

您設計產品時的好朋友！



Forum: [程式軟體與開發工具](#)

Topic: [Atmel studio 7 無法print float](#)

Subject: [Re: Atmel studio 7 無法print float](#)

作者: [jason680](#)

2020年02月07日 00:14:48

看了為了"印出浮點數"...
但需要佔用大量資源(程式跟資料記憶體)...
對於小顆MCU應該會承受不了...

浮點數相關資料IEEE754,請自行參考
浮點數一般來說有
32位元單精度(float)
64位元雙精度(double)
兩者格式均為
符號 + 指數 + 有效值
想要支援"印出"完整浮點數範圍與格式
必然要花費一番心思來設計程式
自然就會耗掉一些資源...
註:這些資源對小型MCU算是很大的負擔
對中大型MCU也是佔一定比率資源跟運算時間

基本上,在個人電腦或大型MCU上(記憶體較多,運算速度快)
只是印出浮點數不是太大問題...

但對一般MCU而言,資源有限,速度不快...
對於只是顯示特定小的範圍值(例:小數2位數或3位數)
可善用一些"整數"運算,來取代%f,做"印出"浮點數... ..

基本上浮點數範圍很大,以下分別討論:
註:以下僅以正數(大於0)討論,
負數先變為正數,再加負號
以下討論使用int或long,
用float或double,請跟據使用範圍,自行修改

1. 值很大,超出int, long等等
一般會改變單位來因應
例如: 35K, 64M, 512G
所以這個最後也是"小"的整數(或帶有小數)

2. 數值不大,但需要小數點後的數字
例: 170.3公分, 3.12V, ...

3. 數值很小(小於1,大於0)
例: 0.000244

=====

以下程式碼,貼上來後,縮排會不見...
放入code區塊,中文又會有問題,
所以最後就保持 文字 模式...
縮排改為(中文)全型空白
編譯之前要把 全型空白 改回來...

=====

簡易方式1:

放大一百倍,加入百分比%

```
void printf_2f(float val){  
    printf("%02d%%", val * 100);  
}
```

優點: 小數2位,程式碼簡易

缺點: 跟一般習慣不同(3.14變為314%)

特點: 適用小數2位,要適應 %符號

註: val乘以百倍後不可超過int(或long)的最大值

簡易方式2:

放大千倍(小數3位)以上...

```
void printf_3f(float val){  
    printf("%03ld@&quot;;", val * 1000);  
}
```

優點: 小數3位,程式碼簡易

缺點: 跟一般習慣不同(1.523變為1523@)

特點: 適用小數3位,要適應(特定) @符號

僅適用於臨時debug使用(不然1532@ 很容易看不懂)...

註: val乘以千倍後不可超過long(%ld)的最大值

標準的顯示方式1:

整數跟小數分開處理,中間再加小數點(.)

```
void printf_3f(float val){  
    int dig, flt;  
    dig = val; // 取得整數  
    flt = (val+0.0005-dig)*1000; // 小數處理, +0.0005 是四捨五入  
    // flt = (val-dig)*1000; // 小數處理, 不含四捨五入  
    printf("%d.%03d", dig, flt);  
    // 因為小數3.015 的小數部分為 15所以一定要補0(補成小數3位)  
}
```

優點: 標準顯示方法,程式簡單跟速度夠快

缺點: 顯示小數範圍固定3位數

特點: 能正常顯示3位小數,程式碼小,速度夠快

整數部份為int(或long)最大值,已經適合一般情形使用

標準的顯示方式2:

需要比較多的小數位數(帶小數位數的參數)

```
void printf_nf(float val, int dot){  
    int dig;  
    char flt[10];
```

```
int cnt;
int tmp;

dig = val; // 取得整數
val -= dig; // 小數處理
for(cnt=0; cnt <=dot;cnt++){
    tmp = val = val * 10;
    flt[cnt] = tmp + '0'; // 轉ASCII
    val -= tmp;
}
flt[cnt] = ' ';
printf("%d.%.sn", dig, flt);
}
```

優點：標準顯示方法,可指定小數位數

缺點：沒有四捨五入

特點：小數部份用字串處理(乘以10,移位...)

可改進空間：整數範圍(int最大值)及小數四捨五入...

寫到這邊,再寫下去,就程式就愈來愈大,運算時間也變多...

就跟一開始所設定,簡易,替代%f弁鄴漸媚 愈來愈遠...