

Lab 9

學號：108070038、108071003

姓名：簡志宇、李彥璋

1. 工作分配

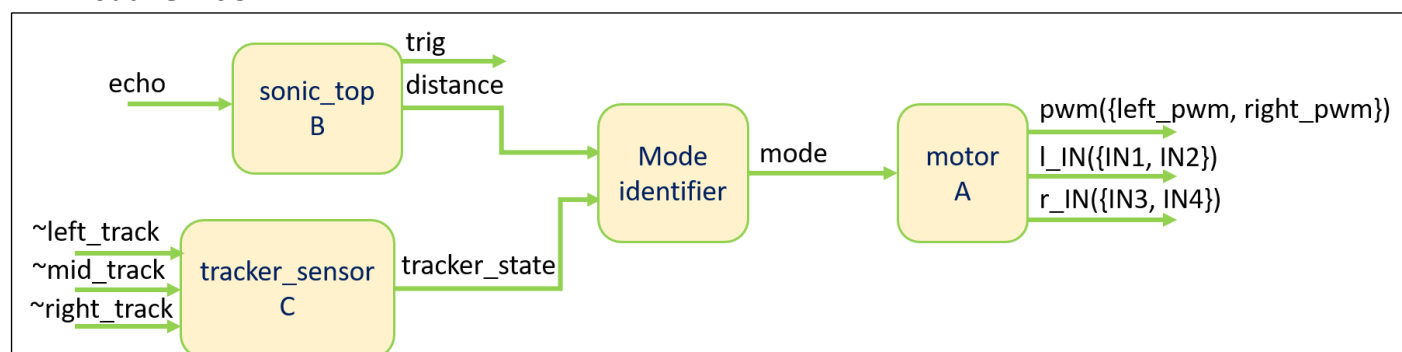
工作內容	負責人
code、測試、debug	簡志宇、李彥璋
demo 回答問題	簡志宇
block diagram	簡志宇
report	李彥璋

2. 實作過程

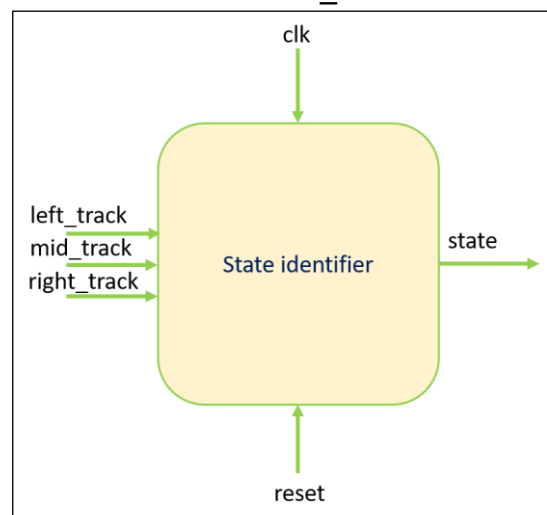
Lab9 要實作一台自駕車，車子沿著黑色的軌道運行、遇到障礙物會停下。

Block diagram :

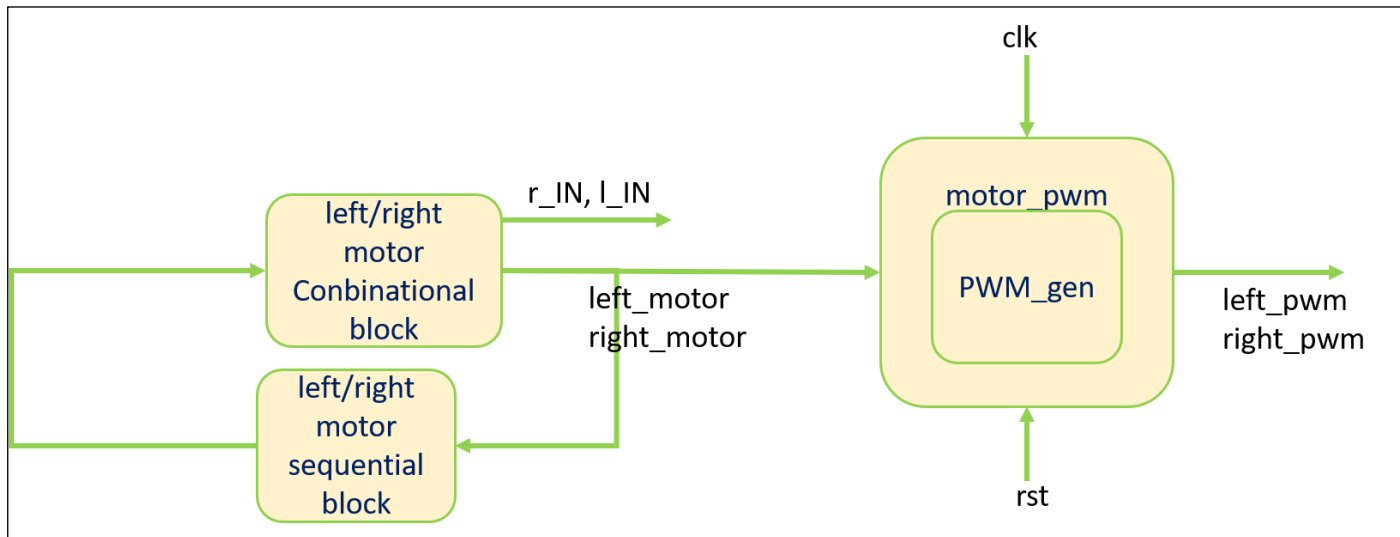
✧ Module Lab9



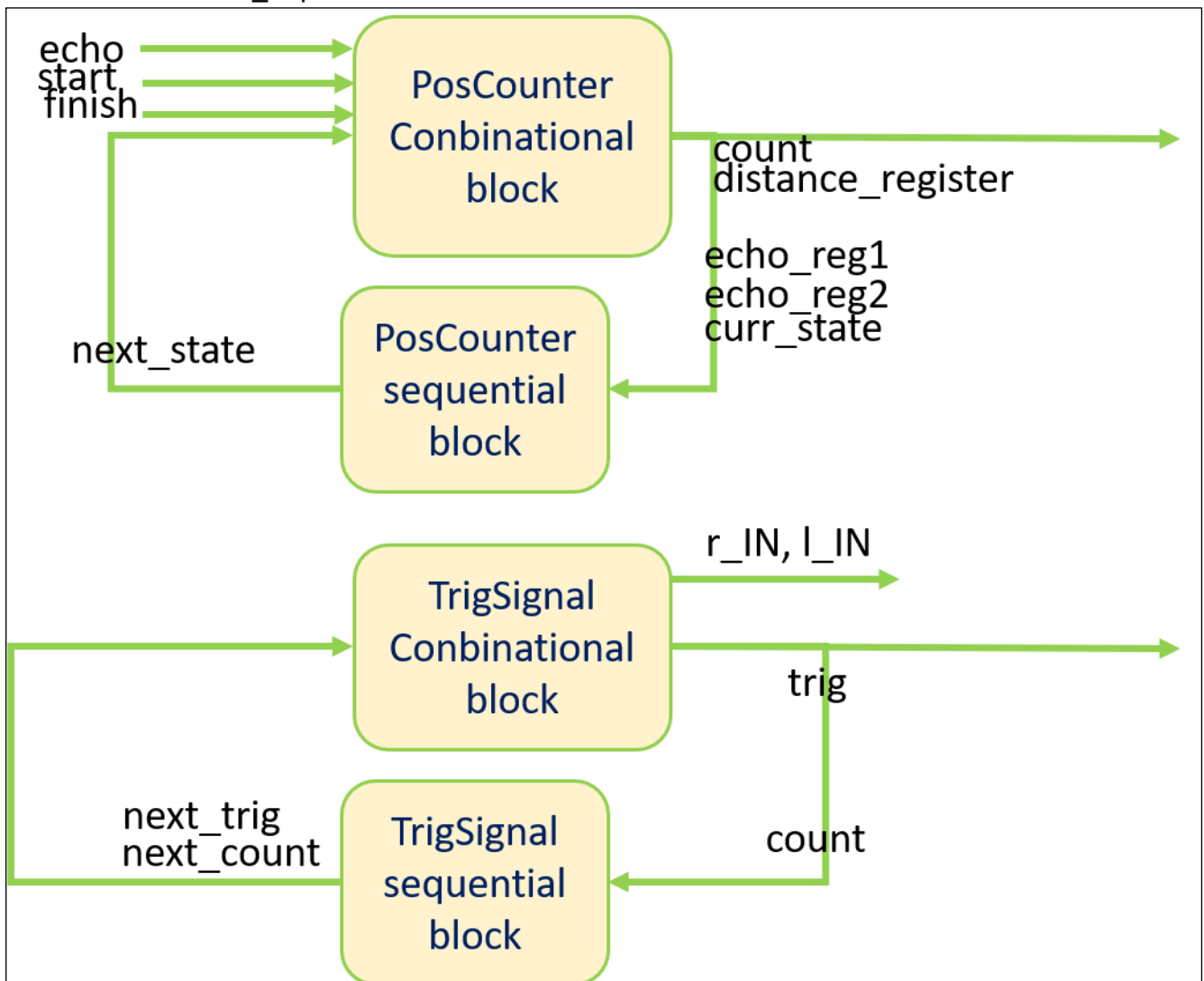
✧ Module tracker_sensor



✧ Module motor



✧ Module sonic_top



Code explanation :

Module tracker_sensor

用三個 independent infrared (IR) sensors (左、中、右) 判斷地板的反射是白色還是黑色，以此設定 state 並回傳給 Module Lab9。

```
326     always @(posedge clk or posedge reset) begin
327         if(reset == 1) begin
328             state <= 3;
329         end else if(mid_track == 1) begin
330             state <= 2;
331         end else if(left_track == 1) begin
332             state <= 0;
333         end else if(right_track == 1) begin
334             state <= 1;
335         end else begin
336             state <= 2;
337         end
338     end
```

Module PosCounter

這個 module 以超聲波計算車子與前方障礙物的距離，以下說明一些重點 code。

首先是超聲波的 start & finish。

```
252     assign start = echo_reg1 & ~echo_reg2;  
253     assign finish = ~echo_reg1 & echo_reg2;
```

echo_reg1 為發出超音波；echo_reg2 為接收超音波。

start 的時間點為發出但沒收到；finish 的時間點為沒發出但收到。

再來是 pulse_length 的計算。

(The length of the pulse is equal to the total travel time of the soundwave)

```
225         case(curr_state)  
226             S0:begin  
227                 if (start) curr_state <= next_state; //S1  
228                 else count <= 0;  
229             end  
230             S1:begin  
231                 if (finish) curr_state <= next_state; //S2  
232                 else count <= count + 1;  
233             end  
234             S2:begin  
235                 distance_register <= count;  
236                 count <= 0;  
237                 curr_state <= next_state; //S0  
238             end  
239         endcase
```

S0 為停止狀態，start 後 S0 -> S1。

S1 會持續遞增 count 直到 finish，S1 -> S2。

S2 將 count 紀錄予 distance_register，日後用以算 distance，最後回歸 S0。

最後計算距離。

```
255     // TODO: trace the code and calculate the distance, output it to <distance_count>  
256     assign distance_count = (distance_register / 2) * 34/1000;
```

公式參考 Lab9_spec。

distance_count 將回傳給 Module Lab9。

Module motor

這個 module 決定車子的速度、兩邊馬達是否啟動、前進或後退。

註：前進、後退的部分會與車子的接線有關，所以可能是相反的。

```
102     always @* begin
103         case(mode)
104             2'b00: begin
105                 next_left_motor = 100;
106                 next_right_motor = 700;
107                 r_IN = 2'b10;
108                 l_IN = 2'b11;
109             end
110             2'b01: begin
111                 next_left_motor = 700;
112                 next_right_motor = 100;
113                 r_IN = 2'b11;
114                 l_IN = 2'b01;
115             end
116             2'b10: begin
117                 next_left_motor = 700;
118                 next_right_motor = 700;
119                 r_IN = 2'b10;
120                 l_IN = 2'b01;
121             end
122             2'b11: begin
123                 next_left_motor = 0;
124                 next_right_motor = 0;
125                 r_IN = 2'b11;
126                 l_IN = 2'b11;
127             end
128         endcase
129     end
```

以 Module Lab9 傳來的 mode，決定車子要左轉、右轉、直走還是停止。

[104-109] 左轉。

[110-115] 右轉。

[116-121] 直走。

[122-127] 停止。

r_IN 及 l_IN 將回傳給 Module Lab9，決定車子馬達是否運作、前進 / 後退。

next_left/right_motor 則會以一個 sequential block 賦值給 left/right_motor，接著再傳給 Module motor_pwm 決定最後的 left/right_pwm(速度)。

Module Lab9

```
25     always @* begin
26         if(rst == 1) begin
27             mode = 2'b10;
28         end else if(distance <= 30) begin
29             mode = 2'b11;
30         end else if(tracker_state == 2'b00) begin
31             mode = tracker_state;
32         end else if(tracker_state == 2'b01) begin
33             mode = tracker_state;
34         end else if(tracker_state == 2'b10) begin
35             mode = tracker_state;
36         end else begin
37             mode = 2'b10;
38         end
39     end
```

以 Module tracker_sensor 及 Module sonic_top 的回傳，

決定要傳給 Module motor 的 mode，

00 左轉；01 右轉；10 直走；11 停止。

值得注意的是遇到障礙物的 if 判斷條件應位於左直右的上方，

因為遇到障礙物就要停下來了(但可能同時會要左直右)。

3. 學到的東西與遇到的困難

✧ 組裝車子

我們在組裝車子的時候裝了很久，為求謹慎很仔細的看 spec 裡的圖片。

✧ 電池

我的電池取出充一下下電就開始用了，非常不穩定。過程中間一度充到綠色了還是一下就沒電，但後來不知道為什麼電池就自己恢復正常了。

✧ mode

一開始我們給 mode 值的時候，障礙物的 if 條件一直放在左轉、右轉、直走下面，導致遇到障礙物車子就直接撞上去...，debug 很久。

✧ 小細節

receiver. If the IR is being reflected by the floor, the sensor will output HIGH. Hence, a white floor reflects IR, and the sensor outputs HIGH; a black floor absorbs IR, and the sensor outputs LOW.

黑色軌道應該是 0。

```
tracker_sensor C(  
    .clk(clk),  
    .reset(rst),  
    .left_track(~left_track),  
    .right_track(~right_track),  
    .mid_track(~mid_track),  
    .state(tracker_state),  
    .led(led)  
);
```

但模板加了 ~，所以黑色軌道要判斷是 1。

本來一直怪怪的，幸好組員火眼金睛 XD，才能發現這個小細節。

4. 想對老師或助教說的話

感謝老師&助教一整學期的辛勞，

剩下 final project 也會加油，

期末順利！寒假快樂！