

ARM Cortex-A7

Embedded Development Platform



**User Manual**

Rev. 1.0

2020/10/13

0755-86200561

Shenzhen 100ask Technology Co. Ltd.



support@100ask.net



Shenzhen, Guangdong, China

# 注意事项与售后维修

1. 注意事项

 使用产品之前，请仔细阅读本手册，并妥善保管，以备将来参考；

 请注意和遵循标注在产品上的所有警示和指引信息；

 请使用配套电源适配器，以保证电压、电流的稳定；

 请在凉爽、干燥、清洁的地方使用本产品；

 请勿在冷热交替环境中使用本产品，避免结露损坏元器件；

 请勿在湿气过重、温度过高或过低环境中使用本产品，使用时注意产品的通风；

 请勿将任何液体泼溅在本产品上，禁止使用有机溶剂或腐蚀性液体清洗本产品；

 请勿在多尘、脏乱的环境中使用本产品，如果长期不使用，请包装好本产品；

 请勿在振动过大的环境中使用，任何跌落、敲打或剧烈晃动都可能损坏线路及元器件；

 请勿在通电情况下，插拔核心板及外围模块(特别是串口模块)；

 请勿自行维修、拆解本产品，如产品出现故障应及时联系本公司进行维修；

 请勿自行修改或使用未经授权的配件，由此造成的损坏将不予保修；

2. 售后维修

1) 保修期限

 底板、核心板：三个月（非人为损坏）

 显示屏：七天（非人为损坏）

2) 保修说明

 7天内：产品（底板、核心板、屏幕）非人为损坏，本公司免费更换/维修，并承担来回运费；

 7天至3个月内：底板、核心板非人为损坏，本公司免费维修，并承担来回运费（屏幕不提供维修）；

 3个月至1年：底板、核心板非人为损坏或人为轻微损坏，只收更换元器件费用，免费维修，买家承担来回运费；

 起始时间以快递签收日为准；

3) 联系方式

官方网站：[www.100ask.net](http://www.100ask.net)

淘宝网站：[100ask.taobao.com](https://100ask.taobao.com/)

地 址：广东省深圳市龙岗区布吉南湾街道平吉大道建昇大厦B1505

联系人：售后维修部

电 话：0755-86200561

邮 编：518114

邮寄须知：保修期限内，寄回本产品请预先垫付邮费，公司不接收任何到付快递。

# 技术支持与开发定制

1. 技术支持范围

1) 本公司提供的各类开发软件的安装，入门使用，环境搭建；

2) 本公司提供的所有裸机代码的烧写验证；

3) 本公司发布的操作系统的编译、烧写；

4) 本公司发布产品的工控板、模块的硬件原理；

5) 本公司发布的各种外设模块驱动及源码；

6) 本公司发布的配套手册在使用过程中遇到的问题；

7) 本公司产品的故障诊断及售后维修服务；

2. 技术讨论范围

由于嵌入式系统知识范围广泛，涉及知识纷繁复杂，我们无法保证对各种问题都能一一解答，以下内容无法供技术支持，只能提供建议。

1) 本公司发布的教程之外的知识；

2) 非本公司发布的U-Boot、Linux内核的编译和移植；

3) 非本公司发布的工控板的各类驱动支持；

4) 非本公司发布的外设模块的硬件原理和驱动设计；

3. 技术支持方式

1) 官方论坛发帖提问(推荐)：[bbs.100ask.net](http://bbs.100ask.org/)

2) 官方淘宝通过阿里旺旺咨询：[100ask.taobao.com](https://100ask.taobao.com/)

3) QQ群咨询（QQ群号咨询淘宝客服，需提供淘宝购买订单号验证加入）；

4) 技术支持邮箱：[support@100ask.net](mailto:support@100ask.org)

5) 联系电话：0755-86200561

4. 技术支持时间

星期一到星期五;上午 9:00—12:00;下午 14:00—17:30;

公司按照国家法定节假日安排休息，在此期间无法提供技术支持，请将问题发送至技术支持邮箱或在论坛对应板块发帖，我们将在工作日尽快给您回复。

5. 投诉和建议

如果您对我们有不满意或者建议，可发送邮件到[support@100ask.net](mailto:support@100ask.org)进行反馈，也可拨打0755-86200561取得联系，我们将不断改进。

6. 定制开发服务

本公司提供嵌入式操作系统底层驱动、硬件板卡的有偿定制开发服务，以缩短您的产品开发周期。请将需求发送邮件到[support@100ask.net](mailto:support@100ask.org)。

# 资料获取与后续更新

1. 资料的获取

1) 百度网盘下载

百度网盘里面有本产品的所有配套资料，包括原理图、发布的U-Boot、内核镜像和源码、所需的开发软件、工具等等。

进入<http://download.100ask.net/>，找到对应的开发板即可。

2) 视频配套教程

后续会为该工控板录制一套裸机、Linux驱动、应用的配套付费教学视频，有需要的客户可以进

入官方淘宝[100ask.taobao.com](https://100ask.taobao.com/)选购。

3) 维基百科教程

维基百科里面会有视频配套的笔记，进入[wiki.100ask.net](http://wiki.100ask.org)，选择对应的板块查看。

2. 后续更新

后续文档、视频等资料的更新，为了确保您的资料是最新状态，请密切关注我们的动态，我们将会通过微信公众号和QQ群公告推送，购买了本产品的客户请添加QQ群（QQ群号咨询淘宝客服，需提供淘宝购买订单号验证加入）或关注微信公众号。



# 版权声明

**百问科技©2020**

深圳百问网科技有限公司版权所有，并保留对本手册及声明的一切权力。

未得到本公司的书面许可，任何单位和个人不得以任何方式或形式对本手册内的任何部分进行复制、摘录、备份、修改、传播、翻译成其他语言、将其全部或部分用于商业用途

# 更新记录

|  |  |
| --- | --- |
| **类别** | 嵌入式开发 |
| **文档名** | STM32MP157裸机开发完全手册 |
| **当前版本** | 1.0 |
| **适用型号** | 100ASK\_STM32MP157 |
| **编辑** | 百问科技文档编辑团队 |
| **审核** | 韦东山 |

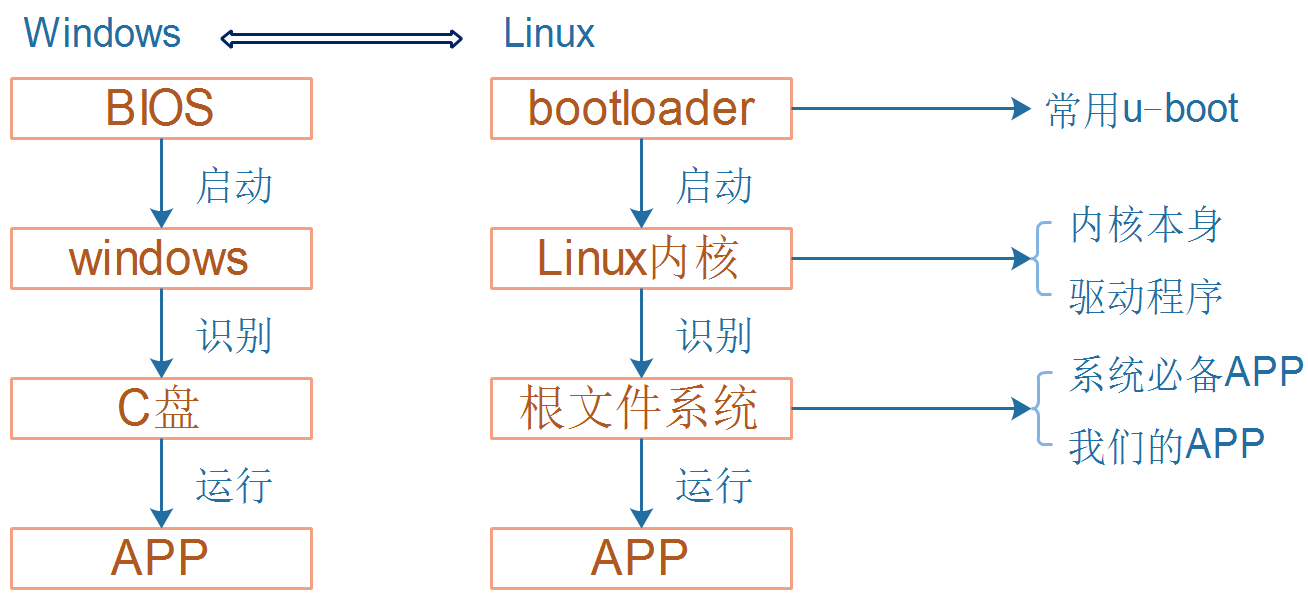
|  |  |  |
| --- | --- | --- |
| **修改日志** | | |
| 版本 | 发布时间 | 更改说明 |
| 1.0 | 2020.10.13 | 初始版本 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 第1 篇资料下载、环境搭建

## 第一章 百问网的视频及学习路线

### 1.1 嵌入式Linux的组成

嵌入式Linux系统，就相当于一套完整的PC软件系统。



很多人喜欢从系统启动流程开始学习：先学习裸机，裸机集合起来就是u-boot，再学习内核移植、驱动开发，接下来学习根文件系统，最后学习APP开发。

学习裸机需要2、3个月，学习u-boot也需要2、3个月，结果工作中u-boot基本不用改，并且u-boot比驱动开发还难！

按这套流程下来，学了后面忘了前面，最惨的是：不能快速上手工作，消耗学习热情！

入门讲究的是快速，入门之后再慢慢深入，

特别是对于急着找工作的学生，对于业余时间挑灯夜读的在职工程师，一定要快！

有些学员时间充裕，打算从单片机慢慢开始学习，把基础打扎实。我们也提供了从单片机开始的教程。

请看下一节。

### 1.2 新教程有哪些内容



在上图中，我们提供了3条学习路线：

① 0基础，不想玩单片机：

如果你对单片机不感兴趣，更关心的是Linux的驱动、APP、项目，那么可以从Linux应用基础、Linux驱动基础、Linux项目入手。

② 想学单片机：

单片机的学习有两种路线：使用HAL库，使用寄存器。

想快速上手，快速做出项目，可以使用HAL库。缺点是HAL库封装了太多的技术细节，你只会调用这些函数，无法深入理解细节，对以后技术的发展不利。

想深入理解单片机开发的本质，可以丢弃HAL库，使用寄存器来开发。在这过程中学习到的知识，对于以后学习各类RTOS、学习Linux的u-boot、学习Linux驱动开发，都很有帮助。

目前，百问网正在开发基于寄存器的课程，就是上图中的“单片机核心/RTOS必备”，你看到本文档时，就已经可以在<http://www.100ask.net>官网观看视频了。

基于HAL库的课程，还在开发中，还没有发布。

### 1.3 什么人需要学习裸机

如果你符合这些条件之一，那么可以从裸机开始学习：

① 没有硬件基础，比如看不懂原理图，想纯粹地掌握硬件编程；

② 没有单片机开发经验，想顺便掌握单片机的开发技能；

使用HAL库开发时，使用KEIL开发时，Windows工具封装了太多细节，在Linux下学习可以学到更多，并且完全覆盖STM32单片机的知识。

③ 有单片机基础，但是想深入内部原理的人(MDK等图形工具屏蔽了太多细节)；

④ 想深入研究U-Boot，但是U-Boot又太复杂，所以先把各个部件按裸机来研究一下；

⑤ 想深入了解主芯片的人：

在日常开发过程中，我们使用Linux驱动来操作硬件，很多时候不涉及寄存器。如果你需要微调驱动、解决BUG，那就需要看芯片手册。在我们的裸机文档里，有深入的讲解。

## 第二章 资料下载

### 2.1 有哪些资料、怎么下载

对于裸机开发，我们把所有资料都放在GIT上了，包括2部分资料：

cortexA7的编译工具、视频配套的源码/文档/烧写工具。

分别位于2个GIT仓库：

<https://e.coding.net/weidongshan/noos/cortexA7_windows_tools.git>

<https://e.coding.net/weidongshan/noos/doc_and_source_for_mcu_mpu.git>

### 2.2 GIT使用简明教程

#### 2.2.1 安装GIT

在Windows下，GIT名为msysGit，从<https://gitforwindows.org/> 上下载安装文件，双击安装即可，安装选项很多，使用默认选项即可。

如果下载慢，可以在百度上搜：Git-2.28.0-64-bit.exe，自行下载。

对于Windows或Linux，它们的命令行用法相似，对于Windows，进入Git命令行的方法是在“开始”->“所有程序”->“Git”下启动Git Bash。

Git Bash的命令用法跟Linux完全一样，比如cd、ls等命令。

#### 2.2.2 GIT常用命令

|  |  |  |
| --- | --- | --- |
| GIT命令 | 说明 | 示例 |
| clone | 克隆，从远程下载仓库 | git clone https://e.coding.net/weidongshan/01\_all\_series\_quickstart.git |
| pull | 拉取，从远程更新仓库 | git pull origin |
| log | 查看本地仓库的记录 | git log，快捷键：f前翻、b后翻、q退出 |
| status | 查看本地仓库状态，  比如有无修改，  修改有无提交进仓库里 | git status |
| tag | 查看标签，或是打标签 | git tag // 查看标签  git tag v2 // 打标签 |
| checkout | 提取出某个版本 | 使用git log查看版本，可以看到这样的版本号：  commit 4eb78f0a27a85957e1d38a23c5b031cc2aa4b93f  这时就可以执行以下命令取出这个版本：  git checkout 4eb78f0a27a85957e1d38a23c5b031cc2aa4b93f  执行上述命令后，当前目录里就是这个版本的源码；  要想提取出最新的代码，执行：  git checkout master |

如果只是使用GIT来下载代码，看后面的示例就可以了。如果要深入学习GIT，用GIT来管理你的代码、协同开发，这有一个图形化介绍GIT的网站：<https://learngitbranching.js.org/?demo=&locale=zh_CN>

#### 2.2.3 使用示例：cortexA7的编译工具

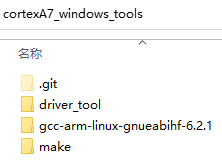
使用git下载资料，需要先知道git仓库的地址。：

<https://e.coding.net/weidongshan/noos/cortexA7_windows_tools.git>

要获取编译器等工具，可以执行以下命令，这称为“克隆”，这会得到一个名为cortexA7\_windows\_tools的目录：

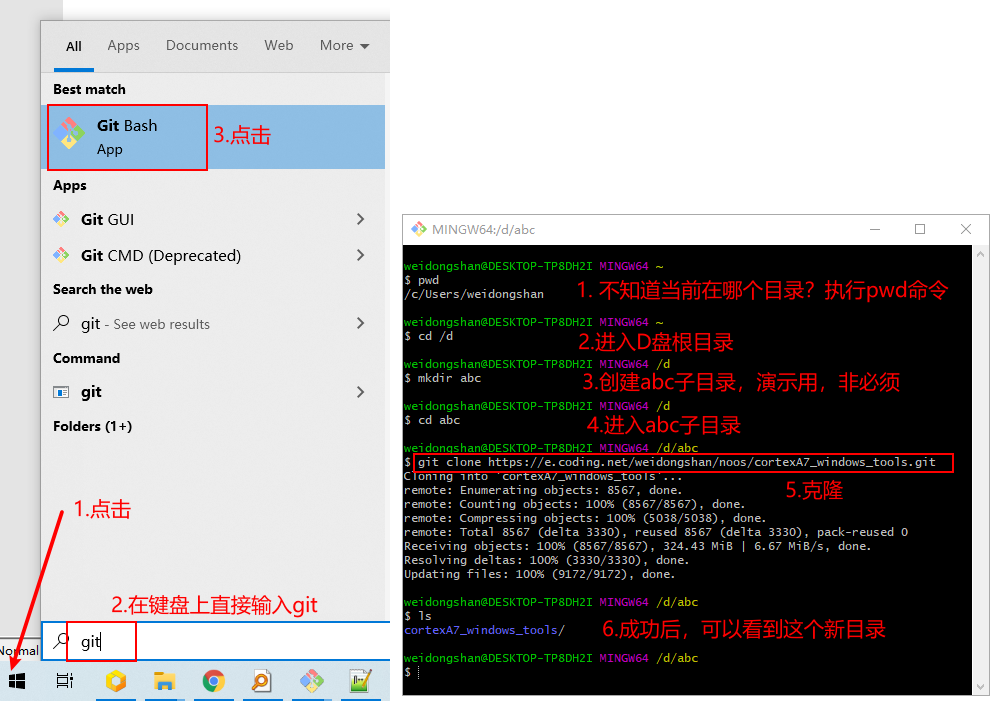
$ git clone <https://e.coding.net/weidongshan/noos/cortexA7_windows_tools.git>

这个目录结构如下：



这个目录的内容比较稳定，基本上不会更新，下载一次就可以了。

示例如下，你不知道当前目录是什么是，可以执行“pwd”命令查看：



#### 2.2.4 使用示例：视频配套的源码/文档/烧写工具

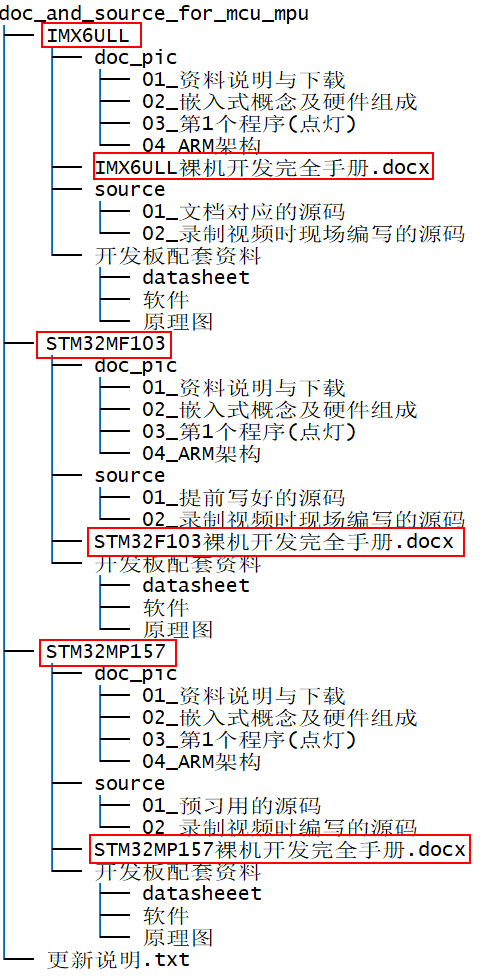
使用git下载资料，需要先知道git仓库的地址。：

<https://e.coding.net/weidongshan/noos/doc_and_source_for_mcu_mpu.git>

要获取编译器等工具，可以执行以下命令，这称为“克隆”，这会得到一个名为doc\_and\_source\_for\_mcu\_mpu的目录：

$ git clone <https://e.coding.net/weidongshan/noos/doc_and_source_for_mcu_mpu.git>

这个目录结构如下，目前支持3款开发板(下图的目录会随着视频的录制而更新)：



如果在你“克隆”之后，我们又更新了源码，你可以先进入该目录，然后更新。

启动git bash后，使用cd命令可以切换目录。假设要进入D:\abc\doc\_and\_source\_for\_mcu\_mpu目录，可以执行以下命令：

$ cd /D

$ cd abc

$ cd doc\_and\_source\_for\_mcu\_mpu

也可以执行一个命令直接进入该目录，注意目录分隔符是“/”而非“\”。

$ cd /D/abc/doc\_and\_source\_for\_mcu\_mpu

在doc\_and\_source\_for\_mcu\_mpu目录下，执行以下命令获得最新版本。

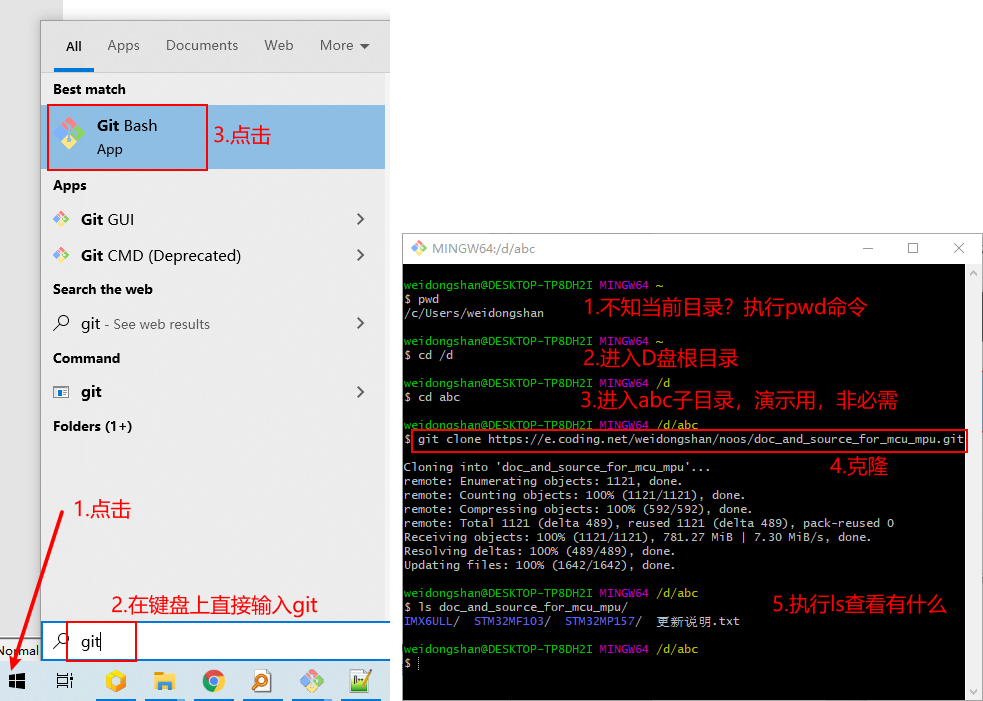
$ git pull origin

下图是在Windows上使用git下载、查看、更新源码的操作步骤。

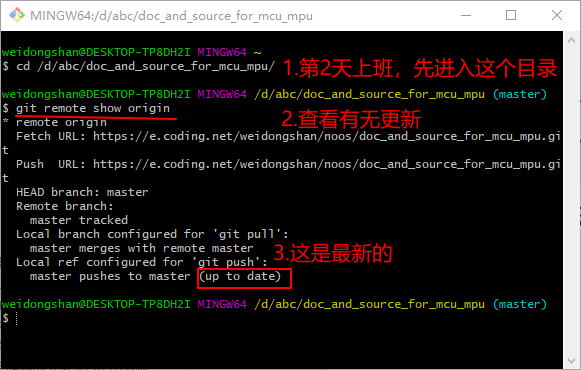
**注意**：建议下载源码后，复制到其他目录去修改；否则以后更新时可能会和你的本地修改产生冲突。

1. 第1天，下载源码

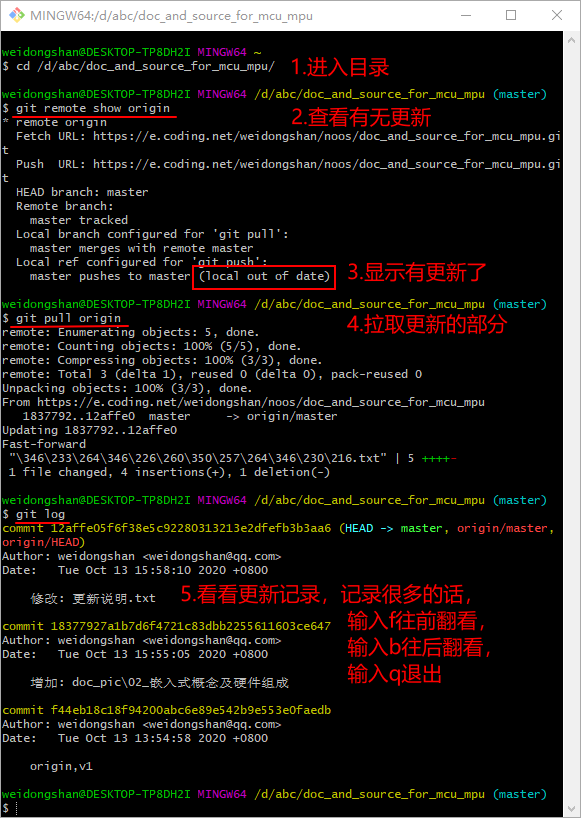
假设你要把源码下载到D盘abc目录，如下图操作：



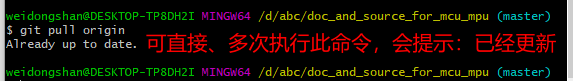
2. 第2天，查看无更新



3. 第3天，查看有更新



**注意**：不执行“git remote show origin”查看状态，而是直接执行“git pull origin”也是可以的，后面这个命令会自动检查，有更新它就会下载更新部分，没有更新也会提示你，如下图：

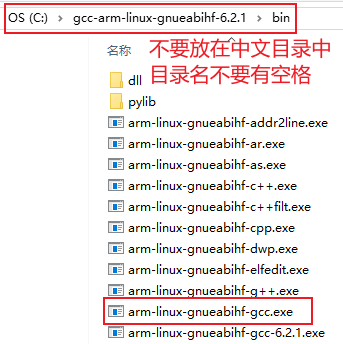


## 第三章 环境搭建

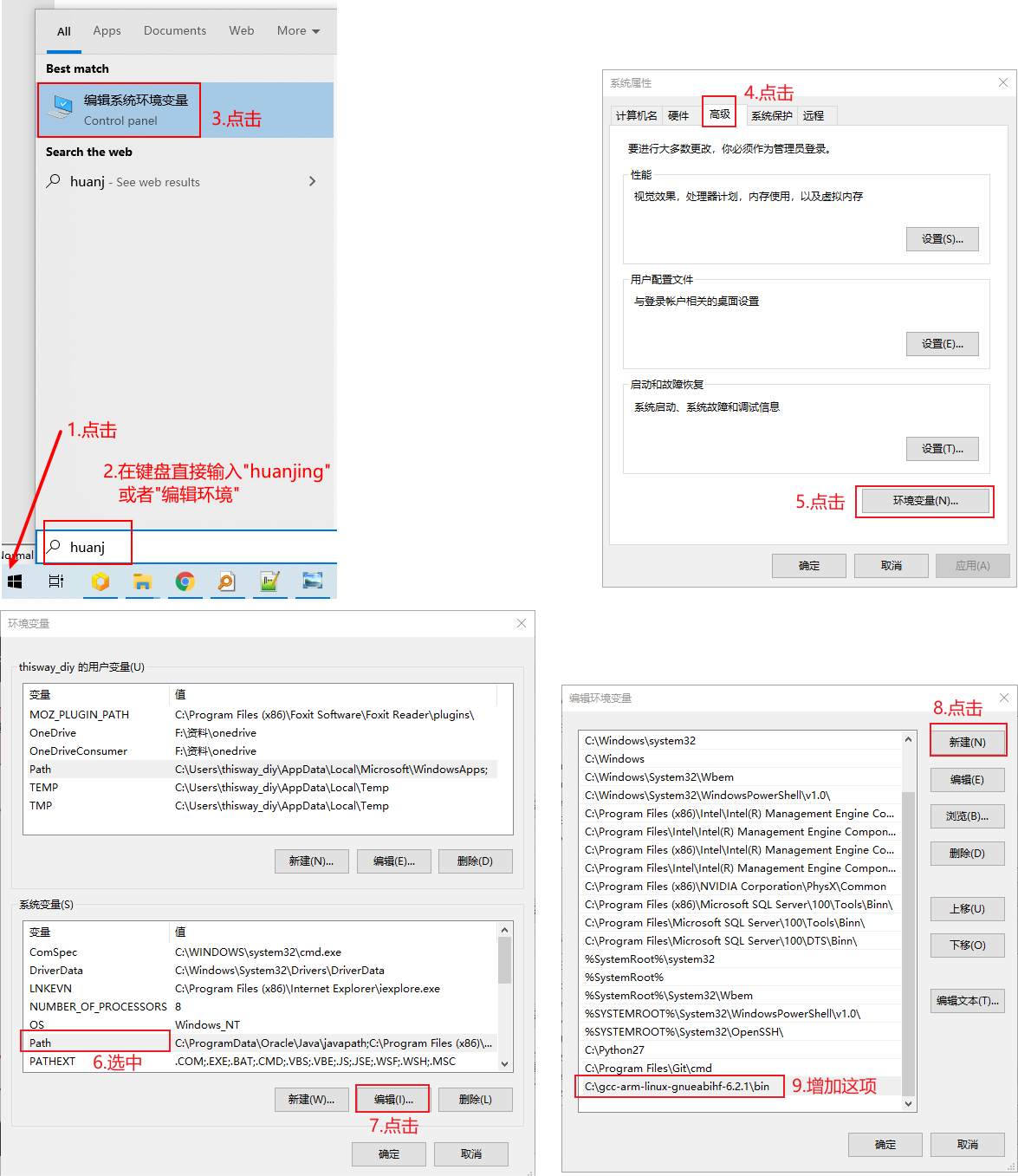
### 3.1 安装编译工具

#### 3.1.1 安装GCC编译器

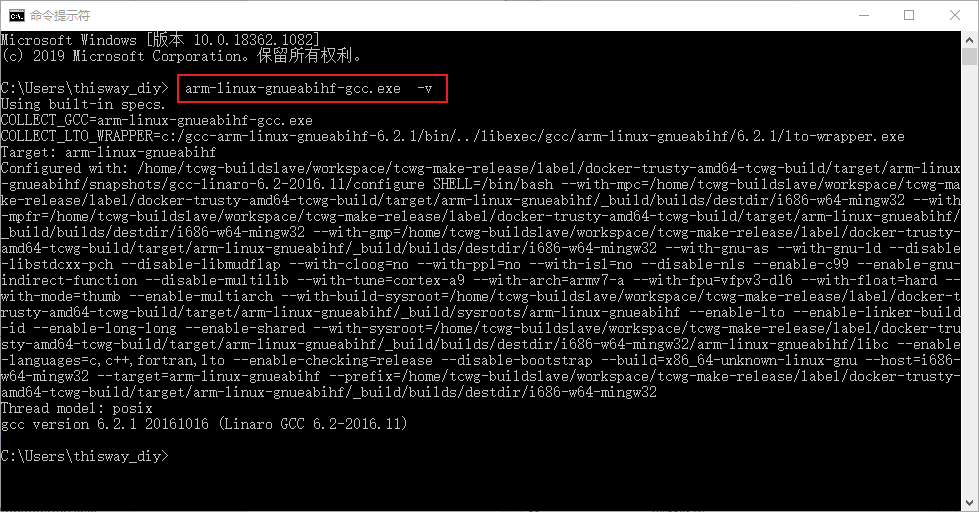
从GIT下载到“cortexA7\_windows\_tools”后，把里面的“gcc-arm-linux-gnueabihf-6.2.1”复制到C盘根目录：



设置环境变量Path，增加“C:\gcc-arm-linux-gnueabihf-6.2.1\bin”：

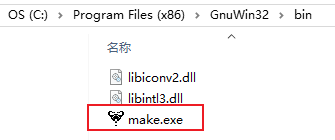


验证工具链：启动命令行，执行“arm-linux-gnueabihf-gcc.exe -v”，如下图所示：

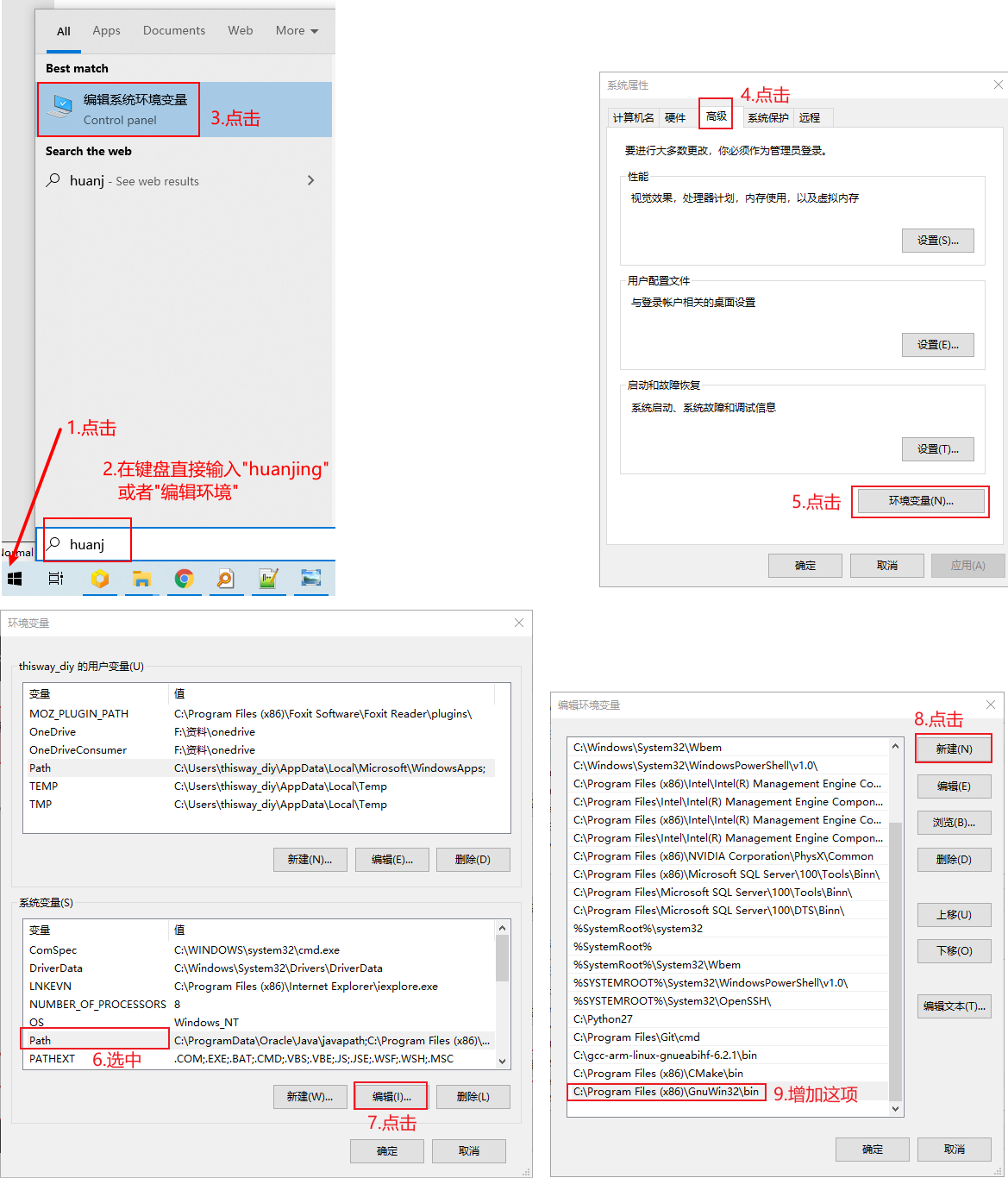


#### 3.1.2 安装make

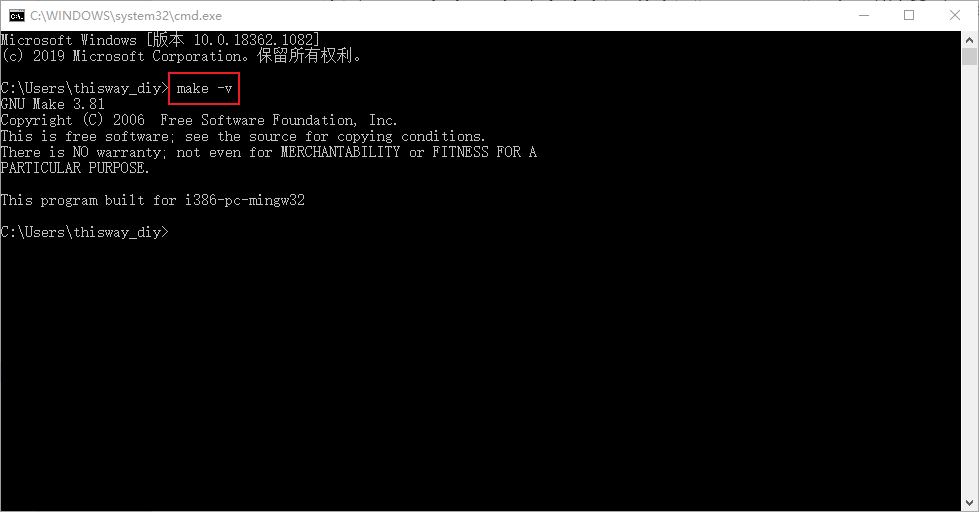
从GIT下载到“cortexA7\_windows\_tools”后，双击安装里面的“make/make-3.81.exe”。安装成功后，可以在安装目录下看到make.exe，类似下图：



设置环境变量Path，增加“C:\Program Files (x86)\GnuWin32\bin”：



验证make命令：启动命令行，执行“make -v”，如下图所示：

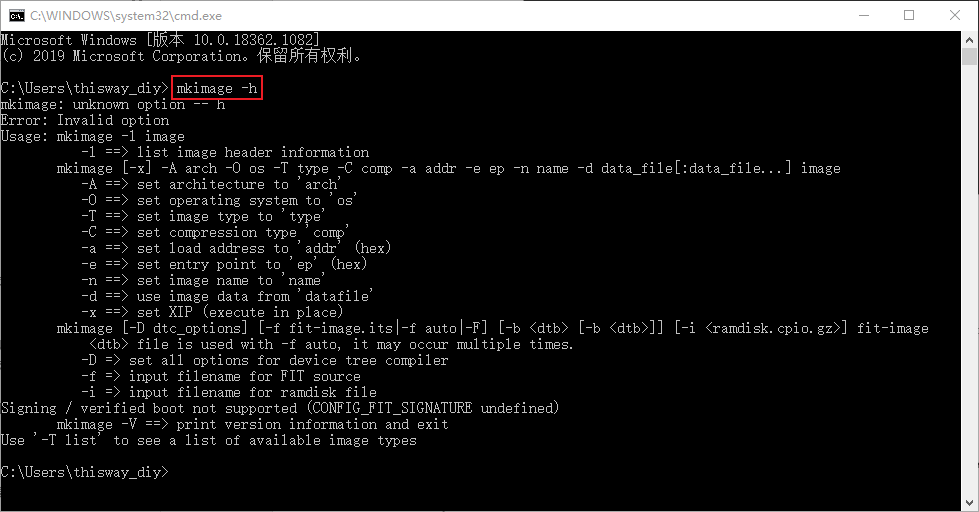


### 3.2 安装mkimage.exe

mkimage工具的源码来自u-boot，它用来给bin文件添加一个头部信息，很多芯片的固件都需要根据这个头部信息把bin文件放到内存去执行。

我们已经事先把mkimage.exe放到GCC工具链的bin目录里，无需单独安装。

验证mkimage.exe：启动命令行，执行“mkimage -h”，如下图所示：



### 3.3 验证Git Bash

在前面的章节已经安装过msysGit，它的命令行被称为“Git Bash”，里面集成了一些Linux命令，比如：cd、ls、pwd、rm等等。

以后我们都会在Git Bash中执行make命令来编译程序，而不是在Windows命令行中执行命令。

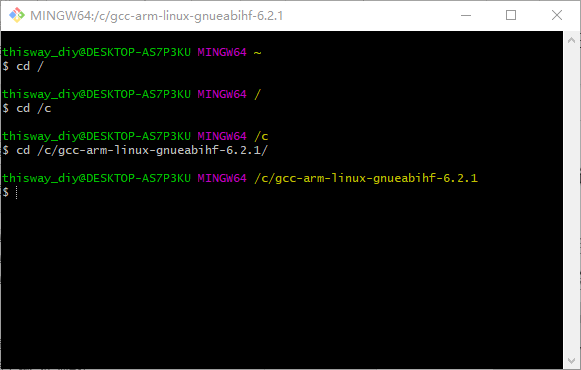
先验证一下Git Bash，启动Git Bash后，可以执行下列命令验证GCC、make、mkimage：

arm-linux-gnueabihf-gcc -v

make -v

mkimage -h

在Git Bash中，最常用的命令就是“cd”，它的含义是“change directory”。要注意的是，Git Bash中使用的Linux风格的目录名，比如Windows下“C:\abc”目录，在Git Bash中用“/C/abc”来表示。操作示例如下：



在Git Bash中，可以使用TAB键来补全命令、补全目录，比如上图中，键入“cd /c/gcc”后按TAB键，就会补全为“cd /c/gcc-arm-linux-gnueabihf-6.2.1/”。

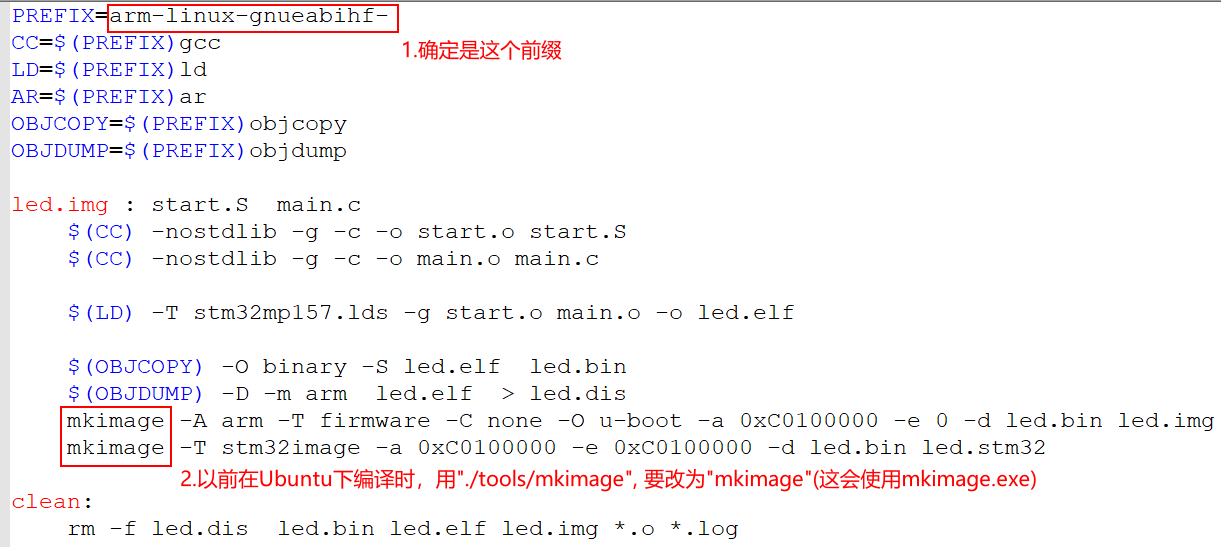
如果想在Git Bash中使用更多的命令，请参考[http://www.100ask.net](http://www.100ask.net/)中的第2篇视频：



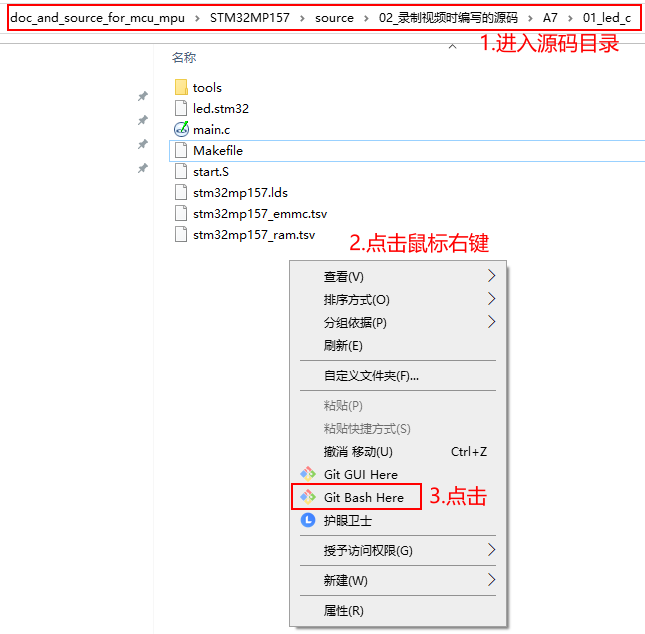
### 3.4 编译第1个裸机程序

使用GIT下载资料后，STM32MP157的第1个A7裸机程序在这个目录下： doc\_and\_source\_for\_mcu\_mpu\STM32MP157\source\02\_录制视频时编写的源码\A7\01\_led\_c。

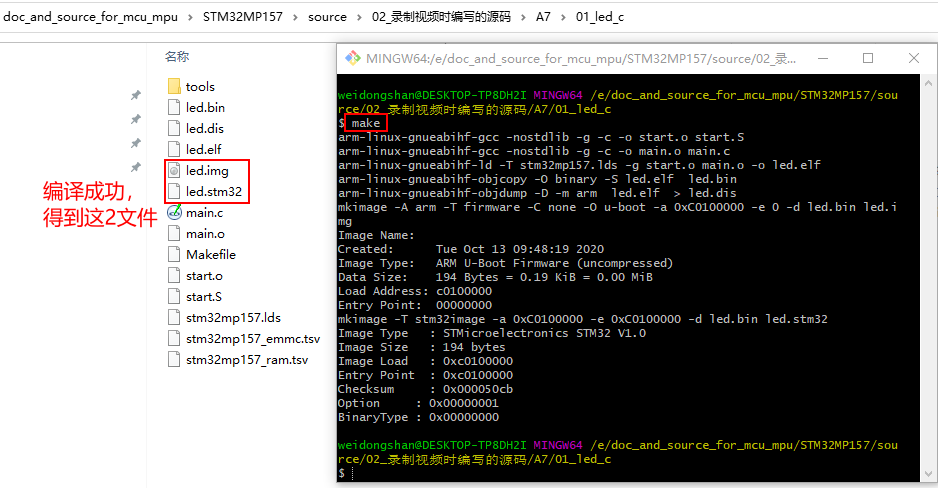
在编译之前，先确定它的Makefile：



启动Git Bash：



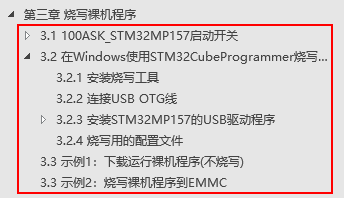
执行make命令：



### 3.5 运行1个裸机程序

请观看视频。

或者参考本文档《第2篇 100ASK\_STM32MP157开发板使用手册》的《第三章 烧写裸机程序》，全部阅读：

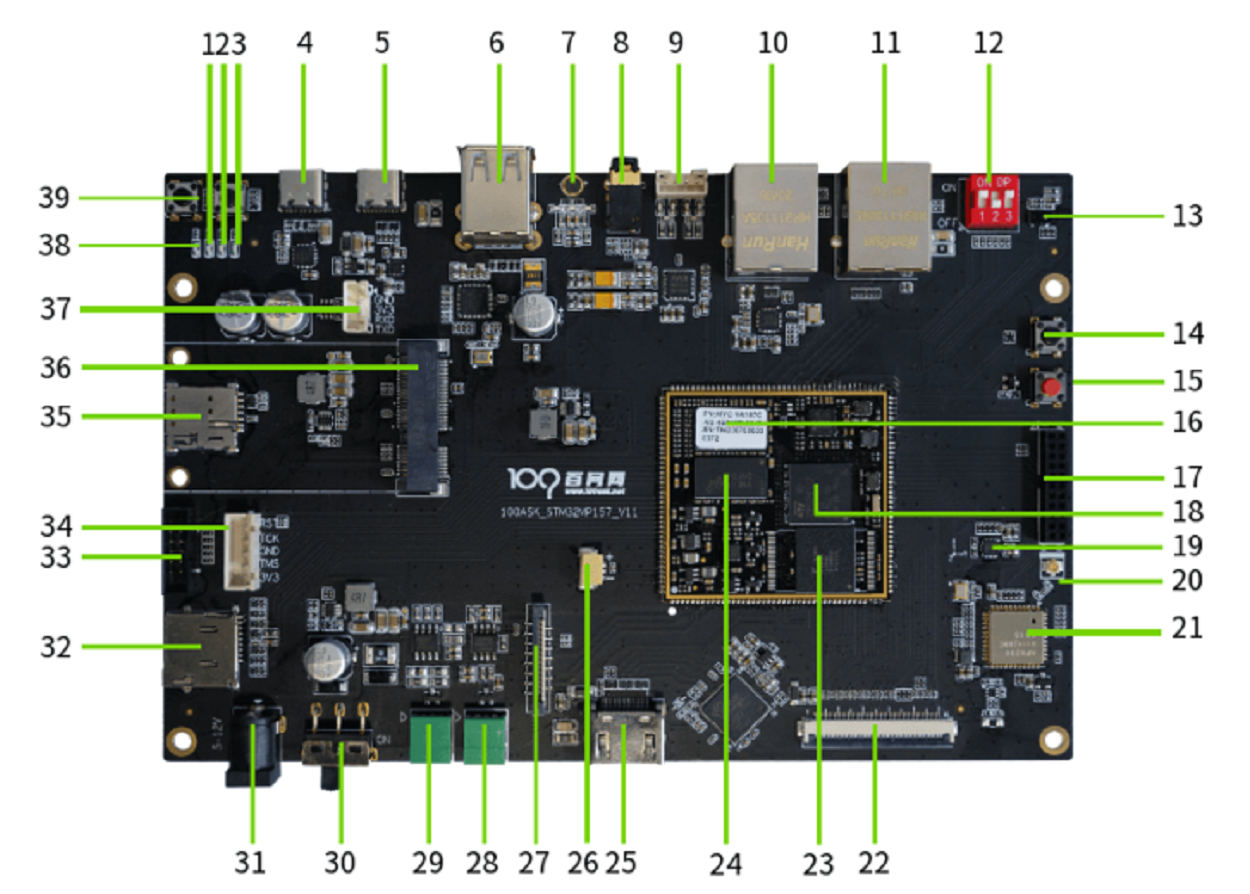


# 第2篇 100ASK\_STM32MP157开发板使用手册

## 第一章100ask\_stm32mp157开发板简介

### 1.1 100ask\_stm32mp157全功能开发板硬件资源简介

开发板图片如下，各个标号对应的硬件在板子背后都写有名字：



上面开发板图片中，各标号的硬件含义如下图：



## 第二章 启动开发板

### 2.1 启动方式选择

板子上的红色拨码开关用来设置启动方式，支持这四种方式：EMMC启动、SD卡启动、USB烧写以及M4（debug）模式。

板子背后画有一个表格，表示这四种方式如何设置。表格中BOOT0对应开关1，BOOT1对应开关2，BOOT2对应开关3，如下：

|  |  |  |  |
| --- | --- | --- | --- |
| BOOT CFG | | | |
| BOOT | BOOT0 | BOOT1 | BOOT2 |
| EMMC | OFF | ON | OFF |
| SD | ON | OFF | ON |
| USB | OFF | OFF | OFF |
| M4(debug) | OFF | OFF | ON |

这四种启动方式的设置示意图如下：

|  |  |  |  |
| --- | --- | --- | --- |
| EMMC启动 | SD/TF卡启动 | USB下载模式启动 | M4 debug模式启动 |
|  |  |  |  |

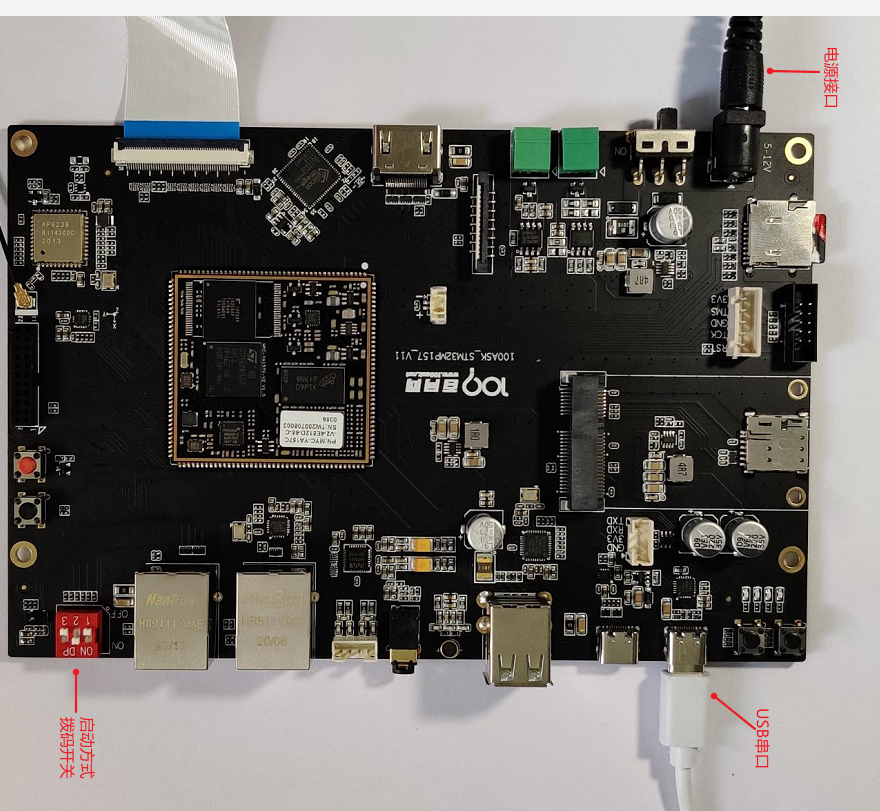
刚出厂的板子在EMMC上烧写了系统，你可以设置为EMMC启动方式。

### 2.2 串口连接

在后面的操作里，都是通过串口与板子进行“交流”。串口是串行接口的简称，是指数据一位一位地顺序传送，其特点是通信线路简单。

#### 2.2.1 连接串口线和电源线、配置串口工具

如下图所示将串口线与电脑、板子连接，开发板插上电源。



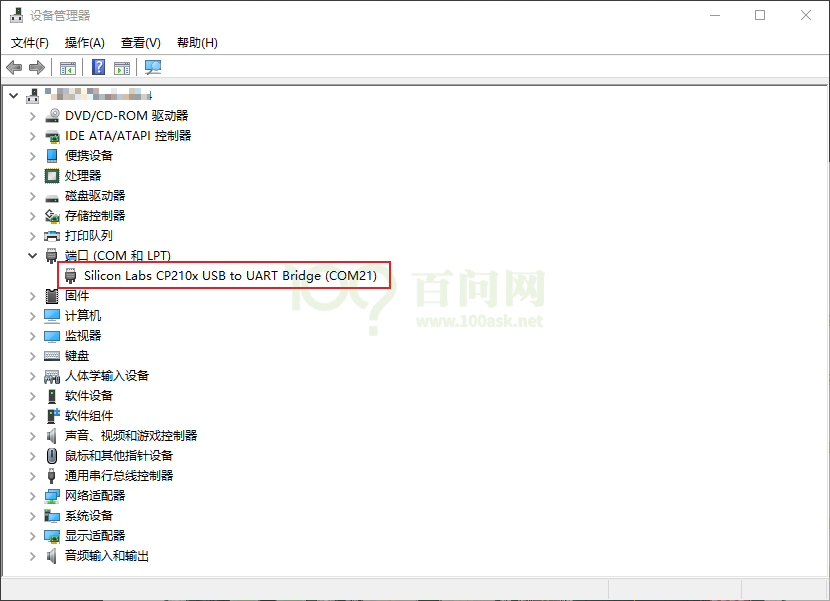
其中特别需要注意的几点：

① 板子的启动选择拨到正确的启动方式，保证该启动方式里面有系统可以运行；

② 板子如图所示插上配套的电源到电源接口，电源开关暂时不用打开。

#### 2.2.2 安装USB串口模块驱动

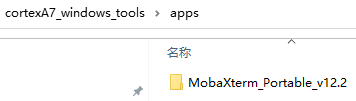
接好Type C usb数据线后，Windows会自动安装驱动(安装可能比较慢，等一分钟左右)。打开电脑的“设备管理器”，在“端口 (COM和LPT)”项下，可以看到如下图中的“(COM21)”。这里的“**COM21**”可能与你电脑上的不一样，记住你电脑显示的数字。



如果电脑没有显示出端口号，就需要手动安装驱动，从驱动精灵官网（[www.drivergenius.com](http://www.drivergenius.com)）下载一个驱动精灵，安装、运行、检测，会自动安装上串口驱动。

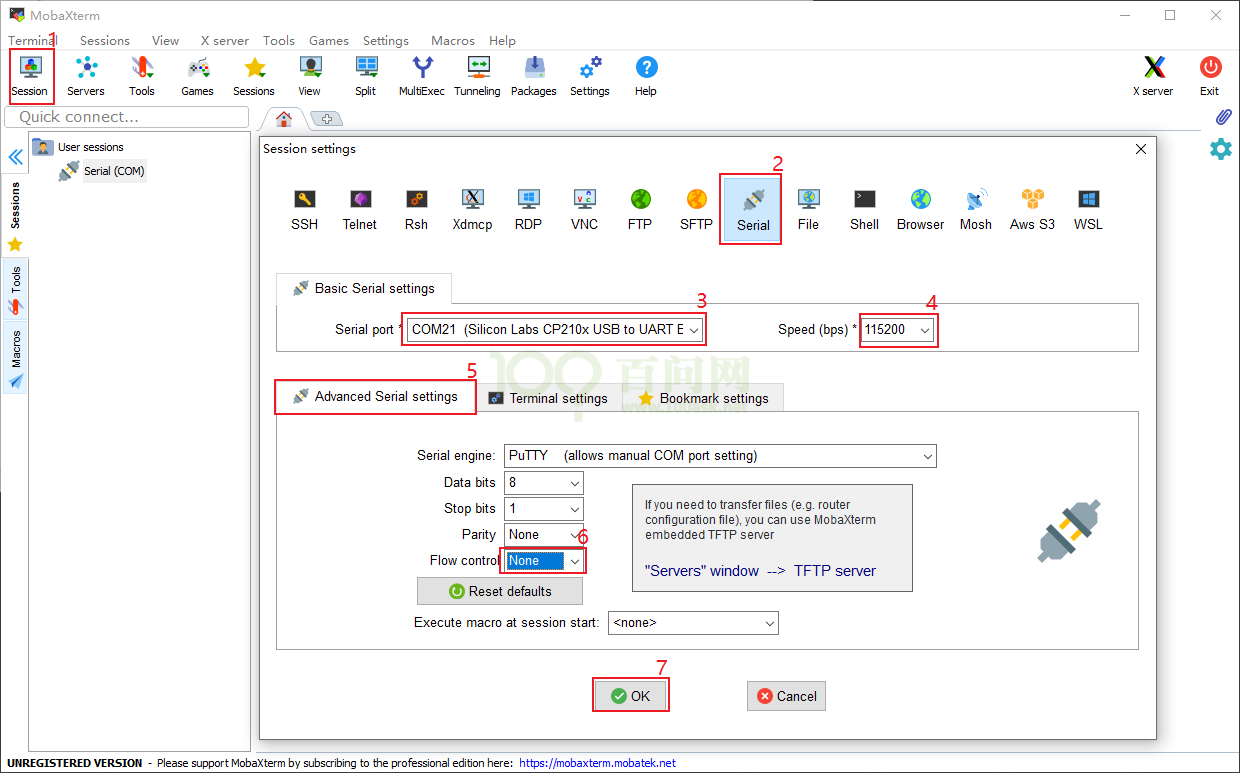
#### 2.2.3 使用MobaXterm软件打开串口

下载GIT资料后，MobaXterm软件位于如下目录：

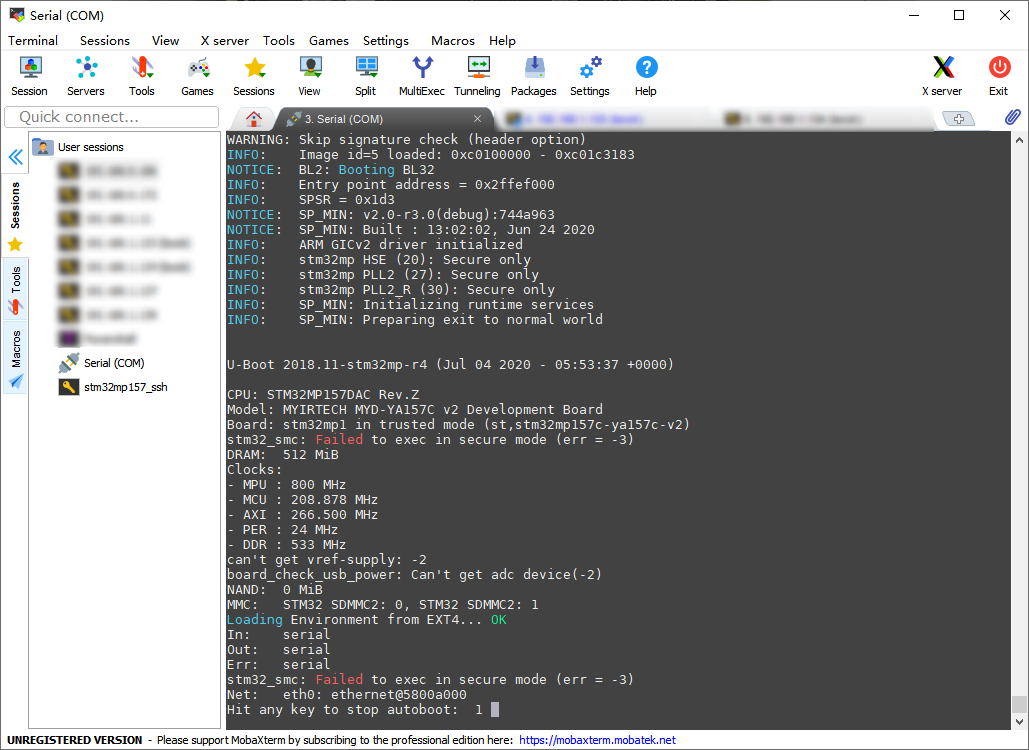


打开MobaXterm，点击左上角的“Session”，在弹出的界面选中“Serial”，如下图所示选择端口号（前面设备管理器显示的端口号COM21）、波特率（Speed 115200）、流控（Flow Control: none）,最后点击“OK”即可。步骤如下图所示。

**注意**：流控（Flow Control）一定要选择none，否则你将无法在MobaXterm中向串口输入数据。



随后显示一个窗口， 此时打开板子的电源开关，将收到板子串口发过来的数据，如下图所示。



#### 2.2.4 开发板登录名是root，无需密码

在串口看到“100ask login:”时，输入“root”并回车即可，如下图所示：



## 第三章 烧写裸机程序

### 3.1 100ASK\_STM32MP157启动开关

#### 3.1.1 全功能版

板子上的红色拨码开关用来设置启动方式，支持这四种方式：EMMC启动、SD卡启动、USB烧写以及M4（debug）模式。

板子背后画有一个表格，表示这四种方式如何设置。表格中BOOT0对应开关1，BOOT1对应开关2，BOOT2对应开关3，如下：

|  |  |  |  |
| --- | --- | --- | --- |
| BOOT CFG | | | |
| BOOT | BOOT0 | BOOT1 | BOOT2 |
| EMMC | OFF | ON | OFF |
| SD | ON | OFF | ON |
| USB | OFF | OFF | OFF |
| M4(debug) | OFF | OFF | ON |

这四种启动方式的设置示意图如下：

|  |  |  |  |
| --- | --- | --- | --- |
| EMMC启动 | SD/TF卡启动 | USB下载模式启动 | M4 debug模式启动 |
|  |  |  |  |

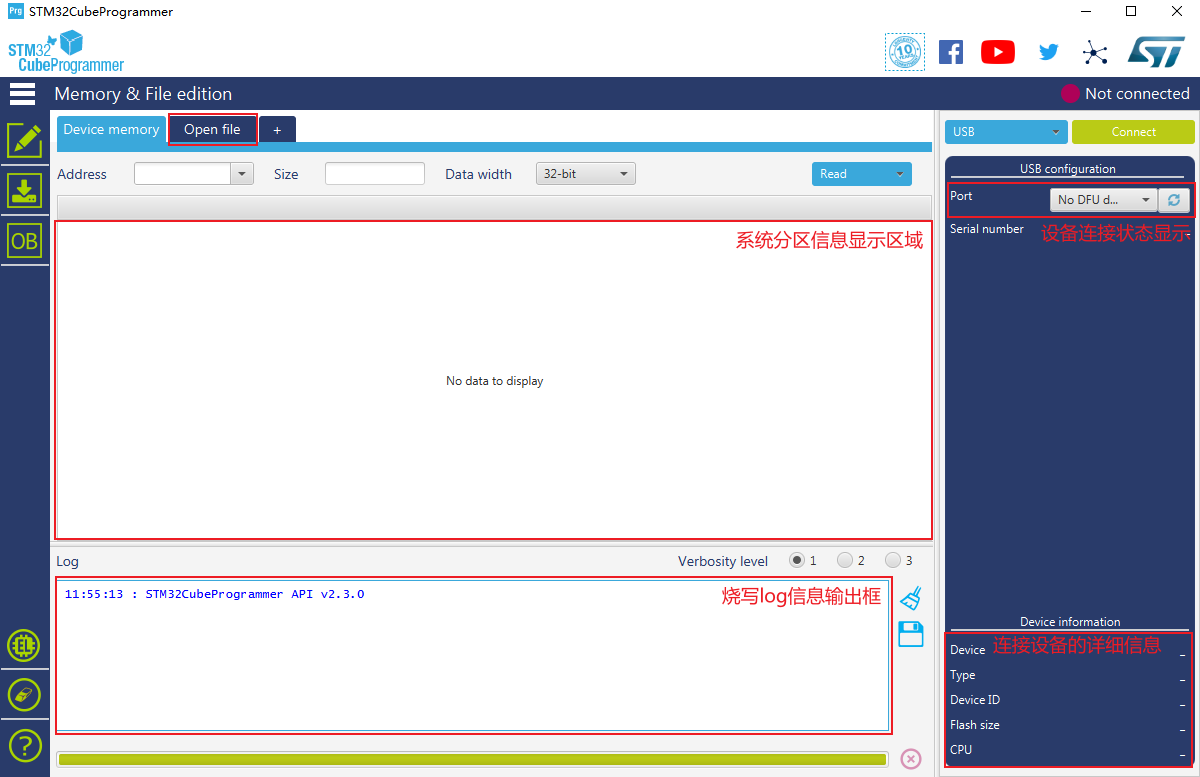
刚出厂的板子在EMMC上烧写了系统，你可以设置为EMMC启动方式。

其中的USB启动模式主要用来烧写系统。

### 3.2 在Windows使用STM32CubeProgrammer烧写工具

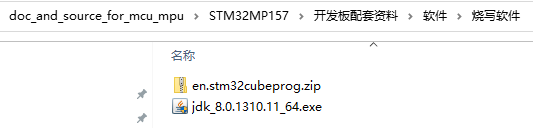
ST公司给stm32mp157系列开发板提供了烧写工具：STM32CubeProgrammer。它的使用界面如下所示：

详细参考页面：<http://wiki.100ask.org/STM32CubeProgrammer>



#### 3.2.1 安装烧写工具

下载GIT资料后，可以看到烧写工具：



解压后会发现此烧写工具支持三种系统运行环境，这里我们默认使用windows版本，其它版本暂不做讲解。

解压后双击执行**SetupSTM32CubeProgrammer-2.1.0.exe** 可执行程序，首次安装如果没有安装java运行环境会有如图3.2.1.1的提示信息，需要安装jdk8来运行此烧写工具。

|  |
| --- |
|  |
| 图3.2.1.1 |

1**)** 在图3.2.1.2所示弹出的Java SE安装界面点击 “**下一步**“，之后在图3.2.1.3所示界面继续点击”**下一步**“。

|  |  |
| --- | --- |
|  |  |
| 图3.2.1.2 | 图3.2.1.3 |

**2)** 继续参考下图3.2.1.4所示点击下一步(不建议修改安装目录)，之后会如图3.2.1.5显示安装状态进度，等待进度条显示完成。

|  |  |
| --- | --- |
|  |  |
| 图3.2.1.4 | 图3.2.1.5 |

**3)** 安装完成后如图3.2.1.6所示，点击关闭即可(自此，java运行环境已经安装完成)。

|  |
| --- |
|  |
| 图3.2.1.6 |

**4)** 接下来我们双击执行**SetupSTM32CubeProgrammer-2.1.0.exe** 可执行程序，如图3.2.1.7所示在弹出的界面上点击 “**Next**”,之后在新的页面如图3.2.1.8继续点击“**Next**“。

|  |  |
| --- | --- |
|  |  |
| 图3.2.1.7 | 图3.2.1.8 |

**5)** 在如图3.2.1.9新弹出的页面选中”**I accept the terms of this lincese agreement.**”，继续点击”**Next**”，进入下个界面，如图3.2.1.10，这里提示选择安装路径，保持默认即可，继续点击“**Next**“。

|  |  |
| --- | --- |
|  |  |
| 图3.2.1.9 | 图3.2.1.10 |

**6)** 在图3.2.1.11新的窗口提示 这个目录将会被创建 ，点击”确定”即可，之后进入到安装软件选择界面，这里保持默认即可，点击”**Next**”。

|  |  |
| --- | --- |
|  |  |
| 图3.2.1.11 | 图3.2.1.12 |

**7)** 如图3.2.1.13所示显示烧写工具的安装进度，等待安装完成，安装完成后如图3.2.1.14所示。

|  |  |
| --- | --- |
|  |  |
| 图3.2.1.13 | 图3.2.1.14 |

**8)** 安装完成后提示是否创建开始菜单和桌面快捷方式以及此软件的使用者，这些参考图3.2.1.15所示，使用默认设置，继续点击”**Next**”,最后提示 STM32CubeProgrammer Installation done 表示安装完成，点击 “**Done**”，即可关闭安装界面，之后我们可以在桌面看到多出来一个STM32CubeProgrammer的应用程序 。

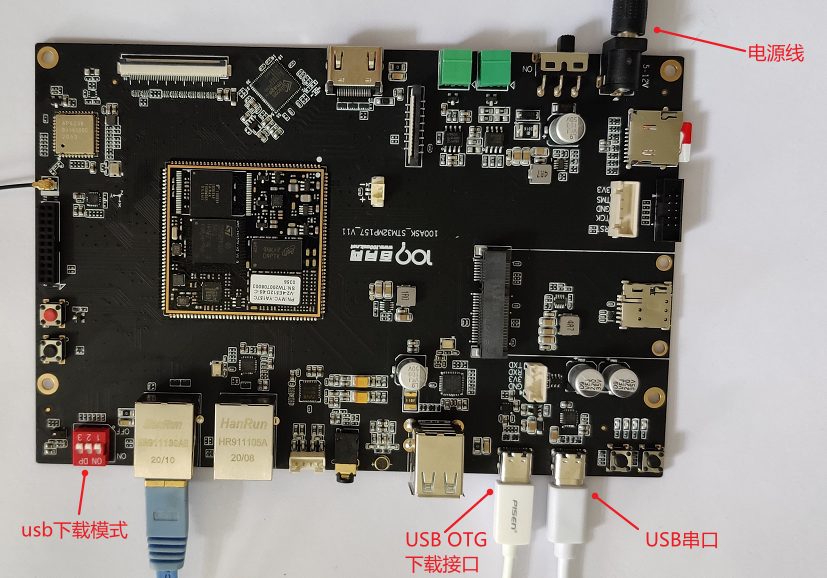
|  |  |
| --- | --- |
|  |  |
| 图3.2.1.15 | 图3.2.1.17 |

#### 3.2.2 连接USB OTG线

接线、设置USB启动的方式如下：

**注意**：USB串口线可接可不接，接上只是为了观察烧写过程。

1. **全功能版接线方式**



#### 3.2.3 安装STM32MP157的USB驱动程序

通过USB下载或是烧写程序时，需要把开发板的OTG口参考3.2.2链接USB　OTG线章节用USB线连接到电脑，并设置为USB启动方式，然后启动电源，此时打开wind10设备管理器，在**通用串行总线设备** 会出来一个名为 “**DFU in HS Mode @Device ID** [**/0x500.@Revision**](mailto:/0x500.@Revision) **ID /0x0000 ”**设备，此时打开STM32CubeProgrammer 烧写工具，如图3.2.3.2所示红色箭头指示，先选择“USB1”再点击刷新按钮，会在 Port 那里显示出来一个 USB1 的设备，这里就表示已经连接成功，可以进行下一步的烧写操作。

|  |  |
| --- | --- |
|  |  |
| 图3.2.3.1 | 图3.2.3.2 |

##### 3.2.3.1 USB驱动没有自动安装

通过USB下载或是烧写程序时，需要把开发板的OTG口用USB线连接到电脑。目前暂未遇到无法安装驱动的问题，如果您在烧写的时候没有设备，请联系百问网相关技术人员以请求支持。

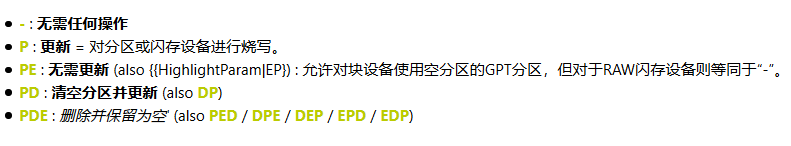
#### 3.2.4 烧写用的配置文件

STM32CubeProgrammer烧写是根据后缀名为 tsv的分区配置文件来匹配分区和文件。

我们可以根据自己的需求来更新自己所需的内容，分区配置信息文件内容简介如下图所示，示例tsv文件解析如下(这只是一个例子)：



其中烧写状态标识区域支持如下图所示5中方式，用户可以根据自己的需要修改相应的烧写状态标识来选择更新/删除某些分区。

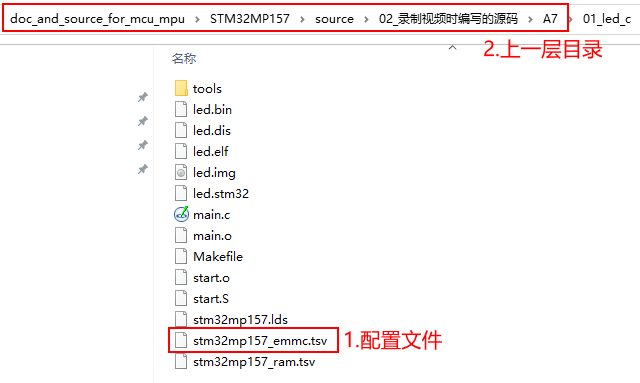


目前仅支持烧写二进制文件，也就是各类映像文件，它们会被原原本本地烧写到EMMC上。

配置文件详细的配置说明请参考页面：<http://wiki.100ask.org/STM32CubeProgrammer_flashlayout>

### 3.3 示例1：下载运行裸机程序(不烧写)

以LED裸机为例，目录如下：



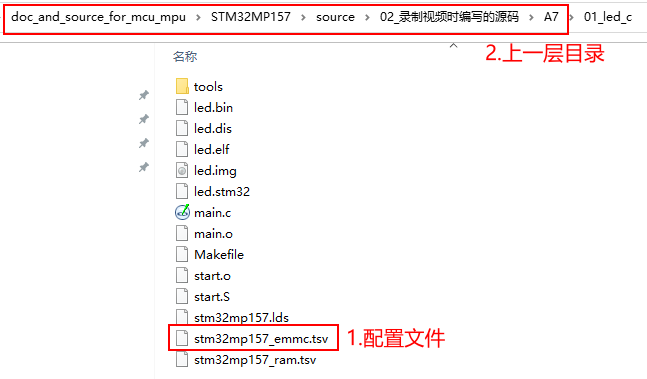
首先，把开发板设置为USB启动，用USB线连接电脑和板子的OTG口，上电。

然后按图操作，一切正常的话可以看到板子上LED闪烁：



### 3.3 示例2：烧写裸机程序到EMMC

以LED裸机为例，目录如下：



首先，把开发板设置为USB启动，用USB线连接电脑和板子的OTG口，上电。

然后按图操作，烧写成功后，把板子设为EMMC启动，重新上电可以看到LED闪烁：

