Encrypted USB Host Bootloader System

asmallri@brushelectronics.com
www.brushelectronics.com

Brush Electronics' Encrypted USB Host Bootloader has been developed to support remote firmware upgrade for the Microchip PIC base Microcontroller systems deployed in the field. The Encrypted USB Host Bootloader is built on top of Microchip Inc standard USB Host Bootloader and the decryption logic derived from our Encrypted Bootloaders that operated on Encrypted image files utilizing the XTEA (eXtended Tiny Encryption Algorithm).

The Brush Electronics' Encrypted USB Host Bootloader System comprises of two main elements:

1. Hex File Encrypter application
2. Bootloader Firmware resident in the target hardware platform

For the sake of subsequent explanation, the term Bootloader refers to the code (firmware) executing in the PIC microprocessor (target) and the Encrypter refers to the application that encrypts the original hex file.

The Brush Electronics' Encrypted USB Host Bootloader is available for the Microchip PIC24/dsPIC and PIC32 families of Microcontrollers as supported by the underlying Microchip USB Host Bootloader.

Details for setting up the Bootloader on an Explorer 16 and for using th can be found in the in the file Getting Started - Running the Host - Bootloaders - Thumbdrive Bootloader.htm located in the Documentation sub directory.

Encrypter Application

The XTEA.EXE Encrypter application is a Windows console application that accepts either one or three command line arguments. These arguments are the source file name of the standard Intel hex file to be encrypted, the 16 byte *Cipher Key* and the number of *Iterations* the cipher should be applied to the cipher text. The *Cipher key* must be exactly 16 bytes and must match the hard coded *XTEA_Key* in the Bootloader source code. The *Iterations* (typically 16) must match the hard coded *XTEA_Iterations* in the Bootloader source code. If only the source filename is specified on the command line then the XTEA's applications hard coded *XTEA_Key* and *XTEA_Iterations* constants are used.

The Encrypter generates the encrypted output file using the same filename as the source substituting *.cry* for the file extension.

Usage: XTEA sample.hex

Usage: XTEA sample.hex 123helloworld321 16

Customization

The bootloader must be customized to support different hardware platforms (targets) and microcontrollers. In general the customisation requires the following steps:

- Create a target "HardwareProfile - <your variant>" header file for your target hardware platform. An example of such a file is the "HardwareProfile - PIC24FJ256GB110 PIM.h" located in the bootloader installation directory.
- Modify the supplied HardwareProfile.h file to including the target HardwareProfile header file generated from the previous step
- In the file boot_main.c and boot_config.c configure the fuse setting for your specific target
- Modify the boot_XTEA with the required encryption key parameters ensuring they match the command line parameters to be passed to the XTEA encrypter application.
- Configure bootloader status reporting, if required, in the file boot_config.h
- In the Microchip IDE, select the processor in the menu Configure / Select Device
- Create the processor specific bootloader linker script for compiling the user application to co-reside with the bootloader. Refer to the sample application linker scripts supplied with the bootloader package which contains

Limitations of the Bootloader

The following limitations of the Bootloader must be taken into account:

- The target's registers are not preserved by a RESET
- No support for WDT fuse bit. WDT support if required must be implemented by the software enabled WDT feature
- The Bootloader ignores Configuration Records and ID records
- The bootloader requires remapping of the processors interrupt vectors. This is accomplished by adding the "MSD Bootloader Remapping.c" file to the source files folder of the project. This file can be found in the "<install directory>\Application Files\Interrupt Remapping" folder. Modify the "MSD Bootloader Remapping.c" file as required for the application.

Need Something Special?

What if you need some unique feature added to the Bootloader or a Bootloader developed for some other product? Brush Electronics specializes in the development of Bootloaders for Microchip Microcontrollers and welcome the opportunity to work with you to develop a custom product that meets your specific needs.

Brush Electronics
2 Brush Court
Canning Vale
Western Australia 6155
Australia

Tel: +61 (0) 894676358
Email: info@brushelectronics.com
www: www.brushelectronics.com
