

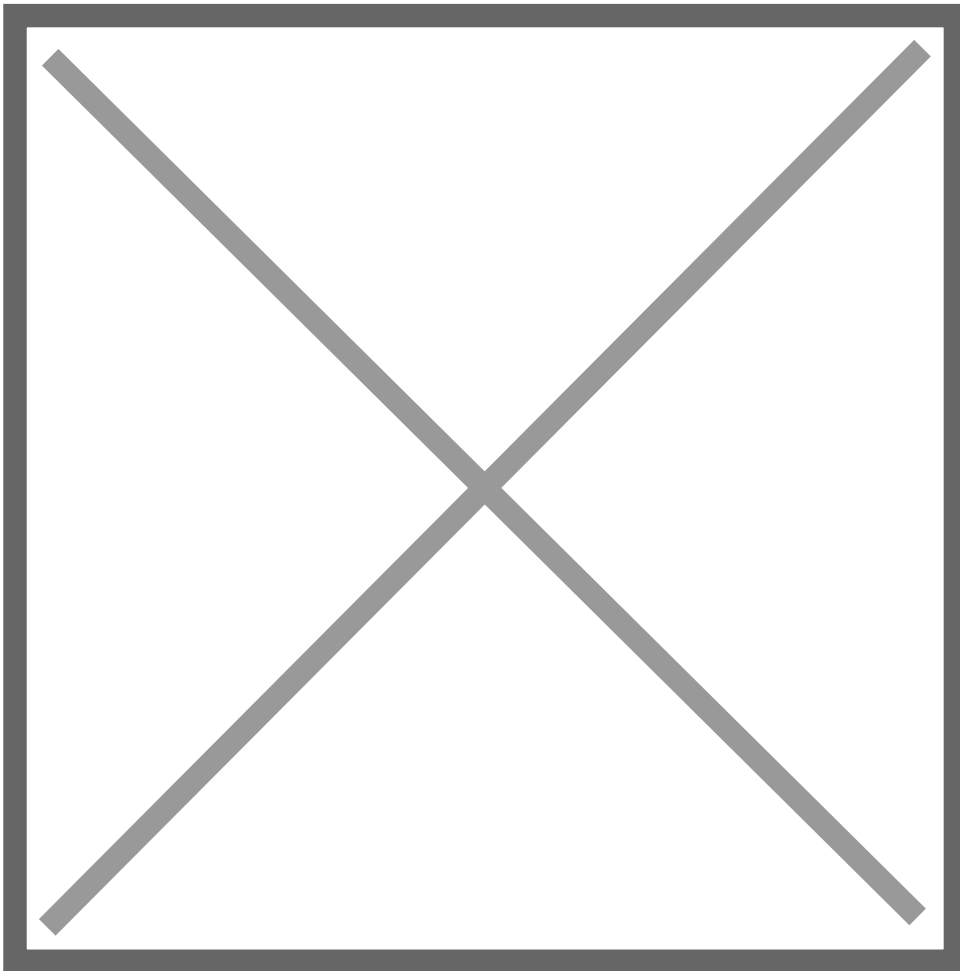
Linux Configuration

This tutorial will help you to execute all the steps before connecting your Dev Chip Board to your computer.

- [Resolving Serial Port Access Issues for the Dev Chip Board on Linux](#)
- [Steps to Configure the Dev Chip Board Port](#)

Resolving Serial Port Access Issues for the Dev Chip Board on Linux

When using the Dev Chip Board on some Linux distributions, especially on Ubuntu, you may encounter issues accessing the serial port. This usually happens due to conflicts with the brltty service or user permission restrictions. Below is an image representing when the program fails to recognize the board's ports.



This guide provides a detailed step-by-step process to diagnose and fix these issues, ensuring that your board is correctly recognized and accessible via `/dev/ttyUSB0` (or another corresponding port).

Steps to Configure the Dev Chip Board Port

Step 1:

First, connect the Dev Chip Board to your computer's USB port and run the following command in the terminal to list the USB devices:

```
lsusb
```

This should display a device similar to:

```
Bus 001 Device 003: ID 10c4:ea70 Silicon Labs CP2105 Dual USB to UART Bridge Controller
```

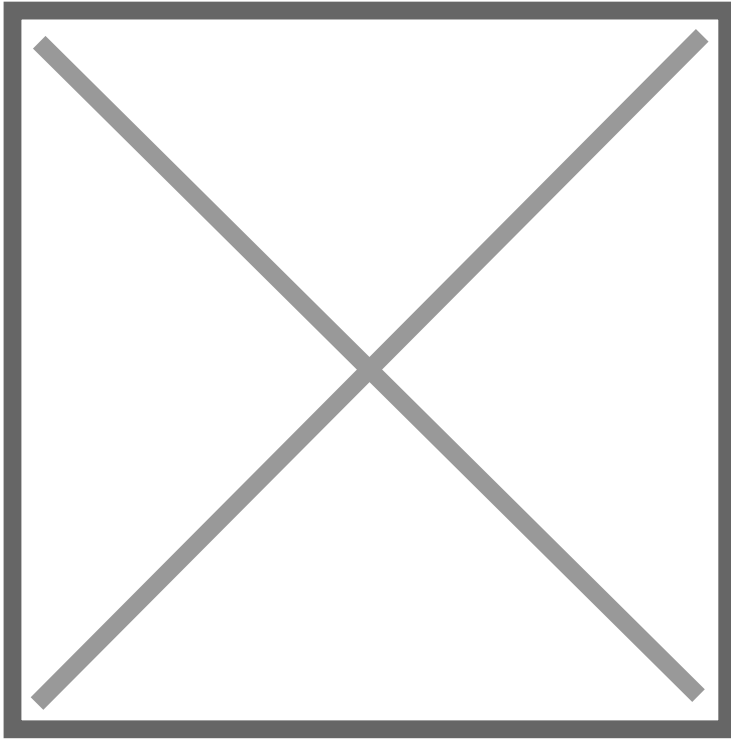
Now, list the available serial devices:

```
ls /dev/ttyUSB*
```

If your board has been detected correctly, you will see:

```
/dev/ttyUSB0 /dev/ttyUSB1
```

If nothing appears, as shown in the image below, you may need to install the drivers.



Step 2 (Optional):

The CP210 driver is already included in the latest versions of the Linux kernel. However, if you need to install or update the drivers manually, run:

```
sudo apt update  
sudo apt install usbutils
```

If the ports are still not recognized, try loading the kernel module:

```
sudo modprobe cp210x
```

To ensure the module is loaded automatically at startup, add it to the `/etc/modules` file:

```
echo "cp210x" | sudo tee -a /etc/modules
```

Step 3:

By default, only the root user can access UART ports. To allow your user to access them, add yourself to the dialout group:

```
sudo usermod -a -G dialout $USER
```

Now, log out and log back in or restart the system for the changes to take effect.

To apply the permissions manually without restarting, run:

```
sudo chmod 666 /dev/ttyUSB0  
sudo chmod 666 /dev/ttyUSB1
```

If you want this configuration to persist after a reboot, create a Udev rule.

Step 4:

Create a new Udev rules file:

```
sudo nano /etc/udev/rules.d/99-cp2105.rules
```

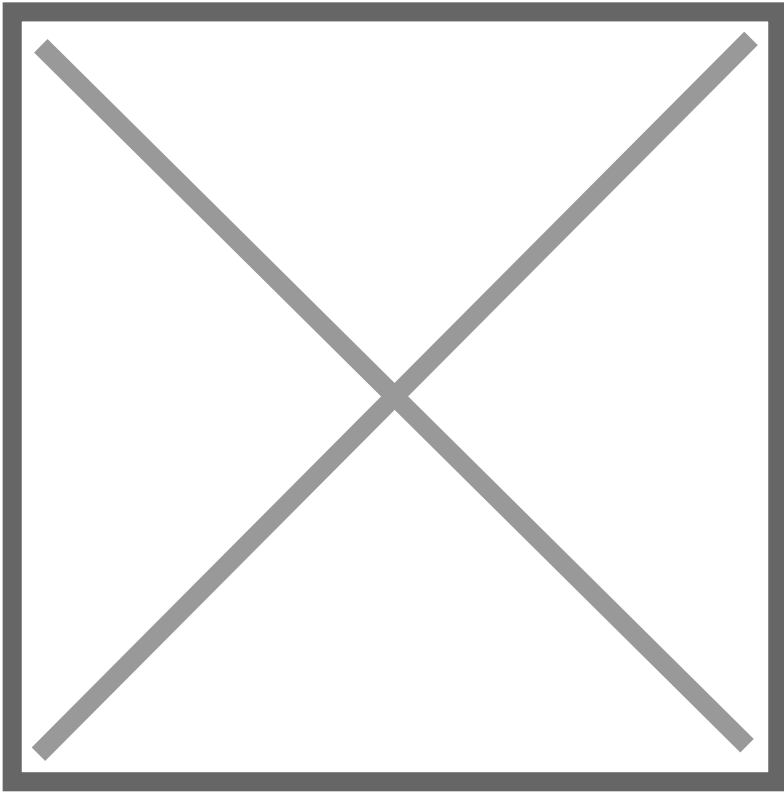
Add the following lines:

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea70", MODE="0666",  
GROUP="dialout"
```

Save (CTRL+X, Y, Enter) and apply the new rules:

```
sudo udevadm control --reload-rules  
sudo udevadm trigger
```

Now, the correct permissions will be automatically applied whenever the device is connected.



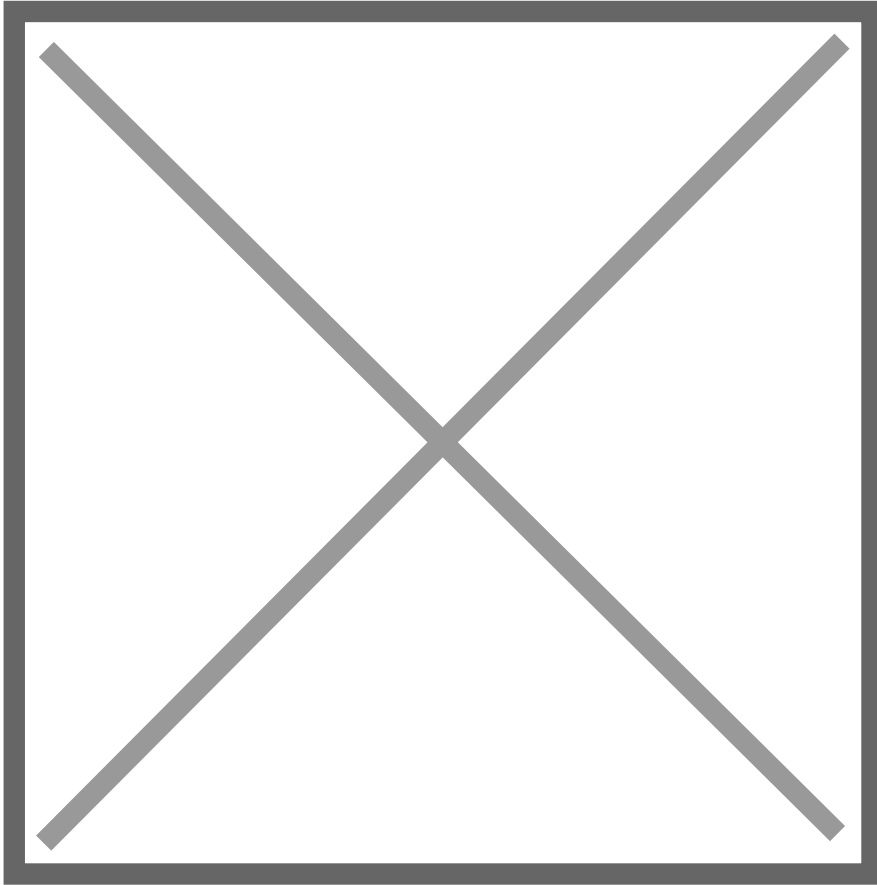
To check if the ports are correctly configured, use:

```
ls -l /dev/ttyUSB*
```

The output should be similar to this:

```
crw-rw-rw- 1 root dialout 188, 0 Mar 3 12:00 /dev/ttyUSB0
crw-rw-rw- 1 root dialout 188, 1 Mar 3 12:00 /dev/ttyUSB1
```

And you will see this image when the connection is successful:



Conclusion

By following these steps, you should be able to access the Dev Chip Board via the serial port without permission denial issues or conflicts with other processes. If the problem persists, check whether the correct drivers are installed and if the USB cable is functioning properly.