

工海專論期末報告

第三組

一、分工表

B06505007 詹心憲：船體設計、螺槳組裝

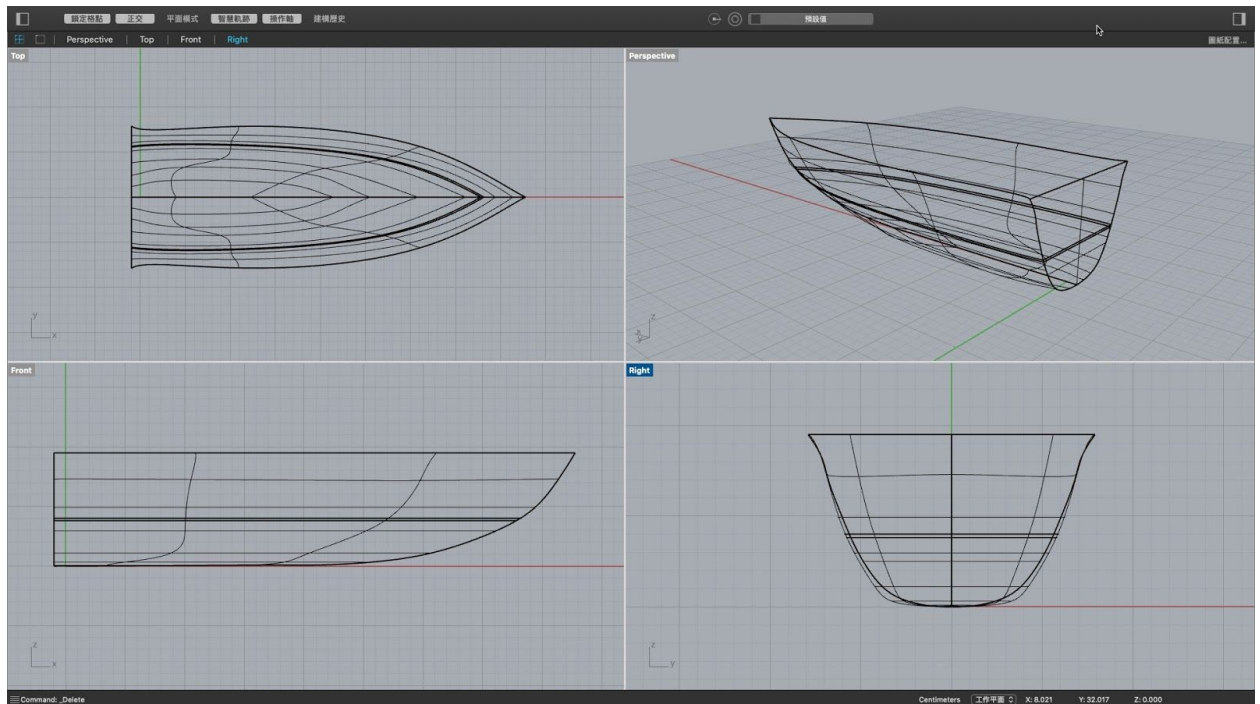
B06602035 李晴妍：電路設計、子母板及電路板接線組裝

B06505032 楊千瑩：電路設計、子母板及電路板接線組裝

B06505054 石乃耘：電路設計、子母板及電路板接線組裝

二、船體設計

Rhino 設計圖



設計理念

船型實際照片



三、電路設計

- 超聲波（三個：左方、前方、右方）

timer	connection	prescaler	period
TIM1 CH1	trigger	7	6000
TIM2	Left echo	7	65536
TIM3	Center echo	7	65536

TIM15	Right echo	7	65536
-------	------------	---	-------

For trigger, we start the timer as PWM

```
HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);
```

- 馬達

timer	connection	prescaler	period
TIM8 CH4	左馬達正轉	7	100
TIM8 CH3	左馬達反轉	7	100
TIM4 CH3	右馬達正轉	7	100
TIM4 CH4	右馬達反轉	7	100

初始化時，運用timer控制PWM輸出

```
HAL_TIM_PWM_Start(&htim8, TIM_CHANNEL4);
```

```
HAL_TIM_PWM_Start(&htim8, TIM_CHANNEL3);
```

```
HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL3);
```

```
HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL4);
```

控制馬達轉速

```
TIM8->CCR4 = 100;
```

```
TIM8->CCR3 = 0;
```

```
TIM4->CCR3 = 100;
```

```
TIM4->CCR = 0;
```

- 超聲波測距

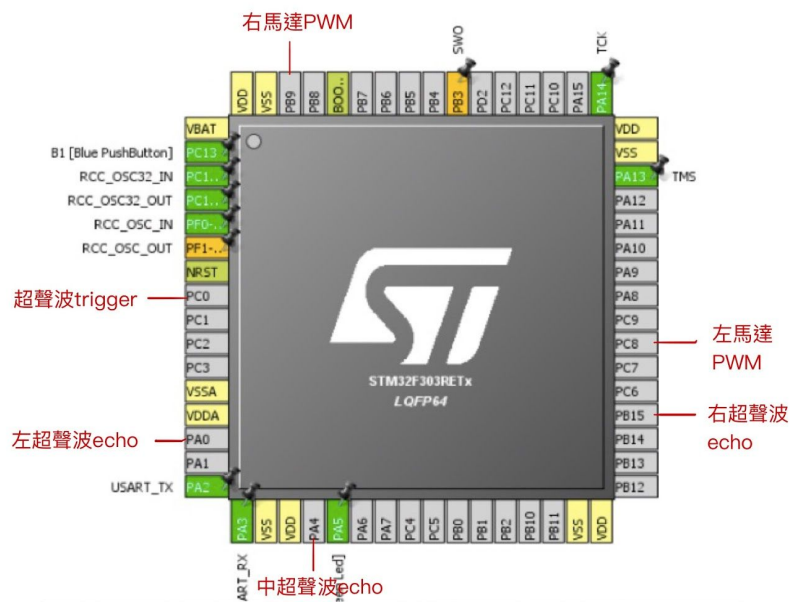
設置三個超聲波以測試距離，分為前方、左方、右方。

其中trig均為同一角位，而echo分為三個角位。

1. 給trig輸入一個長為20us的高電位方波
2. 輸入方波後，會自動發射8個40KHz的聲波，此時echo端電位會由0升至1，並啟動計時器counter計時。
3. 當超聲波返回被接收到時，echo端電位會由1變為0，此時停止計時，記下counter的時間（由發射到返回的總時長）
4. 根據聲音在空氣中的速度為344米/秒，計算出距離。

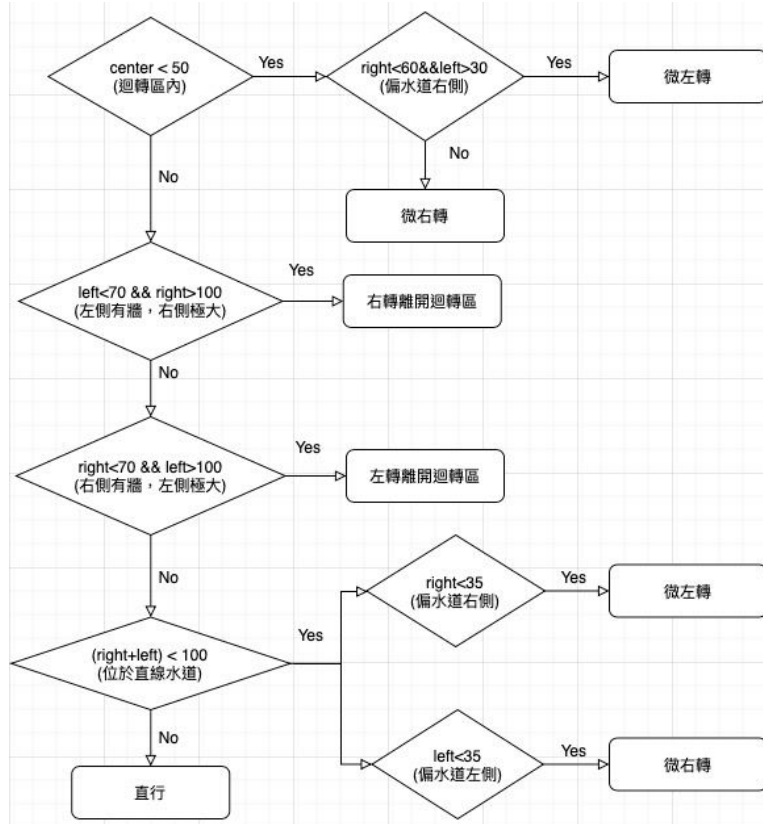
將超聲波的測距，記錄下來，利用距離判斷方位控制自走車

- 電路 schematic 圖



- 演算法 方位判斷

利用PWM控制螺槳轉速，以達到直線校正，或是左右轉



- 程式碼截圖

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef* htim){
    if(htim->Instance==TIM2){
        if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0)==1){
            __HAL_TIM_SET_COUNTER(&htim2,0);

        }else{
            cnt=__HAL_TIM_GET_COUNTER(&htim2);
            left=cnt/(double)58;
            integer=(int)left;
            point=(int)((left-integer)*100);
            char tosend[20]={0};
            sprintf(tosend,"%d.%02d\r\n",integer,point);
            HAL_UART_Transmit(&huart2,"1:",2,0xFFFF);
            HAL_UART_Transmit(&huart2,tosend,sizeof(tosend),0xffff);
        }
    }

    if(htim->Instance==TIM3){
        if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_4)==1){
            __HAL_TIM_SET_COUNTER(&htim3,0);

        }else{
            cnt1=__HAL_TIM_GET_COUNTER(&htim3);
            center=cnt1/(double)58;
            integer1=(int)center;
            point1=(int)((center-integer1)*100);
            char tosend1[20]={0};
            sprintf(tosend1,"%d.%02d\r\n",integer1,point1);
            HAL_UART_Transmit(&huart2,"2:",2,0xFFFF);
            HAL_UART_Transmit(&huart2,tosend1,sizeof(tosend1),0xffff);
        }
    }

    if(htim->Instance==TIM15){
        if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_15)==1){
            __HAL_TIM_SET_COUNTER(&htim15,0);

        }else{
            cnt2=__HAL_TIM_GET_COUNTER(&htim15);
            right=cnt2/(double)58;
            integer2=(int)right;
            point2=(int)((right-integer2)*100);
            char tosend2[20]={0};
            sprintf(tosend2,"%d.%02d\r\n",integer2,point2);
            HAL_UART_Transmit(&huart2,"3:",2,0xFFFF);
            HAL_UART_Transmit(&huart2,tosend2,sizeof(tosend2),0xffff);
        }
    }
}
```

超聲波測距左方距離

超聲波測距前方距離

超聲波測距右方距離

```
//=====Enter=====
```

```
if(center<50){  
    //left  
    if(right<60 && left>30){  
        TIM8->CCR4=0; //left reverse  
        TIM8->CCR3=81; // 81  
        TIM4->CCR3=81; // 81  
        TIM4->CCR4=0;  
    }  
    //right  
    else{  
        TIM8->CCR4=90; //90  
        TIM8->CCR3=0;  
        TIM4->CCR3=0;  
        TIM4->CCR4=87; // 87 right reverse  
    }  
}
```

迴轉區內

```
//=====Exit =====
```

```
else if(left<70 && right>100){  
    //right  
    TIM8->CCR4=100;  
    TIM8->CCR3=0;  
    TIM4->CCR3=0;  
    TIM4->CCR4=100;  
}
```

右轉離開迴轉區

```
else if(right<70 && left>100){  
    //left  
    TIM8->CCR4=0;  
    TIM8->CCR3=100;  
    TIM4->CCR3=100;  
    TIM4->CCR4=0;  
}
```

左轉離開迴轉區

```
else{
```

```
    if((right + left) < 100){  
        if(right<35){ // 30  
            TIM8->CCR4=71; //left  
            TIM8->CCR3=0;  
            TIM4->CCR3=72;//right  
            TIM4->CCR4=0;
```

```
    }
```

偏左偏右校正

```
        if(left<65){ // 50  
            TIM8->CCR4=72; // 75  
            TIM8->CCR3=0;  
            TIM4->CCR3=71;  
            TIM4->CCR4=0;
```

```
        }
```

```
    }
```

```
//straight
```

```
TIM8->CCR4=73; // 78
```

```
TIM8->CCR3=0;
```

```
TIM4->CCR3=75; //80
```

```
TIM4->CCR4=0;
```

直行

```
}
```

```
}
```

```

int main(void)

{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_TIM1_Init();
    MX_TIM15_Init();
    MX_USART2_UART_Init();
    MX_TIM2_Init();
    MX_TIM3_Init();
    MX_TIM4_Init();
    MX_TIM8_Init();

    /* USER CODE BEGIN 2 */
    HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);
    HAL_TIM_IC_Start_IT(&htim2,TIM_CHANNEL_1);
    HAL_TIM_IC_Start_IT(&htim3,TIM_CHANNEL_2);
    HAL_TIM_IC_Start_IT(&htim15,TIM_CHANNEL_2);

    HAL_TIM_PWM_Start(&htim8,TIM_CHANNEL_4);
    // TIM8->CCR4=65;
    HAL_TIM_PWM_Start(&htim8,TIM_CHANNEL_3);
    // TIM8->CCR3=0;
    HAL_TIM_PWM_Start(&htim4,TIM_CHANNEL_3);
    // TIM4->CCR3=60;
    HAL_TIM_PWM_Start(&htim4,TIM_CHANNEL_4);
    // TIM4->CCR4=0;
    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
    }
    /* USER CODE END 3 */
}

```

角位設定

馬達timer PWM設定