



# FRONTGRADE

## APPLICATION NOTE

### UT32M0R500

UART Flash Download

2/7/2019

Version #: 1.0.0

Product Name	Manufacturer Part Number	SMD #	Device Type	Internal Pic Number
Arm Cortex M0+	UT32M0R500	5962-17212	Flash Download	QS30

## 1.0 Overview

This document details the process of creating and downloading a hex or src image. Keil ARM development tools are used to create the image. Once the image is created, a Terminal Window is used to download the image via a

Serial Port. For the purposes of this document, we will use the **helloworld** project from AppNote\_UT32M0R500\_Creating\_Projects.pdf. Using this template, the user should be able to upload a hex or src image file to Flash memory on the UT32M0R500 via UART using a Terminal Window.

## 2.0 Steps to Create and Download an Image to the UT32M0R500



1. Launch Keil uVision
2. From the Project menu, select Options for Target 'Target 1' ... (Figure 1).

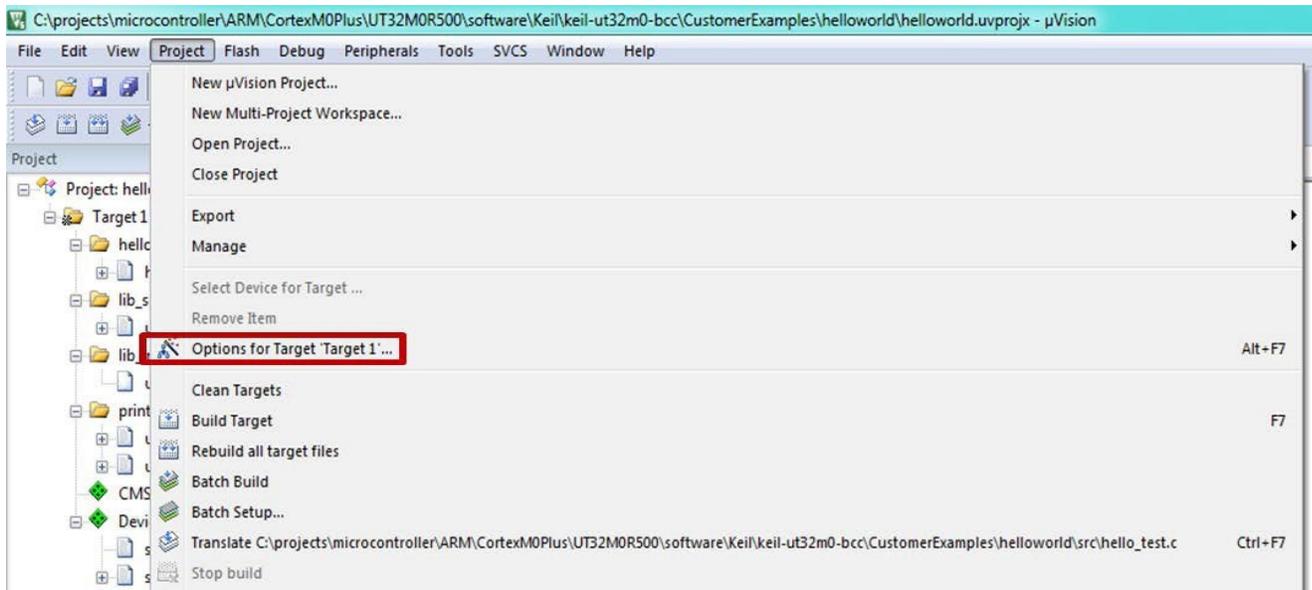


Figure 1: Project Setup

3. In the **Options** dialog box, on the **Output** tab, check **Create HEX file** (Figure 2), and click **OK**

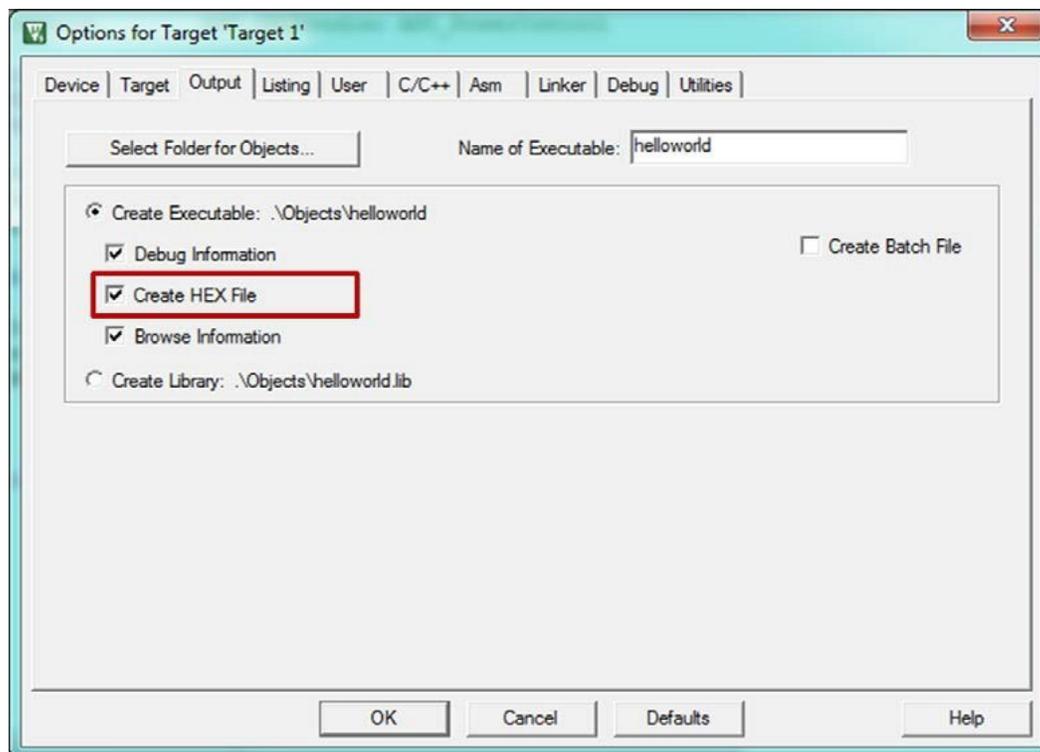


Figure 2: Output Options

4. In the **Project Explorer** view, click on  and **Rebuild** the project.

5. Once the hex file has been created, open **Tera Term**. From the Setup dialog box, select the correct **Port...** and set the port to the following settings (Figure 3).

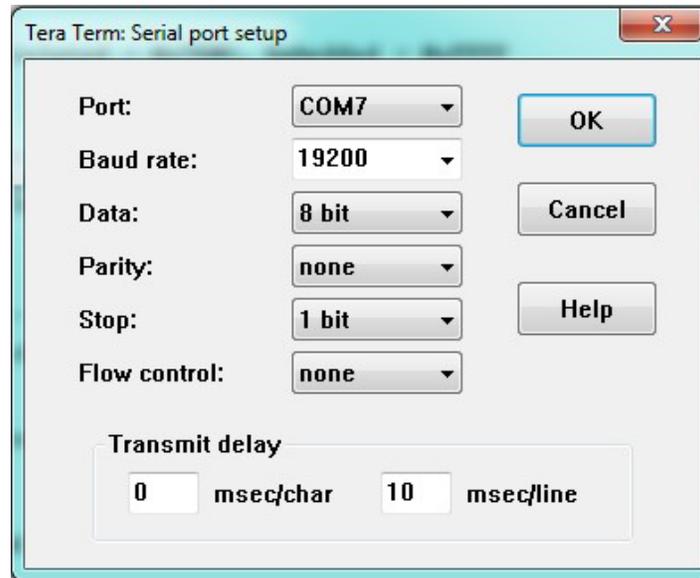


Figure 3: Serial Port Settings

After you've configured the switches for BOOTCFG in the b'10 position, and hit RESET on the evaluation board, the Terminal window displays the following. (For a list of all the commands, see Figure 8 on page 10).

```
Updating from UART...  
  
Welcome to the Cobham AES UT32M0R500 BootROM  
Enter a '?' for the user menu  
=>
```

6. For this app note, we'll use image 0. The command to select the image is `IMG -n#`. First, we'll choose NOR Flash as the device, as in the following illustration.

```
Device to Process: DEV -t#
                   specifies target device
                   for -t (type), # can be:
                   N: for NOR Flash
                   S: for SPI Flash
                   R: for SRAM
```

```
:>DEV -tN
```

Before updating the file, choose the image number, then erase and verify it before uploading it; see below. If the error message "Embedded = 0xFFFF" is returned after `VFY`, it means that there is no image at the specified image; this will be resolved after the image is loaded and the CRC is calculated.

```
:>IMG -n0
```

```
NFC init SUCCESS!
```

```
:>ERS
```

```
:>VFY
```

```
ERROR: CRC mismatch. Calculated = 0x7E0C, Embedded = 0xFFFF
```

```
Image to Process: IMG -n#
                   specifies image for next command series
                   for -n (number), # is:
                   image number: 0..3
```

7. To upload the image, use the command `PGM -f#`, see below.

```
:>PGM -fH
```

- Now, be sure all THREE of the following features are enabled: a) XON/OFF software flow control
- 10ms line pacing
- binary mode

Send/upload the image (hex) file now.

```
Program Image:    PGM -f#
                   writes to-be-uploaded image
                   for -f (format), # can be:
                   H: for Intel Hex records
                   S: for Motorola S19 records
```

To load the file, first set up the Terminal with 10 msec/line “line spacing” and XON/OFF flow control (Figure 4).

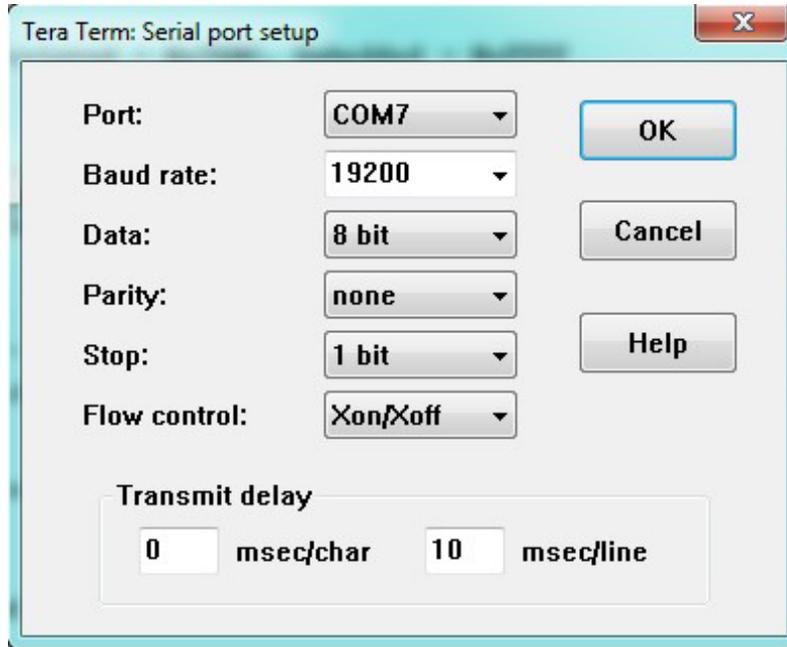


Figure 4: Serial Port Upload Settings

Then, from the **File** drop-down menu, choose Send file... (Figure 5).

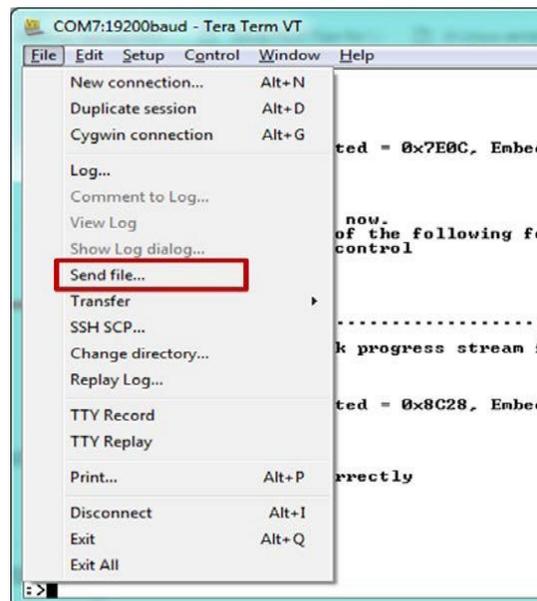


Figure 5: Send File

Open the hex file to be downloaded, and make sure Binary option is selected (Figure 6).

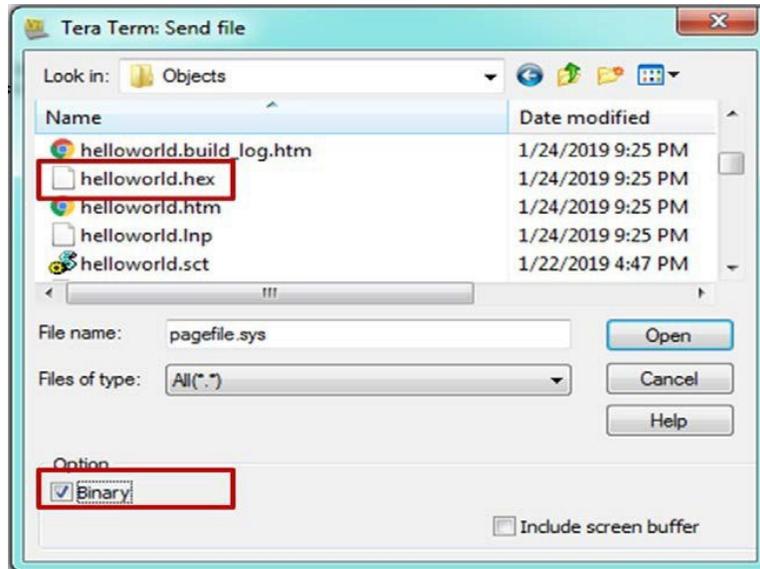


Figure 6: Open Hex File

As the image is being programmed to the NOR Flash, the terminal displays a line of dots as follows:

```
:>.....
```

Once programming is complete, if there are errors, the Terminal will display errors "E/1/2/3", within the line of dots. Errors are defined as follows:

- '1' is for processing Intel Hex record error.
- '2' is for processing Motorola S record error.
- '3' is flash write record error.

Programming complete -- check progress stream for any 'E/1/2/3' (errors) If there are no errors, the Terminal displays only the line of dots.

8. Finally verify the image by issuing VFY and CRC commands, see below.

```
:>VFY
ERROR: CRC mismatch. Calculated = 0x8C28, Embedded = 0xFFFF
:>CRC -c8C28
SUCCESS!! CRC programmed correctly
:>VFY
SUCCESS!! CRC match
```

```
CRC-Stamp Image:   CRC -c####
                   embeds CRC into image for verification
                   for -c <CRC>, #### MUST be:
                   four-digit hexadecimal number, all CAPS
                   (A2C4, for example)

Verify Image:      VFY
                   use embedded CRC to verify image
```

After “SUCCESS!! CRC match”, change BOOTCFG to b’00 and reset or cycle power to the board. The Terminal should display “hello world” (Figure 7).

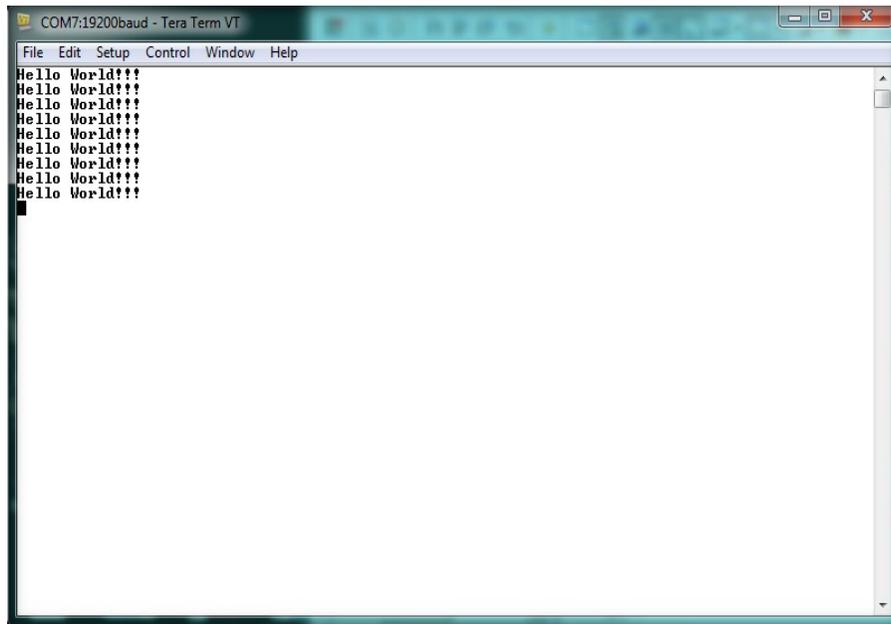
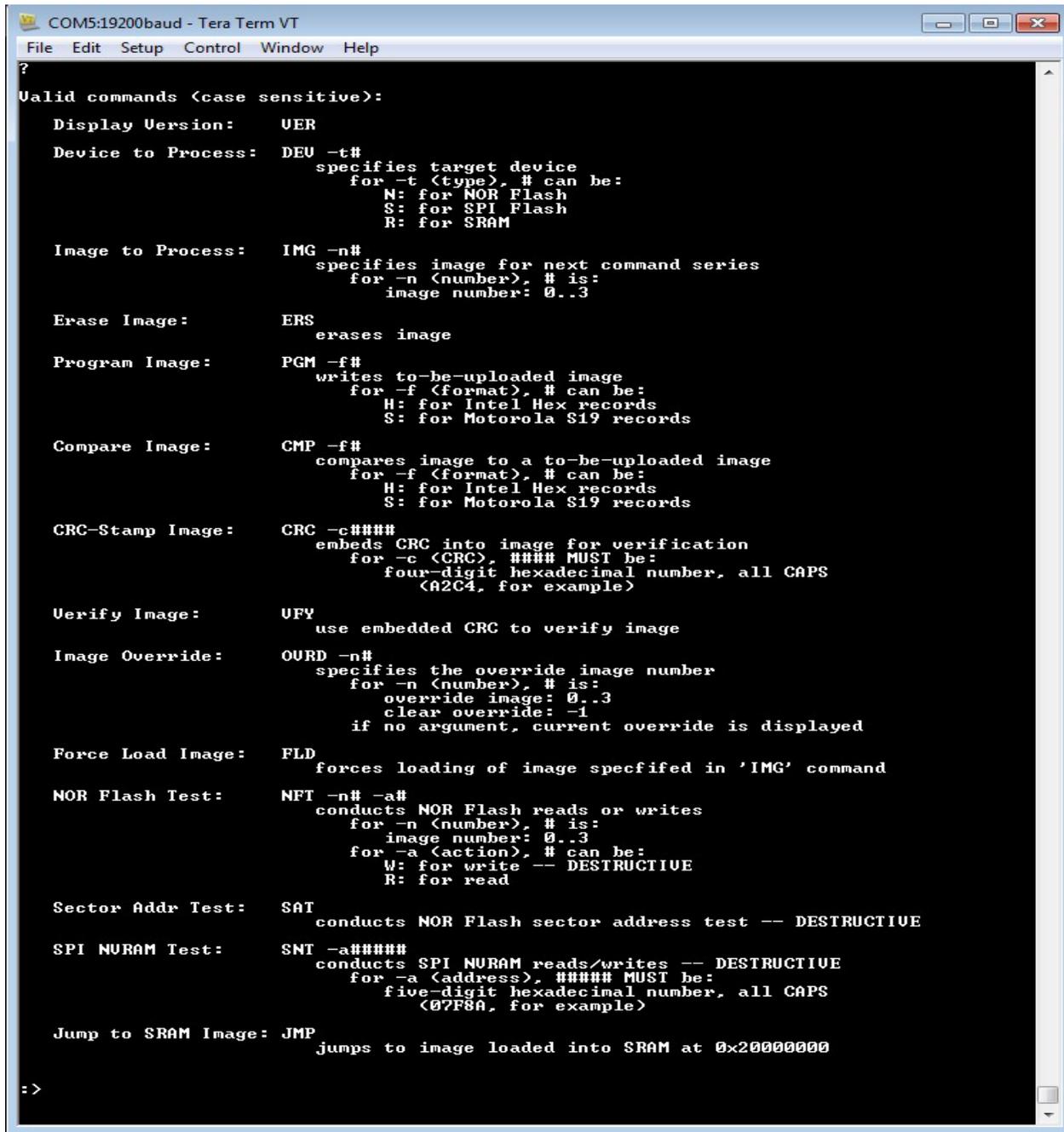


Figure 7: Hello World Display

A screenshot of a Tera Term VT window titled 'COM5:19200baud - Tera Term VT'. The window displays a list of valid commands for flash download, each with its syntax and a brief description. The commands include: Display Version (UER), Device to Process (DEU -t#), Image to Process (IMG -n#), Erase Image (ERS), Program Image (PGM -f#), Compare Image (CMP -f#), CRC-Stamp Image (CRC -c####), Verify Image (UFY), Image Override (OURD -n#), Force Load Image (FLD), NOR Flash Test (NFT -n# -a#), Sector Addr Test (SAT), SPI NURAM Test (SNT -a#####), and Jump to SRAM Image (JMP). The window has a menu bar with 'File', 'Edit', 'Setup', 'Control', 'Window', and 'Help'. The cursor is at the bottom left, showing ':>'.

```
COM5:19200baud - Tera Term VT
File Edit Setup Control Window Help
?
Valid commands (case sensitive):
  Display Version:      UER
  Device to Process:   DEU -t#
                        specifies target device
                        for -t (type), # can be:
                        N: for NOR Flash
                        S: for SPI Flash
                        R: for SRAM
  Image to Process:    IMG -n#
                        specifies image for next command series
                        for -n (number), # is:
                        image number: 0..3
  Erase Image:         ERS
                        erases image
  Program Image:       PGM -f#
                        writes to-be-uploaded image
                        for -f (format), # can be:
                        H: for Intel Hex records
                        S: for Motorola S19 records
  Compare Image:       CMP -f#
                        compares image to a to-be-uploaded image
                        for -f (format), # can be:
                        H: for Intel Hex records
                        S: for Motorola S19 records
  CRC-Stamp Image:     CRC -c####
                        embeds CRC into image for verification
                        for -c (CRC), ##### MUST be:
                        four-digit hexadecimal number, all CAPS
                        (A2C4, for example)
  Verify Image:        UFY
                        use embedded CRC to verify image
  Image Override:      OURD -n#
                        specifies the override image number
                        for -n (number), # is:
                        override image: 0..3
                        clear override: -1
                        if no argument, current override is displayed
  Force Load Image:    FLD
                        forces loading of image specified in 'IMG' command
  NOR Flash Test:      NFT -n# -a#
                        conducts NOR Flash reads or writes
                        for -n (number), # is:
                        image number: 0..3
                        for -a (action), # can be:
                        W: for write -- DESTRUCTIVE
                        R: for read
  Sector Addr Test:    SAT
                        conducts NOR Flash sector address test -- DESTRUCTIVE
  SPI NURAM Test:      SNT -a#####
                        conducts SPI NURAM reads/writes -- DESTRUCTIVE
                        for -a (address), ##### MUST be:
                        five-digit hexadecimal number, all CAPS
                        (07F8A, for example)
  Jump to SRAM Image: JMP
                        jumps to image loaded into SRAM at 0x20000000
:>
```

Figure 8: Flash Download Commands

## Revision History

Date	Revision #	Author	Change Description	Page #
02/07/2019	1.0.0	JA	Initial Release	

**Frontgrade Technologies Proprietary Information** Frontgrade Technologies (Frontgrade or Company) reserves the right to make changes to any products and services described herein at any time without notice. Consult a Frontgrade sales representative to verify that the information contained herein is current before using the product described herein. Frontgrade does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by the Company; nor does the purchase, lease, or use of a product or service convey a license to any patents, rights, copyrights, trademark rights, or any other intellectual property rights of the Company or any third party.