WCHISPTool_CMD Command Line Programming Tool Instruction

1. Introduction

1.1 Software function

WCHISPTool_CMD is a multi-platform command line tool for WCH MCU burning online, which supports firmware download, verification and other operations for series MCU of WCH through USB or serial port. The tool contains ISP library and sample programs for customized development of ISP tool.

Supported operating system: Windows x86/x64, Linux x64, macOS x64/arm64.

Supported chip model: CH54x/CH55x/CH56x/CH643x/CH57x/CH58x/CH59x/CH32F10x/CH3F20x/CH32V00x/ CH32V10x/CH32V20x/CH32V30x/ CH32X03x/CH32L10x.

1.2 Command line

1.2.1 Download

USB mode:

```
sudo ./WCHISPTool_CMD -p /dev/ch37x -c Config.ini -o program -f Target.hex
```

Serial port mode:

```
sudo ./WCHISPTool_CMD -p /dev/ttyISP0 -b 115200 -c Config.ini -o program -f Target.hex
```

1.2.2 Verify

USB mode:

```
sudo ./WCHISPTool_CMD -p /dev/ch37x -c Config.ini -o verify -f Target.hex
```

Serial port mode:

```
sudo ./WCHISPTool_CMD -p /dev/ttyISP0 -b 115200 -c Config.ini -o verify -f Target.hex
```

1.2.3 Parameter Description

Instruction description	Parameters description
-p USB ISP device or serial device node	**/dev/ch37x** Download via USB in Linux **/dev/ttyISPx** Download via serial port in Linux **LocationID** Download via USB in macOS **COM(/dev/tty.*)** Download via serial port in macOS
-b Communication baud rate of serial port	**115200/230400/** Communication baud rate of serial port
-v Print version number	**boot/tool** The boot/tool version
-c The full path name of the configure file	**xxx.ini** The full/relative path
-o The type of operation	**program/verify** Download/Verify
-f The name of Flash file	**xxx.hex/xxx.bin** The full/relative path
-r Disable code protection	Disable code protection before downloading

Notes:

① All commands and parameters must appear in pairs in the format "-x xxx".

② Download or verify operation is required to pass in -p,-c,-o,-f instructions.

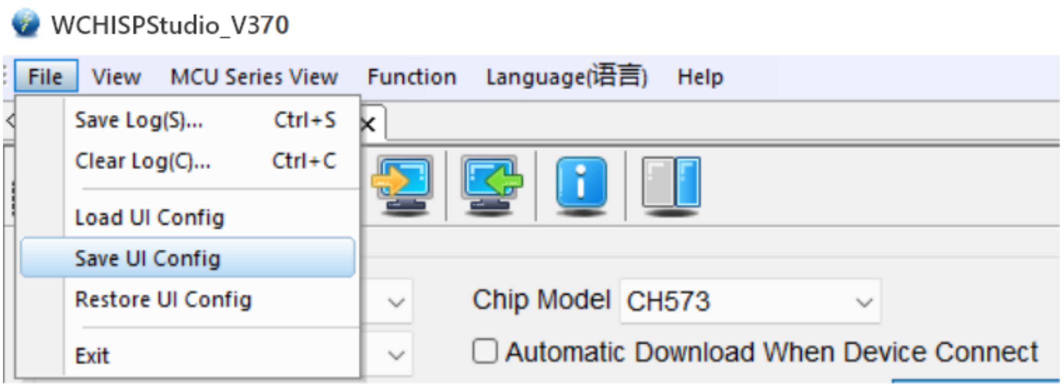
③ For the confirmation method of the USB ISP device or serial port node name corresponding to the - p command, see subsections 3.1 and 4.1.

1.3 Status code

Instruction description	Parameters description
0	Execute successfully
1	Invalid input parameter
2	Failed to get parameters from the configuration file
3	Failed to set ISP parameters
4	The specified serial port name is invalid
5	No device enumerated
6	The specified chip type is not consistent with the actual chip type
7	Failed to get the device information
8	Invalid Flash file path
9	Invalid Flash file length
10	Failed to read the Flash file
11	Failed to convert the Flash file from HEX to BIN format
12	Failed to disable read protection
13	Failed to download
14	Failed to verify
15	Failed to disable write protection
100	Unknown error

1.4 Configuration file

The configuration file is generated by the "Save UI Config" function of WchIspStudio.exe in Windows. The specific operations are as follows: First, open the software and select MCU series on the right side of the software. Next, select series and model of the chip in the interface of “Chip Option” and configure the chip in the interface of “Download Config”. Then click the main menu "File ->Save UI Config". Finally select the name and location of the configuration file. The operation interface is as follows.



1.5 Custom development

The src directory in each system folder contains the source files of the command line burning tool, which can be directly developed based on this code to meet the customized demand of users. The lib directory contains the ISP development dynamic library and header files. For functions and call instructions, users could refer to WCH55XISPDLL. H and other header files in the lib directory.

2. Windows Platform

For details, please refer to: https://www.wch.cn/downloads/WCHISPTool_Setup_exe.html install path \ WCHISPTool_XXX\Doc.

3. Linux Platform

3.1 Instruction

3.1.1 USB download mode

① Plug USB
Make sure MCU is in BOOT download mode, and PID of USB device is 0x55e0.

② Install USB device driver
Open the system terminal, enter the driver folder, and execute the "make install" command.
This operation is only required for the first download.

③ Determine USB ISP device name
Use the "ls" command to check whether the /dev/ch37x character device exists.

④ Execute download instruction
Execute according to the instruction format requirements of the tool, for example,

```
sudo ./WCHISPTool_CMD -p /dev/ch37x0 -c Config.ini -o program -f Target.hex
```

3.1.2 Serial port download mode

① Connect MCU with serial port
Ensure MCU is in BOOT download mode.

② Create serial ISP device name
Confirm the serial port device node name, and then use the "ln" command to create a soft link named "ttyISPx" for this device. The specific command is as follows.

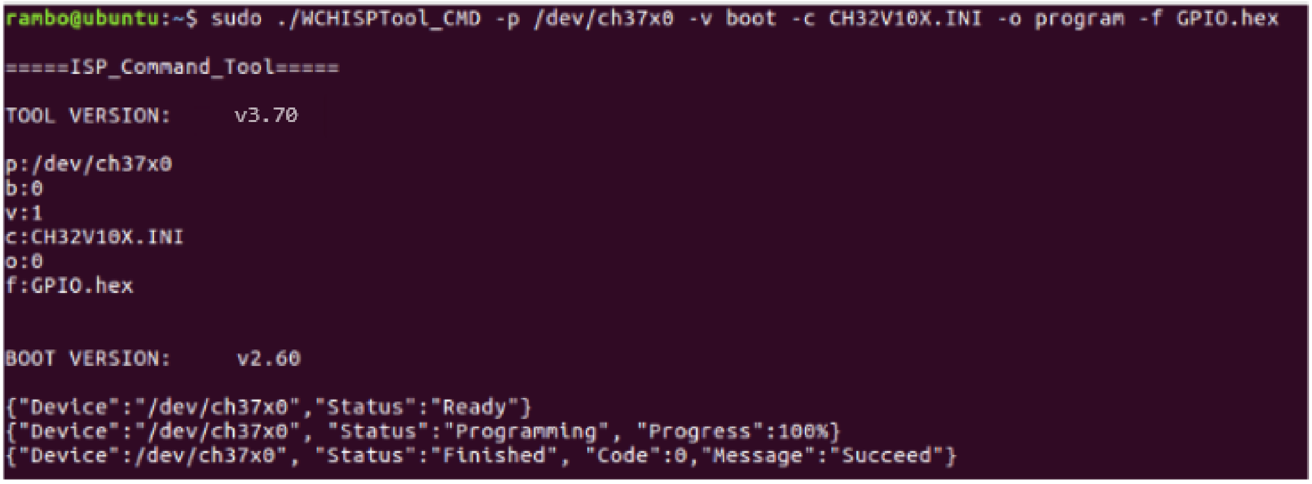
```
sudo ln –s /dev/ttyUSB0 /dev/ttyISP0
```

③ Execute download instruction.
Execute according to the instruction format requirements of the tool, for example,

sudo ./WCHISPTool_CMD -p /dev/ttyISP0 –b 115200 -c Config.ini -o program -f Target.hex

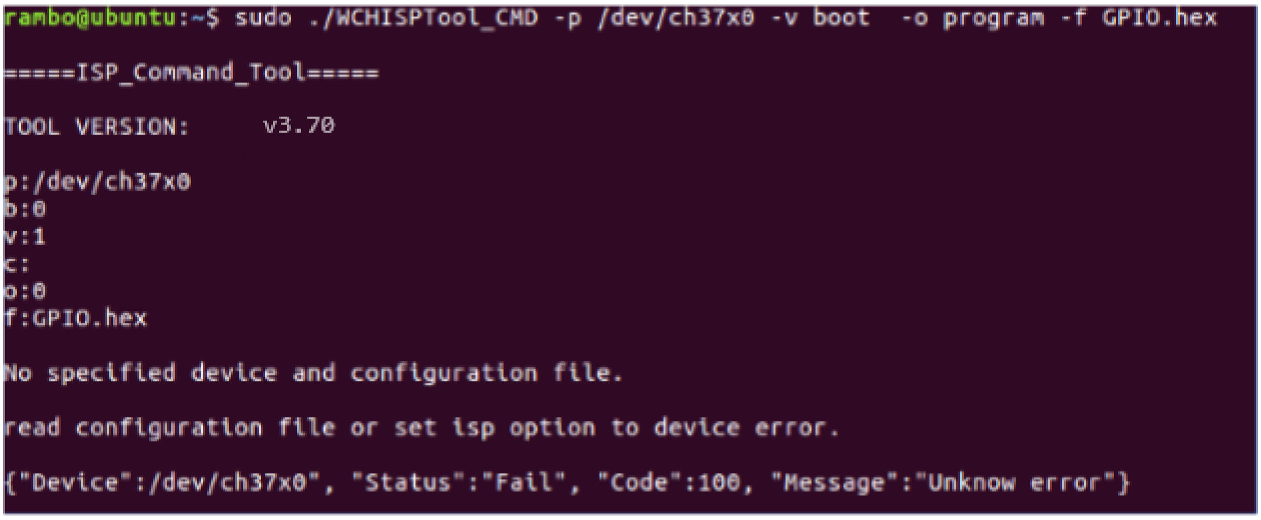
3.2 Run log file

3.2.1 The instance of successful download operation



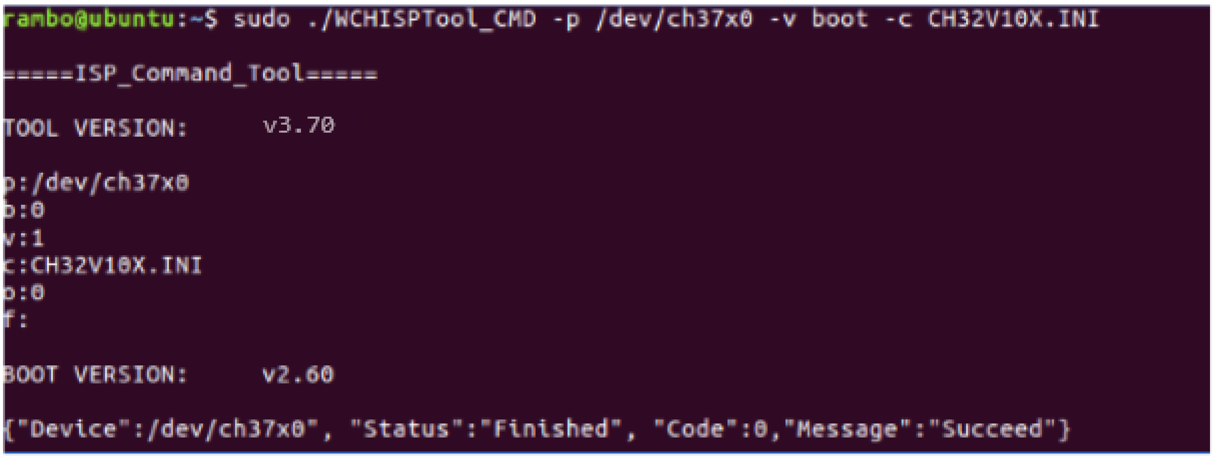
```
rambo@ubuntu:~$ sudo ./WCHISPTool_CMD -p /dev/ch37x0 -v boot -c CH32V10X.INI -o program -f GPIO.hex

=====ISP_Command_Tool=====

TOOL VERSION:     v3.70

p:/dev/ch37x0
b:0
v:1
c:CH32V10X.INI
o:0
f:GPIO.hex


BOOT VERSION:     v2.60

{"Device":"/dev/ch37x0","Status":"Ready"}
{"Device":"/dev/ch37x0", "Status":"Programming", "Progress":100%}
{"Device":/dev/ch37x0", "Status":"Finished", "Code":0,"Message":"Succeed"}
```

3.2.2 The instance of failed download operation



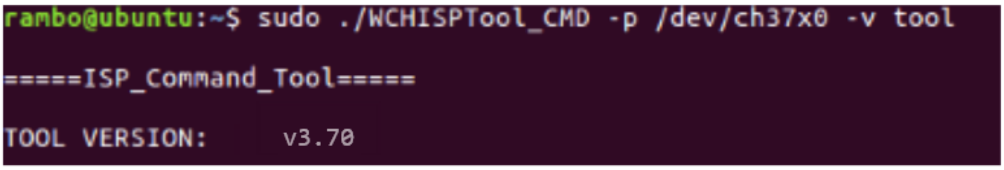
```
rambo@ubuntu:~$ sudo ./WCHISPTool_CMD -p /dev/ch37x0 -v boot  -o program -f GPIO.hex

=====ISP_Command_Tool=====

TOOL VERSION:     v3.70

p:/dev/ch37x0
b:0
v:1
c:
o:0
f:GPIO.hex

No specified device and configuration file.

read configuration file or set isp option to device error.

{"Device":/dev/ch37x0", "Status":"Fail", "Code":100, "Message":"Unknow error"}
```

3.2.3 The instance of getting the BOOT version separately



```
rambo@ubuntu:~$ sudo ./WCHISPTool_CMD -p /dev/ch37x0 -v boot -c CH32V10X.INI

=====ISP_Command_Tool=====

TOOL VERSION:     v3.70

p:/dev/ch37x0
b:0
v:1
c:CH32V10X.INI
o:0
f:

BOOT VERSION:     v2.60

{"Device":/dev/ch37x0", "Status":"Finished", "Code":0,"Message":"Succeed"}
```

3.2.4 The instance of getting the software version separately



```
rambo@ubuntu:~$ sudo ./WCHISPTool_CMD -p /dev/ch37x0 -v tool
=====ISP_Command_Tool=====
TOOL VERSION:      v3.70
```

4. macOS Platform

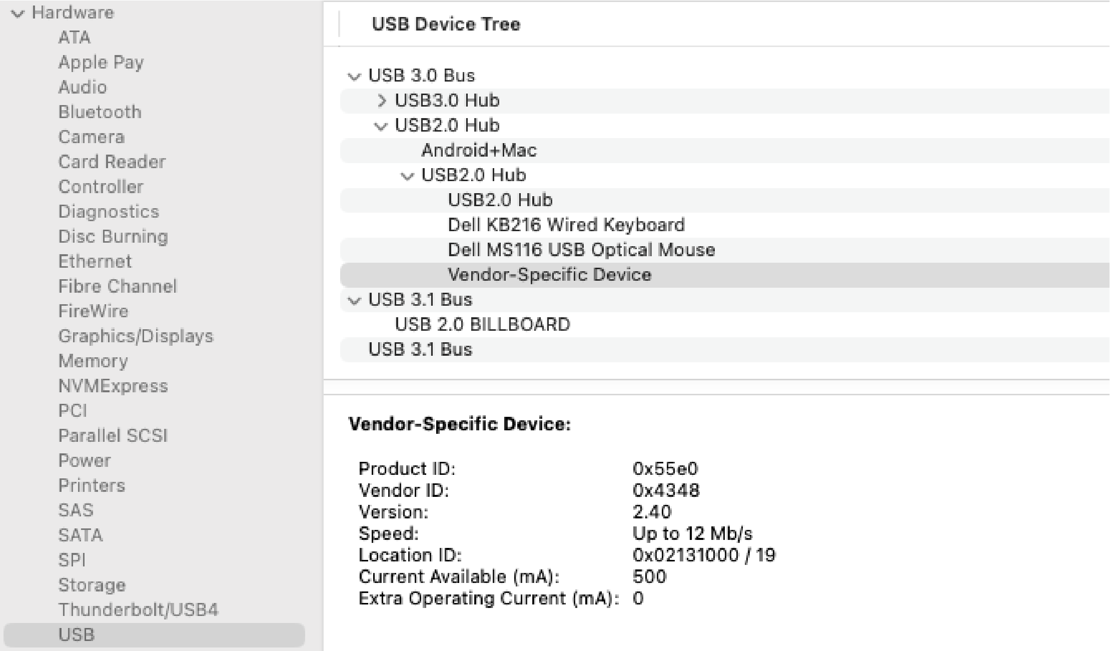
4.1 Instruction

4.1.1 USB download mode

① Plug USB

Make sure MCU is in BOOT download mode, and PID of USB device is 0x55e0.

② Determine the location ID of the USB device in the macOS system. Find the device in the System Report ->Hardware ->USB. The location ID in the USB device tree is shown in the following figure.



Vendor-Specific Device:

```
Product ID:                      0x55e0
Vendor ID:                       0x4348
Version:                         2.40
Speed:                           Up to 12 Mb/s
Location ID:                     0x02131000 / 19
Current Available (mA):          500
Extra Operating Current (mA):    0
```

③ Execute download instruction

Execute according to the instruction format requirements of the tool, for example,

sudo ./WCHISPTool_CMD -p 0x02131000 -c Config.ini -o program -f Target.hex

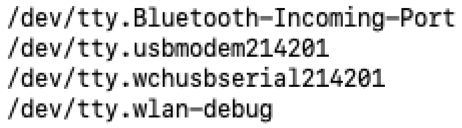
4.1.2 Serial port download mode

① Connect MCU with serial port

Ensure MCU is in BOOT download mode.

② Determine the node name of the serial port on the device, and run the "ls /dev/tty.*" command on the terminal

to check the serial port on the macOS(If the WCH serial port chip is used, install the CH34xVCPDriver of the macOS). A specific operation is shown in the figure below.



```
/dev/tty.Bluetooth-Incoming-Port
/dev/tty.usbmodem214201
/dev/tty.wchusbserial214201
/dev/tty.wlan-debug
```

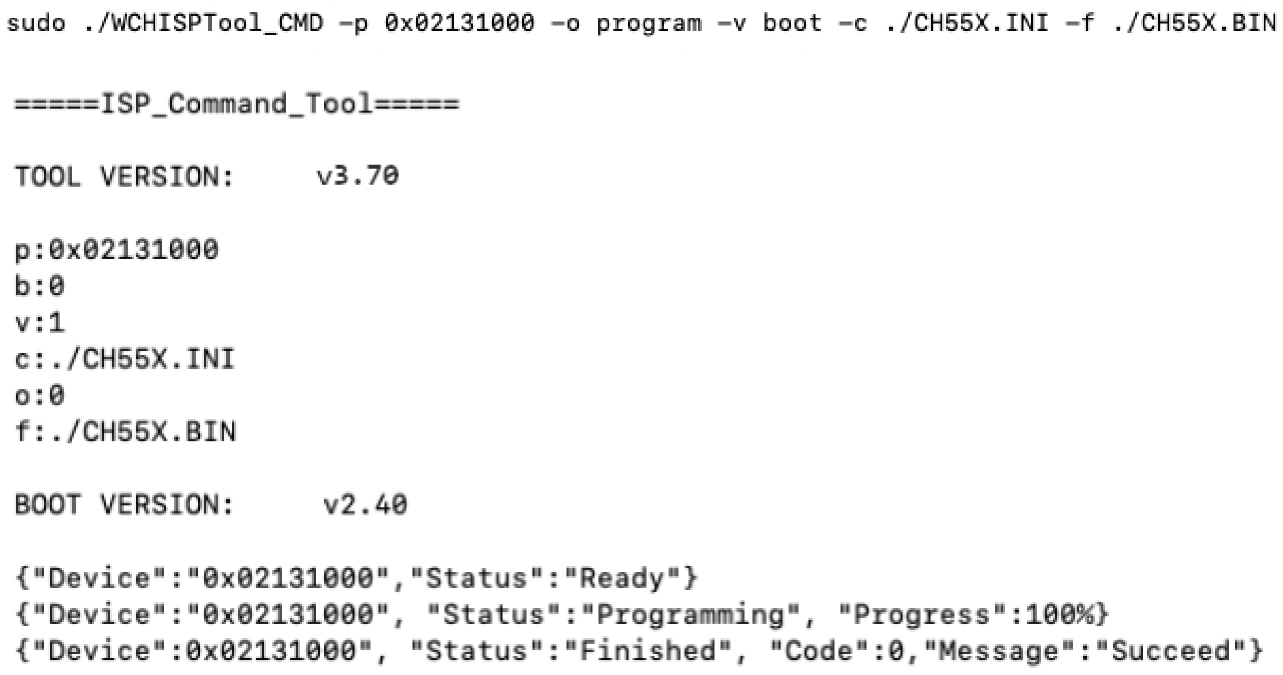
③ Execute download instruction

Execute according to the instruction format requirements of the tool, for example,

sudo ./WCHISPTool_CMD -p tty.wchusbserial214201–b 115200 -c Config.ini -o program -f Target.hex

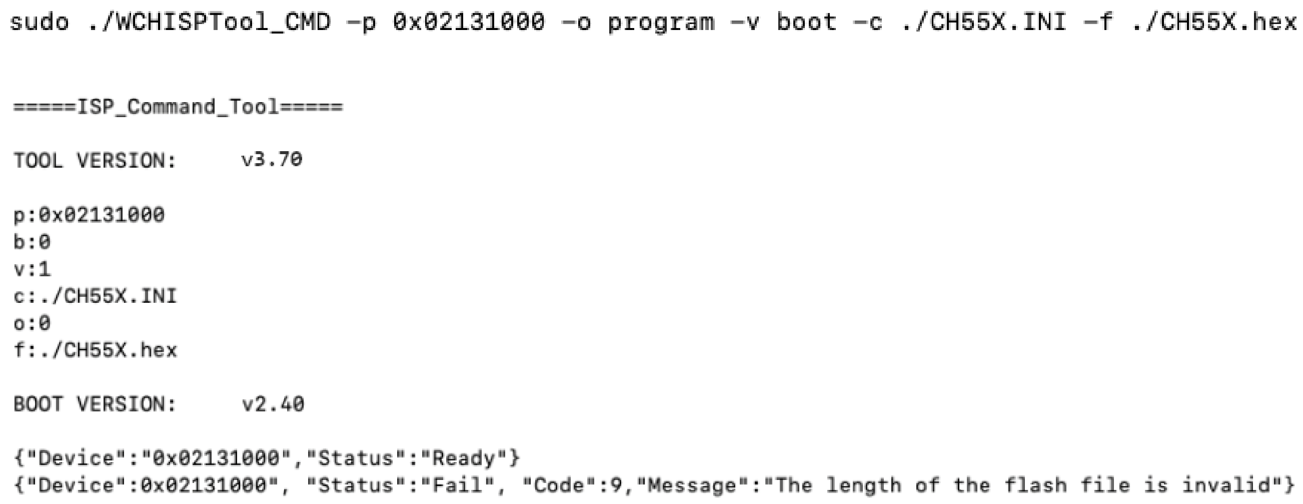
4.2 Run log file

4.2.1 The instance of successful download operation



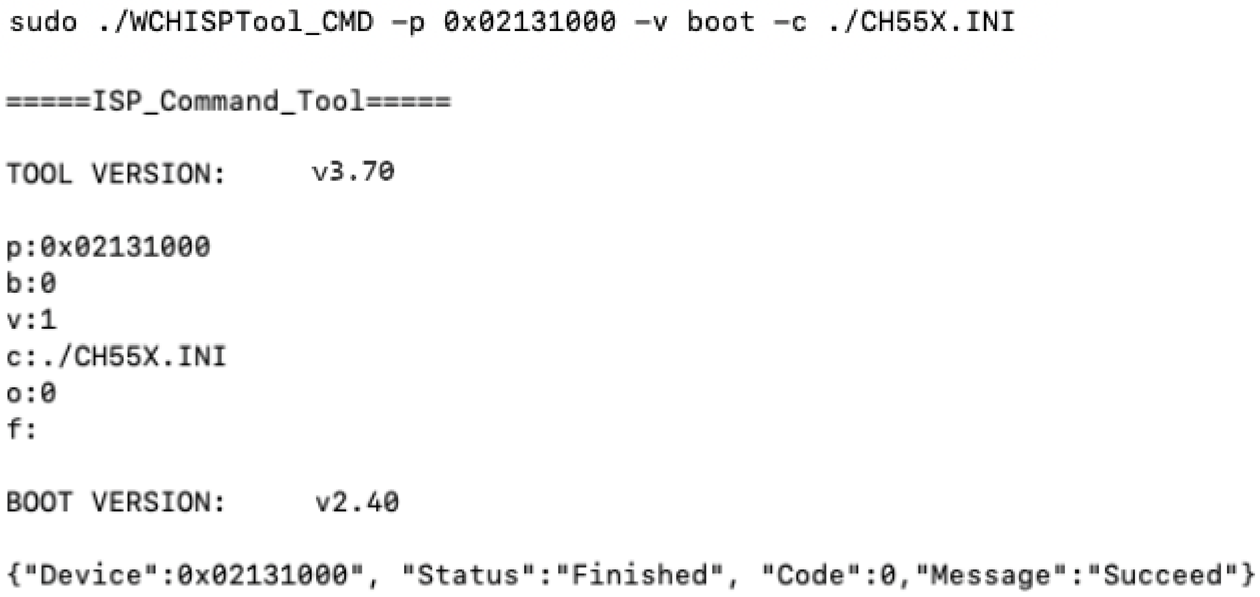
```
sudo ./WCHISPTool_CMD -p 0x02131000 -o program -v boot -c ./CH55X.INI -f ./CH55X.BIN

=====ISP_Command_Tool=====

TOOL VERSION:     v3.70

p:0x02131000
b:0
v:1
c:./CH55X.INI
o:0
f:./CH55X.BIN

BOOT VERSION:     v2.40

{"Device":"0x02131000","Status":"Ready"}
{"Device":"0x02131000", "Status":"Programming", "Progress":100%}
{"Device":0x02131000", "Status":"Finished", "Code":0,"Message":"Succeed"}
```

4.2.2 The instance of failed download operation



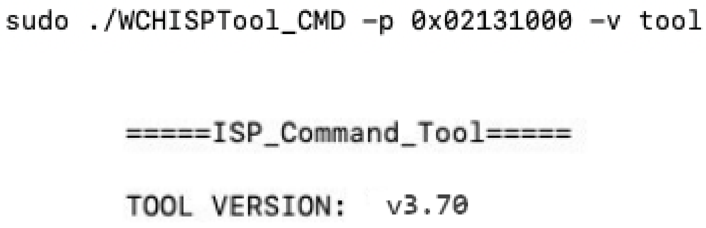
```
sudo ./WCHISPTool_CMD -p 0x02131000 -o program -v boot -c ./CH55X.INI -f ./CH55X.hex

=====ISP_Command_Tool=====

TOOL VERSION:     v3.70

p:0x02131000
b:0
v:1
c:./CH55X.INI
o:0
f:./CH55X.hex

BOOT VERSION:     v2.40

{"Device":"0x02131000","Status":"Ready"}
{"Device":0x02131000", "Status":"Fail", "Code":9,"Message":"The length of the flash file is invalid"}
```

4.2.3 The instance of getting the BOOT version separately



```
sudo ./WCHISPTool_CMD -p 0x02131000 -v boot -c ./CH55X.INI

=====ISP_Command_Tool=====

TOOL VERSION:     v3.70

p:0x02131000
b:0
v:1
c:./CH55X.INI
o:0
f:

BOOT VERSION:     v2.40

{"Device":0x02131000", "Status":"Finished", "Code":0,"Message":"Succeed"}
```

4.2.4 The instance of getting the software version separately



```
sudo ./WCHISPTool_CMD -p 0x02131000 -v tool


=====ISP_Command_Tool=====

TOOL VERSION:   v3.70
```