To use the DAC over i2c, 2 things need to be configured:

* Install the kernel module from TI
* Create a device tree entry so the OS can see the DAC

To compile the kernel module, install the kernel source:

First install Git and the build dependencies:

sudo apt install git bc bison flex libssl-dev make

Next get the sources, which will take some time:

git clone --depth=1 https://github.com/raspberrypi/linux

Prepare the default configuration by running the following commands, depending on your Raspberry Pi model:

For Raspberry Pi 3, 3+, 4, 400 and Zero 2 W, and Raspberry Pi Compute Modules 3, 3+ and 4 default 64-bit build configuration

cd linux

KERNEL=kernel8

make bcm2711\_defconfig

In addition to your kernel configuration changes, you may wish to adjust the LOCALVERSION to ensure your new kernel does not receive the same version string as the upstream kernel. This both clarifies you are running your own kernel in the output of uname and ensures existing modules in /lib/modules are not overwritten.

To do so, change the following line in .config:

CONFIG\_LOCALVERSION="-v7l-MY\_CUSTOM\_KERNEL"

## Configuring the Kernel

The Linux kernel is highly configurable; advanced users may wish to modify the default configuration to customise it to their needs, such as enabling a new or experimental network protocol, or enabling support for new hardware.

Configuration is most commonly done through the make menuconfig interface. Alternatively, you can modify your .config file manually, but this can be more difficult for new users.

### Preparing to Configure

The menuconfig tool requires the ncurses development headers to compile properly. These can be installed with the following command:

sudo apt install libncurses5-dev

### Using menuconfig

Once you’ve got everything set up and ready to go, you can compile and run the menuconfig utility as follows:

make menuconfig

Building the kernel

For the 64-bit kernel:

make -j4 Image.gz modules dtbs

sudo make modules\_install

sudo cp arch/arm64/boot/dts/broadcom/\*.dtb /boot/

sudo cp arch/arm64/boot/dts/overlays/\*.dtb\* /boot/overlays/

sudo cp arch/arm64/boot/dts/overlays/README /boot/overlays/

sudo cp arch/arm64/boot/Image.gz /boot/$KERNEL.img