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 srec 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 srec 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).

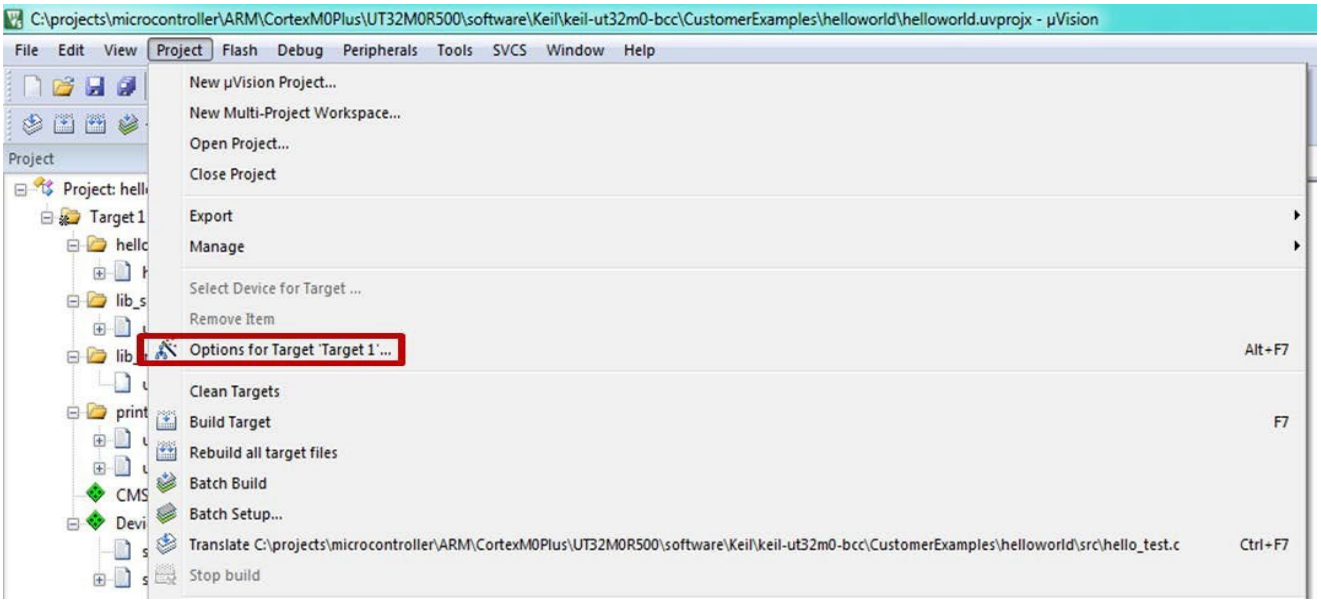

Figure 1: Project Setup

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

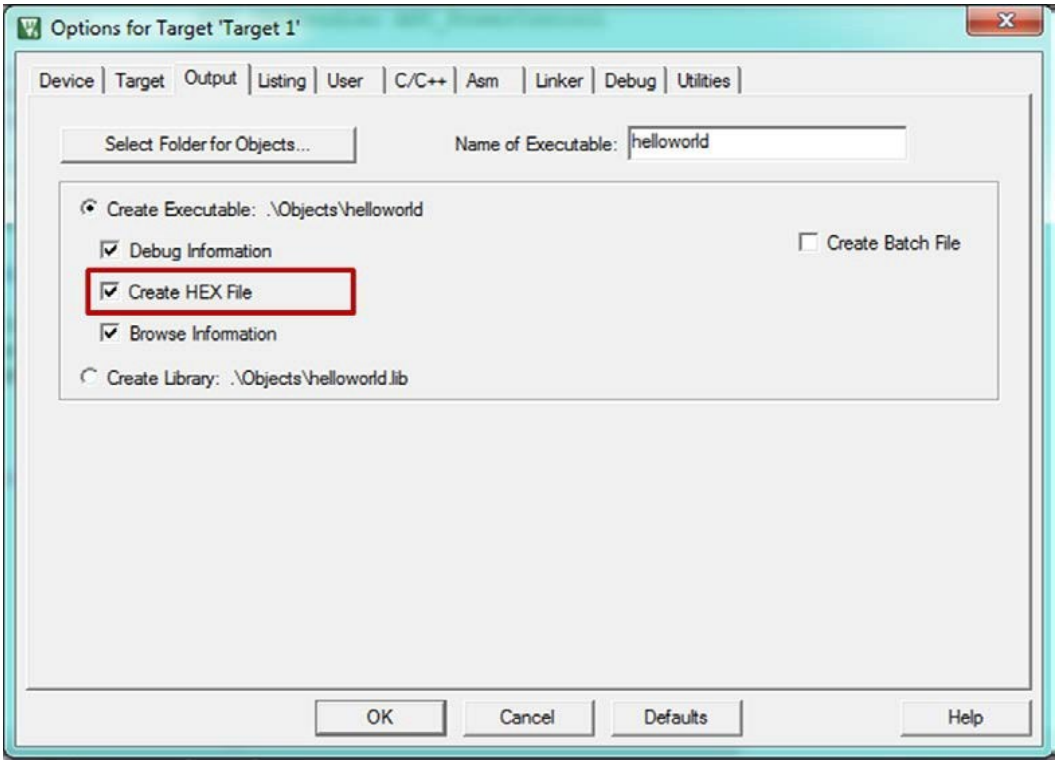

Figure 2: Output Options

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).

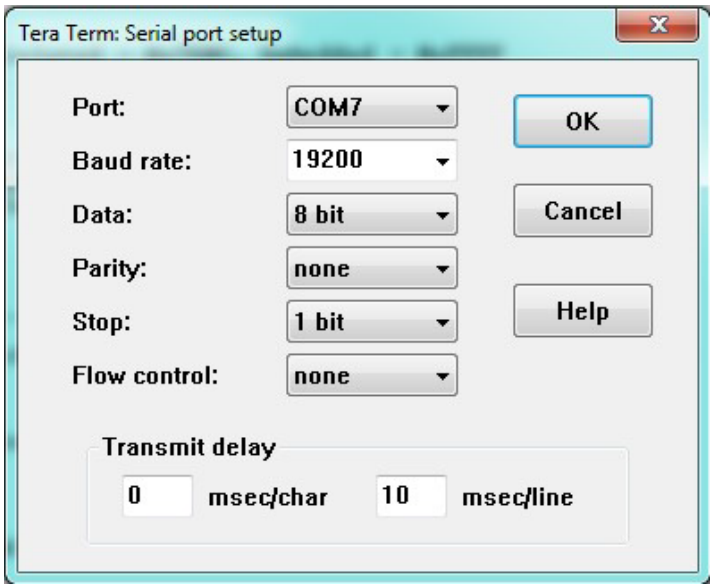

Figure 3: Serial Port Settings

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).

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 –fH, see below.

:>PGM -fH

a) Now, be sure all THREE of the following features are enabled: a) XON/OFF software flow control

b) 10ms line pacing

c) 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).

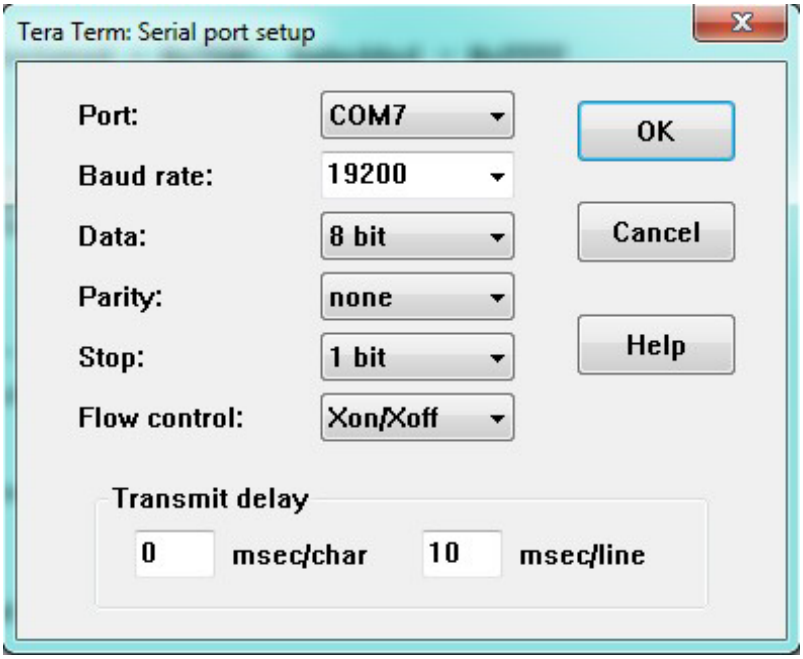

Figure 4: Serial Port Upload Settings

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

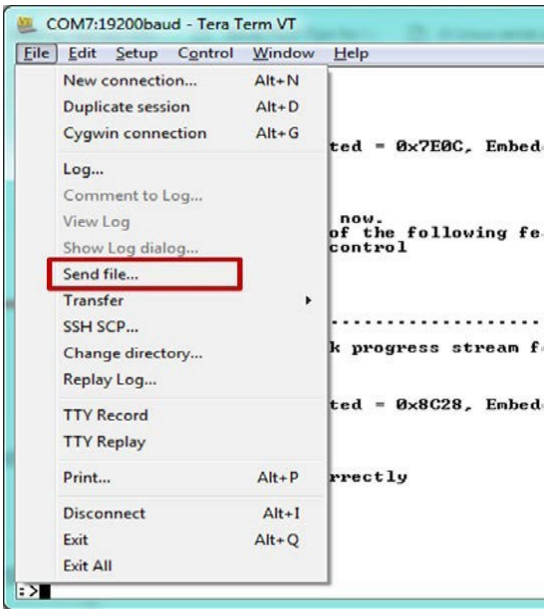

Figure 5: Send File

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

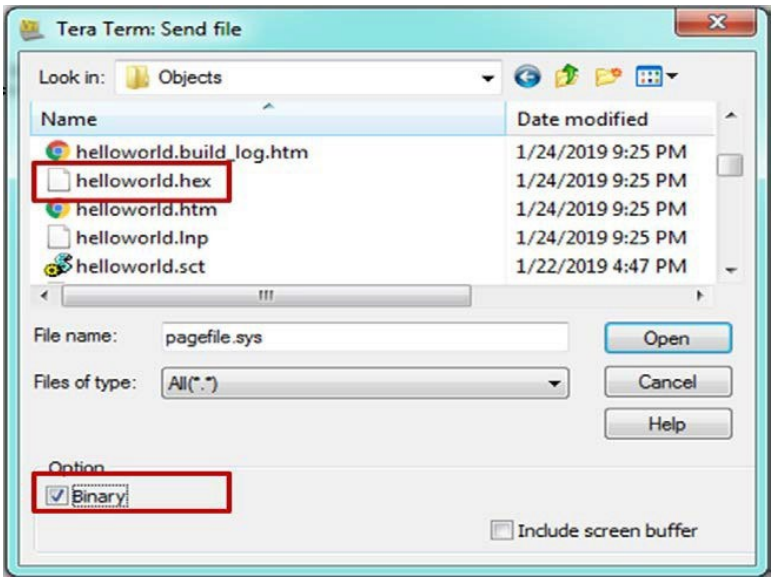

Figure 6: Open Hex File

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).

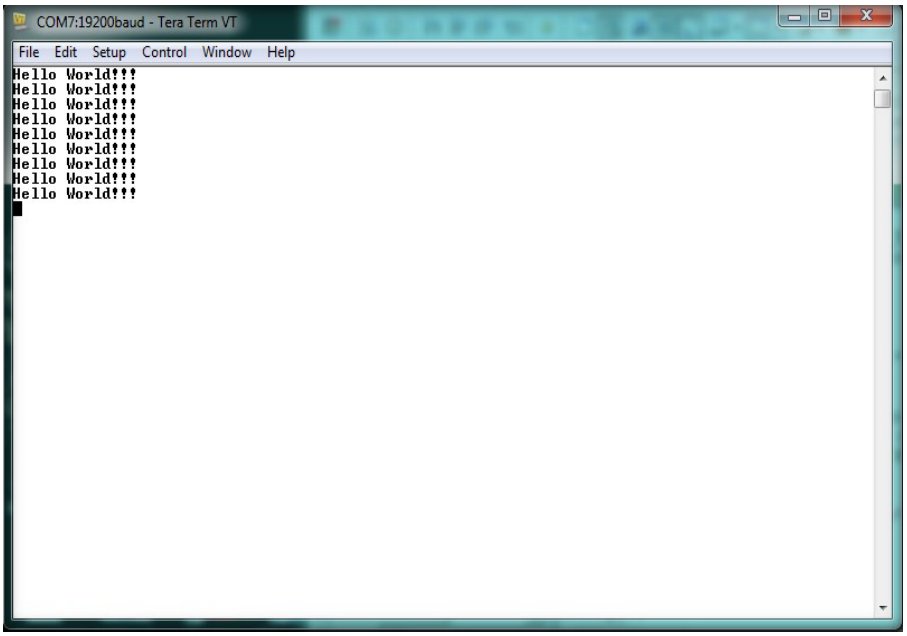

Figure 7: Hello World Display

Figure 8: Flash Download Commands

Revision History

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