

Lab 6: Car

Group 3 : 109060013 張芯瑜 109062328 吳邦寧

Design Explanation

1. Travel Logic Design

以下為感應燈與行走方向之對照表。

Direction State	Left	Center	Right
(00) Backward	0	0	0
(10) Right	0	0	1
(10) Right	0	1	0
(10) Right	0	1	1
(01) Left	1	0	0
(11) Forward	1	0	1
(01) Left	1	1	0
(11) Forward	1	1	1

以下為方向狀態與 left, right 輸出的對照表

State	Left 1	Left 2	Right 1	Right 2
(00) Backward	0	Pwm[1]	0	Pwm[0]
(10) Right	Pwm[1]	0	0	Pwm[0]
(01) Left	0	Pwm[1]	Pwm[0]	0
(11) Forward	Pwm[1]	0	Pwm[0]	0

由上述表格定義規格，並實作對應的數位邏輯，自走車即可順利於環境中完成繞行任務。

2. Sonic

經由追蹤程式碼，再搭配物理知識，可以得到下列數學式

$$\begin{aligned} M &= N \left(\frac{1}{100} s \right) \cdot (100 \div 58) \left(\frac{s}{cm} \right) = N \cdot (100 \div 58) \left(\frac{1}{100} s \right) \left(\frac{cm}{s} \right) \\ &= N \cdot (100 \div 58) \left(\frac{cm}{100} \right) \end{aligned}$$

數學符號	電路代號	意義
M	distance_count	距離有多遠，單位是百分之一公分
N	distance_register	過了幾個百分之一秒

由上述討論得知， M 的單位是百分之一公分，若距離大於四十公分，那麼 M 就會大於 4000。使用下述程式碼，即可命令車輛停下。

```
assign stop = (dis < 20'd4000 ? 1'b1 : 1'b0);
```

由討論區得知，過長的距離會使得變數溢位，因此，我們必須限制計數器的最大數值。

```
next_count = (count > 20'd60_000 ? count : count + 1'b1);
```

我們有以下不等式

$$2^{20} \div 100 \cdot 58 > 60,8174$$

$$2^{20} > 60,0000 \cdot 100 \div 58$$

因此，我使用 60,0000 作為上界，這樣不僅可以避免溢位，還可以儲存夠大的距離。

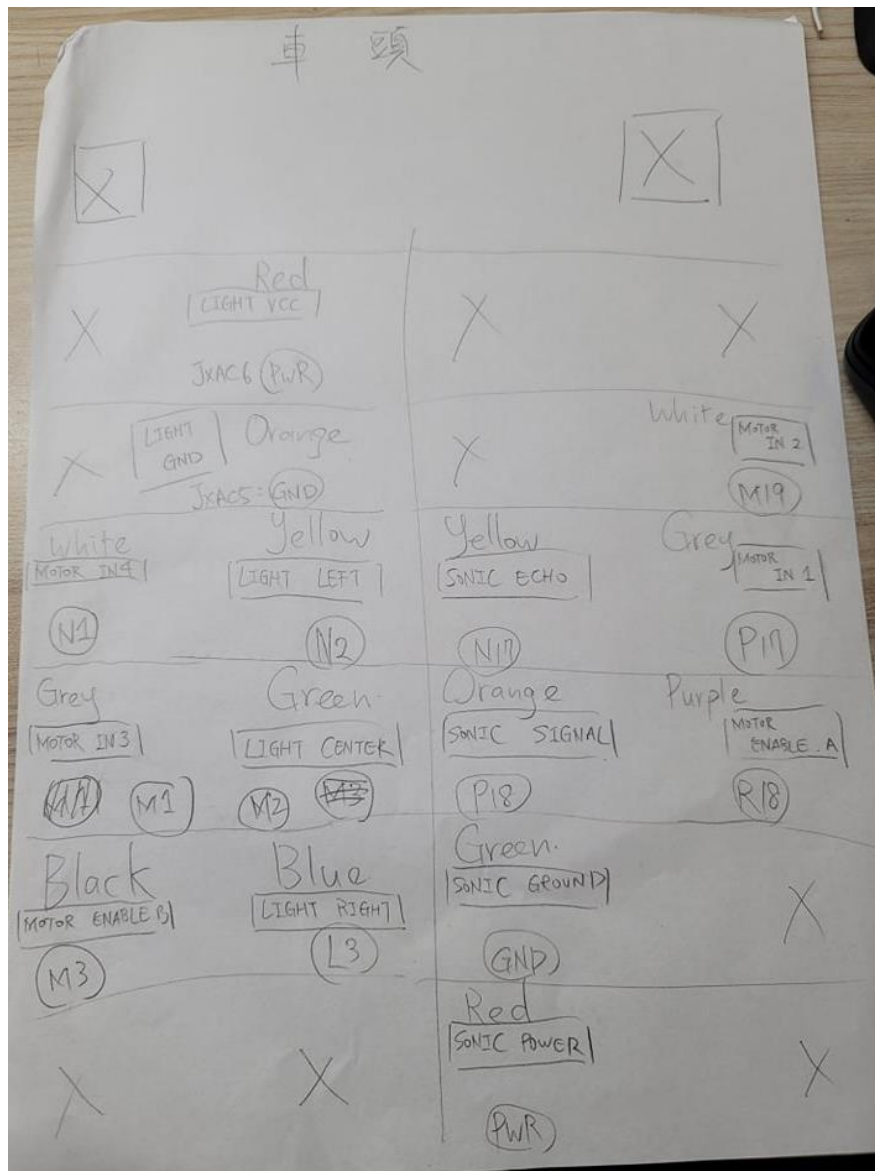
3. Motor

以下為方向狀態與馬達速度的對照表。

State	Left	Right
Backward	1023	1023
Right	800	1023
Left	1023	800
Forward	1023	1023

4. Hardware & Jumpers

下表為各電子元件與 FPGA 的接線腳位圖。



Contribution

1. Lawrence Wu

撰寫報告，端茶遞水，簡易零件組裝。

2. Ariel Chang

設計、組裝車子之軟硬體元件。

What have we learned?

1. 記得幫外部變數估計範圍，免得產生溢位問題
2. 電池接反的話，電池盒會熱熱的（短路）
3. 電池接反的話，車子不會動（短路）
4. 不要亂猜單位，可以好好的用物理知識去推論單位。
5. VMS 接上去才能讓輪子轉動
6. 要把 GND 都接上去，這樣每個電子元件的地板才會一樣高，訊號才不會亂七八糟
7. 遇到問題時，先去看討論區
8. 三用電表很適合用來 de 現實世界的 bug
9. 地板會反光，可以遮住光源避免訊號錯誤