

配置CAN

1. 开机自启, 执行配置CAN脚本

使用 Systemd 实现开机自启动

1. 创建一个 Systemd 服务单元文件

首先, 创建一个以 `.service` 结尾的文件, 用于描述服务或者脚本。假设脚本位于 `/path/to/your/can.sh`, 可以按如下步骤创建服务单元文件:

```
sudo nano /etc/systemd/system/CAN0Conf.service
```

在编辑器中输入以下内容 (根据实际情况修改路径和服务描述):

```
[Unit]
Description=Configures CAN interface can0 on startup
After=network.target

[Service]
ExecStart=/path/to/your/can.sh
Restart=always
RestartSec=10    # 如果发生故障, 每隔10秒重启

[Install]
WantedBy=multi-user.target
```

- `Description`: 描述服务的信息。
- `After=network.target`: 指定启动顺序, 确保在网络服务启动后才启动脚本。
- `ExecStart`: 指定需要执行的脚本或命令的路径。
- `Restart=always` 和 `RestartSec=10`: 在脚本意外退出时自动重启, 并且定义重启间隔时间。

2. 设置文件权限

保存并关闭编辑器后, 确保服务单元文件的权限正确:

```
sudo chmod 644 /etc/systemd/system/CAN0Conf.service
```

3. 启用 Systemd 服务

启用服务, 使其在系统启动时自动运行:

```
sudo systemctl enable CAN0Conf.service
```

这样, 脚本就会在系统启动时自动运行了。

4. 控制服务

- 启动服务: `sudo systemctl start CAN0Conf.service`
- 停止服务: `sudo systemctl stop CAN0Conf.service`
- 重启服务: `sudo systemctl restart CAN0Conf.service`
- 查看服务状态: `sudo systemctl status CAN0Conf.service`

- 禁用服务 (不再开机自启动) : `sudo systemctl disable CAN0Conf.service`

2. 配置CAN脚本

```
#!/bin/bash
# Script: can.sh
# Description: Configures CAN interface can0 on startup

# Check if running as root
if [[ $EUID -ne 0 ]]; then
    echo "This script must be run as root"
    exit 1
fi

# Disable CAN interface can0 if it's up
if ip link show can0 &> /dev/null; then
    ip link set can0 down
fi

# Load necessary kernel modules
modprobe can
modprobe can_raw
modprobe mttcan

# Configure CAN interface can0 with bitrate 250000
ip link set can0 type can bitrate 250000
ip link set up can0

# Check the return status of the last command
if [[ $? -ne 0 ]]; then
    echo "Error configuring CAN interface can0"
    exit 1
fi

echo "CAN interface can0 configured successfully"
exit 0
```