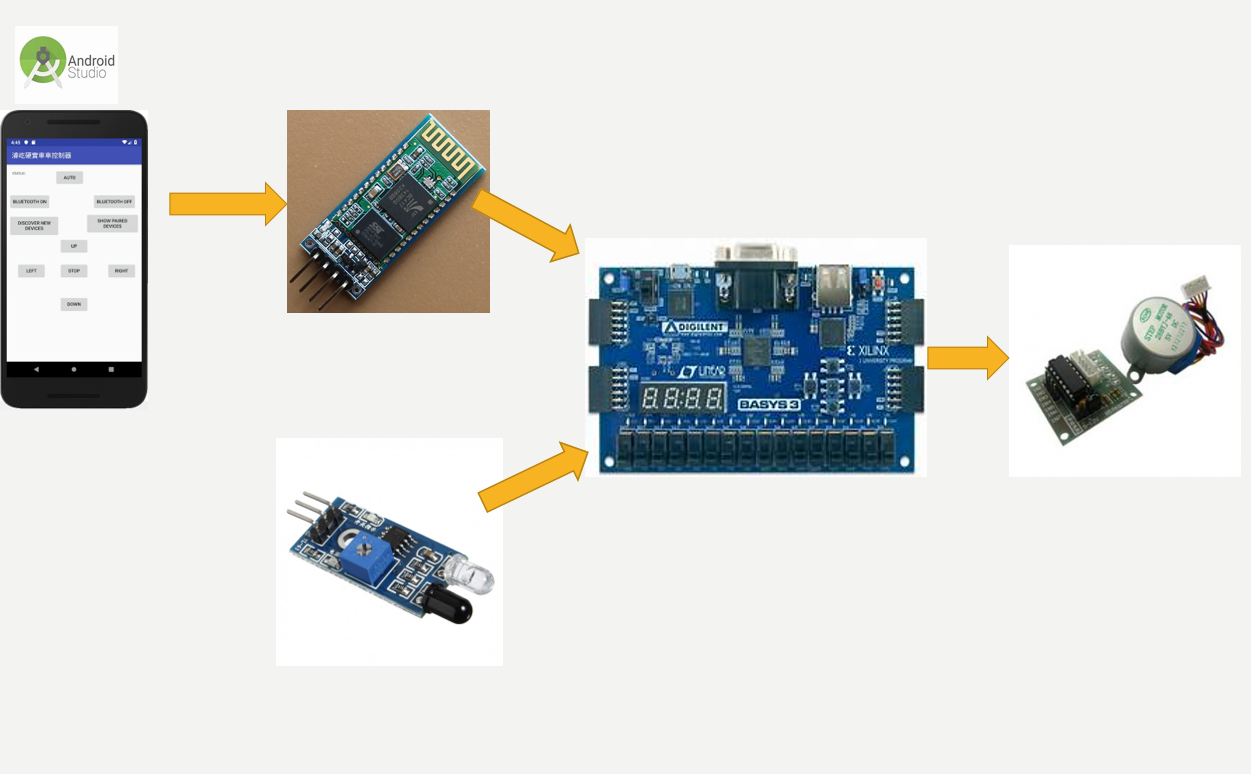
硬體實驗final project

第七組

105030009 黎佑廷

105030015 郭家瑋

1. **標題**：FPGA車
2. **目標**：使用FPGA與verilog做出一台可以由手機藍芽遙控，運用藍芽模組接收遙控訊號的車子，另外的模式是可以讓車子自動行徑而不靠遙控，車子運用紅外線避障模組來感測前方障礙物並且及時轉彎再前行。



其中的兩種模式詳述如下：

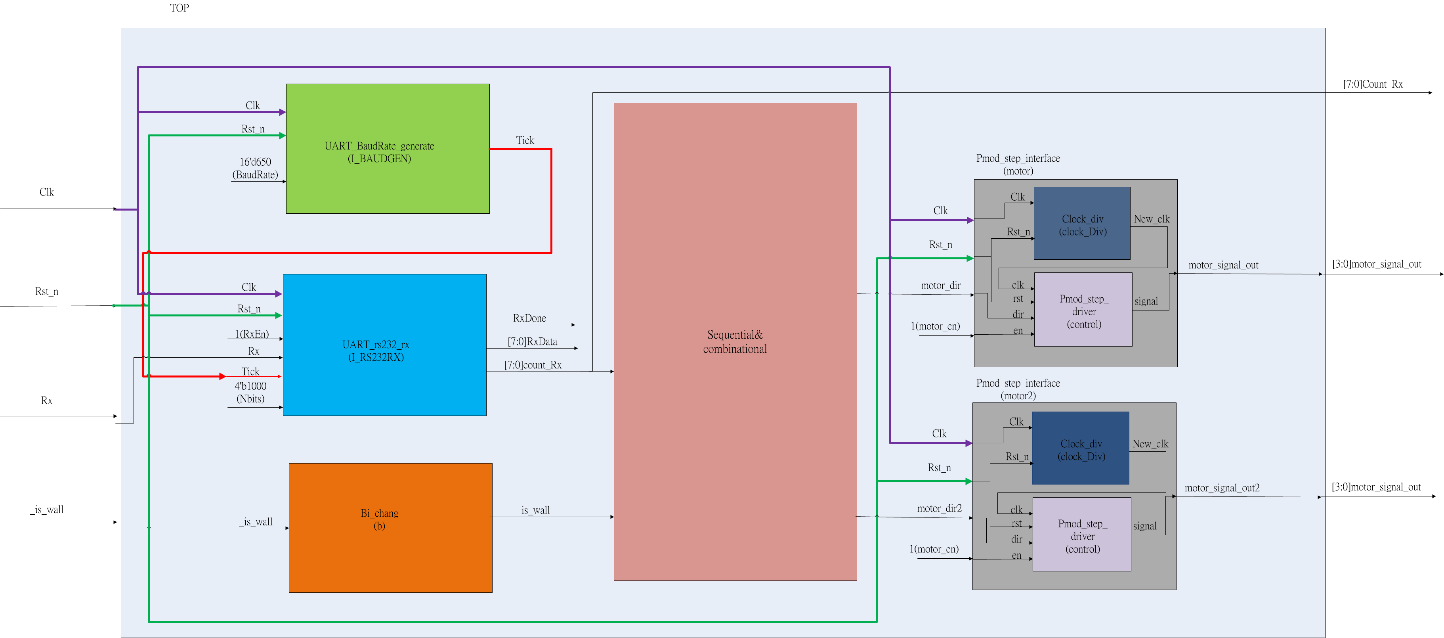
Manual: 於手機控制上、下、左、右

Auto: 車子自動前行，前方有障礙物時，後退、左轉、再前行

手機使用android studio寫app，介面如下。

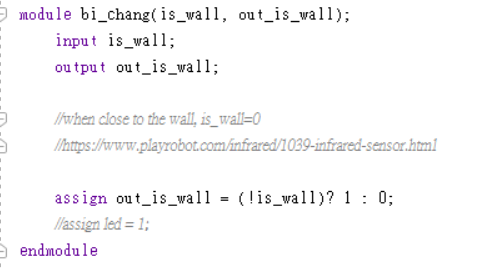


**三、Block Diagram**



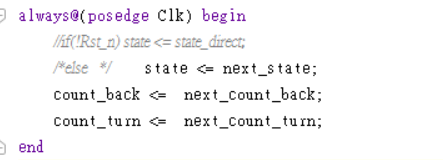
以下將說明各個module：

* **Bi\_chang**

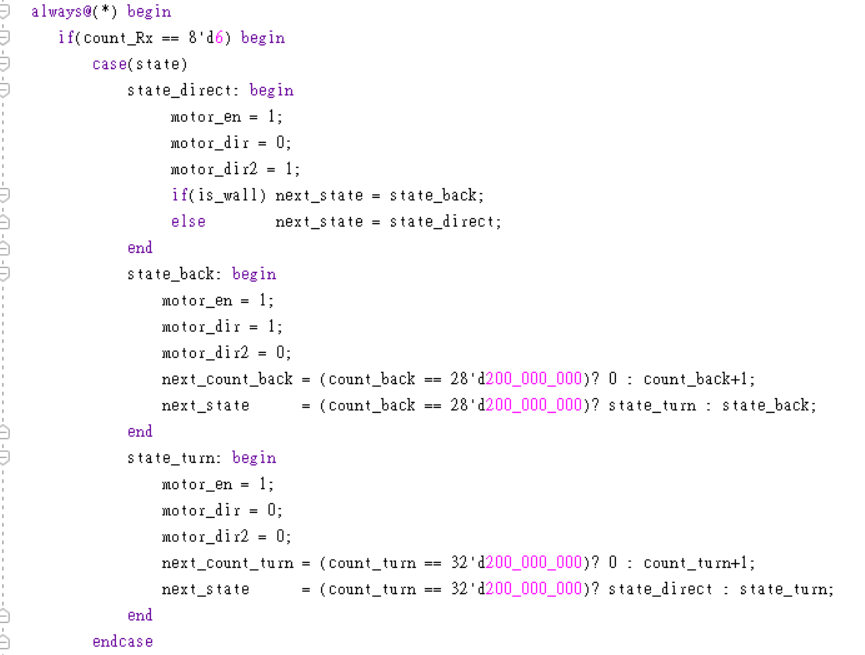


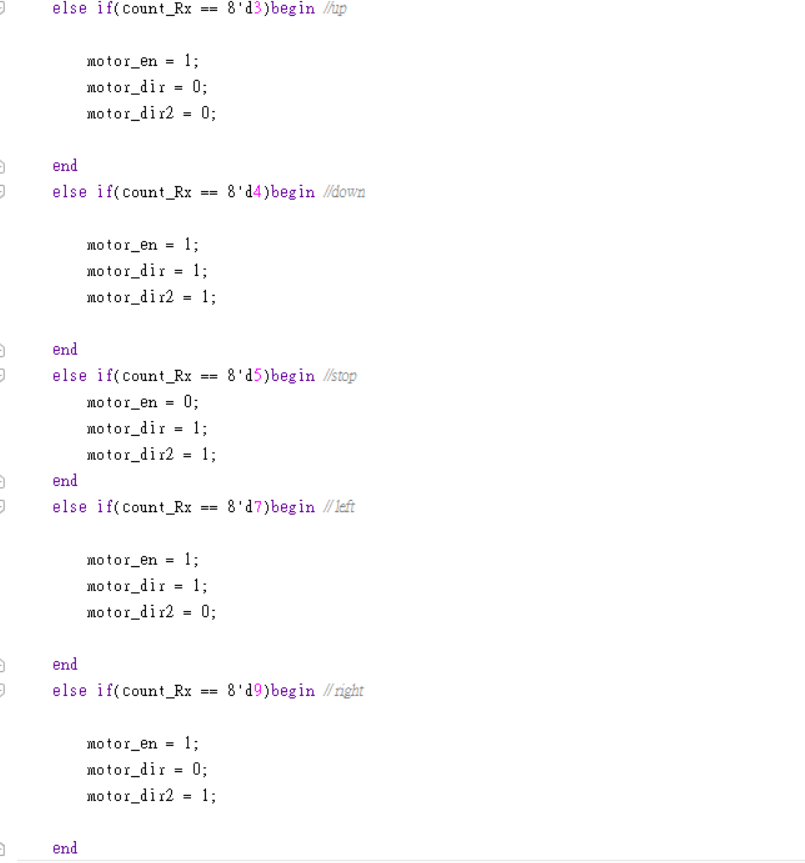
這個module主要處理紅外線避障模組的訊號。我們的紅外線避障模組有三個pin。兩個是電源pin，另外一個是邏輯pin，在前方三公分處有障礙物時，將0訊號傳回FPGA版，沒有障礙物時為1。因為跟邏輯的high、low有點差距，因此在這個module將訊號反向。

* **Sequential & Combinational**



在這個module主要是將來自紅外線、藍芽訊號收集並且給出最後的馬達訊號。



Count\_Rx的來源是我們用手機傳送的藍芽訊號，若傳送的訊號是6，車子進入自動模式，是一個包括三個state的FSM。第一state是state\_direct，會讓車子直行。若前方有障礙物則會進入第二個state，state\_back，會倒退約兩秒。最後進入state\_turn，會向左轉約兩秒(實際操作接近轉角90度)，再回到state\_direct。

若傳送的訊號是3、4、5、7、9則分別代表上、下、停、左、右，分別給出兩個馬達的dir。

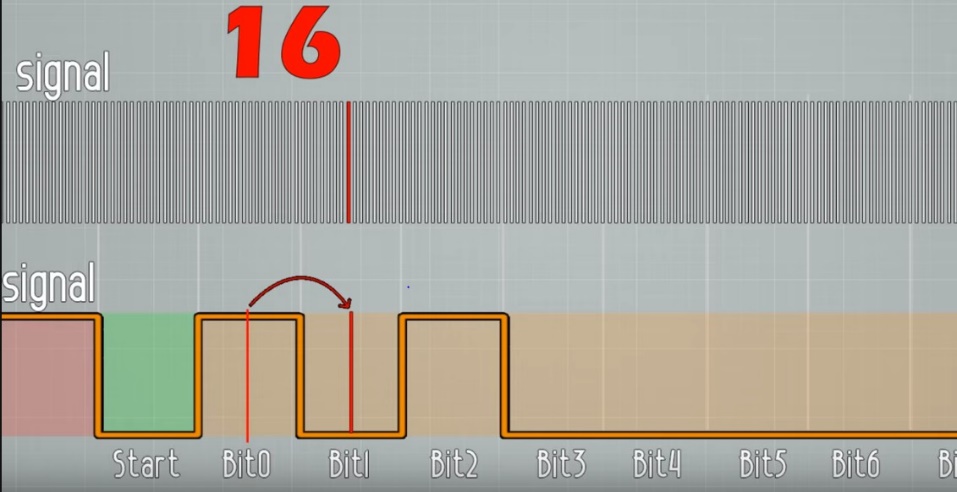
* **UART\_BaudRate\_Generate**



這個module是產出UART\_rs232\_rx module的Tick訊號。基本上就是一個可以被input BaudRate的clock divider。在UART\_rs232\_rx module的講解會說明BaudRate要帶入多少及Tick的功用。

* **UART\_rs232\_rx**

這個module較為複雜，但原理其實不難。原理示意圖如下：



簡單來說，手機藍牙與藍芽模組是透過UART序列平行轉換來傳送資料。我們的藍牙模組的baud rate為9600，我們的FPGA時脈是100MHz，如果我們想要TICK(圖中的上方訊號)的頻率是UART訊號(途中下方訊號)的16倍(每個UART bit寬度是16個TICK)，那我們會需要頻率是16\*9600HZ的時脈。UART的寬度是1/9600，大約是104us；FPGA的時脈寬度是10ns，因此我們會需要FPGA除頻104us/10ns/16=650，因此，在TOP module中，設定Baud rate=650。

為了取得穩定的訊號值，我們只在UART訊號中點讀值。藍牙模組在沒有接收訊號的idle state是high，一接收到訊號後首先是low的start訊號。在開始是low後，我們數8個TICK以讀到start bit的中央，之後每過16個TICK就再讀一個bit，直到讀完8個bit之後出現high的idle state。

以上方法與圖示來源自<https://www.youtube.com/watch?v=QlscDcbKUV4>

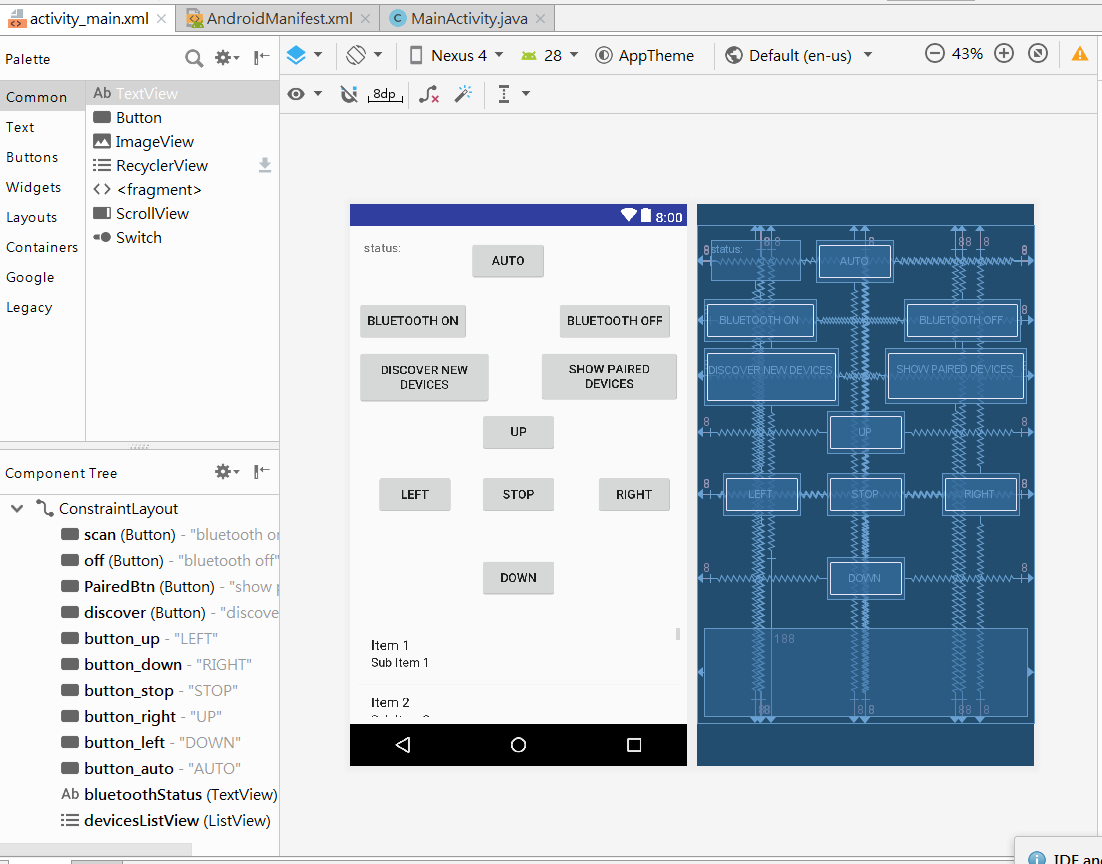
不過我們實際的操作結果經各種debug後，發現在idle state的時候UART事實上是high，並且我們經由手機傳送過去的char轉換成數字型態也是不對的，並且，每次傳相同的東西過去，也會讀到不同的值。因為這一點，我們只能懷疑是時脈出了問題，可能沒對齊，造成每次讀的有長有短，讀出來的值也就不相同。這可能來自於我們在算要給定的Baud rate時，所做的近似有關。不過，我們沒辦法取得示波器，所以也沒辦法實際檢視這一點。

我們想出的解決方法是，盡量不要用到TICK，雖然這樣就沒有用到這個module裡面的大部分code與原理。不過仍是一個方法。在每次有Rx(讀進的每個bit叫Rx)上升緣時，讓一個counter+1，因此，在傳送完一筆資料後，就會有一定量的counter。至於要怎麼再傳完一筆資料後把counter清空，則是判斷若clock經過一定的數量，就把counter清空，意思是，兩次發送資料的時間間隔不能太近，不然有可能出問題。這個時間間隔約是0.01秒，經測試，在手機上是非常難在0.01秒鐘連按兩次發送資料的，因此，算是解決了這個問題。

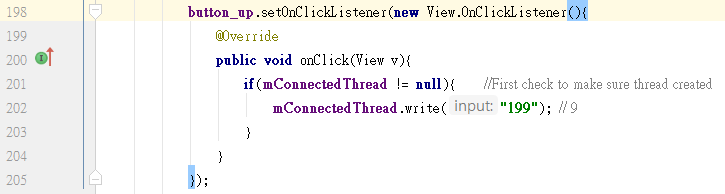
* **APP (Android Studio)**

在寫APP來控制我們的車子方面，我們使用了Android studio這個開發環境來編寫我們的APP，我在這邊簡單介紹一下我們藍芽訊號是怎麼從手機到達FPGA版上的。

在Android studio中我們定義了一些button，如下:



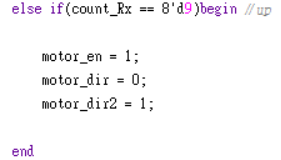
我以讓車子前進的 ”UP” 這個按鈕來舉例說明，當UP這個按鈕被按下時就會觸發下圖第200行的onClick這個function，此外，在傳送資料前我們得先檢查藍芽是否已經接上裝置，不然有可能會出現bug使得我們的APP在執行時shut down，如果藍芽沒有成功連接到裝置，那麼下圖第201行的mConnectedThread就會等於null，當確認完藍芽有正確連接到FPGA上的藍芽模組後，我們就用write這個function將傳送資料出去(當UP這個按鈕被按下時，我傳送字串”199” 到fpga版上的藍芽模組)，如下圖第202行所示:



至於我們為什麼要傳送199呢? 為什麼不是其他的字串? 其實我也不太清楚，199這個字串是trial & error所得到的答案，我有查過ascii code，不過也對不起來，而且很多字串都會傳送同一筆資料到FPGA上的藍芽模組，不過這並不影響我們的設計，因為我們只需要讓FPGA能夠分辨6種我所傳送出去的資料就足夠了(因為我們的APP有6個按鍵會送資料到FPGA版上，分別是AUTO、UP、DOWN、LEFT、RIGHT、STOP這6個按鈕)，trial & error 的結果如下表所示:

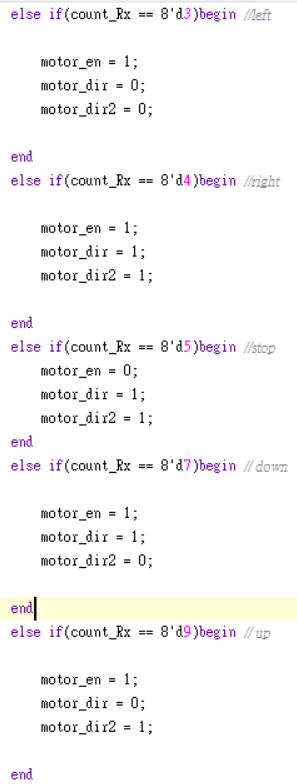
|  |  |
| --- | --- |
| APP傳送出去的資料(字串) | FPGA版上藍芽模組接收到的資料(整數) |
| 1、2、3、4、6、7、9、a | 3 |
| 5、m | 4 |
| 0、8 | 2 |
| 10、18、20、30、40 | 5 |
| 11、12、13、14、16、17、19、21、22、23、50 | 6 |
| 15、100 | 7 |
| 199 | 9 |
| fuck | 14 |

在verilog的TOP.v 中，我們利用count\_Rx這個變數來接收手機所傳送的資料，當我們在手機上按下UP這個按鈕時，我們從手機傳送字串”199”出去，根據上表，FPGA接收到的信號會是整數9，所以當count\_Rx這個變數等於9時，我們就設置一些控制馬達轉動方向的訊號，讓車子向前行進，如下圖所示(motor\_en為enable訊號，如果等於0馬達就不會轉動，motor\_dir是控制第一個馬達轉動的方向，motor\_dir2是控制第二個馬達轉動的方向，至於為什麼一個是0一個是1呢?，這個我在後面講到馬達時會多做說明)。

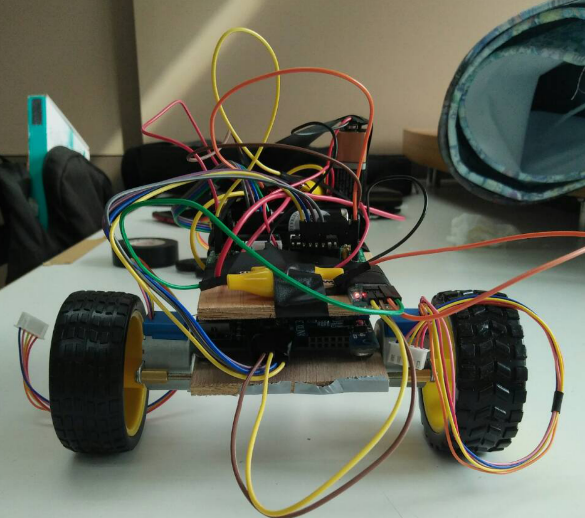


* **pmod\_step\_interface與TOP module中關於馬達部分**

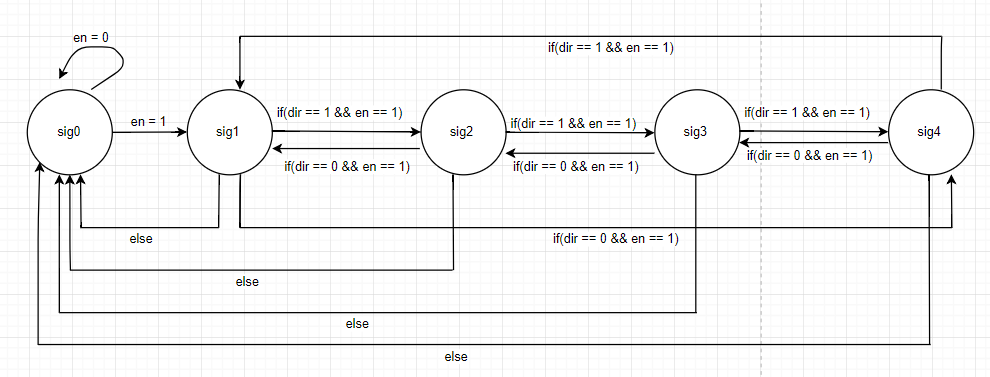
承上所述，在接收到藍芽不同的信號以後，我在verilog(TOP.v)會設置不一樣的訊號給馬達以控制它的轉動，如下圖所示:



motor\_en的運作原理相當簡單，等於1時馬達能動，等於0時馬達不能動，motor\_dir、mortor\_dir2則是分別用來控制第一個馬達與第二個馬達的轉動方向。我以前進為例(上圖最後一個else if，count\_Rx = 8’d9)，為什麼在前進時兩個馬達的旋轉方向不一樣呢? 其實這個原理也相當簡單，因為兩個輪子安裝在相反的方向上，他們是面對面的(如下圖)，所以要前進時，轉動方向要不同，轉彎時轉動方向相同。



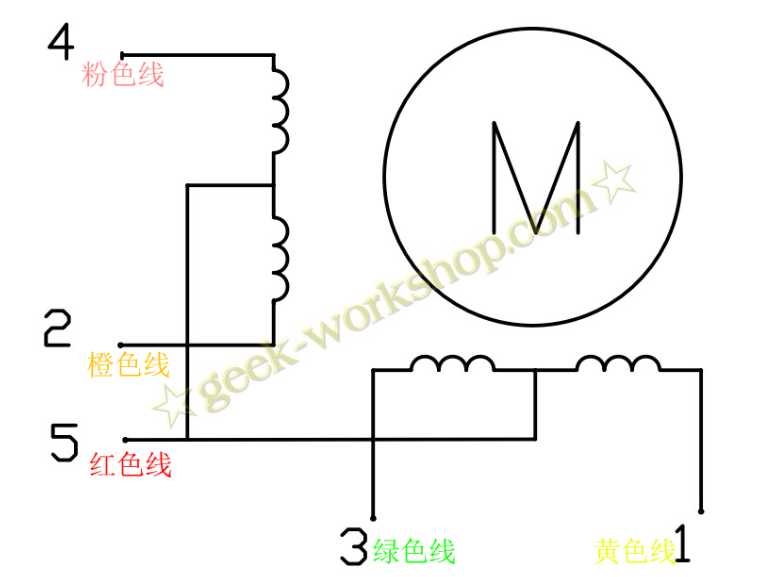
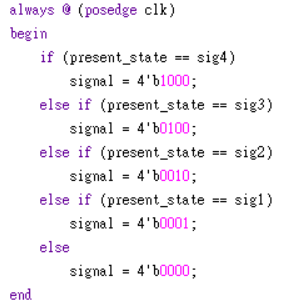
而TOP.v中motor\_en、motor\_dir、motor\_dir2這三個變數會傳送給pmod\_step\_interface這個module，然後這個module會把clk進行除頻後把新的clk以及motor\_en、motor\_dir、motor\_dir2這三個訊號傳給pmod\_step\_driver這個module，它是用來控制步進馬達旋轉的，其state transistion diagram如下:



上圖中sig0是當馬達停止時的state，也就是當en=0時都會回到sig0。

sig1 ~ sig4是控制左下圖中這四個磁鐵的state，如果dir == 1就是正轉，dir == 0則是反轉。

這個finite state machine是一個moore machine，會根據state給馬達訊號，如右下圖所示。

步進馬達結構圖

* 問題與心得：

在改題目之前我們是想做螞蟻養成遊戲，鑒於遊戲類型的final project已經寫過太多次，我們比較不感興趣，並且因為這堂課是硬體實驗，我們還是比較想做出一些實體、硬體的東西。做車子的過程中遇到很多問題，像是 接線問題：我們用的是杜邦線，有時候會需要和電源線相連接，因為沒辦法用插入的，所以常常斷路而不易察覺，有效的解決方式是用單頭的鱷魚夾夾住，或者可用麵包版，但我們沒有使用麵包版因為可能有額外支出。 電壓問題：因為我們買的紅外線模組、藍芽模組在網路上查詢spec後，很多都是需要5V電壓，但FPGA只有output 3.3 V，因此，我們用了許多不同型號的電池進行串、並聯，但也因此加重了重量。在步進馬達上，因為需要input 12V，但12V 需要串聯非常多電池，於是我們買了一個motor driver，可以input 3V左右電壓，輸出12V電壓。

總之，這次project是眾多project中，我們覺得最好玩的，雖然中途換了題目，自己摸索了馬達、藍牙、紅外線遇上不少困難，不過真的成長很多。感謝教授，助教。