Congcong Xie 21009117

Creative Computing

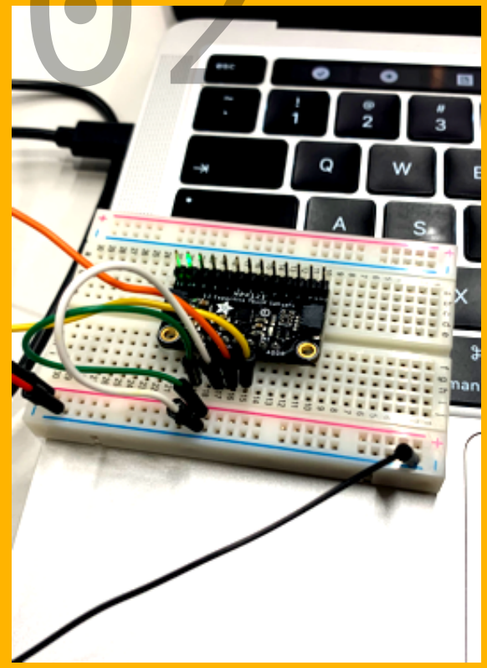
PORTFOLIO

01

Extended Hand

First use the straw as the skeleton of the hand, and then connect it with a thin wire. Make a pressure sensor, then stick it on your hand with hot melt glue, connect the Uno board and the buzzer, upload the code, and a manipulator that can sing is done!

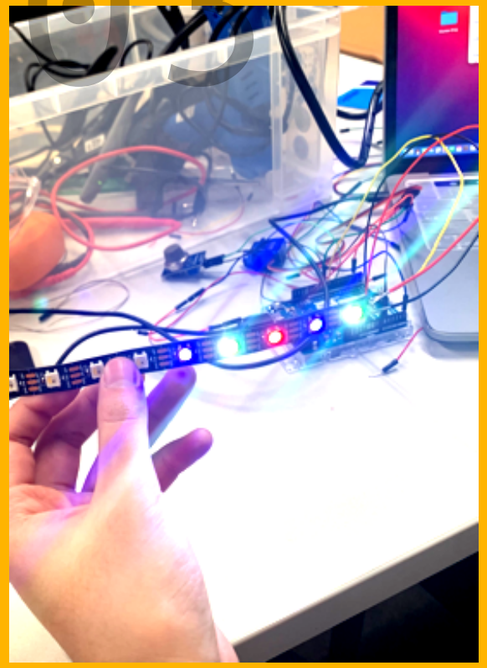
02



Analog piano

This is a small analog piano, it can simulate the 7 syllables of the piano, and you can even use it to play a piece of music, which is very interesting!

03



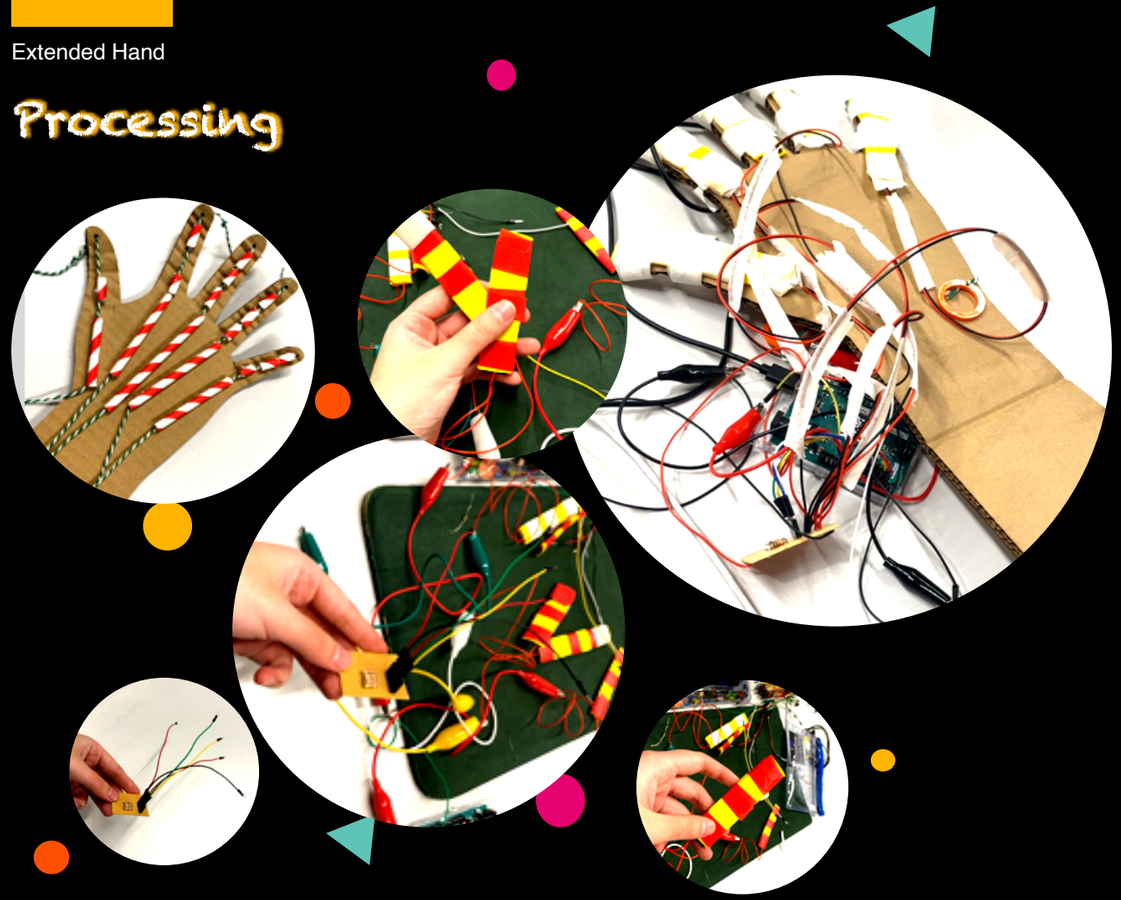
LED light strip

This is an LED light strip that gradually lights up according to a specific frequency. The light strip is connected to the Uno board through a soldered wire. Then write the code according to the desired frequency, and the light strip will light up according to your code instructions!

01

Extended Hand

Processing



Coding

```
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978
```

```
void setup()
{
  pinMode(9,OUTPUT);
  pinMode(13,INPUT);
}

void loop()
{
  if(true)
  {
    int data1 = analogRead(A0); //take an analog read on the knob to read resistance(0 to 1024)
    int data2 = analogRead(A1);
    int data3 = analogRead(A2);
    int data4 = analogRead(A3);
    int data5 = analogRead(A4);

    Serial.print(data1);
    Serial.print("\t ");
    Serial.print(data2);
    Serial.print("\t ");
    Serial.print(data3);
    Serial.print("\t ");
    Serial.print(data4);
    Serial.print("\t ");
    Serial.println(data5); // print it out

//根据开关是否按下发出不同音调。

if(data1 <= 200) tone(9,687,10);
  else noTone(9);

if(data2 <= 500)

   tone(9,810,10);
   else noTone(8);


if(data3 <= 180)
{
  sound2();
}

  if(data4 <= 200)
{
 sound3();
}

  if(data5 <= 450)

   tone(9,610,10);
   else noTone(8);


  delay(500);
}

}

void sound2()
{
  int notes[] = {NOTE_D3,  NOTE_A3, NOTE_B3, NOTE_C4, NOTE_D4, NOTE_E4, NOTE_F4, NOTE_G4, NOTE_A4, NOTE_B4, NOTE_C5 };
  int track [] = {7,8,7,5,7,8,4,3,0,8,3,4,7,5,5,7,8,7,5,7,5,4,3,0,8,3,0,4,3,3};
  int durations[] = {8,4,8,4,4,2,2,4,12,8,4,8,4,12,12,8,4,8,4,4,2,2,4,12,8,4,8,4,12,12}; //4一拍，8两拍
  int size = sizeof(track) / sizeof(track[0]);
  for(int i=0; i<=size; i++)
    {tone(9,track[notes[i]], durations[i]*50);}
}

  void sound3()
  {
    int notes[] = {NOTE_G3,  NOTE_AS3, NOTE_B3, NOTE_C4, NOTE_D4, NOTE_E4, NOTE_F4, NOTE_G4, NOTE_A4, NOTE_B4, NOTE_C5 };
    int track [] = {7,8,7,5,7,8,4,3,0,8,3,4,7,5,5,7,8,7,5,7,5,4,3,0,8,3,0,4,3,3};
    int durations[] = {8,4,8,4,4,2,2,4,12,8,4,8,4,12,12,8,4,8,4,4,2,2,4,12,8,4,8,4,12,12}; //4一拍，8两拍
    int size = sizeof(track) / sizeof(track[0]);
    for(int i=0; i<=size; i++)
      {tone(9,track[notes[i]], durations[i]*50);}
  }
```

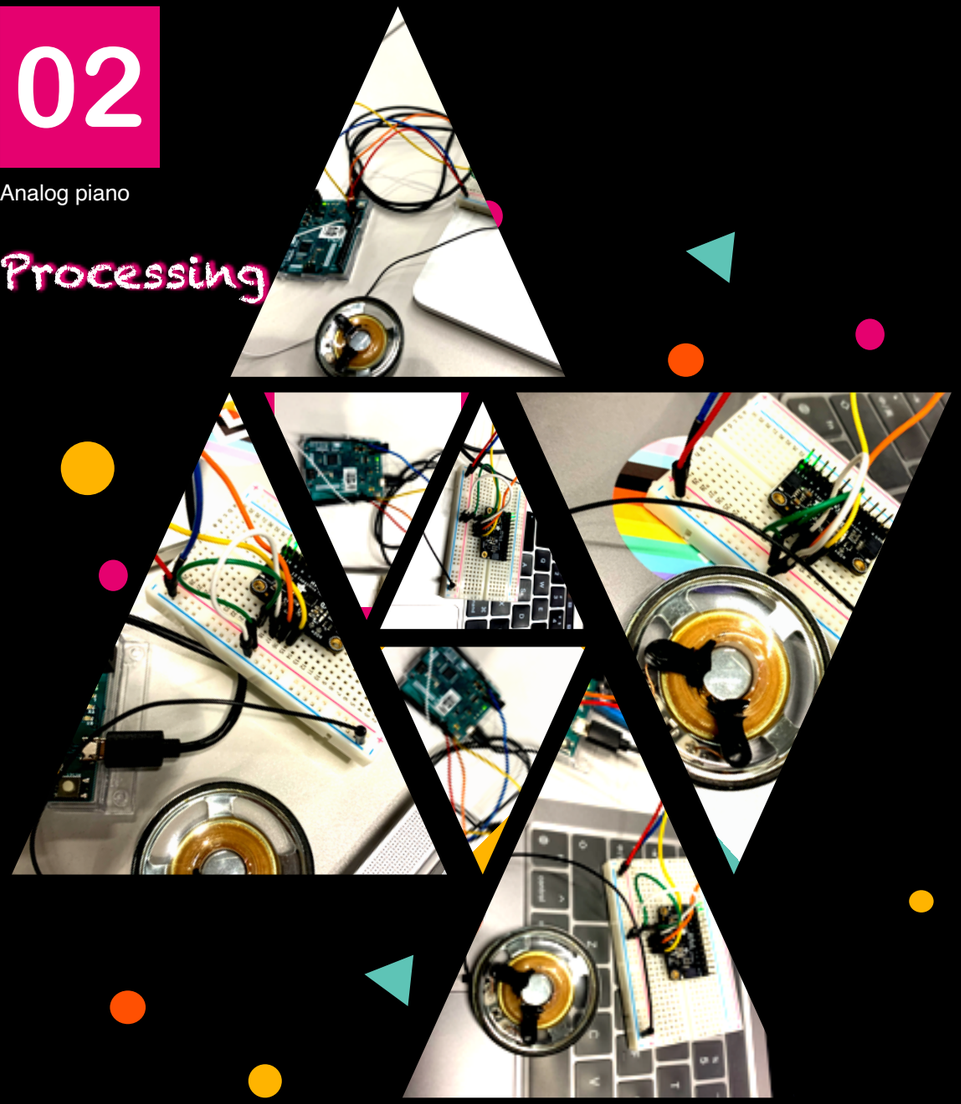
Output

https://youtu.be/eLIDsIH7BSU

02

Analog piano

Processing



Coding

```
#include <Wire.h>
#include "Adafruit_MPR121.h"

#ifndef _BV
#define _BV(bit) (1 << (bit))
#endif

// You can have up to 4 on one i2c bus but one is enough for testing!
Adafruit_MPR121 cap = Adafruit_MPR121();

// Keeps track of the last pins touched
// so we know when buttons are 'released'
uint16_t lasttouched = 0;
uint16_t currtouched = 0;

void setup() {
  Serial.begin(9600);

  while (!Serial) { // needed to keep leonardo/micro from starting too fast!
    delay(10);
  }

  Serial.println("Adafruit MPR121 Capacitive Touch sensor test");

  // Default address is 0x5A, if tied to 3.3V its 0x58
  // If tied to SDA its 0x5C and if SCL then 0x5D
  if (!cap.begin(0x5A)) {
    Serial.println("MPR121 not found, check wiring?");
    while (1);
  }
  Serial.println("MPR121 found!");
}

void loop() {
  // Get the currently touched pads
  currtouched = cap.touched();

  for (uint8_t i=0; i<12; i++) {
    // it if *is* touched and *wasnt* touched before, alert!
    if ((currtouched & _BV(i)) && !(lasttouched & _BV(i)) ) {
      Serial.print(i); Serial.println(" touched");

      tone(9,100*i,50);
      delay(60);
    }
    // if it *was* touched and now *isnt*, alert!
    if (!(currtouched & _BV(i)) && (lasttouched & _BV(i)) ) {
      Serial.print(i); Serial.println(" released");
    }
  }

  // reset our state
  lasttouched = currtouched;

  // comment out this line for detailed data from the sensor!
  return;

  ////////////////////////////////////////////////
  // debugging info, what
  Serial.print("\t\t\t\t\t\t\t\t\t\t\t\t\t 0x"); Serial.println(cap.touched(), HEX);
  Serial.print("Filt: ");
  for (uint8_t i=0; i<12; i++) {
    Serial.print(cap.filteredData(i)); Serial.print("\t");
  }
  ////////////////////////////////////////////////
  // debugging info, what
  Serial.print("\t\t\t\t\t\t\t\t\t\t\t\t\t 0x"); Serial.println(cap.touched(), HEX);
  Serial.print("Filt: ");
  for (uint8_t i=0; i<12; i++) {
    Serial.print(cap.filteredData(i)); Serial.print("\t");
  }
  Serial.println();
  Serial.print("Base: ");
  for (uint8_t i=0; i<12; i++) {
    Serial.print(cap.baselineData(i)); Serial.print("\t");
  }
  Serial.println();

  // put a delay so it isn't overwhelming
  delay(100);
}
```

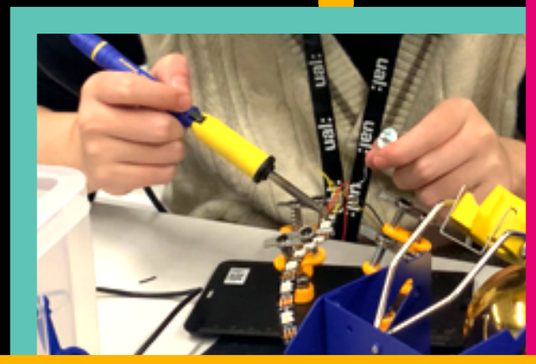
Output

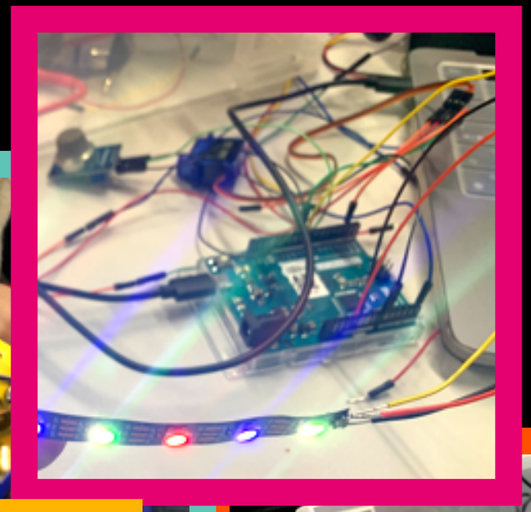
https://youtu.be/Ou2gutQZuvg

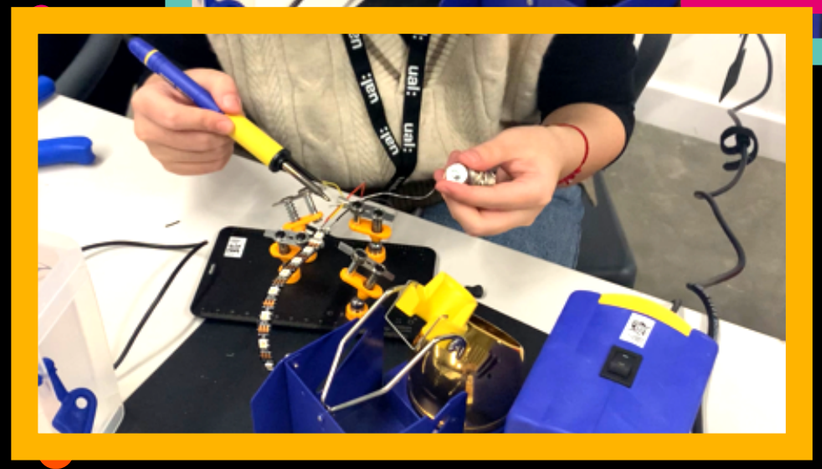
03

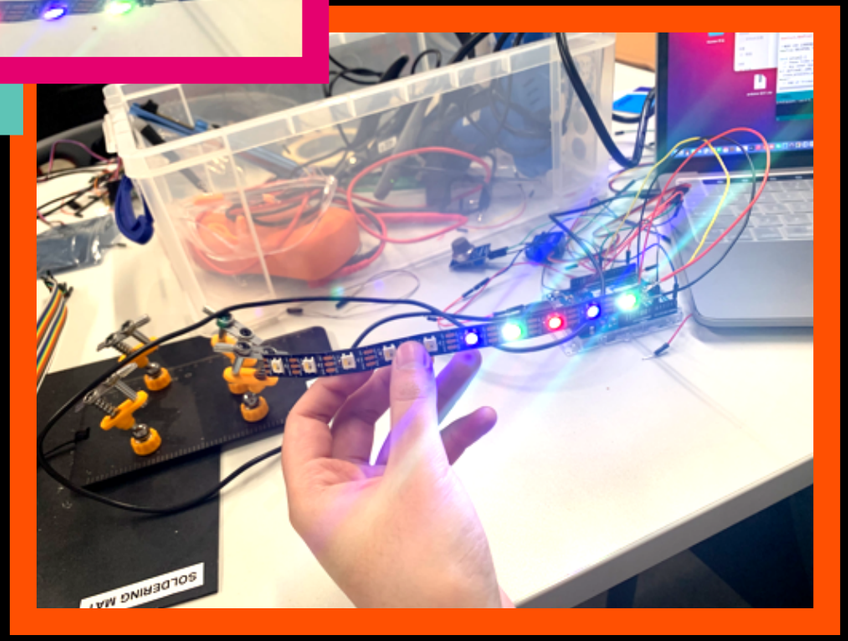
LED light strip

Processing









Coding

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
 #include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// 控制 WS2812 灯条的引脚编号
#define PIN        8

//定义控制的 LED 数量
#define NUMPIXELS 16

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

//相邻 LED 之间的延迟，单位毫秒
#define DELAYVAL 500

void setup() {
  // These lines are specifically to support the Adafruit Trinket 5V 16 MHz.
  // Any other board, you can remove this part (but no harm leaving it):
#if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
  clock_prescale_set(clock_div_1);
#endif
  // END of Trinket-specific code.

  pixels.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)
}

void loop() {
  pixels.clear(); // Set all pixel colors to 'off'

  // The first NeoPixel in a strand is #0, second is 1, all the way up
  // to the count of pixels minus one.
  for(int i=0; i<NUMPIXELS; i++) { // For each pixel...

    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(0, 150, 0));

    pixels.show();   // Send the updated pixel colors to the hardware.

    delay(DELAYVAL); // Pause before next pass through loop
  }
}
```

Output

https://youtu.be/y_3hiHsjCdA