工科專論期末報告 Group 6

1. 分工表:

莊博翰：撰寫控制程式

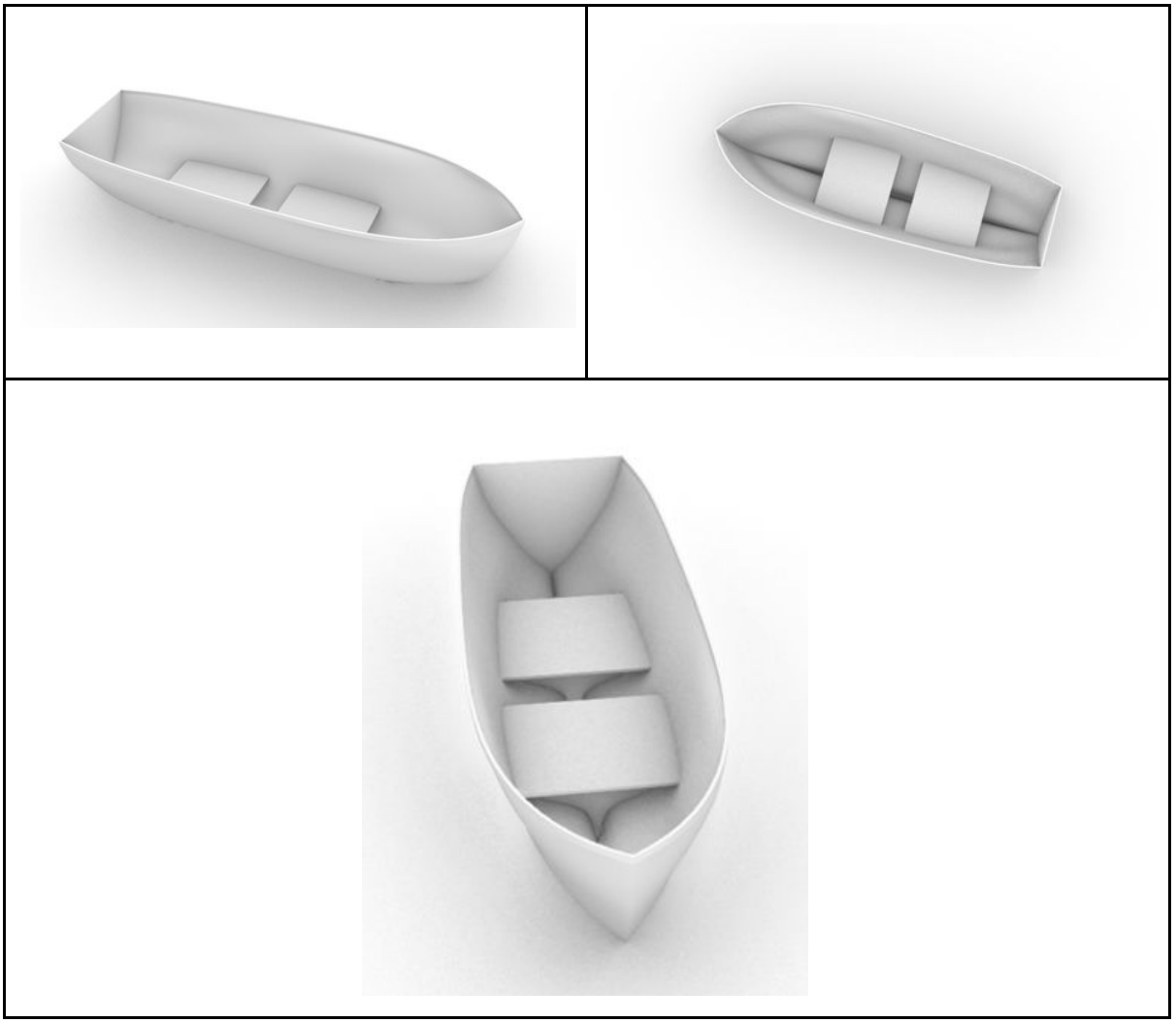
張在然：測試自走船 & 元件、購置設備

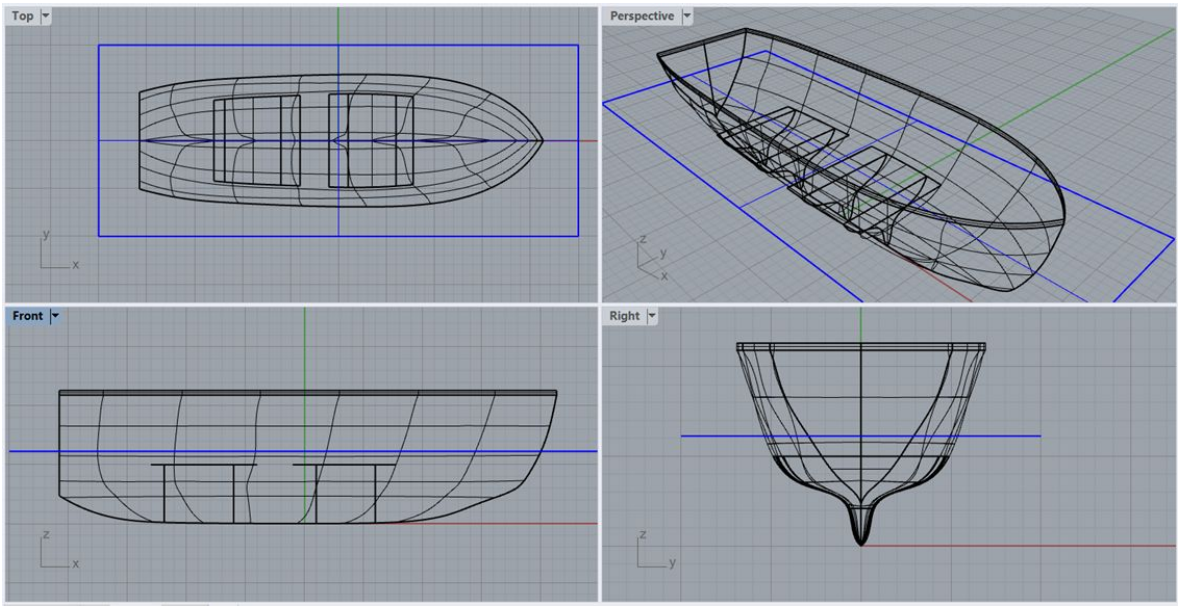
陳銘杰：電路組裝、焊接

陳心怡：船體、羅槳設計與組裝

2. 船型設計:

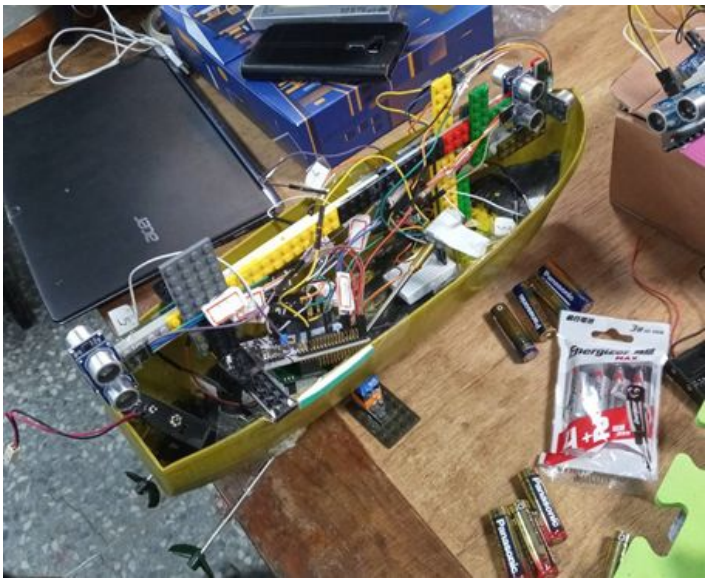
(1) 船模設計圖

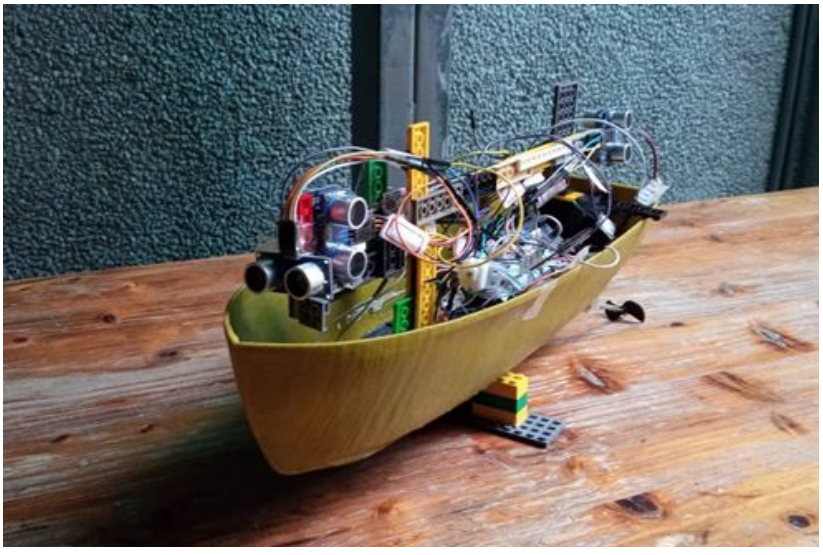




藍色部分為預計水線面

(2) 實際船模照片


Panasonic
Panasonic



(3) 船模設計考量

採用底部細窄、上部寬的設計，在四公分左右船型拉寬以求有更高的穩度與預備浮力，底部考量方便船模直行而設計底部船舵以穩流，底部細長區原先設計可以保護螺槳較不易受碰撞，但後來未調整螺槳距而將孔洞往上移

測試後反省：

穩度比想像中差了一些，應將船型拉寬

最初螺距太短，且底部無法放平螺槳，應考量螺槳長度重新調整

(4) 水線測量

吃水(T)	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
水線面積(Aw)	0	13.2922	20.863	27.1171	36.6579	59.8933	125.0703	220.8804	270.0949	298.8272	319.9166
排水量	0	3.32305	11.86185	23.85688	39.80063	63.93843	110.1793	196.667	319.4108	461.6414	616.3273
moment of displacement	0	1.661525	8.5388	23.92346	52.42133	108.1836	239.4196	526.4927	989.858	1596.133	2332.21
浮心高(KB)	0	0.5	0.719854	1.002791	1.317098	1.691996	2.173	2.677077	3.099012	3.457518	3.784044
吃水(T)	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	10.5
水線面積(Aw)	337.7018	355.487	373.4548	391.3899	408.8094	425.2442	440.2687	453.9247	466.2826	479.472	492.6613
排水量	780.7319	954.0291	1136.265	1327.476	1527.526	1736.039	1952.417	2175.966	2406.017	2642.456	2885.489
moment of displacement	3196.446	4194.016	5334.111	6625.907	8077.357	9694.363	11480.42	13437.32	15566.08	17872.18	20364.09
浮心高(KB)	4.094165	4.396109	4.694427	4.991358	5.28787	5.584185	5.880107	6.175339	6.469644	6.763472	7.057414

總船高　　11.301 cm
水線高　　6.13 cm
水線面積　360.14 cm2
排水量　　1000.3 cm3

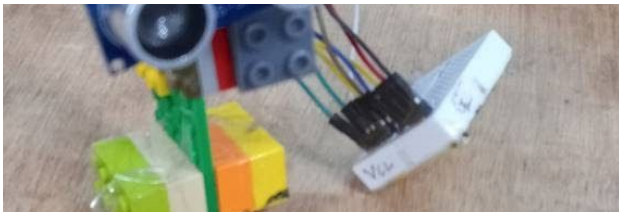
(5) principal dimension

吃水 (T): 6.13 cm
水線面積 (AW): 360.14 cm2
排水量 (▽): 1000.3 cm3
力矩 (M): 4481.7 cm4
浮心高 (KB): 4.4803 cm
總船高 (D): 11.301 cm
乾弦 (f): 5.171 cm
模寬 (B): 13.388 cm
總船長 (LOA): 42.094 cm
水線長 (LWL): 40.463 cm
水線寬 (BWL): 10.707 cm
舯面積: 89.433 cm
舯吃水面積 (AM): 31.113 cm

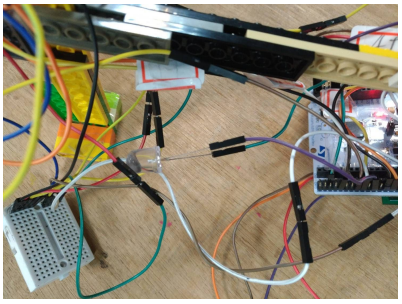
3 電路設計:

(0).外觀

我們的船安裝了五個超聲波感測器，船頭一個，兩側各二



並將VCC 和ground 用麵包板接在一起



安裝額外的LED 用來debug

(1)

Pin 腳對應表

超音波前:PB1 trigger， PC1 echo

超音波左一: PB2 trigger ，PB13 echo

超音波左二: PA10 trigger. PA9 echo

超音波右一: PA0 trigger ,PA1 echo

超音波右二 PC2 trigger ,PC3 echo

LED 燈 高電位 :PB11， 低:ground

還有電供板用了 timer 3:PC8 ,timer4:PC9 來打PWM

並且可以藉由調整 PC9， PB8， set,reset 來反轉馬達

基本以上是使用的pin腳

超聲波測距方式

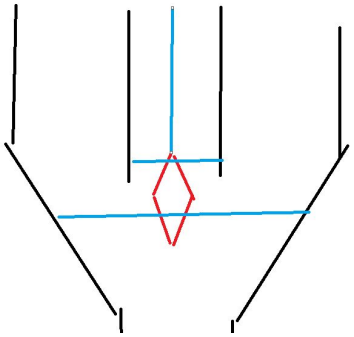
在我們的程式中是以一個while 迴圈的執行次數來量測距離所以不會用到額外的timer

(2)

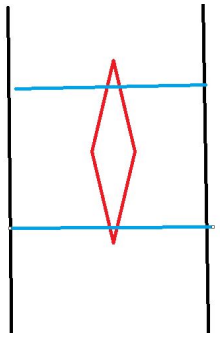
基本上我們控制分三段:

(a1)一開始的直行:

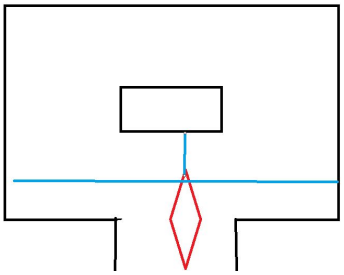
第一個需要辨識的點是進入直行賽道瞬間左右後大，左右前小，正面遠



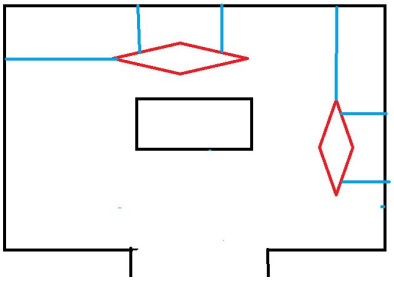
(a2) 接下來直線修正，維持船的左右斜率為零



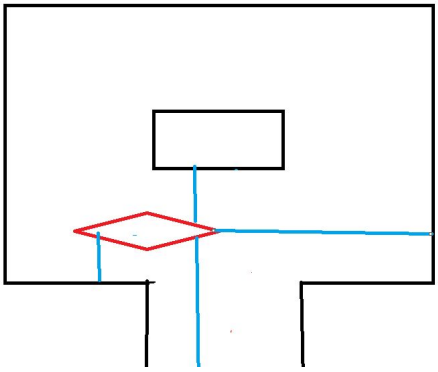
(a3) 直到發現進入迴轉區(前方近，左右前稍遠)，先右轉



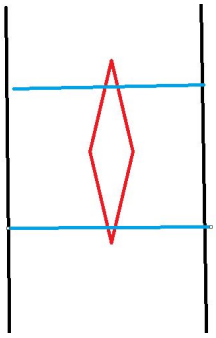
(b1)開始逆時鐘繞圈，按照正面與右方量測距離修正



(b2) 直到發現出口(右前極遠，右後近，(左前近|| 正面指定值))，煞車右轉



(c 1) 直線修正，回家



(3)

code結構

code 分量稍大，另外放，這裡稍為介紹幾個主要function

(a) 用來量測距離的function，分成五個(前，右左前，右左後)

分成五個的原因是有時不同感應器要個別校準，這裡舉例前方

```
int d_h(){//12
        int t2=0;
        int t1=0;
        int tmp=1;
        int tmp2=1;
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, GPIO_PIN_SET);
                HAL_Delay(10);
                HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
                t1=0;
                t2=0;
                while (!HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_1)) {
                    t1++;
                    if (t1 > 20000) {
                        break;
                        t1 = 0;
                    }
                }
                while (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_1)) {
                    t2++;
                    if (t2 > 30000) {
                        break;
                    }
                }
                return t2;
    }
```

(b)用來調整馬達轉速的function

如果設負值就會反轉

```
void setspeed(int l,int r){
    //l=0;r=0;
    if(l<0){
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET);
        l=20+l;
    }else{
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_RESET);
    }
    sConfigOC.Pulse = l;
    HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_3);
    HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3); //right
    if(r<0){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, GPIO_PIN_SET);
        r=20+r;
    }else{
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, GPIO_PIN_RESET);
    }
    sConfigOC.Pulse = r;
    HAL_TIM_PWM_ConfigChannel(&htim4, &sConfigOC, TIM_CHANNEL_4);
    HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_4);
}
```

(c) 簡單的輸出function

```
void printnumber(int num) {
    char buf[10] = { 0 };
    int t2stay=num;
    if (t2stay < 0) {
        t2stay = -t2stay;
        buf[7] = '\n';
        buf[6] = '\r';
        buf[5] = t2stay % 10 + '0';
        buf[4] = t2stay / 10 % 10 + '0';
        buf[3] = t2stay / 100 % 10 + '0';
        buf[2] = t2stay / 1000 % 10 + '0';
        buf[1] = t2stay / 10000 % 10 + '0';
        buf[0] = '-';
        HAL_UART_Transmit(&huart2, (uint8_t*) buf, strlen(buf), 0xFFFF);
    } else {

        buf[6] = '\n';
        buf[5] = '\r';
        buf[4] = t2stay % 10 + '0';
        buf[3] = t2stay / 10 % 10 + '0';
        buf[2] = t2stay / 100 % 10 + '0';
        buf[1] = t2stay / 1000 % 10 + '0';
        buf[0] = t2stay / 10000 % 10 + '0';
        HAL_UART_Transmit(&huart2, (uint8_t*) buf, strlen(buf), 0xFFFF);
    }
}
```

以上三個是比較有實際用到的function，接下來是控制方面

這是(a) part 進迴轉區前的控制

```
    if(ncase==-3){
        setspeed(9,14);
        if((dr2>6000||dl2>6000)&&(dr1<4000||dl1<4000)){
            ncase=-1;
            HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET);
            setspeed(9,14);
            HAL_Delay(1200);
            setspeed(-15,-15);
            HAL_Delay(750);

        }
        HAL_Delay(100);
    }
    else if(ncase==-1){

        if(dh<10000){
            setspeed(7,6);
        }else if(dr1-dr2>1000&&(dr1>200&&dr1<10000)&&(dr2>200&&dr2<10000)){
            setspeed(10,0);
        }else if(dl1-dl2>1000&&(dl1>200&&dl1<10000)&&(dl2>200&&dl2<10000)){
            setspeed(0,10);
        }else if(dl1<dr1){
            setspeed(10,6);
        }
        else{
            setspeed(13,10);
        }
        if((dr2>7000||dr1>7000)&&(dr1>7000||dl1>7000)){
            HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
            ncase=0;
            setspeed(15,-18);
            HAL_Delay(3000);
            setspeed(10,10);
            HAL_Delay(800);
            count1=0;
        }else if(dh<7000&&dr1>5000&&dr2>5000){
            HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
            ncase=0;
            setspeed(15,-18);
            HAL_Delay(3000);
            setspeed(10,10);
            HAL_Delay(800);
            count1=0;
        }
        HAL_Delay(100);
```

這些是在迴轉區中的控制

```
else if (dr1 > 16000 && dh < 20000&&(dr2<7500||dr2>19000)&&count1>65){
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
    ncase=2;
    setspeed(-14,-19);
    HAL_Delay(2500);
    setspeed(19,-19);
    HAL_Delay(3000);
    setspeed(10,13);
    HAL_Delay(3000);
    ncase=1;
 }else{
    if(dh>200&&dh<1500){
        setspeed(-17,0);
        HAL_Delay(300);
        count1+=3;
        backtmp=1;
    }

     else if((dh>200&&dh<3000) || dr2-dr1>2400){
        ncase=12;
        setspeed(-17,19);
        HAL_Delay(100);
    }else if(dr2-dr1>1000 &&dr1>200&&dr2>200){
        ncase=13;
        setspeed(-15,19);
        HAL_Delay(100);
    }else{
        ncase=14;
        setspeed(6,19);
        HAL_Delay(100);
    }
    count1+=1;
 }
```

這是離開迴轉區後的控制

```
} else if (ncase == 1) {
    if (dl1 - dl2 > 600 && (dl1 > 200 && dl1 < 10000)
            && (dl2 > 200 && dl2 < 10000)) {
        setspeed(0, 10);
    } else if (dr1 - dr2 > 600 && (dr1 > 200 && dr1 < 10000)
            && (dr2 > 200 && dr2 < 10000)) {
        setspeed(8, 0);
    } else {
        setspeed(8, 12);
    }
    HAL_Delay(100);
```