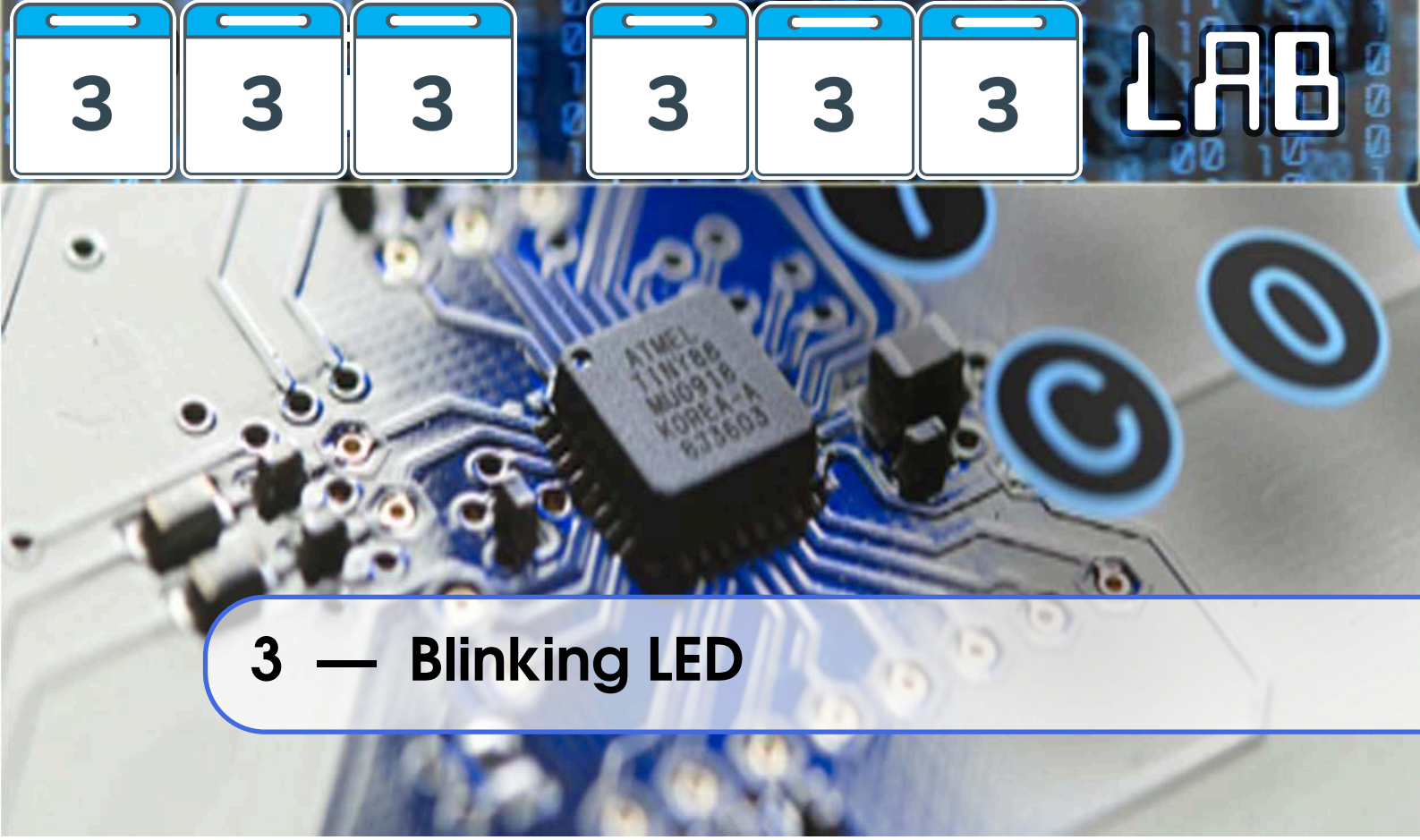

3 — Blinking LED

LAB Objectives

The aim of this LAB experiment is to teach students how to interface and control simple digital output devices such as an LED. This experiment shows the simplest digital output control ; blinking an LED.

3.1 Hardware Required

1. Arduino Uno Board
2. Breadboard
3. Red LED
4. Resistor

3.2 Circuit

The circuit to be implemented in this experiment is a simple interface of an LED to pin 13 of the Arduino board. The schematic of the circuit is shown in Figure 3.1. The implementation of the LED interface using Virtual Breadboard software is show in Figure 3.2.

Calculating The Resistor Value :

To decide on the resistor value to protect the LED, we use the following formula :

$$R = \frac{V_S - V_F}{I_F} \tag{3.1}$$

Assume that the LED is a standard one with forward current $I_F = 10mA$ and a forward voltage $V_F = 2V$. Since the source voltage is $V_S = 5V$ and the LED ends at ground, we can find the value of the resistor as follows : $R = (5 - 2)/(10 \times 10^{-3}) = 300\Omega$.

Building the Circuit :

To build this circuit follow the steps discussed in Experiments 1 and 2 to place and connect the following components :

1. the *VBBExpress* breadboard component,
2. the *Arduino Uno Board*,
3. a *Resistor*, and
4. an *LED*.

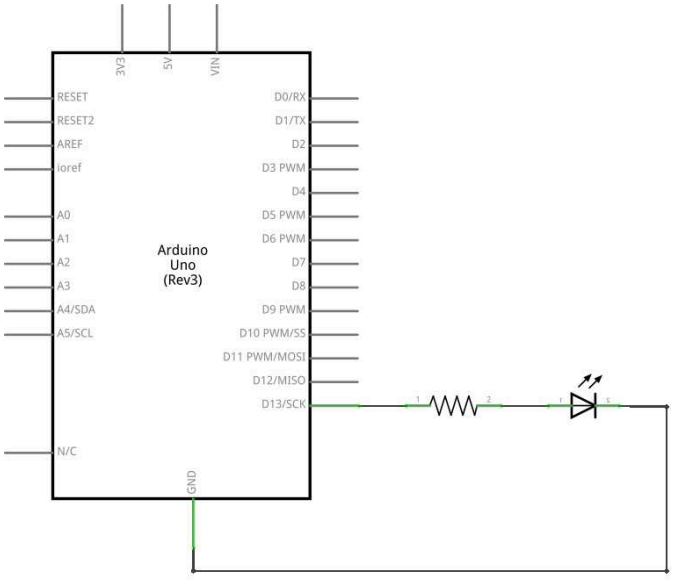

Figure 3.1 – Schematic of the Blinking circuit.

3.3 Program

In this part, you will write a program to blink the LED. Your program should turn ON the LED connected to pin 13 for one second, then turn it OFF for one second, and repeat this process indefinitely.

Writing The Code :

To accomplish this, follow the steps described in Experiment 2 to write the code shown in Program 3.1.

Discussing The Code :

In this program, the first thing you do is to initialize pin 13 as an output pin with the command : pinMode(13, OUTPUT). In the main loop, you turn the LED ON with the command : digitalWrite(13, HIGH). This supplies $5V$ to pin 13. That creates a voltage difference across the pins of the LED, and lights it up. Then you turn it OFF with the

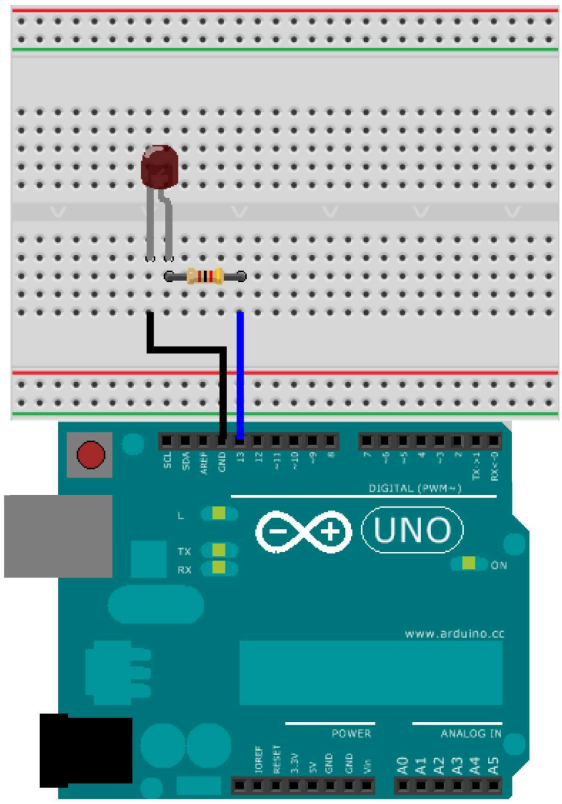

Figure 3.2 – Blinking LED circuit.

command : digitalWrite(13, LOW). That takes pin 13 back to $0V$, and turns the LED OFF. In between the ON and the OFF, you want enough time for a person to see the change, so the delay() commands tell the Arduino to do nothing for 1000 milliseconds, or one second. Note that when you use the delay() command, nothing else happens for that amount of time.

Program 3.1 Blinking LED program.

```
//Turns on an LED on for one second, then off for one second, repeatedly.
import muvium.compatibility.arduino.*;
public class BlinkingLED extends Arduino{
    // give pin 13 a name (say led)
    int led = 13;
    // the setup routine runs once when you press reset:
    void setup() {
       // initialize the digital pin as an output
       pinMode(led, OUTPUT);
    }
    // the loop routine runs over and over again forever:
    void loop() {
       digitalWrite(led, HIGH);   // turn the LED on
       delay(1000);               // wait for a second
       digitalWrite(led, LOW);    // turn the LED off
       delay(1000);               // wait for a second
    }
}
```

3.4 Circuit Emulation

To validate your design you need first to build the source code and then run the emulator. On the *Debug Toolbar*, locate and click the *Build* button to build your code. If there are no errors in the compilation process, emulate the your program by clicking the run button located on the *Application Toolbar*. Observe the blinking action of the LED. Make sure that the LED is ON for approximately one second then OFF for the same duration repeatedly.

Exercise 3.1 Modify the circuit in Figure 3.2 and Program 3.1 to implement a blinking LED wave. The modified circuit should include 3 LEDs that are operated as follows :

1. Turn ON LED 1 for half a second.
2. Turn OFF LED 1.
3. Turn ON LED 2 for half a second.
4. Turn OFF LED 2.
5. Turn ON LED 3 for half a second.
6. Turn OFF LED 3.
7. Repeat steps $1-6$ indefinitely. ■