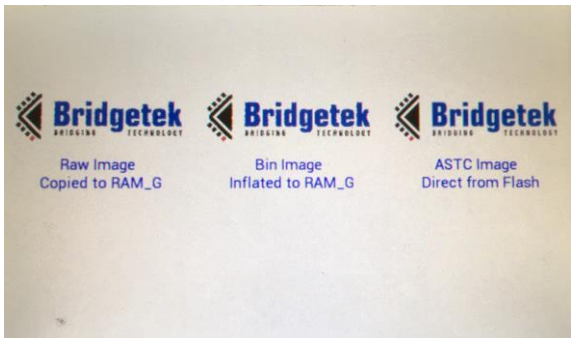
BRT_AN_025 Examples – Image from Flash

Introduction

In this example we demonstrate how to load an image from flash in three different ways.

- One image in raw format, copied from Flash to RAM_G
- One image in compressed (bin) format, inflated from Flash into RAM_G
- One image in ASTC format displayed directly from Flash

This demonstrates one key feature on the BT81x series which supports external flash attached directly to EVE and so avoids the need for your MCU to store the data in its flash memory.



Compatibility

EVE Family	Compatible	Notes
EVE1 (FT80x)	NO	
EVE2 (FT81x)	NO	
EVE3 (BT815/6)	YES	Requires EVE module with flash
EVE4 (BT817/8)	YES	Requires EVE module with flash

This example requires flash and so is only suitable for BT81x modules with flash chip fitted. Due to the coordinates chosen, the example is intended for an 800x480 or larger screen. For smaller screens, the code will still work well but you may need to adjust the coordinates of the VERTEX commands to get the images on the screen.

Associated Files

This example is provided with the following files. The files in blue are needed to run the example and the others are provided for reference.

- eve_example.c — Example code for the application
- eve_example.h — Header file for the example
- eve_fonts.c — Replaces the file provided with BRT_AN_025
- eve_images.c — Replaces the file provided with BRT_AN_025
- Flash Files
  - Flash_BT815_BT816 — Flash output files for BT815/6 (see flash.bin)
  - Flash_BT817_BT818 — Flash output files for BT817/8 (see flash.bin)
- For Reference
  - ARGB1555_ASTC — Converted files for ASTC (for reference)
  - ARGB1555_Bin — Converted files for Bin (for reference)
  - ARGB1555_Raw — Converted files for Raw (for reference)

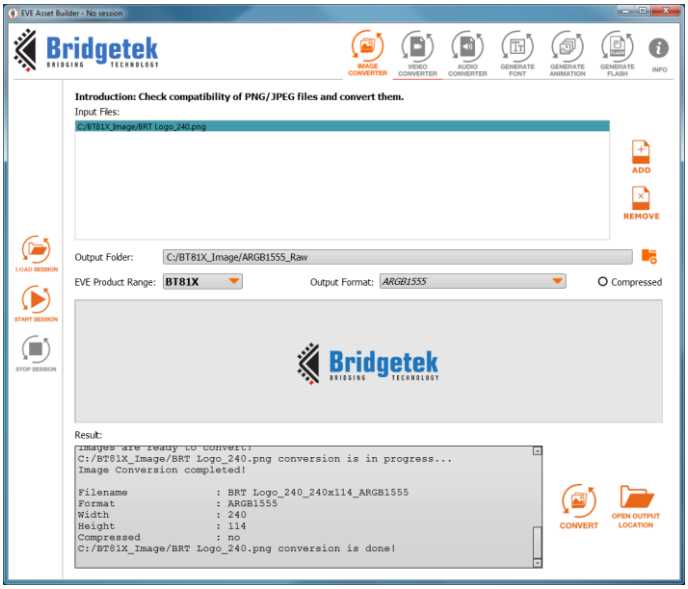
Description

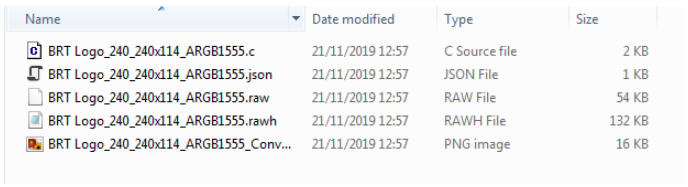
The example will demonstrate the following:

- Put the flash into the full speed mode. It will display an error message if this fails.
- Display three copies of the BRT logo:
  - One image in raw format, copied from Flash to RAM_G
  - One image in compressed (bin) format, inflated from Flash into RAM_G
  - One image in ASTC format displayed directly from Flash
  - Some text using the new line feature of the text command

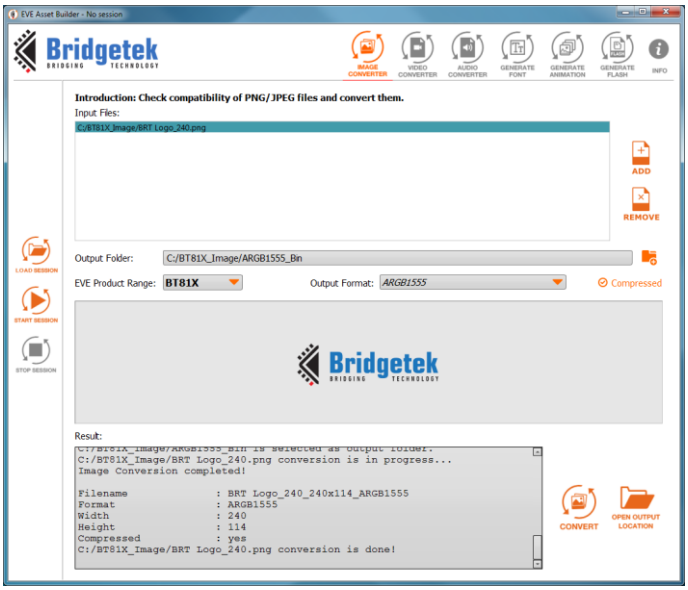
EVE Asset Builder was used to convert the provided logo into the three formats.

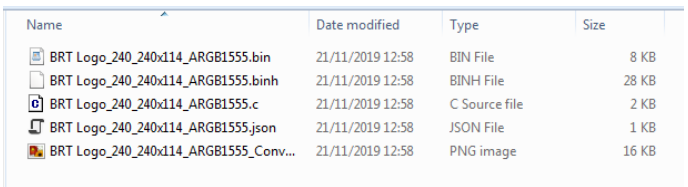
First, convert the png logo to a raw format in ARGB1555



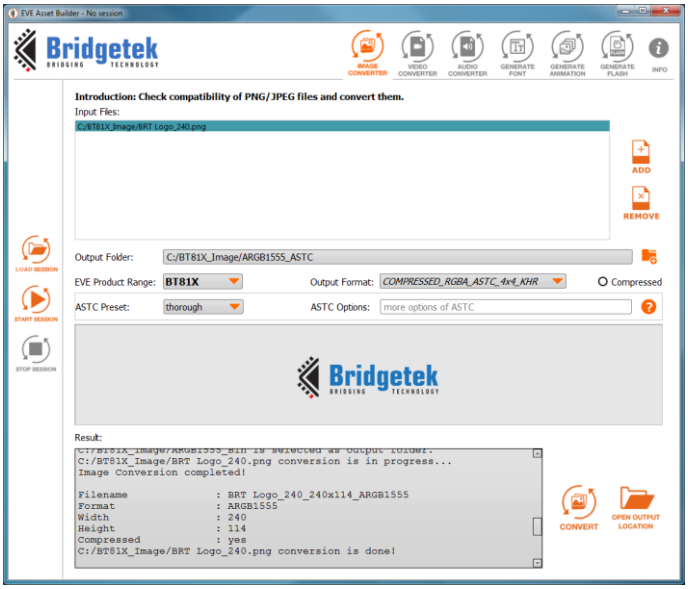


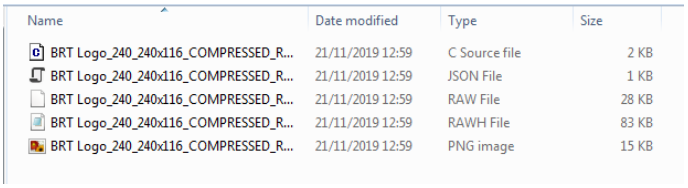
Next, convert the png logo to a compressed format in ARGB1555



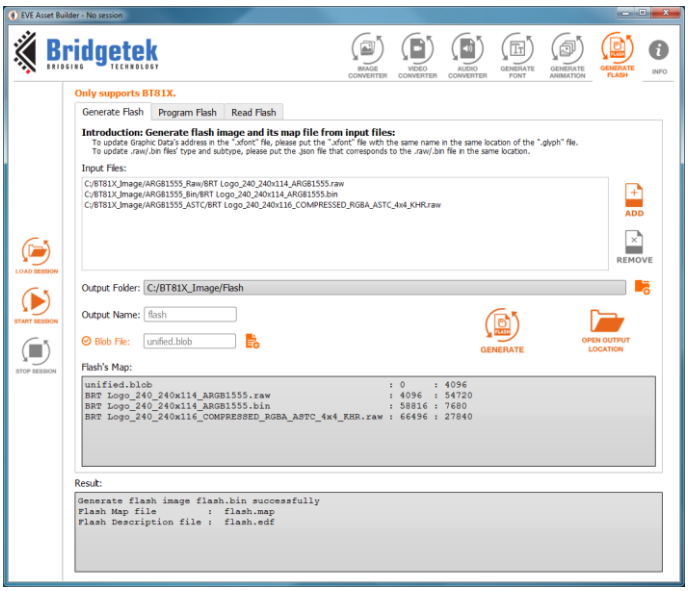


Finally, convert the png logo to an ASTC format





Now, build the flash bin file by adding the three images. The map file shows the starting addresses and sizes after converting



Running the Example

You will require to have the BRT_AN_025 code example running on your MCU as it has the supporting framework needed by the code provided with this example. You can also use this code with other frameworks but you may need to adjust the syntax of the calls to match the functions in your EVE library.

We recommend to check the default example provided as part of the BRT_AN_025 code first to ensure the settings are all correct and it is working well.

Refer to the main BRT_AN_025 document available from www.brtchip.com for details of the following steps. (see section *Running the BRT_AN_025 Examples from Github*)

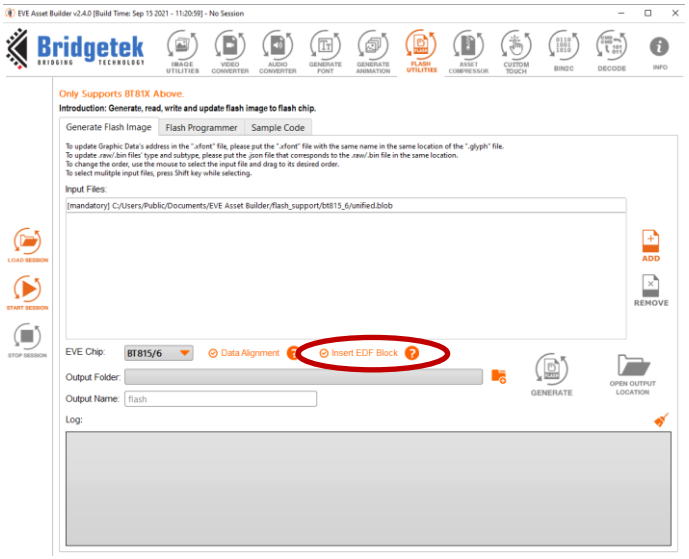
To run this example:

- Load the Flash of your EVE module with the provided .bin file using EVE Asset Builder (select the flash.bin from the correct device folder Flash_BT815_BT816 or Flash_BT817_BT818)
- Copy the example code to your project. Beginning with the standard BRT_AN_025 download, and after ensuring that it compiles and runs)
    - Replace the entire content of the eve_example.c file in your BRT_AN_025 code with the eve_example.c code provided in this example.
    - Replace the entire content of the eve_example.h file in your BRT_AN_025 code with the eve_example.h code provided in this example.
    - Replace the entire content of the eve_fonts.c file in your BRT_AN_025 code with the eve_fonts.c code provided in this example.
    - Replace the entire content of the eve_images.c file in your BRT_AN_025 code with the eve_images.c code provided in this example.
- Build the project, program your MCU and run the code
- You should see the screen display the three copies of the logo as illustrated in the introduction.

Extending the Example

This example is provided to demonstrate the basic principles of using images from flash. When using it in your real application, you can use parameters instead of the hard coded numbers for the addresses etc. which will be helpful if you need to use a large number of images and keep track of where they are in flash and RAM_G.

Note that the latest version of EAB (from 2.4.0 onward) have a new feature called the EDF block. When enabled, it will add a block immediately after the BLOB file (EDF begins at 4096) which contains an index of the flash contents as detailed below. This allows the code to retrieve the information without the developer needing to manually refer to the map file and coding in the addresses.



The format of EDF Block is defined as below:

```
- First 4 bytes is block length.
- Remaining bytes are each asset information stored in a binary structure.
typedef struct Eve_Flash_Asset_Info_t {
uint16_t assetID;
uint32_t startAddress; // offset from 0
uint32_t size; // count in bytes
uint8_t type;
uint16_t subType;
uint8_t compressionMethod; // 0: raw (no compression), 1: bin (deflation)
uint16_t width; // applicable to image/video/animation only
uint16_t height; // applicable to image/video/animation only
};
```