**RPK Software**

**Application Program API**

Document Number: RPK-AUA

Rev. 0.2, December 2017

.

Table of Contents

[1 Introduction 3](#_Toc501630605)

[2 RPK API – Application Program 3](#_Toc501630606)

[2.1 Application Program to External Flash 3](#_Toc501630607)

[2.1.1 Image Types 4](#_Toc501630608)

[2.1.2 Application Program API Functions 4](#_Toc501630609)

# Introduction

This document contains the description of the software application programming interface (API) for the RPK application external flash programming.

# RPK API – Application Program

## Application Program to External Flash

Applications are stored in external flash at the following addresses based on the type of application:



### Image Types

Defines are available for each of the Image Types:

**#define** FICA\_IMG\_TYPE\_NONE 0 // default

**#define** FICA\_IMG\_TYPE\_K64F\_FACT 1 // K64F Factory Image

**#define** FICA\_IMG\_TYPE\_K41Z\_FACT 2 // K41Z Factory Image

**#define** FICA\_IMG\_TYPE\_K64F\_USER 3 // K64F User Image

**#define** FICA\_IMG\_TYPE\_K41Z\_USER 4 // K41Z User Image

**#define** FICA\_IMG\_TYPE\_K64F\_BL\_FACT 5 // K64F Bootloader Factory Image

**#define** FICA\_IMG\_TYPE\_K41Z\_BL\_FACT 6 // K41Z Bootloader Factory Image

**#define** FICA\_IMG\_TYPE\_K64F\_CUR 7 // K64F Current Image

**#define** FICA\_IMG\_TYPE\_K41Z\_CUR 8 // K41Z Current Image

// Total Number of Images Defined in this Version

**#define** FICA\_NUM\_IMG\_TYPES 8

Defines are available for the external flash addresses associated with each type:

// Image Flash Start Addresses for this flash (MT25QL128)

**#define** FICA\_IMG\_FAC\_K64F\_APP\_ADDR 0x00100000

**#define** FICA\_IMG\_FAC\_K41Z\_APP\_ADDR 0x00200000

**#define** FICA\_IMG\_NEW\_K64F\_APP\_ADDR 0x00300000

**#define** FICA\_IMG\_NEW\_K41Z\_APP\_ADDR 0x00400000

**#define** FICA\_IMG\_BL\_FACT\_K64F\_APP\_ADDR 0x00500000

**#define** FICA\_IMG\_BL\_FACT\_K41Z\_APP\_ADDR 0x00600000

**#define** FICA\_IMG\_CUR\_K64F\_APP\_ADDR 0x00700000 // not used at the moment

**#define** FICA\_IMG\_CUR\_K41Z\_APP\_ADDR 0x00800000 // not used at the moment

The CUR (current) applications are not using any physical space in the external flash, other than their ICA entries. The location of the CUR applications are within the corresponding devices’ internal flash, however, an ICA entry is used to keep track of where the CUR application is in the internal flash, and the information associated with that application code (CRC, length, Encryption type, …).

Note: All types and defines listed here are from the External Flash Image Configuration Area (ICA) Version 1.

### Application Program API Functions

#### app\_program\_ext\_init means (Application Program External Flash Initialization)

Call this function to initiate an application external flash program.

int32\_t **app\_program\_ext\_init**(uint32\_t newimgtype, uint32\_t len)

imgtype – see #defines above

len – total length of the new image

Returns APP\_EXT\_NO\_ERROR (0) if no error occurred

Returns APP\_EXT\_ERROR (1) if an error occurred

#### app\_program\_ext\_cont means (Application Program External Flash Continue)

Call this function to program a page of binary application code into the external flash. The buffer page size MUST be 4096 bytes or smaller in length. The call is non-blocking since it can be called repetitively, with no time limits between calls. However, it is blocking when it needs to program a flash page. A flash page will be programmed whenvever a flash page size is reached internally. At this time, the function will not return until the flash page has completed programming, or an error occurs. However again, if the function is called within an RTOS, it will relinquish control to the RTOS as required. However triple, the underlying SPI flash write will ‘lock’ during its critical section, so “good luck, we’re all counting on you” (see Leslie Neilson, Airplane movie).

int32\_t **FICA\_app\_program\_ext\_cont**(**void** \*pbuf, uint32\_t len)

pbuf – pointer to the buffer to be programmed.

len – total length of the passed buffer to be programmed, max buffer size is 4096 bytes.

Returns APP\_EXT\_NO\_ERROR (0) if no error occurred

Returns APP\_EXT\_ERROR (1) if an error occurred

#### app\_program\_ext\_flush means (Application Program External Flash Flush)

Call this function to program (flush) the remaining application bytes to the external flash.

int32\_t **app\_program\_ext\_flush**()

Returns APP\_EXT\_NO\_ERROR (0) if no error occurred

Returns APP\_EXT\_ERROR (1) if an error occurred

#### app\_program\_ext\_calculate\_crc means (Application Program External Flash Calculate CRC)

This function will calculate a CRC value for the application image. If len = 0, the current ICA imgtype length will be used.

int32\_t **app\_program\_ext\_calculate\_crc** (uint32\_t imgtype, uint32\_t len, uint32\_t \*pcrc)

imgtype – see #defines above

len – total length to use for the crc calculation. If len equals 0, the current ICA imgtype length will be used.

pcrc – pointer to the location where the calculated crc will be stored

Returns APP\_EXT\_NO\_ERROR (0) if no error occurred

Returns APP\_EXT\_ERROR (1) if an error occurred

#### app\_program\_ext\_program\_crc means (Application Program External Flash Program CRC)

This function will program a passed CRC for the passed application image type to the external flash.

int32\_t **app\_program\_ext\_program\_crc** (uint32\_t imgtype, uint32\_t crc)

imgtype – see #defines above

crc – crc value to be programmed into the external flash for the passed image type

Returns APP\_EXT\_NO\_ERROR (0) if no error occurred

Returns APP\_EXT\_ERROR (1) if an error occurred

#### app\_program\_ext\_finalize means (Application Program External Flash Finalize)

Call this function to finalize the application flash programming. This function will release the SPI flash to be used by other devices.

int32\_t **app\_program\_ext\_finalize**()

Returns APP\_EXT\_NO\_ERROR (0) if no error occurred

Returns APP\_EXT\_ERROR (1) if an error occurred

***How to Reach Us:***

**Home Page:**

[nxp.com](http://www.nxp.com/)

**Web Support:**

[nxp.com/support](http://www.nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address:

[nxp.com/SalesTermsandConditions.](http://www.nxp.com/SalesTermsandConditions)

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD,

Freescale, and the Freescale logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners.

© 2016 NXP B.V.

Document Number MGSBIUG Revision 1.1, December 2016

.