Installing the driver on Android

Installing the EETI resistive touch screen driver on Android typically requires adaptation based on the device's hardware architecture and Android version. The following are general steps and precautions:

Driver Acquisition

1. Official Channels

Prioritize downloading the driver for your model from the EETI official website (http://www.eeti.com). Driver packages for different Android versions (such as Android 7.0 and above) are typically provided, including kernel modules (driver files) and configuration files in `ko` format.

2. Device Manufacturer Support

For specific embedded devices (such as industrial tablets or development boards), contact the device manufacturer to obtain a pre-adapted EETI driver. The manufacturer typically optimizes the driver for the hardware.

Installation Prerequisites

1. System Permissions

The Android device must have root permissions or an unlocked bootloader, as driver installation requires modifying system partitions (such as the `/system` or `/vendor`

directories).

2. Kernel Matching

The driver file (.ko) must fully match the device's kernel version and CPU architecture (e.g., arm, arm64). Otherwise, loading will fail. (You can check the kernel version using the `uname -a` command.)

3. Tool Preparation

- adb tool (for operating the device from a computer)

- a file manager (e.g., ES File Explorer, for operating files on the device)

Installation Steps (Manual Installation as an Example)

1. Connect the device and enter debugging mode

- Enable "Developer Options" on the device and enable "USB debugging."

- Connect the device to the computer via USB and use the `adb devices` command to confirm that the device is connected.

2. Push the driver file to the device

Push the downloaded driver file (e.g., `eeti_ts.ko`) to the device's temporary directory using ADB:

```bash
adb push eeti_ts.ko /data/local/tmp/
```

```

3. Load the driver module

- Enter the device shell: `adb shell`

- Obtain root privileges: `su`

- Load the driver: `insmod /data/local/tmp/eeti_ts.ko`

(If you receive an "Invalid module format" message, this indicates a mismatch between the driver and kernel. You will need to re-obtain the corresponding version.)

4. Configure driver parameters

Depending on your touchscreen model, modify the configuration file (usually the `.idc` file in the `/system/usr/idc/` or `/vendor/usr/idc/` directory) and set touch parameters (such as resolution and calibration data).

5. Setting Up Automatic Loading at Boot

To prevent the driver from failing after a reboot, add the driver to the boot script:

- Edit `/init.rc` or a device-specific startup script (such as one in `/etc/init.d/`) and add the following:

```bash
insmod /system/lib/modules/eeti_ts.ko
```

- Move the driver file to the system module directory:
```

```bash
mv /data/local/tmp/eeti_ts.ko /system/lib/modules/
```

Verifying the Driver's Effectiveness

1. Run the `lsmod` command to check if the `eeti_ts` module is loaded.

2. Touch the screen and check the driver log using the `dmesg | grep eeti` command to confirm touch event response.

3. If touch is unresponsive, the screen may need to be calibrated. Some EETI drivers provide calibration tools (such as `eeti_calibrator`).

Notes

1. Android Version Restrictions

Higher versions of Android (such as Android 10 and above) have stricter restrictions on system partition permissions. Driver integration may require using the `magisk` module or recompiling the kernel.

2. Embedded Device Compatibility

For industrial embedded devices, it is recommended to directly use the firmware flashing tool provided by the manufacturer to integrate the driver into the system image to avoid manual operations that may cause compatibility issues.

3. Driver Conflicts

If the device already has other touch drivers, uninstall them first (using the `rmmod`

command) to avoid conflicts.