

Device Tree

NUC970 | NUC980

江天文

1 Abstract

This document presents the devicetree usage on NUC972 platform. One shall obtain a NUC972 EVB board to experiment on programming devicetree.

2 Steps

2.1 Build Machine

The build machine requires installing the 64-bit desktop OS **Ubuntu 12.04 LTS**. Browse the site <https://ubuntu.com/download/desktop> to download the right version OS.

2.2 Build Environment Setup

Perform the following commands to setup the build environment.

```
$ sudo apt update
$ sudo apt install git build-essential libncurses-dev python
```

2.3 Obtain Buildroot

Fetch the Buildroot source from github.com.

```
$ git clone https://github.com/symfund/NUC970_Buildroot.git
```

2.4 Load Configuration

In the following paragraphs, **#{BR2}** denotes the root directory of Buildroot. By issuing command **cd #{BR2}**, change working directory to **#{BR2}**.

```
$ make list-defconfigs | grep nuvoton
$ make nuvoton_nuc972_defconfig
```

2.5 Fetch Kernel Source Package

```
$ until make linux-source; do echo "fetch linux source failed, retrying..."; done
```

2.6 Fetch Uboot Source Package

```
$ until make uboot-source; do echo "fetch uboot source failed, retrying..."; done
```

2.7 Override Kernel/Uboot Source Directory

Create a file **local.mk** in the root directory of Buildroot, the contents like below.

```
UBOOT_OVERRIDE_SRCDIR=$(CONFIG_DIR)workspace/uboot-master  
LINUX_OVERRIDE_SRCDIR=$(CONFIG_DIR)/workspace/linux-master
```

2.8 Extract Kernel/Uboot Source Packages

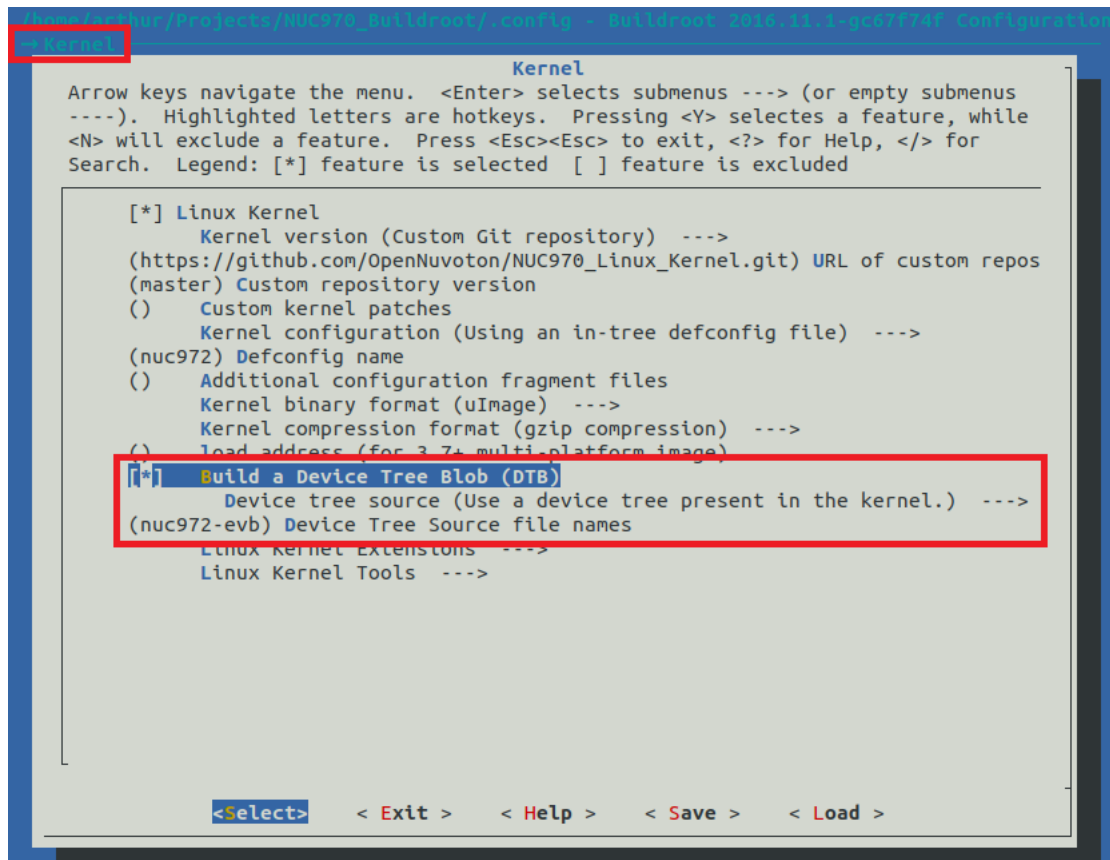
```
$ mkdir -p ${BR2}/workspace  
$ tar xzvf ${BR2}/dl/linux-master.tar.gz -C ${BR2}/workspace/  
$ tar xzvf ${BR2}/dl/uboot-master.tar.gz -C ${BR2}/workspace/
```

2.9 Configure Buildroot

Launch the below command to open a GUI window for configuring Buildroot.

```
$ make menuconfig
```

2.9.1 Enable 'Build a Device Tree Blob (DTB)'



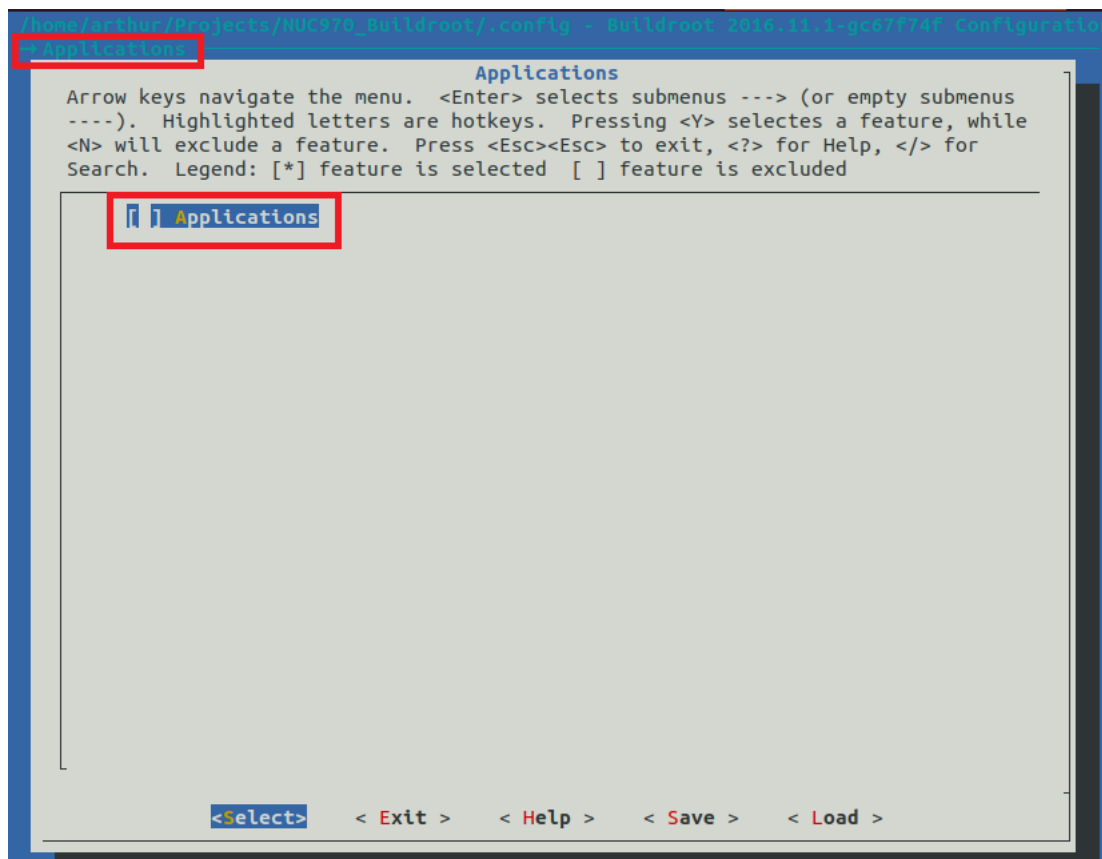
2.9.2 Enable 'Install U-Boot SPL binary image'

```

/home/arthur/Projects/NUC970_Buildroot/.config - Buildroot 2016.11.1-gcc7f74f Configuration
Bootloaders
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu
----). Highlighted letters are hotkeys. Pressing <Y> selects a feature, while
<N> will exclude a feature. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] feature is selected [ ] feature is excluded
↑(-)
[ ] AT91 Bootstrap
[ ] AT91 Bootstrap 3
[ ] AT91 DataFlashBoot
[ ] Barebox
[ ] LPC32XX CDL (kickstart and S1L)
[ ] mxs-bootlets
[ ] s500-bootloader
[*] U-Boot
    Build system (Kconfig) --->
    U-Boot Version (Custom Git repository) --->
    (https://github.com/OpenNuvoton/NUC970_U-Boot_v2016.11.git) URL of custom re
    (master) Custom repository version
    ( ) Custom U-Boot patches
    U-Boot configuration (Using an in-tree board defconfig file) --->
    (nuc970) Board defconfig
    [ ] U-Boot needs dtc
    [ ] U-Boot needs OpenSSL
    U-Boot binary format --->
    [ ] produce a .ift signed image (OMAP)
    [*] Install U-Boot SPL binary image
    (spl/u-boot-spl.bin) U-Boot SPL binary image name (NEW)
    [ ] CRC SPL Image for Altera SoC FPGA (NEW)
    [ ] Environment image ----

<Select> < Exit > < Help > < Save > < Load >
  
```

2.9.3 Disable 'Applications'



2.9.4 Enable 'ubifs root filesystem'

```

/home/arthur/Projects/NUC970_Buildroot/Buildroot-2018.11.1-gc67f74f Configuration
Buildroot images
Filesystem images
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu
----). Highlighted letters are hotkeys. Pressing <Y> selects a feature, while
<N> will exclude a feature. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] feature is selected [ ] feature is excluded

[ ] axfs root filesystem
[ ] cloop root filesystem for the target device
[ ] cpio the root filesystem (for use as an initial RAM filesystem)
[ ] cramfs root filesystem
[ ] ext2/3/4 root filesystem
[ ] initial RAM filesystem linked into linux kernel
[ ] jffs2 root filesystem
[ ] romfs root filesystem
[ ] squashfs root filesystem
[ ] tar the root filesystem
[*] ubifs root filesystem
(0x1f000) logical eraseblock size
(0x800) minimum I/O unit size (NEW)
(2048) maximum logical eraseblock count (NEW)
ubifs runtime compression (lzo) --->
Compression method (lzo) --->
(-F) Additional mkfs.ubifs options
[*] Embed into an UBI image
(0x20000) physical eraseblock size (NEW)
(2048) sub-page size
[ ] Use custom config file (NEW)
(-m 2048 -O 2048) Additional ubinize options
[ ] yaffs2 root filesystem

<select> < Exit > < Help > < Save > < Load >

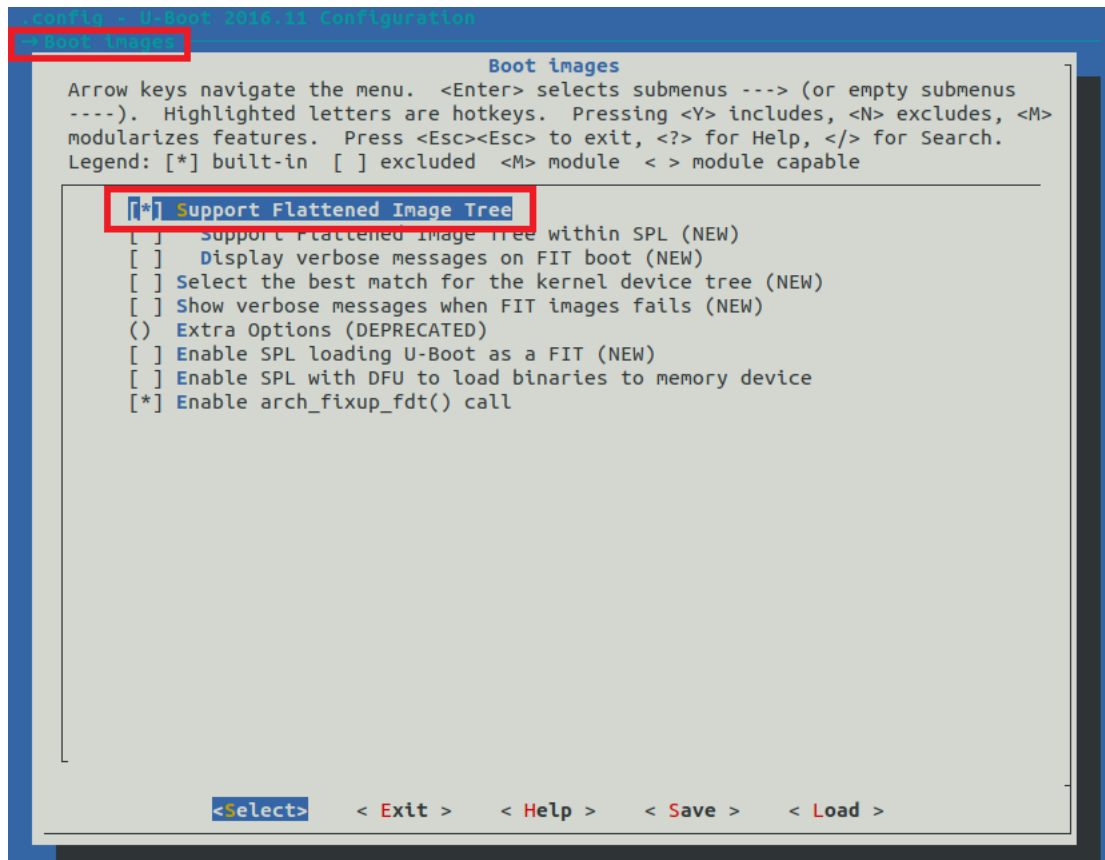
```

2.10 Configure U-Boot

Launch the below command to open a GUI window for configuring U-Boot.

```
$ make uboot-menuconfig
```

2.10.1 Enable 'Support Flattened Image Tree'



2.10.2 Enable 'the FDT library'

```

conf1 - U-Boot 2014.11 Configuration
Library routines
Library routines
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus
----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M>
modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable

[ ] Optimize libraries for speed
[*] Use private libgcc
[ ] Enable tiny printf() version
[*] Enable regular expression support
[ ] Pseudo-random library support
[ ] Support the 'dhry' command to run the dhrystone benchmark
[ ] Use RSA Library
    Hashing Support --->
    Compression Support --->
[ ] Enable function for getting errno-related string message
[*] Enable the FDT library
[ ] Enable the FDT library overlay support
[ ] Enable the FDT library for SPL
[ ] overwrite MTD partitions in DTS through defined in 'mtdparts' (NEW)
    System tables --->
[*] Support running EFI Applications in U-Boot (NEW)

<select> < Exit > < Help > < Save > < Load >

```

2.10.3 Build U-Boot

```
$ make uboot-rebuild
```

2.11 Configure Kernel

Launch the below command to open a GUI window for configuring kernel.

```
$ make linux-menuconfig
```

2.11.1 Disable 'Initial RAM filesystem support'

```

config - Linux/arm v 3.10.108 Kernel Configuration
General setup
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted letters
are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
excluded <M> module < > module capable
↑(-)
[ ] Auditing support
  IRQ subsystem --->
  Timers subsystem --->
  CPU/Task time and stats accounting --->
  RCU Subsystem --->
< > Kernel .config support
(17) Kernel log buffer size (16 => 64KB, 17 => 128KB)
[*] Control Group support --->
-* Namespaces support --->
[ ] Require conversions between uid/gids and their internal representation
[ ] Automatic process group scheduling
[ ] Enable deprecated sysfs features to support old userspace tools
[ ] Kernel->user space relay support (formerly relayfs)
[ ] Initial RAM filesystem and RAM disk (initramfs/initrd) support
[ ] Optimize for size
[ ] Configure standard kernel features (expert users) --->
[ ] Embedded system
  Kernel Performance Events And Counters --->
[ ] Disable heap randomization
  Choose SLAB allocator (SLUB (Unqueued Allocator)) --->
[ ] Profiling support
[ ] Kprobes
[ ] Optimize very unlikely/likely branches
↓(+)
<select> < Exit > < Help > < Save > < Load >

```

2.11.2 Enable 'MTD NAND Flash'

```
.config - Linux/arm 3.10.108 Kernel Configuration
Generic Driver Options
Generic Driver Options
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted letters
are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
excluded <M> module < > module capable

(/sbin/hotplug) path to uevent helper
[*] Maintain a devtmpfs filesystem to mount at /dev
[*] Automount devtmpfs at /dev, after the kernel mounted the rootfs
[ ] Select only drivers that don't need compile-time external firmware
[ ] Prevent firmware from being built
-* Userspace firmware loading support
[ ] Include in-kernel firmware blobs in kernel binary
() External firmware blobs to build into the kernel binary
[ ] Fallback user-helper invocation for firmware loading
[ ] Contiguous Memory Allocator
<*> Nuvoton NUC970/N9H30 FMI function selection
  | Select FMI device to support (Support MTD NAND Flash) --->

<Select> < Exit > < Help > < Save > < Load >
```

2.11.3 Configure 'MTD'

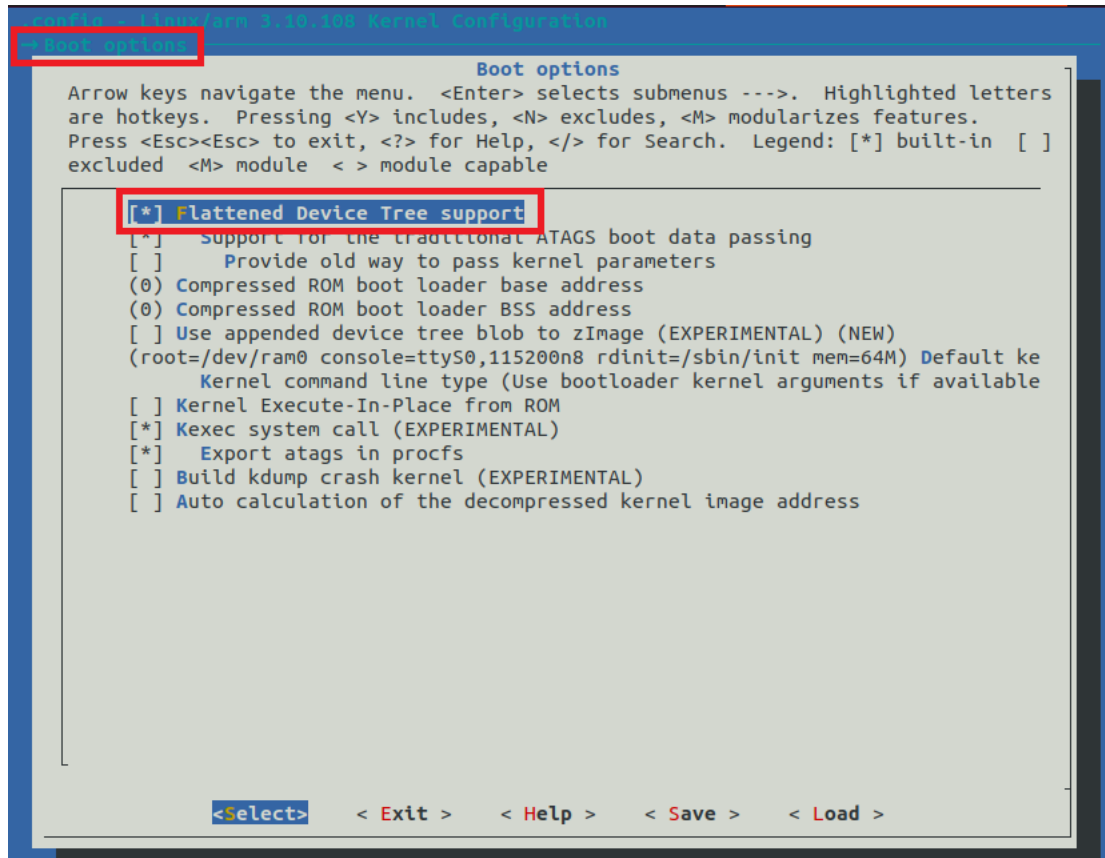
```

.config - Linux/arm 3.10.108 Kernel Configuration
device drivers - Memory Technology Device (MTD) support
Memory Technology Device (MTD) support
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted letters
are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
excluded <M> module <> module capable
↑(-)
<> RedBoot partition table parsing (NEW)
[*] Command line partition table parsing
<> ARM Firmware Suite partition parsing (NEW)
<> TI AR7 partitioning support (NEW)
*** User Modules And Translation Layers ***
* Common interface to block layer for MTD 'translation layers'
[*] Caching block device access to MTD devices
<> FTL (Flash Translation Layer) support (NEW)
<> NFTL (NAND Flash Translation Layer) support (NEW)
<> INFTL (Inverse NAND Flash Translation Layer) support (NEW)
<> Resident Flash Disk (Flash Translation Layer) support (NEW)
<> NAND SSFDC (SmartMedia) read only translation layer (NEW)
<> SmartMedia/xD new translation layer (NEW)
<> Log panic/oops to an MTD buffer (NEW)
<> Swap on MTD device support (NEW)
RAM/ROM/Flash chip drivers --->
Mapping drivers for chip access --->
Self-contained MTD device drivers --->
[ ] NAND ECC Smart Media byte order (NEW)
-*- NAND Device Support --->
<> OneNAND Device Support (NEW) --->
LPDDR flash memory drivers --->
[*] Enable UBI - Unsorted block images --->

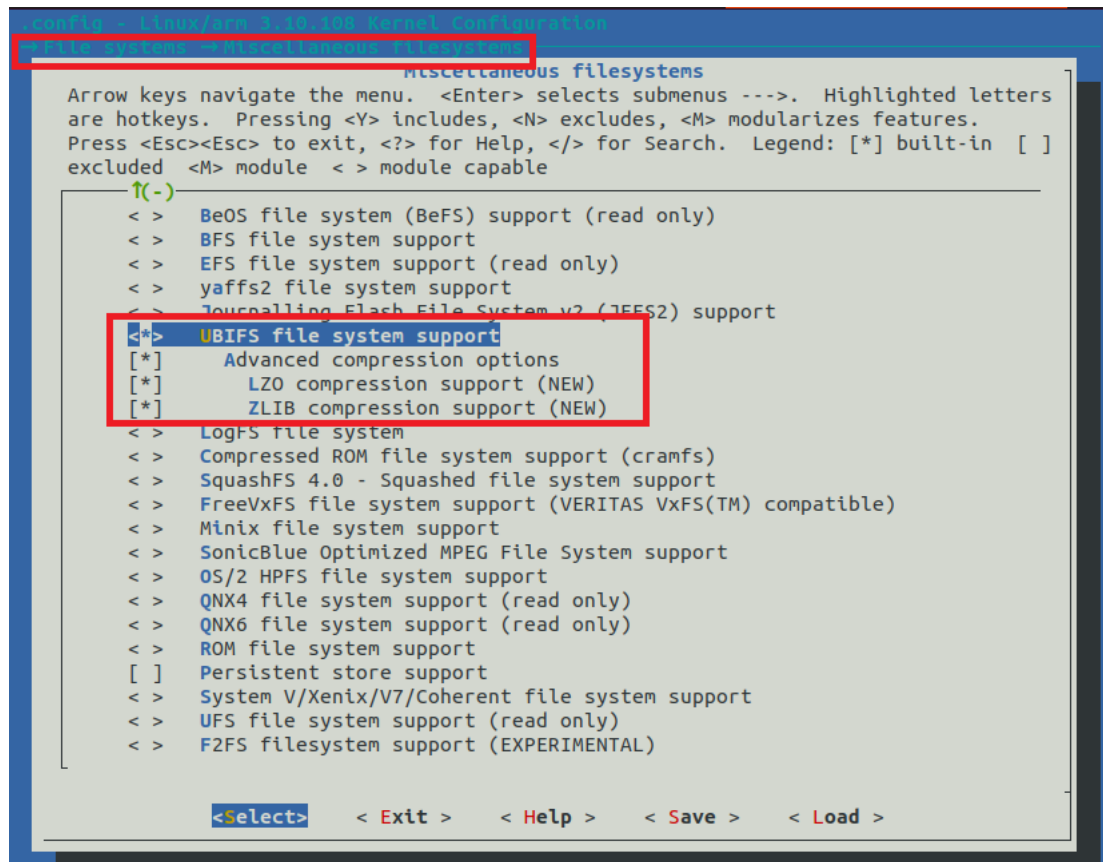
<select> < Exit > < Help > < Save > < Load >

```

2.11.4 Enable 'Flattened Device Tree Support'



2.11.5 Enable 'UBIFS file system support'



2.11.6 Modify DeviceTree

The kernel device tree file is located in

\${BR2}/workspace/linux-master/arch/arm/boot/dts/nuc972-evb.dts

```

apb {
    ...
    uart4: serial@b8000400 {
        status = "disable";
    };
    ...
};

fmi@b000d000 {
    ...
    status = "okay";
    ...
};
    
```

2.11.7 Build Kernel

```
$ make linux-rebuild
```

2.12 Build Images

```
$ make
```

The final blobs are located at the `#{BR2}/output/images` folder.

- **u-boot-spl.bin**
- **u-boot.bin**
- **ulmage**
- **rootfs.ubi**
- **nuc972-evb.dtb**

2.13 Environment File

A environment file `env.txt` is needed by NuWriter. The contents are like below

```
baudrate=115200
bootdelay=3
ethact=emac
ethaddr=00:00:00:11:66:88
stderr=serial
stdin=serial
stdout=serial
serverip=10.130.11.5
ipaddr=10.130.11.120
bootcmd=nand read 0x7fc0 0x200000 0x800000;nand read 0x1400000 0x900000
0x100000;bootm 0x7fc0 - 0x1400000
bootargs=noinitrd ubi.mtd=3 root=ubi0:rootfs rw rootfstype=ubifs
mtdparts=nand0:0x200000@0x0(u-boot),0x700000@0x200000(kernel),0x100000@
0x900000(device-tree),0x10000000@0x1000000(rootfs),-(user)
console=ttyS0,115200n8 rdinit=/sbin/init mem=64M ignore_loglevel
```

2.14 NuWriter

Image name	Image Type	Offset (HEX)
u-boot-spl.bin	uBoot	200
u-boot.bin	Data	100,000
ulmage	Data	200,000
nuc972-evb.dtb	Data	900,000
rootfs.ubi	Data	1000,000
env.txt	Environment	80,000