

高雄市高英高級工商職業學校

Kao Ying Industrial Commercial Vocational High School

專題製作報告



作品名稱:打地鼠機

組長:劉軍威

組員:林子勛、邱子宸、吳承澤

指導老師:蔡忠憲

中華民國 108 年 5 月

目錄

壹、摘要.....	1
貳、研究動機與目的.....	2
一、研究動機.....	2
二、研究目的.....	2
參、理論探討.....	3
一、感測器.....	3
肆、研究方法(過程).....	7
一、流程圖.....	7
二、硬體架構圖.....	7
三、程式碼.....	8
四、製作過程.....	15
伍、研究結果.....	17
一、電子打地鼠優缺點.....	17
陸、結論.....	17
一、討論.....	17
二、結論.....	18
三、心得.....	18
柒、參考資料及其他.....	18
一、網頁連結.....	19
二、書籍參考.....	19

壹、摘要

我們是使用了數位邏輯實習以及電子學實習上課所學的，再結合課本上所學的基本電路知識，來完成我們理想中的打地鼠機。然後也有使用到 Arduino 和 LCD 模組和紅外線循跡感測器模組來讓這個打地鼠機能夠有更多樣的變化，不過我們沒有使用到『地鼠』來製作這項專題，而是以亮燈來代替地鼠出來再下去的方式。我們的打地鼠機外型是使用了木盒來製作，然後在上面分別鑽了能 LCD 組裝上去的洞和 6 個可以塞紅外線循跡感測器模組跟 6 個 LED 的洞來完成整個打地鼠機的外型。

LCD 模組可以用來顯示打一個燈會有 X 分數和時間倒數，最後來看玩家總共按了幾個燈來計算分數，而 LED 是使用 Arduino 裡的隨機變數的函式來控制 LED 能隨機亮，然後能讓玩家按下去 LED 燈會滅掉的是使用了紅外線循跡感測器模組。我們想要製作的並不是什麼特別費功夫的打地鼠機，而是能讓大家玩了都會想起以前的打地鼠機的回憶。



圖 1. 兒時打地鼠機

貳、研究動機與目的

一、研究動機

我們這次要做打地鼠機是因為想到了以前在玩的掌上型打地鼠機，雖然很多遊戲機都有打地鼠這種小遊戲，但是那都是別人做好的東西，所以我們想要做屬於自己的打地鼠機來玩，這樣才有意思。

我們製作的這個打地鼠機玩具，是為了能讓大家想起以前玩打地鼠機的快樂回憶，所以我們並沒有使用了太複雜的工程來製作，反而是利用一些小小的電子零件來完成這個打地鼠機，我們要的目的就是一個可以玩的，又可以回味童年的小玩具，於是我們就想到了這個打地鼠機，這次的打地鼠機也只僅僅用了六個感測器當主要元件，然後蜂鳴器當作我們的音效，LED 燈當地鼠，省去了做那隻地鼠的經費，可是我們的木箱做的太大了，這是我們研究上的缺陷，如果木箱可以在小一點，可以省去更多經費，然後箱子也太厚了，鑽洞的時候費了點功夫，光是鑽那個木箱的洞，就花了快兩個禮拜，但是最後我們還是不負眾望把它做出來了。

二、研究目的

- (一)使用 Arduino 板來和我們的打地鼠機做結合。
- (二)利用紅外線循跡感測器來當作地鼠來打分數。
- (三)讓原本一台很大的打地鼠機，變得可以隨身攜帶，只需要一顆行動充電就可遊玩。
- (四)讓我們學習到了 LCD 模組的功用。例如:能夠顯示各種文字。

參、理論探討

一、感測器

(一)LCD1602：

是一種工業字符型液晶，能夠同時顯示 16x02 即 32 個字符。LCD1602 液晶顯示的原理是利用液晶的物理特性，通過電壓對其顯示區域進行控制，即可以顯示出圖形。

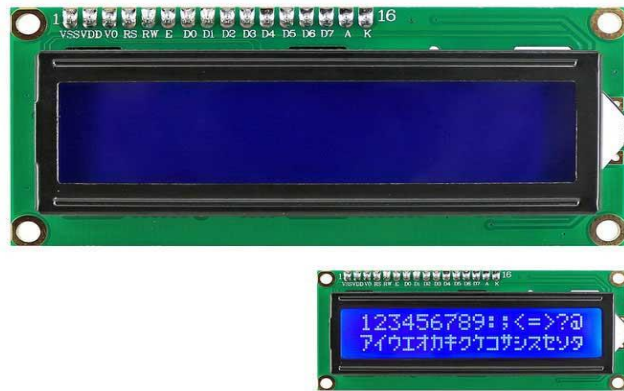


圖 2. LCD1602

(二)無源蜂鳴器：

無源蜂鳴器利用電磁感應現象，為音圈接入交變電流後形成的電磁鐵與永磁鐵相吸或相斥而推動振膜發聲，接入直流電只能持續推動振膜而無法產生聲音，只能在接通或斷開時產生聲音。

無源蜂鳴器的工作原理與揚聲器相同。在使用方波信號源驅動的應反向並聯一個二極管，防止突然斷電時產生的高壓反向電動勢擊穿其他元件以及使用壽命縮短。有源蜂鳴器往往比無源的貴，就是因為裡面多個震盪電路，只需接入額定電壓的直流電即可發出指定頻率的聲音，頻率由內部振盪電路決定，無法改變。無源蜂鳴器的優點是：

1. 製作成本低。
2. 聲音頻率範圍寬，可高分貝的發出某些頻率的超聲波以及可以做出“多來米發索拉西”的效果。
3. 在一些特例中，可以和 LED 復用一個控制口。



圖 3. 無源蜂鳴器

(三)紅外線尋跡感測器：

在紅外線探測器中，熱電元件檢測人體的存在或移動，並把熱電元件的輸出信號轉換成電壓信號。然後，對電壓信號進行波形分析。於是，只有當通過波形分析檢測到由人體產生的波形時，才輸出檢測信號。例如，在兩個不同的頻率範圍內放大電壓信號，且將被放大的信號用於鑑別由人體引起的信號。於是，誤將諸如熱電元件的爆米花噪聲一類噪聲當作為由人體所產生而在準備加以檢測乃得以防止。該紅外線探測器包括紅外線發射器、接收器、以及信號處理器，信號處理器的信號輸出端經紅外線發射電路與紅外線發射器連接；信號輸入端經紅外線接收電路與紅外線接收器連接，其反饋信號輸出端與外圍控制電路連接。

本技術採用微型單片機作為信號處理器產生編碼信號，驅動紅外線發射器發出帶有編碼信號的紅外線信號，並實時檢測經過放大電路處理後的反射信號，其編碼信號能夠保證多個相同型號的傳感器同時同地工作而不相互干擾。而且工作頻率一致、可靠性高、功耗小。



圖 4. 紅外線尋跡感測器

(四)LED(發光二極體):

是一種能發光的半導體電子元件，透過三價與五價元素所組成的複合光源。此種電子元件早在 1962 年出現，早期只能夠發出低光度的紅光，被惠普買下專利後當作指示燈利用。及後發展出其他單色光的版本，時至今日，能夠發出的光已經遍及可見光、紅外線及紫外線，光度亦提高到相當高的程度。用途由初時的指示燈及顯示板等；隨著白光發光二極體的出現，近年逐漸發展至被普遍用作照明用途。

發光二極體只能夠往一個方向導通（通電），叫作順向偏壓，當電流流過時，電子與電洞在其內複合而發出單色光，這叫電致發光效應，而光線的波長、顏色跟其所採用的半導體物料種類與故意摻入的元素雜質有關。具有效率高、壽命長、不易破損、反應速度快、可靠性高等傳統光源不及的優點。白光 LED 的發光效率近年有所進步；每千流明成本，也因為大量的資金投入使價格下降，但成本仍遠高於其他的傳統照明雖然如此，近年仍然越來越多被用在照明用途上。



圖 5. LED(發光二極體)

(五)Arduino 板:

Arduino 是一家製作開源硬體和軟體的公司，同時兼有專案和用戶社群，該公司負責設計和製造單板微控制器和微控制器套件，用於構建數位裝置和互動式物件，以便在物理和數位世界中感知和控制物件。該專案的產品是按照 GNU 寬通用公共許可證 (LGPL) 或 GNU 通用公共許可證 (GPL) 許可的開源硬體和軟體分發的，Arduino 允許任何人製造 Arduino 板和軟體分發 Arduino 板可以以預裝的形式商業銷售，也可以作為 DIY 套件購買。Arduino 電路板設計使用各種微處理器和控制器。這些電路板配有一組數字和類比 I/O 引腳，可以連接各種擴充板或麵包板 (封鎖板) 和其他電路。這些電路板具有串列埠包括某些型號上的通用串列匯流排 (USB)，也用於從個人電腦載入程式。微控制器通常使用 C/C++ 程式語言。除了使用傳統的編譯工具鏈之外，Arduino 專案還提供了一個基於 Processing 語言專案的整合式開發環境 Arduino 專案始於 2003 年，作為義大利伊夫雷亞地區伊夫雷亞互動設計研究所的學生專案，目的是為新手和專業人員提供一種低成本且簡單的方法，以建立使用傳感器與環境相互作用的裝置執行器。適用於初學者愛好者的此類裝置的常見範例包括簡單機器人恆溫器和運動檢測器。

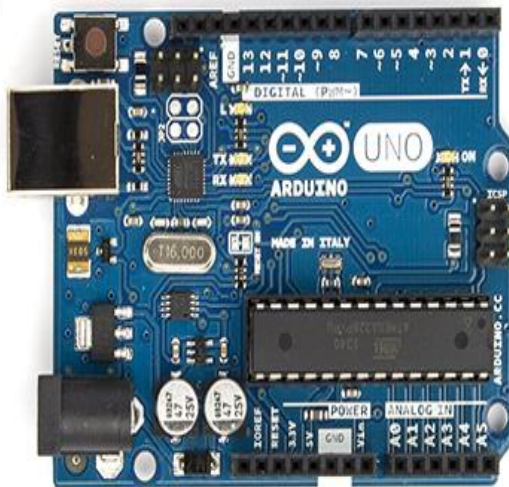


圖 6. Arduino 板

肆、研究方法(過程)

一、流程圖

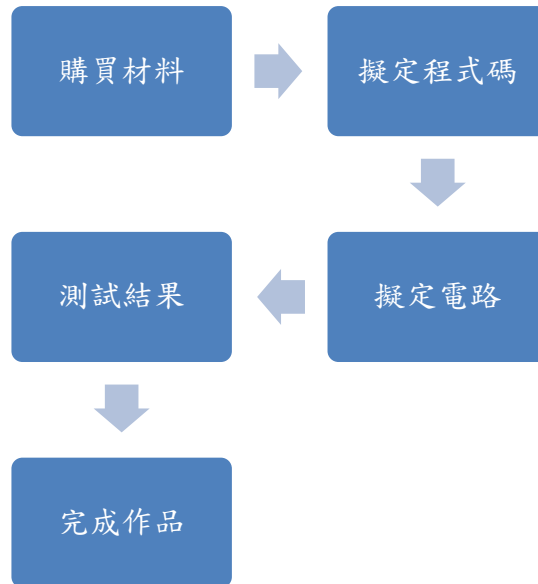


圖 7. 打地鼠機流程圖

二、硬體架構圖

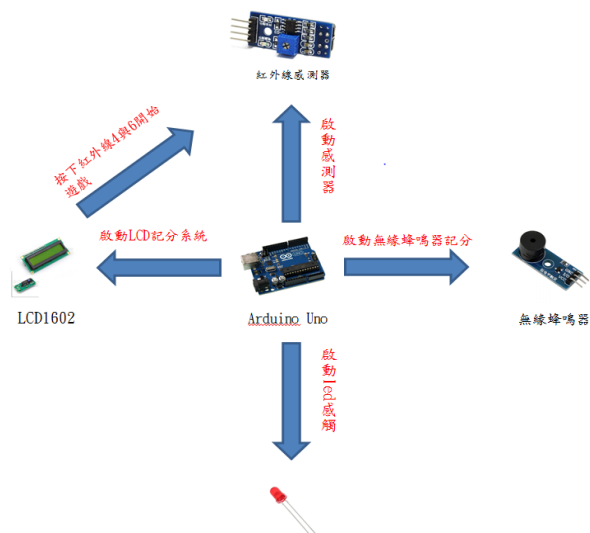


圖 8. 硬體架構圖

三、程式碼

```
#include <Wire.h> //設定 LCD
#include <motoLiquidCrystal_I2C.h> //設定 LCD
LiquidCrystal_I2C mylcd(0x3F, 16, 2); //設定 LCD
byte ir1; //宣告變數
byte ir2; //宣告變數
byte ir3; //宣告變數
byte ir4; //宣告變數
byte ir5; //宣告變數
byte ir6; //宣告變數
byte ircheck; //宣告變數
byte randled; //宣告變數
byte checkpoint; //宣告變數
int score; //宣告變數
long starttime; //宣告變數
int finaltime; //宣告變數
int gametime; //宣告變數
void music() {
    tone(A0, 523, 500); //設定 A0 腳位發出 523 赫茲持續 0.5 秒
}
void music2() {
    tone(A0, 1046, 500); //設定 A0 腳位發出 1046 赫茲持續 0.5 秒
}
void setup()
{
    mylcd.init(); //顯示字體
    mylcd.backlight(); //顯示 LEF 背光
    pinMode(11, OUTPUT); //將腳位 11 設定為輸出模式
    pinMode(13, OUTPUT); //將腳位 13 設定為輸出模式
    pinMode(5, INPUT); //將腳位 5 設定為輸入模式
    pinMode(7, INPUT); //將腳位 7 設定為輸入模式
    pinMode(8, OUTPUT); //將腳位 8 設定為輸出模式
    pinMode(9, OUTPUT); //將腳位 9 設定為輸出模式
    pinMode(10, OUTPUT); //將腳位 10 設定為輸出模式
    pinMode(12, OUTPUT); //將腳位 12 設定為輸出模式
    randomSeed(analogRead(3)); //從亂數種子碼
    pinMode(2, INPUT); //將腳位 2 設定為輸入模式
```

```

pinMode(3, INPUT); //將腳位 3 設定為輸入模式
pinMode(4, INPUT); //將腳位 4 設定為輸入模式
pinMode(6, INPUT); //將腳位 6 設定為輸入模式
  ir1 = 0; //宣告變數
  ir2 = 0; //宣告變數
  ir3 = 0; //宣告變數
  ir4 = 0; //宣告變數
  ir5 = 0; //宣告變數
  ir6 = 0; //宣告變數
  ircheck = 0; //宣告變數
  randled = 0; //宣告變數
  checkpoint = 0; //宣告變數
  score = 0; //宣告變數
  starttime = 0; //宣告變數
  finaltime = 0; //宣告變數
  gametime = 30; //宣告變數
  pinMode(A0, OUTPUT); //將腳位 A0 設定為輸出模式
}
void loop()
{
  if (checkpoint == 0) { //當 checkpoint = 0 為開始的部分
    mylcd.setCursor(0,0); //顯示位置
    mylcd.print("Touch 4 and 6 "); //顯示 Touch 4 and 6
    mylcd.setCursor(0,1); //顯示位置
    mylcd.print("Start Game...."); //顯示開始遊戲
    digitalWrite(11,HIGH); //將腳位 11 設定輸出高電位
    digitalWrite(13,HIGH); //將腳位 13 設定輸出高電位
    if (digitalRead(5) == 1 && digitalRead(7) == 1) { //感應第四
個感應器和第六個感應器
      digitalWrite(8, HIGH); //將腳位 8 設定輸出高電位
      digitalWrite(9, HIGH); //將腳位 9 設定輸出高電位
      digitalWrite(10, HIGH); //將腳位 10 設定輸出高電位
      digitalWrite(11, HIGH); //將腳位 11 設定輸出高電位
      digitalWrite(12, HIGH); //將腳位 12 設定輸出高電位
      digitalWrite(13, HIGH); //將腳位 13 設定輸出高電位
      delay(1000); //延遲 1 秒
      digitalWrite(8, LOW); //將腳位 8 設定輸出低電位
      digitalWrite(9, LOW); //將腳位 9 設定輸出低電位

```

```

digitalWrite(10,LOW); //將腳位 10 設定輸出低電位
digitalWrite(11,LOW); //將腳位 11 設定輸出低電位
digitalWrite(12,LOW); //將腳位 12 設定輸出低電位
digitalWrite(13,LOW); //將腳位 13 設定輸出低電位
delay(1000); //延遲 1 秒
checkpoint = 1; //接下來就要執行主要程式所以將 checkpoint
設為 1
    randed = random(1,7); //隨機亮燈
    starttime = millis(); //開始倒數
    score = 0; //宣告變數
    mylcd.clear(); //清除 LCD 介面
}
}
if (checkpoint == 1) { //當 checkpoint=1 時, 為進入主程式部分
    finaltime = gametime - (millis() - starttime) / 1000; //計算
倒數時間要改為 millis(), 取系統時間
    if (finaltime == 9) { //倒數數字從兩位數變到一位數時先將 LCD
顯示全部清除, 免得有一個位數的殘留
        mylcd.clear(); //清除 LCD 介面
    }
    if (randed == 1) { //從亂數中來決定要亮一顆燈
        digitalWrite(8,HIGH); //將腳位 8 設定輸出高電位
    }
    if (randed == 2) { //從亂數中來決定要亮一顆燈
        digitalWrite(9,HIGH); //將腳位 9 設定輸出高電位
    }
    if (randed == 3) { //從亂數中來決定要亮一顆燈
        digitalWrite(10,HIGH); //將腳位 10 設定輸出高電位
    }
    if (randed == 4) { //從亂數中來決定要亮一顆燈
        digitalWrite(11,HIGH); //將腳位 11 設定輸出高電位
    }
    if (randed == 5) { //從亂數中來決定要亮一顆燈
        digitalWrite(12,HIGH); //將腳位 12 設定輸出高電位
    }
    if (randed == 6) { //從亂數中來決定要亮一顆燈
        digitalWrite(13,HIGH); //將腳位 13 設定輸出高電位
    }
}
}

```

```

    ir1 = digitalRead(2); //取得腳位 2 的值感器沒有被蓋遮的值為 0
    被遮蓋時的值為 1
    ir2 = digitalRead(3); //取得腳位 3 的值感器沒有被蓋遮的值為 0
    被遮蓋時的值為 1
    ir3 = digitalRead(4); //取得腳位 4 的值感器沒有被蓋遮的值為 0
    被遮蓋時的值為 1
    ir4 = digitalRead(5); //取得腳位 5 的值感器沒有被蓋遮的值為 0
    被遮蓋時的值為 1
    ir5 = digitalRead(6); //取得腳位 6 的值感器沒有被蓋遮的值為 0
    被遮蓋時的值為 1
    ir6 = digitalRead(7); //取得腳位 7 的值感器沒有被蓋遮的值為 0
    被遮蓋時的值為 1
    if (ir1 == 1) { //如果亮燈在編號 1 的 LED 燈而且也遮蓋了編號 1
    的感測器就得分且發出聲音
        ircheck = 1; //如果亮燈在編號 1 的 LED 燈而且也遮蓋了編號 1
    的感測器就得分且發出聲音
        if (randled == 1) { //如果亮燈在編號 1 的 LED 燈而且也遮蓋了
    編號 1 的感測器就得分且發出聲音
            music(); //如果亮燈在編號 1 的 LED 燈而且也遮蓋了編號 1 的
    感測器就得分且發出聲音
            score = score + 1; //如果亮燈在編號 1 的 LED 燈而且也遮蓋
    了編號 1 的感測器就得分且發出聲音
        }
    }
    if (ir2 == 1) { //如果亮燈在編號 2 的 LED 燈而且也遮蓋了編號 2
    的感測器就得分且發出聲音
        ircheck = 2; //如果亮燈在編號 2 的 LED 燈而且也遮蓋了編號 2
    的感測器就得分且發出聲音
        if (randled == 2) { //如果亮燈在編號 2 的 LED 燈而且也遮蓋了
    編號 2 的感測器就得分且發出聲音
            music(); //如果亮燈在編號 2 的 LED 燈而且也遮蓋了編號 2 的
    感測器就得分且發出聲音
            score = score + 1; //如果亮燈在編號 2 的 LED 燈而且也遮蓋
    了編號 2 的感測器就得分且發出聲音
        }
    }
    if (ir3 == 1) { //如果亮燈在編號 3 的 LED 燈而且也遮蓋了編號 3
    的感測器就得分且發出聲音

```

```

        ircheck = 3; //如果亮燈在編號3的LED燈而且也遮蓋了編號3
的感測器就得分且發出聲音
        if (randled == 3) { //如果亮燈在編號3的LED燈而且也遮蓋了
編號3的感測器就得分且發出聲音
            music(); //如果亮燈在編號3的LED燈而且也遮蓋了編號3的
感測器就得分且發出聲音
            score = score + 1; //如果亮燈在編號3的LED燈而且也遮蓋
了編號3的感測器就得分且發出聲音
        }
    }
    if (ir4 == 1) { //如果亮燈在編號4的LED燈而且也遮蓋了編號4
的感測器就得分且發出聲音
        ircheck = 4; //如果亮燈在編號4的LED燈而且也遮蓋了編號4
的感測器就得分且發出聲音
        if (randled == 4) { //如果亮燈在編號4的LED燈而且也遮蓋了
編號4的感測器就得分且發出聲音
            music(); //如果亮燈在編號4的LED燈而且也遮蓋了編號4的
感測器就得分且發出聲音
            score = score + 1; //如果亮燈在編號4的LED燈而且也遮蓋
了編號4的感測器就得分且發出聲音
        }
    }
    if (ir5 == 1) { //如果亮燈在編號5的LED燈而且也遮蓋了編號5
的感測器就得分且發出聲音
        ircheck = 5; //如果亮燈在編號5的LED燈而且也遮蓋了編號5
的感測器就得分且發出聲音
        if (randled == 5) { //如果亮燈在編號5的LED燈而且也遮蓋了
編號5的感測器就得分且發出聲音
            music(); //如果亮燈在編號5的LED燈而且也遮蓋了編號5的
感測器就得分且發出聲音
            score = score + 1; //如果亮燈在編號5的LED燈而且也遮蓋
了編號5的感測器就得分且發出聲音
        }
    }
    if (ir6 == 1) { //如果亮燈在編號6的LED燈而且也遮蓋了編號6
的感測器就得分且發出聲音
        ircheck = 6; //如果亮燈在編號6的LED燈而且也遮蓋了編號6
的感測器就得分且發出聲音

```

```

        if (randled == 6) { //如果亮燈在編號 6 的 LED 燈而且也遮蓋了
編號 6 的感測器就得分且發出聲音
            music(); //如果亮燈在編號 6 的 LED 燈而且也遮蓋了編號 6 的
感測器就得分且發出聲音
            score = score + 1; //如果亮燈在編號 6 的 LED 燈而且也遮蓋
了編號 6 的感測器就得分且發出聲音
        }
    }
    if (ircheck != 0) { //ircheck 不等於 0, 代表已有遮蓋過了所以重
新再產生一隨機值(亮燈)並把所有的燈都滅掉
        randled = random(1, 7); //隨機亮燈
        checkpoint = 1; //接下來就要執行主要程式所以將 checkpoint
設為 1
        ircheck = 0; //當 ircheck 等於 0, 代表沒有遮蓋過
        digitalWrite(8, LOW); //將腳位 8 設定輸出低電位
        digitalWrite(9, LOW); //將腳位 9 設定輸出低電位
        digitalWrite(10, LOW); //將腳位 10 設定輸出低電位
        digitalWrite(11, LOW); //將腳位 11 設定輸出低電位
        digitalWrite(12, LOW); //將腳位 12 設定輸出低電位
        digitalWrite(13, LOW); //將腳位 13 設定輸出低電位
        delay(200); //延遲 0.2 秒
    }
    if ((ir1 == 1 && ir3 == 1) && (ir4 == 1 && ir6 == 1)) { //
這為了要重玩來設計的同時按編號 1、3、4、6 的感測器則遊戲會從新開始
        checkpoint = 0; //這為了要重玩來設計的同時按編號 1、3、4、
6 的感測器則遊戲會從新開始
    }
    if (finaltime <= 5) { //當時間剩下五秒或低於五秒發出聲音
        music2(); //當時間剩下五秒或低於五秒發出聲音
    }
    if (finaltime <= 0) { //如果倒數時間小於等於 0, 則到結束的部分
        checkpoint = 2; //如果倒數時間小於等於 0, 則到結束的部分
    }
    mylcd.setCursor(0, 0); //顯示器顯示倒數時間及得分
    mylcd.print(String("Final Time: ") + String(finaltime)); //
顯示器顯示倒數時間及得分
    mylcd.setCursor(0, 1); //顯示器顯示倒數時間及得分

```


四、製作過程



圖 9. 先購買木箱，然後在上面鑽洞。

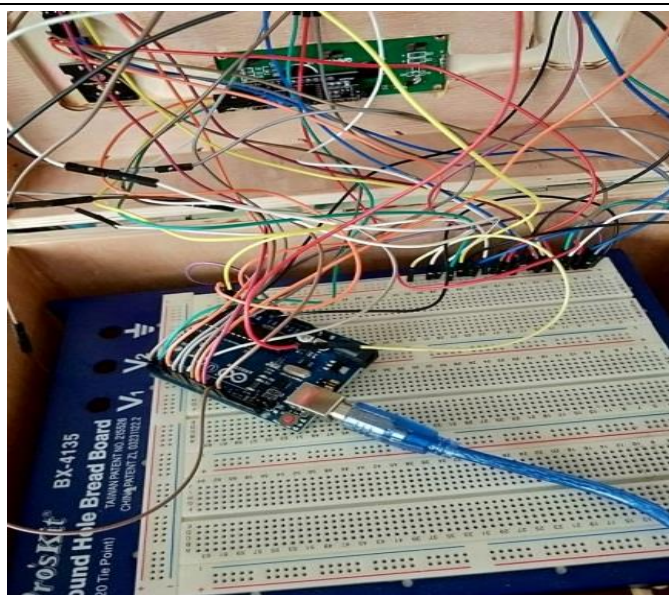


圖 10. 先把LED等等裝在木箱上，後再用麵包板模擬電路。

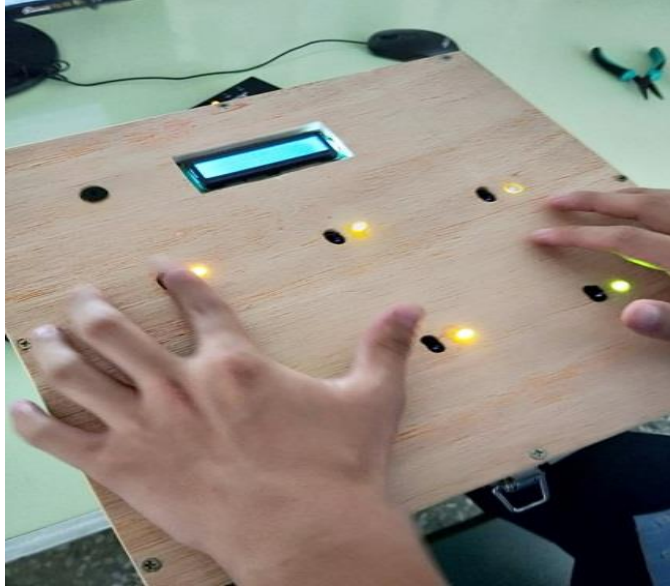


圖 11. 配合 Arduino 程式 進行一次測試，然後從中找出錯誤。



圖 12. 連接行動電源，接上 LED 開始遊戲，完成專題。

伍、研究結果

一、電子打地鼠優缺點

(一)電子打地鼠機優點

1. 材料取得簡單、便宜。
2. 比傳統的打地鼠機功能更多。
3. 能夠隨時拿出來玩。

(二)電子打地鼠機缺點

1. 程式碼的設計有些許困難。
2. 紅外線尋跡模組常常被其他外力受影響。
3. 需要額外用行動電源。

二、與市面上販售的差異

- (一)市面上販售的不需要額外接行動電源，只需要兩個 5V 小電池即可完成。
- (二)市面上販售的電子打地鼠機，花費金錢較高，體積也較小，自己做的體積較大，花費金錢相當來說較便宜。
- (三)市面上販售的電子打地鼠機，比較不易損壞，自己做的線容易卡住之類的，但是這些都是人為可以解決的。

陸、結論

一、討論

這次的打地鼠，我們一開始是使用串聯的方式去把東西連接起來，但是連接起來之後發現，後面的 LED 沒反應，於是後來我們就測試看看並聯接起來，因為並聯的電壓是完全相同，然後串聯會分壓，所以導致我們的作品後面分的電壓不足，後來使用並聯，就完全有電了，還有後來的箱子問題，我們一開始使用的箱子完全沒挖洞過，所以我們就在想辦法去要把作品，led 燈... 等東西裝上去，所以我們就要自己去挖洞，但是一開始挖的不如意，因為我們完全不會，但後來熟能生巧，就都會了，還有我們也有卡在函式庫，我們一開始做完以後測試，發現程式怎麼都上傳不了，都會有誤，經過我們的排除錯誤以後，發現我們沒有載新的函式庫，所以用好之後也就有功能了。

二、結論

這次打地鼠機，我們這組充分的學習到了一些程式，我們以前只知道從網路上找來程式碼 確不知道如何使用函式庫之類的東西，所以這次的作品我們學到了函式庫的使用，以及一開始的插線，我們的作品總共需要 50 幾條，一開始完全不知道怎麼插，但是後來網路上的慢慢學習，我們這組也充分的學習到插線的技巧，這次的專題不是讓我們學習怎麼用程式，怎麼插線而是讓我們學習如何去使用網路查資料，這也是對自己負責的一個方法。

三、心得

這次做的打地鼠機很有意義，不管他的難度難不難，我們整組還是非常的團結的把他做了出來，也學到了很多在 arduino 上的運用，光是一開始的函式庫我們就花了快一個月的時間找了，然而平常我們又沒學到很多的 arduino，但是經由這次的專題我們學到了更多了，一開始我們還很天真的覺得很好做，真正開始做的才發現我們整組的能力不足，有了這次的豐富經驗以後，相信以後上大學以後的專題課，我們能力會更足夠，而不是被隨便一個問題就打倒了，謝謝學校開創這門課，讓我們大家學習到這麼多，同時也進步成獨當一面的人了。

柒、參考資料及其他

一、網頁連結

(電子打地鼠程式碼)

<https://sites.google.com/site/wenyumaker/13ce-shimoto-blockly/13-8da-de-shu>

(LCD 製作方法)

<http://coopermaa2nd.blogspot.com/2010/12/arduino-lab9-2x16-lcd-world.html>

(亂數指令參考)

<https://tw.answers.yahoo.com/question/index?qid=20141027000010KK02475>

(時間倒數參考)

<http://yehnan.blogspot.com/2012/03/arduino.html>

(蜂鳴器參考)

<http://jiader.blogspot.com/2018/08/arduino.html>

二、參考書籍

- 一、作者：楊明豐(2015/04)。Arduino 最佳入門與應用：打造互動設計輕鬆學。出版社：碁峰
- 二、作者：趙英傑(2016/12)。超圖解 Arduino 互動設計入門(第3版)。出版社：旗標
- 三、作者：郭恆鳴(2017/11)。專題製作：Arduino+App Inventor2。出版社：全華

