

壹、緒論

1、製作目的

現在社會科技日新月異各式各樣產品機械化是個目前科技的趨勢，鐵捲門是日常長見到的一樣科技新產品，在現今的社會門有很多種變化從以前簡單的拉門到了現在自動的鐵捲門，帶來了我們生活上的安全和便捷，不時新聞上長有鐵捲門夾傷人的意外，所以我們在這次專題內增加了蜂鳴器讓在開關鐵捲門時候能讓人注意到鐵捲門有在運作避免傷害造成。

96/04/01	台北縣	三重市重陽國小一名13歲男童被學校的電動鐵捲門夾死
97/01/01	桃園縣	大園鄉一位50歲婦人被自家電動鐵捲門壓死
97/05/05	台中縣	太平市鐵皮屋火警電動鐵捲門無法開啟老婦慘遭燒死
98/10/25	台南縣	仁德一位女大生被電動鐵捲門壓頸

2、製作動機

所以這次我們大家組員一起討論雖然各自有各自的想法和想做的東西，最後大家達成共識做個房子模型在加上 8051 單晶片鐵捲門的應用配合，來完成這次我們的專題製作，我們使用簡單的材料減低我的製作的成本，大家分工合作完成一個成品是一件很有成就感的事情。

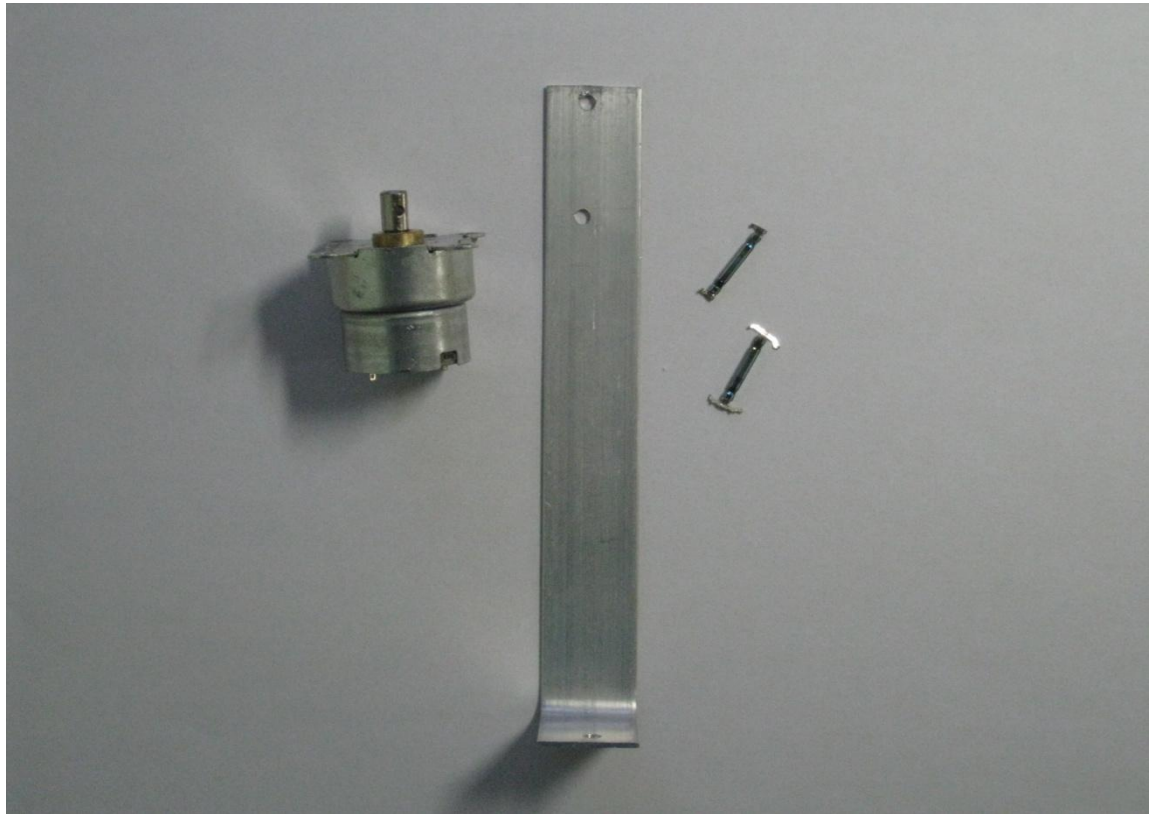
貳、製作材料

1、材料表

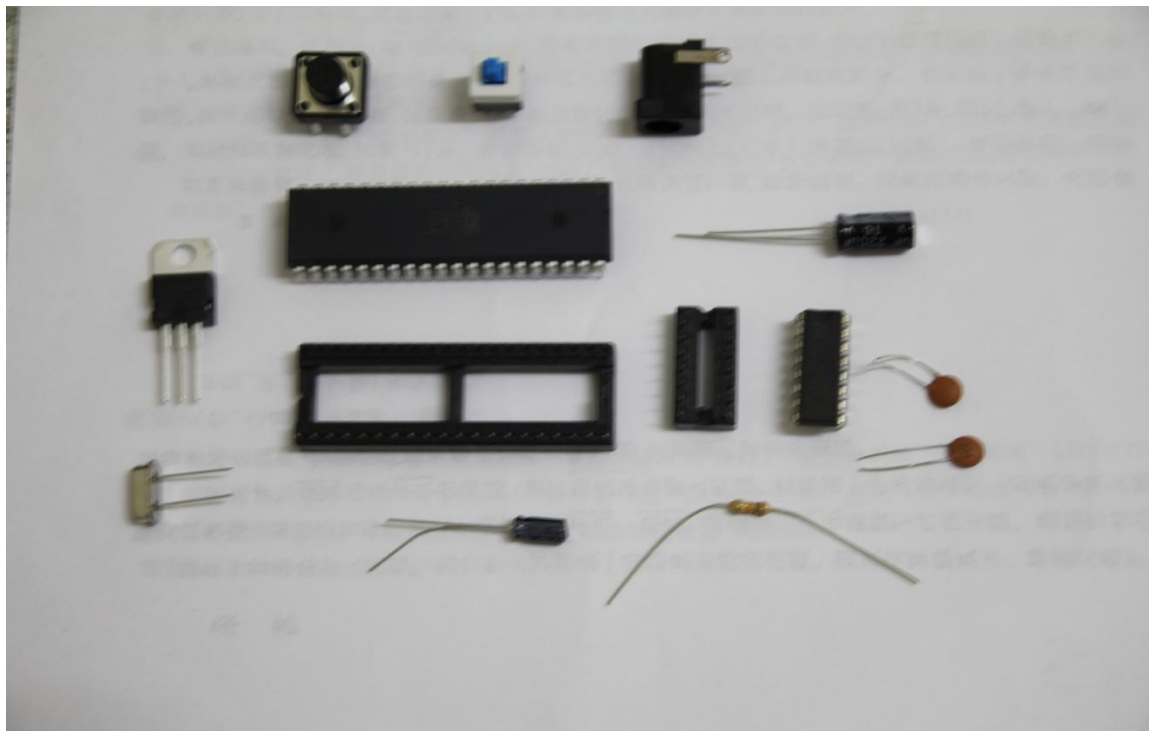
項目	編號	規格	名稱	數量
1	C1	220uF	電解電容	1
2	C2	104	陶瓷電容	1
3	C4, C3	30PF	陶瓷電容	2
4	C5	10uF/25V	電解電容	1
5	J1	DC9V		1
6	LS1	蜂鳴器	蜂鳴器	1
7	MG1	3V	馬達	1
8	R1	300	電阻	1
9	R2	10K	電阻	1
10	SW1	彈簧開關 上	彈簧開關 上	1
11	SW2	彈簧開關 下	彈簧開關 下	1
12	S1	下	按鈕	1

13	S2	停	按鈕	1
14	S3	上	按鈕	1
15	U1	L293	馬達驅動 IC	1
16	U2	78M05	5V 穩壓 IC	1
17	U3	89S51	單晶片	1
18	Y1	12MHZ	振盪器	1
19		60*120	木板	1
20			3M 木質黏 著劑	1
21			鐵條	2
22.			木棍	1
23.			鐵釘	5
24.			螺絲	數根

2. 材料圖片



馬達及鐵片螺絲



各式零件

參. MSC-51 單晶片介紹

3-1. MSC-51 晶片家族介紹

MSC-51 族微處理器包含有很多種晶片，如表 1-1 所列。其結構方塊圖則如圖 1-1 所示。

表 1-1 MSC-51 微處理器

名稱	ROMLESS	RPROM	ROM(位元組)	RAM(位元組)	16 位元計時器	電路型式
8051	8031	(8751)	4K	128	2	HOMS
8051AH	8031AH	8751AH	4K	128	2	HMOS
8052AH	8032AH	8751BH	8K	256	3	HMOS
80C51BH	80C31BH	87C51	2K	128	2	CHMOS

3-2 8051 介紹

8051 是 MCS-51 族裡最原始的一顆晶片，在 1981 年就以製造生產，此時 8051 的功能如下：

- 專為控制應用所設計的八位元 CPU
- 加強了布林代數(單位元邏輯)之運算能力
- 32 條雙向且可被獨立定址之 I/O
- 晶片內有 128 位元組可供儲存資料的 RAM
- 內部有兩個 16 位元計時器(8052 有 3 個)
- 具全雙工 UART
- 5 個中斷源，且具有兩層(高/低)優先權順序之中斷結構
- 晶片內有時脈(clock)振盪器線路
- 晶片內有 4K(8K/8052)位元組的程式記憶體(ROM)
- 城市記憶空間可達 64K 位元組
- 資料記憶體空間可定址到 64K 位元組

8031 與 8051 不同之處是 8031 內部沒有程式記憶體(ROM)，因此 8031 必須由外部城市記憶體(ROM/EPROM)提取指令。

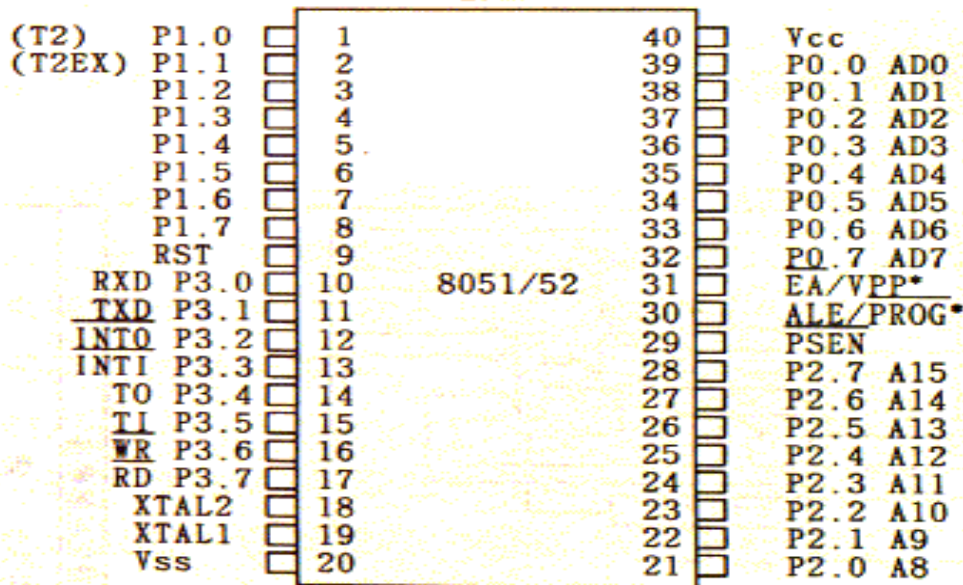
8751 則是 8051 的 EPROM 版，他已不再生產，取而代之的是 8751H。

3-3 8051AH 介紹

8051AH 與 8051 的功能相同，只是 8051AH 是以 HMOS II 的技術製造，其接腳與 8051 完全相容。

8051AH 系列中，內部沒有程式記憶體(ROM)的版本是 8031AH；而 8751H 則為 RROM 版本。

3-4 MSC-51 的 IC 接腳圖



註：

1. (T2), (T2EX) 只存在於 8052
2. Vpp*, PROG 只適用於 8751/52

圖 2-2-1 8051/8052 接腳圖 (DIP) 包裝

1~8 腳(P1.0~P1.7): 這 8 支腳是 8051 的 I/O 稱為 P1。第 1 腳 (P1.0) 是 LSB，第 8 腳(P1.7) 是 MSB。如果是 8052(8032，或 8752) 時，P1.0 又可當作 Timer2 的外部振波輸入腳，P1.1 又當作 T2EX，可當作另外一個外部中段觸發輸入腳。P1 上的每支腳都可推動 4 個 LS TTL。

9 腳(RESET): 8051 的重置(RESET) 輸入腳，當這支腳由外部輸入 High(+5V) 的信號時，8051 就被重置，8051 被重置後就從位址 0000H 開始執行程式。且特殊功能暫存器(SFR) 裡的所有暫存器都會被設成已知狀態。

10~17 腳(P3.0~P3.7):這 8 支腳是 8051 的 I/O ,

PORT(P3.0~P3.7)

18~19 腳:XTAL2 , XTAL1

20 腳:Vss , 系統接地腳

21-28 腳 : PORT 2 (P2.0 ~ P2.7)

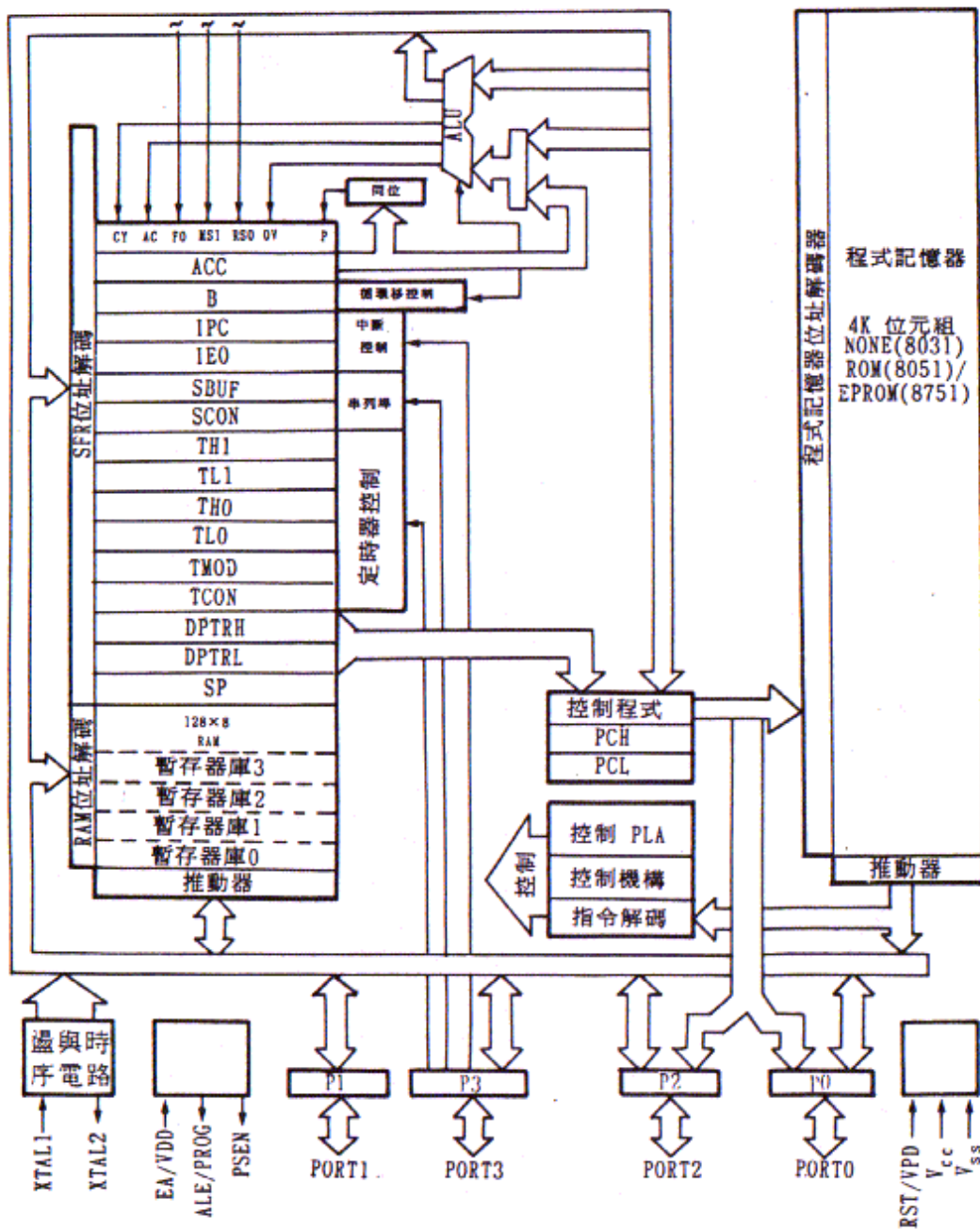
29 腳 : PSEN (程式儲存致能)(負緣觸發) : 讀取存放在外部
程式記憶體(ROM)指令

30 腳 : ALE/PROG(位址鎖定致能) , 8051 燒錄時 , 用來接受
燒錄脈波信號

31 腳 : EA/Vdd , 當接高電位時 , 若 PC 小於 4096(4K) , 執行
內部 ROM 的程式指令 , 當 PC 大於 4096(4K) , 執行外部 ROM
的程式指令。當接低電位時 , 強迫 8051 只使用外部 ROM 的程
式指令

32-39 腳 : PORT 0 (P0.0 ~ P0.7)

40 腳 : Vcc , 系統主要電源 , 接 +5V



3-5 MSC-51 的內部方塊圖

3-6 8051 的內部結構

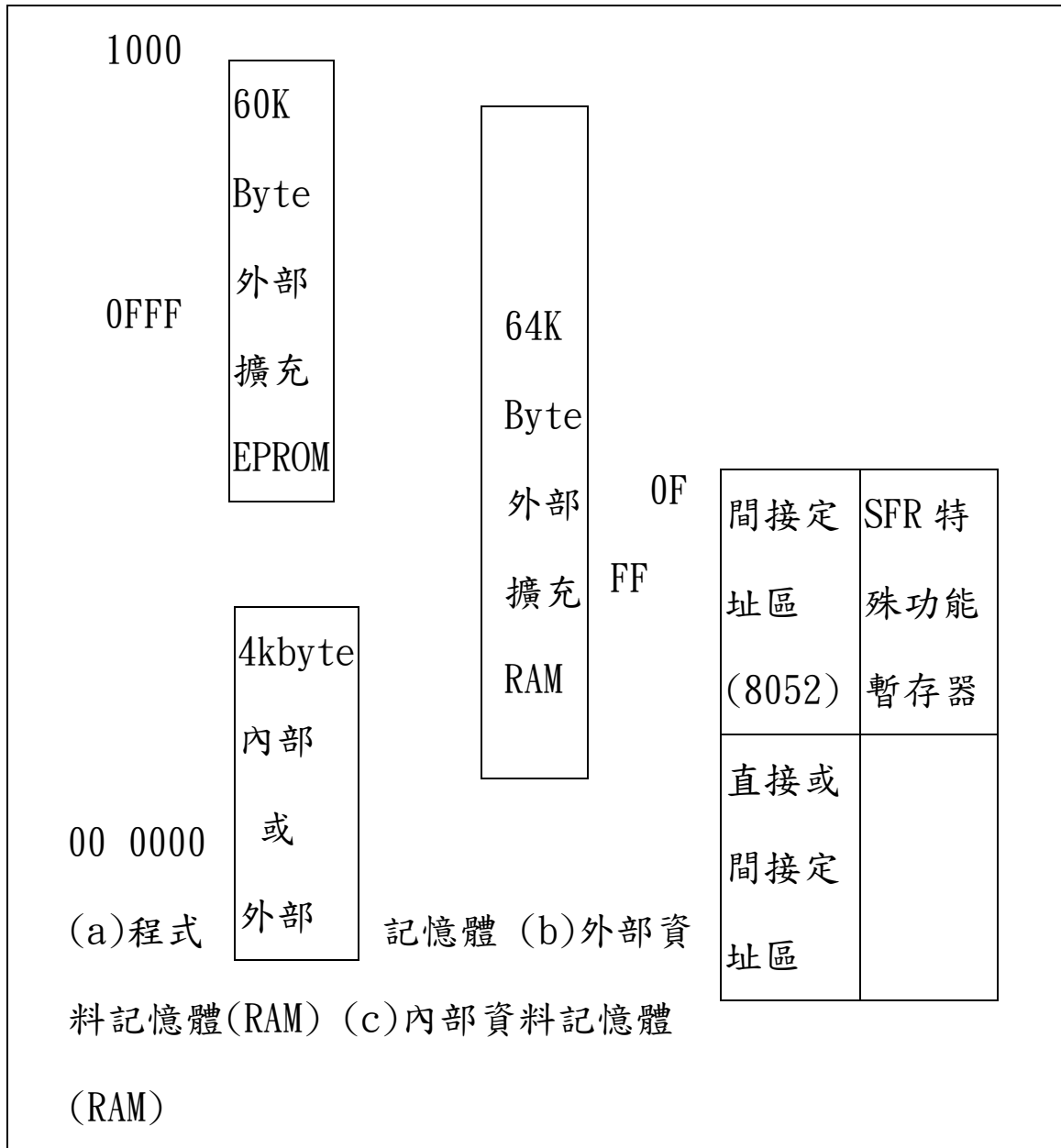
8051 單晶片微電腦的內部結構主要包括 7 個部分

- (1) 中央處理單元(CPU)
- (2) 內部程式記憶體(ROM)
- (3) 內部資料記憶體(RAM)
- (4) 振盪與時序電路
- (5) I/O 埠
- (6) 計時/計數器
- (7) 中斷控制電路

3-7 8051 的記憶體結構

8051 的記憶體可分為 3 塊獨立的記憶體，如圖：

1. 內部加上外部的程式記憶體(ROM)共 64K byte
2. 可在外部擴充 64K byte 資料記憶體(RAM)
3. 內部資料記憶體空間 256byte



8051 的記憶體映像圖

3-8 8051 指令集

算數運算指令		
指令	說明	週期
ADD A, Rn	$A \leftarrow A + Rn$	12
ADD A, direct	$A \leftarrow A + \text{direct}$	12
ADD A, @Ri	$A \leftarrow A + Ri$	12
ADD A, #data	$A \leftarrow A + \text{data}$	12
ADDC A, Rn	$A \leftarrow A + Rn + C$	12
ADDC A, direct	$A \leftarrow A + \text{direct} + C$	12
ADDC A, @Ri	$A \leftarrow A + Ri + C$	12
ADDC A, #data	$A \leftarrow A + \text{data} + C$	12
SUBB A, Rn	$A \leftarrow A - Rn - C$	12
SUBB A, direct	$A \leftarrow A - \text{direct} - C$	12
SUBB A, @Ri	$A \leftarrow A - Ri - C$	12
SUBB A, #data	$A \leftarrow A - \text{data} - C$	12
INC A	$A \leftarrow A + 1$	12
INC Rn	$Rn \leftarrow Rn + 1$	12
INC direct	$\text{direct} \leftarrow \text{direct} + 1$	12

INC @Ri	@Ri←@Ri+1	12
INC DPTR	DPTR←DPTR+1	12
DEC A	A←A-1	12
DEC Rn	Rn←Rn-1	12
DEC direct	direct←direct-1	12
DEC @Ri	@Ri←@Ri-1	12
MUL AB	兩個無符號的 8 位元資料相乘，其中高階 8 位元放入 B 暫存器，低階 8 位元則放入累積器 ACC	24
DIV AB	兩個無符號的 8 位元資料相除，把 ACC 值除以 B 暫存器值，商數放回 ACC，餘數放在 B	48
DA A	累加器作十進制調整	48

邏輯運算指令

指令	說明	週期
ANL A, Rn	$A \leftarrow A \text{ and } Rn$	12
ANL A, direct	$A \leftarrow A \text{ and } \text{direct}$	12
ANL A, @Ri	$A \leftarrow A \text{ and } Ri$	12
ANL A, #data	$A \leftarrow A \text{ and } \text{data}$	12
ANL direct, A	$\text{direct} \leftarrow \text{direct} \text{ and } A$	12
ANL direct, #data	$\text{direct} \leftarrow \text{direct} \text{ and } \text{data}$	24
ORL A, Rn	$A \leftarrow A \text{ or } Rn$	12
ORL A, direct	$A \leftarrow A \text{ or } \text{direct}$	12
ORL A, Rn	$A \leftarrow A \text{ or } Rn$	12
ORL A, @Ri	$A \leftarrow A \text{ or } Ri$	12
ORL A, #data	$A \leftarrow A \text{ or } \text{data}$	12
ORL direct, A	$\text{direct} \leftarrow \text{direct} \text{ or } A$	12
ORL direct, #data	$\text{direct} \leftarrow \text{direct} \text{ or } \text{data}$	24
XRL A, Rn	$A \leftarrow A \text{ xor } Rn$	12

XRL A, direct	A←A xor direct	12
XRL A, @Ri	A←A xor Ri	12
XRL A, #data	A←A xor data	12
XRL direct, A	direct←direct xor A	12
XRL direct, #data	direct←direct xor data	12
CLR A	清除累加器	12
CPL A	累加器反相	12
RL A	累加器向左旋轉	12
RLC A	累加器和 C 左旋	12
RR A	累加器向右旋轉	12
RRC A	累加器和 C 右旋	12
SWAP A	累加器的高低四位元互換	12

資料轉移指令

指令	說明	週期
MOV A, Rn	$A \leftarrow Rn$	12
MOV A, direct	$A \leftarrow \text{direct}$	12
MOV A, @Ri	$A \leftarrow Ri$	12
MOV A, #data	$A \leftarrow \text{data}$	12
MOV Rn, A	$Rn \leftarrow A$	12
MOV Rn, direct	$Rn \leftarrow \text{direct}$	24
MOV Rn, #data	$Rn \leftarrow \text{data}$	12
MOV direct, A	$\text{direct} \leftarrow A$	12
MOV direct, Rn	$\text{direct} \leftarrow Rn$	24
MOV direct, direct	$\text{direct} \leftarrow \text{direct}$	24
MOV direct, @Ri	$\text{direct} \leftarrow Ri$	24
MOV direct, #data	$\text{direct} \leftarrow \text{data}$	24
MOV @Ri, A	$Ri \leftarrow A$	12
MOV @Ri, direct	$Ri \leftarrow \text{direct}$	24
MOV @Ri, #data	$Ri \leftarrow \text{data}$	12

MOV DPTR, #data 16	Ri←16bit data	24
MOVC A, @A+DPTR	A←程式記憶體的資料	24
MOVC A, @A+PC	A←程式記憶體的資料	24
MOVX A, @Ri	A←外部 RAM 的資料(8bit 位址)	24
MOVX A, @DPTR	A←外部 RAM 的資料(16bit 位址)	24
MOVX @Ri, A	外部的 RAM(8bit)←A	24
MOVX @DPTR, A	外部的 RAM(16bit)←A	24
PUSH direc	推疊區←direct	24
POP direc	direct←堆疊區	24
XCH A, Rn	A 和 Rn 互換	12
XCH A, direct	A 和 direct 互換	12
XCH A, @Ri	A 和 Ri 互換	12
XCHD A, @Ri	A 和 Ri 的低四位元互換	12

位元運算指令

指令	說明	週期
CLR C	清除進位旗標	12
CLR bit	清除直接位元	12
SETB C	設定進位旗標	12
SETB bit	設定直接位元	12
CPL C	進位旗標反相	12
CPL bit	直接位元反相	12
ANL C, bit	$C \leftarrow C \text{ and bit}$	24
ANL C, /bit	$C \leftarrow C \text{ and bit(反相)}$	24
ORL C, bit	$C \leftarrow C \text{ or bit}$	24
ORL C, /bit	$C \leftarrow C \text{ or bit(反相)}$	24
MOV C, bit	$C \leftarrow \text{bit}$	12
MOV bit, C	$\text{bit} \leftarrow C$	24
JC rel	若 $C=1$ 跳至 rel	24
JNC rel	若 $C=0$ 跳至 rel	24
JB bit, rel	若 $\text{bit}=1$ 跳至 rel	24
JNB bit, rel	若 $\text{bit}=0$ 跳至 rel	24
JBC bit, rel	若 $\text{bit}=1$ 跳至 rel, 且清除此位元	24

程式跳躍指令

指令	說明	週期
ACALL addr11	絕對式副程式呼叫	24
LCALL addr16	遠程副程式呼叫	24
RET	從副程式返回	24
RETI	從中斷副程式返回	24
AJMP addr11	絕對式跳躍	24
LJMP addr16	遠程跳躍	24
SJMP rel	短程跳躍	24
JMP @A+DPTR	間接跳躍	24
JZ rel	若 A=0 跳至 rel	24
JNZ rel	若 A 不等於 0 跳至 rel	24
CJNE A, direct, rel	若 A 不等於 direct 跳至 rel	24
CJNE A, #data, rel	若 A 不等於 data 跳至 rel	24
CJNE Rn, #data, rel	若 Rn 不等於 data 跳至 rel	24
CJNE	若 Ri 不等於 data 跳至 rel	24

@Ri, #data, rel		
DJNZ Rn, rel	Rn 減 1 不等於 0 跳至 rel	24
DJNZ direct, rel	direct 減 1 不等於 0 跳至 rel	24
NOP	沒動作	12

	說明縮寫符號	備註
縮寫符號	說明	
Rn	暫存器 R0-R7	
direct	8bit 內部資料記憶體，包括 1. 內部資料記憶體(00-7F)的位址 2. 特殊功能暫存器(80-FF)的位址，如 P0, PSW, TMOD.. 等	
@Ri	由暫存器 R0 或 R1 所定址的內部 RAM 資料	
#data	8bit 常數	
#data 16	16bit 常數	
addr 16	16bit 的目的位址，可使跳躍指令跳躍 64k	
addr 11	11bit 的目的位址，可使跳躍指令跳躍 2k	
rel	具正負號的 8 位元位址偏移量，用於相對位址的跳躍	
bit	1 個 bit: 只所有可以位元定址的位元。(如 PSW. 7 為 CY 旗號)	
A	累加器(累積器)	
C 或 CY	進位旗號	
AC	輔助進位旗號	

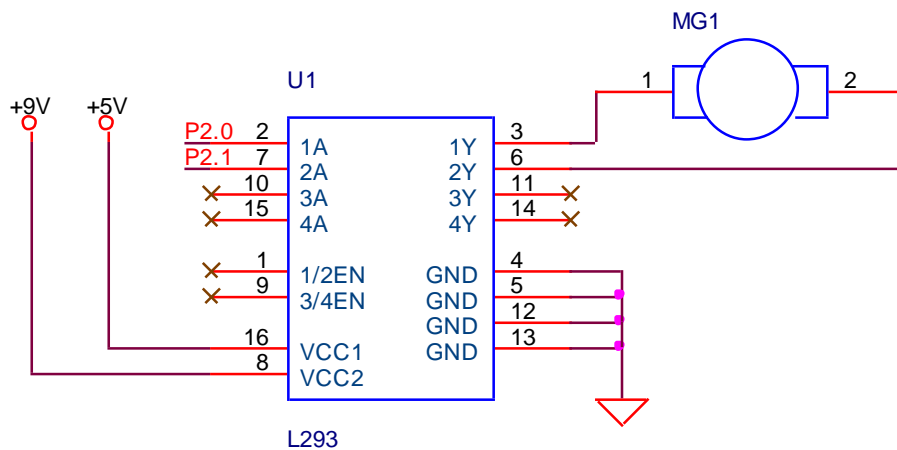
Bb	指定位元 B0-B7	
D	半位元組(4Bit)助憶符號	
F0	旗號 0	
I	中斷	
PC	程式計數器	
P0	I/O port 0	
P1	I/O port 1	
P2	I/O port 2	
P3	I/O port 3	
PSW	程式狀態字組(類似於旗號暫存器)	
SP	堆疊指標暫存器	
B	暫存器 B	
DPTR	資料指標暫存器	
X	外部次要記憶體的助憶符號	
@	間接位址符號	
\$	程式計數迄目前的值	
reg	暫存器	

肆、馬達介紹

4-1 馬達驅動電路

選用 9V 的直流馬達，直流馬達有兩個端，一為正端，一為負端，只要在兩線端輸入直流電壓，便可使馬達轉動，若輸入電壓極性正接可使馬達轉動，若輸入電壓極性反接，則可使馬達逆轉

正反轉控制原理如圖：



伍、8051 程式碼

5-1 8051 程式碼

用筆記本寫 8051 組合語言程式，程式完成時，另存新檔為 ****.ASM。

程式碼如下：

各接腳定義 P1.0 上開鍵，P1.2 下關鍵，P1.1 OFF 鍵，P1.3 上開 SENSOR 鍵，P1.4 下關 SENSOR 鍵

```
ORG 00H

MOV    P1, #11111111B    ;把P1設成高電位，所有動作皆負
緣觸發

SETB   P2.0             ;馬達停止

SETB   P2.1

; -----
;  --      主程式      --
; -----

START: SETB   P1.7      ;靜音

SETB   P2.0             ;馬達停止

SETB   P2.1

JNB    P1.0, DOORUP    ;J為跳，N為否，B為1，也就是說
```

;當P1.0腳不為1則跳到DOORUP

;副程式，若為1表示無按下按鍵，

;繼續下一行程式，不跳到DOORUP

JNB P1.2, DOORDOWN

;J為跳，N為否，B為1，也就是說

;當P1.2為0則跳到DOORDOWN

;副程式，若為1表示無按下按鍵，

;繼續下一行，不跳到DOORDOWN

AJMP START ;跳到START

; -----

; -- 開門副程式 --

; -----

DOORUP:

CLR P1.7 ;蜂鳴器響

JNB P1.0, DOORUP ;從JNB P1.0, DOORUP這行指

SETB P1.7 ;令跳過來後，若手指按下按鍵還沒

;放開時，P1.0會一直為0，當放開

;P1.0時，為1，則往下一行

SETB P2.1 ;馬達轉動，開門

CLR P2.0

JNB P1.1, DUPB ;P1.1是停止鍵，有按下為0，跳到DUPB

JNB P1.3, START ;P1.3腳為最頂的磁簧開關，當門到

;最頂時，磁簧開關會被門片裡的磁鐵吸住，使

P1.3由HIGH變LOW

AJMP DOORUP ;還沒到最頂則跳到DOORUP，馬達

;保持轉動

DUPB:

SETB P2.0 ;馬達停止

SETB P2.1

CLR P1.7 ;蜂鳴器響

JNB P1.1, \$;等待放開按鍵

SETB P1.7 ;蜂鳴器停

JMP START

; -----
; -- 關門副程式 --
; -----

DOORDOWN:

CLR P1.7 ;蜂鳴器響

JNB P1.2, DOORDOWN

SETB P1.7 ;從JNB P1.2, DOORDOWN 這行

;跳過來後，若手指按下按鍵還沒

;放開時，P1.2會一直為0，當放開

;P1.2時，為1，則往下一行

SETB P2.0 ;馬達轉動，關門

CLR P2.1

JNB P1.1, DUPB ;P1.1是停止鍵，有按下為0，跳回

;START主程式

JNB P1.4, START ;P1.4腳為最底的磁簧開關，當門到

;最底時，磁簧開關會被門片裡的磁鐵吸住，使

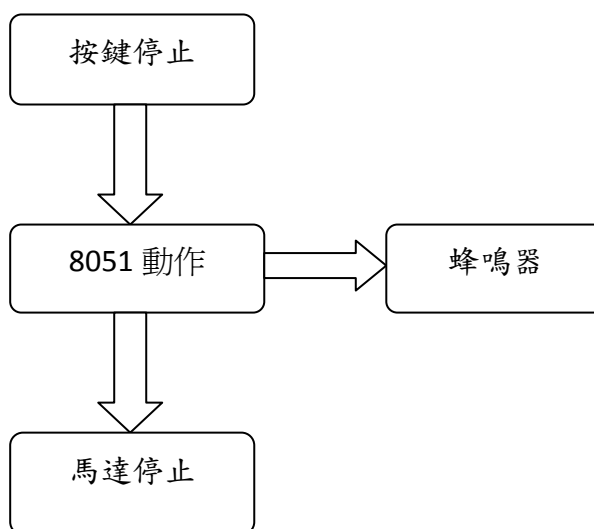
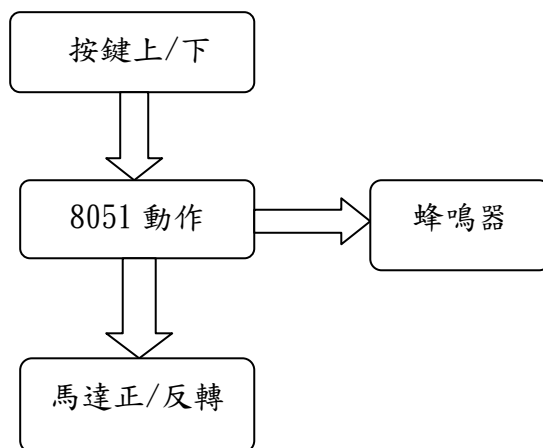
P1.4由HIGH變LOW

AJMP DOORDOWN ;還沒到最底則跳到DOORDOWN，

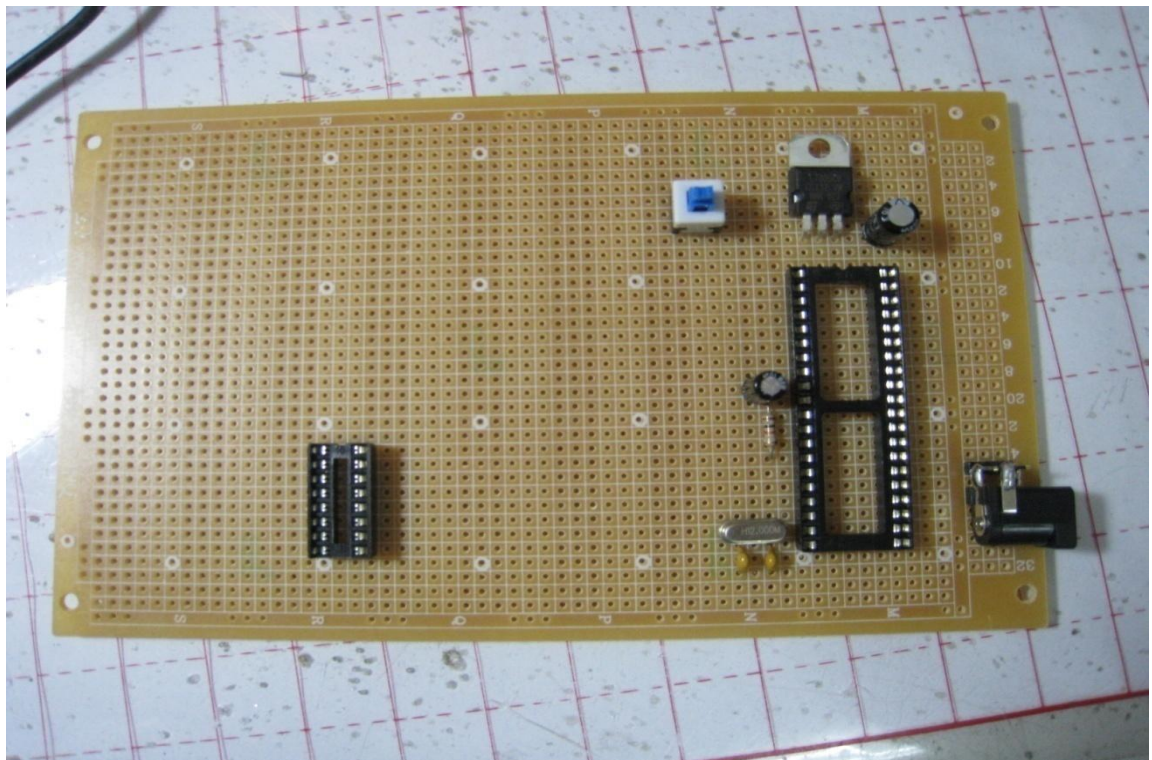
;馬達保持轉動

END

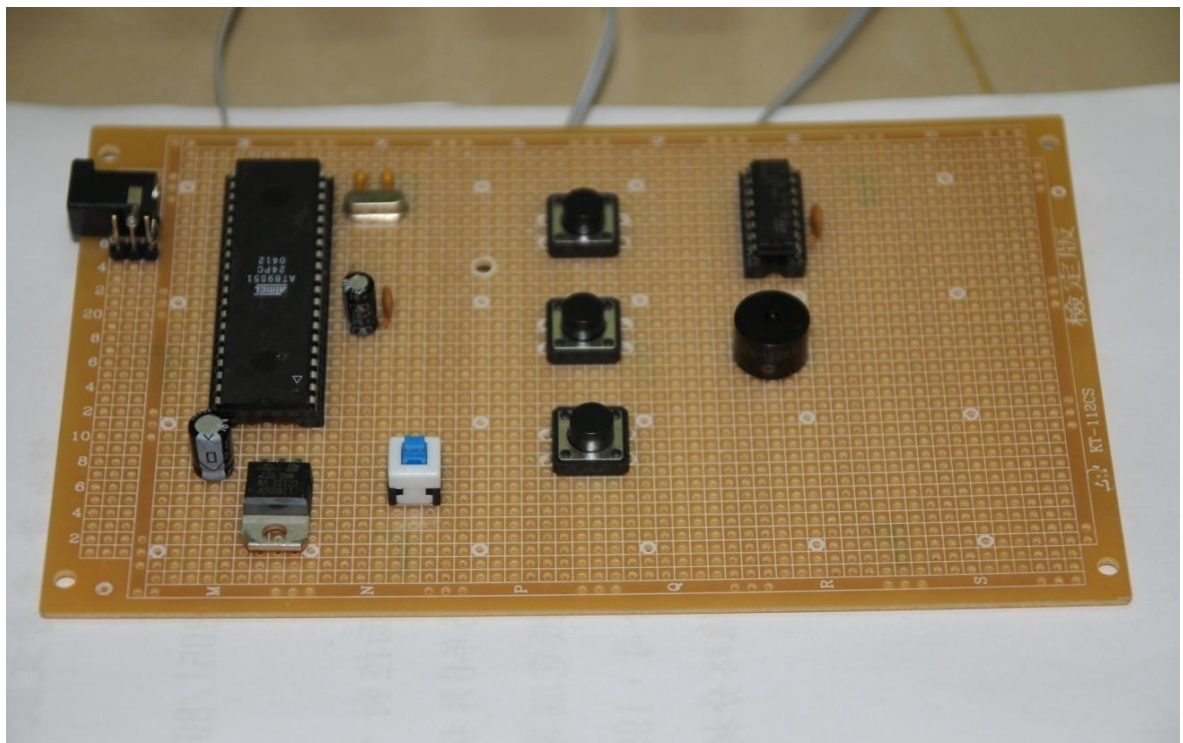
5-2 鐵捲門作動簡易圖



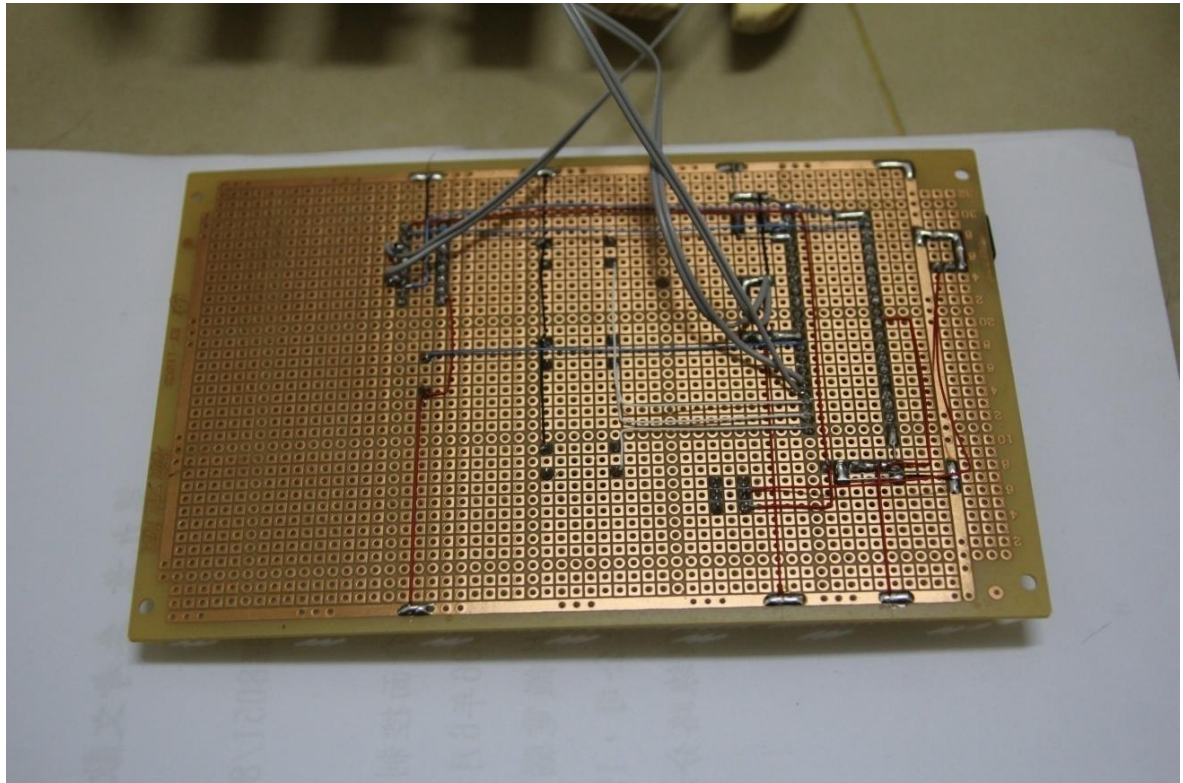
陸、製作過程



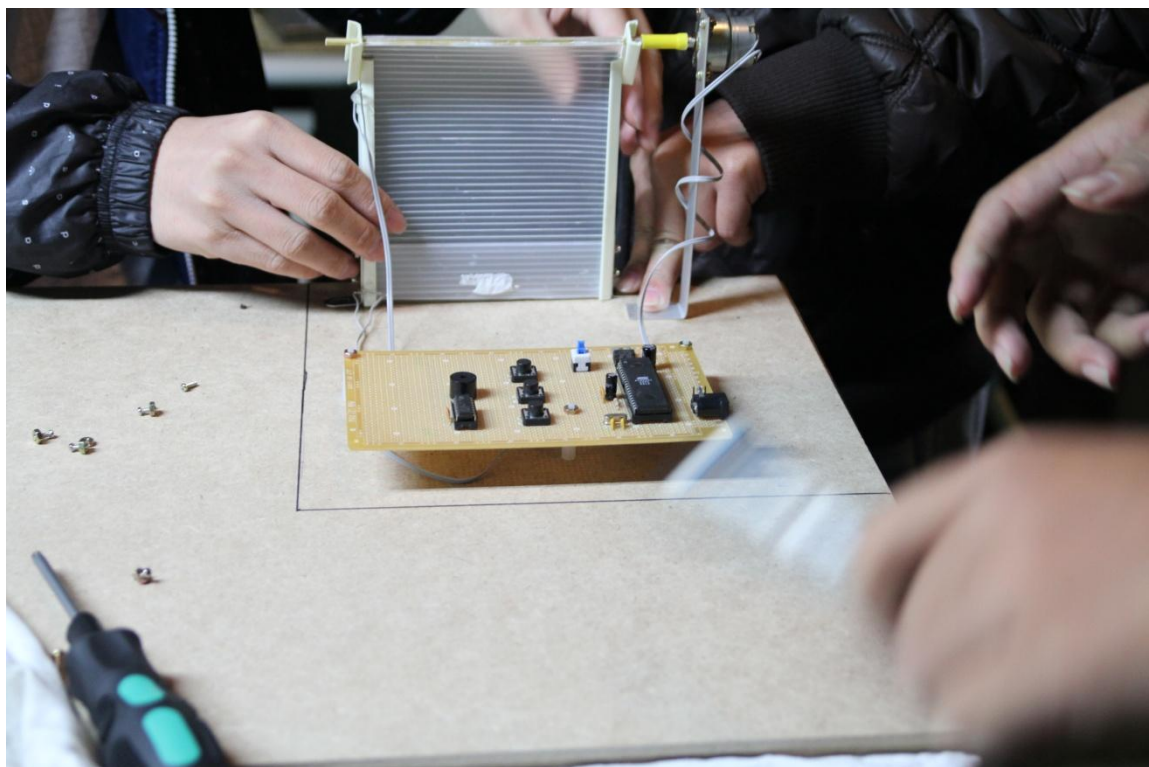
電路板製作圖



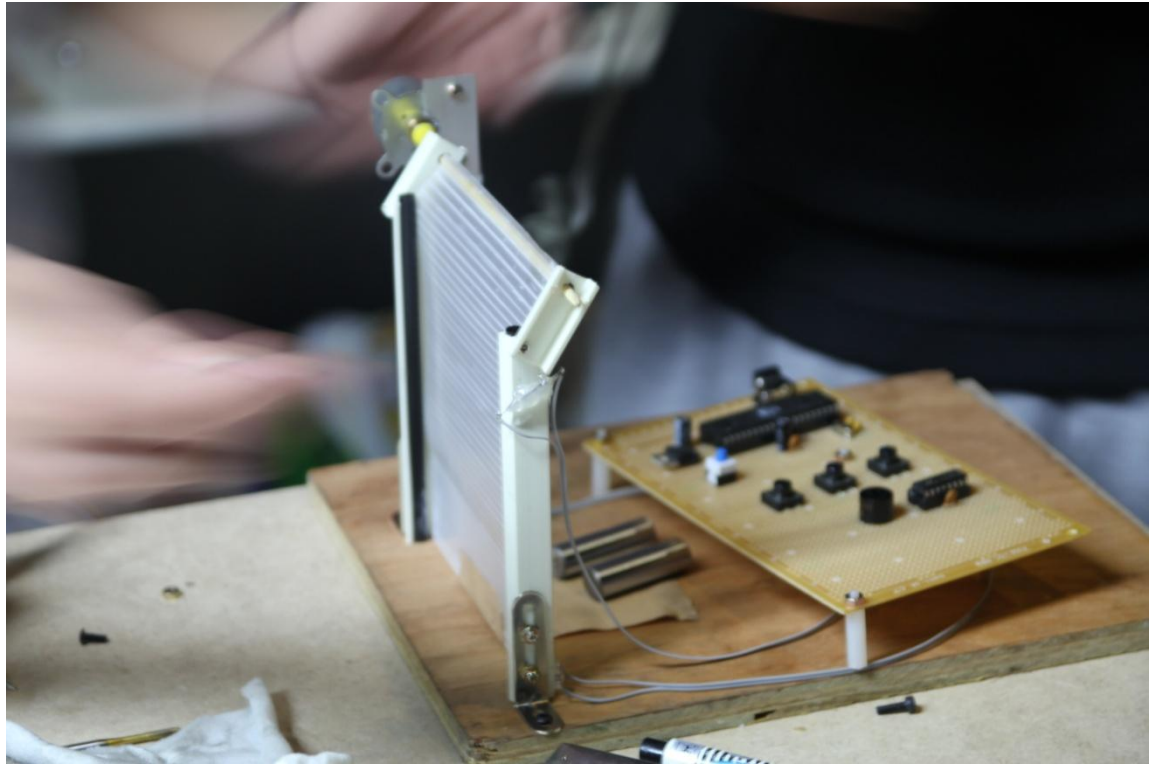
電路板製作圖



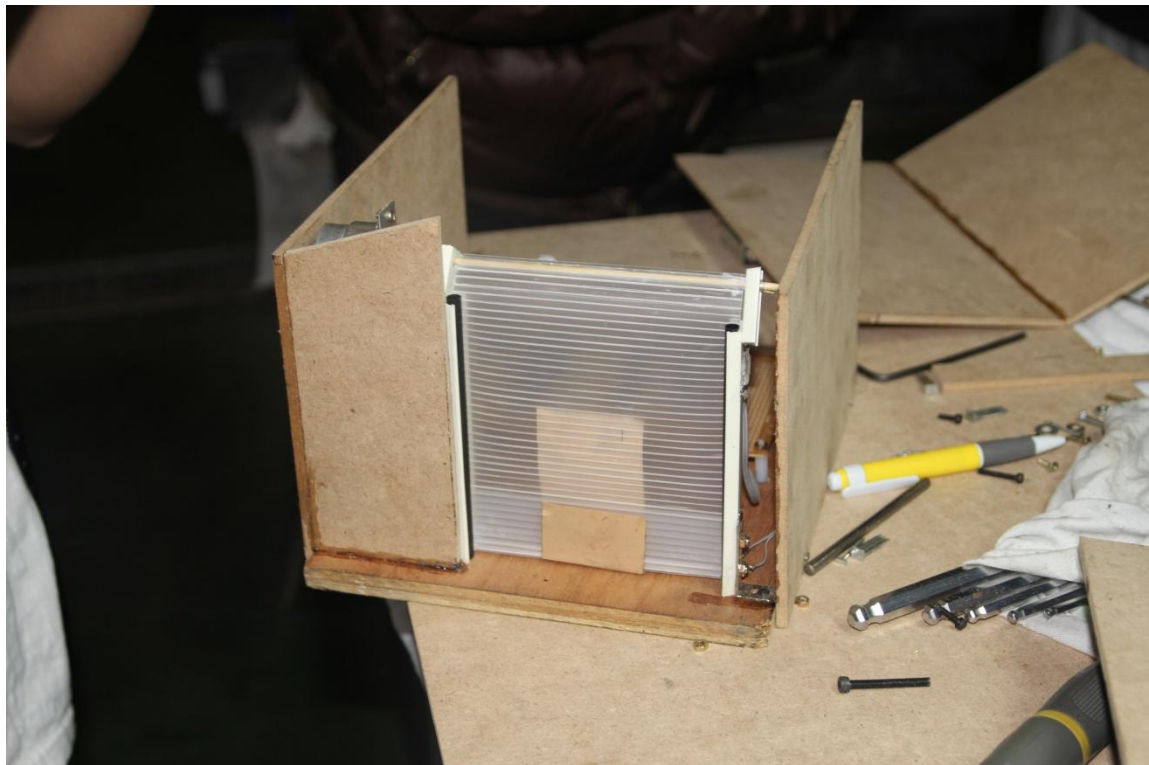
焊接電路板



製作外觀



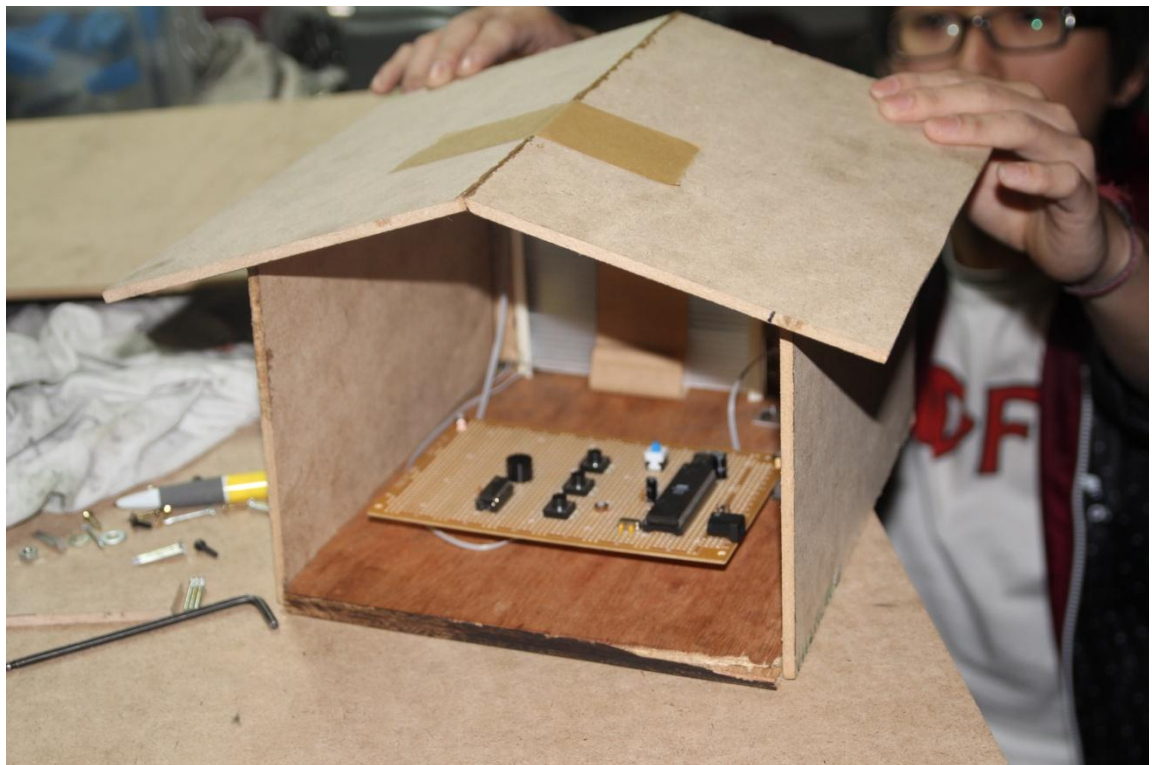
製作外觀過程



製作外觀過程



製作外觀內部圖

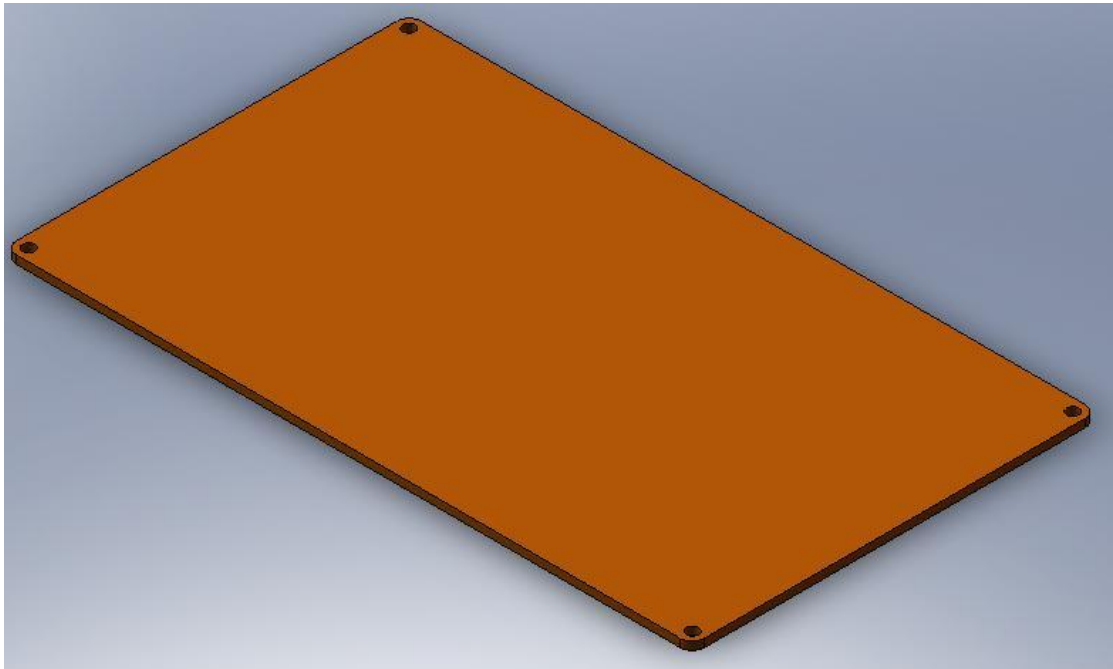


製作屋頂

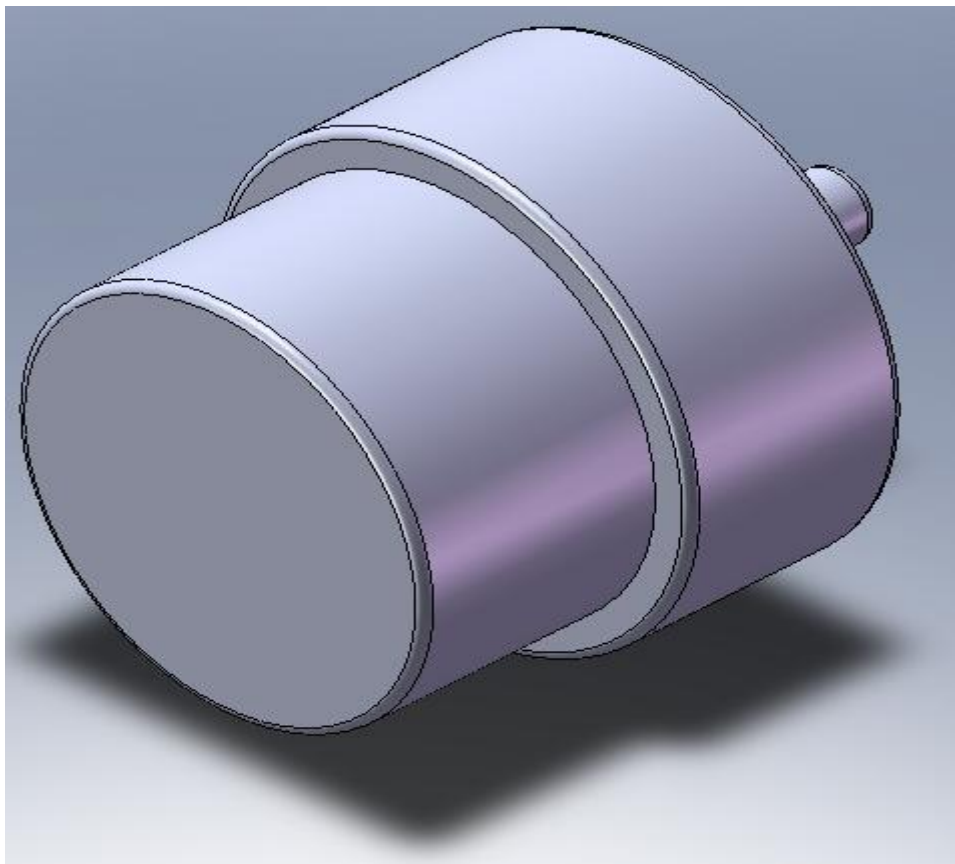


彩繪完成圖

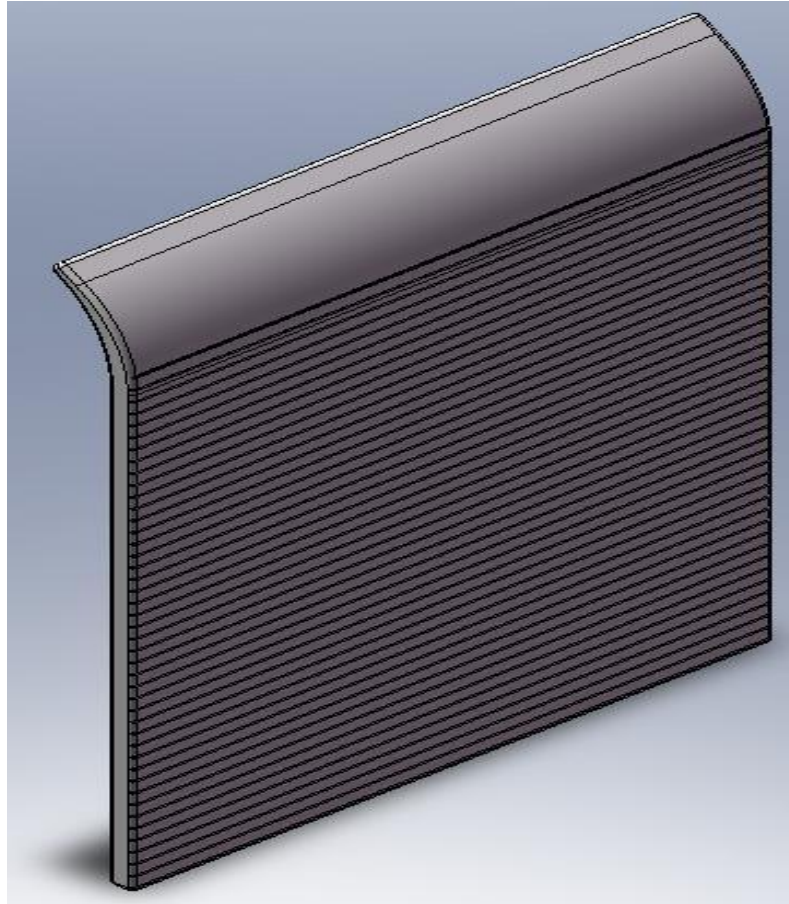
6.2 Solid works 草圖繪製



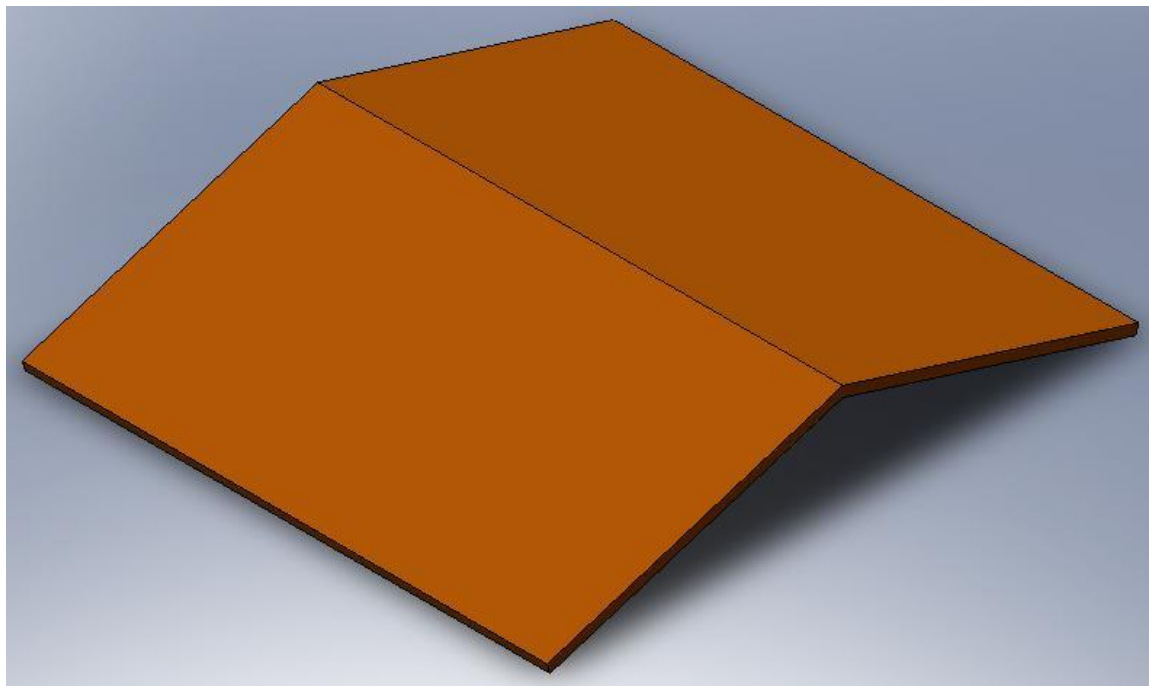
電路板



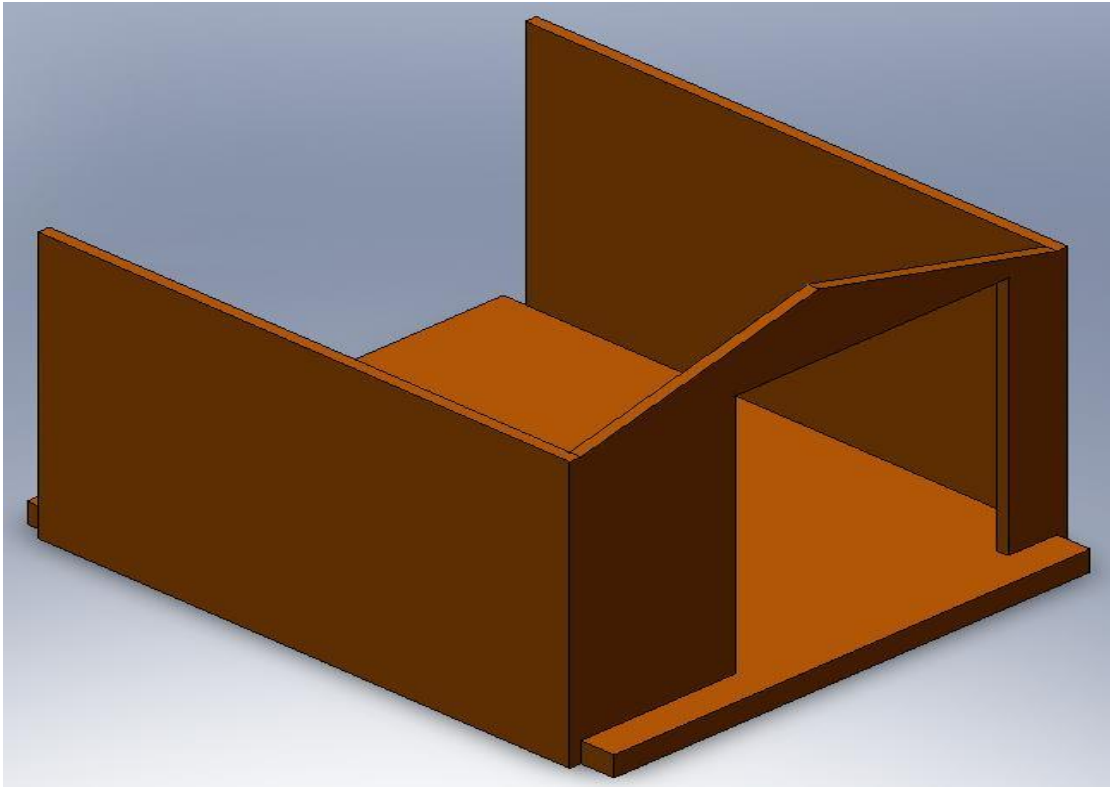
馬達



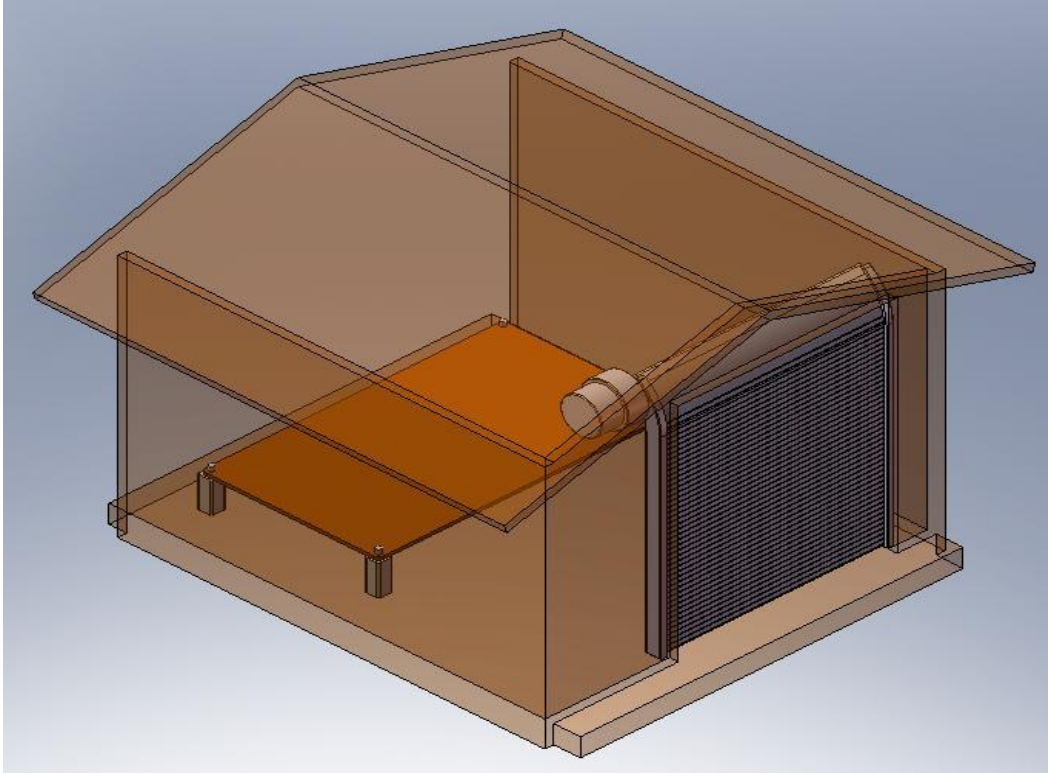
電捲門



屋頂



房屋外觀



完成圖

參考文獻

1. 書名:8051 單晶片實作寶典

作者: 鄧錦城 編著

出版社:宏友圖書開發股份有限公司

2. 書名:8051 單晶片專題製作

作者: 鄧錦城 編著

出版社:宏友圖書開發股份有限公司

3. 書名:8051 入門與介面控制

作者:陳龍三編著

出版社:松崗電腦圖書資料股份有限公司

4. 書名:MSC-51 單晶片微電腦專題製作

作者:王信福

出版社:松崗電腦圖書資料股份有限公司