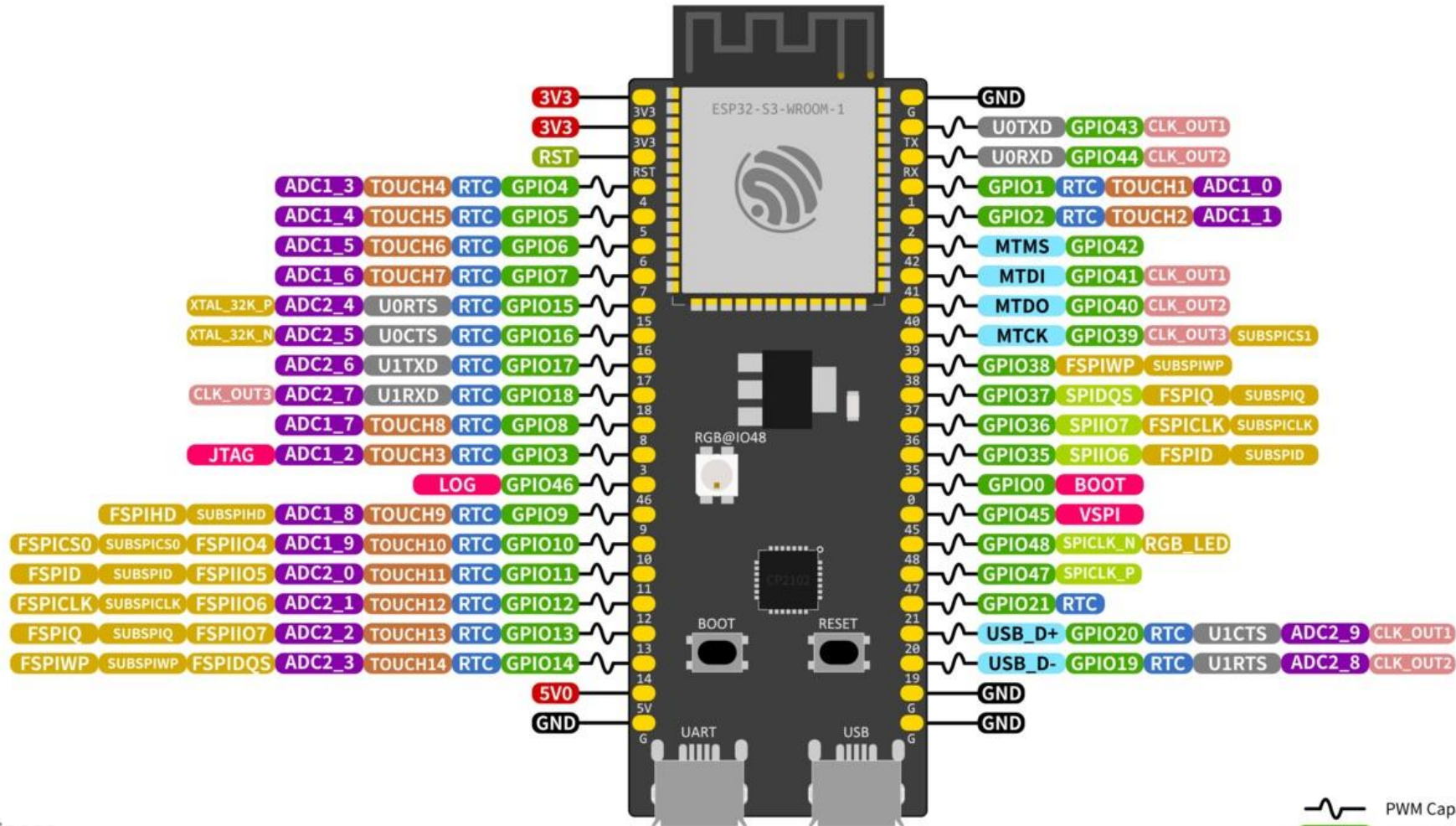


**ChatGPT**

**SKRIV KODE MED AI**

# HARDWARE




# ESP32-S3-DevKitC-1



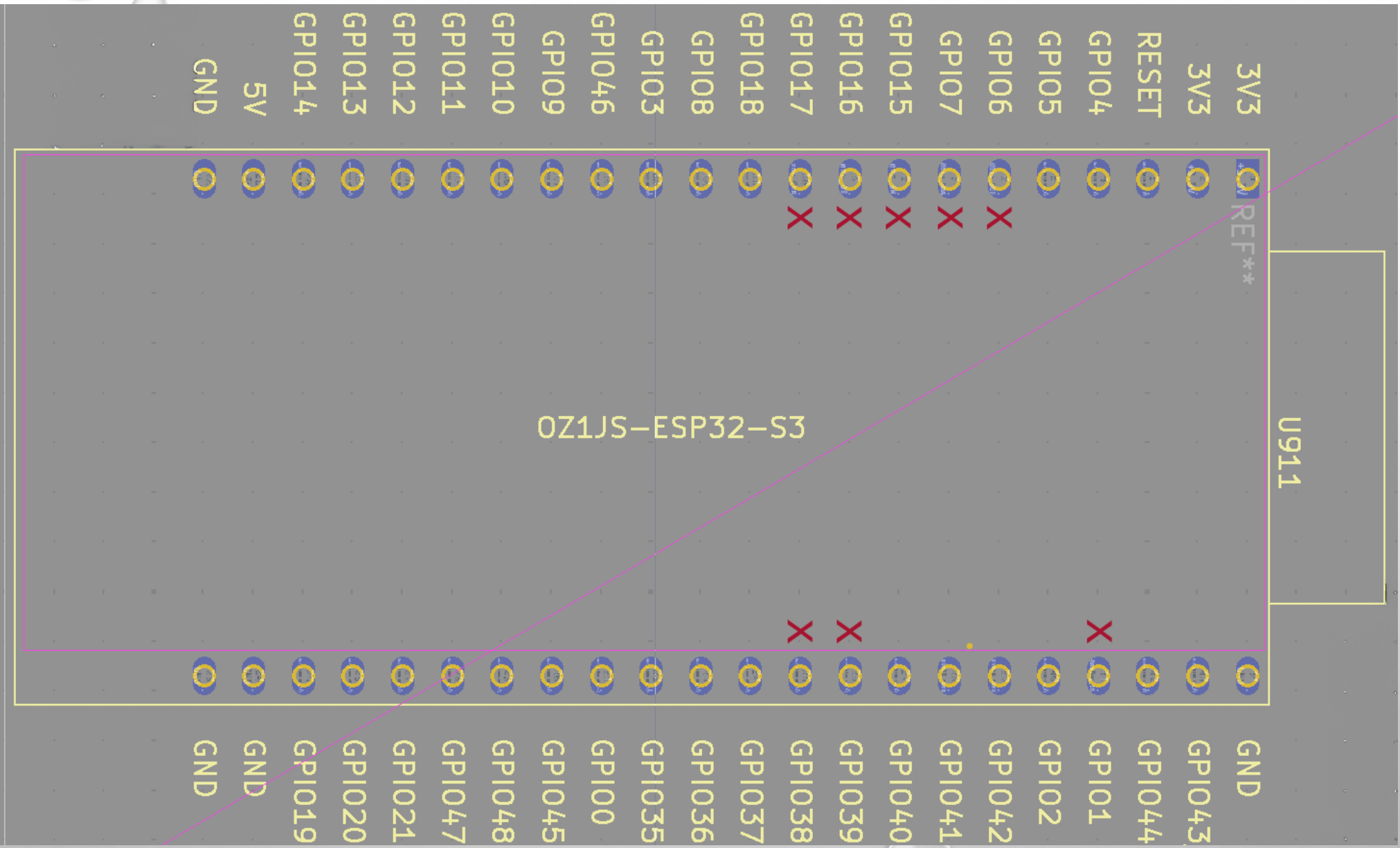
E

## 3 ESP32-S3 Specs

- W 32-bit Xtensa® dual-core @240MHz
- 5 Wi-Fi IEEE 802.11 b/g/n 2.4GHz + BLE 5 Mesh
- 3 512 KB SRAM (16 KB SRAM in RTC)
- 4 384 KB ROM
- 1 45 GPIOs, 4x SPI, 3x UART, 2x I2C,
- T 14x Touch, 2x I2S, RMT, LED PWM, USB-OTG, I2C, 2x 12-bit ADC, 1x LCD interface, DVP

-  PWM Capable Pin
-  GPIOX GPIO Input and Output
-  JTAG/USB JTAG for Debugging and USB
-  ADCX\_CH Analog-to-Digital Converter
-  TOUCHX Touch Sensor Input Channel
-  OTHER Other Related Functions
-  SERIAL Serial for Debug/Programming
-  STRAP Strapping Pin Functions
-  RTC RTC Power Domain (VDD3P3\_RTC)
-  GND Ground

 MISC Miscellaneous/SPI functions



U911

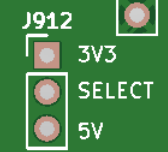
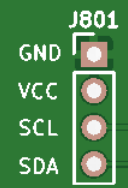
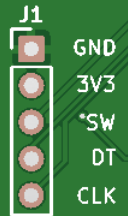
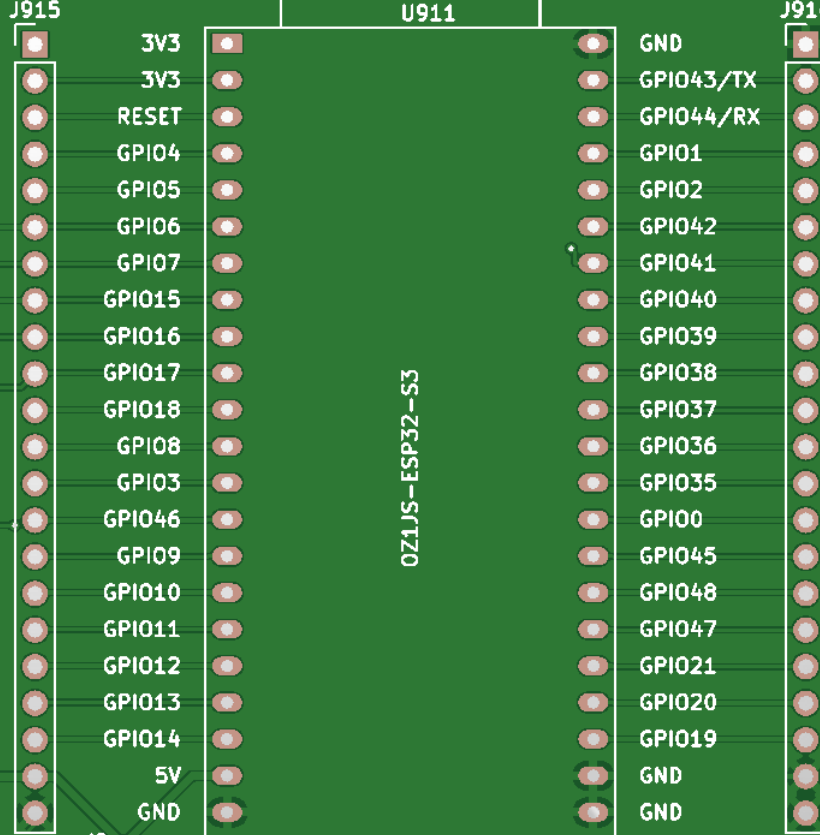
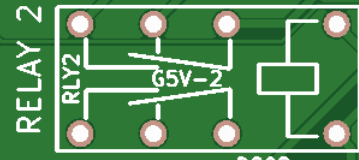
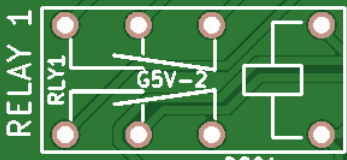
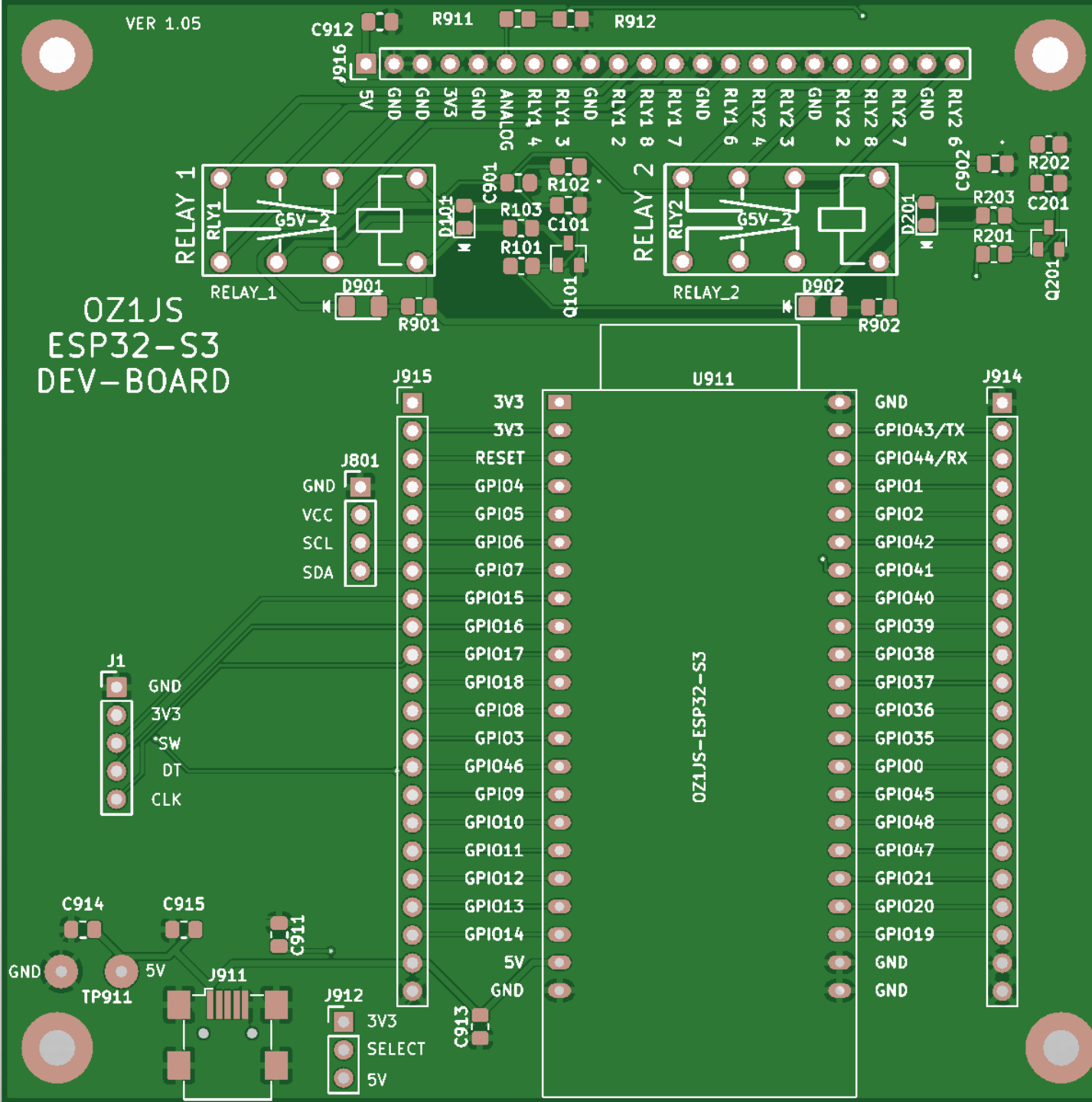
OZ1JS-ESP32-S3

REF\*\*

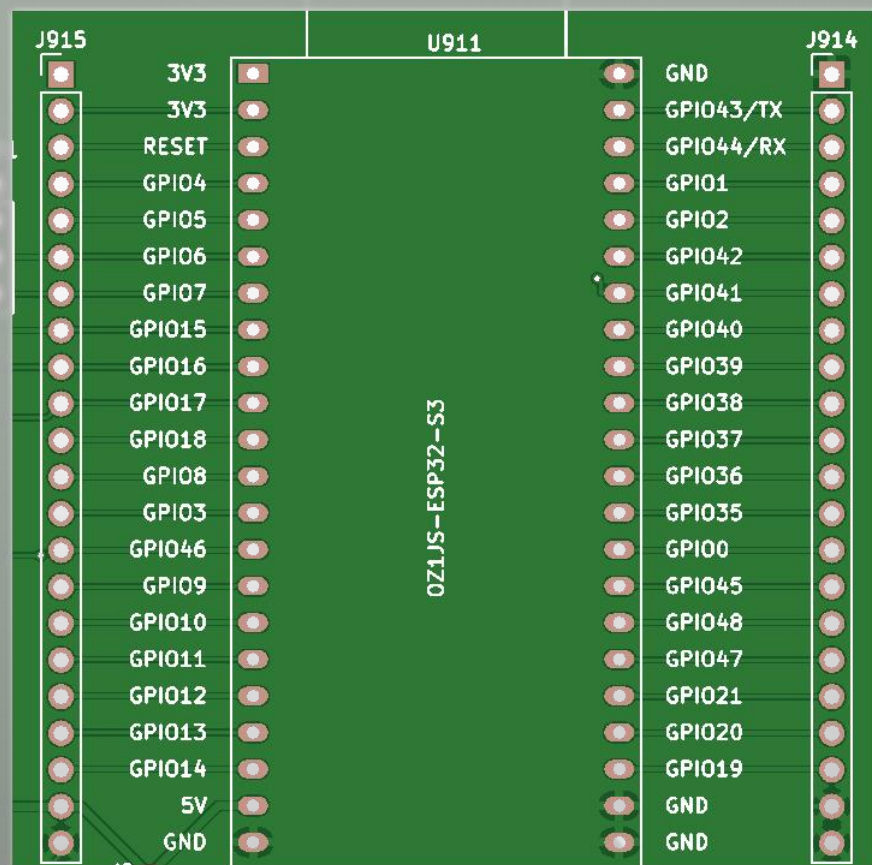


VER 1.05

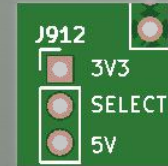
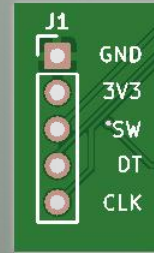
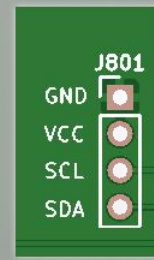
# OZ1JS ESP32-S3 DEV-BOARD



OZ1JS  
ESP32-S3  
DEV-BOARD



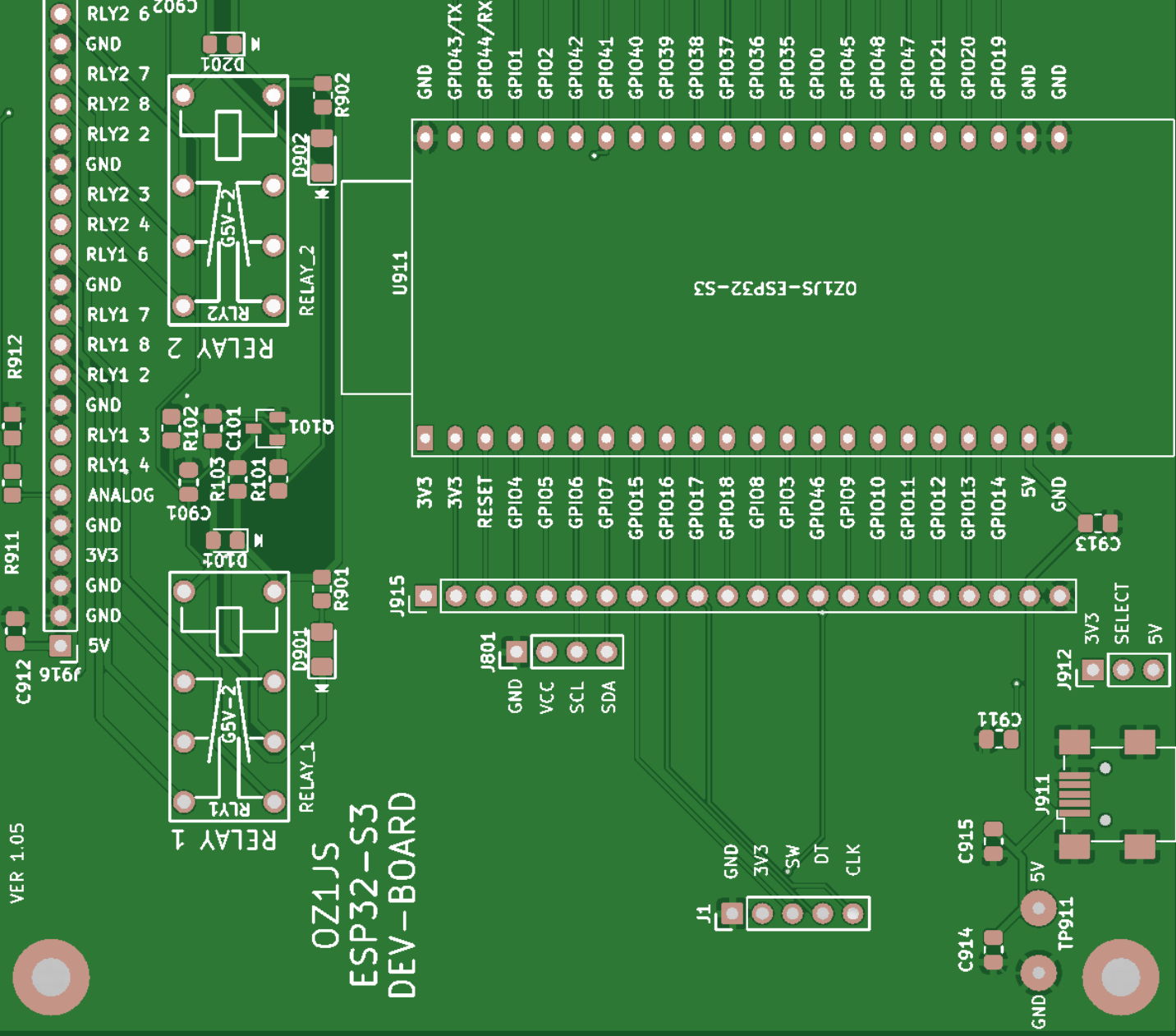
021J5  
ESP32-S3  
DEV-BOARD

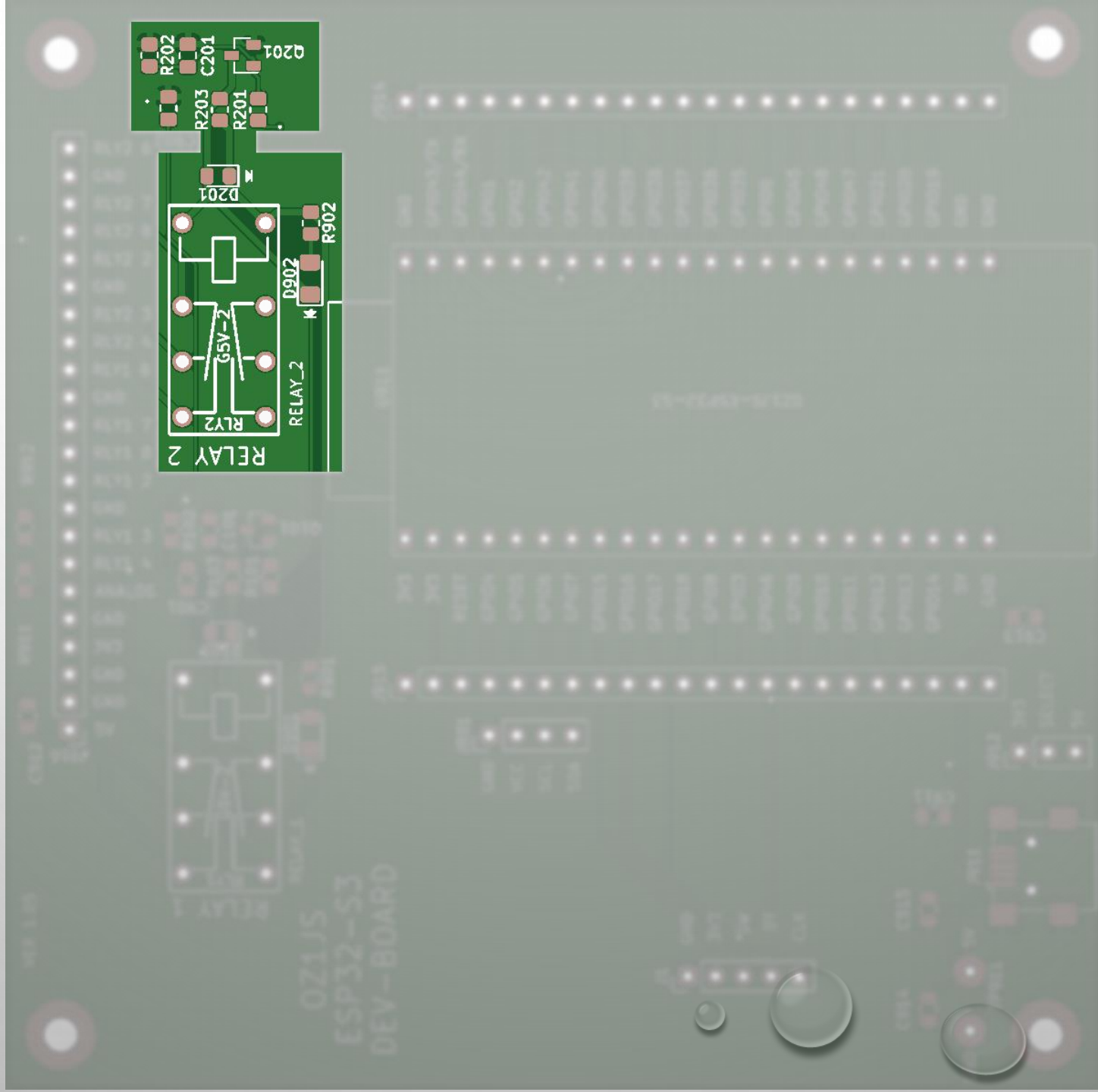




VER 1.05

# OZ1JS ESP32-S3 DEV-BOARD





RELAY 2

RELAY\_2

G5V-2

D902

R902

R201

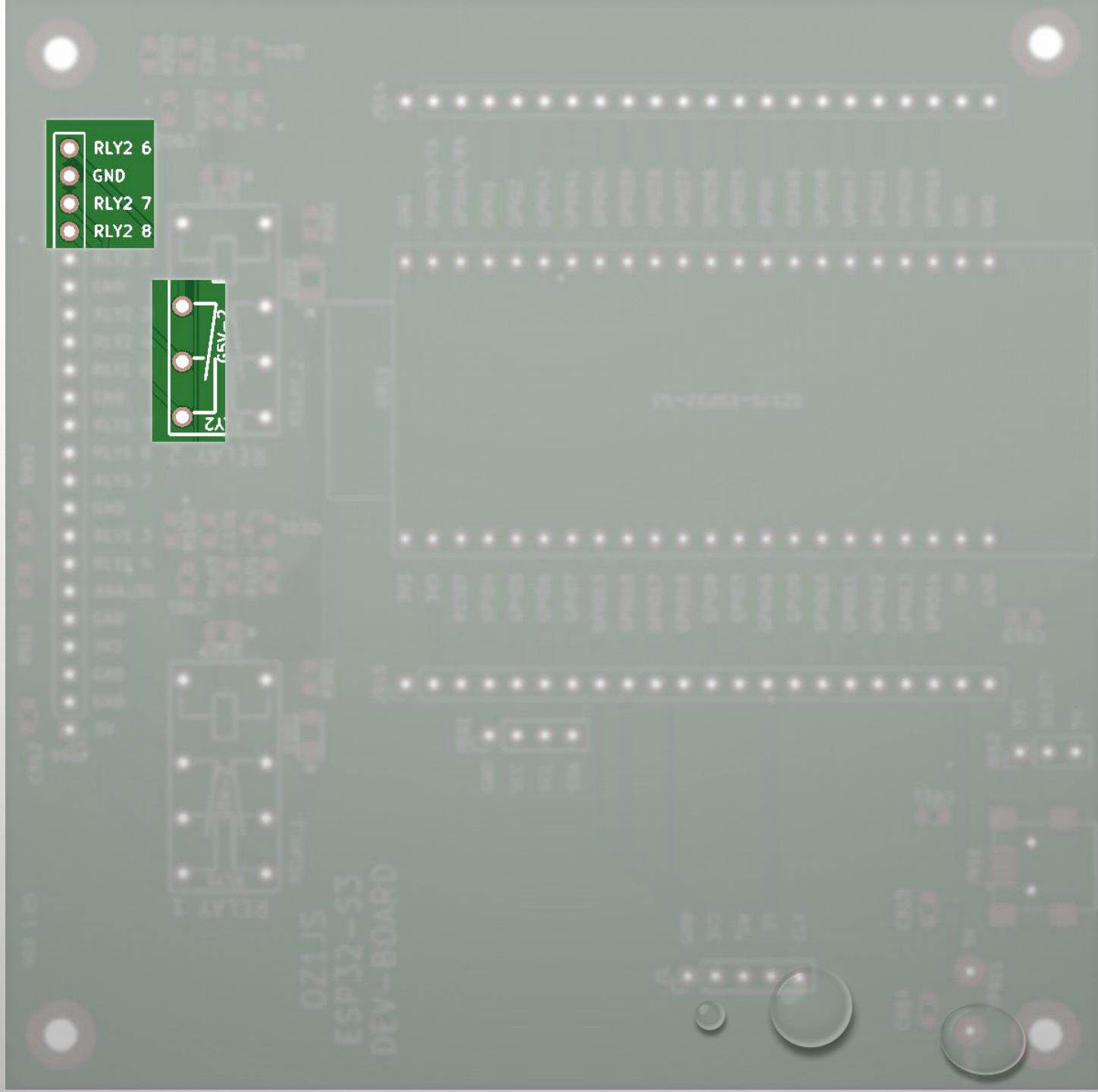
R202

R203

C201

OZ1J5  
ESP32-S3  
DEV-BOARD

- RLY2 6
- GND
- RLY2 7
- RLY2 8

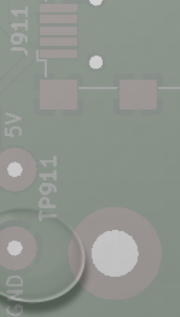
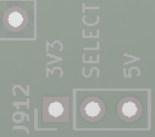
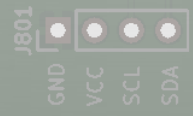
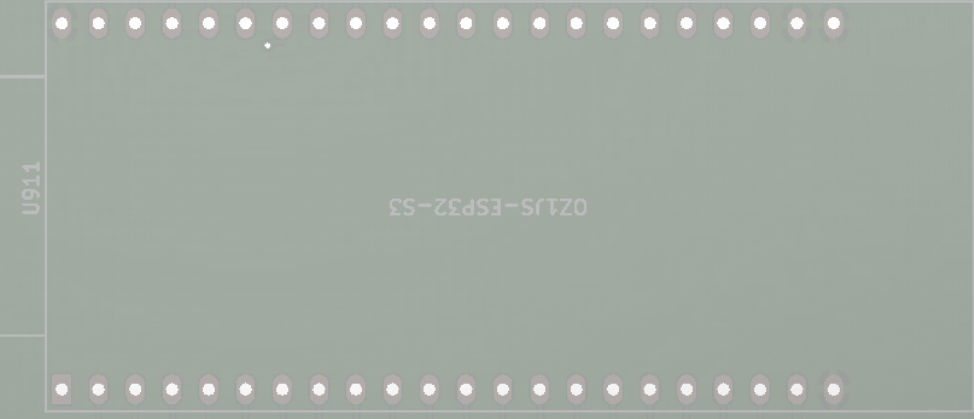
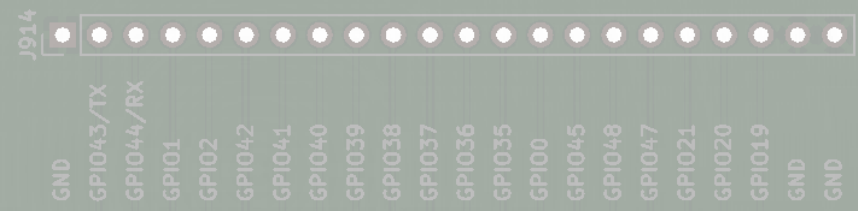
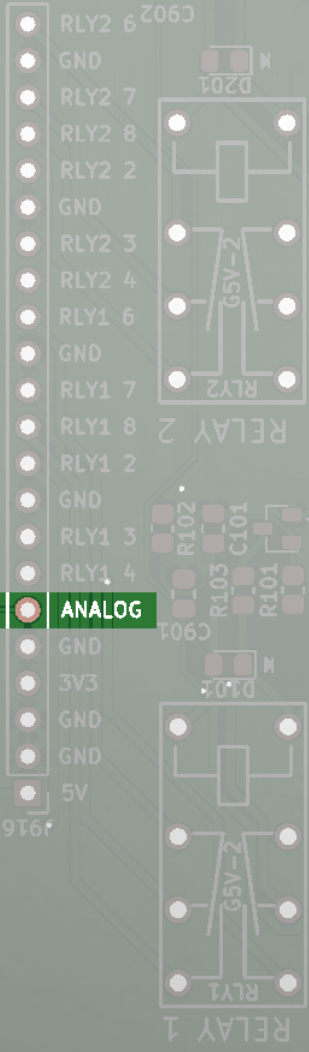


VER 1.05

# OZ1JS ESP32-S3 DEV-BOARD

R911 R912

ANALOG



OZ1JS-ESP32-S3

**3,3 V**

En ting er hardware...

...men vi skal jo ikke glemme, at det er



# ChatGPT

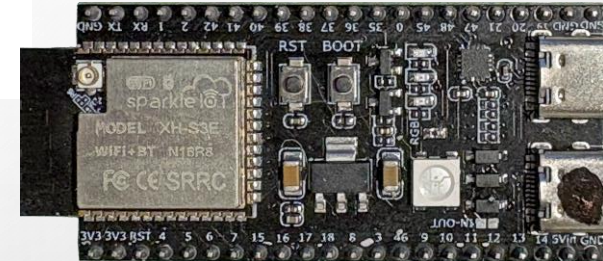
som er omdrejningspunktet for denne workshop.

Sketches kan vi sagtens finde på nettet.

Men med ChatGPT er det muligt at skrive sketches udelukkende på baggrund af egne ideer.



# ChatGPT



Første projekt:

Find den indbyggede RGB-lysdiode



## OZ1JS:

Her er mine grundlæggende oplysninger:

- Jeg arbejder i Arduino IDE version 2.3.7 for Mac.
- Min mikrocontroller er en esp32 S3 (ikke Wroom eller Wrover).
- Lav en sketch der får den indbyggede Neopixel til at skifte mellem grundfarverne.
- Et skift hvert sekund.
- Koden skal være med mange kommentarer, så den let kan forstås af begyndere.



**ChatGPT**

## ChatGPT:

Her er en simpel Arduino-sketch til ESP32-S3, der får den indbyggede NeoPixel (1 LED) til at skifte rød → grøn → blå hvert sekund.

Vigtigt: På mange ESP32-S3 boards kan du bruge `RGB_BUILTIN` (auto). Hvis din ikke virker, så skift `NEOPIXEL_PIN` til den rigtige GPIO (ofte 48 på flere S3 boards).

(NEO\_100.ino)

## NEO\_100.ino

```
1  #include <Adafruit_NeoPixel.h>
2  // ^ Inkluderer Adafruit's bibliotek til NeoPixel (WS2812 / "addressable RGB LED").
3  //   Det giver os funktioner som begin(), setPixelColor(), show() osv.
4
5  // Mange ESP32-S3 board packages definerer RGB_BUILTIN automatisk.
6  // Hvis din ikke gør, så sæt pin manuelt (fx 48 på mange ESP32-S3 boards).
7  #ifndef RGB_BUILTIN
8  |   #define RGB_BUILTIN 48    // <-- prøv 48 først, ellers ret til din boards NeoPixel GPIO
9  #endif
10 // ^ #ifndef betyder: "hvis RGB_BUILTIN ikke allerede er defineret"
11 //   Så definerer vi den her.
12 //   På nogle boards findes RGB_BUILTIN allerede i board-definitionen,
13 //   så slipper du for selv at gætte pin.
14
15 // Vi bruger constexpr for at lave "konstanter" (fast værdi) som ikke ændrer sig.
16 constexpr uint8_t NEOPIXEL_PIN    = RGB_BUILTIN;
17 // ^ NEOPIXEL_PIN er den GPIO/pin hvor den indbyggede NeoPixel sidder.
18
```



ChatGPT

OZ1JS:

Koden kompilerer, upload OK, men sketchen går tilsyneladende ikke i gang.



**ChatGPT**

## ChatGPT:

Det kan meget vel køre helt fint, men du ser det ikke – typisk fordi NeoPixel-pin (eller power-enable) ikke er den, vi gætter på. Når pin er forkert, sker der bare... ingenting (ingen fejl, ingen crash), så det ligner at sketchen ikke starter.

Her er den hurtigste måde at afgøre det på, uden at vi skal kende dit præcise board:

1) Først: bevis at sketchen kører (Serial + “flash”)

Upload denne og åbn Serial Monitor (115200 baud). Hvis du ser tekst, kører programmet.

(NEO\_101\_alive.ino)



**ChatGPT**

## ChatGPT:

2) Pin-finder: prøv de mest almindelige NeoPixel-pins automatisk

Denne sketch prøver en række typiske ESP32-S3 NeoPixel-pins (48, 47, 21, 38 osv.). Når du ser LED'en lyse, kigger du i Serial Monitor og ser hvilken pin den er på.

(NEO\_102\_Search.ino)



ChatGPT

OZ1JS:

Den fandt NEOPIXEL på pin 48. Nu kører det. Det ser ud som om RGB\_BUILTIN er defineret, men til en anden pin.

```
#ifndef RGB_BUILTIN
  #define RGB_BUILTIN 48 // midlertidigt gæt
#endif
```



**ChatGPT**

**ChatGPT:**

(NEO\_103\_Basis)

Bonus: Vis på Serial hvad core'en tror

(NEO\_104\_DefaultNEO)



ChatGPT

OZ1JS:

OK. Nu fandt vi NEOPIXEL på pin 48. Kan du lave en sketch der viser nogle flotte farveforløb ved at udnytte mange af de  $256^3$  muligheder



**ChatGPT**

## ChatGPT:

Ja — her er en “flot farveforløb”-sketch til ESP32-S3 med NeoPixel på GPIO48. Den bruger HSV→RGB og kører et glidende rainbow + shimmer, så den udnytter rigtig mange af de ~16,7 mio. farver ( $256^3$ ) over tid.

(NEO\_105\_Multicolor.ino)



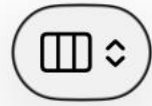
**ChatGPT**

## ChatGPT:

Sådan “tuner” du effekten hurtigt

- Mere lys: øg valBase (fx 120).
- Hurtigere farveskift: øg hueBase += 120; (fx 300).
- Mere rolig: sænk det (fx 40) og/eller øg frame-intervallet fra 16ms til 25ms.

Hvis du vil have flere forskellige “modes” (fx “aurora”, “sunset”, “pastel”, “neon”), kan jeg lave en version med et lille menupunkt (fx via Serial eller knap/encoder) der skifter mellem farveprofiler.



Oversigt



Gruppe



Del



Tilføj mærker



Handling



Dropbox



Søg

 \_AI\_ESP32\_Basic  >

 **NEOPIXEL**  >

 NEO\_100  >

 NEO\_101\_Alive  >

 NEO\_102\_Search  >

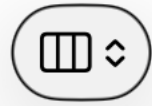
 NEO\_103\_Basis  >

 NEO\_104\_DefaultNEO  >

 NEO\_105\_Multicolor  >

 NEO\_106\_Sun  >

DZ6HR >  AI og kodning >  AI Sketches >  NEOPIXEL



Oversigt



Gruppe



Del



Tilføj mærker




Handling






Dropbox



Søg

 **\_AI\_ESP32\_Basic**  >

 **NEOPIXEL**  >

 **NEO\_103\_Basis**  >

DZ6HR >  AI og kodning >  AI Sketches >  \_AI\_ESP32\_Basic



**ChatGPT**

ChatGPT er en maskine.

Den aner ikke, hvad den laver.

Den arbejder med en enorm vidensbank og vælger sit svar på basis af statistisk sandsynlighed ud fra dens fortolkning af dit input. Desto flere konkrete data du leverer, desto bedre løsning vælger ChatGPT.

I nogen grad kan du regne med, at den "husker" detaljer fra jeres nuværende chat.

Men det skader ikke, at du gentager fakta om dit projekt fra tid til anden.



## Hvilke pins er bedst at benytte til I<sup>2</sup>C.

- De valgte pins skal helst være naboer
- De må ikke kollidere med boot eller andre af ESP32's faste rutiner
- Jeg benytter WiFi og Bluetooth
- Jeg benytter ESP32 S3 (ikke Wroom eller Wrover)

**““ UNDGÅ (TYPISK) DISSE:**

**⚠ GPIO6 + GPIO7**

**KAN KOLLIDERE MED INTERN FLASH/PSRAM PÅ MANGE BOARDS → USTABILITET ELLER BOOT-PROBLEMER.**

**”” FORSLAG DER TYPISK ER “UDEN BØVL” PÅ DIT BOARD  
I<sup>2</sup>C (NABOER)**

- SDA = GPIO6**
- SCL = GPIO7**

**BEGRUNDELSE: 6/7 LIGGER SOM NABOER OG KOLLIDERER  
NORMALT IKKE MED BOOT/USB.**

“

OZ1JS:

DU SKREV:

Undgå (typisk) disse:

⚠ GPIO6 + GPIO7 Kan kollidere med intern flash/PSRAM på mange boards → ustabilitet eller boot-problemer.

”

men senere foreslår du netop de to til mit oled-display.

“**CHATGPT:**

Du har ret: det var inkonsistent.

Korrektionen er:

**ESP32 vs ESP32-S3: GPIO6/7**

På klassisk **ESP32 (WROOM/WROVER)** er **GPIO6–11** næsten altid reserveret til **SPI flash** → dem bør man undgå.

På **ESP32-S3** er flash/PSRAM typisk lagt på **andre GPIO'er** (fx 26–32 og ved Octal også 33–37), så **GPIO6/7** er ofte frie og kan bruges. Derfor kan de være et fint valg *på S3*.

Min tidligere advarsel om 6/7 var altså den “gamle ESP32-regel” og gælder ikke generelt for ESP32-S3.

”