



(<https://sel4.systems>)

**Getting Started** (/GettingStarted)

**Contributing** (/processes)

**Projects** (/projects)

**Tutorials** (/Tutorials)

Search sel4.systems



**seL4 Docs** (/) / **Supported Platforms** (/Hardware/)

/ Raspberry PI 3 Model B and Model B+

Current versions: **seL4-12.1.0** (/releases/sel4/12.1.0)

**camkes-3.10.0** (/releases/camkes/camkes-3.10.0)

**CapDL-0.2.1** (/releases/capdl/0.2.1)


**Announcing new releases:** 

# Raspberry PI 3 Model B and Model B+


This is an experimental port and is not guaranteed to work.

## Hardware

RPI-3B-V1\_2-SCHEMATIC-REDUCED.pdf 

Feature	Part	Datasheet
USB+ETH	SMSC LAN9514	9514.pdf 
SoC	BCM2837	

## Running seL4 on the Raspberry Pi 3B/3B+

Please also find a feature article going on seL4 the Raspberry Pi 3 on our blog  (and check out the rest of the blog too!)

## Serial connection

Serial TX and RX are located at GPIO pins 14 and 15 respectively (See Hardware schematic above)

## Reset

Short the jumper marked “RUN” between the expansion header and USB sockets to perform a soft reset of the CPU.

## U-Boot

Right now, the default U-Boot will not successfully boot an seL4 image on the RPi3, because of cache configuration issues in the seL4 ELFLoader. This problem can be remedied by having U-boot disable caches before loading seL4. Unfortunately, the stock upstream U-boot used to disable caches before loading the kernel image, but as of this patch (<https://github.com/u-boot/u-boot/commit/995eab8b5b580b67394312b1621c60a71042cd18>), U-boot no longer disables caches.

In order to obtain a U-boot binary that disables caching, you can either compile U-Boot from source yourself, or you can **use this prebuilt U-Boot binary image which already works**.

If you choose to build your own U-boot, then clone U-boot from upstream, then revert commit 995eab8b5b580b67394312b1621c60a71042cd18, and then build it:

```
git clone https://github.com/u-boot/u-boot.git u-boot
cd u-boot
git revert 995eab8b5b580b67394312b1621c60a71042cd18
make CROSS_COMPILE=arm-linux-gnueabi- rpi_3_32b_defconfig
make CROSS_COMPILE=arm-linux-gnueabi-
```

This will enable you to get the most up-to-date U-boot which will boot seL4 on the RPi3 successfully.

**NOTE:** Automatic revert is not going to work if you are using the latest version of u-boot. You need to manually revert the change by looking at the changeset.

Building u-boot using configuration `rpi_3_3b_defconfig` is going to produce an image that can boot seL4 on the Raspberry PI Model 3B and 3B+. Unfortunately for the 3B+ tftp boot is not going to work using this version of u-boot (the `tftp` command will report error `No Ethernet found`). In this case you need to build an image specific for the 3B+. Unfortunately the provided defconfig file for the 3B+ model (`rpi_3_b_plus_defconfig`) doesn't appear to build correctly.

In order to use tftp boot on a Raspberry PI Model 3B+, use the defconfig for the 3B (as described above), then manually change the `CONFIG_DEFAULT_DEVICE_TREE` parameter in the `.config` from:

```
CONFIG_DEFAULT_DEVICE_TREE="bcm2837-rpi-3-b"
```

to:





```
CONFIG_DEFAULT_DEVICE_TREE="bcm2837-rpi-3-b-plus"
```

Now the generated image should be able to use the on-board Ethernet device for TFTP.

## SD card setup

The PI boots from the first FAT32 partition on the SD card. Where files are specified, they should be located in the root directory of this partition.

Stage	Filename	Description	Source
-------	----------	-------------	--------

Stage	Filename	Description	Source
FSBL	-	Mounts SD and loads SSBL	ROM
SSBL	bootcode.bin	Loads GPU firmware and boots GPU	<a href="https://github.com/raspberrypi/firmware/tree/master/boot">https://github.com/raspberrypi/firmware/tree/master/boot</a> 
GPU firmware	start.elf or recovery.elf	Loads CPU bootloader and boots CPU	<a href="https://github.com/raspberrypi/firmware/tree/master/boot">https://github.com/raspberrypi/firmware/tree/master/boot</a> 
Usually the Linux kernel, but could also be u-boot	u-boot.bin	u-boot	You can either use our prebuilt U-boot which works  , or compile your own using the instructions above
	config.txt	u-boot parameters	Add enable_uart=1 and kernel=u-boot.bin to the bottom of config.txt (Sample: <a href="http://codepad.org/ykVYFSyP">http://codepad.org/ykVYFSyP</a>  )
	uboot.env	u-boot saved environment	Generated by u-boot (default environment) bootcmd copied to bootcmd_orig bootcmd and bootdelay removed

## Getting seL4 onto your Raspberry Pi 3

In this section you should find the two most convenient methods of getting a kernel image booted on your RPi3 described. All three methods assume that you have already built a kernel image using one of our projects—such as seL4Test:

Checkout the sel4test project using repo as per seL4Test (/seL4Test)

```
repo init -u https://github.com/seL4/sel4test-manifest.git
repo sync
mkdir cbuid
cd cbuid
../init-build.sh -DPLATFORM=rpi3 -DAARCH32=1
# The default cmake wrapper sets up a default configuration for the target platform.
# To change individual settings, run `ccmake` and change the configuration
# parameters to suit your needs.
ninja
```

Generated binaries can be found in the `images/` directory.

The two methods described here are using the SD card and TFTP.

Both methods require you to use an SD card to provide the RPi3 firmware with a U-boot bootloader it can load. The RPi3 doesn't store its bootloader in flash memory, so it looks for the bootloader on the SD card. Please see the section above for how to prepare your SD Card for the base set of requirements; you can glance at the table above to see the files you'll need and how to obtain them.

## SD Card

Get a micro-SD card ready by getting the base files copied onto it (See the previous section).

Following this, copy your seL4 image (such as an seL4test image), onto the SD card in its root directory - that's it. You can now remove the SD Card from your PC and re-insert it into the RPi3, and power the RPi3 on.

When the RPi3 boots up, be sure to interrupt the boot process and drop into the U-boot command prompt. From the U-boot command prompt, type something the following: `fatls mmc 0`. If you don't see your seL4 image file's name printed out, then you might need to double-check your steps so far to make sure you didn't forget something along the way. If you see your file listed, then do something like:

```
fatload mmc 0 0x10000000 sel4test-driver-image-arm-bcm2837
bootelf 0x10000000
```

## TFTP

Be sure you've set up the internal SD card with the required files for U-Boot and the RPi3 firmware (consult the table above). Then make certain that you've started up a TFTP server with your seL4 image being served by that TFTP server. From there:

```
usb start
dhcp
tftp 0x10000000 <YOUR_TFTP_SERVER_IP_ADDRESS>:sel4test-driver-image-arm-bcm2837
bootelf 0x10000000
```


If you are using a Raspberry PI Model 3B+, make sure you build the correct u-boot image as described above.

To use static IP instead of DHCP, use:

```
usb start
setenv ipaddr <RPI_IP_ADDRESS>
setenv serverip <YOUR_TFTP_SERVER_IP_ADDRESS>
tftp 0x10000000 sel4test-driver-image-arm-bcm2837
bootelf 0x10000000
```

## seL4 docs


 [sel4](#)


 [sel4proj](#)

This site is for displaying seL4 related documentation. Pull requests are welcome.

Site last updated: Thu Jan 12 15:17:30 2023 +1100 dc9a925

Page last updated: Thu Jan 12 15:17:30 2023 +1100 dc9a925

[View page on GitHub](#) 

[Edit page on GitHub](#) 

[Sitemap \(/sitemap\)](/sitemap)