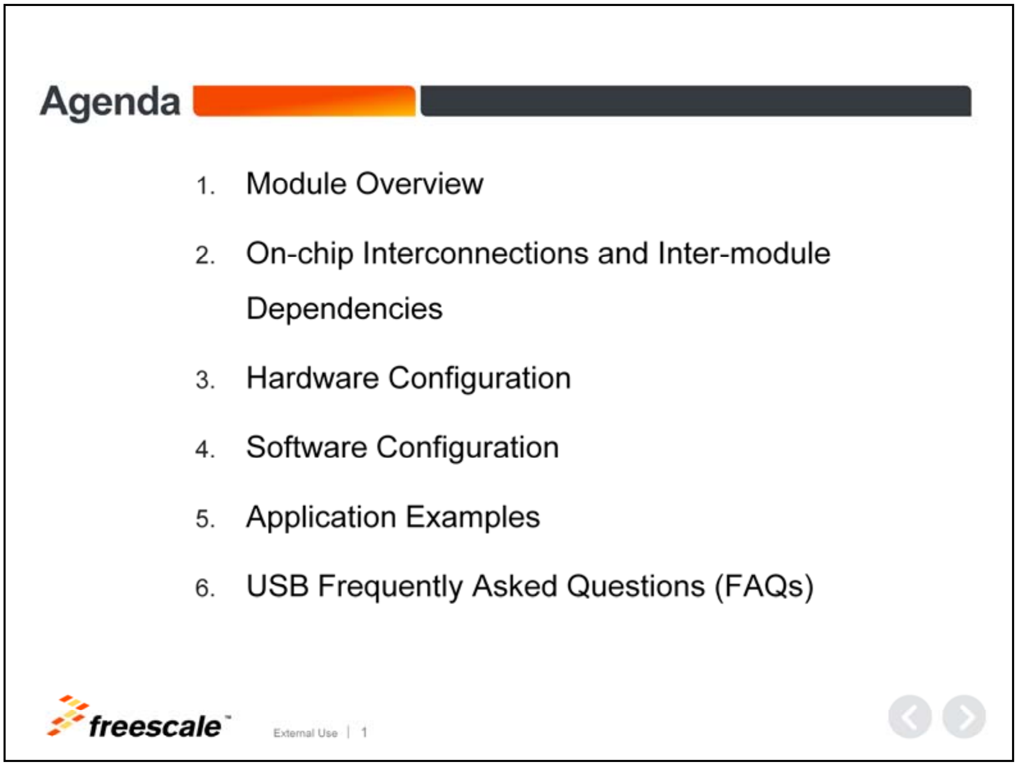
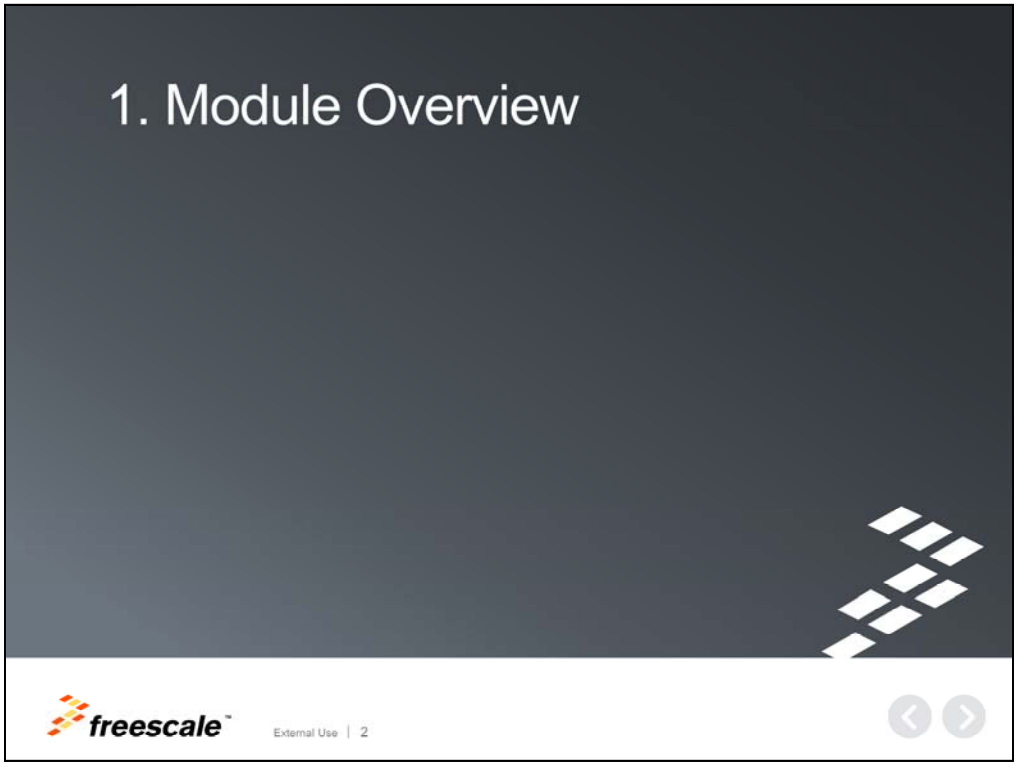
Hello and welcome to this presentation of the Universal Serial Bus, or USB module, for Kinetis MCUs. In this session, you will learn about USB 2.0 full-speed and low-speed, the main features of the USB module, and the application benefits of leveraging this function.

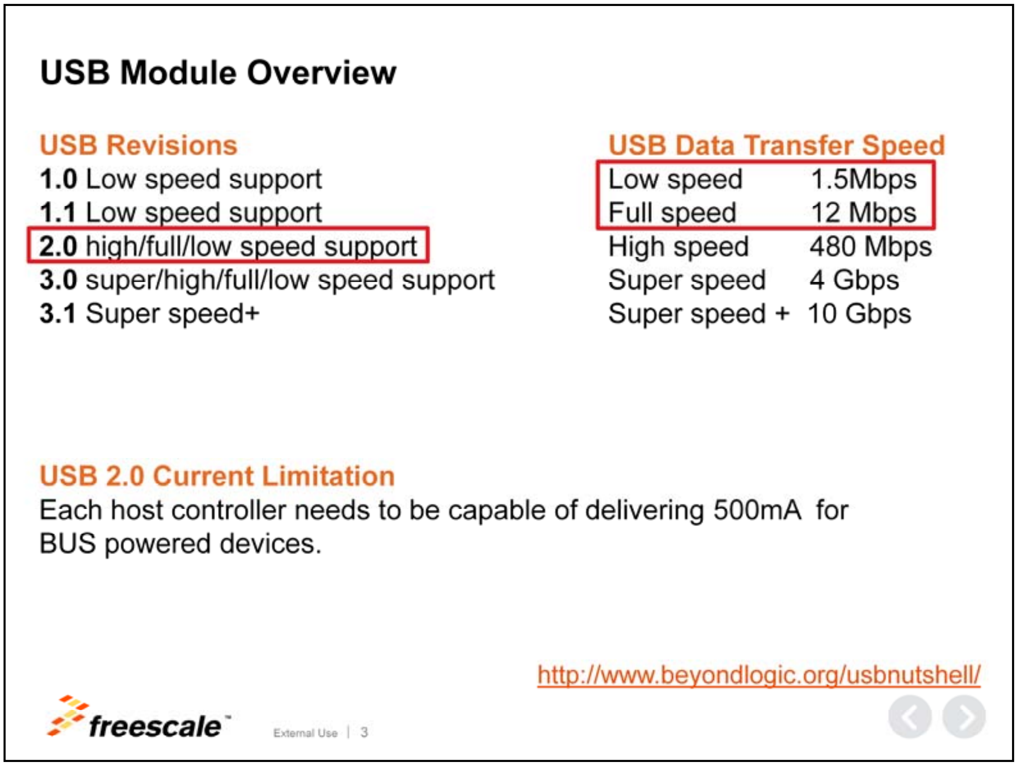


In this presentation, we'll cover:

- An overview of the module
- The on-chip interconnections and inter-module dependencies
- Hardware and software configuration
- Application examples
- And a few frequently asked questions



We first begin with an overview of the module

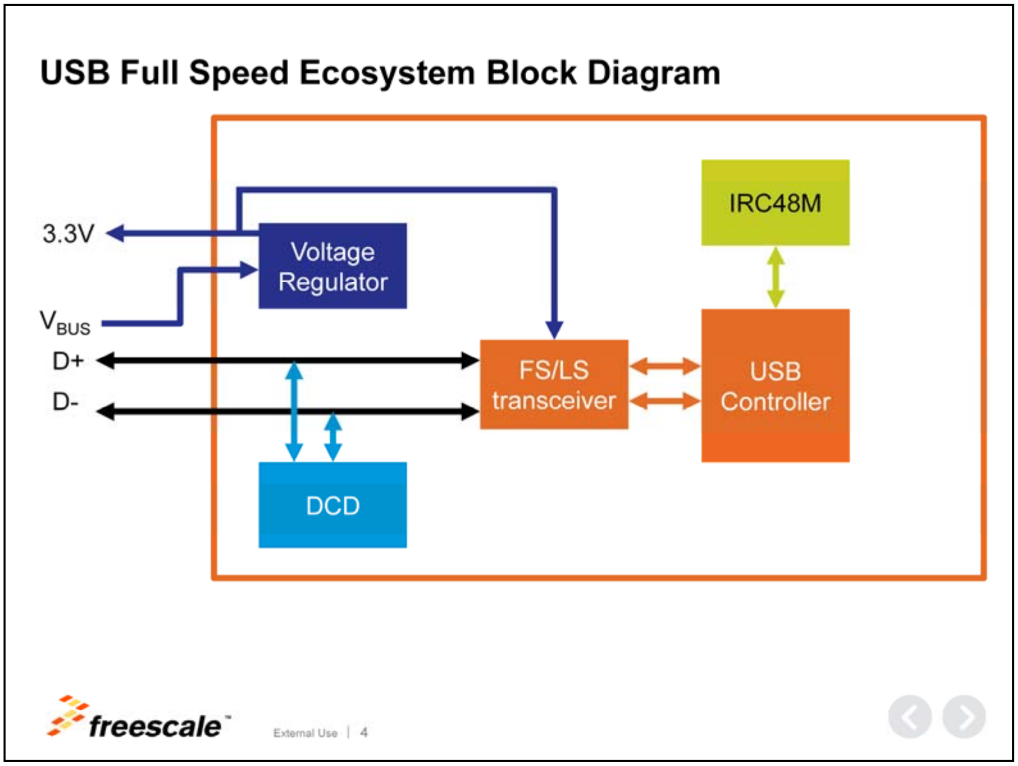


USB Module Overview

There are different revisions for the USB specifications. The data transfer speed is one of the main differentiators in determining the revisions for USB.

This training will focus on the USB 2.0 module that supports full-speed and low-speed.

USB 2.0 specifies that each host needs to be capable of delivering 500mA for VBUS power devices.



USB Full Speed Ecosystem Block Diagram

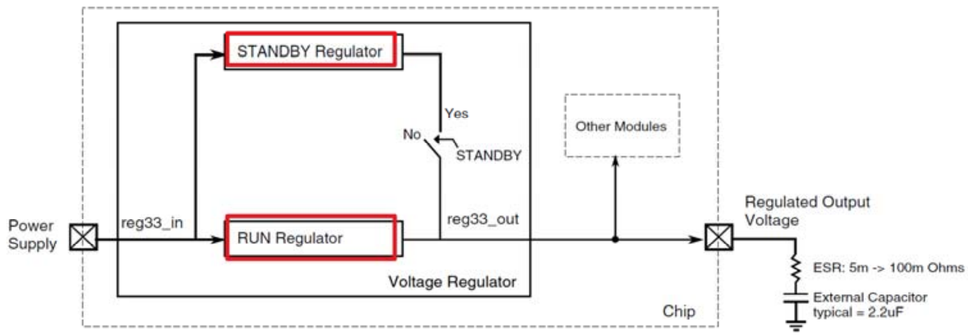
The USB full-speed ecosystem is an integration of three different modules.

- This includes an internal 5V to 3.3V USB regulator that powers the USB transceiver or the MCU.
- The USB module is full-speed, dual-role, USB on the go capable. The module complies with USB 2.0 specifications for USB device charger detection or DCD module and works with the USB transceiver to detect if the USB device is attached to a charging port. This module is also compliant with USB battery charger revision 1.1.
- Some Kinetis devices come with a 48MHz internal oscillator. This IRC48M provides an internally generated clock source for the USB controller.

USB Voltage Regulator

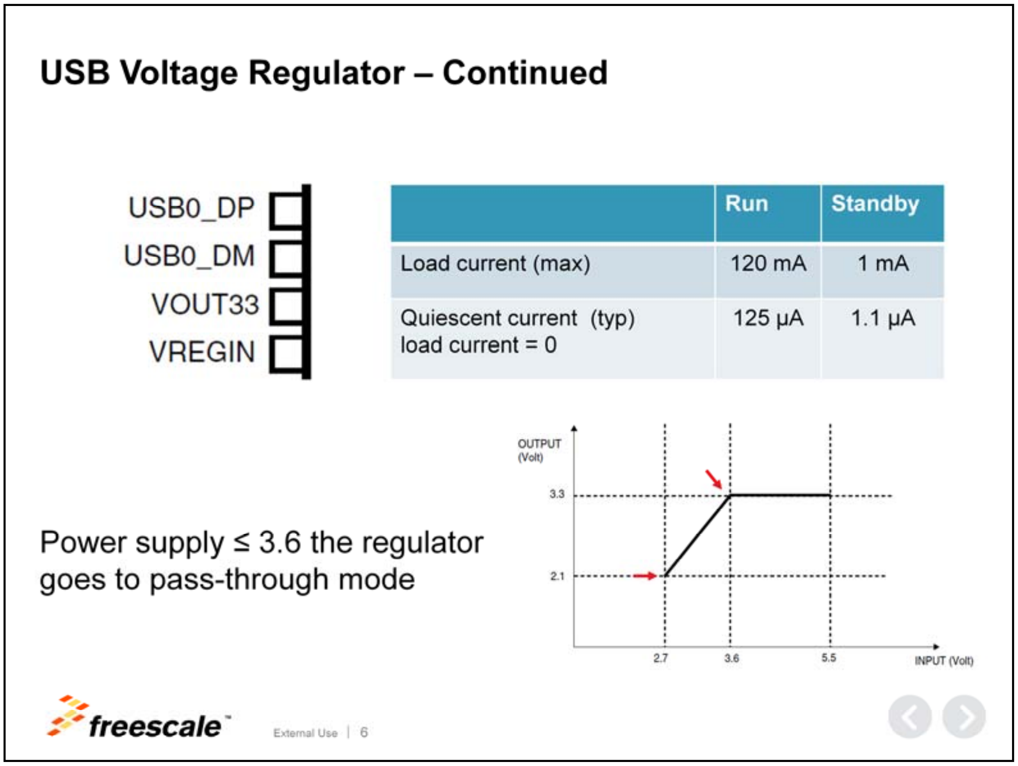
Uses two regulators

1. Run
2. Standby



USB Voltage Regulator

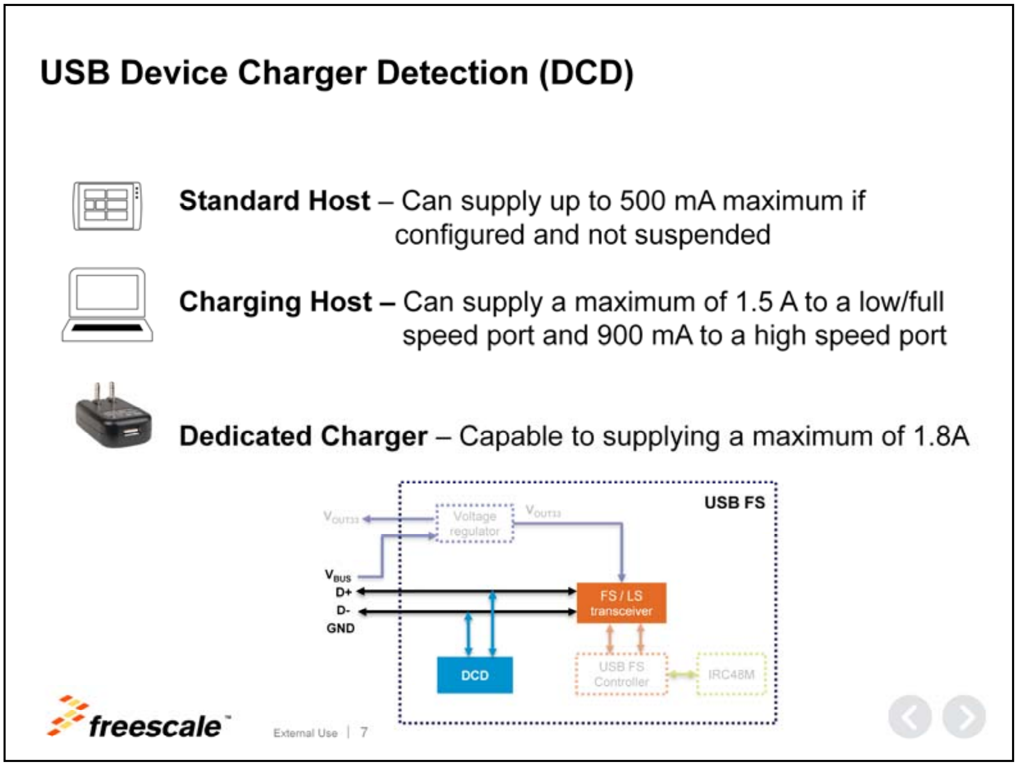
The USB voltage regulator uses two regulators in parallel. A run regulator and a standby regulator. You can select which regulator will be used by using the configuration bits at the system integration module, or SIM.



USB Voltage Regulator – Continued

The input pin of the regulator is called VREGIN, and the output pin is VOUT33. When the USB regulator is in ’run’ mode, VOUT33 load current can be up to 120mA with a typical quiescent current without load of 125µA.

This module is an on-chip LDO linear voltage regulator, that provides 3.3V power from an input varying from 2.7V to 5.5V. The regulator is enabled by default, this means that once the power supply is provided, the module power-up sequence to run mode starts. When the input power supply is below 3.6V, the regulator goes into pass-through mode, which means that the output voltage is the same as the input voltage.

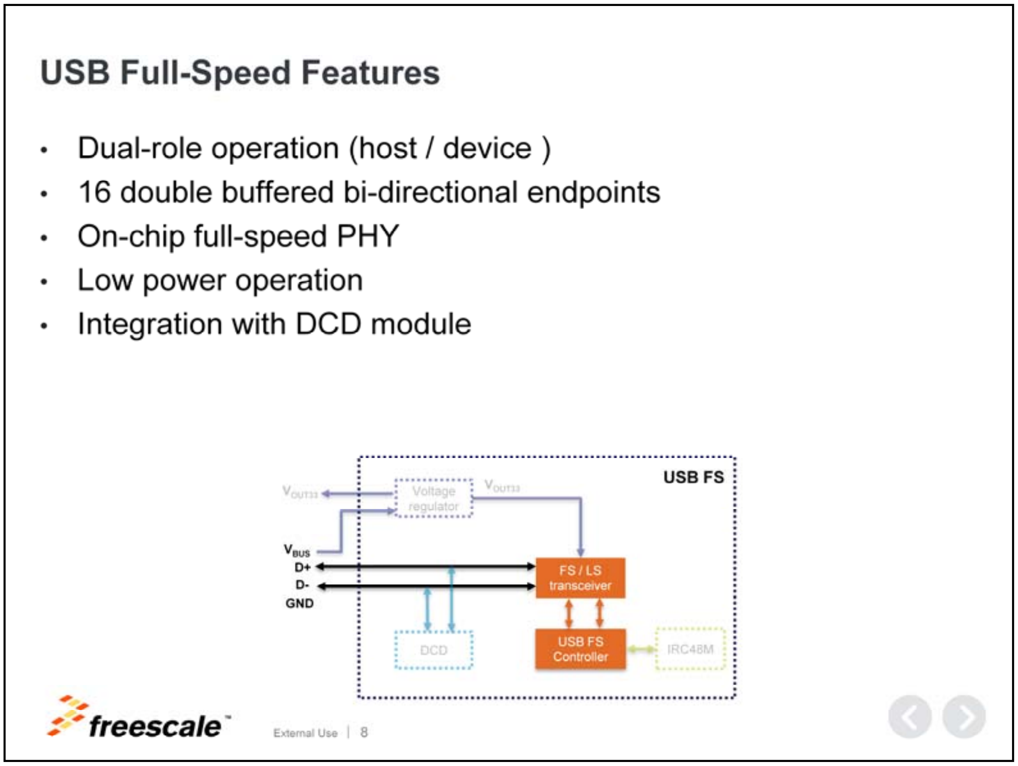


USB Device Charger Detection (DCD)

The USB DCD module works with the USB transceiver to detect if the USB device is attached to a charging port. This module is compliant with the USB battery charger specification version 1.1. The USB battery charger specification defines limits, detection, control, and reporting mechanisms that permit devices to draw current in excess of the USB 2.0 specification.

The battery charger specification establishes three different types of downstream ports:

- Standard downstream port refers to a downstream port on a device that complies with USB 2.0 definition of a host or hub. This port expects a downstream device to draw less than 2.5mA average when disconnected or suspended, and up to 500mA if configured and not suspended.
- Charging downstream port is a port compliant with USB 2.0 definition of a host or a hub, but can supply a maximum of 1.5A to a low- or full-speed port and 900mA to a high-speed port.
- Finally a dedicated charging port is a downstream port on a device that outputs power through a USB connector, but is not capable of enumerating a downstream device. It is able to supply a maximum 1.8A. A dedicated charging port is required to short D+ line to D-.

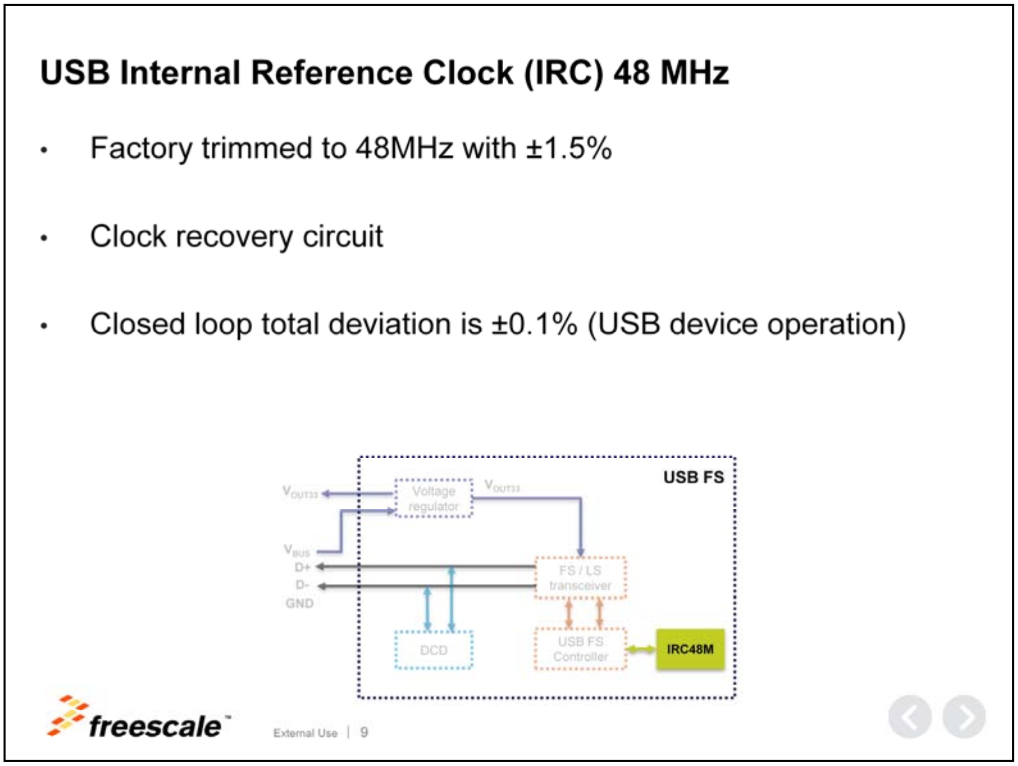


USB Full Speed Features

The USB full-speed subsystem is able to support a dual-role USB on-the-go capable controller that supports a full-speed device, full-speed host or low-speed host.

There is no need for an external PHY because it includes a USB transceiver with internal pulldowns on D+ and D- lines for host functionality. Internal connection to the USB voltage regulator provides a power source to the PHY.

Because Kinetis IOs are 5V tolerant, is possible to detect a valid BUS using a GPIO signal that can wake up the chip in all low-power modes.



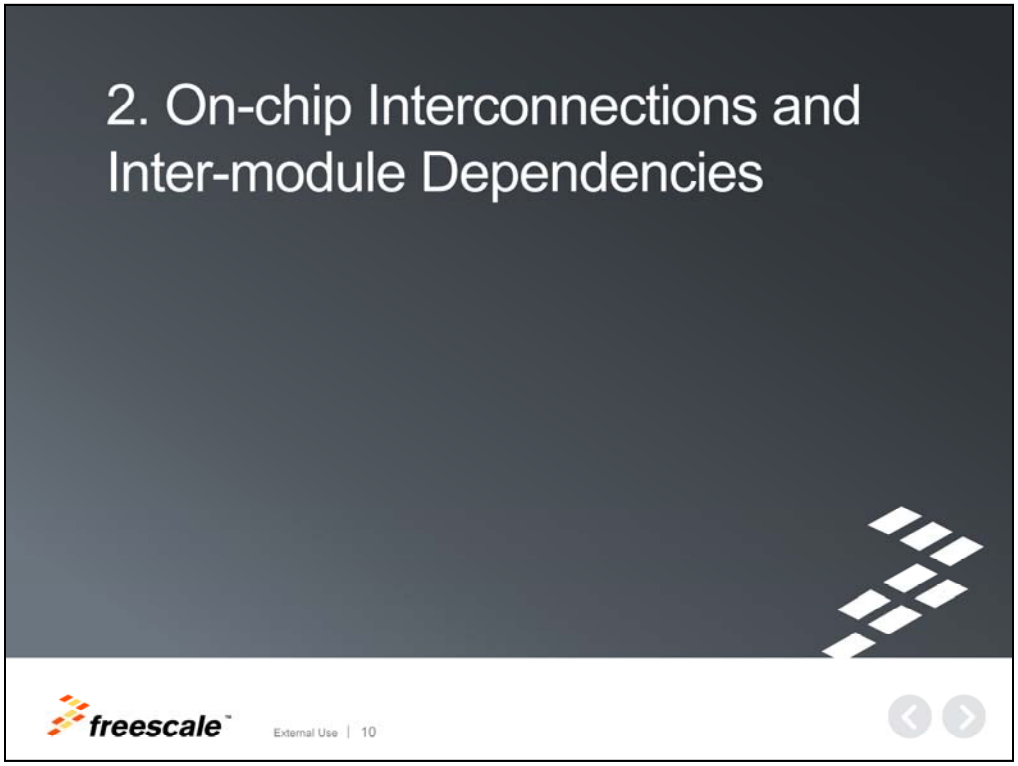
USB Internal Reference Clock (IRC) 48 MHz

Some 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.

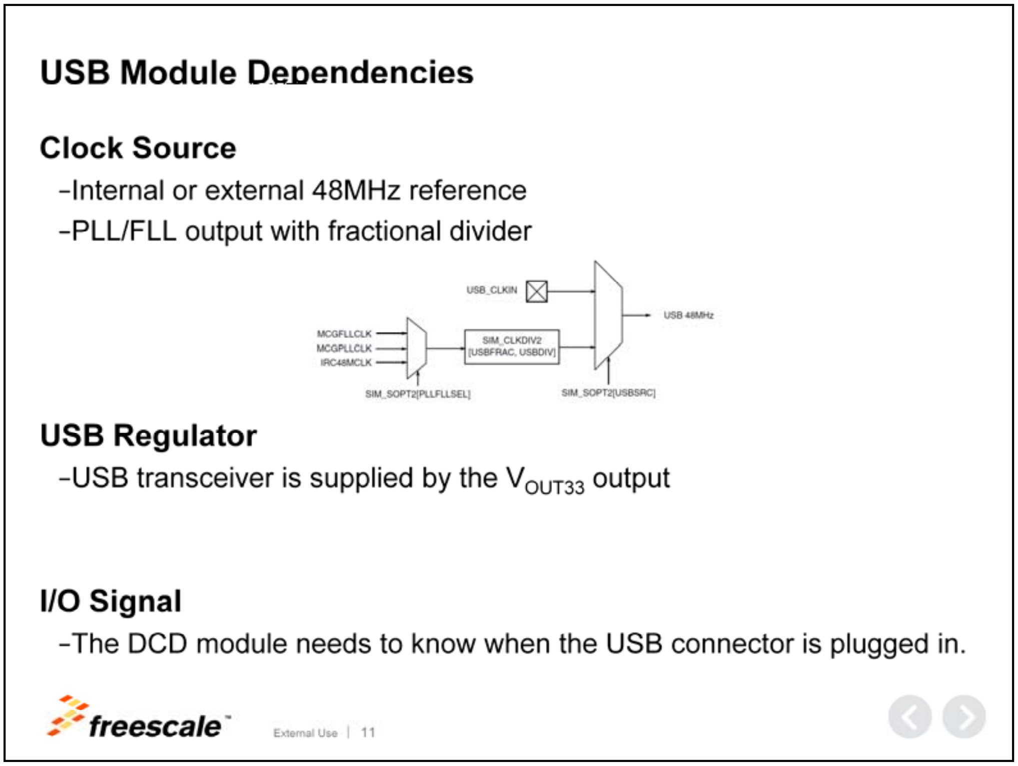
The IRC48M is factory trimmed to 48MHz with a +/- 1.5 percent deviation.

The clock recovery circuit compares start of frame received from the USB host vs the expected start of frame arrival time based on the IRC48M.

The clock recovery circuit uses the delta time between the actual and expected start of frame to update the 48 MHz IRC trim value. The matching between the host clock and the internal clock creates a closed loop total deviation of +/- 0.1 percent.



Now, let's discuss on-chip interconnection and inter-module dependencies.



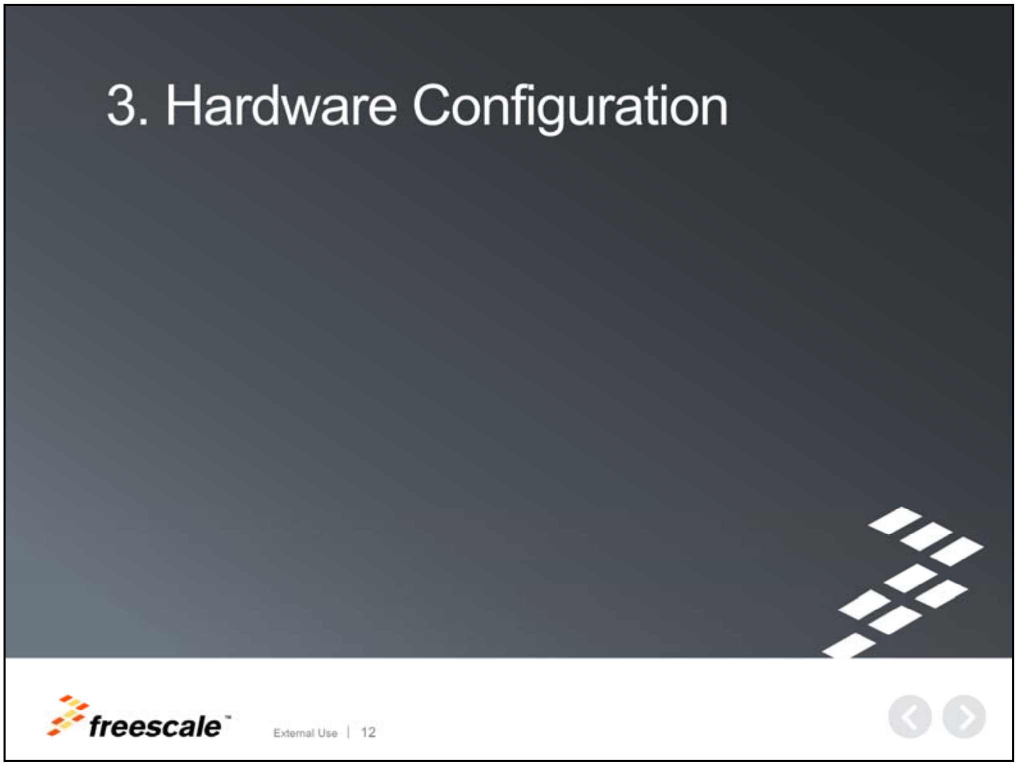
USB Module Dependencies

The USB module needs a 48 MHz clock to operate. There are four possible sources for the USB clock: PLL, FLL, the internal 48 MHz oscillator, and an external pin called USB_CLKIN.

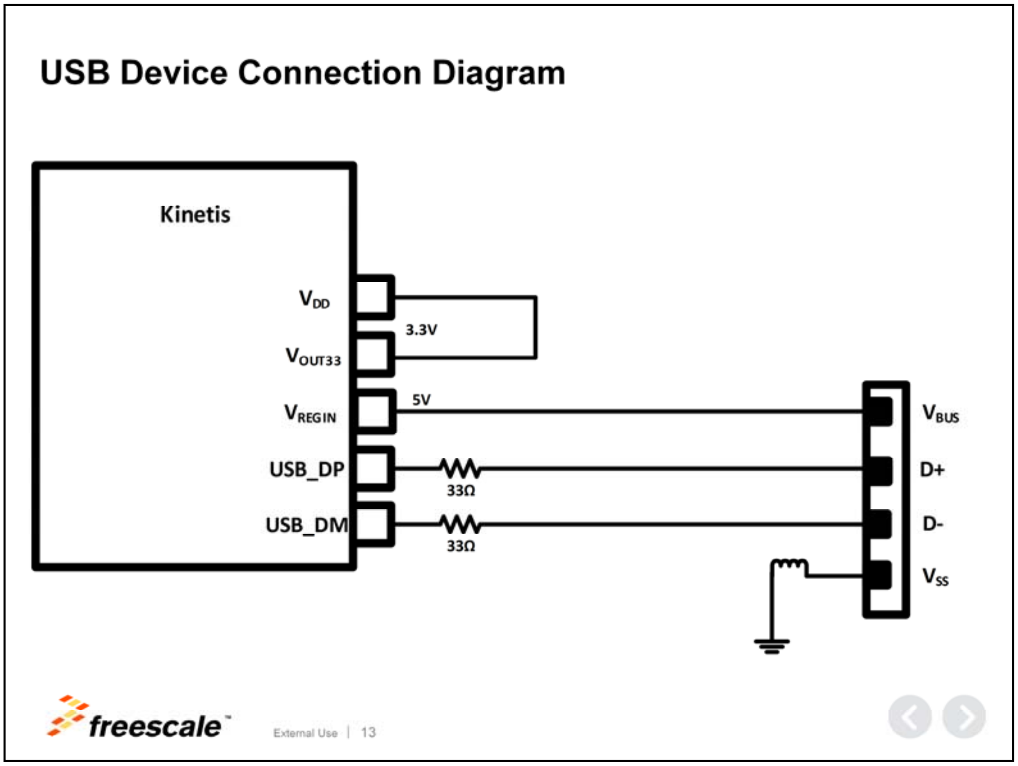
For PLL or FLL source clocks, there is a fractional divider to enable the MCU to operate at frequencies higher than 48 MHz.

The USB voltage regulator is the only power supply for the USB PHY. This regulator works completely independent of the MCU, and can be enabled even if the MCU is not powered.

The device charger detect algorithm can use one of the Kinetis 5V tolerant inputs connected to VBUS, in order to sense when a host is connected.



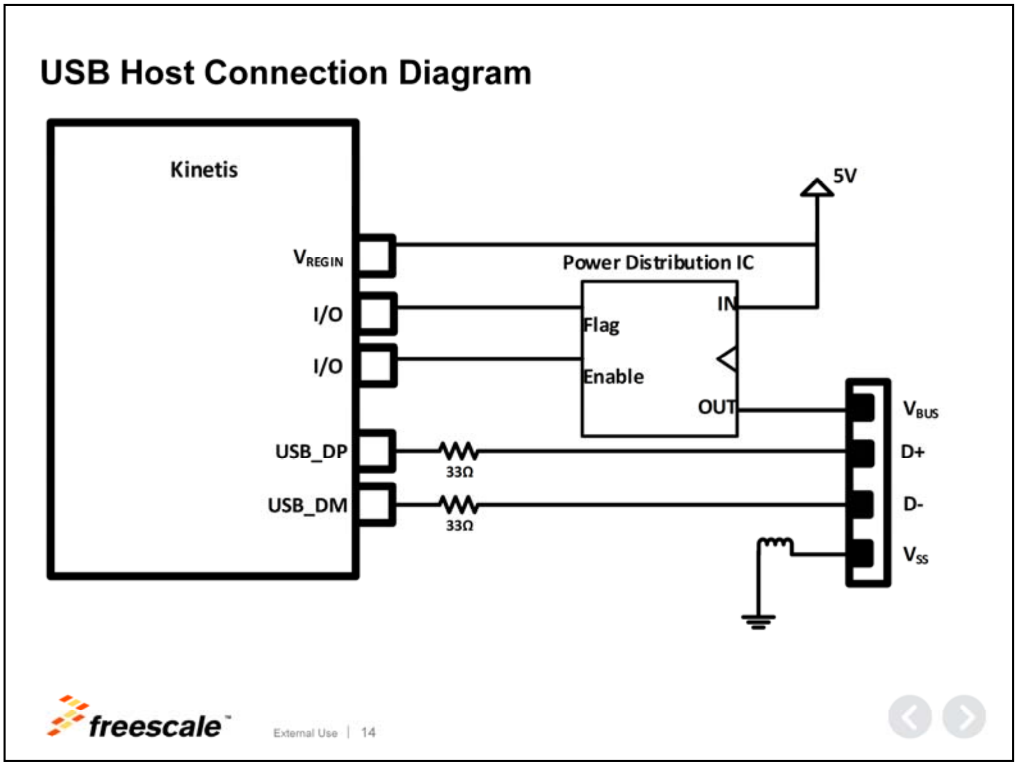
Now, let’s move on to hardware configuration.



USB Device Connection Diagram

In many cases the USB device just needs to communicate with an application running on a PC or a tables with USB 2.0 port. For these type of use cases, the MCU will act as a USB device. The device can be self powered using an external power supply or can be powered by 5V coming from host VBUS pin. USB 2.0 request the D+ and D- signals to be routed to the MCU differential pair input.

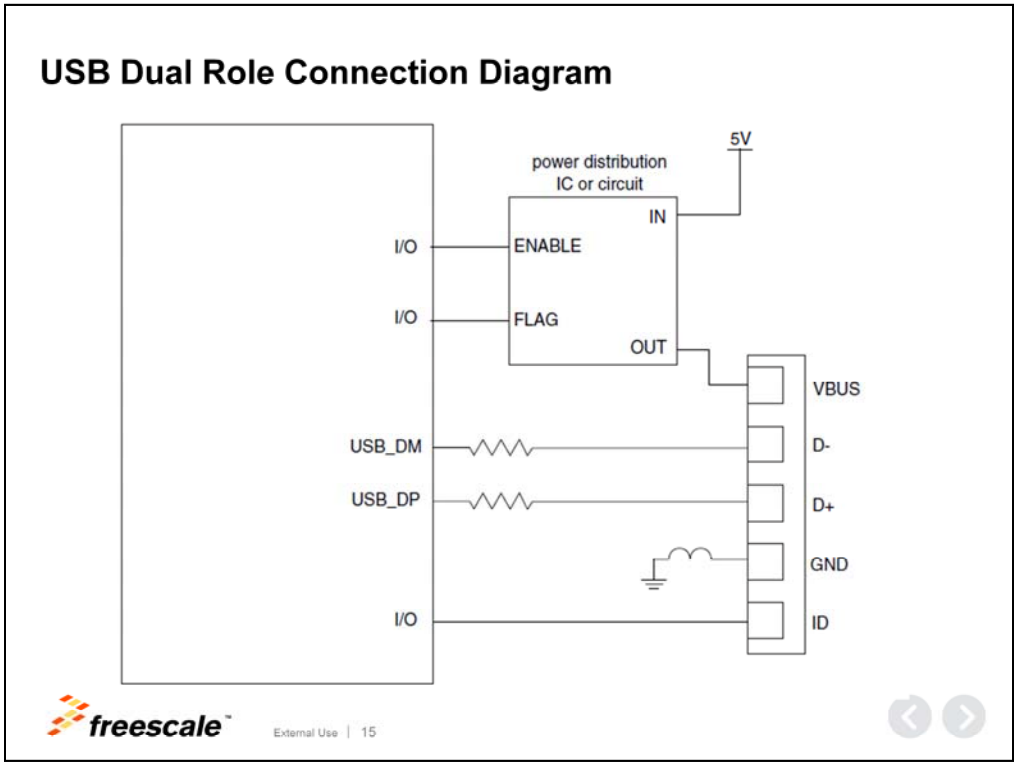
The USB regulator output can be used to supply the MCU voltage



USB Host Connection Diagram

If the USB device is acting as a USB host then it is not necessary to include the ID line in the hardware. However as USB device is a host, the system must provide 5V with enough current to supply the device side. This voltage is typically provided by an external IC controlled by the MCU.

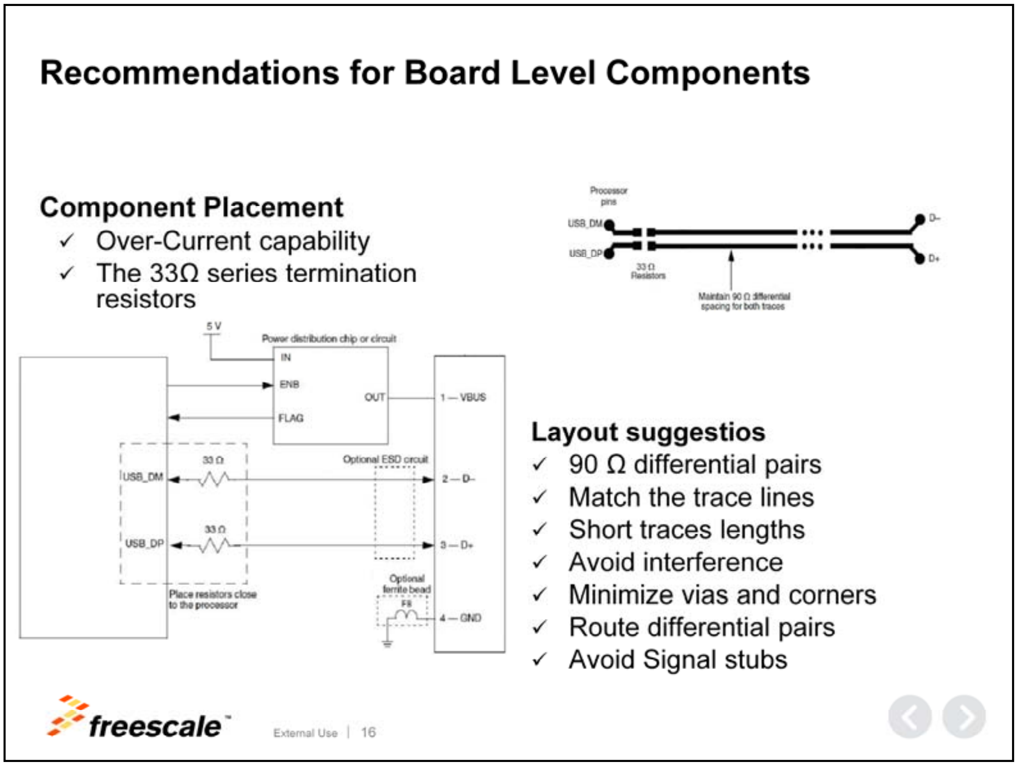
Like the device mode the D+ and D- lines need to be connected.



USB Dual Role Connection Diagram

This mode is used when the application can be connected to a PC or is able to handle external USB devices, such as fingerprint readers, mice, USB flash drives, and so on.

The application running on the MCU will be configured in device mode until the ID signals become low. This indicates that a host mode reconfiguration is needed, and 5V is then applied to the VBUS signal using the external IC.



Recommendations for Board Level Components

Any external power management chip must have an over-current detection capability to be compliant with the USB standard.

The 33 ohm series termination resistors are recommended for the transceiver. These resistors must be placed as close as possible to the transceiver to maximize the eye diagram for the data lines.

Route the USB D+ and D- signals as parallel 90Ω differential pairs.

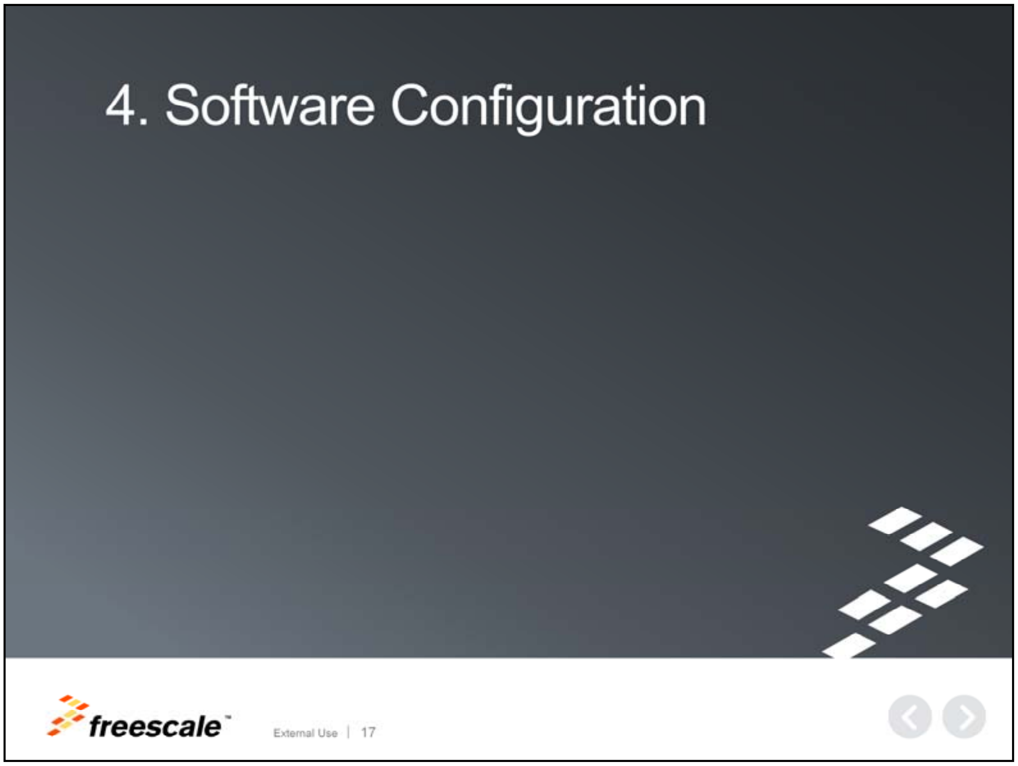
Match the trace lengths as closely as possible. Matching within 150 mils is a good guideline.

Try to maintain short trace lengths, not longer than 15 cm.

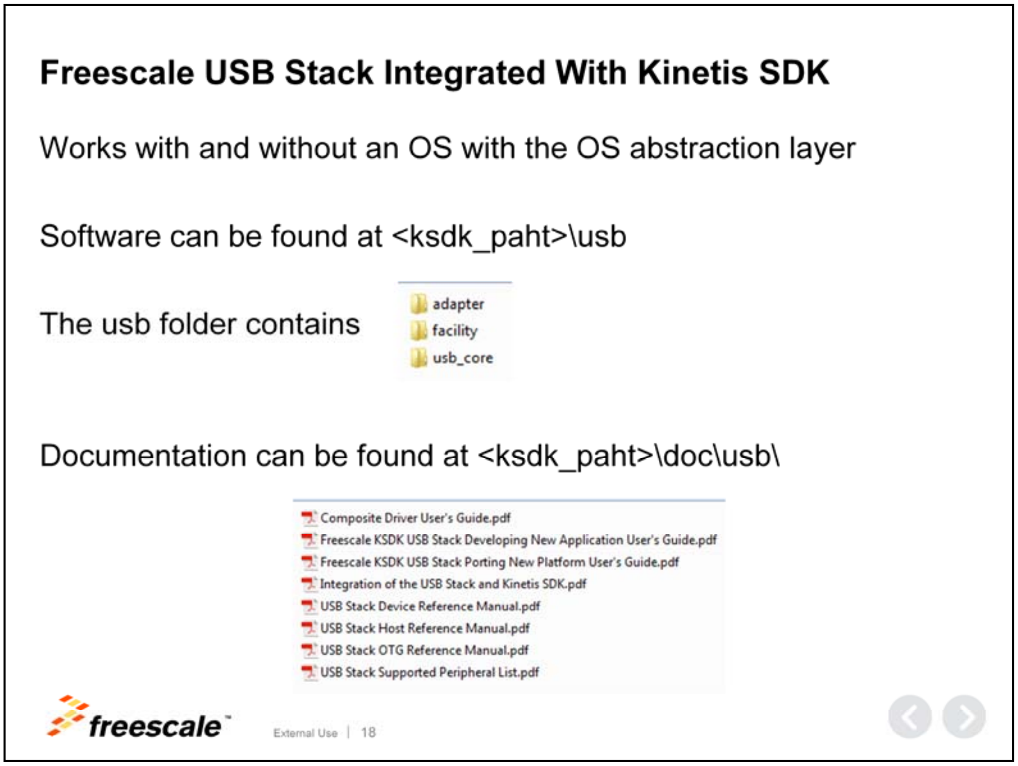
Avoid placing USB differential pairs near signals, such as clocks, periodic signals, and I/O connectors, that might cause interference.

Minimize vias and corners.

Route differential pairs on a signal layer, next to the ground plane.



Let’s move on to discuss software configuration.



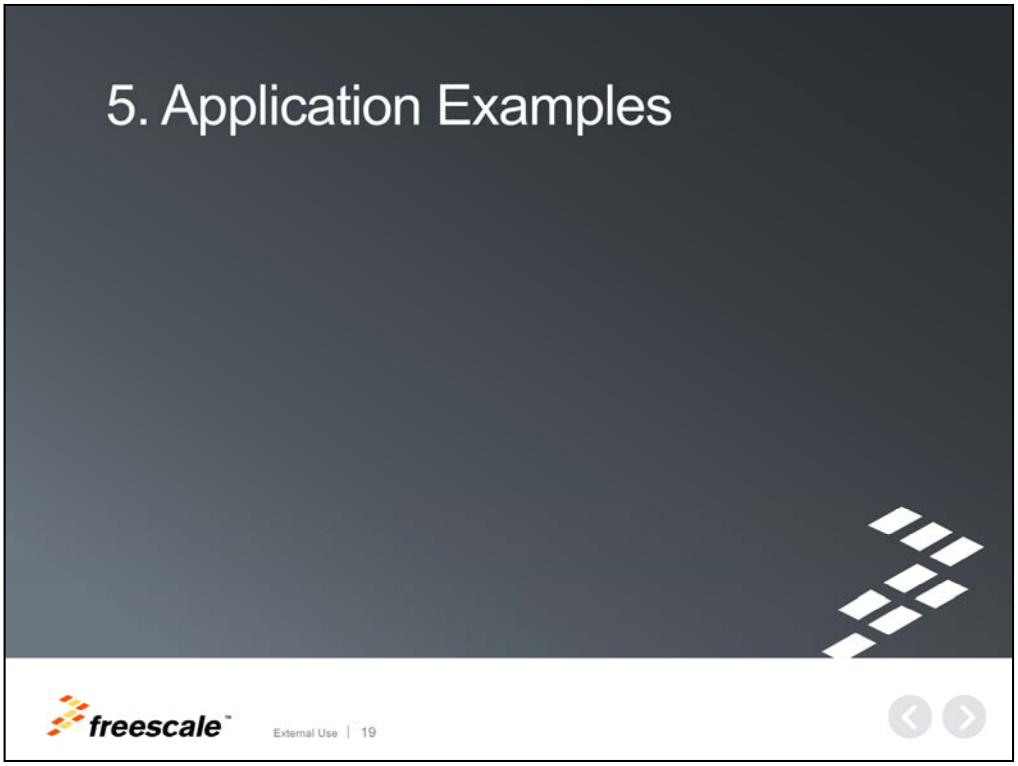
Freescale USB Stack Integrated With Kinetis SDK

Kinetis SDK includes USB stack capable to run on different RTOS with the same USB core code.

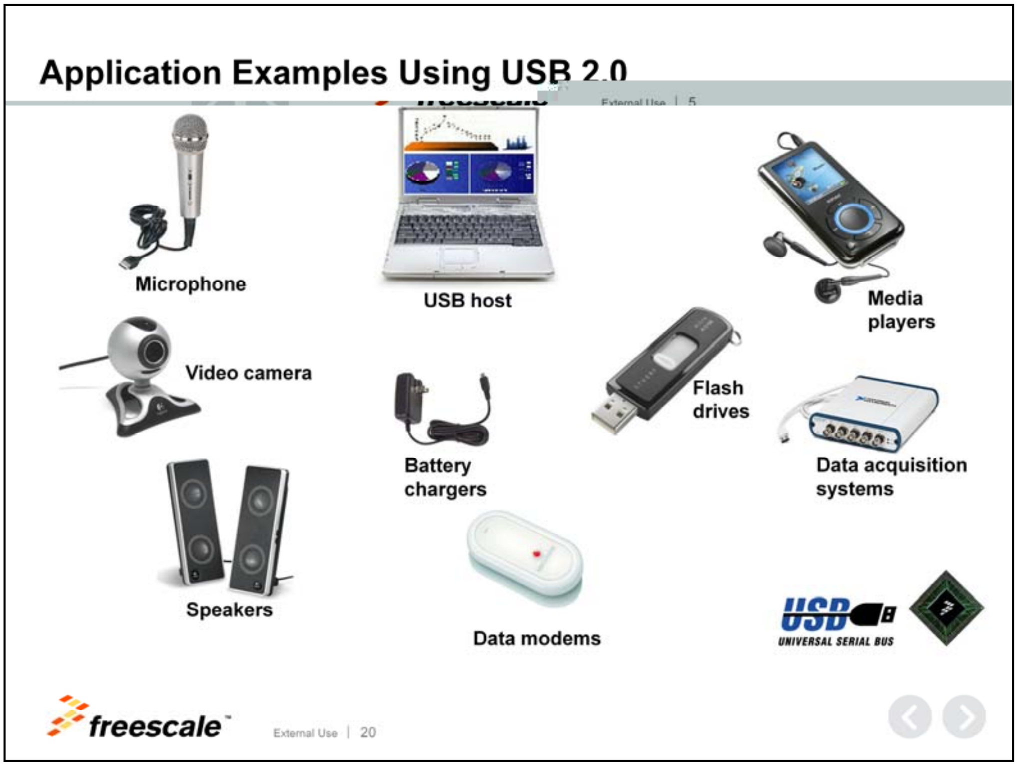
The software can be found at the usb folder. This folder contains:

- A usb_core folder that includes usb source files, such as HAL, controller driver, class drivers, and the USB library projects.
- An adapter folder that includes the adapter files to support the USB stack running on a different RTOS with the same USB core code.
- A facility folder interfaces to accommodates various peripherals to which the USB refers to by, such as UART.

To find more information about the USB stack for Kinetis SDK, go to the USB documentation folder.

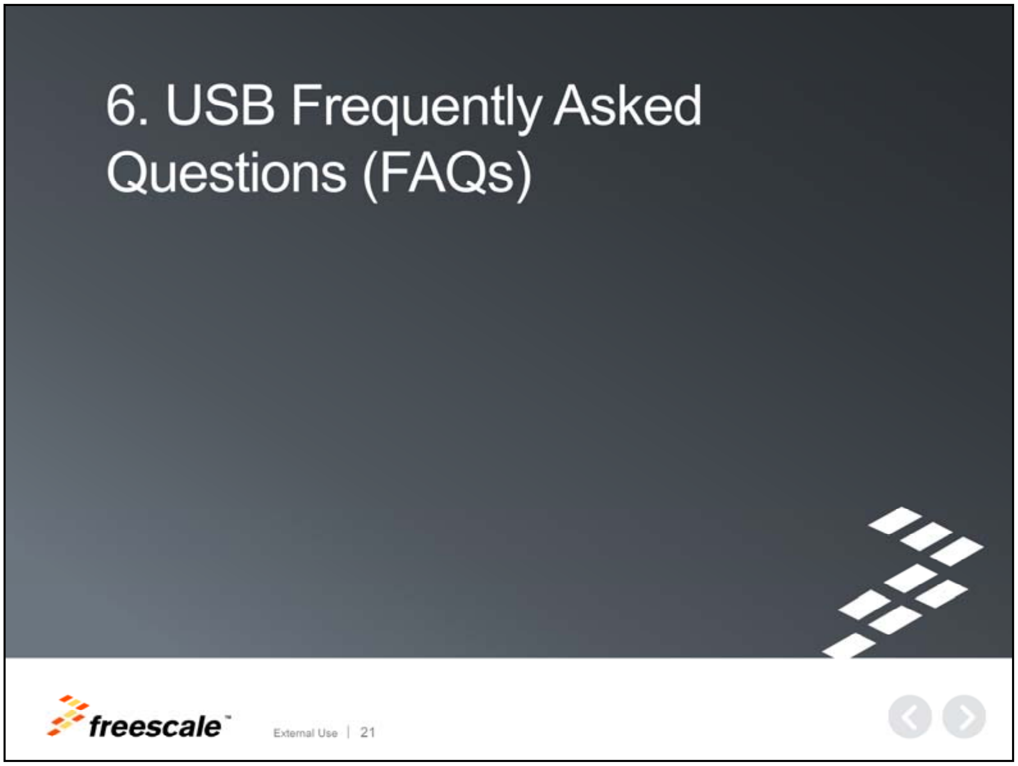


Now, let's review a few application examples.

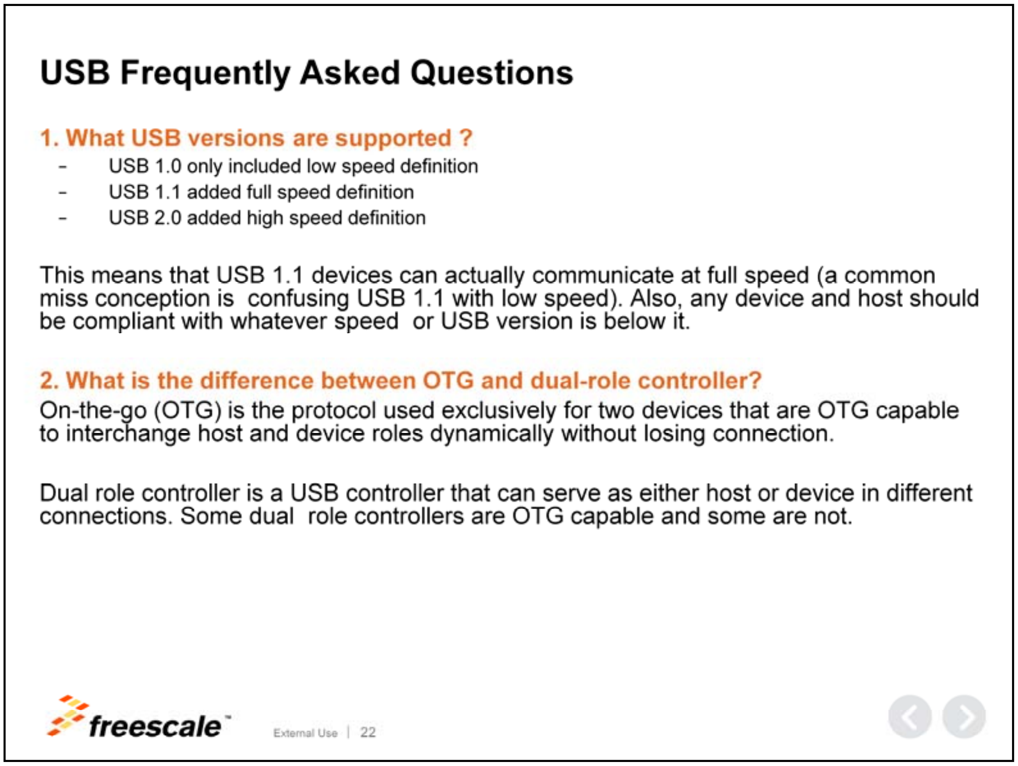


Application Examples Using USB 2.0

These are some application examples that use USB 2.0.



Finally, let's review some frequently asked questions.



USB Frequently Asked Questions

1. What USB versions are supported ?

USB 1.0 only included low speed definition

USB 1.1 added full speed definition

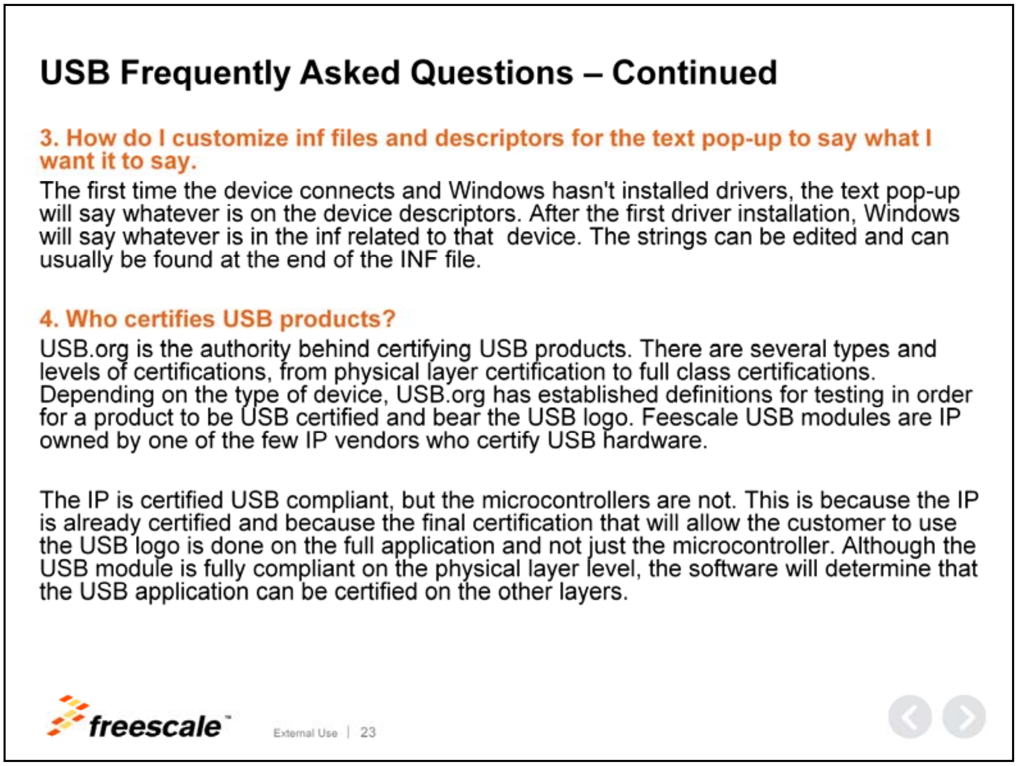
USB 2.0 added high speed definition

This means that USB 1.1 devices can actually communicate at full speed (a common miss conception is confusing USB 1.1 with low speed). Also, any device and host should be compliant with whatever speed or USB version is below it.

2. What is the difference between OTG and dual-role controller?

On-the-go (OTG) is the protocol used exclusively for two devices that are OTG capable to interchange host and device roles dynamically without losing connection.

Dual role controller is a USB controller that can serve as either host or device in different connections. Some dual role controllers are OTG capable and some are not.



USB Frequently Asked Questions – Continued

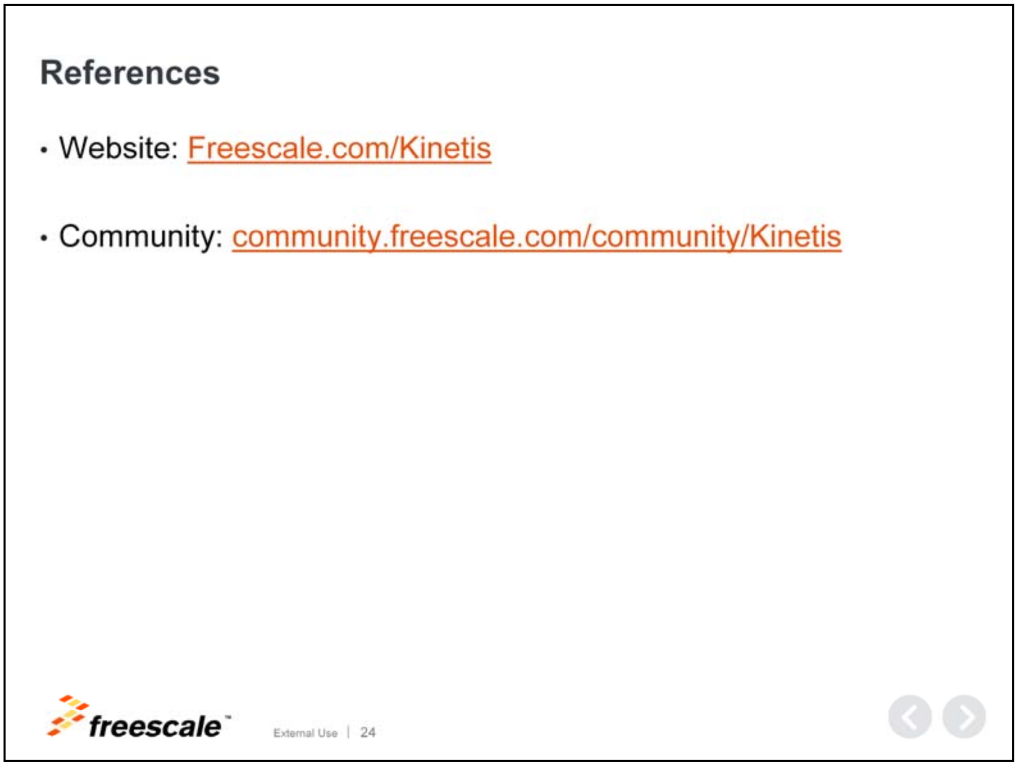
3. How do I customize inf files and descriptors for the text pop-up to say what I want it to say.

The first time the device connects and Windows hasn't installed drivers, the text pop-up will say whatever is on the device descriptors. After the first driver installation, Windows will say whatever is in the inf related to that device. The strings can be edited and can usually be found at the end of the INF file.

4. Who certifies USB products?

USB.org is the authority behind certifying USB products. There are several types and levels of certifications, from physical layer certification to full class certifications. Depending on the type of device, USB.org has established definitions for testing in order for a product to be USB certified and bear the USB logo. Feescale USB modules are IP owned by one of the few IP vendors who certify USB hardware.

The IP is certified USB compliant, but the microcontrollers are not. This is because the IP is already certified and because the final certification that will allow the customer to use the USB logo is done on the full application and not just the microcontroller. Although the USB module is fully compliant on the physical layer level, the software will determine that the USB application can be certified on the other layers.



This concludes our presentation of the USB module for Kinetis MCUs.

For more information, we invite you to visit us on the web at Freescale.com/Kinetis and check out our community page.

www.Freescale.com