

# Mechatronics

Badger Invitational  
Hamilton Middle School  
2019-02-23

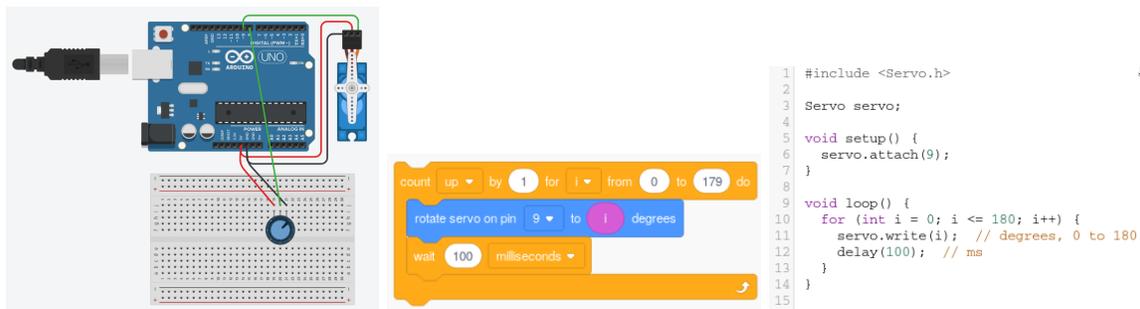
Using your browser, visit the website <http://mechatronics.blaise.zone>. Here you will find links to all of the Tinkercad circuits referred to in this exam. After the competition, the correct answers will be released on this same website.

## 1 Written Test

You will have 25 minutes to complete this portion of the exam. There are four questions, each worth 25 points for a total possible score of 100 points.

### 1.1 servo

Using Tinkercad, open the “servo” circuit. You may choose to use text or blocks to program the Arduino.



#### 1.1.1 10 points

Your Arduino has been pre-programmed to remind you how to use the Servo library. As written, the code will simply move the motor from 0 to 180 degrees and then start over. Roughly how long will the motor take to accomplish one rotation? Modify the code so that the pattern repeats at 0.5 Hz.

#### 1.1.2 5 points

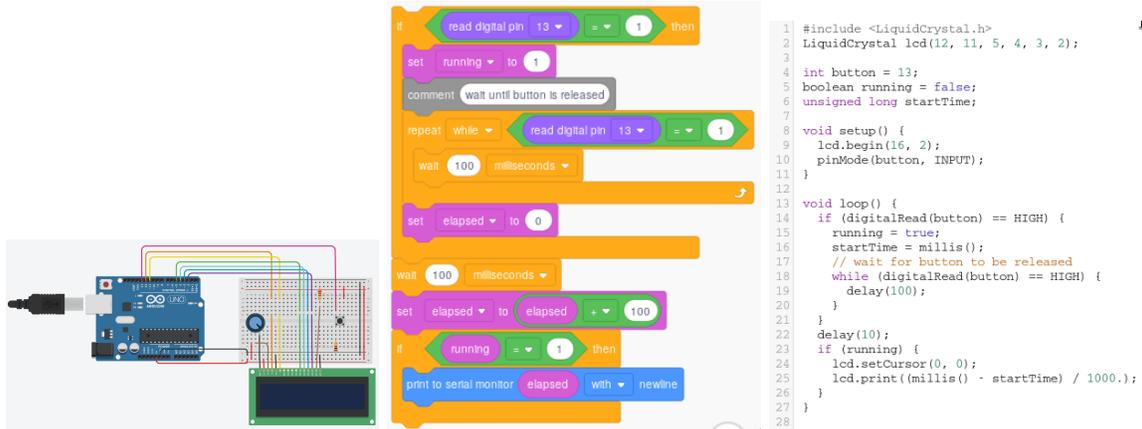
The middle pin of the potentiometer is wired to pin 8 of the Arduino. Is this appropriate? If not, choose an appropriate pin and explain your reasoning.

#### 1.1.3 10 points

Modify your code such that the potentiometer can be used to control the position of the servo motor. Ensure that the dynamic range of the potentiometer (0 to 5 V) agrees with the dynamic range of the servo motor (0 to 180 degrees). Recall that the Arduino has a 10 bit analog-to-digital converter.

## 1.2 stopwatch

Your friend has designed a simple stopwatch using an Arduino, button, and LCD display. How clever of her! The circuit works perfectly, but the code is behaving strangely. She needs your help to debug the project. Open the stopwatch circuit in Tinkercad. You may choose to use text or blocks to program the Arduino.



```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3
4 int button = 13;
5 boolean running = false;
6 unsigned long startTime;
7
8 void setup() {
9   lcd.begin(16, 2);
10  pinMode(button, INPUT);
11 }
12
13 void loop() {
14   if (digitalRead(button) == HIGH) {
15     running = true;
16     startTime = millis();
17     // wait for button to be released
18     while (digitalRead(button) == HIGH) {
19       delay(100);
20     }
21   }
22   delay(10);
23   if (running) {
24     lcd.setCursor(0, 0);
25     lcd.print((millis() - startTime) / 1000.);
26   }
27 }
28
```

### 1.2.1 10 points

The stopwatch starts correctly when the button is pressed the first time. Subsequent button presses reset the stopwatch. This is not intended behavior! Your friend wants her stopwatch to *pause* when the button is pressed a second time.

Briefly, explain to her *why* your friends' code doesn't work as intended. Be sure to describe the relationship between button presses and the "running" variable.

### 1.2.2 10 points

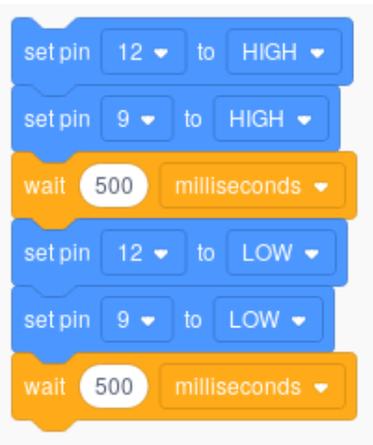
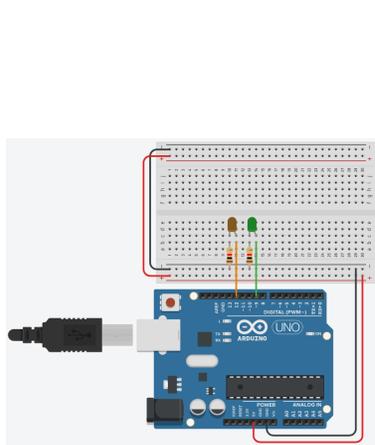
Fix your friends' code. Describe what you changed, and why that change fixes the problem.

### 1.2.3 5 points

Why did your friend add a while loop?

## 1.3 blink 2

Using Tinkercad, open the “blink2” circuit. You may choose to use text or blocks to program the Arduino.



```
1 int led1 = 12;
2 int led2 = 9;
3
4 void setup()
5 {
6   pinMode(led1, OUTPUT);
7   pinMode(led2, OUTPUT);
8 }
9
10 void loop()
11 {
12   digitalWrite(led1, HIGH);
13   digitalWrite(led2, HIGH);
14   delay(500);
15   digitalWrite(led1, LOW);
16   digitalWrite(led2, LOW);
17   delay(500);
18 }
```

### 1.3.1 5 points

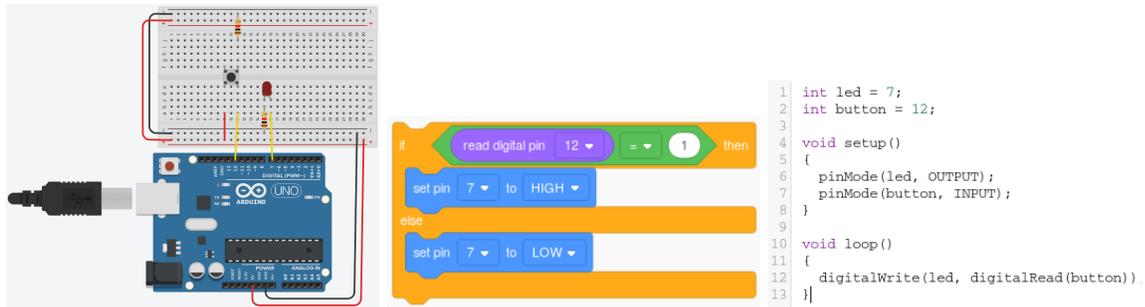
Blink both LEDs, such that the green LED is on when the orange LED is off and vice versa.

### 1.3.2 20 points

Change the frequency of the green LED such that it is blinking three times slower than the orange LED. Keep the duty cycle of both LEDs at 50 %.

## 1.4 switch

Using Tinkercad, open the “switch” circuit. You may choose to use text or blocks to program the Arduino.



### 1.4.1 10 points

Consider the following definitions:

A *toggle switch* changes the state every time it is depressed. Once released, the new state (on or off) is maintained until the switch is pressed again. Example: common light-switch.

A *momentary switch* is on only while being pressed. Once released, the switch automatically turns off again. Example: computer keyboard key.

Is the button acting as a momentary switch or a toggle switch? Is this relationship defined by the circuit, or by the Arduino code? Explain how you know.

### 1.4.2 10 points

Without changing the circuit, modify your Arduino code so that the button acts like a toggle switch. Describe your approach.

### 1.4.3 5 points

How could you change the brightness of the LED?

## 2 Whatchamacallits

You will have 25 minutes to complete this portion of the exam. Complete as many of the following exercises as you can. There are 100 points possible.

Some of these questions use Tinkercad, and others require the usage of real hardware. If you prefer to program using blocks, you must export your Tinkercad code and upload it using the Arduino IDE. If you have never worked with real Arduinos, we will help you get started.

### 2.1 traffic light - 10 points

Using the provided components, made an LED traffic light. Place three LEDs: green, yellow and red. Make them blink in a specific pattern: 1) long green, 2) short yellow, 3) long red, then repeat.

### 2.2 rainbow - 10 points

Using Tinkercad, make an RGB LED cycle through the rainbow (Red, Orange, Yellow, Green, Blue, Indigo, Violet) and back again.

Here is a table of values for each color:

	red	green	blue
Red	255	0	0
Orange	255	127	0
Yellow	255	255	0
Green	0	255	0
Blue	0	0	255
Indigo	75	0	130
Violet	148	0	211

### 2.3 dancing motor - 10 points

Turn on a motor for one second, then off for one second, then on for two seconds, then off for two seconds, then on for three seconds, then off for three seconds, and so on repeating the process to 100 seconds.

## **2.4 click counter - 20 points**

Using Tinkercad, make a click counter. Using an LCD screen as a display, create a circuit that shows how many times a button is depressed. The circuit should count each click separately, even when the button is held down for a prolonged period.

## **2.5 binary counter - 20 points**

Using the provided components, make a three-bit binary counter. Use three LEDs. Count from zero to seven, then repeat. Spend roughly one second on each number.

## **2.6 7-segment - 30 points**

You will be given an Arduino with a pre-wired seven segment display. Each segment is wired to a separate pin (see documentation). Program the Arduino to count from zero to nine, spending around one second on each number.