

高雄縣高英高級工商職業學校  
Kao Ying Industrial Commercial Vocational High School

專題製作報告



沉默鬧鐘

學生姓名：楊 鈞 富

洪 一 忠

劉 柏 亨

指導老師：葉 忠 賢 老師

中 華 民 國 102 年 05 月



## 誌 謝

進入高中職業學校開始學習技藝課程，面對不同的技能專長領域，除了艱辛，更覺漫長。如今，我們小組已來到了三年級，回頭俯看這一路上的學習歷程，細數點滴。我想，是我們該將每個階段的感動，留下紀錄的時候了。

感謝指導老師也是導師葉忠賢老師在這三年的高職學習生涯中，引導我們朝最適切的學習方向來邁進，更謝謝老師的指導，讓我們可以重新發現自己長久以來的不足與缺點，學習研究過程中瞭解、信任與師生之間的情感，如此的感受，對我們小組而言，更是彌足珍貴。

感謝班上同窗好友及小組們的互相協助，在電路設計、實作與各種電腦工具應用上真的幫了我們相當多的忙，回想剛入學的我們，對諸多情事都一竅不通，幸有你們大家的打氣及協助，使讓我們彼此有勇氣邁向這未知的學習旅程，而沒有半途而廢，真的是非常謝謝你們，有這三年同學情誼的陪伴，令我們小組同學內心的感動真是言語所無法形容。

楊鈞富、洪一忠、劉柏亨 謹上 2013/05

# 沉默鬧鐘

## 摘要

本篇研究報告旨在透過單晶片 89C51 的學習，了解單晶片的功能及使用方法，且經由實際製作 PCB 電路的過程中去對單晶片運作有更深入的了解。會想製作沉默鬧鐘，故想要藉由設計一單晶片電路，由於鬧鐘的聲音分貝太大聲，以致我們小組想要做出一個不會吵到別人又可以叫醒自己的鬧鐘，我們與老師討論後，決定以震動來達到此效果，所以我們因此就裝上了震動馬達，裝置在枕頭上。

一方面學校規定三年級學生必須在畢業前完成專題一方面想知道三年來到底學會了多少的專業技術以及知識，另一方面想測試自己對於資訊科這個科系是否抱持著剛學習的態度以及熱忱。由於這個電路有一定的難度，加上我們只做過電子時鐘，所以這個作品我們特別想去嘗試做做看。

**關鍵詞：**單晶片、震動馬達、鬧鐘

# 目 錄

誌謝 .....	I
摘要 .....	II
目錄 .....	III
表目錄 .....	IV
圖目錄 .....	V
壹、前言 .....	1
一、製作動機 .....	1
二、製作目的 .....	1
三、製作架構 .....	1
四、製作預期成效 .....	3
貳、理論探討 .....	4
參、專題製作 .....	16
一、設備及器材 .....	16
二、製作方法與步驟 .....	16
三、專題製作 .....	18
肆、製作成果 .....	22
伍、結論與建議 .....	24
一、結論 .....	24
二、建議 .....	24
參考文獻 .....	26
附錄一 沉默鬧鐘之程式碼 .....	27

## 表目錄

表 3-1-1 專題製作使用儀器（軟體）設備一覽表 .....	16
表 3-3-1 專題製作計畫書 .....	18
表 3-3-2 沉默鬧鐘之材料表 .....	21

## 圖目錄

圖 1-3-1 專題製作流程圖 .....	2
圖 2-1-1 液晶顯示器構造圖 .....	4
圖 2-1-2 蜂鳴器外觀圖 .....	5
圖 2-1-3 震動馬達外觀圖 .....	6
圖 2-1-4 震動馬達內部電路構造圖 .....	6
圖 2-1-5 電晶體接腳圖 .....	7
圖 2-2-1 微電腦硬體介面結構圖 .....	9
圖 2-2-2 單晶片的內部結構圖 .....	12
圖 2-2-3 單晶片 8051 的接腳圖 .....	12
圖 2-2-4 單晶片埠 0 應用於 I/O 時的提升電路圖 .....	13
圖 3-2-1 製作方法與步驟 .....	16
圖 3-3-1 此專題單晶片電路圖 .....	19
圖 3-3-2 此專題電晶體接喇叭、震動馬達電路圖 .....	19
圖 3-3-3 此專題按鈕與 LCD 電路圖 .....	19
圖 3-3-4 沉默鬧鐘之電路板 Layout 圖 .....	20
圖 4-1-1 沉默鬧鐘麵包板測試 .....	22
圖 4-1-2 沉默鬧鐘成品圖 .....	22
圖 4-1-3 沉默鬧鐘完整圖 .....	22
圖 4-1-4 打專題報告 .....	22
圖 4-1-5 成品測試(1).....	23
圖 4-1-6 成品測試(2).....	23





# 壹、前言

## 一、製作動機

大約三千年以前人們開始有了時間觀念，埃及的城鎮上就有方尖碑和日規做為測定時間的日晷，而西元 15 世紀希臘人發明水鐘等等。而現在每戶人家裡至少都有一個時鐘一個鬧鐘甚至手上還有手錶，而鬧鐘是防止睡過頭而設計的，但還是有人會賴床所以現在市面上出現了會跑的鬧鐘，要去找出鬧鐘才能按下停止鍵但是鬧鐘有個缺點就是聲響太大，如果在外面的話尤其是上班族根本不可能帶著鬧鐘又怕睡午覺會睡過頭，因此我們小組想出了震動時鐘既不會吵到別人更不會因此睡過頭，更適合在辦公室中使用不會吵到同事又不會睡過頭因而被主管責罵。

## 二、製作目的

我們小組一再地經過討論與表決後決定將沉默鬧鐘擬定我們小組的專題題目，一方面學校規定三年級學生必須在畢業前完成專題一方面想知道三年來到底學會了多少的專業技術以及知識，另一方面想測試自己對於資訊科這個科系是否抱持著剛學習的態度以及熱忱。由於這個電路有一定的難度，加上我們只做過電子時鐘，所以這個作品我們特別想去嘗試做做看，因為這個作品是可切換成震動或鬧鈴，再加上有顆 LED 與 LCD，LCD 是我們第一次接觸它完全不了解，因為我們之前所做的都是七段式顯示器。

我們小組在製作之前，決定先將有關這個作品的理論以及相關知識，所以我們從網路上以及書籍開始找尋相關資料以及知識，再與指導老師討論與實作。

## 三、製作架構

### (一) 專題製作流程

在經小組一再地問題討論及溝通之後，我們小組終於決定了要來製作沉默鬧鐘。題目確定後，我們小組便開始構想如何去完成這次的專題，開始製作專題報告資料等等，製作沉默鬧鐘首先需要購買材料如：8051 單晶片等等材料。再到麵包板上製作，在製作麵包板時卻出現了一個問題那就是 LCD 顯示器沒辦法插在麵包板上，只好先使用 OK 線，麵包板完成測試後沒有問題後再畫出 Layout 圖，確認無誤後再焊接到 PC 板上，如發現錯

誤等等問題時，即會與教師進行討論及修改。

(二)專題流程圖

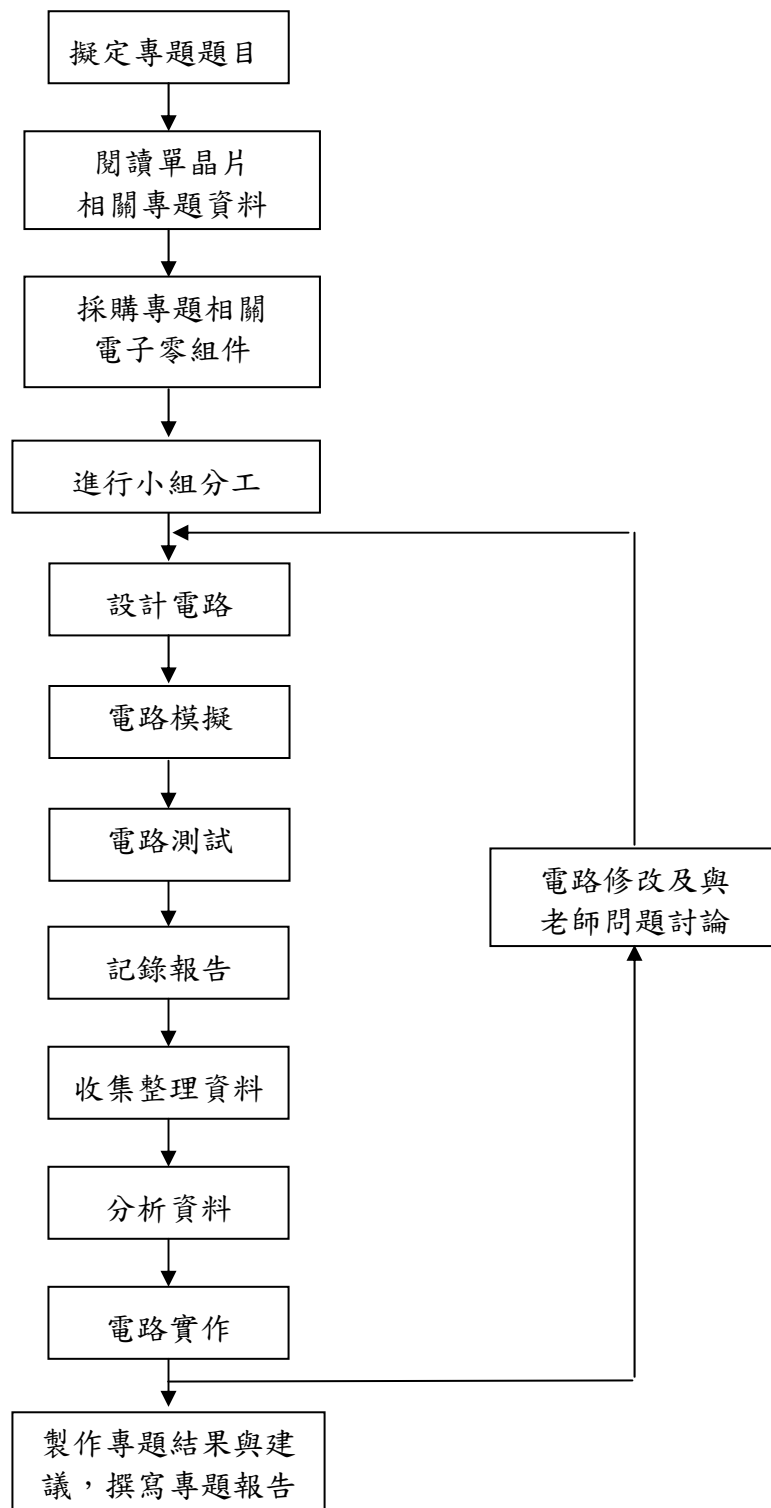


圖 1-3-1 專題製作流程圖

#### 四、製作預期成效

我們小組要開始製作這個作品時可說是非常艱難，因為三人全毫無頭緒，也當然非常擔心是否能夠完成這次的專題，像個無頭蒼蠅般似的上網亂找一通，好險有個指導老師把我們拉到軌道上，願我們將專題定義為：

- (1)透過組合語言編寫，讓沉默鬧鐘順利進行運轉。
- (2)LCD 顯示器可顯示「時：分：秒」。
- (3)經由震動可將使用者叫醒。
- (4)經由鬧鈴叫聲可將使用使叫醒。
- (5)可設定是由震動叫醒或鬧鈴叫醒或兩者一起動作。
- (6)此專題共有 5 個按鈕，分別為重製鈕、設定時間鈕(K1)、顯示震動(鬧鈴)時間(K2)、設定震動(鬧鈴)時間(K3)，K4 為設定完成與停止震動(鬧鈴)。
- (7)LED 為每秒閃爍一次，如開始震動(鬧鈴)就一直閃爍。

## 貳、理論探討

### 一、電子相關零組件

#### (一) LCD 液晶顯示器

液晶顯示器（英語：Liquid Crystal Display，縮寫：LCD）為平面薄型的顯示裝置，由一定數量的彩色或黑白畫素組成，放置於光源或者反射面前方。適用於使用電池的電子裝置。液晶顯示器的每個畫素由以下幾個部分構成：懸浮於兩個透明電極（氧化銦錫）間的一列液晶分子層，兩邊外側有兩個偏振方向互相垂直的偏振過濾片。如果沒有電極間的液晶，光透過其中一個偏振過濾片其偏振方向將和第二個偏振片完全垂直，因此被完全阻擋了。但是如果透過一個偏振過濾片的光線偏振方向被液晶旋轉，那麼它就可以透過另一個偏振過濾片。液晶對光線偏振方向的旋轉可以透過靜電場控制，從而實作對光的控制。液晶分子極易受外加電場的影響而產生感應電荷。將少量的電荷加到每個畫素或者子畫素的透明電極產生靜電場，則液晶的分子將被此靜電場誘發感應電荷並產生靜電扭力，而使液晶分子原本的旋轉排列產生變化，因此也改變透過光線的旋轉幅度。改變一定的角度，從而能夠透過偏振過濾片。一、

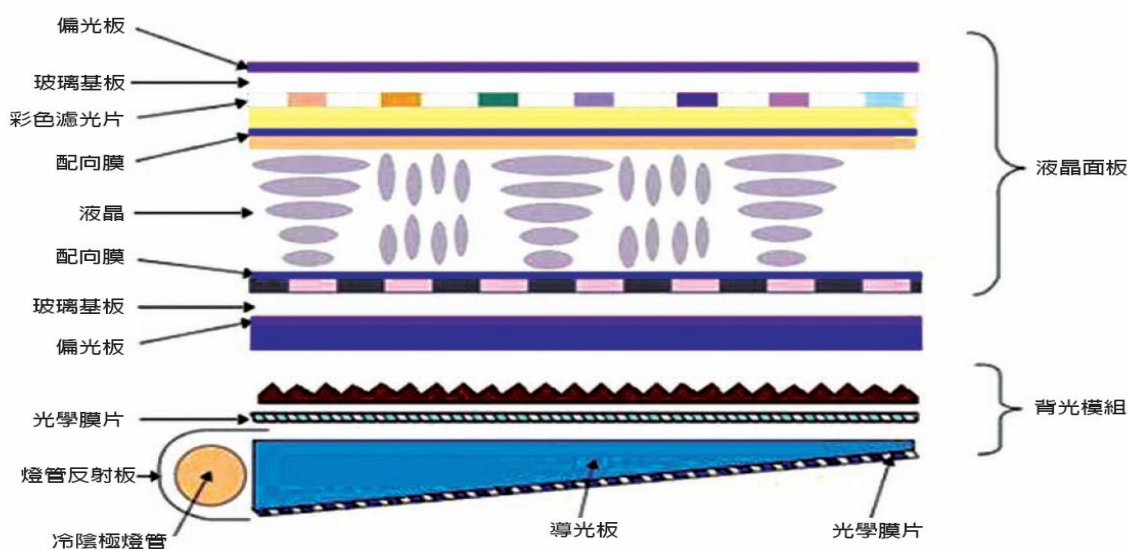


圖 2-1-1 液晶顯示器構造圖

## (二) 蜂鳴器

蜂鳴器是一種一體化結構的電子訊響器，採用直流電壓供電，廣泛應用於計算機、打印機、複印機、報警器、電子玩具、汽車電子設備、電話、定時器等電子產品中作發聲器件。

壓電式蜂鳴器主要由多諧振盪器、壓電蜂鳴片、阻抗匹配器及共鳴箱、外殼等組成。有的壓電式蜂鳴器外殼上還裝有發光二極管。多諧振盪器由晶體管或集成電路構成。當接通電源後（1.5~15V 直流工作電壓），壓電蜂鳴片由銦鈦酸鉛或鈮鎂酸鉛壓電陶瓷材料製成。在陶瓷片的兩面鍍上銀電極，經極化和老化處理後，再與黃銅片或不銹鋼片粘在一起。

電磁式蜂鳴器由振盪器、電磁線圈、磁鐵、振動膜片及外殼等組成。接通電源後，振盪器產生的音頻信號電流通過電磁線圈，使電磁線圈產生磁場。振動膜片在電磁線圈和磁鐵的相互作用下，週期性地振動發聲。



圖 2-1-2 蜂鳴器外觀圖

## (三) 震動馬達

多數振動馬達由一個較小的電動馬達驅動一個偏心配重，馬達分為直流有刷或無刷電機，有兩種基本的結構：硬幣狀(扁平)與圓柱形(條狀)。柱型馬達是採用傳統軸向設計的簡單有刷電機，用於多種應用中，但由於體積較大，不適合在手機中應用。柱型馬達在手機中需要較大的空間，直徑是所有振動馬達中最大的。在電刷切換馬達線圈的電流時，所有的有刷電機在換向點會產生電火花。這些電火花是寬帶射頻干擾(RFI)的最佳發射源。電刷會發生磨損，這是馬達故障的最主要原因。馬達外部為工程塑料製成的外殼，內部除

了外盒外，還有一個微小的直流電動機，驅動偏心輪轉動。此外還有一塊很簡單的集成電路，用以控制啟動和停止電動機的運轉，控制電路接通。馬達軸上面有一個偏心輪，當馬達轉動的時候，偏心輪的圓心質點不在電機的轉心上，使得馬達處於不斷的失去平衡狀態，由於慣性作用引起震動。



圖 2-1-3 震動馬達外觀圖

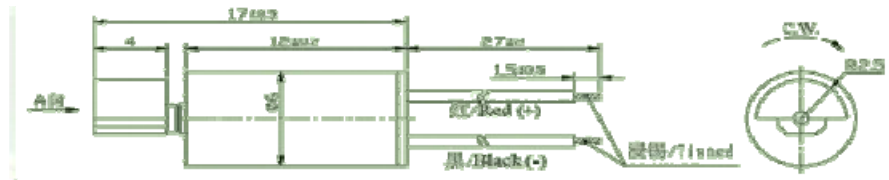


圖 2-1-4 震動馬達內部電路構造圖

#### (四)電晶體

「電晶體(transistor)」是一種半導體元件，也是被使用於電子開關的電子零件，它由三個 N 與 P 型半導體材料所構成，外形上有三個接腳，分別是射極(emitter)、基極(base)、與集極(collector)，有 NPN 與 PNP 兩種基本類型，功能差別在於電流方向，下圖是小功率電晶體的外觀與表示符號。電晶體最主要的功能是放大電流訊號，當基極到射極之間有微量電流導通時，會觸發集極到射極之間的大電流。以下分別對利用 NPN 與 PNP 電晶體常應用到的電子開關電路做說明。因著雙極性電晶體具有兩個 P-N 接面，它們分別是：射極接面和集極接面，而這兩個接面可外接順向偏壓或逆向偏壓，如此便有四種不同的排列組合，這也是電晶體的四種工作模式，它們列於下表中。一般電晶體很少工作在反向模式。在數位式的電子系統中，也就是在裝置的內部是以二進位方式運作的電子電路，電晶體工作在截止和飽和模式，在此兩種模式下，電晶體可視為開關，不是開（二進位系統的 0）便是關（二進位系統的 1）。最後，若是電晶體作放大信號用，則工作在線性模式。

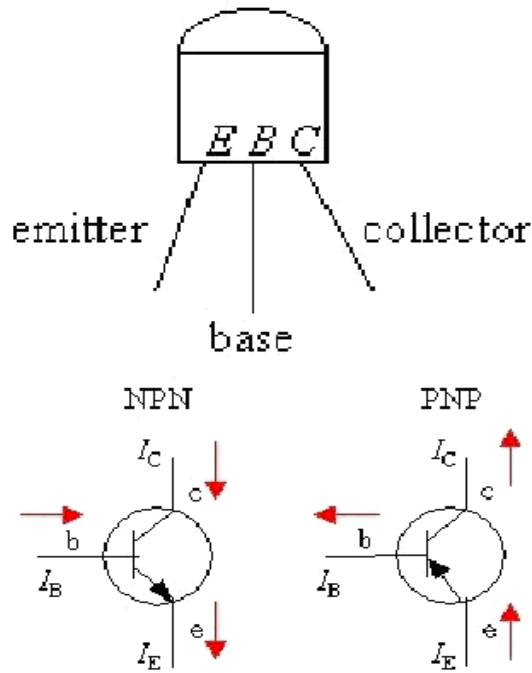


圖 2-1-5 電晶體接腳圖

## 二、單晶片微處理機

### (一) 單晶片微處理機的簡介

一個微電腦需包含微處理器(CPU)，存程式指令(ROM)及存取的資料的 RAM，輸入/輸出埠(I/O 埠)及時脈、計數器、中斷系統等。它們經由位址匯流排(Address Bus)、資料匯流排(Data Bus)和控制匯流排的连接，及透過輸入/輸出埠與週邊裝置連線，構成為電腦系統。由於單晶片微處理機是把為電腦的主要元件製造在一塊晶片上，所以可以把單晶片微處理機看成是一個不帶週邊裝置的微電腦。

單晶片微處理機具有以下特點：

- 1.受密度限制：晶片內記憶容量較小，ROM 小於 64K，RAM 小於 1K。
- 2.可靠性良好：單晶片是依工業控制的要求所設計的，其抗工業雜訊干擾優於一般的 CPU，程式指令及常數資料燒錄在 ROM 內，因其許多訊號通道均在同一個晶片內，故可靠性高。
- 3.易擴充：單晶片具有一般微電腦所需的組件，如三態雙向匯流排，平行及串列的輸入/輸出接腳，可以擴充為各種規模的微電腦系統。
- 4.控制功能強：為了滿足工業控制的要求，單晶片的指令，除了輸入/輸

出控制指令，邏輯判斷指令外，更具有極豐富的條件分歧跳躍指令。

5.看門狗功能：CPU 受雜訊干擾而導致當機是司空見慣的，也是工業界很難接受的，單晶片需具看門狗功能，當機時能自動重新開機，使 CPU 維持正常的運作。

## (二) 單晶片微處理機的應用範圍

- 1.智慧型產品：單晶片與傳統的機械產品相結合，使傳統機械產品結構簡化、控制智慧化、構成新一代的機電整合的產品。例如電打字機採用單晶片，取代近千個機械組件，縫紉機採用單晶片作控制，可執行多功能自動操作、自動調速、控制縫補花樣的選擇。
- 2.智慧型儀表：用單晶片改良原有的測量，控制儀表，能使儀表數位化、智慧化、多功能化、綜合化。而測量儀器中的誤差修正，線性化等問題也可迎刃而解。
- 3.測控系統：用單晶片可以設計各種工業控制系統、環境控制系統、資料控制系統。例如溫室控制、水閘自動控制、電鍍生產線自動控制，及汽輪機電液調節系統。
- 4.數值控制機：在目前數位控制系統的簡易控制機中，採用單晶片可提高其可靠性及增強功能，降低控制機成本。
- 5.智慧型介面：用單晶片進行介面的控制與管理，單晶片與主機平行工作，可大大地提高了系統的執行速度。如在大型資料讀取系統中，用單晶片對 A/D 轉換進行控制不僅可提高讀取速度，還可對資料進行預先處理，如數位濾波、線性化處理與誤差修正等。在通訊界面中使用單晶片可對資料進行編碼解碼、分配管理、接收/發送控制等。

## (三)8051 單晶片

微電腦硬體結構包含中央處理單元、記憶體單元、輸入單元與輸出單元等四個主要單元，其結構關係則如下圖所示。



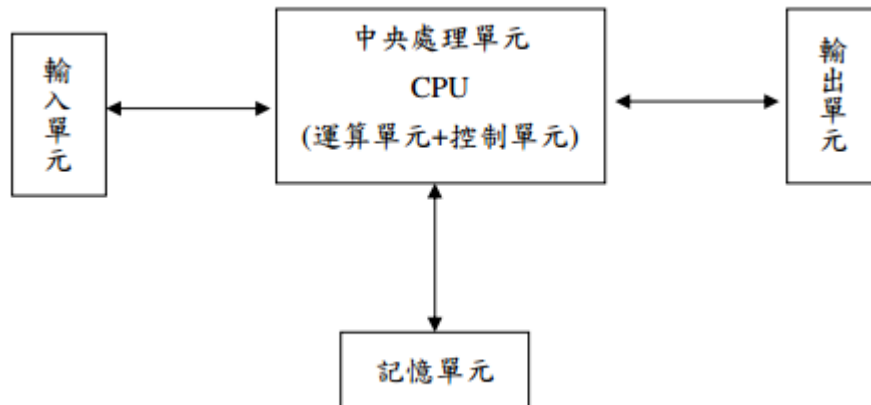


圖 2-2-1 微電腦硬體介面結構圖

其中，中央處理單元則是由運算單元與控制單元兩部分所組成的單元，即是一般所通稱的 CPU(Center Processing Unit)，此為微電腦最重要的部分。以下就微電腦中各單元的功能做簡單介紹：

#### 1. 運算單元(Arithmetic Logic Unit，簡稱 ALU)

運算單元又稱為算數邏輯單元，在中央處理單元中可用於執行算數運算，(如：加、減、乘、除等)，以及邏輯運算(如：AND、OR、NOT 等)，能將記憶體單元或輸入單元送至中央處理單元的資料執行各種運算。當運算完成後再由控制單元將結果資料送至記憶體單元或輸出單元。

#### 2. 控制單元(Control Unit，簡稱 CU)

此單元在中央處理單元中，負責協調與指揮各單元間的資料傳送與運作，使得微電腦可依照指令的要求完成工作。在執行一個指令時，控制單元先予以解碼(Decode)，瞭解指令的動作意義後再執行(Execute)該指令，因此控制單元將指令逐一執行，直到做完整個程式的所有指令為止。

### 3. 輸入單元(Input Unit, 簡稱 IU)

此單元是用以將外部的資訊傳送到 CPU 做運算處理或存入記憶體單元，一般在為電腦的輸入單元有鍵盤、磁碟機、光碟機、滑鼠、光筆、掃描器或讀卡機等週邊設備。

### 4. 輸出單元(Output Unit, 簡稱 OU)

此單元是用以將 CPU 處理過的資料輸出或儲存傳送外部週邊設備，一般在為電腦的輸出單元有顯示器、印表機、繪圖機、燒錄機或磁碟機等週邊設備。

### 5. 記憶體單元(Memory Unit, 簡稱 MU)

記憶體單元是用來儲存輸入單元傳送來的資料，或儲存經過中央處理單元處理完成的資料。記憶體單元之記憶體可分為主記憶體(Main Memory)與輔助記憶體(Auxiliary Memory)兩種，而主記憶體依存取方式不同，又可分為唯讀記憶體(Read Only Memory, 簡稱 ROM)與隨機存取記憶體(Random Access Memory, 簡稱 RAM)。ROM 所儲存的資料，在微電腦中只能被讀出但不能被寫入，也不會因為關機斷電而使資料流失；至於 RAM 在微電腦中，則可被讀出或寫入資料，但在關機斷電後儲存於 RAM 中的資料將會流失。輔助記憶體則是指磁片、硬碟或磁帶等週邊硬體，一般亦為輸出入單元，主要用來彌補主記憶體的不足，其容量可無限制擴充。

#### (四) AT89S51/P89C51 特性

微電腦系統包括 CPU、記憶體(RAM/ROM)及 I/O 介面三大部分，而單晶片微處理器就是把 CPU、記憶體及 I/O 等製作在同一個晶片上，作成體積小、成本低、硬體接線容易及擴充性佳的微電腦控制系統。8051 於 1980 年由 Intel 公司所開發的，迄今已將近三十年，各公司開發相關族系的晶片也很多，不過目前以 ATMEL 的 AT89S51/52 及 PHILIPS 的 P89C51 等兩種族系晶片為主流。其特性如下：

- 8 位元的 CPU。
- 32 條雙向 I/O。
- 128/256bytes 資料記憶體 RAM，可擴充至 64K。(AT89S51/52)

512/1Kbytes 資料記憶體 RAM，可擴充至 64K。(P89C51)

- 2/3 個 16 位元計時器。
- 具全雙工串列埠 UART。
- 6/8 個中斷源。

AT89S51：EX1、TF0、TF1、RI、TI 6 個中斷源。

AT89S52：EX0、EX1、TF0、TF1、RI、TI、TF2、EXF2 8 個中斷源。

- 晶片內具有時脈振盪電路。
- 雙指標暫存器(DPTR)。
- 14 位元看門口計數器(WDT)。
- PCA 計數器陣列(P89C51)。
- 可線上燒錄(ISP, In-System Programmable) 的快閃記憶體(Flash Memory)，只要 5V 電壓，即可燒錄與清除。
- Power off flag。
- 三階程式記憶體鎖碼(Three-level Program Memory Lock)。

Intel 公司所推出的 MCS-51 系列產品，其內部結構如下：

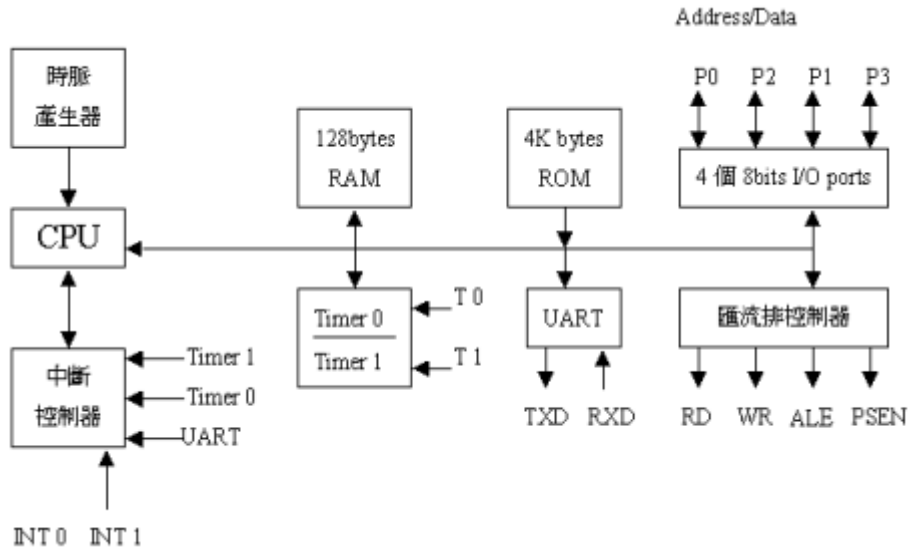


圖 2-2-2 單晶片的內部結構圖

(五) 單晶片的接腳

8051 為 40 支接腳之單晶片，其接腳圖與功能說明如下：

P1.0	1	8 0 5 1  單 晶 片	40	Vcc
P1.1	2		39	P0.0/AD0
P1.2	3		38	P0.1/AD1
P1.3	4		37	P0.2/AD2
P1.4	5		36	P0.3/AD3
P1.5	6		35	P0.4/AD4
P1.6	7		34	P0.5/AD5
P1.7	8		33	P0.6/AD6
R5T	9		32	P0.7/AD7
RXD/p3.0	10		31	$\overline{EA}$
TXD/P3.1	11		30	ALE
$\overline{INT0}$ /P3.2	12		29	$\overline{PSEN}$
$\overline{INT1}$ /P3.3	13		28	P2.7/A15
T0/P3.4	14		27	P2.6/A14
T1/P3.5	15		26	P2.5/A13
$\overline{WR}$ /P3.6	16		25	P2.4/A12
$\overline{RD}$ /P3.7	17		24	P2.3/A11
XTAL2	18		23	P2.2/A10
XTAL1	19		22	P2.1/A9
GND	20		21	P2.0/A8

圖 2-2-3 單晶片 8051 的接腳圖

1. Vcc：+5 電源供應接腳。
2. GND：接地接腳。
3. P0.0~P0.7：埠 0，為開洩極(OpenDrain)雙向 I/O 埠。在做為外部擴充記憶體時，可低八位元位址線(A0~A7addressline)與資料匯流排(databus)雙重功能。在做為一般 I/O 埠時必須加上如下之外部提升電路。

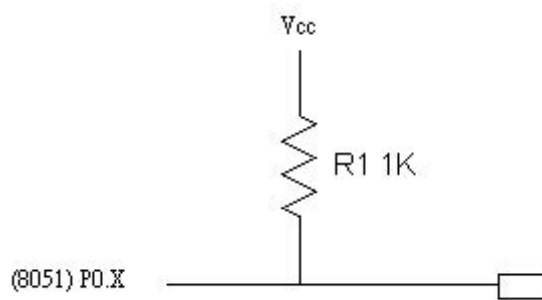


圖 2-2-4 單晶片埠 0 應用於 I/O 時的提升電路圖

4. P1.0~P1.7：埠 1，為具有內部提升電路的雙向 I/O 埠。
5. P2.0~P2.7：埠 2，為具有內部提升電路的雙向 I/O 埠。在做為外部擴充記憶體時，可為高八位元位址線(A8~A15 address line)。
6. P3.0~P3.7：埠 3，為具有內部提升電路的雙向 I/O 埠。此外，埠 3 的每支接腳都具有另一特殊功能，其功能如下：

RXD(P3.0)：串列傳輸的接收端。

TXD(P3.1)：串列傳輸的輸出端。

$\overline{INT0}$  (P3.2)：外部中斷輸入端。

$\overline{INT1}$  (P3.3)：外部中斷輸入端。

T0(P3.4)：計時/計數器外部輸入端。

T1(P3.5)：計時/計數器外部輸入端。

- $\overline{WR}$  (P3.6)：外部資料記憶體寫入激發信號(Strobe)。
- $\overline{RD}$  (P3.7)：外部資料記憶體讀取激發信號(Strobe)。
- 7. RST：重置信號(Reset)輸入端。在單晶片工作時，將此腳保持在“Hi”兩個機械週期，CPU 將重置。
- 8. ALE：位址鎖住致能(Address Latch Enable)，在每個機械週期都會出現，可做為外部電路的時脈源。
- 9.  $\overline{PSEN}$ ：程式激發致能(Program Strobe Enable)，可輸入外部程式記憶體的讀取信號。
- 10.  $\overline{EA}$ ：外部存取致能(External Access Enable)，當 EA 接腳為“L0”時，則讀取外部程式記憶體執行。
- 11. XTAL1：反相振盪放大器的輸入端。
- 12. XTAL2：反相振盪放大器的輸出端。

#### (六) 單晶片程式指令介紹

組合語言程式其定址法可分為六種：

- |          |         |
|----------|---------|
| 1.直接定址法  | 2.間接定址法 |
| 3.暫存器定址法 | 4.立即定址法 |
| 5.索引定址法  | 6.位元定址法 |

##### 1.直接定址法

所謂直接定址法，就是在指令中，直接定運算元所在的位址。僅適用於內部資料記憶體(RAM)及特殊功能暫存器(SFR)。

如下：

MOV A, 3FH；把位址 3FH 的內容存入累加器 A

ADD A, 30H；把位址 30H 的內容加到累加器 A

##### 2.間接定址法

間接定址法是把運算元的位址存放在一個暫存器，這個暫存器就是運算元位址的指標。

##### 3.暫存器定址法

8051 內部 RAM 的每個暫存器庫均含有 8 個暫存器，稱為 RO-R7，若運算元是使用 RO-R7 的位址都稱為暫存器定址法。

如下：

MOV A, R7; 把暫存器 R7 的內容存入累加器 A

MOV A, R6; 把暫存器 R3 的內容加到累加器 A

#### 4.立即定址法

立即定址法是把運算元直接放在運算碼的後面。若運算元是常數資料，則必須以“#”號當作立即值的前置符號。

如下：

MOV A, #30H; 把一個常數 30H 存入累加器 A

MOV R5, #05H; 把一個常數 05H 存入 R5 暫存器

#### 5.索引定址法

8051 的索引定址法僅適用於 ROM(程式記憶體)，而且只能讀出，不能寫入。所謂索引定址法就是以一個基底暫存器的內容，再加上一個索引暫存器的內容，所得的值即是運算元所在的位址。採索引定址法時，當基底暫存器的是 DPTR(資料指標暫存器)或 PC(程式記數器)，當索引暫存器的則是累加器 A。

如下：

MOV A, #30H

MOV DPTR, #300H

MOVC A, @A+DPTR

；將程式記憶體位址 330H(30H+300H)的內容存入累加器 A

#### 6.位元定址法

位元定址法是指對內部資料記憶體(RAM)及特殊功能暫存器(SFR)的某個位元直接設定或清除。就因為 8051 具有位元定址法，所以我們可以輕易的控制功能強大的特殊功能暫存器(SFR)，讓 8051 發揮最大效用，這是 8051 很重要的角色。但是位元定址法，只能使用於可位元址的暫存器。

如下：

SETB C ; 設定進位旗標 C 為 1。

SETB P1.0; 設定埠 1(P1)的第 0 位元為 1。

MOV C, ACC.2; 把累積器 ACC 的第 2 位元的值存入進位旗標。

## 參、專題製作

### 一、設備及器材

表 3-1-1 專題製作使用儀器（軟體）設備一覽表

儀器（軟體） 設備名稱	應用說明
個人電腦	專題報告、電路圖製作及進行專題成品電路測試
數位相機	拍攝小組合作過程、專題功能使用及紀錄整個專題製作流程
雷射印表機	列印專題資料、圖片及專題報告成果
三用電錶	測量零件有無損壞及專題電路板各信號之量測
IC 萬用燒錄器	利用燒錄器將程式燒錄至 89C51 單晶片
電源供應器	提供專題成品所需之電源
Microsoft Office Word	專題報告、製作過程的撰寫
Microsoft Office Power Point	進行口頭報告、製作及專題成品報告呈現
Keil-C	單晶片組合語言程式之編輯、燒錄軟體
Protel 99SE	繪畫專題電路之線路圖

### 二、製作方法與步驟

本專題研究採用的是行動研究法，主要是由循環的研究歷程所構成，包括準備、實驗教學、電路資料分析及報告撰寫等階段。本研究之製作方法與步驟，如圖 3-2-1 所示。



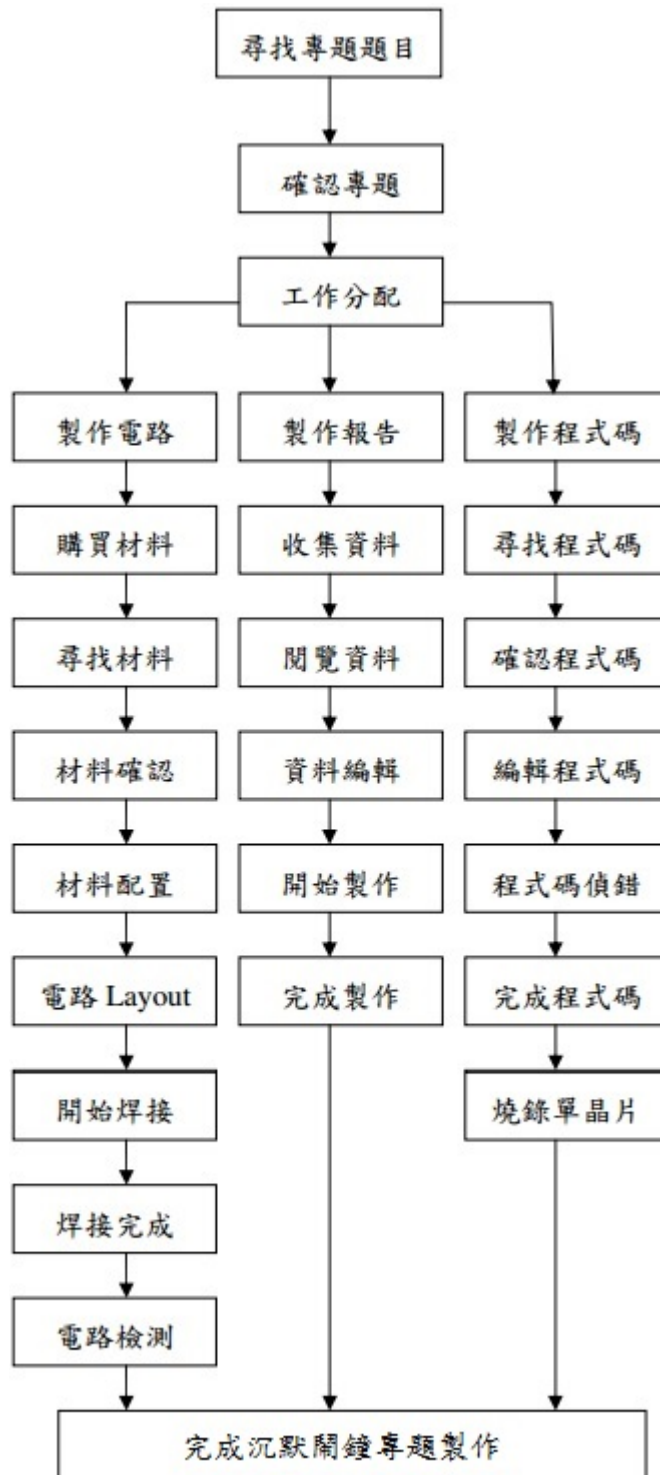


圖 3-2-1 製作方法與步驟

### 三、專題製作

表 3-3-1 專題製作計畫書

專題型別		<input type="checkbox"/> 個人型專題 <input checked="" type="checkbox"/> 團隊型專題	
專題性質		切換式鬧鐘，可切換成震動式與普通鬧鐘。	
科別／年級		資訊科三年級	
專題名稱	中文名稱	沉默鬧鐘	
	英文名稱	Silent clock	
專題內容簡述		利用 LCD 液晶顯示器顯示時：分：秒。利用 89C51 單	
		晶片燒入程式並控制震動馬達，此沉默鬧鐘經由震動或聲	
		音來叫醒使用者，或兩者同時動作，專題運轉時 LED 會每	
		秒閃爍一次，當設定時間到鬧鈴開始動作時，則會一直閃	
		爍。	
指導老師姓名		葉忠賢 老師	
參與同學姓名		楊鈞富	洪一忠
		劉柏亨	
專題執行日期		101 年 09 月 01 日至 102 年 05 月 31 日	

(一) 硬體電路圖: 沉默鬧鐘

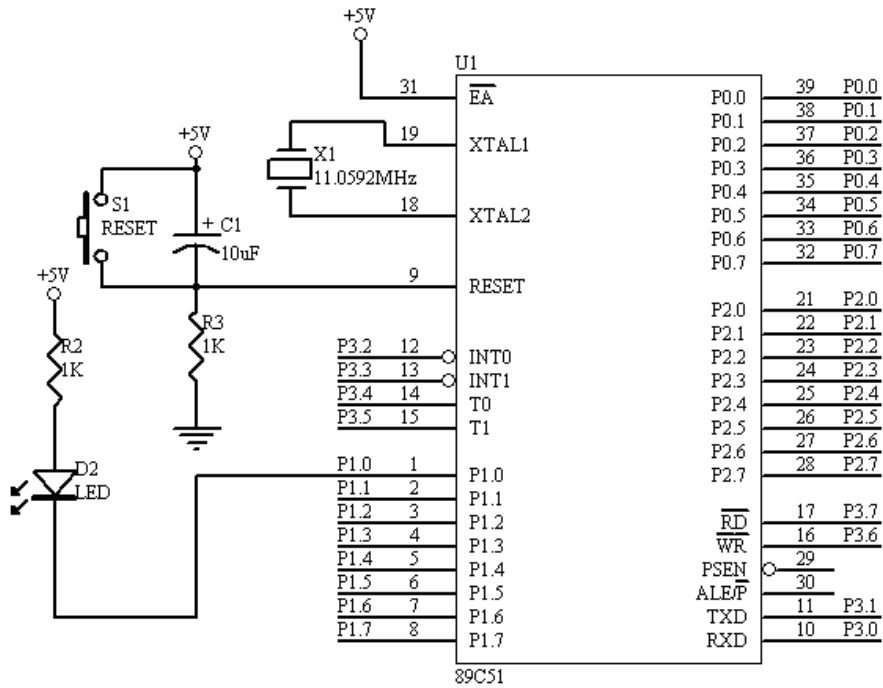


圖 3-3-1 此專題單晶片電路圖

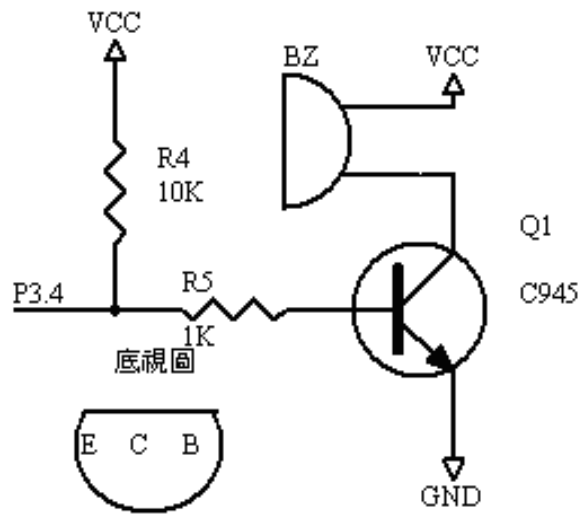


圖 3-3-2 此專題電晶體接喇叭、震動馬達電路圖

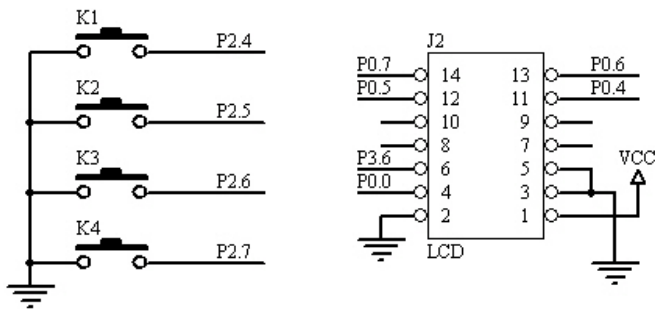


圖 3-3-3 此專題按鈕與 LCD 電路圖

(二) 沉默鬧鐘之電路板 Layout 圖及材料表：

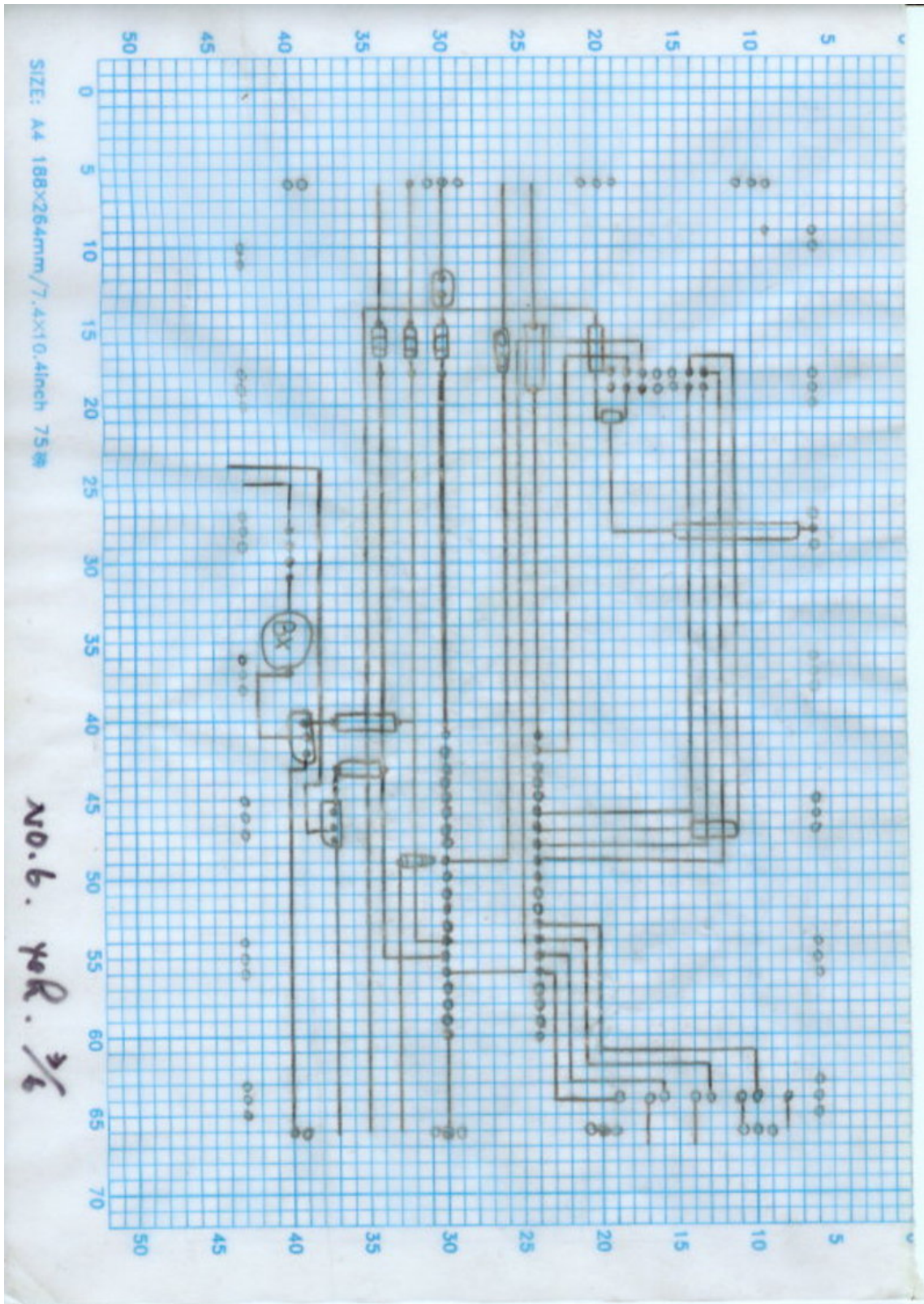


圖 3-3-4 沉默鬧鐘之電路板 Layout 圖

表 3-3-2 沉默鬧鐘之材料表

材料名稱	規格	單位	數量	備註
晶體震盪器	12MHz	個	1	
電容器	10u/16V	個	1	
LCD	16*2	個	1	
電阻	10kΩ	個	2	
電阻	1kΩ	個	2	
震動馬達	微型	個	1	
單晶片	89C51	個	1	
按鈕	2P	個	5	
可變電阻	1K	個	1	
喇叭	ZEETEK	個	1	
IC 座	40Pin	個	1	
電晶體	C1815	個	2	

### (三)小組分工的配置

組長鈞富負責小組的資料，及整合簡報內容要如何去製作專題，與其他小組成員柏亨、一忠一起討論以及商量，由柏亨負責去購買材料跟組長回報材料總共多少錢，在來討論大家平分所要的金額。

柏亨是負責材料及網路相關訊息的收集，與鈞富一起來整理資料與簡報、照片 PhotoImpact 10，設計方格紙，負責搬取焊接用木板，烙鐵等等及上網搜尋相關的電路圖及網站。

一忠負責製作麵包板及PC版，與負責拍照製作過程，及記錄在製作過程中，如發現錯誤在與小組討論辦法與補救，當有問題時，會去徵詢老師的意見。

我們三人會輪流上台去報告沉默鬧鐘的簡報

## 肆、製作成果

我們小組由決定題目，製作模擬電路、繪製設計電路圖，進而完成焊接製作整個電路；這整個流程，我們小組都用數位相機及相關電腦設備將之紀錄下來，經將這些資料整理過後，我們將之呈現在我們的專題報告之中，如下所示：

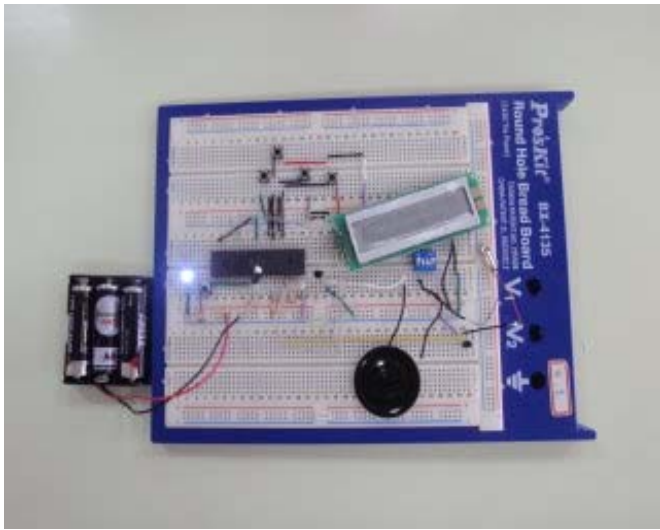


圖 4-1-1 沉默鬧鐘麵包板測試圖



圖 4-1-2 沉默鬧鐘成品圖



圖 4-1-3 沉默鬧鐘完整圖



4-1-4 打專題報告





4-1-5 測試成品(1)



4-1-6 測試成品(2)

圖 4-1-1：這是我們的測試電路後所完成的麵包板製作過程中因為喇叭而做了很多測試。

圖 4-1-2：這是我們所做出來的成品因為老師有提議加入可以調整震動大小的旋轉鈕。

圖 4-1-3 這是把成品的板子把震動馬達裝進鴨子圖案的枕頭中。

## 伍、結論與建議

### 一、 結論

此專題利用震動馬達來達成了，一般鬧鐘叫醒人的目的，可設定為震動來叫你或者由蜂鳴器來叫醒，沉默鬧鐘上還裝上了一顆 LED 燈，每秒閃爍一次，鬧鈴響時則會一直閃爍，此專題如上市了，絕對不會增加成本，因為震動馬達非常地便宜，實用性高，耐用。

透過此次專題製作學習方式能幫助我們提升對課程的學習、獲得知識的建構及整合，且亦可以幫助我們提升其創造思考的能力，培養我們學習者具備問題解決、研究、反省、團體合作及應用資訊科技等多項能力；小組同學認為專題製作學習為一主動探究的學習，學習中強調學習者必須負起主動探究學習的責任，也鼓勵小組成員分工和合作學習的精神。雖然會遇到不同的困難及問題，但看到自己的成品時，會很有成就感。

我們在製作這個專題時，從插麵包板就出現了問題，如 LCD 沒辦法插入麵包板，所以我們就先使用 OK 線，使用 OK 線非常的方便，因為因此 LCD 就不會佔位置等等。後來我們畫 Layout 圖，一開始我們先畫出 89C51 部分，然後再來畫 LCD 與按鈕位置部分，直到要畫震動馬達與蜂鳴器，我們想了很久，究竟要把震動馬達外接還是直接焊到 PC 板上，後來我們決定外接，我們 Layout 圖畫完後，馬上就開始製作 PC 板，做完後我們馬上將震動馬達裝置在枕頭上，馬上進行測試，結果我們老師在測試時發現震動馬達如果壓到的話就不會震動了，於是我們將震動馬達裝了塑膠殼，再次進行測試，因為我們怕裝上了塑膠殼會因此震動度不夠強醒不過來，後來我們將此問題解決了，老師又給我們個寶貴建議，就是因為使用者因人而異，有些人怕震動度太強或太弱，所以我們裝上了可變電阻。

### 二、 建議

我們在進行專題製作學習的過程後，提出以下幾點建議：

- (一) 由遠端控制設定鬧鈴時間。
- (二) 此專題除了裝置在家中的枕頭，建議可裝置在大眾交通工具上，如：火車、公車上等等，因為常常會因為在車上休息，而錯過了下車時間，如能裝置在車子上，記不會吵到別人，又不會錯失下車時間。



- (三) 俱有省電模式，因為大部分又不會每分每秒盯著時鐘看時間。
- (四) 無線震動馬達，就是震動馬達不需要線，不然的話在睡夢中，一不小心把線扯斷了。

## 參考文獻

- 1.吳金成、郭庭吉，2008，單晶片 8051 專題製作使用 Keil AX51，台北市：文魁資訊。
- 2.朱永昌，2007，8051 單晶片微電腦原理與專題製作(上)，台北縣：台科大圖書公司。
- 3.林明德，WonDerSun，2008，專題製作-電子電路篇，台北縣：台科大圖書公司。
- 4.長高企業，1998，U3-TARGET 單晶微電腦實驗裝置，台中市：長高企業公司。
- 5.鍾明政，1999，單晶片 8051 原理與實作，台中市：長高企業公司。
- 6.鄧明發，陳茂璋，2000，微電腦專題製作應用電路，台北市：知行文化公司。
- 7.創意導航核心，2007，PhotoImpact 12 影像哈燒秀，台北市：全華圖書公司。
- 8.蔡朝洋，2007，單晶片電腦 8051/8951 原理與應用，台北縣：全華圖書公司。
- 9.郭庭吉，2008，8051 單晶片微電腦專題製作，台北縣：台科大圖書公司。
- 10.柯南，2001，全能電路設計之 Protel Schematic 99 SE，台北縣：台科大圖書公司。

## 附錄一 沉默鬧鐘之程式碼

```
;-----  
; CK.ASM IO51 PRX LCD CLOCK+ALARM+BZ  
; 8051 ASM USE: 2500 A.D.  
;-----  
; K1 : ADJ TIME K2: HOUR K3: MIN K4: OK  
;-----  
; 5 mS isr  
; MOV TL0, #(8192-5000).MOD.32  
; MOV TH0, #(8192-5000)/32  
LO EQU 24  
HI EQU 99  
;*****  
CO EQU 200 ; ADJ. TO MATCH TIME 200 x 5 mS=1000 mS  
  
ALB REG 20H.0 ; ALARM BIT  
  
HOUR EQU 30H  
MIN EQU 31H  
SEC EQU 32H  
DEDA EQU 33H ; DEDA COUNT 5 mS COUNT  
SEC0 EQU 34H ; OLD SEC  
X EQU 35H ; LCD X POS  
HH EQU 36H ; ALARM HOUR  
MM EQU 37H ; ALARM MIN  
;-----  
; KEY4.....  
K1 REG P2.4  
K2 REG P2.5
```

K3 REG P2.6

K4 REG P2.7

; LCD .....

RS REG P0.0

EN REG P3.6

RY1 REG P3.5

SPK REG P3.4

WLED REG P1.0

;-----

ORG 0H

JMP START

ORG 0BH ; TIMER 0

JMP T0\_INT

;-----

START:

CLR EN ; LCD OFF

CLR RY1

;\*\*\*\*\*

SETB ALB ; ALARM ON

MOV HH,#1 ; DEFAULT ALARM TIME--1:1

MOV MM,#1

CALL SET\_LCD

CALL BZ

CALL BZ

CALL LED\_BL

CALL INIT

CALL INIT\_TIMER

CALL MENU

LOOP:

CALL TIME\_PRO

; TEST KEY .....

JB K1,L1

; K1 KEYED .....SET TIME

CALL BZ

MOV DPTR,#MADJ

MOV A,#1

CALL LCD\_PRINT

CALL SET\_TIME ; ADJ TIME

CALL MENU

JMP LOOP

;

L1: JB K2,L2

; K2 KEYED.....LOOK ALARM TIME

CALL BZ

MOV DPTR,#MLAL

MOV A,#1

CALL LCD\_PRINT

CALL LOOK\_ATIME ; LOOK ALARM TIME

CALL LED\_BL

CALL LED\_BL

CALL MENU

JMP LOOP

;

```

L2:    JB  K3,L3
        JNB K3,$
; K3 KEYED .....SET ALARM TIME
        CALL BZ
        MOV DPTR,#MAADJ
        MOV A,#1
        CALL LCD_PRINT
        CALL SET_ETIME ; SET ALARM TIME
        CALL LED_BL
        CALL LED_BL
        CALL MENU
        JMP LOOP

```

```

;.....

```

```

L3:    JB K4,L4
; K4 KEYED.....ALARM ALB ON/OFF
        CPL ALB
        JNB ALB,L31
; ALARM ON.....
        CALL BZ
        CALL BZ
        CALL BZ
        JMP LOOP

```

```

L31:
; ALARM OFF.....
        CALL BZ
        JMP LOOP

```

```

;.....

```

```

L4:
        JMP LOOP

```

```

;-----
LMESS1: DB "CK 8051 CLOCK ",0
LMESS2: DB "TIME:           ",0
TITLE:  DB "CK.ASM 8051 CLOCK WITH LCD+ALARM+BZ"
;-----

```

INIT:

```

    MOV DEDA,#0
    MOV SEC0,#0
    MOV SEC,#0
    MOV MIN,#0
    MOV HOUR,#1
    RET

```

```

;-----

```

INIT\_TIMER:

```

    MOV TMOD,#00000000B
    MOV IE,  #10000010B
    MOV TL0,#LO
    MOV TH0,#HI
    SETB TR0
    RET

```

```

;-----

```

```

; USE TIMER MODE 0
; COUNT=5ms/1us=5000
;-----

```

T0\_INT:

```

    PUSH ACC
    MOV TL0,#LO
    MOV TH0,#HI
    INC DEDA

```

; 1 SEC TIME OUT CHECK

```
MOV A,DEDA
CJNE A,#CO,TT
MOV DEDA, #0
```

```
CPL WLED
INC SEC
MOV A,SEC
CJNE A,#60,TT
```

; 1 MIN TIME OUT

```
INC MIN
MOV SEC,#0
MOV A, MIN
CJNE A,#60,TT
```

; 1 HOUR TIME OUT

```
INC HOUR
MOV MIN,#0
MOV A, HOUR
CJNE A,#24,TT
MOV SEC,#0 ; RESET TIME TO 0
MOV MIN,#0
MOV HOUR,#0
```

TT:

```
POP ACC
RETI
```

;-----

; K2:HH K3:MM K4:OK

SET\_TIME:

```
CLR TR0
MOV SEC,#0
```



; SCAN 3 KEY TO ADJ.....

S0:

JB K2,S1

; K1 KEYED ADJ HH.....

CALL BZ

INC HOUR

MOV A, HOUR

CJNE A,#24,S11

MOV HOUR,#0

S11:

CALL CONV ; CONV. TO ASCII

MOV R5,#20

CALL DELAY

JMP S0

;.....

S1: JB K3,S2

; K2 KEYED ADJ MM.....

CALL BZ

INC MIN

MOV A, MIN

CJNE A,#60,S21

MOV MIN,#0

S21:

CALL CONV ; CONV. TO ASCII

MOV R5,#20

CALL DELAY

JMP S0

;.....

```

S2:    JB K4,S0
        JNB K4,$
; K3 KEYED  OK .....
        CALL BZ
        CALL BZ
        SETB TR0
        RET

```

```

;-----

```

```

; A: DIG    X: LCD X POS

```

```

; SHOW ON LCD LINE1

```

```

SHOW_DIG:

```

```

    MOV B,#10
    DIV AB
    ADD A,#30H
    PUSH B
    MOV B,X
    CALL LCDP1

```

```

    POP B
    MOV A,B
    ADD A,#30H
    INC X
    MOV B,X
    CALL LCDP1
    RET

```

```

;-----

```

```

; A: DIG    X: LCD X POS

```

```

; SHOW ON LCD LINE2

```

```

SHOW_DIG2:

```

```

    MOV B,#10

```

```
DIV AB
ADD A,#30H
PUSH B
MOV B,X
CALL LCDP2
```

```
POP B
MOV A,B
ADD A,#30H
INC X
MOV B,X
CALL LCDP2
RET
```

```
;-----
; LCD HH:MM:SS
; CONVERT TO ASCII AND SHOW
```

```
CONV:
```

```
; SHOW HOUR.....
```

```
MOV A,HOUR
MOV X,#5
CALL SHOW_DIG2
INC X
MOV A,#':'
MOV B,X
CALL LCDP2
```

```
; SHOW MIN.....
```

```
MOV A,MIN
INC X
CALL SHOW_DIG2
INC X
```

```

MOV A,#'!'
MOV B,X
CALL LCDP2
; SHOW MIN.....
MOV A,SEC
INC X
CALL SHOW_DIG2
RET
;*****
; LCD CONTROL
;*****
SET_LCD:
CLR EN ; en=0
CALL INIT_LCD

MOV R5,#10
CALL DELAY

MOV DPTR,#LMESS1
MOV A,#1 ; LINE1
CALL LCD_PRINT

MOV DPTR,#LMESS2
MOV A,#2 ; LINE2
CALL LCD_PRINT
RET
;-----
INIT_LCD1:
MOV A,#28H
CALL WCOM

```

```
MOV A,#0CH
CALL WCOM
MOV A,#0EH
CALL WCOM
MOV A,#01H
CALL WCOM
RET
```

```
;-----
```

```
CLR_LINE1:
```

```
MOV A,#80H
CALL WCOM
MOV R0,#24
```

```
C1: MOV A,#' '
CALL WDATA
DJNZ R0,C1
RET
```

```
;-----
```

```
; A=LINE 1 OR 2
```

```
; DPTR = MESSAGE POINTER
```

```
LCD_PRINT:
```

```
CJNE A,#1,LINE2
```

```
LINE1: MOV A,#80H
CALL WCOM
CALL CLR_LINE
MOV A,#80H
CALL WCOM
JMP FILL
```

```
LINE2: MOV A,#0C0H
CALL WCOM
```

```

        CALL CLR_LINE
        MOV A,#0C0H
        CALL WCOM
FILL:
        CLR A
        MOVC A,@A+DPTR
        CJNE A,#0,LC1
        RET
LC1:
        CALL WDATA
        INC DPTR
        JMP FILL
        RET
;-----
CLR_LINE:
        MOV R0,#24
CL1:   MOV A,#' '
        CALL WDATA
        DJNZ R0,CL1
        RET
;-----
DE:    MOV  R7, #5
        DJNZ R7,$
        RET
;-----
EN1:
        SETB EN
        CALL DE
        CLR  EN
        CALL DE

```

```

RET
;-----
INIT_LCD:
    MOV P0, #30H
    CALL EN1
    MOV P0, #30H
    CALL EN1
    MOV P0, #30H
    CALL EN1

    MOV P0, #20H
    CALL EN1
    CALL INIT_LCD1
    RET
;-----
WCOM:
; WRITE HI 4 BIT
    MOV P0, A
    CLR RS ; SET COMMAND
    CALL EN1
    RLC A
    RLC A
    RLC A
    RLC A
; WRITE LO 4 BIT
    MOV P0, A
    CLR RS ; SET COMMAND
    CALL EN1
    RET
;-----

```

```

WDATA:
; WRITE HI 4 BIT
    MOV P0, A
    SETB RS ; SET DATA
    CALL EN1
    RLC A
    RLC A
    RLC A
    RLC A
; WRITE LO 4 BIT
    MOV P0, A
    SETB RS ; SET DATA
    CALL EN1
    RET

;-----
; PRINT A CHAR ON LCD LINE 1
; A=ASC DATA
; B=LINE X POS.
LCDP1:
; LINE 1
    PUSH ACC
    MOV A,B
    ADD A,#80H
    CALL WCOM
    POP ACC
    CALL WDATA
    RET

;-----
; PRINT A CHAR ON LCD LINE 2
; A=ASC DATA

```



```

; B=LINE X POS.
LCDP2:
; LINE 1
    PUSH ACC
    MOV A,B
    ADD A,#0C0H
    CALL WCOM
    POP ACC
    CALL WDATA
    RET

;-----
; DELAY R5*10 ms
DELAY:
    MOV R6,#50
D1:   MOV R7,#100
    DJNZ R7,$
    DJNZ R6,D1
    DJNZ R5,DELAY
    RET

;-----
LED_BL:
    MOV R4,#4
LE0:  CPL WLED
    MOV R5,#10
    CALL DELAY
    DJNZ R4, LE0
    RET

;-----
MMENU: DB "1-SET 2-AL 3-SAL",0
MADJ:  DB "K2:H  K3:M K4:OK",0

```

MLAL: DB "NOW ALARM TIME :",0

MAADJ: DB "K2:H K3:M K4:OK",0

MENU:

MOV DPTR,#MMENU

MOV A,#1

CALL LCD\_PRINT

RET

-----

BZ:

MOV R6,#0

B1: CALL DEX

CPL SPK

DJNZ R6,B1

MOV R5,#10

CALL DELAY

RET

-----

DEX: MOV R7,#180

DE1: NOP

DJNZ R7,DE1

RET

-----

; LOOK ALARM TIME

LOOK\_ETIME:

; SHOW HOUR.....

MOV A,HH

MOV X,#5

```

        CALL SHOW_DIG2
        INC X
        MOV A,#':'
        MOV B,X
        CALL LCDP2
; SHOW MIN.....
        MOV A,MM
        INC X
        CALL SHOW_DIG2
        INC X
        MOV A,#':'
        MOV B,X
        CALL LCDP2
; SHOW SEC.....
        MOV A,#0
        INC X
        CALL SHOW_DIG2
        RET
;-----
; K2:HH  K3:MM  K4:OK
SET_ETIME:
        CALL LOOK_ETIME
; SCAN 3 KEY TO ADJ.....
AS0:
        JB K2,AS1
; K1 KEYED  ADJ HH.....
        CALL BZ
        INC HH
        MOV A, HH
        CJNE A,#24,AS11

```

```

        MOV HH,#0
AS11:
        CALL LOOK_ETIME ; CONV. TO ASCII
        MOV R5,#20
        CALL DELAY
        JMP AS0

;.....
AS1:    JB K3,AS2
; K2 KEYED ADJ MM.....
        CALL BZ
        INC MM
        MOV A, MM
        CJNE A,#60,AS21
        MOV MM,#0
AS21:
        CALL LOOK_ETIME ; CONV. TO ASCII
        MOV R5,#20
        CALL DELAY
        JMP AS0

;.....
AS2:    JB K4,AS0
        JNB K4,$
; K3 KEYED OK .....
        CALL BZ
        CALL BZ
        RET

;-----
; UPDATE SEC,MIN,HR AND COMP HOUR=HH MIN=MM--->ALARM

```

```

TIME_PRO:
; UPDATE TIME.....
    MOV A,SEC
    MOV B,SEC0
    CJNE A,B,TI1    ; TIME CHANGE
    JMP TI2
TI1:    CALL CONV    ; CONV. TO ASCII  SHOW TO LCD
        MOV SEC0,SEC
TI2:
; CHECK ALARM ON/OFF.....
    JB ALB,TI3
    RET
TI3:
; CHECK ALARM.....
    MOV A,HOUR
    MOV B,HH
    CJNE A,B,BK

    MOV A,MIN
    MOV B,MM
    CJNE A,B,BK
; TIME MATCH.....
    CALL TIME_OUT
BK:
    RET
;-----
TIME_OUT:
    SETB RY1

M1:    CALL LED_BL

```

```
CALL BZ
; KEY 4 STOP.....
JB K4,M1
JNB K4,$
CALL LED_BL
CLR ALB
CLR RY1
RET
```

-----

-----