自走小車 12組

姓名：張峻瑋

學號：110511194

系級：電機114

1. 整體架構：

Arduino Mega 2560

超音波感測器

馬達編碼器

光達

樹莓派

攝影機

圖1：小車各模組架構

在架構上，我們將小車分為2個運算模組：Arduino板以及樹莓派。除了超音波感測器是在Arduino端進行運算外，其餘運算皆在樹莓派上進行。互動規則如下：

* 1. 攝影機將資料傳給樹莓派，並在樹莓派上進行圖像處理。
  2. 光達將資料傳給樹莓派，並在樹莓派上進行資料處理。
  3. 超音波感測器接在Arduino板上，將資料傳給Arduino板進行距離運算。
  4. 樹莓派綜合攝影機、光達等資訊，處理後形成對馬達的指令，傳給Arduino板。Arduino板有時也會回傳一些flag給樹莓派。
  5. Arduino板會依照自樹莓派收到的馬達指令，搭配超音波的資訊來操控馬達的編碼器。我們沒有做回授機制，故馬達轉速並沒有被回傳給Arduino板或樹莓派。

1. 控制流程
   1. PID循線模式

演算法：

1. 初始化鏡頭、串口，定義找線範圍、斜率範圍、控制器參數
2. While True:
3. 讀取一幀畫面
4. 三角形影像處理
5. 影像處理
6. 存取並繪製左、右車道線
7. 計算PID控制輸出並輸出指令
8. 讀取回傳資料，判定是否跳出迴圈
9. 三角形識別
10. 設定畫面資訊及鍵盤控制
11. 重複3-9步驟30次，之後進入光達模式

控制器參數：。

三角形影像處理：

1. 將圖像轉換為灰階
2. Canny 邊緣檢測
3. 輪廓檢測

影像處理：

1. 建立遮罩
2. 灰階、二值化
3. 侵蝕、膨脹
4. Canny 邊緣檢測
5. 霍夫變換找線

存取並繪製左、右車道線：

1. 如果找到線且斜率在適當範圍內 → 存取並繪製
2. 繪製左右車道平均線
3. 計算兩條平均線的中點
4. 繪製螢幕中央點與車道中央點，若只找到一條線則利用偏移量進行運算
5. 將原本畫面加上平均線和點並輸出

三角形識別：

1. for 每個輪廓
2. 計算逼近多邊形
3. 如果逼近的多邊形有3個頂點，且面積大於3000，則認為是 三角形
4. 用三個頂點的座標判斷三角形指向

PID控制誤差：畫面中央與車道中央的距離。

* 1. 光達模式

演算法：

1. if 自走小車離開循線區
2. 暫停，給光達一點時間初始化
3. if 小車離右牆過近 → 向左轉
4. else
5. while 前面尚未遇到牆
6. 直走
7. if 左邊或右邊有障礙物
8. 相對應左右轉以閃避（模糊控制）
9. 停下來→左轉→停下來
10. 重複5 - 8且盡量靠右，直到前面、右邊皆是牆為止

心得：

一開始會修習這門課是個意外，因為原本選的實驗課停開，剛好看到這門課人數還沒有滿，而且時間上是空堂，故選之。在修習這門課前，並沒有修習任何與自動控制相關的課程，想說課程大綱上也說不需要任何先修課程。對於這點我的組員很訝異，他們兩人當時皆正在修習自控二，他們很難想像那上半學期的作業我是怎麼寫的。不過我覺得助教講得夠清楚其實就沒有太多問題。

我從來沒有想過會拿第一名，應該說直到展示的前一天才開始覺得也許有機會。這門課非常紮實，不但會用到各種控制相關理論，所需程式涵蓋Python, C++, Arduino, MATLAB，也學習了怎麼使用ROS。我到現在還不知道我到底為什麼會花這麼多時間在這門課上，從第一個作業，花了整整三個整天在畫車道線開始，或是更早以前，第一份MATLAB作業開始，我想，與其說熱愛控制，不如說這門課給我的回饋感夠強，好像只要多做一點什麼，就會有些看得到的成果，以致於就不自覺花了許多時間在實踐。

這門課強調的是系統整合，這點與我們的實驗非常相符，將光達、馬達、超音波感測器，以及兩個主要控制單元樹莓派、Arduino做良好的互動，很慶幸最後是選上了這門課，受益良多。