3D 列表機之程式設計

四電三乙 蕭文暉、李權哲
指導老師：郭英哲

零・摘要

3D 列印又稱為快速成形技術，3D 列印技術為近期開始興起的一項先進製造技術。此項技術對於需要事先打模提供樣本以及醫療等各方面需求的產業，此項技術讓各種產業節省了比傳統技術所需要更多時間，因而使效率變高。本專題是以 3D 列印技術，使用步進馬達、RAMPS1.4 馬達驅動擴充板、Arduino 單晶片、LCD 控制面板。將設計完成的圖檔存入 SD 卡中，開始打印成果至成品上。

一・目的

本專題研究目的在於了解 3D 印表機的運作原理，並依照自己所期望的功能，寫出該台印表機的專用韌體，雖然網路上已有專門的通用韌體，但程式碼繁雜，不易理解，所以自行撰寫韌體程式碼，如此一來，在未來需要增加新功能時，因為韌體是自己所撰寫的，所以依照自己需要的方向去添加，也比修改他人的容易許多。此外還可以學到各種軟硬體之間的應用，了解工廠生產時，常使用的 Gcode 指令，以及提升針對設計時產生問題之除錯能力。

二・編譯器

圖一、採用 Arduino IDE 編寫程式

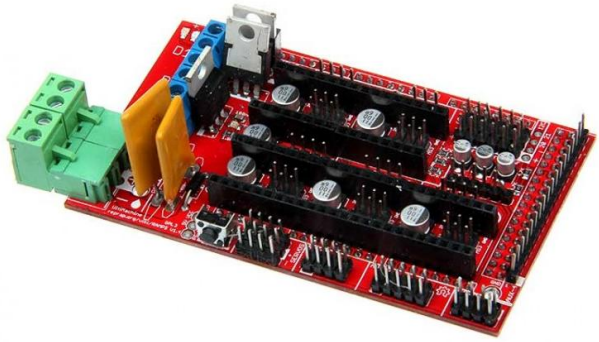
三・硬體架構

表一、重要硬體清單

名稱	數量	圖片編號
Arduino Mega 2560（主控版）	1 片	圖二
RAMPS 1.4（馬達驅動擴充控制板）	1 片	圖三
Smart Controller LCD+SD（顯示控制器）	1 片	圖四
A4988（步進馬達驅動晶片）	4 片	圖五
步進馬達 17HS4417	5 顆	圖六
加熱底板 MK2B	1 片	圖七
J-HEAD 加熱噴頭（0.4mm）	1 顆	圖八

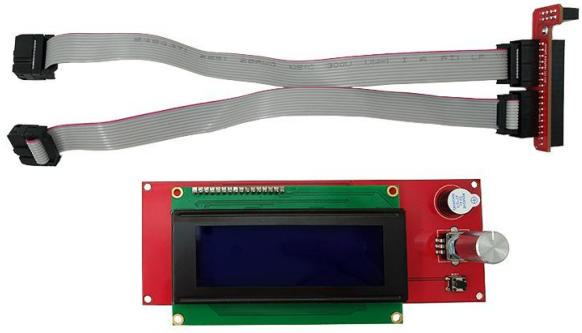
圖二、Arduino Mega 2560

Arduino Mega 2560 擁有 16MHz 時脈（Clock Speed）、256KB 的快閃記憶體（Flash Memory）、8KB 的靜態隨機存取記憶體（SRAM）、以及多達 54 支的輸出入腳位（I/O pins），作為設計 3D 印表機程式之核心晶片已非常充足。



圖三、RAMPS 1.4 擴充板

RAMPS 是 RepRap Arduino Mega Pololu Shield 的縮寫，主要是設計給步進馬達驅動使用的介面電路，1.4 則為電路的版本號，本擴充板直接接在 Mega 2560 主控版上。



圖四、Smart Controller LCD + SD

此顯示控制器上含有 LCD 顯示面板、旋鈕、蜂鳴器，用以控制印表機動作及顯示並提醒印表機當前狀態，背後還有 SD 卡模組，可用來存取 Gcode 列印檔。

圖五、A4988 晶片

此晶片是步進馬達使用的微步驅動器，最大特點為只要輸入一個脈衝，即可驅動步進馬達產生微步，而無需進行複雜的相位順序表。

圖六、步進馬達 17HS4417

在此專題中，我們選擇使用步進馬達，而非伺服馬達，原因在於，伺服馬達會為了精準定位，而產生晃動，這會破壞列印品質，故採用步進馬達。

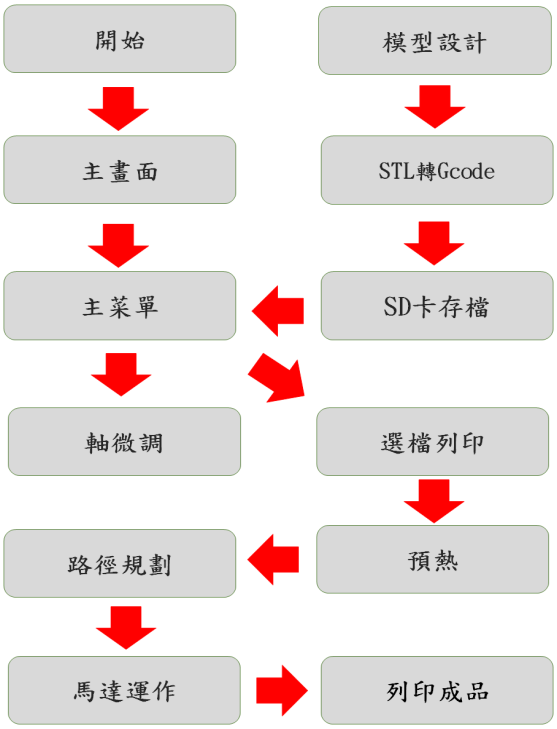
圖七、加熱底板 MK2B

加熱底板的用途在於，剛開始列印時讓玻璃有一定溫度，以增加塑料的黏著性，降低在列印過程中產生蹺曲的狀況，並附有熱敏電阻，讓主控版作溫度控制。

圖八、J-HEAD 加熱噴頭

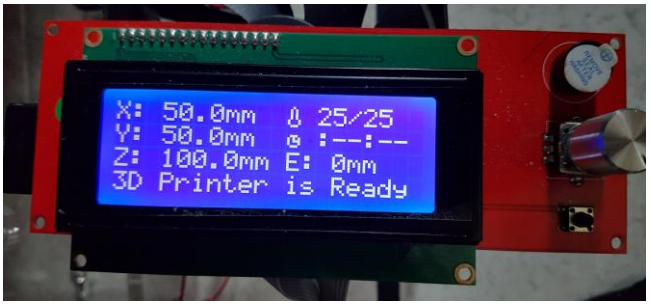
加熱噴頭用以融化並擠出塑料，另有熱敏電阻，讓主控版作溫度控制。

四・系統流程



圖十、系統流程圖

1. **開始**：執行程式，以 LCD 顯示主畫面。
2. **主畫面**：顯示當前各軸座標與加熱頭溫度、加熱板溫度、現在列表機狀態。



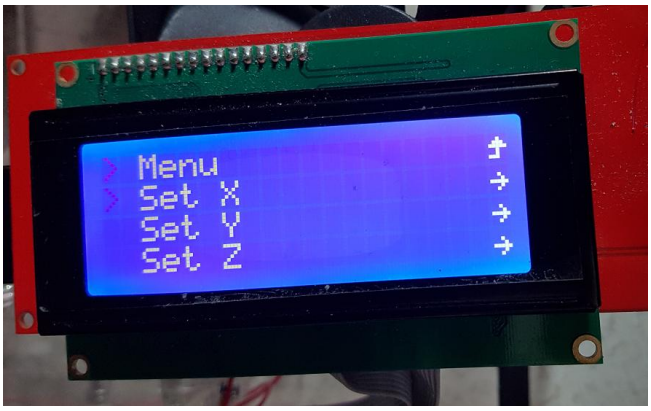
圖十一、LCD 面版主畫面

(1) 第一行:顯示 X 軸作標與加熱頭與加熱版溫度
(2) 第二行:顯示 Y 軸作標與列印所需時間
(3) 第三行:顯示 Z 軸作標與 E 軸長度
(4) 第四行:顯示列表機狀態

3. **主菜單**：顯示返回主畫面、加熱噴頭位置微調、SD 卡檔案選擇。

圖十二、LCD 面版選單

4. **軸微調**：調整加熱噴頭位置，以利剛開始列印時，塑料能固定在板上。



圖十三、LCD 面版軸微調畫面

圖十四、各軸移動範圍畫面

能夠進行各軸的調整，有 10mm、1mm、0.1mm 來進行各軸設定

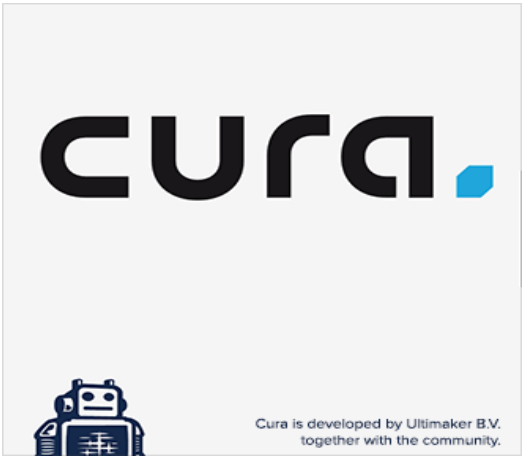
5. **選檔列印：**從 SD 卡內選擇檔案列印。

圖十五、選取列印檔案畫面

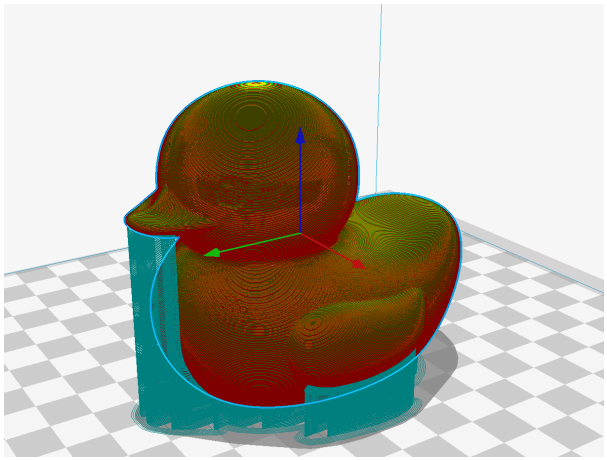
6. **模型設計：**使用 Sketch Up 之類的軟體，建立欲列印之 3D 模型。

圖十六、採用 Sketch up 繪製 3D 模型

7. **STL 轉 Gcode：**使用 Cura 之類的軟體，將 3D 模型檔(STL)，切片分析成 Gcode 指令檔。



圖十六、採用 Cura 轉換成 gcode 檔



圖十七、切片示意圖

8. **SD 卡存檔:**將檔案儲存在 SD 卡，之後讓 Arduino 讀取 Gcode 檔列印。

9. **預熱：**列印模型之前，須將塑料加熱至一定溫度，才能列印，設有溫度控制系統。
10. **路徑規劃：**從 Gcode 檔取出之指令，處理成步進馬達運作時之訊號。

11. **馬達運作：**使用計時器中斷程式，來確保加熱噴頭移動速度一致。

12. **列印成品：**顯示主畫面，並擠出塑料開始列印，直到完成成品。

五．Gcode 指令

Gcode 指令的主要功能為指導機器如何在三個維度上做幾何移動，不過也可以指示機器做非幾何的東西，像是本專題的 3D 印表機，指定塑料擠出速度，或是改變加熱床溫度等。

Gcode 是一種簡單的程式語言，沒有像 C++、JAVA 之類的，擁有條件判斷、循環迴圈、可變因素等先進的結構，只有每行代表某一特定任務，讓印表機一行行的執行指令，直到結束。

3D 印表機上常用的 Gcode 指令：

G0：快速移動噴頭到指定位置。

例：G0 X12.3 Y45.6 Z7.89

即噴頭移動到 X 軸 12.3mm、Y 軸 45.6mm、Z 軸 7.89mm 的位置。

G1：當噴頭擠料時，移動到指定位置。

例：G1 X90.6 Y13.8 E22.4 F5000

即移動噴頭到 X 軸 90.6mm、Y 軸 13.8mm，同時擠料到 22.4mm，以 5000mm/min 的速度移動。

G20：設定單位為 英吋。

G21：設定單位為 公厘。

G28：移動到原點，並為求精確，當觸碰到限位開關時，會後退 1mm，再上去定位一次

G90：設定為絕對座標。

G91：設定為相對座標。

G92：設定位置。

M106：設定風扇轉速。

例：M106 S127

設定風扇以半速旋轉。S 範圍（0~255）

M107：關閉風扇。

M109：設定加熱噴頭溫度，並等待加熱床加熱到指定溫度，再執行下一動作。

例：M109 S200

設噴頭溫度為 200 度並加熱。

M117：顯示訊息。

例：M117 Printing...

在 LCD 上顯示 Printing...字串。

M119：檢測微動開關狀態。

M190：設定加熱床溫度，並等待加熱床加熱到指定溫度，在執行下一動作。

五・程式碼

使用 Arduino IDE 撰寫，為了區分功能方便閱讀，一共分成 18 個檔案，依名稱排序，如表二所示：

表二、程式碼檔案清單

Buzzer.h	LcdDisplay_PrintSet.h
CharTable.h	LcdDisplay_stepper.h
DoBackGround.h	Pin.h
Interrupt.h	PinMode.h
Knob.h	Printer.ino
LcdDisplay.h	Sdcard.h
LcdDisplay_Main.h	Temperature.h
LcdDisplay_Moving.h	WorkAssign.h
LcdDisplay_Prepare.h	WorkAssign_Init.h

在這些檔案裡，可以粗分為初始設定、溫度控制、SD 卡讀取、步進馬達控制、LCD 控制，這幾部分。

1. 初始設定
   (1) Printer.ino

   副檔名.ino 檔為 Arduino IDE 設計程式時，作為開頭跑的檔案，由裡面 setup()函數，設定程式所需變數、腳位之初始值，再用 loop()函數下去跑迴圈重複執行想要做的功能，為了美觀，我們只呼叫其他檔案的函式、不再裡面作運算處理。

   (2) Pin.h

   在這裡面，我們定義腳位的名稱，方便後面作使用，也定義一些常用的數值。

   (3) PinMode.h

   用來設置腳位為輸出入的初始設定。

2. 溫度控制
   (1) Temperature.h

   裡面設有一個大陣列，用來快速轉換由熱敏電阻讀出的數值。

   函式部分一共四個分別計算加熱噴頭、加熱板之溫度，以及兩者之加熱控制。

3. SD 卡讀取
   (1) Sdcard.h

   負責 SD 初始設定，並讀取 SD 卡內部所有 Gcode 檔名，至於讀取 Gcode 檔之指令被寫在 LcdDisplay_PrintSet.h 內部。

4. 步進馬達控制
   (1) Interrupt.h

   控制馬達是採用計時器中斷的方式進行，以

20 微秒的精度進行一次中斷，讓移動速度能夠保持一致，不會被其他程式，如：更新 LCD 畫面之類的，造成多餘延遲。

另外設置一條陣列用來儲存處理好的馬達移動指令，可以減少讀取 Gcode 指令到處理成馬達用的移動指令這段時間造成的延遲。

(2) LcdDisplay_Moving.h

用於選擇軸微調時，相對應之馬達移動指令輸出。有各軸 10mm、1mm、0.1mm 三種距離之移動指令轉換，再呼叫 Interrupt.h 內部函數，變成實際馬達運作。

(3) LcdDisplay_PrintSet.h

負責 Gcode 檔指令的讀取，並對其作路徑規劃，再呼叫 Interrupt.h 內部函數，變成實際馬達運作。

5. LCD 控制

(1) CharTable.h

由於此款 LCD 支援自定義圖案，所以設置了一些來使用。如：溫度計、箭頭符號。

(2) LcdDisplay.h

負責宣告 LCD 的顯示畫面大小。

(3) LcdDisplay_Main.h

負責主畫面的顯示，包含 XYZ 軸位置、加熱噴頭、加熱板溫度、E 軸已擠出長度、列表機狀態之資訊回報，以及顯示主選單項目及連結各項目對應之函數。

(4) LcdDisplay_stepper.h

負責顯示微調四軸時的項目及連結各項目對應之函數。

(5) WorkAssign.h

用於連接現在 LCD 該顯示的畫面，使用代碼和 switch、case 實現。

(6) WorkAssign_Init.h

宣告 LCD 常用函數

6. 其他

(1) Buzzer.h

這是用來作為提醒音效用的，如列印完成時，會響起一首短曲。

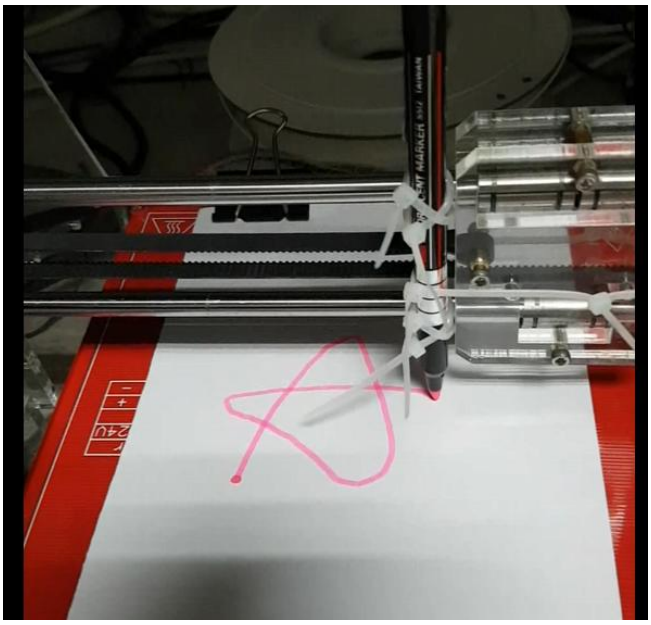
(2) DoBackGround.h

這是用腳位訊號作出類似中斷效果的程式，用來判斷有無回歸原點用的、以及發生噴頭或是本子移動要超出範圍時的保險，以避免印表機損壞。

六・研究過程

硬體部分，由於是買套件，所以沒什麼困難就組裝完成。之後開始寫程式，我們分成兩個部分同時進行，一個人設計 Arduino 的 LCD 選單顯示部分，另一個人使用與 Arduino IDE 相似的 DEV C++進行 Gcode 指令檔轉換步進馬達用脈衝訊號，並輸出檔案，讓 Arduino 直接執行步進馬達訊號檔。

完成大致功能後，先以在 3D 印表機上綁上筆做繪圖，並製作一個簡單的繪圖軟體轉 Gcode 檔，來測試 Gcode 指令檔之轉換有無問題，來防止實機運作時出錯。測試結果時，當中有幾個小問題，像是馬達反向運動、脈波問題、跟運算有些微誤差等等，但很快就修正了。



圖十八、綁筆繪製測試

之後將DEV C++的程式放入Arduino後，讓Arduino靠自己完成 Gcode 檔之轉換。結果出現了一個大問題，

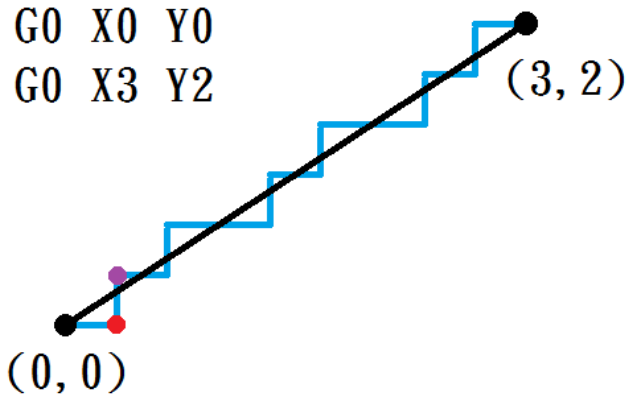
移動路徑是對的，但馬達移動速度變的很緩慢，尤其是轉彎的時候。

後來發現原因在於，程式內使用 Delay 函數來間隔下一次發送脈衝訊號的時間，所以在 Delay 函數運行時，Arduino 是完全閒置的，直到發送馬達訊號出去後才去計算下一次發送訊號所需的間隔，無形中造成了不少浪費，且在 Gcode 指令換行執行時，需要從 SD 卡內讀出下一行指令，再將指令轉換成所需的數字，最後做計算路徑的前置處理，才能開始判斷哪軸的步進馬達，要在哪個時間點動作，這一段程式處理下來，所需要的時間並不短，所以造成轉彎時，會有非常明顯的停頓感，至於直線部分，少了前段的處理程式，故只有略微變慢。

這個問題，我們改用計時器中斷來改善，如此一來可以保證不會因為其他程式造成多餘延遲，也可以在等待觸發計時器中斷時，先行處理下一個要發送的訊號。

並且重新寫出新的 Gcode 指令檔轉馬達脈衝訊號，精簡程式的計算式。

原本為四軸(X、Y、Z、E 軸)計算各自將會動作的時間點列表，透過一起比大小排出動作順序，再算出兩兩相隔的 Delay 函數用時間、並輸出欲動作之腳位。此方法，若有不須動作軸，則容易跑大量無用迴圈，因為四軸都要判斷比動作時間點大小。



圖十九、斜率判斷法

利用初始點與終點算出初始斜率，來判斷他是否該移動 X 軸或 Y 軸，若斜率大於 1（角度為 45 度） 則移動 Y 軸，若小於等於 1 則移動 X 軸，該圖中初始判斷移動 X 軸，使得噴頭位置移向紅點，然後再用紅點算出與終點的斜率來判斷，得出方向移動 Y 軸到達紫點，重複循環上述步驟，直至噴嘴抵達終點，如圖十九之藍線路徑。

改成使用斜率來快速判斷 X 或 Y 軸的步進馬達動作順序，E 軸則每執行數次 X 或 Y 軸時，就動作一步，Z 軸則放在 XY 軸執行完後再運作，因為，至於延遲的時間則已在內部定義一個數值，這樣一來，對 Arduino 晶片的需要運算的量能有效降低。

再來又重試了一遍，整體速度有了提升，但轉彎時的停頓感依舊很重，想出來的方法就是使用大量陣列儲存已完成的步進馬達訊號，來取代一次處理一個訊號完成後，等待中斷完成上一個指令發送訊號後，才接收這次指令的窘境，以此解決轉彎時的停頓感，因為有陣列儲存之前處理好的訊號，讓中斷程式慢慢去處理，爭取到更多時間，來處理 Gcode 檔換行時需要的運算。

七・列印成果

圖二十、3D 列印骰子

圖二十一、3D 列印鴨子

八・參考文獻

1. 3D 印表機的 7 大成型技術
http://www.techbang.com/posts/18161-3d-printer-technology-talk

2.3D 列印
https://zh.wikipedia.org/wiki/3D%E6%89%93%E5%8D%B0

3.由步進馬達的基礎認識到使用方法
https://www.orientalmotor.com.tw/image/web_seminar/stkiso/20130307_stkiso_seminar.pdf

4.A4988 步進馬達控制器
https://chenfuguo.gitbooks.io/arduino/content/Shields/a4988Controller.html

5. RepRapDiscount Smart Controller
http://reprap.org/wiki/RepRapDiscount_Smart_Controller

6.從 Arduino 到 AVR 晶片 (2) -- Interrupts 中斷處 理 (作者：Cooper Maa)
http://programmermagazine.github.io/201407/htm/article1.html

7.attachInterrupt() 與外部中斷
http://coopermaa2nd.blogspot.tw/2011/04/attachinterrupt.html

8. ABS 或 PLA？挑選線材的正確方法
http://www.makezine.com.tw/make2599131456/abspla