**Pin Notes**

You'll discover most of the pins on the ESP32 cam are already used or have wonky behaviors.  
To make matters a bit more confusing the layout seems to be done by a inebriated engineer or (more likely) a trace auto-router. Some of those poor decisions (e.g. not exposing the I2C pins) significantly cripple the possible applications. However pins listed as "not exposed" could be accessible using a soldering iron, or a custom FPC in the camera interface.

**GPIO0 - CAM\_PIN\_XCLK or CSI\_MCLK**

* Pull to ground (at reset) to put board into flash mode
* Internally has a 3.3v 10k pullup resistor (R19)
* CSI\_MCLK is used by Camera (line 12 on FPC)
* strapping pin - [ESP32\_S datasheet](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf), default:pull-up spi-boot:1, download-boot:0
* Supports Capactive Sensing T1 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* ADC2\_CH1- 12bit SAR ADC
* Signal: HSPIHD - Parallel QSPI
* Signal: EMAC\_TX\_CLK - Ethernet MAC MII/RII interface
* listed in ESP32\_S datasheet as "RTC\_Function2" I2C\_SDA
* designated as a (wpu) "weak pull up" by ESP32\_S datasheet v3.4 pg53 IO/Mux Addendum

**GPIO1 - U0TXD**

* used for serial output
* designated as OUTPUT ONLY

**GPIO2 - HS2\_DATA0 (IO2)**

* Used by SD-Card
* might be usable as a SPI MISO (need to verify)
* strapping pin - [ESP32\_S datasheet](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf), default:pull-down -- must be down to download boot on GPIO0)
* Supports Capactive Sensing T2 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* ADC2\_CH2- 12bit SAR ADC, SD Memory card v3.01
* Signal: HSPIWP - Parallel QSPI
* designated as a (wpd) "weak pull down" by ESP32\_S datasheet v3.4 pg53 IO/Mux Addendum

**GPIO3 - U0RXD**

* used for serial input
* Signal: EMAC\_RXD2 - Ethernet MAC MII/RII interface
* designated as I1? (input only?)

**GPIO4 - HS\_DATA1**

* Used by SD Card
* has a 47Kohm resistor (R11) on the SD1/MicroSD line
* connected to onboard 3030 SMD LED (Flashlight)
* might be usable as a SPI MOSI (need to verify)
* Supports Capactive Sensing T0 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* ADC2\_CH0- 12bit SAR ADC, Supports SD Memory Card v3.01
* Signal: EMAC\_TX\_ER - Ethernet MAC MII/RII interface
* listed in ESP32\_S datasheet as "RTC\_Function2" I2C\_SCL
* designated as a (wpd) "weak pull down" by ESP32\_S datasheet v3.4 pg53 IO/Mux Addendum
* 1-bit SD Card 'hack' initialize the microSD card as follows, then the microSD card won’t use the GPIO4, GPIO12, GPIO13 data lines (HS\_DATA1, HS\_DATA2, HS\_DATA3 respectively)!
* // enable SD\_MMC in menuconfig > Arduino section
* #include "SD\_MMC.h"
* ...
* SD\_MMC.begin("/sdcard", true)

<https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>

**GPIO5 - (not exposed)**

* CSI\_D0 esp\_camera.h:CAM\_PIN\_D0 -> Camera FPC Y2
* strapping pin - [ESP32\_S datasheet](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf), default:pull-up w/GPIO5 controls Timing of SDIO slave (see documentation)
* Signal: VSPICS0 - Parallel QSPI
* Signal: EMAC\_RX\_CLK - Ethernet MAC MII/RII interface

**GPIO6 - (not exposed)**

* SD\_CLK, HS1\_CLK
* SPICLK - Parallel QSPI

**GPIO7 - (not exposed)**

* SD\_DATA0, HS1\_DATA0

**GPIO8 - (not exposed)**

* SD\_DATA1, HS1\_DATA1

**GPIO9 - (not exposed)**

* SD\_DATA2, HS1\_DATA2
* SPIHD - Parallel QSPI

**GPIO10 - (not exposed)**

* SD\_DATA3, HS1\_DATA3
* SPIWP - Parallel QSPI

**GPIO11 - (not exposed)**

* SD\_CMD, HS1\_CMD
* SPICS0 - Parallel QSPI

**GPIO12 - HS2\_DATA2**

* ESP32\_S datasheet reference: MTDI
* Hint: use the SD-Card 1bit mode trick described on GPIO4 to use this pin!
* used as strapping pin - [ESP32\_S datasheet](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf), default:pull-down selects voltage (3.3v:0, 1.8v:1)
* Supports Capactive Sensing T5 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* JTAG, ADC2\_CH5- 12bit SAR ADC
* Signal HSPIQ - Parallel QSPI
* Signal: EMAC\_TXD3 - Ethernet MAC MII/RII interface
* also RTC\_GPIO15
* MTDI (GPIO12) is used as a bootstrapping pin to select the output voltage of an internal regulator (VDD\_SDIO) which powers the flash chip.
* when connected as SDA (for I2C) causes "RTCWDT\_RTC\_REST"exception ets\_main.c 371 at boot due to flash voltage not being set properly.
* To resolve: it is (Recommended) by EspressIf to Burn the flash voltage selection eFuses. This will permanently configure the internal regulator’s output voltage to 3.3 V, and GPIO12 will not be used as a bootstrapping pin. After that, connect a pull-up resistor to GPIO12.
* Reference & Instructions: [overview of compatibility](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/sd_pullup_requirements.html#compatibility-overview-espressif-hw-sdio)

**GPIO13 - HS2\_DATA3**

* ESP32\_S datasheet: MTCK
* Hint: use the SD-Card 1bit mode trick described on GPIO4 to use this pin!
* Supports Capactive Sensing T4 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* JTAG, ADC2\_CH4- 12bit SAR ADC, SD Memory card v3.01
* Signal HSPID - Parallel QSPI
* Signal: EMAC\_RX\_ER - Ethernet MAC MII/RII interface

**GPIO14 - HS2\_CLK**

* ESP32\_S datasheet: MTMS
* might be usable as an SPI CLK (need to verify)
* Supports Capactive Sensing T6 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* JTAG, ADC6\_CH0- 12bit SAR ADC, SD memory card v3.01
* Signal: HSPICLK - Parallel QSPI
* Signal: EMAC\_TXD2 - Ethernet MAC MII/RII interface
* also RTC\_GPIO16

**GPIO15 - HS2\_CMD**

* ESP32\_S datasheet: MTDO
* might be usable as an SPI CS/CMD (need to verify)
* strapping pin - [ESP32\_S datasheet](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf), default:pull-up enable/disable debugging log print over U0TXD during boot (active:1, silent:0)
* Supports Capactive Sensing T3 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* JTAG, ADC2\_CH3- 12bit SAR ADC
* Signal HSPICS0 - Parallel QSPI
* Signal: EMAC\_RXD3 - Ethernet MAC MII/RII interface
* designated as a (wpu) "weak pull up" by ESP32\_S datasheet v3.4 pg53 IO/Mux Addendum

**GPIO16 - U2RXD "useless gpio"**

* is not RTC (can't be used for pwm or precision measurement)
* Note that GPIO 16 is not an ADC pin, so you can’t read analog sensors on this pin.
* Additionally, GPIO 16 is not an RTC GPIO, so it can’t be used as an external wake-up source.
* has a 10Kohm pull-up resistor
* is connected to CS# pin1 of onboard PSRAM64 (64Mbit/8mb CMOS SRAM) that is required for *most* high resolution camera applications
* Signal: EMAC\_CLK\_OUT - Ethernet MAC MII/RII interface

**GPIO17 - (not exposed) PSRAM\_CLK**

* Signal: EMAC\_CLK\_OUT\_180 - Ethernet MAC MII/RII interface

**GPIO18 - (not exposed)**

* CSI\_D1 esp\_camera.h:CAM\_PIN\_D1 -> Camera FPC Y3
* Signal VSPICLK - Parallel QSPI

**GPIO19 - (not exposed)**

* CSI\_D2 esp\_camera.h:CAM\_PIN\_D2 -> Camera FPC Y4
* Signal VSPIQ - Parallel QSPI
* Signal: EMAC\_TXD0 - Ethernet MAC MII/RII interface

**GPIO20 - ??**

**GPIO21 - (not exposed)**

* esp\_camera.h:CAM\_PIN\_D3 -> Camera FPC Y5
* Signal VSIHD - Parallel QSPI
* Signal: EMAC\_TX\_EN - Ethernet MAC MII/RII interface

**GPIO22 - (not exposed)**

* CS1\_PCLK esp\_camera.h:CAM\_PIN\_PCLK
* Signal VSPIWP - Parallel QSPI
* Signal: EMAC\_TXD1 - Ethernet MAC MII/RII interface

**GPIO23 - (not exposed)**

* CS1\_HSYNC esp\_camera.h:CAM\_PIN\_HREF
* Signal VSPID - Parallel QSPI

**GPIO24 - (not exposed) - ??**

**GPIO25 - (not exposed)**

* esp\_camera.h:CAM\_PIN\_VSYNC -> Camera FPC CS1\_VSYNC
* DAC\_1, ADC2\_CH8- 12bit SAR ADC
* Signal: EMAC\_RXD0 - Ethernet MAC MII/RII interface
* also RTC\_GPIO6

**GPIO26 - (not exposed)**

* TW1\_SDA esp\_camera.h:CAM\_PIN\_SIOD -> Camera FPC SIO\_D TW1\_SCK (line 20)
* could be used for I2C SDA (if it was exposed)
* has 4.7k (R17) resistor, maybe retrofittable with a wire (but unsure of location)
* DAC\_2, ADC2\_CH9- 12bit SAR ADC
* Signal: EMAC\_RXD1 - Ethernet MAC MII/RII interface
* also RTC\_GPIO7

**GPIO27 - (not exposed)**

* TW1\_SCK esp\_camera.h:CAM\_PIN\_SIOC -> Camera FPC SIO\_C TW1\_SCK (line 22)
* could be used for I2C SCL (if it was exposed)
* has 4.7k (R18) resistor, maybe retrofittable with a wire (but unsure of location)
* Supports Capactive Sensing T7 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* ADC2\_CH7- 12bit SAR ADC
* Signal: EMAC\_RX\_DV - Ethernet MAC MII/RII interface
* also RTC\_GPIO17

**GPIO28 - (not exposed)**

* ?? not defined in the ESP32\_S specifications

**GPIO29 - (not exposed)**

* ?? not defined in the ESP32\_S specifications

**GPIO30 - (not exposed)**

* ?? not defined in the ESP32\_S specifications

**GPIO31 - (not exposed)**

* ?? not defined in the ESP32\_S specifications

**GPIO32 - (not exposed)**

* esp\_camera.h:CAM\_PIN\_PWDN - Reset Switch "K1"
* Analog Pin name: 32K\_XP
* esp32cam pcb line has a 0.1uF coupled to GND
* esp32cam pcb line has 3.3v 10Kohm (R6) pullup
* ADC1\_CH4
* Supports Capactive Sensing T9 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* also RTC\_GPIO9

**GPIO33 - (sort of exposed) pcb LED**

* Analog pin name: 32K\_XN
* Supports Capactive Sensing T8 (see ESP32\_S datasheet section 4.1.5 "Touch Sensor")
* Is connected to the internal BLUE(authentic)/RED(clone) LED on the PCB.
* If this is on, the WIFI won't work.
* ADC1\_CH5
* also RTC\_GPIO8

**GPIO34 - (not exposed)**

* esp\_camera.h:CAM\_PIN\_D6 -> Camera FPC Y8
* Analog pin name: VDET\_1
* ADC1\_CH6
* also RTC\_GPIO4

**GPIO35 - (not exposed)**

* esp\_camera.h:CAM\_PIN\_D7
* Analog pin: VDET\_2
* ADC1\_CH7
* also RTC\_GPIO5

**GPIO36 - (not exposed)**

* esp\_camera.h:CAM\_PIN\_D4
* Analog pin name: SENSOR\_VP
* ADC1\_CH0
* also RTC\_GPIO0

**GPIO37 - (not exposed)**

* Analog pin name: SENSOR\_CAPP
* ADC1\_CH1
* also RTC\_GPIO1

**GPIO38 - (not exposed)**

* Analog pin name: SENSOR\_CAPN
* ADC1\_CH2
* also RTC\_GPIO2

**GPIO39 - (not exposed)**

* esp\_camera.h:CAM\_PIN\_D5 -> Camera FPC Y9
* Analog pin name: SENSOR\_VN
* ADC1\_CH3
* also RTC\_GPIO3

**GPIO General Purpose**

According to the ESP32\_S docs "Any GPIO Pins" can be used to:

* Motor PWM
* Three channels of 16bit timers generate PWM waveforms, three fault detection signals, three event capture signals, three sync signals
* Two UART Devices with hardware flow control & DMA
* I2C
* devices in slave or master mode
* I2S
* Stereo input/output, Parallel LCD data output, Parallel Camera Data input
* Infrared Remote Controller
* Eight channels for an IR transmitter & receiver of various waveforms
* General Purpose SPI
* LED PWM (16 independent channels @80mhz with duty accuracy of 16bits),
* Pulse Counter (pcnt\_sig\_ch[0-1]\_in[0-7])
* Signals: EMAC\_MDC\_out, EMAC\_MDI\_in, EMAC\_MDO\_out, EMAC\_CRS\_out, EMAC\_COL\_out - Ethernet MAC MII/RII interface