Wenqing Pan
Professor Vivian Xu
INFOSCI 201:Interaction and Physical Computation
19 May 2023

# Life Clock

## 1. Problem Statement

The topic of time and life has been a constant throughout the ages, and it has been mentioned over and over again. We all know that we have to cherish time, and time is life. However, in our lives, we are wasting time constantly without recognizing it. There are so many elements in our life that can drive us to waste time, including losing passion, feeling mentally exhausted, lack of organization, being surrounded by distractions, etc(Marketing 2020). All these elements can cause people to keep wasting small periods of time without realizing it. According to recent research, people spend 1 to 2.34 hours every day checking emails while 30% of them are not important("13 Shocking Facts about How We Waste Time" 2018). People also spend about 2 to 35 minutes per day deciding what to eat and 3 to 16 minutes deciding what to wear("13 Shocking Facts about How We Waste Time" 2018). Although the time spent on each of the unimportant things is not much, the total amount of time will be huge when adding them together.

To address this issue, it is crucial to visualize the amount of time people waste. Compared to other forms of communication, visual communication offers numerous advantages. Firstly, visual communication can deliver information more directly (Nishadha 2018). Unlike verbal communication, visual communication is easily comprehensible to individuals of any language and culture. Moreover, visual communication requires less time to convey information, as it does not require extensive verbal explanations. Secondly, visual communication can capture people's attention and encourage greater engagement (Nishadha 2018). For instance, people remember only 10% of what they read and 20% of what they hear, but they recall 80% of what they see (Nishadha 2018). Consequently, visual communication can be a more effective way to convey information. Lastly, visual communication can elicit an emotional response in the audience, making it more impactful than other forms of communication (Nishadha 2018).

Therefore, there is an urgent need to visualize the passage of time and life. With the device, people can better understand the preciousness of time and inspire the audience to think more about their life.

## 2. Design Concept & Value

In order to address the issue of people wasting time unconsciously, we proposed the idea of making a life clock, using an Arduino hardware kit and Processing. This project is aimed at making the passage of time visualized, and conveying the message that time flows incessantly without stopping and life is cyclical. To illustrate this concept, the hardware system will consist of several elements, including a clock, mechanical flowers, and LED lights, and the software will be based on heart rate to create two scenes to represent the passage of time.

Firstly, with the combination of clock and heart rate, I want to express the message that time cannot be stopped. Time is passing fast with every heartbeat. Just like we can't stop the heartbeat, we can't stop the passage of time. When the audience puts on the heart rate sensor, the software system starts to read the data, and the clock hand in the hardware system starts to run. No matter how fast the heart beats, the hand is designed to run at a constant speed.

Secondly, the blossoming and falling mechanical flowers represent the beginning and end of life. The fully blossomed flowers represent birth, while the falling ones symbolize the end of life. As the heart rate reaches a certain range, the corresponding flower will fully blossom, and other flowers will bloom in clockwise order. The constant opening and closing of the flowers indicate the fleeting nature of life, while the succession of the flowers represents the idea of inheritance.

Thirdly, the usage of LED light also represents life. However, it showcases the hope that life can continue. In many cultures, light represents goodness in the world and is often associated with the source and provider of life(Christopher 2020). No matter what an individual ’ s spiritual beliefs are, there tends to be a unified understanding that light is the source and provider of life(Christopher 2020). In Christianity, God creates light before creating life. Similarly, scientific theories argue that an expanding source of light is responsible for the big bang which creates our universe. By incorporating the element of light into the system, we aim to add another layer of meaning to the project.

Last but not least, a Processing scene is also designed: randomly flying butterflies. The colors of butterflies will be decided by heart rate data. According to several pieces of research, color can be used to represent people’s mode(London Image Institute 2020). I also imply this theory and combine with to my project.

Different categories of heart rate data will lead to different colors of the elements in the scenes.

The system is designed to visualize the passage of time and life without inducing anxiety in the audience. It aims to inspire viewers to reflect on their lives and encourage them to see the system from different perspectives.

## 3. System Design

### 3.1 Task Flow

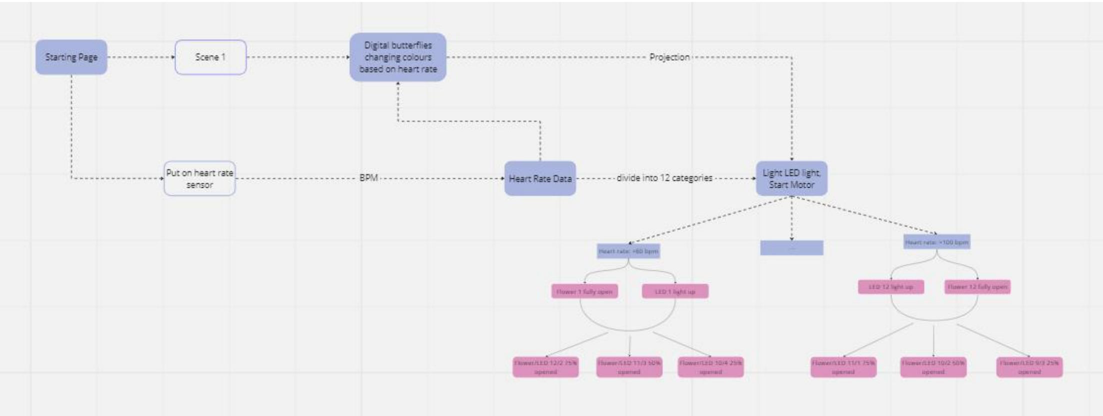
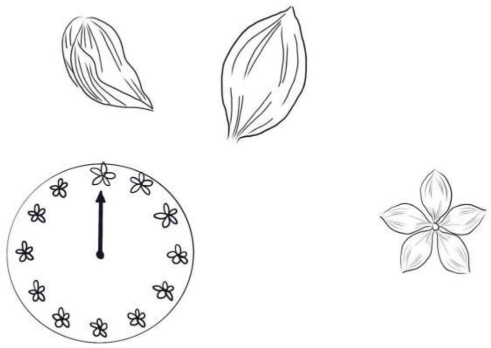



*Figure1*. Task flow of the sensor and hardware

The task flow demonstrates the process of interacting with the system. Ranging from putting on the heart rate sensor to operating some physical reactions on the hardware(LED lights and flowers), together with using the interaction pages made by Processing.

### 3.2 Wireframe



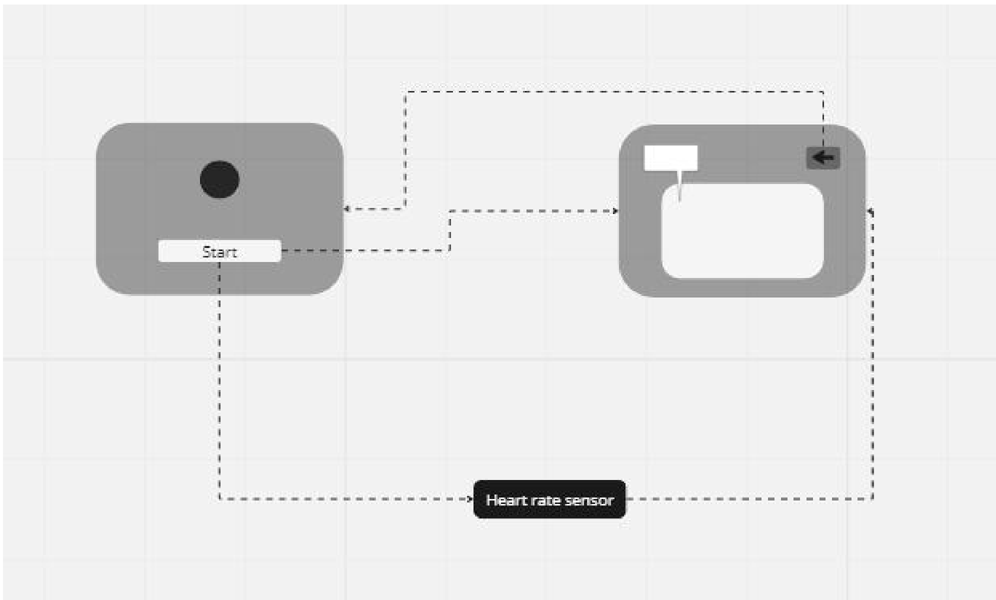



*Figure2*. Wireframe of the hardware

The wireframe showcases the general layout of each UI and how software and hardware connect with each other.

### 3.3 User Flow

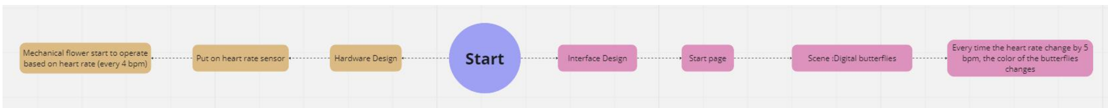



*Figure3*. User flow of the completed system

The user flow includes both the software system and the hardware system.

## 4. Proposed Outcome

The final outcomes of the interactive clock project pertain to an interactive kinetic sculpture constructed using an Arduino Kit. The project incorporates a software system that acquires a user's heart rate utilizing a heart rate sensor. Upon detecting a user's heart rate, the clock hand initiates motion. Arduino technology is utilized to allocate the data to different motors and LED lights, regulating the activation of target lights and mechanical flowers at specific intervals. I will also use Processing to make digital butterflies and change the colors of them in my scenes based on different heart rate data

The project primarily aims to communicate three fundamental concepts. Firstly, it aims to visualize the progression of time. Secondly, it intends to symbolize that life commences and concludes swiftly and perpetually. Lastly, it aims to express that the inheritance of culture will persist beyond time. The project integrates various enlightening components that inspire the audience to contemplate from diverse perspectives.

## 5. Prototype1

### 5.1 Model Design

I planned to use a 3D printer to print the whole flower for my final project, including the floral axis and the petal. The 3D model for the flower will be divided into three parts: 1) Flower petal, 2) Flower axis, and 3) Connecting components. I have two types of models for the flower axis. One model uses 3D-printed connecting components as the parts which are indirectly connected to servos to move the flower petals. Depending on the gravity and the movement of servos, the mechanical flower will bloom and fall. For the second model, I plan to use strings to control the movement of petals. One end of the string is connected to the servo and the other is connected to a piece of petal. When the servo moves down, it drags the string and the flower will fall. When the servo moves upwards, the flower will bloom.

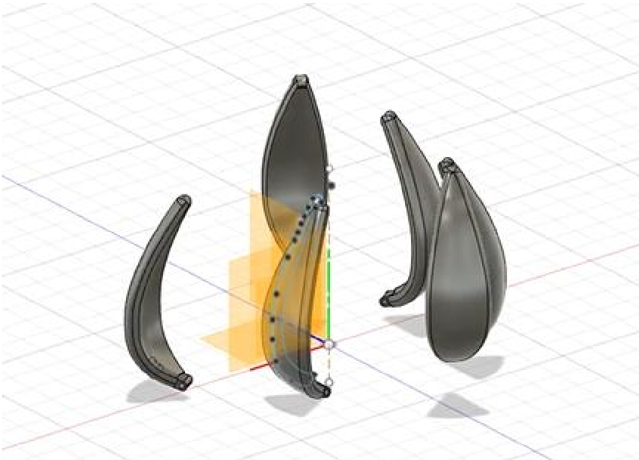


*Figure4. Flower petal model*

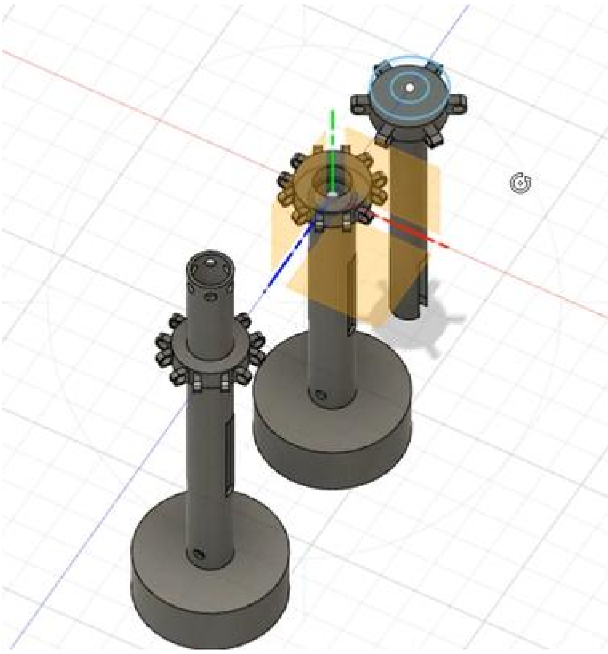


*Figure5. Flower axis model*

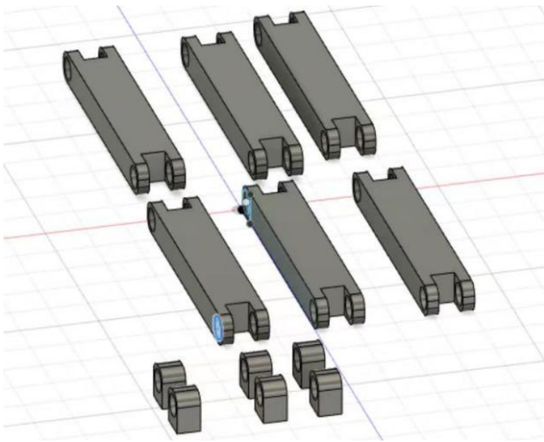
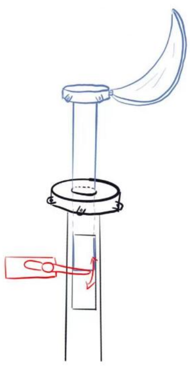


*Figure6. Connecting components model*



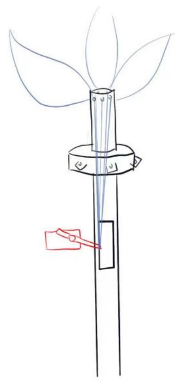


*Figure7.The mechanics of model1* *Figure8. The mechanics of model2*

### 5.2 Hardware Design

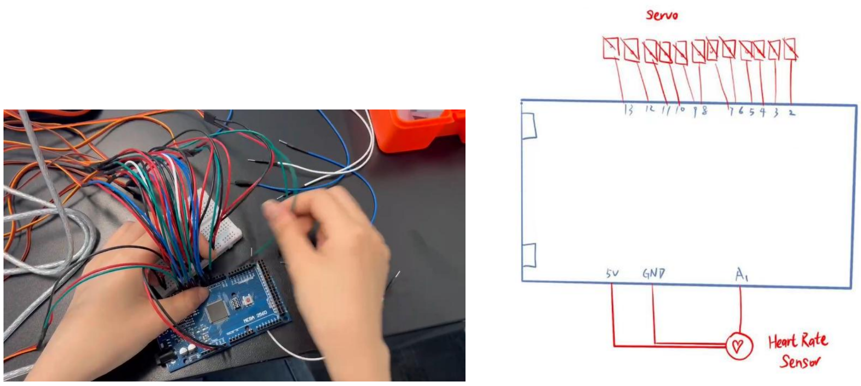



*Figure9. Circuit1*

*Figure10. Sensor*

For the hardware section in prototyp1. I tried two ways to control the servos, using a heart rate sensor. One is dividing the heart rate values in the interval 60 to 100 into 12 parts. Whenever the heart rate value is in a certain interval, the corresponding servos turn.

**5.3 Software Design**

To make the servos run with different heart rate values, I used Arduino to read the data from the heart rate sensor, recalculated the data, and sent the bpm value to different servos. Since the original data we get from the heart rate sensor is ranging in a large interval, I map the data into a usual interval with the unit of beat per minute. I designed the system to have 12 servos. Every time a servo is assigned to move, it will also make the two servos beside it move too. For example, the servo in the middle will run 90 degrees together with its nearby servos. After that, it will run from 90 degrees to 180 degrees and returns from 180 to 90. Then the three servos will return from 90 degrees to 0 degrees together.

```
void loop() {
  uint8_t rateValue;
  heartrate.getValue(heartratePin);  // A1 foot sampled values
  rateValue = heartrate.getRate();
  if (rateValue) {
```

*Figure11. Map the original data to a smaller interval*

```
if (rateValue) {

  if (rateValue <= 60) {
    for (pos = 0; pos <= 90; pos += 1) {
      servo1.write(pos);
      servo2.write(pos);
      servo12.write(pos);
       Serial.println(rateValue);
      delay(15);
    }
    for (pos = 90; pos <= 180; pos += 1) {
      servo1.write(pos);

      delay(15);
    }

    for (pos = 180; pos >= 90; pos -= 1) {
      servo1.write(pos);
      delay(15);
    }
    for (pos = 90; pos >= 0; pos -= 1) {
      servo1.write(pos);
      servo2.write(pos);
      servo12.write(pos);
      delay(15);
    }
  }
```

*Figure12.Simultaneously control three servos and rotate them at different angles*

**5.3 Mechanical Flower Design**

I used iron strings to make the flower petals. After the petals are made, I used flower-making liquid, a polish similar to nail polish, to fill the gaps of the petals. After the drying is completed, I wrapped the petals onto the wire and wrapped the wire into the 3D printed model. I also used string to connect the flower petals. Every time we dragged the strings, the petals will move up and down.

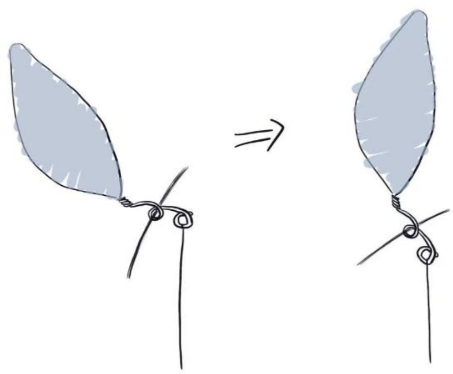


*Figure13.Mechanical flower working principle*

**5.4 User Feedback for Prototyp1**

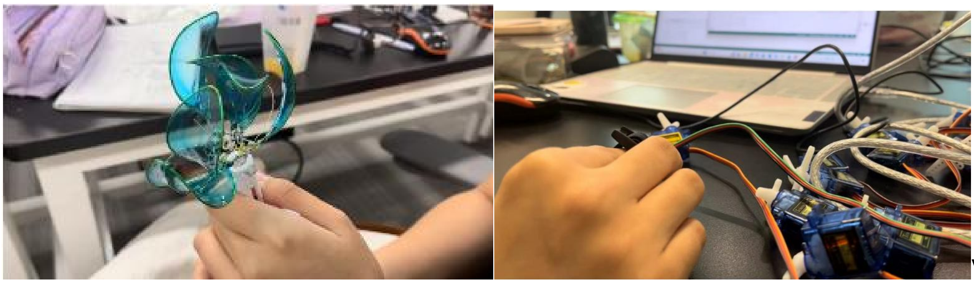


*Figure14. User testing 1*

After asking different people to try and give feedback about the four parts of my prototype one separately, I got multiple suggestions on how to improve my previous version.

For the 3D model section, two of my user testing participants suggested giving up using a 3D printer to make the flower petal, since the extra material which will be used to support the main part will be difficult to clean and will cost too much time to print them. Additionally, one of the participants also mentioned reducing the horizontal part of the model and making the size thicker, otherwise, the model will be unstable.

As for the hardware and software section, most of my testing participants mentioned that the sensor was unstable, it could not read data constantly and sometimes need to wait for a long time before the next data was read. Moreover, sometimes, the computer also crushed when too many servos were receiving data and making reactions.

For the mechanical flower design part, users reflected that the petals could not keep in balance, every time they drag the line, the petals are moving around in the iron ring. Also, since every petal was handmade, its center of gravity is not in the same position, so much so that the petals open and close differently each time the wire is pulled.

In conclusion, there are many parts for me to improve in my prototype 2. Firstly, I will need to change my 3D model and simplify it, including removing the horizontal parts and deleting the 3D petal model. For the sections of hardware and software, I will reduce the number of servos to four and use other plugs to give more power to the system. In this way, the servos can run more smoothly. Last but not least, I will also change the way I connect the iron petals to the iron rings so that they will be more in balance.

## 6. Prototype2

### 6.1 3D Model Design.

In the prototype 2, I simplified the model which only contains twelve vertical flower axis.

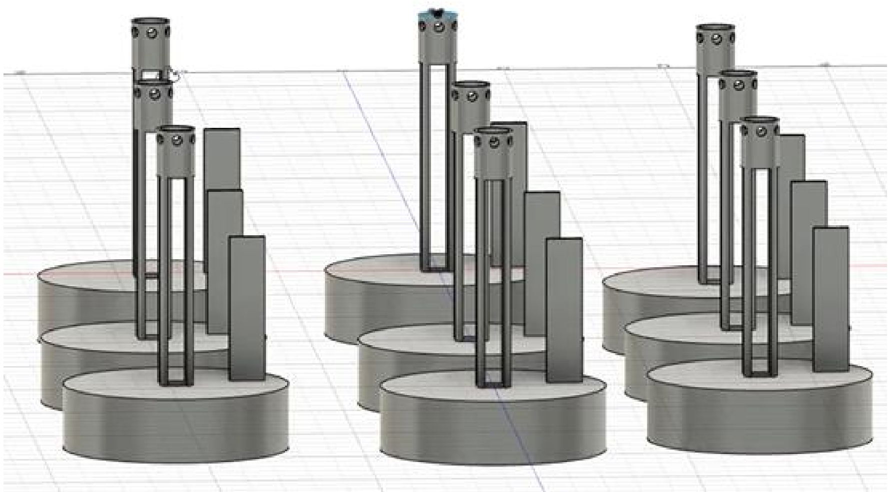


*Figure15. 3D Floral Axis*

### 6.2 Hardware Design

Based on the suggestions I got in prototype 1, I decided to decrease the number of my servos to four, which can be used to represent 12:00, 3:00, 6:00, and 9:00 on the clock face. The intervals for the heart rate changed to “60-70”,” 70-80”,” 80-90” and “90-100”. In addition, I attached four led lights to the Arduino board. Each time the heart rate value reached a specific value, a certain servo will start to run 180 degrees and they will also run 180 degrees after it one by one. The led light which is assigned to that interval, will also be turned on. The led light will only be off when a new run starts. In this way, the audience can visualize which interval their heart rate value is in. Apart from that, I also applied a servo that can run 360 degrees. I made the servo to run clockwise at a fast speed. I also put a 3D printing clock hand on the servo to make it look more like a clock.

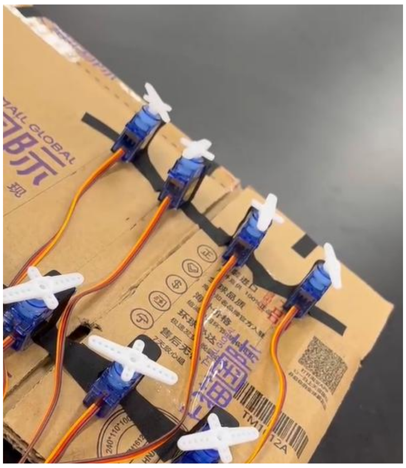


*Figure16. 4 Servo Connection*

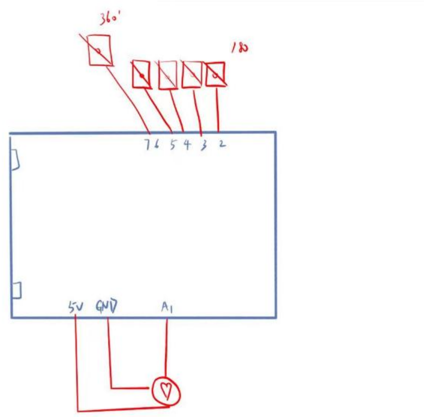



*Figure16. 4 Servo Connection*

### 6.3 Software Design

Since the number of servos decreased in prototype 2, sometimes the computer read data too frequently to send the next order to a certain servo before it finished its previous run. In this way, I decided to change the flow of the system. Instead of making the center servo run 180 degrees and the surround servos run 90 degrees which are designed in prototype 1, I decided to make all servos run when a heart rate value is read. For example, when the system read a certain heart rate value, it will send it to a certain servo. After the servo runs 180 degrees, the servo next to it will run one after the other in clockwise order.

Arduino Mega or Mega 2560

final_180360.ino

```
if(rateValue)  {
  if (!servoBusy) { // 舵机空闲状态
    servoBusy = true;
  if (rateValue > 60 && rateValue <= 70 ) {
    digitalWrite(LED1,HIGH);
    digitalWrite(LED2,LOW);
    digitalWrite(LED3,LOW);
    digitalWrite(LED4,LOW);
    for (pos = 0; pos <= 180; pos += 1) {

      servo1.write(pos);
      //servo2.write(pos);
      //servo4.write(pos);
      Serial.println(rateValue);
      delay(15);
    }
    for (pos = 180; pos >= 0; pos -= 1) {
     servo1.write(pos);

      delay(15);
    }
    for (pos = 0; pos <= 180; pos += 1) {

      servo2.write(pos);
      delay(15);
    }
    for (pos = 180; pos >= 0; pos -= 1) {
     servo2.write(pos);

      delay(15);
    }
```

*Figure17. Control LED lights and servos*

Apart from using a heart rate sensor to control 5 servos, I also used the heart rate values to control the color of digital butterflies. I divided the heart rate values from 60 to 100 into eight categories, each has an interval of 5. When the heart rate is below 60, the butterflies will be gray; When the heart rate value is between 60 and 65, the value is white; “65-70” is brown; “70-75” is pink; “75-80” is purple; “80-85” is blue; “85-90” is green; “90-95” is yellow; “95-100” is orange; above 100 or null is red. The butterflies is designed to run in different directions at random speeds. Every time a butterfly disappears in the scene, a new butterfly will appear. In this way, the digital butterfly will look more realistic.

processing demo1 butterfly | butterfly

```
import processing.serial.*;
Butterfly[] butterfly = new Butterfly[31];


Serial myPort; // 声明一个串口对象
float sensorValue; // 用于存储心率传感器的值
color ballColor = color(255); // 设置小球的初始颜色为白色

void setup() {
  size(800, 800); // 创建一个400x400像素的窗口
  noSmooth();
  myPort = new Serial(this, "COM5", 115200); // 打开串口，注意替换“COM3”为你的Arduino串口号
  myPort.bufferUntil('\n'); // 设置串口接收数据的结束标志为“\n”
  for (int i =0; i<butterfly.length; i++){
  butterfly[i]=new Butterfly();
}
}

void draw() {
  background(0); // 设置背景颜色为白色
 // 在窗口中央画一个小球
     for (int i =0; i<butterfly.length; i++){
  butterfly[i].fly();
  butterfly[i].show();
}
  fill(255);
  textSize(15);
```

*Figure18. Use Arduino to control Processing*

文件 编辑 速写本 调试 工具 帮助 processing_demo1_butterfly | Proc...

processing demo1 butterfly | butterfly

```
void serialEvent(Serial myPort) { // 当串口接收到数据时调用的函数
  String inString = myPort.readStringUntil('\n'); // 读取串口数据
  if (inString != null) { // 如果读到了数据
    inString = trim(inString); // 去除数据中的空格和换行符
    sensorValue = float(inString); // 将读到的数据转换为浮点型
    println(sensorValue);
    if (sensorValue < 60) {
      ballColor = color(128); // 小于60时设置小球为灰色
      println(sensorValue);
    } else if (sensorValue < 65) {
      ballColor = color(255); // 60-65时设置小球为白色
      println(sensorValue);
    } else if (sensorValue < 70) {
      ballColor = color(139, 69, 19); // 65-70时设置小球为棕色
      println(sensorValue);
    } else if (sensorValue < 75) {
      ballColor = color(255, 192, 203); // 70-75时设置小球为粉色
      println(sensorValue);
    } else if (sensorValue < 80) {
      ballColor = color(204, 179, 255); // 75-80时
      println(sensorValue);
     } else if (sensorValue < 85) {
      ballColor = color(153, 255, 255); // 80-85时设置小球为蓝色
      println(sensorValue);
    } else if (sensorValue < 90) {
      ballColor = color(170, 255, 128); // 85-90时设置小球为绿色
      println(sensorValue);
```

*Figure19. Use heart rate value to control the color of digital butterflies*

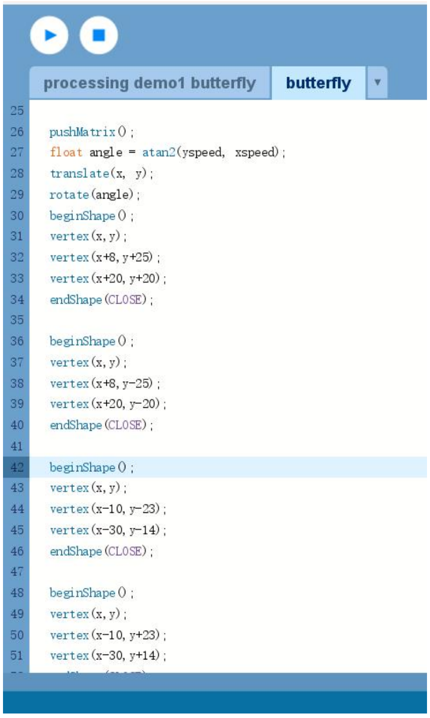



*Figure20. Draw digital butterfly with Processing*

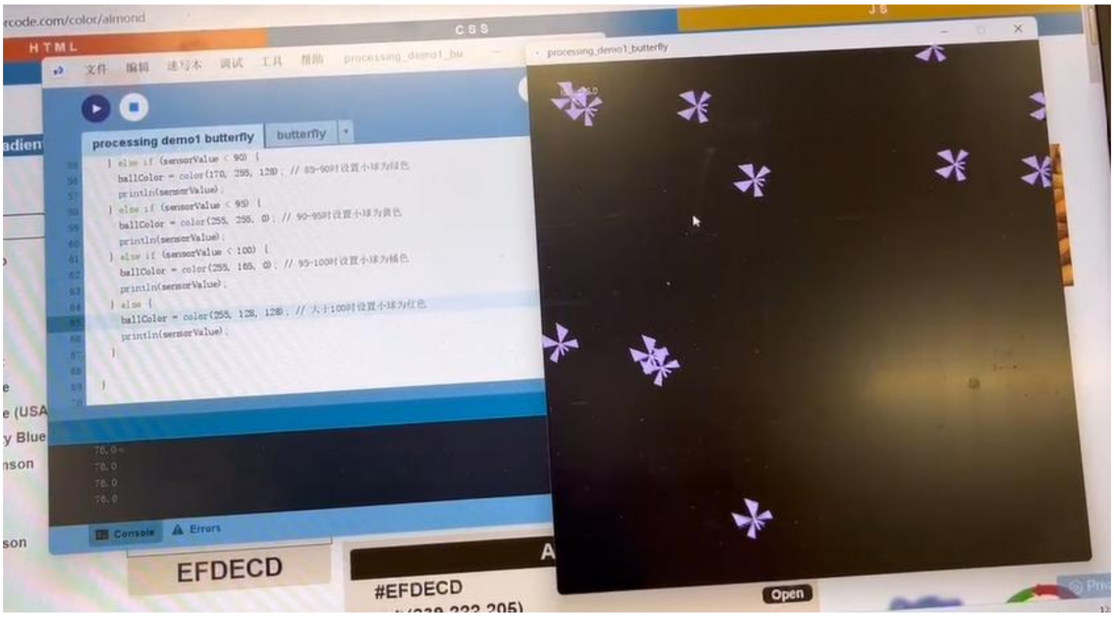



*Figure21. Digital Butterflies in Processing*

### 6.4 Mechanical Flower Design

To improve my mechanical flower in prototype 1, I redesign the way that the flower petals rotate around the iron ring. I made two iron loops for each petal. In this way, each flower petal will be balanced. In addition, I also made six loops in the iron ring. Placing each petal’s loops on the two sides of the loop in the iron ring, the petal will be limited to a certain location. All it can do is rotate.

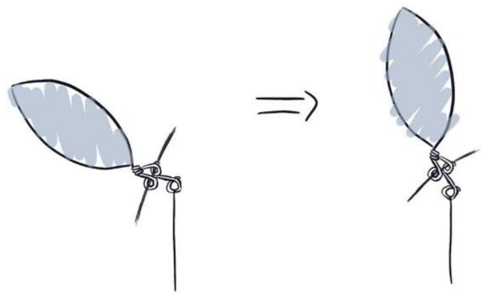


*Figure22.Mechanical flower working principle 2*

*Figure23.Mechanical flower working principle 2.1*

After finished making four mechanical flowers, I also made a clock face to put the flowers on it. In the middle of the clock, there is a hole to put the 360-degree servo into. At the location of 12:00, 3:00, 6:00, and 9:00, I separately put the four mechanical flowers. I also attached a led light to every mechanical flower.

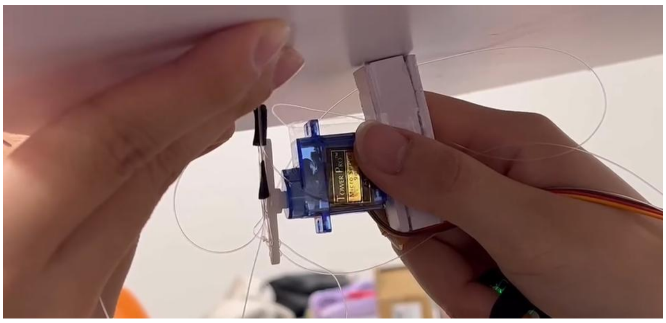


*Figure24.Supporting Material under the clock face*

*Figure24.Final Outcome*

**6.6 User Feed back**

*Figure26. User Testing 2*

After several rounds of user testing, we gain some valuable feedback from the participants. Most of them who participate in the testing of prototype 1 agree the final prototype looks better and runs more smoothly. It read heart rate data frequently and the servos act more sensitively. Moreover, the mechanical flower petals move more smoothly and are balanced. When I ask about the topic, most of the participants said that they can quickly get the main idea of the project, and the combination of mechanical flowers, a clock hand, lights, and butterflies make the project impressive to them. Two of the participant especially show their curiosity and favor toward the flower petal. They think the material I choose to make the flower is very right. The flower-making liquid shows the lightness and translucency of real petals.

In addition, lots of participants give suggestions on how to further improve the project. One mentioned that if the digital butterflies in Processing look more realistic will give people a better visual experience. Right now, the butterflies can only move in a straight line and cannot rotate their direction during the move. Also, many of the participants show their interest in seeing more mechanical flowers in the project. Since a clock has twelve whole points, they are wondering whether a larger clock face that contains twelve flowers will look better. Last, but not least, when running the mechanical flowers, nearly one-third of the users found some flower petals cannot open or close fully. The reason for this situation may be that the wire has been twisted too many times and thus rusted, or the wire connecting the petals is not tied tightly to the servo.

## 7. Conclusion

Nowadays, with more and more access to information and various ways of entertainment, it has become easier for people to waste time without knowing it. My project is aimed at showing the loss of time differently and impressively--displayed as a flower blooming and falling and a running clock hand.

Through completing the "life clock" system, I have achieved the following outcomes. Firstly, I successfully use a heart rate sensor which is connected to an Arduino board to control the color of digital butterflies in Processing. Based on Color Psychology, different colors will represent different moods. Since different values of heart rate can also indicate various moods, I combine the two thoughts and use heart rate to control color. Secondly, I used heart rate values to control five servos. One acts as the clock hand and others are used to control mechanical flowers. Every time a heart rate value is read, one certain flower will bloom and the led light inside it will be on. The flowers and lights in their clockwise direction will also open in turn.

Apart from the outcomes I achieved in the system, there are still several limitations in the project. Firstly, the digital butterflies in Processing don’t seem natural. Although they can move in random directions at random speeds, they cannot turn their direction during the flight and cannot flap their wings. Secondly, since the mechanical flowers are hand-made, they are different from each other. In this way, the degree the flowers are opened and closed is different. Under this circumstance, the clock may look a bit messy. Last but not least, due to the limitation of the number of servos which can be connected to our computer, I can only put four mechanical flowers on the clock face.

After completing the project, there are several works I can do in future work to improve my project. Firstly, I need some research about how to make the digital butterflies change direction randomly during their flying process. Secondly, I may need to change the way I made the flower petal rotate. There may be some better ways for them to rotate more smoothly without being influenced by the errors made by manual production. Thirdly, I will make a bigger clock face and put more mechanical flowers on it. In this way, I hope it will look more impressive.

Reference


Christopher. 2020. “Symbolism of Light (8 Surprising Meanings).” Symbolism and Metaphor. November 18, 2020. https://symbolismandmetaphor.com/light-meaning/.

London Image Institute . 2020. “Color Psychology: How Do Colors Affect Mood & Emotions?” London Image Institute. February 5, 2020. https://londonimageinstitute.com/how-to-empower-yourself-with-color-psychology/.

Marketing, Think. 2020. “7 Main Reasons That Could Drive You to Waste Time.” Think Marketing. November 18, 2020. https://thinkmarketingmagazine.com/7-reasons-that-could-drive-you-to-waste-time/.

Nishadha. 2018. “5 Amazing Advantages of Visual Communication You Can’t Ignore - Creately Blog.” Creately Blog. October 9, 2018. https://creately.com/blog/diagrams/visual-communication-benefits/.

“13 Shocking Facts about How We Waste Time.” 2018. Hugh Culver. March 31, 2018. https://hughculver.com/13-shocking-facts-about-how-we-waste-time/.