



Bootloader Solution for Kinetis MCUs

AMF-DES-T1057

Anthony Huereca | Freescale Applications Engineer

M A R . 2 0 1 5



External Use

Freescale, the Freescale logo, AN520, C-S, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, MagniV, mdsMGT, PEG, PowerQUICC, Prosecc Expert, QorIQ, QorIQ Qonvergence, Qorivos, Ready Files, SafeAssure, the SafeAssure logo, StarCore, Synclonix, Vortiga, Vybrid and Xilinx are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. AirMat, BeeKit, FreeStack, CoreNet, Flexis, LayerStack, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink and UMEMS are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2015 Freescale Semiconductor, Inc.



Agenda



- Overview of the Kinetis Bootloader
- Kinetis Bootloader Configurations
- Release Packages and Schedule
- The Flow of the Bootloader
- Explore Major Components
- Recap





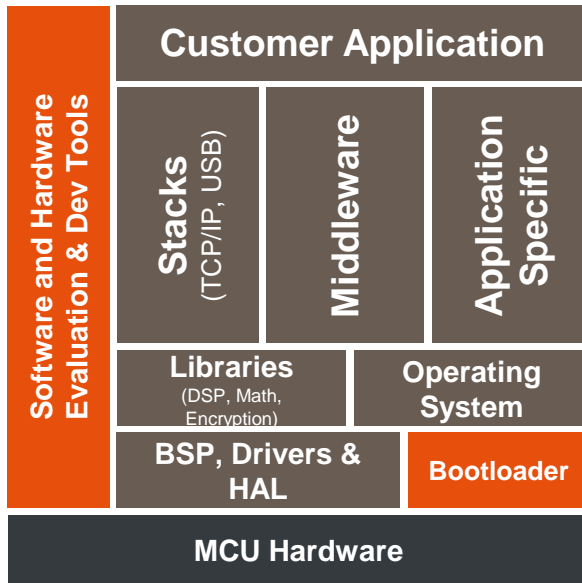
Kinetis Bootloader



Flash programming over a serial connection: erase, program, verify



Flash, ROM or RAM based bootloader with open-source software and host-side programming utilities.



Open Source Initiative

Product Features

- A common bootloader for Kinetis MCUs
- C/C++ Source code provided under a permissive BSD open source license
- Serial communications with a host via UART, SPI, I2C, and USB HID
 - Active peripheral detection
 - Common packet-based protocol for all peripherals
- Packet error detection and retransmission
- Configurable options for executing bootloader at startup or application runtime
- Command-line and GUI tools provided for Windows
- Designed to be flash, ROM or RAM resident
 - Failsafe boot mechanism on Kinetis devices with ROM
- Pre-programmed into flash (on devices without a dedicated ROM) and executed from RAM for built-in factory programming
- Fully customizable for use in customer applications providing reliable field updates



Introduction

Definitions:

- **Bootloader (BL)** – Firmware that is executed on an embedded microcontroller at reset or during runtime if called by the main application. It configures one or more data interfaces, accept an incoming firmware image, validate it, and program it to an internal or connected memory device. Bootloader firmware may be located in ROM or flash memory region.
- **Flashloader (FL)** – Special implementation of a bootloader intended for one-time use to load an initial image into flash during customer factory programming. A flashloader can eliminate the need for a debugger interface for production flash programming.
- **Bootloader Host Application (BLHost)** – Command-line application running on a computer system to allow the user to load software into the memory of a silicon device.
- **Kinetis Updater Application** – GUI application running on a computer system to allow the user to load software into the memory of a silicon device.
- **BUSPAL** – UART to I2C/SPI converter running on a Freedom(KL25Z) board

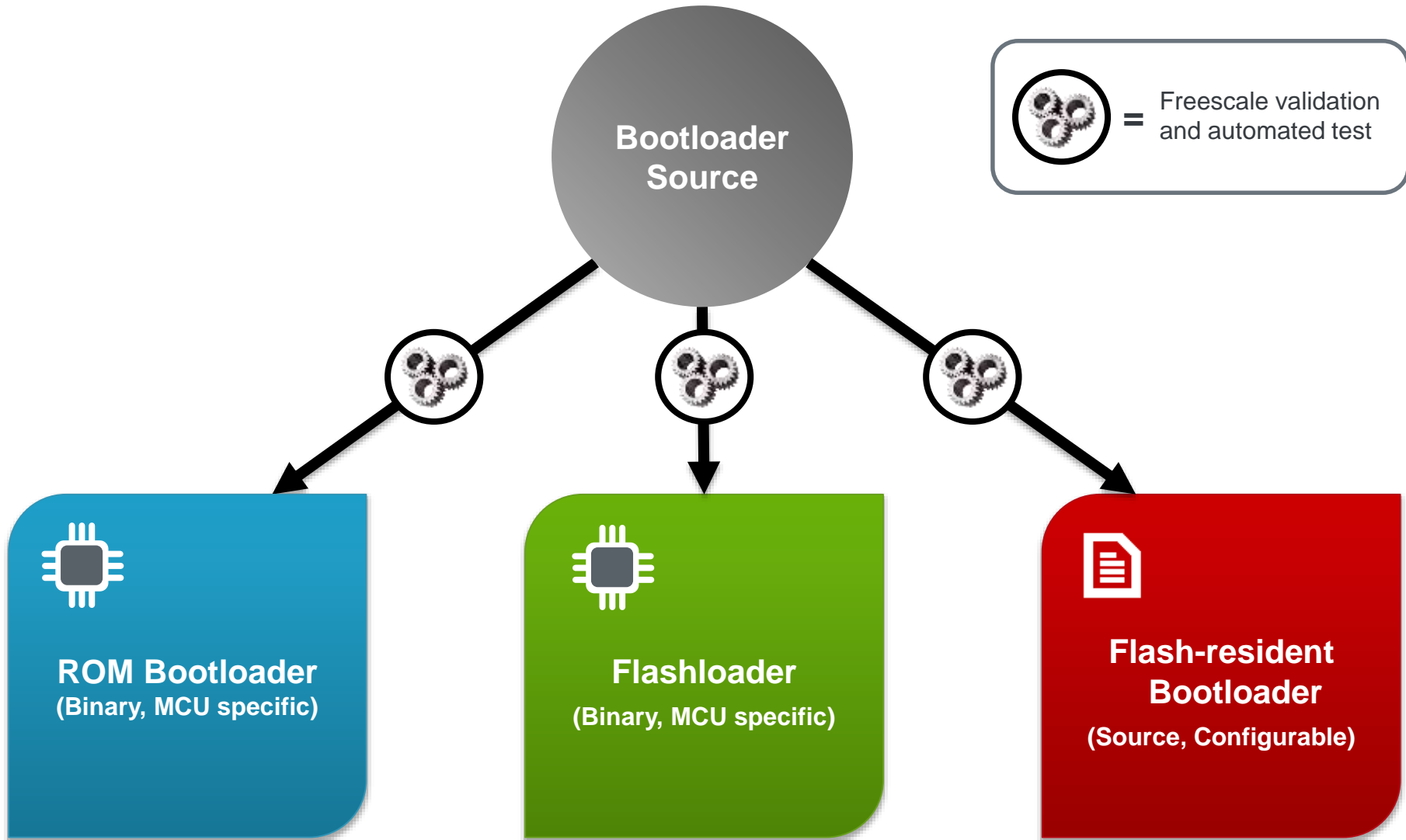
Releases:

- **Kinetis Bootloader** – Source code released 3 times per year, adding new device support and new features. This allows you to develop your own Flash-Resident Bootloader.
- **ROM Bootloader** – Delivered as part of silicon
- **Flashloader** – Programmed at the Freescale manufacturing factory.



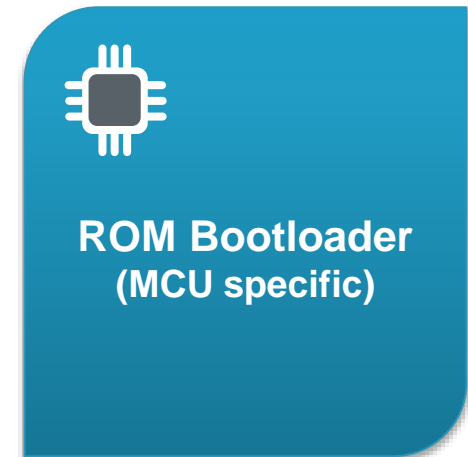
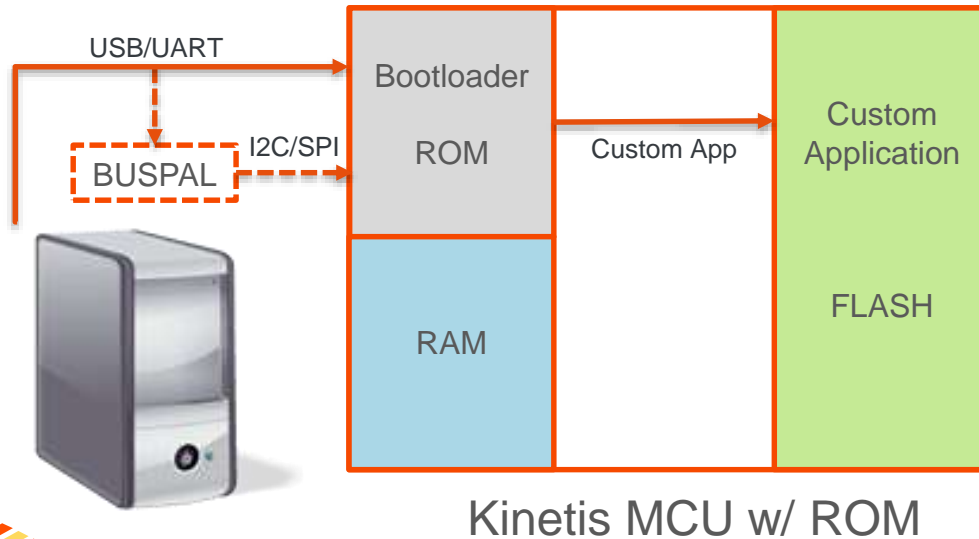
Kinetis Bootloader Configurations

Kinetis Bootloader Configurations



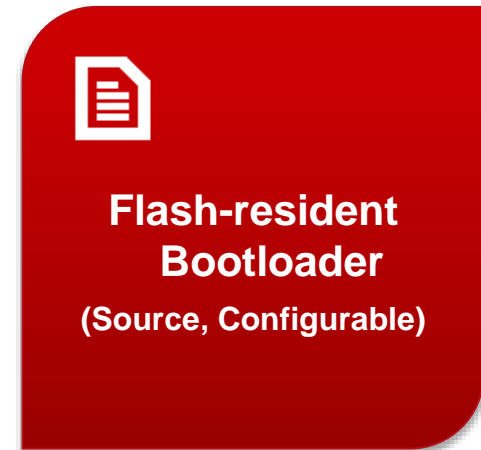
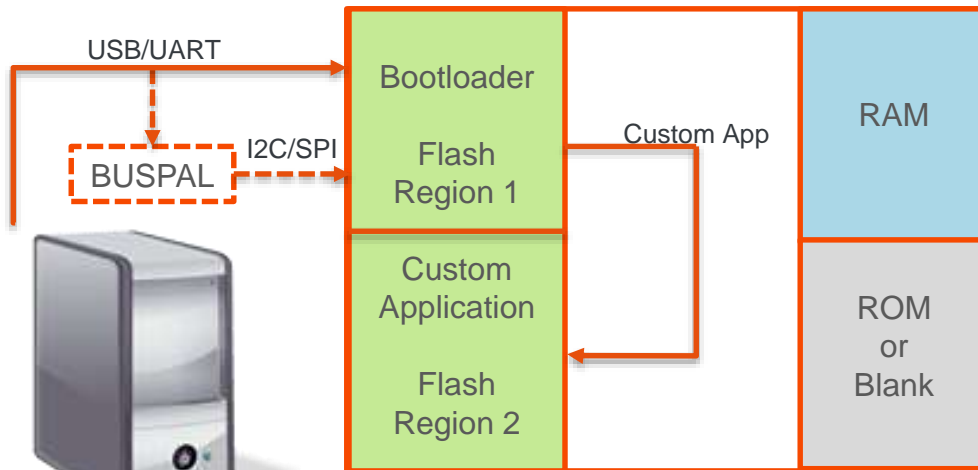
ROM Bootloader

- Available on KL03Z and future Kinetis devices featuring a boot ROM.
- **Pre-programmed in device ROM**; failsafe boot mechanism for factory and field programming
- Configured specifically for each MCU family
 - Peripheral interfaces
 - Command set
- User configurable via parameters stored in user flash (bootloader configuration area, BCA)
- Can run at system start-up and is callable from user application at runtime



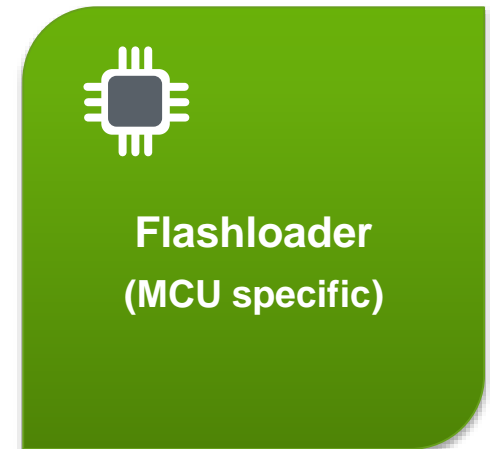
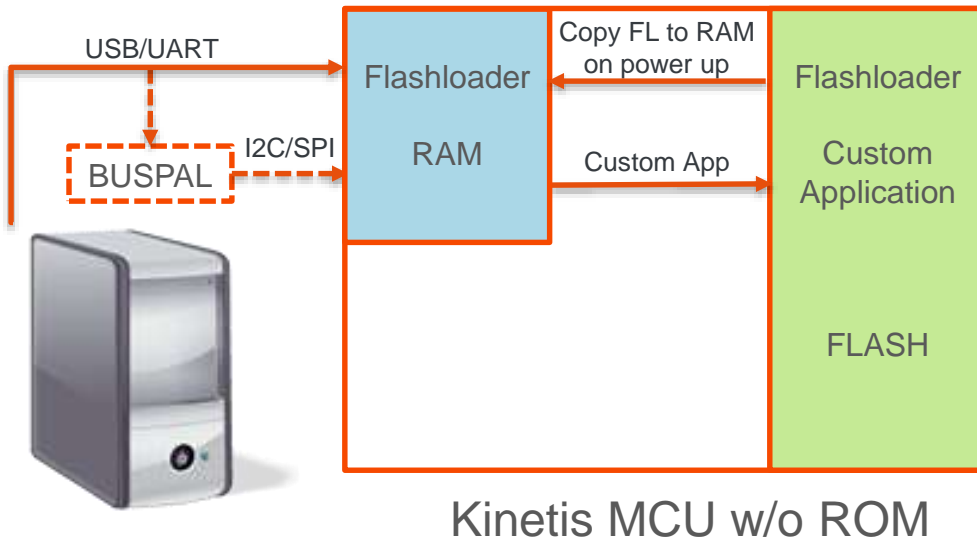
Flash-resident Bootloader

- Resides in flash and stays persistent in flash
- Used to write user applications to remaining areas of flash
- Source code provided by FSL to customer
- User configurable via parameters stored in user flash (BCA)
 - Available peripheral interfaces
 - Command set
 - Timeout period
- Can run at system start-up and callable from user application at runtime



Flashloader

- Provides a “one-shot” flash programming mechanism for factory programming
- Pre-programmed into user flash by Freescale on new Kinetis devices without a boot ROM (e.g. K22F devices)
- Runs at system startup – loaded to RAM and executed from there making entire flash array available for programming



Bootloader Configuration Comparison

Bootloader Configuration	ROM Bootloader	Flashloader	Flash-resident Bootloader
Use Case	Factory Flash Programming & Field Update	Factory Flash Programming	Field Update
Delivery Mechanism	Binary preprogrammed in ROM by Freescale	Binary preprogrammed in flash by Freescale	Source code provided in major release
Supported Devices	All Kinetis Devices with a boot ROM	Select Kinetis Devices without a ROM*	Select Kinetis Devices*
Clock Configuration	Configurable by user	Configured by FSL to a default setting	Configurable by user
Feature	Can run at system startup or callable from user application	Always run at system start-up	Can run at system startup or callable from user application
	Can jump to user application after peripheral timeout	Overwritten by user application	Can jump to user application after peripheral timeout

*List of supported devices on www.freescale.com/kboot

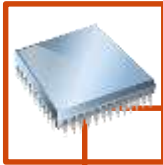
Bootloader vs. Flashloader

ROM and Flash-resident Bootloader	Flashloader
Clocks and other parameters can be configured	Uses defaults
Can jump to user app after peripheral timeout	Overwritten by user app, app started by POR
Can be called by app	Not present with app
Support flash-security-disable command	Flash security disabled by factory



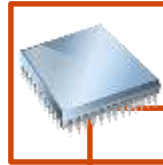
Release Packages and Schedule

Release Packages



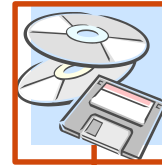
NPI w/o ROM

- Flashloader binary (pre-programmed on MCU)
- Documentation



NPI w/ ROM

- ROM Bootloader binary (pre-programmed on MCU)
- Documentation



Kinetis Bootloader

- Source code & project examples
- Command-line host application
- Kinetis Updater application
- Documentation

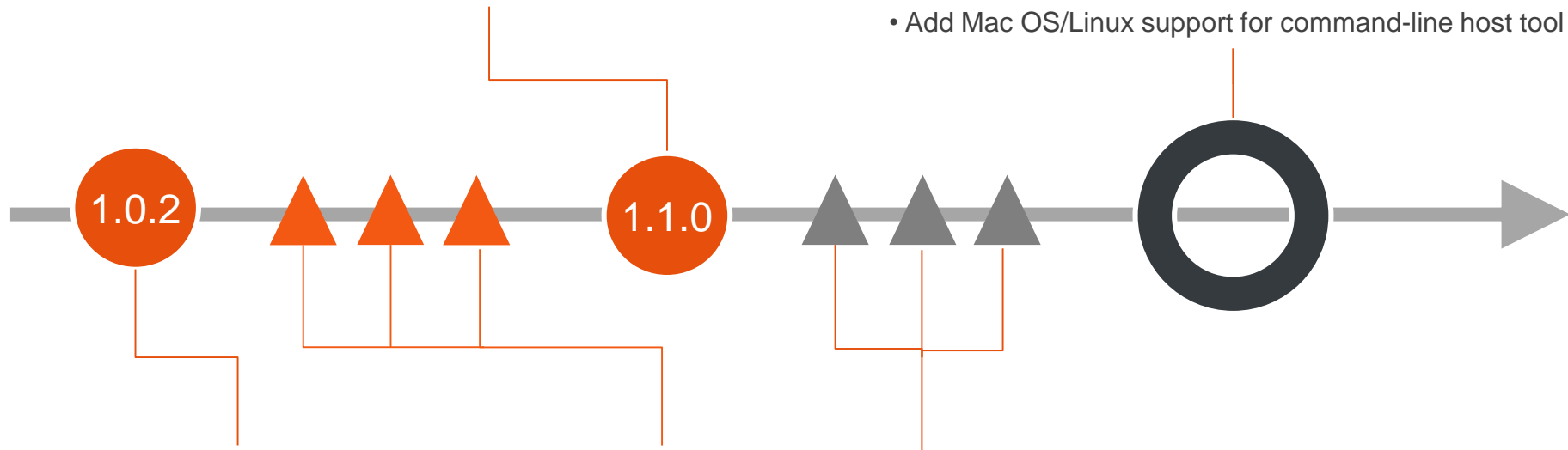
Kinetis Bootloader Release Schedule

1.1.0 – Dec, 2014

- Add support for K22F and KL25Z
- Update GUI host tool functionality and Windows 8 support
- Improved technical documentation set

1.2.0, 1.3.0

- Collection of new MCU support since previous release
- Additional features and more optimized design
- Enhanced integration with the SDK HAL drivers
- Add Mac OS/Linux support for command-line host tool



1.0.2 - July, 2014

- Source code and IAR project examples
- Support for K24F, K63F, K64F via FRDM-K64F
- Support for UART, USB-HID, I2C, SPI

NPI Support – Time of product launch

- Flashloader or ROM-bootloader support for applicable Kinetis MCUs
- Preprogrammed on MCUs when shipped to customers
- Flashloader/ROM-bootloader chapter in each MCU reference manual

All Supported Devices*

- **Flash-resident Bootloader Source Code**
 - K24F, K63F and K64F MCUs via the FRDM-K64F and TWR-K64F120M
 - K22F 100MHz and 120MHz MCUs via the FRDM-K22F and TWR-K22F120M
 - KL25Z MCUs via the FRDM-KL25Z and TWR-KL25Z48M
- **ROM Bootloader**
 - KL03, KL17, KL27, KL33, KL43
- **Flashloader**
 - K02, K22F, K24_256K, KV30, KV31

* Please refer to www.freescale.com/KBOOT for a full list of support device part numbers

Documentation

- **Generic**

- Bootloader Reference Manual
 - Porting chapter for **easy migration** to existing Kinetis devices
- Getting Started with ROM Bootloader
- Getting Started with Flashloader
- Host Tool User's Guide

- **Device-specific**

- Bootloader chapter in NPI Reference Manual
- User's Guide with Flash-resident Bootloader



The Flow of the Bootloader

Entering the Bootloader

ROM-based Bootloader

- Bootloader is executed out of reset if either of these are true:
 - FOPT[BOOTSRC_SEL] = 0b11 or 0b10
 - FOPT is a Flash Option byte at 0x40D in flash – flash erased state defaults to 0b11.
 - BOOTCFG0 pin is enabled and asserted (FOPT[BOOTPIN_OPT] = 0b0)
- A user application may execute the bootloader by calling the bootloader entry point (can be determined programmatically)

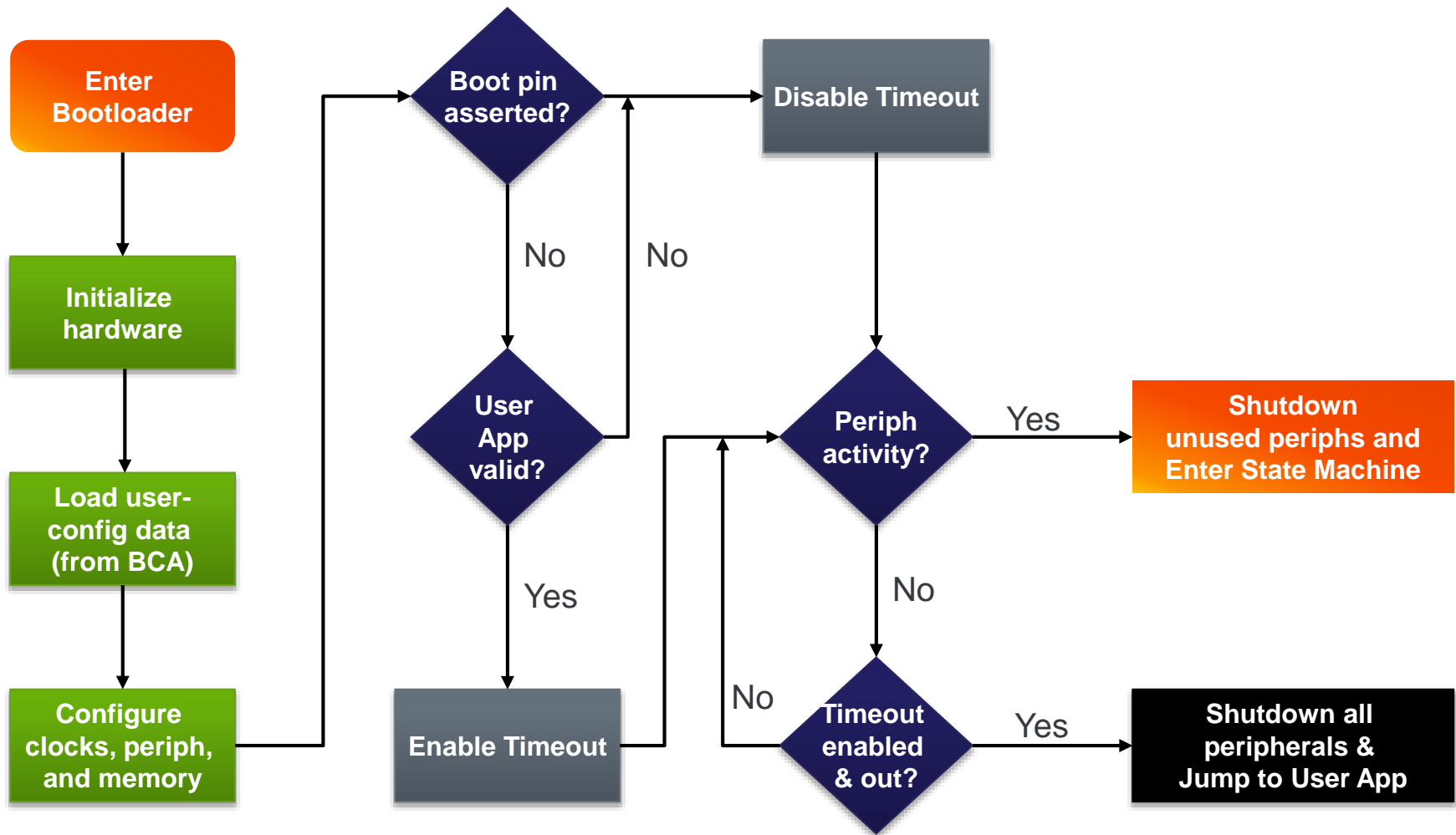
Flash-based Bootloader

- Completely configurable by the user
- Bootloader is executed out of reset if the Reset Vector points to the bootloader
- is_boot_pin_asserted() can be customized to utilize a user-selected GPIO pin
- A user application may execute the bootloader by calling the bootloader entry point (can be determined programmatically)

RAM-based Bootloader (Flashloader)

- Always executes out of reset
- Is intended as a “one-shot” flash programming method

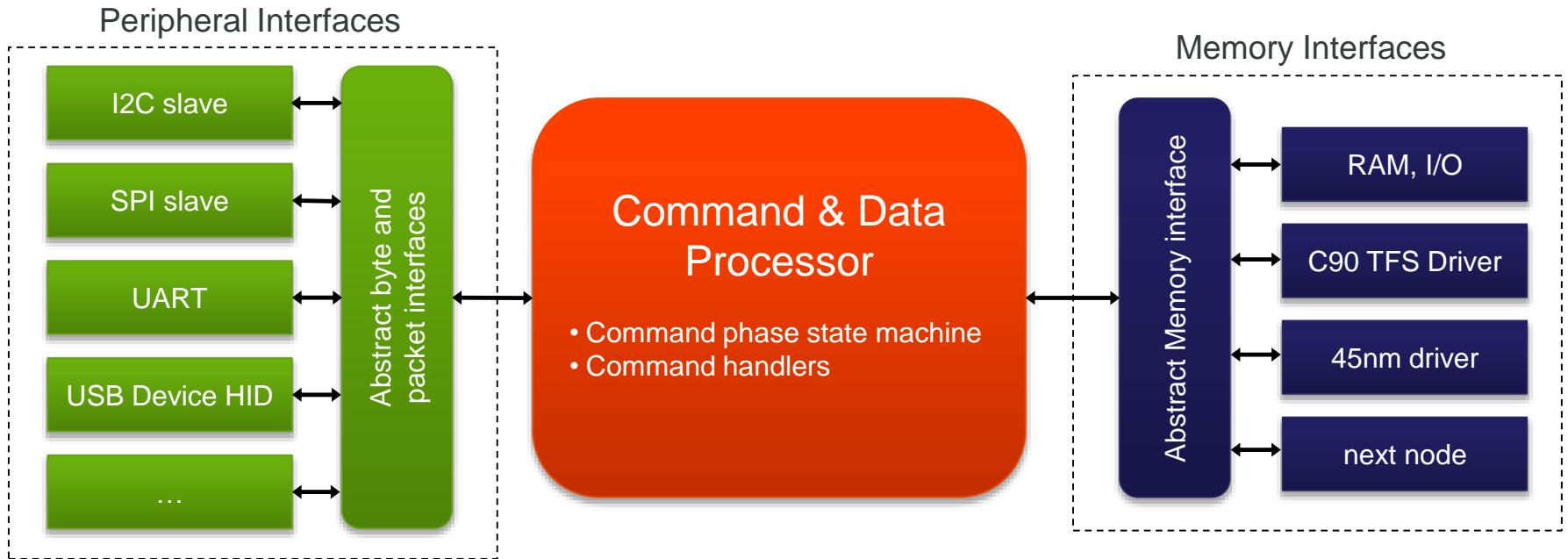
Bootloader Startup Flow (ROM/default flash)





Bootloader Major Components

Kinetis Bootloader Block Diagram



Command Processor Overview - Packets

Command & Data Processor

- Command phase state machine
- Command handlers

- All data sent between host and target is packetized
- Types of packets include framing, command, and data
- Framing packets
 - Used for flow control and error detection (via CRC-16) on serial interfaces without built-in packetization and flow control
 - Types of framing packets include:
 - ACK
 - NAK
 - AckAbort
 - Command
 - Data
 - Ping
 - PingResponse
- Command packets
 - Holds the command and parameters to be executed by the bootloader
- Data packets
 - Contents of a data packet is simply the data itself

Command Processor Overview - Commands

Name	Description	Supported when flash is secure?
FlashEraseAll	Erase the entire flash array.	No
FlashEraseRegion	Erase a range of sectors of flash.	No
ReadMemory	Get data from memory.	No
ReadMemoryResponse	Send the contents of memory.	No
WriteMemory	Write data to memory.	No
FillMemory	Fill memory with a pattern.	No
FlashSecurityDisable	Attempt to unlock flash security using the backdoor key.	Yes
GetProperty	Get the current value of a property.	Yes
GetPropertyResponse	Send the requested property value.	No
ReceiveSBFile	Receive and process an SB-format programming image.	No
Execute	Invoke a function that never returns control to the bootloader.	No
Call	Invoke a function that is expected to return.	No
Reset	Reset the chip.	Yes
SetProperty	Attempt to modify a writable property.	Yes
FlashEraseAllUnsecure	Erase the entire flash array, including protected sectors.	Yes

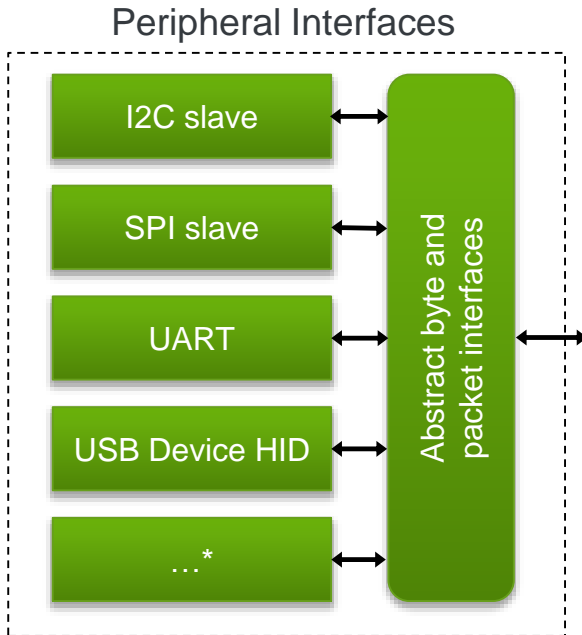
Bootloader Properties

- Properties are the defined units of data that can be accessed with the `GetProperty` or `SetProperty` commands
- Properties may be read-only or read-write
 - All read-write properties are 32-bit integers
- Not all properties are available on all platforms
 - If a property is not available, `GetProperty` and `SetProperty` will return `kStatus_UnknownProperty`

Bootloader Properties - Continued

Name	Writable	Size	Description
CurrentVersion	no	4	Current bootloader version.
AvailablePeripherals	no	4	The set of peripherals supported available on this chip.
FlashStartAddress	no	4	Start address of program flash.
FlashSizeInBytes	no	4	Size in bytes of program flash.
FlashSectorSize	no	4	The size in bytes of one sector of program flash. This is the minimum erase size.
FlashBlockCount	no	4	Number of blocks in the flash array.
AvailableCommands	no	4	The set of commands supported by the bootloader.
CRCCheckStatus	no	4	Status code from the last CRC check operation. Available only if the CRC check feature is supported.
VerifyWrites	yes	4	Boolean controlling whether the bootloader will verify writes to flash. A value of 0 means no verification is done, non-zero values enable verification. This feature is enabled by default.
MaxPacketSize	no	4	Maximum supported packet size for the currently active peripheral interface.
ReservedRegions	no	n	List of memory regions reserved by the bootloader. Returned as value pairs
ValidateRegions	yes	4	Boolean controlling whether the bootloader will validate attempts to write to memory regions. This feature is enabled by default.
RAMStartAddress	no	4	Start address of RAM
RAMSizeInBytes	no	4	Size in bytes of RAM
SystemDeviceId	no	4	Value of Kinetis System Device Identification register
FlashSecurityState	no	4	Boolean indicating whether flash security is enabled. 0 means disabled, 1 is enabled.
UniqueDeviceId	no	n	Device Unique ID values. The number of value items is indicated by the parameter count in the response packet

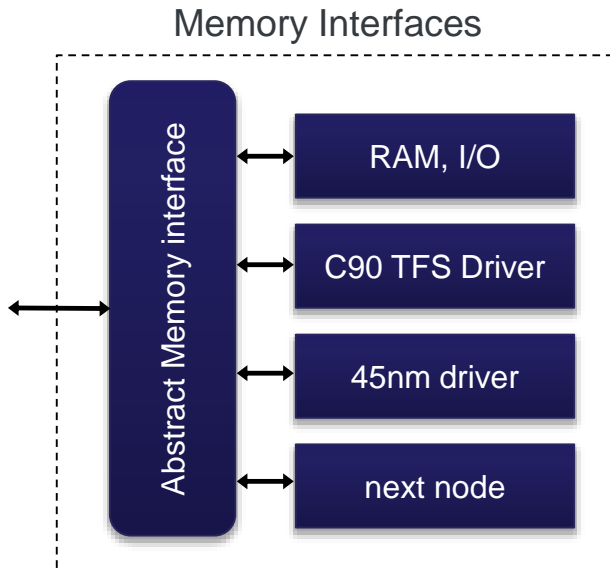
Kinetis Bootloader Supported Peripherals



- User can enable/disable any peripheral
- I²C slave
 - Uses framing packets
 - Default 7-bit slave address is 0x10—programmable in the bootloader configuration area (BCA)
- SPI slave
 - Uses framing packets
 - Phase = 1; data is sampled on rising edge
 - Polarity = 1; idle low
 - MSB transmitted first
 - 0x00 is sent for any transfer where there is not actual data to send—host uses framing packets to identify real data vs. dummy 0x00
- UART
 - Uses framing packets
 - Starts communication with autobaud detection sequence
- USB HID
 - Does *not* use framing packets; instead relies on the packetization inherent in USB protocol for flow control and error detection
 - Default VID/PID and strings assigned – user programmable in BCA

Kinetis Bootloader Memory Interface

- Provides a common, abstract interface
- Implements memory read/write/fill commands
- Uses a memory map table specific to the target MCU



```
/*! @brief Memory map for K64F12.
 **/
/*! This map is not const because it is updated at runtime with the
 **/ actual sizes of flash and RAM for the chip we're running on.
memory_map_entry_t g_memoryMap[] = {
    { 0x00000000, 0x000fffff, &g_flashMemoryInterface }, // Flash
    { 0x1fff0000, 0x2002ffff, &g_normalMemoryInterface }, // SRAM (256K)
    { 0x40000000, 0x4007ffff, &g_deviceMemoryInterface }, // peripherals
    { 0x400ff000, 0x400fffff, &g_deviceMemoryInterface }, // GPIO
    { 0xe0000000, 0xe00fffff, &g_deviceMemoryInterface }, // M4 peripherals
    { 0 } // Terminator
};
```

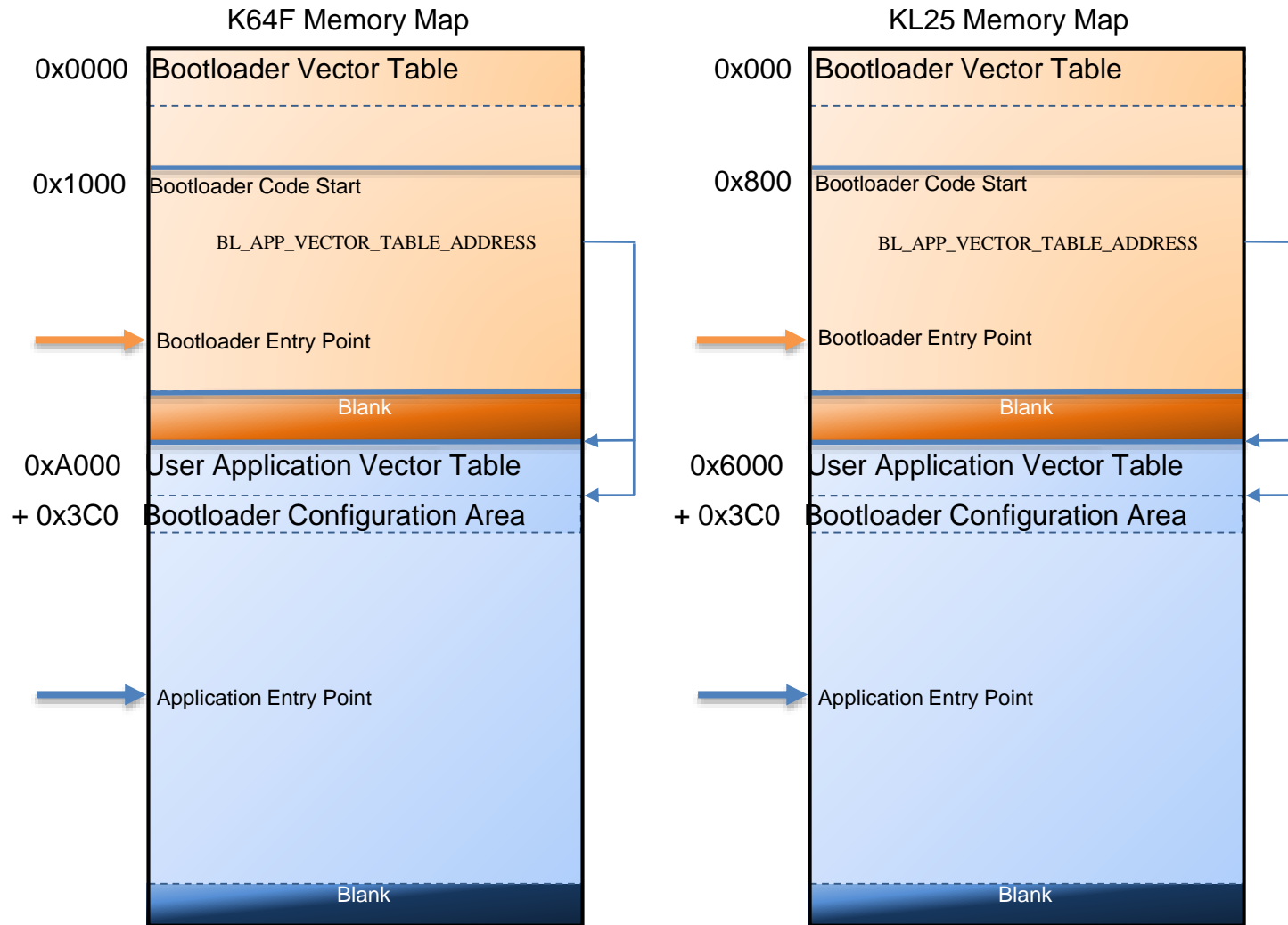
Bootloader Configuration Area

- The Bootloader configuration area (BCA) holds optional configuration parameters
- Location changes based on the target
 - Always in sector 0 of flash at address 0x3C0 for ROM targets
 - At offset 0x3C0 from user application for Flash targets
- Can be modified by the *write memory* command or can be set by the application image (similar to flash protection and security area)
- Includes options such as enabled peripherals, peripheral-specific settings, and bootloader timeout

Bootloader Configuration Area – Fields and Layout

Offset	Name	Description
0x00-0x03	tag	Magic number to verify bootloader config is valid. Must be set to 'kcfg'.
0x04-0x07	crcStartAddress	Start address of range to compute CRC.
0x08-0x0b	crcByteCount	Number of bytes on which CRC should be computed.
0x0c-0x0f	crcExpected	Expected CRC32 result.
0x10	enabledPeripherals	Bitfield of peripherals to enable.
0x11	i2cSlaveAddress	If not 0xFF, used as the 7-bit I2C slave address.
0x12-0x13	peripheralDetectionTimeout	Timeout in milliseconds for active peripheral detection.
0x14-0x15	usbVid	Custom USB VID value.
0x16-0x17	usbPid	Custom USB PID value.
0x18-0x1b	usbStringsPointer	n/a
0x1c	clockFlags	Enable/disable high speed mode
0x1d	clockDivider	Divider to use for core and bus clocks when in high speed mode.
0x1e-x01f	–	Reserved.

Flash-based Bootloader Memory Map Examples



Kinetis Bootloader Size

Target	Configuration	Supported Peripherals	ROM/flash Usage	RAM used
KL03Z	ROM, min	I2C, SPI, UART	8012 bytes (~8KB)	1028 bytes
KL43Z	ROM, min	I2C, SPI, UART, USB HID	16272 bytes (~16KB)	3601 bytes
K64F	Flash, full	I2C, SPI, UART, USB HID	20384 bytes (20KB)	4018 bytes
K22F	Flash, full	I2C, SPI, UART, USB HID	21776 bytes (~21KB)	4081 bytes



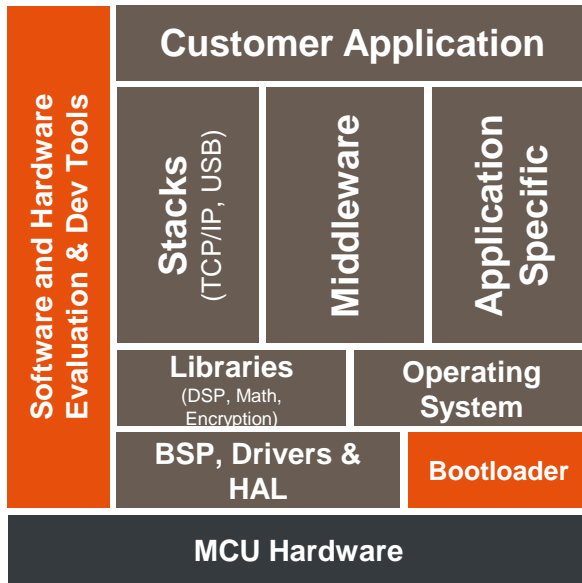
Kinetis Bootloader



Flash programming over a serial connection: erase, program, verify



Flash, ROM or RAM based bootloader with open-source software and host-side programming utilities.



Open Source Initiative

Product Features

- A common bootloader for all Kinetis MCUs
- C/C++ Source code provided under a permissive BSD open source license
- Serial communications with a host via UART, SPI, I2C, and USB HID
 - Active peripheral detection
 - Common packet-based protocol for all peripherals
- Packet error detection and retransmission
- Configurable options for executing bootloader at startup or application runtime
- Command-line and GUI tools provided for Windows
- Designed to be flash, ROM or RAM resident
 - Failsafe boot mechanism on Kinetis devices with ROM
- Pre-programmed into flash (on devices without a dedicated ROM) and executed from RAM for built-in factory programming
- Fully customizable for use in customer applications providing reliable field updates



Additional Resources



Community

<https://community.freescale.com/community/kinetis>



Web

www.freescale.com/kboot





www.Freescale.com