Journal by Yujie Gao

Team members:Yujie Gao, Pinsi Wang

Week5

Project week

After confirming the group members, we worked on our projects individually. I got inspiration through videos and pictures, and conceived of several possible solutions that could be implemented.

Plan1. Birthday wishes

In my life, as I get older, my close friends around me have started to put in work and there is no way to spend birthdays together like when I was a child, so I receive fewer birthday wishes. How can I receive more birthday wishes on my birthday? My idea is that I can make a cartoon character with a cute shape, using ultrasonic sensing combined with an Mp3 module, as well as a light sensor and a helm. This would allow the birthday person to hear many birthday wishes from friends throughout the day, as if the friends were around.

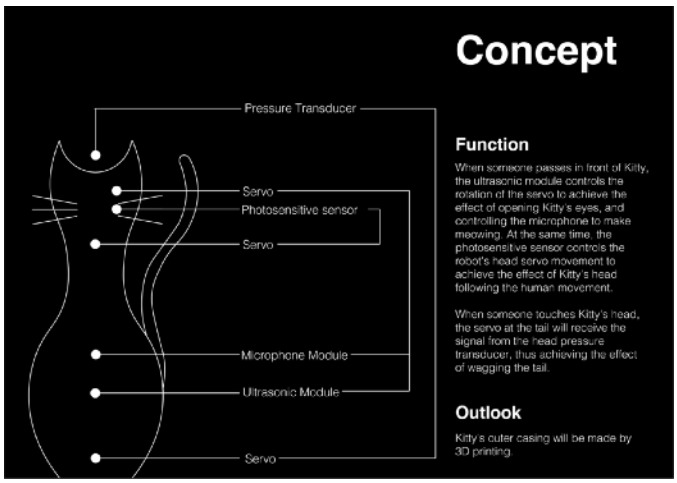
Plan2. Electronic cat

Many young people want to have a pet but have to give up the idea because of their jobs, and the expenses of having a pet can be quite high nowadays, which is even more of a reason not to have a pet for young people who have just come out of society. The main source of my inspiration is the epidemic at the moment, many students have to stay at school all the time, but they are not allowed to have pets at school, so many of them make their own pets by hand using I was inspired by this and wanted to design a mechanical pet to meet the needs of young people today.

After this Wang and I had our first group discussion and eventually confirmed the general direction of the pet theme, but we changed the context of some of the opening questions and the interactive effects of some of the sensors.

To explore and study the development of cats in the future world, with the development of the times and the advent of digital technology, with the continuous improvement of human civilisation, how will cats develop

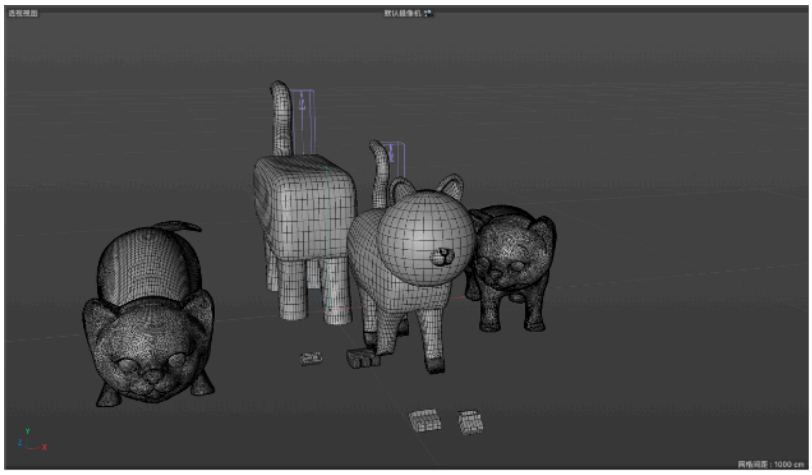
in the course of human development, and what will the pet cat of the future be like, this is the topic we need to study.
To address this issue, we firstly identified three approaches, we needed to implement the internal mechanical structure through Arduino, secondly for the design of the cat we considered using cardboard boxes or wood carvings or 3d modeling to achieve the design. Secondly, we initially wanted to try using sound sensors in combination with a servo, but we found that the results were not good, then we tried ultrasonic sensors connected to the servo and the results were significantly improved. design using a servo, photosensitive sensor, ultrasonic module, microphone sensor and pressure sensor, placing the pressure sensor on the cat's head. When someone strokes the cat's head, the servo connected to the cat's tail will start to work, thus controlling the wagging of the cat's tail. Through the working principle of the photosensitive sensor, when someone passes in front of the cat, the cat's head will start to wag, thus feeling like the cat is always staring at the people passing by. Finally, there is the ultrasonic sensor, the ultrasonic sensor working process is mainly when someone passes by, the ultrasonic sensor will make The ultrasonic sensor works by capturing when someone passes by, thus controlling the servo to open the eyes and the microphone sensor to make the cat's cry. We used a total of two inputs and three outputs to give the cat a more mechanical element to its movement. Secondly, Wang and I created a PowerPoint for presentation for the opening of the project, using our ideas as a starting point, and Wang refined the content of the PowerPoint.



Week6

In the second week after the presentation we signed up a tutorial based on the teacher's advice from the presentation. We therefore took the teacher's advice and chose to start with other sensors first.

Our main work this week was to shop for sensors and to design the appearance of the cat, as we needed to complete the modeling, so I chose to sketch the diorama directly in C4D. body and tail parts were modeled separately.

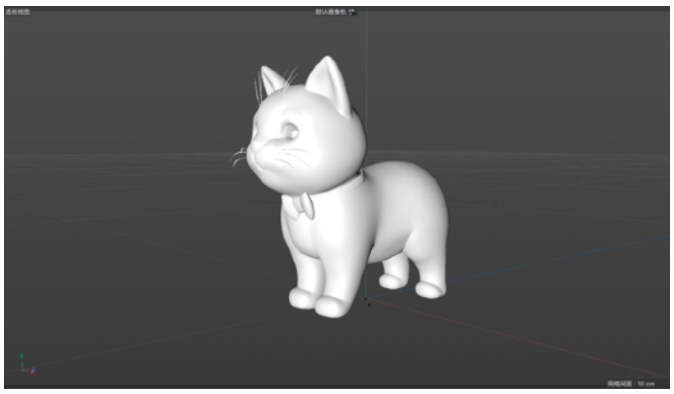


As we have a pressure sensor, we tried the pressure sensor first. We found out through code testing that the pressure sensor has to use a lot of pressure to reach the value we need, so we are going to give the pressure sensor a value first.
Because ours was going to use the pressure sensor on the head of the cat, we needed to set aside a suitable place to store the pressure sensor when modeling it. Secondly through understanding the pressure sensor I found out that there were relatively many modules to use for the pressure sensor and this was one of the things we needed to consider.

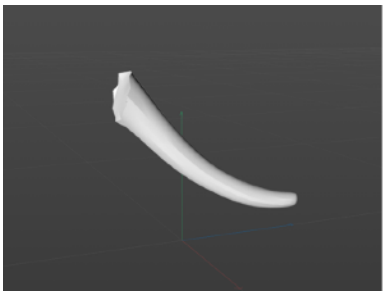
```
project1
#include<Hx711.h>
#include<Servo.h>
#include<SoftwareSerial.h>

SoftwareSerial mySerial(0,1);
float Weight=0
int pos=0;
int HX711=4;
Servo myservo;

Void setup()
{
  Serial.begin (9600) ;
  Init_Serial();
  Init_HX711();
  myservo.attach(9,500,2500)

  Serial.print("Welcome to use!\n");
  Get_Maopi();
  delay(3000);
  }

  void loop()
  {
    weight=-int(Get_Weight())-1300;
    if(weight<0)
    {
      weight=-weight;
    }
    Serial.print()
    Serial.println(weight);

    if(weight>=20)
    pos=60
    else{
      pos=0
      }
    }
```

I then discussed the shape of the cat with Wang and we finally combined the sketches and we settled on the shape of the cat, the final sketch was drawn by Wang.

Week7
I modeled the cat from the final sketch produced by Wang, again using C4D.

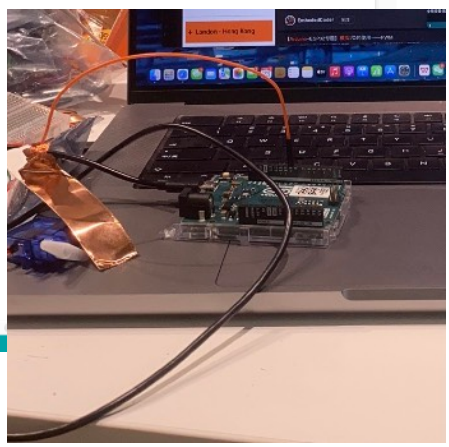


During my sketching process, we modified the use of the sensors through the advice of the seniors. As the pressure sensor had more elements, he recommended that we use copper tape instead of the pressure sensor, which would simplify the process but achieve the same effect, and also allow us to attach the copper tape with insulation tape directly to the top of the cat's head to make it look better overall. So I took the advice of my seniors and replaced the pressure sensor with copper tape. When the resistance of the pressure sensor changes at the touch, the servo drives the cat's tail. I therefore divided the cat into two parts in the modeling process, and Wang was responsible for the tail making.

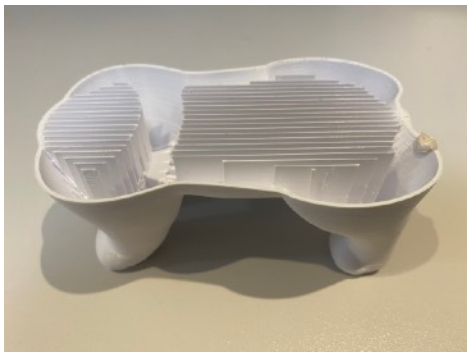


After modeling I signed up for a 3D printing course and then I was ready to try my hand at 3D printing the model, but I found that the school's 3D printer was not capable of printing very complex models, so just in case I removed the collar and bow from the cat's neck. The bow and collar could be handmade from some fabric and also achieved the effect of covering the ultrasonic sensor. I then

```
sketch_copper
int sensorValue = 0;
void setup()
{
 pinMode(7,INPUT);
 Serial.begin(9600);
}

void loop()
{
  sensorValue = digitalRead(7);
  Serial.println(sensorValue);
  delay(100);
}
```



tested the copper tape and found that it was easier and better than the pressure sensor to achieve our results, but in doing so I also found problems with the copper tape. There were some errors in the tape, sometimes the resistance of the tape changed when the hand was just hovering over the tape, but it turned out that we were missing a resistor to cause this, and after adding the resistor I tried a few more times with no further errors.
I then booked in to the 3D printer and started printing our model, as it was quite long and couldn't be done in a day, so half way through the print in the evening the print stopped as we ran out of PLA wire, and



when I added to the line the next day to continue the print, I found that it was already off, so that was the end of the first version of the print.
We then started the second version and, having learnt from the first, I changed the wire for the second time and thought I had finished, but I still printed half way through, but the machine went through and finished, which forced me to print a third time. The third time I cut the model in half and chose to print the top half of the model directly and ended up with a hot melt gun to glue them together, this would have saved almost half the time on the print but as the familiar machines were already being used I had to pass the model onto the software for the small printer but the Original Prusa Drivers did not have the default support and the outer wall The density wasn't adjusted properly and this time the print came out as a grinning cat.
Having learnt from the previous lessons, I finally finished the model section after the fourth print,

and during the week also completed the sanding of the model and the booking of the laser cutting course.

```
sketch_servo §
#include<Servo.h>
int pos=0;
int sensorValue = 0;
Servo myservo;

void setup()
{
  myservo.attach(9);
 pinMode(7,INPUT);
 Serial.begin(9600);
}

void loop()
{
  sensorValue = analogRead(7);
  Serial.println(sensorValue);
 if(sensorValue>=350){
for (pos = 0; pos <= 60; pos ++) {
    myservo.write(pos);
    delay(5);
  }
  for (pos = 60; pos >= 0; pos --) {
    myservo.write(pos);
    delay(5);
  }
}
else{
myservo.write(0);
delay(100);}
  }
```

Next I tried to combine the copper tape with the servo to control the rotation of the servo by the change in resistance of the copper tape, but after combining the two, I found that the numbers on the serial monitor did not jump very much, and when at rest, the numbers did not stay within a controlled range for a long time, which I tried several times.

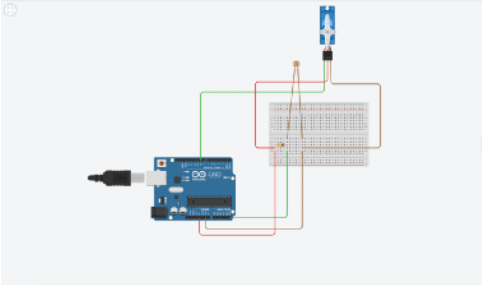
Week8

```
servo_search §
#include<Servo.h>
int pos=30;
int sensorValue = 0;
bool xunhuan=false;
Servo myservo;

void setup()
{
  myservo.attach(9);
 pinMode(7,INPUT);
 Serial.begin(9600);
}

void loop()
{
  sensorValue = analogRead(7);
  Serial.println(sensorValue);
 if(sensorValue>=350 && xunhuan==false){
  xunhuan=true;
for (pos = 30; pos <= 60; pos ++) {
    myservo.write(pos);
    delay(5);
  }
  for (pos = 60; pos >= 30; pos --) {
    myservo.write(pos);
    delay(5);
  }

  }
  if(sensorValue>=350 && xunhuan==true){
    xunhuan=false;
for (pos = 30; pos >= 0; pos --) {
    myservo.write(pos);
    delay(5);
  }
  for (pos = 0; pos <= 30; pos ++) {
    myservo.write(pos);
    delay(5);
  }
}
if(sensorValue<350){
myservo.write(30);
delay(100);}
   }
```

I laser cut the my sketch of the bow, then I refined the code for the copper tape control servo from the previous week, this time I found that the resistance values were back to normal, but the servo would sometimes jitter, but sometimes it would work again.

Effect of ultrasonic sensor combined with bow tie

In response to this problem, I suspected that the copper tape might still be the cause of the problem, so I tried replacing the copper tape with a photosensitive sensor, firstly to test whether the photosensitive sensor would work, and secondly to combine the servos.



```
sketch_mix1 §
#include<CapacitiveSensor.h>
#include<Servo.h>
CapacitiveSensor  Sensor =  CapacitiveSensor(4,6);
long val;
int pos=30;
Servo myservo;

void setup()
{
  myservo.attach(9);
  Serial.begin(9600);
  pinMode(9,OUTPUT);
}

void loop()
{

  val=Sensor.capacitiveSensor(30);
  Serial.println(val);
  if(val >= 1000)
  for (pos = 30; pos <= 60; pos ++) {
    myservo.write(pos);
    delay(500);
    for (pos = 60; pos >= 30; pos --) {
    myservo.write(pos);
    delay(500);
  }

     if(val <1000)
     myservo.write(30);
     delay(1000);}
  }
```

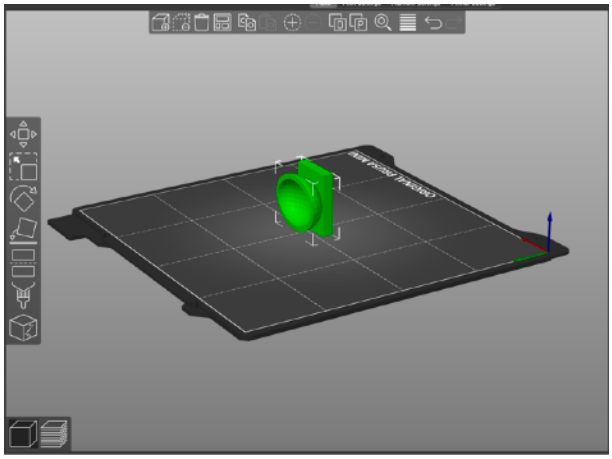
Wang tried to get the cat's eyelids to open and close the effect, but failed. However, she chose to use small light bulbs to control the effect of the eyelids instead of a tiller. She tested the operation of the ultrasonic sensor and the small bulb separately and combined them in such a way that when the ultrasonic sensor recognises that someone is in range, the small bulb will start to light up and when the person leaves the range, the bulb will stop working.

```
sketch_distance1.0
int TrigPin = 8;
int EchoPin = 9;
int LedPin = 12;
int LedPin1 = 13;
int val;
void setup()
{
Serial.begin(9600);
pinMode(TrigPin, OUTPUT);
pinMode(EchoPin, INPUT);
pinMode(LedPin,OUTPUT);
pinMode(LedPin1,OUTPUT);
}
void loop()
{
digitalWrite(TrigPin, LOW);
delayMicroseconds(2);
digitalWrite(TrigPin, HIGH);
delayMicroseconds(10);
int cm = pulseIn(EchoPin, HIGH) / 20.0;
cm = (int(cm * 100.0)) / 100.0;
Serial.println(cm);
Serial.println("cm");
delay(50);
if (58>=cm)
{
digitalWrite(LedPin, HIGH);
    delay(20);
digitalWrite(LedPin1, HIGH);
    delay(20);
}else{
digitalWrite(LedPin,LOW);
digitalWrite(LedPin1,LOW);
}
}
```

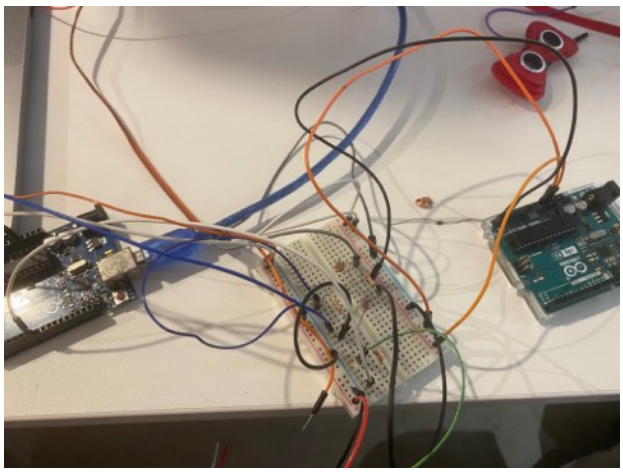
I then added the Mp3 module to the code, using the ultrasonic sensor to control the Mp3 module to play a pre-recorded cat call and

```
DFPlayer2.0
#include "SoftwareSerial.h"
#include "DFPlayer_Mini_Mp3.h"
int TrigPin = 8;
int EchoPin = 9;
int LedPin = 12;
int LedPin1 = 13;
int val;
SoftwareSerial mySerial(10, 11);

void setup()
{
Serial.begin(9600);
mySerial.begin(9600);
mp3_set_serial(mySerial);
mp3_set_volume(15);
while (!Serial){
  ;
}
Serial.println(1);
pinMode(TrigPin, OUTPUT);
pinMode(EchoPin, INPUT);
pinMode(LedPin,OUTPUT);
pinMode(LedPin1,OUTPUT);
}
void loop()
{
digitalWrite(TrigPin, LOW);
delayMicroseconds(2);
digitalWrite(TrigPin, HIGH);
delayMicroseconds(10);
int cm = pulseIn(EchoPin, HIGH) / 20.0;
cm = (int(cm * 100.0)) / 100.0;
Serial.println(cm);
Serial.println("cm");
delay(50);
if (58>=cm)
{
digitalWrite(LedPin, HIGH);
    delay(20);
digitalWrite(LedPin1, HIGH);
    delay(20);
 mp3_play(1);
}else{
digitalWrite(LedPin,LOW);
digitalWrite(LedPin1,LOW);
mp3_stop();
}
}
上传成功。
```

control the small light to light up. We added a layer of translucent material to the outside of the small light to create the effect of an eyeball.

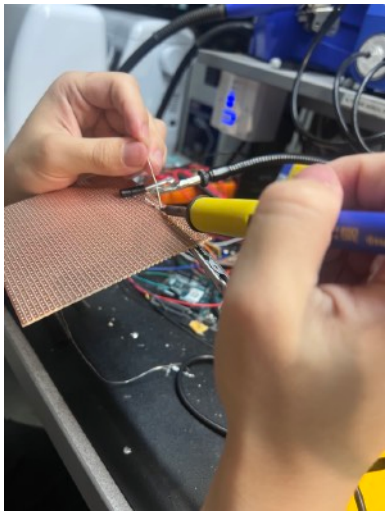


Week9



After drawing the wiring diagram in tinkercad and completing the code to run through the breadboard, I soldered all the wires needed on both breadboards. When I first started soldering, I often soldered wrong because I was rather rusty, but I got more familiar with it after a few times.

However, in the process of soldering, I found that if I soldered both sides of the Mp3 module, there would be a short circuit problem, because the left and right sides of the board are connected in series, the upper left corner is positive but the upper right corner is not, so I was torn by this



problem for a long time. When I first started soldering I soldered the resistors wrong twice in quick succession, the resistors needed to be connected to the board as they were in parallel on the breadboard but on the solder board I needed to connect them in series to run, but I also needed to scrape the copper off the middle otherwise there would be a short circuit problem too.

After soldering I retried testing the circuit but I found that the Mp3 module was not under control after the circuit was soldered and by replacing the Mp3 module it still did the same thing so I tried changing the code. The final result of the code was to allow the ultrasonic sensor to control the small light and the Mp3 module via the board.

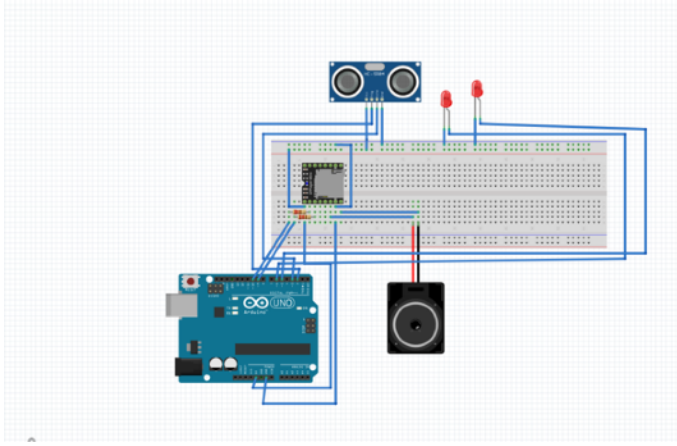
DFPlayer3.0 §

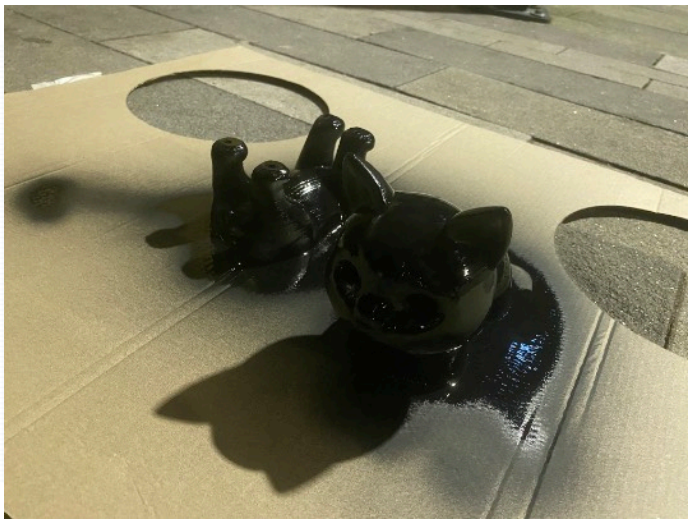
```
#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"
int TrigPin = 8;
int EchoPin = 9;
int LedPin = 12;
int LedPin1 = 13;
int val;
SoftwareSerial mySerial(10, 11);
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600);
  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));
  if (!myDFPlayer.begin(mySerial)) {
    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(10);  //Set volume value. From 0 to 30
  myDFPlayer.play(1);
  pinMode(TrigPin, OUTPUT);
  pinMode(EchoPin, INPUT);
  pinMode(LedPin, OUTPUT);
  pinMode(LedPin1, OUTPUT);
}
void loop()
{
 // static unsigned long timer = millis();

  digitalWrite(TrigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(TrigPin, HIGH);
  delayMicroseconds(10);
  int cm = pulseIn(EchoPin, HIGH) / 20.0;
  cm = (int(cm * 100.0)) / 100.0;
  Serial.println(cm);
  Serial.println("cm");
  delay(50);
  if (40 >= cm)
  {
    digitalWrite(LedPin, HIGH);
```

DFPlayer3.0 §

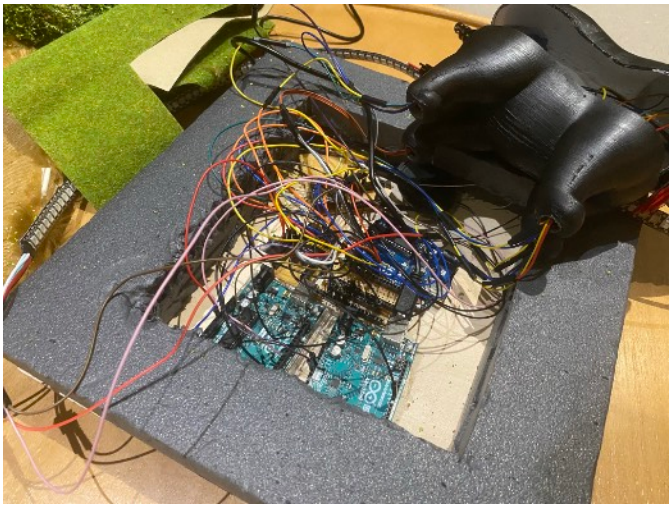
```
    digitalWrite(LedPin, HIGH);
    digitalWrite(LedPin1, HIGH);
//    timer = millis();
    myDFPlayer.play(1);  //Play next mp3 every 3 second.
    delay(1000);
//    if (myDFPlayer.available()) {
//      printDetail(myDFPlayer.readType(), myDFPlayer.read());
//
//    }
  }
  digitalWrite(LedPin, LOW);
  digitalWrite(LedPin1, LOW);
}

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:
      Serial.print(F("Number:"));
      Serial.print(value);
      Serial.println(F(" Play Finished!"));
      break;
    case DFPlayerError:
      Serial.print(F("DFPlayerError:"));
      switch (value) {
        case Busy:
          Serial.println(F("Card not found"));
          break;
        case Sleeping:
          Serial.println(F("Sleeping"));
          break;
        case SerialWrongStack:
          Serial.println(F("Get Wrong Stack"));
          break;
        case CheckSumNotMatch:
          Serial.println(F("Check Sum Not Match"));
```

```
DFPlayer3.0 §
      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:
      Serial.print(F("Number:"));
      Serial.print(value);
      Serial.println(F(" Play Finished!"));
      break;
    case DFPlayerError:
      Serial.print(F("DFPlayerError:"));
      switch (value) {
        case Busy:
          Serial.println(F("Card not found"));
          break;
        case Sleeping:
          Serial.println(F("Sleeping"));
          break;
        case SerialWrongStack:
          Serial.println(F("Get Wrong Stack"));
          break;
        case CheckSumNotMatch:
          Serial.println(F("Check Sum Not Match"));
          break;
        case FileIndexOut:
          Serial.println(F("File Index Out of Bound"));
          break;
        case FileMismatch:
          Serial.println(F("Cannot Find File"));
          break;
        case Advertise:
          Serial.println(F("In Advertise"));
          break;
        default:
          break;
      }
      break;
    default:
      break;
  }
}
```





The next step was to paint the model, as I chose a black look, but we also needed to add some white accents like a nose to the exterior, so we also used white acrylic paint, which we painted on with a brush.



Finally, we wired the circuitry into the body of the cat, placed the solder board and the uno board in the base we designed, made a nest for the kitten and added some decorations to complete the project.