

Lab 9

學號: 109062202, 109062203

姓名: 陳禹辰, 林祐禾

1. 實作過程

這次的 lab 主要就是先 trace code 過之後從 template 下去改，要做的分為三個部分，分別是 motor(馬達)、sonic(超聲波感測器)、tracker_sensor(紅外線感測器)。我們設計的車輛共有 stop, go, right, left 4 個模式，分別進行停止、直線前進、右轉、左轉四個動作。

馬達:

在 motor 模組有一個輸入訊號 mode 代表目前車輛要做的動作，我們根據 mode 訊號來對馬達進行控制，使車輛進行對應的動作。

馬達的控制包含了控制轉向以及控制轉速兩個部分。馬達的旋轉方向由兩個訊號進行控制，如左輪的馬達由 IN1 與 IN2，也就是 motor.v 中的 l_IN 控制，當 l_IN 等於 01 時其旋轉方向會使車輛向前，等於 10 時使車輛向後，等於 00 或 11 時則不動，而右馬達因安裝的方向與左馬達相反，在 r_IN 等於 01 跟 10 時的效果與左輪的相反。我們也透過改變輪子的轉向使車輛改變方向，當右輪向前轉，左輪向後轉時車輛方向就會轉向左邊，右轉時則相反。因此當車子直線前進時{l_IN, r_IN}輸出 0110，右轉時輸出 0101，左轉時輸出 1010，停止時輸出 0000。

```
always @(posedge clk or posedge rst) begin
    if(rst)begin
        r_IN <= 2'b00;
        l_IN <= 2'b00;
    end else begin
        case(mode)
            stop: {l_IN, r_IN} <= 4'b0000;
            go: {l_IN, r_IN} <= 4'b0110;
            right: {l_IN, r_IN} <= 4'b0101;
            left: {l_IN, r_IN} <= 4'b1010;
            default:{l_IN, r_IN} <= 4'b0000;
        endcase
    end
end
```

控制馬達轉速則是透過改變 duty cycle 來達成，在實作時我們的作法為控制 motor.v 裡面的 left_motor 跟 right_motor 訊號，我們在車輛前進時使用的數值為 750，大約為 75% duty cycle，而在轉彎時我們讓向後轉的輪子轉的稍為較慢，使用的數值為 600，大約為 60% duty cycle。

```
always @(posedge clk or posedge rst) begin
    if(rst)begin
        left_motor <= 0;
        right_motor <= 0;
    end else begin
        case(mode)
            stop: begin
                left_motor <= 0;
                right_motor <= 0;
            end
            go: begin
                left_motor <= 750;
                right_motor <= 750;
            end
            right: begin
                left_motor <= 750;
                right_motor <= 600;
            end
            left: begin
                left_motor <= 600;
                right_motor <= 750;
            end
            default: begin
                left_motor <= 0;
                right_motor <= 0;
            end
        endcase
    end
end
```

超聲波感測器:

這部分所要做的是讓 sonic.v 中的 PosCounter 模組能夠計算並且輸出與前方障礙物的距離。我們按照 spec 中的算式 $(\text{pulse_length} / 2) * 340(\text{m/s})$ 進行實作，由於 PosCounter 使用的是 1Mhz 的 clock，每經過一個 clock cycle 就代表經過 1us，而 distance_counter 紀錄的為經過多少 clock cycle，因此 distance_counter 的值就是我們要的 pulse_length，單位為 us。且由於我們所要輸出的單位為 cm，在進行單位換算後就能夠得到最終的算式。

```
assign distance_count = distance_register * 17 / 1000;
```

紅外線感測器:

紅外線感測器的部分就是根據偵測到的來決定要往左往右或是往前，我們的判斷方式為如果左邊的感測器感測到黑色就讓車輛往左，右邊感測到黑色就讓車輛往右，否則就是往前直行。

```
always @(*) begin
    if(left_track == 1) state = left;
    else if (right_track == 1) state = right;
    else state = go;
end
```

最後再於 top module 中根據超聲波感測器及紅外線感測器輸出的值來決定車輛的行為並控制馬達，首先若超音波感測器輸出的距離 <15 就讓車輛停下來，否則車輛就依照紅外線感測器輸出的 tracker_state 的方向移動。

```
always @(*) begin
    if(distance < 15) mode = stop;
    else begin
        if(tracker_state == right) mode = right;
        else if (tracker_state == left) mode = left;
        else mode = go;
    end
end
```

分工:

陳禹辰：馬達、report

林祐禾：超聲波感應器、紅外線感應器、report

2. 學到的東西與遇到的困難

學會控制馬達的旋轉速度及方向，還有學會使用超音波感測器與紅外線感測器所偵測的訊號。也學到如何透過外接電源讓 fpga 在沒有連接電腦時使用。

遇到的第一個困難是接線的部分，在接線時需要一個一個訊號慢慢找對應的接口，而且還要注意不能接錯，非常的考驗耐心。

遇到的另一個困難是剛開始沒有注意到 spec 上建議的 duty cycle 範圍，所以剛開始因為 duty cycle 設的太低導致馬達不會動卡了很久。

3. 想對老師或助教說的話

When asked to draw a flowchart of my code

