# AN12847

## KM35Z512 Migration Guide

Migrating from KM34Z256 to KM35Z512

Rev. 0 — 05/2020

Application Note

by: NXP Semiconductors

Contents



## 1 Introduction

This document describes how to migrate from Kinetis MKM34Z256xxx7 MCUs with up to 256 KB of Flash to MKM35Z512xxx7 MCUs with up to 512 KB of Flash with emphasis on metering applications. For simplicity reasons, this document refers to these devices as KM34 and KM35. This document is focused on addressing the changes in functionality between these two Kinetis M MCUs.

## 2 Overview

The KM35 MCU is pin-to-pin compatible with the KM34 MCU in all packages available (100LQFP and 144LQFP). Almost all peripherals are the same on both devices. The main differences between the KM35 and KM34 MCUs are related to the memory and the peripheral instances, as shown in Table 1.

Table 1. Differences between KM34 and KM35

| Device | Flash | SRAM | SPI | LPTMR | LLWU |
|---|---|---|---|---|---|
| KM34 | 256 KB | 32 KB | 2 | 1 | 2 LLWU filters |
| KM35 | 512 KB<br>Dual bank | 64 KB | 3 | 2 | 4 LLWU filters |

### 2.1 Pinout considerations

The KM34 and KM35 MCUs are pin-to-pin compatible in all packages available (100LQFP and 144LQFP). The KM35 MCU supports extra SPI and LPTMR and two more LLWU filters. Therefore, the signals of the newly-added instances are not available for multiplexing on the KM34 MCU. The **bold** alternatives are only included in the KM35 MCU.

Table 2. KM35/KM34 pinout comparative

| Pin Name | ALT0 | ALT1 | ALT2 | ALT3 | ALT4 | ALT5 | ALT6 | ALT7 |
|---|---|---|---|---|---|---|---|---|
| PTB2 | LCD_P33 | PTB2 | **SPI2_PCS0** | | | | | LCD_P33 |
| PTB3 | LCD_P34 | PTB3 | **SPI2_SCK** | | | | | LCD_P34 |
| PTB4 | LCD_P35 | PTB4 | **SPI2_MISO** | | | | | LCD_P35 |
| PTB5 | LCD_P36 | PTB5 | **SPI2_MOSI** | | | | | LCD_P36 |
| PTC5 | ADC0_SE0<br>/CMP2_IN0 | PTC5<br>/LLWU_P12 | UART0_RTS_b | | **LPTMR1_ALT1** | | | |

*Table continues on the next page...*

Table 2. KM35/KM34 pinout comparative (continued)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PTJ4 | | PTJ4 | LPUART0_CTS_b | LPTMR1_ALT1 | | | | |
| PTJ7 | | PTJ7 | LPTMR1_ALT2 | | | | | |
| PTD3 | | PTD3 | UART1_CTS_b | SPI0_MOSI | LPTMR1_ALT2 | | | |
| PTK0 | | ADC0_SE12 | PTK0 | LPTMR1_ALT3 | | | | |
| PTD4 | ADC0_SE3 | PTD4 /LLWU_P9 | UART1_RTS_b | SPI0_MISO | LPTMR1_ALT3 | | | |
| PTG7 | LCD_P14 | PTG7 | | LPTMR1_ALT1 | | | | |
| PTH4 | LCD_P19 | PTH4 | | LPTMR1_ALT2 | | | | |
| PTH5 | LCD_P20 | PTH5 | | LPTMR1_ALT3 | | | | |

## 2.2 Flash memory

Both the KM35 and KM34 MCUs integrate a similar FTFA flash IP. All of the register addresses and fields are identical, so the flash-programming command and algorithm are the same.

The KM35 MCU supports 512/256 KB of program Flash with no FlexRAM, while the KM34 Flash supports 256 KB of program Flash with no FlexRAM. Besides the Flash size difference, the KM35 MCU also supports the dual bank RWW (read while write) feature, which is partitioned into:

- One 256 KB program Flash array divided into 2-KB sectors. The Flash addresses range from 0x0000_0000 to 0x0003_FFFF.
- One 256 KB program Flash array divided into 2-KB sectors. The Flash addresses range from 0x0004_0000 to 0x0007_FFFF.

Table 3. KM35 Flash memory map

| System Address Range | Memory Type | Size |
|---|---|---|
| 0x0000_0000 - 0x0003_FFFF | P-flash block 0 | 256 KB |
| 0x0004_0000 - 0x0007_FFFF | P-flash block 1 | 256 KB |

The KM34 Flash does not support dual bank. It only supports the 256-KB Flash, which is partitioned into:

- One 256-KB program Flash array divided into 2-KB sectors. The Flash addresses range from 0x0000_0000 to 0x0003_FFFF.

Table 4. KM34 Flash memory map

| System Address Range | Memory Type | Size |
|---|---|---|
| 0x0000_0000 - 0x0003_FFFF | P-flash | 256 KB |

The KM35 512-KB Flash with the dual bank feature enables you to download the new firmware to the inactive Flash block without disturbing the software image that is running. It allows simultaneous Flash operations. That is, while reading code from the Flash block 0, it allows the program sector to erase the Flash block 1, as shown in Table 5.

Table 5. KM35 allowed simultaneous memory operations

| | | P-Flash block 0 | | | P-Flash block 1 | | |
|---|---|---|---|---|---|---|---|
| | | Read | Program | Sector Erase | Read | Program | Sector Erase |
| P-Flash block 0 | Read | - | | | | OK | OK |
| | Program | | - | | OK | | |
| | Sector Erase | | | - | OK | | |
| P-Flash block 1 | Read | | OK | OK | - | | |
| | Program | OK | | | | - | |
| | Sector Erase | OK | | | | | - |

## 2.3 SRAM memory

Compared to the KM34 MCU, the KM35 MCU increases the SRAM size from 32 KB to 64 KB with expanded SRAM addresses, that is, with different start and end addresses, as shown below. Take care of the linker file to make sure that the definition of the SRAM address range is valid when migrating from KM34 to KM35.

Table 6. KM35 SRAM memory map

| System Address Range | Memory Type | Size |
|---|---|---|
| 0x1FFF_C000- 0x2000_BFFF | SRAM | 64 KB |

Table 7. KM34 SRAM memory map

| System Address Range | Memory Type | Size |
|---|---|---|
| 0x1FFF_E000 - 0x2000_5FFF | SRAM | 32 KB |

## 2.4 Peripheral enhancements

There are several improvements on the KM35 MCU when compared to the KM34 MCU:

- Enhanced OSC32K monitor scheme to improve EMC performance to avoid hang-up during EMC test.
- One new SPI module.
- One new LPTMR module.
- Two more LLWU digital filters to improve noise immunity and ESD performance.
- FIFO in LPUART, added to improve throughput and reduce CPU load.

### 2.4.1 OSC32K improvement

On the KM34 MCUs, the BUS hangs up if the RTC register is accessed when the OSC32K is not ready. The KM35 MCUs feature the enhanced OSC32K monitor scheme to improve EMC performance to avoid hang-up during the EMC test. The detailed changes include:

- Added SIM_MISC_CTL[13:12] bits to indicate whether OSC32K is ready.
- Added hard fault triggered by accessing the RTC register when the RTC OSC32K (VBAT domain) is not ready.

There is no hardware change needed when you migrate from KM34 to KM35. From the software perspective, the OSC32K improvement is covered by the SDK software and the pre-SDK driver. The demo code is available in SDK CLOCK_CONFIG_EnableRtcOsc(). If you want to develop the drive on your own, add a delay in the software to wait for the OSC32K to be ready by checking the SIM_MISC_CTL[13:12] bits. There is a hard fault to remind you that it is not ready instead of the MCU hang-up.

### 2.4.2 New SPI module

The KM34 MCU offers two SPI modules, while the KM35 MCU offers three SPI modules. The pinout of the existing SPI 0/1 is exactly the same as on the KM34 MCU. It enables you to port your existing SPI-related code without any hardware changes and you can use the extra SPI module for communication.

The detailed changes include:

- New SPI module.
- Added registers set for SPI2 which shares the same interrupt vector with SPI0/SPI1.
- Added SPI2 signals on pinouts.
- Added SPI2 clock gating in SIM.

For software considerations, when porting software from KM34 to KM35, the interrupt vector name of the SPI module is changed to "SPI0_SPI1_SPI2_IRQHandler" in the *SDK startup_MKM35Z7.s* file.

### 2.4.3 New LPTMR module

The KM34 MCU offers one LPTMR module, while the KM35 MCU offers two LPTMR modules. The detailed changes include:

- New LPTMR module.
- Added register set for LPTMR1 which shares the same interrupt vector with LPTMR0.
- Added LPTMR1 signals on pinouts.
- Added LTMR1 clock gating in SIM.
- Added LPTMR1 as XBAR input source and AWIC wakeup.

When porting software from KM34 to KM35, remember that the interrupt vector name of the LPTMR module is changed to "LPTMR0_LPTMR1_IRQHandler" in the SDK *startup_MKM35Z7.s* file.

### 2.4.4 New LLWU digital filters

The KM34 MCU offers two LLWU digital filters, while the KM35 MCU offers four LLWU digital filters to improve the noise immunity and ESD performance. The existing two filters are exactly the same on the KM34 and KM35 MCUs. For the two new filters, the KM35 has two new registers of LLWU_FILT3/4.

### 2.4.5 New FIFO in LPUART

There is no FIFO implemented in the LPUART on the KM34 MCU to improve the throughput and reduce the CPU load during the serial communication. The KM35 MCU has an 8-bit FIFO in the LPUART. This improvement is transparent to your software and hardware design.

## 3 SDK considerations

The MCUXpresso SDK has open-source drivers, middleware, and reference example applications to speed up your software development. You can visit the MCUXpresso web page (mcuxpresso.nxp.com) and download the SDK by searching for "KM34" or "KM35".

The KM34 SDK was released two years ago with version SDK 2.2.0 and it has not been updated since then. The KM35 SDK is based on the latest SDK, so there are some changes (mainly in the APIs). When porting from the KM34 SDK to the KM35 SDK, pay attention to the changes listed in Table 8.

Table 8. Important changes

| Difference | Drivers | Details | Migration |
|---|---|---|---|
| Drivers with API changes | GPIO | Refined naming of API. The main change is the update API with prefix of _PinXXX() and _PortXXX. | Replace the API names if they use the KM34 SDK and move to the KM35 SDK. |
| Drivers with files structure change | debug_console | Added the serial manager and UART wrapper layer in the *components* folder | Add new files into the project and include the related *.h* and *.c* files. |
| | | The parameters of DbgConsole_Init changed<br>From: status_t DbgConsole_Init(uint32_t baseAddr, uint32_t baudRate, uint8_t device, uint32_t clkSrcFreq)<br>to : status_t DbgConsole_Init(uint8_t instance, uint32_t baudRate, serial_port_type_t device, uint32_t clkSrcFreq)<br>Some “defined symbols” also need adjustment, such as adding “SERIAL_PORT_TYPE_UART=1” for the debug console to use the UART. | |
| | Flash | Reorganize FTFx Flash driver source file, extract Flash cache and flexnvm driver from FTFx driver. | Add new files into the project and include the related *.h* and *.c* files. |
| | | Some enumeration and struct names’ prefix changed from from “flash” to “ftfx”. | Replace the names if using the KM34 SDK Flash driver and move to the KM35 SDK Flash driver. |
| Drivers with new feature added | SPI slave | SPI slave config structure adds an element of “pinMode” and added SPI pin mode(Bidirectional mode) for transfer set API. | Optional |

The overall migration consideration is that only one driver was changed in the API (GPIO) and two drivers/components were changed in the structure. Replace the API names that use the KM34 SDK, move to the KM35 SDK, and add the extra files into the project (the included files may need updates). The change logs of the driver are provided in the Release Notes to show you the driver’s change history. Other new features/bug fixes are transparent for the migration from the old SDK (2.2.0) to the latest SDK.

# 4 Hardware kit TWR considerations

The KM3x family comes with fully compatible boards and parts. The TWR-KM35Z75M module for KM35 is fully compatible as a superset to the previous TWR-KM34Z75M module for KM34.

Date of release: 05/2020

Document identifier: AN12847