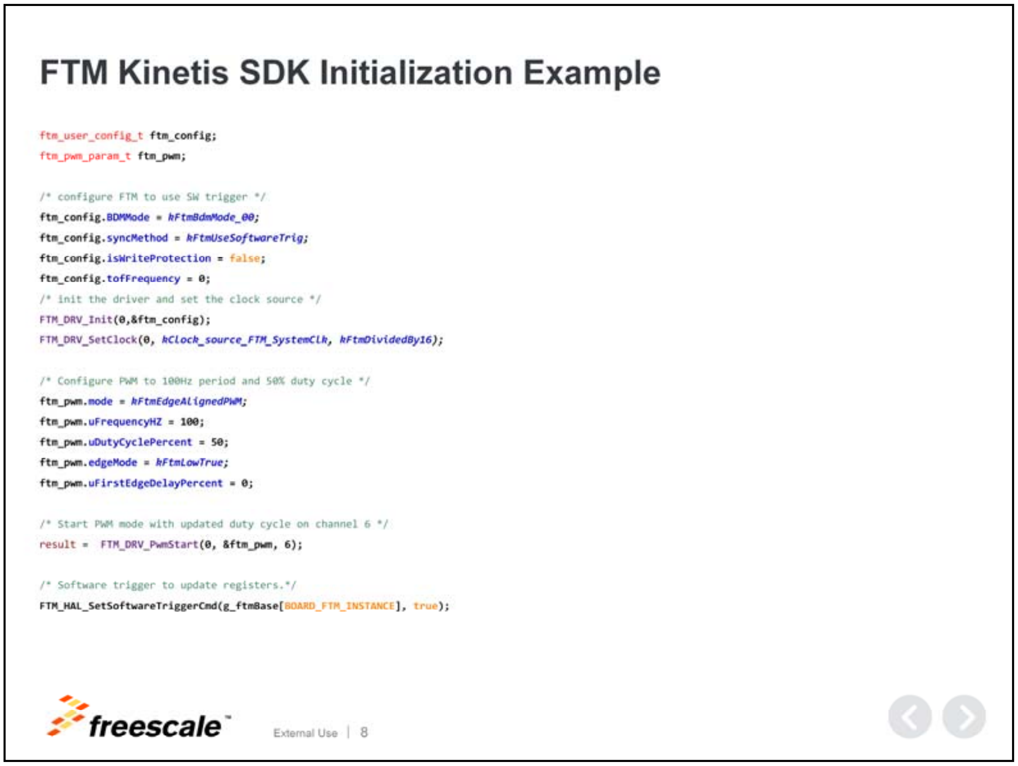
- The FTM module allows synchronization hardware triggers from different modules like the comparator, the programmable delay block, or from other FTM modules.
- The analog comparator and FTM pins can be used for fault detection.
- The comparator or FTM dedicated pins can be used for input capture functionality.
- The FTM module can trigger a DMA transaction or a PDB timer for ADC sampling. This is useful for the feedback loop of motor control algorithms.



3. Software Configuration

External Use | 7

In this next section we’ll discuss software configuration.



FTM Kinetis SDK Initialization Example

```
ftm_user_config_t ftm_config;
ftm_pwm_param_t ftm_pwm;

/* configure FTM to use SW trigger */
ftm_config.BDMMode = kFtmBdmMode_00;
ftm_config.syncMethod = kFtmUseSoftwareTrig;
ftm_config.isWriteProtection = false;
ftm_config.tofFrequency = 0;
/* init the driver and set the clock source */
FTM_DRV_Init(0,&ftm_config);
FTM_DRV_SetClock(0, kClock_source_FTM_SystemClk, kFtmDividedBy16);

/* Configure PWM to 100Hz period and 50% duty cycle */
ftm_pwm.mode = kFtmEdgeAlignedPWM;
ftm_pwm.uFrequencyHZ = 100;
ftm_pwm.uDutyCyclePercent = 50;
ftm_pwm.edgeMode = kFtmLowTrue;
ftm_pwm.uFirstEdgeDelayPercent = 0;

/* Start PWM mode with updated duty cycle on channel 6 */
result =  FTM_DRV_PwmStart(0, &ftm_pwm, 6);

/* Software trigger to update registers.*/
FTM_HAL_SetSoftwareTriggerCmd(g_ftmBase[BOARD_FTM_INSTANCE], true);
```

FTM Kinetis SDK Initialization Example

The Kinetis SDK integrates a FTM peripheral driver that can be used to enable PWM output.

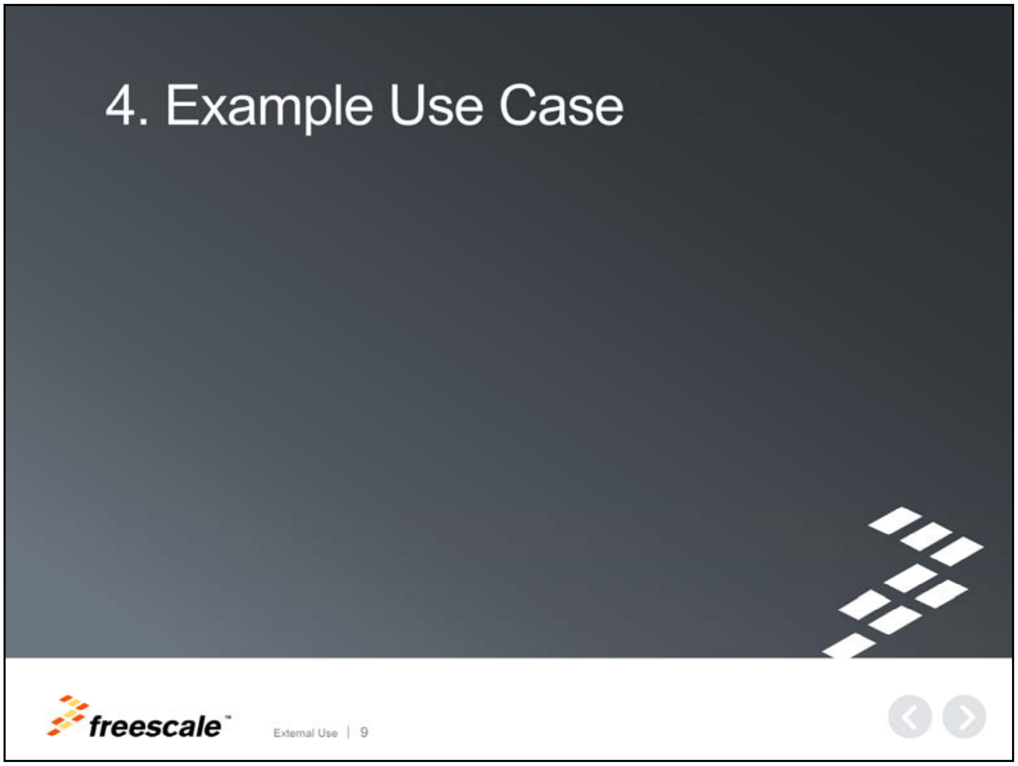
First, a configuration structure must be filled with:

- The FTM behavior when the MCU is in debug mode
- The sync method used for writing the registers, and
- The write protected registers which can be modified after initial configuration

Once the structure is filled, the API should be called to initialize the module and set the FTM clock with the desired prescaler.

In this FTM initialization example, the structure is configured for PWM generation with a 100Hz period and 50% duty cycle. Next, the API should be called to configure the PWM channel – in this case, we use channel 6.

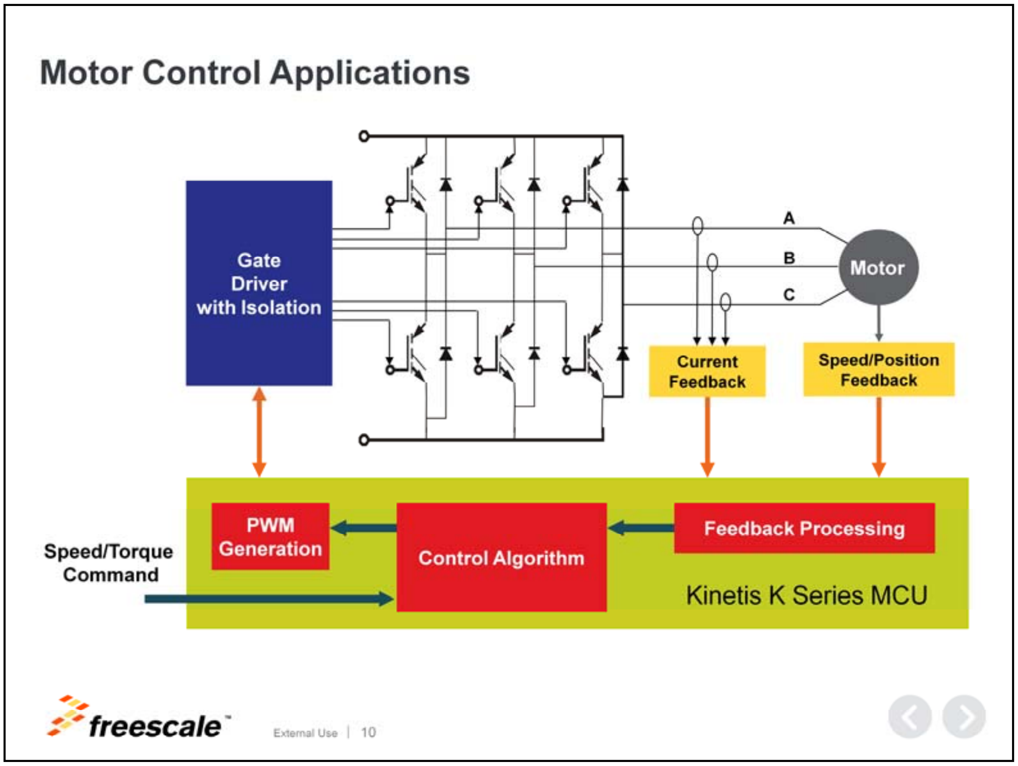
A trigger is required to get the previous configuration loaded into the FTM registers. In this example, we use a software trigger.



4. Example Use Case

External Use | 9

Let’s review an example use case.



Motor Control Applications

External Use | 10

Motor Control Applications

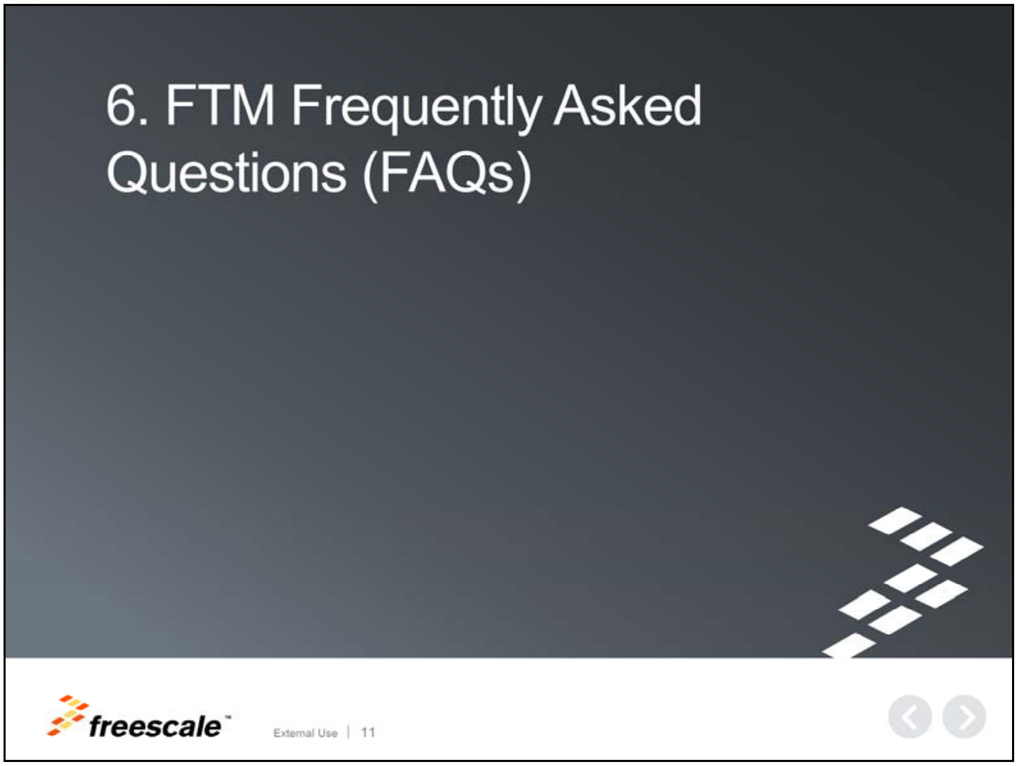
The FTM module eases the development of motor control applications.

The control algorithm updates the different PWM channels to match the speed and torque requirements. These PWM channels are connected to external analog circuitry that controls the motor.

The FTM and PDB with ADC synchronization can be used to measure the current used by the motor.

For certain motor control applications, position feedback is required. The quadrature decoder on the FTM module can provide this information.

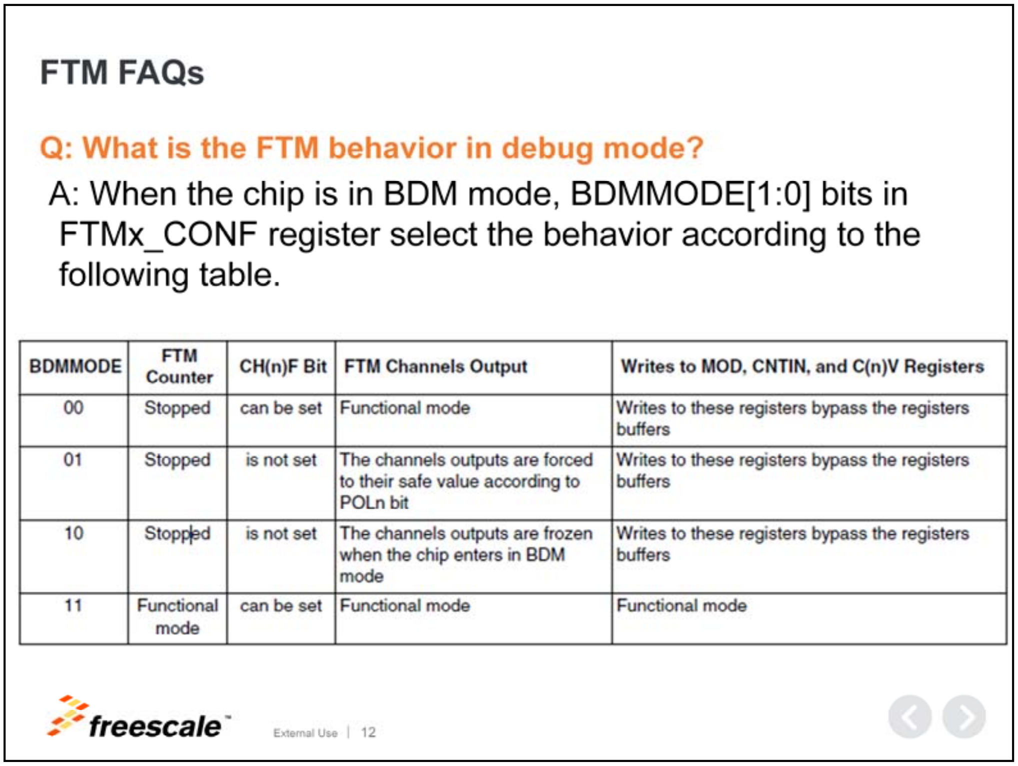
This information is processed and then sent to the control algorithm to adjust the PWM.



6. FTM Frequently Asked Questions (FAQs)

freescale External Use | 11

Finally, let’s discuss some frequently asked questions.



FTM FAQs

Q: What is the FTM behavior in debug mode?

A: When the chip is in BDM mode, BDMMODE[1:0] bits in FTMx_CONF register select the behavior according to the following table.

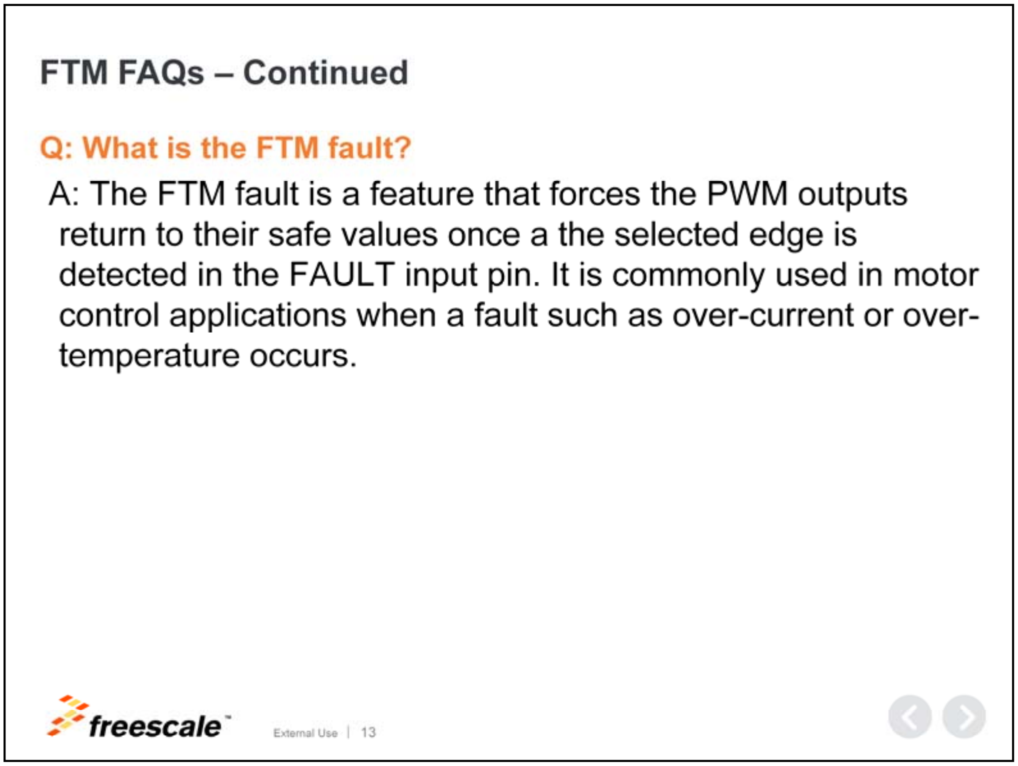
BDMMODE	FTM Counter	CH(n)F Bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
00	Stopped	can be set	Functional mode	Writes to these registers bypass the registers buffers
01	Stopped	is not set	The channels outputs are forced to their safe value according to POLn bit	Writes to these registers bypass the registers buffers
10	Stopped	is not set	The channels outputs are frozen when the chip enters in BDM mode	Writes to these registers bypass the registers buffers
11	Functional mode	can be set	Functional mode	Functional mode

External Use | 12

FTM FAQs

Question: What is the FTM behavior in debug mode?

Answer: The FTM module behavior during BDM mode or debug mode is configurable as specified in this table.



FTM FAQs – Continued

Q: What is the FTM fault?

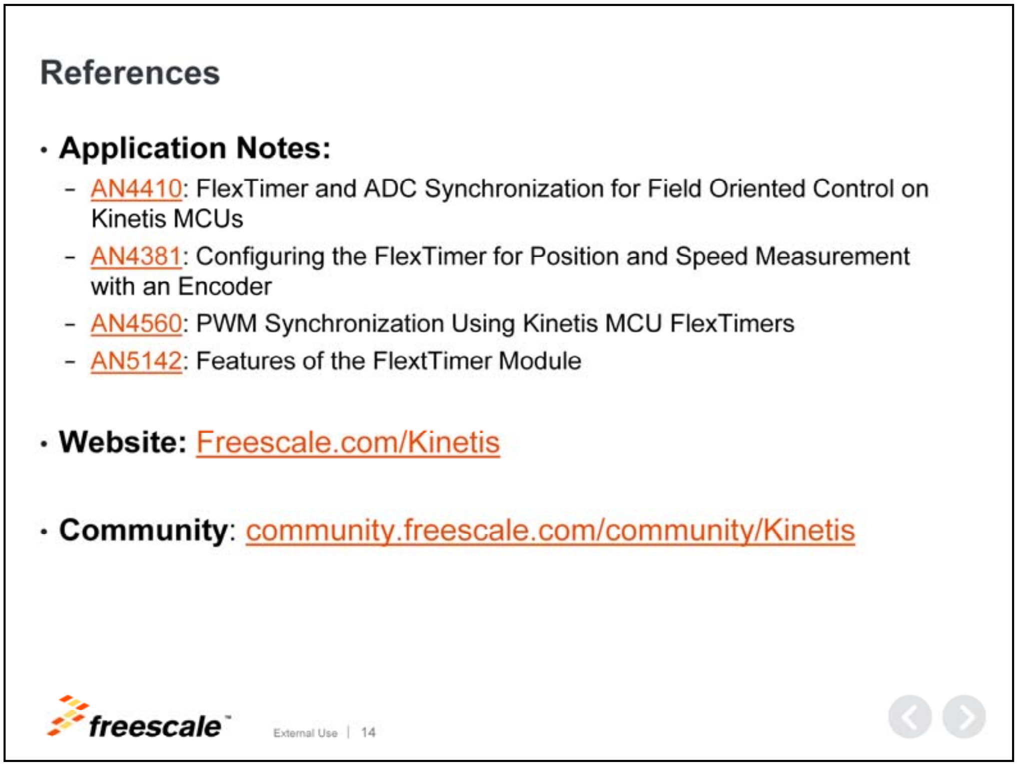
A: The FTM fault is a feature that forces the PWM outputs return to their safe values once a the selected edge is detected in the FAULT input pin. It is commonly used in motor control applications when a fault such as over-current or over-temperature occurs.

External Use | 13

FTM FAQs – Continued

What is the FTM Fault?

The FTM fault is a feature on the FTM module that will automatically shutdown PWM outputs when the fault pins are asserted, based on the configured polarity. This feature is important in control applications such as motor control where in the case of over current, the outputs can be automatically and quickly disabled to avoid hardware damage.



References

- **Application Notes:**
  - AN4410: FlexTimer and ADC Synchronization for Field Oriented Control on Kinetis MCUs
  - AN4381: Configuring the FlexTimer for Position and Speed Measurement with an Encoder
  - AN4560: PWM Synchronization Using Kinetis MCU FlexTimers
  - AN5142: Features of the FlextTimer Module
- **Website:** Freescale.com/Kinetis
- **Community**: community.freescale.com/community/Kinetis

freescale External Use | 14

References

For more information, please visit the FTM application notes listed here.

We also invite you to visit us on the web at Freescale.com/Kinetis and check out our community page.

www.Freescale.com