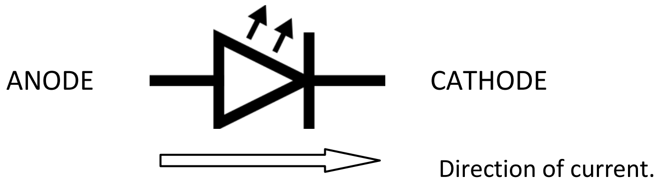
Electronics Merit Badge Kit
Theory of Operation

This is an explanation of how the merit badge kit functions. There are several topics worthy of discussion. These are:

1. LED operation.
2. Resistor function in this design.
3. Switch operation.
4. Microcontroller operation.

LED Operation

The schematic symbol for an LED is:

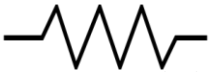


LED stands for light emitting diode. A <u>diode</u> is a more general term that describes an electronic component that allows electrons to flow through it (current) in only 1 direction. Current will only flow from the anode to the cathode. A <u>light emitting diode</u> has an additional characteristic, in that when enough current flows through it, the LED will light up. The amount of light the LED puts out is proportional to how much current flows through it, up to a limit. If too much current flows through it, the LED can overheat and 'burn up'.

It is important to install an LED correctly in the PCB when assembling the kit. If installed backwards, no current will flow through the diode, and the LED will not turn on.

Resistor Operation

The schematic symbol for a resistor is:



Resistor Symbol

A resistor is an electrical component that resists the flow of electrons (current). The units for resistance is ohms. The larger the resistor value, in ohms, the more resistance to electron flow the resistor exhibits.

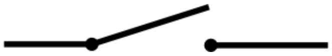
This is important in the operation of the kit, because a resistor is used to control how much current is allowed to flow through the LED. This accomplishes two things in this design:

Firstly, and most obviously, setting the resistor value (in our case, it is set to 200 ohms) will limit how much current can flow through the diode, and consequently how much light the diode will emit.

Secondly, this resistor value will have a direct effect on battery life.   If we used a smaller resistor, then more current would flow through the diode, more light would be emitted, but the batteries would run down more quickly.

Switch Operation

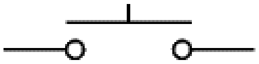
There are two types of switches used in the kit design. The first switch is a SPST, or single pole, single throw switch. The schematic symbol for that is:



SPST Switch

This switch is used to connect the battery to the rest of the circuit. It is the on/off switch. If the switch is in the off position, there is no connection from the battery to the circuit board components. Hence, the circuit is off.   Note that this type of switch is used in many places, though the size and amount of current the switch can handle varies. A good example of general use of this switch type is the wall switch that turns on the lights in a room.

The second type of switch used in this design is the momentary, push button switch. The schematic symbol for this switch is:



Momentary Switch

There are 2 of these switches used in this design. One switch is used as a 'START' button, and the second is used as a 'MODE' button. The operation of this switch is fairly straight forward. Pressing the button down will connect the two terminals of the switch, while releasing the switch disconnects the two terminals.

Microcontroller Operation

The microcontroller is a small computer chip that can be programmed to perform a specific function. In the case of this merit badge kit, the microcontroller (micro) waits for the 'START' or 'MODE' button press, and controls the LED display accordingly.

Several display modes are available, but we will only discuss in detail the single LED mode (available when the circuit is powered up, before the 'MODE' button is pressed), and discuss briefly the two LED mode. At the end of this document, you can find a table that lists out all the modes programmed into this kit.

First, let's discuss the technique used to turn on an LED. For this design, there are 12 LEDs that can be lit. The way the circuit is designed, and the way the software has been written, only 1 LED is turned on at any moment. This is necessary for those LEDs 1-8 and 12, since all these LEDs share 1 current limiting resistor. Since all these LEDs share a single resistor, if 2 LEDs where turned on at the same time, their light intensity would go down, because total current is determined by the resistor. This would basically allow only half the amount of current to go through each of these 2 LEDs, and their intensity would be dramatically less. So, for this circuit's operation, only 1 LED may be turned on at a time.

LED Multiplexing

I use a simple technique called multiplexing when lighting the LEDs. Let's use the case when we want to show 2 LEDs on at the same time (the 2 LED mode). Since we are only allowed to turn on 1 LED at a time by virtue of the circuit design, we turn one LED on for a short time, and then turn the second one on for a short time, and we repeat this sequence very quickly. Even though we are turning 1 LED on and then off, and turning the other LED on and then off, to our eyes it appears that both LEDs are turned on at the same time. This technique can be extended so that all 12 LEDs can appear to be turned on at the same time. This is a very popular technique used on many displays.

A benefit of using this technique is that it reduces the number of components used for this design, and more importantly, it reduces how much current we need from the batteries, which makes the batteries last longer. Since we are only turning on one LED at a time, we use a minimal amount of current from the battery. Imagine if the circuit was designed such that all 12 LEDs could be turned on simultaneously, we would use the batteries up 12 times faster.

Circuit Operation

So, here is a description of the software that is running in the micro, and how that relates to the circuit operation.

Once the circuit is powered up, a single LED is illuminated, which indicates the micro is ready for an input from one of the switches. If the 'MODE' button is pressed, the micro adjusts itself to change how the LEDs will be controlled. When the 'START' button is pressed, the micro begins displaying the LEDs in the selected sequence. As long as the 'START' button is pressed, the LED spin rate will stay at its maximum rate. Once the 'START' button is released, the micro will begin slowing the spin at a predetermined rate. The randomness of the circuit (which is the last LED lit when the spin cycle is completed?) is determined by how long the 'START' button is pressed.

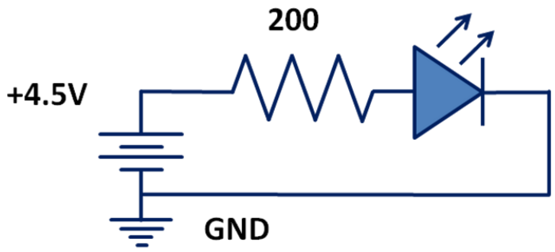
The micro consists of a CPU (central processing unit), memory and I/O.

The CPU executes the command sequences defined by the program I wrote to do this function. This program is written in the 'C' programming language.

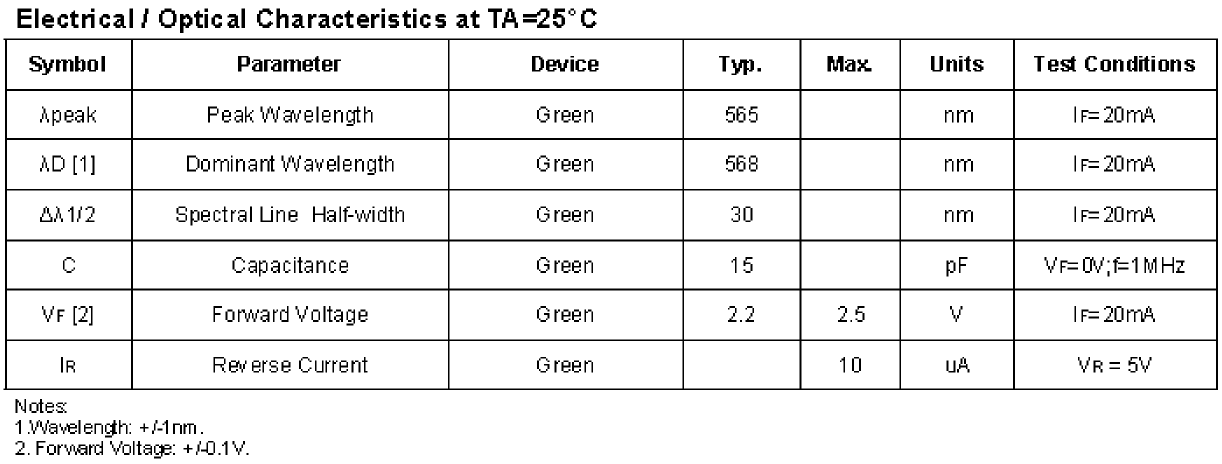
Memory holds temporary values that the program uses while controlling the circuit. An example of a memory item would be which LED is currently being displayed. This would take on the value of 0 to 11 (for 12 LEDs), and would be changed when a different LED is to be displayed.

I/O stands for Input/Output, and is the part of the microcontroller that interfaces to our circuit. There are 2 inputs defined for this design; 'START' and 'MODE'. When one of these buttons is pressed, the voltage that is present at that particular input will be connected to ground (0 volts). When the button is released, the voltage will return to a voltage value that is the same as the battery voltage (approximately 4.5 volts). Logically we can say that when the button is pressed, a logical 0 is presented to the microcontroller pin, and when the button is released, a logical '1' is presented to the pin. The operating program is continuously monitoring these 2 inputs, and can detect when the input changes from a 1 to a 0 and when it changes from a 0 to a 1. The program modifies its operation based on how it interprets these 2 inputs.

There are 12 outputs in this design, each one controlling a single LED. When the output pin is driven to a logical '1', the LED connected to that pin will light up. The following schematic diagram illustrates one LED circuit.



For the purposes of this discussion, we will use a green LED as an example. From a datasheet of a typical green LED:



Electrical / Optical Characteristics at TA=25°C

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λpeak	Peak Wavelength	Green	565		nm	I_F=20mA
λD [1]	Dominant Wavelength	Green	568		nm	I_F=20mA
Δλ1/2	Spectral Line Half-width	Green	30		nm	I_F=20mA
C	Capacitance	Green	15		pF	V_F=0V;f=1MHz
V_F [2]	Forward Voltage	Green	2.2	2.5	V	I_F=20mA
I_R	Reverse Current	Green		10	uA	V_R = 5V

Notes:
1.Wavelength: +/-1nm.
2. Forward Voltage: +/-0.1V.

Of importance from this datasheet, the forward voltage drop across this LED is typically 2.2V, but a maximum of 2.5V. If we assume a maximum voltage drop across the LED, we can calculate how much current will go through the LED in our circuit.

From Ohms LAW: I = V / R. Since V_BATTERY = 4.5V, and V_LED = 2.5V, the current will be:

(4.5-2.5) / 200 = .010 Amps => 10 milliamps.

From this, we can say that the each LED, when lit, sees about 10 milliamps of current. This determines the light intensity of the LED (and as previously discussed, the longevity of the batteries).

So, when the micro sets one of its outputs to logical 1, the LED will turn on, and its intensity will be determined by the 200 ohm resistor. The micro program will control the order of LEDs and timing of LEDs, until the predetermined sequence has concluded.

I will make a final comment about R2-R4, which are individual 200 ohm resistors dedicated to a single micro output and LED. These separate circuits are identical in function to the other LEDs that share a single resistor, but the separation is necessary when I want to write a new control program into the micro. There are 4 special pins used when writing a new program, three of which are also used to control LEDs.

Also, R5, which is a 56K resistor, connects to a special pin on the micro that has no functional purposes when operating the circuit, but does have a purpose when operating the micro in a special debug mode, which is used when writing the program.

LED Modes

Currently, in the code that controls the kit, there are 10 different display modes available. Pressing the mode button will increment the selection, according to the table below:

	Mode	**Description**
1	single_led_forward	A single led will run in a clockwise direction
2	single_led_reverse	A single led will run in a counterclockwise direction
3	double_led_forward	Two adjacent leds will run in a clockwise direction
4	double_led_reverse	Two adjacent leds will run in a counterclockwise direction
5	double_led_bidirectional	Two leds, starting at the same place on the circle, each run in opposite directions
6	double_led_chase	Two leds, on opposite sides of the circle, run in the same direction
7	double_led_forward_halfspeed	Two leds, starting at the same location on the circle, run in the same direction, with one running at full speed and the other running at half speed
8	single_led_random	A single led running in a random pattern
9	double_led_forward_random	Two leds, one running in a random pattern, and the second running in a clockwise direction
10	double_led_forward_reverse_half	Two leds running in opposite directions, one at full speed and one at half speed

After mode 10 is selected, pressing the mode button again will restart the selection at mode 1.