

## 如何配置LEDE工作

LEDE使用UCI（统一配置接口）来存储和操作其所有配置。

配置LEDE系统的推荐方法是使用UCI的命令行界面。

如果您使用它，您不需要担心在配置文件中或在文件实际位于哪里发生语法错误。

当使用命令行或Web界面修改值时，所有更改都会暂停，并且不会直接保存到文件中，因此请记住在设置更改后保存更改。

### Uci配置内部结构

配置分为子系统，每个子系统分为几个部分，每个部分包含一个key='value'条目的列表。

可用的sybs系统有：**dnsmasq**，**dropbear**，**firewall**，**fstab**，**net**，**qos**，**samba**，**system**，**wireless**。

所有UCI配置都存储为一个如下所示的值列表：

**<选项> = <string>**

其中<option>是可选的。

这里是一个例子，从uci的命令行界面中查看“系统”子系统的前两个部分：

```
root @ lede: ~#uci显示系统
系统.@系统[0] =系统
系统.@系统[0].hostname = '莱德'
系统.@系统[0].timezone = 'UTC'
system.ntp =时间服务器
system.ntp.server = '0.lede.pool.ntp.org' '1.lede.pool.ntp.org' '2.lede.pool.ntp.org' '3.lede.pool.ntp.org'
system.ntp.enabled = '1'
system.ntp.enable_server = '0'
```

这是两个相同的两个部分在实际的配置文件内部；

```
root @ LEDE: ~#cat / etc / config / system
配置系统
    选项主机名'lede'
    选项时区'UTC'

config timeerver'ntp'
    列表服务器'0.lede.pool.ntp.org'
    列表服务器'1.lede.pool.ntp.org'
    列表服务器'2.lede.pool.ntp.org'
    列表服务器'3.lede.pool.ntp.org'
    选项启用'1'
    选项enable_server'0'
```

### Uci命令行工具内置帮助文本

用法: uci [<options>] <command> [<arguments>]

命令:

```
批量
导出[<config>]
导入[<config>]
更改[<config>]
提交[<config>]
添加<config> <section-type>
add_list <config>.<section>.<option> = <string>
del_list <config>.<section>.<option> = <string>
show [<config> [. <section> [. <option>]]]
get <config>.<section> [. <option>]
设置<config>.<section> [. <option>] = <value>
删除<config> [. <section> [[. <option>] [= <id>]]]
重命名<config>.<section> [. <option>] = <name>
恢复<config> [. <section> [. <option>]]
重新排序<config>.<section> = <position>
```

选项:

```
-c <path>设置配置文件的搜索路径 (默认值: / etc / config)
-d <str>为uci show中的列表值设置分隔符
-f <file>使用<file>作为输入, 而不是stdin
-m在导入时, 将数据合并到现有包中
-n名称未命名部分导出 (默认)
-N不要命名未命名的部分
-p <path>添加配置更改文件的搜索路径
-P <path>添加配置更改文件的搜索路径, 并使用默认值
-q安静模式 (不打印错误信息)
-s强制严格模式 (停止解析器错误, 默认)
-S禁用严格模式
-X不要在'show'上使用扩展语法
```

## 显示整个当前配置

```
root @ lede: /#uci显示
```

## 显示子系统的当前配置

```
root @ lede: /#uci show SUBSYSTEM_NAME
```

可用的sybsystems是: **defaults, dnsmasq, dropbear, firewall, fstab, net, qos, samba, system, wireless**。  
这里有一个例子:

```
root @ lede: /#uci显示系统
系统.@系统[0] =系统
系统.@系统[0].hostname = '莱德'
系统.@系统[0].timezone = 'UTC'
system.ntp =时间服务器
system.ntp.server = '0.openwrt.pool.ntp.org' '1.openwrt.pool.ntp.org' '2.openwrt.pool.ntp.org' '3.openwrt.pool.ntp.org'
system.ntp.enabled = '1'
system.ntp.enable_server = '0'
```

## 设置单值选项

```
root @ lede: /#uci set key = 'value'
```

例如:

```
root @ lede: /#uci set system.@ system [0].hostname = 'foo'
```

## 在列表选项上附加或删除值

```
root @ lede: /#uci add_list key = 'value'
root @ lede: /#uci delete_list key = 'value'
```

这是操作上面的`system.ntp.server`键所必须做的。

## 向子系统配置添加新的部分

```
root @ lede: /# uci添加SECTION_NAME SUBSYSTEM_NAME
```

将在称为SUBSYSTEM\_NAME的子系统内生成一个名为SECTION\_NAME的新部分，之后可以按常规向该部分添加密钥。它将打印一个字母数字代码，您可以将其用作“部分”，以进一步添加键。看这个例子：

```
root @ LEDE: ~# uci添加防火墙规则
cfg2092bd
root @ LEDE: ~# uci set firewall.cfg2092bd.src ='wan'
```

## 显示尚未保存的修改值

```
root @ lede: /# uci更改
```

## 保存单个子系统的修改值

```
root @ lede: /# uci commit SUBSYSTEM_NAME
root @ lede: /# reload_config
```



原文

```
uci batch <<EOF set <config>.$rule_name.<option1>='value'
```

提供更好的翻译建议

## 保存所有修改的值

```
root @ lede: /# uci commit
root @ lede: /# reload_config
```

## 使用简单的复制粘贴生成完整的uci部分

当您添加新的部分（见上文）时，此代码块捕获uci打印的代码，并将其重新添加到要添加的所有新密钥之后。这将自动化一个非常有趣的打字或复制粘贴作业。您也可以在脚本中执行此操作。

通用版本：

```
rule_name = $(uci add <config> <section-type>)
uci batch << EOF set <config>.$ rule_name.<option1> ='value'
设置<config> $ rule_name.<option2> ='value'
设置<config> $ rule_name .. <option3> ='value'
#...其中任何一个
EOF
uci承诺
```

一个工作实例：

```
rule_name = $(uci添加防火墙规则)
uci批处理<< EOF防火墙$ rule_name.enabled ='1'
设置防火墙$ rule_name.target ='ACCEPT'
设置防火墙$ rule_name.src ='wan'
设置防火墙$ rule_name.proto ='tcp udp'
设置防火墙$ rule_name.dest_port ='111'
设置防火墙$ rule_name.name ='NFS_share'
EOF
uci承诺
```

## 手动编辑配置文件

所有配置文件都存储在 /etc/config/ 文件夹中，它们是以可读的方式写入的纯文本文件。

每个文件都是UCI子系统，它包含多个选项，选项可以是单个值或值列表。

您可以手动编辑这些文件，但请记住，即使是较小的语法错误也会使整个配置不可读，因此推荐的方法是使用UCI的命令行界面，如上所述。

```
root @ lede: /# cat etc / config / system
配置系统
 选项主机名
 选项时区UTC

配置时间服务器ntp
 列表服务器0.openwrt.pool.ntp.org
 列表服务器1.openwrt.pool.ntp.org
 列表服务器2.openwrt.pool.ntp.org
 列表服务器3.openwrt.pool.ntp.org
 选项启用1
 选项enable_server 0
```

