



Witmem
知存科技

基于存内计算芯片开发板

验证语音识别



——训练手册

2023/10/21

目录

实验环境搭建

模块一：软件包下载及环境搭建

动手实验：简单语音识别系统开发

模块一：简介

模块二：搭建算法训练工程，完成算法训练与量化

模块三：算法模型转换

模块四：算法模型烧写

模块五：算法模型在芯片运行推理

实验环境搭建

模块一：软件包下载及环境搭建

步骤一：搭建 docker、mappper 环境：

①Linux 下通过 Docker 直接下载，获取指令：

```
docker pull witin/toolchain:v001.000.034
```

②Window 环境，可以通过 docker desktop 来使用 docker:

1) 下载安装 Docker desktop(win10 或以上)：

[Docker Desktop: The #1 Containerization Tool for Developers | Docker](#)

2) 通常需要更新 WSL，下载链接如下，更新后需要重启生效

[旧版 WSL 的手动安装步骤 | Microsoft Learn](#)

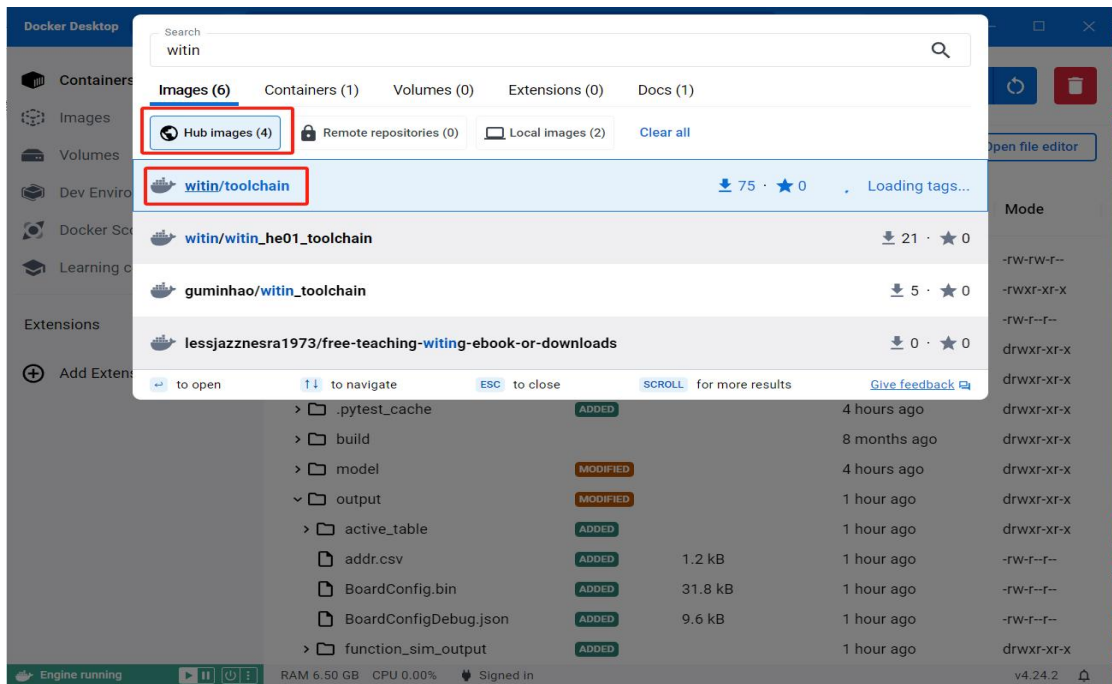
3) Docker desktop 基本使用教程：

[Docker-desktop\(Docker 桌面版\)——入门篇_dockerdesktop 干嘛用的-CSDN 博客](#)

4) Docker desktop 通常默认安装在 c:\Program File\docker，可以通过软连接的形式修改 Docker 安装路径：

[如何将 Docker \(Windows 桌面版\)自定义安装目录_自定义 docker 安装路径-CSDN 博客](#)

5) 在 Docker desktop 里，可以通过搜索获得 witin_toolchain，我们需要的是 034 版本（ps:Hub 反应慢可以开 VPN 获取，或者使用镜像路径，具体操作方式见 3 链接）



5) 测试：

①：管理员模式下打开命令行窗口

docker run -it --name XXXX witin/toolchain:v001.000.034

②：默认进入 workspace 目录下，可以进入 witin_mapper 下执行测试脚本：

cd witin_mapper

python3

tests/python/frontend/onnx/witin/wtm2101/precision/XXXX.py

```
管理员: 命令提示符 - docker attach 92f68360bd03
root@92f68360bd03:/workspace/witin_mapper# python3 tests/python/frontend/onnx/witin/wtm2101/precision/test_forward_TDNN.py
[07:14:16] [INFO]low_level_mapping/src/session/pass/array_extend_pass.cpp:201: Array Alloc: have set reserved column 128[07:
14:16] [INFO]low_level_mapping/src/memory/array/array_alloc_interface.cpp:43: Array Alloc: have set reserved column 128
[07:14:16] [INFO]low_level_mapping/src/session/pass/array_extend_pass.cpp:201: Array Alloc: have set reserved column 1282023
9 18 7 14 19
create ./output/output_test_forward_TDNN18_Oct_2023_07_14_16_570292/map//array_space.png
[07:14:22] [WARNING]witin_mapper_pro/witin_mapper/src/runtime/graph/npu_graph_runtime.cc:434: Warning: cannot find "in" amon
g input, have input node:
0 : in_kws;
use default input 0
[07:14:22] [WARNING]witin_mapper_pro/witin_mapper/src/runtime/graph/npu_graph_runtime.cc:434: Warning: cannot find "in" amon
g input, have input node:
0 : in_kws;
use default input 0
[07:14:22] [INFO]low_level_mapping/src/session/pass/array_extend_pass.cpp:201: Array Alloc: have set reserved column 128[07:
14:22] [INFO]low_level_mapping/src/session/pass/array_extend_pass.cpp:201: Array Alloc: have set reserved column 128[07:14:2
2] [INFO]low_level_mapping/src/memory/array/array_alloc_interface.cpp:43: Array Alloc: have set reserved column 128
Enter json mode!
/workspace/witin_mapper/witin_sim/BB04P1/filter.config
model version: v001.000.034
model name: model_1_net_in_kws_tdn9
chip_id: ChipId::BB04P1
data: 2023-10-18-7-14-18
net count: 1
npu_simulator cost 850.6357940000ms
func simulation result is equal to cmodel result!
root@92f68360bd03:/workspace/witin_mapper#
```

③：使用 exit 退出，再次进入可按如下操作：

④：通过 docker ps -a 获取容器 id，然后打开进入容器

```
管理员: 命令提示符 - docker attach 92f68360bd03
C:\WINDOWS\system32>docker run -it --name witin_toolchain_34 witin/toolchain:v001.000.034
root@92f68360bd03:/workspace# exit
exit

C:\WINDOWS\system32>docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS   NAMES
92f68360bd03   witin/toolchain:v001.000.034        "/bin/bash"            About a minute ago    Exited (0) 31 s     witin_toolchain_34
econds ago
140bc8ae49a6   witin/toolchain:v001.000.021        "/bin/bash"            4 hours ago         Exited (0) 2 ho     witin_toolchain
urs ago


C:\WINDOWS\system32>docker start 92f68360bd03
92f68360bd03

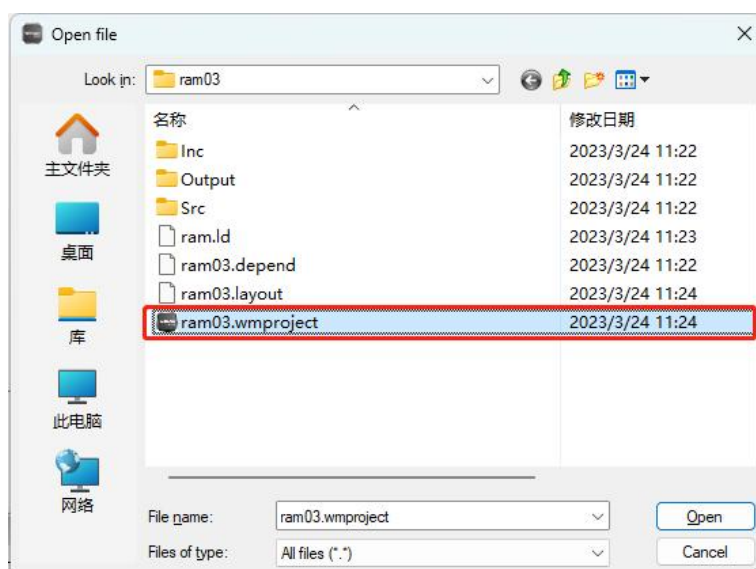
C:\WINDOWS\system32>docker attach 92f68360bd03
root@92f68360bd03:/workspace#
```

步骤二：搭建 IDE 环境

①预先下载安装包，下载安装

②默认安装路径，直接点击安装

③从主菜单的 File->Open…菜单，或者点击工具栏的  按钮，打开选择文件对话框，选择要打开的项目文件，即***.wmpoject 文件即可，点击【打开】按钮即可打开工程。如下图所示。



步骤三：其他需要安装的软件

- ① 请确保安装以下环境：python, git, pytorch

步骤四：下载训练数据

- ① 最新数据集在软件安装包中

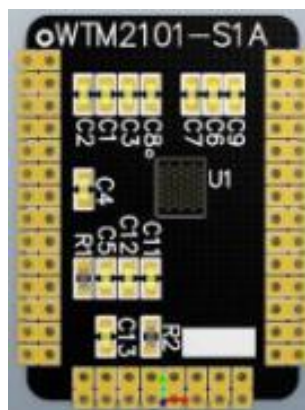
动手实验：简单语音识别系统开发

模块一：简介

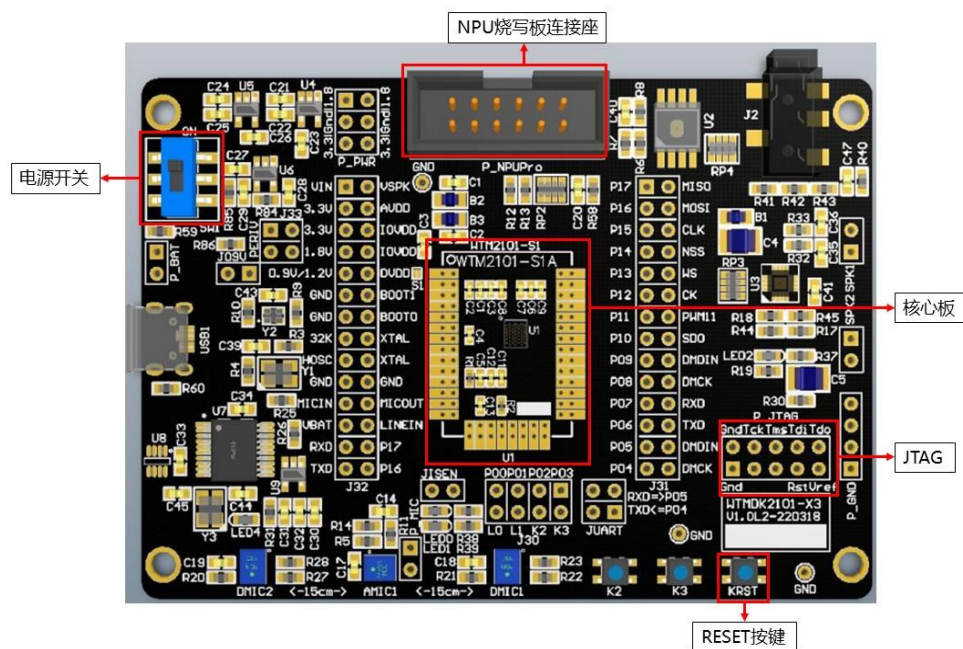
1、WTMDK2101-X3 介绍

WTMDK2101-X3 是针对 WTM2101 AI SOC 设计的评估板，包含：

- (1) WTM2101 核心板，即我们的存算芯片。
- (2) 和 I/O 板：WTM2101 运行需要的电源、以及应用 I/O 接口等。



核心板示意图



WTMDK2101-X3 I/O 板示意图

2, AISHELL-WakeUp-1 数据集介绍

AISHELL-WakeUp-1 数据集是中英文唤醒词语音数据库，命令词为“你好，米雅”“hi, mia”，语音数据库中唤醒词语音 3936003 条，1561.12 小时，邀请 254 名发言人参与录制。录制过程在真实家居环境中，设置 7 个录音位，使用 6 个圆形 16 路 PDM 麦克风阵列录音板做远讲拾音(16kHz, 16bit)、1 个高保真麦克风做近讲拾音(44.1kHz, 16bit)。此数据库可用于声纹识别、语音唤醒识别等研究使用。

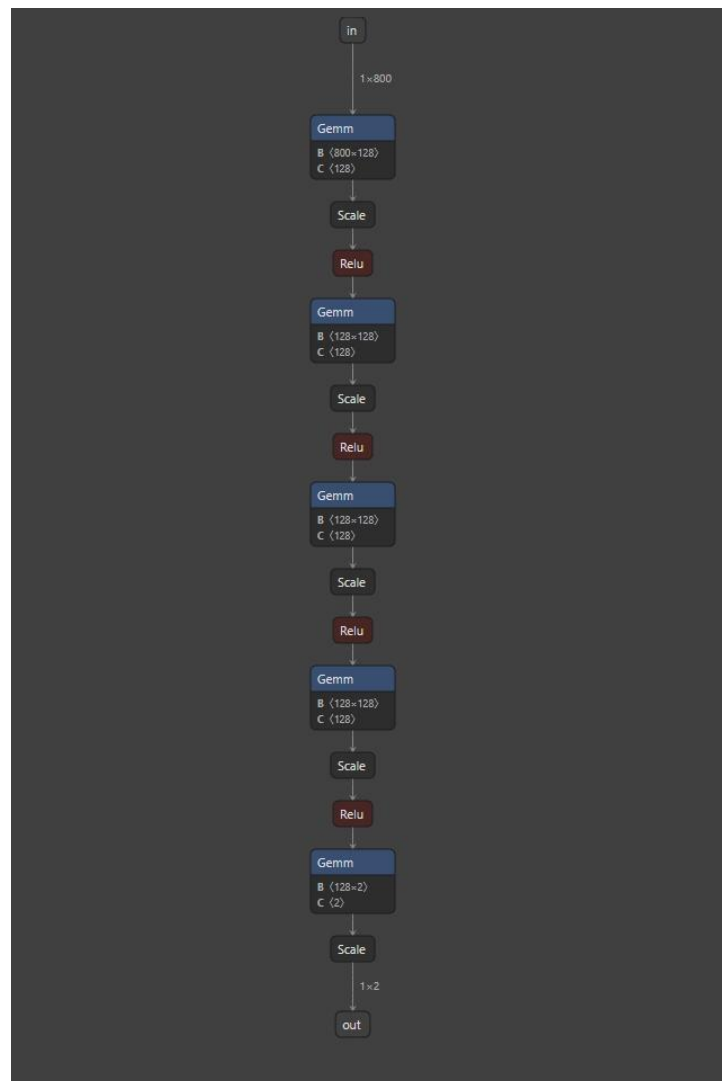
本 demo 以该数据集为例，用不同网络结构展示模型训练及移植过程。

模块二：搭建算法训练工程，完成算法训练与量化

1. 模型训练工程搭建：

本 demo 提供 DNN 和 DNN_DEEP 两种网络结构的示例，网络结构如下，

本教程以 DNN 为例。



DNN

步骤一：配置 python/config.py，参数释义见代码注释。

步骤二：运行 python/train.py，模型训练完毕后，在 models/net_type 文件夹下生成 bestModel.pth，此即我们的模型权重。

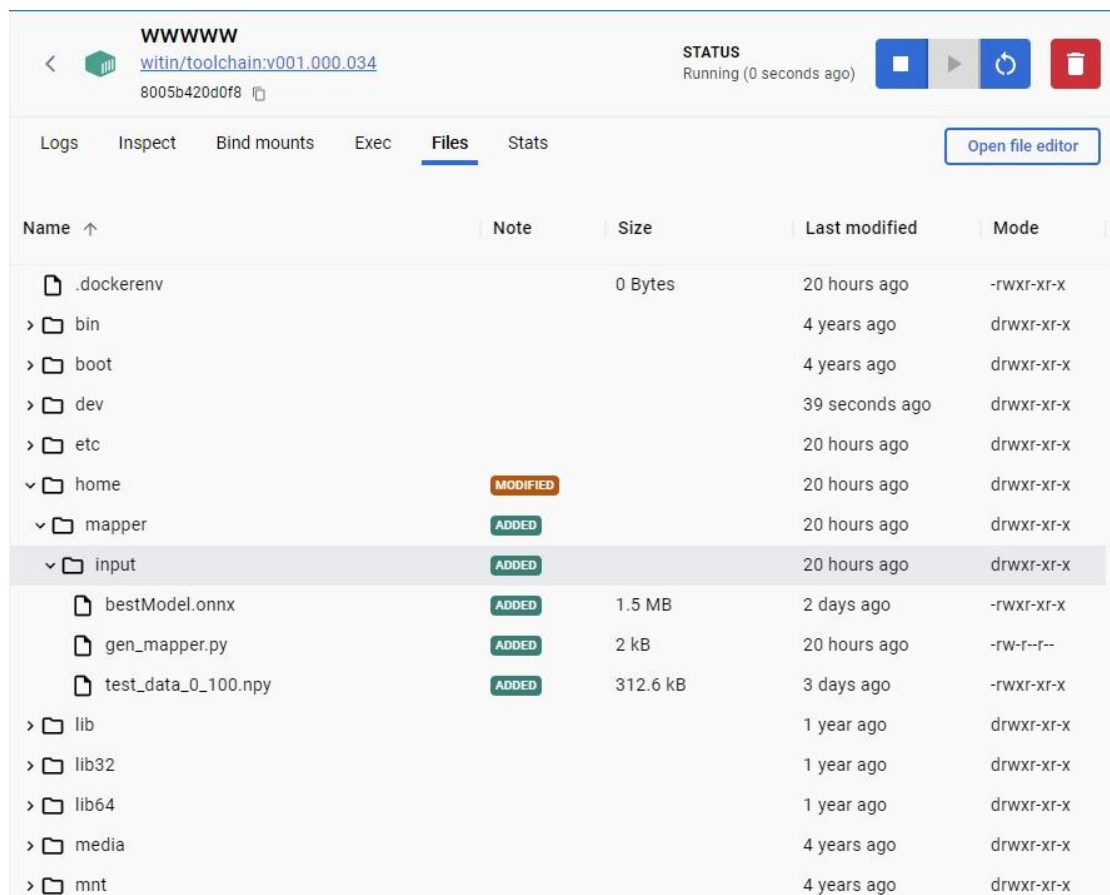
步骤三：运行 python/onnx_converter.py，在 models/net_type 文件夹下生成 bestModel.onnx。此步骤即完成原始模型到知存 onnx 格式模型的转换。

模块三：算法模型转换

1, Docker 下 Mapper 转换流程

步骤一：拷贝至指定文件夹

我们将 mapper/input 拷贝至 witin/toolchain:v001.000.034 的指定文件夹下（通常为/home，需与 gen_mapper.py 文件里描述一致）



Name	Note	Size	Last modified	Mode
.dockerenv		0 Bytes	20 hours ago	-rwxr-xr-x
bin			4 years ago	drwxr-xr-x
boot			4 years ago	drwxr-xr-x
dev			39 seconds ago	drwxr-xr-x
etc			20 hours ago	drwxr-xr-x
home	MODIFIED		20 hours ago	drwxr-xr-x
mapper	ADDED		20 hours ago	drwxr-xr-x
input	ADDED		20 hours ago	drwxr-xr-x
bestModel.onnx	ADDED	1.5 MB	2 days ago	-rwxr-xr-x
gen_mapper.py	ADDED	2 kB	20 hours ago	-rw-r--r--
test_data_0_100.npy	ADDED	312.6 kB	3 days ago	-rwxr-xr-x
lib			1 year ago	drwxr-xr-x
lib32			1 year ago	drwxr-xr-x
lib64			1 year ago	drwxr-xr-x
media			4 years ago	drwxr-xr-x
mnt			4 years ago	drwxr-xr-x

步骤二：在 workplace\witin_mapper 下执行 gen_mapper.py

docker start id

docker attach id

cd witin_mapper

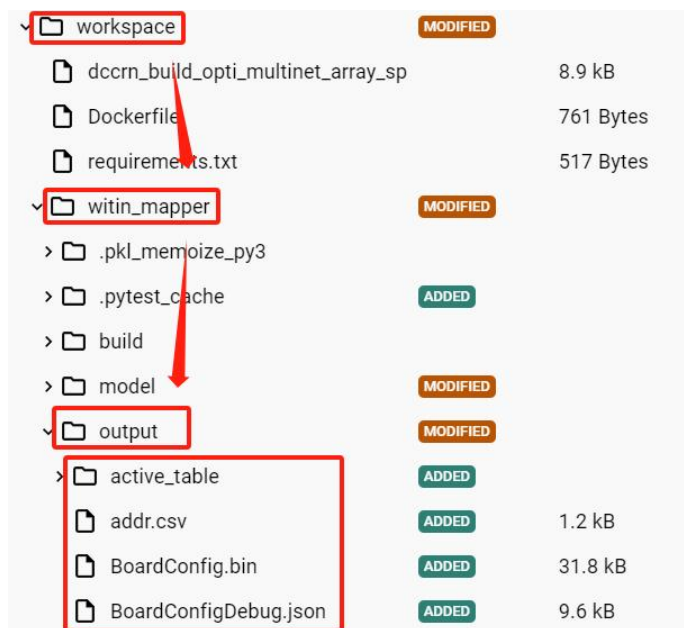
python3 /home/mapper/input/gen_mapper.py

管理员: 命令提示符 - docker attach 140bc8ae49a6

```
C:\WINDOWS\system32>docker start 140bc8ae49a6
140bc8ae49a6

C:\WINDOWS\system32>docker attach 140bc8ae49a6
root@140bc8ae49a6:/workspace# cd witin_mapper
root@140bc8ae49a6:/workspace/witin_mapper# python3 /home/mapper/input/gen_mapper.py
Warning: Checker does not support models with experimental ops: Scale
Warning: Checker does not support models with experimental ops: Scale
Warning: Checker does not support models with experimental ops: Scale
Warning: Checker does not support models with experimental ops: Scale
src shape: (100, 28, 28, 1)
[07:21:06] [INFO] low_level_mapping/src/op/conv2d.cpp:59: Conv_4:padding right has cha
[07:21:06] [INFO] low_level_mapping/src/op/conv2d.cpp:63: Conv_4:padding bottom has ch
[07:21:06] [INFO] low_level_mapping/src/op/conv2d.cpp:59: Conv_2:padding right has cha
[07:21:06] [INFO] low_level_mapping/src/op/conv2d.cpp:63: Conv_2:padding bottom has ch
[07:21:06] [INFO] low_level_mapping/src/tensor/array_alloc_cp.cpp:127: Array Alloc: ha
[07:21:12] [INFO] low_level_mapping/src/session/util.cpp:2239: ./output//map/map.json
[07:21:17] [WARNING] low_level_mapping/src/target/rpt/base_codegen.cpp:510: WARNING:"C
ray additional data use diff_strategist tool.
create ./output/map//array_space.png
root@140bc8ae49a6:/workspace/witin_mapper#
```

步骤三：在对应的 output 文件下获得输出

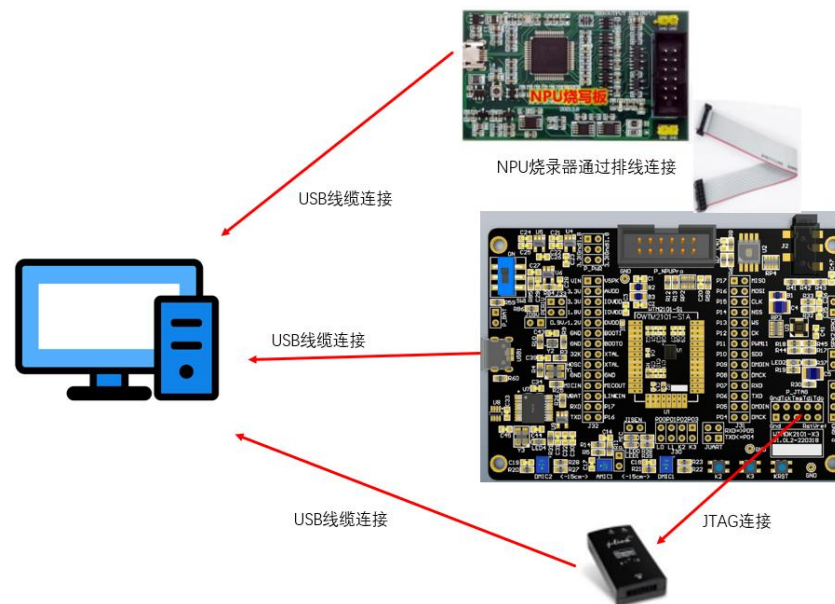


模块四：算法模型烧写

步骤一：系统连接：

进行模型烧录和开发时，我们需要将 JTAG，核心板，NPU 烧写板连接好，并打开开关，如系统连接示意图所示。

步骤二：

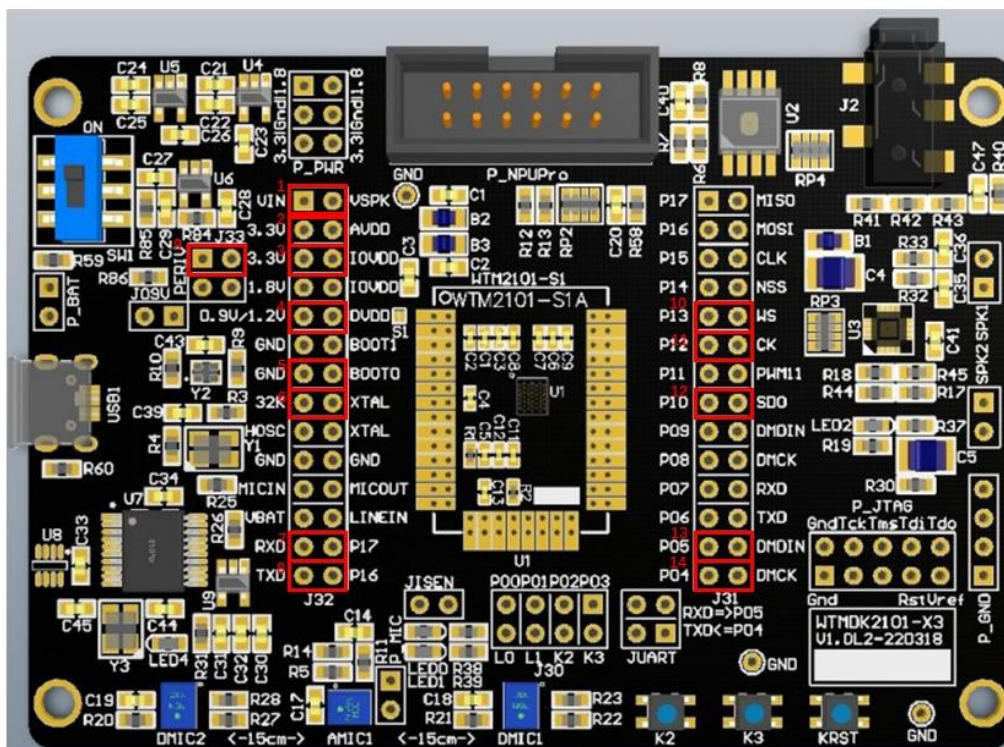


系统连接示意图

步骤三：跳线帽连接：

如跳线帽连接示意图所示，按照红框标注进行跳线连接。含义解释：

跳线	编号	跳帽连接	含义
1	VIN	VSPK	Audio DAC 芯片供电，5V
2	3.3V	AVDD	WTM2101 芯片模拟供电，3.3V
3	3.3V	IOVDD	WTM2101 芯片 I/O 供电，3.3V
4	0.9/1.2V	DVDD	用 WTM2101BC 芯片时需接跳帽
5	GND	BOOT0	启动模式，SRAM 启动
6	32K	XTAL	晶振
7	RXD	P17	串口
8	TXD	P16	
9	PERIV	3.3V	QSPI Flash、数字麦克风、晶振供电，与 IOVDD 选择一致，即 3.3V
10	P13	WS	I2S 功放的 WS
11	P12	CK	I2S 功放的 CK
12	P10	SDO	I2S 功放的 SDO
13	P05	DMDIN	数字麦克风的 Din
14	P06	DMCK	数字麦克风的 CK



跳线帽连接示意图

步骤四：

使用 project/ WitinProgramTool_WTM2101 下的 WitinProgramTool.exe 进行模型权重烧写。烧写时的开发板接线请参考其他文档。

步骤五：烧写指令:

`.\WitinProgramTool.exe -m init`

`.\WitinProgramTool.exe -m program -i XXXX\map.csv -k 2`

其中 XXXX 为步骤 2. (3) 中生成的 mapper/output/map

示例:

```

PS D:\code\mnist_demo_lstm_v3\mnist_demo_v3\segger\WitinProgramTool_WTM2101> .\WitinProgramTool.exe -m program -i D:\co
de\mnist_demo_lstm_v3\mnist_demo_v3\mapper\output\map\map.csv -k 2
Downloading Firmware ... Done!
*****
** Witmem Program Tool for WTM2101 V1.3.1.0 **
** Build Time: 2023.01.17 **
*****
Read Map File Start! (2023-09-01 11:59:32)
Chip Initialization Start! (2023-09-01 11:59:32)
Chip Initialization Done! (2023-09-01 12:00:48)
Program Step 1 Start! (2023-09-01 12:00:48)
Program Step 2 Start! (2023-09-01 12:01:06)
>>>>20%>>>>40%>>>>60%>>>>80%>>>>100%>
Program Step 3 Start! (2023-09-01 12:01:48)

Refine: 2/2 ...

Program Step 1 Start! (2023-09-01 12:01:53)
Program Step 2 Start! (2023-09-01 12:02:11)
>>>>20%>>>>40%>>>>60%>>>>80%>>>>100%>
Program Step 3 Start! (2023-09-01 12:02:54)

Tag: NO TAG
Program Finished! (2023-09-01 12:03:04)
Total program time: 212 sec.

```

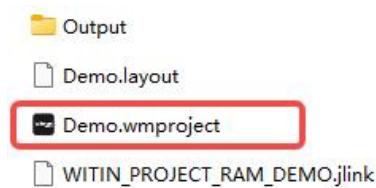

模块五：算法模型在芯片运行推理

步骤一：

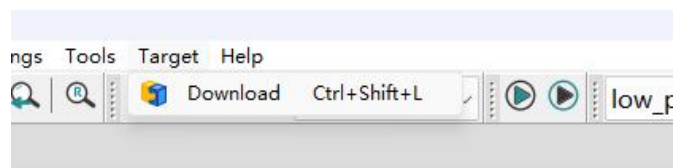
从官网下载知存 IDE Witmem Studio。

步骤二：

生成的 mapper/output/register.c 放在 project/Model，使用 Witmem Studio 打开 project/Project/SES-RISCV/Demo.wmproject。



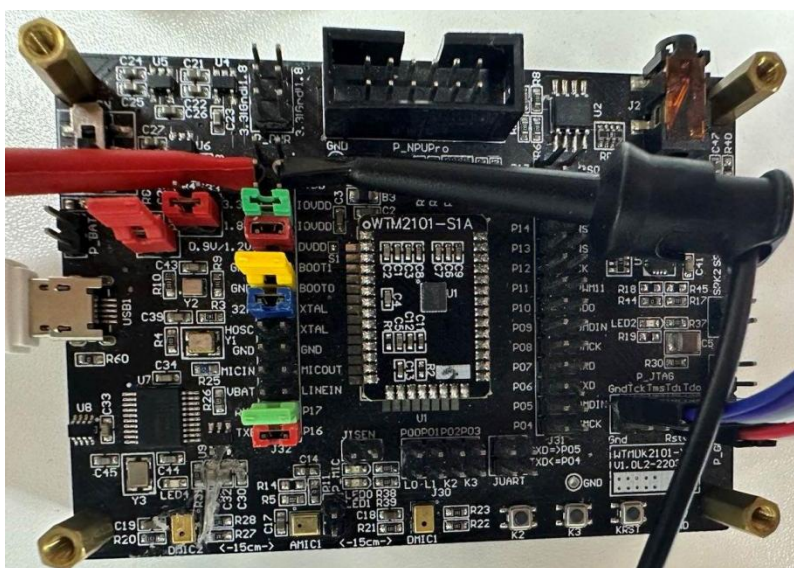
步骤三：Target->Download 下载工程：



注：本 demo 所用为 x3 开发板，所用串口 GPIO 为 16,17。若是其他开发板，请根据情况修改串口。

步骤四：

打开 tools 中的串口工具，设置波特率 115200，查看准确率输出



至此，我们完成了语音识别从训练到部署的全流程，本教程结束。