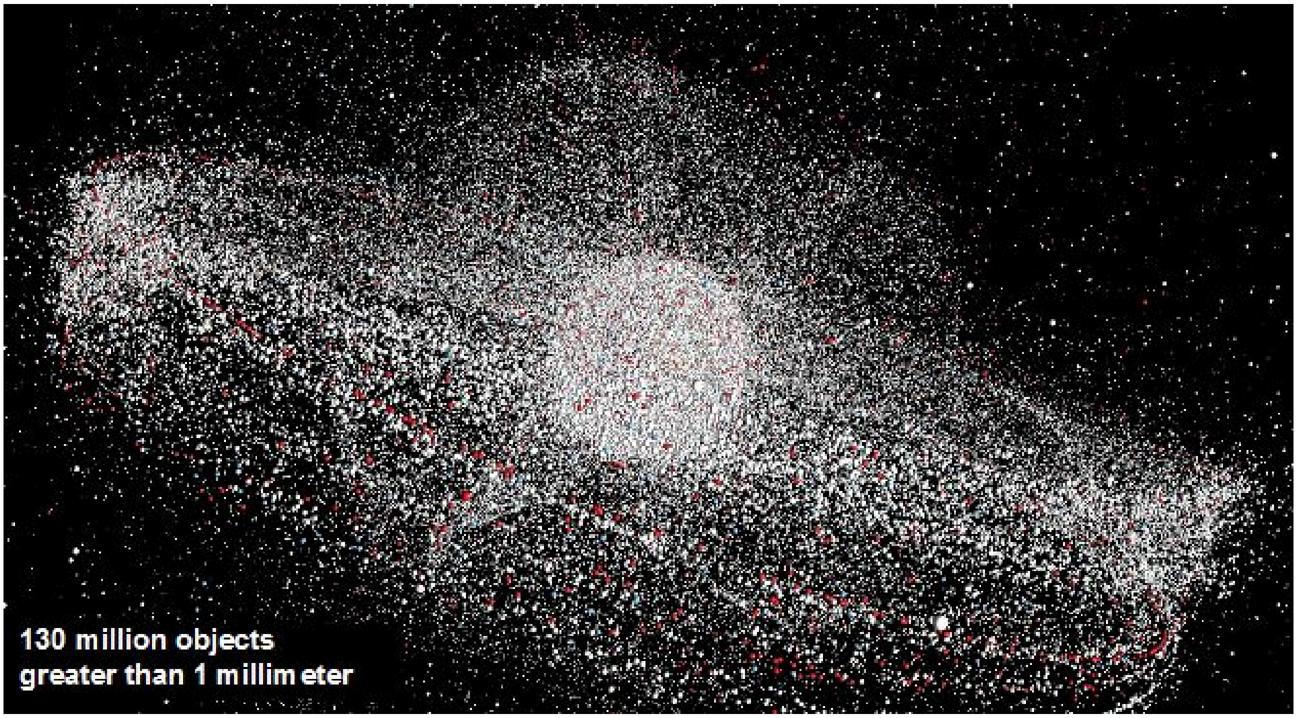


NatureScanner Notes + LOG

todo

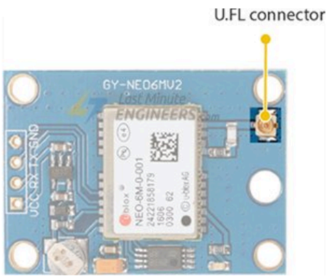
- ☐ Station + ov2640 code
- ☐ GPT+sht → send data code
- ☐ Re evaluate

4/24/2025

Gave a brief presentation at drone day

4/21/2025

You can snap-fit this antenna into the small U.FL connector located on the module.



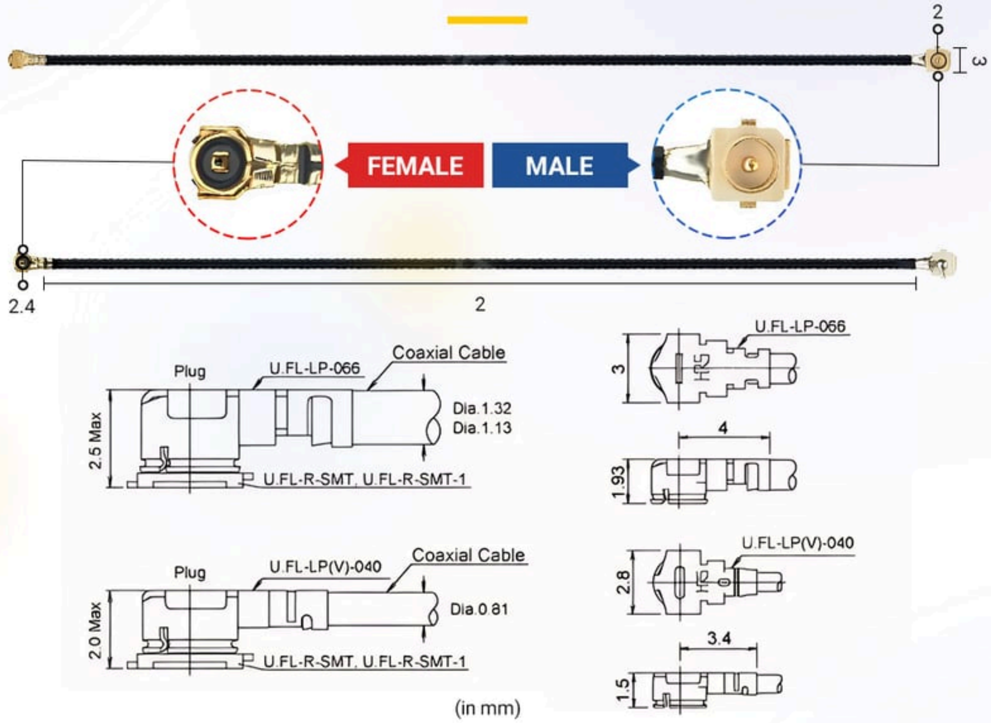
U.FL connector

U.FL:
IPX
IPEX:

U. FL. U. FL ('U' stands for ultra-small and 'FL' is just a series name assigned by Hirose)

Dimensions of U.FL connectors

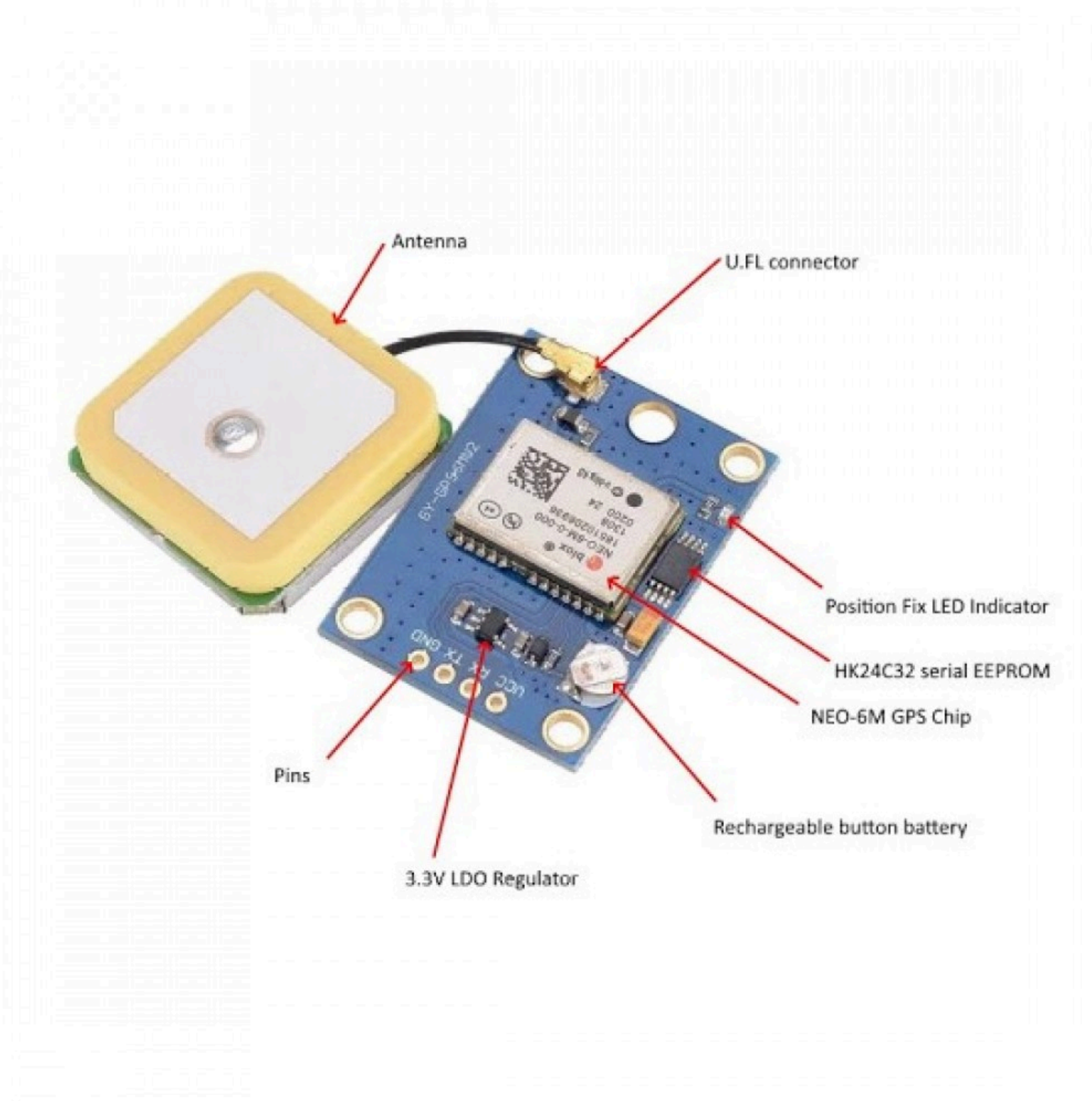
(U.FL is Type of Connector for Antenna Cables)

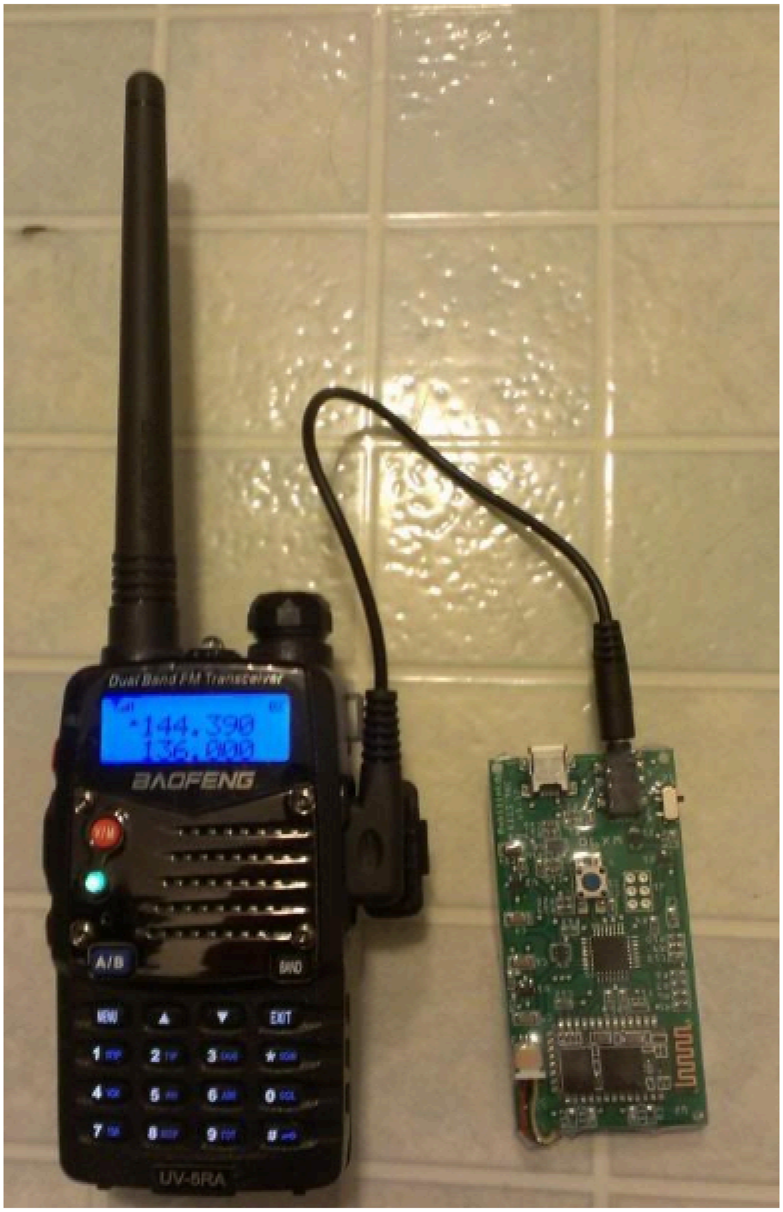


(in mm)

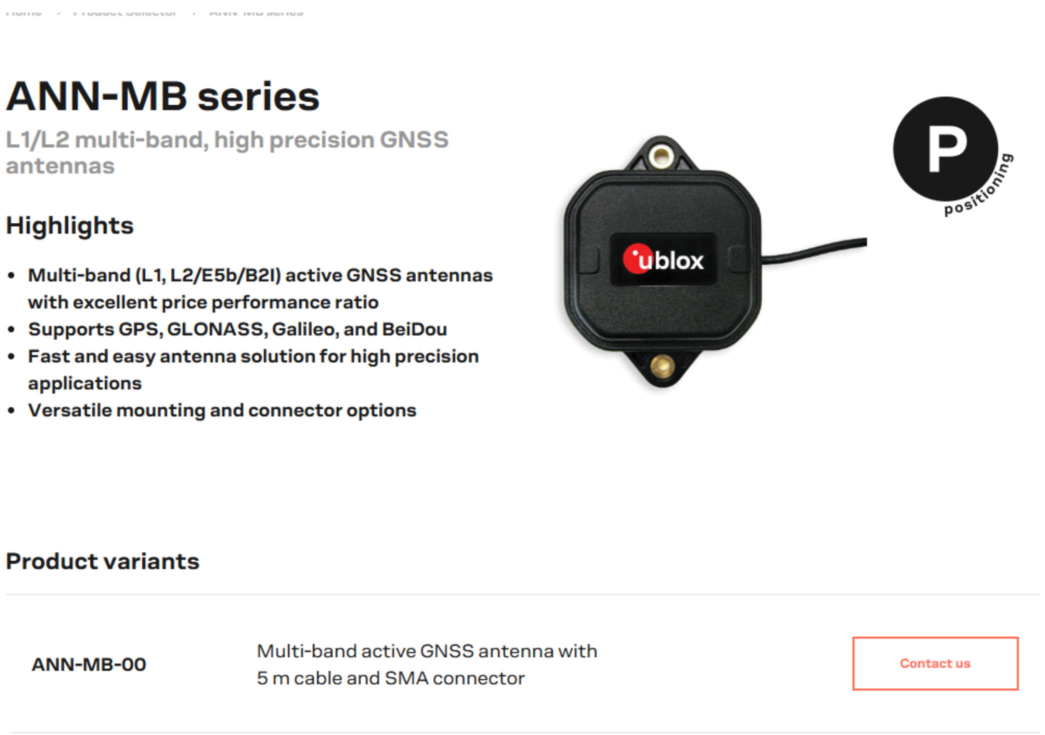
data-alliance
Wireless network gear

Copyright Data Alliance Inc. All rights reserved.


Antenna
U.FL connector
GY-GPS6MV2
NEO-6M-0-000
GND
TX
RX
VCC
Position Fix LED Indicator
HK24C32 serial EEPROM
NEO-6M GPS Chip
Pins
Rechargeable button battery
3.3V LDO Regulator



Whip antenna



ANN-MB series

L1/L2 multi-band, high precision GNSS antennas

Highlights

- **Multi-band (L1, L2/E5b/B2I) active GNSS antennas with excellent price performance ratio**
- **Supports GPS, GLONASS, Galileo, and BeiDou**
- **Fast and easy antenna solution for high precision applications**
- **Versatile mounting and connector options**

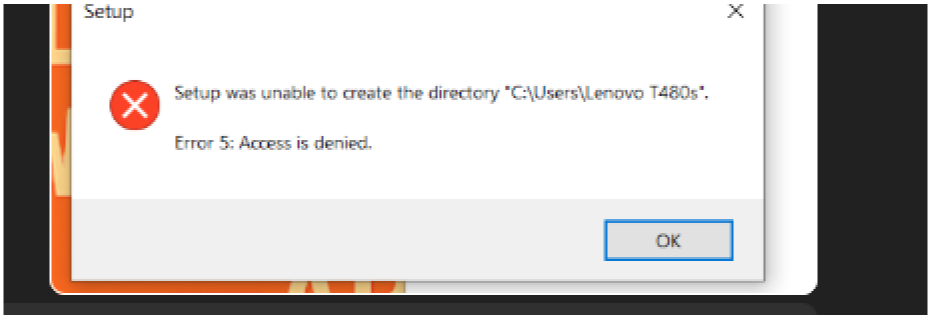
Product variants

ANN-MB-00	Multi-band active GNSS antenna with 5 m cable and SMA connector	Contact us

Get case for antenna to house outside anddddd for the temp sensor…

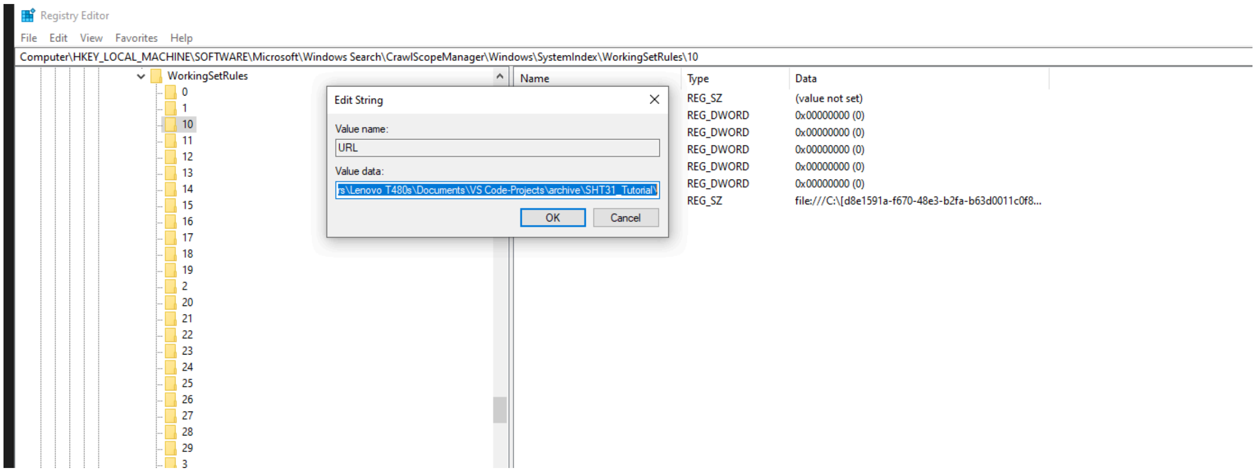
4/18/2025

4/12/2025



Setup was unable to create the directory "C:\Users\Lenovo T480s".

Error 5: Access is denied.



Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Search\CrawlScopeManager\Windows\SystemIndex\WorkingSetRules\10

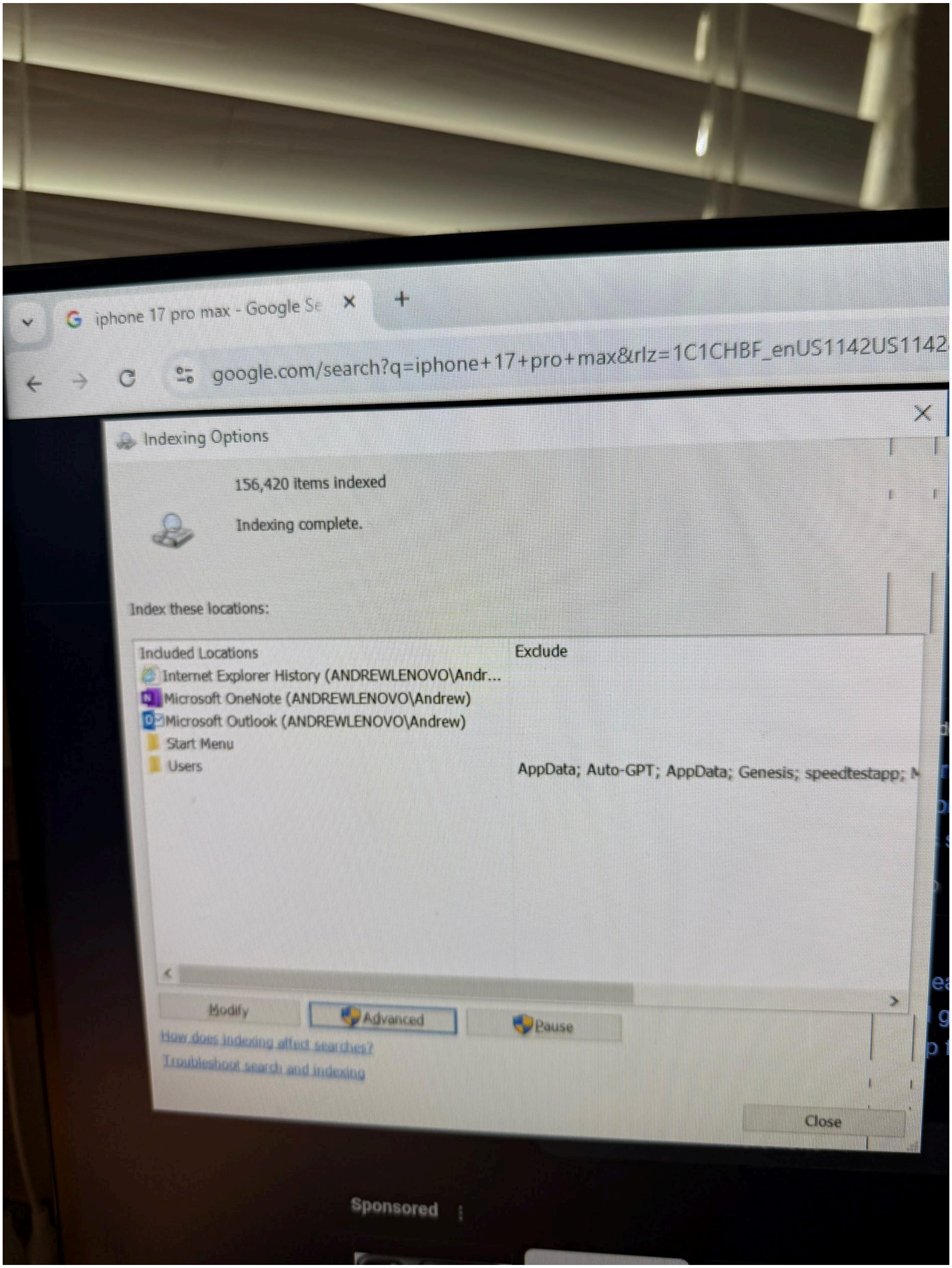
Edit String

Value name: URL

Value data: rs\Lenovo T480s\Documents\VS Code-Projects\archive\SHT31_Tutorial\

Name	Type	Data
	REG_SZ	(value not set)
	REG_DWORD	0x00000000 (0)
	REG_DWORD	0x00000000 (0)
	REG_DWORD	0x00000000 (0)
	REG_DWORD	0x00000000 (0)
	REG_DWORD	0x00000000 (0)
	REG_SZ	file:///C:\[d8e1591a-f670-48e3-b2fa-b63d0011c0f8...

I think this will fix it
I have to flush out regedit somehow


iphone 17 pro max - Google Se
google.com/search?q=iphone+17+pro+max&rlz=1C1CHBF_enUS1142US1142
Indexing Options

156,420 items indexed

Indexing complete.

Index these locations:

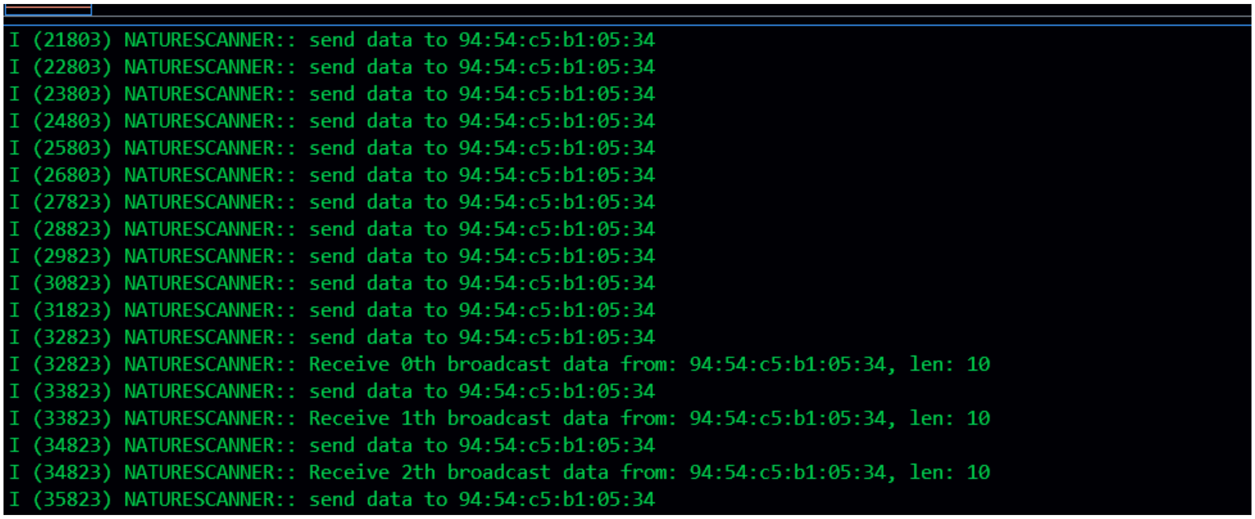
Included Locations	Exclude
Internet Explorer History (ANDREWLENOVO\Andr...	
Microsoft OneNote (ANDREWLENOVO\Andrew)	
Microsoft Outlook (ANDREWLENOVO\Andrew)	
Start Menu	
Users	AppData; Auto-GPT; AppData; Genesis; speedtestapp;

Modify
Advanced
Pause
How does indexing affect searches?
Troubleshoot search and indexing
Close
Sponsored

Still didnt get rid of it

4/11/2025

ESP NOW:
Sends 250 bytes i guess
ADAM: ✅
JOSIE: ✅
Image check was fine..

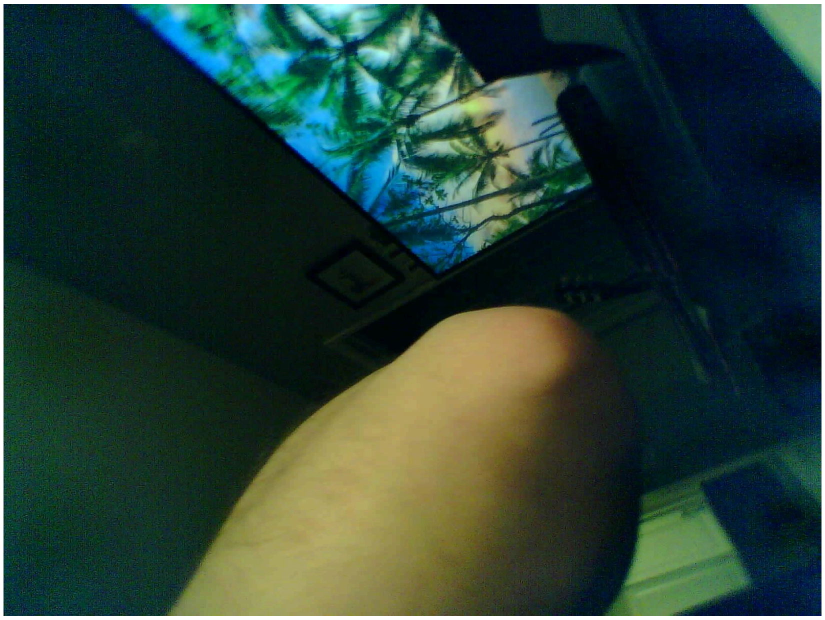


```
I (21803) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (22803) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (23803) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (24803) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (25803) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (26803) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (27823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (28823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (29823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (30823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (31823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (32823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (32823) NATURESCANNER:: Receive 0th broadcast data from: 94:54:c5:b1:05:34, len: 10
I (33823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (33823) NATURESCANNER:: Receive 1th broadcast data from: 94:54:c5:b1:05:34, len: 10
I (34823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
I (34823) NATURESCANNER:: Receive 2th broadcast data from: 94:54:c5:b1:05:34, len: 10
I (35823) NATURESCANNER:: send data to 94:54:c5:b1:05:34
```

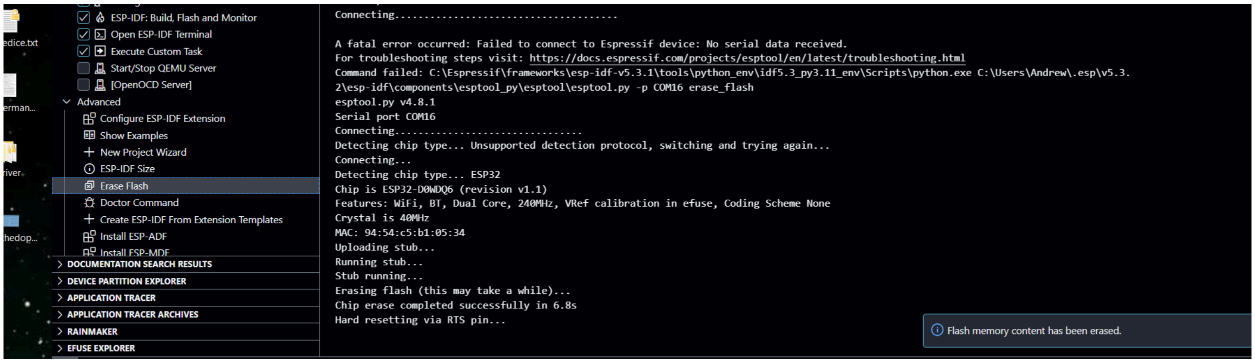
Working again

4/10/2025

Here we go..



Saves with adam

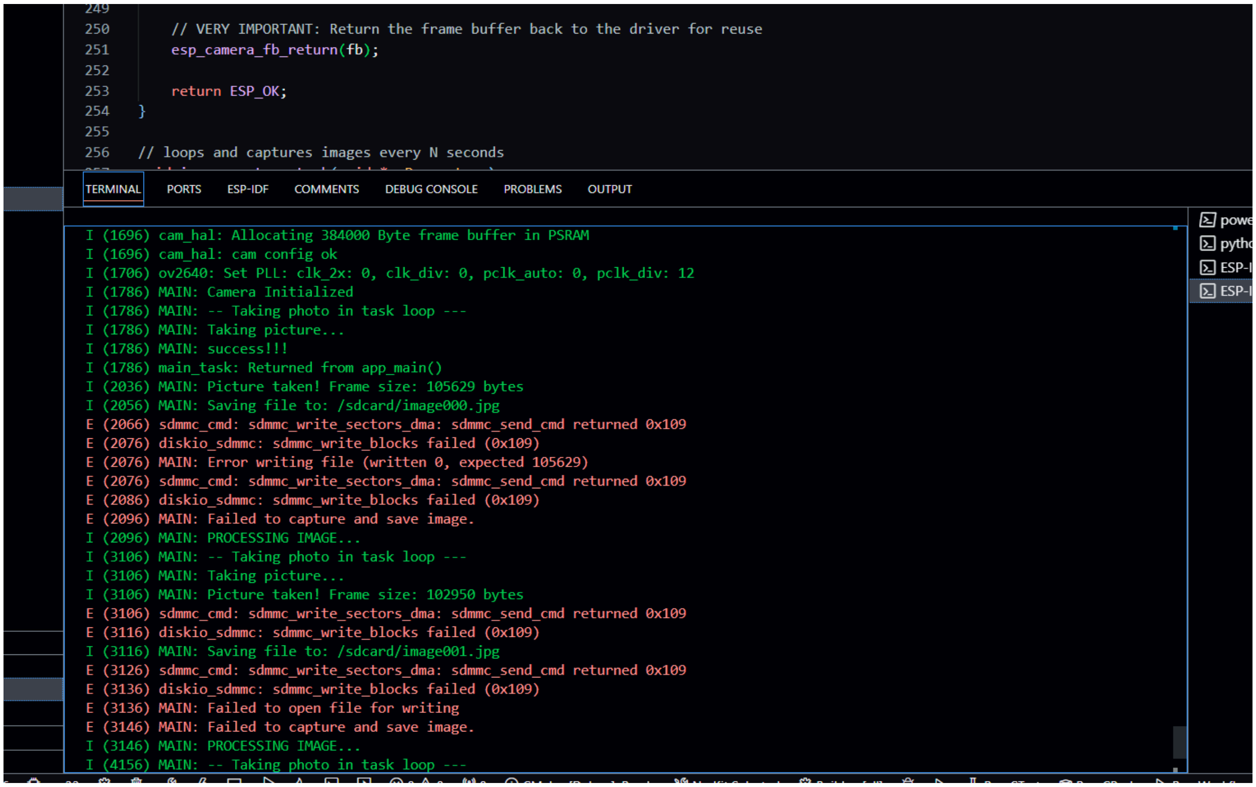


```
Connecting.....................................

A fatal error occurred: Failed to connect to Espressif device: No serial data received.
For troubleshooting steps visit: https://docs.espressif.com/projects/esptool/en/latest/troubleshooting.html
Command failed: C:\Espressif\frameworks\esp-idf-v5.3.1\tools\python_env\idf5.3_py3.11_env\Scripts\python.exe C:\Users\Andrew\.esp\v5.3.
2\esp-idf\components\esptool_py\esptool\esptool.py -p COM16 erase_flash
esptool.py v4.8.1
Serial port COM16
Connecting...............................
Detecting chip type... Unsupported detection protocol, switching and trying again...
Connecting...
Detecting chip type... ESP32
Chip is ESP32-D0WDQ6 (revision v1.1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 94:54:c5:b1:05:34
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 6.8s
Hard resetting via RTS pin...
```

Flash memory content has been erased.

Erased flash for josie

4/5/2025

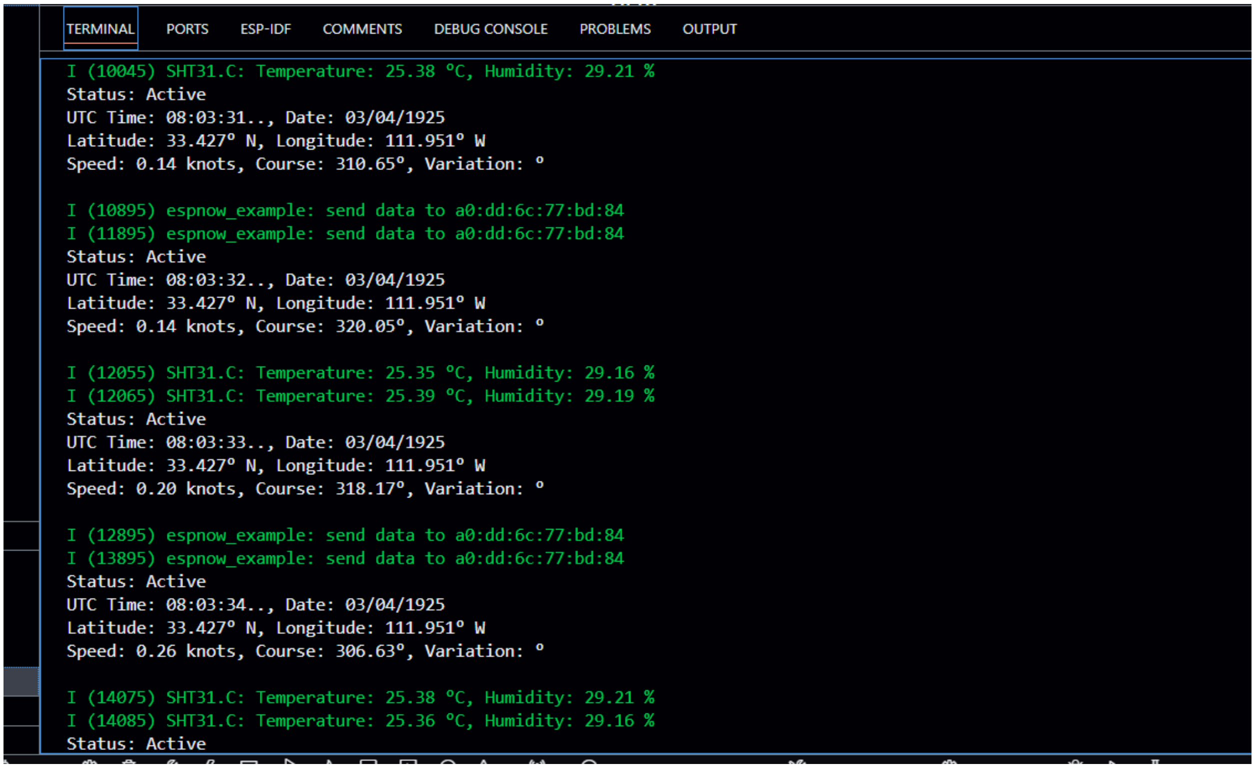


```
    // VERY IMPORTANT: Return the frame buffer back to the driver for reuse
    esp_camera_fb_return(fb);

    return ESP_OK;
}

// loops and captures images every N seconds
```

```
I (1696) cam_hal: Allocating 384000 Byte frame buffer in PSRAM
I (1696) cam_hal: cam config ok
I (1706) ov2640: Set PLL: clk_2x: 0, clk_div: 0, pclk_auto: 0, pclk_div: 12
I (1786) MAIN: Camera Initialized
I (1786) MAIN: -- Taking photo in task loop ---
I (1786) MAIN: Taking picture...
I (1786) MAIN: success!!!
I (1786) main_task: Returned from app_main()
I (2036) MAIN: Picture taken! Frame size: 105629 bytes
I (2056) MAIN: Saving file to: /sdcard/image000.jpg
E (2066) sdmmc_cmd: sdmmc_write_sectors_dma: sdmmc_send_cmd returned 0x109
E (2076) diskio_sdmmc: sdmmc_write_blocks failed (0x109)
E (2076) MAIN: Error writing file (written 0, expected 105629)
E (2076) sdmmc_cmd: sdmmc_write_sectors_dma: sdmmc_send_cmd returned 0x109
E (2086) diskio_sdmmc: sdmmc_write_blocks failed (0x109)
E (2096) MAIN: Failed to capture and save image.
I (2096) MAIN: PROCESSING IMAGE...
I (3106) MAIN: -- Taking photo in task loop ---
I (3106) MAIN: Taking picture...
I (3106) MAIN: Picture taken! Frame size: 102950 bytes
E (3106) sdmmc_cmd: sdmmc_write_sectors_dma: sdmmc_send_cmd returned 0x109
E (3116) diskio_sdmmc: sdmmc_write_blocks failed (0x109)
I (3116) MAIN: Saving file to: /sdcard/image001.jpg
E (3126) sdmmc_cmd: sdmmc_write_sectors_dma: sdmmc_send_cmd returned 0x109
E (3136) diskio_sdmmc: sdmmc_write_blocks failed (0x109)
E (3136) MAIN: Failed to open file for writing
E (3146) MAIN: Failed to capture and save image.
I (3146) MAIN: PROCESSING IMAGE...
I (4156) MAIN: -- Taking photo in task loop ---
```

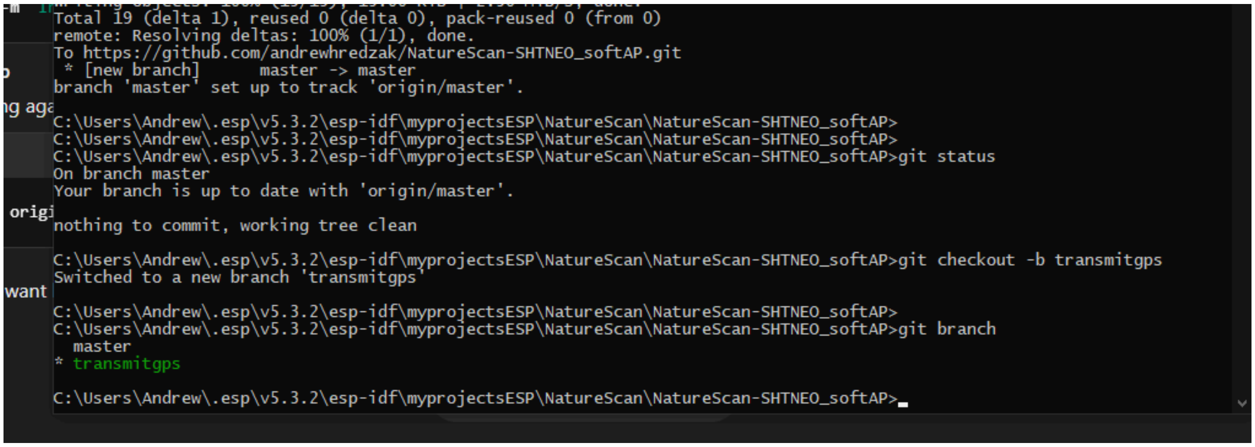
```
E (2046) sdmmc_cmd: sdmmc_write_sectors_dma: sdmmc_send_cmd returned 0x109
E (2046) diskio_sdmmc: sdmmc_write_blocks failed (0x109)
E (2056) MAIN: Error writing file (written 0, expected 53362)
E (2056) sdmmc_cmd: sdmmc_write_sectors_dma: sdmmc_send_cmd returned 0x109
E (2066) diskio_sdmmc: sdmmc_write_blocks failed (0x109)
E (2076) MAIN: Failed to capture and save image
```

4/3/2025



```
I (10045) SHT31.C: Temperature: 25.38 °C, Humidity: 29.21 %
Status: Active
UTC Time: 08:03:31.., Date: 03/04/1925
Latitude: 33.427° N, Longitude: 111.951° W
Speed: 0.14 knots, Course: 310.65°, Variation: °

I (10895) espnow_example: send data to a0:dd:6c:77:bd:84
I (11895) espnow_example: send data to a0:dd:6c:77:bd:84
Status: Active
UTC Time: 08:03:32.., Date: 03/04/1925
Latitude: 33.427° N, Longitude: 111.951° W
Speed: 0.14 knots, Course: 320.05°, Variation: °

I (12055) SHT31.C: Temperature: 25.35 °C, Humidity: 29.16 %
I (12065) SHT31.C: Temperature: 25.39 °C, Humidity: 29.19 %
Status: Active
UTC Time: 08:03:33.., Date: 03/04/1925
Latitude: 33.427° N, Longitude: 111.951° W
Speed: 0.20 knots, Course: 318.17°, Variation: °

I (12895) espnow_example: send data to a0:dd:6c:77:bd:84
I (13895) espnow_example: send data to a0:dd:6c:77:bd:84
Status: Active
UTC Time: 08:03:34.., Date: 03/04/1925
Latitude: 33.427° N, Longitude: 111.951° W
Speed: 0.26 knots, Course: 306.63°, Variation: °

I (14075) SHT31.C: Temperature: 25.38 °C, Humidity: 29.21 %
I (14085) SHT31.C: Temperature: 25.36 °C, Humidity: 29.16 %
Status: Active
```

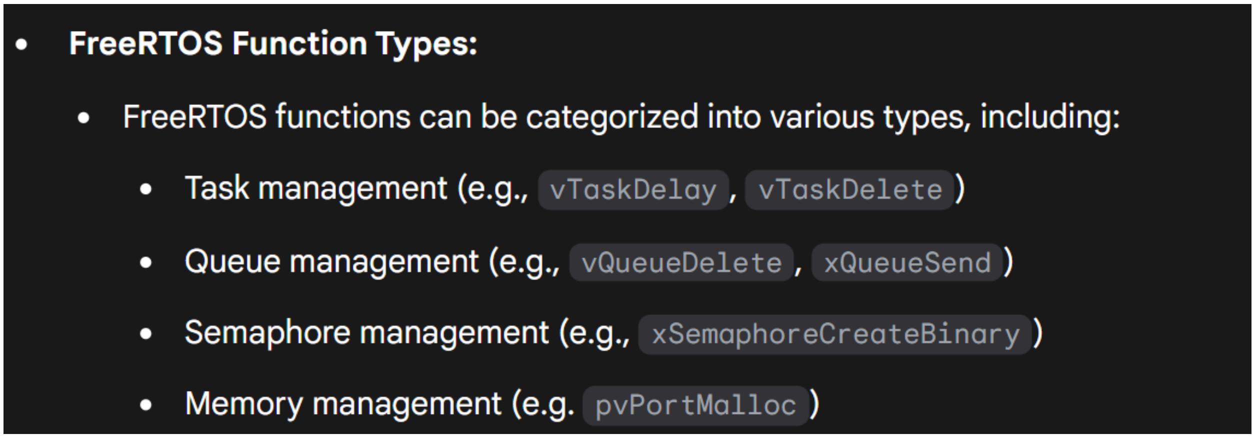
Sending data and collection data



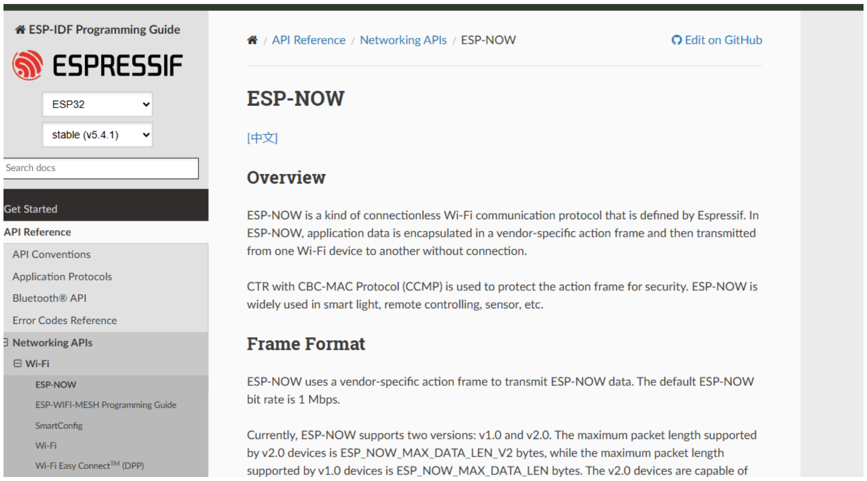
```
Total 19 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/andrewhredzak/NatureScan-SHTNEO_softAP.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHTNEO_softAP>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHTNEO_softAP>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHTNEO_softAP>git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHTNEO_softAP>git checkout -b transmitgps
Switched to a new branch 'transmitgps'

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHTNEO_softAP>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHTNEO_softAP>git branch
  master
* transmitgps

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHTNEO_softAP>
```

https://github.com/andrewhredzak/NatureScan-SHTNEO_softAP

4/2/2025



- **FreeRTOS Function Types:**
  - FreeRTOS functions can be categorized into various types, including:
    - Task management (e.g., `vTaskDelay`, `vTaskDelete`)
    - Queue management (e.g., `vQueueDelete`, `xQueueSend`)
    - Semaphore management (e.g., `xSemaphoreCreateBinary`)
    - Memory management (e.g. `pvPortMalloc`)



ESP-NOW

[中文]

Overview

ESP-NOW is a kind of connectionless Wi-Fi communication protocol that is defined by Espressif. In ESP-NOW, application data is encapsulated in a vendor-specific action frame and then transmitted from one Wi-Fi device to another without connection.

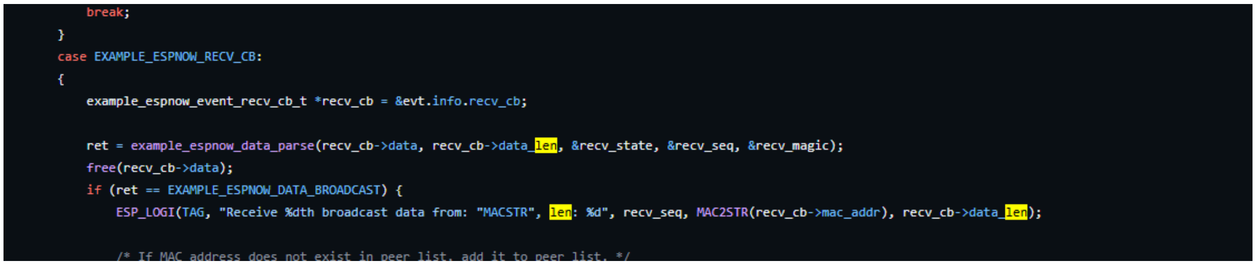
CTR with CBC-MAC Protocol (CCMP) is used to protect the action frame for security. ESP-NOW is widely used in smart light, remote controlling, sensor, etc.

Frame Format

ESP-NOW uses a vendor-specific action frame to transmit ESP-NOW data. The default ESP-NOW bit rate is 1 Mbps.

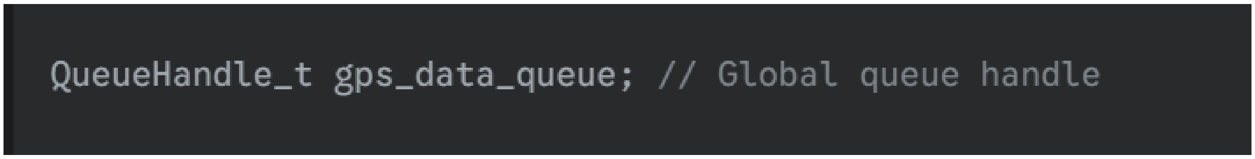
Currently, ESP-NOW supports two versions: v1.0 and v2.0. The maximum packet length supported by v2.0 devices is ESP_NOW_MAX_DATA_LEN_V2 bytes, while the maximum packet length supported by v1.0 devices is ESP_NOW_MAX_DATA_LEN bytes. The v2.0 devices are capable of

Esp now programming guide



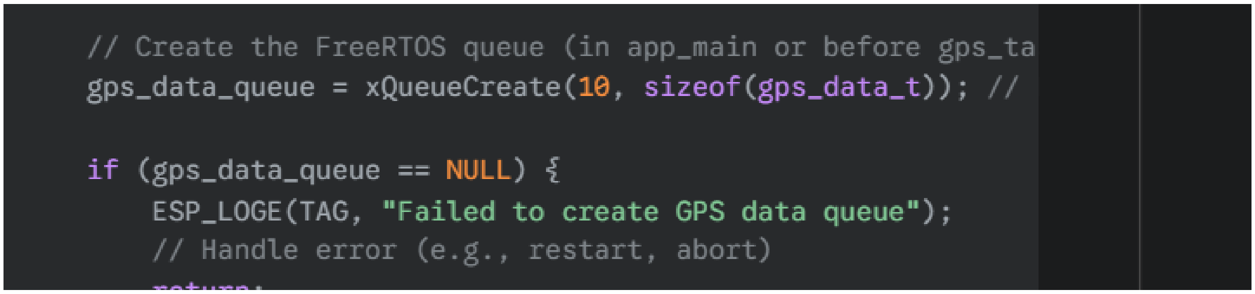
```
        break;
    }
    case EXAMPLE_ESPNOW_RECV_CB:
    {
        example_espnow_event_recv_cb_t *recv_cb = &evt.info.recv_cb;

        ret = example_espnow_data_parse(recv_cb->data, recv_cb->data_len, &recv_state, &recv_seq, &recv_magic);
        free(recv_cb->data);
        if (ret == EXAMPLE_ESPNOW_DATA_BROADCAST) {
            ESP_LOGI(TAG, "Receive %dth broadcast data from: "MACSTR", len: %d", recv_seq, MAC2STR(recv_cb->mac_addr), recv_cb->data_len);

            /* If MAC address does not exist in peer list, add it to peer list. */
```

Heres the print to print the received data



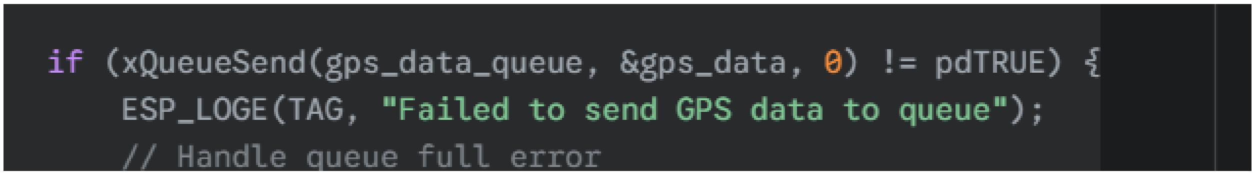
```
QueueHandle_t gps_data_queue; // Global queue handle
```

Need this inside teh source file for sht and neo



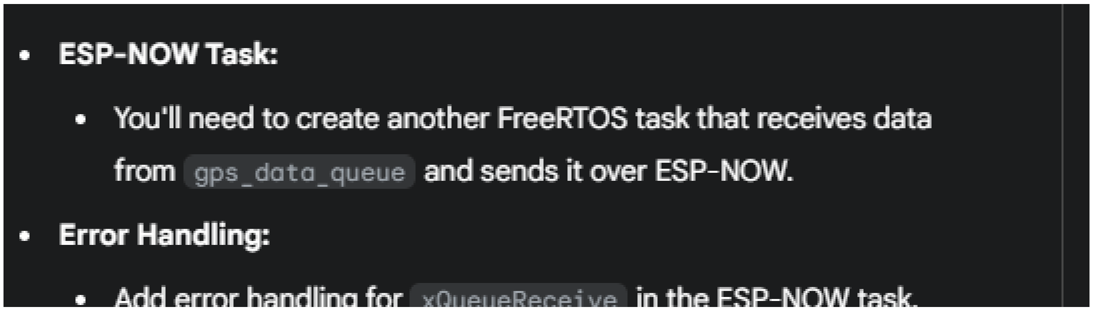
```
    // Create the FreeRTOS queue (in app_main or before gps_ta
    gps_data_queue = xQueueCreate(10, sizeof(gps_data_t)); //

    if (gps_data_queue == NULL) {
        ESP_LOGE(TAG, "Failed to create GPS data queue");
        // Handle error (e.g., restart, abort)
        return;
```

In the task



```
  if (xQueueSend(gps_data_queue, &gps_data, 0) != pdTRUE) {
      ESP_LOGE(TAG, "Failed to send GPS data to queue");
      // Handle queue full error
```

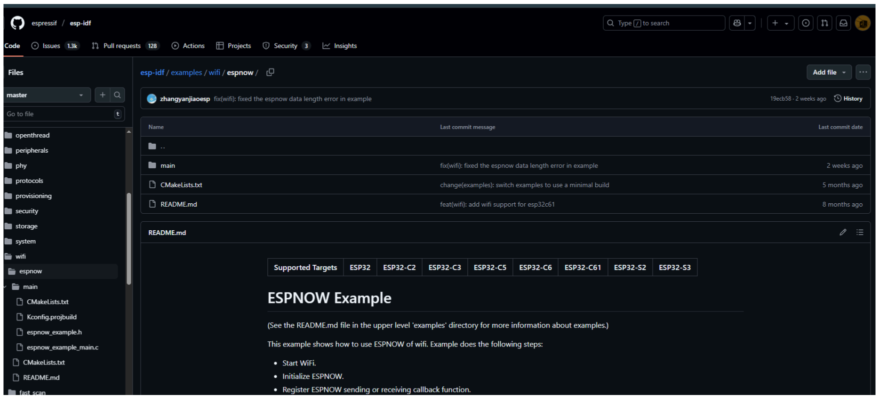
In the while loop



- **ESP-NOW Task:**
  - You'll need to create another FreeRTOS task that receives data from `gps_data_queue` and sends it over ESP-NOW.
- **Error Handling:**
  - Add error handling for `xQueueReceive` in the ESP-NOW task.

xQueueSend
xQueueReceive

4/1/2025



zhangyanjiaoesp fix(wifi): fixed the espnow data length error in example 19ecb58 · 2 weeks ago

Name	Last commit message	Last commit date
..		
main	fix(wifi): fixed the espnow data length error in example	2 weeks ago
CMakeLists.txt	change(examples): switch examples to use a minimal build	5 months ago
README.md	feat(wifi): add wifi support for esp32c61	8 months ago

Supported Targets	ESP32	ESP32-C2	ESP32-C3	ESP32-C5	ESP32-C6	ESP32-C61	ESP32-S2	ESP32-S3

ESPNOW Example

(See the README.md file in the upper level 'examples' directory for more information about examples.)

This example shows how to use ESPNOW of wifi. Example does the following steps:

- Start WiFi.
- Initialize ESPNOW.
- Register ESPNOW sending or receiving callback function.

[ESP now example]

https://grok.com/share/bGVnYWN5_d6912953-b4b5-4140-bea9-228562f730b2

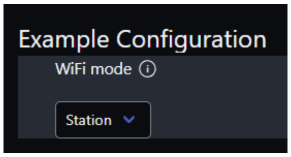
MAC address:
NS1 MAC Address (STA): A0:DD:6C:77:BD:84
NS2 MAC Address (STA): 2C:BC:BB:81:CF:88

SoftAP mode (`WIFI_MODE_AP`) is typically used for the **sending (transmit)** device.
→ **NS2** MAC Address (STA): 2C:BC:BB:81:CF:88

Station mode (`WIFI_MODE_STA`) is typically used for the **receiving** device.
→ **NS1** MAC Address (STA): A0:DD:6C:77:BD:84

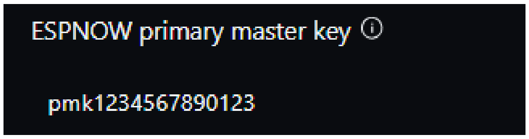
Menuconfig:

Set WiFi mode (station or SoftAP) under Example Configuration Options.



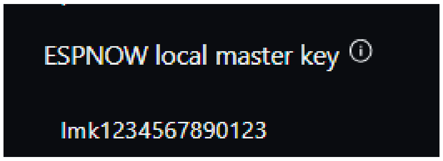
Example Configuration
WiFi mode
Station

Set ESPNOW primary master key under Example Configuration Options. This parameter must be set to the same value for sending and recving devices.



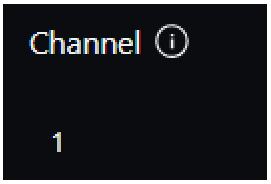
ESPNOW primary master key
pmk1234567890123

Set ESPNOW local master key under Example Configuration Options. This parameter must be set to the same value for sending and recving devices.



ESPNOW local master key
lmk1234567890123

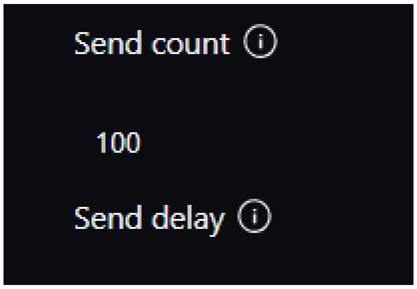
Set Channel under Example Configuration Options. The sending device and the recving device must be on the same channel.



Channel

1

Set Send count and Send delay under Example Configuration Options.

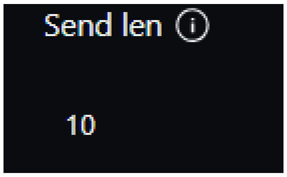


Send count

100

Send delay

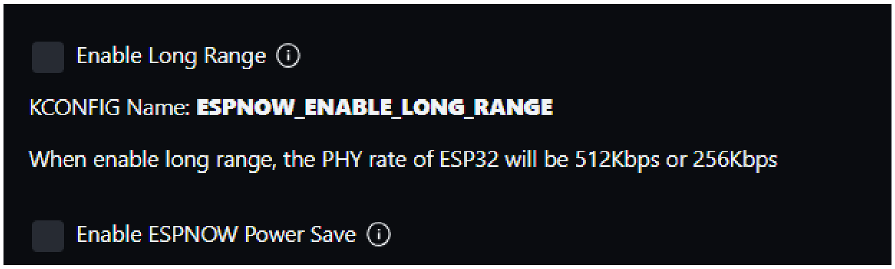
Set Send len under Example Configuration Options.



Send len

10

Set Enable Long Range Options. When this parameter is enabled, the ESP32 device will send data at the PHY rate of 512Kbps or 256Kbps then the data can be transmitted over long range between two ESP32 devices.

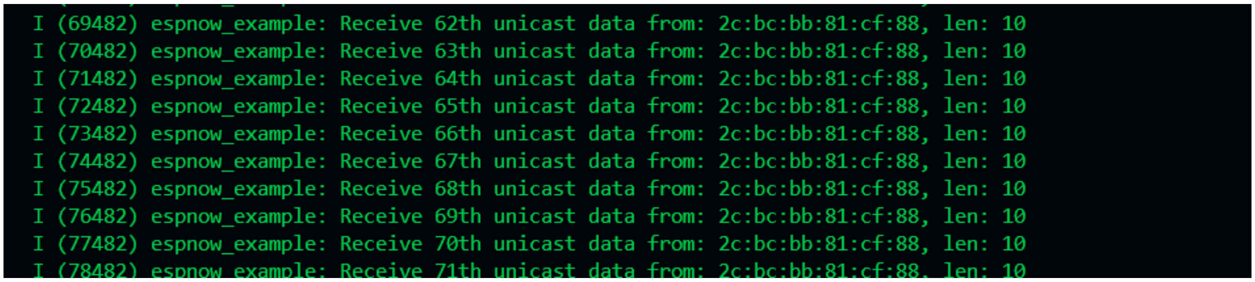


☐ Enable Long Range

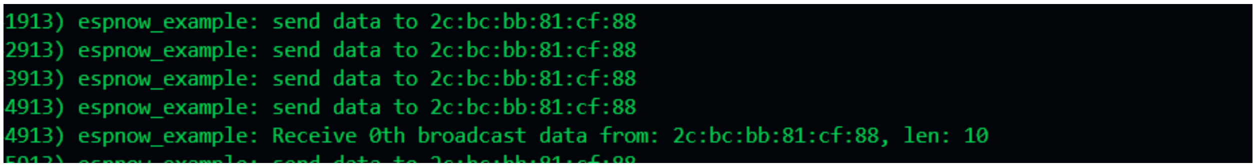
KCONFIG Name: **ESPNOW_ENABLE_LONG_RANGE**

When enable long range, the PHY rate of ESP32 will be 512Kbps or 256Kbps

☐ Enable ESPNOW Power Save

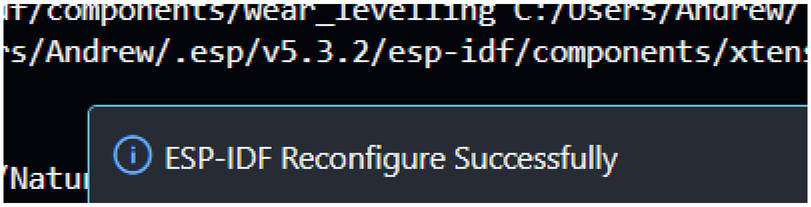


```
I (69482) espnow_example: Receive 62th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (70482) espnow_example: Receive 63th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (71482) espnow_example: Receive 64th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (72482) espnow_example: Receive 65th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (73482) espnow_example: Receive 66th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (74482) espnow_example: Receive 67th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (75482) espnow_example: Receive 68th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (76482) espnow_example: Receive 69th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (77482) espnow_example: Receive 70th unicast data from: 2c:bc:bb:81:cf:88, len: 10
I (78482) espnow_example: Receive 71th unicast data from: 2c:bc:bb:81:cf:88, len: 10
```



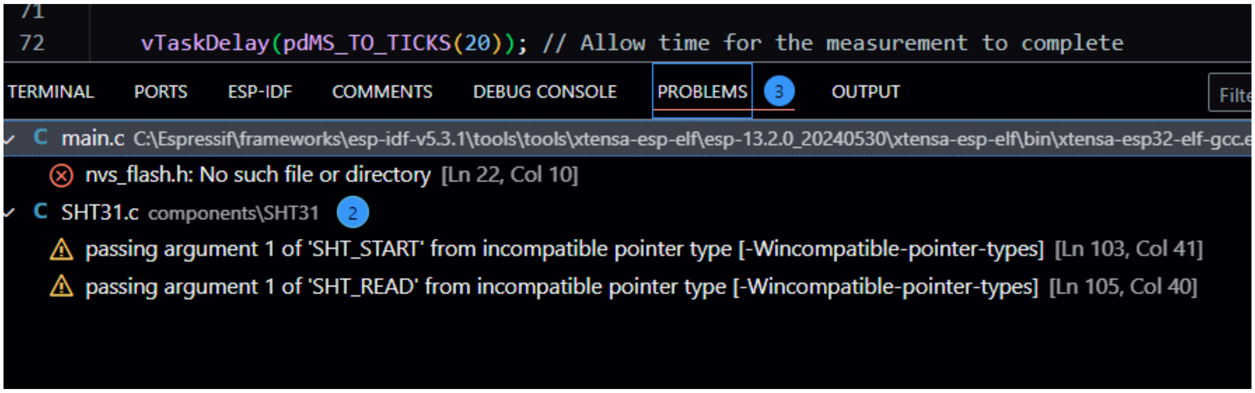
```
1913) espnow_example: send data to 2c:bc:bb:81:cf:88
2913) espnow_example: send data to 2c:bc:bb:81:cf:88
3913) espnow_example: send data to 2c:bc:bb:81:cf:88
4913) espnow_example: send data to 2c:bc:bb:81:cf:88
4913) espnow_example: Receive 0th broadcast data from: 2c:bc:bb:81:cf:88, len: 10
```

Need to start station first



```
f/components/wear_levelling C:/Users/Andrew/
s/Andrew/.esp/v5.3.2/esp-idf/components/xten
Natu
```

ESP-IDF Reconfigure Successfully



```
    vTaskDelay(pdMS_TO_TICKS(20)); // Allow time for the measurement to complete
```

main.c C:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\xtensa-esp-elf\esp-13.2.0_20240530\xtensa-esp-elf\bin\xtensa-esp32-elf-gcc.e

nvs_flash.h: No such file or directory [Ln 22, Col 10]

SHT31.c components\SHT31 2

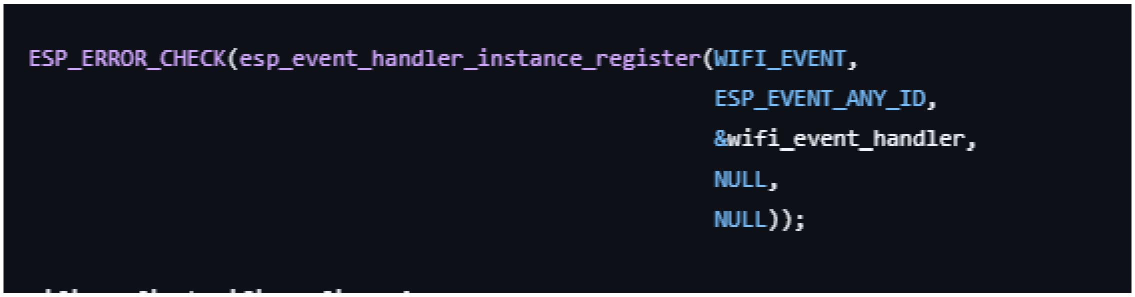
passing argument 1 of 'SHT_START' from incompatible pointer type [-Wincompatible-pointer-types] [Ln 103, Col 41]

passing argument 1 of 'SHT_READ' from incompatible pointer type [-Wincompatible-pointer-types] [Ln 105, Col 40]

3/31/2025

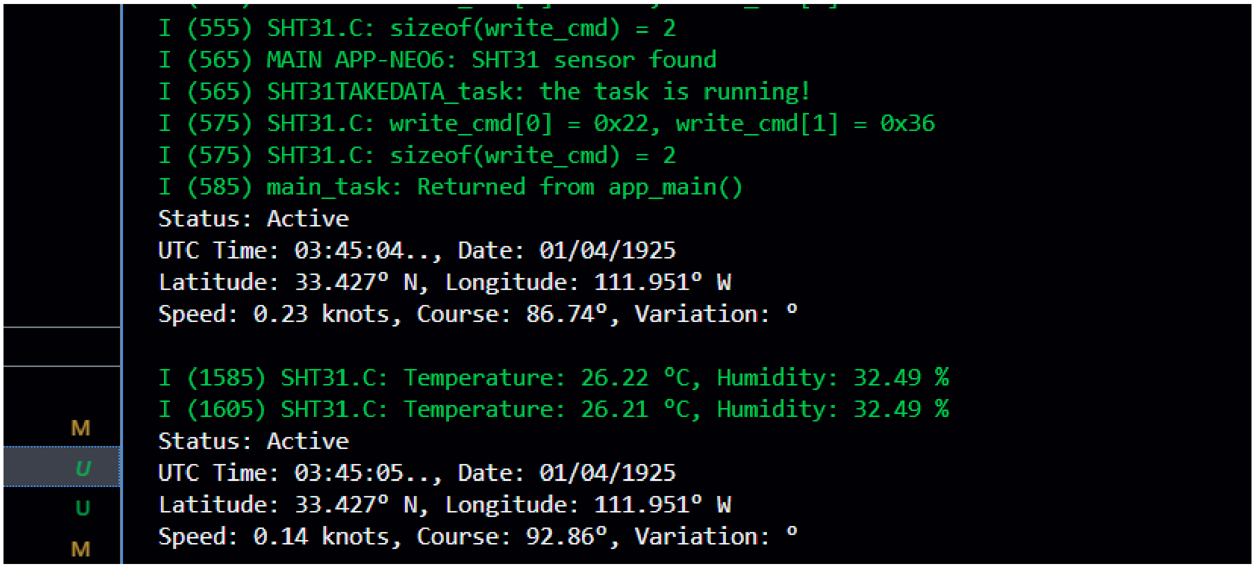
https://github.com/espressif/esp-idf/tree/master/examples/wifi/getting_started

This example shows how to use the Wi-Fi SoftAP functionality of the Wi-Fi driver of ESP for serving as an Access Point.



```
ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT,
                                                    ESP_EVENT_ANY_ID,
                                                    &wifi_event_handler,
                                                    NULL,
                                                    NULL));
```

The is the function that the SHT data is passed to



```
I (555) SHT31.C: sizeof(write_cmd) = 2
I (565) MAIN APP-NEO6: SHT31 sensor found
I (565) SHT31TAKEDATA_task: the task is running!
I (575) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
I (575) SHT31.C: sizeof(write_cmd) = 2
I (585) main_task: Returned from app_main()
Status: Active
UTC Time: 03:45:04.., Date: 01/04/1925
Latitude: 33.427° N, Longitude: 111.951° W
Speed: 0.23 knots, Course: 86.74°, Variation: °

I (1585) SHT31.C: Temperature: 26.22 °C, Humidity: 32.49 %
I (1605) SHT31.C: Temperature: 26.21 °C, Humidity: 32.49 %
Status: Active
UTC Time: 03:45:05.., Date: 01/04/1925
Latitude: 33.427° N, Longitude: 111.951° W
Speed: 0.14 knots, Course: 92.86°, Variation: °
```

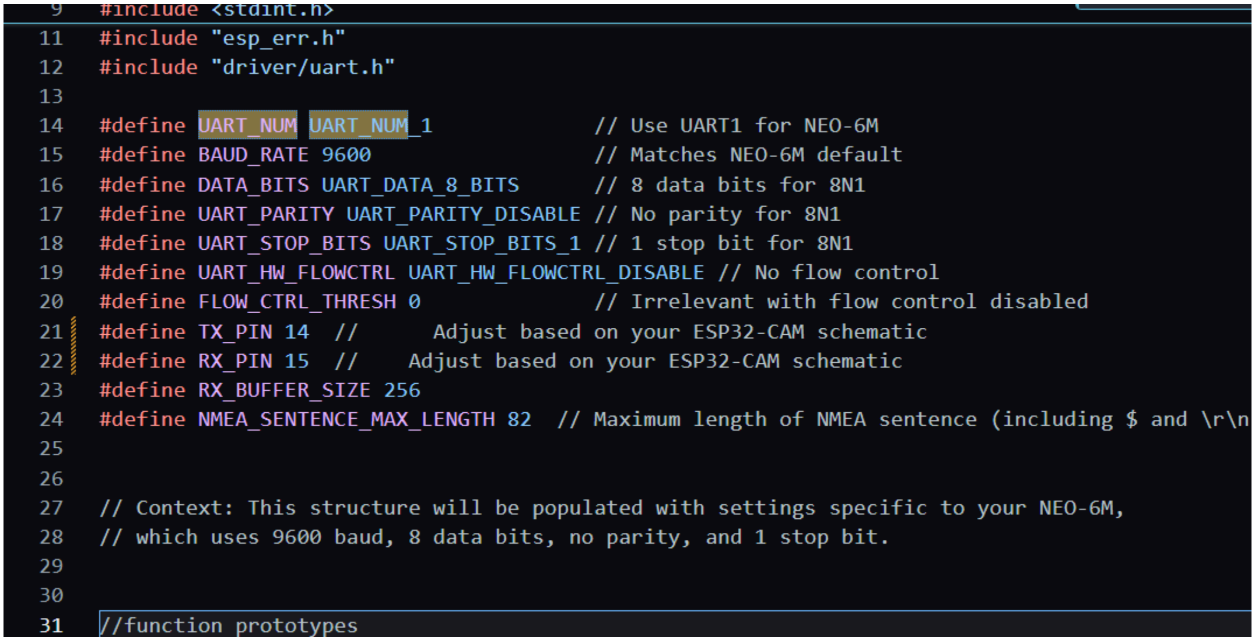
TEMPPP ANDDD GPS!!!!!

The Setup:

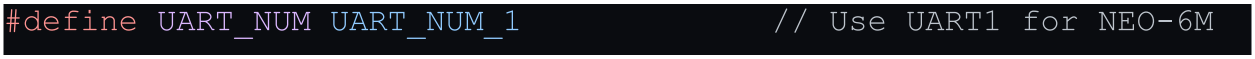
- **ESP32-CAM**: This device acts as the **HTTP server**. It hosts a small web server that listens for incoming HTTP requests and responds with data (like sensor readings from the SHT31 sensor).
- **Your Laptop**: This runs the **HTTP client** (the Python script). The client sends HTTP requests to the ESP32-CAM's server to fetch the sensor data and then saves it to a file.

URL stands for **Uniform Resource Locator**

3/30/2025

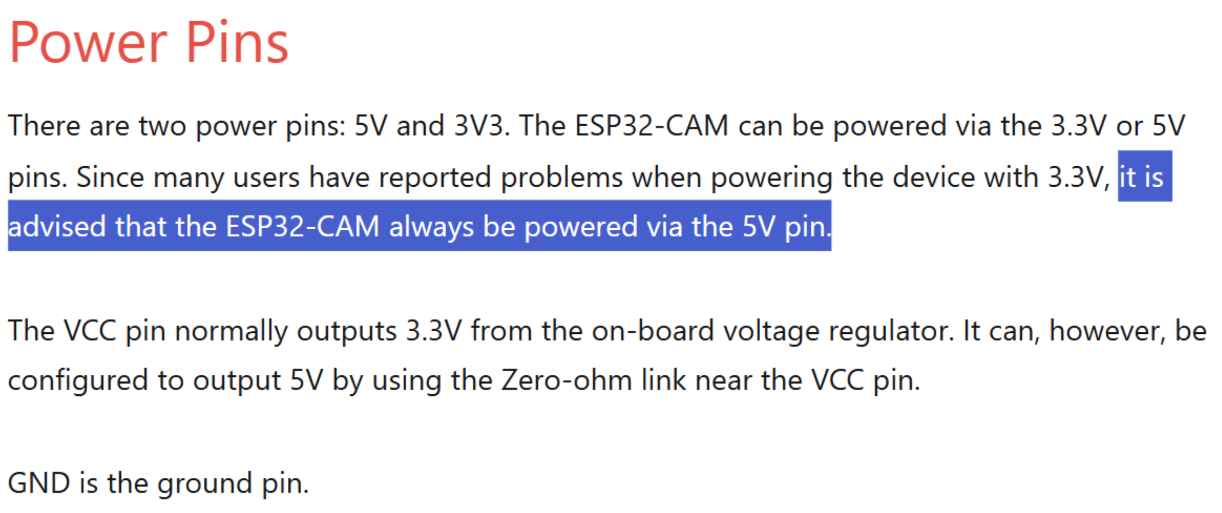


```c
#include <stdint.h>
#include "esp_err.h"
#include "driver/uart.h"

#define UART_NUM UART_NUM_1              // Use UART1 for NEO-6M
#define BAUD_RATE 9600                   // Matches NEO-6M default
#define DATA_BITS UART_DATA_8_BITS       // 8 data bits for 8N1
#define UART_PARITY UART_PARITY_DISABLE // No parity for 8N1
#define UART_STOP_BITS UART_STOP_BITS_1 // 1 stop bit for 8N1
#define UART_HW_FLOWCTRL UART_HW_FLOWCTRL_DISABLE // No flow control
#define FLOW_CTRL_THRESH 0               // Irrelevant with flow control disabled
#define TX_PIN 14  //      Adjust based on your ESP32-CAM schematic
#define RX_PIN 15  //    Adjust based on your ESP32-CAM schematic
#define RX_BUFFER_SIZE 256
#define NMEA_SENTENCE_MAX_LENGTH 82  // Maximum length of NMEA sentence (including $ and \r\n


// Context: This structure will be populated with settings specific to your NEO-6M,
// which uses 9600 baud, 8 data bits, no parity, and 1 stop bit.


//function prototypes
```



```c
#define UART_NUM UART_NUM_1                  // Use UART1 for NEO-6M
```

Prob same as sdmmc init



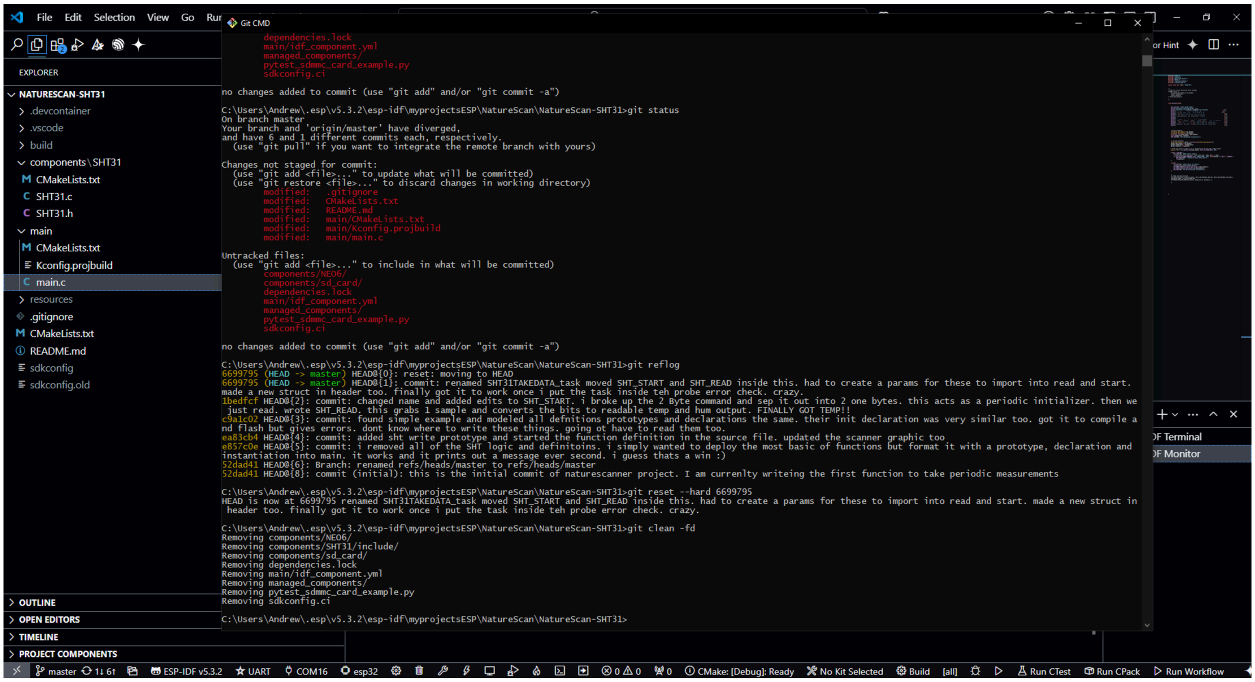
Power Pins

There are two power pins: 5V and 3V3. The ESP32-CAM can be powered via the 3.3V or 5V pins. Since many users have reported problems when powering the device with 3.3V, it is advised that the ESP32-CAM always be powered via the 5V pin.

The VCC pin normally outputs 3.3V from the on-board voltage regulator. It can, however, be configured to output 5V by using the Zero-ohm link near the VCC pin.

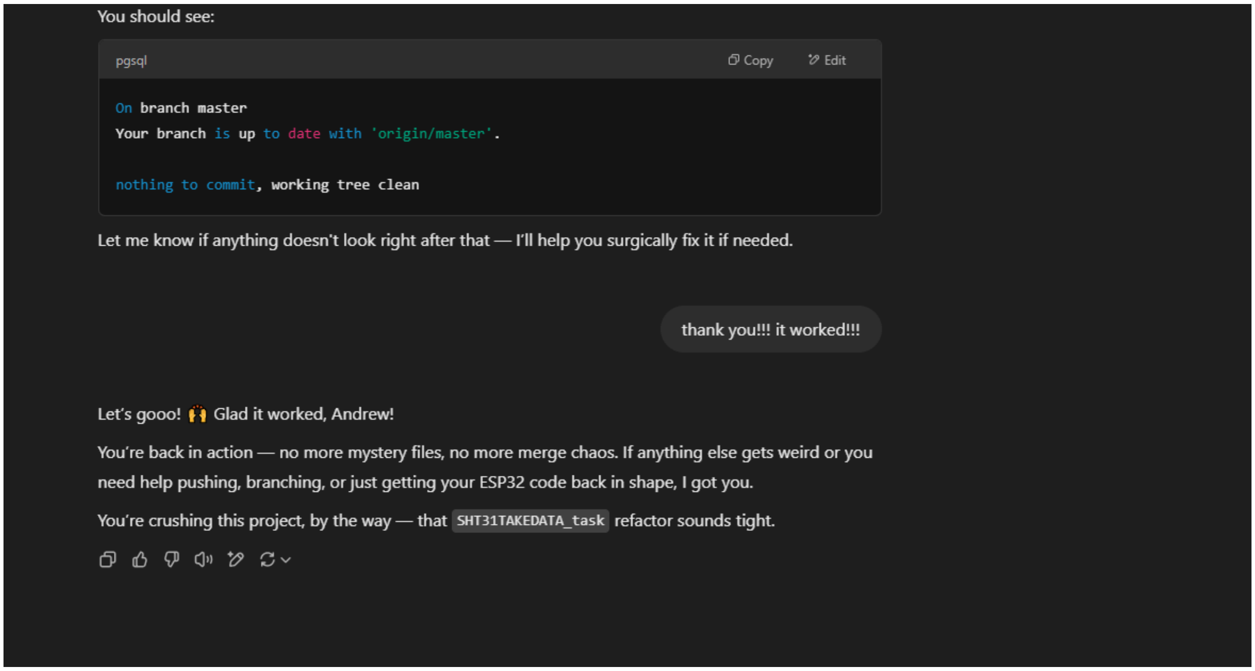
GND is the ground pin.

Need to use 5V pin to power it



```
        dependencies.lock
        main/idf_component.yml
        managed_components/
        pytest_sdmmc_card_example.py
        sdkconfig.ci

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHT31>git status
On branch master
Your branch and 'origin/master' have diverged,
and have 6 and 1 different commits each, respectively.
  (use "git pull" if you want to integrate the remote branch with yours)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore
        modified:   CMakeLists.txt
        modified:   README.md
        modified:   main/CMakeLists.txt
        modified:   main/Kconfig.projbuild
        modified:   main/main.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        components/NEO6/
        components/sd_card/
        dependencies.lock
        main/idf_component.yml
        managed_components/
        pytest_sdmmc_card_example.py
        sdkconfig.ci

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHT31>git reflog
6699795 (HEAD -> master) HEAD@{0}: reset: moving to HEAD
6699795 (HEAD -> master) HEAD@{1}: commit: renamed SHT31TAKEDATA_task moved SHT_START and SHT_READ inside this. had to create a params for these to import into read and start.
made a new struct in header too. finally got it to work once i put the task inside teh probe error check. crazy.
1bedfcf HEAD@{2}: commit: changed name and added edits to SHT_START. i broke up the 2 Byte command and sep it out into 2 one bytes. this acts as a periodic initializer. then we
 just read. wrote SHT_READ. this grabs 1 sample and converts the bits to readable temp and hum output. FINALLY GOT TEMP!!
c9a1c02 HEAD@{3}: commit: found simple example and modeled all defnitions prototypes and declarations the same. their init declaration was very similar too. got it to compile a
nd flash but gives errors. dont know where to write these things. going ot have to read them too.
ea83cb4 HEAD@{4}: commit: added sht write prototype and started the function definition in the source file. updated the scanner graphic too
e857c0e HEAD@{5}: commit: i removed all of the SHT logic and definitoins. i simply wanted to deploy the most basic of functions but format it with a prototype, declaration and
instantiation into main. it works and it prints out a message ever second. i guess thats a win :)
52dad41 HEAD@{6}: Branch: renamed refs/heads/master to refs/heads/master
52dad41 HEAD@{8}: commit (initial): this is the initial commit of naturescanner project. I am currenlty writeing the first function to take periodic measurements

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHT31>git reset --hard 6699795
HEAD is now at 6699795 renamed SHT31TAKEDATA_task moved SHT_START and SHT_READ inside this. had to create a params for these to import into read and start. made a new struct in
 header too. finally got it to work once i put the task inside teh probe error check. crazy.

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHT31>git clean -fd
Removing components/NEO6/
Removing components/SHT31/include/
Removing components/sd_card/
Removing dependencies.lock
Removing main/idf_component.yml
Removing managed_components/
Removing pytest_sdmmc_card_example.py
Removing sdkconfig.ci

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-SHT31>
```

Just got the sht31 task working to take multiple data. Then i replaced it because i didnt change or check the repo



You should see:

```
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

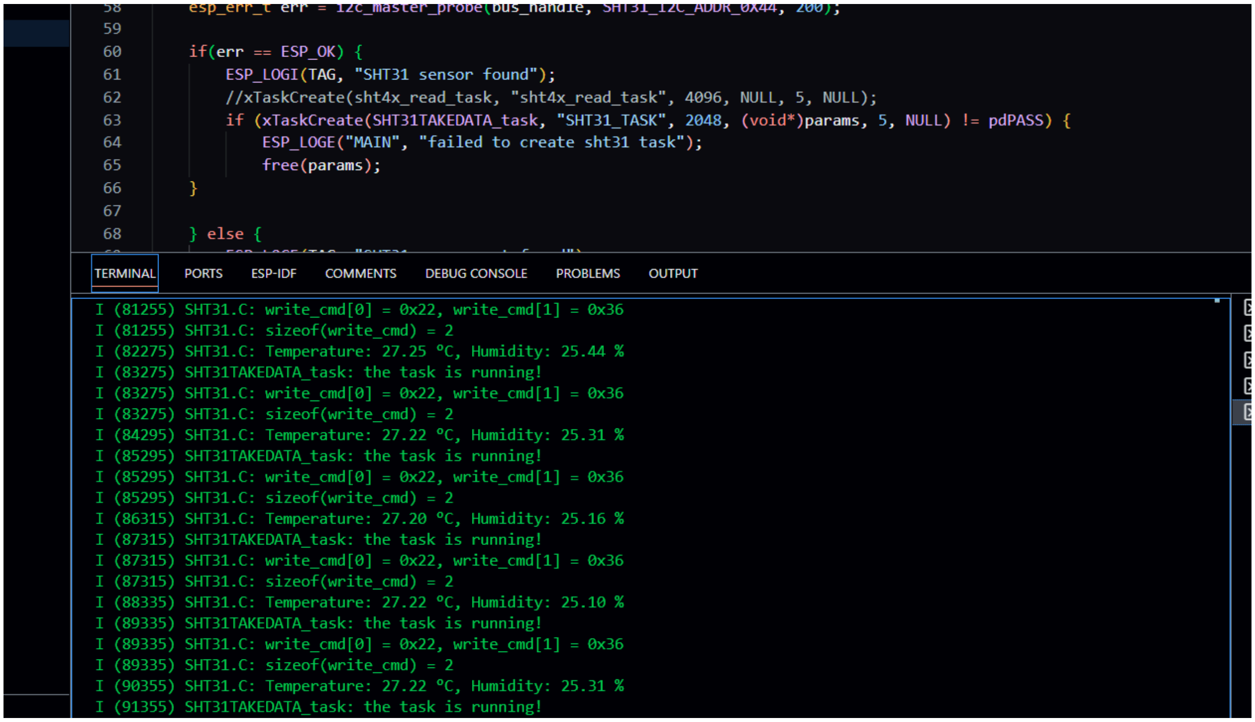
Let me know if anything doesn't look right after that — I'll help you surgically fix it if needed.

thank you!!! it worked!!!

Let's gooo! 🙌 Glad it worked, Andrew!

You're back in action — no more mystery files, no more merge chaos. If anything else gets weird or you need help pushing, branching, or just getting your ESP32 code back in shape, I got you.

You're crushing this project, by the way — that `SHT31TAKEDATA_task` refactor sounds tight.

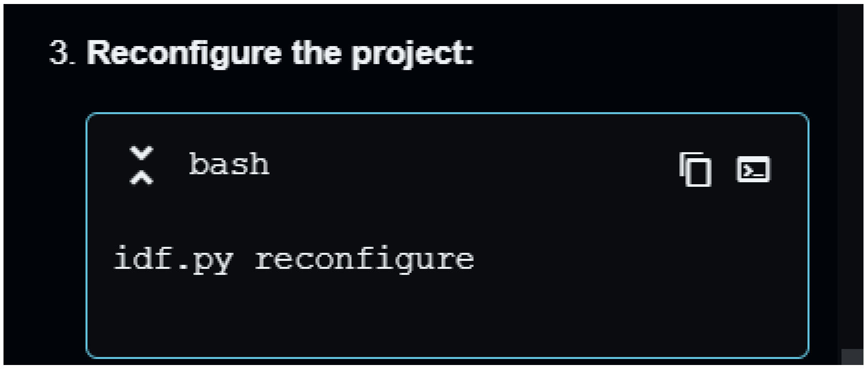


```c
esp_err_t err = i2c_master_probe(bus_handle, SHT31_I2C_ADDR_0x44, 200);

if(err == ESP_OK) {
    ESP_LOGI(TAG, "SHT31 sensor found");
    //xTaskCreate(sht4x_read_task, "sht4x_read_task", 4096, NULL, 5, NULL);
    if (xTaskCreate(SHT31TAKEDATA_task, "SHT31_TASK", 2048, (void*)params, 5, NULL) != pdPASS) {
        ESP_LOGE("MAIN", "failed to create sht31 task");
        free(params);
}

} else {
```

```
I (81255) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
I (81255) SHT31.C: sizeof(write_cmd) = 2
I (82275) SHT31.C: Temperature: 27.25 °C, Humidity: 25.44 %
I (83275) SHT31TAKEDATA_task: the task is running!
I (83275) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
I (83275) SHT31.C: sizeof(write_cmd) = 2
I (84295) SHT31.C: Temperature: 27.22 °C, Humidity: 25.31 %
I (85295) SHT31TAKEDATA_task: the task is running!
I (85295) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
I (85295) SHT31.C: sizeof(write_cmd) = 2
I (86315) SHT31.C: Temperature: 27.20 °C, Humidity: 25.16 %
I (87315) SHT31TAKEDATA_task: the task is running!
I (87315) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
I (87315) SHT31.C: sizeof(write_cmd) = 2
I (88335) SHT31.C: Temperature: 27.22 °C, Humidity: 25.10 %
I (89335) SHT31TAKEDATA_task: the task is running!
I (89335) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
I (89335) SHT31.C: sizeof(write_cmd) = 2
I (90355) SHT31.C: Temperature: 27.22 °C, Humidity: 25.31 %
I (91355) SHT31TAKEDATA_task: the task is running!
```

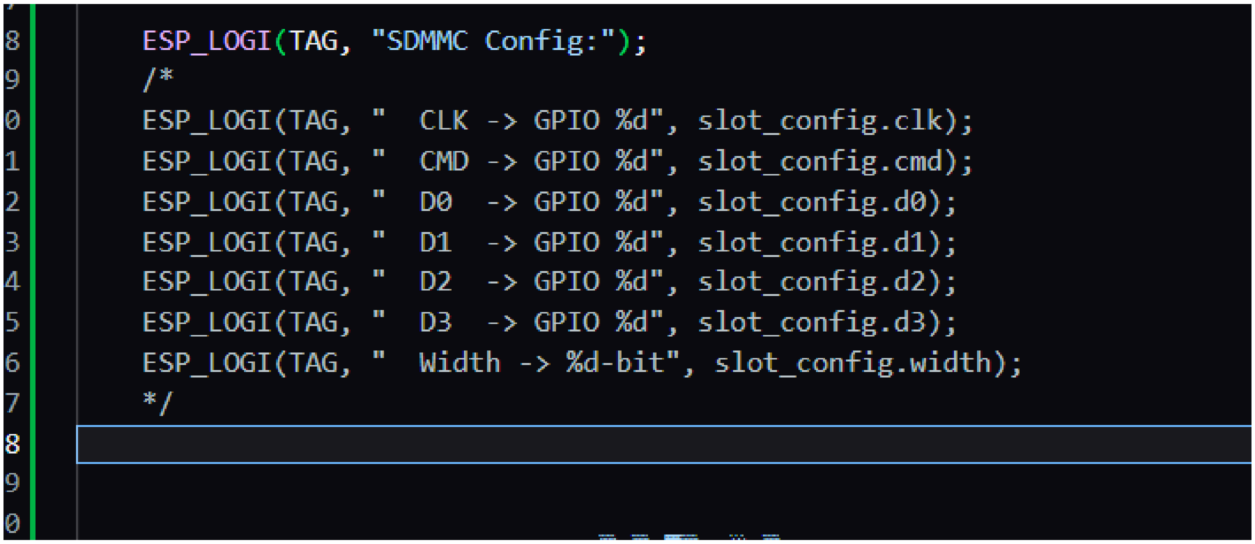
Taking data so pumped

For nature scanner, make sure you reconfigure for the camera before you rebuild



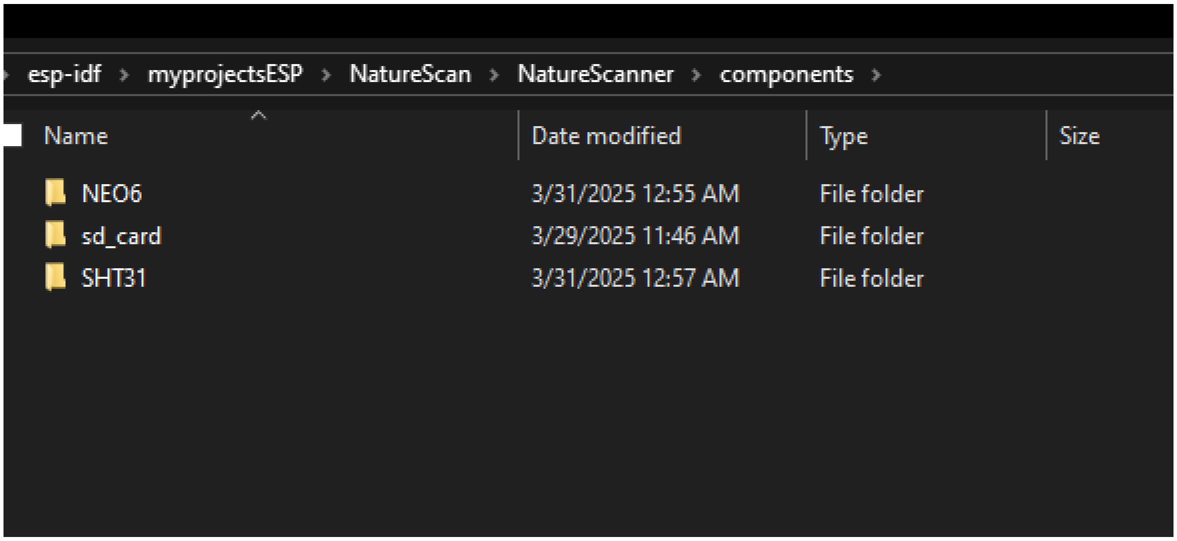
3. Reconfigure the project:

```bash
idf.py reconfigure
```



```
    ESP_LOGI(TAG, "SDMMC Config:");
    /*
    ESP_LOGI(TAG, "  CLK -> GPIO %d", slot_config.clk);
    ESP_LOGI(TAG, "  CMD -> GPIO %d", slot_config.cmd);
    ESP_LOGI(TAG, "  D0  -> GPIO %d", slot_config.d0);
    ESP_LOGI(TAG, "  D1  -> GPIO %d", slot_config.d1);
    ESP_LOGI(TAG, "  D2  -> GPIO %d", slot_config.d2);
    ESP_LOGI(TAG, "  D3  -> GPIO %d", slot_config.d3);
    ESP_LOGI(TAG, "  Width -> %d-bit", slot_config.width);
    */
```

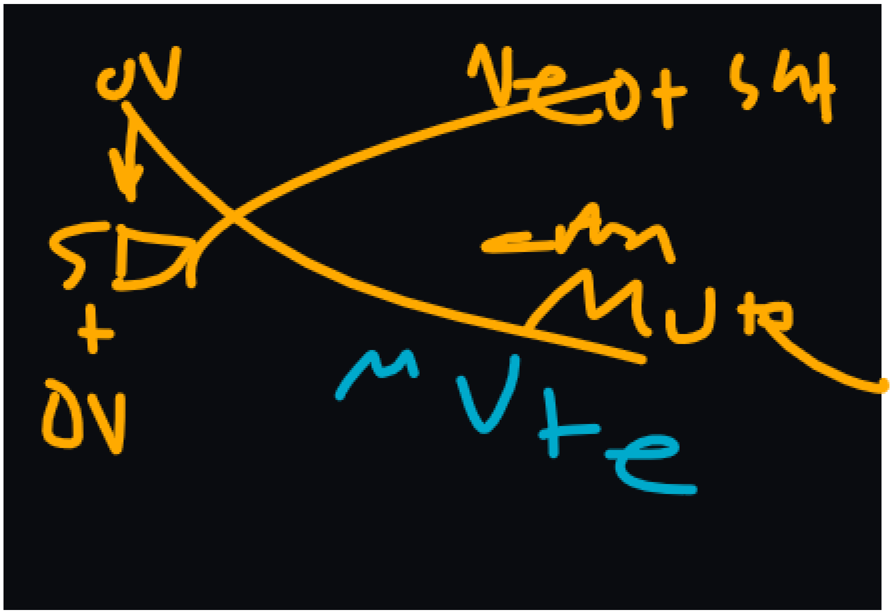
It was these

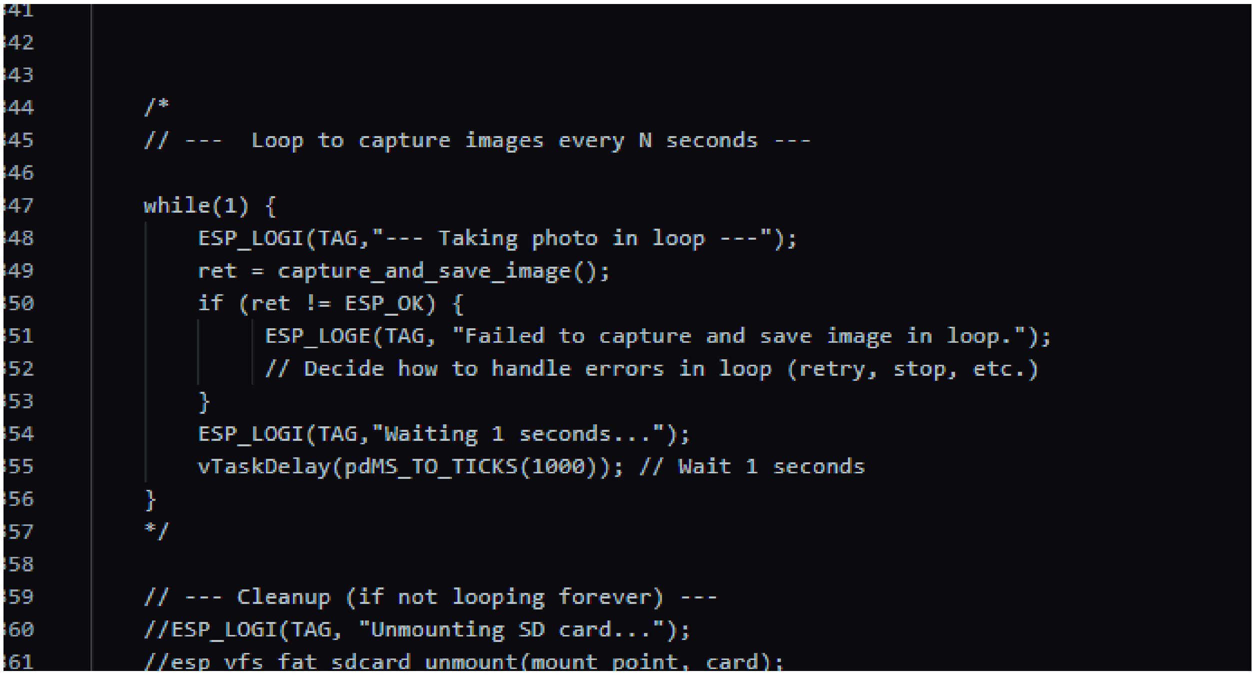


esp-idf › myprojectsESP › NatureScan › NatureScanner › components ›

Name	Date modified	Type	Size
NEO6	3/31/2025 12:55 AM	File folder	
sd_card	3/29/2025 11:46 AM	File folder	
SHT31	3/31/2025 12:57 AM	File folder	

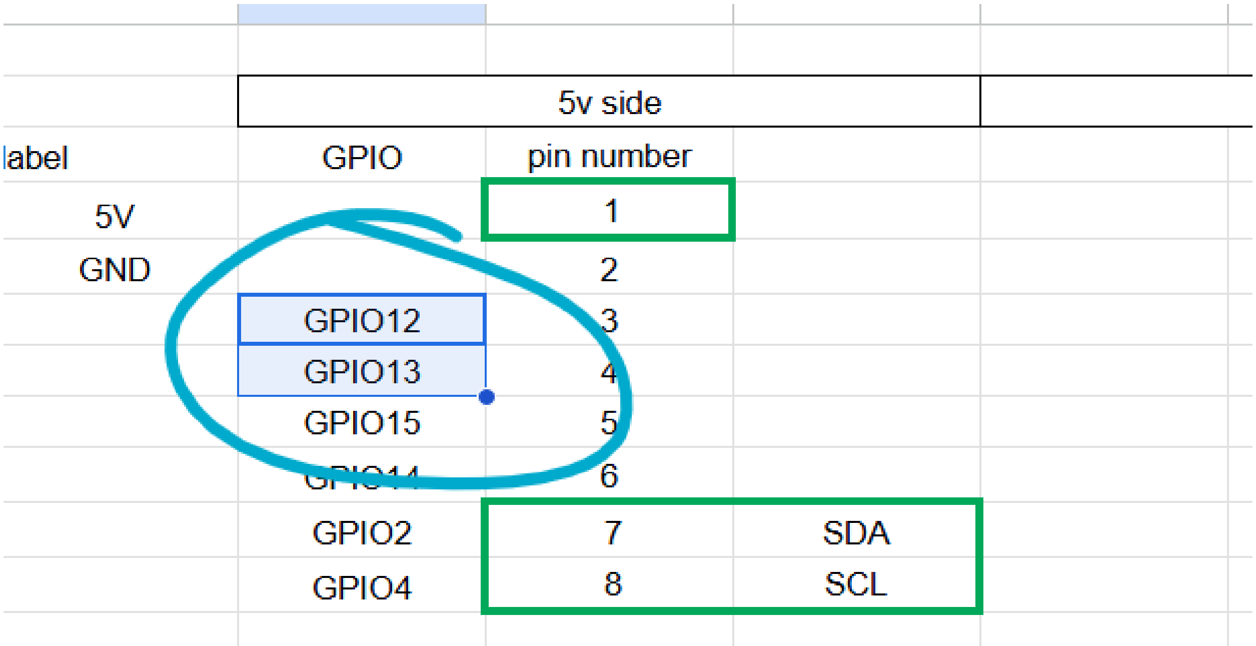
Boom



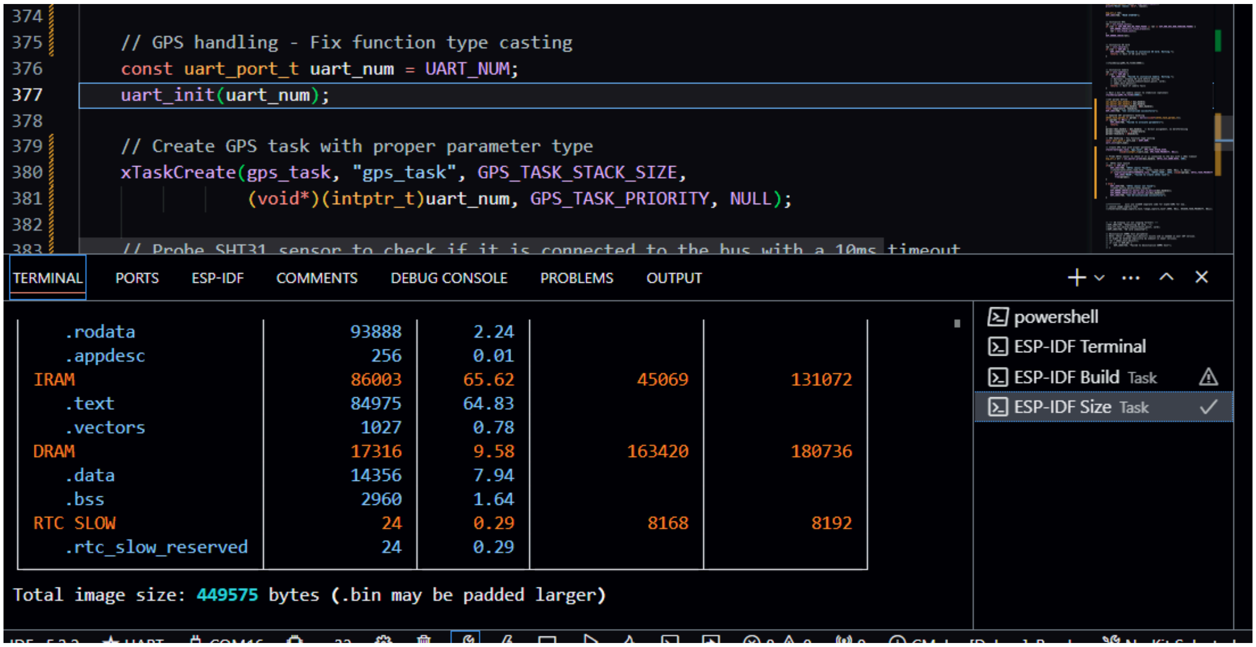


```
/*
// ---  Loop to capture images every N seconds ---

while(1) {
    ESP_LOGI(TAG,"--- Taking photo in loop ---");
    ret = capture_and_save_image();
    if (ret != ESP_OK) {
        ESP_LOGE(TAG, "Failed to capture and save image in loop.");
        // Decide how to handle errors in loop (retry, stop, etc.)
    }
    ESP_LOGI(TAG,"Waiting 1 seconds...");
    vTaskDelay(pdMS_TO_TICKS(1000)); // Wait 1 seconds
}
*/

// --- Cleanup (if not looping forever) ---
//ESP_LOGI(TAG, "Unmounting SD card...");
//esp_vfs_fat_sdcard_unmount(mount_point, card);
```

Im muting cam and sd card
Will flash with ov2640 for that build



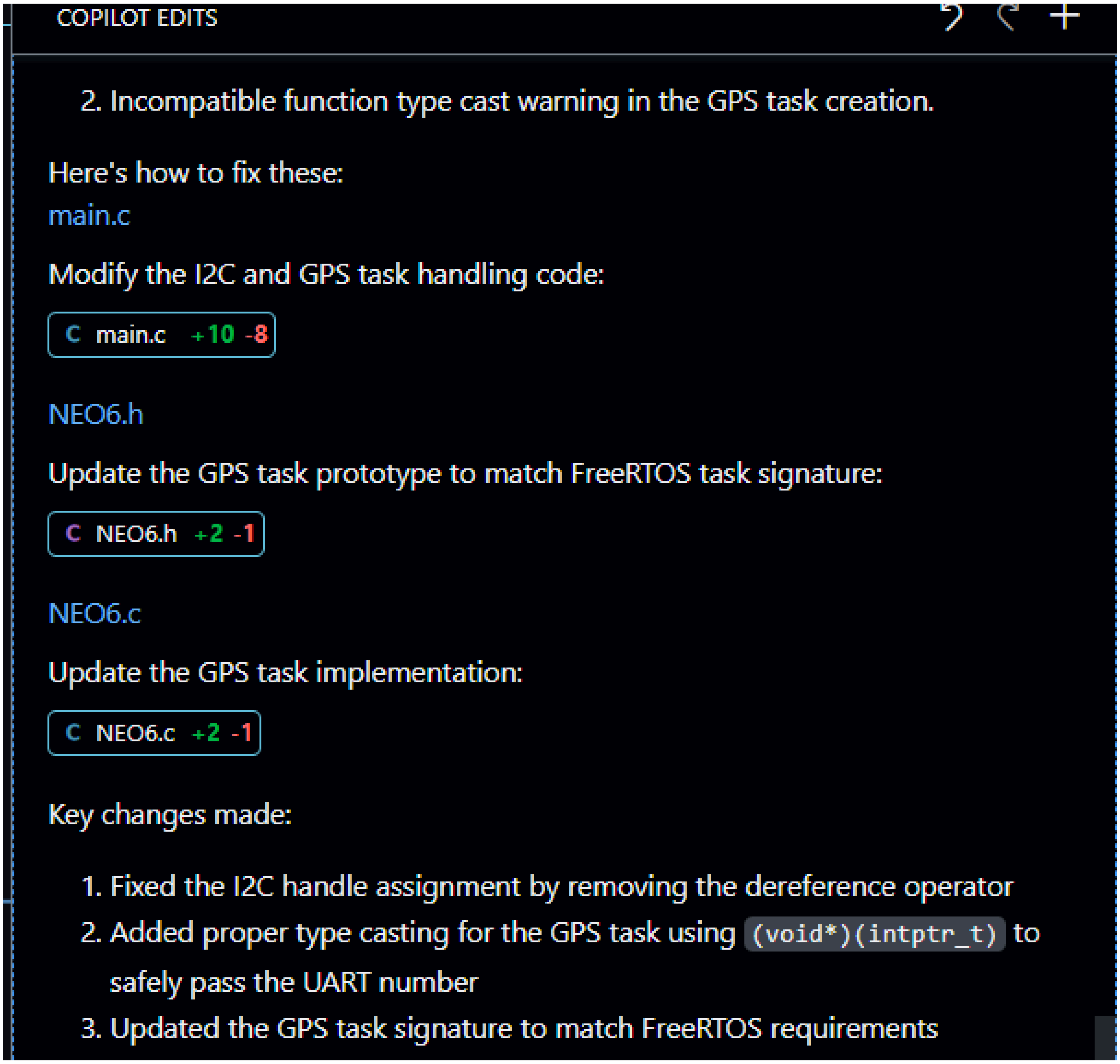
label	GPIO	pin number		
	5v side			
label	GPIO	pin number		
5V		1		
GND		2		
	GPIO12	3		
	GPIO13	4		
	GPIO15	5		
	GPIO14	6		
	GPIO2	7	SDA	
	GPIO4	8	SCL	



```
    // GPS handling - Fix function type casting
    const uart_port_t uart_num = UART_NUM;
    uart_init(uart_num);

    // Create GPS task with proper parameter type
    xTaskCreate(gps_task, "gps_task", GPS_TASK_STACK_SIZE,
                (void*)(intptr_t)uart_num, GPS_TASK_PRIORITY, NULL);

    // Probe SHT31 sensor to check if it is connected to the bus with a 10ms timeout
```

```
│    .rodata           │   93888 │   2.24 │              │              │
│    .appdesc          │     256 │   0.01 │              │              │
│ IRAM                 │   86003 │  65.62 │        45069 │       131072 │
│    .text             │   84975 │  64.83 │              │              │
│    .vectors          │    1027 │   0.78 │              │              │
│ DRAM                 │   17316 │   9.58 │       163420 │       180736 │
│    .data             │   14356 │   7.94 │              │              │
│    .bss              │    2960 │   1.64 │              │              │
│ RTC SLOW             │      24 │   0.29 │         8168 │         8192 │
│    .rtc_slow_reserved│      24 │   0.29 │              │              │
Total image size: 449575 bytes (.bin may be padded larger)
```

BUILT
Hail Mary



2. Incompatible function type cast warning in the GPS task creation.

Here's how to fix these:

main.c

Modify the I2C and GPS task handling code:

NEO6.h

Update the GPS task prototype to match FreeRTOS task signature:

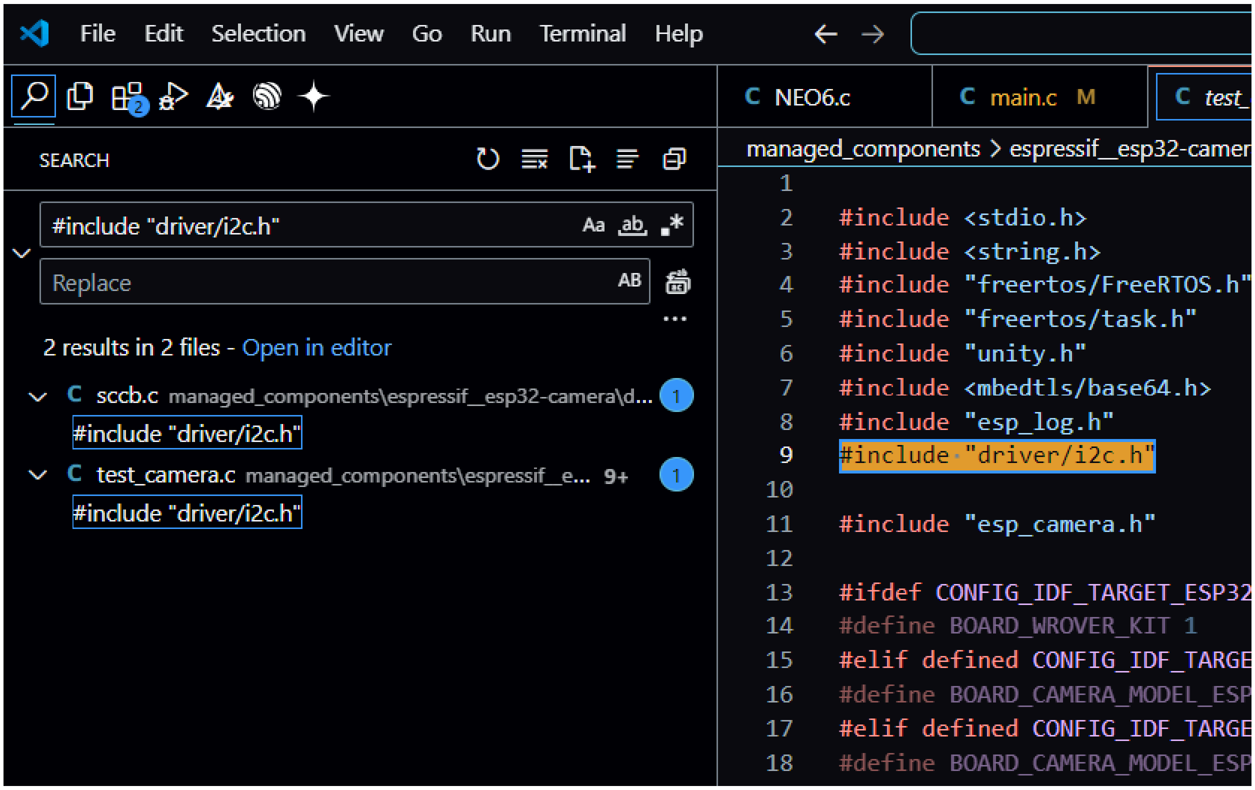
NEO6.c

Update the GPS task implementation:

Key changes made:

1. Fixed the I2C handle assignment by removing the dereference operator
2. Added proper type casting for the GPS task using `(void*)(intptr_t)` to safely pass the UART number
3. Updated the GPS task signature to match FreeRTOS requirements

I only denied one thing



managed_components > espressif__esp32-camer

Search: #include "driver/i2c.h"

2 results in 2 files - Open in editor

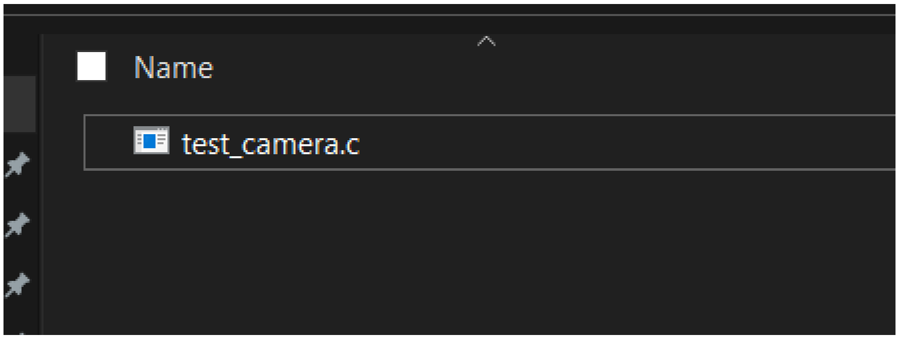
sccb.c managed_components\espressif__esp32-camera\d... 1
#include "driver/i2c.h"

test_camera.c managed_components\espressif__e... 9+ 1
#include "driver/i2c.h"

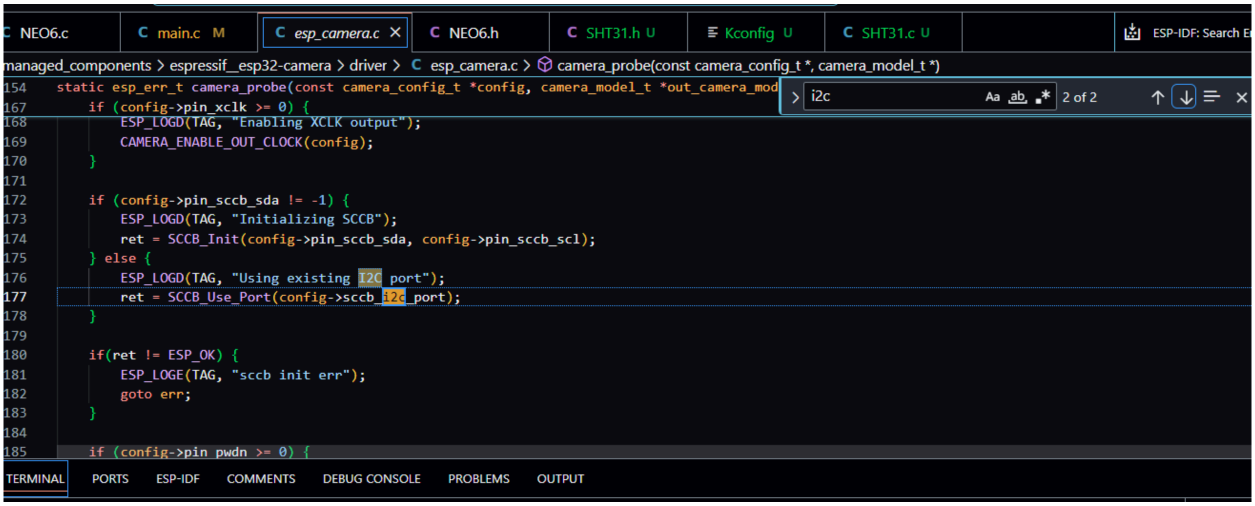
```

#include <stdio.h>
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "unity.h"
#include <mbedtls/base64.h>
#include "esp_log.h"
#include "driver/i2c.h"

#include "esp_camera.h"

#ifdef CONFIG_IDF_TARGET_ESP32
#define BOARD_WROVER_KIT 1
#elif defined CONFIG_IDF_TARGE
#define BOARD_CAMERA_MODEL_ESP
#elif defined CONFIG_IDF_TARGE
#define BOARD_CAMERA_MODEL_ESP
```

"C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScanner\managed_components\espressif__esp32-camera\test\test_camera.c"

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScanner\managed_components\espressif__esp32-camera\driver



No test camera = pass

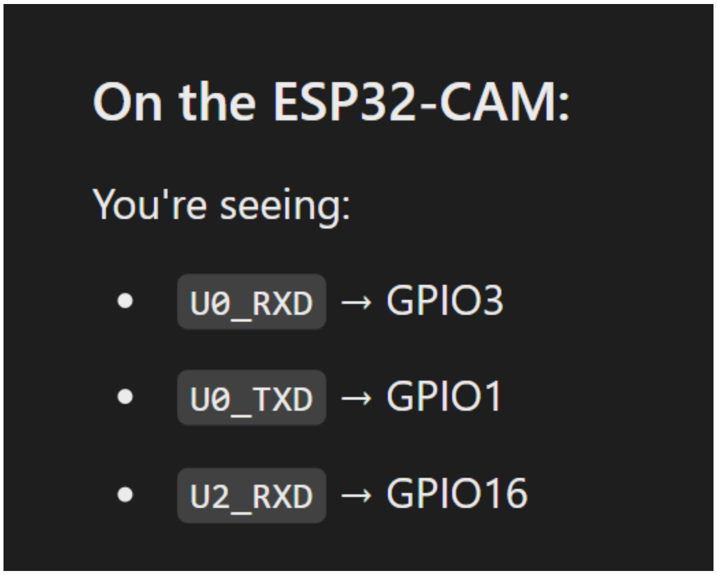


```c
static esp_err_t camera_probe(const camera_config_t *config, camera_model_t *out_camera_mod
    if (config->pin_xclk >= 0) {
        ESP_LOGD(TAG, "Enabling XCLK output");
        CAMERA_ENABLE_OUT_CLOCK(config);
    }

    if (config->pin_sccb_sda != -1) {
        ESP_LOGD(TAG, "Initializing SCCB");
        ret = SCCB_Init(config->pin_sccb_sda, config->pin_sccb_scl);
    } else {
        ESP_LOGD(TAG, "Using existing I2C port");
        ret = SCCB_Use_Port(config->sccb_i2c_port);
    }

    if(ret != ESP_OK) {
        ESP_LOGE(TAG, "sccb init err");
        goto err;
    }

    if (config->pin_pwdn >= 0) {
```

Only one call

3/29/2025

[Geminiblog gen](#)
[Chatgpt blog gen](#)

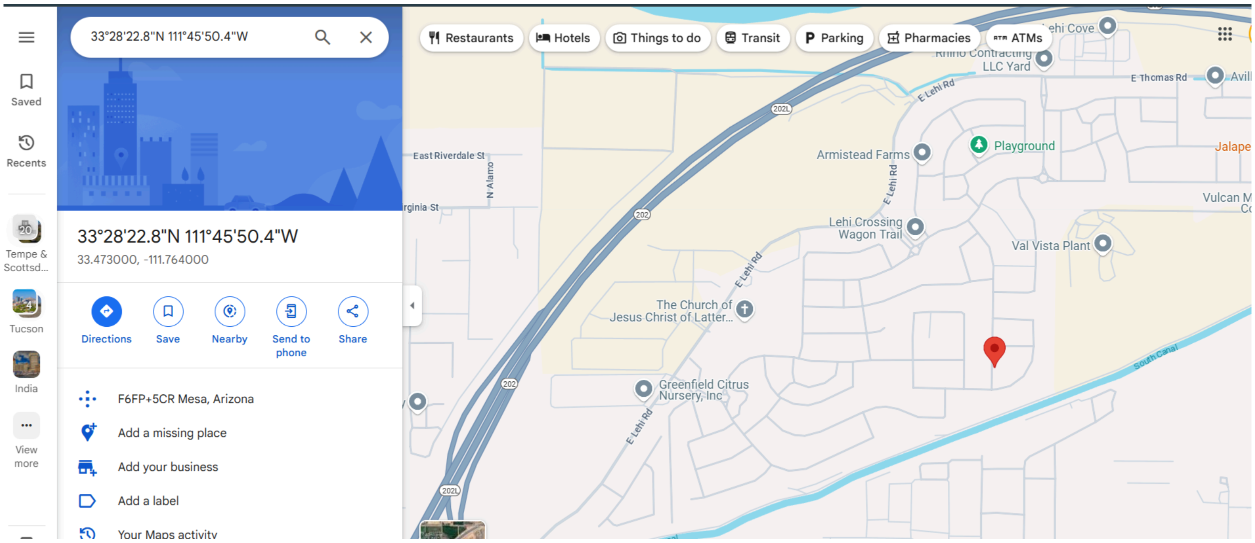


On the ESP32-CAM:

You're seeing:

- `U0_RXD` → GPIO3
- `U0_TXD` → GPIO1
- `U2_RXD` → GPIO16

Want:
GPIO12: TXD
GPIO13: RXD

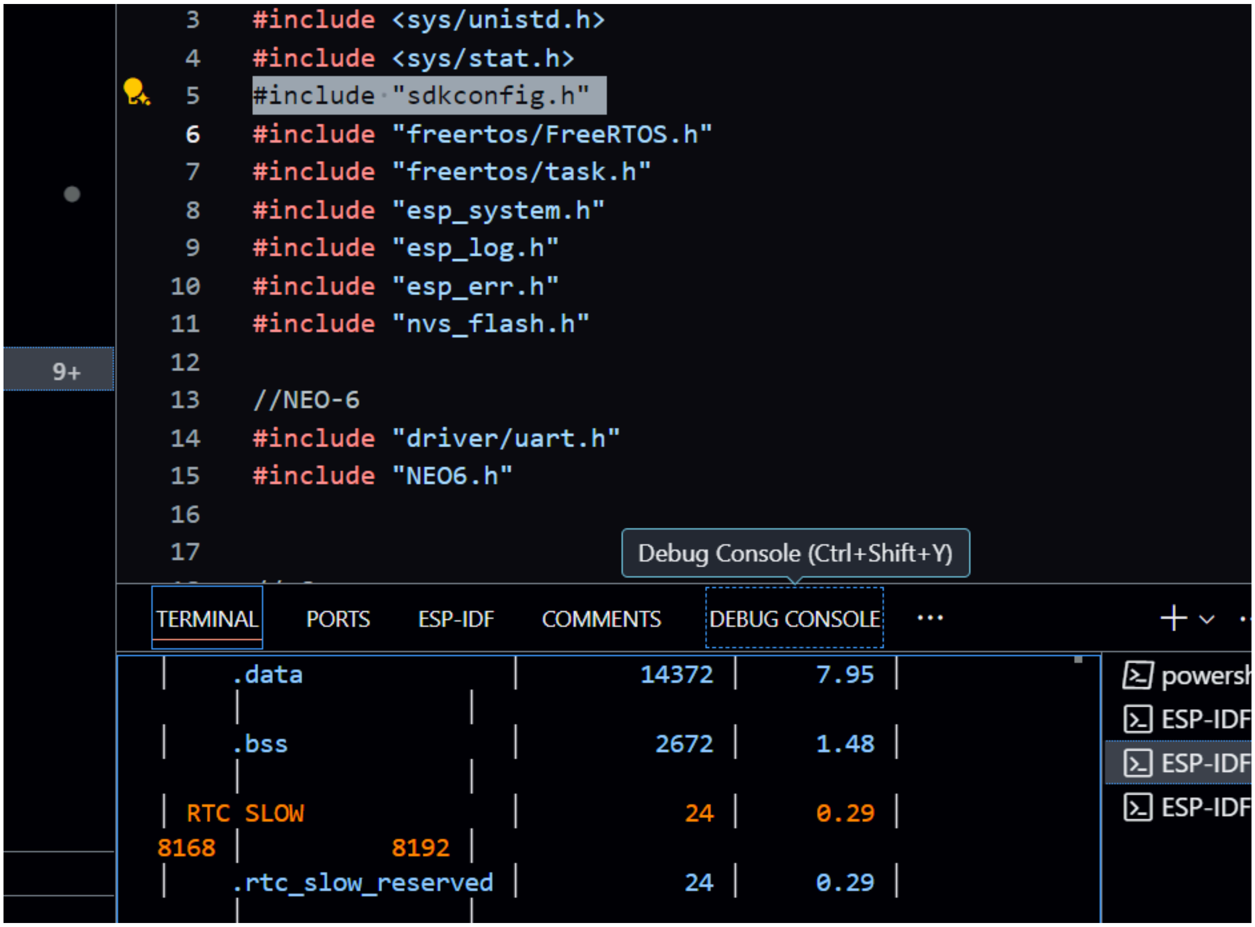


33°28'22.8"N 111°45'50.4"W

33°28'22.8"N 111°45'50.4"W
33.473000, -111.764000

F6FP+5CR Mesa, Arizona

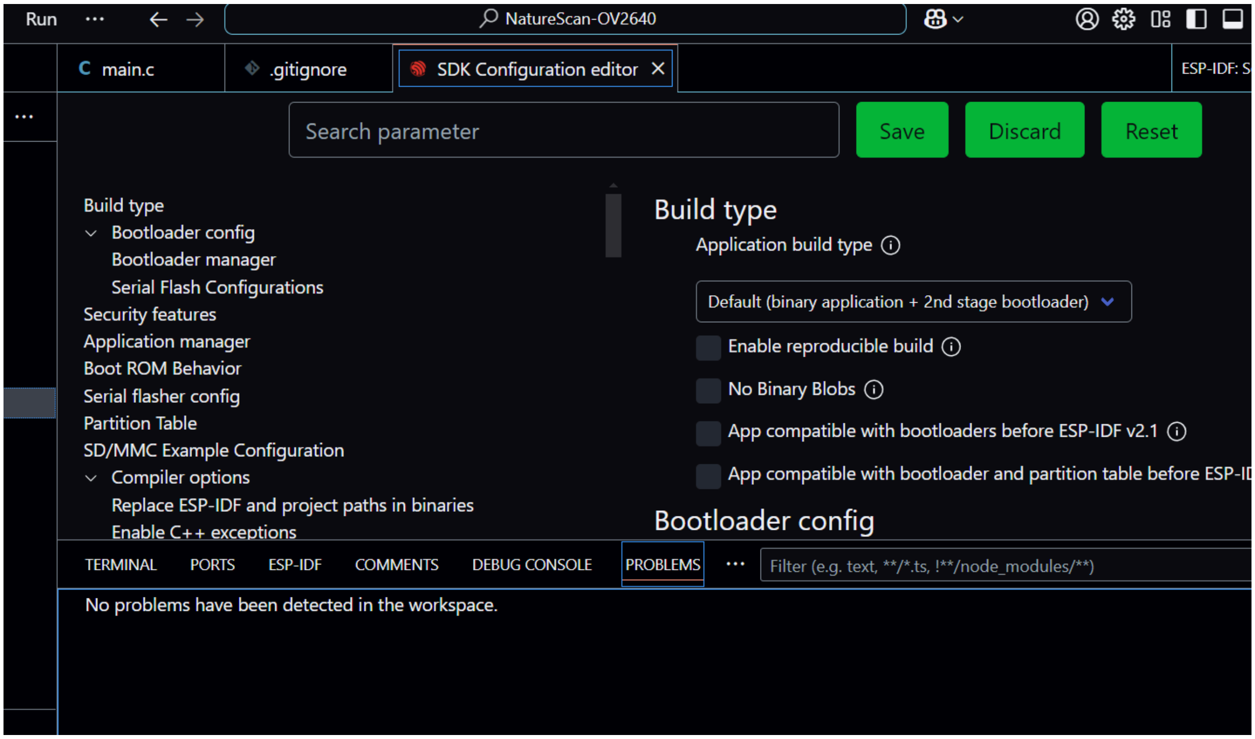
Works at dustins place



```
#include <sys/unistd.h>
#include <sys/stat.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
#include "esp_log.h"
#include "esp_err.h"
#include "nvs_flash.h"

//NEO-6
#include "driver/uart.h"
#include "NEO6.h"

```

```
|     .data                 |       14372 |      7.95 |
     |                 |
|     .bss                  |        2672 |      1.48 |
     |                 |
| RTC SLOW                  |          24 |      0.29 |
8168 |            8192 |
|     .rtc_slow_reserved    |          24 |      0.29 |
     |                 |
```

❌All of those missing pins errors were because i did have skdconfig.h in the included



Build type

Application build type: Default (binary application + 2nd stage bootloader)

☐ Enable reproducible build

☐ No Binary Blobs

☐ App compatible with bootloaders before ESP-IDF v2.1

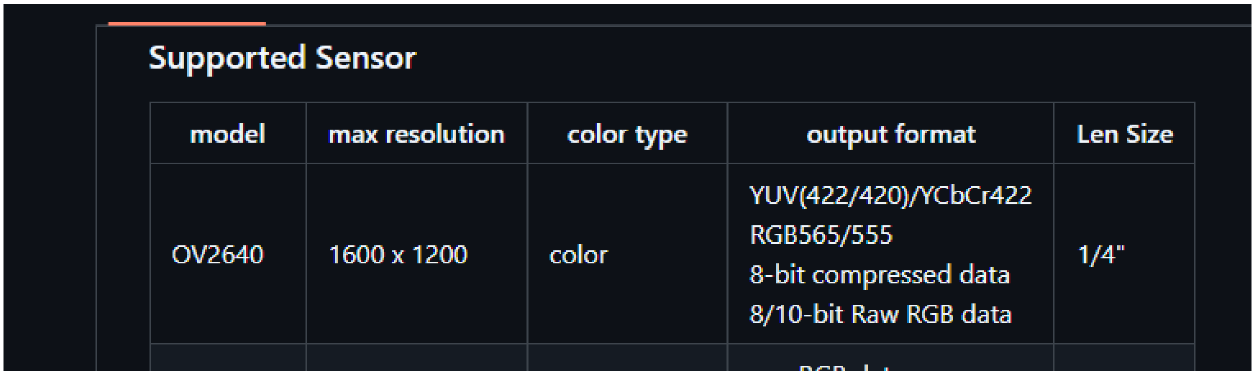
☐ App compatible with bootloader and partition table before ESP-ID

Bootloader config

No problems have been detected in the workspace.

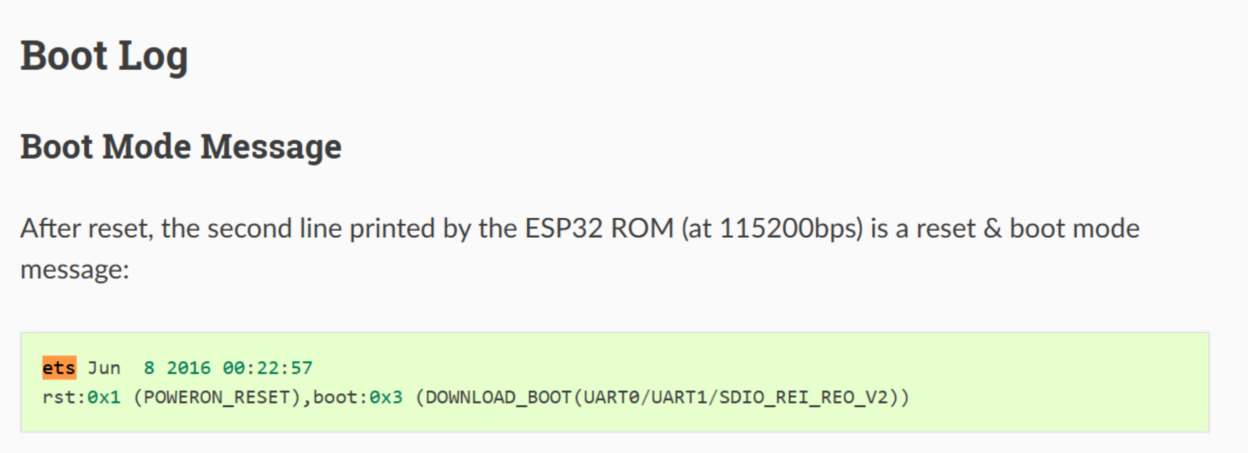
Its because the board was not plugged into the computer and we couldnt run menuconfig

3/28/2025



Supported Sensor

model	max resolution	color type	output format	Len Size
OV2640	1600 x 1200	color	YUV(422/420)/YCbCr422 RGB565/555 8-bit compressed data 8/10-bit Raw RGB data	1/4"



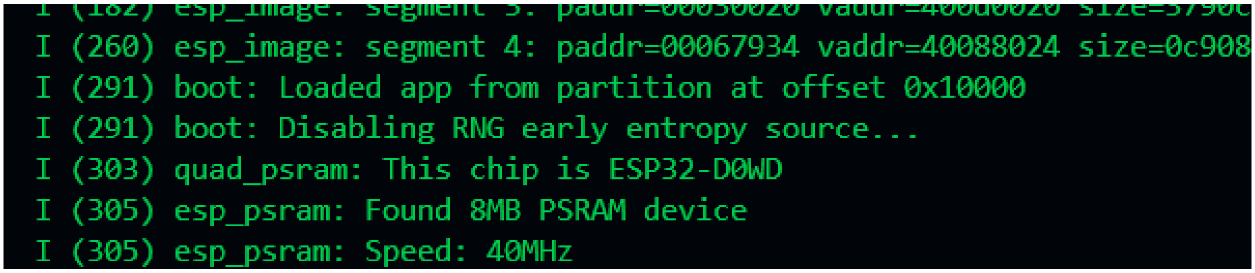
Boot Log

Boot Mode Message

After reset, the second line printed by the ESP32 ROM (at 115200bps) is a reset & boot mode message:

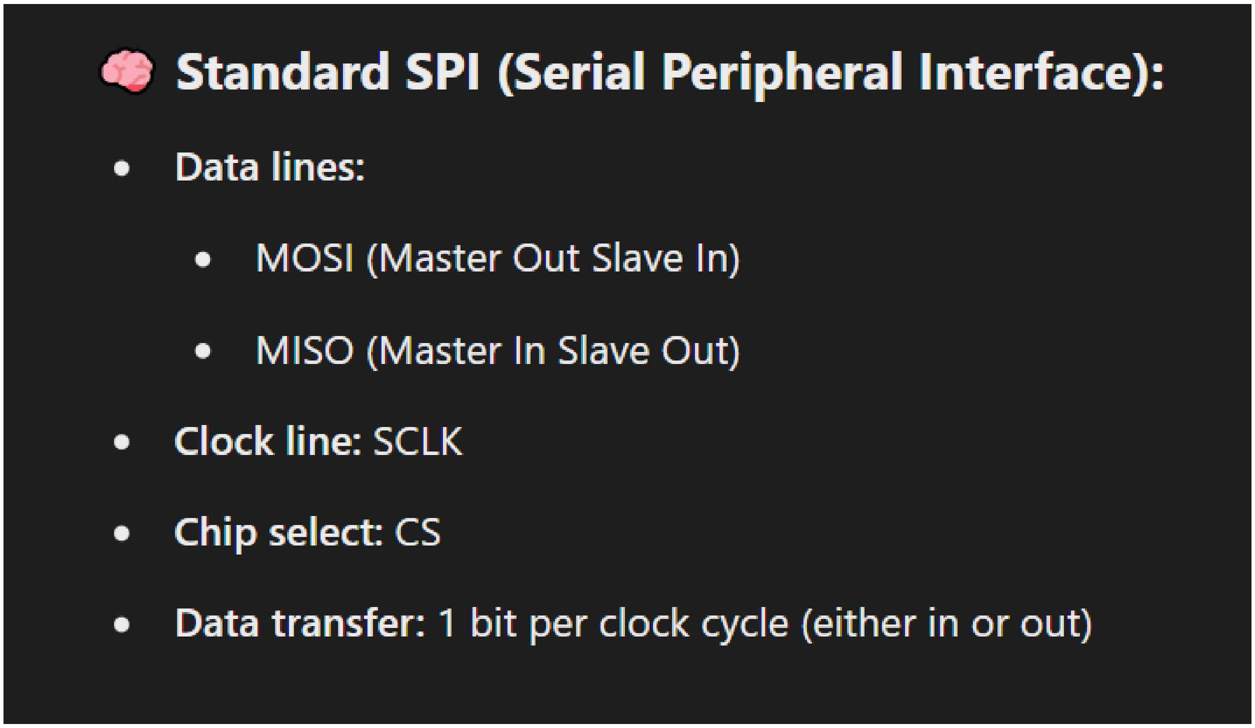
```
ets Jun  8 2016 00:22:57
rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
```

Boot log



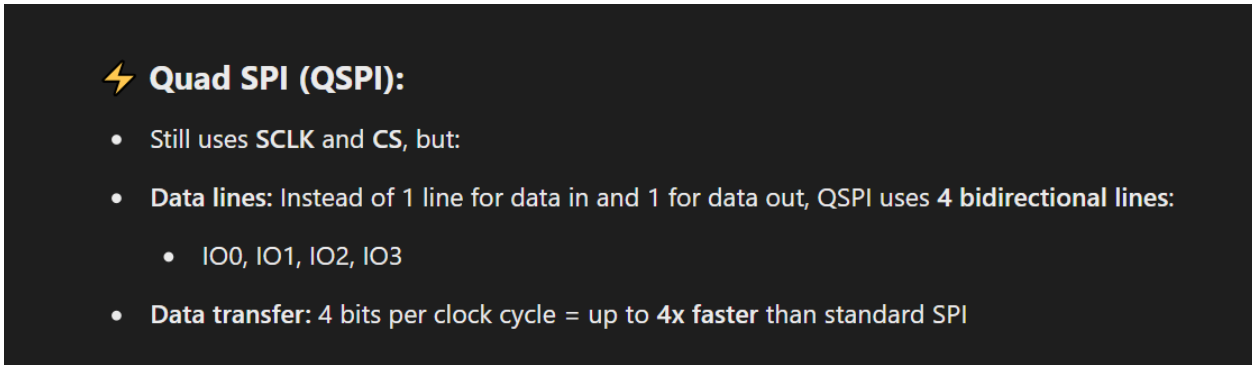
```
I (182) esp_image: segment 3: paddr=00030020 vaddr=400d0020 size=3790c
I (260) esp_image: segment 4: paddr=00067934 vaddr=40088024 size=0c908
I (291) boot: Loaded app from partition at offset 0x10000
I (291) boot: Disabling RNG early entropy source...
I (303) quad_psram: This chip is ESP32-D0WD
I (305) esp_psram: Found 8MB PSRAM device
I (305) esp_psram: Speed: 40MHz
```

This chip is ESP32-D0WD



🧠 Standard SPI (Serial Peripheral Interface):

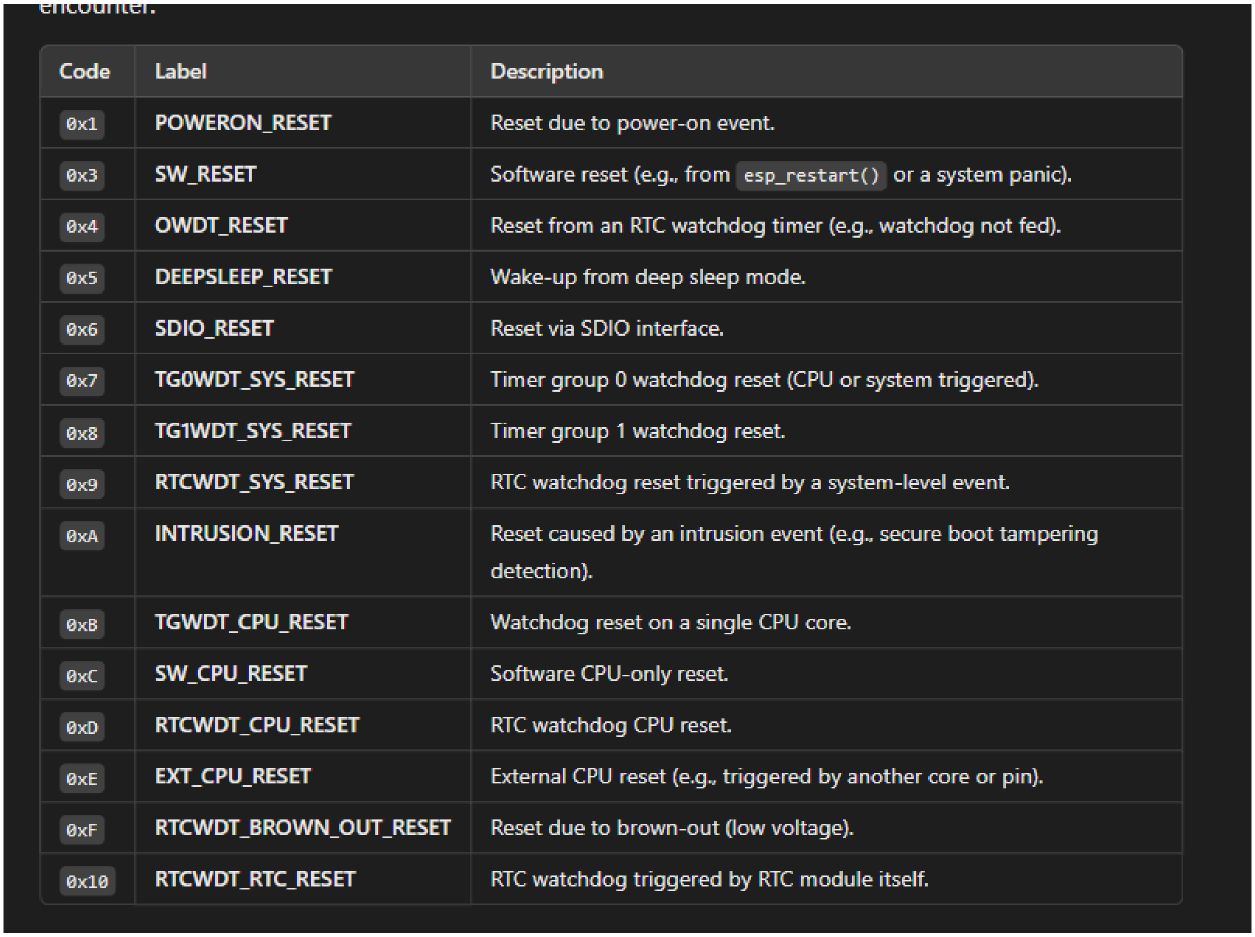
- **Data lines:**
  - MOSI (Master Out Slave In)
  - MISO (Master In Slave Out)
- **Clock line:** SCLK
- **Chip select:** CS
- **Data transfer:** 1 bit per clock cycle (either in or out)



⚡ Quad SPI (QSPI):

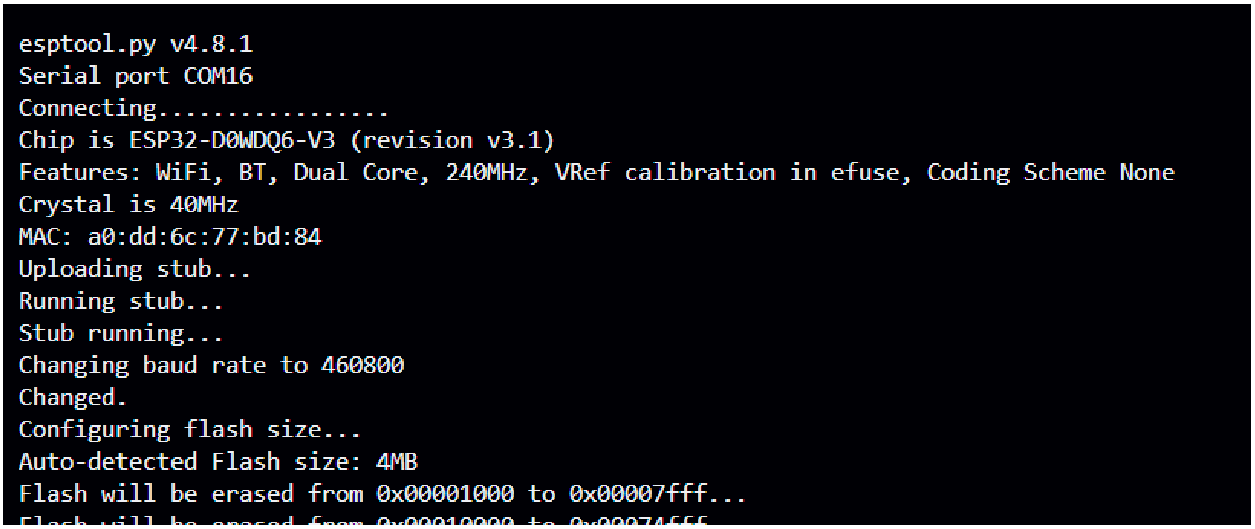
- Still uses **SCLK** and **CS**, but:
- **Data lines:** Instead of 1 line for data in and 1 for data out, QSPI uses **4 bidirectional lines**:
  - IO0, IO1, IO2, IO3
- **Data transfer:** 4 bits per clock cycle = up to **4x faster** than standard SPI

Esp32 reset types



encounter.

Code	Label	Description
`0x1`	POWERON_RESET	Reset due to power-on event.
`0x3`	SW_RESET	Software reset (e.g., from `esp_restart()` or a system panic).
`0x4`	OWDT_RESET	Reset from an RTC watchdog timer (e.g., watchdog not fed).
`0x5`	DEEPSLEEP_RESET	Wake-up from deep sleep mode.
`0x6`	SDIO_RESET	Reset via SDIO interface.
`0x7`	TG0WDT_SYS_RESET	Timer group 0 watchdog reset (CPU or system triggered).
`0x8`	TG1WDT_SYS_RESET	Timer group 1 watchdog reset.
`0x9`	RTCWDT_SYS_RESET	RTC watchdog reset triggered by a system-level event.
`0xA`	INTRUSION_RESET	Reset caused by an intrusion event (e.g., secure boot tampering detection).
`0xB`	TGWDT_CPU_RESET	Watchdog reset on a single CPU core.
`0xC`	SW_CPU_RESET	Software CPU-only reset.
`0xD`	RTCWDT_CPU_RESET	RTC watchdog CPU reset.
`0xE`	EXT_CPU_RESET	External CPU reset (e.g., triggered by another core or pin).
`0xF`	RTCWDT_BROWN_OUT_RESET	Reset due to brown-out (low voltage).
`0x10`	RTCWDT_RTC_RESET	RTC watchdog triggered by RTC module itself.



```
esptool.py v4.8.1
Serial port COM16
Connecting.................
Chip is ESP32-D0WDQ6-V3 (revision v3.1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: a0:dd:6c:77:bd:84
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash will be erased from 0x00001000 to 0x00007fff...
```

Changing baud rate when flashing…

3/27/2025

`fprintf`, `fopen`, and `fclose` are standard C library functions, specifically belonging to the **stdio.h (Standard Input/Output) library**

ANSI stands for **American National Standards Institute**

C89/C90 Standard Library Headers: The Core of C

The image lists the 15 standard library headers introduced in C89/C90 (ANSI C), which was the first standardized version of C:

1. <assert.h> - Diagnostics (assertions)
2. <ctype.h> - Character handling
3. <errno.h> - Error codes
4. <float.h> - Floating-point characteristics
5. <limits.h> - Integer type limits
6. <locale.h> - Localization
7. <math.h> - Mathematical functions
8. <setjmp.h> - Non-local jumps
9. <signal.h> - Signal handling
10. <stdarg.h> - Variable arguments
11. <stddef.h> - Standard definitions (e.g., NULL, size_t)
12. <stdio.h> - Standard input/output
13. <stdlib.h> - General utilities (memory, random numbers, etc.)
14. <string.h> - String handling
15. <time.h> - Date and time functions

Total in C89: 15 headers

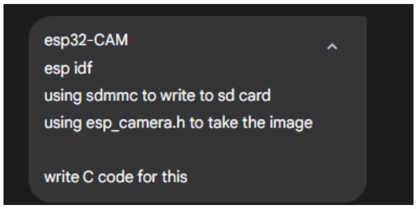
Check which version of C you have:

```
gcc -std=c11 -v -o test test.c
```

**sdmmc example

static esp_err_t **s_example_write_file**(const char *path, char *data)
static esp_err_t **s_example_read_file**(const char *path)

Prompt:



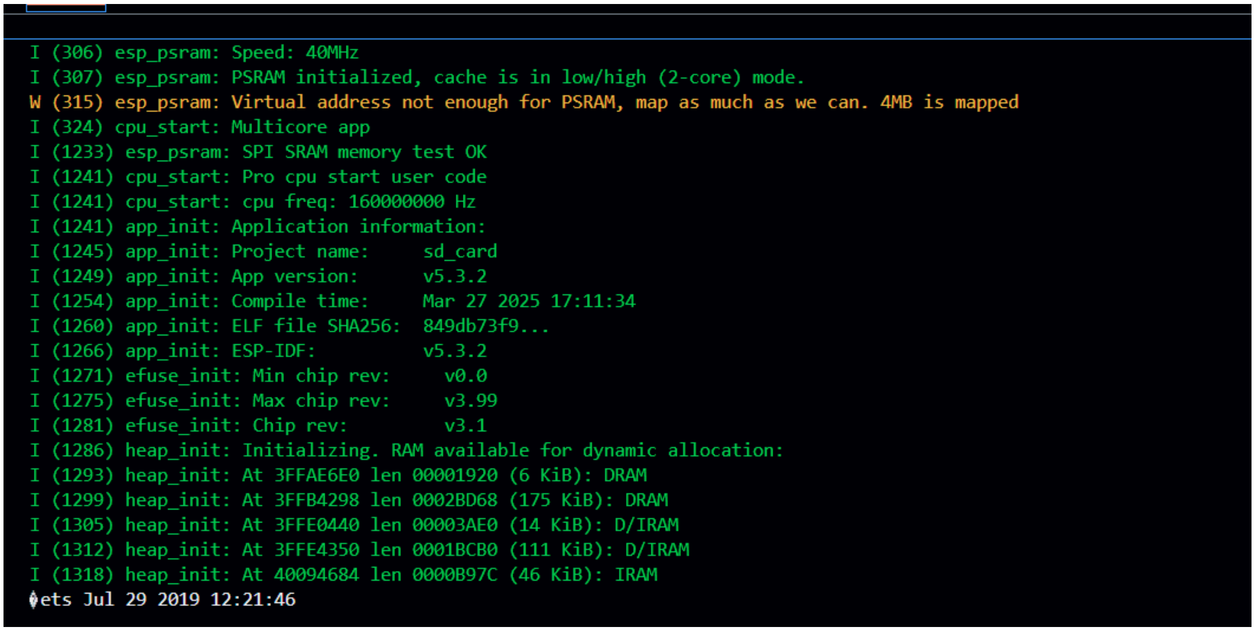
esp32-CAM
esp idf
using sdmmc to write to sd card
using esp_camera.h to take the image

write C code for this

Gemini code link

static esp_err_t init_camera()
static esp_err_t init_sdcard()
static esp_err_t capture_and_save_image()

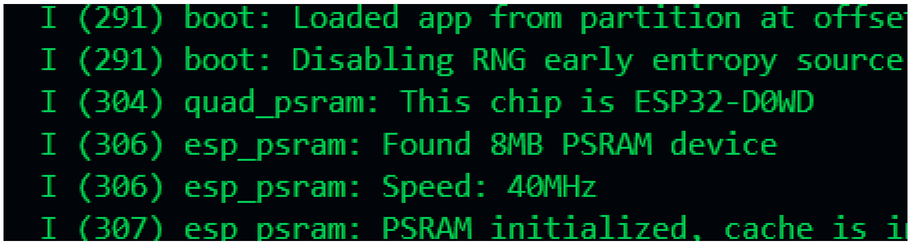
if everything is set up correctly, the ESP32-CAM will initialize the camera and SD card, take a picture, and save it as `imageXXX.jpg` to the root directory of your SD card.



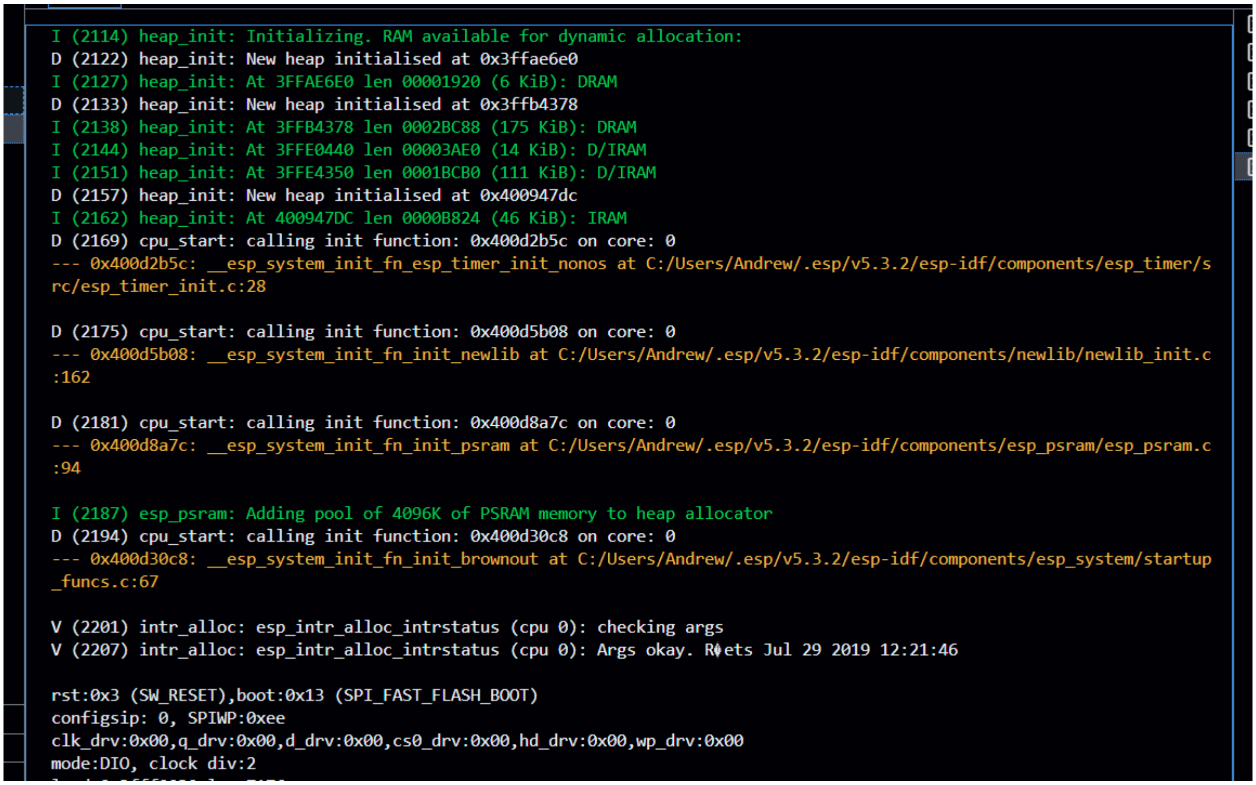
```
I (306) esp_psram: Speed: 40MHz
I (307) esp_psram: PSRAM initialized, cache is in low/high (2-core) mode.
W (315) esp_psram: Virtual address not enough for PSRAM, map as much as we can. 4MB is mapped
I (324) cpu_start: Multicore app
I (1233) esp_psram: SPI SRAM memory test OK
I (1241) cpu_start: Pro cpu start user code
I (1241) cpu_start: cpu freq: 160000000 Hz
I (1241) app_init: Application information:
I (1245) app_init: Project name:     sd_card
I (1249) app_init: App version:      v5.3.2
I (1254) app_init: Compile time:     Mar 27 2025 17:11:34
I (1260) app_init: ELF file SHA256:  849db73f9...
I (1266) app_init: ESP-IDF:          v5.3.2
I (1271) efuse_init: Min chip rev:     v0.0
I (1275) efuse_init: Max chip rev:     v3.99
I (1281) efuse_init: Chip rev:         v3.1
I (1286) heap_init: Initializing. RAM available for dynamic allocation:
I (1293) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (1299) heap_init: At 3FFB4298 len 0002BD68 (175 KiB): DRAM
I (1305) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (1312) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (1318) heap_init: At 40094684 len 0000B97C (46 KiB): IRAM
�ets Jul 29 2019 12:21:46
```

4MB mapped of PSRAM

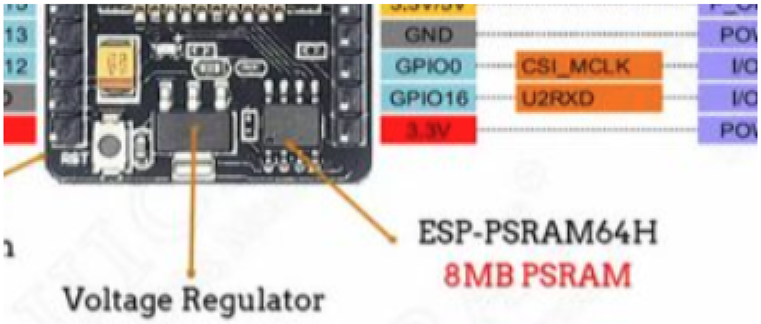
**Found 8MB PSRAM device



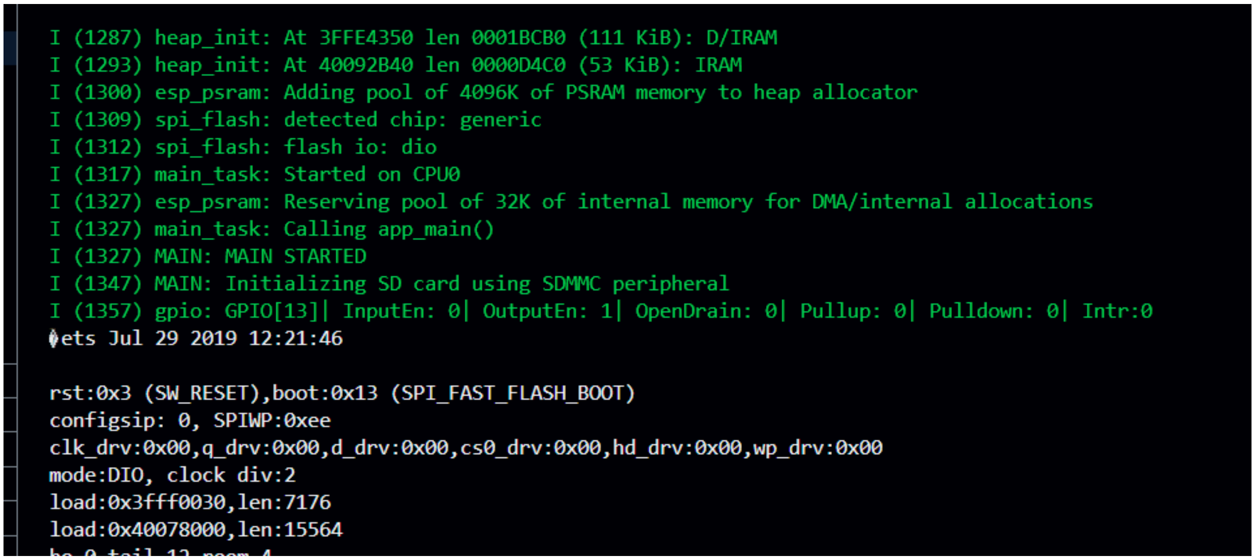
```
I (291) boot: Loaded app from partition at offse
I (291) boot: Disabling RNG early entropy source
I (304) quad_psram: This chip is ESP32-D0WD
I (306) esp_psram: Found 8MB PSRAM device
I (306) esp_psram: Speed: 40MHz
I (307) esp_psram: PSRAM initialized, cache is i
```



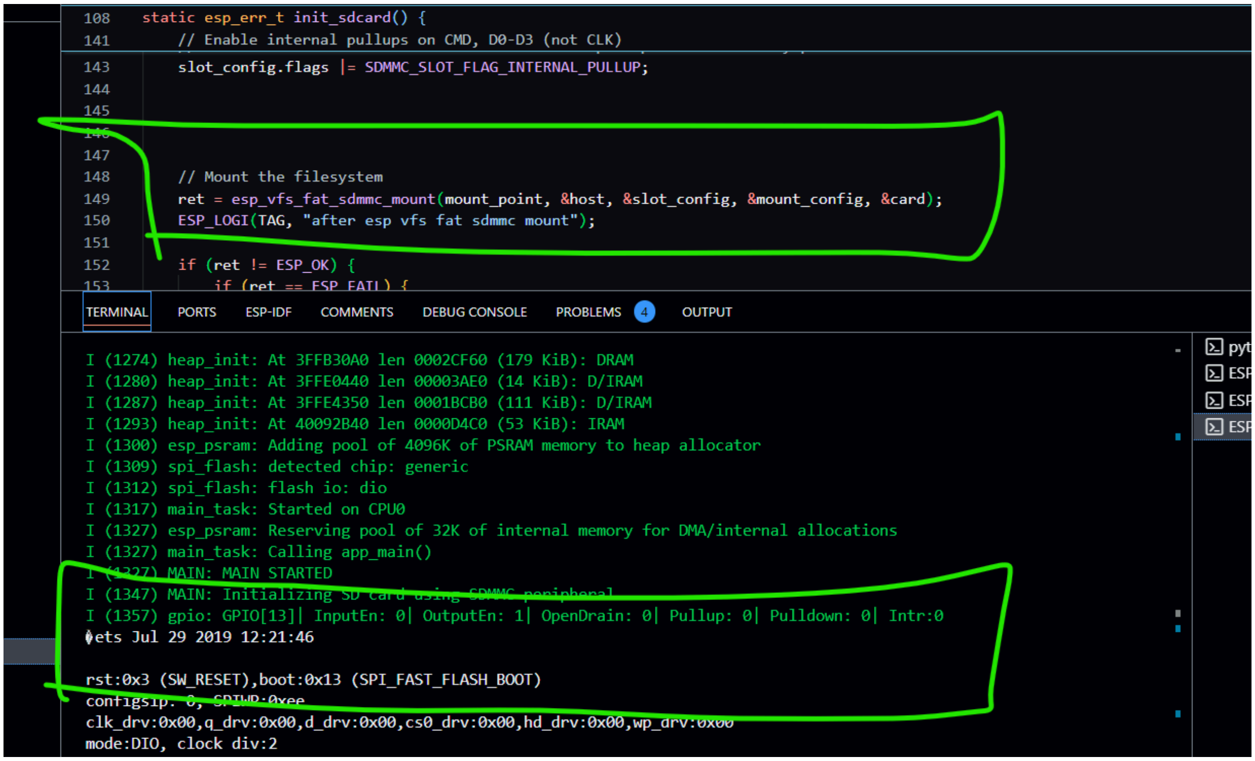
```
I (2114) heap_init: Initializing. RAM available for dynamic allocation:
D (2122) heap_init: New heap initialised at 0x3ffae6e0
I (2127) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
D (2133) heap_init: New heap initialised at 0x3ffb4378
I (2138) heap_init: At 3FFB4378 len 0002BC88 (175 KiB): DRAM
I (2144) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (2151) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
D (2157) heap_init: New heap initialised at 0x400947dc
I (2162) heap_init: At 400947DC len 0000B824 (46 KiB): IRAM
D (2169) cpu_start: calling init function: 0x400d2b5c on core: 0
--- 0x400d2b5c: __esp_system_init_fn_esp_timer_init_nonos at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_timer/s
rc/esp_timer_init.c:28

D (2175) cpu_start: calling init function: 0x400d5b08 on core: 0
--- 0x400d5b08: __esp_system_init_fn_init_newlib at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/newlib/newlib_init.c
:162

D (2181) cpu_start: calling init function: 0x400d8a7c on core: 0
--- 0x400d8a7c: __esp_system_init_fn_init_psram at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_psram/esp_psram.c
:94

I (2187) esp_psram: Adding pool of 4096K of PSRAM memory to heap allocator
D (2194) cpu_start: calling init function: 0x400d30c8 on core: 0
--- 0x400d30c8: __esp_system_init_fn_init_brownout at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/startup
_funcs.c:67

V (2201) intr_alloc: esp_intr_alloc_intrstatus (cpu 0): checking args
V (2207) intr_alloc: esp_intr_alloc_intrstatus (cpu 0): Args okay. R�ets Jul 29 2019 12:21:46

rst:0x3 (SW_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
```



64H
PSRAM 8MB??



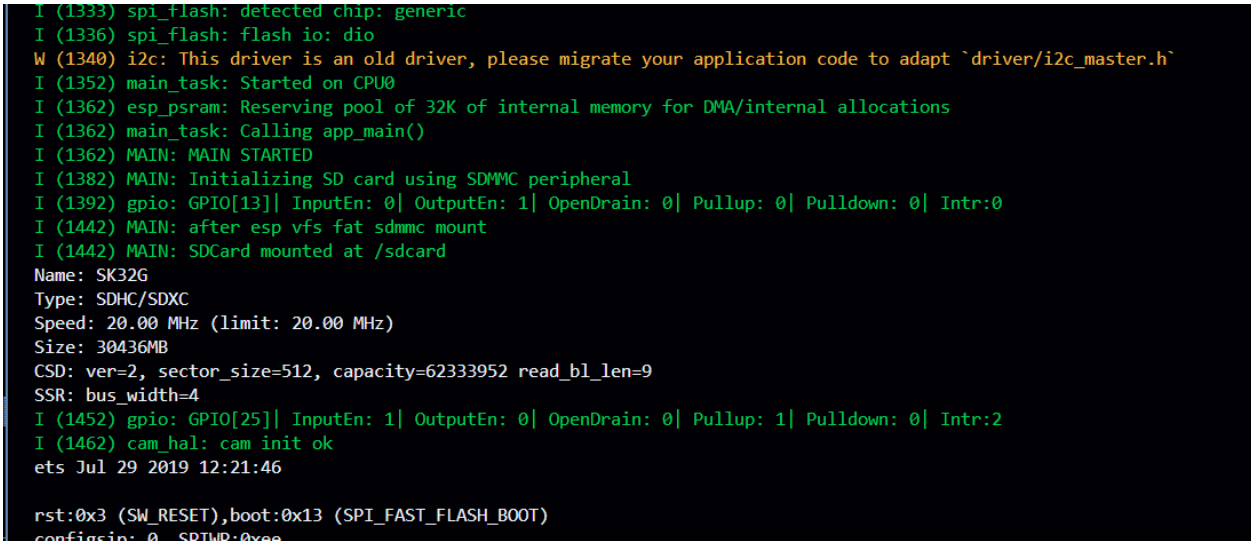
```
I (1287) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (1293) heap_init: At 40092B40 len 0000D4C0 (53 KiB): IRAM
I (1300) esp_psram: Adding pool of 4096K of PSRAM memory to heap allocator
I (1309) spi_flash: detected chip: generic
I (1312) spi_flash: flash io: dio
I (1317) main_task: Started on CPU0
I (1327) esp_psram: Reserving pool of 32K of internal memory for DMA/internal allocations
I (1327) main_task: Calling app_main()
I (1327) MAIN: MAIN STARTED
I (1347) MAIN: Initializing SD card using SDMMC peripheral
I (1357) gpio: GPIO[13]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0
�ets Jul 29 2019 12:21:46

rst:0x3 (SW_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:7176
load:0x40078000,len:15564
```



```
static esp_err_t init_sdcard() {
    // Enable internal pullups on CMD, D0-D3 (not CLK)
    slot_config.flags |= SDMMC_SLOT_FLAG_INTERNAL_PULLUP;




    // Mount the filesystem
    ret = esp_vfs_fat_sdmmc_mount(mount_point, &host, &slot_config, &mount_config, &card);
    ESP_LOGI(TAG, "after esp vfs fat sdmmc mount");

    if (ret != ESP_OK) {
        if (ret == ESP_FAIL) {
```

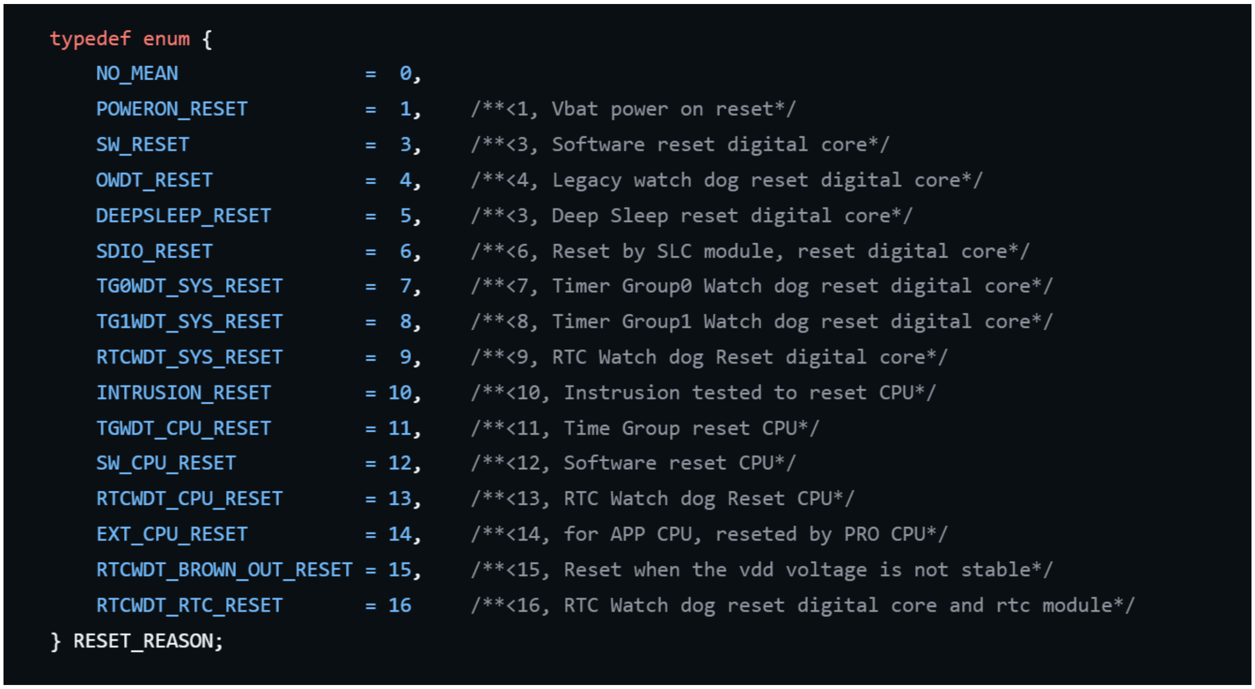
```
I (1274) heap_init: At 3FFB30A0 len 0002CF60 (179 KiB): DRAM
I (1280) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (1287) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (1293) heap_init: At 40092B40 len 0000D4C0 (53 KiB): IRAM
I (1300) esp_psram: Adding pool of 4096K of PSRAM memory to heap allocator
I (1309) spi_flash: detected chip: generic
I (1312) spi_flash: flash io: dio
I (1317) main_task: Started on CPU0
I (1327) esp_psram: Reserving pool of 32K of internal memory for DMA/internal allocations
I (1327) main_task: Calling app_main()
I (1327) MAIN: MAIN STARTED
I (1347) MAIN: Initializing SD card using SDMMC peripheral
I (1357) gpio: GPIO[13]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0
�ets Jul 29 2019 12:21:46

rst:0x3 (SW_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
```

Was getting this error. I removed the sd card then re inserted. This worked!!

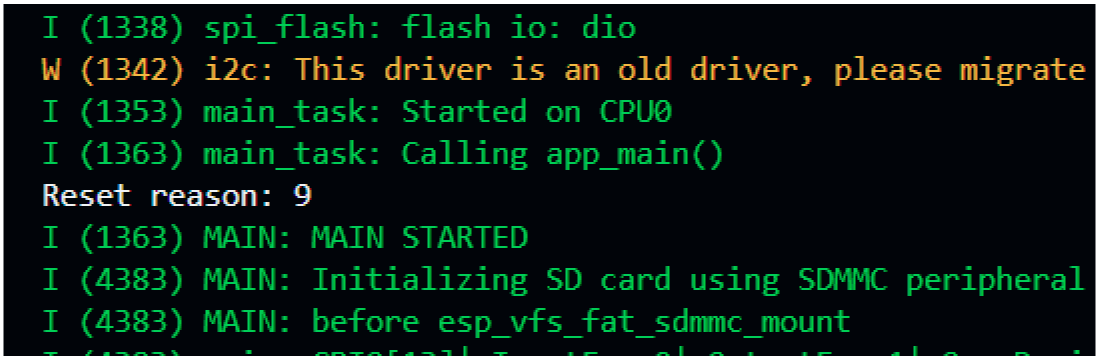


```
I (1333) spi_flash: detected chip: generic
I (1336) spi_flash: flash io: dio
W (1340) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (1352) main_task: Started on CPU0
I (1362) esp_psram: Reserving pool of 32K of internal memory for DMA/internal allocations
I (1362) main_task: Calling app_main()
I (1362) MAIN: MAIN STARTED
I (1382) MAIN: Initializing SD card using SDMMC peripheral
I (1392) gpio: GPIO[13]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0
I (1442) MAIN: after esp vfs fat sdmmc mount
I (1442) MAIN: SDCard mounted at /sdcard
Name: SK32G
Type: SDHC/SDXC
Speed: 20.00 MHz (limit: 20.00 MHz)
Size: 30436MB
CSD: ver=2, sector_size=512, capacity=62333952 read_bl_len=9
SSR: bus_width=4
I (1452) gpio: GPIO[25]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 1| Pulldown: 0| Intr:2
I (1462) cam_hal: cam init ok
ets Jul 29 2019 12:21:46

rst:0x3 (SW_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
```

Still failed i think



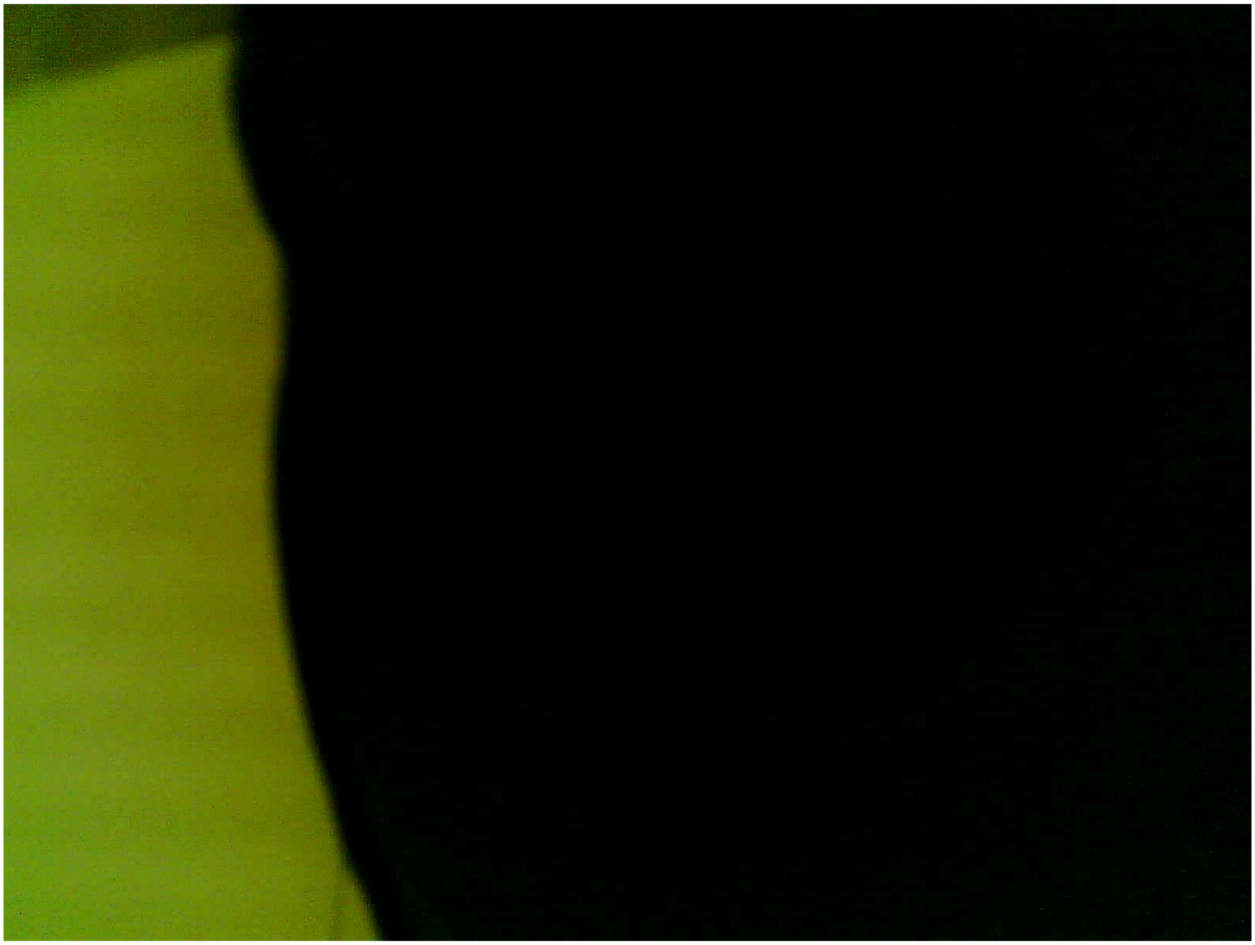
```
typedef enum {
    NO_MEAN                =  0,
    POWERON_RESET          =  1,    /**<1, Vbat power on reset*/
    SW_RESET               =  3,    /**<3, Software reset digital core*/
    OWDT_RESET             =  4,    /**<4, Legacy watch dog reset digital core*/
    DEEPSLEEP_RESET        =  5,    /**<3, Deep Sleep reset digital core*/
    SDIO_RESET             =  6,    /**<6, Reset by SLC module, reset digital core*/
    TG0WDT_SYS_RESET       =  7,    /**<7, Timer Group0 Watch dog reset digital core*/
    TG1WDT_SYS_RESET       =  8,    /**<8, Timer Group1 Watch dog reset digital core*/
    RTCWDT_SYS_RESET       =  9,    /**<9, RTC Watch dog Reset digital core*/
    INTRUSION_RESET        = 10,    /**<10, Instrusion tested to reset CPU*/
    TGWDT_CPU_RESET        = 11,    /**<11, Time Group reset CPU*/
    SW_CPU_RESET           = 12,    /**<12, Software reset CPU*/
    RTCWDT_CPU_RESET       = 13,    /**<13, RTC Watch dog Reset CPU*/
    EXT_CPU_RESET          = 14,    /**<14, for APP CPU, reseted by PRO CPU*/
    RTCWDT_BROWN_OUT_RESET = 15,    /**<15, Reset when the vdd voltage is not stable*/
    RTCWDT_RTC_RESET       = 16     /**<16, RTC Watch dog reset digital core and rtc module*/
} RESET_REASON;
```



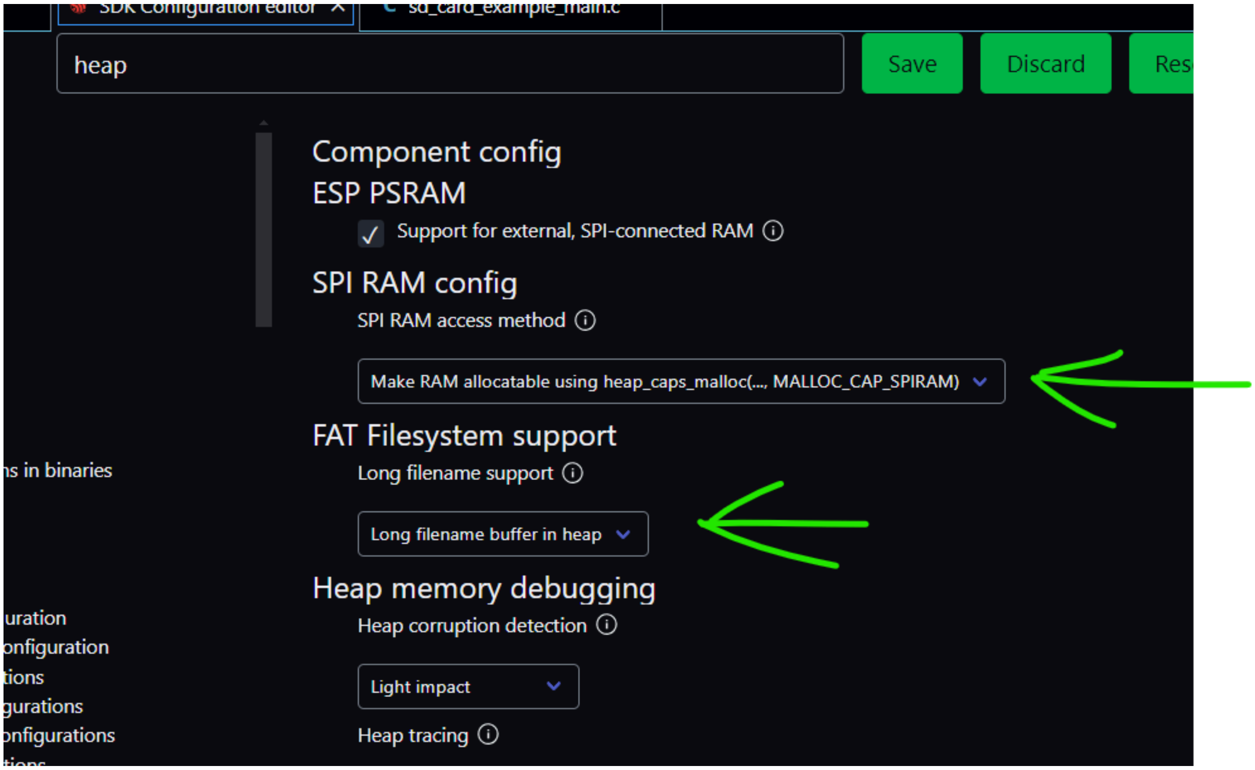
```
I (1338) spi_flash: flash io: dio
W (1342) i2c: This driver is an old driver, please migrate
I (1353) main_task: Started on CPU0
I (1363) main_task: Calling app_main()
Reset reason: 9
I (1363) MAIN: MAIN STARTED
I (4383) MAIN: Initializing SD card using SDMMC peripheral
I (4383) MAIN: before esp_vfs_fat_sdmmc_mount
```

RTCWDT_SYS_RESET = 9, /**<9, RTC Watch dog Reset digital core*/

Watchdog!!!



Picture after multiple resets



heap

Component config

ESP PSRAM

☑ Support for external, SPI-connected RAM ⓘ

SPI RAM config

SPI RAM access method ⓘ

Make RAM allocatable using heap_caps_malloc(..., MALLOC_CAP_SPIRAM)

FAT Filesystem support

Long filename support ⓘ

Long filename buffer in heap

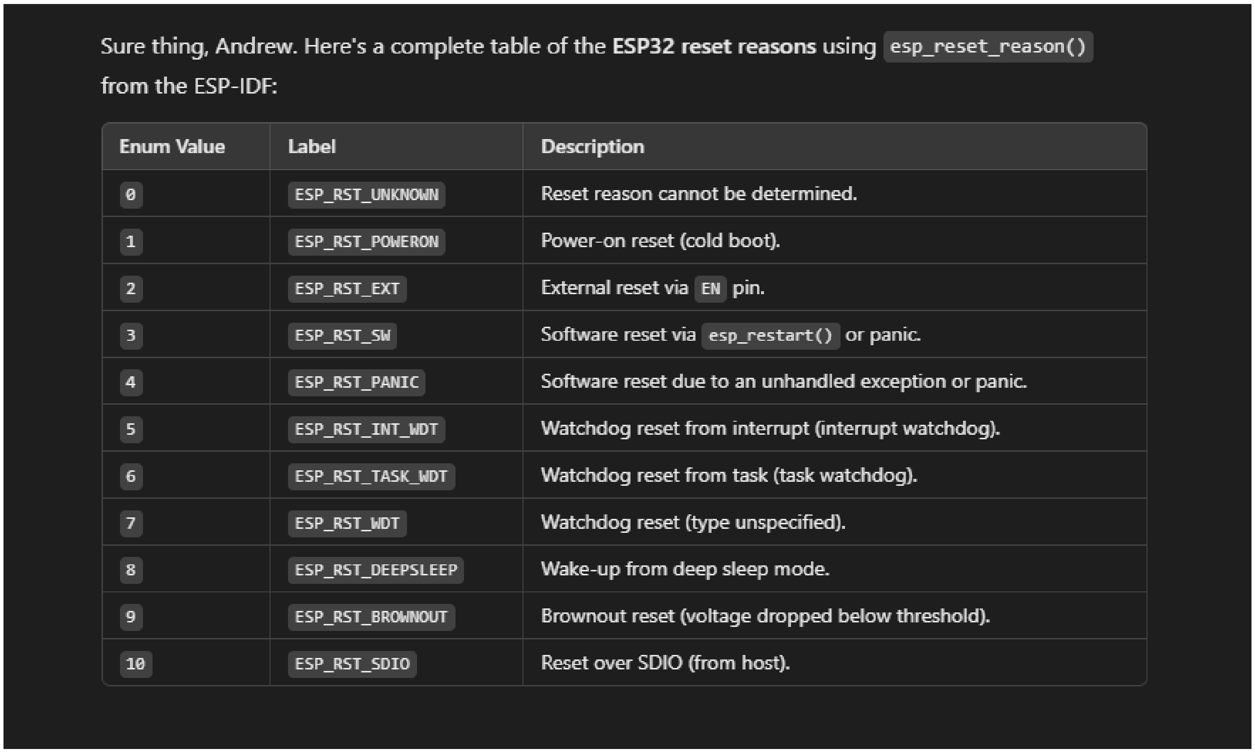
Heap memory debugging

Heap corruption detection ⓘ

Light impact

Heap tracing ⓘ

Latest changes

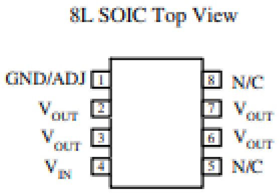


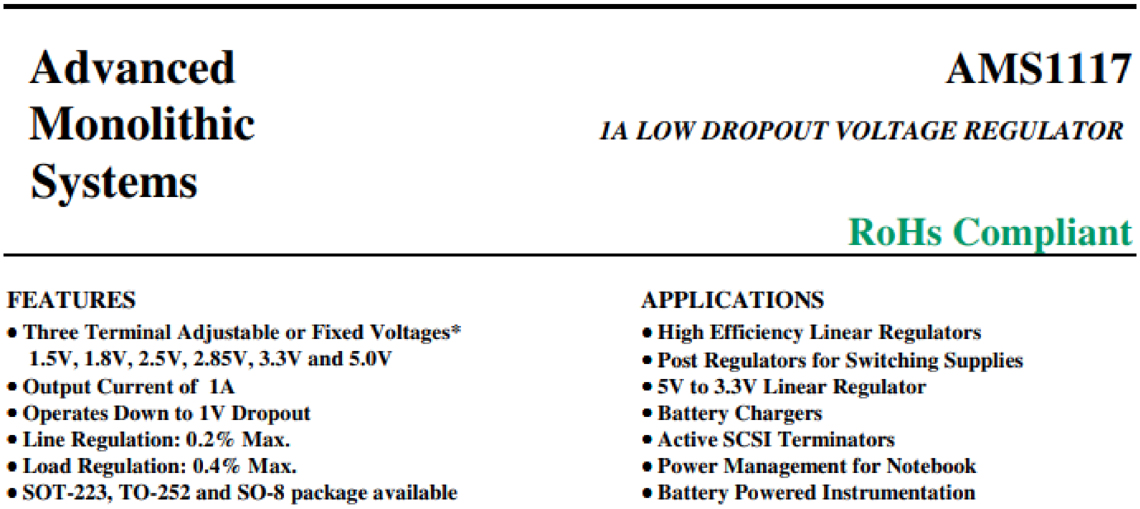
Sure thing, Andrew. Here's a complete table of the **ESP32 reset reasons** using `esp_reset_reason()` from the ESP-IDF:

Enum Value	Label	Description
`0`	`ESP_RST_UNKNOWN`	Reset reason cannot be determined.
`1`	`ESP_RST_POWERON`	Power-on reset (cold boot).
`2`	`ESP_RST_EXT`	External reset via `EN` pin.
`3`	`ESP_RST_SW`	Software reset via `esp_restart()` or panic.
`4`	`ESP_RST_PANIC`	Software reset due to an unhandled exception or panic.
`5`	`ESP_RST_INT_WDT`	Watchdog reset from interrupt (interrupt watchdog).
`6`	`ESP_RST_TASK_WDT`	Watchdog reset from task (task watchdog).
`7`	`ESP_RST_WDT`	Watchdog reset (type unspecified).
`8`	`ESP_RST_DEEPSLEEP`	Wake-up from deep sleep mode.
`9`	`ESP_RST_BROWNOUT`	Brownout reset (voltage dropped below threshold).
`10`	`ESP_RST_SDIO`	Reset over SDIO (from host).

Reasons dont line up with chatgpt reasons

I think its a brown out





Advanced Monolithic Systems

AMS1117

1A LOW DROPOUT VOLTAGE REGULATOR

RoHs Compliant

FEATURES

- **Three Terminal Adjustable or Fixed Voltages***
  1.5V, 1.8V, 2.5V, 2.85V, 3.3V and 5.0V
- **Output Current of 1A**
- **Operates Down to 1V Dropout**
- **Line Regulation: 0.2% Max.**
- **Load Regulation: 0.4% Max.**
- **SOT-223, TO-252 and SO-8 package available**

APPLICATIONS

- **High Efficiency Linear Regulators**
- **Post Regulators for Switching Supplies**
- **5V to 3.3V Linear Regulator**
- **Battery Chargers**
- **Active SCSI Terminators**
- **Power Management for Notebook**
- **Battery Powered Instrumentation**

Measured current data (no regulator)

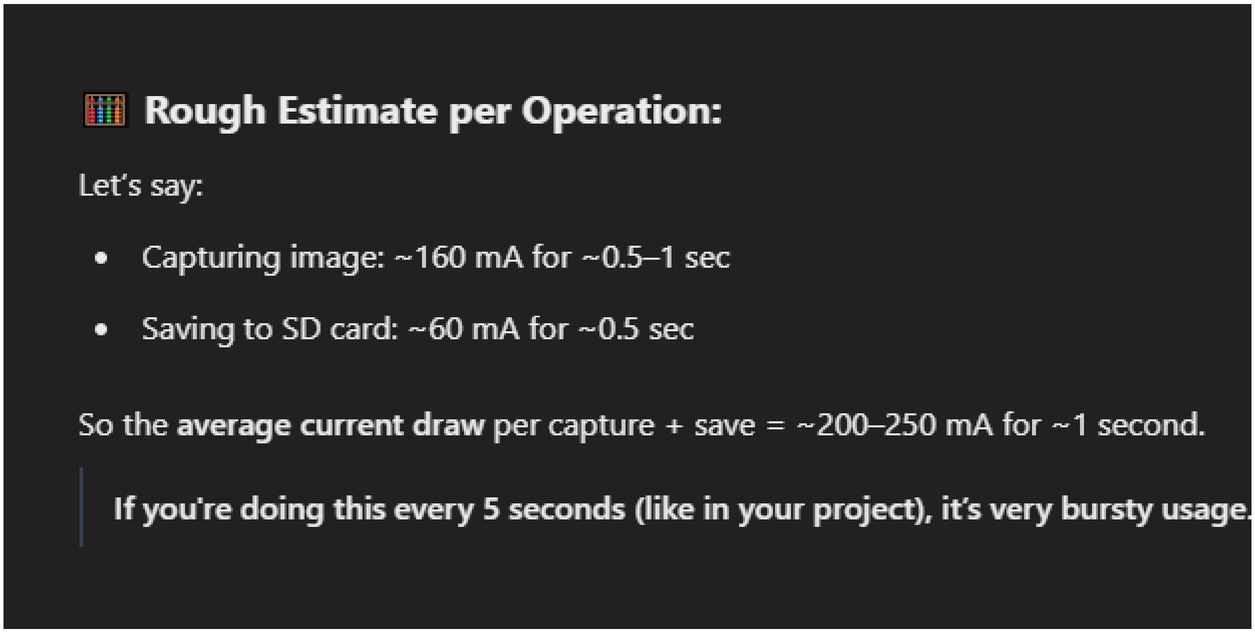
5V esp32 WROOM
1.182A

FTD board
But it varies wildly. On the FTD board
Up to 0.902
Down to 0.52

Measured current data (with regulator)

5V esp32 WROOM
1.182A

FTD board
But it varies wildly. On the FTD board
Up to 0.50
Down to 0.48



🧮 Rough Estimate per Operation:

Let's say:

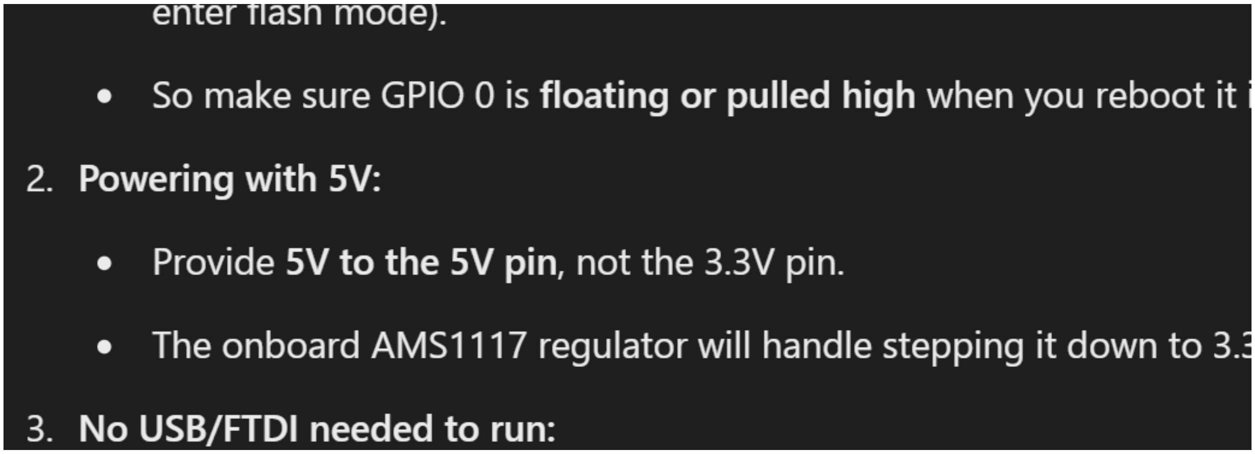
- Capturing image: ~160 mA for ~0.5–1 sec
- Saving to SD card: ~60 mA for ~0.5 sec

So the **average current draw** per capture + save = ~200–250 mA for ~1 second.

> **If you're doing this every 5 seconds (like in your project), it's very bursty usage.**

It wasnt working with the voltage regulator setup. Maybe it has to be connected to FTDI

Or maybe those pics were from the wroooom



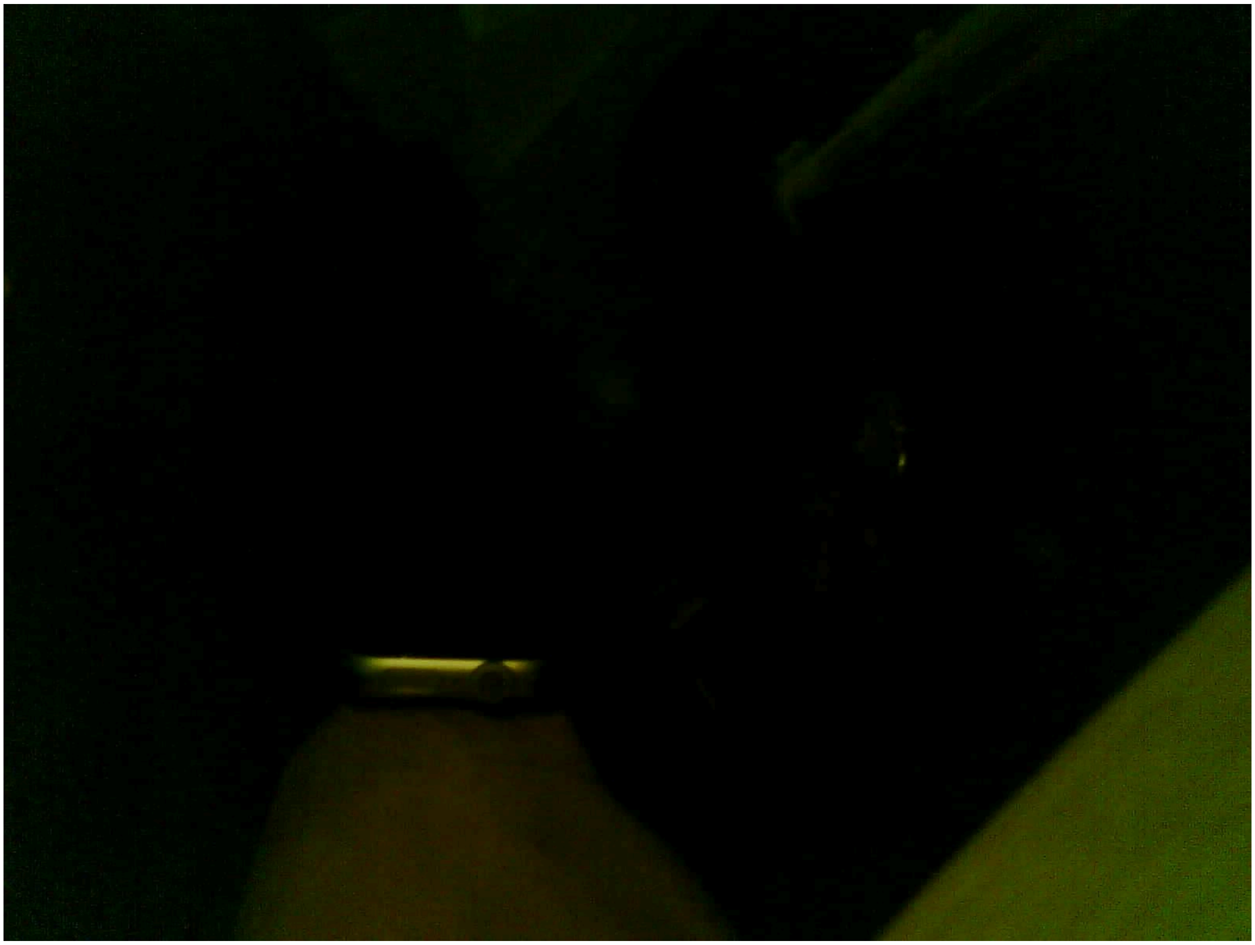
enter flash mode).

   - So make sure GPIO 0 is **floating or pulled high** when you reboot it

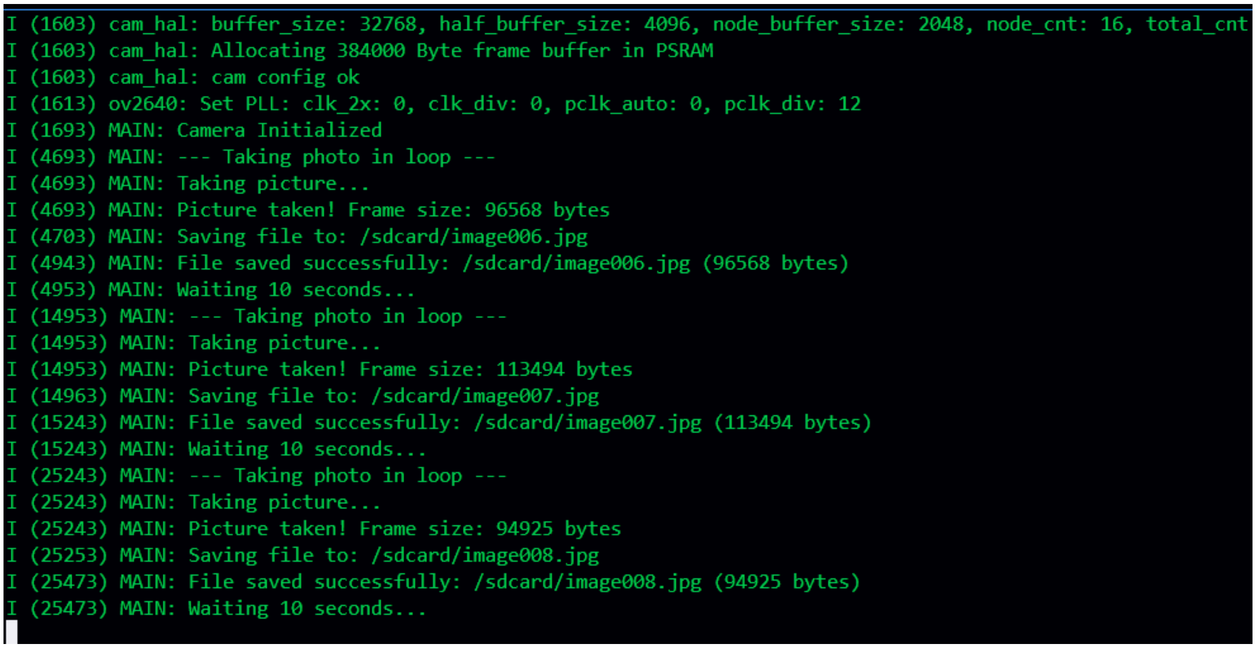
2. **Powering with 5V:**
   - Provide **5V to the 5V pin**, not the 3.3V pin.
   - The onboard AMS1117 regulator will handle stepping it down to 3.3

3. **No USB/FTDI needed to run:**

This is a 3.3v regulator !!!!!!!!!!

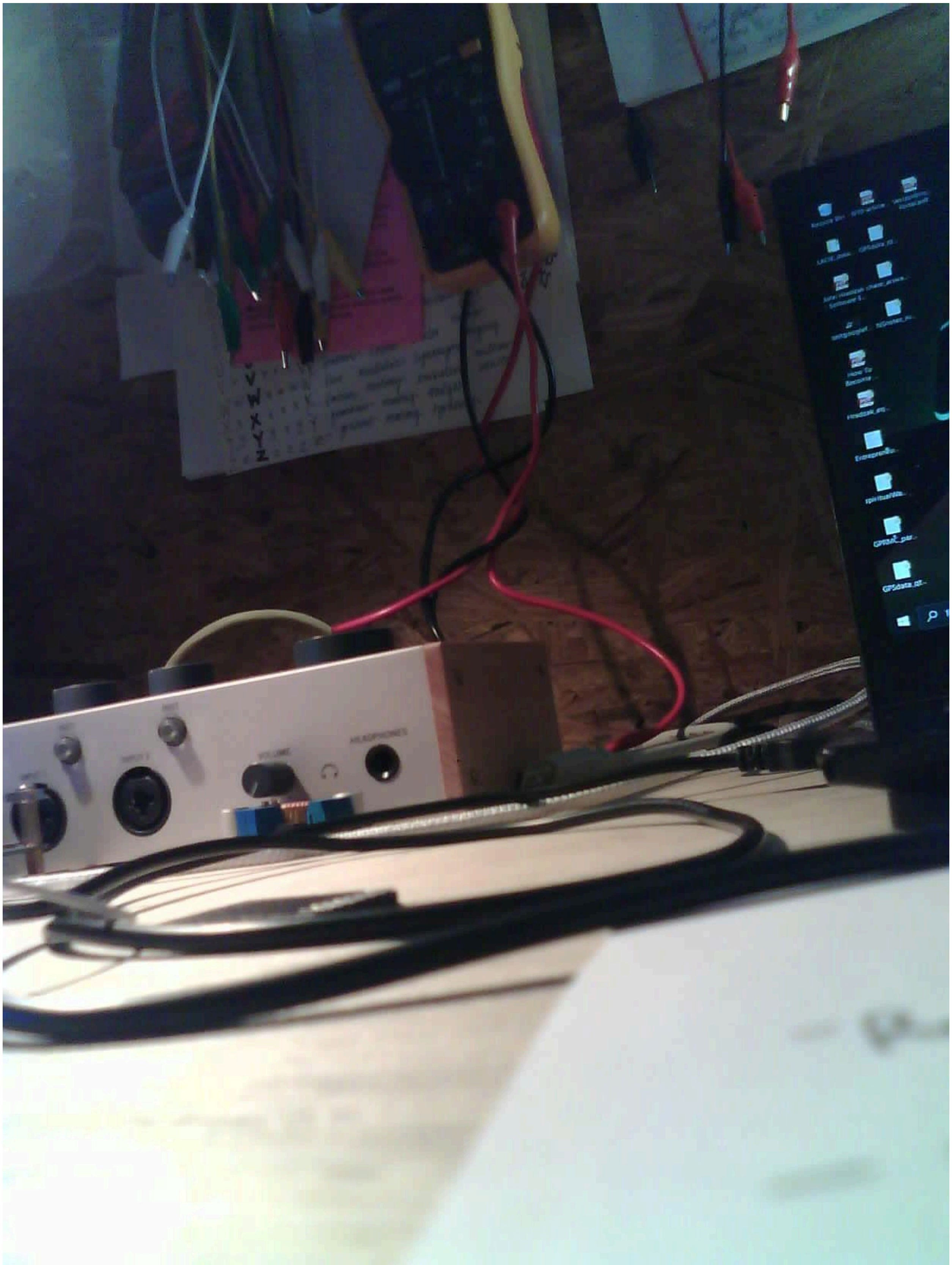


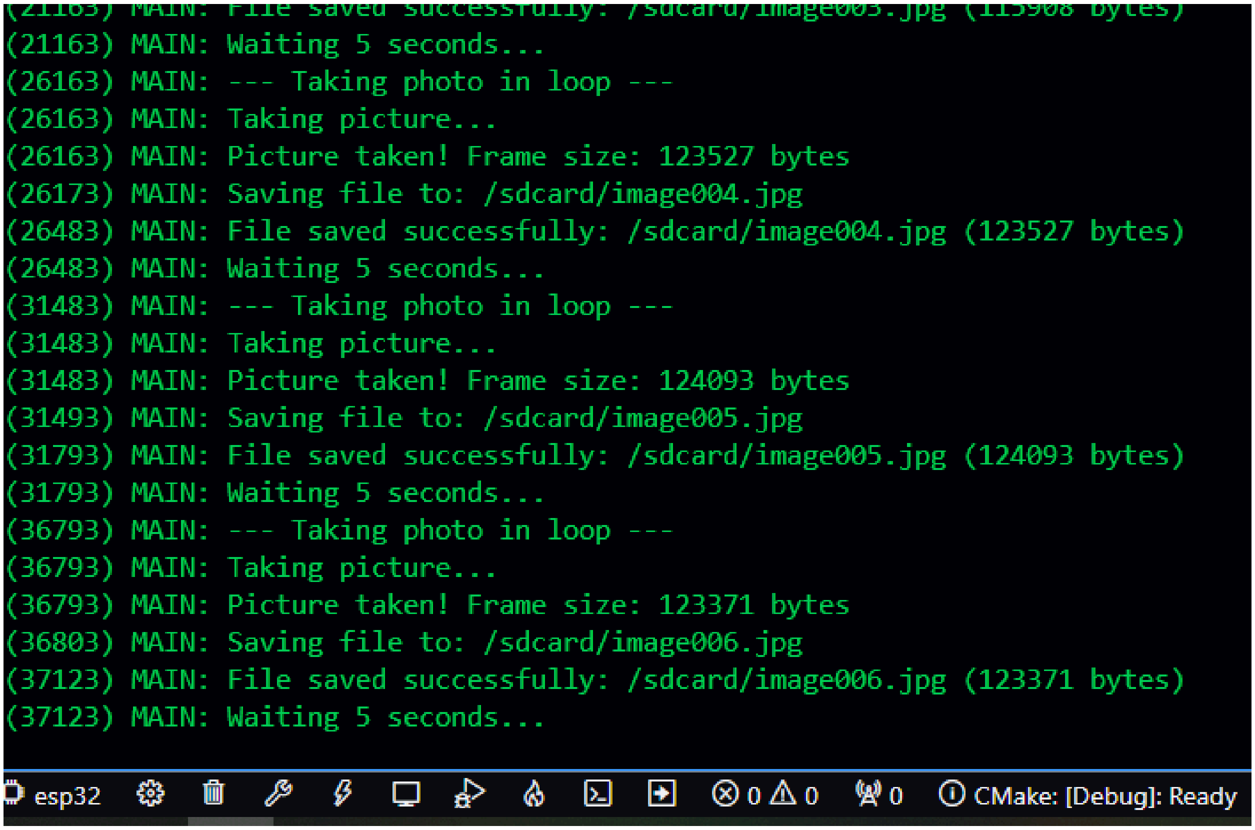
First detached photo!!!
Taken from the 5v line on the wroom!!!



```
I (1603) cam_hal: buffer_size: 32768, half_buffer_size: 4096, node_buffer_size: 2048, node_cnt: 16, total_cnt
I (1603) cam_hal: Allocating 384000 Byte frame buffer in PSRAM
I (1603) cam_hal: cam config ok
I (1613) ov2640: Set PLL: clk_2x: 0, clk_div: 0, pclk_auto: 0, pclk_div: 12
I (1693) MAIN: Camera Initialized
I (4693) MAIN: --- Taking photo in loop ---
I (4693) MAIN: Taking picture...
I (4693) MAIN: Picture taken! Frame size: 96568 bytes
I (4703) MAIN: Saving file to: /sdcard/image006.jpg
I (4943) MAIN: File saved successfully: /sdcard/image006.jpg (96568 bytes)
I (4953) MAIN: Waiting 10 seconds...
I (14953) MAIN: --- Taking photo in loop ---
I (14953) MAIN: Taking picture...
I (14953) MAIN: Picture taken! Frame size: 113494 bytes
I (14963) MAIN: Saving file to: /sdcard/image007.jpg
I (15243) MAIN: File saved successfully: /sdcard/image007.jpg (113494 bytes)
I (15243) MAIN: Waiting 10 seconds...
I (25243) MAIN: --- Taking photo in loop ---
I (25243) MAIN: Taking picture...
I (25243) MAIN: Picture taken! Frame size: 94925 bytes
I (25253) MAIN: Saving file to: /sdcard/image008.jpg
I (25473) MAIN: File saved successfully: /sdcard/image008.jpg (94925 bytes)
I (25473) MAIN: Waiting 10 seconds...
```

Taking multi image!!


V
W
X
Y
Z
INPUT 2
VOLUME
HEADPHONES

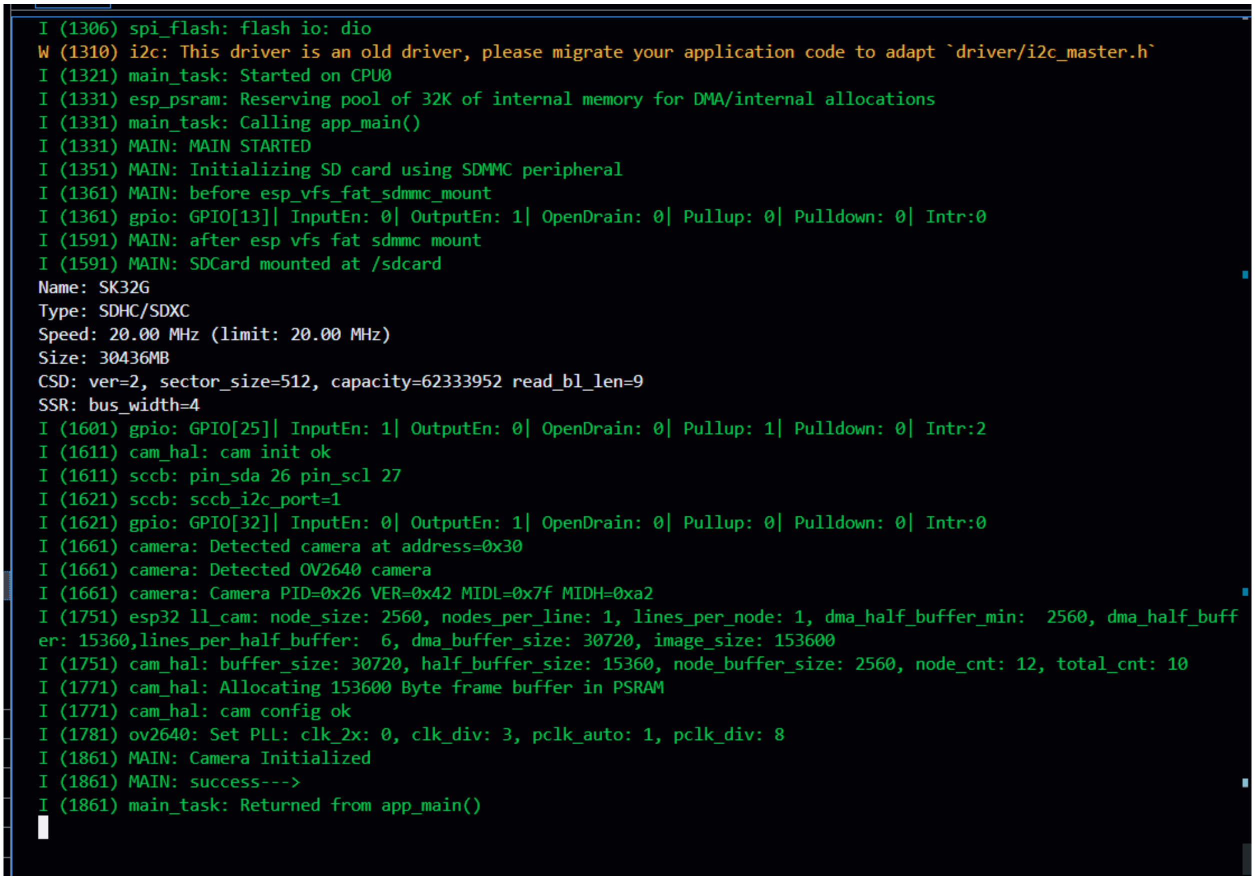


```
(21163) MAIN: Waiting 5 seconds...
(26163) MAIN: --- Taking photo in loop ---
(26163) MAIN: Taking picture...
(26163) MAIN: Picture taken! Frame size: 123527 bytes
(26173) MAIN: Saving file to: /sdcard/image004.jpg
(26483) MAIN: File saved successfully: /sdcard/image004.jpg (123527 bytes)
(26483) MAIN: Waiting 5 seconds...
(31483) MAIN: --- Taking photo in loop ---
(31483) MAIN: Taking picture...
(31483) MAIN: Picture taken! Frame size: 124093 bytes
(31493) MAIN: Saving file to: /sdcard/image005.jpg
(31793) MAIN: File saved successfully: /sdcard/image005.jpg (124093 bytes)
(31793) MAIN: Waiting 5 seconds...
(36793) MAIN: --- Taking photo in loop ---
(36793) MAIN: Taking picture...
(36793) MAIN: Picture taken! Frame size: 123371 bytes
(36803) MAIN: Saving file to: /sdcard/image006.jpg
(37123) MAIN: File saved successfully: /sdcard/image006.jpg (123371 bytes)
(37123) MAIN: Waiting 5 seconds...
```

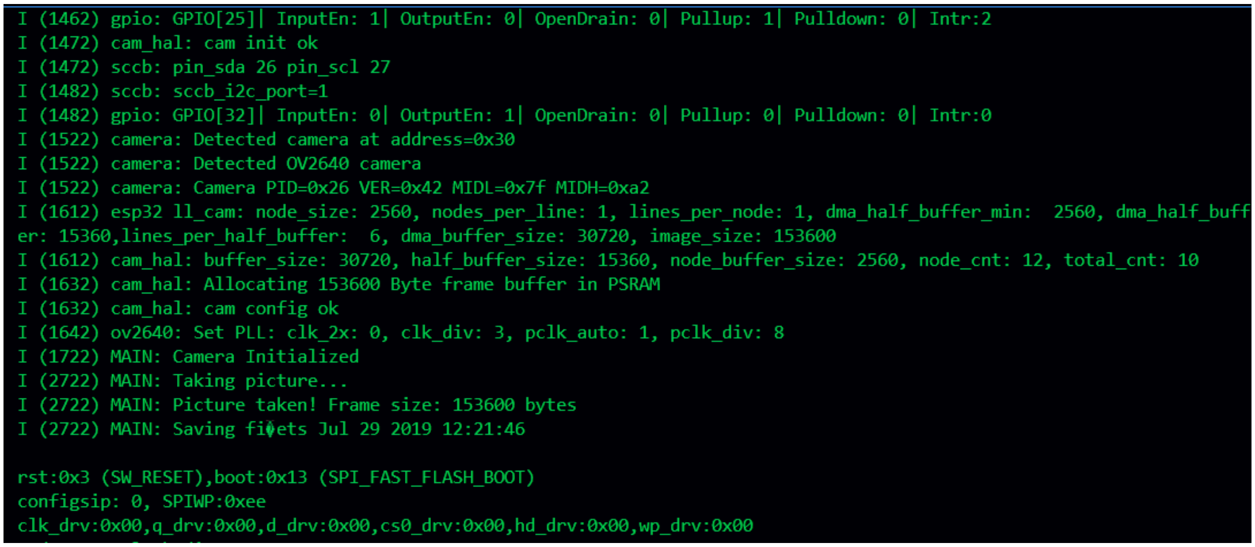
Every 5secs works

I think its my laptop USB ports that are giving the power/current issues!!
It works everytime on teh power brick

Cycled power cable then removed/returned sd card

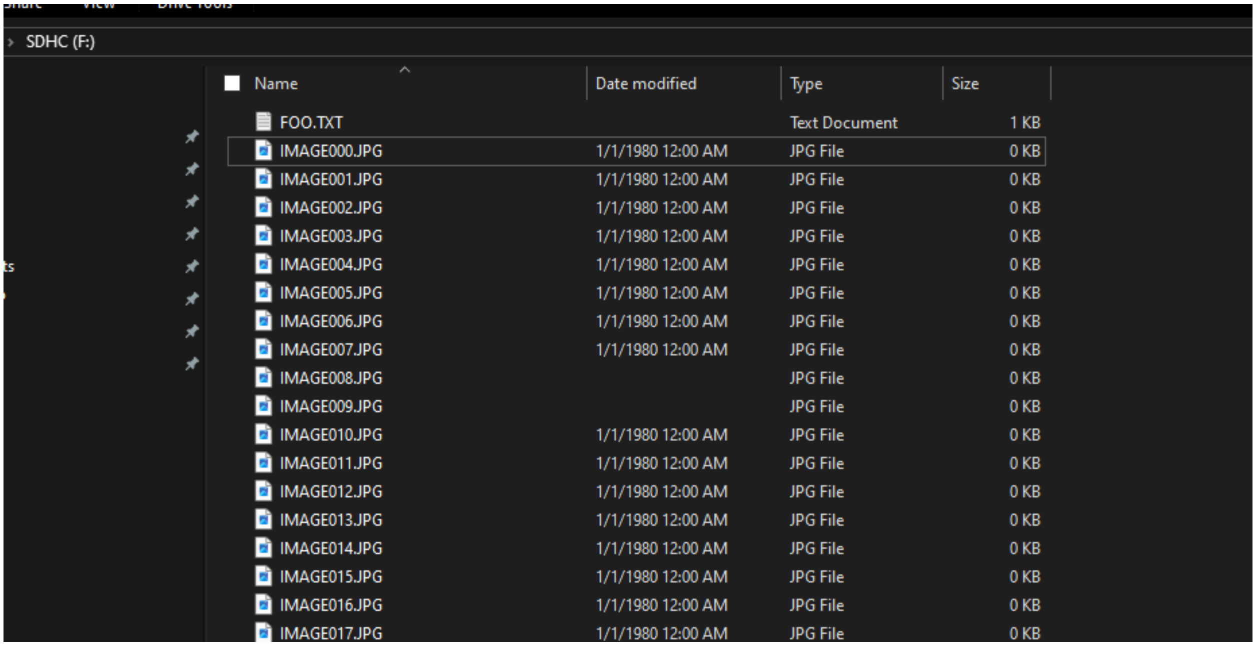


```
I (1306) spi_flash: flash io: dio
W (1310) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (1321) main_task: Started on CPU0
I (1331) esp_psram: Reserving pool of 32K of internal memory for DMA/internal allocations
I (1331) main_task: Calling app_main()
I (1331) MAIN: MAIN STARTED
I (1351) MAIN: Initializing SD card using SDMMC peripheral
I (1361) MAIN: before esp_vfs_fat_sdmmc_mount
I (1361) gpio: GPIO[13]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0
I (1591) MAIN: after esp vfs fat sdmmc mount
I (1591) MAIN: SDCard mounted at /sdcard
Name: SK32G
Type: SDHC/SDXC
Speed: 20.00 MHz (limit: 20.00 MHz)
Size: 30436MB
CSD: ver=2, sector_size=512, capacity=62333952 read_bl_len=9
SSR: bus_width=4
I (1601) gpio: GPIO[25]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 1| Pulldown: 0| Intr:2
I (1611) cam_hal: cam init ok
I (1611) sccb: pin_sda 26 pin_scl 27
I (1621) sccb: sccb_i2c_port=1
I (1621) gpio: GPIO[32]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0
I (1661) camera: Detected camera at address=0x30
I (1661) camera: Detected OV2640 camera
I (1661) camera: Camera PID=0x26 VER=0x42 MIDL=0x7f MIDH=0xa2
I (1751) esp32 ll_cam: node_size: 2560, nodes_per_line: 1, lines_per_node: 1, dma_half_buffer_min:  2560, dma_half_buff
er: 15360,lines_per_half_buffer:  6, dma_buffer_size: 30720, image_size: 153600
I (1751) cam_hal: buffer_size: 30720, half_buffer_size: 15360, node_buffer_size: 2560, node_cnt: 12, total_cnt: 10
I (1771) cam_hal: Allocating 153600 Byte frame buffer in PSRAM
I (1771) cam_hal: cam config ok
I (1781) ov2640: Set PLL: clk_2x: 0, clk_div: 3, pclk_auto: 1, pclk_div: 8
I (1861) MAIN: Camera Initialized
I (1861) MAIN: success--->
I (1861) main_task: Returned from app_main()
```



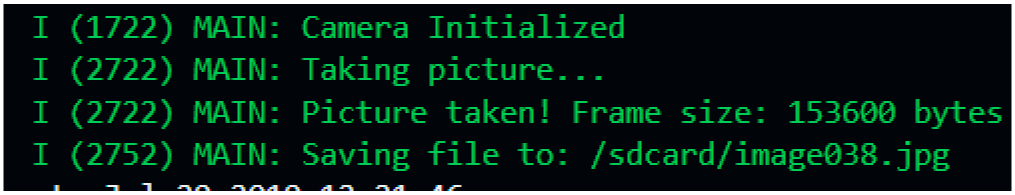
```
I (1462) gpio: GPIO[25]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 1| Pulldown: 0| Intr:2
I (1472) cam_hal: cam init ok
I (1472) sccb: pin_sda 26 pin_scl 27
I (1482) sccb: sccb_i2c_port=1
I (1482) gpio: GPIO[32]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0
I (1522) camera: Detected camera at address=0x30
I (1522) camera: Detected OV2640 camera
I (1522) camera: Camera PID=0x26 VER=0x42 MIDL=0x7f MIDH=0xa2
I (1612) esp32 ll_cam: node_size: 2560, nodes_per_line: 1, lines_per_node: 1, dma_half_buffer_min:  2560, dma_half_buff
er: 15360,lines_per_half_buffer:  6, dma_buffer_size: 30720, image_size: 153600
I (1612) cam_hal: buffer_size: 30720, half_buffer_size: 15360, node_buffer_size: 2560, node_cnt: 12, total_cnt: 10
I (1632) cam_hal: Allocating 153600 Byte frame buffer in PSRAM
I (1632) cam_hal: cam config ok
I (1642) ov2640: Set PLL: clk_2x: 0, clk_div: 3, pclk_auto: 1, pclk_div: 8
I (1722) MAIN: Camera Initialized
I (2722) MAIN: Taking picture...
I (2722) MAIN: Picture taken! Frame size: 153600 bytes
I (2722) MAIN: Saving fi�ets Jul 29 2019 12:21:46

rst:0x3 (SW_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
```

Picture taken!!



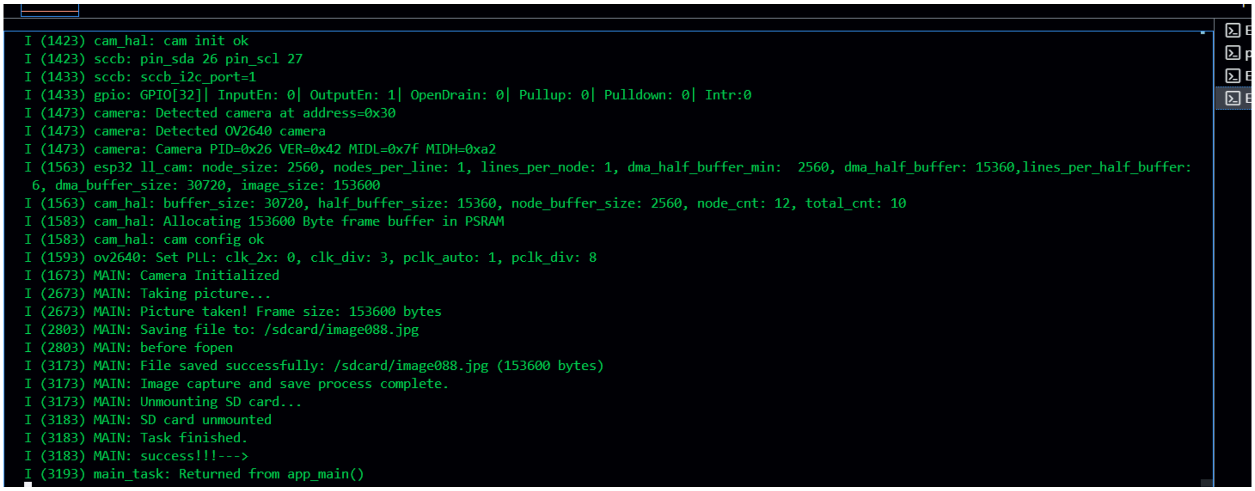
Name	Date modified	Type	Size
FOO.TXT		Text Document	1 KB
IMAGE000.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE001.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE002.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE003.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE004.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE005.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE006.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE007.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE008.JPG		JPG File	0 KB
IMAGE009.JPG		JPG File	0 KB
IMAGE010.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE011.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE012.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE013.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE014.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE015.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE016.JPG	1/1/1980 12:00 AM	JPG File	0 KB
IMAGE017.JPG	1/1/1980 12:00 AM	JPG File	0 KB

No data saved but the images are saving!!



```
I (1722) MAIN: Camera Initialized
I (2722) MAIN: Taking picture...
I (2722) MAIN: Picture taken! Frame size: 153600 bytes
I (2752) MAIN: Saving file to: /sdcard/image038.jpg
```

Says 153600 bytes image??



```
I (1423) cam_hal: cam init ok
I (1423) sccb: pin_sda 26 pin_scl 27
I (1433) sccb: sccb_i2c_port=1
I (1433) gpio: GPIO[32]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0
I (1473) camera: Detected camera at address=0x30
I (1473) camera: Detected OV2640 camera
I (1473) camera: Camera PID=0x26 VER=0x42 MIDL=0x7f MIDH=0xa2
I (1563) esp32 ll_cam: node_size: 2560, nodes_per_line: 1, lines_per_node: 1, dma_half_buffer_min:  2560, dma_half_buffer: 15360,lines_per_half_buffer:
 6, dma_buffer_size: 30720, image_size: 153600
I (1563) cam_hal: buffer_size: 30720, half_buffer_size: 15360, node_buffer_size: 2560, node_cnt: 12, total_cnt: 10
I (1583) cam_hal: Allocating 153600 Byte frame buffer in PSRAM
I (1583) cam_hal: cam config ok
I (1593) ov2640: Set PLL: clk_2x: 0, clk_div: 3, pclk_auto: 1, pclk_div: 8
I (1673) MAIN: Camera Initialized
I (2673) MAIN: Taking picture...
I (2673) MAIN: Picture taken! Frame size: 153600 bytes
I (2803) MAIN: Saving file to: /sdcard/image088.jpg
I (2803) MAIN: before fopen
I (3173) MAIN: File saved successfully: /sdcard/image088.jpg (153600 bytes)
I (3173) MAIN: Image capture and save process complete.
I (3173) MAIN: Unmounting SD card...
I (3183) MAIN: SD card unmounted
I (3183) MAIN: Task finished.
I (3183) MAIN: success!!!--->
I (3193) main_task: Returned from app_main()
```

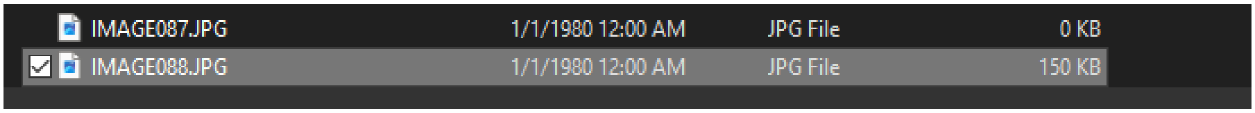
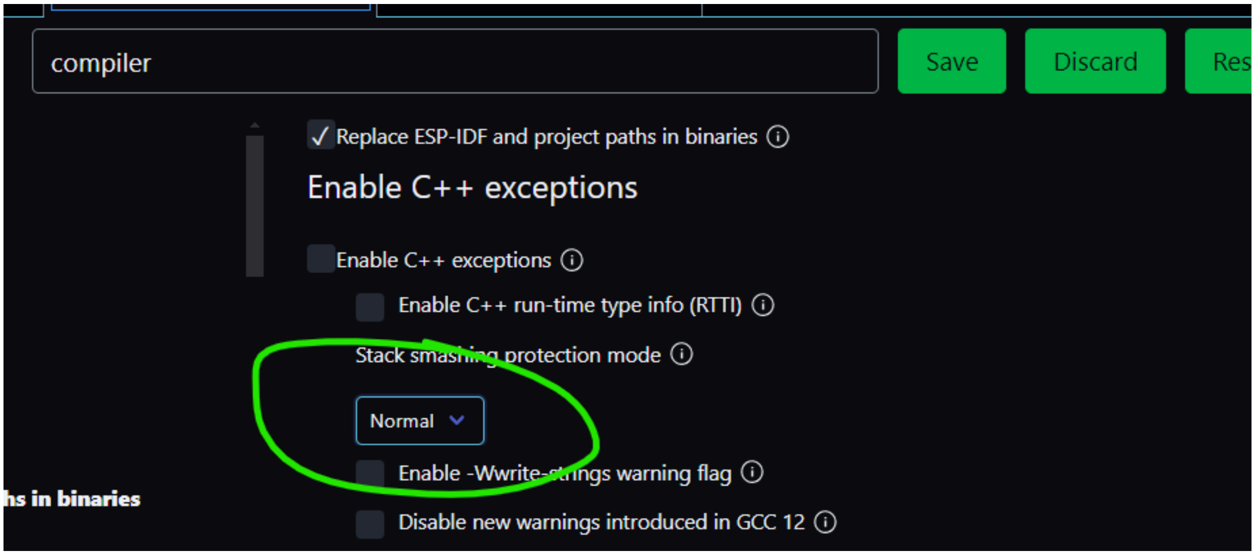


IMAGE087.JPG	1/1/1980 12:00 AM	JPG File	0 KB
☑ IMAGE088.JPG	1/1/1980 12:00 AM	JPG File	150 KB



☑ Replace ESP-IDF and project paths in binaries

Enable C++ exceptions

☐ Enable C++ exceptions

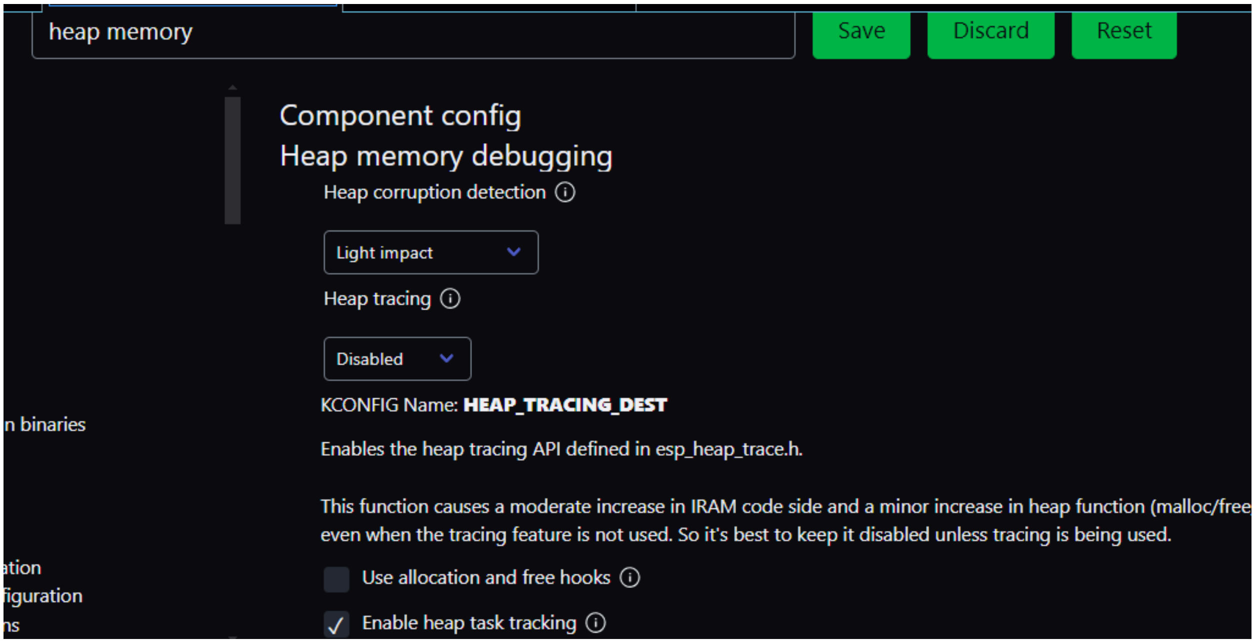
☐ Enable C++ run-time type info (RTTI)

Stack smashing protection mode

Normal

☐ Enable -Wwrite-strings warning flag

☐ Disable new warnings introduced in GCC 12



Component config

Heap memory debugging

Heap corruption detection

Light impact

Heap tracing

Disabled

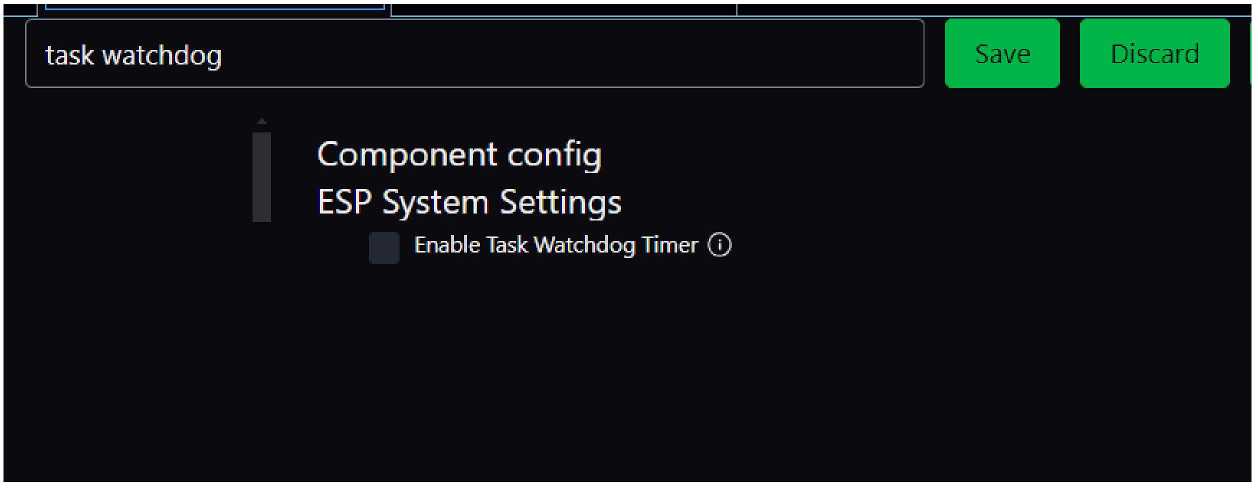
KCONFIG Name: **HEAP_TRACING_DEST**

Enables the heap tracing API defined in esp_heap_trace.h.

This function causes a moderate increase in IRAM code side and a minor increase in heap function (malloc/free
even when the tracing feature is not used. So it's best to keep it disabled unless tracing is being used.

☐ Use allocation and free hooks

☑ Enable heap task tracking



task watchdog

Component config
ESP System Settings
☐ Enable Task Watchdog Timer

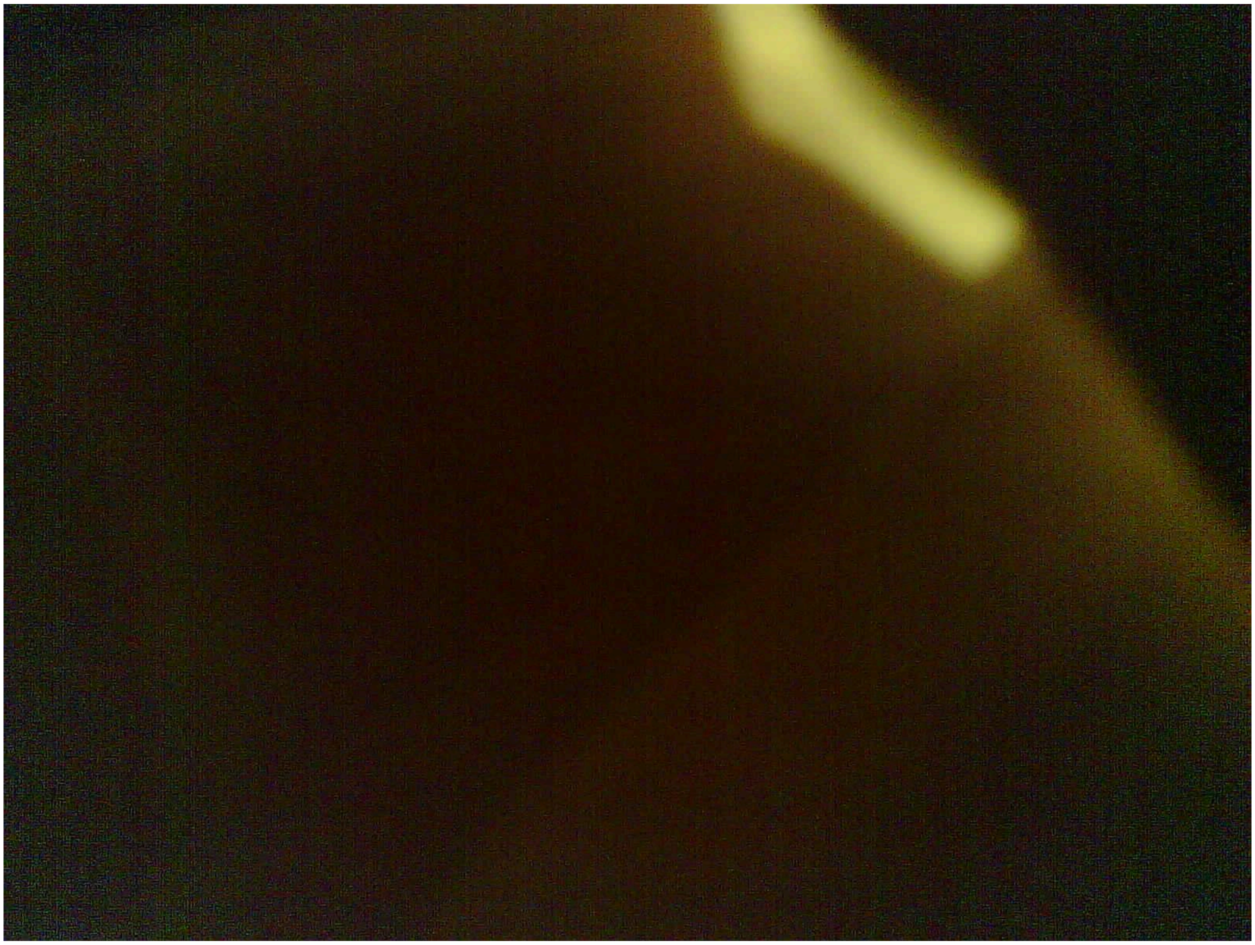
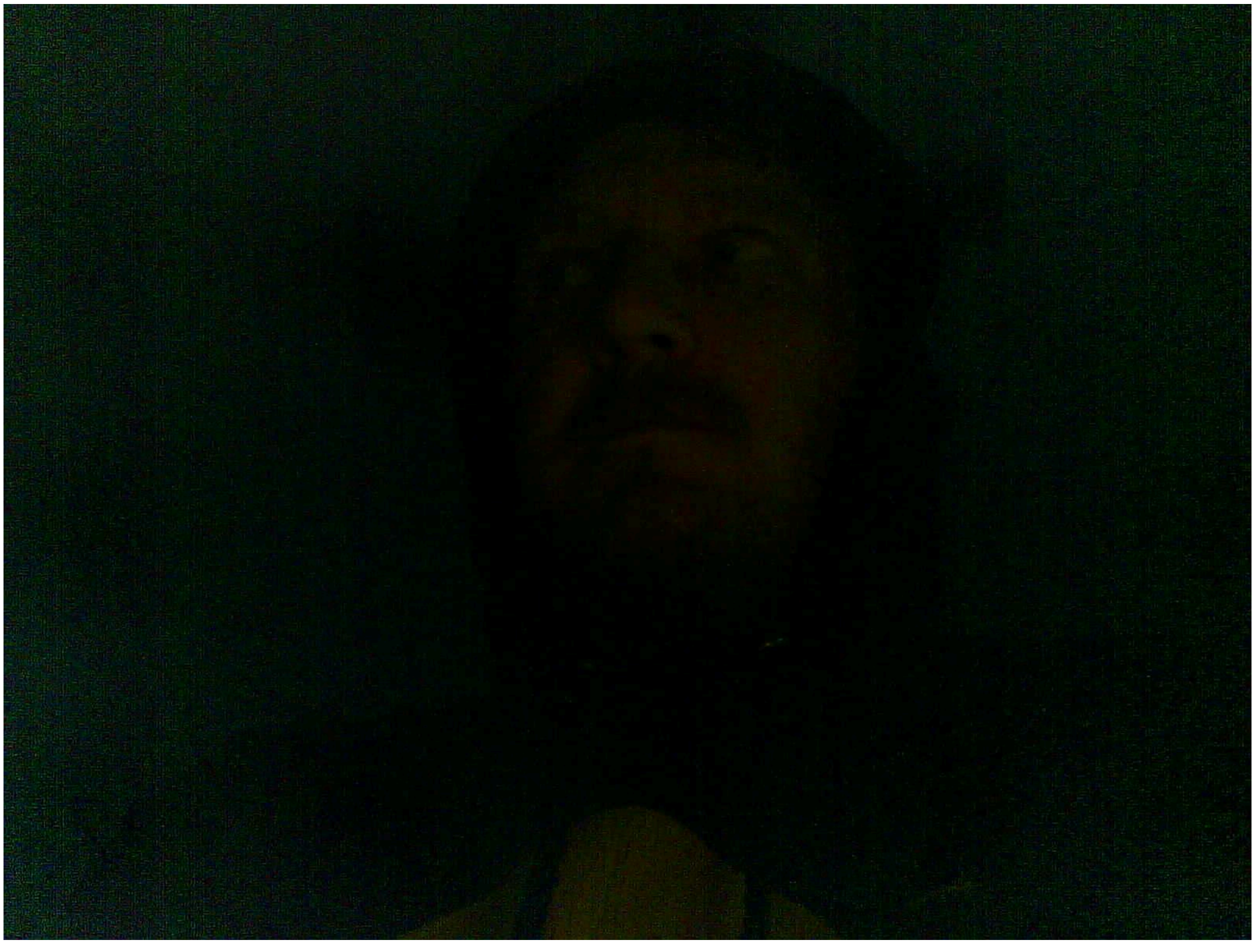
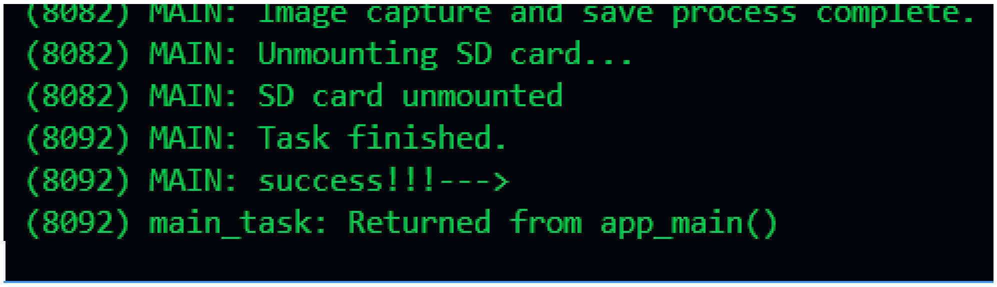
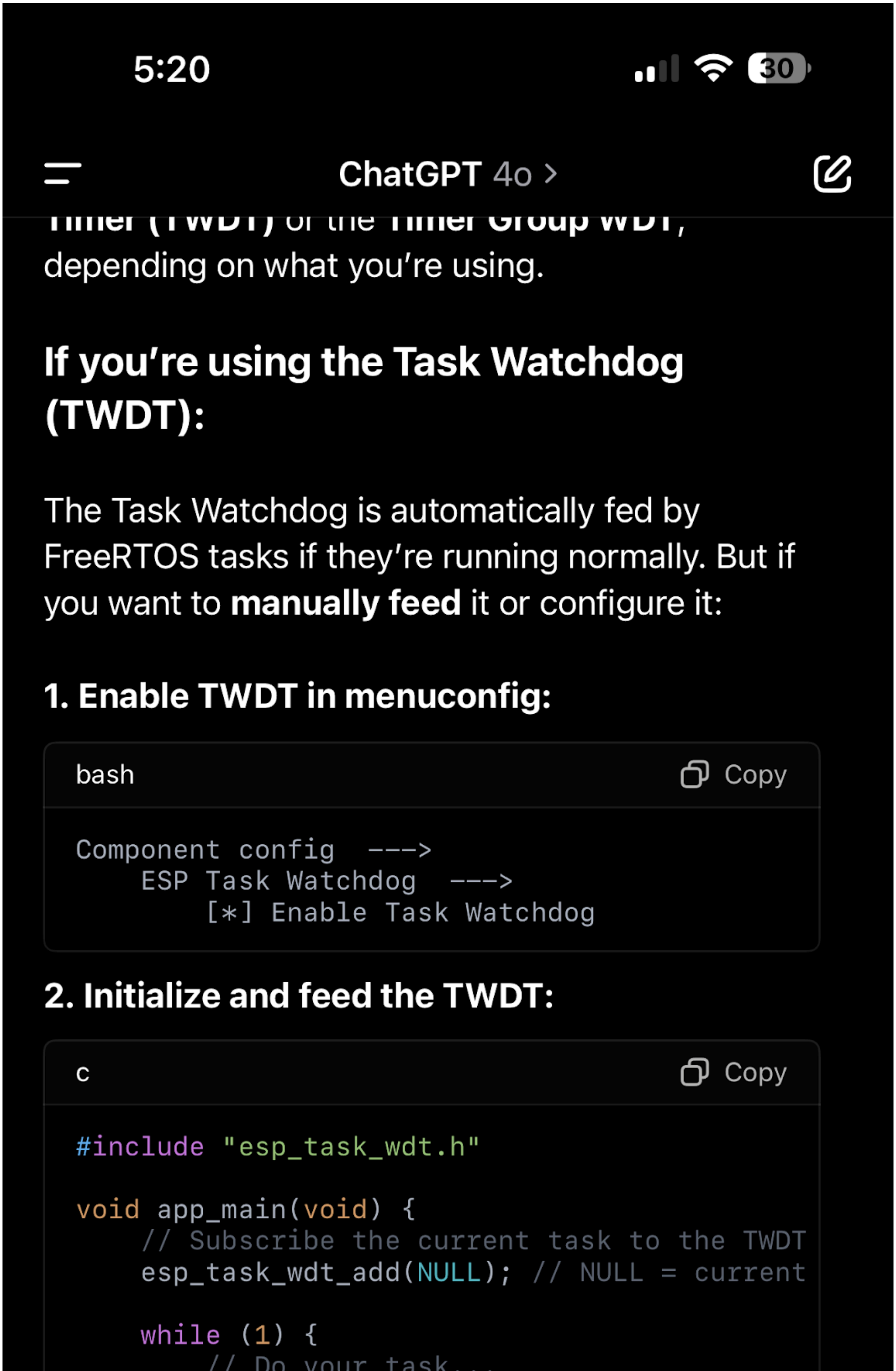


Image taken!!!!!!





```
(8082) MAIN: Image capture and save process complete.
(8082) MAIN: Unmounting SD card...
(8082) MAIN: SD card unmounted
(8092) MAIN: Task finished.
(8092) MAIN: success!!!--->
(8092) main_task: Returned from app_main()
```



Timer (TWDT) or the **Timer Group WDT**, depending on what you're using.

If you're using the Task Watchdog (TWDT):

The Task Watchdog is automatically fed by FreeRTOS tasks if they're running normally. But if you want to **manually feed** it or configure it:

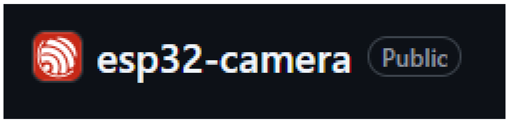
1. Enable TWDT in menuconfig:

```bash
Component config  --->
    ESP Task Watchdog  --->
        [*] Enable Task Watchdog
```

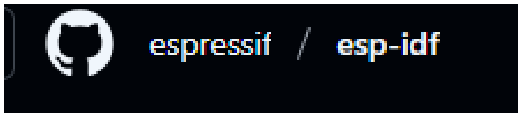
2. Initialize and feed the TWDT:

```c
#include "esp_task_wdt.h"

void app_main(void) {
    // Subscribe the current task to the TWDT
    esp_task_wdt_add(NULL); // NULL = current

    while (1) {
        // Do your task...
```

3/26/2025


esp32-camera Public

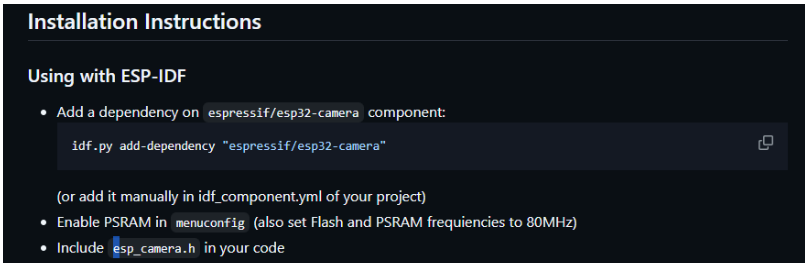
esp camera ov2640 Githib link


espressif / esp-idf

Sdmmc driver examples github link

Next steps

- ☑ ~~Create new project from the examples/storage/sd_card/sdmmc example~~
- ☑ ~~Rename it Naturescanner-OV2640~~
- ☑ ~~Add a dependency on espressif/esp32-camera component (camera read me)~~



Installation Instructions

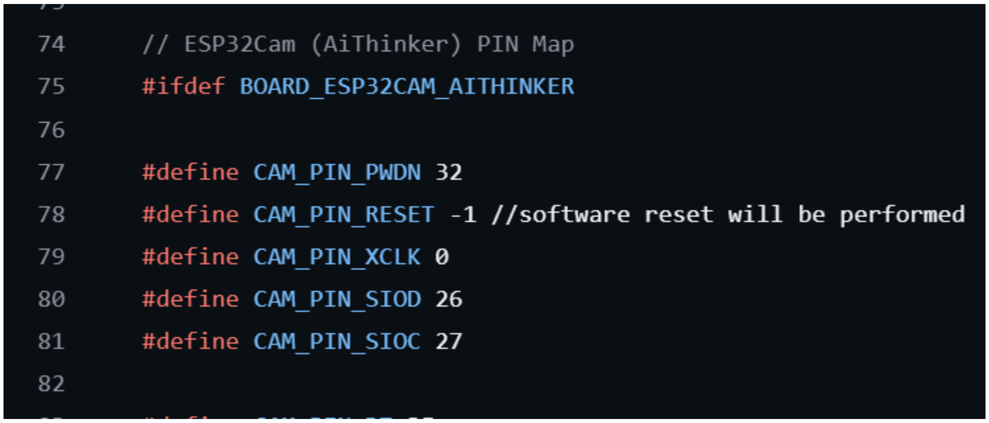
Using with ESP-IDF

- Add a dependency on `espressif/esp32-camera` component:

```
idf.py add-dependency "espressif/esp32-camera"
```

(or add it manually in idf_component.yml of your project)

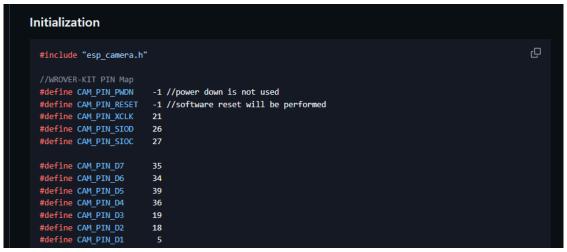
- Enable PSRAM in `menuconfig` (also set Flash and PSRAM frequiencies to 80MHz)
- Include `esp_camera.h` in your code



```
// ESP32Cam (AiThinker) PIN Map
#ifdef BOARD_ESP32CAM_AITHINKER

#define CAM_PIN_PWDN 32
#define CAM_PIN_RESET -1 //software reset will be performed
#define CAM_PIN_XCLK 0
#define CAM_PIN_SIOD 26
#define CAM_PIN_SIOC 27

```

Use esp32CAM pin map from take_picture.c

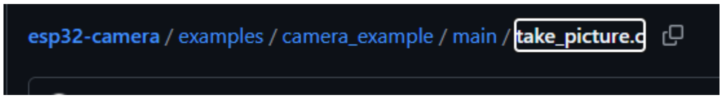
- ☑ ~~Add camera initialization code until all is incorporate and can run~~



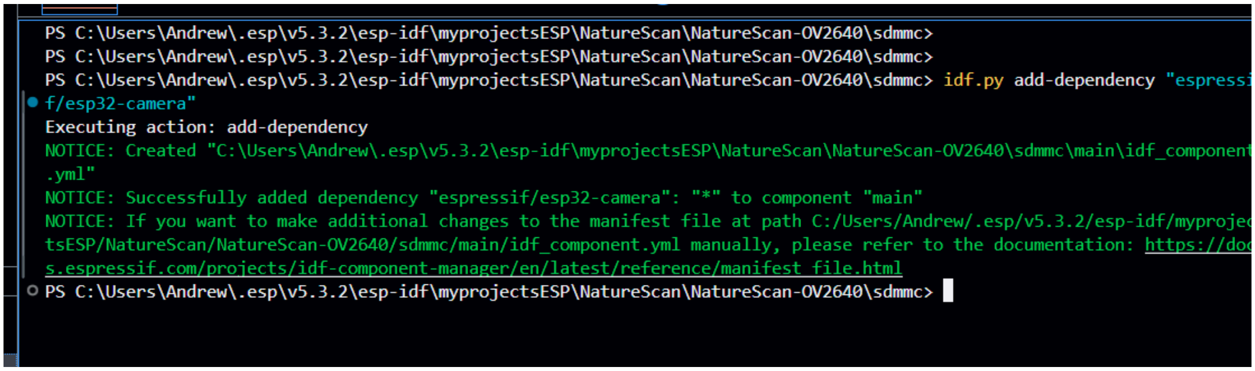
Initialization

```
#include "esp_camera.h"

//WROVER-KIT PIN Map
#define CAM_PIN_PWDN    -1 //power down is not used
#define CAM_PIN_RESET   -1 //software reset will be performed
#define CAM_PIN_XCLK    21
#define CAM_PIN_SIOD    26
#define CAM_PIN_SIOC    27

#define CAM_PIN_D7      35
#define CAM_PIN_D6      34
#define CAM_PIN_D5      39
#define CAM_PIN_D4      36
#define CAM_PIN_D3      19
#define CAM_PIN_D2      18
#define CAM_PIN_D1       5
```

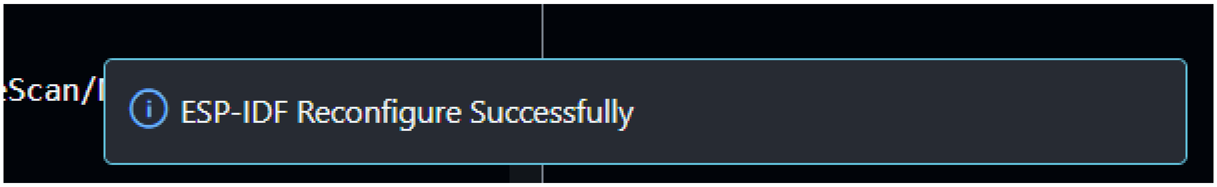
- ☑ ~~Pull camera take picture code from take_picture.c~~



esp32-camera / examples / camera_example / main / take_picture.c



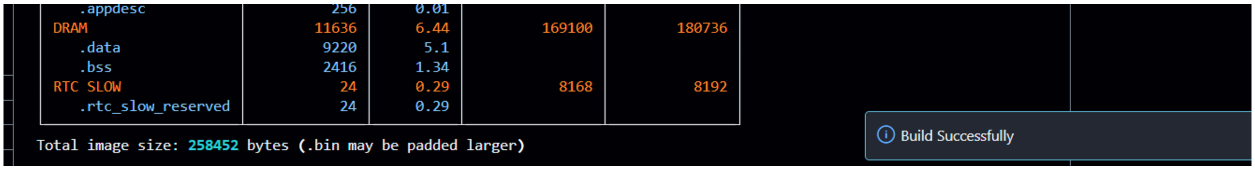
```
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-OV2640\sdmmc>
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-OV2640\sdmmc>
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-OV2640\sdmmc> idf.py add-dependency "espressi
f/esp32-camera"
Executing action: add-dependency
NOTICE: Created "C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-OV2640\sdmmc\main\idf_component
.yml"
NOTICE: Successfully added dependency "espressif/esp32-camera": "*" to component "main"
NOTICE: If you want to make additional changes to the manifest file at path C:/Users/Andrew/.esp/v5.3.2/esp-idf/myprojec
tsESP/NatureScan/NatureScan-OV2640/sdmmc/main/idf_component.yml manually, please refer to the documentation: https://doc
s.espressif.com/projects/idf-component-manager/en/latest/reference/manifest_file.html
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-OV2640\sdmmc>
```



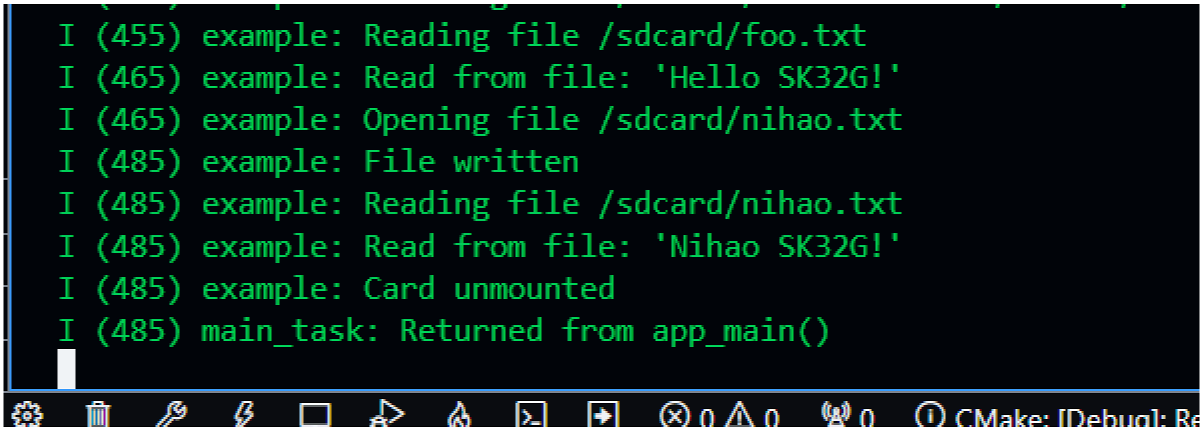
Scan/I

ESP-IDF Reconfigure Successfully

I think i didnt generate the .json config file at the beginning and that fucked something up



```
   .appdesc              256     0.01
DRAM                   11636     6.44        169100        180736
   .data                9220      5.1
   .bss                 2416     1.34
RTC SLOW                  24     0.29          8168          8192
   .rtc_slow_reserved     24     0.29

Total image size: 258452 bytes (.bin may be padded larger)
```

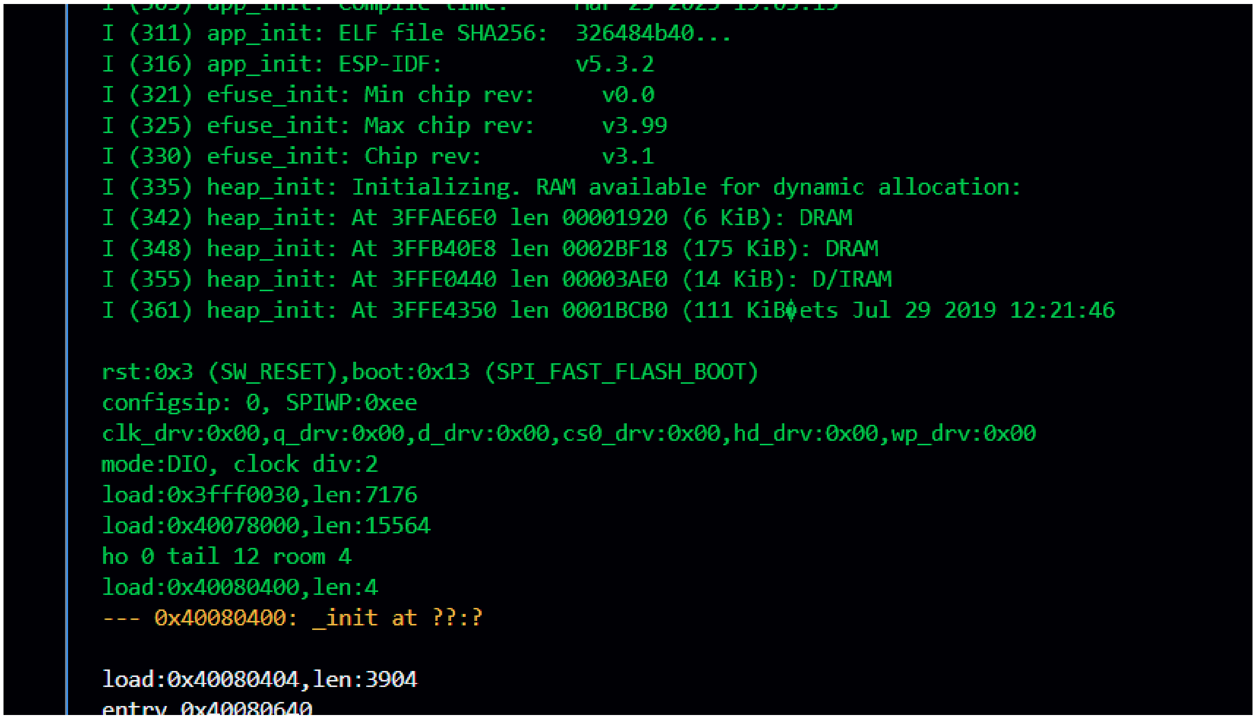


```
I (455) example: Reading file /sdcard/foo.txt
I (465) example: Read from file: 'Hello SK32G!'
I (465) example: Opening file /sdcard/nihao.txt
I (485) example: File written
I (485) example: Reading file /sdcard/nihao.txt
I (485) example: Read from file: 'Nihao SK32G!'
I (485) example: Card unmounted
I (485) main_task: Returned from app_main()
```

It worked with the camera managed components

3/25/2025

SPIFFS *stands for* ***Serial Peripheral Interface Flash File System***

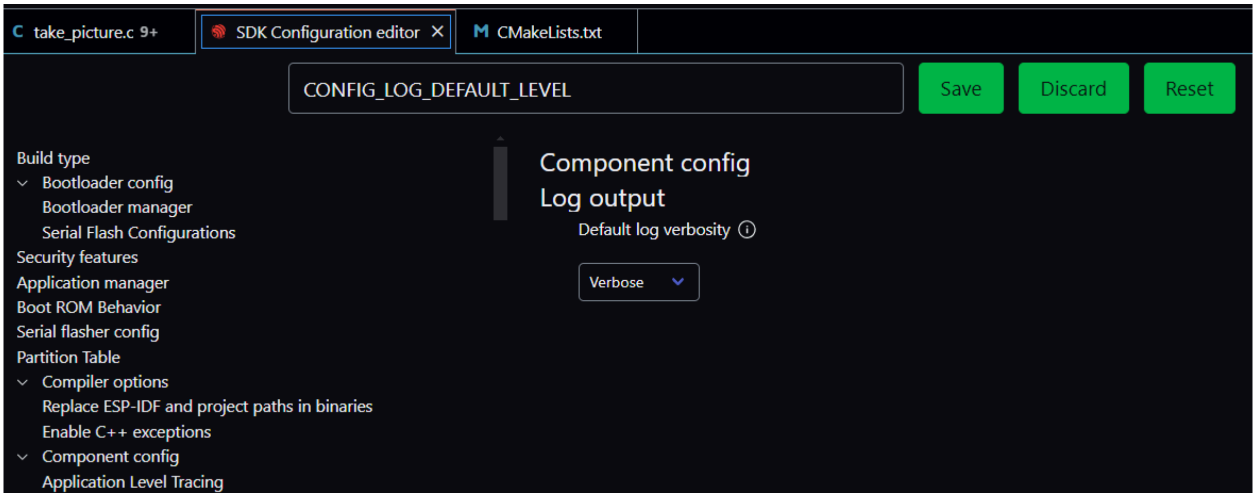


```
I (311) app_init: ELF file SHA256:  326484b40...
I (316) app_init: ESP-IDF:          v5.3.2
I (321) efuse_init: Min chip rev:     v0.0
I (325) efuse_init: Max chip rev:     v3.99
I (330) efuse_init: Chip rev:         v3.1
I (335) heap_init: Initializing. RAM available for dynamic allocation:
I (342) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (348) heap_init: At 3FFB40E8 len 0002BF18 (175 KiB): DRAM
I (355) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (361) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB�ets Jul 29 2019 12:21:46

rst:0x3 (SW_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:7176
load:0x40078000,len:15564
ho 0 tail 12 room 4
load:0x40080400,len:4
--- 0x40080400: _init at ??:?

load:0x40080404,len:3904
entry 0x40080640
```

```
I (335) heap_init: Initializing. RAM available for dynamic allocation:
I (342) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (348) heap_init: At 3FFB40E8 len 0002BF18 (175 KiB): DRAM
I (355) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (361) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB�ets Jul 29 2019 12:21:46
```

I think it resets after heap init
� symbol suggests a display or encoding issue

ROM firmware version: Jul 29 2019 12:21:46

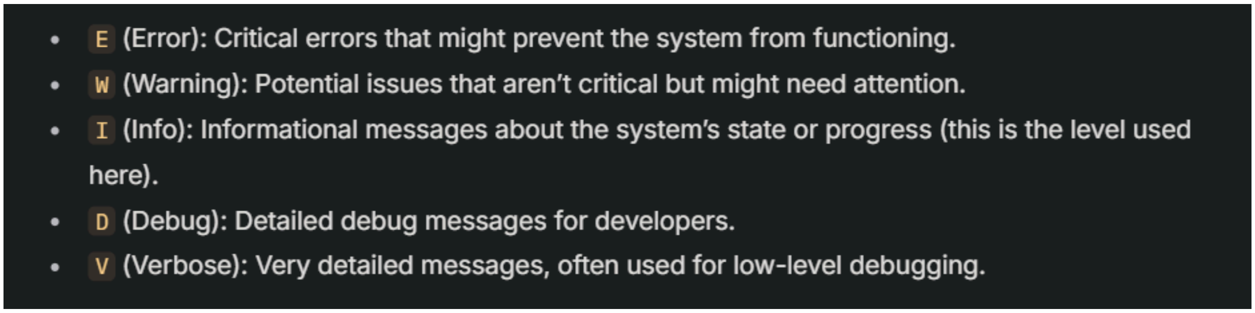


CONFIG_LOG_DEFAULT_LEVEL

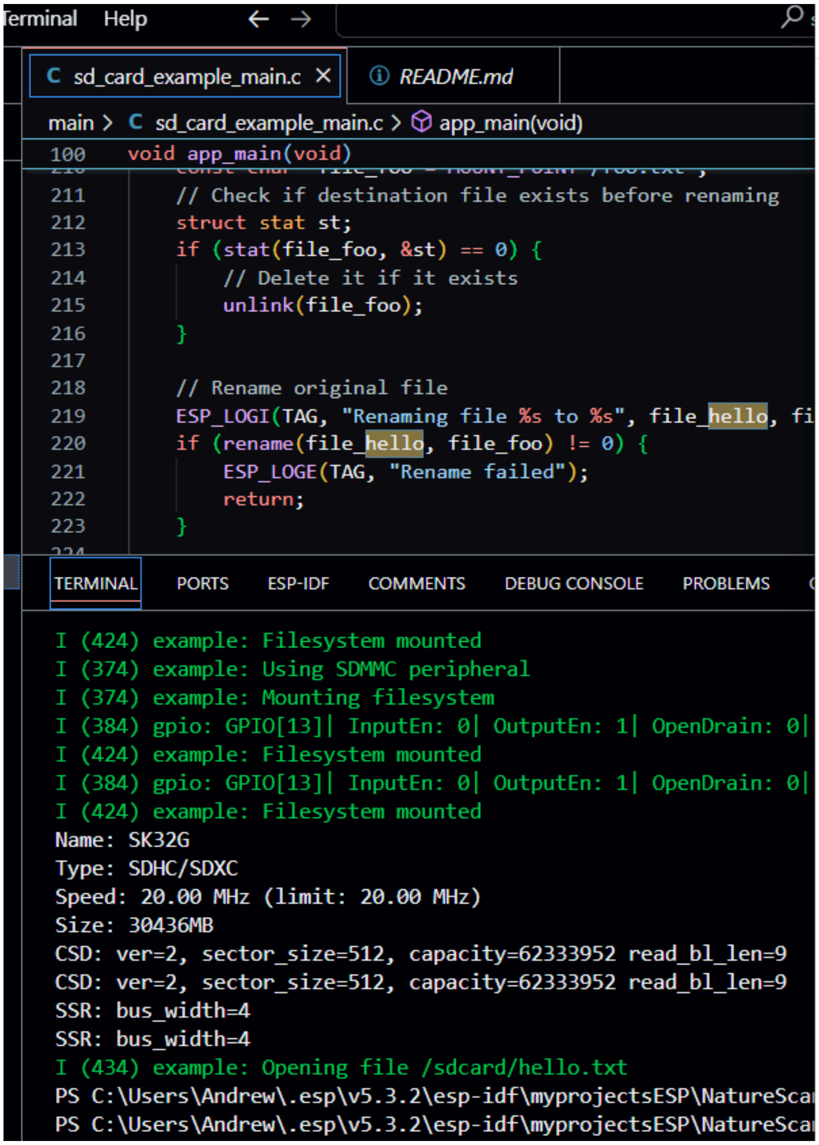
Component config
Log output
Default log verbosity ⓘ
Verbose

To debug
Verbose
Component config log output

Message logging levels:

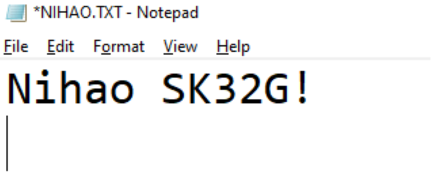


- `E` (Error): Critical errors that might prevent the system from functioning.
- `W` (Warning): Potential issues that aren’t critical but might need attention.
- `I` (Info): Informational messages about the system’s state or progress (this is the level used here).
- `D` (Debug): Detailed debug messages for developers.
- `V` (Verbose): Very detailed messages, often used for low-level debugging.



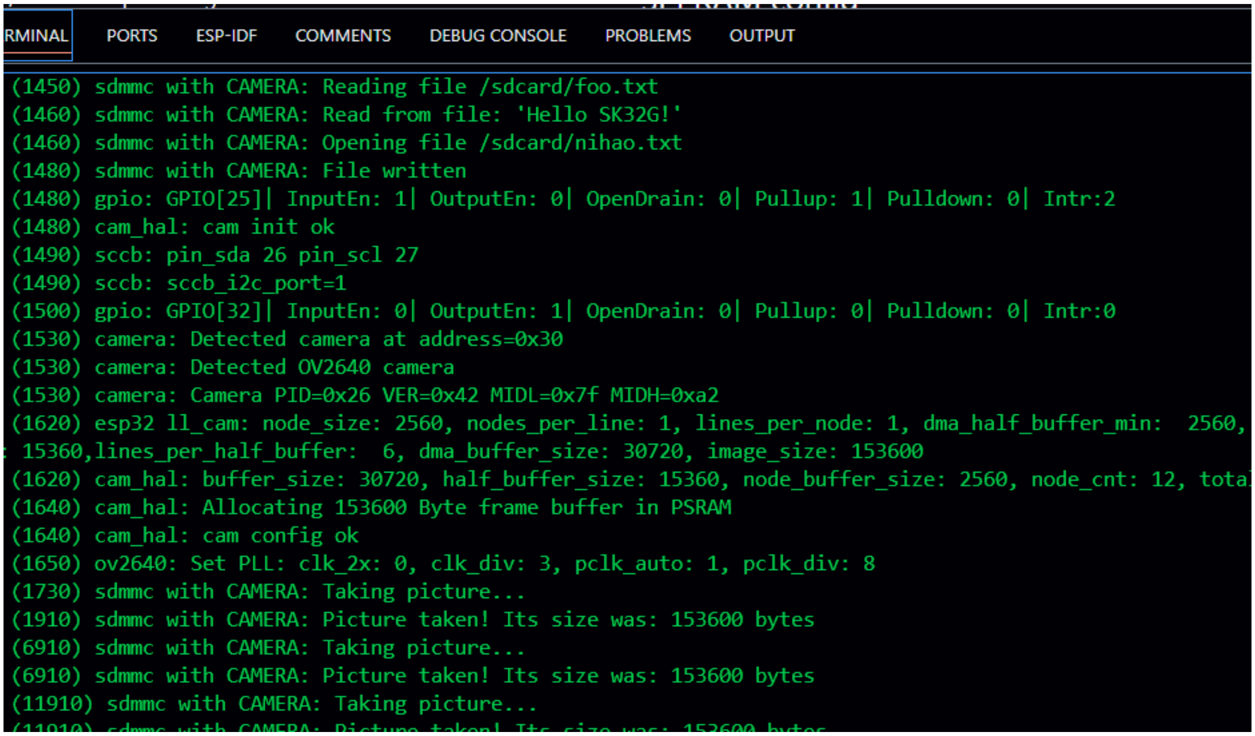
```c
void app_main(void)
    // Check if destination file exists before renaming
    struct stat st;
    if (stat(file_foo, &st) == 0) {
        // Delete it if it exists
        unlink(file_foo);
    }

    // Rename original file
    ESP_LOGI(TAG, "Renaming file %s to %s", file_hello, fi
    if (rename(file_hello, file_foo) != 0) {
        ESP_LOGE(TAG, "Rename failed");
        return;
    }
```

```
I (424) example: Filesystem mounted
I (374) example: Using SDMMC peripheral
I (374) example: Mounting filesystem
I (384) gpio: GPIO[13]| InputEn: 0| OutputEn: 1| OpenDrain: 0|
I (424) example: Filesystem mounted
I (384) gpio: GPIO[13]| InputEn: 0| OutputEn: 1| OpenDrain: 0|
I (424) example: Filesystem mounted
Name: SK32G
Type: SDHC/SDXC
Speed: 20.00 MHz (limit: 20.00 MHz)
Size: 30436MB
CSD: ver=2, sector_size=512, capacity=62333952 read_bl_len=9
CSD: ver=2, sector_size=512, capacity=62333952 read_bl_len=9
SSR: bus_width=4
SSR: bus_width=4
I (434) example: Opening file /sdcard/hello.txt
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureSca
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureSca
```



```
Nihao SK32G!
```

Used the sd card example. Got it to work



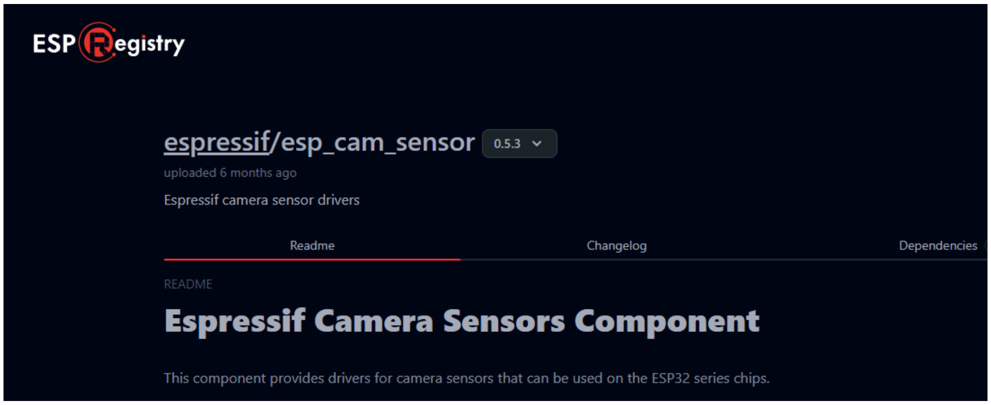
```
(1450) sdmmc with CAMERA: Reading file /sdcard/foo.txt
(1460) sdmmc with CAMERA: Read from file: 'Hello SK32G!'
(1460) sdmmc with CAMERA: Opening file /sdcard/nihao.txt
(1480) sdmmc with CAMERA: File written
(1480) gpio: GPIO[25]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 1| Pulldown: 0| Intr:2
(1480) cam_hal: cam init ok
(1490) sccb: pin_sda 26 pin_scl 27
(1490) sccb: sccb_i2c_port=1
(1500) gpio: GPIO[32]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0
(1530) camera: Detected camera at address=0x30
(1530) camera: Detected OV2640 camera
(1530) camera: Camera PID=0x26 VER=0x42 MIDL=0x7f MIDH=0xa2
(1620) esp32 ll_cam: node_size: 2560, nodes_per_line: 1, lines_per_node: 1, dma_half_buffer_min:  2560,
: 15360,lines_per_half_buffer:  6, dma_buffer_size: 30720, image_size: 153600
(1620) cam_hal: buffer_size: 30720, half_buffer_size: 15360, node_buffer_size: 2560, node_cnt: 12, tota
(1640) cam_hal: Allocating 153600 Byte frame buffer in PSRAM
(1640) cam_hal: cam config ok
(1650) ov2640: Set PLL: clk_2x: 0, clk_div: 3, pclk_auto: 1, pclk_div: 8
(1730) sdmmc with CAMERA: Taking picture...
(1910) sdmmc with CAMERA: Picture taken! Its size was: 153600 bytes
(6910) sdmmc with CAMERA: Taking picture...
(6910) sdmmc with CAMERA: Picture taken! Its size was: 153600 bytes
(11910) sdmmc with CAMERA: Taking picture...
(11910) sdmmc with CAMERA: Picture taken! Its size was: 153600 bytes
```

Saved a file and took a picture!!!!

3/24/2025

Sd card github:
https://github.com/espressif/esp-idf/tree/master/examples/storage/sd_card/sdmmc

SDMMC: Secure Digital MultiMedia Card



espressif/esp_cam_sensor

uploaded 6 months ago

Espressif camera sensor drivers

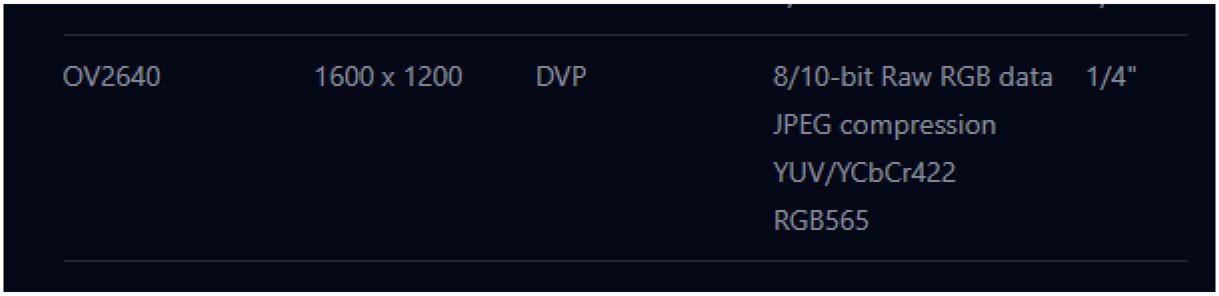
README

Espressif Camera Sensors Component

This component provides drivers for camera sensors that can be used on the ESP32 series chips.

Espressif Camera Sensors Component

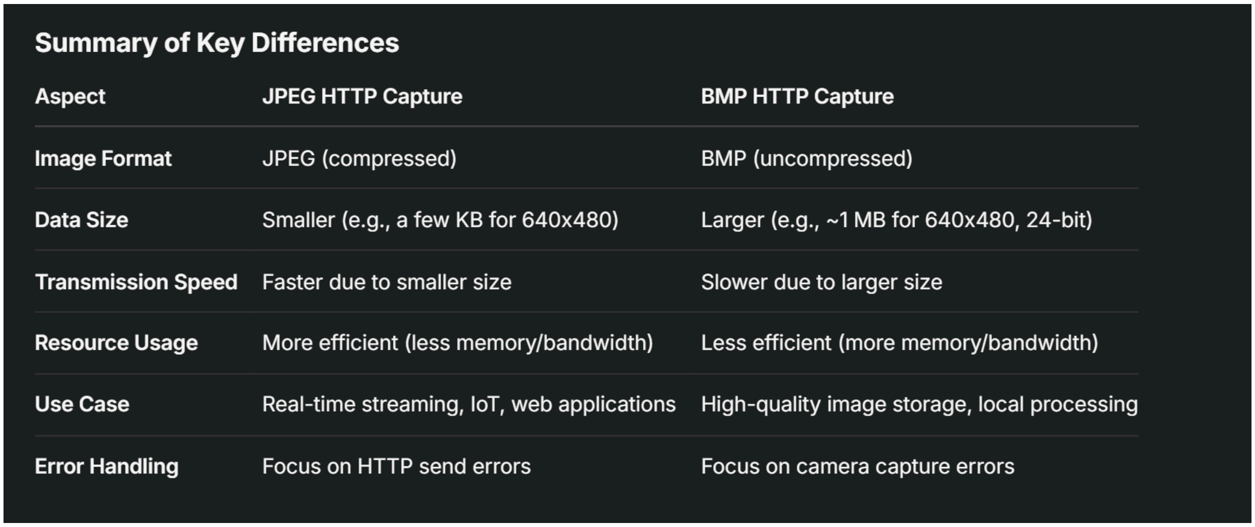
Supports:



OV2640	1600 x 1200	DVP	8/10-bit Raw RGB data JPEG compression YUV/YCbCr422 RGB565	1/4"

OV2640 espressif github Arduino IDE and ESP IDF
https://github.com/espressif/esp32-camera/blob/master/sensors/ov2640.c

Grok
[Deeper research]()

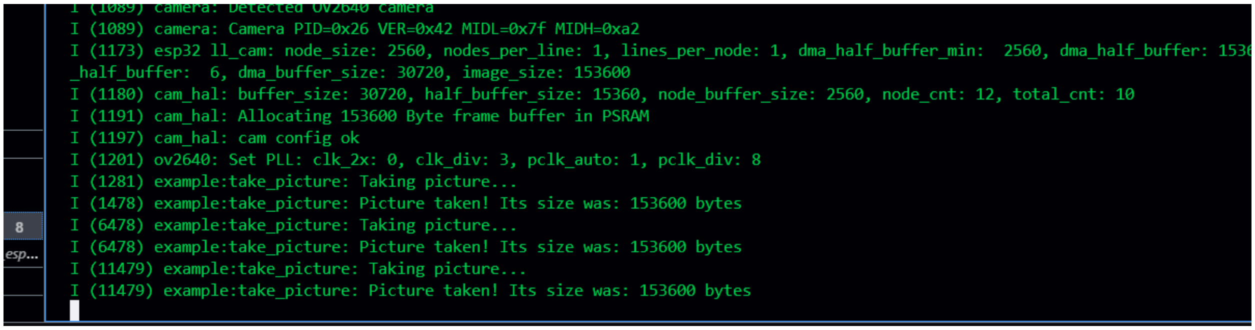


Summary of Key Differences

Aspect	JPEG HTTP Capture	BMP HTTP Capture
Image Format	JPEG (compressed)	BMP (uncompressed)
Data Size	Smaller (e.g., a few KB for 640x480)	Larger (e.g., ~1 MB for 640x480, 24-bit)
Transmission Speed	Faster due to smaller size	Slower due to larger size
Resource Usage	More efficient (less memory/bandwidth)	Less efficient (more memory/bandwidth)
Use Case	Real-time streaming, IoT, web applications	High-quality image storage, local processing
Error Handling	Focus on HTTP send errors	Focus on camera capture errors

writing a BMP image takes longer. For example, writing a 921 KB BMP at 2 MB/s would take about 0.46 seconds per image, while a 50 KB JPEG would take only 0.025 seconds.

JPEG will work just fine..

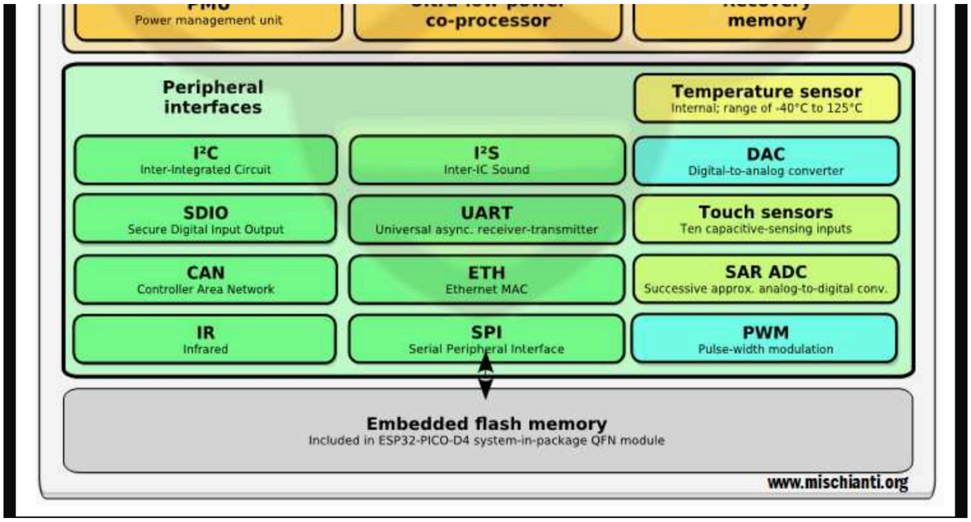


```
I (1089) camera: Detected OV2640 camera
I (1089) camera: Camera PID=0x26 VER=0x42 MIDL=0x7f MIDH=0xa2
I (1173) esp32 ll_cam: node_size: 2560, nodes_per_line: 1, lines_per_node: 1, dma_half_buffer_min:  2560, dma_half_buffer: 1536
_half_buffer:  6, dma_buffer_size: 30720, image_size: 153600
I (1180) cam_hal: buffer_size: 30720, half_buffer_size: 15360, node_buffer_size: 2560, node_cnt: 12, total_cnt: 10
I (1191) cam_hal: Allocating 153600 Byte frame buffer in PSRAM
I (1197) cam_hal: cam config ok
I (1201) ov2640: Set PLL: clk_2x: 0, clk_div: 3, pclk_auto: 1, pclk_div: 8
I (1281) example:take_picture: Taking picture...
I (1478) example:take_picture: Picture taken! Its size was: 153600 bytes
I (6478) example:take_picture: Taking picture...
I (6478) example:take_picture: Picture taken! Its size was: 153600 bytes
I (11479) example:take_picture: Taking picture...
I (11479) example:take_picture: Picture taken! Its size was: 153600 bytes
```

3/20/2025

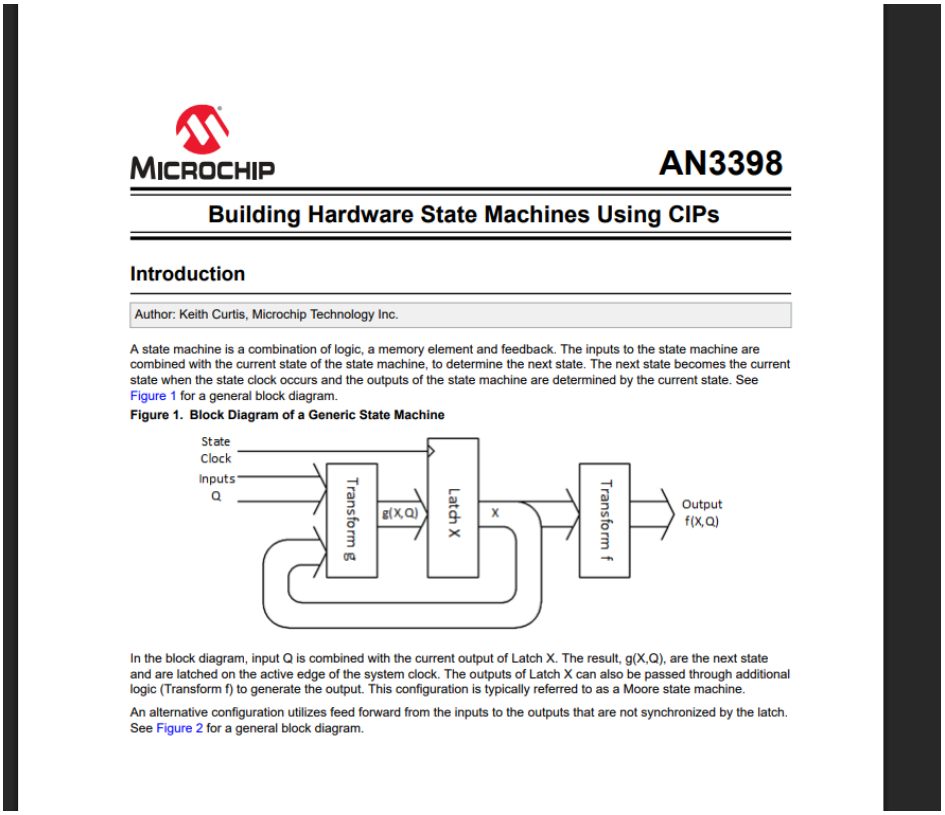
Acronyms for 0v2640

fb: frame buffer
BMP: Bitmap. It's a raster graphics image file format used to store bitmap digital images
DIB: Device Independent Bitmap
VFS: Virtual File System A layer that abstracts file system operations, enabling support for file systems like FAT.
FAT: File Allocation Table A file system format used for storage devices like SD cards.
SDMMC: Secure Digital MultiMedia Card A protocol and controller for interfacing with SD cards or MultiMedia Cards.
SDSPI: Secure Digital Serial Peripheral Interface A method to interface with SD cards using the SPI protocol.
SDIO: secure digital input output
NVS: non volatile storage
POSIX stands for **Portable Operating System Interface**


Power management unit
co-processor
memory
Peripheral interfaces
Temperature sensor
Internal; range of -40°C to 125°C
I²C
Inter-Integrated Circuit
I²S
Inter-IC Sound
DAC
Digital-to-analog converter
SDIO
Secure Digital Input Output
UART
Universal async. receiver-transmitter
Touch sensors
Ten capacitive-sensing inputs
CAN
Controller Area Network
ETH
Ethernet MAC
SAR ADC
Successive approx. analog-to-digital conv.
IR
Infrared
SPI
Serial Peripheral Interface
PWM
Pulse-width modulation
Embedded flash memory
Included in ESP32-PICO-D4 system-in-package QFN module
www.mischianti.org

Senserion sht31 materials:
https://sensirion.com/products/catalog/SHT31-DIS-B

State machines are important.



MICROCHIP AN3398

Building Hardware State Machines Using CIPs

Introduction

Author: Keith Curtis, Microchip Technology Inc.

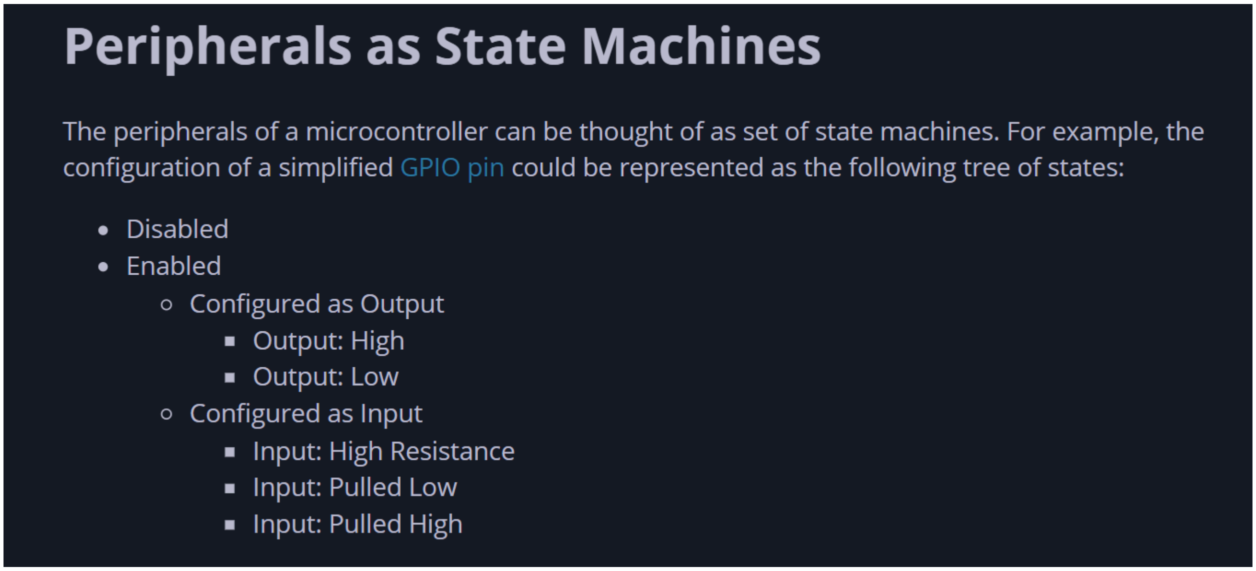
A state machine is a combination of logic, a memory element and feedback. The inputs to the state machine are combined with the current state of the state machine, to determine the next state. The next state becomes the current state when the state clock occurs and the outputs of the state machine are determined by the current state. See Figure 1 for a general block diagram.

Figure 1. Block Diagram of a Generic State Machine

In the block diagram, input Q is combined with the current output of Latch X. The result, g(X,Q), are the next state and are latched on the active edge of the system clock. The outputs of Latch X can also be passed through additional logic (Transform f) to generate the output. This configuration is typically referred to as a Moore state machine.

An alternative configuration utilizes feed forward from the inputs to the outputs that are not synchronized by the latch. See Figure 2 for a general block diagram.

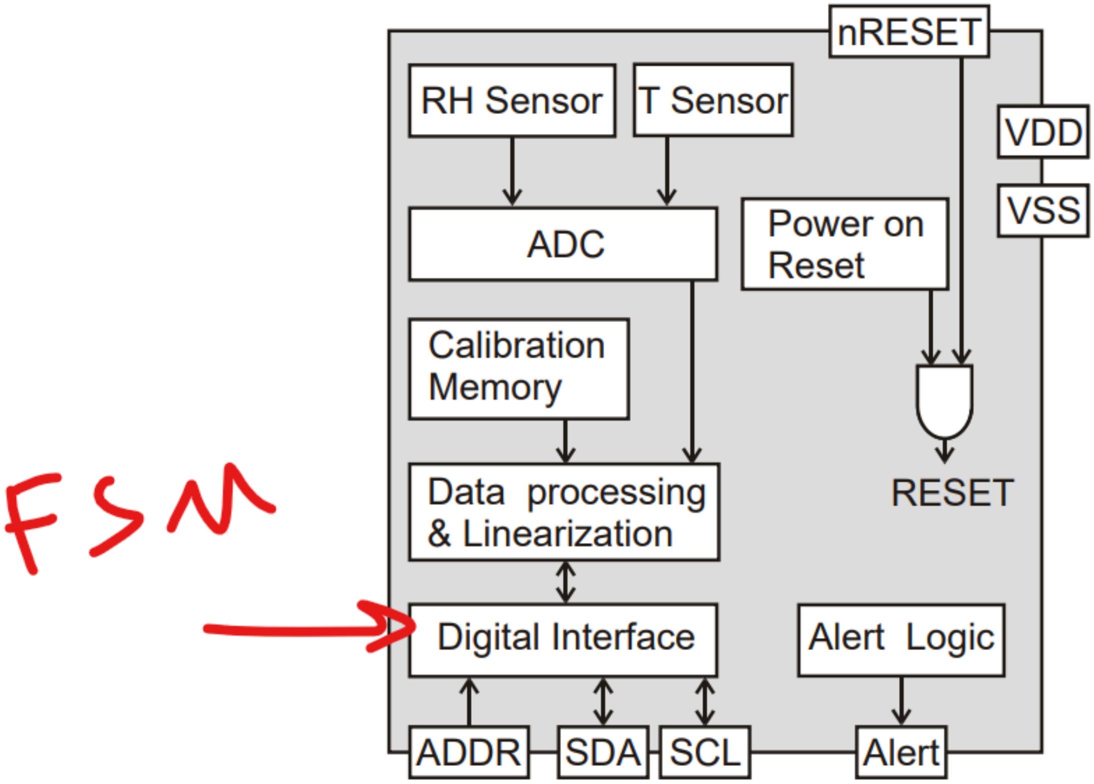
CIP: core independent peripheral



Peripherals as State Machines

The peripherals of a microcontroller can be thought of as set of state machines. For example, the configuration of a simplified GPIO pin could be represented as the following tree of states:

- Disabled
- Enabled
    - Configured as Output
        - Output: High
        - Output: Low
    - Configured as Input
        - Input: High Resistance
        - Input: Pulled Low
        - Input: Pulled High



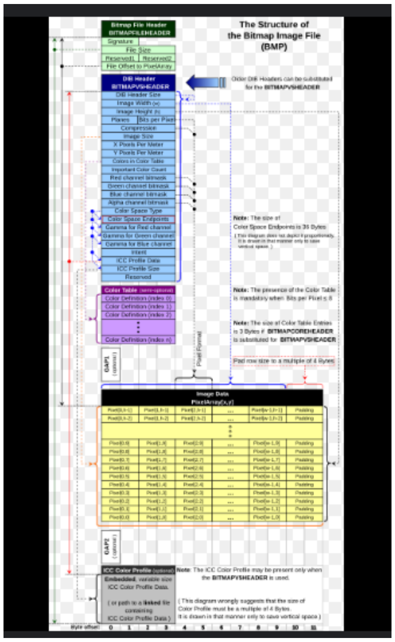
FSM

Configuration Registers: You interact with a peripheral by reading from and writing to its *control and status registers*. These registers are memory-mapped, meaning they appear at specific addresses in the CPU's memory space.

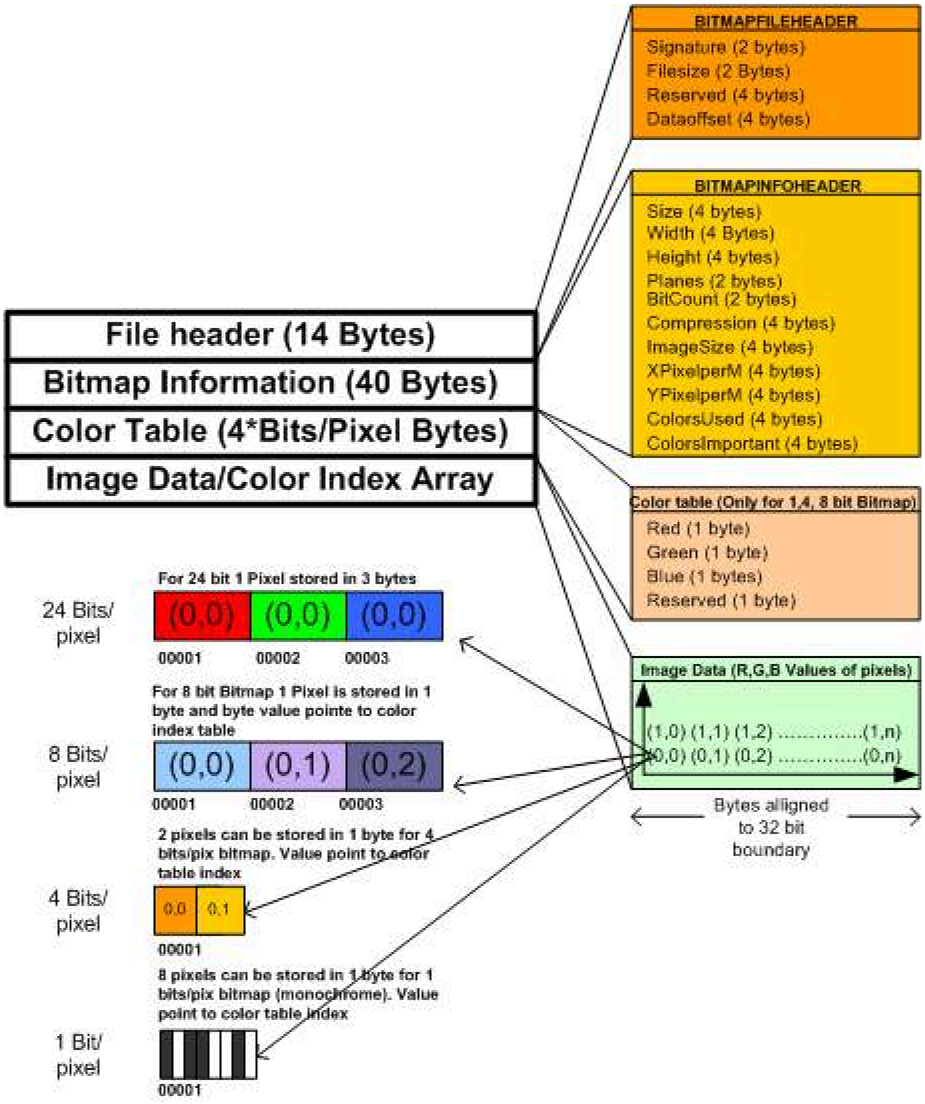
- **Control Registers:** Writing to control registers changes the peripheral's configuration and behavior. This is how you "tell" the peripheral what to do (e.g., set a GPIO pin as an output, set the baud rate of a UART, start a timer). This is analogous to changing the *state* of the peripheral's state machine.
- **Status Registers:** Reading from status registers tells you the current state of the peripheral (e.g., whether a GPIO pin is high or low, whether a UART has received data, whether a timer has overflowed).

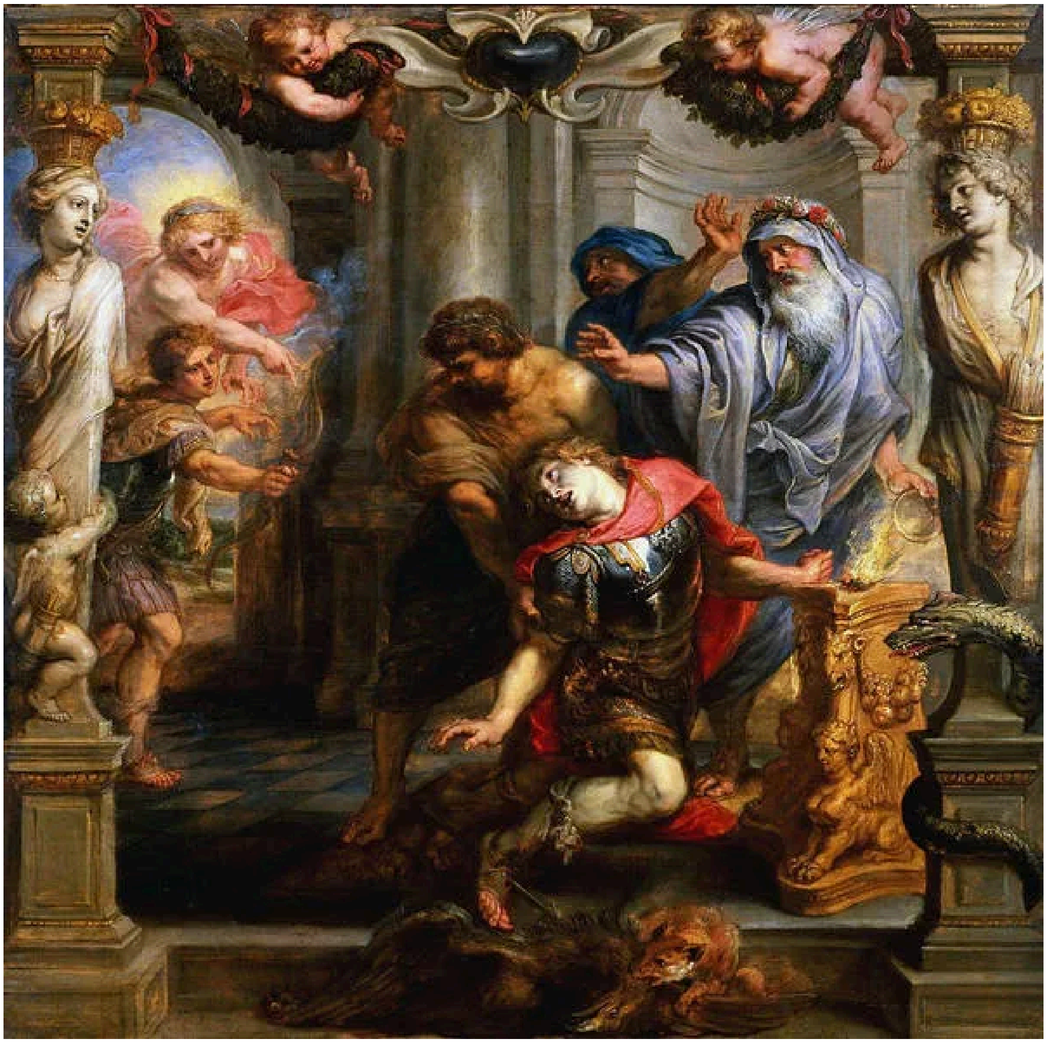
3/20/2025

fb: frame buffer
BMP stands for Bitmap. It's a raster graphics image file format used to store bitmap digital images
DIB (Device Independent Bitmap)

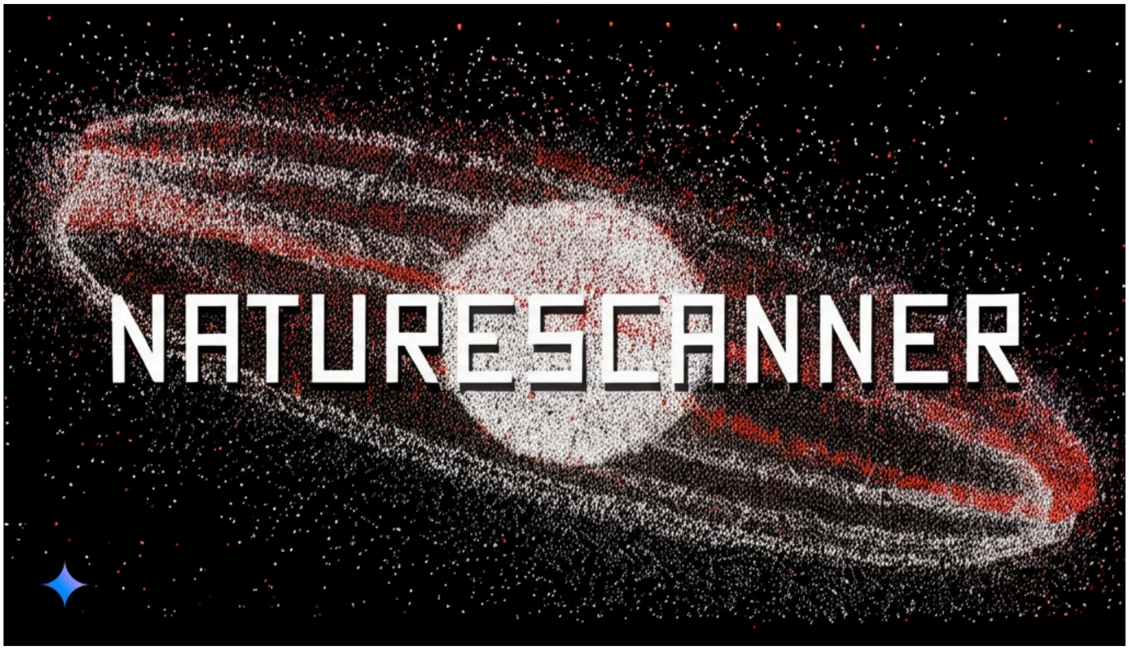


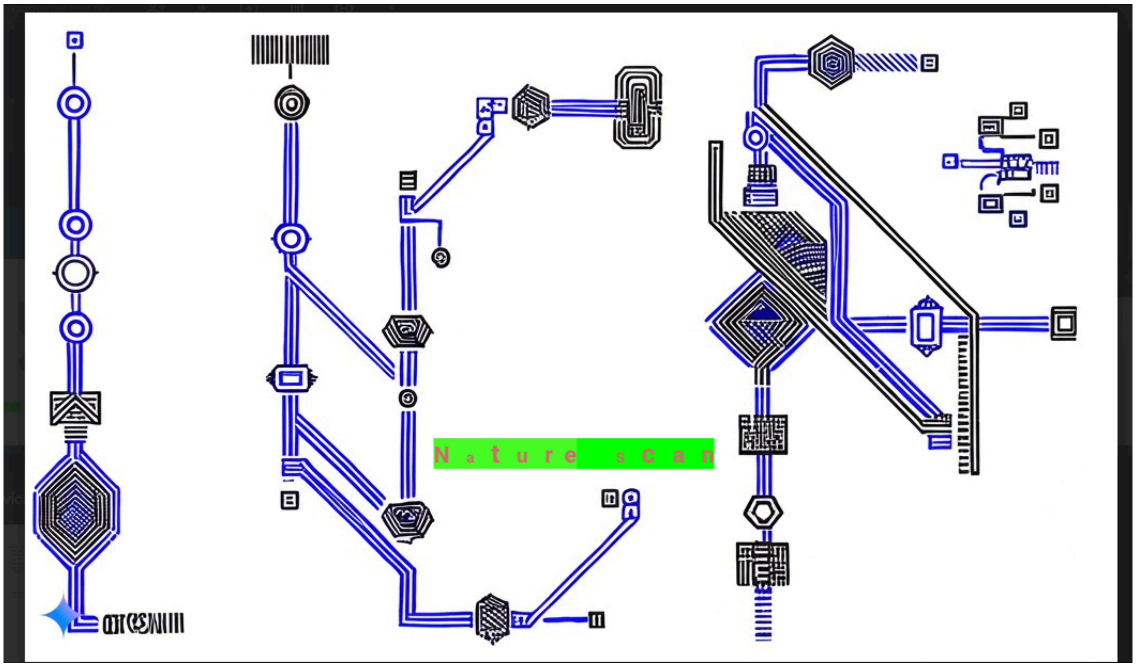
The structure of a BMP bitmap image file

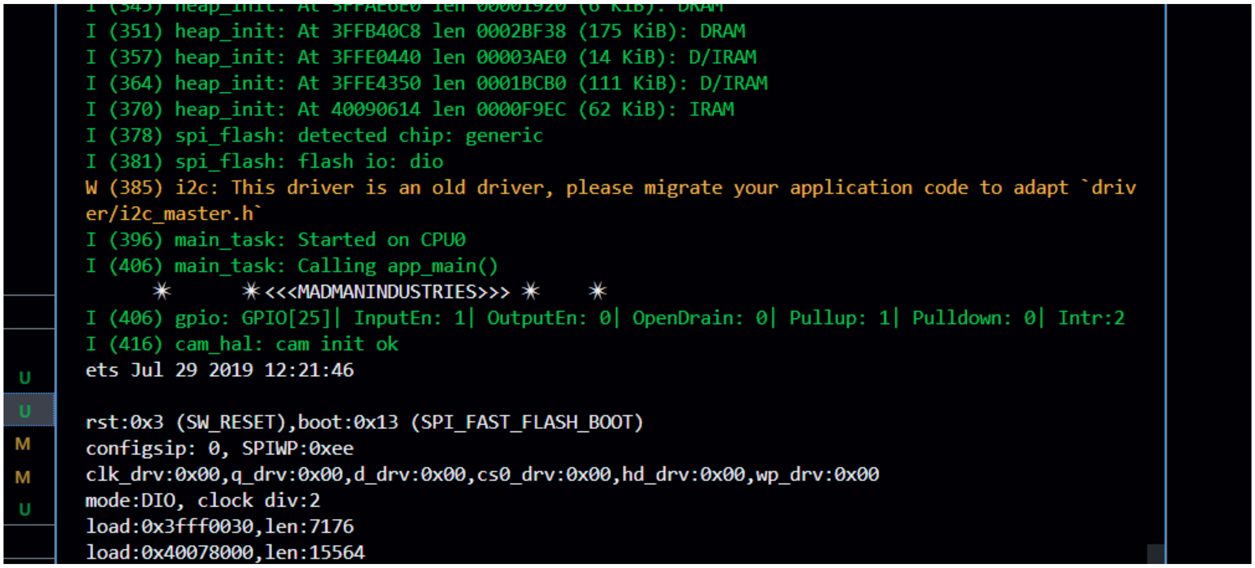

BITMAPFILEHEADER
Signature (2 bytes)
Filesize (2 Bytes)
Reserved (4 bytes)
Dataoffset (4 bytes)
BITMAPINFOHEADER
Size (4 bytes)
Width (4 Bytes)
Height (4 bytes)
Planes (2 bytes)
BitCount (2 bytes)
Compression (4 bytes)
ImageSize (4 bytes)
XPixelperM (4 bytes)
YPixelperM (4 bytes)
ColorsUsed (4 bytes)
ColorsImportant (4 bytes)
File header (14 Bytes)
Bitmap Information (40 Bytes)
Color Table (4*Bits/Pixel Bytes)
Image Data/Color Index Array
Color table (Only for 1,4, 8 bit Bitmap)
Red (1 byte)
Green (1 byte)
Blue (1 bytes)
Reserved (1 byte)
For 24 bit 1 Pixel stored in 3 bytes
24 Bits/ pixel
(0,0)
(0,0)
(0,0)
00001
00002
00003
Image Data (R,G,B Values of pixels)
(1,0) (1,1) (1,2) ....(1,n)
(0,0) (0,1) (0,2) ....(0,n)
Bytes alligned to 32 bit boundary
For 8 bit Bitmap 1 Pixel is stored in 1 byte and byte value pointe to color index table
8 Bits/ pixel
(0,0)
(0,1)
(0,2)
00001
00002
00003
2 pixels can be stored in 1 byte for 4 bits/pix bitmap. Value point to color table index
4 Bits/ pixel
0,0
0,1
00001
8 pixels can be stored in 1 byte for 1 bits/pix bitmap (monochrome). Value point to color table index
1 Bit/ pixel
00001



Death of Achilles

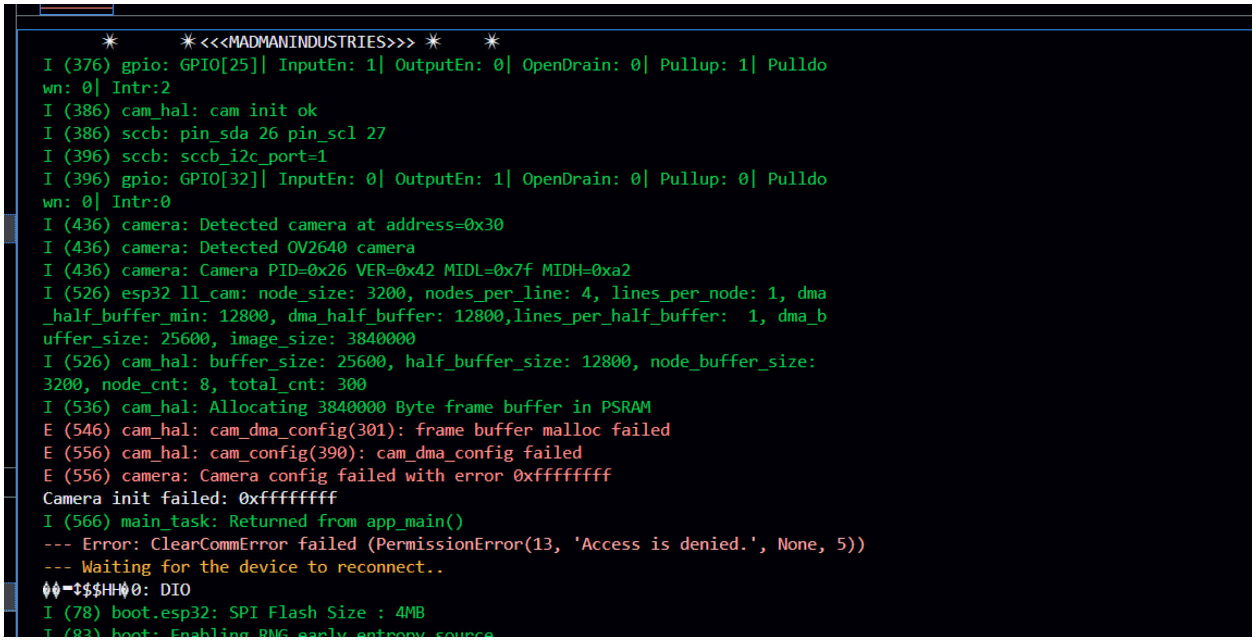

NATURESCANNER


Nature scan



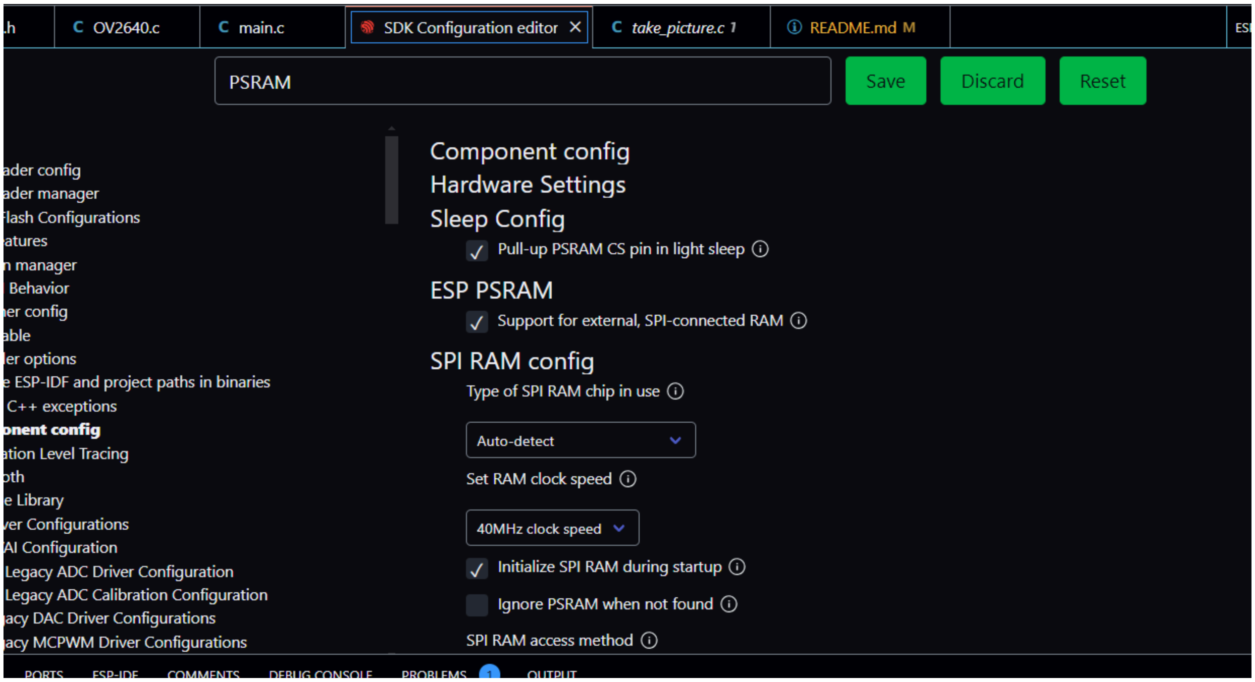
```
I (345) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (351) heap_init: At 3FFB40C8 len 0002BF38 (175 KiB): DRAM
I (357) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (364) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (370) heap_init: At 40090614 len 0000F9EC (62 KiB): IRAM
I (378) spi_flash: detected chip: generic
I (381) spi_flash: flash io: dio
W (385) i2c: This driver is an old driver, please migrate your application code to adapt `driv
er/i2c_master.h`
I (396) main_task: Started on CPU0
I (406) main_task: Calling app_main()
      ✷       ✷<<<MADMANINDUSTRIES>>> ✷    ✷
I (406) gpio: GPIO[25]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 1| Pulldown: 0| Intr:2
I (416) cam_hal: cam init ok
ets Jul 29 2019 12:21:46

rst:0x3 (SW_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:7176
load:0x40078000,len:15564
```

New error



```
      ✷       ✷<<<MADMANINDUSTRIES>>> ✷    ✷
I (376) gpio: GPIO[25]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 1| Pulldo
wn: 0| Intr:2
I (386) cam_hal: cam init ok
I (386) sccb: pin_sda 26 pin_scl 27
I (396) sccb: sccb_i2c_port=1
I (396) gpio: GPIO[32]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldo
wn: 0| Intr:0
I (436) camera: Detected camera at address=0x30
I (436) camera: Detected OV2640 camera
I (436) camera: Camera PID=0x26 VER=0x42 MIDL=0x7f MIDH=0xa2
I (526) esp32 ll_cam: node_size: 3200, nodes_per_line: 4, lines_per_node: 1, dma
_half_buffer_min: 12800, dma_half_buffer: 12800,lines_per_half_buffer:  1, dma_b
uffer_size: 25600, image_size: 3840000
I (526) cam_hal: buffer_size: 25600, half_buffer_size: 12800, node_buffer_size:
3200, node_cnt: 8, total_cnt: 300
I (536) cam_hal: Allocating 3840000 Byte frame buffer in PSRAM
E (546) cam_hal: cam_dma_config(301): frame buffer malloc failed
E (556) cam_hal: cam_config(390): cam_dma_config failed
E (556) camera: Camera config failed with error 0xffffffff
Camera init failed: 0xffffffff
I (566) main_task: Returned from app_main()
--- Error: ClearCommError failed (PermissionError(13, 'Access is denied.', None, 5))
--- Waiting for the device to reconnect..
��━↕$$HH�0: DIO
I (78) boot.esp32: SPI Flash Size : 4MB
I (83) boot: Enabling RNG early entropy source
```

Detected camera



Component config
Hardware Settings
Sleep Config
☑ Pull-up PSRAM CS pin in light sleep
ESP PSRAM
☑ Support for external, SPI-connected RAM
SPI RAM config
Type of SPI RAM chip in use
Auto-detect
Set RAM clock speed
40MHz clock speed
☑ Initialize SPI RAM during startup
☐ Ignore PSRAM when not found
SPI RAM access method

Enabled psram

3/17/2025

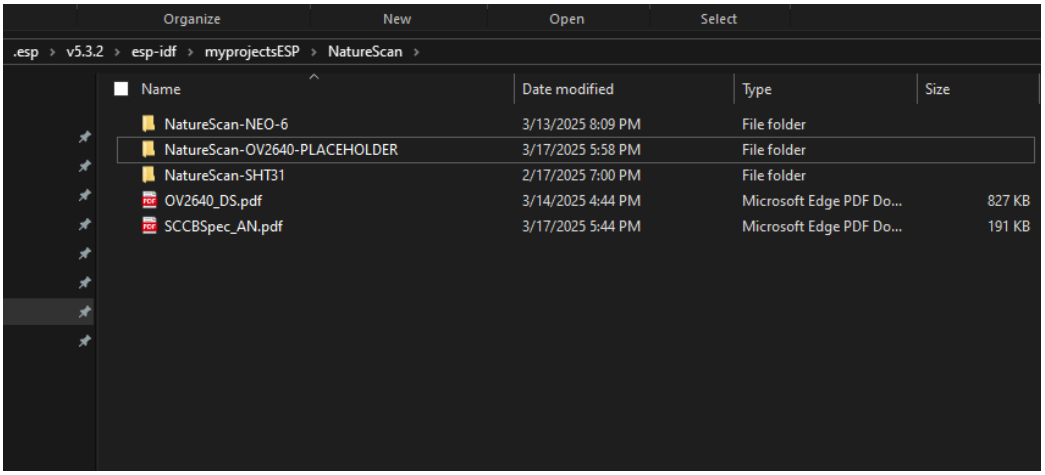
Next steps:

- ☑ ~~Read camera documentation~~
- ☑ ~~Read esp idf lib documentation spi master~~
- ☑ ~~Create a new blank project~~
- ☑ ~~Upload project to github~~

OV stands for **OmniVision**
UXGA stands for **Ultra Extended Graphics Array**
Serial Camera Control Bus (SCCB) interface.
Automatic Exposure Control (AEC),
Automatic Gain Control (AGC),
Automatic White Balance (AWB),
Automatic Band Filter (ABF), and
Automatic Black-Level Calibration (ABLC)

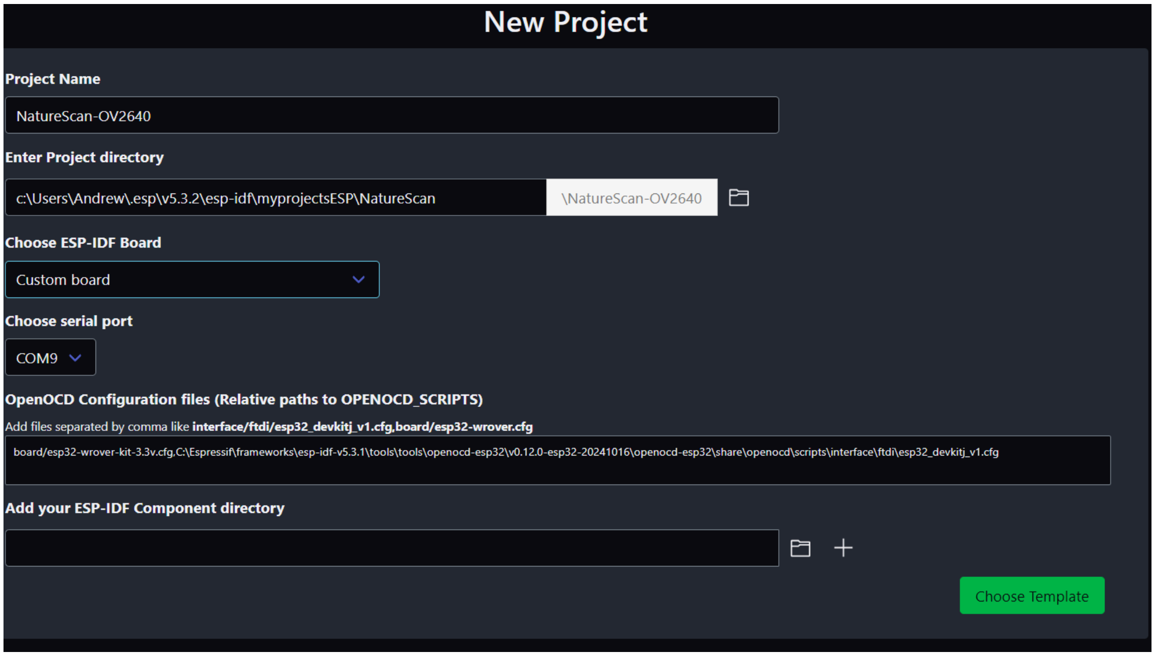
CSP2 in the context of the OV02640 camera sensor refers to **Chip Scale Package 2**

RESETB: **Reset Bar**. This is the active-low reset signal for the OV2640. When this pin is driven low, the sensor resets to its initial state.
PWDN: **Power Down**. This pin controls the power-down mode of the OV2640. When asserted (typically high), the sensor enters a low-power state, shutting down most of its functions to save energy.
XVCLK: **External Video Clock**. This is the external clock input provided to the OV2640, used to drive its internal timing. The OV2640 uses this clock to synchronize its operations, such as pixel data output.
MHSYNC: **Master Horizontal Sync**. This is an input signal from the master device to the OV2640, indicating the start of a new horizontal line (row) of pixel data in slave mode.
MVSYNC: **Master Vertical Sync**. This is an input signal from the master device to the OV2640, indicating the start of a new frame (image) in slave mode.
MCLK: **Master Clock**. This is the clock signal provided by the master device to the OV2640 in slave mode, used to synchronize the output of pixel data.



Name	Date modified	Type	Size
NatureScan-NEO-6	3/13/2025 8:09 PM	File folder	
NatureScan-OV2640-PLACEHOLDER	3/17/2025 5:58 PM	File folder	
NatureScan-SHT31	2/17/2025 7:00 PM	File folder	
OV2640_DS.pdf	3/14/2025 4:44 PM	Microsoft Edge PDF Do...	827 KB
SCCBSpec_AN.pdf	3/17/2025 5:44 PM	Microsoft Edge PDF Do...	191 KB

NatureScan-OV2640



Project Name

NatureScan-OV2640

Enter Project directory

c:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan \NatureScan-OV2640

Choose ESP-IDF Board

Custom board

Choose serial port

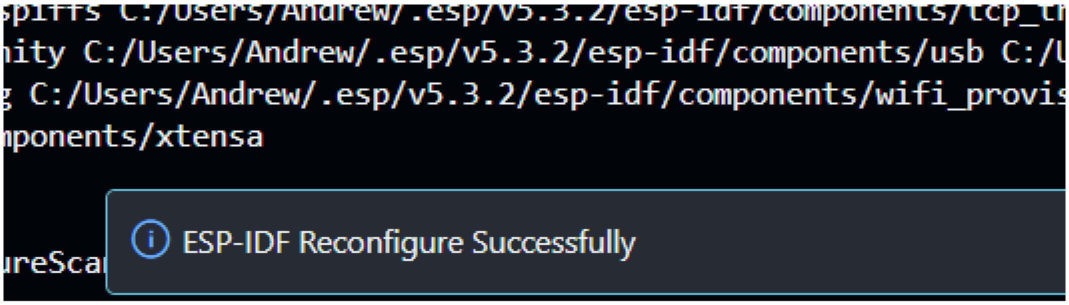
COM9

OpenOCD Configuration files (Relative paths to OPENOCD_SCRIPTS)

Add files separated by comma like **interface/ftdi/esp32_devkitj_v1.cfg,board/esp32-wrover.cfg**

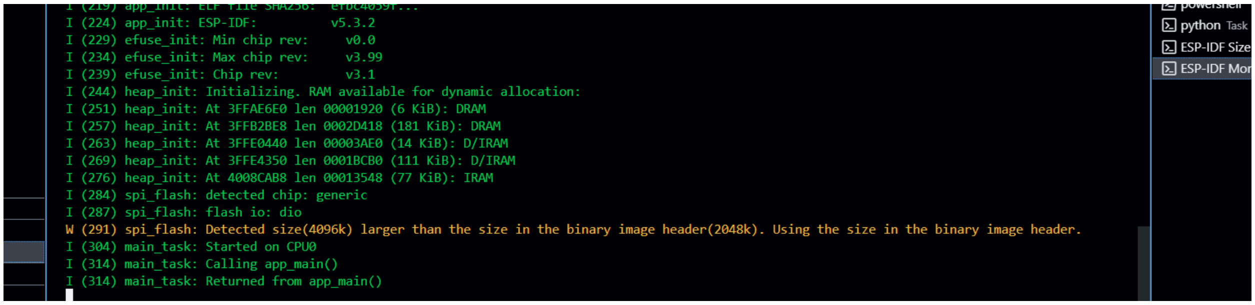
board/esp32-wrover-kit-3.3v.cfg,C:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\openocd-esp32\v0.12.0-esp32-20241016\openocd-esp32\share\openocd\scripts\interface\ftdi\esp32_devkitj_v1.cfg

Add your ESP-IDF Component directory

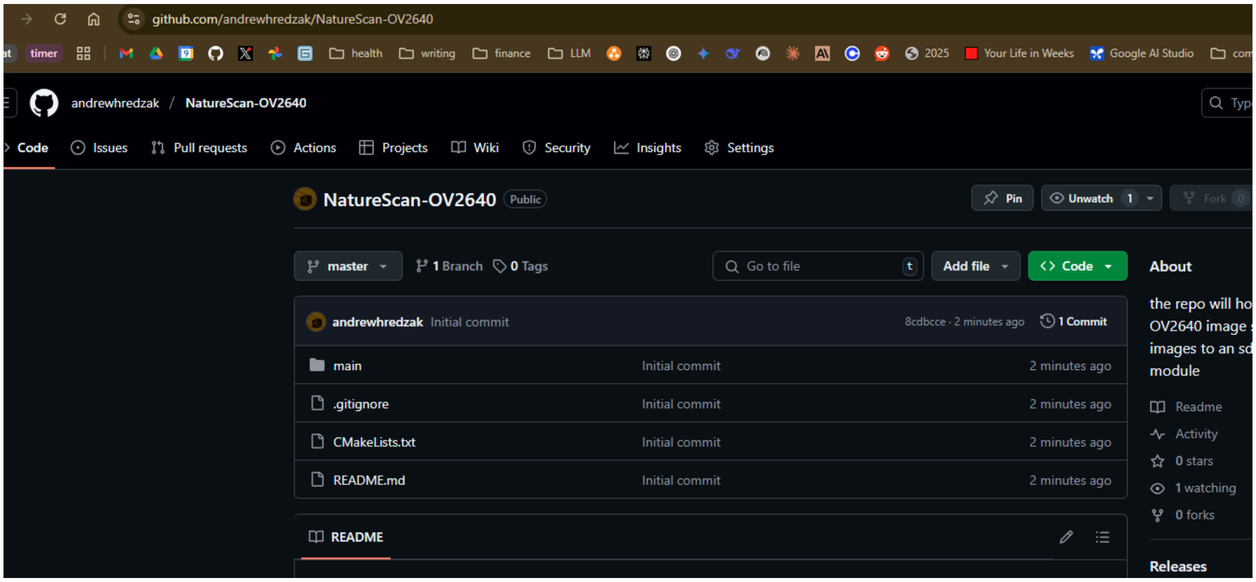


```
spiffs C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/tcp_tr
nity C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/usb C:/U
g C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/wifi_provis
mponents/xtensa

ureScan
```

ESP-IDF Reconfigure Successfully



```
I (219) app_init: ELF file SHA256:  efbc4059f...
I (224) app_init: ESP-IDF:          v5.3.2
I (229) efuse_init: Min chip rev:     v0.0
I (234) efuse_init: Max chip rev:     v3.99
I (239) efuse_init: Chip rev:         v3.1
I (244) heap_init: Initializing. RAM available for dynamic allocation:
I (251) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (257) heap_init: At 3FFB2BE8 len 0002D418 (181 KiB): DRAM
I (263) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (269) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (276) heap_init: At 4008CAB8 len 00013548 (77 KiB): IRAM
I (284) spi_flash: detected chip: generic
I (287) spi_flash: flash io: dio
W (291) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
I (304) main_task: Started on CPU0
I (314) main_task: Calling app_main()
I (314) main_task: Returned from app_main()
```



NatureScan-OV2640

andrewhredzak Initial commit 8cdbcce · 2 minutes ago 1 Commit

main	Initial commit	2 minutes ago
.gitignore	Initial commit	2 minutes ago
CMakeLists.txt	Initial commit	2 minutes ago
README.md	Initial commit	2 minutes ago

About

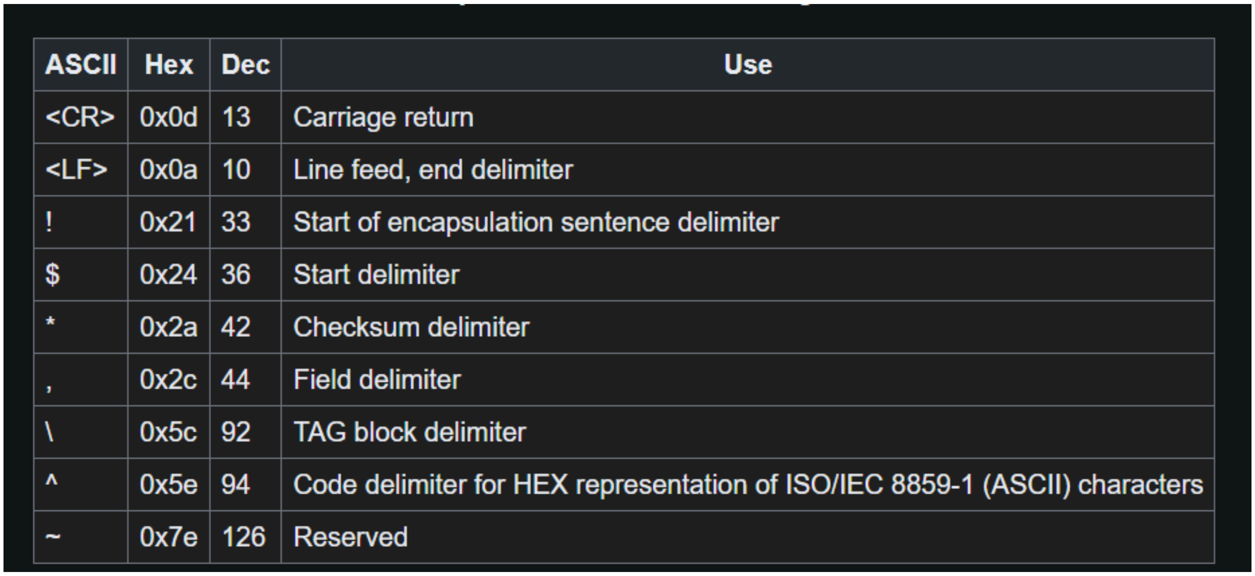
the repo will ho
OV2640 image
images to an sd
module

Git created
Initial code ran!!!

3/14/2025

3/13/2025

NMEA 0183 sentences



ASCII	Hex	Dec	Use
<CR>	0x0d	13	Carriage return
<LF>	0x0a	10	Line feed, end delimiter
!	0x21	33	Start of encapsulation sentence delimiter
$	0x24	36	Start delimiter
*	0x2a	42	Checksum delimiter
,	0x2c	44	Field delimiter
\	0x5c	92	TAG block delimiter
^	0x5e	94	Code delimiter for HEX representation of ISO/IEC 8859-1 (ASCII) characters
~	0x7e	126	Reserved

NMEA sentences are ASCII strings, where each character (including commas, dollar signs, and the asterisk) is represented by 1 byte in memory.

$GPRMC,030742.00,A,2232.7830,N,11404.58520,E,0.356,,070314,,A*77

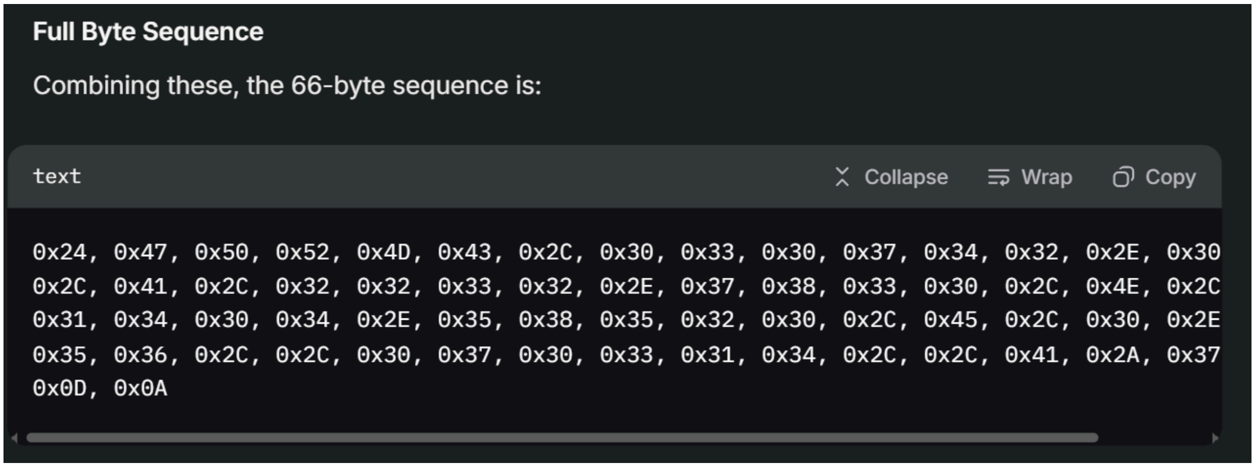
About 66 byes long including carriage characters /n

ASCII Values

Here’s the breakdown of the sentence into ASCII values (in hexadecimal for clarity):

- $GPRMC → 0x24, 0x47, 0x50, 0x52, 0x4D, 0x43
- , → 0x2C
- 030742.00 → 0x30, 0x33, 0x30, 0x37, 0x34, 0x32, 0x2E, 0x30, 0x30
- , → 0x2C
- A → 0x41
- , → 0x2C
- 2232.7830 → 0x32, 0x32, 0x33, 0x32, 0x2E, 0x37, 0x38, 0x33, 0x30
- , → 0x2C
- N → 0x4E
- , → 0x2C
- 11404.58520 → 0x31, 0x31, 0x34, 0x30, 0x34, 0x2E, 0x35, 0x38, 0x35, 0x32, 0x30
- , → 0x2C
- E → 0x45
- , → 0x2C

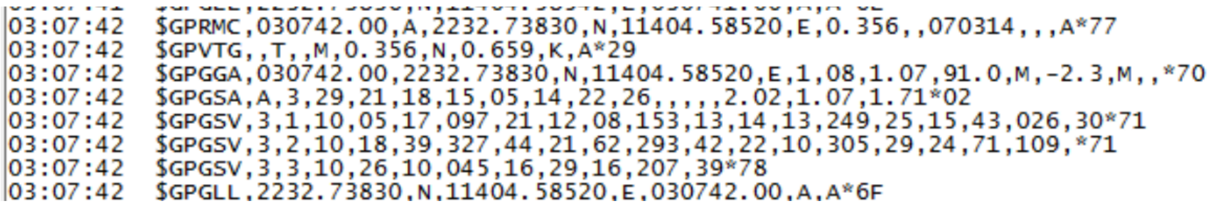
- 0.356 → 0x30, 0x2E, 0x33, 0x35, 0x36
- , → 0x2C
- , → 0x2C (empty field)
- 070314 → 0x30, 0x37, 0x30, 0x33, 0x31, 0x34
- , → 0x2C
- , → 0x2C (empty field)
- A → 0x41
- *77 → 0x2A, 0x37, 0x37
- \r\n → 0x0D, 0x0A



Full Byte Sequence

Combining these, the 66-byte sequence is:

```
0x24, 0x47, 0x50, 0x52, 0x4D, 0x43, 0x2C, 0x30, 0x33, 0x30, 0x37, 0x34, 0x32, 0x2E, 0x30
0x2C, 0x41, 0x2C, 0x32, 0x32, 0x33, 0x32, 0x2E, 0x37, 0x38, 0x33, 0x30, 0x2C, 0x4E, 0x2C
0x31, 0x34, 0x30, 0x34, 0x2E, 0x35, 0x38, 0x35, 0x32, 0x30, 0x2C, 0x45, 0x2C, 0x30, 0x2E
0x35, 0x36, 0x2C, 0x2C, 0x30, 0x37, 0x30, 0x33, 0x31, 0x34, 0x2C, 0x2C, 0x41, 0x2A, 0x37
0x0D, 0x0A
```

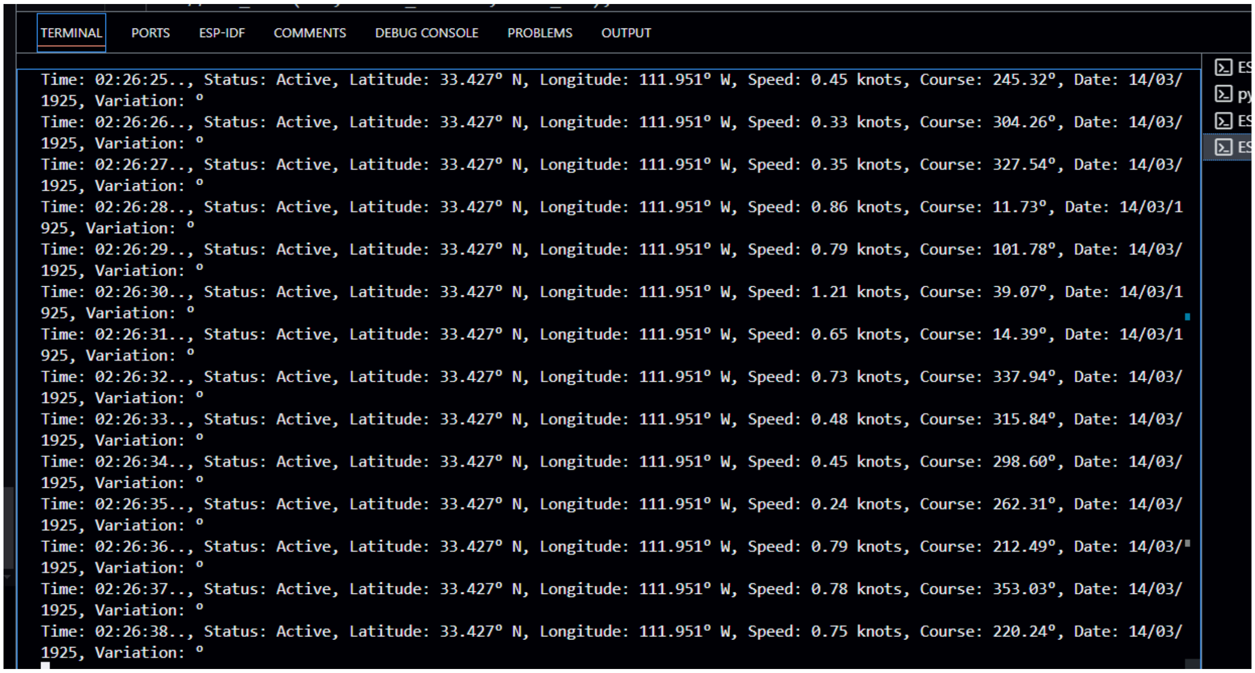


```
03:07:42   $GPRMC,030742.00,A,2232.73830,N,11404.58520,E,0.356,,070314,,,A*77
03:07:42   $GPVTG,,T,,M,0.356,N,0.659,K,A*29
03:07:42   $GPGGA,030742.00,2232.73830,N,11404.58520,E,1,08,1.07,91.0,M,-2.3,M,,*70
03:07:42   $GPGSA,A,3,29,21,18,15,05,14,22,26,,,,,2.02,1.07,1.71*02
03:07:42   $GPGSV,3,1,10,05,17,097,21,12,08,153,13,14,13,249,25,15,43,026,30*71
03:07:42   $GPGSV,3,2,10,18,39,327,44,21,62,293,42,22,10,305,29,24,71,109,*71
03:07:42   $GPGSV,3,3,10,26,10,045,16,29,16,207,39*78
03:07:42   $GPGLL,2232.73830,N,11404.58520,E,030742.00,A,A*6F
```

The number of bytes depends on the NMEA sentence:

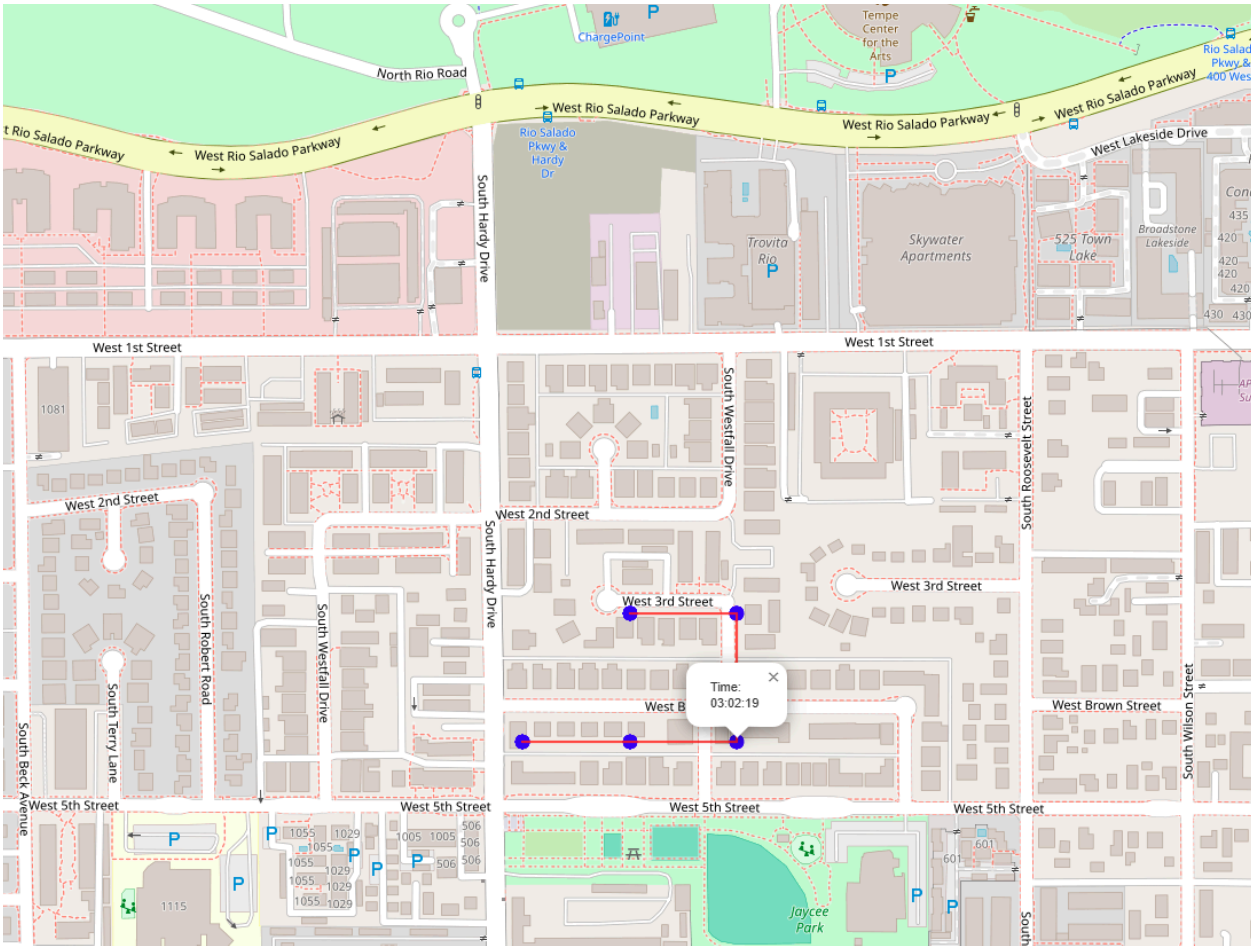
- $GPRMC: 66 bytes (including \r\n).
- $GPVTG: 35 bytes.
- $GPGGA: 69 bytes.
- $GPGSA: 47 bytes.
- $GPGSV: 54 bytes.
- $GPGLL: 42 bytes.

UART parse thread link



```
Time: 02:26:25.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.45 knots, Course: 245.32°, Date: 14/03/
1925, Variation: °
Time: 02:26:26.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.33 knots, Course: 304.26°, Date: 14/03/
1925, Variation: °
Time: 02:26:27.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.35 knots, Course: 327.54°, Date: 14/03/
1925, Variation: °
Time: 02:26:28.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.86 knots, Course: 11.73°, Date: 14/03/1
925, Variation: °
Time: 02:26:29.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.79 knots, Course: 101.78°, Date: 14/03/
1925, Variation: °
Time: 02:26:30.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 1.21 knots, Course: 39.07°, Date: 14/03/1
925, Variation: °
Time: 02:26:31.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.65 knots, Course: 14.39°, Date: 14/03/1
925, Variation: °
Time: 02:26:32.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.73 knots, Course: 337.94°, Date: 14/03/
1925, Variation: °
Time: 02:26:33.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.48 knots, Course: 315.84°, Date: 14/03/
1925, Variation: °
Time: 02:26:34.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.45 knots, Course: 298.60°, Date: 14/03/
1925, Variation: °
Time: 02:26:35.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.24 knots, Course: 262.31°, Date: 14/03/
1925, Variation: °
Time: 02:26:36.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.79 knots, Course: 212.49°, Date: 14/03/
1925, Variation: °
Time: 02:26:37.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.78 knots, Course: 353.03°, Date: 14/03/
1925, Variation: °
Time: 02:26:38.., Status: Active, Latitude: 33.427° N, Longitude: 111.951° W, Speed: 0.75 knots, Course: 220.24°, Date: 14/03/
1925, Variation: °
```

Getting human readable data!!



Points plotted
There were many overlapping points

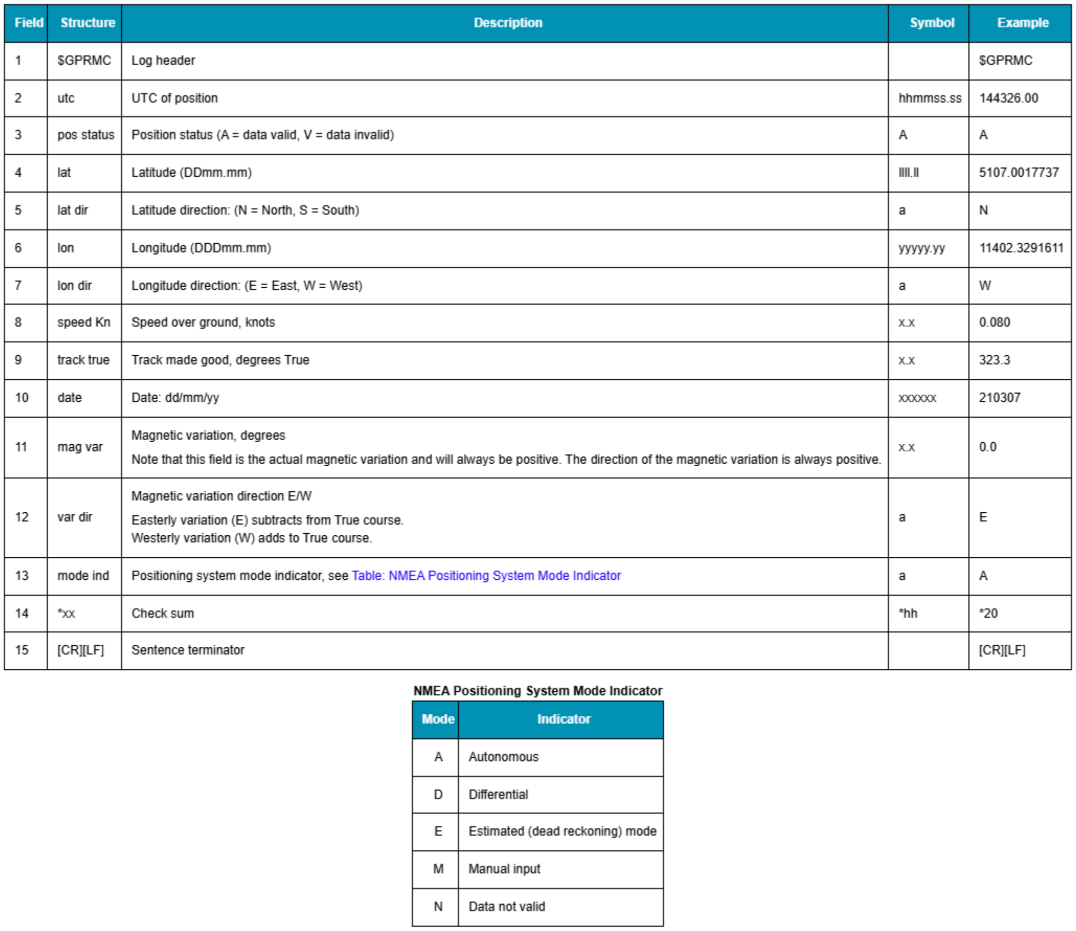
3/11/2025

Next steps:

- ☑ ~~Look at read/gpsget function~~
- ☑ ~~Understand what data type it is returning in the buffer~~
- ☑ ~~See if you can parse the $GPRMS by sentence or by byte chunk~~

3/9/2025

Optimize performant code for a backpacking trip. Calculate power consumption. Decide and battery weight. Develop an algorithm to leverage sleep mode and other more efficient modes like periodic mode and camera resolution.

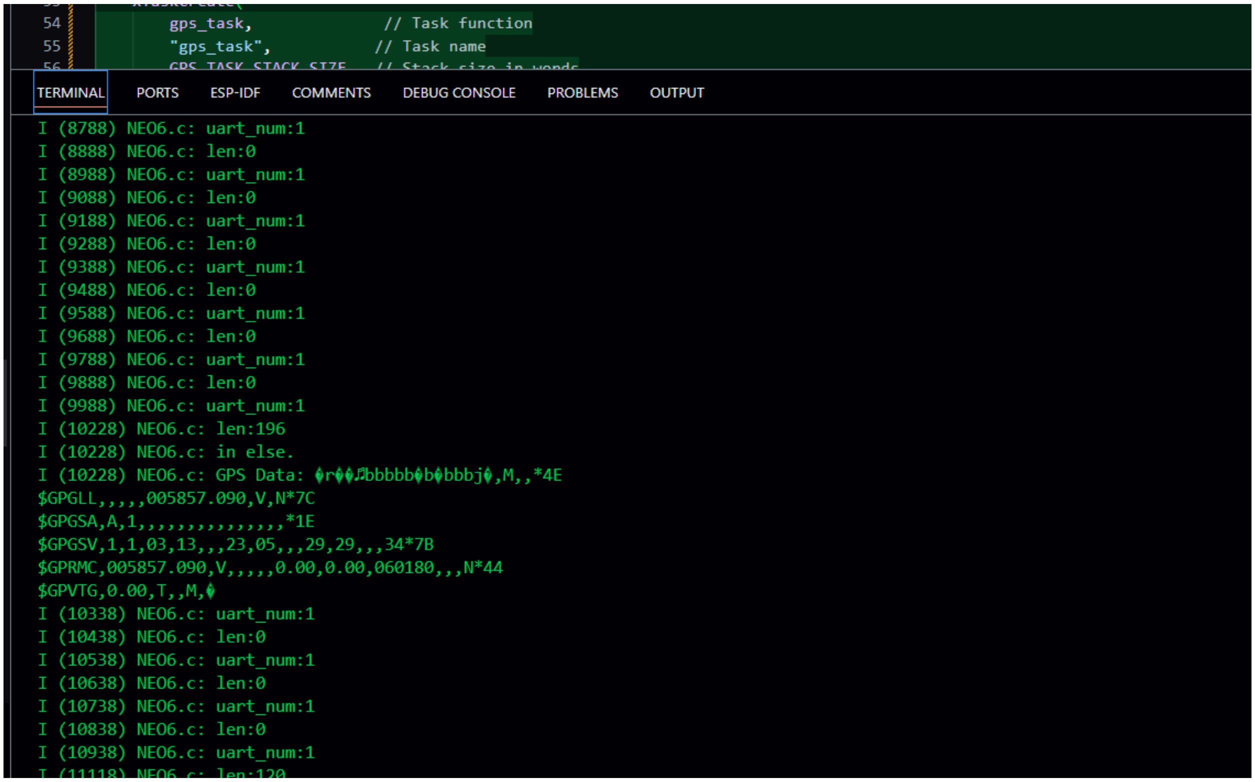


Field	Structure	Description	Symbol	Example
1	$GPRMC	Log header		$GPRMC
2	utc	UTC of position	hhmmss.ss	144326.00
3	pos status	Position status (A = data valid, V = data invalid)	A	A
4	lat	Latitude (DDmm.mm)	llll.ll	5107.0017737
5	lat dir	Latitude direction: (N = North, S = South)	a	N
6	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.3291611
7	lon dir	Longitude direction: (E = East, W = West)	a	W
8	speed Kn	Speed over ground, knots	x.x	0.080
9	track true	Track made good, degrees True	x.x	323.3
10	date	Date: dd/mm/yy	xxxxxx	210307
11	mag var	Magnetic variation, degrees Note that this field is the actual magnetic variation and will always be positive. The direction of the magnetic variation is always positive.	x.x	0.0
12	var dir	Magnetic variation direction E/W Easterly variation (E) subtracts from True course. Westerly variation (W) adds to True course.	a	E
13	mode ind	Positioning system mode indicator, see Table: NMEA Positioning System Mode Indicator	a	A
14	*xx	Check sum	*hh	*20
15	[CR][LF]	Sentence terminator		[CR][LF]

NMEA Positioning System Mode Indicator

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

3/8/2025



```
        gps_task,               // Task function
        "gps_task",             // Task name
        GPS_TASK_STACK_SIZE     // Stack size in words
```

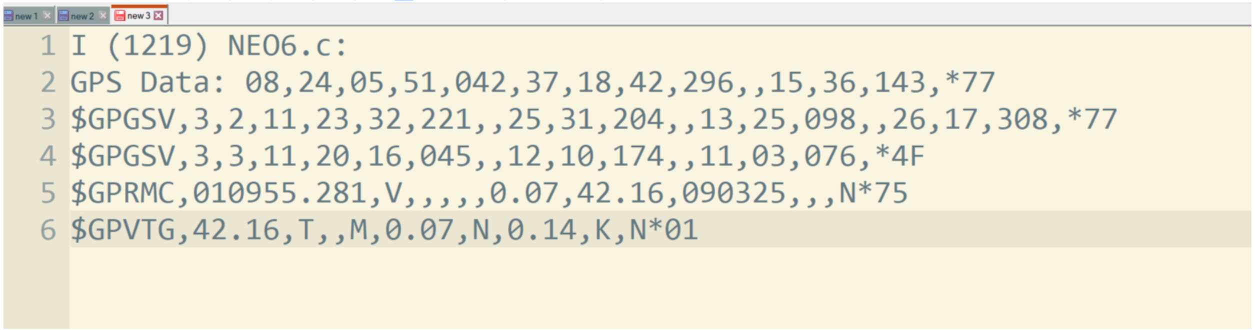
```
I (8788) NEO6.c: uart_num:1
I (8888) NEO6.c: len:0
I (8988) NEO6.c: uart_num:1
I (9088) NEO6.c: len:0
I (9188) NEO6.c: uart_num:1
I (9288) NEO6.c: len:0
I (9388) NEO6.c: uart_num:1
I (9488) NEO6.c: len:0
I (9588) NEO6.c: uart_num:1
I (9688) NEO6.c: len:0
I (9788) NEO6.c: uart_num:1
I (9888) NEO6.c: len:0
I (9988) NEO6.c: uart_num:1
I (10228) NEO6.c: len:196
I (10228) NEO6.c: in else.
I (10228) NEO6.c: GPS Data: �r��♫bbbbb�b�bbbj�,M,,*4E
$GPGLL,,,,,005857.090,V,N*7C
$GPGSA,A,1,,,,,,,,,,,,,,,*1E
$GPGSV,1,1,03,13,,,23,05,,,29,29,,,34*7B
$GPRMC,005857.090,V,,,,,0.00,0.00,060180,,,N*44
$GPVTG,0.00,T,,M,�
I (10338) NEO6.c: uart_num:1
I (10438) NEO6.c: len:0
I (10538) NEO6.c: uart_num:1
I (10638) NEO6.c: len:0
I (10738) NEO6.c: uart_num:1
I (10838) NEO6.c: len:0
I (10938) NEO6.c: uart_num:1
I (11118) NEO6.c: len:120
```

We have something that resembles GPS data

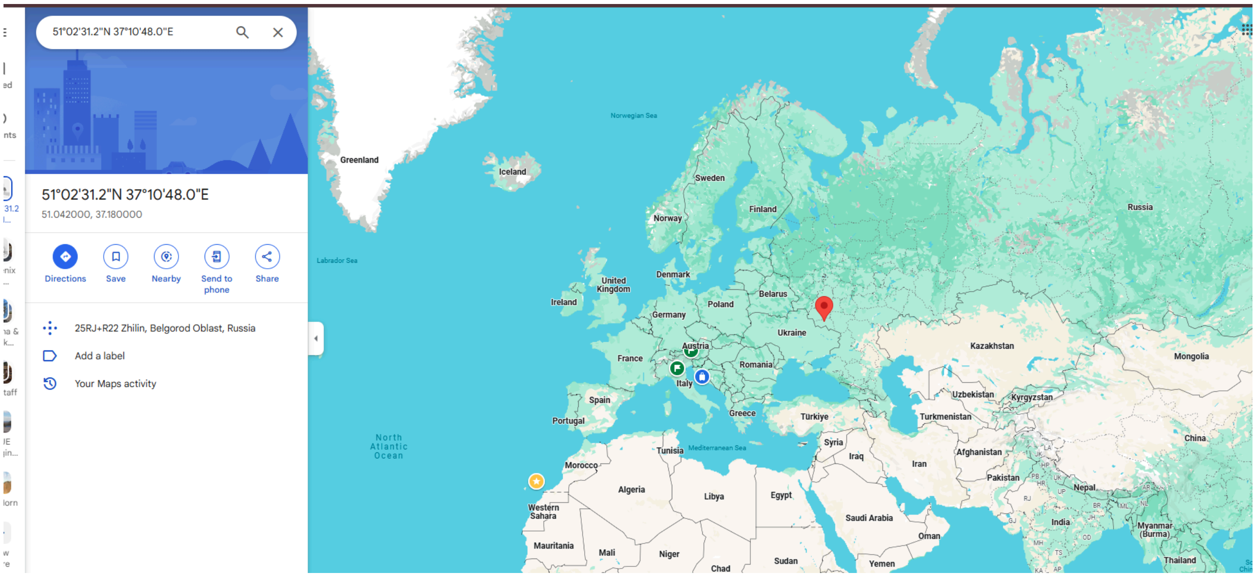
Summary of Acronyms:

- **GPS**: Global Positioning System – The satellite system providing the data.
- **GPGSV**: GPS Satellites in View – Details about visible satellites (ID, elevation, azimuth, SNR).
- **GPRMC**: Recommended Minimum Sentence C – Core GPS data (time, position, speed, date).
- **GPVTG**: Track Made Good and Ground Speed – Speed and course over the ground.

GPGLL: Geographic position, latitude, longitude. GPVTG: Course and speed information relative to the ground. GPRMC: Time, date, position, course, and speed data. GPGSA: GPS receiver operating mode, satellites used in the position solution, and DOP values



```
I (1219) NEO6.c:
GPS Data: 08,24,05,51,042,37,18,42,296,,15,36,143,*77
$GPGSV,3,2,11,23,32,221,,25,31,204,,13,25,098,,26,17,308,*77
$GPGSV,3,3,11,20,16,045,,12,10,174,,11,03,076,*4F
$GPRMC,010955.281,V,,,,,0.07,42.16,090325,,,N*75
$GPVTG,42.16,T,,M,0.07,N,0.14,K,N*01
```

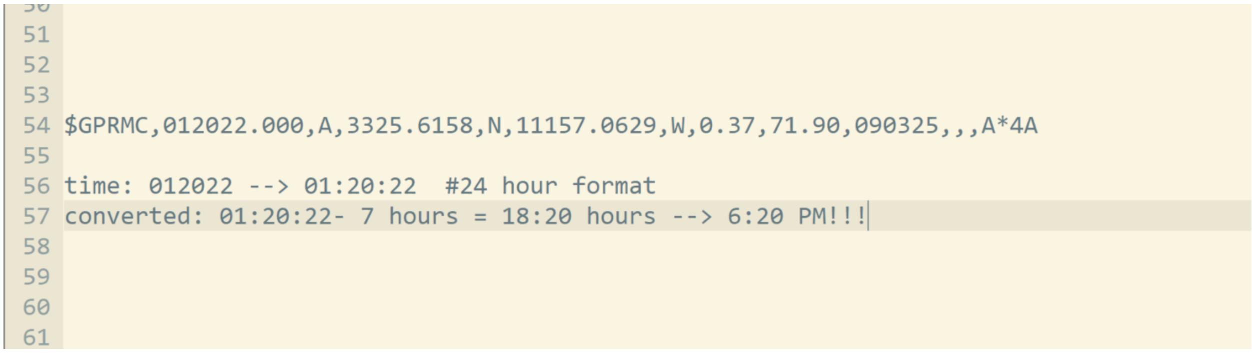


51°02'31.2"N 37°10'48.0"E

51.042000, 37.180000

25RJ+R22 Zhilin, Belgorod Oblast, Russia

Nope thats not it



```
$GPRMC,012022.000,A,3325.6158,N,11157.0629,W,0.37,71.90,090325,,,A*4A

time: 012022 --> 01:20:22  #24 hour format
converted: 01:20:22- 7 hours = 18:20 hours --> 6:20 PM!!!
```

NMEA stands for **National Marine Electronics Association**.

GPRMC: global positioning recommended minimum data type C

$GPRMC,210230,A,3855.4487,N,09446.0071,W,0.0,076.2,130495,003.8,E*69

In this example:
$GPRMC indicates the sentence type.
210230 is the time (UTC).
A indicates a valid fix.
3855.4487,N is the latitude.
09446.0071,W is the longitude.
The remaining fields contain speed, course, date, magnetic variation, and a checksum.

Calculation:
Convert to 24-hour format: 210230 represents 21:02:30 in 24-hour format.
Subtract the time difference: Subtract 7 hours from 21:02:30 UTC.
21:02:30 - 7 hours = 14:02:30.

3/2/2025

Gnd
Tx orange → GPIO15
Rx yellow → GPIO14
Vcc

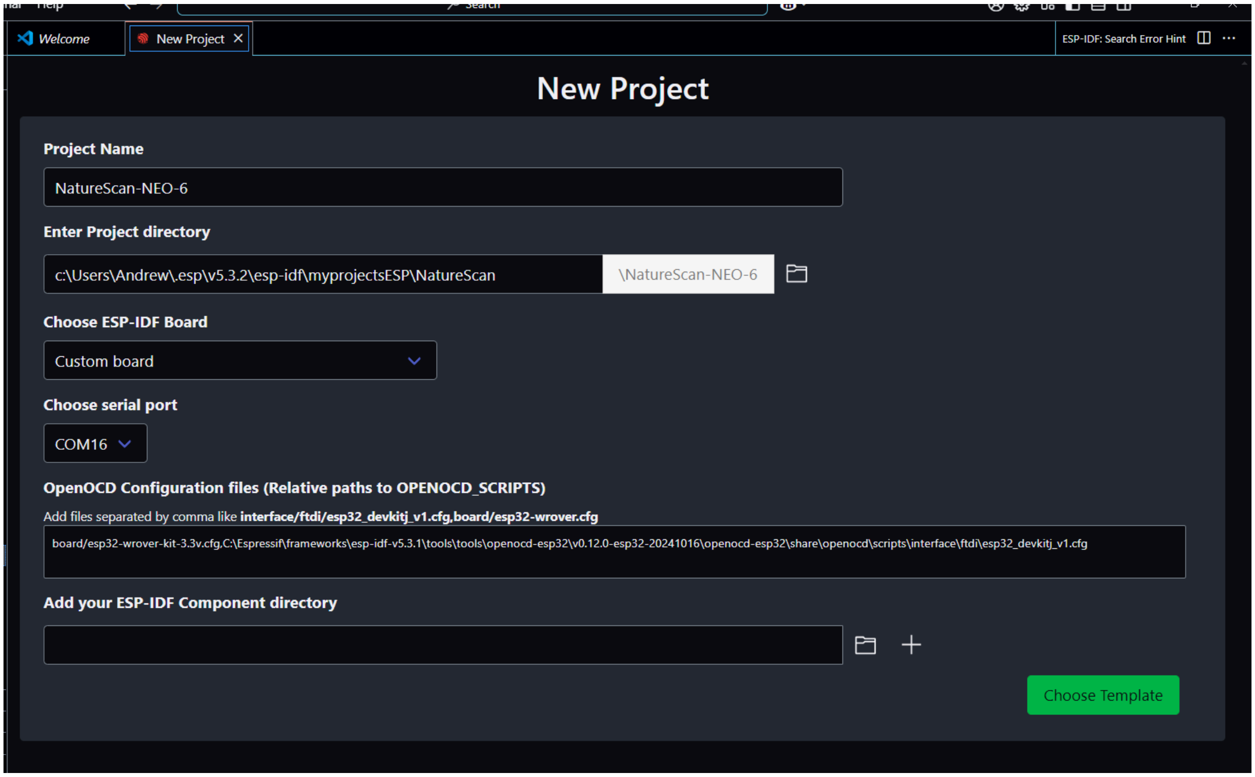
ESP IDF debug
Tools debug check
Version check

```
idf.py --version
python --version
cmake --version
ninja --version
```

NatureScan-NEO-6
Make project
New project

c:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan

board/esp32-wrover-kit-3.3v.cfg,C:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\openocd-esp32\v0.12.0-esp32-20241016\openocd-esp32\share\openocd\scripts\interface\ftdi\esp32_devkitj_v1.cfg
Components left blank



New Project

Project Name
NatureScan-NEO-6

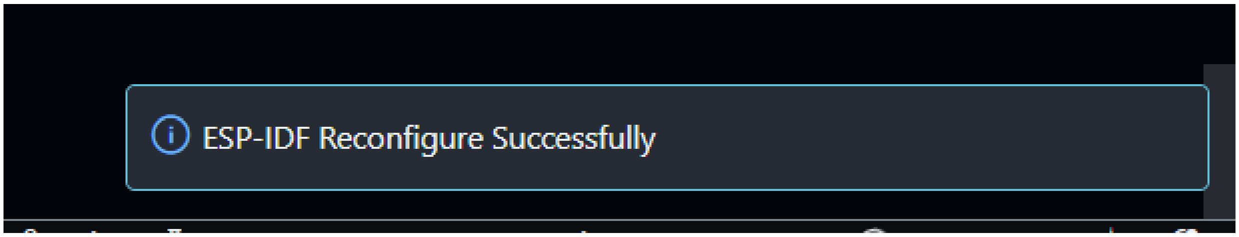
Enter Project directory
c:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan \NatureScan-NEO-6

Choose ESP-IDF Board
Custom board

Choose serial port
COM16

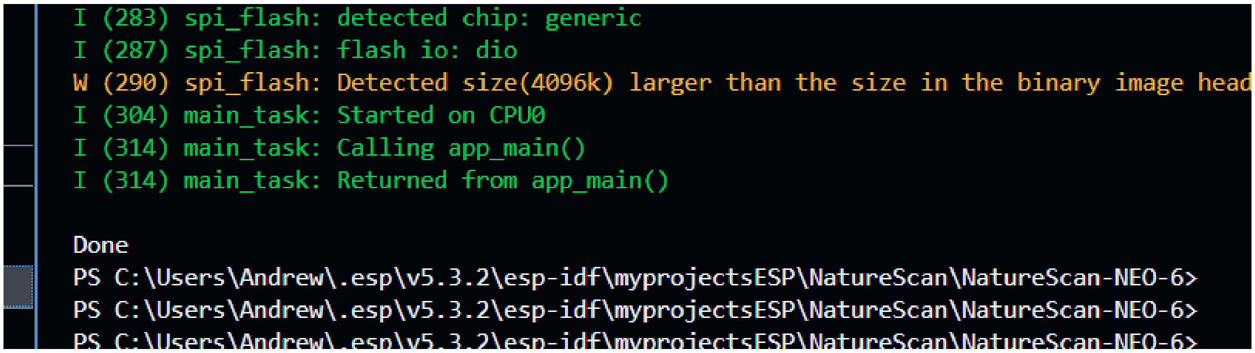
OpenOCD Configuration files (Relative paths to OPENOCD_SCRIPTS)
Add files separated by comma like **interface/ftdi/esp32_devkitj_v1.cfg,board/esp32-wrover.cfg**
board/esp32-wrover-kit-3.3v.cfg,C:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\openocd-esp32\v0.12.0-esp32-20241016\openocd-esp32\share\openocd\scripts\interface\ftdi\esp32_devkitj_v1.cfg

Add your ESP-IDF Component directory

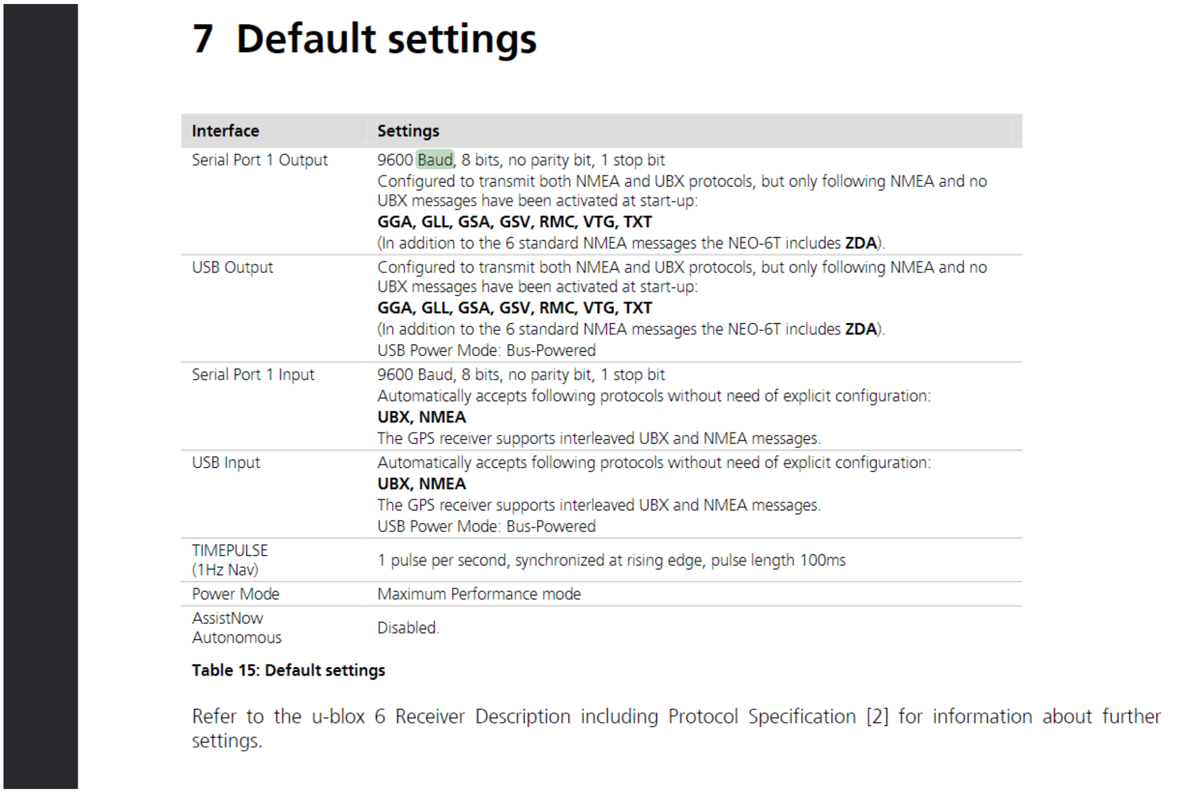


ESP-IDF Reconfigure Successfully

Lfg



```
I (283) spi_flash: detected chip: generic
I (287) spi_flash: flash io: dio
W (290) spi_flash: Detected size(4096k) larger than the size in the binary image head
I (304) main_task: Started on CPU0
I (314) main_task: Calling app_main()
I (314) main_task: Returned from app_main()

Done
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6>
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6>
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6>
```

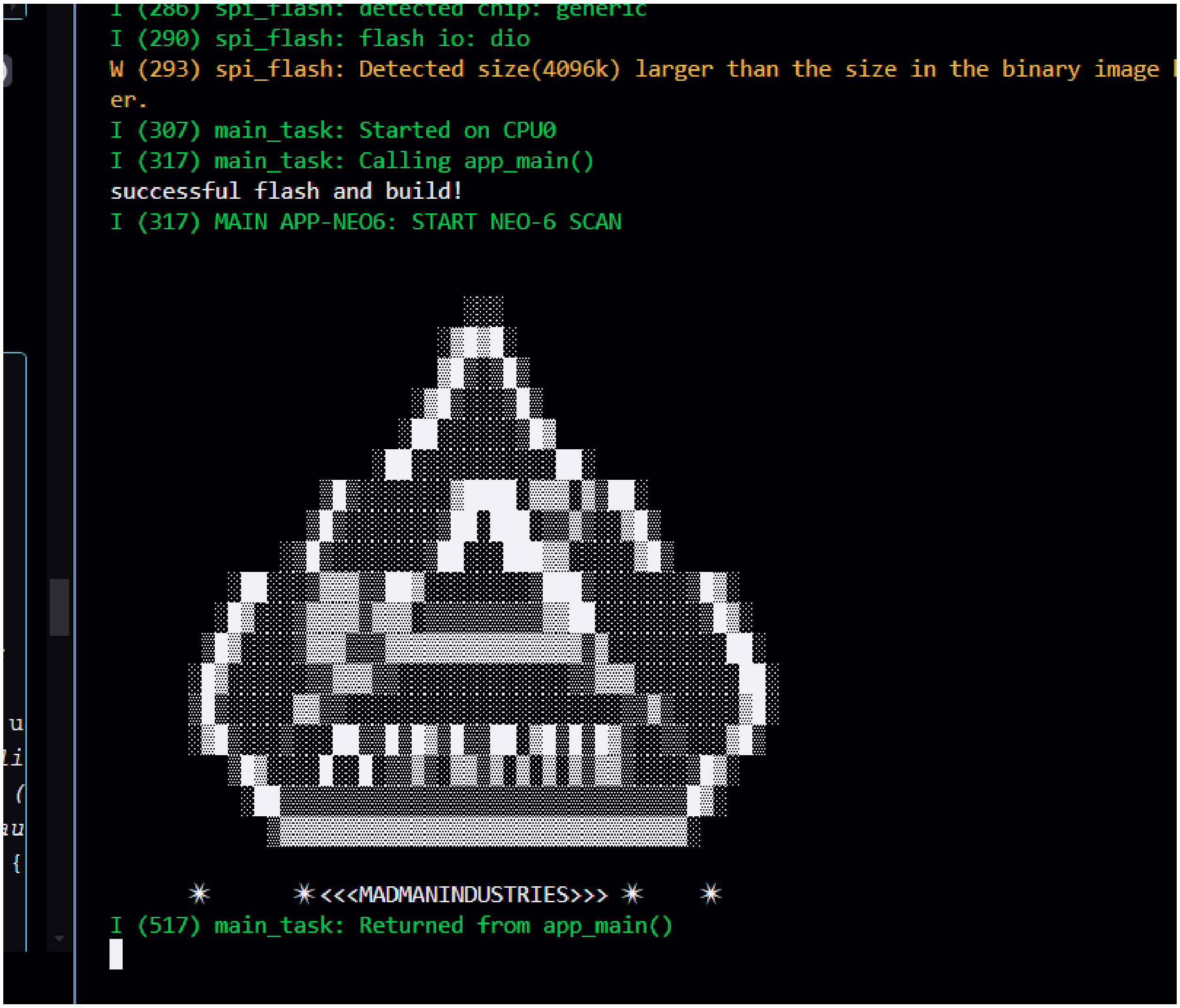


7 Default settings

Interface	Settings
Serial Port 1 Output	9600 Baud, 8 bits, no parity bit, 1 stop bit Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: **GGA, GLL, GSA, GSV, RMC, VTG, TXT** (In addition to the 6 standard NMEA messages the NEO-6T includes **ZDA**).
USB Output	Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: **GGA, GLL, GSA, GSV, RMC, VTG, TXT** (In addition to the 6 standard NMEA messages the NEO-6T includes **ZDA**). USB Power Mode: Bus-Powered
Serial Port 1 Input	9600 Baud, 8 bits, no parity bit, 1 stop bit Automatically accepts following protocols without need of explicit configuration: **UBX, NMEA** The GPS receiver supports interleaved UBX and NMEA messages.
USB Input	Automatically accepts following protocols without need of explicit configuration: **UBX, NMEA** The GPS receiver supports interleaved UBX and NMEA messages. USB Power Mode: Bus-Powered
TIMEPULSE (1Hz Nav)	1 pulse per second, synchronized at rising edge, pulse length 100ms
Power Mode	Maximum Performance mode
AssistNow Autonomous	Disabled.

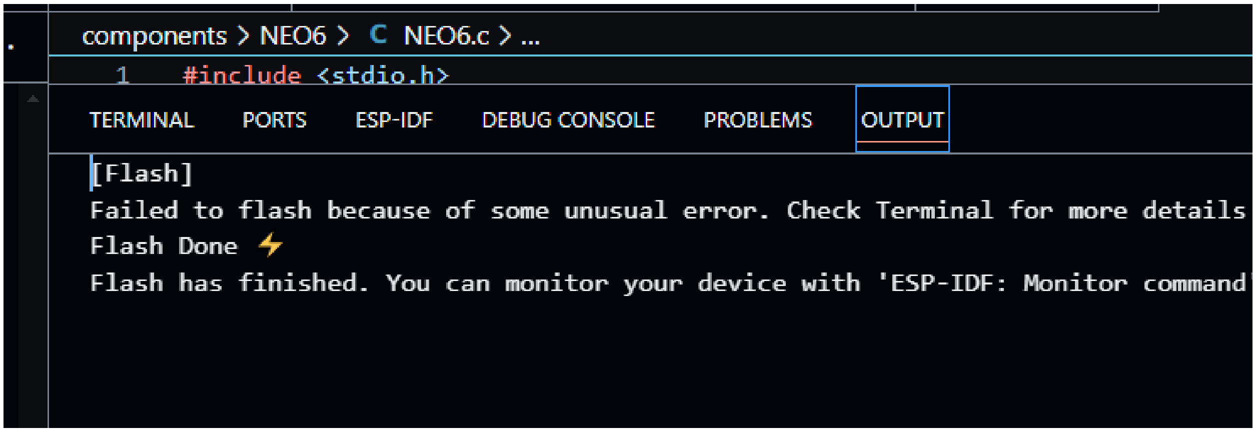
Table 15: Default settings

Refer to the u-blox 6 Receiver Description including Protocol Specification [2] for information about further settings.



```
I (286) spi_flash: detected chip: generic
I (290) spi_flash: flash io: dio
W (293) spi_flash: Detected size(4096k) larger than the size in the binary image 
er.
I (307) main_task: Started on CPU0
I (317) main_task: Calling app_main()
successful flash and build!
I (317) MAIN APP-NEO6: START NEO-6 SCAN

      ✷       ✷<<<MADMANINDUSTRIES>>> ✷     ✷
I (517) main_task: Returned from app_main()
```

UART INIT SUCCESS!!!

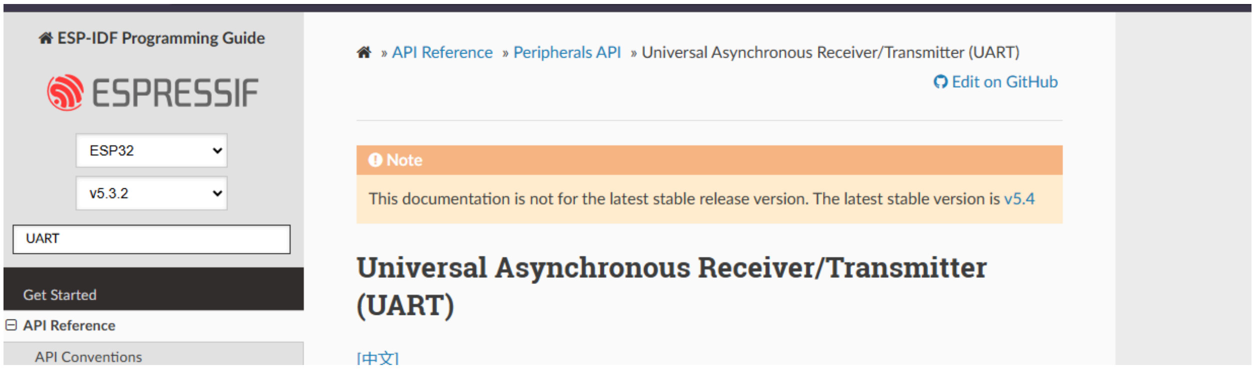


```
#include <stdio.h>
```

```
[Flash]
Failed to flash because of some unusual error. Check Terminal for more details
Flash Done ⚡
Flash has finished. You can monitor your device with 'ESP-IDF: Monitor command'
```

Flash
⚡

3/1/2025


ESP-IDF Programming Guide
ESPRESSIF
ESP32
v5.3.2
UART
Get Started
⊟ API Reference
API Conventions
» API Reference » Peripherals API » Universal Asynchronous Receiver/Transmitter (UART)
Edit on GitHub

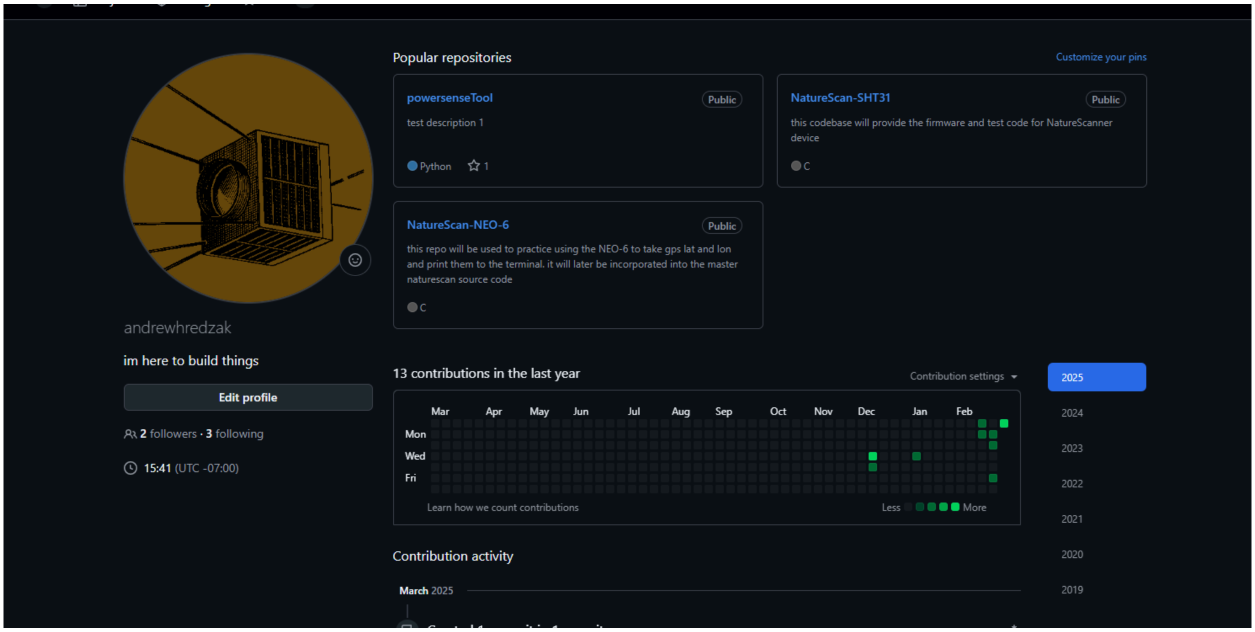
Note

This documentation is not for the latest stable release version. The latest stable version is v5.4

Universal Asynchronous Receiver/Transmitter (UART)

[中文]

link



Popular repositories

Customize your pins

powersenseTool Public
test description 1
Python ☆ 1

NatureScan-SHT31 Public
this codebase will provide the firmware and test code for NatureScanner device
C

NatureScan-NEO-6 Public
this repo will be used to practice using the NEO-6 to take gps lat and lon and print them to the terminal. it will later be incorporated into the master naturescan source code
C

andrewhredzak

im here to build things

Edit profile

2 followers · 3 following

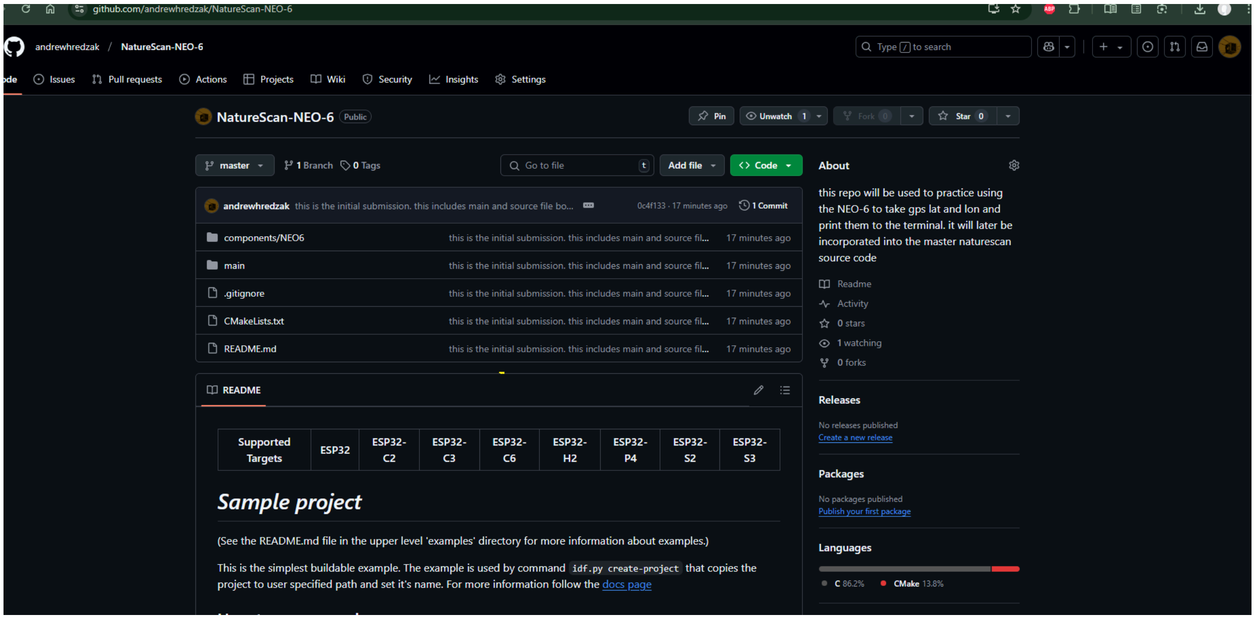
15:41 (UTC -07:00)

13 contributions in the last year

Contribution settings

Contribution activity

March 2025



github.com/andrewhredzak/NatureScan-NEO-6

andrewhredzak / NatureScan-NEO-6

NatureScan-NEO-6 Public

1 Branch 0 Tags

andrewhredzak this is the initial submission. this includes main and source file bo... 0c4f133 · 17 minutes ago 1 Commit

components/NEO6	this is the initial submission. this includes main and source fil...	17 minutes ago
main	this is the initial submission. this includes main and source fil...	17 minutes ago
.gitignore	this is the initial submission. this includes main and source fil...	17 minutes ago
CMakeLists.txt	this is the initial submission. this includes main and source fil...	17 minutes ago
README.md	this is the initial submission. this includes main and source fil...	17 minutes ago

About

this repo will be used to practice using the NEO-6 to take gps lat and lon and print them to the terminal. it will later be incorporated into the master naturescan source code

Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

C 86.2% CMake 13.8%

README

Supported Targets	ESP32	ESP32-C2	ESP32-C3	ESP32-C6	ESP32-H2	ESP32-P4	ESP32-S2	ESP32-S3

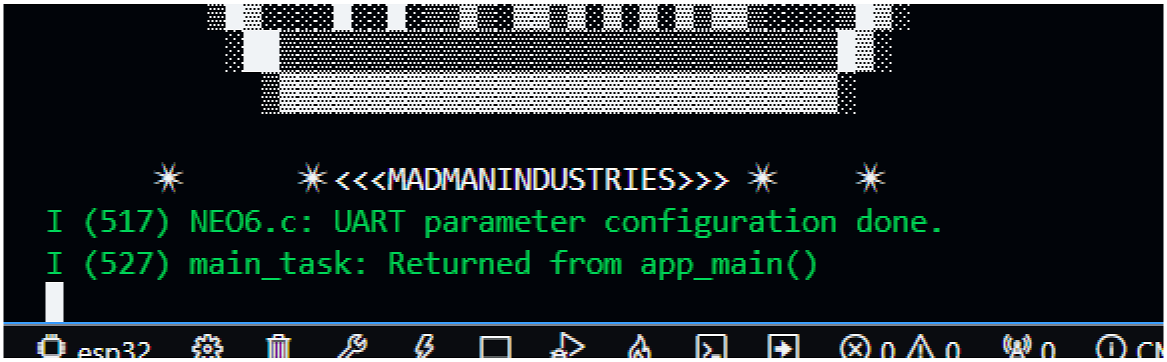
Sample project

(See the README.md file in the upper level 'examples' directory for more information about examples.)

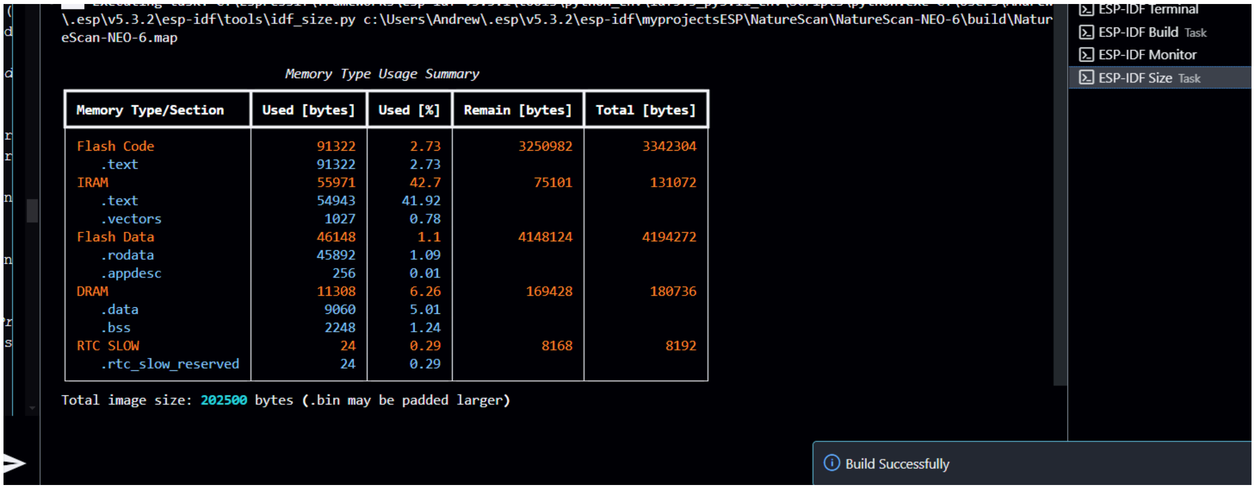
This is the simplest buildable example. The example is used by command `idf.py create-project` that copies the project to user specified path and set it's name. For more information follow the docs page

d

Posted to github as well

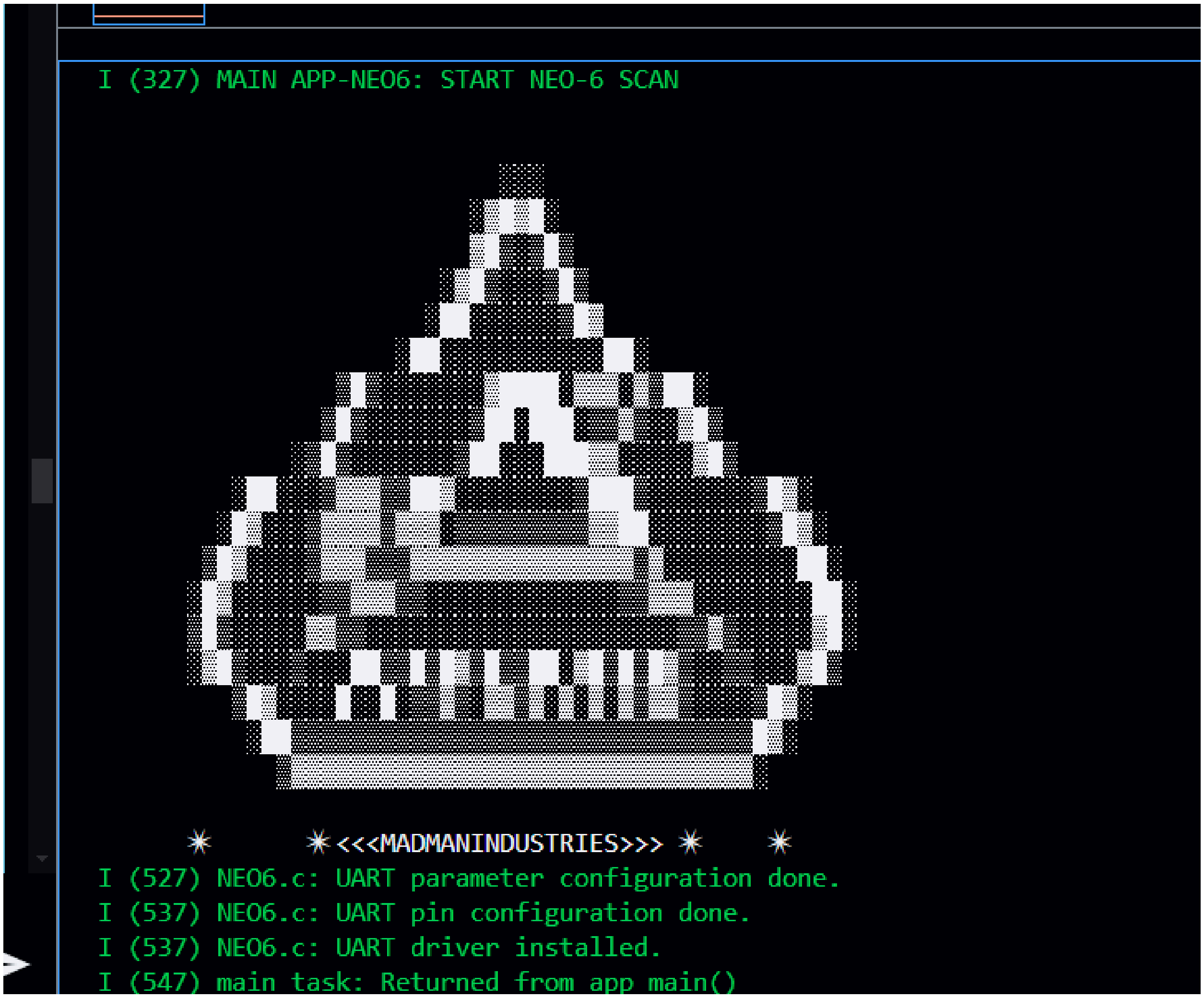


```
 ✷       ✷<<<MADMANINDUSTRIES>>> ✷     ✷
I (517) NEO6.c: UART parameter configuration done.
I (527) main_task: Returned from app_main()
```



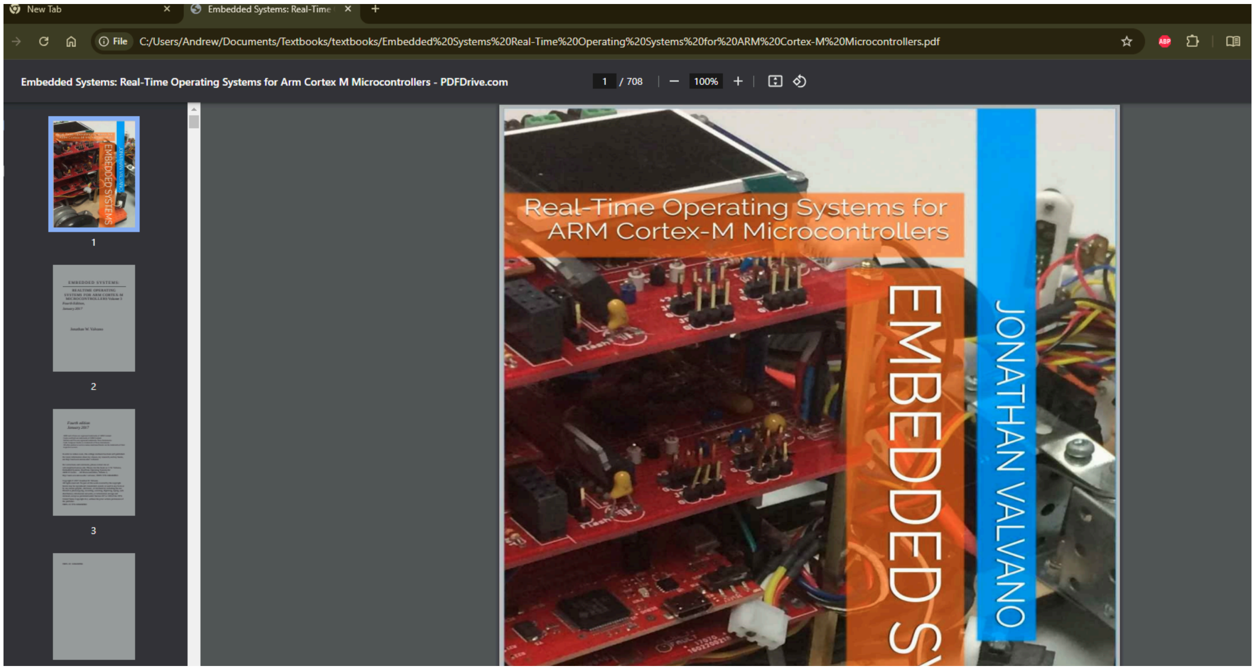
```
\.esp\v5.3.2\esp-idf\tools\idf_size.py c:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6\build\Natur
eScan-NEO-6.map

                              Memory Type Usage Summary
┌───────────────────────┬──────────────┬──────────┬────────────────┬───────────────┐
│ Memory Type/Section   │ Used [bytes] │ Used [%] │ Remain [bytes] │ Total [bytes] │
├───────────────────────┼──────────────┼──────────┼────────────────┼───────────────┤
│ Flash Code            │        91322 │     2.73 │        3250982 │       3342304 │
│    .text              │        91322 │     2.73 │                │               │
│ IRAM                  │        55971 │     42.7 │          75101 │        131072 │
│    .text              │        54943 │    41.92 │                │               │
│    .vectors           │         1027 │     0.78 │                │               │
│ Flash Data            │        46148 │      1.1 │        4148124 │       4194272 │
│    .rodata            │        45892 │     1.09 │                │               │
│    .appdesc           │          256 │     0.01 │                │               │
│ DRAM                  │        11308 │     6.26 │         169428 │        180736 │
│    .data              │         9060 │     5.01 │                │               │
│    .bss               │         2248 │     1.24 │                │               │
│ RTC SLOW              │           24 │     0.29 │           8168 │          8192 │
│    .rtc_slow_reserved │           24 │     0.29 │                │               │
└───────────────────────┴──────────────┴──────────┴────────────────┴───────────────┘
Total image size: 202500 bytes (.bin may be padded larger)
```

Build Successfully

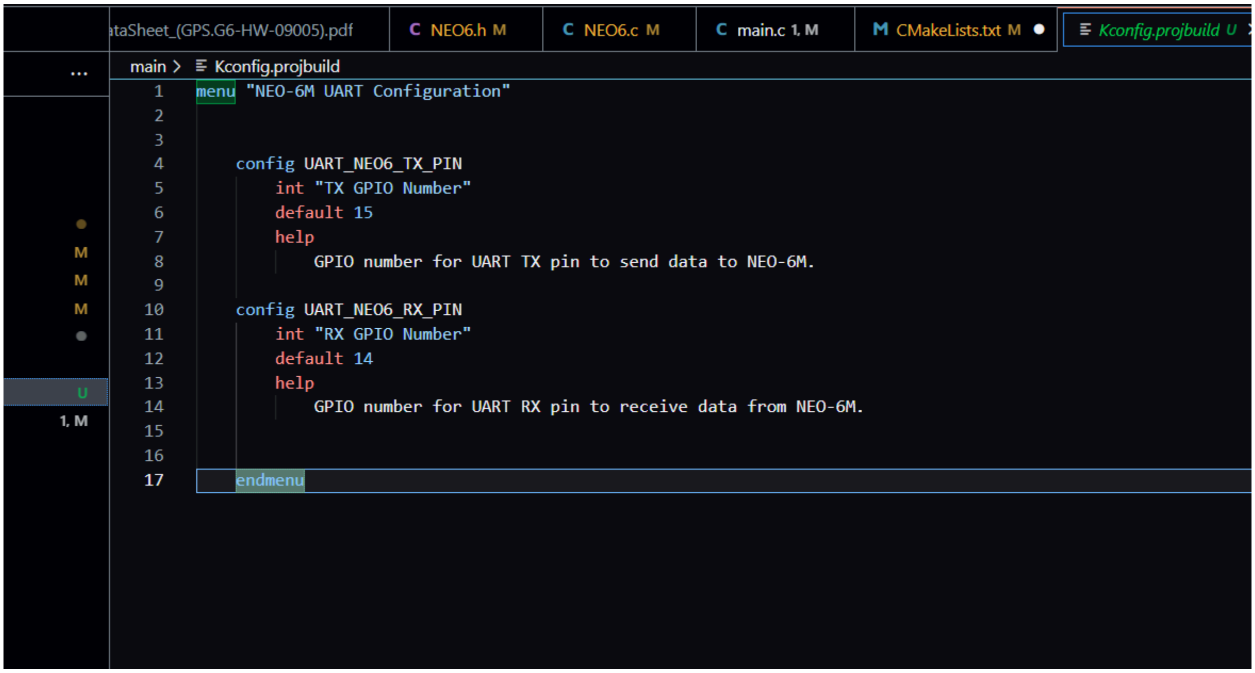


```
I (327) MAIN APP-NEO6: START NEO-6 SCAN

     ✷       ✷<<<MADMANINDUSTRIES>>> ✷     ✷
I (527) NEO6.c: UART parameter configuration done.
I (537) NEO6.c: UART pin configuration done.
I (537) NEO6.c: UART driver installed.
I (547) main task: Returned from app main()
```

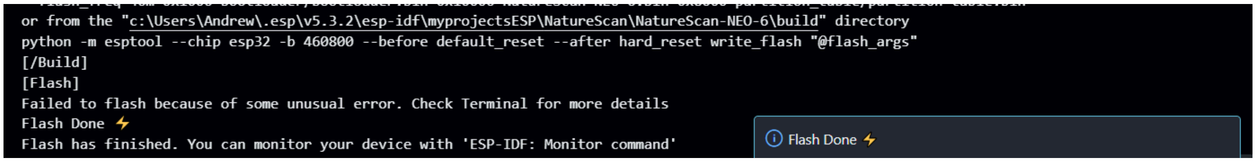
Lets goooooo!!!


New Tab
Embedded Systems: Real-Time
C:/Users/Andrew/Documents/Textbooks/textbooks/Embedded%20Systems%20Real-Time%20Operating%20Systems%20for%20ARM%20Cortex-M%20Microcontrollers.pdf
Embedded Systems: Real-Time Operating Systems for Arm Cortex M Microcontrollers - PDFDrive.com
1 / 708
100%
Real-Time Operating Systems for ARM Cortex-M Microcontrollers
EMBEDDED S
JONATHAN VALVANO

Grok robots suggested

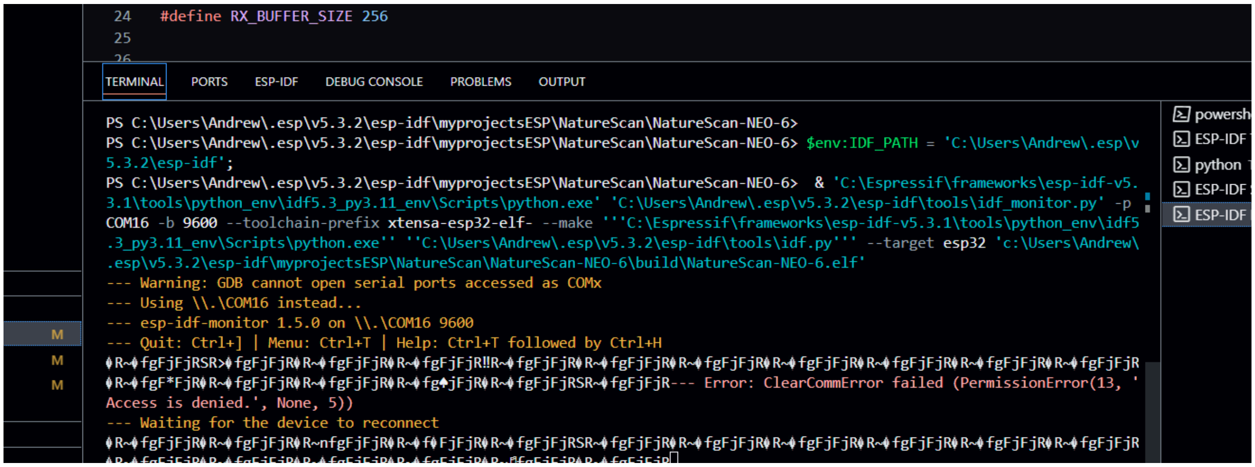


```
menu "NEO-6M UART Configuration"


    config UART_NEO6_TX_PIN
        int "TX GPIO Number"
        default 15
        help
            GPIO number for UART TX pin to send data to NEO-6M.

    config UART_NEO6_RX_PIN
        int "RX GPIO Number"
        default 14
        help
            GPIO number for UART RX pin to receive data from NEO-6M.


    endmenu
```



```
or from the "c:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6\build" directory
python -m esptool --chip esp32 -b 460800 --before default_reset --after hard_reset write_flash "@flash_args"
[/Build]
[Flash]
Failed to flash because of some unusual error. Check Terminal for more details
Flash Done ⚡
Flash has finished. You can monitor your device with 'ESP-IDF: Monitor command'
```

Resettt



```
#define RX_BUFFER_SIZE 256
```

```
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6>
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6> $env:IDF_PATH = 'C:\Users\Andrew\.esp\v
5.3.2\esp-idf';
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6>  & 'C:\Espressif\frameworks\esp-idf-v5.
3.1\tools\python_env\idf5.3_py3.11_env\Scripts\python.exe' 'C:\Users\Andrew\.esp\v5.3.2\esp-idf\tools\idf_monitor.py' -p
COM16 -b 9600 --toolchain-prefix xtensa-esp32-elf- --make '''C:\Espressif\frameworks\esp-idf-v5.3.1\tools\python_env\idf5
.3_py3.11_env\Scripts\python.exe'' ''C:\Users\Andrew\.esp\v5.3.2\esp-idf\tools\idf.py''' --target esp32 'c:\Users\Andrew\
.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan\NatureScan-NEO-6\build\NatureScan-NEO-6.elf'
--- Warning: GDB cannot open serial ports accessed as COMx
--- Using \\.\COM16 instead...
--- esp-idf-monitor 1.5.0 on \\.\COM16 9600
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H
�R~�fgFjFjRSR>�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR‼R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR
�R~�fgF*FjR�R~�fgFjFjR�R~�fgFjFjR�R~�fg♠jFjR�R~�fgFjFjRSR~�fgFjFjR--- Error: ClearCommError failed (PermissionError(13, '
Access is denied.', None, 5))
--- Waiting for the device to reconnect
�R~�fgFjFjR�R~�fgFjFjR�R~nfgFjFjR�R~�f�FjFjR�R~�fgFjFjRSR~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR�R~�fgFjFjR
```

Thats where we leave it

2/28/2025

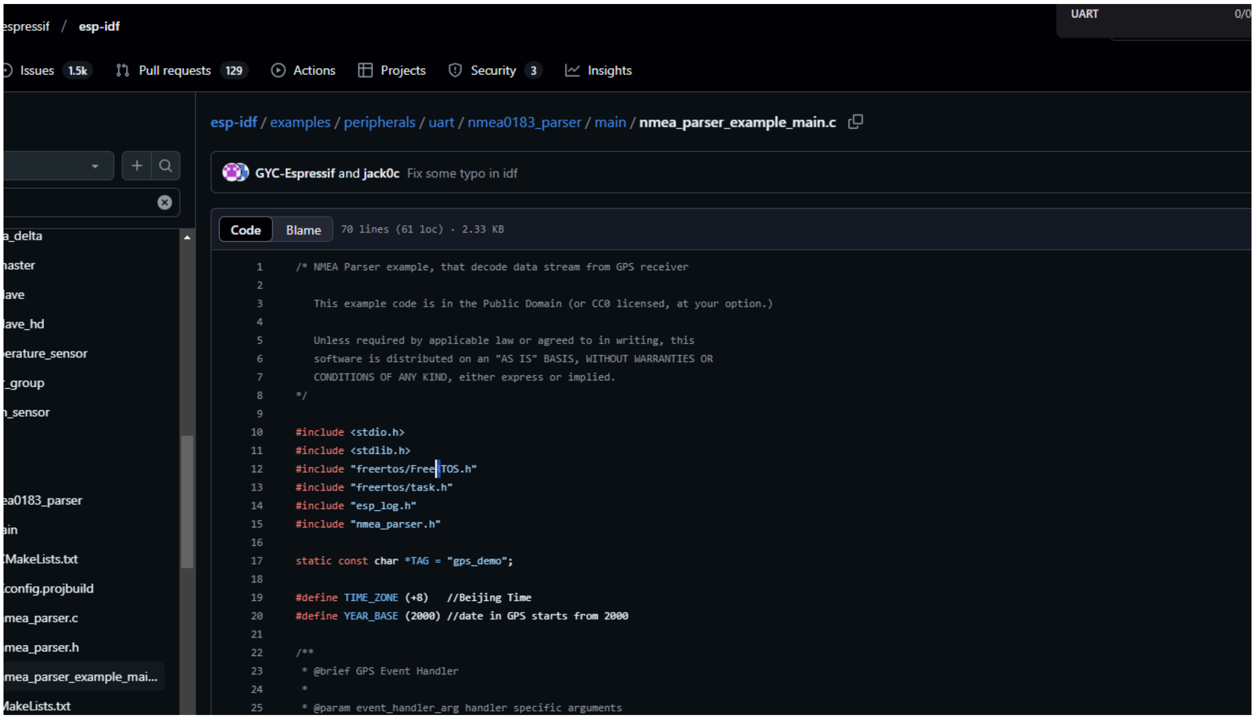
UART for GPS

So i want to focus on understanding UART protocol and how it works.
I also want to focus on building from the documents so the first three tasks will be

- ☑ ~~Write a clear prompt that will be tailored to building a gps codebase for models~~
- ☐ Read NEO-M8N documentation
- ☑ ~~Find and read UART library documentation~~
    - ☑ ~~#include "driver/uart.h"~~

☐ #include "driver/gpio.h"
☐ "driver/uart_vfs.h"

Virtual filesystem (VFS) component provides a unified interface for drivers which can perform operations on file-like objects. These can be real filesystems (FAT, SPIFFS, etc.) or device drivers which provide a file-like interface.



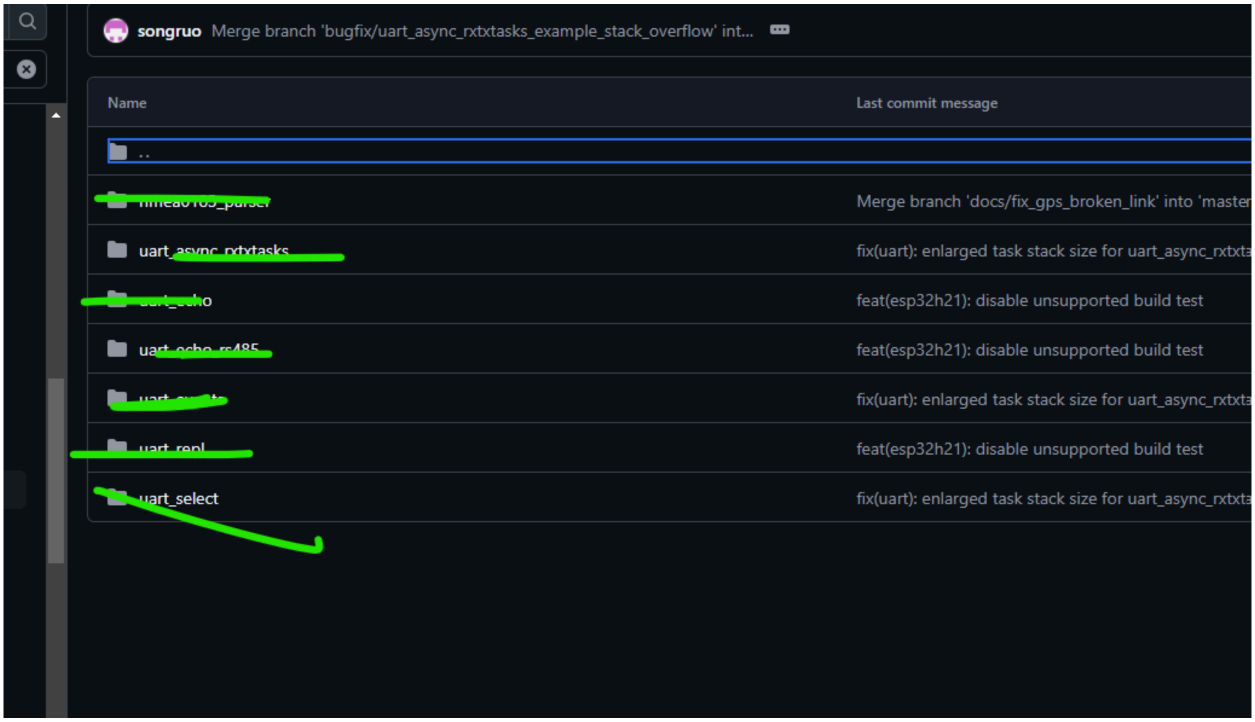
esp-idf / examples / peripherals / uart / nmea0183_parser / main / nmea_parser_example_main.c

GYC-Espressif and jack0c Fix some typo in idf

70 lines (61 loc) · 2.33 KB

```
/* NMEA Parser example, that decode data stream from GPS receiver

   This example code is in the Public Domain (or CC0 licensed, at your option.)

   Unless required by applicable law or agreed to in writing, this
   software is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
   CONDITIONS OF ANY KIND, either express or implied.
*/

#include <stdio.h>
#include <stdlib.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_log.h"
#include "nmea_parser.h"

static const char *TAG = "gps_demo";

#define TIME_ZONE (+8)   //Beijing Time
#define YEAR_BASE (2000) //date in GPS starts from 2000

/**
 * @brief GPS Event Handler
 *
 * @param event_handler_arg handler specific arguments
```

Gps example

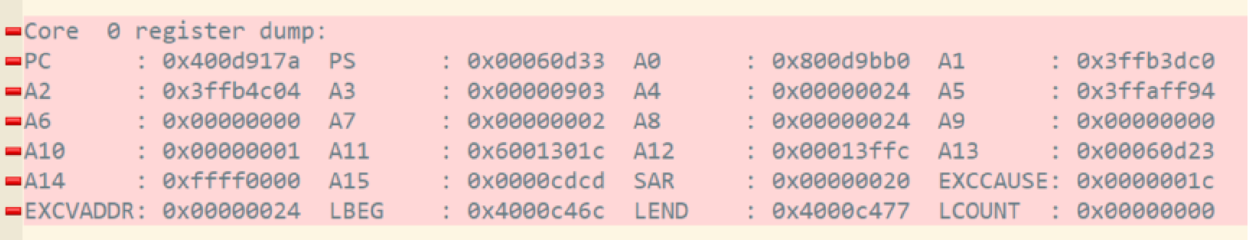


songruo Merge branch 'bugfix/uart_async_rxtxtasks_example_stack_overflow' int...

Name	Last commit message
..	
~~nmea0183_parser~~	Merge branch 'docs/fix_gps_broken_link' into 'master
~~uart_async_rxtxtasks~~	fix(uart): enlarged task stack size for uart_async_rxtxta
~~uart_echo~~	feat(esp32h21): disable unsupported build test
~~uart_echo_rs485~~	feat(esp32h21): disable unsupported build test
~~uart_events~~	fix(uart): enlarged task stack size for uart_async_rxtxta
~~uart_repl~~	feat(esp32h21): disable unsupported build test
uart_select	fix(uart): enlarged task stack size for uart_async_rxtxta

2/27/2025

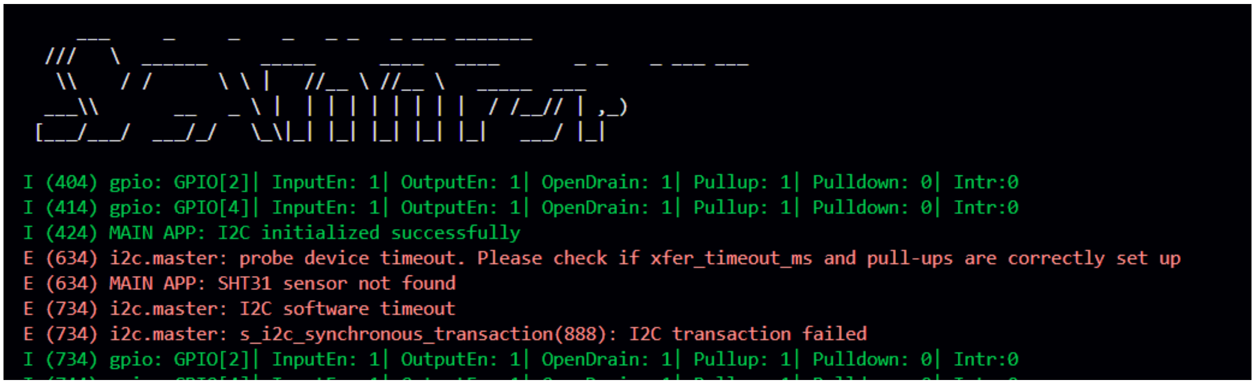
Register dump



```
Core  0 register dump:
PC      : 0x400d917a  PS      : 0x00060d33  A0      : 0x800d9bb0  A1      : 0x3ffb3dc0
A2      : 0x3ffb4c04  A3      : 0x00000903  A4      : 0x00000024  A5      : 0x3ffaff94
A6      : 0x00000000  A7      : 0x00000002  A8      : 0x00000024  A9      : 0x00000000
A10     : 0x00000001  A11     : 0x6001301c  A12     : 0x00013ffc  A13     : 0x00060d23
A14     : 0xffff0000  A15     : 0x0000cdcd  SAR     : 0x00000020  EXCCAUSE: 0x0000001c
EXCVADDR: 0x00000024  LBEG    : 0x4000c46c  LEND    : 0x4000c477  LCOUNT  : 0x00000000
```

Label	Value	Description
Core	0	Indicates this is from Core 0 of a multi-core processor.
PC	0x400d917a	Program Counter: Address of the next instruction before the exception.
PS	0x8006d33	Processor Status: Contains control bits (e.g., interrupt enable, mode).
A0	0x8006d33	General-purpose register (e.g., might hold a function argument or address).

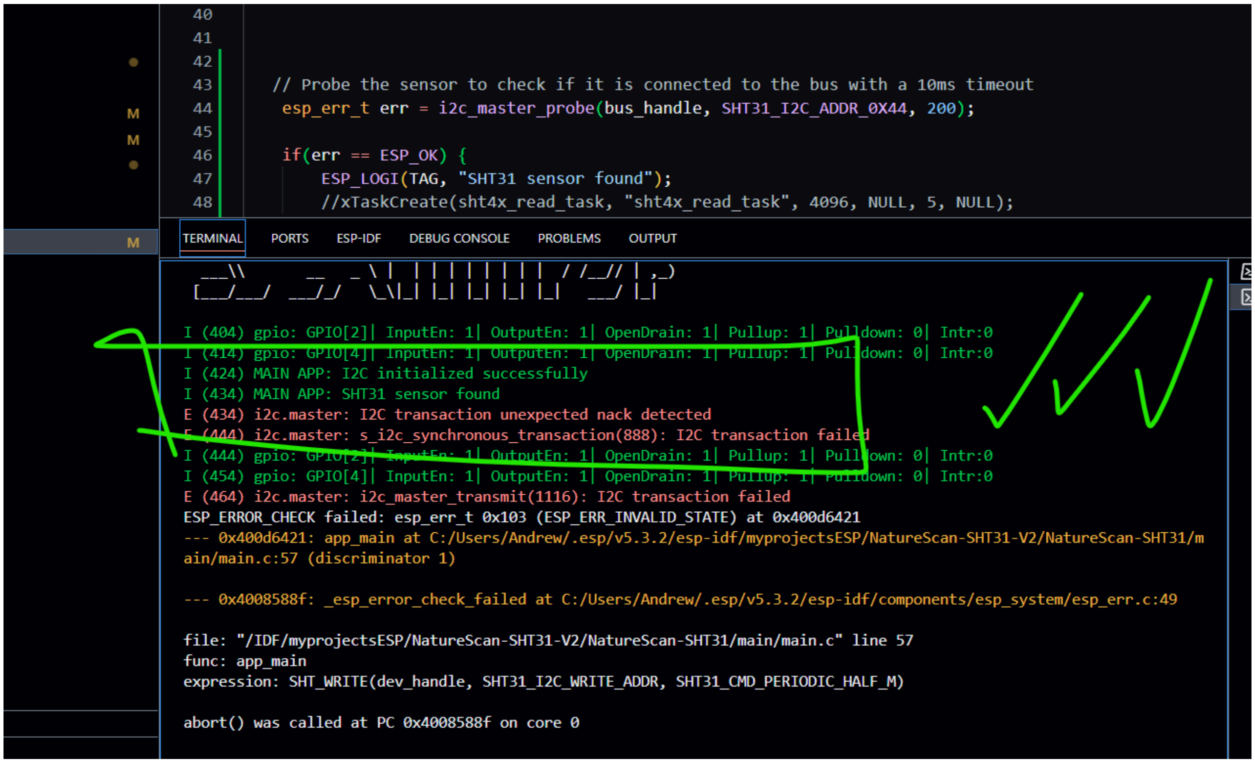
A1	0x3ffbdcc	General-purpose register.
A2	0x3ffb4c04	General-purpose register.
A3	0x00000003	General-purpose register.
A4	0x00000024	General-purpose register.
A5	0x3ffaff94	General-purpose register.
A6	0x00000000	General-purpose register.
A7	0x00000002	General-purpose register.
A8	0x00000000	General-purpose register.
A9	0x00000000	General-purpose register.
A10	0x00000001	General-purpose register.
A11	0x00000001	General-purpose register.
A12	0x6001301c	General-purpose register.
A13	0x00013ffc	General-purpose register.
A14	0x00000000	General-purpose register.
A15	0x4000c4cc	General-purpose register.
SAR	0x4000c4cc	Special Address Register: Used for address arithmetic or loops.
EXCCAUSE	0x0000001c	Exception Cause: Reason for the exception (e.g., illegal instruction or memory error).
EXCVADDR	0x80000024	Exception Address: Memory address that triggered the exception.
LBEG	0x4000c4cc	Loop Begin: Start address of a hardware loop.
LEND	0x4000c477	Loop End: End address of a hardware loop.
LCOUNT	0x00000000	Loop Count: Number of iterations for the loop (0 indicates inactive).



```
    ___      _      _     _    _ _    _ ___ _______
  ///   \  ______       _____       ____   ____       _ _    _ ___ ___
   \\    / /       \ \ |   //__ \ //__ \   _____  ___
  ___\\        __   _ \ |  | | | | | | | |  / /__// | ,_)
 [___/___/   ___/_/   \_\|_| |_| |_| |_| |_|   ___/ |_|

I (404) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (414) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (424) MAIN APP: I2C initialized successfully
E (634) i2c.master: probe device timeout. Please check if xfer_timeout_ms and pull-ups are correctly set up
E (634) MAIN APP: SHT31 sensor not found
E (734) i2c.master: I2C software timeout
E (734) i2c.master: s_i2c_synchronous_transaction(888): I2C transaction failed
I (734) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
```

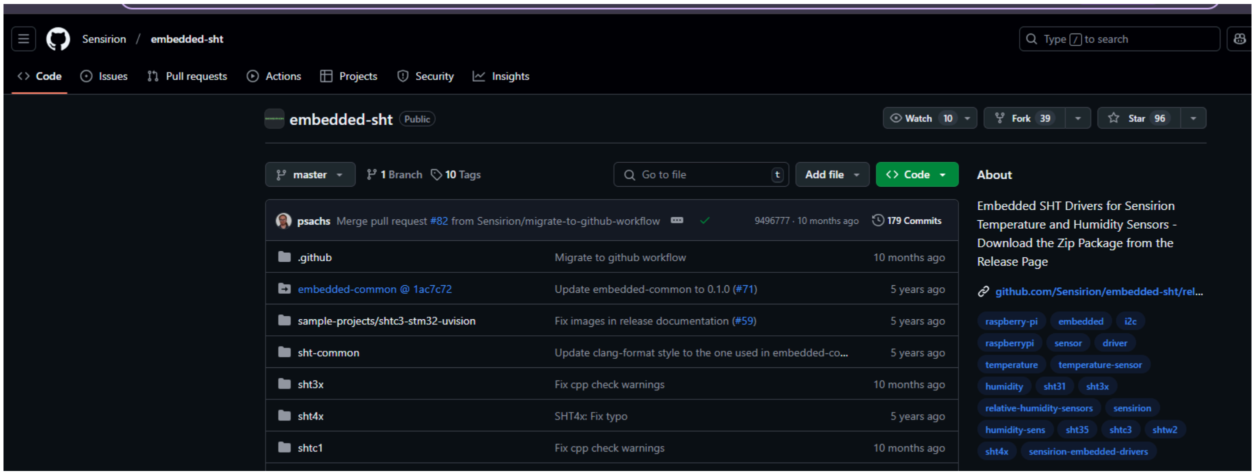
I2C software timeout. Probe didnt work

DIDNT PLUG IT IN.
SENSOR FOUND!!



```c



    // Probe the sensor to check if it is connected to the bus with a 10ms timeout
     esp_err_t err = i2c_master_probe(bus_handle, SHT31_I2C_ADDR_0X44, 200);

     if(err == ESP_OK) {
         ESP_LOGI(TAG, "SHT31 sensor found");
         //xTaskCreate(sht4x_read_task, "sht4x_read_task", 4096, NULL, 5, NULL);
```

```
  ___\\        __   _ \ |  | | | | | | | |  / /__// | ,_)
 [___/___/   ___/_/   \_\|_| |_| |_| |_| |_|   ___/ |_|

I (404) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (414) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (424) MAIN APP: I2C initialized successfully
I (434) MAIN APP: SHT31 sensor found
E (434) i2c.master: I2C transaction unexpected nack detected
E (444) i2c.master: s_i2c_synchronous_transaction(888): I2C transaction failed
I (444) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (454) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
E (464) i2c.master: i2c_master_transmit(1116): I2C transaction failed
ESP_ERROR_CHECK failed: esp_err_t 0x103 (ESP_ERR_INVALID_STATE) at 0x400d6421
--- 0x400d6421: app_main at C:/Users/Andrew/.esp/v5.3.2/esp-idf/myprojectsESP/NatureScan-SHT31-V2/NatureScan-SHT31/m
ain/main.c:57 (discriminator 1)

--- 0x4008588f: _esp_error_check_failed at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/esp_err.c:49

file: "/IDF/myprojectsESP/NatureScan-SHT31-V2/NatureScan-SHT31/main/main.c" line 57
func: app_main
expression: SHT_WRITE(dev_handle, SHT31_I2C_WRITE_ADDR, SHT31_CMD_PERIODIC_HALF_M)

abort() was called at PC 0x4008588f on core 0
```



psachs Merge pull request #82 from Sensirion/migrate-to-github-workflow 9496777 · 10 months ago 179 Commits

.github	Migrate to github workflow	10 months ago
embedded-common @ 1ac7c72	Update embedded-common to 0.1.0 (#71)	5 years ago
sample-projects/shtc3-stm32-uvision	Fix images in release documentation (#59)	5 years ago
sht-common	Update clang-format style to the one used in embedded-co...	5 years ago
sht3x	Fix cpp check warnings	10 months ago
sht4x	SHT4x: Fix typo	5 years ago
shtc1	Fix cpp check warnings	10 months ago

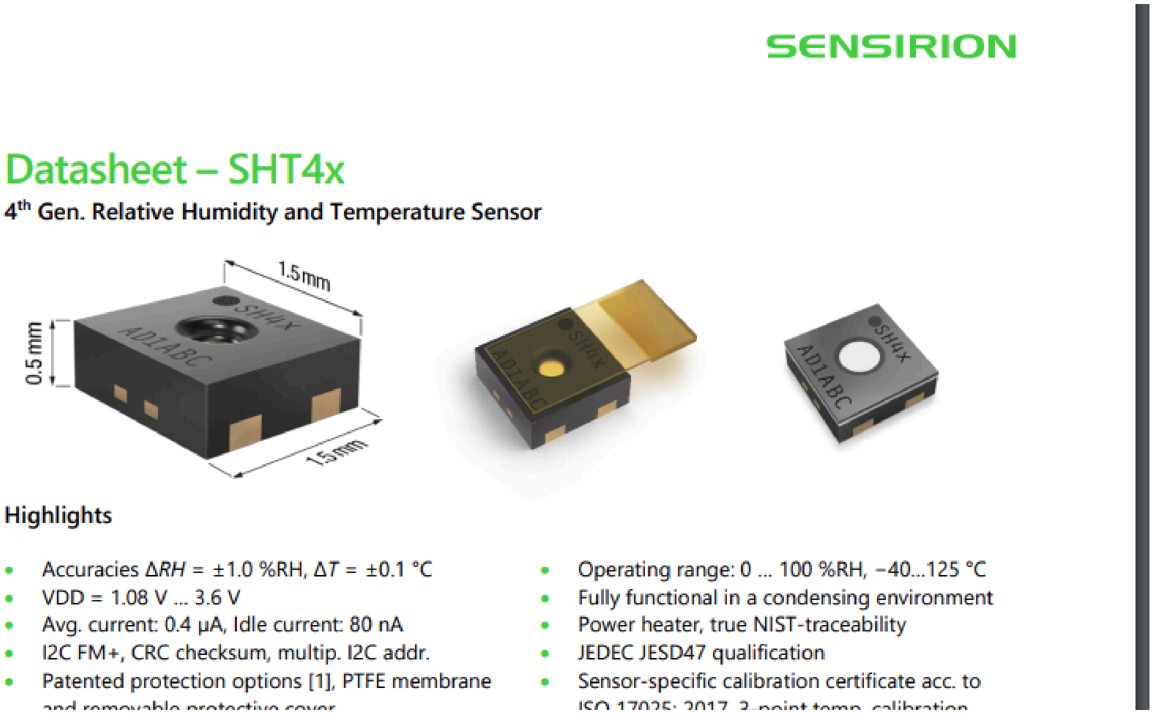
About

Embedded SHT Drivers for Sensirion Temperature and Humidity Sensors - Download the Zip Package from the Release Page

github.com/Sensirion/embedded-sht/rel...

raspberry-pi embedded i2c raspberrypi sensor driver temperature temperature-sensor humidity sht31 sht3x relative-humidity-sensors sensirion humidity-sens sht35 shtc3 shtw2 sht4x sensirion-embedded-drivers

Sensurion github repo



SENSIRION

Datasheet – SHT4x

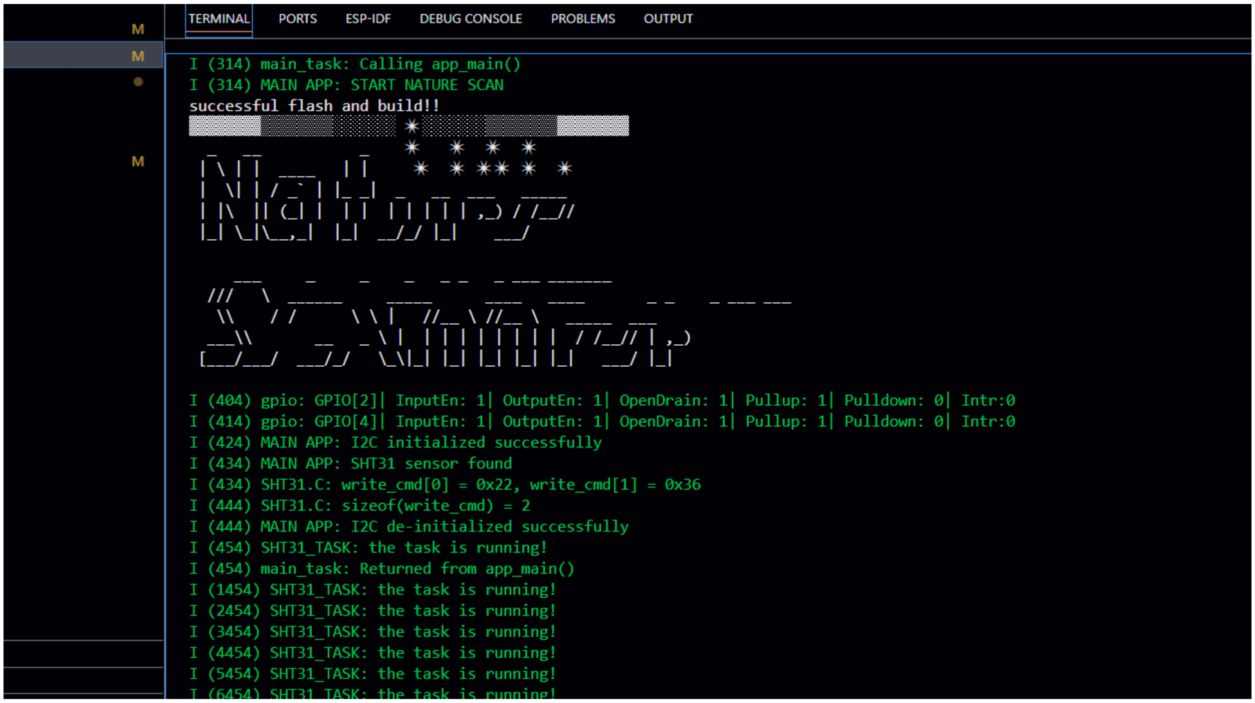
4th Gen. Relative Humidity and Temperature Sensor

1.5mm
0.5mm
1.5mm
SH4X
AD1ABC

Highlights

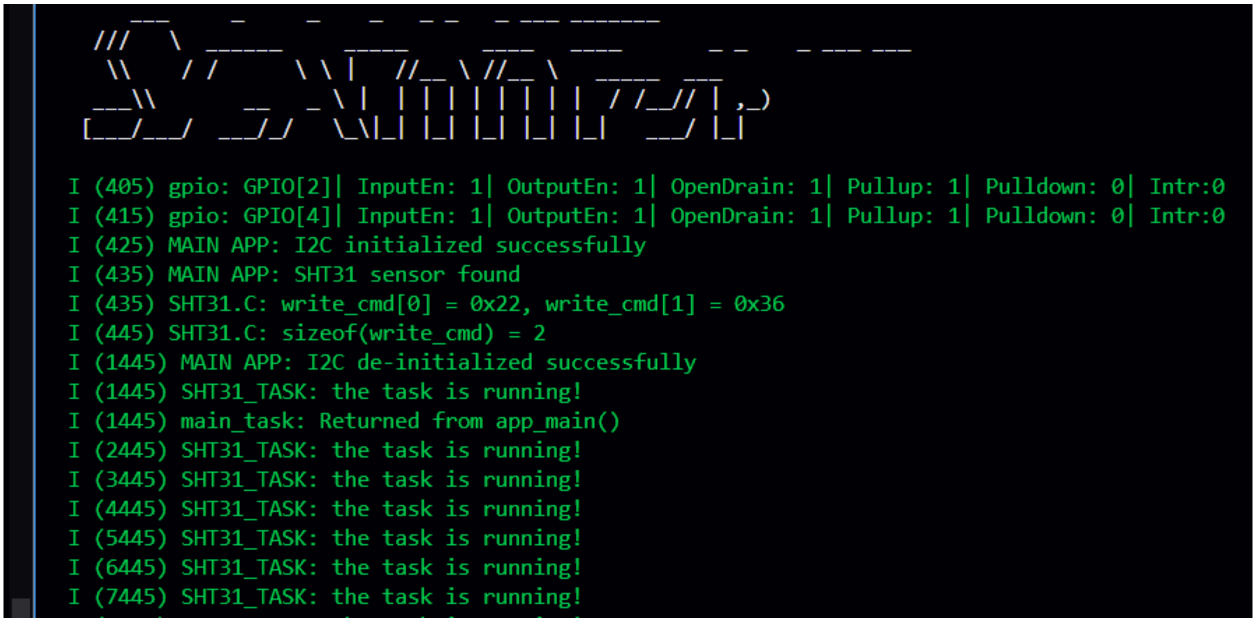
- Accuracies ΔRH = ±1.0 %RH, ΔT = ±0.1 °C
- VDD = 1.08 V ... 3.6 V
- Avg. current: 0.4 μA, Idle current: 80 nA
- I2C FM+, CRC checksum, multip. I2C addr.
- Patented protection options [1], PTFE membrane and removable protective cover
- Operating range: 0 ... 100 %RH, −40...125 °C
- Fully functional in a condensing environment
- Power heater, true NIST-traceability
- JEDEC JESD47 qualification
- Sensor-specific calibration certificate acc. to ISO 17025: 2017, 3-point temp. calibration

SHT4 datasheet



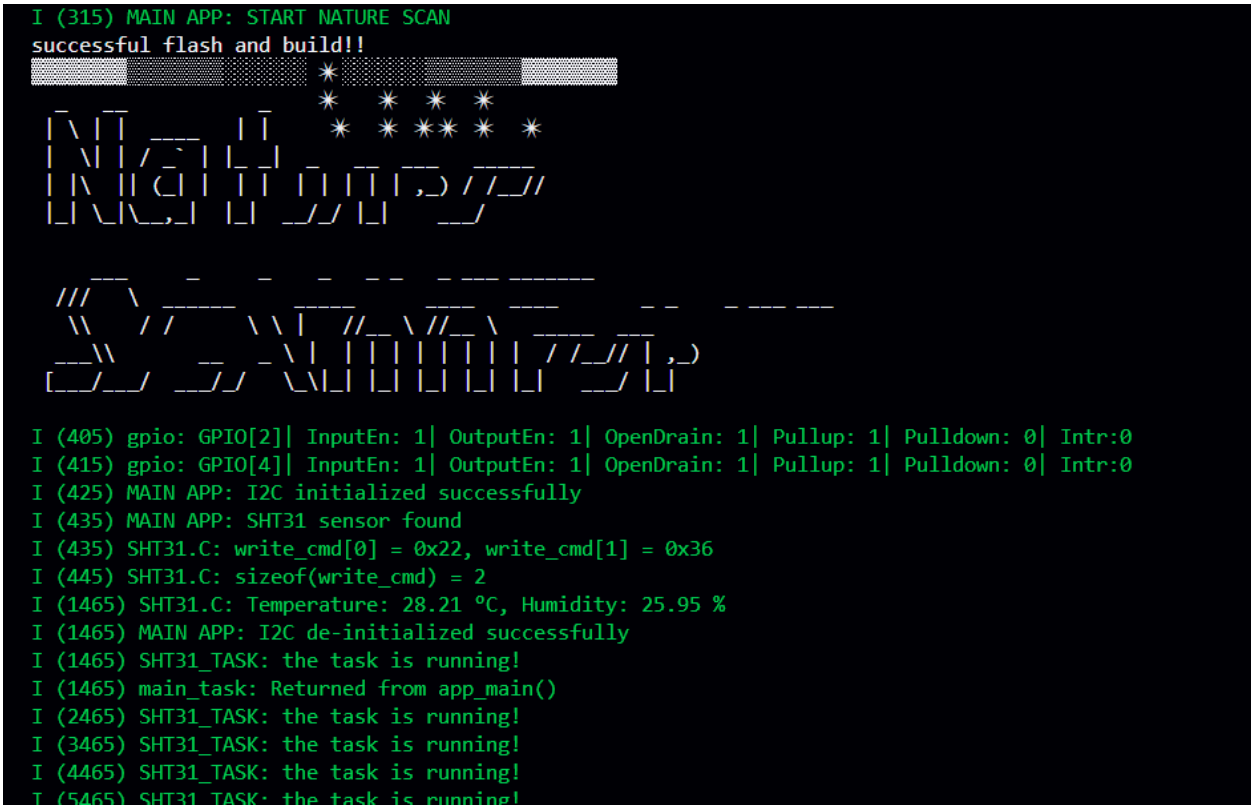
```
I (314) main_task: Calling app_main()
I (314) MAIN APP: START NATURE SCAN
successful flash and build!!
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ✳ ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
  _   __          _        ✳    ✳   ✳   ✳
 | \ | |  ____   | |        ✳    ✳  ✳✳  ✳    ✳
 |  \| | / _` | |_ _| _  __  ___   _____
 | |\  || (_| |  | |  | | | | | ,_) / /__//
 |_| \_|\__,_|  |_|  __/_/ |_|    ___/


    ___      _      _     _    _ _    _ ___ _______
  ///   \  ______    _____      ____  ____      _ _    _ ___ ___
   \\    / /      \ \ |  //__ \ //__ \   _____  ___
  ___\\       __   _ \ |  | | | | | | | |  / /__// | ,_)
 [___/___/  ___/_/   \_\|_| |_| |_| |_| |_|   ___/ |_|

I (404) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (414) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (424) MAIN APP: I2C initialized successfully
I (434) MAIN APP: SHT31 sensor found
I (434) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
I (444) SHT31.C: sizeof(write_cmd) = 2
I (444) MAIN APP: I2C de-initialized successfully
I (454) SHT31_TASK: the task is running!
I (454) main_task: Returned from app_main()
I (1454) SHT31_TASK: the task is running!
I (2454) SHT31_TASK: the task is running!
I (3454) SHT31_TASK: the task is running!
I (4454) SHT31_TASK: the task is running!
I (5454) SHT31_TASK: the task is running!
I (6454) SHT31_TASK: the task is running!
```

ALL GREEN FOR PERIODIC!



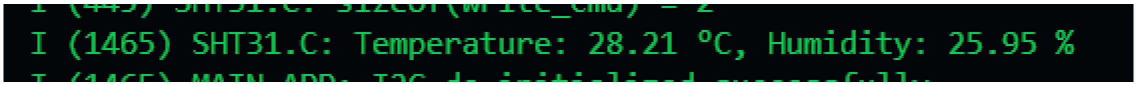
```
     ___     _     _    _   _ _   _ ___ _______
   ///   \  ______    _____     ____  ____      _ _   _ ___ ___
    \\    / /     \ \ |   //__ \ //__ \   _____ ___
   ___\\      __   _ \ |  | | | | | | | |  / /__// | ,_)
  [___/___/  ___/_/   \_\|_| |_| |_| |_| |_|   ___/ |_|

 I (405) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
 I (415) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
 I (425) MAIN APP: I2C initialized successfully
 I (435) MAIN APP: SHT31 sensor found
 I (435) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
 I (445) SHT31.C: sizeof(write_cmd) = 2
 I (1445) MAIN APP: I2C de-initialized successfully
 I (1445) SHT31_TASK: the task is running!
 I (1445) main_task: Returned from app_main()
 I (2445) SHT31_TASK: the task is running!
 I (3445) SHT31_TASK: the task is running!
 I (4445) SHT31_TASK: the task is running!
 I (5445) SHT31_TASK: the task is running!
 I (6445) SHT31_TASK: the task is running!
 I (7445) SHT31_TASK: the task is running!
```

ALL GREEN!!! Now we need to print..

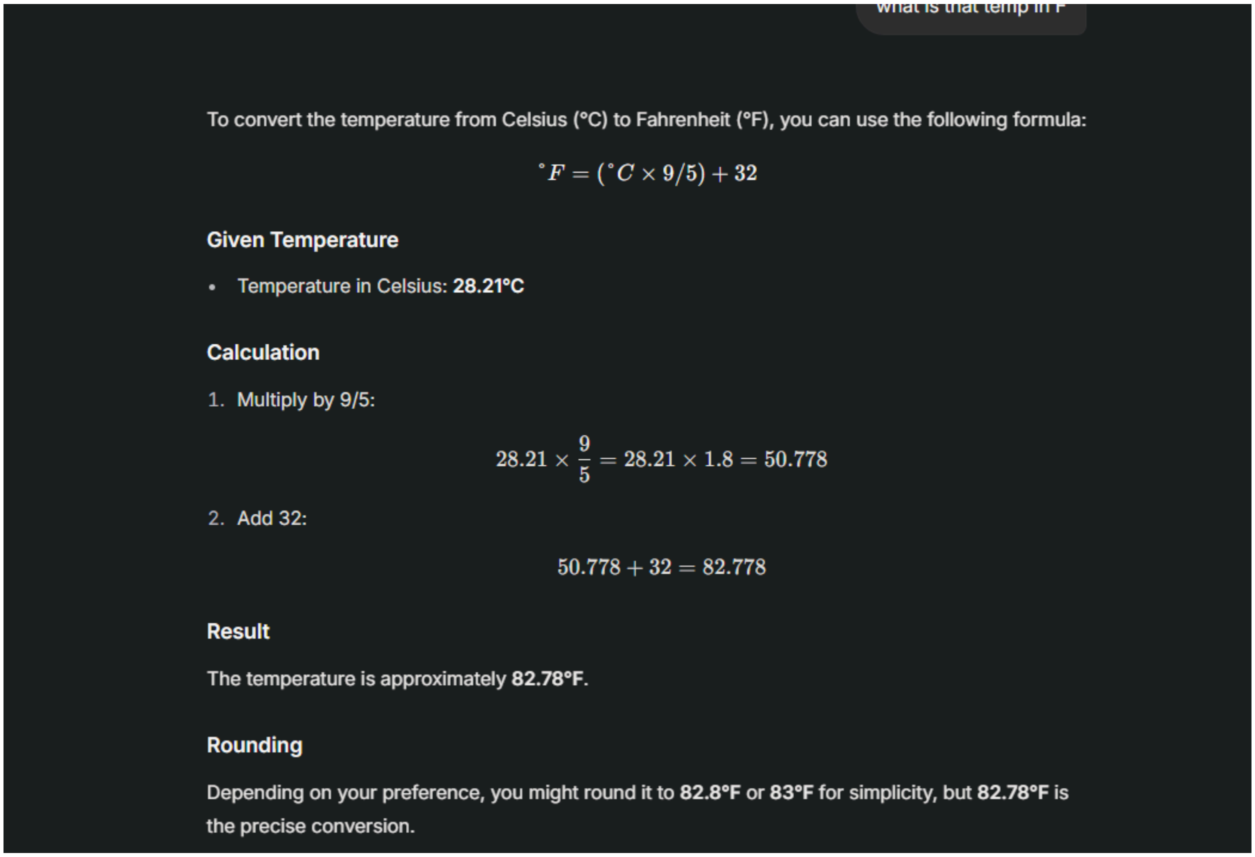


```
I (315) MAIN APP: START NATURE SCAN
successful flash and build!!
░░░░░▒▒▒▒▒▓▓▓▓ ✳ ▓▓▓▓▒▒▒▒▒░░░░░
  _  __              _      ✳   ✳   ✳   ✳
 | \ | |  ____    | |     ✳   ✳  ✳✳  ✳    ✳
 |  \| | / _` | |_ _|  _  __  ___  _____
 | |\  || (_| |  | |  | | | | | ,_) / /__//
 |_| \_|\__,_|  |_|  __/_/ |_|    ___/

     ___     _     _    _   _ _   _ ___ _______
   ///   \  ______    _____     ____  ____      _ _   _ ___ ___
    \\    / /     \ \ |   //__ \ //__ \   _____ ___
   ___\\      __   _ \ |  | | | | | | | |  / /__// | ,_)
  [___/___/  ___/_/   \_\|_| |_| |_| |_| |_|   ___/ |_|

I (405) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (415) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (425) MAIN APP: I2C initialized successfully
I (435) MAIN APP: SHT31 sensor found
I (435) SHT31.C: write_cmd[0] = 0x22, write_cmd[1] = 0x36
I (445) SHT31.C: sizeof(write_cmd) = 2
I (1465) SHT31.C: Temperature: 28.21 °C, Humidity: 25.95 %
I (1465) MAIN APP: I2C de-initialized successfully
I (1465) SHT31_TASK: the task is running!
I (1465) main_task: Returned from app_main()
I (2465) SHT31_TASK: the task is running!
I (3465) SHT31_TASK: the task is running!
I (4465) SHT31_TASK: the task is running!
I (5465) SHT31_TASK: the task is running!
```

FEBRUARY 27TH 2025 11:22 PM
TEMPE, WE HAVE TEMPERATURE



```
I (1465) SHT31.C: Temperature: 28.21 °C, Humidity: 25.95 %
```



what is that temp in F

To convert the temperature from Celsius (°C) to Fahrenheit (°F), you can use the following formula:

$$^{\circ}F = (^{\circ}C \times 9/5) + 32$$

Given Temperature

- Temperature in Celsius: **28.21°C**

Calculation

1. Multiply by 9/5:

$$28.21 \times \frac{9}{5} = 28.21 \times 1.8 = 50.778$$

2. Add 32:

$$50.778 + 32 = 82.778$$

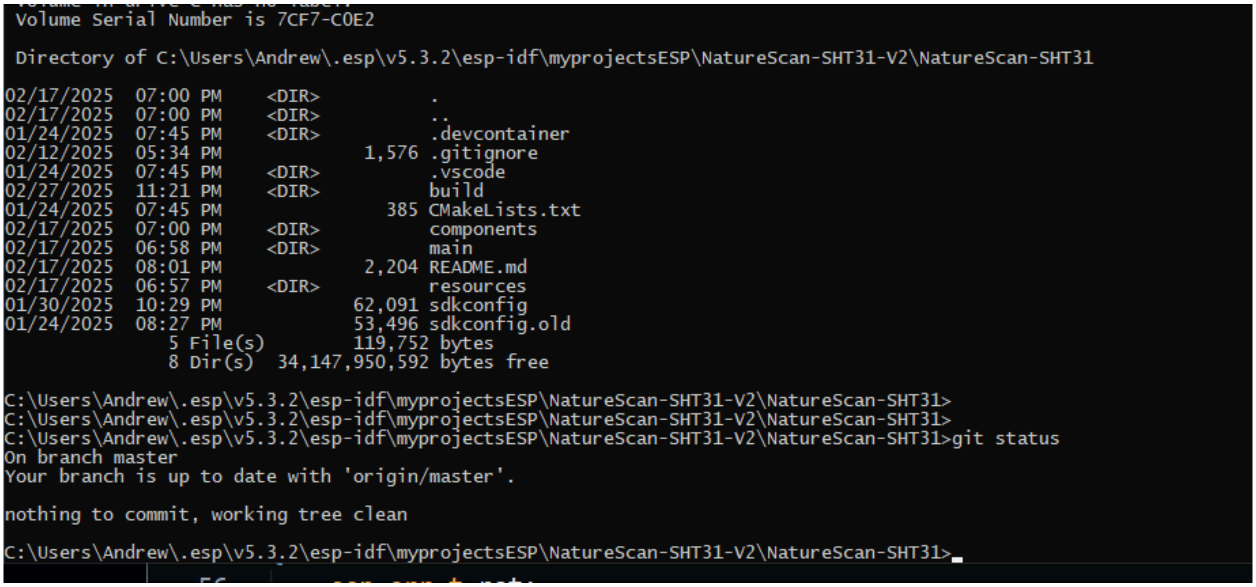
Result

The temperature is approximately **82.78°F**.

Rounding

Depending on your preference, you might round it to **82.8°F** or **83°F** for simplicity, but **82.78°F** is the precise conversion.

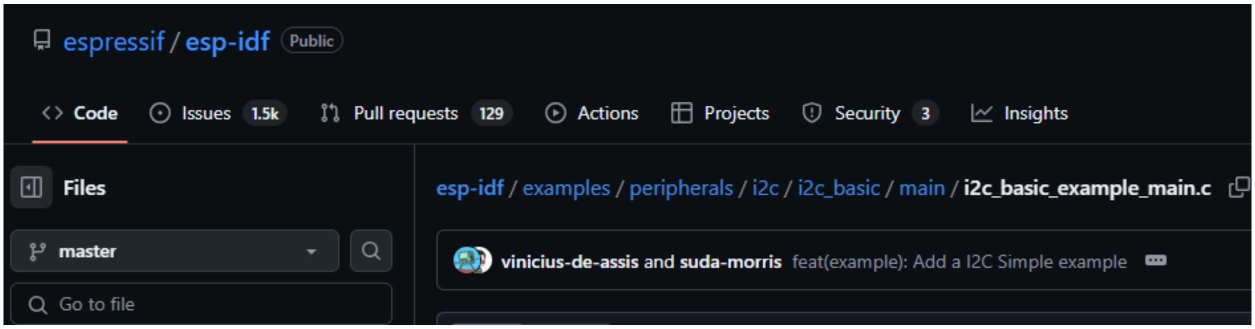
82.778 deg F



```
 Volume Serial Number is 7CF7-C0E2

 Directory of C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31

02/17/2025  07:00 PM    <DIR>          .
02/17/2025  07:00 PM    <DIR>          ..
01/24/2025  07:45 PM    <DIR>          .devcontainer
02/12/2025  05:34 PM             1,576 .gitignore
01/24/2025  07:45 PM    <DIR>          .vscode
02/27/2025  11:21 PM    <DIR>          build
01/24/2025  07:45 PM               385 CMakeLists.txt
02/17/2025  07:00 PM    <DIR>          components
02/17/2025  06:58 PM    <DIR>          main
02/17/2025  08:01 PM             2,204 README.md
02/17/2025  06:57 PM    <DIR>          resources
01/30/2025  10:29 PM            62,091 sdkconfig
01/24/2025  08:27 PM            53,496 sdkconfig.old
               5 File(s)        119,752 bytes
               8 Dir(s)  34,147,950,592 bytes free

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
```

2/25/2025

Put the initialization function logic in the sht31.c source file.



esp-idf / examples / peripherals / i2c / i2c_basic / main / i2c_basic_example_main.c

vinicius-de-assis and suda-morris feat(example): Add a I2C Simple example

Use this as a 1 file template. You can “break” it into 3 files with a chatbot

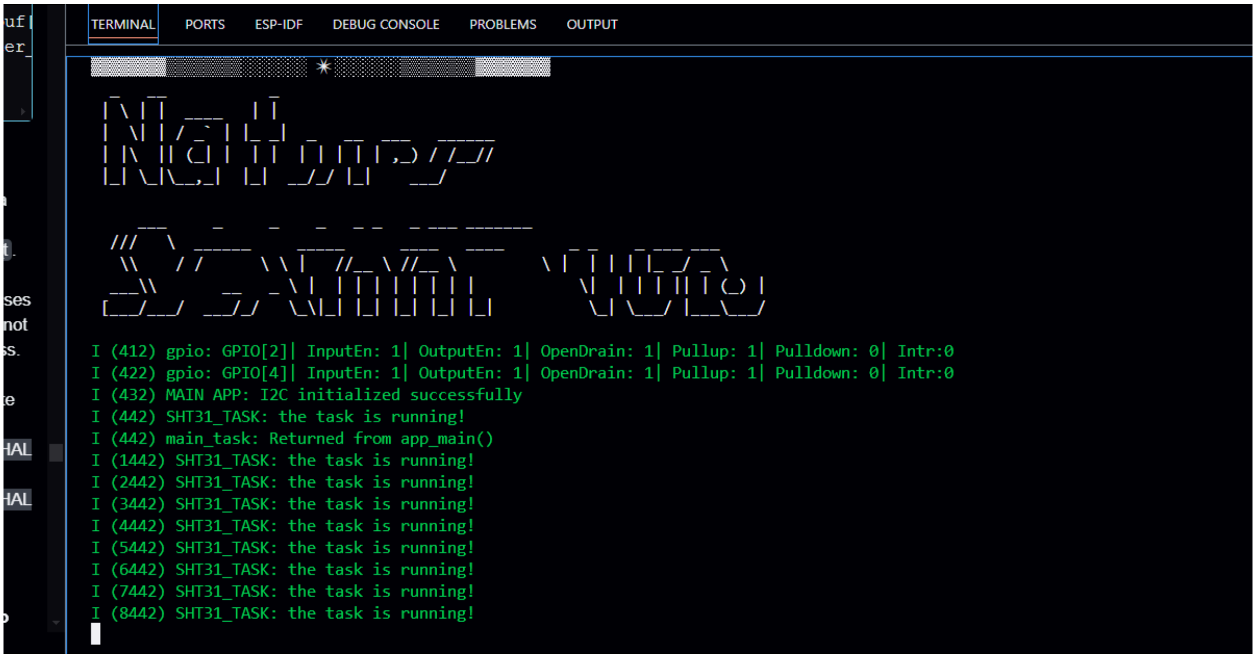
i2c.c (Source File)

This file contains the implementations of the functions declared in i2c.h. It includes i2c.h to access the declarations.

mpu9250_register_read
mpu9250_register_write_byte
i2c_master_init

3 functions

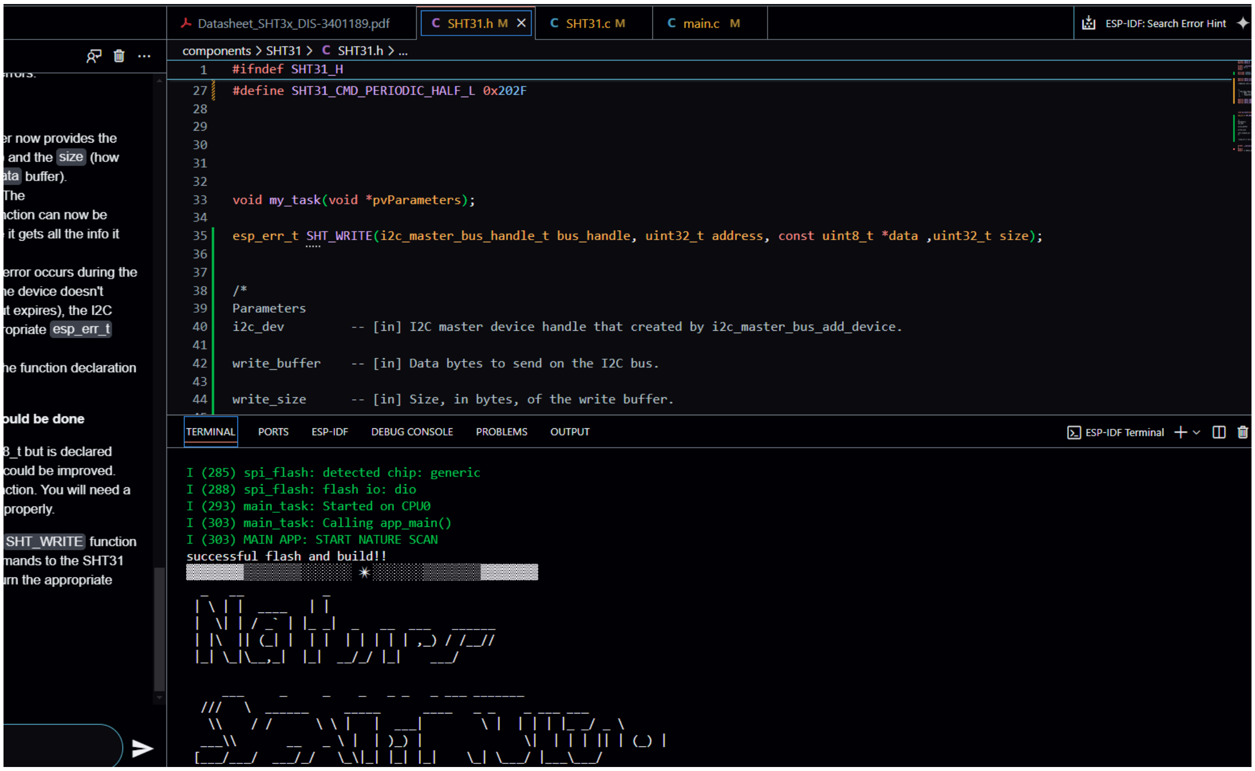
The write function will be i2c_master_transmit
Teh read will be i2c_master_transmit_receive



```
I (412) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (422) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (432) MAIN APP: I2C initialized successfully
I (442) SHT31_TASK: the task is running!
I (442) main_task: Returned from app_main()
I (1442) SHT31_TASK: the task is running!
I (2442) SHT31_TASK: the task is running!
I (3442) SHT31_TASK: the task is running!
I (4442) SHT31_TASK: the task is running!
I (5442) SHT31_TASK: the task is running!
I (6442) SHT31_TASK: the task is running!
I (7442) SHT31_TASK: the task is running!
I (8442) SHT31_TASK: the task is running!
```

Initialized

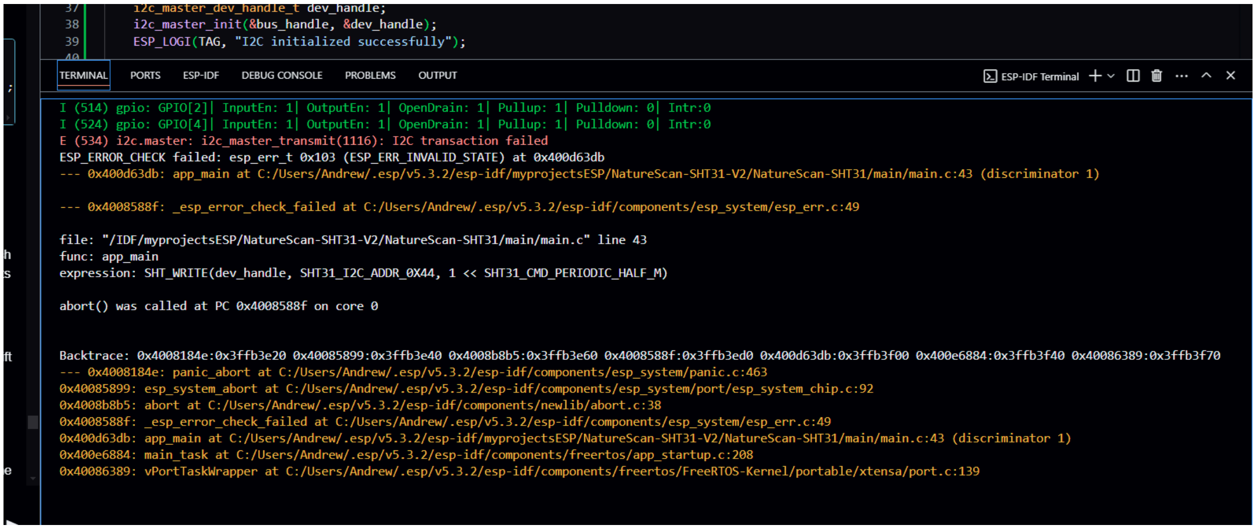
2/24/2025



```c
#ifndef SHT31_H
#define SHT31_CMD_PERIODIC_HALF_L 0x202F





void my_task(void *pvParameters);

esp_err_t SHT_WRITE(i2c_master_bus_handle_t bus_handle, uint32_t address, const uint8_t *data ,uint32_t size);


/*
Parameters
i2c_dev         -- [in] I2C master device handle that created by i2c_master_bus_add_device.

write_buffer    -- [in] Data bytes to send on the I2C bus.

write_size      -- [in] Size, in bytes, of the write buffer.
```

```
I (285) spi_flash: detected chip: generic
I (288) spi_flash: flash io: dio
I (293) main_task: Started on CPU0
I (303) main_task: Calling app_main()
I (303) MAIN APP: START NATURE SCAN
successful flash and build!!
```

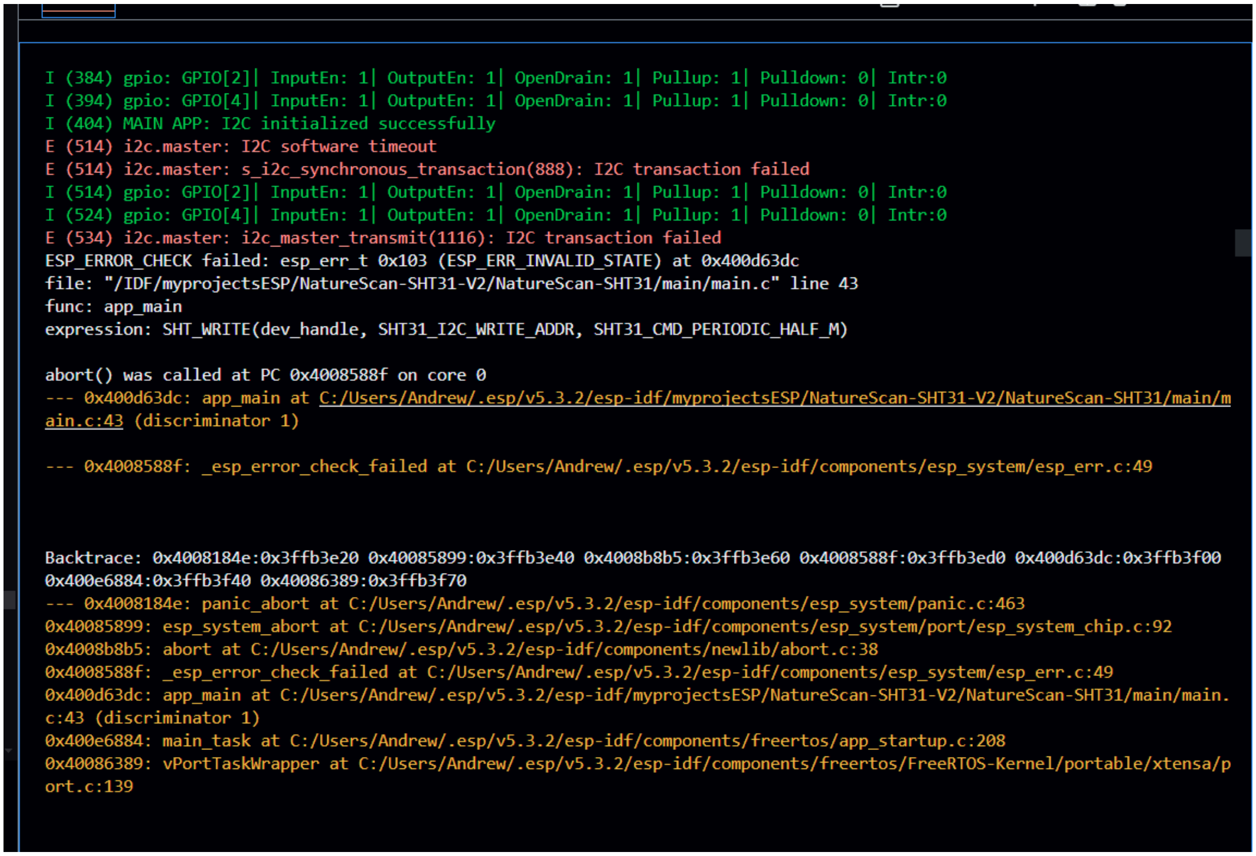
Successful prototype



```c
    i2c_master_dev_handle_t dev_handle;
    i2c_master_init(&bus_handle, &dev_handle);
    ESP_LOGI(TAG, "I2C initialized successfully");
```

```
I (514) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (524) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
E (534) i2c.master: i2c_master_transmit(1116): I2C transaction failed
ESP_ERROR_CHECK failed: esp_err_t 0x103 (ESP_ERR_INVALID_STATE) at 0x400d63db
--- 0x400d63db: app_main at C:/Users/Andrew/.esp/v5.3.2/esp-idf/myprojectsESP/NatureScan-SHT31-V2/NatureScan-SHT31/main/main.c:43 (discriminator 1)

--- 0x4008588f: _esp_error_check_failed at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/esp_err.c:49

file: "/IDF/myprojectsESP/NatureScan-SHT31-V2/NatureScan-SHT31/main/main.c" line 43
func: app_main
expression: SHT_WRITE(dev_handle, SHT31_I2C_ADDR_0X44, 1 << SHT31_CMD_PERIODIC_HALF_M)

abort() was called at PC 0x4008588f on core 0


Backtrace: 0x4008184e:0x3ffb3e20 0x40085899:0x3ffb3e40 0x4008b8b5:0x3ffb3e60 0x4008588f:0x3ffb3ed0 0x400d63db:0x3ffb3f00 0x400e6884:0x3ffb3f40 0x40086389:0x3ffb3f70
--- 0x4008184e: panic_abort at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/panic.c:463
0x40085899: esp_system_abort at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/port/esp_system_chip.c:92
0x4008b8b5: abort at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/newlib/abort.c:38
0x4008588f: _esp_error_check_failed at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/esp_err.c:49
0x400d63db: app_main at C:/Users/Andrew/.esp/v5.3.2/esp-idf/myprojectsESP/NatureScan-SHT31-V2/NatureScan-SHT31/main/main.c:43 (discriminator 1)
0x400e6884: main_task at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/freertos/app_startup.c:208
0x40086389: vPortTaskWrapper at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/freertos/FreeRTOS-Kernel/portable/xtensa/port.c:139
```

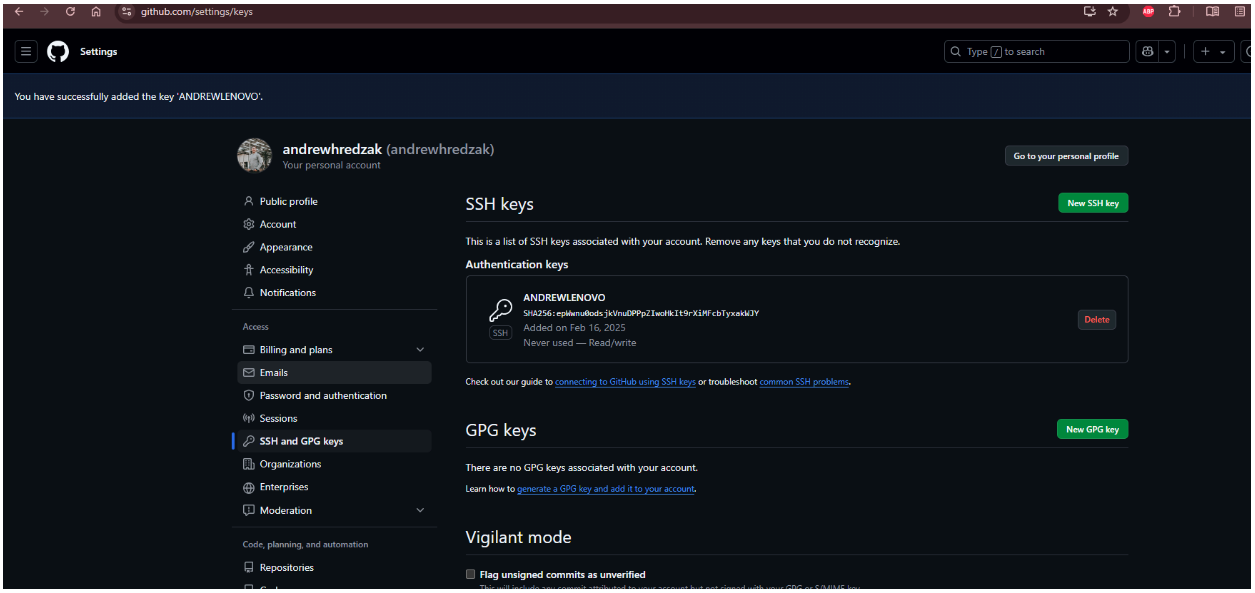
Successful build and flash now were back to this



```
I (384) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (394) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (404) MAIN APP: I2C initialized successfully
E (514) i2c.master: I2C software timeout
E (514) i2c.master: s_i2c_synchronous_transaction(888): I2C transaction failed
I (514) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (524) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
E (534) i2c.master: i2c_master_transmit(1116): I2C transaction failed
ESP_ERROR_CHECK failed: esp_err_t 0x103 (ESP_ERR_INVALID_STATE) at 0x400d63dc
file: "/IDF/myprojectsESP/NatureScan-SHT31-V2/NatureScan-SHT31/main/main.c" line 43
func: app_main
expression: SHT_WRITE(dev_handle, SHT31_I2C_WRITE_ADDR, SHT31_CMD_PERIODIC_HALF_M)

abort() was called at PC 0x4008588f on core 0
--- 0x400d63dc: app_main at C:/Users/Andrew/.esp/v5.3.2/esp-idf/myprojectsESP/NatureScan-SHT31-V2/NatureScan-SHT31/main/m
ain.c:43 (discriminator 1)

--- 0x4008588f: _esp_error_check_failed at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/esp_err.c:49



Backtrace: 0x4008184e:0x3ffb3e20 0x40085899:0x3ffb3e40 0x4008b8b5:0x3ffb3e60 0x4008588f:0x3ffb3ed0 0x400d63dc:0x3ffb3f00
0x400e6884:0x3ffb3f40 0x40086389:0x3ffb3f70
--- 0x4008184e: panic_abort at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/panic.c:463
0x40085899: esp_system_abort at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/port/esp_system_chip.c:92
0x4008b8b5: abort at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/newlib/abort.c:38
0x4008588f: _esp_error_check_failed at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/esp_system/esp_err.c:49
0x400d63dc: app_main at C:/Users/Andrew/.esp/v5.3.2/esp-idf/myprojectsESP/NatureScan-SHT31-V2/NatureScan-SHT31/main/main.
c:43 (discriminator 1)
0x400e6884: main_task at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/freertos/app_startup.c:208
0x40086389: vPortTaskWrapper at C:/Users/Andrew/.esp/v5.3.2/esp-idf/components/freertos/FreeRTOS-Kernel/portable/xtensa/p
ort.c:139
```

2/17/2025

2/16/2025



You have successfully added the key 'ANDREWLENOVO'.

andrewhredzak (andrewhredzak)
Your personal account

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

ANDREWLENOVO
`SHA256:epWwnu0odsjkVnuDPPpZIwoHkIt9rXiMFcbTyxakWJY`
Added on Feb 16, 2025
Never used — Read/write

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

GPG keys

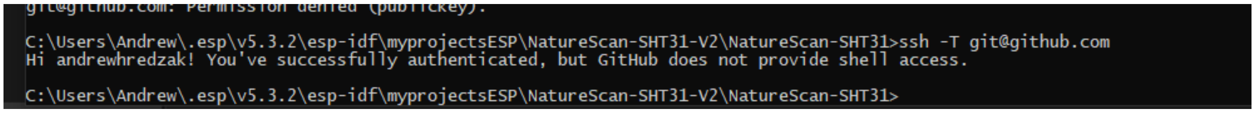
There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

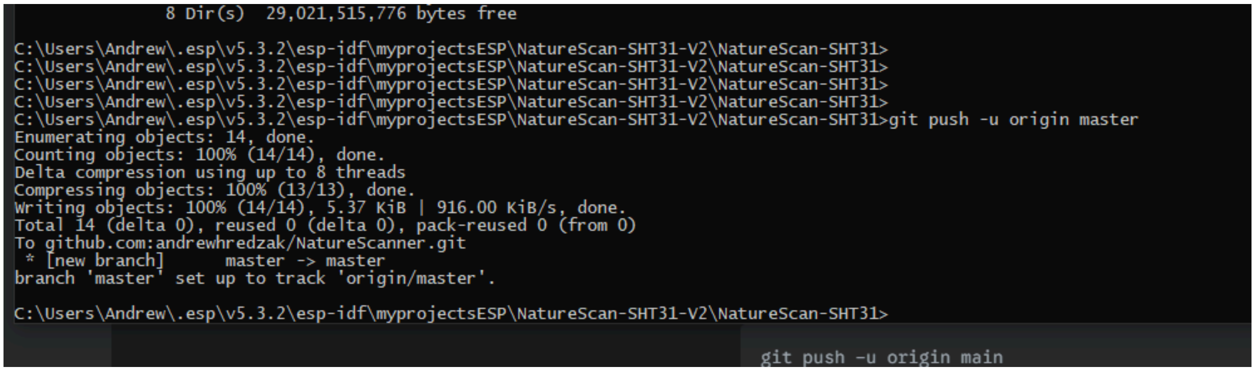
Vigilant mode

☐ **Flag unsigned commits as unverified**

Ssh added to github to link my computer to this repo

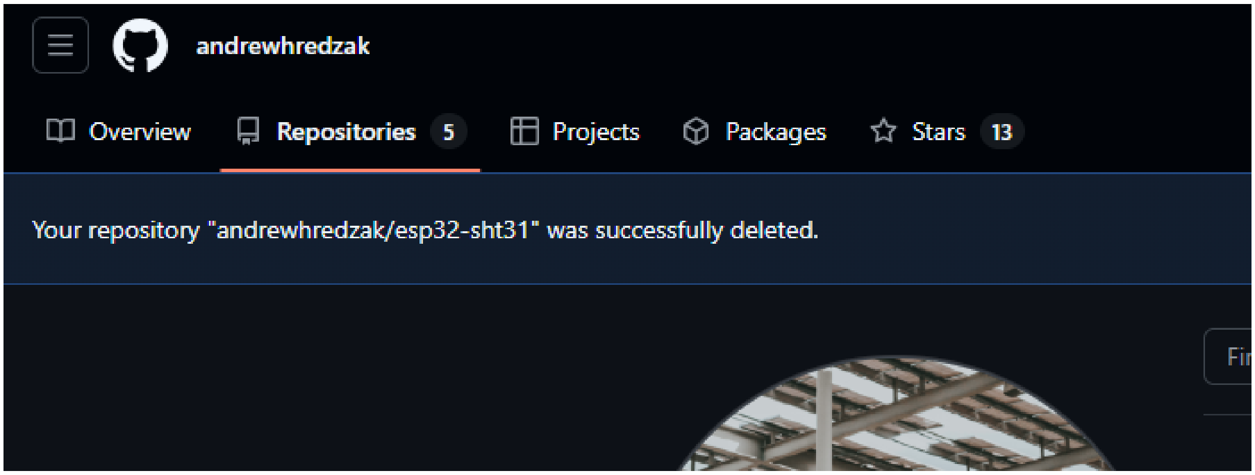


```
git@github.com: Permission denied (publickey).

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>ssh -T git@github.com
Hi andrewhredzak! You've successfully authenticated, but GitHub does not provide shell access.

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
```



```
               8 Dir(s)  29,021,515,776 bytes free

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>git push -u origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (14/14), 5.37 KiB | 916.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:andrewhredzak/NatureScanner.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31>
```

```
git push -u origin main
```

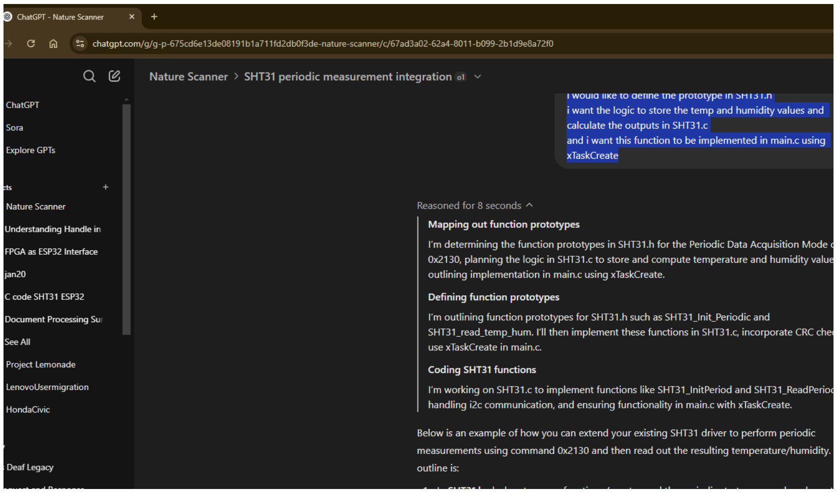


Your repository "andrewhredzak/esp32-sht31" was successfully deleted.

Deleted

2/12/2025

this is my project directory: NATURESCAN-SHT31/ - CMakeLists.txt - sdkconfig - components/ - SHT31/ - CMakeLists.txt - Kconfig - SHT31.c - SHT31.h - main/ - CMakeLists.txt - main.c - Kconfig.projbuild I already have i2c_master_dev_init configured. i would like to start with a basic measurement command from SHT31: Periodic Data Acquisition Mode the command is : 0x2130 i would like to define the prototype in SHT31.h i want the logic to store the temp and humidity values and calculate the outputs in SHT31.c and i want this function to be implemented in main.c using xTaskCreate



i would like to define the prototype in SHT31.h
i want the logic to store the temp and humidity values and calculate the outputs in SHT31.c
and i want this function to be implemented in main.c using xTaskCreate

Reasoned for 8 seconds

Mapping out function prototypes

I'm determining the function prototypes in SHT31.h for the Periodic Data Acquisition Mode 0x2130, planning the logic in SHT31.c to store and compute temperature and humidity value outlining implementation in main.c using xTaskCreate.

Defining function prototypes

I'm outlining function prototypes for SHT31.h such as SHT31_Init_Periodic and SHT31_read_temp_hum. I'll then implement these functions in SHT31.c, incorporate CRC che use xTaskCreate in main.c.

Coding SHT31 functions

I'm working on SHT31.c to implement functions like SHT31_InitPeriod and SHT31_ReadPerio handling i2c communication, and ensuring functionality in main.c with xTaskCreate.

Below is an example of how you can extend your existing SHT31 driver to perform periodic measurements using command 0x2130 and then read out the resulting temperature/humidity. outline is:

Link

Next tasks:

- ☑ ~~Write a basic print function but develop the prototype in header, "logic" in source, then call it in main~~
- ☐ Understand all SHT31 commands
- ☐ Come up with a name for the main data acquisition function
    - ☐ SHT_data_acquisition()
        - ☐ Period
        - ☐ One shot
- ☐ Group commands and start to think how they would be laid out in the header file

Review:
What is a handle
Special operators
`i2c_device_config_t` is how the ESP32 (as a master) sees other slave devices on the bus,
`i2c_slave_config_t` is how you set up the ESP32 to *become* that slave device.

Push to github:

2/2/2025

Next question:
What should periodic do?
I think it should write then read

Because it writes the periodic command to the sht31 reg and then the device digital interface continually keeps writing to the data register. Then it should sent it over to the controller

```
    esp_err_t err = i2c_master_bus_add_device(I2C_NUM_0, &dev_cfg, &dev_handle);
```

I2c master driver major changes in concepts

Link to esp idf

2/1/2025

what is a handle?

why does dev_config need to be a constant and bus_handle doesn't?
why does dev_config and ret_handle need to be pointers?

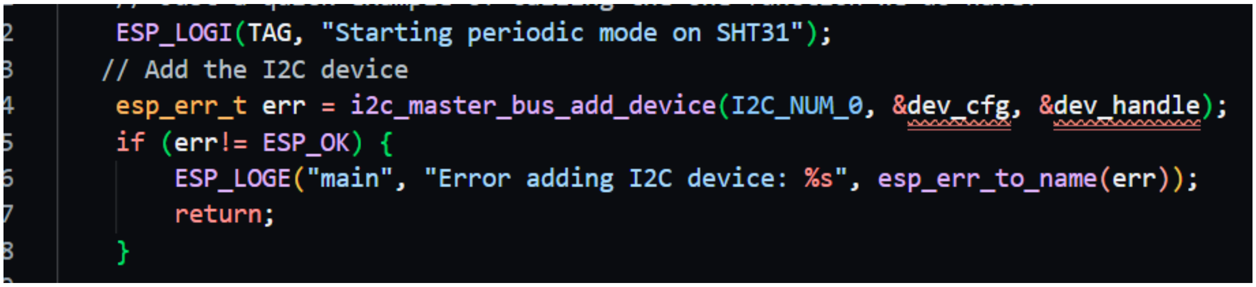
what is a pointer?
What is a type definition?

operators

Operator Type	Description	Operators
Arithmetic Operators	Perform mathematical calculations	+, -, *, /, %
Relational Operators	Compare values	==, !=, >, <, >=, <=
Logical Operators	Combine logical conditions	&&, `
Bitwise Operators	Manipulate individual bits	&, `
Assignment Operators	Assign values	=, +=, -=, *=, /=, %=
Increment and Decrement Operators	Increase or decrease values	++, --
Conditional Operator	Evaluate a condition and return a value	?:
Special Operators	Perform specific operations	sizeof, &, *, ->, .

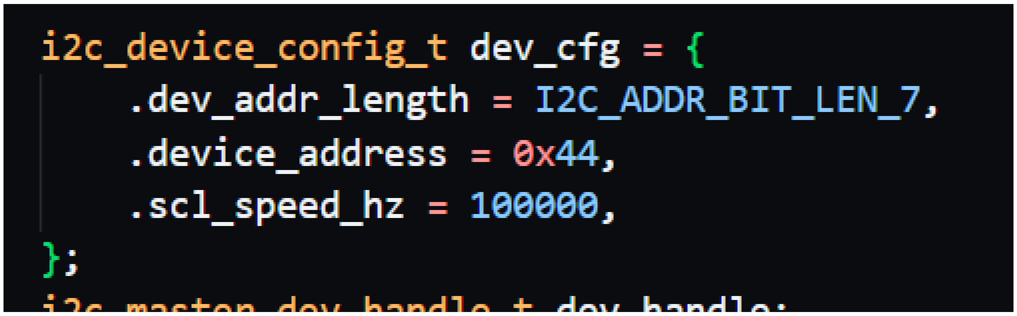
Examples of special operators

&: returns memory address



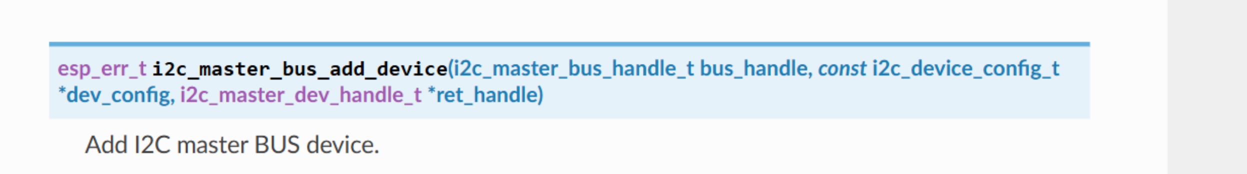
```
    ESP_LOGI(TAG, "Starting periodic mode on SHT31");
   // Add the I2C device
    esp_err_t err = i2c_master_bus_add_device(I2C_NUM_0, &dev_cfg, &dev_handle);
    if (err!= ESP_OK) {
        ESP_LOGE("main", "Error adding I2C device: %s", esp_err_to_name(err));
        return;
    }
```

. accesses a member of a structure or union directly



```c
i2c_device_config_t dev_cfg = {
    .dev_addr_length = I2C_ADDR_BIT_LEN_7,
    .device_address = 0x44,
    .scl_speed_hz = 100000,
};
i2c_master_dev_handle_t dev_handle;
```

“*” dereferences a pointer, accessing the value stored at the memory address pointed to by the pointer

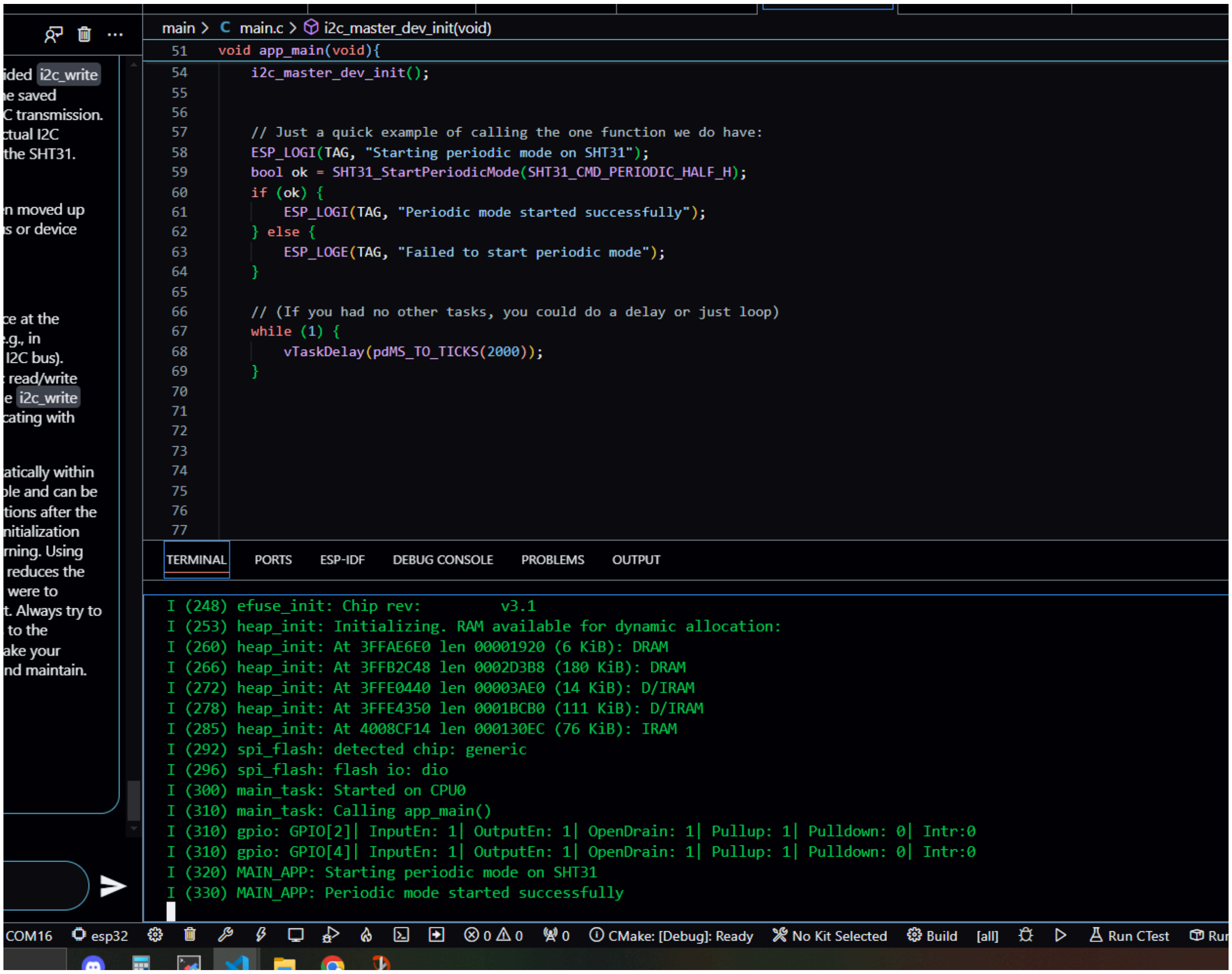


esp_err_t **i2c_master_bus_add_device**(i2c_master_bus_handle_t bus_handle, *const* i2c_device_config_t *dev_config, i2c_master_dev_handle_t *ret_handle)

Add I2C master BUS device.

1/31/2025

write ---> i2c_master_transmit()
read ---> i2c_master_receive()
read and write ---> i2c_master_transmit_receive()



ided i2c_write
e saved
C transmission.
tual I2C
the SHT31.

n moved up
s or device

e at the
.g., in
I2C bus).
read/write
e i2c_write
ating with

atically within
le and can be
tions after the
nitialization
ning. Using
reduces the
were to
t. Always try to
to the
ake your
nd maintain.

```c
void app_main(void){
    i2c_master_dev_init();


    // Just a quick example of calling the one function we do have:
    ESP_LOGI(TAG, "Starting periodic mode on SHT31");
    bool ok = SHT31_StartPeriodicMode(SHT31_CMD_PERIODIC_HALF_H);
    if (ok) {
        ESP_LOGI(TAG, "Periodic mode started successfully");
    } else {
        ESP_LOGE(TAG, "Failed to start periodic mode");
    }

    // (If you had no other tasks, you could do a delay or just loop)
    while (1) {
        vTaskDelay(pdMS_TO_TICKS(2000));
    }
```

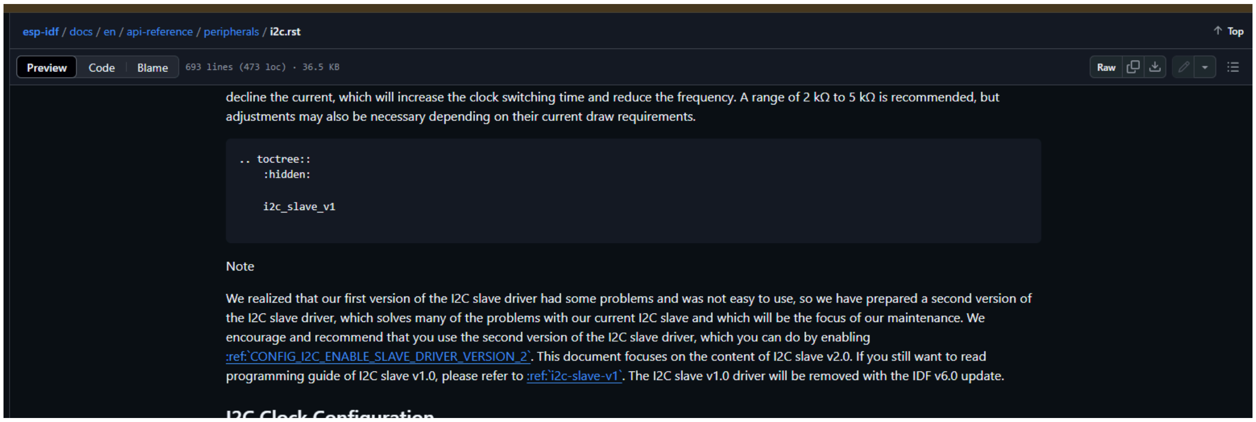
```
I (248) efuse_init: Chip rev:         v3.1
I (253) heap_init: Initializing. RAM available for dynamic allocation:
I (260) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (266) heap_init: At 3FFB2C48 len 0002D3B8 (180 KiB): DRAM
I (272) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (278) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (285) heap_init: At 4008CF14 len 000130EC (76 KiB): IRAM
I (292) spi_flash: detected chip: generic
I (296) spi_flash: flash io: dio
I (300) main_task: Started on CPU0
I (310) main_task: Calling app_main()
I (310) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (310) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (320) MAIN_APP: Starting periodic mode on SHT31
I (330) MAIN_APP: Periodic mode started successfully
```

Built and flashed. We got to print!

1/30/2025

Hi Gemini!!

[I2C master rst document]


esp-idf / docs / en / api-reference / peripherals / i2c.rst
Top
Preview
Code
Blame
693 lines (473 loc) · 36.5 KB
Raw

decline the current, which will increase the clock switching time and reduce the frequency. A range of 2 kΩ to 5 kΩ is recommended, but adjustments may also be necessary depending on their current draw requirements.

```
.. toctree::
    :hidden:

    i2c_slave_v1
```

Note

We realized that our first version of the I2C slave driver had some problems and was not easy to use, so we have prepared a second version of the I2C slave driver, which solves many of the problems with our current I2C slave and which will be the focus of our maintenance. We encourage and recommend that you use the second version of the I2C slave driver, which you can do by enabling :ref:`CONFIG_I2C_ENABLE_SLAVE_DRIVER_VERSION_2`. This document focuses on the content of I2C slave v2.0. If you still want to read programming guide of I2C slave v1.0, please refer to :ref:`i2c-slave-v1`. The I2C slave v1.0 driver will be removed with the IDF v6.0 update.

I2C Clock Configuration

1/28/2025

esp_err_t

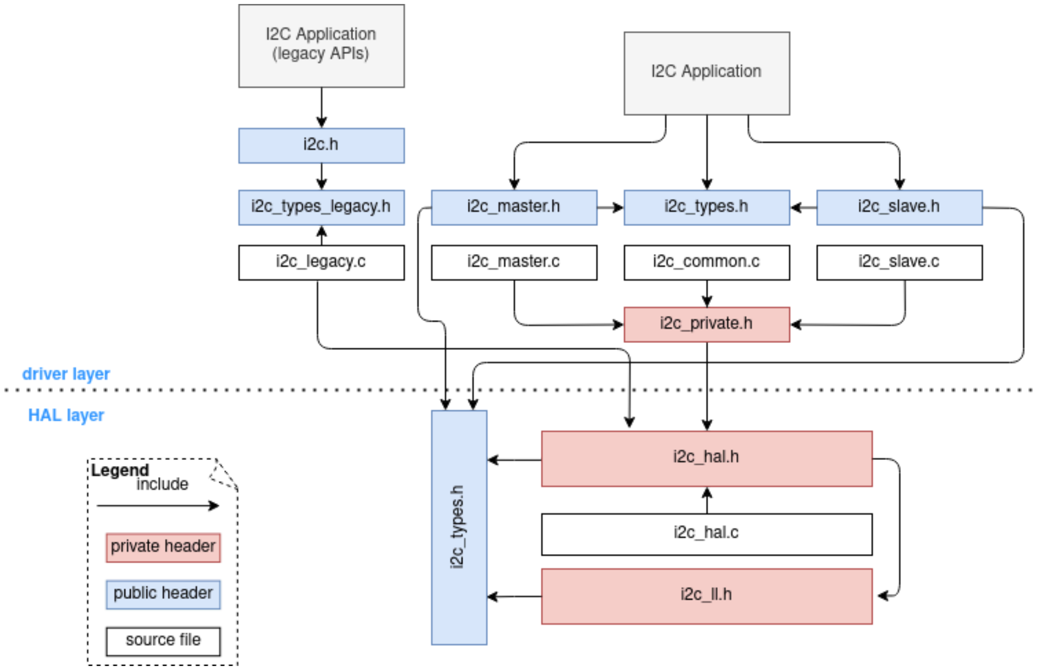
`esp_err_t` is a data type used to represent error codes returned by ESP32 functions

Common `esp_err_t` Values

- `ESP_OK`: Indicates that the function executed successfully without any errors.
- `ESP_ERR_INVALID_ARG`: Signals that an invalid argument was passed to the function.
- `ESP_ERR_NO_MEM`: Means that the function failed to allocate memory.
- `ESP_ERR_NOT_FOUND`: Indicates that a requested resource or device was not found.
- `ESP_FAIL`: Represents a generic error condition.

1/27/2025

Next steps are
read the I2C master driver, documentation programming guide ✅
Then Reed the three files eeprom example✅



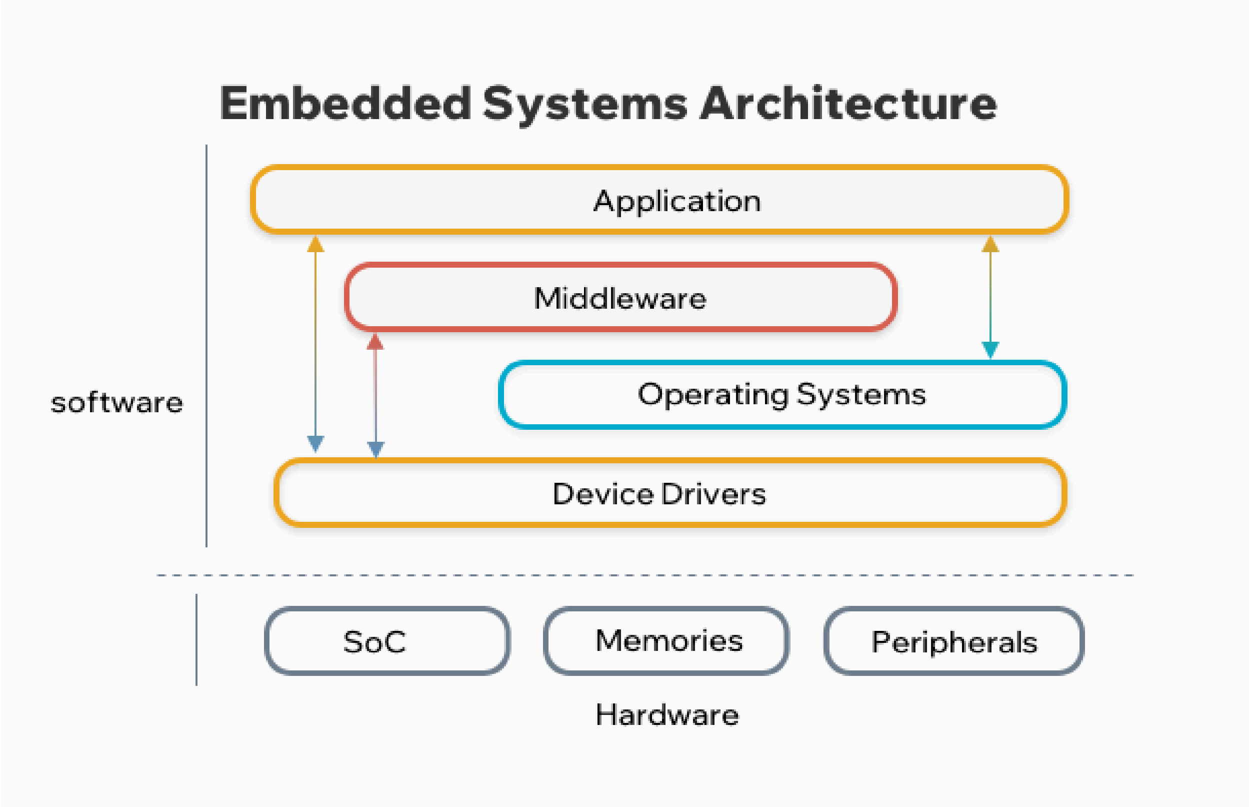
ESP32 v5.4 Inter-Integrated Circuit (I2C) programming guide

HAL: hardware abstraction layer

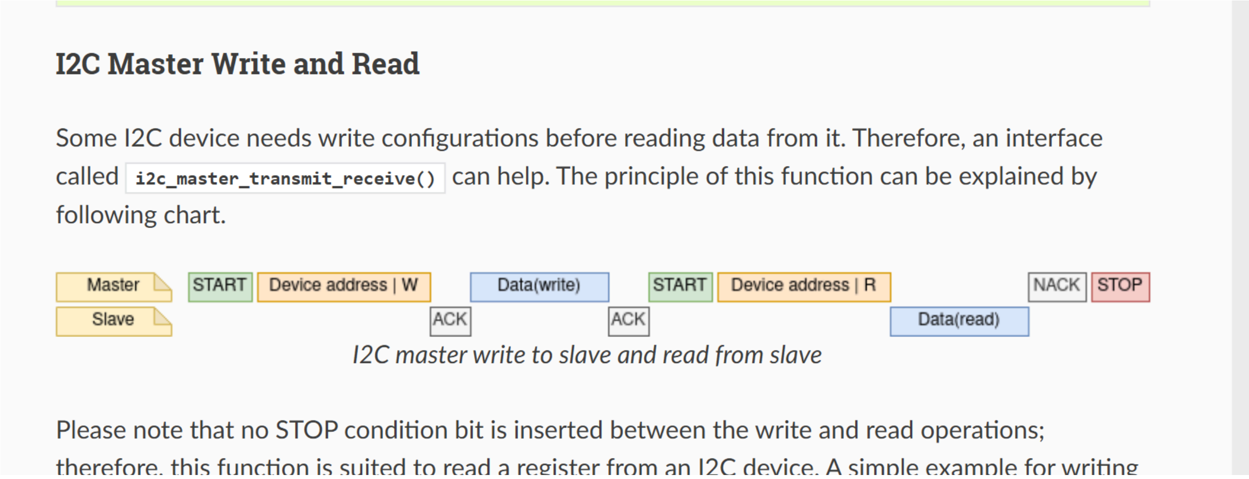
Hardware layers

Layer	**Description**	**Example in ESP32/SHT31 context**
Physical Layer	The actual hardware components.	ESP32 microcontroller, SHT31 sensor, I2C bus, resistors, etc.
Hardware Abstraction Layer (HAL)	Abstracts the specifics of the hardware, providing a standardized interface.	ESP32's I2C HAL provided by the ESP-IDF SDK.
Driver Layer	Communicates directly with the sensor using the HAL.	A driver specifically for the SHT31 sensor, or a generic I2C device driver.
Application Programming Interface (API)	Provides functions for the application to interact with the sensor.	Functions like getTemperature() and getHumidity().
Application Layer	Your code that uses the API to read sensor data	Code to display temperature and humidity on a screen, send data to the

	and perform actions.	cloud, or control a thermostat.



SoC: system on chip



I2C Master Write and Read

Some I2C device needs write configurations before reading data from it. Therefore, an interface called `i2c_master_transmit_receive()` can help. The principle of this function can be explained by following chart.

I2C master write to slave and read from slave

Please note that no STOP condition bit is inserted between the write and read operations; therefore, this function is suited to read a register from an I2C device. A simple example for writing

I think this is what we are doing when sending a command to the temp sensor

In C, the ampersand symbol (&) is the **address-of operator**. It gives you the memory address of a variable.

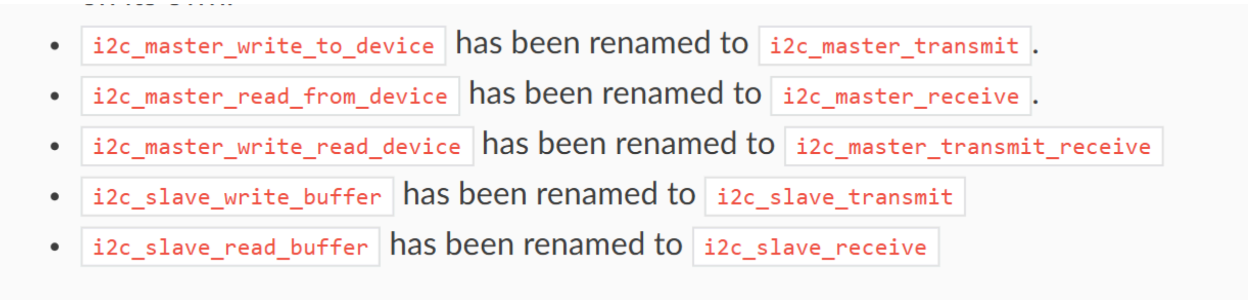
Layer	Description	Code Example	Pros	Cons
Application Code	Your program logic using I2C for specific tasks (e.g., reading sensor data).	i2c_master_transmit(...); i2c_master_receive(...)	Easiest to use; hides complexity.	Least control over I2C hardware.
ESP-IDF I2C Driver	Provides an API with functions to simplify I2C communication.	i2c_master_bus_add_device(...); i2c_master_start(...)	Simplifies I2C operations; handles low-level details.	Less control than lower levels.
ESP32 HAL (Hardware Abstraction Layer)	Abstracts hardware-specific details, providing a more generic interface to the I2C peripheral.	gpio_hal_iomux_func_sel(...); i2c_dev_t *i2c = I2C_DEV(i2c_num);	More portable code; easier to maintain.	Less control than direct register access.
ESP32 I2C Peripheral	The hardware block within the ESP32 that implements the I2C protocol.	-	Handles I2C signaling, clock synchronization, etc.	Not directly accessible by software.
Registers and Bit Fields	Memory locations within the ESP32 that control the I2C peripheral's behavior.	I2C0.ctr.en = 1; I2C0.scl_low_period.scl_low_period = 10;	Most control over I2C hardware.	Most complex; requires deep hardware knowledge.
Memory-Mapped Registers	Accessing I2C registers using their memory addresses.	`*(volatile uint32_t *)(0x3ff48070)	#ERROR!	Maximum control and potential performance gains.

Add to code notes:

```
i2c_master_transmit()
i2c_master_receive()
```

`i2c_master_transmit_receive()`

`i2c_master_probe()`



- `i2c_master_write_to_device` has been renamed to `i2c_master_transmit`.
- `i2c_master_read_from_device` has been renamed to `i2c_master_receive`.
- `i2c_master_write_read_device` has been renamed to `i2c_master_transmit_receive`
- `i2c_slave_write_buffer` has been renamed to `i2c_slave_transmit`
- `i2c_slave_read_buffer` has been renamed to `i2c_slave_receive`

APB frequency: advanced peripheral bus

1/24/2025

How to create an ESP-IDF component

2 December 2024 · 12 mins ·

I2C Registry Component ESP-IDF Driver Library

AUTHOR

Pedro Minatel

Developer Relations at Espressif

Using a monolithic architecture to develop complex applications tightly integrated with all the business logic, peripheral drivers, protocols, cloud

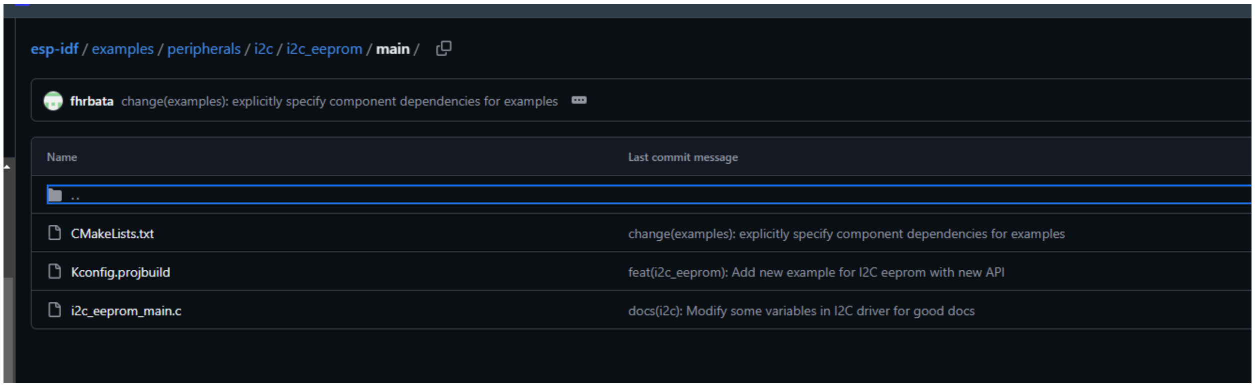
Blog link

This guys blog is worth checking out but it is not necessarily the application i am looking for. His blog gives instructions on how to create a component for the ESP IDF registry. While, i may want to create a component down the line. I would just like a straightforward way to create a driver for my specific temp/humidity sensor. I would like to create this locally and not submit it to the registry just yet. I believe that this would be valuable and i will do this once i get it working and add some more configurable and versatile functionality.

Here is esspressif's documentation on what procedure pedro is following packaging ESP-IDF components

I think there are two types of file structures
Built in
Project

Eeprom example:

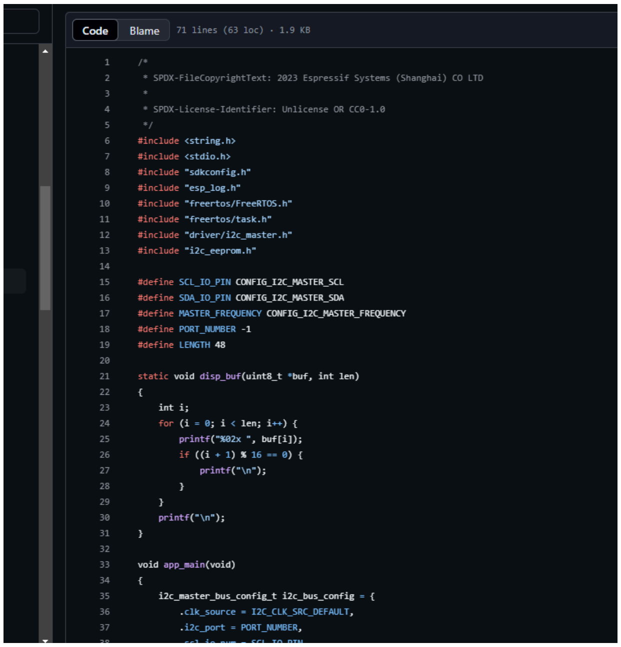


esp-idf / examples / peripherals / i2c / i2c_eeprom / **main** /

fhrbata change(examples): explicitly specify component dependencies for examples

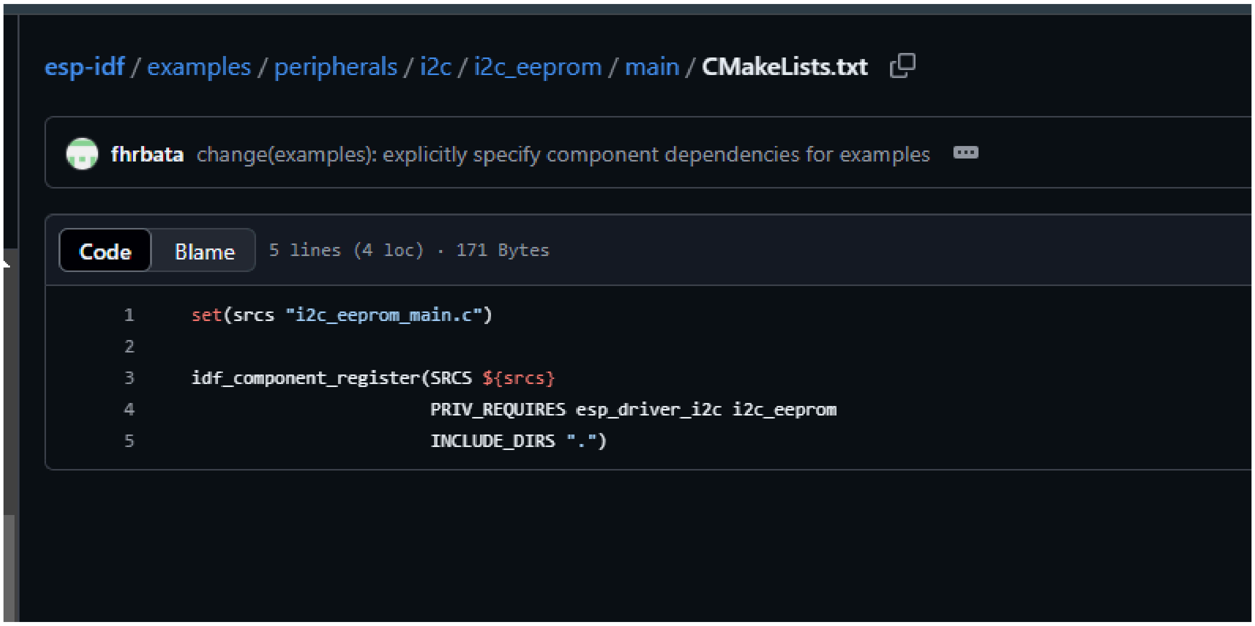
Name	Last commit message
..	
CMakeLists.txt	change(examples): explicitly specify component dependencies for examples
Kconfig.projbuild	feat(i2c_eeprom): Add new example for I2C eeprom with new API
i2c_eeprom_main.c	docs(i2c): Modify some variables in I2C driver for good docs

Main folder structure



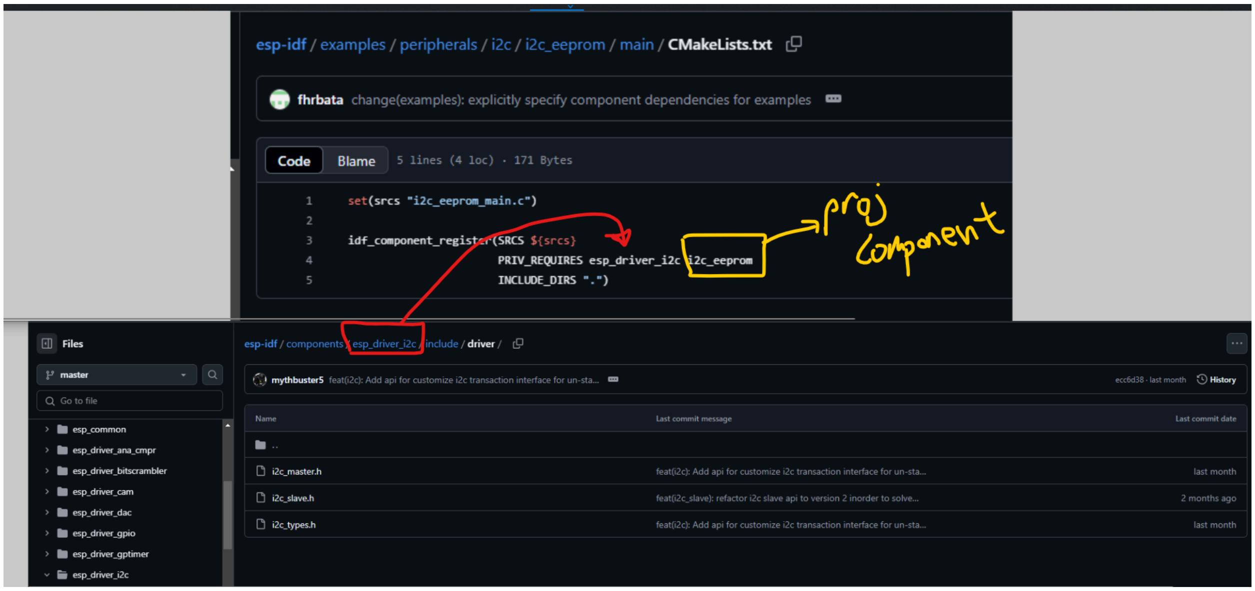
```c
/*
 * SPDX-FileCopyrightText: 2023 Espressif Systems (Shanghai) CO LTD
 *
 * SPDX-License-Identifier: Unlicense OR CC0-1.0
 */
#include <string.h>
#include <stdio.h>
#include "sdkconfig.h"
#include "esp_log.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "driver/i2c_master.h"
#include "i2c_eeprom.h"

#define SCL_IO_PIN CONFIG_I2C_MASTER_SCL
#define SDA_IO_PIN CONFIG_I2C_MASTER_SDA
#define MASTER_FREQUENCY CONFIG_I2C_MASTER_FREQUENCY
#define PORT_NUMBER -1
#define LENGTH 48

static void disp_buf(uint8_t *buf, int len)
{
    int i;
    for (i = 0; i < len; i++) {
        printf("%02x ", buf[i]);
        if ((i + 1) % 16 == 0) {
            printf("\n");
        }
    }
    printf("\n");
}

void app_main(void)
{
    i2c_master_bus_config_t i2c_bus_config = {
        .clk_source = I2C_CLK_SRC_DEFAULT,
        .i2c_port = PORT_NUMBER,
```

I2c_eeprom_main.c [declaration inside main]


esp-idf / examples / peripherals / i2c / i2c_eeprom / main / CMakeLists.txt
fhrbata change(examples): explicitly specify component dependencies for examples
Code
Blame
5 lines (4 loc) · 171 Bytes

```
set(srcs "i2c_eeprom_main.c")

idf_component_register(SRCS ${srcs}
                       PRIV_REQUIRES esp_driver_i2c i2c_eeprom
                       INCLUDE_DIRS ".")
```

Cmake config inside main. This includes the src file which is main and


esp-idf / examples / peripherals / i2c / i2c_eeprom / main / CMakeLists.txt
fhrbata change(examples): explicitly specify component dependencies for examples
Code
Blame
5 lines (4 loc) · 171 Bytes
proj
Component
Files
master
Go to file
esp_common
esp_driver_ana_cmpr
esp_driver_bitscrambler
esp_driver_cam
esp_driver_dac
esp_driver_gpio
esp_driver_gptimer
esp_driver_i2c
esp-idf / components / esp_driver_i2c / include / driver /
mythbuster5 feat(i2c): Add api for customize i2c transaction interface for un-sta...
ecc6d38 · last month
History

```
set(srcs "i2c_eeprom_main.c")

idf_component_register(SRCS ${srcs}
                       PRIV_REQUIRES esp_driver_i2c i2c_eeprom
                       INCLUDE_DIRS ".")
```

Name	Last commit message	Last commit date
..		
i2c_master.h	feat(i2c): Add api for customize i2c transaction interface for un-sta...	last month
i2c_slave.h	feat(i2c_slave): refactor i2c slave api to version 2 inorder to solve...	2 months ago
i2c_types.h	feat(i2c): Add api for customize i2c transaction interface for un-sta...	last month

In the main CMakeLists.txt we need to include the IDF component folder. In this case its “esp_driver_i2c”

Next we need to import the local project component which is i2c_eeprom

Angle brackets (<>) are used for standard header files found in **system include directories**.

1

`#include "driver/i2c_master.h"` and **`#include "esp_err.h"`**: These use double quotes (" "). This instructs the compiler to first search for the header files in the **current directory**

NatureScan-SH31 Folder Structure

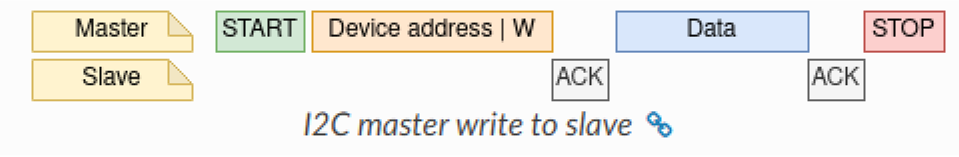
Ref: [i2c_eeprom]

📁 NatureScan-SHT31/

.. more questions:
What do these config files do:
sdkconfig
idf_component.yml # Component manager config. Import the ESPIDF framework components??
Kconfig.projbuild   % give config for menuconifg???

Main i2c_master functions

I2C Master Write



I2C master write to slave

Example:

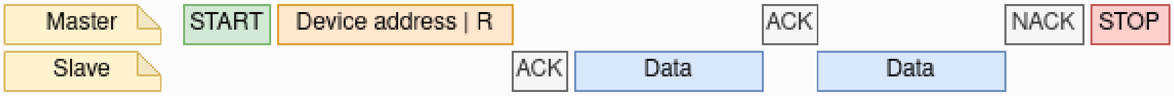
I2C Action	Binary	Hex	Explanation
Start Condition	10000000	0x80	Signals the beginning of an I2C transaction
Device Address (Write)	1000000	0x40	7-bit address (0x20) shifted left by 1 bit, with the least significant bit (LSB) set to 0 for write operation

Acknowledge (ACK)	0		Slave acknowledges address
Data Byte	1000010	0x42	The data byte to be written
Acknowledge (ACK)	0		Slave acknowledges data byte
Stop Condition	10010000	0x90	Signals the end of the I2C transaction

10000000 01000000 0 01000010 0 10010000

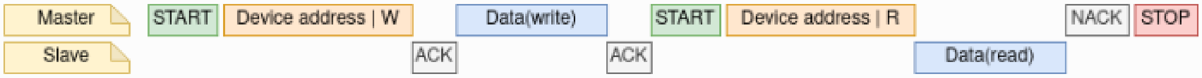
Hex:
0x80 0x40 0x42 0x90

I2C Master Read



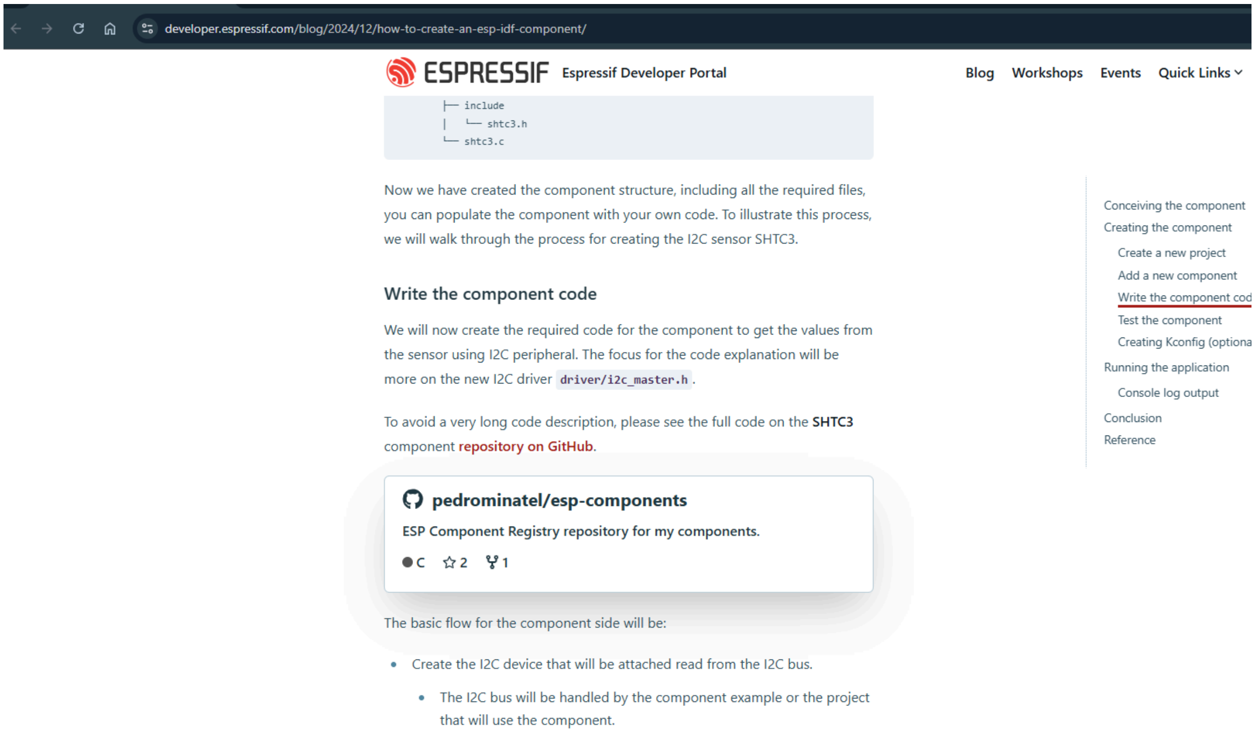
I2C master read from slave

I2C Master Write and Read



I2C master write to slave and read from slave

+++++++++++++++++++++++++++++++++=


developer.espressif.com/blog/2024/12/how-to-create-an-esp-idf-component/
ESPRESSIF Espressif Developer Portal
Blog Workshops Events Quick Links

```
├─ include
│   └─ shtc3.h
└─ shtc3.c
```

Now we have created the component structure, including all the required files, you can populate the component with your own code. To illustrate this process, we will walk through the process for creating the I2C sensor SHTC3.

Write the component code

We will now create the required code for the component to get the values from the sensor using I2C peripheral. The focus for the code explanation will be more on the new I2C driver `driver/i2c_master.h`.

To avoid a very long code description, please see the full code on the **SHTC3** component repository on GitHub.

pedrominatel/esp-components

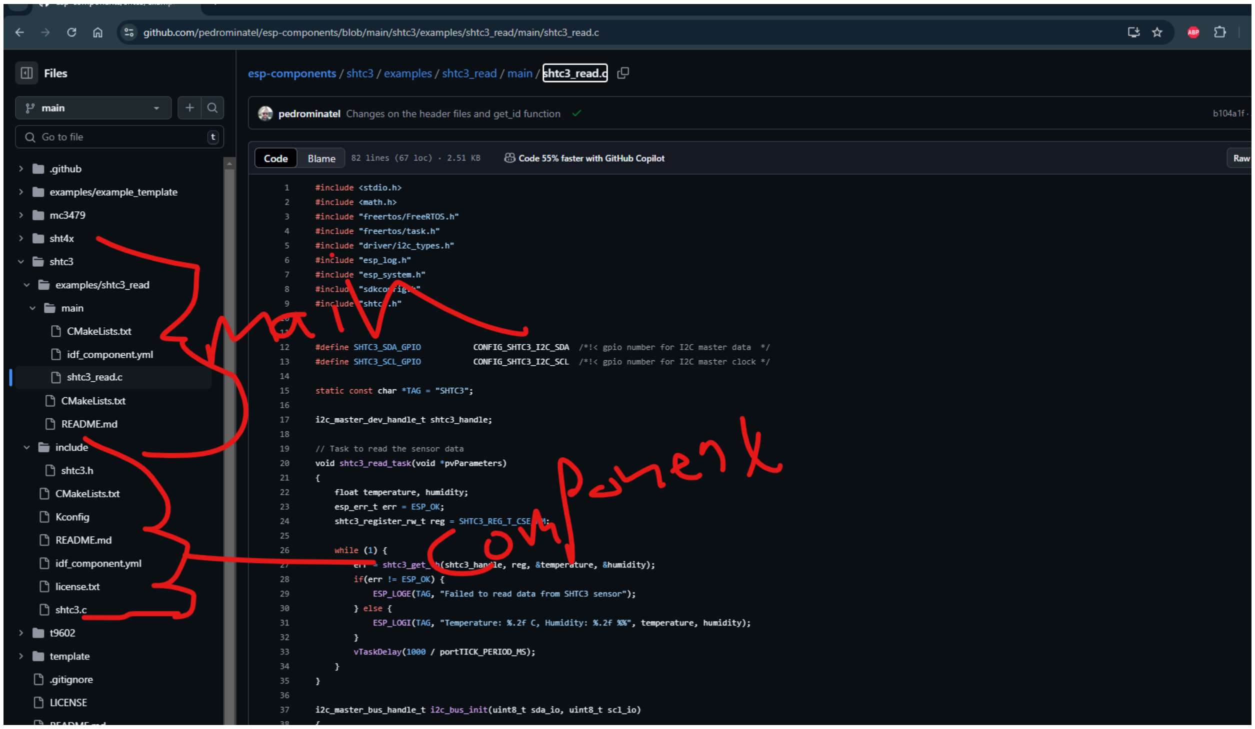
ESP Component Registry repository for my components.

● C ☆ 2 ⑂ 1

The basic flow for the component side will be:

- Create the I2C device that will be attached read from the I2C bus.
  - The I2C bus will be handled by the component example or the project that will use the component.

Conceiving the component
Creating the component
Create a new project
Add a new component
Write the component cod
Test the component
Creating Kconfig (optiona
Running the application
Console log output
Conclusion
Reference


github.com/pedrominatel/esp-components/blob/main/shtc3/examples/shtc3_read/main/shtc3_read.c
Files
main
Go to file
esp-components / shtc3 / examples / shtc3_read / main / shtc3_read.c
Code
Blame
82 lines (67 loc) · 2.51 KB
Code 55% faster with GitHub Copilot
Raw
main
component

- .github
- examples/example_template
- mc3479
- sht4x
- shtc3
  - examples/shtc3_read
    - main
      - CMakeLists.txt
      - idf_component.yml
      - shtc3_read.c
    - CMakeLists.txt
    - README.md
  - include
    - shtc3.h
  - CMakeLists.txt
  - Kconfig
  - README.md
  - idf_component.yml
  - license.txt
  - shtc3.c
- t9602
- template
- .gitignore
- LICENSE

pedrominatel Changes on the header files and get_id function ✓

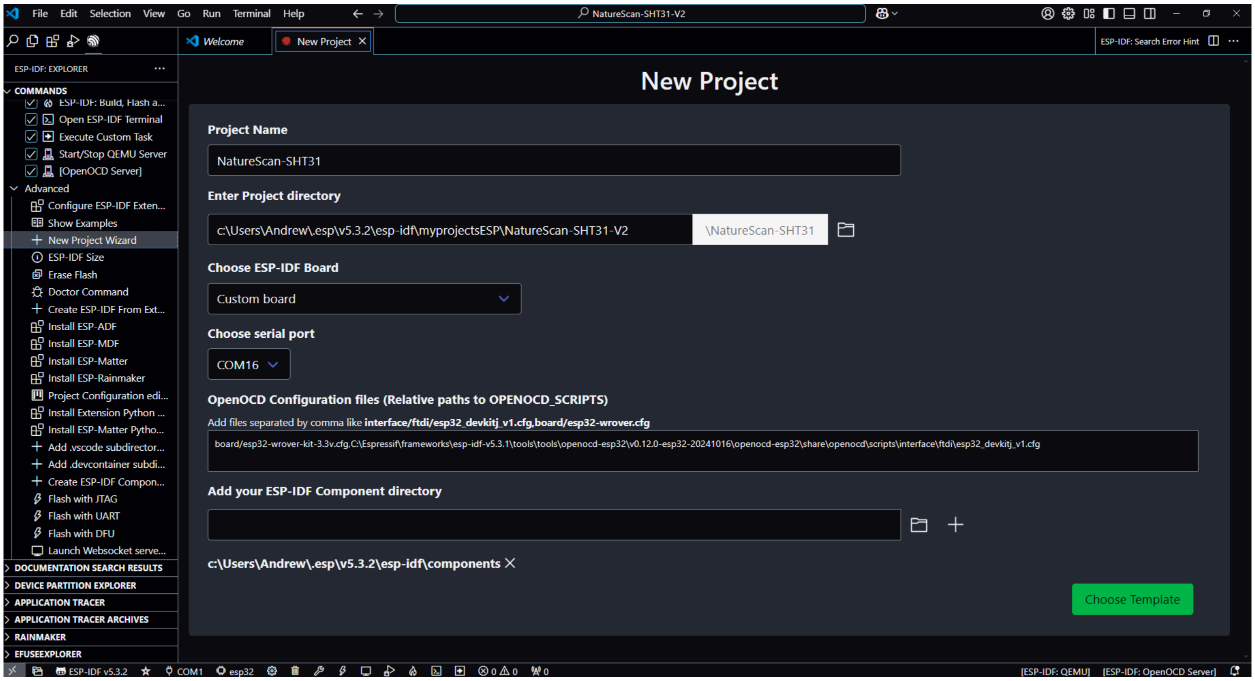
```c
#include <stdio.h>
#include <math.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "driver/i2c_types.h"
#include "esp_log.h"
#include "esp_system.h"
#include "sdkconfig.h"
#include "shtc3.h"


#define SHTC3_SDA_GPIO          CONFIG_SHTC3_I2C_SDA  /*!< gpio number for I2C master data  */
#define SHTC3_SCL_GPIO          CONFIG_SHTC3_I2C_SCL  /*!< gpio number for I2C master clock */

static const char *TAG = "SHTC3";

i2c_master_dev_handle_t shtc3_handle;

// Task to read the sensor data
void shtc3_read_task(void *pvParameters)
{
    float temperature, humidity;
    esp_err_t err = ESP_OK;
    shtc3_register_rw_t reg = SHTC3_REG_T_CSE_NM;

    while (1) {
        err = shtc3_get_th(shtc3_handle, reg, &temperature, &humidity);
        if(err != ESP_OK) {
            ESP_LOGE(TAG, "Failed to read data from SHTC3 sensor");
        } else {
            ESP_LOGI(TAG, "Temperature: %.2f C, Humidity: %.2f %%", temperature, humidity);
        }
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}

i2c_master_bus_handle_t i2c_bus_init(uint8_t sda_io, uint8_t scl_io)
```

C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2\NatureScan-SHT31

LFG

1/23/2025

🐋Deepseek thread



New Project

Project Name

NatureScan-SHT31

Enter Project directory

c:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2 \NatureScan-SHT31

Choose ESP-IDF Board

Custom board

Choose serial port

COM16

OpenOCD Configuration files (Relative paths to OPENOCD_SCRIPTS)

Add files separated by comma like **interface/ftdi/esp32_devkitj_v1.cfg,board/esp32-wrover.cfg**

board/esp32-wrover-kit-3.3v.cfg,C:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\openocd-esp32\v0.12.0-esp32-20241016\openocd-esp32\share\openocd\scripts\interface\ftdi\esp32_devkitj_v1.cfg

Add your ESP-IDF Component directory

c:\Users\Andrew\.esp\v5.3.2\esp-idf\components

Here are my new project new config params

***3/2/2025

Dont add the components folder. It will add all the esp idf components

Component directory

❌

```
📁 NatureScan-SHT31/
├── 📁 main/
│   ├── 📄 CMakeLists.txt   # Main component CMake
│   ├── 📄 main.c          # App entry point
│   └── 📁 components/     # (Optional) For project-specific components
│       └── 📁 SHT/
│           ├── 📄 CMakeLists.txt  # SHT component config
│           ├── 📄 SHT.c         # Sensor driver
│           └── 📄 SHT.h         # Header
├── 📁 components/          # ESP-IDF components (auto-imported)
├── 📄 CMakeLists.txt      # Root project CMake
├── 📄 sdkconfig           # Project configuration
└── 📄 idf_component.yml     # Component manager config
```

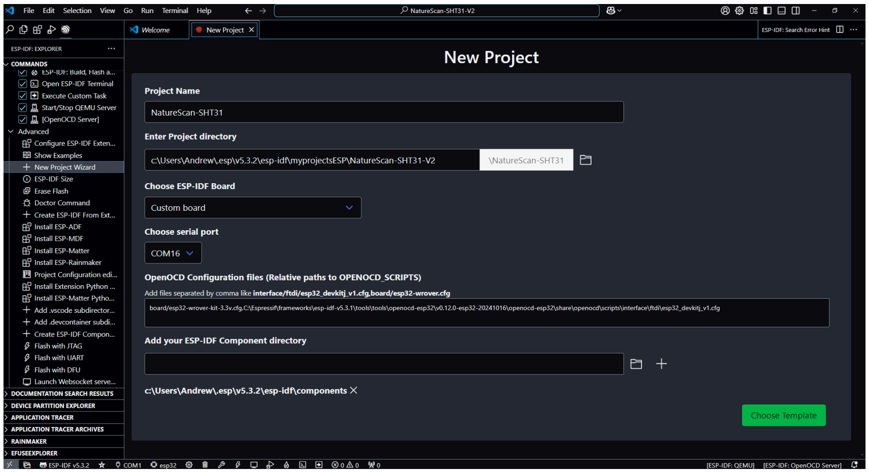
I think this file structure is wrong.

It might be right. Idk if there is a better structure for built in vs project components

New project flow

Esp idf new project wizard

Fill out you config



New Project

Project Name

NatureScan-SHT31

Enter Project directory

c:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\NatureScan-SHT31-V2 \NatureScan-SHT31

Choose ESP-IDF Board

Custom board

Choose serial port

COM16

OpenOCD Configuration files (Relative paths to OPENOCD_SCRIPTS)

Add files separated by comma like **interface/ftdi/esp32_devkitj_v1.cfg,board/esp32-wrover.cfg**

board/esp32-wrover-kit-3.3v.cfg,C:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\openocd-esp32\v0.12.0-esp32-20241016\openocd-esp32\share\openocd\scripts\interface\ftdi\esp32_devkitj_v1.cfg

Add your ESP-IDF Component directory

c:\Users\Andrew\.esp\v5.3.2\esp-idf\components ✕

Select basic project
Build project to instantiate it
Then do a blink

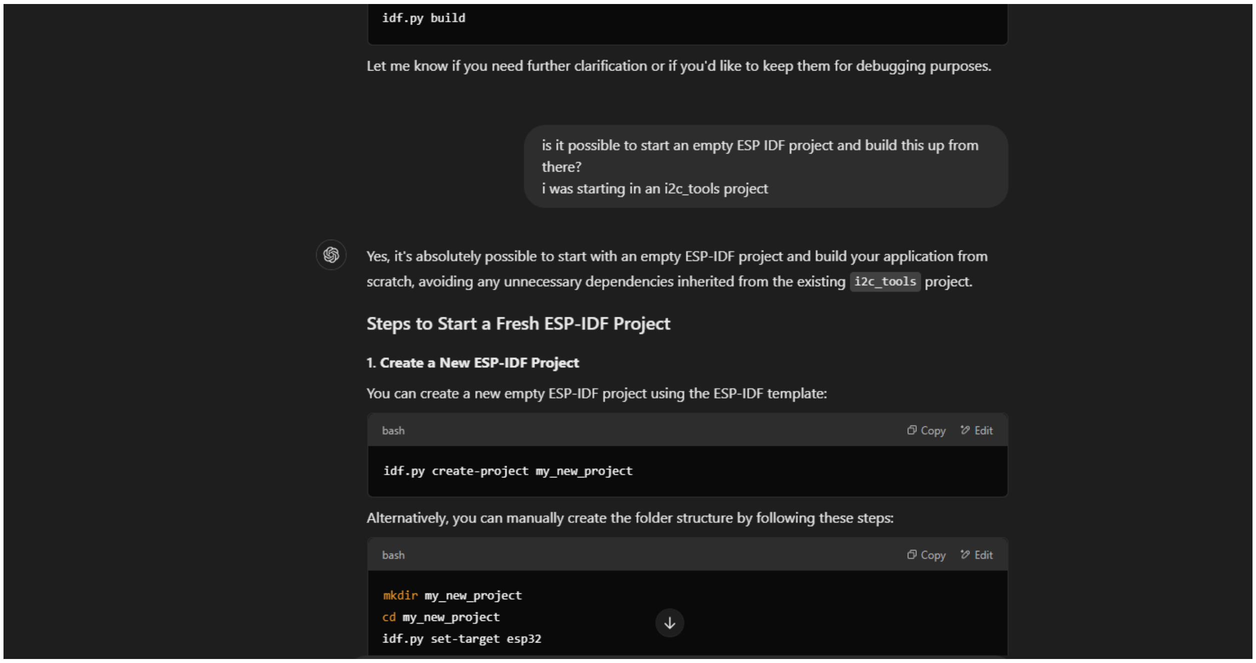
1/20/2025

this is the structure of my main folder:
- main/- CMakeLists.txt
  - SHT.c
  - SHT.h
  -main.c

- idf_component.yml

Good tryi

For next time
Create fresh new project



```
idf.py build
```

Let me know if you need further clarification or if you'd like to keep them for debugging purposes.

is it possible to start an empty ESP IDF project and build this up from there?
i was starting in an i2c_tools project

Yes, it's absolutely possible to start with an empty ESP-IDF project and build your application from scratch, avoiding any unnecessary dependencies inherited from the existing `i2c_tools` project.

Steps to Start a Fresh ESP-IDF Project

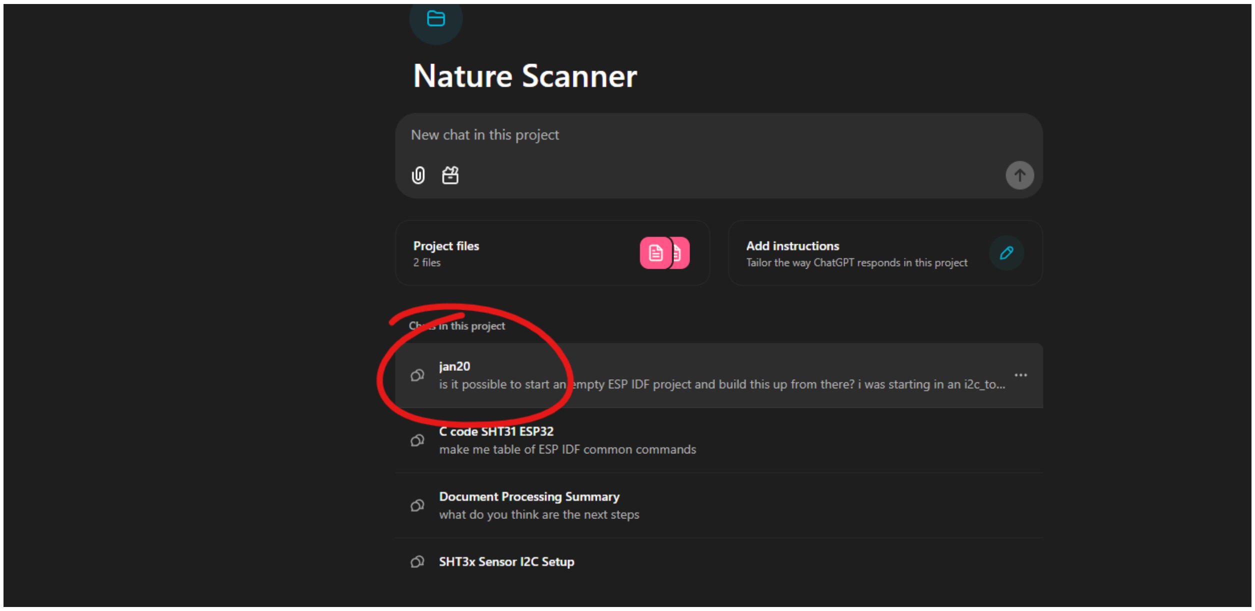
1. Create a New ESP-IDF Project

You can create a new empty ESP-IDF project using the ESP-IDF template:

```bash
idf.py create-project my_new_project
```

Alternatively, you can manually create the folder structure by following these steps:

```bash
mkdir my_new_project
cd my_new_project
idf.py set-target esp32
```



Nature Scanner

Project files
2 files

Add instructions
Tailor the way ChatGPT responds in this project

Chats in this project

jan20
is it possible to start an empty ESP IDF project and build this up from there? i was starting in an i2c_to...

C code SHT31 ESP32
make me table of ESP IDF common commands

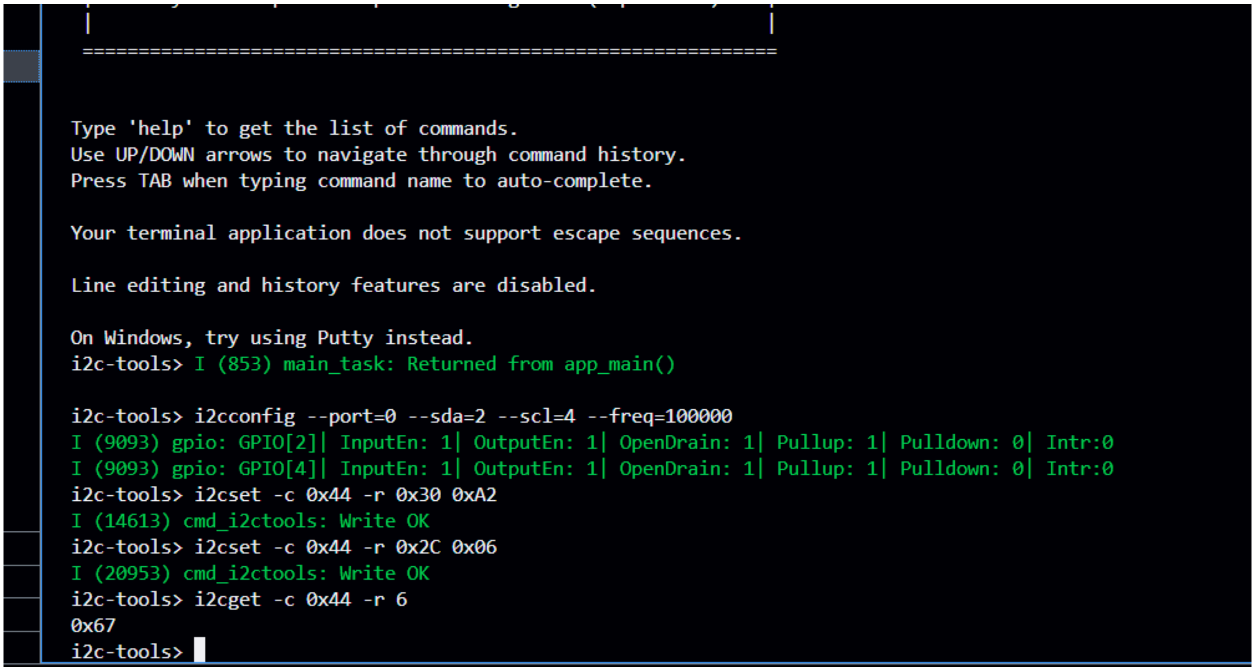
Document Processing Summary
what do you think are the next steps

SHT3x Sensor I2C Setup

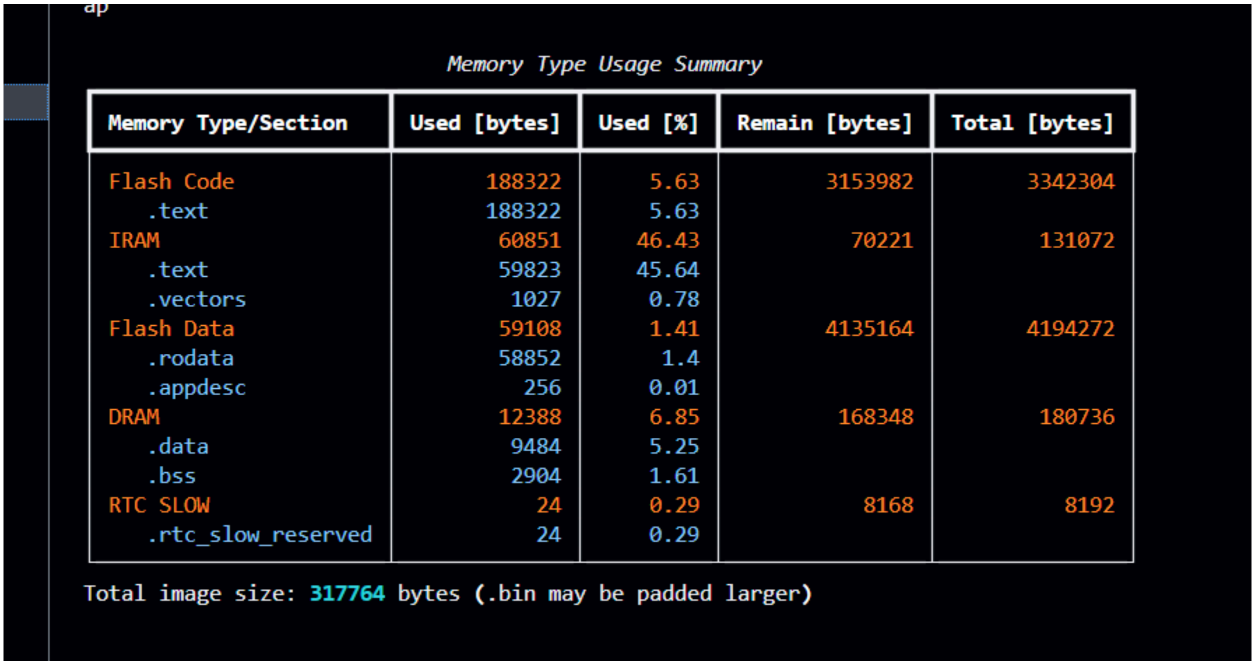
1/19/2025

Temp data!!!!!!!!!!!!!!!!!!!!!

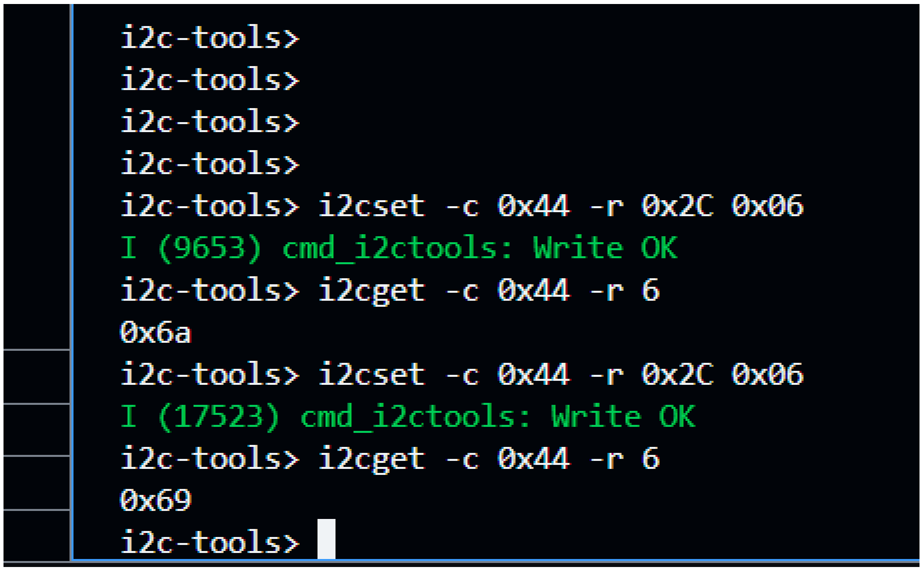
Lets do the calc



```
 |                                                            |
 ==============================================================


Type 'help' to get the list of commands.
Use UP/DOWN arrows to navigate through command history.
Press TAB when typing command name to auto-complete.

Your terminal application does not support escape sequences.

Line editing and history features are disabled.

On Windows, try using Putty instead.
i2c-tools> I (853) main_task: Returned from app_main()

i2c-tools> i2cconfig --port=0 --sda=2 --scl=4 --freq=100000
I (9093) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (9093) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
i2c-tools> i2cset -c 0x44 -r 0x30 0xA2
I (14613) cmd_i2ctools: Write OK
i2c-tools> i2cset -c 0x44 -r 0x2C 0x06
I (20953) cmd_i2ctools: Write OK
i2c-tools> i2cget -c 0x44 -r 6
0x67
i2c-tools> 
```

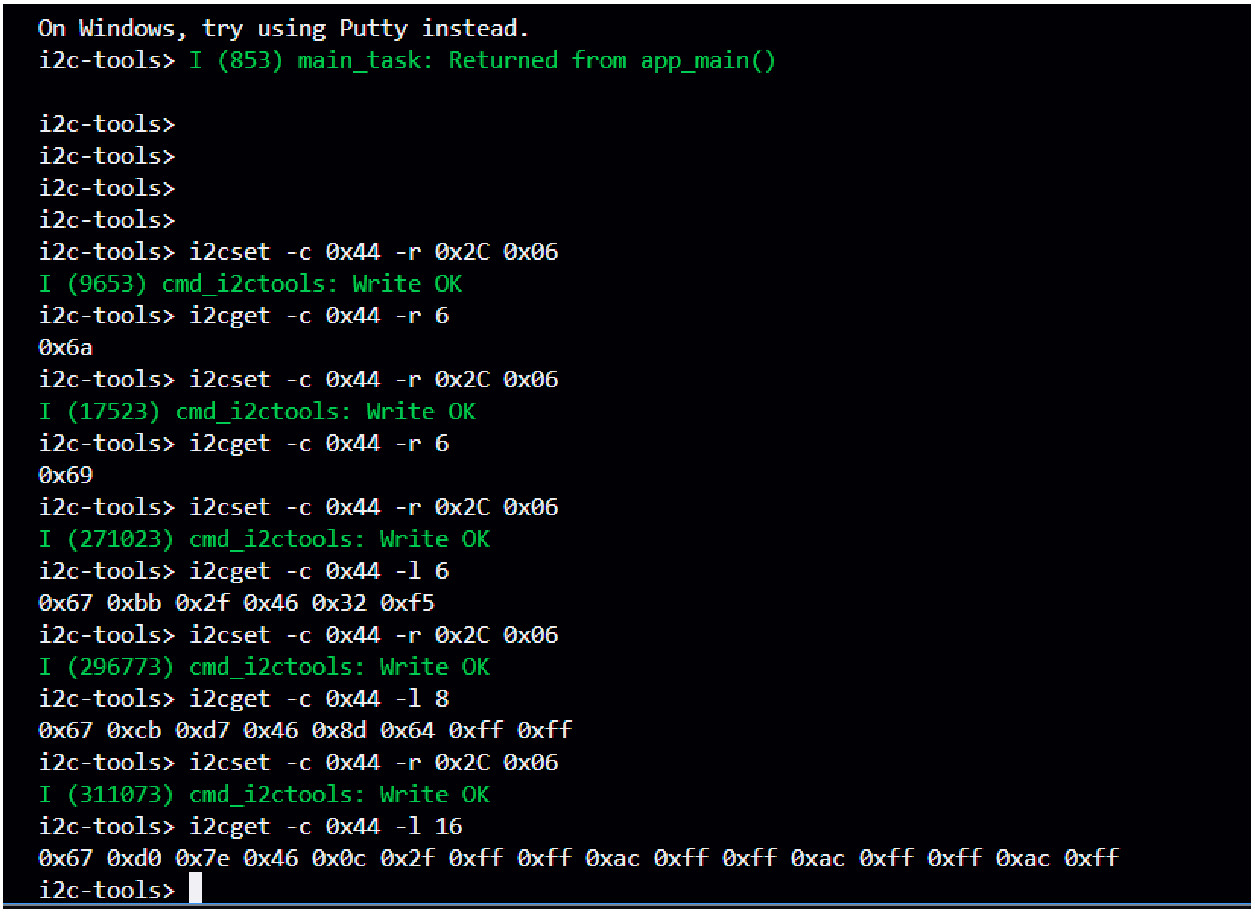


```
ap

                       Memory Type Usage Summary
┌────────────────────────┬─────────────┬──────────┬────────────────┬───────────────┐
│ Memory Type/Section    │ Used [bytes]│ Used [%] │ Remain [bytes] │ Total [bytes] │
├────────────────────────┼─────────────┼──────────┼────────────────┼───────────────┤
│ Flash Code             │      188322 │     5.63 │        3153982 │       3342304 │
│    .text               │      188322 │     5.63 │                │               │
│ IRAM                   │       60851 │    46.43 │          70221 │        131072 │
│    .text               │       59823 │    45.64 │                │               │
│    .vectors            │        1027 │     0.78 │                │               │
│ Flash Data             │       59108 │     1.41 │        4135164 │       4194272 │
│    .rodata             │       58852 │      1.4 │                │               │
│    .appdesc            │         256 │     0.01 │                │               │
│ DRAM                   │       12388 │     6.85 │         168348 │        180736 │
│    .data               │        9484 │     5.25 │                │               │
│    .bss                │        2904 │     1.61 │                │               │
│ RTC SLOW               │          24 │     0.29 │           8168 │          8192 │
│    .rtc_slow_reserved  │          24 │     0.29 │                │               │
└────────────────────────┴─────────────┴──────────┴────────────────┴───────────────┘
Total image size: 317764 bytes (.bin may be padded larger)
```



```
i2c-tools>
i2c-tools>
i2c-tools>
i2c-tools>
i2c-tools> i2cset -c 0x44 -r 0x2C 0x06
I (9653) cmd_i2ctools: Write OK
i2c-tools> i2cget -c 0x44 -r 6
0x6a
i2c-tools> i2cset -c 0x44 -r 0x2C 0x06
I (17523) cmd_i2ctools: Write OK
i2c-tools> i2cget -c 0x44 -r 6
0x69
i2c-tools>
```

The reason that the light goes off and it crashes is because maybe the data is removed. If i set and get the it works



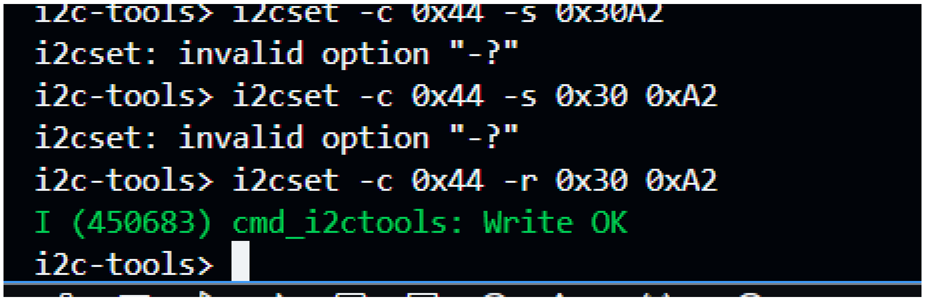
```
On Windows, try using Putty instead.
i2c-tools> I (853) main_task: Returned from app_main()

i2c-tools>
i2c-tools>
i2c-tools>
i2c-tools>
i2c-tools> i2cset -c 0x44 -r 0x2C 0x06
I (9653) cmd_i2ctools: Write OK
i2c-tools> i2cget -c 0x44 -r 6
0x6a
i2c-tools> i2cset -c 0x44 -r 0x2C 0x06
I (17523) cmd_i2ctools: Write OK
i2c-tools> i2cget -c 0x44 -r 6
0x69
i2c-tools> i2cset -c 0x44 -r 0x2C 0x06
I (271023) cmd_i2ctools: Write OK
i2c-tools> i2cget -c 0x44 -l 6
0x67 0xbb 0x2f 0x46 0x32 0xf5
i2c-tools> i2cset -c 0x44 -r 0x2C 0x06
I (296773) cmd_i2ctools: Write OK
i2c-tools> i2cget -c 0x44 -l 8
0x67 0xcb 0xd7 0x46 0x8d 0x64 0xff 0xff
i2c-tools> i2cset -c 0x44 -r 0x2C 0x06
I (311073) cmd_i2ctools: Write OK
i2c-tools> i2cget -c 0x44 -l 16
0x67 0xd0 0x7e 0x46 0x0c 0x2f 0xff 0xff 0xac 0xff 0xff 0xac 0xff 0xff 0xac 0xff
i2c-tools>
```

It worked!!!!!!!!!!!!!!!

I needed to grab the 6 bytes

1/18/2025

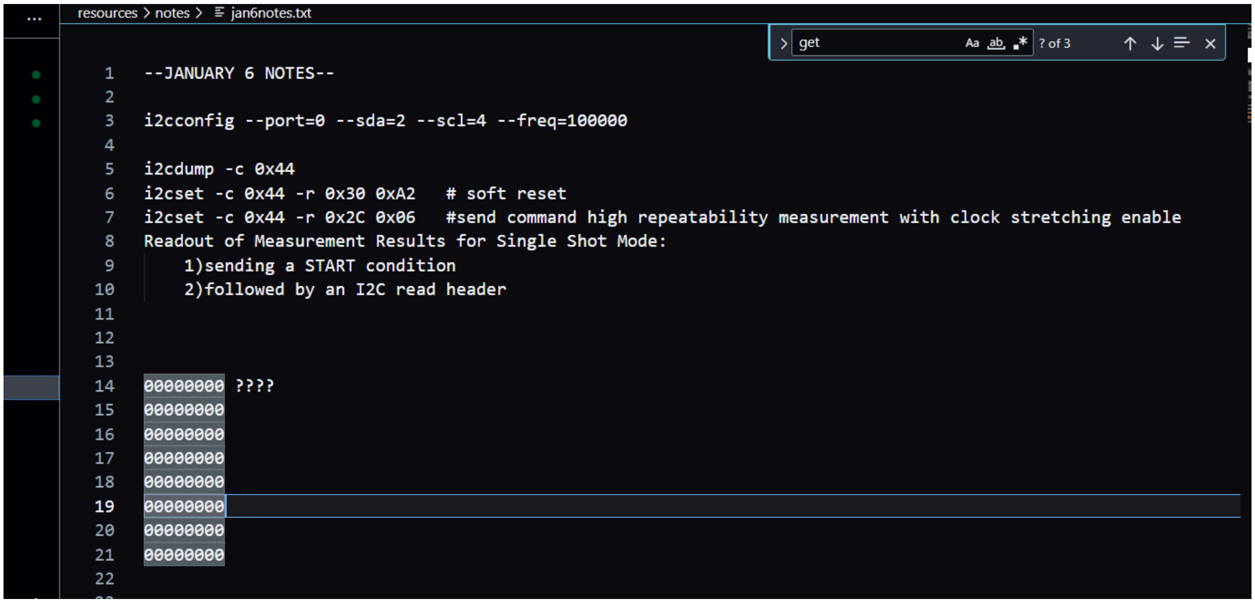
So im writing my own code from scratch right. Where did i leave off…



```
i2c-tools> i2cset -c 0x44 -s 0x30A2
i2cset: invalid option "-?"
i2c-tools> i2cset -c 0x44 -s 0x30 0xA2
i2cset: invalid option "-?"
i2c-tools> i2cset -c 0x44 -r 0x30 0xA2
I (450683) cmd_i2ctools: Write OK
i2c-tools> 
```

Successfully set a soft reset.

well i worked on my nature scanner project. i was able to write a one shot command to the esp32 0x44 register to take a single temp and humidity measurement on the SHT31. this is progress. now i need to be able to read it. more to come....



```
--JANUARY 6 NOTES--

i2cconfig --port=0 --sda=2 --scl=4 --freq=100000

i2cdump -c 0x44
i2cset -c 0x44 -r 0x30 0xA2   # soft reset
i2cset -c 0x44 -r 0x2C 0x06   #send command high repeatability measurement with clock stretching enable
Readout of Measurement Results for Single Shot Mode:
    1)sending a START condition
    2)followed by an I2C read header



00000000 ????
00000000
00000000
00000000
00000000
00000000
00000000
00000000

```

Next step is to read this..

Good prompting for gpt o1 too. This was great

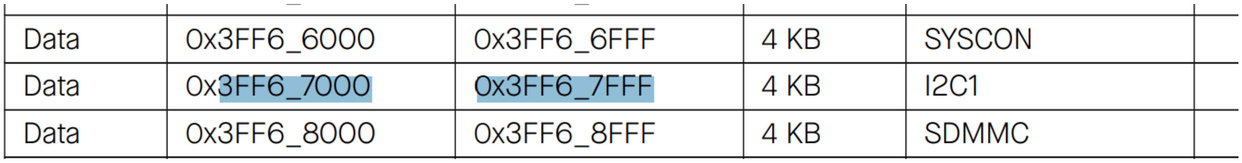
Commanding finally clicked for me. Well it was unclear where the commands were going you know.

So to summarize.
I was able to get a positive confirmation about writing commands.
There is a register and i think for ever command the buffer expands and contracts relative to what needs to happen. Which would make sense as the device needs to be efficient.
We were able to see 6 bytes that get saved. For a single shot sample, there is 6 bytes of data that get sent which include raw temp and humidity hex with CRC check

I will put together a python script to do a rough calculation.
I think i will keep pushing this until we are certain that we are looking at the right registers.
Maybe we arent though. The temp isnt that far off.

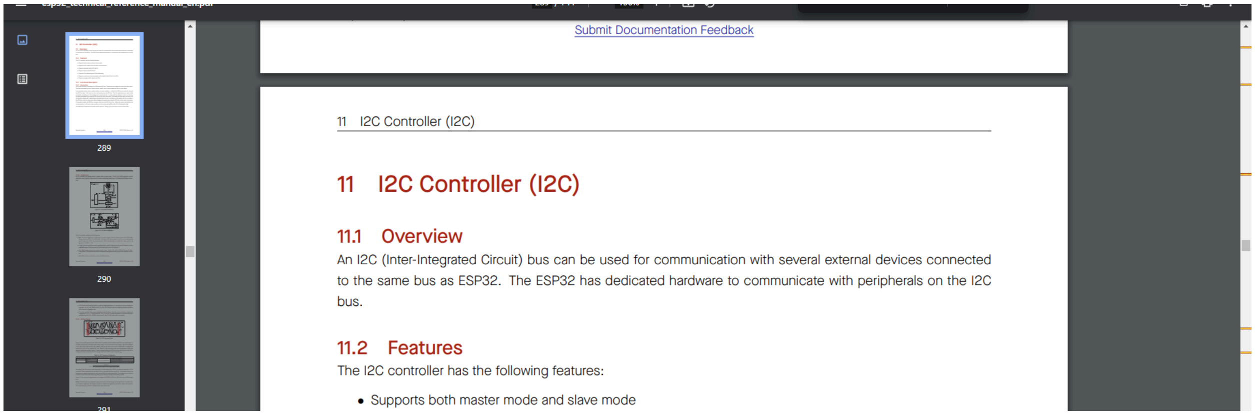
1/7/2025

```
idf term output:
               3ffbf3e4     dec 1073476580
i2C0 highend: 3FF53FFF             1073037311
i2C1 low |  3FF67000            1073115136  high  0x3FF67FFF 1073119231
```



Data	0x3FF6_6000	0x3FF6_6FFF	4 KB	SYSCON	
Data	0x3FF6_7000	0x3FF6_7FFF	4 KB	I2C1	
Data	0x3FF6_8000	0x3FF6_8FFF	4 KB	SDMMC	

In I2C1..



11 I2C Controller (I2C)

11 I2C Controller (I2C)

11.1 Overview

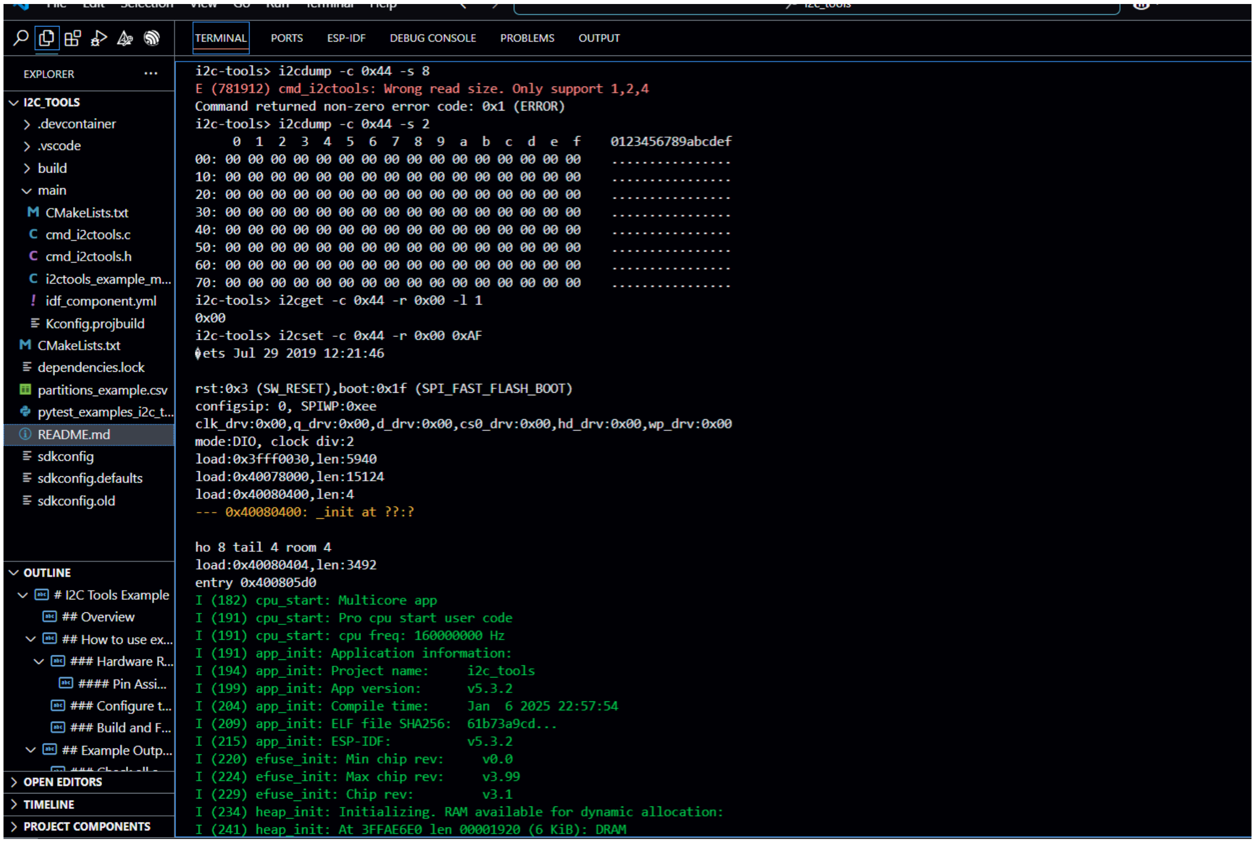
An I2C (Inter-Integrated Circuit) bus can be used for communication with several external devices connected to the same bus as ESP32. The ESP32 has dedicated hardware to communicate with peripherals on the I2C bus.

11.2 Features

The I2C controller has the following features:

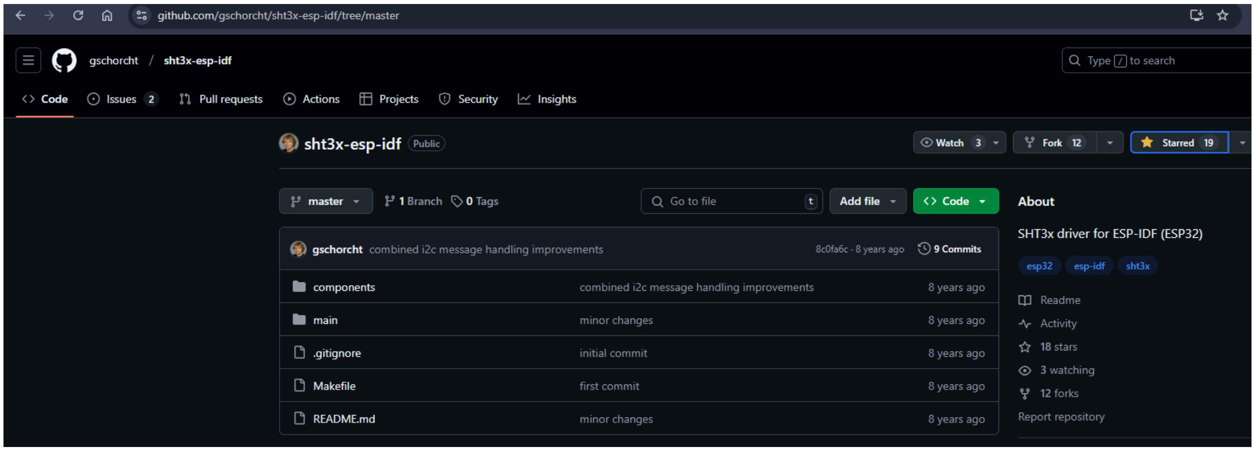
- Supports both master mode and slave mode

[Esp32 technical ref manual]()



```
i2c-tools> i2cdump -c 0x44 -s 8
E (781912) cmd_i2ctools: Wrong read size. Only support 1,2,4
Command returned non-zero error code: 0x1 (ERROR)
i2c-tools> i2cdump -c 0x44 -s 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
i2c-tools> i2cget -c 0x44 -r 0x00 -l 1
0x00
i2c-tools> i2cset -c 0x44 -r 0x00 0xAF
�ets Jul 29 2019 12:21:46

rst:0x3 (SW_RESET),boot:0x1f (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:5940
load:0x40078000,len:15124
load:0x40080400,len:4
--- 0x40080400: _init at ??:?

ho 8 tail 4 room 4
load:0x40080404,len:3492
entry 0x400805d0
I (182) cpu_start: Multicore app
I (191) cpu_start: Pro cpu start user code
I (191) cpu_start: cpu freq: 160000000 Hz
I (191) app_init: Application information:
I (194) app_init: Project name:     i2c_tools
I (199) app_init: App version:      v5.3.2
I (204) app_init: Compile time:     Jan  6 2025 22:57:54
I (209) app_init: ELF file SHA256:  61b73a9cd...
I (215) app_init: ESP-IDF:          v5.3.2
I (220) efuse_init: Min chip rev:     v0.0
I (224) efuse_init: Max chip rev:     v3.99
I (229) efuse_init: Chip rev:         v3.1
I (234) heap_init: Initializing. RAM available for dynamic allocation:
I (241) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
```

Wrote to register and started repeating



SHT3x driver for ESP-IDF (ESP32)

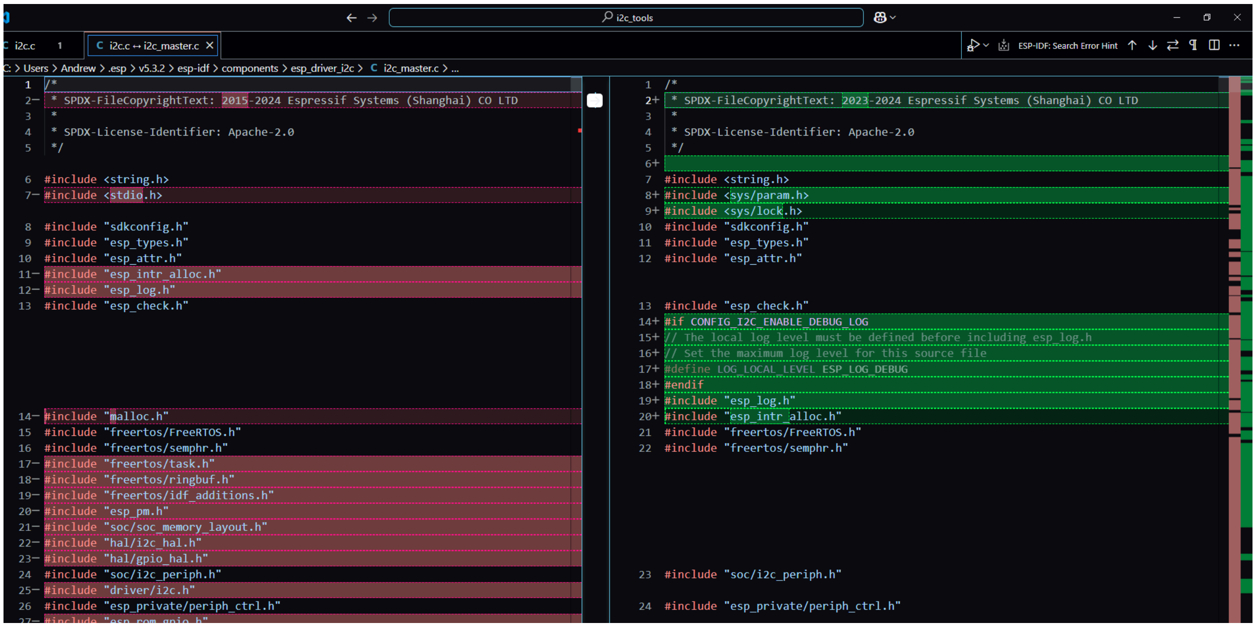
gschorcht combined i2c message handling improvements 8c0fa6c · 8 years ago 9 Commits

components	combined i2c message handling improvements	8 years ago
main	minor changes	8 years ago
.gitignore	initial commit	8 years ago
Makefile	first commit	8 years ago
README.md	minor changes	8 years ago

Could be worth a try
gschorcht
Gunar Schorcht

1/5/2025

More organizing of vscode and my fork of esp idf.
Moved previously archived sht files into my new archive file
Found the old uncleRus tutorial and confirmed the old i2c driver


i2c_tools
i2c.c 1
i2c.c ↔ i2c_master.c
C: > Users > Andrew > .esp > v5.3.2 > esp-idf > components > esp_driver_i2c > i2c_master.c > ...
ESP-IDF: Search Error Hint

```
/*
 * SPDX-FileCopyrightText: 2015-2024 Espressif Systems (Shanghai) CO LTD
 *
 * SPDX-License-Identifier: Apache-2.0
 */

#include <string.h>
#include <stdio.h>

#include "sdkconfig.h"
#include "esp_types.h"
#include "esp_attr.h"
#include "esp_intr_alloc.h"
#include "esp_log.h"
#include "esp_check.h"







#include "malloc.h"
#include "freertos/FreeRTOS.h"
#include "freertos/semphr.h"
#include "freertos/task.h"
#include "freertos/ringbuf.h"
#include "freertos/idf_additions.h"
#include "esp_pm.h"
#include "soc/soc_memory_layout.h"
#include "hal/i2c_hal.h"
#include "hal/gpio_hal.h"
#include "soc/i2c_periph.h"
#include "driver/i2c.h"
#include "esp_private/periph_ctrl.h"
```

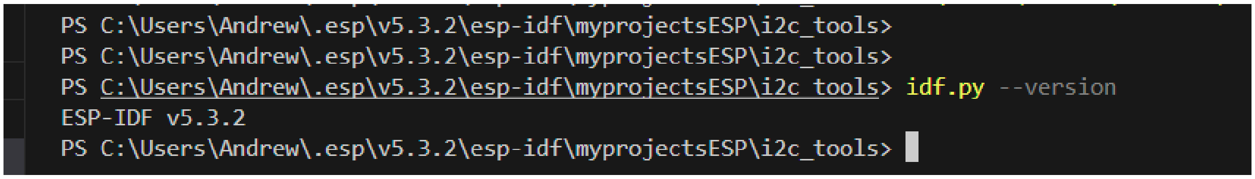
```
/*
 * SPDX-FileCopyrightText: 2023-2024 Espressif Systems (Shanghai) CO LTD
 *
 * SPDX-License-Identifier: Apache-2.0
 */

#include <string.h>
#include <sys/param.h>
#include <sys/lock.h>
#include "sdkconfig.h"
#include "esp_types.h"
#include "esp_attr.h"


#include "esp_check.h"
#if CONFIG_I2C_ENABLE_DEBUG_LOG
// The local log level must be defined before including esp_log.h
// Set the maximum log level for this source file
#define LOG_LOCAL_LEVEL ESP_LOG_DEBUG
#endif
#include "esp_log.h"
#include "esp_intr_alloc.h"
#include "freertos/FreeRTOS.h"
#include "freertos/semphr.h"







#include "soc/i2c_periph.h"

#include "esp_private/periph_ctrl.h"
```

Found the diff function and confirmed that i2c.c and i2c_master.c are vastly different

If i am going to do another tutorial i need to use one with i2c master as the esp32 chip i am using uses that.

I also need to decide how im going to do my development
I know that i need to hammer out if this setup actually works.
I need to write and read to the register and be able to take a temp reading and convert it i think

1/4/2025



```
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\i2c_tools>
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\i2c_tools>
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\i2c_tools> idf.py --version
ESP-IDF v5.3.2
PS C:\Users\Andrew\.esp\v5.3.2\esp-idf\myprojectsESP\i2c_tools>
```

Successfully reinstalled ESP IDF no more issues!!!

12/22/2024

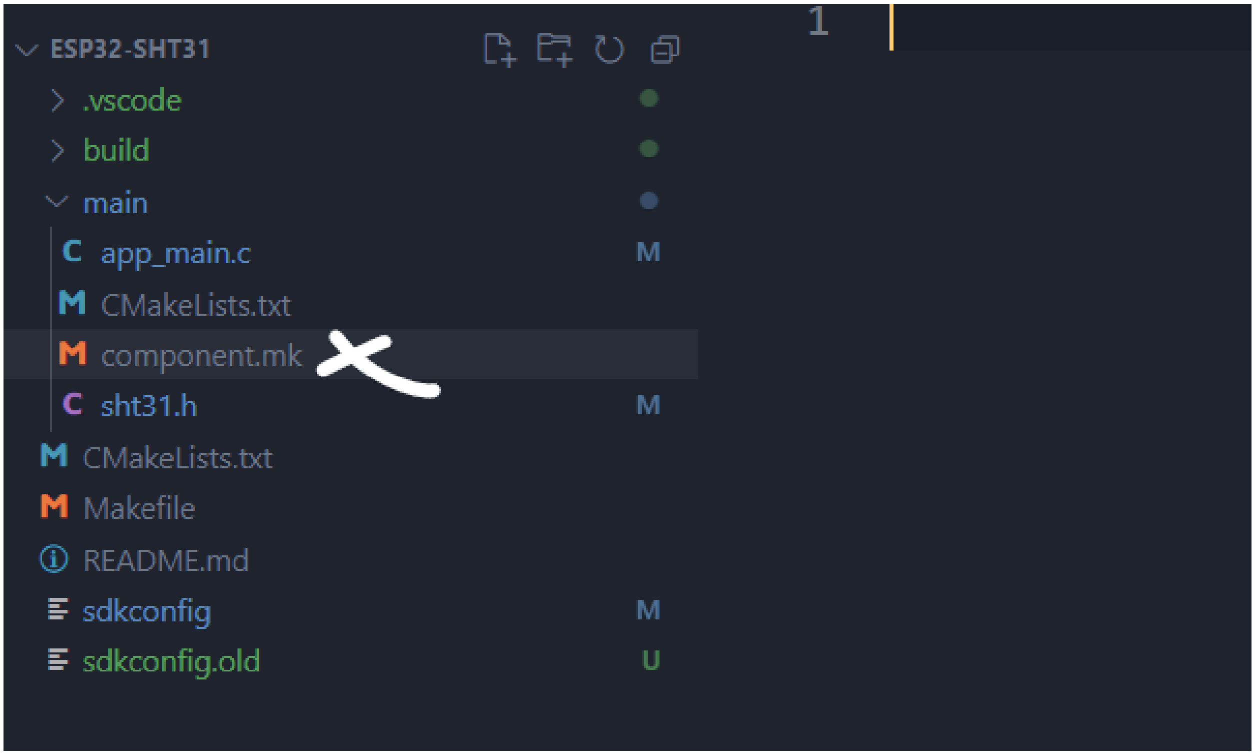
What project am i on? Where am i??
C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_i2ctools_scratch

Scratch

12/16/2024

12/2/2024

11/9/2024



```
ESP32-SHT31
  > .vscode
  > build
  v main
      app_main.c          M
      CMakeLists.txt
      component.mk
      sht31.h             M
    CMakeLists.txt
    Makefile
    README.md
    sdkconfig             M
    sdkconfig.old         U
```

Going to get rid of mk the make file to see if it still works

IDF Component Manager —-> CMake

```
#include "esp_log.h"
#include "driver/i2c.h"
```

Looks like only 2 dependencies??????

${IDF_PATH} = C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf

Set log to: C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf\components\log

Set i2c to: C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf\components\esp_driver_i2c

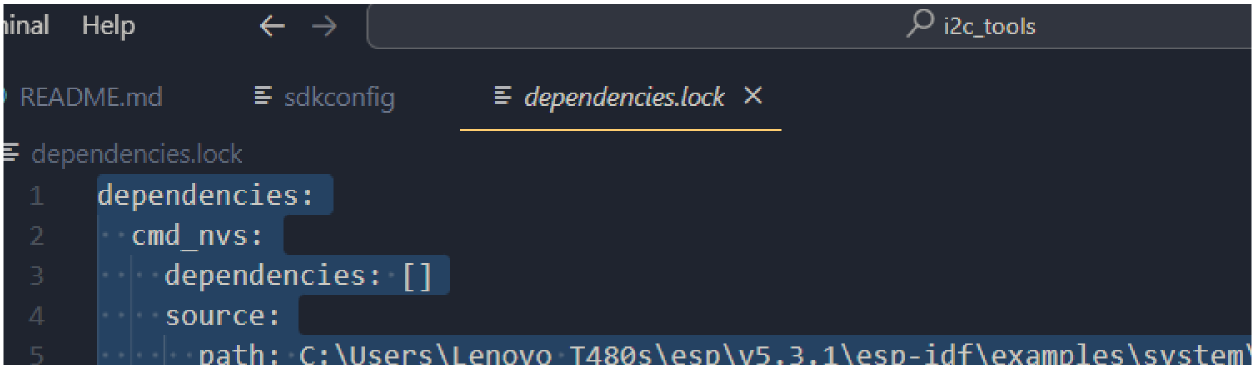
Example:

```
idf.py add-dependency --path="../../my_component" example/cmp^3.3.3
```

```
idf.py add-dependency –path=”${IDF_PATH}/components/log”  esp32-sht31/log
```

```
idf.py add-dependency –path=”${IDF_PATH}\components\log”  esp32-sht31\log
```

```
idf.py add-dependency –path=”C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf\components\log\components\log”
```



```
dependencies:
  cmd_nvs:
    dependencies: []
    source:
      path: C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf\examples\system
```

cmd_nvs

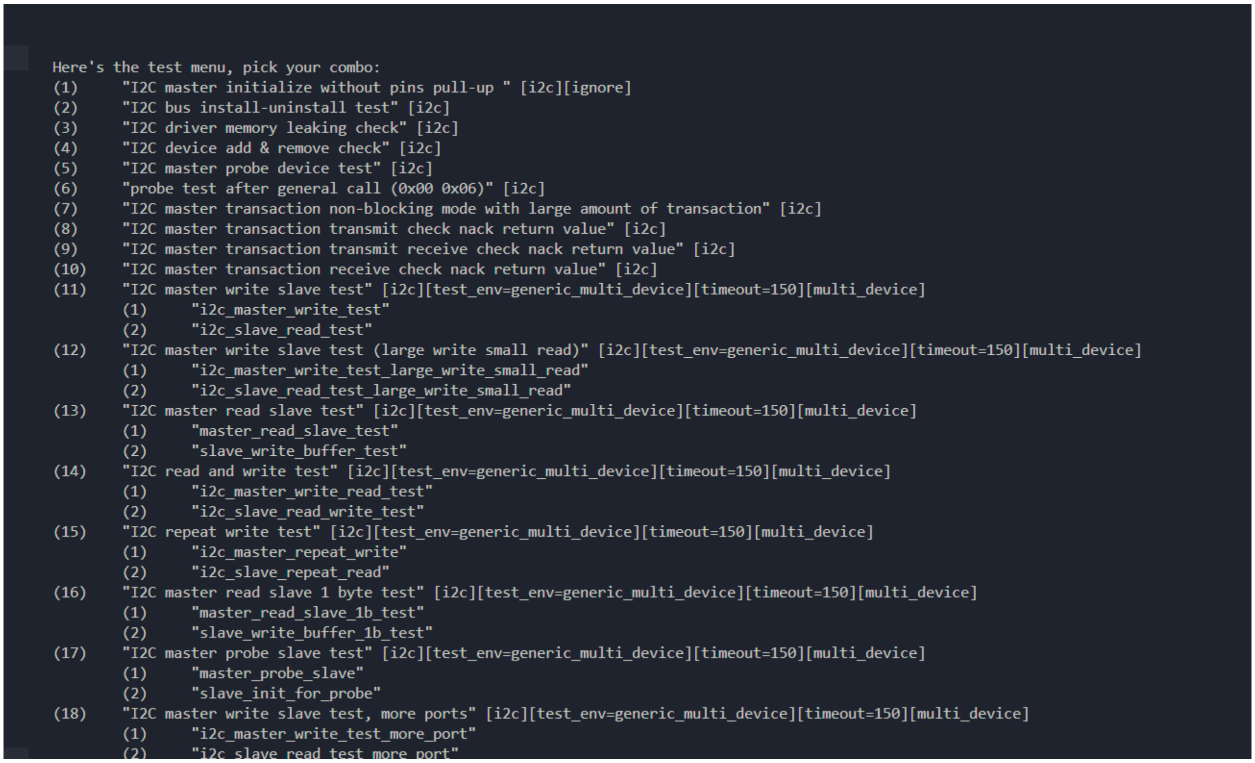
```
dependencies:
  cmd_nvs:
    dependencies: []
    source:
      path: C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf\examples\system\console\advanced\components\cmd_nvs
      type: local
    version: '*'
```

The issue is once again i2c master:

Two i2c folders:

"C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf\components\esp_driver_i2c\i2c_master.c"

"C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf\components\driver\i2c\include\driver\i2c.h"



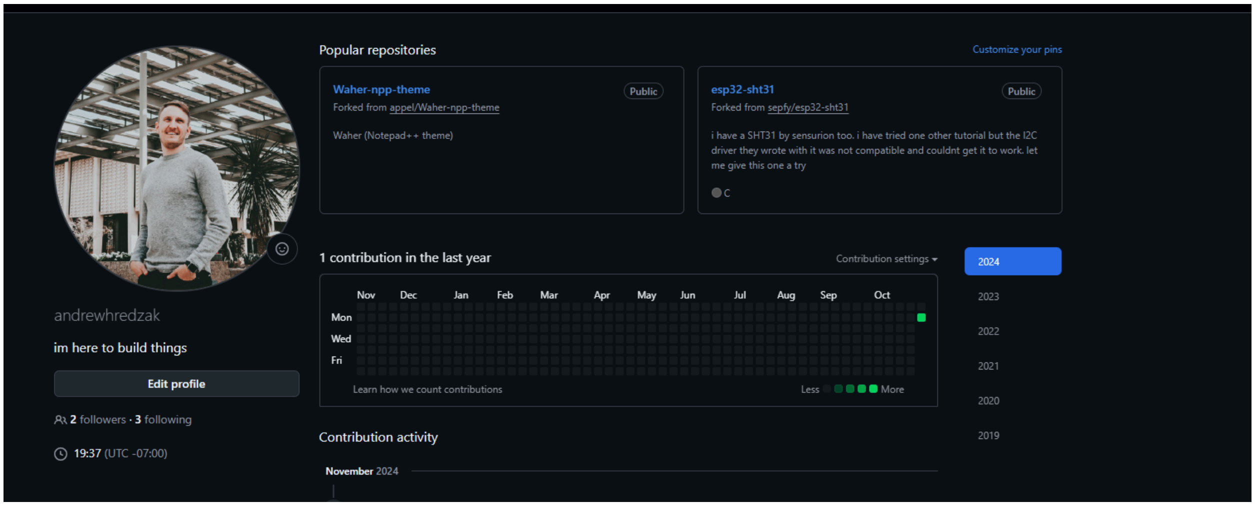
```
Here's the test menu, pick your combo:
(1)     "I2C master initialize without pins pull-up " [i2c][ignore]
(2)     "I2C bus install-uninstall test" [i2c]
(3)     "I2C driver memory leaking check" [i2c]
(4)     "I2C device add & remove check" [i2c]
(5)     "I2C master probe device test" [i2c]
(6)     "probe test after general call (0x00 0x06)" [i2c]
(7)     "I2C master transaction non-blocking mode with large amount of transaction" [i2c]
(8)     "I2C master transaction transmit check nack return value" [i2c]
(9)     "I2C master transaction transmit receive check nack return value" [i2c]
(10)    "I2C master transaction receive check nack return value" [i2c]
(11)    "I2C master write slave test" [i2c][test_env=generic_multi_device][timeout=150][multi_device]
        (1)     "i2c_master_write_test"
        (2)     "i2c_slave_read_test"
(12)    "I2C master write slave test (large write small read)" [i2c][test_env=generic_multi_device][timeout=150][multi_device]
        (1)     "i2c_master_write_test_large_write_small_read"
        (2)     "i2c_slave_read_test_large_write_small_read"
(13)    "I2C master read slave test" [i2c][test_env=generic_multi_device][timeout=150][multi_device]
        (1)     "master_read_slave_test"
        (2)     "slave_write_buffer_test"
(14)    "I2C read and write test" [i2c][test_env=generic_multi_device][timeout=150][multi_device]
        (1)     "i2c_master_write_read_test"
        (2)     "i2c_slave_read_write_test"
(15)    "I2C repeat write test" [i2c][test_env=generic_multi_device][timeout=150][multi_device]
        (1)     "i2c_master_repeat_write"
        (2)     "i2c_slave_repeat_read"
(16)    "I2C master read slave 1 byte test" [i2c][test_env=generic_multi_device][timeout=150][multi_device]
        (1)     "master_read_slave_1b_test"
        (2)     "slave_write_buffer_1b_test"
(17)    "I2C master probe slave test" [i2c][test_env=generic_multi_device][timeout=150][multi_device]
        (1)     "master_probe_slave"
        (2)     "slave_init_for_probe"
(18)    "I2C master write slave test, more ports" [i2c][test_env=generic_multi_device][timeout=150][multi_device]
        (1)     "i2c_master_write_test_more_port"
        (2)     "i2c_slave_read_test_more_port"
```

Current

1. PASS
2. PASS
3. PASS
4. PASS
5. FAIL
6. FAIL
7. PASS

8. FAIL
9. FAIL
10. FAIL
11. FAIL

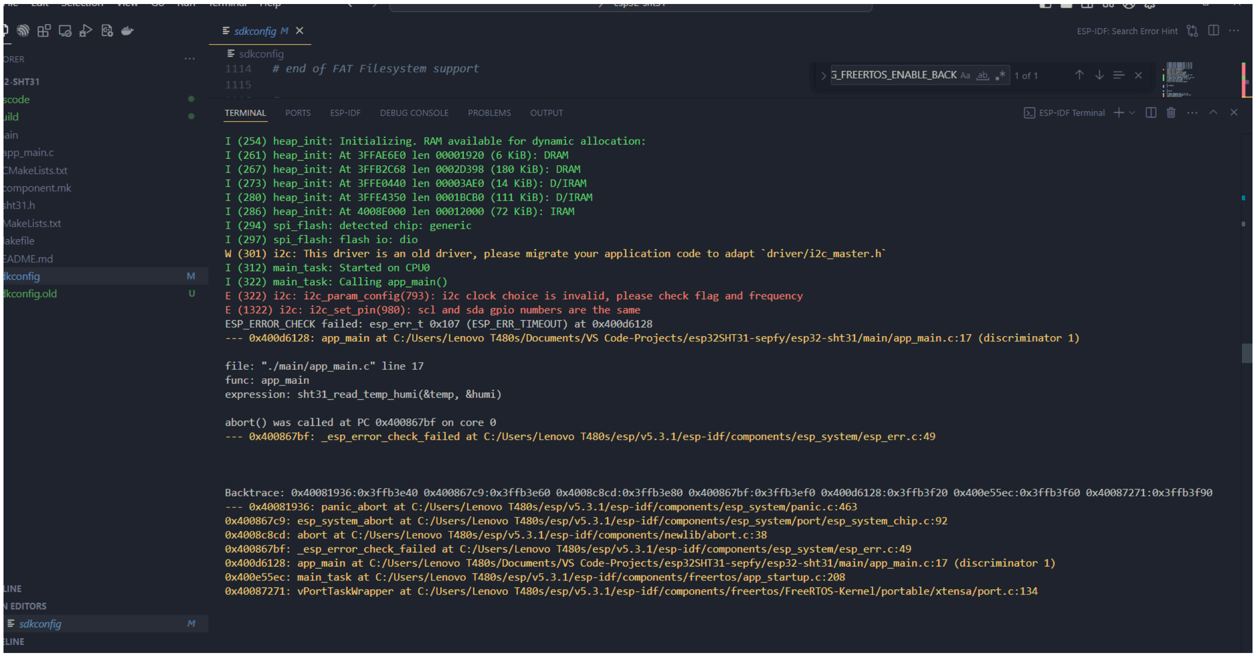
11/4/2024



i have a SHT31 by sensurion too. i have tried one other tutorial but the I2C driver they wrote with it was not compatible and couldnt get it to work. let me give this one a try

im here to build things

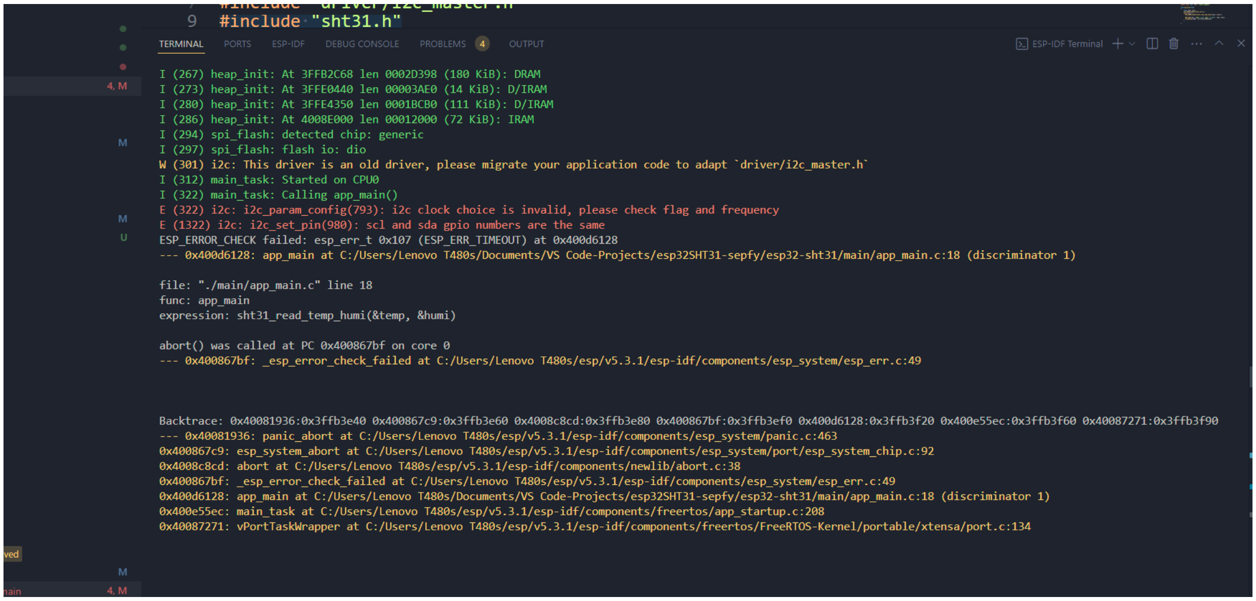
First github contribution



```
sdkconfig
1114    # end of FAT Filesystem support
1115
```

```
I (254) heap_init: Initializing. RAM available for dynamic allocation:
I (261) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (267) heap_init: At 3FFB2C68 len 0002D398 (180 KiB): DRAM
I (273) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (280) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (286) heap_init: At 4008E000 len 00012000 (72 KiB): IRAM
I (294) spi_flash: detected chip: generic
I (297) spi_flash: flash io: dio
W (301) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (312) main_task: Started on CPU0
I (322) main_task: Calling app_main()
E (322) i2c: i2c_param_config(793): i2c clock choice is invalid, please check flag and frequency
E (1322) i2c: i2c_set_pin(980): scl and sda gpio numbers are the same
ESP_ERROR_CHECK failed: esp_err_t 0x107 (ESP_ERR_TIMEOUT) at 0x400d6128
--- 0x400d6128: app_main at C:/Users/Lenovo T480s/Documents/VS Code-Projects/esp32SHT31-sepfy/esp32-sht31/main/app_main.c:17 (discriminator 1)

file: "./main/app_main.c" line 17
func: app_main
expression: sht31_read_temp_humi(&temp, &humi)

abort() was called at PC 0x400867bf on core 0
--- 0x400867bf: _esp_error_check_failed at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esp_system/esp_err.c:49



Backtrace: 0x40081936:0x3ffb3e40 0x400867c9:0x3ffb3e60 0x4008c8cd:0x3ffb3e80 0x400867bf:0x3ffb3ef0 0x400d6128:0x3ffb3f20 0x400e55ec:0x3ffb3f60 0x40087271:0x3ffb3f90
--- 0x40081936: panic_abort at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esp_system/panic.c:463
0x400867c9: esp_system_abort at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esp_system/port/esp_system_chip.c:92
0x4008c8cd: abort at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/newlib/abort.c:38
0x400867bf: _esp_error_check_failed at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esp_system/esp_err.c:49
0x400d6128: app_main at C:/Users/Lenovo T480s/Documents/VS Code-Projects/esp32SHT31-sepfy/esp32-sht31/main/app_main.c:17 (discriminator 1)
0x400e55ec: main_task at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/freertos/app_startup.c:208
0x40087271: vPortTaskWrapper at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/freertos/FreeRTOS-Kernel/portable/xtensa/port.c:134
```

Similar with this code too



```
#include "sht31.h"
```

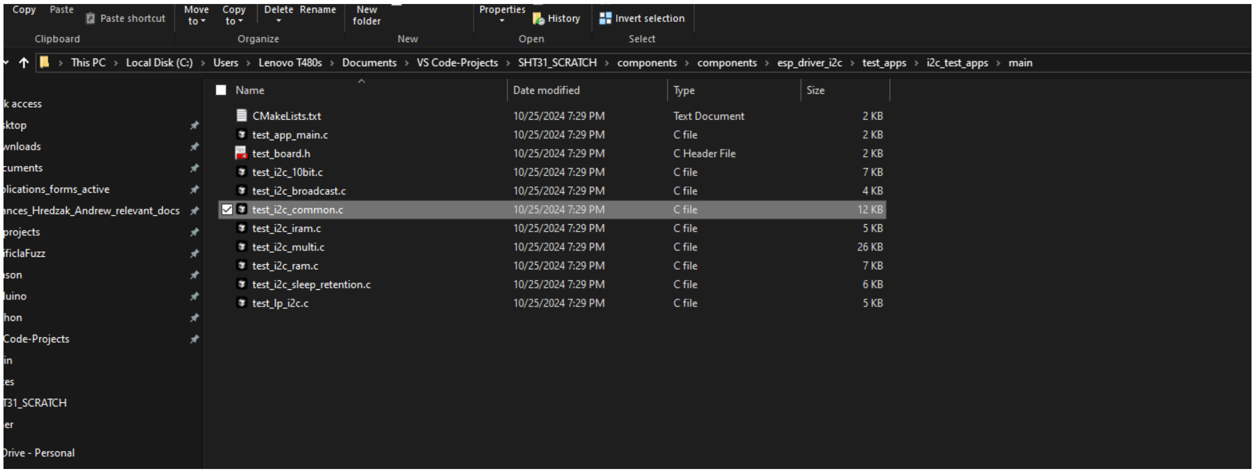
```
I (267) heap_init: At 3FFB2C68 len 0002D398 (180 KiB): DRAM
I (273) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (280) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (286) heap_init: At 4008E000 len 00012000 (72 KiB): IRAM
I (294) spi_flash: detected chip: generic
I (297) spi_flash: flash io: dio
W (301) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (312) main_task: Started on CPU0
I (322) main_task: Calling app_main()
E (322) i2c: i2c_param_config(793): i2c clock choice is invalid, please check flag and frequency
E (1322) i2c: i2c_set_pin(980): scl and sda gpio numbers are the same
ESP_ERROR_CHECK failed: esp_err_t 0x107 (ESP_ERR_TIMEOUT) at 0x400d6128
--- 0x400d6128: app_main at C:/Users/Lenovo T480s/Documents/VS Code-Projects/esp32SHT31-sepfy/esp32-sht31/main/app_main.c:18 (discriminator 1)

file: "./main/app_main.c" line 18
func: app_main
expression: sht31_read_temp_humi(&temp, &humi)

abort() was called at PC 0x400867bf on core 0
--- 0x400867bf: _esp_error_check_failed at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esp_system/esp_err.c:49



Backtrace: 0x40081936:0x3ffb3e40 0x400867c9:0x3ffb3e60 0x4008c8cd:0x3ffb3e80 0x400867bf:0x3ffb3ef0 0x400d6128:0x3ffb3f20 0x400e55ec:0x3ffb3f60 0x40087271:0x3ffb3f90
--- 0x40081936: panic_abort at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esp_system/panic.c:463
0x400867c9: esp_system_abort at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esp_system/port/esp_system_chip.c:92
0x4008c8cd: abort at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/newlib/abort.c:38
0x400867bf: _esp_error_check_failed at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esp_system/esp_err.c:49
0x400d6128: app_main at C:/Users/Lenovo T480s/Documents/VS Code-Projects/esp32SHT31-sepfy/esp32-sht31/main/app_main.c:18 (discriminator 1)
0x400e55ec: main_task at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/freertos/app_startup.c:208
0x40087271: vPortTaskWrapper at C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/freertos/FreeRTOS-Kernel/portable/xtensa/port.c:134
```

I thought i changed the pins

11/3/2024

I2c_master path

"C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_SCRATCH\components\components\esp_driver_i2c\include\driver\i2c_master.h"



This PC › Local Disk (C:) › Users › Lenovo T480s › Documents › VS Code-Projects › SHT31_SCRATCH › components › components › esp_driver_i2c › test_apps › i2c_test_apps › main

☐ Name	Date modified	Type	Size
CMakeLists.txt	10/25/2024 7:29 PM	Text Document	2 KB
test_app_main.c	10/25/2024 7:29 PM	C file	2 KB
test_board.h	10/25/2024 7:29 PM	C Header File	2 KB
test_i2c_10bit.c	10/25/2024 7:29 PM	C file	7 KB
test_i2c_broadcast.c	10/25/2024 7:29 PM	C file	4 KB
☑ test_i2c_common.c	10/25/2024 7:29 PM	C file	12 KB
test_i2c_iram.c	10/25/2024 7:29 PM	C file	5 KB
test_i2c_multi.c	10/25/2024 7:29 PM	C file	26 KB
test_i2c_ram.c	10/25/2024 7:29 PM	C file	7 KB
test_i2c_sleep_retention.c	10/25/2024 7:29 PM	C file	6 KB
test_lp_i2c.c	10/25/2024 7:29 PM	C file	5 KB

11/2/2024

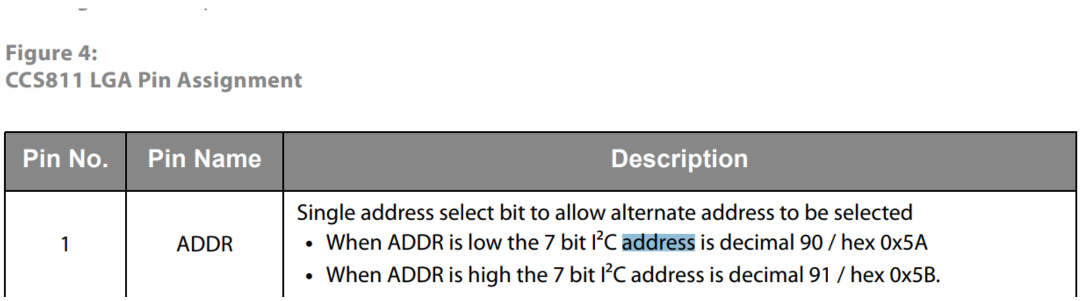
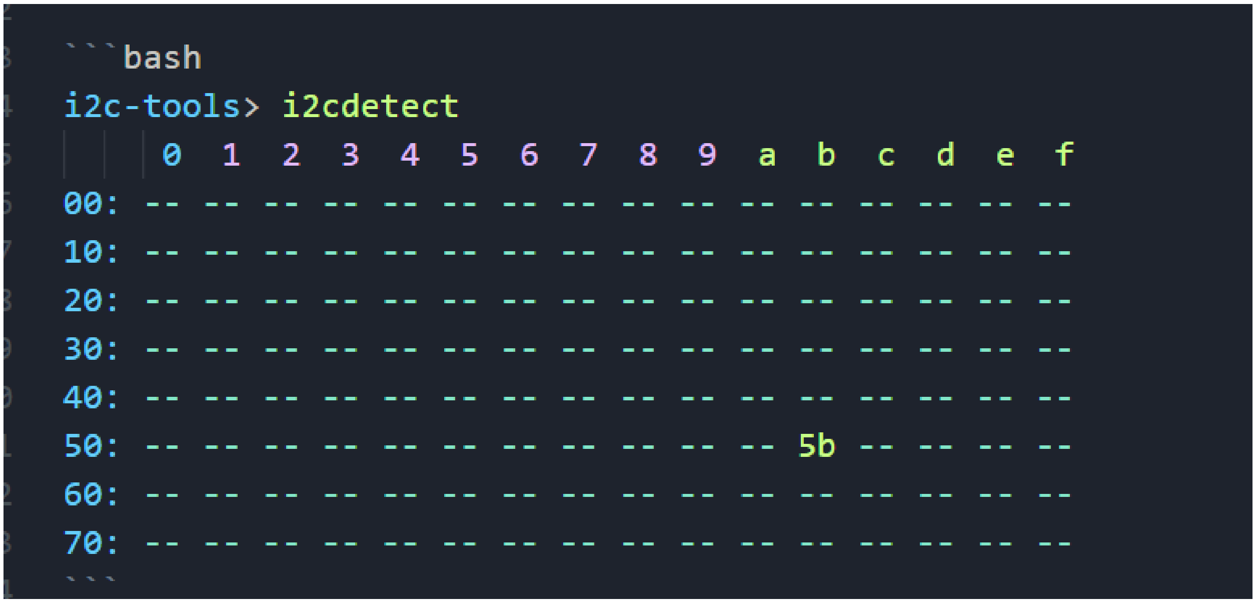


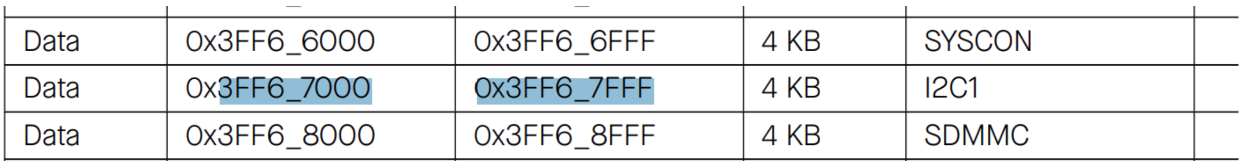
Figure 4:
CCS811 LGA Pin Assignment

Pin No.	Pin Name	Description
1	ADDR	Single address select bit to allow alternate address to be selected • When ADDR is low the 7 bit I²C address is decimal 90 / hex 0x5A • When ADDR is high the 7 bit I²C address is decimal 91 / hex 0x5B.



```bash
i2c-tools> i2cdetect
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- 5b -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

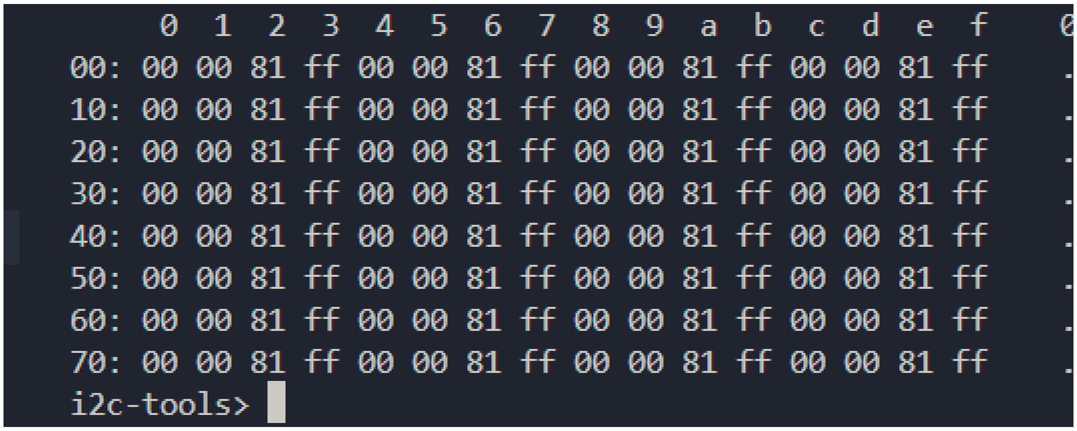
```
idf term output:
              3ffbf3e4     dec 1073476580
i2C0 highend: 3FF53FFF            1073037311
i2C1 low   3FF67000           1073115136  high  0x3FF67FFF 1073119231
```



Data	0x3FF6_6000	0x3FF6_6FFF	4 KB	SYSCON	
Data	0x3FF6_7000	0x3FF6_7FFF	4 KB	I2C1	
Data	0x3FF6_8000	0x3FF6_8FFF	4 KB	SDMMC	

In I2C1..

Weird cuz we selected 0



```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0
00: 00 00 81 ff 00 00 81 ff 00 00 81 ff 00 00 81 ff    .
10: 00 00 81 ff 00 00 81 ff 00 00 81 ff 00 00 81 ff    .
20: 00 00 81 ff 00 00 81 ff 00 00 81 ff 00 00 81 ff    .
30: 00 00 81 ff 00 00 81 ff 00 00 81 ff 00 00 81 ff    .
40: 00 00 81 ff 00 00 81 ff 00 00 81 ff 00 00 81 ff    .
50: 00 00 81 ff 00 00 81 ff 00 00 81 ff 00 00 81 ff    .
60: 00 00 81 ff 00 00 81 ff 00 00 81 ff 00 00 81 ff    .
70: 00 00 81 ff 00 00 81 ff 00 00 81 ff 00 00 81 ff    .
i2c-tools> 
```

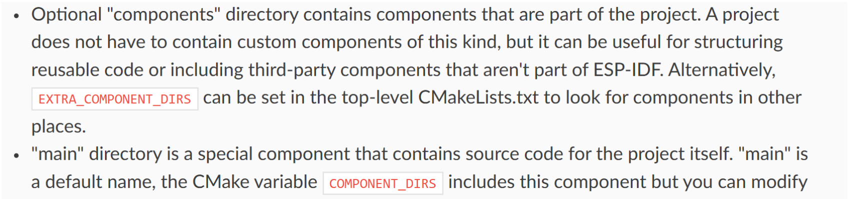
16x8 bytes = 128 bytes

10/27/2024

🗺️*TODO*

We have two directives:

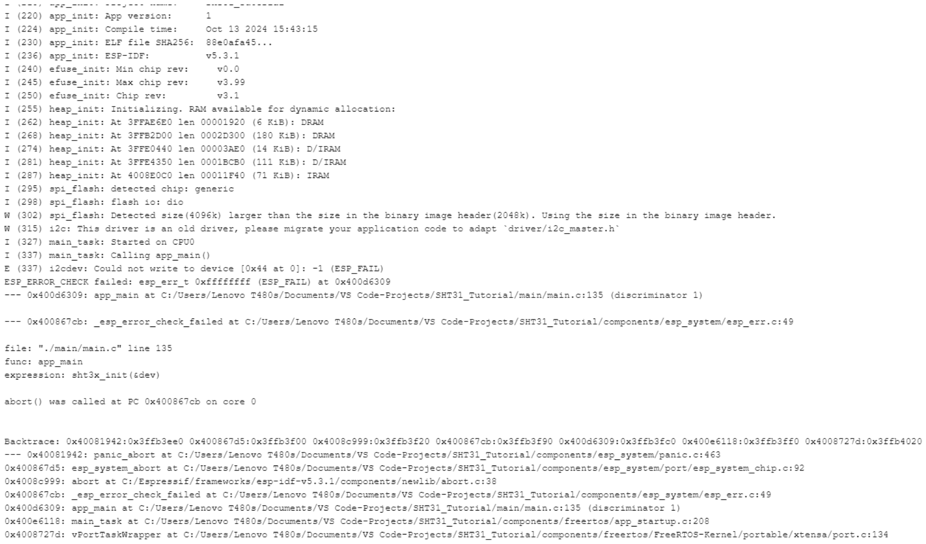
- ☐ SHT tutorial
    - ☐ Update i2c master



- Optional "components" directory contains components that are part of the project. A project does not have to contain custom components of this kind, but it can be useful for structuring reusable code or including third-party components that aren't part of ESP-IDF. Alternatively, `EXTRA_COMPONENT_DIRS` can be set in the top-level CMakeLists.txt to look for components in other places.
- "main" directory is a special component that contains source code for the project itself. "main" is a default name, the CMake variable `COMPONENT_DIRS` includes this component but you can modify

Try EXTRA_COMPONENT_DIRS

    - ☐ idf.py tools→ debug backtrack



```
I (220) app_init: App version:      1
I (224) app_init: Compile time:     Oct 13 2024 15:43:15
I (230) app_init: ELF file SHA256:  88e0afa45...
I (236) app_init: ESP-IDF:          v5.3.1
I (240) efuse_init: Min chip rev:     v0.0
I (245) efuse_init: Max chip rev:     v3.99
I (250) efuse_init: Chip rev:         v3.1
I (255) heap_init: Initializing. RAM available for dynamic allocation:
I (262) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (268) heap_init: At 3FFB2D00 len 0002D300 (180 KiB): DRAM
I (274) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (281) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (287) heap_init: At 4008E0C0 len 00011F40 (71 KiB): IRAM
I (295) spi_flash: detected chip: generic
I (298) spi_flash: flash io: dio
W (302) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
W (315) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (327) main_task: Started on CPU0
I (337) main_task: Calling app_main()
E (337) i2cdev: Could not write to device [0x44 at 0]: -1 (ESP_FAIL)
ESP_ERROR_CHECK failed: esp_err_t 0xffffffff (ESP_FAIL) at 0x400d6309
--- 0x400d6309: app_main at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/main/main.c:135 (discriminator 1)

--- 0x400867cb: _esp_error_check_failed at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/esp_system/esp_err.c:49

file: "./main/main.c" line 135
func: app_main
expression: sht3x_init(&dev)

abort() was called at PC 0x400867cb on core 0


Backtrace: 0x40081942:0x3ffb3ee0 0x400867d5:0x3ffb3f00 0x4008c999:0x3ffb3f20 0x400867cb:0x3ffb3f90 0x400d6309:0x3ffb3fc0 0x400e6118:0x3ffb3ff0 0x4008727d:0x3ffb4020
--- 0x40081942: panic_abort at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/esp_system/panic.c:463
0x400867d5: esp_system_abort at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/esp_system/port/esp_system_chip.c:92
0x4008c999: abort at C:/Espressif/frameworks/esp-idf-v5.3.1/components/newlib/abort.c:38
0x400867cb: _esp_error_check_failed at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/esp_system/esp_err.c:49
0x400d6309: app_main at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/main/main.c:135 (discriminator 1)
0x400e6118: main_task at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/freertos/app_startup.c:208
0x4008727d: vPortTaskWrapper at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/freertos/FreeRTOS-Kernel/portable/xtensa/port.c:134
```

☐ SHT from scratch

- ☐ Layout all files needed
- ☐ File structure
- ☑ ~~Tools tutorial runthough all 5 commands for SHT~~
- ☐ Read SHT manual
- ☑ ~~Write cheat sheet for SHT31 like the tools readme file for CCS811 sensor~~
- ☐ Write sht31.h headerfile

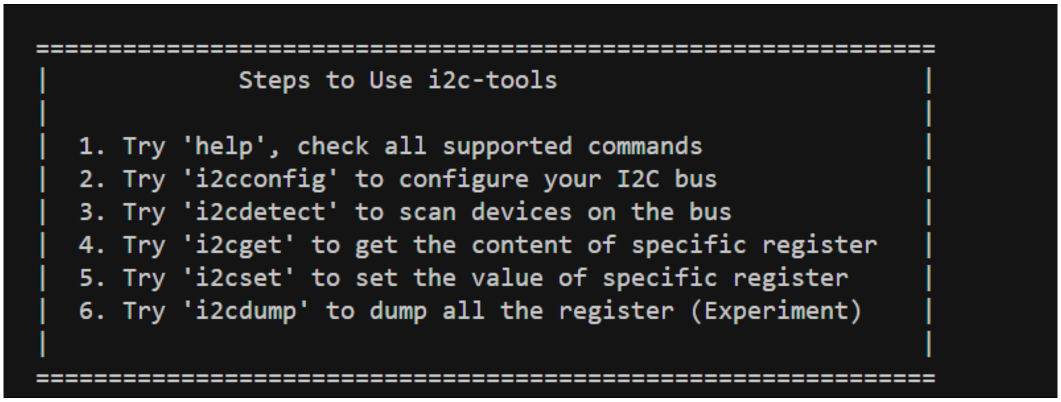
Commands can be found in nature scanner: SHT_command_readme.md

`high`	`enabeled`	`0x2C06`
`medium`	`enabeled`	`0x2C0D`
`low`	`enabeled`	`0x2C10`
`high`	`disabled`	`0x2400`
`medium`	`disabled`	`0x240B`
`low`	`disabled`	`0x2416`

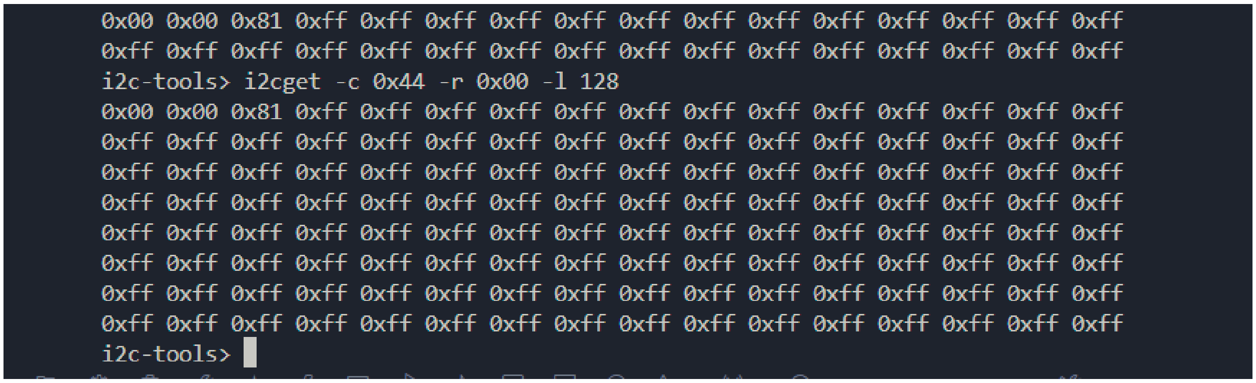
I2cconfig
Port = 0 freq = 1000000?? SDA= 2 SCL = 4

I2cset `0x2C06`

I2cget ??

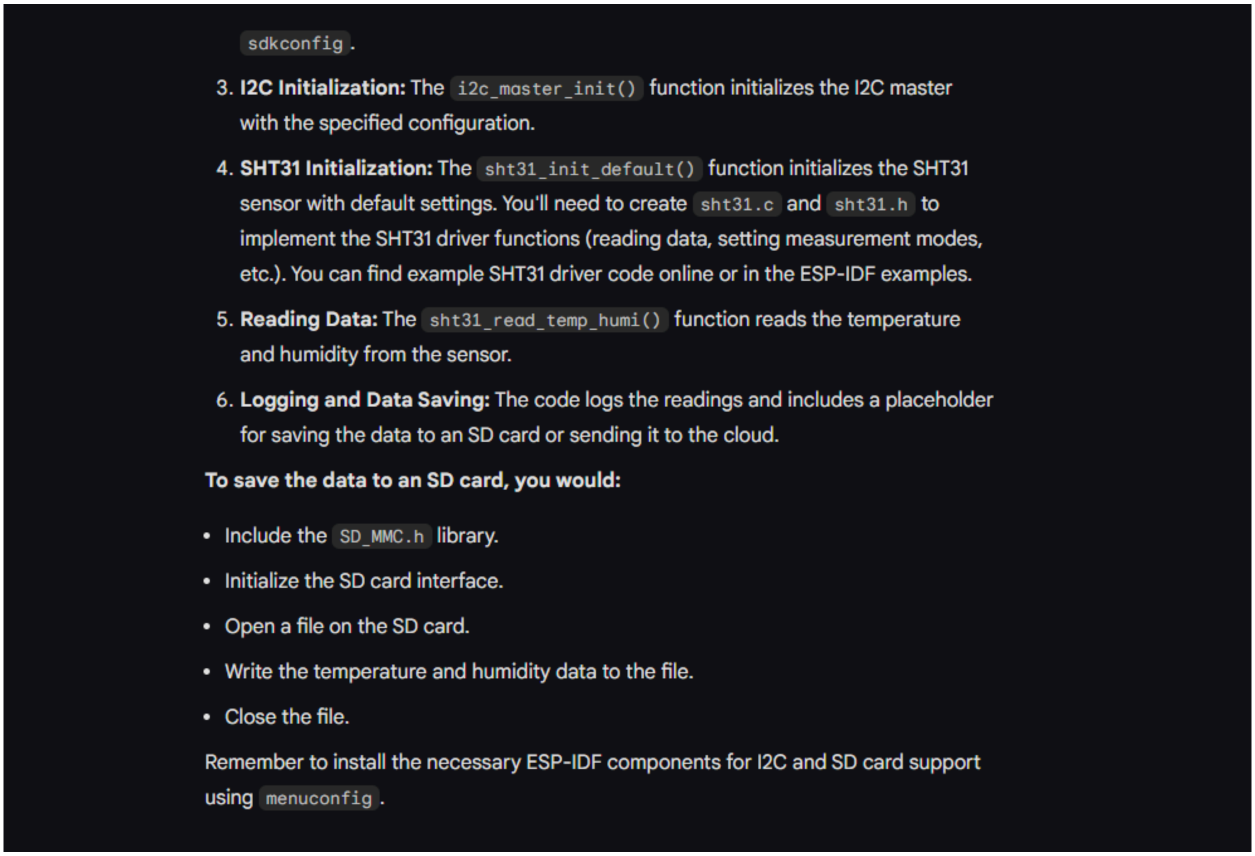


```
==============================================================
|             Steps to Use i2c-tools                         |
|                                                            |
|  1. Try 'help', check all supported commands               |
|  2. Try 'i2cconfig' to configure your I2C bus              |
|  3. Try 'i2cdetect' to scan devices on the bus             |
|  4. Try 'i2cget' to get the content of specific register   |
|  5. Try 'i2cset' to set the value of specific register     |
|  6. Try 'i2cdump' to dump all the register (Experiment)    |
|                                                            |
==============================================================
```



```
0x00 0x00 0x81 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
i2c-tools> i2cget -c 0x44 -r 0x00 -l 128
0x00 0x00 0x81 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
i2c-tools> 
```

Start of SHT + save to micro CD card



`sdkconfig`.

3. **I2C Initialization:** The `i2c_master_init()` function initializes the I2C master with the specified configuration.
4. **SHT31 Initialization:** The `sht31_init_default()` function initializes the SHT31 sensor with default settings. You'll need to create `sht31.c` and `sht31.h` to implement the SHT31 driver functions (reading data, setting measurement modes, etc.). You can find example SHT31 driver code online or in the ESP-IDF examples.
5. **Reading Data:** The `sht31_read_temp_humi()` function reads the temperature and humidity from the sensor.
6. **Logging and Data Saving:** The code logs the readings and includes a placeholder for saving the data to an SD card or sending it to the cloud.

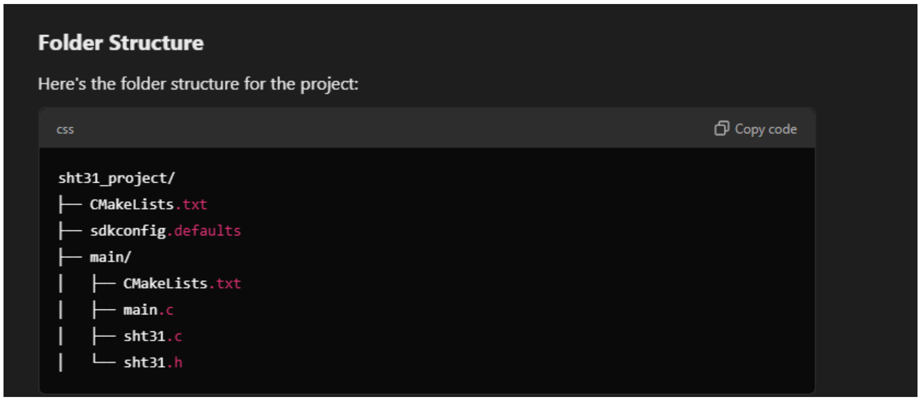
To save the data to an SD card, you would:

- Include the `SD_MMC.h` library.
- Initialize the SD card interface.
- Open a file on the SD card.
- Write the temperature and humidity data to the file.
- Close the file.

Remember to install the necessary ESP-IDF components for I2C and SD card support using `menuconfig`.

01 preview NatureScan 1 shot:

ChatGPT

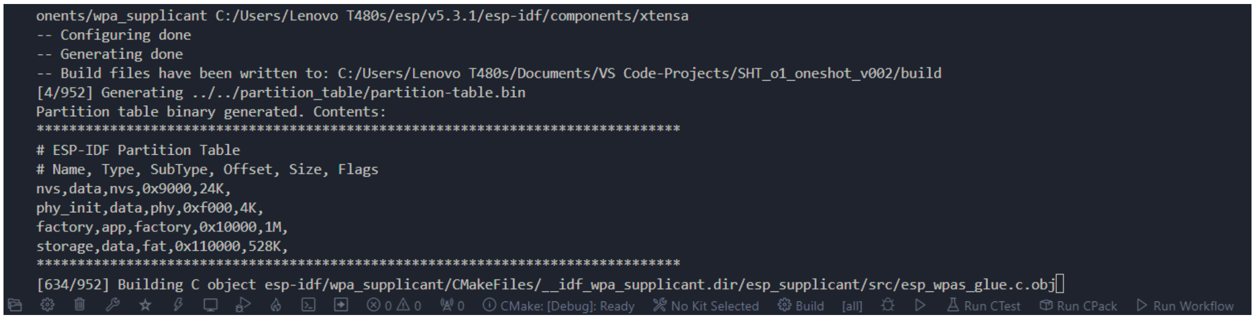


Folder Structure

Here's the folder structure for the project:

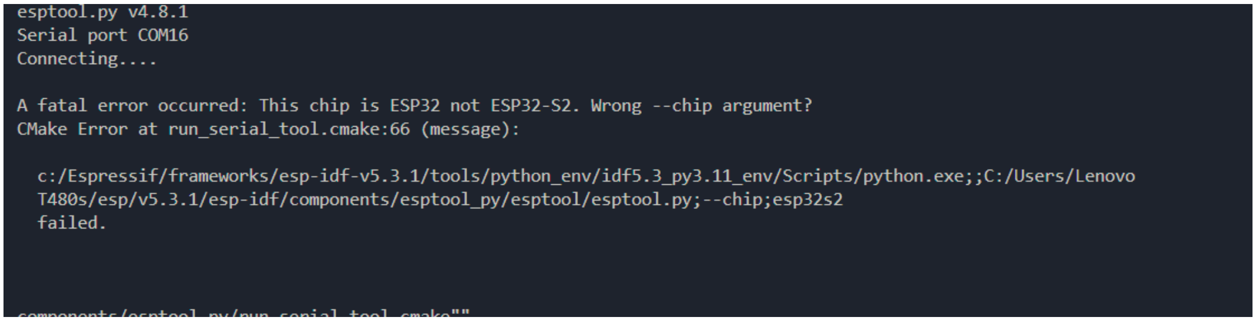
```css
sht31_project/
├── CMakeLists.txt
├── sdkconfig.defaults
├── main/
│   ├── CMakeLists.txt
│   ├── main.c
│   ├── sht31.c
│   └── sht31.h
```

Made a new project from the tools template
Great idea. I need to do that for my scratch file

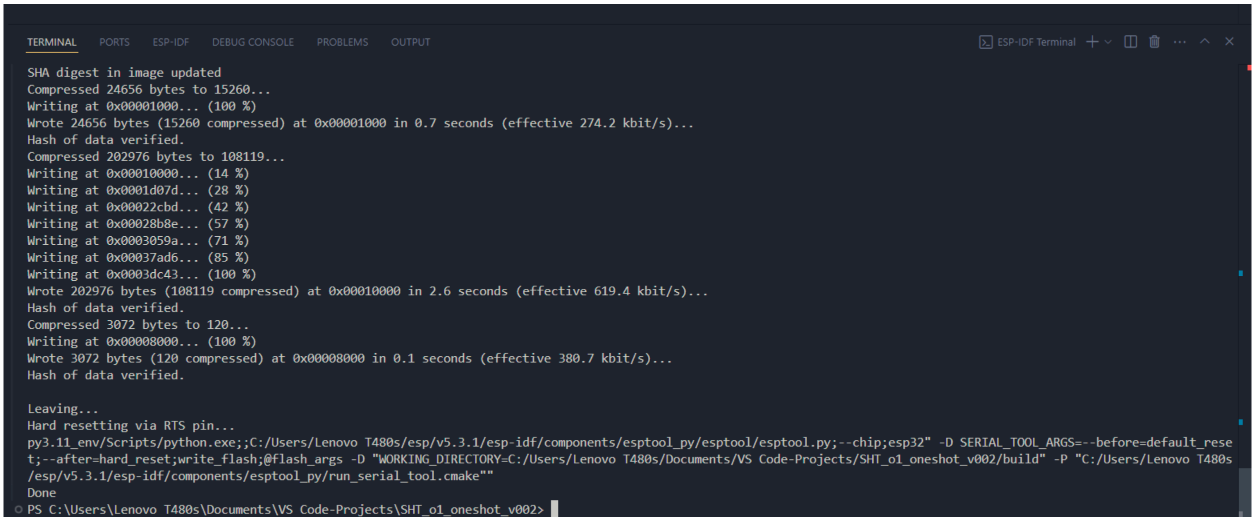


```
onents/wpa_supplicant C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/xtensa
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT_o1_oneshot_v002/build
[4/952] Generating ../../partition_table/partition-table.bin
Partition table binary generated. Contents:
*******************************************************************************
# ESP-IDF Partition Table
# Name, Type, SubType, Offset, Size, Flags
nvs,data,nvs,0x9000,24K,
phy_init,data,phy,0xf000,4K,
factory,app,factory,0x10000,1M,
storage,data,fat,0x110000,528K,
*******************************************************************************
[634/952] Building C object esp-idf/wpa_supplicant/CMakeFiles/__idf_wpa_supplicant.dir/esp_supplicant/src/esp_wpas_glue.c.obj
```

It build and is currently flashing

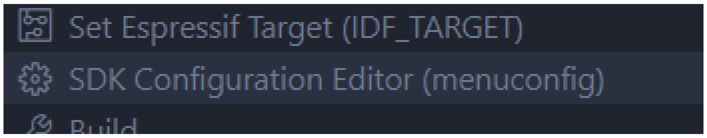


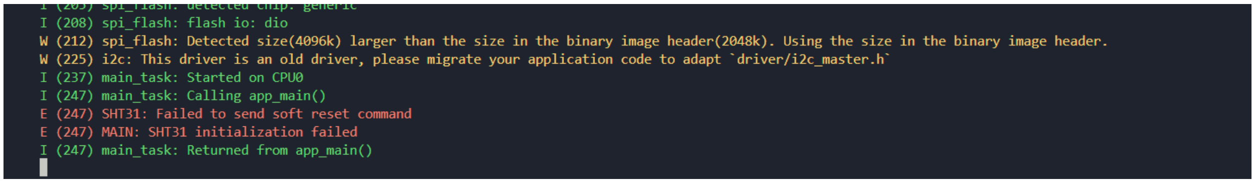
```
esptool.py v4.8.1
Serial port COM16
Connecting....

A fatal error occurred: This chip is ESP32 not ESP32-S2. Wrong --chip argument?
CMake Error at run_serial_tool.cmake:66 (message):

  c:/Espressif/frameworks/esp-idf-v5.3.1/tools/python_env/idf5.3_py3.11_env/Scripts/python.exe;;C:/Users/Lenovo
  T480s/esp/v5.3.1/esp-idf/components/esptool_py/esptool/esptool.py;--chip;esp32s2
  failed.



components/esptool_py/run_serial_tool.cmake""
```



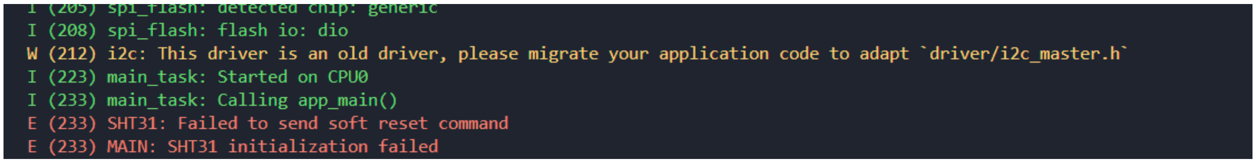
```
SHA digest in image updated
Compressed 24656 bytes to 15260...
Writing at 0x00001000... (100 %)
Wrote 24656 bytes (15260 compressed) at 0x00001000 in 0.7 seconds (effective 274.2 kbit/s)...
Hash of data verified.
Compressed 202976 bytes to 108119...
Writing at 0x00010000... (14 %)
Writing at 0x0001d07d... (28 %)
Writing at 0x00022cbd... (42 %)
Writing at 0x00028b8e... (57 %)
Writing at 0x0003059a... (71 %)
Writing at 0x00037ad6... (85 %)
Writing at 0x0003dc43... (100 %)
Wrote 202976 bytes (108119 compressed) at 0x00010000 in 2.6 seconds (effective 619.4 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 120...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (120 compressed) at 0x00008000 in 0.1 seconds (effective 380.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
py3.11_env/Scripts/python.exe;;C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/esptool_py/esptool/esptool.py;--chip;esp32" -D SERIAL_TOOL_ARGS=--before=default_rese
t;--after=hard_reset;write_flash;@flash_args -D "WORKING_DIRECTORY=C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT_o1_oneshot_v002/build" -P "C:/Users/Lenovo T480s
/esp/v5.3.1/esp-idf/components/esptool_py/run_serial_tool.cmake""
Done
PS C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT_o1_oneshot_v002>
```

I had it set to esp32s2 and not esp32
I fixed it by going to target and then configuration. There is a bash command too


Set Espressif Target (IDF_TARGET)
SDK Configuration Editor (menuconfig)
Build

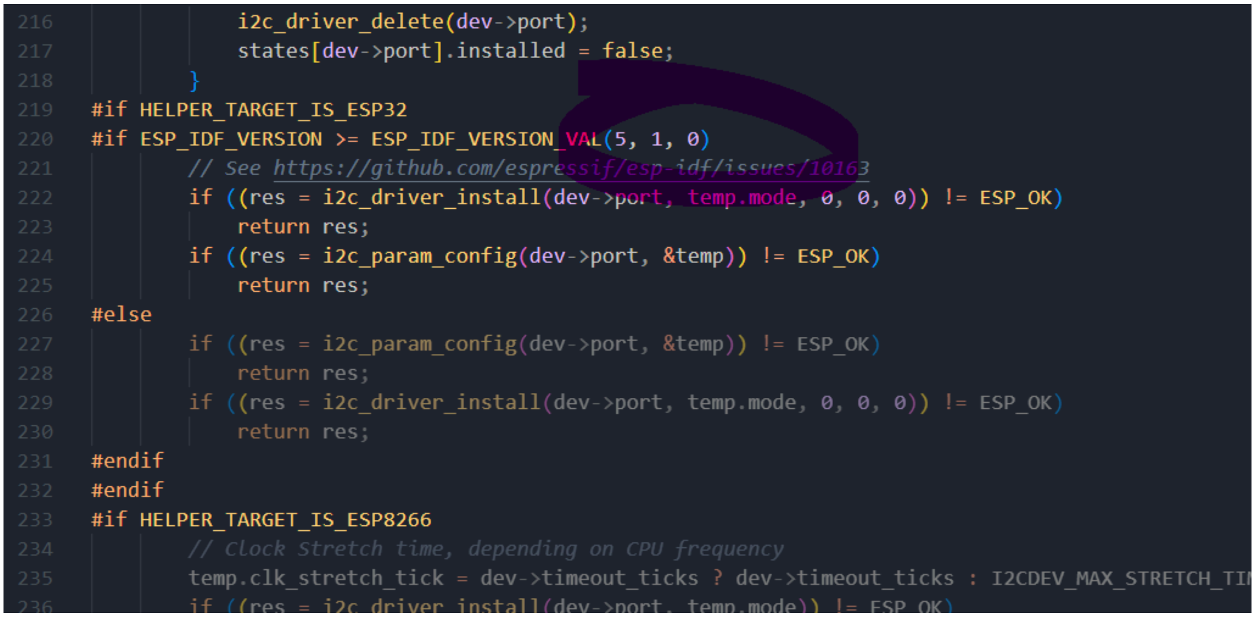


```
I (205) spi_flash: detected chip: generic
I (208) spi_flash: flash io: dio
W (212) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
W (225) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (237) main_task: Started on CPU0
I (247) main_task: Calling app_main()
E (247) SHT31: Failed to send soft reset command
E (247) MAIN: SHT31 initialization failed
I (247) main_task: Returned from app_main()
```



```
I (205) spi_flash: detected chip: generic
I (208) spi_flash: flash io: dio
W (212) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (223) main_task: Started on CPU0
I (233) main_task: Calling app_main()
E (233) SHT31: Failed to send soft reset command
E (233) MAIN: SHT31 initialization failed
```

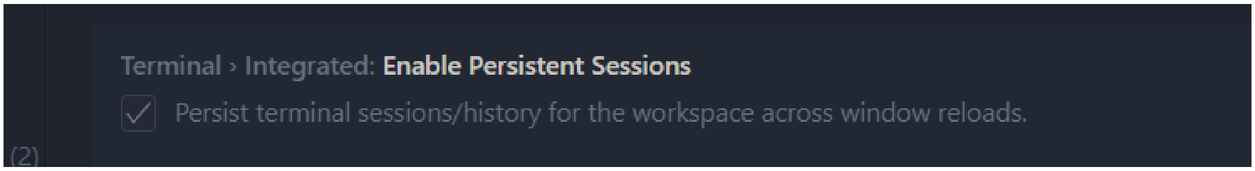
10/26/2024



```
            i2c_driver_delete(dev->port);
            states[dev->port].installed = false;
        }
#if HELPER_TARGET_IS_ESP32
#if ESP_IDF_VERSION >= ESP_IDF_VERSION_VAL(5, 1, 0)
        // See https://github.com/espressif/esp-idf/issues/10163
        if ((res = i2c_driver_install(dev->port, temp.mode, 0, 0, 0)) != ESP_OK)
            return res;
        if ((res = i2c_param_config(dev->port, &temp)) != ESP_OK)
            return res;
#else
        if ((res = i2c_param_config(dev->port, &temp)) != ESP_OK)
            return res;
        if ((res = i2c_driver_install(dev->port, temp.mode, 0, 0, 0)) != ESP_OK)
            return res;
#endif
#endif
#if HELPER_TARGET_IS_ESP8266
        // Clock Stretch time, depending on CPU frequency
        temp.clk_stretch_tick = dev->timeout_ticks ? dev->timeout_ticks : I2CDEV_MAX_STRETCH_TI
        if ((res = i2c_driver_install(dev->port, temp.mode)) != ESP_OK)
```

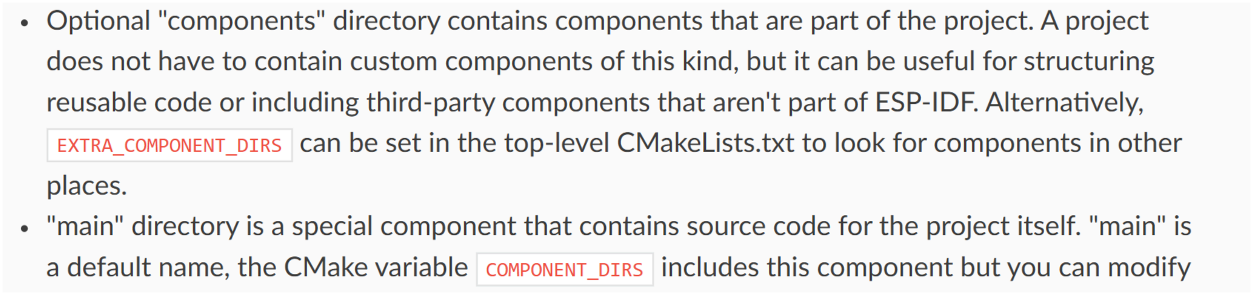
I think it needs 5.1.0 and we have 5.3.1

Nonono
It said >=



Terminal › Integrated: Enable Persistent Sessions

☑ Persist terminal sessions/history for the workspace across window reloads.



- Optional "components" directory contains components that are part of the project. A project does not have to contain custom components of this kind, but it can be useful for structuring reusable code or including third-party components that aren't part of ESP-IDF. Alternatively, `EXTRA_COMPONENT_DIRS` can be set in the top-level CMakeLists.txt to look for components in other places.
- "main" directory is a special component that contains source code for the project itself. "main" is a default name, the CMake variable `COMPONENT_DIRS` includes this component but you can modify

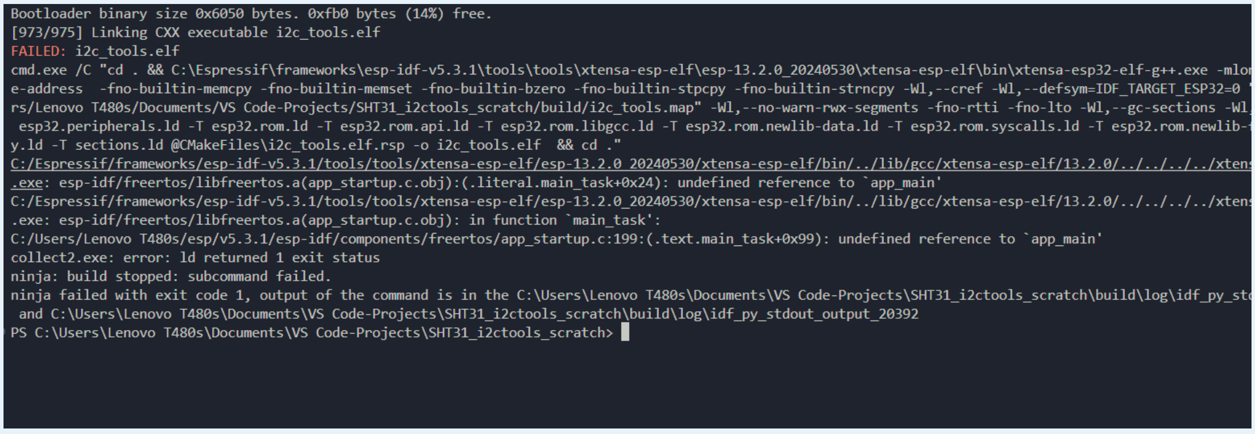
Could link it to extra components

10/25/2024

The recommended value for pull-up resistors usually ranges from:

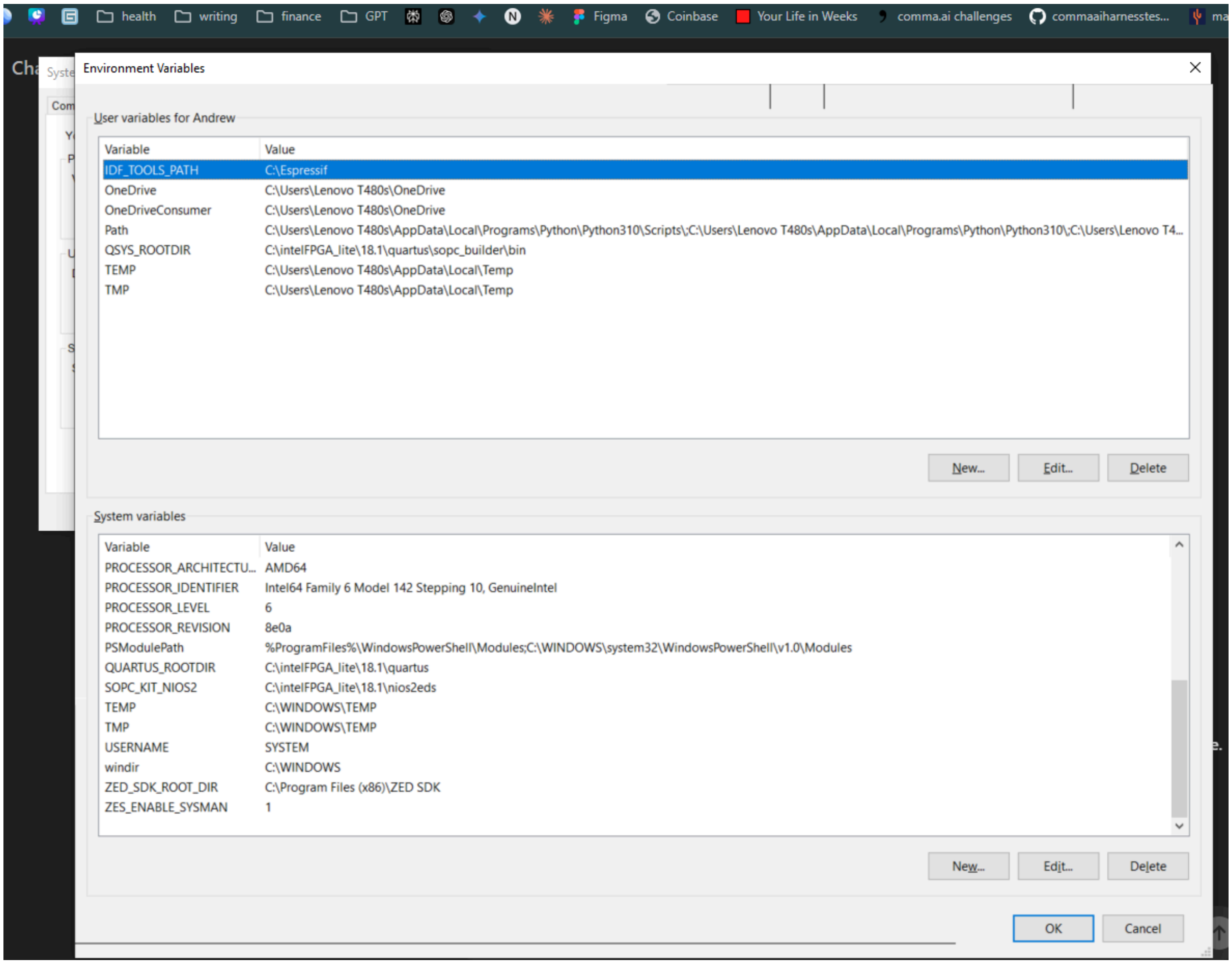
1K Ohms -10K Ohms.

the higher the frequency, the smaller the pull-up resistor should be (but not less than 1 KOhms)



```
Bootloader binary size 0x6050 bytes. 0xfb0 bytes (14%) free.
[973/975] Linking CXX executable i2c_tools.elf
FAILED: i2c_tools.elf
cmd.exe /C "cd . && C:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\xtensa-esp-elf\esp-13.2.0_20240530\xtensa-esp-elf\bin\xtensa-esp32-elf-g++.exe -mlon
e-address  -fno-builtin-memcpy -fno-builtin-memset -fno-builtin-bzero -fno-builtin-stpcpy -fno-builtin-strncpy -Wl,--cref -Wl,--defsym=IDF_TARGET_ESP32=0
rs/Lenovo T480s/Documents/VS Code-Projects/SHT31_i2ctools_scratch/build/i2c_tools.map" -Wl,--no-warn-rwx-segments -fno-rtti -fno-lto -Wl,--gc-sections -Wl,
 esp32.peripherals.ld -T esp32.rom.ld -T esp32.rom.api.ld -T esp32.rom.libgcc.ld -T esp32.rom.newlib-data.ld -T esp32.rom.syscalls.ld -T esp32.rom.newlib-
y.ld -T sections.ld @CMakeFiles\i2c_tools.elf.rsp -o i2c_tools.elf  && cd ."
C:/Espressif/frameworks/esp-idf-v5.3.1/tools/tools/xtensa-esp-elf/esp-13.2.0_20240530/xtensa-esp-elf/bin/../lib/gcc/xtensa-esp-elf/13.2.0/../../../../xtens
.exe: esp-idf/freertos/libfreertos.a(app_startup.c.obj):(.literal.main_task+0x24): undefined reference to `app_main'
C:/Espressif/frameworks/esp-idf-v5.3.1/tools/tools/xtensa-esp-elf/esp-13.2.0_20240530/xtensa-esp-elf/bin/../lib/gcc/xtensa-esp-elf/13.2.0/../../../../xtens
.exe: esp-idf/freertos/libfreertos.a(app_startup.c.obj): in function `main_task':
C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf/components/freertos/app_startup.c:199:(.text.main_task+0x99): undefined reference to `app_main'
collect2.exe: error: ld returned 1 exit status
ninja: build stopped: subcommand failed.
ninja failed with exit code 1, output of the command is in the C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_i2ctools_scratch\build\log\idf_py_std
 and C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_i2ctools_scratch\build\log\idf_py_stdout_output_20392
PS C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_i2ctools_scratch> 
```

Needs app main!!



User variables for Andrew

Variable	Value
IDF_TOOLS_PATH	C:\Espressif
OneDrive	C:\Users\Lenovo T480s\OneDrive
OneDriveConsumer	C:\Users\Lenovo T480s\OneDrive
Path	C:\Users\Lenovo T480s\AppData\Local\Programs\Python\Python310\Scripts\;C:\Users\Lenovo T480s\AppData\Local\Programs\Python\Python310\;C:\Users\Lenovo T4...
QSYS_ROOTDIR	C:\intelFPGA_lite\18.1\quartus\sopc_builder\bin
TEMP	C:\Users\Lenovo T480s\AppData\Local\Temp
TMP	C:\Users\Lenovo T480s\AppData\Local\Temp

System variables

Variable	Value
PROCESSOR_ARCHITECTU...	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 10, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	8e0a
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules;C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules
QUARTUS_ROOTDIR	C:\intelFPGA_lite\18.1\quartus
SOPC_KIT_NIOS2	C:\intelFPGA_lite\18.1\nios2eds
TEMP	C:\WINDOWS\TEMP
TMP	C:\WINDOWS\TEMP
USERNAME	SYSTEM
windir	C:\WINDOWS
ZED_SDK_ROOT_DIR	C:\Program Files (x86)\ZED SDK
ZES_ENABLE_SYSMAN	1

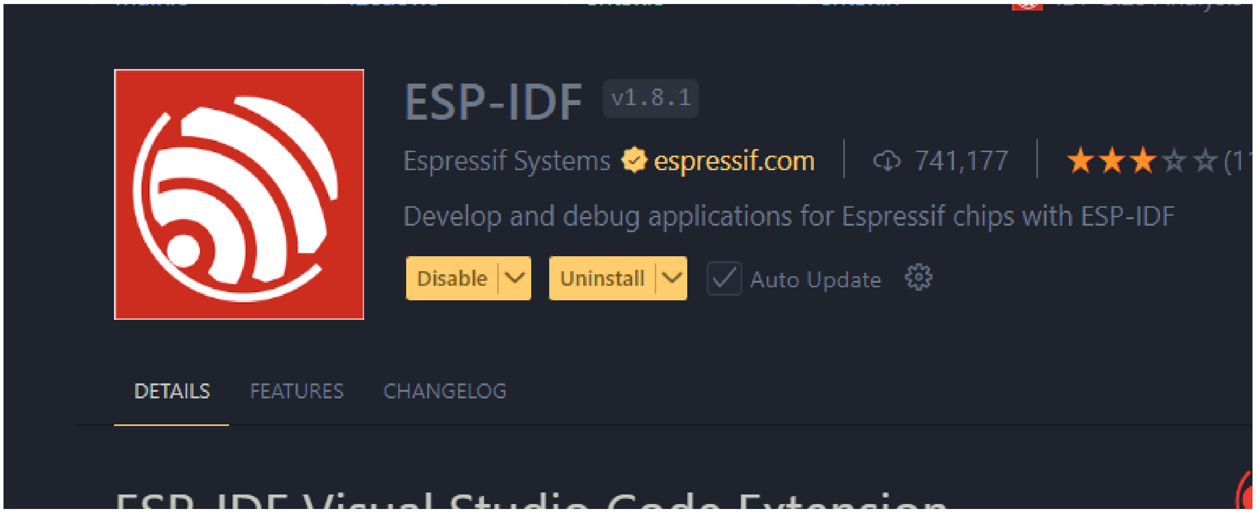
Old variable
No idf path

eeprom:

```
>>> print(os.getenv('IDF_PATH'))
C:\Users\Lenovo T480s\esp\v5.3.1\esp-idf
>>> print(os.getenv('IDF_TOOLS_PATH'))
c:\Espressif\frameworks\esp-idf-v5.3.1\tools
```

SHT TUTORIAL:

```
>>> print(os.getenv('IDF_PATH'))
None
>>> print(os.getenv('IDF_TOOLS_PATH'))
C:\Espressif
```

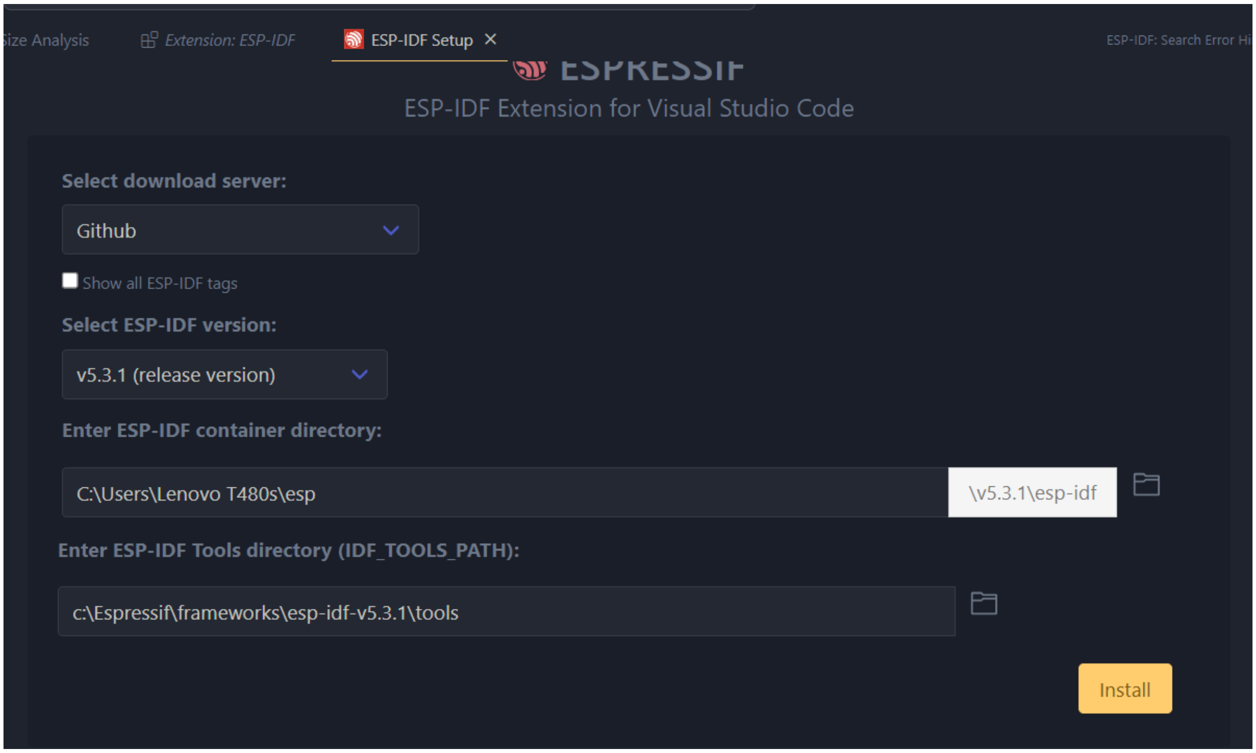


ESP-IDF v1.8.1

Espressif Systems espressif.com | 741,177 | ★★★☆☆

Develop and debug applications for Espressif chips with ESP-IDF

Why this old version?? No this is the vscode plugin version



ESP-IDF Extension for Visual Studio Code

Select download server: Github

☐ Show all ESP-IDF tags

Select ESP-IDF version: v5.3.1 (release version)

Enter ESP-IDF container directory: C:\Users\Lenovo T480s\esp \v5.3.1\esp-idf

Enter ESP-IDF Tools directory (IDF_TOOLS_PATH): c:\Espressif\frameworks\esp-idf-v5.3.1\tools

New install locations

Gemini suggestion:

IDF_PATH

- **Purpose:** This environment variable tells the ESP-IDF extension and build system where to find the main ESP-IDF framework files.

Suggested Value:

```
C:\Users\Lenovo T480s\esp\esp-idf-v5.3.1
```

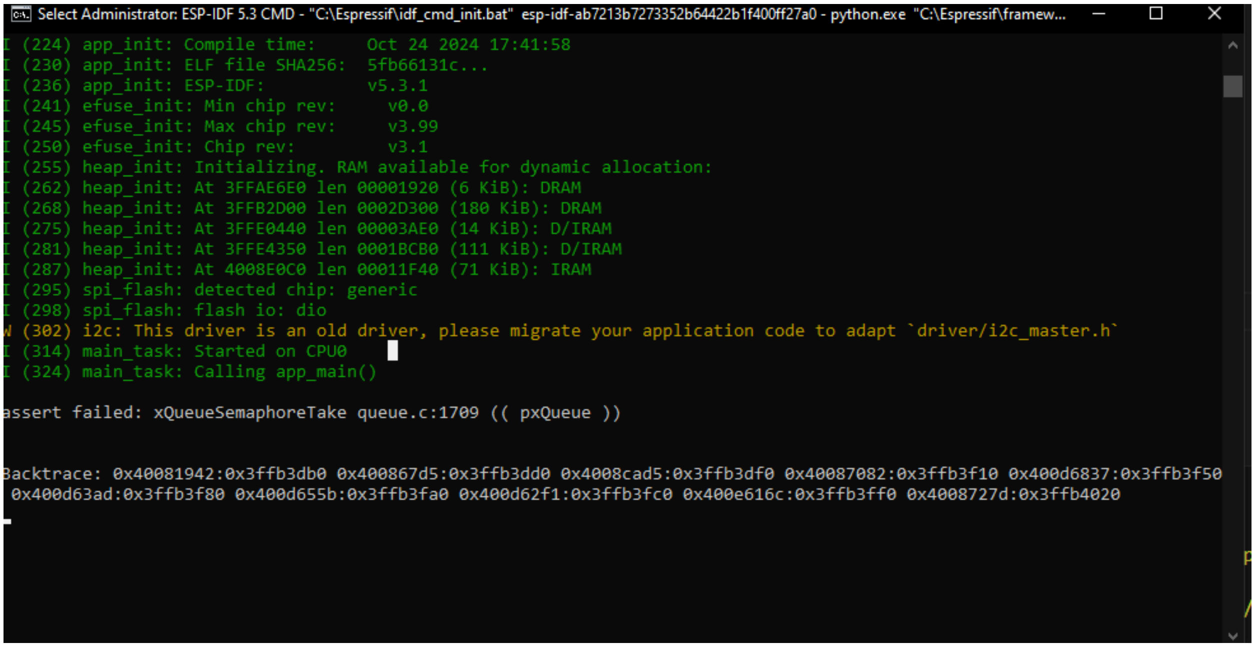
-
    - **Your Path:** This assumes you installed the ESP-IDF in the container directory you mentioned earlier (`C:\Users\Lenovo T480s\esp`).
    - **Version-Specific:** It includes the specific version of the ESP-IDF you're using (`esp-idf-v5.3.1`). This is important if you have multiple ESP-IDF versions installed.

IDF_TOOLS_PATH

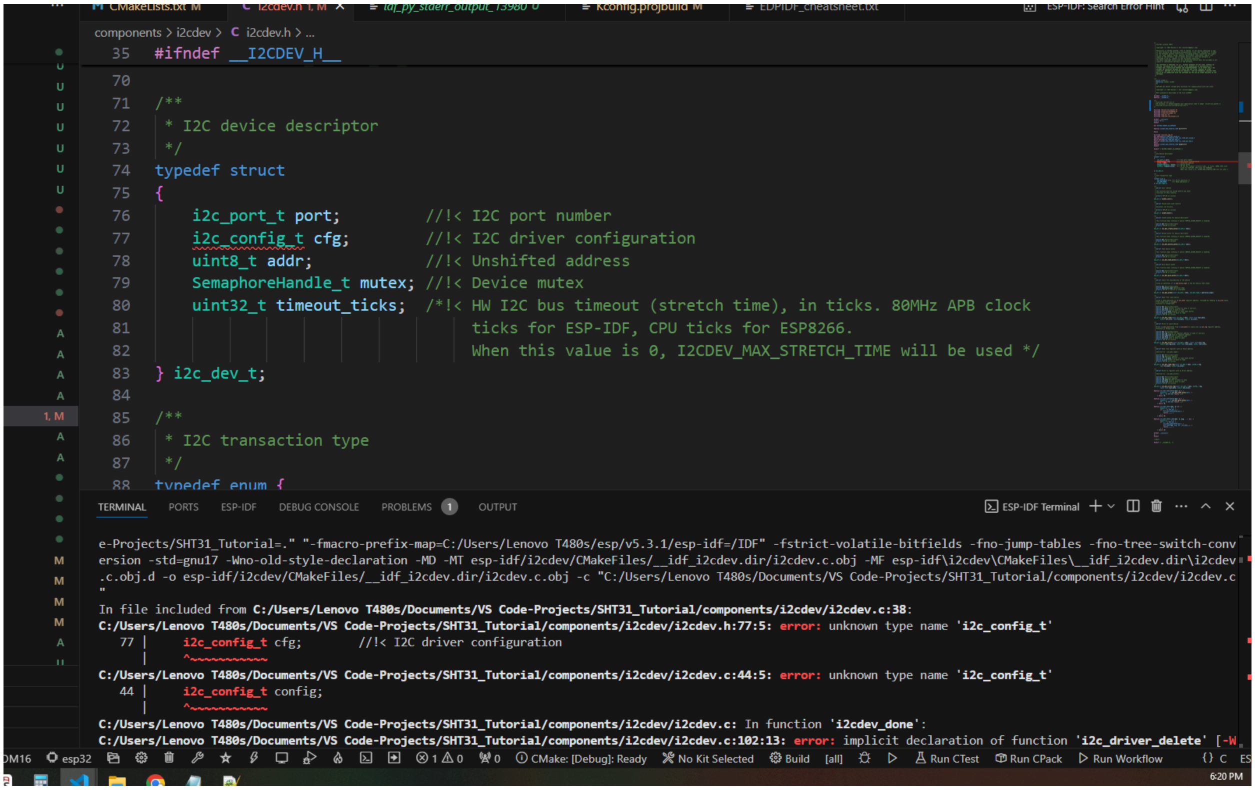
- **Purpose:** This environment variable points to the directory containing the ESP-IDF tools (compiler toolchain, debugger, etc.).
- **Suggested Value:**

```
C:\Espressif frameworks\esp-idf-v5.3.1\tools
```

10/24/2024


Select Administrator: ESP-IDF 5.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-ab7213b7273352b64422b1f400ff27a0 - python.exe "C:\Espressif\framew...

```
I (224) app_init: Compile time:     Oct 24 2024 17:41:58
I (230) app_init: ELF file SHA256:  5fb66131c...
I (236) app_init: ESP-IDF:          v5.3.1
I (241) efuse_init: Min chip rev:     v0.0
I (245) efuse_init: Max chip rev:     v3.99
I (250) efuse_init: Chip rev:         v3.1
I (255) heap_init: Initializing. RAM available for dynamic allocation:
I (262) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (268) heap_init: At 3FFB2D00 len 0002D300 (180 KiB): DRAM
I (275) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (281) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (287) heap_init: At 4008E0C0 len 00011F40 (71 KiB): IRAM
I (295) spi_flash: detected chip: generic
I (298) spi_flash: flash io: dio
W (302) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (314) main_task: Started on CPU0
I (324) main_task: Calling app_main()

assert failed: xQueueSemaphoreTake queue.c:1709 (( pxQueue ))


Backtrace: 0x40081942:0x3ffb3db0 0x400867d5:0x3ffb3dd0 0x4008cad5:0x3ffb3df0 0x40087082:0x3ffb3f10 0x400d6837:0x3ffb3f50
 0x400d63ad:0x3ffb3f80 0x400d655b:0x3ffb3fa0 0x400d62f1:0x3ffb3fc0 0x400e616c:0x3ffb3ff0 0x4008727d:0x3ffb4020
```

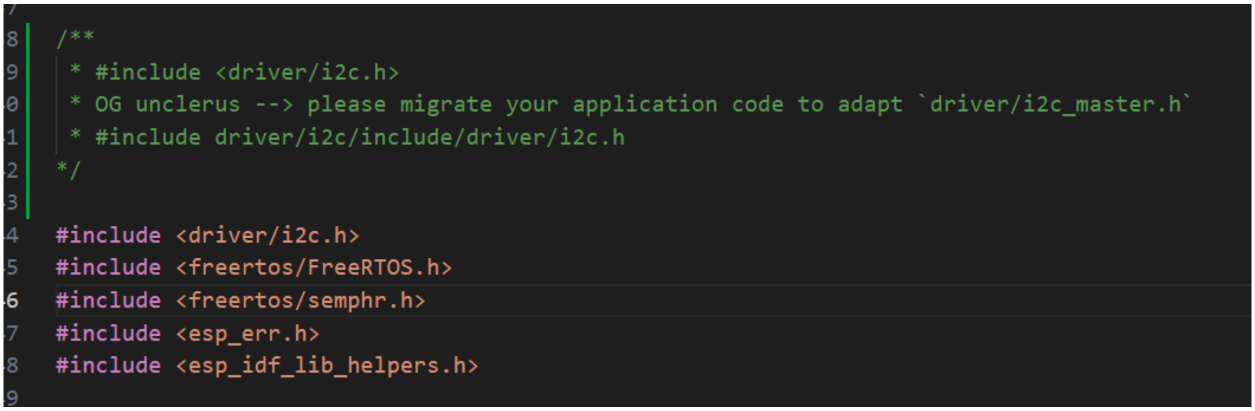


```c
#ifndef __I2CDEV_H__

/**
 * I2C device descriptor
 */
typedef struct
{
    i2c_port_t port;         //!< I2C port number
    i2c_config_t cfg;        //!< I2C driver configuration
    uint8_t addr;            //!< Unshifted address
    SemaphoreHandle_t mutex; //!< Device mutex
    uint32_t timeout_ticks;  /*!< HW I2C bus timeout (stretch time), in ticks. 80MHz APB clock
                                  ticks for ESP-IDF, CPU ticks for ESP8266.
                                  When this value is 0, I2CDEV_MAX_STRETCH_TIME will be used */
} i2c_dev_t;

/**
 * I2C transaction type
 */
typedef enum {
```

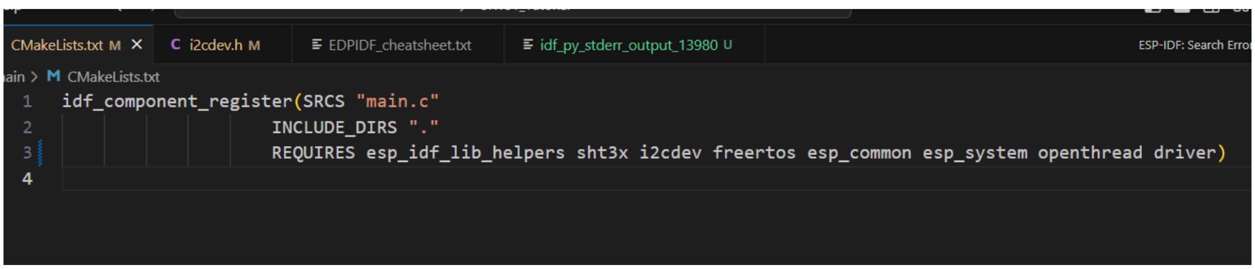
```
e-Projects/SHT31_Tutorial=." "-fmacro-prefix-map=C:/Users/Lenovo T480s/esp/v5.3.1/esp-idf=/IDF" -fstrict-volatile-bitfields -fno-jump-tables -fno-tree-switch-conv
ersion -std=gnu17 -Wno-old-style-declaration -MD -MT esp-idf/i2cdev/CMakeFiles/__idf_i2cdev.dir/i2cdev.c.obj -MF esp-idf\i2cdev\CMakeFiles\__idf_i2cdev.dir\i2cdev
.c.obj.d -o esp-idf/i2cdev/CMakeFiles/__idf_i2cdev.dir/i2cdev.c.obj -c "C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/i2cdev/i2cdev.c
"
In file included from C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/i2cdev/i2cdev.c:38:
C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/i2cdev/i2cdev.h:77:5: error: unknown type name 'i2c_config_t'
   77 |     i2c_config_t cfg;        //!< I2C driver configuration
      |     ^~~~~~~~~~~~
C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/i2cdev/i2cdev.c:44:5: error: unknown type name 'i2c_config_t'
   44 |     i2c_config_t config;
      |     ^~~~~~~~~~~~
C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/i2cdev/i2cdev.c: In function 'i2cdev_done':
C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/i2cdev/i2cdev.c:102:13: error: implicit declaration of function 'i2c_driver_delete' [-W
```

——---------

Config



```c
/**
 * #include <driver/i2c.h>
 * OG unclerus --> please migrate your application code to adapt `driver/i2c_master.h`
 * #include driver/i2c/include/driver/i2c.h
*/

#include <driver/i2c.h>
#include <freertos/FreeRTOS.h>
#include <freertos/semphr.h>
#include <esp_err.h>
#include <esp_idf_lib_helpers.h>
```



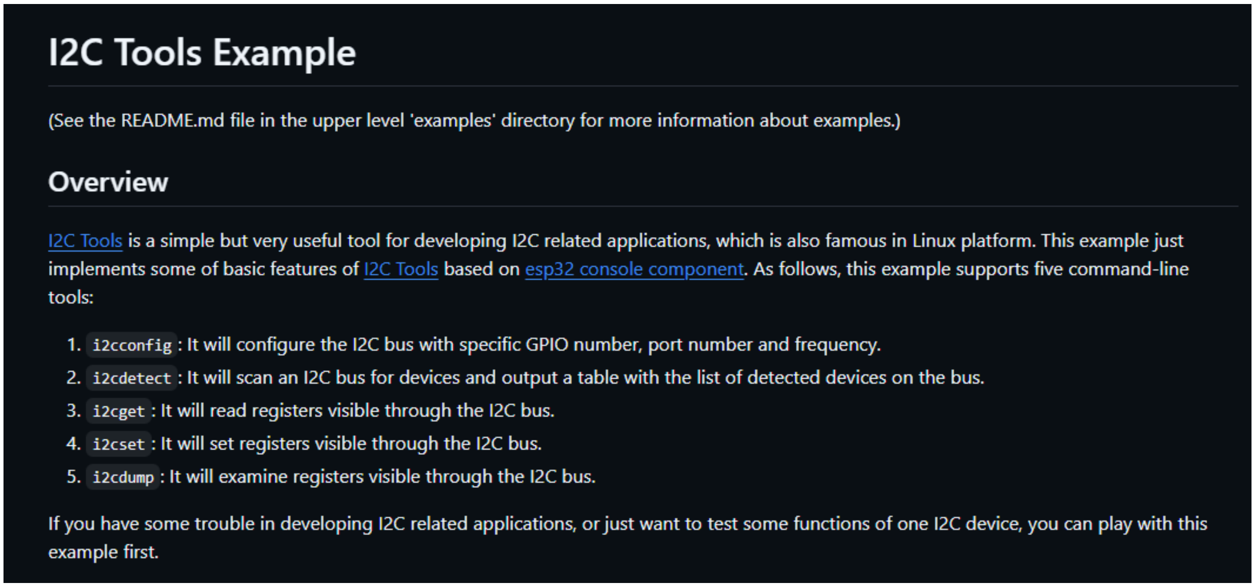
```cmake
idf_component_register(SRCS "main.c"
                    INCLUDE_DIRS "."
                    REQUIRES esp_idf_lib_helpers sht3x i2cdev freertos esp_common esp_system openthread driver)
```

Reverted back to i2c.h cuz thats whats in the actual idf component file not “master”

Then i included driver in the requires
So maybe its just looking for the driver file

—----------

I2C Tools Example



I2C Tools Example

(See the README.md file in the upper level 'examples' directory for more information about examples.)

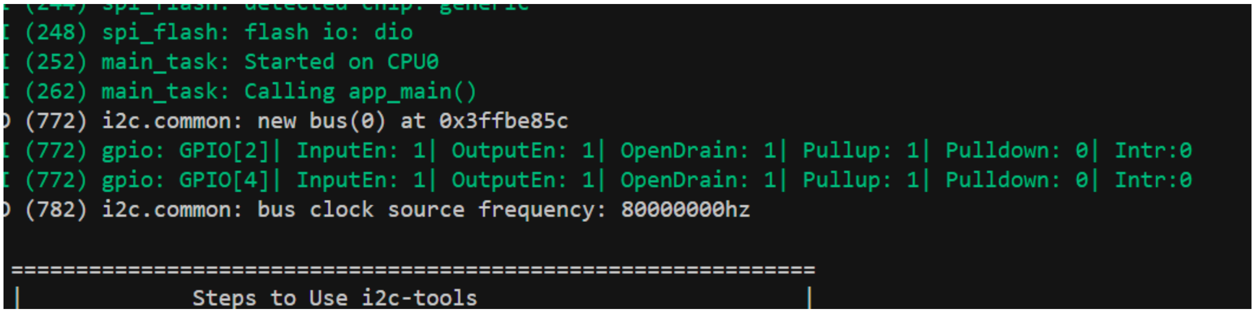
Overview

I2C Tools is a simple but very useful tool for developing I2C related applications, which is also famous in Linux platform. This example just implements some of basic features of I2C Tools based on esp32 console component. As follows, this example supports five command-line tools:

1. `i2cconfig`: It will configure the I2C bus with specific GPIO number, port number and frequency.
2. `i2cdetect`: It will scan an I2C bus for devices and output a table with the list of detected devices on the bus.
3. `i2cget`: It will read registers visible through the I2C bus.
4. `i2cset`: It will set registers visible through the I2C bus.
5. `i2cdump`: It will examine registers visible through the I2C bus.

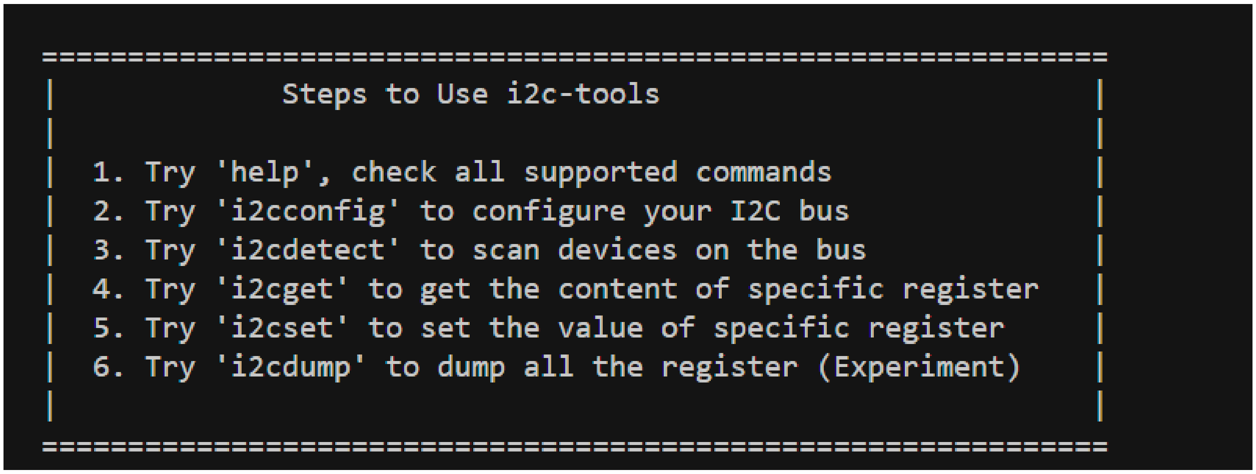
If you have some trouble in developing I2C related applications, or just want to test some functions of one I2C device, you can play with this example first.

Started another project: i2ctools:



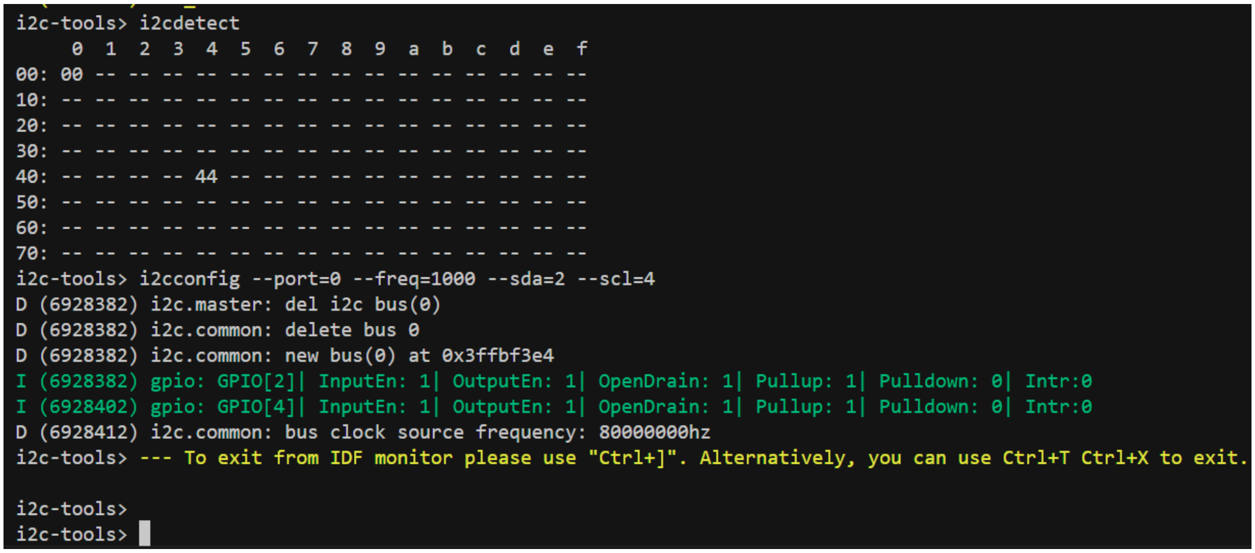
```
I (244) spi_flash: detected chip: generic
I (248) spi_flash: flash io: dio
I (252) main_task: Started on CPU0
I (262) main_task: Calling app_main()
D (772) i2c.common: new bus(0) at 0x3ffbe85c
I (772) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (772) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
D (782) i2c.common: bus clock source frequency: 80000000hz

 ==============================================================
 |             Steps to Use i2c-tools                         |
```

It worked and recognized the pullup resistor



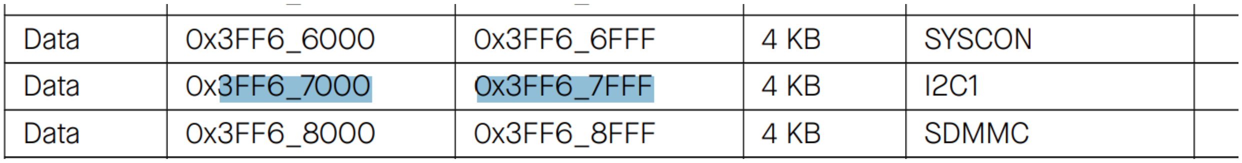
```
===================================================================
|                 Steps to Use i2c-tools                          |
|                                                                 |
|  1. Try 'help', check all supported commands                    |
|  2. Try 'i2cconfig' to configure your I2C bus                   |
|  3. Try 'i2cdetect' to scan devices on the bus                  |
|  4. Try 'i2cget' to get the content of specific register        |
|  5. Try 'i2cset' to set the value of specific register          |
|  6. Try 'i2cdump' to dump all the register (Experiment)         |
|                                                                 |
===================================================================
```

These are basically the commands and what order i need to to them in. i also have to find out a way to set the mode.



```
i2c-tools> i2cdetect
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: 00 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- 44 -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
i2c-tools> i2cconfig --port=0 --freq=1000 --sda=2 --scl=4
D (6928382) i2c.master: del i2c bus(0)
D (6928382) i2c.common: delete bus 0
D (6928382) i2c.common: new bus(0) at 0x3ffbf3e4
I (6928382) gpio: GPIO[2]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (6928402) gpio: GPIO[4]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
D (6928412) i2c.common: bus clock source frequency: 80000000hz
i2c-tools> --- To exit from IDF monitor please use "Ctrl+]". Alternatively, you can use Ctrl+T Ctrl+X to exit.

i2c-tools>
i2c-tools> 
```

```
idf term output:
              3ffbf3e4     dec 1073476580
i2C0 highend: 3FF53FFF            1073037311
i2C1 low | 3FF67000            1073115136  high  0x3FF67FFF 1073119231
```



Data	0x3FF6_6000	0x3FF6_6FFF	4 KB	SYSCON	
Data	0x3FF6_7000	0x3FF6_7FFF	4 KB	I2C1	
Data	0x3FF6_8000	0x3FF6_8FFF	4 KB	SDMMC	

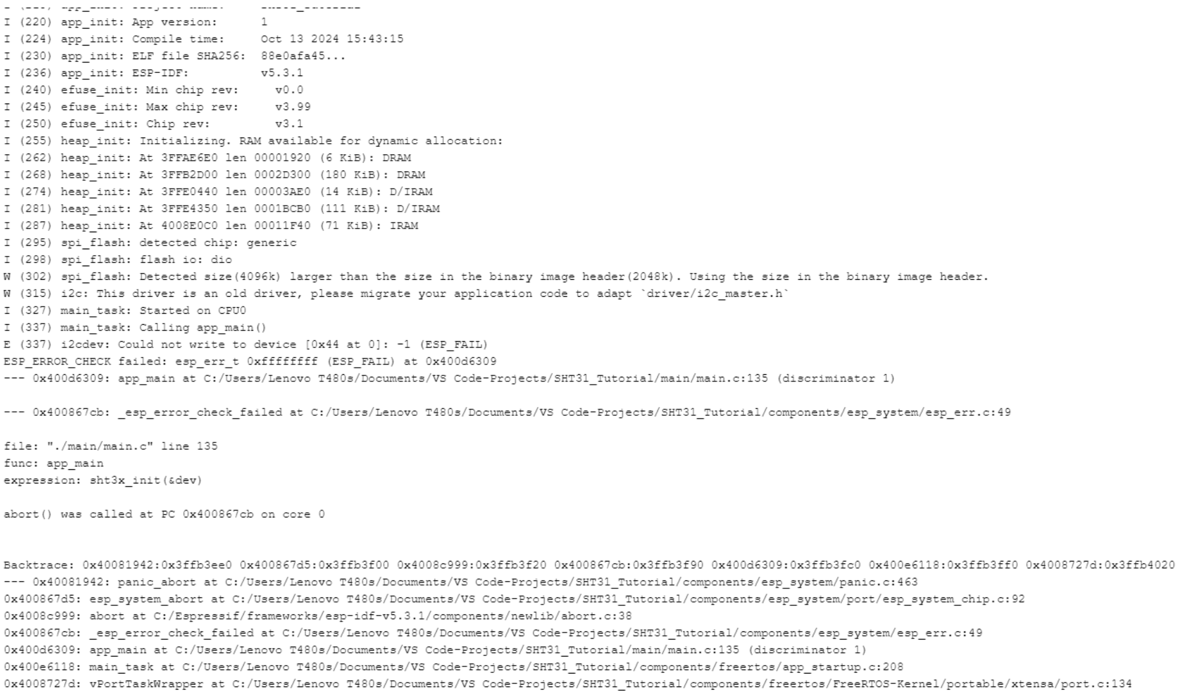
In I2C1..

Weird cuz we selected 0

10/21/2024

Next steps:

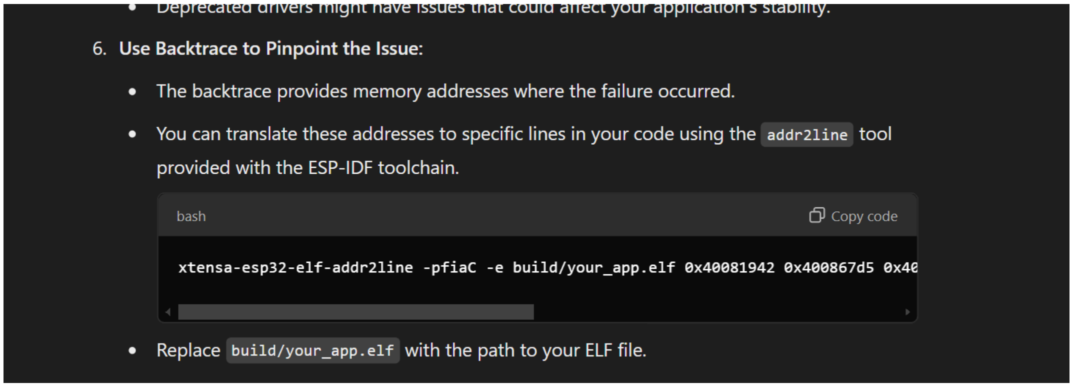
- ☑ ~~Read section 1 *System and Memory* from esp32_technical_reference_manual_en.pdf~~
- ☐ Review Espressif chips ROM Bootloader Utility: esptool.py
  - ☑ ~~Create a cmd line cheat sheet for esp IDF and include idf.py and esptool.py commands~~
  - ☑ ~~Include the common to index registers~~
- ☑ ~~Document the `panic_abort` list and index each address~~
  - ☑ ~~Review where these addresses are located~~
- ☑ ~~Review data logs file locations and think of a streamlined system to preview errors and to document them~~
- ☑ ~~Review Partition Tables~~
  - ☐ Wiki
  - ☐ Espressif chip partition tables



```
I (220) app_init: App version:      1
I (224) app_init: Compile time:     Oct 13 2024 15:43:15
I (230) app_init: ELF file SHA256:  88e0afa45...
I (236) app_init: ESP-IDF:          v5.3.1
I (240) efuse_init: Min chip rev:     v0.0
I (245) efuse_init: Max chip rev:     v3.99
I (250) efuse_init: Chip rev:         v3.1
I (255) heap_init: Initializing. RAM available for dynamic allocation:
I (262) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (268) heap_init: At 3FFB2D00 len 0002D300 (180 KiB): DRAM
I (274) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (281) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (287) heap_init: At 4008E0C0 len 00011F40 (71 KiB): IRAM
I (295) spi_flash: detected chip: generic
I (298) spi_flash: flash io: dio
W (302) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
W (315) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (327) main_task: Started on CPU0
I (337) main_task: Calling app_main()
E (337) i2cdev: Could not write to device [0x44 at 0]: -1 (ESP_FAIL)
ESP_ERROR_CHECK failed: esp_err_t 0xffffffff (ESP_FAIL) at 0x400d6309
--- 0x400d6309: app_main at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/main/main.c:135 (discriminator 1)

--- 0x400867cb: _esp_error_check_failed at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/esp_system/esp_err.c:49

file: "./main/main.c" line 135
func: app_main
expression: sht3x_init(&dev)

abort() was called at PC 0x400867cb on core 0


Backtrace: 0x40081942:0x3ffb3ee0 0x400867d5:0x3ffb3f00 0x4008c999:0x3ffb3f20 0x400867cb:0x3ffb3f90 0x400d6309:0x3ffb3fc0 0x400e6118:0x3ffb3ff0 0x4008727d:0x3ffb4020
--- 0x40081942: panic_abort at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/esp_system/panic.c:463
0x400867d5: esp_system_abort at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/esp_system/port/esp_system_chip.c:92
0x4008c999: abort at C:/Espressif/frameworks/esp-idf-v5.3.1/components/newlib/abort.c:38
0x400867cb: _esp_error_check_failed at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/esp_system/esp_err.c:49
0x400d6309: app_main at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/main/main.c:135 (discriminator 1)
0x400e6118: main_task at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/freertos/app_startup.c:208
0x4008727d: vPortTaskWrapper at C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/components/freertos/FreeRTOS-Kernel/portable/xtensa/port.c:134
```

Backtrace esp error

10/19/2024



- Deprecated drivers might have issues that could affect your application's stability.

6. **Use Backtrace to Pinpoint the Issue:**
   - The backtrace provides memory addresses where the failure occurred.
   - You can translate these addresses to specific lines in your code using the `addr2line` tool provided with the ESP-IDF toolchain.

```bash
xtensa-esp32-elf-addr2line -pfiaC -e build/your_app.elf 0x40081942 0x400867d5 0x40
```

   - Replace `build/your_app.elf` with the path to your ELF file.

xtensa-esp32-elf-addr2line -pfiaC -e build/SHT31_Tutorial.elf 0x40081942 0x400867d5 0x4008cad5 0x40087082 0x400d6837 0x400d63ad 0x400d655b 0x400d62f1 0x400e616c 0x4008727d

—-----------

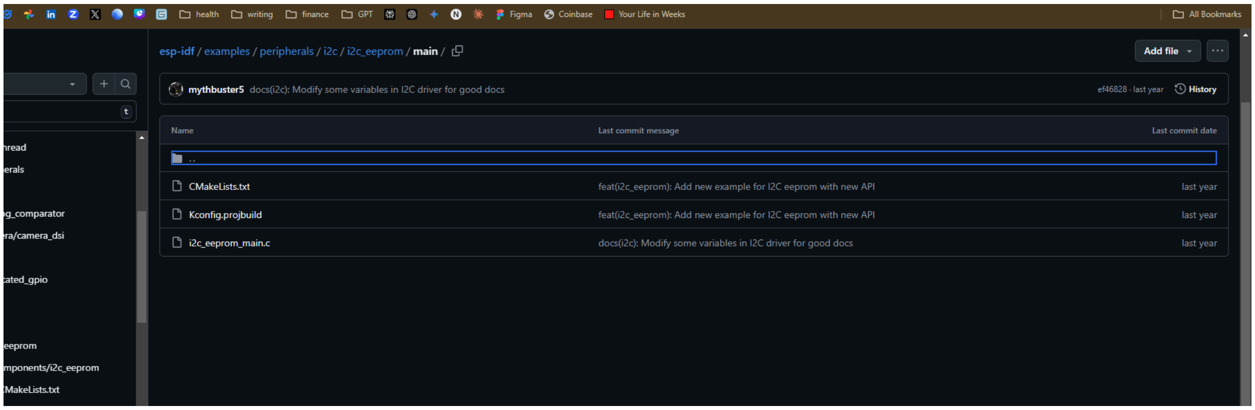
C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial>xtensa-esp32-elf-addr2line -pfiaC -e build/SHT31_Tutorial.elf

```
0x40081942 0x400867d5 0x4008cad5 0x40087082 0x400d6837 0x400d63ad 0x400d655b
0x400d62f1 0x400e616c 0x4008727d
0x40081942: panic_abort at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/components/esp_system/panic.c:463
0x400867d5: esp_system_abort at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/components/esp_system/port/esp_system_chip.c:92
0x4008cad5: __assert_func at
C:/Espressif/frameworks/esp-idf-v5.3.1/components/newlib/assert.c:80
0x40087082: xQueueSemaphoreTake at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/components/freertos/FreeRTOS-Kernel/queue.c:1709
(discriminator 1)
0x400d6837: i2c_dev_take_mutex at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/components/i2cdev/i2cdev.c:151
0x400d63ad: send_cmd at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/components/sht3x/sht3x.c:131
0x400d655b: sht3x_init at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/components/sht3x/sht3x.c:237
0x400d62f1: app_main at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/main/main.c:137
0x400e616c: main_task at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/components/freertos/app_startup.c:208
0x4008727d: vPortTaskWrapper at C:/Users/Lenovo T480s/Documents/VS
Code-Projects/SHT31_Tutorial/components/freertos/FreeRTOS-Kernel/portable/xtensa/port.c:13
4
```

"px" is a naming convention commonly used in C programming (which is often used for embedded systems like ESP32). It's a prefix that suggests the variable is a pointer.

- **Pointers:** In C, a pointer is a variable that holds the memory address of another variable. Think of it like a street address that tells you where a house is located.

So, `pxQueue` likely means "pointer to a queue".



esp-idf / examples / peripherals / i2c / i2c_eeprom / main /

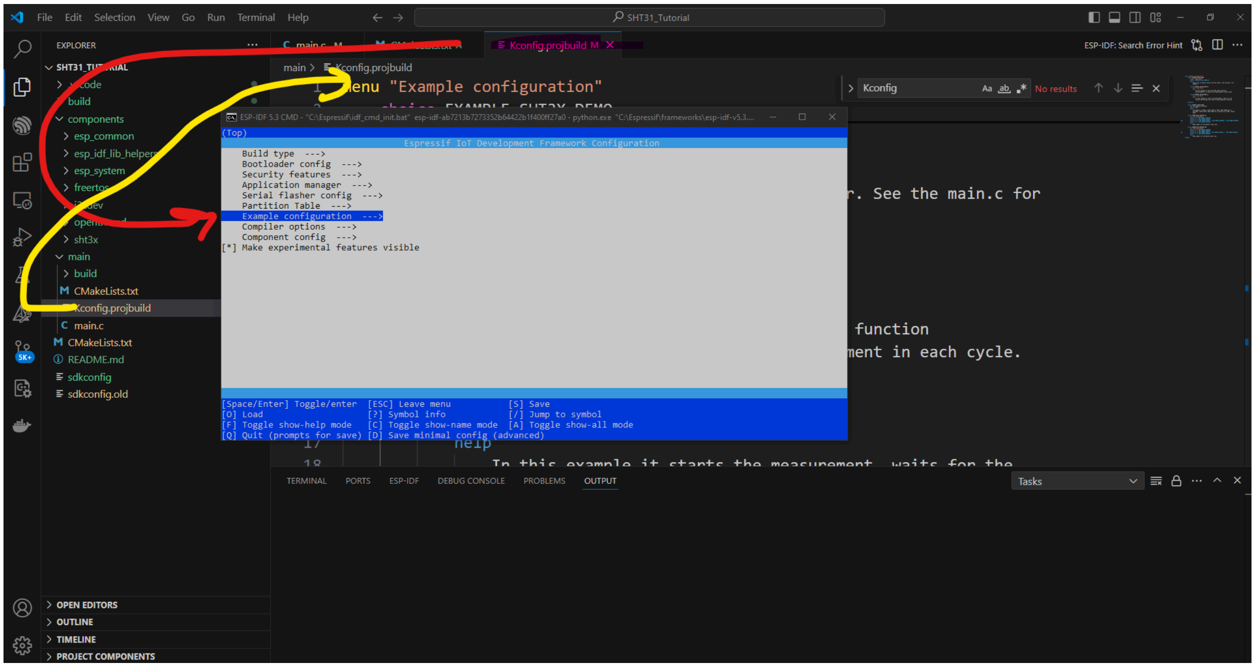
mythbuster5 docs(i2c): Modify some variables in I2C driver for good docs — ef46828 · last year

Name	Last commit message	Last commit date
..		
CMakeLists.txt	feat(i2c_eeprom): Add new example for I2C eeprom with new API	last year
Kconfig.projbuild	feat(i2c_eeprom): Add new example for I2C eeprom with new API	last year
i2c_eeprom_main.c	docs(i2c): Modify some variables in I2C driver for good docs	last year

If SHT isnt working… EEPROM

Try:

```
CONFIG_EXAMPLE_SHT3X_DEMO_HL
CONFIG_EXAMPLE_SHT3X_DEMO_LL
```



main > Kconfig.projbuild

```
menu "Example configuration"
```

ESP-IDF 5.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-ab7213b7273352b64422b1f400ff27a0 - python.exe "C:\Espressif\frameworks\esp-idf-v5.3....

```
(Top)
                    Espressif IoT Development Framework Configuration
    Build type  --->
    Bootloader config  --->
    Security features  --->
    Application manager  --->
    Serial flasher config  --->
    Partition Table  --->
    Example configuration  --->
    Compiler options  --->
    Component config  --->
[*] Make experimental features visible

[Space/Enter] Toggle/enter   [ESC] Leave menu          [S] Save
[O] Load                     [?] Symbol info           [/] Jump to symbol
[F] Toggle show-help mode    [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save)  [D] Save minimal config (advanced)
```

10/17/2024

Made v002 firmware procedure
Found idf sht tutorial on youtube. Still getting no errors and getting these warnings:

```
assert failed: xQueueSemaphoreTake queue.c:1709 (( pxQueue ))
```

10/15/2024

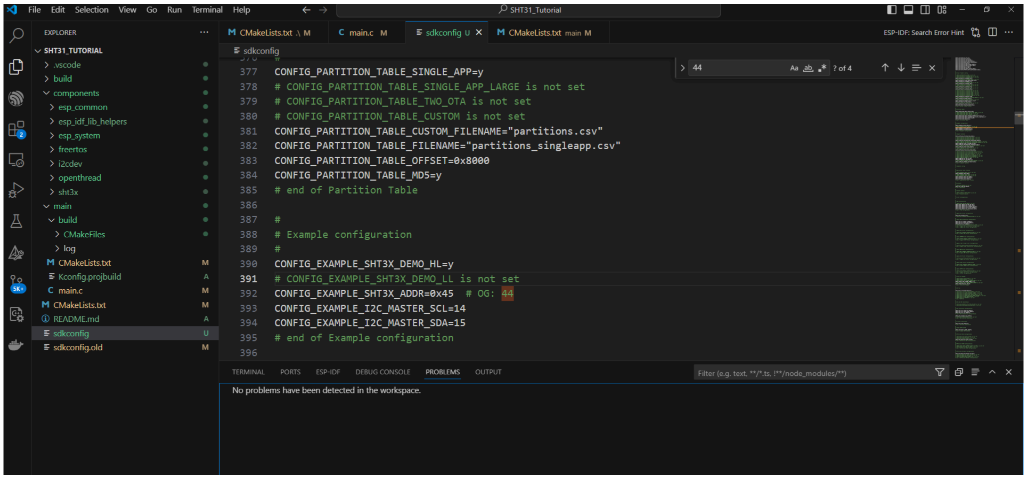
From build log:

```
ESP_ERROR_CHECK failed: esp_err_t 0xffffffff (ESP_FAIL) at 0x400d6309
```

From cmd line output:
ESP_ERROR_CHECK failed: esp_err_t 0xffffffff (ESP_FAIL) at 0x400d631e

Hex:
0x400d631e - `0x400d6309 =`

So, the difference between `0x400d631e` and `0x400d6309` is **21** in decimal, or **0x15** in hex.



```
CONFIG_PARTITION_TABLE_SINGLE_APP=y
# CONFIG_PARTITION_TABLE_SINGLE_APP_LARGE is not set
# CONFIG_PARTITION_TABLE_TWO_OTA is not set
# CONFIG_PARTITION_TABLE_CUSTOM is not set
CONFIG_PARTITION_TABLE_CUSTOM_FILENAME="partitions.csv"
CONFIG_PARTITION_TABLE_FILENAME="partitions_singleapp.csv"
CONFIG_PARTITION_TABLE_OFFSET=0x8000
CONFIG_PARTITION_TABLE_MD5=y
# end of Partition Table

#
# Example configuration
#
CONFIG_EXAMPLE_SHT3X_DEMO_HL=y
# CONFIG_EXAMPLE_SHT3X_DEMO_LL is not set
CONFIG_EXAMPLE_SHT3X_ADDR=0x45  # OG: 44
CONFIG_EXAMPLE_I2C_MASTER_SCL=14
CONFIG_EXAMPLE_I2C_MASTER_SDA=15
# end of Example configuration

```

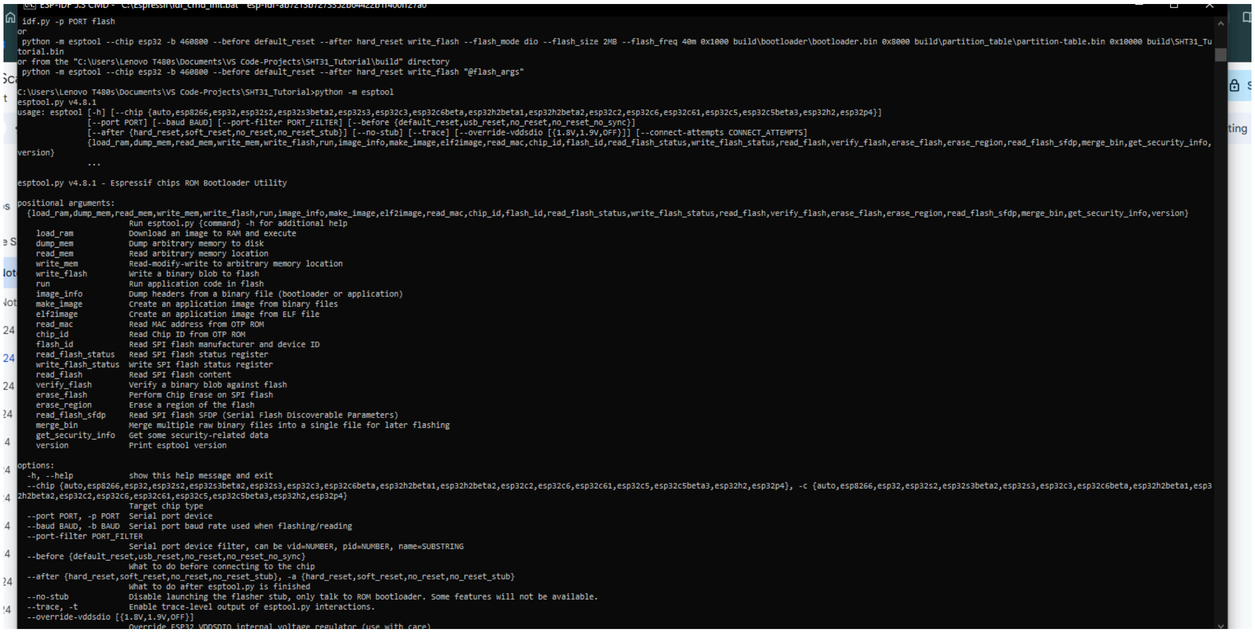
No problems have been detected in the workspace.

Failed at x45 too
in app_main/nE (337) i2cdev: Could not write to device [0x45 at 0]: -1 (ESP_FAIL)

```
in app_main/nE (337) i2cdev: Could not write to device [0x45 at 0]: -1 (ESP_FAIL)
ESP_ERROR_CHECK failed: esp_err_t 0xffffffff (ESP_FAIL) at 0x400d631e
file: "./main/main.c" line 137
func: app_main
```

expression: sht3x_init(&dev)

abort() was called at PC 0x400867cb on core 0



```
 idf.py -p PORT flash
or
 python -m esptool --chip esp32 -b 460800 --before default_reset --after hard_reset write_flash --flash_mode dio --flash_size 2MB --flash_freq 40m 0x1000 build\bootloader\bootloader.bin 0x8000 build\partition_table\partition-table.bin 0x10000 build\SHT31_Tu
torial.bin
or from the "C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial\build" directory
 python -m esptool --chip esp32 -b 460800 --before default_reset --after hard_reset write_flash "@flash_args"

C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial>python -m esptool
esptool.py v4.8.1
usage: esptool [-h] [--chip {auto,esp8266,esp32,esp32s2,esp32s3beta2,esp32s3,esp32c3,esp32c6beta,esp32h2beta1,esp32h2beta2,esp32c2,esp32c6,esp32c61,esp32c5,esp32c5beta3,esp32h2,esp32p4}]
               [--port PORT] [--baud BAUD] [--port-filter PORT_FILTER] [--before {default_reset,usb_reset,no_reset,no_reset_no_sync}]
               [--after {hard_reset,soft_reset,no_reset,no_reset_stub}] [--no-stub] [--trace] [--override-vddsdio [{1.8V,1.9V,OFF}]] [--connect-attempts CONNECT_ATTEMPTS]
               {load_ram,dump_mem,read_mem,write_mem,write_flash,run,image_info,make_image,elf2image,read_mac,chip_id,flash_id,read_flash_status,write_flash_status,read_flash,verify_flash,erase_flash,erase_region,read_flash_sfdp,merge_bin,get_security_info,
version}
               ...

esptool.py v4.8.1 - Espressif chips ROM Bootloader Utility

positional arguments:
  {load_ram,dump_mem,read_mem,write_mem,write_flash,run,image_info,make_image,elf2image,read_mac,chip_id,flash_id,read_flash_status,write_flash_status,read_flash,verify_flash,erase_flash,erase_region,read_flash_sfdp,merge_bin,get_security_info,version}
                        Run esptool.py {command} -h for additional help
    load_ram            Download an image to RAM and execute
    dump_mem            Dump arbitrary memory to disk
    read_mem            Read arbitrary memory location
    write_mem           Read-modify-write to arbitrary memory location
    write_flash         Write a binary blob to flash
    run                 Run application code in flash
    image_info          Dump headers from a binary file (bootloader or application)
    make_image          Create an application image from binary files
    elf2image           Create an application image from ELF file
    read_mac            Read MAC address from OTP ROM
    chip_id             Read Chip ID from OTP ROM
    flash_id            Read SPI flash manufacturer and device ID
    read_flash_status   Read SPI flash status register
    write_flash_status  Write SPI flash status register
    read_flash          Read SPI flash content
    verify_flash        Verify a binary blob against flash
    erase_flash         Perform Chip Erase on SPI flash
    erase_region        Erase a region of the flash
    read_flash_sfdp     Read SPI flash SFDP (Serial Flash Discoverable Parameters)
    merge_bin           Merge multiple raw binary files into a single file for later flashing
    get_security_info   Get some security-related data
    version             Print esptool version

options:
  -h, --help            show this help message and exit
  --chip {auto,esp8266,esp32,esp32s2,esp32s3beta2,esp32s3,esp32c3,esp32c6beta,esp32h2beta1,esp32h2beta2,esp32c2,esp32c6,esp32c61,esp32c5,esp32c5beta3,esp32h2,esp32p4}, -c {auto,esp8266,esp32,esp32s2,esp32s3beta2,esp32s3,esp32c3,esp32c6beta,esp32h2beta1,esp3
2h2beta2,esp32c2,esp32c6,esp32c61,esp32c5,esp32c5beta3,esp32h2,esp32p4}
                        Target chip type
  --port PORT, -p PORT  Serial port device
  --baud BAUD, -b BAUD  Serial port baud rate used when flashing/reading
  --port-filter PORT_FILTER
                        Serial port device filter, can be vid=NUMBER, pid=NUMBER, name=SUBSTRING
  --before {default_reset,usb_reset,no_reset,no_reset_no_sync}
                        What to do before connecting to the chip
  --after {hard_reset,soft_reset,no_reset,no_reset_stub}, -a {hard_reset,soft_reset,no_reset,no_reset_stub}
                        What to do after esptool.py is finished
  --no-stub             Disable launching the flasher stub, only talk to ROM bootloader. Some features will not be available.
  --trace, -t           Enable trace-level output of esptool.py interactions.
  --override-vddsdio [{1.8V,1.9V,OFF}]
                        Override ESP32 VDDSDIO internal voltage regulator (use with care)
```

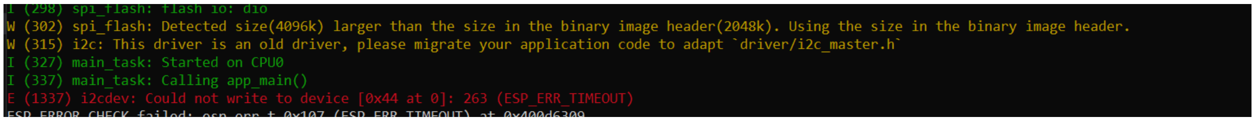
python -m esptool **for debugging

10/14/2024

ESP32 build procedure

cd C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial(*whatever is top level CMakeLists.txt)
idf.py clean
idf.py fullclean (to delete build folder)
Remove daisy chain Vcc
Reset 5v
Insert programming pin
idf.py menuconfig
idf.py build
idf.py -p COM16 flash

Remove programming pin
Insert daisy chain Vcc
idf.py -p COM16 monitor



```
I (298) spi_flash: flash io: dio
W (302) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
W (315) i2c: This driver is an old driver, please migrate your application code to adapt `driver/i2c_master.h`
I (327) main_task: Started on CPU0
I (337) main_task: Calling app_main()
E (1337) i2cdev: Could not write to device [0x44 at 0]: 263 (ESP_ERR_TIMEOUT)
ESP_ERROR_CHECK failed: esp_err_t 0x107 (ESP_ERR_TIMEOUT) at 0x400d6309
```

I2c driver?

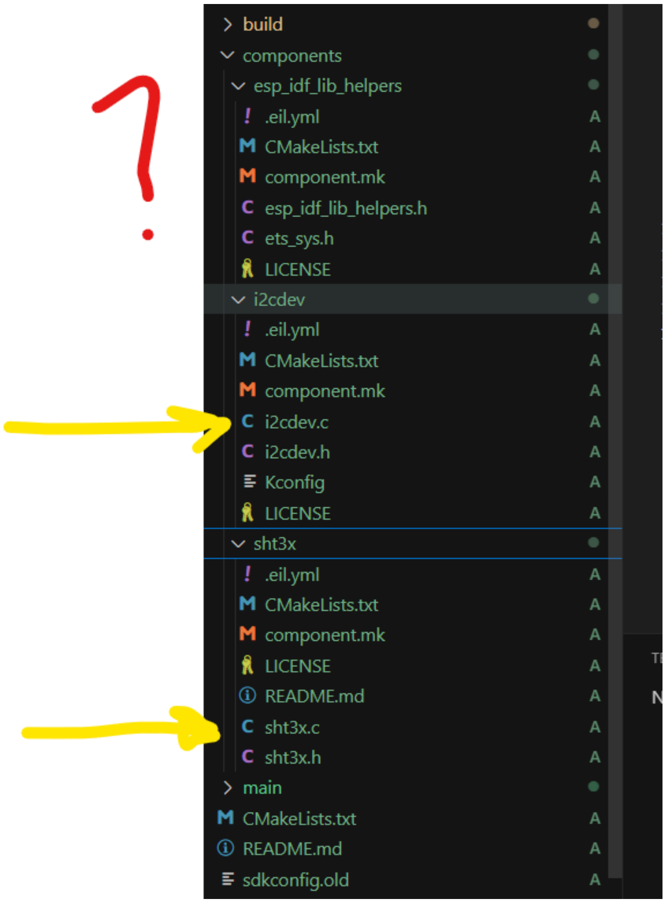
```
W (315) i2c: This driver is an old driver, please migrate your
application code to adapt driver/i2c_master.h
```

- This warning indicates that the current I2C driver is outdated, and you are advised to update your code to use the newer `i2c_master.h`. The warning itself doesn't cause a crash, but it's good practice to migrate to the newer driver to avoid potential issues in the future.

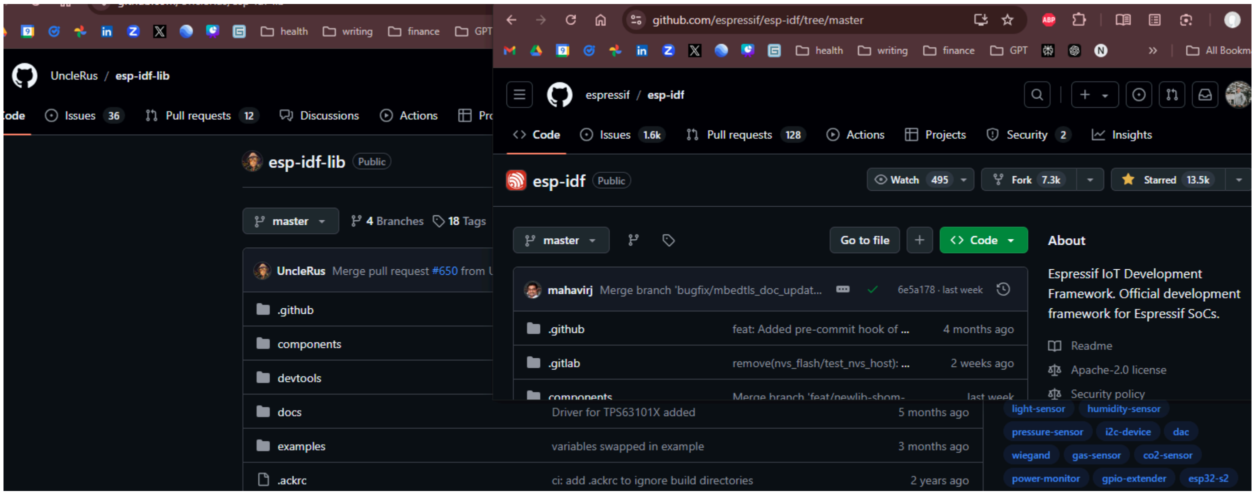
```
E (337) i2cdev: Could not write to device [0x44 at 0]: -1 (ESP_FAIL)
```

- This error is crucial. It indicates that the I2C communication to the device at address `0x44` has failed. The `ESP_FAIL` return code means that the ESP32 could not establish a successful I2C communication, which might be due to several reasons:
    - ☐ Incorrect wiring (SDA and SCL lines).
        - Double-check that the SDA and SCL lines are connected properly between the ESP32 and the SHT3x sensor.
        - Ensure that the pull-up resistors (usually 4.7kΩ) are connected on the I2C bus, unless the sensor breakout board has built-in resistors.
        - Ensure the sensor is powered correctly (check the voltage requirements—likely 3.3V or 5V).
    - ☐ Incorrect I2C address.
    - ☐ The sensor might not be powered correctly.
    - ☐ There could be an issue with the I2C peripheral settings (clock speed, etc.).

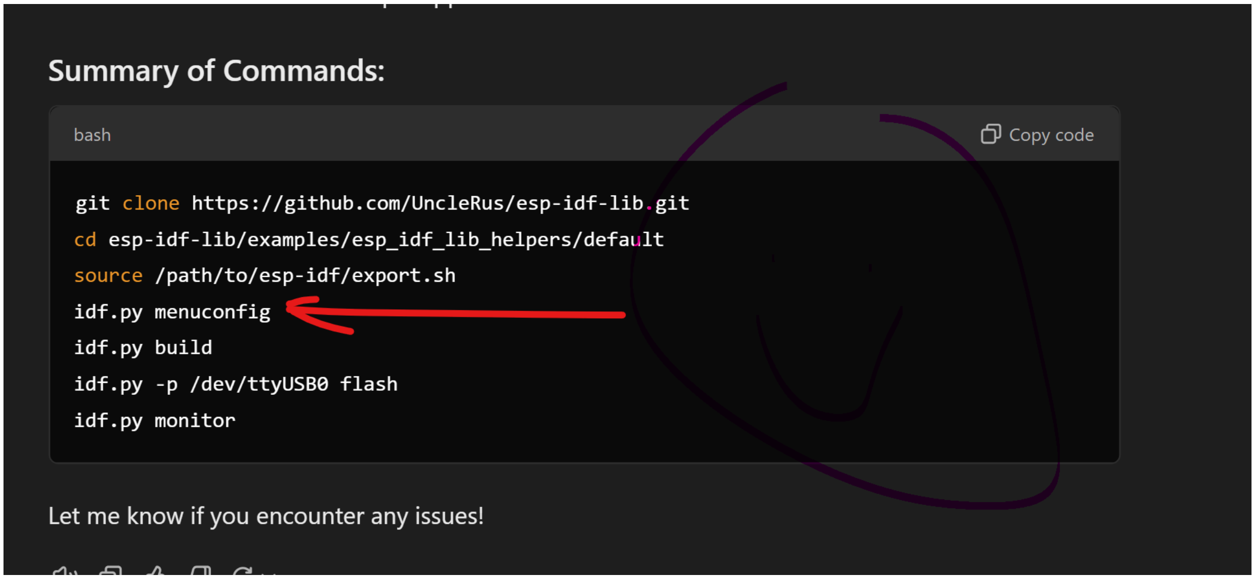
10/12/2024



No C code. For lib helpers
And the idf lib names dont line up


UncleRus / esp-idf-lib
Code
Issues 36
Pull requests 12
Discussions
Actions
esp-idf-lib Public
master
4 Branches
18 Tags
UncleRus Merge pull request #650 from
.github
components
devtools
docs
examples
.ackrc
Driver for TPS63101X added 5 months ago
variables swapped in example 3 months ago
ci: add .ackrc to ignore build directories 2 years ago
github.com/espressif/esp-idf/tree/master
espressif / esp-idf
Code
Issues 1.6k
Pull requests 128
Actions
Projects
Security 2
Insights
esp-idf Public
Watch 495
Fork 7.3k
Starred 13.5k
master
Go to file
Code
mahavirj Merge branch 'bugfix/mbedtls_doc_updat... 6e5a178 · last week
.github feat: Added pre-commit hook of ... 4 months ago
.gitlab remove(nvs_flash/test_nvs_host): ... 2 weeks ago
components Merge branch 'feat/newlib-shom- last week
About
Espressif IoT Development Framework. Official development framework for Espressif SoCs.
Readme
Apache-2.0 license
Security policy
light-sensor
humidity-sensor
pressure-sensor
i2c-device
dac
wiegand
gas-sensor
co2-sensor
power-monitor
gpio-extender
esp32-s2

Esp-idf-lib vs esp-idf

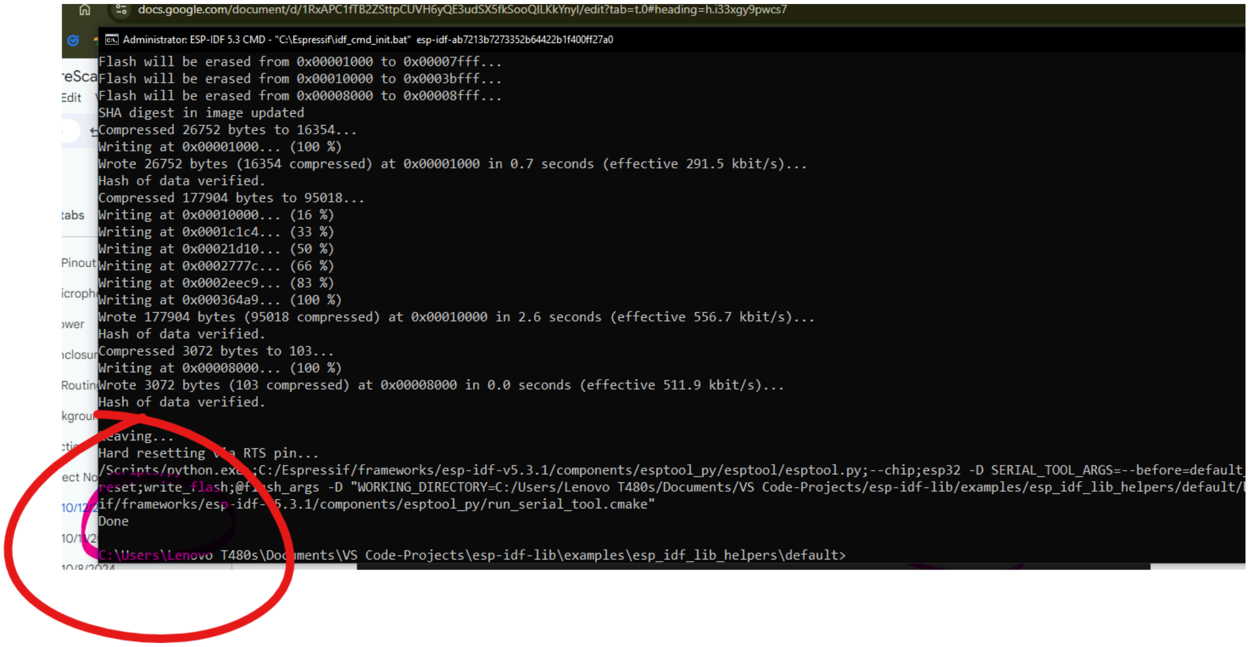


Summary of Commands:

```bash
git clone https://github.com/UncleRus/esp-idf-lib.git
cd esp-idf-lib/examples/esp_idf_lib_helpers/default
source /path/to/esp-idf/export.sh
idf.py menuconfig
idf.py build
idf.py -p /dev/ttyUSB0 flash
idf.py monitor
```

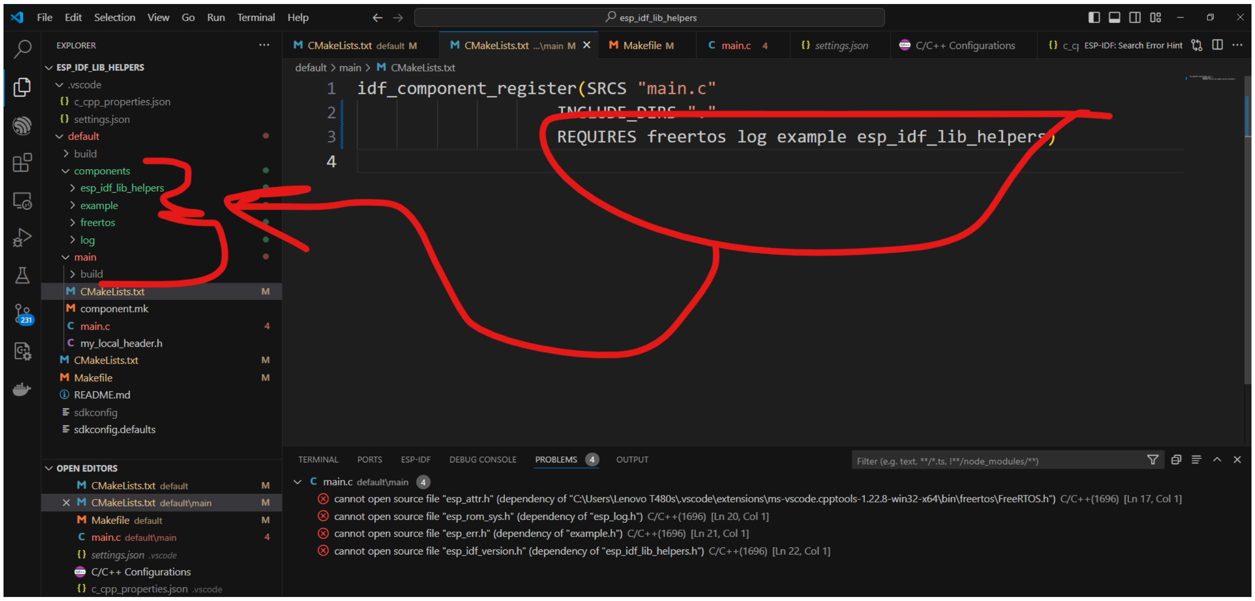
Let me know if you encounter any issues!

I needed to run this. I thought it was just for changing settings



docs.google.com/document/d/1RxAPC1fTB2ZSttpCUVH6yQE3udSX5fkSooQILKkYnyl/edit?tab=t.0#heading=h.i33xgy9pwcs7
Administrator: ESP-IDF 5.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-ab7213b7273352b64422b1f400ff27a0

```
Flash will be erased from 0x00001000 to 0x00007fff...
Flash will be erased from 0x00010000 to 0x0003bfff...
Flash will be erased from 0x00008000 to 0x00008fff...
SHA digest in image updated
Compressed 26752 bytes to 16354...
Writing at 0x00001000... (100 %)
Wrote 26752 bytes (16354 compressed) at 0x00001000 in 0.7 seconds (effective 291.5 kbit/s)...
Hash of data verified.
Compressed 177904 bytes to 95018...
Writing at 0x00010000... (16 %)
Writing at 0x0001c1c4... (33 %)
Writing at 0x00021d10... (50 %)
Writing at 0x0002777c... (66 %)
Writing at 0x0002eec9... (83 %)
Writing at 0x000364a9... (100 %)
Wrote 177904 bytes (95018 compressed) at 0x00010000 in 2.6 seconds (effective 556.7 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 103...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.0 seconds (effective 511.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
/Scripts/python.exe;C:/Espressif/frameworks/esp-idf-v5.3.1/components/esptool_py/esptool/esptool.py;--chip;esp32 -D SERIAL_TOOL_ARGS=--before=default_
reset;write_flash;@flash_args -D "WORKING_DIRECTORY=C:/Users/Lenovo T480s/Documents/VS Code-Projects/esp-idf-lib/examples/esp_idf_lib_helpers/default/
if/frameworks/esp-idf-v5.3.1/components/esptool_py/run_serial_tool.cmake"
Done

C:\Users\Lenovo T480s\Documents\VS Code-Projects\esp-idf-lib\examples\esp_idf_lib_helpers\default>
```

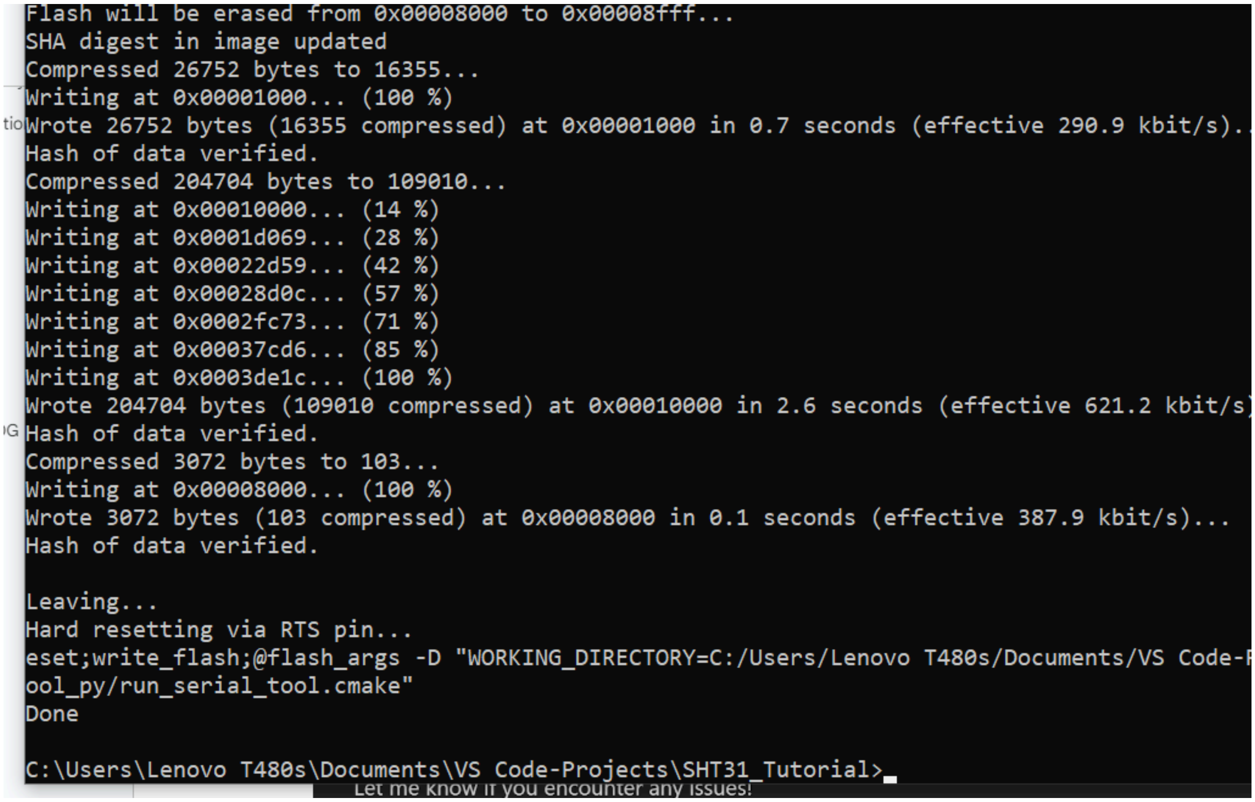


```
idf_component_register(SRCS "main.c"
                    INCLUDE_DIRS "."
                    REQUIRES freertos log example esp_idf_lib_helpers)
```

```
main.c default\main 4
cannot open source file "esp_attr.h" (dependency of "C:\Users\Lenovo T480s\.vscode\extensions\ms-vscode.cpptools-1.22.8-win32-x64\bin\freertos\FreeRTOS.h") C/C++(1696) [Ln 17, Col 1]
cannot open source file "esp_rom_sys.h" (dependency of "esp_log.h") C/C++(1696) [Ln 20, Col 1]
cannot open source file "esp_err.h" (dependency of "example.h") C/C++(1696) [Ln 21, Col 1]
cannot open source file "esp_idf_version.h" (dependency of "esp_idf_lib_helpers.h") C/C++(1696) [Ln 22, Col 1]
```

The include DIRC
Needs to be the same as the components
I think thats the secret

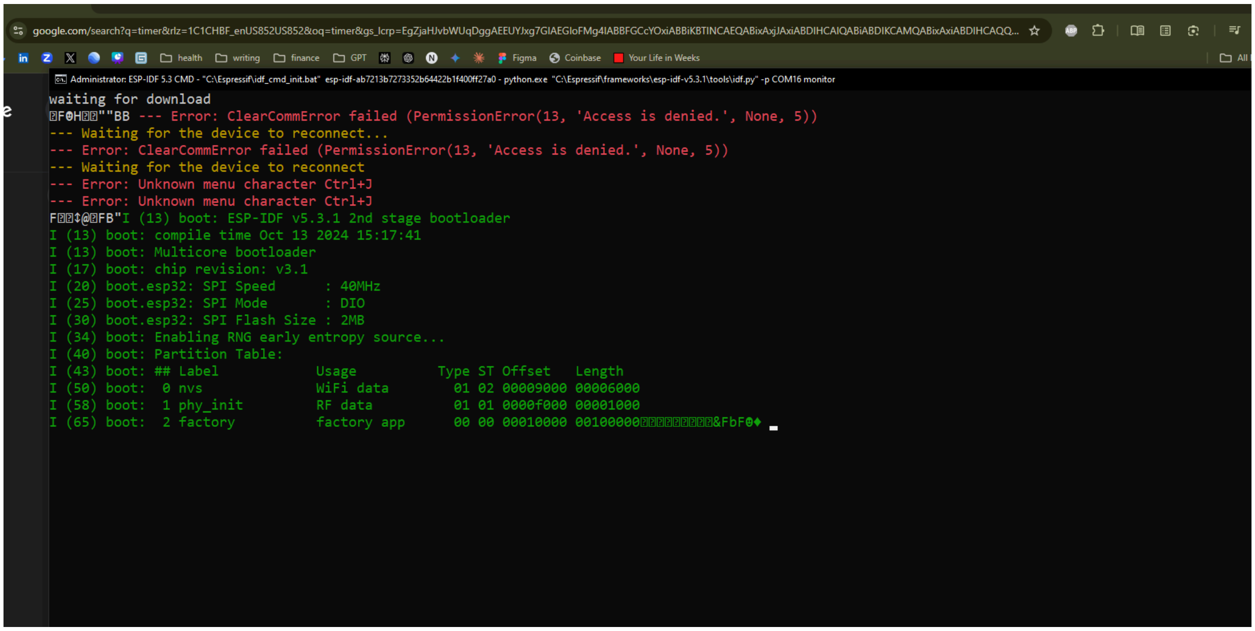
String



```
Flash will be erased from 0x00008000 to 0x00008fff...
SHA digest in image updated
Compressed 26752 bytes to 16355...
Writing at 0x00001000... (100 %)
Wrote 26752 bytes (16355 compressed) at 0x00001000 in 0.7 seconds (effective 290.9 kbit/s).
Hash of data verified.
Compressed 204704 bytes to 109010...
Writing at 0x00010000... (14 %)
Writing at 0x0001d069... (28 %)
Writing at 0x00022d59... (42 %)
Writing at 0x00028d0c... (57 %)
Writing at 0x0002fc73... (71 %)
Writing at 0x00037cd6... (85 %)
Writing at 0x0003de1c... (100 %)
Wrote 204704 bytes (109010 compressed) at 0x00010000 in 2.6 seconds (effective 621.2 kbit/s
Hash of data verified.
Compressed 3072 bytes to 103...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.1 seconds (effective 387.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
eset;write_flash;@flash_args -D "WORKING_DIRECTORY=C:/Users/Lenovo T480s/Documents/VS Code-
ool_py/run_serial_tool.cmake"
Done

C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial>
```

Let me know if you encounter any issues!

Flashed



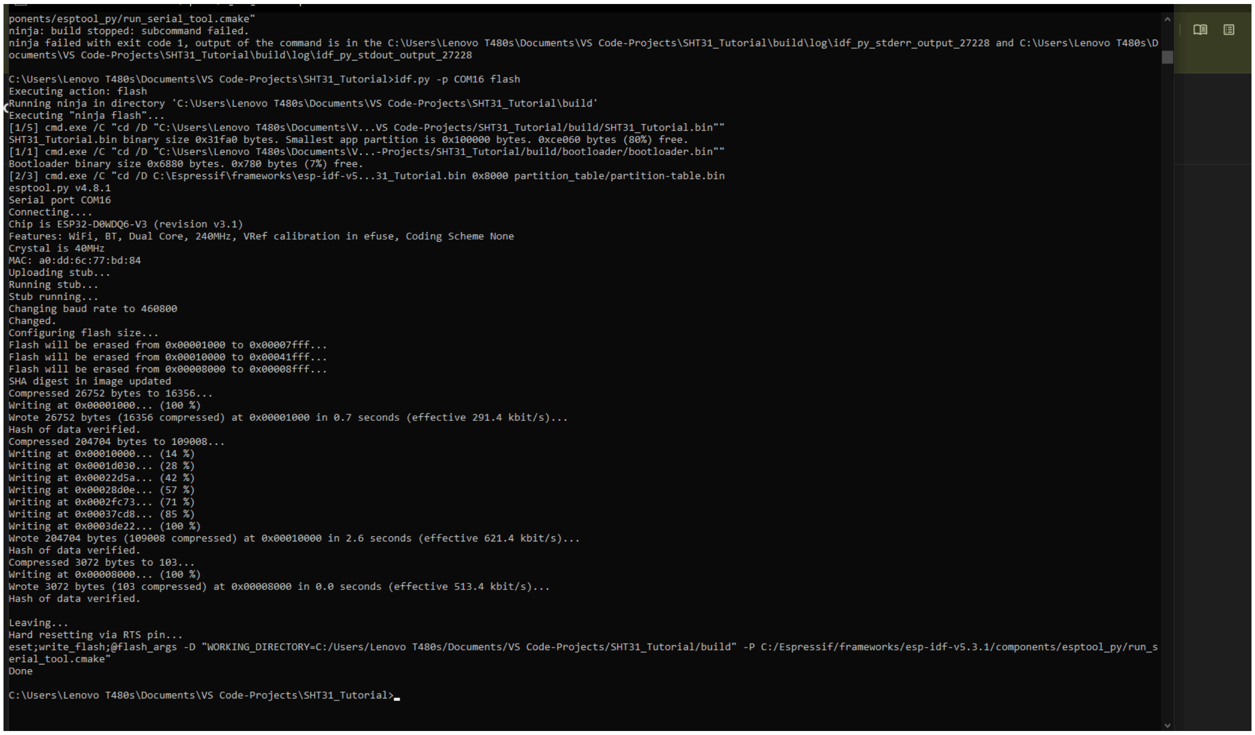
google.com/search?q=timer&rlz=1C1CHBF_enUS852US852&oq=timer&gs_lcrp=EgZjaHJvbWUqDggAEEUYJxg7GIAEGIoFMg4IABBFGCcYOxiABBiKBTINCAEQABixAxjJAxiABDIHCAIQABiABDIKCAMQABixAxiABDIHCAQQ...
health
writing
finance
GPT
Figma
Coinbase
Your Life in Weeks

Administrator: ESP-IDF 5.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-ab7213b7273352b64422b1f400ff27a0 - python.exe "C:\Espressif\frameworks\esp-idf-v5.3.1\tools\idf.py" -p COM16 monitor

```
waiting for download
▯F⊕H▯▯""BB --- Error: ClearCommError failed (PermissionError(13, 'Access is denied.', None, 5))
--- Waiting for the device to reconnect...
--- Error: ClearCommError failed (PermissionError(13, 'Access is denied.', None, 5))
--- Waiting for the device to reconnect
--- Error: Unknown menu character Ctrl+J
--- Error: Unknown menu character Ctrl+J
F▯▯↕@▯FB"I (13) boot: ESP-IDF v5.3.1 2nd stage bootloader
I (13) boot: compile time Oct 13 2024 15:17:41
I (13) boot: Multicore bootloader
I (17) boot: chip revision: v3.1
I (20) boot.esp32: SPI Speed      : 40MHz
I (25) boot.esp32: SPI Mode       : DIO
I (30) boot.esp32: SPI Flash Size : 2MB
I (34) boot: Enabling RNG early entropy source...
I (40) boot: Partition Table:
I (43) boot: ## Label            Usage          Type ST Offset   Length
I (50) boot:  0 nvs              WiFi data        01 02 00009000 00006000
I (58) boot:  1 phy_init         RF data          01 01 0000f000 00001000
I (65) boot:  2 factory          factory app      00 00 00010000 00100000▯▯▯▯▯▯▯▯▯▯&FbF⊕♦
```

Getting there

I unplugged the peripheral SHT data and clock and power
I reset the board by unplug plug of 5v
Then reflashed then it worked

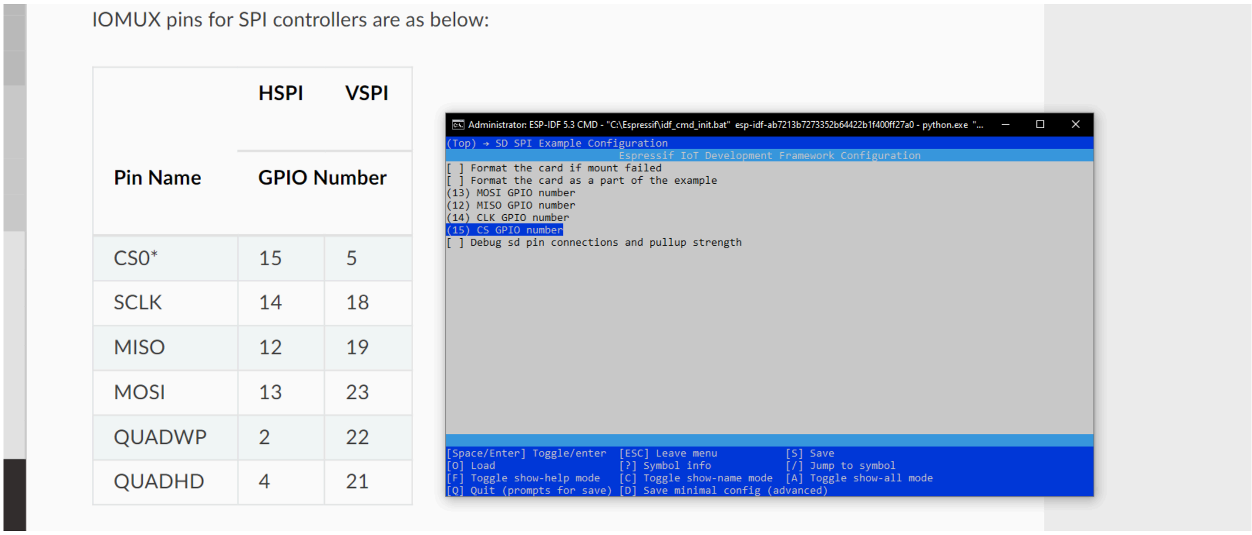


```
ponents/esptool_py/run_serial_tool.cmake"
ninja: build stopped: subcommand failed.
ninja failed with exit code 1, output of the command is in the C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial\build\log\idf_py_stderr_output_27228 and C:\Users\Lenovo T480s\D
ocuments\VS Code-Projects\SHT31_Tutorial\build\log\idf_py_stdout_output_27228

C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial>idf.py -p COM16 flash
Executing action: flash
Running ninja in directory 'C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial\build'
Executing "ninja flash"...
[1/5] cmd.exe /C "cd /D "C:\Users\Lenovo T480s\Documents\V...VS Code-Projects/SHT31_Tutorial/build/SHT31_Tutorial.bin""
SHT31_Tutorial.bin binary size 0x31fa0 bytes. Smallest app partition is 0x100000 bytes. 0xce060 bytes (80%) free.
[1/1] cmd.exe /C "cd /D "C:\Users\Lenovo T480s\Documents\V...-Projects/SHT31_Tutorial/build/bootloader/bootloader.bin""
Bootloader binary size 0x6880 bytes. 0x780 bytes (7%) free.
[2/3] cmd.exe /C "cd /D C:\Espressif\frameworks\esp-idf-v5...31_Tutorial.bin 0x8000 partition_table/partition-table.bin
esptool.py v4.8.1
Serial port COM16
Connecting....
Chip is ESP32-D0WDQ6-V3 (revision v3.1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: a0:dd:6c:77:bd:84
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Flash will be erased from 0x00001000 to 0x00007fff...
Flash will be erased from 0x00010000 to 0x00041fff...
Flash will be erased from 0x00008000 to 0x00008fff...
SHA digest in image updated
Compressed 26752 bytes to 16356...
Writing at 0x00001000... (100 %)
Wrote 26752 bytes (16356 compressed) at 0x00001000 in 0.7 seconds (effective 291.4 kbit/s)...
Hash of data verified.
Compressed 204704 bytes to 109008...
Writing at 0x00010000... (14 %)
Writing at 0x0001d030... (28 %)
Writing at 0x00022d5a... (42 %)
Writing at 0x00028d0e... (57 %)
Writing at 0x0002fc73... (71 %)
Writing at 0x00037cd8... (85 %)
Writing at 0x0003de22... (100 %)
Wrote 204704 bytes (109008 compressed) at 0x00010000 in 2.6 seconds (effective 621.4 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 103...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.0 seconds (effective 513.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
eset;write_flash;@flash_args -D "WORKING_DIRECTORY=C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/build" -P C:/Espressif/frameworks/esp-idf-v5.3.1/components/esptool_py/run_s
erial_tool.cmake"
Done

C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial>
```

```
#define I2C_SDA 14
#define I2C_SCL 15
```

I removed the “3.3 volt pin” for the daisy chained power source and were running the program

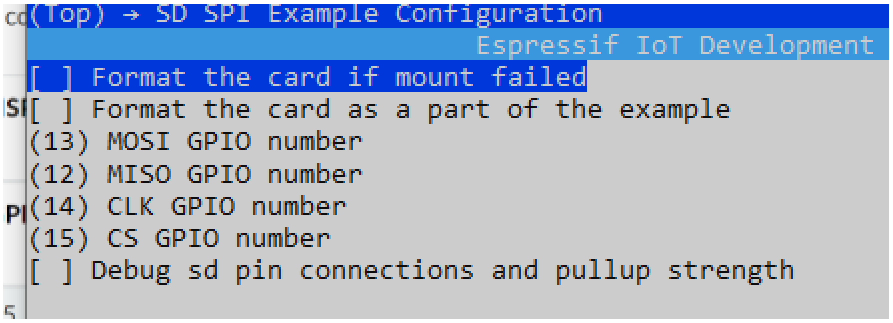
10/11/2024


Administrator: ESP-IDF 5.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-ab7213b7273352b64422b1f400ff27a0 - python.exe "...

IOMUX pins for SPI controllers are as below:

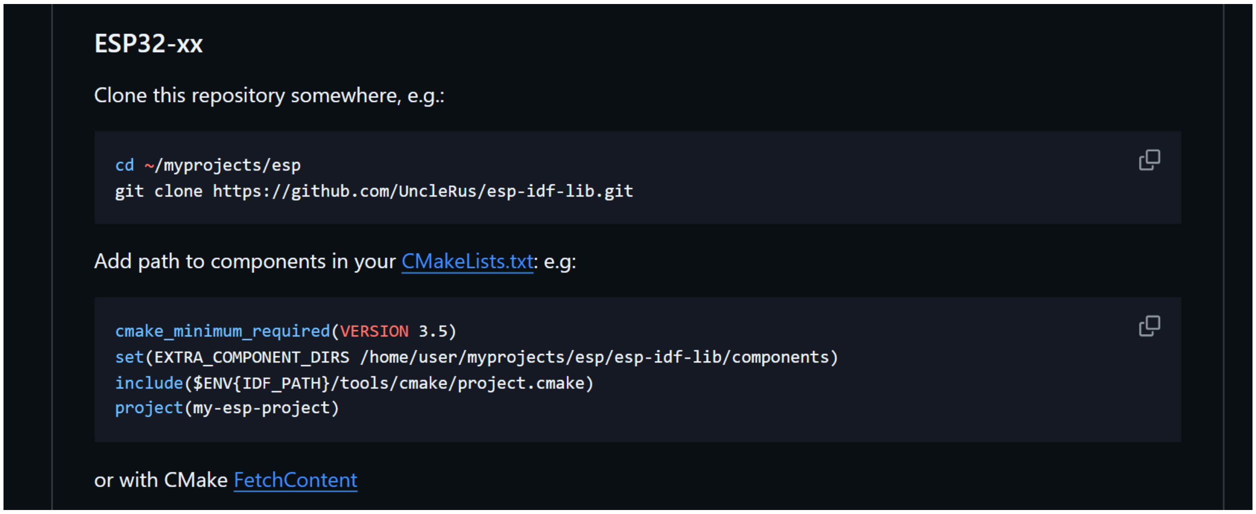
Pin Name	HSPI	VSPI
	GPIO Number	
CS0*	15	5
SCLK	14	18
MISO	12	19
MOSI	13	23
QUADWP	2	22
QUADHD	4	21

```
(Top) → SD SPI Example Configuration
                     Espressif IoT Development Framework Configuration
[ ] Format the card if mount failed
[ ] Format the card as a part of the example
(13) MOSI GPIO number
(12) MISO GPIO number
(14) CLK GPIO number
(15) CS GPIO number
[ ] Debug sd pin connections and pullup strength

[Space/Enter] Toggle/enter  [ESC] Leave menu           [S] Save
[O] Load                    [?] Symbol info            [/] Jump to symbol
[F] Toggle show-help mode   [C] Toggle show-name mode  [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```



```
(Top) → SD SPI Example Configuration
                     Espressif IoT Development
[ ] Format the card if mount failed
[ ] Format the card as a part of the example
(13) MOSI GPIO number
(12) MISO GPIO number
(14) CLK GPIO number
(15) CS GPIO number
[ ] Debug sd pin connections and pullup strength
```

10/8/2024



ESP32-xx

Clone this repository somewhere, e.g.:

```
cd ~/myprojects/esp
git clone https://github.com/UncleRus/esp-idf-lib.git
```

Add path to components in your CMakeLists.txt: e.g:

```
cmake_minimum_required(VERSION 3.5)
set(EXTRA_COMPONENT_DIRS /home/user/myprojects/esp/esp-idf-lib/components)
include($ENV{IDF_PATH}/tools/cmake/project.cmake)
project(my-esp-project)
```

or with CMake FetchContent

Try this ^^^^^^^

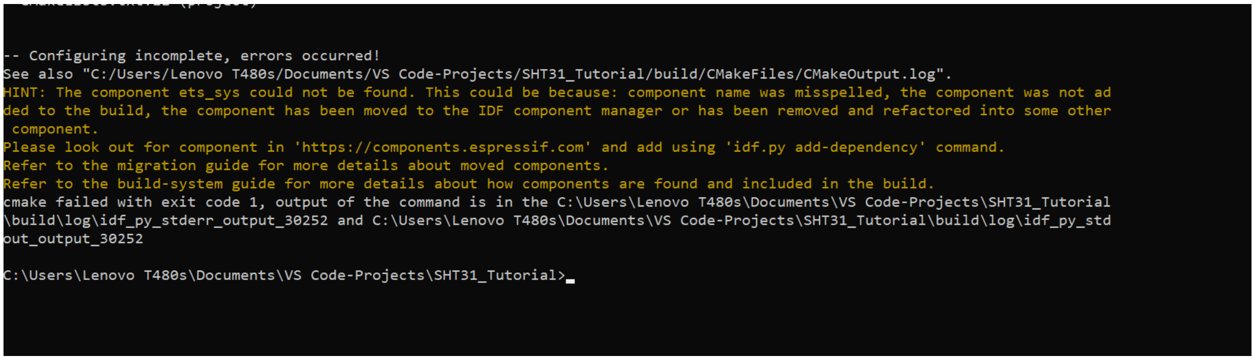
Github link

Should be:
set(EXTRA_COMPOENENT_DIRS /lenovo/docuemtns/vscode/tempsensorEXAMPLE)

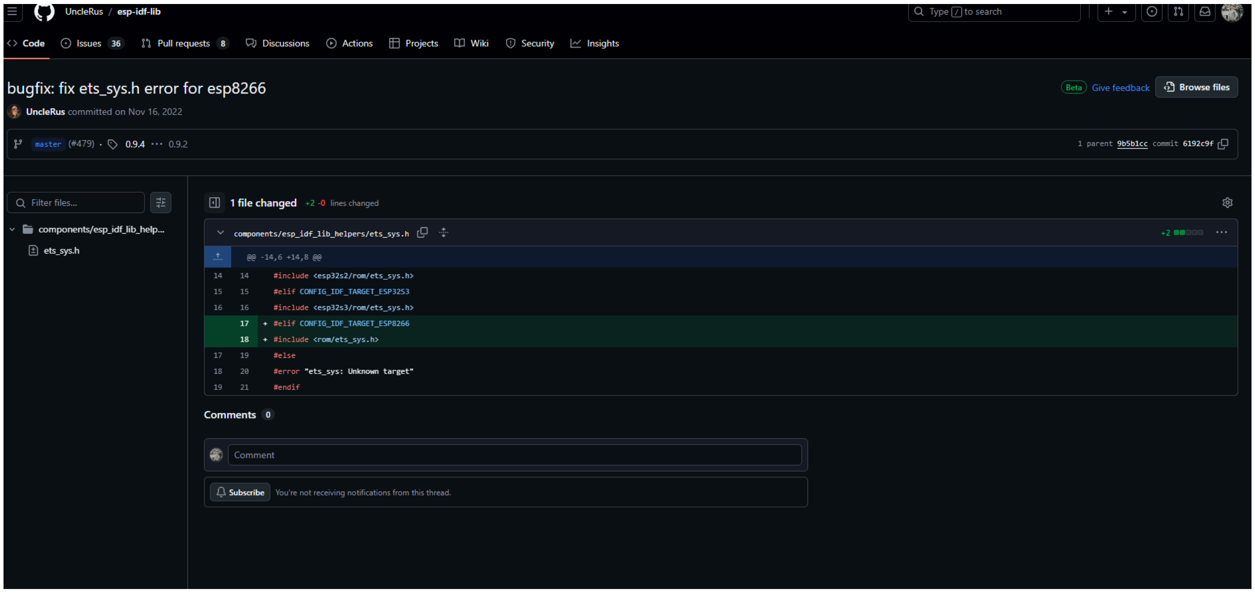
#include errors detected. Consider updating your compile_commands.json or includePath. Squiggles are disabled for this translation unit (C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial\components\sht3x\sht3x.c).

compile_commands.json

espressif.com' and add using 'idf.py add-dependency' command.



```
-- Configuring incomplete, errors occurred!
See also "C:/Users/Lenovo T480s/Documents/VS Code-Projects/SHT31_Tutorial/build/CMakeFiles/CMakeOutput.log".
HINT: The component ets_sys could not be found. This could be because: component name was misspelled, the component was not ad
ded to the build, the component has been moved to the IDF component manager or has been removed and refactored into some other
 component.
Please look out for component in 'https://components.espressif.com' and add using 'idf.py add-dependency' command.
Refer to the migration guide for more details about moved components.
Refer to the build-system guide for more details about how components are found and included in the build.
cmake failed with exit code 1, output of the command is in the C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial
\build\log\idf_py_stderr_output_30252 and C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial\build\log\idf_py_std
out_output_30252

C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial>_
```



bugfix: fix ets_sys.h error for esp8266

UncleRus committed on Nov 16, 2022

master (#479) · 0.9.4 ··· 0.9.2

1 parent 9b5b1cc commit 6192c9f

1 file changed +2 -0 lines changed

components/esp_idf_lib_helpers/ets_sys.h

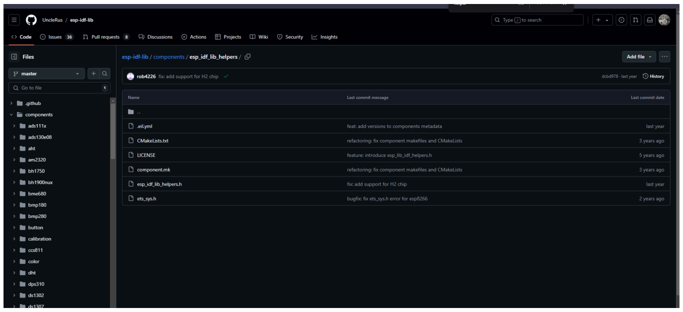
```
@@ -14,6 +14,8 @@
  #include <esp32s2/rom/ets_sys.h>
  #elif CONFIG_IDF_TARGET_ESP32S3
  #include <esp32s3/rom/ets_sys.h>
+ #elif CONFIG_IDF_TARGET_ESP8266
+ #include <rom/ets_sys.h>
  #else
  #error "ets_sys: Unknown target"
  #endif
```

Comments 0

Comment

You're not receiving notifications from this thread.

[Link](#)



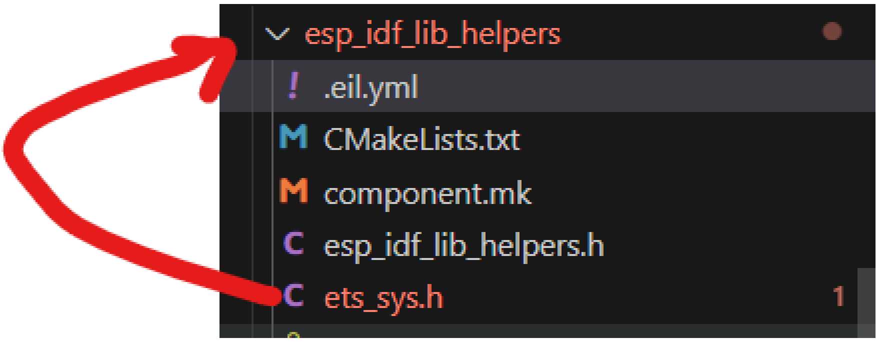
esp-idf-lib / components / esp_idf_lib_helpers /

rob4226 fix: add support for H2 chip ✓ dcbd978 · last year

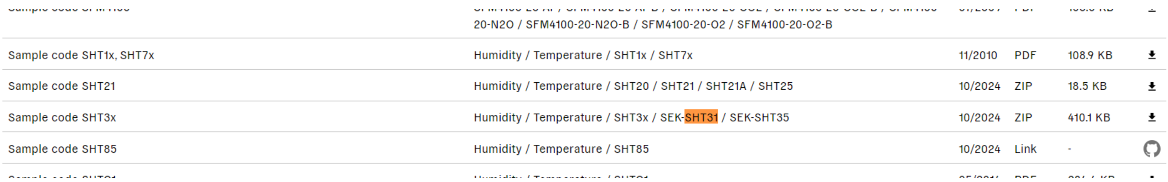
Name	Last commit message	Last commit date
..		
.eil.yml	feat: add versions to components metadata	last year
CMakeLists.txt	refactoring: fix component makefiles and CMakeLists	3 years ago
LICENSE	feature: introduce esp_lib_idf_helpers.h	5 years ago
component.mk	refactoring: fix component makefiles and CMakeLists	3 years ago
esp_idf_lib_helpers.h	fix: add support for H2 chip	last year
ets_sys.h	bugfix: fix ets_sys.h error for esp8266	2 years ago

```
HINT: The component ets_sys could not be found.
```

Dug into this library

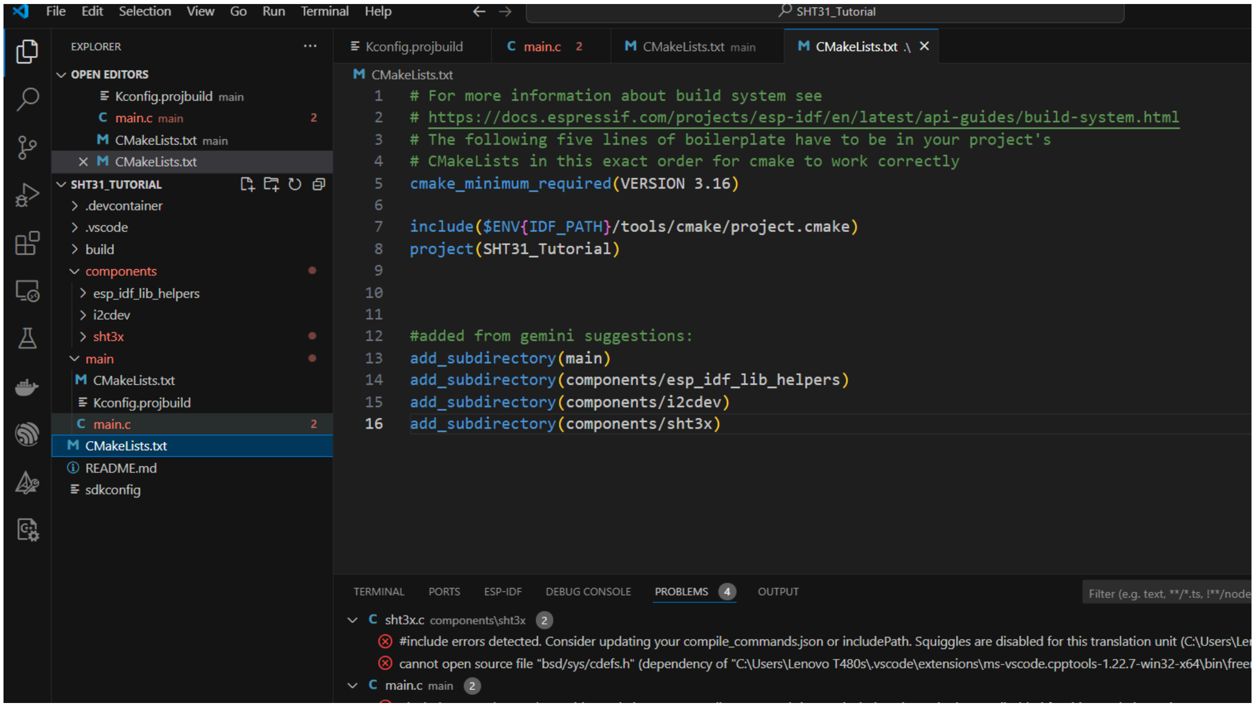


Different name



	20-N2O / SFM4100-20-N2O-B / SFM4100-20-O2 / SFM4100-20-O2-B				
Sample code SHT1x, SHT7x	Humidity / Temperature / SHT1x / SHT7x	11/2010	PDF	108.9 KB	⬇
Sample code SHT21	Humidity / Temperature / SHT20 / SHT21 / SHT21A / SHT25	10/2024	ZIP	18.5 KB	⬇
Sample code SHT3x	Humidity / Temperature / SHT3x / SEK-SHT31 / SEK-SHT35	10/2024	ZIP	410.1 KB	⬇
Sample code SHT85	Humidity / Temperature / SHT85	10/2024	Link	-	

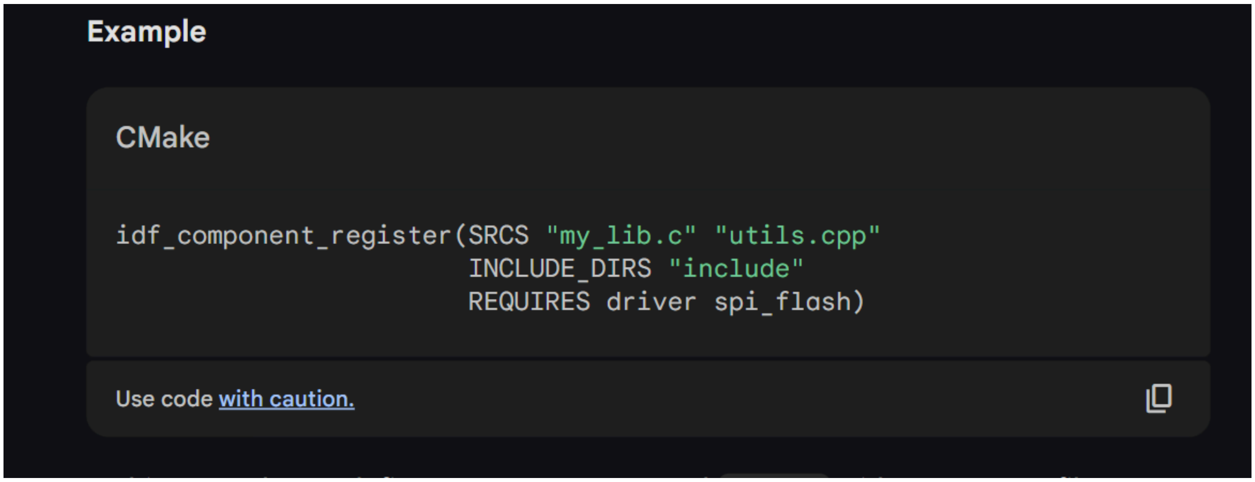
10/6/2024



```
# For more information about build system see
# https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/build-system.html
# The following five lines of boilerplate have to be in your project's
# CMakeLists in this exact order for cmake to work correctly
cmake_minimum_required(VERSION 3.16)

include($ENV{IDF_PATH}/tools/cmake/project.cmake)
project(SHT31_Tutorial)



#added from gemini suggestions:
add_subdirectory(main)
add_subdirectory(components/esp_idf_lib_helpers)
add_subdirectory(components/i2cdev)
add_subdirectory(components/sht3x)
```

```
sht3x.c components\sht3x 2
  #include errors detected. Consider updating your compile_commands.json or includePath. Squiggles are disabled for this translation unit (C:\Users\Le
  cannot open source file "bsd/sys/cdefs.h" (dependency of "C:\Users\Lenovo T480s\.vscode\extensions\ms-vscode.cpptools-1.22.7-win32-x64\bin\free
main.c main 2
```

Added this



Example

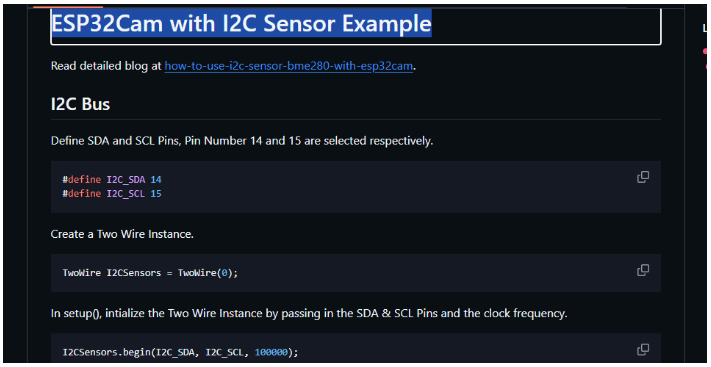
CMake

```
idf_component_register(SRCS "my_lib.c" "utils.cpp"
                       INCLUDE_DIRS "include"
                       REQUIRES driver spi_flash)
```

Use code with caution.

10/6/2024

ESP32Cam with I2C Sensor Example



ESP32Cam with I2C Sensor Example

Read detailed blog at how-to-use-i2c-sensor-bme280-with-esp32cam.

I2C Bus

Define SDA and SCL Pins, Pin Number 14 and 15 are selected respectively.

```
#define I2C_SDA 14
#define I2C_SCL 15
```

Create a Two Wire Instance.

```
TwoWire I2CSensors = TwoWire(0);
```

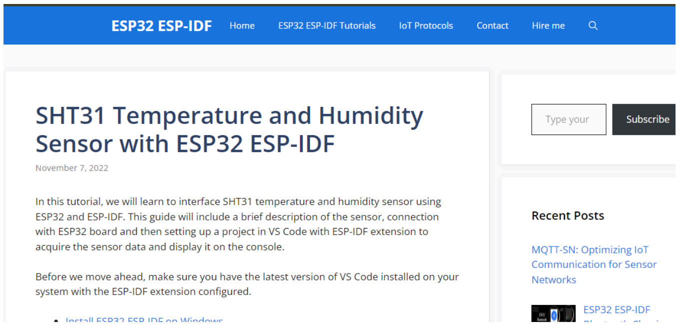
In setup(), intialize the Two Wire Instance by passing in the SDA & SCL Pins and the clock frequency.

```
I2CSensors.begin(I2C_SDA, I2C_SCL, 100000);
```

#define I2C_SDA 14
#define I2C_SCL 15

Mainly following:

[SHT31 Temperature and Humidity Sensor with ESP32 ESP-IDF
November 7, 2022]()



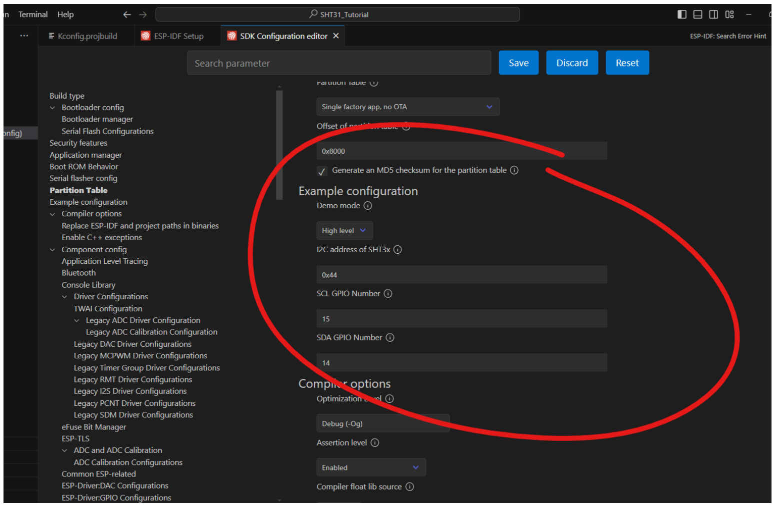
SHT31 Temperature and Humidity Sensor with ESP32 ESP-IDF

November 7, 2022

In this tutorial, we will learn to interface SHT31 temperature and humidity sensor using ESP32 and ESP-IDF. This guide will include a brief description of the sensor, connection with ESP32 board and then setting up a project in VS Code with ESP-IDF extension to acquire the sensor data and display it on the console.

Before we move ahead, make sure you have the latest version of VS Code installed on your system with the ESP-IDF extension configured.

- Install ESP32 ESP-IDF on Windows


Terminal
Help
SHT31_Tutorial
Kconfig.projbuild
ESP-IDF Setup
SDK Configuration editor
ESP-IDF: Search Error Hint
Search parameter
Save
Discard
Reset
Build type
Bootloader config
Bootloader manager
Serial Flash Configurations
Security features
Application manager
Boot ROM Behavior
Serial flasher config
Partition Table
Example configuration
Compiler options
Replace ESP-IDF and project paths in binaries
Enable C++ exceptions
Component config
Application Level Tracing
Bluetooth
Console Library
Driver Configurations
TWAI Configuration
Legacy ADC Driver Configuration
Legacy ADC Calibration Configuration
Legacy DAC Driver Configurations
Legacy MCPWM Driver Configurations
Legacy Timer Group Driver Configurations
Legacy RMT Driver Configurations
Legacy I2S Driver Configurations
Legacy PCNT Driver Configurations
Legacy SDM Driver Configurations
eFuse Bit Manager
ESP-TLS
ADC and ADC Calibration
ADC Calibration Configurations
Common ESP-related
ESP-Driver:DAC Configurations
ESP-Driver:GPIO Configurations

Partition Table: Single factory app, no OTA

Offset of partition table: 0x8000

☑ Generate an MD5 checksum for the partition table

Example configuration

Demo mode: High level

I2C address of SHT3x: 0x44

SCL GPIO Number: 15

SDA GPIO Number: 14

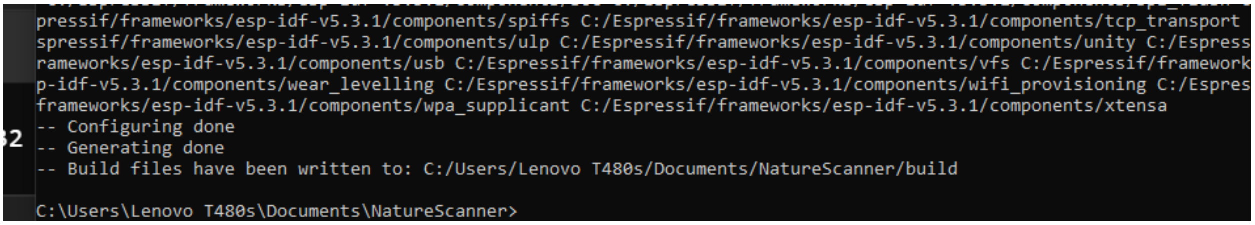
Compiler options

Optimization Level: Debug (-Og)

Assertion level: Enabled

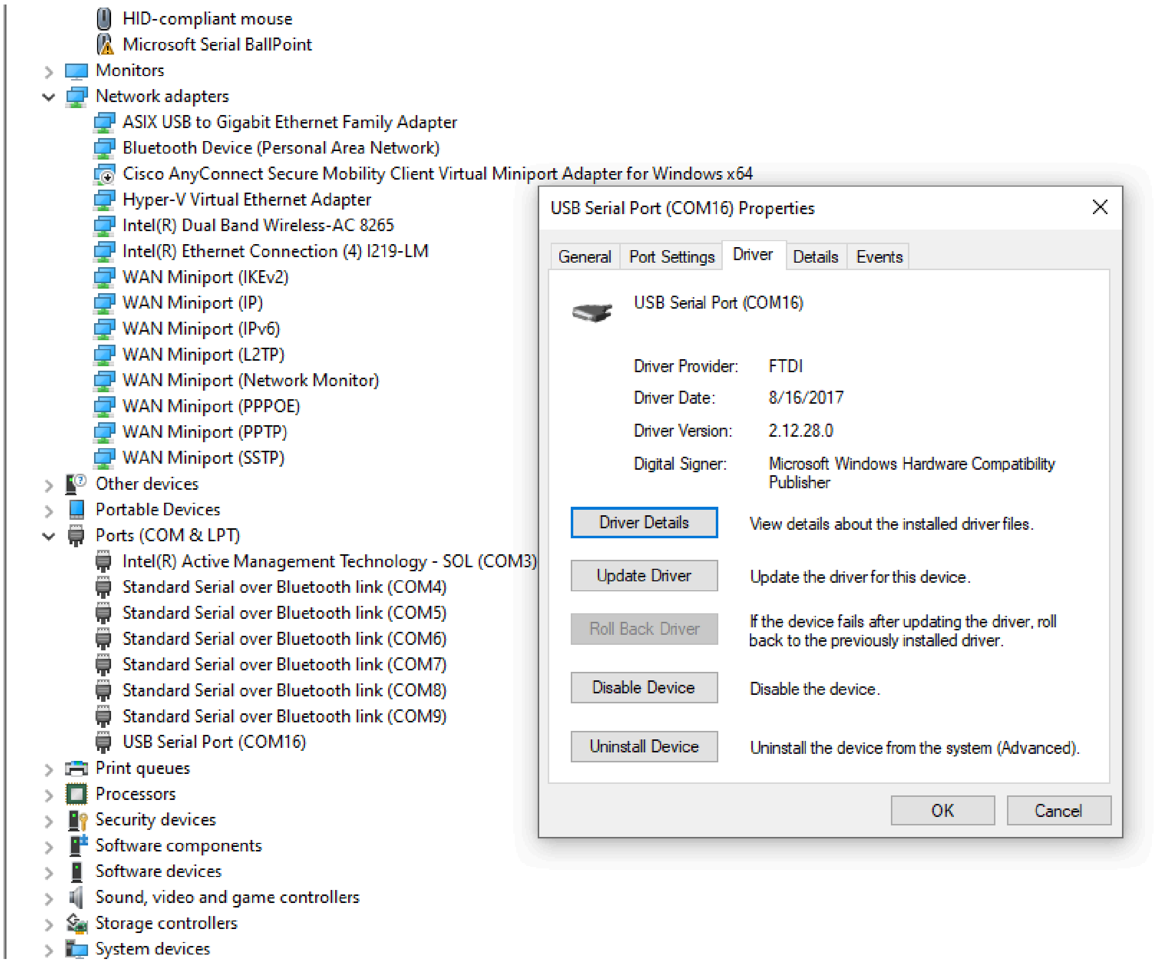
Compiler float lib source

10/5/2024



```
pressif/frameworks/esp-idf-v5.3.1/components/spiffs C:/Espressif/frameworks/esp-idf-v5.3.1/components/tcp_transport
spressif/frameworks/esp-idf-v5.3.1/components/ulp C:/Espressif/frameworks/esp-idf-v5.3.1/components/unity C:/Espress
rameworks/esp-idf-v5.3.1/components/usb C:/Espressif/frameworks/esp-idf-v5.3.1/components/vfs C:/Espressif/framework
p-idf-v5.3.1/components/wear_levelling C:/Espressif/frameworks/esp-idf-v5.3.1/components/wifi_provisioning C:/Espres
frameworks/esp-idf-v5.3.1/components/wpa_supplicant C:/Espressif/frameworks/esp-idf-v5.3.1/components/xtensa
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/Lenovo T480s/Documents/NatureScanner/build

C:\Users\Lenovo T480s\Documents\NatureScanner>
```

Built successful



HID-compliant mouse
Microsoft Serial BallPoint
Monitors
Network adapters
  ASIX USB to Gigabit Ethernet Family Adapter
  Bluetooth Device (Personal Area Network)
  Cisco AnyConnect Secure Mobility Client Virtual Miniport Adapter for Windows x64
  Hyper-V Virtual Ethernet Adapter
  Intel(R) Dual Band Wireless-AC 8265
  Intel(R) Ethernet Connection (4) I219-LM
  WAN Miniport (IKEv2)
  WAN Miniport (IP)
  WAN Miniport (IPv6)
  WAN Miniport (L2TP)
  WAN Miniport (Network Monitor)
  WAN Miniport (PPPOE)
  WAN Miniport (PPTP)
  WAN Miniport (SSTP)
Other devices
Portable Devices
Ports (COM & LPT)
  Intel(R) Active Management Technology - SOL (COM3)
  Standard Serial over Bluetooth link (COM4)
  Standard Serial over Bluetooth link (COM5)
  Standard Serial over Bluetooth link (COM6)
  Standard Serial over Bluetooth link (COM7)
  Standard Serial over Bluetooth link (COM8)
  Standard Serial over Bluetooth link (COM9)
  USB Serial Port (COM16)
Print queues
Processors
Security devices
Software components
Software devices
Sound, video and game controllers
Storage controllers
System devices

USB Serial Port (COM16) Properties

USB Serial Port (COM16)

Driver Provider: FTDI
Driver Date: 8/16/2017
Driver Version: 2.12.28.0
Digital Signer: Microsoft Windows Hardware Compatibility Publisher

View details about the installed driver files.

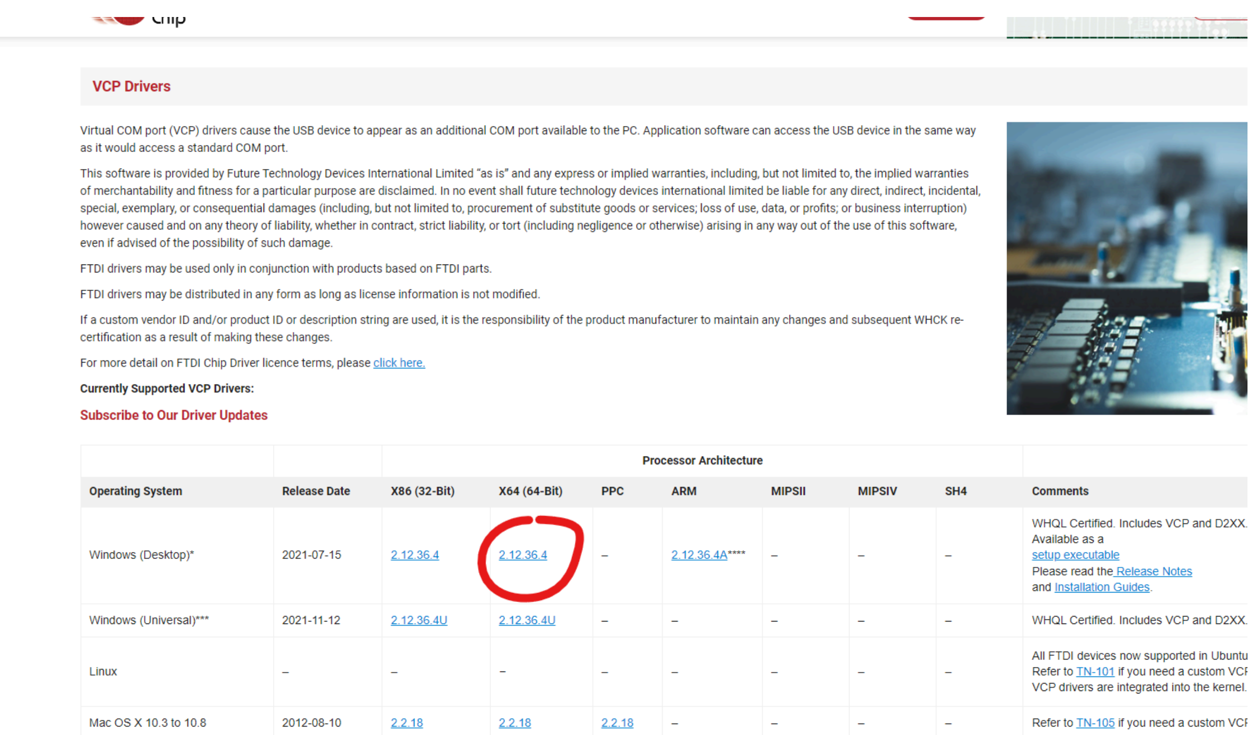
Update the driver for this device.

If the device fails after updating the driver, roll back to the previously installed driver.

Disable the device.

Uninstall the device from the system (Advanced).

FTDI is just USB Serial Port (COM16)



VCP Drivers

Virtual COM port (VCP) drivers cause the USB device to appear as an additional COM port available to the PC. Application software can access the USB device in the same way as it would access a standard COM port.

This software is provided by Future Technology Devices International Limited "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall future technology devices international limited be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

FTDI drivers may be used only in conjunction with products based on FTDI parts.

FTDI drivers may be distributed in any form as long as license information is not modified.

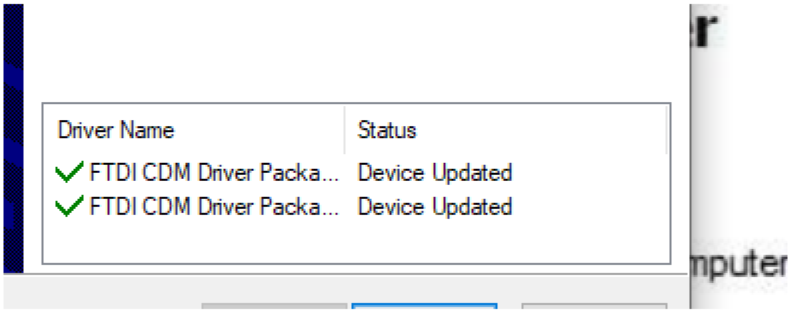
If a custom vendor ID and/or product ID or description string are used, it is the responsibility of the product manufacturer to maintain any changes and subsequent WHCK re-certification as a result of making these changes.

For more detail on FTDI Chip Driver licence terms, please click here.

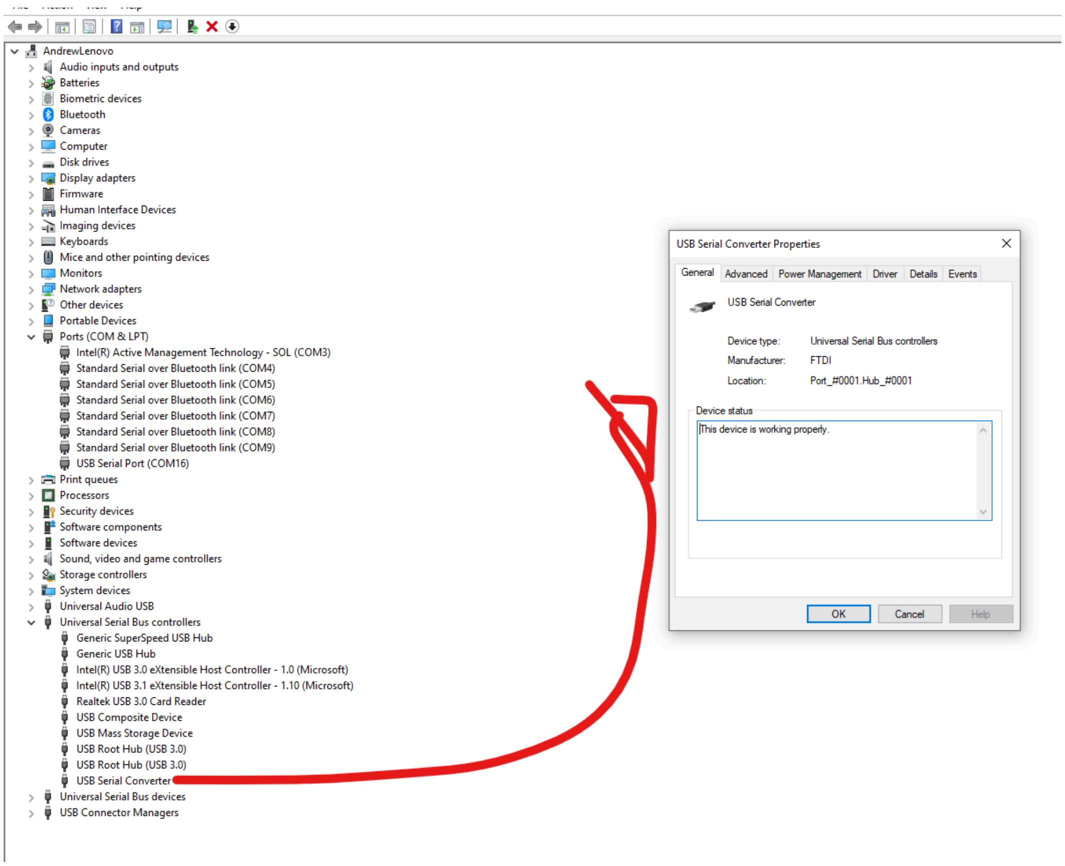
Currently Supported VCP Drivers:

Subscribe to Our Driver Updates

		Processor Architecture							
Operating System	**Release Date**	**X86 (32-Bit)**	**X64 (64-Bit)**	**PPC**	**ARM**	**MIPSII**	**MIPSIV**	**SH4**	**Comments**
Windows (Desktop)*	2021-07-15	2.12.36.4	2.12.36.4	–	2.12.36.4A****	–	–	–	WHQL Certified. Includes VCP and D2XX. Available as a setup executable Please read the Release Notes and Installation Guides.
Windows (Universal)***	2021-11-12	2.12.36.4U	2.12.36.4U	–	–	–	–	–	WHQL Certified. Includes VCP and D2XX.
Linux	–	–	–	–	–	–	–	–	All FTDI devices now supported in Ubuntu Refer to TN-101 if you need a custom VC VCP drivers are integrated into the kernel.
Mac OS X 10.3 to 10.8	2012-08-10	2.2.18	2.2.18	2.2.18	–	–	–	–	Refer to TN-105 if you need a custom VC



Driver Name	Status
✓ FTDI CDM Driver Packa...	Device Updated
✓ FTDI CDM Driver Packa...	Device Updated

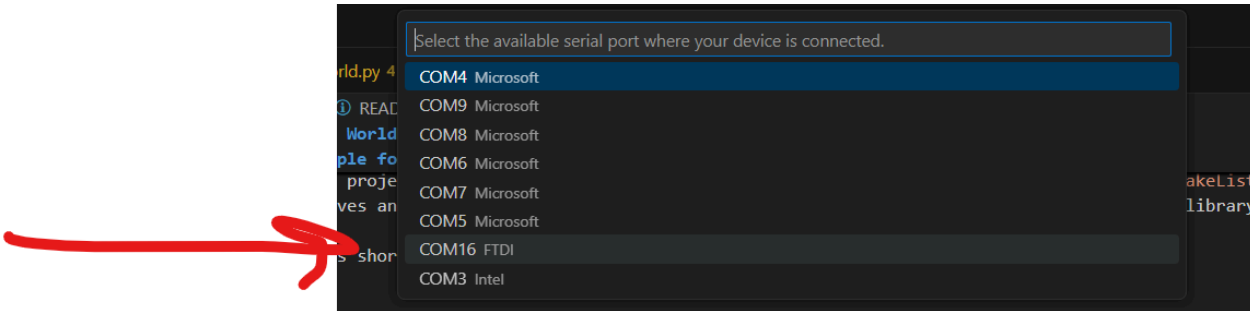

AndrewLenovo
Audio inputs and outputs
Batteries
Biometric devices
Bluetooth
Cameras
Computer
Disk drives
Display adapters
Firmware
Human Interface Devices
Imaging devices
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Other devices
Portable Devices
Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM3)
Standard Serial over Bluetooth link (COM4)
Standard Serial over Bluetooth link (COM5)
Standard Serial over Bluetooth link (COM6)
Standard Serial over Bluetooth link (COM7)
Standard Serial over Bluetooth link (COM8)
Standard Serial over Bluetooth link (COM9)
USB Serial Port (COM16)
Print queues
Processors
Security devices
Software components
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Audio USB
Universal Serial Bus controllers
Generic SuperSpeed USB Hub
Generic USB Hub
Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
Intel(R) USB 3.1 eXtensible Host Controller - 1.10 (Microsoft)
Realtek USB 3.0 Card Reader
USB Composite Device
USB Mass Storage Device
USB Root Hub (USB 3.0)
USB Root Hub (USB 3.0)
USB Serial Converter
Universal Serial Bus devices
USB Connector Managers
USB Serial Converter Properties
General
Advanced
Power Management
Driver
Details
Events
OK
Cancel
Help

USB Serial Converter

Device type: Universal Serial Bus controllers
Manufacturer: FTDI
Location: Port_#0001.Hub_#0001

Device status

This device is working properly.

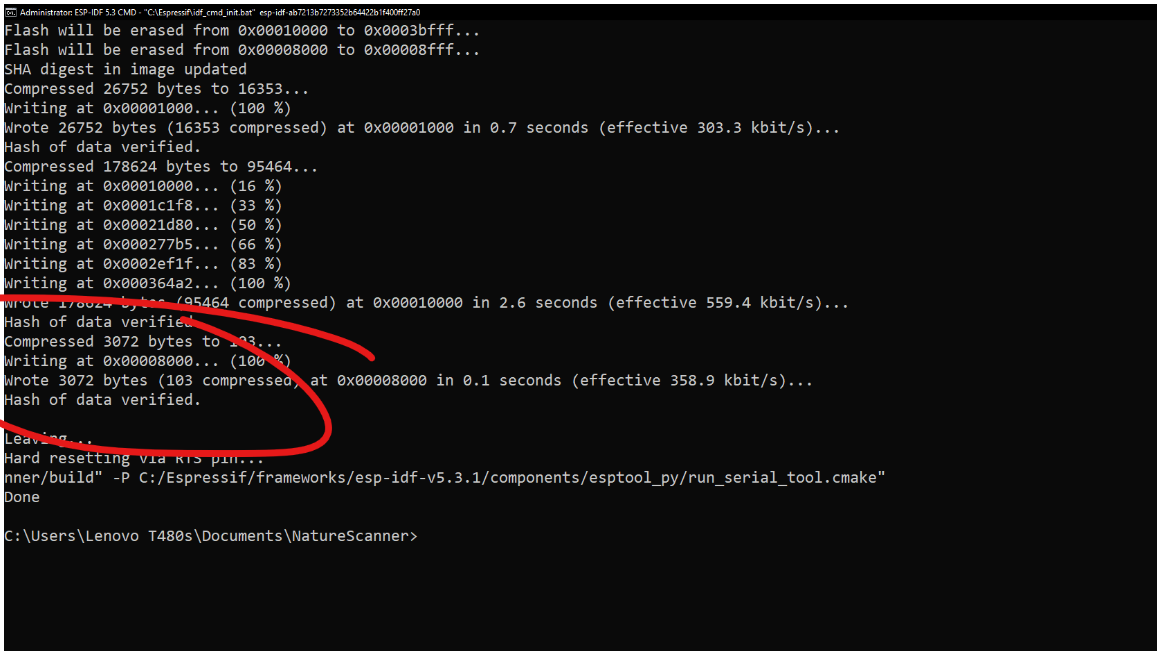


Debug:

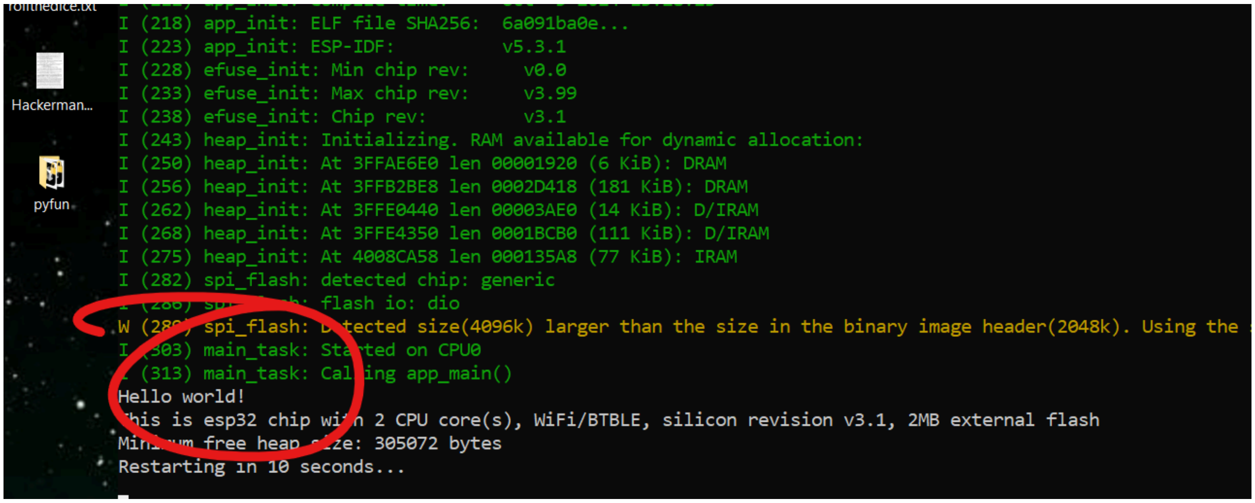
- ☑ ~~Check wiring~~
- ☐

Flashing procedure used:

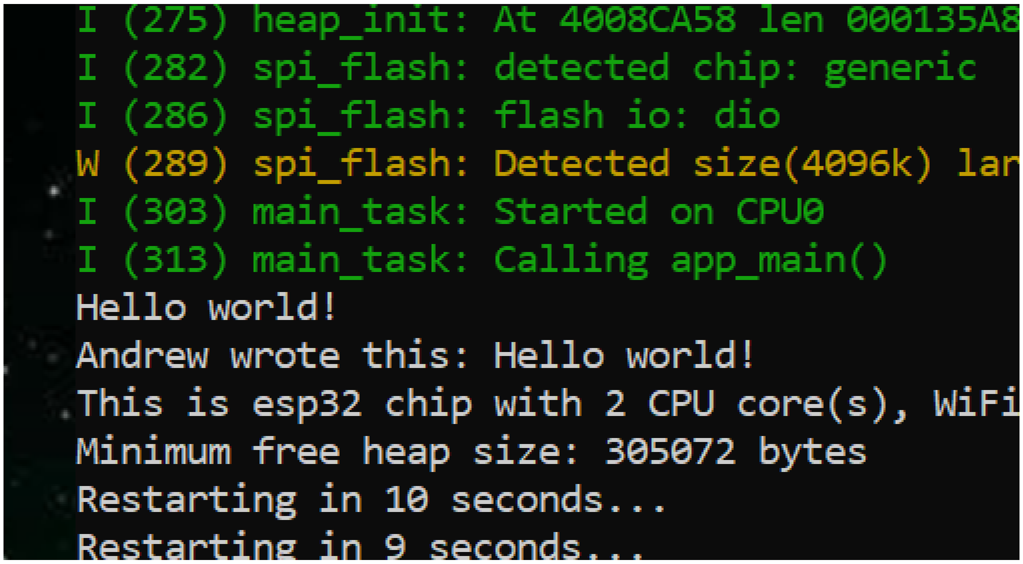
```
cd %IDF_PATH%\examples\get-started\hello_world
idf.py set-target esp32
idf.py menuconfig
idf.py build
```


Administrator: ESP-IDF 5.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-ab7213b7273352b64422b1f400ff27a0

```
Flash will be erased from 0x00010000 to 0x0003bfff...
Flash will be erased from 0x00008000 to 0x00008fff...
SHA digest in image updated
Compressed 26752 bytes to 16353...
Writing at 0x00001000... (100 %)
Wrote 26752 bytes (16353 compressed) at 0x00001000 in 0.7 seconds (effective 303.3 kbit/s)...
Hash of data verified.
Compressed 178624 bytes to 95464...
Writing at 0x00010000... (16 %)
Writing at 0x0001c1f8... (33 %)
Writing at 0x00021d80... (50 %)
Writing at 0x000277b5... (66 %)
Writing at 0x0002ef1f... (83 %)
Writing at 0x000364a2... (100 %)
Wrote 178624 bytes (95464 compressed) at 0x00010000 in 2.6 seconds (effective 559.4 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 103...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.1 seconds (effective 358.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
nner/build" -P C:/Espressif/frameworks/esp-idf-v5.3.1/components/esptool_py/run_serial_tool.cmake"
Done

C:\Users\Lenovo T480s\Documents\NatureScanner>
```



```
I (218) app_init: ELF file SHA256:  6a091ba0e...
I (223) app_init: ESP-IDF:          v5.3.1
I (228) efuse_init: Min chip rev:     v0.0
I (233) efuse_init: Max chip rev:     v3.99
I (238) efuse_init: Chip rev:         v3.1
I (243) heap_init: Initializing. RAM available for dynamic allocation:
I (250) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (256) heap_init: At 3FFB2BE8 len 0002D418 (181 KiB): DRAM
I (262) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (268) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (275) heap_init: At 4008CA58 len 000135A8 (77 KiB): IRAM
I (282) spi_flash: detected chip: generic
I (286) spi_flash: flash io: dio
W (28[illegible]) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the
I (303) main_task: Started on CPU0
I (313) main_task: Calling app_main()
Hello world!
This is esp32 chip with 2 CPU core(s), WiFi/BTBLE, silicon revision v3.1, 2MB external flash
Minimum free heap size: 305072 bytes
Restarting in 10 seconds...
```



```
I (275) heap_init: At 4008CA58 len 000135A8
I (282) spi_flash: detected chip: generic
I (286) spi_flash: flash io: dio
W (289) spi_flash: Detected size(4096k) lar
I (303) main_task: Started on CPU0
I (313) main_task: Calling app_main()
Hello world!
Andrew wrote this: Hello world!
This is esp32 chip with 2 CPU core(s), WiFi
Minimum free heap size: 305072 bytes
Restarting in 10 seconds...
Restarting in 9 seconds...
```

10/4/2024

(maybe around this time????)
***taken from DDD

🏍️*Build procedure used:*

ESP32 build procedure v001

cd C:\Users\Lenovo T480s\Documents\VS Code-Projects\SHT31_Tutorial(*whatever is top level CMakeLists.txt)
~~idf.py clean~~
idf.py clean (to delete build folder) **[idf.py all clean→ builds and creates esp32 image??]
    Executes ninja all
Remove daisy chain Vcc
Remove remove programming pin
Reset 5v
Insert programming pin
idf.py menuconfig
    *Loads configuration: sdkconfig
idf.py build **we might be ok to skip and ready to flash at this point
Reset 5v
idf.py -p COM16 flash
Remove programming pin
Insert daisy chain Vcc pin

Insert SDA and SCL pins
idf.py -p COM16 monitor

—---|

ESP32 build procedure v002

working……..

Remove programming pin
Remove daisy chain Vcc
Remove SDA and SCL
Reset 5vN
idf.py all clean
Reset 5vN
Insert programming pin
idf.py -p COM16 flash
Remove programming pin
Insert daisy chain Vcc pin
Insert SDA and SCL pins

—---|

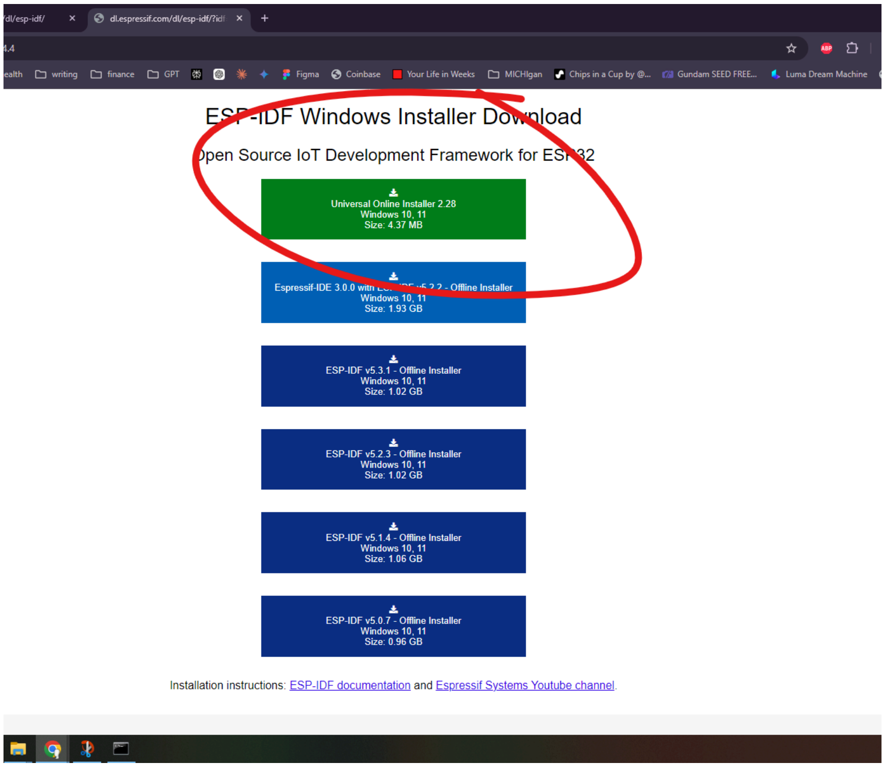
ESP32 build procedure v003

working……..

Wire ESP32-CAM to FT232RL
Wire SHT31 to ESP32-CAM
***add diagram
Reset USB cable from FT232RL to PC
idf.py all clean
idf.py -p COM16 flash
Remove programming pin
Click reset button

—---|

10/3/2024



ESP-IDF Windows Installer Download

Open Source IoT Development Framework for ESP32

Universal Online Installer 2.28
Windows 10, 11
Size: 4.37 MB

Espressif-IDE 3.0.0 with ESP-IDF v5.2.2 - Offline Installer
Windows 10, 11
Size: 1.93 GB

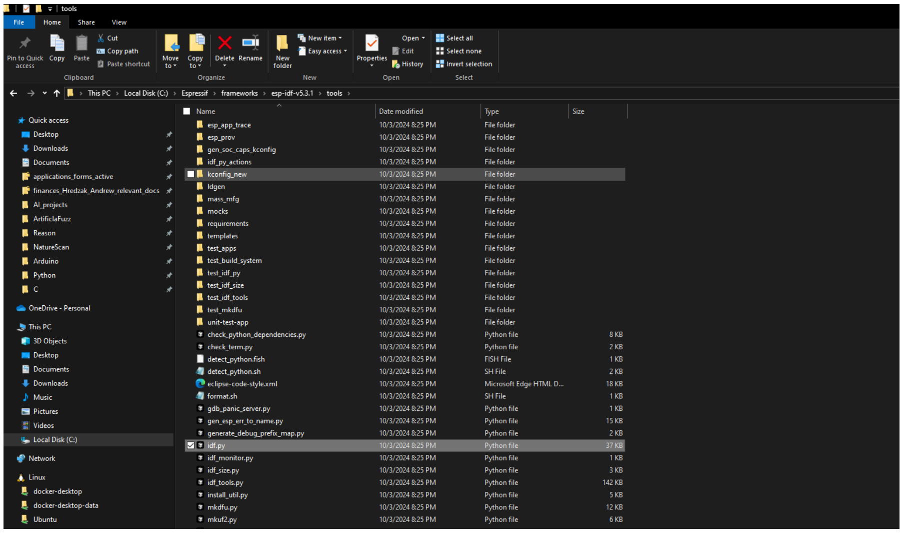
ESP-IDF v5.3.1 - Offline Installer
Windows 10, 11
Size: 1.02 GB

ESP-IDF v5.2.3 - Offline Installer
Windows 10, 11
Size: 1.02 GB

ESP-IDF v5.1.4 - Offline Installer
Windows 10, 11
Size: 1.06 GB

ESP-IDF v5.0.7 - Offline Installer
Windows 10, 11
Size: 0.96 GB

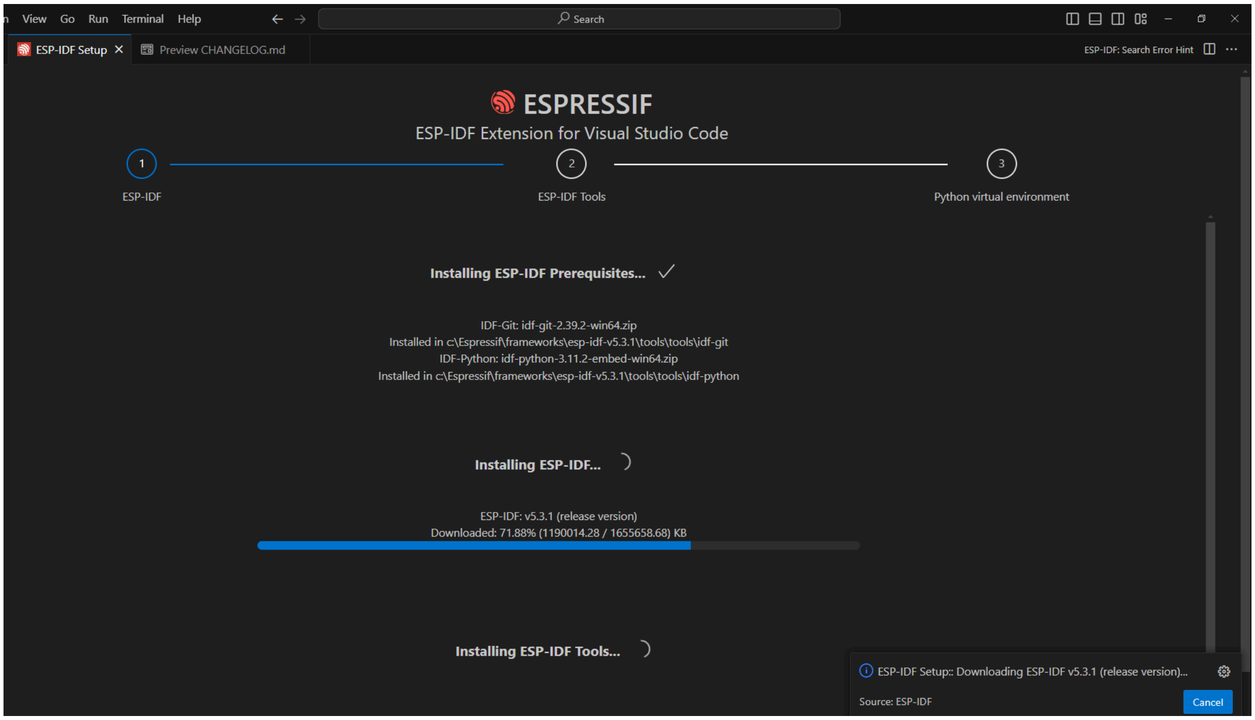
Installation instructions: ESP-IDF documentation and Espressif Systems Youtube channel.



This PC › Local Disk (C:) › Espressif › frameworks › esp-idf-v5.3.1 › tools ›

Name	Date modified	Type	Size
esp_app_trace	10/3/2024 8:25 PM	File folder	
esp_prov	10/3/2024 8:25 PM	File folder	
gen_soc_caps_kconfig	10/3/2024 8:25 PM	File folder	
idf_py_actions	10/3/2024 8:25 PM	File folder	
☐ kconfig_new	10/3/2024 8:25 PM	File folder	
ldgen	10/3/2024 8:25 PM	File folder	
mass_mfg	10/3/2024 8:25 PM	File folder	
mocks	10/3/2024 8:25 PM	File folder	
requirements	10/3/2024 8:25 PM	File folder	
templates	10/3/2024 8:25 PM	File folder	
test_apps	10/3/2024 8:25 PM	File folder	
test_build_system	10/3/2024 8:25 PM	File folder	
test_idf_py	10/3/2024 8:25 PM	File folder	
test_idf_size	10/3/2024 8:25 PM	File folder	
test_idf_tools	10/3/2024 8:25 PM	File folder	
test_mkdfu	10/3/2024 8:25 PM	File folder	
unit-test-app	10/3/2024 8:25 PM	File folder	
check_python_dependencies.py	10/3/2024 8:25 PM	Python file	8 KB
check_term.py	10/3/2024 8:25 PM	Python file	2 KB
detect_python.fish	10/3/2024 8:25 PM	FISH File	1 KB
detect_python.sh	10/3/2024 8:25 PM	SH File	2 KB
eclipse-code-style.xml	10/3/2024 8:25 PM	Microsoft Edge HTML D...	18 KB
format.sh	10/3/2024 8:25 PM	SH File	1 KB
gdb_panic_server.py	10/3/2024 8:25 PM	Python file	1 KB
gen_esp_err_to_name.py	10/3/2024 8:25 PM	Python file	15 KB
generate_debug_prefix_map.py	10/3/2024 8:25 PM	Python file	2 KB
☑ idf.py	10/3/2024 8:25 PM	Python file	37 KB
idf_monitor.py	10/3/2024 8:25 PM	Python file	1 KB
idf_size.py	10/3/2024 8:25 PM	Python file	3 KB
idf_tools.py	10/3/2024 8:25 PM	Python file	142 KB
install_util.py	10/3/2024 8:25 PM	Python file	5 KB
mkdfu.py	10/3/2024 8:25 PM	Python file	12 KB
mkuf2.py	10/3/2024 8:25 PM	Python file	6 KB

Found it



Installing ESP-IDF Prerequisites... ✓

IDF-Git: idf-git-2.39.2-win64.zip
Installed in c:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\idf-git
IDF-Python: idf-python-3.11.2-embed-win64.zip
Installed in c:\Espressif\frameworks\esp-idf-v5.3.1\tools\tools\idf-python

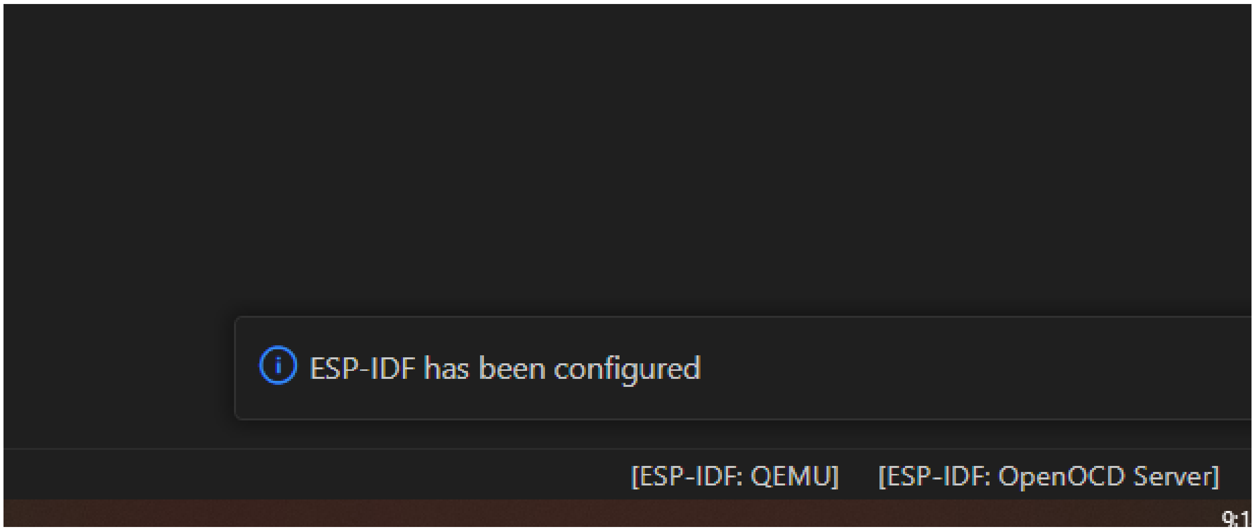
Installing ESP-IDF...

ESP-IDF: v5.3.1 (release version)
Downloaded: 71.88% (1190014.28 / 1655658.68) KB

Installing ESP-IDF Tools...

ESP-IDF Setup: Downloading ESP-IDF v5.3.1 (release version)...
Source: ESP-IDF

I just had to select the right version.



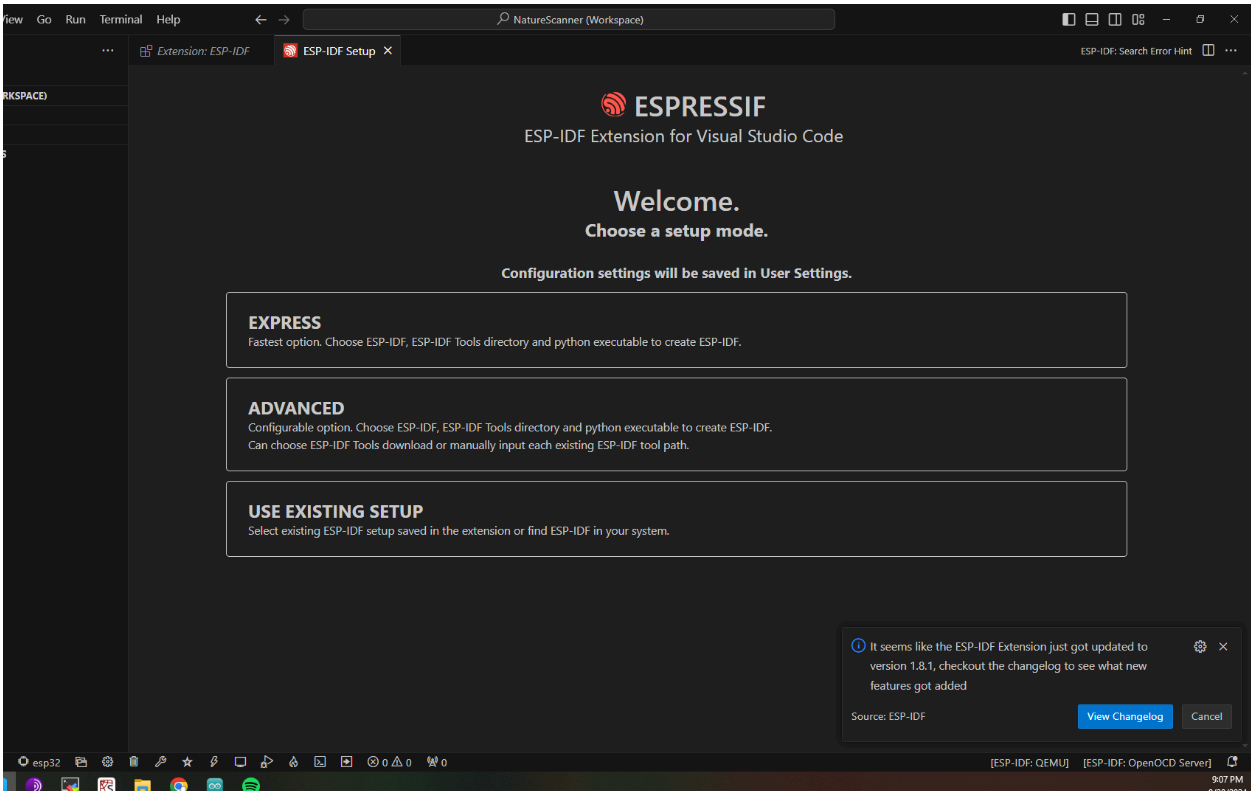
ESP-IDF has been configured

Reference this video for running hello world on your ESP32 chip:
Getting Started with ESP-IDF on Windows

Next steps:

- ☑ ~~File structure for project~~
- ☑ ~~VScode config~~
- ☑ ~~Wait for USB UART board to arrive~~

9/30/2024



ESPRESSIF

ESP-IDF Extension for Visual Studio Code

Welcome.

Choose a setup mode.

Configuration settings will be saved in User Settings.

EXPRESS
Fastest option. Choose ESP-IDF, ESP-IDF Tools directory and python executable to create ESP-IDF.

ADVANCED
Configurable option. Choose ESP-IDF, ESP-IDF Tools directory and python executable to create ESP-IDF.
Can choose ESP-IDF Tools download or manually input each existing ESP-IDF tool path.

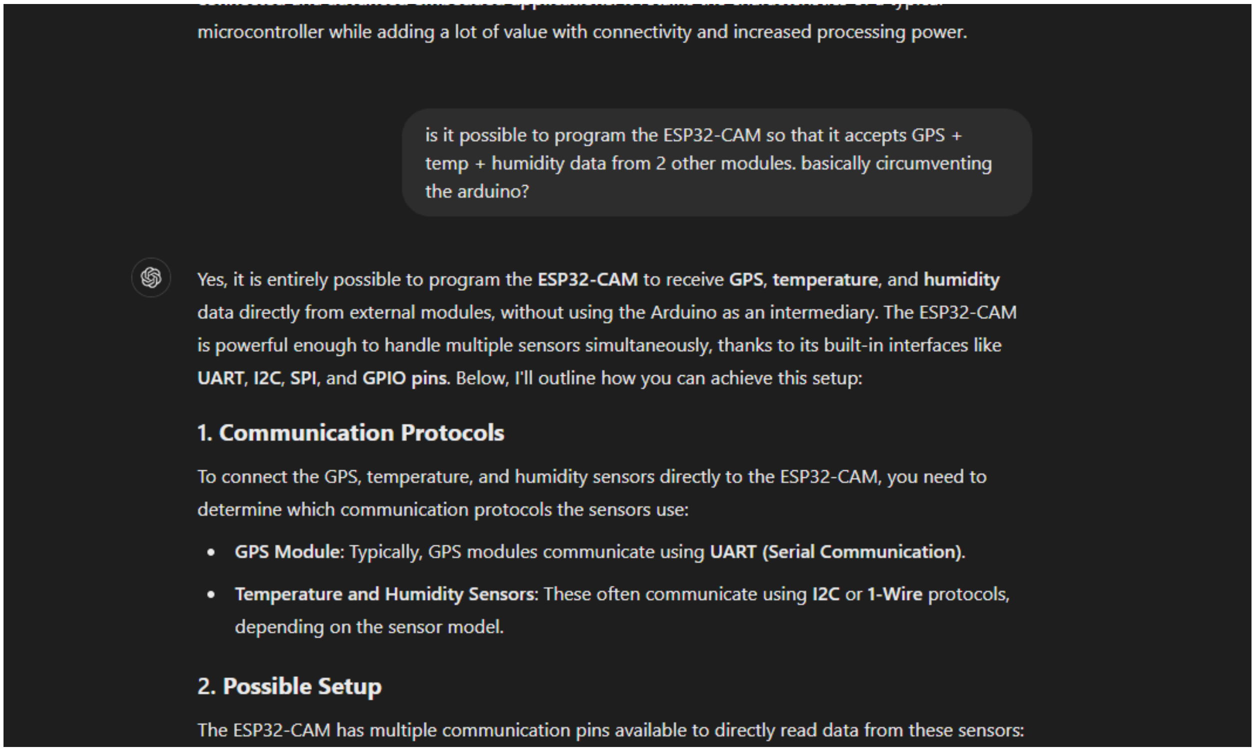
USE EXISTING SETUP
Select existing ESP-IDF setup saved in the extension or find ESP-IDF in your system.

It seems like the ESP-IDF Extension just got updated to version 1.8.1, checkout the changelog to see what new features got added

Source: ESP-IDF

We cooking

9/29/2024



microcontroller while adding a lot of value with connectivity and increased processing power.

is it possible to program the ESP32-CAM so that it accepts GPS + temp + humidity data from 2 other modules. basically circumventing the arduino?

Yes, it is entirely possible to program the **ESP32-CAM** to receive **GPS**, **temperature**, and **humidity** data directly from external modules, without using the Arduino as an intermediary. The ESP32-CAM is powerful enough to handle multiple sensors simultaneously, thanks to its built-in interfaces like **UART**, **I2C**, **SPI**, and **GPIO pins**. Below, I'll outline how you can achieve this setup:

1. Communication Protocols

To connect the GPS, temperature, and humidity sensors directly to the ESP32-CAM, you need to determine which communication protocols the sensors use:

- **GPS Module**: Typically, GPS modules communicate using **UART (Serial Communication)**.
- **Temperature and Humidity Sensors**: These often communicate using **I2C** or **1-Wire** protocols, depending on the sensor model.

2. Possible Setup

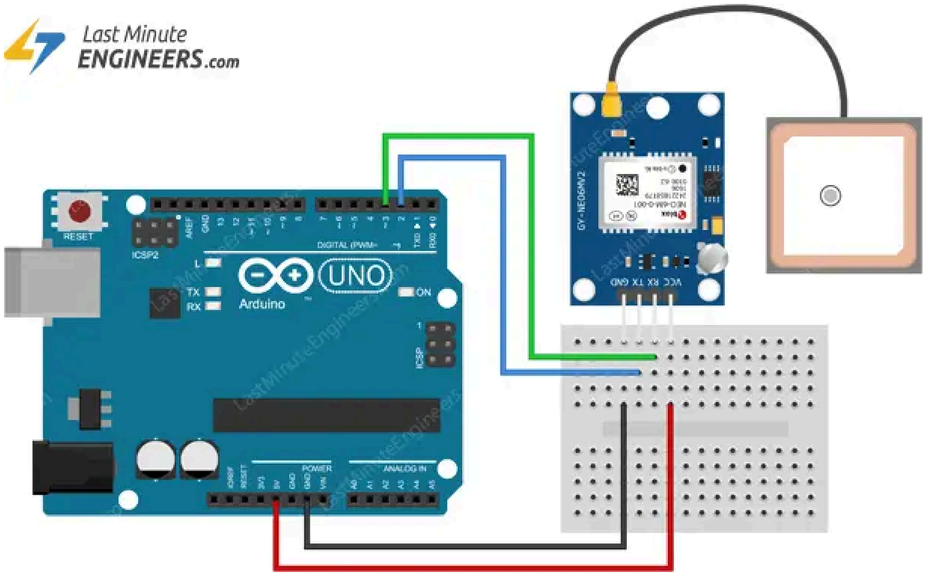
The ESP32-CAM has multiple communication pins available to directly read data from these sensors:

But how do i power it????

9/23/2024

Maybe around this time??
Pulled from DDD

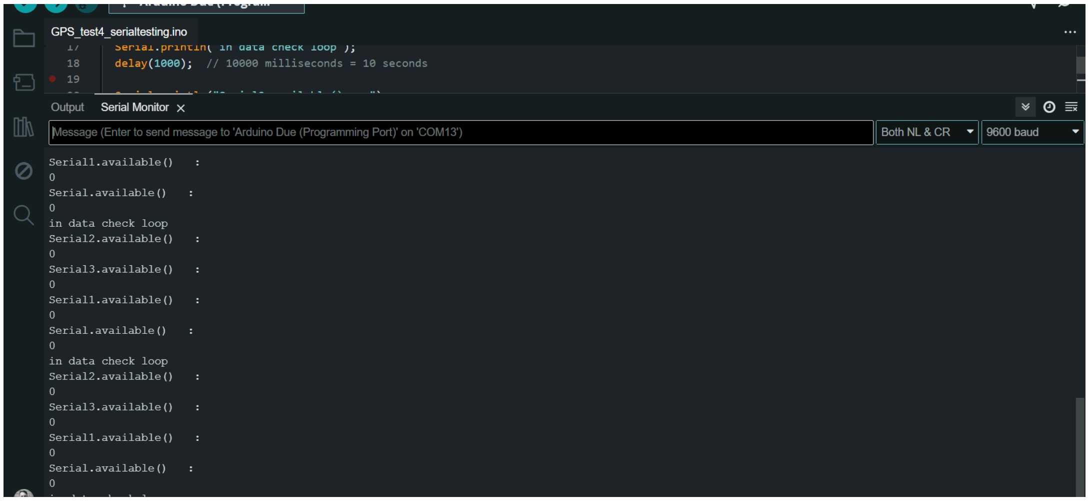
Wiring NEO-6M GPS Module to Arduino UNO (example)



Serial port connections

For `Serial1`: Connect pin 18 (TX1) to pin 19 (RX1).
For `Serial2`: Connect pin 16 (TX2) to pin 17 (RX2).
For `Serial3`: Connect pin 14 (TX3) to pin 15 (RX3).

GPS debug



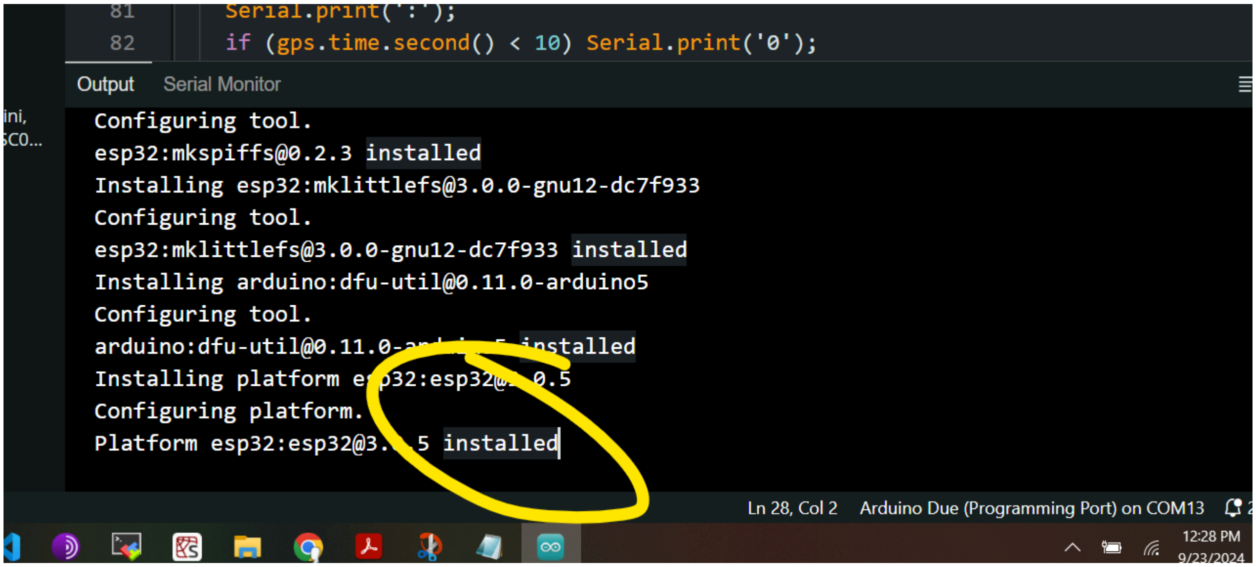
```
Serial.println( in data check loop );
delay(1000);  // 10000 milliseconds = 10 seconds
```

```
Message (Enter to send message to 'Arduino Due (Programming Port)' on 'COM13')
```

```
Serial1.available()   :
0
Serial.available()   :
0
in data check loop
Serial2.available()   :
0
Serial3.available()   :
0
Serial1.available()   :
0
Serial.available()   :
0
in data check loop
Serial2.available()   :
0
Serial3.available()   :
0
Serial1.available()   :
0
Serial.available()   :
0
```

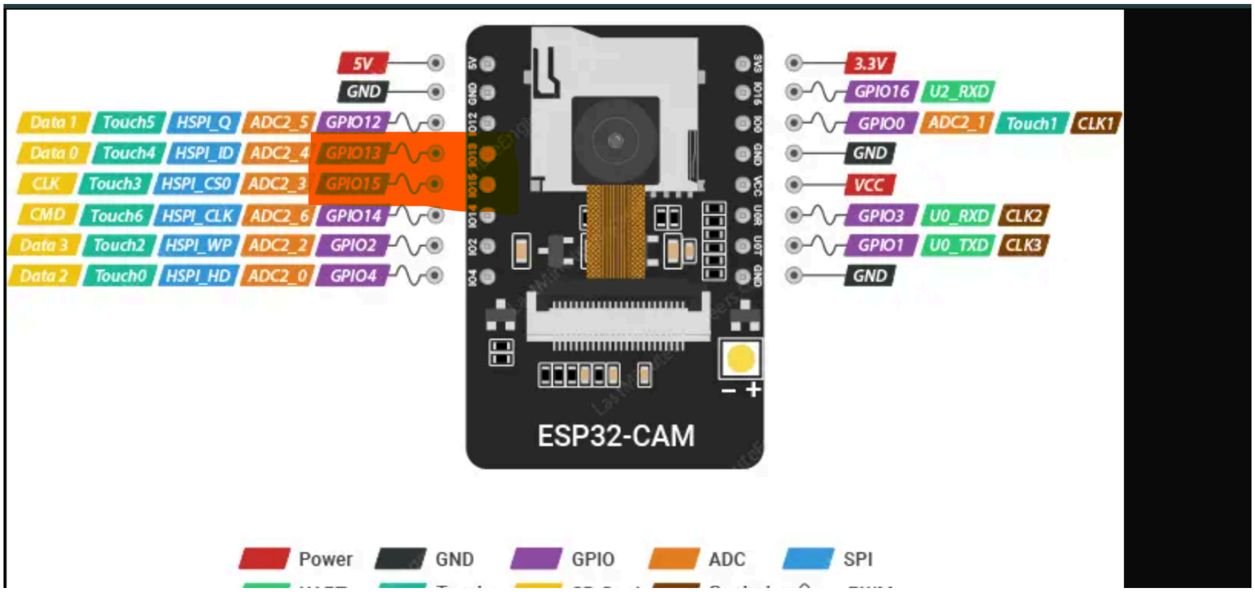
Nothing on any serial ports
Seeing the TX LED light up each time though

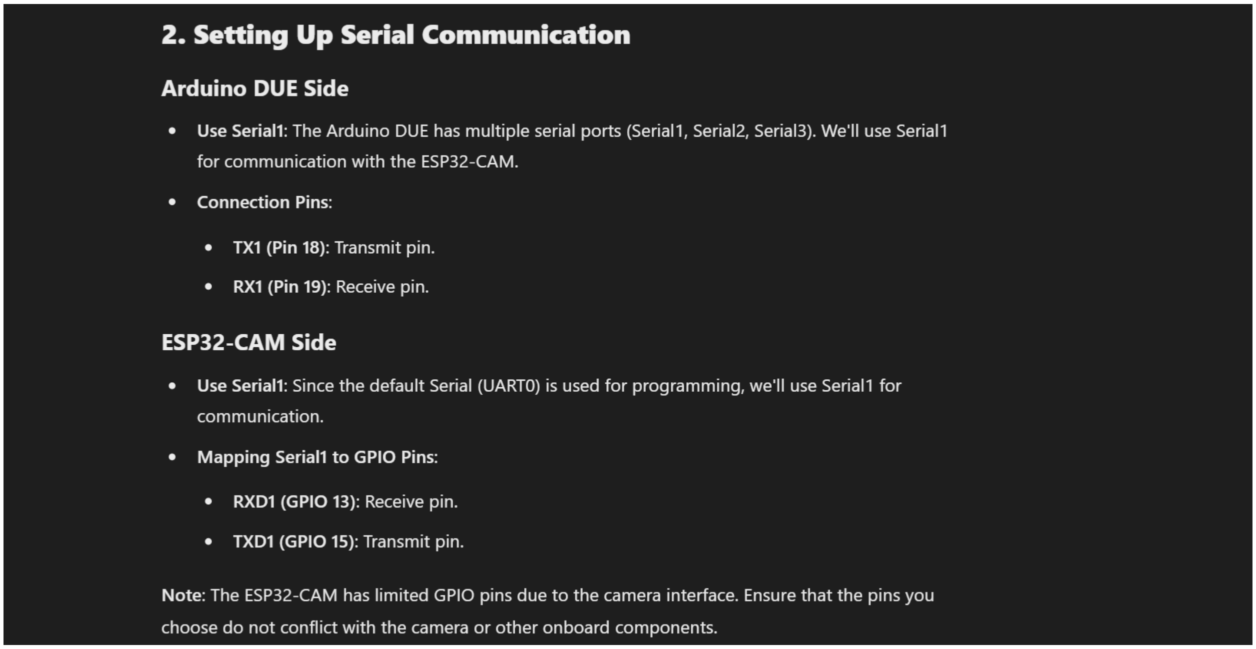
9/23/2024



```
Serial.print(':');
if (gps.time.second() < 10) Serial.print('0');
```

```
Configuring tool.
esp32:mkspiffs@0.2.3 installed
Installing esp32:mklittlefs@3.0.0-gnu12-dc7f933
Configuring tool.
esp32:mklittlefs@3.0.0-gnu12-dc7f933 installed
Installing arduino:dfu-util@0.11.0-arduino5
Configuring tool.
arduino:dfu-util@0.11.0-arduino5 installed
Installing platform esp32:esp32@3.0.5
Configuring platform.
Platform esp32:esp32@3.0.5 installed
```





2. Setting Up Serial Communication

Arduino DUE Side

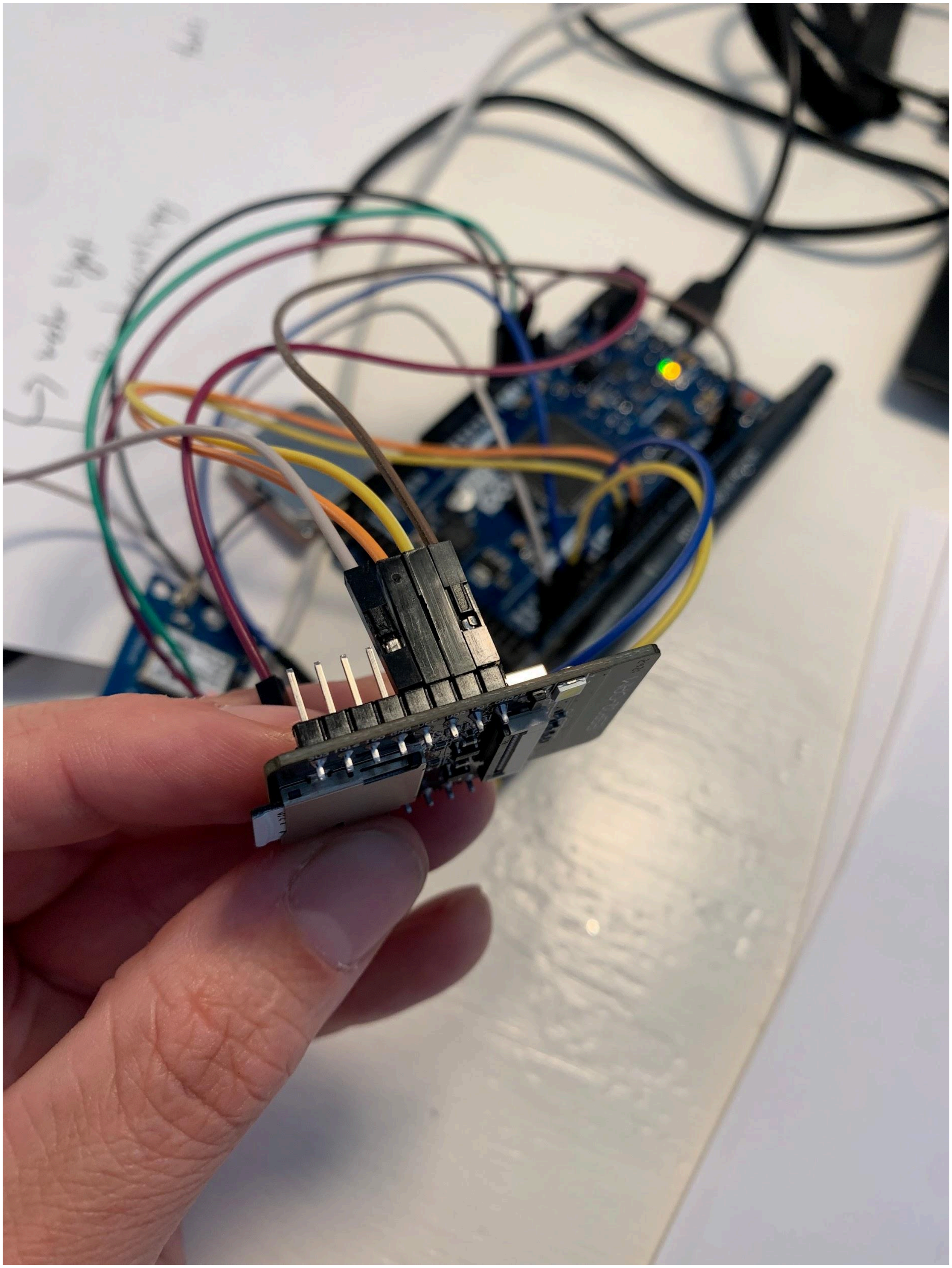
- **Use Serial1**: The Arduino DUE has multiple serial ports (Serial1, Serial2, Serial3). We'll use Serial1 for communication with the ESP32-CAM.
- **Connection Pins**:
  - **TX1 (Pin 18)**: Transmit pin.
  - **RX1 (Pin 19)**: Receive pin.

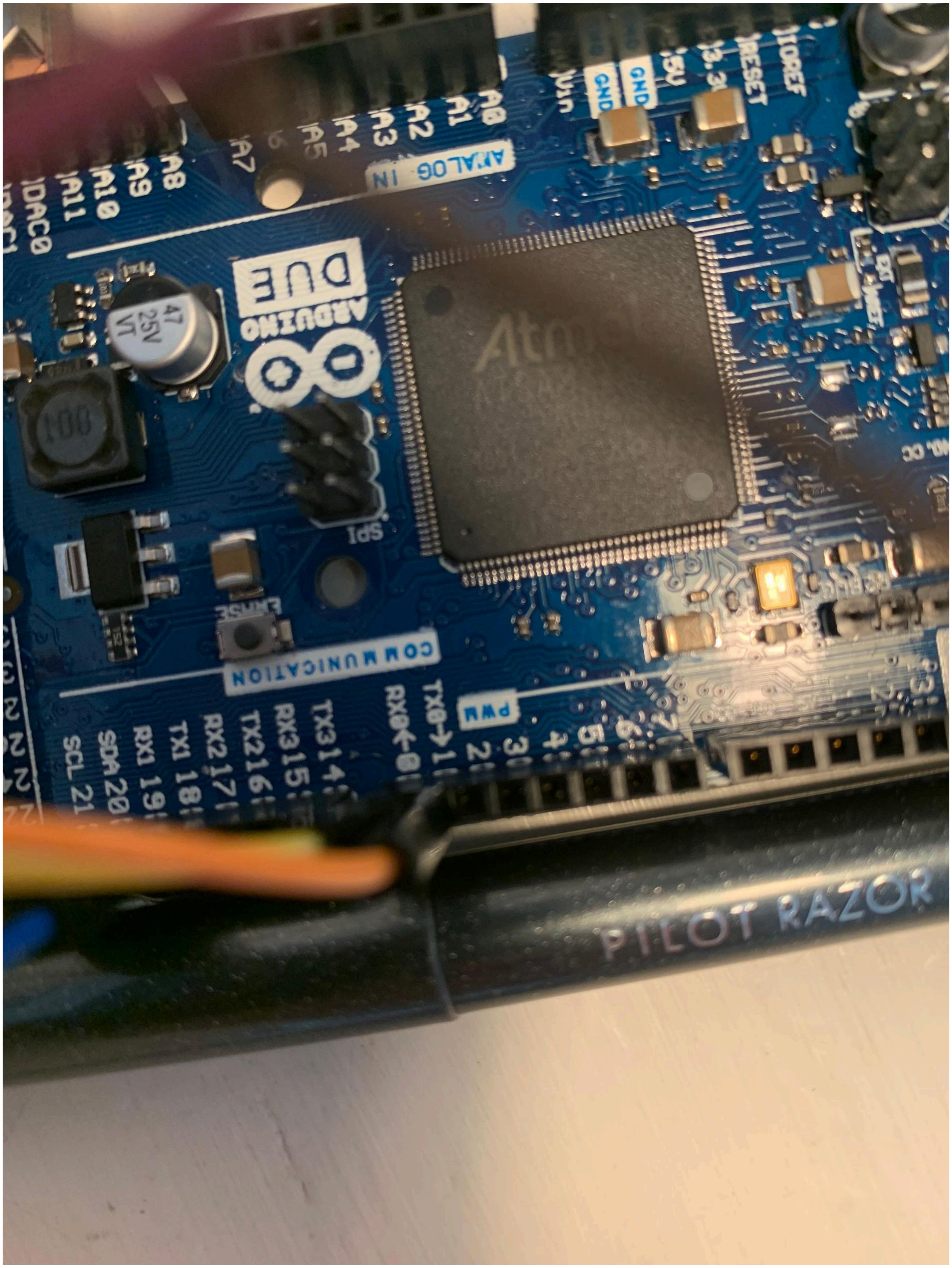
ESP32-CAM Side

- **Use Serial1**: Since the default Serial (UART0) is used for programming, we'll use Serial1 for communication.
- **Mapping Serial1 to GPIO Pins:**
  - **RXD1 (GPIO 13)**: Receive pin.
  - **TXD1 (GPIO 15)**: Transmit pin.

Note: The ESP32-CAM has limited GPIO pins due to the camera interface. Ensure that the pins you choose do not conflict with the camera or other onboard components.

V1 config:



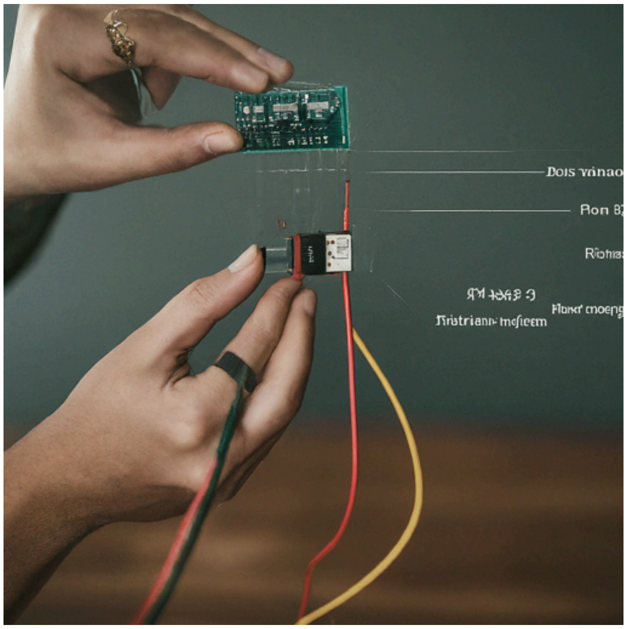

ARDUINO
DUE
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
A7
A8
A9
A10
A11
DAC0
DAC1
SPI
Atmel
COMMUNICATION
PWM
TX0→1
RX0←0
TX3 14
RX3 15
TX2 16
RX2 17
TX1 18
RX1 19
SDA 20
SCL 21
PILOT RAZOR

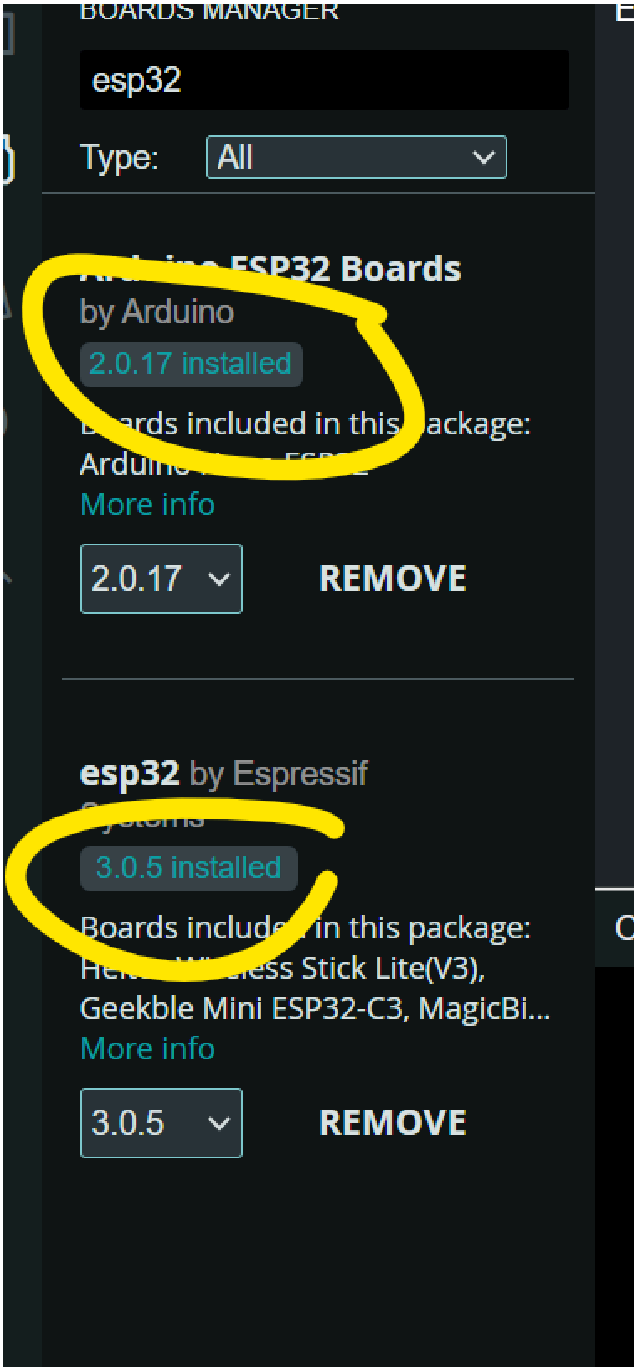
V2 config

Wiring Diagram

- **Arduino DUE TX1 (Pin 18) ➔ ESP32-CAM RXD1 (GPIO 13)**
- **Arduino DUE RX1 (Pin 19) ➔ ESP32-CAM TXD1 (GPIO 15)**
- **Common Ground**: Connect GND from both devices together.

(Note: Since I can't include images, please refer to the pinouts of both devices for accurate wiring.)





BOARDS MANAGER

esp32

Type: All

Arduino ESP32 Boards by Arduino

2.0.17 installed

Boards included in this package:
Arduino Nano ESP32

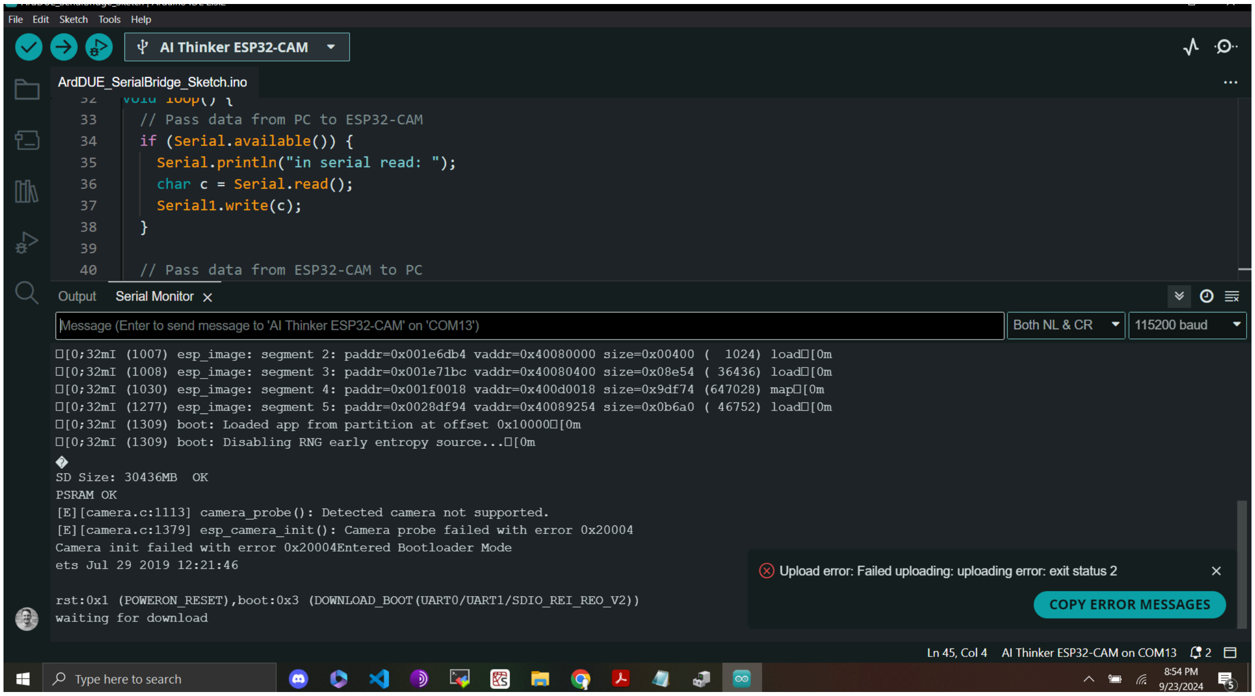
More info

esp32 by Espressif Systems

3.0.5 installed

Boards included in this package:
Heltec Wireless Stick Lite(V3), Geekble Mini ESP32-C3, MagicBi...

More info

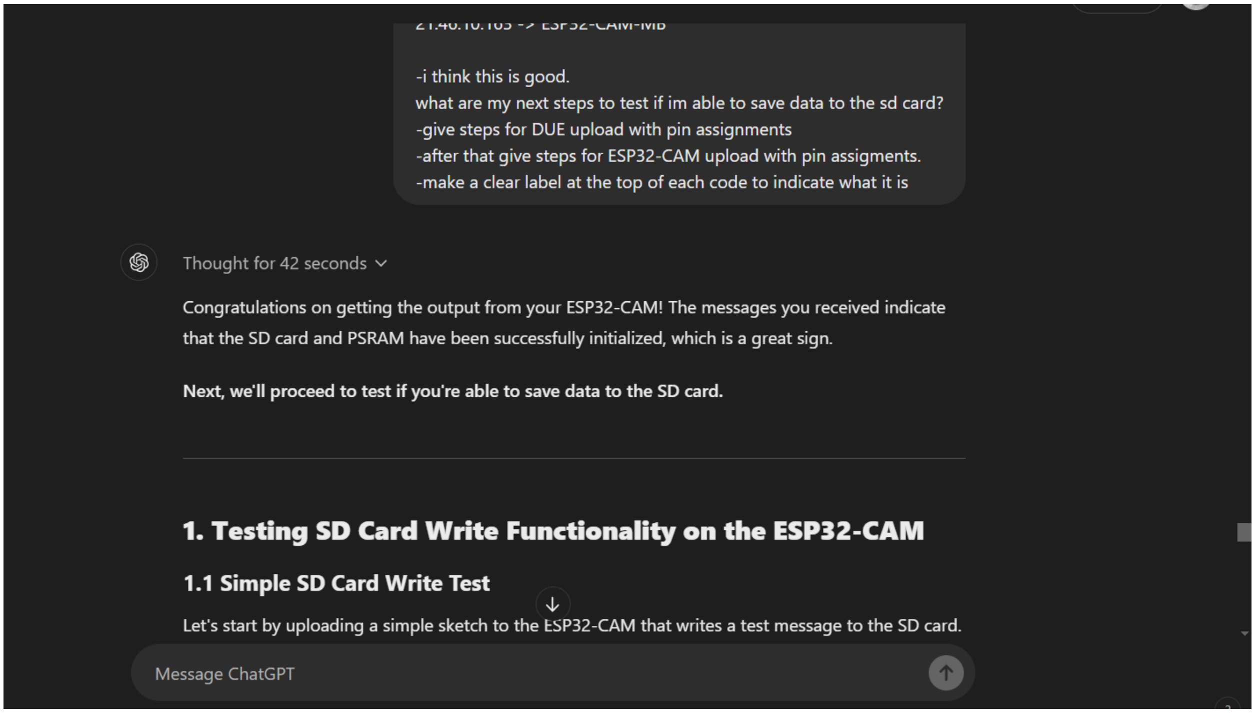


```cpp
void loop() {
  // Pass data from PC to ESP32-CAM
  if (Serial.available()) {
    Serial.println("in serial read: ");
    char c = Serial.read();
    Serial1.write(c);
  }

  // Pass data from ESP32-CAM to PC
```

Message (Enter to send message to 'AI Thinker ESP32-CAM' on 'COM13')

```
□[0;32mI (1007) esp_image: segment 2: paddr=0x001e6db4 vaddr=0x40080000 size=0x00400 (  1024) load□[0m
□[0;32mI (1008) esp_image: segment 3: paddr=0x001e71bc vaddr=0x40080400 size=0x08e54 ( 36436) load□[0m
□[0;32mI (1030) esp_image: segment 4: paddr=0x001f0018 vaddr=0x400d0018 size=0x9df74 (647028) map□[0m
□[0;32mI (1277) esp_image: segment 5: paddr=0x0028df94 vaddr=0x40089254 size=0x0b6a0 ( 46752) load□[0m
□[0;32mI (1309) boot: Loaded app from partition at offset 0x10000□[0m
□[0;32mI (1309) boot: Disabling RNG early entropy source...□[0m
�
SD Size: 30436MB  OK
PSRAM OK
[E][camera.c:1113] camera_probe(): Detected camera not supported.
[E][camera.c:1379] esp_camera_init(): Camera probe failed with error 0x20004
Camera init failed with error 0x20004Entered Bootloader Mode
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
waiting for download
```

Upload error: Failed uploading: uploading error: exit status 2



-i think this is good.
what are my next steps to test if im able to save data to the sd card?
-give steps for DUE upload with pin assignments
-after that give steps for ESP32-CAM upload with pin assigments.
-make a clear label at the top of each code to indicate what it is

Thought for 42 seconds

Congratulations on getting the output from your ESP32-CAM! The messages you received indicate that the SD card and PSRAM have been successfully initialized, which is a great sign.

Next, we'll proceed to test if you're able to save data to the SD card.

---

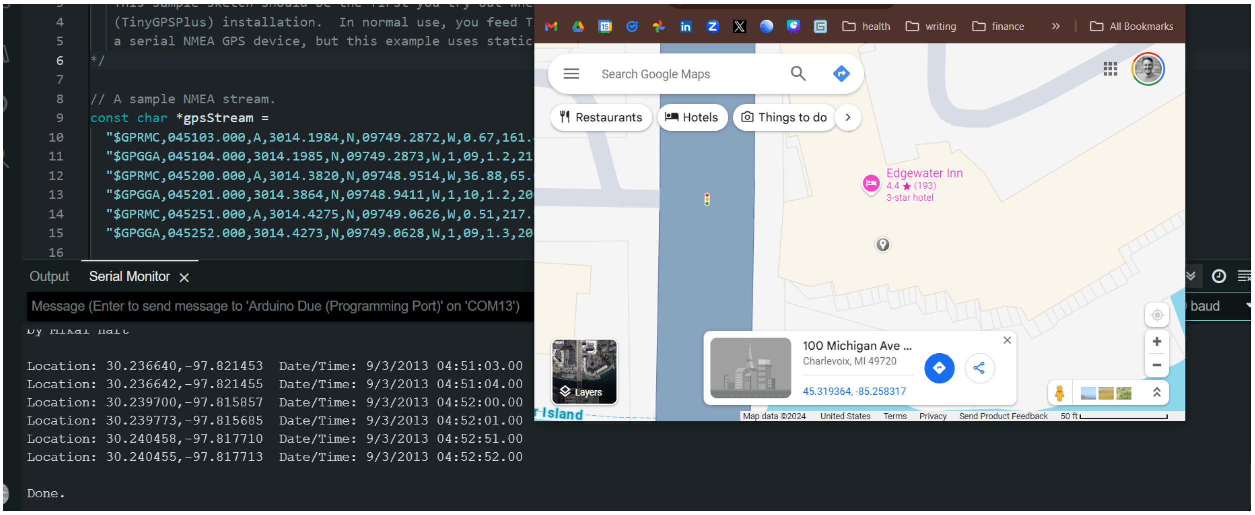
1. Testing SD Card Write Functionality on the ESP32-CAM

1.1 Simple SD Card Write Test

Let's start by uploading a simple sketch to the ESP32-CAM that writes a test message to the SD card.

1:19 AM

9/21/2024

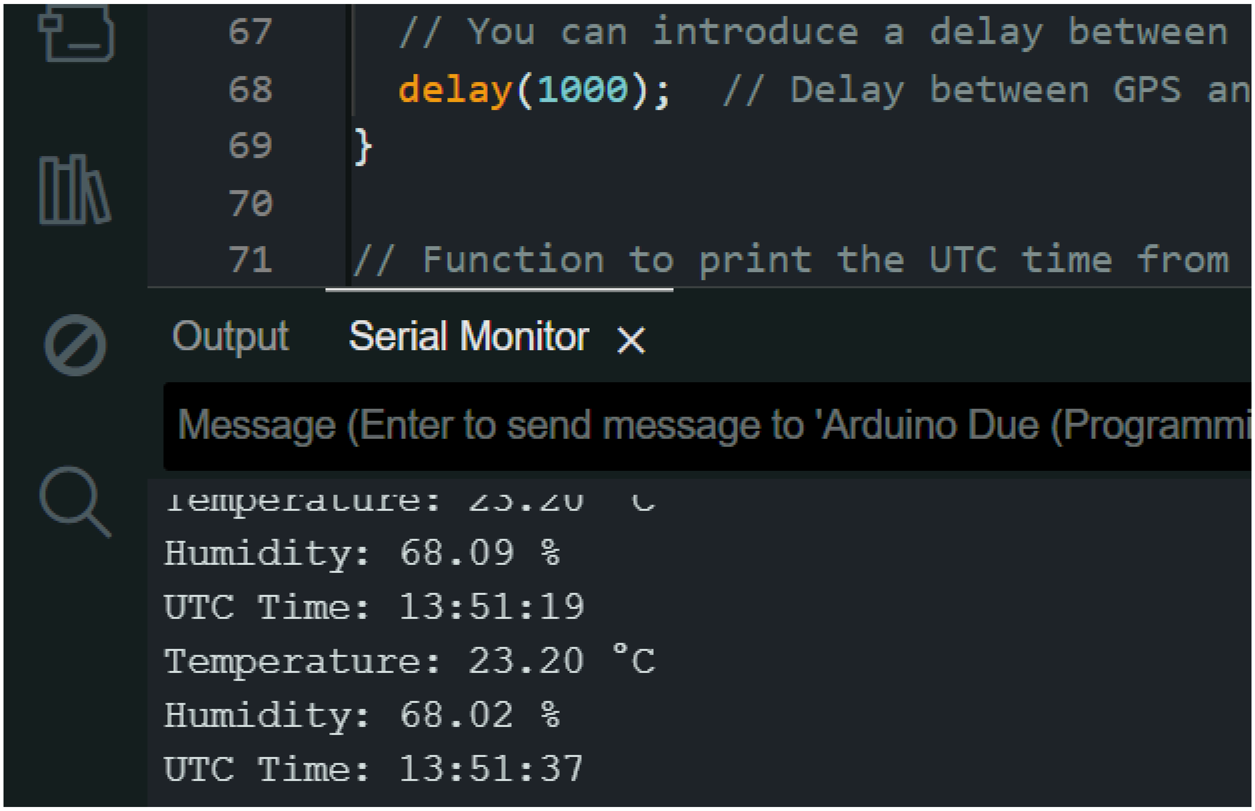


```
  This sample sketch should be the first you try out whe
  (TinyGPSPlus) installation.  In normal use, you feed T
  a serial NMEA GPS device, but this example uses static
*/

// A sample NMEA stream.
const char *gpsStream =
  "$GPRMC,045103.000,A,3014.1984,N,09749.2872,W,0.67,161.
  "$GPGGA,045104.000,3014.1985,N,09749.2873,W,1,09,1.2,21
  "$GPRMC,045200.000,A,3014.3820,N,09748.9514,W,36.88,65.
  "$GPGGA,045201.000,3014.3864,N,09748.9411,W,1,10,1.2,20
  "$GPRMC,045251.000,A,3014.4275,N,09749.0626,W,0.51,217.
  "$GPGGA,045252.000,3014.4273,N,09749.0628,W,1,09,1.3,20
```

```
by Mikal Hart

Location: 30.236640,-97.821453  Date/Time: 9/3/2013 04:51:03.00
Location: 30.236642,-97.821455  Date/Time: 9/3/2013 04:51:04.00
Location: 30.239700,-97.815857  Date/Time: 9/3/2013 04:52:00.00
Location: 30.239773,-97.815685  Date/Time: 9/3/2013 04:52:01.00
Location: 30.240458,-97.817710  Date/Time: 9/3/2013 04:52:51.00
Location: 30.240455,-97.817713  Date/Time: 9/3/2013 04:52:52.00

Done.
```

Got GPS working!! 9/21/2024

.. no i think thats way off. i t just pirinted one example lollll

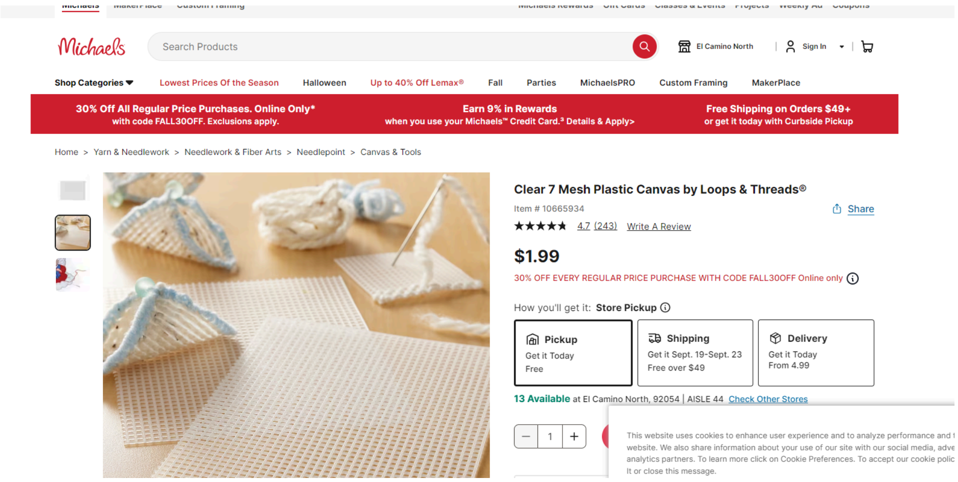


```
  // You can introduce a delay between
  delay(1000);  // Delay between GPS an
}

// Function to print the UTC time from
```

```
Temperature: 23.20 °C
Humidity: 68.09 %
UTC Time: 13:51:19
Temperature: 23.20 °C
Humidity: 68.02 %
UTC Time: 13:51:37
```

Ok now i did it!!! Time for a bagel

Mesh guard for temp sensor



30% Off All Regular Price Purchases. Online Only*
with code FALL30OFF. Exclusions apply.

Earn 9% in Rewards
when you use your Michaels™ Credit Card.³ Details & Apply>

Free Shipping on Orders $49+
or get it today with Curbside Pickup

Home > Yarn & Needlework > Needlework & Fiber Arts > Needlepoint > Canvas & Tools

Clear 7 Mesh Plastic Canvas by Loops & Threads®

Item # 10665934

★★★★★ 4.7 (243)

$1.99

30% OFF EVERY REGULAR PRICE PURCHASE WITH CODE FALL30OFF Online only

How you'll get it: **Store Pickup**

Pickup
Get it Today
Free

Shipping
Get it Sept. 19-Sept. 23
Free over $49

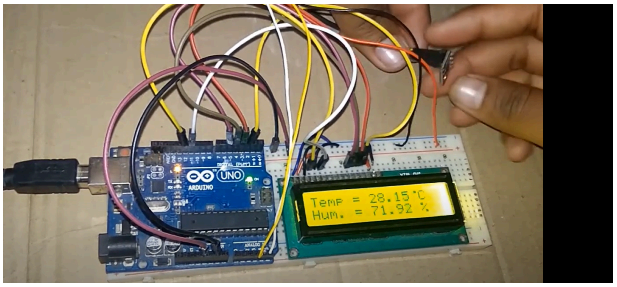
Delivery
Get it Today
From 4.99

13 Available at El Camino North, 92054 | AISLE 44

This website uses cookies to enhance user experience and to analyze performance and t
website. We also share information about your use of our site with our social media, adve
analytics partners. To learn more click on Cookie Preferences. To accept our cookie polic
It or close this message.

Link

Humidity sensor tutorial
Link



Guide to NEO-6M GPS Module with Arduino

This guide shows how to use the NEO-6M GPS module with the Arduino to get GPS data. GPS stands for *Global Positioning System* and can be used to determine position, time, and speed if you're travelling.

You'll learn how to:

- Wire the NEO-6M GPS module to the Arduino UNO

[Guide to NEO-6M GPS Module with Arduino]

[NEO-6 u-blox 6 GPS Modules Data Sheet]