Manufacturing Tool V2 (MFGTool2) Linux or Android Firmware Development Guide

Contents

How to Configure U-Boot and Kernel Image for MFG.

U-Boot:

1. Add Mfgtool configuration header file in : ${u_boot_root}\include\configs
   The file name should align with existing configuration file name. Ex. mx6q_sabresd.h is used to generate normal U-Boot and mx6q_sabresd_mfg.h is used to generate the U-Boot for MFGTool firmware.
   Generally, you can create it from the existing configuration file.
   The following MACROS must be defined:

   #define CONFIG_MFG
   #define CONFIG_CMDLINE_TAG
   #define CONFIG_REVISION_TAG
   #define CONFIG_SETUP_MEMORY_TAGS
   #define CONFIG_INITRD_TAG
   #define CONFIG_BOOTDELAY 0
   #define CONFIG_BOOTARGS "console=ttymxc0,115200 "
   "rdinit=/linuxrc"
   #define CONFIG_BOOTCOMMAND "bootm 0x10800000 0x10c00000"
   ; 0x10800000 is the load address of the kernel image. It neesd to be changed according to your specific platform memory configuration.
   ; 0x10c00000 is the initrd address. It needs to be changed according to your specific platform memory configuration.
   #define CONFIG_ENV_IS_NOWHERE 1;avoid reading U-Boot command from storage.

   If a new bsp arg needs to be added, one can add it by following the contents in CONFIG_BOOTARGS, for instance:

   #define CONFIG_BOOTARGS "console=ttymxc0,115200 "
   "rdinit=/linuxrc arm_freq=800"

2. Build the uboot bin for MFGTool V2.
3. Copy the uboot bin to the “Profiles\${TARGET_PROFILE_NAME}\OS Firmware”, such as: “Profiles\MX6Q Linux Update\OS Firmware”. This uboot bin file must be located in this directory.
4. Rename the uboot bin to what you want. Please note that it should align with the file name assigned in the ucl2.xml operation list.
   For example, in your operation list, there should be a <*CMD/*> as below:
   <CMD state="BootStrap" type="boot" body="BootStrap" file ="u-boot-mx6q-arm2.bin" >Loading U-boot</CMD>
   The name of your uboot bin should be “u-boot-mx6q-arm2.bin”.

Kernel:

1. Add new configuration for MFG firmware. Normally, you can copy it from the existing configuration file.
   If you take the i.MX 6 for instance, the configuration file is usually located in arch/arm/configs/. There is an imx6_defconfig which is used for i.MX 6 serial. We can add an imx6_updater_defconfig for MFG firmware. Note the following for the imx6_updater_defconfig:
   - Must build in USB and storage related driver, such as SD Card ...
   - Must build in mass storage gadget class driver
   - Must define CONFIG_FSL_UTP=y
   - Build in initramfs support
   - Enable watchdog
     CONFIG_WATCHDOG=y
     CONFIG_SOFT_WATCHDOG=y
2. Build the kernel image for MFGTool V2.
3. Copy kernel image to the "Profiles\${TARGET_PROFILE_NAME}\OS Firmware", such as: "Profiles\MX6Q Linux Update\OS Firmware". This kernel image file must be located in this directory.
4. Rename kernel image to what you want. Please note that it should align with the file name assigned in the ucl2.xml operation list.
   For example, in your operation list, there should be a *<CMD/>* as below:

   <CMD state="BootStrap" type="load" file="uImage" address="0x10800000"
   loadSection="OTH" setSection="OTH" HasFlashHeader="FALSE" >Loading Kernel.</CMD>

   The name of your uboot bin should be "uImage".

How to Generate U-Boot and Kernel Image for MFG

Please refer to the BSP User Guide, chapter Build Manufacturing Firmware.

How to Add UUC into Rootfs for MFGTool.

UUC is an application which is running firmware on the target device. It is based on the UTP/USB, receiving and executing the commands from Host side transferred through the <CMD/> defined in ucl2.xml. It should be added into rootfs for MFGTool.

1. Configuration file:

   Add new configuration file in config\platform\imx, such as mx6q_arm2_mfg_config.

   Please refer to config\platform\imx\mx6q_arm2_mfg_config as an example.

2. Add config/platform/imx/imxXX_updater.cf

- config/plaftorm/imx/main.lkc , add U-Boot new configuration when PKG_KERNEL_UPDATER is defined.
- config/platform/imx/main.lkc, add kernel new configuration when PKG_KERNEL_UPDATER is defined.
- Add CONFIG_PKG_UUC=y in imxXX_updater.cf to enable "UUC" package.
- Add new cf at config/platform/imx/preconfigs.lkc, default imx6q_updater.cf if (PCF_PLATFORM_IMX6Q && PCF_UPDATER_PROFILE) is selected.

3. ./ltib, choose your platform and "mfg firmware profile", then config/platform/imx/.config imxXX_updater.cf

MFG Linux Structure Demonstration

UTP Device Driver

Kernel driver works in UTP protocol. It is located in:

KERNEL_ROOT/drivers/usb/gadget/file_storage.c

KERNEL_ROOT/drivers/usb/gadget/fsl_updater.c

utp_handle_message is the entry of each UTP command. The command is defined in file_storage.c

It is a char (in fact, it is registered as misc driver) device driver, and combines with usb mass storage driver file_storage. It mainly handles UTP messages according to freescale UTP protocol and kinds of message types. It supplies the read/write interface for user application to get the UTP message and returns its execution results.

There are two wait queues, utp_context.wq and utp_context.list_full_wq. The utp_context.wq is used to sync read/write UTP messages, and the utp_context.list_full_wq is used to avoid so many unhandled UTP PUT(the data from the host to device) messages.

At the very beginning of utp_handle_command defined in uu.c, a function utp_can_busy is called to decide if a command needs a busy stage. If the command needs it, then UUC sends busy state to UTP device.

There are two double-linked lists for read/write UTP message. The UTP message will be listed at these two double-linked lists.

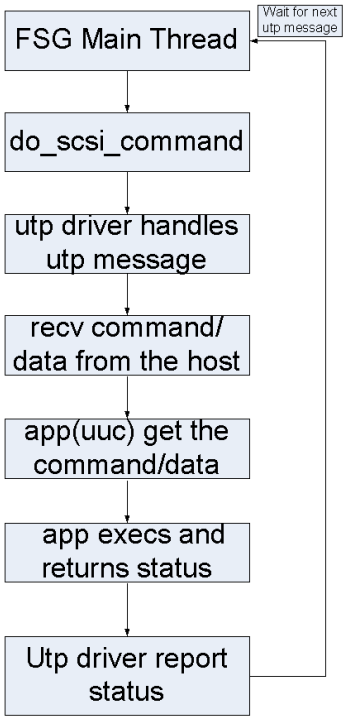
UUC

UUC is responsible for command execution in a shell. It is located in: uuc-xxx-version/uu.c

It is the user program that decodes the UTP message and executes it. It is blocked when reading UTP message. The device driver will unblock the UUC process after it gets the UTP message from the host.

The UUC returns error message with non-zero value in UTP message to the device driver.

Below is the work flow for UTP message handling. EXEC and PUT data are taken as examples:



How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support