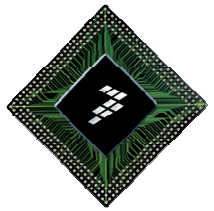
September, 2010

MFG Tool Overview



Freescale / MX28 DFAE Training

freescale™ semiconductor

Overview

- What is the MfgTool?
  - Prepare multiple devices in parallel.
  - Where can I get it?
- Basic Operation
  - Physical setup
  - Configure USB Ports
  - Selecting profile
- UTP_UPDATE vs. MX_UPDATE
- Supported Devices - MX23, MX28, MX25, MX35, MX51….
- Supported OSes
  - Host – Windows XP, Windows 7
  - Firmware - Linux, WinCE

Basic Functions – Host Component

• The Mfg Tool host component is an operator friendly GUI interface for the firmware imaging process.

• The GUI associates a physical USB port to the firmware imaging operations and provides feedback to the operator.

• The Mfg Tool Framework is an architecture that supports:

  • Communication with various USB device drivers,

  • Loading firmware to ROM device enabling extended ROM functionality or complete application functionality.

  • Invoking commands supported by currently executing firmware.

Basic Functions – Firmware Component

- The Mfg Tool firmware component enables these basic functions:
  - Erasing the media.
  - Allocating the media.
  - Writing firmware to the media allocation(s).
- Additional functionality is important for consumer devices:
  - Initializing the file system on the media.
  - Preloading content in data area of media
  - Managing Fuses

Universal Command Engine (UCE)

Recovery

```
<UCL>
  <CFG>
    <STATE name="Recovery" dev="MX28-ROM"/>
    <DEV name="MX28-ROM" vid="15A2" pid="0047"/>
```

```
<LIST name="Singlechip NAND" desc="Install on singlechip NAND"
  <CMD type="boot" body="Recovery" file="updater.sb">Booting update firmware.</CMD>
  <!-- "boot" Finds configured device, forces it to "body" device and downloads "file". -->
```

```
<CMD type="find" body="Updater" timeout="60"/>
<!-- "find" - Waits for the "body" device to connect. -->
```

Updater

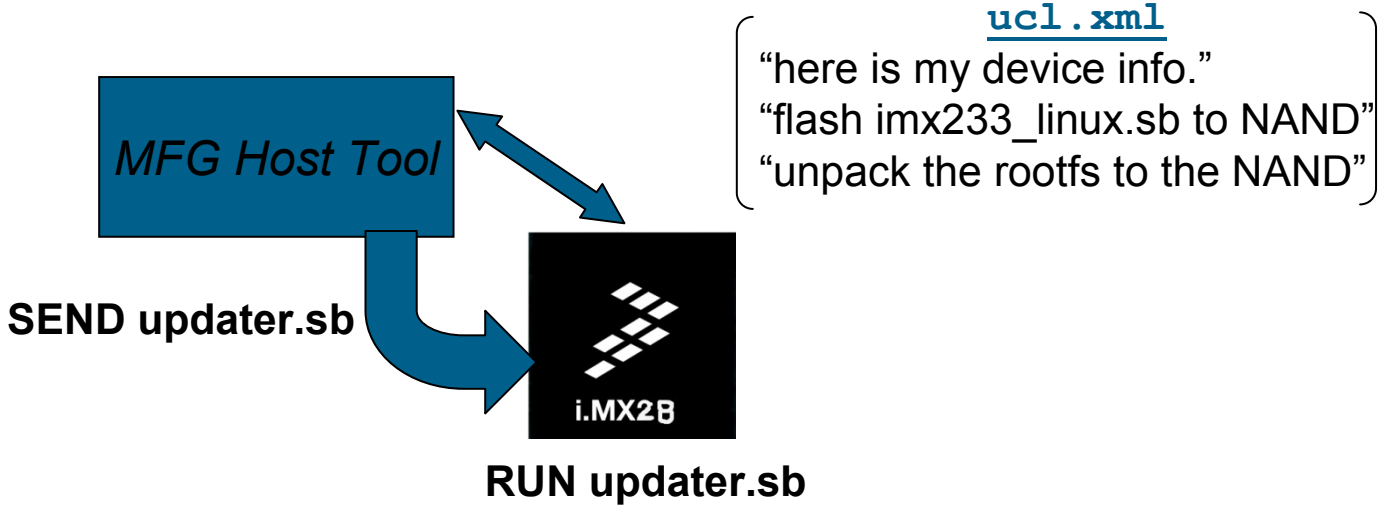
```
<UCL>
  <CFG>
    <STATE name="Updater" dev="MX28-UPD"/>
    <DEV name="MX28-UPD" vid="066F" pid="37FF"/>
```

```
  <CMD type="pull" body="?" file="device.xml">Getting device info.</CMD>
  <!-- "pull" - Reads "body" data from device and saves it to "file" on host. -->
  <CMD type="show" file="device.xml" />
  <!-- "show" - Parse and show device info in "file". -->
  <CMD type="push" body="$ for module in lzo ubifs ubi mtdchar mtdconcat gpmi; do modprobe
$module; done">Install modules</CMD>
  <CMD type="push" body="mknod class/mtd,mtd0,/dev/mtd0"/>
  <!-- "push" - Sends "body" string to device. -->
  <CMD type="push" body="mknod class/mtd,mtd1,/dev/mtd1"/>
  <CMD type="push" body="mknod class/misc,ubi_ctrl,/dev/ubi_ctrl"/>
  <CMD type="push" body="send" file="imx28_linux.sb">Sending firmware</CMD>
  <CMD type="push" body="$ kobs-ng init -d $FILE">Flashing firmware</CMD>
  <CMD type="push">Done</CMD>
</LIST>
```


freescale semiconductor

What is “Updater Firmware”?

- For i.MX28 & i.MX233, it’s an executable called updater.sb (or update_ivt.sb for Hab enabled boot flow). This executable enables the device to “talk” to the host (via commands defined in ucl.xml).
- Manufacturing Tool sends updater.sb to device to be run. After booting updater.sb, the device can accept Host Updater Commands.



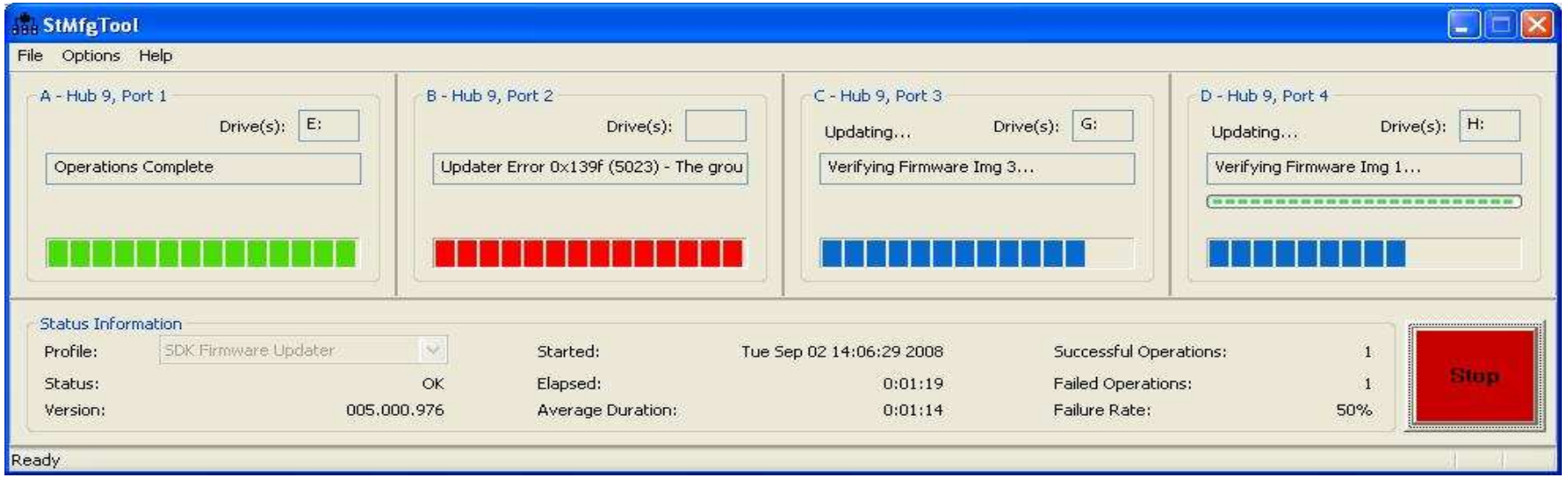
- Use of ltib to customize the “updater” for your board.
  `./ltib -selectype` (and select “Mfg firmware” profile)

A look at an example profile in ucl.xml

```
<LIST name="SD(without uboot)" desc="Install to SD card">
  <CMD type="boot" body="Recovery" file="updater_ivt.sb" timeout="60" >Booting update firmware.</CMD>
  <CMD type="find" body="Updater" timeout="60"/>
  <CMD type="pull" body="?" file="device.xml">Getting device info</CMD>
  <CMD type="show" file="device.xml"/>
  <CMD type="push" body="mknod block,mmcblk0,/dev/mmcblk0,block"/>
  <CMD type="push" body="send" file="fdisk-u.input">Sending fdisk input</CMD>
  <CMD type="push" body="$ fdisk -u /dev/mmcblk0 < $FILE">Partitioning SD card</CMD>
  <CMD type="push" body="mknod block/mmcblk0,mmcblk0p1,/dev/mmcblk0p1,block"/>
  <CMD type="push" body="mknod block/mmcblk0,mmcblk0p2,/dev/mmcblk0p2,block"/>
  <CMD type="push" body="mknod block/mmcblk0,mmcblk0p3,/dev/mmcblk0p3,block"/>
  <CMD type="push" body="send" file="files/imx28_ivt_linux.sb">Sending linux image</CMD>
  <CMD type="push" body="$ sdimage -f $FILE -d /dev/mmcblk0">Writing linux image</CMD>
  <CMD type="push" body="$ mkdir -p /mnt/mmcblk0p3"/>
  <CMD type="push" body="$ mkfs.ext3 -j /dev/mmcblk0p3">Formatting rootfs partition</CMD>
  <CMD type="push" body="$ mount /dev/mmcblk0p3 /mnt/mmcblk0p3"/>
  <CMD type="push" body="pipe tar -jxv -C /mnt/mmcblk0p3" file="files/rootfs.tar.bz2">Sending and writting root
  <CMD type="push" body="frf">Finishing rootfs write</CMD>
  <CMD type="push" body="$ umount /mnt/mmcblk0p3">Unmounting rootfs partition</CMD>
  <CMD type="push" body="!3">Done</CMD>
</LIST>

</UCL>
```

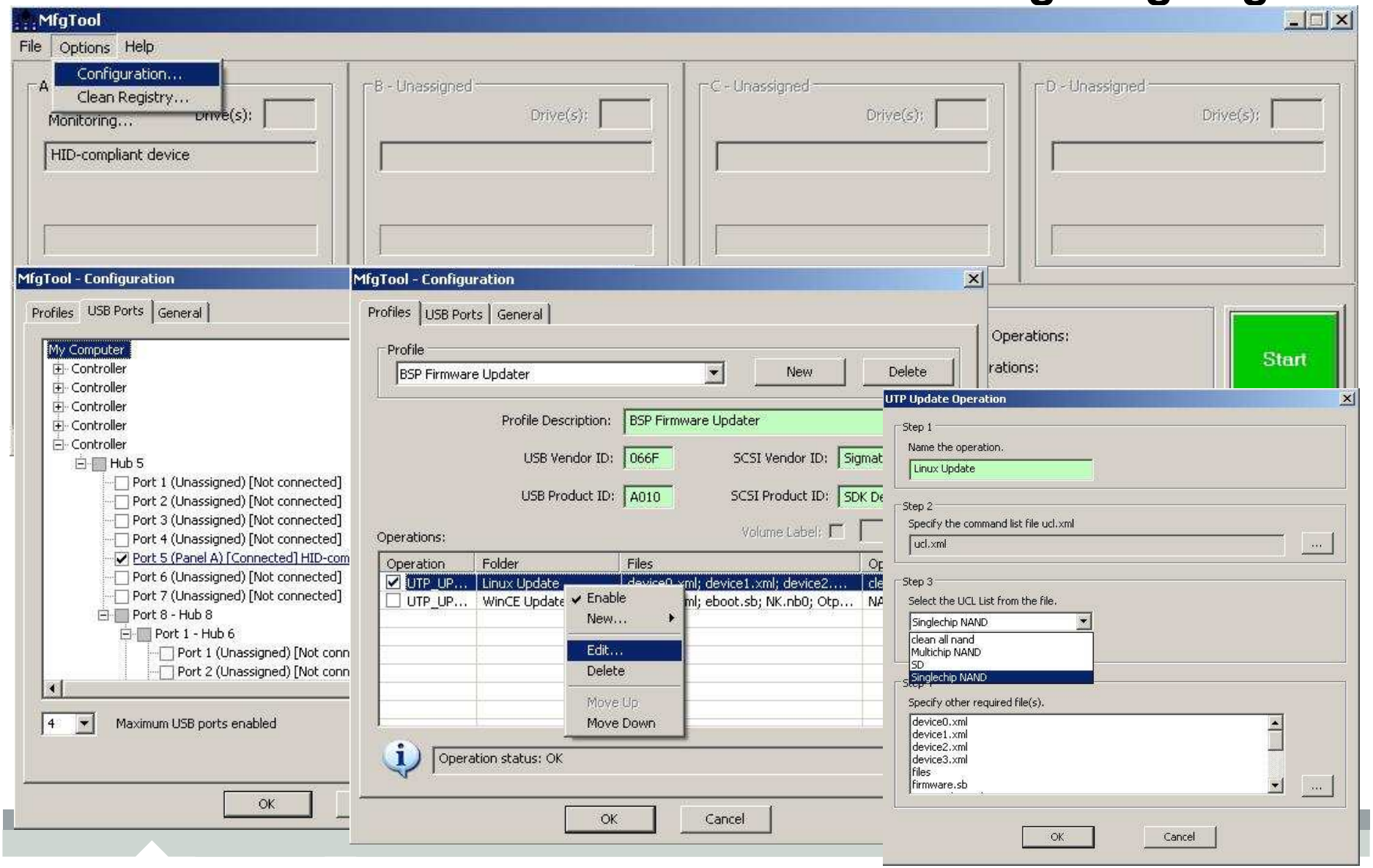
GUI and Architecture



<table>
<tr><td colspan="4">Port Manager GUI</td></tr>
<tr><td colspan="4">USB Port</td></tr>
<tr><td colspan="4">Device</td></tr>
<tr><td>Recovery-Mode</td><td colspan="2">UTP Updater-Mode</td><td>MX Updater-Mode</td></tr>
<tr><td>BLTC API</td><td colspan="2">UTP API</td><td>SDP API</td></tr>
<tr><td rowspan="2">Boot Loader Transport Control (BLTC)</td><td colspan="2">Updater Transport Protocol (UTP)</td><td rowspan="2">Serial Download Protocol (SDP)</td></tr>
<tr><td>Volume (volsnap.sys)
Disk (disk.sys)</td><td>Bulk-Only Transport</td></tr>
<tr><td>USB HID (hidusb.sys)</td><td>USB MSC (usbstor.sys)</td><td colspan="2">Jungo/WinUSB/WDF Bulk I/O</td></tr>
<tr><td>ROM</td><td colspan="2">"updater.sb" (ThreadX, WinCE, Linux)</td><td>ROM</td></tr>
<tr><td>STMP 37xx/i.MX23/28</td><td colspan="2">STMP 37xx/i.MX23/i.MX51/25</td><td>i.MX51/25/35</td></tr>
</table>

freescale ™ semiconductor

Configuring Mfgtool



freescale™ semiconductor


freescale™
semiconductor