Broadcast Radio in AAOS

POC with the VIM3 reference board

C O N J U R E

INTERFACING THE FUTURE

Introducing Conjure…

Interfacing The Future

Conjure is a product-focused digital studio

We help ambitious brands and organisations differentiate their digital products and services through outstanding strategy, innovation, design and technical execution.

We engineer extraordinary digital experiences for a better future.

Some of our awards…

uk app awards

THE DRUM DADI AWARDS

THE F1® CONNECTIVITY INNOVATION PRIZE

CAR HMI UX REDEFINED

CV MAGAZINE WINNER CREATIVE AGENCY 2019

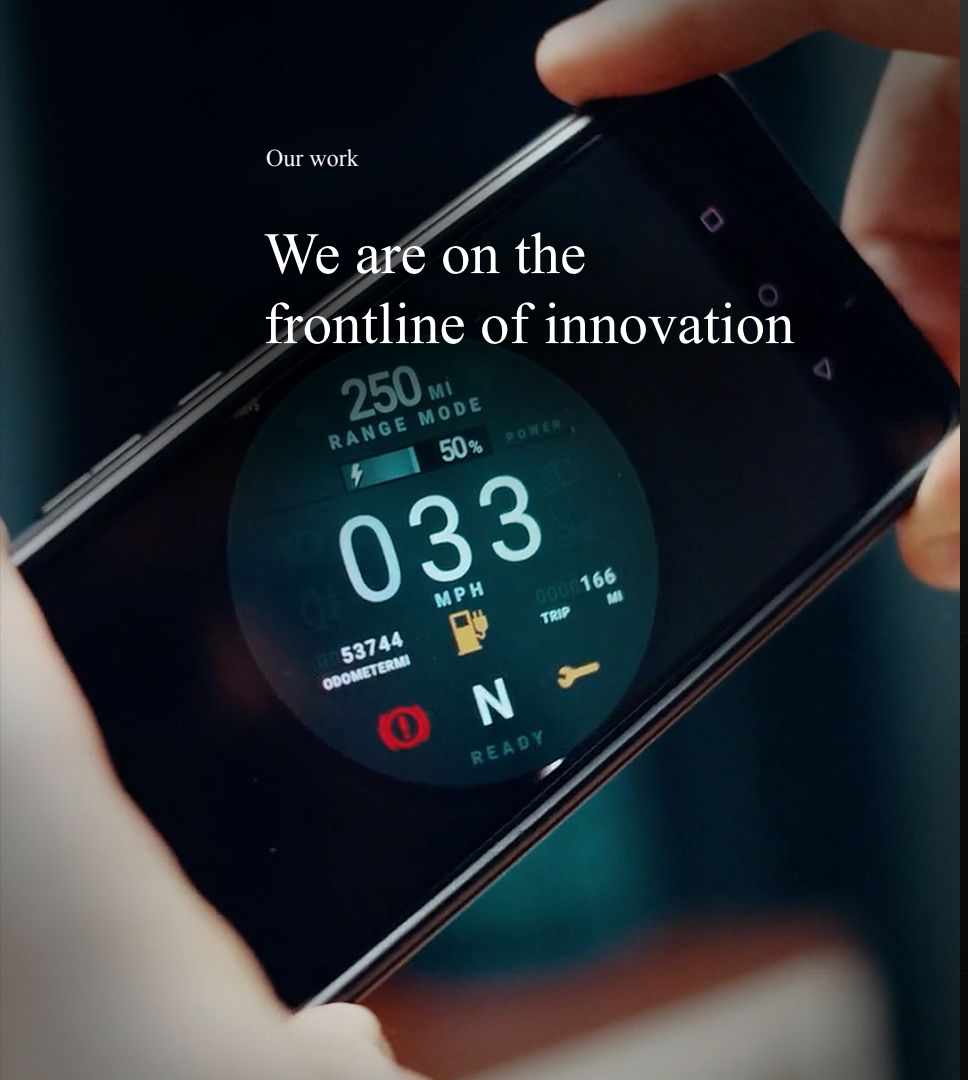
AURA

Powered by
Android Automotive

Automotive
Meets Android

Our work

We are on the frontline of innovation



GRUNDFOS

BIG BUS TOURS

LEGO

HOLM

SCIENCE MUSEUM

Transport for London

TRIUMPH

BENTLEY

McLaren AUTOMOTIVE

THE WILSON

GAIN CAPITAL

ir.deto

WHAT3WORDS

JAGUAR LAND ROVER

fsb Experts in Business

NHS

EPIC GAMES

Broadcast Radio in AAOS

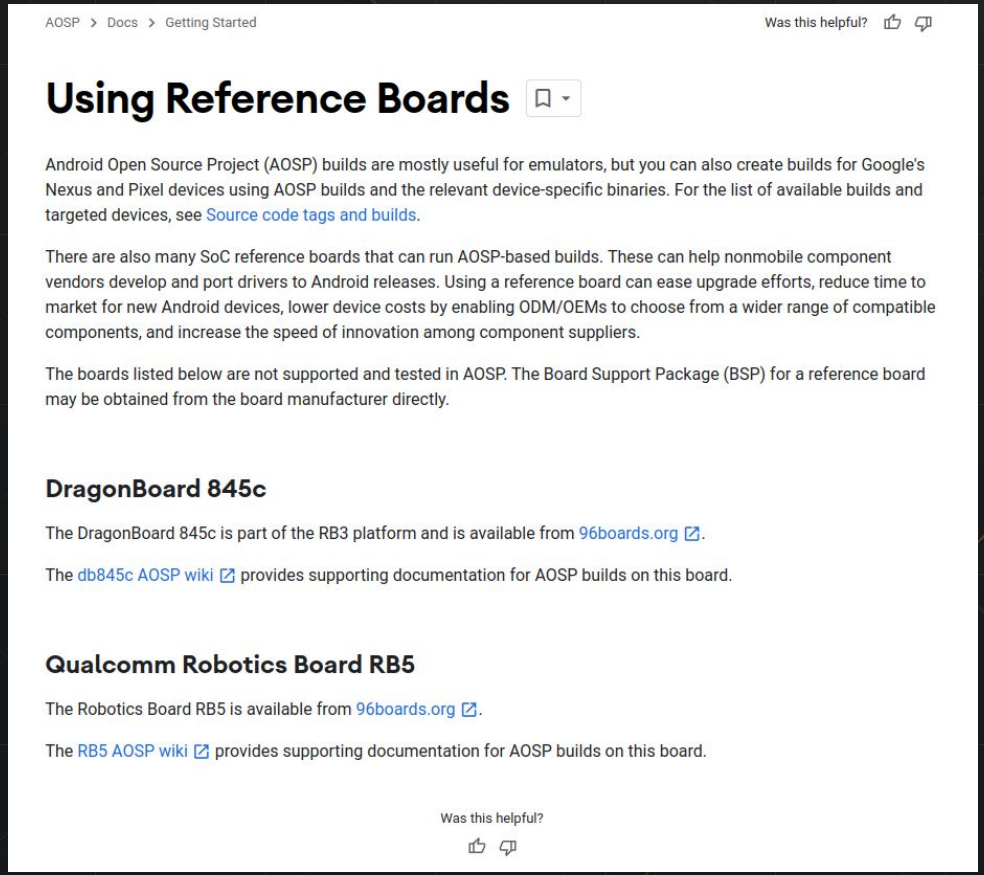
POC with the VIM3 reference board

C O N J U R E

INTERFACING THE FUTURE

VIM3

Reference Board(?)



AOSP > Docs > Getting Started

Was this helpful?

Using Reference Boards

Android Open Source Project (AOSP) builds are mostly useful for emulators, but you can also create builds for Google's Nexus and Pixel devices using AOSP builds and the relevant device-specific binaries. For the list of available builds and targeted devices, see Source code tags and builds.

There are also many SoC reference boards that can run AOSP-based builds. These can help nonmobile component vendors develop and port drivers to Android releases. Using a reference board can ease upgrade efforts, reduce time to market for new Android devices, lower device costs by enabling ODM/OEMs to choose from a wider range of compatible components, and increase the speed of innovation among component suppliers.

The boards listed below are not supported and tested in AOSP. The Board Support Package (BSP) for a reference board may be obtained from the board manufacturer directly.

DragonBoard 845c

The DragonBoard 845c is part of the RB3 platform and is available from 96boards.org.

The db845c AOSP wiki provides supporting documentation for AOSP builds on this board.

Qualcomm Robotics Board RB5

The Robotics Board RB5 is available from 96boards.org.

The RB5 AOSP wiki provides supporting documentation for AOSP builds on this board.

Was this helpful?

Broadcast Radio

What is Broadcast Radio?

1. Audio transmitted via radio waves
2. AM, FM, DAB, HD Radio,...
3. Additional data via RDS, DLS,...
4. Challenges
    - Real time, location-dependent, moving cars
    - Different standards across the world

Broadcast Radio
Let’s focus on FM Radio (in Europe)

1. 87.5 - 108.0 MHz (100 kHz spacing)
    - Scan & Tune -> 2x FM-Receiver
2. RDS - Radio Data System
    - PI - Program Identifier
    - PS - Program Service Name
    - Program Type, Clock, Frequencies, Radio Text, Traffic announcements,...

Hardware Setup

Hardware Setup
Current Status

1. Only a single FM tuner
2. Arduino scans for FM stations once after startup and then sends this list every 10s via CAN
3. No other features supported
4. Only the happy path implemented (no error handling on most layers)
5. It’s a POC :)

Hardware Setup

Decisions, decisions,...

1. CAN is common in Automotive
2. Android Automotive supports CAN
3. Linux SocketCAN - Network device can be accessed by multiple applications
4. Separation of concerns
    - Arduino can be updated easily
    - 1…x tuners
5. VIM3 has ready to go CAN configuration for Ubuntu

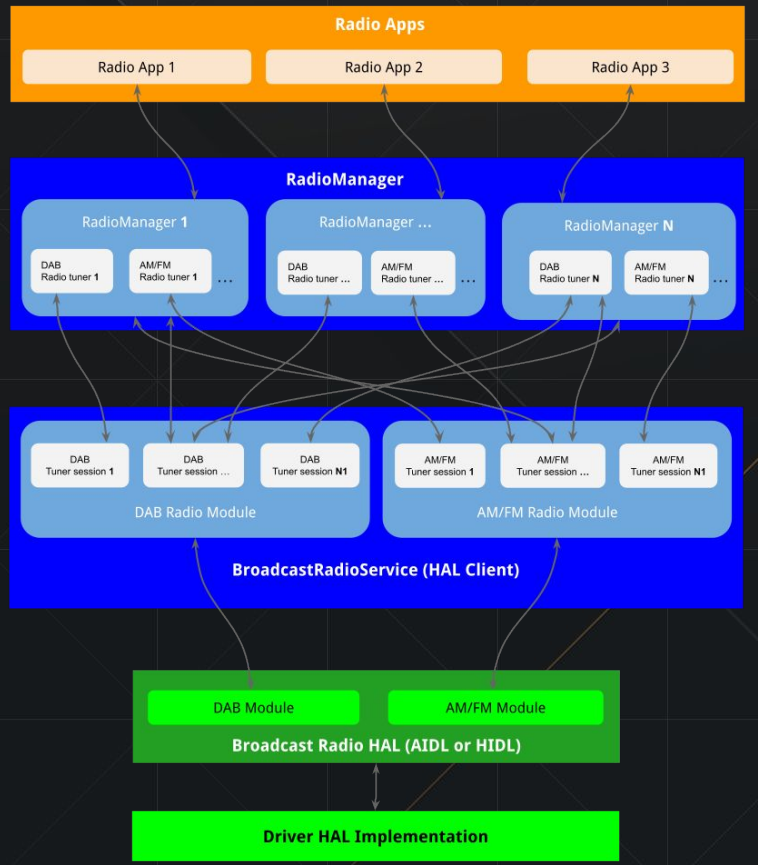
Software Setup
VIM3 running AA 12

1. System Components for Broadcast Radio
2. BroadcastRadioService
3. Challenges with VIM3

Software - VIM3 Android Automotive

1. Radio App (should) implement Media Session and Media Browser Service
2. RadioManager is the system service to access the Radio API

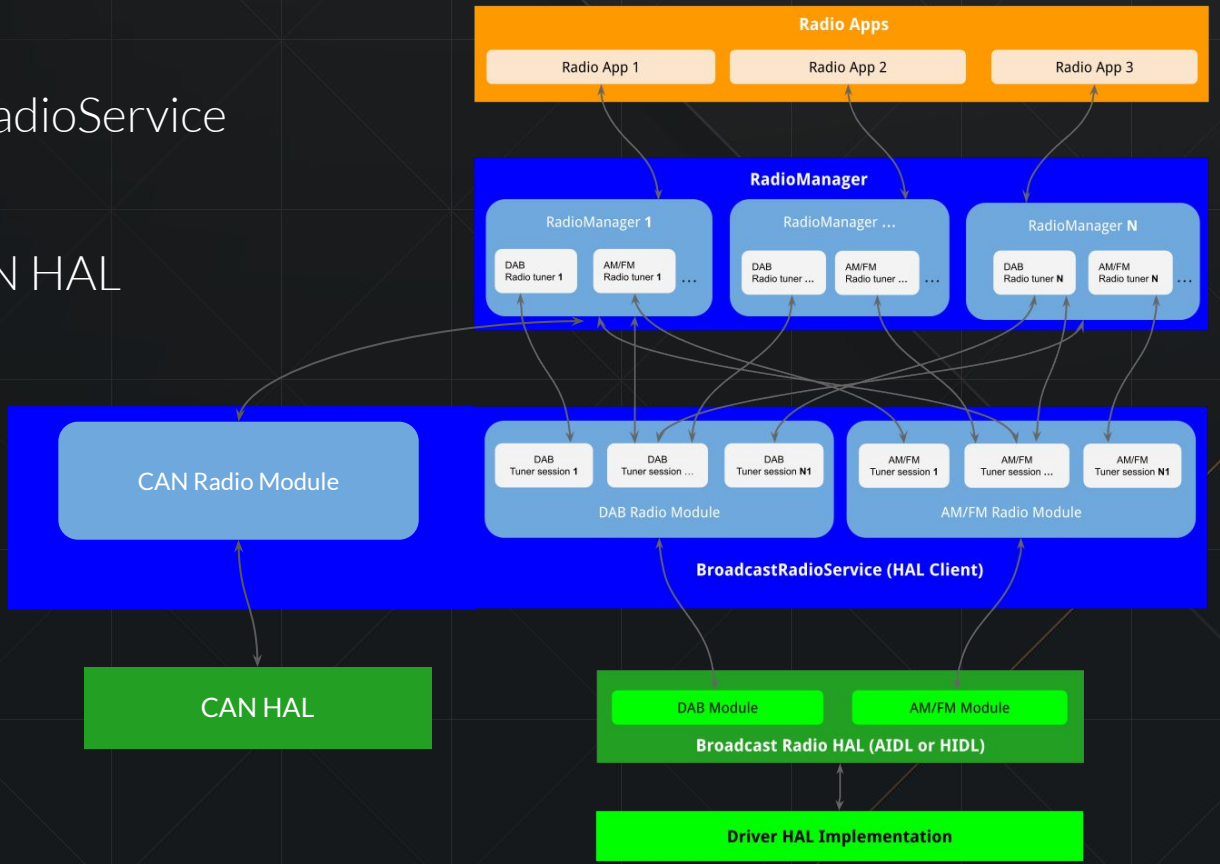
   `context.getSystemService(Context.RADIO_SERVICE)`
3. BroadcastRadioService system service started by SystemServer
4. HAL



Source: https://source.android.com/docs/devices/automotive/broadcast-radio

Software - VIM3 Android Automotive

1. Custom BroadcastRadioService implementation
2. Default Android CAN HAL

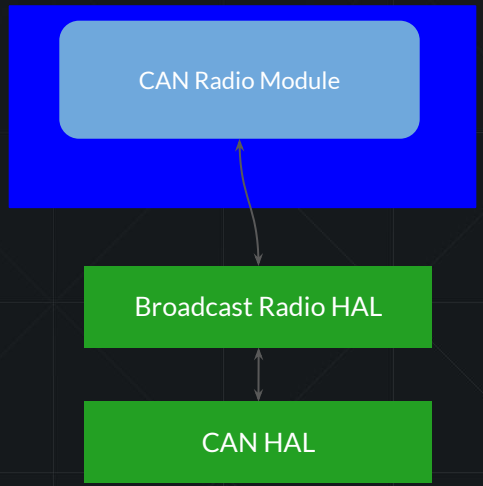


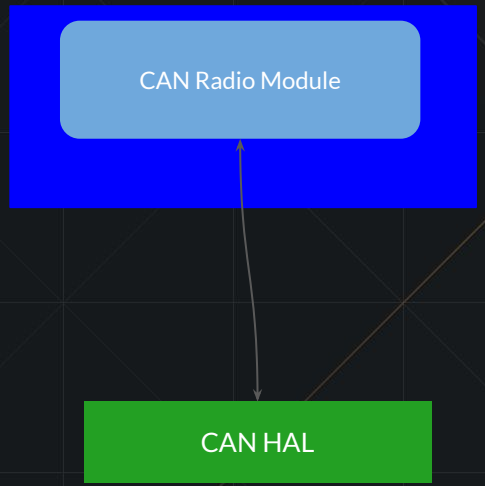
Source: https://source.android.com/docs/devices/automotive/broadcast-radio

HAL responsibilities?

1. BroadcastRadioHal
   - IBroadcastRadio.hal
   - ITunerSession.hal
   - ….

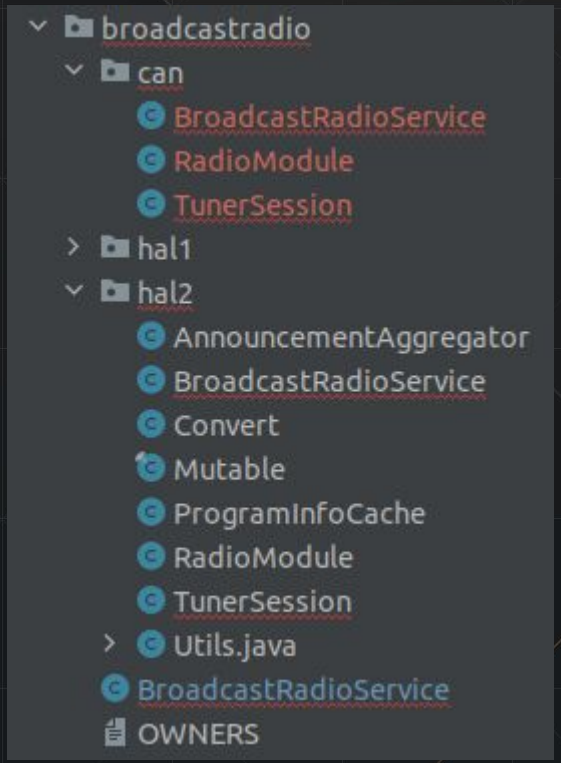
2. CAN BroadcastRadioService
   - Seek, scan, step, tune are shifted to the external Arduino Radio Module
   - SocketCAN implementation in Service





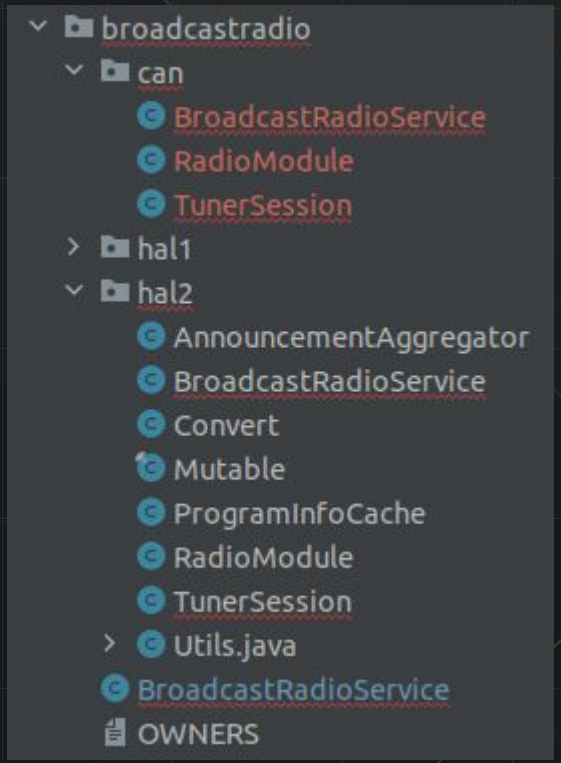
BroadcastRadioService

1. BroadcastRadioService.java
    - Extends SystemService
    - Loads the modules
    - Acts as proxy



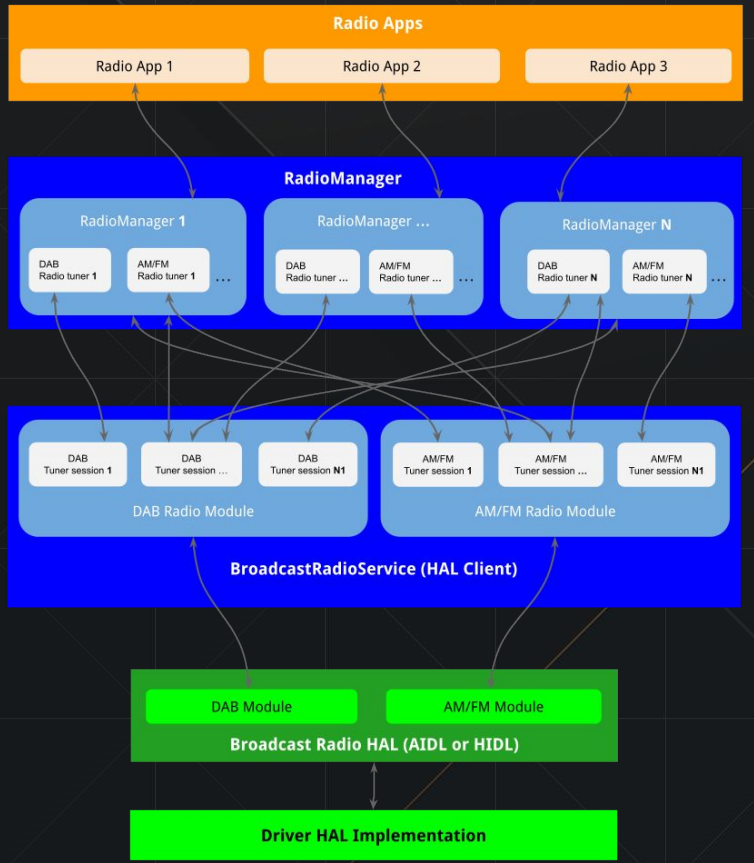
BroadcastRadioService

2. CAN service implementation
    - BroadcastRadioService
        - Listens for CAN service registration
        - Loads RadioModule
    - RadioModule
        - Provides information (RadioManager.ModuleProperties)
        - Opens TunerSession
        - Brings CAN interface up
    - TunerSession
        - Extends ITuner.stub
        - Communicates via CAN



Where to find what

1. Radio App
   /packages/apps/Car/Radio/
2. RadioManager
   /frameworks/base/core/java/android/hardware/radio/
3. BroadcastRadioService
   /frameworks/base/services/core/java/com/android/server/broadcastradio/
4. BroadcastRadio HAL
   /hardware/interfaces/broadcastradio/
5. CAN HAL
   /hardware/interfaces/automotive/can/



How to add the right packages

- Example can be found at /device/generic/car/common/car.mk
- Modify /device/amlogic/yukawa/yukawa.mk

```
$(call inherit-product, device/amlogic/yukawa/car/car.mk)
$(call inherit-product, device/amlogic/yukawa/car/can.mk)
```

- car.mk

```
PRODUCT_PACKAGES += \
    android.hardware.automotive.vehicle@2.0-service \
    android.hardware.broadcastradio@2.0-service \
    android.hardware.automotive.can@1.0-service \
```

```
$(call inherit-product, packages/services/Car/car_product/build/car.mk)
```

- can.mk https://github.com/linux-can/can-utils

Challenges with the VIM3

- Poor documentation
  - Documentation got migrated to a new system just when I started working with it
- AOSP Kernel has no device tree configuration for CAN on VIM3
- AOSP Kernel has no drivers for GPIO interrupts
- Ubuntu Kernel has (wrong) frequency hardcoded in mcp2515 driver
- Ubuntu Kernel uses custom device-tree property for GPIO interrupt (int-gpio) pin configuration

Kernel

- GPIO
  - pinctrl-meson.c
  - irq-meson.c
- CAN
  - mcp251x.c
- Device tree
- Kernel Configuration

Device Tree

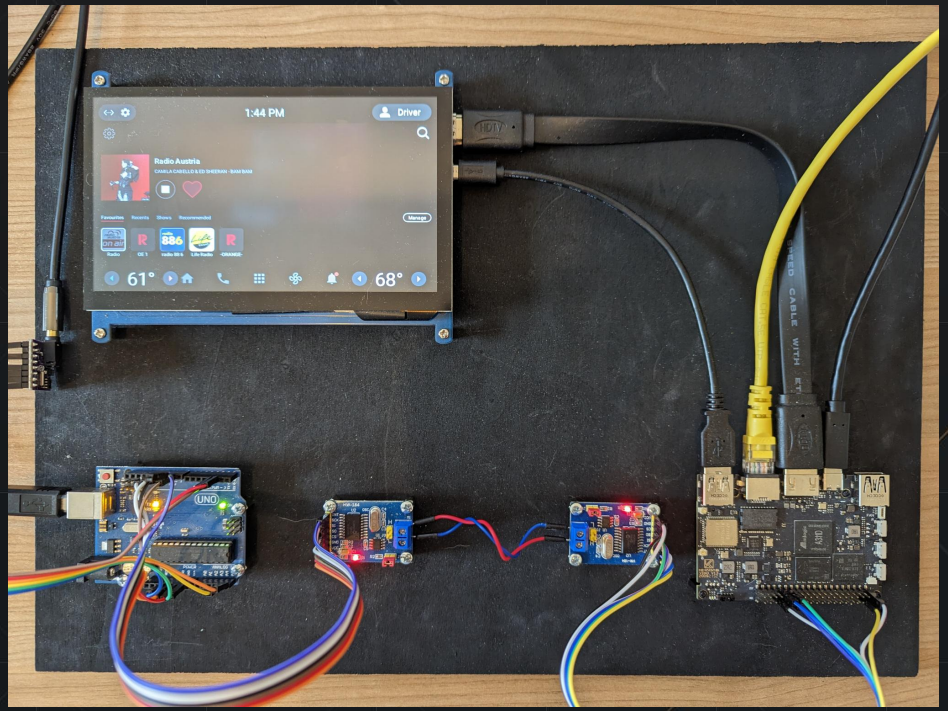
- meson-g12b-a311d-khadas-vim3-android.dts
- meson-g12-common.dtsi

- Disable i2c3 and uart_C
- Add can0 to spicc1 —>

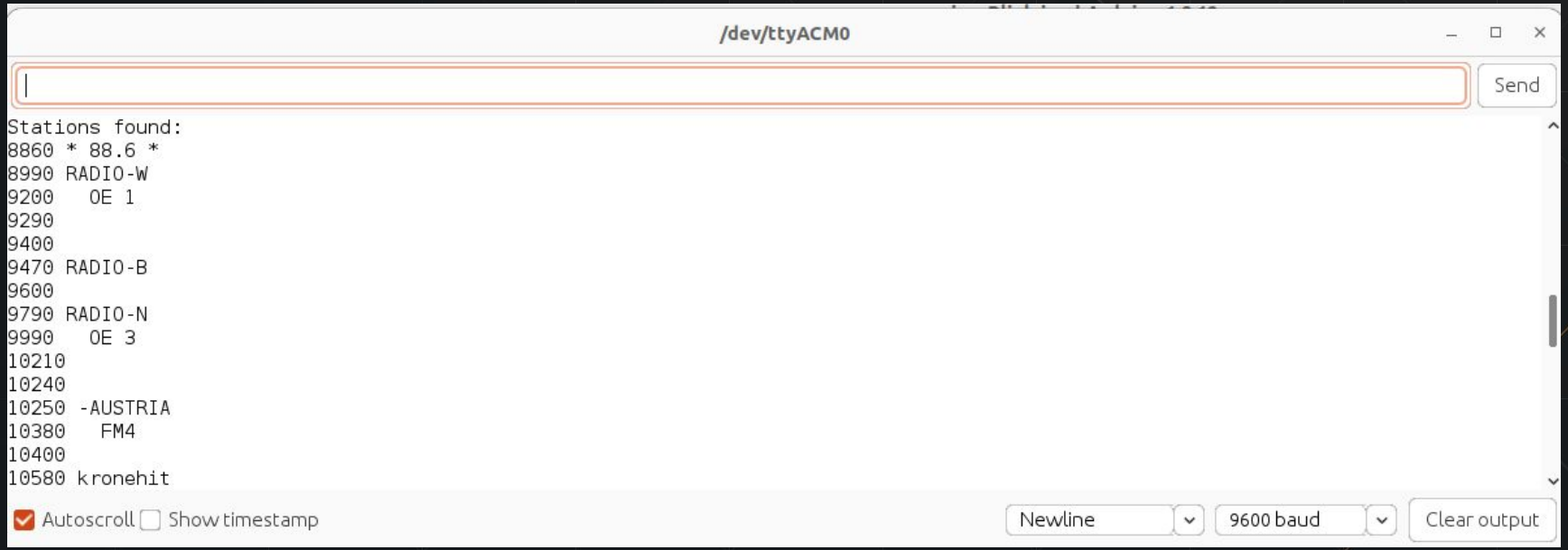
```
fragment@102 {
	target-path = "/";
	__overlay__
	{
	clk_can:clk_can {
			compatible = "fixed-clock";
			clock-frequency = <16000000>;
			#clock-cells = <0>;
		};
	};
};

fragment@103 {
	target = <&spicc1>;

	__overlay__ {
		status = "okay";
		pinctrl-names = "default";
		pinctrl-0 = <&spicc1_pins>;
		cs-gpios = <&gpio GPIOH_6 0>;
	can0: mcp2515@0 {
			label="mcp2515";
			clocks = <&clk_can>;

			spi-max-frequency = <2000000>;
			compatible = "microchip,mcp2515";
			reg = <0>;
			//IRQ_TYPE_EDGE_FALLING
			int-gpio = <&gpio GPIOA_15 GPIO_ACTIVE_LOW>;
			status = "okay";
			interrupt-parent = <&gpio>;
			interrupts = <GPIOA_15 GPIO_ACTIVE_LOW>;
		};
	};
```

Demo

Time to see it in Action



Arduino Serial Monitor



ifconfig

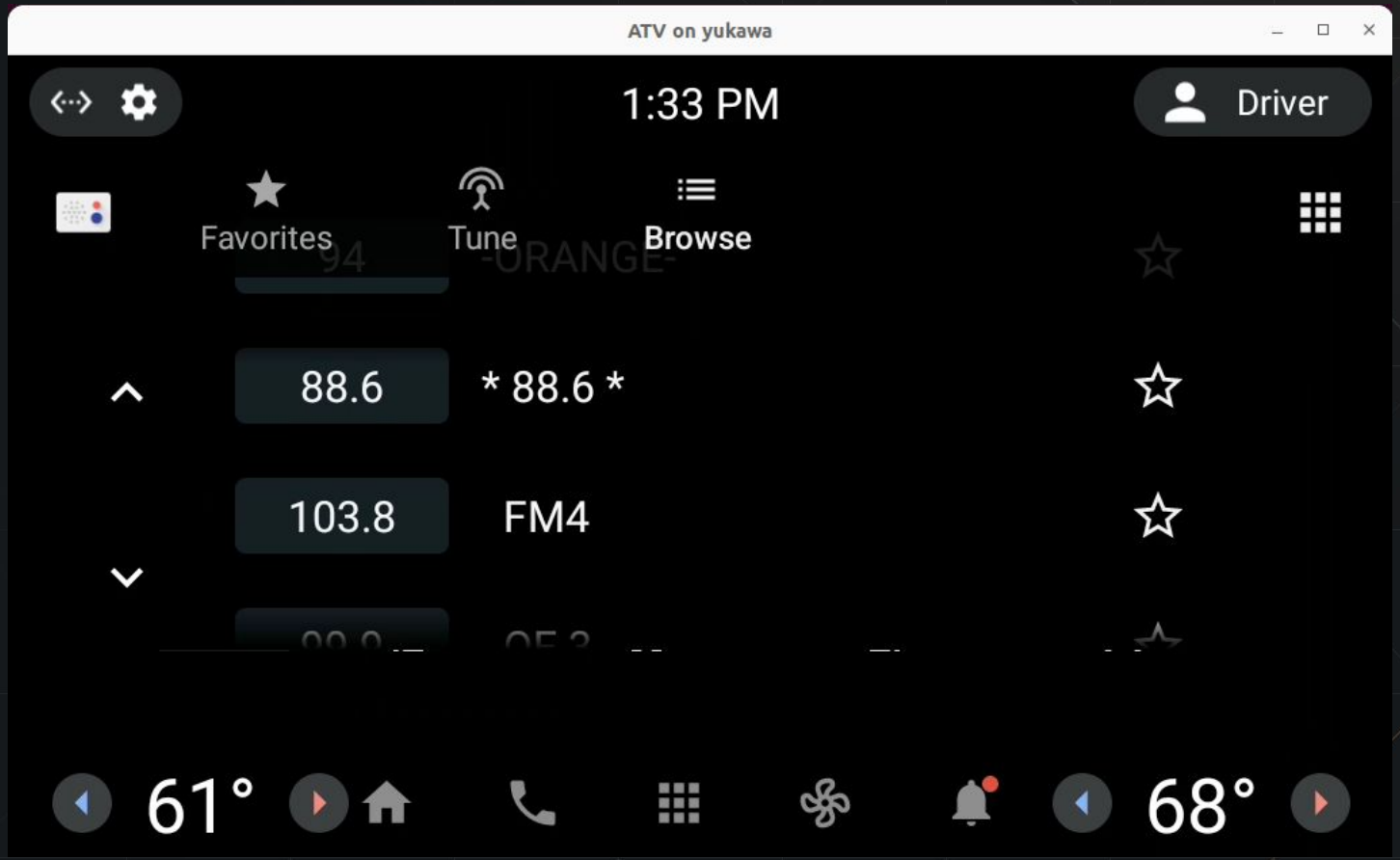
```
vim3:/ $ ifconfig
dummy0    Link encap:Ethernet  HWaddr 6a:e1:19:9c:3e:b2
          inet6 addr: fe80::68e1:19ff:fe9c:3eb2/64 Scope: Link
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 TX bytes:140

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 TX bytes:0
```

```
vim3:/ $ ifconfig
dummy0    Link encap:Ethernet  HWaddr 6a:e1:19:9c:3e:b2
          inet6 addr: fe80::68e1:19ff:fe9c:3eb2/64 Scope: Link
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 TX bytes:210

can0      Link encap:UNSPEC    Driver mcp251x
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:212 TX bytes:0

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 TX bytes:0

eth0      Link encap:Ethernet  HWaddr 12:77:07:72:8a:47  Driver meson8b-dwmac
          inet addr:192.168.2.186  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::8517:d4d2:d639:cf54/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:46 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14468 TX bytes:6603
          Interrupt:14
```

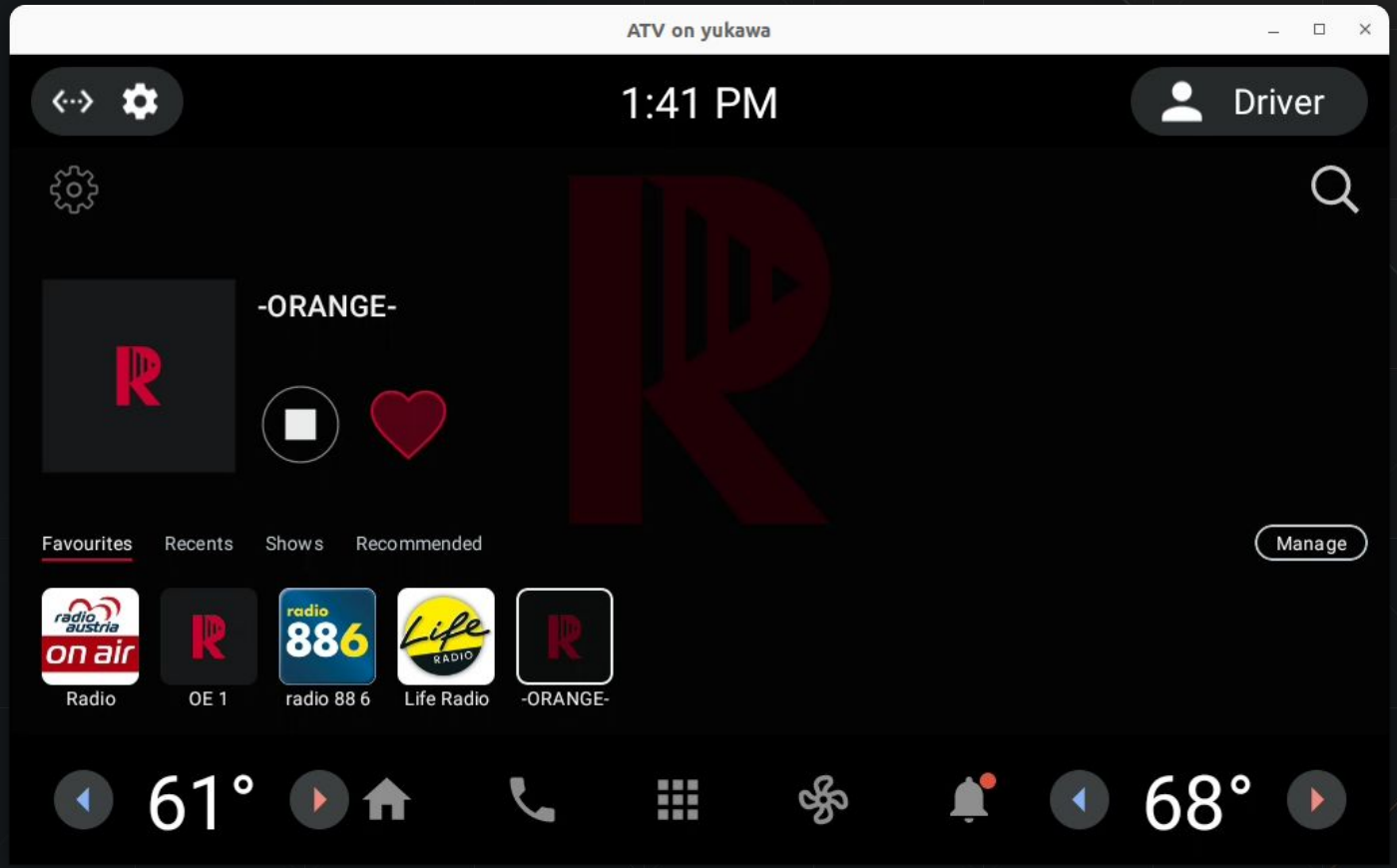
candump

```
vim3:/ $ candump can0
  can0  070   [7]  00 22 60 00 00 00 00
  can0  070   [7]  01 22 60 00 00 00 00
  can0  069   [1]  00
  can0  070   [7]  00 22 9C 2A 20 38 38
  can0  070   [7]  01 22 9C 2E 36 20 2A
  can0  070   [7]  00 23 1E 52 41 44 49
  can0  070   [7]  01 23 1E 4F 2D 57 20
  can0  070   [7]  00 23 F0 20 20 4F 45
  can0  070   [7]  01 23 F0 20 31 20 20
  can0  070   [7]  00 24 4A 00 00 00 00
  can0  070   [7]  01 24 4A 00 00 00 00
  can0  070   [7]  00 24 B8 00 00 00 00
  can0  070   [7]  01 24 B8 00 00 00 00
  can0  070   [7]  00 26 3E 00 00 00 00
  can0  070   [7]  01 26 3E 00 00 00 00
  can0  070   [7]  00 27 06 00 00 00 00
  can0  070   [7]  01 27 06 00 00 00 00
  can0  070   [7]  00 27 60 00 00 00 00
  can0  070   [7]  01 27 60 00 00 00 00
  can0  070   [7]  00 28 0A 00 00 00 00
  can0  070   [7]  01 28 0A 00 00 00 00
  can0  070   [7]  00 28 8C 20 20 46 4D
  can0  070   [7]  01 28 8C 34 20 20 20
  can0  070   [7]  00 29 54 00 00 00 00
  can0  070   [7]  01 29 54 00 00 00 00
  can0  070   [7]  00 2A 30 00 00 00 00
  can0  070   [7]  01 2A 30 00 00 00 00
  can0  070   [7]  00 22 2E 00 00 00 00
  can0  070   [7]  01 22 2E 00 00 00 00
  can0  070   [7]  00 22 60 00 00 00 00
  can0  070   [7]  01 22 60 00 00 00 00
```

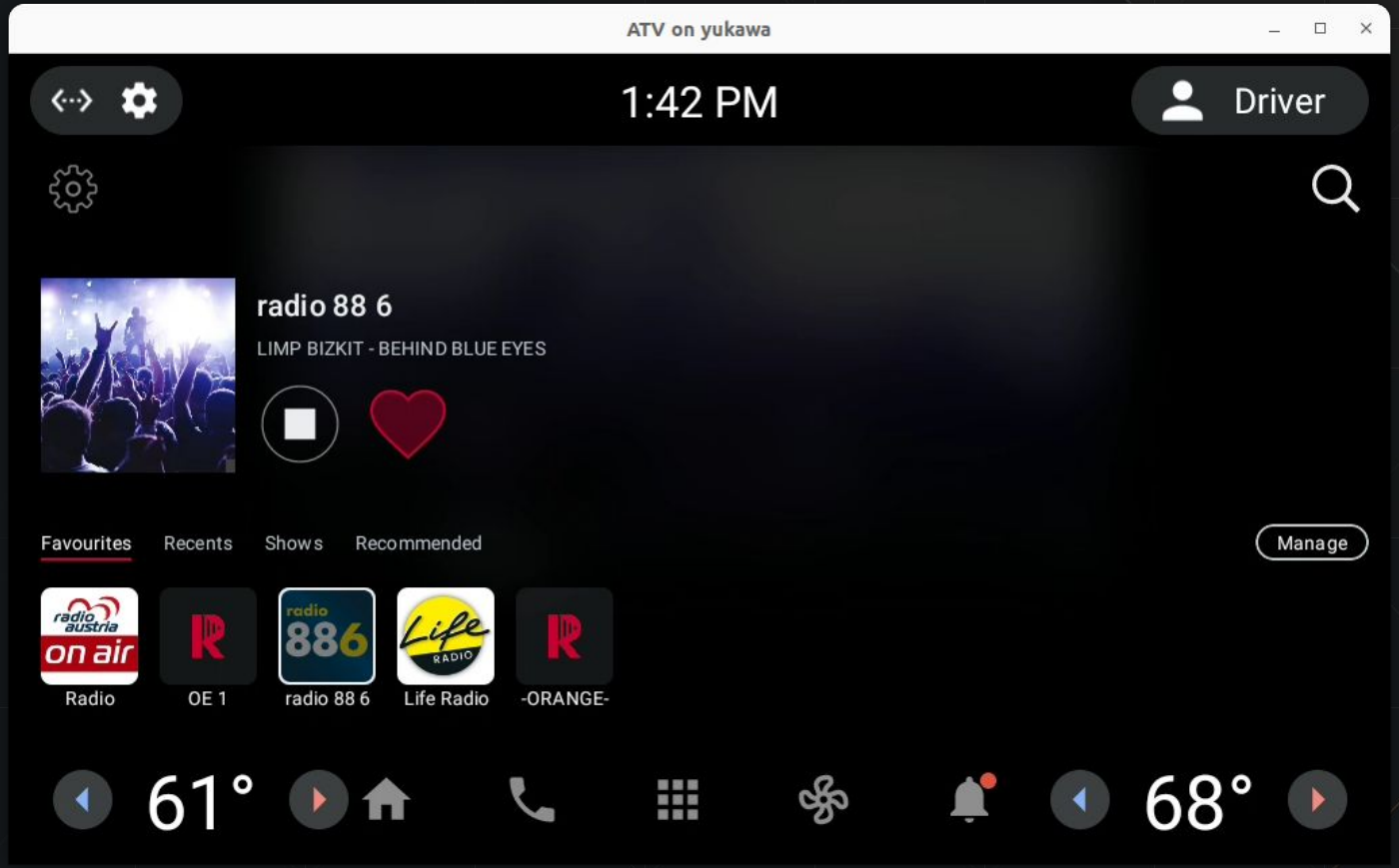
Radio


ATV on yukawa
1:33 PM
Driver
Favorites
Tune
Browse
88.6
* 88.6 *
103.8
FM4
61°
68°

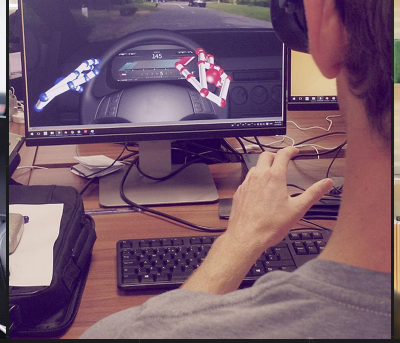
Radioplayer



Radioplayer








C O N J U R E

Simon Osim, Head of Technology
simon.osim@conjure.co.uk
+43 (0)664 40 41 7 41