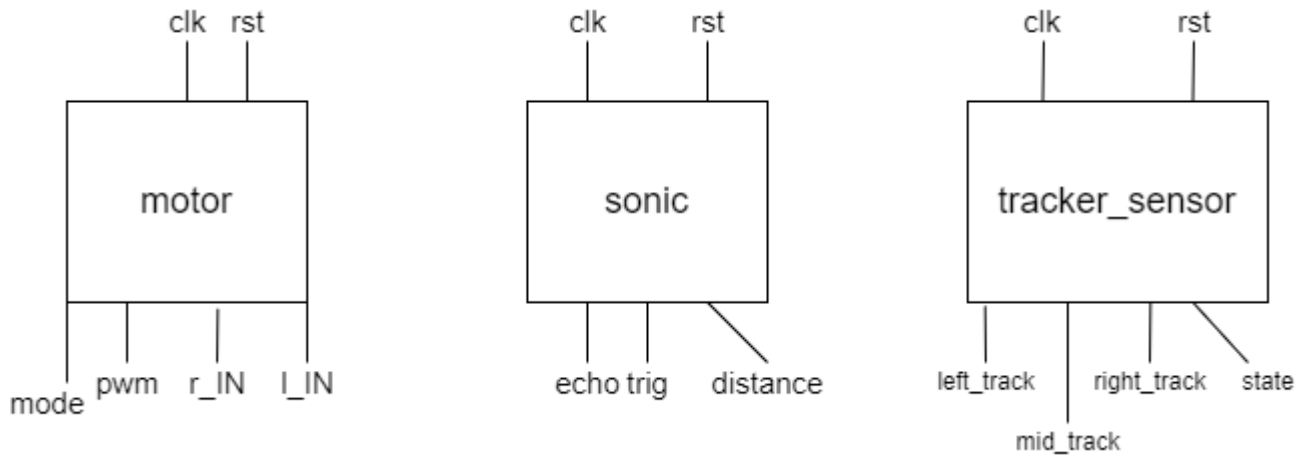


## Lab 9

學號: 106022122、109062318

姓名: 房昀凱、簡弘哲

### 1. 實作過程



FSM:

```
parameter go = 2'd0;
parameter turn_right = 2'd1;
parameter turn_left = 2'd2;
parameter hold = 2'd3;
always @(posedge clk, posedge reset) begin
    if (reset) begin
        state <= hold;
    end else begin
        if (mid_track)
            state <= go;
        else if (right_track)
            state <= turn_right;
        else if (left_track)
            state <= turn_left;
    end
end
```

在 tracker\_sensor.v 裡面一共分為 4 個 state，分別是 go:直走、turn\_right:右轉、turn\_left:左轉、hold:停車，然後根據從 top module 傳入的 left、mid、right\_track 去決定 state，又因為 left、mid、right\_track 在從 top.v 傳進 tracker\_sensor 的時候有做反向，因此當上圖 mid\_track=1 的時候代表中間的紅外線感測器底下是黑色，就繼續往前直走，左轉右轉也是一樣的道理。

在 top module 中也利用一樣的方法去處理 mode:

```
parameter HOLD = 2'd0;
parameter GO = 2'd1;
parameter TURN_LEFT = 2'd2;
parameter TURN_RIGHT = 2'd3;

always @(posedge clk, posedge rst) begin
    if (rst) begin
        mode <= HOLD;
    end else begin
        if (distance <= 20) begin
            mode <= HOLD;
        end else begin
            case (tracker_state)
                2'd0: mode <= GO;
                2'd1: mode <= TURN_RIGHT;
                2'd2: mode <= TURN_LEFT;
                2'd3: mode <= HOLD;
            endcase
        end
    end
end
```

從 tracker\_sensor.v 的 output(tracker\_state)可以得知目前為哪一個 state，再利用這個資訊決定要傳入 motor.v 的 mode 值為何。這邊也處理了障礙物要停下來部分，判斷從 sonic\_top.v output 出來的 distance(單位為 cm)的值有沒有超過 20cm，如果車子與障礙物的距離小於 20cm 就停車。

## Motor.v:

```

always @(*) begin
    case (mode)
        2'd0: begin // HOLD
            next_left_motor = 0;
            next_right_motor = 0;
            r_IN = 0;
            l_IN = 0;
        end
        2'd1: begin // GO
            next_left_motor = 1024 * 7/10;
            next_right_motor = 1024 * 7/10;
            r_IN = 2'b10; // actually left
            l_IN = 2'b01; // actually right
        end
        2'd2: begin // TURN_LEFT
            next_left_motor = 1024 * 6/10;
            next_right_motor = 1024 * 7/10;
            r_IN = 2'b10;
            l_IN = 2'b01;
        end
        2'd3: begin // TURN_RIGHT
            next_left_motor = 1024 * 7/10;
            next_right_motor = 1024 * 6/10;
            r_IN = 2'b10;
            l_IN = 2'b01;
        end
    endcase
end

```

從 mode 可以知道目前車子的狀態，再去調整相對應的馬達轉速，l\_IN, r\_IN 分別控制一組馬達，r\_IN, l\_IN 都 assign 成 0 的時候馬達不會轉、assign 成 10、01 的時候就是轉。要達到轉彎的效果就要讓輪子一快一慢，我們組選擇用 7 成當作預設去讓馬達轉動，那另一邊的輪子就選用 6 成。

最後是計算障礙物與車子的距離，直接將上課講義中的公式帶入即可，只須避免浮點數的運算，將 0.034 寫成 340/10000 就可以避免掉浮點數的問題，順便將單位換算成 cm。

```
assign distance_count = (distance_register >> 1) * 340/10000
```

### 2. 學到的東西與遇到的困難

這次 lab 的實作過程非常順利，沒有遇到棘手的 bug。

### 3. 想對老師或助教說的話

這次的 lab 非常簡單，一下就做完了，全部的 lab 也都做完了。

### 4. 分工

房昀凱: top.v, tracker\_sensor.v, debug

簡弘哲: motor.v, sonic.v, debug