# Module4 – Propeller C: Simple Circuits

Alternate Wikibooks Text: [C Programming/Simple Input and Output](https://en.wikibooks.org/wiki/C_Programming/Simple_input_and_output)

OK. Let’s have fun this week building some circuits, connecting them to the I/O pins, and watching them work! The first couple are going to seem a little familiar. This group of exercises is meant to get you familiar with the contents of the parts pack and learn how to build circuits on the proto board. But, watch for the various functions that pop up in the programs. Most come from a library of functions called **simpletools.h**. This is part, among others, of the Learn Library that you download with the IDE.

WHILE PROGRAMMING HAS BEEN THE MAIN FOCUS, NOTE, THAT FROM HERE OUT WE WILL BE PLAYING WITH THE HARDWARE IN YOUR KIT. EVERY ITEM EXCEPT RESISTORS HAS AN INFORMATION SHEET ABOUT IT. YOU FIND THESE IN THE PARALLAX STORE (LINK AT THE BOTTOM OF EVERY WEB PAGE). FOR INSTANCE, TO FIND OUT SPECIFIC INFO ABOUT THE HARDWARE IN THE LAST SECTION OF THIS MODULE, THE 7 SEGMENT DISPLAY, YOU GO TO THE STORE, PUT “7 SEGEMENT DISPLAY” IN THE SEARCH, AND CLICK ON ITS LINK. AT THE BOTTOM OF THE PAGE FOR THE 7 SEGMENT DISPLAY YOU WILL FIND LINKS TO DOWN LOADS AND OTHER DATA SHEETS.

WHAT A COMPANY!

## [Blink a Light](http://learn.parallax.com/propeller-c-simple-circuits/blink-light)

You’ve been here before in Week 0. BUT THIS TIME build a circuit using the schematic diagram (in blue) with LEDs and resistors from the parts pack on the protoboard. You pick the color. Use different I/O pins though; say, pin 8 and pin 9. Modify the program to reflect the changes you are making in hardware. Have fun with this…

Insert a picture here and paste in a copy of your program from **Try This** for **Blink a Light**. Don’t forget to comment the program.

Insert a picture here and paste in a copy of your program from **Your Turn** for **Blink a Light**. Don’t forget to comment the program.

## [Check Push-Buttons](http://learn.parallax.com/propeller-c-simple-circuits/check-pushbuttons)

OK, Blink a Light was an OUTPUT function. Now let’s do an INPUT function. Mount the switches shown and make your LEDs light in response to a pushbutton.

Insert a picture here and paste in a copy of your program from **Try This** for **Check Push-Buttons**. Don’t forget to comment the program.

Insert a picture here and paste in a copy of your program from **Your Turn** for **Blink a Light**. Don’t forget to comment the program.

## [Piezo Beep](http://learn.parallax.com/propeller-c-simple-circuits/piezo-beep)

If you didn’t do this exercise in Week 0, give it a go here and learn about the **freqout** function. Adjust some of the parameters to get different effects. If you did this in Week 0, turn in that work here.

Insert a picture here and paste in a copy of your program from **Try This** for **Piezo Beep**. Don’t forget to comment the program.

Insert a picture here and paste in a copy of your program from **Your Turn** for **Piezo Beep**. Don’t forget to comment the program.

## [Measure Volts](http://learn.parallax.com/propeller-c-simple-circuits/measure-volts)

Here’s an INPUT exercise that actually measures volts using the A/D Converter on the board. Definite input limits for voltage are established by the A/D Converter. This Converter responds to voltages from 0 to 5V. Its output (digital value) responds to this 0 to 5V range with digital values from 0 to 4096. The **adc\_ function** calculates the actual value of voltage input from this digital value.

Insert a picture here and paste in a copy of your program from **Try This** for **Measure Volts**. Don’t forget to comment the program.

Insert a picture here and paste in a copy of your program from **Your Turn** for **Measure Volts**. Don’t forget to comment the program.

## [Set Volts](http://learn.parallax.com/propeller-c-simple-circuits/set-volts)

The reverse of A/D conversion is D/A conversion. The microcontroller can send a desired voltage value to the outside world using the D/A converter (near the audio jack). If we feed its output back into the A/D converter, we can monitor and verify the desired voltage.

Insert a picture here and paste in a copy of your program from **Try This** for **Set Volts**. Don’t forget to comment the program.

Insert a picture here and paste in a copy of your program from **Your Turn** for **Set Volts**. Don’t forget to comment the program.

## [Sense Light](http://learn.parallax.com/propeller-c-simple-circuits/sense-light)

This is your first real “sensor”. Applying electrical energy to a silicon PN junction causes the junction to conduct, but light can perform the same function.

Insert a picture here and paste in a copy of your program from **Try This** for **Sense Light**. Don’t forget to comment the program.

Insert a picture here and paste in a copy of your program from **Your Turn** for **Sense Light**. Don’t forget to comment the program.

## [Seven Segment Display](http://learn.parallax.com/propeller-c-simple-circuits/seven-segment-display)

Now we arrange 7 LEDs into something that looks like the number 8 and we can turn them on and off to get representations of the other 9 digits. You will need 7 output pins to perform this exercise. If you have taken a digital class in the electronics program, you are familiar with the 7447 Seven-Segment BCD decoder. We’re having the Propeller perform this function here.

Insert a picture here and paste in a copy of your program from **Try This** for **Seven Segment Display**. Don’t forget to comment the program.

Insert a picture here and paste in a copy of your program from **Your Turn** for **Seven Segment Display**. Don’t forget to comment the program.