

# XDS110 Support Notes



## Overview

The XDS110 support directory contains the following:

- `xdsdfu` -- Console application to program the XDS110's firmware.
- `xds110reset` -- Console application to reset the board using the XDS110.
- `firmware_x.x.x.x.bin` -- Current XDS110 firmware where x.x.x.x is the version.
- `boot_loader.bin` -- Current boot loader for the XDS110.

This document includes instructions for how to use `xdsdfu` and `xds110reset`. At the end of the document are instructions for alternate methods of flashing and recovering a bricked XDS110.

The Windows installer also installs the OS drivers necessary to connect to the XDS110.

The Linux installation includes a `udev` directory with the rules file needed to use the XDS110 as a user account. An installer script for the rules file is also included.

Mac and Linux systems require `libusb` to be installed. Go to [libusb.info](http://libusb.info) for more details.

## `xdsdfu` Flash Utility

The `xdsdfu` console application allows you to update the firmware and set the serial number of the XDS110 debug probe. Note that Code Composer Studio will automatically update the XDS110 to the current firmware when starting a debug session. `xdsdfu` is only required for firmware updates when Code Composer Studio is not being used.

## `xdsdfu` Options

`xdsdfu` accepts the following command line options. All commands except the `-e` command will operate on the first XDS110 found. While using `xdsdfu`, be sure to have only one XDS110 plugged into the computer.

`-e`

Show information about all XDS110s connected to the computer.

`-m`

Switch into DFU mode if XDS110 is currently in runtime mode. The XDS110 reconfigures itself as a Tiva Firmware Update device and waits for instructions to program the flash.

`-f <FILE>`

Download the given firmware file to the XDS110.

-n <TEXT>

Set XDS110 serial number to any four character string (no spaces). This option preserves the board prefix (if any) which is the first four characters of the serial number.

-s <TEXT>

Set XDS110 serial number to any eight character string (no spaces). This option replaces the entire serial number which overwrites the board prefix value.

-c <NUM>

Set the configuration of the device. Use -c ? to show a list of available device configurations. (Configurations were implemented starting with firmware version 2.3.0.15.)

-r

Must be used in combination with another option. Reset the XDS110 on completion of the other operation.

-? or -h

Display the help text.

## Device Configurations

Device configurations are set using the `-c` option in `xdsdfu`.

Note that in firmware versions before 2.3.0.15, setting a configuration will have no effect; only the standard configuration will be used.

Also note that if you select a configuration that is not supported by the hardware, the standard configuration will be used instead.

The following configurations are currently implemented in the current firmware:

1 -- Standard configuration: Features include XDS Debug, App COM Port, Aux COM Port, CMSIS-DAP 1.0, and EnergyTrace support.

2 -- Drag-n-Drop configuration: Features include XDS Debug, App COM Port, Aux COM Port, CMSIS-DAP 1.0, and Drag-n-Drop Flash.

The following configuration was first implemented in firmware version 3.0.0.8:

4 -- CMSIS-DAP 2.0 configuration: Features include XDS Debug, App COM Port, Aux COM Port, CMSIS-DAP 2.0, EnergyTrace support, and Drag-n-Drop Flash (Drag-n-Drop Flash is only enabled on embedded XDS110 probes.)

## Updating the Firmware Using `xdsdfu`

To program the firmware, follow these steps:

1. Plug the XDS110 debug probe into your computer. Make sure that you only have one XDS110 class debug probe plugged in. The `xdsdfu` program will attempt to flash the first XDS110 debug probe it finds.

2. Run the following two commands from directory with xdsdfu:

```
xdsdfu -m  
xdsdfu -f firmware.bin -r
```

Use the actual filename of the firmware file. Starting with firmware version 3.0.0.8, the version number is part of the filename (e.g. firmware\_3.0.0.8.bin). You may need to pause after the first command to give the OS time to recognize that the XDS110 has reconfigured as a different USB device.

Once the second command has completed, the firmware is updated, and the XDS110 should be ready to use.

### **Setting the Serial Number Using xdsdfu**

You can set the XDS110 serial number using the xdsdfu program. This will allow you to have more than one XDS110 connected to your system by letting the host locate the specific one to use.

#### *Viewing the Current Serial Number*

1. Plug the XDS110 debug probe into your computer.
2. Run the following command from directory with xdsdfu:

```
xdsdfu -e
```

xdsdfu will display the status, firmware revision, and serial number of all XDS110s connected to the computer.

#### *Setting the Serial Number with Board Prefix*

Some LaunchPads and EVM boards with XDS110 will use a four character prefix to identify the board as part of the serial number. To preserve the board ID while setting a serial number follow these steps:

1. Plug the XDS110 debug probe into your computer. Make sure that you only have one XDS110 class debug probe plugged in. The xdsdfu program will attempt to flash the first XDS110 debug probe it finds.
2. Run the following two commands from directory with xdsdfu:

```
xdsdfu -m  
xdsdfu -n xxxx -r
```

where xxxx is a serial number, any combination of letters and numbers, from 1 to 4 characters in length.

You may need to pause after the first command to give the OS time to recognize that the XDS110 has reconfigured as a different USB device.

Once the second command completes, the serial number is set, and the debug probe should be ready to use.

### *Setting the Complete Serial Number String*

If you don't need to preserve the board prefix ID, then you may do the following to set the entire serial number string. But note that certain tools require a board prefix to properly identify the target board. If you overwrite the board prefix, these tools will fail to operate correctly.

1. Plug the XDS110 debug probe into your computer. Make sure that you only have one XDS110 class debug probe plugged in. The xdsdfu program will attempt to flash the first XDS110 debug probe it finds.
2. Run the following two commands from directory with xdsdfu:

```
xdsdfu -m  
xdsdfu -s xxxxxxxx -r
```

where xxxxxxxx is a serial number, any combination of letters and numbers, from 1 to 8 characters in length.

You may need to pause after the first command to give the OS time to recognize that the XDS110 has reconfigured as a different USB device.

Once the second command completes, the serial number is set, and the debug probe should be ready to use.

### **Drag-n-Drop Flash**

The XDS110 Drag-n-Drop feature allows the user to flash a program to the target device without needing Uniflash, Code Composer Studio, or another debug environment installed on their computer. The XDS110 appears as a virtual drive in their computer's file system. This allows the user to copy a file to the XDS110 to flash that file to the target. After flashing is complete, the target device is reset to allow the program to start running.

### **Configuration**

To use the Drag-n-Drop feature, the XDS110 must first be configured to enable it.

Begin by placing the XDS110 into programming mode:

```
xdsdfu -m
```

Then select configuration #2 (Drag-n-Drop) or #4 (CMSIS-DAP 2.0), and tell the XDS110 to reset to enable the feature:

```
xdsdfu -c 2 -r
```

The Drag-n-Drop feature replaces Energy Trace support on the XDS110 in configuration #2. To disable Drag-n-Drop and restore Energy Trace support, select configuration #1 (Standard):

```
xdsdfu -m
```

```
xdsdfu -c 1 -r
```

## Operation

When plugged into your computer, the XDS110 appears as a drive named "XDS110".

This drive contains two files: a read me file with brief instructions on using Drag-n-Drop and a status file reporting the results of the last operation.

Copy a file to the drive to flash that file to the target. Simply click on the file, drag it to the drive, and drop it. Or you may use a copy command from the command line.

After the file has been flashed, the drive will reset itself, and the status file will report the result.

On success, the file contains the following message:

The file was flashed to the target device. <n> bytes were written to flash.

If an error occurred during the flashing operation, the status file will change to report the error. On fail, the file contains information about why it failed (in this case, TCK was disconnected from the target):

Failed to flash program into target device.  
Error occurred while establishing debug connection to target.  
Error code: -615.

## Supported Target Devices

The XDS110 Drag-n-Drop feature supports the following target devices:

- MSP432P4
- MSP432E4
- CC13x0
- CC26x0
- CC13x2
- CC26x2
- CC3220SF (internal flash only)

## Supported File Formats

The XDS110 Drag-n-Drop feature supports the following file formats:

- Extended Tektronix Hex format
- Intel Hex format
- Raw binary format

To build hex file formats in CCS projects, open properties for the project, and enable the ARM Hex Utility. Set both the memory width and ROM width to be 8 bits. Then choose the format (Extended Tektronix is the default). The project will then generate a .hex file in addition to the .out file when built.

## Requirements

The XDS110 in Drag-n-Drop configuration should work on Windows, Linux, and Mac OSX with minimal software installation. Drag-n-Drop will only work on embedded XDS110s.

If already configured for Drag-n-Drop, Windows will automatically recognize the XDS110 virtual drive without the XDS110 driver installed. Windows will report that it failed to install a driver when you first plug it in, but the driver is not required for the virtual drive used by the Drag-n-Drop feature. A Windows driver and the emupack must be installed to update the firmware and to configure the XDS110.

Linux should allow access to the XDS110 virtual drive without a driver or udev rules file (tested on Ubuntu 16.04LTS). The emupack must be installed to update the firmware and configure the XDS110.

Mac OSX will allow access to the XDS110 virtual drive without needing any driver or other files installed. The emupack must be installed to update the firmware and configure the XDS110.

### **xds110reset Reset Utility**

The xds110reset console application allows you to reset the LaunchPad or target board using the nSRST pin of the JTAG connection. This is equivalent to pressing a reset button on the board, but done under the command of the XDS110 debug probe.

To reset the board, run the xds110reset command. The XDS110 will pull the nSRST signal low momentarily triggering a board reset.

### **xds110reset Options**

xds110reset accepts the following command line options:

-a, --action <NAME>

Choose a specific action to perform. NAME may be "assert", "deassert", or "toggle". If not specified, toggle will be executed by default.

-d, --delay <VALUE>

Set the on-time for the reset toggle in milliseconds. This has no effect if action is "assert" or "deassert". If not specified, the delay will be set to 50 ms.

-s, --serial <TEXT>

Select the XDS110 probe by serial number. TEXT is the serial number to use, up to eight characters. If not specified, the first XDS110 found will be used.

-h, --help

Display the help text.

### **xds110reset Example**

The following example toggles the reset pin, asserting the pin for 100 ms, using XDS110 with serial number "01234567":

```
xds110reset -a toggle -d 100 -s 01234567
```

## Updating Firmware Using a JTAG Connection

In the case that the XDS110 fails to enumerate as a USB device, you can attempt restore functionality by updating the firmware using a JTAG connection. Contact your vendor for details on how to attach a JTAG cable and for what programmer tool you should use. Perform the following steps to restore the firmware:

1. Connect the flash tool to the XDS110's CPU using JTAG connection.
2. Use the flash tool to erase the entire flash of the XDS110.
3. Flash the boot\_loader.bin file to address 0x0000.
4. Power cycle the XDS110 (unplug and re-plug it into the USB port).
5. Use the xdsdfu utility to update the firmware as detailed above.

The firmware is now updated, and the XDS110 debug probe should be ready to use.

## Recovering a Bricked XDS110 without JTAG

In the case that your XDS110 fails to enumerate as a USB device, or it fails to enter DFU programming mode, you can attempt the following steps to force it into DFU mode to recover:

1. Ground the JTAG TDO pin of the XDS110's Tiva CPU. This is pin 97 on the 128 pin device. (Check the TM4C1294NCPDT datasheet for the pin location; it is located on a corner and easy to access.)
2. Unplug and re-plug the XDS110 into the host computer while the pin is grounded.
3. The XDS110 should now be in DFU programming mode.
4. Remove the connection to ground while the XDS110 remains powered.
5. Use the xdsdfu utility to update the firmware as detailed above.