Contents: [Dobrica PavlinuÅ¡iÄ 's random unstructured stuff]

https://github.com/raphaelbs/esp32-cam-ai-thinker/blob/master/docs/about-esp32-cam.md

connection, flashing

connected to pl2303 serial

pl2303	esp32cam
3v3	not connected
rxd	UnR
rxd	UOT
gnd	GND
5v	5V

To program the board, I userd jumper to jump GPIO0 with GND pin next to it.

https://github.com/raphaelbs/esp32-cam-ai-thinker/blob/master/docs/esp32cam-pin-notes.md

spi

SDI = IO12

SDO = IO13

SCK = IO14

CS = IO15

improved example app

https://github.com/easytarget/esp32-cam-webserver

```
   cp myconfig.sample.h myconfig.h
   vi myconfig.h
dpavlin@nuc:/nuc/esp32/esp32-cam-webserver$ platformio run

dpavlin@nuc:/nuc/esp32/esp32-cam-webserver$ pio run -t upload --upload-port /dev/ttyUSB2
"/home/dpavlin/.platformio/penv/bin/python" "/home/dpavlin/.platformio/packages/tool-esptoolpy/es
--chip esp32 --port "/dev/ttyUSB3" --baud 460800 --before default_reset --after hard_reset \
write_flash -z --flash_mode dio --flash_freq 40m --flash_size detect \
0x1000 /home/dpavlin/.platformio/packages/framework-arduinoespressif32/tools/sdk/bin/bootloader_d
0x8000 /nuc/esp32/esp32-cam-webserver/.pio/build/esp32cam/partitions.bin \
0xe000 /home/dpavlin/.platformio/packages/framework-arduinoespressif32/tools/partitions/boot_app0
0x10000 .pio/build/esp32cam/firmware.bin
```

timelapse

- https://bitluni.net/esp32camtimelapse
- https://github.com/bitluni/ESP32CamTimeLapse

ocr on device

https://github.com/jomjol/AI-on-the-edge-device

https://github.com/jomjol/AI-on-the-edge-device/wiki/Installation

Remove glue from lens (very hard, using sharp knife), and rotate lens by 45 degrees until picture is sharp (I had to use pliers to do this).

```
dpavlin@nuc:/nuc/esp32/AI-on-the-edge-device$ vi sd-card/wlan.ini

dpavlin@nuc:/nuc/esp32/AI-on-the-edge-device/code$ pio run

dpavlin@nuc:/nuc/esp32/AI-on-the-edge-device/code$ pio run -v -t upload --upload-port /dev/ttyUSB

"/home/dpavlin/.platformio/penv/bin/python" "/home/dpavlin/.platformio/packages/tool-esptoolpy/es
--chip esp32 --port "/dev/ttyUSB3" --baud 460800 --before default_reset --after hard_reset \
write_flash -z --flash_mode dio --flash_freq 40m --flash_size detect \
0x1000 /nuc/esp32/AI-on-the-edge-device/code/.pio/build/esp32cam/bootloader.bin \
0x8000 /nuc/esp32/AI-on-the-edge-device/code/.pio/build/esp32cam/partitions.bin \
0xd000 /nuc/esp32/AI-on-the-edge-device/code/.pio/build/esp32cam/ota_data_initial.bin \
0x10000 .pio/build/esp32cam/firmware.bin

# original flashing instructions
esptool write_flash 0x01000 bootloader.bin 0x08000 partitions.bin 0x10000 firmware.bin

# download raw picture
wget 192.168.3.112/img_tmp/raw.jpg
```

---

old, obsolete problems

It seems that my module is usually known as AI thinker variant. It has terrible picture which starts with huge green bias.

It also doesn't work for me in resolutions below 1024x768 (in current esp32 example as of 2019-08-02).

Plugging it into external 5V power supply did not helped much.

To solve green tint, I just left esp32cam module plugged in whole day and night. I guess that image sensor got discharged during night, but next day picture was fine.

Problem with image resolution was fixed by updating to more recent version of ESP32 support for Arduino (as of 2020-04-20 it works fine)

Home Assistant

https://jamesachambers.com/cheap-esp32-cam-home-assistant-esphome-camera-guide/

```
esphome:
  name: esp32cam
  friendly_name: esp32cam

esp32:
  board: esp32cam
  framework:
    type: arduino

# Enable logging
logger:
  level: VERBOSE
  tx_buffer_size: 256

# Enable Home Assistant API
api:
  encryption:
    key: "MsJJJiDv9FTjZ1w8dfoy3Z8cQWjGOsk0m4Wgge0B+8w="
  services:  # change camera parameters on-the-fly
  - service: camera_set_param
    variables:
      name: string
      value: int
    then:
      - lambda: |-
          bool state_return = false;
          if (("contrast" == name) && (value >= -2) && (value <= 2)) { id(espcam).set_contrast(va
          if (("brightness" == name) && (value >= -2) && (value <= 2)) { id(espcam).set_brightnes
          if (("saturation" == name) && (value >= -2) && (value <= 2)) { id(espcam).set_saturatio
          if (("special_effect" == name) && (value >= 0U) && (value <= 6U)) { id(espcam).set_spec
          if (("aec_mode" == name) && (value >= 0U) && (value <= 1U)) { id(espcam).set_aec_mode((
          if (("aec2" == name) && (value >= 0U) && (value <= 1U)) { id(espcam).set_aec2(value); s
          if (("ae_level" == name) && (value >= -2) && (value <= 2)) { id(espcam).set_ae_level(va
          if (("aec_value" == name) && (value >= 0U) && (value <= 1200U)) { id(espcam).set_aec_va
          if (("agc_mode" == name) && (value >= 0U) && (value <= 1U)) { id(espcam).set_agc_mode((
          if (("agc_value" == name) && (value >= 0U) && (value <= 30U)) { id(espcam).set_agc_valu
          if (("agc_gain_ceiling" == name) && (value >= 0U) && (value <= 6U)) { id(espcam).set_ag
          if (("wb_mode" == name) && (value >= 0U) && (value <= 4U)) { id(espcam).set_wb_mode((es
          if (("test_pattern" == name) && (value >= 0U) && (value <= 1U)) { id(espcam).set_test_p
          if (true == state_return) {
            id(espcam).update_camera_parameters();
          }
          else {
            ESP_LOGW("esp32_camera_set_param", "Error in name or data range");
          }

ota:
```

```
  password: "09e4b58a1d186b8b33d100548f33d796"

wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password

  power_save_mode: none

  # Enable fallback hotspot (captive portal) in case wifi connection fails
  ap:
    ssid: "Esp32Cam Fallback Hotspot"
    password: "GTIKgjitx2Re"

captive_portal:

# Example configuration entry
esp32_camera:
  id: espcam
  name: esp-cam
  external_clock:
    pin: GPIO0
    frequency: 20MHz
  i2c_pins:
    sda: GPIO26
    scl: GPIO27
  data_pins: [GPIO5, GPIO18, GPIO19, GPIO21, GPIO36, GPIO39, GPIO34, GPIO35]
  vsync_pin: GPIO25
  href_pin: GPIO23
  pixel_clock_pin: GPIO22
  power_down_pin: GPIO32

  resolution: 800x600
  jpeg_quality: 10  # max. 63
  max_framerate: 1.0fps
  idle_framerate: 0.2fps
  vertical_flip: true
  horizontal_mirror: false
  brightness: 2 # -2 to 2
  contrast: 1 # -2 to 2
  special_effect: none
  # exposure settings
  aec_mode: auto
  aec2: false
  ae_level: 0
  aec_value: 300
  # gain settings
  agc_mode: auto
  agc_gain_ceiling: 2x
  agc_value: 0
  # white balance setting
  wb_mode: auto
output:
# white LED
  - platform: ledc
    channel: 2
    pin: GPIO4
    id: espCamLED
# red status light
  - platform: gpio
    pin:
      number: GPIO33
      inverted: True
    id: gpio_33
light:
  - platform: monochromatic
    output: espCamLED
```

```
    name: esp-cam light
  - platform: binary
    output: gpio_33
    name: esp-cam led
switch:
  - platform: restart
    name: esp-cam restart
binary_sensor:
  - platform: status
    name: esp-cam status
```