

中華民國第 53 屆中小學科學展覽會

作品說明書

高職組 電子、電機及資訊科

最佳創意獎

091006

示波器動畫

學校名稱：國立新竹高級工業職業學校

作者： 職二 陳奕捷 職二 黃偉哲 職三 張子晏	指導老師： 王文雄
---	------------------

關鍵詞：示波器

示波器動畫

摘要

示波器，替代我們的雙眼，深入電路，呈現最真實的電壓信號。連結CH1，將輸出訊號以閃電般的光束穿梭在電路和示波器之間。這回，不再是週期規律的正弦波，更不是排列縝密的方波，而是一場前所未有的視覺饗宴。若你對示波器的定義，只不過是台工具罷了，那可就大錯特錯了，我們將一段經由策劃、計算，轉換而成的音訊信號輸出，打破以往的觀念，經由示波器顯示，引領各位重回70年代以單調色彩、詼諧內容所發行的黑白電影，讓電波活靈活現，在屏幕上放映著未對外公開的大作。

壹、研究動機

高二，開始接觸示波器的應用，起初接觸的當然就是基本的 Normal Mode，僅顯示電壓波形、相位角度及大小……等等。

後來，藉由課程介紹了解 X-Y Mode，進而表現李沙育圖形，這回又讓我們接觸到了一個新的表示方法，原來示波器用途如此廣闊，能進而以類似數學方法表現，使使用者能夠更一目了然。

就因如此，便引起了很大的興趣，希望能夠更進一步的瞭解示波器，探索其它更鮮為人知，但實用的功能所在。

經由網路以及老師、學長的解答下，我們又得知了 Matlab 這個運算程式，可以更完整表達我們腦中的思想，這又讓我們更加興奮了，原來示波器的應用可以到那麼廣，還可配合電腦達到播放動畫的效果。

於是，就此興趣，便組了一個的團隊，開始對示波器展開一番研究，已達到滿足我們對示波器瞭解的渴望。

貳、研究目的

一、在示波器上製造出動畫或圖形:

單單只會做出工作波形，這並非難事，你我應該都會做，但要做出一個真正流暢的影片那可真不容易，除了要真正了解其構造外還需幾分數學方面的巧思。

二、藉由示波器動畫帶來不一樣的實作心得及上課樂趣:

為了更加方便了解示波器的顯示原理，不單單僅有電壓方面的呈現，而是更深層的科學體驗，希望將日常生活中的平凡加註在我們的示波器上；在實習課裡，旋轉著呆版的時間旋鈕，免強稱得上生動，但只是敷衍而已，因此為了讓課堂多幾分樂趣，我們需要讓

電壓以生動的方式呈現在你我眼前。

參、 研究設備與器材

一、軟硬體設備與器材:

研究中所使用的硬體與軟體，如下表所示：

硬體設備

名稱	規格	數量(單位)
個人電腦	CPU：i7-2600K RAM：DDR3 1333 6G HDD：SATA 3.0Gb/s 320GB	一台
Leader 20 MHz 示波器	20 MHz	一台
音源線	3.5mm to RCA	一條

表 1 硬體設備

軟體設備

名稱	版本	數量
音樂播放軟體	Media player	一套
PC 作業系統	Windows 7	一套
轉譯軟體	Matlab	一套

表 2 軟體設備

二、數學軟體 Matlab

Matlab是MATrix LABoratory的縮寫，是一款由美國The MathWorks公司出品的商業數學軟體。Matlab是一種用於演算法開發、資料視覺化、資料分析以及數值計算的高階技術計算語言和互動式環境。除了矩陣運算、繪製函式/資料影像等常用功能外，MATLAB還可以用來建立使用者介面及與呼叫其它語言（包括C，C++和FORTRAN）編寫的程式。儘管 MATLAB 主要用於數值運算，但利用為數眾多的附加工具箱（Toolbox）它也適合不同領域的應用，例如控制系統設計與分析、影像處理、訊號處理與通訊、金融建模和分析等。另外還有一個配套軟體包 Simulink，提供了一個視覺化開發環境，常用於系統類比、動態/嵌入式系統開發等方面。

(一)MATLAB主要提供以下功能：

- 1、可用於技術計算的高階語言。
- 2、可對代碼、檔案和資料進行管理的開發環境。
- 3、可用於線性代數、統計、傅立葉分析、篩選、最佳化以及數值積分等的數學函式。
- 4、可用於視覺化資料的二維和三維圖形函式。
- 5、可用於構建自訂的圖形化使用者界面的各種工具。
- 6、可將基於 MATLAB 的演算法與外部應用程式和語言（如 C、C++、Fortran、Java、COM 以及 Microsoft Excel）整合的各種函式。

以下是Matlab程式的執行畫面：

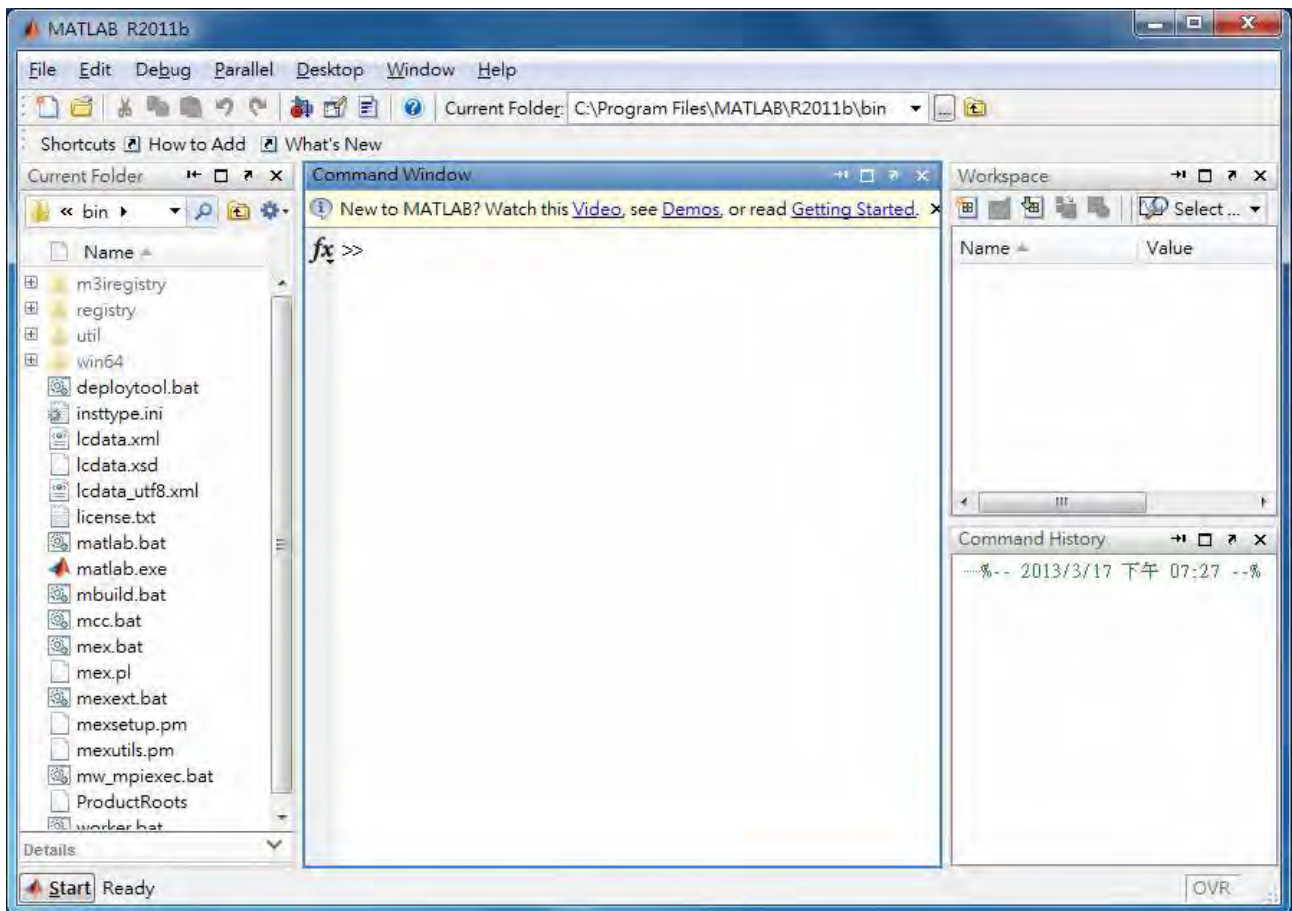


圖 1 Matlab 執行畫面

肆、研究過程或方法

一、研究流程

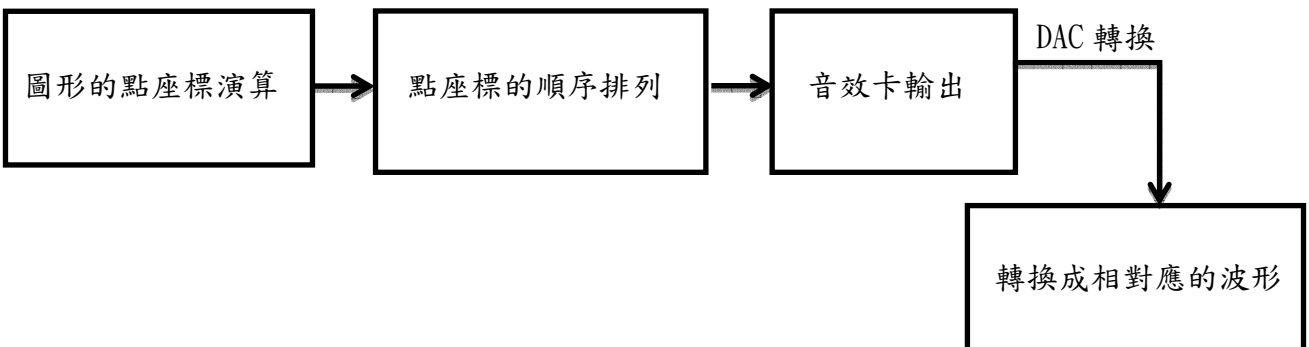


圖 2 研究流程

二、示波器的原理

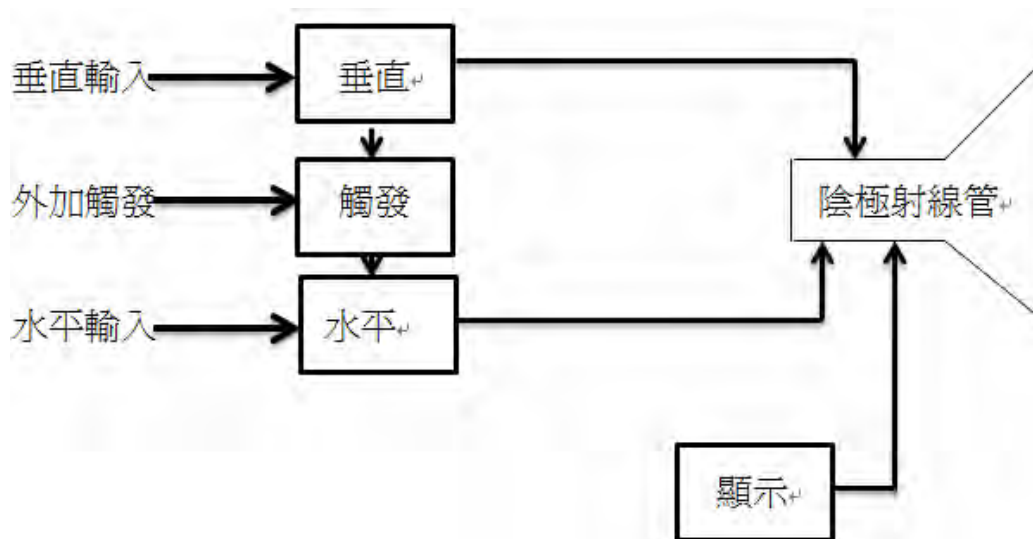


圖 3 示波器內部構造

(一)何謂 X-Y 模式？

首先，示波器顯示，其背後靠著一股強勁的力量推近，藉由陰極線管發射電子束，輸入待測電壓投射在垂直偏向板，再進而產生鋸齒波，給以掃描動作，並經水平偏向板以單位時間做為調整轉軸；而 X-Y 模式，則是 CH1 信號接水平偏向板取代原有的鋸齒波，而 CH2 信號接垂直偏向板，以類似數學點座標方式呈現，而兩間所產生的電場變化即是我們要觀察的結果，如下圖。



圖 4 X - Y 模式實際表示

1.李沙育圖形

輸入兩正弦波信號至 CH1 和 CH2，並旋鈕轉至 X-Y mode，若兩信號頻率比為有理數，使電子連續打出加上電場的規律變化，形成規律運動的圖形，螢光屏幕上即顯示李沙育圖形。

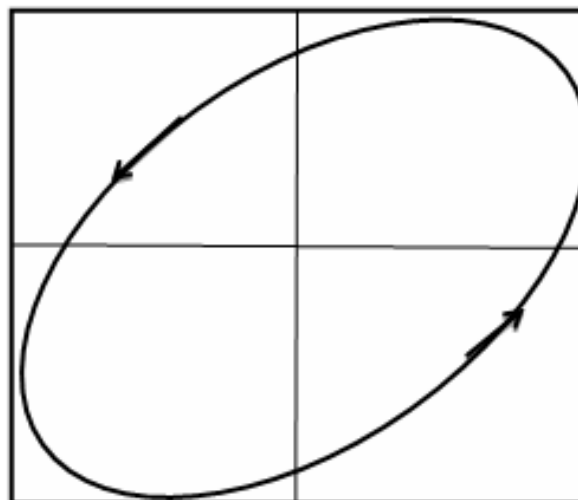


圖 5 李沙育圖形

但，要精準的控制電壓波形似乎還不夠，必須利用 X-Y 模式，給予其準確的電壓座標，

就好比數學象限，以快速而連續的指定座標，才可呈現出所想要的動畫，以電影影格模式達到視覺暫留的成像效果。

(二)DAC 轉換

Digital to analog converter，英文簡稱DAC，是一種將數位信號轉換為類比信號的設備，以電流、電壓或電荷的形式存在。而類比數位轉換(ADC)即為其的反向功能。可藉由這些轉換器，將數位信號轉換為容易讀取或是別的類比信號，方便生活方面的應用。

(三)將數位信號轉為音訊

藉由 MATLAB 寫出數學函式，描繪出圖形，用電腦轉換為音訊，由音效卡輸出，將數位的音訊信號藉由類比方式輸出，賦予示波器電壓信號，逐點描繪，連線著成一圖形。電腦中的聲音檔大部分都是 wav.，將音訊用數字向量來表現，若檔案過大，我們可以只取他部份的點

例:

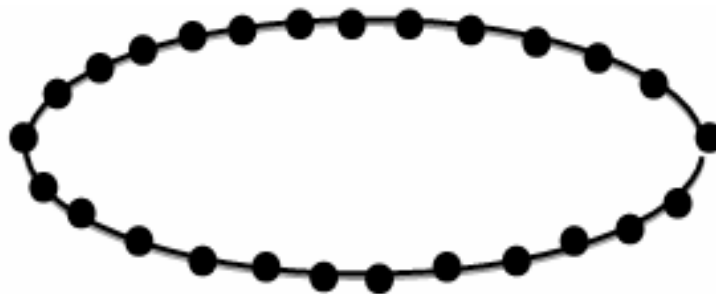


圖 6 逐點描繪

三、音效卡

(一)數位信號與類比信號

1.何謂類比信號？

類比即為一種「連續的變化量」，如溫度或速度的變化。例如電壓，由 12V→12.1V→12.12V
.....

2.何謂數位信號？

數位為一種分散式的信號，無法以連續方式表現，而是以單位量作為增減，例如 100 人、200 隻。

(1)數位信號的優點

- (1)不易受雜訊干擾
- (2)容易儲存、控制
- (3)傳輸過程中，衰減或失真信號容易被還原
- (4)容易做信號匹配

(二)音效卡

由於數位信號是 0 與 1 之間的位階變化，而非類比信號連續不間斷，能有效控制輸出，因此我們藉由 DAC (digital to analog converter)轉換器，將電腦內部的數位信號轉換為示波器看得懂的類比信號，得以間接控制。

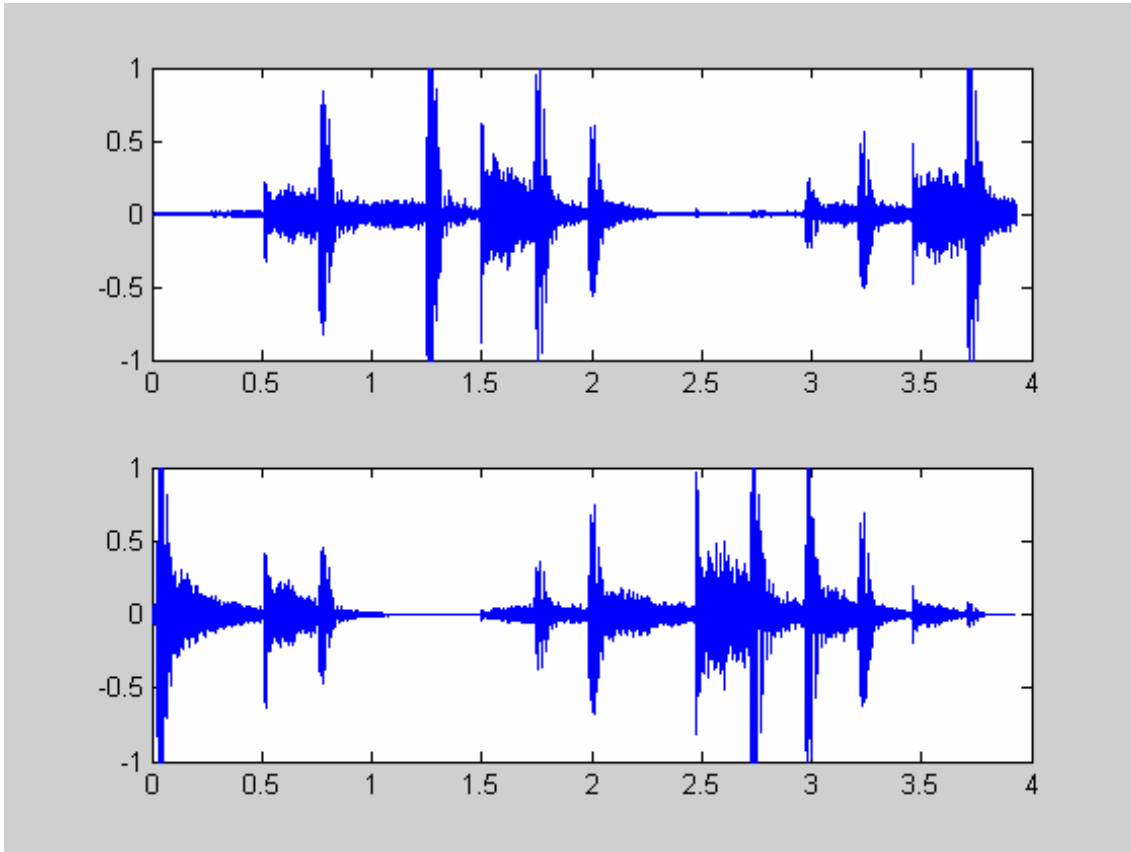


圖 7 音效卡輸出

為了得到較佳的播放效果，在 Matlab 當中可使用指令 `soundsc`，針對音訊數值進行正規化，使其介於 -1 和 1 之間，在經由音效卡傳送至喇叭輸出，以達到極佳的表現。Matlab 轉換完成後輸出為音效 `.wav` 檔，在 Matlab 下，為兩行陣列，一行為右聲道，另一則為左聲道，以 `x` 軸做為右聲道的資料，`y` 軸做為左聲道，直接在 Matlab 下合併成聲音，此轉換法已寫成 Matlab 程式。

(二)蝙蝠俠程式碼

$$\left(\frac{x}{7}\right)^2 \sqrt{\frac{|x|-3}{|x|-3}} + \left(\frac{y}{3}\right)^2 \sqrt{\frac{y + \frac{3\sqrt{33}}{7}}{y + \frac{3\sqrt{33}}{7}}} - 1 = 0$$

由上數學方程式，我們可將其拆解分析，並更為Matlab程式碼輸入，得知4圖可和為一橢圓；又因根式內必須大於零，再加上座標的限制，因此使橢圓被拆成兩半。

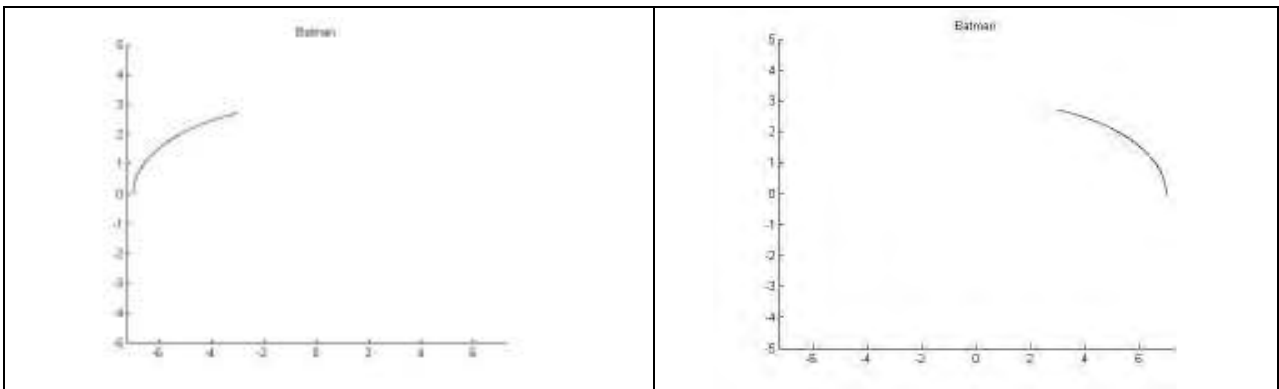


圖 8 程式碼

圖 9 程式碼

<pre>>>x4=linspace(-7,-3); y4=(3.*sqrt(1-(x4./7).^2));</pre>	<pre>>>x12=linspace(3,7); y12=(3.*sqrt(1-(x12./7).^2));</pre>
--	---

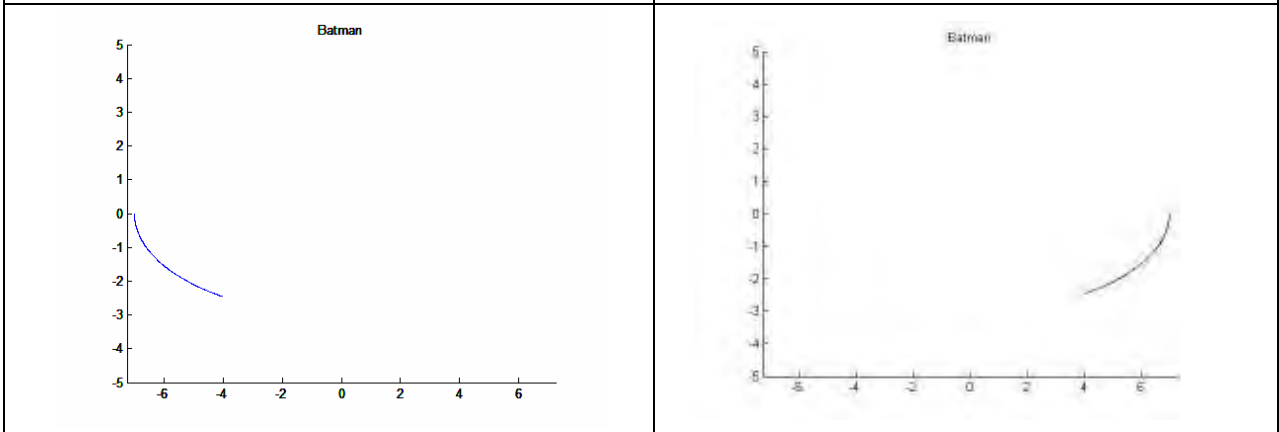


圖10 程式碼

圖11 程式碼

<pre>>>x3=linspace(-4,-7); y3=(-3.*sqrt(1-(x3./7).^2));</pre>	<pre>>> x1=linspace(7,4); y1=(-3.*sqrt(1-(x1./7).^2));</pre>
---	--

$$\left| \frac{x}{2} \right| - \left(\frac{3\sqrt{33} - 7}{112} \right)^2 - 3 + \sqrt{1 - (||x| - 2| - 1)^2} - y = 0$$

同上，根式必須大於零，再加上座標化，使其分為4點。

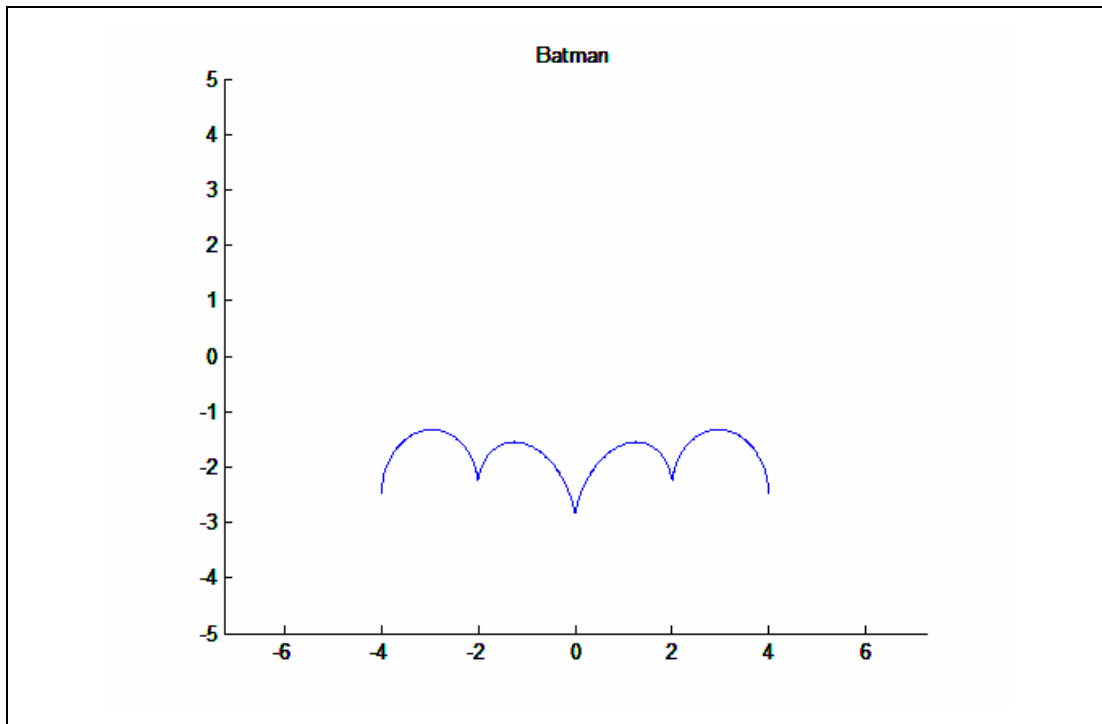
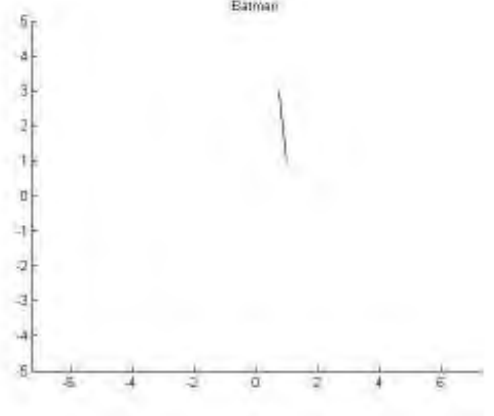
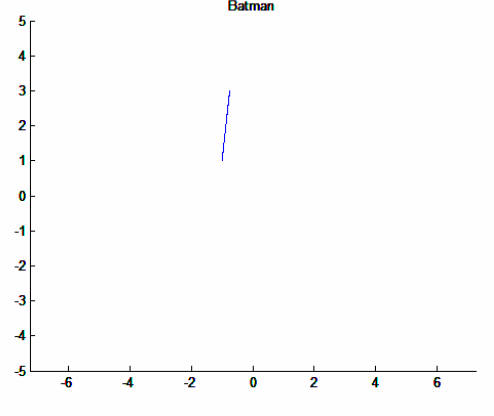


圖 12 程式碼

```
>>x2=linspace(4,-4);
y2=abs(x2./2)-((3*sqrt(33)-7)/112).*x2.^2-3+sqrt(1-(abs(abs(x2)-2)-1).^2);
```

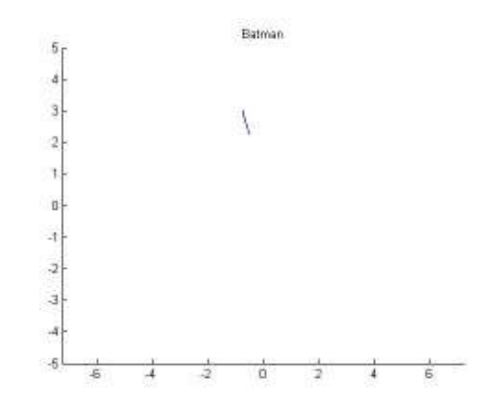
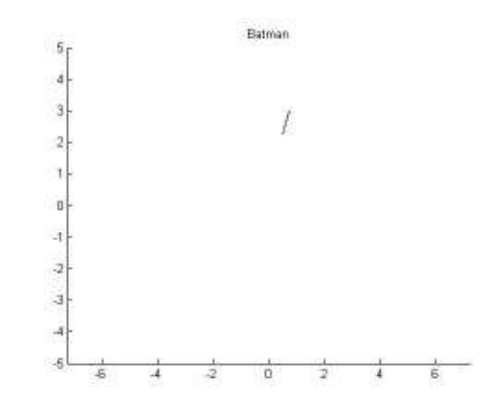
$$9 \sqrt{\frac{(|x| - 1)(|x| - 0.75)}{(1 - |x|)(|x| - 0.75)}} - 8|x| - y = 0$$

根式大於零，顯然可看出其為直線方程式。

	
<p>圖 13 程式碼</p>	<p>圖 14 程式碼</p>
<pre>>> x10=linspace(0.75,1); y10=9-8.*x10;</pre>	<pre>>> x6=linspace(-1,-0.75); y6=9+8.*x6;</pre>

$$3|x| + 0.75 \sqrt{\frac{(|x| - 0.75)(|x| - 0.5)}{(0.75 - |x|)(|x| - 0.5)}} - y = 0$$

明顯可看出其為兩條直線。

	
<p>圖 15 程式碼</p>	<p>圖 16 程式碼</p>
<pre>>> x7=linspace(-0.75,-0.5); y7=0.75-3.*x7;</pre>	<pre>>> x9=linspace(0.5,0.75); y9=0.75+3.*x9;</pre>

$$2.25 \sqrt{\frac{|(x-0.5)(x+0.5)|}{(0.5-x)(0.5+x)}} - y = 0$$

顯然是一條直線。

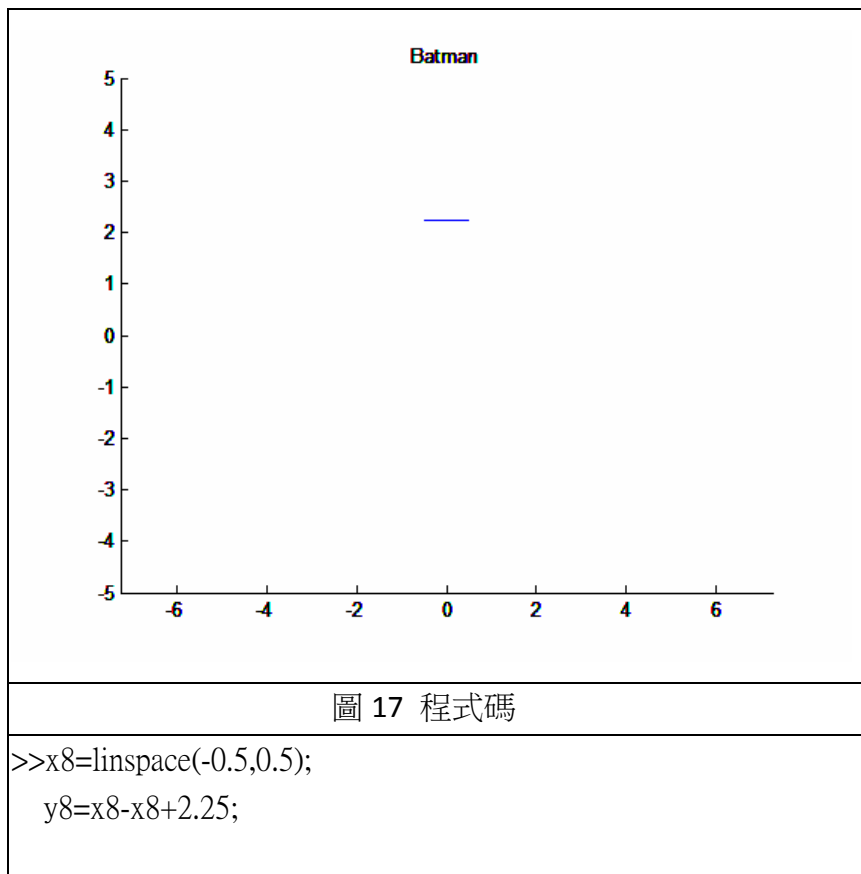
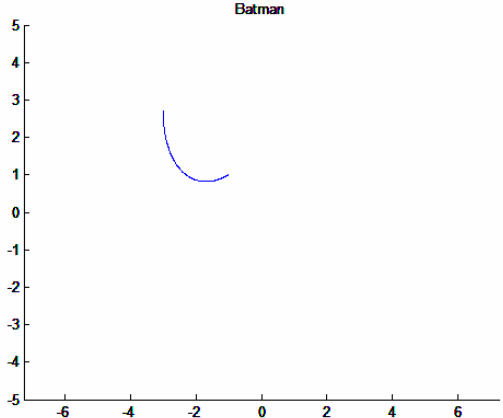
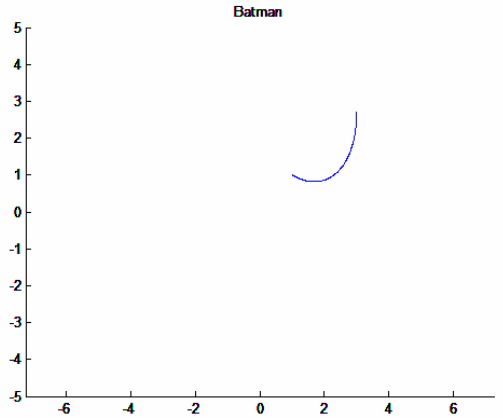


圖 17 程式碼

```
>>x8=linspace(-0.5,0.5);
y8=x8-x8+2.25;
```

$$\frac{6\sqrt{10}}{7} + (1.5 - 0.5|x|) \sqrt{\frac{||x|-1|}{|x|-1}} - \frac{6\sqrt{10}}{14} \sqrt{4 - (|x|-1)^2} - y = 0$$

根式內為零，可求出其為兩條曲線。

	
<p>圖 18 程式碼</p>	<p>圖 19 程式碼</p>
<pre>>> x5=linspace(-3,-1); y5=6*sqrt(10)/7+(1.5-.5*abs(x5)).*sqrt(a bs(abs(x5)-1)/(abs(x5)-1))-(6*sqrt(10)/14) *sqrt(4-(abs(x5)-1).^2);</pre>	<pre>>> x11=linspace(1,3); y11=6*sqrt(10)/7+(1.5-.5*abs(x11)).* sqrt(abs(abs(x11)-1)/(abs(x11)-1))-(6 *sqrt(10)/14)*sqrt(4-(abs(x11)-1).^2) ;</pre>

綜合上述所有結果，即可拼湊完成。

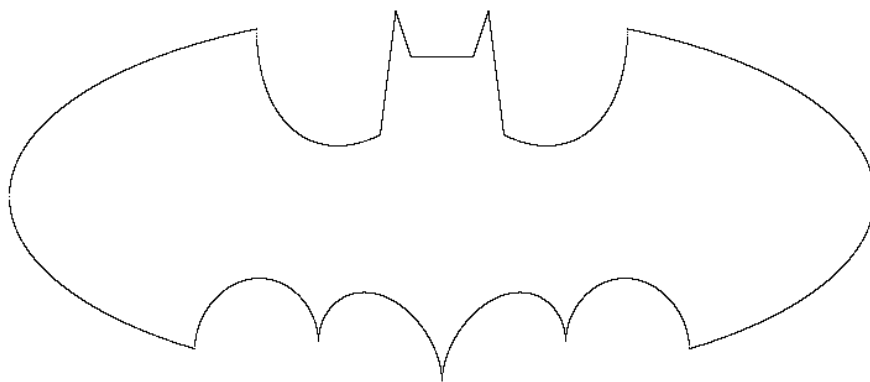


圖20 蝙蝠俠

由此可證，經由音效卡，可將數位信號轉換為類比信號輸出至示波器顯示。

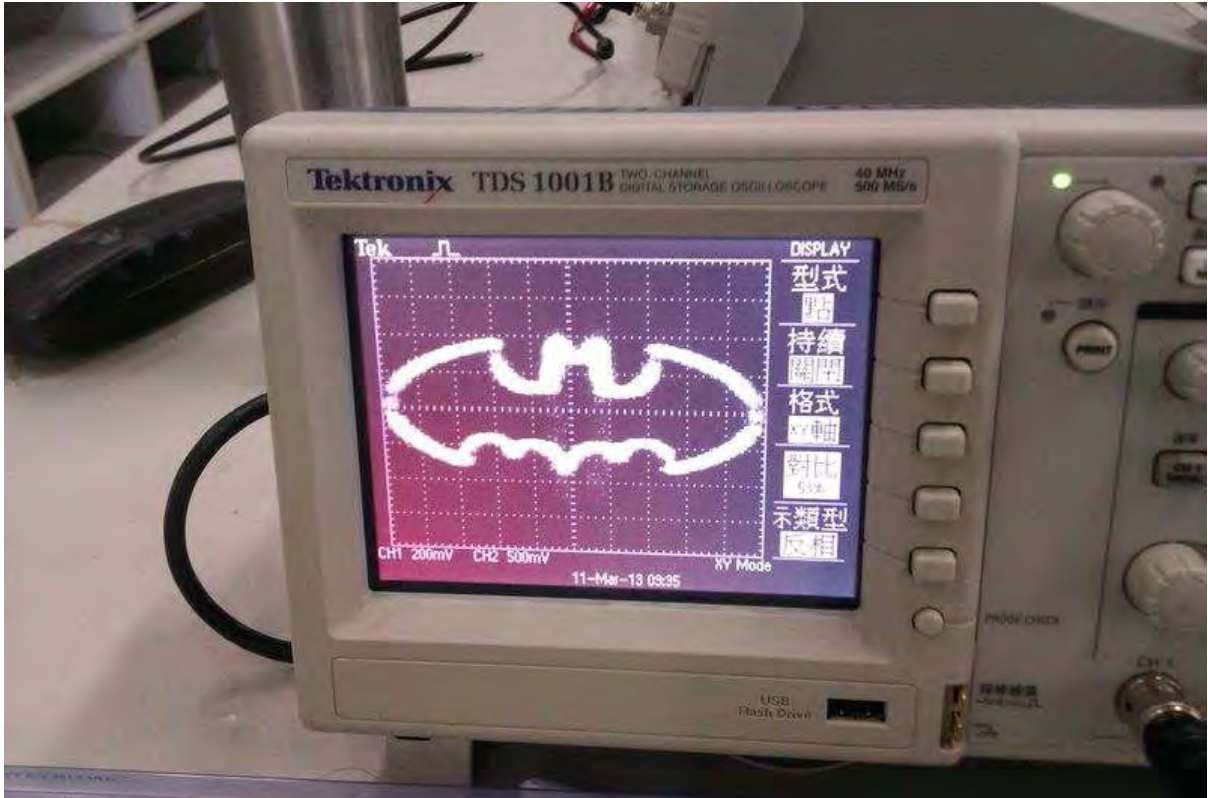


圖21 蝙蝠俠

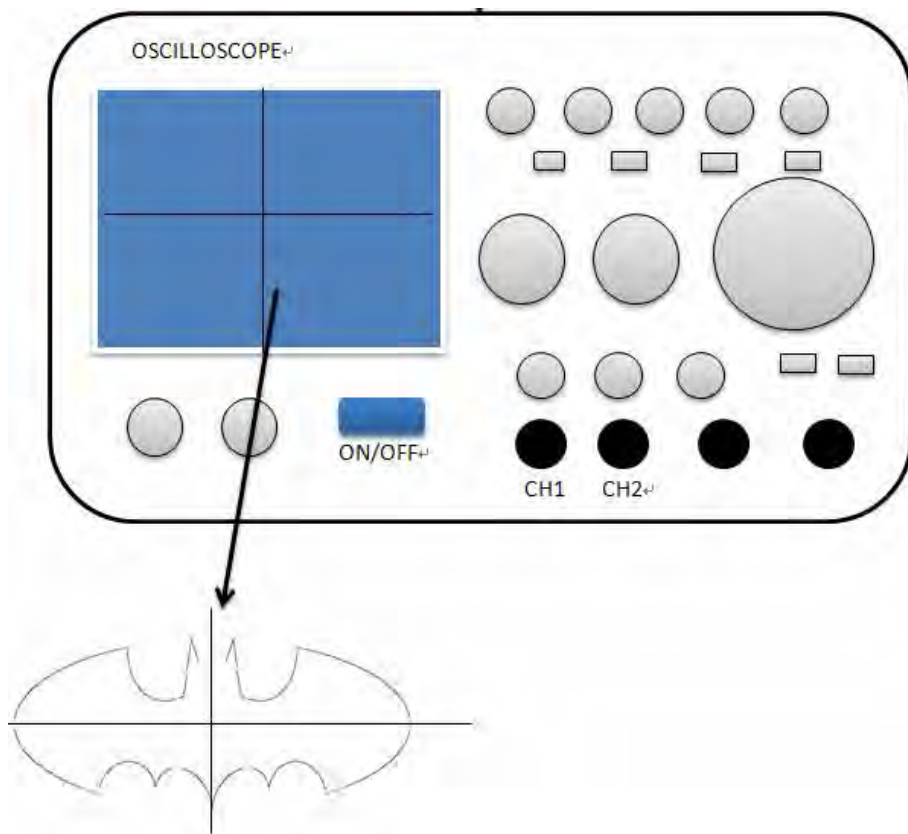


圖 22 示波器

四、裝置的聯接

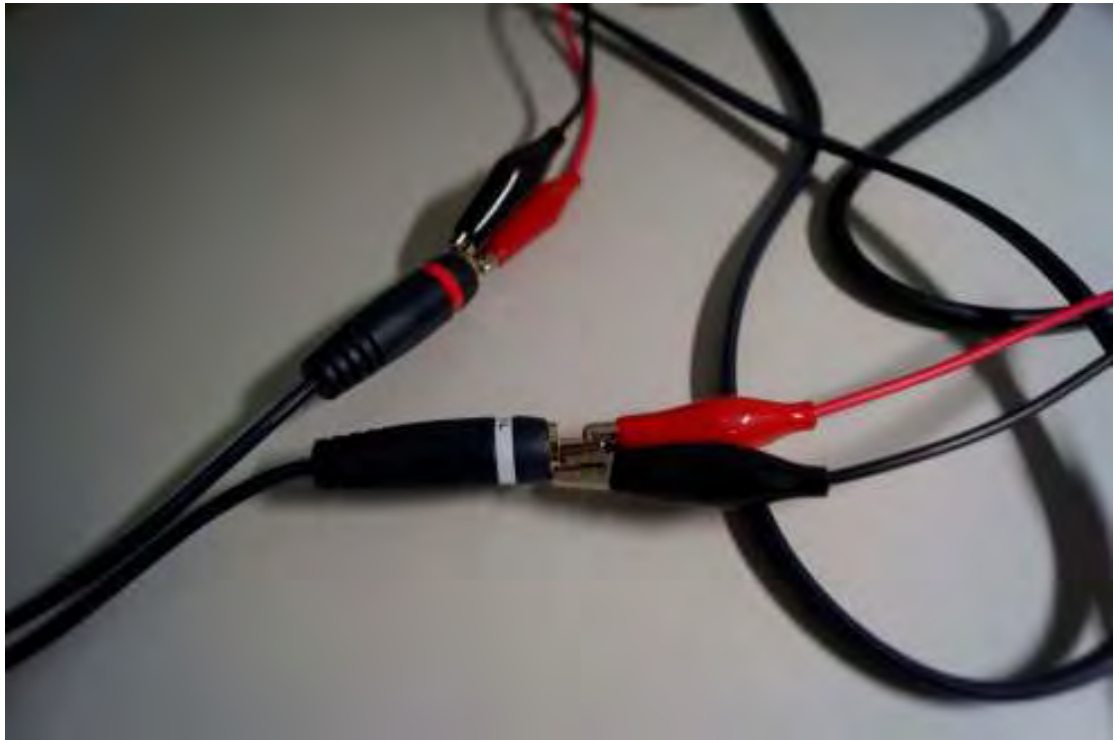


圖 23 裝置連接

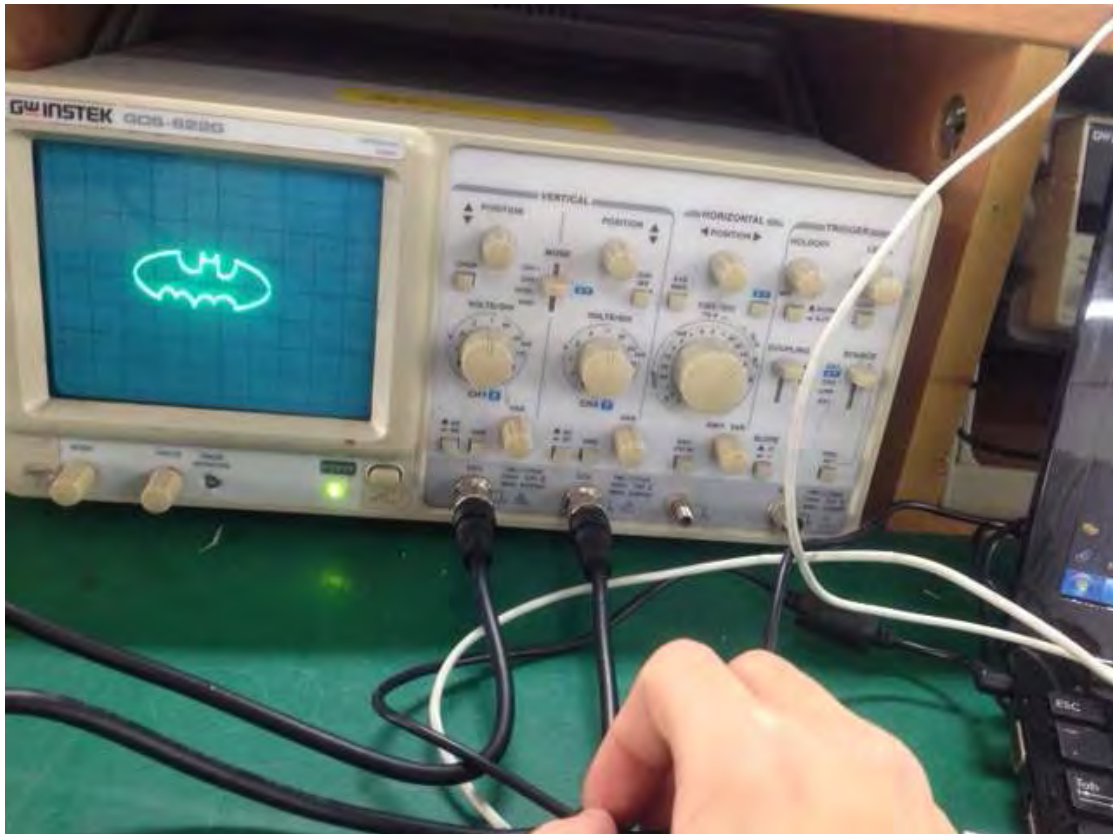


圖24 裝置連接

伍、研究結果

研究的結果針對輸出至示波器的音源檔進行播放與除錯，坐標軸轉換聲音檔與音效卡輸出的穩定，以及外在雜訊干擾，都大大影響了示波器上瀏覽圖形的畫面品質與穩定度。這讓示波器這項原本看似單調的顯示功能，有了另一種趣味的使用方法。

一、 示波器動畫

首先，我們以 Adobe Flash 製做一個單色影片，圖形以黑色填滿，其餘部分皆留白最後儲存為.avi 格式影片，其中一張張圖形，在快速且連續不斷的播放下產生視覺的暫留且有動畫的效果，將其輸入至 Matlab 中，並將影片一一拆解成一張張圖片，以邊緣函數 `edge(I,'prewitt')` 去分析提取每一張圖片的輪廓，可得圖形邊緣的點資料，圖 25,26 為經過邊緣函數 `edge()` 所處理過的對照圖。。

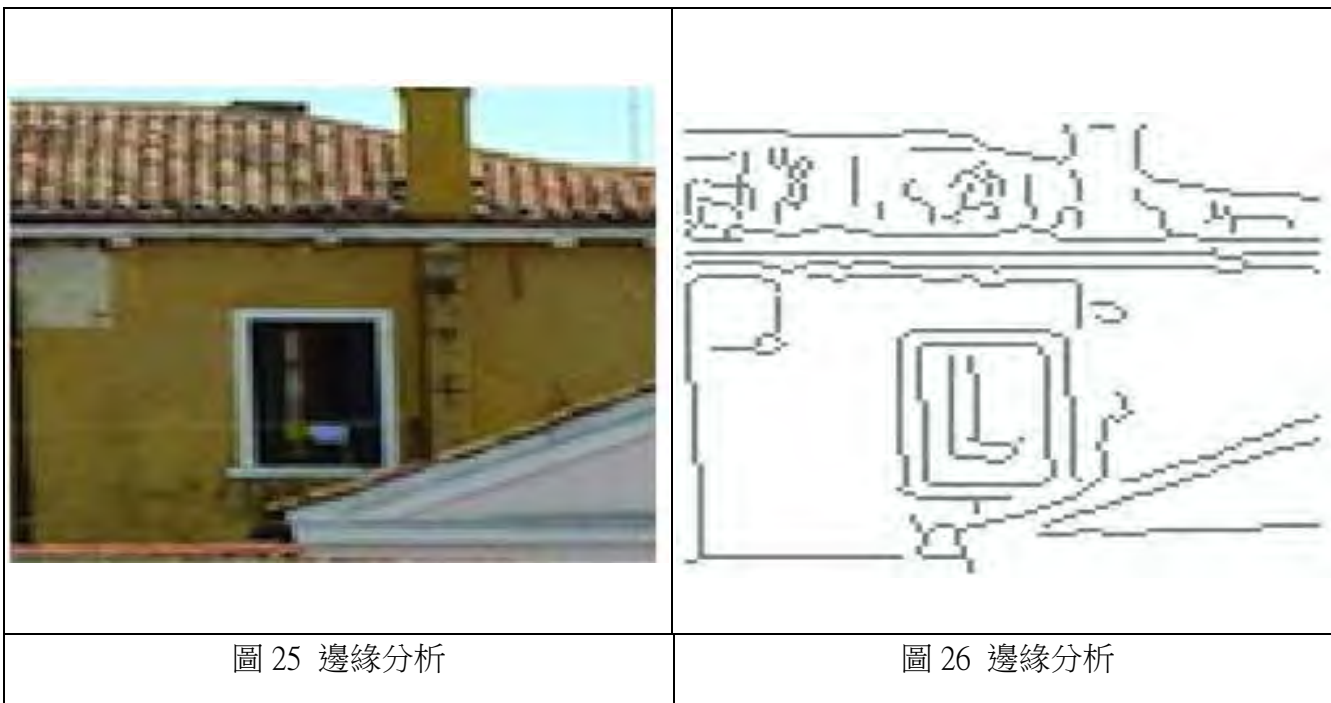


圖 28 殘圓

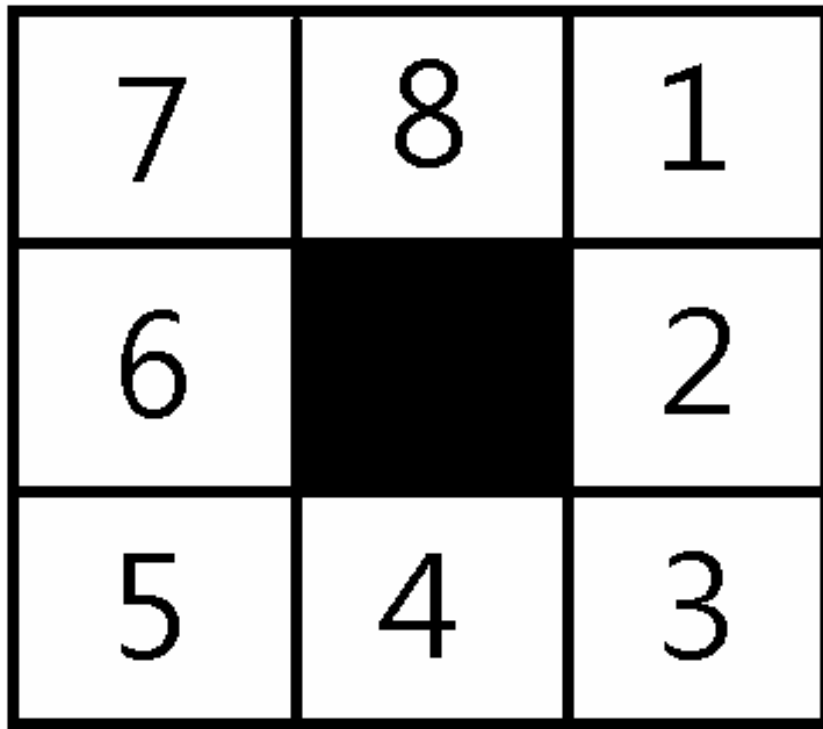


圖 29 九宮格

以下是完成的結果，如圖 30,31

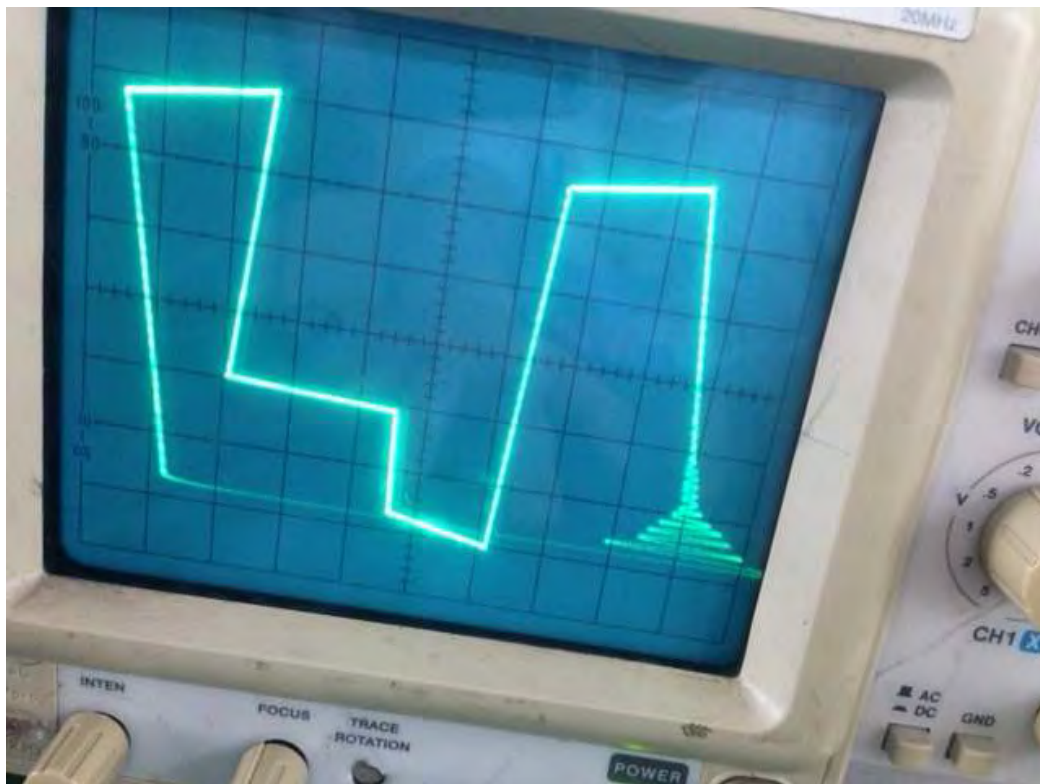


圖 30 大樓



圖 31 街舞

二、人機介面-----以電腦作為人機介面控制示波器圖形顯示

人機介面，可以更簡單表達思緒，一切只需經由雙手，將腦中的靈感經由滑鼠傳遞至電腦，連結示波器，呈現出不一樣的感覺。

以下是 Matlab 程式碼，藉此完成溝通。

```
function varargout = gui2(varargin)
% GUI2 MATLAB code for gui2.fig
%
% GUI2, by itself, creates a new GUI2 or raises the existing
% singleton*.
%
% H = GUI2 returns the handle to a new GUI2 or the handle to
% the existing singleton*.
%
% GUI2('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in GUI2.M with the given input arguments.
%
% GUI2('Property','Value',...) creates a new GUI2 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before gui2_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to gui2_OpeningFcn via varargin.
%
```

```

%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help gui2
% Last Modified by GUIDE v2.5 23-May-2013 01:14:51
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @gui2_OpeningFcn, ...
                  'gui_OutputFcn',  @gui2_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before gui2 is made visible.
function gui2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui2 (see VARARGIN)
% 宣告"滑鼠按下"的標誌變量 isPressed 為"假"
setappdata(hObject, 'isPressed', false);
% 宣告全域變數 sound
global sound
sound = [];
% 設定 axes1 的參數
set(handles.axes1, 'units', 'normalized');
title('Click and drag your mouse in this window!');
% Choose default command line output for gui2
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);
% UIWAIT makes gui2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = gui2_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on mouse press over figure background, over a disabled or
% --- inactive control, or over an axes background.
function figure1_WindowButtonDownFcn(hObject, eventdata, handles)
% hObject      handle to figure1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% 雙擊左鍵
if strcmp(get(gcf, 'selectiontype'), 'open')
    % 彈出顏色設置對話框
    col = uisetcolor(get(handles.axes1, 'colororder'), '選擇畫筆顏色');
    % 設定畫筆顏色
    set(handles.axes1, 'colororder', col);
    % 單擊左鍵
else
    % 更新"滑鼠按下"標誌變量為"真"
    setappdata(hObject, 'isPressed', true);
end
% --- Executes on mouse motion over figure - except title and menu.
function figure1_WindowButtonMotionFcn(hObject, eventdata, handles)
% hObject      handle to figure1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% 取得"滑鼠按下"標誌變量
isPressed = getappdata(hObject, 'isPressed');
% 取得當前座標
currPt = get(handles.axes1, 'currentpoint');
%若滑鼠"按下"狀態
if isPressed

```

```

% 宣告全域變數 sound
global sound
% 分配 currPt 於 x, y
x = currPt(1, 1);
y = currPt(1, 2);
% 限制 x, y 大小範圍不超過 axes1
if x>1,x=1;end
if x<-1,x=-1;end
if y>1,y=1;end
if y<-1,y=-1;end
%x, y 併入 sound 中
sound = [sound; x,y];
% 繪製曲線
line(x, y, 'marker', '.');
end
% --- Executes on mouse press over figure background, over a disabled or
% --- inactive control, or over an axes background.
function figure1_WindowButtonUpFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% 宣告全域變數 sound
global sound
sound
% 更新"滑鼠按下"標誌變量為"假"
setappdata(hObject, 'isPressed', false);
% --- Executes on button press in PushbuttonClear.
function PushbuttonClear_Callback(hObject, eventdata, handles)
% hObject    handle to PushbuttonClear (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%sound = getappdata(hObject, 'sound');
% 宣告全域變數 sound
global sound
% 清空 sound
sound = [];
clc;
%清除曲線
delete(findobj('type', 'line', 'parent', handles.axes1));
% --- Executes on button press in PushbuttonStart.

```

```
function PushbuttonStart_Callback(hObject, eventdata, handles)
% hObject    handle to PushbuttonStart (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% 宣告全域變數 sound
global sound
soundadd=sound;
%使 sound 長度為三秒
while(length(sound)<150000)
    sound=[sound;soundadd];
end
%播放
soundsc(sound,50000,16);
```

以下是此人機介面的主要畫面

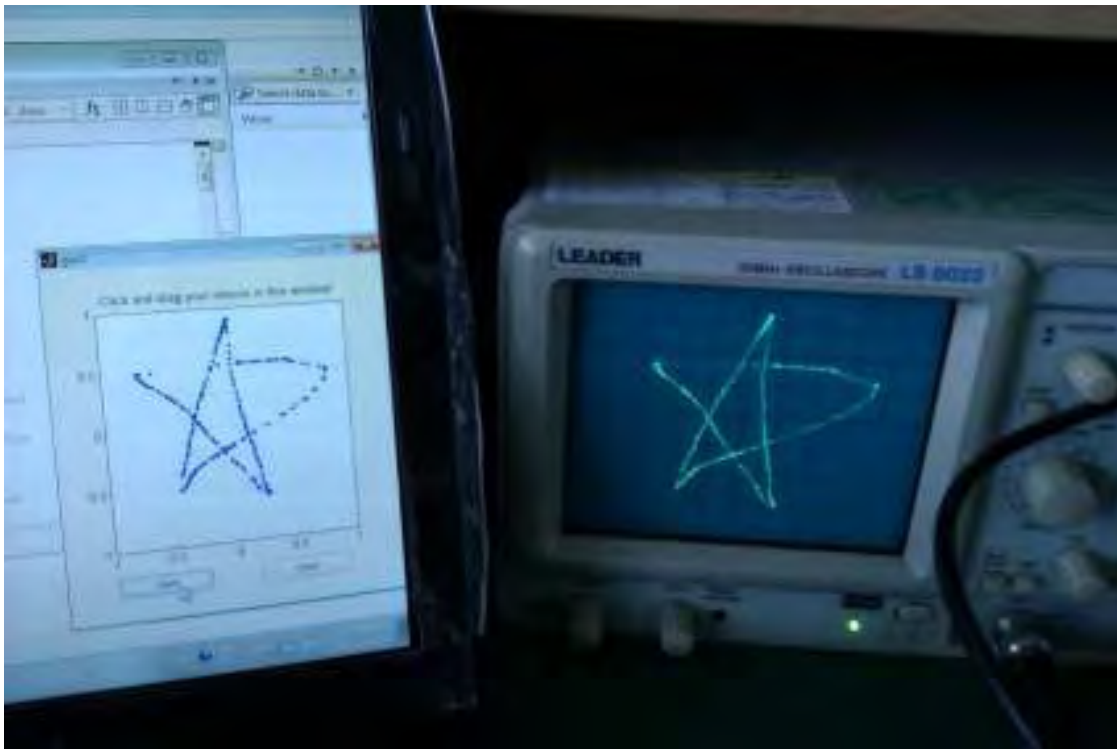


圖 32 人機介面

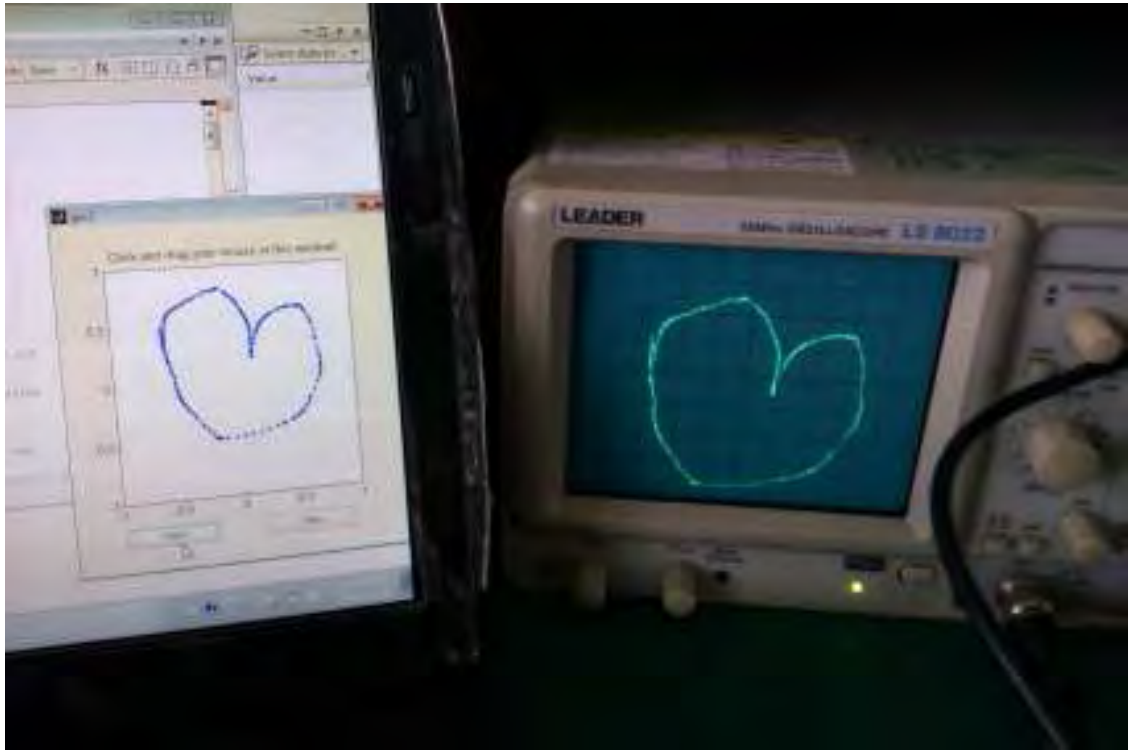


圖 33 人機介面



圖 34 人機介面

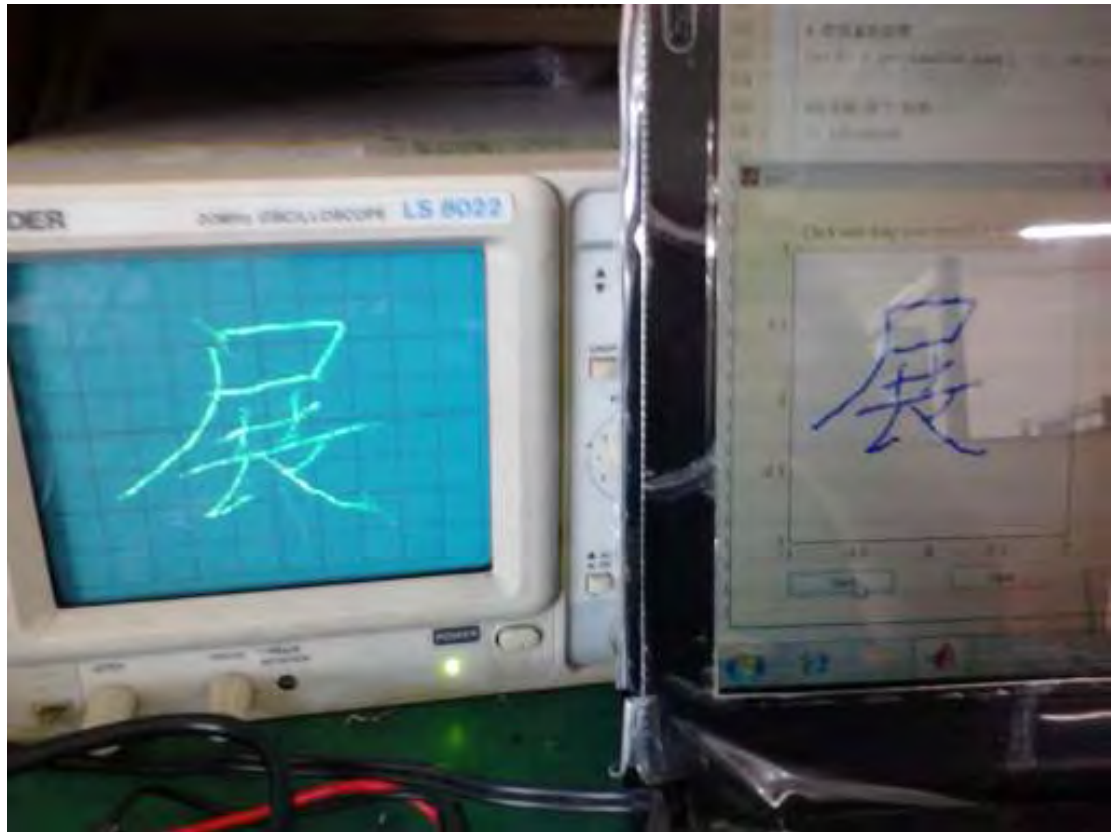


圖 35 人機介面

陸、討論

在製作並輸出的過程中，我們發現最容易遇到問題的地方就是座標描點轉換與音效卡輸出了，有時候輸出至示波器的圖形非常奇怪，甚至有大量的雜訊。在這個過程當中，我們發現，原來電腦裡的特殊音樂效果的程式，也大大的影響了輸出圖形，這也間接讓我們了解到這些程式，以及電腦內部的軟體互相影響會造成甚麼樣的結果，不管是許多電腦內建的杜比音效軟體，或是播放時所使用的程式，有時候，直接使用手機播放音效反而比電腦的雜訊要來的少，也更清晰呢!

這次的製作融合了我們在專業領域上所使用到的重要工具以及專業知識，還使用了許多專業的軟體，做到一半也常發現問題，這些問題一開始根本摸不著頭緒如何去解決它，但在經驗的累積下，也開始熟能生巧，不僅越做越順利，遇上問題也能夠很快地去解決它，我也了解到，事前若不完整的規劃，等到遇上問題時，就越需要花時間去解決它，還好有著老師與同學的幫助，才得以完成。

柒、結論

我們現在在做的，也就是最早電視的雛型吧!在陰極射線管上施加電壓，使得在螢光幕上快速運行的電子產生偏移，其實就是我們的重點，透過現代化的數學軟體的輔助，不必麻煩的去製作 IC 與電路板來輔助，使用函數圖形來繪點只是簡單的第一步，往後使用軟體直接轉換，影片輸出，讓更多人了解到示波器並不只是光看波形、電壓大小或是週期而已，能讓更多第一次接觸此儀器的人能夠更快速並清楚的了解到它的原理，才去懂得如何去使用，常看到許多人，使用一些特殊的工具，雖然會使用他，但並不是真正的了解它的原理是如何，為什麼它能這麼輕鬆簡單的完成這一些事，示波器對初次接觸電學的人來說是必備的工具，但也對很多人來說是相當的複雜，控制面板上許許多多的按鈕與旋鈕令我第一次接觸時也覺得眼花撩亂，希望這次所製作的示波器動畫能讓更多人快速了解到示波器之原理及其使用方法。

這一次所使用到的特殊工具為電機電子群都常接觸到的示波器，以及電腦，雖然看似簡單，但其實越不簡單，一開始要找可以符合我們所需要的轉譯軟體，找了好久，終於找到這套 Matlab 軟體，就是努力的開始了，雖然這套軟體有著 C 語言的基礎，但是要完整製作出我們所需要使用的程式碼還是費了我們一番心思，同時也學到了團隊合作的重要性，大家分工合作完成，令我們覺得相當的值得。

捌、參考資料及其他

晶片應用 102 期論-李沙育圖形法

<http://ic102.xtreemhost.com/redirect.php?tid=252&goto=lastpost>

附錄、示訊(波)器的介紹及應用

<http://www.scu.edu.tw/physics/science-scu/M302/20.htm>

示波器的介紹及使用

<http://cc.feu.edu.tw/~jeffrey/mech-elec/OSCILLOSCOPE.pdf>

示波器狀態顯示解說&使用調整方法說明

<http://sun.cis.scu.edu.tw/~lab/tools/part4.htm>

數位邏輯自編教材/數位邏輯 ch1

<http://163.32.98.35/eefile/%E6%95%B8%E4%BD%8D%E9%82%8F%E8%BC%AF%E8%87%AA%E7%B7%A8%E6%95%99%E6%9D%90/%E6%95%B8%E4%BD%8D%E9%82%8F%E8%BC%AFch1.pdf>

維基百科-MATLAB

<http://zh.wikipedia.org/wiki/MATLAB>

維基百科-數位類比轉換器

<http://zh.wikipedia.org/wiki/%E6%95%B8%E4%BD%8D%E9%A1%9E%E6%AF%94%E8%BD%89%E6%8F%9B%E5%99%A8#.E5.BA.94.E7.94.A8>

MATLAB 程式設計《入門篇》

<http://mirilab.org/jang/books/matlabProgramming4beginner/slide/>

MATLAB 之工程應用-指令 Linspace 之使用

<http://bime-matlab.blogspot.tw/2006/09/23-linspace.html>

百度百科-linspace

<http://baike.baidu.com/view/3612582.htm>

邊緣分析

<http://www.vlfeat.org/overview/imdisttf.html>

【評語】 091006

1. 示波器是一種電子零件與系統之量測儀器，本組作品居然可以應用於動畫上，增加示波器使用測量功能，是一件創意性極佳的作品。
2. 作品利用 Matlab 軟體應用，加上演算法開發，資料視覺化，資料分析及數值計算的高階語言來組成一件完整的作品。
3. 本組作品偏重軟體語言程式設計，至於硬體設備則較為缺乏，導致系統功能受到限制。
4. 本組作品系統原理與功能可以類比近來較夯的 3D 列印技術。
5. 學生操作與解說表現極為優良。