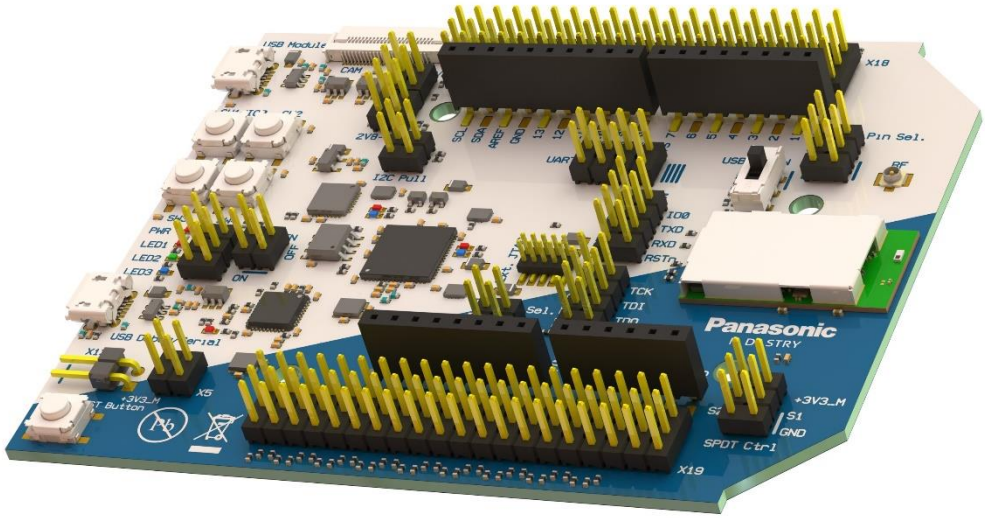
Panasonic
INDUSTRY

PAN9520 ETU

Easy-To-Use Evaluation Tool

User Guide

Rev. 1.0



Wireless Connectivity

Overview

The PAN9520 ETU is an evaluation board for the PAN9520 embedded Wi-Fi module based on the Espressif® ESP32-S2.

PAN9520 Features

- Embedded 2.4 GHz Wi-Fi 802.11 b/g/n module
- Xtensa® single-core 32-bit LX7 microprocessor, up to 240 MHz
- Chip internal 128 kB ROM, 320 kB SRAM, and 16 kB low power SRAM
- Integrated QSPI Flash and PSRAM (a variety of memory densities are available)
- Ultra-Low Power (ULP) co-processor usable in deep sleep mode
- All security features required for WPA2 and WPA3 personal
- Espressif IoT Development Framework (ESP-IDF) with a multitude of examples available for software development
- Supports 20 MHz and 40 MHz bandwidths in 2.4 GHz band with data rates up to 150 Mbps
- Simultaneous support for Infrastructure Station, SoftAP, and promiscuous modes
- 802.11mc Fine Time Measurement (FTM)
- 36× programmable GPIOs with a rich set of alternative functionalities

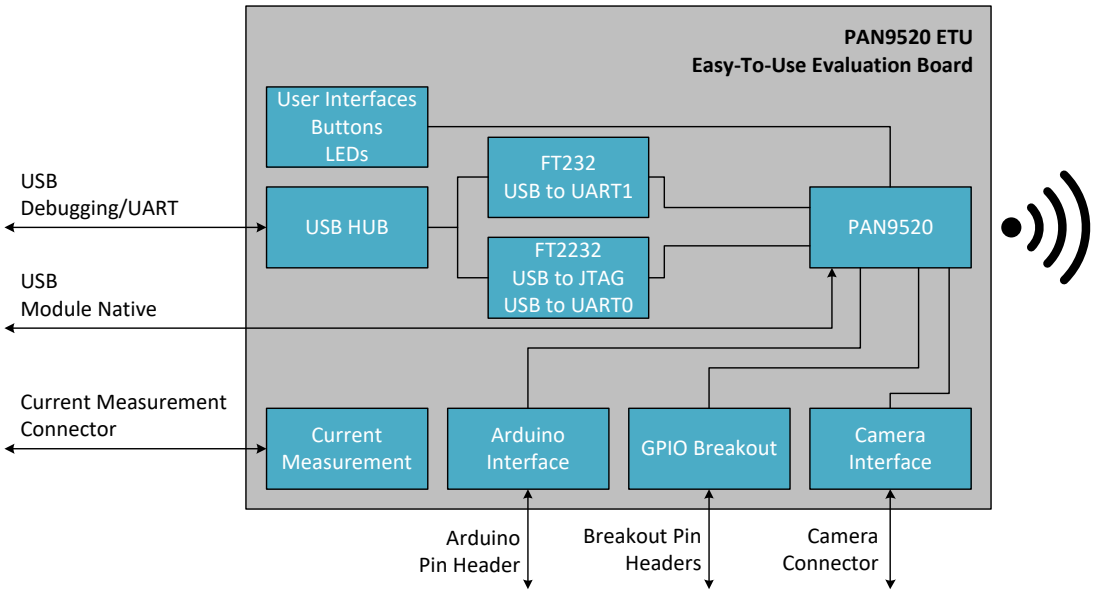
PAN9520 Characteristics

- Surface Mount Type (SMT): 24 mm × 13 mm × 3.1 mm
- Tx power: up to 19.5 dBm at IEEE 802.11b 1 Mbps
- Rx sensitivity: of -97 dBm at IEEE 802.11b 1 Mbps
- Power supply: 3 V to 3.6 V
- Current consumption: 260 mA Tx (average at 11b, 11 Mbps), 76 mA Rx (40 MHz channel), 310 mA Tx peak
- Deep sleep mode: <100 µA typical power consumption (RTC timer only and V_{DD_SPI} disconnected)
- Wide temperature range from -40 °C to 85 °C

Evaluation Tool Features

- Arduino Interface configurable as shield or board
- All GPIO break out
- FTDI FT2232 USB to dual interface
  - USB to JTAG
  - USB to UART0
- FTDI FT232 USB to UART1 interface
- Peripherals are deactivatable for low power applications
- 4× user Buttons, 3× user LEDs
- Module native USB interface
- Camera interface
- Contains module variant ENW49D01A1KF with 4 MB of QSPI Flash and 2 MB of QSPI PSRAM

Block Diagram



By purchase of any of the products described in this document the customer accepts the document's validity and declares their agreement and understanding of its contents and recommendations. Panasonic Industrial Devices Europe GmbH (Panasonic) reserves the right to make changes as required at any time without notification.

This User Guide does not lodge the claim to be complete and free of mistakes.

The information contained herein is presented only as guidance for Product use. No responsibility is assumed by Panasonic for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.

Description of hardware, software, and other information in this document is only intended to illustrate the functionality of the referred Panasonic product. It should not be construed as guaranteeing specific functionality of the product as described or suitable for a particular application.

Any provided (source) code shall not be used or incorporated into any products or systems whose manufacture, use or sale is prohibited under any applicable laws or regulations.

Any outlined or referenced (source) code within this document is provided on an “as is” basis without any right to technical support or updates and without warranty of any kind on a free of charge basis according to § 516 German Civil Law (BGB) including without limitation, any warranties or conditions of title, non infringement, merchantability, or fitness for a particular purpose. Customer acknowledges that (source) code may bear defects and errors.

The third party tools mentioned in this document are offered by independent third party providers who are solely responsible for these products. Panasonic has no responsibility whatsoever for the performance, product descriptions, specifications, referenced content, or any and all claims or representations of these third party providers. Panasonic makes no warranty whatsoever, neither express nor implied, with respect to the goods, the referenced contents, or any and all claims or representations of the third party providers.

To the maximum extent allowable by Law Panasonic assumes no liability whatsoever including without limitation, indirect, consequential, special, or incidental damages or loss, including without limitation loss of profits, loss of opportunities, business interruption, and loss of data.

Table of Contents

1 About This Document

1.1 Purpose and Audience

This User Guide is intended to give a detailed description of the Easy-To-Use (ETU) evaluation board components and functionalities.

It is intended for hardware design, application, and Original Equipment Manufacturers (OEM) engineers.

The product is referred to as “the PAN9520 ETU” or “the ETU” within this document.

1.2 Revision History

Revision	Date	Modifications/Remarks
1.0	2021-06-21	First version

1.3 Use of Symbols

Symbol	Description
	**Note** Indicates important information for the proper use of the product. Non-observance can lead to errors.
	**Attention** Indicates important notes that, if not observed, can put the product’s functionality at risk.
	**Tip** Indicates useful information designed to facilitate working with the module and software.
⇨ [chapter number] [chapter title]	**Cross reference** Indicates cross references within the document. **Example:** Description of the symbols used in this document ⇨ 1.3 Use of Symbols.
✓	**Requirement** Indicates a requirement that must be met before the corresponding tasks can be completed.
➔	**Result** Indicates the result of a task or the result of a series of tasks.
This font	**GUI text** Indicates fixed terms and text of the graphical user interface. **Example:** Click **Save**.

Symbol	Description
Menu > **Menu item**	**Path** Indicates a path, e.g. to access a dialog. **Example:** In the menu, select **File** > **Setup page**.
`This font`	**File names, messages, user input** Indicates file names or messages and information displayed on the screen or to be selected or entered by the user. **Examples:** `pan1760.c` contains the actual module initialization. The message `Failed to save your data` is displayed. Enter the value `Product 123`.
Key	**Key** Indicates a key on the keyboard, e.g. **F10**.

1.4 Related Documents

For related documents please refer to the Panasonic website ⇨ 4.2 Product Information.

2 Overview

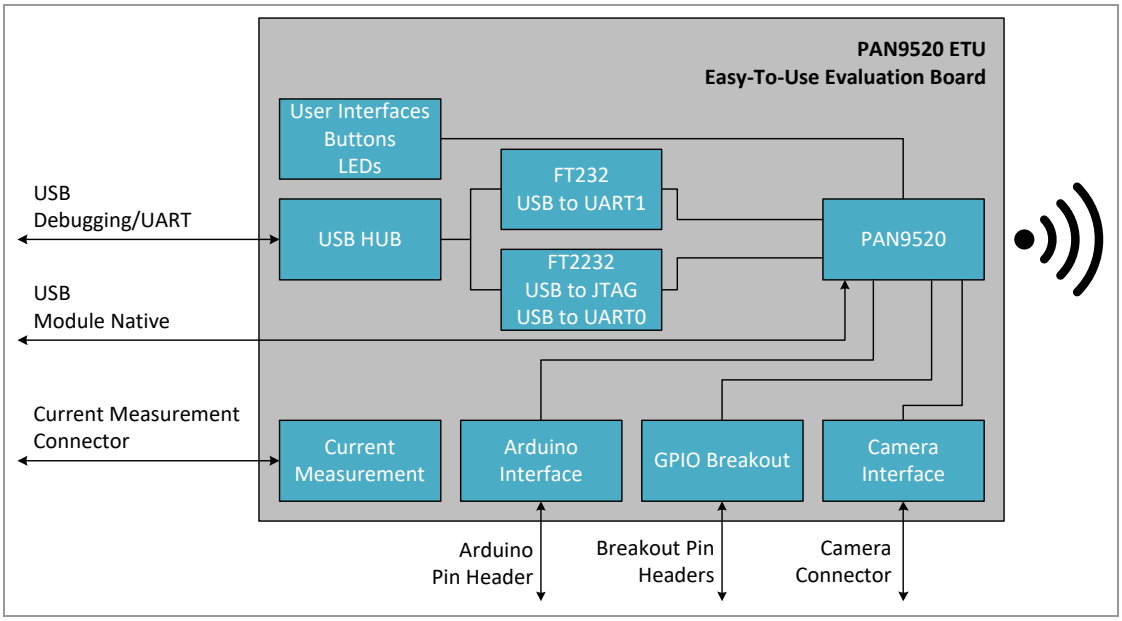
The PAN9520 ETU is an evaluation board for the PAN9520 embedded Wi-Fi module, based on the Espressif ESP32-S2.

It gives access to the PAN9520 over several different Interfaces like USB, UART, JTAG, and GPIOs. With the PAN9520 ETU, an evaluation of the PAN9520 can be easily done, which results in a high reduction of development time.

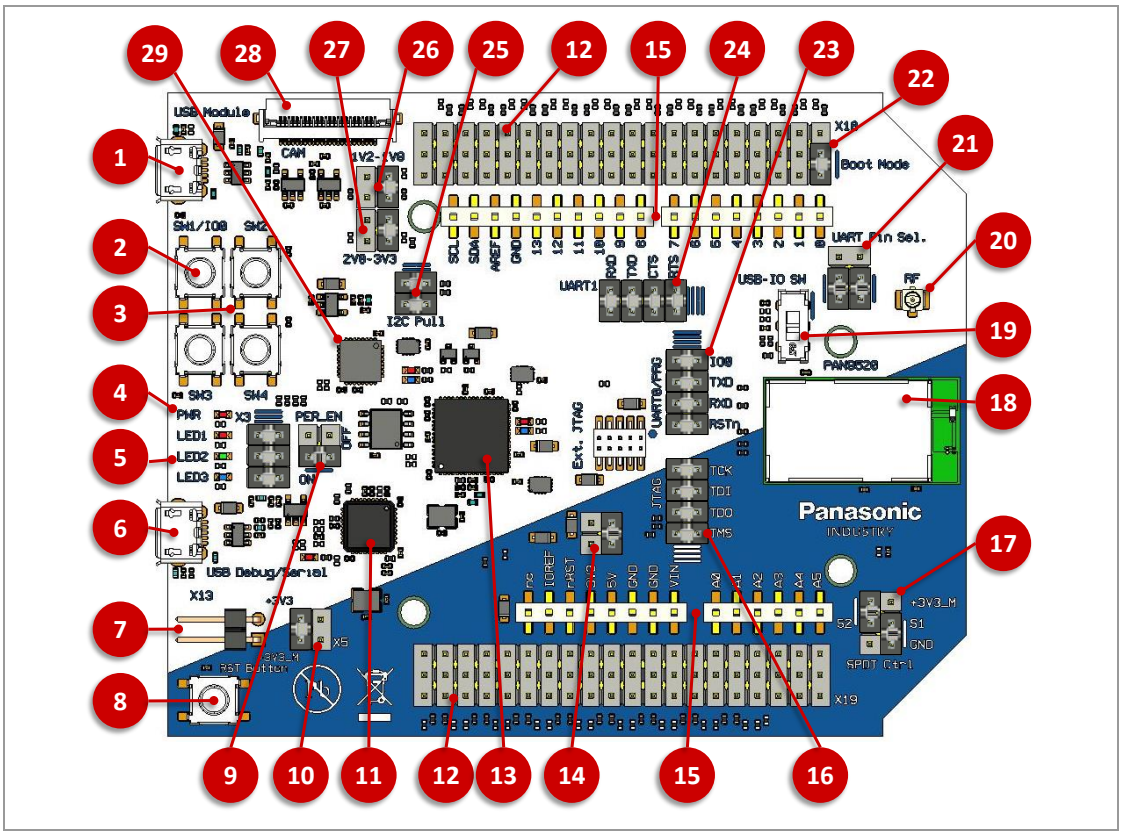
For related documents please refer to ⇨ 4.2 Product Information.

3 PAN9520 ETU

3.1 Block Diagram



3.2 Board Overview



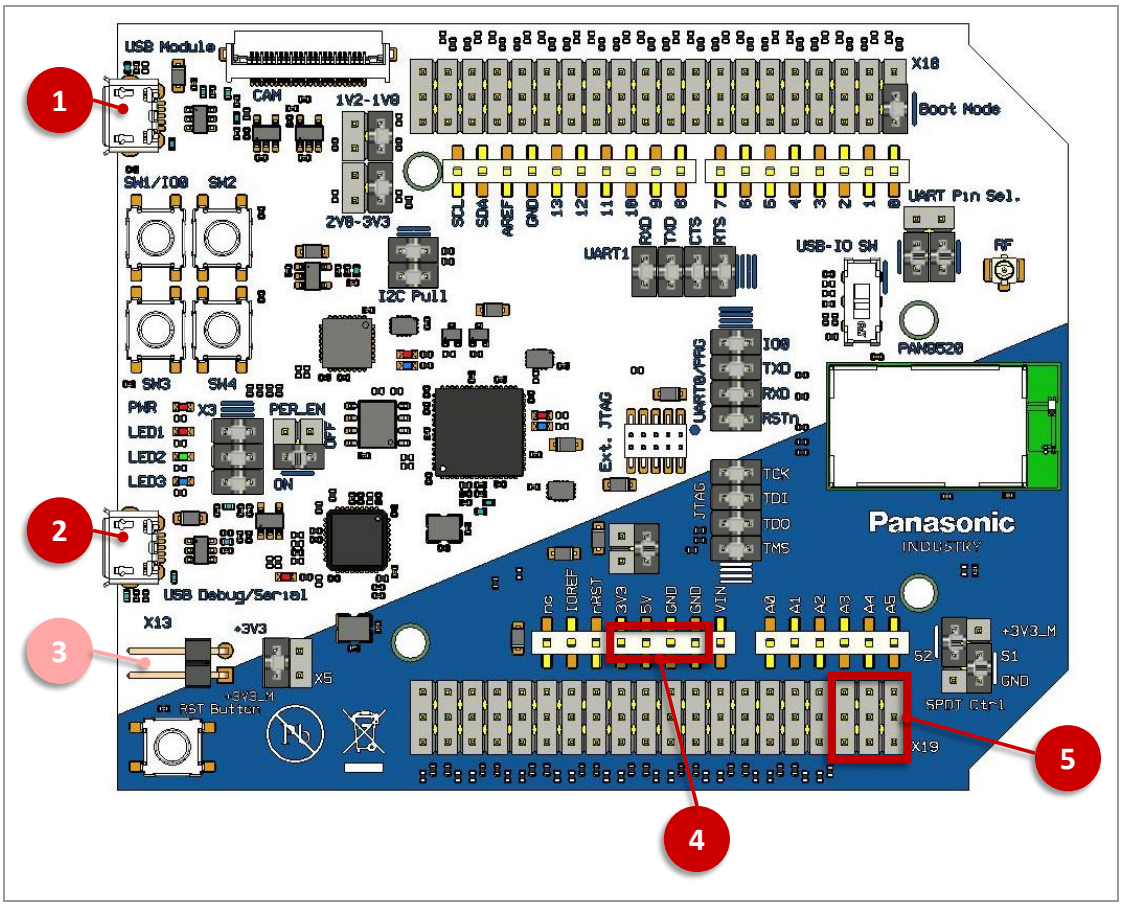
No.	Name	Function
1	Module native USB connector	Connected to GPIO 19 and GPIO 20, which can be used for USB. ⇨ 3.11 USB-IO Switch
2	GPIO0/boot button	Button SW1 to be used for controlling the Boot Mode at start-up or as general-purpose button in application. ⇨ 3.7 User Buttons ⇨ SW1/IO0 at Start-Up
3	General-purpose buttons	Buttons SW2 to SW4 are for general purpose. ⇨ 3.7 User Buttons
4	Power LED	LED that indicates power on +3V3 net.
5	General purpose LEDs	LEDs 1 to 3 can be used for general purpose. ⇨ 3.6 User LEDs
6	USB connector	USB connector for FT232 (UART1) and FT2232 (JTAG and UART0).
7 10	Current measurement pin headers	If no jumper is connected to pin 7 and pin 10, these pins can be used for current measurement. ⇨ 3.10 Current Measurement
8	Reset button	Resets the PAN9520.
9	Peripheral power supply	Pin header to control the supply of peripheral devices as FT232 and FT2232. ⇨ 3.14 Disabling Peripherals
11	USB hub IC	USB hub for connecting FT232 and FT2232 to the USB connector.
12	Break out pin header	All GPIOs of the PAN9520 are connected to these headers. ⇨ 3.5 Breakout Pin Header
13	FT2232 IC (JTAG and UART0)	USB to dual port IC (port0: JTAG, port1: UART0)
14	5V input/output control	Configures, if 5 V pins are connected via a diode (input) or directly (input and/or output). ⇨ 3.3 Powering Options ⇨ 3.9 Arduino Board/Shield Configuration
15	Arduino socket	Arduino compatible socket. ⇨ 3.8 Arduino Interface
16	JTAG signals	Pin header for connecting JTAG signals (FT2232 or external debugger) to the module's pins.
17	SPDT control pin header	Pin header for controlling the SPDT RF switch on the PAN9520. ⇨ 3.15 PAN9520 RF Switch (SPDT) Control
18	PAN9520 module	PAN9520 802.11 b/g/n embedded Wi-Fi module

No.	Name	Function
19	USB IO switch	Switch for connecting either periphery like e.g. the GPIO header pins or USB related pull resistors to the native USB traces. ⇨ 3.11 USB-IO Switch
20	U.FL connector	Is connected to the PAN9520's RF pad. ⇨ 3.15 PAN9520 RF Switch (SPDT) Control
21	Arduino UART signal selector	Pin header to swap the Arduino interface's pins for UART TX and RX signals. ⇨ 3.9 Arduino Board/Shield Configuration
22	GPIO0/boot pull-up jumper	For entering the application Boot Mode per default, when powering the module, the pull-up on GPIO0 must be connected. ⇨ 3.4 Default Boot Mode
23	UART0/program signals	Pin header for connecting UART0 (TX and RX), RSTn (DTR controlled), and IO0 (RTS controlled) of the module to FT2232.
24	UART1 signals	Pin header for connecting the module's UART1 signals (TX, RX, CTS, RTS) to FT232.
25	I²C pull-ups	4.7 kΩ pull-up resistors can be connected to GPIO 7 and GPIO 8 via these pins. GPIO 7 and GPIO 8 are used for I²C (Inter Integrated Circuit) on the Arduino socket and camera connector. ⇨ 3.12 I²C Pull-Up Resistors
26	Camera voltage control 1.2 to 1.8 V	Low supply voltage control of camera interface. 1.2, 1.5 and 1.8 V can be configured by several jumper settings. ⇨ 3.13.2.1 1V2-1V8 Pin Header
27	Camera voltage control 2.8 to 3.3 V	High supply voltage control of camera interface. 2.8 V, 3 V, and 3.3 V can be configured by several jumper settings. ⇨ 3.13.2.2 2V8-3V3 Pin Header
28	Camera interface	Connector for using different camera modules. ⇨ 3.13 Camera Interface
29	FT232 IC (UART1)	USB to serial IC that is used for interfacing UART1 (inclusive flow control).

3.3 Powering Options

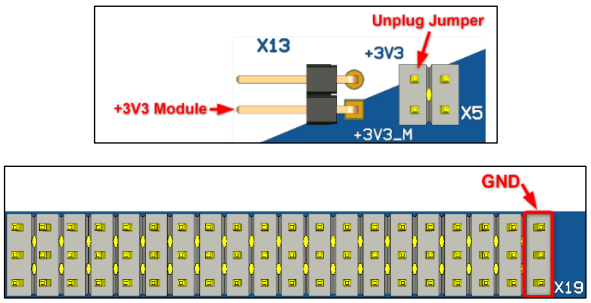
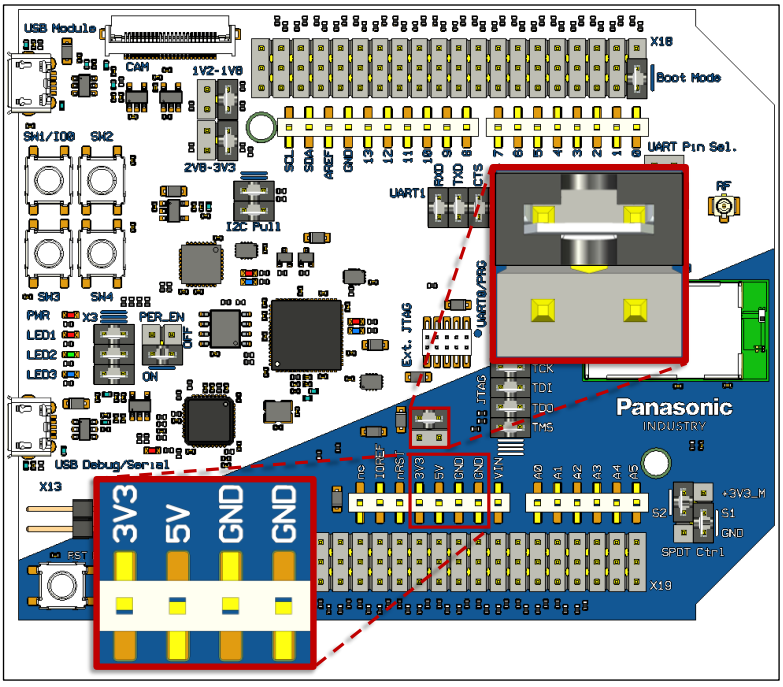
The ETU can be powered by the following different sources:

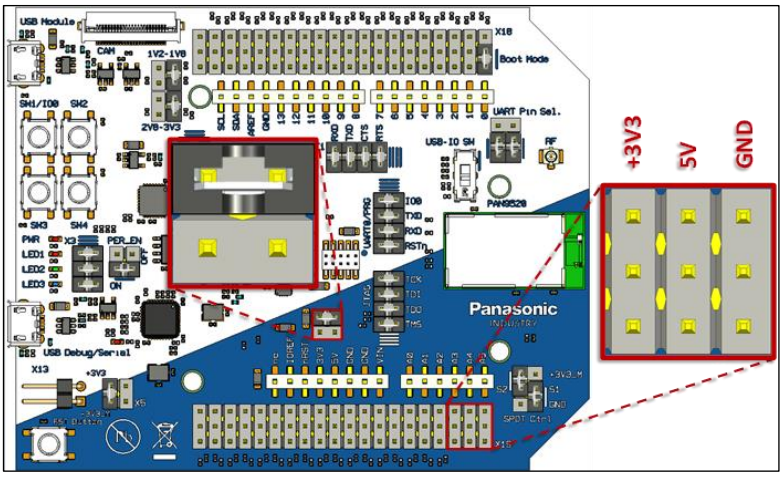
(Light red: GND pin is not included on this pin header.)



Risk of Damage the Board Components (no. 4 and no. 5)

Do not supply 5 V on the 3.3 V pin (“Arduino pin header” and “breakout pins”). This could lead to damage on board components.

No.	Powering Option	Description
1	Module native USB connector	The whole board can be powered over the USB connector.
2	USB connector	The whole board can be powered over the USB connector. The PAN9520 is still powered if the peripherals are deactivated over PER_EN.
3	Current measurement pin header	The current measurement pin header X13 can be used for powering the PAN9520 only. Please note that GND must be connected via a different connector (e.g. X19 as shown below). 
4	Arduino pin header	For power supply, the pin 3V3 and the pin 5 V of the Arduino socket can be used. If the pin 5 V is used, it is recommended to set “5 V Sel.” to “input”. This will add a diode into the supply path, which avoids problems when connecting USB cables. 

No.	Powering Option	Description
5	Breakout pins	For power supply, the pin 3.3 V and the pin 5 V of the breakout pin headers can be used. If the pin 5V is used, it is recommended to set “5 V Sel.” to “input”. This will add a diode into the supply path, which avoids problems when connecting USB cables. 

3.4 Default Boot Mode

Workaround

Please note that the described setting shall ensure that the default application Boot Mode is also entered directly after powering the board. This Default Mode is also set by the ESP32-S2, but does only work, if the device is already powered and then reset.

Please use SW1 for entering the Download Mode ⇨ SW1/IO0 at Start-Up.

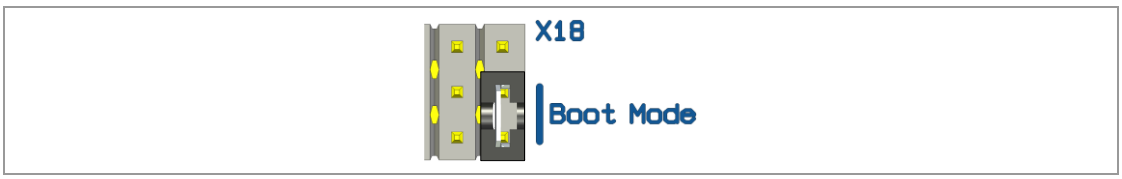
Pull-Up GPIO0

It is recommended to connect a pull-up resistor on GPIO0 by placing a jumper on the “Boot Mode” pins. This ensures that the PAN9520 boots the loaded application directly after powering the device.

At the start-up of the PAN9520, the ESP32-S2 IC evaluates the so-called “strapping pins”. The default values of them are determined by internal pull-resistors of the ESP32-S2. These work fine for start-ups caused by resets. Anyway, if the IC is powered, the pull-resistors need some time to be configured and the device does not start in the Default Mode. This is especially important to note for GPIO0, which configures the Boot Mode after start-up.

For ensuring that the PAN9520 boots the loaded application after powering the board, it is recommended to connect the related pull-up resistor on GPIO0. The caption “Boot Mode” on the PAN9520 ETU indicates the corresponding jumper placement.

If the jumper is not placed, after powering the module, the module will enter the Download Mode and an additional reset is needed to start the device in the usual application Boot Mode.

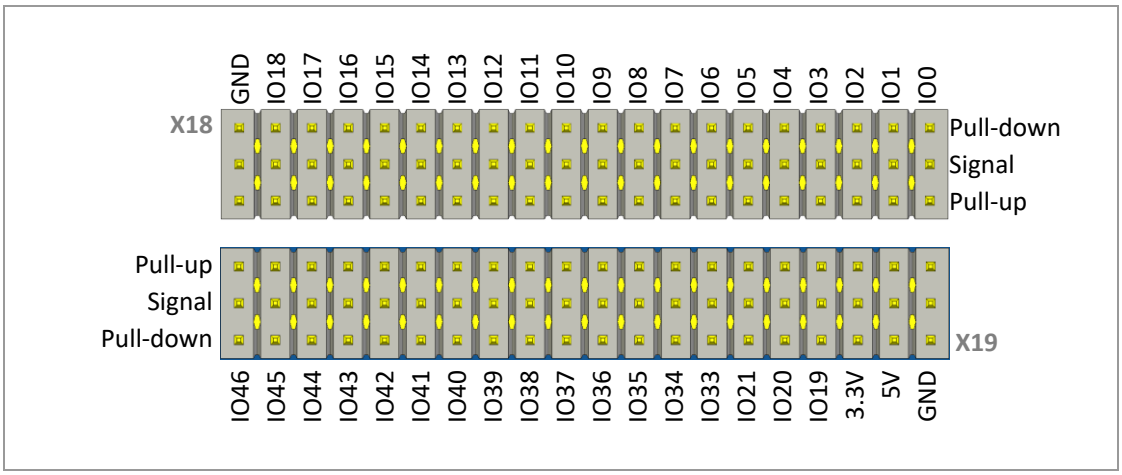


For further information on the “strapping pins” and boot options please refer to the “PAN9520 Product Specification”.

3.5 Breakout Pin Header

Every GPIO of the PAN9520 can be accessed through the breakout pin headers X18 and X19. All signal pins are located on the center row of the three-row pin headers. The two pins next to each signal pin are connected to 100 kΩ pull-resistors. The outer row’s pins are connected to pull-down resistors and those of the inner row are connected to pull-up resistors. In contrast, all power signals (GND, 3.3 V, and 5 V) are connected to all three pins.

The following figure and table gives an overview about the connection between ETU, PAN9520, and ESP32-S2.



ETU Pin	PAN9520/ESP32-S2 Pin Name	PAN9520 Footprint	ESP32-S2 Footprint
IO0	GPIO0	I11	5
IO1	GPIO1	B11	6
IO2	GPIO2	C11	7

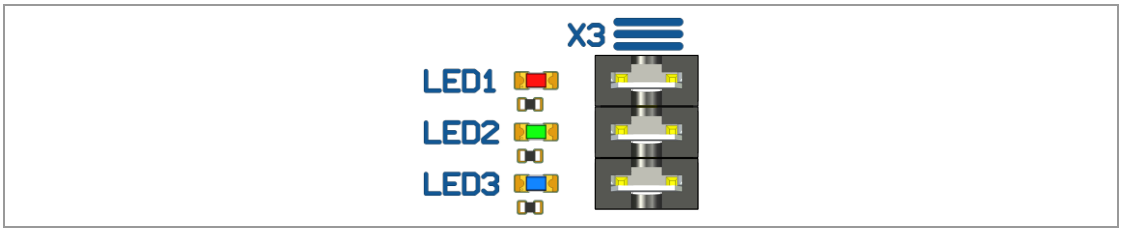
ETU Pin	PAN9520/ESP32-S2 Pin Name	PAN9520 Footprint	ESP32-S2 Footprint
IO3	GPIO3	C12	8
IO4	GPIO4	C13	9
IO5	GPIO5	D11	10
IO6	GPIO6	D13	11
IO7	GPIO7	E11	12
IO8	GPIO8	E12	13
IO9	GPIO9	E13	14
IO10	GPIO10	F13	15
IO11	GPIO11	F11	16
IO12	GPIO12	G13	17
IO13	GPIO13	G12	18
IO14	GPIO14	B11	19
IO15	XTAL_32K_P	I10	21
IO16	XTAL_32K_N	I9	22
IO17	DAC_1	I8	23
IO18	DAC_2	I7	24
IO19	GPIO19	H7	25
IO20	GPIO20	H6	26
IO21	GPIO21	H11	28
IO33	GPIO33	C1	37
IO34	GPIO34	B1	38
IO35	GPIO35	A2	39
IO36	GPIO36	A3	40
IO37	GPIO37	A4	41
IO38	GPIO38	A5	42
IO39	MTCK	E1	43
IO40	MTDO	F1	44
IO41	MTDI	G1	46
IO42	MTMS	H1	47
IO43	U0TXD	A7	48
IO44	U0RXD	A8	49
IO45	GPIO45	A10	50
IO46	GPIO46	A11	55
3.3 V	The maximum output current is 500 mA (if no other 3.3 V pin is used as output).		
5 V	The maximum output current depends on the USB supply.		
GND			

3.6 User LEDs

There are three LEDs that can be used for general purposes on the PAN9520 ETU.

The LEDs are connected to GPIOs via pin header X3 and hence every LED can be disconnected by unplugging the related jumper.

The following figure and table show details on the assigned pins.



LED	Color	PAN9520/ESP32-S2 Pin Name	PAN9520 Footprint	ESP32-S2 Footprint
LED1	Red	GPIO21	H11	28
LED2	Green	GPIO33	C1	37
LED3	Blue	GPIO38	A5	42

3.7 User Buttons

For interacting with the PAN9520, there are four user buttons (tactile switches) on the ETU. After start-up, all of these buttons can be used as general purpose buttons (for additional functionality of SW1 please refer to ⇨ SW1/IO0 at Start-Up).

When pressing a button, the corresponding GPIO will be connected to ground, which makes the buttons active-low. Needed pull-up resistors can be either connected on the ESP32-S2 IC by software, or externally by using jumpers on the GPIO breakout header (for details please refer to ⇨ 3.5 Breakout Pin Header).

The following table describes which GPIOs are connected to the buttons.

Button Name	PAN9520/ESP32-S2 Pin Name	PAN9520 Footprint	ESP32-S2 Footprint
SW1/IO0	GPIO0	I11	5
SW2	GPIO1	B11	6
SW3	GPIO2	C11	7
SW4	GPIO3	C12	8

SW1/IO0 at Start-Up

Pull-Up GPIO0

It is recommended to connect a pull-up resistor on GPIO0 by placing a jumper on the “Boot Mode” pins. This ensures that the PAN9520 boots the loaded application directly after powering the device. If the device is already powered and reset, the default application Boot Mode will be also configured by an ESP32-S2 internal pull-up resistor.

At start-up, the level on GPIO0 is evaluated for configuring the Boot Mode. If the pin’s level is high, the device will enter the usual application Boot Mode. If the pin’s level is low, the device will enter the Download Mode, which allows loading software to the PAN9520 .

Because SW1 is connected to GPIO0, the button can be used for entering this Download Mode manually. Therefore, SW1 must be pressed while resetting the module.

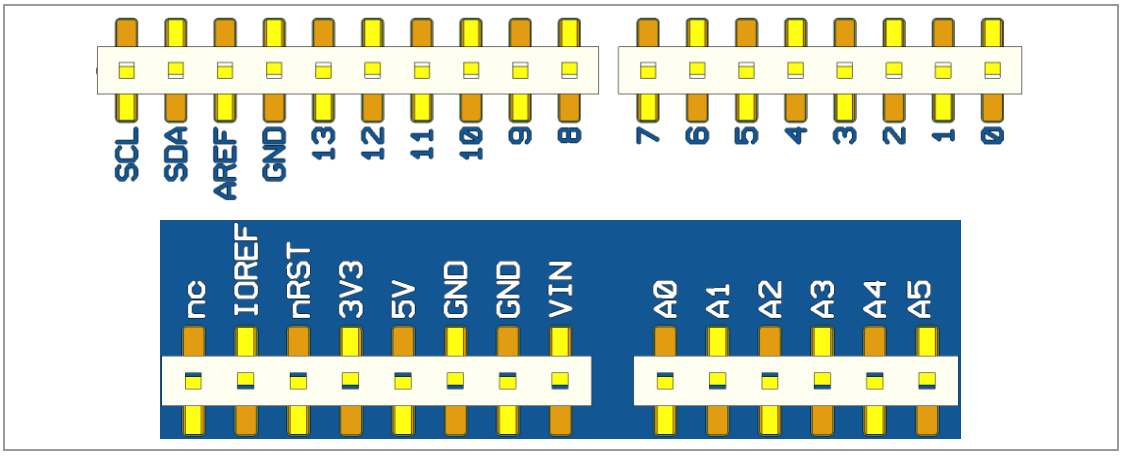
IO0 is Controlled by FT2232

Usually, it is not required to set the Boot Mode manually. IO0 and RSTn are controlled by the USB to UART0 interface of the FT2232 when downloading software.

For further information please refer to ⇨ 3.4 Default Boot Mode and the “PAN9520 Product Specification”.

3.8 Arduino Interface

The Arduino interface can be used to stack the ETU with other boards and shields with Arduino connectors.



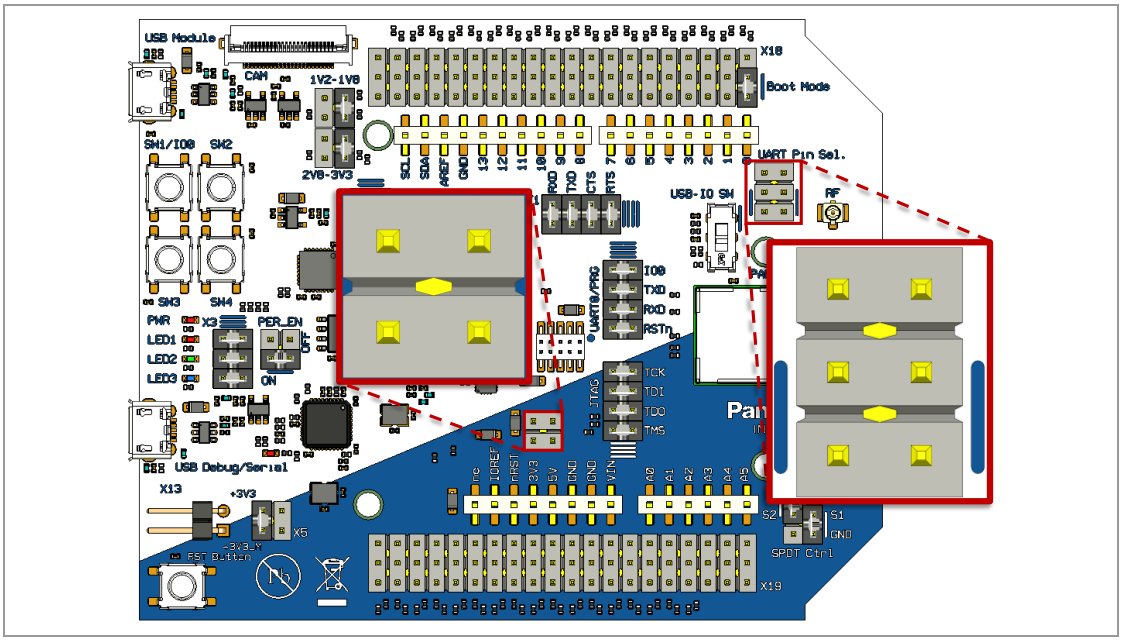
Arduino Pin	Function	PAN9520/ESP32-S2 Pin Name	PAN9520 Footprint	ESP32-S2 Footprint
IOREF	3.3 V Ref Voltage Out			
nRST	Module Reset	CHIP PU	A12	56
3V3	3.3 V	The maximum output current is 500 mA (if no other 3.3 V pin is used as output).		
5V	5V input/output	The maximum output current depends on the USB supply.		
GND	Ground			
GND	Ground			
VIN	Not connected			
A0	Analog Input	GPIO1	B11	6
A1	Analog Input	GPIO2	C11	7
A2	Analog Input	GPIO3	C12	8
A3	Analog Input	GPIO4	C13	9
A4	Analog Input	GPIO5	D11	10
A5	Analog Input	GPIO6	D13	11
SCL	I^2C Clock	GPIO7	E11	12
SDA	I^2C Data	GPIO8	E12	13
AREF	Not connected			
GND	Ground			
13	GPIO/SPI_SCK	GPIO36	A3	40
12	GPIO/SPI_MISO	GPIO37	A4	41
11	GPIO/SPI_MOSI	GPIO35	A2	39
10	GPIO/SPI_SS	GPIO34	B1	38
9	GPIO	GPIO9	E13	14
8	GPIO	GPIO10	F13	15
7	GPIO	GPIO11	F11	16
6	GPIO	GPIO12	G13	17
5	GPIO	GPIO13	G12	18
4	GPIO	GPIO14	G11	19
3	GPIO	GPIO15 (XTAL_32K_P)	I10	21
2	GPIO	GPIO16 (XTAL_32K_N)	I9	22
1	GPIO/UART (TX/RX: depending on “UART Pin Sel.” setup)	GPIO17 (DAC_1)/GPIO18 (DAC_2)	I8/I7	23/24

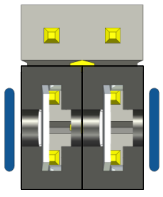
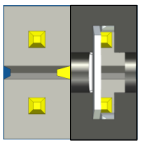
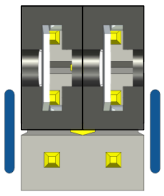
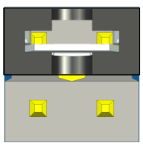
Arduino Pin	Function	PAN9520/ESP32-S2 Pin Name	PAN9520 Footprint	ESP32-S2 Footprint
0	GPIO/UART (RX/TX: depending on “UART Pin Sel.” setup)	GPIO18 (DAC_2)/GPIO17 (DAC_1)	I7/I8	24/23

3.9 Arduino Board/Shield Configuration

The ETU can be used either as Arduino board or as Arduino shield. The UART communication and the 5 V Power input/output configuration is different between these both modes. Jumpers on pin headers “UART Pin Sel.” and “5V Sel.” are used for configuring the desired mode.

The following figure and table shows where the pin headers are located and which jumper settings can be used.



Configuration	UART Pin Sel.	5V Sel.	Description
Board			Ard. Pin 0: GPIO18 (U1RXD) Ard. Pin 1: GPIO17 (U1TXD) 5V Sel: “Out” → no diode
Shield			Ard. Pin 0: GPIO17 (U1TXD) Ard. Pin 1: GPIO18 (U1RXD) 5V Sel: “In” → protection diode in 5 V power path

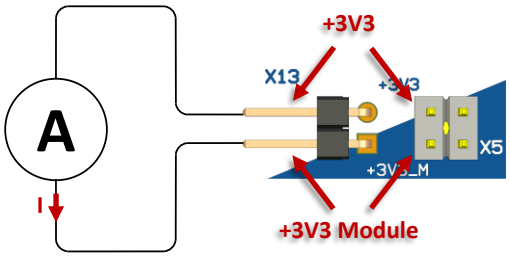
3.10 Current Measurement

Unplug Jumper on X5

To cut the direct power supply to the PAN9520, the Jumper on X5 must be unplugged. Otherwise a current measurement will not work.

The ETU provides the feature to measure the current of the PAN9520, independent from the peripheral components.

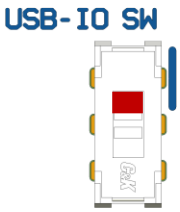
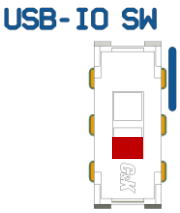
The following setup can be used for the current measurement:



If a power profiling is needed, the “nRF Power Profiler Kit II” from Nordic can be used.

3.11 USB-IO Switch

GPIO 19 and GPIO 20 can be configured to work as USB interface and are thus connected to the module native USB connector “USB Module”. The connection between the module and the USB connector is permanent. To avoid bad signal integrity, caused by stubs or other disturbances, all other peripherals can be disconnected from the USB signal lines. Therefore, the switch “USB-IO SW” is used. The “USB-IO SW” connects either a set of USB-related pull-resistors or the additional periphery (breakout pin header and FT232 RTS and CTS signals) to the USB lines.

Connection Description	Switch State
Peripherals are connected to the USB traces and thus GPIO 19 and GPIO 20. Connected peripherals are the breakout pin header and CTS/RTS lines of the FT232 USB to UART1 IC (can be still disconnected by “UART1” pin header).	
All peripherals are disconnected from the USB traces. Instead, a 1.5 kΩ pull-up resistor is connected to USB D+ (GPIO 20), which is the configuration for a full-speed USB device.	

3.12 I²C Pull-Up Resistors

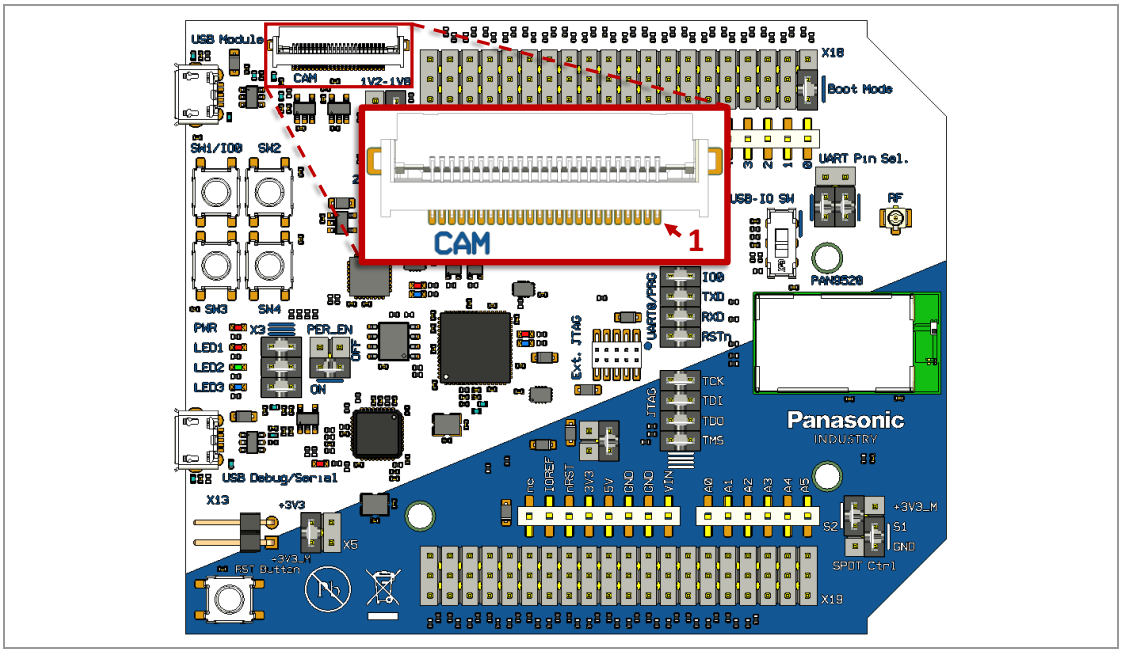
GPIOs 7 and GPIO 8 are used for I²C pins on the Arduino socket and SCCB (Serial Camera Control Bus) pins on the camera connector. Therefore, 4.7 kΩ pull-up resistors are connected to these GPIOs per default. They can be disconnected by unplugging the jumpers on the pin header “I²C Pull”.

3.13 Camera Interface

3.13.1 Pin Assignment

The PAN9520 ETU contains a 24-pin FPC connector for operating a camera module via an 8-bit parallel camera interface and SCCB (Serial Camera Control Bus).

Placement of the Connector (on the ETU)



Pin Assignment

Camera Pin	Function	PAN9520/ESP32-S2 Pin Name	PAN9520 Footprint	ESP32-S2 Footprint
1	Not connected			
2	Not connected			
3	D4	DAC_2 (GPIO18)	I7	24
4	D3	DAC_1 (GPIO17)	I8	23
5	D5	XTAL_32K_N (GPIO16)	I9	22
6	D2	XTAL_32K_P (GPIO15)	I10	21
7	D6	GPIO14	G11	19
8	PCLK	GPIO13	G12	18
9	D7	GPIO12	G13	17
10	GND (digital)			
11	D8	GPIO11	F11	16

Camera Pin	Function	PAN9520/ESP32-S2 Pin Name	PAN9520 Footprint	ESP32-S2 Footprint
12	XCLK	GPIO10	F13	15
13	D9	GPIO6	D13	11
14	+3V3 (DOVDD)			
15	1V2-1V8 (DVDD)			
16	HREF	GPIO5	D11	10
17	PWDN	GPIO4	C13	9
18	VSYNC	GPIO3	C12	8
19	RST_N (permanent pull-up)			
20	SIOC (SCCB clock)	GPIO7	E11	12
21	2V8-3V3 (AVDD)			
22	SIOD (SCCB data)	GPIO8	E12	13
23	GND (analog)			
24	Not connected			

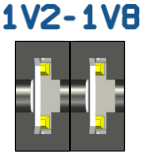
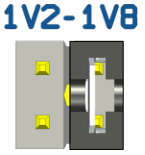
3.13.2 Supply Voltage

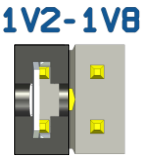
To support a variety of camera modules with different image sensors, the digital core voltage (DVDD) and analog voltage (AVDD) can be configured via the pin headers “1V2-1V8” and “2V8-3V3”. The following chapters describe the voltage levels for each pin header respectively.

3.13.2.1 1V2-1V8 Pin Header

Do not Leave “1V2-1V8” Open

Please note that the DVDD voltage will be 3.3 V if the pin header “1V2-1V8” is left open. This can damage a connected camera module.

Voltage on 1V2-1V8 (DVDD)	Jumper Configuration
1.2 V	
1.5 V	

Voltage on 1V2-1V8 (DVDD)	Jumper Configuration
1.8 V	

If none of the described voltage levels is suitable for the used camera: custom voltages in the range from 1.2 V to 3.3 V can be configured by using resistors instead of jumpers for connecting the pins.

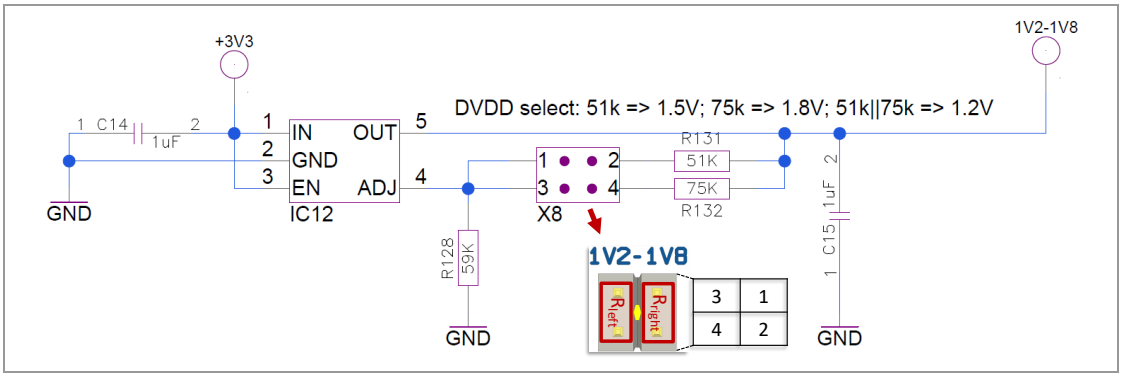
The output voltage can be determined by formula ⇨ (1):

- *R* is the resistance that results from the applied circuit (see equation ⇨ (2))
- R_{left} is the resistance of the left
- R_{right} is the one of the right jumper replacement

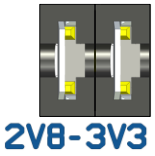
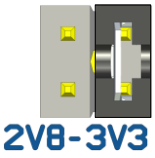
$$DVDD = 0.8V \cdot \frac{R}{59k\Omega} \quad (1)$$

$$R = (51k\Omega + R_{right})||(75k\Omega + R_{left}) \quad (2)$$

The following figure shows the circuit. Pin header “1V2-1V8” is called “X8” in the circuit diagram. The relation between the pin header orientation in the schematic and on the board is also illustrated.



3.13.2.2 2V8-3V3 Pin Header

Voltage on 2V8-3V3 (AVDD)	Jumper Configuration
1.87 V (unusual)	
2.8 V	
3 V	
3.3 V	

If none of the described voltage levels is suitable for the used camera: custom voltages in the range from 1.87 V to 3.3 V can be configured by using resistors instead of jumpers for connecting the pins.

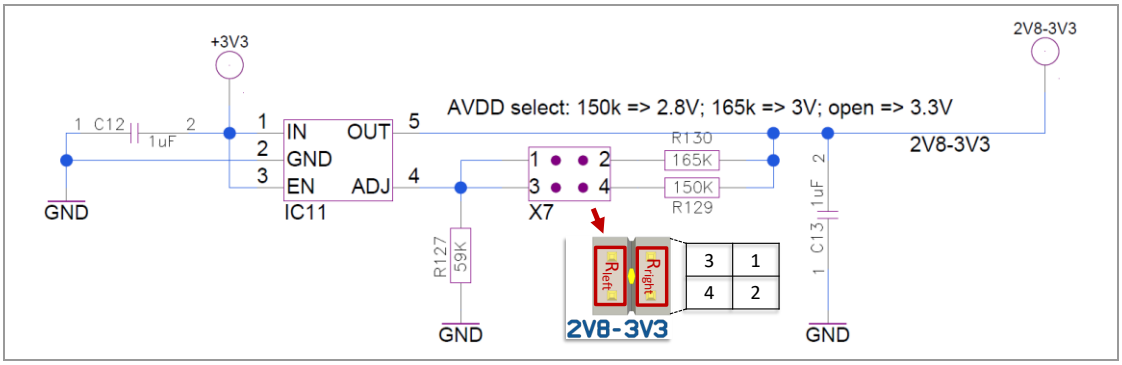
The output voltage can be determined by formula ⇨ (3):

- *R* is the resistance that results from the applied circuit (see equation ⇨ (4))
- R_{left} is the resistance of the left
- R_{right} is the one of the right jumper replacement

$$AVDD = 0.8V \cdot \frac{R}{59k\Omega} \quad (3)$$

$$R = \left(165k\Omega + R_{right}\right)||(150k\Omega + R_{left}) \quad (4)$$

The following figure shows the circuit. Pin header “2V8-3V3” is called “X7” in the circuit diagram. The relation between the pin header orientation in the schematic and on the board is also illustrated.



3.14 Disabling Peripherals

The ETU peripheral components can be deactivated to save energy when it is powered by battery for example.

Peripheral Status	Jumper Configuration
Enabled	
Disabled	

The following table shows the status of the ETU components when PER_EN is in “Disabled” configuration:

Peripheral	Status
USB hub	Disabled
FT232 USB to UART interface (UART1)	Disabled
FT2232 USB to dual interface (JTAG and UATZ0)	Disabled
User buttons	Enabled
Reset button	Enabled
Powering via peripheral USB	Enabled
User LEDs	Enabled

Peripheral	Status
Current measurement	Enabled
PAN1781	Enabled
Module native USB interface	Enabled
Arduino interface	Enabled
Breakout pins	Enabled
Camera interface	Enabled

3.15 PAN9520 RF Switch (SPDT) Control

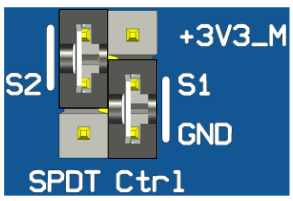
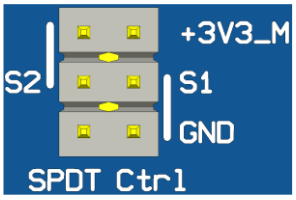
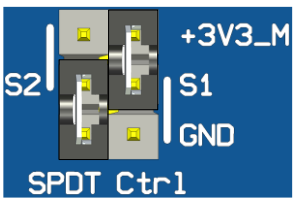
For configuring the RF connection of the PAN9520, the RF connection contains a SPDT (Single Pole Double Throw) RF switch (called SPDT herein after).

The switch connects the ESP32-S2's RF pin to either the module's on-board chip antenna or the RF bottom pad. On the PAN9520 ETU, the bottom pad is connected to the U.FL connector "RF". The SPDT control lines can be accessed via the pin header "SPDT Ctrl".

The following table shows valid configurations.

Please note that other configurations (than those that are shown below) lead to unspecified states of the SPDT RF switch.

Valid Configurations

Configuration Description	Pin Configuration on "SPDT Ctrl"
Module's on-board chip antenna is used	
Module's on-board chip antenna is used (configured by on-board pull-resistors)	
U.FL connector "RF" is used	

3.16 Overview: Interfaces' Pin Assignment

The following table gives a quick overview of the GPIOs and the interfaces they are used for. Please note that all GPIOs are additionally connected to the GPIO breakout headers.

Pin Name	LEDs	Buttons	UART0	Camera Interface	Arduino Interface	JTAG	USB
GPIO0		SW1/IO0	Evaluated at start-up (strapping pin)				
GPIO1		SW2			A0		
GPIO2		SW3			A1		
GPIO3		SW4		VSYNC	A2		
GPIO4				PWDN	A3		
GPIO5				HREF	A4		
GPIO6				D9	A5		
GPIO7				SIOC	I²C SCL		
GPIO8				SIOD	I²C SDA		
GPIO9					D9		
GPIO10				XCLK	D8		
GPIO11				D8	D7		
GPIO12				D7	D6		
GPIO13				PCLK	D5		
GPIO14				D6	D4		
XTAL_32K_P (GPIO15)				D2	D3		
XTAL_32K_N (GPIO16)				D5	D2		
DAC_1 (GPIO17)				D3	D1 U1RX/TX		
DAC_2 (GPIO18)				D4	D0 U1TX/RX		
GPIO19							USB D-
GPIO20							USB D+
GPIO21	LED1						
GPIO33	LED2						
GPIO34					D10 (SS)		
GPIO35					D11 (MOSI)		
GPIO36					D13 (SCK)		
GPIO37					D12 (MISO)		

Pin Name	LEDs	Buttons	UART0	Camera Interface	Arduino Interface	JTAG	USB
GPIO38	LED3						
MTCK (GPIO39)						MTCK	
MTDO (GPIO40)						MTDO	
MTDI (GPIO41)						MTDI	
MTMS (GPIO42)						MTMS	
U0TXD (GPIO43)			U0TXD				
U0RXD (GPIO44)			U0RXD				
GPIO45	Evaluated at start-up (strapping pin)						
GPIO46	Evaluated at start-up (strapping pin)						

3.17 Software Development

Espressif provides several software tools and frameworks. This chapter gives an overview of the major software components Espressif IoT Development Framework (ESP-IDF) and Espressif AT-Software (ESP-AT).

3.17.1 Espressif IoT Development Framework (ESP-IDF)

The PAN9520 is a radio certified module. There are conditions on hardware and software which must be met for the modular approval to be valid.

For detailed information please refer to “PAN9520 Module Integration Guide”.

Espressif provides the IoT Development Framework (ESP-IDF) for software development. This includes a multitude of examples that can be used as starting point for the software development.

Software, general information, and instructions are provided by Espressif on e.g. following web pages:

- ESP-IDF Documentation (Master Branch for ESP32-S2): https://docs.espressif.com/projects/esp-idf/en/latest/esp32s2/index.html
- ESP-IDF Software Repository: https://github.com/espressif/esp-idf
- ESP-IDF with Visual Studio Code: https://docs.espressif.com/projects/esp-idf/en/latest/esp32s2/get-started/vscode-setup.html?highlight=visual%20studio#getting-started-with-vs-code-ide

3.17.2 Espressif AT-Software

> The PAN9520 is a radio certified module. There are conditions on hardware and software which must be met for the modular approval to be valid.
>
> For detailed information please refer to “PAN9520 Module Integration Guide”.

Instead of developing embedded software directly on the PAN9520, AT software for controlling the module by a host can be used. This kind of software is provided by Espressif and is available as pre-built binary set or as source code.

> If the pre-built binary files are used, it is recommended to use the WROVER binaries, because the WROVER features the same memory densities as the ENW49D01A1KF.
>
> Please regard the adaptions described in ⇨ 3.17.3 Customized Software Parts.

The following web pages are related to the AT software:

- ESP-AT Source Code (v2.1.0.0_esp32s2): https://github.com/espressif/esp-at/tree/release/v2.1.0.0_esp32s2
- ESP-AT User Guide (v2.1.0.0_esp32s2): https://docs.espressif.com/projects/esp-at/en/release-v2.1.0.0_esp32s2/index.html
- ESP-AT ESP32-S2 Binaries: https://docs.espressif.com/projects/esp-at/en/release-v2.1.0.0_esp32s2/AT_Binary_Lists/ESP32-S2_AT_binaries.html

3.17.3 Customized Software Parts

When flashing the AT software binaries, two files are recommended to be replaced: `factory_param.bin` and `phy_init_data.bin` (mandatory for a valid modular approval). This chapter describes both files and the required changes. It explains how the files can be flashed in a convenient way.

RF Power Settings and Limits in the “phy_init” Binary

> Please note that a “phy_init” binary that complies with the regulations must be flashed instead of the original `phy_init.bin`.
>
> For details on the requirements please refer to the “PAN9520 Module Integration Guide”.

The PAN9520 has been certified and received a modular approval. As requirement for this approval to be valid, the power for different modulations and channels must be limited depending on the location the device is operated in.

The related settings are configured by the “phy_init_data” binary. Panasonic provides two version of this file (FCC/IC and RED), which can be used for this purpose. For more information please refer to the “PAN9520 Module Integration Guide”.

Make sure that a “phy_init” binary is flashed that complies with the regulations to address “0xF000”.

3.17.3.1 Default Settings in the “factory_param” Binary

For a convenient usage of the pre-built AT software with the PAN9520 ETU, it is recommended to replace the file `factory_param.bin`. This configures the default settings for e.g. the pin assignment. On the PAN9520 ETU, the standard UART1 pins (GPIO 17 to 20) are connected to the FT232 USB to UART interface. In contrast, the default WROVER `factory.bin` configures GPIO 21 as pin RX instead of GPIO 18. By flashing an adapted binary, the on-board USB-to-UART1 interface can be used for controlling the AT software.

The “factory_param” binary is located in the folder **ESP32-S2-WROVER_AT_BIN_VX.X.X.X** > **customized_partitions** (AT binary download folder). It is recommended to edit a copy of the existing binary by using a tool like e.g. “HxD”.

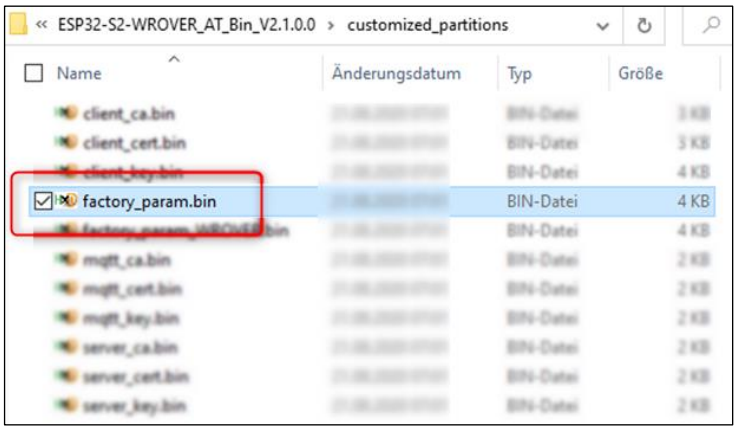
Other methods are also possible, but require the AT source code to be downloaded (for details please refer to https://docs.espressif.com/projects/esp-at/en/release-v2.1.0.0_esp32s2/Compile_and_Develop/How_to_create_factory_parameter_bin.html#modify-factory-param-data).

Adapting Binary

The following requirement must be met:

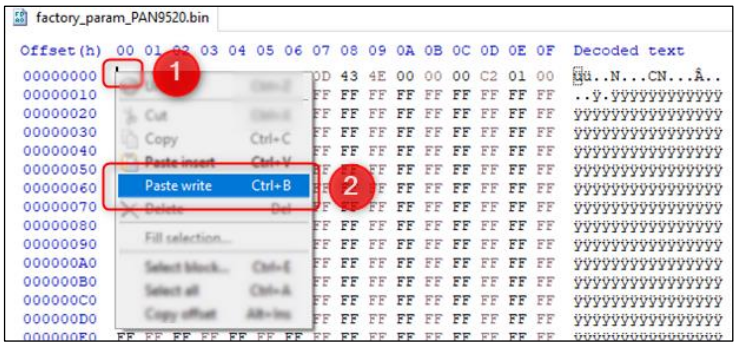
✓ A Hex Editor is installed. In the following the tool “HxD” is used.

1. Navigate to the folder **customized_partitions**.
2. Open either `factory_param.bin` or `factory_param_WROVER.bin` with **HxD**.

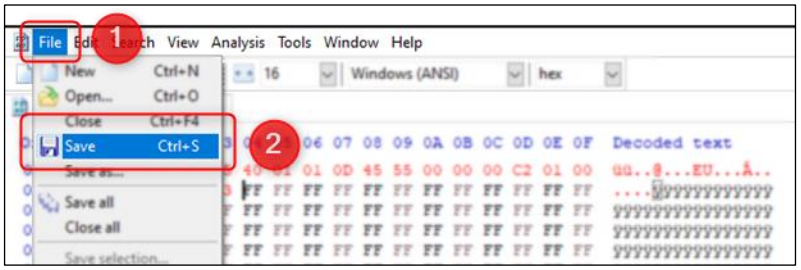


3. Optional: Save a copy of the file with a new name.

4. Copy the bytes (hexadecimal format)
   FC FC 02 05 40 01 01 0D 45 55 00 00 00 C2 01 00 11 12 14 13. Right-click on position **00** (1) > **Paste write** (2) to paste the bytes.



5. Click **File** (1) > **Save** (2) to save the changes.



➔ Now, this file can be flashed to address **0x2D000** instead of the original “factory_param” binary.

For details on adapting the “factory_param” binary please refer to https://docs.espressif.com/projects/esp-at/en/release-v2.1.0.0_esp32s2/Compile_and_Develop/How_to_create_factory_parameter_bin.html#how-to-create-factory-parameter-bin.

Please note that the provided data set contains default country code settings. These are:

• Country Code:	EU	bytes 8 to 11
• Start Channel:	1	byte 6
• Number of Channels:	13	byte 7
• Maximum Power:	(64×0.25=16) dBm	byte 4

3.17.3.2 Flashing the AT Software and adapted Binaries

This section gives a short description on how the AT software and the adapted binaries can be flashed.

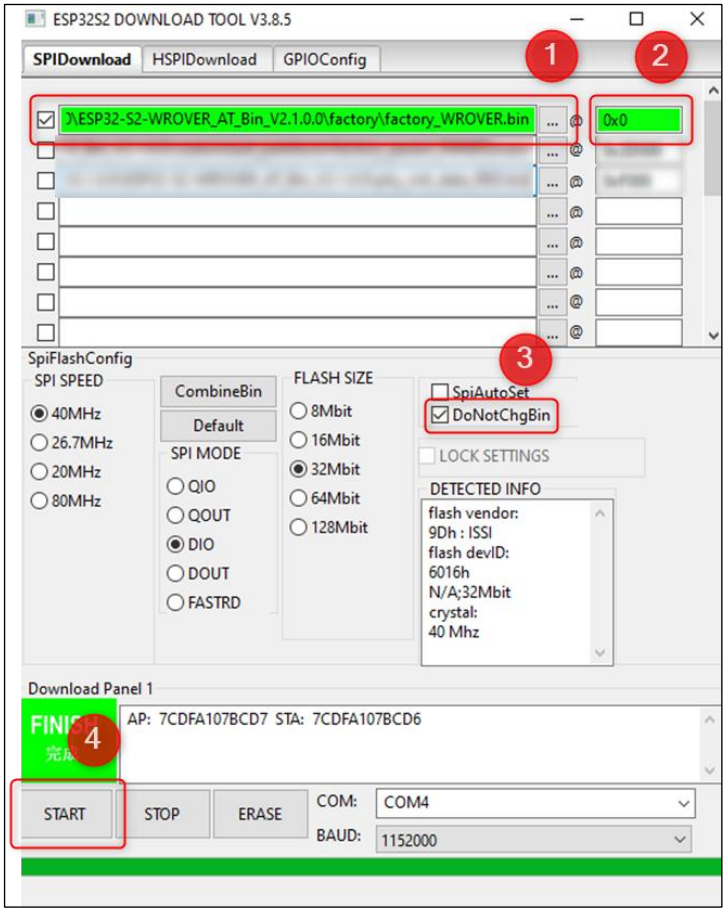
> Please note that instead of flashing the combined “factory_WROVER” binary, all files could be downloaded separately.
>
> For details about this option please refer to https://docs.espressif.com/projects/esp-at/en/release-v2.1.0.0_esp32s2/Get_Started/Downloading_guide.html#downloading-guide.

The following requirements must be met:

- ✓ The binary is adapted ⇨ 3.17.3.1 Default Settings in the “factory_param” Binary.
- ✓ A flash download tool is installed. In the following the tool “ESP DOWNLOAD TOOL” is used.

Write the combined binary of the original AT software:

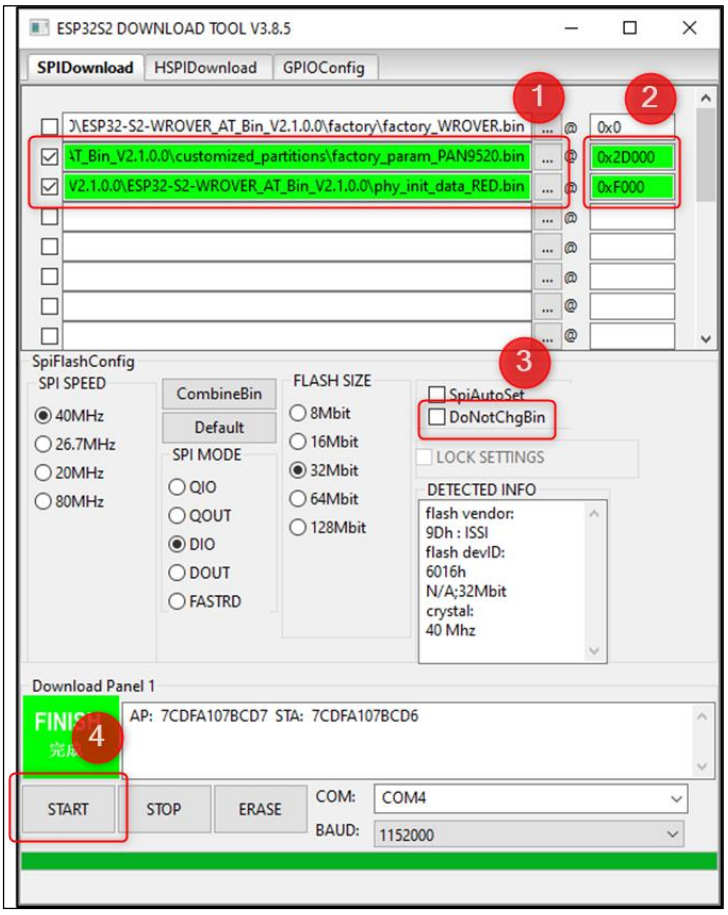
1. Open **ESP DOWNLOAD TOOL**.
2. Select the combined binary file
   B1: **[AT_binary_folder]** > **factory** > `factory_WROVER.bin` (1).



3. Enter the address **0x0** (2).
4. Enable the option **DoNotChgBin** (3).
5. Click **Start** (4) to flash the AT software binary.
   ➔ The combined AT software binary has been flashed.

Overwrite partitions by using the adapted binary files:

6. Select the adapted binary file:
   B2: `factory_param_PAN9520.bin` (example file name) (1)
   B3: `phy_init_data_RED.bin` (example file name) (1).



7. Enter the address **0x2D000** (2) and address **0xF000** (2).
8. Diseable **DoNotChgBin** (5).
9. Click **Start** (4) to flash the adapted binary files.
   ➔ Now the adapted binary files are flashed and ready to be used.

4 Contact Details

4.1 Contact Us

Please contact your local Panasonic Sales office for details on additional product options and services:

For Panasonic Sales assistance in the **EU**, visit
https://eu.industrial.panasonic.com/about-us/contact-us
Email: wireless@eu.panasonic.com

For Panasonic Sales assistance in **North America**, visit the Panasonic website "Sales & Support" to find assistance near you at
https://na.industrial.panasonic.com/distributors

Please visit the **Panasonic Wireless Technical Forum** to submit a question at
https://forum.na.industrial.panasonic.com

4.2 Product Information

Please refer to the Panasonic Wireless Connectivity website for further information on our products and related documents:

For complete Panasonic product details in the **EU**, visit
http://pideu.panasonic.de/products/wireless-modules.html

For complete Panasonic product details in **North America**, visit
http://www.panasonic.com/rfmodules