**Memory Map:**

The flash memory of 28069F is divided as follows:

|  |  |  |
| --- | --- | --- |
| Start address | Length | Name |
| 0x3D8000 | 0x4000 = 16K | H |
| 0x3DC000 | 0x4000 = 16K | G |
| 0x3E0000 | 0x4000 = 16K | F |
| 0x3E4000 | 0x4000 = 16K | E |
| 0x3E8000 | 0x4000 = 16K | D |
| 0x3EC000 | 0x4000 = 16K | C |
| 0x3F0000 | 0x4000 = 16K | B |
| 0x3F4000 | 0x3FF8 = 16K-8 word | A |

**Boot sector**

The start address of the code is 0x3F 7FF6. (See the section on Boot Behavior below)

The boot sector must be Sector A unless we want to program the OTP, a risky business as an error can kill the DSP.

This means that the flash password cannot be dealt with without deleting the boot, and we assume that the flash remains unlocked all the time with the addresses 0x3F 7FF8 …. 0x3F 7FFF unprogrammed (reads 0xffffffffffffffff).

**Valid program range**

The least valid program address is 0x3D8000

The maximum valid program address is 0x3F3EFF

**Statistics and personality range**

The statistics range is 0x3F3F00 to 0x3F3FFF.

The contents of this address range are:

|  |  |  |
| --- | --- | --- |
| Address | Length | Contents |
| 0x3F3F00 | 0x40 | In the beginning God created the heaven and the earth. |
| 0x3F3F40 | 2 | End address of code |
| 0x3F3F42 | 2 | Adler-32 Checksum |
| 0x3F3F80 | 2 | CAN address of device - Burner |
| 0x3F3F82 | 2 | CAN address of device – J1939 |
| 0x3F3F84 | 2 | Serial number of the device. |
| 0x3F3F86 | 0x3A | Other management data TBD |

**Checksums**

Checksums are according to the Adler32 algorithm (https://en.wikipedia.org/wiki/Adler-32). This algorithm is preferred over standard checksum as it accounts not only for the values in the code but also their order. The following is an excerpt from the above Wikipedia link:

**Calculation**

An Adler-32 checksum is obtained by calculating two [16-bit](https://en.wikipedia.org/wiki/16-bit) checksums *A* and *B* and concatenating their bits into a 32-bit integer. *A* is the sum of all [bytes](https://en.wikipedia.org/wiki/Byte) in the stream plus one, and *B* is the sum of the individual values of *A* from each step.

At the beginning of an Adler-32 run, *A* is initialized to 1, *B* to 0. The sums are done [modulo](https://en.wikipedia.org/wiki/Modular_arithmetic) 65521 (the largest [prime number](https://en.wikipedia.org/wiki/Prime_number) smaller than 216). The bytes are stored in one 32bit unsigned integer, *B* occupying the two most significant bytes.

The function may be expressed as

*A* = 1 + *D*1 + *D*2 + ... + *Dn* (mod 65521)

*B* = (1 + *D*1) + (1 + *D*1 + *D*2) + ... + (1 + *D*1 + *D*2 + ... + *Dn*) (mod 65521)

= *n*×*D*1 + (*n*−1)×*D*2 + (*n*−2)×*D*3 + ... + *Dn* + *n* (mod 65521)

*Adler-32*(*D*) = *B* × 65536 + *A*

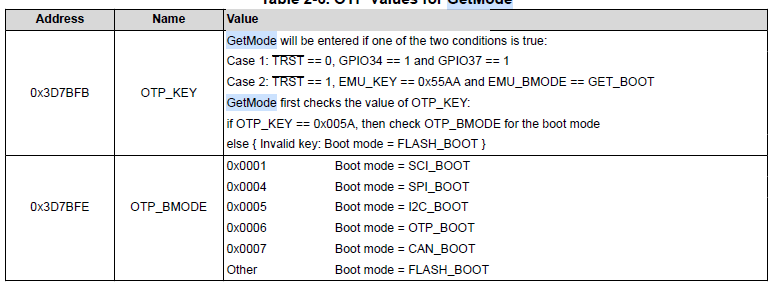
where *D* is the string of bytes for which the checksum is to be calculated, and *n* is the length of *D*.

# Appendix: Manual excerpts

## Boot behavior

Given that on reset wakeup the values are GPIO37 = 1 , GPIO34 = 1, the boot behaves by “GetMode()” – a function in the BootRom that scans OTP for further instructions.

The GetMode behavior is summarized below:



We assume that the OTP memory is pristine, so OTP\_KEY is not valid and the result shall be FLASH\_BOOT.

Excerpt from the Ref manual:

*Jump to flash is the default behavior of the Get Mode boot option. Jump to flash is also available as an*

*emulation boot option.*

*In this mode, the boot ROM software configures the device for C28x operation and branches directly to*

*location 0x3F 7FF6. This location is just before the 128-bit code security module (CSM) password locations.*

*You are required to have previously programmed a branch instruction at location 0x3F 7FF6 that will redirect*

*code execution to either a custom boot-loader or the application code.*