HANDS-ON LAB INSTRUCTION SHEET – Learning Kit MODULE 4

CONTROLLING ON-BOARD AND EXTERNAL LEDS

NOTES:
If you did not finish Modules 1 through 3, be sure to finish them NOW *before starting this Module or you will fall behind the rest of the class.* ***Labs MUST be done in order.***

BILL OF MATERIALS

(1) **Arduino UNO R3 Microcontroller & USB Cable**
(1) **RED** Light-Emitting Diode (LED)
(1) **GREEN** Light-Emitting Diode (LED)*
(1) **YELLOW** Light-Emitting Diode (LED) *
(1) **1KΩ (1000 Ohm)** Resistor (*actually any value **220Ω** through **1KΩ***)
*(*We could actually use 3 @ RED LEDs or any other colors!)*

BACKGROUND AND REVIEW

DIGITAL I/O FUNCTIONS IN "BLINK"

These I/O commands are placed inside the braces for the setup or loop functions.
In the Setup function we include one-time instructions that do not change,
For example:

pinMode(pin#,state);

where state = INPUT or OUTPUT
Configures the specified pin to behave either as an input or an output
Syntax: **pinMode(pin, mode);**
Example: pinMode(13, OUTPUT); // sets Digital I/O Pin #13 to output mode

In the Loop function we have instructions which are repeated over and over:

digitalWrite(pin#,state);

where state = HIGH (=ON) or LOW (=OFF)
Writes a HIGH or a LOW value to a digital pin
If set to OUTPUT with pinMode() then 5V for HIGH, 0V (ground) for LOW.
Syntax: **digitalWrite(pin, value);**
Example 1: digitalWrite(13, HIGH); // +5 volts to I/O pin 13
Example 2: digitalWrite(13, LOW); // I/O pin 13 connects to Ground

delay(#);

*where **#** is in milliseconds (1000ms = 1 second)*

Note:
1) ALL Arduino functions are ***CASE-SENSITIVE!***
2) ALL Arduino functions ***MUST*** end in a semicolon "**;**".

Revision 211021: This document is available electronically at:
http://www.CharlesRubenstein.com/222/**04Lab_LK.pdf**

THE BLINK SKETCH

In Module 3 you set up the Arduino IDE on your computer, installed the various example sketches from the USKcode or ElegooCode zip files, and then loaded the BLINK example sketch onto BOTH your IDE and your Arduino board. The BLINK program code is accessed within the IDE by clicking: **File → Examples → 01.Basics → Blink**

Project 3.0: BLINKING THE ONBOARD LED

Let's review each line of the BLINK code in each of the three sections of the sketch - Header, Setup, and Loop. (*Note Line Numbers and Section names have been added here but are not in the actual code file…)*

THE BLINK SKETCH EXAMPLE *(Section and line #s added)*

```
                              /Section 1: HEADER & COMMENTS:/
Line 01:  // Blink 1.0: Turn on-Board LED (#13) ON
Line 02:  // 1 second and then OFF for 1 second, Repeat
                                        /Section 2: SETUP:/
Line 03:  void setup() // Runs only once
Line 04:  {           // open block of code
Line 05:  pinMode(13, OUTPUT); // Pin 13 = Output
Line 06:  }           // close block of code
                                    /Secion 3: LOOP Function:/
Line 07:  void loop() // function runs and then returns here
Line 08:  {         // open block of code
Line 09:  digitalWrite(13, HIGH); // Turns ON the "L" LED
Line 10:  delay(1000); // Wait for 1000 milliseconds (1 second)
Line 11:  digitalWrite(13, LOW);     // Turns OFF the "L" LED
Line 12:  delay(1000);  // Wait for 1000 milliseconds (1 second)
Line 13:  } // closes block of code and returns to top of loop()
```

NOTE: *Setting our digital I/O to be **pin #13** we can check out the BLINK code **without** using any additional components by looking at **L –** the Arduino's on-board LED.*

Project 4A: BLINKING THE ONBOARD LED AND AN EXTERNAL LED

Now that the onboard **L** LED is flashing once per second, we can add an external LED to show that digital I/O port #13 is actually turning ON and OFF at the same time.

First disconnect the Arduino Board from the USB port to remove all power from the board. Connect a wire from the +5 volt power bus and ground to the breadboard.

Connect Digital I/O **pin #13** on the Arduino board to an external LED with a 1000 ohm series resistor to ground. *The external LED must have a series resistor with a value between 220 and 1000 Ohms to limit the current flowing through it. The maximum output for each port is 40 mA. If we have a 5 volt supply,*

Ohm's Law tell us the lowest resistor value we can use is:

R = V/I = 5v / 0.04A = 125 Ohms

However, typical LED currents are in the range of up to 20 mA will require that we use a higher valued resistor.

Revision 211021: This document is available electronically at:
http://www.CharlesRubenstein.com/222/**04Lab_LK.pdf**

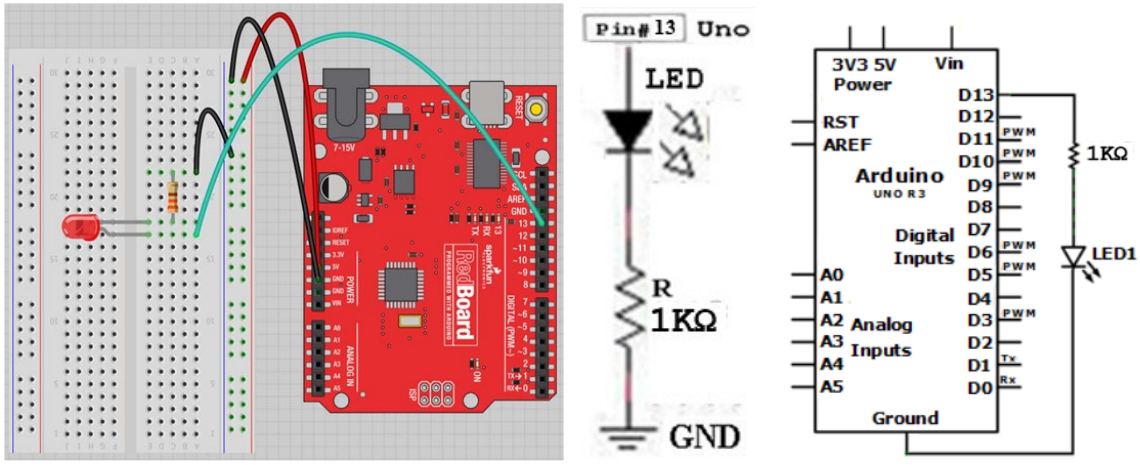

Figure 4.1 Pictorial Diagram, Schematic, and Block Diagram

Reconnect the Arduino Board to the USB cable and you should immediately note that the LED Connected to digital I/O pin #13 blinks at the same time as the onboard "**L**" LED. This is due to the fact that although we removed power from the Arduino Board, it has stored the Blink code in non-volatile memory and is merely running the program already uploaded to it.

You can check to see that no other digital I/O port is actually turning ON/OFF by taking the wire in I/O #13 and putting it into any of the other digital I/O ports.

QUESTION 4A.1: What is the value to have the output current limited to 20mA?
4A.1 ANS: _____

QUESTION 4A.2: What is the value of current for the 1000 Ohm resistor we are using?
4A.2 ANS: _____

ACTIVITY 4A.3: Take a photo of your wiring

ACTIVITY 4A.4: Take a short video named "**4A**" show the onboard and external LED flashing.

MODIFYING THE SKETCH CODE

The next projects show a few options for increasing the usability of the code by changing the on/off duty cycle and by increasing the number of output devices the Arduino is controlling.

IMPORTANT NOTE:
Changing the Sketch code on the IDE does NOT change the program stored in the microcontroller!

*The **THREE** steps required to change the controller's code are:*

1. ***Change and Save the code,***
2. ***Verify/compile the code, and then***
3. ***Upload the code to see the results…***

Revision 211021: This document is available electronically at:
http://www.CharlesRubenstein.com/222/**04Lab_LK.pdf**

Project 4B: BLINKING AN EXTERNAL LED

Now that you saw we can add an external LED at pin #13, change the code to another output pin, e.g., #10, and show you can control the external LED without having the **L** LED blink. There are two ways to do this:

You can revise the Blink code changing all code lines (05, 09, and 11) that define the output from digital I/O pin #13 to read digital I/O pin #10:

ORIGINAL CODE LINES:
Line 05: **pinMode(13, OUTPUT);** // Pin 13 = Output
Line 09: **digitalWrite(13, HIGH);** // Turns ON the "L" LED and an external LED @ I/O #13
Line 11: **digitalWrite(13, LOW);** // Turns OFF the "L" LED and an external LED @ I/O #13

MODIFIED CODE LINES:
Line 05: **pinMode(10, OUTPUT);** // Pin 10 = Output
Line 09: **digitalWrite(10, HIGH);** // Turns ON an external LED @ I/O Pin #10
Line 11: **digitalWrite(10, LOW);** // Turns OFF an external LED @ I/O Pin #10

Make this code change. Then:
ACTIVITY 4B.1: Verify and upload the new sketch and save the code as **Blink10**

ACTIVITY 4B.2: In a short video clip named "**4B**" show that only the **external** LED is flashing.

QUESTION 4B.3: Is the external LED connected to Digital I/O Pin #10 flashing?
4B.1 ANS: _____ YES or _____ NO

QUESTION 4B.4: Is the onboard **L** LED flashing?
4B.2 ANS: _____ YES or _____ NO

DEFINING NAMED VARIABLES: INTEGERS

Instead of changing each line in the sketch when we want to change an input or output port, we can define one (or more) named variables and place those definitions in the Header section of the sketch, prior to the setup function (between lines #02 and #03 in the Blink Code). Here we can define a digital I/O pin variable named '**led**' and after changing the digital I/O pin **#13** to **led** in the setup and loop function lines (05, 09, and 11) we need only set **led** equal to digital I/O pin **#10** as an integer named **led**.
We can define a variable to replace the exact number of any input or output by setting its [name] to the I/O port [value]. The syntax for this is: **int [name] = [value];**
In our case initially we can define **digital I/O Pin #13** as **led** using the code **int led = 13;**
Now, whenever the compiler sees '**led**' it would substitute '**13**' for the value. Should we decide to use another I/O port, say #**5**; then we only have to change the definition once in the header and it will be used throughout the sketch. It is important to recall that named variables are case-sensitive. Thus, the variable **led** is NOT the same as **Led** or **LED**.

ADDING INTEGER VARIABLES:
ORIGINAL CODE:
Line 05: **pinMode(13, OUTPUT);** // Pin 13 = Output
Line 09: **digitalWrite(13, HIGH);** // Turns ON the "L" LED
Line 11: **digitalWrite(13, LOW);** // Turns OFF the "L" LED

Revision 211021: This document is available electronically at:
http://www.CharlesRubenstein.com/222/**04Lab_LK.pdf**

MODIFIED CODE:
LINE 02"a": **int led = 13;** // Variable led is assigned to I/O Port #13
Line 05: **pinMode(led, OUTPUT);** // "led" = Output
Line 09: **digitalWrite(led, HIGH);** // Turns ON I/O Pin "led"
Line 11: **digitalWrite(led, LOW);** // Turns OFF I/O Pin "led"

This code will flash the onboard **L** LED and an LED connected to I/O Port #13.
To change the I/O Port used from #13 to #10 now requires changing only LINE 02"a"
FROM: **int led = 13;** // Variable led is assigned to I/O Port #13
TO: **int led = 10;** // Variable led is assigned to I/O Port #10
ACTIVITY 4B.5: Verify and upload the new sketch and save the code as **BlinkLED**

Project 4C: BLINKING TWO EXTERNAL LEDS AT SAME TIME

Blinking the onboard **L** LED (Port #13) ***and*** an external LED @ Port #10 simultaneously is also possible. As is changing the code to blink the onboard, an external LED @ Port #13 as well as an external LED at Port #10.
We are starting here with the code "**BlinkLED**" which has defined one variable, **led** and has it assigned to Port #10. We need to add another I/O Port definition in order to blink two Ports simultaneously. Let's rename led to be **led13** and define a new port **led10**:
MODIFIED CODE:
LINE 02"a": **int led10 = 10;** // Variable assigned to I/O Port #10
LINE 02"b": **int led13 = 13;** // Variable assigned to I/O Port #13 & **L** LED
Now whenever the compiler sees **led10** or **led13** it will make the appropriate substitution.
To turn both external LEDs ON and OFF simultaneously, revise the BlinkLED code further:
CURRENT CODE:
Line 05: **pinMode(led, OUTPUT);** // "led" = Output
Line 09: **digitalWrite(led, HIGH);** // Turns ON I/O Pin "led"
Line 11: **digitalWrite(led, LOW);** // Turns OFF I/O Pin "led"
MODIFIED CODE:
Line 05a: **pinMode(led10, OUTPUT);** // "10" = Output
Line 05b: **pinMode(led13, OUTPUT);** // "13" = Output
Line 09a: **digitalWrite(led10, HIGH);** // Turns ON I/O Pin "10"
Line 09b: **digitalWrite(led13, HIGH);** // Turns ON I/O Pin "13"
Line 11a: **digitalWrite(led10, LOW);** // Turns OFF I/O Pin "10"
Line 11b: **digitalWrite(led13, LOW);** // Turns OFF I/O Pin "13"
ACTIVITY 4C.1: Verify and upload the new sketch and save the code as **BlinkTWO**
ACTIVITY 4C.2: Take a short video named "**4C**" of the flashing LED pair

Project 4D: ALTERNATELY BLINKING TWO EXTERNAL LEDS

To turn the external LEDs ON and OFF ***alternately***, revise the BlinkLED code further:
MODIFIED CODE:
Line 05a: **pinMode(led10, OUTPUT);** // "10" = Output
Line 05b: **pinMode(led13, OUTPUT);** // "13" = Output
Line 09a: **digitalWrite(led10, HIGH);** // Turns ON I/O Pin "10"
Line 09b: **digitalWrite(led13, LOW);** // Turns OFF I/O Pin "13"
Line 11a: **digitalWrite(led10, LOW);** // Turns OFF I/O Pin "10"
Line 11b: **digitalWrite(led13, HIGH);** // Turns ON I/O Pin "13"
ACTIVITY 4D.1: Verify and upload the new sketch and save the code as **BlinkALT**
ACTIVITY 4D.2: Take a short video named "**B9**" of the alternately flashing LED pair

Revision 211021: This document is available electronically at:
http://www.CharlesRubenstein.com/222/**04Lab_LK.pdf**

Project 4E: BLINKING EXTERNAL LEDS FASTER

To turn the external LEDs ON and OFF ***alternately***, but with a 200 millisecond ON and Off cycle, we need only change the values of our delay code in lines #10 and #12:

ORIGINAL CODE:
Line 10: **delay(1000);** // Wait for 1000 milliseconds *(1 second)*
Line 12: **delay(1000);** // Wait for 1000 milliseconds *(1 second)*
MODIFIED CODE:
Line 10: **delay(200);** // Wait for 200 milliseconds *(0.2 seconds)*
Line 12: **delay(200);** // Wait for 200 milliseconds *(0.2 seconds)*
ACTIVITY 4E.1: Verify and upload the new sketch and save the code as **Blink200**
ACTIVITY 4E.2: Take a short video named "**4E**" of the quickly flashing LED pair

Project 4F: DETERMINING THE SPEED OF BLINK

In this lab, you will see determine the fastest blink rate of an LED that the human eye can still detect: Start with a 20 millisecond ON and Off cycle by changing the values of our delay code in lines #10 and #12:

MODIFIED CODE:
Line 10: **delay(20);** // Wait for 20 milliseconds *(0.02 seconds)*
Line 12: **delay(20);** // Wait for 20 milliseconds *(0.02 seconds)*
Lower the value of ON/OFF times until you can not see any change in ON/OFF.
ACTIVITY 4F.1: Verify and upload the new sketch and save the code as **BlinkFast**
QUESTION 4F.2: Is the fastest rate when we use a 1 ms delay? A 5 ms delay? A 10 ms delay?
4F.2Ans: _____ milliseconds

Project 4G: BLINKING 'HEART' LEDS

In this lab, you will see how we can alter the ON/OFF cycles of LEDs to simulate the 'lub-dub' beating of your heart. Disconnect the I/O Port #10 external LED as we will only be paying attention to the activity of the I/O Port #13 LED, and make the following changes to the values of our delay code in lines #10 and #12 and ADD several additional lines (12b-12e):

ORIGINAL CODE:
Line 10: **delay(1000);** // Wait for 1000 milliseconds *(1 second)*
Line 12: **delay(1000);** // Wait for 1000 milliseconds *(1 second)*
MODIFIED CODE:
Line 10: delay(500); // wait for **500** milliseconds
Line 12a: delay(100); // wait for **100** milliseconds

AND ADD NEW CODE:
Line 12b: digitalWrite(13,HIGH); // set the "L" LED ON
Line 12c: delay(1000); // wait for **one second**
Line 12d: digitalWrite(13, LOW); // set the "L" LED OFF
Line 12e: delay(100); // wait for **100** milliseconds

ACTIVITY 4G.1: Verify and upload the new sketch and save the code as **BlinkHeart**
ACTIVITY 4G.2: Take a short video named "**4G**" of the Lub-Dub flashing LED

Revision 211021: This document is available electronically at:
http://www.CharlesRubenstein.com/222/**04Lab_LK.pdf**