

# Arduino 课程设计：光电法测量脉搏波

覃柏霖 17307110251

## 一、课程设计

### 1. 课题背景——脉搏波

脉搏波源自心脏的收缩与舒张：心脏心室肌肉收缩，主动脉瓣开放，血液进入主动脉，主动脉向外膨胀，内部压力增大，完成心脏射血；心脏心室肌肉舒张，主动脉瓣闭合，血管内壁压力减小。主动脉每次收缩时，血液在血管压力的作用下在血管中流动，从而在手腕处形成脉搏波动，如图 1 所示。心脏收缩出血和舒张回血时，心脏心室肌肉收缩和舒张的程度不同，所以射血和回血的速度也有差异，在脉搏的波形上体现为脉搏主波上升和下降的急缓程度。

心脏跳动节奏的快慢直接影响脉搏搏动的速率，而心脏跳动快慢变化往往与内脏器官相关联，所以通过分析脉搏搏动速率变化对研究人体内脏器官的健康状态进行分析具有重要的指导作用。健康心脏的脉搏波呈现急速上升和平缓下降的平滑脉象，如果脉搏波形上升动力不足和下降多抖动则说明心脏或其他器官可能有疾病。

脉搏波的特点是微弱、低频，波形比较复杂，容易受到多种因素的影响。

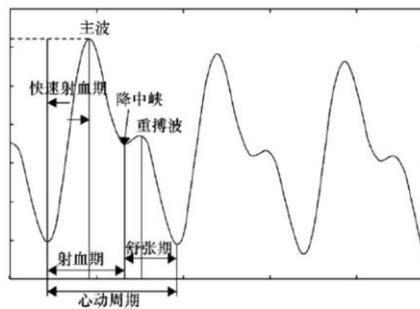


图 1. 脉搏波示意图

### 2. 光电法

肌肉和血液组织对射入的光线有一定的吸收作用，浓度越大，对光的吸收越多。机体组织受到光线照射时，光线经过皮下组织、血管等，再经过折射或反射过程穿出皮肤，光线入射前后的强度变化在一定程度上可以反映出被照射部位的组织结构和物质组分。人体上肢含有大量血管组织，尤其手指处血流量大，脉搏波动显著，所以可以用单色光照射手指的指尖，并测量透射光强或反射光强的变化。

光电传感器分为两类：透射式和反射式。当 LED 光射向皮肤，透过皮肤组织透射

或反射的光被光敏传感器接收并转换成电信号，再经过 Arduino 主板的 AD 转换得到数字信号，即脉搏信号——浓度信号——光强信号——数字信号，最后通过 Arduino 的串口展示波形，即可得到模拟的脉搏波。

3. 实验所需器材: Arduino Uno 主板, 面包板, 杜邦线若干条, 光敏传感器, LED 灯 (红色、绿色), 电阻 220  $\Omega$  若干, 黑色胶带等. 元件接线如图 2 所示.

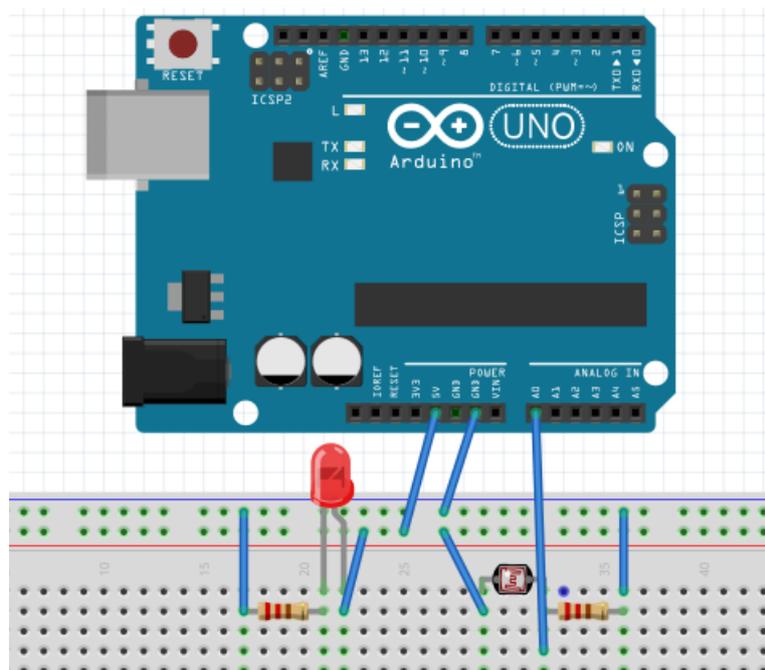


图 2. 元件接线图 (绘图: fritzing)

## 二、实验过程和结果:

### 1. 接收光信号方式的选择

经过皮肤等组织透射或反射后的光携带有人体的信息，所以接收这一光信号的方式有两种：透射式和反射式。如图 3.1 和图 3.2 所示，分别为透射式和反射式的两种接线及使用方式。

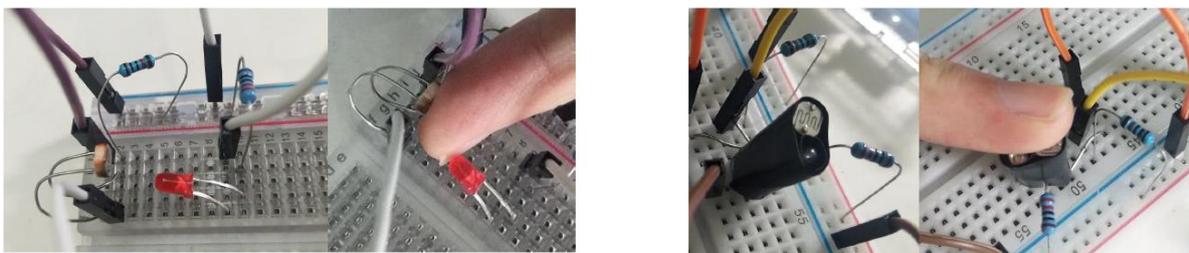


图 3. 实物接线图 (左: 透射式; 右: 反射式)

### (1) 选用透射式测量脉搏波

测量结果如图 4 所示。可以看到波形非常混乱，分析原因有：透射式的测量方式容易漏光，导致不均匀；其次容易受到环境光的干扰，造成巨大的误差；同时，透射式较难固定，手指在测量过程中由于不稳定等原因会发生抖动，影响测量结果。

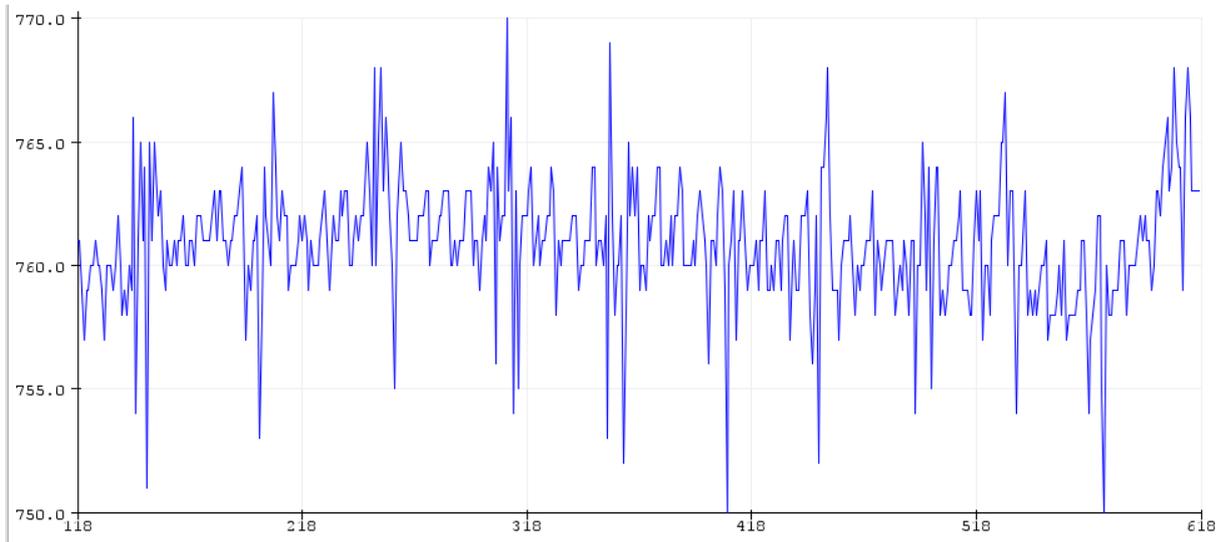


图 4. 透射式测量结果

### (2) 选用反射式测量脉搏波

反射式的测量中，为了更好的避免漏光和外界光的干扰，采用了黑色胶带将光敏电阻和 LED 绑在了一起，中间也用胶带区分开，从而保证光敏电阻接收到的光是从手指反射回来的光信号（如图 5 所示）。测量结果如图 6 所示。波形相对透射式测量法的结果更加稳定，但依然有“毛刺”出现，且在测量过程中容易产生基线的偏移。

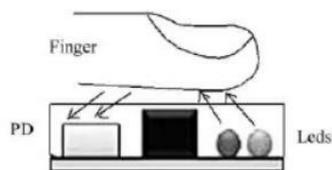


图 5. 反射式光电探测装置示意图

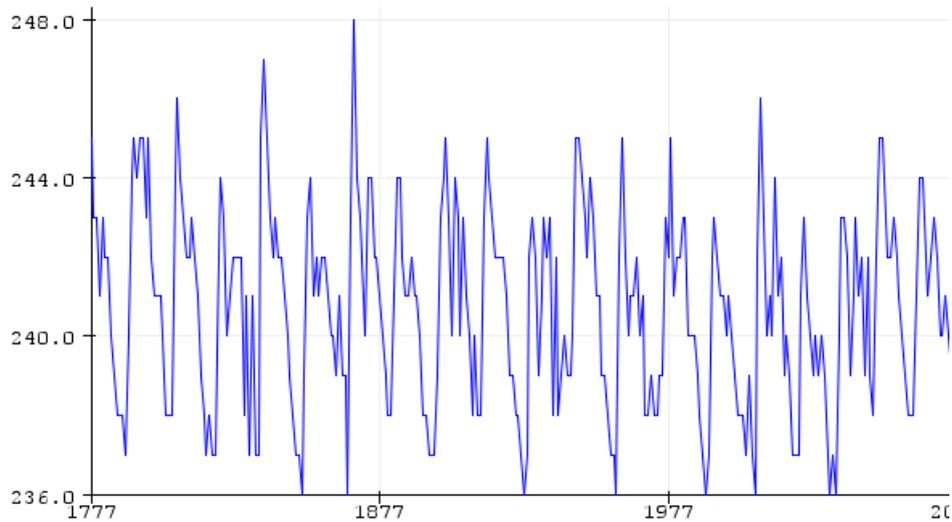


图 6. 反射式测量结果

因此，基于以上的实验，决定采用反射式采集法测量脉搏波。

## 2. LED 颜色的选择

LED 发射的光的颜色即一定的波长，应与被探测的物质相关。我们的探测对象是手指指端里的血液，当光照透过皮肤组织再反射回光敏电阻，这一过程中光强会有一些的衰减，其中组分变化较大的是血液，其他组织如肌肉、骨骼等对光的吸收并不会随脉搏变化而变化。而由于红细胞富含血红蛋白而呈红色，因此更容易吸收绿光(500-560nm)，从而导致反射光会有显著的变化，有利于信号的采集；而红光(605-700nm)则更容易穿透过皮肤组织，信号的变化比较小。换言之，如果入射光是绿光，心脏搏动导致的反射光变化较大；如果入射光是红光，心脏搏动导致的反射光变化较小。因此，考虑到信号采集的便捷性以及信噪比，绿光将会是更好的选择。

但是人体皮肤除了血液对绿光吸收得多，皮肤中的黑色素也对绿光具有较高的吸收率。如果皮肤的黑色素含量高，那么绿光将几乎被皮肤和血液全部吸收，反射光反而更弱了，这时候信号反而会更差。同理，由于绿光的穿透能力弱，采用绿光所测量到的脉搏波信号只能是皮下较浅的组织变化情况。经过市场调查发现，市场上大多数具有类似功能的手环、智能手机等都采用绿光，但医院临床使用的装置却使用红光，这表明不能简单的排除红光。

以上的分析我们可以知道，用绿光测得的信号信噪比高，即对于检测及信号处理要求较低，容易得到较稳定的信号，而且已经被广泛使用，技术比较成熟，容易开发。而红光信噪比低，检测和处理的难度大，容易受干扰，相对绿光来说技术不够成熟，没有

被各大手环手表厂商采用。而医院临床使用的装置的信号处理能力较强，开发难度相对较大，但可以有效获取真实的脉搏波信号，既避免了绿光被全部吸收的肤色问题，又能检测脉搏波信号。

出于便于开发的角度，本实验选择绿光作为入射光的光源。

### 3. 信号处理

采用 Fourier 变换的方法从时域数据得到频域数据，从而得到某个频率上存在峰值，大约在 0.6-5Hz 范围内，从而可以换算成心率。另外，也可以利用硬件（利用简单的电容、电阻等）进行滤波和放大电路，从而提高信号的信噪比。在硬件的设计上，掌握了焊接技巧，学会了电烙铁的基本操作。

## 三、结果与分析

实验中获得较好的脉搏波信号如图 7 所示。由于测量过程中手指的运动等因素，信号依然存在基线漂移等问题，另外，由于电信号的幅度较小，信号依然存在截断的情况，这是由于 Arduino 主板对信号的敏感度或分辨率有限而导致的。可以通过设计更好的放大电路加以改进。但这一问题并不影响对心率的计算。

另外，这一信号结果已经较好的除去了噪声，提高了信噪比。

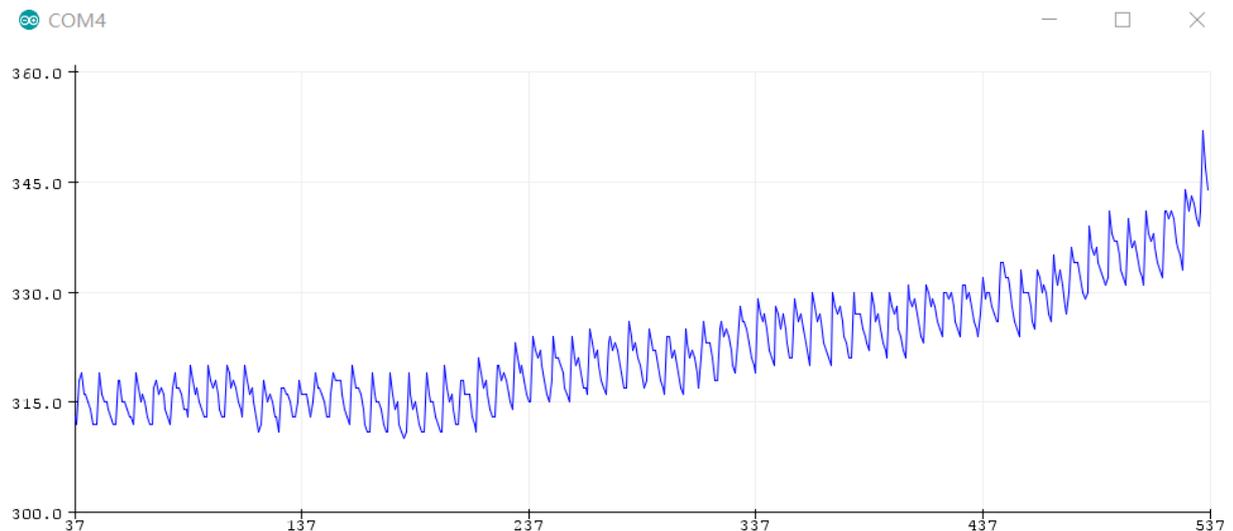
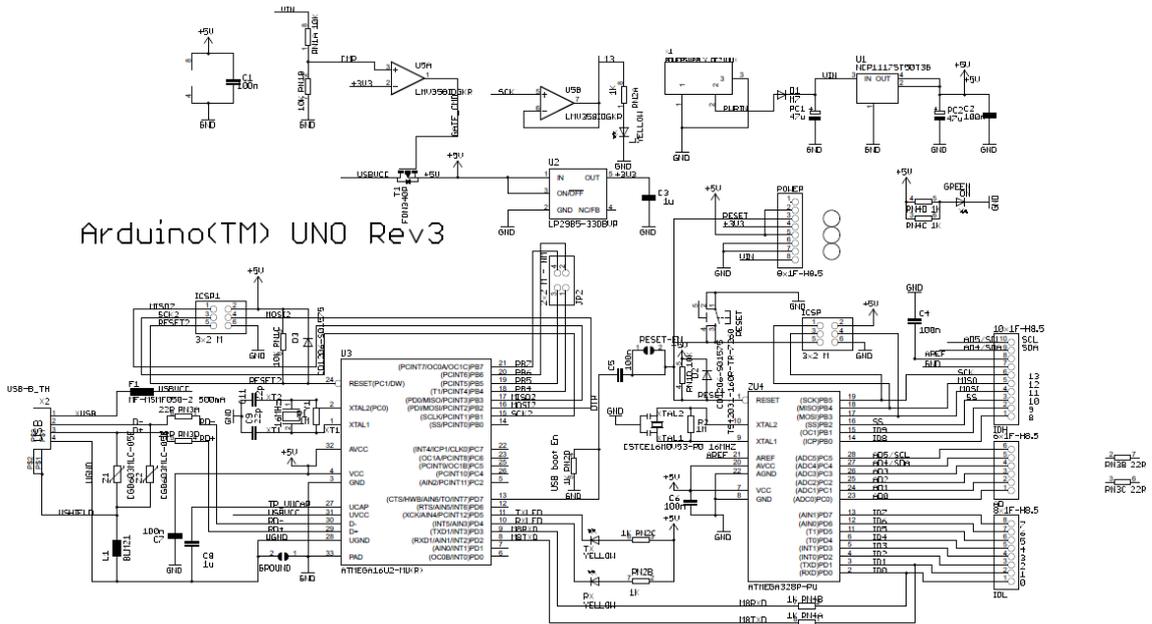


图 7. 脉搏波测量信号

# 附：其他基于 Arduino 的小项目的实验记录笔记

## 1. Arduino Uno 主板示意图



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark. Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

## 2. 作业笔记：液晶显示温度计



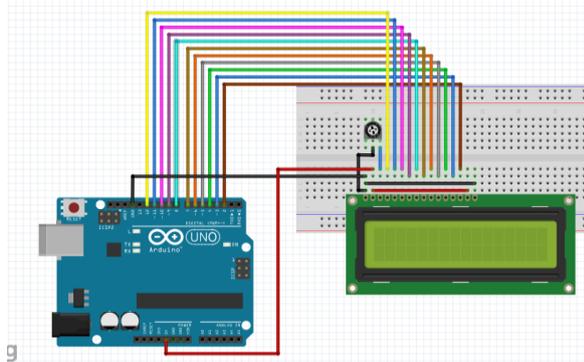
配置：ArduinoUno 开发板，面包板，5V 电源，各种接线；DHT11 温湿度传感器模块，1602LCD 液晶屏模块，电位器。

功能：温湿度传感器探测温度（Temp，单位°C）和湿度（Hum，单位%），并在液晶显示屏输出。

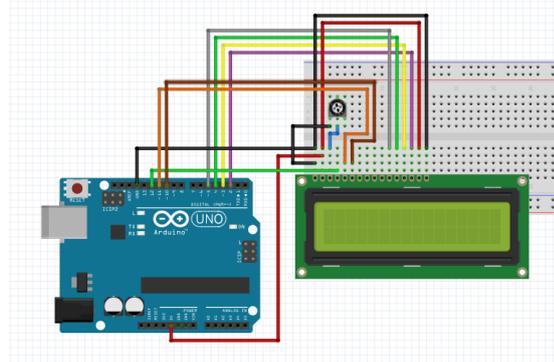
精度说明：温度 $\pm 2^{\circ}\text{C}$ ；湿度 $\pm 5\%$ 。

接线：

8 位线法（采用）



4 位线法



代码及说明：

```
#include <LiquidCrystal.h> //导入液晶库（系统自带的）
#include <DHT11.h> //导入DHT11库，自己下载，放到Arduino安装文件夹中的libraries中.注意库的文件名字（DHT11）要和<>里面的相同
dht11 DHT11_1; //把dht11这个库（也就是DHT11，要改成小写）里的东西引用，记为DHT11_1
#define DHT11PIN 2 //传感器信号脚位记为2
LiquidCrystal lcd(12,11,10,9,8,7,6,5,4,3,13); //8位线法，也是把液晶库引用成lcd，后面用lcd
void setup() {
  lcd.begin(16,2); //液晶设定：两行，每行16个字符（固定的设置）
  Serial.begin(9600); //串口输出，用来调试
}
void loop() {
  DHT11_1.read(DHT11PIN); //从传感器脚位读取传感器的数据
  // 串口输出
  Serial.print("Humidity (%): ");
  Serial.println((float)DHT11_1.humidity, 2); //浮点型数据，读取湿度，两位小数
  Serial.print("Temperature (oC): ");
  Serial.println((float)DHT11_1.temperature, 2);
  // 液晶输出
  lcd.clear(); //清屏
  lcd.print("Temp(oC): "); //在第一行第一个位置开始输出
  lcd.print(DHT11_1.temperature); //紧接着输出刚才读出来的数据，输出温度数据
  lcd.print("+2"); //温度精度
  lcd.setCursor(0,1); //把光标移到第一(0)列第二(1)行
  lcd.print("Hum(%): "); //从这个光标处开始输出内容
  lcd.print(DHT11_1.humidity); //输出湿度数据
  lcd.print("+5");
  delay(2000); //延迟2s，然后更新刷屏数据
}
```

### 3. 利用 74HC595 芯片点亮多个 LED

74HC595 芯片的作用：扩充脚位。

原理：

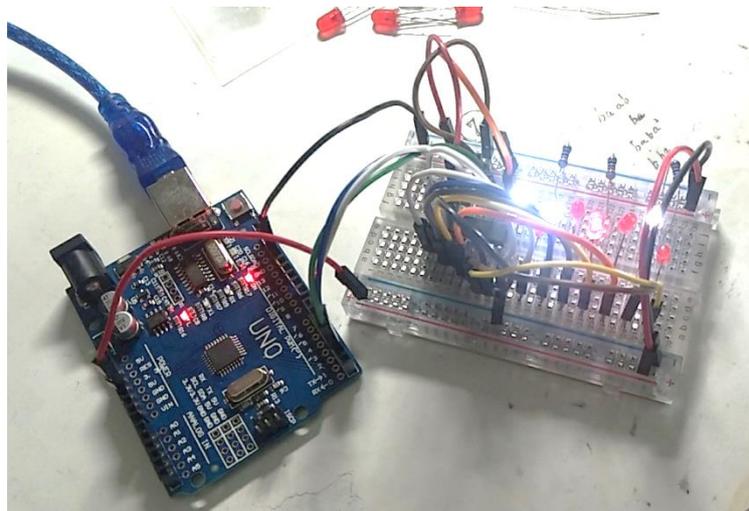
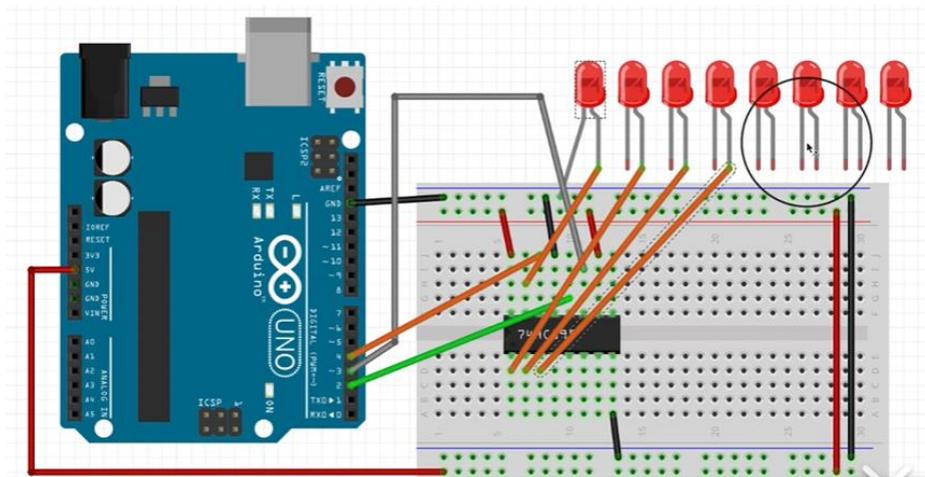
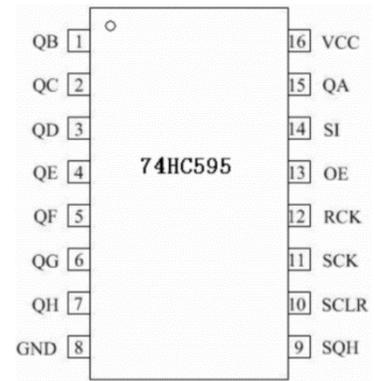
Vin 输入电压，GND 接地，0-7 是八个可以用的脚位；

OE: output enable, 接地；

MR: reset, 低电压的时候会重置，接高电压；

ST: 大平台； SH: 活塞； DS: 资料。

接线：大平台接 pin2，活塞接 pin3，资料接 pin4，0-7 号脚位接 LED 或电阻。



代码：

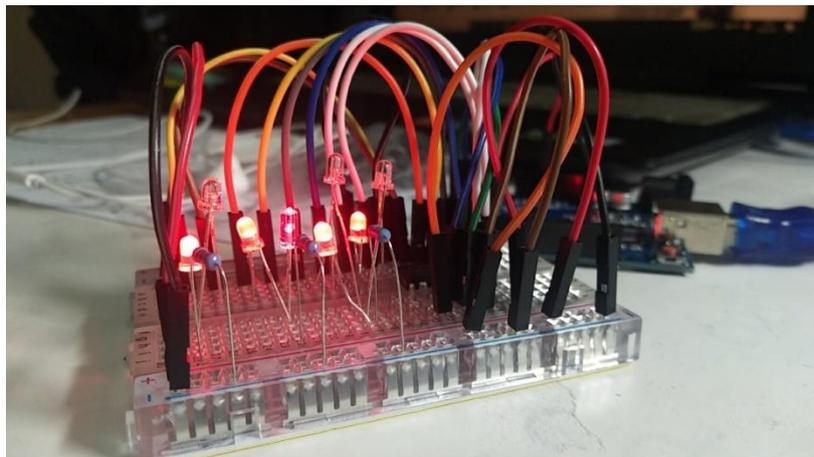
```

int big = 2; //ST: big platform
int push = 3; // SH: pusher
int datain = 4; //DS: data
int datainput[11][8]={0,1,0,1,0,1,0,1},
{1,1,0,0,0,0,0,0},
{1,1,1,0,0,0,0,0},
{1,1,1,1,0,0,0,0},
{1,1,1,1,0,0,0,0},
{1,1,1,1,1,0,0,0},
{1,1,1,1,1,1,0,0},
{1,1,1,1,1,1,1,0},
{1,1,1,1,1,1,1,1},
{0,0,0,0,0,0,0,0},
{1,1,1,1,1,1,1,1},
{0,0,0,0,0,0,0,0}};

void setup() {
  pinMode(big, OUTPUT);
  pinMode(push, OUTPUT);
  pinMode(datain, OUTPUT);
}
void loop() {
  for (int j=0;j<11;j++) {
    digitalWrite(big,LOW); //lay down the platform
    for (int i=0;i<8;i++) {
      putsth(datainput[j][i]);
    }
    digitalWrite(big,HIGH); //put up the platform
    delay(100);
  }
}
void putsth(int number){
  digitalWrite(push,LOW); //pusher left
  digitalWrite(datain,number); //data number
  digitalWrite(push,HIGH); //pusher right
}

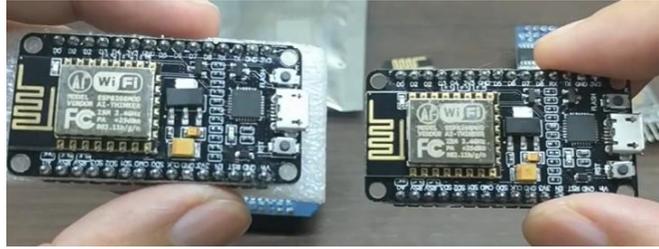
```

效果:



## 4. WiFi 模块——用手机或电脑控制 LED 开关

选材：ESP8266



安装开发板管理器：文件——首选项——附件开发板管理器

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

工具——开发板——开发板管理员， 在界面那里的文本框输入 esp8266 并安装在 IDE 选对应的开发板：NodeMCU 1.0(ESP-12E Module)，确定端口。

```
#include <ESP8266WiFi.h>
const char* ssid = "TP-LINK_29C7E4"; //name of WiFi
const char* password = "987321456"; //password of WiFi
WiFiServer server(80); //port 80

void setup() {
  Serial.begin(115200); // open the serial of the pc, rate at 115200
  delay(10);
  pinMode(2, OUTPUT); // set pin2 as output, D4
  digitalWrite(2, 0);
  //connect with wifi
  Serial.println(); //print space
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password); //using wifi to connect!!
  while (WiFi.status() != WL_CONNECTED) { //not stop printing dot until connected
    delay(500);
    Serial.print(".");
  }
  Serial.print("");
  Serial.print("WiFi connected");
  server.begin(); // open the server
  Serial.println("Server started");
  Serial.println(WiFi.localIP());
}

void loop() {
  // making sure if still connecting, otherwise jump out
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
}
```

```

return;
}
//waiting for the information in forever
Serial.println("new client");
while (!client.available()){
  delay(1);
}
//receive information successfully, store it as req
String req = client.readStringUntil('\r');
Serial.println(req); //print the information
client.flush();
//do sth for the information
int val;
if (req.indexOf("/gpio/0") != -1) //if receive /gpio/0
  val = 0;
else if (req.indexOf("/gpio/1") !=-1) //if receive /gpio/1
  val = 1;
else {
  Serial.println("invalid request");
  client.stop();
  return;
}
digitalWrite(2, val); // information to the LED
client.flush();
// ready to react to the sender, in html form, sender would receive a simple website
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html>\r\n\rGPIO is now ";
s += (val)?"high":"low";
s += "</html>\n";
//ready, send to the sender!
client.print(s);
delay(1);
Serial.println("Client disconnected");

```

效果:

