# Final Project Report

109061229 電機系 24 級 曾旭廷

Hsuting22822@gmail.com

2022/6/6

---

壹、簡介

本次期末專題是設計一台能夠跟隨提前繪製好之黑線地圖之自走車，加上一整學期以來學習的各項電子零件，組合馬達零件 Servo、測距離零件 Ping、無線控制零件 XBee、輪速偵測光圈零件 Optical Encoder、偵測黑線零件 QTI Sensor。

貳、零件

甲、伺服馬達

本次期末專題我使用的是 Parallax High Speed Continuous Rotation Servo，可以經過調變脈衝寬度來調整旋轉角度，擁有比一般伺服馬達三倍的轉速。

- Calibration 校正：1500 之脈衝寬度，整體馬達應處以停止狀態，若有轉動的情況發生，需轉動馬達後方的校準螺絲，如發生抖動的情況，有可能導致內部齒輪磨損。

乙、超音波距離感測器

本次期末專題我使用的是 Parallax PING Ultrasonic Sensor，藉由發出超聲波，到目標的距離通過計算波遇到目標反射後再次接收的時間。

丙、無線控制零件

本次期末專題我使用的是 Digi International XBee-S2C Module，藉由建立 UART 溝通管道，使用兩個 XBee，一個連接電腦，一個裝置在自走車上，以此即時回傳自走車速度及行走距離。

丁、輪速偵測光圈零件

本次期末專題我使用的是 Parallax Boe Bot Digital Encoder Kit，裝置在輪圈後方，偵測每次亮暗輪圈上的洞間距，來測量總共旋轉幾圈。

戊、黑線偵測裝置

本次期末專題我使用的是 Parallax QTI Sensor，裝置上有紅外線發射器及接收器，藉由電路設計搭配電容來區別黑線與白地，黑線會吸掉發出的能量，白地則會完全返回，就能夠區別電容的放電時間。

參、零件安裝配置

| | |
|---|---|
| Servo Motor Left | D10 |
| Servo Motor Right | D11 |
| Optical Encoder | D3 |
| PING Ultrasonic module | D9 |
| XBee | D1, D0, 9600 |
| QTI Sensor | D4, D5, D6, D7 |

肆、Check point design

甲、Line Following using QTI

QTI Sensor 在一開始比較不容易上手，我參考了 youtube 影片，https://youtu.be/eEpd7MirKEg，來設計我的 QTI 軟件。QTI signal 的 data type 需設計成 Digital In/Out，在過程當中此 pin 需要先當成 output，再轉為 input。QTI 設計原理，一開始先讓其充電，約待 1ms 後，將 pin 轉為輸入，然後間隔「特定時間」測量其輸入。

**特定時間：需自行測量，與實驗所設計之地板材質、顏色有關，以我的白色 PP 板為例，時間約需要落在 400 微秒左右。**

→ Function **qti_sensing**

乙、Servo Motor Module

QTI 需搭配 Servo Motor 的設計，來達到 Line Following 的功能。考量到右轉、左轉、原地迴轉，需要考慮到地板材質以及整體車子重量、旋轉角度，需要設計不同的時間來達成，故本函數希望能設計暨能調控左輪速度、右輪圈速度、以及持續時間。

```
void Moving(){
        int time = 0;
        pin5.period_ms(20);  // Set the PWM period 20ms
        pin6.period_ms(20);
        while (1) {          // loop
            wL = 0;
            wR = 0;
            if (US < 25)     // PING Distance < 25 cm then
            {
                wL = 60;     // rotate 180 degree
                wR = 60*0.6;
                time = 200000;   // last for 200000 times loop time
            }
```

```
// 輸入訊號為 1000，需要大幅度的左轉
            else if (Sensor1 == 1 && Sensor2 == 0 && Sensor3 == 0 && Se
nsor4 == 0){
                wL = velocity * 0.05;
                wR = -velocity;
                time = 1;
            }
// 輸入訊號為 1100，需要中幅度的左轉
            else if (Sensor1 == 1 && Sensor2 == 1 && Sensor3 == 0 && Se
nsor4 == 0){
                wL = velocity * 0.2;
                wR = -velocity;
                time = 1;
            }
// 輸入訊號為 1110，這是左轉訊號，持續 40000 個 machine cycle.
            else if (Sensor1 == 1 && Sensor2 == 1 && Sensor3 == 1 && Se
nsor4 == 0){
                time = 40000;
                wL = 100 * 0.1;
                wR = -100;
            }
// 輸入訊號為 0111，這是右轉訊號，持續 40000 個 machine cycle.
            else if (Sensor1 == 0 && Sensor2 == 1 && Sensor3 == 1 && Se
nsor4 == 1){
                time = 40000;
                wL = 100;
                wR = -100 * 0.1;
            }
// 輸入訊號為 0110，代表車子走在正確的路上，故繼續前進。 Velocity = 60
            else if (Sensor1 == 0 && Sensor2 == 1 && Sensor3 == 1 && Se
nsor4 == 0){
                wL = velocity;
                wR = -velocity;
                time = 1;
                }
// 輸入訊號為 0011，需要中幅度的右轉
```

```
            else if (Sensor1 == 0 && Sensor2 == 0 && Sensor3 == 1 && Se
nsor4 == 1){
                wL = velocity;
                wR = -velocity * 0.2;
                time = 1;
            }
// 輸入訊號為 0100，需要小幅度的左轉
            else if (Sensor1 == 0 && Sensor2 == 1 && Sensor3 == 0 && Se
nsor4 == 0) {
                wL = velocity * 0.8;
                wR = -velocity ;
                time = 1;
            }
// 輸入訊號為 0010，需要小幅度的右轉
            else if (Sensor1 == 0 && Sensor2 == 0 && Sensor3 == 1 && Se
nsor4 == 0) {
                wL = velocity;
                wR = -velocity * 0.8;
                time = 1;
            }
// 輸入訊號為 0001，需要大幅度的右轉
            else if (Sensor1 == 0 && Sensor2 == 0 && Sensor3 == 0 && Se
nsor4 == 1){
                wL = velocity;
                wR = -velocity * 0.05;
                time = 1000;
            }
// 輸入訊號為 0000，代表完全出界，車子停止。
            else if (Sensor1 == 0 && Sensor2 == 0 && Sensor3 == 0 && Se
nsor4 == 0) {
                wL = 0;
                wR = 0;
                time = 1;
            }
// 輸入訊號為 1111，代表終止線，車子停止。
            else if (Sensor1 == 1 && Sensor2 == 1 && Sensor3 == 1 && Se
nsor4 == 1) {
                    wL = 0;
```

```
                    wR = 0;
                    time = 1;
            }
```

由上述判斷式決定左轉、右轉、持續時間後，用此 loop 來調變。

```
            for (int j = 1 ; j <= time ; j++) {
                servo0_control(wL);
                servo1_control(wR);
            }
        }
    }
```

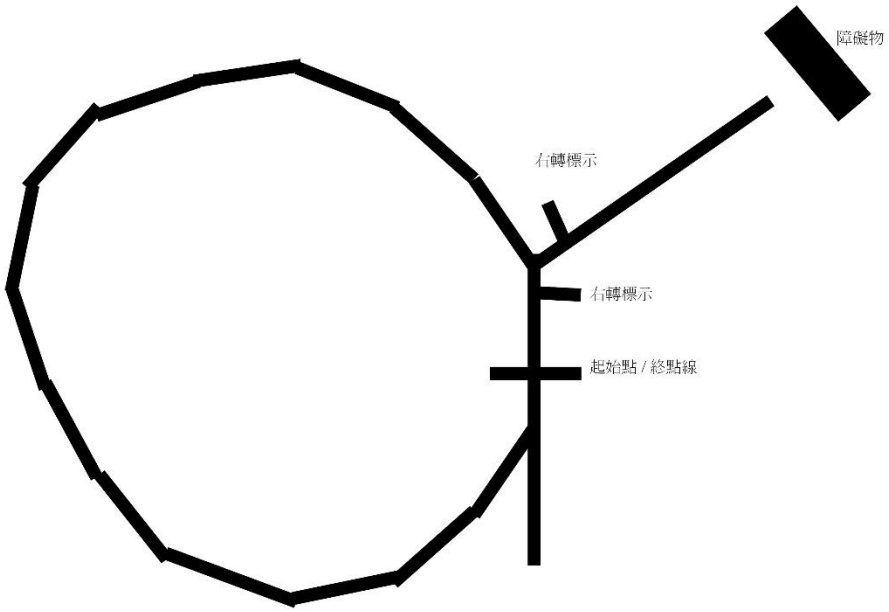
丙、XBee

本次期末專題需要搭配遠端控制功能，所以我們需要先加入 RPC 相關的 Header file，把 erpc service function 以 thread 的方式 Run，但因為 service 是一個無限迴圈，就算是以 thread 的方式也會導致其他 thread 無法正確運行，所以就必須再適當的時間才開啟 thread。亦可以使用 EventQueue 來達到間隔一段時間才開啟的功能，但是因為我們回傳的資料應該是即時性的，所以時間頻率應該要能配合，否則就會讓車子變成走一步停一步的感覺，停一步的原因即是等待 service 開關的時間。

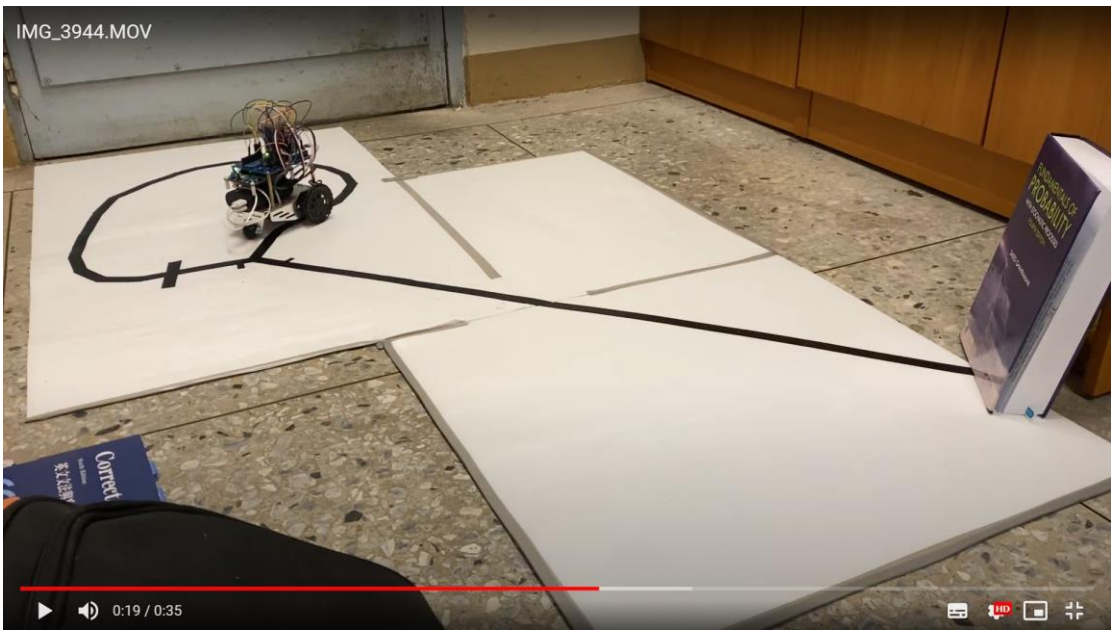
使用 ERPC 藉由接收遠距指令，我在程式中另設一變數，用來開關 service，而 XBee 傳輸資料可以直接使用 UART 的溝通橋樑，當然也可以設計 ERPC 功能來傳送資料。但本次設計我以 ERPC 功能來控制傳送資料的開關。一旦開始接收 XBee 訊號，旋即關掉 Erpc 的 service，這樣做就不會有衝突，當每次要傳資料的時候再打開，就不會一直開啟而影響到其他執行緒的運行。

伍、地圖設計

本次地圖設計須含有一個頭尾相接的封閉圓形以及一個分支路口，個人選擇以最簡潔的方式設計地圖。




陸、實拍



影片連結：https://drive.google.com/file/d/1TuC36W-4JihVcrF4FJaFk1Ey7bcNT3y1/view?usp=sharing
Github: https://github.com/hsuting0627/final
Code package: https://drive.google.com/file/d/1U1aQzE8mom6O4qhhBQPVqsas9JMpjEMr/view?usp=sharing