

原

中文字符点阵信息的显示和插入新字符（基于HZK16 ASC16软字库）

2015年03月24日 20:28:40

尼奥普兰

阅读数：14941

标签：

c语言

源码

更多

本系统是对中文字库HZK16和ASC16字库进行操作

HZK16字库中每一个中文字符使用的是32字节的点阵信息，ASC16字库是16字节的点阵信息打印

本文实现了中文汉字的点阵信息打印和字符的操作，包括插入未知字符（囧）（需要借助软件PCtoLCD.exe实现点阵信息的提取）

```
1  /*****
2  ** this C source code is made for HZK16 and ASC16 characters system **
3  **
4  **      newplan 2013.9  in UESTC      **
5  *****/
6  #include <stdio.h>
7  #include <stdlib.h>
8  #include <string.h>
9  #define FONTSIZE 32
10
11 //typedef unsigned int  WORD;
12 //typedef unsigned char  BYTE;
13
14 /*****/
15 int Get_Asc_Code(unsigned char *Get_Input_Char, char buff[]);
16 int Get_HzK_Code(unsigned char *Get_Input_Char, char buff[]);
17 void Print_Asc_Char(char *mat, char *Out_Put_1, char *Out_Put_2);
18 void Print_HzK_Char(char *mat, char *Out_Put_1, char *Out_Put_2);
19 unsigned long Judge_type_char(unsigned char *Get_Input_Char, unsigned long *offset, int *length);
20 int Delete_Char_From_Lib(char *character, unsigned long offset, char *Lib_Name);
21 int Get_Char_Model(unsigned char buff_input[FONTSIZE * FONTSIZE / 8]);
22 int Not_In_Lib(char buff[FONTSIZE * FONTSIZE / 8]);
23 int transpose(char chanted[FONTSIZE][FONTSIZE]);
24 void distranspose(char mat[][FONTSIZE / 8], char **array);
25 int asistant_insert(unsigned char *mat);
26 void change(char mat[][FONTSIZE / 8], char **array);
27 /*****/
28
29 int output_system(void);
30 int insert_system(void);
31 int delete_system(void);
32
33
34
35
36 /*****
37 // Method:  Get_Asc_Code
38 // FullName:  Get_Asc_Code
39 // Access:  public
40 // Returns:  int
41 // Qualifier: 得到英文字符的字模信息，存入数组
42 // Parameter: unsigned char * Get_Input_Char 要得到字模信息的字符指针
43 // Parameter: char buff[] 存储得到字模信息的数组
44 *****/
45 int Get_Asc_Code(unsigned char *Get_Input_Char, char buff[])
46 {
47     unsigned long offset;
48     FILE *ASC;
49     /*打开字库文件asc16*/
50     if ((ASC = fopen("ASC16", "rb+")) == NULL)
51     {
52         printf("Can't open asc,Please add it!");
53         system("pause");
54         exit(0);
```

```

55     } 56 |         offset = *(Get_Input_Char) * 16 + 1;          /*通过ascii码算出偏移量*/
57     fseek(ASC, offset, SEEK_SET);          /*将文件指针移动到偏移量的位置*/
58     fread(buff, 16, 1, ASC);              /*从偏移量的位置读取32个字节*/
59     printf("ASCII:%d,offset:%d\n\r", *Get_Input_Char, offset);
60     fclose(ASC);
61     return 1;
62 }
63
64 //*****
65 // Method: Print_Asc_Char
66 // FullName: Print_Asc_Char
67 // Access: public
68 // Returns: void
69 // Qualifier: 根据字模信息输出英文字符
70 // Parameter: char * mat 字模指针
71 // Parameter: char * Out_Put_1 字模中为1的点显示的字符，也就是前景字符
72 // Parameter: char * Out_Put_2 字模中为0的点显示的字符，也就是背景字符
73 //*****
74 void Print_Asc_Char(char *mat, char *Out_Put_1, char *Out_Put_2)
75 {
76     int i, j;
77     for (i = 0; i < 16; i++)              /* 8x16的点阵，一共有16行*/
78     {
79         for (j = 0; j < 8; j++)            /*横向一个字8位，依次判断每位是否为0*/
80             if (mat[i] & (0x80 >> j))      /*测试当前位是否为1*/
81                 printf("%s", Out_Put_1);    /*为1的显示为字符c1*/
82             else printf("%s", Out_Put_2);    /*为0的显示为字符c2*/
83         printf("\n");                      /*输完一行以后，进行换行*/
84     }
85 }
86
87 //*****
88 // Method: Get_HzK_Code                      ***
89 // FullName: Get_HzK_Code                    ***
90 // Access: public                            ***
91 // Returns: int                             ***
92 // Qualifier: 得到汉字符的字模信息，存入数组 ***
93 // Parameter: unsigned char * Get_Input_Char 要得到字模信息的字符指针 ***
94 // Parameter: char buff[] 存储字模信息的数组 ***
95 //*****
96 int Get_HzK_Code(unsigned char *Get_Input_Char, char buff[])
97 {
98     int not_find = 0;
99     unsigned char qh, wh;
100     unsigned long offset;
101     FILE *HZK;
102     char file_name[] = "HZK32";
103     if ((HZK = fopen(file_name, "rb+")) == NULL) /*打开字库文件hzk16*/
104     {
105         printf("Can't open %s,Please add it?\n", file_name);
106         system("pause");
107         exit(0);
108     }
109     /*区码=内码(高字节)-160 位码=内码(低字节)-160*/
110     qh = *(Get_Input_Char) - 0xa0;          /*10进制的160等于16进制的A0*/
111     wh = *(Get_Input_Char + 1) - 0xa0;      /*获得区码与位码*/
112     offset = (94 * (qh - 1) + (wh - 1)) * FONTSIZE * FONTSIZE / 8; /*计算该汉字在字库中偏移量*/
113     not_find = fseek(HZK, offset, SEEK_SET); /*将文件指针移动到偏移量的位置*/
114     if (not_find)
115     {
116         printf("未查到该区！error！！\n");
117         fclose(HZK);
118         return 0;
119     }
120     fread(buff, FONTSIZE * FONTSIZE / 8, 1, HZK); /*从偏移量的位置读取32个字节*/
121     if (Not_In_Lib(buff))
122     {
123         fclose(HZK);
124         printf("有未识别字符!\n");
125         system("pause");

```



1



```
126         return 0;127    }
128     printf("qh:%d,wh:%d,offset:%ld\n\r", qh, wh, offset);
129     fclose(HZK);
130     return 1;
131 }
132
133 //*****
134 // Method:  Print_HzK_Char          ***
135 // FullName:  Print_HzK_Char        ***
136 // Access:  public                  ***
137 // Returns:  void                    ***
138 // Qualifier: 根据字模信息输入汉字字符          ***
139 // Parameter: char *mat 字模指针          ***
140 // Parameter: char *Out_Put_1 字模中为1的点显示的字符，也就是前景字符 ***
141 // Parameter: char *Out_Put_2 字模中为0的点显示的字符，也就是背景字符 ***
142 //*****
143 void Print_HzK_Char2(char *mat, char *Out_Put_1, char *Out_Put_2)
144 {
145     int i, j, k;
146     for (i = 0; i < FONTSIZE; i++)          /*16x16点阵汉字，一共有16行*/
147     {
148         for (j = 0; j < FONTSIZE / 8; j++)          /*横向有2个字节，循环判断每个字节的*/
149             for (k = 0; k < 8; k++)          /*每个字节有8位，循环判断每位是否为1*/
150                 if (mat[i * FONTSIZE / 8 + j] & (0x80 >> k)) /*测试当前位是否为1*/
151                     printf("%s", Out_Put_1);          /*为1的显示为字符c1*/
152                 else printf("%s", Out_Put_2);          /*为0的显示为字符c2*/
153             printf("\n");          /*输完一行以后，进行换行*/
154     }
155 }
156
157 void Print_HzK_Char(char *mat, char *Out_Put_1, char *Out_Put_2)
158 {
159     int i, j, k;
160     for (j = 0; j < FONTSIZE / 8; j++)          /*横向有2个字节，循环判断每个字节的*/
161         for (k = 0; k < 8; k++)          /*每个字节有8位，循环判断每位是否为1*/
162         {
163             for (i = 0; i < FONTSIZE; i++)          /*16x16点阵汉字，一共有16行*/
164             {
165                 if (mat[i * FONTSIZE / 8 + j] & (0x80 >> k)) /*测试当前位是否为1*/
166                     printf("%s", Out_Put_1);          /*为1的显示为字符c1*/
167                 else printf("%s", Out_Put_2);          /*为0的显示为字符c2*/
168             }
169             printf("\n");          /*输完一行以后，进行换行*/
170         }
171     }
172 }
173
174 //*****
175 // Method:  Not_In_Lib          ***
176 // FullName:  Not_In_Lib        ***
177 // Access:  public              ***
178 // Returns:  int                ***
179 // Qualifier: /*判断字符是不是在字库里 ***
180 // Parameter: char buff[]          ***
181 //*****
182 int Not_In_Lib(char buff[FONTSIZE * FONTSIZE / 8])
183 {
184     for (int i = 0; i < FONTSIZE * FONTSIZE / 8; i++)
185     {
186         if (buff[i])          /*如果有一个不为0，表明buff【】已经被修改过，字库存在此字退出此函数
187             return 0;
188         }
189     }
190     return 1;
191 }
192
193 //*****
194 // Method:  Select_type_char          ***
195 // FullName:  Select_type_char        ***
196 // Access:  public                    ***
197 // Returns:  int 0 表示成功;          ***
198 // Qualifier: 判断是中文字符还是英文单字节字符，并算出偏移量 ***
```



1



```

198 // Parameter: unsigned char * Get_Input_Char          ***
199 // Parameter: int * offset 偏移量                      ***
200 // Parameter: int * length 缓冲区的长度                ***
201 //*****
202 unsigned long Judge_type_char(unsigned char *Get_Input_Char, unsigned long *offset, int *length)
203 {
204     if (Get_Input_Char[2] == 0 && Get_Input_Char[1] == 0) //asc16
205     {
206         *offset = *(Get_Input_Char) * 16 + 1;
207         *length = 16;
208     }
209     else //HZK16
210     {
211         unsigned char qh = (unsigned char)Get_Input_Char[0] - 0xa0; /*10进制的160等于16进制的
212         unsigned char wh = (unsigned char)Get_Input_Char[1] - 0xa0; /*获得区码与位码*/
213         *offset = (94 * (qh - 1) + (wh - 1)) * 32L; /*计算该汉字在字库中偏移量*/
214         *length = 32;
215     }
216     return *offset;
217 }
218
219 //*****
220 // Method: Get_Char_Model          ***
221 // FullName: Get_Char_Model        ***
222 // Access: public                  ***
223 // Returns: int 0 表示字模读取失败 1表示读取成功      ***
224 // Qualifier: 区汉字字模          ***
225 // Parameter: unsigned char * character 指向一个要加入字库的字符
226 // Parameter: char buff_input 指向字模的指针          ***
227 //*****
228 int Get_Char_Model(unsigned char buff_input[FONTSIZE * FONTSIZE / 8])
229 {
230     char st[] = "C:/experience/PCtoLCD/PCtoLCD2002.exe"; // 已给定命令内容的字符变量
231     char file_read_buffer[2000];
232     FILE * model_read_stream;
233     int number = 0, current = 0, decade = 0, bits = 0;
234     memset(file_read_buffer, 0, sizeof(file_read_buffer));
235     printf("保存的字模文件请务必以 : model.TXT文件名保存\n");
236     printf("读取汉字字模应用程序即将打开!\n");
237
238     system("pause");
239     system(st); // 运行文件PCtoLCD.exe
240     model_read_stream = fopen("C:/experience/PCtoLCD/lib/model.TXT", "r");
241     if (!model_read_stream)
242     {
243         printf("文件打开失败!\n");
244         return 0;
245     }
246     fread(file_read_buffer, sizeof(file_read_buffer), sizeof(char), model_read_stream);
247     system("pause");
248     printf("\n%s\n", file_read_buffer);
249     fclose(model_read_stream);
250     // system("del C:/experience/PCtoLCD/lib/model.TXT");
251     // system("del C:/experience/PCtoLCD/lib/model.TXT_index.TXT");
252     printf("wait for a minute!\n");
253     system("pause");
254     for (current = 0; current < sizeof(file_read_buffer); current++)
255     {
256         if (file_read_buffer[current] == 'H')
257         {
258             decade = file_read_buffer[current - 2] < 'A' ?
259             file_read_buffer[current - 2] - '0' : file_read_buffer[current - 2] - 'A' + 10;
260             bits = file_read_buffer[current - 1] < 'A' ?
261             file_read_buffer[current - 1] - '0' : file_read_buffer[current - 1] - 'A' + 10;
262             buff_input[number] = (unsigned char)(decade * FONTSIZE + bits);
263             number++;
264         }
265         if (number >= FONTSIZE * FONTSIZE / 8)
266             return 1;
267     }
268     return 0;

```



1



```
269 }
270 |
271 //*****
272 // Method: Delete_Char_From_Lib ***
273 // FullName: Delete_Char_From_Lib ***
274 // Access: public ***
275 // Returns: int ***
276 // Qualifier: 删除字库中的某个文字 ***
277 // Parameter: unsigned char * character 要删除的字符 ***
278 // Parameter: int offset 删除字符的偏移量 ***
279 // Parameter: char * Lib_Name 删除字符的库 ***
280 //*****
281 int Delete_Char_From_Lib(char *character, unsigned long offset, char *Lib_Name)
282 {
283     char buff_clear_chinese[32], buff_clear_english[16];
284     memset(buff_clear_chinese, 0, sizeof(buff_clear_chinese));
285     memset(buff_clear_english, 0, sizeof(buff_clear_english));
286     FILE *open_file = fopen(Lib_Name, "rb+");
287     fseek(open_file, offset, SEEK_SET);
288     if (!open_file)
289     {
290         printf("can't open file: %s\n", Lib_Name);
291         exit(0);
292     }
293     if (character[2] == 0 && character[1] == 0)
294     {
295         fwrite(buff_clear_english, 16, 1, open_file);
296     }
297     else
298     {
299         system("pause");
300         fwrite(buff_clear_chinese, 32, 1, open_file);
301     }
302     printf("delete successfully!\n");
303     fclose(open_file);
304     system("pause");
305     return 0;
306 }
307
308 //*****
309 // Method: transpose ***
310 // FullName: transpose ***
311 // Access: public ***
312 // Returns: int ***
313 // Qualifier: 转置函数 ***
314 //因为字模软件读取的汉字按照正常运算是被转置的 ***
315 // Parameter: char changed[16][16] ***
316 //*****
317 int transpose(char changed[FONTSIZE][FONTSIZE])
318 {
319     char temp;
320     int i = 0, j = 0;
321     for (i; i < FONTSIZE; i++)
322     {
323         for (j = i; j < FONTSIZE; j++)
324         {
325             temp = changed[i][j];
326             changed[i][j] = changed[j][i];
327             changed[j][i] = temp;
328         }
329     }
330     return 1;
331 }
332
333 //*****
334 // Method: distranspose
335 // FullName: distranspose
336 // Access: public
337 // Returns: void
338 // Qualifier: /*把二进制的mat数组转变成字符change数组*/
339 // Parameter: char mat[][2]
```



1



```
340 // Parameter: char ** array341 | // Parameter: int m
342 // Parameter: int n
343 //*****
344 void distranspose(char mat[][FONTSIZE / 8], char **array)
345 {
346     int i, j, k, l = 0, n = FONTSIZE;
347     for (i = 0; i < FONTSIZE; i++)
348     {
349         for (j = 0; j < FONTSIZE / 8; j++)
350         {
351             for (k = 0; k < 8; k++)
352             {
353                 if (*(char*)array + n * i + l == '#') //根据每个字节的0和1的状态计算出可用的字模
354                 {
355                     mat[i][j] |= (0x80 >> k);
356                 }
357                 else
358                 {
359                     mat[i][j] &= ~(0x80 >> k);
360                 }
361                 l++;
362             }
363         }
364         l = 0;
365     }
366     return ;
367 }
368
369 //*****
370 // Method: change
371 // FullName: change
372 // Access: public
373 // Returns: void
374 // Qualifier: /*把二进制的mat数组转变成字符change数组*/
375 // Parameter: char mat[][2]
376 // Parameter: char ** array
377 // Parameter: int m
378 // Parameter: int n
379 //*****
380 void change(char mat[][FONTSIZE / 8], char **array)
381 {
382     int i, j, k, l = 0, n = FONTSIZE;
383     for (i = 0; i < FONTSIZE; i++)
384     {
385         for (j = 0; j < FONTSIZE / 8; j++)
386         {
387             for (k = 0; k < 8; k++)
388             {
389                 if (mat[i][j] & (0x80 >> k)) //提出每个字节的0和1的状态
390                 {
391                     *(char*)array + n * i + l = '#';
392                     l++;
393                 }
394                 else
395                 {
396                     *(char*)array + n * i + l = '-';
397                     l++;
398                 }
399             }
400         }
401         l = 0;
402     }
403
404     return ;
405 }
406
407
408 //*****
409 // Method: assistant_insert
410 // FullName: assistant_insert
411 // Access: public
```



1



```

412 // Returns: int
413 // Qualifier: 辅助插入字符函数
414 // Parameter: unsigned char * mat
415 //*****
416 int asistant_insert(unsigned char *mat)
417 {
418     int j = 0, i = 0;
419     char array[ FONTSIZE ][ FONTSIZE ], mat_temp[ FONTSIZE ][ FONTSIZE / 8 ];
420
421     memset(array, 0, sizeof(array));
422     memset(mat_temp, 0, sizeof(mat_temp));
423
424     for (j = 0, i = 0; j < FONTSIZE; j++)
425     {
426         mat_temp[j][0] = mat[i];
427         mat_temp[j][1] = mat[i + 1];
428         mat_temp[j][2] = mat[i + 2];
429         mat_temp[j][3] = mat[i + 3];
430         i = i + FONTSIZE / 8;
431     }
432     change(mat_temp, (char **)array);
433     transpose(array);
434     distranspose(mat_temp, (char **)array);
435     for (j = 0, i = 0; j < FONTSIZE; j++)
436     {
437         mat[i] = mat_temp[j][0];
438         mat[i + 1] = mat_temp[j][1];
439         mat[i + 2] = mat_temp[j][2];
440         mat[i + 3] = mat_temp[j][3];
441         i = i + FONTSIZE / 8;
442     }
443     return 0;
444 }
445
446
447
448 //*****
449 // Method: out_put_system
450 // FullName: out_put_system
451 // Access: public
452 // Returns: int
453 // Qualifier: 显示输出系统 ( 包括中英文字符 )
454 // Parameter: void
455 //*****
456 int output_system(void)
457 {
458     int count = 0;
459     char Buffer_English[16], Buffer_Chinese[ FONTSIZE * FONTSIZE / 8 ];
460     unsigned char word[3] = {0};
461     char *Output_String1 = (char *)"●", *Output_String2 = (char *)"o";
462     while (1)
463     {
464         memset(Buffer_Chinese, 0, sizeof(Buffer_Chinese));
465         memset(Buffer_English, 0, sizeof(Buffer_English));
466         printf("输入要生成字模的汉字(可以多个输入)[Q表示退出:");
467         for (;;)
468         {
469             word[2] = getchar();
470
471             /*****
472             **根据汉字编码的规定, 汉字字符使用双字节表示, 而且第一个字节的表示成数字是 **
473             **大于128, 根据getchar()是否大于128可以判断出是单字节字符还是双字节字符 **
474             **另外根据计数变量count计算出当前操作的字符是双字节字符还是单字节字符。。 **
475             *****/
476             if (((unsigned char)word[2] < 128) && (count % 2 == 0))
477             {
478                 if (word[2] == '\n')
479                 {
480                     break;
481                 }
482                 else if (word[2] == 'Q')

```



1



```

483 |
484 | {
485 |     fflush(stdin);
486 |     return 1;
487 | }
488 | if (Get_Asc_Code(&word[2], Buffer_English))
489 | {
490 |     Print_Asc_Char(Buffer_English, Output_String1, Output_String2);
491 |     memset(Buffer_English, 0, sizeof(Buffer_English));
492 | }
493 | continue ;
494 | }
495 | if ((count % 2) == 0)
496 | {
497 |     word[0] = word[2];
498 |     count++;
499 |     continue;
500 | }
501 | word[1] = word[2];
502 | word[2] = 0;
503 | count++;
504 | if (Get_HzK_Code(word, Buffer_Chinese))
505 | {
506 |     Print_HzK_Char(Buffer_Chinese, Output_String1, Output_String2);
507 |     memset(Buffer_Chinese, 0, sizeof(Buffer_Chinese));
508 | }
509 | else continue;
510 | }
511 | }
512 | return 0;
513 | }
514 |
515 | /*******
516 | // Method: insert_system
517 | // FullName: insert_system
518 | // Access: public
519 | // Returns: int
520 | // Qualifier: 插入字符
521 | // Parameter: void
522 | /*******
523 | int insert_system(void)
524 | {
525 | //变量申明部分
526 |     char get_char[3];
527 |     FILE *fp = NULL;
528 |     int not_find = 0;
529 |     unsigned char qh, wh;
530 |     unsigned long offset;
531 |     char buff[FONTSIZE * FONTSIZE / 8];
532 |     char file_name[] = "HZK32";
533 |     memset(get_char, 0, sizeof(get_char));
534 |     printf("请输入您想加入字库的字:");
535 |     gets(get_char);
536 |     if (strcmp(get_char, "ZZ") <= 0)
537 |     {
538 |         printf("您输入了非法字符!\n");
539 |         system("pause");
540 |         return 0;
541 |     }
542 |     if ((fp = fopen(file_name, "rb+")) == NULL) /*打开字库文件hzk16*/
543 |     {
544 |         printf("Can't open haz16,Please add it?");
545 |         system("pause");
546 |         exit(0);
547 |     }
548 |     /*区码=内码(高字节)-160 位码=内码(低字节)-160*/
549 |     qh = (unsigned char)get_char[0] - 0xa0; /*10进制的160等于16进制的A0*/
550 |     wh = (unsigned char)get_char[1] - 0xa0; /*获得区码与位码*/
551 |     offset = (94 * (qh - 1) + (wh - 1)) * FONTSIZE * FONTSIZE / 8; /*计算该汉字在字库中偏移量*/
552 |     not_find = fseek(fp, offset, SEEK_SET); /*将文件指针移动到偏移量的位置*/
553 |     if (not_find)

```



1




```
554 |         {555 |             printf("未查到该区！error！！\n");
555 |             fclose(fp);
556 |             return 0;
557 |         }
558 |     }
559 |     memset(buff, 0, sizeof(buff));
560 |     fread(buff, FONTSIZE * FONTSIZE / 8, 1, fp);    /*从偏移量的位置读取32个字节*/
561 |     if (!Not_In_Lib(buff))
562 |     {
563 |         printf("字库HZK16存在此字！！\n");
564 |         system("pause");
565 |         fclose(fp);
566 |         return 0;
567 |     }
568 |     fseek(fp, offset, SEEK_SET); //重置字库中的偏移量
569 |     unsigned char buff_time[FONTSIZE * FONTSIZE / 8];
570 |     memset(buff_time, 0, sizeof(buff_time));
571 |     if (!Get_Char_Model(buff_time))
572 |     {
573 |         printf("字模读取失败\n");
574 |         fclose(fp);
575 |         return 0;
576 |     }
577 |     asistant_insert(buff_time);
578 |     fwrite(buff_time, FONTSIZE * FONTSIZE / 8, 1, fp) ? printf("字符加入成功!\n") : printf("字符加入失败!\n");
579 |     printf("qh:%d,wh:%d,offset:%d\n\r", qh, wh, offset);
580 |     system("pause");
581 |     fclose(fp);
582 |     return 0;
583 | }
584 |
585 | //*****
586 | // Method:  delete_system
587 | // FullName: delete_system
588 | // Access:  public
589 | // Returns: int
590 | // Qualifier: 删除字符部分
591 | // Parameter: void
592 | //*****
593 | int delete_system(void)
594 | {
595 |     char character[3], *lib_name;
596 |     int length = 0;
597 |     unsigned long offset = 0;
598 |     memset(character, 0, sizeof(character));
599 |     lib_name = (char *)malloc(30 * sizeof(char));
600 |     memset(lib_name, 0, sizeof(lib_name));
601 |     printf("输入您要删除的字符：");
602 |     gets(character);
603 |     offset = Judge_type_char(character, &offset, &length);
604 |     if (length == 16)
605 |     {
606 |         strcat(lib_name, "ASC16");
607 |     }
608 |     else
609 |     {
610 |         strcat(lib_name, "HZK16");
611 |     }
612 |     Delete_Char_From_Lib(character, offset, lib_name);
613 |     return 0;
614 | }
615 |
616 | //*****
617 | // Method:  main
618 | // FullName: main
619 | // Access:  public
620 | // Returns: int
621 | // Qualifier: 函数入口
622 | // Parameter: void
623 | //*****
624 | int main(void)
```



1



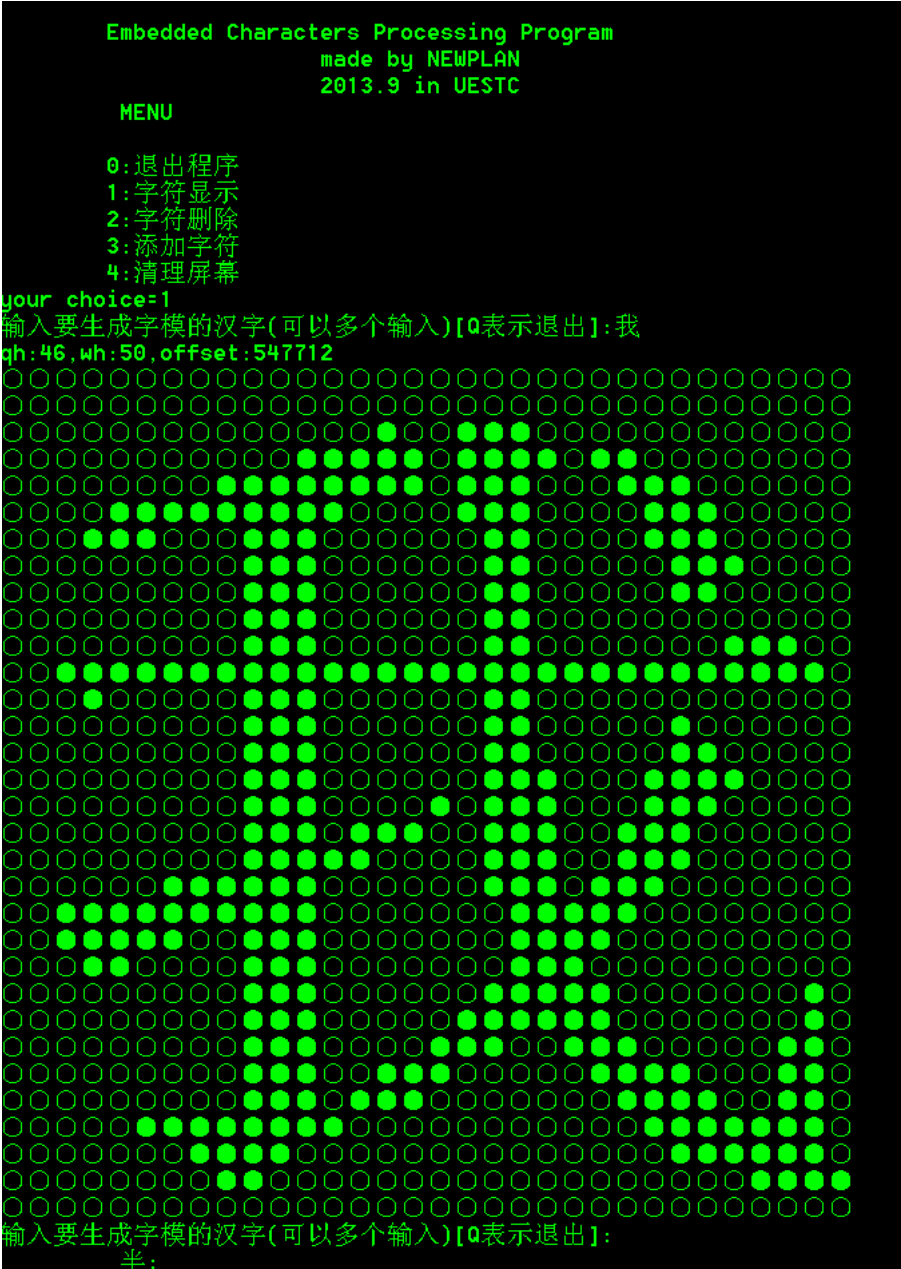
```
625 | {626 |         char choice[2] = {0, 0};
627 |
628 |         int number = -1;
629 |         for (;;)
630 |         {
631 |             printf("\n\tEmbedded Characters Processing Program\n");
632 |             printf("\t\t\tmade by NEWPLAN\n");
633 |             printf("\t\t\t2013.9 in UESTC\n");
634 |             printf("\t MENU\n");
635 |             printf("\t0:退出程序\n");
636 |             printf("\t1:字符显示\n");
637 |             printf("\t2:字符删除\n");
638 |             printf("\t3:添加字符\n");
639 |             printf("\t4:清理屏幕\n");
640 |             printf("your choice=");
641 |             gets(choice);
642 |             number = atoi(choice);
643 |             fflush(stdin);
644 |             switch (number)
645 |             {
646 |             case 0 :
647 |                 printf("退出程序！\n");
648 |                 system("pause");
649 |                 return 0;
650 |             case 1 :
651 |                 output_system();
652 |                 break;
653 |             case 2 :
654 |                 delete_system();
655 |                 break;
656 |             case 3 :
657 |                 insert_system();
658 |                 break;
659 |             case 4 :
660 |                 system("cls");
661 |                 break;
662 |             default :
663 |                 printf("错误输入！\n\n");
664 |                 break;
665 |             }
666 |             number = -1;
667 |         }
668 |     }
669 |     return 0;
670 | }
```



1



测试结果：



1



收藏

分享

一插上电,50平米内都暖和了!3天一度电,今日特惠!

优诺 · 顶新



想对作者说点什么

下载

常用软字库(HZK16,HZK32,ASC16)和示例代码

常用软字库(HZK16,HZK32,ASC16)和示例代码

03-18

点阵字库 包含ASC12/16/48 , HZK12/14/16/24/40/48

点阵字库 包含ASC12/16/48 , HZK12/14/16/24/40/48 多数是在UCDOS7.0拿来的，其他的是在网上搜到的。 共享出来方便需要用到的人士下载。 字...

下载

hzk16+asc16字库

hzk16+asc16字库文件，我的博客有使用demo

06-08

一插上电,50平米内都暖和了!3天一度电,今日特惠!

优诺 · 顶新

下载

最全HZK12,HZK16,HZK24,HZK32,HZK40,HZK48,ASC12, ASC16,ASC24, ASC32, ASC48字库+...

01-1

这几天接了一个项目，需要制作点阵文字，所以接触了字库这东西，由于网上字库挺多的，但是不全，有的字库全，但是字库的读取代码只有一部分，所以还是花费了一天时间搞这...

下载

Delphi实现ASC16和HZK16点阵字的显示

09-2

这是一个使用Delphi实现的ASC16和HZK16 点阵字库显示。字库文件已经取模在单元文件里，无需加载字库文件即可显示。

点阵字库HZK12 HZK16 HZK24 ASC12 ASC16 简介 及 使用方法[附源码]

5894

点阵字库HZK12 HZK16 HZK24 ASC12 ASC16 简介 及 使用方法[附源码] 如何在嵌入式系统中使用大量的汉字和...

来自：zhc的博客

下载

8*16点阵字库ASC16

07-2

UCDOS的8*16点阵字库文件，用于LCD显示英文、数字和符号

下载

最全的点阵字库（ASC12, ASC16, HZK16, HZK24,...）

09-

最全的点阵字库，包括ASC12, ASC16, HZK12, HZK14, HZK16, HZK24, HZK24F, HZK24H, HZK24K, HZK24S, HZK24T等。

陈小春坦言：这游戏不充钱都能当全服大哥，找到充值入口算我输！

贪玩游戏 · 顶新

下载

HZK12,HZK16,HZK24,HZK32,HZK40,HZK48,ASC12, ASC16,ASC24, ASC32, ASC48字库+读取...

02-27

这几天接了一个项目，需要制作点阵文字，所以接触了字库这东西，由于网上字库挺多的，但是不全，有的字库全，但是字库的读取代码只有一部分，所以还是花费了一天时间搞这...

文章热词

CAVLC解码上下文信息 CABAC编解码上下文信息 物联网ISP收集统计信息 片exif信息dng格式 图像修复显示细节

相关热词

c#显示点阵 c++插入一行文本信息 c#显示数据库信息 bootstrap显示后台信息 c++怎么判断一个字符是不是中文字符 人工智能新工科试点班 人工智能新工科实

16进制到ASC 的互转

2889

short CxT_task3_appDlg::hex_asc(unsigned char *hex, unsigned char *asc, long length) { UCHAR hLowbi...

来自：scutedu的博客

断了猫
13篇文章
排名:千里之外
[关注](#)

walk_ing
11篇文章
排名:千里之外
[关注](#)

Mr_zlone
3篇文章
排名:千里之外
[关注](#)

fanwenjieok
169篇文章
排名:千里之外
[关注](#)

12864带字库显示屏滚屏显示

504

/*****//描述：LCD12864的驱动//作者：happygrilch//时间...

来自：happygrilch的博客

下载

ASCII 16x16点阵字库

03-15

ASCII 16x16点阵字库，C语言定义，可在单片机等中使用

GB2312和ASCII码点阵字库HZK, ASC说明使用心得，全

5064

这几天接了一个项目，需要制作点阵文字，所以接触了字库这东西，由于网上字库挺多的，但是不全，有的字库全...

来自：lvjing2的专栏

陈小春坦言：这游戏不充钱都能当全服大哥，找到充值入口算我输！

贪玩游戏 · 顶新

关于十六进制、asc码

1301

在串口调试助手、网络调试助手中可以见到十六进制发送、十六进制接收等选项，对于字符，选中十六进制时，是...

来自：u012252959的博客

下载

16*16和24*24汉字点阵十六进制生成器（源代码）

03-14

16*16和24*24汉字点阵十六进制生成器（源代码）可以得到每一个汉字的不同点阵形式的代码！希望对大家有帮助

点阵字体显示系列之一：ASCII码字库的显示

2666

http://blog.csdn.net/subfate/article/details/6444578 起因：早在阅读tslib源代码时就注意到里面有font_8x...

来自：开挂的熊猫

排序算法 直接插入排序

634

直接插入排序：将序列中的第一个元素作为一个有序序列，然后jiang

来自：四夕日月

<h3>cgi交互</h3> <p>今天终于知道cgi不响应的问题了！原来我设置了boa的根目录为/home/xxx/www，结果html中submit调用的是/v...</p>		445	来自： 李迟的专栏
<h3>一插上电,50平米内都暖和了!3天一度电,今日特惠!</h3> <p>优诺 · 顶新</p>		1	
<h3>ubuntu 源</h3> <p>wangzhi: http://wiki.ubuntu.org.cn/%E6%BA%90%E5%88%97%E8%A1%A8</p>		283	来自： 永立的专栏
<h3>GB2312和ASCII码点阵字库HZK, ASC整理</h3> <p>需求：提取HZK和ASC文件数据,显示,并转换成Unicode编码？ 开发工具&平台：Qt4.7 & Windows7 概述： 1. ...</p>		7963	来自： 来自工作的思考
<h3>ASCII码字库的显示</h3> <p>tslib源代码时就注意到里面有font_8x8.c和font_8x16.c两个文件(它们来自Linux内核，具体目录是,/drivers/video)...</p>		514	来自： bitowan
<div>下载</div> <h3>点阵字库文件-ASC16-24-HZK16-24-32-40-48</h3> <p>点阵字库 ASC16 ASC24 HZK16 HZK24 HZK32 HZK40 HZK48 库文件，用法demo-C语言源码，运行截图及说明</p>		02-06	
<h3>点阵字体文件查看器 c#(HZK16)</h3> <p>在MCU项目中，准备自己写个简单的UI，在中文文字显示，打算选择 UCDOS 中的 HZK16，为了查看字体文件， ...</p>		7249	来自： 的米-漠石's Blog
<h3>别再玩假传奇了，这款传奇爆率9.8，你找到充值入口算我输！</h3> <p>贪玩游戏 · 顶新</p>			
<div>下载</div> <h3>常用的几个字体库---ASC16、HZK16、HZK12,附HZK16的使用资料</h3> <p>常用的几个字体库---ASC16、HZK16、HZK12,附HZK16的使用资料</p>		01-21	
<h3>16*32点阵乱闪问题</h3> <p>本人采用16个74hc595级联控制8快点阵显示汉字，每两块595控制一个点真的行和列，如图(1)片上一样前两排八个595是控制行的，后两排是控制列的...</p>			
<div>下载</div> <h3>12*12,16*16完整点阵字库（包含所有unicode字符，涵粗体，正体，斜体）</h3> <p>取模软件导出的字库，修改部分显示不正确的字符！涵盖unicode编码中所有字符的点阵字模数据（15M），格式：DZK</p>		01-10	
<h3>ASCII字符点阵字库的制作和使用</h3> <p>开发环境：Win7，Eclipse，MinGW 1、生成ASCII字符文件ASCII编码的可打印字符是0x20~0x7E,先用运行下面...</p>		1.8万	来自： Bob's Blog
<h3>UCDOS点阵字库</h3> <p>ASC12 ASCII字库文件 12X6 ASC16 ASCII字库文件 16X8 ASC48 ASCII字库文件 48X24 Hzk...</p>		1763	来自： c1505011056的专栏
<h3>{GPS接收机},信号稳定,免费培训</h3> <p>陕西远程测量 · 顶新</p>			
<h3>babyos（六）—— 显示ASCII字符和汉字</h3> <p>注：本程序为原创，若发现bug，万望指出，若有问题，欢迎交流，转载请注明出处。若能有助于一二访客，幸甚...</p>		4220	来自： 孤舟钓客
<div>下载</div> <h3>GB18030汉字16x16点阵字库</h3> <p>GB18030汉字16x16点阵字库，取模方式：纵向取模，字节逆序</p>		01-23	
<h3>UCDOS中的点阵字库HZK12,HZK16,HZK24,ASC12,ASC16(转)</h3> <p>原文：http://cache.baidu.com/c?m=9f65cb4a8c8507ed4fece7631046893b4c4380147780914c34c3933fc23...</p>		8555	来自： jszj的专栏
<h3>使用UCDOS的HZK16点阵字库的演示程序</h3> <p>最近看到嵌入式里显示汉字都是用的UCDOS的字模 我的8051开发板一直没能玩得起来 所以写了个测试程序 体验一...</p>		1259	来自： 断了的猫的博客
<h3>HZK16点阵字库程序</h3> <p>最近(很久之前的最近)在弄硬件，买了一块彩屏，需要字库，所以就把很久以前会的知识拿出来温习了一遍。果然好...</p>		592	来自： walk_ing的博客
<h3>劲爆！！！本地人都在玩这样技能酷炫,画面超赞的页游!</h3> <p>热门推荐</p>			

<div>点阵字符</div> <div>题目描述 点阵字符是计算机系统存储、显示文字字型信息的一种方式，其中每个字型（字母、数字、标点符号等）...</div>	<div>410</div> <div>来自： 龍、逸的博客</div>
<div>使用HZK16字库文件提取文字并显示到OLED显示屏上（Java Android实现）</div> <div>最近在做要把文字（英文、汉字）显示到OLED显示屏上这方面内容，英文还好说，无非几十个字符，网上一搜也都...</div>	<div>5562</div> <div>来自： breeze_wf的专栏</div>
<div><div>下载</div><div>HZK16中文字库文件</div></div> <div>很多人都在找HZK16这个中文字库文件,居然还有人要资源分。我这里有,传一份上来</div>	<div>08-1-</div>
<div>关于HZK16数字和字母的字模提取</div> <div>HZK16汉字的字模我知道怎么提取，但是字母和数字这些字符的要怎么提取呢？比如字母“a”的字模信息应该怎么算呢，希望知道的不吝赐教。</div>	<div><</div> <div>></div>
<div>字符理论--hzk16的介绍以及简单的使用方法（转）</div> <div>HZK16字库是符合GB2312标准的16×16点阵字库,HZK16的GB2312-80支持的汉字有6763个，符号682个。其中...</div>	<div>417</div> <div>来自： L_Backkom的专栏</div>
<div>点阵纵向取模以及放大算法</div> <div>这两天在调d'y</div>	<div>4351</div> <div>来自： 君莫笑</div>
<div>字模的原理和获取(一)</div> <div>一、什么是字模单片机应用经常会用到液晶显示模块，是用点阵的方式显示的，要显示汉字或字符的时候会用到字...</div>	<div>6590</div> <div>来自： 起步</div>
<div><div>下载</div><div>ASC 16*24 点阵字库</div></div> <div>ASC 16*24 点阵字库</div>	<div>10-10</div>
<div><div>下载</div><div>ASC12字库、ASC16字库、HZK12字库、HZK16字库</div></div> <div>ASC12字库、ASC16字库、HZK12字库、HZK16字库</div>	<div>03-04</div>
<div>linux字符点阵的显示-韦东山第三期笔记</div> <div>最近在学习韦东山系列课程的第三期，本篇文章做一些记录。 本篇，我的目标是学习字符点阵的显示。首先要...</div>	<div>337</div> <div>来自： qq_26879985的博客</div>
<div>hzk16的介绍以及简单的使用方法</div> <div>hzk16的介绍以及简单的使用方法 HZK16 字库是符合GB2312标准的16×16点阵字库,HZK16的GB2312-80支持的...</div>	<div>1907</div> <div>来自： fanwenjieok的专栏</div>
<div><div>下载</div><div>利用java开发点阵字体显示系(ASCII码字库的显示)</div></div> <div>读取ASCII字库信息，并显示出来的完整代码。程序支持字符串输入，输出以*为信息的点阵信息图，如：I Love Java!显示如下 ...</div>	<div>10-28</div>
<div>基于嵌入式Linux的LCD液晶点阵显示的基本实现</div> <div>本文以嵌入式Linux为板载系统。写一个测试LCD液晶点阵的小例子，在这个小例子当中主要实现液晶点阵的中文和...</div>	<div>551</div> <div>来自： TECH_PRO的博客</div>
<div>点阵字库HZK12 HZK16 HZK24 ASC12 ASC16 简介 及 使用方法</div> <div>如何在嵌入式系统中使用大量的汉字和字符呢? DOS前辈们经过艰辛的努力，将制作好的字模放到了一个个标准的...</div>	<div>2050</div> <div>来自： xyc310898673的专栏</div>
<div>HZK16汉字16*16点阵字库的使用及示例程序</div> <div>http://www.cnblogs.com/nbsofer/archive/2012/11/01/2749026.html 最近在弄硬件, 买了一块彩屏, 需要字库,...</div>	<div>1853</div> <div>来自： zenwanxin的专栏</div>
<div><div>下载</div><div>hzk16所有字体字模</div></div> <div>hzk16所有字体 下面是在WINDOWS下面生成的不同字体风格的HZK16： 宋 - - HZK16S 仿 - - HZK16F 黑 - - HZK16H 楷 - - HZK16K 仿与楷字型相近而风...</div>	<div>04-21</div>
<div>hzk16的使用与介绍</div> <div>HZK16字库是符合GB2312标准的16×16点阵字库,HZK16的GB2312-80支持的汉字有6763个，符号682个。其中...</div>	<div>1630</div> <div>来自： 行者</div>
<div>Eclipse中离线安装ADT插件详细教程</div> <div>在搭建Android开发环境的时候，我们需要为Eclipse安装ADT (Android Development Tools) 插件，这个插件可...</div>	<div>49489</div> <div>来自： Dr.Neo的专栏</div>
<div>将Excel文件导入数据库（POI+Excel+MySQL+jsp页面导入）第一次优化</div> <div>本篇文章是根据我的上篇博客，给出的改进版，由于时间有限，仅做了一个简单的优化。相关文章：将excel导入数...</div>	<div>5322</div> <div>来自： Lynn_Blog</div>

<div>jquery/js实现一个网页同时调用多个倒计时(最新的)</div> <div>jquery/js实现一个网页同时调用多个倒计时(最新的) 最近需要网页添加多个倒计时. 查阅网络,基本上都是干遍一律...</div>	<div>👁 5021</div> <div>来自： websites</div>
<div>前后端分离之Springboot后端</div> <div>这是上一篇博客前后端分离之Java后端的重写. 源码 前后端分离的后端主要解决的就2个问题：跨域访问(CORS)和to...</div>	<div>👁 9834</div> <div>来自： jimo_lonely的博客</div>
<div>人脸检测工具face_recognition的安装与应用</div> <div>人脸检测工具face_recognition的安装与应用</div>	<div>👁 7820</div> <div>来自： roguesir的博客</div>
<div>linux上安装Docker(非常简单的安装方法)</div> <div>最近比较有空，大四出来实习几个月了，作为实习狗的我，被叫去研究Docker了，汗汗！ Docker的三大核心概念...</div>	<div>👁 78671</div> <div>来自： 我走小路的博客</div>
<div>Docker下实战zabbix三部曲之一：极速体验</div> <div>用docker来缩减搭建时间，让读者们尽快投入zabbix系统的体验和实践</div>	<div>👁 1401^</div> <div>来自： boling_cavalry的博</div>
<div>Java面试题全集（上）</div> <div>2013年年底的时候，我看到了网上流传的一个叫做《Java面试题大全》的东西，认真的阅读了以后发现里面的很多...</div>	<div>👁 912537</div> <div>来自： 骆昊的技术专栏</div>
<div>关于计算时间复杂度和空间复杂度</div> <div>相信学习编程的同学，或多或少都接触到算法的时间复杂度和空间复杂度了，那我来讲讲怎么计算。 常用的算...</div>	<div>👁 33644</div> <div>来自： 杨威的博客</div>
<div>关于SpringBoot bean无法注入的问题（与文件包位置有关）</div> <div>问题场景描述整个项目通过Maven构建，大致结构如下： 核心Spring框架一个module spring-boot-base service...</div>	<div>👁 27791</div> <div>来自： 开发随笔</div>
<div>ZooKeeper系列之二:Zookeeper常用命令</div> <div>ZooKeeper服务命令： 在准备好相应的配置之后，可以直接通过zkServer.sh 这个脚本进行服务的相关操作 1. 启...</div>	<div>👁 284072</div> <div>来自： xiaolang85的专栏</div>
<div>12种排序算法详解</div> <div>作者：寒小阳 时间：2013年9月。 出处：http://blog.csdn.net/han_xiaoyang/article/details/12163251。 声明...</div>	<div>👁 26626</div> <div>来自： u014682691的专栏</div>
<div>Java设计模式学习08——组合模式</div> <div>一、组合模式适用场景把部分和整体的关系用树形结构来表示，从而使客户端可以使用统一的方式对部分对象和整...</div>	<div>👁 696</div> <div>来自： 小小本科生成长之路</div>



尼奥普兰

关注

原创52

粉丝51

喜欢59

评论24

等级： 博客 5

积分： 5641

勋章： 恒

访问： 65万+

排名： 7257

反馈意见：

视觉干扰

内容不宜

不感兴趣

其他原因

[返回](#)

最新文章

如何快速截取论文中的latex公式

IO多路复用和epoll反应堆

计算机书单阅读推荐

CentOS7源码安装MxNet

centos7从源码安装GPU版本的caffe

个人分类

ACM18篇

操作系统4篇

spice1篇

算法导论（ 算法实现C/C++ 版 ）19篇

编译原理4篇

展开

归档

2018年12月1篇

2018年11月1篇

2018年10月1篇

2018年7月1篇

2018年5月2篇

展开

热门文章

堆排序
阅读量：59017

整数拆分问题的四种解法
阅读量：32034

递归下降分析法（ 编译原理 ）
阅读量：25478

农夫过河问题算法设计与实现
阅读量：19535

迷宫问题算法设计与实现
阅读量：17354

最新评论

Ubuntu14.04（ 64位 ） ...
sinat_38347209：感谢博主，照着你的教程，总
管把spice协议装上了。

俄罗斯方块的源码实现
weixin_42095121：想问你一下这个(((m->area[m->y + 0] >> m->...)

编译caffe2遇到的坑
u011901279：再次感谢，看到这几行字让我甚是
欣慰哈哈。 [100%] Linking CXX shared li...

编译caffe2遇到的坑
u011901279：其实，我是能看出来CMakeFile中
缺少mpi库的链接信息，但是不太懂cmake文件的
写法，所以...

编译caffe2遇到的坑
u011901279：我靠，这个问题搞了我好几天，我
以为是我cuda+opencv版本不兼容，就把cuda和
opencv...

1





恶霸犬多少钱



联系我们



微信客服



QQ客服

 QQ客服

 kefu@csdn.net

 客服论坛

 400-660-0108

工作时间 8:30-22:00

关于我们

招聘

广告服务

网站地图

 百度提供站内搜索 京ICP证09002463号

©1999-2019 江苏乐知网络技术有限公司


江苏知之为计算机有限公司 北京创新乐知

信息技术有限公司版权所有

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心



1

