


PHY 440 LAB 11 – ARDUINO UNO R3

Arduino microcontrollers run simple programs called “sketches.” The microcontrollers can interface with hardware components using a variety of input and output (I/O) channels on the microcontroller board.

Configuring Arduino

The first step is to ensure that your Arduino software is configured for your board. Open the Arduino software on the computer and connect the board to your computer with the USB cable. On the computer, choose **Tools > Board > Arduino/Genuino Uno**. Then choose, **Tools > Port** and select the **COM** with the highest number, which may also have **Arduino/Genuino Uno** in parentheses. If this fails, you will know it because you will not be able to upload any sketches. Then try a different **Port** number. Once your Arduino software is configured, you can upload sketches (code) to your board. After a sketch is uploaded, the Arduino Uno R3 will run that sketch whenever powered up. Sketches tend to be loops of indefinite duration. Power may be supplied over USB or from a battery.

Expt. 1. Blink

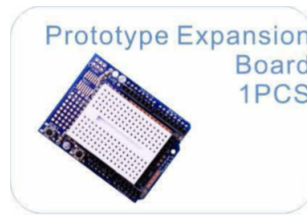
(a) When you first powered up your board, it ran the most-recently uploaded sketch. Replace that unknown sketch with the Blink sketch. The Blink sketch does not require any I/O connections on the board. Choose **File > Examples > 01.Basics > Blink**. To upload and run the sketch either go to **Sketch > Upload** or click the cyan arrow  pointing to the right at the top of the sketch. You should see an amber LED blinking on your Arduino board.

(b) Disconnect your board from the computer, and power it up with a 9-V battery. Verify that the program runs on battery power, then disconnect the battery and reconnect the board to the computer.



(c)-2p Study the Blink code and change the amount of time that the LED is on/off. The timing should be clearly different from the original 1-s on/1-s off. Upload the revised Blink sketch and verify the new timing. Keep this timing for the next step.

(d)-2p You can use the Blink program to run an external LED as well. The LED on the board is powered by digital pin 13. This pin will output a value of either 0 or about 5 volts. The board also has a ground connection labeled **GND**. Use **Digital(I/O) pin 13** and **GND** to power an external LED. Be sure to place a resistor, $R \gtrsim 220 \Omega$, in series with the external LED to limit the current. To make these connections, use either an external breadboard (protoboard) or the prototype expansion board with mini breadboard. The prototype expansion board connects directly on top of the Uno R3 and gives access to all of the I/O.



Now change the output to digital pin 9. Above the line containing `void setup()` type `const int outPin=9;` This declares the variable `outPin` as a constant integer equal to 9. Within the program `LED_BUILTIN` appears three times, referring to the LED built in to the board. Replace `LED_BUILTIN` with `outPin`. Connect your circuit to pin 9 and verify its functionality. Note that the board LED no longer blinks. Have your instructor verify operation and indicate that in your lab notebook.

Expt. 2. Fade

(a) Use pulse-width-modulation (PWM) to control the brightness of your LED. Pins marked with a \sim are capable of PWM. Choose `File > Examples > 01.Basics > Fade` and upload the program. Increase the `delay` on the last line of the program to 300 ms. Pin 9 is operating in “analog” write mode.

(b)-1p Monitor the DC voltage at pin 9 with a multimeter. What range of voltages do you observe?

(c)-1p Now monitor the voltage at pin 9 on the oscilloscope. What voltages do you observe? The `analogwrite()` function varies the duty cycle. When brightness is 0, the LED is always off, and when it is 255, the LED is always on. Watch the duty cycle change from 0% to 100%. What is the period of the PWM? Try adjusting `fadeAmount` to adjust how smoothly the duty cycle varies.

(d)-2p Use your computer to take a snapshot of the oscilloscope screen that clearly shows the high and low voltages at some arbitrary point in time.

Expt. 3. Digital Output

(a) Use the following sketch to send a digital signal to pin D3. You may type it or copy sketch 6-01 from arduinobook.com > [download code]. Note that the `setup()` declares pin 3 to be an output with the `pinMode` command.

```
const int outPin = 3;


void setup()
{
  pinMode(outPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Enter 1 or 0");
}
```

```

}

void loop()
{
  if (Serial.available() > 0)
  {
    char ch = Serial.read();
    if (ch == '1')
    {
      digitalWrite(outPin, HIGH);
    }
    else if (ch == '0')
    {
      digitalWrite(outPin, LOW);
    }
  }
}

```

(b)-2p Upload the sketch. Open the Serial Monitor under **Tools > Serial Monitor** or by clicking the  in the upper right of the window. Enter '1' to set the output high to ~5 V, and enter '0' to set it low to 0 V. As you enter '1's and '0's, monitor the output of digital pin 3 with a multimeter or LED through a 220-ohm resistor.

Expt. 4 Digital Input

(a) Use the following code to read if a digital signal is high or low at pin D5. You may type it or copy sketch 6-02 from arduinobook.com > [\[download code\]](#).


```

const int inputPin=5;

void setup()
{
  pinMode(inputPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  int reading = digitalRead(inputPin);
  Serial.println(reading);
  delay(1000);
}

```

(b)-2p Open the Serial Monitor, , to view if the input is low (0) or high (1). When disconnected from a voltage source the input is said to be floating and you may easily encounter voltages above or below the threshold value of 2.5 V. Use a jumper wire to make connections to pin D5. You can find 5 V (high) and Ground (low) pins on the Arduino board. Confirm that the serial monitor correctly reports the high and low values.


Expt. 5. Analog Input

(a) Use the following code to read an analog signal. You may type it or copy sketch 6-09 from arduinobook.com > [download code].

```
const int analogPin = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int reading = analogRead(analogPin);
  float voltage = reading / 204.6;
  Serial.print('Reading=');
  Serial.print(reading);
  Serial.print('\t\tVolts=');
  Serial.println(voltage);
  delay(500);
}
```

(b)-3p Use the serial monitor, , to monitor the voltage at analog pin 0. To create an analog voltage, use the 5 V and ground pins on the Arduino board, along with a trim potentiometer as a variable voltage divider. Demonstrate this operation to your instructor.



Expt. 6. Passive Buzzer

For the rest of the lab, use the documentation and sketches found at <https://www.elegoo.com/download/>. Download the materials for the Elegoo UNO R3 Super Starter Kit. In the **English** folder, you will find code, libraries, and PDF documentation.

(a)-3p Build the circuit and run the sketch for Lesson 7, The Passive Buzzer,



You will need to include the “pitches” library so that the software knows how to translate a note name to a frequency. Go to **Sketch > Include Library > Add .ZIP Library** and select the pitches library from the **Libraries** folder in your download.

(b)-3p Play a different melody by changing the sketch code. Available notes range from `NOTE_B0 = 31 Hz` to `NOTE_DS8 = 4978 Hz`. All accidentals are denoted with an **S** for sharp.

Expt. 7. Basic Project

(a)-3p Choose one lesson from the following list:

Lesson 9 Servo

Lesson 10 Ultrasonic sensor module

Lesson 11 Temperature and humidity sensor

Lesson 12 Analog joystick module

Lesson 13 IR receiver module.

Demonstrate your project to your instructor.

Expt. 8. Eight LEDs with 74HC595 Shift Register

(a)-4p Build the circuit and run the sketch for lesson 16.

(b)-2p Upload the sketch from lesson 17 and control your LEDs with the serial monitor.

(c)-2p Add the photocell from lesson 18 to your circuit and run the associated sketch. Expose the photocell to varying amounts of light.

Demonstrate your work from lesson 18 to your instructor.

Expt. 9. Advanced Project

(a)-3p Time permitting, choose either lesson 19: the 7 segment display, or lesson 21: the DC motor.

Demonstrate your project to your instructor.