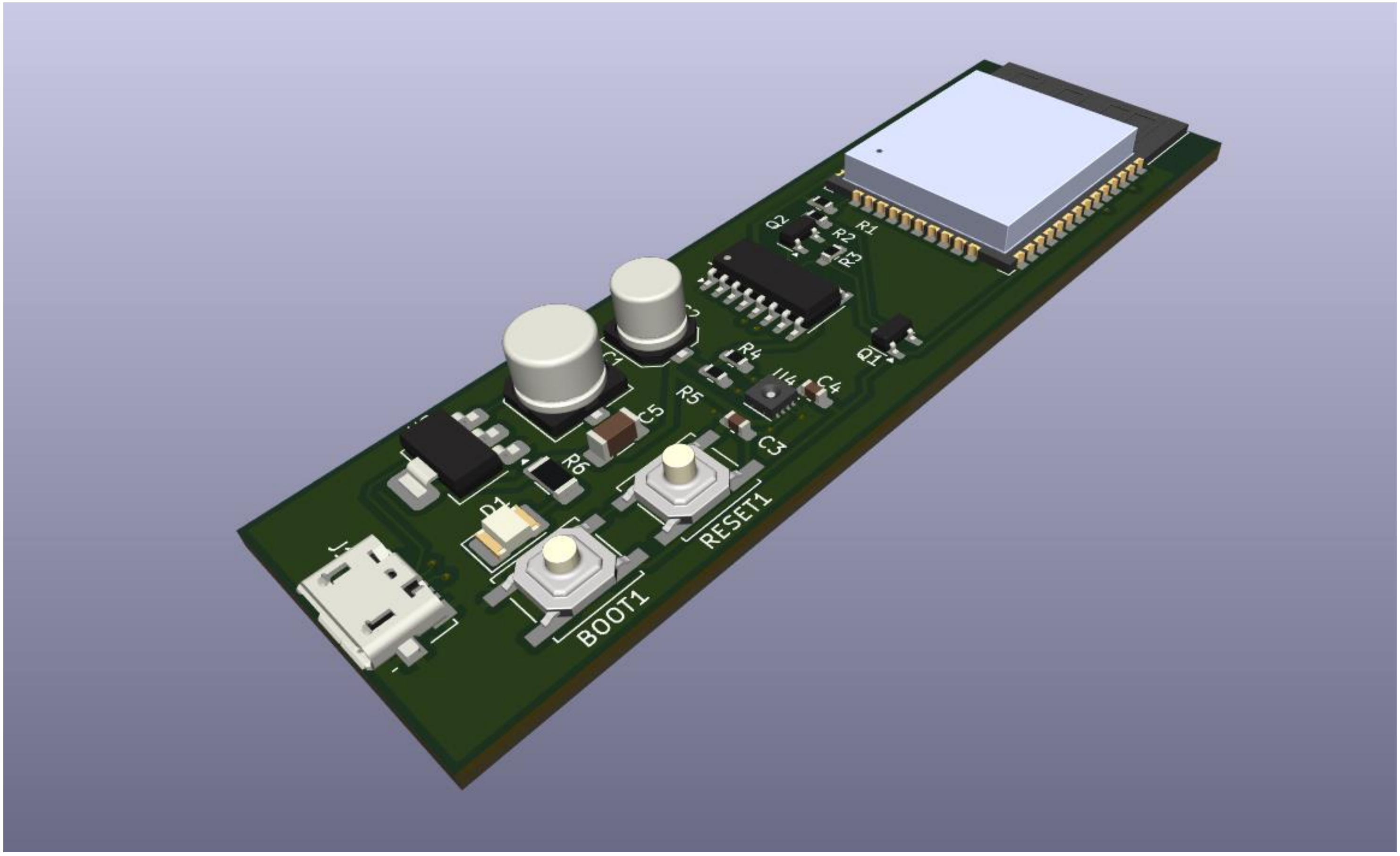


Objective:

Design a schematic and PCB based on the provided details: The PCB should have an ESP32 Chip (not development board, see reference) that is connected to a temperature and humidity sensor (HTU21D / SHT30 or a similar sensor of your choice) and also a USB port (You can use a UART-USB bridge IC of your choice). Design for USB power input with basic power regulation (e.g., 5V to 3.3V using an LDO regulator or switching regulator). Targeting a compact PCB size would be a plus

Steps Involved:

1. Create Schematic
2. Create a Blank PCB Layout
3. Foot print Assignment
4. Electrical Rules Check
5. Designing Your PCB Stackup
6. Design Rules Check
7. Place Components
8. Route Traces
9. Generate Design Files

Software Used : KICAD

Components Used:

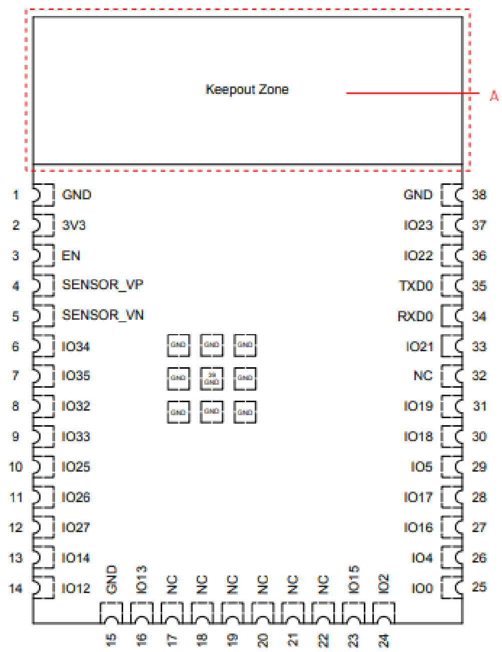
Component	Model	Quantity
ESP32 Chip	ESP32-WROOM-32E-N4	1
USB to UART interface	CH340G	1
LDO	AMS1117	1
Temperature and Humidity sensor	SHT30-DIS	1
BJT	BC547	2
switches	SMD Push Button	2
Resistors	10k,1k	5, 1
LED	Red SMD	1
Capacitors	47uF, 22UF , 0.1uF, 100nF	1, 1, 2, 1

ESP32-WROOM-32E-N4

ESP32-WROOM-32E are two powerful, generic Wi-Fi + Bluetooth + Bluetooth LE MCU modules that target a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding. ESP32-WROOM-32E comes with a PCB antenna, with a connector for an external antenna. The information in this datasheet is applicable to both modules.

Pin Layout :

The pin diagram below shows the approximate location of pins on the module. For the actual diagram drawn to scale, please refer to Figure



To design a circuit integrating the **ESP32-WROOM-32E** module with the **CH340G** USB-to-serial converter, **SHT30**, and **boot** and **reset** buttons, here's overview of each component's role and connections:

1. Power Supply

The **ESP32-WROOM-32E** operates at 3.3V, so it requires a stable 3.3V supply. The CH340G usually works with 5V from the USB, and an **LDO regulator** converts this 5V down to 3.3V for the ESP32.

Decoupling Capacitors: Place capacitors (47 µF and 22 µF) near the power pins of both the ESP32 and the CH340G to filter out noise and ensure stable operation.

2. USB-to-Serial Converter (CH340G)

USB Data Lines: The **D+** and **D-** pins from the CH340G are connected to the USB port's D+ and D- lines, enabling data transfer between the computer and the ESP32 via serial.

TXD and RXD Pins: The CH340G's **TXD** (Transmit Data) pin connects to the ESP32's **RX0** (GPIO3) pin, and the **RXD** (Receive Data) pin connects to the ESP32's **TX0** (GPIO1) pin. This setup allows serial communication between the ESP32 and the computer for programming and debugging.

3. Boot and Reset Buttons

Boot Button: This button connects to **GPIO0** of the ESP32 and is pulled up to 3.3V. Pressing this button pulls GPIO0 to ground, putting the ESP32 into bootloader mode for flashing new firmware.

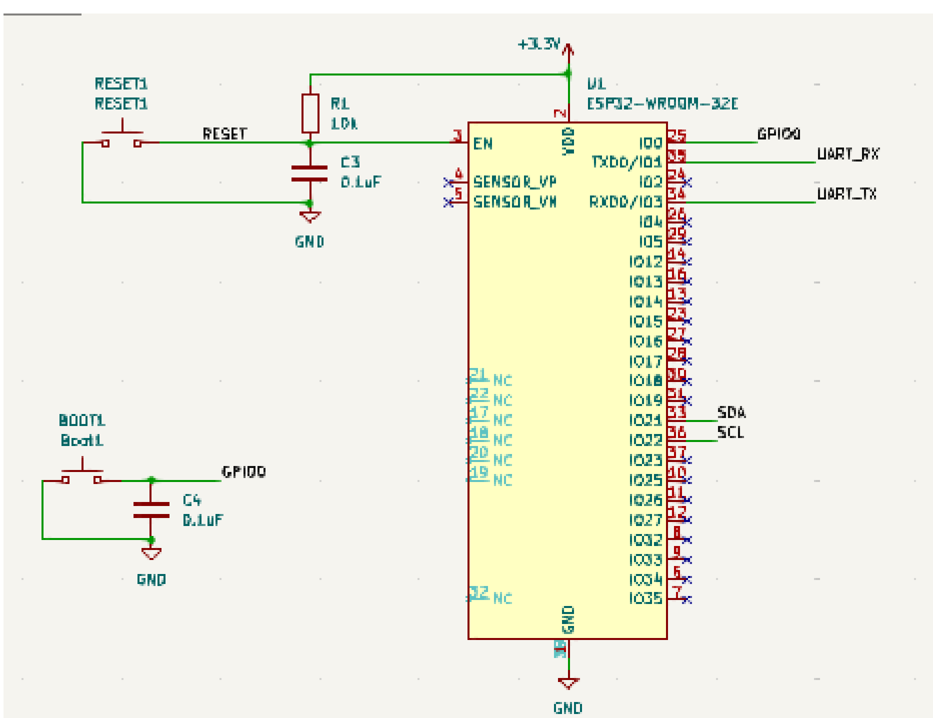
Reset Button: This button connects to the **EN (Enable)** pin of the ESP32 and is normally pulled up to 3.3V. Pressing the reset button pulls EN to ground, resetting the ESP32.

4. I2C Interface (SDA and SCL)

- **SDA (Data Line)**: Connected to **GPIO21**.
- **SCL (Clock Line)**: Connected to **GPIO22**.
- Both **SDA** and **SCL** lines have **pull-up resistors** (typically 10kΩ each) to 3.3V, required for proper I2C communication.

6. Ground Connections

Ensure all **GND pins** from the ESP32, CH340G, and other peripherals are connected to a common ground to maintain a stable reference and prevent signal noise issues.



CH340G USB To Serial TTL Converter IC – SOP-16 Package

CH340 is a USB bus conversion chip, it can realize USB to UART interface or USB to printer interface. In serial UART mode, CH340 provides a common MODEM liaison signal, used to expand the UART interface of the computer or upgrade the common serial device to a USB bus directly. This Product is known as CH340G, CH340G IC SMD SOP 16 Package USB to Serial TTL Converter IC, Component, Driver and Interface, Electronic Components, IC, Integrated Circuits, USB to Serial TTL Converter IC, CH340G (SMD SOP-16 Package) USB to Serial TTL Converter IC, CH340G USB to Serial Converter Chip.

Now let’s look into SHT-30 circuit in the design.

1. Micro USB Port:

The circuit starts with a **Micro USB port**, which receives power and data from the computer. The Micro USB has 5 pins:

- **VCC (Pin 1)**: Provides +5V from the USB port.
- **D+ (Pin 2)** and **D- (Pin 3)**: Data lines for USB communication.
- **ID (Pin 4)**: Used for identifying the USB type but often not connected.
- **GND (Pin 5)**: Ground.

2. Power Supply and Bypass Capacitors:

The VCC pin from the USB is connected to the CH340G’s **VCC** and **GND** pins to power the chip. A **bypass capacitor** (e.g., 0.1 µF) is placed close to these pins to stabilize the power supply and filter out noise.

3. CH340G USB Data Lines:

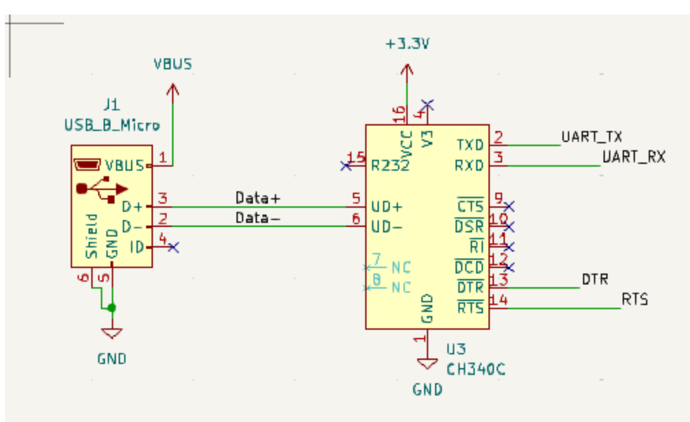
The **D+** and **D-** lines from the Micro USB port connect directly to the **D+** and **D-** pins of the CH340G, allowing data transmission between the computer and CH340G.

5. TXD and RXD Pins (Serial Data):

The CH340G has **TXD** (Transmit Data) and **RXD** (Receive Data) pins for serial communication with a microcontroller or other serial device.TXD sends data from the CH340G to the microcontroller, while RXD receives data from the microcontroller to send to the computer.

6. DTR and RTS (Optional Flow Control):

DTR (Data Terminal Ready) and **RTS (Request to Send)** lines, which can control data flow, ensuring smooth communication between devices.



AMS1117 LDO Voltage Adjustable Regulator Circuit

Its key feature is that it is a low-drop regulator, which means that it reaches its maximum efficiency with an input voltage that only needs to be a little bit higher than the output voltage, or around 0.9V to 1.2V.
For example, if the load requires 3.3V, we can just supply it with an input voltage of around 4.5V to 5V. It can also supply a current of up to 1A, despite being small and economical.
The output accuracy is also at 1.5%, similar to the 78xx regulator IC series.

AMS1117 Pinout

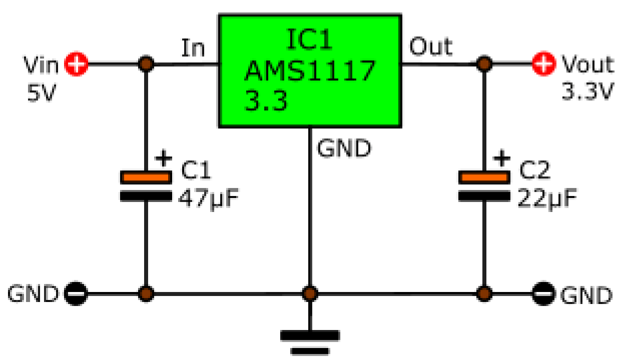
we are going to be using the SOT-223 because of its small size and suitability for use in a small circuit that requires a current lower than 500mA.

AMS1117 Fixed Voltage Regulator Circuit :

Now let’s look into LDO circuit in the design.

5V to 3.3V Regulator Circuit :

The information provided by ESP, the manufacturer of the ESP8266, recommends having 3.0-3.6V of excess voltage. But since the power source we have only supplies 5V, it would not work. If we were to use the LM317L, it would not work either, because it requires up to 5.8V.



5V to 3.3V Regulator circuit using AMS1117-3.3

Instead, we chose the AMS1117-3.3. The fact that it is a **low-dropout regulator** (**LDO regulator**) allows it to only take 1V to run, or 1.2V in case you need the full 1A current.C1 and C2 help filter the current to reduce the ripple voltage. The capacitance of C2 should be around 10µF to 22µF; anything more than this would not help.

Temperature and Humidity sensor SHT30-DIS :

SHT3x-DIS is the next generation of Sensirion’s temperature and humidity sensors. It builds on a new CMOSens sensor chip that is at the heart of Sensirion’s new humidity and temperature platform. The SHT3x-DIS has increased intelligence, reliability and improved accuracy specifications compared to its predecessor.

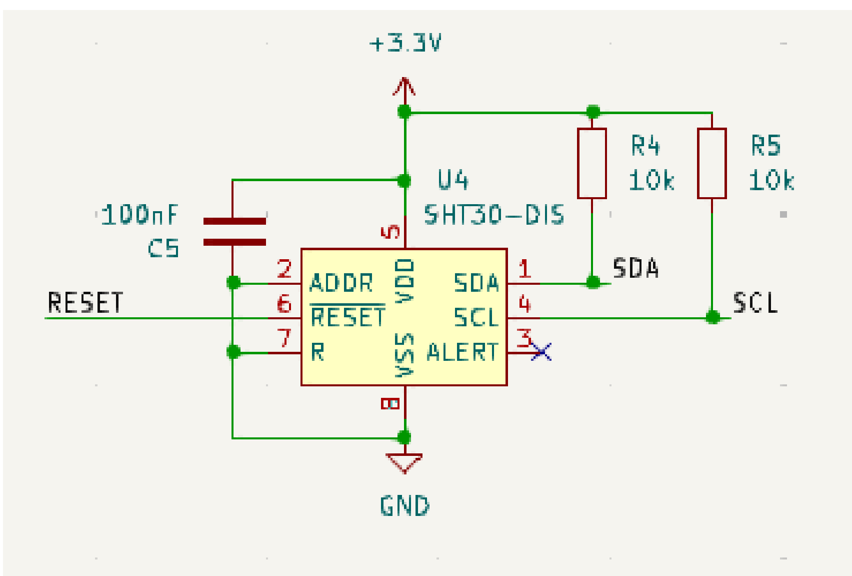
Key features

- Fully calibrated, linearized, and temperature compensated digital output
- Wide supply voltage range, from 2.15 V to 5.5 V
- I2C Interface with communication speeds up to 1
- Very fast start-up and measurement time
- Tiny 8-Pin DFN package
- High signal-to-noise ratio

SHT30 circuit:

Now let’s look into SHT-30 circuit in the design.

- Connect VCC and GND pins to the 3.3V and ground rails.
- Connect the I2C SDA and SCL pins to ESP32 GPIO 21 and GPIO 22, respectively, with pull-up resistors (10 kΩ each) to 3.3V.



Flow control circuit:

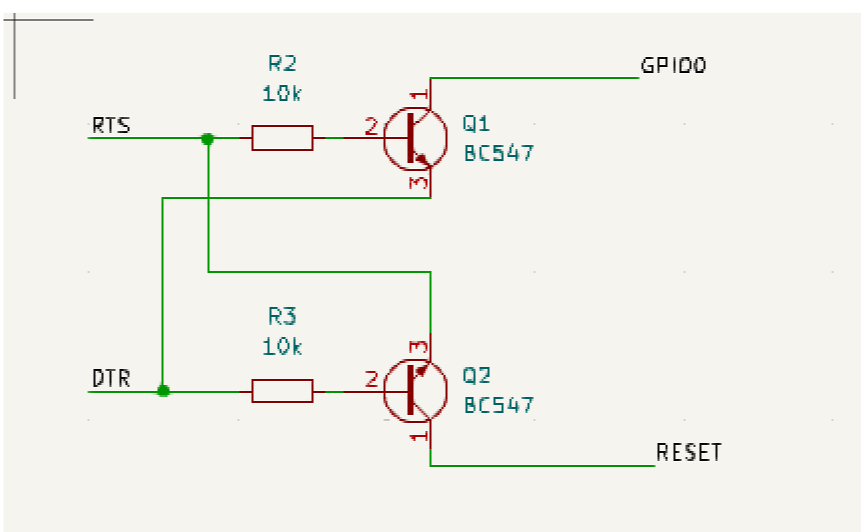
The flow control circuit using a **BC547** NPN transistor with the **ESP32-WROOM-32E** is often implemented to manage the **RTS (Request to Send)** and **DTR (Data Terminal Ready)** lines. This setup automates the flashing and reset process for the ESP32, allowing the CH340G USB-to-serial converter to control the ESP32 without manual intervention. Here’s how it works:

- **BC547** NPN transistor is used here as a switch, triggered by signals from the CH340G’s **RTS** and **DTR** lines. These signals allow the ESP32 to enter programming mode and then reset automatically after programming.
- The RTS and DTR signals are controlled by the USB-to-serial adapter on the CH340G and are used to control the ESP32's **GPIO0** and **EN** (Enable) pins.
- **RTS Line**: The RTS signal is connected to the base of one BC547 transistor via a **current-limiting resistor** 10kΩ. This transistor’s collector is connected to the **EN** pin on the ESP32, and the emitter is grounded.
- **DTR Line**: Similarly, the DTR signal is connected to the base of a second BC547 transistor with a current-limiting resistor. The collector of this transistor is connected to **GPIO0** of the ESP32, while the emitter is grounded.
- When the RTS and DTR lines are toggled in a specific sequence (usually by the programming software), the transistors control the **EN** and **GPIO0** pins.
- For flashing:

  EN is briefly pulled low by the RTS transistor, which resets the ESP32.

  GPIO0 is simultaneously pulled low by the DTR transistor, setting the ESP32 in **bootloader mode**.

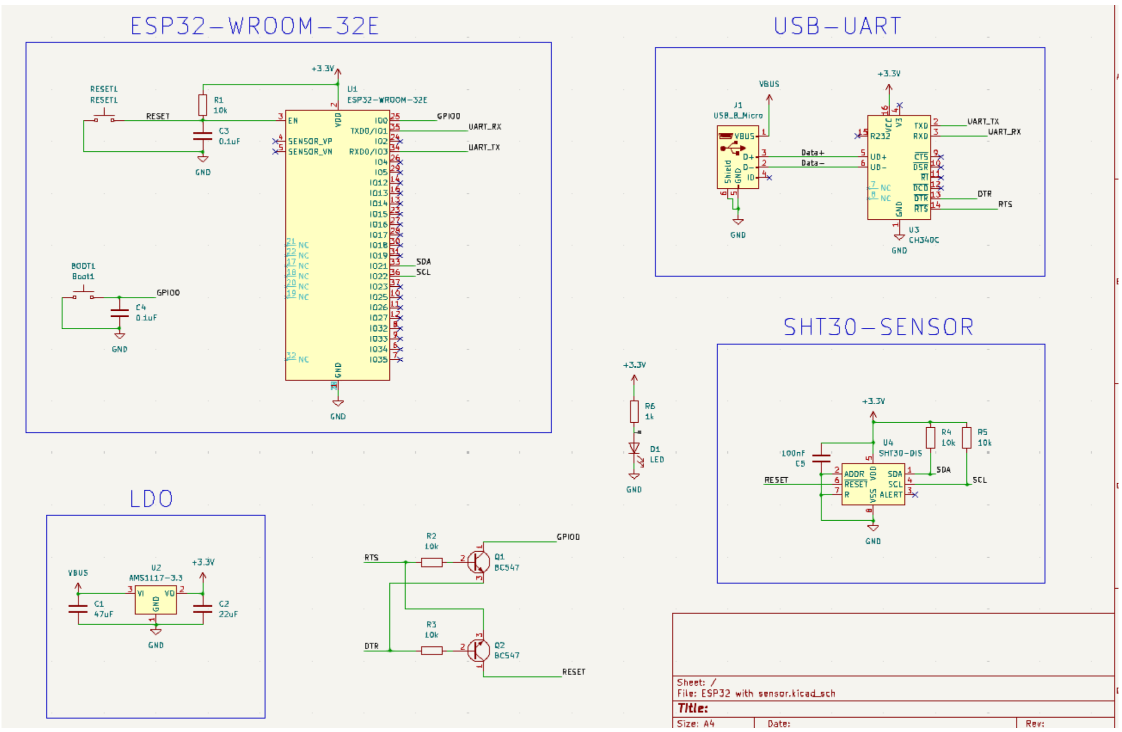
- This configuration allows the CH340G to automatically reset and put the ESP32 into programming mode.



Printed Circuit Board Design

1. Create Schematic

- Open **KiCAD** and create a new project. Start with **Eeschema** to design the circuit schematic.
- Place symbols for all the components you need from the library and wire them to complete the circuit.
- Annotate the schematic (assign reference designators like R1, C1).



2. Electrical Rules Check

- Perform another **ERC** on the schematic to ensure there are no connectivity issues or unconnected pins.

3. Footprint Assignment

- Assign footprints to each component in **Eeschema** or in PCBNew. This links each component symbol to a physical footprint on the board.
- Ensure each component has the correct footprint to match its physical dimensions.

4. Create a Blank PCB Layout

- Once the schematic is complete, open **PCBNew** to create the PCB layout.
- Import the schematic netlist into PCBNew to load the components and their connections.
- You’ll see the components placed in an arbitrary layout outside the board outline.

5. Place Components

- Arrange the components on the PCB. Start with components with fixed placements (like connectors) and then arrange other parts in a logical manner.
- Try to minimize trace lengths and ensure that critical components (like oscillators) are placed optimally to reduce noise.

6. Designing Your PCB Stackup

- Set up the **layer stack** in PCBNew. This involves selecting the number of layers (typically 2 or 4 for simple designs) and defining the materials and thickness for each layer.
- Specify power and ground planes, if required, in the design.

7. Route Traces

- Connect the components by routing traces in **PCBNew**. Start with power and ground traces, then proceed to signal lines.
- Follow the design rules you set earlier and keep trace lengths short to reduce parasitics.
- Utilize via stitching, ground planes, or pour copper for better grounding and EMI control, if required.

8. Design Rules Check (DRC)

- Define **Design Rules** in PCBNew to set minimum trace width, spacing, and other layout constraints based on manufacturing capabilities.
- Set these constraints according to the manufacturing specifications for your PCB.

9. Generate Design Files

- Once routing and design are complete, perform a **final DRC** to ensure there are no errors.
- Generate the necessary **Gerber files**, drill files, and assembly drawings required by the PCB manufacturer.
- Review the files to ensure everything is accurate before submitting them for production.

PCB Design layout & 3D:

The design a circuit integrating the **ESP32-WROOM-32E** module with the **CH340G** USB-to-serial converter, **SHT30**, and **boot** and **reset** buttons, here’s overview PCB deisgn layout & 3D images :