



物聯網

作業三
M1354020 林昀佑



功能說明

車子會左轉回正

車子沿著黑線倒退
沒有黑線後停下



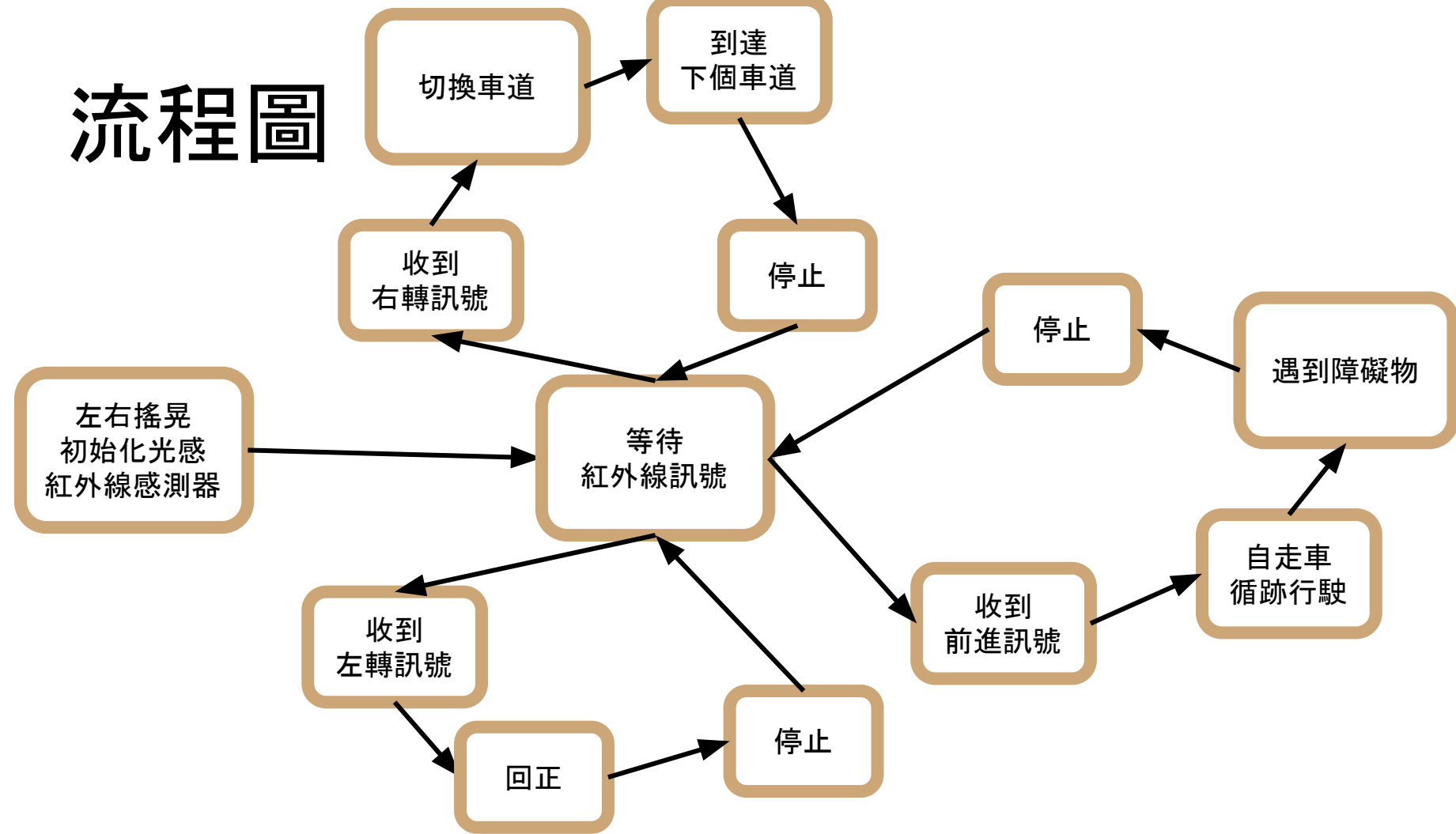
車子沿著黑線循跡前進
如有偏移則會自動回正
遇到黑線盡頭或前方有障礙物
就停下等候指令

車子會右轉變換車道
直到遇到黑線停止

程式架構

- 馬達控制模組
 - 負責步進功能
 - 馬達功能包裝
 - 前、後、左、右、停止
 - 右輪、左輪 細部設定
- 感測器模組
 - 負責感知器部分功能
 - 紅外線訊號
 - 超音波
 - 光感測器
- 主程式
 - 統合需要感測與馬達同時作用的功能
 - 以狀態機為基礎的系統

流程圖



程式碼 - 馬達控制模組 - 初始化部分

```
#include "Arduino.h"
```

```
#include "sensor_mod.h"//引入library
```

```
#define LPWM 6
```

```
#define left_foward A0
```

```
#define left_back A1
```

```
#define RPWM 5
```

```
#define right_back A3
```

```
#define right_foward A2 //設定Pin腳
```

```
unsigned int last_proportional = 0;
```

```
long integral = 0;
```

```
int Distance = 0;//運算用的共通函數
```

```
void init_motor(int max) { //初始化馬達腳位
```

```
    maximumSpeed = max;
```

```
    pinMode(LPWM, OUTPUT);
```

```
    pinMode(RPWM, OUTPUT);
```

```
    pinMode(left_foward, OUTPUT);
```

```
    pinMode(left_back, OUTPUT);
```

```
    pinMode(right_back, OUTPUT);
```

```
    pinMode(right_foward, OUTPUT);
```

```
}
```

程式碼 - 馬達控制模組 - 兩輪細部設定&前後左右

```
void left_speed_set(int speed) { //設定左輪速度 可支援前進後退
    if (speed >= 0) {
        analogWrite(LPWM, speed);
        digitalWrite(left_foward, HIGH);
        digitalWrite(left_back, LOW);
    } else {
        analogWrite(LPWM, -1 * speed);
        digitalWrite(left_foward, LOW);
        digitalWrite(left_back, HIGH);
    }
}

void right_speed_set(int speed) { //設定右輪速度 可支援前進後退
    if (speed >= 0) {
        analogWrite(RPWM, speed);
        digitalWrite(right_foward, LOW);
        digitalWrite(right_back, HIGH);
    } else {
        analogWrite(RPWM, speed);
        digitalWrite(right_foward, HIGH);
        digitalWrite(right_back, LOW);
    }
}
```

```
void forward(int speed) { //前進
    left_speed_set(speed);
    right_speed_set(speed);
}

void backward(int speed) { //後退
    left_speed_set(-1 * speed);
    right_speed_set(-1 * speed);
}

void left(int speed) { //原地左轉
    left_speed_set(-1*speed);
    right_speed_set(speed);
}

void right(int speed) { //原地右轉
    left_speed_set(speed);
    right_speed_set(-1*speed);
}

void stop() { //停止
    left_speed_set(0);
    right_speed_set(0);
}
```

程式碼 - 感測器模組-初始化

```
#include <TRSensors.h>
#include <IRremote.h>
```

```
#define NUM_SENSORS 5
#define ECHO 2
#define TRIG 3
//遙控器的按鈕對應值
#define ir_forward 0xFF18E7 // 2
#define ir_backward 0xFF4AB5 // 8
#define ir_left 0xFF10EF // 4
#define ir_right 0xFF5AA5 // 6
#define ir_line 0xFF38C7 // 5
```

```
TRSensors trs;
unsigned int sensorValues[NUM_SENSORS];
```

```
IRrecv irrecv(4); //紅外線pin腳位設定
decode_results results; // 紅外線解碼出來的資料
unsigned long lastReceiveTime = 0;
unsigned long timeout = 200;
```

```
void init_sensor() {
    pinMode(ECHO, INPUT); // 測障超音波 測回波
    pinMode(TRIG, OUTPUT); // 測障超音波 發射波
    irrecv.enableIRIn(); //紅外線
    trs = TRSensors(); // 光感
    for (int i=0; i<200; i++) {
        trs.calibrate();
    }
    delay(500);
}
```

程式碼 - 感測器模組-用到的功能

//取得光感資料

```
unsigned int getSV(int num) {  
    return sensorValues[num];  
}
```

// 超音波偵測

```
int Distance_test() {  
    digitalWrite(TRIG, LOW);  
    delayMicroseconds(2);  
    digitalWrite(TRIG, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(TRIG, LOW);  
    return (int)(pulseIn(ECHO, HIGH)/58);  
}
```

// 紅外線接收

```
String getAction() {  
    if (irrecv.decode(&results)) {  
        switch (results.value) {  
            case ir_forward://2  
                return "track";  
            case ir_line://5  
                return "forward";  
            case ir_backward://8  
                return "backward";  
            case ir_right://6  
                return "right";  
            case ir_left://4  
                return "left";  
        }  
        irrecv.resume();  
    }  
}
```


程式碼 - 主程式 - 初始化 & 綜合循跡

```
#include "motor_mod.h"
#include "sensor_mod.h"
//公共變數定義
// 執行速度
int speed = 100;
//轉彎速度
int turn_speed = 60;

void setup() {
    //初始化馬達模組
    init_motor(speed);
    //初始化感應器模組
    init_sensor();
    Serial.begin(115200);
}
//初始化動作變數(狀態機需要)
String action = "none";
```

```
// 可前進或後退的循跡
void track(String act) {
    if (
        getSV(3)<400 &&
        getSV(2)<400 &&
        getSV(1)<400) { // 偵測是否在線上
        stop();
        action = "none";
    }
    else {
        if (act.equals("track")) {
            forward(speed);
        }
        else if (act.equals("backward")){
            backward(speed);
        }
    }
}
```

程式碼 - 主程式 - 狀態機

```
void loop() {  
    action=getAction();//取得IR控制資料  
    Serial.println(action);  
  
    if (action.equals("track")) {  
        if (Distance_test() <= 15){  
            stop();  
            action = "none";  
        }else{  
            track(action);  
        }  
    }  
    else if (action.equals("forward")) {  
        if (Distance_test() <= 15){  
            stop();  
            action = "none";  
        }else{  
            forward(100);  
        }  
    }  
}
```

```
    else if (action.equals("backward")) {  
        track(action);  
    }  
    else if (action.equals("right")) {  
        left_speed_set(turn_speed);  
        right_speed_set(0);  
        delay(300);  
        stop();  
        action = "none";  
    }  
    else if (action.equals("left")) {  
        left_speed_set(0);  
        right_speed_set(turn_speed);  
        delay(300);  
        stop();  
        action = "none";  
    }  
}
```

問題討論

1. 後退時沒有偵測器可用，導致車容易超出範圍

解決方式

1. 盡量維持直線以及增加偵測頻率
(減少delay時間)

成果影片

<https://youtube.com/shorts/f7r7PuQJKYE>