控工實驗 LAB2 結報

LEGO 機器人 part2

Group 2

林珮玉 E24084096　　林哲緯 E24086129

林玠志 E24083074　　薛博文 F14071148

蔡孟宗 F44071055

Control Engineering Laboratory – Lab1 Report

Group 2 - 林珮玉 林哲緯 林玠志 薛博文 蔡孟宗

I. Objectives

1. 超音波 sensor 範例
2. 陀螺儀 sensor 範例
3. 顏色 sensor 範例
4. 實驗練習

II. Exploration

1. 將事先組好的輪型機器人，加上超音波 sensor，做出可避障的機器人。

Code

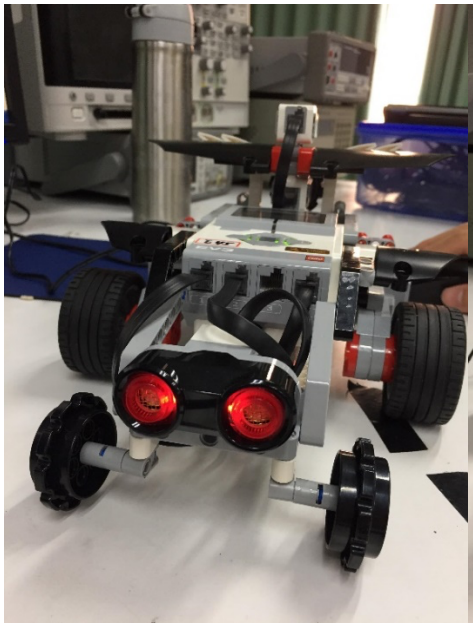
```
#pragma config(Sensor, S4, US, sensorEV3_Ultrasonic)
#pragma config(Motor, motorB, MotorB, tmotorEV3_Large,PIDControl,
encoder)
//*!!Code automatically generated by 'ROBOTC' configuration
wizard !!*//
void TurnLeft(int angle);
void TurnRight(int angle);
void set2MotorSpeed(int speed);
#define Near 20 //cm
#define Motor1 motorB
#define Motor2 motorC

task main() {
   while(true){
      if(getTouchValue(S1) == 1)
         break;
      delay(50);
   }

   while (true) {
      if (getUSDistance(S4) > Near)
         set2MotorSpeed(30);
      else
         break;
   }
```

```
    TurnRight(45);
    set2MotorSpeed(30);
    delay(2100);
    TurnLeft(45);
    set2MotorSpeed(30);
    delay(1500);
    set2MotorSpeed(0);
}

void TurnRight(int angle){
    resetGyro(S2);
    repeatUntil(getGyroDegrees(S2) > angle){
        setMotorSpeed(Motor2, -50);
        setMotorSpeed(Motor1, 50);
    }
    set2MotorSpeed(0);
    return;
}

void TurnLeft(int angle){
        resetGyro(S2);
    repeatUntil(getGyroDegrees(S2) < -1*angle){
        setMotorSpeed(Motor2, 50);
        setMotorSpeed(Motor1, -50);
    }
    set2MotorSpeed(0);
    return;
}

void set2MotorSpeed(int speed){
    setMotorSpeed(Motor1,speed);
    setMotorSpeed(Motor2,speed);
}
```

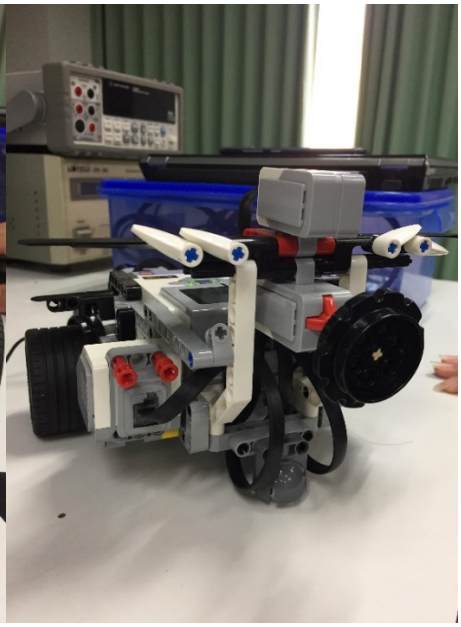
程式中先寫好 TurnRight(int angle)、TurnLeft(int angle)以及 set2MotorSpeed(int speed)三個函式，再回到主函式中，利用 getTouchValue(S1) 控制機器人開始運作。透過 Sensor 的 getUSDistance(S4)偵測距離，搭配 if-else 判斷目前位置與屏障的距離是否超過預設 cm。如果是的話則就代表要執行繞過避障。首先先做

TurnRight(45)向右轉 45 度，並設定 set2MotorSpeed(30) 以 30%的 power 前進，走了 2.1s 後繞過避障，再 TurnLeft(45) 把角度轉正，一樣以 30%的 power 前進 2.1s，確定完整繞過之後，就把速度設為 0 停止。

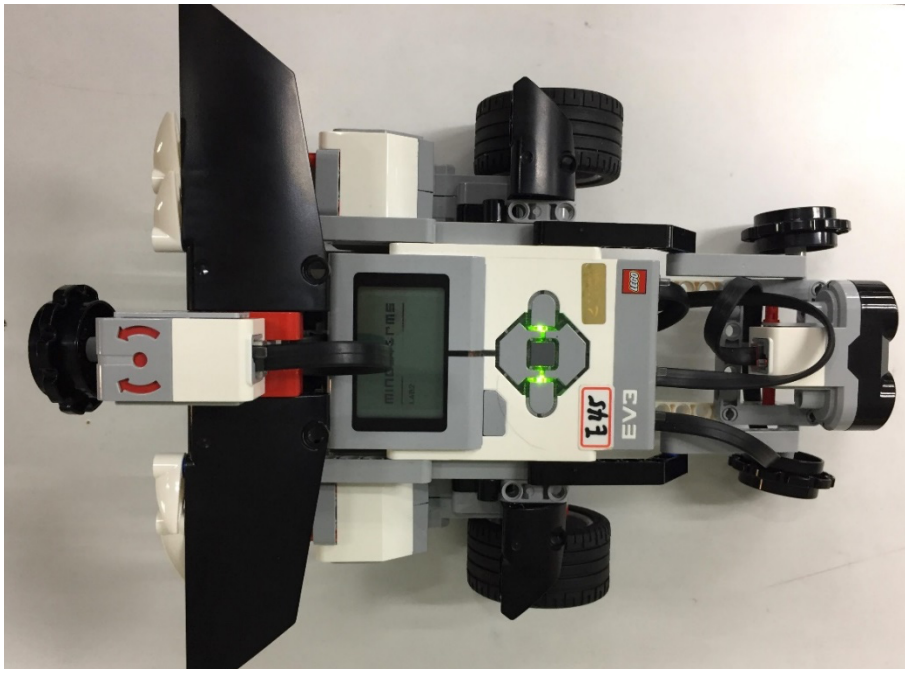
Result



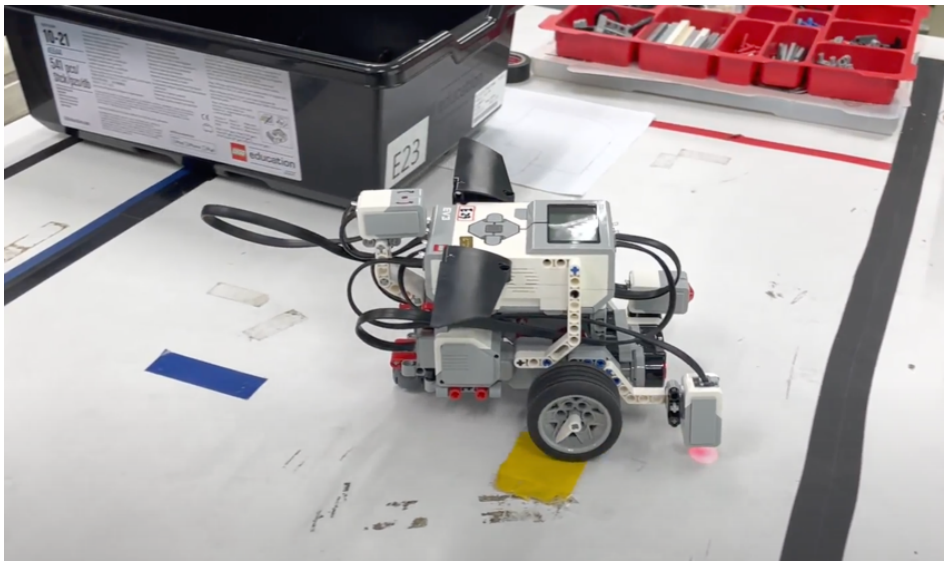
（a）超音波顯示器



（b）按鈕控制



（c）模型車的俯視圖



（d）繞過屏障的影片：Lab2-1 影片連結

2. 偵測到紅色停止，綠色直走，藍色左轉 90 度，黃色右轉 90 度

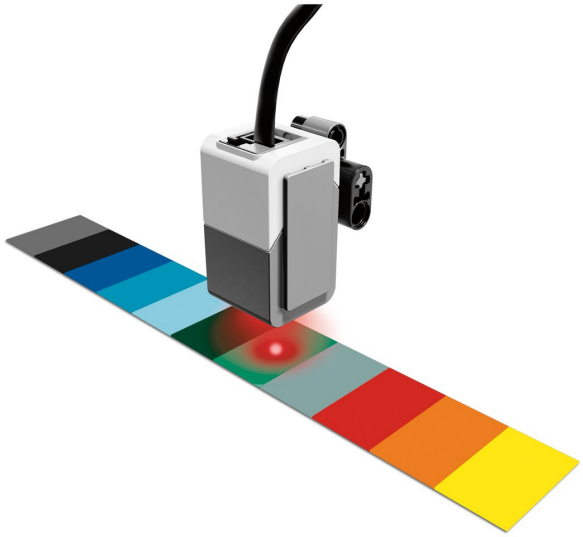
Code

```
#pragma config(Sensor, S4, US, sensorEV3_Ultrasonic)
#pragma config(Motor, motorB, MotorB, tmotorEV3_Large,PIDControl,
encoder)
//*!!Code automatically generated by 'ROBOTC' configuration
wizard !!*//
void TurnLeft(int angle);
void TurnRight(int angle);
void set2MotorSpeed(int speed);
#define Near 20 //cm
#define Motor1 motorB
#define Motor2 motorC

task main() {
   while(true){
      if(getTouchValue(S1) == 1)
         break;
      delay(50);
   }

   repeat(forever){
      if(getColorName(S3) == colorRed){
         set2MotorSpeed(0);
         delay(1000);
      }else if(getColorName(S3) == colorGreen){
         set2MotorSpeed(30);
         delay(1000);
      }else if(getColorName(S3) == colorBlue){
         TurnLeft(90);
         delay(1000);
      }else if(getColorName(S3) == colorYellow){
         TurnRight(90);
         delay(1000);
      }
   }
}
```

```
void TurnRight(int angle){
    resetGyro(S2);
    repeatUntil(getGyroDegrees(S2) > angle){
        setMotorSpeed(Motor2, -50);
        setMotorSpeed(Motor1, 50);
    }
    set2MotorSpeed(0);
    return;
}

void TurnLeft(int angle){
    resetGyro(S2);
    repeatUntil(getGyroDegrees(S2) < -1*angle){
        setMotorSpeed(Motor2, 50);
        setMotorSpeed(Motor1, -50);
    }
    set2MotorSpeed(0);
    return;
}

void set2MotorSpeed(int speed){
    setMotorSpeed(Motor1,speed);
    setMotorSpeed(Motor2,speed);
}
```

如同上題一樣先寫好 TurnRight(int angle)、TurnLeft(int angle)以及 set2MotorSpeed(int speed)三個函式。回到主函式，利用顏色 Sensor 的 getColorName(S3)尋找物體顏色並存成字串，透過判斷式判斷物體顏色是 colorRed、colorGreen、colorBlue 還是 colorYellow，若讀到紅色物體就暫停，若讀到綠色物體就繼續前進，若讀到藍色物體則向左轉 90 度，若讀到黃色物體則向右轉 90 度。用 repeat(forever){ } 方程式讓 Sensor 執行重複偵測。最後全部執行一輪之後就將速度設為 0 停止。

Result



（忘記拍照紀錄了，附上網路示意圖）

3. 使機器人沿著桌面上貼的黑線走

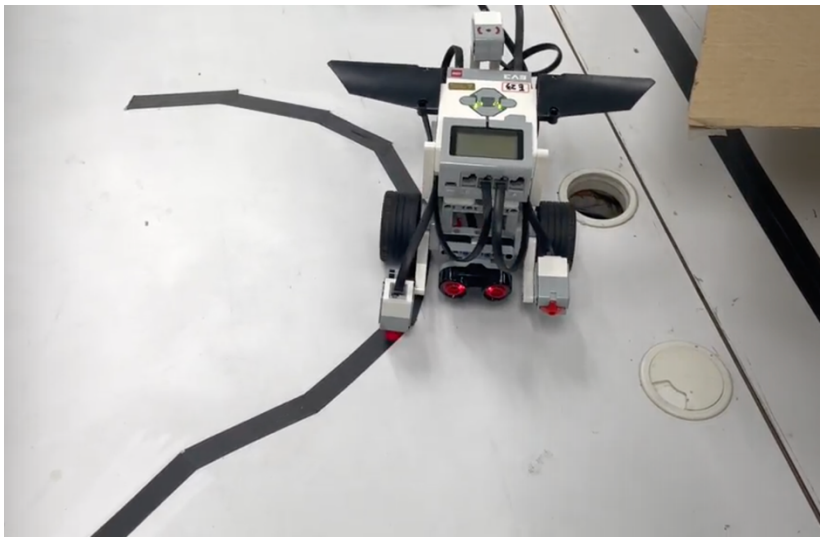
Code

```
#pragma config(Sensor, S4, US, sensorEV3_Ultrasonic)
#pragma config(Motor, motorB, MotorB, tmotorEV3_Large,PIDControl,
encoder)
//*!!Code automatically generated by 'ROBOTC' configuration
wizard !!*//
void TurnLeft(int angle);
void TurnRight(int angle);
void set2MotorSpeed(int speed);
#define Near 20 //cm
#define Motor1 motorB
#define Motor2 motorC

task main() {
   while(true){
      if(getTouchValue(S1) == 1)
         break;
      delay(50);
   }
   resetGyro(S2);
   while(true){
      if(getColorReflected(S3)>20){
```

```
            if( getGyroRate(S2)>0)
                TurnLeft(getColorReflected(S3)*0.2);
            else
                TurnRight(getColorReflected(S3)*0.2);
        }
        else
            set2MotorSpeed(10);
    }
}

void TurnRight(int angle){
    resetGyro(S2);
    repeatUntil(getGyroDegrees(S2) > angle){
        displayTextLine(2,"%d",getGyroDegrees(S2));

        setMotorSpeed(Motor2, -10);
        setMotorSpeed(Motor1, 10);
    }
    set2MotorSpeed(0);
    return;
}

void TurnLeft(int angle){
        resetGyro(S2);
    repeatUntil(getGyroDegrees(S2) < -1 * angle){
        displayTextLine(2,"%d",getGyroDegrees(S2));

        setMotorSpeed(Motor2, 10);
        setMotorSpeed(Motor1, -10);
    }
    set2MotorSpeed(0);
    return;
}

void set2MotorSpeed(int speed){
    setMotorSpeed(Motor1,speed);
    setMotorSpeed(Motor2,speed);
}
```

如同上題一樣先寫好 TurnRight(int angle)、TurnLeft(int angle)以及 set2MotorSpeed(int speed)三個函式。再回到主函式，利用顏色 Sensor 的 getColorReflected(S3)讀取反射率，如果反射率大於設定值，則利用陀螺儀 getGyroRate(S2)判斷目前是向左轉還是向右轉，如果目前是向左轉，就執行 TurnLeft()直到方向向右。值得注意的是 TurnLeft()、TurnLeft()裡面放的變數是反射率*0.2，也就是說向左轉或者向右轉會隨著讀取到的反射值有所不同。這就是為什麼我們可以做到循跡轉彎的原因。

Result



<進階> 循跡轉彎的影片：Lab2-3 影片連結

III. Conclusion

1. 林珮玉

這次實驗使用了超音波測距儀與按鈕以及陀螺儀模組，樂高提供的超音波測距儀與市售常見的 HC-05 模組看起來就相當的不一樣，裝上超音波測距之後自走車便可以判斷前方的障礙物，並且以轉彎的方式避免碰撞，特別有趣。這次有三個實驗，其實到最後一個 Lab3 大家有一點小累了，不過我的夥伴好厲害，幾乎一個人 Cover 全部，意外地效果很棒，甚至做到了期末 project 要展示的循跡轉彎功能！第一次看著機器人慢慢轉彎，覺得好可愛，希望期末我們可以做的更精準且順暢！

2. 林哲緯

這次實驗用到了超音波測距和顏色辨識的 sensor，讓車子有辦法做到障礙物檢測，並控制馬達讓車子有辦法避開障礙物，或者是讓車子沿著黑線走動。這次主要負責把車子接起來而已，過程十分的舒壓，組員們十分厲害的實現控制車子的邏輯，最後也有實際跑出來，覺得更是厲害！

3. 林玠志

這次實驗是用樂高租出一個輪型機器人並且加裝各種 sensor 去讓它做出各種決策來完成 lab。lab1 是讓機器人可以進行避障，lab2 是用偵測到的顏色來控制機器人的動作，lab3 則是讓機器人可以循跡。其實這些 lab 的程式內容都必須考慮到很細節的東西，必須對 function 內部有一些了解才能避開一些 bug。這次實驗有賴於隊友的幫忙才能順利完成，希望下次可以做更有趣的東西。

4. 薛博文

這次的實驗有三個部分，分別是避障、顏色判斷、延著黑線走，我們首先創了 turnLeft(), turnRight() 兩個 function 方便後續轉向使用，按鈕的部分用一個 while 迴圈困住狀態，直到按鈕被按下，後續三個實驗的做法分別是，判斷距離夠進就開始轉彎、判斷顏色做出相應的運動、判斷 refleciton 的值是否夠大，第三個實驗孟宗用了蠻神奇的幾行程式碼就實踐出來，雖然還沒辦法讓車子應付各種狀況，但效果出奇地好，可以達成直線前進，甚至可以過一些彎，做完這個覺得很期待之後的期末 project，應該會很好玩～

5. 蔡孟宗

這次主要有三個實驗，分別是避障，顏色識別以及沿著黑色直線移動，其中最值得一提的是讓機器人沿黑線直線移動的實驗，這個實驗中我應用了控制工程的負回授的概念，根據機器人接收到的反射值決定機器人

偏移角度，反射值越大(也就是偏離軌道越多)則會偏轉越多角度，而偏轉的方向和陀螺儀指向相反，由於每次偏轉前都會將陀螺儀歸零，因此陀螺儀實際上紀錄的是前一次機器人偏轉的方向，若前一次往右偏轉(讀值為正)，則下一次需要偏轉時機器人會往左偏轉，反之亦然，這樣做可以確保機器人不會在錯誤的方向越偏越多，假設機器人是從黑線左側偏離，這時他應該往右邊偏轉才能回到正軌，若此時機器人往左偏轉，會接受到更大的反射值，因此下一次偏轉會是更大角度且反向的偏轉，如此便能修正偏轉錯誤的角度且往正確的方向開始偏轉。

理論上是這樣，但是我也不確定效果究竟如何，而依照這樣的思路打出來的 code 效果比我想像中好很多，甚至可以做到循跡轉彎，所以我印象滿深刻的，後來觀察實驗結果，我想出了一個可以改進的地方，就是負回授的權重，目前負回授的權重是常數，我認為可以把它變成常數+bias，隨著機器人找不到軌跡的次數 bias 會越變越大，一旦找到軌道便把 bias 歸零。這次的實驗很有趣，期待之後的實驗。