

API for flash controller opens realm of custom applications

By Damien Col, Hyperstone

This article discusses the many applications accessible through Hyperstone Application Programming Interface (API). Users now have the option to supplement dedicated NAND flash management firmware from Hyperstone with their own Customer Firmware Extension (CFE) developed with the API.

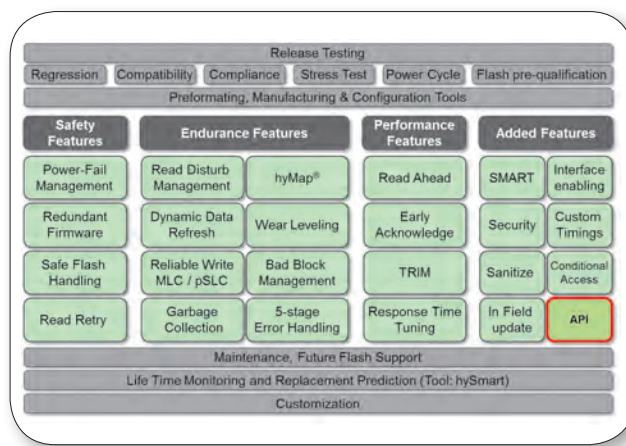


Figure 1. hyReliability flash management with API enhancement

Hyperstone have earned the reputation of providing the market with the most robust embedded storage solution by implementing proprietary Intellectual Property (IP) and patented techniques to manage the flash memory. Its flash management firmware is the cornerstone of its expertise, and the guarantee for long-term support for new flashes. With the addition of an Application Programming Interface (API) layer, and its unique offer among all flash memory controllers, Hyperstone now becomes a system solution enabler.

The API increases the realm of possibilities, by allowing customers to use world class Hyperstone Flash Management firmware and develop a customer specific firmware build, whether it is to add security or safety features, or to add interface to sensors, or communication interfaces for example. The ultimate goal is to integrate key differentiators. The developing party fully controls the ownership of the Customer Firmware Extension (CFE) and therefore protects its own IP. The CFE is completely independent of the firmware, and as such, can be reused on new generations of flash memories but also on different controllers, an assurance for the same long-term support.

As described in figure 1, API is another brick added to an already extensive set of advanced features to enhance the safety, reliability, endurance

and performance of the latest generation of NAND flash memories, including 3D-NAND, and to enable maintenance and diagnostic required in the most demanding applications.

The API provides access to embedded firmware features and additional interfaces (figure 2) through the host interface in order to implement customized application added-value features, enabling a level of differentiation driven solely by the user's own expertise in the specific market to be tackled. The customer retains full control of his added functional features. The code can be changed at individual convenience, and never needs to be disclosed to Hyperstone.

The CFE becomes part of the final firmware code and is integrated alongside flash management advanced features. A majority of them are executed in the background and are transparent to the user. Those are associated with the safety, the endurance and the performance of the flash memories. These are the results of Hyperstone expertise to bring out the best of each generation of flash memories (SLC, pSLC, MLC, TLC, 3D-NAND).

The user binary can be written or updated by simple data write commands (prefixed by a vendor command that unlocks access to the reserved address range). As such, the customer

code is also maintained under the same flash data management procedures (Wear Levelling, Error Correction, etc). If the customer data is moved, it will be handled by the flash translation layer (FTL) which translates logical to physical accesses. By means of call back functions, customers can intercept for example read or write commands, and add own routines. Also for example, based on vendor commands, users can establish communication between any kind of application software running on the host system with either the controller basic firmware (e.g. SMART) or certain of its hardware resources (e.g. AES, SPI, I2C or ISO7816) as well as tunnelling commands through the controller to a connected device such as a smartIC via the ISO7816 UART or a WLAN baseband. As an example, the CFE can have access to a range of features of Hyperstone advanced flash firmware, allowing it to easily manage its own private data outside the commonly user-accessible address space of a storage device.

As the customer binary is not dependent on the flash type used, the API program indeed only needs to be developed once and the same program can be used also if the flash type or Hyperstone controller changes. This way the system can always run with the latest flash technology, but without any additional development effort. Hyperstone firmware and the

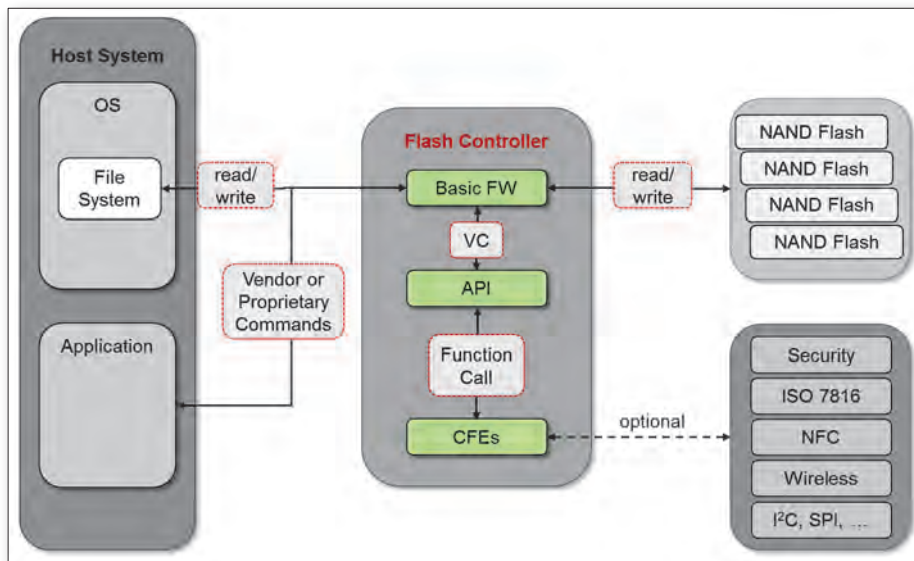


Figure 2. API hardware and firmware eco-system

CFE work in osmosis (figure 3). Hyperstone brings all its expertise in the flash management firmware, and will update it to support any new or changed technology, directly benefiting from its close relationship with the flash manufacturers. Complementarily, the CFE is the result of the customer expertise in its field of application. Both work hand-in-hand, but are completely independent from each other, hence providing a full flexibility. API serves several of our customers' needs: it enables new applications, strengthens them in their domain of expertise, creates differentiation features against one another, facilitates new feature development without being limited by Hyperstone resources, and enables new firmware feature development. It would be impossible to list the number of applications that can be addressed with the API. However, hot topics are certainly secure

storage and secure transactions. For example, certain applications require a dedicated proprietary firmware in order to utilize the ISO 7816 or SPI interface, to implement encryption key management, and to communicate securely and reliably with a host application. This includes: secure storage, mobile payment, mobile Pay TV, secure mobile communication (data/voice), digital rights management (DRM) (conditional access, license management, content protection), data privacy (e.g. medical), FIDO U2F security key (anti-phishing protection/built-in smart card) with AES data encryption, authentication and IoT for wearables. In addition, other serial interfaces available on Hyperstone controllers can be enabled and made accessible for applications. Behind the scenes, the CFE is stored as binary code in a reserved range not presented to the host (figure 4). This is called the CFE load area.

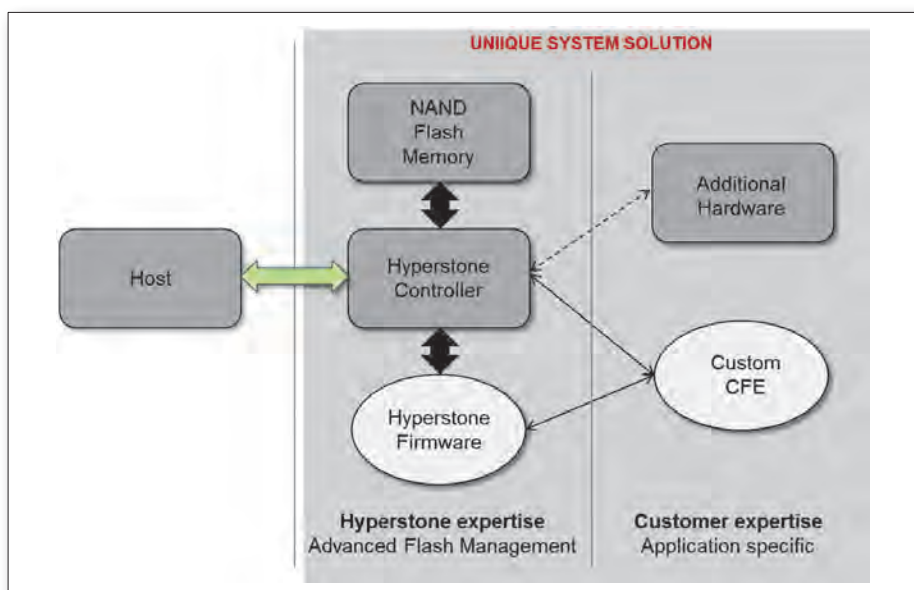


Figure 3. Hyperstone hardware and firmware + flash + CFE = unique system solution

The source code of such CFEs is solely owned and controlled by the customer. A customer firmware extension (plug-in) binary file, also referenced as CFE, consists of: an authentication section (always located at the beginning of the file), a header section (located directly after the authentication section), and up to three data sections (static, volatile, overlay). This is however transparent to most developers who will not have to worry about it. All firmware extension binaries are encrypted with AES-256 in CBC mode.

To execute API commands through vendor commands, it suffices to register the vendor command in the CFE (either as DataToHost-Callback or DataFromHostCallback) and to execute the vendor command using the proprietary Hyperstone hsfmt tool, delivered with the kit. The tool is supplied as C source code and can be compiled for different operating systems. Different options exist for debugging, such as using the debug interface of Eclipse or the debug-host via the debug-UART interface.

For further information, specific application notes are available. Breakpoints can be set dynamically at runtime. A simple debug possibility is the capability to send (debug) strings with the debug-UART and print them on the host. The key advantages for the API user can be summarized in the following set of added values: full control over application IPs, full control of releases, no synchronization needed with Hyperstone firmware releases, no source code transfer required, unique differentiator through application IPs, and fast software development. The API kit is available for purchase and without recurrent licensing cost including API function support from Hyperstone and firmware field update (FFU).

Different models exist for the development of new applications. The API software development kit (SDK) can be offered to system integrators or end customers directly. It is not required to understand the complex flash management features to develop a CFE. The kit makes it possible for anyone with basic firmware knowledge to develop a new application and implement key differentiators. In some other cases, it is also possible to out-source the tasks to third party software development service providers.

With Hyperstone flash controller API, many applications are imaginable to enhance the value proposition of your system. Security applications can be explained in more detail as an example, one among many others. With API, security features embedded in the Hyperstone controller, like AES encryption engine, SHA engine, secure random number generator can be enabled and managed. It allows the implementation of arbitrary security features

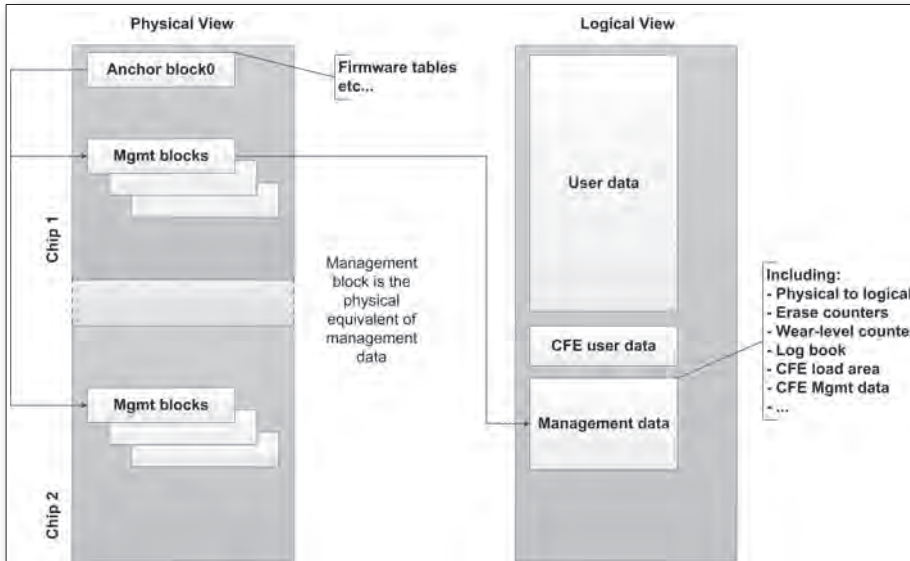


Figure 4. Anchor block and corresponding CFE load area (hyMap)

and access control systems. In this context, having full ownership of your system is vital. The realm of such applications is limitless,

but to mention a few: build own access control system (key management/hashing), build and manage private data space, secure read/

write, user key or automatic key generation, encrypted keys, and crypto erase, the fastest way to invalidate data. But the applications do not end here. As a further example, a wireless module can be connected to the flash controller, to get lifetime information, or to save or restore a flash memory image wirelessly.

Hyperstone flash controller API enables the development of dedicated proprietary features through a user-friendly environment, delivered in a kit. Using API, a customer firmware extension can be developed under the strict control of the system integrator, who keeps full ownership of the developed value-added application feature. The serenity of the CFE is also insured as it can be transferred to new generations of flash, without any additional investment. The porting effort across different Hyperstone controllers is minimal. The API is available for S8, U9 and subsequent designs. It has been used for applications now in mass production. Hyperstone will demonstrate the API at Embedded World (Hall 1/Stand 1-301). ■