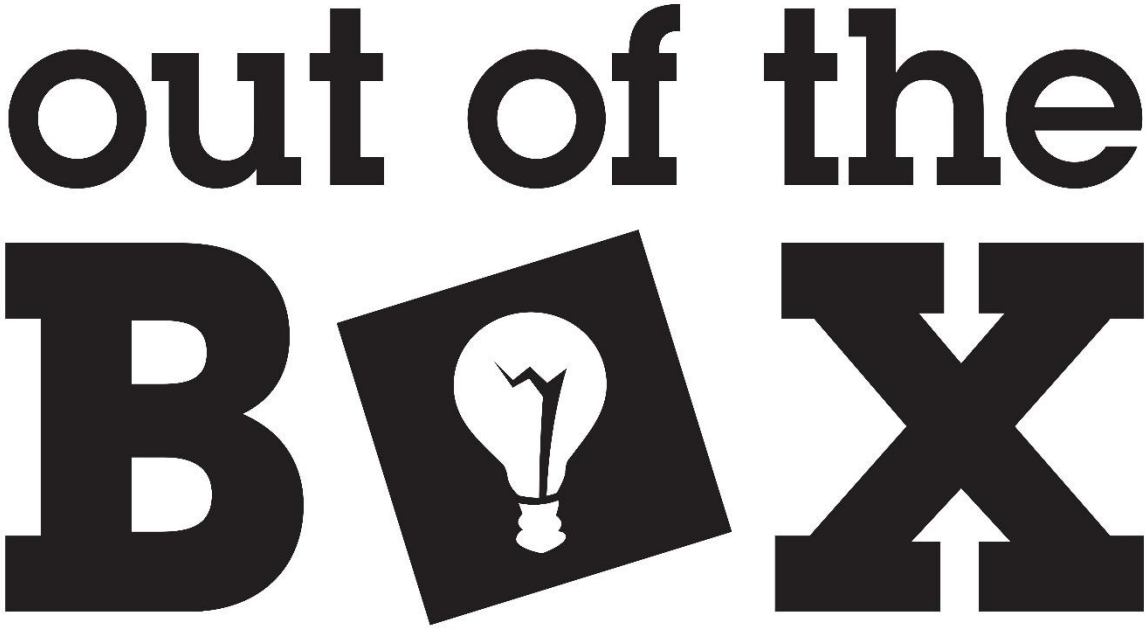
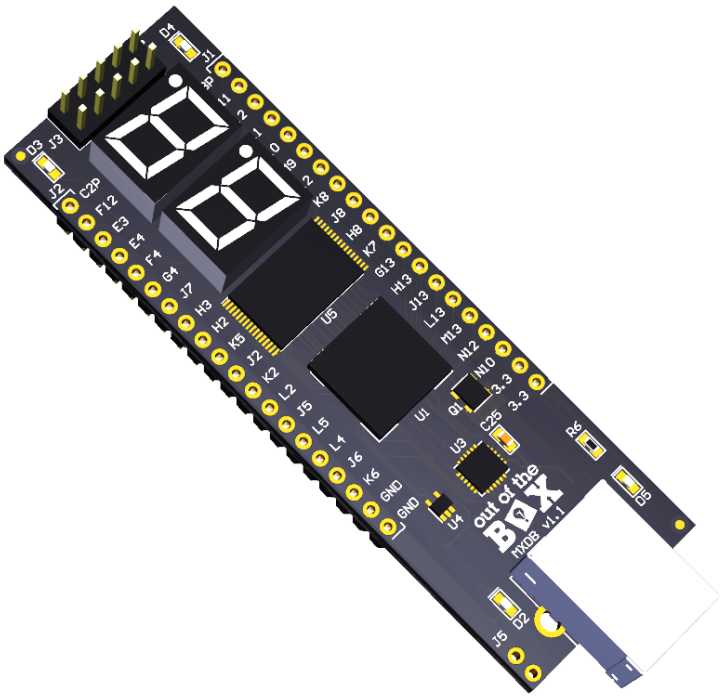


MXDB

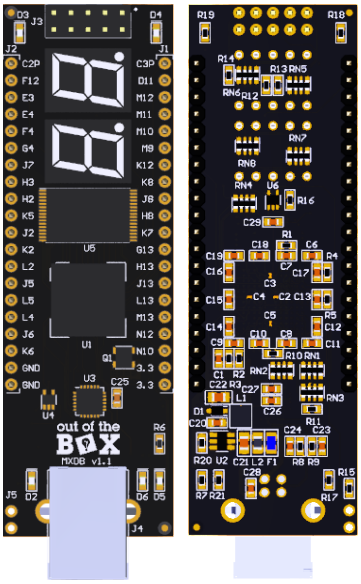
MAX 10 Development Board

Last revision 6/14/2019

Table of Contents


out of the
BOX
MKDB v1.1

Overview



The MXDB is Out of the Box's 2nd generation programmable logic development platform. The MXDB is a platform designed specifically for learning the basics of digital electronics from the lowest level of logic gates to microprocessors. The board features a USB FIFO bridge, external flash memory and accompaniment software to program data to the flash memory with a PC. The board's form-factor also makes it ideal for interfacing with a solderless breadboard.

Warning Read Before Continuing

The MXDB is a FPGA platform that also contains additional ICs capable of driving signals to the FPGA controlled via read strobe pins. There is a jumper **J5** that when left open will help prevent inadvertent IO conflicts, but this should not be considered a complete solution! It is the user's responsibility to review the device schematic and the FPGA pin planner assignments before uploading any design to the device.

Features

MAX 10 CPLD

The MXDB is by default configured with a MAX 10 2K logic element FPGA. Unlike most FPGAs the MAX 10 family of devices has built in non-volatile memory allowing the device to function like a CPLD. This means that the program on the device will not be lost when the device is powered down. The device will simply resume operation upon being powered again.

I/O and Indicators

The MXDB has 34 GPIO pins 2 of which are optionally used a global clock inputs. There are also 4 general purpose LEDs, 2 seven segment LED displays, and a single power LED.

USB

The MXDB is equipped with a FT-240X USB to FIFO bridge that allows the FPGA to send and receive data from a PC USB connection. This device is utilized for the Out of the Box Flash Programmer GUI, that can program and read from the on-board parallel flash IC.

External Flash Memory

The MXDB features parallel flash memory connected to the FPGA. This ROM could be used for simple look up tables, soft processor program memory etc.

USB Flash Programming

As mentioned above the Out of the Box Flash Programmer GUI interfaces with the MXDB's USB connection in order to program to and read from the on-board flash memory.

IO Protection

The MXDB has a jumper **J5** used to allow the use of the read strobe signals to the USB to FIFO device and the parallel flash. With this jumper open the read strobes will be pulled false. This jumper is used to **help prevent** bus conflicts if an inexperienced user accidently assigns a design pin to a strobe erroneously. Doing so could allow the flash or USB devices to drive pins to the FPGA. **Note: to utilize the flash Programmer GUI this jumper must be present.**

Electrical Characteristics

Electrical Characteristics

Item	Min	Nominal	Max	Unit
Un-programmed Current draw	< 20			mA
3.3V current output[1][2]		~700[2]	1500[2]	mA
GPIO VIH			3.6	V
GPIO VIL	-0.3		0.8	V

1) This includes the current required to power the board itself

2) The current output is likely limited by efficiency of the regulator and maximum current of USB 2.0 (500mA). The 3.3V regulator is rated for 1.5A output.

Programming the MXDB

The MXDB more specifically the MAX 10 (10M02SCU169C8G) can be programmed with any Altera (now Intel) compatible JTAG programmer. This tutorial will use the Out of the Box CPLD Programmer, but other devices such as the Terasic USB Blaster Cable can be used as well. The only difference will be the programmer device names seen in the Quartus programmer window.

Requirements

- Altera (Intel) compatible USB blaster cable and respective cable to connect to the computer running Quartus.
    - This tutorial will use the Out of the Box CPLD Programmer
- MXDB
    - Including USB cable for connecting the board to the computer.
- Quartus 15 or a later version.

Procedure

Use the following steps to program the MXDB

Step 1: Install the drivers for the Programmer to be used

For the Out of the Box CPLD Programmer reference the “Out of the Box CPLD Programmer” document to setup the drivers for this device.

Step 2: Launch Quartus

Open Quartus (version 15.0) or later.

Step 3: Power the MXDB.

Use a standard USB cable (USB A to USB B) to connect the MXDB to the computer’s USB port. **The MXDB cannot be programmed without being powered over USB.**

Step 4: Connect the Programmer

Connect the programmer to the computer using the applicable USB cable. Also connect the 10-pin JTAG cable between the programmer and the MXDB. The JTAG cable should be pointing away from the MXDB as shown in the following image.

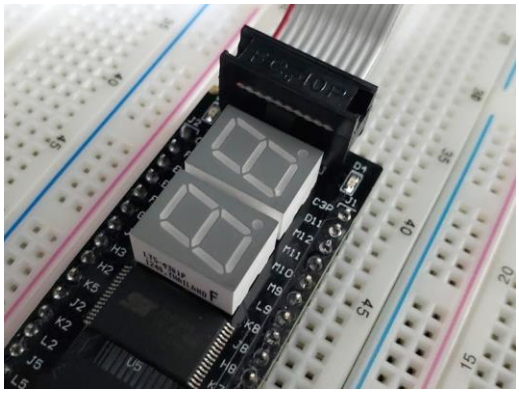
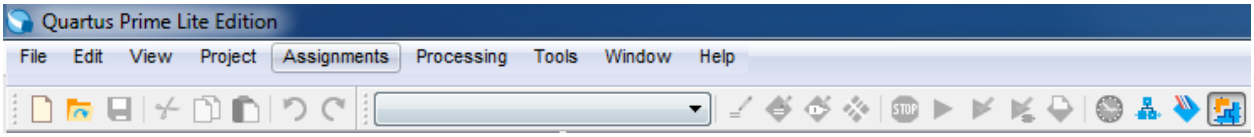


Figure 1: JTAG Cable Orientation

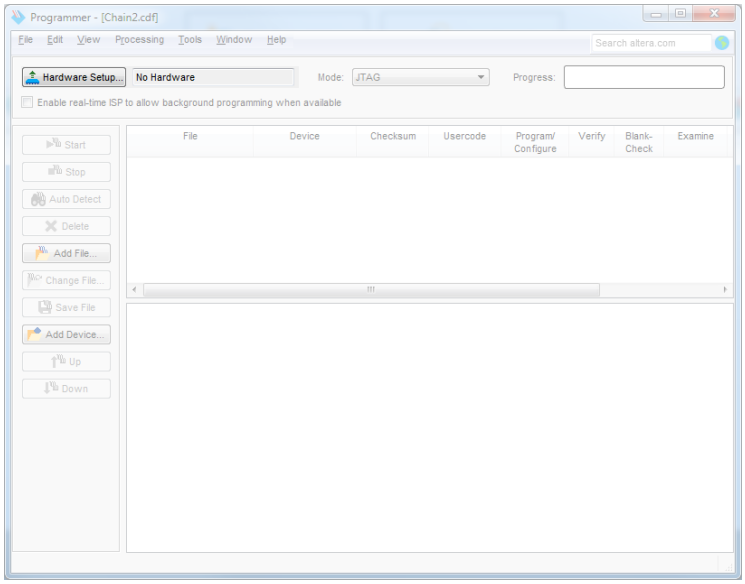
Step 5: Open the Programmer Window in Quartus

The Programmer window is accessed via the button in Quartus' top pane.



Step 6: Hardware Setup

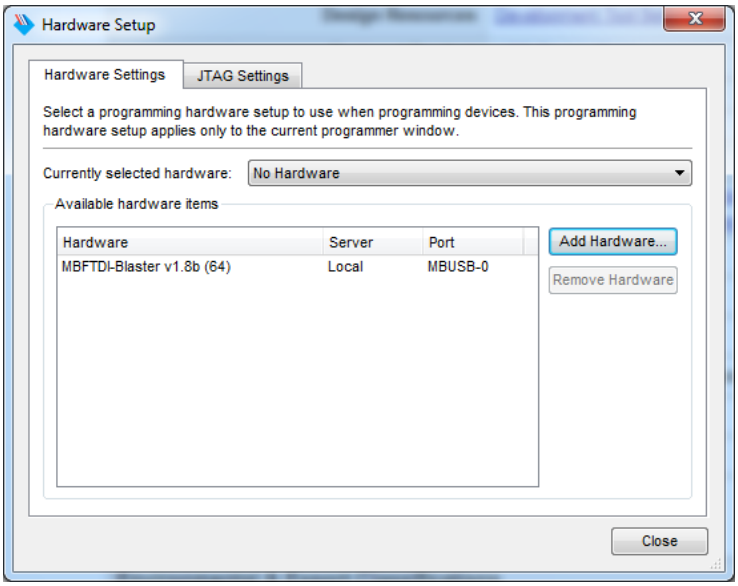
When the Programmer Window is opened check to see if the hardware has been recognized. Reference the highlighted portion of the following image to find out what hardware is connected.



If the Programmer Window **does not** display “No Hardware” and instead lists “MBFTDI-Blaster v1.8b (64)” (be advised a portion of this text could be cut off due to the size of the dialog) proceed to **Step 7**. If the window does display “No Hardware” proceed to **Step 6.1**.

Step 6.1 Manual Connection to Programmer

Press Hardware Setup... to tell Quartus what tool to select. Doing so will open the following window shown below.



If the device “MBFTDI-Blaster v1.8b (64)” is displayed as below proceed to **Step 6.2**. Otherwise, continue.

If the Out of the Box Programmer is connected to the computer and there is no device listed under hardware there are two possible explanations.

1. The device drivers were not installed. To do this go back to **Step 1**.
2. The device was not yet recognized by the computer before this window was opened. The Windows Operating system, by default, plays a sound when devices are both connected and disconnected from the system. Try closing this window; plug in the programmer then wait for the system to audibly confirm the device connection. The device will likely show up and be selected by default. If not try pressing Hardware Setup... again to manually select the device. If the device is still not listed the problem again is likely related to the driver installation in **Step 1**.

Step 6.2 Manual Connection to Programmer Cont.

Double click “MBFTDI-Blaster v1.8b (64)” to select the programmer. (This is the device name in Quartus for the Out of the Box CPLD Programmer). Then exit the Hardware Setup Window. After a small delay the Programmer Window should display the selected device name next to Hardware Setup....

Step 7: Add File

In this step select a file to be programmed to the MXDB. To do this use the 

button in the Quartus Programmer Window.

> **Note:**
>
> *Both SOF and POF files can be used. SOF files are volatile meaning they are erased once the MXDB is powered down. POF files are non-volatile. These contents will remain on the device until erased.*

Upon adding a file ensure that the device is "10M02SCU169" this is the device installed upon the MXDB. For SOF files the only option when programming is Program/Configure while with POF files there are options to Verify, Blank-Check, Erase and others that will not be discussed here. For detail on these options reference the following.

Programmer Options functions

These functions are useful when programming a device to ensure the contents are as expected.

Verify - Ensures that the data in the device is the same as the programming file.

Blank - Check- Ensures that a device was successfully erased

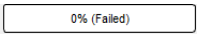
Erase - Erases the contents of the device.

Step 8: Programming the MXDB with a SOF/POF File

Once an appropriate file has been selected for the MXDB press the 

button in the programmer window to begin programming. Be advised that POF files will take a fair amount of time to program especially with added options to the process. SOF files on the other hand will program much faster.

The progress of the device programming will be displayed by a bar graph. Upon successful completion the display should read 

. If there is a result of 

Try the following

- Ensure that the JTAG connection of the programmer is indeed connected to the MXDB.
- Try selecting the option of just "Erase" first then program. If this works, try programming again.
    - If not, disconnect the MXDB from power and from the JTAG cable for 30 seconds. Then plug everything back in and attempt an erase again.

Appendix I: Pin Allocations

General Overview of BGA Pin Designators

The MXDB MAX 10 FPGA comes in a BGA (Ball Grid Array) footprint. BGA pins have an alpha numeric designation since BGA chips organize their pins in both rows and columns as shown in the footprint diagram below.

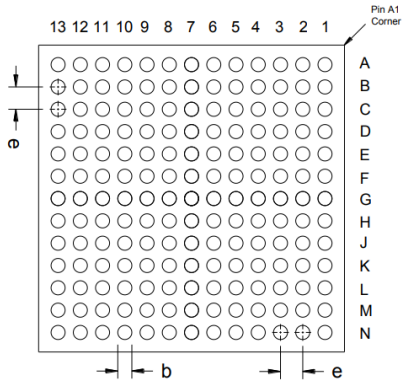


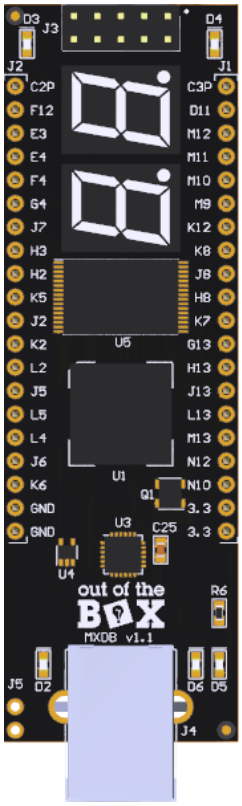
Figure 2: MAX 10 UBGA 169 footprint proveded by Intel (Bottom View)

When programming the MAX 10 FPGA with Quartus, the pin assignments will be alpha numeric; row letter followed by column number.

Breadboard Connections

The GPIO pins of the MXDB are broken out to header pins for use with a breadboard. The Pins with exception of **C2P** (**Pin G9**) and **C3P** (**Pin F13**) have silkscreen labels of the respective BGA pins. **C2P** and **C3P** are special IO pins that also can serve as global clock inputs.

Pin	Type
C2P(G9)	IO/CLK
F12	IO
E3	IO
E4	IO
F4	IO
G4	IO
J7	IO
H3	IO
H2	IO
K5	IO
J2	IO
K2	IO
L2	IO
J5	IO
L5	IO
L4	IO
J6	IO
K6	IO
GND	PWR
GND	PWR



Pin	Type
C3P(F13)	IO/CLK
D11	IO
M12	IO
M11	IO
M10	IO
M9	IO
K12	IO
K8	IO
J8	IO
H8	IO
K7	IO
G13	IO
H13	IO
J13	IO
L13	IO
M13	IO
N12	IO
N10	IO
3.3V	PWR
3.3V	PWR

Figure 3:Breadboard Connections

General Purpose LEDs

The MXDB has 4 General purpose LEDs.

LED Reference Designator	FPGA Pin	Comment
D3	L12	Intended as a general status LED
D4	E10	Intended as a general status LED
D5[1]	N11	Intended as a general status LED. Also intended to indicate USB bytes received.
D6[1]	K13	Intended as a general status LED. Also intended to indicate USB bytes transmitted.

Figure 4: GPIO LEDs and respective pins

1)These LEDs are intended to be used for USB data transmission indicators during flash programming

Clocks

The MXDB's FPGA has many dedicated clock pins. 4 of the available clock pins are available, or which 2 do not already have an on-board clock attached to them.

Clock	FPGA Pin	Source	Schematic label
CLK0p	H6	FTDI USB IC (6MHz default)	FT_CLK
CLK1p	H4	On-board oscillator (4MHz)	EXTCLK
CLK2p	G9	N/A Breadboard connection	CLK2p
CLK3p	F13	N/A Breadboard connection	CLK3p

Figure 5: MXDB available clock sources

Although not available on the MXDB, the MAX 10 FPGA supports differential clock inputs which can be used to trigger events on rising and falling clock edges. This differential capability is the reason behind the CLKxy naming convention. X represents the clock pin number. Y in the example denotes the differential signal polarity (p for positive and n for negative).

Seven Segment Displays

The MXDB has two seven segment LED displays highlighted below. The left display is U8 and right is U7.

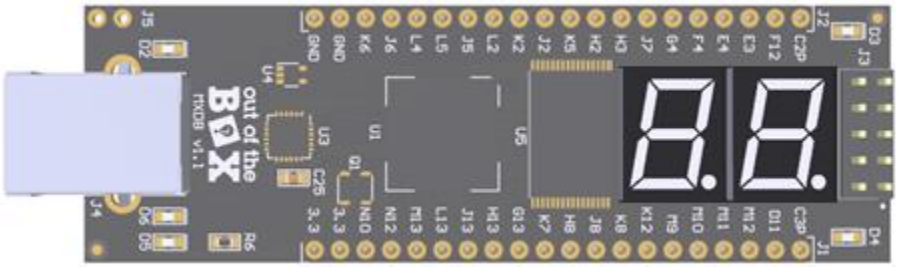


Figure 6:Seven segment displays

The display segment names are shown in the next figure

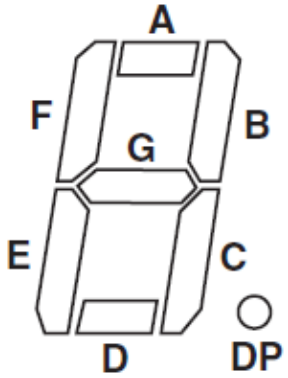


Figure 7:Display LED Labels

Segment Signal (U8 Left)	FPGA Pin		Segment Signal (U7 Right)	FPGA Pin
A	H10		A	D9
B	F8		B	C9
C	E8		C	G12
D	J12		D	L10
E	H9		E	K10
F	J10		F	J9
G	L11		G	E6
DP	E9		DP	E12

Figure 8:Display LED assignments

USB Pin Assignments

The FT-240x device is a parallel interface USB device. The pin assignments for the Device are as follows.

FPGA PIN	Function	Description
N7	USB DATA0	Data Bus pin
N5	USB DATA1	Data Bus pin
N6	USB DATA2	Data Bus pin
M2	USB DATA3	Data Bus pin
M7	USB DATA4	Data Bus pin
M4	USB DATA5	Data Bus pin
N4	USB DATA6	Data Bus pin
M5	USB DATA7	Data Bus pin
M8	USB $\overline{\text{RXF}}$	Low true indication of RX data available
N9	USB $\overline{\text{TXE}}$	Low true indication of TX function ready to receive data
J1	USB $\overline{\text{RD}}$	Low true USB RX buffer read strobe
K1	USB $\overline{\text{WR}}$	Low true USB TX buffer write strobe
M1	SI/$\overline{\text{WU}}$	Low true Send Immediate strobe. Also used to wake up device from sleep
N8	$\overline{\text{PWREN}}$(FT-240 CBUS6 pin)	Output is low after device is ready
H6	FT_CLK (FT-240 CBUS5 pin configured as CLK6MHz)	6MHz oscillator output can be used as a global clock to the FPGA

Figure 9:USB pin assignments

Flash Memory Pin Assignments

The MXDB pin assignments for the on-board parallel flash memory are as follows

FPGA PIN	Function	Description
C13	A0	Flash address pin
D12	A1	Flash address pin
C11	A2	Flash address pin
C12	A3	Flash address pin
F1	A4	Flash address pin
E1	A5	Flash address pin
D1	A6	Flash address pin
C1	A7	Flash address pin
B5	A8	Flash address pin
A5	A9	Flash address pin
A7	A10	Flash address pin
B6	A11	Flash address pin
B1	A12	Flash address pin
A4	A13	Flash address pin
B4	A14	Flash address pin
C2	A15	Flash address pin
B2	A16	Flash address pin
A3	A17	Flash address pin
A2	A18	Flash address pin
B13	FL IO 0	Flash data pin
B12	FL IO 1	Flash data pin
A12	FL IO 2	Flash data pin
B11	FL IO 3	Flash data pin
A11	FL IO 4	Flash data pin
B10	FL IO 5	Flash data pin
A10	FL IO 6	Flash data pin
A9	FL IO 7	Flash data pin
B3	FL $\overline{\text{WE}}$	Flash Write enable
A6	FL $\overline{\text{OE}}$	Flash Output enable
A8	FL $\overline{\text{CE}}$	Flash Chip enable

Figure 10: Flash memory pin assignments

Appendix II: Flash Programmer GUI

The Flash Programmer GUI (Graphical User Interface) is a program used to read data from and write data to the on-board flash of the MXDB.

Perquisites

- Before the GUI can be use with the MXDB, the FlashProgrammer POF or SOF files need to be loaded to the MXDB. These files enable communication with the board via USB to the on-board flash. Using the SOF version of the programmer is quick, but upon powering down of the board this program will be erased. If using the POF version of the FlashProgrammer subsequent SOF files will take precedent in the device, but the program will be useable once again upon a power cycle of the device.
    - It is advised to use the POF version of the FlashProgrammer file since a power cycle (un-powering then powering again) will revert the MXDB to being ready to use with the GUI.
- The Jumper location **J5** of the board **must have a jumper present**. This jumper allows for the reading of data from the USB and Flash devices. The jumper exists to break this connection for new users whom may accidently assign the data pins of the Flash/USB devices which could result in damage to the board.

Using the Flash Programmer GUI Interface

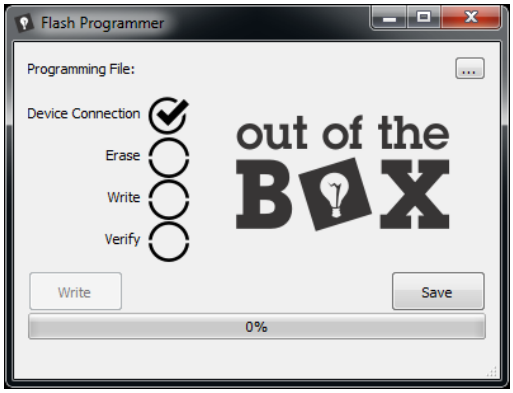


Figure 11: Flash Programmer GUI Window

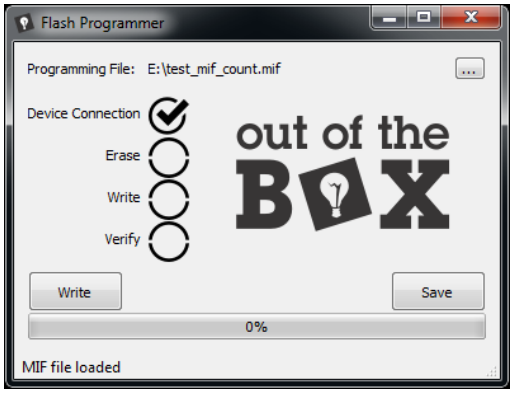
Device Connection

When a MXDB board is connected to the Flash Programmer GUI the device connection will display a check mark

When no device is present the device connection will display a spinning wheel.

Writing to Flash

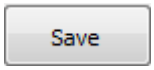
To write to the MXDB on-board flash memory select the browse button ... . From the browser window that pops up, locate a MIF file to load to the flash. Once a file has been selected the **Write** button will become clickable.



Once **Write** is clicked the GUI will begin the Erase, Write, and Verify processes. Each process if successful will be denoted by a  .

If instead a  is displayed, then the process has failed. Often the reason for this failure is that the jumper J5 is not present. Another possibility is that the FPGA does not contain the FlashProgrammer SOF/POF file required for communication with the GUI.

Reading from Flash

To read from the MXDB on-board flash memory select the 

button. In the subsequent browser window create a file or select a file to load the data into. After making this selection the progress bar will become active. If this does not happen then the process has failed. Often the reason for this failure is that the jumper J5 is not present. Another possibility is that the FPGA does not contain the FlashProgrammer SOF/POF file required for communication with the GUI.

Appendix III: Troubleshooting

- **Programmer not showing up in Quartus**
  - If the Flash Programmer GUI is open this program may be interfering with the Quartus Programmer. When programming the FPGA close the Flash Programmer GUI. If the GUI is closed refer to this section for further debugging.
- **The Flash Programmer GUI is failing to write/read**
  - Make sure that 1) jumper J5 is present and 2) that the FlashProgrammer sof/pof file is loaded to the FPGA.
- **Can't find C2P or C3P in the pin planner**
  - These labels refer to the clock input not the FPGA pin. C2P is pin G9 and C3P is pin F13.
- The FPGA failed to program
  - Refer to this section.