Introduction

The öchìn CM4 it's a tiny carrier board for the Raspberry Pi Compute Module 4. It is designed for applications where a powerful machine with low consumption and small dimensions is required. The small form factor makes it interesting for all those applications where the space available is not much and containing the weight is important, such as in robotics, home automation and IOT.

The board is compatible with all Raspberry Pi CM4 modules equipped with eMMC. Depending on your needs, you can select a CM4 module with an SDRAM starting from 1GB up to 8GB and the eMMC from 8GB up to 32GB, with or without the WiFi / BT4 connection.

Design

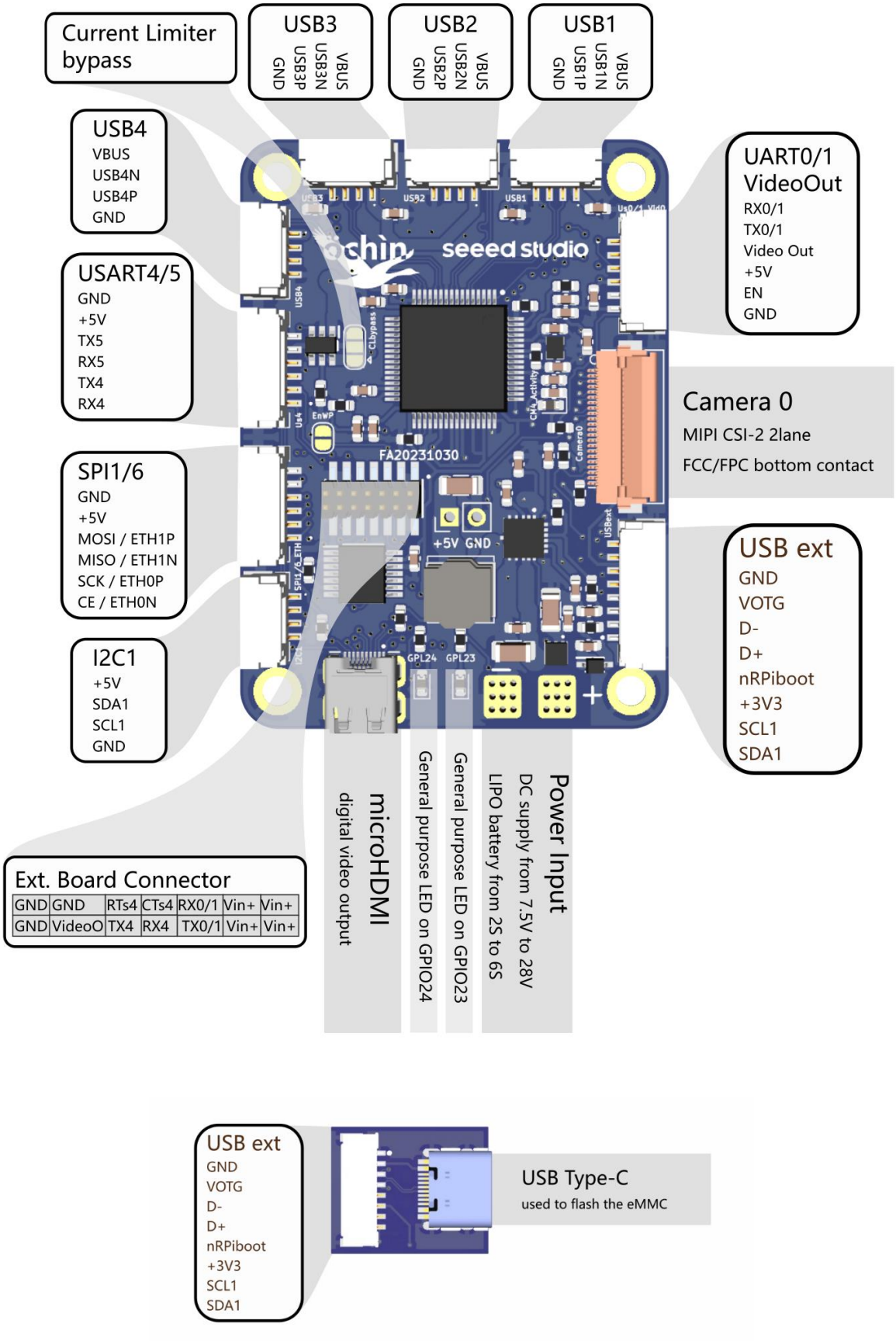
The board has the same dimensions as the RPi CM4 module, 55mm x 40mm with rounded corners. öchìn CM4 is composed of three parts, the carrier board itself and two covers, meant to protect the carrier board and the RPi CM4 module (to be purchased separately). The four cards thus form a stack of 18mm tall.
Several interfaces like USARTs, USBs are addressable bye the SM06B-GHS and SM04B-GHS connectors, available on the sides of the board. Two FCC/FPC connector are available to connect two MIPI CSI-2 cameras. The front cam is the "camera 1" which is a MIPI CSI-2 4lane camera. The side cam is the "camera 0" which is a MIPI CSI-2 2lane camera. The two FCC/FPC connectors have contacts on the bottom, the interface is the same used on the Raspberry Pi Zero.
On top of the board there is a 14x1.27mm connector, that could be used to connect a custom board on top, replacing the top PCB cover.
On the bottom of the öchìn carrier board there are the DF40HC(3.0)-100DS-0.4v mating connectors for the CM4 module. The bottom cover is meant to protect the boards and

mount the whole stack. The bottom cover could be eventually replaced by a commercial heatsink for the CM4 module.

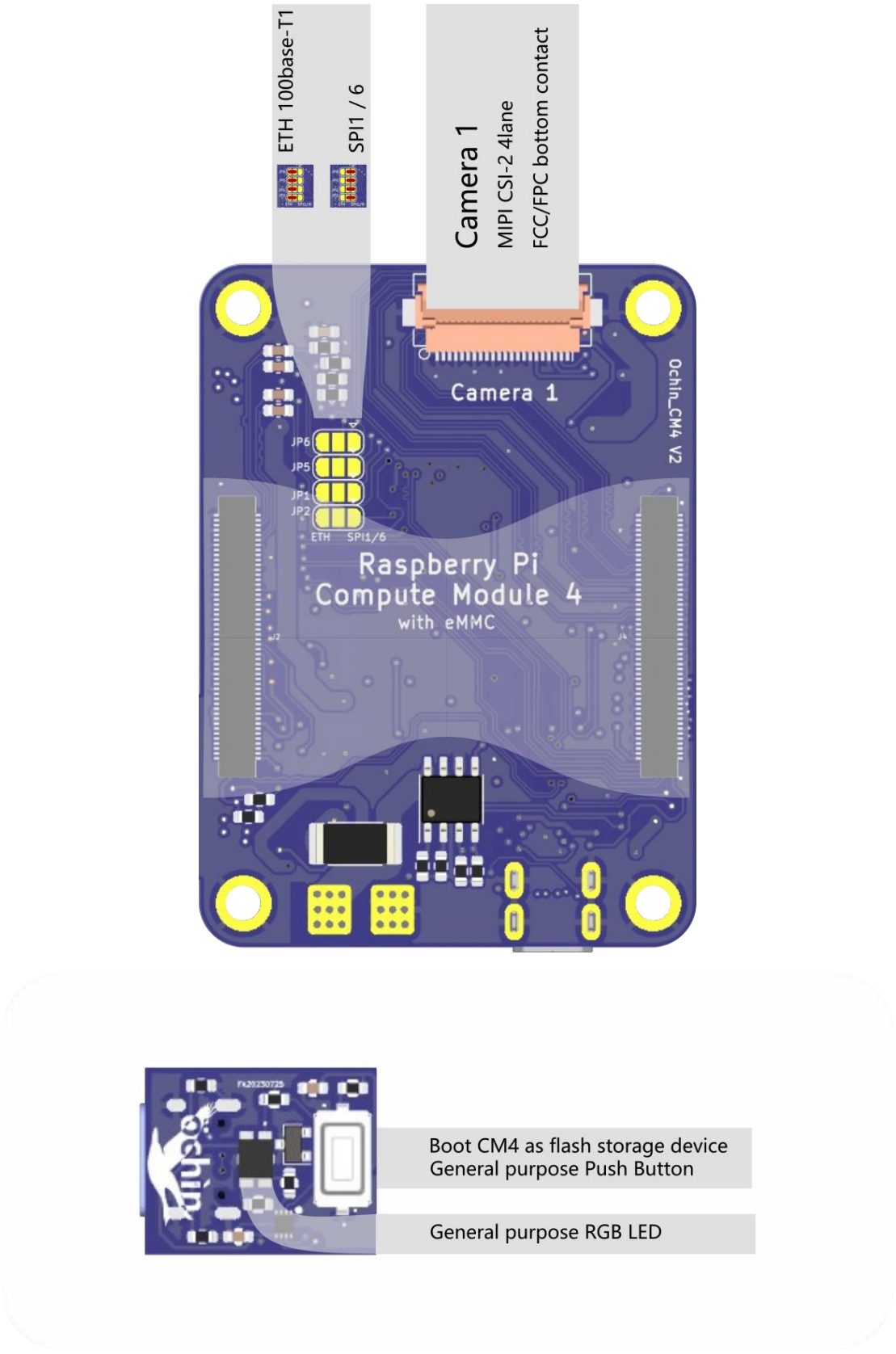
Hardware interfaces

The öchìn CM4 board provides the following interfaces on the side's connectors:

- 4x USB 2.0 480Mbps (4x SM04B-GHS-TB(LF)(SN) connectors)
- 1x USB Type-C (for flashing eMMC)
- 2x CSI camera (2x FH12-22S-0.5SH (55) connectors)
- I2C1 (SM04B-GHS-TB(LF)(SN) connector)
- SPI1 / 6 (SM06B-GHS-TB(LF)(SN) connector)
- UART0 / 1 + Video Out (SM06B-GHS-TB(LF)(SN) connector)
- UART4 / UART5 (SM06B-GHS-TB(LF)(SN) connector)
- 1x microHDMI
- 2x general purpose LEDs
- 1x RGB LED on external tiny board
- 1x general purpose button on external tiny board


Current Limiter bypass
USB3
VBUS
USB3N
USB3P
GND
USB2
VBUS
USB2N
USB2P
GND
USB1
VBUS
USB1N
USB1P
GND
USB4
VBUS
USB4N
USB4P
GND
USART4/5
GND
+5V
TX5
RX5
TX4
RX4
SPI1/6
GND
+5V
MOSI / ETH1P
MISO / ETH1N
SCK / ETH0P
CE / ETH0N
I2C1
+5V
SDA1
SCL1
GND
UART0/1
VideoOut
RX0/1
TX0/1
Video Out
+5V
EN
GND
Camera 0
MIPI CSI-2 2lane
FCC/FPC bottom contact
USB ext
GND
VOTG
D-
D+
nRPiboot
+3V3
SCL1
SDA1
Ext. Board Connector
GND GND RTs4 CTs4 RX0/1 Vin+ Vin+
GND VideoO TX4 RX4 TX0/1 Vin+ Vin+
microHDMI
digital video output
General purpose LED on GPIO24
General purpose LED on GPIO23
Power Input
DC supply from 7.5V to 28V
LIPO battery from 2S to 6S
USB ext
GND
VOTG
D-
D+
nRPiboot
+3V3
SCL1
SDA1
USB Type-C
used to flash the eMMC

Top view


ETH 100base-T1
SPI1 / 6
Camera 1
MIPI CSI-2 4lane
FCC/FPC bottom contact
Camera 1
Ochin_CM4 V2
JP6
JP5
JP1
JP2
ETH SPI1/6
Raspberry Pi
Compute Module 4
with eMMC
Boot CM4 as flash storage device
General purpose Push Button
General purpose RGB LED

Bot view

Tools for tests

1. ochin CM4 carrier board
2. Raspberry Pi CM4 module with eMMC
3. Power Supply 0-30Vdc
4. USB cable Type-C
5. JST GHR-04V-S with cables
6. JST GHR-06V-S with cables
7. USART to USB adapter
8. USB Type-A female adapter
9. RaspiCam for Raspberry Pi Zero (FPC-22 pin flat cable)
10. USB memory stick

To test the functionality of the USBs and other interfaces you need some JST GH series connectors:

- USB1-4 : JST GHR-04V-S
- I2C1 : JST GHR-04V-S
- UART0/1 Video Out : JST GHR-06V-S
- UART4/5 : JST GHR-06V-S
- SPI1/6 : JST GHR-06V-S
- USBxFlash: JST GHR-08-S <-> JST GHR-08-S

Test Plan

The test plan is thought to verify the design and the fabrication of the board.

The first step is to verify the power supply circuit:

1. Test with forward voltage and +5Vdc regulator
2. Test of the inverted voltage protection

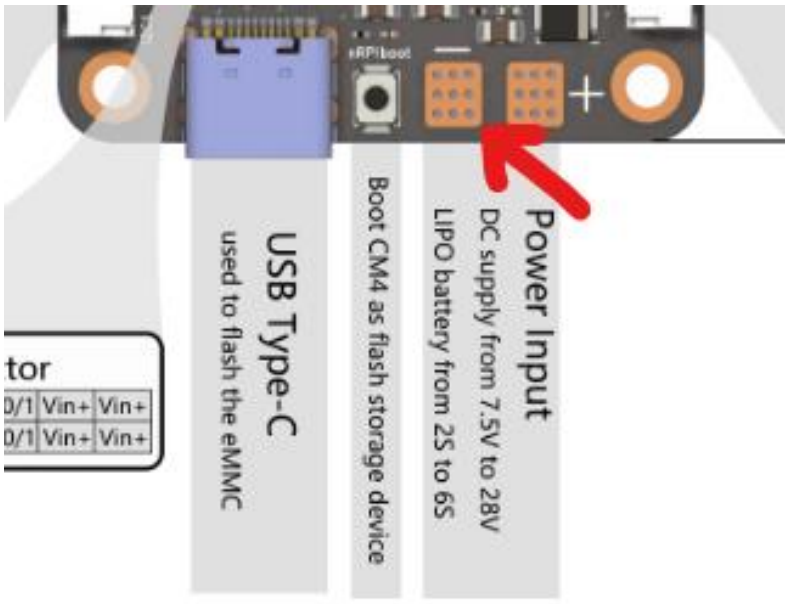
Once the supply circuit is tested, we must flash the eMMC of the CM4 module and verify that the other peripherals are working fine.

1. Test the UART0, used by default as debug terminal.
2. Redirect terminal on UART4, UART5 and test the interfaces

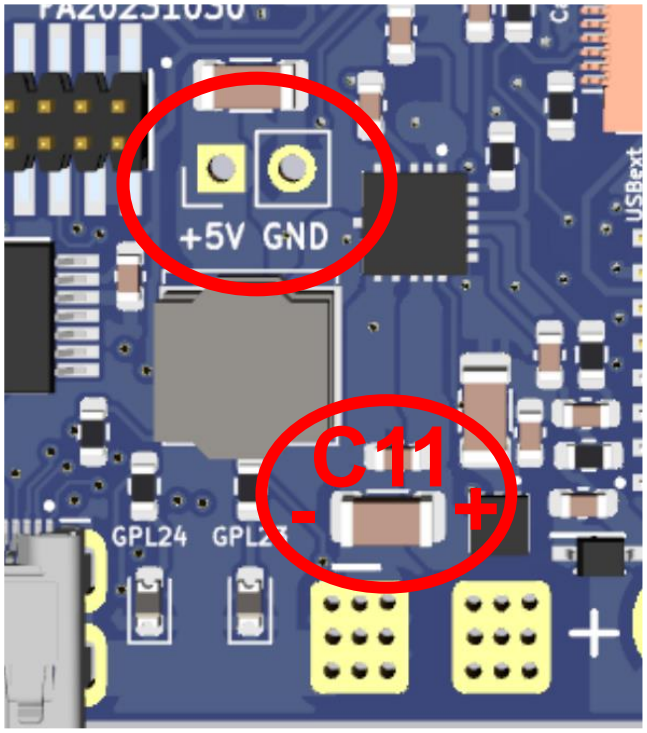
3. Verify that the USBs works fine.
4. Verify that the MIPI Camera0 and Camera1 works fine.

1) Power Supply circuits. Forward Voltage circuit Test:

1. In this phase the CM4 module **SHOULD NOT BE MOUNTED** on the ochin carrier board
2. Apply a voltage between 7,5Vdc and 25Vdc on the power pad. The positive pole is the one on the right and negative pole is on the left (top view).



3. Verify the voltage on the source pin of the SIA471DJ mosfet, placing the multimeter across the C11 capacitor. The voltage passes through the mosfet and we should find on C11 the same voltage of Vin.
4. If the voltage on C11 is fine, it is possible to verify the +5V on the output of the switching regulator. To check the +5V we can measure the voltage across +5V connector over the inductor L1. The measured voltage should be +5Vdc.

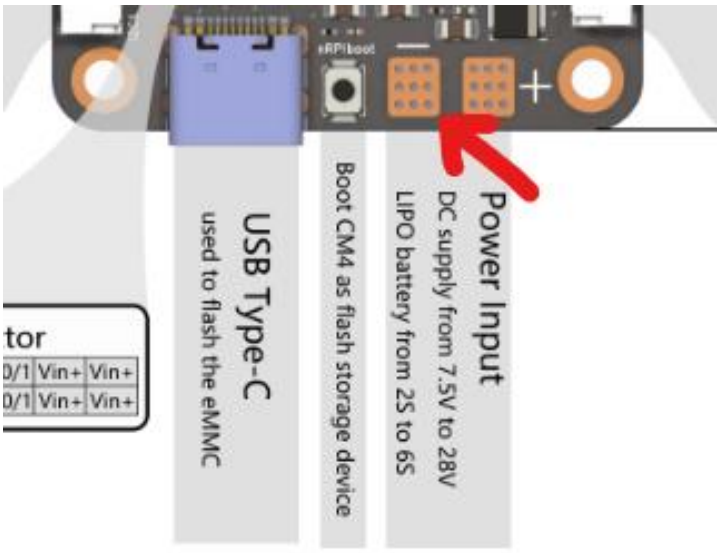


2) Power Supply circuits. Inversion polarity protection circuit Test:

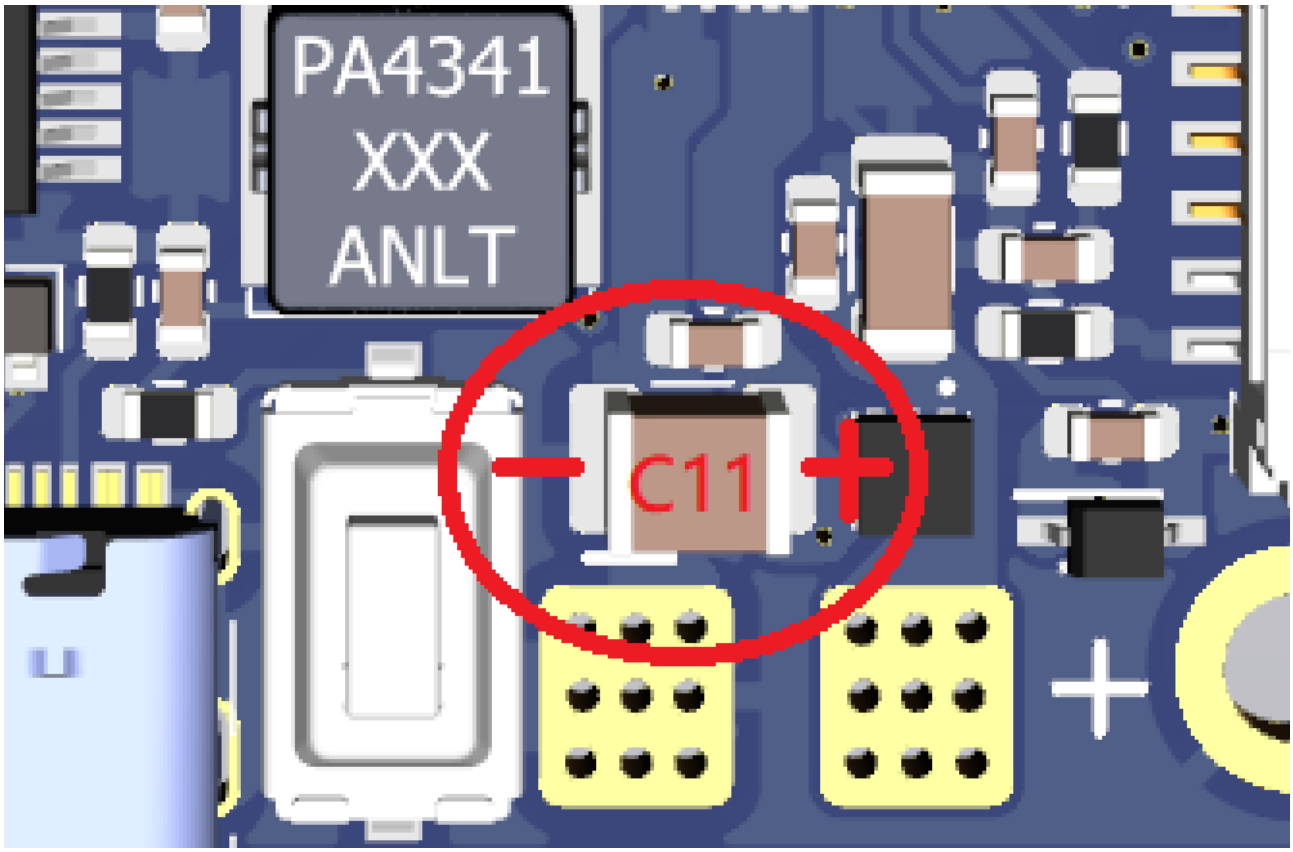
The ochin board is equipped with a reverse protection circuit. The power supply is blocked in case of inversion thanks to the U1-SIA471DJ mosfet. In the case that the Vin is mistakenly reversed, the gate of the mosfet-p opens the junction, preventing current to pass and damage the board.

To verify its operation, it is sufficient to arbitrarily invert the polarity of Vin and therefore check that the voltage across C11 is equal to zero.

1. To prevent the CM4 module from being damaged in the event of a protection malfunction, it is advisable to **perform the test with the CM4 module NOT mounted.**
2. Apply a reverse voltage between 7,5Vdc and 25Vdc on the power pad. The positive pole is the one on the left and negative pole is on the right (top view).



Verify the “Zero Volt” present on the source of the SIA471DJ mosfet, placing the multimeter across the C11 capacitor.



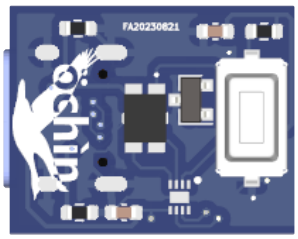
Power up the CM4 Module

The procedure to flash the CM4 module is well explained in the Raspberry Pi Documentation:

https://www.raspberrypi.org/documentation/hardware/computemodule/cm-emmc-flashing.md

The procedure it's straightforward, what you need to do in synthesis is the following:

1. Power up the board with boot configuration as "mass storage device".
    - To do so you need to power off the board (no Vin connected)
    - Connect the board to your PC using the USB Type-C port
    - Press the "nRPiboot" button and, keeping it pressed power up the board providing Vin (the USBC does not power up the board).



2. Run the "RPiboot" software, downloaded from the Raspberry Pi website (github). After the software starts, the PC will see the partition of the eMMC like if it would an SD into the SD-card reader.
3. Flash the eMMC as you normally do when you want to flash an image to the SD card. You could use Win32DiskImager on Windows or "dd" command on Linux.

UARTs test

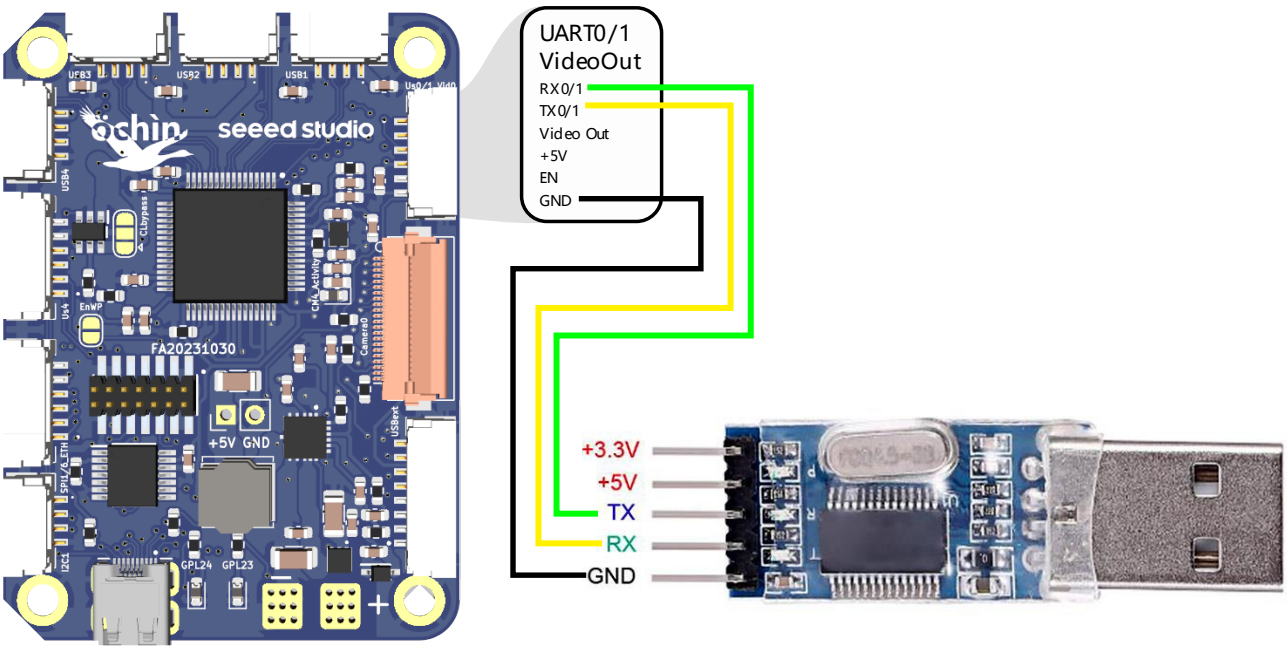
To have the terminal output on the UART0 port, we must disable the BT4 module and enable the UART0 in the "config.txt" file, adding the following line at the end of the file:

dtoverlay=disable-bt

BOOT_UART=1

The “config.txt” file could be found within the “boot” partition. To be noted is that the “boot” partition is also accessible when you connect the CM4 to the PC booted in MSD mode, since it is formatted as FAT32.

To connect the UART0 to the pc and connect the debug terminal, we must use an USB-UART adapter to the UART0:



To interface to the CM4, a serial terminal software like Putty or Teraterm is needed.

To test the other UART interfaces we need to enable it from the “config.txt” file:

dtoverlay=uart3, # without flow control pins

dtoverlay=uart4,ctsrts # with flow control pins

dtoverlay=uart5,

And connect them to other UART2USB adapters.

To test the functionality of all the UARTs, it is possible to redirect one UART to another.

$ tty #identify the actual UARTs

$ exec &> >(tee >(cat >&/dev/ttyAMAx) #redirect to ttyAMAx

USBs test

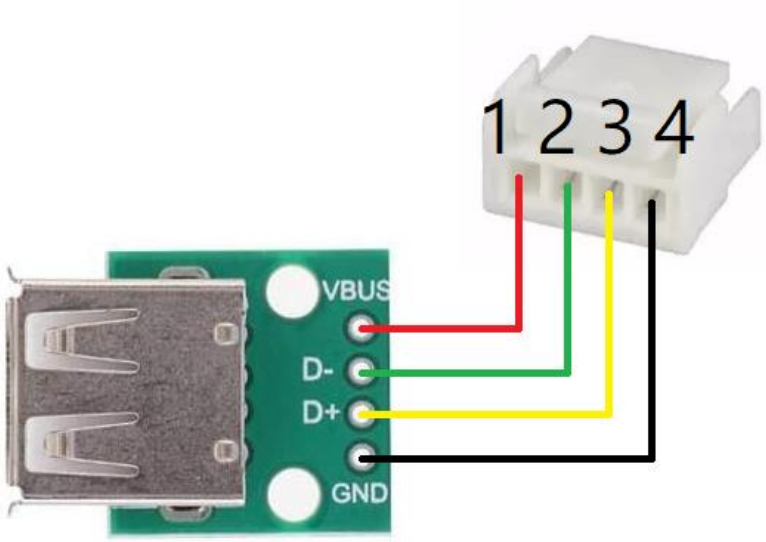
Once you have the system running for the first time, the USB interface will not work because it is disabled by default. To use the USBs, you need to enable it in the "config.txt" file, adding the following line at the end of the file:

dtoverlay=dwc2,dr_mode=host

To enable the USBs1-4, the USB Type-C cable must be disconnected from the board!! Otherwise, the USB hub is powered off.

To test the USBs functionality is possible to make a speed test to an external USB Memory Storage device, for example an USB Stick.

To connect the USB stick to the ochin board in necessary to make an adapter cable using a USB Type-A female connector and a JST GHR-04V-S with cables:



To test the write and read speed to the external device use the following command from the Linux prompt of the raspberry pi cm4.

Go to the external drive location under /media/pi/
First test the write speed:

```
$ dd if=/dev/zero of=/media/usb/output.file bs=20M count=5 oflag=direct
```

at the end you will receive the write speed to the external disk:

```
104857600 bytes (105 MB, 100 MiB) copied, 0.815855 s, 129 MB/s s
```

Run the following command to clear the memory cache:

```
$ sudo sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
```

Then measure the read speed:

```
$ dd if=/media/usb/output.file of=/dev/zero bs=20M count=5 oflag=dsync
104857600 bytes (105 MB, 100 MiB) copied, 0.5172 s, 203 MB/s
```

CSI-2 MIPI Cameras test

To test the CSI camera, a streaming sw is necessary. One of my favorites is MJPG-streamer, use the following commands to install it:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install cmake libjpeg8-dev
git clone https://github.com/jacksonliam/mjpg-streamer
tar xvzf mjpg-streamer.tar.gz
sudo apt-get install libjpeg8-dev
sudo apt-get install imagemagick
cd mjpg-streamer/mjpg-streamer
```

comment out the two #include inside of “mjpg-streamer/utils.c”

//#include <linux/stat.h>

//#include <sys/stat.h>

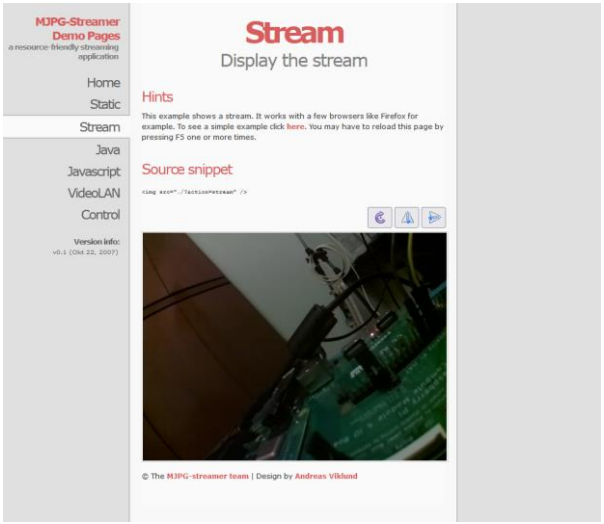
```
make
```

To start streaming from the “Camera1” use the following line:

```
$ ./mjpg_streamer -o "output_http.so -w ./www" -i "input_raspicam.so"
```

To view the stream, open the following URL from a pc within the same LAN/WLAN

http://raspberrypi.local:8080/stream.html



GPL1 and GPL2 leds test

To test the two general purpose LEDs you can use the python script called LEDs_test.py. The program will turn on the two LEDs alternately to test their operation.

RGB led test

To test the operation of the RGB LED mounted on the external board, you can run the python script called KTD2026_test.py. The program will run a loop in which the basic colors (RGB) and some secondary colors will be produced.