Lab 9

學號: 109000162 姓名: 黃志偉
學號: 109000168 姓名: 許媄香

1. 實作過程

In this lab, we are doing a self-driving car by tracking black line (using IR tracker). We use the template given by the TA and just simply fill up the "to do" code.

- **Sonic Module**
  In this file, we need to calculate the length of signal that bounce to the receiver (**echo**) in order to calculate the distance between the object in front of it.

  We calculate it by the formula written in the lab spec.

  Calculate the distance, that is, (pulse_length / 2) * 340(m/s) .

  Because ½ is the same as shift right by one bit. Therefore, we just simply shift it by one bit. And then we need to multiply it by 0.034cm/s since we are using cm to measure the distance. To multiply it, we shift **distance_register** (pulse length) by 5 bits. the reason is because $1/2^5 = 0.031$ which is close to 0.034. And it doesn't matter that if the calculation is not really accurate since we just need to stop the car in certain distance. And the code as like below (shifting right by 1 + 5 = shifting right by 6 bit).

```
assign distance_count = distance_register >> 6;
```

- **lab9_top Module**
  After we got the **distance** from ultrasonic, we set the car will stop if the **distance** is smaller than 20cm. To do it, we set the **mode** state to '0'. The other state will be explained more later. But if the **distance** of the object in front of the car is bigger than 20cm, then **mode** output will depend on the **tracker_state** (state that we determined according to infrared sensors detection).

```
always@(posedge clk, posedge rst) begin
    if(rst) begin
        mode <= 2'b00;
    end else begin
        if(distance > 20) begin
            mode <= tracker_state;
        end else begin
            mode <= 2'b00;
        end
    end
end
```

- **Motor Module**
  aside from in the case where RESET signal is TRUE, **left_motor** and **right_motor** signals which control how fast the motor should work, in other word how fast the car will be going is updated depends on what mode the car is in (turn left, right, or go straight).

```
always@(posedge clk, posedge rst) begin
    if(rst) begin
        left_motor <= 10'd0;
        right_motor <= 10'd0;
    end else begin
        left_motor <= next_left_motor;
        right_motor <= next_right_motor;
    end
end
```

Using the table of reference given in the specs file, which is:

IN1 (IN3)	IN2 (IN4)	Spinning Direction
LOW	LOW	Motor off
HIGH	LOW	Forward
LOW	HIGH	Backward
HIGH	HIGH	Motor off

Left wheel and right wheel signal is what represent the IN1, IN2, IN3, IN4 signal on the reference.

Mode	Binary Value		Decimal Value	
	Left Wheel	Right Wheel	Left Motor	Right Motor
Stop	00	00	0	0
Left	01	01	750	750
Right	10	10	750	750
Straight	01	10	750	750

```
case(mode)
    2'b00 : begin    // stop
        next_left_motor = 10'd0;
        next_right_motor = 10'd0;
        l_temp = 2'b00; // off
        r_temp = 2'b00; // off
    end
    2'b01 : begin    // turn left
        next_left_motor = 10'd750;
        next_right_motor = 10'd750;
        l_temp = 2'b01; // backward
        r_temp = 2'b01; // forward
    end
    2'b10 : begin    // turn right
```

```
    2'b10 : begin    // turn right
        next_left_motor = 10'd750;
        next_right_motor = 10'd750;
        l_temp = 2'b10; // forward
        r_temp = 2'b10; // backward
    end
    2'b11 : begin    // go forward
        next_left_motor = 10'd750;
        next_right_motor = 10'd750;
        l_temp = 2'b01; // forward
        r_temp = 2'b10; // forward
    end
endcase
```

There is a little contrary with the table and the reference table, maybe due to the mismatch of the motor cable during installation.

- **Tracker Sensor Module**

to control the car direction when the “black tape” is detected in its way, there are three binary signals to show what signal the tracker sensor get, is it a black surface of white surface.

Binary signal : 0 → black surface detected
1 → white surface detected

Thus, depends on the received signal, different command is given/set to the car motor for operation.

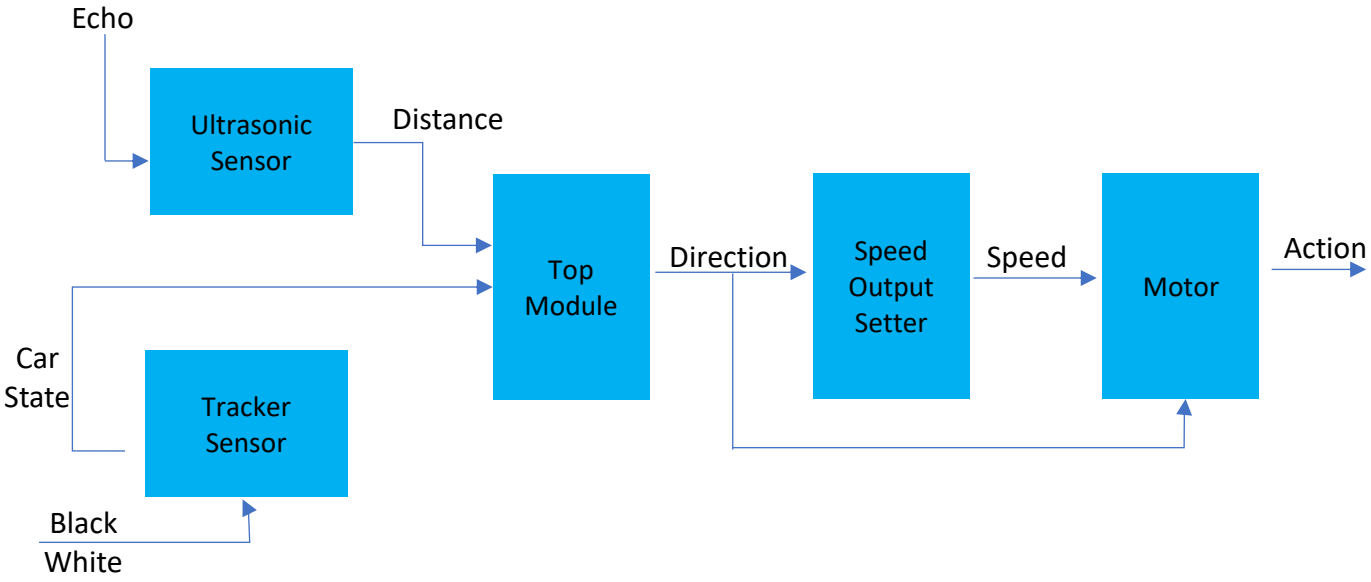
{ left_track, mid_track, right_track}	State
101	Forward
110	Left
100	Left
011	Right
001	Right

000	Right
111	Forward
Others	Stop

Code reference based on the table above :

```
always@(posedge clk, posedge reset) begin // 1 = white, 0 = black
    if(reset) begin
        state <= 2'b00; // stop
    end else begin
        if(left_track == 1 && mid_track == 0 && right_track == 1) begin
            state <= 2'b11; // forward
        end else if(left_track == 1 && mid_track == 1 && right_track == 0) begin
            state <= 2'b01; // left
        end else if(left_track == 1 && mid_track == 0 && right_track == 0) begin
            state <= 2'b01; // left
        end else if(left_track == 0 && mid_track == 1 && right_track == 1) begin
            state <= 2'b10; // right
        end else if(left_track == 0 && mid_track == 0 && right_track == 1) begin
            state <= 2'b10; // right
        end else if(left_track == 0 && mid_track == 0 && right_track == 0) begin
            state <= 2'b10; // right
        end else if(left_track == 1 && mid_track == 1 && right_track == 1) begin
            state <= 2'b11; // forward
        end else begin
            state <= 2'b00;  // stop
        end
    end
end
```

Control Flow :



- **Work distribution**
  sonic : 許媄香
  motor : 黃志偉
  tracker_sensor : 黃志偉
  lab9_top : 許媄香
  Report : 黃志偉 & 許媄香

2. 學到的東西與遇到的困難

When we first ran our car, the car kept stopping in the middle of road. And we were confused why it was happening. After a few moments, we realized that the black tape is thinner than we thought, and the car stopped because the IR sensor couldn't reach (detect) the black tape. And at that time, we set the car to stop when there's no black line detected. Therefore, we modified our movement policies a bit, and the car ran well.

3. 想對老師或助教說的話