

The following comments are written with the best possible intent: to provide another set of eyes.

Page 6: “..fully C compliant Propeller GCC compiler” A pointer to something that describes C would be good here. I want to learn about C in general, and would like to be able to find a complete syntax description. Perhaps a list of reserved words with definitions? I know your libraries are well documented, what about the language itself?

Page 25 ,Figure 1-22 : It would be useful to suggest attempting to redefine variable c (as in `int c=a-b`) to see what happens. Perhaps as an Exercise?

Page 27, Project 2: I think you ought to say something about precedence here. Maybe even MyDearAuntSally.

Page 30: Actually, I think a lot of people just say led, all one syllable.

Page 31: “Many circuit diagrams..” I would have said “The standard schematic symbol is the jagged line...”.

Page 31: “This is numbered...is...” s/b “This numbered...is...”

Page 32: Quite aside from this book, it is a shame that resistor color codes are so difficult to read these days (indistinct, faded colors). Back in the day, when resistors were 1 or 2 watt carbon comp, you could read them.

Page 35 or 36: Somewhere in here, I think you should warn about turning power off when building circuits. I know you do the steps on page 37, but by the time most people get there, they’ve already built the circuit and it’s too late. I think you need to bump it up a little.

Page 36, Figure 2-6. The LED is NOT wired directly to power, it is powered through a resistor. Important distinction.

Page 43 LED-OnOff: I really truly most sincerely believe that allowing students to use literal pin numbers, even at the very beginning, is to teach bad programming practices. Introduce `#constant`, or whatever you want to substitute (e.g. `int ledPin=14`) but do not allow literals. Okay, I won’t talk about it any more.

Page 43: I couldn’t find a Learn > Examples > WAMM folder. Pretty sure I have up to date stuff.

Page 44: I would have bragged a little about how nicely the tabs work while keying in curly braces! Just a little.

Page 46: Actually, you don’t need to comment out the extra curly braces. Works just fine with them there.

Page 53, Figure 2-14. The yellow LED is shown wired to 3.3V rather than pin 15. Also the schematic.

Page 62: I would say something here about color fusing (it starts out looking pretty yellow. Explain why) Even better, an exercise that starts out with the colors very distinct and then fusing. For (`x=200, x>0, x--`)

Page 63, 7: Name at least three C variable types.

Page 64: Q3: You can connect 5-socket groups to connect even more wires.

Page 69: The warning box about power is really great and I think it would be even better on page 35 or 36.

Page 78: "Active High box": "Active high" and "active low" have so many meanings that I think I would just avoid them. Perhaps "activated" or "pressed" or "mashed" would be better. In any case "pull" doesn't work very well here.

Page 85: The \n in this string is pretty redundant, isn't it? But your typical new guy is gonna think it isn't.

Page 91: Period missing at end of 2nd print command (...light turn red.) Two places on 91, also 99 and 101.

Page 91: comment for `pause(1);` should be `delay 1 millisecond`

Page 92: You know, I put empty curly brace pairs `{}` under `while`s without code blocks just to remind me.

Page 95: "...using a trick..." I don't think it is a trick at all. It is just using the operators.

Page 105, Figure 4-2: item (3) is the horn, not the screw.

Page 105, Figure 4-3: The 20 msec includes the pulse itself, not the low period between pulses.

Page 109, figure 4-8. Since we are going to put the pushbuttons back on later, it would have made sense to leave them on here. (see activity 6, page 127.)

Page 126. Comments should be `// 0 to 180, steps of 0.6 degrees` and `// 180 to 0, steps of 1.2 degrees`

Page 130:, code fragment: `print ("%c angle = %d.%d %c", HOME, angle/10, angle%10, CLREOL);` gives a more nearly correct display and shows how to display fractions in an integer environment.

Page 147 ff: I changed the printout at the beginning to say "x prior to the for loop" and at the end to say "x following the for loop=". This emphasizes that they are not within the for loop. Further, it is *all* within main, isn't it?

Page 155: After the `cog_run`, I added `print("Returned Value: %d\n %d\n", cog, *cog);` . This displays the pointer and the pointer content (which is the cog number). Somehow I managed to convince myself that this gives me a better feel for "C" pointers.

Page 188: I don't think you ever explain what how `%4d` works in the print statement. The display makes it clear if you look closely but some folks won't.

Page 212: It says `cog_run(blink, 20);` in two places. Should be 128 (at least that what it says on page 211).