

第三章 系統硬體架構

本系統硬體除了採用市面上現成之硬體設備外，也自製了額外之硬體設備，以滿足額外之需求，而軟體則針對嵌入式電腦語音溝通系統和替代性輸入裝置之功能進行設計。本章將針對硬體部份作深入的討論。

第一節 採用現成之硬體設備

在發展系統的過程中，我們採用了 MSTI-3000 Windows CE 5.0 發展系統，其中包含了 ICOP Vortex86-6047LV 板子， 8.4 吋 TFT(Thin Film Transistor)彩色觸控螢幕(面板)，以及作業系統發展軟體。其中發展系統如圖 3-1[13]。



圖 3-1 MSTI-3000 Windows CE 5.0 發展系統

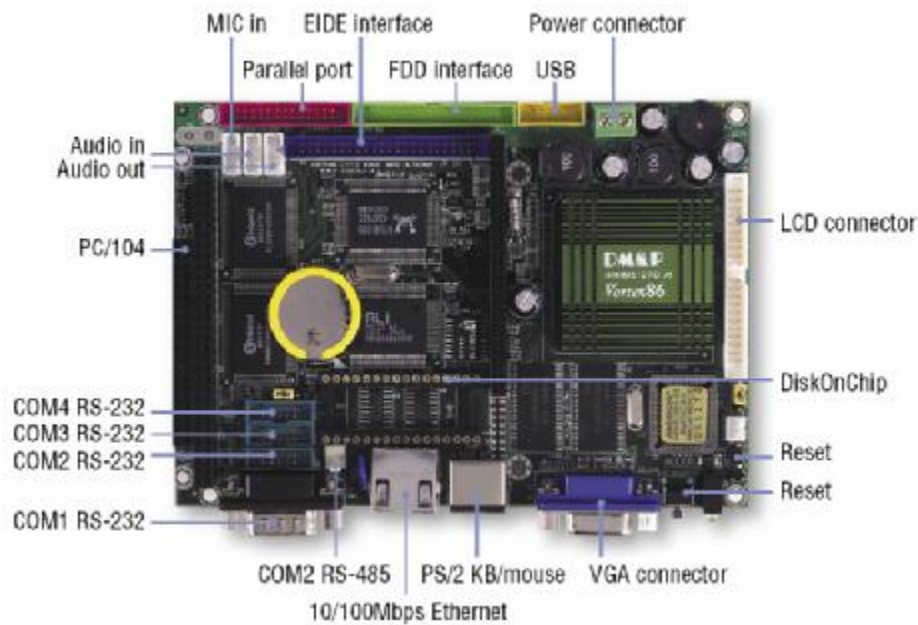


圖 3-2 ICOP Vortex86-6047LV Board

圖 3-2[14]為發展系統之主機板圖，CPU 採用了 DM&P(SiS)Vortex86™ 的 System-on-Chip CPU-166MHz，記憶體則為 128MB 之 SDRAM，並具有網路及音效設備。多種的 I/O 界面提供了不同的功能，其中透過更換不同大小之 Disk on Module(DOM)，能讓系統之硬碟容量隨之變更，因此發展軟體其容量大小之彈性更大。此外透過 TFT 彩色觸控螢幕，除了能讓圖形更加生動化以外，也讓使用者可透過觸控的方式而達到輸入的效果。

在系統操作的過程中，使用者經由觸控面板而對系統下達動作指令，透過相關硬體的配合來處理指令之動作，並將回應訊息透過 TFT 彩色螢幕來呈現，進而完成與使用者間的互動[3]。圖 3-3 為系統硬體架構圖[15]。由於採用市面上現成之硬體架構，因此將不針對該硬體設計做完整的介紹。本節主要介紹我們實際所用到之硬體設備，其架構圖如圖 3-4。

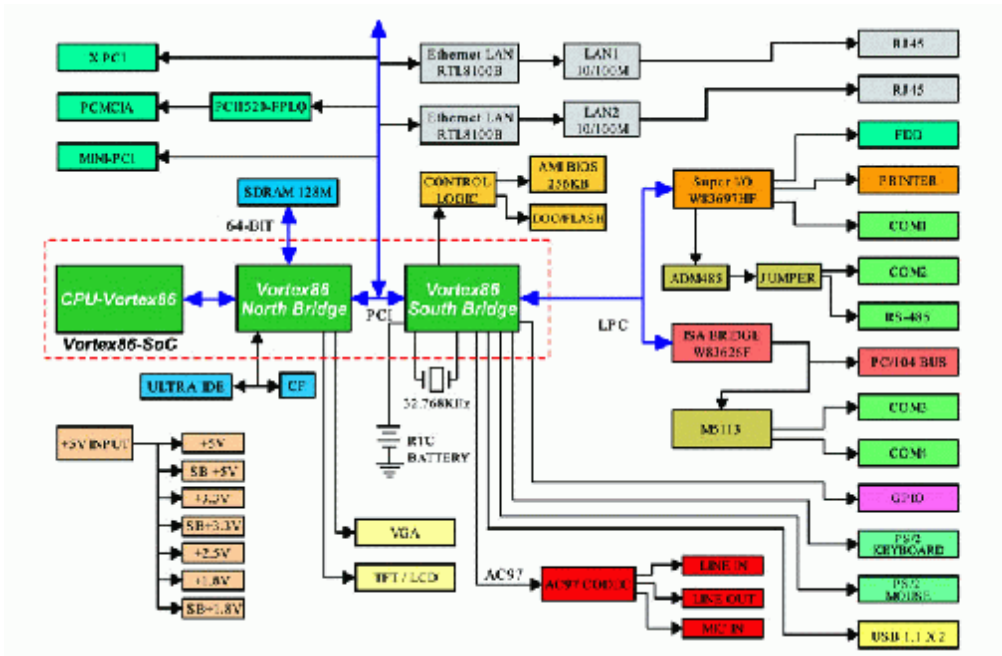


圖 3-3 系統硬體架構圖

本系統所使用的硬體設備包含了中心、顯示、觸控、週邊及電腦連接界面等部份，如圖 3-4。

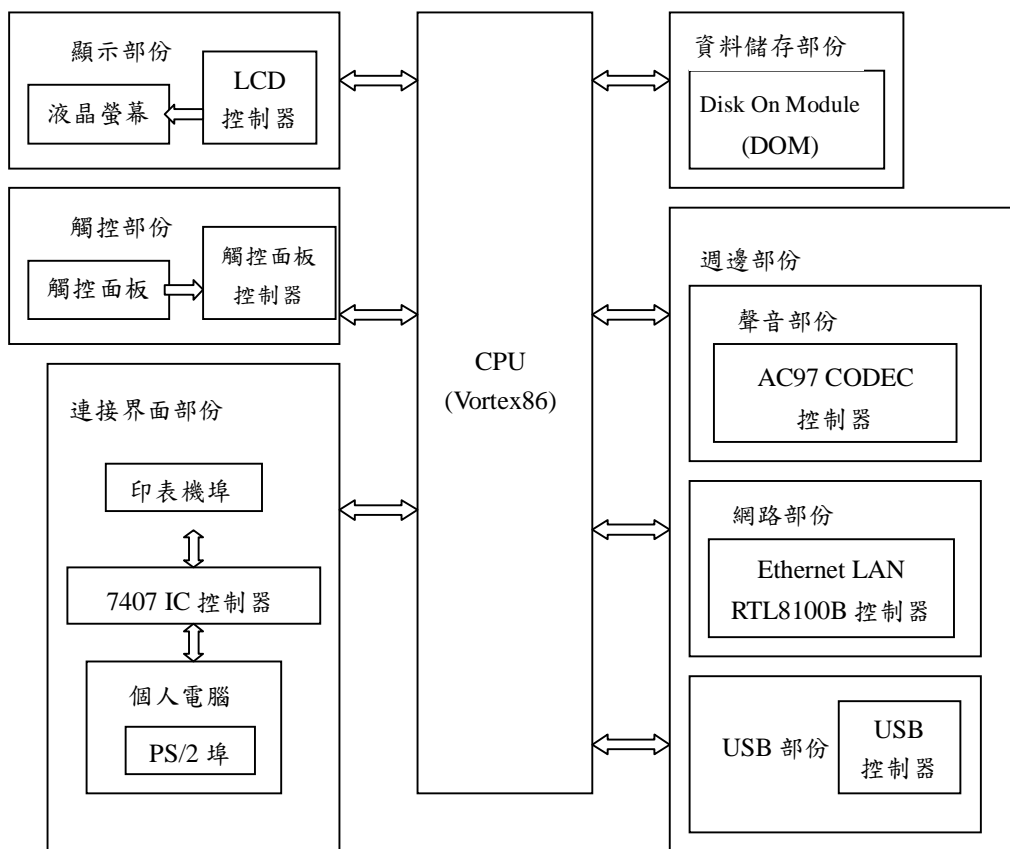


圖 3-4 本系統所使用之硬體架構圖

一、中心部份

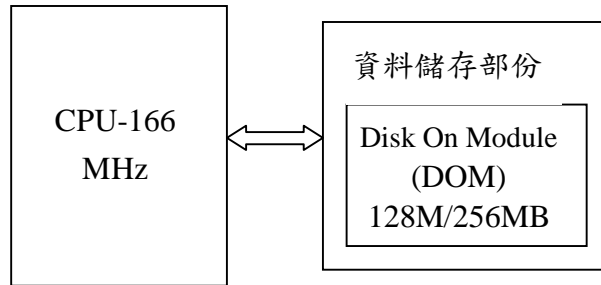


圖 3-5 中心部份架構圖

圖 3-5 為本系統的中心部份架構圖，包含了 CPU 及嵌入式小硬碟(DOM)。

為了追求高效能，CPU 部份我們選擇了對嵌入式系統而言算是高頻率的 166MHz 頻率之 CPU。資料儲存的部份則以嵌入式小硬碟作為我們的程式、資料等儲存空間。它是一個 IDE 界面的硬碟，由於其大小彈性大，因此對嵌入式系統在發展不同大小的程式有很大的彈性空間，普通常見之硬碟大小有 64MB、128MB、256MB、512MB、1GB 等。

根據我們所發展不同大小的作業系統映像檔(Image:nk.bin)，所建立最小的映像檔檔案大約 11MB，最大可至 30MB，取決於需加入至系統的功能之多寡。除了作業系統，程式的大小也會影響使用之嵌入式小硬碟的大小。以我們所研發之程式，其大小約為 1MB，不過圖形檔和聲音檔則額外佔了約 45MB 的記憶體空間，因此選擇 128MB 之嵌入式小硬碟最為合適。

二、顯示部份

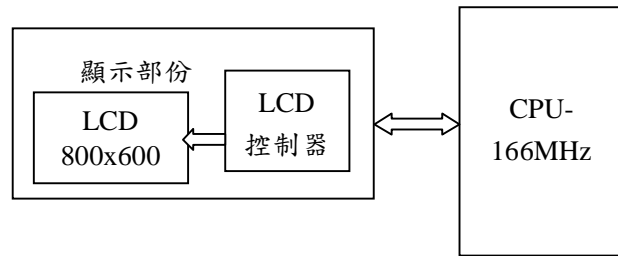


圖 3-6 顯示部份架構圖

顯示部份架構圖如圖 3-6 所示。為了讓使用者能看到系統所呈現之資訊，系統必須具備可以作為顯示的裝置。本系統在顯示架構中以彩色液晶螢幕作為顯示器，使用它的原因不外乎體積小且重量輕之優點。至於尺寸的選擇，本系統採用了 10.4 吋的液晶螢幕，其解析度為 800x600。除此之外，液晶螢幕之大小也可視使用者之需求而更換。若重量是其最大考量，則可選擇尺寸較小之顯示器，否則尺寸大之顯示器則有助於視力較不佳或肢體障礙較嚴重之使用者。

LCD 的顯示透過 LCD 控制器來控制，而顯示內容則是根據使用者所選擇之圖形版面來決定。

三、觸控部份

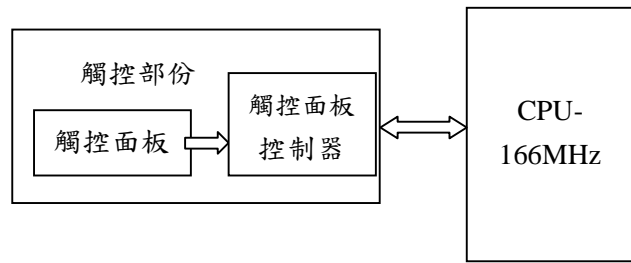


圖 3-7 觸控部份架構圖

有了顯示裝置後，要讓使用者能根據系統顯示的資訊產生回應，進而要求系統達成使用者所希望的動作結果，中間就需要一個界面讓使用者能跟系統互動及溝通。這部份的需求就透過觸控部份的架構完成，如圖 3-7。觸控面板能將使用者藉由肢體或輔助工具在螢幕上所接觸到的位置，透過 TFT 控制器，系統程式可從中取得某值，加上內建程式的轉換，就能確認使用者所觸控之正確位置坐標，系統程式根據此位置資訊做出相對應的系統動作，進而完成使用者的動作要求。

我們的系統所提供之觸控面板和 USB 控制器皆使用同一種 USB 界面，而本系統也提供了兩組相同之 USB 界面，以方便觸控和 USB 之控制同時進行。

四、週邊部份

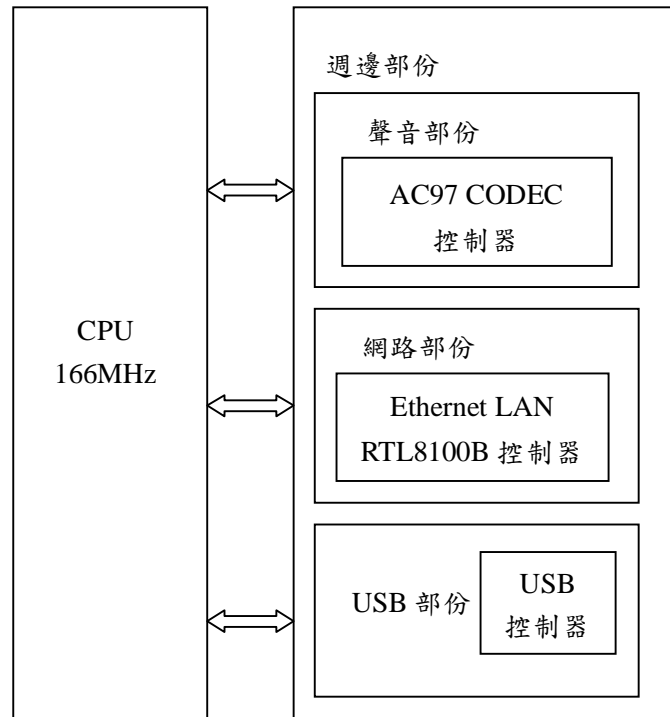


圖 3-8 周邊部份架構圖

如圖 3-8 的架構圖所示，此部份為系統所使用之週邊裝置，包括聲音、網路和 USB 的部份，其功能描述如下：

(一)聲音部份

本系統具有語音輸出之功能，因此需要用到聲音控制器，透過系統程式內建之函式，將讀入之聲音檔(.wav)透過控制器的編解碼，送出至音效卡，展示發聲的功能。聲音之內容則根據使用者在觸控面板所選擇之圖形按鍵。

(二)網路部份

本系統在研發的過程中使用了網路功能進行對系統之測試，如圖 3-8 所示，網路卡為 Realtek 8100B，其速度為 10/100Mbps。此功能對於使用者而

言有不同的選擇。若使用者除了使用本系統所提供之語音溝通系統和替代式輸入裝置的功能外，也希望透過本系統進行上網，是能輕易達成的，只要設定正確的 ip 位置，連線成功後就能具備上網的功能。

(三)USB 部份

由於本系統提供了兩個 USB 界面，其中一個為連接觸控面板，另一個界面則連接到 USB 控制器進行控制。由於在發展系統的過程中我們需要用到 USB 隨身碟進程式或資料存取的動作，因此利用了系統所提供之 USB 接頭完成。

至於第五部份的電腦連接界面，由於其中大部份為自訂之硬體架構，因此在此第三節我們將進行更詳細之介紹。

第二節 鍵盤界面之介紹

由於替代性輸入裝置的研發必須具備對鍵盤相關技術之了解，因此本節將說明鍵盤的工作原理。每當鍵盤偵測到按鍵時，必須送出相關信號告知電腦有按鍵被按下。鍵盤和電腦則透過資料(Data)和時脈(Clock)兩條信號線來與電腦做雙向傳輸，藉由資料和時脈的狀態即可判斷鍵盤是否可傳送信號。資料主要以 Make Code 和 Break Code 組成。Make Code 為鍵盤偵測到按鍵被按下後送出至電腦的資料，用來告知電腦按鍵狀況的發生以及是哪個按鍵被按下，而 Break Code 則是按鍵被鬆開時所送出的資料，用來告訴電腦是哪個按鍵被鬆開。以英文字母"a"鍵為例，它的 Make Code 為 1CH(十六進位)。當"a"鍵被按下時，鍵盤將會送出 1C(Make Code)至資料信號線。若我們持續按著該按鍵不放，而達至鍵盤所設定之單鍵延長時間時，則另外一個 1C 掃描碼將會被送出。直到按鍵被鬆開後，鍵盤將送出 FOH(十六進位)和原來之掃描碼 1C，告訴電腦該鍵已被放開。此二碼則為"a"鍵之 Break Code。圖 3-9 顯示了鍵盤上各個按鍵之掃描碼。

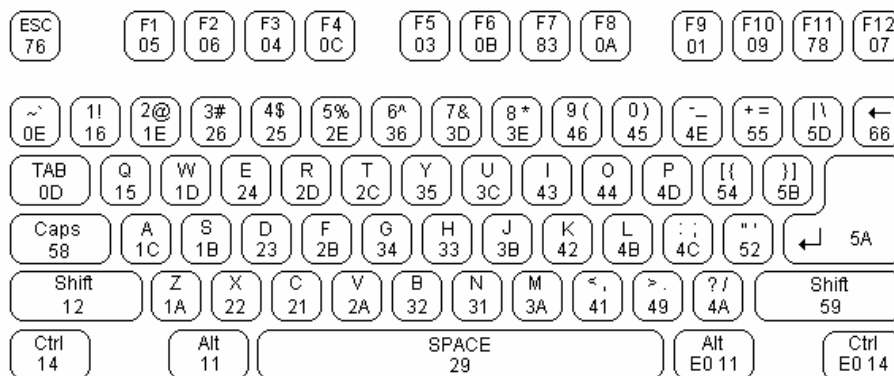


圖 3-9 鍵盤掃描碼

瞭解了鍵盤之原理和認識了按鍵之掃描碼後，接下來我們將介紹鍵盤的通訊

協定(protocol)。鍵盤的通訊協定可分為兩種，即為鍵盤至主機端及主機至鍵盤端的資料傳送。

一、鍵盤至主機端(Keyboard to Host)

由於鍵盤信號的傳輸方式是以雙向傳輸在進行，因此鍵盤可送資料至主機端，而主機也可送資料至鍵盤端。主機端能在任何時段傳送信號至鍵盤端。相對的，鍵盤端則必須等待鍵盤資料信號和時脈信號同時為 1(high)時才能傳送資料至主機端。若主機端將鍵盤時脈信號變為 0(low)，則鍵盤會先儲存其資料在緩衝區(buffer)直到鍵盤時脈信號被釋放，也就是其值為 1。當主機端將鍵盤之資料信號變為 0 時，鍵盤就可以準備接受從主機端傳送過來之資料。

資料信號是以串列的方式傳送，並每次傳送 11 位元的串列資料，包括一個 Start Bit，一個 Make Code 或 Break Code，一個 Parity Bit(Odd Parity)和一個 End Bit。每個字元將會在時脈信號的下降緣(falling edge)被讀出，而每個時脈信號佔據了約 40 μ s 的時間大小。Start Bit 為觸控面板偵測到按鍵被按下後送出至電腦端資料的第一個位元，其值固定為 0。Make Code 或 Break Code 的資料則包含在第二至第九個送出之位元。Parity Bit 是用來檢查 Make Code 或 Break Code 在傳送過程中是否有產生錯誤的檢查碼，檢查碼之檢查方式包括以 Odd Parity 或 Even Parity 作為其 Parity Bit，而鍵盤的檢查碼則採用 Odd Parity 的方式。[23]

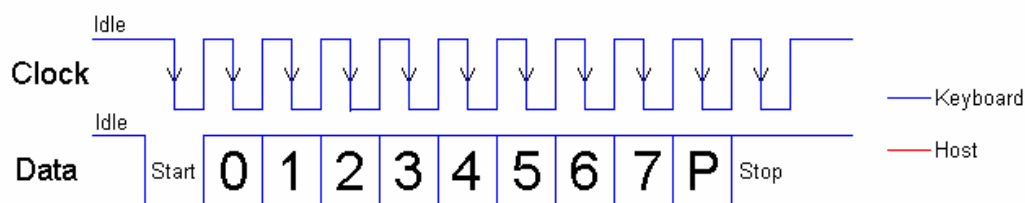


圖 3-10 鍵盤至主機端資料和時脈信號之時序圖

圖 3-10 之時序圖代表從鍵盤傳送一個按鍵碼至電腦端的波形圖。鍵盤之資料信號不一定如上圖在時脈信號上升緣時被觸發，資料信號只需在時脈信號下降緣之前被送出即可，時脈信號是由鍵盤產生的。

二、主機至鍵盤端(Host to Keyboard)

主機至鍵盤端協定產生的情況必須是當主機之資料信號傳回 0 至鍵盤的狀況。為了防止鍵盤傳送資料至主機端的同時而主機卻正傳送資料至鍵盤端的情況，主機會回傳其時脈信號線並停留在 0 的時間大於 $60\mu\text{s}$ 。這是因為 $60\mu\text{s}$ 已經超過了一個字元被傳送的時間長度($40\mu\text{s}$)，因此鍵盤資料信號此時就知道有資料將會從主機端傳送過來了。過了 $60\mu\text{s}$ 後，主機端會將時脈信號由 0 變為 1，鍵盤資料信號就可轉變為 0 等待接受從主機傳來之資料。

鍵盤端將開始產生約 $10\mu\text{s}$ 的時脈信號，當第一個下降緣被偵測到後，主機端將會開始載入第一個傳送字元至鍵盤資料信號上，此字元將會在下一個時脈信號下降緣時被鍵盤讀入。當 8 個位元(一個字元)被傳送完畢後，Parity Bit(Odd Parity)將會被讀入，如圖 3-11。

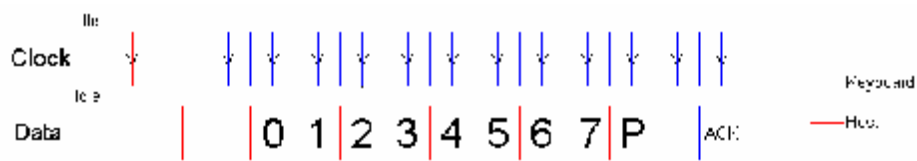


圖 3-11 主機至鍵盤端資料和時脈信號之時序圖

當 Parity Bit 傳送完畢後，鍵盤的資料信號線將處於 1 的狀況，直到下一個時脈信號的產生，這時鍵盤將會回傳一個 ack(0)，代表鍵盤可開始接收新資料。若鍵盤之資料信號在第 11 個字元(Start, 8 Data bits + Parity)不為 0 時，鍵盤將會繼續傳送鍵盤時脈信號直到其資料信號為 0。

第三節 自製之界面電路

為了達到替代性輸入裝置之功能，我們製作了額外的界面電路。首先我們利用印表機埠，透過開集極輸出界面 7407IC，連接成 PS/2 界面，作為從本系統至電腦端鍵盤的界面，進而達成將本系統作為替代性輸入之裝置。因此連接界面包含了從印表機埠至 7407IC 控制器之系統連接界面和從 7407IC 控制器至 PS/2 連接器之電腦連接界面，如圖 3-12。

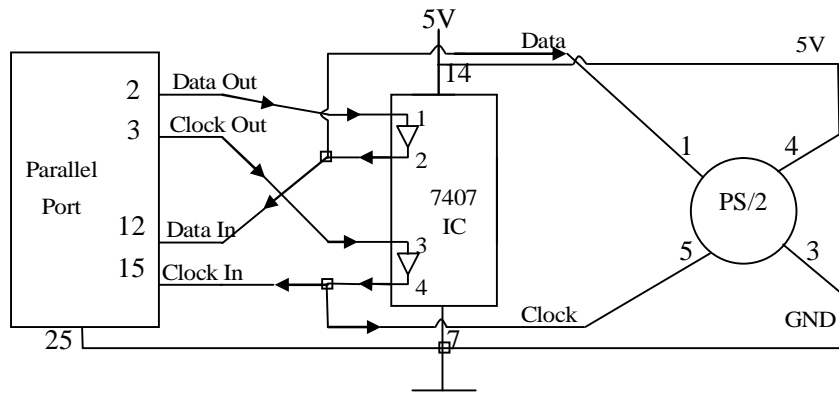


圖 3-12 替代性輸入裝置硬體架構圖

印表機埠提供多種以上之界面方式，包括 Standard Parallel Port(SPP)、Enhanced Parallel Port(EPP)及 Extended Capabilities Port(EPC)。為了配合本系統所提供之印表機埠界面，我們採用了 EPC 之界面。首先我們利用了印表機埠的第二、第三腳(輸出資料)，分別接到 7407IC 控制器的第一和第三腳，作為本系統輸出之資料和時脈信號；而印表機埠的十二和十五腳(只可輸入資料)，亦分別接到 PS/2 之資料和時脈信號，接收電腦端送來的信號。印表機埠的接地腳也需與 7407IC 控制器的接地腳相連。

在此也說明為何利用 7407IC 控制器作為系統至電腦之主要連接界面。

7407IC 是具有緩衝(Buffer)功能的開集極(Open Collector)TTL IC。在使用一般的邏輯 IC，當兩元件 A 和 B 有的輸出接在一起時，可能會因為某一元件 A 有高電位輸出而另一位元 B 有低電位輸出，而造成在 A 的 pull-up transistor 到 B 的 pull-down transistor 這條線上的短路現象，導致過大的電流而造成元件的損毀。所謂開集極電路是指在 TTL IC 的輸出端，只提供 pull-down 而不提供 pull-up transistor，因此元件的輸出值就無法保持在高電位的狀態。因此當有 High-Low 的共同輸出時會穩定的停留在低電位的狀態而不會有短路現象。而若在使用開集極的元件時，我們也應外提升加電阻(External pull-up transistor)，以便在所有元件均為高電位輸出時，能維持高電位的結果。

因此透過以上所列出的自製界面電路，本系統就能達到替代性輸入裝置之功能。