AN4434
Application note

CMOS F8H automotive EEPROM cycling endurance and data retention

Introduction

The automotive applications require a very high level of reliability and robustness, each single electronic device must therefore offer an ultra-high level of quality. In order to fulfill these requirements, STMicroelectronics developed a new family of advanced automotive EEPROM MXXxxx-A125/A145, based on a new improved architecture and produced with the CMOS F8H automotive process.

This application note details the improved cycling and data retention performances of the advanced automotive EEPROM MXXxxx-A125/A145 products, detailed in *Table 1*.

Table 1. Applicable products

Type	Serial interface	Root Part Numbers
Standard Serial EEPROM	I^2C bus	M24C02-A125, M24C04-A125, M24C08-A125, M24C16-A125, M24C32-A125, M24C64-A125 M24128-A125, M24256-A125, M24512-A125 M24M01-A125.
	SPI bus	M95020-A125, M95040-A125/145, M95080-A125/A145, M95160-A125/A145, M95320-A125/A145, M95640-A125/A145 M95128-A125/A145, M95256-A125/A145, M95512-A125/A145 M95M01-A125/A145.
	MicroWire bus	M93C46-A125, M93C56-A125, M93C66-A125 M93C76-A125, M93C86-A125

Contents

List of tables

List of figures

1 Cycling endurance

This section intends to offer all details concerning the cycling capabilities of the CMOS F8H EEPROM cell used in Automotive EEPROM MXXxxx-A125/A145[a].

1.1 Cycling values specified in automotive MXXxxx-A125/A145 datasheets

The automotive MXXxxx-A125/A145 EEPROM devices offer outstanding cycling performance, as detailed in *Table 2*.

Table 2. CMOS F8H process Automotive devices (-40°C to +145°C)

Products	Each cell inside an MXXxxx-A125/A145 EEPROM can be cycled:
M95xxx-A125/A145, M24xxx-A125 and M93xxx-A125 as defined in *Table 1*	4 million cycles (at 25°C) 1.2 million cycles (at 85°C) 0.6 million cycles (at 125°C) 0.4 million cycles (at 145°C)

1.2 CMOS F8H automotive process and MXXxxx-A125/A145 cycling performance

1.2.1 Cycling and temperature dependence

Glossary:

- ***Cycle*** = *Internal write cycle executed inside the EEPROM.*
- ***Cycling*** = *cumulated number of write cycles*

As specified in the CMOS F8H automotive MXXxxx-A125/A145 datasheets, the cycling endurance depends on the operating temperature (and is independent of the value of the supply voltage): the higher the temperature, the lower the cycling performance**.**

This safe cycling operating area can be represented by the following equation and/or by *Figure 1*.

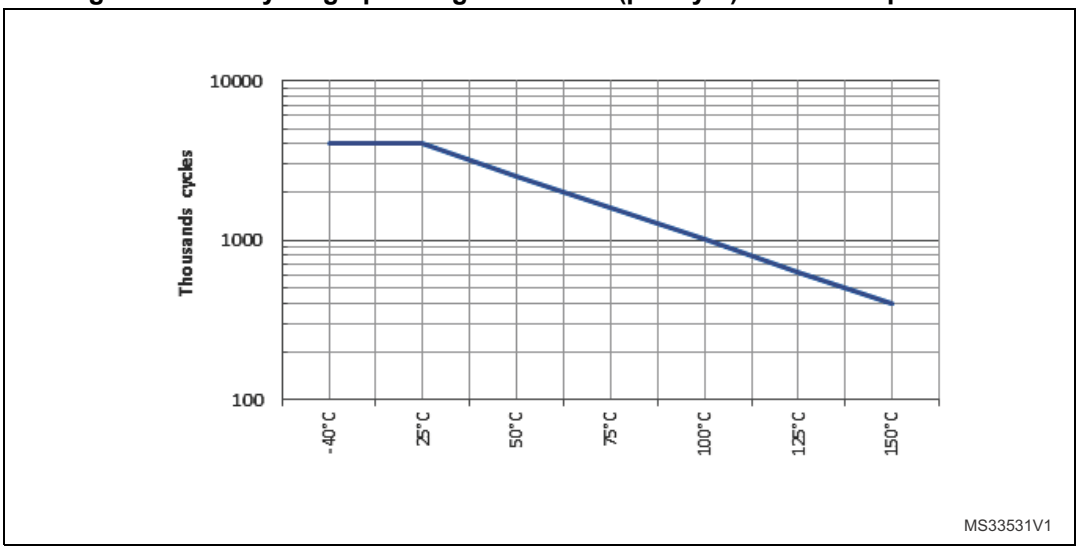
$$\text{Number of cycles} = 4\ \text{Million} * e^{(-k*(t° - 25))}$$

a. Automotive EEPROM based on CMOS F8H process are detailed in “MXXxxx-A125/A145” data sheets (ex: M24C64-A125 data sheet, M95256-A125 data sheet, M93C66-A125 data sheet)

Where:

- k = 0.018971
- t^o is defined in Celsius degree and is greater than 25°C

Figure 1. Safe cycling operating conditions (per byte) versus temperature(1)



1. For temperatures lower than 25°C, the safe operating conditions are considered as 4 million cycles (the real limit is actually higher)

For a robust application design, the safe cycling operating area shown in *Figure 1* has to be considered as a maximum cycling value for each byte of the memory(b), going above this safe operating area is not recommended.

Note: The cycling limits measured on the CMOS F8H automotive devices are much more above the safe conditions offered on Figure 1.

1.2.2 Cycling qualification method

The CMOS F8H automotive devices qualification procedure identifies the cell cycling intrinsic(c) performance over the full temperature range (-40°C to +145°C). During the qualification phase, the parts are cycled after then pass through a bake phase of 96 hours at 200°C. The memory content is then read in order to locate eventual failing bit.

In STMicroelectronics, the EEPROM intrinsic failure criterion is defined as 1 failing cell (or less) over 10 million cells.

Note: The cycling quality target is 0 within the 4 million cycles limit (specified in the datasheet at 25°C), thanks to the ECC architecture (this is explained further in Section 1.2.3).

b. For devices embedding the ECCx4 logic, it is important to notice that a Write on one byte is also cycling the 3 other bytes of the word (see *Section 1.2.3* for additional details)

c. intrinsic = belonging to the essential nature or constitution of the EEPROM die (extrinsic = originating from random event)

1.2.3 ECC and cycling budget

The Error Code Correction (ECC) is a specific logic, embedded in all CMOS F8H automotive EEPROM devices, that is able to correct a single bit error when reading a byte.

ECCx1 Error Correction Code

The ECCx1 is implemented in the 1Kbit up to 16Kbit CMOS F8H automotive devices.

In EEPROMs with ECC x 1 logic, 4 bits of ECC are added on each data byte of the memory array: if a single bit out of the data byte happens to be erroneous during a Read operation, the ECC bits detect the wrong bit inside the data byte and restore the correct value. The read reliability is therefore much improved.

The Error Correction Code (ECC x 1) is transparent for the user.

ECCx4 Error Correction Code

The ECCx4 is implemented in the CMOS F8H automotive devices when the memory size is equal or larger than 32 Kbits.

In EEPROMs with ECCx4 logic, 6 bits of ECC are implemented for each group of four bytes, making up a word (word = 4 bytes + 6 ECC bits). If a single bit out of the four data bytes happens to be erroneous during a Read operation, the ECC bits detects the wrong bit inside the data byte and restores the correct value. The read reliability is therefore much improved.

The Error Correction Code (ECC x 4) is transparent for the user, however it is interesting to consider the cycling budget:

- Byte(s) write
  If one byte inside one group is written, the ECC function also writes/cycles the three other bytes located in the same group. Therefore the cycling seen by a word is the sum of the cycles seen by each byte inside the word and this sum must not exceed the cycling performance specified in the datasheet.

  Example 1: Cycling equally each byte inside a word.
  Rule is: for a cycling limit of N cycles, each byte 0, byte 1, byte 2 and byte 3 of a word can be equally cycled N/4 times so that the word cycling budget is:
  N/4 + N/4 +N/4 + N/4 = N cycles.
  Simple case: for 4 million cycles (at 25°C), each byte 0, byte 1, byte 2 and byte 3 inside a word can be equally cycled 1 million times.

  Example 2: Cycling unequally each byte inside a word.
  Rule is: for a cycling of N cycles, and for bytes inside the same word, byte 0 can be cycled A times, byte 1 can be cycled B times, byte 2 can be cycled C times and byte 3 can be cycled D times, so that the word cycling budget is A + B + C+ D = N cycles.
  Simple case: for a 4 million cycles (at 25°C), and for bytes inside the same word, byte 0 can be cycled 2 million times, byte 1 can be cycled 1 million times, byte 2 and byte 3 can be cycled 1/2 million times (2 + 1 + ½ + ½ = 4).
- Word write
  If the application writes data by word(s), the 4 bytes making up the word are always updated at the same time and the number of write cycles is pulled to the highest cycling possible value. The application designer has only to check that each word in the EEPROM never exceeds the cycling performance specified in the datasheet (for

devices specified as 4 million cycles, a word can be cycled 4 million times).
For more information, refer to the application note AN2440 on *www.st.com*.

1.2.4 Overall number of write cycles

When evaluating the cycling performances of an application, it must be stressed out that the number of cycles can be defined either for each memory cell or for the overall number of cycles decoded by the whole memory:

- the max cycling value defined in the datasheet is the max number of cycles for each byte
- the overall number of cycles is the number of cycles correctly decoded and executed by the device, spread over all addressed locations in the memory.

The characterization trials performed on automotive CMOS F8H products

(MXXxxx-A125/A145) equal or larger than 32Kbit have demonstrated that the overall number of write cycles exceeds 128 million cycles, at 25°.

1.3 Cycling strategy in the end application

In order to ensure the safest EEPROM cycling conditions, it is strongly recommended to evaluate the number of write cycles and the relative temperature profile of the cycling performed by the EEPROM during the life of the application, that is:

- define the main temperature ranges at which the EEPROM is operating in the end application,
- for each temperature range, estimate the number of write cycles executed for each data block,
- for each data block (with different cycling profiles), calculate the cumulated cycling effect using the following equation or *Table 3*.

$$\sum_{i=1}^{n} \frac{\text{Number of cycles at temp}(i)}{\text{Max Number of cycles specified at temp}(i)} \leq 1$$

Table 3. Application cycling profile evaluation(1)

Temperature	Number of Cycles(2)	% of the max cycling value specified in *Table 2*
25°C	w	(w/4M) x 100 = a %
55°C	x	(x/2.3M) x 100 = b %
85°C	y	(y/1.2M) x 100 = c %
125°C	z	(z/600K) x 100 = d %
Total	w + x + y + z	(a + b + c + d) %

1. The table can be adapted according to the temperature profile by taking care of putting down the maximum cycling for each temperature
2. w,x,y and z are forecast number of cycles for a specific data block

If the total percentage of cumulated cycles (last row in *Table 3*) is lower than 100%, the data stored in the EEPROM are safely cycled.

If the total percentage of cumulated cycles is above 100%, the intrinsic safe margin for cycling is exceeded and a data relocation strategy must be defined. This can be done by distributing the number of cycles over several memory locations as follows:

- define a cycling limit for each data block according to the application needs and product performance (as shown in *Table 3*).
- count the numbers of cycles executed on each data block (counter value can be stored in the EEPROM).
- when the counter exceeds the defined limit, the cycled data block must be relocated to another physically independent memory address. The software developer should define this new data block to be duplicated in a location inside a different page and, when possible, not with the same byte address inside the new page. The counter itself must also be stored in a new location.

In addition, to optimize the number of cycles in the EEPROM and keep the other data blocks safe in the memory array:

- define data classes (located in the same page) where data with similar update rates are gathered together. This will optimize the use of the Page mode instead of the byte mode.
- the areas containing the read-only parameters and the cycled items should be separated and made as much as possible independent from each other. Two types of data should not share the same pages and, where possible, the same locations inside the related page.
- for EEPROM with embedded ECCx4 (see *Section 1.2.3*), it is recommended to store data bytes sharing the same class inside N groups (ex: 9 data bytes should be stored in 3 groups, that is group1=4 data bytes, group2=4 data bytes and group3=1 data byte plus 3 bytes unused). In this way, another class of data (let's call it class2) cannot share a group used by class1 and each write cycle will concern either class1 data or class2 data but never class1 data and class2 data.

2 Data retention

This chapter intends to offer all details concerning the data retention capabilities of the Automotive EEPROM MXXxxx-A125/A145 products based on CMOS F8H process.

Glossary:

– *data retention: at t_0, bytes are written after what no Write is executed on these bytes. The data retention time is the time after t_0 during which the bytes can still be correctly read (the MXXxxx-A125/A145 being DC supplied or not).*

2.1 Data retention values specified in automotive MXXxxx-A125/A145 datasheets

The automotive MXXxxx-A125/A145 EEPROM devices (based on the automotive CMOS F8H process) offer outstanding data retention performances, as defined in *Table 4*.

Table 4. Data retention for CMOS F8H process Automotive devices

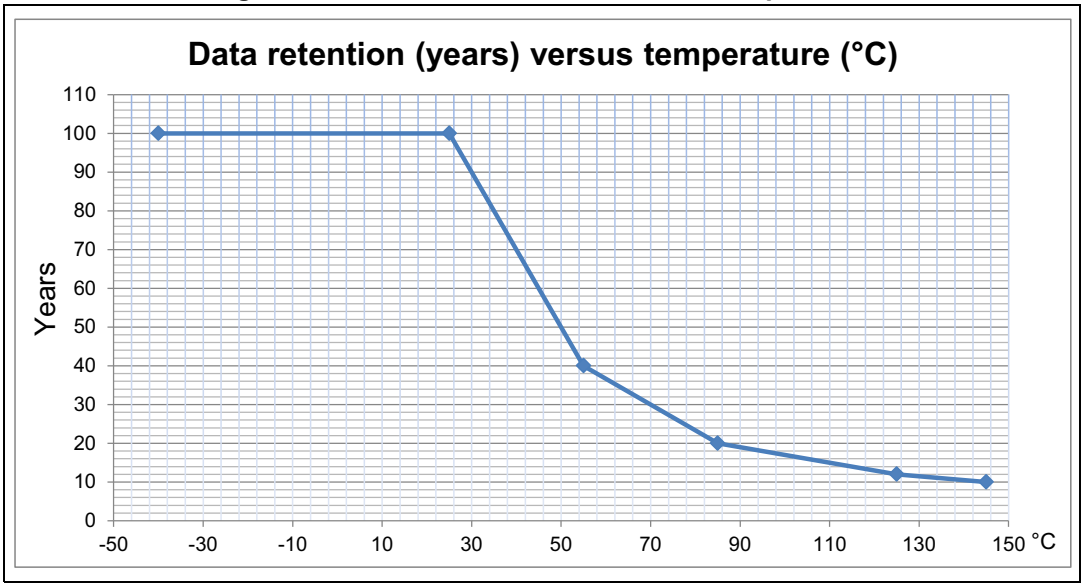
Products	Data retention
M95xxx-A125/A145, M24xxx-A125 and M93xxx-A125 as defined in *Table 1*	more than 100 years at 25°C more than 50 years at 125°C

2.2 CMOS F8H automotive process and MXXxxx-A125/A145 data retention performance

2.2.1 Data retention and temperature dependence

The CMOS F8H automotive EEPROM (MXXxxx-A125/A145) data retention is temperature dependent and is independent of the value of V_{CC}: the higher the temperature, the lower the data retention time (as clearly shown in *Figure 2*).

Figure 2. Years of data retention versus temperature



The data retention safe values are:

- more than 100 years (at 25°C or less)
- more than 40 years (at 55°C)
- more than 20 years (at 85°C)
- more than 12 years (at 125°C)[(d)]
- more than 10 years (at 145°C).

2.2.2 Data retention qualification method

Data retention qualification procedure for the CMOS F8H automotive EEPROMs checks that the data written into the EEPROM remain readable with a safe programming level. The ST qualification method is:

– the part is cycled according to the max cycling values defined in the data sheet, at several temperatures,

– the part is stored in an oven for 8000h @200°C (no supply voltage on pin V_{CC}),

– the part content is then checked.

d. After cycling the device 100 kcycles at 150°C, the data retention safe value is 50 years (at 125°C).

The data retention follows an Arrhenius law, this allows to extrapolate, from the different qualification tests performed at different temperatures, the CMOS F8H automotive data retention limits. These limits are above the safe values defined in datasheets and in *Figure 2*.

2.2.3 Data retention and bit failure rate

ECCx4 and data retention

CMOS F8H automotive EEPROM devices (with memory size equal to or larger than 32 Kbits) embed an internal ECCx4[e] logic which combines each group 4 bytes (ECC x 4) with the corresponding 6 additional ECC bits. As a result, if a single bit out of 4 bytes happens to be erroneous during a read operation, the ECC detects it and replaces this bit with the correct value in the stream of read data.

The main benefit of the Error Correction Code is that the ECC filters out extrinsic defaults and pushes the data retention limit far beyond the limit of the first extrinsic failing bit.

Why this?

Design and process rules used to make ST EEPROM are qualified to ensure EEPROM cells with a theoretical very long data retention time (in the hundreds of years range): this is the intrinsic cell quality. However, in the real world, one bit can lose faster its data due some unexpected events (like the SILC effect: Stress Induced Leakage Current): this is an extrinsic phenomenon.

From a significant amount of CMOS F8H automotive EEPROM wafers[f], the very first test flow (EWS[f]) allows to identify the number of failing EEPROM cells, this offers an image of the extrinsic quality of the CMOS F8H automotive EEPROM cell. The EWS results show that the total number of bits screened as failing over the cumulated number of bits tested (total number of EEPROM bits over the tested wafers) is less than one bit failing per 10 million tested bits. The probability P0 that 1 cell fails can be then defined as less than 10^{-7}.

This extrinsic probability P0 can be starting point statistical considerations, as explained hereafter.

Let's consider the case where the first bit losing its data[g] is located inside wordA. Losing one bit content is an extrinsic event which can be defined with an occurrence probability. Let's define this probability as P1 (probability that one bit out of the whole memory loses its data).

$$P_1 = nP_0$$

e. ECCx4 = Error Correction Code, specified in the related EEPROM data sheets (32Kb and larger capacity devices).

f. A wafer is a thin slice of semiconductor material on which the integrated circuits (like EEPROM) are processed. Once processed, the wafers are tested at EWS (Electrical Wafer Sort).

g. At this step, during a Read, the ECC will correct the failing bit, therefore wordA is read correctly.

Where:

– n is the number of bits in the EEPROM

– P0 is the probability that 1 cell fails

If one bit fails, the ECC architecture will correct this bit value and therefore P1 is not significant for the end user as the whole memory can still be read without a single bit error. If we consider now that a next extrinsic event would lead to one more bit losing its data, this bit being randomly located over the whole memory array, the probability P2 that this second falling bit could be located in the same word A is expressed as

$$P_2 = knP_0^2(k-1)/\ 64$$

Where k is the number of bits of one word (for ECCx4: 4 bytes and 6 ECC bits, this lead to k=4*8 +6=38).

The value of P_2 is depending upon the value of n (number of bits in the EEPROM); *Table 5* shows the probability P_2 value that one word is not read correctly (a second bit failing in the same word cannot be recovered by ECC logic) for different memory densities.

Table 5. ECCx4 memories: probability that one word is not correctly read

Memory capacity	P2 (ppm(1))
1 Mbit	0.23
512 Kbit	0.12
256 Kbit	0.06
128 Kbit	0.03
64 Kbit	0.01
32 Kbit	0.01

1. ppm: part per million

Note:
If we consider that, during the life of an application:

- *Data written only once (Read only data) are not stressed by further Write cycles, there is therefore no SILC effect and the data retention is not impacted. For these data, P_2 value is much less than the values given in Table 5.*
- *Data written several times may be impacted by the SILC effect but, after each Write cycle, the data are refreshed and therefore the data retention time is "reset".*

Based on these comments, it is wise to consider that the data retention in an application is significantly better than the worst case values computed in Table 5 and Table 6.

ECCx1 and data retention

CMOS F8H automotive EEPROM devices (1Kb to 16Kb) embed an internal ECCx1 logic that compares each byte with the 4 additional ECC bits. As a result, if a single bit happens to be erroneous during a read operation, the ECC detects it and replaces this bit with the correct value in the read byte. The main benefit of the Error Correction Code is that the ECC

filters out extrinsic defaults and pushes the data retention limit far beyond the limit of the first extrinsic failing bit.

If we use the same analysis detailed in *ECCx4 and data retention*, the probability P_2 that a second bit could be located in the same byte is equal to

:

$$P_2 = knP_0^2(k-1)/\ 64$$

Where:

- k is the number of bits of the word (for ECCx1: 1 bytes and 4 ECC bits, this leads to k = 12)
- n is the number of bits of the whole memory.
- P_0 is the probability that 1 cell fails ($P_0 = 10^{-7}$)

As the value of P_2 is depending on the value of n (number of bits in the EEPROM), *Table 7* offers the probability P_2 that one byte is not read correctly (a second bit failing in the same byte cannot be recovered by the ECC logic).

Table 6. ECCx1 memories: probability that one byte is not correctly read

Memory capacity	P_2 (ppm[1])
16 Kbit	0.001
8 Kbit	0.001
4 Kbit	0.000
2 Kbit	0.000
1 Kbit	0.000

1. ppm: part per million

2.3 Data retention strategy in the end application

The data retention time is defined in the datasheets with specific temperatures. In order to ensure the safest EEPROM data retention, it is wise to evaluate the amount of time during which the end application remains within some temperature range to evaluate the data retention profile, that is:

- define the time (in year) during which the EEPROM remains inside each temperature range (that is a typical temperature profile of the end application),
- for each temperature range, estimate the data retention value, in percentage, as defined in the following equation or in *Table 7*.

$$\sum_{i=1}^{n} \frac{\text{Number of years at temp}(i)}{\text{Max Number of years specified for temp}(i)} \leq 1$$

Table 7. MXXxxx-A125/A145 data retention profile evaluation example(1)

Temperature	Data retention (max)	Application ambient temperature (in years)(2)	% of data retention capability
-20°C	V = 100 years	v	(v/V) x 100 = a
25°C	W = 100 years	w	(w/W) x 100 = b
85°C	X = 20 years	x	(x/X) x 100 = c
125°C	Y = 12 years	y	(y/Y) x 100 = d
Total		v + w + x + y	(a + b + c + d)

1. The table can be adapted according to the temperature profile by taking care of putting down the maximum cycling for each temperature
2. v,w, x and y are the forecasted number of cycles for a specific data block

Example

An MXXxxx-A125/A145 automotive EEPROM (as defined in *Table 1*) is implemented in an application with a temperature profile defined as:

- 3 years at 125°C, that is 25% of the maximum data retention time at 125°C;
- 5 years at 85°C, that is 25% of the maximum data retention time at 85°C;
- 20 years at 25°C, that is 20% of the maximum data retention time at 25°C;
- 5 years at -20°C, that is 5% of the maximum data retention time at -20°C.

This application will keep safe data value during 33 years, with 75% of data retention capability (3 + 5 + 20 + 5 is 33 years, with 25% + 25% + 20% + 5% = 75% of data retention capability).

3 Revision history

Table 8. Document revision history

Date	Revision	Changes
24-Feb-2014	1	Initial release
22-Apr-2014	2	Changed document classification
05-Mar-2015	3	Updated *Table 1: Applicable products*, *Table 2: CMOS F8H process Automotive devices (-40°C to +145°C)*, *Table 3: Application cycling profile evaluation*, *Table 4: Data retention for CMOS F8H process Automotive devices* and *Table 7: MXXxxx-A125/A145 data retention profile evaluation example*. Updated *Figure 2: Years of data retention versus temperature*. Updated *ECCx4 Error Correction Code*, *Section 2.2.1: Data retention and temperature dependence*, *ECCx4 and data retention* and *Example* in *Section 2.3*. Added Footnote *d* in *Section 2.2.2*.