Instrumentation and Equipments 仪器与设备, 2022, 10(3), 210-216
Published Online September 2022 in Hans. http://www.hanspub.org/journal/iae
https://doi.org/10.12677/iae.2022.103028

Hans汉斯

基于STM32处理器的Codec2语音压缩算法移植

曾　裕，苏本磊，侯阳阳

西南民族大学电子信息学院，四川　成都

收稿日期：2022年8月13日；录用日期：2022年9月12日；发布日期：2022年9月21日

摘　要

本文介绍了一种语音压缩算法的代码移植及测试结果分析，从压缩算法在语音处理方面的特点出发，分析了Codec2语音编解码算法在传输中的优越性，并基于特定处理器硬件平台进行代码移植。文中论述了Codec2语音处理的算法原理、硬件电路的框架，以及代码的移植过程。该处理系统实现了语音的采集压缩以及传输，并通过测试表明该算法在低码率情况下具有低时延的特点。

关键词

语音编解码，Codec2，码率

Transplantation of Codec2 Speech Compression Algorithm Based on STM32 Processor

Yu Zeng, Benlei Su, Yangyang Hou

College of Electronic and Information, Southwest Minzu University, Chengdu Sichuan

Received: Aug. 13th, 2022; accepted: Sep. 12th, 2022; published: Sep. 21st, 2022

Abstract

This paper introduces the code transplantation of a speech compression algorithm and the analysis of the test results. Starting from the characteristics of the compression algorithm in speech processing, the superiority of the Codec2 speech encoding and decoding algorithm in transmission is analyzed, and the code is transplanted based on a specific processor hardware platform. This paper discusses the algorithm principle of Codec2 speech processing, the frame of hardware circuit, and the transplanting process of the code. The processing system realizes the acquisition, compression and transmission of speech, and the test shows that the system has the characteris-

文章引用: 曾裕, 苏本磊, 侯阳阳. 基于 STM32 处理器的 Codec2 语音压缩算法移植[J]. 仪器与设备, 2022, 10(3): 210-216. DOI: 10.12677/iae.2022.103028

tics of low bit rate and low delay.

Keywords

Voice Codec, Codec2, Code Rate

1. 引言

在现实生活中，无线通信的环境往往是不稳定的，以往的编解码技术不能提供变速率，在实际的应用中效果往往不理想。为了解决这种现状，我们需要一种提供变速率的编解码器，即随环境质量的不同而提供不同的速率。Codec2 语音编解码器支持多速率的工作要求，它能够提供多种不同的编码速率，适应不同网络及终端的实际要求。Codec2 语音编解码器还能够动态地改变比特率，使通话质量随着环境的不同而得到优化。因此，研究 Codec2 语音编解码器具有现实意义[1]。

语音压缩算法的研究始于上世纪六十年代，人们对于性能良好的语音压缩算法的研究一直没有停止。从 20 世纪 70 年代开始，随着数字信号处理理论逐渐走向成熟，工程师们开始对语音压缩算法的进行研究，并取得了一些基础性成果。1978 年 Lim 和 Opennheim 提出了语音处理的维纳滤波方法，1979 年，Boll 提出了经典谱减法，1984 年，Ephrain 和 Malah 提出了基于 MMSE 短时谱估计的语音处理算法，1987 年，Paliwal 将卡尔曼滤波引入语音处理领域，奠定了噪声抑制的理论基础。国际电信联盟(IUT)在 1996 年提出了 G729 编解码算法，其核心是共轭结构代数码激励线性预测。

Codec2 工程是由 David Rowe 在 2002 年启动的一个项目，Codec2 编解码器是一套主要针对语音的音频压缩算法。该算法可以在较低门槛的语音应用系统中实现高性能的语音编码。另外与目前其它传统的音频编解码器相比较，Codec2 算法的低时延输出性能在网络应用中也表现卓越，尤其在窄带无线网络应用上有着自己独特的优势[2]。

2. Codec2 压缩算法介绍

Codec2 编解码器最初目标是为了实现一种能够利用低时延信道传递语音信息的算法，对于业余无线电来说，这种方式格外地适合 HF 和 VHF 频段上的语音信号传输。Codec2 编解码器是一种开源的 D-Star AMBE 制式。与闭源模式相比，它是全开放的，爱好者们可以通过自己的修改和剪裁来为其他的硬件来增加 Codec2 支持[3]。

2.1. Codec2 算法基本原理

采用正弦波的总和来对语音进行建模如式(1)，式(2)：

$$\text{for}(m=1;m<=L;m++) \tag{1}$$

$$s[n]+=A[m]*\cos(w_o*m*n+phi[m]); \tag{2}$$

正弦波是基频 Wo(omega-naught)的倍数，对于每一帧，我们分析语音信号并提取一组参数如式(3)：

$$w_o,\{A\},\{phi\} \tag{3}$$

{A}是一组 L 振幅，{phi}是一组 L 相位。选择 L 等于 4 kHz 带宽中可以容纳的谐波数如式(4)：

$$L = \text{floor}(pi / w_o) \tag{4}$$

Wo 以归一化为 4 kHz 的弧度表示，使得 pi 弧度 = 4 kHz。以 Hz 为单位的基频如式(5)：

$$F_0 = (8000 / (2 * pi)) * w_o \tag{5}$$

其次，我们需要编码 Wo，{A}，{phi}，并将它们传输到重建语音的解码器。一帧的长度可能为 10~20 ms，因此我们每 10~20 ms 更新一次参数(100 到 50 Hz 更新速率)。

2.2. Codec2 编解码器框图

语音音频是通过将语音建模为正弦波谐波的总和来重建的，这些正弦波称为线谱对 LSP (Link-State Packet)的独立振幅，位于说话者声音(音高)的确定基频之上。谐波的(量化)音高和幅度(能量)被编码，LSP 以数字格式通过通道交换。LSP 系数代表频域中的线性预测编码 LPC (Link Control Protocol)模型，有助于实现 LPC 参数的稳健而高效的量化[4]。其编解码流程图如图 1，图 2 所示。

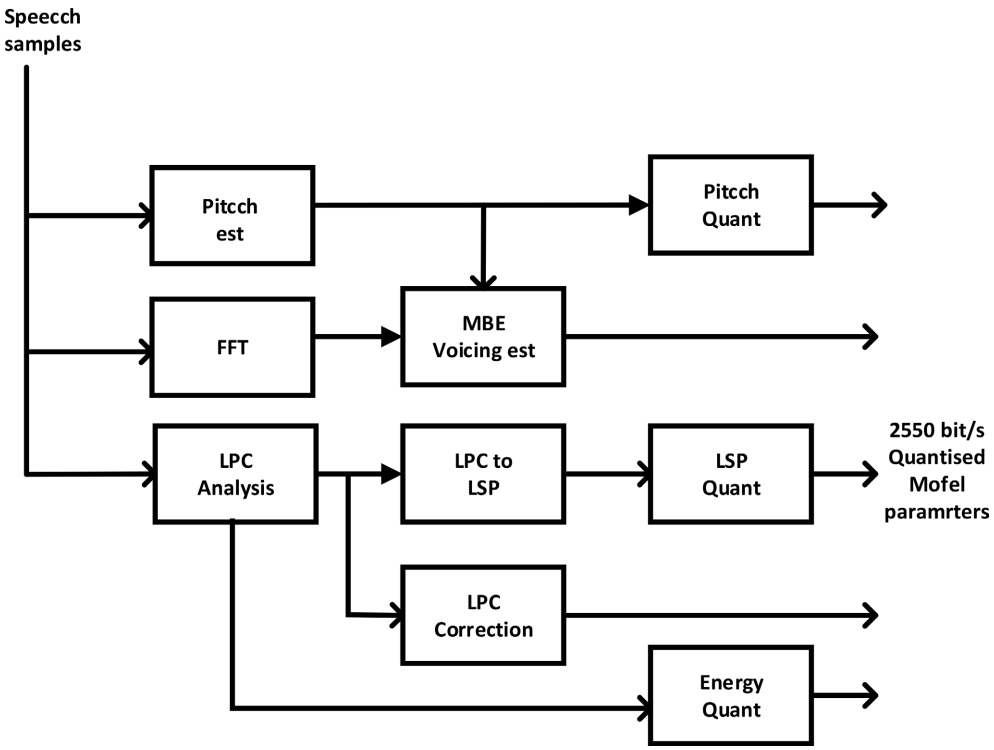

Figure 1. Codec2 encoding flow chart
图 1. Codec2 编码流程图

3. STM32F4 处理器的应用设计

3.1. STM32F4 系列处理器介绍

STM32F405XX 系列是基于高性能 ARM®Cortex®-M4 32 位 RISC 核心，其工作频率最高可达 168 MHZ。Cortex-M4 内核配有 1 个 FPU 浮点单元(floating point unit，浮点单元)，支持所有的 ARM (random-access memory)数据类型以及单精度数据处理指令。此外它配有内存保护单元 MPU (Micro Processor Unit)可以执行一套完整的 DSP (Digital Signal Processing)指令[5]。

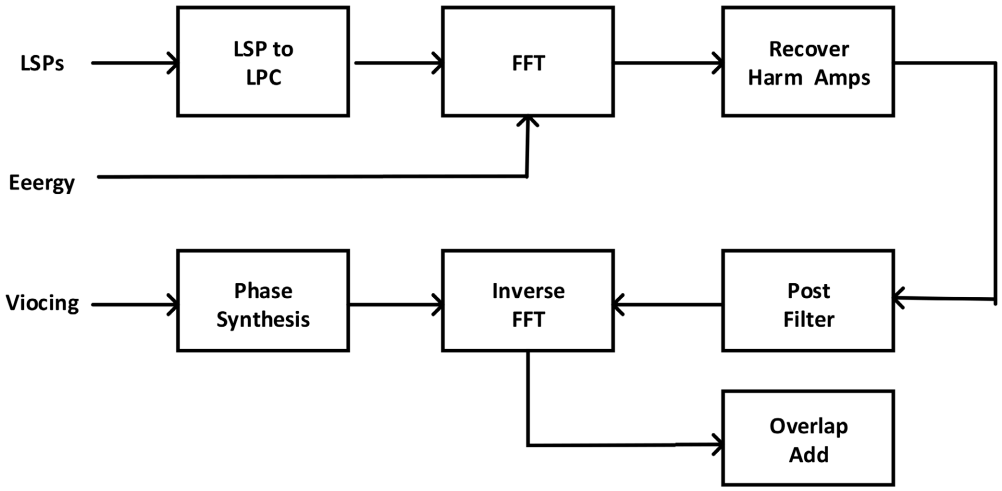

Figure 2. Codec2 decoding flow chart
图 2. Codec2 解码流程图

STM32F405xx 系列配备了最高可达 192 千字节的高速嵌入式存储器和大容量的备份 SRAM (Static Random-Access Memory)以及外围设备和增强的 I/O [6]。所有的设备都配备了 ADC、DAC、低功耗 RTC (Real_Time Clock)、通用 16 位定时器，这些都提高了芯片的整体性能，为实现许多复杂的算法提供了条件[7]。

3.2. 硬件流程设计

硬件中断事件驱动 WM8974 进行数据采集，SRAM 来进行数据缓冲暂存，当完成数据采集之后，再次触发中断函数，在中断函数设置数据采集结束的信号标志，主线程通过 Codec2 来完成数据的编码，编码完成后压缩的 Codec2 数据再通过串口发送到接收端处理，处理完成后通过 iis 总线送到 WM8947 进行语言输出[8]。STM32F405 内部资源配置设计如图 3 所示。语言系统发送端和接收端的 PCB 板如图 4，图 5 所示。

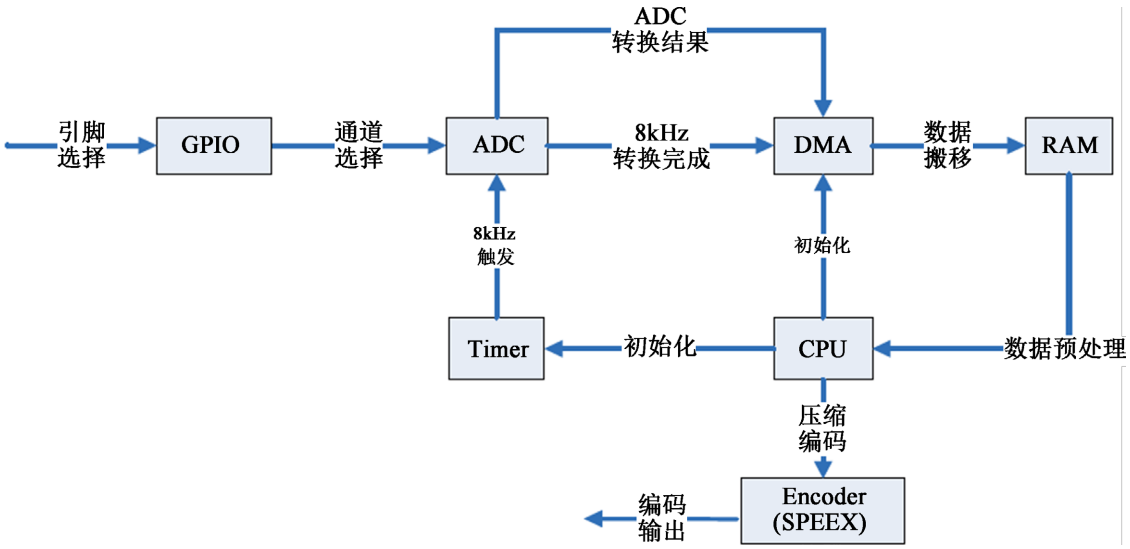

Figure 3. STM32F405 internal resource configuration design
图 3. STM32F405 内部资源配置设计

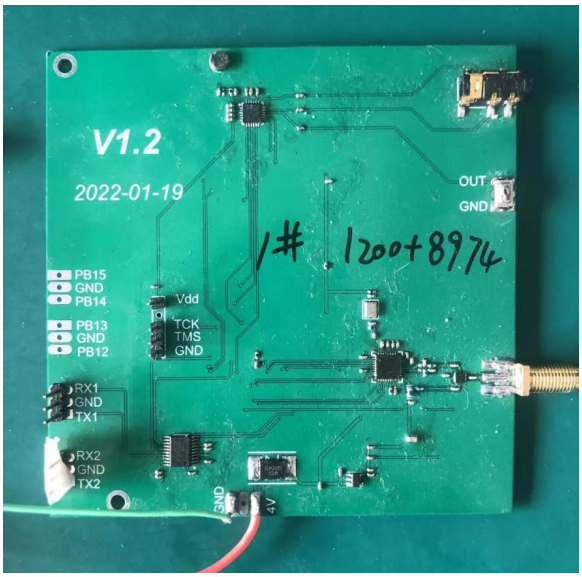

Figure 4. Language system sender
图 4. 语言系统发送端

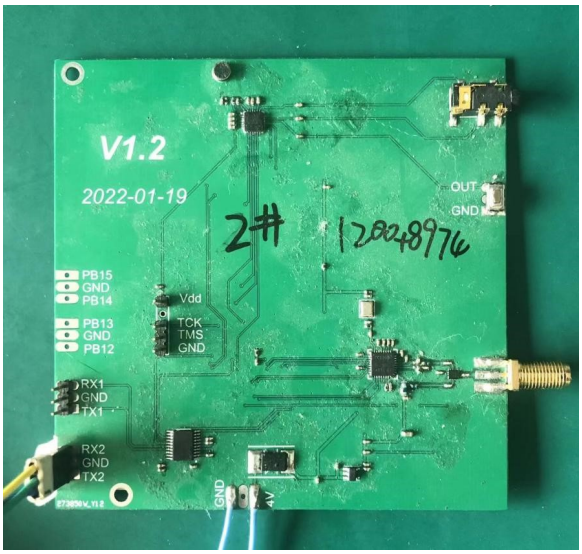

Figure 5. Language system receiver
图 5. 语言系统接收端

3.3. 软件流程设计

软件采用单线程框架结构，当完成一帧数据采集后，触发中断函数，在中断函数中设置数据采集完成标志信号，主线程获取一帧语音数据并调用编码器函数进行数据压缩编码，编码器返回压缩后的Codec2数据及长度值。软件流程如图 6 所示。

4. Codec2 算法移植

Codec2 音频压缩算法的核心部分与硬件平台没有关系。数据采样操作，不需要调整移植算法的代码。只需要更改与特定硬件平台相关的部分，例如音频信号的进入接口，数据缓冲地址以及内存资源的分配[7]。配置文件 config.h 的部分内容如图 7 所示。

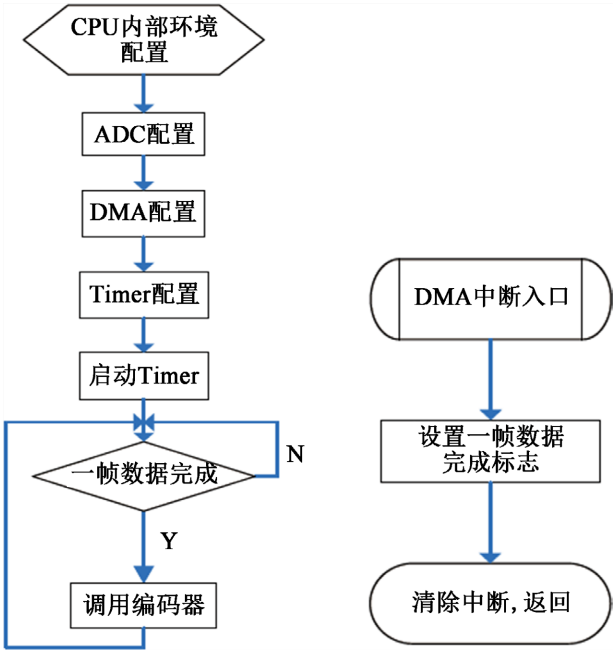

Figure 6. The software process design of speech acquisition and coding
图 6. 语音采集编码软件流程设计

```
#define FIXED_POINT                                  / / 采用固定点运算
#define DISABLE_FLOAT_API                            / / 禁用浮点接口
#define DISABLE_VBR                                  / / 禁用可变比特率
#define FRAME__S IZE 160                             / / 帧大小为160
#define DISABLE_WIDEBAND                             / / 禁用Codec宽带模式

#define SPEEXENC PERSIST STACK SIZE 5000
#define NB_ENC_STACK      SPEEXENC_SCRATCH_STACK_SIZE      / / 窄带模式编
                                                           / / 堆栈大小为500
#define SPEEXDEC_PERSIST_STACK_SIZE 2500
#define NB DEC STACK SPEEXDEC SCRATCH STACK SIZE         / / 窄带模式解码
                                                           / / 堆栈大小为2500
```

Figure 7. config.h file configuration
图 7. config.h 文件配置

编码步骤：

1) 定义一个 CodecBit 类型变量 ebits 和一个 Codec 编码器状态变量 enc_state。

2) 调用 Codec_bits_init(&ebits)初始化。

3) 调用 Codec_encoder_init(&Codec_nb_mode)来初始化 enc_state。其中 Codec_nb_mode 是 CodecMode 类型的变量，代表的是窄带模式。还有 Codec_wv_mode 表示宽带模式、Codec_uwb_mode 表示超宽带模式。

4) 调用函数 intCodec_encoder_ctl(void *state, int request, void *ptr)来设定编码器的参数，其中参数 state 表示编码器的状态；参数 request 表示要定义的参数类型，如 CODEC GET FRAME SIZE 表示设置帧

大小，CODEC_SET_QUALITY 表示量化大小，这决定了编码的质量；参数 ptr 表示要设定的值。

5) 编码结束后，调用函数 Codec_bits_destory(&ebits)来销毁编码器。

解码步骤和编码步骤类似，详细过程不再叙述。

5. 整机测试

测试 Codec2 算法的处理性能时，系统时钟主频为 600 MHz，选择的音频文件总长 2039 帧，语音编码速率设置在 2400~9600 bps 范围内时。在编码之前和解码之后采集时间值，两个结果的差值就是编解码测试文件的总的时间。测试结果如表 1 所示。

Table 1. Time-consuming table of different codec algorithms
表 1. 不同编解码算法耗时表

项目 / 算法	总耗时	平均一帧耗时
G729A 标准算法	19,119 (ms)	9.37 (ms)
Codec2 算法	4710 (ms)	2.31 (ms)

以上测试结果表明：将 G729A 标准算法源代码直接移植到硬件上，在码率为 2550 以下时，Codec2 算法语音质量明显比 G729A 算法好。处理 2039 帧语音数据总耗时为 19,119 ms，G729A 算法平均处理一帧语音数据的耗时为 9.37 ms，基本能够达到算法实现实时编解码的要求(<10 ms)，但属于临界区域。将 Codec2 算法移植到硬件上，总耗时减少为 4710 ms，平均处理一帧语音数据的时间为 2.31 ms，完全满足语音实时处理的要求，在低码率情况下实现了低时延的效果。

6. 结束语

本文基于 STM32F4 系列处理器实现了 Codec2 算法移植，在针对语音传输的应用系统中，可根据实际的网络环境，通过配置不同的语音编解码算法，满足了实时语音通话的业务需求。在系统测试中，该算法具有低码率的情况下低时延的优越性。

参考文献

[1] 秦腾祥. 高音质 Audio Codec 中数字滤波器的设计与实现[D]: [硕士学位论文]. 武汉: 华中科技大学, 2019. https://doi.org/10.27157/d.cnki.ghzku.2019.002973

[2] 王峰. 基于 ARM 平台的语音编解码算法分析及其在无线传输系统的设计与实现[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2019.

[3] 钟结实, 张科. Speex 语音编解码的研究及实现[J]. 计算机测量与控制, 2010, 18(8): 1883-1885.

[4] 付璐, 刘哲, 韩琨. 基于 Speex 算法的语音压缩解压缩程序设计[J]. 电脑知识与技术, 2012, 8(26): 6357-6359+6412.

[5] 綦振禄, 杨宏, 崔元成. 基于 STM32 处理器的 Speex 语音压缩算法移植[J]. 信息技术与信息化, 2020(11): 59-63.

[6] 肖娟, 张志强. 基于 STM32 的实时语音处理系统设计[J]. 电子世界, 2014(15): 152.

[7] 李华辉, 肖云波, 沈勇, 邓斌. 基于 FreeRTOS 和 Speex 编解码器的语音处理系统设计[J]. 单片机与嵌入式系统应用, 2022, 22(2): 81-84+87.

[8] 王玉琼, 赵圆圆, 尚宁, 张小雷. 基于编译码算法的设计与仿真研究[J]. 成都工业学院学报, 2021, 24(4): 45-48. https://doi.org/10.13542/j.cnki.51-1747/tn.2021.04.008