中華民國第 54 屆中小學科學展覽會

作品說明書

高中組　生活與應用科學科

佳作

040802

百元語音辨識系統

學校名稱：國立竹北高級中學

作者：	指導老師：
高二 賴詩雨	聶智慧

關鍵詞：語音辨識、演算法、微處理器

作品名稱：百元語音辨識系統

摘要：

探討在低成本下，完成非特定人語音辨識的可行性，讓一般廉價的小家電，擁有精準的語音辨識能力。

爲使成本降低，必須要設計一個 ROM、RAM 使用很少、反應速度很快、辨識能力高的語音辨識系統，整理作品重點如下：

一． **音節簡化**：發音去掉聲音調後，再將同類聲音進行整併，整理出 133 個音節➔資料量是繁體 13060 字數的 1 / 100

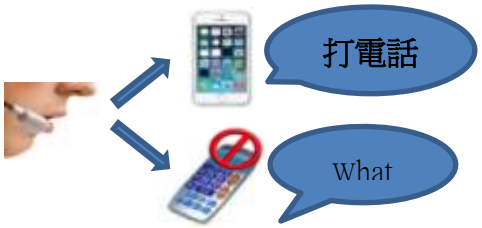
二． **資料庫化簡**：使用整數的指數取代符點運算➔資料量爲傳統的語音特徵的 1 / 20

三． **語音辨識演算法**：使用梅爾倒頻譜係數、隱藏馬可夫模型

四． **處理器**：用 16 位元的整數運算，可在低價的處理器上執行

五． **訓練用的語音**：向中華民國計算語言學學會（MAT）購買純語音的資料庫

完成麵包板上的百元語音辨識系統：

現況	百元語音辨識系統例子
	

壹、 研究動機

自從 iPhone 及 Android 提供手機上的語音辨識功能後，讓大眾驚豔於語音辨識的方便性，甚至在不久前 Panasonic 的電視廣告中，還推出了語音辨識頻道的電視機。

由於微處理器的運算能增強，加上我對於語音辨識的興趣，有一天上電腦課後想到：一般的家電爲何不能使用語音來控制？這中間一定藏有很多不爲人知的秘密。

經過觀察後發現：數千元以上的高價設備，已逐漸將語音辨識，變成主要的人機界面之一，但是對於一般廉價（千元以下）的家用產品，卻無語音辨識功能，追究原因：語音辨識須要計算複雜的數學公式，導致硬體的成本很高，加上辨識率問題，這是一般家電上無法使用語音辨識技術的主要門坎。

所以如果能改進語音辨識的演算法，建構一個低運算量、使用 ROM、RAM 都很少的語音辨識系統，就能降低產品的成本，加上辨識率精準、直接可以使用，也許可以克服此一問題。

貳、 研究目的

我的研究目的，是探討在低成本下，完成非特定人的語音辨識可行性：

1	成本	百元以內
2	辨識命令	小量詞句（50 詞以內）
3	事前語音訓練	不用（非特定人）
4	使用方式	插上電源即可使用
5	辨識率	精準（80% 以上）

我們可以想像一個 200 元的電視遙控器，相較一個 250 元，但具備語音辨識的電視遙控器，哪一個對消費者有吸引力？這類的系統，可以協助我們擴展家電產品的新穎性，讓週遭的小家電（例如：遙控器、 電燈、冰箱、 … ），具有簡易的語音辨識能力，讓產品更具方便性。

低成本的語音辨識系統，有很多實際上的商業應用：

例一：對電視遙控器說“大愛電視台”，即可自動轉到特定電視台，不用再記頻道號碼

例二：躺在床上看完書後說“我要睡了”，即可將臥房的主燈切換成夜燈

例三：可以和小朋友互動的洋娃娃，並進行簡單的對話

參、 研究設備及器材

一. 硬體：

1	筆電	
2	麵包板、微處理器及麥克風等電子零件若干	一般電子材料行購買
3	燒錄程式工具 PICKit3	microchipDIRECT 網站購買
4	PIC32 / PIC24 開發板	

二. 軟體：

1	PC 上的編譯器 (Compiler) Borland C++ Builder 5.0 （BCB5）
2	微處理器（Microchip）的開發環境：MPLAB IDE, MPLAB C

二. 開發流程：

第一步：在 PC 上完成演算法的開發
第二步：再轉移到 PIC32 / PIC24 開發板上驗証
第三步：在麵包板上組合出精簡的電路

❶ 發展語音辨識

❷ 在開發板驗証

❸ 百元語音辨識系統

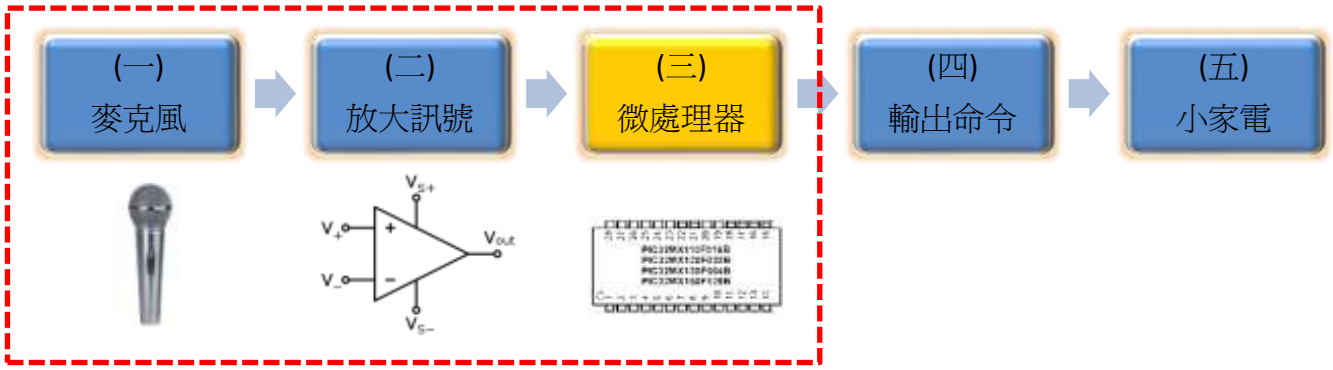
肆、 研究過程及方法

一、 語音辨識之技術分類：紅字爲作品規格

	分類	項目
1	**語音特性**	(1) 說話方式、口音 (2) 變化情緒、壓力 (3) 汽車、電話等噪音 (4) 回音、發音人的距離等
2	**詞彙大小**	(1) 小字彙（小於 100 詞）← 作品規格 (2) 中字彙（100 - 1000 詞） (3) 大字彙（大於 1000 詞）
3	**系統訓練**	(1) 特定人：使用前需訓練 (2) 非特定人：任意詞且不需訓練 ← 作品規格
4	**發音方式**	(1) 單字音 (2) 句子 ← 作品規格 (3) 連續語音
5	**辨識技術**	(1) 語音模版 (2) 動態時間校準 (3) 隱藏馬可夫模型（簡稱 HMM ） ← 作品規格 (4) 類神經網路
6	**語音特徵**	(1) 梅爾倒頻譜係數（簡稱 MFCC）← 作品規格 (2) 線性預估係數（簡稱 LPC）

二、 語音辨識系統硬體簡介：

圖表 1 爲硬體簡圖，其中紅色虛線部份爲作品：



圖表 1

(一) **麥克風**：聲音是以聲波的形式來傳遞，這種波的訊號稱爲類比訊號（Analog signal），透過麥克風的壓電轉換，可以將聲波的振幅，轉爲相對應的電壓訊號。

(二) **訊號放大**：由於麥克風輸出的訊號很微小，所以一般會使用運算放大器，將訊號加以放大、並使用低通濾波器，過濾掉 4000Hz 以上的頻率。

(三) **微處理器**：

1. AD（類比轉數位）轉換：將聲波的訊號轉爲對應的電壓值，就是所謂的 AD 轉換，轉換的速度必須夠快才行，一般人的聲音頻率在 4000Hz 以下，故依據尼奎斯特定理（Nyquist-Shannon），每秒至少要做 8000 次的 AD 轉換，才能對語音做正確的取樣。

2. 振幅正規化：使用演算法自動調整強弱不同的音量，不會導致音量忽大忽小，此功能一般是由硬體完成，稱爲自動增益控制（Auto Gain Control, AGC），作品中爲了節省成本，使用軟體模擬，但僅能達成部份 AGC 的效果。

3. 辨認語音：使用梅爾倒頻譜係數（MFCC），做爲語音的特徵參數，運用隱藏馬可夫模型（簡稱 HMM ）和資料庫模板進行比對。

其中 HMM 技術廣用於各種知名的語音辨識系統，例如 Apple 的 SIRI，Google 的 Voice Search 等，是已知最適合語音辨識的演算法。

(四) **輸出命令**： 輸出辨識結果，控制其他的電器。

(五) 小家電：如電燈、對話娃娃等。

三．參考上頁的（三）微處理器，在微處理器中包含的主要軟體系統簡：

圖表 2，辨識系統的軟體簡圖：其中有 獨 者表示作品的獨創技術，以分辨公知技術。

圖表 2

(一) **定時中斷**：每秒產生 8000 次的中斷，每次中斷，會進行一次 AD 轉換。

(二) **AD 轉換**：將類比訊號轉爲數位的過程，稱之爲**採樣**，一般來說**採樣的頻率**及**採樣的解析度**決定了聲音採樣的品質，採樣的解析度一般以位元數表示，位元數越大，所能記錄聲音的種類越多越細膩，就越接近原來的聲音。

	採樣頻率	採樣解析度	說明
語音	8KHz	16 位元	降低採樣解析度可以節省硬體成本
DVD	96KHz	24 位元	
作品	8KHz	10 位元	

獨 (三) **端點偵測**：端點偵測可以分析連續的聲音中，真正屬於語音的**起始點**及**結束點**，正確的判斷這兩個端點，有助於降低資料量。

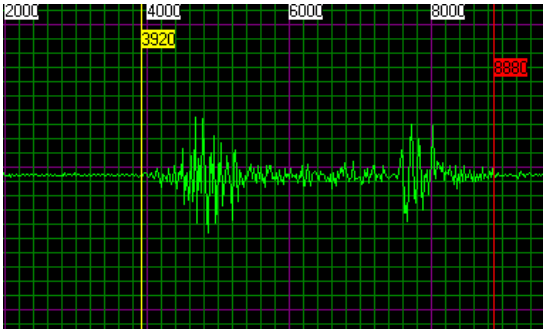
(四) **MFCC**：計算梅爾倒頻譜係數（MFCC），此係數因爲考慮到人耳對於不同頻率的感受程度，故特別適合用在語音辨識。

獨 (五) **比對資料庫**：運用隱藏馬可夫模型（簡稱 HMM ），將上述的 MFCC 係數和資料庫的模板比較，計算兩者相似的機率大小，決定辨識的結果。
（作品中的可處理溢位的隱藏馬可夫模型，讓系統可在 16 位元處理器下執行）

獨 (六) **資料庫模板**：由向 MAT 購買的純語音資料庫，計算出純語音對應的 MFCC 係數，再反覆經由維特比演算法（Viterbi）產生出來。
（作品中將語音資料量縮小 1 / 100、特徵縮小爲 1 / 20，共計縮小 1 / 2000）

四．參考上頁的（三）端點偵測，說明判斷語音端點的兩種演算法：

圖表 3：此演算法用來判別語音的起點（3920，黃線）及結束點（8880，紅線）。



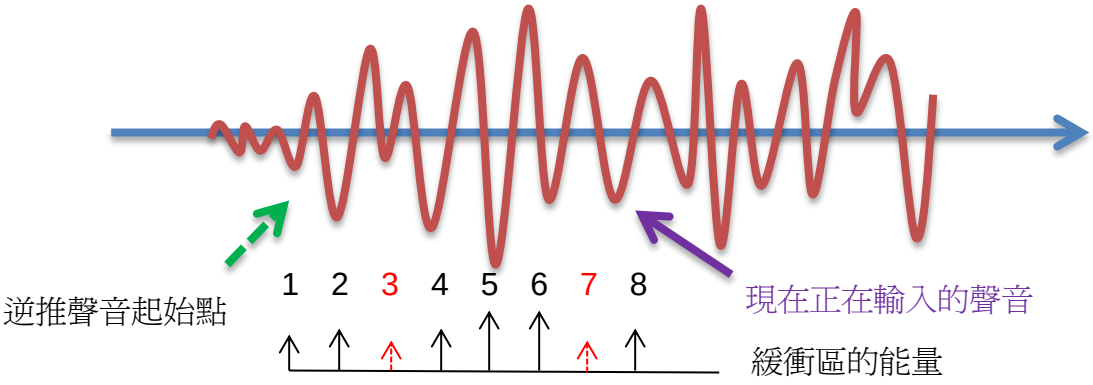
圖表 3

語音端點偵測的正確與否，會大幅影響辨識率，一般僅依靠過零率及粗略的聲音能量，常會發生錯誤的判斷，因此我設計了兩個方法，可以很精準的決定語音的起始點及結束點。

(一) **使用緩衝區偵測起點**：請參考圖表 4，經過觀察後發現：聲音由無聲變爲有聲的過程，不能依據單一點的能量來判斷，須要有一段觀察期間才能判斷準確，所以設計了緩衝區來做爲判斷的基準 ，方法如下：

1. 方法：當緩衝區中含有足夠多的大聲音能量，即可判斷爲聲音的起點。
2. 假設：有一可儲存最近 8 組聲音能量的緩衝區，其中至少要有 6 組聲音能量較大時，即是語音的起點。
3. 例子：聲音由 1、2、... 8、9、... 持續地輸入緩衝區，直到到最近的 8 組緩衝區中含有 6 組能量大的聲音時（下圖中的黑色），才決定了語音的開始點。

由圖表 4 可知，當第 8 組聲音輸後，我們才能確定第 1 組聲音是起點，這時會將語音的起點，逆推回到第 1 組的時間點，並且正式地由 1 組開始進行語音辨識。



圖表 4：黑色：能量較大，紅色：能量較小

(二) **狀態圖**：請參考圖表 5，有了緩衝區後，即可依緩衝區內能量的情況，決定語音現在的狀態分辨了不同的語音狀態，即可以對辨識系統進行相應的校調。

1. 方法：

(1) 定義：無聲、有聲、小聲、靜音、結束：5 種狀態。

(2) 定義 **A**、**B**、**C**、**D**、**E**、**F** 六種〔**轉移情境**〕，每一種狀態在符合規定的情境時，可以轉移到下個狀態。

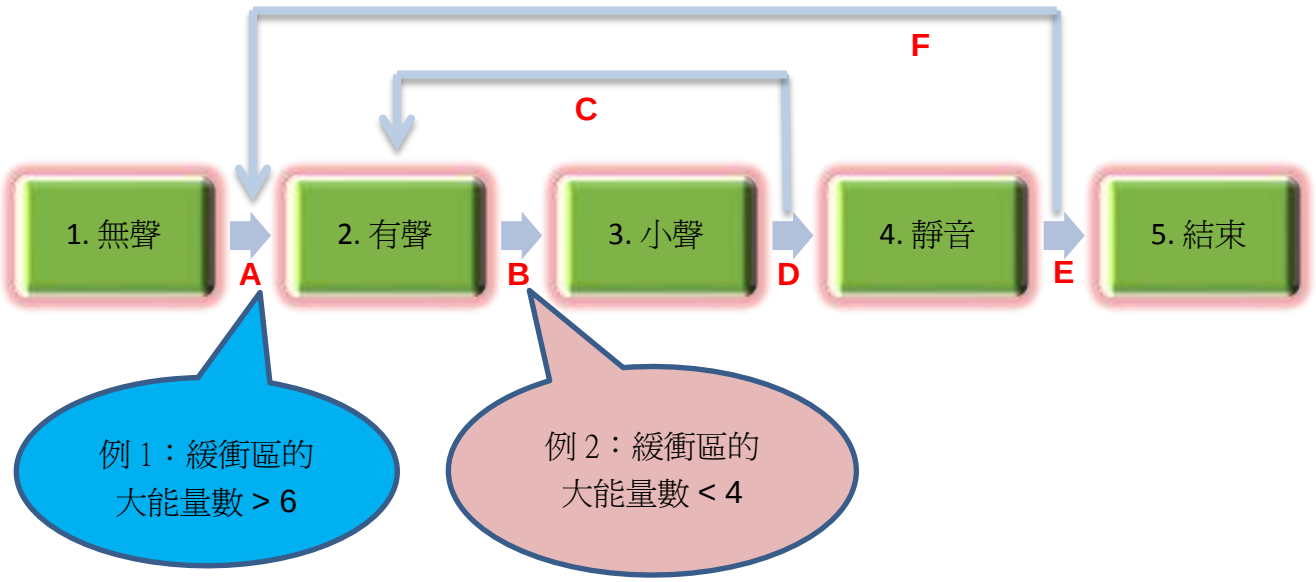
例如：無聲 ➔ 〔**情境** A〕 ➔ 有聲

而小聲可能是：小聲 ➔ 〔**情境** D〕 ➔ 靜音

或：小聲 ➔ 〔**情境** C〕 ➔ 有聲

2. 例子 1：上頁 (一) 即爲 1. 無聲 ➔ 2. 有聲，而轉移的〔**情境** A〕即爲 (一) 的方法。

3. 例子 2：我們可以仿(一)的方式，定義〔**情境** B〕，例如：緩衝區中的聲音能量較大的組數小於 4 時，即可由有聲狀態 ➔ 轉換爲小聲狀態。



圖表 5

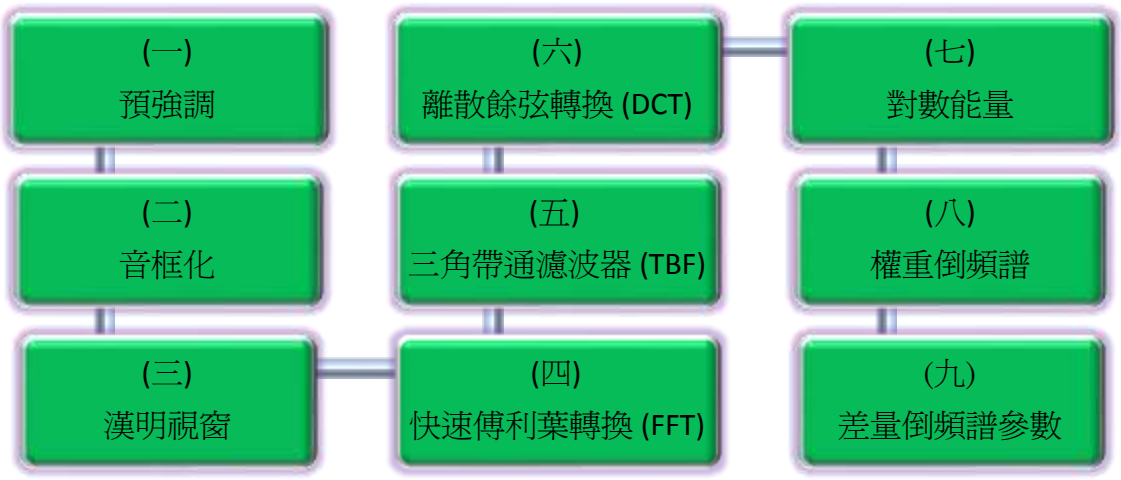
由於在不同的環境下（例如：吵雜的環境，或安靜的房間），緩衝區的大小及其對應的轉移條件，可能會有所不同。

下表爲使用上述方法在 MAT 純語音資料庫的實驗結果：

總句子字數	聲音端分析正確句子字數	百分比
92,726	89,761	96.80%

五・MFCC：

考圖表 6，梅爾倒頻譜系係數方法（這是一個公知的演算法，但實作上則有很大的差異）。



圖表 6

(一) 預強調（Preemphasize）：目的就是爲了消除發聲過程中聲帶和嘴唇的效應，來補償語音信號受到發音系統所壓抑的高頻部分。

(二) 音框化：先將 200 個取樣點集合成一個觀測單位，稱爲音框（Frame），涵蓋的時間約爲 25 ms 左右，每個音框之間重疊 120 個取樣點。

(三) 漢明視窗（Hamming Window）：將每一個音框乘上漢明視窗，以增加音框左端和右端的連續性。

(四) 快速傅利葉轉換（Fast Fourier Transform, 簡稱 FFT）：由於訊號在時域（Time domain）上的變化通常很難看出訊號的特性，所以通常將它轉換成頻域（Frequency domain）上的能量分佈來觀察，不同的能量分佈，就能代表不同語音的特性。

(五) 三角帶通濾波器（Triangular Bandpass Filters）：將頻譜能量乘以 22 個三角帶通濾波器，求得每一個濾波器輸出的對數能量（Log Energy）。
上述的三角帶通濾波器，是依梅爾頻率建立。
梅爾頻率代表一般人耳對於頻率的感受度，在低頻部分，人耳感受是比較敏銳 在高頻部分，人耳的感受就會越來越粗糙 。

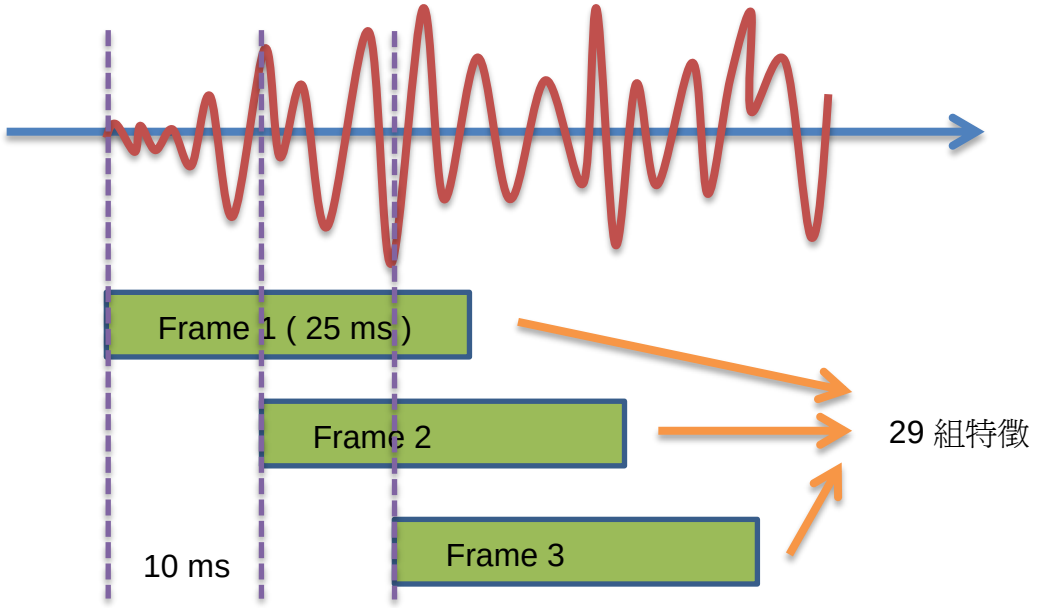
(六) 離散餘弦轉換（Discrete cosine transform, 簡稱 DCT）：將上述的 22 個對數能量代入離散餘弦轉換公式，求出 12 階的倒頻譜參數。

(七) 對數能量（Log energy）：加上一個音框的音量（即能量），使得每一個音框基本的語音特徵就有 13 維，包含了 1 個對數能量和 12 階的倒頻譜參數。

(八) 權重倒頻譜（lifters Weighted Cepstrum）：爲使訊號更平順，在頻率域有濾波器（Filter），而對應於倒頻域的則是（Lifter），使用權重可以讓每一個特徵的重要性不同。

(九) 差量倒頻譜參數（Delta cepstrum）：差量倒頻譜參數，顯示倒頻譜參數對時間的變化它的意義爲倒頻譜參數相對於時間的斜率，也就是代表倒頻譜參數在時間上的動態變化，最後取出 29 組特徵。

請參考圖表 7，經由 MFCC 取出 29 組特徵的時序圖：



圖表 7

六・漢語拼音的音節簡化：

表示文字的方法有很多種：本作品中以注音爲基礎再對應到漢語拼音。

	表示法	說明
1	繁體	台灣 13060 字
3	注音	音標有 1300 個
4	漢語拼音	412 個音 ← 作品
5	通用拼音	MAT 純語音資料庫

爲使資料庫能夠變小，必須忽略一些聲調上的細節，我採用了三個方法因應：

(一) **去掉聲調**：例如：竹（ㄓㄨˊ）➔ ㄓㄨ，如此就可以由 1300 個注音，降低爲 417 個去聲調的注音。

(二) **整併近似音**：例如：將 ㄚ、ㄠ、ㄢ、ㄤ ➔ 整合爲同一個聲音，但是缺點就是系統會分不出 ㄅㄚ、ㄅㄠ、 ㄅㄢ 、ㄅ ㄤ的區別，下表爲近似的注音的例子。

1	ㄚ、ㄠ、ㄢ、ㄤ
2	ㄜ、ㄟ、ㄣ、ㄥ
3	ㄛ、ㄡ
4	ㄓ、ㄔ、ㄕ、ㄖ、ㄗ、ㄘ、ㄙ
5	ㄣ、ㄥ
6	（ㄚ ㄢ ㄤ ㄠ ㄣ ㄥ ㄡ）、一（ㄚ ㄢ ㄤ ㄠ ㄣ ㄥ ㄡ）
…	

(三) **將注音分爲前音節及後音節**：例如：篡（ㄘㄨㄢ）➔ 分爲前音節 ㄘㄨ++ 及後音節 ++ㄨㄢ（++表示會連接另一個音節）。

這樣分割的好處是相同的音節可以重覆利用，進一步減少資料庫的數量。

因爲相同的音節可以共用，最後整合出 99 個前音節，34 個後音節，共 133 種音節，下表爲各種碼的對應關係的簡表，其中 51、52、53 共用相同的前音節：

	漢語拼音	通用拼音	注音去音標	中文字	前音節++	++後音節
0	A	a	ㄚ	阿	ㄚ++	++ㄚ
1	Ai	ai	ㄞ	艾	ㄞ++	++ㄞ
…						
51	cuan	cuan	ㄘㄨㄢ	篡	ㄘㄨ++	++ㄨㄢ
52	cui	cuei	ㄘㄨㄟ	脆	ㄘㄨ++	++ㄨㄟ
53	cun	cun	ㄘㄨㄣ	寸	ㄘㄨ++	++ㄨㄣ
…						

共用相同前音節

七．對音節進行細切：

一個簡單的注音發音，對於辨認系統來說，仍然是很大的聲音單位，所以必須進行細切，才能將這個發音的特性表現出來，並依據前音節及後音節的關係，將每個發音切爲六等份，前音節兩個等份，後音節四個等份，如此，才可以辨識細膩的聲音變化。

訊（ㄒㄩㄣ）					
前音節（ㄒ++）		後音節（++ㄩㄣ）			
ㄒ_0	ㄒ_1	ㄩㄣ_0	ㄩㄣ_1	ㄩㄣ_2	ㄩㄣ_3

經由上述的細切之後，就是實際的資料庫，所以 "訊" 這個聲音，包含有 6 筆資料庫，分別是：（ㄒ_0、ㄒ_1、ㄩㄣ_0、ㄩㄣ_1、ㄩㄣ_2、ㄩㄣ_3）。

八・資料庫的產生：

利用 MAT 純語音資料庫來產生 133 個音節的資料，而 MAT 的資料庫中，因爲採用了通用拼音，所以必須先轉成注音或漢語拼音後才可以使用。

資料庫初始化的方法如下：

(一) 對每一個發音計算 MFCC 係數。

(二) 因爲我們不知道一個音如何分割成前音節 2、後音節 4（ 6 等分），所以先對每一個發音進行 6 等份的**均切**。

(三) 取均切後的平均值，做爲第一版的資料庫。

重覆下面（四）（五）（六）步驟，用舊的資料庫，產生新的資料庫，直到收斂爲止：

(四) 由多變數的高斯機率密度函數去計算出 6 等分（前音節 2 等份加上後音節 4 等分）和資料庫的相似機率值：

$$G_i(X_t) = \frac{1}{\sqrt{(2\pi)^d |R_i|}} e^{-\frac{1}{2}(X_t-\tau_{Ri})^T R_i^{-1}(X_t-\tau_{Ri})}$$

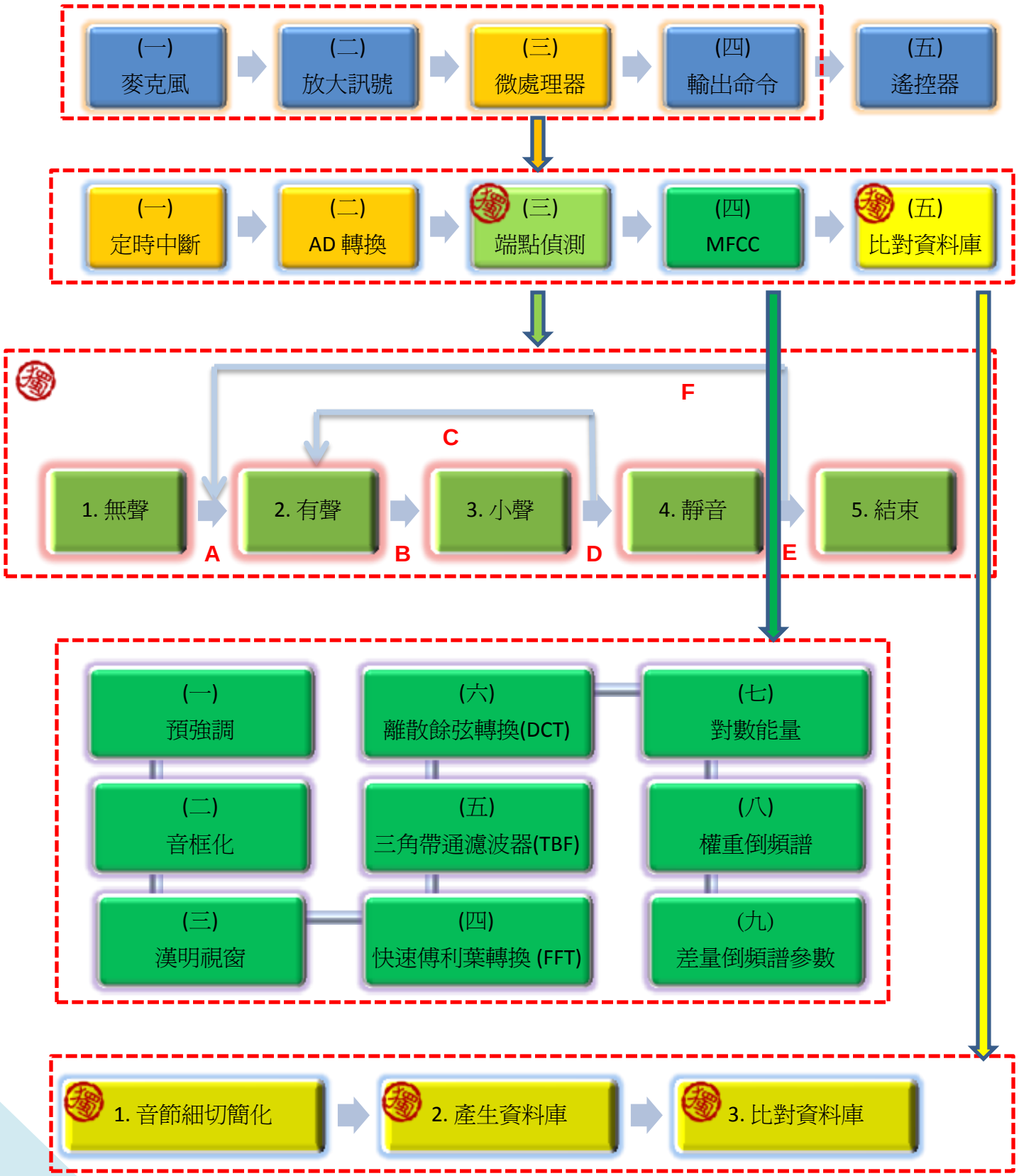
	d 爲特徵向量維度 X_t 爲特徵向量 τ_{Ri} 爲狀態的平均值 R_i 此密度函數的共變異矩陣 $G_i(X_t)$ X_t 組特徵向量和第 i 個狀態相似值

(五) 使用**維特比演算法**（ Viterbi ）調整狀態分割，維特比演算法可以找出總機率值爲最大的路徑，經由反推此路徑，重新切分發音，變成新的 6 等份，如下表黃區。

輸入特徵	原前音節ㄒ++		原後音節：++ㄩㄣ			重新切分
1	77E					新前音節 ㄒ++
2	ED6	DFD				
3	15F6	143B	1402			新後音節 ++ㄩㄣ
4	1D02	1A80	19BD	1A2C		
5		209E	1F2A	1F8A	20C1	
6			2492	24BC	260B	27AA
7			29D0	2A20	2B27	2CDE
8				2F3D	307E	31FA
9				3484	35A3	374D

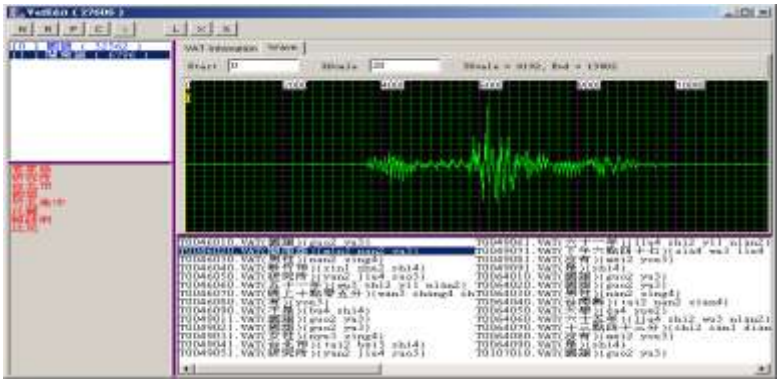
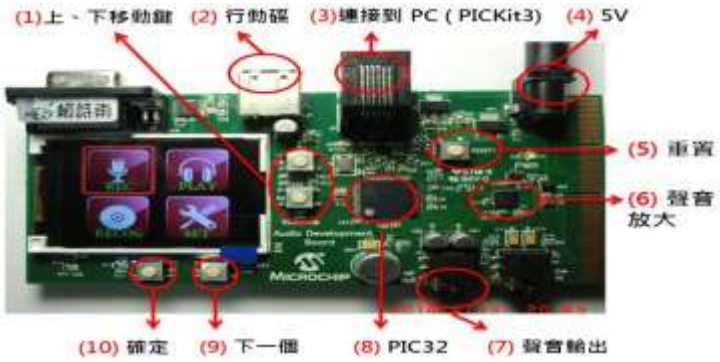
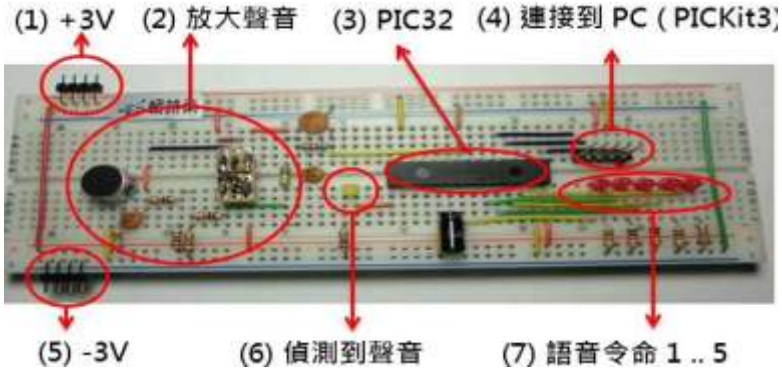
(六) 產生新的資料庫

九・整合系統： 中有 獨 者表示作品的獨創技術，以分辨公知技術：



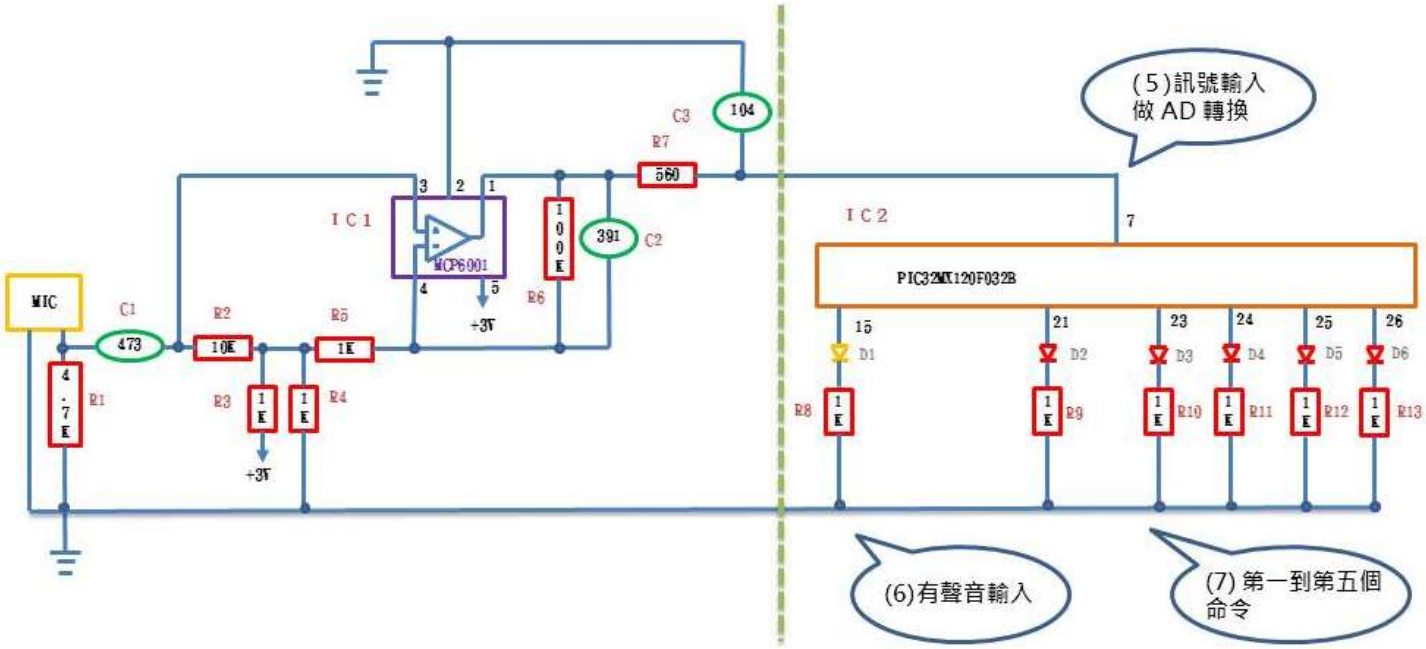
伍、 研究結果

(一). 產出：

	平台	產出	圖示
1	PC	(1) 辨認率分析： RecogFile.Exe (2) 資料庫建立： CalcData.Exe (3) 辨認展示： VatEdit2.Exe (4) 拒絕率測試： Refuse.Exe	
2	開發板	PIC32 / PIC24 原廠開發板上的語音辨識展示	
3	麵包板	作品操作方式如下： 第一步：(1)(5) 上電 第二步：黃燈 (6) 息滅 第三步：說出下面之一命令：（分別是英語、國語、客家語及台語四種語音命令） 1. Eagle （英語念 ：ㄧ、ㄍㄡ） 2. 語音辨識 3. 客家 （客家話念：ㄏㄚˇ 、ㄍㄚㄏˊ） 4. 說話 （台語念 ：ㄍㄨㄙ 、ㄨㄟ） 5. 通訊錄 PS: 辨認的結果對應 (7) 由左到右的 5 個紅色的 LED PS: 黃燈 (6) 息滅後才能講下一個命令	

(二). 作品的麵包板精簡硬體線路：參考圖表 8：可對應到上頁的麵包板實體圖。

1	**放大聲音**	IC1（MC6001），100 倍放大（R6 / R5）
2	**高通濾波器**	C1，R2（頻率約 >= 340 Hz 可通過）
3	**低通濾波器**	R7，C3（頻率約 <= 4000 Hz 可通過）
4	**一階反鋸齒**	C2 ，IC1（MC6001）
5	**進行 AD 轉換**	放大的訊號 PIC32 PIN7（AN5 / RB3 ）
6	**D1 亮 ➔ 有聲音輸入**	PIC32 PIN 15
7	**D2 .. D6 亮 ➔ 辨認第一到第五個命令**	PIC32 PIN 21 .. PIN 26



圖表 8

(三). PC 上的辨識率實驗結果（使用 MAT 純語音資料庫）：

	MAT 語音	MAT-2000 句子數	MAT-400 句子數	MAT-160 句子數
1	2 字詞	42,282 / 52,162 = **81%**	7481 / 8540 = **87%**	2977 / 3390 = **87%**
2	3 字詞	14,052 / 15,913 = **88%**	2440 / 2640 = **92%**	995 / 1069 = **93%**
3	4 字詞	19,625 / 22,755 = **86%**	3403 / 3744 = **90%**	1401 / 1507 = **92%**
4	5 字詞	1,532 / 1,896 = **81%**	276 / 305 = **90%**	111 / 130 = **85%**
5	2-5 字混合	75,414 / 92,726 = **81%**	13274 / 15229 = **87%**	5386 / 6096 = **88%**

(四). 系統須求：

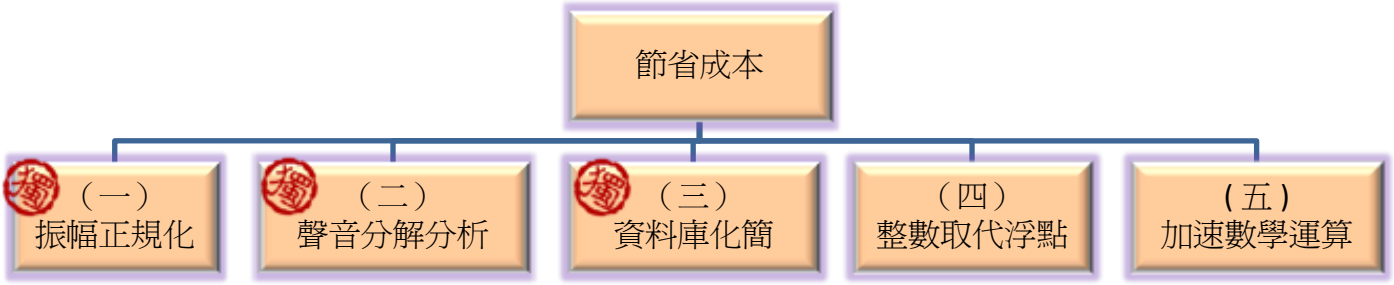
1	程式	**11K**
2	RAM	**2.5K**
3	辨識速度	(1) 命令數多，則變慢 (2) 處理器至少要 10 MIPS 以上的速度
4	單筆前音節資料量	**62 Bytes**
5	單筆後音節資料量	**124 Bytes**
6	資料庫	依命令數而定，約 **1K .. 40K**

(五). 估計辨識 5 個語音命令的成本：

1	PIC32MX120F032B	55 元
2	電路板	4 元
3	麥克風等其他零件	28 元
	總計	87 元

陸、 討論

低成本的語音辨識，有下面的**關鍵因素**，請參考圖表 9：



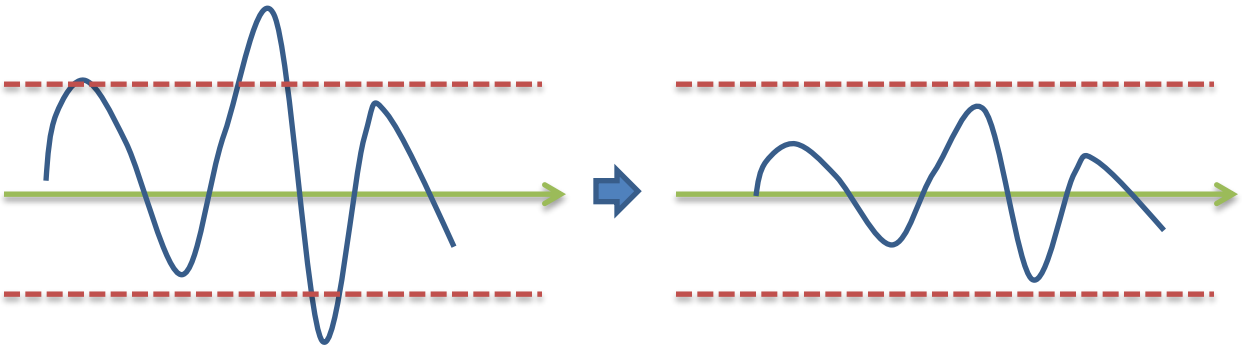
圖表 9

(一). 振幅正規化：

振幅正規化，可以自動調整強弱不同的音量，一般是由硬體完成。

使用軟體的方式實作，會導致系統反應時間變慢，而且觸發自動調整的上下限閥值，不容易設定。

針對上述的缺點，我設計了對每一個音框（200 點聲音）正規化（Normalize）處理，將每一個音框的語音最大振幅，自動縮小或放大到期望的振幅，即可即時完成振幅的修正，達到部份 AGC 的效果（圖表 10）。



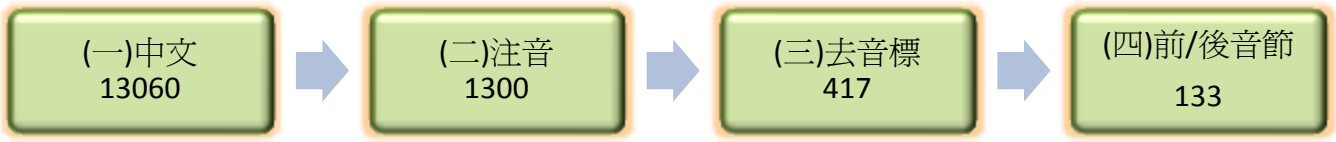
圖表 10

實驗結果如下：經過振幅正規化的調整後，聲音振幅變化對辨識率影響很小。

	振幅調整實驗	總句子數	正確句子字數	辨識率
1	原聲	92,726	75,414	81%
2	放大到 20000		75,398	81%
3	縮小到 1024		75,453	81%

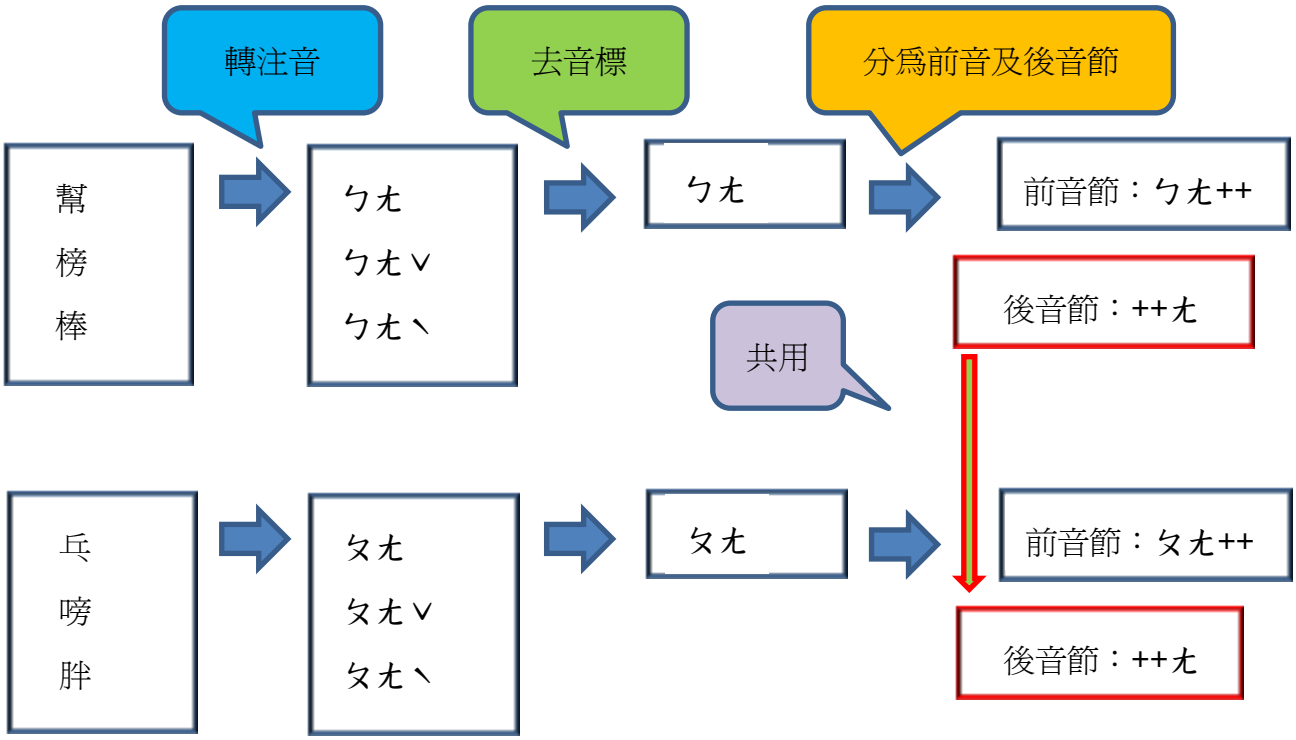
(二).聲音分解分析：

對中文的重新分析，可以將資料庫縮小爲總字數的 **1 / 100** ，請參考圖表 11。



圖表 11

圖表 12 說明如何由中文字對應到前音節及後音節的例子，其中後音節 ++ㄤ 可以共用同一資料，所以使用前音節及後音節的資料量（133）比較去音標的注音（417）更少。



圖表 12

(三). 資料庫化簡：

一般的學術論文都使用 39 組 MFCC 浮點數來記錄一筆特徵：

	MFCC 係數	雙精確度的浮點數（佔 8 Bytes）
1	39 組平均值	➔312 Bytes
2	39 組變異數	➔312 Bytes
3	1 組變異數乘積和	➔8 Bytes
小計		共佔 632 Bytes

經多次實驗，在影響辨識率很小的情形下，取 29 組特徵，並使用整數的指數取代浮點數：

	MFCC 係數	整數
1	29 組平均值	➔29 Bytes
2	29 組變異數	
3	1 組變異數乘積和	➔2 Bytes
小計		共佔 31 Bytes

故改進的資料量較一般的 MFCC 係數縮小 1 / 20。

加上上一章的方法，可以使資料量縮小爲原字數的 1 / 2000。

作品如果要辨識全部語音的最小資料庫規格如下：

可辨識字數	漢語拼音數	音節數	一個音節對應一組特徵的資料庫大小
繁體 13060 簡體 6768	412	133	10354 Bytes（對應全部中文的音節）

(四).整數取代浮點：

微處理器的等級，決定運算的時間及價格：

	處理器	各種數學運算的時間	價格
1	高級	整數 = 浮點 = 乘法 = 除法 = 左右旋轉	高
2	低級	整數 = 乘法 = 左右旋轉	低 ← 作品
3	便宜型	整數	最低

浮點運算及除法都要花費很多時間，可以使用**整數**取代浮點運算及除法來加速運算。

由下表可知：以 Microchip 的處理器來說，整數運算約較浮點運算快約 20 倍。

	PIC32 的乘法週期	PIC24 的乘法週期
整數	6	15
雙精度浮點	159	317

1. 取代浮點運算的例子：

(1) 假設：A 爲整數 = 192，要將 A x 0.8 ➔ 153

(2) 取代方法：STEP1：先將 0.8 x 1024 = 819
STEP2：將 A x 819 / 1024 = 153 ➔ 同 (1) 答案

2. 取代除法運算的例子：

(1) 假設：A 爲整數 = 192，要將 A x 0.8 ➔ 153

(2) 取代方法：STEP1：先將 0.8 x 1024 = 819
STEP2：將 (A x 819) >> 10 = 153 ➔同 (1) 答案

上項中的 >> 10 (右旋 10 位)，在電腦運算上等同於除 1024 （ $= 2^{10}$ ）

(五). 加速數學運算：

語音辨識中，大量的使用各種數學運算，須要很長的計算時間，可使用查表法加速：

1. **已知參數的數學運算使用查表法例子：**

(1) 假設：A 爲整數 = 192，要將 A x sin($\frac{1.3*\pi}{4}$) ➔ 163

(2) 取代方法：STEP1：先將 sin($\frac{1.3*\pi}{4}$) x 1024 = 873

STEP2：將 A x 873 / 1024 = 163 ➔ 答案

可以將 STEP1 事先先計算出來，建成表格

2. 針對未知參數的數學運算例子：

(1) 平方根：可以用牛頓法快速計算

(2) Ln(N)：可用查表法，先計算出 Log_2(N)

再使用下面公式，配合整數運算，計算出來 Ln(N)：

$Log_2(N) = \frac{Ln(N)}{Ln(2)}$

(六). 整數溢位的討論：

整數運算必須處理溢位的問題， 例如：2 位數加 2 位數，其果可能是 3 位數，如果將此結果存回 2 位數的儲存體，就會發生溢位。

一般用多少**位元數**表示微處理器能處理的整數範圍：

	位元數	整數範圍
1	16 位元	0..65535 ← 本系統
2	32 位元	0.. 4294967295

如果希望能在 16 位元的微處理器上辨認，表示任兩次的數學運算，必須確定不會發生溢位問題（超過 65535 ）。

作品在 **16 位元**（PIC24）或 **32 位元**（PIC32）上均可執行。

（使用可處理溢位的隱藏馬可夫模型，確保在運算隱藏馬可夫模型陣列時，不會發生益位）

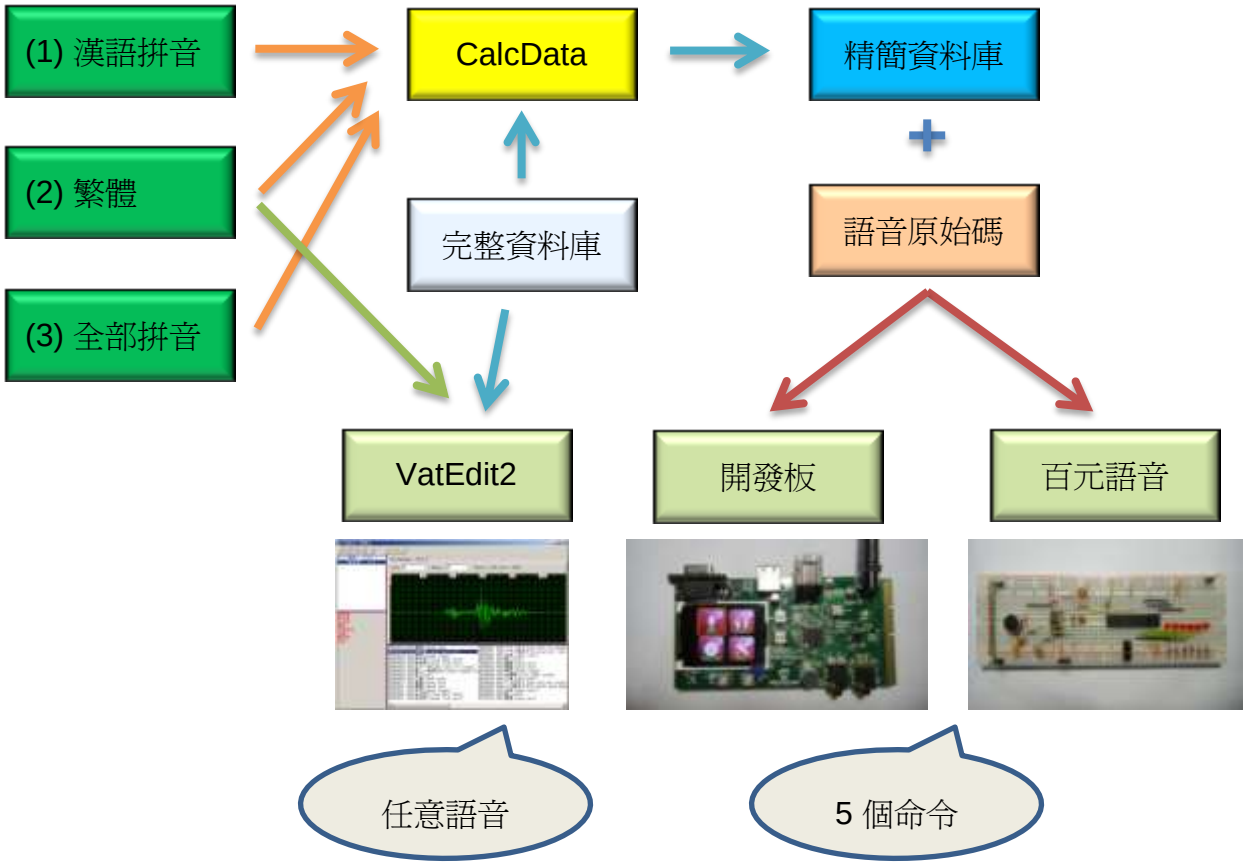
(七). 進一步依指定的命令重新粹取（縮小）資料庫：

針對特定的電器，僅須一些特定的命令，這些有限的命令，不會包含全部的音節，所以可以放入需要音節到資料庫即可。

例如：作品在 PC 上的程式（VatEdit2），使用 133 個音節，可以任意組合、辨識 13060 個繁體中文，但是在百元語音辨識系統上，可能僅須識別 5 組命令、13 種音節，所以經由 CalcData 程式粹取後，重新產生新的精簡資料庫，僅有必要的音節才會燒錄到百元語音辨識系統上，可進一步節省 ROM 的大小。

	系統	命令數	前音節	後音節	資料庫	說明
1	VatEdit2	任意	99	34	10,534 Bytes	1. 左方爲每一音節一個中心點計算（最精簡的資料庫情況） 2. 作品採用每一音節 4 個中心點，以提高辨識率，故在實際應用上資料庫大小爲左方值的 4 倍
2	開發板	5 個	11	9	1,798 Bytes	
3	百元語音	5 個	11	9	1,798 Bytes	

圖表 13 將不同形式的命令 (1)漢語拼音、(2)繁體、(3)全部拼音 ➔ 粹取對應的精簡資料庫。



圖表 13

(八). 提高語音命令的辨識率：（解決：說出語音命令，結果辨識錯誤）

1. 法一：多產生一些同音的資料庫，例如：將錯誤的語音收集後，重新產生新的資料庫。

2. 法二：多產生一些將近似的命令，例如：開燈、打開燈等，均視爲相同的命令。

(九). 誤辨語音命令的討論：（解決：說出其他聲音，結果辨識成語音命令的錯誤）

常見一些和命令不相干的聲音，會被誤辨成語音命令，這種情形，可以使用一些方法避免：

1. 法一： 使用硬體開關，例如：要先按下遙控器的按鍵後，才能說出語音命令，說完語音命令後，再放開按鍵。

2. 法二： 使用關建詞來引導語音命令，例如：可以給作品一個匿稱：愛麗斯，語音命令要先說匿稱，再加上命令，如：愛麗斯—開燈。

3. 法三：可以使用一些吸收錯誤字的方式來減少：下例實驗中，將 MAT 語音資料和 5 個語音命令進行辨識，原本的誤認比例爲 83.53%，在隨意加上三個吸收錯誤的命令後降爲 50.34%。

由此可知，加入吸收錯誤字，可以明顯降低雜音（非命令）➔ 誤辨成語音命令的情況。

	語言	內容	發音	誤認字數	加入吸收錯誤字的錯誤字數	
1	英文	Eagle	ㄧ、ㄍㄡ	11,170	8,692	
2	國語	語音辨識		9,613	4,382	
3	客語	客家	ㄏㄚˇ 、ㄍㄚˊ	18,627	11,769	
4	台語	說話	ㄍㄨㄥ 、ㄨㄟ	19,064	13,690	
5	國語	通訊錄		18,982	8,154	
小計				77,456(83.53%)	46,687(50.34%)	
6	國語	哈哈哈（吸收錯誤字用）			4,044	吸收錯誤字數 = **30,792**
		共共共（吸收錯誤字用）			12,828	
		死死死（吸收錯誤字用）			13,920	

柒、 結論

一. 關鍵因素對於作品的影響：

	關鍵因素	減少硬體成本	減少 ROM	減少 RAM	增加運算速度
1	振幅正規化	●			
2	聲音分解分析	●	●	●	●
3	資料庫化簡	●	●	●	●
4	整數取代浮點	●	●	●	●
5	加速數學運算	●			●

二. 作品中有關 的獨創方法摘要：

	功能	說明
1	**端點偵測**	1. 設計緩衝區音塊來觀察較長時間的能量變化 2. 設計狀態圖及轉移方式描述語音的生命週期
2	**資料庫模板**	1. 分析聲音，設計 133 個音節對應全部的中文字 2. 特徵值採用整數的指數，取代符點數
3	**資料庫比對**	1. 設計可處理溢位的隱藏馬可夫模型
4	**振幅正規化**	1. 取代部份的硬體 AGC 功能

三. 作品中比較於其他商業產品 的規格說明：

	規格	作品	比較一般商業的產品
1	程式	11K	一般均在數百 K 以上
2	RAM	2.5K	一般均在數十 K 以上
3	5 個命令資料庫	1K – 4K	一般均在數百 K 以上
4	微處理器	16 位元以上	32 位元以上
6	硬體實作成本	50 – 100 元	數千元以上
7	辨識率	優	

四. 心得：

想要獨立完成一個語音辨識系統，對於我來說，實在是很大的挑戰，因爲雖然電腦語音辨識已經慢慢流行起來，網路上可以找到一些資料，但是紙上談兵的理論很多，在實作方面，可以用到的資料，仍是非常少，所以上網查資料，自已做實驗，變成我最重要的知識來源。

這個題目，我總共做了一年 10 個月，其間跌跌撞撞，算是吃了不少苦，尤其在繁重的課業中，要抽出時間找資料或做實驗，或厚顏請學校幫忙申請 MAT 的純語音資料庫，都是自己完成，這些經驗中有苦有樂，但在長時間的堅持之後，回想經歷的過程，卻是甘甜無比，對我的人生幫助很大，並且學習到了很多書本上學不到的知識，我把這些心路歷程記錄下來，請參考參考資料中【我的研發文件】。

本次作品在演算法的改進後，是全球**最小**的非特定人語音辨識系統，（僅須 16K ROM＋ 4K RAM＋ 16 位元的低成本處理器），甚至我在網路上可以找到的最小的商用語音辨識系統，在相同的條件下，都須要比作品多數十倍以上的資源才可以執行，這些方法都不適合低成本家電使用，所以可以說：本作品是現今適合低成本家電語音辨識的唯一方案，具有廉價非特定人語音辨識在技術上的突破及真實的商業價值。

受限於時間及知識，作品僅是原型機的展示，但是已經証明了低價語音識別的可行性， 也許未來的不久，可以看到週遭的人，對著電器自言自語，可不要覺得太奇怪喔！

捌、 參考資料及其他

【一本書】

語音訊號處理-王小川（全華科技圖書）
Visual Basic 與語音辨識（揚振光）
PC 電腦語音辨認實作（陳明熒）

【學位論文】

陳奕宏（2006）32 位元處理器之定點數 MFCC 演算法的改進與探討
周俞璋（2007）整數型態之 FFT 的實作與改進，及其在語音辨識之應用

【我的研發文件】

[ASR-5002] 語音辨識關鍵因素.txt
[ASR-5011] 辨認簡介.txt
[ASR-5012] 語音元素分解.txt
[ASR-5013] 原始語音收集.txt
[ASR-5020] 隱藏馬可夫模型.docx
[ASR-5100] 語音辨識 IC 分析.txt
[ASR-5200] 編譯環境.txt
[ASR-5300] 硬體分析.txt
[ASR-5400] 軟體初始化.txt
[ASR-5410] 微處理器及算放大器說明.docx
[ASR-5420] 麵包板和 Microchip 展示板比較.docx
[ASR-5500] 語音辨認測試記錄.txt
[ASR-5600] 展示式說明.txt
[ASR-5610] VatEdit2 說明.docx
[ASR-5620] CalcData 說明.docx
[ASR-5630] VerifyX 說明.docx
[ASR-5640] RecogFile 說明.docx
[ASR-5650] Refuse 說明.docx
[ASR-5660] PIC_100 說明.docx
[ASR-5900] 漢語拼音和注音對照表.txt
[ASR-5901] 繁體和注音去音標對照表.txt

【評語】040802

能夠利用成本低廉的元件，使用獨特的端點偵測方法，加上資料庫模板的概念，完成一個簡易型語音辨識，本件作品完成度高，並具有市場潛力。