

# Host-Device Communication via USB CDC (Windows)

This guide walks you through setting up a **Windows PC as the host controller** and an **ESP32-S3-based robot driver board as the device**, connected through USB CDC.

Once configured, you can use the PC to send commands, run automation scripts, and build complex robot control workflows.

 **GitHub Repository:**



[https://github.com/EffectsMachine/robot\\_driver\\_with\\_esp32s3\\_lite](https://github.com/EffectsMachine/robot_driver_with_esp32s3_lite)




## 1. What Is USB CDC?

**USB CDC (Communications Device Class)** is a standard protocol that lets a USB device appear as a **virtual serial port** on your computer.

It allows direct communication between your PC (the host) and the driver board (the device) — without requiring a separate USB-to-UART converter.

## 2. Why Use USB CDC?

Feature	Description	Benefits
 <b>No Converter Required</b>	The board includes a native USB interface.	Simplifies wiring, reduces parts and cost.
 <b>High-Speed Data</b>	Transfer rate up to 60–65 packets/sec.	Stable, fast, and ideal for debugging or real-time control.

Feature	Description	Benefits
 <b>More Reliable</b>	Uses the ESP32-S3's native USB port — no extra CH340/CP2102 chip.	Fewer connection issues, higher durability.
 <b>Plug &amp; Play</b>	Automatically recognized by most OSes.	No manual driver installation needed.
 <b>Cross-Platform Support</b>	Works on Windows / macOS / Linux / Android.	One unified communication method.

### 3. Graphical Debugging with YAT

💡 If you're new to USB CDC, start here.

This method uses a simple graphical terminal to test communication before running automation scripts.

#### Step 1: Install YAT

Download and install **YAT (Yet Another Terminal)**:

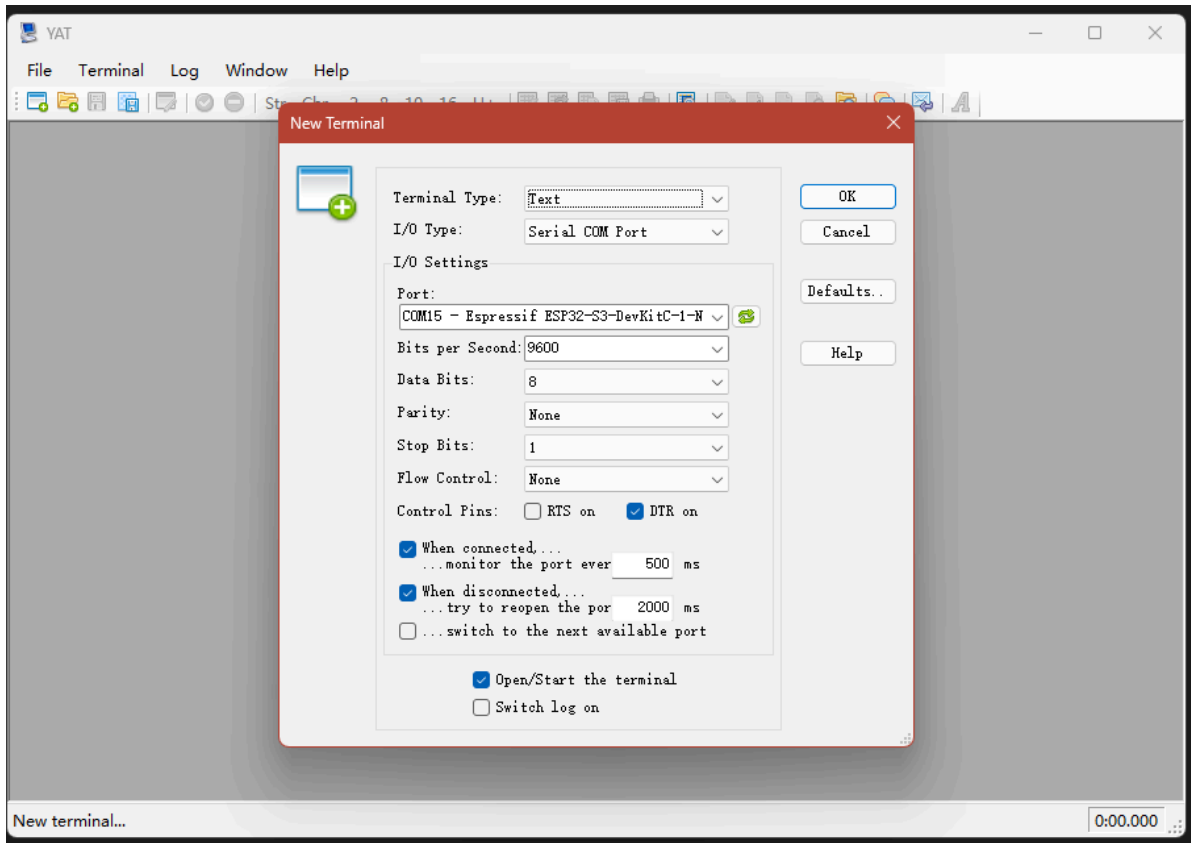
[YAT-2.7.2\\_x64\\_\(64-bit\)\\_installer.zip](#)

#### Step 2: Connect the Board

- Use the **middle USB port** on the driver board (⚠️ *not the UART port*).
- Wait a few seconds after connecting for Windows to detect the device.

#### Step 3: Open YAT

- Launch YAT → press **Ctrl + N** to open a new terminal session.



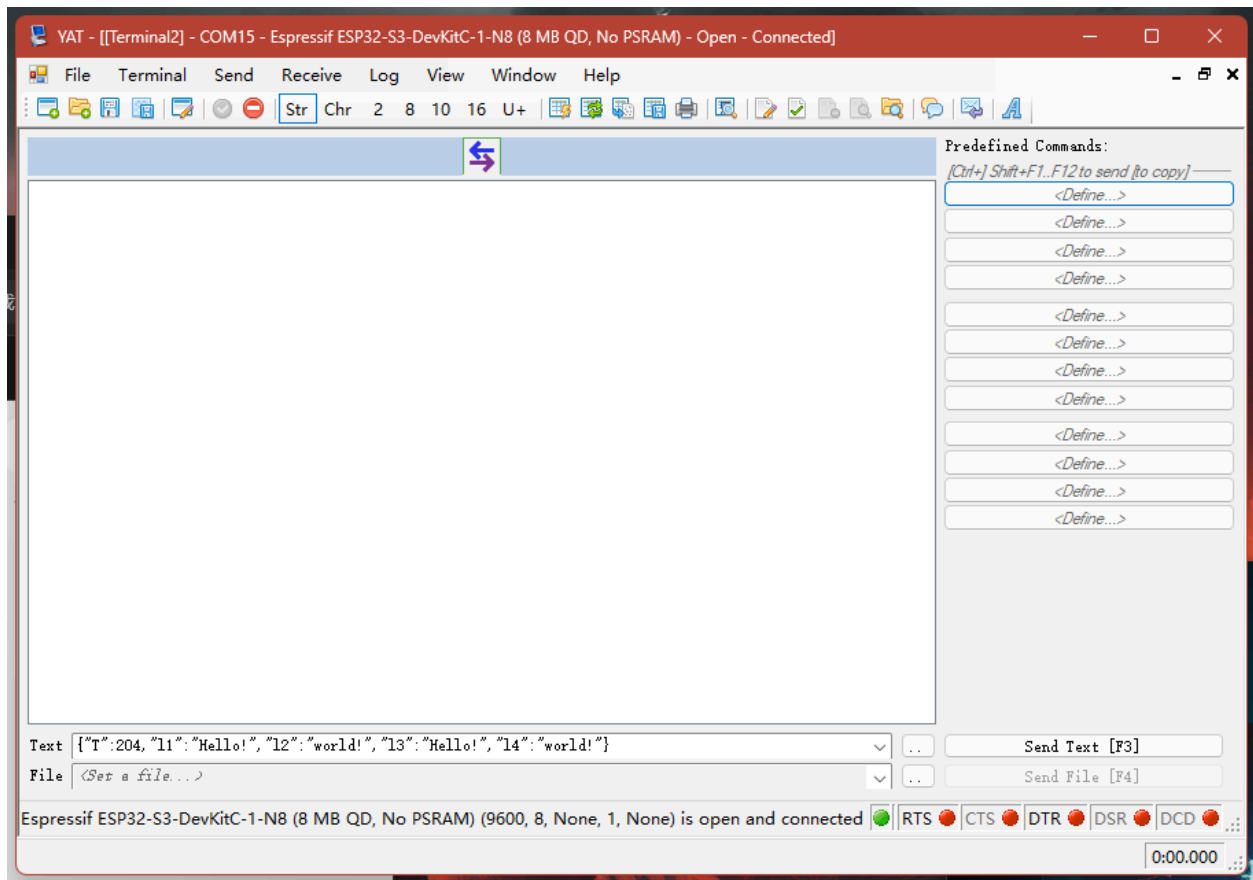
- In the **Port** dropdown, select **Espressif ESP32-S3...** (the name may vary).
- Enable **DTR on**, then click **OK**.

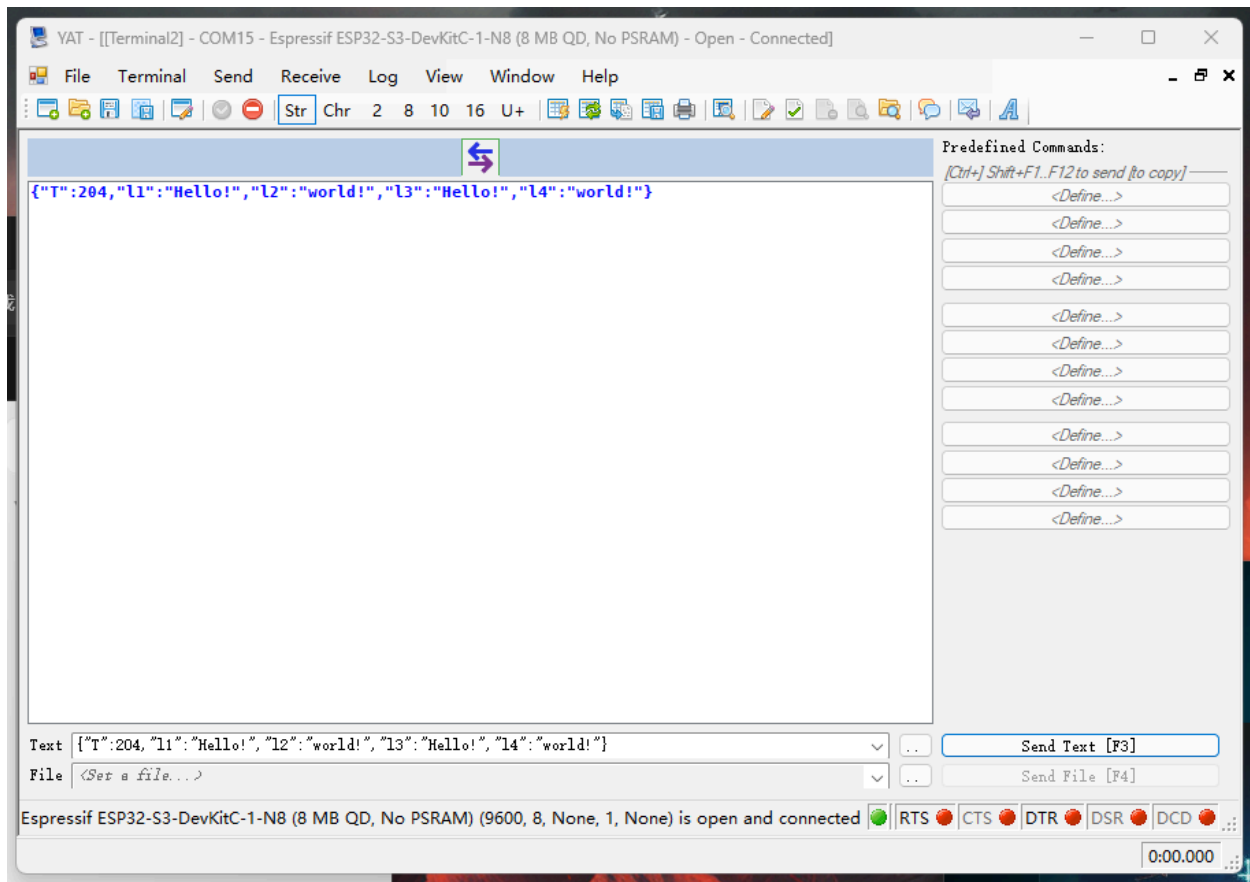
*(Baud rate settings are ignored for USB CDC.)*

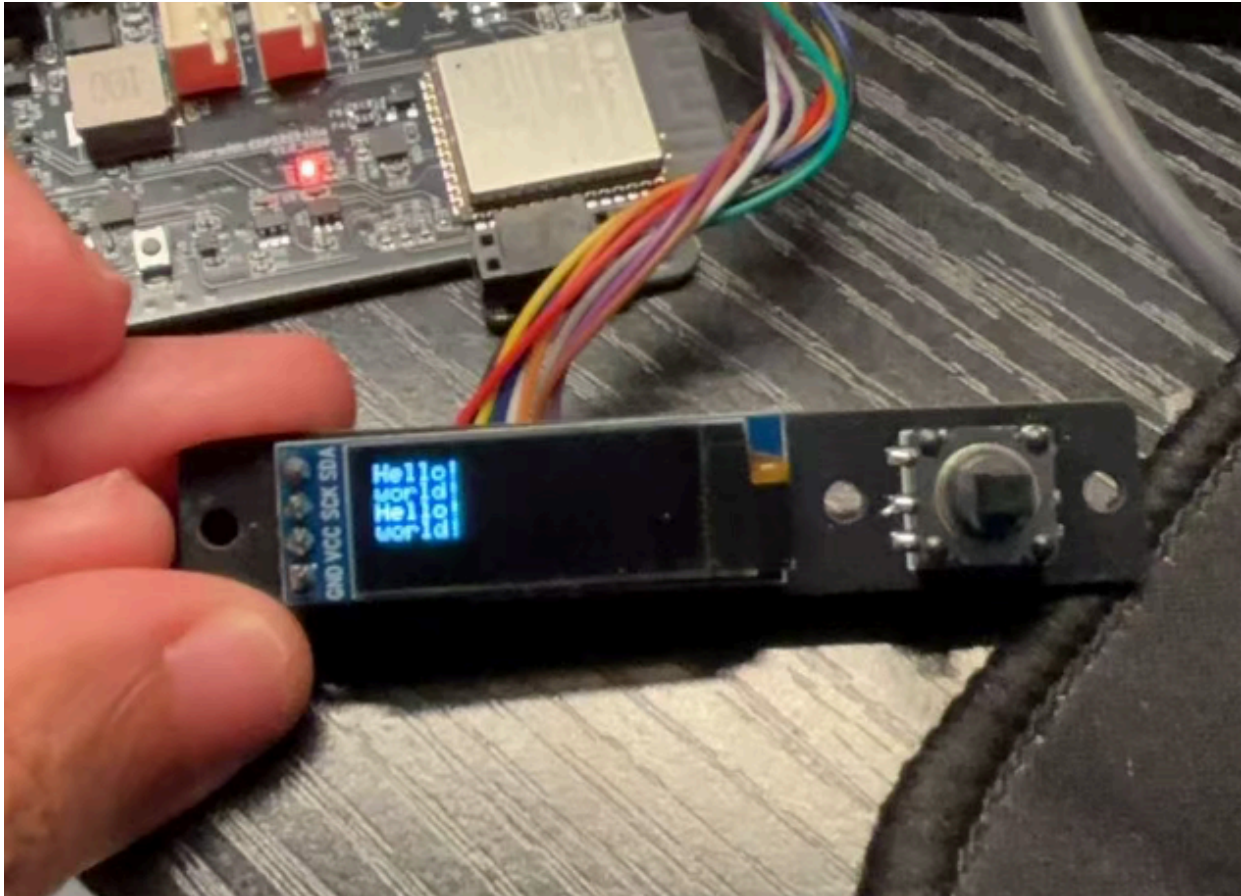
## Step 4: Send a Test Command

In the "Text" input box, enter the following JSON command and click **Send Text**:

```
{"T":204,"I1":"Hello!","I2":"World!","I3":"Hello!","I4":"World!"}
```







If the OLED screen on your board displays new text, congratulations — **USB CDC communication is working!**

✅ Once verified, close YAT — leaving it open will block your Python script from accessing the same COM port.

## 4. Setting Up the Python Environment

This section shows how to control the board via Python using USB CDC communication.

### Step 1: Install Python

1. Go to [python.org/downloads](https://python.org/downloads).
2. Download and install **Python 3.10+**.
3. During installation, **check** the option **Add Python to PATH**.

## Step 2: Locate Example Files

Inside the cloned GitHub project, navigate to:

```
Example for Robot Driver Lite/python_example/usb_cdc
```

This folder contains the ready-to-run example scripts.

---

## 5. Creating a Virtual Environment

 This step only needs to be done once.

It creates an isolated Python environment so different projects won't conflict.

1. Right-click inside the folder → **Open in Terminal**

The path should match the folder above.

2. Run:

```
python -m venv venv
```

3. Activate the environment:

```
.\venv\Scripts\activate
```

4. Install dependencies:

```
pip install -r requirements.txt
```

---

## 6. Editing the Example Script

Open `usb_cdc_example.py` in your preferred editor (VS Code, Sublime Text, or even Notepad).

### Step 1: Set the Port

Locate the line defining `PORT`, and replace it with your board's COM port (e.g. `"COM11"`):

```
PORT = "COM11"
```

USB CDC does not require a baud rate — you can leave that parameter as-is.

## Step 2: (Optional) Edit the Test Command

You can modify the JSON data sent to the board:

```
data = {"T":202, "line":1, "text":"Hello, world!", "update":1}  
send_json(data)
```

Save the file after making changes.

## 7. Running the Example

In the terminal (with your virtual environment activated), run:

```
python usb_cdc_example.py
```

If everything is set up correctly, your board will respond to the command — for example, updating the display or sending data back via USB CDC.

## 8. Summary

At this point, you've:

- Verified communication using YAT
- Configured Python and dependencies
- Sent and received JSON commands via USB CDC

You're now ready to build your own automation scripts, real-time robot control systems, or custom interfaces on top of the provided example.