# 第三章 系統設計與實作

本研究提出一以嵌入式系統晶片實現可攜式脈搏生理訊號監測系統，整體系統分為軟硬設計兩部份。硬體使用方面涵蓋：感測器、訊號處理電路、單晶片微電腦、嵌入式系統開發平台、PC 電腦等；軟體使用方面包含：PHP、C、組合語言、Ajax、JavaScript 等程式語言，透過整合所設計之硬體及軟體，將量測得到之人體脈搏訊號儲存於遠端資料庫並顯示於網頁上，使醫護人員得以經由網路從遠端做即時觀察。除此之外，設計一脈搏訊號相似度演算法，以比對資料庫上儲存之人體脈搏訊號資料。詳細討論分述於本章各節。

## 3.1 系統整體架構

圖3-1為本研究提出之整體系統架構圖。整體架構主要分為三個研究構面，一是前端感測系統之開發，透過非侵入式感測器將人體生理訊號轉為電壓訊號，訊號經訊號前置處理電路，濾除訊號中雜訊並轉為數位訊號，接著經由傳輸介面將資料送至嵌入式系統開發平台。

本研究第二個構面分為兩部分，部分一：使用嵌入式系統開發平台，接收來自傳輸介面資料，透過資料接收、還原與合併等資料處理動作之後，經由開發平台上無線網卡將資料傳至遠端資料庫之中，以達本研究所提出之可攜式目的；部分二：架設一網頁伺服器，顯示病患個人生理訊號情形，以使醫護端可經由網路從遠處觀察量測者即時脈搏訊號情況。

本研究第三個構面重點則在於伺服器端量測者生理資訊的儲存與分析探勘軟體系統設計。透過設計之人體脈搏生理訊號分析演算法，長期分析資料庫內人體脈搏訊號，分析結果提供給醫護端作為疾病分析及預防上的參考。本研究整體系統生理訊號處理流程說明如下：

**Step1.** 經由非侵入式感測器，量測人體脈搏生理訊號。在本研究中，採用

壓電式感測器，以量測人體脈搏生理訊號。

**Step2.** 訊號處理電路

1. 由於經感測器所得到之人體脈搏訊號相當微小，因此在訊號處理電路前端設計一儀表差動放大電路，以放大感測器輸出之微小脈搏生理訊號，除此之外，儀表發動放大電路亦可消除訊號中共模雜訊部份。
2. 經由感測器所得到之生理訊號可能包含著不必要之高低頻雜訊，為了濾除不必要之高低頻雜訊在儀表放大器後端加入了高低通濾波器。
3. 由於經感測器所量測之人體脈搏生理訊號具有訊號為負的成分，為了使得後端電路利於處理，本研究中，加入電壓箝位電路，以使整體訊號皆位於零準位以上。

**Step3.** 為了使得訊號得以傳輸至電腦端，需透過類比數位電路將類比訊號轉換為數位訊號。

**Step4**. 採用8051單晶片設計傳輸介面以搭配嵌入式系統發展平台作為資料傳輸之用。

**Step5**. 本研究中為了降低系統負荷度，接收來自8051單晶片之資料並未馬上傳輸出去，待接收一定量之後再加以傳輸。本研究設定每接收兩百筆資料再加以傳輸。

**Step6**. 利用嵌入式系統發展平台上之無線網卡，將人體脈搏生理訊號傳送至遠端伺服器資料庫。

**Step7**. 將人體脈搏生理訊號顯示於網頁上，醫護端可經由網頁觀察量測者生理概況。

**Step8**. 以設計之脈搏訊號比對演算法分析量測者脈搏生理訊號，將分析結果提供於醫護端。

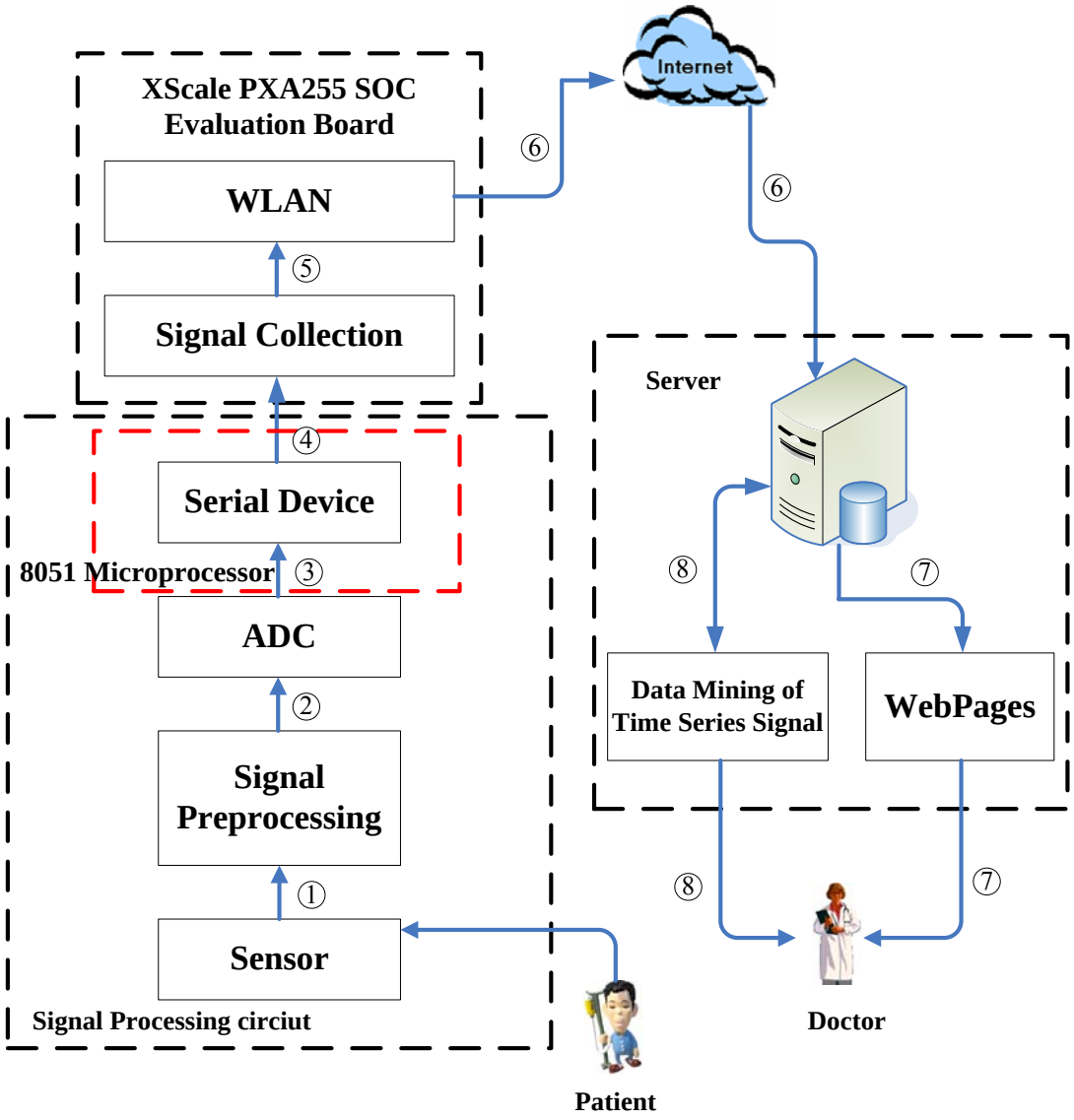


**圖 3-1 系統整體架構**

## 3.2 系統開發環境及開發工具

本節主要介紹本論文在實現系統上所使用之開發環境、儀器、開發平台等，詳述於下列小節。

### 3.2.1 嵌入式系統開發平台

本研究提出一以嵌入式系統晶片實現可攜式居家脈搏生理訊號監測系統。主要採用嵌入式系統晶片作為本研究硬體核心，原因為嵌入式系統晶

片可內嵌作業系統，因此，可設計相關應用程式內嵌其中以利於使用者操作；嵌入式系統晶片可支援一般電腦週邊設備模組，透過安裝週邊模組，可很快速的增加系統功能。因此，與以往的微處理機相較起來，嵌入式系統晶片除了在效能上較為優異，在未來系統的發展上可因環境與時空做即時設計上的變更，而僅需要使用者作些微的調整，較具系統開發上的彈性。

本論文在嵌入式系統開發平台上，採用華亨代理之 XScale PXA255 嵌入式系統開發平台。其主要規格如下：

處理器：Intel® PXA255 400MHz

作業系統：Windows CE5.0 或 Linux

SDRAML：Samsung 64Mbyte

Flash：Intel® strata flash 32MByte

**圖 3-2　XScale PXA255 嵌入式系統開發平台**

### 3.2.2 8051 單晶片

本論文採用 8051 單晶片實作傳輸介面，以將量測電路所得到之脈搏生理訊號經由實作之傳輸介面傳進嵌入式系統平台之中。8051 單晶片其內部含有 4K 位元組可重複燒錄的內部程式記憶體(EEPROM)，最高工作可至 16MHz，產品材質為 HMOS(高性能金屬-氧化物-半導體集成電路)，40-Pins 的 DIP 包裝。在程式燒錄到晶片時只須加電壓約 5~10 秒的方式即可清除其內容，之後再將新的程式燒入至晶片之中。接腳圖如圖 3-3 所示，圖 3-4

為其內部功能方塊圖[49]。

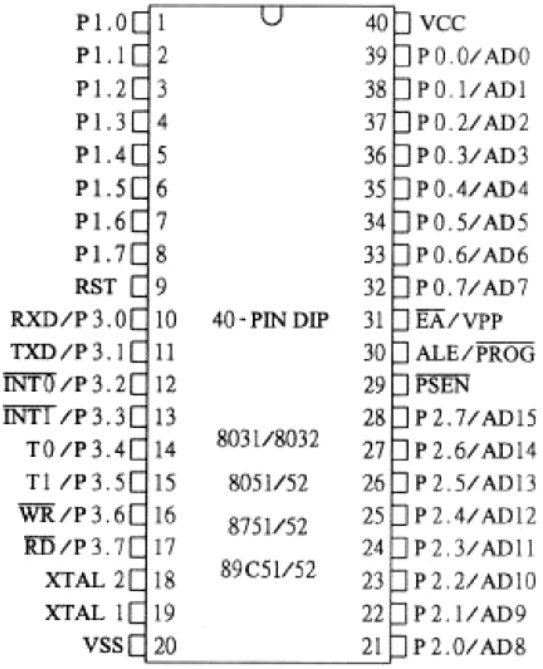


**圖 3-3 8051 單晶片外觀圖**

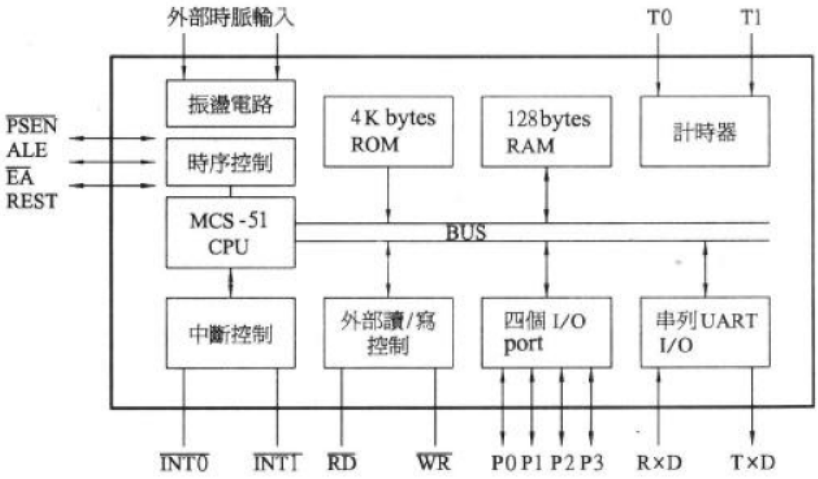


**圖 3-4 8051 單晶片內部功能方塊圖**

### 3.2.3 感測器

本研究在脈搏生理訊號量測上，採用國揚儀器代理之 MLT1010 Pulse Transducer 壓電元件感測器，此感測器無須外加電源，壓力產生變化時即輸出電壓訊號。詳細規格如表 3-1，圖 3-5 為其外觀圖。

**表 3-1　MLT1010 Pulse Transducer 規格**

| | |
|---|---|
| Frequency response | 2.5 to 5000Hz |
| Operating signal range | 0 to 1 V for linear output range when applied across a 1 MΩ resistor.<br>Typically 100 mV output for finger pulse (from 20 mV to 500 mV depending on person) |
| Weight | 25g |
| Size (diameter x thickness) | 22 x 12mm(0.87” x 0.47”) |

**圖 3-5 MLT1010 Pulse Transducer**

### 3.2.4 Linux 作業系統

一般而言，基於嵌入式系統晶片本身及其週邊硬體上限制，我們希望嵌入至晶片上之作業系統能盡量做到精簡化，以提昇系統整體效能；再加上 Linux 作業系統驅動程式模組化的特性－可以動態地將驅動程式給予安裝與卸除，因此本論文選用開放性原始碼之 Linux 作業系統作為本論文在

系統開發程式上之作業系統環境。

## 3.3 硬體電路設計

經由感測器所量測出來之人體生理訊號往往相當微弱且具有相當多不必要之雜訊存在，因此需針對所量測之生理訊號特性，設計相關前置電路，以得到較為正確之生理訊號。以下小節詳述針對人體脈搏訊號所設計之前置電路。

### 3.3.1 儀表差動放大電路

由於感測器輸出可能包含共模雜訊且量測到之訊號亦相當微小，因此採用儀表差動放大電路[50]以濾除共模雜訊，同時放大微小之輸出訊號。其放大倍率，依據前章所述為：

$$A=\frac{Vout}{Va-Vb}=\left(1+2\times\frac{R12+R5}{R9}\right)\times\frac{R11}{R7} \tag{3-1}$$

本研究在實驗過程中，放大倍率約調為 100 倍。圖 3-6 為本研究採用之儀表差動放大電路。

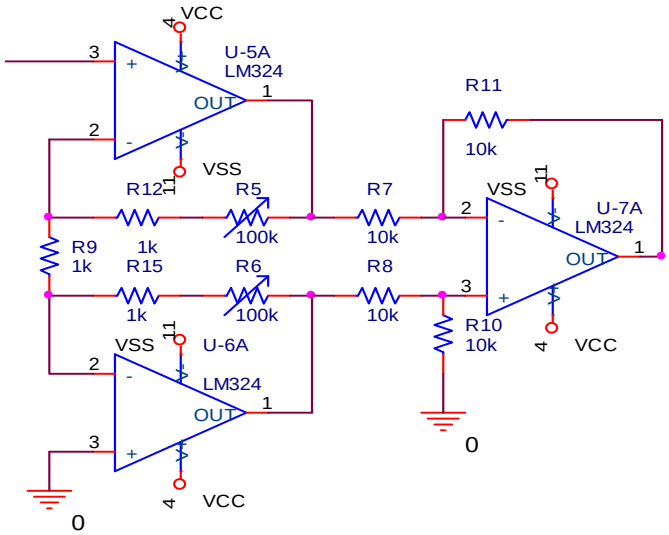


**圖 3-6 儀表差動放大器電路**

### 3.3.2 高低通濾波電路

由於訊號中可能包含許多高低頻雜訊，因此須以高低通濾波器，以濾除訊號中高低頻雜訊部份。在本研究中，採用 Sallen-Key 高低通濾波器[51]以濾除訊號中高低頻雜訊，由前章知其截止頻率式子為：

$$fc = f_H = f_L = \frac{1}{2\pi\sqrt{R_{11}R_{12}C_{11}C_{12}}} \tag{3-2}$$

人體脈搏訊號約為 1~2Hz 左右，藉由調整 Sallen-Key 高低通濾波器電路中，電阻與電容比例以使其高頻截止頻率為 3Hz，低頻截止頻率為 1Hz。

(1)低通濾波器

經過儀表差動放大電路之脈搏訊號，首先經低通濾波器以濾除訊號中高頻雜訊部份。為了驗證所設計之低通濾波器確實能濾除訊號中高頻雜訊部份，我們先以 OrCAD Pspice 模擬軟體建立低通濾波電路，並以 OrCAD Pspice 電路模擬功能驗證所設計之低通濾波器的可行性。圖 3-7 為用 OrCAD Pspice 軟體建立之 Sallen-Key Low-Pass 濾波器，將圖中電阻與電容代入上式，可得到

$$fc = f_H = f_L = \frac{1}{2\pi\sqrt{350k \times 350k \times 0.22u \times 0.1u}} = 3.066 \cong 3Hz \tag{3-3}$$

圖 3-8 為利用 PSpice 電路模擬功能，模擬低通濾波器頻率響應，由圖中可觀察到，當訊號經過 3Hz 之後，其增益開始急速下降，因此可濾除訊號中高於 3Hz 成份，完成高頻濾波效果。

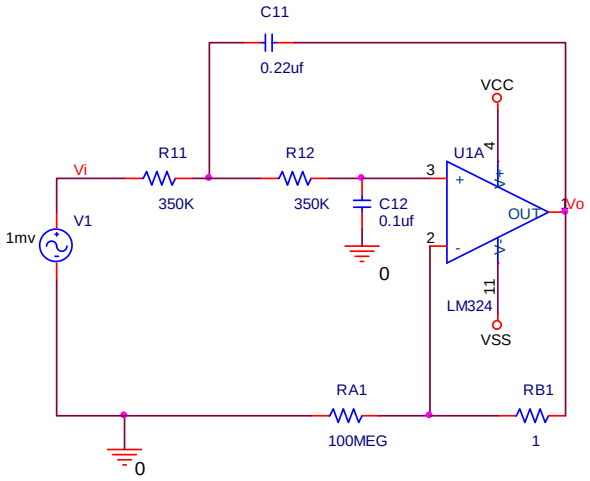


**圖 3-7 Sallen-Ken 低通濾波電路**

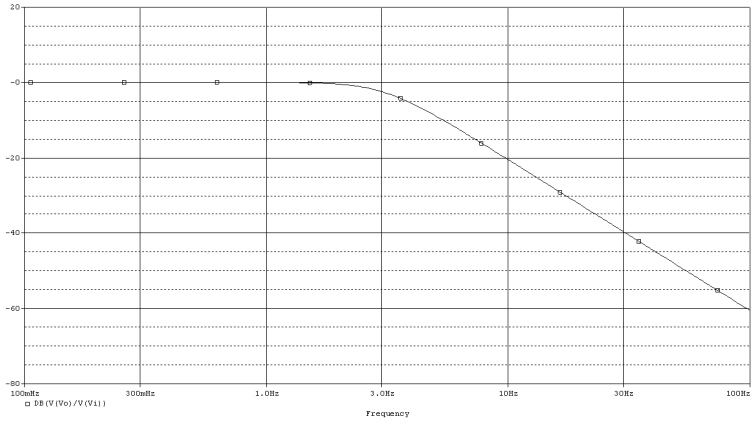


**圖 3-8 OrCAD Pspice 模擬 Sallen-Key 低通濾波器頻率響應圖**

(2) 高通濾波器

訊號經過低通濾波器後，仍含有高頻雜訊成份，因此需再加以高通濾波器以濾除低頻雜訊部份。同樣的，以 OrCAD Pspice 建立模擬電路，並以 OrCAD Pspice 電路模擬功能驗證電路可行性。

圖 3-9 為用 OrCAD Pspice 軟體建立之 Sallen-Key High-Pass 濾波器，將圖中電阻與電容代入式子 $fc = f_H = f_L = \frac{1}{2\pi\sqrt{R_{11}R_{12}C_{11}C_{12}}}$，可得到

$$fc = f_H = f_L = \frac{1}{2\pi\sqrt{150k \times 150k \times 1u \times 1u}} = 1.06 \cong 1Hz \qquad (3\text{-}4)$$

圖 3-10 為利用 OrCAD PSpice 電路模擬功能，模擬高通濾波器頻率響應，由圖中可觀察到，訊號在 1Hz 之前，其增益急速下降，因此可濾除訊

號中低於 1Hz 成份，完成低頻濾波效果。

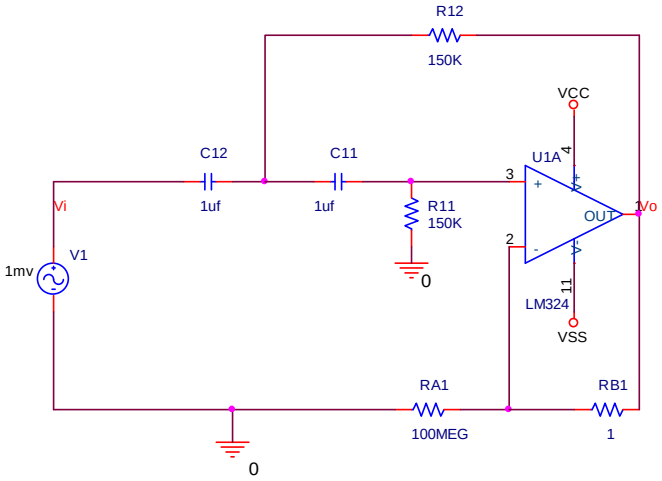


**圖 3-9 Sallen-Key 高通濾波電路**

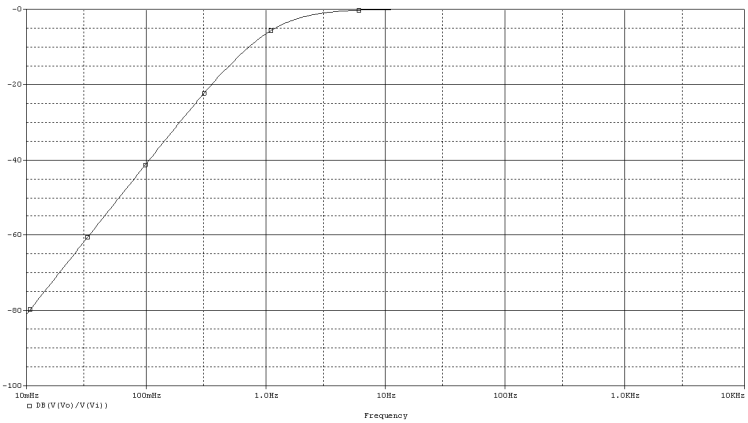


**圖 3-10 OrCAD Pspice 模擬 Sallen-Key 高通濾波器頻率響應圖**

### 3.3.3 電壓箝位電路

經由高低通濾波器之脈搏訊號仍為一類比訊號，因此必須採用類比數位轉換電路，將類比訊號轉為數位訊號，但因為所量測之脈搏生理訊號具有負的電壓成分，而一般之 A/D 轉換 IC 無法轉換負的電壓，所以為了使得負的電壓成份也能順利的轉為數位訊號，因此採用電壓箝位電路將訊號準位提升，使訊號整體皆位於零準位以上，以使 A/D 轉換 IC 亦可轉換原為負成分之電壓訊號。圖 3-11 為利用 OPA 所設計之電壓箝位電路[51]，可經由微調可變電阻 R19，適當的提升輸入訊號 Vin 準位。在本研究實驗中，利用電壓箝位電路將電壓準位提升 2.5V。

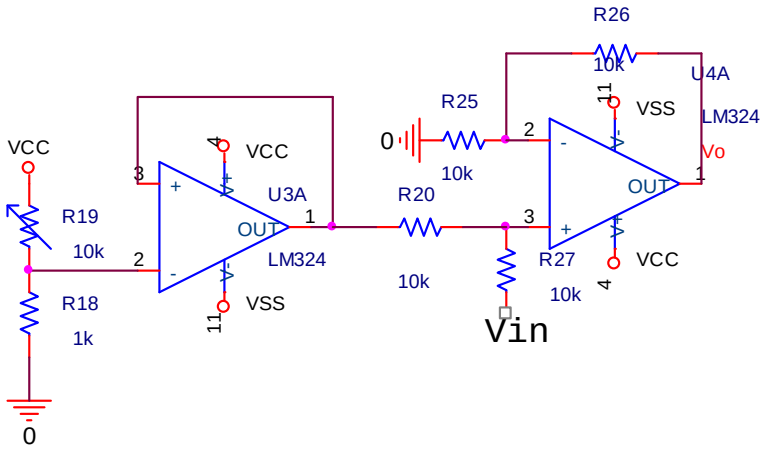


**圖 3-11 電壓箝位電路**

### 3.3.4 A/D 轉換電路與 8051 單晶片實現串列傳輸

經由電壓箝位電路所輸出之脈搏訊號仍為類比訊號，因此須將此類比訊號經過 A/D 轉換電路轉為數位訊號，以利於資訊傳輸至微處理器。而為了使得轉換後之數位訊號得以被嵌入式系統發展平台所接收，需設計某一傳輸介面以傳輸資料。在本研究中，採用串列傳輸介面作為與嵌入式系統發展平台之傳輸介面。

本研究在類比轉數位設計上，採用 ADC0804 此顆類比數位轉換 IC，其解析度為 5V/255，經此 IC 將類比訊號轉為數位訊號後傳進 8051 之中；採用 8051 單晶片實作串列傳輸介面，以將轉換後之數位脈搏訊號傳送至嵌入式系統發展平台 COM 埠上。圖 3-12 為使用 8051 單晶片微電腦及 ADC0804 類比數位轉換 IC，實作之 A/D 轉換電路及串列傳輸介面，圖 3-13 為其接線圖。

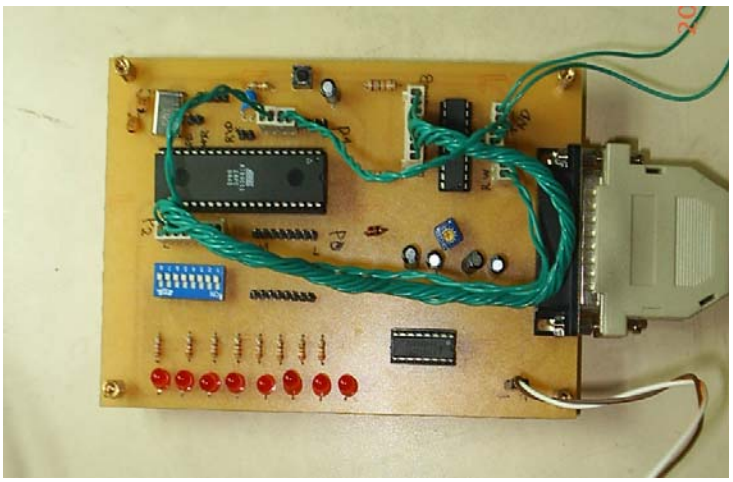

圖 3-12 A/D 轉換電路與單晶片串列傳輸電路

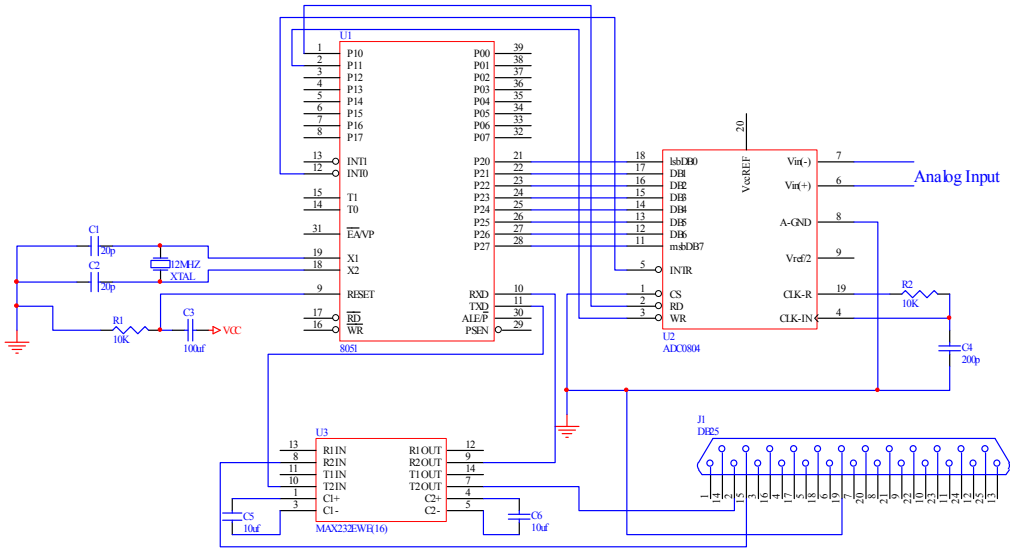


圖 3-13 A/D 轉換電路與單晶片串列傳輸電路接線圖

## 3.4 Linux 串列傳輸設定

本研究提出一以嵌入式系統晶片實現可攜式脈搏生理訊號監測系統，經由量測電路所得到之脈搏訊號經過 A/D 轉換後，透過 8051 單晶片以串列傳輸方式，將脈搏訊號傳輸進嵌入式發展平台之中，接著利用嵌入式發展平台上之無線網路，將脈搏訊號資料傳送至遠端伺服器資料庫之中。經由前節介紹，本研究以 Linux 作業系統為開發環境，但在 Linux 作業系統

上，串列傳輸協定並無法像 Windows 作業系統那樣簡便設定，而需自行撰寫串列傳輸協定。

為了在 Linux 系統下完成串列傳輸協定，我們採用 Unix 所提供但亦可在 Linux 下使用之 termio 結構，此結構如下所示：

```
#include<termios.h>
Struct termio
{
    unsigned short c_iflag
    unsigned short c_oflag;
    unsigned short c_cflag;
    unsigned short c_lflag;
    unsigned char c_cline;
    unsigned char c_cc[NCC];
}
```

在此 termio 結構當中最為重要的為 c_cflag 結構成員，我們透過 c_cflag 結構成員設置傳輸鮑率、數據位、停止位、奇偶校驗位等。另外也需設定 c_iflag 和 c_cc 結構成員，以實現串列傳輸介面。底下以表 3-2、表 3-3、表 3-4 分述 termio 結構中 c_cflag、c_iflag 和 c_cc 結構成員於實現串列傳輸介面需設定項目。

**表 3-2 c_cflag 結構成員設定**

| | |
|---|---|
| CLOCAL | 設定本地連接 |
| CREAD | 致能接收字元 |
| BAUDRATE | 設定鮑率 |
| CSTOPB | 設定停止位數 |
| PARENB | 致能校驗 |
| PARODD | 使用奇效驗而不使用偶校驗 |
| CSIZE | 數據位的位煙碼 |
| CRTSCTS | 輸出資料硬體流量控制 |
| CS8 | 設定 8 位元數據 |

**表 3-3 c_iflag 結構成員設定**

| | |
|---|---|
| IXON | 啟動輸出硬體流量控制 |
| INPCK | 致能奇偶校驗 |
| IGNPAR | 忽略奇偶校驗錯誤 |

**表 3-4 c_cc 結構成員設定**

| | |
|---|---|
| VMIN | 指定最少讀取字元數 |
| VTIME | 指定最少讀取每一字元等待時間 |
| VINTR | 中斷控制 |

### 3.5 脈搏訊號於網頁呈現

本研究規劃將所量測之人體生理訊號顯示於網頁上，以使得醫護端可從遠處經由網路存取網頁以觀察受測者目前生理狀況。在所規劃之顯示頁面上，除了即時顯示所量測到之脈搏訊號波形外，還包括量測者之心跳頻率；而所設計之脈搏顯示頁面，脈搏訊號圖形會根據所量測到之脈搏訊號數據，以類似示波器顯示方式做動態之變化，同時心跳頻率也根據數據作動態變更。

在心跳頻率計算上，本論文採用 Pan J, Tompkins, WJ 所提之" A real-time QRS detection algorithm"演算法[41]，而之所以採用此演算法乃是考量脈搏訊號類似於心電訊號。圖 3-14 為心跳頻率計算流程。底下詳述其中步驟：

(1)微分：

目的為取出脈搏生理訊號波形的變化特徵，亦即計算波形的斜率變化，由於脈搏生理訊號在波形最高點其斜率變化較為明顯，所以能藉此運算取出脈搏訊號最高點即前一章脈搏波形圖形之 F 點。

(2)平方：

平方過後以使小於零的資料皆大於零，以利於後續資料處理，除此之外，平方過後使得資料之間的差異更為明顯，更利於求出脈搏圖形之 F 點。

(3)時間平均濾波器：

透過平方處理後，信號在波型部分出現鋸齒成份，對於偵測波形 F 點會造成一定程度的影響。使用時間平均濾波器讓 F 點波形更完整的呈現。

(4)取最高點三分之二處做為判斷 F 點基準

在本研究中，脈搏生理訊號經過以上三個步驟之後，設定圖形中最高點三分之二處做為判斷 F 點基準，以決定圖形 F 點。

(5)找出高於基準之點

以(4)步驟所決定之基準點比對圖形數值，當數值超過此基準點時即為一 F 點，以此找出所有圖形 F 點。

(6)找出斜率轉折點

由(5)步驟找出圖形之所有 F 點之後，以斜率正負變化決定 F 點位於圖形之位置。

(7)計算轉折點間隔時間

由(6)步驟找出其轉折點後，計算轉折點之間時間間隔，以計算心跳頻率。

(8)得出心跳頻率

由(7)步驟得到轉折點之間時間間隔之後，可經由帶入式子(3.5.1)得到心跳頻率。

$$f_h = \frac{1}{td} \times 60 \tag{3-5}$$

其中 $f_h$ 為心跳頻率，td 為轉折點間隔時間。

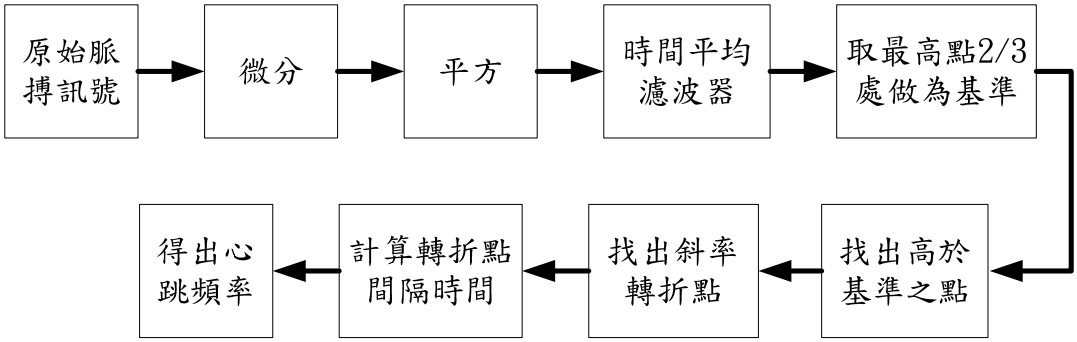


**圖 3-14 心跳頻率計算流程圖**

### 3.5 時間序列相似度比較

本研究所提出之架構，在最後端我們加入一資料探勘技術以分析人體生理訊號，期望找出可能蘊含於生理訊號中之病徵。由於生理訊號屬於一時間序列訊號，整個資料量相當龐大，要如何去縮減此龐大的資料量，但仍能保留原始數據特性，是一個重要的考量條件，另外，設計一效能佳的資料探勘演算法是另一個重點。

#### 3.5.1 資料維度縮減

首先針對龐大的時間序列資料量，本研究使用符號轉換的方式縮減資料量方面維度，其主要概念為將原本數值型態的時間序列，依其數值作適當切割，並以字元符號取代每一個切割等分，將原本數值型態之資料，轉為字串符號型態，以減低時間序列之資料維度，接著利用目前現有之處理字串演算法，以分析轉換後之時間生理訊號。

將數值型態的資料轉換成符號型態的資料，最主要是為了達到維度縮減，但這樣的方式往往造成資料的完整性不足。但若直接對數值型態的資料做分析，這樣的方式需要耗費極高的處理時間。因此，若對數值資料直接做分析，在面對資料量迅速增加的情況就會變得難以負荷。所以為了簡化時間上之複雜度，又同時保留資料原本特性，在符號轉換的過程中加入了角度轉換處理。

(1)角度轉換

假設有一時間序列向量 S，我們首先將原本的時間序列向量 S 轉換成 S’；S 代表原本的時間序列向量，而 T 代表著時間序列紀錄的時間點，如下所示：

$$\begin{aligned} S &= s_1 s_2 s_3 ....s_{l(S)} \\ T &= t_1 t_2 t_3 ....t_{l(S)} \\ S' &= s_1^{'} s_2^{'} s_3^{'} ....s_{1(S)-1}^{'} \\ S_i^{'} &= \frac{S_{i+1} - S_i}{t_{i+1} - t_i}, 1 \le i \le l(S) - 1 \end{aligned} \tag{3-6}$$

經由式子(3-6)將原本的時間序列轉換成 S’，這樣的轉換是希望保留住原始序列上升及下降的的變化，即保留序列原始特性。經由轉換的方式，原有的時間序列資料改變為只紀錄其區段的序列變化情形，再配合後續的的符號轉換，即能保存原始序列特性並達到資料維度縮減，以達到效率的改進。

(2)符號轉換

角度轉換目的在於轉換後之資料保有原始資料特性；而符號轉換則是希望減少原始資料維度，增進其處理效能。雖然採用符號轉換可能導致轉換後之資料完整性不足，但此點我們可在資料完整性與效率上取一平衡點，當我們所需較大之資料完整性時，我們可對原始資料作較小區域的切割，以更多的符號取代原始數值資料，提高轉換後對原始數據資料的完整性；相反的，若需要較佳之處理效能，則減少取代數值資料之符號數量，此點可依據應用情況而定。

符號轉換其步驟如下：

a.首先利用(z-score)正規化法

$$v_i^{'} = \frac{v_i - \bar{v}}{\sigma^{'}} \tag{3-7}$$

$\bar{v}$為所有時間序列平均值，$\sigma$為所有時間序列標準差。

採用式子(3-7)z 分數正規化法，主要目的是將不同標準之下所記錄的資料轉換到同一個標準，使得資料之間具有可比較性，以便提高分析時的準確度。資料的正規化會將資料重新分佈在一個較小而且特定的範圍內。經由 z 分數正規化後之資料，其整體平均值(Mean)為零，標準差(Standard Deviation) 為 1，使整體數值呈現一常態分佈形式(Normal Distribution)，當我們決定採用多少符號量取代掉原始數值資料時，便可採用表 3-5 決定不同符號間之分割點，而等分了整個常態分佈。

**表 3-5 常態分布之切點**

| SN | Cut Point |
|---|---|
| 3 | (-0.43,0.43) |
| 4 | (-0.67,0,0.67) |
| 5 | (-0.84,-0.25,0.25,0.84) |
| 6 | (-0.97,-0.43,0,0.43,0.97) |
| 7 | (-1.07,-0.57,-0.18,0.18,-0.57,-1.07) |
| 8 | (-1.15,-0.67,-0.32,-0.14,0.14,0.32,0.67,1.15) |
| 9 | (-1.22,-0.76,-0.43,-0.14,0.14,0.43,0.76,1.22) |
| 10 | (-1.28,-0.84,-0.52,-0.25,0,0.25,0.52,0.84,1.28) |

b.接下來利用上表切點，將原本的角度取代為符號。如圖 3-15 所示，假設我們希望以 5 個符號數取代原始數值序列其角度變化，經由上表可知，其切點分別為：(-0.84)、(-0.25)、(0.25)、(0.84)，因此將整體常態分佈分割為 5 個區域，分別以 ABCDE 代表此五個區域，則原本的角度變化會轉為 DEAAEDBDBBD 此字串。

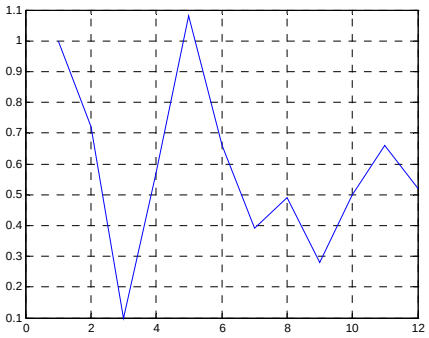


**圖 3-15 時間序列角度變化**

利用這樣的符號轉換，我們可將所有的時間序列資料轉換成以符號代表，接著利用現存有效率的字串演算法，以計算出序列間相似度。

### 3.5.2 餘弦相似度(Cosine Similarity Measure)和漢明距離(Hamming Distance)相似度演算法

前面我們已透過相關演算法，將原本為數值型之時間序列資料轉為符號型之時間序列，本研究中採餘弦相似度及漢明距離，以計算兩符號時間序列之間相似度，底下詳述之：

a. 餘弦相似度演算法(Cosine Similarity Measure)

餘弦相似度（cosine similarity）乃是傳統文件分類中，最常被拿來度量文件間距離的基本度量方法，其以兩個 d 維向量間的角度差異來度量該向量間的距離，所得數據介於 0 ~ 1 之間，當兩向量角度越相近時，所求出的餘弦距離越接近 1；反之，則越接近 0。假設在 d 維空間中有兩點 $a=[a_1,a_2,\ldots, a_d]$，$b=[b_1, b_2,\ldots, b_d]$則其餘弦相似度可表示為式子(3.5.4)。

$$Sim(a,b)=\frac{a\cdot b}{|a||b|} \tag{3-8}$$

底下以一範例說明，使用餘弦相似度計算字串間之相似度：

假設已利用符號轉換轉換出兩字串，分別為：

a=[A,B,C,D,E] 、 b=[A,B,D,E,F] ， 首 先 取 兩 序 列 聯 集 ， 可 得 到 Uab=[A,B,C,D,E,F] ，接著分別以此聯集分別比對 a,b 字串，得到 Ua=(1,1,1,1,1,0)、Ub=(1,1,0,1,1,1)，最後將 Ua 與 Ub 帶進上式中，求出 a 與 b 相似度如下所示：

$$Sim(a,b)=\frac{\vec{Ua}\cdot\vec{Ub}}{\left|\vec{Ua}\right|\left|\vec{Ub}\right|}=\frac{4}{\sqrt{5}\cdot\sqrt{5}}=0.8 \tag{3-9}$$

b. 漢明距離(Hamming Distance)

漢明距離是以理查德・衛斯裡・漢明的名字命名的。在信息論中，兩個等長字符串之間的漢明距離是兩個字符串對應位置的不同字符的個數。換句話說，它就是將一個字符串變換成另外一個字符串所需要替換的字符個數。底下以幾個例子說明漢明距離大致上概念：

1011101 與 1001001 之間的漢明距離是 2。

2143896 與 2233796 之間的漢明距離是 3。

"toned" 與 "roses" 之間的漢明距離是 3。

c. 餘弦相似度(Cosine Similarity Measure)和漢明距離(Hamming Distance)相似度演算法

在前一小節我們利用餘弦相似度去計算字串之間相似度，但此法會有下列之所舉之情況發生。

當兩字串序列分別為 a=[A,B,C,D,E]、b=[E,D,C,B,A]，取兩序列之聯集 Uab=[A,B,C,D,E]，接著以此聯集比對 a,b 字串，得到 Ua=(1,1,1,1,1)、Ub=(1,1,1,1,1)，之後得到兩字串序列之間相似度為：

$$Sim(a,b)=\frac{\vec{Ua}\cdot\vec{Ub}}{\left|\vec{Ua}\right|\left|\vec{Ub}\right|}=\frac{5}{\sqrt{5}\cdot\sqrt{5}}=1 \tag{3-10}$$

由以上例子可以看出當兩字串序列具有相同成分時，使用餘弦相似度

計算時，兩字串序列其相似度為 1；但對於一時間序列資料來說，這樣的結果顯然是不正確的。因此，在字串的相似度比較上，本研究採用結合餘弦相似度演算法及漢明距離兩種架構之演算法[42]，計算字串序列間之相似度。其演算法架構為：

$$Hsim(A,B) = (1 - \frac{H_{AB}}{Max(H)}) \times Sim(A,B) \qquad (3\text{-}11)$$

其中 $Max(H_k) = Max(H_{k-1}) + 2 \times (k \setminus 2), k = 1,2,3...$，$H_{AB} = \sum_{i=1}^{m} \left| L(T_{Ai}) - L(T_{Bi}) \right|$，$H_0$ 為 0，k 為參考序列之長度。

底下以此架構再次計算上個例子，得到：

$H_{ab} = 4+2+0+2+4 = 12$、$\mathrm{Max}(H_5) = 12$

$$Hsim(A,B) = (1 - \frac{12}{12}) \times 1 = 0$$

由上結果可知，所採用之結合餘弦相似度演算法及漢明距離兩種架構之相似度演算法，計算兩字串之相似度時，可得到一較合理之結果。