第 5 章

节奏大师之疯狂打地鼠项目设计[①]

本项目基于 Arduino 开发板设计一款具有计时功能、将节奏大师与打地鼠结合起来的游戏，玩家可以选择根据地鼠出现的顺序或者 LED 的提示，实现演奏功能。

5.1 功能及总体设计

本项目根据地鼠出现的顺序（按照音乐《两只老虎》的节奏），或者彩灯输出的顺序按下按键实现演奏音乐与打地鼠同步。在游戏里还加入了计时功能，记录玩家完成演奏一首音乐的时间并生成一条纪录（如果完成时间变短则更新纪录，否则保持之前纪录不变），以此鼓励玩家不断地进行游戏并打破之前的纪录。

要实现上述功能需将作品分成四部分进行设计，即总体输入、处理部分、传输部分和输出部分。总体输入选用了焊接在万用板上的 6 个按键，每个按键对应一种音调，同时也各自对应 1 个 LED 和网页端的地鼠洞，当对应的 LED 亮或者出现地鼠的时候，按下按键即可演奏音乐，打下地鼠；传输部分选用了 ESP8266 模块配合 Arduino 开发板实现，将 LED 输出的信息（程序生成的字符）同步传输到网页端，由暗变亮或者由亮变暗的 LED 对应地鼠出现或被打下；处理部分由本地服务器和前端构成，将 ESP8266 传送的信息进行处理；输出部分使用 6 个 LED 和网页端上显示的打地鼠界面实现。

1. 整体框架图

整体框架如图 5-1 所示。

图 5-1 整体框架图

① 本章根据陈文恺、谢岳项目设计整理而成。

2．系统流程图

系统流程如图 5-2 所示。

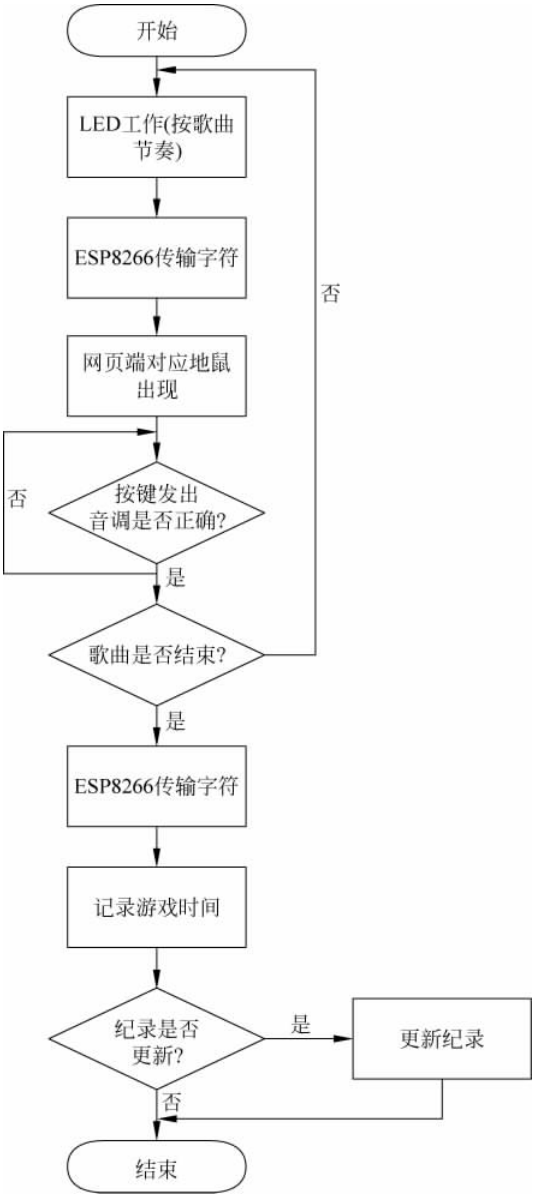

图 5-2　系统流程图

预先在代码中写入了《两只老虎》的节奏。游戏开始之后，LED 按照音乐的节奏由暗变亮。ESP8266 将此信息传至服务器和前端，对此信息进行处理，控制网页端对应地鼠出现。根据网页端地鼠出现或者 LED 变亮的信息，正确按下对应按键，让蜂鸣器发出准确的音调，LED 由亮变暗，网页端的地鼠被打下；未按或者按错按键，不会发出声音或音调错误，则 LED 和网页端的地鼠都保持不变直至玩家按下正确的按键。

3. 总电路图

总电路如图 5-3 所示，引脚连接如表 5-1 所示。

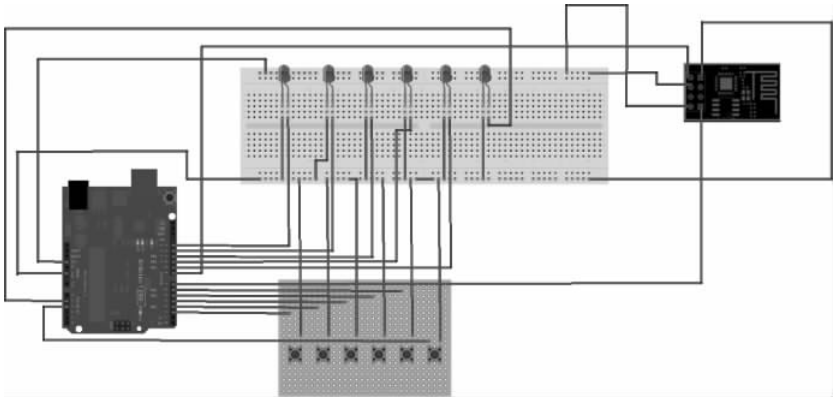

图 5-3　总电路图

表 5-1　引脚连接表

元件及引脚名		Arduino 开发板引脚
ESP8266	UTXD	8
	CH_PD	3.3V
	VCC	3.3V
	URXD	7
	GND	GND
LED	LED1	13
	LED2	12
	LED3	11
	LED4	10
	LED5	9
	LED6	A1
开关	KEY1	2
	KEY2	3
	KEY3	4
	KEY4	5
	KEY5	6
	KEY6	A2

5.2　模块介绍

本项目主要包括主程序部分(控制 LED 按照歌曲节奏输出以及设置按键、控制蜂鸣器音调)、ESP8266 模块(传输信息)、本地服务器和前端部分。下面分别给出各模块的功能介绍及相关代码。

5.2.1 主程序模块

本部分包括主程序模块的功能介绍及相关代码。

1. 功能介绍

主程序模块实现 6 个按键设置分别对应不同的音调，当按下按键的时候，蜂鸣器发出对应音调；将 6 个 LED 与《两只老虎》节奏对应，当演奏某音调时，对应 LED 由暗转亮，结束时，对应 LED 由亮转暗；将 6 个 LED 与 6 个按键一一对应，每当按下一次按键，对应 LED 由亮转暗。元件包括 6 个按键开关、6 个 LED、Arduino 开发板、蜂鸣器、万用板和导线若干，电路如图 5-4 所示。

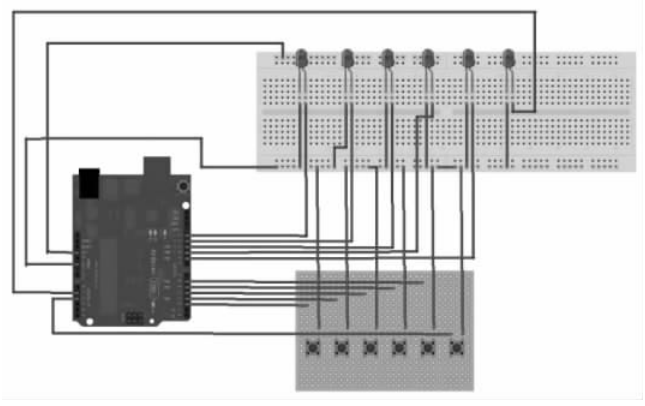

图 5-4 LED、按键开关与 Arduino 开发板连接图

2. 相关代码

```
#define LED1 13
#define LED2 12
#define LED3 11
#define LED4 10
#define LED5 9
#define LED6 A1
#define KEY1 2
#define KEY2 3
#define KEY3 4
#define KEY4 5
#define KEY5 6
#define KEY6 A2
int ledPin = A3;
int capval1,capval2,capval3,capval4,capval5,capval6,capval7,capval8;
int KEY1_NUM = 0;                          //存放变量
int KEY2_NUM = 0;
int KEY3_NUM = 0;
int KEY4_NUM = 0;
int KEY5_NUM = 0;
int KEY6_NUM = 0;
int a = 1;
```

```
int b = 0;
int c = 1;
int flag = 0;
int flagg = 0;
void setup()
{
  pinMode(LED1,OUTPUT);                    //定义 LED 为输出引脚
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
  pinMode(LED4,OUTPUT);
  pinMode(LED5,OUTPUT);
  pinMode(LED6,OUTPUT);
  pinMode(KEY1,INPUT_PULLUP);              //定义 KEY 为带上拉输入引脚
  pinMode(KEY2,INPUT_PULLUP);
  pinMode(KEY3,INPUT_PULLUP);
  pinMode(KEY4,INPUT_PULLUP);
  pinMode(KEY5,INPUT_PULLUP);
  pinMode(KEY6,INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
  }
void loop()
{
if(flag!= 1){
switch(a) {
case 1: led = '1'; a = 0;c++;digitalWrite(LED1,1);b = 1;
break;
case 2:led = '2'; a = 0;c++;digitalWrite(LED2,1);b = 2;
break;
case 3:led = '3'; a = 0;c++;digitalWrite(LED3,1);b = 3;
break;
case 4:;led = '1'; a = 0;c++;digitalWrite(LED1,1);b = 1;
break;
case 5:led = '1'; a = 0;c++; digitalWrite(LED1,1);b = 1;
break;
case 6:led = '2'; a = 0;c++;digitalWrite(LED2,1);b = 2;
break;
case 7:led = '3'; a = 0;c++;digitalWrite(LED3,1);b = 3;
break;
case 8:led = '1'; a = 0;c++;digitalWrite(LED1,1);b = 1;
break;
case 9:led = '3'; a = 0;c++;digitalWrite(LED3,1);b = 3;
break;
case 10:led = '4';updateSensorData(); a = 0;c++; digitalWrite(LED4,1);b = 4;
break;
case 11:led = '5'; a = 0;c++;digitalWrite(LED5,1);b = 5;
break;
case 12:led = '3'; a = 0;c++; digitalWrite(LED3,1);b = 3;
break;
case 13:led = '4'; a = 0;c++;digitalWrite(LED4,1);b = 4;
break;
case 14:led = '5'; a = 0;c++;digitalWrite(LED5,1);b = 5;
```

```
break;
case 15:led = '5'; a = 0;c++;digitalWrite(LED5,1);b = 5;
break;
case 16:led = '6'; a = 0;c++;digitalWrite(LED6,1);b = 6;
break;
case 17:led = '5'; a = 0;c++;digitalWrite(LED5,1);b = 5;
break;
case 18:led = '4'; a = 0;c++;digitalWrite(LED4,1);b = 4;
break;
case 19:led = '3'; a = 0;c++;digitalWrite(LED3,1);b = 3;
break;
case 20:led = '1'; a = 0;c++;digitalWrite(LED1,1);b = 1;
break;
case 21:led = '5'; a = 0;c++;digitalWrite(LED5,1);b = 5;
break;
case 22:led = '6'; a = 0;c++;digitalWrite(LED6,1);b = 6;
break;
case 23:led = '5'; a = 0;c++;digitalWrite(LED5,1);b = 5;
break;
case 24:led = '4'; a = 0;c++;digitalWrite(LED4,1);b = 4;
break;
case 25:led = '3'; a = 0;c++;digitalWrite(LED3,1);b = 3;
break;
case 26:led = '1'; a = 0;c++;digitalWrite(LED1,1);b = 1;
break;
case 27:led = '3'; a = 0;c++;digitalWrite(LED3,1);b = 3;
break;
case 28:led = '5'; a = 0;c++;digitalWrite(LED5,1);b = 5;
break;
case 29:led = '1'; a = 0;c++;digitalWrite(LED1,1);b = 1;
break;
case 30:led = '3'; a = 0;c++;digitalWrite(LED3,1);b = 3;
break;
case 31:led = '5'; a = 0;c++;digitalWrite(LED5,1);b = 5;
break;
case 32:led = '1';a = 0;c++;digitalWrite(LED1,1);b = 1;
break;
default:
break;
}
  }
if(a == 33)
{
  flag = 1;
  led = '8';
  a++;
  }
  switch(b){
  case 1: ScanKey1(); break;            //按键扫描程序,当按下时,子程序会修改 KEY_NUM 的值
  case 2: ScanKey2(); break;
  case 3: ScanKey3(); break;
```

```
  case 4: ScanKey4(); break;
  case 5: ScanKey5(); break;
  case 6: ScanKey6(); break;
  }
  if(KEY1_NUM == 1)  //是否按下,如果< span style = "font - family:Arial, Helvetica, sans -
                     //serif;"> ScanKey 函数扫描到按键就会设置 KEY_NUM 值为 1 </span >
  {
    digitalWrite(LED1,0);              //LED1 灭
    }
  if(KEY2_NUM == 1)
  {
    digitalWrite(LED2,0);              //LED2 灭
  }

  if(KEY3_NUM == 1)
  {
    digitalWrite(LED3,0);              //LED4 灭
  }

  if(KEY4_NUM == 1)
  {
    digitalWrite(LED4,0);              //LED4 灭
  }
  if(KEY5_NUM == 1)
  {
    digitalWrite(LED5,0);              //LED5 灭
  }
  if(KEY6_NUM == 1)
  {
    digitalWrite(LED6,0);              //LED6 灭
  }
}
void ScanKey1()                        //按键扫描程序
{
  KEY1_NUM = 0;                        //清空变量
  if(digitalRead(KEY1) == LOW)         //有按键按下
  {
    delay(20);                         //延时去抖动
    if(digitalRead(KEY1) == LOW)       //有按键按下
    {
      KEY1_NUM = 1;                    //变量设置为 1
      a = c;
      while(digitalRead(KEY1) == LOW){
      tone(ledPin, 262, 10);
      };                               //等待按键松手
    }
  }
}
void ScanKey2()                        //按键扫描程序
{
  KEY2_NUM = 0;                        //清空变量
```

```
  if(digitalRead(KEY2) == LOW)                    //有按键按下
  {
    delay(20);                                    //延时去抖动
    if(digitalRead(KEY2) == LOW)                  //有按键按下
    {
      KEY2_NUM = 1;                               //变量设置为 1
      a = c;
      while(digitalRead(KEY2) == LOW){
      tone(ledPin, 294, 10);
      };                                          //等待按键松手
    }
  }
}
void ScanKey3()                                   //按键扫描程序
{
  KEY3_NUM = 0;                                   //清空变量
  if(digitalRead(KEY3) == LOW)                    //有按键按下
  {
    delay(20);                                    //延时去抖动
    if(digitalRead(KEY3) == LOW)                  //有按键按下
    {
       KEY3_NUM = 1;                              //变量设置为 1
       a = c;
       while(digitalRead(KEY3) == LOW){
       tone(ledPin, 330, 10);
      };                                          //等待按键松手
    }
  }
}
void ScanKey4()                                   //按键扫描程序
{
  KEY4_NUM = 0;                                   //清空变量
  if(digitalRead(KEY4) == LOW)                    //有按键按下
  {
    delay(20);                                    //延时去抖动
    if(digitalRead(KEY4) == LOW)                  //有按键按下
    {
      KEY4_NUM = 1;                               //变量设置为 1
       a = c;
    while(digitalRead(KEY4) == LOW){
    tone(ledPin, 350, 10);
      };                                          //等待按键松手
    }
  }
}
void ScanKey5()                                   //按键扫描程序
{
  KEY5_NUM = 0;                                   //清空变量
  if(digitalRead(KEY5) == LOW)                    //有按键按下
  {
    delay(20);                                    //延时去抖动
```

```
    if(digitalRead(KEY5) == LOW)                    //有按键按下
    {
       KEY5_NUM = 1;                                //变量设置为 1
       a = c;
       while(digitalRead(KEY5) == LOW){
       tone(ledPin, 393, 10);
      };                                            //等待按键松手
    }
  }
}
void ScanKey6()                                     //按键扫描程序
{
  KEY6_NUM = 0;                                     //清空变量
  if(digitalRead(KEY6) == LOW)                      //有按键按下
  {
    delay(20);                                      //延时去抖动
    if(digitalRead(KEY6) == LOW)                    //有按键按下
    {
       KEY6_NUM = 1;                                //变量设置为 1
       a = c;
       while(digitalRead(KEY6) == LOW){
       tone(ledPin, 441, 10);
      };                                            //等待按键松手
    }
  }
}
```

5.2.2　ESP8266 模块

本部分包括 ESP8266 模块的功能介绍及相关代码。

1. 功能介绍

ESP8266 将主程序生成的字符传输到服务器控制网页的打地鼠游戏。元件包括 ESP8266 模块、Arduino 开发板和导线若干，电路如图 5-5 所示。

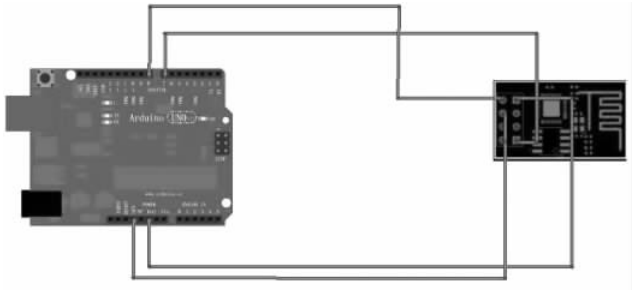

图 5-5　ESP8266 模块与 Arduino 开发板连线图

2. 相关代码

```
#include <Wire.h>                                   //调用库文件
#include "./ESP8266.h"
```

```
#define SSID "iPhone"
#define PASSWORD "12345678"
#define WLAN_SECURITY WLAN_SEC_WPA2
#define IDLE_TIMEOUT_MS 3000                          //无数据等待时间
#define HOST_NAME "172.20.10.4"                       //可改成自己的服务器地址和端口
#define HOST_PORT (8081)
#include <SoftwareSerial.h>
SoftwareSerial mySerial(8, 7);                        //定义软串口,引脚 7 为 RX,引脚 8 为 TX
ESP8266 wifi(mySerial);
//ESP8266 wifi(Serial1);                              //定义一个 ESP8266 的对象
unsigned long net_time1 = millis();                   //数据上传服务器时间
char led = '7';
String postString;                                    //用于存储发送数据的字符串
//String jsonToSend;                                  //用于存储发送的 JSON 格式参数
void setup()
{
//初始化串口波特率
Wire.begin();
Serial.begin(9600);
while(!Serial);
   //ESP8266 初始化
Serial.print("setup begin\r\n");
Serial.print("FW Version:");
Serial.println(wifi.getVersion().c_str());
if (wifi.setOprToStationSoftAP()) {
    Serial.print("to station + softap ok\r\n");
  } else {
    Serial.print("to station + softap err\r\n");
  }
  if (wifi.joinAP(SSID, PASSWORD)) {                  //加入无线网
    Serial.print("Join AP success\r\n");
    Serial.print("IP: ");
    Serial.println(wifi.getLocalIP().c_str());
  } else {
    Serial.print("Join AP failure\r\n");
  }
  if (wifi.disableMUX()) {
    Serial.print("single ok\r\n");
  } else {
    Serial.print("single err\r\n");
  }
  Serial.print("setup end\r\n");
    if (wifi.createTCP(HOST_NAME, HOST_PORT)) {
//建立 TCP 连接,如果失败,不能发送该数据
    Serial.print("create tcp ok\r\n");}
}
void loop()
{
updateSensorData();
}
```

```
void updateSensorData() {
//postString 将存储传输请求,格式很重要
  postString = "POST ";                                    //POST 发送方式,后要有空格
  postString += "/process_post?led = ";                    //接口 process
  postString += led;                                       //要发送的数据
  postString += " HTTP/1.1";                               //空格 + 传输协议
  postString += "\r\n";
  postString += "Host: ";                                  //Host: + 空格
  postString += HOST_NAME;
  postString += "\r\n";
  postString += "Content - Type: application/x - www - form - urlencoded\r\n";     //编码类型
  postString += "\r\n";                                    //不可删除
  const char * postArray = postString.c_str();             //将 str 转化为 char 数组
  Serial.println(postArray);
  wifi.send((const uint8_t * )postArray, strlen(postArray));
   //send 发送命令,参数必须是这两种格式,尤其是(const uint8_t * )
  Serial.println("send success");
}
```

5.2.3　服务器模块

本部分包括服务器模块的功能介绍及相关代码。

1. 功能介绍

服务器主要是通过 HTTP 协议接收 ESP8266 传输过来的数据,并通过 socket.io 和前端建立 websocket 连接,推送内容给前端展示。

2. 相关代码

```
var express = require('express');
var app = express();
var bodyParser = require('body - parser');
var fs = require('fs');
var util = require('util');
//创建 application/x - www - form - urlencoded 编码解析
var urlencodedParser = bodyParser.urlencoded({ extended: false })
app.use(express.static('public'));
var aj;
var server = app.listen(8081, function() {
   var host = server.address().address
   var port = server.address().port
   console.log("应用实例,访问地址为 http://%s:%s", host, port)
})
var io = require('socket.io').listen(server);
io.sockets.on('connection', function(socket) {
   console.log('User connected');
   socket.on('disconnect', function() {
       console.log('User disconnected');
   });
});
app.post('/process_post', urlencodedParser, function(req, res) {
```

```
    //输出 JSON 格式
    var response = {
        "led": req.query.led
    };
    console.log(response);
    console.log("get post/r/n");
    io.sockets.emit('message', { text: response.led });
    res.end(JSON.stringify(response));
})
```

5.2.4 前端模块

本部分包括前端模块的功能介绍及相关代码。

1. 功能介绍

前端主要是用于展示游戏页面，地鼠的出现和被敲打，并记录游戏时长、完成计时功能以及记录完成游戏的最快时间。

2. 相关代码

1）CSS 部分

```
* {
    margin: 0;
    padding: 0;
}
.first {
    margin-left: auto;
    margin-right: auto;
}
.first img {
    position: absolute;
    width: 100%;
    text-align: center;
    z-index: -1;
}
li {
    list-style: none;
}
.first ul {
    position: fixed;
    left: 43%;
    top: 37%;
}
button {
    border: 0px;
    background-color: transparent;
    background-image: none;
    font-family: 华文彩云;
    font-size: 2em;
    line-height: 2em;
    color: rgb(226, 246, 10);
```

```
}
button:focus {
    outline: none;
}
button:hover {
    font-size: 2.2em;
}
.options {
    display: none;
}
#btnback {
    display: none;
    position: absolute;
    left: 18%;
    top: 7%;
}
.dishu_bd li,
.dishu li,
.hammer li {
    position: absolute;
    display: none;
    z-index: -1;
}
.dishu li img {
    width: 135%;
}
#one_1 {
    left: 20%;
    top: 26%;
}
#two_1 {
    left: 43%;
    top: 26%;
}
#three_1 {
    left: 68%;
    top: 26%;
}
#four_1 {
    left: 15%;
    top: 73%;
}
#five_1 {
    left: 43%;
    top: 73%;
}
#six_1 {
    left: 71%;
    top: 73%;
}
#one_2 {
```

```
        left: -8%;
        top: -18%;
    }
    #two_2 {
        left: 15.5%;
        top: -18%;
    }
    #three_2 {
        left: 40%;
        top: -17.5%;
    }
    #four_2 {
        left: -12%;
        top: 30%;
    }
    #five_2 {
        left: 16%;
        top: 31%;
    }
    #six_2 {
        left: 43%;
        top: 31%;
    }
    #one_3 {
        left: -8%;
        top: -12%;
    }
    #two_3 {
        left: 15.5%;
        top: -12%;
    }
    #three_3 {
        left: 39.5%;
        top: -12%;
    }
    #four_3 {
        left: -12%;
        top: 37%;
    }
    #five_3 {
        left: 16%;
        top: 37%;
    }
    #six_3 {
        left: 43%;
        top: 37%;
    }
    .time {
        position: absolute;
        right: 10%;
        top: 8%;
```

```
    font-family: 华文彩云;
    font-size: 5em;
    line-height: 2em;
    color: white;
}
#recordd {
    position: absolute;
    right: 48%;
    top: 40%;
    font-family: 华文彩云;
    font-size: 5em;
    line-height: 2em;
    color: rgb(226, 246, 10);
    display: none;
}
```

2）HTML 部分

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>节奏大师之疯狂打地鼠</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="css/dishu.css" rel="stylesheet">
</head>
<script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.0.3/socket.io.js"></
script>
<script src="dishu.js"></script>
<body>
    <main class="first">
        <img src="photo/bgd.jpg" alt="">
        <ul class="main_menu" id="ulmenu">
            <li><button type="button" onclick="Begin();">游戏开始</button></li>
            <li><button type="button" onclick="Options();">查看记录</button></li>
        </ul>
        <button type="button" id="btnback" onclick="Back();">返回</button>
    </main>
    <main class="second">
        <ul class="dishu">
            <li id="one_1"><img src="photo/dishu.png" alt=""></li>
            <li id="two_1"><img src="photo/dishu.png" alt=""></li>
            <li id="three_1"><img src="photo/dishu.png" alt=""></li>
            <li id="four_1"><img src="photo/dishu.png" alt=""></li>
            <li id="five_1"><img src="photo/dishu.png" alt=""></li>
            <li id="six_1"><img src="photo/dishu.png" alt=""></li>
        </ul>
        <ul class="dishu_bd">
            <li id="one_2"><img src="photo/dishu_bd.png" alt=""></li>
            <li id="two_2"><img src="photo/dishu_bd.png" alt=""></li>
            <li id="three_2"><img src="photo/dishu_bd.png" alt=""></li>
```

```
            <li id="four_2"><img src="photo/dishu_bd.png" alt=""></li>
            <li id="five_2"><img src="photo/dishu_bd.png" alt=""></li>
            <li id="six_2"><img src="photo/dishu_bd.png" alt=""></li>
        </ul>
        <ul class="hammer">
            <li id="one_3"><img src="photo/hammer.png" alt=""></li>
            <li id="two_3"><img src="photo/hammer.png" alt=""></li>
            <li id="three_3"><img src="photo/hammer.png" alt=""></li>
            <li id="four_3"><img src="photo/hammer.png" alt=""></li>
            <li id="five_3"><img src="photo/hammer.png" alt=""></li>
            <li id="six_3"><img src="photo/hammer.png" alt=""></li>
        </ul>
        <p class="time" id="timer">0</p>
        <p id="recordd">暂无</p>
    </main>
</body>
</html>
```

3）JavaScript 部分

```
var aj = -1;
var led;
var jishi = 0;
var record = 500;
var socket = io.connect('http://127.0.0.1:8081');
var x;
socket.on('message', function(data) {
    led = data.text;
    console.log(led);
    var one_1 = document.getElementById('one_1');
    var two_1 = document.getElementById('two_1');
    var three_1 = document.getElementById('three_1');
    var four_1 = document.getElementById('four_1');
    var five_1 = document.getElementById('five_1');
    var six_1 = document.getElementById('six_1');
    var one_2 = document.getElementById('one_2');
    var two_2 = document.getElementById('two_2');
    var three_2 = document.getElementById('three_2');
    var four_2 = document.getElementById('four_2');
    var five_2 = document.getElementById('five_2');
    var six_2 = document.getElementById('six_2');
    var one_3 = document.getElementById('one_3');
    var two_3 = document.getElementById('two_3');
    var three_3 = document.getElementById('three_3');
    var four_3 = document.getElementById('four_3');
    var five_3 = document.getElementById('five_3');
    var six_3 = document.getElementById('six_3');
    switch (Number(led)) {
        case 1:
            switch (aj) {
                case 1:
```

```
                one_1.style.display = 'none';
                one_2.style.display = 'block';
                one_3.style.display = 'block';
        var t = setTimeout("one_2.style.display = 'none';one_3.style.display = 'none';", 300)
                break;
            case 2:
                two_1.style.display = 'none';
                two_2.style.display = 'block';
                two_3.style.display = 'block';
        var t = setTimeout("two.style.display = 'none';two_3.style.display = 'none';", 300)
                break;
            case 3:
                three_1.style.display = 'none';
                three_2.style.display = 'block';
                three_3.style.display = 'block';
         var t = setTimeout("three_2.style.display = 'none';three_3.style.display = 'none';",
300)
                break;
            case 4:
                four_1.style.display = 'none';
                four_2.style.display = 'block';
                four_3.style.display = 'block';
        var t = setTimeout("four_2.style.display = 'none';four_3.style.display = 'none';", 300)
                break;
            case 5:
                five_1.style.display = 'none';
                five_2.style.display = 'block';
                five_3.style.display = 'block';
        var t = setTimeout("five_2.style.display = 'none';five_3.style.display = 'none';", 300)
                break;
            case 6:
                six_1.style.display = 'none';
                six_2.style.display = 'block';
                six_3.style.display = 'block';
          var t = setTimeout("six_2.style.display = 'none';six_3.style.display = 'none';", 300)
                break;
            default:
                break;
        }
        one_1.style.display = 'block';
        aj = 1;
        break;
    case 2:
        switch (aj) {
            case 1:
                one_1.style.display = 'none';
                one_2.style.display = 'block';
                one_3.style.display = 'block';
        var t = setTimeout("one_2.style.display = 'none';one_3.style.display = 'none';", 300)
                break;
            case 2:
```

```
                two_1.style.display = 'none';
                two_2.style.display = 'block';
                two_3.style.display = 'block';
var t = setTimeout("two_2.style.display = 'none';two_3.style.display = 'none';", 300)
                break;
            case 3:
                three_1.style.display = 'none';
                three_2.style.display = 'block';
                three_3.style.display = 'block';
var t = setTimeout("three_2.style.display = 'none';three_3.style.display = 'none';", 300)
                break;
            case 4:
                four_1.style.display = 'none';
                four_2.style.display = 'block';
                four_3.style.display = 'block';
var t = setTimeout("four_2.style.display = 'none';four_3.style.display = 'none';", 300)
                break;
            case 5:
                five_1.style.display = 'none';
                five_2.style.display = 'block';
                five_3.style.display = 'block';
var t = setTimeout("five_2.style.display = 'none';five_3.style.display = 'none';", 300)
                break;
            case 6:
                six_1.style.display = 'none';
                six_2.style.display = 'block';
                six_3.style.display = 'block';
var t = setTimeout("six_2.style.display = 'none';six_3.style.display = 'none';", 300)
                break;
            default:
                break;
        }
        two_1.style.display = 'block';
        aj = 2;
        break;
case 3:
        switch (aj) {
            case 1:
                one_1.style.display = 'none';
                one_2.style.display = 'block';
                one_3.style.display = 'block';
var t = setTimeout("one_2.style.display = 'none';one_3.style.display = 'none';", 300)
                break;
            case 2:
                two_1.style.display = 'none';
                two_2.style.display = 'block';
                two_3.style.display = 'block';
var t = setTimeout("two_2.style.display = 'none';two_3.style.display = 'none';", 300)
                break;
            case 3:
                three_1.style.display = 'none';
```

```
            three_2.style.display = 'block';
            three_3.style.display = 'block';
var t = setTimeout("three_2.style.display = 'none';three_3.style.display = 'none';", 300)
            break;
        case 4:
            four_1.style.display = 'none';
            four_2.style.display = 'block';
            four_3.style.display = 'block';
var t = setTimeout("four_2.style.display = 'none';four_3.style.display = 'none';", 300)
            break;
        case 5:
            five_1.style.display = 'none';
            five_2.style.display = 'block';
            five_3.style.display = 'block';
var t = setTimeout("five_2.style.display = 'none';five_3.style.display = 'none';", 300)
            break;
        case 6:
            six_1.style.display = 'none';
            six_2.style.display = 'block';
            six_3.style.display = 'block';
var t = setTimeout("six_2.style.display = 'none';six_3.style.display = 'none';", 300)
            break;
        default:
            break;
    }
    three_1.style.display = 'block';
    aj = 3;
    break;
case 4:
    switch (aj) {
        case 1:
            one_1.style.display = 'none';
            one_2.style.display = 'block';
            one_3.style.display = 'block';
var t = setTimeout("one_2.style.display = 'none';one_3.style.display = 'none';", 300)
            break;
        case 2:
            two_1.style.display = 'none';
            two_2.style.display = 'block';
            two_3.style.display = 'block';
var t = setTimeout("two_2.style.display = 'none';two_3.style.display = 'none';", 300)
            break;
        case 3:
            three_1.style.display = 'none';
            three_2.style.display = 'block';
            three_3.style.display = 'block';
var t = setTimeout("three_2.style.display = 'none';three_3.style.display = 'none';", 300)
            break;
        case 4:
            four_1.style.display = 'none';
            four_2.style.display = 'block';
```

```
                four_3.style.display = 'block';
var t = setTimeout("four_2.style.display = 'none';four_3.style.display = 'none';", 300)
                break;
            case 5:
                five_1.style.display = 'none';
                five_2.style.display = 'block';
                five_3.style.display = 'block';
var t = setTimeout("five_2.style.display = 'none';five_3.style.display = 'none';", 300)
                break;
            case 6:
                six_1.style.display = 'none';
                six_2.style.display = 'block';
                six_3.style.display = 'block';
var t = setTimeout("six_2.style.display = 'none';six_3.style.display = 'none';", 300)
                break;
            default:
                break;
        }
        four_1.style.display = 'block';
        aj = 4;
        break;
    case 5:
        switch (aj) {
            case 1:
                one_1.style.display = 'none';
                one_2.style.display = 'block';
                one_3.style.display = 'block';
var t = setTimeout("one_2.style.display = 'none';one_3.style.display = 'none';", 300)
                break;
            case 2:
                two_1.style.display = 'none';
                two_2.style.display = 'block';
                two_3.style.display = 'block';
var t = setTimeout("two_2.style.display = 'none';two_3.style.display = 'none';", 300)
                break;
            case 3:
                three_1.style.display = 'none';
                three_2.style.display = 'block';
                three_3.style.display = 'block';
var t = setTimeout("three_2.style.display = 'none';three_3.style.display = 'none';", 300)
                break;
            case 4:
                four_1.style.display = 'none';
                four_2.style.display = 'block';
                four_3.style.display = 'block';
var t = setTimeout("four_2.style.display = 'none';four_3.style.display = 'none';", 300)
                break;
            case 5:
                five_1.style.display = 'none';
                five_2.style.display = 'block';
                five_3.style.display = 'block';
```

```
  var t = setTimeout("five_2.style.display = 'none';five_3.style.display = 'none';", 300)
                break;
            case 6:
                six_1.style.display = 'none';
                six_2.style.display = 'block';
                six_3.style.display = 'block';
  var t = setTimeout("six_2.style.display = 'none';six_3.style.display = 'none';", 300)
                break;
            default:
                break;
        }
        five_1.style.display = 'block';
        aj = 5;
        break;
    case 6:
        switch (aj) {
            case 1:
                one_1.style.display = 'none';
                one_2.style.display = 'block';
                one_3.style.display = 'block';
var t = setTimeout("one_2.style.display = 'none';one_3.style.display = 'none';", 300)
                break;
            case 2:
                two_1.style.display = 'none';
                two_2.style.display = 'block';
                two_3.style.display = 'block';
var t = setTimeout("two_2.style.display = 'none';two_3.style.display = 'none';", 300)
                break;
            case 3:
                three_1.style.display = 'none';
                three_2.style.display = 'block';
                three_3.style.display = 'block';
var t = setTimeout("three_2.style.display = 'none';three_3.style.display = 'none';", 300)
                break;
            case 4:
                four_1.style.display = 'none';
                four_2.style.display = 'block';
                four_3.style.display = 'block';
var t = setTimeout("four_2.style.display = 'none';four_3.style.display = 'none';", 300)
                break;
            case 5:
                five_1.style.display = 'none';
                five_2.style.display = 'block';
                five_3.style.display = 'block';
var t = setTimeout("five_2.style.display = 'none';five_3.style.display = 'none';", 300)
                break;
            case 6:
                six_1.style.display = 'none';
                six_2.style.display = 'block';
                six_3.style.display = 'block';
var t = setTimeout("six_2.style.display = 'none';six_3.style.display = 'none';", 300)
```

```
                break;
            default:
                break;
        }
        six_1.style.display = 'block';
        aj = 6;
        break;
    case 7:
        x = setInterval("clk()", 1000);
        break;
    case 8:
        window.clearInterval(x);
        if (record >= jishi) {
            record = jishi;
        }
        jishi = 0;
        switch (aj) {
            case 1:
                one_1.style.display = 'none';
                one_2.style.display = 'block';
                one_3.style.display = 'block';
    var t = setTimeout("one_2.style.display = 'none';one_3.style.display = 'none';", 300)
                break;
            case 2:
                two_1.style.display = 'none';
                two_2.style.display = 'block';
                two_3.style.display = 'block';
    var t = setTimeout("two_2.style.display = 'none';two_3.style.display = 'none';", 300)
                break;
            case 3:
                three_1.style.display = 'none';
                three_2.style.display = 'block';
                three_3.style.display = 'block';
    var t = setTimeout("three_2.style.display = 'none';three_3.style.display = 'none';", 300)
                break;
            case 4:
                four_1.style.display = 'none';
                four_2.style.display = 'block';
                four_3.style.display = 'block';
    var t = setTimeout("four_2.style.display = 'none';four_3.style.display = 'none';", 300)
                break;
            case 5:
                five_1.style.display = 'none';
                five_2.style.display = 'block';
                five_3.style.display = 'block';
    var t = setTimeout("five_2.style.display = 'none';five_3.style.display = 'none';", 300)
                break;
            case 6:
                six_1.style.display = 'none';
                six_2.style.display = 'block';
                six_3.style.display = 'block';
```

```
            var t = setTimeout("six_2.style.display = 'none';six_3.style.display = 'none';", 300)
                    break;
                default:
                    break;
            }
            aj = 8;
            break;
    }
})
function Begin() {
    var ullist = document.getElementById('ulmenu');
    var btnback = document.getElementById('btnback');
    var btnback = document.getElementById('btnback');
    ullist.style.display = 'none';
    btnback.style.display = 'block';
    console.log('ok');
}
function Options() {
    var btnback = document.getElementById('btnback');
    var ullist = document.getElementById('ulmenu');
    var recordd = document.getElementById('recordd');
    ullist.style.display = 'none';
    btnback.style.display = 'block';
    recordd.innerHTML = record;
    recordd.style.display = 'block';
    console.log('ok');
}
function Back() {
    var btnback = document.getElementById('btnback');
    var ullist = document.getElementById('ulmenu');
    var recordd = document.getElementById('recordd');
    ullist.style.display = 'block';
    btnback.style.display = 'none';
    recordd.style.display = 'none';
}
function clk() {
    var timer = document.getElementById('timer');
    jishi++;
    timer.innerHTML = jishi;
}
```

5.3　产品展示

整体外观如图 5-6 所示，左边计算机上显示的是打地鼠游戏界面端，右边是 Arduino 开发板及其连接的 LED、万用板及其连接的按键开关和 ESP8266 模块。网页端游戏开始界面如图 5-7 所示，游戏进行界面如图 5-8 所示，记录完成游戏时间界面如图 5-9 所示。

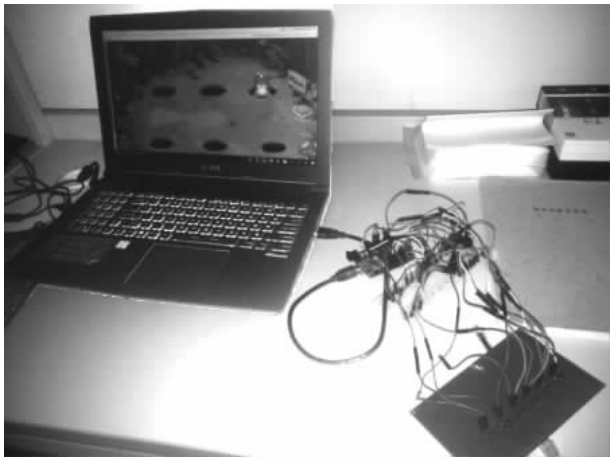

图 5-6　整体外观图

图 5-7　游戏开始界面

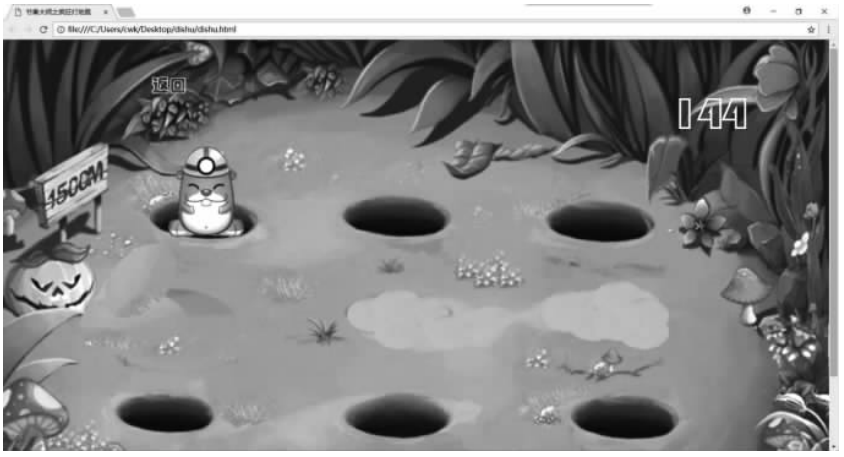

图 5-8　游戏进行界面

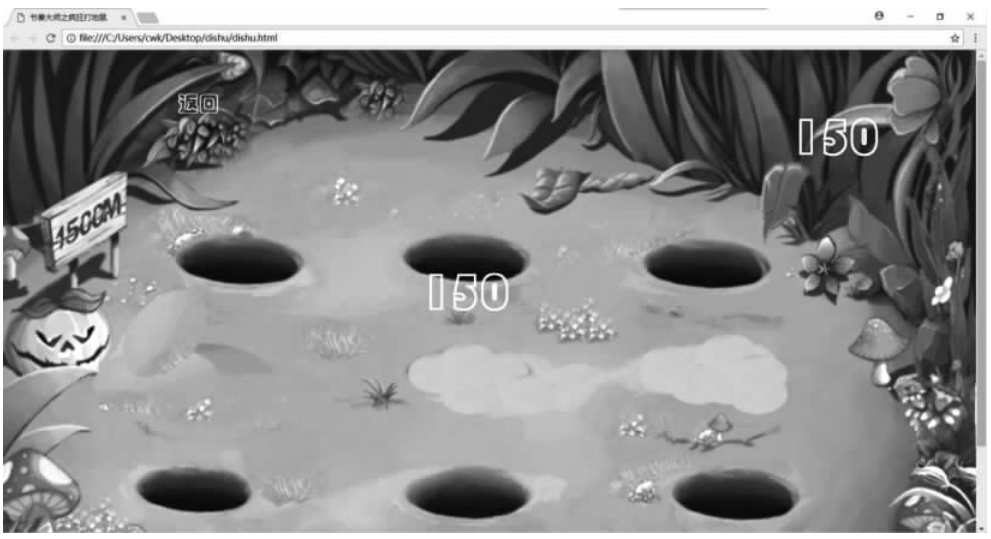

图 5-9　记录完成游戏时间界面

5.4　元件清单

完成本项目所用到的元件及数量如表 5-2 所示。

表 5-2　元件清单

元件/测试仪表	数量
Arduino 开发板	1 个
ESP8266 模块	1 个
导线	若干
LED 彩灯	6 个
按键开关	6 个
面包板	1 个
万用板	1 个