

# 离线语音模块使用手册

wireless-tag

版本 1.0

2020 年 8 月 25 日

深圳启明云端科技有限公司

## 免责声明和版权公告

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

## 注意

由于产品版本升级或其他原因，本手册内容有可能变更。深圳市启明云端科技有限公司 保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用 指导，深圳市启明云端科技有限公司尽全力在本手册中提供准确的信息，但是深圳市启明云端科技有限公司并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。

历史版本				
版本	作者	审核	时间	描述
1.0.0			2020-08-25	第一次创建

## 目录

1. 概述.....	1
2. 产品说明.....	1
2.2 主要特征.....	1
3. 获取源码.....	3
3.1 前期准备.....	3
3.2 创建产品.....	5
3.3 产品版本开发.....	8
3.3.1 创建产品版本.....	8
3.3.2 配置终端能力.....	9
4. 烧录固件.....	21
5. 搭建开发环境.....	32
6. 软件架构.....	35
7. 二次开发.....	41

目前只是体验账户，方便大家提前熟悉了解离线语音相关开发，尊重他人的劳动成果，可以体验请不要乱删项目

## 1. 概述

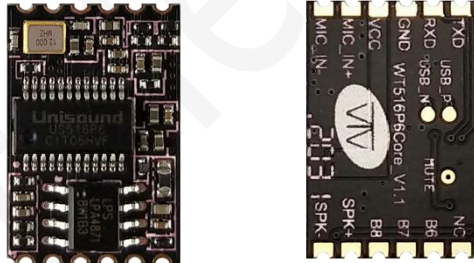
WT516P6Core 模组是由启明云端科技开发的、低功耗高性价比的嵌入式音频模块。该模块核心处理器 US516P6，模组集成了功率放大器、麦克风模块。为客户提供超低成本的离线语音识别方案，可广泛且快速应用于智能家居，各类智能小家电，86 盒，玩具，灯具等需要语音操控的产品。

模组供电电压 5V，最大供应电流建议不小于 1A。外接 12MHz 晶振。应用场景：智能穿戴设备、蓝牙室内导航、运动健身、智能工业、信息安全、零售支付、数据传输、楼宇自动化、安防、智能家居、等物联网应用的需求。

## 2. 产品说明

### 2.1 硬件说明

WT516P6Core 模组是由启明云端科技开发的、低功耗高性价比的嵌入式音频模块。该模块核心处理器 US516P6，模组集成了功率放大器、麦克风模块，为客户提供超低成本的离线语音识别方案。



### 2.2 主要特征

配置	规格
蜂鸟 M 芯片	高速 SRAM
	FLASH 2MB
mic 数量	1
UART 通信接口	1
LED 接口	有
音频接口	I2S, ADC, DAC

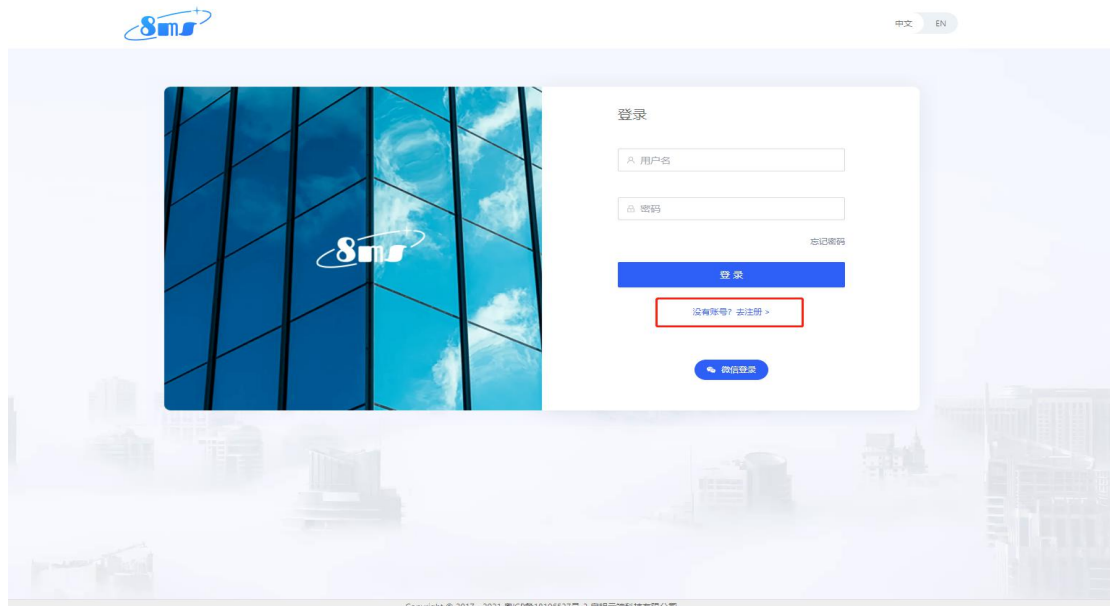
### 2.3 软件功能

序号	功能	功能描述
1	单麦拾音	<ul style="list-style-type: none"> <li>单麦克风方案</li> <li>支持家居场景 5m 远讲</li> </ul>
2	语音唤醒	<ul style="list-style-type: none"> <li>高性能唤醒引擎</li> <li>低功耗</li> <li>支持带口音的普通话</li> <li>低误唤醒率 (&lt; 1 false in 48 hours)</li> </ul>
3	离线识别	<ul style="list-style-type: none"> <li>支持本地 100 条控制指令识别</li> </ul>
4	多轮对话	一次唤醒连续对话，语音操作更加便捷自然。
5	多种发音人音色	<ul style="list-style-type: none"> <li>提供标准女声、甜美女声、可爱女声、台湾女声、标准男声、女童声、男童声七种音色可选</li> </ul>
6	UART 主板对接	云知声提供标准 UART 协议，也支持对接用户自有协议
7	智能设备平台	云知声设备平台提供： <b>唤醒词自定义，命令词自定义，回复播报语自定义，发音人音色选择</b> 等产品自定义配置

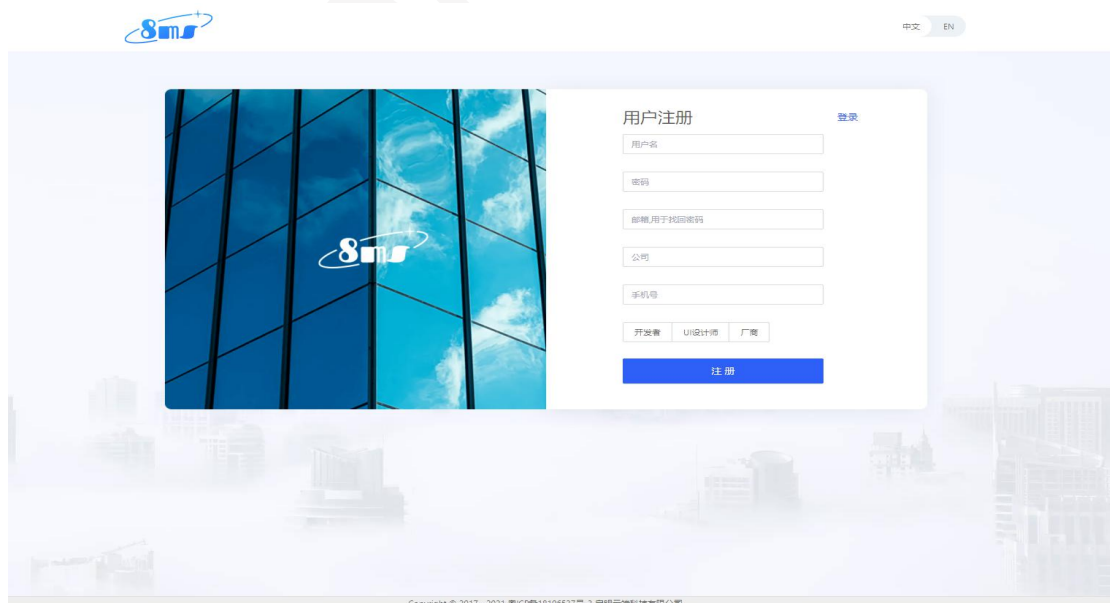
### 3. 获取源码

#### 3.1 前期准备

(1)：账号创建 (注意：如果使用的是新版本的谷歌浏览器，Google chrome 需要开启跨站 cookie 地址栏输入 chrome://flags/搜索：samesite，然后把里面的配置项目 samesite by default cookie 关了)  
进入 **8ms 平台** <http://8ms.xyz>，点击“没有账号？去注册”



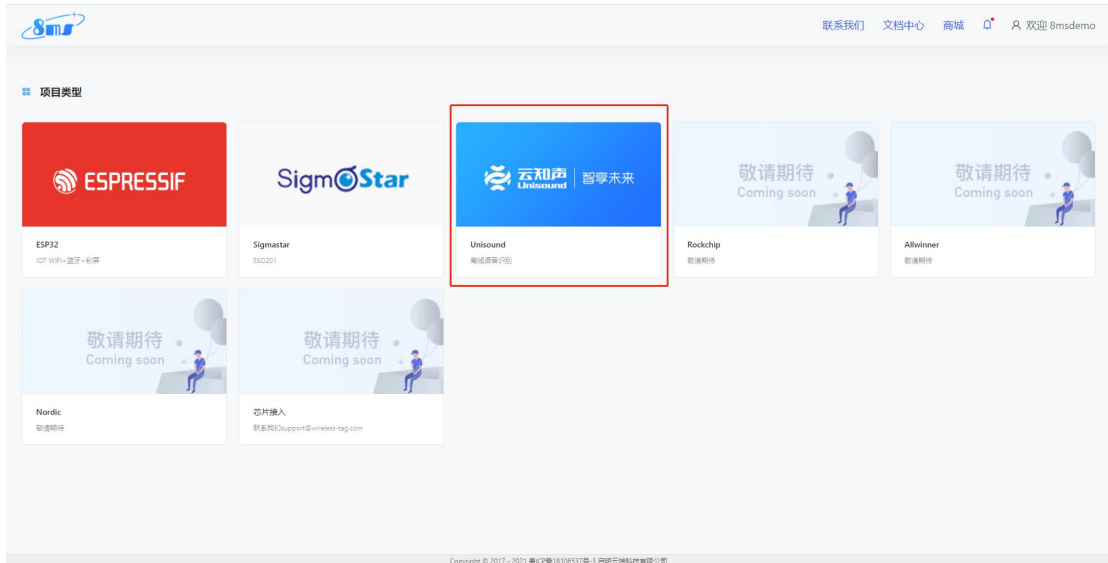
点击右下角后，弹出如下页面



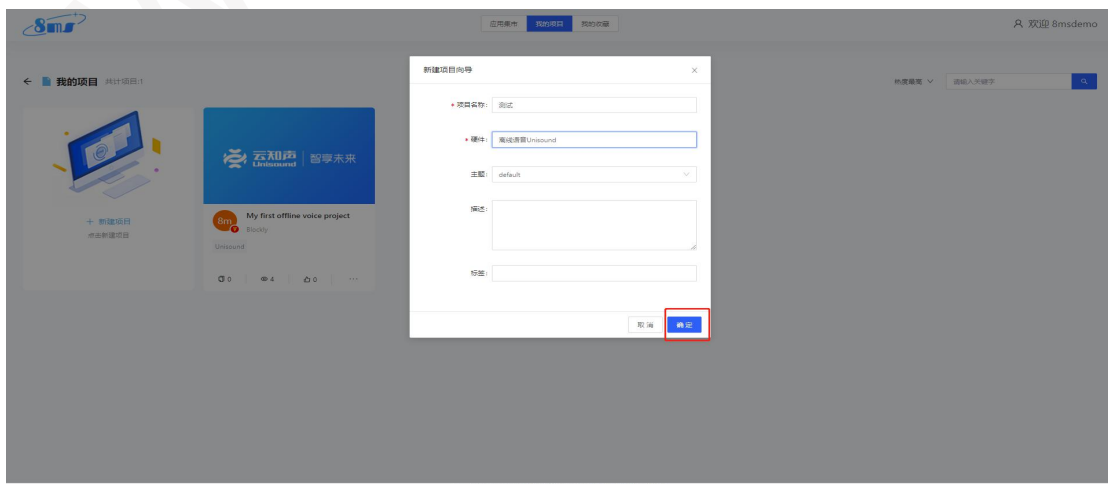
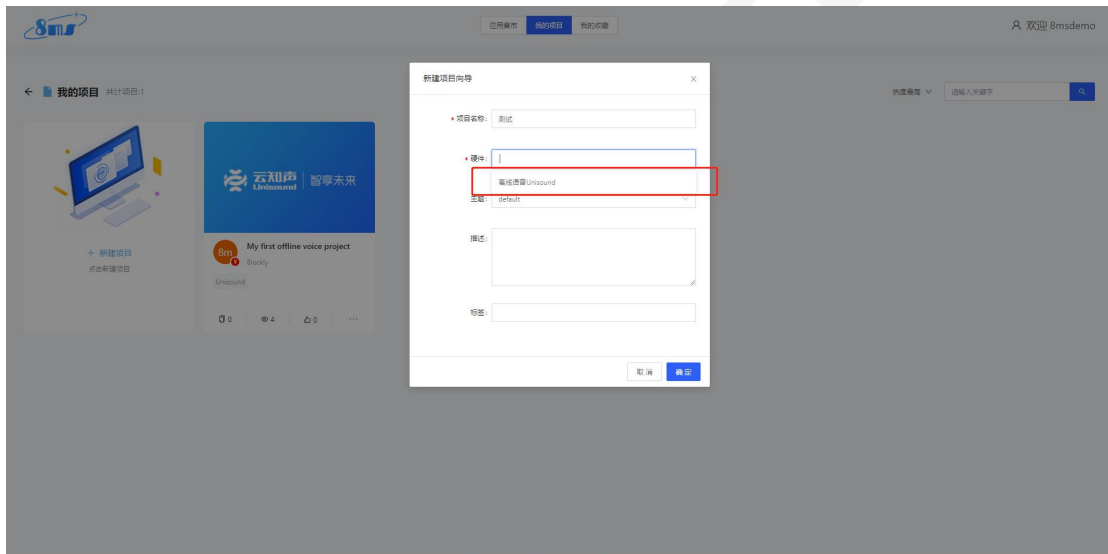
如无账号，请点击**用户注册**

如已有 8ms 平台账号请忽略本步骤

(2)：账号注册后，再登录，点击 **Unisound** 进入离线语音页面，点击**我的项目**

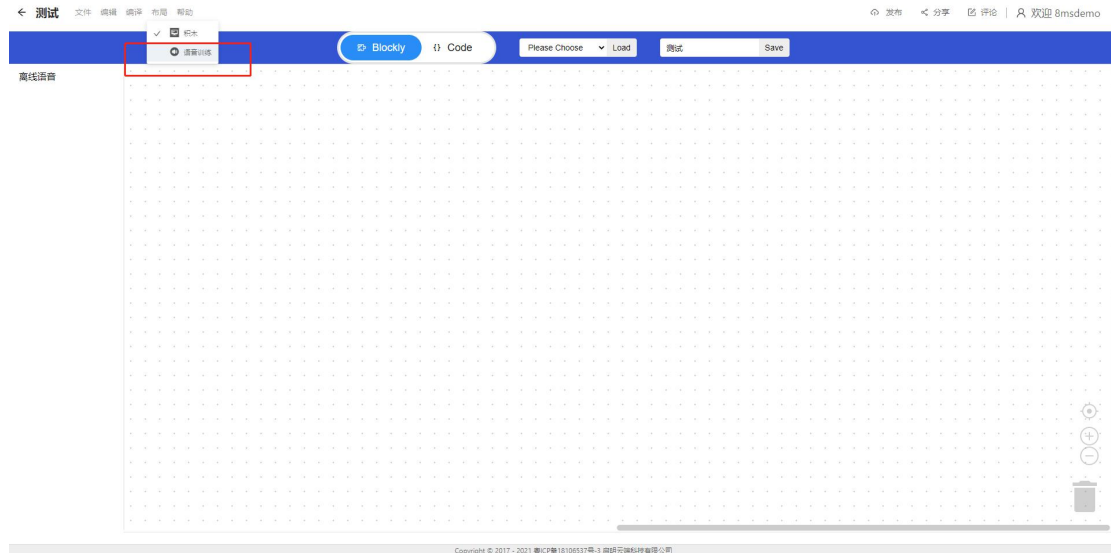


(3)：点击新建项目，弹出如下窗口-输入**项目名称**-选择**硬件-离线语音 Unisound**

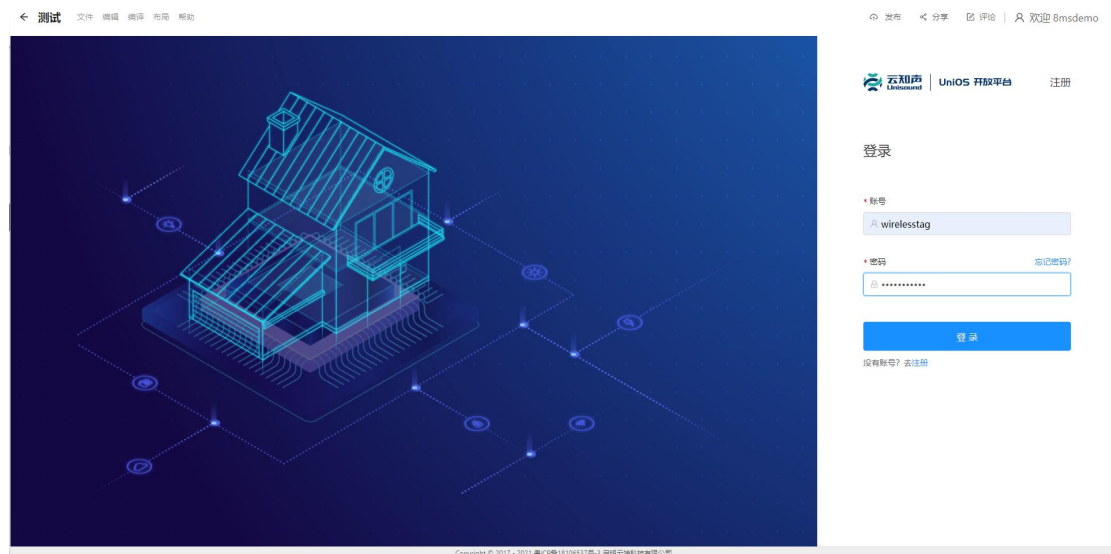


(4)：新项目创建完成后，页面自动跳转到以下页面

点击左上角-布局-选择语音训练



(5)：页面自动跳转至云知声登录页面，用户登录体验账号：wirelesstag（密码和账号同号）



### 3.2 创建产品

(1)：用户登录后，进入产品控制台后，点击 **创建产品**



(2) :云知声智能设备平台提供不同场景，用以定制不同语音产品形态，此处选择智能家居（纯离线）



(3) :选择系统与芯片平台，选择 蜂鸟 M 芯片 该页面点击芯片旁的? 按钮，可以获取蜂鸟芯片产品手册。



(4) :填写产品信息，包括名称与描述



(5) :确认所选项无问题, 点击 **保存且下一步**, 即初步创建产品成功



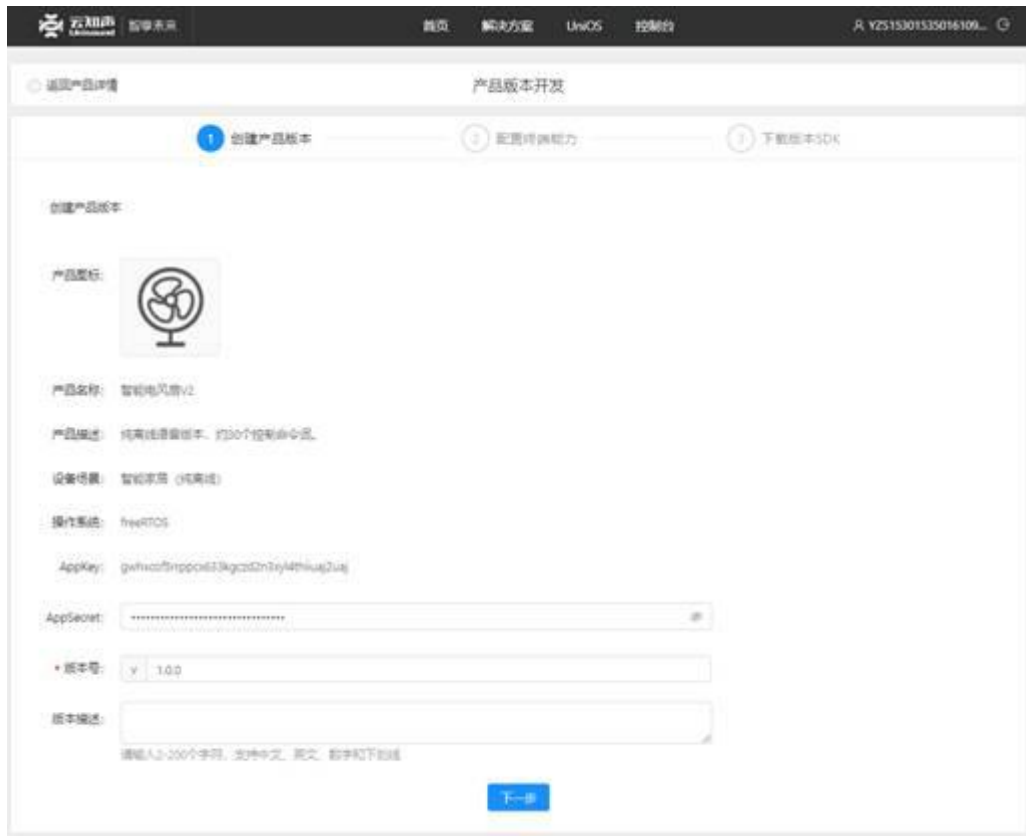
### 3.3 产品版本开发

#### 3.3.1 创建产品版本

1) : 创建产品版本, 填写版本号及版本信息, 点击下一步

产品创建成功后, 系统会自动为该产品分配 Appkey 与 AppSecret, 此为该产品唯一识别 ID 和密匙

一个产品可维护多个版本, 可通过版本号与版本描述来区分



### 3.3.2 配置终端能力

进入终端能力配置，首先是语音产品基础能力表，介绍了一个语音类型产品应有的基本能力。



### 1. 自定义产品唤醒词，以及唤醒回复语

支持 3-6 个字的唤醒词自定义配置；

为保证语音产品有最佳唤醒体验，平台针对唤醒词的发音提供打分功能，并会提供具体指导信息；

涉及黄赌毒、政治敏感、低俗辱骂等不合规内容的唤醒词不可使用；

完整唤醒词定义规则可点击界面上《唤醒词自定义规则》

唤醒回复最多支持 5 条，设置后将随机播报



### 2. 按照规则填写命令词与应答语

如下图所示，左部分为语法格式说明与规则介绍，右部分则为输入框，可以在页面内直接输入命令词与应答语的定义，如不满足语法格式要求平台则会进行实时提示。



### 语法格式说明与规则介绍

为方便产品构建，平台定义了一套语法格式，语法格式组成为：

```
action=命令词 1|命令词 2...@回复语
```

举例：**TempSet15=设置十五度|十五度@已设为十五度**

**action** 为一个控制指令的唯一标识，用户对设备说出“设置十五度”“十五度”并被语义理解时，如已对接设备，语义理解模块会将 TempSet15 传给设备。

**命令词** 为想要定义的语音话术，用户必须按照定义的话术说出才有效。如用户可以使用“设置十五度”“十五度”来实现同一个设置温度 15 度的控制。

**回复语** 为针对该条控制指令的设备回复播报。

action、命令词、回复语均由用户定义。等号前的 action 为唯一值；等号后的命令词可输入多个，以 | 符号分割；@后为回复语，每个控制指令可以定义对应的回复。

点击语法格式后的 **问号**，可看到语法格式要求：

- action 由英文、下划线“\_”和数字组成，必须英文开头，不区分大小写，15 个字符内
- 命令词最多支持 100 条，每条限 2 - 10 个字符，仅支持中文
- 一个 action 仅支持一条回复语，回复语总字数不得超过 500 个字符，支持中文、数字、逗号、句号、问号

点击语法句式后的 **问号**，可以看到针对回复播报特殊读音的标签使用说明：

- `<py>`：需要对单个汉字的发音进行纠正的场合。

注：拼音声调范围为 1 - 5，1 - 4 对应一声到四声，5 对应轻声。

例：已调`<py>tiao2</py>`至中`<py>zhong1</py>`风档

播报为：已调(tiao2)至中(zhong1)风档

- `<value>`：需要将数字按照数值读法播报

例：已设为`<value>15</value>`度

播报为：已设为十五度

- `<code>`：需要将数字按照数字串逐位播报

例：已设为`<code>15</code>`度

播报为：已设为一五度

**开发者注意：**

1. 产品默认支持音量大小调节，音量调节控制默认句式如下：

**volumeUpUni=增大音量**

**volumeDownUni=减小音量**

如需自定义音量与主动退出的控制话术与应答播报，可在语法文件内添加以上控制句式，只需保持=号前的 action 不变，修改=后的内容即可。

举例如下：

volumeUpUni=大声一点|调大声点@已为您调高音量

exitUni=拜拜|再见@好的，那我先退下了

2. 产品内置音量等级三档控制逻辑，默认不开启。如需使用该逻辑，可在定义离线命令词与播报语处，添加词条后生效：

最大音量 **volumeMaxUni**

中等音量 **volumeMidUni**

最小音量 **volumeMinUni**

如希望添加最大音量调节，可参考下方举例添加句式：

volumeMaxUni=最大音量|音量调到最大@已调至最大音量

3. 用户可以在离线命令词自定义部分，指定特定的 action，定义非语音交互的应答语播报，用于非正常语音交互的情景。

开发者需从指定文件夹获取此部分定制的语音音频文件，并根据产品定义修改源码。可参见文内 5.3 播报逻辑开发。

格式如下：

Special\_Replies@回复语 1|回复语 2|回复语 3

如警报播报：以智能烧水壶为例，产品内传感器检测到缺水状态，自动语音警报；

如极限播报：空调指令“升高温度”，正常播报为“温度已升高”；当温度到上限时，播报“温度已调至最高”。

举例如下：

Special\_Replies@内丹干烧|水壶缺水|没水啦|主人请加水

4. 用户可以在离线命令词自定义部分，指定特定的竞争词（可理解为无效命令词），防止一定概率的误识别。

格式如下：

Special\_Gabages=命令词 1|命令词 2|命令词 3|...

以空调产品为例，并没有高于 32 度的温控指令，用户说出“三十六度”时，不希望空调识别到“十六度”；以空调产品为例，没有自动风的指令，当用户说“自动风”时，不希望空调识别到“自然风”（如果阈值设置较低）。

举例如下：

举例：Special\_Gabages=三十六度|四十六度|五十六度|六十六度

### 3) 免唤醒命令词

无需唤醒，说出命令词即可控制设备，此处的免唤醒命令词，需先在离线命令词定制模块设定后，方可进行选择配置。

免唤醒命令词生效后，设备会处于待唤醒状态。

免唤醒命令词与唤醒词可同时存在，也可单一存在，两者数量之和不可超过 10 个。如针对某些特定语音产品，可只使用免唤醒命令词来替代唤醒词。



### (4) 选择语音识别模型

蜂鸟 M 芯片方案支持对声学模型进行选择，默认为云知声针对蜂鸟 M 芯片定制的经过算法优化的通用声学模型，即“系统默认”。

如当前系统默认的声学模型性能无法满足用户需求时（注：此处用户需求，指经过语音性能量化测试的明确指标），可以使用云知声模型训练平台，采集特定离线命令词的语音训练数据并进行模型定制。可点击选择框右侧文字跳转至模型训练平台。



#### (5) 针对产品定位，选择适合产品形态的发音人

- 活泼可爱的萱萱和 KiYo，童真童趣的糖糖和天天，温柔可亲的玲玲，庄重严肃的小雯玉与小峰，七种音色可供选择；
- 可调节音量与语速；
- 提供在线试听，方便用户进行决策；



## 6) 其它配置

### 1) 开机音配置



产品默认为开机铃声, 可在页面点击试听。

如需进行开机欢迎语的自定义, 也可以选择自定义内容, 产品则会在开机后自动进行播报, 欢迎语播报自定义限中文, 可输入 30 个字符; 也可选择不使用开机音。



## 2) 识别状态退出相关配置

语音产品为保证用户体验，唤醒后一段时间内无语音输入产品会主动退出识别状态，如需输入命令词，则需再次唤醒。

此段等待用户输入语音命令的有效时间，即为“唤醒超时退出时间”。

- 为确保产品有较好的用户体验，支持 5s – 60s 的超时退出时间设置，默认为 10 秒
- 超时退出回复最多可设置 2 条

当用户想要结束对话时，也可以主动要求退出识别状态。

- 默认退出命令为“退下”“再见”，可根据产品定义修改，最多可设置 3 条
- 主动退出回复最多可设置 2 条

The screenshot displays a configuration interface with two main sections: '超时退出' (Timeout Exit) and '主动退出' (Active Exit). Each section includes a '超时时间' (Timeout Time) field, a '退出回复' (Exit Reply) list, and a '+添加一条' (Add one) button. The '超时退出' section shows a timeout of 10s and a reply '有需要再叫我'. The '主动退出' section shows default exit commands '退下' and '再见', and a reply '有需要再叫我'.

如配置完成，则可点击下一步，进入版本发布页面。

### 3.3.3 下载版本

(1) 产品配置开发完成，进入发布页面后，可点击 **立即发布**



(2) 产品发布需要一定时间，可以有两种方法获取到发布成功的产品

- 发布界面下，版本发布成功后，会有下载提示



- 也可以直接返回产品详情，或关闭页面一段时间再进入智能设备平台的产品详情，页面中的版本列表处点击下载



### 3.3.4 产品版本维护

产品详情页面可以查看已创建的产品，可以查看每个版本的配置信息，也可以对已发布成功的版本进行下载或者删除操作



### 3.3.5 新版本迭代

同一个产品如有新需求的修改或迭代，可在产品详情页面点击 **+开发新版本**，后续的操作步骤同创建产品版本时的流程一致。

提供选择任一历史版本进行迭代开发的选项，默认为基于最新修改的版本，也可根据需求进行其它版本的选择。

#### 4. 烧录固件

用户在页面开发语音功能，需要等几个小时后再在里面下载固件

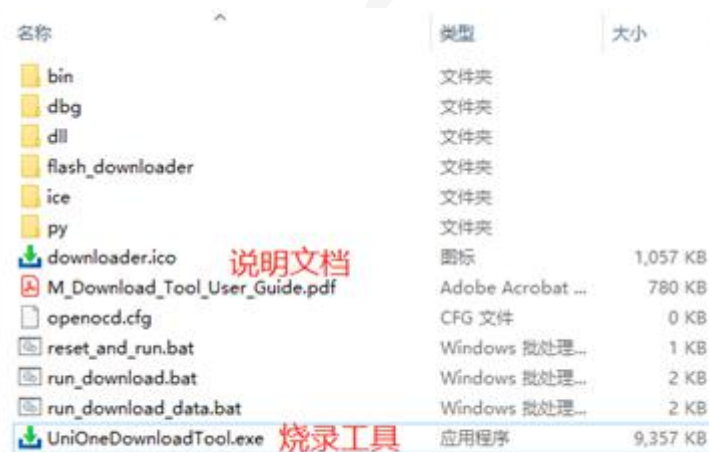
##### 4.1 文件介绍

- 产品的软件烧录包下载成功后，解压文件，可以看到如下方文件目录



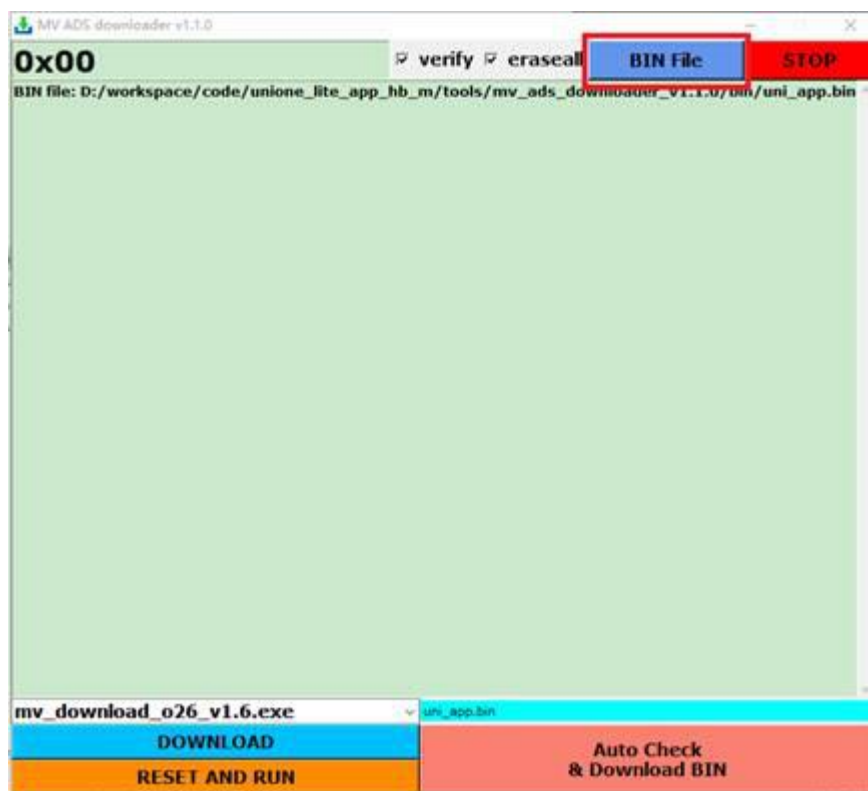
名称	类型	大小
image_demo	文件夹	
unione_lite_app_hb_m	文件夹	
build.properties	PROPERTIES 文件	62 KB

- 如想直接使用，体验语音功能，请打开“image\_demo/Hummingbird-M-Production-Tool”文件夹，双击“UniOneDownloadTool”，运行烧录工具



名称	类型	大小
bin	文件夹	
dbg	文件夹	
dll	文件夹	
flash_downloader	文件夹	
ice	文件夹	
py	文件夹	
downloader.ico	图标	1,057 KB
M_Download_Tool_User_Guide.pdf	Adobe Acrobat ...	780 KB
openocd.cfg	CFG 文件	0 KB
reset_and_run.bat	Windows 批处理...	1 KB
run_download.bat	Windows 批处理...	2 KB
run_download_data.bat	Windows 批处理...	2 KB
UniOneDownloadTool.exe	应用程序	9,357 KB

- 烧录工具配置界面如下图所示：



## 4.2 烧录指南

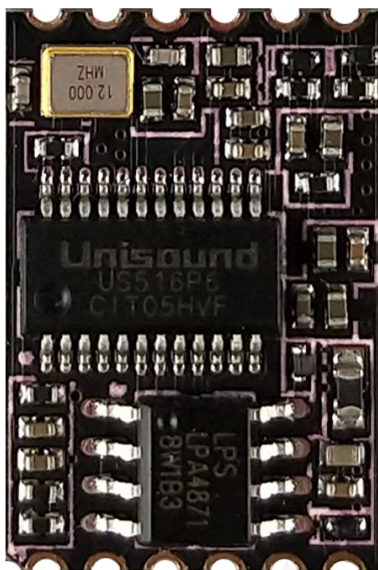
### 4.2.1 烧录前介绍

开始烧录之前，需要准备：

- Micro USB 连接线，用于供电
- 烧录器，用于烧录
- 蜂鸟 M 开发板
- 固件包



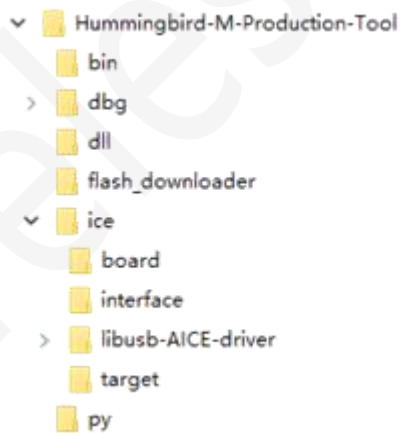
烧录器



WT516P6Core 开发板示例图

开始烧录之前，需要准备：

打开烧录包，Hummingbird-M-Production-Tool 内，可看到以下目录结构：



和用户相关的目录有以下几个：

- \bin : 用户存放 bin/mva 文件的目录
- \flash\_downloader : flash downloader 存放目录，需要提取相应芯片 SDK 中提供的 downloader.exe
- \ice\libusb-AICE-driver : 仿真器驱动

名称	类型	大小
bin	文件夹	
dbg	文件夹	
dll	文件夹	
flash_downloader	文件夹	
ice	文件夹	
py	文件夹	
build.properties	PROPERTIES 文件	10 KB
demo_ap82.png	PNG 文件	16 KB
downloader.ico	图标	1,057 KB
openocd.cfg	CFG 文件	0 KB
readme.pdf	Adobe Acrobat ...	874 KB
reset_and_run.bat	Windows 批处理...	1 KB
run_download.bat	Windows 批处理...	2 KB
run_download_data.bat	Windows 批处理...	2 KB
UniOneDownloadTool.exe	应用程序	9,357 KB

根目录有以下用户相关的文件:

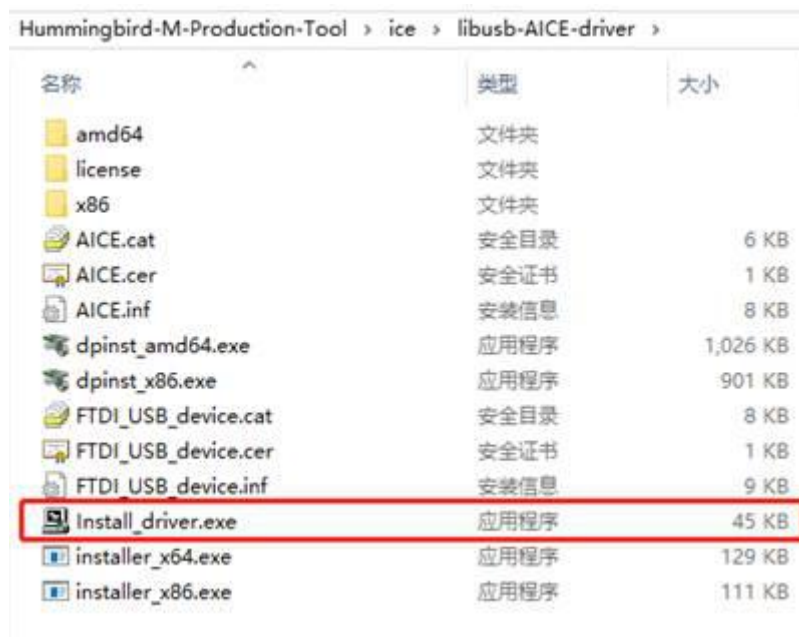
- UniOneDownloadTool\_x64.exe : 烧录工具启动 exe
- run\_download.bat : 批量执行脚本例程

#### 4.2.2 安装驱动 (如已安装, 可跳过)

插入烧录器, 如在设备管理器看到未知设备, 则需要安装驱动。



- 1、 准备固件：发布的固件包，如 Hummingbird-M-Production-Tool
- 2、 打开 Hummingbird-M-Production-Tool\ice\libusb-AICE-driver， 双击 Install\_driver.exe， 点击同意安装

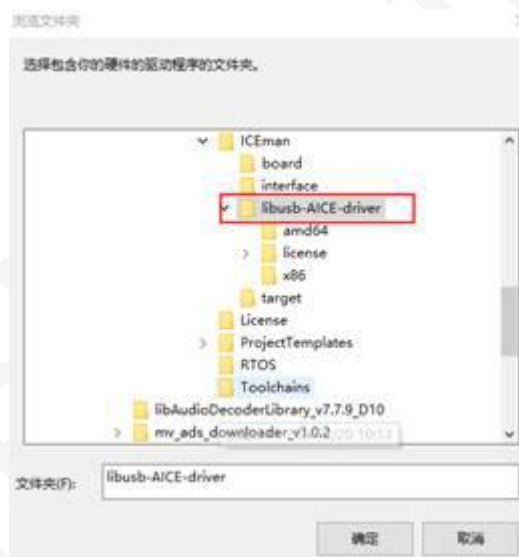


- 3、 处于设备管理器页面， 右击 AICE（未知设备）， 更新驱动程序



#### 4、 选择浏览计算机查找驱动

驱动路径： Hummingbird-M-Production-Tool\ice\libusb-AICE-driver



- 5、 点击确定后，回到更新驱动程序页面，点击下一步，等待驱动安装成功。

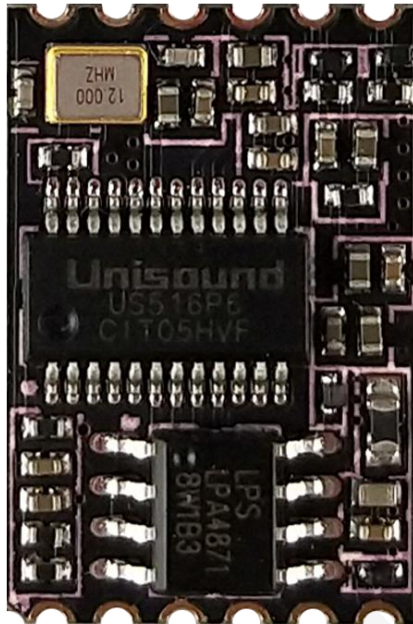


### 4.2.3 进入烧录模式

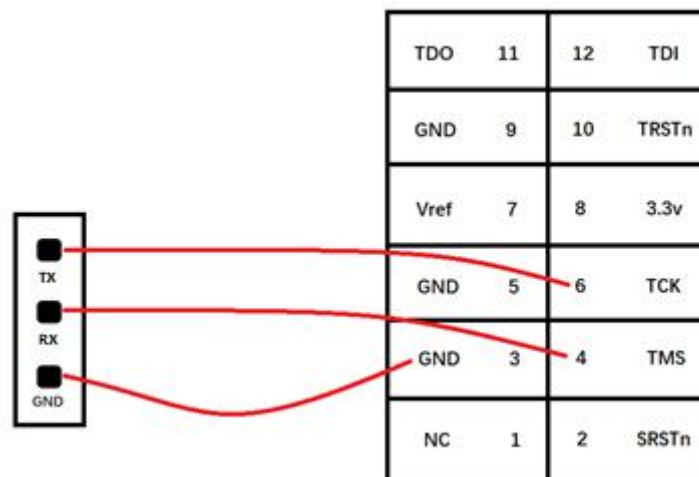


#### 4.3.1 接入烧录器

在开发板上找到“Debug”接口，该接口有三根信号线，按开发板上丝印分别为 GND、RX、TX。



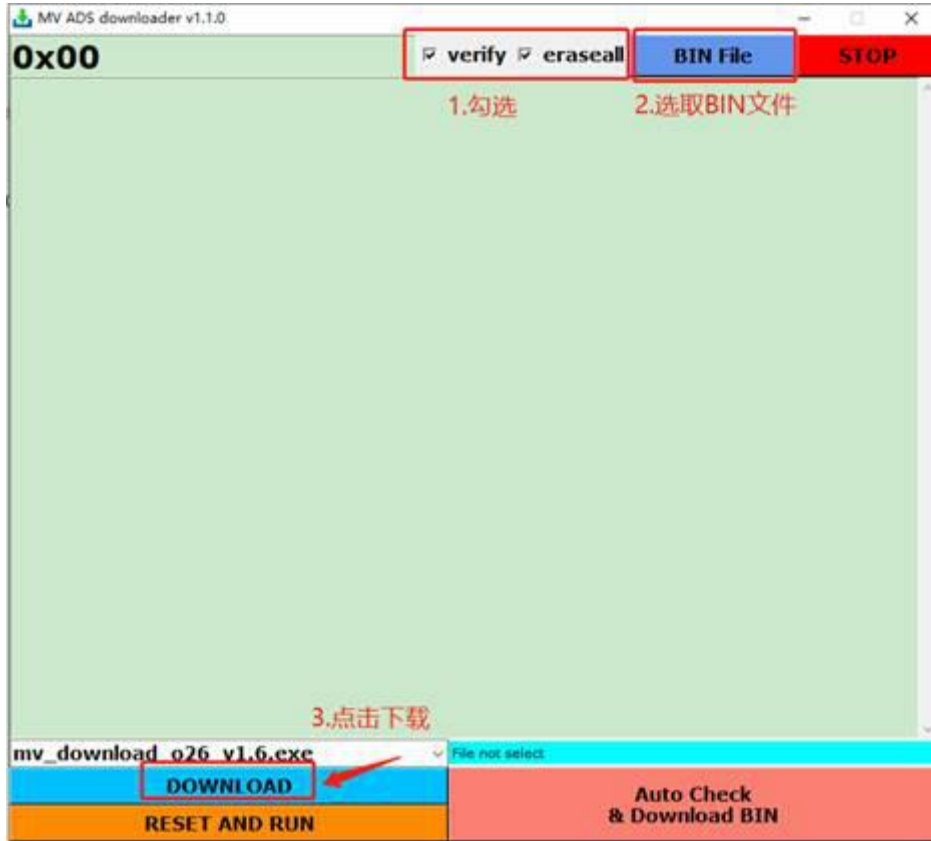
三个接口分别对应烧录器上 GND (3 号脚)、TMS (4 号脚)、TCK(6 号脚), 接线示意图如下所示:



#### 4.3.2 开发板供电, 开始烧录

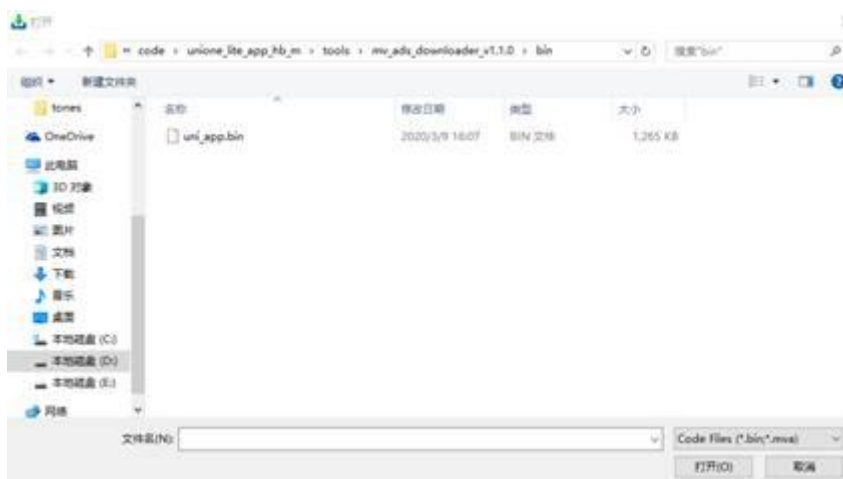
使用 USB 连接线给开发板上电，即接入电源。可直接接电脑，或者任意 5V 电源适配器（如手机充电器）。

点击烧录工具界面“BIN File”按钮，选择待烧录 bin 文件：

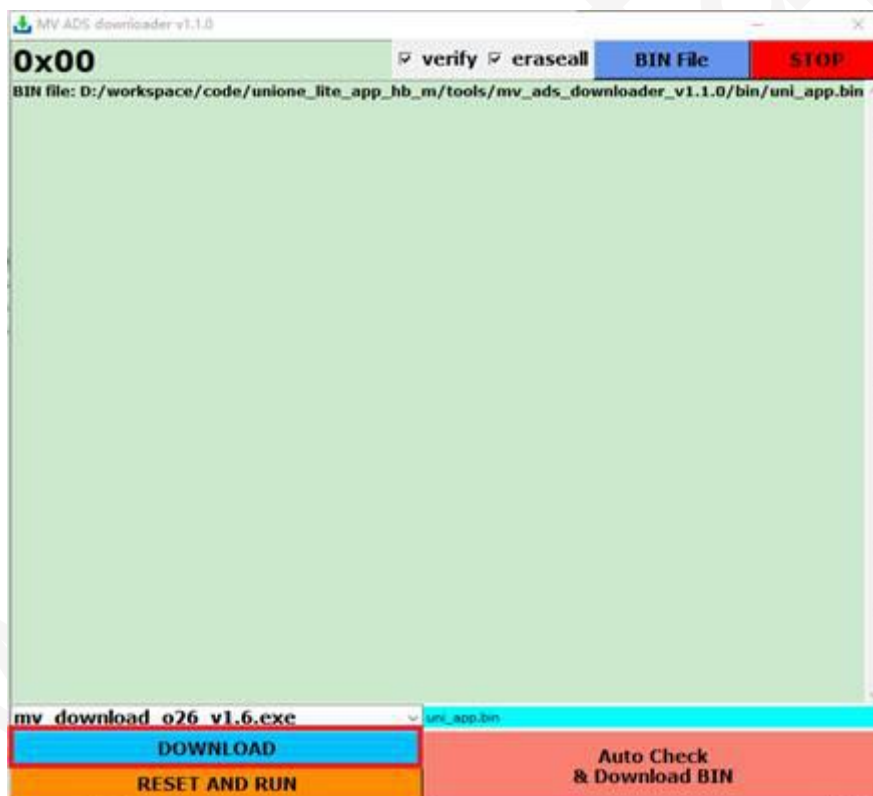


默认选择路径为“image\_demo/Hummingbird-M-Production-Tool/bin”，也可通过文件浏览器选择其他路径：

注：目前发布包中同时包含 uni\_app\_debug.bin 和 uni\_app\_release.bin 两个版本，其中 release 为正式版本，debug 为带 log 调试版本，与 release 版本相比响应速度较慢，仅用于调试使用。



选择烧录 bin 文件后，点击烧录工具界面“DOWNLOAD”按钮，开始烧录：



烧录成功则烧录工具界面打印烧录进度，直到打印“Verify success”信息：

```
82% Verify Done
82% Verify Done
83% Verify Done
84% Verify Done
84% Verify Done
85% Verify Done
86% Verify Done
86% Verify Done
87% Verify Done
87% Verify Done
88% Verify Done
89% Verify Done
89% Verify Done
90% Verify Done
91% Verify Done
91% Verify Done
92% Verify Done
92% Verify Done
93% Verify Done
94% Verify Done
94% Verify Done
95% Verify Done
96% Verify Done
96% Verify Done
97% Verify Done
98% Verify Done
98% Verify Done
99% Verify Done
99% Verify Done
100% Verify Done
Verify success.

Flash burn success.
Spend time: 55.86 seconds
No BIN file selected.
```

如未显示则表示未识别到 COM 口，请检查：

- ① 串口线是否接好
- ② 串口线是否反接
- ③ 是否安装 UART 对应驱动。

而烧录失败界面将输出错误信息，如下图：

```
BIN file: D:/workspace/code/test_hb_m_ci/uni_hb_m_solution/image_demo/mv_ads_downloader_v1.1
/bin/uni_app.bin
Open On-Chip Debugger 0.10.0+dev-g161b439 (2018-06-04-10:33)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
Andes ICEman v4.2.2 (OpenOCD) BUILD_ID: 2018060410
Burner listens on 9900
Tclnet port: 9901
TCL port: 6666
<- Can not open usb (vid=0x1cfc, pid=0x0) -->
<- ICEman exit... -->
<- Can not open usb (vid=0x1cfc, pid=0x0) -->
<- ICEman exit... -->
assertion "target" failed: file "/home/sqa/build-ast312/build-system-3/source-packages/openocd-0.10
/src/jtag/alice/alice_apis.c", line 2051, function: nds_freerun_all_targets
ICEman connect failed, return!
```

如烧录失败，请检查：

- ① 开发板是否供电
- ② 开发板 SW Debug 接口是否接好
- ③ SW 线是否按照要求反接，确认后请回到步骤 1 重新短接上电，重新进入烧录模式
- ④ 其他错误，请按照烧录界面提示信息排查问题

## 5. 搭建开发环境

### 5.1 环境准备

(1) 下载编译工具

推荐使用 Ubuntu 16.04 / centos7 以上版本作为开发环境。

交叉编译工具下载地址：链接：[https://pan.baidu.com/s/1EHa10q7xkoySt5MA8\\_abew](https://pan.baidu.com/s/1EHa10q7xkoySt5MA8_abew)

提取码：cdbx

下载交叉编译工具并解压到开发环境“/opt”路径中，或其他路径，然后修改 makefile 文件中配置选项：

```
brainbrain-VirtualBox:/mnt/test_hb_m_ci/uni_hb_m_solution$ ls -l /opt/Andestech/85Py422/toolchains/nds32le-elf-mcu1b-v3a
总用量 28
drwxr-xr-x 2 root root 4096 2月 11 14:31 bin
drwxr-xr-x 3 root root 4096 2月 11 14:31 conf
drwxr-xr-x 3 root root 4096 2月 11 14:31 include
drwxr-xr-x 3 root root 4096 2月 11 14:31 lib
drwxr-xr-x 3 root root 4096 2月 11 14:31 libexec
drwxr-xr-x 6 root root 4096 2月 11 14:31 nds32le-elf
drwxr-xr-x 6 root root 4096 2月 11 14:31 share
```

(2) 已下载的版本里除了可直接烧录使用的 image\_demo 外，文件夹内包含产品方案源码，如下图所示：

```
brain@brain-VirtualBox:/mnt/test_hb_m_ci/uni_hb_m_solution$ ls -l
总用量 129
drwxrwxrwx 1 root root 0 3月 6 18:47 build
-rwxrwxrwx 1 root root 65322 3月 6 18:40 build.properties
-rwxrwxrwx 1 root root 459 3月 6 18:40 build.sh
-rwxrwxrwx 1 root root 867 3月 6 18:40 ci.yml
drwxrwxrwx 1 root root 0 3月 6 18:47 driver
drwxrwxrwx 1 root root 0 3月 6 18:47 image_demo
drwxrwxrwx 1 root root 0 3月 6 18:47 include
drwxrwxrwx 1 root root 0 3月 6 18:47 lib
drwxrwxrwx 1 root root 0 3月 6 18:47 middleware
-rwxrwxrwx 1 root root 18285 3月 6 18:40 nds32-ae210p.ld
-rwxrwxrwx 1 root root 782 3月 6 18:40 nds32-ae210p.sag
-rwxrwxrwx 1 root root 29920 3月 6 18:40 readme.txt
drwxrwxrwx 1 root root 0 3月 6 18:47 src
drwxrwxrwx 1 root root 0 3月 6 18:47 startup
drwxrwxrwx 1 root root 0 3月 6 18:47 tools
-rwxrwxrwx 1 root root 1651 3月 6 18:40 uni_ci.yml
```

(3) 修改 build/makefile 中交叉工具链的路径为实际解压到的工具路径;

```
#####
# Automatically-generated file. Do not edit!
#####
CODE_ROOT=../
ifndef CROSS_COMPILE
CROSS_COMPILE=/opt/Andestech/BSPv422/toolchains/nds32le-elf-mcu/lib-v3s/bin/nds32le-elf
endif
ifndef SECONDARY_OUTPUT_PATH
SECONDARY_OUTPUT_PATH=$(CODE_ROOT)/output
endif
$(shell mkdir -p $(SECONDARY_OUTPUT_PATH))
-include ../makefile.init
RM := rm -rf

# All of the sources participating in the build are defined here
-include sources.mk
-include subdir.mk
-include driver/driver_api/src/subdir.mk
-include middleware/audio/src/subdir.mk
-include middleware/fatfs/src/subdir.mk
-include middleware/mv_utils/src/subdir.mk
-include middleware/rtos/freertos/src/subdir.mk
-include middleware/rtos/rtos_api/src/subdir.mk
-include src/app/src/subdir.mk
-include src/app/src/sessions/subdir.mk
-include src/hal/src/subdir.mk
-include src/sdk/audio/audio_player/src/subdir.mk
-include src/sdk/idle_detect/src/subdir.mk
-include src/sdk/player/src/pcm/src/subdir.mk
-include src/sdk/player/src/subdir.mk
-include src/sdk/uart/src/subdir.mk
-include src/sdk/vui/src/subdir.mk
-include src/utlis/aprt/src/subdir.mk
-include src/utlis/auto_string/src/subdir.mk
-include src/utlis/bitmap/src/subdir.mk
-include src/utlis/black_board/src/subdir.mk
-include src/utlis/c350k/src/subdir.mk
-include src/utlis/config/src/subdir.mk
-include src/utlis/crc16/src/subdir.mk
-include src/utlis/data_buf/src/subdir.mk
-include src/utlis/event/src/subdir.mk
-include src/utlis/event_list/src/subdir.mk
-include src/utlis/event_route/src/subdir.mk
-include src/utlis/float2string/src/subdir.mk
-include src/utlis/fsm/src/subdir.mk
-include src/utlis/hash/src/subdir.mk
-include src/utlis/interruptible_sleep/src/subdir.mk
build/makefile: 152 lines, 5106 characters
```

(4) 安装 python 环境: sudo apt-get install python

(5) 安装 lame 音频处理工具: sudo apt-get install lame

(6) 安装 32 位兼容库 (如果系统未安装过): sudo apt-get install lib32stdc++6 lib32z1

lib32ncurses5 lib32bz2-1.0 (仅适用于 Ubuntu 16.04 版本, 其他版本请根据系统说明解决兼容性问题)

## 5.2 编译

### (1) 执行编译

执行./build.sh, 编译成功会看到如下信息:

```
Finished building target: uni_app.adx
Invoking: nm (symbol listing)
/opt/Andestech/BSPv422/toolchains/nds32le-elf-mculib-v3s/bin/nds32le-elf-nm -n -l -C "uni_app.adx" > ../output/symbol.txt
Finished building: ../output/symbol.txt

Invoking: readelf (ELF info listing)
/opt/Andestech/BSPv422/toolchains/nds32le-elf-mculib-v3s/bin/nds32le-elf-readelf -a "uni_app.adx" > ../output/readelf.txt
Finished building: ../output/readelf.txt

Invoking: objcopy (object content copy)
/opt/Andestech/BSPv422/toolchains/nds32le-elf-mculib-v3s/bin/nds32le-elf-objcopy -S -O binary "uni_app.adx" ../output/uni_app.bin
Finished building: ../output/uni_app.bin

Invoking: size (section size listing)
/opt/Andestech/BSPv422/toolchains/nds32le-elf-mculib-v3s/bin/nds32le-elf-size "uni_app.adx" | tee ../output/.PHONY.size
  text (code + rodata)  data      bss      dec      hex filename
1293350 (213774 + 1079576)  1936    34356 1329642 1449ea uni_app.adx
Finished building: ../output/.PHONY.size

/mnt/uni_app_lite_app_hb_m
===== build done ==> output/ =====
```

(2) “output”目录下可以看到编译生成的完整固件, 使用章节 3 中的下载工具即可进行烧录使用, 目前 build.sh 脚本会同时编译生成 debug 和 release 两个版本, 如需编译特定版本或者查看 symbol 信息, 可以进入 build 目录, 运行 make 指令手动编译, 该方式生成 bin 文件及 symbol.txt 文件, 存放在 output 目录中以供查看:

```
brain@brain-VirtualBox:/mnt/uni_app_lite_app_hb_m$ ls output/ -l
总用量 1648
-rwxrwxrwx 1 root root 252110 3月  9 17:43 readelf.txt
-rwxrwxrwx 1 root root 134267 3月  9 17:43 symbol.txt
-rwxrwxrwx 1 root root 1295312 3月  9 17:43 uni_app.bin
```

## 6. 软件架构

### 6.1 软件架构

芯片源码框架采用分层设计，各个层次完成特定的功能封装，最终为二次开发提供 user 层功能接口，基本框架结构如下图所示：



各个层次功能描述如下：

- ✓ **HAL 层**：提供包括线程调度、内存管理、信号量系统级统一接口，以及 Audio、Record、GPIO、Timer、I2C 等外设驱动统一接口；
- ✓ **UTILS 层**：提供常用的工具集代码，包括如：事件队列、Ring Buffer、List、状态机、Hash 运算、Json 解析等，供 SDK 及应用层使用；
- ✓ **SDK 层**：提供核心功能接口，如：Audio Play、MP3 解码、语音识别等；

✓ **APP 层**: 语音识别基础业务功能实现, 其核心由一个事件调度和状态机构成, 将语音识别业务抽象成 wakeup、settings、music、watchdog 等不同的 session, 每个 session 维护一个状态机, 根据 SDK 层的识别结果事件驱动状态机运行, 最终实现语音识别、播报业务逻辑;

✓ **USER 层**: 提供用户二次开发接口, 包括语音识别、音频播放的控制接口, 以及 GPIO、UART、Flash、I2C、Timer 等常用的外设驱动接口, 还提供了 UCP 通用串口协议。

## 6.2 源码目录结构

源码各个文件功能如下所示:

```

├─ build -----> Makefile 系统
├─ build.sh -----> 编译脚本
├─ ci.yml -----> 自动化平台构建脚本, 对用户无用
├─ include -----> 语音识别引擎及其他自动化生成外部头文件, 不可修改
├─ lib -----> 语音识别引擎及其他底层驱动库
├─ middleware -----> RTOS 系统
├─ nds32-ae210p.ld -----> 链接信息脚本, 不可修改
├─ nds32-ae210p.sag -----> 内存段分布配置, 不可修改
├─ readme.txt -----> 发布版本信息
├─ src -----> 架构代码文件夹
|   └─ app -----> APP 层代码文件夹
|       └─ inc
|       └─ src
|           └─ main.c -----> 系统启动主程序, main 函数入口
|           └─ sessions -----> sessions 代码文件夹

```

- | | | | └─ uni\_setting\_session.c -----> setting 类事件处理 session
- | | | | └─ uni\_wakeup\_session.c -----> wakeup 类事件处理 session
- | | | | └─ uni\_watchdog\_session.c -----> watchdog 事件处理 session
- | | | └─ uni\_record\_save.c -----> 录音保存功能实现, 蜂鸟 M 暂不支持
- | | | └─ uni\_session.c -----> 创建释放 session 对象
- | | | └─ uni\_session\_manage.c -----> 管理 session 注册
- | | | └─ uni\_user\_meeting.c -----> APP 层与 USER 层交互接口
- | | └─ hal -----> HAL 层实现代码
- | | └─ sdk -----> SDK 层实现代码
- | | | └─ audio -----> Audio 播放器
- | | | └─ idle\_detect -----> 设备空闲计时管理
- | | | └─ player -----> MP3 解码器
- | | | └─ vui -----> 语音识别功能
- | | └─ utils -----> UTILS 层实现代码
- | | └─ arpt -----> ARPT 自动化测试工具
- | | └─ auto\_string -----> 变长字符串
- | | └─ bitmap -----> 二值状态变量集合
- | | └─ black\_board -----> 系统状态管理
- | | └─ cJSON -----> JSON 格式解析
- | | └─ config -----> config.bin 文件内容解析
- | | └─ crc16 -----> CRC16 算法
- | | └─ data\_buf -----> 一个不用互斥锁管理的 Ring Buffer
- | | └─ event -----> 创建事件对象

- | |— event\_list -----> 事件队列
- | |— event\_route -----> 事件分发
- | |— float2string -----> 浮点转字符串, 用于无 float 类型打印能力的 printf
- | |— fsm -----> 状态机
- | |— hash -----> 一个简易 HASH 算法
- | |— interruptable\_sleep -----> 非阻塞的 sleep 方式
- | |— list -----> 通用链表
- | |— log -----> 带等级控制的 LOG 输出接口
- | |— string -----> 一套简易的 string 操作接口
- | |— timer -----> 基于 RTOS 系统的 Timer
- | |— uart -----> 通用的 UART 接口
- |— startup -----> 芯片启动代码, 不可修改
- |— tools -----> 自动化构建工具
- | |— scripts -----> 自动化构建脚本
- | |— aik\_debug.json -----> Debug 固件对应的 AIK 配置文件
- | |— aik\_release.json -----> Release 固件对应的 AIK 配置文件
- | |— asrfix.dat -----> 声学模型
- | |— cmd\_reply\_data.json -----> UDP 平台用户定制命令词和回复语信息
- | |— config\_debug.bin -----> Debug 固件对应的应用配置文件
- | |— config\_release.bin -----> Release 固件对应的应用配置文件
- | |— custom\_config.json -----> UDP 平台用户定制系统配置信息
- | |— default\_tones -----> 默认保底音频文件文件夹
- | |— grammar.dat -----> 语法模型

- | |— grammar\_jsgf.zip -----> 语法模型对应的构建脚本
- | |— grammar.zip -----> 语法模型文件压缩包
- | |— input.txt -----> 用户定制回复语列表
- | |— pcm.bin -----> MP3 音频 flash 固件, 自动生成的中间文件
- | |— pcm\_map.txt -----> MP3 音频文件名及内容列表
- | |— res\_build\_tool.py -----> 自动化构建脚本
- | |— thresh.dat -----> 唤醒词阈值推荐表
- | |— tones -----> MP3 音频文件夹
- | |— wav\_tones -----> WAV 音频文件夹, 自动转换到 tones
- |— uni\_ci.yml -----> 构建平台脚本, 对用户无用
- └— user -----> USER 层实现代码
  - |— inc
    - | |— unione.h -----> USER 层使用的底层头文件
    - | |— user\_config.h -----> USER 可配置项, 包括串口、音量等
  - └— src
    - |— examples -----> 包含个别 USER 模块的示例代码
    - |— user\_asr.c -----> 语音识别控制接口
    - |— user\_event.c -----> USER 事件分发机制, 底层调用 USER 注册的事件回调函数
    - |— user\_file.c -----> SD 卡文件系统操作接口, 蜂鸟 M 暂不支持
    - |— user\_flash.c -----> Flash 操作接口
    - |— user\_gpio.c -----> GPIO 操作接口
    - |— user\_main.c -----> 用户代码入口, 参考示例实现 user\_main()接口以增加业务

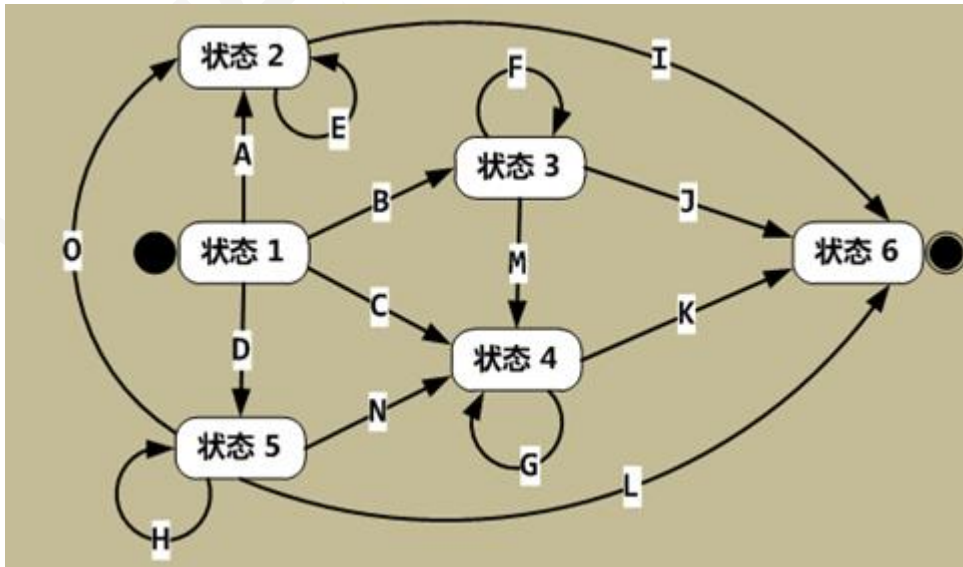
逻辑

- |— user\_player.c -----> 音频播放控制接口
- |— user\_power.c -----> 功耗操作接口, 蜂鸟 M 暂不支持
- |— user\_pwm.c -----> PWM 操作接口
- |— user\_record.c -----> 录音控制接口, 蜂鸟 M 暂不支持
- |— user\_timer.c -----> Timer 操作接口
- |— user\_uart.c -----> UART 操作接口
- |— user\_uni\_ucp.c -----> 通用串口协议操作接口

### 3.APP 层 session 状态机

为保证代码业务逻辑的额可扩展性和健壮性, 在 APP 层设计中使用了事件分发和状态机的机制, 每个 session 都维护一个独立的状态机, 底层事件分发后, 根据各个 session 状态机的状态激发对应 session 的事件处理, 如此一直运行下去。

代码中使用的是有限状态机, 也称为 FSM(Finite State Machine), 其在任意时刻都处于有限状态集合中的某一状态。当其获得一个输入事件时, 将从当前状态转换到另一个状态, 或者仍然保持在当前状态。任何一个 FSM 都可以用状态转换图来描述:



## 7. 二次开发

### 1. 运行一个 Demo 示例

为方便用户二次开发，源码中提供了大量的示例 Demo，以供参考，相关源码可在“user\src\examples”路径下找到。

如果想要运行一个已有 Demo，可通过配置“user\inc\user\_config.h”内的宏定义来选择 Demo 功能，如运行 ASR 控制相关示例，需修改两个宏定义如下：

```
#define USER_RUN_DEMO      1                // 1: 打开 Demo 功能

#define USER_RUN_DEMO_SELECT USER_DEMO_ASR_CONTROL // 选择 ASR 控制
Demo
```

然后，运行./build.sh 脚本，编译生成固件，烧录后重启开发板即可测试，默认唤醒词为“你好魔方”。

### 2. user\_main()入口函数

user\_main()接口是 USER 层二次开发入口函数，相当于系统中的 main 函数，定制化代码需加入到 user\_main()函数中以编译运行，该函数可在“user\src\user\_main.c”中找到。

从架构上看，该函数会在系初始化完毕后自动被调用，调用完毕后开始播放开机启动欢迎语，并开始进入循环等待实现 watch dog 喂狗操作，因此，作为一个限制，user\_main()函数内代码不能是阻塞的，建议该函数中仅仅做系统初始化和事件注册操作，具体实现，可以参考 Demo 程序：  
user\src\examples\hb\_smart\_ac.c。

### 3. 第一个定制程序

以下代码，作为一个简单示例，实现了用户定制的语音识别指令的打印：

```
static void _custom_setting_cb(USER_EVENT_TYPE event,

                               user_event_context_t *context) {

    event_custom_setting_t *setting = NULL;

    if (context) {

        setting =& context->custom_setting;

        printf("Action:  %s\n", setting->cmd);    // 对应 UDP 平台定制基本 action

        printf("ASR word:  %s\n", setting->word_str);    // 识别词

        printf("Reply list: %s\n", setting->reply_files);    // UDP 平台定制回复语编号列表[1,

2]

        user_player_reply_list_random(setting->reply_files);// 随机播放一条回复语

    }

}

static void _register_event_callback(void) {

    user_event_subscribe_event(USER_CUSTOM_SETTING, _custom_setting_cb);

}

int user_main(void) {
```

```
printf("Hello World !\n");

_register_event_callback(); // 注册事件回调

return 0;

}
```

运行后说“你好魔方”唤醒设备，然后说“打开风扇”打印结果为：

Hello World !

Action: ac\_power\_on

ASR word: 打开风扇

Reply list: [104, 105, 106]

然后设备播报回复语可能是一下三条其中一条（根据当前设备状态）：

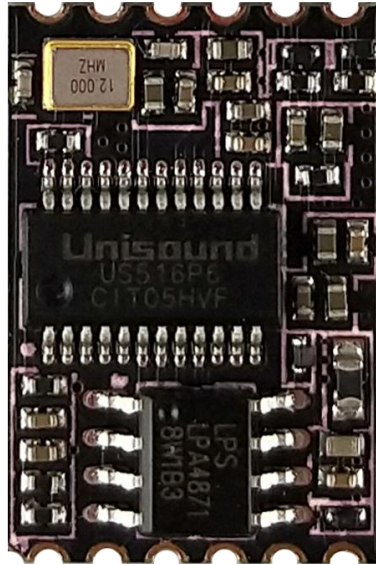
“已为您打开风扇，当前为自然风”

“已为您打开风扇，当前为正常风”

“已为您打开风扇，当前为睡眠风”

#### 4. 串口 LOG 调试

由于芯片本身只有 1 个硬件 UART，默认定义用于外设通信，因此增加 Software UART 功能，设置 B8 引脚（也可修改代码设置为其他引脚）为 log 输出 Tx 端口，在开发板上丝印为“B8”，连接 PC 后使用常见串口工具（如 putty、xshell、secureCRT 等），配置波特率（57600bps）、数据位（8）、停止位（1）、校验（None）即可。



TX 连接串口调试器 RX, GND 连接 GND(开发板上只有 1 个 GND 接口, 为方便烧录和调试, 可以将烧录器和串口调试器连接在同一台电脑或 USB Hub 上, 以保持共地, 则串口调试器只需连接 Tx (B8) 引脚即可), 设备上电后, 可在串口工具中观察到如下启动信息:

```

**\**\**\**\**\**\**\**\**\**\**\**\**\**\**\**\**\**\**\**\**\**\
**\//**\//**\//**\//**\//**\//**\//**\//**\//**\//**\//**\//
>>Unione Version : v0.0.1--2020030917
>>Build Data : 2020030918
[T] 0<main>_system_start_process:137->System start woking !
    
```

注: 在连接 PC 进行调试时, 如需掉电重启, 请将端口串口调试器与主控板连接后掉电重启, 否则可能由于串口调试器漏掉输入使开发板进入低电压模式无法掉电重启。