Physical-Computing

GitHub link：https://github.com/Fennel-hub/Physical-Computing-

WEEK6

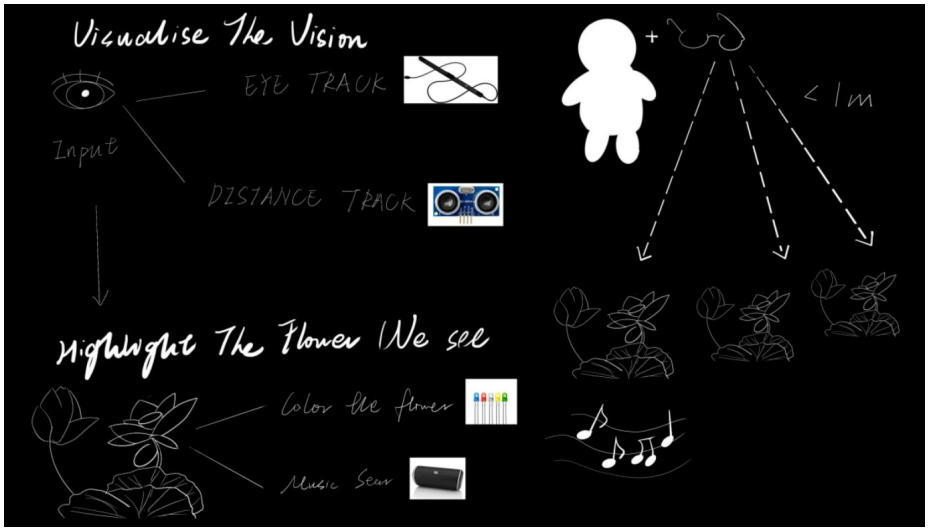
Brainstorm:

I learned about the concept of quantum entanglement a long time ago, and I was very interested in it and wanted to use it as my project concept. But the concept is very abstract and obscure, so I read a lot of books and papers explaining the concept to learn more about its practical applications in life. After learning more about quantum entanglement, I was surprised to find a striking similarity between this concept and my favorite Chinese philosophical theory "idealism", so I really wanted to combine the physical concept of quantum entanglement with the traditional Chinese philosophical theory and express it in my work. Based on this, I designed my first Arduino installation.

The expression of this installation is inspired by Wang Yangming's story "Flowering in the Rock" about "idealism". In this story, Wang Yangming's friend pointed to a flowering tree in the rock and asked, "There is nothing outside of my mind. Wang Yangming replied, "When you do not look at this flower, it will return to silence with you; when you come to see it, the color of the flower will come to you for a moment, and you will know that the flower is not outside your mind."

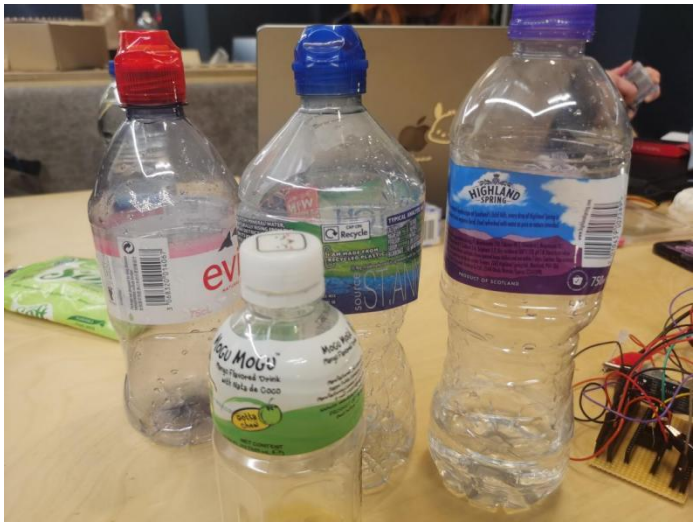
In order to express this story in a concrete way, I designed the shape of the installation as a cluster of flowers, hoping that whichever flower the eye looks at will light up, and at the same time music will play to create the atmosphere of this flower being different. Therefore, the input is designed as an eye-tracking device and ultrasonic distance sensor, and the output is designed as RGB LED and DF player.

I hope to convey the theme of "everything is changed for me" to the experience.



WEEK7

This week I got infected with the new crown virus, so very little work progressed. The main work was designing the look of the Arduino. My initial idea was to make a plastic water bottle into a flower, which was made by heat shrink gun and glue. However, during practice, I found that I could not control the shape of the flower well, and the finished flower was too small and not visually appealing enough, so I gave up the idea.



After rethinking, since the project is based on Wang Yangming's theory of mindfulness, I wanted to reflect the characteristics of traditional Chinese culture. I modified the packaging design to be more Chinese. The material was changed to dreamy and light organza, and the color was changed to light pink and black to simulate Chinese ink painting. The type of flower was determined to be the lotus, a traditional Chinese flower.

I went to the fabric store in Oxford Street to buy organza and wire.

WEEK8

I made a tutorial with Phoenix for online tutoring this week, and after communication, I found out that the eye sensor cannot be used as Arduino input and needs to be modified.

In order to find a sensor that fits the Arduino input and the theory of teleology, I studied and understood more about Arduino. I changed the original concept of "what the eye can see" to "what the mind can see", which not only fits better with the "theory of mind", but also can change the eye sensor into a pulse sensor that fits the Arduino input. This not only fits better with the "theory of mind", but also allows me to change the eye-tracking device to a pulse sensor that fits the Arduino input.

After determining the final input, I started designing the exact representation of the output. The outputs were LEDs and DFplayer.

At first, RGBLEDs, designed to change seven colors of light, were used. But during the experiment, I found that the three-headed RGBLED was difficult to connect with other originals and achieve the function. At the same time, the lighting effect is not good, a bit like the cheesy light of the bar, so finally changed to white LED. code design as follows.

```
int red_light_pin= 10;
int green_light_pin = 9;
int blue_light_pin = 8;
void setup() {
  pinMode(red_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);
}
void loop() {
  RGB_color(255, 0, 0); // Red
  delay(1000);
  RGB_color(0, 255, 0); // Green
  delay(1000);
  RGB_color(0, 0, 255); // Blue
  delay(1000);
  RGB_color(255, 255, 125); // Raspberry
  delay(1000);
  RGB_color(0, 255, 255); // Cyan
  delay(1000);
  RGB_color(255, 0, 255); // Magenta
  delay(1000);
  RGB_color(255, 255, 0); // Yellow
  delay(1000);
  RGB_color(255, 255, 255); // White
  delay(1000);
}
void RGB_color(int red_light_value, int green_light_value, int blue_light_value)
 {
  analogWrite(red_light_pin, red_light_value);
  analogWrite(green_light_pin, green_light_value);
  analogWrite(blue_light_pin, blue_light_value);
}
```

The effect video is placed in GITHUB.

After changing to normal LED, at first, I designed the light as a breathing light to reflect the connection with the "heart", the code design is as follows.

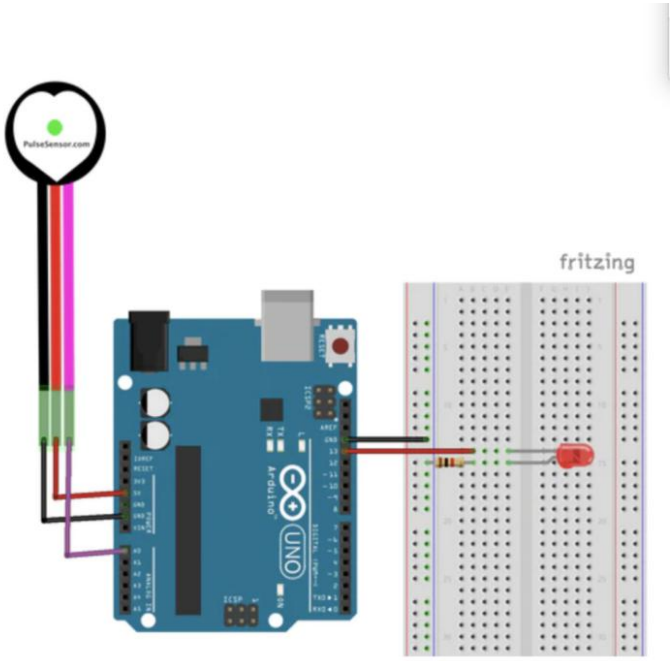
```
//led呼吸灯
void setup(){
    pinMode(13,OUTPUT);
    Serial.begin(9600);


}

void loop(){
    //渐亮
    for (int i = 0; i < 255; i++)
    {
        analogWrite(13,i);
        delay(1);
    }
    //渐暗
    for (int i = 255; i >0; i--)
    {
        analogWrite(13,i);
        delay(1);
    }
```

The effect video is placed in GITHUB.

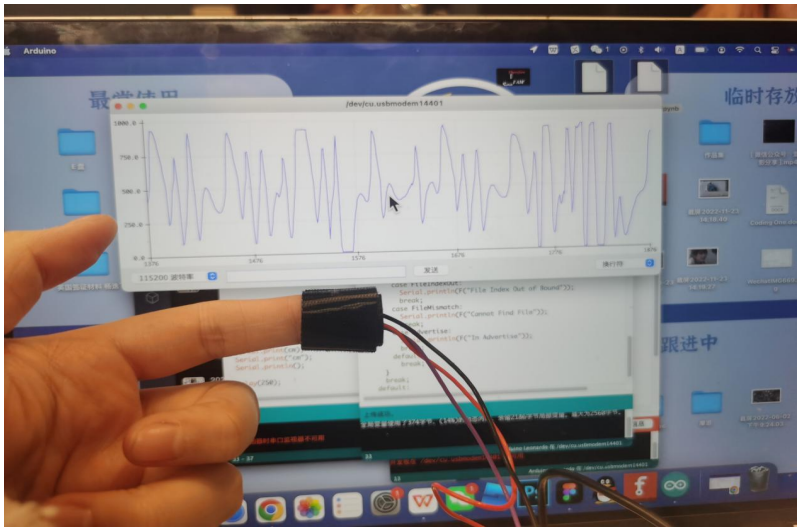
In the third version, I eliminated this design. To better reflect the importance of the "heart", I linked the frequency of the light to the frequency of the heartbeat. When the user wears the pulse sensor, the light will flash according to the user's pulse rate. The code design is as follows

```
//  Variables
int PulseSensorPurplePin = 0;        // Pulse Sensor PURPLE WIRE connected to ANALOG PIN 0
int LED13 = 13;   //  The on-board Arduion LED


int Signal;                // holds the incoming raw data. Signal value can range from 0-1024
int Threshold = 550;            // Determine which Signal to "count as a beat", and which to ingore.


// The SetUp Function:
void setup() {
  pinMode(LED13,OUTPUT);         // pin that will blink to your heartbeat!
   Serial.begin(9600);         // Set's up Serial Communication at certain speed.

}

// The Main Loop Function
void loop() {

  Signal = analogRead(PulseSensorPurplePin);  // Read the PulseSensor's value.
                                              // Assign this value to the "Signal" variable.

   Serial.println(Signal);                    // Send the Signal value to Serial Plotter.


   if(Signal > Threshold) {                          // If the signal is above "550", then "turn-on" Arduino's on-Board LED.
     digitalWrite(LED13,LOW);
   }
   else {
     digitalWrite(LED13,HIGH);                //  Else, the sigal must be below "550", so "turn-off" this LED.
   }


delay(10);
```

The circuit diagram is designed as follows.



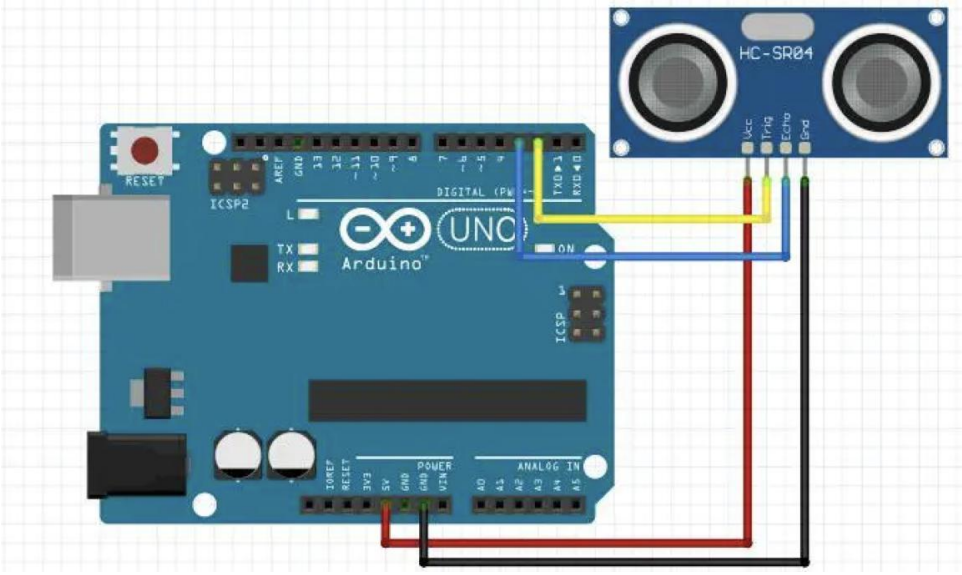
The rendering is as follows, and the effect video is placed in GITHUB.



Distance sensor as atrigger, only after the device is close enough to light up, but also to show the impact of people on the physical world. The specific code is as follows.

```
// Read the signal from the sensor: a HIGH pulse whose
// duration is the time (in microseconds) from the sending
// of the ping to the reception of its echo off of an object.
pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);
if(sonar.ping_cm()<=30){
  digitalWrite(LED13,HIGH);
}
else{digitalWrite(LED13,LOW);
}
Serial.println( sonar.ping_cm() );
delay( 100 );
```

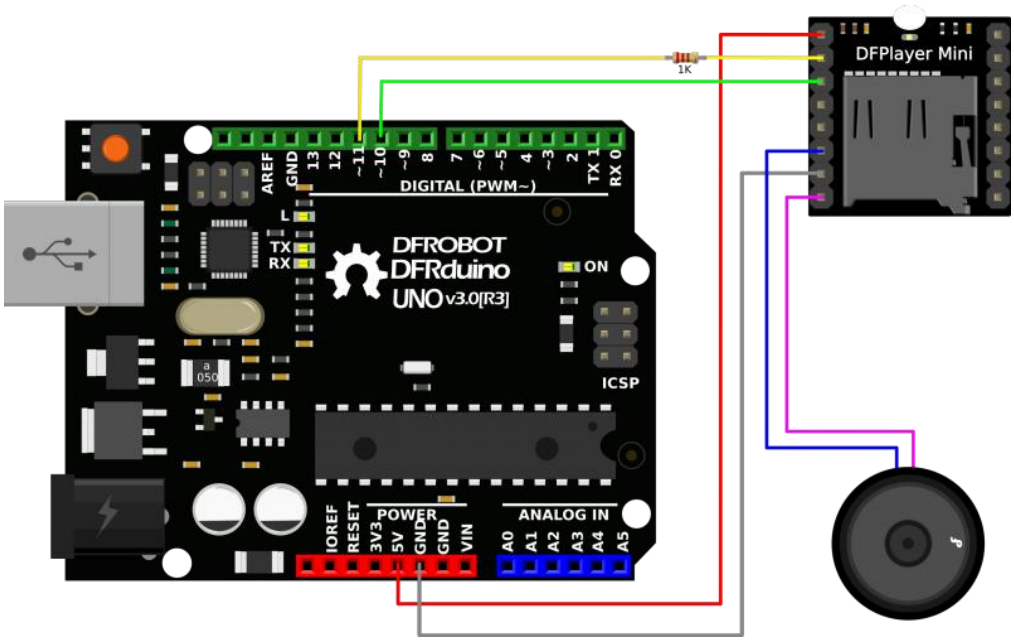
The circuit diagram is designed as follows.



The output DFplayer is also linked to the pulse sensor and will play music when worn.
At first, the sound played was a humming sound. After checking the data, I found that I need to add a resistor and it will play smoothly.
The code design is as follows.

```
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true);
  }

  Serial.println(F("DFPlayer Mini online."));

  myDFPlayer.volume(0);
  delay(100);

  myDFPlayer.volume(30);  //Set volume value. From 0 to 30
  myDFPlayer.play(1);  //Play the first mp3. to loop, exchange 'play' for 'loop'
}

void loop()
{
  if (myDFPlayer.available()) {
    printDetail(myDFPlayer.readType(), myDFPlayer.read());
  }
}

void printDetail(uint8_t type, int value){
  switch (type) {
    case TimeOut:
      Serial.println(F("Time Out!"));
      break;
    case WrongStack:
      Serial.println(F("Stack Wrong!"));
      break;
    case DFPlayerCardInserted:
      Serial.println(F("Card Inserted!"));
      break;
    case DFPlayerCardRemoved:
      Serial.println(F("Card Removed!"));
      break;
    case DFPlayerCardOnline:
      Serial.println(F("Card Online!"));
      break;
    case DFPlayerPlayFinished:
```

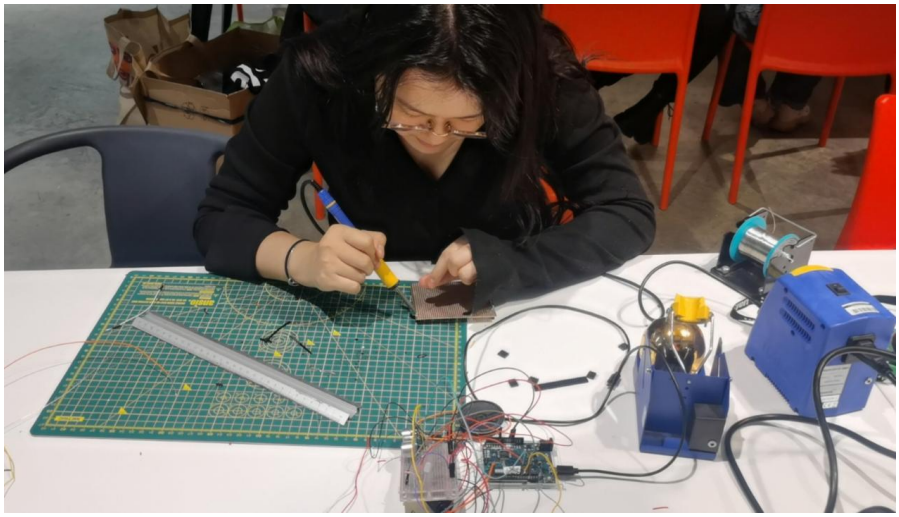
The circuit diagram is designed as follows.

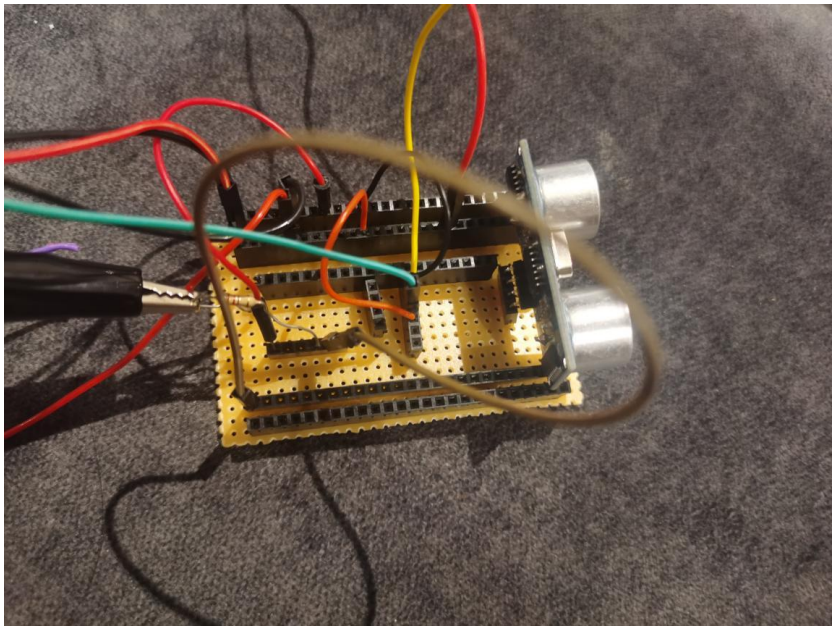


The above functions are programmed and implemented on the Arduino, and the circuit is connected on the breadboard.

WEEK9

This week for modification, welding and shape making.
Welding:





In order to further reflect the theme of "everything is changed for me", I changed the ultrasonic distance sensor control LED in a certain distance to light up this function. The intensity of the light changes with distance: the closer the distance, the brighter the light.

This piece of code is very difficult to implement, I spent a long time learning and debug to be able to solve. Because there is no hardware PMW for the light input, you have to use the timer library timer one to control the lights.
The code is as follows.

```
void blinkLED(void)
{
  js++;
  if(js>100)
    js=0;
  if(js< 100-Value_cm)
  {
    digitalWrite(led, 1);
  }
  else
  {
    digitalWrite(led, 0);
  }
}

float getddistance(int Trig,int Echo)
{
  pinMode(TrigPin, OUTPUT);
  pinMode(EchoPin, INPUT);
  digitalWrite(Trig, LOW); //低高低电平发一个短时间脉冲去TrigPin
  delayMicroseconds(2);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  //读取一个引脚的脉冲（HIGH或LOW）。例如，如果value是HIGH，pulseIn()会等待引脚变为HIGH，开始计时，再等待引脚变为LOW并停止计时。
  //返回脉冲的长度，单位微秒。如果在指定的时间内无脉冲函数返回。
  //此函数的计时功能由经验决定，长时间的脉冲计时可能会出错。计时范围从10微秒至3分钟。（1秒=1000毫秒=1000000微秒）
  //接收到的高电平的时间（us）* 340m/s / 2 = 接收到高电平的时间（us） * 17000 cm / 1000000 us = 接收到高电平的时间 * 17 / 1000 （cm）
  return float( pulseIn(Echo, HIGH) * 17 )/1000; //将回波时间换算成cm

}
```

Finalized the outer package design and production.

The overall color scheme imitates traditional Chinese ink painting. The lotus leaves are made of black organza, and the lotus flowers are made of white and light pink with a sense of jaggedness.