Things I have learned about programming with SimpleIDE:

GOTCHAS! Gotcha (programming), an unexpected or unintuitive, but documented, behavior in a computer system (as opposed to a bug)

00. When using an external power supply rather than the battery box, be sure to use a 7.5 Volt supply (blue connector) and NOT a 9.0 Volt supply. Some 9.0 Volt supplies actually can supply a higher voltage which will cause permanent damage to the Activity Board.

0. Important: updated LEARN folder goes into Documents > SimpleIDE, and NOT Programs>SimpleIde. I made that mistake TWICE!

1. Plug the Activity board into the computer and turn it on BEFORE opening SimpleIDE. If you reverse that order the board may not be recognized. This may be corrected in future versions of SimpleIDE.

2. When you open SimpleIDE, check that the serial port has been chosen in the upper right corner. If it has not been chosen automatically, then choose it yourself, otherwise the program will not download to the activity board.

3. When you want to make a new project: Click Project > New > My Projects> then give your project a name and save it. After that you can enter code, attempt to run it, and then debug it. Be sure to start a new PROJECT (.side file), unless you really do want to be developing a new file within the previous project. If you are in a previous project and write a new .c file there may be complications because of the other files in that project. (Been there, done that!)

4. If you are using an “#Include” directive (which you almost ALWAYS do), you MUST enter it by clicking on the Add Simple Library icon which looks like this: . If you simply type the #Include directive directly into the program it is likely that the necessary path will not be added to the project, causing the compilation process to fail. If you don’t understand what that is all about, just use the icon! This may also be fixed in future versions os SimpleIDE.

5. Library “servo.h” handles the standard servo and the continuous rotation servo. It should not be used with the high speed continuous rotation (HSCR) servos because their potentiometer is not calibrated. This library ignores the EPROM calibration and the encoders, assuming they are not there.

6. Library abdrive.h is for the HSCR servos with encoders. Before using those functions you must complete the calibration routine located at Documents> SimpleIDE> Learn> Examples> ActivityBot> ActivityBotCalibrate.side Those functions make things much easier.

7. If you intend to use the reset button in a program you have written, or if you want to use the brownout beep trick, the program must be loaded into EEPROM. Neither will work if the program is being run from RAM.

8. If you have loaded a program into RAM and then lost power (or browned-out, or pushed the reset button) the program in EEPROM and not the one in ram is what will be executed. This can cause a lot of confusion!

9. If running ActivityBotCalibrate.side does not seem to be working properly, it might be that the batteries are low and the program keeps resetting. Try running with external power (on a long extension!) or installing fresh batteries.

10. The activity board has a 64 K EEPROM. The addresses run from 0 through 65535. If you use the Load EEPROM button on SimpleIDE it copies the program to the board's EEPROM, filling addresses 0 to 32767. Therefore, 32768 is the first available EEPROM memory address for data storage. Some Simple Libraries keep calibration data in the EEPROM. By convention, these libraries use the highest available EEPROM addresses. Libraries that use EEPROM this way list the memory locations they use in their documentation. For example the Activity Bot Calibration routine documentation says that it uses addresses 63418-65470. If your application is logging data, it’s best to start from 32768 and allow data to accumulate upward to higher addresses. If it runs above 63418 it will overwrite the calibration data used for the servos and encoders. You will have to run ActivityBotCalibrate.side to recalibrate your Activity Bot before using those functions.