Logic Design Lab

Final Project



Team34

組員：

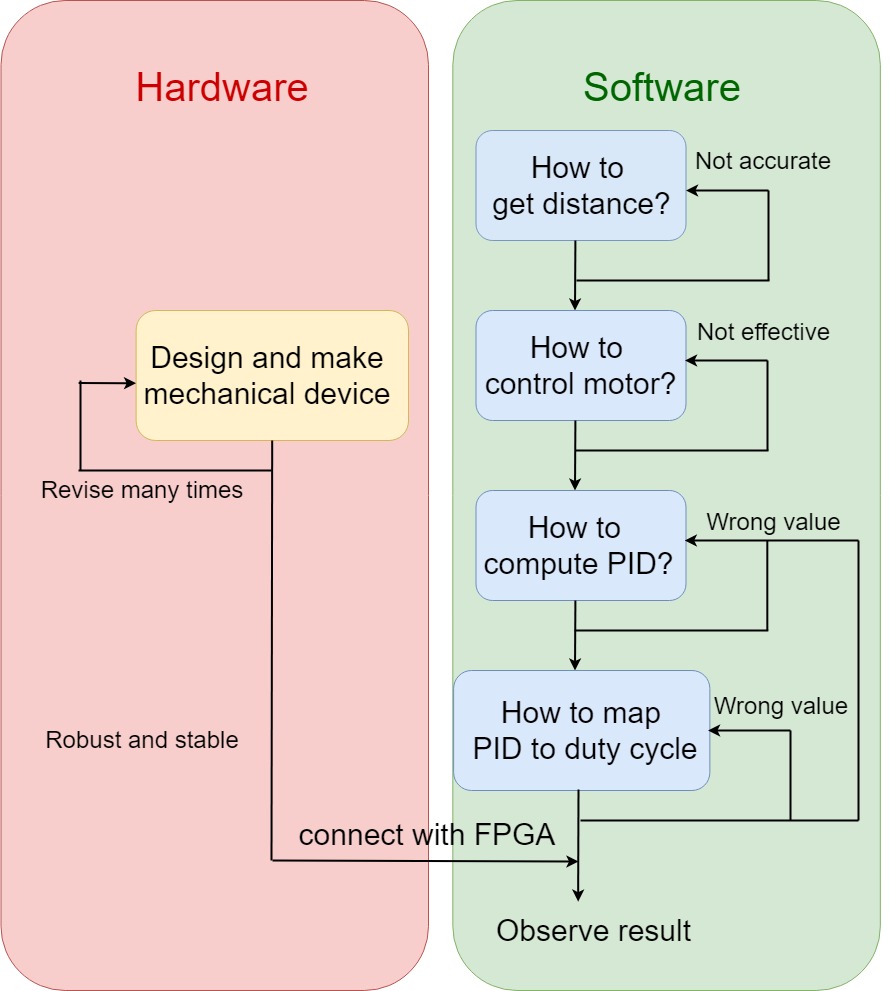
106030009 葉蓁

106030028 劉多聞

**PID balance ball**

Mode 1:

1. Motivation
2. Design flow



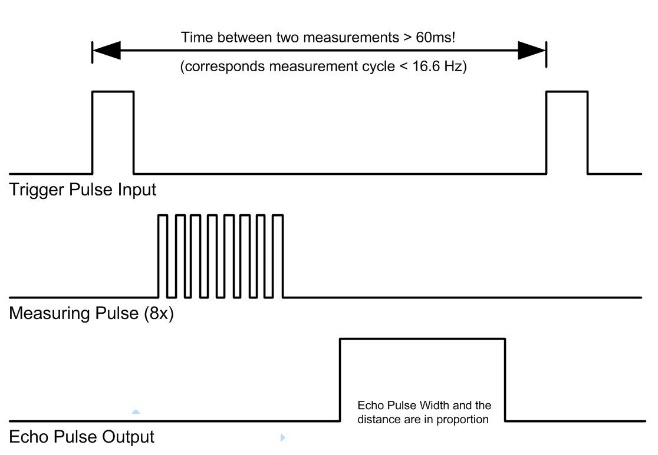
如上圖，基本上硬體軟體兩邊是同步進行的，以下先由軟體方面說明：

1. Software

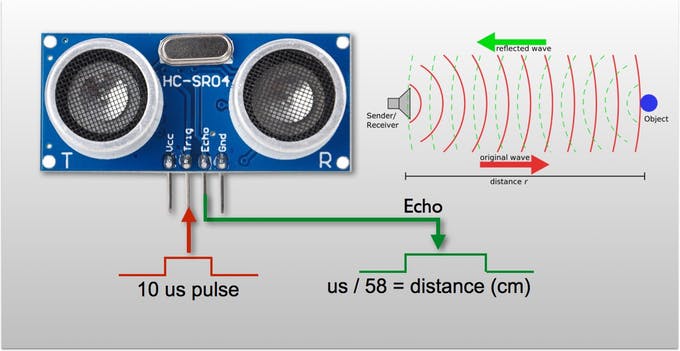
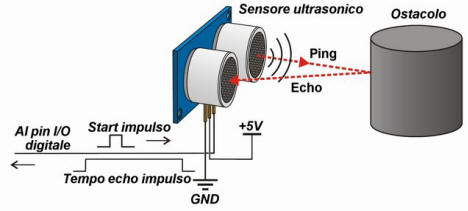
軟體方面先構思此project需要做到哪些事情。我分為三件事，剛好搭配block Diagram來看分別為讀取距離、計算PID值、驅動馬達：

1. Get Distance

這個project要成功的先決條件，就是要能準確的讀取球現在距離。如此才能得到誤差，並做後續的運算。而根據我所學過這類的sensor，較簡單的就是arduino最常用的超音波測距模組，如下圖(HC-SR04)

這個超音波運作原理相當簡易，可以看到該模組有四個腳位，扣掉power supply的Vcc跟gnd後，一個是output的trigger pin, 一個是input 的 echo pin。使用者根據右上的波形圖，當要量測距離時，將trigger pin設為High並持續至少10ms，然後再設回low，此時模組就會發出八個連續的measure pulse。接著使用者只要等到echo pin收到High訊號的時候，開始計時，計到變回Low後，這段High持續的時間，就是「兩倍」超聲波走這段距離所需的時間。兩倍是因為包含了去跟回程。而超聲波的速度就是聲波的速度，在空氣中常溫下大約340公尺/秒，用所求單位的公式換算，可得出要的距離單位。以下為網路上找到助於理解的運作波形圖。



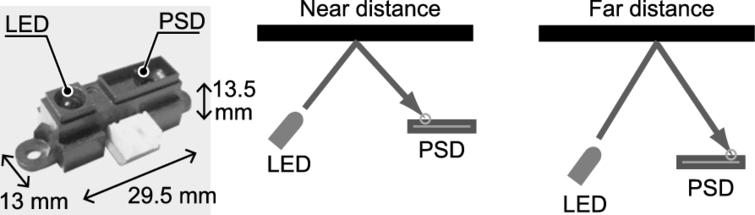
而我有實際做出用FPGA成功讀取距離，再用七段顯示器呈現出來的Ultrasonic module。Block diagram如下：

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

後來因為超音波sensor好像必須偵測比較有穩定平面的物件，比如我們的球，他可能會繞過球導致波的傳遞有偏差，量出的距離非常不精準。有效範圍非常小，大概只有15公分左右，在百般不願意之下我們選擇換另一個測距模組。

我們採用網路上許多人用的SHARP GP2Y0A21YK0F 有效範圍10-80cm(datasheet宣稱)的紅外線測距sensor。

但當時時間比較趕，它的運作原理也稍複雜，短時間內可能來不及用FPGA刻出來，因此我們想說能否透過Arduino讀取資料，再將十進位的distance用decimal to binary conversion轉換，包成6個bit的資料用jumper傳給FPGA。如此一來實驗結果：真的能有效讀取資料並傳送到FPGA板(用Seven segment顯示驗證)。我們終於順利解決第一關，進而進入下個環節。

1. Compute PID

首先要先了解PID control的原理。參考網路上的資料，我先畫出以下簡易回授圖：

一張含有 螢幕擷取畫面 的圖片

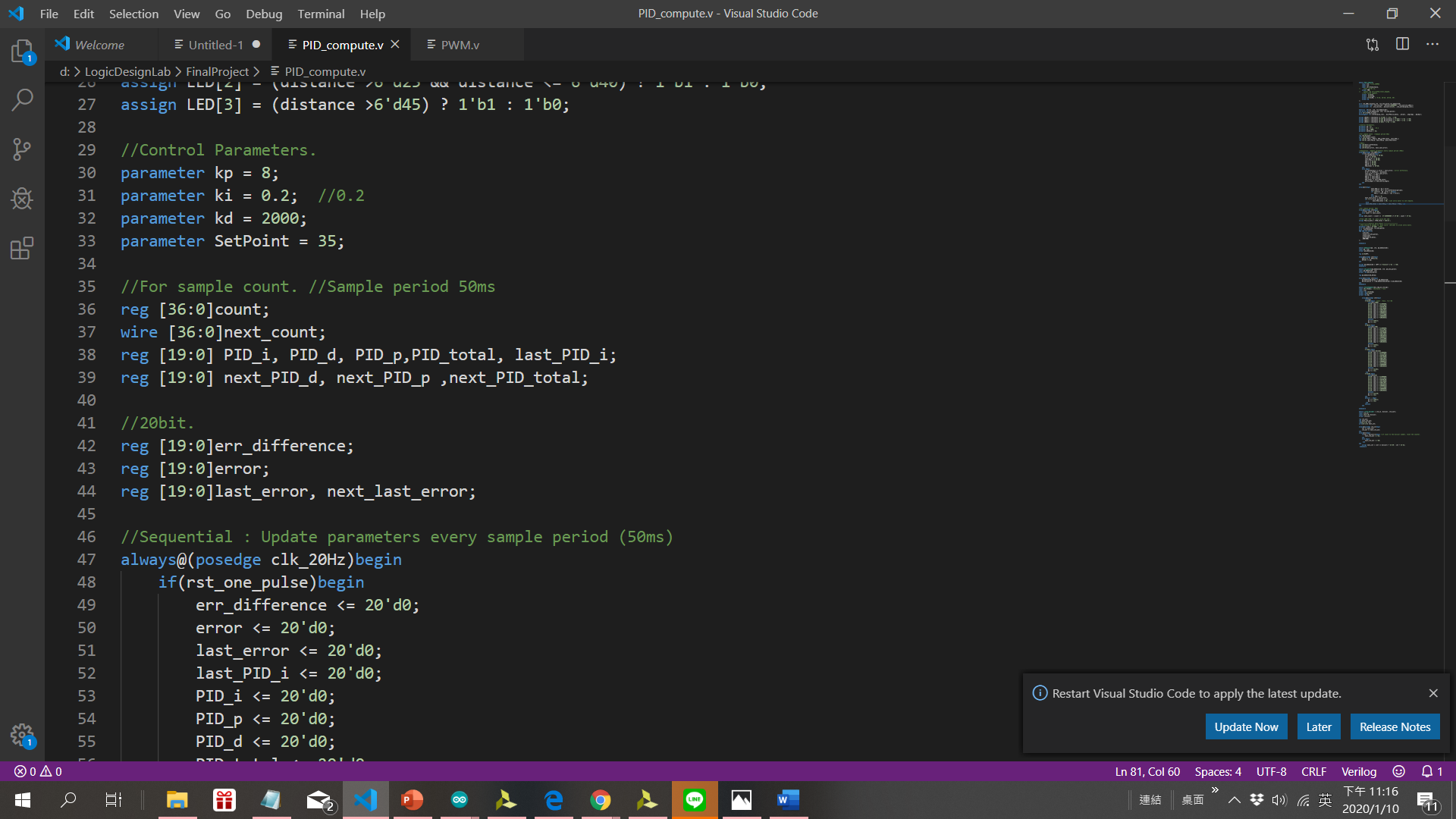
自動產生的描述

簡單解釋一下：受控場(plant)是平衡架，由伺服馬達(actuator)驅動它，得到的output是球的位置(Ball Position)。藉由一個IR sensor去讀取output，然後回授到input固定的位置(set point)，兩者相減得到一個誤差值。這個誤差值如果最簡單的控制方法，就是用一個單純的P gain去放大然後送給驅動器，讓他去追命令，減少誤差。這種只有一個放大倍率的控制叫做P control。但這種簡易的控制器有它的限制，可能會有太多的過衝量(overshoot)，或是因為system type是0的原因，即使input只是一個step還是會有穩態誤差。所以我們想用PID control，就是PD加上PI，D是Differential微分，將誤差微分後乘以比例係數(ki)，再加上I是Integral積分，最後送給驅動器。

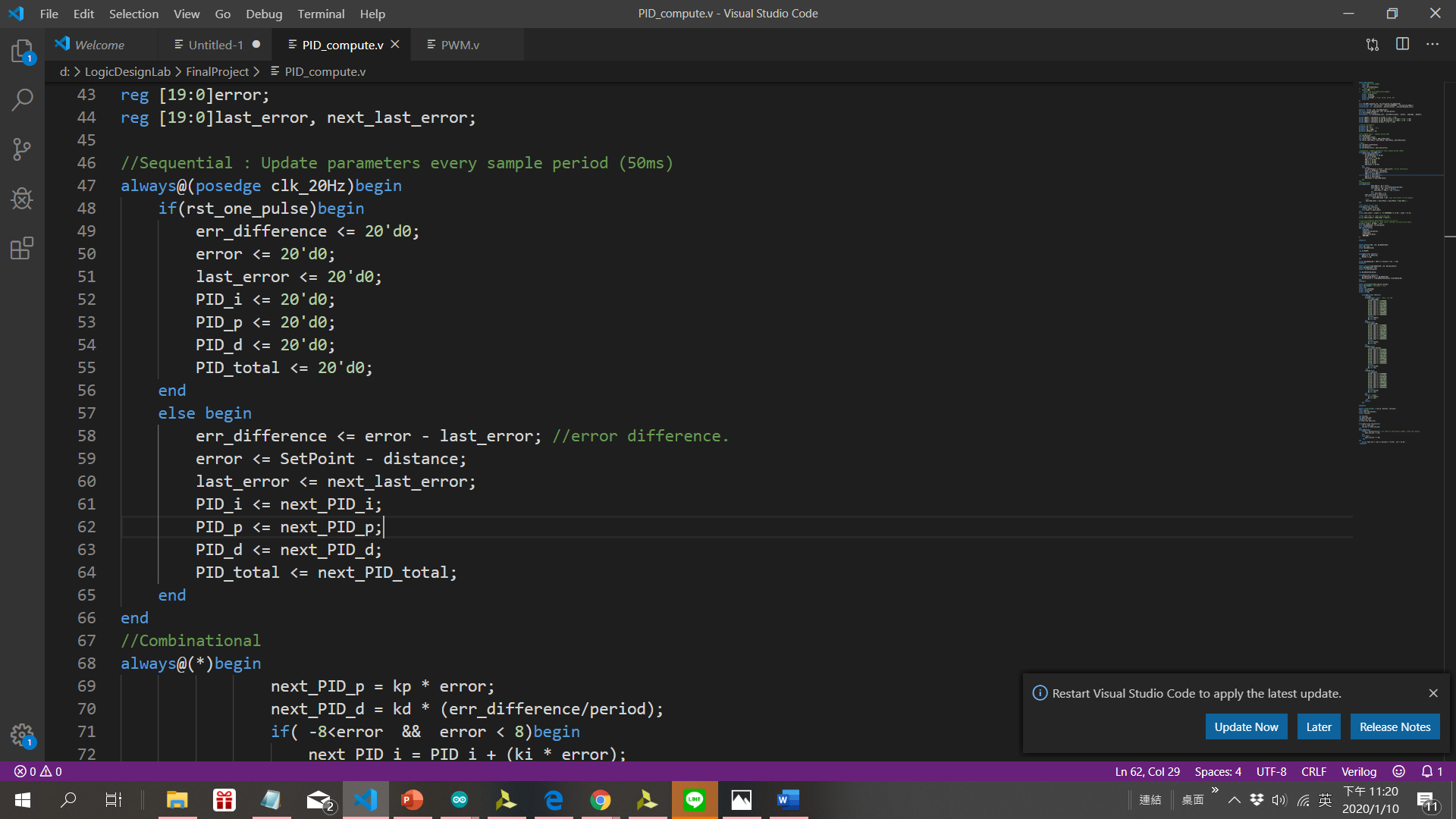
簡單講PID三個項的目的：以我們project為例：P 能夠讓我們的Ball去趨近中間setpoint的位置，D能夠考慮球的速度(位置微分)改變馬達輸出的大小，減少球過衝停不下來的問題，而I則是避免球卡再一個很靠近(誤差3-5公分卻停住不動)但錯的位置的窘境。

因此，根據上面的分析，我們的code大致如下：

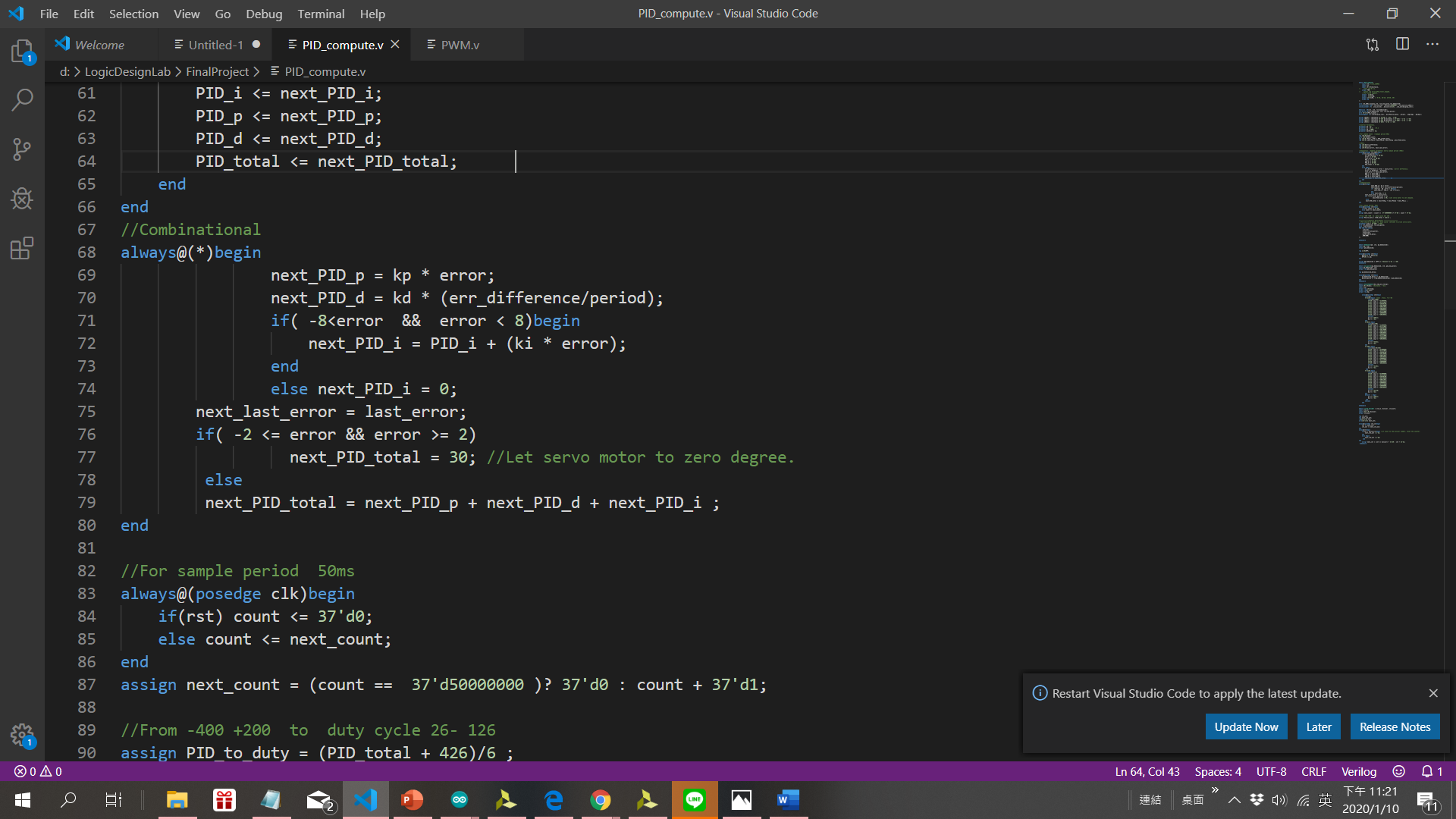
首先宣告PID的參數：



然後是Sequential的部分：更新PID運算的各種變數：



最後是Combinational，當input一變化，馬上改變值的部分。

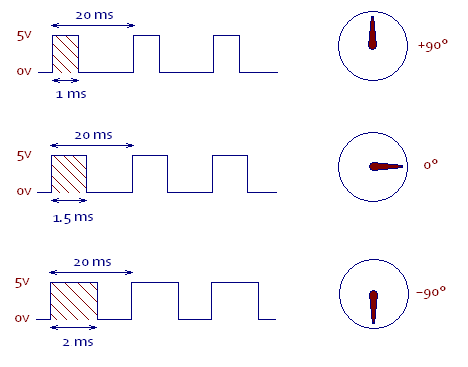
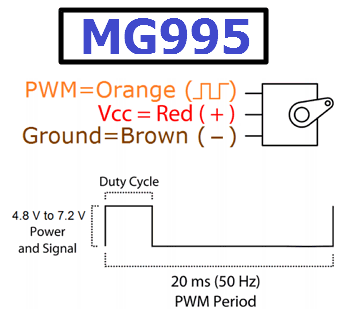
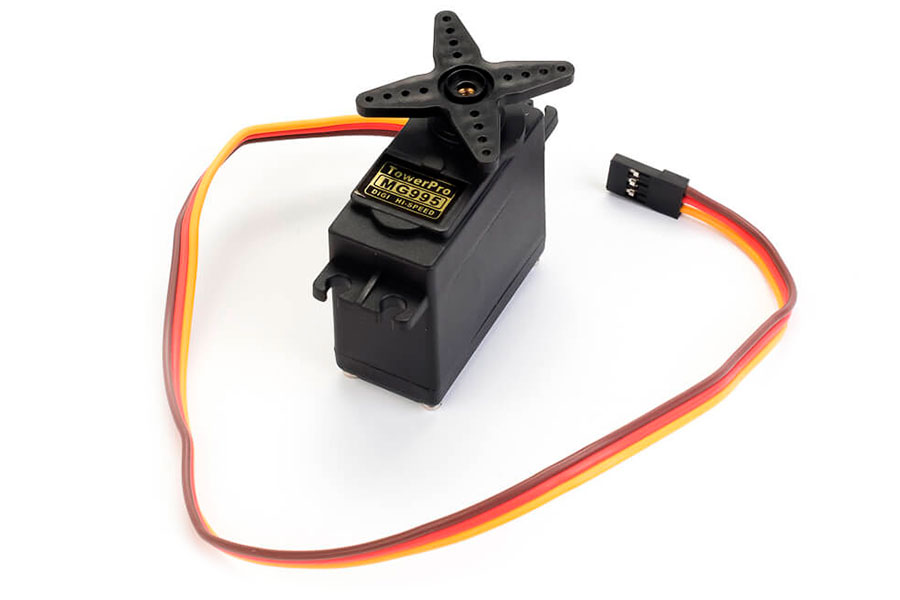


再71~74行的部分，我們讓PID的I項部分，在球的誤差小於8公分內才作用，因為實驗發現球在這個範圍內很容易卡住。加了這項有助於隨著時間變長，誤差會透過這項慢慢放大，機器才會繼續去平衡它。

然後在76~79的部分：如果誤差小於2公分內，我們就讓balance beam停住(穩定在0度角)因為發現球很容易在+-2公分左右，明明看起來平衡了，可能因為雜訊或干擾，導致sensor讀到的值一直覺得不穩定，給FPGA後會不停的震盪，雖然是輕微震盪，但我們想要看到它平衡停住的樣子，所以算是硬寫了一個邊界條件。

PID的compute到這邊算是完成了。接著要看如何用這個結果有效送給馬達，讓他聽從我們的命令去運作。

1. Drive Servo motor

我們選用的這顆MG995 Servo motor，主要原因是它能夠提供高達13公斤的扭力，因為要推動平衡架+球應該需要不少力量，我們選用這顆馬達。伺服馬達的運作原理是PWM，簡易說明圖如下：

PWM的原理就是根據不同的duty cycle，會有不同的波形結果。Pulse-Width-Modulation，脈衝寬度調變，顧名思義就是要調變脈衝的寬度，就是要有多寬的High(or low)。Duty cycle是指一個周期內，有多少比例(%)的時間signal是High的訊號。以上面波形圖觀察，我們這顆馬達datasheet寫說最大運轉角度是180度，如果我們取中間值為0度，剛好讓平衡架維持在0度的地方，那我們想讓它正負各轉90度。根據lab5的音樂模組交的PWM，我幾乎完全沿用，因為原理是一樣的!都是要產生PWM的訊號。唯一要注意的地方有兩點：

1. 這個伺服馬達PWM的頻率是 50Hz，所以freq = 50。
2. 根據duty cycle的比較，我們最小有2.5%，最大12.5%，跟1024比較大約是26 ~ 128的值。所以關鍵在這裡：

如何將PID算出來的值mapping到合理的duty cycle 範圍？

因為如果超出這個範圍，馬達很有可能有非預期的行為。

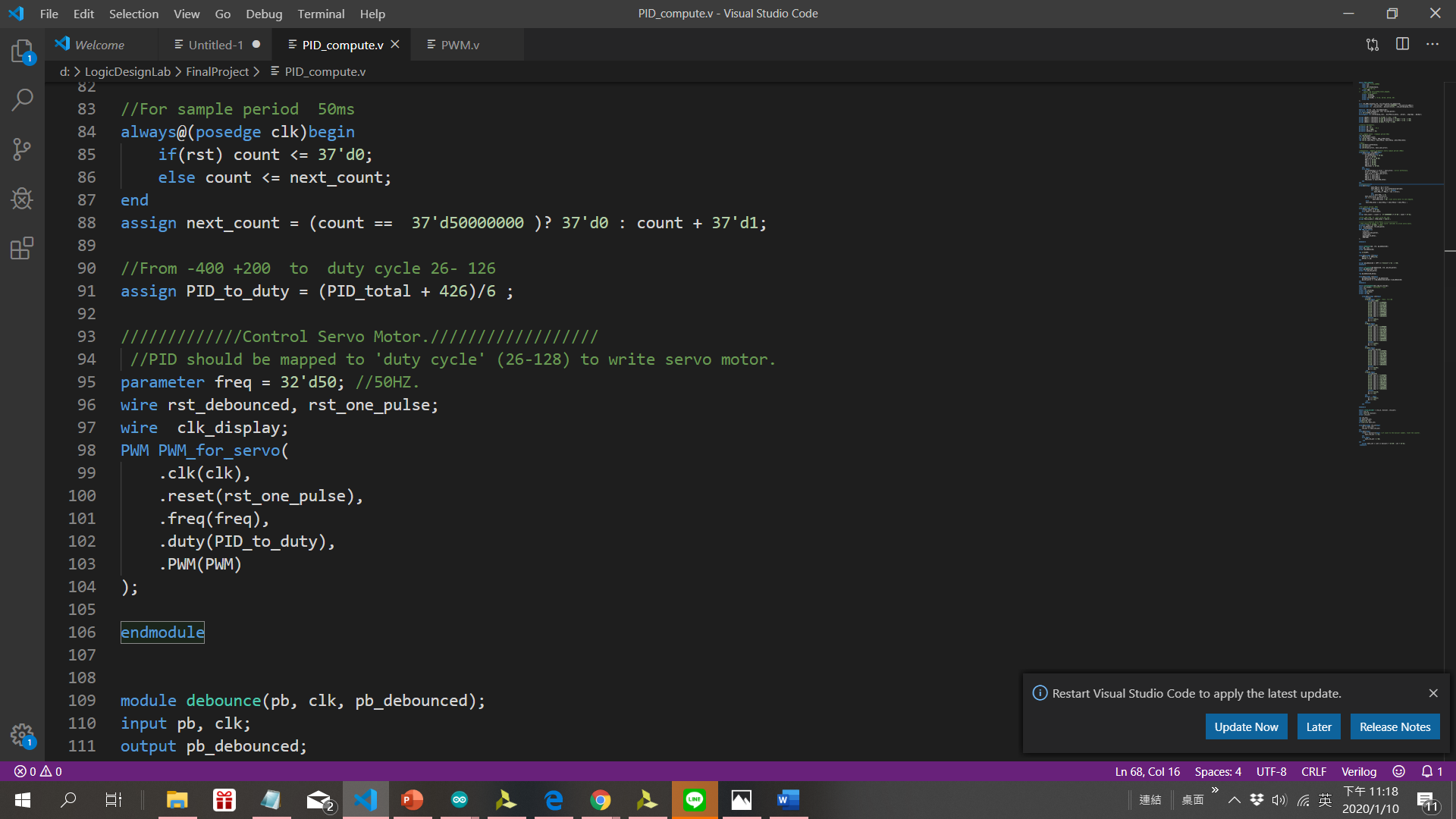
我們先單純測PID的值，讓他印出在7 segment去觀察值的範圍：

發現大概介於-400到+200之間。

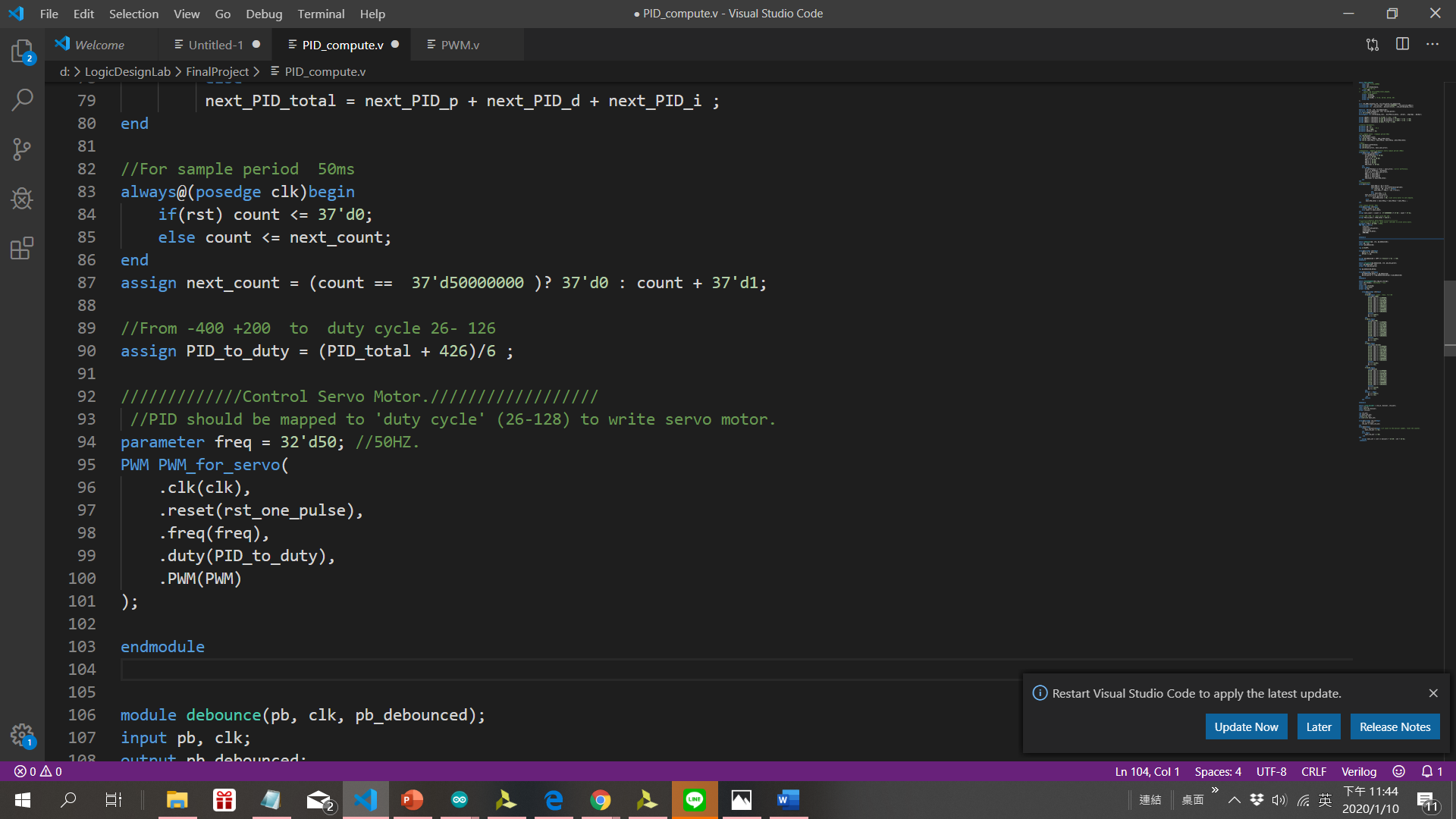
要轉換到26~126之間，我們用最簡單的比例方式，

就是先shift到同個基準，在除以6。

如此一來理應要有合理的轉換，code如下：



接著就是把map的值送給PWM模組去驅動servo motor!



伺服馬達有三條線，分別為Vcc,gnd跟signal。Signal的線我們就接到xdc檔案設定的JB port，(上圖是PWM這個signal，我記得設成JB9或10的樣子)。這樣就可以傳送訊號了！

最後附上整個Top module的block diagram一張含有 螢幕擷取畫面 的圖片

自動產生的描述:

1. Mechanism & Schematics. (Block diagram)

Schematics :

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

1. Results
2. Final Assembly:

所有機構組裝完成後的總組合圖如下：(正視圖 & 等角視圖)

一張含有 牆, 室內, 天空 的圖片

自動產生的描述 一張含有 牆, 室內 的圖片

自動產生的描述

1. Effectiveness Assesment:

最後試驗的結果：在不斷的調整參數以及修改code後，平衡架是可以將球平衡的，只是隨著球干擾的程度不同，需要的平衡時間不一樣。有時候不小心會把球停在偏差較大(約5公分)的位置，我們也還在想原因。因為時間不夠長，所以其實還有很大的進步空間。但至少基本的平衡功能有成功，這是最欣慰的QQ。

1. Discuss

迫於時間有限，demo的時候只能以目前的狀態，其實我有想到幾個問題，我們可以繼續改善的部分：

1. 能否不用Arduino 傳資料，全用FPGA去讀取紅外線sensor的資料呢？ 應該是可以的，只是可能要完全了解那個sensor運作的原理，像超音波sensor那樣，才有可能寫出他的module去讀值。
2. PID是平常廣為人知的控制理論，但它也有它的限制，我們的控制老師其實很不喜歡人家隨便用PID然後亂調kp,ki,kd三個參數的值去逼近理想結果(結果我們就再用這件事XD)也許未來有機會可以用到lead-lag control。今天聽到老師說可能要加個Q-learning，我沒有很了解reinforcement learning的理論，但有想過如果不只用控制，加入machine learning的方法，也許會讓成果更穩定。
3. 如果我們一維的做好，能否推展到2維呢？不過首先sensor就不能只靠紅外線了QQ，需要三軸加速陀螺儀等等的，然後機構方面也變複雜許多，再來code方面也需要好好思考。不過我覺得是相當有挑戰性又有趣的題目，未來有機會希望可以繼續研究!
4. 心得

葉蓁：

首先感謝這次的final project，讓我有機會實作一個PID控制的練習，也練習如何透過FPGA板子整合所有硬體和sensor。期末真的是太爆炸了，我這學期也才五門課，就有五個final project…(還有兩個還沒開始弄XD)，所以其實前兩周進度比我們proposal預期的大概再慢一半QQ。我們是星期三中午才考完所有的考試，才開始最後全力衝刺。主要的code之前有初步打好雛形，但星期三當天才正式決定要換測距的sensor，所以code方面真的滿趕的，加上硬體方面debug也de很久。這個project挑戰的地方真的滿多的，雖然看起來很普通，但光架設穩定的機構就修了滿多次的，剛開始用的超音波模組也花了不少時間debug才弄成功，興奮之餘卻馬上發現無法偵測球，宣告失敗(好挫折又可惜QQ但人生何嘗不是不停的失敗、檢討、再重新開始呢？)硬體部分確保它很robust後，大概星期四中午，我們開始全力衝刺code，從十進位換成二進位，計算pid的結果如何mapping，到如何送給馬達控制。(其實中間有很多問題，初始pid合成的，馬達根本不動，不論球的位置多遠。不確定是因為sample period太長或甚麼原因。我們一直嘗試，直到demo前一天的晚上10點多，終於成功了!當下真的非常激動，就希望成功可以在當天demo給大家看。但demo當天不曉得為何，有點不太穩定QQ這是最遺憾的，加上我太緊張presentation講的有點亂，真是抱歉QQ

最後謝謝鈺智助教支援我們HDMI轉VGA接頭，否則熬夜趕的PPT就付諸流水了~謝謝助教們和老師看我們demo~雖然很爛QQ

劉多聞 :

**分工：**

兩人一起把final project趕出來!