

第十三組-怪獸鴨

B10201026 許竣翔

主程式

```
module proje(clock,reset,out25ms,out,outh,outend,LED,Dout);
input clock,reset;
output out25ms;
output out,outh,outend,Dout;
output [6:0]LED;
wire k;
wire [7:0] q;
wire [7:0] SW;

mode555 U1( .clock(clock),
            .reset(reset),
            .OUT(k)
            );

mode225 U2( .clock(k),
            .reset(reset),
            .q(q),
            .OUT(out25ms)
            );

mode15 U3 ( .clk(k),
            .reset(reset),
            .Q(SW)
            );

cmp U4 ( .W(SW),
        .X(q),
        .OUT(out)
        );

decoder3x8 U5 ( .clock(out),
               .reset(reset),
               .DEOUT(outh)
               );

result U6 ( .x(out),
           .y(outh),
           .OUT(outend)
           );

Mn U7 ( .F50M(clock),
       .reset(reset),
       .Dout(Dout),
       .LED(LED)
       );
endmodule
```

各個小程式-mode555

- `module mode555(clock,reset,OUT);`
- `input clock,reset;`
- `output OUT;`
- `reg OUT;`
- `reg [9:0]q;`
- `always@(negedge reset,posedge clock)`
- `begin`
- `OUT = q[9];`
- `if(!reset);`
- `else if (q==10'd554) q<=0;`
- `else q<=q+1;`
- `end`
- `endmodule`

mode225

- module mode225(clock,reset,q,OUT);
- input clock,reset;
- output [7:0]q;
- reg [7:0]q;
- output OUT;
- reg OUT;
- always@(negedge reset,posedge clock)
- begin
- OUT = q[7];
- if(!reset);
- else if (q==8'd224) q<=0;
- else q<=q+1;
- end
- endmodule

mode15

- module mode15(clk,reset,Q);
- input clk,reset;
- output [7:0]Q;
- reg [7:0]Q;
- reg [9:0]q;

- always@(negedge reset,posedge clk)
- begin
- if (reset==0) q<=9'd0;
- else
- if(q==10'd1500) q<=9'd0;
- else q<=q+1;

- end
- always@(negedge reset,posedge q[9])
- begin
- if (reset==0) Q<=9'd0;
- else
- if(Q==9'd225) Q<=9'd0;
- else Q<=Q+1;

- end
- endmodule

cmp

- module cmp(W,X,OUT);
- input [7:0]W;
- input [7:0]X;
- output OUT;
- reg OUT;
- reg [7:0]W1;
- always@(W,X)
- begin
- W1 = {W}+8'd45;
- if(W1>X) OUT=1;
- else OUT=0;
- end
- endmodule

decoder3x8

- module decoder3x8(clock,reset,DEOUT);
- input clock,reset;
- output [7:0]DEOUT;
- reg [7:0]DEOUT;
- reg [2:0] q;

- always@ (negedge reset,posedge clock)
- begin
- if(!reset) q<=0;
- else q<=q+1;
- end

- always@(q)
- begin
- case(q)
- 3'd0 : DEOUT = 8'b00000001;
- 3'd1 : DEOUT = 8'b00000010;
- 3'd2 : DEOUT = 8'b00000100;
- 3'd3 : DEOUT = 8'b00001000;
- 3'd4 : DEOUT = 8'b00010000;
- 3'd5 : DEOUT = 8'b00100000;
- 3'd6 : DEOUT = 8'b01000000;
- default : DEOUT=8'b10000000;
- endcase
- end
- endmodule

result

- module result (x,y,OUT);
- input x;
- input [7:0]y;
- output OUT;
- reg OUT;
- always@(x,y)
- begin
- if (y[0]==1&&x==1) OUT=1;
- else OUT=0;
- end
- endmodule

架構介紹

- mode555 : $555 > 512 = 2^9$
- 位元為[9:0]
- 因為要把起始值當為 0 所以 $555-1 = 554$
- 555 的來源為藉由計算最後出來的2.5ms，往前推所得的值

- Mode225 : $225 < 256 = 2^8$
- 位元為[7:0]
- 因為要算 0 所以 $225-1 = 224$
- 225的來源為 :我們所要轉動的180度角，在加上為了維持 0 ms，所需的45度角，相加為225

架構介紹

- mode 15 : 為代替 SW 輸入的計數器值，藉由在其中另設一個頻率，讓計數器的值，由 0 ~ 180 度轉動，依我們所設的頻率，來轉動
- cmp : 讓 SW 指撥的數值，跟暫存器的值作比較。
- 但是要維持零度的值，所以基本底給 45 的值，給 179 的話就是直接輸出 2.5ms

架構介紹

- `decoder3x8` : 除8的部分是維持總週期20ms。下面的編碼是控制我們所需要的通道部分。這樣我們就可以控制8個不同馬達
- `result` : 單純的AND閘，這樣的設計是方便架構，讓 `cmp` 和 `decoder3x8` 所輸出的值做AND，成為最後的輸出

心得

我很辛苦的做完這次專題....希望不會被當



The end~~~