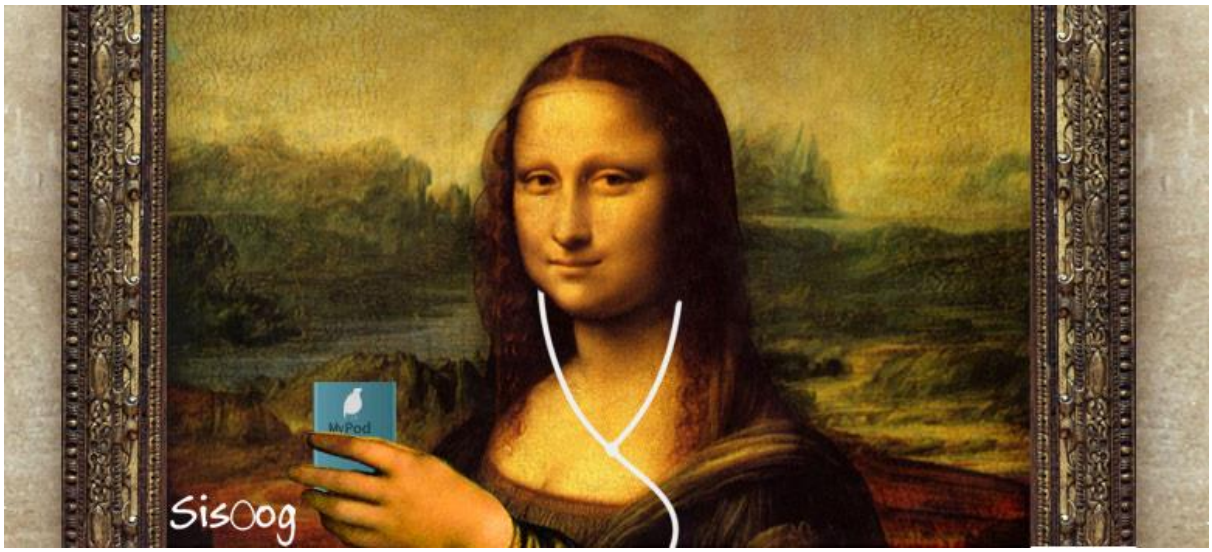
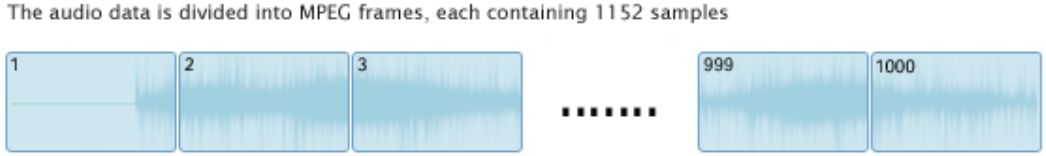
تهیه شده توسط تیم سیسوگ



همه چیز درباره دکد نرم افزاری MP3 به کمک میکروکنترلر به همراه سورس کد

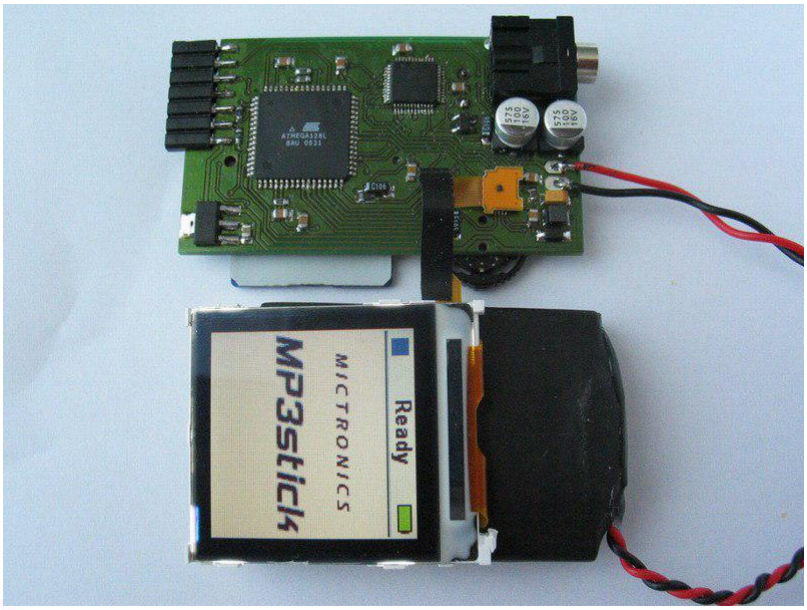
مقدمه

در مقاله پیشین فایل MP3 چیست و از کجا آمده است به چگونگی پیدایش فایل های MP3 و مزیت ها و قابلیت های آن پرداختیم ، در این مقاله قصد داریم که ساز و کار و نحوه دکد و پخش فایل های MP3 را تشریح کنیم. هامنطور که در مقاله پیش اشاره شد فایل mp3 بر خلاف فایل های wav جهت کاهش حجم فایل صوتی ، دست به فشرده سازی و رمزگذاری صدا می کند و صدا را به بخش های کوچکی که فریم نام دارد می شکند.



برای این که بتوانیم خروجی مناسب صوتی داشته باشیم ، باید قادر باشیم که فریم های مجزا را در زمان مناسبی از حالت فشرده خارج کنیم و برای پخش به واحد دیجیتال به آنالوگ بفرستیم و هیچ وقفه ای هم نباید وجود داشته باشد تا قطعی و پرش در صوت ایجاد نشود ، با توجه به قابلیت های میکروکنترلر های قدیمی ، این امکان وجود نداشت چرا که به دلیل محدودیت های حافظه و پردازشگر ، سرعت تبدیل فریم ها به فایل خام پایین تر از بیت ریت پخش صوت بود و همین امر باعث می شد که این نوع میکروکنترلر ها قادر به پخش این گونه فایل ها نباشند. برای رفع این نقص شرکت های تولید کننده چیپ ، دست به کار شدند و دکدر سخت افزاری خود را وارد بازار کردند. اغلب استفاده کننده ها برای پخش فایل های صوتی با فرمت MP3 دست به دامن این گونه چیپ ها می شدند.

یکی از شرکت های شناخته شده در خصوص تولید چیپ های دکدر شرکت VLSI است که به واسته چیپ VS1003 در ایران محبوبیت زیادی پیدا کرد. و چه پخش کننده های همراه و غیر همراهی که با استفاده از این چیپ طراحی نشد. به عنوان نمونه عکس زیر که به کمک همین چیپ و میکروکنترلر AVR یک پخش کننده MP3 قابل حمل ساخته شده است.

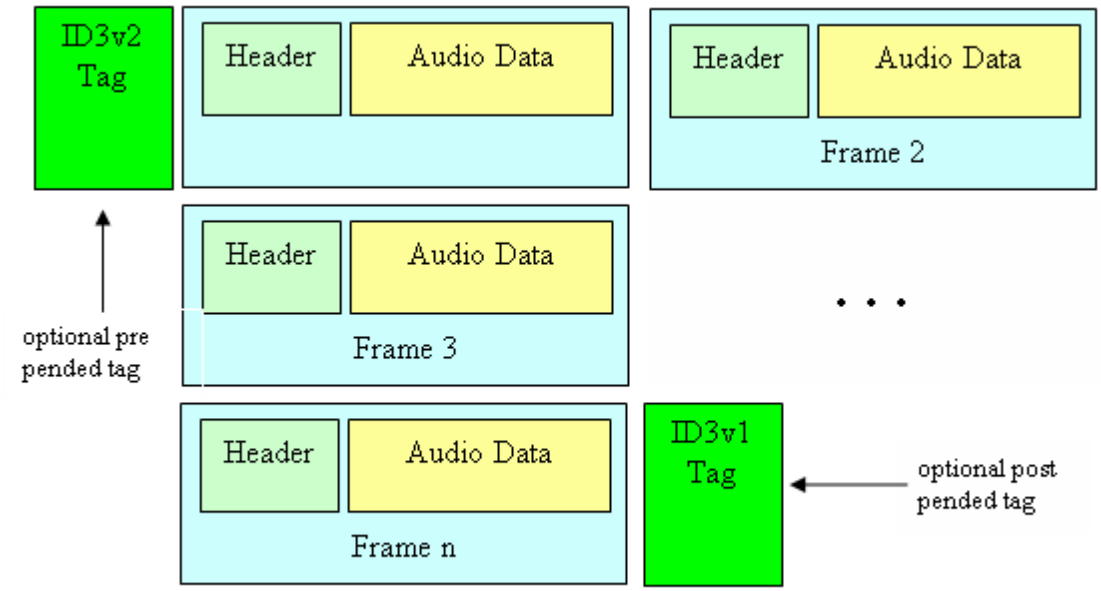


با این وجود ، این گونه طراحی ، مشکلات خاص خودش را به همراه داشت ، از جمله حجیم شدن مدار ، مصرف انرژی بیشتر ، بالا رفتن قیمت تمام شده و .... ، اما چاره دیگری وجود نداشت و استفاده از این نوع چیپ ها اجتناب ناپذیر بود.

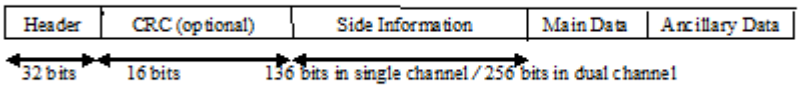
تا این که میکروکنترلر های ARM معرفی شدند که علاوه بر مصرف پایین ، سرعت پردازشی به مراتب بالاتر داشتند ، میزان حافظه و سرعت پردازش بالای این خانواده از میکروکنترلر ، امکان دکد و پخش مستقیم فایل mp3 را ایجاد کرد به نحوی که برای دکد و پخش نیاز به هیچ المان خارجی نباشد و تمام فرایند ها در درون خود میکروکنترلر میسر باشد.

استخراج فریم ها

تا اینجا با ساختار کلی فایل MP3 آشنایی پیدا کردیم ، برای شروع فرایند دکد و پخش ، اولین قدم پیدا کردن فریم ها و استخراج آنها از فایل است ، همانطور که میدانید یک فایل MP3 تشکل شده است از مجموعه از فریم های صوتی.



همانطور که در عکس فوق مشاهده می کنید ، هر فریم داده دارای هدری است که علاوه بر مشخص کردن شروع یک فریم ، مجموعه ای از اطلاعات در خصوص آن فریم است. برای روشن تر شدن موضوع به عکس زیر دقت کنید



هر فریم دارای جزئیات فوق است که 32 بیت(4 بایت) اول اطلاعات سینک و هدر را در خود دارد و 16 بیت(2 بایت) بعدی که به صورت آپشنال است یعنی ممکن است وجود نداشته باشد حاوی اطلاعات اعتبار سنجی است و 256/136 بیت بعدی حاوی اطلاعت مربوط به کانال صوتی است (همانطور که میدانید یک فایل صوتی ممکن است منو یا استریو باشد) و مابقی داده ها مربوط به داده های صوتی است.

برای این که بتوانیم داده های صوتی را از حالت فشرده و رمز شده خارج کنیم ، لازم است اول اطلاعات مربوط به فریم را از هدر استخراج کنیم. و بر اساس داده های به دست آمده شروع به رمزگشایی داده های صوتی بکنیم.

دکد – Mp3 رمزگشایی هدرMP3

همانطور که قبلا اشاره کردیم هدر حاوی 32 بیت است که هر بیت معنی و مفهوم خاص خود را دارد ،اگر بیت ها را به تفکیک و رنگ مرتبط با معنی آنها دسته بندی کنیم ، نتیجه به شکل زیر خواهد بود.

AAAAAAAA AAABBCCD EEEEFFGH IIJJKLMM

- 11 بیت) (A) از بیت 21 تا 31 : (بیت های شناسایی (همه 1 هستند) که برای تشخیص شروع یک فریم به کار برده می شوند.
- 2بیت) (B) بیت 20 و 19 : (ورژن هدر می باشد که حاوی 4 حالت است.
  - 00 – MPEG Version 2.5
  - 01 – reserved
  - 10 – MPEG Version 2
  - 11 – MPEG Version 1
- 2بیت) (C) بیت 18 و 17 :(توضیح دهنده لایه
  - 00 – reserved
  - 01 – Layer III
  - 10 – Layer II
  - 11 – Layer I
- 1بیت) (D) بیت 16 :(بیت اعتبار سنجی
  - 0– قسمت CRC وجود خواهد داشت.
  - 1– قسمت CRC وجود نخواهد داشت.
- 4بیت) (E) از بیت 12 تا 15 : (بیت ریت خروجی (مطابق جدول زیر)

Bitrate index

bits	V1,L1	V1,L2	V1,L3	V2,L1	V2, L2 & L3
0000	free	free	free	free	free
0001	32	32	32	32	8
0010	64	48	40	48	16
0011	96	56	48	56	24
0100	128	64	56	64	32
0101	160	80	64	80	40
0110	192	96	80	96	48
0111	224	112	96	112	56
1000	256	128	112	128	64
1001	288	160	128	144	80
1010	320	192	160	160	96
1011	352	224	192	176	112
1100	384	256	224	192	128
1101	416	320	256	224	144
1110	448	384	320	256	160
1111	bad	bad	bad	bad	bad

NOTES: All values are in kbps
V1 - MPEG Version 1
V2 - MPEG Version 2 and Version 2.5
L1 - Layer I
L2 - Layer II
L3 - Layer III

- دو بیت (F) بیت 10 و 11 :سمپل ریت مطابق جدول زیر.

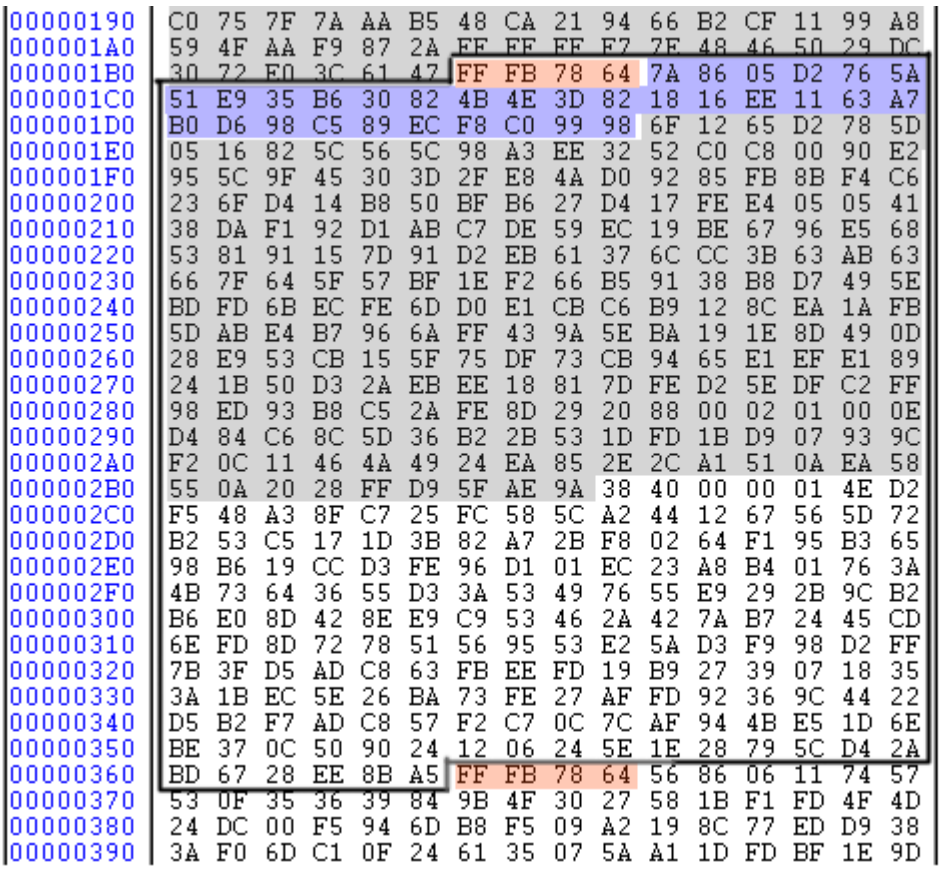
Sampling rate frequency index

bits	MPEG1	MPEG2	MPEG2.5
00	44100 Hz	22050 Hz	11025 Hz
01	48000 Hz	24000 Hz	12000 Hz
10	32000 Hz	16000 Hz	8000 Hz
11	reserv.	reserv.	reserv.

- بیت) (G) بیت شماره 9Padding bit :)
  - 0 – frame is not padded
  - 1 – frame is padded with one extra slot

- بیت) (H) بیت 8. Private bit. This one is only informative. :)
- 2بیت) (I) بیت 6 و 7Channel Mode :)
  - 00 – Stereo
  - 01 – Joint stereo (Stereo)
  - 10 – Dual channel (2 mono channels)
  - 11 – Single channel (Mono)
- 2بیت) (J) بیت 4 و 5(Mode extension (Only used in Joint stereo :)
- بیت) (K) بیت 3Copyright :)
  - Audio is not copyrighted
  - Audio is copyrighted
- بیت) (L) بیت 2Original :)
  - 0 – Copy of original media
  - 1 – Original media
- 2بیت) (M) بیت 0 و 1Emphasis :)

بعد از چهار بایت توصیف کننده هدر بنا به وضعیت بیت D در این توصیف کننده ممکن است که 2 بایت CRC وجود داشته باشد یا نه ، بعد از این دو بایت اختیاری ، بسته به این که فریم تک کانال باشد یا دو کانال (دو بیت 6 , 7)، 17 یا 32 بایت اطلاعات مربوط به کانال ها وجود خواهد داشت ، و بعد از آن داده های صوتی قرار خواهند داشت. دقت داشته باشید در شرایط خیلی نادر ممکن است که فریم دارای داده های صوتی نباشد. در عکس زیر یک فریم را درون یک فایل MP3 مشخص کرده ایم که 4 بایت قرمز رنگ هدر و 32 بایت آبی رنگ اطلاعات کانال می باشد باقیه بایت خا هم داده های صوتی هستند. توجه داشته باشید که باید های مروبط به خطایابی وجود ندارد.

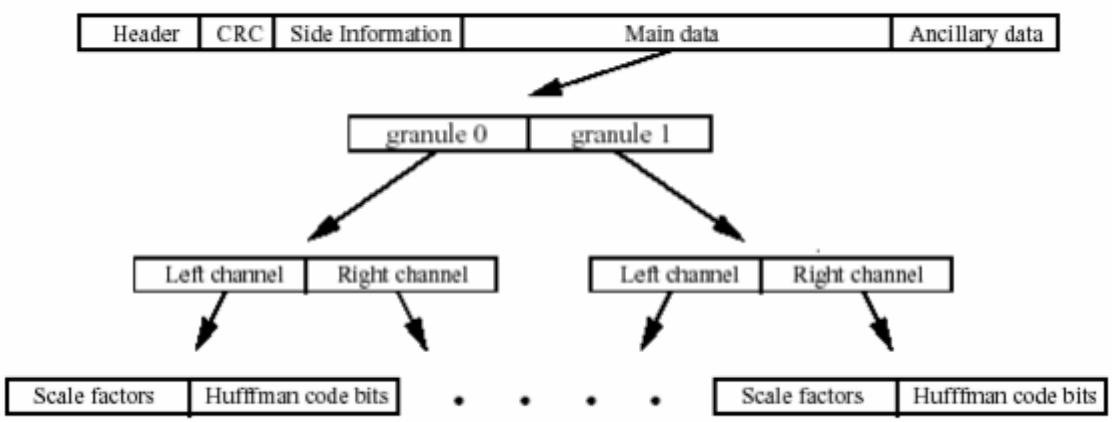


```
00000190  C0 75 7F 7A AA B5 48 CA 21 94 66 B2 CF 11 99 A8  Àu▯zª▯HÊ!▯f²Ï▯▯¨
000001A0  59 4F AA F9 87 2A FF FF FF F7 7F 48 46 50 29 DC  YOªù▯*ÿÿÿ÷^HFP)Ü
000001B0  30 72 E0 3C 61 47 FF FB 78 64 7A 86 05 D2 76 5A  0rà<aGÿûxdz▯▯Òv
000001C0  51 E9 35 B6 30 82 4B 4E 3D 82 18 16 EE 11 63 A7  Qé5¶0▯KN=▯▯▯î▯c§
000001D0  B0 D6 98 C5 89 EC F8 C0 99 98 6F 12 65 D2 78 5D  °Ö▯Å▯ìøÀ▯▯o▯eÒx]
000001E0  05 16 82 5C 56 5C 98 A3 EE 32 52 C0 C8 00 90 E2  ▯▯▯\V\▯£î2RÀÈ▯▯â
000001F0  95 5C 9F 45 30 3D 2F E8 4A D0 92 85 FB 8B F4 C6  ▯\▯E0=/èJÐ´▯û▯ôÆ
00000200  23 6F D4 14 B8 50 BF B6 27 D4 17 FE E4 05 05 41  #oÔ▯¸P¿¶'Ô▯þä▯▯A
00000210  38 DA F1 92 D1 AB C7 DE 59 EC 19 BE 67 96 E5 68  8Úñ´Ñ«ÇÞYì▯¾g▯åh
00000220  53 81 91 15 7D 91 D2 EB 61 37 6C CC 3B 63 AB 63  S▯´▯}´Òëa7lÌ;c«c
00000230  66 7F 64 5F 57 BF 1E F2 66 B5 91 38 B8 D7 49 5E  f▯d_W¿▯òfµ´8¸×I^
00000240  BD FD 6B EC FE 6D D0 E1 CB C6 B9 12 8C EA 1A FB  ½ýkìþmÐáËÆ¹▯▯ê▯û
00000250  5D AB E4 B7 96 6A FF 43 9A 5E BA 19 1E 8D 49 0D  ]«ä·▯jÿC▯^º▯▯▯I▯
00000260  28 E9 53 CB 15 5F 75 DF 73 CB 94 65 E1 EF E1 89  (éSË▯_ußsË▯eáïá▯
00000270  24 1B 50 D3 2A EB EE 18 81 7D FE D2 5E DF C2 FF  $▯PÓ*ëî▯▯}þÒ^ßÂÿ
00000280  98 ED 93 B8 C5 2A FE 8D 29 20 88 00 02 01 00 0E  ▯í▯¸Å*þ▯) ▯▯▯▯▯▯
00000290  D4 84 C6 8C 5D 36 B2 2B 53 1D FD 1B D9 07 93 9C  Ô▯Æ▯]6²+S▯ý▯Ù▯▯▯
000002A0  F2 0C 11 46 4A 49 24 EA 85 2E 2C A1 51 0A EA 58  ò▯▯FJI$ê▯.,¡Q▯êX
000002B0  55 0A 20 28 FF D9 5F AE 9A 38 40 00 00 01 4E D2  U▯ (ÿÙ_®▯8@▯▯▯NÒ
000002C0  F5 48 A3 8F C7 25 FC 58 5C A2 44 12 67 56 5D 72  õH£▯Ç%üX\¢D▯gV]r
000002D0  B2 53 C5 17 1D 3B 82 A7 2B F8 02 64 F1 95 B3 65  ²SÅ▯▯;▯§+ø▯dñ▯³e
000002E0  98 B6 19 CC D3 FE 96 D1 01 EC 23 A8 B4 01 76 3A  ▯¶▯ÌÓþ▯Ñ▯ì#¨´▯v:
000002F0  4B 73 64 36 55 D3 3A 53 49 76 55 E9 29 2B 9C B2  Ksd6UÓ:SIvUé)+▯²
00000300  B6 E0 8D 42 8E E9 C9 53 46 2A 42 7A B7 24 45 CD  ¶à▯B▯éÉSF*Bz·$EÍ
00000310  6E FD 8D 72 78 51 56 95 53 E2 5A D3 F9 98 D2 FF  ný▯rxQV▯SâZÓù▯Òÿ
00000320  7B 3F D5 AD C8 63 FB EE FD 19 B9 27 39 07 18 35  {?Õ-Ècûîý▯¹'9▯▯5
00000330  3A 1B EC 5E 26 BA 73 FE 27 AF FD 92 36 9C 44 22  :▯ì^&ºsþ'¯ý´6▯D"
00000340  D5 B2 F7 AD C8 57 F2 C7 0C 7C AF 94 4B E5 1D 6E  Õ²÷-ÈWòÇ▯|¯▯Kå▯n
00000350  BE 37 0C 50 90 24 12 06 24 5E 1E 28 79 5C D4 2A  ¾7▯P▯$▯▯$^▯(y\Ô*
00000360  BD 67 28 EE 8B A5 FF FB 78 64 56 86 06 11 74 57  ½g(î▯¥ÿûxdV▯▯▯tW
00000370  53 0F 35 36 39 84 9B 4F 30 27 58 1B F1 FD 4F 4D  S▯569▯▯O0'X▯ñýOM
00000380  24 DC 00 F5 94 6D B8 F5 09 A2 19 8C 77 ED D9 38  $Ü▯õ▯m¸õ▯¢▯▯wíÙ8
00000390  3A F0 6D C1 0F 24 61 35 07 5A A1 1D FD BF 1E 9D  :ðmÁ▯$a5▯Z¡▯ý¿▯▯
```

دکد – Mp3 رمز گشایی داده های صوتی

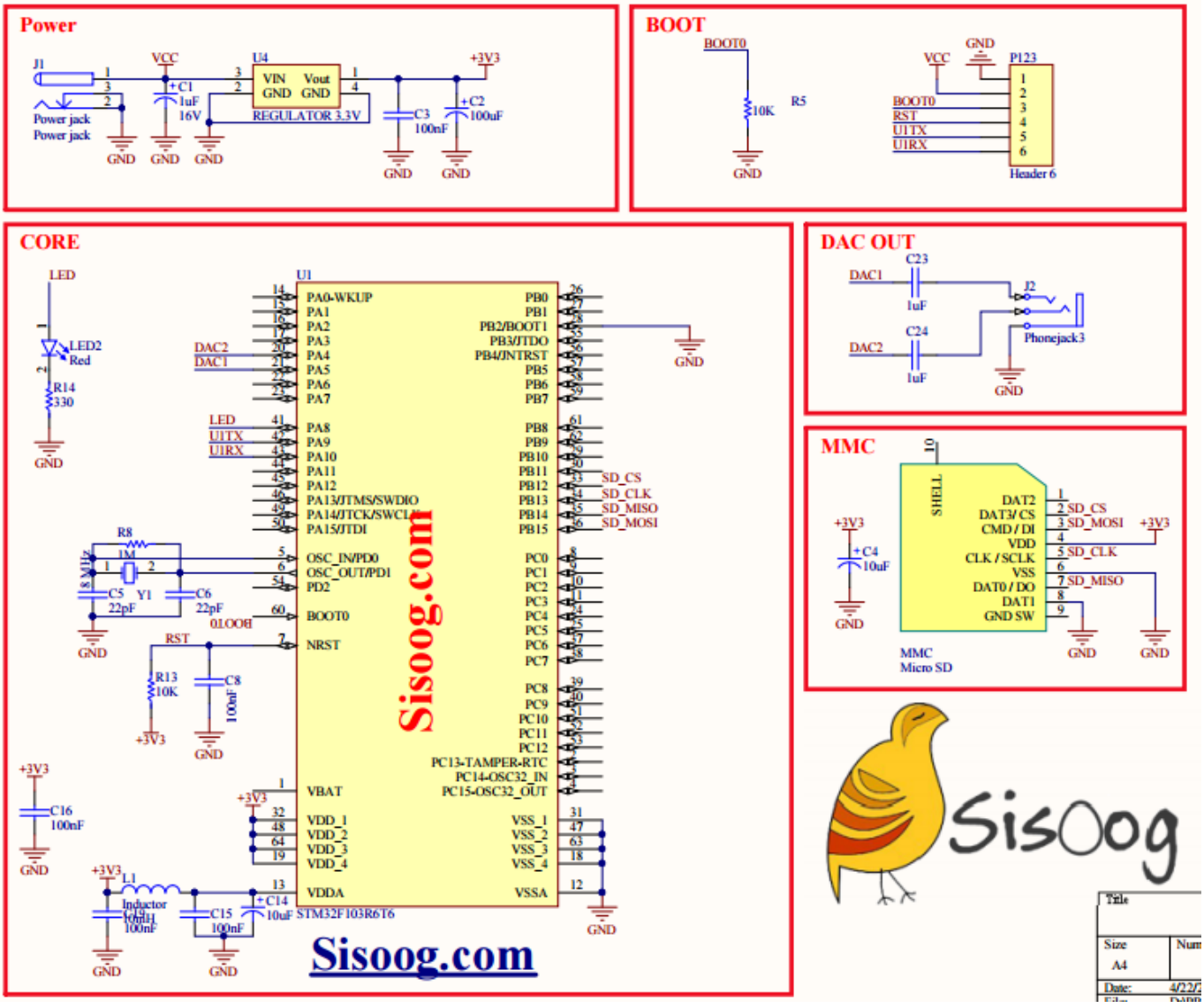
برای رمز گشایی داده های موجود در این قسمت اول لازم است با الگوریتم هافمن آشنایی داشته باشید. در واقع ذوش گدگذاری هافمن یک روش فشرده سازی داده است که توسط دیود هافمن پایه گذاری شد. در روش هافمن داده ها 20 الی 90 درصد فشرده می شوند. در الگوریتم هافمن از ساختمان داده صف اولویت استفاده می شود که توضیح تحوه عملکرد الگوریتم هافمن در این مقاله نمی گنجد و علاقه مندان جهت یادگیری میتوانند به وبسایت های مربوطه مراجعه کنند.

برای دکدینگ این قسمت همانطور که در عکس مشخص شده است باید از درخت هافمن استفاده کنید.



توضیح سخت افزار

سخت افزار مورد استفاده جهت دکد فایلMP3 ، همانطور که در تصویر زیر مشخص شده یک سخت افزار بسیار ساده است که جزء اصلی آن را یک میکروکنترلر STM32F103RET تشکیل داده است و تمام اتفاقات درون میکروکنترلر می افتد.


Power
J1
Power jack
Power jack
VCC
+C1
1uF
16V
U4
VIN Vout
GND GND
REGULATOR 3.3V
C3
100nF
+C2
100uF
+3V3
GND
BOOT
BOOT0
R5
10K
VCC
GND
P123
BOOT0
RST
U1TX
U1RX
Header 6
CORE
U1
LED
LED2
Red
R14
330
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13/JTMS/SWDIO
PA14/JTCK/SWCLK
PA15/JTDI
PB0
PB1
PB2/BOOT1
PB3/JTDO
PB4/JNTRST
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
SD_CS
SD_CLK
SD_MISO
SD_MOSI
DAC2
DAC1
LED
U1TX
U1RX
R8
1M
Y1
C5
22pF
C6
22pF
OSC_IN/PD0
OSC_OUT/PD1
PD2
BOOT0
NRST
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
PC11
PC12
PC13-TAMPER-RTC
PC14-OSC32_IN
PC15-OSC32_OUT
VBAT
VDD_1
VDD_2
VDD_3
VDD_4
VSS_1
VSS_2
VSS_3
VSS_4
VDDA
VSSA
RST
R13
10K
C8
100nF
C16
100nF
L1
Inductor
C15
100nF
+C14
10uF
STM32F103R6T6
Sisoog.com
DAC OUT
C23
1uF
C24
1uF
DAC1
DAC2
J2
Phonejack3
MMC
SHELL
DAT2
DAT3/ CS
CMD / DI
VDD
CLK / SCLK
VSS
DAT0 / DO
DAT1
GND SW
+C4
10uF
MMC
Micro SD
Size
A4
Date:

برای تبدیل داده های دیجیتال از مبدل های دیجیتال به آنالوگ (DAC) داخلی میکروکنترلر استفاده شده است که قادر است نرخ یک مگاسمپل بر ثانیه و دقت 12 بیت را ارائه نماید. کارت حافظه هم با استفاده از رابط SPI به میکروکنترلر وصل شده است. در واقع سعی شده تا جای ممکن مدار ساده و قابل فهم باشد.

نرم افزار

نرم افزار توسط کامپایلر کیل نوشته شده (علارغم میل باطنی) و قادر است فایل های MP3 تا بیت ریت 320 کیلو و فایلهای Wav را تا بالاترین بیت ریت ممکن پخش نماید. برای سرعت عمل بیشتر در دکد فایل های Mp3 بخشی از کد به زبان اسمبلی نوشته شده است. تا حد ممکن سعی شده است که منابع موجود کمال استفاده شود، برای انتقال داده های صوتی به مبدل های DAC از واحد DMA استفاده شده است تا کمترین درگیری CPU در این خصوص وجود داشته باشد.

مطابق برنامه فوق ، برنامه بعد از نصب و پیکر بندی کارت حافظه ، ابتدا به دنبال فایل start.mp3 درون کارت حافظه می گردد ، در صورت وجود چنین فایلی آن را اجرا کرده و بعد از اتمام ، به دنبال فایل start.wav می گردد تا آن را پخش نماید.

شما به راحتی با اضافه کردن توابع و بخش های مورد نظر خود میتوانید این کد را تبدیل به یک Mp3 Player همراه کنید.

```c
int  main(void){

    RCC_Configuration();
    GPIO_Configuration();
    NVIC_Configuration();
    buffer_Init();                          /* Uart init RX / TX buffers
  USART1_Init();

    while(disk_initialize(0)==STA_NOINIT)       /*Init SdCard Disk*/
    {
        SET_LED();
    }RESET_LED();

    while(f_mount(0,&fs)!=RES_OK)                   /*Mount Fat File System*/
    {
        SET_LED();
    }RESET_LED();


    printf("Zeus@sisoog.com\r\n");
    MP3_PlayAudioFile("start.mp3"); /*Play Mp3 File*/
    play_wave("start.wav");                 /*Play Wave File*/


    while(1)
    {

    }

}
```

دانلود سورس کد دیکدر MP3