

文档版本	
发布日期	

# TG7100C 开发板用户手册



芯片开放社区  
Open Chip Community

# 开发板 用户手册

版本：1.0

版权 @ 2020

1	环境配置	3
2	代码编译	5
3	下载运行	6
3.1	芯片选择	6
3.2	配置程序下载方式	7
3.3	配置下载参数	8
3.4	下载程序	9

aos-cube 是 AliOS Things 基于命令行的开发管理工具，主要功能包括：工程配置与编译、Image 下载调试、组件生成、组件安装管理、设备管理、代码审查、OTA 工具等功能。它依赖于 Python（64bits，2.7.14 和 3.5 已验证）。Linux 下 AliOS-Things 开发环境的搭建主要分为两部分：python 和 pip 安装、基于 pip 安装 aos-cube 及相关的依赖包。

#### 1. 安装 python、pip 和 git:

```
$ sudo apt-get install -y python python-pip git
```

# 完成 python 和 pip 安装后，再安装依赖库和 aos-cube，步骤如下：

```
$ python -m pip install setuptools wheel aos-cube
```

---

注解：如果在安装过程中遇到网络问题可以使用国内镜像。

### 安装/升级 pip

```
$ python -m pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/ --upgrade pip
```

### 基于 pip 依次安装第三方包和 aos-cube

```
$ pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/ setuptools
```

```
$ pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/ wheel
```

```
$ pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/ aos-cube
```

### 如需要使用 doubanio 作备用源

```
$ pip install --trusted-host pypi.doubanio.com -i http://pypi.doubanio.com/simple/ aos-cube
```

### 如需指定版本，可改成如 aos-cube==0.2.50

---

因涉及多种开发环境和具体版本的依赖，针对开发者的实际情况，还给出一种简单方便且不影响当前系统环境的方法——基于虚拟环境 virtualenv 的方法，作为备用。

2. python 和 python-pip 安装完成后，再执行以下步骤：

```
# pip 安装 virtualenv, 如安装成功，可以在 pip list 查看版本
```

```
$ python -m pip install virtualenv
```

```
# 如遇网络问题，可使用阿里云镜像
```

```
$ pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/ virtualenv
```

```
# 建立虚拟环境，如安装到 ~/venv(可更改)
```

```
$ virtualenv ~/venv --no-download
```

```
# 启动虚拟环境，下面执行使用'.'
```

```
$ . ~/venv/bin/activate
```

```
# 安装 aos-cube 和必要的依赖包
```

```
$ 请参考上文 aos-cube 及相关依赖安装
```

```
# 退出虚拟环境
```

```
$ deactivate
```

详见链接：[https://help.aliyun.com/document\\_detail/161037.html](https://help.aliyun.com/document_detail/161037.html)

进入代码的根目录，执行命令：`./build.sh example smart_outlet tg7100c SINGAPORE ONLINE 1`

TG Flash Environment 提供用户下载程序的功能，并且支持时钟、flas 等参数配置，用户可根据自身需求决定是否对程序进行加密、添加签名、更换程序启动时的信息文件、用户资源文件、分区表等功能配置。

具体的功能如下：

1. 支持 IOT 应用程序和 MCU 应用程序的下载
2. 下载通讯接口支持 UART 和 JLink 两种方式

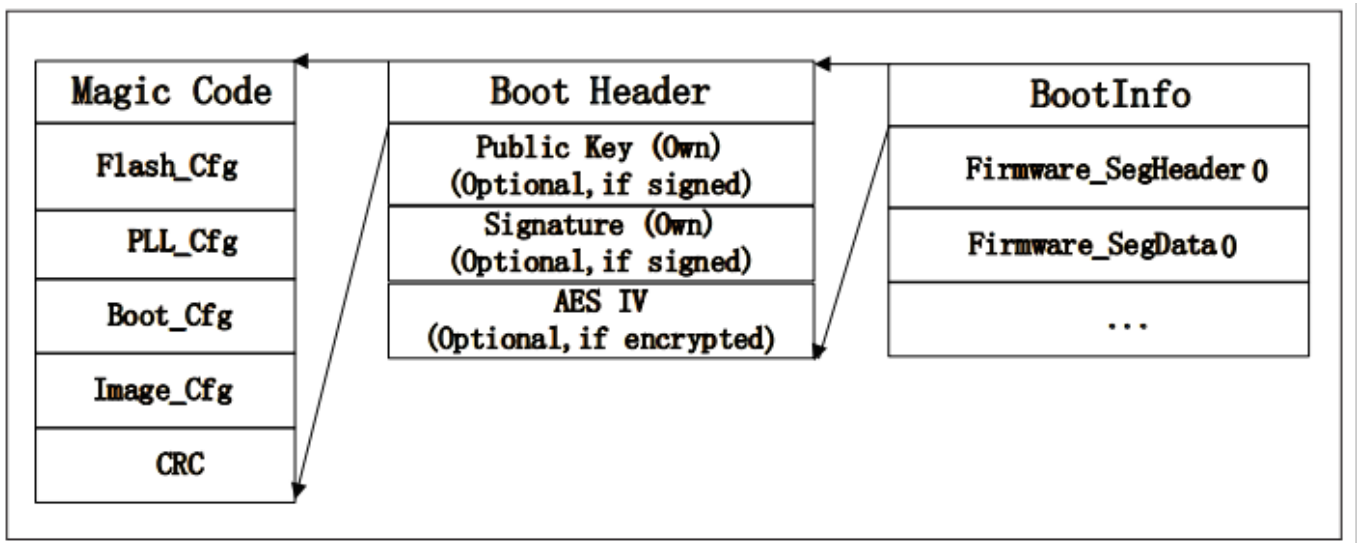


图 3.1: 下载内容布局

如果只下载应用程序无法使芯片正常工作，必须要将引导信息下载到指定位置。引导信息包含对 PLL、Boot、Flash 等的配置；固件是用户自己编写的应用程序。

### 3.1 芯片选择

进入下载工具目录，执行 `python3 TGFlashEnv.py`，在 Chip Selection 对话框中选择对应的芯片型号，点击 Finish 进入主界面。

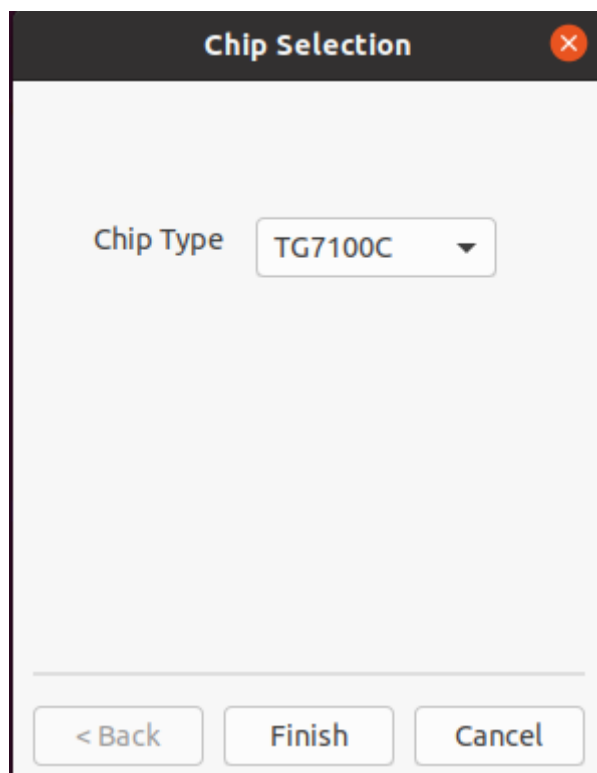


图 3.2: 芯片选择界面

## 3.2 配置程序下载方式

配置参数包括:

- **Interface:** 用于选择烧录的通信接口, 这里选择 **Uart** 进行下载
- **COM Port:** 当选择 **UART** 进行下载的时候这里选择与芯片连接的 **COM** 口号, 可以点击 **Refresh** 按钮进行 **COM** 号的刷新
- **Uart Rate:** 当选择 **UART** 进行下载的时候, 填写波特率, 推荐下载频率 **2MHz**, 不宜过高
- **Board:** 选择所使用的板子型号, 这里选择 **IoTKitA**, 当板子选定后, **Xtal** 会自动更新成与板子匹配的默认值, 当然用户也是可以再次更改的



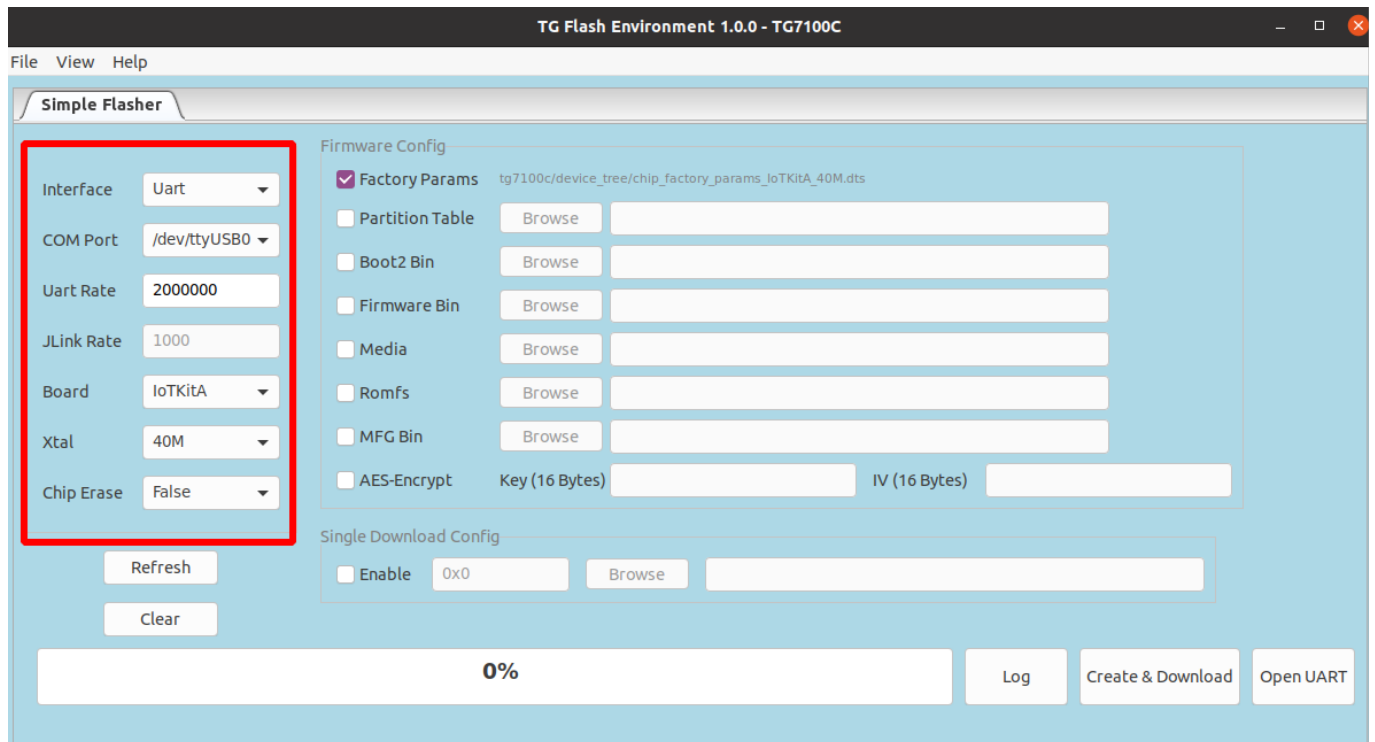


图 3.3: 程序下载方式选择界面

### 3.3 配置下载参数

配置参数包括：

- **Partition Table:** 使用工具目录下对应芯片型号 **partition** 文件夹中的分区表，默认选择 2M 的文件
- **Boot2 Bin:** 它是系统启动后运行的第一个 **Flash** 程序，负责建立安全环境，并引导主程序运行，使用工具目录下对应芯片型号 **builtin\_imgs** 文件夹中的 **Boot2** 文件
- **Firmware Bin:** 用户编译生成的 **bin** 文件路径
- **Media/Romfs:** **Media** 和 **Romfs** 二选一，如果勾选 **Media**，选择的是文件，如果勾选 **Romfs**，则选择的是文件夹
- **MFG Bin:** 选择 **MFG** 文件
- **AES-Encrypt:** 如果使用加密功能，需要将 **AES-Encrypt** 选项选中，并在旁边的文本框中输入加密所使用的 **Key** 和 **IV**。输入的是十六进制对应的“0”~“F”，一个 **Byte** 由两个字符构成，所以 **Key** 和 **IV** 分别要求输入 32 个字符。需要注意的是 **IV** 的最后 8 个字符（即 4Bytes）必须全为 0

目录选择：

- **Partition Table:** work/tg7100c/partition/partition\_cfg\_2M.toml
- **Boot2 Bin:** work/tg7100c/builtin\_imgs/chips\_boot2\_release.bin
- **Firmware Bin:** work/out/smart\_outlet@tg7100c/smart\_outlet@tg7100c.bin

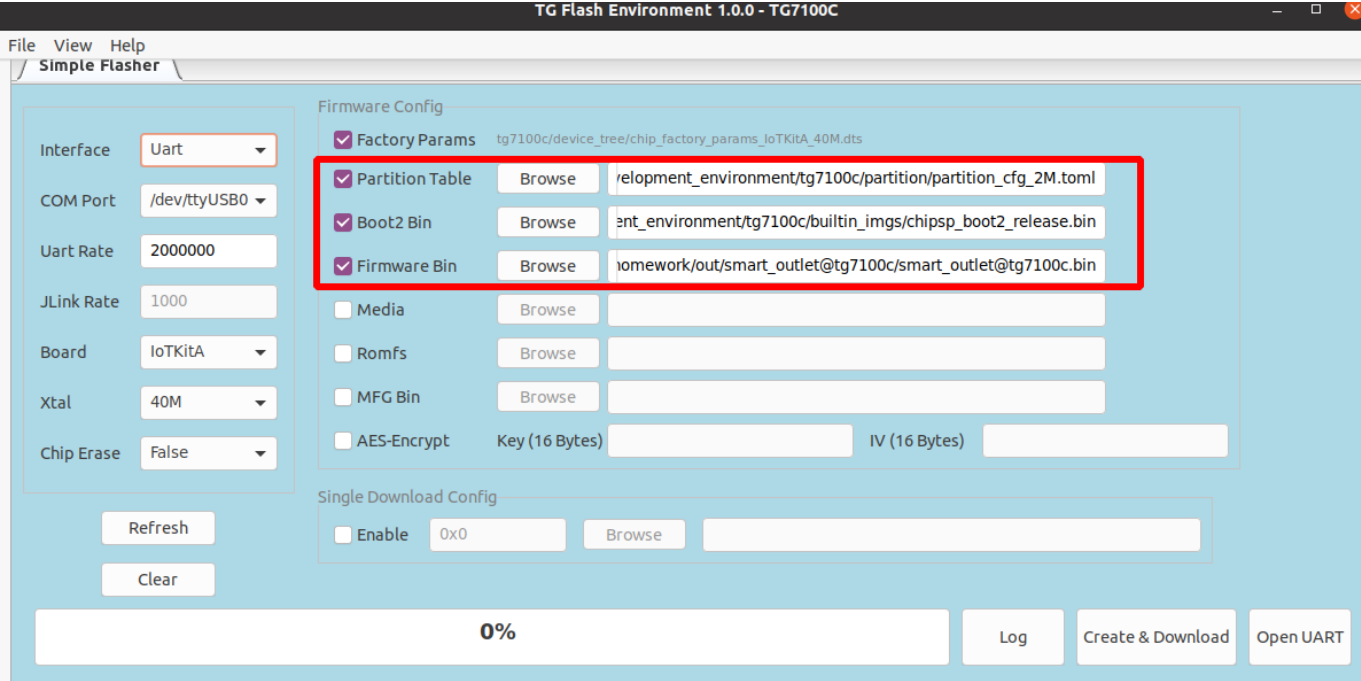


图 3.4: 下载程序选择界面

3.4 下载程序

- 将板子的 BOOT 引脚保持高电平，并且使得芯片复位，使其处于 UART 引导下载的状态。点击 **Create&Download**，开始下载。当弹出 **download success** 时，下载成功。下图: 下载硬件连接。

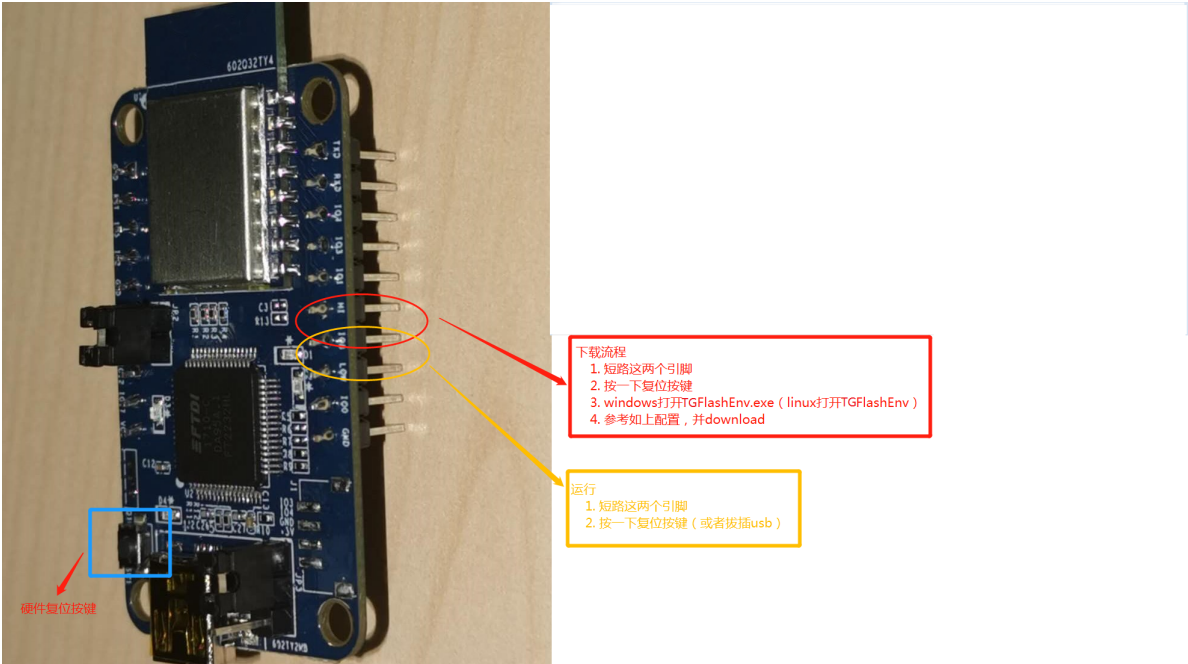


图 3.5: 硬件连接

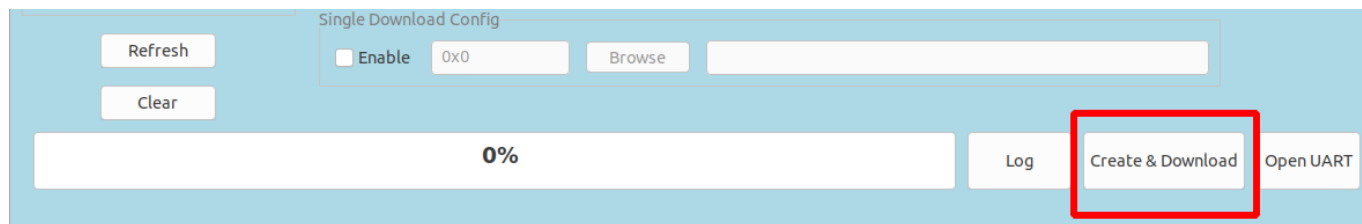


图 3.6: 执行下载界面

- 下载成功后，将板子的 **BOOT** 引脚保持低电平，并且使得芯片复位，使其从 **Flash** 启动。