RTD1619 NAS SDK

Image-Builder

Image-Builder

- Build Realtek SoC Image file
- Prepare Files for Rescue System
- Customization
    - Specify Firmware Layout in Storage
        - Edit feed.conf to descript storage layout
    - Support Customer's Root Filesystem
        - Edit feed.conf to descript content of root partition
    - Easy to Set
        - Kernel Boot Argument
        - MAC Address
        - IP information for Bootcode
        - Edit file, bootargs.conf.{spi,emmc}

install.img

- Realtek SoC Image file which contains
  - Bluecore.audio
  - Kernel Image
  - DTB for Normal Boot up
  - DTB for Rescue Boot up
  - Root Filesystem for Rescue System
  - Root Filesystem for Normal Boot up (if eMMc)
  - Layout description file, config.txt
  - Storage Writer, installer

Directories

- build_image.sh
  - Build Script
- arm_bin/
  - arm binary files for running in rescue system
- x86_bin/
  - X86 binary files for creating install.img

Directories

- rescue-rootfs/
  - Rescue Root File System.
  - Use initramfs.sh to compress/decompress rescue root filesystem image.
  - Size limit is 1 MB (1048576 bytes)
    - If exceeding the limit, change CONFIG_ROOTFS_RESCUE_SIZE in Bootcode's U-Boot64/include/configs/rtd161x_qa_{board}.h as in device tree
    - For example, define CONFIG_ROOTFS_RESCUE_SIZE as 0x200000 in rtd161x_qa_spi_64.h when device tree has initrd-start=<0x02200000> and initrd-end=<0x02400000> where initrd start and end are defined in kernel source's include/soc/realtek/memory.h as ROOTFS_RESCUE_START and ROOTFS_RESCUE_END respectively
- feed/
  - Gather Firmware Files
  - Edit feed.conf to descript storage layout
    - Two example files, feeds.conf.emmc, feeds.conf.spi

feed.conf Content

- storage
  - Define storage type
- storage_size
  - Define size of storage in unit of byte
- storage_align
  - Define the alignment of storage block in unit of byte
- storage_start_address
  - The start address of available are for storing firmware. DO NOT CHANGE!
- lzma
  - Use LZMA to compress bluecore.audio and kernel. Recommend for SPI

feed.conf Content

- bootargs
  - Insert U-Boot variables
  - Configuration File
    - bootargs.conf.emmc
    - bootargs.conf.spi
- Content of bootargs.conf.{emmc,spi}
  - ethaddr=00:10:20:30:40:50
  - gatewayip=192.168.100.254
  - ipaddr=192.168.100.1
  - netmask=255.255.255.0
  - kernelargs=mtdparts=RtkSFC:2176k(Boot)ro,-(FW) init=/etc/init root=/dev/sda1 rootfstype=ext4 rootwait

feed.conf Content/bootargs

- Content of bootargs.conf.{emmc,spi}
  - ethaddr
    - Device MAC address
  - gatewayip ➔Bootcode ifconfig
  - ipaddr ➔Bootcode ifconfig
  - netmask ➔Bootcode ifconfig
  - Kernelargs
    - Part of kernel boot arguments
    - Kernel boot arguments is combined with kernelargs and bootargs in DTB files
      - Bootargs = bootargs in dtb + kernelargs
      - Kernelargs could be edited in bootcode and easy to change
        - Init=/et/init or /lib/systemd
        - root=/mmcblk0px
        - rootfstype=squashfs or ext4
        - loglevel

feed.conf Content

- update_1stfw, update_2ndfw
  - Image-Builder Supports Dual Firmware Updating.
  - Set to ‘y’ to update the set of firmwares.
  - Set to ‘n’ to ignore the set of firmwares while burning the storage.
- seqnum_1stfw, seqnum_2ndfw
  - Firmware version
  - Bootcode compares sequnum to decide the latest firmware set. Bigger number is the latest firmware.

feed.conf Content

- kerneldtb_file
  - Filename of DTB for normal booting
- kerneldtb_zone
  - The size of a storage area for storing the kerneldtb_file
- kerneldtb_1stfw_addr, kerneldtb_2ndfw_addr
  - The start add of first/second kerneldtb_zone

ALL values MUST aligned by storage_align, in unit of byte.

feed.conf Content

- rescuedtb_file
  - Filename of DTB for rescue system booting
- rescuedtb_zone
  - The size of a storage area for storing the rescuedtb_file
- rescuedtb_1stfw_addr, rescuedtb_2ndfw_addr
  - The start add of first/second rescuedtb_zone

ALL values MUST aligned by storage_align, in unit of byte.

feed.conf Content

- rescuefs_file
  - Filename of initramfs image for rescue system booting
  - The example file is rescue-rootfs/rescue_rootfs.cpio.gz
- rescuefs_zone
  - The size of a storage area for storing the rescuedtb_file
- rescuefs_1stfw_addr, rescuefs_2ndfw_addr
  - The start add of first/second rescuefs_zone

ALL values MUST aligned by storage_align, in unit of byte.

feed.conf Content

- bluecore_file
  - Filename of bluecore.audio
  - The example file is Packages/fw/bluecore.audio/bluecore.audio.zip
  - Unzip it first
- bluecore_zone
  - The size of a storage area for storing the bluecore_file
- bluecore_1stfw_addr , bluecore_2ndfw_addr
  - The start add of first/second bluecore_zone

ALL values MUST aligned by storage_align, in unit of byte.

feed.conf Content

- kernel_file
  - Filename of kernel image
  - The file is kernel/arch/arm64/boot/Image
- kernel_zone
  - The size of a storage area for storing the kernel_file
- kernel_1stfw_addr , kernel_2ndfw_addr
  - The start add of first/second kernel_zone

ALL values MUST aligned by storage_align, in unit of byte.

feed.conf Content

- bootlogo_file
  - Filename of boot logo image
  - For videoplayback configuration only
- bootlogo_zone
  - The size of a storage area for storing the kernel_file
- bootlogo_1stfw_addr , bootlogo_2ndfw_addr
  - The start address of bootlogo_zone
  - By default, there is only one bootlogo file will be placed in the storage. Therefore, addr. of bootlogo in two fw. entries are the same.
- For more information about customizing bootlogo, please read the readme file for boot logo.(readme.bootlogo.pdf)

ALL values MUST aligned by storage_align, in unit of byte.

feed.conf Content

- bootpart_dir
  - Directory name of root filesystem
  - For eMMc only
  - Customer can build rootfs by themselves
- bootpart_type
  - For eMMc, support squashfs only
- bootpart_zone
  - The size of a storage area for storing the boot partition
- bootpart_addr
  - The start address of bootpart_zone

ALL values MUST aligned by storage_align, in unit of byte.

feed.conf Content

- swap_part
  - Set ‘y’ to enable eMMc swap partition
- swap_part_size_MB
  - The size of swap partition, in unit of MB

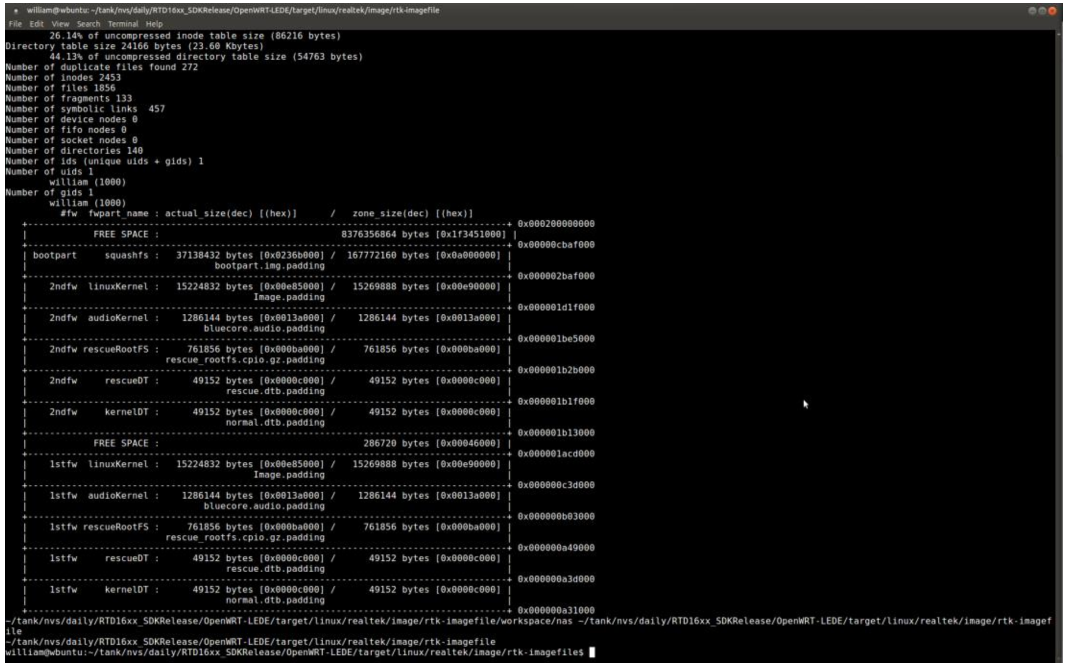
ALL values MUST aligned by storage_align, in unit of byte.

How to Use Image-Builder

- Prepare Firmwares
  - Kernel
    - arch/arm64/boot/Image
  - Rescue System DTB
    - arch/arm64/boot/dts/realtek/rtd16xx/rtd-1619-nas-qa-rescue.dtb
  - Normal Opernation DTB
    - arch/arm64/boot/dts/realtek/rtd16xx/rtd-1619-nas-mjolnir-2GB.dtb
  - Bluecore.audio
    - Packages/fw/bluecore.audio/bluecore.audio.zip, unzip it
  - Rescue System Root Filesystem
    - Image-Builder/rescue-rootfs/rescue_rootfs.cpio.gz
  - Root Filesystem
    - OpenWRT-LEDE/build_dir/target-aarch64_cortex-a55_glibc/root-realtek
- Copy Firmwares to Image-Builder/feed

How to Use Image-Builder

- Edit feed/feed.conf
  - Reference files, feed.conf.{spi,emmc}
- In Image-Builder
  - Run command./build-image.sh feed
  - X86/storage_layout/ layout-checker helps to check the position of each firmware zone.


william@wbuntu: ~/tank/nvs/daily/RTD16xx_SDKRelease/OpenWRT-LEDE/target/linux/realtek/image/rtk-imagefile
File
Edit
View
Search
Terminal
Help

```
        26.14% of uncompressed inode table size (86216 bytes)
Directory table size 24166 bytes (23.60 Kbytes)
        44.13% of uncompressed directory table size (54763 bytes)
Number of duplicate files found 272
Number of inodes 2453
Number of files 1856
Number of fragments 133
Number of symbolic links  457
Number of device nodes 0
Number of fifo nodes 0
Number of socket nodes 0
Number of directories 140
Number of ids (unique uids + gids) 1
Number of uids 1
        william (1000)
Number of gids 1
        william (1000)
          #fw  fwpart_name : actual_size(dec) [(hex)]      /   zone_size(dec) [(hex)]
   +---------------------------------------------------------------------------------------+ 0x000200000000
   |            FREE SPACE :                                 8376356864 bytes [0x1f3451000] |
   +---------------------------------------------------------------------------------------+ 0x00000cbaf000
   | bootpart     squashfs :   37138432 bytes [0x0236b000] /  167772160 bytes [0x0a000000] |
   |                                  bootpart.img.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000002baf000
   |    2ndfw  linuxKernel :   15224832 bytes [0x00e85000] /   15269888 bytes [0x00e90000] |
   |                                         Image.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000001d1f000
   |    2ndfw  audioKernel :    1286144 bytes [0x0013a000] /    1286144 bytes [0x0013a000] |
   |                                bluecore.audio.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000001be5000
   |    2ndfw rescueRootFS :     761856 bytes [0x000ba000] /     761856 bytes [0x000ba000] |
   |                         rescue_rootfs.cpio.gz.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000001b2b000
   |    2ndfw     rescueDT :      49152 bytes [0x0000c000] /      49152 bytes [0x0000c000] |
   |                                    rescue.dtb.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000001b1f000
   |    2ndfw     kernelDT :      49152 bytes [0x0000c000] /      49152 bytes [0x0000c000] |
   |                                    normal.dtb.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000001b13000
   |            FREE SPACE :                                     286720 bytes [0x00046000] |
   +---------------------------------------------------------------------------------------+ 0x000001acd000
   |    1stfw  linuxKernel :   15224832 bytes [0x00e85000] /   15269888 bytes [0x00e90000] |
   |                                         Image.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000000c3d000
   |    1stfw  audioKernel :    1286144 bytes [0x0013a000] /    1286144 bytes [0x0013a000] |
   |                                bluecore.audio.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000000b03000
   |    1stfw rescueRootFS :     761856 bytes [0x000ba000] /     761856 bytes [0x000ba000] |
   |                         rescue_rootfs.cpio.gz.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000000a49000
   |    1stfw     rescueDT :      49152 bytes [0x0000c000] /      49152 bytes [0x0000c000] |
   |                                    rescue.dtb.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000000a3d000
   |    1stfw     kernelDT :      49152 bytes [0x0000c000] /      49152 bytes [0x0000c000] |
   |                                    normal.dtb.padding                                 |
   +---------------------------------------------------------------------------------------+ 0x000000a31000
~/tank/nvs/daily/RTD16xx_SDKRelease/OpenWRT-LEDE/target/linux/realtek/image/rtk-imagefile/workspace/nas ~/tank/nvs/daily/RTD16xx_SDKRelease/OpenWRT-LEDE/target/linux/realtek/image/rtk-imagef
ile
~/tank/nvs/daily/RTD16xx_SDKRelease/OpenWRT-LEDE/target/linux/realtek/image/rtk-imagefile
william@wbuntu:~/tank/nvs/daily/RTD16xx_SDKRelease/OpenWRT-LEDE/target/linux/realtek/image/rtk-imagefile$ 
```

How to Use Image-Builder--Output

- Image-Builder/install.img
  - Rescue system use install.img to upgrade firmwares
- Image-Builder/workspace/rescue
  - Rescue system files
    - {spi,emmc}.uImage
    - rescue.{spi,emmc}.dtb
    - rescue.root.{spi,emmc}.cpio.gz_pad.img