Name: Date: Class:

# Activity Worksheet Answer Key

## Part 2: Understanding the Arduino Microcontroller

**Instructions:** Implement a code to turn the LED on for 300 ms, off for 100 ms, on for 500 ms, and then off for 500 ms.

```
// Blink program
// the setup function runs once when you press reset or power the board
void setup() {
 // initialize digital pin LED_BUILTIN 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(300);                        // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(100);                       // wait for a second
digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(500);                        // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(500);                       // wait for a second
}
```

## Part 3: Understanding the MAX30102 Spectrometer

**Instructions:** After watching this video, https://www.youtube.com/watch?v=MHPgamGQmDY (7:36 minutes), explain the importance of red light, infrared light, and the optical sensors to measure blood oxygen rate in MAX30102.

MAX30102 is an optical sensor that measures the absorbance of pulsating blood through a photodetector after emitting two wavelengths of light from two LEDs - a red and an infrared one. The MAX30102 works by shining visible red light and infrared light and both lights onto the finger and measuring the amount of reflected light using a photodetector.

## Part 4: Understanding the AD8232 Sensor

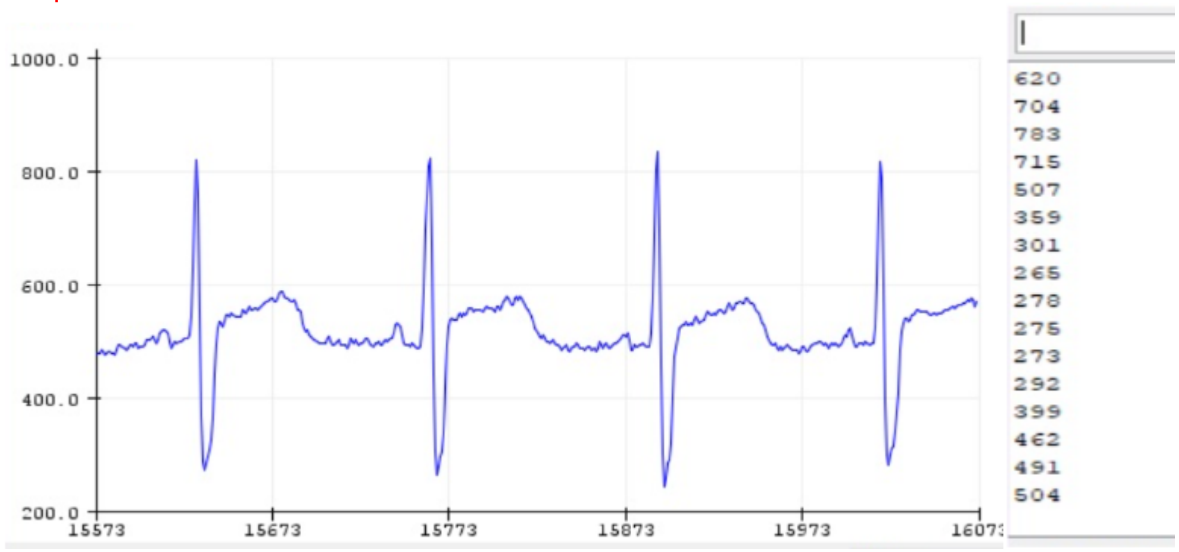
**Instructions:** First, view the video on Slide 15, https://www.youtube.com/watch?v=01y_Vu_sAQU&t=463s (8:59 minutes), which shows how to connect the Arduino to the AD8232 sensor. Then:

1. Connect the AD8323 sensor to the Arduino as in the video.
2. Copy and paste the program in Slide 18.
3. Compile and run.
4. Choose serial plotter to capture ECG.
5. Place the leads on the right position as in the video.

**Name:** **Date:** **Class:**

6. Compute the peaks for 15 seconds.
7. Save or take screenshots of heart rate and oxygen level readings.

Sample screenshot



```
620
704
783
715
507
359
301
265
278
275
273
292
399
462
491
504
```

## Data Collection

**Instructions:** Collect data from the sensors.

1. Test the MAX30102 and AD8323 on three students.
2. For each student tested, write down the oxygen level and heart rate (HR) using MAX30102.
3. Use serial plotter for AD8323 to calculate the (HR) = 4 * number of the peak per 15 seconds.
4. Complete the data table below.

| Student Name | Oxygen Level Using MAX30102 | HR Using MAX30102 | Calculated HR Using AD8323 | Arrhythmia Tachy or Brady | Reason for Arrhythmia Case |
|---|---|---|---|---|---|
| Student 1 | 98 | 89 | 95 | Normal | |
| Student 2 | 99 | 130 | 140 | Tachycardia | HR> 100 |
| Student 3 | 97 | 40 | 50 | Bradycardia | HR<60 |

## Reflection Questions

1. Think about and write down the reasons for getting different HR measurements for the same student. Which one is more accurate?
   Answers will vary.

2. How could you improve the design to get more accurate results?
   Answers will vary.

**Optional: Application of Computational Thinking:**

3. How did you apply decomposition, pattern recognition, abstraction, and algorithm design in this activity?
   Answers will vary. Potential answers:
   **Decomposition:** I broke down a complex problem—measuring heart activity—into smaller steps. First, I learned how the heart works. Then I connected sensors, wrote code, and analyzed data. Each part of the activity had a clear task, like installing the Arduino IDE, wiring the MAX30102 sensor, or interpreting ECG graphs.

   **Pattern Recognition:** I identified patterns in the data from both sensors. For example, I looked at repeating peaks in the ECG graph to find my heartbeats, and I watched how my oxygen level changed over time.

   **Abstraction:** I used graphs and numerical readouts to represent what's happening in the body. I didn't need to see the actual heart to understand its electrical activity—I used simplified models (graphs and numbers) to make sense of the system.

   **Algorithm Design:** I followed specific steps to make the Arduino collect and display data. I also uploaded code that tells the sensor how to behave, such as how long to keep an LED on or how to calculate heart rate from sensor input.

4. Can you think of other real-world applications where similar data analysis techniques might be useful?
   Answers will vary. Potential answers:
   These techniques are used in many real-world areas:
   - Healthcare: Doctors use ECGs, pulse oximeters, and other sensors to diagnose heart and lung problems.
   - Sports and fitness: Athletes use wearables to track heart rate, oxygen levels, and recovery.
   - Emergency response: Paramedics monitor vitals to make quick decisions in emergencies.
   - Sleep studies: Researchers analyze heart rate and oxygen patterns to detect sleep apnea.
   - Space medicine: NASA monitors astronauts' vitals in real-time during space missions.