

Installing the driver in Linux

Installing the EETI resistive touch screen driver in Linux involves loading the driver module, setting up the configuration file, and calibrating it. The specific steps are as follows:

Driver Obtaining and Preparation

1. Obtaining the Compatible Driver

- First, download the corresponding Linux driver from the EETI official website (<http://www.eeti.com>). This typically includes the kernel module (.ko file) and configuration tool.
- For embedded Linux devices (such as ARM architecture development boards), ensure that the driver version matches the kernel version (check using `uname -r`) and CPU architecture.

2. Installing Dependent Tools

Ensure the necessary tools are installed on your system:

```
``bash

sudo apt-get update

sudo apt-get install -y module-assistant dkms # For kernel module
management

...

```

Installing the Driver Module

1. Manually Loading the Driver (Temporary Test)

- Copy the driver file (e.g., `eeti_ts.ko`) to the system and execute the

following command to load it:

```
```bash
```

```
sudo insmod /path/to/eeti_ts.ko # Replace `` with the actual driver path
```

```
```
```

- Verify that the driver was loaded successfully:

```
```bash
```

```
lsmod | grep eeti # If there is output, the driver has been loaded
```

```
dmesg | grep -i touch # Check the driver initialization log
```

```
```
```

2. Permanently Installing the Driver (Automatically Loaded at Boot)

- Copy the driver module to the system module directory:

```
```bash
```

```
sudo cp /path/to/eeti_ts.ko /lib/modules/$(uname
-r)/kernel/drivers/input/touchscreen/
```

```
```
```

- Update module dependencies:

```
```bash
```

```
sudo depmod -a
```

```
```
```

- Create a load configuration to ensure automatic loading at boot:

```
```bash
```

```
echo "eeti_ts" | sudo tee /etc/modules-load.d/eeti.conf
```

...

## Configuring the Touchscreen Device

### 1. Device Permission Settings

Ensure the touch device node (usually `/dev/input/eventX`) has read and write permissions:

```
``bash
```

```
View the event node corresponding to the touchscreen device (observe changes by plugging and unplugging the device)
```

```
ls -l /dev/input/
```

```
Temporarily set permissions (will expire after reboot)
```

```
sudo chmod 666 /dev/input/eventX # Replace with the actual event node
```

```
Permanently set permissions: Create a udev rule
```

```
sudo tee /etc/udev/rules.d/99-eeti-touch.rules <<EOF
```

```
SUBSYSTEM=="input", ATTRS{name}=="EETI Touch Screen", MODE="0666"
```

```
EOF
```

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

...

### 2. Xorg Configuration (GUI Adaptation)

If using the X11 desktop, you need to create an Xorg configuration file to allow the system to recognize the touch screen:

```
``bash
```

```
sudo tee /etc/X11/xorg.conf.d/99-eeti.conf <<EOF
```

Section "InputDevice"

Identifier "EETI Touchscreen"

Driver "eeti"

Option "Device" "/dev/input/eventX" # Replace with the actual event node

Option "MinX" "0"

Option "MaxX" "4095" # Adjust according to screen resolution

Option "MinY" "0"

Option "MaxY" "4095"

Option "SwapXY" "0" # If the touch coordinates are reversed, set this to 1.

EndSection

Section "ServerLayout"

Identifier "Default Layout"

InputDevice "EETI Touchscreen" "CorePointer"

EndSection

EOF

...

## Touchscreen Calibration

### 1. Using the EETI Official Calibration Tool

Some driver packages include calibration tools (such as `eeti\_calibrator`).

After running, follow the prompts to click the calibration points:

```
```bash
```

```
sudo eeti_calibrator /dev/input/eventX # Replace with the actual event node
```

...

2. Universal Calibration Tool (xinput_calibrator)

If the official tool is not available, install the universal calibration tool:

```
```bash
```

```
sudo apt-get install xinput-calibrator
```

xinput\_calibrator # Follow the prompts to complete the calibration, generate the configuration, and save it.

...

## Verification and Troubleshooting

### 1. Verifying Touch Functionality

- Run `xinput list` to check if the EETI touch device is recognized.
- Use `evtest /dev/input/eventX` to test touch events (press Ctrl+C to exit).

Coordinates should be output when touching the screen.

### 2. Troubleshooting

- Driver Load Failure: Check whether the driver and kernel versions match and re-obtain the corresponding driver version.
- Touch Unresponsive: Confirm that the device node is correct and permissions are set correctly. Check `dmesg` logs to troubleshoot hardware issues.
- Coordinate Inversion/Offset: Adjust parameters such as `SwapXY` and `MinX/MaxX` in the Xorg configuration, or recalibrate.

## Notes

- Embedded Linux systems (such as those built with Buildroot or Yocto) typically require integrating drivers into the kernel compilation process. For details, refer to the development documentation provided by the device manufacturer.

- If using the Wayland display server, the configuration method differs from Xorg. Refer to the Wayland Input Device Configuration Specification.

If you encounter specific error messages, please provide logs for further analysis and resolution.