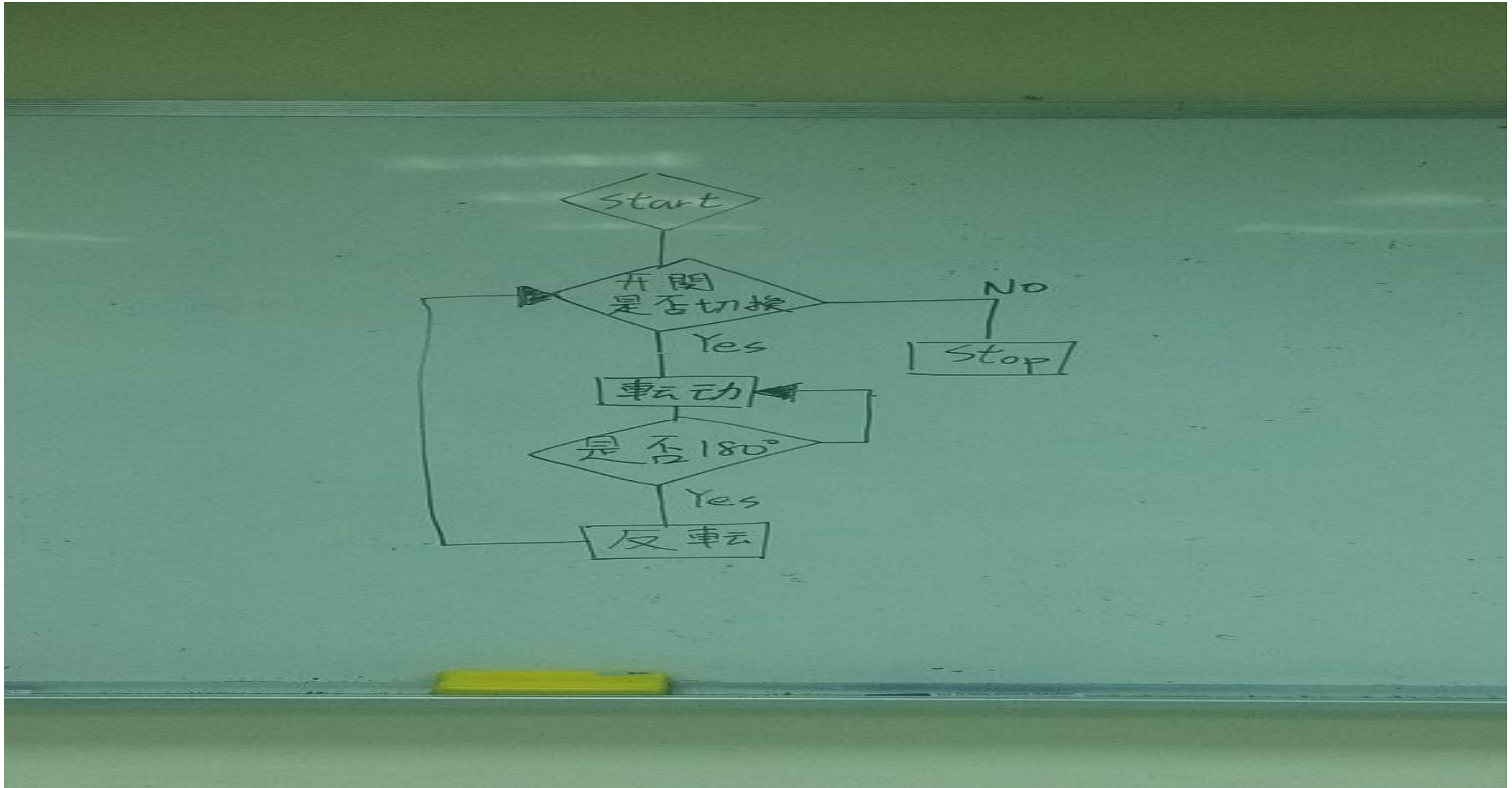
PWM創意馬達

愛國者控制飛彈

組員

- b09601103　何建儒
- B09901041　鄧金國
- B09801214　張庭維
- B09801111　謝仁翔
- B09901111　杜彥寬
- B09901205　邱俊豪
- B09901217　林念澤
- B09901302　黃建智
- B09901082　李俊宏

架構



程式

```
1  module mode555(clock,reset,OUT);
2  input  clock,reset;
3  output OUT;
4  reg    OUT;
5  reg    [9:0]A;
6  always@(posedge clock,negedge reset)
7      begin
8          OUT=A[9];
9          if(!reset);
10         else if (A==10'd554)A<=0;
11             else A<=A+1;
12     end
13 endmodule
14
```

```
1  module mode225(clock,reset,A,OUT);
2  input  clock,reset;
3  output [7:0]A;
4  reg    [7:0]A;
5  output OUT;
6  reg    OUT;
7  always@(posedge clock,negedge reset)
8      begin
9          OUT=A[7];
10         if(!reset);
11         else if (A==8'd224)A<=0;
12             else A<=A+1;
13     end
14 endmodule
```

```
module contral(out,clk,reset);
input  clk,reset;
output [7:0]out;
reg    [3:0]sw;
reg     [6:0]cunt;
reg    [7:0]out;
reg    [3:0]Q;
reg     s;
always@(posedge clk,negedge reset)
    begin
        if(!reset)  Q<=4'd0;
        else if (Q==4'd9)
                    begin
                            Q<=4'd0;
                            s=1;
                    end
        else
            begin
                Q<=Q+1;
                s=0;
            end
    end
/*always@(sw)
    begin
        case(sw)
            4'd0:out=8'd90;
            4'd1:out=8'd60;
            default:out=8'b00000000;
        endcase
    end*/
always@(posedge Q[3],negedge reset)
    begin
        if (!reset)
            cunt<=0;
        else if (cunt==7'd99)
            cunt<=7'd0;
        else cunt<=cunt+1;
    end
always@(cunt)
    begin
        case(cunt)
            7'd0    :out=8'd0;
            7'd1    :out=8'd0;
            7'd2    :out=8'd0;
            7'd3    :out=8'd0;
            7'd4    :out=8'd0;
            7'd5    :out=8'd0;
            7'd6    :out=8'd0;
            7'd7    :out=8'd0;
            7'd8    :out=8'd0;
            7'd9    :out=8'd0;
            7'd10   :out=8'd0;
            7'd11   :out=8'd0;
            7'd13   :out=8'd0;
            7'd14   :out=8'd0;
            7'd15   :out=8'd0;
            7'd16   :out=8'd0;
            7'd17   :out=8'd0;
            7'd18   :out=8'd0;
            7'd19   :out=8'd0;
            7'd20   :out=8'd0;
            7'd21   :out=8'd0;
            7'd22   :out=8'd0;
            7'd23   :out=8'd0;
            7'd24   :out=8'd0;
            7'd25   :out=8'd0;
            7'd26   :out=8'd0;
            7'd27   :out=8'd0;
            7'd28   :out=8'd0;
            7'd29   :out=8'd0;
            7'd30   :out=8'd0;
            7'd31   :out=8'd0;
            7'd32   :out=8'd0;
            7'd33   :out=8'd0;
            7'd34   :out=8'd0;
            7'd35   :out=8'd0;
            7'd36   :out=8'd0;
            7'd37   :out=8'd0;
            7'd38   :out=8'd0;
            7'd39   :out=8'd0;
            7'd40   :out=8'd0;
            7'd41   :out=8'd0;
            7'd42   :out=8'd0;
            7'd43   :out=8'd0;
            7'd44   :out=8'd0;
            7'd45   :out=8'd0;
            7'd46   :out=8'd0;
            7'd47   :out=8'd0;
            7'd48   :out=8'd0;
            7'd49   :out=8'd0;
            /*6'd50 :out=8'd0;
            6'd51   :out=8'd0;
            6'd52   :out=8'd0;
            6'd53   :out=8'd0;
            6'd54   :out=8'd0;
            6'd55   :out=8'd0;
            6'd56   :out=8'd0;
            6'd57   :out=8'd0;
            6'd58   :out=8'd0;
            6'd59   :out=8'd0;
            6'd60   :out=8'd0;
            6'd61   :out=8'd0;
            6'd62   :out=8'd0;
            6'd63   :out=8'd0;
            6'd64   :out=8'd0;
            6'd65   :out=8'd0;
            6'd66   :out=8'd0;
            6'd67   :out=8'd0;
            6'd68   :out=8'd0;
            6'd69   :out=8'd0;
            6'd70   :out=8'd0;
            6'd71   :out=8'd0;
            6'd72   :out=8'd0;
            6'd73   :out=8'd0;
            6'd74   :out=8'd0;
            6'd76   :out=8'd0;
            6'd77   :out=8'd0;
            6'd80   :out=8'd1;
            6'd81   :out=8'd1;
            6'd82   :out=8'd1;
            6'd83   :out=8'd1;
            6'd84   :out=8'd1;
            6'd85   :out=8'd1;
            6'd86   :out=8'd1;
            6'd87   :out=8'd1;
            6'd88   :out=8'd1;
            6'd89   :out=8'd1;
            6'd90   :out=8'd1;
            6'd91   :out=8'd1;
            6'd92   :out=8'd1;
            6'd93   :out=8'd1;
            6'd94   :out=8'd1;
            6'd95   :out=8'd1;
            6'd96   :out=8'd1;
            6'd97   :out=8'd1;
            6'd98   :out=8'd1;*/
            default:out=8'd179;
        endcase
```

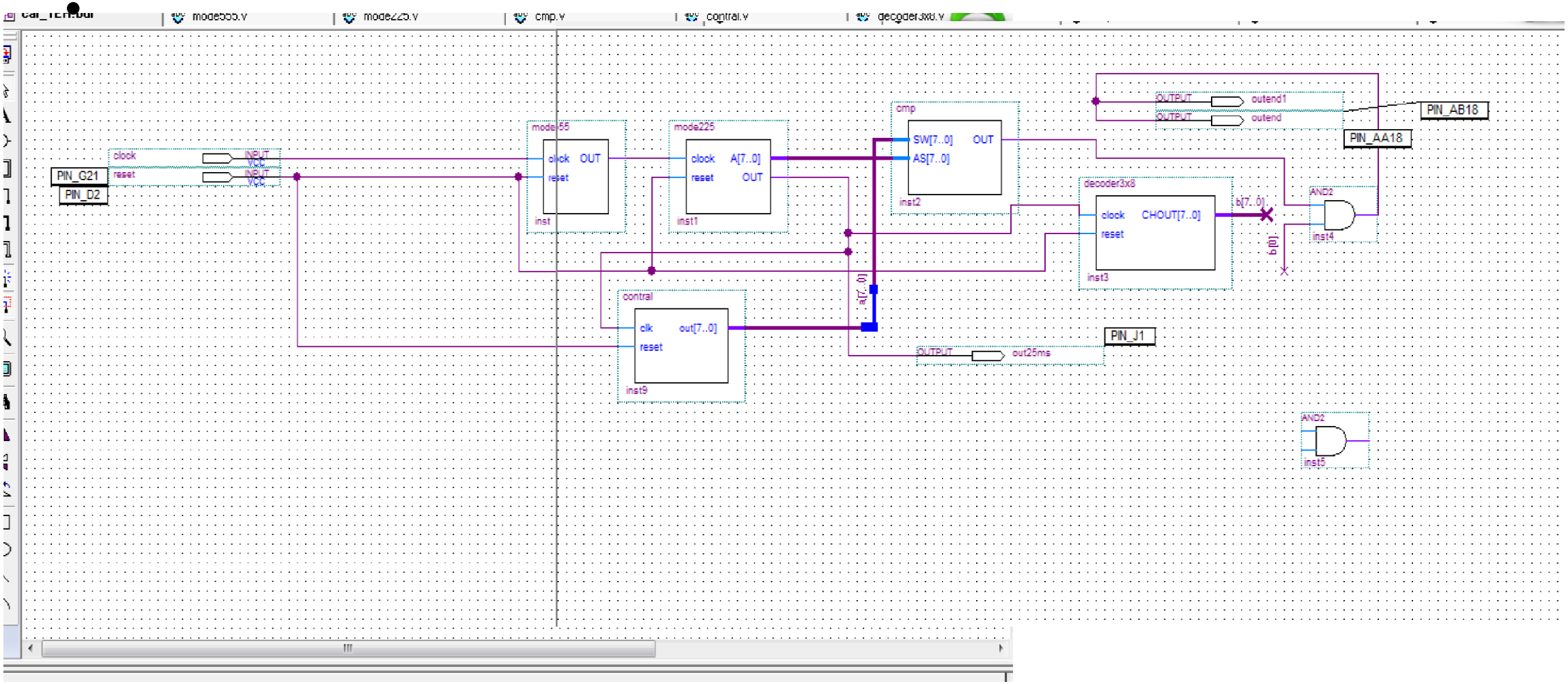
1ER.bdf* | mode555.v | mode225.v | cmp.v

```
1  module decoder3x8(clock,reset,CHOUT);
2  input  clock,reset;
3  output [7:0]CHOUT;
4  reg    [7:0]CHOUT;
5  reg    [2:0]A;
6  always@(posedge clock,negedge reset)
7      begin
8          if(!reset);
9          else A<=A+1;
10     end
11 always@(A)
12     begin
13         case(A)
14         3'd0:CHOUT=8'b00000001;
15         3'd1:CHOUT=8'b00000010;
16         3'd2:CHOUT=8'b00000100;
17         3'd3:CHOUT=8'b00001000;
18         3'd4:CHOUT=8'b00010000;
19         3'd5:CHOUT=8'b00100000;
20         3'd6:CHOUT=8'b01000000;
21         default:CHOUT=8'b10000000;
22         endcase
23     end
24 endmodule
```

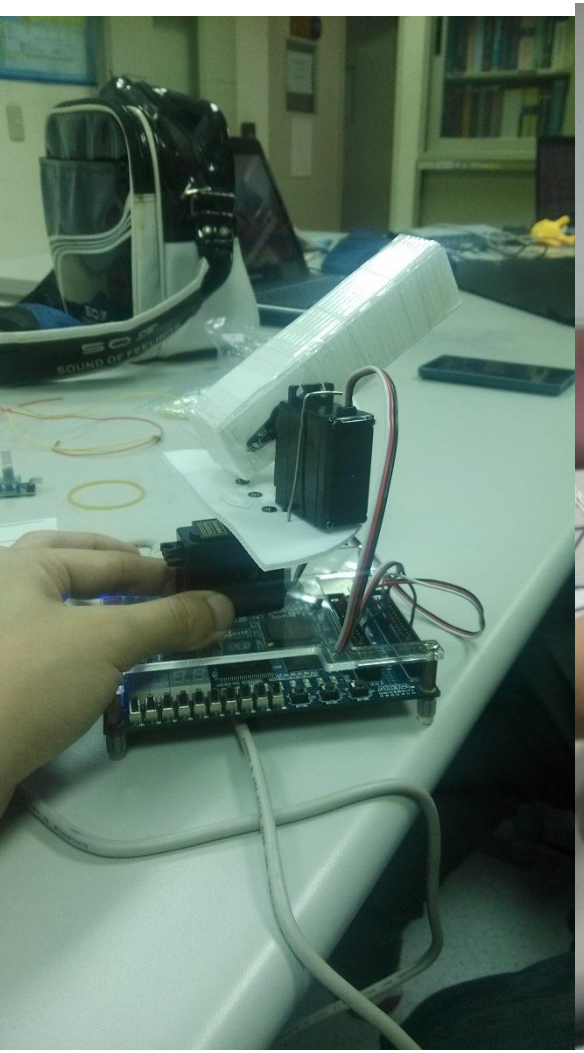
r_1ER.bdf* | mode555.v | mode225.v

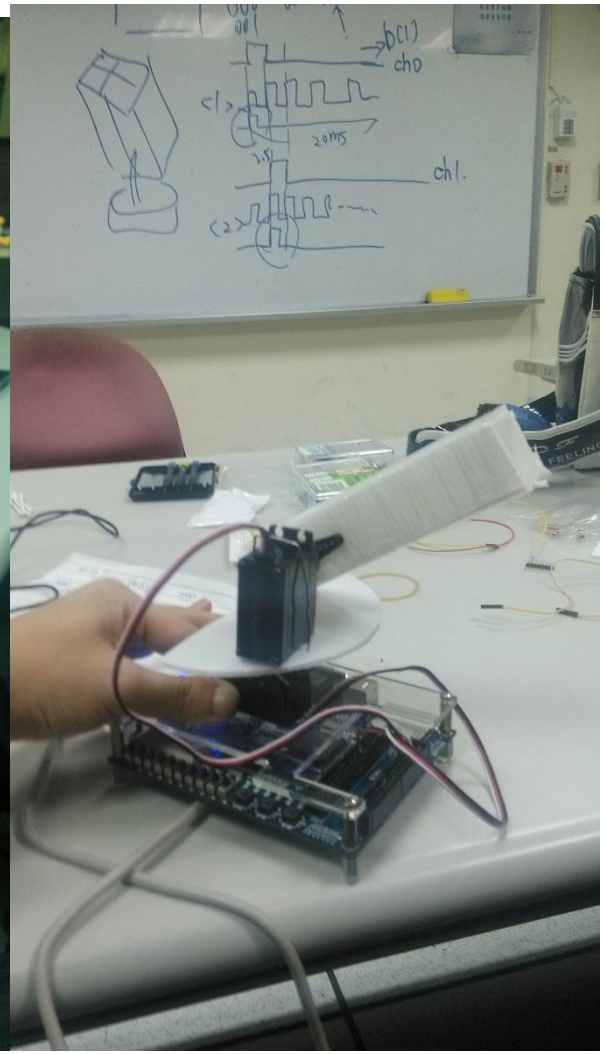
```
1  module cmp(SW,AS,OUT);
2  input  [7:0]SW;
3  input  [7:0]AS;
4  output OUT;
5  reg    [7:0]SW1;
6  reg    OUT;
7  always@(SW,AS)
8      begin
9          SW1={SW}+8'd45;
10         if(SW1>AS) OUT=1;
11         else OUT=0;
12     end
13 endmodule
```

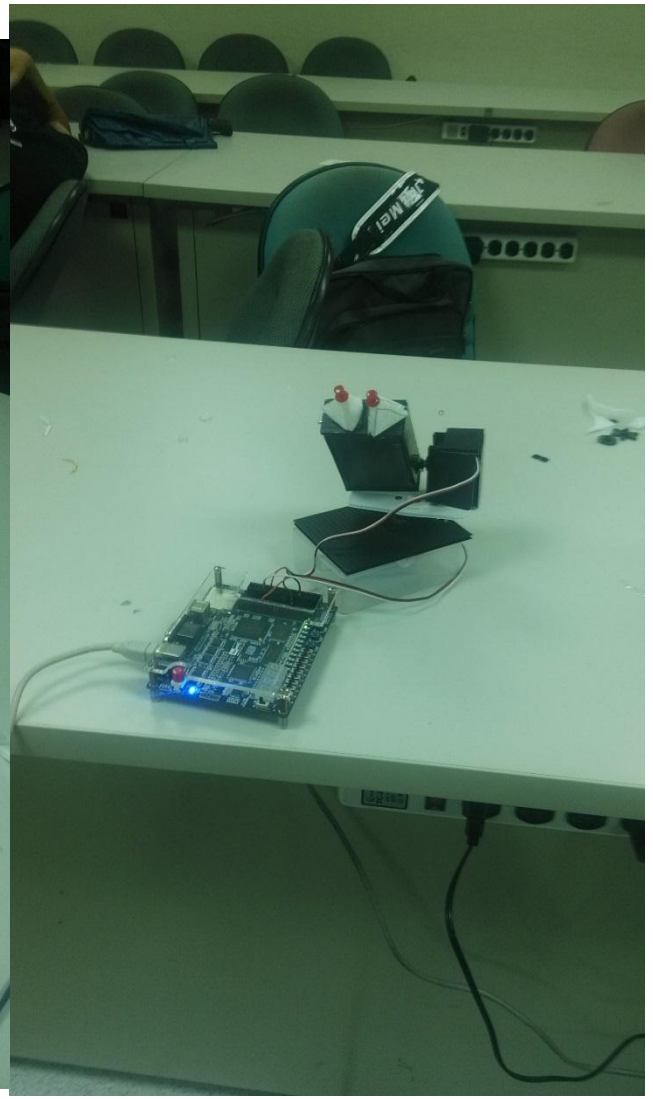
電路圖


mode55.v
mode225.v
cmp.v
contral.v
decoder3x8.v
clock
reset
PIN_G21
PIN_D2
INPUT
VCC
mode_55
clock
reset
OUT
inst
mode225
clock
reset
A[7..0]
OUT
inst1
cmp
SW[7..0]
AS[7..0]
OUT
inst2
contral
clk
reset
out[7..0]
inst9
a[7..0]
decoder3x8
clock
reset
CHOUT[7..0]
inst3
b[7..0]
b[0]
OUTPUT
outend1
outend
out25ms
PIN_AB18
PIN_AA18
PIN_J1
AND2
inst4
inst5

成品圖流程圖







影片