



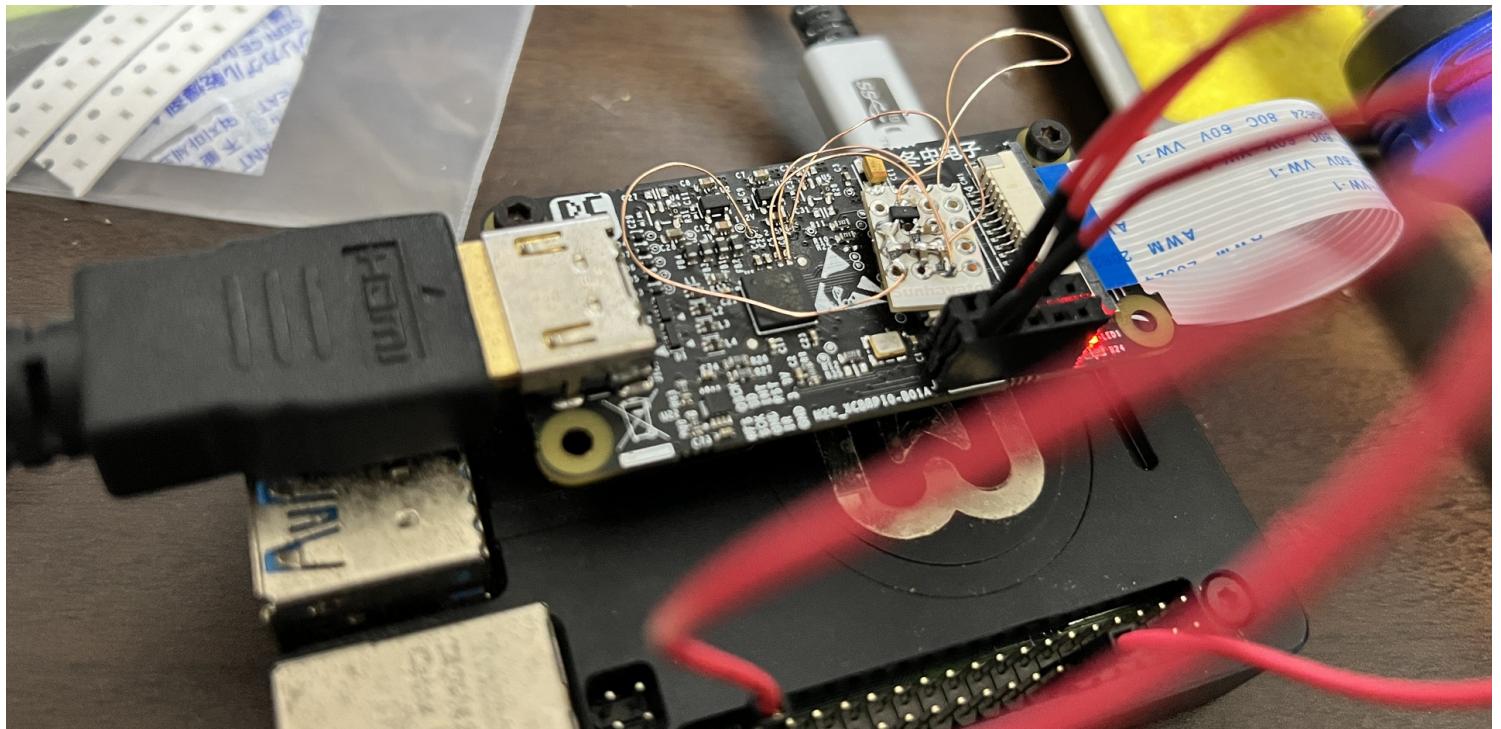
Japonais



Anglais



## MENU



# *Capture sound with Raspberry Pi CM4 and CSI-2 connection HDMI input board*



by mzyy94

18 minute read • 2022-03-01

CATEGORY Multimedia TAG hdmi, i2s, raspberry-pi, tc358743

[In the article about capturing HDMI input with Raspberry Pi that I posted two years ago , I only captured video, but this HDMI input board is equipped with a chip \[TC358743 that can also capture audio.\]\(#\)](#) ↗ The HDMI input board I purchased had a terminal that could be imported, so I attached a

pin header and looked at the signal, but I couldn't get the audio output normally. I'll keep a record of how I've hunted around and solved this problem.

## Making full use of HDMI input with Raspberry Pi | Dog Icon Mickey

In 2019, I wanted to do something like video distribution to keep up with the unabated technological trends of recent years. I didn't have much time to do anything, and before I knew it, it was 2020. I thought I wouldn't have much time this year, but thanks to working from home every day, my commuting time became zero, and I had some leisure time, so I...

10th Apr 2020



### TABLE OF CONTENTS

## # 2020 trying to extract audio with HDMI input board

Actually, at that time, I was going through a lot of trial and error until just before posting the blog post to see if I could get the correct audio output. Some of the links used for the investigation are left in the [reference link](#). However, it was not easy and I could not get audio output, so I discarded the draft I was writing at that time. Although it was resolved after a while, I didn't [have the motivation](#) to complete the article, and it was planned to be shelved. I was. So the content is a record of two years ago, but it has been brushed up.

At that time, there was no new Bullseye-based Raspberry Pi OS, so the contents of this article are all about the Buster-based Raspberry Pi OS. In addition, at the time of writing this article (2022/02/26), the latest version of the following configuration has been re-checked.

- ▶ OS: Raspberry Pi OS
- ▶ Hardware: Raspberry Pi 4B
- ▶ Kernel: 5.10.63-v7l+
- ▶ Firmware: Dec 1 2021 15:02:46

## # HDMI input board H2C-RPI-B01 and TC358743XBG

The HDMI input board used in the [previous article](#), [H2C-RPI-B01](#) Raspbian It was processed by the standard driver of Raspberry Pi OS, it was recognized as a V4L2 device similar to [Raspberry Pi Camera Module just by connecting it, and it was able to handle video input](#). It was good that it worked without doing anything, but because it didn't do anything, it could only handle images like the Camera Module.

TC358743XBG, the chip installed in H2C-RPI-B01, can actually handle HDMI audio input. The data sheet says that audio data is exchanged using the I2S bus standard.



**TC358743XBG**

CMOS Digital Integrated Circuit Silicon Monolithic

## TC358743XBG

Mobile Peripheral Devices

### Overview

The HDMI®-RX to MIPI® CSI-2-TX is a bridge device that converts HDMI stream to MIPI CSI-2 TX.

The current and next generation Application Processors and Baseband chips have been designed without video streaming input port except CSI-2 for Camcorder input. Smart Phone Processors are being used in several applications that required Video Input

TC358743XBG takes in HDMI input and converts to CSI-2 that looks like a Camcorder input.



P-TFBGA64-0606-0.65-001  
Weight: 76 mg (Typ.)

### Features

- HDMI-RX Interface
  - ✧ HDMI 1.4
  - Video Formats Support (Up to 1080P @60fps)
  - RGB, YCbCr444: 24-bpp @60fps
- Audio Output Interface
  - Either I2S or TDM Audio interface available (pins are multiplexed)
  - I2S Audio Interface

Source: TC358743XBG\_datasheet\_en\_20171026.pdf [TC358743XBG | HDMI® Interface Bridge | Toshiba Electronic Devices & Storage Corporation | Japan](#) ↗

## # I2S (Inter-IC Sound Bus)

I2S is an audio communication bus standard called Inter-IC Sound.

[I2S - Wikipedia](#) ↗

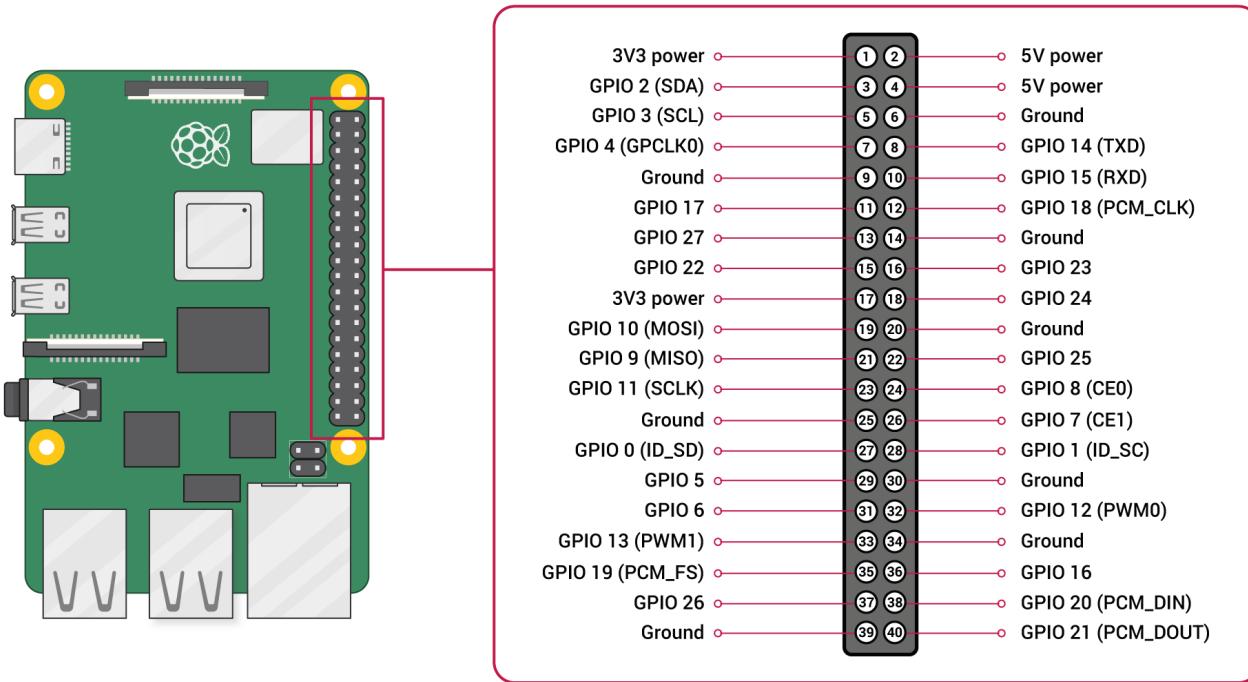
NXP Semiconductors defined the specification in 1986, but the original specification has sunk into the sea of the Internet and can only be found in archives.

[I2SBUS.pdf - Internet Archive](#) ↗

It is a serial bus standard that exchanges clock and data with four pins, and the summary is written in the Japanese Wikipedia article.

[Inter-IC Sound - Wikipedia](#) ↗

The Raspberry Pi's GPIO terminals support both audio input and output, and the pins labeled starting with PCM\_ (GPIO 18, 19, 20, 21) are I2S.



Source: [GPIO and the 40-pin Header - Raspberry Pi Documentation](#) ↗

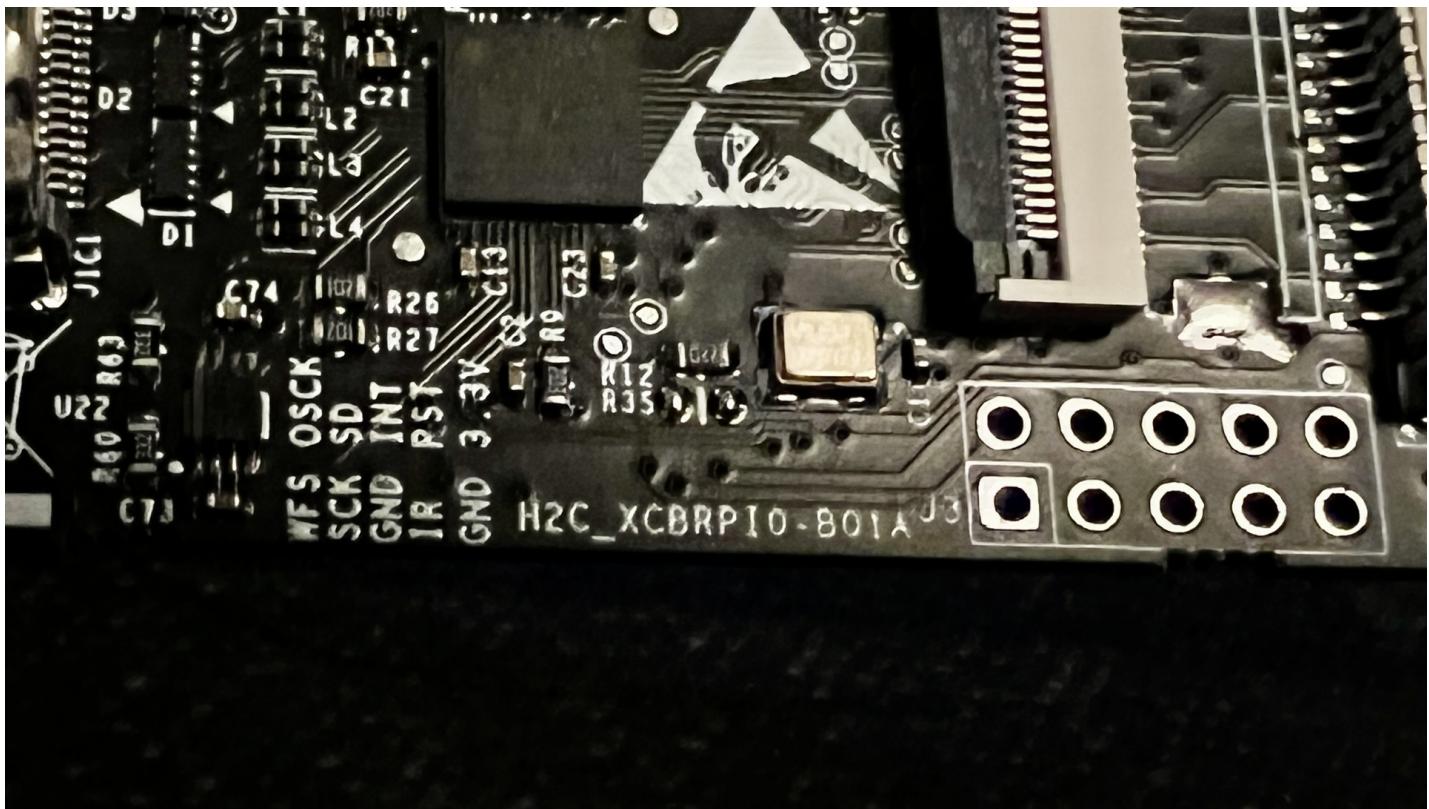
## # First attempt to extract audio (April 2020)

We will make I2S hardware connection and firmware settings and try to capture audio.

### ## Hardware wiring

Connect the output pin of the HDMI input board to the I2S pin of the Raspberry Pi 40 pin header.

The HDMI input board has a through hole for attaching a 10-pin input/output terminal. I bought many similar products other than H2C-RPI-B01, but most of them are on boards with the same TC358743XBG.



A little further away is a piece of silk showing the function of each through-hole.

HDMI input board silk	Raspberry Pi pin number	Raspberry Pi GPIOs
SCK	12	PCM_CLK (GPIO 18)
WFS	35	PCM_FS (GPIO 19)
SD	38	PCM_DIN (GPIO 20)

## ## Raspberry Pi OS firmware settings

Since you can't handle audio input just by wiring I2S, let's set the firmware and load the driver.

## ### Device Tree

When Linux loads the drivers, it selects the appropriate driver based on the attached device information. Device Tree is the hardware configuration information that manages connected devices. **If you look into the proc file system of /proc/device-tree**, you can see roughly what is recognized by Device Tree . The document is detailed below.

- [Device Trees, overlays, and parameters - Raspberry Pi Documentation](#)
  - [Device Tree Reference - eLinux.org](#)

Media/i2c/tc358743.c, the driver for TC358743, finds the device and initializes (probes) the driver as follows.

```
static const struct of_device_id tc358743_of_match[] = {
    { .compatible = "toshiba,tc358743" },
    {},
};

MODULE_DEVICE_TABLE(of, tc358743_of_match);
```

### [linux/drivers/media/i2c/tc358743.c#L2255-L2259 ↗](#)

If there is a device listed as "toshiba,tc358743" in compatible, the tc358743 driver will be loaded. However, in the Raspberry Pi OS where nothing is set, there is no device with such a description, and vc04\_services/bcm2835-camera.c of Camera Module is loaded as a driver.

### [linux/bcm2835-camera.c at 1.20220120 raspberrypi/linux ↗](#)

Overwrite the Device Tree and set the firmware to load the TC358743 driver media/i2c/tc358743.c instead of the Camera Module driver.

### ### Device Tree Overlay

The name of the function that overwrites the Device Tree is Device Tree Overlay. That's it. The Raspberry Pi OS can overwrite the Device Tree by reading this. The Raspberry Pi OS stores the Device Tree Overlay configuration file, Device Tree Blob Overlay (.dtbo), in **/boot/overlays**.

### [firmware/boot/overlays at 1.20220120 raspberrypi/firmware ↗](#)

Blobs, as the name implies, are compiled into binary files, but the source code for these can be found in **/arch/arm/boot/dts/overlays** on raspberrypi/linux.

### [linux/arch/arm/boot/dts/overlays at 1.20220120 raspberrypi/linux ↗](#)

There is an overlay for tc358743 in it and it says:

```
__overlay__ {
    #address-cells = <1>;
```

```
#size-cells = <0>;
status = "okay";

tc358743@0f {
    compatible = "toshiba,tc358743";
    reg = <0x0f>;
    status = "okay";
```

### [linux/tc358743-overlay.dts at 1.20220120 raspberrypi/linux ↗](#)

compatible = "toshiba,tc358743"; So if you read this dtbo, tc358743 will be recognized as a device and the driver will be loaded.

However, this alone cannot handle voice input. In order to exchange audio with the HDMI input board by I2S, it is necessary to load the Device Tree definition of the audio interface and the driver. Also the definition for I2S in TC358743XBG, RaspbianIt is provided in the Device Tree Overlay of the Raspberry Pi OS.

### [linux/tc358743-audio-overlay.dts at 1.20220120 raspberrypi/linux ↗](#)

I will add to /boot/config.txt while understanding [how to write](#) these Device Tree Overlay and I2S parameter settings in /boot/config.txt.

```
dtparam=i2s=on
dtparam=audio=on
dtoverlay=tc358743
dtoverlay=tc358743-audio
```

After rebooting, you `lsmod` can see that the `media/i2c/tc358743.c` driver is loaded. You can check if the overlay is loaded correctly with e.g. `dmesg sudo vcdbg log msg`

```
pi@raspberrypi:~ $ dmesg | grep tc358743
[    4.137243] tc358743 0-000f: tc358743 found @ 0x1e (bcm2835 I2C adapter)
pi@raspberrypi:~ $ lsmod | grep tc358743
tc358743                  40960  1
v4l2_dv_timings            36864  2 bcm2835_unicam,tc358743
```

```
v4l2_fwnode          20480  2 bcm2835_unicam,tc358743
v4l2_common           16384  3 bcm2835_unicam,bcm2835_v4l2,tc358743
videodev              200704  9 bcm2835_unicam,v4l2_fwnode,bcm2835_codec,
media                  36864  5 bcm2835_unicam,bcm2835_codec,videodev,v4l2
pi@raspberrypi:~ $ ls /dev/video*
/dev/video0  /dev/video10  /dev/video11  /dev/video12
pi@raspberrypi:~ $
```

## ## Capturing audio output signals

After completing the firmware and hardware settings, we will extract the audio signal with I2S.

Unlike normal I2S audio devices, TC358743 needs to be instructed by I2C through CSI-2 to control audio signals. There was information on the forum regarding this control.

[HDMI to CSI-2 TC358743 I2S Audio - Raspberry Pi Forums ↗](#)

Detailed operations are done by the driver, so it seems that v4l2 can be used for general settings.

```
pi@raspberrypi:~ $ wget https://raw.githubusercontent.com/mzyy94/ns-remo1
pi@raspberrypi:~ $ v4l2-ctl --set-edid=file=720P30EDID.txt
pi@raspberrypi:~ $ v4l2-ctl --set-dv-bt-timings query
```

If you have made it this far, try outputting audio from the device connected to HDMI, and you `v4l2-ctl --list-ctrls` should be able to confirm that `audio_present` is `value=1`.

```

CTA-861 Video Capability Descriptor
RGB Quantization Range: yes
YCC Quantization Range: no
PT: Supports both over- and underscan
IT: Supports both over- and underscan
CE: Supports both over- and underscan
[pi@raspberrypi:~ $ v4l2-ctl --query-dv-timings
Active width: 1280
Active height: 720
Total width: 1650
Total height: 750
Frame format: progressive
Polarities: -vsync -hsync
Pixelclock: 74250000 Hz (60.00 frames per second)
Horizontal frontporch: 0
Horizontal sync: 370
Horizontal backporch: 0
Vertical frontporch: 0
Vertical sync: 30
Vertical backporch: 0
Standards:
Flags:
[pi@raspberrypi:~ $ v4l2-ctl --list-ctrls
User Controls
audio_sampling_rate 0x00981980 (int) : min=0 max=768000 step=1 default=0 value=48000 flags=read-only
audio_present 0x00981981 (bool) : default=0 value=1 flags=read-only
Digital Video Controls
power_present 0x00a00964 (bitmask): max=0x00000001 default=0x00000000 value=0x00000001 flags=read-only
[pi@raspberrypi:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: tc358743 [tc358743], device 0: bcm2835-i2s-dir-hifi dir-hifi-0 [bcm2835-i2s-dir-hifi dir-hifi-0]
Subdevices: 1/1
Subdevice #0: subdevice #0
[pi@raspberrypi:~ $ ]

```

pi@raspberrypi:~ \$ v4l2-ctl --list-ctrls

## User Controls

```

audio_sampling_rate 0x00981980 (int) : min=0 max=768000 s1
audio_present 0x00981981 (bool) : default=0 value=1 1

```

## Digital Video Controls

```

power_present 0x00a00964 (bitmask): max=0x00000001 defa

```

When this happens, all you have to do is take it in. However, when I try to record the I2S audio input, no data comes. And there seems to be some kind of error.

pi@raspberrypi:~ \$ sudo dmesg -C

```

pi@raspberrypi:~ $ arecord -D plughw:1 -c1 -r 48000 -f S32_LE -t wav -V n
Recording WAVE '/tmp/rec.wav' : Signed 32 bit LittleEndian, Rate 48000 Hz
^CAborted by signal Interrupt...

```

```
arecord: pcm_read:2145: read error: Interrupted system call
pi@raspberrypi:~ $ dmesg
[ 560.152152] bcm2835-i2s fe203000.i2s: I2S SYNC error!
pi@raspberrypi:~ $
```

This is an error that prevents synchronization around the I2S clock. When I observed the I2S signal with an oscilloscope, I realized that the waveform was indeed strange and there was likely a problem on the board.

固定資産税減免申請

@mzyy94 · [フォローする](#)

返信先: @chorntwiさん

ケーブル4本をはんだで繋げてみたんですが、I2S信号が期待通りではなくてALSAで取り込めませんでした😭

午前3:14 · 2020年5月26日 (i)

---

1    返信    リンクをコピー

[4件の返信を読む](#)

I burned the oscilloscope while I was trying various things, so I gave up on audio input from the HDMI input board and decided to go through the UAC Gadget.

固定資産税減免申請

@mzyy94 · [Follow](#)

プローブがショートして白い煙出しながらオシロスコープ発火してモニタ一消えたので終わり。

11:38 AM · May 21, 2020 (i)

---

7    Reply    Copy link

[Read more on Twitter](#)

## Import audio output from Nintendo Switch to Raspberry Pi with UAC Gadget | Dog Icon Mickey

In the previous article, I introduced how to handle HDMI video input with Raspberry Pi, and at the end, I wrote that there are still issues regarding audio capture. Although we are trying to solve the essential problem of capturing audio from HDMI input, we are facing a rather difficult problem, so we will limit the target and try to solve it partially. The device I'm mainly...

17th Apr 2020



### # Revenge to take out the voice

A few months after the article about inputting audio from UAC Gadget, a method was posted on the Raspberry Pi Forums to solve the problem of not being able to capture audio with a similar HDMI input board by applying a hardware patch. I was.

[\[UPDATED, SOLVED\] Why “cheap” Chinese HDMI-to-CSI2 adapters \(eg18810-1 C779\) don't provide I2S sound - Raspberry Pi Forums ↗](#)

This showed me that I could patch it and get a signal, so I decided to give it a try.

### ## Missing wiring and I2S APLL patch

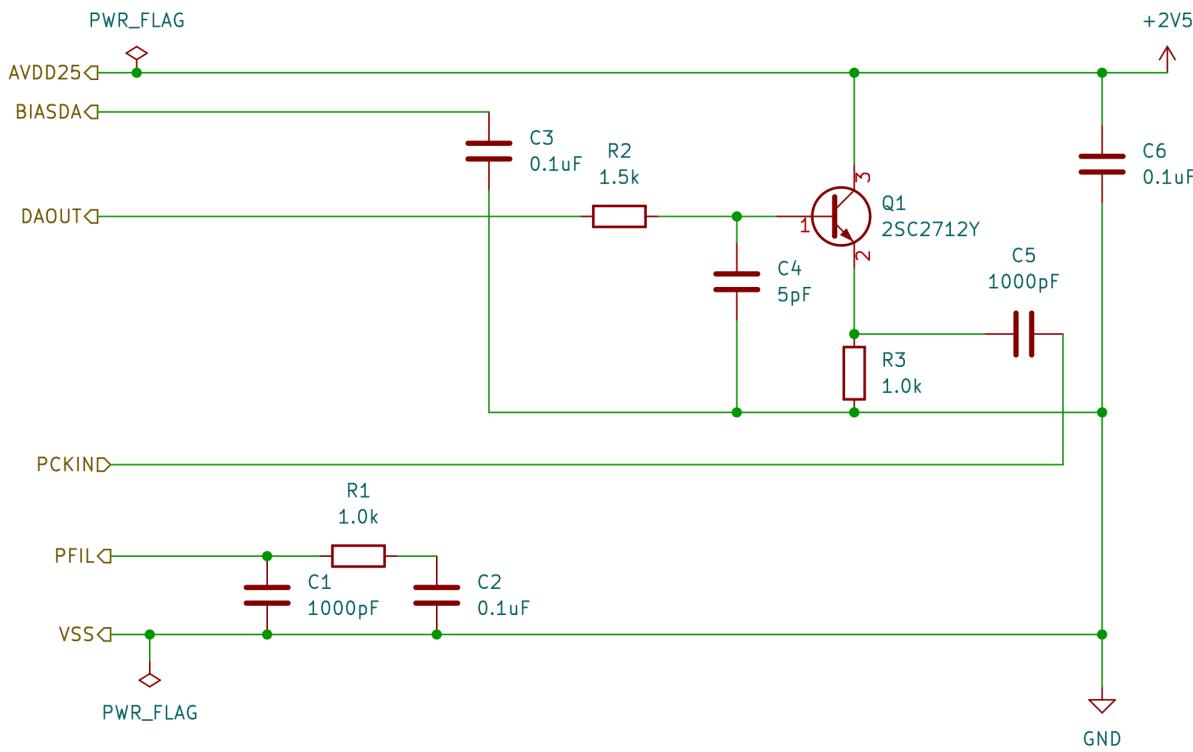
It seems that the wiring for correctly handling I2S signals with TC358743 is written in the specifications called Functional Specification. I say "likely" because the specifications cannot be obtained without an NDA with Toshiba, and we have not been able to confirm the actual product.

It's a document under NDA, so it's hard to find it in the sea of the Internet, but Firefly has released the specifications of TC358749XBG, which is a brother chip whose number has changed from 3 to 9.

[\(U18\)TC358749XBG\\_V074.pdf ↗](#)

I will use this as a reference to create a patch.

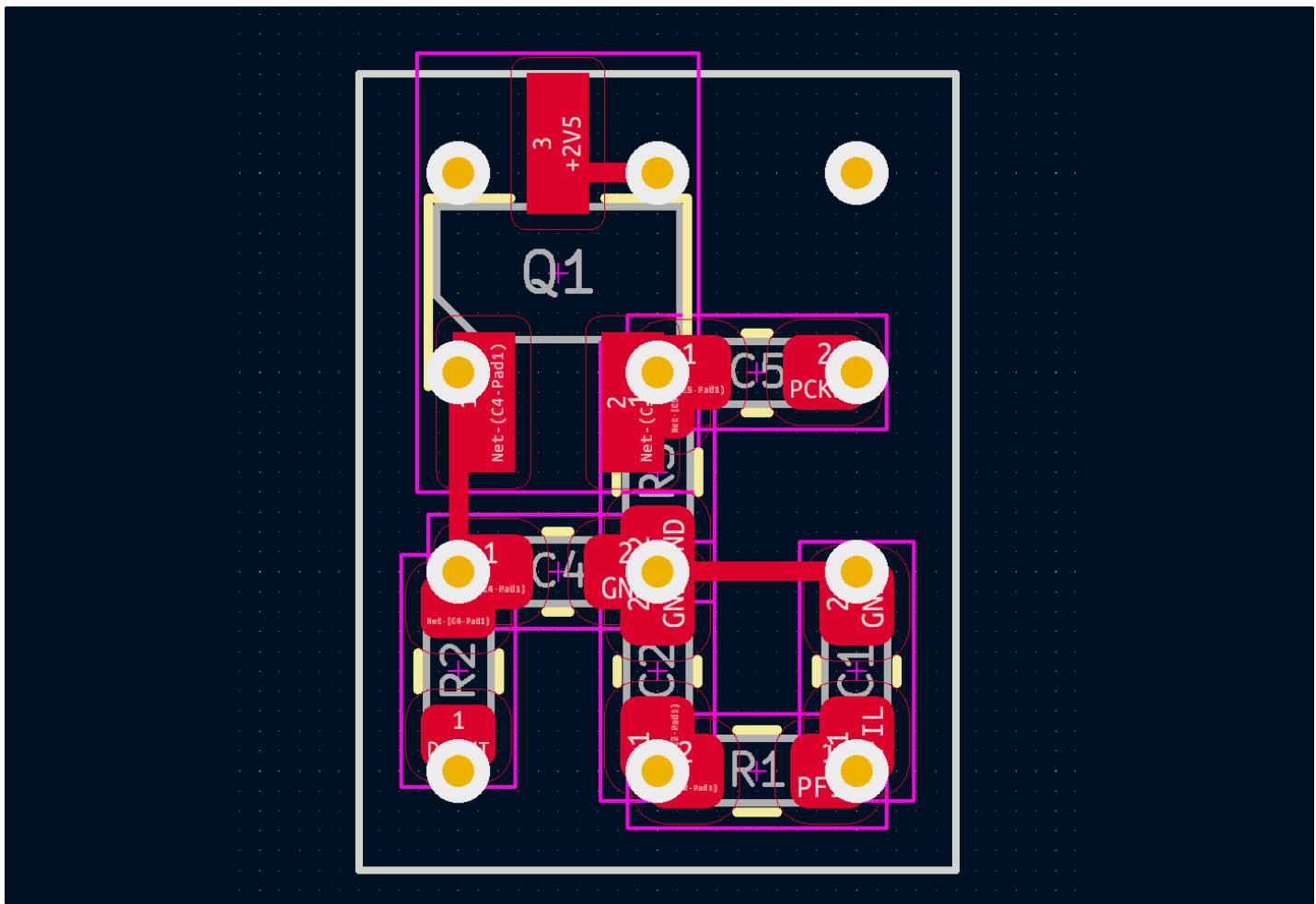
According to a post on the Raspberry Pi Forums, the I2S signal is strange because the audio signal phase locked loop (PLL) is not implemented. When I searched for the functional specifications related to this Audio PLL from the specifications of the brother chip TC358749XBG, I found that the following circuit should be prepared.



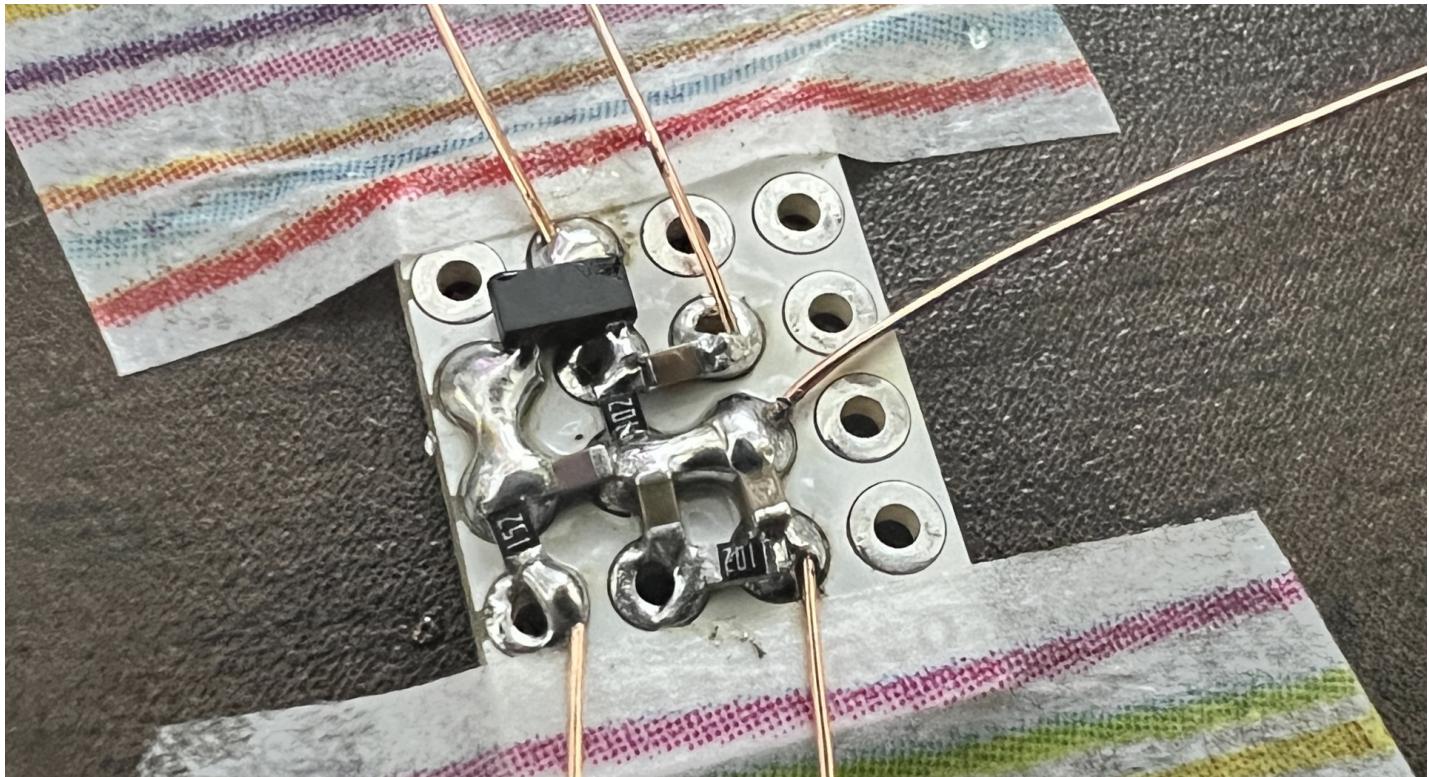
The resistance and capacitance are estimated by reading many other [TC358 series specifications](#).  
↗

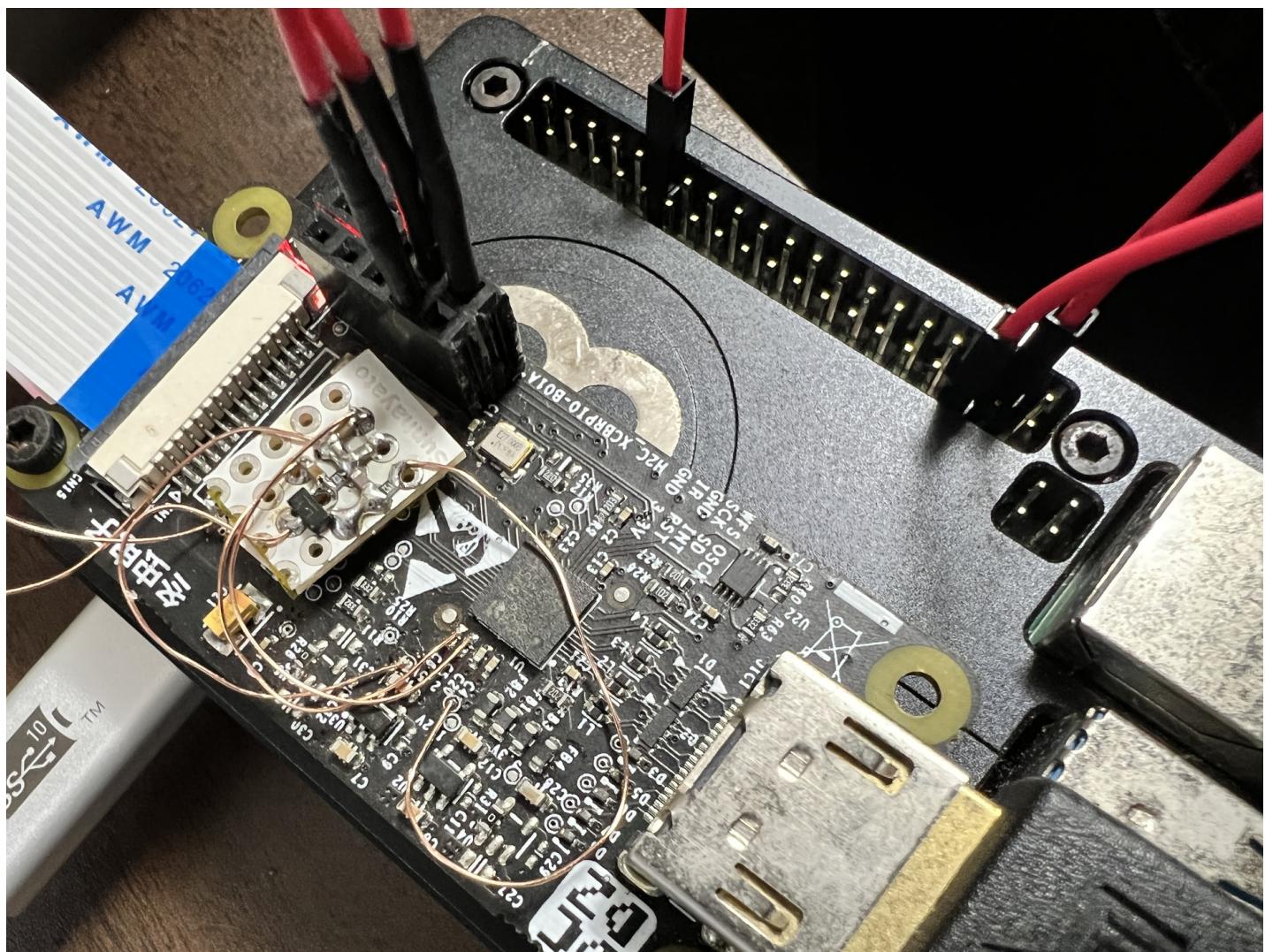
### ### Mounted on a universal board

In the circuit diagram above, C3 and C6 are mounted on the H2C-RPI-B01, so I wired the rest and arranged them as follows.



We will mount this on a universal board and connect it to the H2C-RPI-B01 while referring to the analysis results of similar boards on the forum.





## ## try recording again

Just like the previous challenge, I finished setting Device Tree Overlay and v4l2 and recorded with arecord. was.

```

デスクトップ - pi@raspberrypi: ~ — ssh pi@192.168.38.24 — 143x38

audio_present 0x00981981 (bool) : default=0 value=1 flags=read-only

Digital Video Controls

power_present 0x00a00964 (bitmask): max=0x00000001 default=0x00000000 value=0x00000001 flags=read-only
[pi@raspberrypi:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: tc358743 [tc358743], device 0: bcm2835-i2s-dir-hifi dir-hifi-0 [bcm2835-i2s-dir-hifi dir-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
[pi@raspberrypi:~ $ arecord -v -D hw:tc358743 -c2 -r 48000 -f S16_LE -t wav -V stereo /tmp/rec.wav
Recording WAVE '/tmp/rec.wav' : Signed 16 bit LittleEndian, Rate 48000 Hz, Stereo
Hardware PCM card 1 'tc358743' device 0 subdevice 0
Its setup is:
stream      : CAPTURE
access      : RW_INTERLEAVED
format      : S16_LE
subformat   : STD
channels    : 2
rate        : 48000
exact rate  : 48000 (48000/1)
msbits      : 16
buffer_size : 24000
period_size : 6000
period_time : 125000
tstamp_mode : NONE
tstamp_type : MONOTONIC
period_step : 1
avail_min   : 6000
period_event: 0
start_threshold : 1
stop_threshold  : 24000
silence_threshold: 0
silence_size  : 0
boundary     : 1572864000
appl_ptr     : 0
hw_ptr       : 0
#####
  32%|34%#####
+#####

```

## # Make it available for reTerminal and CM4

Now that you've successfully recorded with the Raspberry Pi 4B, it's time to put it in a case and use it as an extension module for reTerminal! Although I was enthusiastic about it, I2C passing through CSI-2 could not communicate well and it did not work with reTerminal.

```

デスクトップ - pi@raspberrypi: ~ — ssh pi@192.168.63.114 — 125x16

[pi@raspberrypi:~ $ dmesg | grep tc3
[ 10.534939] tc358743 10-000f: i2c_rd: reading register 0x0 from 0xf failed
[ 10.534957] tc358743 10-000f: not a TC358743 on address 0x1e
[pi@raspberrypi:~ $ ls /dev/i2c-*
/dev/i2c-0  /dev/i2c-1  /dev/i2c-10  /dev/i2c-20  /dev/i2c-21  /dev/i2c-22  /dev/i2c-3
[pi@raspberrypi:~ $ i2cdetect -y 22
  0: -- -- -- -- -- -- -- -- -- --
  10: -- -- -- -- -- -- -- -- -- --
  20: -- -- -- -- -- -- -- -- -- --
  30: -- -- -- -- -- -- -- -- -- --
  40: -- -- -- -- -- -- -- -- -- --
  50: -- -- -- -- -- -- -- -- -- --
  60: -- -- -- -- -- -- -- -- -- --
  70: -- -- -- -- -- -- -- -- -- --
pi@raspberrypi:~ $

```

pi@raspberrypi:~ \$ dmesg | grep tc3

[ 10.534939] tc358743 10-000f: i2c\_rd: reading register 0x0 from 0xf f

[ 10.534957] tc358743 10-000f: not a TC358743 on address 0x1e

After playing around with I2C settings, researching error messages and two CSI-2 specifics to the Compute Module, I stumbled upon a forum post that led me straight to the solution.

## [Two B102 TC358743 simultaneously on CM4 - Raspberry Pi Forums ↗](#)

There are two MIPI CSI-2 buses in CM3/CM4, and I was told that it is necessary to change the I2C bus in the Device Tree to operate with CSI0. The forum post is a little old, so I fixed it to match the latest firmware, and it was recognized properly and became able to record. I put the patch on GitHub.

## [reTerminal-HDMI-input/overlays at master mzyy94/reTerminal-HDMI-input ↗](#)

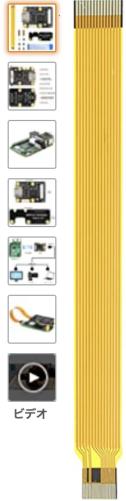
### # summary

Finally, I was able to make it work as an HDMI input expansion module. The results of this time are also summarized in the following repository, so please refer to those who are interested.

## [mzyy94/reTerminal-HDMI-input: HDMI input expansion module for reTerminal ↗](#)

By the way, while I was writing this article, I learned that a cheap HDMI input board **with this patch that can also capture audio was released in January of this year**. It's a bit expensive, but you don't have to bother to create an APLL patch. What are your hardships so far? . . . 

ⓘ 最後にこの商品を購入したのは2022/2/27です。  
[注文の詳細を表示](#)





Geekworm Raspberry Pi Hdmi to CSI-2 (Hdmi入力の最大1080p/25fps) X630 拡張ボード、Raspberry Pi 4B/3B+/3B/Pi Zero/Zero Wに適用  
[Geekwormのストアを表示](#)

¥3,699

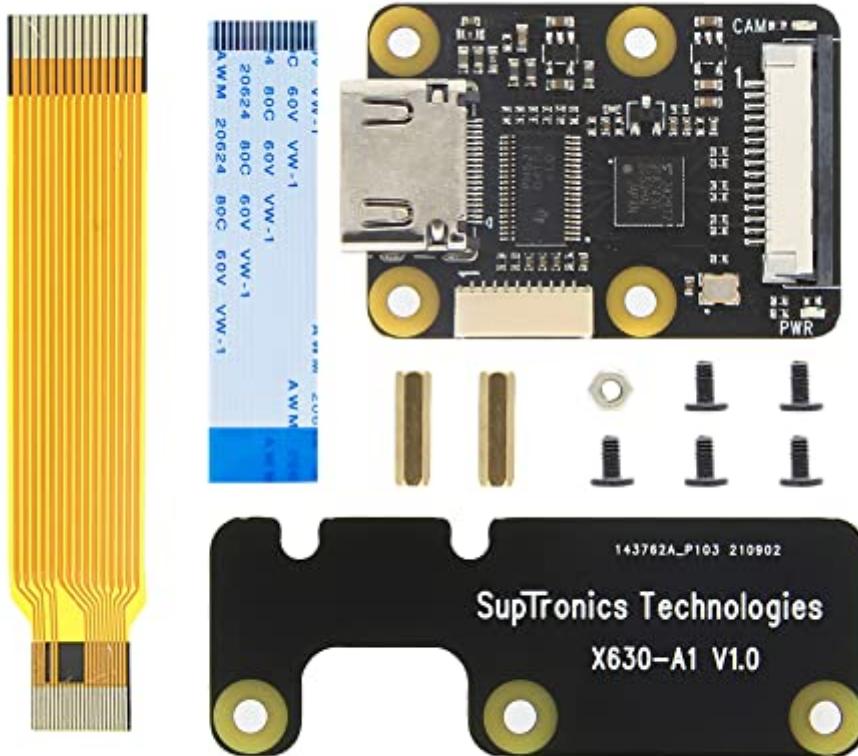
✓prime

Amazon Prime Mastercard新規ご入会で7,000ポイントプレゼント  
入会特典をこの商品に利用した場合 **0円 3,699円** に

OS	Linux
CPUモデル	Unknown
ブランド	Geekworm
CPUメーカー	ARM
ハードディスクインターフェイス	Unknown

画像にマウスを合わせると拡大されます

amen



[Amazon | Geekworm Raspberry Pi Hdmi to CSI-2 \(Maximum 1080p / 25fps for Hdmi input\) X630 expansion board, applied to Raspberry Pi 4B / 3B+ / 3B / Pi Zero / Zero W | Geekworm | Barebone PC mail order](#)

### X630 & X630-A2 hdmi to CSI-2 Module for Raspberry Pi 4B/3B+/3B



**Not include Raspberry Pi**

[HDMI to CSI 2 Module, X630 HDMI CSI 2 for Raspberry Pi 4B/3B+/3B/Pi Zero Demo Board](#)[Accessories - AliExpress ↗](#)

BI

Tweet

Share

LinkedIn

Reddit

## この記事はどうでしたか？

7 Responses



グッド



おもしろい



これすき



新発見



なんだこれ



しょぼーん

0 件のコメント

1 ログイン

G

コメントを投稿する...

下記でログインして投稿

またはDISQUSに登録 ?

名前



共有

評価順

新着順

投稿順

最初のコメントをしましょう！

[更新を受取る](#)[Privacy](#)[Do Not Sell My Data](#)**Forward**[←3D printed electronics board case for reTerminal with Shapeways](#)**to the next**[Introduced Photon Mono 4K stereolithography 3D printer →](#)



© 2023 mzyy94. Powered by Jekyll & So Simple .