

# 您設計產品時的好朋友！



Forum: [32-bit PIC](#)

Topic: C32除法的問題

Subject: Re: C32除法的問題

作者: Eigen

2011年11月17日 08:41:54

常數定義，讓數值具有可讀性並沒有錯

有問題的應該是常數的處理，你是在 compiler 還是 preprocessor 處理

如果你用

```
#define pulse_const (1.0/0.001)
或
#define pulse_const 1000 ;(1.0/0.001)
```

之後再

```
WORD dw =pulse_const
```

結果應該就會不一樣 另一點，1000 和 999 ，相差也才 0.1% ，你會覺得差很大是因為 1.0/0.001 心算就知道是整數

如果今天是 ( 2.57/0.021) 出來的是 121 122 或 123 你怎麼解讀對或錯？

重點在於數值的合理範圍 及單位大小

(1.0/0.001) 這是一般人單位受限於 整數、小數的習慣

如果我來寫，會變成 1000/1 ，整個思考單位變成 0.001 為一個單位而不是 1 整數 ，思考單位的不一樣，反應出來的就是數值的合理性及程式的大小及操作速度。

數值範圍反應在 這個例子更是顯著

```
Vref+ = Vdd = 5V
#define adc2volt(n) ( n* 5 - (n*30+128)/256)
#define adc2volt(n) ( n* 4 - (n*226L+128)/256)
```

```
Vref+ = Vdd = 3.3V
#define adc2volt(n) ( n* 4 - (n*199L+128)/256)
#define adc2volt(n) ( n* 3 - (n*57+128)/256)
```

這種寫法上更是常見

```
一般的 adc2volt
#define adc2volt(adc) (adc /1024*5.0*100)
#define adc2volt(adc) (adc /1024*3.3*100)
```

我來寫會變成

```
#define adc2volt(adc) ((adc*0x7d) >>8)
#define adc2volt(adc) ((adc*0xA5) >>9)
```

單單 float 乘除法和 char 乘法+位移，速度和空間兩者相差數千倍

當然，系統 loading 不大，不會被 code review，你要怎麼寫是沒差

但是追求卓越時，換個思考方式，過程會大大的不同（雖然結果會很接近）

（正確的計算應該是 /1023 而不是 /1024，但是如果你用 excle 驗證，

你會發現兩者顯示到小數後兩位，結果是一樣的）

有心，我會建議去讀讀 misra C 的規範，一百多條不會很多