

11010EECS207001Logic Design Lab

**LAB6 : Peripheral Components: VGA,
Mouse, and Dual FPGA**

TEAM9

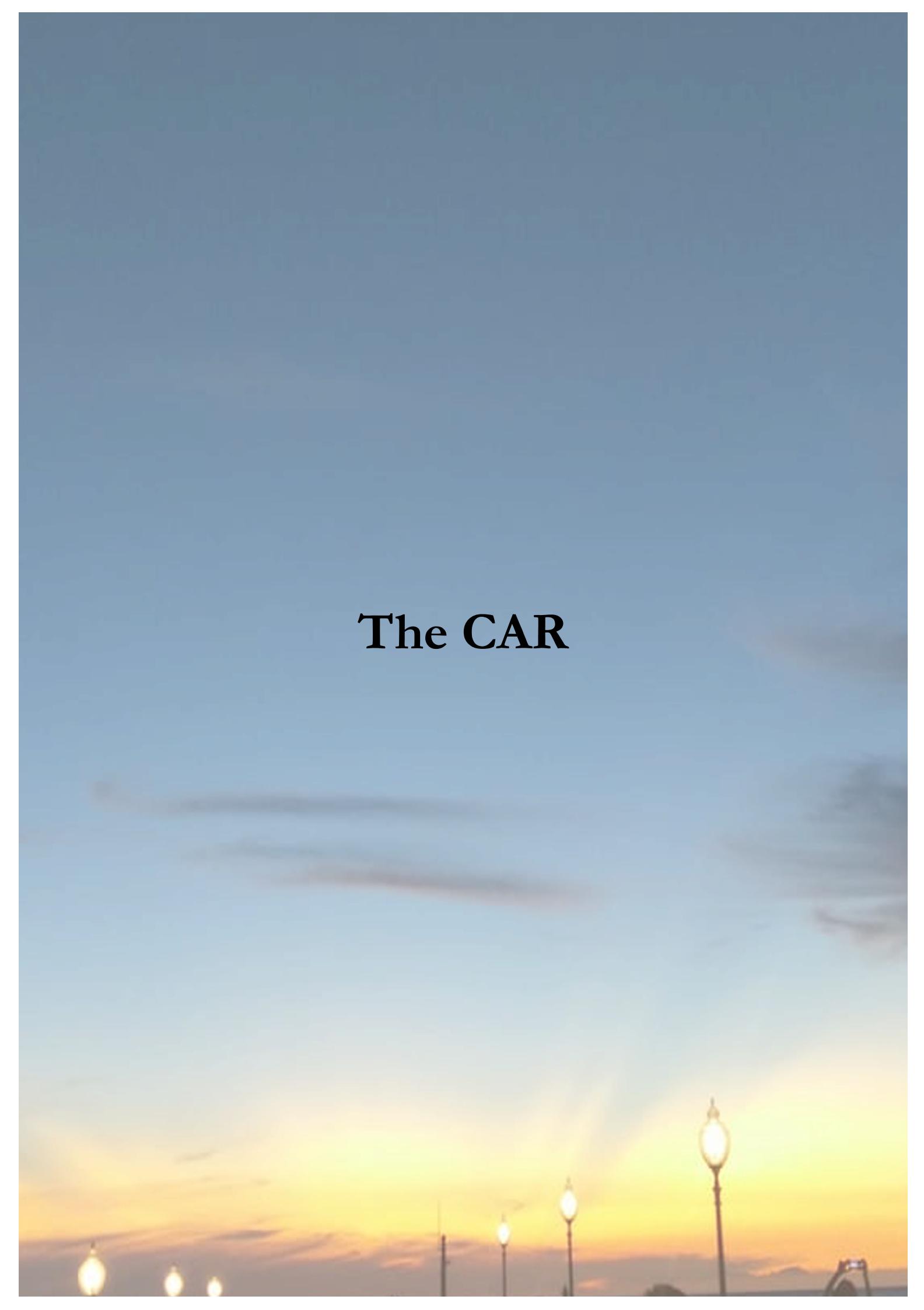
組長：108062213 顏浩昀

組員：106062304 黃鈺舒

Prof. Chun-Yi Lee

2021.12.23

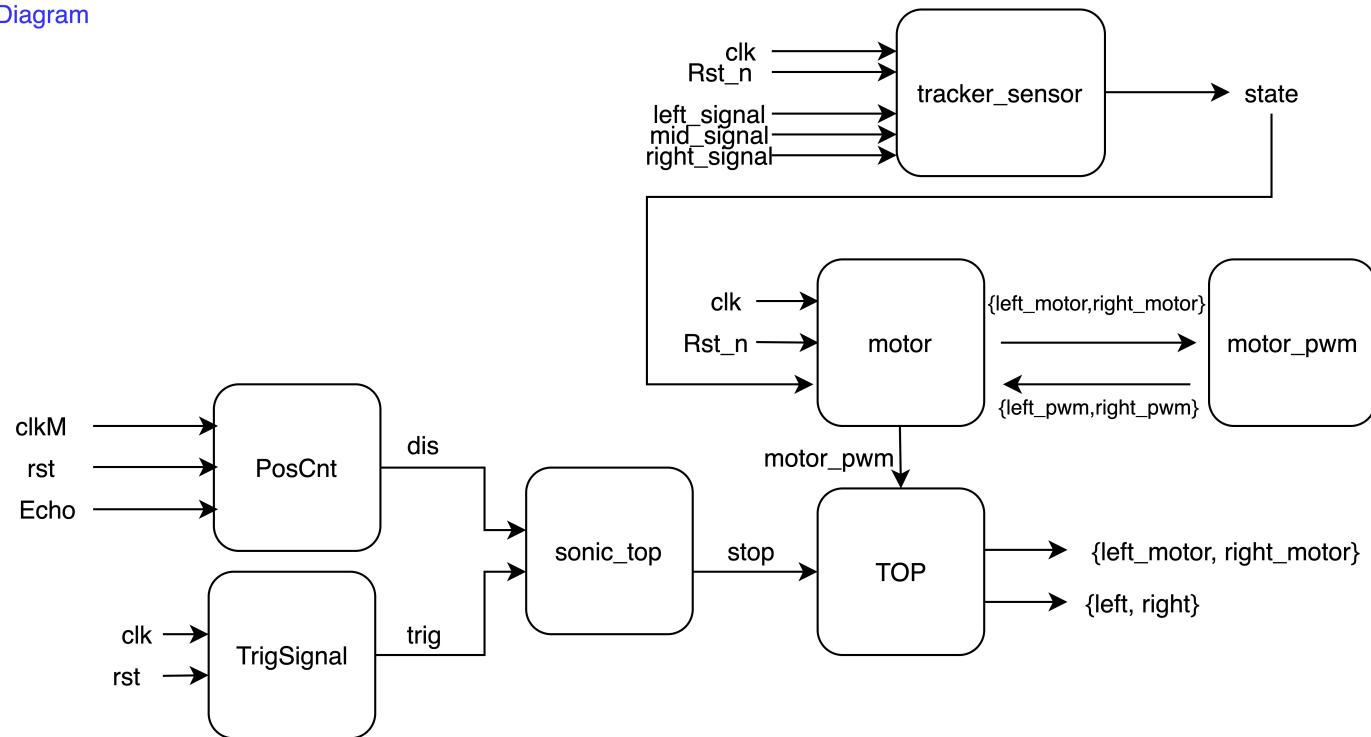
The CAR



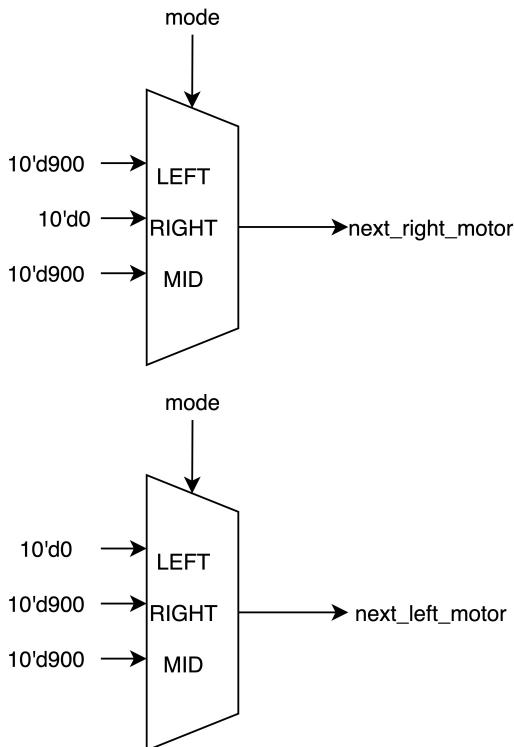
I. Diagram

(i) Block diagram

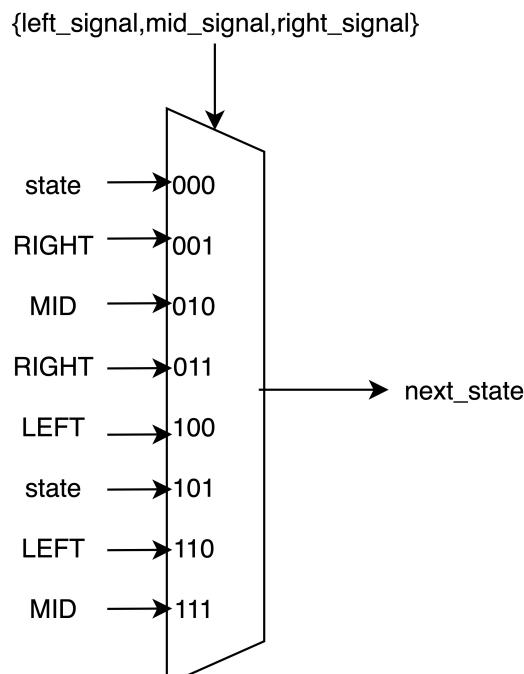
Block Diagram



Motor



Tracker_Sensor



II. Explanation

本次實作分成幾個module，首先，tracker_sensor會偵測賽道顏色，白色為1，黑色為0去改變state在下一步要左轉還是右轉，EX:如果為001代表下一步應該要右轉，如果為110應該要左轉。若偵測到為000（全黑）或101（左右白、中間黑），會維持上一個時刻的行為。

```
always@(*) begin //1 for white, 0 for black
    case({left_signal, mid_signal, right_signal})
        3'b000: next_state = state;
        3'b001: next_state = RIGHT;
        3'b010: next_state = MID;
        3'b011: next_state = RIGHT;
        3'b100: next_state = LEFT;
        3'b101: next_state = state;
        3'b110: next_state = LEFT;
        3'b111: next_state = MID;
        default: next_state = state;
    endcase
end
```

接著，用拿到的state去控制馬達轉速，motor會再根據左右motor傳入motor_pwm跟PWM_gen去實現左右兩邊不同轉速。PWM_gen用clk_divider先將週期除頻，在一個周期內去給PWM值1或0，得到馬達運轉的波型，如果1越長代表移動的越多，轉速會越快。（從下圖可以看出，right/left_motor即pwm_gen中的duty，數值越大代表轉動越快）

```
case(mode)
    LEFT: begin
        next_right_motor = 10'd900;
        next_left_motor = 10'd0;
    end
    RIGHT:begin
        next_right_motor = 10'd0;
        next_left_motor = 10'd900;
    end
    MID: begin
        next_right_motor = 10'd900;
        next_left_motor = 10'd900;
    end
    default: begin
        next_right_motor = 10'd0;
        next_left_motor = 10'd0;
    end
endcase
```

```
motor_pwm m0(clk, rst, left_motor, left_pwm);
motor_pwm m1(clk, rst, right_motor, right_pwm);

module motor_pwm (
    input clk,
    input reset,
    input [9:0]duty,
    output pmod_1 //PWM
);

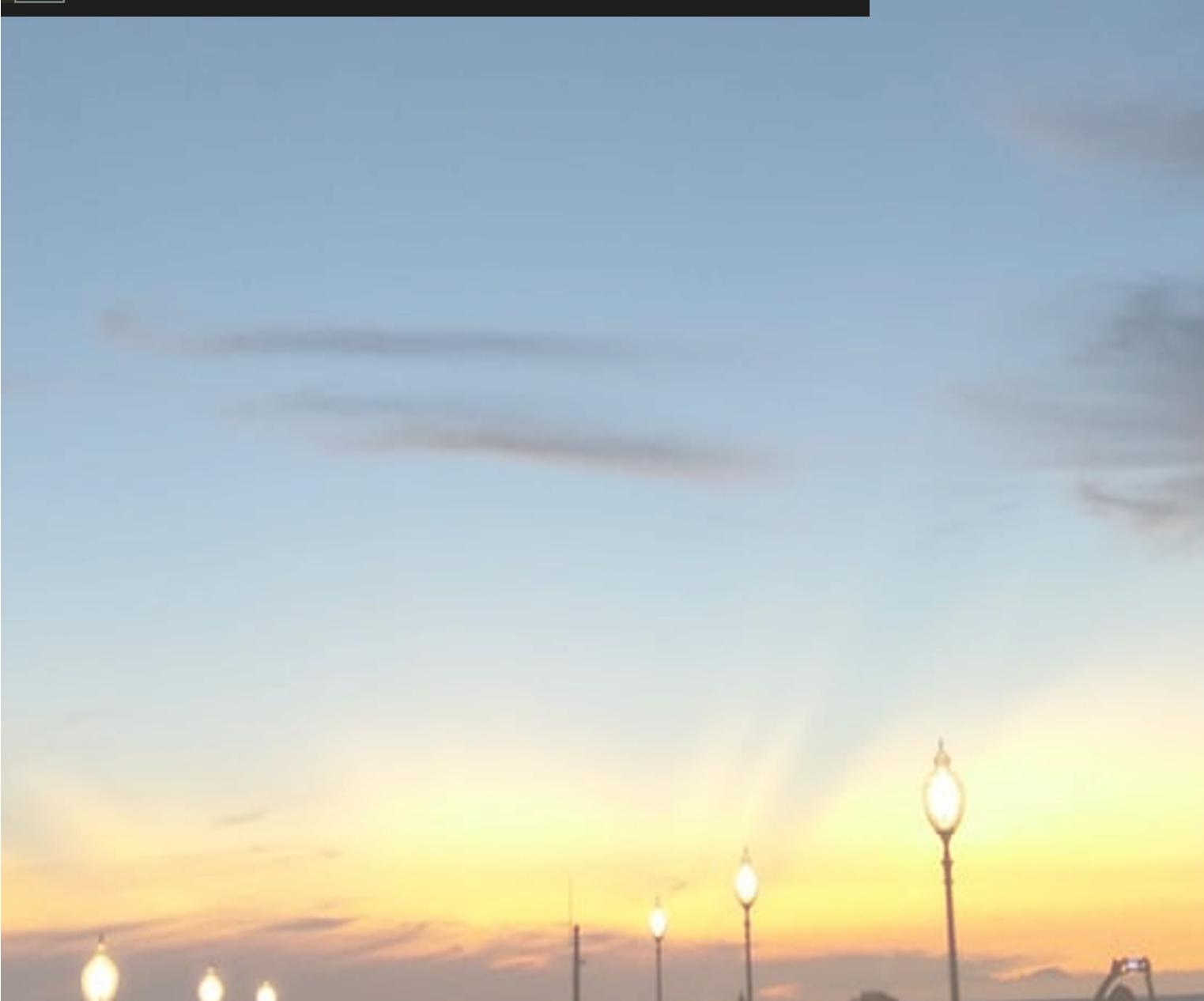
PWM_gen pwm_0 (
    .clk(clk),
    .reset(reset),
    .freq(32'd25000),
    .duty(duty),
    .PWM(pmod_1)
);
```

II. Explanation

另外sonic top是負責偵測障礙物的部分，TrigSignal會定期發出超音波並用PosCounter計算距離，如果距離小於40cm就將stop設為1傳給Top module，這時left與right會被設為都會被設為0，即馬達訊號為0，車子就會停下來。

```
// [T0-D0] calculate the right distance to trig stop(triggered when the distance is lower than 40 cm)
// Hint: using "dis"
assign stop = (dis < 4100) ? 1 : 0; //dis mm unit
```

```
always @(*) begin
    // [T0-D0] Use left and right to set your pwm
    if(stop)
        {left, right} = 4'b0000;
    else
        {left, right} = 4'b1001;
end
```



What are we learned from this Lab

車子的實作我們遇到蠻多問題在於線路上的連接，尤其是在馬達的部分。雖然知道in1,in2可以控制馬達順時鐘、逆時中旋轉，但根本分不清楚該從哪格方向去判斷順逆，結果一直錯邊，導致車子常常在後退。這其實應該是個很基本的東西，或許較一個國中、高中學生來考試都會，但當真的要用到這項能力，要將馬達接上去時，我們是不是還能記得這些學來的知識，算是從Lab中學到了超越知識的東西。

這次code implement時間不算太久，大部分都是在Trace code，從不同module之中的連接，一一找出其中的關聯性，增進了不少Trace的能力。然後在硬體的實作中，我們需要很仔細地確定每一條線對應的port，像一開始就把track sensor的左右搞混，將電源與接地插到對應right, mid的port。

最後，這台車子真的好耗電，我們一顆新的電池，大概在跑過2~3圈basic的賽道後就已經不能使用，我們認為應該電池還有電，但電壓不夠支撐那麼多sensor與motor運作，但實在沒有什麼好方法解決。在demo的時候看到多組使用電腦或行動充電器作為電源，也許是未來可以考慮的方式。（也可以當作給之後課程的學生小建議，不然真的用掉好多電池）



Cooperation

108062213 顏浩昀：

Car實作、電路圖繪製、report製作

106062304 黃鈺舒：

report製作

