

臺北市立大安高級工業職業學校專題實作及創意競賽
「專題組」作品說明書



群別：電機與電子群

作品名稱：NICE CAR

關鍵詞：FPV 鏡頭、ESP8266、第一人稱遙控車

目錄

壹、摘要.....	1
貳、研究動機	1
參、主題與課程之相關性或教學單元之說明	1
一、硬體製作.....	1
(一)數位邏輯實習.....	1
(二)電子學實習.....	2
二、程式撰寫.....	2
(一)Arduino.....	2
(二)C#.....	3
肆、研究方法.....	
一、研究流程.....	3
二、使用材料.....	4
三、使用軟體與服務.....	5
伍、研究結果	6
一、主體結構.....	6
二、軟體通訊架構.....	6
(一)電腦端.....	7
(二)ESP8266 及 Ardunio UNO 端	8
陸、討論.....	9
一、更大的電池.....	9
二、FPV 傳輸範圍跟穩定性	9
三、更輕的產品.....	9
柒、結論.....	10
捌、參考資料	10
玖、附錄：	11
一、作品分工表	11
二、競賽日誌.....	11

圖目錄

圖 1	ARDUINO	2
圖 2	C#	3
圖 3	ARDUINO UNO	4
圖 4	ESP8266	4
圖 5	Inventor	5
圖 6	ARDUINO IDE	5
圖 7	UNITY	5
圖 8	流程圖	6
圖 9	Unity 介面.....	7
圖 10	C#介面	7
圖 11	UNO 程式碼	8
圖 12	ESP8266 程式碼.....	8

表目錄

表 1	Arduino UNO	4
表 2	ESP8266	4

【NICE CAR】

壹、摘要

這個專題旨在透過將 FPV（第一人稱視角）鏡頭安裝在遙控車上，實現一種更直觀的駕駛體驗。利用 FPV 鏡頭，使用者可以從車輛的視角進行控制，仿佛置身於車內，達到沉浸式的操作感。為了實現遙控車的遠程控制，我們選擇了 ESP8266 作為無線通信的核心模塊，通過 Wi-Fi 將指令傳輸至遙控車，並且使用方向盤與油門控制車輛的移動與轉向。

這個專題不僅增強了駕駛過程中的互動性，也提供了一個創新的方式來駕駛遙控車，實現了第一人稱視角的駕駛體驗。

貳、研究動機

我們的研究動機源自於對駕駛高性能跑車或賽車的嚮往。然而，對於大多數人來說，擁有這樣的車輛往往需要龐大的財力支撐，對我們學生而言更是難以企及。因此，我們開始思考，是否有可能用較低的成本創造一種能夠模擬駕駛快感的方式，讓更多人能夠體驗到駕駛的樂趣。這種願望成為了我們研究的初心。

除了經濟限制，我們也注意到傳統的遙控車操作方式較為單調，使用者僅能通過遙控器以第三人稱視角進行操作，缺乏身臨其境的沉浸感。因此，我們希望透過技術創新，讓遙控車的操作更具互動性與真實感，實現從駕駛者視角操作車輛的效果，進一步提升使用體驗。

為了實現這一目標，我們決定結合 FPV（第一人稱視角）技術與遙控車。FPV 技術讓使用者能夠直接觀看車輛的視角畫面，彷彿親自坐在車內操控一樣，提供更逼真的駕駛感受。同時，我們選擇了 ESP8266 作為無線通訊核心模組，以低成本實現遠程控制功能，並搭配方向盤與油門，模擬實車的操作方式，讓駕駛體驗更為完整。

透過這個專題，我們希望突破現有遙控車的使用框架，讓它不僅僅是一個娛樂工具，更成為一種創新的駕駛模擬平台。我們的目標不僅是滿足自身對駕駛的熱愛，還希望為更多有相同夢想的人提供一個低門檻、富有樂趣的駕駛體驗方式。

參、主題與課程之相關性或教學單元之說明

一、硬體製作

(一)、數位邏輯實習

我們利用高二時所學的麵包板接線技術來對車上的供電系統進行分配，並且對線路進行整理。

(二)、電子學實習

我們利用高二電子學實習時所學，對電子元件進行焊接，並且使用降壓模組來對系統電壓降至能夠供給 Arduino UNO 及 ESP8266 進行正常且穩定的運行。

二、程式撰寫

(一)Arduino

我們利用高二智慧居家監控實習中學到的 Arduino（圖 1），作為 NICE CAR 核心晶片 ESP8266 和 Arduino UNO 的驅動程式，使其能連接網路並完成轉向和油門數值的接收、解碼和輸出。

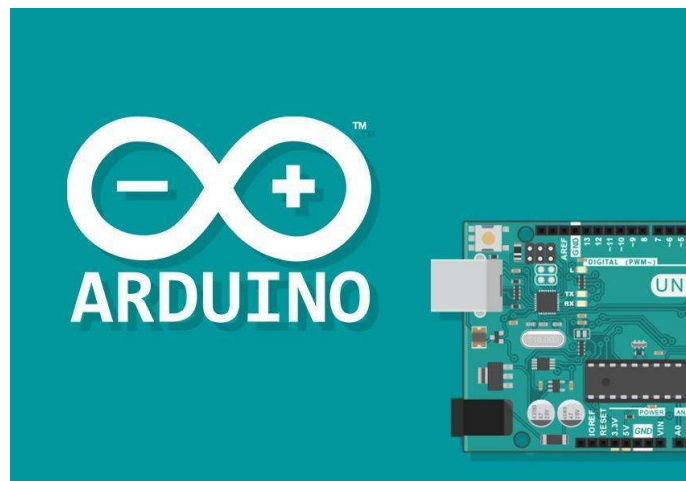


圖1 Arduino

(二)、C#

C#(圖 2) 是由微軟開發的現代物件導向程式語言，主要用於 .NET 平台。它的語法簡單、強型別，支援物件導向特性如類別、繼承和多型。C#擁有自動垃圾回收、LINQ 查詢語法、並且支援跨平台開發(.NET Core)。它適用於桌面、Web 和行動應用程式開發。

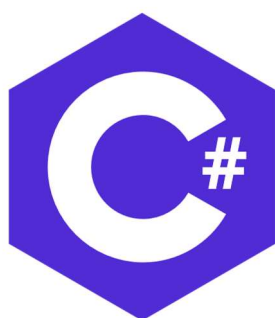


圖2 C#

肆、研究方法

一、 研究流程

在六月底訂下專題題目後，組員們開始卯足全力查找相關網路資料，並在八月底買了第一批測試元件，著手麵包板接線測試和基礎程式編寫。十月初做 WIFI 無線連接和訊號類型測試。十月中旬連線及訊號傳輸調教完成。在十一月時進行硬體組裝和除錯，並優化軟體系統和硬體接線。到了十二月尾聲車身也被設計完成。其餘時間都花在拍攝影片和除錯，且在隔年一月測試了供電穩定性和動作影片拍攝。

二、 使用材料

(一)、Arduino UNO

Arduino 系列電路板的設計大多使用 Atmel AVR 單片機。這些電路板配有一組數字和類比 I/O 引腳，可以連接各種擴展板或麵包板(Shields 擴充版)和其他電路。我們使用 Arduino UNO(圖 3)來進行六位數訊號的解碼，並傳送至遙控車執行動作。其規格如表 1。



圖3 Arduino UNO

表1 Arduino UNO

Arduino UNO	
尺寸	53.4mm*75mm
輸入電壓	3.0V ~ 3.6V
外接電源	7.4V 鋰電池
頻率	16MH

(二)、ESP8266

ESP8266(圖 4)內置了 1MB 快閃記憶體，其允許單晶片設備能夠直接連接到 Wi-Fi 進行使用。這是我們最重要的通訊設備，我們用它來接收由電腦端發送的訊號並傳送給 Arduino UNO 使用。

其規格如表 2。

表2 ESP8266

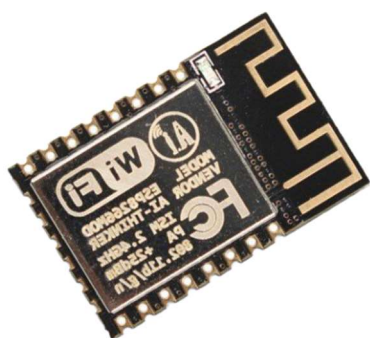


圖4 ESP8266

ESP8266	
晶片	ESP8266-12F
輸入電壓	7.4V 鋰電池
輸出電流	800MA
頻率	80 MHZ&160 MHZ

三、 使用軟體與服務

(一)、Autodesk Inventor

一款能輕鬆入門的工業繪圖軟體，擔任這次專題的硬體設計擔當，我們用它來建模和繪製草圖，是硬體製作的根本。



圖5 Inventor

(二)、Arduino IDE

作為 Arduino 語法的開發環境，能配合多種晶片進行編譯和多樣的程式庫可供使用。我們用其對 ESP8266 晶片和 Arduino UNO 版進行程式的編譯。



圖6 Arduino IDE

(三)、Unity

作為免費且最泛用的開發引擎，能整合各種外接設備，同時兼具 WIFI 連接，與 Arduino 互動等功能。我們用其讀取方向盤及踏板的數值，並對 ESP8266 晶片和 Arduino UNO 版進行資料傳遞。



圖 7 Unity

伍、研究結果

一、主體結構

以遙控車為主體，裝載兩種開發版，分別負責網路連線和發送動作指令。加上 FPV 鏡頭和傳輸設備，實現第一人稱駕駛。

二、軟體通訊架構

設備通訊方面，我們最終選擇使用網路通訊，原因有兩點：第一，網路伺服器我們已有相關架設經驗，會比使用藍牙系統更好上手和除錯；第二，藍牙模組有時會和晶片產生衝突並且來要另外多加一顆藍牙模組在系統中，大幅增加操作難度和整體大小。

在整個通訊過程中使用了 2 種程式語言，C++ 和 C#，透過網路傳輸裡最合適的 UDP 協定，延遲和資料丟失率都大幅減少，同時 unity 和 Arduino 都具備 UDP 協定的傳接功能，流程圖如(圖 8)。

1.Unity 軟件讀取方向盤和踏板數值

2. 用數學式整合成 6 位數變數

3. 透過 UDP 連網傳輸數值給 ESP8266

4. Esp8266 接收後用類比訊號傳遞給 Arduino UNO

5. Arduino UNO 將數值拆解成轉速、轉向

6. 把兩個數值發送給遙控車

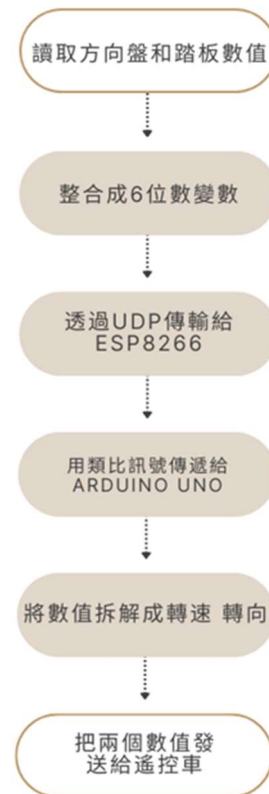


圖8 流程圖

(一)、電腦端

負責執行數值的轉換，把檢測到方向盤的值，通過 UNITY(圖 5)經過數學式轉換成 6 位數的值後(圖 6)，通過 WIFI 傳給遙控車。

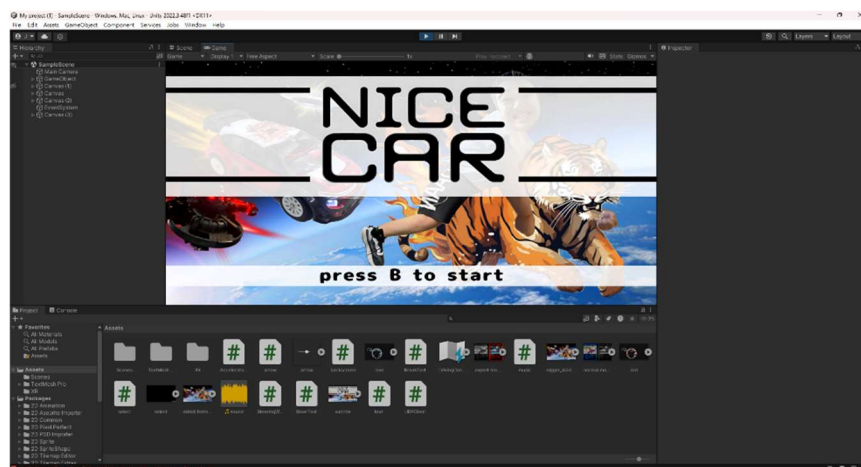


圖 9 Unity 介面

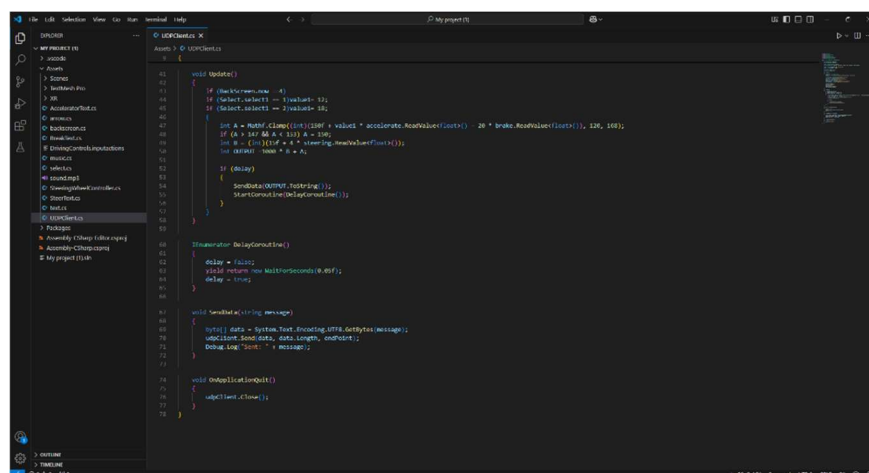


圖 10 C#介面

(二)、ESP8266 及 Arduino UNO 端

UDP 協定一次只能傳輸一種資料類型，遙控車控制轉速、轉向又限定 int 變數，我們選擇將兩個數值整合成一個變數，而不是使用陣列傳輸後轉變資料類型，這樣會消耗過多內存空間和造成大量延遲。

遙控車控制轉速、轉向的程式邏輯是透過輸入 1100~1900 μ s 區間的電流週期(圖 8)，實現伺服馬達和無刷馬達的變向、變速。我們先是將方向盤的數值(str:-1~1)等比放大成 1100~1900；踏板的數值(acc:0~1)等比放大成(1300~1700)，再分別將個位數去除，使其變為兩個三位數字，再將方向盤(str/10)數值乘上 1000

後與踏板($acc/10$)數值相加($str*1000+acc$)發送給 esp8266，Arduino UNO(圖 7)收到後會先除 1000 還原方向盤數值(小數點後面會自動去除，那是 int 資料類型在 C++語法裡的內建功能)，乘 10 使遙控車能夠辨識(此時方向盤數值會回到最初的 str)，再將收到數值(receive)減去 100 倍方向盤數值($receive-100*str$)，乘 10 還原原本的踏板數值(acc)，再分別發送給無刷馬達、伺服馬達。

```

36
37 void loop() {
38   unsigned long currentMillis = millis(); // 取得當前時間
39   // 每隔 1000 毫秒輸出 1500 微秒的 PWM 波
40   myServo.writeMicroseconds(ppmWriteWidth); // 輸出到舵機的 PWM 頻率
41   myServo.writeMicroseconds(ppmWriteWidth);
42   myServo.writeMicroseconds(MPin);
43
44   // 檢查是否已經過了 3 秒
45   if (currentMillis - previousMillis >= interval) {
46     // 每隔 3 秒接收新的舵機輸入
47     if (!isReady) {
48       isReady = true;
49       Serial.println("Ready to receive input after 3 seconds.");
50     }
51   }
52
53   // 如果已經過了 3 秒，並從串口中輸入
54   if (isReady && Serial.available() > 0) {
55     long input = Serial.parseInt(); // 讀取從串口中接收到的整數數字
56     result = input/100000;
57     motor = result*10;
58     str = (input - motor*10000)/1000;
59     str = str * 100;
60     acc = 10*(input - motor*10000 - str*10);
61
62     // 確保讀到的是有用的數字 (如果沒有讀到有效數字，parameter 是 0)
63     if (acc != 0) {

```

圖 11 UNO 程式碼

```

18
19 void setup() {
20   Serial.begin(115200);
21   Serial.begin(115200);
22 }
23 // 配置 GPIO 和 TP 和 CM
24 //WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS);
25
26 // 選擇 Wi-Fi
27
28 // 選擇 ESP8266 的 TP 地址
29 WiFi.begin(ssid, password);
30 while (WiFi.status() != WL_CONNECTED) {
31   delay(1000);
32   Serial.println("Connecting to WiFi...");
33 }
34
35 // 配置 ESP8266 的 TP 地址
36 Serial.println("Connected to WiFi!");
37 Serial.print("ESP IP Address: ");
38 Serial.println(WiFi.localIP());
39
40 // 開始接收 UDP 數據
41 UDP.begin(5555);
42 Serial.println("Listening for UDP packets on port 5555");
43 Serial.begin(115200);
44 WiFi.mode(WIFI_STA);
45 }
46
47 void loop() {

```

圖 12 ESP8266 程式碼

陸、討論

一、更大的電池

目前我們的專題只能運行 13 分鐘，對於遊玩體驗是大打折扣，所以我們希望能找到更輕便但容量大的電池。

二、FPV 傳輸範圍跟穩定性

連線實驗，我們明顯發現連線強度在長距離，會出現嚴重的閃爍，因為 FPV 傳輸器是利用無線電通訊，並用天線接收 FPV 傳輸器發射的訊號來讀取畫面。所以如果車子跑太遠或訊號雜亂，都會使其閃爍或

斷線。

為此，我們希望能想出其他的傳輸來取代目前的無線電傳輸，在現階段我們討論出的最佳做法是也使用 WIFI 傳輸，這樣能讓畫面更清楚並穩定。

三、更輕的產品

我們專題的初衷是讓開車走向人群，並讓人們方便開車和隨心所欲的駕駛，但礙於各類板子的重量，我們的產品始終無法做到符合我們期望的重量。

為此我們希望研發自己的板子。事實上經過討論傳輸板子不難自製，我們都知道它的原理，搭配上電路板雕刻技術，實現自製更輕的板子應該可行，不然直接更換傳輸方法也是值得嘗試的選擇。

柒、結論

本專題成功地將 FPV（第一人稱視角）技術應用於遙控車的操作，實現了從駕駛者視角進行操控的目標，為使用者提供了一種沉浸式的駕駛體驗。透過將 FPV 鏡頭安裝於遙控車上，並利用 ESP8266 作為核心的無線通信模組，我們成功建立了穩定的 Wi-Fi 連接，實現了遠程指令傳輸和車輛的即時操控。使用者可藉由方向盤與油門進行直觀的移動與轉向控制，仿佛親臨車內駕駛，增強了操作的真實感與趣味性。

在實現過程中，我們克服了硬體整合、通信延遲與穩定性等技術挑戰，最終完成了一個功能完善且互動性強的系統。這項成果不僅為遙控車的操作方式帶來了創新，也展示了 FPV 技術在娛樂與教育領域的應用潛力。此外，專題的開發過程為無線技術與沉浸式互動結合的應用研究提供了參考價值，並對未來更先進的遙控技術設計具有啟發意義。

綜上所述，本專題的成功不僅實現了技術上的創新，還為探索更具沉浸式的遙控操作模式開啟了新的可能性，進一步推動了 FPV 技術在多元領域的應用發展。

捌、參考資料

書面資料

1. 金奎曾 (2017/09/01)。Unity 3D/2D 手機遊戲開發：從學習到產品（電子書）。清華大學出版社。
2. Marco Schwartz (2017/06/14)。實戰物聯網開發使用 ESP8266(電子書)。基峰。
3. 曹永忠、許智誠、蔡英德 (2013)。Arduino 遙控車設計與製作。

渥瑪數位有限公司。

網路資料

1. GitHub(2017)，取自於

[Unity-Technologies/InputSystem](https://github.com/Unity-Technologies/InputSystem)。 <https://github.com/Unity-Technologies/InputSystem>

2. 池丞昊（2021/09/09）。Unity 介面與物件基礎操作。取自於
<https://hackmd.io/@HowHao/BJEQosBuP>

3. projecthub.arduino(2024)。Remote control car running on Arduino。取自於

<https://projecthub.arduino.cc/hibit/remote-control-car-running-on-arduino-2e4358>

玖、附錄：

一、作品分工表

參賽學生	工作任務
A	程式撰寫、影片拍攝
B	3D 建模
C	程式撰寫、書面報告製作
D	程式撰寫、硬體組裝

二、競賽日誌

			進度	紀錄	工作分配
113	5	6	資料蒐集	地點：科辦 器材：手機、電腦 時數：3 小時	同學 A：資料查詢 同學 B：資料查詢 同學 C：資料查詢 同學 D：資料查詢
113	6	6	討論主題	地點：科辦 時數：1 小時	同學 A：討論主題 同學 B：討論主題 同學 C：討論主題 同學 D：討論主題
113	6	20	結構討論 功能討論	地點：組長家 時數：3 小時	同學 A：功能討論 同學 B：功能討論 同學 C：結構討論 同學 D：結構討論

113	7	2	結構討論 功能討論	地點：組長家 時數：3 小時	同學 A：結構討論 同學 B：結構討論 同學 C：結構討論 同學 D：結構討論
113	7	25	討論專題細節	地點：組長家 器材：手機、紙 時數：2 小時	同學 A：功能討論 同學 B：功能討論 同學 C：結構討論 同學 D：結構討論
113	8	2	材料購買	地點：光華商場 時間：3 小時	同學 A：材料購買 同學 B：材料購買 同學 C：材料購買 同學 D：材料購買
113	9	12	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆 電 時間：6 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：程式撰寫 同學 D：程式撰寫
113	9	18	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆 電 時間：6 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：程式撰寫 同學 D：硬體組裝
113	9	23	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆 電 時間：7 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：硬體整合 同學 D：硬體組裝
113	9	25	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆 電 時間：7 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：硬體整合 同學 D：硬體組裝
113	9	26	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆 電 時間：6 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：硬體整合 同學 D：硬體組裝
113	9	27	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆 電 時間：7 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：硬體整合 同學 D：硬體設計

113	10	28	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆電 時間：7 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：硬體整合 同學 D：硬體組裝
113	11	16	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆電 時間：7 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：功能測試 同學 D：硬體組裝
113	11	23	軟體製作 3D 建模 硬體製作	地點：實習工廠 器材：手機、筆電 時間：7 小時	同學 A：程式撰寫 同學 B：3D 建模 同學 C：功能測試 同學 D：硬體組裝
113	12	23	簡報製作	地點：實習工廠 器材：手機、筆電 時間：7 小時	同學 A：簡報製作 同學 B：簡報製作 同學 C：簡報製作 同學 D：簡報製作
114	1	1	簡報製作	地點：組長家 器材：手機、筆電 時間：7 小時	同學 A：簡報製作 同學 B：簡報製作 同學 C：簡報製作 同學 D：簡報製作
114	1	3	簡報製作	地點：組長家 器材：手機、筆電 時間：7 小時	同學 A：影片構思 同學 B：影片構思 同學 C：影片構思 同學 D：影片構思
114	1	4	影片拍攝、剪輯	地點：組長家 器材：手機、筆電 時間：7 小時	同學 A：影片拍攝、剪輯 同學 B：影片拍攝 同學 C：影片拍攝 同學 D：影片拍攝
114	1	7	簡報製作	地點：組長家 器材：手機、筆電 時間：7 小時	同學 A：簡報製作 同學 B：簡報製作 同學 C：簡報製作 同學 D：簡報製作

114	1	7	簡報製作 最終測試	地點：實習工廠 器材：手機、筆電 時間：7 小時	同學 A：簡報修改 同學 B：報告練習 同學 C：最終測試 同學 D：最終測試
-----	---	---	--------------	--------------------------------	--