Tools Advisory: fl_getNextBootImage(): invalid boot image

IN THIS DOCUMENT

1 Products affected

All versions of xTIMEcomposer up to and including 14.3.0

2 Issue

It has been discovered during product testing that the function `fl_getNextBootImage()` contained in the libraries libflash and libquadflash, is capable of returning an invalid boot image. This causes DFU to fail.

When `fl_getNextBootImage()` is called it scans through flash memory looking for the expected image header tag on sector boundaries. When it finds an image header tag it returns this as the next valid boot image.

In this product test case, an upgrade image was correctly identified in flash memory by DFU and the first page of that image was erased to invalidate it. DFU then continued scanning the rest of flash memory to verify that replacing the found upgrade image won't result in other upgrade images being overwritten. It was during this verification stage that DFU using `fl_getNextBootImage()` found another image tag within this upgrade images space. The image tag found was in fact a literal within the the const pool of the upgrade application and was coincidentally also sector boundry aligned in flash memory. This prevented the full erase of the existing upgrade image and subsequent successful replacement with the new upgrade image. Only the factory image could be booted from then on.

3 Resolution

The function `fl_getNextBootImage` in the libraries libflash and libquadflash has been updated to perform a CRC check of the first page of a potential image when it finds an image tag on a sector boundary. It compares the CRC calculated against what is expected for the first page in the image header table. The image is only returned if the CRC check is passed. This fix is only available in xTIMEcomposer 14.3.1 and above. Note that the factory image MUST be generated with this version of XTC in order to have the fix.

4 xflash_image_check

XMOS has provided the xflash_image_check application to scan existing xflash generated binary files for possible image tags on sector boundaries that are not valid images. This will advise if there is a potential for a DFU failure with exitisting images. xflash_image_check is available on XMOS.com along with instruction for use.

5 Recommendation

For customers who are already using libflash and libquadflash for DFU it is mandatory that any new upgrade images are checked to ensure that when written to flash memory will not result in an invalid image tag value being sector boundary aligned.

For all new developments of applications using DFU and or libflash/libquadflash then it is recommended that these customers upgrade to xTIMEcomposer 14.3.1 or above to make use of the fix provided. xTIMEcomposer 14.3.1 is fully compatible with the xTIMEcomposer 14.X suite.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the “Information”) and is providing it to you “AS IS” with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.