[MPEG2/TS 包结构简述](http://blog.csdn.net/myaccella/article/details/6981522)

分类： [流媒体编程](http://blog.csdn.net/myaccella/article/category/936905)2011-11-17 16:01 2722人阅读 [评论](http://blog.csdn.net/myaccella/article/details/6981522#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/myaccella/article/details/6981522#report)

[byte](http://www.csdn.net/tag/byte)[加密](http://www.csdn.net/tag/%e5%8a%a0%e5%af%86)

**TS 包由以下3部分组成：**

包头： 4字节， 同步，识别，检错和加密用

自适配域：可选

包数据：最大184字节（不存在自适配域）

|  |  |  |
| --- | --- | --- |
| **4字节** | 184字节 | |
| 包头 | 自适配域 | 包数据 |

**4字节包头结构：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1字节 | 1比特 | 1比特 | 1比特 | 13比特 | 2比特 | 2比特 | 4比特 |
| 同步字节 | 传输错误描述符 | 载荷起始描述符 | 传输优先 | 包识别 | 传输加扰控制 | 适配域控制 | 连续计数器 |

**自适配域结构包含：**

域长度：1字节

域标志：1字节， PCR标志就是其中一个重要标志

域信息： 由域长度指定

**包数据的类型一般有如下几种：**

PSI信息:  PAT/PMT/CAT/NIT表

视频流

音频流

字幕流

除了PSI信息，视频流、音频流和字幕流会做PES打包。

**PAT表结构**

完整的PAT表包含一个头结构和若干个节目结构，

最后还要加上CRC校验。

**PAT表头结构(总共8字节)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1Byte | 1bit | 1bit | 2bit | 12bit | 2Byte | 2bit | 5bit | 1bit | 1Byte | 1Byte |
| 表标识 | 段语法标识符 | 填0 | 保留 | 段长度含CRC | 传输流标识 | 保留 | 版本号 | 有效标识 | 段号 | 最后段号 |

**PAT表节目结构**

|  |  |  |
| --- | --- | --- |
| **2Byte** | 3bit | 13bit |
| 节目号 | 保留 | 节目映射表PID |

**PMT表结构**

一个完整的PMT表包含一个表头和若干个流结构，

最后还要加上CRC校验。

**PMT表流结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1Byte** | 3bit | 13bit | 4bit | 12bit |
| 流类型 | 保留 | 基本元素PID | 保留 | 节目元素长度 |

**PMT头结构(总共12字节)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1Byte | 1bit | 1bit | 2bit | 12bit | 2Byte | 2bit | 5bit | 1bit | 1Byte | 1Byte | 3bit | 13bit | 4bit | 12bit |
| 表标识 | 段语法标识符 | 填0 | 保留 | 段长度含CRC | 节目数 | 保留 | 版本 | 有效标识 | 段号 | 最后段号 | 保留 | PCR PID | 保留 | 节目信息长度 |

一个简单的从指定MPEG2/TS文件中获取PMT PID的函数，

注意代码中的注释。

#define MAKE\_PID(p) (((p[0]<<8) + p[1]) & 0x1fff)

**[html]** [view plaincopy](http://blog.csdn.net/myaccella/article/details/6982596)

1. ushort GetPmtPid(const char\* filename)
2. {
3. #define TSPACKET\_SIZE     188
4. ushort pid;
5. uchar buf[TSPACKET\_SIZE];
6. uchar\* p = buf;
7. FILE\*  fp;
9. fp = fopen(filename, "rb");
10. if (fp == NULL) {
11. fprintf(stderr, "Failed to open the file %s: %s\n", filename, strerror(errno));
12. return 0;
13. }
15. while (fread(buf, 1, TSPACKET\_SIZE, fp) == TSPACKET\_SIZE) {
16. if (buf[0] != MPEG\_TS\_SYNC\_BYTE) {
17. fprintf(stderr, "No sync byte where one was expected!\n");
18. break;
19. }
21. p = buf;
22. pid = MAKE\_PID((p+1));
23. if ( pid != 0 ) {
24. // 不是PAT包
25. continue;
26. }
28. // Skip the TS header
29. p += MPEG\_TS\_HEADER\_SIZE; //@MPEG\_TS\_HEADER\_SIZE = 4
31. // Adaptation field exist or not
32. if ((buf[3] **>>** 4 & 0x03) & 0x2) {
33. fprintf(stderr, "Adaptation field exist\n");
34. p += 1 + p[0];  // p[0]是Adaptation field长度, 1为长度域
35. }
37. // p[0] 为 Point field长度， 1为Point field
38. p += 1 + p[0];
40. // 8 为PAT表头长度，如果定义了表头，则用sizeof运算符
41. p += 8;
43. // NIT Program exist??
44. if( \*((ushort\*)p) == 0) {
45. p += 4; // PAT的节目结构为4字节
46. }
48. p += 2; // 节目号为两个字节
50. fclose(fp);
51. return MAKE\_PID(p);
52. }
54. fclose(fp);
55. return 0;
56. }

[MPEG2/TS获取PCR PID方法](http://blog.csdn.net/myaccella/article/details/6982762)

分类： [流媒体编程](http://blog.csdn.net/myaccella/article/category/936905)2011-11-17 17:10 3776人阅读 [评论](http://blog.csdn.net/myaccella/article/details/6982762#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/myaccella/article/details/6982762#report)

[header](http://www.csdn.net/tag/header)[byte](http://www.csdn.net/tag/byte)[fp](http://www.csdn.net/tag/fp)[null](http://www.csdn.net/tag/null)[file](http://www.csdn.net/tag/file)

一个简单的从指定MPEG2/TS文件中获取PCR PID的函数，

PMT的头结构中有PCR PID,我们首先得到PMT PID，然后分析PMT包，就可以得到PCR PID了。

GetPmtPid函数的实现见：<http://blog.csdn.net/myaccella/article/details/6982596>

MAKE\_PID:

#define MAKE\_PID(p) (((p[0]<<8) + p[1]) & 0x1fff)

**[html]** [view plaincopy](http://blog.csdn.net/myaccella/article/details/6982762)

1. ushort GetPcrPid(const char\* filename)
2. {
3. #define TSPACKET\_SIZE     188
4. ushort pid, pmtpid, pcrpid = 0;
5. uchar buf[TSPACKET\_SIZE];
6. uchar\* p = buf;
7. FILE\*  fp;
9. pmtpid = GetPmtPid(filename);
10. if (pmtpid == 0) {
11. fprintf(stderr, "Unable to get PMT PID!\n");
12. return 0;
13. }
15. fp = fopen(filename, "rb");
16. if (fp == NULL) {
17. fprintf(stderr, "Failed to open the file %s: %s\n", filename, strerror(errno));
18. return 0;
19. }

22. while (fread(buf, 1, TSPACKET\_SIZE, fp) == TSPACKET\_SIZE) {
23. if (buf[0] != MPEG\_TS\_SYNC\_BYTE) { // @MPEG\_TS\_SYNC\_BYTE = 0x47
24. fprintf(stderr, "No sync byte!\n");
25. break;
26. }
28. p = buf;
29. pid = MAKE\_PID((p+1));
30. if( pid != pmtpid ) {
31. // Not a  PMT packet
32. continue;
33. }
35. // payload\_unit\_start\_indicator 必须为1
36. if (!(buf[1] **>>** 6 & 0x01)) {
37. fprintf(stderr, "No payload\_unit\_start\_indicator!\n");
38. continue;
39. }
41. // Skip the TS header
42. p += MPEG\_TS\_HEADER\_SIZE; //@MPEG\_TS\_HEADER\_SIZE = 4
44. // Adaptation field exist or not
45. if ((buf[3] **>>** 4 & 0x03) & 0x2) {
46. fprintf(stderr, "Adaptation field exist\n");
47. p += 1 + p[0];  // p[0]是Adaptation field长度, 1为长度域
48. }
50. // p[0] 为 Point field长度， 1为Point field
51. p += 1 + p[0];
53. // 8 为PCR PID在PMT表头中的偏移
54. p += 8;
56. // 取到的PCR PID如果是0x1fff, 则需要继续取
57. pcrpid = MAKE\_PID(p);
58. if (pcrpid == 0x1fff) {
59. pcrpid = 0;
60. continue;
61. }
63. break;
64. }
65. fclose(fp);
66. return pcrpid;
67. }
69. [MPEG2/TS 获取Video PID](http://blog.csdn.net/myaccella/article/details/6983015)
70. 分类： [流媒体编程](http://blog.csdn.net/myaccella/article/category/936905)2011-11-17 18:00 1484人阅读 [评论](http://blog.csdn.net/myaccella/article/details/6983015#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/myaccella/article/details/6983015#report)
71. [video](http://www.csdn.net/tag/video)[stream](http://www.csdn.net/tag/stream)[header](http://www.csdn.net/tag/header)[byte](http://www.csdn.net/tag/byte)[struct](http://www.csdn.net/tag/struct)[fp](http://www.csdn.net/tag/fp)
72. 从PMT表中获取Video PID的函数代码
74. GetPmtPid函数的实现见：<http://blog.csdn.net/myaccella/article/details/6982596>
75. #define MAKE\_PID(p) (((p[0]<<8) + p[1]) & 0x1fff)  
    #define MAKE\_WORD(h, l) (((h) << 8) | (l))
77. Little endian 的结构定义
78. typedef struct {  
      unsigned stream\_type  :8;
79. unsigned elementary\_PID\_high :5;  
      unsigned    :3;
80. unsigned elementary\_PID\_low :8;
81. unsigned ES\_info\_length\_high :4;  
      unsigned    :4;
82. unsigned ES\_info\_length\_low :8;  
    }MpegPmtStream\_t;
84. typedef struct {  
      unsigned table\_id   :8;
85. unsigned section\_length\_high  :4;  
      unsigned     :2;  
      unsigned    :1;  
      unsigned section\_syntax\_indicator :1;  
        
      unsigned section\_length\_low  :8;  
        
      unsigned program\_number\_high  :8;
86. unsigned program\_number\_low  :8;
87. unsigned current\_next\_indicator :1;  
      unsigned version\_number  :5;  
      unsigned     :2;
88. unsigned section\_number  :8;
89. unsigned last\_section\_number  :8;  
        
      unsigned PCR\_PID\_high   :5;  
      unsigned    :3;
90. unsigned PCR\_PID\_low   :8;  
        
      unsigned program\_info\_length\_high :4;  
      unsigned    :4;  
        
      unsigned program\_info\_length\_low :8;  
    }MpegPmtHeader\_t;

**[html]** [view plaincopy](http://blog.csdn.net/myaccella/article/details/6983015)

1. ushort GetVideoPid(const char\* filename)
2. {
3. #define TSPACKET\_SIZE     188
4. #define CRC\_SIZE          4
5. ushort pid, pmtpid, videopid = 0;
6. uchar buf[TSPACKET\_SIZE];
7. uchar\* p = buf;
8. MpegPmtHeader\_t\* header;
9. MpegPmtStream\_t\* stream;
10. FILE\*  fp;
11. int pos = 0;
12. ushort section\_len;
14. pmtpid = GetPmtPid(filename);
15. if (pmtpid == 0) {
16. fprintf(stderr, "Unable to get PMT PID!\n");
17. return 0;
18. }
20. fp = fopen(filename, "rb");
21. if (fp == NULL) {
22. fprintf(stderr, "Failed to open the file %s: %s\n", filename, strerror(errno));
23. return 0;
24. }

27. while (fread(buf, 1, TSPACKET\_SIZE, fp) == TSPACKET\_SIZE) {
28. if (buf[0] != MPEG\_TS\_SYNC\_BYTE) { // @MPEG\_TS\_SYNC\_BYTE = 0x47
29. fprintf(stderr, "No sync byte!\n");
30. break;
31. }
33. p = buf;
34. pid = MAKE\_PID((p+1));
35. if( pid != pmtpid ) {
36. // Not a  PMT packet
37. continue;
38. }
40. // payload\_unit\_start\_indicator 必须为1
41. if (!(buf[1] **>>** 6 & 0x01)) {
42. fprintf(stderr, "No payload\_unit\_start\_indicator!\n");
43. continue;
44. }
46. // Skip the TS header
47. p += MPEG\_TS\_HEADER\_SIZE; //@MPEG\_TS\_HEADER\_SIZE = 4
49. // Adaptation field exist or not
50. if ((buf[3] **>>** 4 & 0x03) & 0x2) {
51. fprintf(stderr, "Adaptation field exist\n");
52. p += 1 + p[0];  // p[0]是Adaptation field长度, 1为长度域
53. }
55. // p[0] 为 Point field长度， 1为Point field
56. p += 1 + p[0];
58. // Now p 指向 PMT表头
59. header = (MpegPmtHeader\_t\*)p;
60. pos +=  sizeof(MpegPmtHeader\_t);
61. pos += MAKE\_WORD(header-**>**program\_info\_length\_high, header-**>**program\_info\_length\_low);;
63. // Parse stream table
64. section\_len = MAKE\_WORD(header-**>**section\_length\_high, header-**>**section\_length\_low);
65. while (pos **<** (section\_len + 2) - CRC\_SIZE) {
66. stream = (MpegPmtStream\_t\*)(p + pos);
67. // @MPEG\_TS\_STREAM\_TYPE\_VIDEO\_2 = 0x02
68. if (stream-**>**stream\_type == MPEG\_TS\_STREAM\_TYPE\_VIDEO\_2) {
69. videopid = MAKE\_WORD(stream-**>**elementary\_PID\_high, stream-**>**elementary\_PID\_low);
70. break;
71. }
72. // @MPEG\_PMT\_STREAM\_HEADER\_SIZE = 5
73. pos +=  MPEG\_PMT\_STREAM\_HEADER\_SIZE
74. + MAKE\_WORD(stream-**>**ES\_info\_length\_high, stream-**>**ES\_info\_length\_low);
75. }
76. }
78. fclose(fp);
79. return videopid;
80. }

http://blog.csdn.net/myaccella/article/details/6976199