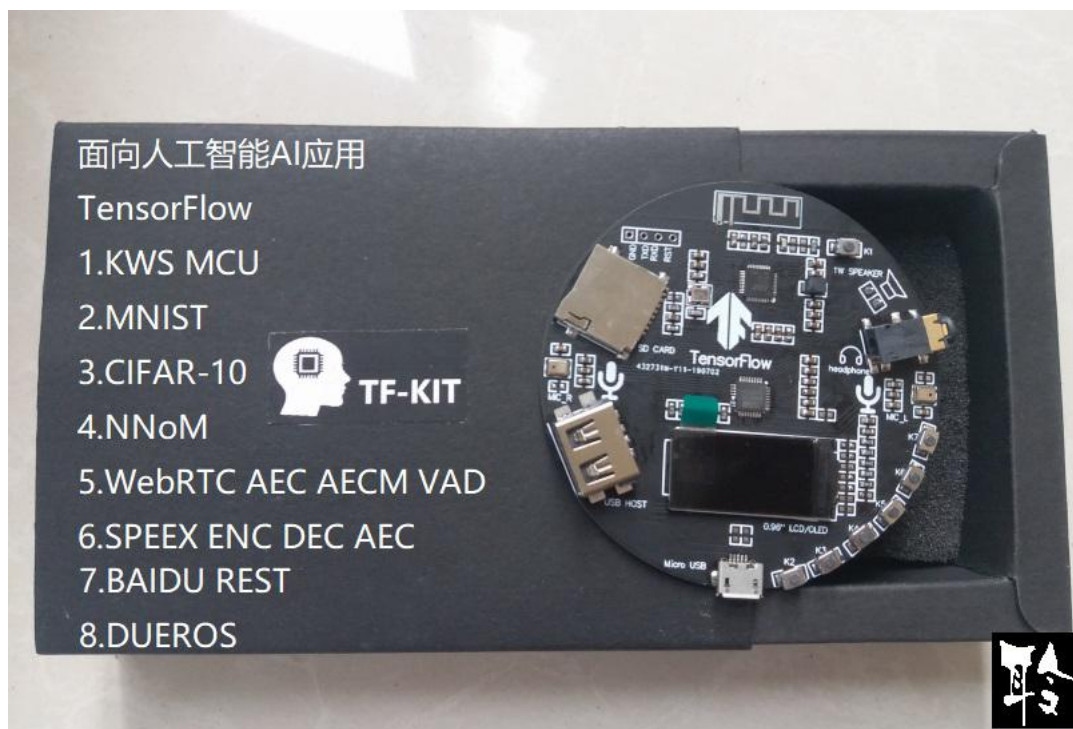
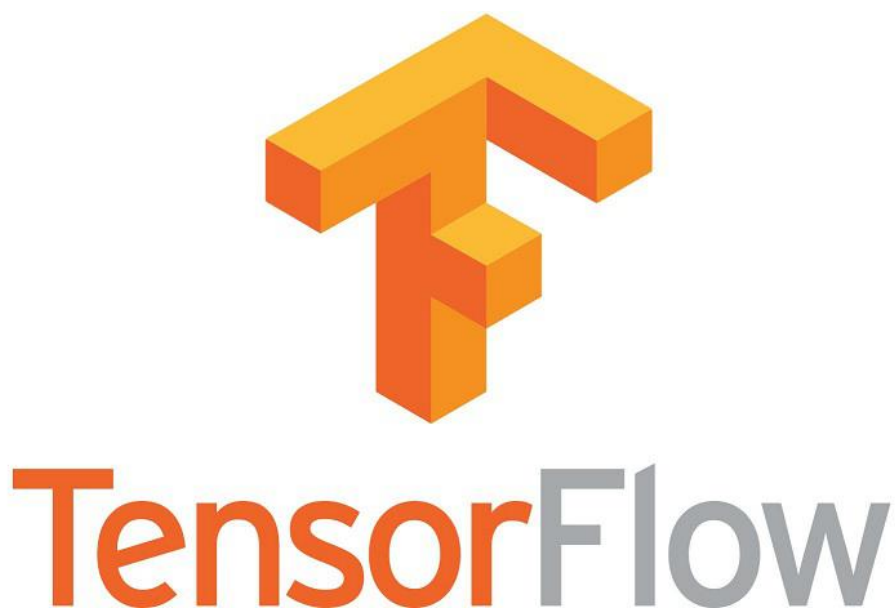


TF-KIT 人工智能开发板快速使用说明



人工智能时代来临，嵌入式开发迎来新挑战，现在 AI 的大潮势不可挡，最为著名谷歌开源的 tensorflow 加快了深度神经学习框架，成为当今新的一股热流。为此 TF-KIT 开发板应运而生！



TensorFlow 开发板资源介绍：

- 1.采用了 ARM®Cortex®-M4 主控内核，160MHZ 448K SRAM，资源丰富，I2S*2,SDIO,UART*2,SPI,HSPI,SDCARD,USB-HOST,USB-DEVICE 等丰富接口。
- 2.外接 CODEC 芯片，双声道 MEMS MIC 输入，耳机，支持 3W 喇叭输出。



3.外接 65K 全彩 LCD, 160*80 分辨率

4.扩展 SD 卡, USB 连接 U 盘, USB 摄像头

5.支持 WIFI, 2.4G 高速无线通讯,

6.支持多达 6 键 ADC KEY 能力。

tensorflow 开发板软件能力:

1.可以调试学习 TensorFlow/examples/speech_commands, 支持 DNN RNN CNN 等算法, 尤其 DS-CNN 的 KWS 高达 95%识别率

2.可以跑通著名的图像分类 FICAR-10 学习集, 32*32 图像集轻松完成。

3.支持 MNIST 28*28 手写数字精确识别。

4.加入了 NNOM 库, Octave, Dense 多种模型支持, 高效的语音模型算法。

5.WebRTC AEC AECM VAD 等语音算法, 代表着全球的前端技术。

6.SPEEX ENC DEC AEC 库, 加入了无专利的语音压缩, 消混音处理算法。

7.BAIDU REST API 百度语音识别, 合成支持。

8.著名的 DUEROS 对话式人工交互 OS。

除去上述能力, 该 KIT 的操作系统带有 Lwip 轻量级 TCP 协议, 分分钟连接外网。大量的 SDK 的例子等你来验证。

TensorFlow 的 SDK WIN 环境搭建:

Windows 开发环境需要预先安装以下软件:

Keil MDK (>= 5.17)

Python (2.7.10)

除此之外, 建议使用命令行工具 Msys2, 更方便地进行编译。

下面将详细介绍如何配置 mbed 编译环境。

1、 下载安装 Msys2: 官网: www.msys2.org 根据需要下载相应版本并安装。

2、 安装配置 mbed-cli 命令行工具 假设 Keil MDK 和 Python 安装目录分别为: C:/Keil_v5 与 C:/Python27 将路径 C:/Python27 和 C:/Python27/Scripts 添加到系统环境变量 PATH 打开 msys2 软件, 执行 python --version 和 pip --version 保证 python 工具可用 执行 pip install mbed-cli 安装 mbed-cli 工具, 安装完成后执行 mbed --version 保证工具可用

3 执行 pip install -r [SDK 根目录]/requirements.txt 安装其他编译相关的工具 执行 mbed config --global ARM_PATH C:/Keil_v5/ARM/ARMCC 设置编译器路径 3、 编译 SDK SDK 中包含若干 test case, 以 timer 为例, 需要在 SDK 根目录执行编译命令: mbed compile -m UNO_91H -t ARM --source TESTS/TARGET_RDA/timer/ --source ./ 其中各参数选项含义: -m 指定 MCU Target 名称 -t 指定编译器工具链 --source 指定参与编译的源码目录 -c 编译前 clean BUILD 目录 编译完成后将产生文件: .\BUILD\UNO_91H\ARM\timer.bin, 可以将其烧录到硬件板卡 Flash 中运行。



4、烧录 bin 文件 首先在我司 bbs 上下载“RDA5981 单口下载工具”，地址为：链接：

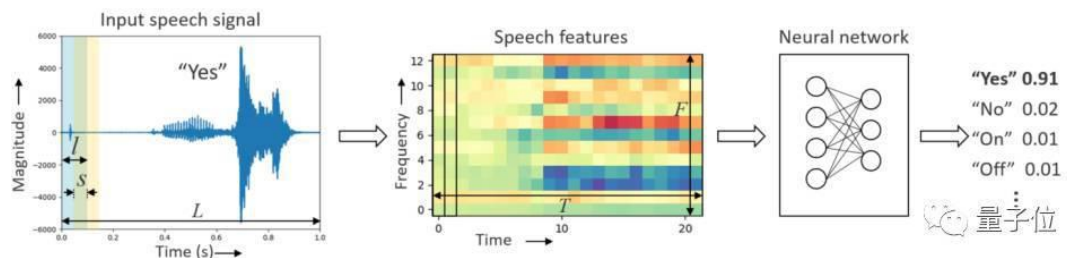
https://pan.baidu.com/s/1Umg_FWsFpRUYxE65HcWYow 提取码：15kf 将 DEMO 板通过 USB 转串口数据线连接至 PC，并记录下对应的 COM 编号。连接时只需要连接上 DEMO 板上的 TXD、RXD 和 GND 三个引脚，不同 DEMO 板的引脚位置不同，将在后文中说明。将下载下来的“RDA5981 单口下载工具”解压，运行其中的 flashtest.exe，如下图所示。在“Port”处选择上一步中对应 COM 编号，在“BinFile”处选择要烧录的 bin 文件。点击下载工具右上角的“Start”按钮，DEMO 板上电后按一下 RST 按键后开始下载，下载成功后会显示“Download finish”。

TensorFlow 的例子测试：

1, KWS 测试

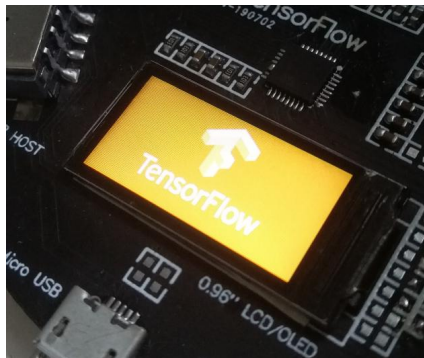
KWS 是一个把 Google 的基于 TensorFlow 开源语音识别项目(Keyword spotting, 简称 KWS)应用于 MCU 上的实例,唤醒的本质在于关键字识别(keyword spotting)，即 kws。我们的工作基于 GitHub 上的 kws 开源项目。

首先进入网址 <https://github.com/ARM-software/ML-KWS-for-MCU/tree/master/Deployment>，上面给出了在基于 arm 的开发板上运行 kws 的详细指导及源码，进入 example 目录可以看到，源码分为两个例程，分别为 simple_test 和 realtime_test，二者都是 kws 程序，区别在于在第一个程序中，KWS 是在通过.h 文件提供的音频数据上运行的；后一个程序则是实时 kws。



使用 `<mbed compile -m UNO_91H -t ARM --source TESTS/TARGET_RDA/tensorflow/ --source ./ -c>` 指令编译出 bin，烧录。

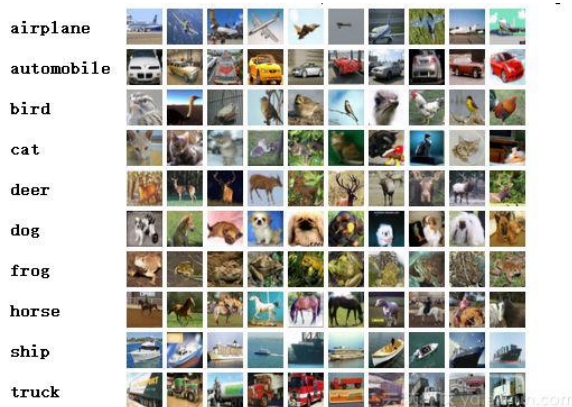
需要准备一张 FAT32 格式化的 SD 卡，拷贝一个语音样本命名 test.wav 根目录下。上电启动开发板子。如图，K3 是录音，K4 是回放，K5 是测试按键，按键 K5，会进行识别测试，并且播放 test.wav，LCD 提示结果识别时间 us，识别 ID 结果。详细实现参阅源码。



2. CIFAR-10 测试

该数据集共有 60000 张彩色图像，这些图像是 32*32，分为 10 个类，每类 6000 张图。这里面有 50000 张用于训练，构成了 5 个训练批，每一批 10000 张图；另外 10000 用于测试，单独构成一批。测试批的数据里，取自 10 类中的每一类，每一类随机取 1000 张。抽剩下的就随机排列组成了训练批。注意一个训练批中的各类图像并不一定数量相同，总的来看训练批，每一类都有 5000 张图。

下面这幅图就是列举了 10 各类，每一类展示了随机的 10 张图片：



使用 `<mbed compile -m UNO_91H -t ARM --source TESTS/TARGET_RDA/cifar10/ --source ./ -c>` 指令编译出 bin，烧录。准备一张 FAT16 格式化的 SD 卡，拷贝一个 32*32 图片样本(需要转换 bmp 格式)命名 test.bmp 根目录下。上电启动开发板子。LCD 左上角显示识别的图片，提示结果识别时间 us，识别 ID 结果。详细实现参阅源码。

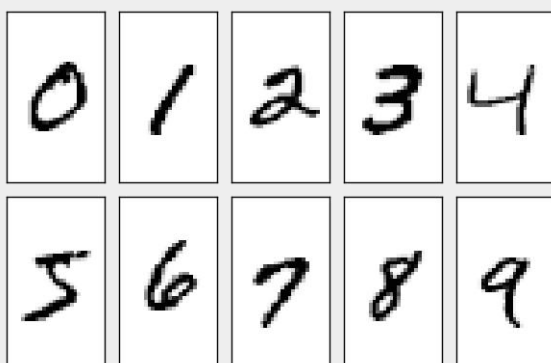
3. MNIST 测试

MNIST 数据集已经是一个被“嚼烂”了的数据集，很多教程都会对它“下手”，几乎成为一个“典范”。不过有些人可能对它还不是很了解，下面来介绍一下。

MNIST 数据集可在 <http://yann.lecun.com/exdb/mnist/> 获取，它包含了四个部分：

- Training set images: train-images-idx3-ubyte.gz (9.9 MB, 解压后 47 MB, 包含 60,000 个样本)
- Training set labels: train-labels-idx1-ubyte.gz (29 KB, 解压后 60 KB, 包含 60,000 个标签)
- Test set images: t10k-images-idx3-ubyte.gz (1.6 MB, 解压后 7.8 MB, 包含 10,000 个样本)
- Test set labels: t10k-labels-idx1-ubyte.gz (5KB, 解压后 10 KB, 包含 10,000 个标签)

MNIST 数据集来自美国国家标准与技术研究所, National Institute of Standards and Technology (NIST)。训练集 (training set) 由来自 250 个不同人手写的数字构成, 其中 50% 是高中学生, 50% 来自人口普查局 (the Census Bureau) 的工作人员。测试集 (test set) 也是同样比例的手写数字数据。



使用 `<mbed compile -m UNO_91H -t ARM --source TESTS/TARGET_RDA/nnom/ --source ./ -c>` 指令编译出 bin，上电启动开发板子。LCD 左上角显示识别的图片，提示结果识别时间 us，识别 ID 结果。K3,K4 可以选择识别的图片，注释 nnom/nnom/nnom.cpp 文件中 weights.h 选择不同的模型，详细实现参阅源码。



4. SPEEX 测试



Speex aec 测试

准备一张 SD 卡，格式化 FAT32 格式，拷贝 micin.wav 和 speaker.wav 到 sd 卡根目录，上电启动后，会自动进入 AEC 运算，micin 回音输入，speaker.wav 参考音源输入，整个文件 AEC 处理过程大约需要 1 分钟，需耐心等待，经过 AEC 处理后的数据命名 out.wav 保存 SD 卡根目录下，然后进行播放 out.wav 文件并到喇叭输出。

Speex enc 测试

speex 压缩解压的测试 8k 质量 4，准备一张 SD 卡，格式化 FAT32 格式，拷贝 micin.wav 到 sd 卡根目录，上电启动后，会自动依次调用，speex_enc()负责压缩，压缩 sd 卡根目录下的 micin.wav 后，另命名 out.spx 保存 sd 卡根目录下。，speex_dec()负责解压缩，将 sd 根目录下 out.spx 解码后另命名 fout.pcm 保存 sd 卡根目录下。，SpeexDecPlay() 负责播放刚才的 fout.pcm 文件，通过喇叭输出。



5. WebRTC AECM 测试



AECM 是移动终端使用的回音消除算法，准备一张 SD 卡，格式化 FAT32 格式，拷贝 micin.wav 和 speaker.wav 到 sd 卡根目录，上电启动后，会自动进入 AEC 运算，micin 回音输入，speaker.wav 参考音源输入，经过 AEC 处理后的数据命名 out.wav 保存 SD 卡根目录下，整个文件 AEC 处理过程大约需要 1 分钟，然后进行播放 out.wav 文件并到喇叭输出。

更多实验内容陆续更新中.....