



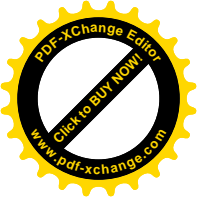
ASR6601

Program Development Quick Start Guide

Version: 1.3.1

Date: 2024-03-11

Copyright © 2024 ASR Technology



About this document

This document describes the development environment settings and programming of the IoT LPWAN SoC chip ASR6601 SDK, which helps users to quickly develop programs for ASR6601.

Target Audience

This document is mainly intended for the following engineers:

- hardware development engineer
- software engineer
- technical support engineer

Product numbering

Product models corresponding to this document:

Model	Flash	SRAM	Core	Package	Frequency
ASR6601SE	256 KB	64 KB	32-bit 48 MHz Arm China STAR-MC1 Processor	QFN68, 8*8 mm	150 ~ 960 MHz
ASR6601CB	128 KB	16 KB	32-bit 48 MHz Arm China STAR-MC1 Processor	QFN48, 6*6 mm	150 ~ 960 MHz
ASR6601SER	256 KB	64 KB	32-bit 48 MHz Arm China STAR-MC1 Processor	QFN68, 8*8 mm	150 ~ 960 MHz
ASR6601CBR	128 KB	16 KB	32-bit 48 MHz Arm China STAR-MC1 Processor	QFN48, 6*6 mm	150 ~ 960 MHz

Copyright Notice

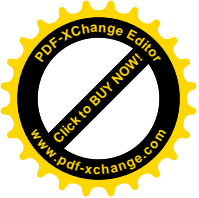
Copyright © 2024 ASR Technology Co., Ltd. All rights reserved. No part or all of this document may be reproduced, transmitted, transcribed, stored or translated in any form or by any means without the written permission of ASR Technology Co., Ltd.

Trademark Notice



ASR, ASR and other ASR logos are trademarks of ASR Technology Co., Ltd.

All other trade names, trademarks and registered trademarks mentioned in this document are the property of their respective owners and are hereby acknowledged.



Disclaimer

ASR Technology Co., Ltd. makes no warranty of any kind for the contents of this document and will update the contents of this document or the products described in this document from time to time.

This document is only used as a guide for use, and all contents of this document do not constitute any form of warranty. The information in this document is subject to change without notice.

This document does not assume any responsibility, including responsibility for infringement of any proprietary rights arising from the use of the information in this document.

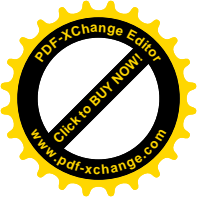
ASR Technology Ltd.

Address: 9th Floor, Building 10, Zhangjiang Innovation Park, 399 Keyuan Road, Pudong New District, Shanghai Postal Code: 201203

URL : <http://www.asrmicro.com/>

Document Revision History

Date	Version	Release Notes
2020.05	V0.1.0	Initial Release.
2020.08	V0.2.0	<ul style="list-style-type: none">● Added Keil environment development instructions.● Added new Q&A content.
2020.09	V0.3.0	Updated images.
2020.10	V0.4.0	Updated with ASR6601SE-EVAL v2.0 diagram.
2021.01	V1.1.0	Removed the overview from Chapter 1 and merged its contents into the "About This Document" section of the preface.
2021.03	V1.2.0	Modified Chapter 3 to simplify the compilation process.
2022.06	V1.3.0	Updated the methods for obtaining ASR SDK and GCC toolchain.
2024.03	V1.3.1	Updated description of Section 2.1.



目录

1. 硬件准备	1
1.1 ASR6601 开发板说明	1
1.2 跳线连接	3
2. 使用 KEIL 开发环境	4
2.1 连接 J-Link	4
2.2 获取 SDK	4
2.3 生成 KEIL 工程文件	4
2.4 配置 GCC 工具链	5
2.5 配置 Flash Programming Algorithm	6
2.6 编译与烧录	6
3. 使用 Make 命令行	7
3.1 准备	7
3.1.1 开发环境安装	7
3.1.2 SDK 获取	7
3.2 软件编译与烧录	8
3.2.1 编译工程	8
3.2.2 烧录	9
3.2.3 运行	10
4. Q&A	11
4.1 使用 KEIL 烧录时，一直没有出现 SW Deice，怎么办？	11
4.2 使用 MSYS2 环境进行烧录时，找不到对应的串口设备，怎么办？	11
4.3 使用 MSYS2 环境进行烧录时，能看到串口设备，但烧录一直不成功，怎么办？	11



表格

表 1-1 ASR6601SE-EVAL v2.0 接口说明.....	2
表 1-2 跳线连接状态.....	3

ASR Confidential



插图

图 1-1 ASR6601SE-EVAL v2.0 正面	1
图 1-2 ASR6601SE-EVAL v2.0 反面	2
图 2-1 SWD 接口定义	4
图 2-2 配置 GCC 工具链	5
图 2-3 配置 Flash Programming Algorithm	6
图 3-1 进入下载模式的示意图	9

ASR Confidential

1.

硬件准备

LoRa 节点必需硬件列表如下：

- (1) ASR6601 开发板 1 个
- (2) 天线 1 根
- (3) USB 线 1 根
- (4) PC 机 1 台

1.1 ASR6601 开发板说明

开发板 ASR6601SE-EVAL v2.0 的正反面如图 1-1 和图 1-2 所示：

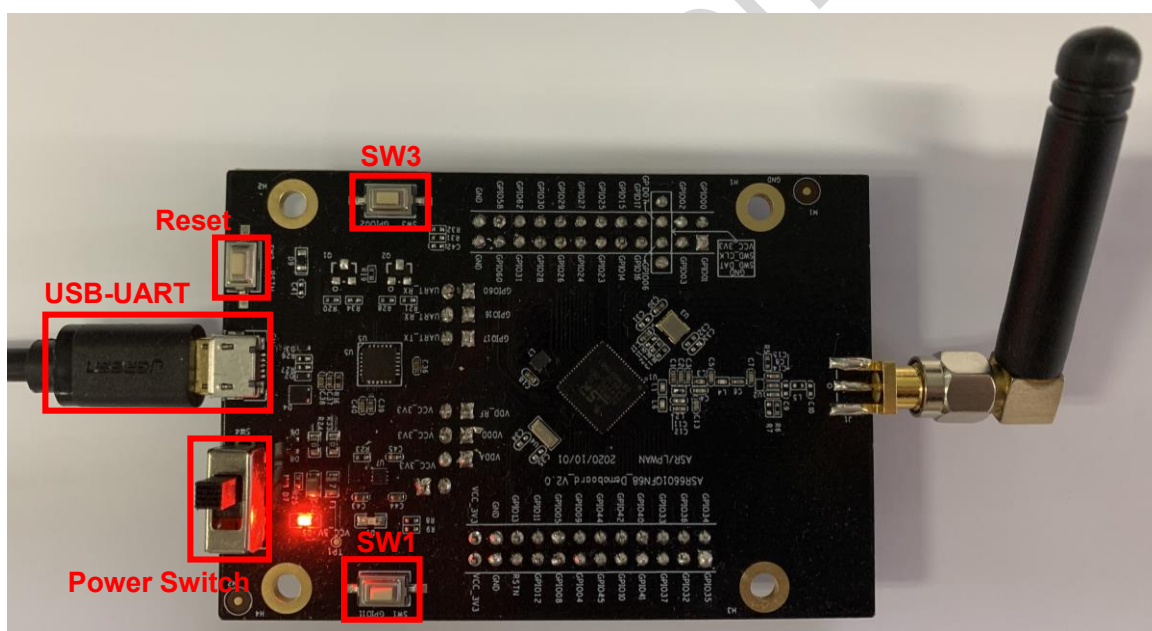


图 1-1 ASR6601SE-EVAL v2.0 正面

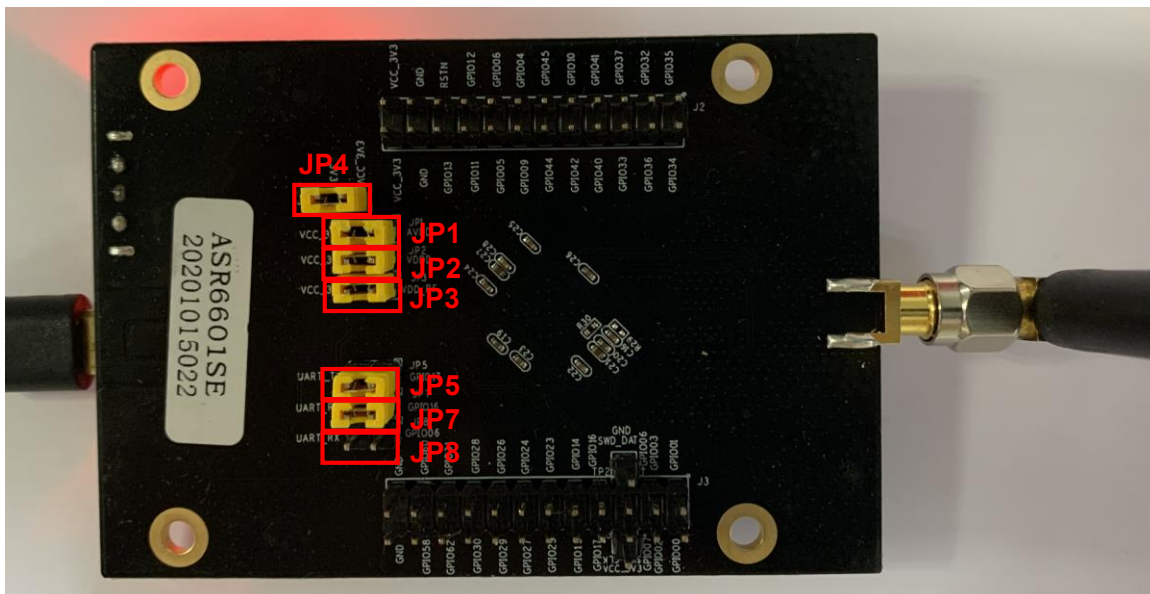
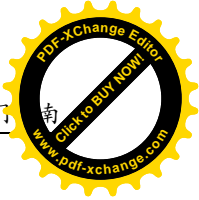
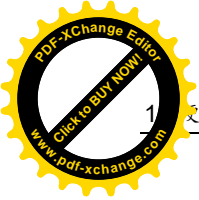


图 1-2 ASR6601SE-EVAL v2.0 反面

表 1-1 ASR6601SE-EVAL v2.0 接口说明

接口	描述
USB-UART	USB 转串口
Power Switch	电源开关
Reset	Reset 按钮
SW3	Download 按钮，按下后，GPIO02 拉高
SW1	User 按钮，按下后，GPIO11 拉低
JP1	电源跳线
JP2	电源跳线
JP3	电源跳线
JP4	电源跳线，可测试板子总功耗
JP5	UART_TX 跳线，跳线连通选择 UART0_TX，具体请参考原理图
JP6（仅存在于 ASR6601CB-EVAL）	UART_TX 跳线，跳线连通选择 LPUART_TX，具体请参考原理图
JP7	UART_RX 跳线，跳线连通选择 UART0_RX，具体请参考原理图
JP8	UART_RX 跳线，跳线连通选择 LPUART_RX，具体请参考原理图



1.2 跳线连接

在进行 ASR6601 开发板测试过程中，请保证下面跳线的状态正确。

表 1-2 跳线连接状态

跳线	连接状态
JP1	连通
JP2	连通
JP3	连通
JP4	连通
JP5	连通
JP6（仅存在于 ASR6601CB-EVAL）	断开
JP7	连通
JP8	断开



2. 使用 KEIL 开发环境

2.1 连接 J-Link

VTref	1 ●	● 2	3V3
NC	3 ●	● 4	GND
NC	5 ●	● 6	GND
SWDIO	7 ●	● 8	GND
SWCLK	9 ●	● 10	GND
NC	11 ●	● 12	GND
SWO	13 ●	● 14	GND
RESET	15 ●	● 16	GND
NC	17 ●	● 18	GND
5V	19 ●	● 20	GND

SWD接口定义

图 2-1 SWD 接口定义

ASR6601 使用 J-Link 时需要连 4 根线，将上图中 1、7、9 和 20 脚连接到板子的对应 pin 脚即可。需注意，禁止将 J-LINK 的 RESET 引脚连接到 ASR6601 的 RESET 引脚上，可能会导致 flash 擦除的风险。

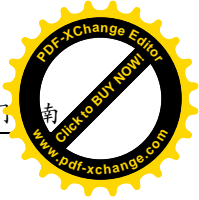
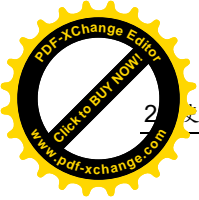
2.2 获取 SDK

可以联系 ASR 技术支持人员来获取，或者通过下方命令从 GitHub 上下载。

```
git clone https://github.com/asrlora/asr_lora_6601.git
```

2.3 生成 KEIL 工程文件

SDK 中未提供 KEIL 工程文件，可以运行示例程序中的 keil.bat 生成 KEIL 工程文件。



2.4 配置 GCC 工具链

- (1) 通过下方链接下载 GNU Arm Embedded Toolchain，然后解压缩。

<https://developer.arm.com/-/media/Files/downloads/gnu-rm/9-2020q2/gcc-arm-none-eabi-9-2020-q2-update-win32.zip>

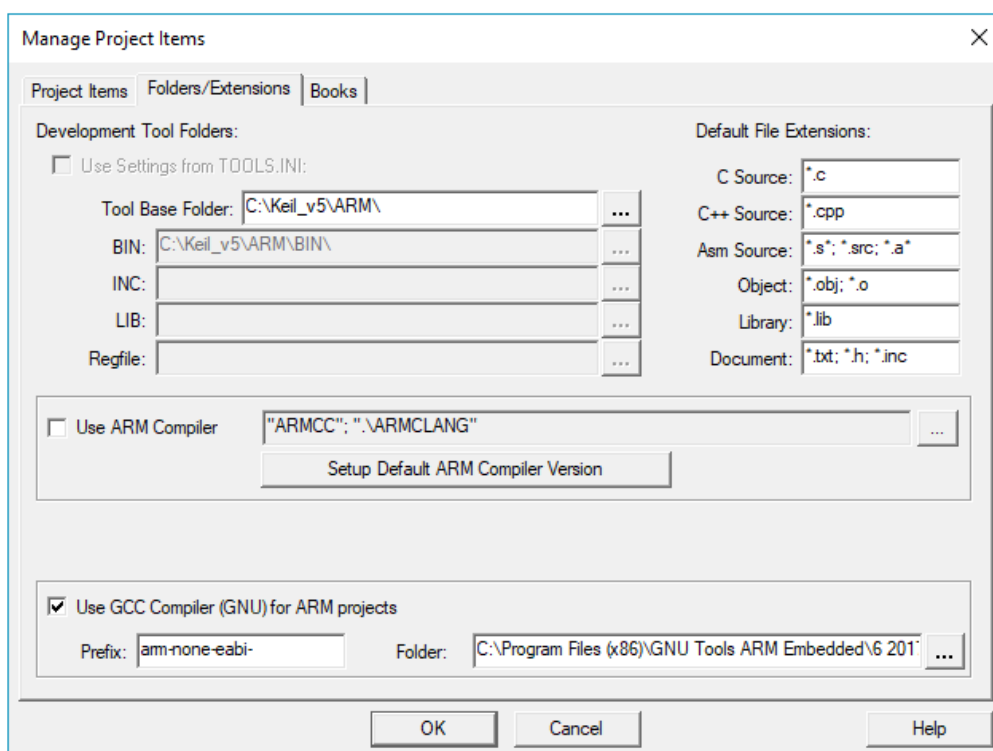
- (2) 按照 KEIL 的用户指南文档设置 GCC 工具链，文档链接如下：

https://www.keil.com/support/man/docs/uv4/uv4_gnucomp.htm

设置 Tool Base Folder 为刚解压后的目录，例如：D:\ASR6601_rel\tools\toolchain

To use GCC, download the **GNU Arm Embedded** toolchain, and then:

1. Open the dialog **Project - Manage - Project Items - Folders/Extensions**.



2. Enable Use GCC Compiler (GNU) for Arm projects. Refer to [Folders/Extensions](#) for field descriptions.

图 2-2 配置 GCC 工具链

2.5 配置 Flash Programming Algorithm

首先，将 \tools\FLM 目录下的 ASR6601.FLM 文件复制到 Keil 的 Flash 目录，例如：
C:\Keil_v5\ARM\Flash。

如果仍然无法烧录，请按照 KEIL 的用户指南文档修改 Flash Download 配置，文档链接如下：
https://www.keil.com/support/man/docs/uv4/uv4_fl_dlconfiguration.htm，主要配置如下内容：

- (1) **Download Function**: 勾选 “Erase Sectors”、“Program” 和 “Verify”
- (2) **RAM for Algorithm**: 配置 Start 为 0x20000000，Size 为 0x2000
- (3) **Programming Algorithm**: 添加 ASR6601 的 Flash Programming Algorithm 文件。

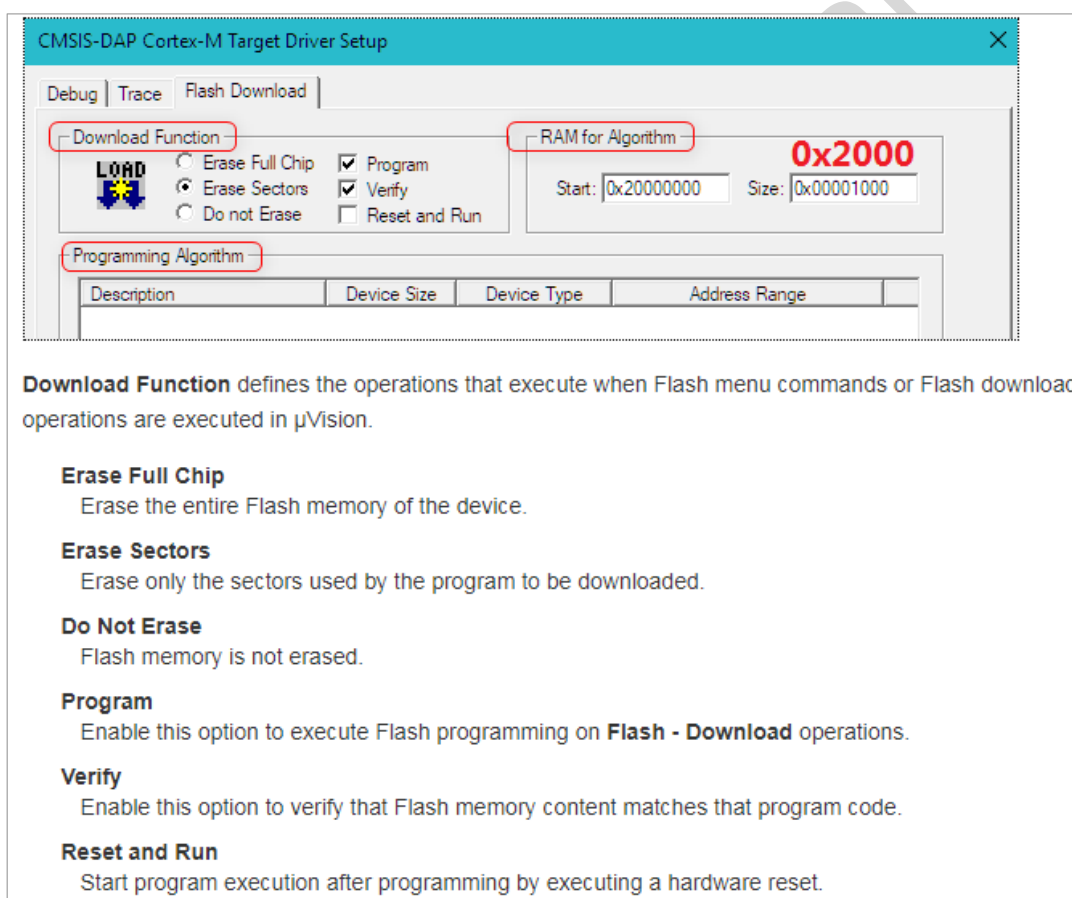
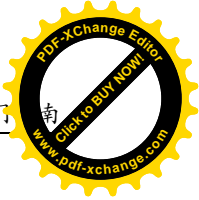
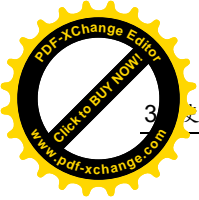


图 2-3 配置 Flash Programming Algorithm

2.6 编译与烧录

- (1) 点击“Build”按钮进行编译
- (2) 点击“Download”按钮进行烧录



3. 使用 Make 命令行

3.1 准备

3.1.1 开发环境安装

3.1.1.1 Ubuntu 环境 (Ubuntu18.04)

运行下面命令安装 Python 及其他必要软件：

```
sudo apt-get install gcc-arm-none-eabi git vim python python-pip  
pip install pyserial configparser
```

3.1.1.2 Windows 环境

(1) 安装 MSYS2

请登录官网 (<https://www.msys2.org/>) 下载安装包，并按照要求进行安装。

(2) 安装相关程序

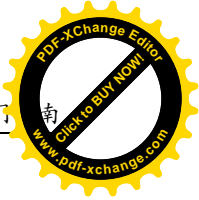
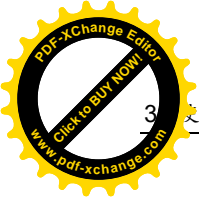
打开 MSYS2，然后安装相关程序：`pacman -S git vim make unzip python python-pip wget`

使用 pip 安装 pyserial：`pip install pyserial configparser`

3.1.2 SDK 获取

可以联系 ASR 技术支持人员来获取，或者通过下方命令从 GitHub 上下载。

```
git clone https://github.com/asrlora/asr_lora_6601.git
```



3.2 软件编译与烧录

下面以 `uart_printf` 工程为例讲解软件的编译与烧录过程。

3.2.1 编译工程

按如下步骤编译工程：

- (1) 执行下面命令配置环境变量：`source build/envsetup.sh`
- (2) 进入 `uart_printf` 目录：`cd projects/ASR6601CB-EVAL/examples/uart/uart_printf`
- (3) 执行 `make` 命令编译程序：`make`

编译成功后，显示结果如下：

```
Build completed.
```

```
arm-none-eabi-size out/uart_printf.elf
```

text	data	bss	dec	hex	filename
9972	1080	4164	15216	3b70	out/uart_printf.elf

Please run 'make flash' or the following command to download the app

```
python /home/ruilinhao/work/ASR6601_rel/build/scripts/tremo_loader.py -p /dev/ttyUSB0 -b 921600 flash 0x08000000 out/uart_printf.bin
```

3.2.2 烧录

烧录有两种方式：

- 使用烧录工具进行烧录，具体可以参考文档 [《ASR6601_烧录工具使用说明》](#)。
- 使用命令行烧录。

下面重点介绍命令行烧录的步骤：

(1) 串口配置

首先执行命令 `ls /dev/` 来查看开发板使用的串口。

通常在 `MSYS2` 下面，会有 `ttyS*` 设备，即为串口设备，其与 Windows 下面的 COM 端口号有对应关系，如 COM6 对应在 `MSYS2` 中为 `/dev/ttyS5`；在 Ubuntu 下面串口设备通常为 `/dev/ttyUSB*`。

找到串口设备后，修改 `uart_printf` 工程的 `Makefile`，去除 `SERIAL_PORT` 前面的“#”符号，并将 `SERIAL_PORT` 修改成对应的串口号。

`SERIAL_BAUDRATE` 和 `$(PROJECT)_ADDRESS` 如无特殊需求，可以不修改，使用默认值。

```
SERIAL_PORT      :=/dev/ttyS5
#SERIAL_BAUDRATE :=
#$(PROJECT)_ADDRESS :=
```

(2) 进入下载模式

烧录前，请按住板子上的 `SW3` 按钮，使 `GPIO02` 拉高，然后点击 `Reset` 按钮重启，进入下载模式。

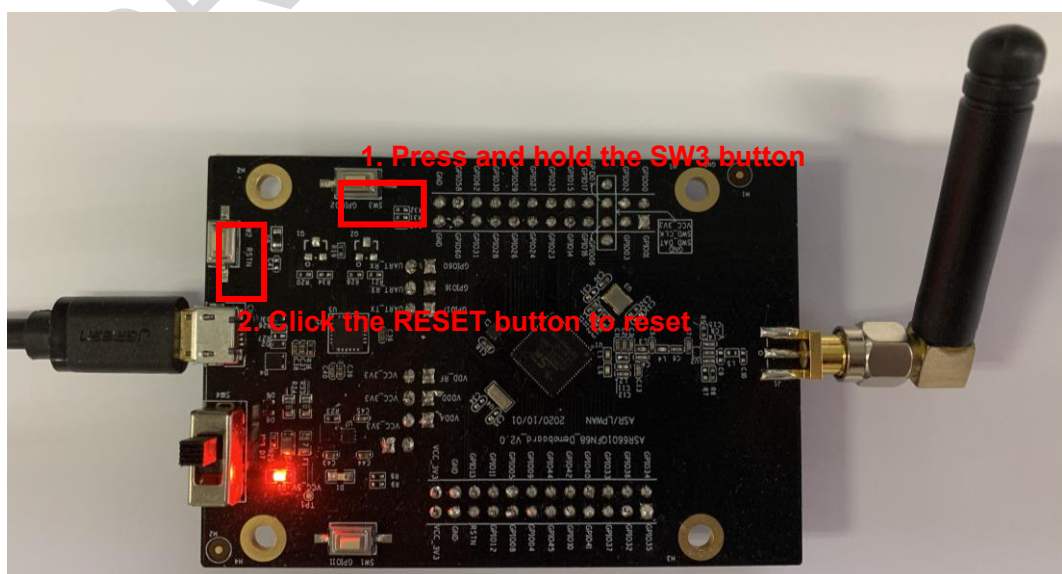
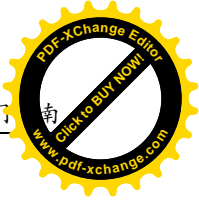
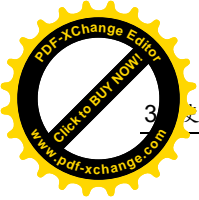


图 3-1 进入下载模式的示意图



(3) 执行烧录

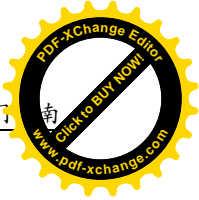
最后执行 `make flash` 命令或者使用 `tremo_loader.py` 自定义命令进行烧录。

如烧录成功，会显示如下信息。如一直无法烧录成功，请参考第 4 章的相关 QA 内容。

```
Connecting...
Connected
('send: ', 512)
('send: ', 1024)
('send: ', 1536)
('send: ', 2048)
('send: ', 2560)
('send: ', 3072)
('send: ', 3584)
('send: ', 4096)
('send: ', 4100)
Download files successfully
```

3.2.3 运行

烧录完成后，重启即可顺利运行程序。串口工具界面打印：`hello world`



4.

Q&A

4.1 使用 KEIL 烧录时，一直没有出现 SW Deice，怎么办？

没有出现 SW Device 可能是由以下两种情况造成的：

1. 硬件连接问题，需检查接线和电源等硬件连接。
2. 出现硬故障或者 MCU 进入低功耗等情况，会导致 SW device 没有出现，此时可以用杜邦线将 GPIO02 pin 拉高，然后重启，使 MCU 进入 bootloader，即可使 SW device 出现并可再次烧录。

4.2 使用 MSYS2 环境进行烧录时，找不到对应的串口设备，怎么办？

MSYS2 中有最大串口设备数量（版本不同，可能是 64 或者 128），如果串口设备端口号过大，在 MSYS2 中就会找不到，可以把串口号改小一点，就可以在 MSYS2 中出现了。

4.3 使用 MSYS2 环境进行烧录时，能看到串口设备，但烧录一直不成功，怎么办？

1. 检查是否有其他软件打开了该串口，如串口工具等。
2. 在某些 Windows 版本中，直接使用 `/dev/ttyS*` 会失败，可以尝试在 Makefile 中将 `SERIAL_PORT` 修改成 `COM*`。