How to Use the Out of the Box Flash Interface

Table of Contents

Overview of the Flash Interface

The Out of the Box MAX V Development Board features on-board flash memory. This memory can be programmed and read back using the Out of the Box Flash Programmer GUI over the board's USB interface.

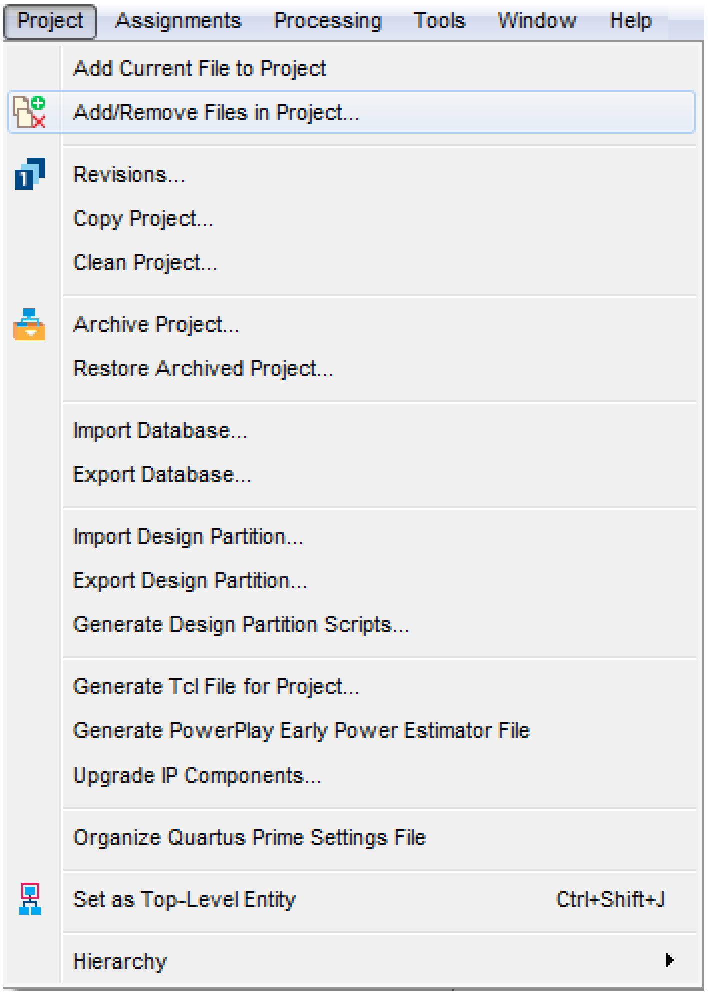
Out of the Box provides a Flash Interface consisting of several VHDL files. These files are used to switch between two modes use of the Flash Programmer GUI, and use of the Flash for a user application. These modes are intended to be selected by the logic level of a CPLD pin.

Using the Flash Interface in a Project

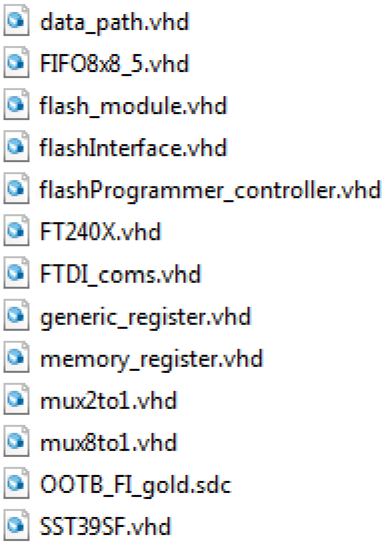
To utilize the Flash Interface in a project use the following procedure.

Step 1: Importing Flash Interface Files

- Import all Flash Interface Files into a project.
    - From the **Project** tab select Add/Remove Files in Project…


Project
Assignments
Processing
Tools
Window
Help
Add Current File to Project
Add/Remove Files in Project...
Revisions...
Copy Project...
Clean Project...
Archive Project...
Restore Archived Project...
Import Database...
Export Database...
Import Design Partition...
Export Design Partition...
Generate Design Partition Scripts...
Generate Tcl File for Project...
Generate PowerPlay Early Power Estimator File
Upgrade IP Components...
Organize Quartus Prime Settings File
Set as Top-Level Entity
Ctrl+Shift+J
Hierarchy

- Select ... to browse for the **flashInterface** folder Therein is a folder **FlashInterface SRC** containing the VHLD and SDC files of the Flash Interface. Select all the files therein and press Open. These files are listed below.



data_path.vhd
FIFO8x8_5.vhd
flash_module.vhd
flashInterface.vhd
flashProgrammer_controller.vhd
FT240X.vhd
FTDI_coms.vhd
generic_register.vhd
memory_register.vhd
mux2to1.vhd
mux8to1.vhd
OOTB_FI_gold.sdc
SST39SF.vhd

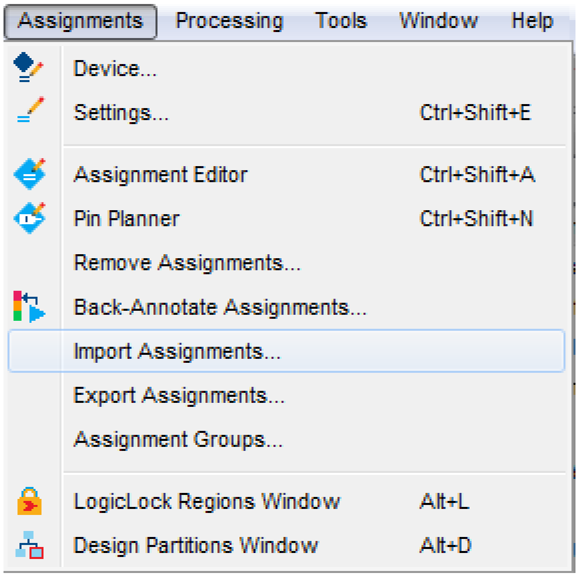
- Once returned to the Settings menu select Apply. The Flash Interface files will now be added to the Quartus project.

Step 2: Import FlashInterface_Block

The file **FlashInterface_Block.bdf** contains all hardware interface pins and signals of the Flash Interface. Copy the contents of the **FlashInterface_Block.bdf** file and paste them into a project bdf file. The user application should be built using the signals **CEn**, **OEn**, **WEn**, **A[14..0]**, and **output[7..0]**.

Step 3: Importing Pin Assignments

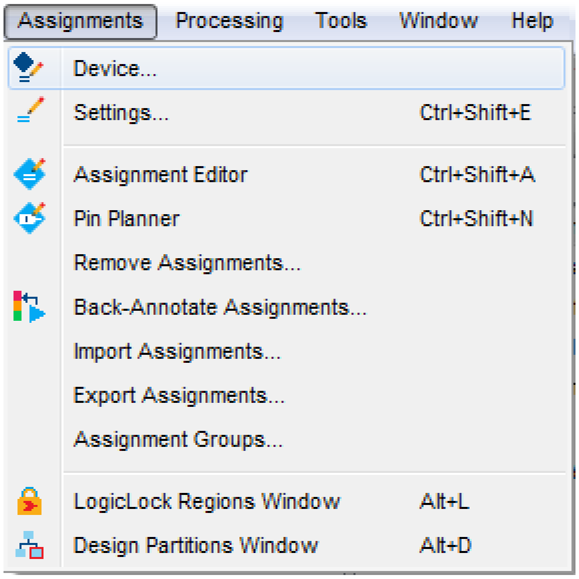
- Once the project is ready for compilation run Analysis & Synthesis by either pressing ✓ on the toolbar, or double click on ▶ Analysis & Synthesis in the Task window.
- Once Analysis & Synthesis succeeds use the Assignments tab to select import Assignments.



- From the Import Assignments window use ... to select FlashInterface.qsf. Once selected press OK. Now all FlashInterface pins will be assigned.

Step 4: Assign Remaining Pins

- Use the Pin Planner, opened by the toolbar button, to assign CPLD pins. Pins are set by using the Location column. The only signals that should need assignment are those not associated with the Flash Interface. Reference the silkscreen of the Out of the Box MAX V Development Board to determine what pins should be assigned to the remaining unallocated signals.
- Before proceeding ensure that unused pins are set to input tri-stated.
  - From the Assignments tab select Device



  - From the device menu select Device and Pin Options…
  - Select the Unused Pins category.
    - Use the Reserve all unused pins drop down menu to select As input tri-stated.
- Once all pins have been assigned compile the design using the ▶ button on the toolbar or double clicking ▶ Compile Design from the Task Window.

Step 5: Program CPLD

Use the Quartus programmer to program the CPLD with the project's pof file.

Using the Flash Programmer GUI

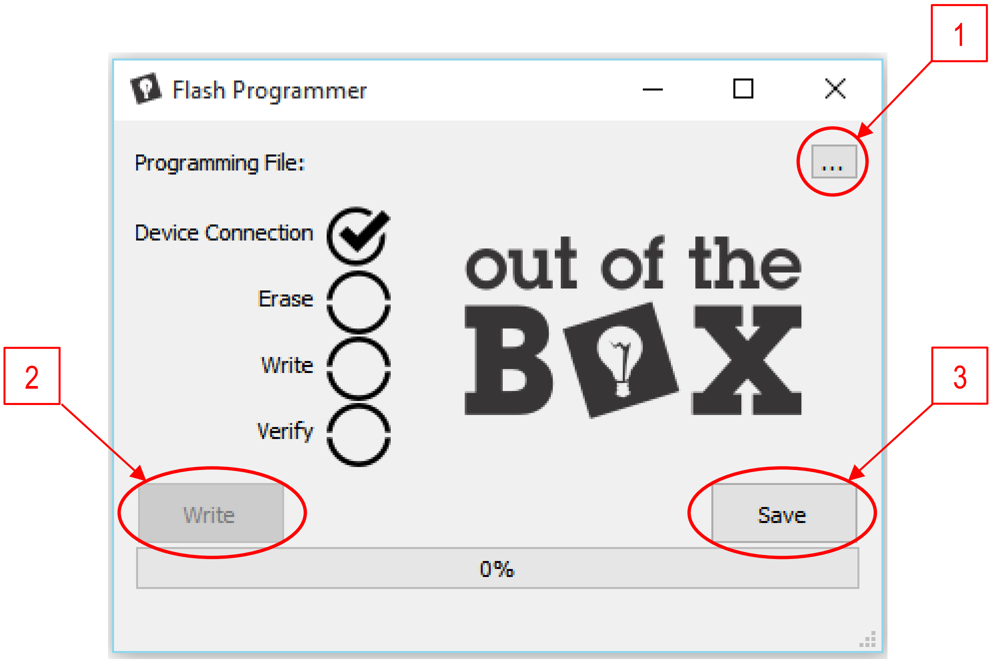
Selecting Programming Mode

Pin 50 of the CPLD is used as the PROG_en signal. This pin is pulled high per the Flash Interface pin assignments. This means that by default the Flash is readable via the user application only. To allow the Out of the Box MAX V Development Board to enter Programming Mode, (allows the device to interface with the Flash Programmer GUI) pin 50 must be pulled LOW. When in programming mode LED ST1 will light up. Note make sure to return the Out of the Box MAX V Development Board to normal operation once use of the Flash Programmer GUI has concluded. The user application will only work when Programming Mode is disabled.

GUI Inputs

There are three buttons used to interface with the flash module located on the Out of the Box MAX V Development Board. These buttons are the programming file browse button (1), the "Write" button (2), and the save button (3). These buttons are circled in the figure below. Their functions are as follows:

1. Program File Browse - This button is used to select a ".mif" file to write to the onboard flash memory. Pressing this button will open the windows file explorer window. Navigate to the directory containing the desired file, highlight it, then select open. Once selected, the absolute path to the file will appear to the right of "Program File:".
2. Write -The write button is used to write the selected program file to the onboard flash memory. When no program file selected, or the selected file is invalid, the write button will appear greyed out as shown in the figure below.
3. Save -This Save button is used to read the data stored on the flash memory. Once a file name and directory is chosen this data will be saved as a .mif.

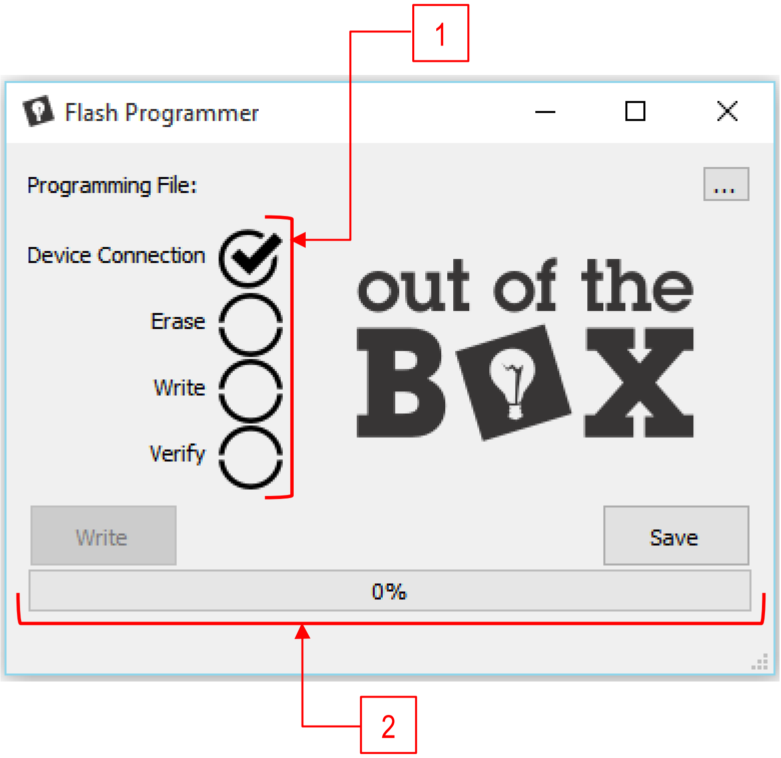


GUI Statuses

There are two separate ways to monitor status. There are the general process status circles (1) and the current action status bar (2). These status units are marked in the figure below. Their functions are as follows:

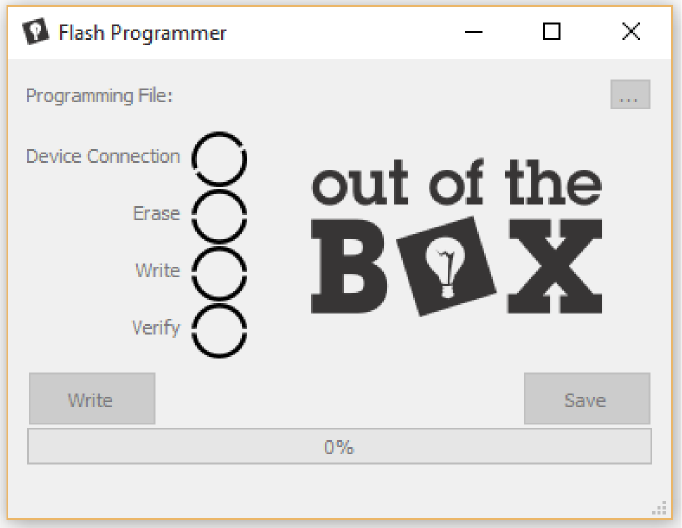
1. **General Process Status Circles** - These circles give a pass-or-fail status of the individual functions of the programmer. When a function is performed successfully, the circle is populated with a check mark as seen in the "Device Connection" circle in the figure below. When a function fails, the circle is populated with an exclamation mark (not shown). These status circles are updated live, allowing the user to see the current step the programmer is on and which step, if any, failed during a program cycle.

2. **Current Action Status Bar** - This bar displays the percentage complete of the current function. This allows the user to both track progress and identify an issue with the programmer. The bar will both update the percent value located in the center of the bar, and progress a green bar as the process runs.

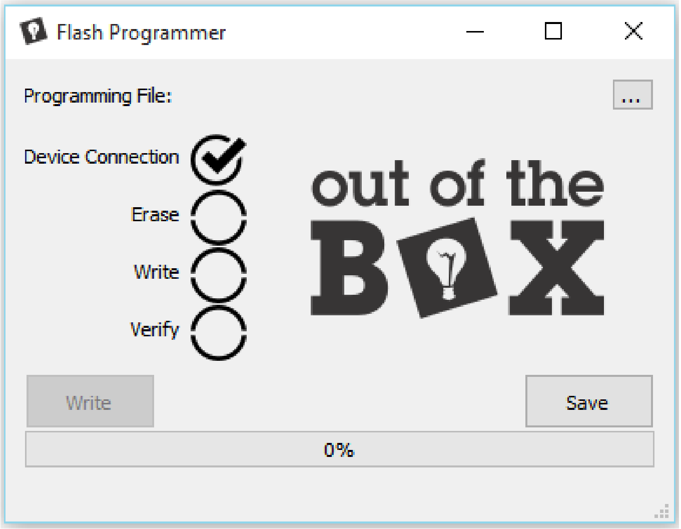


Connecting a Device

For the Flash Programmer application to function, the programmer .pof file must be programmed onto the Out of the Box MAX V Development Board. Once this has been performed the device can communicate to the host PC.

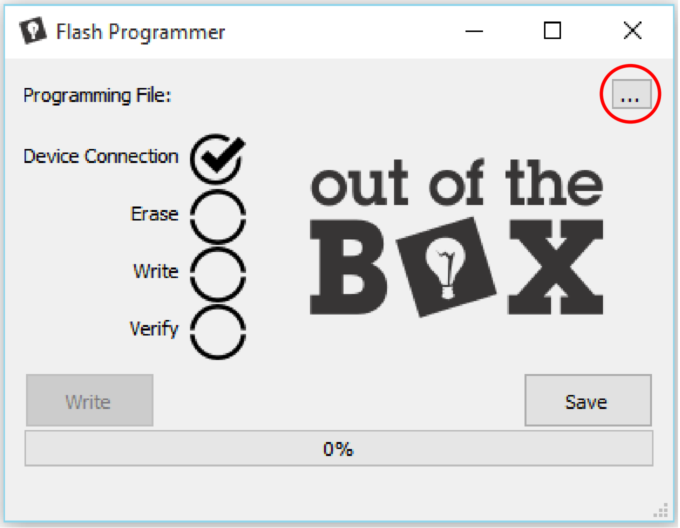


While the Flash Programmer searches for an Out of the Box MAX V Development Board the Device Connection indicator will be an empty spinning circle as shown above. Once a development board is found the indicator will be filled by a check mark as seen below.



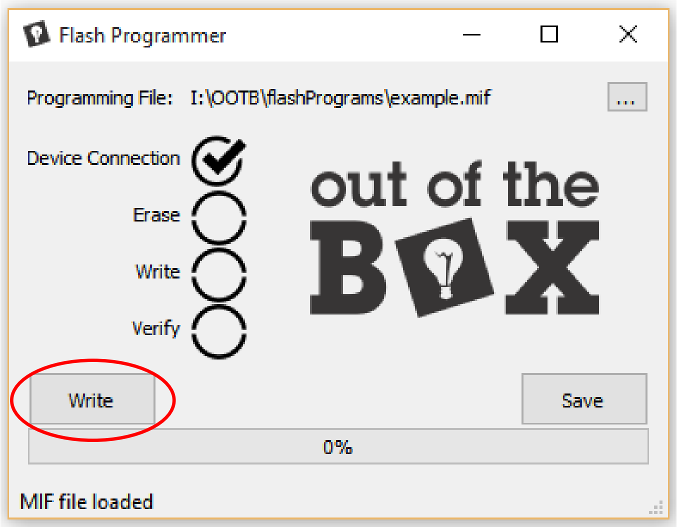
Write to Flash

Once a device has been successfully connected the next step is to select a file to be programmed. Use the circled button below to browse for a file to program. Once this is done, the file path will be displayed next to the button.



In the example below, the '.mif' file to be programmed is located at "I:\OOTB\flashPrograms" and is called "example.mif". The write button, no longer greyed out, indicates that a Write operation can now be performed.

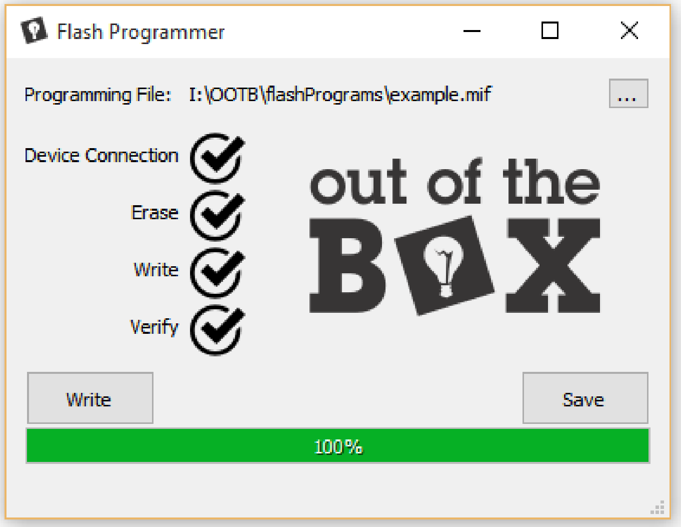
Once the Write button is pressed (circled below) the erase write and verify processes will follow one after another. The current status of the process will be indicated by the current action status bar, and once complete the process' indicator will be filled with a check mark.



Programming File: I:\OOTB\flashPrograms\example.mif

MIF file loaded

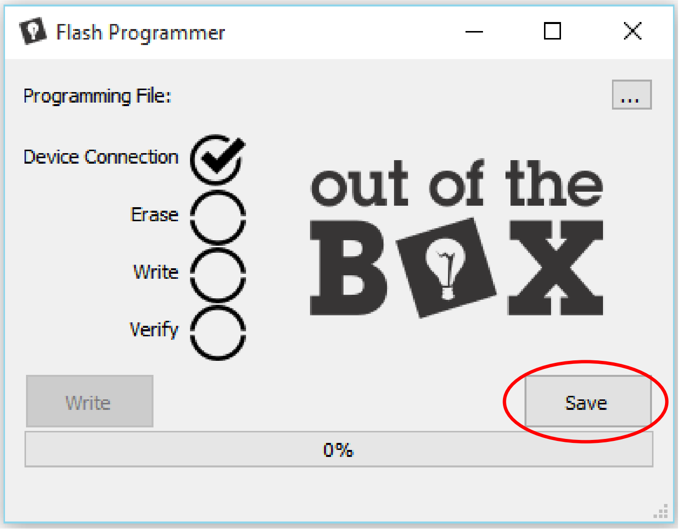
Upon successful completion of all of the processes the window will appear as below.



Programming File: I:\OOTB\flashPrograms\example.mif

Read from Flash

Press the Save button (circled below) To read the contents from the onboard flash memory. This button will open a dialog where a path and file name can be specified. The file will be saved in .mif format.


Flash Programmer
Programming File:
...
Device Connection
Erase
Write
Verify
out of the BOX
Write
Save
0%