

NKCAN Linux 用户手册

版本:1.0.0

日期:24/06/04

作者:EA



NKCAN_Linux

用户手册

NODKA

1.0.0

目录内容:

1. 历史	2
2. 产品介绍	3
2.1. 规格参数	3
2.2. 系统需求	4
2.3. 供货信息	4
3. 驱动程序和 API	5
3.1. Linux	5
3.1.1. 安装驱动	5
3.1.2. 卸载驱动	7
3. 使用 ifconfig -a 查看 CAN 设备	7
3.1.3. 配置 CAN 总线	8
3.1.4. 使用 CAN -utils 操作 CAN 总线	9
3.1.5. 例程	10
4. 升级	14
4.1. 升级软件	14
4.2. 升级步骤	15

1. 历史

版本	日期	作者	描述
1.0.0	2023-1-10	EA	初始版本

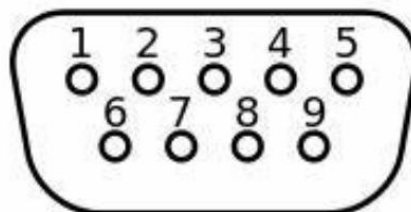
2. 产品介绍

NKCAN 适配器是可经由 USB 把一台电脑轻松连接到 CAN 网络，可连接 CAN 总线,用相应的电缆可最多连接 2 路 CAN，每个 CAN 通道单独隔离 USB，最大 500V。该卡支持 CAN2.0A 和 CAN2.0B 协议规范新特性，因此 CAN 节点可用于现有 CAN 网络。

2.1. 规格参数

- 高速 USB 2.0 适配器（兼容 USB 1.1 和 USB 3.0）。
- 通过连接电缆和 D-Sub, 9 针连接器连接 CAN 总线。
- 120 欧姆的终端电阻放置在 D-Sub 连接器上，可通过开关接通或关闭。
- 兼容 CAN 规范 2.0A(11 位 ID)和 2.0B(29 位 ID)。
- CAN 支持 5k bits/s 到 1M bits/s 的比特率。
- 总线负载测量包括错误帧和过载帧。
- CAN 经由 USB 供电。
- 工作温度范围-40—85°C (-40 to 185 °F)。

D-Sub 接口引脚图



管脚	管脚定义
----	------

1	没有连接/可选+5V 输出
2	CAN_L
3	GND
4	没有连接
5	没有连接
6	GND
7	CAN_H
8	没有连接
9	没有连接

2.2. 系统需求

- 计算机上空闲的全尺寸 miniPCIE 插槽。
- 操作系统 Windows 10、Windows11 (x64/ARM64),或 Linux。

2.3. 供货信息

- miniPCIE CAN 卡。
- 连接电缆，包括每个通道的 D-Sub 插头。
- 用于 Windows 11 (x64/ARM64), 10 (x64), and Linux 的驱动
- PDF 格式的用户手册。

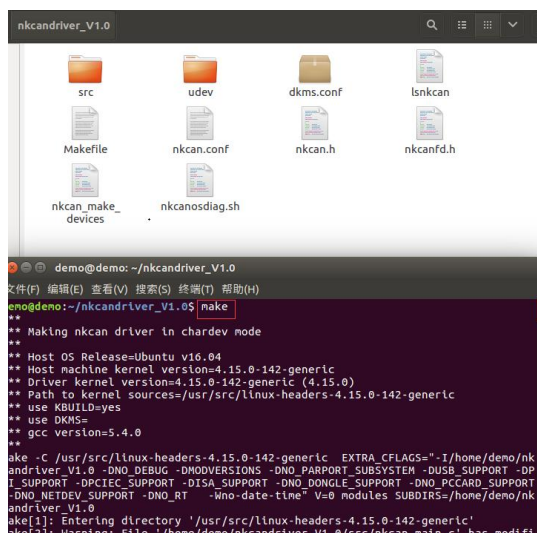
3. 驱动程序和 API

3.1. Linux

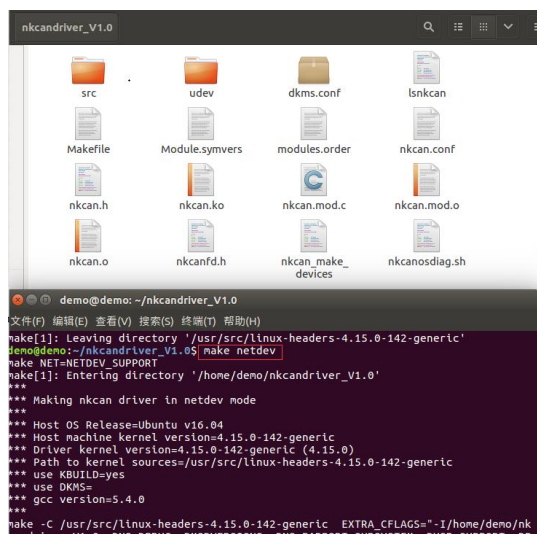
本章将说明如何在 Linux 平台上构建和安装驱动程序，包括测试方法。

3.1.1. 安装驱动

1. 在文件下打开终端并输入：make



2. 在终端输入：Make netdev



编译完成后会生成.o/.ko 文件。

3. 在终端输入：sudo make install

4. 在终端输入：sudo modprobe nkcan

```
demo@demo:~/nkcdriver_V1.0$ sudo make install
mkdir -p /usr/local/bin
cp -f udev/nkcan_usb_minor_check.bash /usr/local/bin
chmod 744 /usr/local/bin/nkcan_usb_minor_check.bash
cp -f udev/45-nkcan.rules /etc/udev/rules.d
Info: Copied 45-nkcan.rules to /etc/udev/rules.d.
cp -f udev/blacklist-nk.conf /etc/modprobe.d
chmod 644 /etc/modprobe.d/blacklist-nk.conf
Info: mainline drivers removed and blacklisted in
/etc/modprobe.d/blacklist-nk.conf
udevadm control --reload-rules
- manually installing nkcan under /lib/modules/4.15.0-142-generic/misc...
mkdir -p /lib/modules/4.15.0-142-generic/misc
cp -f nkcan.ko /lib/modules/4.15.0-142-generic/misc/nkcan.ko
- Building dependencies...
depmod 4.15.0-142-generic
demo@demo:~/nkcdriver_V1.0$ sudo modprobe nkcan
```

5. 驱动安装完成可以查看驱动是否安装正确，在终端输入：lsmod

```
demo@demo:~/nkcdriver_V1.0$ lsmod
Module              Size  Used by
nkcan                15152  0
nls_iso8859_1        16384  0
uas                  24576  0
usb_storage          69632  1 uas
can_dev              28672  1 nkcan
snd_soc_skl          86016  0
intel_rapl           20480  0
snd_soc_skl_ipc      65536  1 snd_soc_skl
snd_hda_ext_core     24576  1 snd_soc_skl
snd_soc_sst_dsp      32768  1 snd_soc_skl_ipc
snd_soc_sst_ipc      16384  1 snd_soc_skl_ipc
snd_soc_acpi         16384  1 snd_soc_skl
intel_telemetry_pltdrv 20480  0
snd_soc_core         241664  1 snd_soc_skl
intel_punit_ipc      16384  1 intel_telemetry_pltdrv
snd_hda_codec_hdmi   49152  1
snd_compress         20480  1 snd_soc_core
ac97_bus             16384  1 snd_soc_core
intel_telemetry_core  16384  1 intel_telemetry_pltdrv
intel_pmc_ipc        20480  1 intel_telemetry_pltdrv
snd_pcm_dmaengine    16384  1 snd_soc_core
snd_hda_intel        45056  2
```

6. 使用 ifconfig -a 查看 CAN 设备

```
demo@demo:~/nkcdriver_V1.0$ ifconfig -a
can0    Link encap:未指定  硬件地址 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        NOARP  MTU:16  跃点数:1
        接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
        发送数据包:0  错误:0  丢弃:0  过载:0  载波:0
        碰撞:0  发送队列长度:10
        接收字节:0 (0.0 B)  发送字节:0 (0.0 B)

can1    Link encap:未指定  硬件地址 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        NOARP  MTU:16  跃点数:1
        接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
        发送数据包:0  错误:0  丢弃:0  过载:0  载波:0
        碰撞:0  发送队列长度:10
        接收字节:0 (0.0 B)  发送字节:0 (0.0 B)

enp1s0  Link encap:以太网  硬件地址 00:60:17:10:00:d9
        UP BROADCAST MULTICAST  MTU:1500  跃点数:1
        接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
        发送数据包:0  错误:0  丢弃:0  过载:0  载波:0
        碰撞:0  发送队列长度:1000
        接收字节:0 (0.0 B)  发送字节:0 (0.0 B)
```

以上步骤完成，驱动即可安装完成。

3.1.2. 卸载驱动

1. 在终端输入：sudo make install

```
demo@demo:~/nkcandriver_V1.0$ sudo make uninstall
rmmod nkcan 2> /dev/null || true
if [ -f /lib/modules/4.15.0-142-generic/misc/nkcan.ko ]; then\
rm -f /lib/modules/4.15.0-142-generic/misc/nkcan.ko;\
depmod 4.15.0-142-generic;\
fi
rm -f /usr/src/nk-linux-driver-8.15.2 \
/tmp/nk-linux-driver-8.15.2-dkms.stderr
rm -rf /var/lib/dkms/nk-linux-driver
rm -f /usr/include/nkcan.h \
/usr/include/nkcanfd.h \
/usr/local/bin/nkcan_make_devices \
/usr/local/bin/lsnkcan \
/usr/local/bin/nkcanosdiag.sh \
/etc/udev/rules.d/45-nkcan.rules \
/usr/local/bin/nkcan_usb_minor_check.bash \
/etc/modprobe.d/blacklist-nk.conf \
/etc/modprobe.d/nkcan.conf
```

2. 卸载完成可以查看驱动是否卸载完毕，在终端输入：lsmod

```
demo@demo:~/nkcandriver_V1.0$ lsmod
Module              Size  Used by
nls_iso8859_1       16384  0
uas                 24576  0
usb_storage         69632  1 uas
can_dev             28672  0
snd_soc_skl         86016  0
intel_rapl          20480  0
snd_soc_skl_ipc     65536  1 snd_soc_skl
snd_hda_ext_core    24576  1 snd_soc_skl
snd_soc_sst_dsp     32768  1 snd_soc_skl_ipc
snd_soc_sst_ipc     16384  1 snd_soc_skl_ipc
snd_soc_acpi        16384  1 snd_soc_skl
intel_telemetry_pltdrv 20480  0
snd_soc_core        241664  1 snd_soc_skl
intel_punit_ipc     16384  1 intel_telemetry_pltdrv
snd_hda_codec_hdmi  49152  1
snd_compress        20480  1 snd_soc_core
ac97_bus            16384  1 snd_soc_core
intel_telemetry_core 16384  1 intel_telemetry_pltdrv
intel_pmc_ipc       20480  1 intel_telemetry_pltdrv
snd_pcm_dmaengine   16384  1 snd_soc_core
snd_hda_intel       45056  4
```

可以看到没有 nkcan 驱动设备。

3. 使用 ifconfig -a 查看 CAN 设备

```
demo@demo:~/nkcandriver_V1.0$ ifconfig -a
enp1s0: Link encap:以太网 硬件地址 00:60:17:10:00:d9
UP BROADCAST MULTICAST MTU:1500 跃点数:1
  接收数据包:0 错误:0 丢弃:0 过载:0 帧数:0
  发送数据包:0 错误:0 丢弃:0 过载:0 载波:0
  碰撞:0 发送队列长度:1000
  接收字节:0 (0.0 B) 发送字节:0 (0.0 B)
  Memory:91200000-9121ffff
enp2s0: Link encap:以太网 硬件地址 00:60:17:10:00:da
UP BROADCAST MULTICAST MTU:1500 跃点数:1
  接收数据包:0 错误:0 丢弃:0 过载:0 帧数:0
  发送数据包:0 错误:0 丢弃:0 过载:0 载波:0
  碰撞:0 发送队列长度:1000
  接收字节:0 (0.0 B) 发送字节:0 (0.0 B)
  Memory:91100000-9111ffff
lo: Link encap:本地环回
inet 地址:127.0.0.1 掩码:255.0.0.0
inet6 地址: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 跃点数:1
  接收数据包:29748 错误:0 丢弃:0 过载:0 帧数:0
  发送数据包:29748 错误:0 丢弃:0 过载:0 载波:0
  碰撞:0 发送队列长度:1000
```

可以看到没有 can 设备。

3.1.3. 配置 CAN 总线

下面的示例将“can0”配置为比特率 500000。

1.查看 CAN 设备

在命令终端中输入：

```
ifconfig -a
```

得到如下结果：

```
demo@demo:~$ ifconfig -a
can0      Link encap:未指定  硬件地址 00-00-00-00-00-00-00-00-00-00-00-00-
00-00
          NOARP  MTU:16  跃点数:1
          接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
          发送数据包:0  错误:0  丢弃:0  过载:0  载波:0
          碰撞:0  发送队列长度:10
          接收字节:0 (0.0 B)  发送字节:0 (0.0 B)

can1      Link encap:未指定  硬件地址 00-00-00-00-00-00-00-00-00-00-00-
00-00
          NOARP  MTU:16  跃点数:1
          接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
          发送数据包:0  错误:0  丢弃:0  过载:0  载波:0
          碰撞:0  发送队列长度:10
          接收字节:0 (0.0 B)  发送字节:0 (0.0 B)

enp1s0    Link encap:以太网  硬件地址 00:60:17:10:00:d9
          UP BROADCAST MULTICAST  MTU:1500  跃点数:1
          接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
          发送数据包:0  错误:0  丢弃:0  过载:0  载波:0
          碰撞:0  发送队列长度:1000
          接收字节:0 (0.0 B)  发送字节:0 (0.0 B)
          Memory:91200000-9121ffff

enp2s0    Link encap:以太网  硬件地址 00:60:17:10:00:da
          UP BROADCAST MULTICAST  MTU:1500  跃点数:1
          接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
          发送数据包:0  错误:0  丢弃:0  过载:0  载波:0
          碰撞:0  发送队列长度:1000
          接收字节:0 (0.0 B)  发送字节:0 (0.0 B)
          Memory:91100000-9111ffff

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  跃点数:1
          接收数据包:420  错误:0  丢弃:0  过载:0  帧数:0
          发送数据包:420  错误:0  丢弃:0  过载:0  载波:0
          碰撞:0  发送队列长度:1000
          接收字节:32656 (32.6 KB)  发送字节:32656 (32.6 KB)
```

如上图 can0, can1 就是我们的 Can 卡设备了。

2.设置 can 设备的波特率

在命令终端中输入：

```
sudo ip link set can0 type can bitrate 500000
```

3.启动 can 设备

在命令终端中输入：

```
sudo ip link set up can0
```

4.关闭 can 设备

在命令终端中输入：

```
sudo ip link set down can0
```

3.1.4. 使用 CAN -utils 操作 CAN 总线

当 can 设备启用后，将可以使用许多实用程序，来进行 can 通信了。首先我们通过在命令终端中输入以下命令来安装 can-utils：

```
sudo apt-get install can-utils
```

➤ candump

candump 可以实时显示接收到的 can 消息。

要实时显示设备 can0 上的所有 can 消息，在命令终端中输入以下命令：

```
candump can0
```

接收到的数据：

```
can0 456 [8] 12 34 56 78 98 76 54 32
```

此条指令将在接口 can0 上接收消息，其 ID 为 0x456，数据长度为 8，数据为[0x12、0x34、0x56、 0x78、0x98、0x76、0x54、0x32]。

➤ cansend

cansend 可以将单个 CAN 帧发送到总线上。您将必须指定设备，标识符和要发送的数据字节。在命令终端输入：

```
cansend can0 123#1122334455667788
```

此条指令将在 can0 接口上发送一条消息，其 ID 为 0x123，数据为[0x11、0x22、0x33、0x44、0x55、0x66、0x77、0x88]。 请注意，此数据以十六进制给出。

➤ cangen

cangen 可以生成随机的 CAN 数据，这对于测试很有用。

有关更多的用法信息，请在命令终端中输入：

```
cangen -help
```

➤ cansniffer

cansniffer 可以显示总线上接收到的 CAN 消息，而且可以过滤掉数据不变的帧。这对于逆向工程 CAN 总线系统非常有用。有关更多信息，请在命令终端中输入：

```
cansniffer -help
```

3.1.5. 例程

CAN 总线卡支持 Linux 系统的 socket-can 接口，以下代码段是 SocketCAN API 的工作

示例，该 API 使用原始接口发送和接收 CAN1、CAN2 的数据包：

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <net/if.h>
#include <sys/ioctl.h>
#include <sys/socket.h>
#include <linux/can.h>
```

```
#include <linux/can/raw.h>
#include <string.h>
#define commde "echo 123456 |sudo -S ip link set can0 type can bitrate 500000"
#define up "sudo ip link set up can0"
#define commde1 "echo |sudo -S ip link set can1 type can bitrate 500000"
#define up1 "sudo ip link set up can1"
int main(void)
{
    system(commde);
    system(up);
    system(commde1);
    system(up1);
    int nbytes = 0 ;
    struct sockaddr_can sockCan0,sockCan1;
    const char *ifname = "can0";
    const char *ifname1 = "can1";
    socklen_t len = 0;
    struct ifreq ifr;
    struct can_frame frame[2] = {{0},{0}};
    int s = socket(PF_CAN,SOCK_RAW,CAN_RAW);
    struct sockaddr_can addr;
    addr.can_family = AF_CAN;
    addr.can_ifindex = 0;
    bind(s,(struct sockaddr *)&addr,sizeof(addr));

    strcpy(ifr.ifr_name,ifname);
    ioctl(s,SIOCGIFINDEX,&ifr);
    sockCan0.can_family = AF_CAN;
    sockCan0.can_ifindex = ifr.ifr_ifindex;

    strcpy(ifr.ifr_name,ifname1);
    ioctl(s,SIOCGIFINDEX,&ifr);
    sockCan1.can_family = AF_CAN;
    sockCan1.can_ifindex = ifr.ifr_ifindex;
    setsockopt(s,SOL_CAN_RAW,CAN_RAW_FILTER,NULL,0);
    struct can_filter rfilter[2];
    /*rfilter[0].can_id = 0x123;
    rfilter[0].can_mask = CAN_EFF_MASK;
    rfilter[1].can_id = 0x124;
    rfilter[1].can_mask = CAN_EFF_MASK;*/
    setsockopt(s,SOL_CAN_RAW,CAN_RAW_FILTER,rfilter,sizeof(rfilter));
```

```
frame[0].can_id = 3;
frame[0].can_dlc = 8;
frame[0].data[0] = 0x40;
frame[0].data[1] = 0x01;
frame[0].data[2] = 0x10;
frame[0].data[3] = 0x00;
frame[0].data[4] = 0x00;
frame[0].data[5] = 0x00;
frame[0].data[6] = 0x00;
frame[0].data[7] = 0x00;

frame[1].can_id = 123;
frame[1].can_dlc = 8;
frame[1].data[0] = 0x40;
frame[1].data[1] = 0x01;
frame[1].data[2] = 0x10;
frame[1].data[3] = 0x00;
frame[1].data[4] = 0x00;
frame[1].data[5] = 0x00;
frame[1].data[6] = 0x00;
frame[1].data[7] = 0x00;
for(int c=0; c<300; c++)
{
    nbytes = sendto(s,&frame[0],sizeof(struct can_frame),0,(struct
sockaddr*)&sockCan0,sizeof(sockCan0));
    frame[0].can_id++;
    usleep(2000);
    printf("%s Wrote %d bytes\n",ifname,nbytes);
    if(nbytes < sizeof(struct can_frame))
    {
        printf("Error:sendto can0\n");
    }
    nbytes = 0;
    nbytes = sendto(s,&frame[1],sizeof(struct can_frame),0,(struct
sockaddr*)&sockCan1,sizeof(sockCan1));
    frame[1].can_id++;
    usleep(1000);
    printf("%s Wrote %d bytes\n",ifname1,nbytes);
    if(nbytes < sizeof(struct can_frame))
    {
        printf("Error:sendto can1\n");
    }
}
```

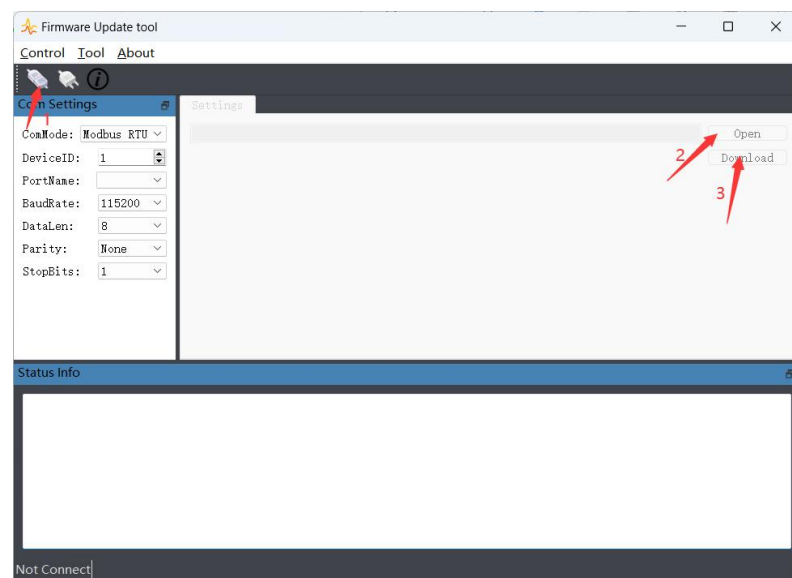
```
while(1)
{
    for(int j=0; j<2; j++)
    {
        recvfrom(s,&frame[j],sizeof(struct can_frame),0,(struct
sockaddr*)&addr,&len);
        printf("Receive OK\n");
        printf("ID=0x%X          DLC=%d          Data:
",frame[j].can_id,frame[j].can_dlc);
        for(int i = 0; i<frame[j].can_dlc; ++i)
        {
            printf("0x%02X ",frame[j].data[i]);
        }
        printf("\n");
    }
}
close(s);
return 0;
```

4. 升级

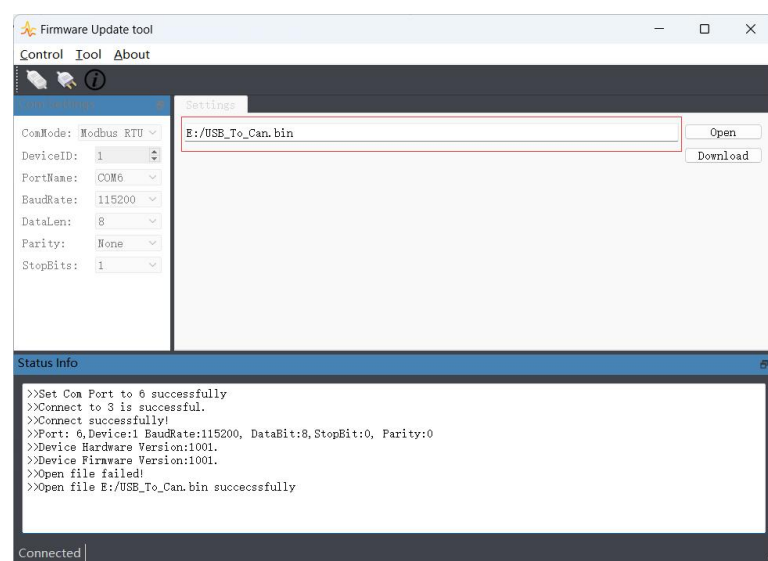
4.1. 升级软件

使用本公司自带的升级软件 NKMcuFwUpdateTool 进行升级，支持 windows10，windows11。

使用方法：



升级文件为 bin 文件。



4.2. 升级步骤

当 CAN0 发送扩展帧，ID 为 0x1fffffe，数据为 0xffffffffffffee，发送该数据，则进入升级，升级需要重新拔插 USB 以便识别串口，升级所需要的文件均为 bin 文件。在使用升级软件升级过后，重新拔插 USB 即可升级完毕。