CHAPTER 2 – ORGANIZATION

It is very important to keep in mind that the Arduino will repeat all of these functions hundreds or thousands of times per second.

> Add stronger statement that loop() must be designed to repeat very fast – no blocking code.

CHAPTER 3 – SERVO

We need to add a reference to the servo library at the top of the code

> Add explanation that library is pre-written code that can be used in programs to simplify common tasks.

As the range of angles is 0 to 180 a byte will be sufficient.

> Add "… since a byte can hold values from 0 to 255".

It is much better to define a variable (such as servoMin) rather than just include the number (20 in this case) directly in your code.

> "… define a named constant (such as servoMin) …"

We can use the delay() function in setup() because it will only happen once and we won't need anything else to happen at the same time.

> Add statement that delay() should be avoided in loop().

CHAPTER 4 – POTENTIOMETER

The ADC gives us a 10bit value with a maximum value of 1023 so that won't fit into a byte and we need an int.

> Add "… since an int can hold values from -32,768 to +32,767".

to tell the compiler to temporarily use long variables while doing its sums.

> Add "… since a long int can hold values of up to plus or minus 2 million".

CHAPTER 5 – LEDs

The delay() function holds up everything until it is finished.

> Add "In our example, it would prevent the potentiometer being read, the servo position being changed and the push buttons being read."

CHAPTER 6 - SWITCH BUTTONS

button0state = digitalRead(button0in);

In the spirit of encapsulating detail in functions, how about changing the variable to button0pressed and this statement to button0pressed = !digitalRead(button0in)? This would make the code in setFlashPeriod() clearer.

CHAPTER 7 - QUESTION FOR USER

byte buffSize = 31;

In the first code fragment, need to correct to const byte buffSize = 31.