

Ezairo 8310-1

NVM Programming Information Note

Ezairo 8300 SDK 1.8 | Macronix MX25R1635F

Purpose

This note explains the NVM programming setting required when using Ezairo 8300 SDK 1.8 with Ezairo 8310 devices containing the Macronix MX25R1635F NVM.

Affected Hardware

Ezairo 8310 devices are available with different external NVM devices:

Device	NVM device
Ezairo 8310-0	onsemi LE25S161
Ezairo 8310-1	Macronix MX25R1635F

Important: The SDK 1.8 NVM project examples and documentation use the NVMUpdate device name `DEV_BOARD` by default. `DEV_BOARD` identifies the LE25S161 used on the Ezairo 8300 evaluation board and Ezairo 8310-0 hardware. It must not be used for an Ezairo 8310-1 device.

For Ezairo 8310-1, use the NVMUpdate device name `MX25`.

Why This Change Is Required

Although the LE25S161 and MX25R1635F use compatible SPI command sets, their device identification values and maximum chip-erase times are different. NVMUpdate checks the connected NVM device ID against the selected `--nvmdevice` profile and uses profile-specific timing values. Selecting `DEV_BOARD` for an MX25R1635F can therefore cause device identification or programming to fail.

The MX25R1635F may require up to 60 seconds for a chip erase, whereas the LE25S161 may require up to 2.4 seconds. Selecting `MX25` applies the correct device ID and erase timeout values.

Required NVMUpdate Change

Change the `--nvm` argument from `DEV_BOARD` to `MX25` for Ezairo 8310-1 hardware.

Read an Ezairo 8310-1 NVM

```
nvmupdate.exe i2c --nvm MX25 read --address 0 --length 81920 --out NVMreadback.hex
```

The corresponding command for LE25S161 hardware remains:

```
nvmupdate.exe i2c --nvm DEV_BOARD read --address 0 --length 81920 --out NVMreadback.hex
```

For programming, apply the same change to erase, program, and verify commands:

```
nvmupdate.exe i2c --nvm MX25 eraseAll --wipe
nvmupdate.exe i2c --nvm MX25 program --image blinky_nvm.hex --wipe
nvmupdate.exe i2c --nvm MX25 verify --image blinky_nvm.hex
```

Use the appropriate programmer configuration for the connected hardware, for example:

```
nvmupdate.exe i2c --nvm MX25 --config Pulsar_CM3_CFG program --image blinky_nvm.hex --wipe
```

Preserving Manufacturing Data

When updating an existing image and retaining manufacturing or calibration data, use `--keep_manufacturing` where appropriate:

```
nvmupdate.exe i2c --nvm MX25 --config Pulsar_CM3_CFG program --image blinky_nvm.hex  
--keep_manufacturing
```

Caution: `--wipe` overwrites existing NVM contents. Follow the product and manufacturing-data requirements for the target device before using it.

Changing an SDK NVM Project

1. Select **File > New > Project** in the Ezairo 8300 SDK IDE.
2. Select **SK5**, then select **SK5 NVM Project**.
3. Select the **Minimum** project type and finish the wizard.
4. Right-click the NVM project and select **Properties**.
5. Add the application ELF files under the application settings. Add bootable applications to **Applications to Boot** as required. Only one bootable application is allowed per core.
6. Open the **Build Stepstab**.
7. Locate the post-build NVMUpdate command and replace every occurrence of:

```
--nvm DEV_BOARD
```

with:

```
--nvm MX25
```

For example, change:

```
nvmupdate i2c --nvm DEV_BOARD program --image $ProjDirPath/Release/$ProjName.hex  
--keep_manufacturing
```

to:

```
nvmupdate i2c --nvm MX25 program --image $ProjDirPath/Release/$ProjName.hex  
--keep_manufacturing
```
8. Apply the project settings and build the NVM project. The output is placed in the project's Release directory. The generated .hexfile is the image used by NVMUpdate.
9. Confirm that the generated post-build command uses MX25 before programming an Ezairo 8310-1 device.

Production Script Updates

Any production script, Python application, batch file, or other automation that invokes NVMUpdate must select the correct NVM profile. For Ezairo 8310-1 hardware, update commands from `--nvm DEV_BOARD` to `--nvm MX25`.

Hardware	NVMUpdate selection
Ezairo 8310-0 / LE25S161	--nvm DEV_BOARD
Ezairo 8310-1 / MX25R1635F	--nvm MX25

There is no universal default in SDK 1.8 that safely replaces hardware selection. The programmer or production software must know which NVM is installed.

References

Relevant Ezairo 8300 SDK documentation topics:

- NVM Project Support in the Ezairo 8300 SDK

- Creating an NVM Project
- NVM Project Options
- Building the NVM Project
- Building and Using the NVMUpdate Tool

NVMUpdate command help is available with:

```
NVMUpdate --help  
NVMUpdate <command> --help
```

Summary

For Ezairo 8310-1 devices containing the Macronix MX25R1635F NVM, use --nvm MX25 in NVMUpdate commands and in the NVM project's post-build settings. Keep --nvm DEV_BOARD for LE25S161 hardware, including the original Ezairo 8300 evaluation board configuration and Ezairo 8310-0 devices.

onsemi, **Onsemi**, and other names, marks, and brands are registered and/or common law trademarks of Semiconductor Components Industries, LLC dba "**onsemi**" or its affiliates and/or subsidiaries in the United States and/or other countries. **onsemi** owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of **onsemi**'s product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent-Marking.pdf. **onsemi** reserves the right to make changes at any time to any products or information herein, without notice. The information herein is provided "as-is" and **onsemi** makes no warranty, representation or guarantee regarding the accuracy of the information, product features, availability, functionality, or suitability of its products for any particular purpose, nor does **onsemi** assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. Buyer is responsible for its products and applications using **onsemi** products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by **onsemi**. "Typical" parameters which may be provided in **onsemi** data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. **onsemi** does not convey any license under any of its intellectual property rights nor the rights of others. **onsemi** products are not designed, intended, or authorized for use as a critical component in life support systems or any FDA Class 3 medical devices or medical devices with a same or similar classification in a foreign jurisdiction or any devices intended for implantation in the human body. Should Buyer purchase or use **onsemi** products for any such unintended or unauthorized application, Buyer shall indemnify and hold **onsemi** and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that **onsemi** was negligent regarding the design or manufacture of the part. **onsemi** is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

ADDITIONAL INFORMATION

TECHNICAL PUBLICATIONS:

Technical Library: www.onsemi.com/design/resources/technical-documentation
onsemi Website: www.onsemi.com

ONLINE SUPPORT: www.onsemi.com/support

For additional information, please contact your local Sales Representative at
www.onsemi.com/support/sales