

目录

一、简介.....	1
二、实验.....	1
三、训练.....	5

一、简介

普通 mcu 受限于资源，很难做一些复杂的深度学习。不过虽然难，但是仍然可以做。上次使用 CNN 进行手写数字识别。这次使用 RNN 进行音频降噪。本次音频降噪使用的是定点降噪，将 MFCC 和增益作为输入参数进行训练。

MFCC，梅尔频率倒谱，常用来作音频特征提取。

实际进行语音降噪（滤波）的部分，是一个均衡器，也就是大家播放器内调节低音高音。而均衡器(Equalizer)的本质是很多个平行的带通滤波器(Bandpass Filter)。我们神经网络的输出，就是均衡器内各个 filter band 的增益。RNN 通过调节不同频率的增益来达到降噪的效果。

二、实验

1 所需工具：FRDM-K64，Python3.7，pip，IAR

2 下载深度学习框架源码，<https://github.com/majianjia/nnom> 这是个纯 C 框架，并不依赖硬件结构，移植很方便

3 移植，我们选择 bubble 这个例程，将 inc，port 和 src 文件夹添加到工程中，如图

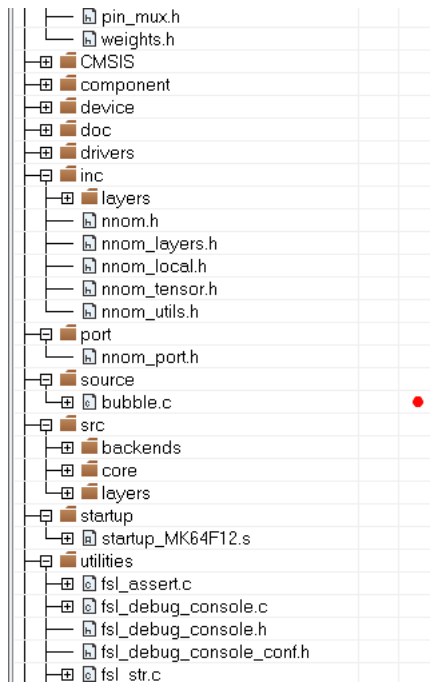
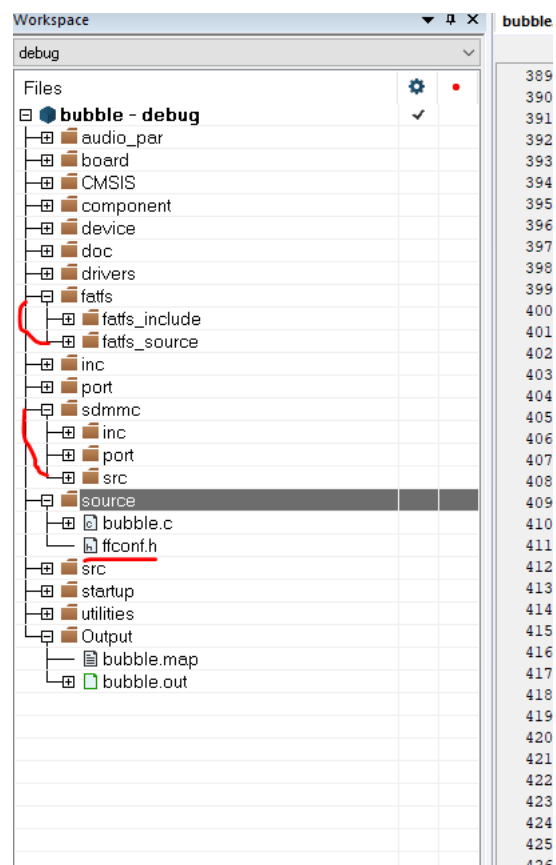


图 1

打开 port.h, 将 NNOM_LOG 的串口改成 PRINTF(_VA_ARGS_), 打开 icf 文件, 将堆大小改成 0x5000, define symbol __size_heap__ = 0x5000;
这个库用的 malloc, 就是从这里分配内存, 小了的话, 就没办法运行网络。

4 移植 fatfs 文件系统到 bubble 工程

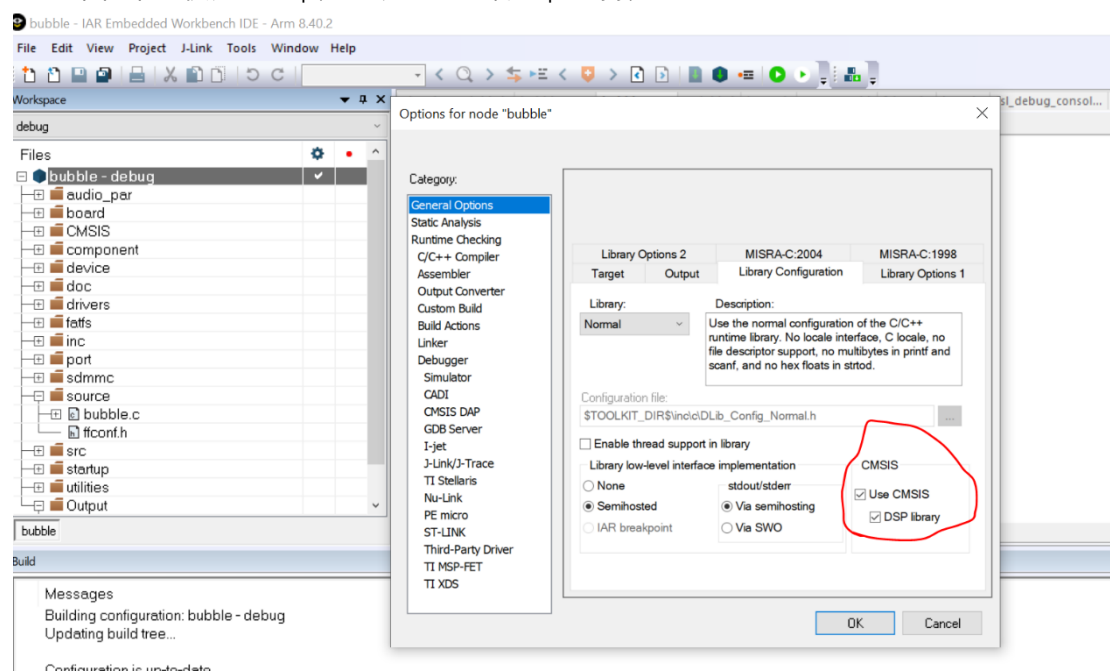


5 在 bubble.c 文件里，添加头文件

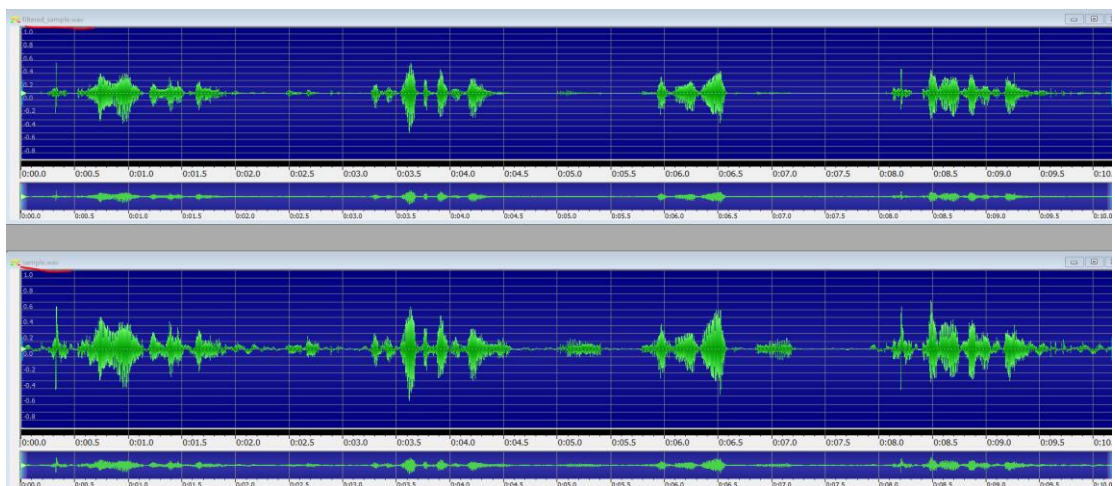
```
#include "nnom_port.h"
#include "nnom.h"
#include "weights.h"
#include "denoise_weights.h"
#include "mfcc.h"
#include "wav.h"
#include "equalizer_coeff.h"
```

6 找到框架里 examples\rnn-denoise 下的 main.c，main_arm.c 是为 stm32 使用的。Main.c 是为 windows 运行例程提供的，这个代码是通过打开音频文件，然后运行网络，最后生成降噪音频。用这个代码方便进行实验。这个文件里的关于文件操作的 api 都需要我们手动改成 mcu 的 api。我增加了进度条显示，可以看附件。

7 这个框架还使用了 dsp，所以需要开启 dsp 支持，如图



8 添加宏定义



三、训练

通过以上步骤，我们就实现了音频降噪，那么这个数据是怎么训练的？你需要安装 tensorflow，以及 keras。具体怎么训练的在 README_CN.md 文件有写。这个 tensorflow 需要安装 gpu 版本，并且需要安装 cuda。请自行查阅安装

- 1 下载语音数据 <https://github.com/microsoft/MS-SNSD>，放入 MS-SNSD 文件夹下
- 2 使用 pip3 安装 requirements.txt 里的工具
- 3 直接运行 noisyspeech_synthesizer.py，这个会报错，需要把 15 行的 float 改成 int 类型这样就不回报错了
- 4 运行脚本 gen_dataset.py，生成 mfcc 和增益 gain
- 5 运行 main.py，这样就会生成 sample 的噪音文件，可以给我们用来测试，同时也生成 weights.h
denoise_weights.h
equalizer_coeff.h