Documentation on User Specify Flash Program in i.MX21 HAB ToolKit for TO2

This document is used to program some guide for the user to modify the flash program so that they can compile the Flash program as binary and download through i.MX21 HAB ToolKit v2.0.exe.

Please also refer to the example project "NOR32 Flash Loader v2.1 Bootstrap"

Add \Hab Tools.c and tht_memory_map_defines.h to the project.

Please remove all the printf statement in the code.

1. main()

For the Flash program `main()` function, please follow the following

```
void main ()
{
     U32       SourceAddress = *(volatile U32*)(0xC000FFF0);
     U32       TargetAddress = *(volatile U32*)(0xC000FFF4);
     U32       NumberOfByte  = *(volatile U32*)(0xC000FFF8);

     U32  Word = NumberOfByte / 4;

     CheckHABStatus();
     FlashLoader(TargetAddress, SourceAddress, Word);
}
```

`SourceAddress` is the SDRAM address that image was downloaded to sdram. This field is located in 0xC000FFF0.

`TargetAddress` is the Flash Address that image to be programmed, this field is located in 0xC000FFF4.

`NumberOfByte` is the image files size to be programmed in bytes, this field is located in 0xC000FFF8.

`CheckHABStatus()` is the function call to check the HAB status.

`FlashLoader(TargetAddress, SourceAddress, Word)` is the function to call the flash program including erase, write and verify.

2. FlashLoader(TargetAddress, SourceAddress, Word)

For the `FlashLoader(TargetAddress, SourceAddress, Word)`, this is the flash function all program that includes the flash erase, flash program and flash verified.

```
void FlashLoader(TargetAddress, SourceAddress, Word)
{

     U32  nAddress;
     U32  Errors;
     U32  HAB_Status;

     // Status Code
     U32 FlashComplete         = 0x67676767;
     U32 FlashTimeOut          = 0x57575757;
     U32 FlashError            = 0x47474747;

     nAddress = fAddress + (nWords - 1) * sizeof(fAddress);
     // Flash Erase
     FlashSectorErase(fAddress,nAddress);

     // Flash Program
     FlashWrite(fAddress, (U32*)rAddress, nWords);

     // Flash verify
     Errors = FlashVerify(fAddress,rAddress,nWords);

     if (Errors)
     {
          CheckFlashComplete(FlashError);
     }
     else
     {
          CheckFlashComplete(FlashComplete);
     }
     return;
}
```

Inside the `FlashLoader()`, after the flash was programmed, the flash program must return the status, either Flash is ok [`CheckFlashComplete(FlashComplete);`] or Flash Error [`CheckFlashComplete(FlashError);`]

3. FlashSectorErase(x)

This is the erase portion of the Flash program, please add `CheckFlashComplete(0x17171717);` to feedback the status of erase, this can be sent periodically during the erase process, say about once in 0.5s.

4. FlashWrite(x)

This is the write portion of the Flash program, please add `CheckFlashComplete(0x27272727);` to feedback the status of write, this can be sent periodically during the write process, say about once in 0.5s.

5. Flash Verify(x)

This is the verify portion of the Flash program, please add `CheckFlashComplete(0x37373737);` to feedback the status of verify, this can be sent periodically during the verify process, say about once in 0.5s.

6. Image requirement

The Flash program image must compile at 0xC0004000 with size < 0xBFF0.
And the Flash program will be download to address 0xC0000000. After generated the FlashLib.bin image, please use "FlashLibTools.exe" to create the FlashLib that used to input to the HAB Toolkit.

7. Input the Flash Library

Please select Flashing with User Specify and Browse the flashprogram.bin

8. Save the setting as profile

After select the image file, user can save all the setting using the Load/Save Profile to avoid configuration for each time