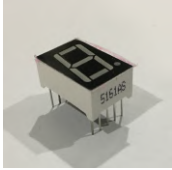
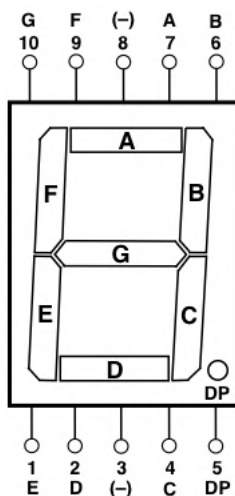


**Unit 2:
Displays****Lesson:
The 1-Digit 7-Segment Display (SSD)****Lesson No.
2-02****Figure 1****Lesson Objectives:**

In this lesson, students will:

- learn how a 1-digit 7-segment display (SSD) works (Figure 1).
- Learn how to connect the display to an Arduino micro-controller.
- write an Arduino sketch to create a digital counter that continuously increments each second from "0" to "9."

**Figure 2**

7-segment displays consist of 7 LEDs, called segments, arranged in the shape of an "8." Most 7-segment displays actually have eight segments, with a dot on the bottom right side of the digit that serves as a decimal point. Each segment is named with a letter A to C and DP for the decimal point.

Each segment on the display can be controlled individually, just like a regular LED (light-emitting diode).

1-digit 7-segment displays typically have 10 pins (5 on each end). Two pins connect to ground (-), and the other eight pins connect to each of the remaining pins for A-C and the decimal point segment DP (Figure 1 and 2). (Note: Refer to the specific manufacturer for their pin locations/connections).

		Digit to Display									
		0	1	2	3	4	5	6	7	8	9
Segment On/Off Status	A	on	off	on	on	off	on	on	on	on	on
	B	on	on	on	on	on	off	off	on	on	on
	C	on	on	off	on	on	on	on	on	on	on
	D	on	off	on	on	off	on	on	off	on	off
	E	on	off	on	off	off	off	on	off	on	off
	F	on	off	off	off	on	on	on	off	on	on
	G	off	off	on	on	on	on	on	off	on	on
	DP	off	off	off	off	off	off	off	off	off	off
		0	1	2	3	4	5	6	7	8	9

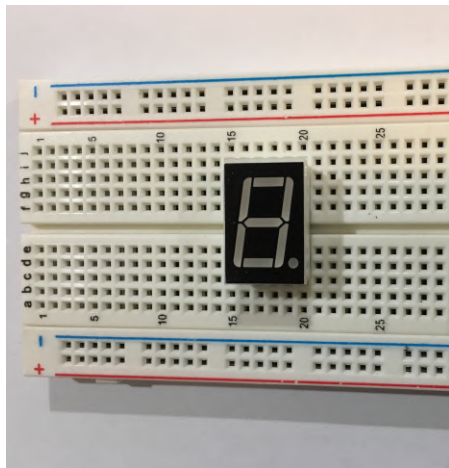
Figure 3

Digits 0–9 are displayed by turning specific segments ON or OFF. The segments that are lit form the shape of the digits.

In Figure 3, the On/Off status of the eight segments (A-G and the DP) is provided for each digit to be displayed. The resultant digit shape is represented by the large bold-faced digit at the bottom of each column.

CONNECTING A 1-Digit 7-SEGMENT DISPLAY**Table 1**

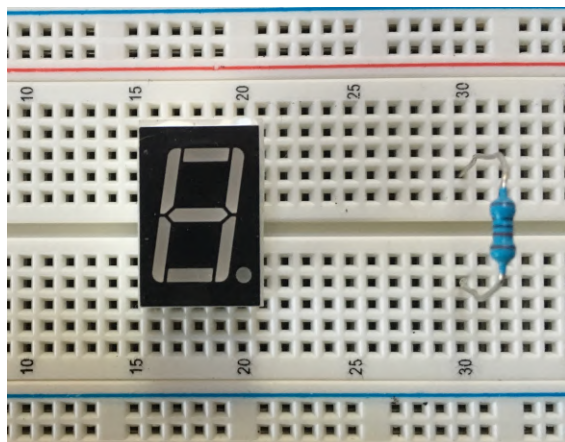
Components and Equipment	Quantity	Possible Source
Arduino Uno R3 Microcontroller A00006 or similar such as the Elegoo EL-CB-001 Uno R3 Board	1	Amazon
Solderless breadboard: Elegoo EL-CP-003 MB-102 (830 point) or similar	1	Amazon
1-Digit 7-Segment Display: Microtivity IS111 or similar	1	Amazon
1k ohm resistor	1	Amazon
Jumper wires (male to male): Elegoo-CP-004 or similar	13	Amazon

**Figure 4.****Step 1:**

Mount the 1-Digit 7-Segment Display on a solderless breadboard as shown in Figure 4.

Note that the Display is mounted in such a way that the decimal point appears at the bottom right side.

Press the 5 pins on each side into the breadboard sockets being careful not to bend the pins.

**Figure 5.****Step 2:**

Mount a 1K ohm resistor on the breadboard by shaping the resistor wire leads in a "U" shape and carefully pressing them in sockets as shown in Figure 5.

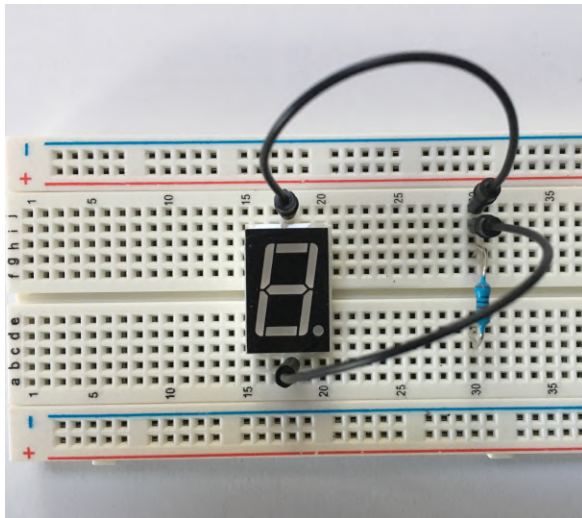


Figure 6.

Step 3:

Connect two jumper wires in line with one side of the 1K resistor as shown in Figure 6. Connect each of the other jumper wire ends to the ground pins

(-) on the Display. These are the center pins on each end (refer to Figure 2: Pin numbers 3 and 8).

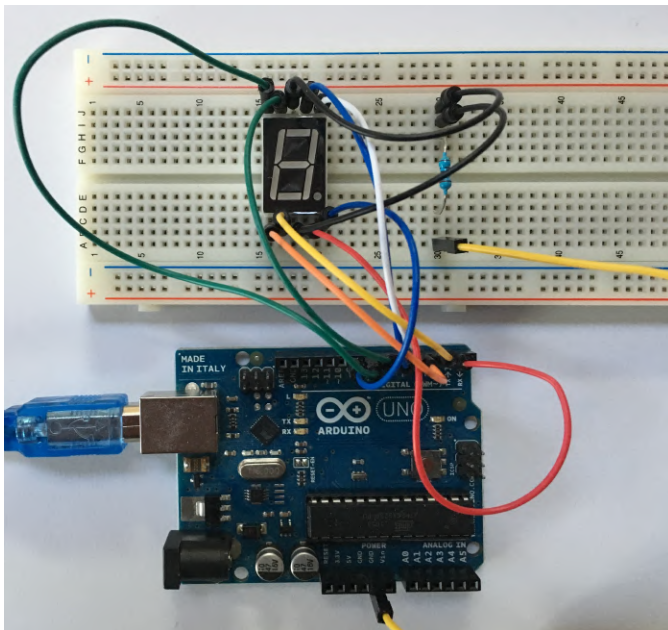


Figure 7.

Step 4:

Connect jumper wires to the Arduino micro-controller as listed in Table 2 below referring to Figure 2 for the Segment pin locations

Table 2

Segment Pin	Arduino Pin
A - 7	6
B - 6	5
C - 4	2
D - 2	3
E - 1	4
F - 9	7
G - 10	8
DP - 9	9

Step 5.

Connect the Arduino board to a computer with the Arduino software loaded.

Step 6

Open the Arduino program and type the following code to create an Arduino “sketch” that creates a digital counter that continuously increments each second from “0” to “9.”

Make sure that the “SevSeg.h” is an available library to include.

Code to type:

```
#include "SevSeg.h"
SevSeg sevseg;
void setup(){
}
void loop()
{
    byte numDigits = 1;
    byte digitPins[] = {};
    byte segmentPins[] = {6, 5, 2, 3, 4, 7, 8, 9};
    bool resistorsOnSegments = true;

    byte hardwareConfig = COMMON_CATHODE;
    sevseg.begin(hardwareConfig, numDigits, digitPins, segmentPins, resistorsOnSegments);
    sevseg.setBrightness(90);

    sevseg.setNumber(1);
    sevseg.refreshDisplay();
    delay(1000);

    sevseg.setNumber(2);
    sevseg.refreshDisplay();
    delay(1000);

    sevseg.setNumber(3);
    sevseg.refreshDisplay();
    delay(1000);

    sevseg.setNumber(4);
    sevseg.refreshDisplay();
    delay(1000);

    sevseg.setNumber(5);
    sevseg.refreshDisplay();
    delay(1000);
```

```
sevseg.setNumber(6);
sevseg.refreshDisplay();
delay(1000);

sevseg.setNumber(7);
sevseg.refreshDisplay();
delay(1000);

sevseg.setNumber(8);
sevseg.refreshDisplay();
delay(1000);

sevseg.setNumber(9);
sevseg.refreshDisplay();
delay(1000);

sevseg.setNumber(0);
sevseg.refreshDisplay();
delay(1000);
}
```

Save this sketch as “Single_7-Segment_Counter

Final Note:

If your wiring and programming is correct, the display should first begin displaying a “0” and counts up by one to “9” then loops back to “0” and repeats counting up. Each count is one second long (1000 milliseconds in the code).

Extended Tasks:

1. How would the code need to be changed to count down from “9” to “0”?
2. How would the code be changed so that the digits would change every 3 seconds?
3. Does it matter which end of the 1K ohm resistor faces (i.e., which end gets connected to ground)?
4. Try shortening the sketch by eliminating nine of the three-line groupings:
sevseg.setNumber(0);
sevseg.refreshDisplay();
delay(1000);

Rewrite the code to increment the number to be displayed by 1 in one set of lines rather than Repeating the set of lines ten times.