Flash读取操作指导

版本1.0.0

启明云端

目录

前言

本文档介绍了关于flash数据读取工具的功能介绍、安装使用、命令说明以及flash数据读取的具体操作方法。

1. 工具介绍

esptool.py是乐鑫提供的开源库工具，用于乐鑫ESP8285、ESP8266、ESP32、ESP32-S等系列芯片和ROM Bootloader（即：一级bootloader）通讯，从而实现：固件烧录，flash擦除，flash读取，读MAC地址，读flash id，elf文件转bin等常用功能；flash校验,读取内存，载入bin到RAM执行，读内存，写内存，读flash状态，写flash状态，读chip id，组装bin等高级功能。

2. 安装方法

2.1. 方法一

2.1.1. 下载esptool.py源码

git clone https://github.com/espressif/esptool.git

2.1.2. 目录导入到全局环境变量

以ubuntu为例：

1、将export PATH=/home/username/esp/esptool:$PATH 添加到/etc/profile文件结尾。

2、执行 source /etc/profile安装成功后可以通过esptool.py version查看版本。

注意： esptool.py v3.0版本后，才对ESP32-S系列支持。

如果版本过低可以通过git pull来更新esptool.py版本。

2.2. 方法二（最简单）

通过shell命令安装，以下命令人选其一，执行成功即可：

1、pip install esptool

2、python -m pip install esptool

3、pip2 install esptool

2.3. 方法三（esp-idf自带）

如果已经拥有了esp-idf环境的前提下，可以直接到esp-idf目录下的esptool文件夹来使用esptool工具。

具体位置：esp-idf/components/esptool_py/esptool

保证当前目录下包含esptool.py脚本即可。

3. esptool.py说明

通用参数说明(命令前参数)

可通过esptool.py -h查看所有命令和通用参数详细说明。

-h或--help：显示帮助文档

--chip或-c：指定芯片，可选auto、esp8266、esp32、esp32s2

--port或-p：指定串口

--baud或-b：指定波特率

--before：指定esptool.py命令执行前预做的，可选default_reset、no_reset、no_reset_no_sync，具体参考文档

--after或-a：指定esptool.py命令执行后将做的，可选hard_reset、soft_reset、no_reset具体参考文档

--no-stub：禁用Boot Stub,不让其管理flash操作，具体参考文档

--trace或-t：打开esptool.py所有交互细节

--override-vddsdio：VDDSDIO内部电压调节

--connect-attempts：指定esptool.py尝试连接次数，默认7

4. 常用命令

4.1. Flash读取——read_flash

4.1.1. 参数说明

--spi-connection或-sc：指定ESP32SPI/HSPI连接配置

--no-progress或-p：禁用进度条打印

4.1.2. 命令参考用法

esptool.py read_flash [-h] [--spi-connection SPI_CONNECTION][--no -progress] address size filename

4.1.3. 示例1：自动读取

读取从0x0地址开始的4KB内容，保存到dump.bin文件

esptool.py read_flash 0x0 0x1000 dump.bin

4.1.4. 示例2：指定参数读取

在ubuntu下，指定串口/dev/ttyUSB1,波特率460800,从0x10000地址读取1MB内容到dump.bin文件

esptool.py -p /dev/ttyUSB1 -b 460800 read_flash 0x10000 0x100000 dump.bin

4.2. 固件写入——write_flash

4.2.1. 参数说明

--erase-all或-e：在写固件时，擦除所有flash上所有sector(默认只擦除要 写区域的sector)

--flash_freq或-ff：可选keep、40m、26m、20m、80m,指定SPI速率

--flash_mode或-fm：可选keep、qio、qout、dio、dout,指定SPI模式

--flash_size或-fs：可选1MB、2MB,4MB、8MB、16M 指定flash大小

--spi-connection或-sc：指定ESP32SPI/HSPI连接配置

--no-progress或-p：禁用进度条打印

--verify：在flash上验证刚刚写入的数据

--encrypt：写入数据时应用flash加密（需要正确的efuse设置）

--ignore-flash-encryption-efuse-setting：忽略flash加密的efuse设置

--compress：传输中压缩数据（默认--no-stub未指定）

--no-compress：传输中禁用压缩数据（默认--no-stub已指定）

4.2.2. 命令参考用法

esptool.py write_flash [-h] [--erase-all]
[--flash_freq{keep,40m,26m,20m,80m}] [--flash_mode {keep,qio,qout,dio,dout}]
[--flash_sizeFLASH_SIZE][--spi-connection SPI_CONNECTION]
[--no-progress][--verify][--encrypt][--ignore-flash-encryption-efuse-setting]
[--compress | --no-compress]<address> <filename>[<address> <filename>...]

4.2.3. 示例一：自动写入

向flash的0x0地址烧录factory.bin文件

esptool.py write_flash 0x0 factory.bin

4.2.4. 指定参数烧录

指定芯片ESP32，串口/dev/ttyUSB0

0x0地址烧录bootloader.bin,

0x10000地址烧录app.bin,

0x8000地址烧录partitions.bin

esptool.py -c esp32 -p /dev/ttyUSB0 -b 115200 write_flash 0x0
build/bootloader/bootloader.bin 0x10000 build/app.bin 0x8000 build/partitions.bin