

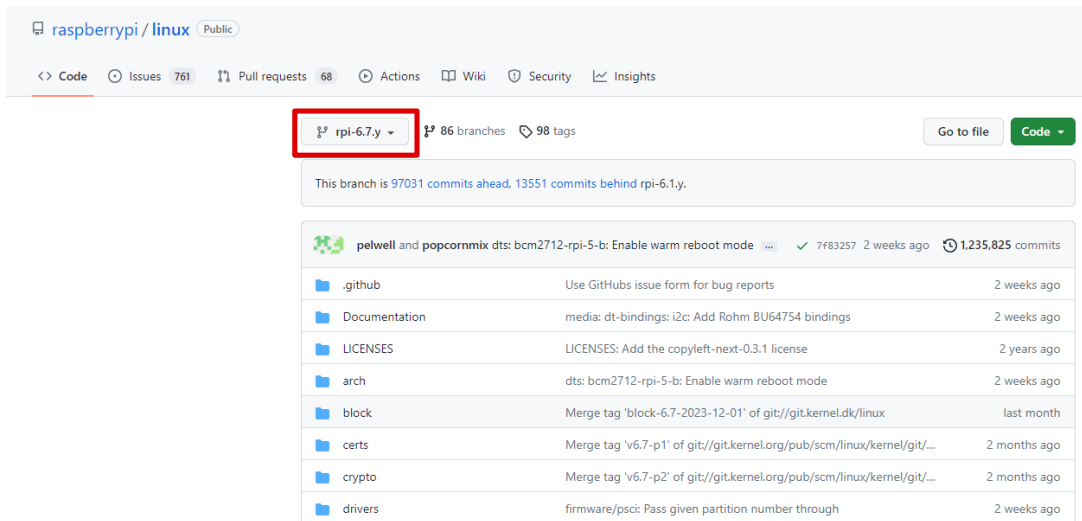
Raspberry Pi 4B Bring up

Download the latest kernel for Pi

- Command

#git clone --branch <branch> https://github.com/raspberrypi/linux

where <branch> is the name of the branch that you wish to download. We can choose **rpi-6.7.y** for Pi 4B



The screenshot shows the GitHub repository page for `raspberrypi/linux`. The `rpi-6.7.y` branch is selected and highlighted with a red box. The page displays commit history and file listings.

Commit history:

Commit	Author	Message	Time
7f83257	pelwell and popcornmix	dts: bcm2712-rpi-5-b: Enable warm reboot mode	2 weeks ago
1,235,825			commits

File listings:

File	Message	Time
.github	Use GitHub's issue form for bug reports	2 weeks ago
Documentation	media: dt-bindings: i2c: Add Rohm BU64754 bindings	2 weeks ago
LICENSES	LICENSES: Add the copyleft-next-0.3.1 license	2 years ago
arch	dts: bcm2712-rpi-5-b: Enable warm reboot mode	2 weeks ago
block	Merge tag 'block-6.7-2023-12-01' of git://git.kernel.dk/linux	last month
certs	Merge tag 'v6.7-p1' of git://git.kernel.org/pub/scm/linux/kernel/git/...	2 months ago
crypto	Merge tag 'v6.7-p2' of git://git.kernel.org/pub/scm/linux/kernel/git/...	2 months ago
drivers	firmware/psci: Pass given partition number through	2 weeks ago

Cross compile environment setting

First, you will need a suitable Linux cross-compilation host. We tend to use Ubuntu; since Raspberry Pi OS is also a Debian distribution, it means many aspects are similar, such as the command lines.

You can either do this using VirtualBox (or VMWare) on Windows, or install it directly onto your computer.

- Necessary installation packages

```
#sudo apt install git bc bison flex libssl-dev make libc6-dev libncurses5-dev
```

- Cross compiler for Pi 4B

```
#sudo apt-get install gcc-aarch64-linux-gnu
```

```
#sudo apt-get install crossbuild-essential-arm64
```

Cross compiling the kernel

- Config the kernal, use menuconfig

#make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- menuconfig

- Or, for Raspberry Pi 4B, default 64-bit build configuration

#make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- bcm2711_defconfig

- Build the kernel

#make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- Image modules
dtbs j32

Prepare the SD card

The best way to set the environment of sd card is to flash an official imager into SD card

- Download the .iso file from below link:

https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2023-12-06/2023-12-05-raspios-bookworm-arm64.img.xz?_gl=1*1smomv*_ga*MTgyODI3MDM2NC4xNzA0MjY0MzQz*_ga_22FD70LWDS*MTcwNDI2NDM0My4xLjEuMTcwNDI2OTIwMS4wLjAuMA..

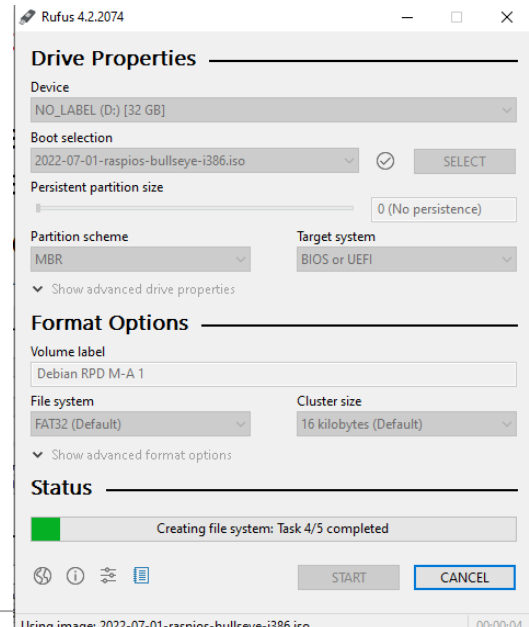
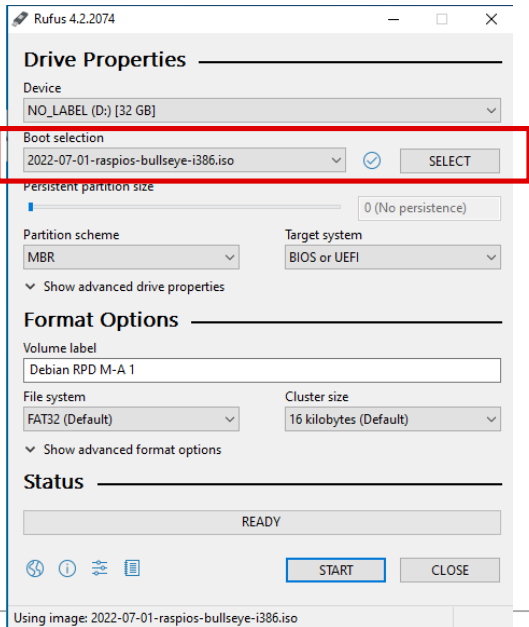
Install Directly onto the SD Card

For building the SD card, we recommend to use rufus.exe to flash .iso file.

- Rufus.exe download link:

<https://rufus.ie/downloads/>

Select the .iso file



Remount the SD card disk

- First, use command `#ls /dev/sd*` before and after plugging in your SD card to identify it, will see the below additional disk

```
lpa@lpa-VB:~/Desktop$ ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sdb /dev/sdb1 /dev/sdb2  
lpa@lpa-VB:~/Desktop$
```



- Mount the sdb1 and sdb2 for image updating

```
mkdir -p mnt  
mkdir -p mnt/fat32  
mkdir -p mnt/ext4  
mount /dev/sdb1 mnt/fat32  
mount /dev/sdb2 mnt/ext4
```

Update image&modules&dtbs into the SD Card

- Copy the kernel into the kernel path, but with a different filename - for instance, kernel-myconfig.img - rather than overwriting the kernel.img file. You can then edit the config.txt file to select the kernel that the Raspberry Pi will boot,
- For example:

```
kernel=kernel-myconfig.img
```

- Commands:

```
#export KERNEL=kernel-myconfig
```

```
#export KERNEL_PATH=<Kernel path>
```

```
#make PATH=$KERNEL_PATH ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- INSTALL_MOD_PATH=mnt/ext4  
modules_install
```

```
#cp mnt/fat32/$KERNEL.img mnt/fat32/$KERNEL-backup.img
```

```
#cp arch/arm64/boot/Image mnt/fat32/$KERNEL.img
```

```
#cp arch/arm64/boot/dts/broadcom/*.dtb mnt/fat32/
```

```
#cp arch/arm64/boot/dts/overlays/*.dtb* mnt/fat32/overlays/
```

```
#cp arch/arm64/boot/dts/overlays/README mnt/fat32/overlays/
```

Unmount the SD card disk

- Umount the folders:

```
umount mnt/fat32  
umount mnt/ext4
```

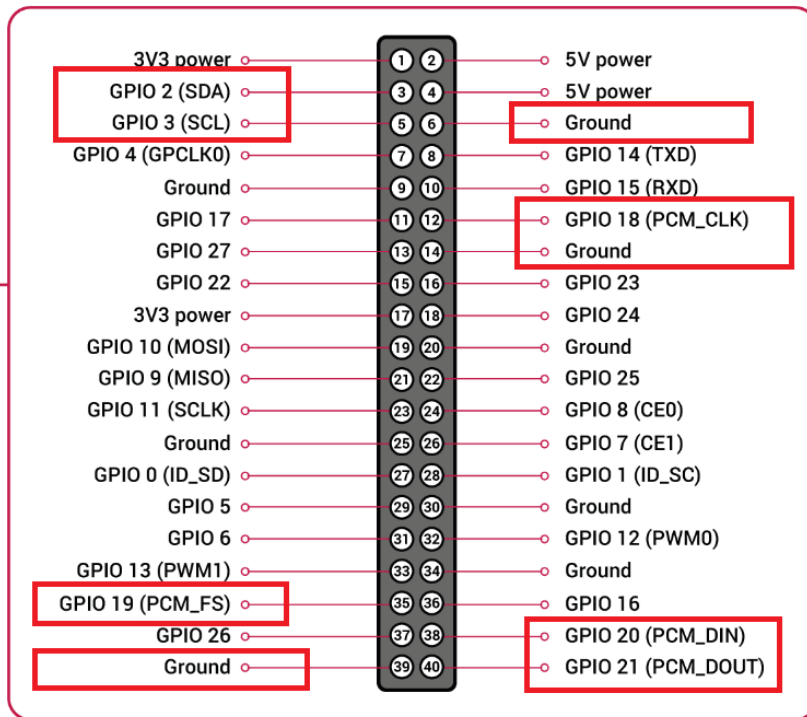
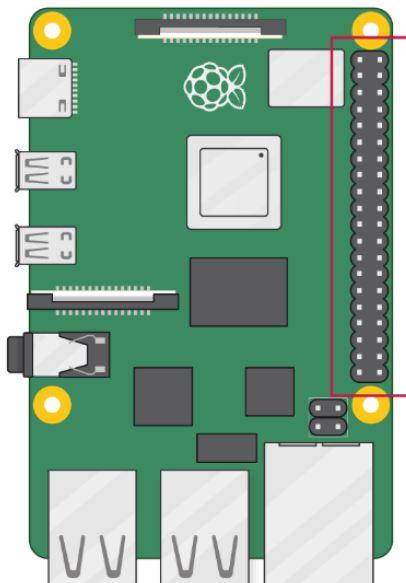
- Plug in the SD card and boot up Raspberry Pi
- Use below command in Raspberry Pi's terminal to check whether the loading is successful

uname -a

Linux driver integration as module

– Based on Raspberry Pi 4B

Hardware connection



Compile the driver to .ko file

- Will need below files in a separate folder

```
root@lpa-VB:/home/lpa/pi/sound/sound/soc/codecs# ls *.c *.h Makefile -l
-rw-r--r-- 1 root root 305 1月 10 15:29 Makefile
-rwxr-xr-x 1 root root 49374 1月 17 15:04 tac5x1x.c
-rwxr-xr-x 1 root root 9839 1月 17 15:04 tac5x1x.h
-rwxr-xr-x 1 root root 1840 1月 17 15:04 tac5x1x-i2c.c
-rwxr-xr-x 1 root root 1911 1月 17 15:04 tac5x1x-spi.c
```

- After execute make command, generate a .ko file

```
root@lpa-VB:/home/lpa/pi/sound/sound/soc/codecs# make clean
rm -f *.ko *.o *.mod.o *.symvers *.cmd *.mod.c *.order
root@lpa-VB:/home/lpa/pi/sound/sound/soc/codecs# make
make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- -C /home/lpa/pi/linux M=/home/lpa/pi/sou
nd/sound/soc/codecs modules
make[1]: Entering directory '/home/lpa/pi/linux'
CC [M] /home/lpa/pi/sound/sound/soc/codecs/tac5x1x.o
CC [M] /home/lpa/pi/sound/sound/soc/codecs/tac5x1x-i2c.o
LD [M] /home/lpa/pi/sound/sound/soc/codecs/snd-soc-tac5x1x.o
MODPOST /home/lpa/pi/sound/sound/soc/codecs/Module.symvers
CC [M] /home/lpa/pi/sound/sound/soc/codecs/snd-soc-tac5x1x.mod.o
LD [M] /home/lpa/pi/sound/sound/soc/codecs/snd-soc-tac5x1x.ko
make[1]: Leaving directory '/home/lpa/pi/linux'
```

Compile the dtb & dtbo

- Modify the *bcm2711-rpi-4-b.dts*, add below contents

```
i2s_clk_producer: &i2s {  
    pinctrl-names = "default";  
    pinctrl-0 = <&i2s_pins>;  
    #sound-dai-cells = <0>;  
    dmas = <&dma 2>, <&dma 3>;  
    dma-names = "tx", "rx";  
};
```

- Create *tac5x1x-soundcard-overlay.dts*, which including DTS containing sound card and i2c node



tac5x1x-soundcard-overlay.dts

Update the dtb & dtbo

- Copy new dts file into corresponding folder

```
# cp arm_broadcom/bcm2711-rpi-4-b.dts $(kernel)/arch/arm/boot/dts/broadcom/  
# cp tac5x1x-soundcard-overlay.dts $(kernel)/arch/arm64/boot/dts/overlays/
```

- Modify the \$(kernel)/arch/arm64/boot/dts/overlays/Makefile

Add new dtbo name at the tail of dtbo list

```
310         w1-gpio-pullup.dtbo \
311         w5500.dtbo \
312         watterott-display.dtbo \
313         waveshare-can-fd-hat-mode-a.dtbo \
314         waveshare-can-fd-hat-mode-b.dtbo \
315         wittypi.dtbo \
316         wm8960-soundcard.dtbo \
317         tac5x1x-soundcard.dtbo \
318         tlv320aic3110-soundcard.dtbo
319
320 targets += dtbs dtbs_install
321 targets += $(dtbo-y)
```

- Go back to kernel root path, run make to compile dtbs

```
# make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- dtbs
```

- Update *.dtb in SD card

```
#cp arch/arm64/boot/dts/broadcom/*.dtb mnt/fat32/
```

(This step only needs to be done once, it depends on the platform)

- Check the *.dtbo file in overlays

```
root@lpa-VB: /home/lpa/pi/linux/arch/arm64/boot/dts/overlays# ls tac5x1x*.dtbo -l
-rw-r--r-- 1 root root 2338  1月 11 19:57 tac5x1x-soundcard.dtbo
```

Prepare environment in platform

- After generating *.dtbo file and *.ko file, move both files into Pi's local folder, like /home/(username)/, any new update only changes these two files
 - tac5x1x-soundcard.dtbo
 - snd-soc-tac5x1x.ko
- check the depends of *.ko (as right image), if any of depends is lacked, then use modprobe command to install it.
Example:
 - # sudo modprobe regmap-i2c

```
root@lpa-VB:/home/lpa/pi/sound/sound/soc/codecs# modinfo snd-soc-tac5x1x.ko
filename:      /home/lpa/pi/sound/sound/soc/codecs/snd-soc-tac5x1x.ko
license:      GPL
author:       Kevin Lu <kevin-lu@ti.com>
description:  ASoC tac5x1x codec driver
license:      GPL
author:       Kevin Lu <kevin-lu@ti.com>
description:  ASoC tac5x1x codec driver
srcversion:   FD4FD8E4A4376CB6E773BF7
alias:        i2c:tac5x1x
alias:        i2c:tac5412
alias:        i2c:tac5x1x
alias:        of:N*T*Cti,tac5311C*
alias:        of:N*T*Cti,tac5311
alias:        of:N*T*Cti,tac5312C*
alias:        of:N*T*Cti,tac5312
alias:        of:N*T*Cti,tac5411C*
alias:        of:N*T*Cti,tac5411
alias:        of:N*T*Cti,tac5412C*
alias:        of:N*T*Cti,tac5412
alias:        of:N*T*Cti,tac5111C*
alias:        of:N*T*Cti,tac5111
alias:        of:N*T*Cti,tac5211C*
alias:        of:N*T*Cti,tac5211
alias:        of:N*T*Cti,tac5112C*
alias:        of:N*T*Cti,tac5112
alias:        of:N*T*Cti,tac5212C*
alias:        of:N*T*Cti,tac5212
alias:        of:N*T*Cti,tac5x1xC*
alias:        of:N*T*Cti,tac5x1x
depends:       snd-soc-core,snd-pcm,regmap-i2c,snd
name:         snd_soc_tac5x1x
vermagic:     6.7.0-rc6-v8+ SMP preempt mod_unload modversions aarch64
```

Install dtbo

- Create an overlay sub folder:

```
# mkdir -p /sys/kernel/config/device-tree/overlays/tac5x1x
```

- Copy *.dtbo into sys folder as dtbo

```
# cp /home/lpa/tac5x1x-soundcard.dtbo /sys/kernel/config/device-tree/overlays/tac5x1x/dtbo
```

- Utilize a shell script for quickly updating

```
lpa@raspberrypi:~ $ cat update_dts.sh
#!/bin/bash
cd /sys/kernel/config/device-tree/overlays/
mkdir -p tac5x1x
cp /home/lpa/tac5x1x-soundcard.dtbo ./tac5x1x/dtbo
```

- After updating dtbo, check the device tree nodes

```
lpa@raspberrypi:~ $ ls /proc/device-tree/soc/sound/
compatible  simple-audio-card,bitclock-master  simple-audio-card,format  status
name        simple-audio-card,codec            simple-audio-card,frame-master
phandle     simple-audio-card,cpu             simple-audio-card,name
lpa@raspberrypi:~ $ cat /proc/device-tree/soc/sound/simple-audio-card,name
ti,tac5x1x-soundcardlpa@raspberrypi:~ $
```

insmod *.ko file

- Make sure all depends are prepared, then run insmod command
 - # `sudo insmod snd-soc-tac5x1x.ko`
- Check installed modules
 - # `lsmod |grep tac`

```
lpa@raspberrypi:~$ lsmod |grep tac
snd_soc_tac5x1x          57344  1
regmap_i2c              12288  1 snd_soc_tac5x1x
snd_soc_core            303104  6 snd_soc_bcm2835_i2s,vc4,snd_soc_hdmi_codec,snd_soc_tac5x1x,snd_soc_simple_c
ard_utils,snd_soc_simple_card
snd_pcm                 147456  8 snd_soc_bcm2835_i2s,snd_bcm2835,snd_soc_hdmi_codec,snd_compress,snd_soc_tac
5x1x,snd_soc_simple_card_utils,snd_soc_core,snd_pcm_dmaengine
snd                     118784  15 snd_seq,snd_seq_device,snd_bcm2835,snd_soc_hdmi_codec,snd_timer,snd_compre
ss,snd_soc_tac5x1x,snd_soc_core,snd_pcm
```

- Remove modules (if needed)
 - #`sudo rmmod -f snd_soc_tac5x1x`

Confirm new sound card

- Use below commands to confirm sound card status

- #cat /proc/asound/pcm
- #cat /proc/asound/cards
- #alsamixer -c (card number)
- #amixer contents -c (card number)
- #aplay -L
- #arecord -L

```
lpa@raspberrypi:~$ cat /proc/asound/pcm
00-00: MAI PCM i2s-hifi-0 : MAI PCM i2s-hifi-0 : playback 1
01-00: MAI PCM i2s-hifi-0 : MAI PCM i2s-hifi-0 : playback 1
02-00: fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0 : fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0 : playback 1 : capture 1

lpa@raspberrypi:~$ cat /proc/asound/cards
0 [vc4hdmi0      ]: vc4-hdmi - vc4-hdmi-0
                  vc4-hdmi-0
1 [vc4hdmi1      ]: vc4-hdmi - vc4-hdmi-1
                  vc4-hdmi-1
2 [titac5x1xsoundc]: simple-card - ti,tac5x1x-soundcard
                  ti,tac5x1x-soundcard
```

```
hw:CARD=titac5x1xsoundc,DEV=0
ti,tac5x1x-soundcard, fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0
Direct hardware device without any conversions
plughw:CARD=titac5x1xsoundc,DEV=0
ti,tac5x1x-soundcard, fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0
Hardware device with all software conversions
sysdefault:CARD=titac5x1xsoundc
ti,tac5x1x-soundcard, fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0
Default Audio Device
dmix:CARD=titac5x1xsoundc,DEV=0
ti,tac5x1x-soundcard, fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0
Direct sample mixing device
usbstream:CARD=titac5x1xsoundc
ti,tac5x1x-soundcard
USB Stream Output
```

```
hw:CARD=titac5x1xsoundc,DEV=0
ti,tac5x1x-soundcard, fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0
Direct hardware device without any conversions
plughw:CARD=titac5x1xsoundc,DEV=0
ti,tac5x1x-soundcard, fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0
Hardware device with all software conversions
sysdefault:CARD=titac5x1xsoundc
ti,tac5x1x-soundcard, fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0
Default Audio Device
dsnoop:CARD=titac5x1xsoundc,DEV=0
ti,tac5x1x-soundcard, fe203000.i2s-tac5x1x-hifi tac5x1x-hifi-0
Direct sample snooping device
usbstream:CARD=titac5x1xsoundc
ti,tac5x1x-soundcard
USB Stream Output
```

TI Confidential – NDA Restrictions

Test Commands | Playback

- 16-bit mono TDM
 - #amixer -c [card id] cset numid=[id] [value]
 - #aplay **-c 1 -f S16_LE** -r 48000 -d 15 --device="hw:0,0" TDM-1slot-16b-play.wav
- 16-bit 2-slots TDM
 - #amixer -c [card id] cset numid=[id] [value]
 - #aplay **-c 2 -f S16_LE** -r 48000 -d 15 --device="hw:0,0" TDM-2slot-16b-play.wav

Test Commands | Recording

- 16-bit mono TDM
 - #amixer -c [card id] cset numid=[id] [Value]
 - #arecord **-c 1 -f S16_LE** -r 48000 -d 15 --device="hw:0,0" TDM-1slot-16b-record.wav
- 16-bit 2-slots TDM
 - #amixer -c [card id] cset numid=[id] [value]
 - #arecord **-c 2 -f S16_LE** -r 48000 -d 15 --device="hw:0,0" TDM-2slot-16b-record.wav

Thank You