

Bi-weekly Personal Journal

Name: 林士程

Group: 第 26 組

Role: 機電系統控制

Objectives

討論出優先度較高或確定要使用的機電系統元件，並大致規劃其用途。畫出上述元件的線路連接圖。

Summary of Outcomes

-Task 1:

目前初步規劃將使用 Arduino Uno R3 來當作微處理器，讀取 MPU-6050 量測到的三軸加速度值和角速度值，解碼出編碼器的 AB 相數位訊號，以推算出直流馬達的角位移量，並透過馬達驅動模組 L298N 控制其轉動。

-Task 2:

Arduino Uno R3 的主要用途，為透過 I2C 接口讀取 MPU-6050 的訊號，和輸出 PWM 訊號以控制馬達。

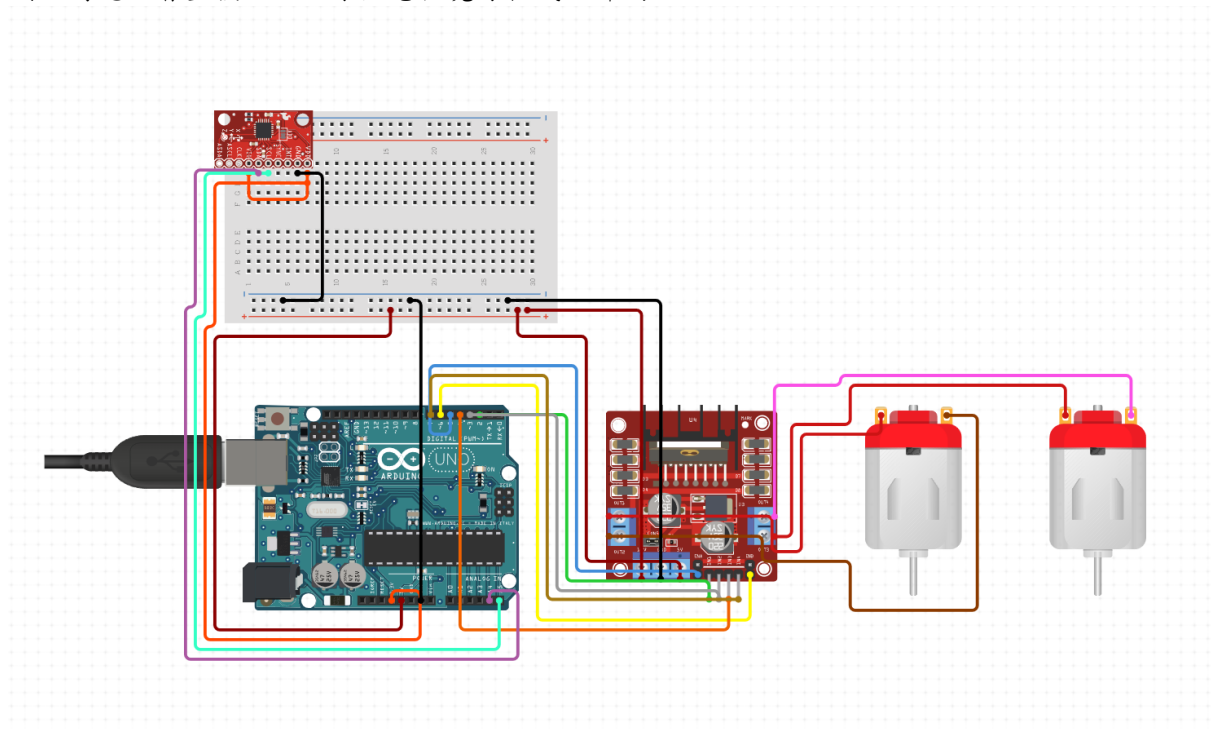
MPU-6050 其主要用途為當測量到的向下加速度高於一定值時，觸發風帆展開，次要用途為量測目前車體的三軸角度。

25GA-370 整合了直流馬達、齒輪箱和編碼器為一個模組以方便使用，其兩項主要用途為控制風帆的展開和調整風帆的轉向。

L298N 的主要用途為驅動 25GA-370 轉動，次要用途為提供微處理器供電，但這部分考慮到可能會因微處理器端阻抗不夠高而影響直流馬達的轉動，可能會另外加 buffer 提高阻抗或不使用其供電，直接獨立直流馬達和微處理器的電源。

-Task 3:

元件的線路圖大致如下圖，作為實際接線路的參考，此處微處理器未使用電池，因為之後測試時應該會直接用 usb 線供電和燒錄程式編譯碼。



Name: 林士程

Week 3

Page number 1

- Challenge 1:

目前只能大致預估風帆的重量和長寬，因而在挑選直流馬達時只能大概選用較大的扭矩量來避免扭矩不夠。

- Challenge 2:

目前還不確定要不要使用編碼輪，若使用的話可能可以完全取代 MPU-6050 的功能，這部分希望可以等聽過第五周的課再做考量，但目前認為用編碼輪有可以完整的確定車體在車道上的位置的優勢，這是加速度計相對無法做到的。

Perspectives

之前在社團就有稍微碰過編碼器和馬達相關了，但相對來說陀螺儀和加速度計就沒什麼經驗，這次是第一次使用，也藉由這次機會，了解了其腳位接法、供電電壓限制、原理和其局限性。

Plans for Next Week

- Perspective 1:

Challenge 1 應該還是要等到機構那邊稍微設計完，或是測量完帆和密集板的重量才能稍微估算出來，規劃上來說可能要到下幾周才能達成。

- Perspective 2:

Challenge 2 如上所述，希望聽完第五周的課後再來做決定，會在下一次的日記呈現。

- Perspective 3:

鑒於上述的問題可能都要等到第五周才能解決，我決定下周先在網路上找和寫一下之前沒碰過的 MPU-6050 的程式庫和程式。

Remarks

因為目前只決定用 MPU-6050，所以編碼輪的部分在上述沒有提很多，但這應該還是專題相關的，所以想在這邊提一下目前的想法，如果要確立車體在車道上的位置，至少要有兩顆分別量測 x 和 y 方向的位移量，但這樣一來就必須將其固定在一定方向，這必定會增加摩擦力使車體行進距離變短。但若能確立車體位置，並透過流場模擬軟體來決定風帆在哪裡轉向能得到最大的推力，感覺對車子行進應該還是利大於弊。

Objectives

搜索 MPU-6050 相關的程式庫，並利用其寫讀取三軸加速度值和角速度值的程式。

Summary of Outcomes

-Task1:

目前找到了由 Adafruit 和 Jeff Rowberg 提供的 MPU-6050 相關程式庫。

-Task 2:

Adafruit 的可以由其提供的範例，可以用其程式庫的函式直接定義陀螺儀和加速度計的測量範圍，還可以選擇濾波器頻寬來濾除雜訊，並可直接讀取三軸加速度值和角速度值。

Jeff Rowberg 的就相對複雜，這部分我直接參考了網上的教學影片和 MPU-6050 的 DATASHEET，來學習如何透過 I2C 將它從 Sleepmode 喚醒，定義陀螺儀和加速度計的測量範圍，和取得三軸加速度值和角速度值。

Issues and Challenge

- Challenge 1:

這周花最久理解的部分都是理解別人寫的程式的部分，畢竟沒有一些先備知識，只有計算機程式學到的一些 c 語言基礎，所以一開始看到 Wire.write() 內的 16 進制的代碼不知道是什麼意思。

- Challenge 2:

也是程式理解的部分，Wire.read() << 8 | Wire.read() 當中的 bitwise shift operator 和 bitwise or operator，要查才知道其運作原理。

Perspectives

學到 I2C 的通訊方法，能透過 DATASHEET 上提供的 register 16 進制代碼用 Wire.write() 寫入數據，改變 MPU-6050 的各項參數或讀值等功能。也學到了如何透過 bitwise operator 快速地將兩個 8bit 的值合併成一個 16bit 的值。

Plans for Next Week

- Perspective 1:

因為 MPU-6050 很容易受雜訊干擾，網路上是推薦用卡爾曼濾波算法來濾波，可以稍微研究看看，但若太過複雜可以考慮直接用 Adafruit 的寫好的濾波函式就好。

- Perspective 2:

下禮拜應該就會拿到 MPU-6050，將實際利用這周寫好的兩個程式測量三軸加速度值和角速度值。

- Perspective 3:

利用上學期量測原理的客參考網路上的教學影片寫的程式，寫出透過編碼器測量馬達的角位移量的程式。

Remarks

這周基本上都在處理跟程式相關的，想到剛好之前有碰過 Git 和 GitHub 一下，所以趁這個機會順便建了一個專門處理這次實務專題的 remote repository，來方便之後用筆電拿程式資料和測試，或許下次開會能跟組員分享一下，但感覺 google cloud 共編和 excel 還是蠻方便地，另外，也順便看了一下之前沒學的 branch 方面，結果看了之後的心得是好像用不到哈哈，畢竟程式量也不是那麼大，add、commit、push 和 pull 應該就很夠用了，也順便複習 Markdown 的語法來寫 README，雖然還是不太知道要寫啥，之後再說吧。GitHub 網址：[rumsey000/Practice-of-Mechanical-Engineering \(github.com\)](https://github.com/rumsey000/Practice-of-Mechanical-Engineering)。