RTK Bootcode for RTD1619

Agenda

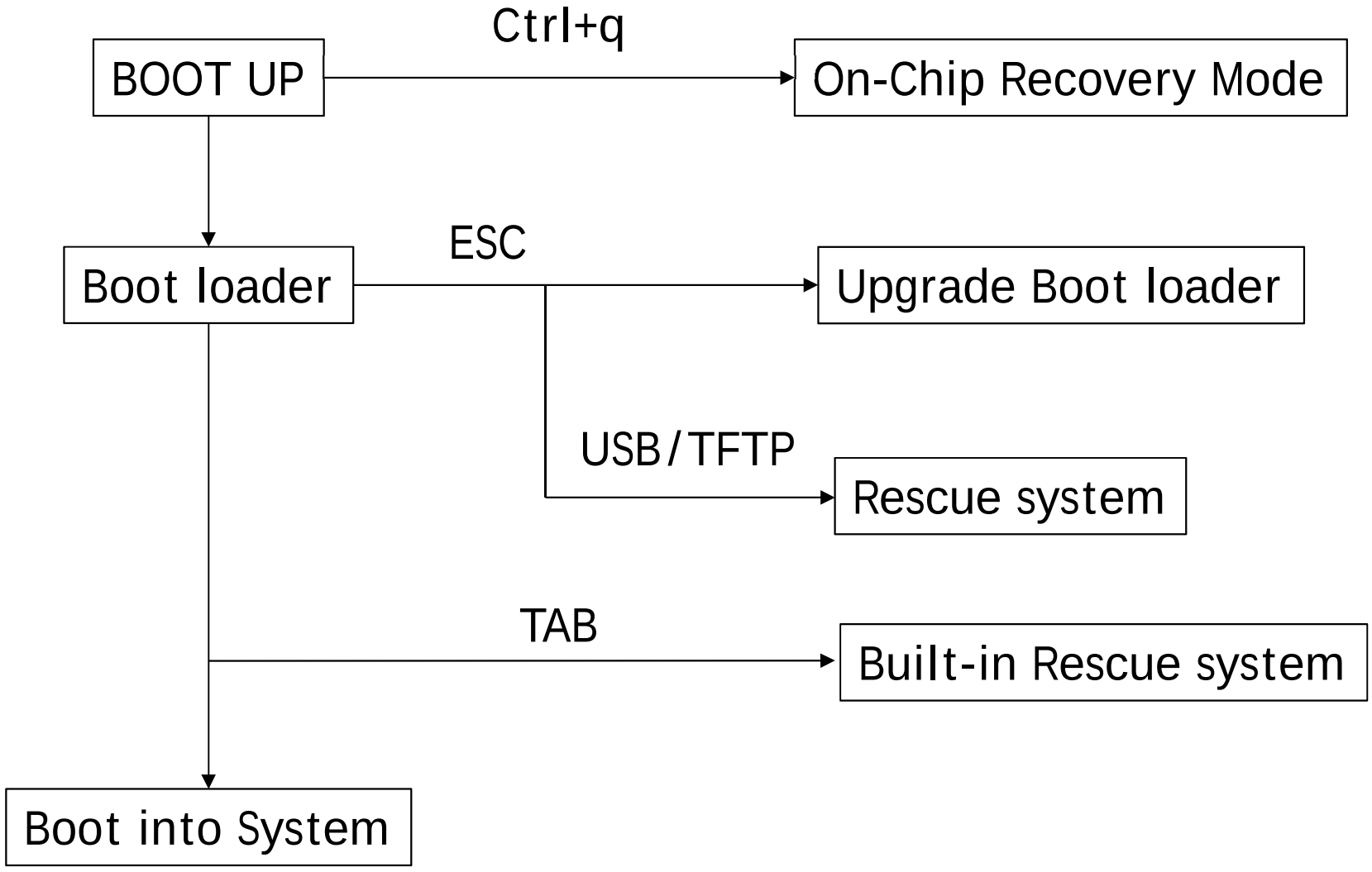
- Build Boot loader
- On-Chip Recovery mode
- Upgrade Boot loader
- Rescue System
- Dual Firmware Boot
- Customize kernel boot arguments

Build Boot loader - SPI

- Realtek uses U-Boot as Boot Loader in RTD1619
- Path: RTD16xx_SDKRelease/Bootcode/U-Boot64
- Build script: build.sh
- Usage: ./build.sh RTD16xx_spi
- Output: RTD16xx_SDKRelease/Bootcode/U-Boot64/DVRBOOT_OUT/RTD16xx_spi/
  - (1) Bootloader with burning program
    - A01-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_spi.bin
  - (2) Bootloader
    - A01-Recovery-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_spi.bin
  - (3) HWsetting
    - hw_setting/0001-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_spi.bin

Build Boot loader - eMMC

- Realtek uses U-Boot as Boot Loader in RTD1619
- Path: RTD16xx_SDKRelease/Bootcode/U-Boot64
- Build script: build.sh
- Usage: ./build.sh RTD16xx_emmc
- Output: RTD16xx_SDKRelease/Bootcode/U-Boot64/DVRBOOT_OUT/RTD16xx_emmc/
  - (1) Bootloader with burning program
    - A01-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin
  - (2) Bootloader
    - 1. A01-Recovery-uda-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin
    - 2. A01-Recovery-boot-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin
  - (3) HWsetting
    - hw_setting/0001-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin


BOOT UP
Ctrl+q
On-Chip Recovery Mode
Boot loader
ESC
Upgrade Boot loader
USB/TFTP
Rescue system
TAB
Built-in Rescue system
Boot into System

On-Chip Recovery mode - SPI

- This mode is used for bootloader recovery when the bootloader inside is corrupted
- Hyperterm is recommended to use
- Press ctrl+q when booting up until following console appear
  - d/g/r>
- Press h and then transfer Hwsetting(3) by y-modem
  - hw_setting/0001-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-spi.bin
- Press d and then transfer bootloader(2) by y-modem
  - A01-Recovery-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-spi.bin
- Press g to start recovery process

On-Chip Recovery mode - eMMC

- This mode is used for bootloader recovery when the bootloader inside is corrupted
- Hyperterm is recommended to use
- Press ctrl+q when booting up until following console appear
  - d/g/r>
- Press h and then transfer Hwsetting(3) by y-modem
  - hw_setting/0001-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin
- Press d and then transfer bootloader(2) -1 by y-modem
  - A01-Recovery-uda-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin
- Press g to start recovery process
- Press d and then transfer bootloader(2) -2 by y-modem
  - A01-Recovery-boot-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin
- Press b1 to start recovery process
- Press b1 to start recovery process

Upgrade Boot loader

- This is for boot loader upgrade when current boot loader is workable
- Press ESC when booting up to enter U-Boot console
  - Realtek>
- Transfer bootloader with burning program(1) to DRAM
  - By USB
    - usb start
    - fatload usb 0:1 0x1500000 A01-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_spi.bin or RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin
    - go 0x1500000
  - By tftp
    - tftp 0x1500000 A01-RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_spi.bin or RTD161x_hwsetting_BOOT_2DDR4_8Gb_s2666-nas-RTD16xx_emmc.bin
    - go 0x1500000

Rescue system

- This is used to install install.img onto RTD1619
- There are 2 ways to boot into rescue system
  - 1. use built-in rescue system
  - 2. use rescue system from USB or TFTP

- 1. For build-in rescue system, just press TAB when booting up, then the built-in rescue system is used
- Once booting into rescue system, put install.img in USB top directory and plug into RTD1619, the install process will start automatically

Rescue system

- 2. For rescue system from USB or TFTP, we should copy following 3 files to USB or TFTP directory

path: RTD16xx_SDKRelease/OpenWRT-LEDE/target/linux/realtek/image/rtk-imagefile/workspace/rescue/

  - 1. spi.uImage
  - 2. rescue.spi.dtb
  - 3. rescue.root.spi.cpio.gz_pad.img
- USB: press ESC into U-Boot console mode with USB attached and them press ‘go r’
- TFTP: press ESC and then input following command
  - tftp $kernel_loadaddr spi.uImage;tftp $fdt_loadaddr rescue.spi.dtb;tftp $rootfs_loadaddr rescue.root.spi.cpio.gz_pad.img;go k
- put install.img in USB top directory and plug into RTD1619, the install process will start automatically

TFTP configuration

- TFTP is configurable in boot loader console mode
- Press ESC into console mode and then edit tftp related environment variables

Realtek> env set ipaddr 192.168.0.9

Realtek> env set serverip 192.168.0.56

Realtek> env set gatewayip 192.168.0.254

Realtek> env set netmask 255.255.255.0

Realtek> env save

- ping command can be used to test network

Realtek> ping 192.168.0.56

Dual Firmware Boot

```
      #fw  fwpart_name : actual_size(dec) [(hex)]      /   zone_size(dec) [(hex)]
+-------------------------------------------------------------------------------------+ 0x000001000000
|            FREE SPACE :                                    2719744 bytes [0x00298000] |
+-------------------------------------------------------------------------------------+ 0x000000d68000
|    2ndfw  linuxKernel :    3698688 bytes [0x00387000] /    4128768 bytes [0x003f0000] |
|                                  Image.lzma.padding                                 |
+-------------------------------------------------------------------------------------+ 0x000000978000
|    2ndfw  audioKernel :     311296 bytes [0x0004c000] /     393216 bytes [0x00060000] |
|                           bluecore.audio.lzma.padding                               |
+-------------------------------------------------------------------------------------+ 0x000000918000
|    2ndfw rescueRootFS :    1048576 bytes [0x00100000] /    1048576 bytes [0x00100000] |
|                         rescue_rootfs.cpio.gz.padding                               |
+-------------------------------------------------------------------------------------+ 0x000000818000
|    2ndfw     rescueDT :      49152 bytes [0x0000c000] /      49152 bytes [0x0000c000] |
|                                  rescue.dtb.padding                                 |
+-------------------------------------------------------------------------------------+ 0x00000080c000
|    2ndfw     kernelDT :      49152 bytes [0x0000c000] /      49152 bytes [0x0000c000] |
|                                  normal.dtb.padding                                 |
+-------------------------------------------------------------------------------------+ 0x000000800000
|            FREE SPACE :                                     413696 bytes [0x00065000] |
+-------------------------------------------------------------------------------------+ 0x00000079b000
|    1stfw  linuxKernel :    3698688 bytes [0x00387000] /    4128768 bytes [0x003f0000] |
|                                  Image.lzma.padding                                 |
+-------------------------------------------------------------------------------------+ 0x0000003ab000
|    1stfw  audioKernel :     311296 bytes [0x0004c000] /     393216 bytes [0x00060000] |
|                           bluecore.audio.lzma.padding                               |
+-------------------------------------------------------------------------------------+ 0x00000034b000
|    1stfw rescueRootFS :    1048576 bytes [0x00100000] /    1048576 bytes [0x00100000] |
|                         rescue_rootfs.cpio.gz.padding                               |
+-------------------------------------------------------------------------------------+ 0x00000024b000
|    1stfw     rescueDT :      49152 bytes [0x0000c000] /      49152 bytes [0x0000c000] |
|                                  rescue.dtb.padding                                 |
+-------------------------------------------------------------------------------------+ 0x00000023f000
|    1stfw     kernelDT :      49152 bytes [0x0000c000] /      49152 bytes [0x0000c000] |
|                                  normal.dtb.padding                                 |
+-------------------------------------------------------------------------------------+ 0x000000233000
```

Dual Firmware Boot

- There are 2 copies of firmware table and images (kernel, dtb, rescue rootfs… etc) installed in RTD1619
- Each firmware table has a sequence number (0-255)
- Boot loader will choose the latest (bigger) firmware table to bootup
- Once the latest firmware table or its content is corrupted, another firmware table will be chosen to bootup

Customize kernel boot arguments

Path: arch/arm/lib/bootm.c

Function: set_custom_boot_args()

- This function provides room to append programmable boot arguments.
- For example, the path of boot device might be "/dev/mmcblk0p0" or "/dev/mmcblk0p1", depending on circumstances. This can be done by adding conditional statement in this function to control boot arguments.
- The parameter "custom_boot_args" of this function is combined with kernelargs, and finally become part of kernel boot arguments.