

Building a Yocto Image for RPi CM4 Integrating the Sona IF573 PCIe Wi-Fi Module



Purpose

This document provides step-by-step guidance on how to build a Linux image with Yocto release Scarthgap for the Raspberry Pi CM4 Compute Module including Ezurio's Sona IF573 PCIe Wi-Fi M2 card (Part Number [453-00120](#)).

Scope

This document concentrates on Wi-Fi functionality. Integrating Bluetooth functionality will be addressed in subsequent revisions.

Prerequisites

This document assumes that the user has basic knowledge about Linux and Yocto software and Raspberry Pi hardware.

Required Hardware

- **Ubuntu 22.04 PC with internet connection for cross-compiling the Yocto image**
- Raspberry Pi CM4 IO board with CM4 module
([Buy a Compute Module 4 IO Board – Raspberry Pi](#))



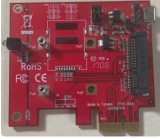
- **Sona IF573 DVK**
([453-00120-K1 | Ezurio](#))
[Sona™ IF573 - CYW55573 Module | Ezurio](#)

Note: The Sona-IF573 M2 Wi-Fi module requires a 32kHz SUSCLK signal on pin50 which is not supplied by the CM4. This DVK provides that signal to the module.



- 1xPCIe to miniPCIe adapter

[Buy Renkforce GBIC \[1x Mini PCI Express - 1x PCI Express\] | Conrad Electronic](#)



- miniPCIe to M2 E-key adapter

[Delock Products 63909 Delock Adapter Mini PCIe > M.2 Key E slot](#)



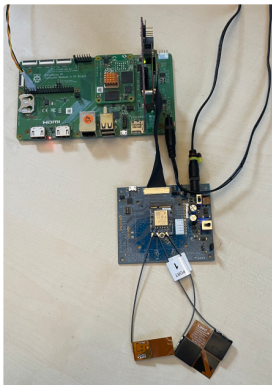
Note: both of the above adapters were used on top of each to connect the Sona-IF573 M2 card to the PCIe slot of the RPi CM4 IO board. There might be a single adapter solution in the market though.

- USB to serial converter to attach to the debug UART of the Raspberry Pi

For example: [USB - TTL / UART / RS232 Adapterkabel mit PL2303HX Chipsatz](#)



Complete hardware setup



Software Installation

Preparation of Yocto build environment and first build

Update build PC

```
1 sudo apt update
2 sudo apt install gawk wget git diffstat unzip texinfo gcc-multilib \
3 build-essential chrpath socat cpio python3 python3-pip python3-pexpect \
4 xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa \
5 libssl1.2-dev pylint xterm
```

Clone Yocto environment

```

1 mkdir -p ~/yocto/cm4
2 cd ~/yocto/cm4
3 git clone -b scarthgap git://git.yoctoproject.org/poky
4 cd poky
5 git clone -b scarthgap https://github.com/agherzan/meta-raspberrypi.git
6 git clone -b scarthgap git://git.openembedded.org/meta-openembedded

```

Configuration for first boot

Enter build environment

```

1 source oe-init-build-env

```

Add layers to conf/bblayers.conf by executing these commands in the build directory

```

1 bitbake-layers add-layer ../meta-openembedded/meta-oe
2 bitbake-layers add-layer ../meta-openembedded/meta-python
3 bitbake-layers add-layer ../meta-openembedded/meta-networking
4 bitbake-layers add-layer ../meta-raspberrypi

```

Set machine and options by editing conf/local.conf

At the end of local.conf add:

```

1 # added for CM4 build
2 MACHINE = "raspberrypi4-64"
3 ENABLE_UART = "1"
4 ENABLE_SPI_BUS = "1"
5 ENABLE_I2C = "1"
6 IMAGE_INSTALL:append = " python3"

```

Build image

```

1 bitbake core-image-minimal

```

Note: This will take quite a while. Fetch errors can often be overcome by simply repeating the bitbake process.

Successful build:

```

Sstate summary: Wanted 1636 Local 0 Mirrors 0 Missed 1636 Current 112 (0% match, 0% complete)##### | ETA: 0:00:00
Initialising tasks: 100% |#####| Time: 0:00:04
NOTE: Executing Tasks
NOTE: Task Summary: Attempted 3715 tasks of which 279 didn't need to be rerun and all succeeded.
alex@bigNuc:~/yocto/cm4/poky/build$

```

Integrating the Sona IF573

Placing Ezurio meta-layer

```

1 cd ..
2 git clone https://github.com/Ezurio/meta-summit-radio/
3 cd meta-summit-radio
4 git checkout lrd-12.103.0.x
5 cd ..
6 source oe-init-build-env

```



Note: at the time of using this document newer releases might be available. Feel free to use the latest: [Releases](#)

[es · Ezurio/Connectivity_Stack_Release_Packages](#)

Adding Ezurio meta-layer to conf/bblayers.conf

execute this command in the build directory:

```
1 bitbake-layers add-layer ../meta-summit-radio/meta-summit-radio
```

Patches

Two patches are required for running the Sona-IF573 on the Raspberry Pi CM4:

1. 0001-PATCH-PCIe-Bring-up-for-Rpi-CM4.patch
2. 0002-PATCH-brcmcfmac-Set-PCI-dma-mask-to-64-bits.patch

 [RPI CM4 patches | Ezurio](#)

Download the patches and extract them to:

```
1 ../poky/meta-summit-radio/meta-summit-radio/recipes-bsp/summit-backports/files
```

Modify to add the patches:

```
1 ../poky/meta-summit-radio/meta-summit-radio/recipes-bsp/summit-backports/summit-backports.inc
```

The following section must be modified as shown in the second code block:

```
1 SRC_URI += "\
2     ${SUMMIT_URI}/summit-backports-${PV}.tar.bz2;name=summit-backports \
3     file://0001-fix-genlmsg_multicast_allns.patch \
4     file://0002-Fixed-UBSAN-out-of-bounds-errors.patch \
5     file://0003-brcmcfmac-Demote-DPC-error-message.patch \
6     file://0004-Fix-stop-shutdown-mode-entry.patch \
7     "
```

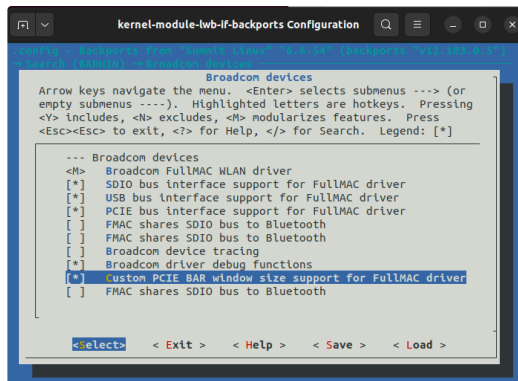
Edit it as shown below:

```
1 SRC_URI += "\
2     ${SUMMIT_URI}/summit-backports-${PV}.tar.bz2;name=summit-backports \
3     file://0001-fix-genlmsg_multicast_allns.patch \
4     file://0002-Fixed-UBSAN-out-of-bounds-errors.patch \
5     file://0003-brcmcfmac-Demote-DPC-error-message.patch \
6     file://0004-Fix-stop-shutdown-mode-entry.patch \
7     file://0001-PATCH-PCIe-Bring-up-for-Rpi-CM4.patch \
8     file://0002-PATCH-brcmcfmac-Set-PCI-dma-mask-to-64-bits.patch \
9     "
```

Configure Backports

```
1 bitbake -c menuconfig kernel-module-lwb-if-backports
```

enable Custom PCIE BAR window size support for FullMac driver



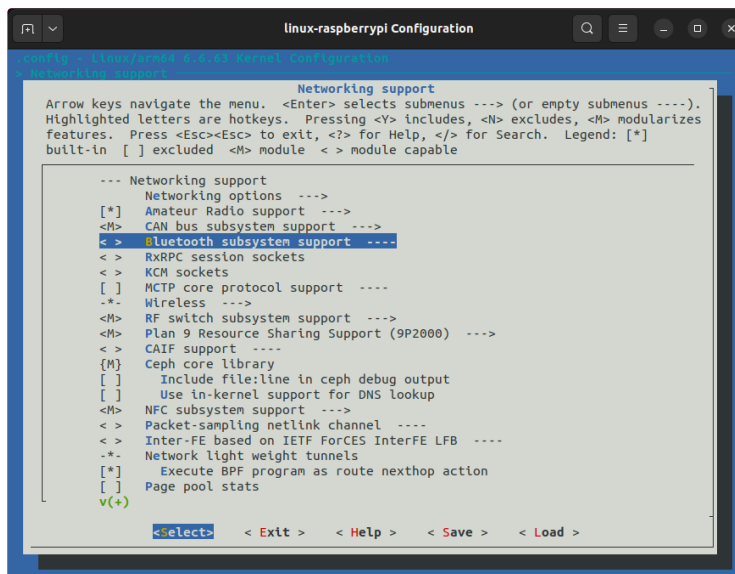
Navigate the menu as follows:

Wireless LAN → Broadcom devices → Custom PCIE BAR window size support for FullMAC driver

Configure Linux kernel

```
1 bitbake -c menuconfig virtual/kernel
```

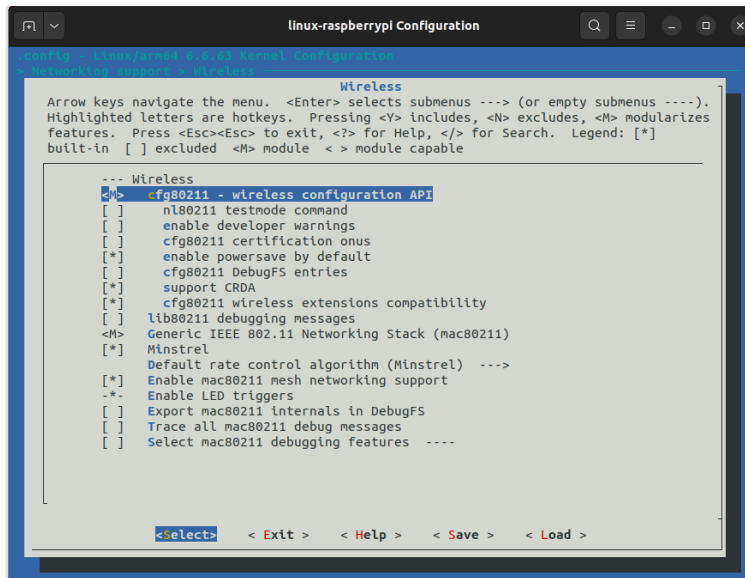
Choose **disable Bluetooth subsystem support** as it is put into the kernel by Backports.



Navigate the menu as follows:

Networking support → Bluetooth subsystem support ----

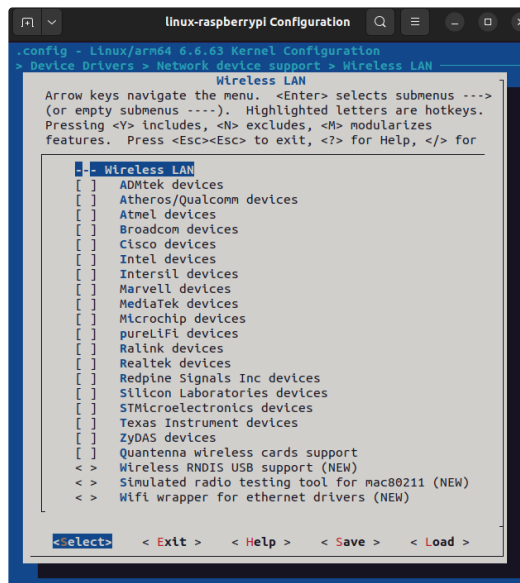
Next, **set Wireless <M>**



Navigate the menu as follows:

Networking support → Wireless → cfg80211 - wireless configuration API

remove other *Wireless LAN* drivers (but NOT the subsystem “Wireless LAN”)



Adding required Ezurio recipes to conf/local.conf

At the end of local.conf add:

```
1 # added for IF573
2 IMAGE_OVERHEAD_FACTOR = "6"
3
4 CORE_IMAGE_EXTRA_INSTALL += "\
5     packagegroup-tools-bluetooth \
6     pulseaudio \
7     hostapd \
8     iptables \
9     iperf3 \
```


Flashing Linux image to CM4 eMMC memory

Compiling usbboot to mount eMMC memory as USB drive

Make sure you have the `libusb` library installed:

```
sudo apt install libusb-1.0-0-dev
```

Download the Raspberry Pi usbboot repository:

```
1 $ git clone --depth=1 https://github.com/raspberrypi/usbboot
```

Change directory into the `usbboot` directory and build `rpiboot`:

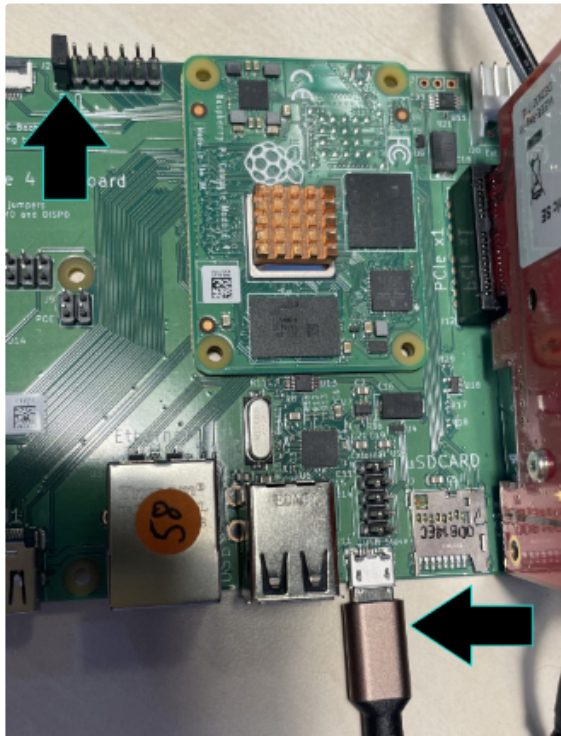
```
1 $ cd usbboot
2 $ make
```

Now there should be an `rpiboot` executable in the directory.

Flashing the CM4

For this the CM4 must be put into USB mass storage mode.

Place a jumper on J2 across pins 1 and 2 and connect the microUSB port of the IO board to your Ubuntu development PC.



To mount the eMMC storage, run:

```
1 $ sudo ./rpiboot
```

```
alex@bigNUC:~/usbboot$ sudo ./rpiboot
[sudo] password for alex:
RPiBOOT: build-date Dec 7 2023 version 20220815-145439 1e651a88
Waiting for BCM2835/6/7/2711...
Loading embedded: bootcode4.bin
Sending bootcode.bin
Successful read 4 bytes
Waiting for BCM2835/6/7/2711...
Loading embedded: bootcode4.bin
Second stage boot server
Loading embedded: start4.elf
File read: start4.elf
Second stage boot server done
```

Then, with lsblk you should see that your CM4 eMMC memory is mounted as a block device (sdb in this case):

```
alex@bigNUC:~/usbboot$ lsblk | grep sd
sda      8:0    0   1,9T  0 disk
├─sda1   8:1    0   512M  0 part /boot/efi
├─sda2   8:2    0   1,9T  0 part /
└─sdb    8:16   1   7,3G  0 disk
   ├─sdb1  8:17   1   130M  0 part /media/alex/boot
   └─sdb2  8:18   1   1,8G  0 part /media/alex/root
```

Locate the generated Linux image:

```
1 ../poky/build/tmp/deploy/images/raspberrypi4-64
2 ls *.wic.bz2 -al
3 -rw-r--r-- 2 alex alex 92657639 Jun 24 17:07 core-image-minimal-raspberrypi4-64.rootfs-20250624144522.wic.bz2
4 lrwxrwxrwx 2 alex alex          64 Jun 24 17:07 core-image-minimal-raspberrypi4-64.rootfs.wic.bz2 -> core-image-minimal-raspberrypi4-64.rootfs-20250624144522.wic.bz2
```

```
alex@bigNUC:~/yocto/cn4/poky/build/tmp/deploy/images/raspberrypi4-64$ ls *.wic.bz2 -al
-rw-r--r-- 2 alex alex 92657639 Jun 24 17:07 core-image-minimal-raspberrypi4-64.rootfs-20250624144522.wic.bz2
lrwxrwxrwx 2 alex alex          64 Jun 24 17:07 core-image-minimal-raspberrypi4-64.rootfs.wic.bz2 -> core-image-minimal-raspberrypi4-64.rootfs-20250624144522.wic.bz2
alex@bigNUC:~/yocto/cn4/poky/build/tmp/deploy/images/raspberrypi4-64$
```

Unpack it as follows:

```
1 bunzip2 -f core-image-minimal-raspberrypi4-64.rootfs.wic.bz2
```

```
alex@bigNUC:~/yocto/cn4/poky/build/tmp/deploy/images/raspberrypi4-64$ bunzip2 -f core-image-minimal-raspberrypi4-64.rootfs.wic.bz2
```

Use dd to flash the memory.

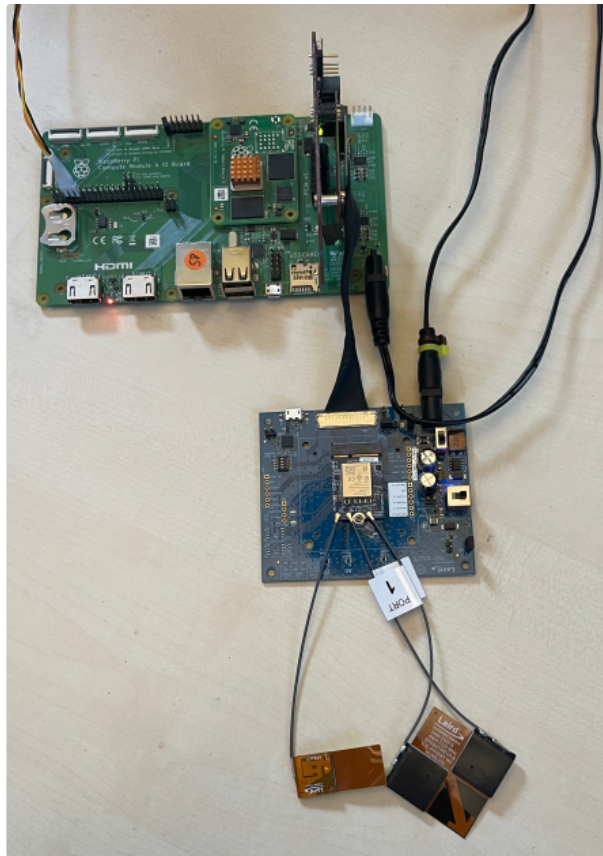
 Make sure to select the block device that was created for your Raspberry Pi!  

```
1 sudo dd if=core-image-minimal-raspberrypi4-64.rootfs.wic of=/dev/sdb bs=1M conv=sync
```

```
alex@bigNUC:~/yocto/cn4/poky/build/tmp/deploy/images/raspberrypi4-64$ sudo dd if=core-image-minimal-raspberrypi4-64.rootfs.wic of=/dev/sdb bs=1M conv=sync
[sudo] password for alex:
1956+0 records in
1956+0 records out
2051014656 bytes (2,1 GB, 1,9 GiB) copied, 499,832 s, 4,1 MB/s
alex@bigNUC:~/yocto/cn4/poky/build/tmp/deploy/images/raspberrypi4-64$ sync
```

After successful download of the image, carefully remove the jumper on J2 and the microUSB cable and remove power.

Set up hardware and apply power to both boards:



📌 Login username is root, no password required.

With `iw dev` you now should see that the Wi-Fi interface was created:

```
root@raspberrypi4-64:~# iw dev
phy#0
    Unnamed/non-netdev interface
        wdev 0x2
        addr ea:cb:f5:60:3b:32
        type P2P-device
        txpower 31.00 dBm
    Interface wlan0
        ifindex 3
        wdev 0x1
        addr e8:cb:f5:60:3b:32
        type managed
        channel 1 (5955 MHz), width: 20 MHz, center1: 5955 MHz
        txpower 31.00 dBm
root@raspberrypi4-64:~#
```