投稿類別：資訊類

題目：

物聯網智慧傳感器之手指關節體感控制的 3D 投影

作者：
蔡昀珈。台北市私立景文高級中學。資訊科三年 1 班
吳孟哲。台北市私立景文高級中學。資訊科三年 1 班
張凱倫。台北市私立景文高級中學。資訊科三年 1 班

指導老師：
吳永義老師
陳泰岳老師

壹、前言

一、研究動機：

物聯網應用在汽車、能源、智慧大樓、智慧家庭、工業或者醫療等等方面，經常看到了一些有關於在工作使發生的意外事件，技術人員都必須要透過自己的雙手去操控危險的機器，那些機器可能是高溫、高壓，甚至漏電等危險的場所，而那樣的工作環境下是不安全，這讓我們造成工作的危險性，我們想要去從這個地方去探討，想出改善的方法，所以我們決定做出一套系統，並且加入了 Leap Motion 體感控制和全像投影，可以保障操作機器的工作人員的安全，讓他們可以處在一個安全的環境工作下。

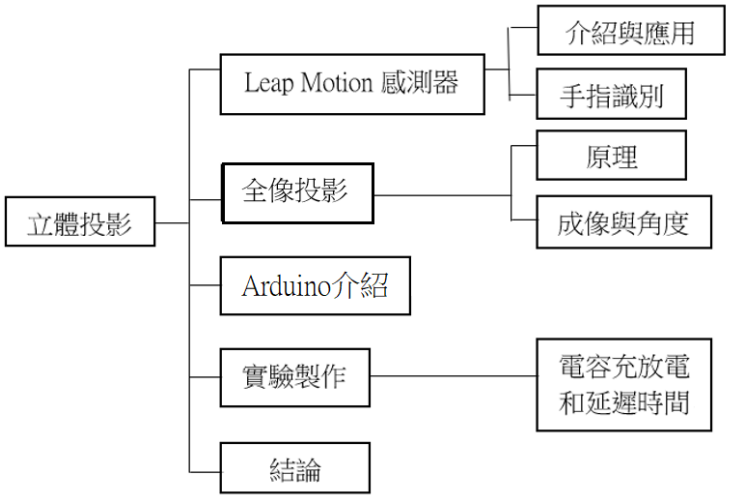
二、研究目的：

本研究為了瞭解金字塔中的立體投影，投影的成像與金字塔的角度是否有關連，以及體感裝置在重複偵測到後，所產生的彈跳現象。

（一）探討體感控制應用及投影的角度是否會影響成像
（二）探討延遲時間和充放電的關係

三、研究方法：

對於本研究我們使用實地模擬操作實驗法，利用 Leap Motion 體感控制及壓克力板作為主要材料，當使用者做出不同手勢時，Leap Motion 體感控制會進行同步動作並偵測空間位置，在將結果以投射的方式到壓克力面板上，藉此達到 3D 立體的效果。研究架構，如下圖(一)表示。



圖(一)研究架構

貳、正文

一、Leap Motion 介紹與應用

體感控制英文原名稱是 Leap Motion，為一種採用體感控制模組的傳感器手指手勢作為輸入。這種傳感器使人能夠以物理方式與 VR 交互。體感控制的研究專注於手指手勢的檢測，體感控制的手勢識別已經廣泛應用，已研究使用跳躍運動傳感器的美國手語識別、在機器學習和應用或者應用在 Leap Motion 體感模塊用於檢測放鬆和抓取手指使用體感控制傳感器切換。它應用領域包括汽車，創意，教育，娛樂，醫療保健，零售和智能設備。例如在醫學應用「**一些研究採用了 Kinect 開發一種低成本的基於遊戲的平衡康復工具以及治療帕金森病患者的傳感器**」(黃柏翰，2015)。

二、體感控制識別手指

圖(二)體感控制器

圖(三)體感控制器

體感控制器的主要工作原理是通過兩個鏡頭模擬人眼捕捉經過紅外 LED 燈照亮的手部影像，利用雙鏡頭畫面建立手的三維立體模型並分析手勢的變化。機體內內置的通信晶片採用標準 USB 傳輸技術，將採集的圖像資訊數位化，轉化的資料傳輸到電腦內，電腦經過圖像識別和運算還原手勢變化並將手部動作傳輸到桌面應用程式實現手勢的直接控制，如圖(二)、圖(三)所示。

在使用過程中，**「Leap Motion 傳感器會定期的發送關於手的運動資訊，每份這樣的信息稱爲「幀」(frame)。每一個這樣的幀包含檢測到的，即所有手指和工具的清單及資訊」**。(CSDN 博客，2016)。 Leap 傳感器會給所有這些分配一個唯一標識（ID），在手掌、手指、工具保持在視野範圍內時，是不會改變的。根據這些 ID，**「可以通過 Frame::hand( )，Frame::finger( ) 等函數來查詢每個運動對象的信息。frame() 函數隨時輪詢幀資料，調用 frame() 或 frame(0) 來獲取最近的幀，Frame 類表示在單個幀中檢測到的一組手和手指跟蹤資料」**(CSDN 博客，2016)。Leap Motion 設備檢測跟蹤區域內的手和手指，以畫面播放速率的方式報告其位置、方向、手勢和運動。**「通過 Hand 類可以得到檢測到的手的身體特徵如手掌位置和速度、手掌正常的方向和手指的方向、適合手的球體屬性和附加的手指的列表. 其中 palm Velocity 方法返回向量表示手掌速度的座標 (X, Y, Z)，direction 方法返回從手掌位置朝向手指的方向的向量」**(王宇航、朱運

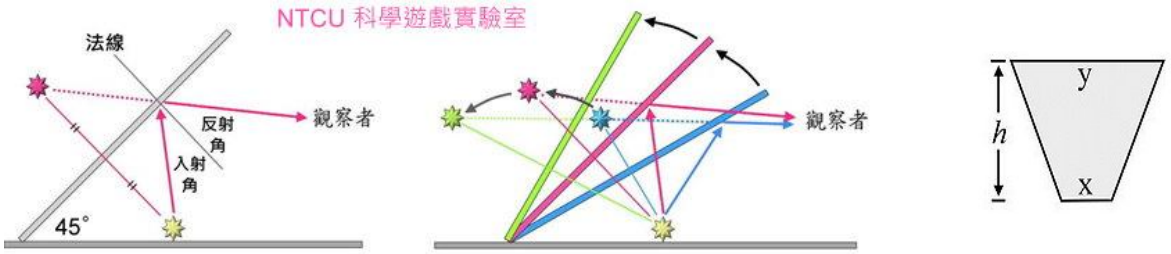
生、孫曉燕，2018) 。**「Frame 類代表了在一幀中檢測到的一組手和手指的追蹤資料。該類中有一個返回值為 Hand List 的 hands 函數,該函數返回了手的物件清單，該清單包含了在當前幀中檢測到的所有手物件」**(劉春、李秋雨，2016)，並把每個手勢的資訊輸出。通過使用上述的類和方法，來實現從 Leap Motion 中獲取到手勢資訊。

三、全息投影原理

可以記錄立體影像的全息影像技術，**「最早可追溯至 1947 年，英國物理學家鄧尼斯・蓋伯在增強電子顯微鏡性能的時候意外發現的技術」**(INSIDE，2016)，使用光波干涉和繞射的現象，紀錄並還原立體的影像。**「另一部分雷射作為參考光束射到全息底片上，和物光束疊加產生干涉，把物體光波上各點的位相和振幅轉換成在空間上變化的強度，從而利用干涉條紋間的反差和間隔將物體光波的全部資訊記錄下來。」**(每日頭條科技版，2016)

四、成像與角度

投影的原理如圖(四)所示，螢幕影像（淡黃色星星）的光線入射到梯形的透明膠片後，經過反射進入觀察者的眼睛（反射角等於入射角），眼睛所看到的影像就會成像在紅色星星的位置。



圖(四) 投影的原理 (資料來源:NTCU 科學教育與應用學系網站) 、圖(五) 立體梯形的尺寸

另一面，觀察者所看到的影像位置，和透明膠片與螢幕的角度有關，例如圖(四)的角度分別為 30 度（藍色）、45 度（紅色）、60 度（綠色），則看到的影像位置，以 45 度（紅色）為最高，而 30 度（藍色）與 60 度（綠色）的影像高度則相同。換言之，透明膠片與螢幕的角度由 30 度、45 度到 60 度時，影像的高度是先上升後下降。

為了看到高度適當的影像，就必須注意立體梯形的尺寸。如圖(五)所示的上底（y）、下底（x）以及高（h），不同的比例就決定了影像的高度。角度 θ 為 45 度可以使影像的高度為最高，更方便觀察。經過計算，**「當 θ 為 45 度，y - x＝1.41・h（1.41 為根號 2 近似值）。因此當 x ＝1cm 時，y＝1+1.41・h。換言之， x ＝1cm 時，y、h 的數值為 6.0cm、3.54cm（或 10.0cm、6.36cm 等等）」**(國立台中教育大學 NTCU 科學教育與應用學系網站，2015)。

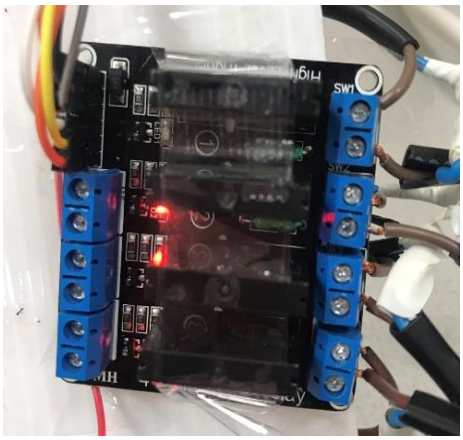
五、 固態繼電器

(一) 工作原理說明

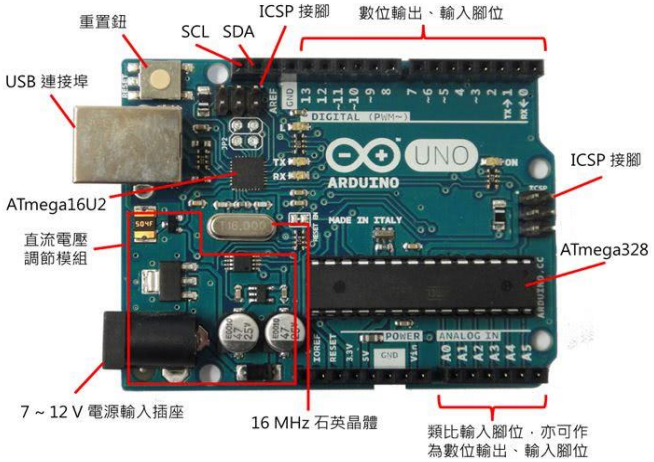
「固態繼電器（Solid State Relay，SSR）是利用一顆 LED 與一顆光 電晶體作成之光耦合器，觸發矽控整流器（SCR）或雙向矽控整流器（TRIAC），因此可以接受低壓（DC 或 AC）信號輸入，而驅 動高壓之輸出」(劉承志、柯啟德 、王文壽，99)。具隔離輸出入及控制高功率輸出之效果。 固態繼電器按負載電源類型可分為交流型和直流型。按開關型式 可分為常開型和常閉型。按隔離型式可分為混合型、變壓器隔離 型和光電隔離型，以光電隔離型為最多，如圖(六)所示。

(二) 固態繼電器的優點

1. 開關速度快、2.工作頻率高、3.使用壽命長、4.雜訊低和工作可靠 (劉承志、柯啟德 、王文壽，2010)。



圖(六) SSR 固態繼電器



圖(七)arduino uno 介紹

六、arduino uno 介紹

「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輸入/輸出引腳，可以連接各種擴充板或麵包板和其他電路。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，也用於從個人電腦載入程式」(維基百科，2018)。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 語言專案的整合開發環境，如圖(七)所示。

六、討論與分析

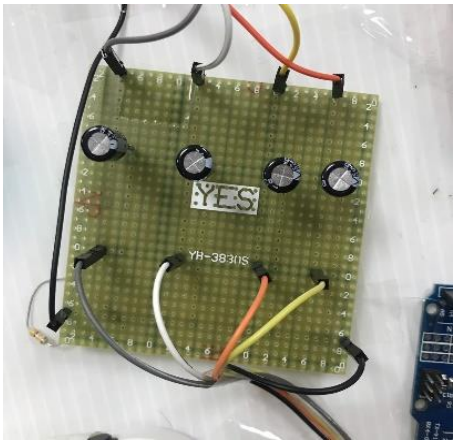
(一) 實驗過程

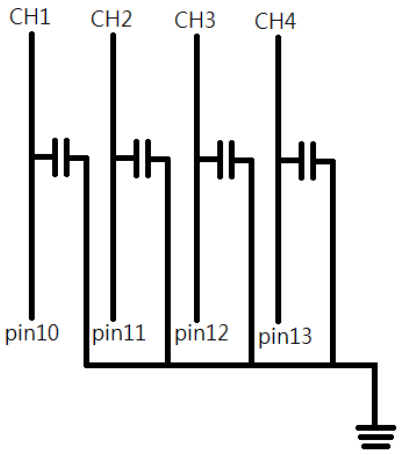
本研究先利用簡單的燈泡進行測驗，利用手掌在體感控制上做出動作來簡單控制四顆圓球，只要把電腦螢幕顯示出來的虛擬手放置在四顆不同的圓球上面，即可對應在左邊燈泡上做出開關動作。握拳代表開啟開關；張開手則是關閉開關。在初期的實驗當中，我們是利用二維座標判斷，當虛擬手的位置在圓球上則進行判斷是張開手還是握拳來控制，其操作介面如圖(八)、圖(九)所示。實驗結果不是很理想，燈的開關常常有彈跳現象，而且對於張開手跟握拳的判定不明確。

圖(八) 操作介面

圖(九) 操作介面

接下來的實驗改用了用三維座標來判斷以改善上述彈跳問題，實驗結果非常顯著，可以明確的判斷手的姿勢做出應對的反應，但是燈泡仍有彈跳現象。為了要消除彈跳現象，我們在程式上加了延遲 500 毫秒來解決彈跳問題，可是延遲也會間接地影響到虛擬手的速度，所以我們在 Arduino 跟繼電器中間加上了一組旁路電容來改善彈跳問題如圖(十)。在使用程式延遲跟外接一組旁路電容上，我們做了一個實驗記錄，如表(一)所示。





圖(十) 旁路電容

表(一) 測試量測結果

		紅燈		綠燈		藍燈		黃燈	
C μF	ms	彈跳次數	百分比	彈跳次數	百分比	彈跳次數	百分比	彈跳次數	百分比
0	0	15 次	15%	21 次	21%	13 次	13%	22 次	22%
0	20	5 次	5%	4 次	4%	3 次	3%	7 次	7%
100	0	3 次	3%	3 次	3%	3 次	3%	5 次	5%
100	20	0 次	0%	0 次	0%	1 次	1%	3 次	3%

(二) 問題改善

這個實驗室在做每一顆燈泡個實測 100 次開關所發生的彈跳現象。第一組實驗是在沒有延遲跟旁路電容下情況下所做出來的結果，很容易發生彈跳現象；第二組則是在有延遲 20 毫秒沒有旁路電容的情況下所做出的實驗，彈跳次數明顯下降每一顆燈泡的正確執行率都高於 90%以上；第三組實驗跟第二組實驗的結果沒有太大的差異；第四組則是使用延遲 20 毫秒跟外加 1 組電容大小 100 μF 的旁路電容來做實驗，正確開關率已經到達 95%以上。這次的實驗黃燈錯誤率明顯比每一組實驗的燈泡都高，其原因是因為手掌、手指必須與 Leap Motion 垂直，而黃燈的位置是四顆燈泡中最高的，當手掌升高時 Leap Motion 跟手指的位 置又距離太遠，因此所讀取的位置較容易出錯，時常出現出明明手的姿勢是比五，但是電腦螢幕上的虛擬手勢出現出食指或中指一指的指頭朝下，導致出現錯誤判斷因而有彈跳現象。其他三顆燈泡也是一樣，有錯誤的部分大多數都是 Leap Motion 偵測手的位置有問題因而倒置彈跳現象產生，由紀錄證明增加旁路電容可以改善彈跳效應的問題。

參、結論

Leap Motion 探測器已經實現了使用放鬆和抓握手指的體感控制開關。該系統包括虛擬現實中出現的彩色圓圈，Leap Motion 傳感器模塊，Arduino UNO 控制器模塊，固態繼電器模塊，彩色燈泡和交流電源。該系統可以控制任何交流或直流電氣設備，但不僅僅是本研究中的燈泡，因此，發生了一些閃爍和錯誤，分別在程式和 Arduino 的 I / O 引腳中插入了 20 ms 的時間延遲和 100 μF 的旁路電容。經過 100 次時間延遲和旁路電容的組合後，結果表明使用 100 μF 旁路電容而沒有時間延遲的誤差最小（小於 5%）。因此，交換機系統的進一步應用將促進 VR 與物理世界之間的操作介面，本研究加入了全像投影功能，讓操作畫面從 2D 變成 3D，達到功能的目的。

未來建議

本項研究應用於物聯網智慧城市之穿戴式元件，在生產製造業，可以避免在工作中不遭受到危險的事件，在危險的生產工作提升作業自動化，可以提高生產效率及安全的完成生產工作的智慧工廠，也能夠應用於智慧生活的遊戲方面，例如:虛擬實境，讓玩家體驗到三維空間的立體感。

肆、引註資料

(書籍資料)

- 黃柏翰(2015)。交通大學生醫工程研究所學位元論文。用 Leap Motion 實作手指顫抖的分析並與帕金森氏症做連結。
- 王宇航,、朱運生、孫曉燕(2018)。杭州師範大學杭州國際服務工程學院。基於 Leap Motion 的手術室無干擾交互技術。
- 劉春、李秋雨(2016)。瀋陽航空航太大學航空製造工藝數位化國防重點學科實驗室 。LeapMotion 體感控制器及其在飛機結構展示系統中的應用。
- 劉承志、柯啟德 、王文壽(2010)。崑山科技大學電子工程系四技部專題研究報告。跳機。

(網路資料)

- CSDN 博客(2016)。2016 年 08 月 26 日，取自 https://blog.csdn.net/andylauren/article/details/52326995
- INSIDE 博客(2016)。2016 年 3 月 2 日，取自 https://www.inside.com.tw/2016/03/02/new-viral-craze-create-holograms-with-your-smartphone
- 每日頭條科技版(2016)。2016 年 7 月 11 日，取自 https://kknews.cc/zh-tw/tech/yy845k.html
- 國立台中教育大學 NTCU 科學教育與應用學系網站(2015)。2015 年 8 月 1 日，取自
- http://scigame.ntcu.edu.tw/light/light-039.html
- 維基百科(2018)。2018 年 10 月 2 日 ，取自 https://zh.wikipedia.org/wiki/Arduino

伍、附錄: Processing 程式

```
import de.voidplus.leapmotion.*;
import processing.serial.*;
Serial port;
LeapMotion leap;
int i;                                                                    //boolean a=false,b=false,c=false,d=false;
float d;
void setup() {
                    size(800, 800, OPENGL);
                background(50);
                   println(Serial.list());
                String arduinoPort = Serial.list()[0];
                port = new Serial(this, arduinoPort, 9600);
                leap = new LeapMotion(this);
               }
void draw() {
                background(50);
                fill(255, 255, 0);                              //uper_Yellow
                noStroke();                                     //設定不使用邊線
                ellipse(400, 150, 200, 200);                    //(x,y,xr,yr) UP_ON
                fill(255, 255, 0);                              //Lower_Green
                noStroke();                                     //設定不使用邊線
                ellipse(400, 650, 200, 200);                    //(x,y,xr,yr) Lower_Off
                fill(255, 255, 0);                              //Left_Red
                noStroke();                                     //設定不使用邊線
                ellipse(150, 400, 200, 200);                    //(x,y,xr,yr) Left_On
                fill(255, 255, 0);                              //設定填滿的顏色為隨機的
                noStroke();                                     //設定不使用邊線
                ellipse(650, 400, 200, 200);                    //(x,y,xr,yr) Right_On
for (Hand hand : leap.getHands ()) {
                                                        hand.draw(20);
     for (Finger finger : hand.getFingers()) {
                                                              PVector hand_position = hand.getPosition();
                                                              PVector finger_position = finger.getPosition();
       finger.drawLines();
       finger.drawJoints(20);
       println("X:",hand_position.x,"Y:",hand_position.y,"Z:",hand_position.z);
       println("Xf:",finger_position.x,"Yf:",finger_position.y,"Zf:",finger_position.z);
              d=sqrt((hand_position.x-finger_position.x)*(hand_position.x-    finger_position.x)+(hand_position.y-finger_position.y)*
                 (hand_position.y-finger_position.y)+(hand_position.z-finger_position.z)*(hand_position.z-finger_position.z));
       println("d:",d);
       if(((hand_position.x-400)*(hand_position.x-400)+(hand_position.y-150)*(hand_position.y-150))<10000){
                           fill(255, 255, 0);                   //設定填滿的顏色為隨機的
                           noStroke();                          //設定不使用邊線
                           ellipse(400, 150, 200, 200);         //Upper_On
        if (d<100){
                    port.write("G");
                  }
        if (d>120){
                   port.write("F");                             //else
                  }
                        }
         else if(((hand_position.x-400)*(hand_position.x-400)+(hand_position.y-650)*(hand_position.y- 650))<10000){
                           fill(255, 255, 0);                   //設定填滿的顏色為隨機的
                           noStroke();                          //設定不使用邊線
                           ellipse(400, 650, 200, 200);         //Upper_On
         if (d<100){
                    port.write("C");
                   }
```

```
if ((hand_position.x-finger_position.x)*(hand_position.x-finger_position.x)>10000)
          if (d>120) {
                          port.write("B");
                         }
           loadPixels();
           for (int i =(width*height/2)+1; i < width*height; i++) {
                                                                  pixels[(width*height/2)*2-i] =pixels[i];
                                                                 }
           updatePixels();
           loadPixels();
           for (int i =0; i < height; i++) {
                                             for (int j=i;j<height-i; j++){
                                                                           pixels[j*height+i] =pixels[i*width+j];
                                                                          }
                                           }
           updatePixels();
           loadPixels();
           for (int j =0; j < width; j++) {
                                           for (int i=j;i<height; i++){
                                                                       pixels[j*height+i] =pixels[i*width+j];
                                                                      }
                                         }
           updatePixels();
                      }
else if(((hand_position.x-150)*(hand_position.x-150)+(hand_position.y-400)*(hand_position.y-400))<10000){
                          fill(255, 255, 0);                          //設定填滿的顏色為隨機的
                          noStroke();                                 //設定不使用邊線
                          ellipse(150, 400, 200, 200);                //Upper_On
if (d<100){
           port.write("M");
          }
if ((hand_position.x-finger_position.x)*(hand_position.x-finger_position.x)>10000)
          if (d>120) {
                       port.write("L");
                      }
                 }
else if(((hand_position.x-650)*(hand_position.x-650)+(hand_position.y-400)*(hand_position.y-400))<10000){
                           fill(255, 255, 0);                         //設定填滿的顏色為隨機的
                           noStroke();                                //設定不使用邊線
                           ellipse(650, 400, 200, 200);               //Upper_On
          if (d<100){
                     ort.write("U");
                     for(i=1; i<500;i++)
       {
         port.write("U");
         delay(1);
         port.write("R");
         delay(50);
       }
            }
          if ((hand_position.x-finger_position.x)*(hand_position.x-finger_position.x)>10000)
          if (d>120)   {
                        port.write("R");
                       }
              delay(20);
               }
                 }
                   }
```