

# 實驗九 ADC 的轉換應用

## 1. 實驗目的

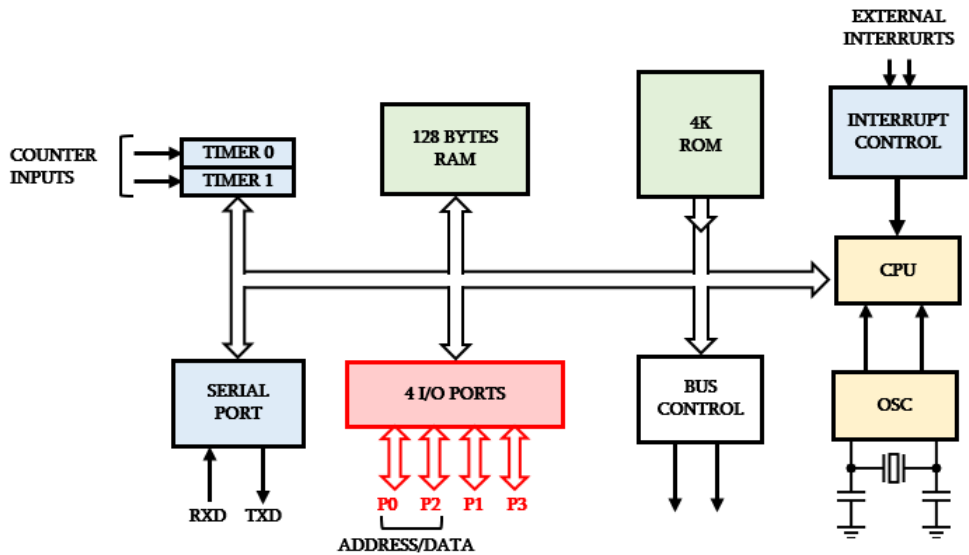


圖 9-1、Block Diagram of the 8051 Core

使用 ADC 將類比電壓訊號轉換為數位電壓訊號，將轉換後的訊號傳給 8051 的 GPIO，再由 8051 讀取數值。利用可變電阻輸入電壓分壓給 ADC0804 轉為數位訊號，並由 8051 接收後控制 8 顆 LED 燈在可變電阻旋轉時由全不亮漸變到全亮。

## 2. 材料清單

表 9-1、材料清單

器材名稱		數量
AT89S51		1
12MHz 石英震盪器		1
ADC0804		1
可變電阻 100K		1
電阻	1kΩ	9
	10 kΩ	1
電容	20pF	2
	151pF	1
	10μF	1

### 3. 元件原理

#### 類比數位轉換原理

因為 8051 只能接收數位訊號，所以在我們要讀取類比訊號時，必須先經由 ADC ( Analog-digital converter ) 類比-數位轉換器的轉換後，我們才可將類比訊號轉為數位訊號讓 8051 讀取。

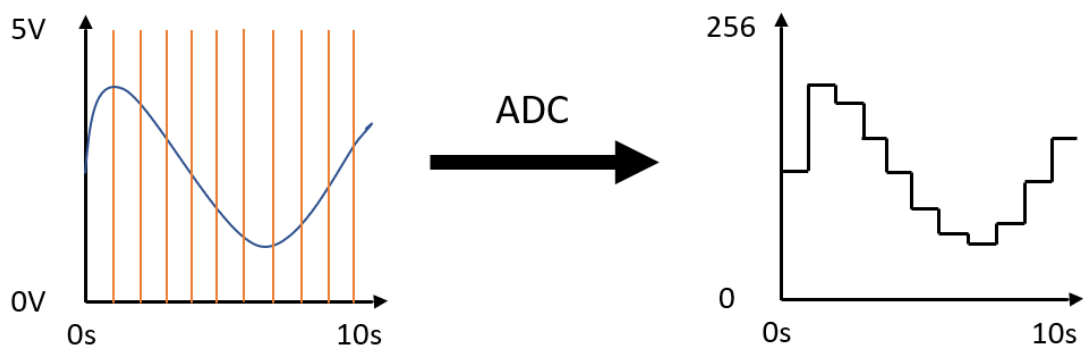


圖 9-2、類比訊號轉數位訊號原理

而根據對訊號的取樣率和取樣的解析度，會影響我們得到的數位訊號。取樣率為經過多久的時間對輸入的類比訊號做一次轉換，轉換的頻率越高，所得到的數位訊號資料越連續，但相對的也需要較快的處理速度以及存放空間。解析度影響到每次轉換後的資料準確度，解析度越高，所得到的數位訊號資料越精確，量化誤差越小，但相對的每筆資料儲存所需要的位元數將更多。

例如一般而言 8 位元的解析度，我們可將輸入的類比訊號解碼成  $2^8=256$  階 Step size ( 階層電壓值 ) 的類比序號。如果使用 8 位元解析度，且電壓上限為 0V 到 5V，則數位訊號上升一個階層需要的電壓為  $(5V-0V) \div (2^8) = 0.0195V$ 。若收到的數位訊號為  $200_{(10)}$ ，則轉換前的類比電壓為  $200 * 0.0195V = 3.9V$ 。

### 並列式類比-數位轉換

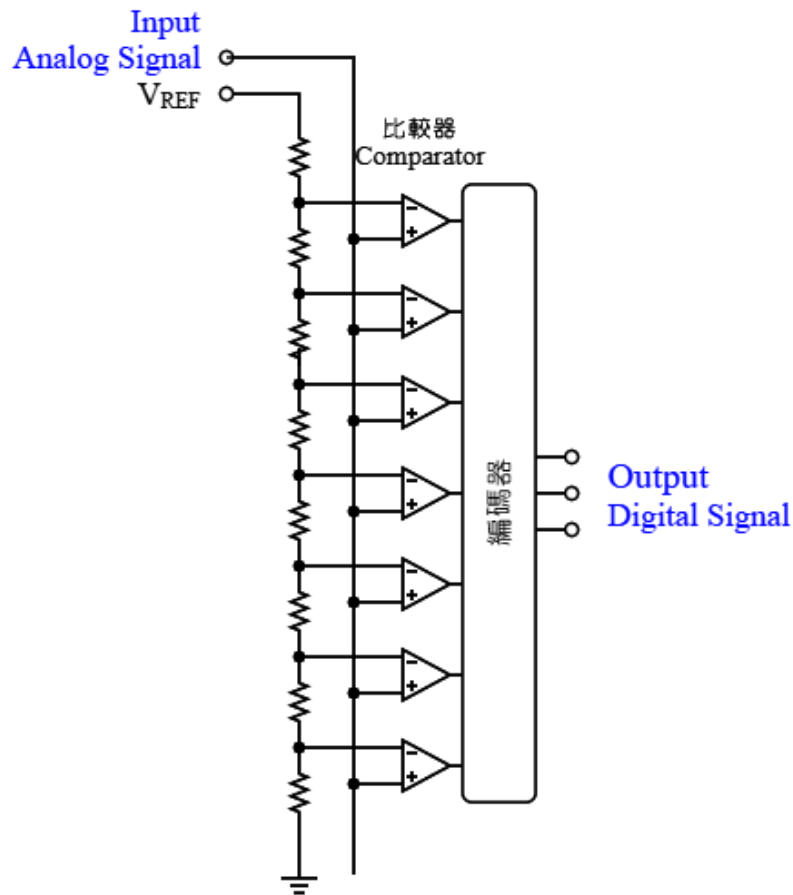


圖 9-3、並列式類比-數位轉換基本構造

並列式類比-數位轉換器以多個比較器並列偵測輸入的類比訊號，然後根據偵測輸入結果編碼，即可輸出數位訊號。此方法轉換速度快，但所需要的電路較為複雜。

### 連續接近式類比-數位轉換

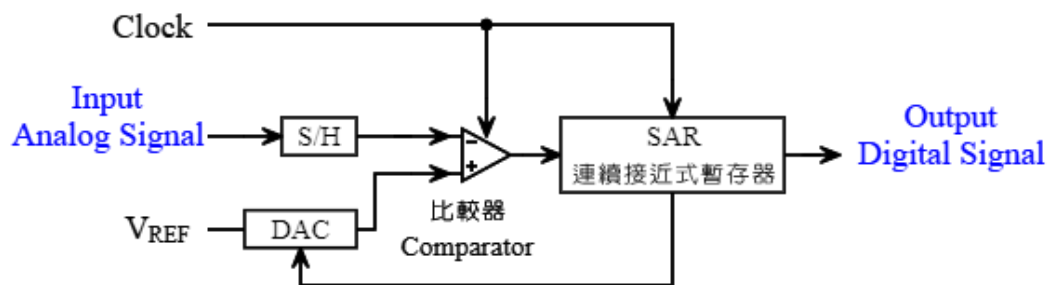


圖 9-4、連續接近式類比-數位轉換基本構造

- DAC ( digital-to-analog converter )：數位類比轉換器。
- SAR ( successive approximation register )：連續逼近暫存器，用於存放轉換的中間結果和最終結果。

- S/H ( sample and hold circuit ) : 取樣保持電路。
- Comparator : 電壓比較器，對數位類比轉換器輸出的電壓與輸入電壓進行比較
- VREF : 參考電壓。

連續接近式類比-數位轉換器，用快速比對接近的方式，將類比訊號轉為數位訊號。若以 8 位元的轉換器為例，一開始計數器為 1，一開始的參考電壓  $V_s$  為轉換後的解析度的一半  $128 (256/2)$  經過數位轉類比轉換器轉換出來的階層電壓值。若輸入的類比電壓大於  $V_s$ ，則將轉換結果暫存器的值加上  $2^n$  ( 解析度位元-計數器-1 )。如果輸入的類比電壓小於  $V_s$ ，則將轉換結果暫存器的值減去  $2^n$  ( 解析度位元-計數器-1 )。判斷完後，計數器加 1，並且繼續將轉換結果暫存器的內容送給數位類比轉換器，直到計數器達到轉換位數 ( 這裡為 8 ) 時，再比較一次，若輸入電壓較小，則將最小位元清除為零；輸入電壓較大則不動，轉換結果暫存器即儲存結果。

例如，計數器為 1，結果暫存器為  $1000_{(2)}$ ，而參考電壓為 8。比較 8 轉成的類比電壓和輸入電壓，輸入電壓較低，故將結果暫存器的值減去  $2^n (4-1-1) = 0100_{(2)}$ ，並將計數器加 1。再比較 4 轉成的類比電壓，輸入電壓較高故將結果暫存器的值加上  $2^n (4-2-1) = 0110_{(2)}$ ，並將計數器加 1。再比較 6 轉成的類比電壓，輸入電壓較高故將結果暫存器的值加上  $2^n (4-3-1) = 0111_{(2)}$ ，並將計數器加 1。當計數器等於 4 時，最後比較 7 轉成的類比電壓，輸入電壓較低，故將最小位數清 0，獲得結果  $0110_{(2)}$ 。

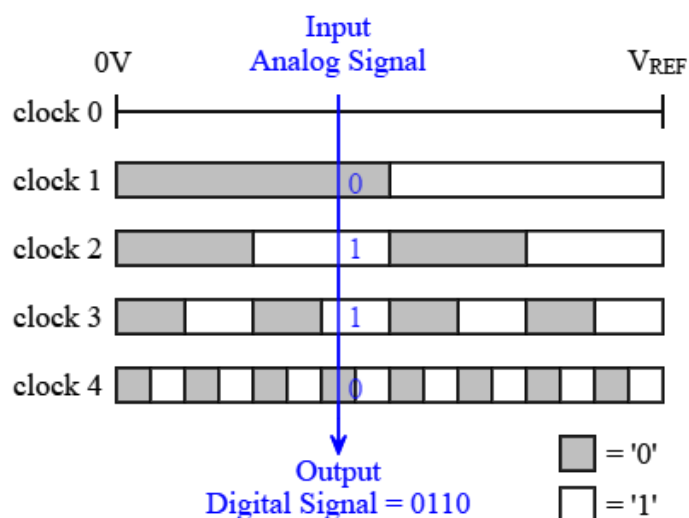


圖 9-5、重複比較大小

## 類比數位轉換器 ADC0804

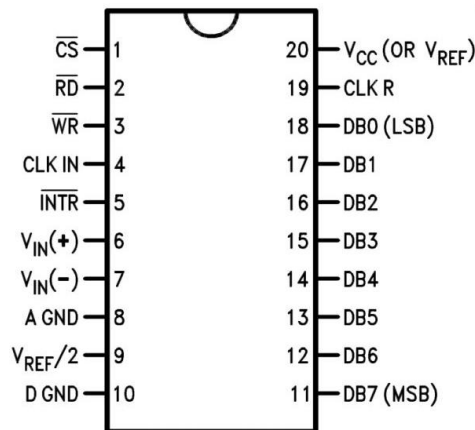


圖 9-6、ADC0804 腳位

此圖為 ADC0804 接腳圖，以下說明各接腳用途：

- CS：此接腳須為 0 時，ADC0804 才會動作；若此接腳為 1，輸出接腳（DB0 ~ DB7）將會呈現高阻抗狀態。
- RD：資料讀取接腳，若此接腳為 0 且 CS 接腳也為 0，則可由輸出接腳讀取轉換過後的資料，若此接腳為 1，輸出接腳（DB0 ~ DB7）將會呈現高阻抗狀態。
- WR：此接腳控制 ADC0804 是否開始轉換，若 WR 為 0 則 ADC 做清除的動作，系統重置，若 WR 由 0 變 1 時，開始執行類比轉數位的轉換，此時 INTR 變為 1。
- INTR：若 INTR 接腳等於 0，代表 ADC0804 已完成轉換，當資料被讀取後恢復為 1。這個接腳通常也和外部中斷搭配使用。
- CLK IN / CLK R：CLK R 可以搭配電阻器產生時鐘脈波。CLK IN 則是輸入時鐘脈波的接腳，故我們可以使用 CLK R 輸出的時鐘脈波輸入給 CLK IN 使用。
- V<sub>REF</sub>/2：提供 ADC0804 轉換的參考電壓，電壓準位為最大輸入類比電壓的一半。若空接則為 V<sub>CC</sub>/2。
- V<sub>IN</sub>(+)/V<sub>in</sub>(-)：類比電壓輸入接腳，正負電壓分別接至 V<sub>IN</sub>(+)和 V<sub>in</sub>(-)端，若為單一正電壓，則接到 V<sub>IN</sub>(+)，再將 V<sub>in</sub>(-)接地。
- D GND/A GND：數位/類比訊號接地接腳，通常將 A GND 和 D GND 接腳相接。

後再接地。

- DB0 ~ DB7：數位資料輸出接腳，可直接與 8051 資料匯流排相接。

## ADC0804 操作方式

ADC0804 有兩種操作方法，其中一種為連續轉換，一種為交握式控制：

### 連續轉換

ADC0804 不停的進行轉換類比資料，要注意轉換頻率也是有極限的，若 ADC 時脈為 30MHz 而解析度為 8bits 時， $30\text{MHz}/8$  為轉換頻率極限。將 CS 與 RD 接腳直接接地，並把 INTR 和 WR 接腳對接，使輸出完成後可以直接執行下一次轉換。

### 交握式控制

將接腳與 8051 的輸出埠相接，當 INTR 輸出完成訊號時，8051 經由讀取接腳得知，即可開始讀取轉換後的數位輸入訊號，若將這個 INTR 訊號連接外部中斷接腳，則可使用中斷方式讀取轉換資料。

## 4. 實驗內容

利用可變電阻輸入電壓分壓給 ADC0804 轉為數位訊號，8051 接收後判定是否超過電壓門檻（設為輸入最大電壓的一半），若超過則 LED 燈全亮。

## 5. 實驗電路圖

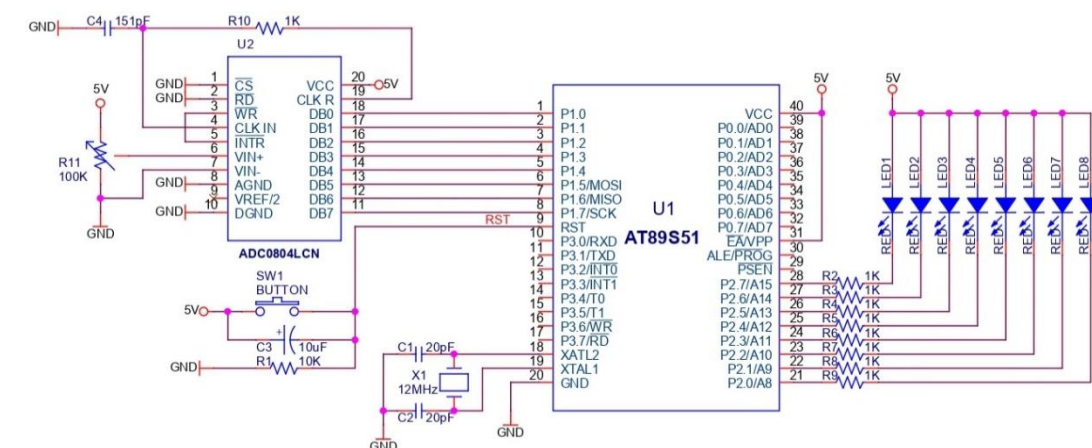


圖 9-7、實驗九基礎題參考電路圖-連續轉換

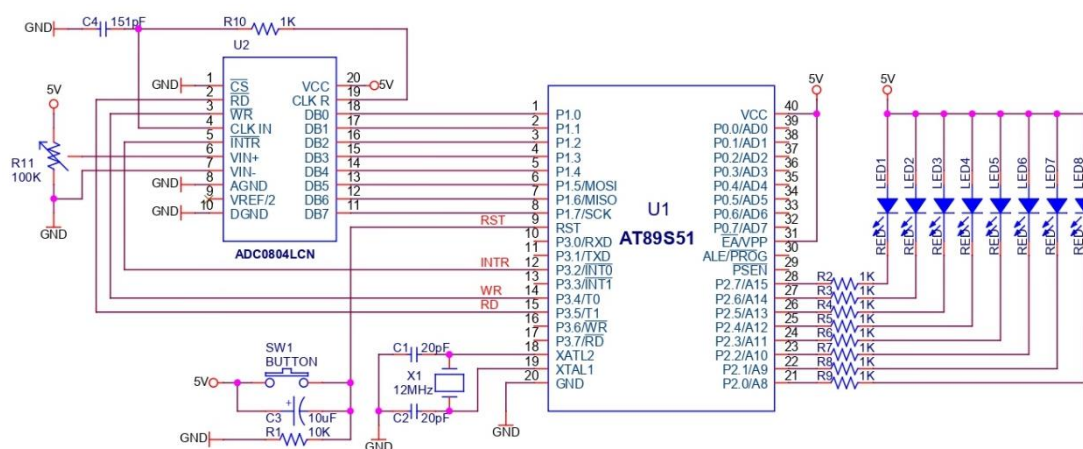


圖 9-8、實驗九基礎題參考電路圖-交握式控制

## 6. 軟體流程圖

### 輪詢法

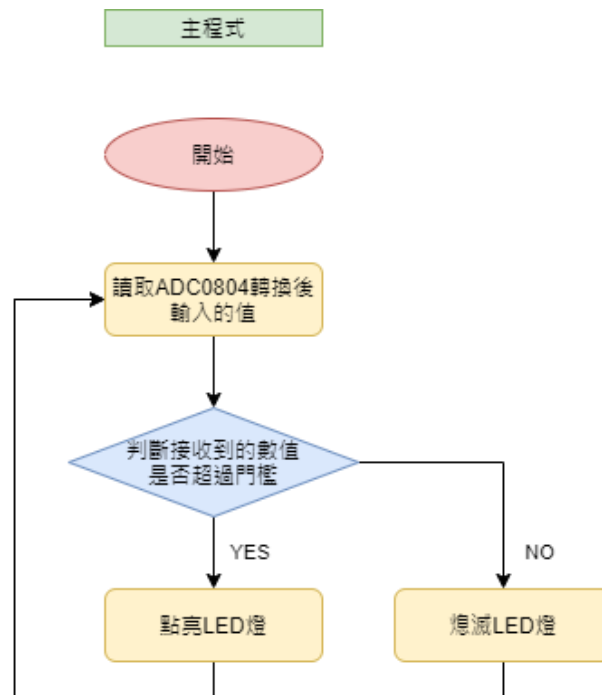


圖 9-9、ADC 轉換應用-輪詢法

### 中斷法

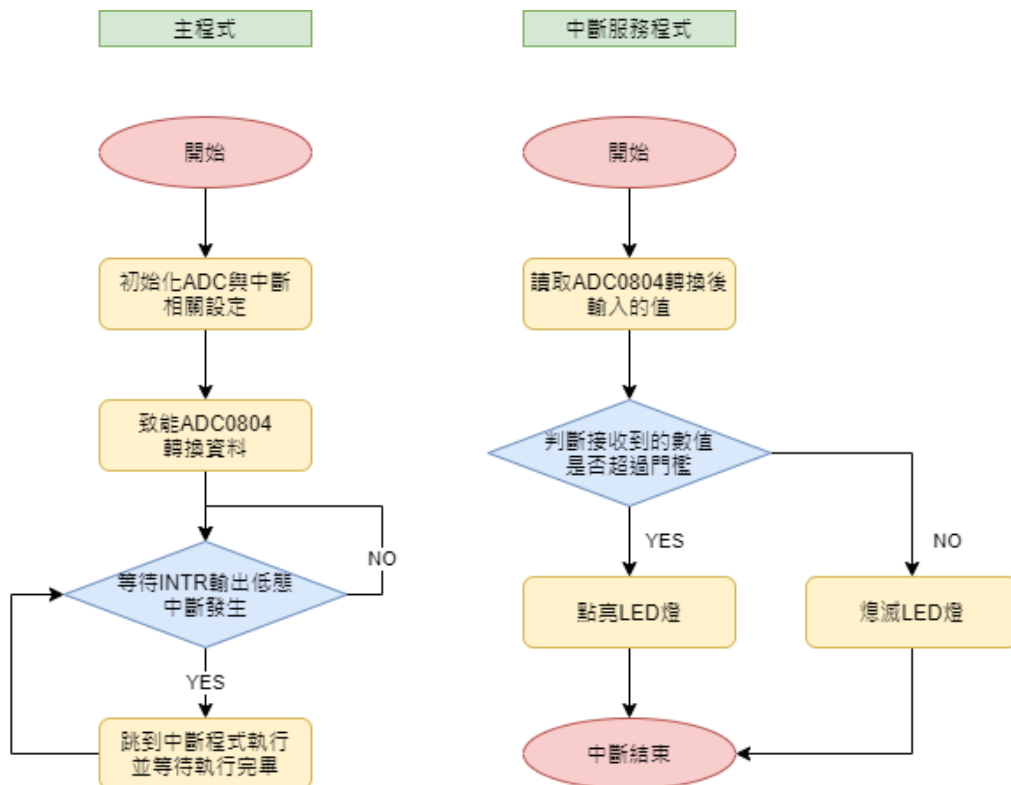


圖 9-10、ADC 轉換應用-中斷法



## 7. 範例程式碼

### 輪詢法

輪詢法使用第一種連續轉換模式，接電路時須將 INTR 和 WR 接腳對接且把 RD 腳接地。

```
1  #include <regx51.h>
2
3  void main( )
4  {
5      while (1)
6      {
7          if (P1 > 256 / 2)    //if P1 greater than 256/2
8              P2 = 0;         //LED on
9          else
10             P2 = 255;        //LED off
11     }
12 }
```

### 中斷法

中斷法使用第二種連交握式模式。

```
1  #include <regx51.h>
2
3  void main( )
4  {
5      IT0 = 0;                //enable a low-level signal on external interrupt
6      EX0 = 1;                //enable INT0
7      EA = 1;                 //enable interrupt
8      P3_4 = 0;               //WR=0,clean the data
9      P3_4 = 1;               //WR=1,analog convert to digital
10     while(1);               //infinite loop
11 }
12
13 void int_0(void) interrupt 0 //INT0 interrupt function
14 {
15     P3_5 = 0;                //RD=0, enable read
16     if (P1 > 256 / 2)
17     {
18         P2 = 0;
19     }
20     else
21     {
22         P2 = 255;
23     }
24     P3_5 = 1;                //RD=1, stop read
```

```
25     P3_4= 0;           //WR=0,clean the data
26     P3_4 = 1;          //WR=1,analog convert to digital
27 }
```

## 8. 整理的題目，選擇/是非題