

## Re: scaling coefficients for c

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2006-09/msg02836.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-09/msg02836.html)

---

- *From:* "Ancient\_Hacker" <[grg2@xxxxxxxxxxxx](mailto:grg2@xxxxxxxxxxxx)>
  - *Date:* 21 Sep 2006 08:41:28 -0700
- 

sonos wrote:

Hi,  
I understand the requirement of scaling coefficients for fixed point microcontrollers. However, I cannot seem to get past a simple issue... How do I code a fraction in binary form for use in c code or asm language?

for example, consider the fraction in base 10:  
dec == 0.58642

It depends. if your microcontroller has 8 bit arithmetic only, you'll have to scale your fractions to fit the range 0..255 or -128..127. Not only that, you'll have to scale your intermediate results so they don't overflow or underflow.

if your microcontroller has 16 bit arithmetic, you'll have to scale your fractions to fit the range 0..65535 or -32768..32767

For the 8-bit case, you're going to lose a lot of resolution. One first cut might be to scale everything up by a factor of 100, so in your case:

```
int dec=59
```

but I suspect your C can do 16-bit math, so a better choice might be 1000:

```
int dec=586;
```

again it depends on the exact math you want to do. For If you're going to be multiplying, you have to ensure the product doesnt overflow 15 or 16 bits, so you have to scale the inputs so they're no more than 8 bits each.

A smarter way would be to take the  $\log_2()$  of each operand and scale them so the sum of the logs is less than 16 or 15.

Re: scaling coefficients for c

Perhaps if you gave us some more example code we could make better suggestions.

then whenever you add numbers, you just make sure all the operands have been similarly scaled, and the sum will be scaled by 100 also.

.