

Li-Gyro 開源程式

➤ 檔案列表

表格 1 Li-Gyro 檔案列表

檔名	內容
Li-Gyro. ino	主程式，控制主要程序執行順序。
Options.h	選項設定程式，設定開啟/關閉特定狀態訊息。
Motor.h	馬達程式標頭檔
Motor.cpp	馬達程式
PID.h	PID 控制程式標頭檔
PID.cpp	PID 控制程式
Sensor.h	感測器程式標頭檔
Sensor.cpp	感測器程式

表格 1 為 Li-Gyro 程式的檔案列表，其分類方式是按功能做粗略的劃分，以增加程式的可讀性，以下章節將簡介各組程式的內容。

➤ Li-Gyro 主程式

Li-Gyro 於程式啟動的時候，開啟 softAP 模式，並初始化馬達和 MPU6050，啟動完成後，即開始等待智慧手機透過 WiFi 與之建立連線。

在等待連線建立的期間，Li-Gyro 會抑制油門輸出，並將伺服馬達限制於 90 度。當連線建立後，Li-Gyro 會按以下的步驟循環執行

1. 讀取 MPU605 狀態，取得飛行器的飛行姿態(包含 roll、pitch 和 yaw);
2. 讀取智慧手機傳送之 V7RC 命令，命令格式需符合 V7RC 協定，以 SRV 為字首，其後接續 8 個伺服馬達命令，如 SRV150015001500150015001500-15001500;
3. 收到命令後，將前一回合得到的控制命令轉換為預計控制飛行器的三軸旋轉值，即翻滾角(Roll)、俯仰角(Pitch)和偏航角(Yaw);
4. 根據步驟 1 和步驟 3 取得的數據，計算飛行器的三軸飛行誤差，再根據誤差值和預設的 PID 控制參數，計算飛行器三軸控制的補償命令，即為將飛行器拉回預期航線的修正命令;
5. 再將步驟 3 和步驟 4 取得的控制命令，以飛行器的控制特性，進行混控，轉換為直流馬達和伺服馬達的控制命令;
6. 將最後取得的馬達控制命令，下達給直流馬達和伺服關，進行控制;

7. 將本回合得到的控制命令餵進PID控制器，作為下一回命PID控制計算使用。

➤ Motor 程式

Motor 於程式啟動的時候，為直流馬達和伺服馬達進行初始化，並在收到 Li-Gyro 主程式的呼叫時，依照控制命令來驅動對應的直流馬達和伺服馬達。

➤ PID 控制程式

PID 提供 Li-Gyro 主程式步驟 3、4 和 7 時，所需要的操作。

目前程式中提供馬達控制的參考樣本有：雙馬達差速遙控飛機、雙馬達差速遙控氣墊船和單馬達固定翼遙控飛機等，其設定如下：

- 雙馬達差速遙控飛機：其設定原理為，先將雙邊馬達設定為預期油門轉速，再依方向控制進行加/減轉速控制，最後依偏航角補償命令為雙馬達進行轉速修正；
 - $m1_command_scaled = thro_des + yaw_weight * (yaw_des/maxYaw) + yaw_PID;$
 - $m2_command_scaled = thro_des - yaw_weight * (yaw_des/maxYaw) - yaw_PID;$
 - $s1_command_scaled = 0;$
 - $s2_command_scaled = 0;$
- 雙馬達差速遙控氣墊船：其設定原理為，將中置馬達(提供氣墊船浮力用)設成預期油門控制。再將雙馬達的轉速控制設定為預期俯仰角控制，作為氣墊船前進/後面和移動速度的控制命令，接著再雙馬達的轉速控制依預期偏航角控制進行加/減，以實現氣墊船轉向控制，最後最後依偏航角補償命令為雙馬達進行轉速修正；(注意：雙馬達需另接直流馬達電調，才能將伺服馬達控制命令，轉換為直流馬達正反轉控制命令)
 - $m1_command_scaled = thro_des;$
 - $m2_command_scaled = thro_des;$
 - $s1_command_scaled = (pitch_des/maxPitch) + (yaw_des/maxYaw) + yaw_PID;$
 - $s2_command_scaled = (pitch_des/maxPitch) - (yaw_des/maxYaw) - yaw_PID;$
- 單馬達固定翼遙控飛機：
 - $m1_command_scaled = thro_des;$
 - $m2_command_scaled = thro_des;$
 - $s1_command_scaled = (yaw_des/maxYaw) + yaw_PID;$
 - $s2_command_scaled = (pitch_des/maxPitch) - pitch_PID;$

➤ Sensor 程式

Sensor 於程式啟動的時候，為 MPU6050 進行初始化設定，以及校準的操作。並在每個控制回合中，取得飛行器的飛行姿態，提供相關參數給 Li-Gyro 進行計算。由於網路上已有許多相關教學，本文在此即不再贅述 MPU6050 的動作。