

Host-Device Communication via USB CDC (Linux / macOS)

This guide explains how to use your **Linux or macOS computer as the host controller** and an **ESP32-S3-based robot driver board as the device**, communicating through USB CDC (Communications Device Class).

Once connected, you can send JSON commands, run Python automation scripts, and create your own robot control logic.






 **GitHub Repository:**

https://github.com/EffectsMachine/robot_driver_with_esp32s3_lite

1. Why USB CDC?

USB CDC (Communications Device Class) allows a microcontroller to appear as a *virtual serial port* when connected to your computer.

It's a simple, high-speed, and driver-free way to send data between your PC and the board.

Advantage	Description	Benefit
 No Converter Needed	Uses native USB—no CH340 or CP210x chips.	Cleaner wiring, lower cost.
 Faster Communication	Direct USB connection (up to 60–65 packets/sec).	Smooth real-time control.
 Stable & Reliable	Fewer conversion layers = less interference.	Lower failure rate.
 Plug & Play	Automatically recognized on Linux/macOS.	No driver setup required.
 Cross-Platform	Works on Windows, macOS, and Linux.	Develop anywhere.

2. Quick Test via Terminal (Recommended)

💡 No extra software needed — Linux and macOS include everything required for USB CDC testing.

Step 1: Connect the Device

Connect your robot driver board to your computer using a **USB-C cable**.

Once powered, your system will detect it as a serial device.

Find the device name using:

```
ls /dev/ttyACM* /dev/ttyUSB* /dev/tty.usbmodem* /dev/cu.usbmodem*
```

Example outputs:

```
/dev/ttyUSB0
```

or on macOS:

```
/dev/tty.usbmodem12345
```

Take note of the correct port.

Step 2: Open a Serial Terminal

You can use `screen`, `minicom`, or `picocom`.

`screen` comes preinstalled on most macOS and Linux systems.

```
screen /dev/ttyUSB0 115200
```

Note: USB CDC does not actually use a baud rate, but screen requires one — any number is fine.

Once connected, type the following JSON command and press Enter:

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

If the text updates on your board's display, communication is working perfectly.

To exit `screen` :

Press `Ctrl + A` , then `K` , then confirm with `y` .

3. Running the Python Example

You can automate commands using the provided Python example.

Step 1: Check or Install Python

Most systems already have Python preinstalled. Confirm with:

```
python3 --version
```

If not, download and install Python 3.10+ from python.org.

Step 2: Clone the Project

```
git clone https://github.com/EffectsMachine/robot_driver_with_esp32s3_lite.git
```

```
cd robot_driver_with_esp32s3_lite/Example\ for\ Robot\ Driver\ Lite/python_example/usb_cdc
```

Step 3: Create a Virtual Environment

You only need to do this once:

```
python3 -m venv venv
```

Activate it:

```
source venv/bin/activate
```

Install dependencies:

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

Step 4: Edit the Example File

Open `usb_cdc_example.py` in your text editor (VS Code, Sublime Text, or nano).

Find and update the serial port path to match your device:

```
PORT = "/dev/ttyUSB0" # or "/dev/tty.usbmodem12345" on macOS
```

Baud rate can stay as is — USB CDC ignores it.

Optionally, modify the JSON message to send:

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

Save and close the file.

Step 5: Run the Script

```
python3 usb_cdc_example.py
```

Expected output:

```
>>> Sent: {"T":202,"line":1,"text":"Hello, world!","update":1}
```

Your board should now display or react according to the command.

4. How It Works

The `usb_cdc_example.py` script:

- Opens the USB CDC port.
- Sends JSON commands to the driver board.
- Prints responses for debugging.
- Can be extended for automation or interactive control.

You can use this example as a base to:

- Create custom robot control programs
 - Send chained commands for movement sequences
 - Build cloud-connected or ROS2-based control systems
-

5. Summary

Using **USB CDC on Linux/macOS**, you can:

- Connect instantly — no drivers or extra tools
- Communicate directly from the terminal
- Run Python scripts to automate robot control

From here, you can expand into complex control logic, monitor feedback, or integrate remote operation — all using a single USB cable.