# Lab 1: Arduino Mega Familiarization Report

Mason Wheeler 2032634 3/29/2023
Assignment: 1

**Abstract:**

This lab report outlines the steps taken to familiarize oneself with the Arduino Mega microcontroller board. The primary objectives were to learn how to install and set up the Arduino IDE, build and run basic sketches using Arduino libraries, modify and demonstrate blinking light code, and produce a speaker output tone. The lab demonstrated the ease of use of the Arduino platform, the flexibility of the hardware, and the versatility of the software.

## 1. Introduction:

The Arduino Mega is a microcontroller board based on the ATmega2560 microcontroller. It offers 54 digital input/output pins, 16 analog inputs, and multiple communication interfaces such as UART, SPI, and I2C. The board can be easily programmed using the Arduino IDE, a user-friendly software environment that allows for code development, compilation, and uploading to the microcontroller. This lab report details the process of setting up and configuring the Arduino IDE, building and modifying sketches, connecting hardware, and producing speaker output.

## 2. Methods and Techniques:

The lab involved setting up the Arduino IDE on a computer and working with example sketches to modify and develop new functionality. Specifically, the Blink sketch was adapted to control both the on-board and off-board LEDs, while a tone was used to generate a 250Hz speaker output. The hardware setup included connecting an external LED through a 250 Ohm resistor and connecting an 8 Ohm speaker to the microcontroller. The lab required skills in writing and modifying Arduino code, as well as basic electronics and troubleshooting.

## 3. Procedures and Results

### 3.1 Arduino IDE Setup:

The first step was to install the Arduino IDE on a computer. After downloading the appropriate installer from the Arduino website, the IDE was installed, and the Blink example sketch was opened. The IDE was then configured to work with the Arduino Mega by selecting the appropriate board and serial port.

### 3.2 Blinking On-board and Off-board LEDs:

The Blink example sketch was modified to blink the on-board LED at a 200 ms interval. After uploading the modified sketch, the on-board LED blinked at the desired rate. Following this, an

external LED was connected to the Arduino Mega through a 250 Ohm resistor and the sketch was further modified to blink the off-board LED at the same 200 ms interval. Upon uploading the new sketch, both LEDs blinked simultaneously as expected.

```
// Define pins
int onboard_led = 13;
int external_led = 10;

// Setup
void setup() {
  pinMode(onboard_led, OUTPUT);
  pinMode(external_led, OUTPUT);
}

// Loop
void loop() {
  // Turn both LEDs on
  digitalWrite(onboard_led, HIGH);
  digitalWrite(external_led, HIGH);
  delay(200);

  // Turn both LEDs off
  digitalWrite(onboard_led, LOW);
  digitalWrite(external_led, LOW);
  delay(200);
}
```

Code for part 3.2

**3.3 Generating Speaker Output:**

An 8-Ohm speaker was connected to pin 2 and the +3.3V pin on the Arduino Mega. The previously modified Blink sketch was further altered to generate a 250 Hz tone on the speaker while maintaining the 200 ms blinking interval for both LEDs. The code was uploaded, and the speaker emitted the desired tone without any audible glitches, while the LEDs continued to blink as expected.

```
/*
  Blink LEDs and Make Tone
  Turns on an on-board LED for 200ms, then off for 200ms, repeatedly.
  Turns on an off-board LED connected to pin 10 for 200ms, then off for 200ms,
repeatedly.
  Makes a 250Hz tone on pin 2.
 */

// Define pins
int onboard_led = 13;
int external_led = 10;
int speaker_pin = 2;

// Define speaker settings
const int freq = 250; // frequency of speaker tone in Hz

// Setup
void setup() {
  pinMode(onboard_led, OUTPUT);
  pinMode(external_led, OUTPUT);
  pinMode(speaker_pin, OUTPUT);
}

// Loop
void loop() {
  // Turn both LEDs on
  digitalWrite(onboard_led, HIGH);
  digitalWrite(external_led, HIGH);
  delay(200);

  // Turn both LEDs off
  digitalWrite(onboard_led, LOW);
  digitalWrite(external_led, LOW);
  delay(200);

  // Emit a 250Hz tone continuously without interrupting LED blinking
  tone(speaker_pin, freq);
}
```

Code for part 3.3

4. **Code Documentation:**

The code was organized into separate tasks, including LED control and speaker tone generation. Initialization processes were added in the setup() function, while the main code logic resided in the loop() function. Each task and initialization process was documented with comments and line number references, making it easy to understand the purpose and function of each code segment.

5. **Overall Performance Summary:**

The demo session was successful in meeting the learning objectives and demonstrating the capabilities of the Arduino Mega platform. The on-board and off-board LEDs blinked at the specified intervals, and the speaker produced a 250 Hz tone without any audible glitches. The lab showcased the ease of use and versatility of the Arduino platform for hardware-software integration.

6. **Teamwork Breakdown:**

Since this lab report was done by a single individual working on the lab, all tasks were completed by Mason Wheeler. Mason was responsible for installing and setting up the Arduino IDE, modifying the Blink sketch, connecting the hardware components, generating the speaker output, and documenting the code.

7. **Discussion:**

Throughout the lab, the Arduino Mega proved to be an easy-to-use and versatile platform for hardware-software integration. The Arduino IDE provided an intuitive environment for writing, compiling, and uploading sketches to the microcontroller. The hardware setup and configuration were straightforward, enabling quick and efficient prototyping of various tasks.

The lab demonstrated the ease with which one can build and modify sketches to control various hardware components, such as LEDs and speakers. Furthermore, the Arduino Mega's extensive range of digital and analog pins allows for the incorporation of additional components and functionality as needed.

## 8. Conclusion:

This lab served as an introduction to the Arduino Mega microcontroller platform and provided a foundation for future projects involving hardware-software integration. The Arduino IDE, together with the Arduino Mega, proved to be a powerful and flexible platform for rapid prototyping and learning. With a solid understanding of the Arduino environment and its capabilities, more complex and sophisticated projects can now be undertaken, further exploring the potential of the Arduino platform.