

# 嵌入式編程中的高低位交換如何實現？

嵌入式大雜燴 2021-12-19 21:58

以下文章來源於技術讓夢想更偉大，作者李肖遙



**技術讓夢想更偉大**

一個認真做技術的職場老鳥，創業公司技術負責人，孵化編程，樂於分享，講原理，摳...



[關注](#)、[星標公眾號](#) · 直達精彩內容



微信搜一搜

嵌

ID：技術讓夢想更偉大

作者：李肖遙

最近的協議中遇到字節高低位轉換的問題，於是偷懶上網查看，遇到類似的問題，也認識一個新的名字，叫做**蝶式交換**

## 問題是這樣子的

協議要求字節低位在左，高位在右，對每個字節做轉換處理，逐個交換其高低位，例如 **11010001**，經過 **0->7**、**1->6**、**2->5**、**3->4** 對應位的交換，變成 **10001011**。

這裡就需要有位操作的知識，參考 [位操作運算有什麼奇技淫巧?\(附源碼\)](#)

這裡我們以8位數據高低位轉換為例子：

```
1  #include <stdio.h>
2
3  unsigned char highAndLowShiftHEX(unsigned char data);
```

```

4 void printBin(int n);
5
6 int main () {
7
8     highAndLowShiftHEX(209);
9     return 0;
10 }
11
12 unsigned char highAndLowShiftHEX(unsigned char data)
13 {
14     unsigned char i;
15     unsigned char tmp=0x00;
16     for(i=0;i<8;i++)
17     {
18         tmp=((data>>i)&0x01)|tmp;
19         //printBin(tmp);
20         if(i<7)
21             tmp=tmp<<1;
22     }
23     printf("\nafter shift data:");
24     printBin(tmp);
25     return tmp;
26 }
27
28 //由于二进制直观，故写了一个打印二进制的函数
29 void printBin(int n)
30 {
31     int len = sizeof(n)*8;//总位数。
32     int i;
33     if(i == 0)
34     {
35         printf("0");
36         return;
37     }
38     for(i = len-1; i >= 0; i --)//略去高位0.
39     {
40         if(n&(1<<i)) break;
41     }
42
43     for(;i>=0; i --)
44         printf("%d", (n&(1<<i)) != 0);
45 }

```

熟悉位操作，上述代碼實現起來就比較簡單。在嵌入式開發中這類問題通常使用蝶式交換法和查表法來實現。

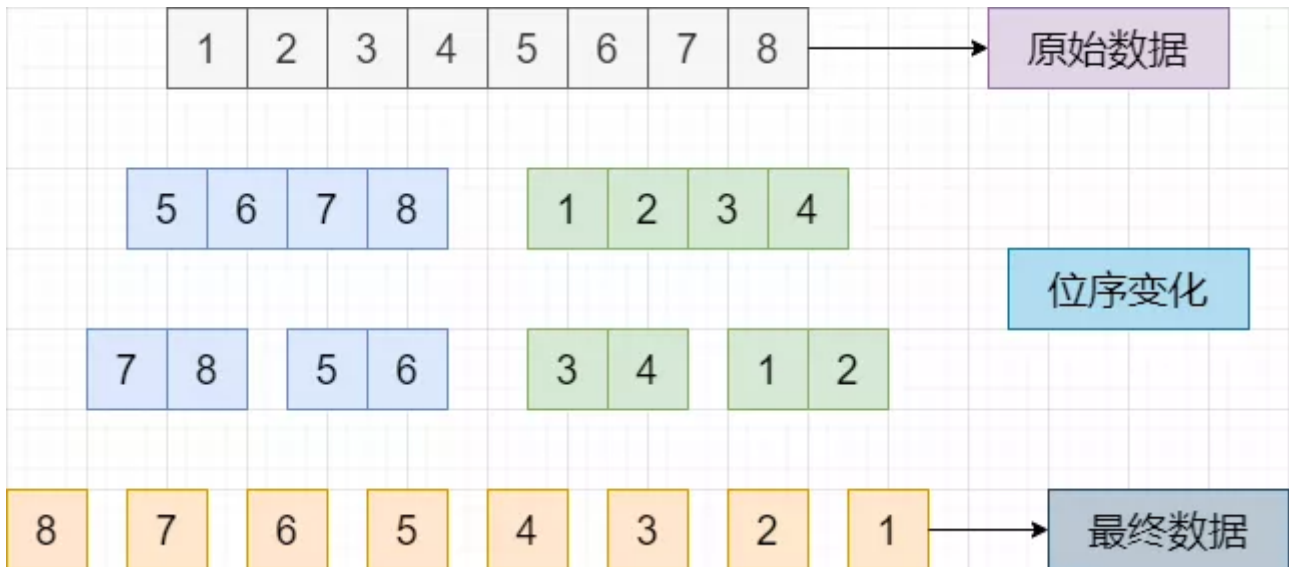
## 高階實現？

查表法就是將一些值存到內存中，在需要計算的時候查表，但是這種方法會佔用額外的存儲空間。

所以這裡主要介紹一下蝶式交換法，我們以8位數據轉換為例。

- 假設原始序列為： 1 2 3 4 5 6 7 8
- 目標序列為： 8 7 6 5 4 3 2 1

那麼流程圖如下：



由此完成了整個位的逆序轉換，同樣以 11010001 為例，下面是具體的實現代碼：

```

1  #include <stdio.h>
2
3  unsigned char highAndLowShiftHEX(unsigned char );
4  void printBin(int );
5
6  int main () {
7
8      highAndLowShiftHEX(209);
9      return 0;
10 }
11
12 unsigned char highAndLowShiftHEX(unsigned char data)
13 {
14     data=(data<<4)|(data>>4);
15     data=((data<<2)&0xcc)|((data>>2)&0x33);
16     data=((data<<1)&0xaa)|((data>>1)&0x55);
17     printf(" after shift data=%x \n",data);
18     printBin(data);
19     return data;
20 }
21
22 //由于二进制直观，故写了一个打印二进制的函数
23 void printBin(int n)
24 {
25     int len = sizeof(n)*8;//总位数。
26     int i;
27     if(i == 0)

```

```

28     {
29         printf("0");
30         return;
31     }
32     for(i = len-1; i >= 0; i --)//略去高位0.
33     {
34         if(n&(1<<i)) break;
35     }
36
37     for(;i>=0; i --)
38         printf("%d", (n&(1<<i)) != 0);
39 }

```

交换字节的高低位並不是一個很常見的問題，遇到該問題時，需要經過仔細的分析，加上對C語言位操作的熟練掌握，就能夠很好的解決這一類的問題。

## 擴展

那麼我們將其擴展到16位半字數據的高低位轉換。

其實原理是跟8位的時候是一樣的，採用簡單的移位方式對數據的高低位進行轉換。熟悉位操作的話代碼就相對簡單了。

下面是對該思路的具體實現。

```

1  #include <stdio.h>
2
3  void expandPrintBin(int val2);
4  unsigned short HighAndLowSwitchHEX(unsigned short data);
5
6  int main () {
7
8      HighAndLowSwitchHEX(38491);
9      return 0;
10 }
11
12
13 //由于二进制直观，故写了一个扩展的打印二进制的函数
14 void expandPrintBin(int val2)
15 {
16     int i,k;
17     unsigned char *p = (unsigned char*)&val2 + 3; //从低位到高位, 低端字节计算机
18     for( k = 0; k <= 3; k++)
19     {
20         int val2 = *(p-k);
21         for (i = 7; i >= 0; i--)
22         {
23             if(val2 & (1 << i))
24                 printf("1");
25             else

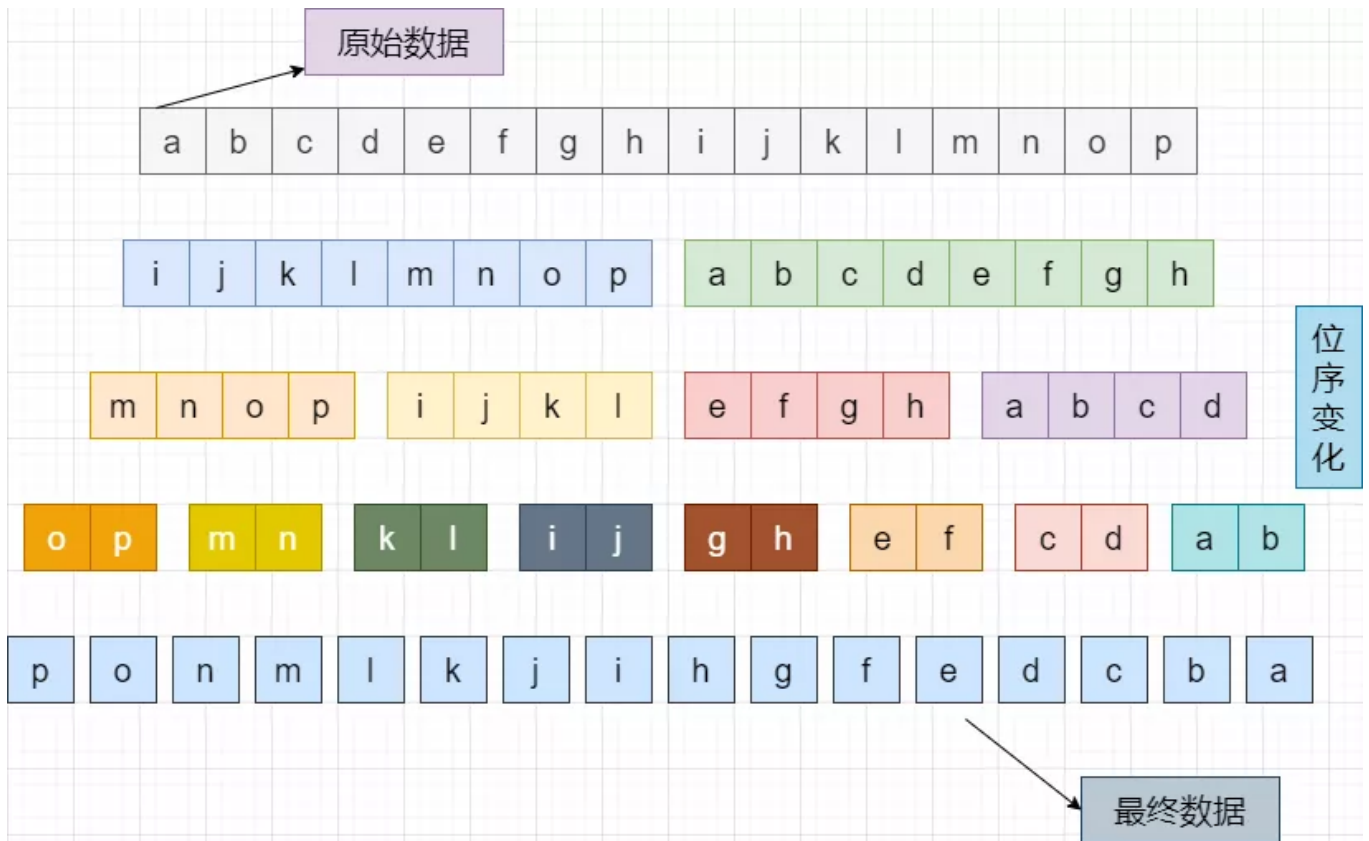
```

```
26         printf("0");
27     }
28     printf(" ");
29 }
30 }
31 unsigned short HighAndLowSwitchHEX(unsigned short data)
32 {
33     unsigned char i = 0;
34     unsigned short temp = 0x0000;
35
36     for(i = 0; i < 16; i++)
37     {
38         temp = ((data >> i) & 0x0001) | temp;
39         if(i < 15)
40         {
41             temp = temp << 1;
42         }
43     }
44     printf("temp:%x\n\n",temp);
45     expandPrintBin(temp);
46     return temp;
47 }
```

同樣的所謂的蝶式交換法，我引用了字節交換法的例子，我們可以演算一下：

- 假設原始序列為： a b c d e f g h i j k l m n o p
- 目標序列為： p o n m l k j i h g f e d c b a

那麼流程圖如下：



由此完成了整个位的逆序转换，完成了算法的拓展，以 **10010111001011011** 为例，下面是具体的实现代码：

```

1  #include <stdio.h>
2
3  unsigned short highAndLowShiftHEX(unsigned short data);
4  void expandPrintBin(int val2);
5
6  int main () {
7
8      highAndLowShiftHEX(38491);
9      return 0;
10 }
11
12 unsigned short highAndLowShiftHEX(unsigned short data)
13 {
14     data = (data << 8) | (data >> 8); //0101101110010110
15     data = ((data << 4) & 0xF0FF) | ((data >> 4) & 0xFF0F); //1011010101101001
16     data = ((data << 2) & 0xC000) | ((data >> 2) & 0x3333); //1110010110010110
17     data = ((data << 1) & 0xAAAA) | ((data >> 1) & 0x5555); //1101101001101001
18     printf(" after shift data=%x \n",data);
19     expandPrintBin(data);
20     return data;
21 }
22
23 //由于二进制直观，故写了一个扩展的打印二进制的函数
24 void expandPrintBin(int val2)
25 {
26     int i,k;
27     unsigned char *p = (unsigned char*)&val2 + 3; //从低位到高位，低端字节计算机
28     for( k = 0; k <= 3; k++)
29     {

```

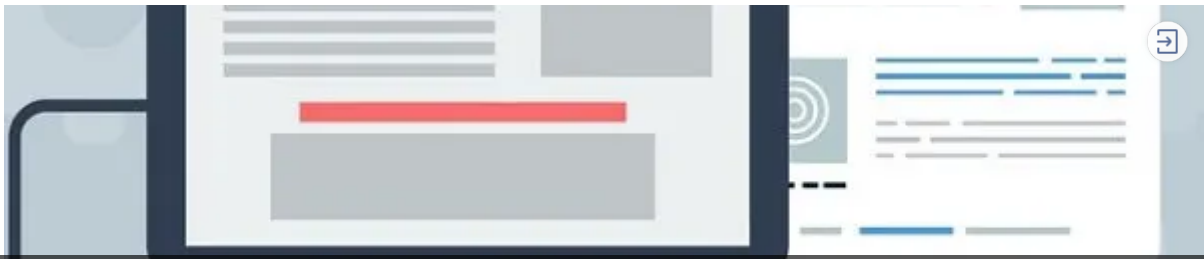
```
30     int val2 = *(p-k);
31     for (i = 7; i >= 0; i--)
32     {
33         if(val2 & (1 << i))
34             printf("1");
35         else
36             printf("0");
37     }
38     printf(" ");
39 }
40 }
```

對於這種交換位的問題，在嵌入式開發中會經常遇到，熟悉掌握位操作還有調試手段很重要。熟練的利用好C語言的移位操作，就能快速的解決此類問題。

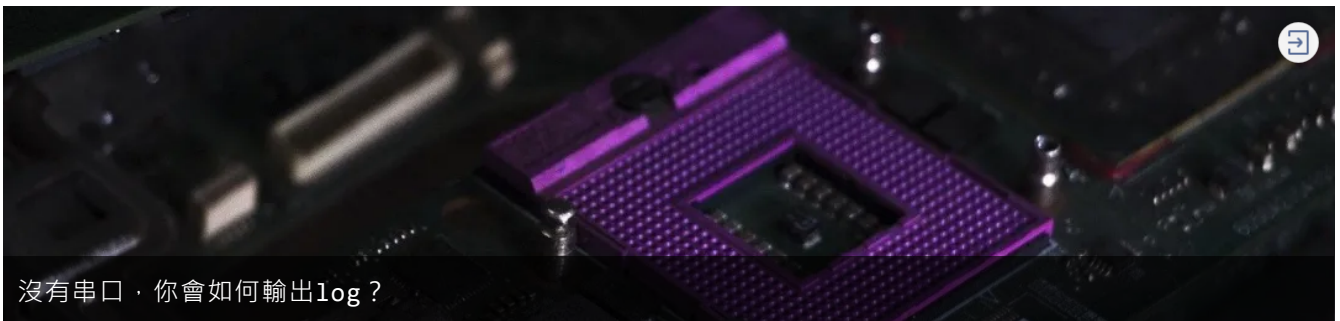
## 總結

上面的問題關鍵還是位操作的靈活妙用，其次是寫了兩個打印二進制的函數，方便使用，代碼可以直接使用，調試不易，歡迎點贊，在看，轉發也是厚愛，我們下一期，再見！

— — — — — *E N D* — — — — —

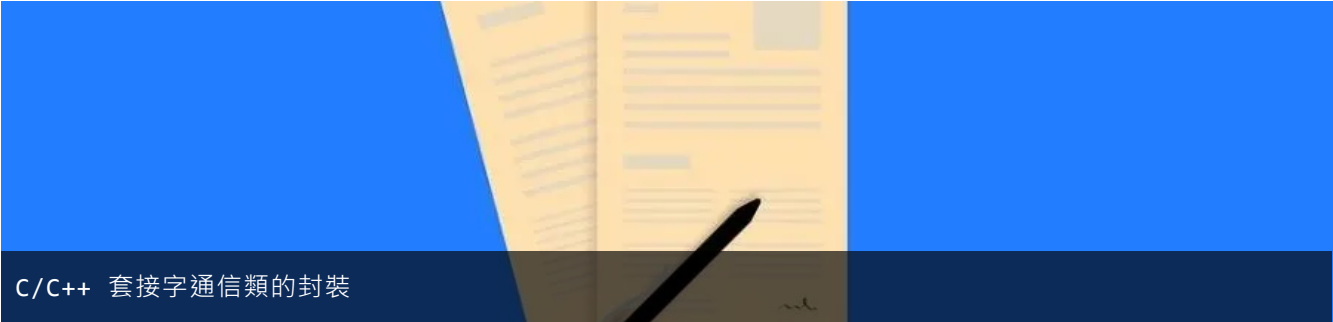


入坑電子後悔嗎？談談我的電子入“坑”記



沒有串口，你會如何輸出log？





喜歡此內容的人還喜歡

嵌入式開發比較常用的C語言知識點  
硬件攻城獅



17條C語言編程小知識  
C語言與C++ 編程



嵌入式狀態機編程-QP狀態機框架與常見狀態機方法  
技術讓夢想更偉大

