The factory loader application is designed to be used with a companion script that implements the factory loader HIO commands over UART. Figure 1 shows a block diagram of this process.

## Command Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Description** | **Input** | **Output** |
| flash\_write | Writes image file to target address in flash | file\_image, flash\_addr | N/A |
| enroll | Enrolls key contents in flash | keyfile | N/A |
| from\_json | Programs partition table into flash | Partition JSON configuration file | N/A |
| to\_json | Reads partition table from flash into a json file |  | Partition information in JSON format |
| get\_status | Returns the status of *factory loader* application |  | Returns 0 if factory\_loader application is loaded |
| provision\_ap | Provisions AP parameters and connects to AP | ssid, passphrase | Returns 0 on successful connection |
| get\_http\_bin | Downloads and flashes any binary file to target flash addr | url, port, filename, flash\_addr | Returns 0 on success |
| get\_http\_file | Downloads file and writes it into target file system | url, port, filename, mount\_point | Returns 0 on success |
| write\_file | Writes a file to target file system | mount\_point, filename | Returns 0 on success |
| file\_enc | Encrypts existing file in the file system. Requires that a keyfile has been previously enrolled | mount\_point, filename | Returns 0 on success |
| mount\_fs | Mounts a file system | mount\_addr, img\_size, mount\_point | Returns 0 on success |

Table 2: Commands with description, input, and output