Freescale Semiconductor
Application Note

Document Number: AN4905
Rev 1, 10/2014

Crystal-less USB operation on Kinetis MCUs

by: Melissa Hunter

1 Introduction

Some of the newer Kinetis devices include a new feature that allows for USB full-speed device operation with no external crystal or oscillator required for the processor. Elimination of the external clock reduces the overall system cost associated with developing a USB device application. In order to support the crystal-less USB device operation, a 48 MHz internal reference clock (IRC) and a USB clock recovery circuit have been included on these microcontrollers.

This application note will discuss the features of the 48 MHz IRC and the USB clock recovery circuit, including how they can be used to implement a full-speed USB device without a crystal. Also covered will be the initialization required to configure the USB to use the 48 MHz IRC as the clock source, and enabling of the clock recovery.

2 On-chip circuitry to support crystal-less USB operation

The following sections describe the features and use of the new 48 MHz IRC and the USB clock recovery block that have been added to allow for crystal-less USB operation.

Contents

2.1 48 MHz Internal Reference Clock (IRC48M)

The USBOTG module requires a 48 MHz clock input, so the first step to having USB device operation without an external clock source is to add an internal clock source capable of generating the 48 MHz clock. Some Kinetis MCUs include a 48 MHz internal reference clock (IRC) for this purpose. The IRC is factory trimmed to 48 MHz, but the frequency accuracy of the internal clock can vary. Even with the factory trim the clock frequency can vary by as much as +/-1.5%. This frequency variation is too high to guarantee correct operation of USB. The USB clock recovery circuit is used to solve this problem.

2.2 USB clock recovery

The USB clock recovery circuit monitors the start-of-frame (SOF) packets that are sent by the USB host approximately every 1ms. The clock recovery circuit compares the actual time that the SOF from the host arrives to the expected SOF arrival time based on the IRC48M. The clock recovery circuit uses the delta time between the actual and expected SOF to update the 48 MHz IRC trim value. With the USB clock recovery circuit active, the IRC48M is automatically adjusted to match the USB host's clock. The matching between the host clock and the internal clock source allows for correct USB device operation even though the nominal untuned accuracy of the 48 MHz IRC does not meet USB requirements.

3 Configuring the MCU to use USB without a crystal

The following sections describe the initialization of the IRC48M and the USB clock recovery circuits for crystal-less USB device operation, handling USB clock recovery errors, and configuring the clocks for the core and system.

3.1 USB device mode clocking initialization using 48 MHz IRC

Follow the steps below to configure the USB module to use the 48 MHz IRC as its clock source and enable the clock recovery feature.

1. Enable the peripheral clock to the USB module by setting the SIM_SCGC4[USBOTG] bit.
2. Select the 48 MHz IRC as the optional peripheral clock by setting the SIM_SOPT2[PLLFLLSEL] field to 0x3.
3. Configure the USB clock source to use the clock selected by the PLLFLLSEL field instead of the USB_CLKIN pin by setting the SIM_SOPT2[USBSRC] bit.
4. The USB clock passes through a programmable fraction divider controlled by the SIM_CLKDIV2 register. The default value of this divider is the desired divide by one, so this register should not need to be modified.
5. Enable the 48 MHz IRC clock (set USB_CLK_RECOVER_IRC_EN[IRC_EN]).
6. Enable the USB clock recovery feature by setting USB_CLK_RECOVER_CTRL[CLOCK_RECOVER_EN].
7. Continue with regular USB device mode initialization.

3.2 USB clock recovery errors

The USB clock recovery has an overflow error bit (USB_CLK_RECOVER_INT_STATUS[OVF_ERROR]). If this bit sets it indicates that the clock tuning algorithm is not able to adjust the IRC48M frequency, because the required adjustment exceeds the fine trim limits.

During normal USB communication overflow errors should not be seen. If errors occur, they indicate a serious problem with the host clock and/or the USB signals reaching the Kinetis device. In the rare event that an error occurs follow this sequence:

1. Disconnect from the USB bus. Abort any queued transmissions and disable the pullup on the D+ pin by clearing USB_OTGCTL[DPHIGH] or USB_CONTROL[DPPULLUPNONOTG].
2. Reset USB module and the IRC48M by setting USB_USBTRC0[USBRESET].

3. Re-initialize the IRC48M, USB clock recovery (follow the steps in the initialization section).

After performing the re-initialization, the clock recovery should resume. If the overflow condition is detected three times in a row after performing the steps above between each overflow, then this signals a catastrophic error. If this occurs, USB connection attempts should be discontinued.

3.3 System clocking

In a true crystal-less USB system where there is no external clock source for the MCU at all, the generation of the core and system clocks for the rest of the processor needs to be considered. The Multipurpose Clock Generator (MCG) module is responsible for generating the MCGOUTCLK which passes through the dividers configured by SIM_CLKDIV1 to provide the clocks to the core/system, bus, FlexBus, and flash.

There are three internal clock sources available to the MCG that can be used to generate the MCGOUTCLK-- IRC48M, the fast clock reference (nominally 4 MHz), and the slow clock reference (nominally 32.768 kHz). The fast and slow clock references are both clock sources internal to the MCG and are used for the MCG internally clocked modes. Even though the IRC48M is internal to the MCU, it is an external input to the MCG module (as OSCCLK1), so the IRC48M would be used with MCG externally clocked modes.

The table below lists the internal clock sources that are available and the clock modes that could be used with each source to generate the core/system clock. The core/system clock must be at least 20 MHz when the USB is being used, so some of these modes should only be used as a low power option when the USB connection is not active. Transitional modes that would be used when moving between the modes listed instead of being used for extended periods of time are not included.

Table 1. Core/system clock generation options when system has no external clock source available

Internal clock source used	MCG clock mode	MCGOUTCLK nominal frequency	Comments
IRC48M	PLL Engaged External (PEE)	48 - 120 MHz	IRC48M clock is divided down to 2 MHz or 4 MHz (using PRDIV0) to get a valid input to the PLL, then multiplied up (using VDIV0).
	FLL Engaged External (FEE)	20 - 80 MHz	IRC48M is divided by 1536 to get a 31.25kHz input to the FLL (FRDIV), then multiplied by the FLL factor (DMX32).
	Bypassed Low Power External (BLPE)	48 MHz	IRC48M is used directly as MCGOUTCLK.
Fast reference	Bypassed Low Power Internal (BLPI)	4 MHz	Fast reference is used directly as MCGOUTCLK.
Slow reference	FLL Engaged Interal (FEI)	24 - 96 MHz	Internal slow reference is used as the input to the FLL, then multiplied by the FLL factor (DMX32). This is the processor's default clock mode.
	Bypassed Low Power Internal (BLPI)	32.768 kHz	Slow reference is used directly as MCGOUTCLK.

NOTE

On some MCUs, if the IRC48M is being used to generate the core/system clock or peripheral clocks, then you should not set the USB_TRC0[USBRESET] bit. This will disable the IRC48M, and you'll lose your core and/or peripheral clocks. Refer to the next section for more details.

NOTE

If the IRC48M clock is being used to generate the core/system clocks, then the clock frequency of the system clocks can shift as the USB clock recovery tuning occurs.

NOTE

If your system does include an external 32.768 kHz crystal for the RTC, then you can run in FEE or BLPE modes using the RTC oscillator as the external clock source. The RTC oscillator cannot be used as an input to the PLL, because the PLL requires an input in the 2-4 MHz range.

4 Changes to the crystal-less USB implementation

The K64, K63, and K24 family devices with 1 MByte of flash were the first Kinetis MCUs to implement the crystal-less USB feature. Some improvements to the implementation have been made on newer devices that make the 48 MHz IRC more independent from the USB module.

Originally, the 48 MHz IRC could only be enabled by setting the USB_CLK_RECOVER_IRC_EN[IRC_EN] bit. This bit would reset any time the USB module is reset which can cause issues if the IRC48M is being used to generate the system clock. On new devices the 48 MHz IRC is enabled if any of the following are true:

- USB_CLK_RECOVER_IRC_EN[IRC_EN] = 0x1
- MCG_C7[OSCSEL] = 0x2
- SIM_SOPT2[PLLFLLSEL] = 0x3

5 Special considerations

Below is a list of special cases and things to keep in mind when using the 48 MHz IRC and USB clock recovery features:

- The 48 MHz IRC should not be used as the USB module clock source for host mode operations. The USB clock recovery circuit requires incoming SOFs from a host to function. Without the USB clock recovery the 48 MHz IRC is not accurate enough to meet USB requirements. When using the USB module in host mode, an 8-12 MHz external clock sourcing the PLL or an external 48 MHz clock used directly as the USB_CLKIN can be used as the USB module clock source.
- The 48 MHz IRC can only be used while the processor is in RUN or WAIT mode. Before entering any low power modes including VLPR and VLPW, the IRC should be disabled by software. Software will also need to re-enable the 48 MHz IRC after exiting the low power mode in order to use it again.

6 Further reading

For more information, see the following documents available on www.freescale.com.

- The Clock Distribution, System Integration Module (SIM), and Universal Serial Bus Full Speed OTG Controller (USBFSOTG) chapters of the applicable device's reference manual
- Data sheet for the applicable Kinetis device
- The Universal Serial Bus Revision 2.0 specification

7 Revision history

Revision	Date	Changes
0	3/2014	Initial release.
1	10/2014	Changed description to apply to all Kinetis devices with crystal-less USB instead of just K64, K63, and K24. Added new section describing the changes to the implementation.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number AN4905
Revision 1, 10/2014

freescale™