1. Overview

AIMEZ-V (part #: AKD5818AIM-EZ-V) is a rapid prototyping kit for AKM's AK5818AIM millimeter-wave radar signal processing IC, which integrates:

- AKM AK5818AIM millimeter-wave radar RFIC + antenna module
- Espressif ESP32-P4 SoC with dual-core RISC-V® processor at 400 MHz and 32MB RAM
- Espressif ESP32-C6-MINI-1 Wi-Fi® 6 (2.4 GHz) / Bluetooth® 5 (LE) connectivity module
- 100BASE-TX Ethernet connectivity with RJ45 connector
- USB type-C connector for power supply and communication with ESP32-P4
- 4 RGB LEDs for visual feedback through Kinetic Technologies KTD2052B I²C LED driver
- JST SH connectors for embedded use and debug

in a compact 29.5 (height) x 51 / 72 / 108 (width) mm 4-layer FR-4 board. Board width depends on desired functions / connectivity: 51 mm when only UART/I²C is necessary, 72 mm with Wi-Fi®/BLE or USB-C, and 108 mm with Ethernet.

This document describes hardware configuration and flashing method.
All supporting software, including the firmware and GUI, can be downloaded from this link.

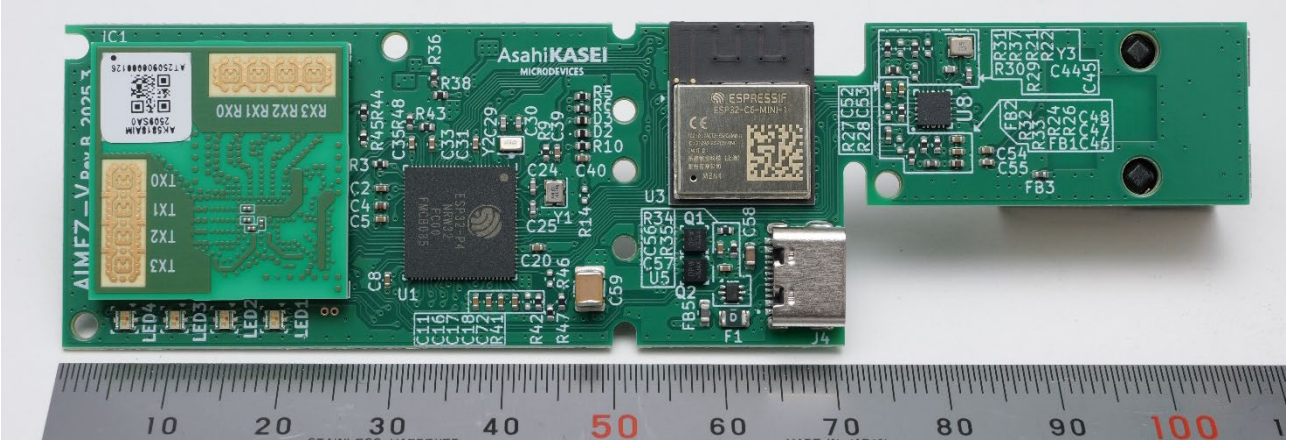


Fig.1.3.1.1. Top view

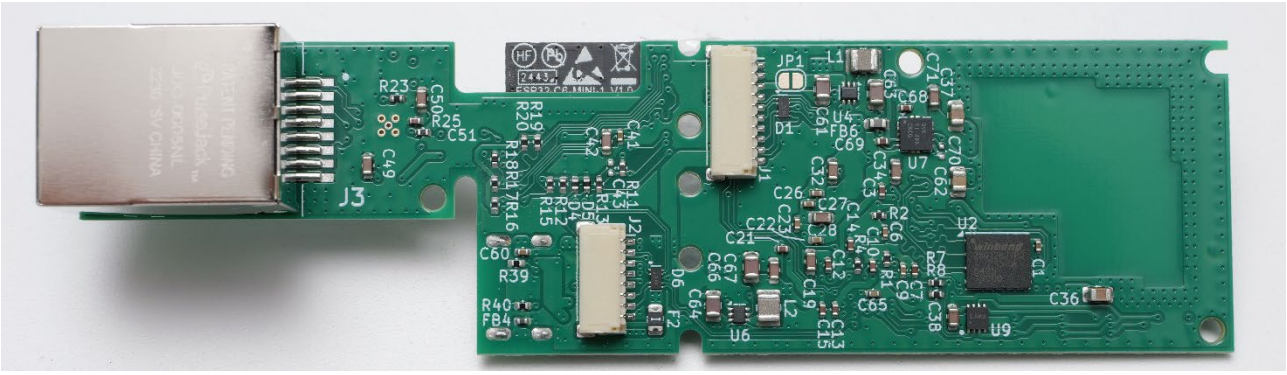


Fig. 1.3.1.2. Bottom view

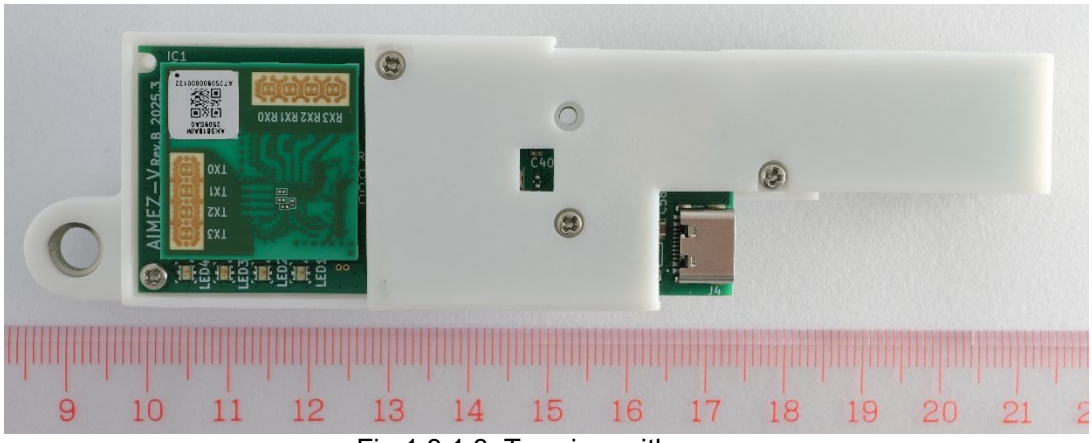


Fig.1.3.1.3. Top view with case

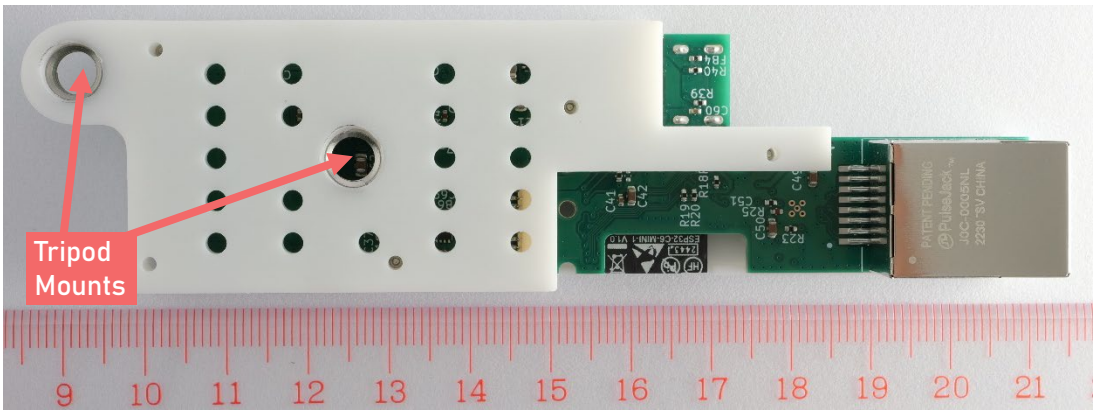


Fig.1.3.1.4. Bottom view with case

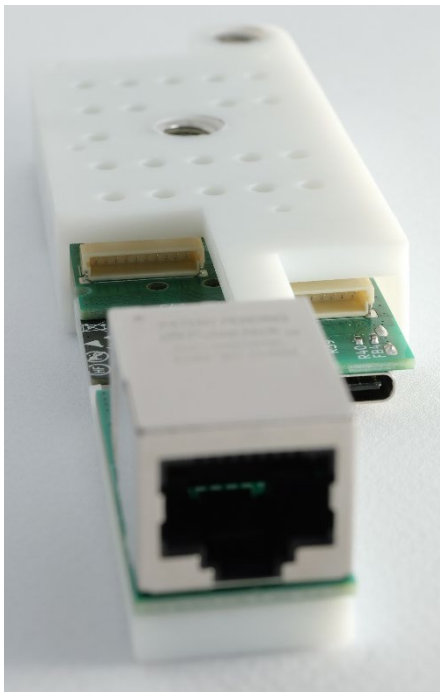


Fig.1.3.1.5. Side view with case

2. Table of Contents

3. Hardware

3.1. Overview

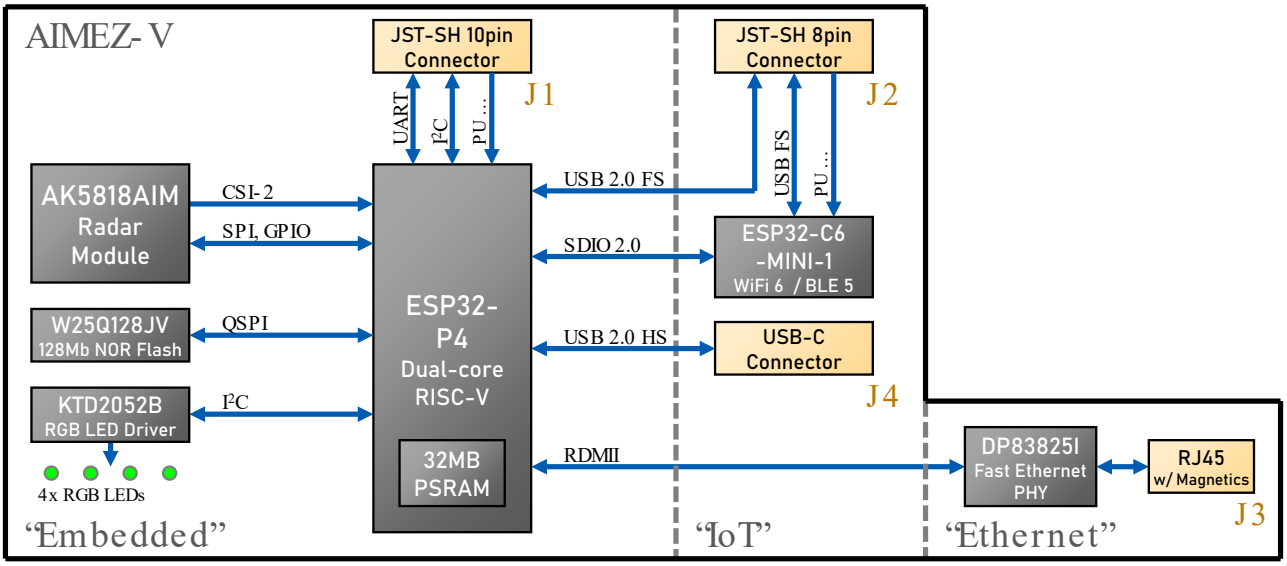


Fig. 3.1.1. Block diagram

Fig. 3.1.1 shows the block diagram of AIMEZ-V board. This board incorporates AK5818AIM millimeter-wave radar module and ESP32-P4 SoC as a processor. To support capturing and processing raw ADC data in the SoC, AK5818AIM radar output and the SoC is connected with mipi® CSI-2 interface.

Depending on user's connectivity requirement, AIMEZ-V board can be cut at the gray dashed line in Fig. 3.1.1, which corresponds to pink dashed line in Fig. 3.1.2. Cut board should look like ones in Fig. 3.1.3. If cut at line "A", or in "Embedded" configuration, users can use UART, I²C interfaces and 4x RGB LEDs. In this configuration, 5V power must be supplied through J1 connector. Please note that USB 2.0 Full-Speed (Serial / JTAG) port of ESP32-P4 is not available in connector J1, so if you need JTAG connectivity during development phase, we recommend to keep the "IoT" part, or connector J2.

If cut at line "B", or in "IoT" configuration, you will gain access to ESP32-C6 Wi-Fi® 6 / Bluetooth® 5 BLE module, USB type-C connector for 5V power supply and data transfer to/from ESP32-P4, and USB signals for flashing / debugging ESP32-P4 and C6 via JST-SH connector J2.

For Ethernet applications, please use the board as-is.

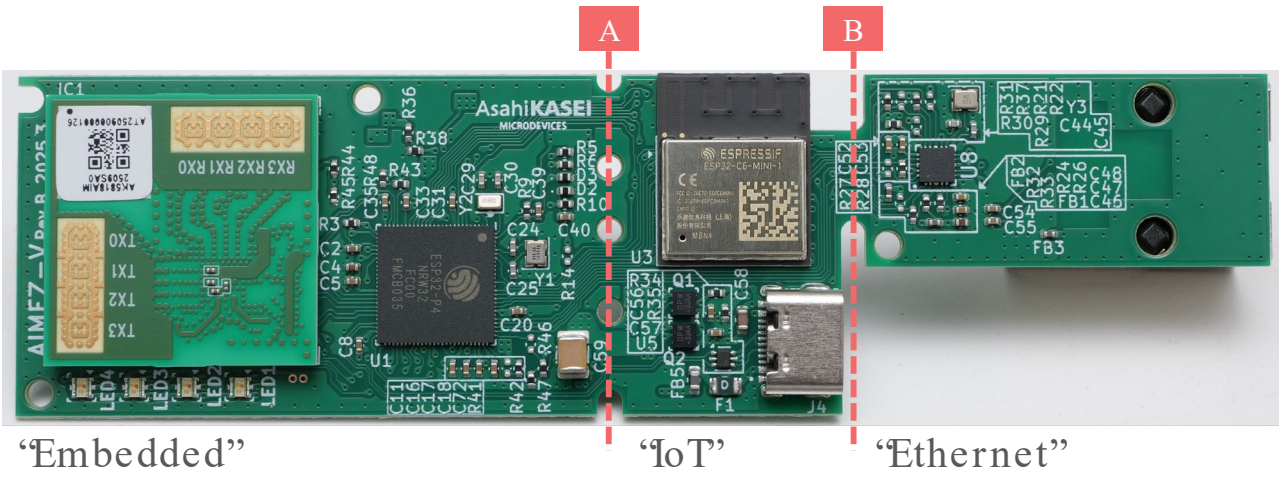
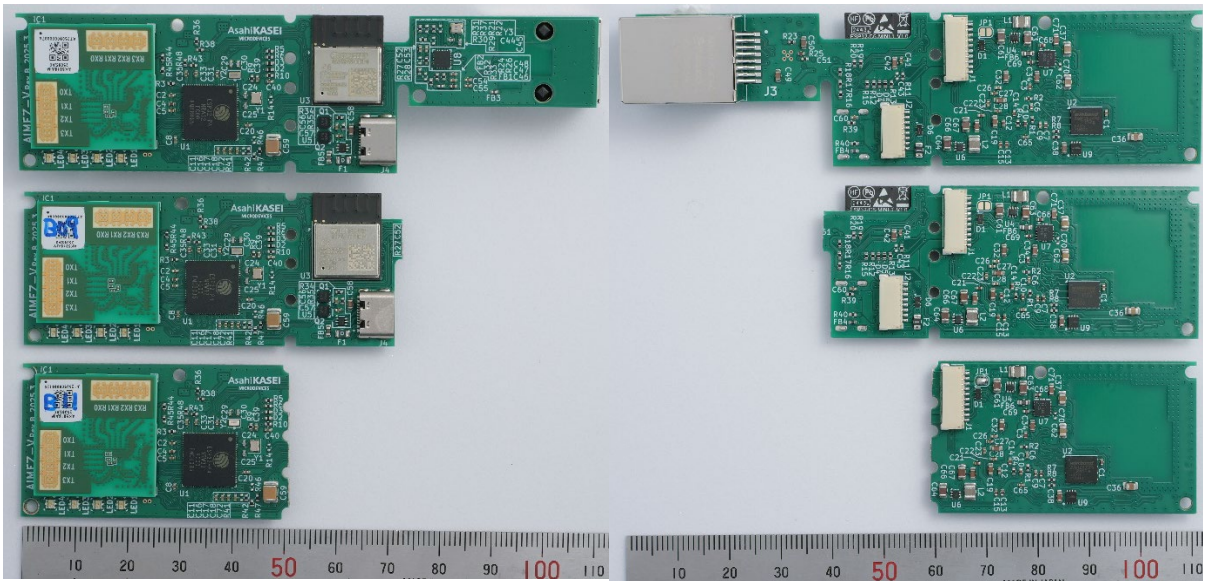


Fig. 3.1.2. Top view with cut line



(a) Top view (b) Bottom view

Fig. 3.1.3. Cut board

3.2. AK5818AIM

(a) upside down (b) sideways (c) ANT_ARRAY=0

Fig. 3.2.1. AK5818AIM placement

AK5818AIM is AKM's 4Tx 4Rx 60-GHz band radar module with integrated antenna. In AIMEZ-V board, if AK5818's RPU (Radar Processing Unit) is used for DoA (Direction of Arrival), as it expects antenna placement shown in Fig. 3.2.1 (c), placement (a) will result in upside down image, and (b) results in sideways image.

AK5818AIM is controlled with SPI access (pins CSN, CDTO, CDTI, and CCLK, 20MHz max.), power-down related pins (PDN and RSTN), automatic run mode pin (EXEC), and manual trigger pin (TRIG). Status or error can be notified through IRQ pin. AK5818AIM radar result can be read through SPI, or CSI-2 interface. Table. 3.2.1-3 shows detailed connections.

CSI-2 connection from AK5818AIM has swapped P/N wiring to ESP32-P4, due to physical pin order. Therefore, PNSWAP_DCLK register of AK5818AIM (at page 0, address 0x0E, bit [6] : users should write data 0x40 to this address 0x0E) should be set.

1.0V power supply to AK5818AIM is controlled by "RF_LDO_E" net at GPIO21 of ESP32-P4. This signal must be high (enabled) before de-asserting PDN and RSTN.

For detailed information on AK5818AIM, please refer to AK5818AIM datasheet.

Table. 3.2.1. CSI-2 Connections

ESP32-P4 Pin #	ESP32-P4 Pin Name	AK5818AIM Pin Name	AK5818AIM Pin #
42	CSI_DATAN0	D0N	R22
43	CSI_DATAP0	D0P	T22
44	CSI_CLKP	DCLKN *1	U22
45	CSI_CLKN	DCLKP *1	V22
46	CSI_DATAN1	D1N	W22
47	CSI_DATAP1	D1P	X22

*1: Need to swap P/N inside AK5818AIM with register PNSWAP_DCLK = 1.

Table. 3.2.2. SPI / GPIO Connections

ESP32-P4 Pin #	ESP32-P4 Pin Name	AK5818AIM Pin Name	AK5818AIM Pin #
55	GPIO26	TRIGI	AA21
56	GPIO27	CSN	AA20
57	GPIO28	CDTO	AA19
58	GPIO29	CDTI	AA18
60	GPIO30	CCLK	AA17
61	GPIO31	IRQ	AA16
63	GPIO32	EXEC	AA15
64	GPIO33	RSTN	AA14
65	GPIO34	PDN	AA13

Table. 3.2.3. 1.0V LDO Connections

ESP32-P4 Pin #	ESP32-P4 Pin Name	TPS74801TDRCRQ1 Pin Name	Pin #
23	GPIO21	EN	5

1.0V LDO EN pin is pulled down with 100k-Ohm resistor R48.

3.3. I²C RGB LED Driver (KTD2052B)

AIMEZ-V integrates I²C RGB LED Driver KTD2052B and 4 RGB LEDs (19-337C/RSBHGHC-A88/4T) for easy implementation of visual feedback. Please refer to KTD2052 datasheet at https://www.kinet-ic.com/ktd2052/ . LED1-4 correspond to LED1-4 pins of the IC. Since KTD2052 “B” variant is used, 4 LEDs blinks in blue after being powered up.

Table. 3.3.1. I2C Connections to KTD2052B

ESP32-P4 Pin #	ESP32-P4 Pin Name	KTD2052B Pin Name	KTD2052B Pin #
24	GPIO22	SDA	2
25	GPIO23	SCL	1

I²C signals are pulled up to 3.3V via 2.2k-Ohm resistors R7 and R8.

3.4. 10-pin Expansion Connector J1

On the bottom side of AIMEZ-V board, JST SH 10-pin expansion connector (SM10B-SRSS-TB) is available for embedded use. This connector provides 5V power supply input, UART, I²C interface, and pins necessary for putting ESP32-P4 into download mode.

Please note that if you use a “straight” cable, pin 1 of AIMEZ-V will be connected to pin 10 of another board.

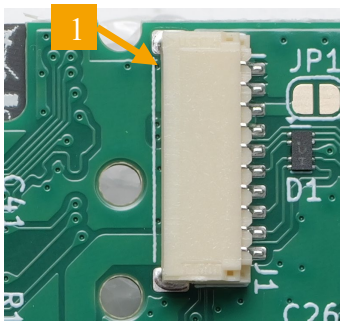


Fig. 3.4.1. J1 connector

Table. 3.4.1. J1 connector pinout

ESP32-P4 Pin #	ESP32-P4 Pin Name	J1 Pin Name	J1 Pin #
-	-	GND	1
-	-	GND	2
-	-	VDD5_J1	3
-	-	VDD5_J1	4
8	GPIO8	EXT_SCL	5
7	GPIO7	EXT_SDA	6
103	CHIP_PU	P4_RSTN	7
70	GPIO38	P4_0_RXD	8
69	GPIO37	P4_0_TXD	9
66	GPIO35 (Boot Mode)	P4_35_PROG_L	10

P4_RSTN and P4_35_PROG_L are pulled up to 3.3V with 100k-Ohm resistors R9 and R10, respectively. I^2C signals are pulled up to 3.3V with 2.2k-Ohm resistors R5 and R6.

If you prefer SPI signaling over UART / I^2C, these 4 pins can be configured to form a SPI interface. In this case, please remove the pull-up resistors R5 and R6. If AIMEZ-V-PLUS board is used in this configuration, please remove jumpers on J5 and J6 of AIMEZ-V-PLUS board.

To turn ESP32-P4 into download mode, please pull P4_35_PROG_L down before de-asserting (rising) P4_RSTN, and de-assert (release / rise) P4_35_PROG_L after P4_RSTN rise. In this mode, ESP32-P4 can be flashed via UART signals P4_0_RXD/TXD. Please see Section.5 “Writing Firmware” for details.

3.5. Wireless Connectivity Module (ESP32-C6-MINI-1)

AIMEZ-V incorporates ESP32-C6-MINI-1 Wi-Fi 6 (2.4GHz) / Bluetooth 5 BLE module in “IoT” region of the board, connected to ESP32-P4 via SDIO interface. ESP32-C6-MINI-1 is shipped with ESP-Hosted-MCU firmware. For details, please refer to https://github.com/espressif/esp-hosted-mcu .

Table. 3.5.1. Connections between ESP32-C6-MINI-1 and ESP32-P4

ESP32-P4 Pin #	ESP32-P4 Pin Name	ESP32-C6-MINI-1 Pin Name	Pin #
15	GPIO14	GPIO18 / SDIO_CMD	24
16	GPIO15	GPIO19 / SDIO_CLK	25
17	GPIO16	GPIO20 / SDIO_DATA0	26
18	GPIO17	GPIO21 / SDIO_DATA1	27
19	GPIO18	GPIO22 / SDIO_DATA2	28
20	GPIO19	GPIO23 / SDIO_DATA3	29
98	GPIO54	EN	8
66	GPIO35 *1	GPIO9 (Boot Mode)	23
6	GPIO6	GPIO2	5

*1 Disconnected without R14. Please mount 0-Ohm jumper at R14 to connect.

SDIO_* signals are pulled up to 3.3V via 10k-Ohm resistors R15-20. EN signal is also pulled up to 3.3V via 100k-Ohm resistor R11.

GPIO8 / 9 of ESP32-C6-MINI-1 are pulled up to 3.3V via 10k-Ohm resistors R12-13, reducing the need to change GPIO8 level to put the module into download mode. To put the module into download mode, pull GPIO9 down before asserting (setting high) EN pin, and release GPIO9 after asserting EN pin. In this mode, ESP32-C6-MINI-1 can be flashed via USB signals available in J2 connector. Please see Section.5 “Writing Firmware” for details.

3.6. USB Type-C Connector J4

AIMEZ-V has a USB type-C connector (J4) in “IoT” region of the board for 5V power supply and data communication with ESP32-P4 SoC. For power supply configuration, please see Section 3.9. Since this port is connected to USB 2.0 OTG High-Speed port of ESP32-P4 as shown in Table. 3.6.1, it can be used as a DFU (Device Firmware Update) port. Please see Section 5.3 “Writing in DFU Mode” for details.

If this port is used as an OTG port, 5V power cannot be supplied from AIMEZ-V board as is, as there is an automatic 5V selector (see Fig. 3.9.1). One possibility is to supply 5V power from connector J1 and jump the top of C59 (selected 5V) and bottom of C58 (VBUS) as shown in Fig. 3.6.1.

Table. 3.6.1. Connection between ESP32-P4 and USB-C connector

ESP32-P4 Pin #	ESP32-P4 Pin Name	J4 Pin Name	J4 Pin #
50	USB_DP	DP1, DP2	A6, B6
49	USB_DN	DN1, DN2	A7, B7

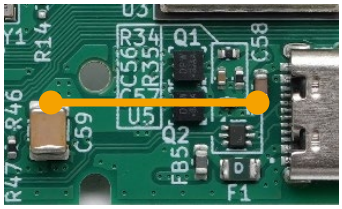


Fig. 3.6.1 Bypassing 5V auto-select circuit

3.7. 8-pin Expansion Connector J2

On the bottom side of AIMEZ-V board, in “IoT” region, JST SH 8-pin expansion connector (SM8B-SRSS-TB) is available for debugging and flashing ESP32-C6-MINI-1 and ESP32-P4. This connector provides pins necessary for putting ESP32-C6-MINI-1 into download mode. Also, pin 6/7 can be configured to use USB 2.0 OTG Full-Speed instead of USB Serial/JTAG of ESP32-P4, external USB device can be connected. Different from connector J4, this connector supports 5V output for external USB device.

Please note that if you use a “straight” cable, pin 1 of AIMEZ-V will be connected to pin 8 of another board.

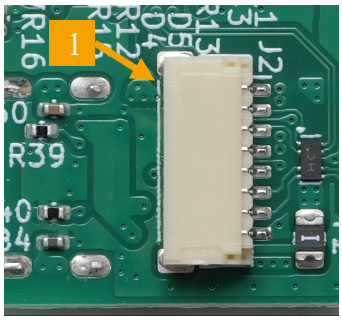


Fig. 3.7.1. J2 JST-SH 8-pin connector

Table. 3.7.1. J2 connector pinout

ESP32-P4 Pin #	ESP32-P4 Pin Name	J2 Pin Name	J2 Pin #
[C6] 23	[C6] GPIO9	C6_9_PROG_L	1
[C6] 18	[C6] GPIO13/USB_D+	C6_USBFS_+	2
[C6] 17	[C6] GPIO12/USB_D-	C6_USBFS_-	3
[C6] 8	[C6] EN	C6_EN	4
-	-	VDD5EXT	5
53	GPIO25/USB_D+	P4_USBFS_+	6
52	GPIO24/USB_D-	P4_USBFS_-	7
-	-	GND	8

3.8. Ethernet PHY and Connector

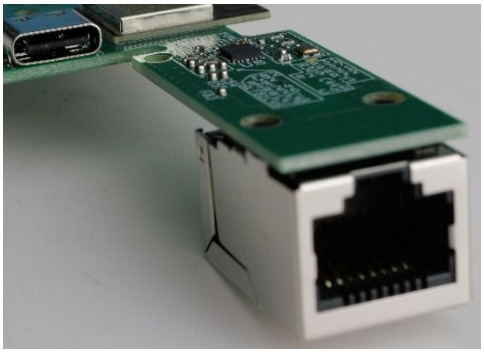


Fig. 3.8.1. RJ45 connector and PHY

In “Ethernet” region of the board, Fast Ethernet (100MbE) PHY DP83825I and RJ45 connector with integrated magnetics J0C-0005NLT are mounted. DP83825I PHY IC is intended to run in RMII Master Mode, providing 50MHz Clock from its 50MHzOut pin to ESP32-P4. Table. 3.8.1 shows the connection between ESP32-P4 and DP83825I, and Table. 3.8.2 shows the bootstrap pins configuration.

Table. 3.8.1. ESP32-P4 RMII connection

ESP32-P4 Pin #	ESP32-P4 Pin Name	DP83825I Pin Name	DP83825I Pin #
92	GPIO49	MDIO	15
90	GPIO48	MDC	16
89	GPIO47	RX_D1	17
88	GPIO46	RX_D0	18
87	GPIO45	CRS_DV	20
81	GPIO40	TX_EN	1
82	GPIO41	TX_D0	23
83	GPIO42	TX_D1	24
86	GPIO44	50MHzOut	2
80	GPIO39	RST_N	5

MDC pin is pulled up to 3.3V with 2.2k-Ohm resistor R21.

Table. 3.8.2. DP83825I Bootstrap

Strap Name	DP83825I Pin Name	Pin #	Function at 0, 2.49k-Ohm Location	Function at 1, 2.49k-Ohm Location
PhyAdd[1]	CRS_DV	20	PHY_ADDR[1] = 0 R28 (default)	PHY_ADDR[1] = 1 R27
PhyAdd[0]	RX_D0	18	PHY_ADDR[0] = 0 R30 (default)	PHY_ADDR[0] = 1 R29
Slave	RX_D1	17	RMII Master Mode R31 (default)	RMII Slave Mode R37
RX_DV_En	50MHzOut	2	Pin 20 = CRS_DV R33 (default)	Pin 20 = RX_DV R32
A-MDIX	RX_ER	22	Auto MDIX Enable R25 (default)	Auto MDIX Disable n/a
ANeg_Dis	LED0	4	Auto Negotiation Enable R26 (default)	Auto Negotiation Disable n/a

3.9. Power supply

Power supply network of AIMEZ-V board is shown in Fig. 3.9.1. This board can be powered from 1) J1 10-pin connector or 2) J4 USB-C connector. Both connector J1 and J2 take 5V input, and whichever has the higher voltage is selected in automatic selection circuitry located in “IoT” region. If you are using only “Embedded” part and cut the board, you need to solder-jump jumper JP1 on the bottom side, as shown in Fig. 3.9.2, since the selection circuitry is only present in “IoT” part.

This selected 5V input is supplied to 3.3V Buck converter U4 (TPS62A02A), which powers most devices through a ferrite bead (FB6). AK5818AIM will use 3.3V before the bead. This 5V power is also supplied to J2 connector, for powering external USB device in OTG mode.

From 3.3V, ESP32-P4 core 1.2V is generated by Buck converter U6 (TLV62569A), which is controlled by EN_DCDC pin of ESP32-P4. This 1.2V is also used as a source for 1.0V LDO U7 (TPS74801). This 1.0V supplies AK5818AIM, and enabled by RF_LDO_E signal, which is GPIO21 of ESP32_P4 SoC. Please write high value before de-asserting PDN pin of AK5818AIM to allow necessary power to be supplied for the module.

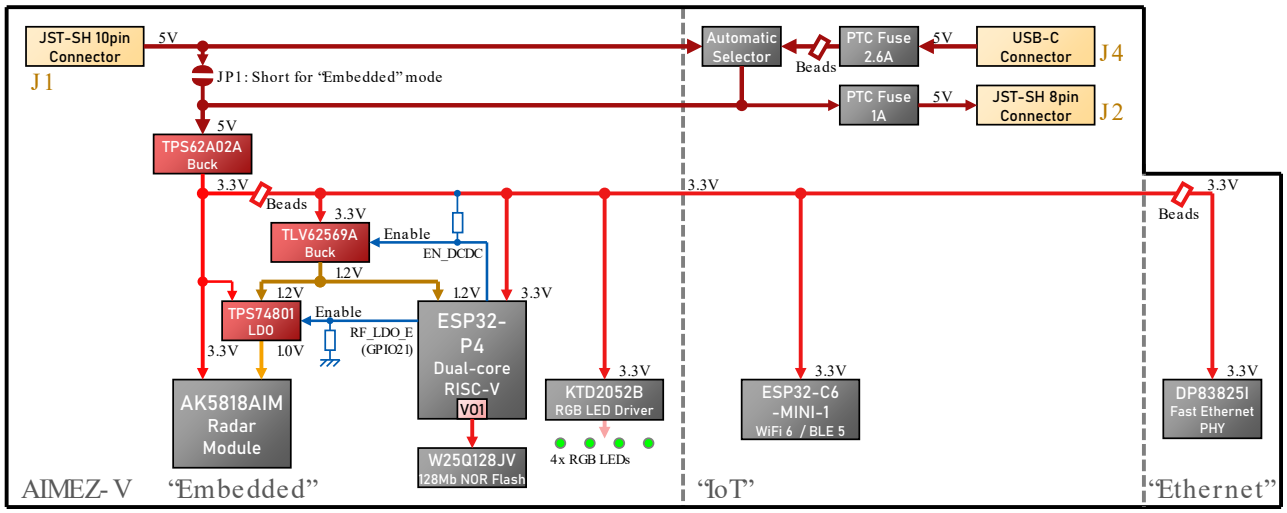
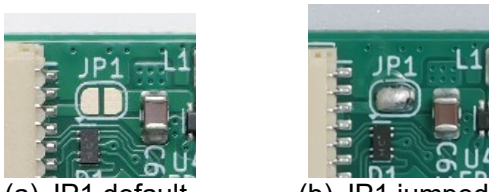


Fig. 3.9.1. Power supply configuration



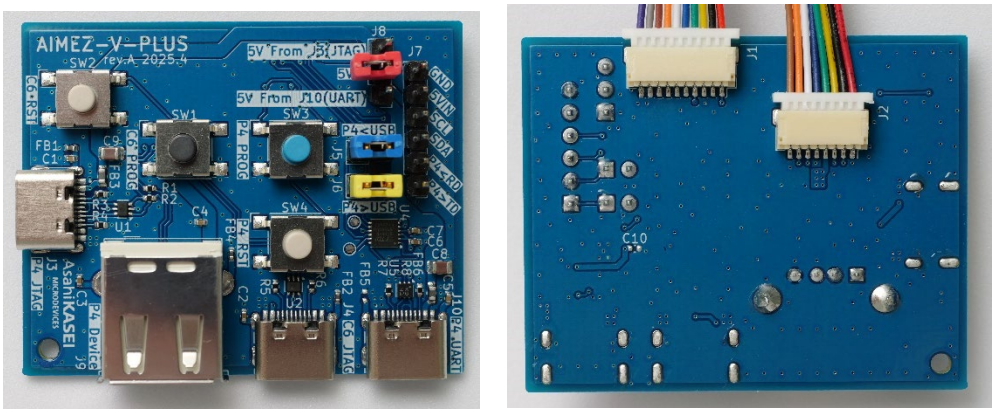
(a) JP1 default (b) JP1 jumped

Fig. 3.9.2. JP1 for “Embedded” use

3.10. AIMEZ-V-PLUS expansion board

For easy integration with other MCUs, and for flashing and debugging, “AIMEZ-V-PLUS” expansion board is available (Fig. 3.10.1). This board offers:

- A 6-pin 2.54mm pitch header for power supply, UART, and I²C (or SPI)
- USB-UART conversion IC with a USB-C port, connected to UART0 port of ESP32-P4
- USB-C and USB-A port, connected to USB 2.0 OTG Full-Speed / USB-Serial/JTAG port of ESP32-P4
- USB-C port, connected to USB-Serial/JTAG port of ESP32-C6-MINI-1
- Reset and Program switches for ESP32-P4 and ESP32-C6-MINI-1 .



(a) Top view (b) Bottom view

Fig. 3.10.1. Photo of AIMEZ-V-PLUS board

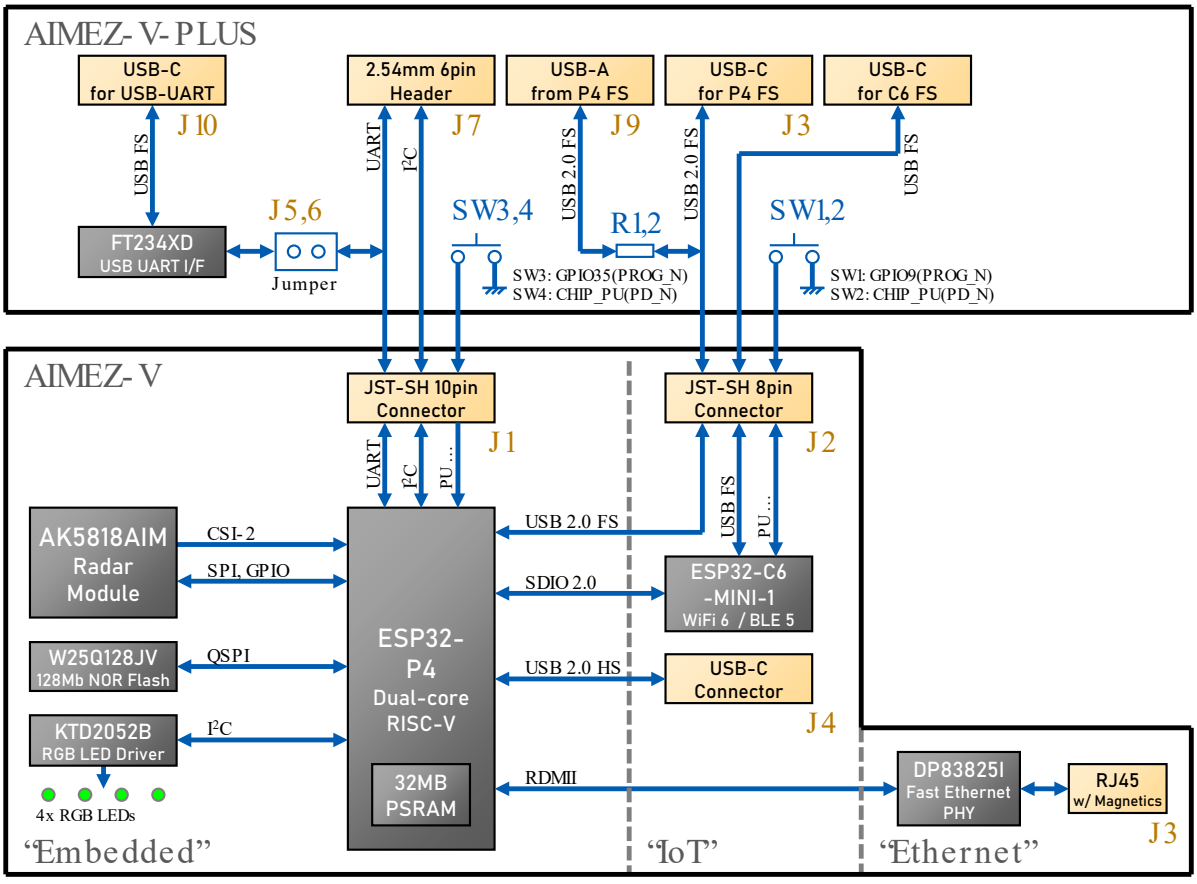
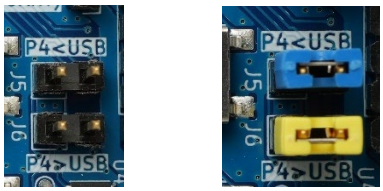


Fig. 3.10.2. Block diagram of AIMEZ-V and AIMEZ-V-PLUS

Please connect both J1 and J2 of AIMEZ-V board with 10-pin and 8-pin JST SH cable to AIMEZ-V-PLUS board. Jumpers J5 and J6 decide if UART signals are also connected to FT234XD USB-UART conversion IC. If you are using UART with J7 header, please leave J5/6 open. If you are using USB-UART and not J7 for UART, please jump J5/6 as shown in Fig. 3.10.3.



(a) For J7 (b) for USB-UART

Fig. 3.10.3. J5 and J6 for USB-UART

Fig. 3.10.4 shows the power network of AIMEZ-V and AIMEZ-V-PLUS. AIMEZ-V-PLUS can feed 5V DC from USB 2.0 OTG / USB-Serial/JTAG port J3 or USB-UART port J10 with jumper J8. This is useful for debugging and flashing, since the whole system can be powered from J3 or J10, without using J4 of AIMEZ-V motherboard. Fig. 3.10.5 shows J8 setup for powering from J10 USB-UART port. Please do not use J8 if power is supplied from J7.

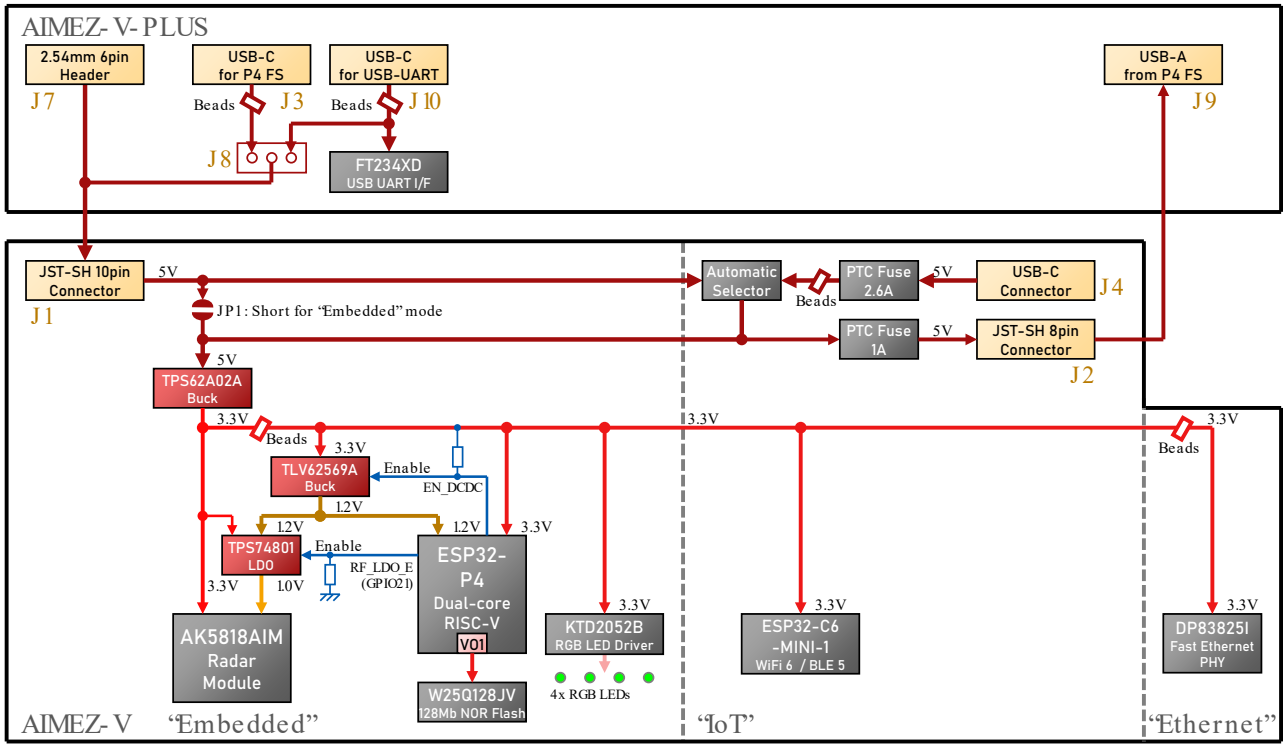


Fig. 3.10.4. Power network for AIMEZ-V and AIMEZ-V-PLUS

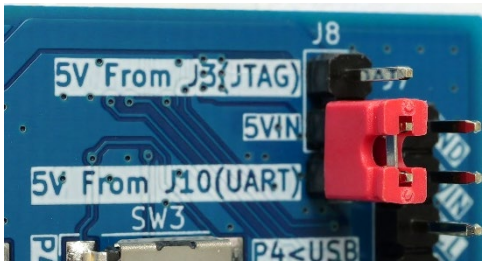


Fig. 3.10.5. Powering AIMEZ-V from USB-UART port

4. Firmware Development

We recommend using ESP-IDF from Espressif to develop the software / firmware on ESP32-P4 device. Their website https://idf.espressif.com/ does not mention ESP32-P4 yet, though their GitHub page https://github.com/espressif/esp-idf states the device is supported from ESP-IDF v5.3. Programming guide for ESP32-P4 is found here:
https://docs.espressif.com/projects/esp-idf/en/latest/esp32p4/index.html .

For installation, we recommend setting up as an extension of Visual Studio Code. Please follow their instruction at https://github.com/espressif/vscode-esp-idf-extension/blob/master/README.md .

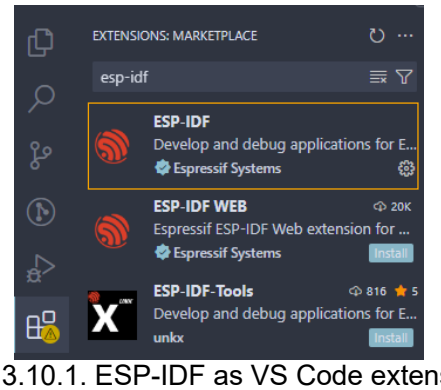


Fig. 3.10.1. ESP-IDF as VS Code extension

5. Flashing Firmware

5.1. Flashing Methods

ESP32-P4 supports the following 3 methods to flash its firmware.

- Via DFU (Device Firmware Update) through USB2.0 High-Speed OTG port (Connector J4)
- Via UART0 port (Connector J1)
- Via USB Serial/JTAG interface (Connector J2)

ESP32-C6-MINI-1 supports flashing via USB Serial/JTAG interface (Connector J2)

Except for flashing non-empty device via USB Serial/JTAG interface, devices must be put in download mode using GPIO35 for ESP32-P4 and GPIO9 for ESP32-C6-MINI-1. Details will be shown in the following sections.

5.2. Flashing Standalone: ESP32-P4 via DFU

Without external circuits connected to J1 and J2 connector, only ESP32-P4 can be flashed. To put the SoC into download mode, GPIO35 must be low at power-up. Please follow the procedure:

1. Unplug PC-side port of USB cable connected to USB type-C connector J4.
2. Jump both terminal of C40 with a metal tweezer.
3. Plug the PC-side port of the USB cable while keeping C40 shorted.
   You can also unplug/plug AIMEZ-V side connector, though it's more difficult to keep tweezer in place in that way.
4. After the board receives power, tweezer can be removed.

Fig. 5.2.1. Turning ESP32-P4 into download mode

Windows users may not be able to connect to DFU instance of ESP32-P4 without additional setting. This is because wrong driver is loaded. Please download Zadig from https://zadig.akeo.ie/ and change the driver of "ESP32-P4 (Interface 2)" to "WinUSB" as shown in Fig. 5.2.2. If you cannot see "Interface 2", ESP32-P4 may not be in download mode. Please retry the procedure 1-4 above. You can also refer to Espressif's manual at
https://docs.espressif.com/projects/esp-idf/en/stable/esp32p4/api-guides/dfu.html .

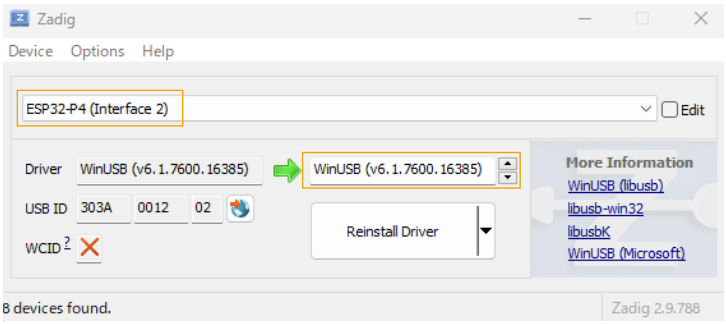


Fig. 5.2.2. Installing correct driver for ESP32-P4 DFU with Zadig

To prepare DFU binary, select "DFU" in "Select Flash Method" located at the bottom of the VS Code window. Select COM port of ESP32-P4.

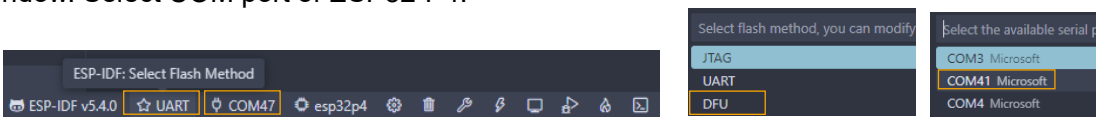


Fig. 5.2.3. Selecting "DFU" as flash method

Unfortunately, ESP-IDF v5.4.0 still thinks ESP32-P4 DFU is not supported. Therefore, please follow the steps:

1. Build your project with Build icon.
2. Open ESP-IDF terminal with icon.
3. Create DFU binary with `idf.py dfu` command. You will have dfu.bin file in build/ folder.

```
> idf.py dfu
/build/dfu.bin" has been written. You may proceed with DFU flashing.
Done
```

4. Flash DFU binary with `idf.py dfu-flash` command.
   Flashing DFU binary may fail once. In that case, please run `idf.py dfu-flash` again.

```
> idf.py dfu-flash
dfu-util 0.11

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2021 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

Opening DFU capable USB device...
Device ID 303a:0012
Run-Time device DFU version 0110
Claiming USB DFU (Run-Time) Interface...
Setting Alternate Interface zero...
Determining device status...
DFU state(0) = appIDLE, status(0) = No error condition is present
Device really in Run-Time Mode, send DFU detach request...
Device will detach and reattach...
Opening DFU USB Device...
Claiming USB DFU Interface...
Setting Alternate Interface #0 ...
Determining device status...
DFU state(2) = dfuIDLE, status(0) = No error condition is present
DFU mode device DFU version 0110
Device returned transfer size 64
Copying data from PC to DFU device
Download        [=========================] 100%       573440 bytes
Download done.
DFU state(2) = dfuIDLE, status(0) = No error condition is present
Done!
error detaching
Done
```

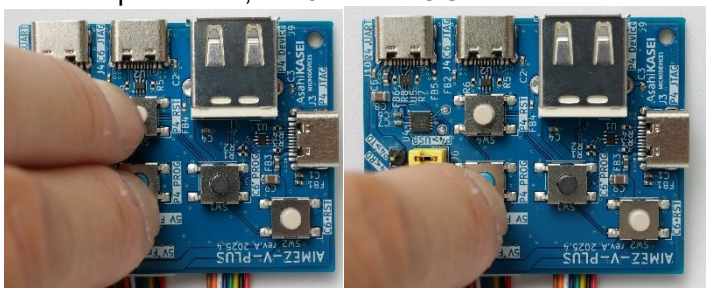
```
dfu-util 0.11

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2021 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

Opening DFU capable USB device...
Device ID 303a:0012
Run-Time device DFU version 0110
Claiming USB DFU Interface...
Setting Alternate Interface #0 ...
Determining device status...
DFU state(2) = dfuIDLE, status(0) = No error condition is present
DFU mode device DFU version 0110
Device returned transfer size 64
Copying data from PC to DFU device
Download        [=========================] 100%       573440 bytes
Download done.
DFU state(2) = dfuIDLE, status(0) = No error condition is present
Done!
Done
```

5. Unplug the USB cable and re-plug (power-cycle). ESP32-P4 will restart into the new firmware.

If you have got only dfu.bin file from developer for flashing, create a dummy project and build. Copy the dfu.bin file into build/ folder, and run `idf.py dfu-flash` as in step 4 above.

5.3. Flashing with AIMEZ-V-PLUS: ESP32-P4 via DFU Mode

With AIMEZ-V-PLUS, tuning ESP32-P4 into download mode is much easier. Supply power to AIMEZ-V board, and simply press SW3 “P4 PROG” and SW4 “P4 RST” buttons at once, and only release SW4 “P4 RST” as shown in . After this procedure, SW3 “P4 PROG” can be released.



(a) Press SW3 and SW4 (b) Release SW4

Fig. 5.3.1. Putting ESP32-P4 into download mode

After ESP32-P4 boots into download mode, please follow the DFU flashing procedure in the previous section. You do not need to unplug the USB cable to power cycle. Just press SW4 “P4 RST” button to restart into new firmware.

5.4. Flashing with AIMEZ-V-PLUS: ESP32-P4 via UART

Please follow the procedure:

1. Connect USB cable to J10 “P4_UART” port of AIMEZ-V-PLUS.
2. Put ESP32-P4 into download mode as shown in Section 5.3.
3. Select “UART” as Flash Method and select the COM port of FTDI FT234XD on AIMEZ-V-PLUS board as shown in Fig. 5.4.1.
4. Build your project with Build icon if not built yet.
5. Flash with Flash icon.
6. Press SW4 “P4 RST” button to restart the device.

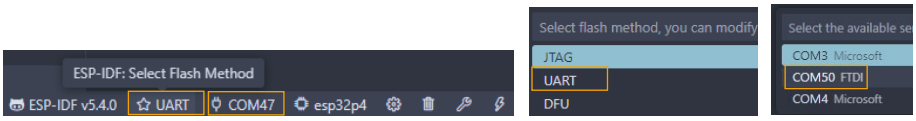


Fig. 5.4.1. Selecting “UART” as flash method

```
esptool.py v4.8.1
Serial port COM50
Connecting.....................
Chip is ESP32-P4 (revision v1.0)
Features: High-Performance MCU
Crystal is 40MHz
MAC: 30:ed:a0:e1:08:63
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Flash will be erased from 0x00002000 to 0x00007fff...
Flash will be erased from 0x00010000 to 0x00092fff...
Flash will be erased from 0x00008000 to 0x00008fff...
SHA digest in image updated
Compressed 23280 bytes to 14293...
Wrote 23280 bytes (14293 compressed) at 0x00002000 in 0.7 seconds (effective 261.8 kbit/s)...
Hash of data verified.
Compressed 535696 bytes to 268759...
Wrote 535696 bytes (268759 compressed) at 0x00010000 in 7.3 seconds (effective 583.8 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 103...
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.1 seconds (effective 383.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

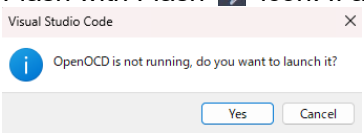
Fig. 5.4.2. UART flashing log example

If flashing fails at `Connecting.....................`, ESP32-P4 may not be in download mode. Please retry the procedure in Section 5.3.

5.5. Flashing with AIMEZ-V-PLUS: ESP32-P4 via USB Serial/JTAG

Please follow the procedure:

1. Connect USB cable to J3 “P4_JTAG” port of AIMEZ-V-PLUS.
2. Put ESP32-P4 into download mode as shown in Section 5.3.
3. Select “JTAG” as Flash Method and select the COM port of USB-Serial/JTAG of ESP32-P4.
4. Build your project with Build icon if not built yet.
5. Flash with Flash icon. If asked to launch OpenOCD, click “Yes”.



6. Press SW4 “P4 RST” button to restart the device.

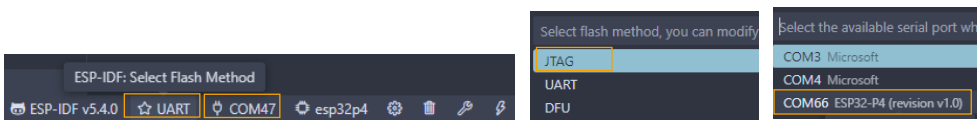


Fig. 5.5.1. Selecting “JTAG” as flash method

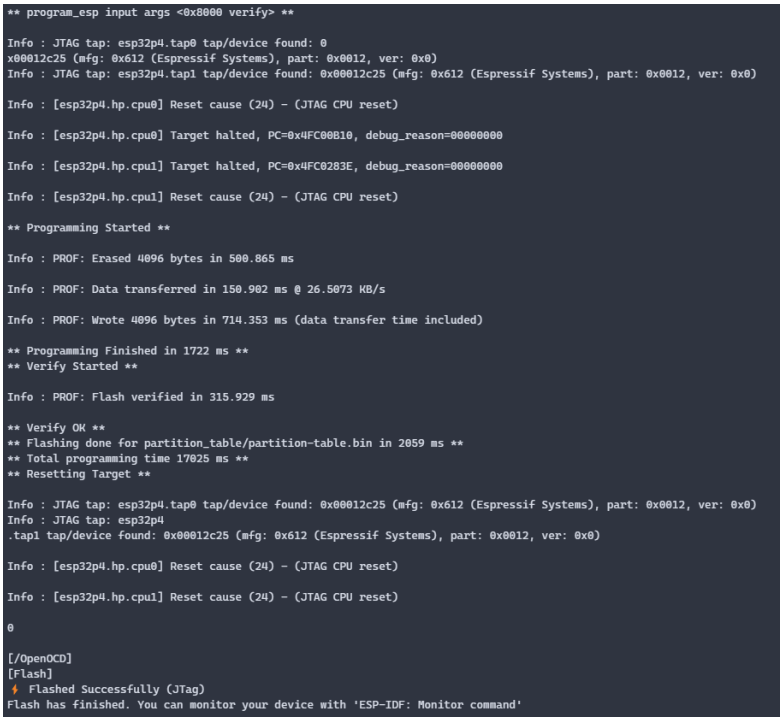


Fig. 5.5.2. JTAG flashing log example

5.6. Flashing with AIMEZ-V-PLUS: ESP32-C6-MINI-1 via USB Serial/JTAG

Please follow the procedure:

1. Connect USB cable to J4 “C6 JTAG” port of AIMEZ-V-PLUS
2. (optional) Put ESP32-C6-MINI-1 into download mode by pressing SW1 “C6 PROG” and SW2 “C6 RST” buttons at once, and only release SW2 “C6 RST”. After that, SW1 “C6 PROG” can be released.
3. Select “JTAG” as Flash Method and select the COM port of ESP32-C6-MINI-1
4. Build your project with Build icon if not built yet.
5. Flash with Flash icon.
6. Press SW2 “C6 RST” button to restart the device.

6. Revision History

Date (Y/M/D)	Revision	Reason	Page	Contents
2025/5/26	00	First edition	−	−
2025/6/19	01	Fixed typo	15	idf_py → idf.py

7. Notes

All information included in this document are provided only to illustrate the operation and application examples of Asahi Kasei Microdevices products. Asahi Kasei Microdevices neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Asahi Kasei Microdevices or any third party with respect to the information in this document. You are fully responsible for use of such information contained in this document in your product design or applications.

AKM ASSUMES NO LIABILITY FOR ANY LOSSES INCURRED BY YOU OR THIRD PARTIES ARISING FROM THE USE OF SUCH INFORMATION IN YOUR PRODUCT DESIGN OR APPLICATIONS.

All names specified in this documentation, including product names and brand names, are the trademarks or registered trademarks of Asahi Kasei Microdevices or the respective companies.