

<조선해양공학계획>

Term Project 최종보고서(요약 보고서)  
-VLCC를 기준선형으로 초음파센서를 이용한 모형선의  
자동제어-



7조 Renovatia

김두용   김영현   노재욱   심훈섭   이주현

## <목 차>

1. 선체부 설계 .....	1
1.1. 기초설계 .....	1
1.1.1. 기준선의 선정 .....	1
1.1.2. 주요치수의 결정 .....	2
1.1.3. 경하중량의 계산 .....	3
1.1.4. 개략배치도 및 구획배치도 .....	3
1.2. 상세설계 .....	5
1.2.1. 선형 설계 과정 .....	5
1.2.2. 구획 설계 과정 .....	7
1.2.3. 세부 경하중량의 추정 .....	10
2. 선체부 제작 .....	14
3. 제어부 설계 .....	17
3.1. 하드웨어 설계 .....	17
3.1.1. CONTROLLER 설계(1차) .....	17
3.1.2. CONTROLLER 설계(2차) .....	18
3.2. 소프트웨어 설계(알고리즘) .....	19
4. 제어부 제작 .....	20
4.1. 하드웨어 .....	20
4.1.1. Controller 제작(1차) .....	20
4.1.2. Controller 제작(2차) .....	20

4.2. 소프트웨어.....	21
<b>5. Contest.....</b>	<b>22</b>
5. 완성된 모형선 .....	22
5.2. 1차 Contest .....	23
5.2.1. 1차 컨테스트 준비 .....	23
5.2.2. 1차 컨테스트 결과 및 향후 과제 .....	25
5.3. 2차 Contest .....	26
5.3.1. 2차 Contest 준비과정 .....	26
5.3.2. 2차 Contest 결과 및 향후 과제 .....	31
<b>6. 역할분담 및 세부일정 .....</b>	<b>32</b>
6.1. 역할분담.....	32
6.2.세부일정 .....	33

## 〈그림 목차〉

〈그림 1〉 개략배치도(Proposal) .....	2
〈그림 2〉 중앙평행부 면적 근사 .....	3
〈그림 3〉 모형선의 G/A.....	4
〈그림 4〉 모형선의 구획배치도.....	4
〈그림 5〉 fairing 전 선형 곡면 모델, cp curve .....	5
〈그림 6〉 fairing 후 선형 곡면 모델, cp curve .....	6
〈그림 7〉 fairing 후 cp curve 상세 정보 .....	6
〈그림 8〉 일차 선형 주요치수.....	7
〈그림 9〉 최종 구획 모델 생성 과정.....	7
〈그림 10〉 전폭 선형 이진 파일의 모습.....	8
〈그림 11〉 선형 이진 파일과 구획 분할 모델을 합성 .....	8
〈그림 12〉 Proposal 당시의 구획 배치도.....	9
〈그림 13〉 Ezship을 이용해 나타낸 구획의 모습 .....	9
〈그림 14〉 부재 단면도.....	12
〈그림 15〉 단계별 선체 제작 사진 .....	16
〈그림 19〉 Rudder 단면.....	17
〈그림 20〉 controller 회로도.....	18
〈그림 50〉 2차 controller 회로도.....	19
〈그림 21〉 주어진 경로를 주행하는 알고리즘 .....	19
〈그림 62〉 Controller 의 모습.....	20
〈그림 65〉 최종 controller 앞면 .....	21

<그림 71> 완성된 모형선의 개략배치도.....	22
<그림 75> 자동제어기관a.....	23
<그림 76> 거리에 따른 센서값 .....	24
<그림 79> 최초의 제어부 연결 .....	25
<그림 80> SRF04 작동 원리 .....	28
<그림 81> 최종 제어부 연결.....	30

## <표 목차>

<표 1> 구획배치 .....	4
<표 2> 각 구획별 배치 품목 .....	10
<표 3> 선각 중량 .....	11
<표 4> 내부 부재별 중량 합계.....	12
<표 5> 기관부 중량 내역 .....	12
<표 6> 총 경하중량.....	13
<표 7> 경하중량 .....	22
<표 8> 역할 분담 .....	32
<표 9> 세부일정 .....	33

## 1. 선체부 설계

본 장에서 선체부 설계를 크게 두 단계로 나누어 기술 하였다. 먼저 Proposal 단계에서의 기준선 선정과 개략 경하 중량의 추정 및 그에 따른 주요 치수의 결정 과정을 기초 설계로 칭해 기술하였고, 개략적으로 결정된 주요 치수를 토대로 EzShip 을 활용한 선형/구획 설계 및 세부 경하 중량의 추정과 그에 따른 설계 선형의 확정 과정을 상세 설계로 칭해 기술한 것이 두 번째 단계이다.

### 1.1. 기초설계

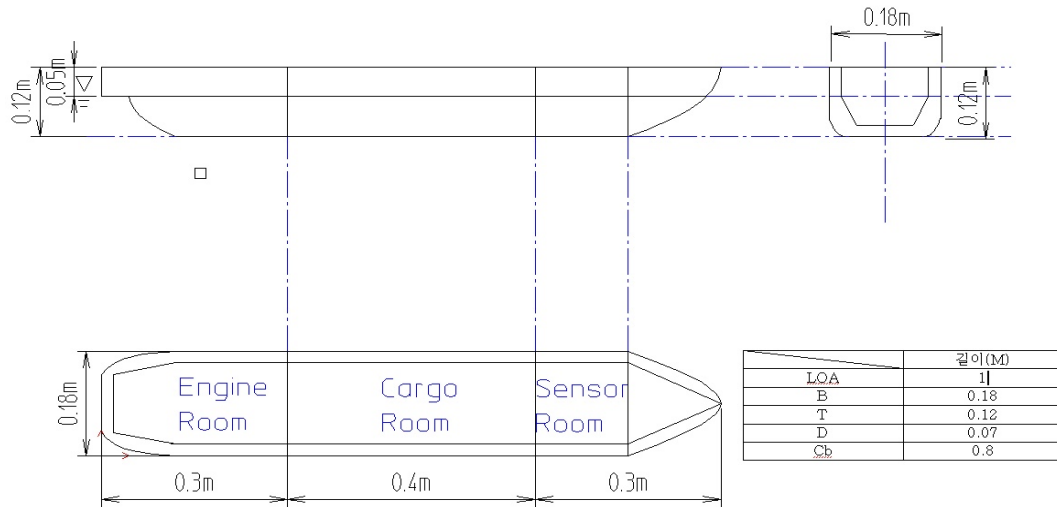
#### 1.1.1. 기준선의 선정

먼저 본 조의 설계에서 고려된 파제의 주요 요구조건은 다음과 같았다.

- 정해진 재화중량(6kg)을 실을 수 있어야 함
- 벽면과의 일정거리를 유지하는 자동제어 시스템의 구축
- 유사시 리모트 컨트롤러로의 전환을 통한 선박 조종성 유지

이번 프로젝트의 초점이 창의적 선형 설계에 있기 보다는 선박의 자동제어능력에 있다고 할 수 있다. 따라서 VLCC나 대형 화물선, 즉 방형계수가 커서 선회능력이 좋고 화물을 싣기 용이한 배들을 기준선으로 채택하는 것이 합리적 판단이라 생각하였다.

제안 당시 개략 설제도면 및 배치도는 다음과 같았다.



<그림 1> 개략배치도(Proposal)

### 1.1.2. 주요치수의 결정

1차 설계 단계에서 개략 경하 중량의 추정은 주요 치수(L, B, D, T)를 결정하기 위한 가장 핵심적인 요소이므로 선형설계시스템인 Ezship을 활용한 추정이 이루어지기 전까지 1차 주요치수의 결정을 위해 비교적 정확하게 이루어져야 한다.

과정은 다음과 같이 이루어진다

- L, B, T, D, Cb 의 선정
- 위 치수에 따른 만재배수량 계산
- 경하중량의 추정
- $DWT = \text{만재배수량} - \text{경하중량}$  의 관계식에서 DWT가 요구재화중량인 6kg 을 만족하는지 확인
- 만족하지 않으면 다시 처음 단계에서 주요수치 수정 후 동일 과정 반복을 통한 주요 치수의 확정

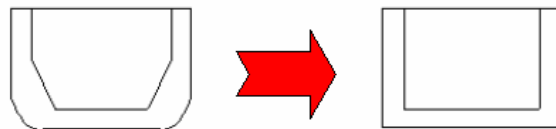
이 과정에서 어려운 부분은 경하중량의 추정이다. 주요 치수의 수정에

다른 선각 및 내부 부재의 중량의 변화를 계속적으로 계산하여야 하기 때문이다. 따라서 이 부분을 본 조에서는 마이크로소프트사의 엑셀을 이용 L, B, D, T, Cb 및 각종 부재의 크기, 개수 등을 입력 받아서 경하중량을 추정하게 하계끔 하는 프로그램을 작성하였다.

### 1.1.3. 경하중량의 계산

선체의 중량은 선각 중량과 내부 부재 중량으로 나누어 계산할 수 있다.

선각 중량은 구각의 부피를 구할 때 쓰는 방법으로 선각의 바깥쪽 면으로 이루어진 입체의 부피에서 선각의 안쪽 면으로 이루어진 입체의 부피를 빼는 방식으로 구하였고 내부 부재 중량은 단면을 단순화 시켜서 개략적으로 그 면적을 구하였다.

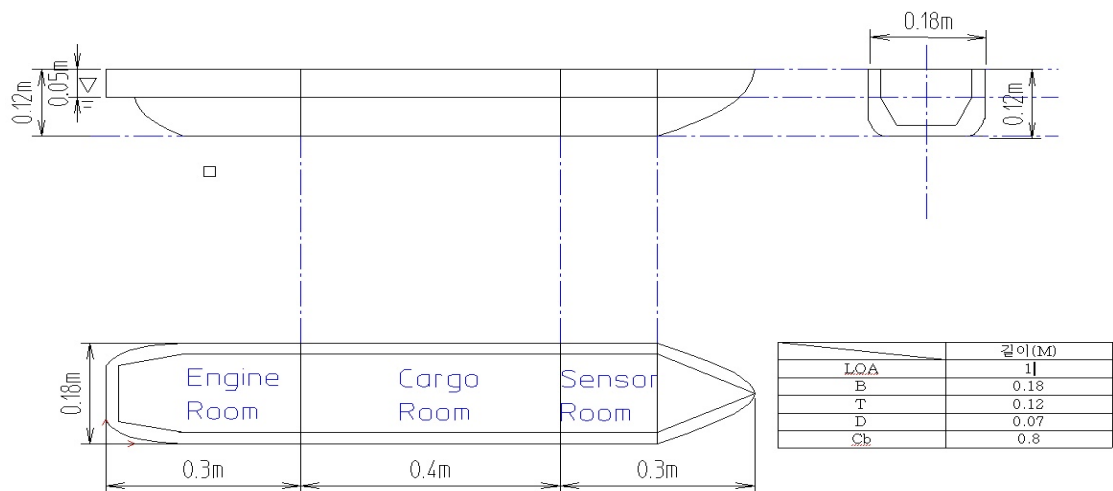


<그림 2> 중앙평행부 면적 근사

### 1.1.4. 개략배치도 및 구획배치도

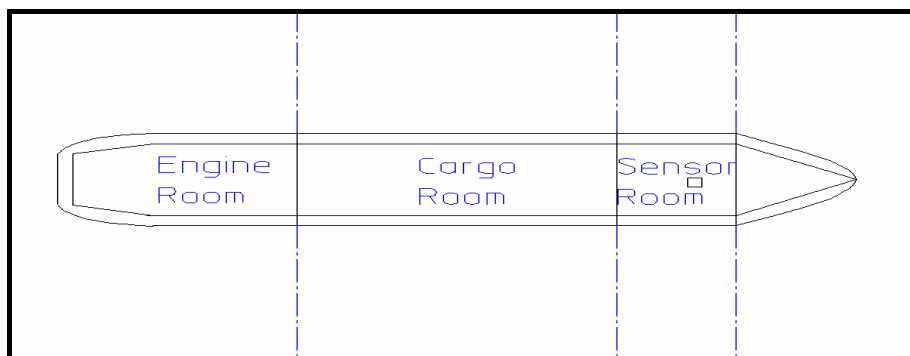
다음은 우리 조가 제작 할 모형선박의 개략 배치도이다.





<그림 3> 모형선의 G/A

다음은 우리 조가 제작 할 모형선박의 구획 배치도이다.



<그림 4> 모형선의 구획배치도

<표 1> 구획배치

	배치될 품목
Engine Room	러더, DC모터, 서버모터, 추진축, 프로펠러, 수신기, 저항 및 변속기
Cargo Room	배터리, 제어기

Sensor Room	센서,
-------------	-----

## 1.2. 상세설계

### 1.2.1. 선형 설계 과정

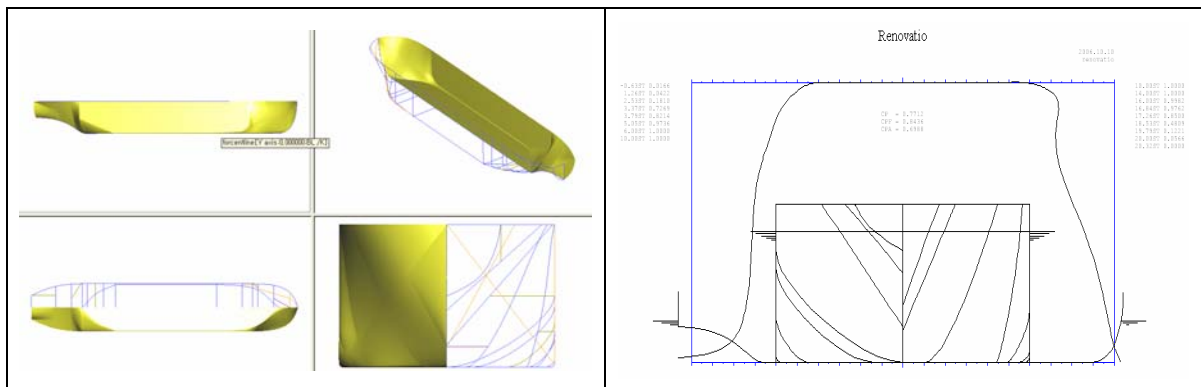
본 설계를 위하여 EzShip System을 활용하는데 이는 크게 두 단계로 구분된다.

첫째 단계는 EzHULL을 이용한 선형 설계,

둘째 단계는 EzCOMPART를 활용하여 개략배치도(G/A)에 맞게 구획을 배치하고 모형선 제작에 필요한 내부 부재의 형상을 결정하게 된다.

#### 1.2.1.1 선형 곡면 모델 생성 및 곡면 fairing

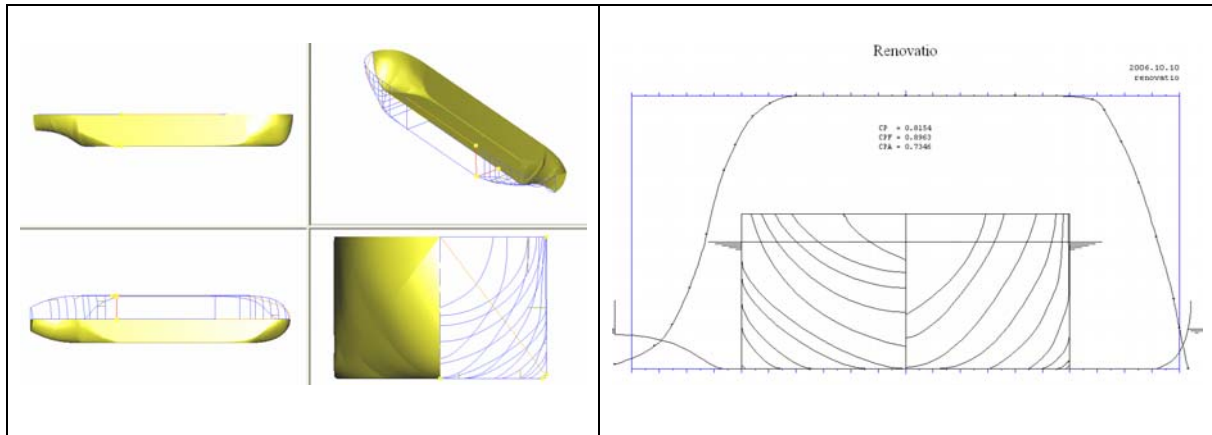
이렇게 일차적으로 작성된 파일을 이용하여 선형 곡면 모델을 생성해본 결과 다음과 같았다.



<그림 5> fairing 전 선형 곡면 모델, cp curve

출력된 cp curve에서 선수부의 곡선이 매끄럽지 못함을 알 수 있다. 이것은 횡단면적의 불규칙적인 변화를 의미하며 선체 저항에 영향을 미쳐 배의 성능에 악영향을 줄 수 있다.

따라서 선수부와 선미부의 side라인을 더 추가하고 직선을 곡선으로 보강해 준 후 cp curve를 확인하는 과정을 여러 번 거쳐 마침내 바람직한 선형을 얻게 되었다. Fairing 후 선형 곡면 모델 및 cp curve는 다음과 같다.



<그림 6 >fairing 후 선형 곡면 모델, cp curve

L O A = 100.00 M	L/B = 5.2778	LCB = 3.94 % LPP
L B P = 95.00 M	B/T = 2.5714	VCB = 3.68 M
BEAM = 18.00 M	CB = 0.8151	KMT = 7.6797 M
DEPTH = 12.00 M	CBF = 0.8959	LE = 28.500 M(30.00 %)
DRAFT = 7.00 M(TF)	CBA = 0.7343	LX = 38.000 M(40.00 %)
= 7.00 M(TA)	CM = 0.9996	LR = 28.500 M(30.00 %)
SPEED = 0.00 KTS	CW = 0.9363	AFP = 18.796 M^2(14.92 % AM)
FN = 0.0000	CWF = 0.9584	BL = 0.000 M( 0.00 % LPP)
SCH = 0.000 M	CWA = 0.9142	WSA = 2607.0 M^2
DIA = 0.000 M	CB/ (L/B) = 0.1544	VOL = 9756.6 M^3

<그림 7> fairing 후 cp curve 상세 정보

fairing 후 cp curve를 보게 되면 fairing 전에 비해 선수부 선미부 모두 부드러워진 것을 볼 수 있다. 일차적으로 설계 완료된 모형선의 주요 치수는 다음과 같다.

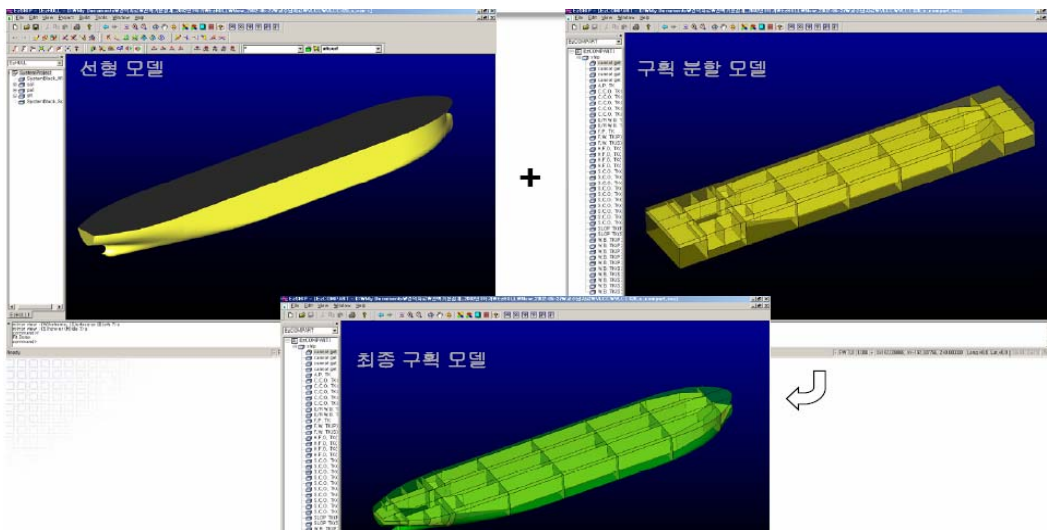
L O A = 100.00 M  
 L B P = 95.00 M  
 BEAM = 18.00 M  
 DEPTH = 12.00 M  
 DRAFT = 7.00 M(TF)  
 = 7.00 M(TA)

### <그림 8> 일차 선형 주요치수

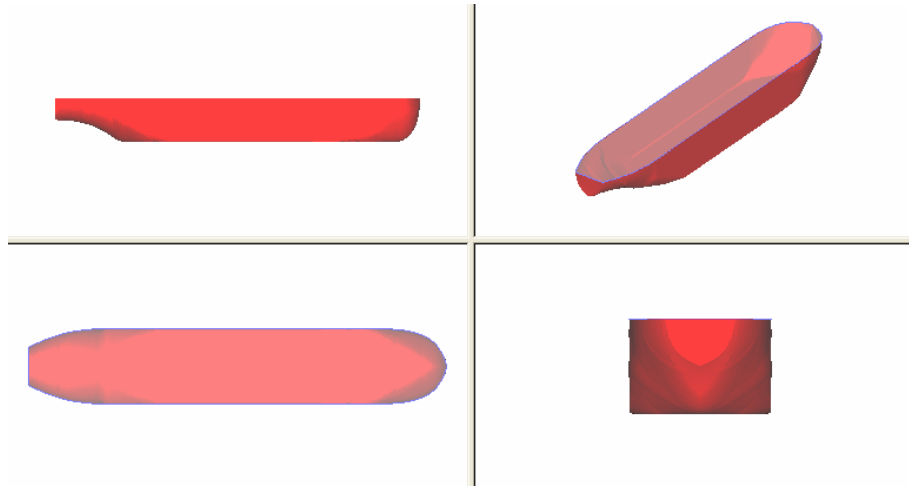
위 cp curve 상세 정보를 보게 되면 방형계수(Cb)는 0.8154 만재 배수량은 9.756kg 으로 Proposal 당시 결정된 주요치수와 거의 동일하게 일차 선형 설계가 완료되었다.

### 1.2.2. 구획 설계 과정

EzCompart 를 이용하여 구획 모델링을 하는데 앞서 제작된 선형 모델과 구획 분할 모델을 합쳐 최종 구획 모델을 형성하게 된다.



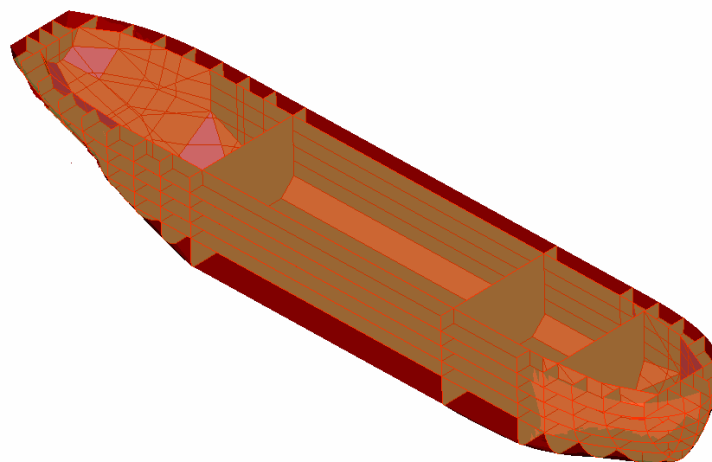
<그림 9>최종 구획 모델 생성 과정



<그림 10> 전폭 선형 이진 파일의 모습

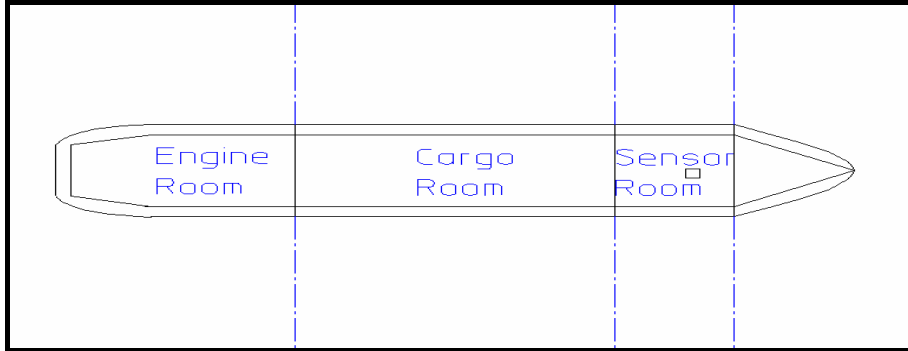
위의 스크립트는 격벽 배치를 통한 구획 분할 뿐만 아니라 모형선 제작 시 필요한 부재의 모양을 결정하는 내부 블록의 모양을 설정한다.

Xplane은 격벽 및 횡방향 부재의 위치를, yplane은 용골의 위치를, zplane은 종방향(테두리) 부재위 위치를 결정하게 된다. 아래의 그림은 물론 실제 부재의 모양은 아니다. 실제 네스팅 및 커팅하게 될 부재의 모양은 drawGA를 통해 각 부재의 절단면을 AUTOCAD파일로 export함으로써 도면으로 출력된다.



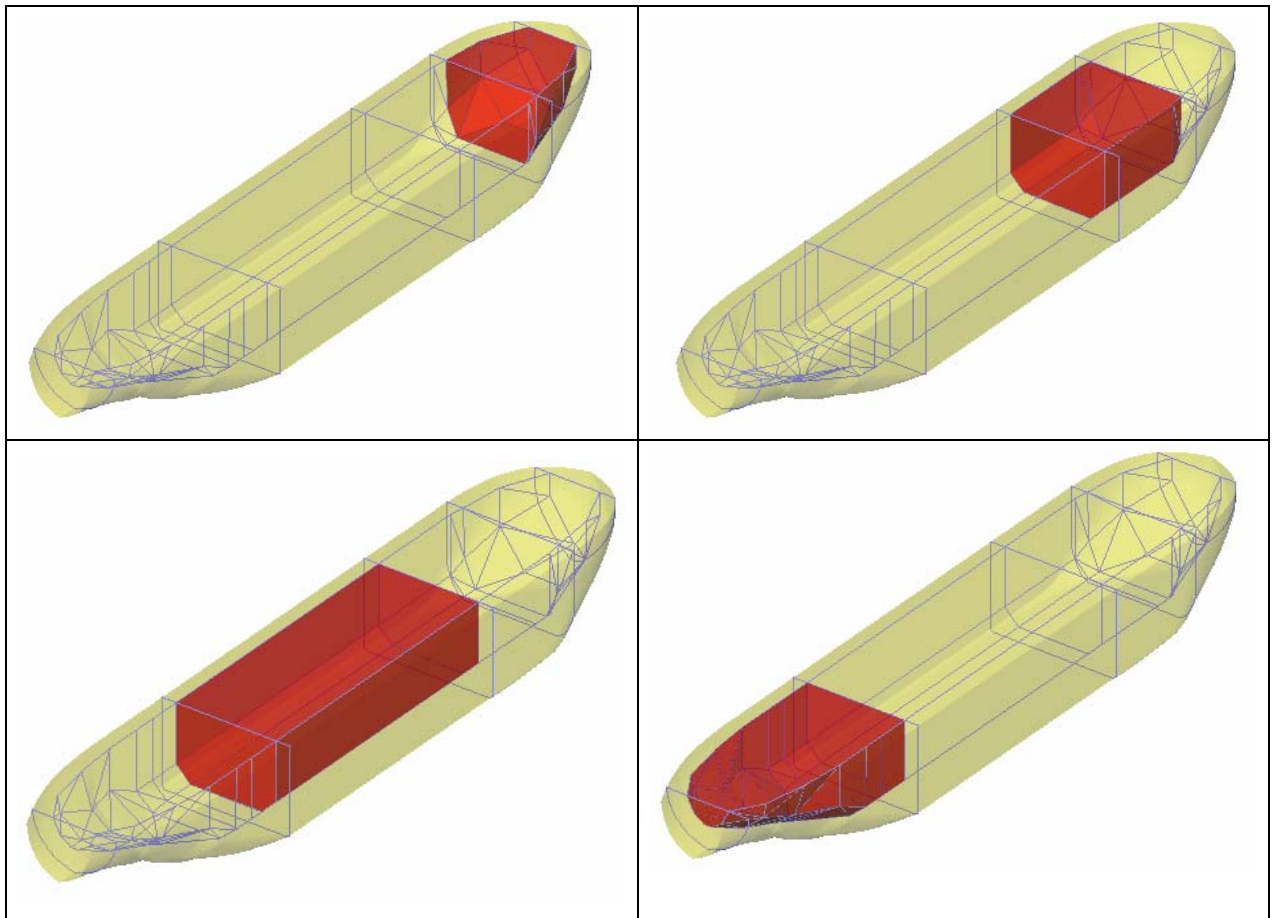
<그림 11> 선형 이진 파일과 구획 분할 모델을 합성

<그림 9>과 <그림 10>을 참조하면 Proposal시와 동일하게 격벽을 배치하여 구획을 나눠놓은 것을 알 수 있다.



<그림 12> Proposal 당시의 구획 배치도

아래 그림들에서 빨간 색으로 반전된 공간이 하나의 구획이다.



<그림 13> Ezship을 이용해 나타낸 구획의 모습

<표 2> 각 구획별 배치 품목

	배치될 품목
Engine Room	러더, DC모터, 서버모터, 추진축, 프로펠러, 수신기, 저항 및 변속기
Cargo Room	배터리, 제어기
Sensor Room	센서,

### 1.2.3. 세부 경하중량의 추정

EzHull과 EzCompart 를 이용하여 선박 형상 및 구획 배치를 끝내게 되면 실제 모형선 제작 시 조립하게 될 부재의 도면까지 출력할 수 있게 된다. 따라서 이 단계에서는 Proposal시의 1차 개략 경하중량 추정보다 더 정확한 세부 경하중량 추정이 가능하다. 각 부재의 단면적과 선각 면적을 EzHull 및 AutoCAD 의 도움을 받아 정확하게 계산할 수 있고 이를 통해 각 부재 및 선각의 부피를 구하여 밀도를 곱해 계산하면 선체중량을 알 수 있게 된다. 이에 의장에 해당하는 기관부 중량을 더 해주면 앞으로 제작하게 될 모형선의 세부 경하중량 추정이 가능하게 된다.

#### 1.2.3.1.선체중량

선체 중량을 Hull에 해당하는 선각 중량과 내부 부재로 나누어 생각할 수 있다.

##### ※선각 중량

선각 중량은 선각 부분의 총 단면적에 그 재료의 두께와 밀도를 곱하여 그 중량을 추정하게 된다. 이 때 단면적의 정확한 계산을 위해 EzHull 프로그램을 이용한다. 한편, 선각의 단면적은 결국 배 전체가 물에 잠겼을

때 침수 면적과 같으므로 EzHull 툴을 이용해 흘수를 배의 Depth 까지 올려서 침수 면적을 구함으로써 선각의 단면적을 알 수 있게 된다. 이는 cp curve에서 확인할 수 있는데 흘수를 본 조의 모형선의 Depth 인 12까지 올려서 cp curve를 통해 단면적을 알아본 결과 다음과 같았다.

- 선각의 단면적 :  $3694.9 \text{ cm}^2$

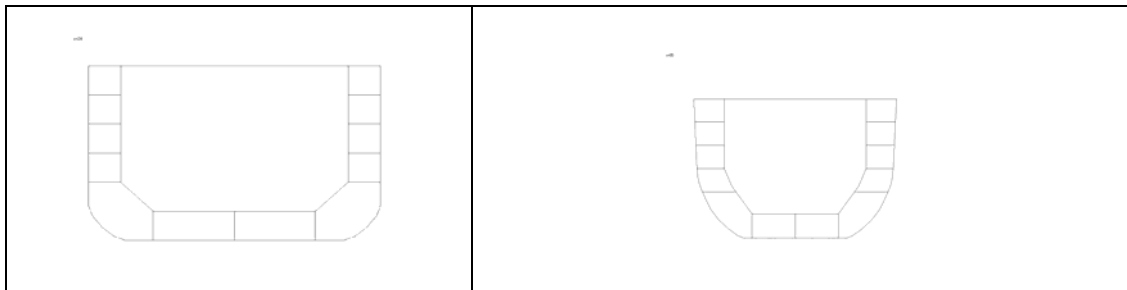
이를 이용해 선각 부분을 구성하게 되는 우드락, 에폭시 및 퍼티 각각의 무게 및 선각 총 중량을 표로 정리해보면 다음과 같다.

<표 3> 선각 중량

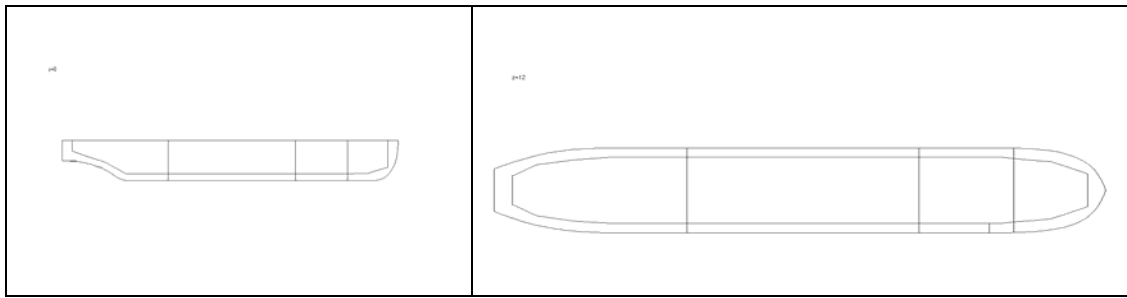
	두께(cm)	부피( $\text{cm}^3$ )	중량(g)
우드락	0.2	738.98	517.286
에폭시	0.05	184.745	277.1175
퍼티	0.1	369.49	498.8115
계			1293.215

#### 1.2.3.2. 내부 부재 중량

AutoCAD 프로그램을 이용함으로써 내부 부재들의 넓이를 정확하고 간단하게 계산할 수 있다. AutoCAD에서 화면 출력해본 내부 부재의 모양들은 다음과 같았다.







<그림 14> 부재 단면도

이러한 단면들을 AutoCAD를 통해 각 부재의 면적을 구하고, 단면 및 위치별 부재 중량을 종합하여 내부 부재의 총 중량을 추산해보면 다음과 같다.

<표 4> 내부 부재별 중량 합계

	두께(cm <sup>2</sup> )	부피(cm <sup>3</sup> )	중량(g)
횡방향부재	0.2	738.98	500.8873483
종방향부재	0.05	184.745	452.881
용골	0.05	184.745	98.0425
계			1100.832

#### 1.2.3.3. 기관부 중량

기관부 중량은 앞서 계산한 선체부 중량에 비해 고정적이고 실측 중량으로 그 무게를 쉽게 알 수 있다 그 내역과 중량은 다음과 같다.

<표 5> 기관부 중량 내역

분류		중량(kg)
추진부 및	추진축	0.05
	DC모터	0.05

제어부 중량	서브모터	0.04
	RUDDER	0.01
	수신기	0.02
	배터리	0.5
	프로펠러	0.01
	저항+ 변속기	0.06
	초음파 센서	0.10
	회로기관	0.16
계		1

#### 1.2.3.4. 총 경하중량의 추정과 재화중량

앞서 부분별로 구한 중량을 합산하여 표로 나타내면 다음과 같다.

<표 6> 총 경하중량

분류	중량(g)	계(g)
선각부	1293.215	
내부		2394.047147
선체중량 부재	1100.832147	
기관부중량	1000	1000
계		3394.047147

이를 통해 본 설계선의 재화중량을 계산해 보았다.

$$W=LWT+ DWT$$

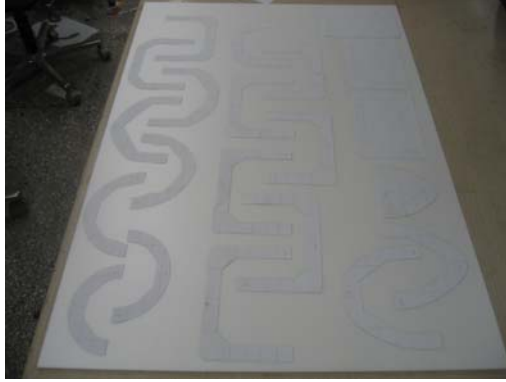
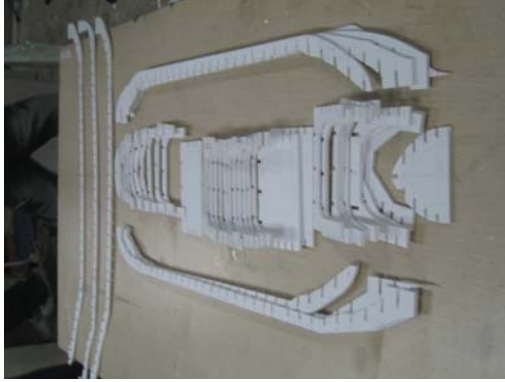
에서 만재배수량  $W=9.756\text{kg}$  이었고 <sup>1</sup> 경하중량  $LWT=3.494\text{kg}$ 으로

<sup>1</sup> 1.1절 참조

추정되었으므로 본 설계선은  $DWT=W-LWT$  에서 6.262kg의 재화중량을 가질 것으로 추정된다. 이는 요구조건인 재화중량 6kg을 약간 상회하는 것으로써 약간의 마진을 고려한다면 설계선의 추가적인 홀수나 선형 변형이 필요 없다는 것을 알 수 있었다. 이는 Proposal 당시의 개략 경하 중량 추정값<sup>2</sup>과 이번 EzShip System과 AutoCAD를 이용한 세부 경하중량 추정 값이 거의 일치 했기 때문이라고 판단된다. 이로써 1차 설계 그대로 모형선 제작에 착수하게 되었다.

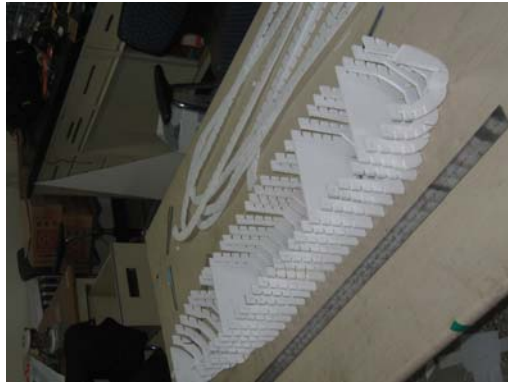
## 2. 선체부 제작

Ez compart로 구획설계까지 마친 후 AutoCad를 이용해 도면을 출력하고 출력된 도면 부재를 우드락에 붙여 Nesting & Cutting 작업을 하고 조립과 외관접합, 방수처리를 함으로서 모형선박을 완성한다.

 <p style="text-align: center;">Nesting</p> <p>부재 소모를 줄이기 위해 최적의 배치를 하는 작업을 하였다</p>	 <p style="text-align: center;">Cutting</p> <p>Nesting을 끝낸 후 칼과 자를 이용해 도면에 맞게 부재를 자른다.</p>
--	--

---

<sup>2</sup> 3.427kg



부재조립

Cutting 작업 후 Keel 을 기준으로  
모든 부재를 조립한다.



외판접합

부재 조립이 끝난 후 1mm  
우드락으로 2번 선체 외판을  
접합하였다.



Putty

선체 외판에 Putty를 바르고 사포를  
이용해 표면을 매끄럽게 하고



Painting

방수처리를 확실히 하기위해  
Epoxy를 얇게 골고루 바른 후  
Painting 을 하였다.

	
<p>흘수표시</p>	<p>추진기 설치</p> <p>Motor, Propeller, Shaft 축을 이용하여 Glue gun과 Silicone으로 방수 처리를 하였다.</p>

<그림 15> 단계별 선체 제작 사진

## Rudder 제작

러더는 선체 제작과 병행하여 설계 및 제작 하였다.

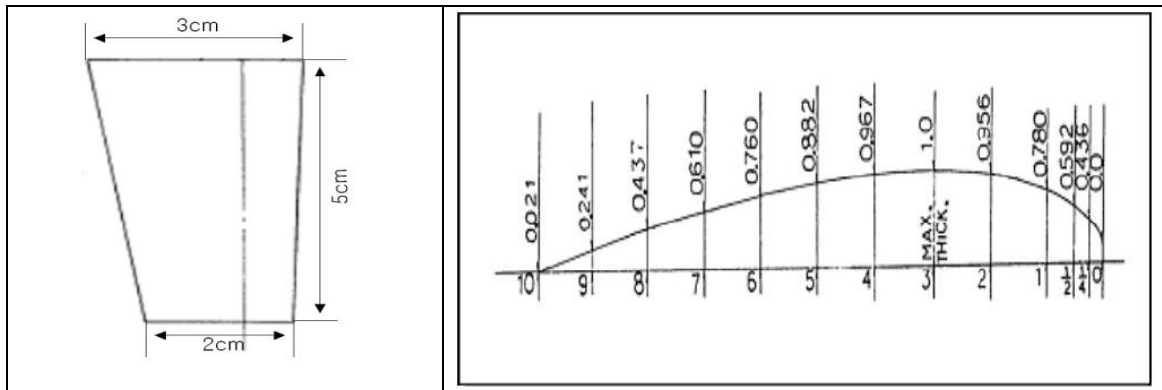
우리 조의 러더는 스페이드 타로 선택하여 제작하였다. 현재 실제 선박에 많이 사용되는 semi-spade rudder type과 비슷하고 제작이 더 용이한 spade rudder를 선택하게 되었다.

## Rudder 치수의 결정

DnV(노르웨이 선급) 면적 추정식을 이용한 계산해 본 결과 다음과 같았다.

$$\frac{A_R}{L \cdot d} = 0.01 + 0.5 \cdot \left( \frac{C_b}{L/B} \right)^2 = 0.021777$$

이 수치에서 Rudder의 면적 값을 계산해본 결과  $A_R = 14.48 \text{cm}^2$  임을 알 수 있었다. 그리고 Rudder 단면은 NACA 단면을 출력해 활용하기로 하였다



<그림 16> Rudder 단면

## Rudder 제작

Rudder가 제작 된 결과

Spade Type으로 제작 된 Rudder의 면적이  $13.75\text{cm}^2 \sim 15\text{cm}^2$  임 을 알 수 있었고 설계 목표 값인  $A_R=14.48\text{cm}^2$  을 잘 만족함을 알 수 있었다.

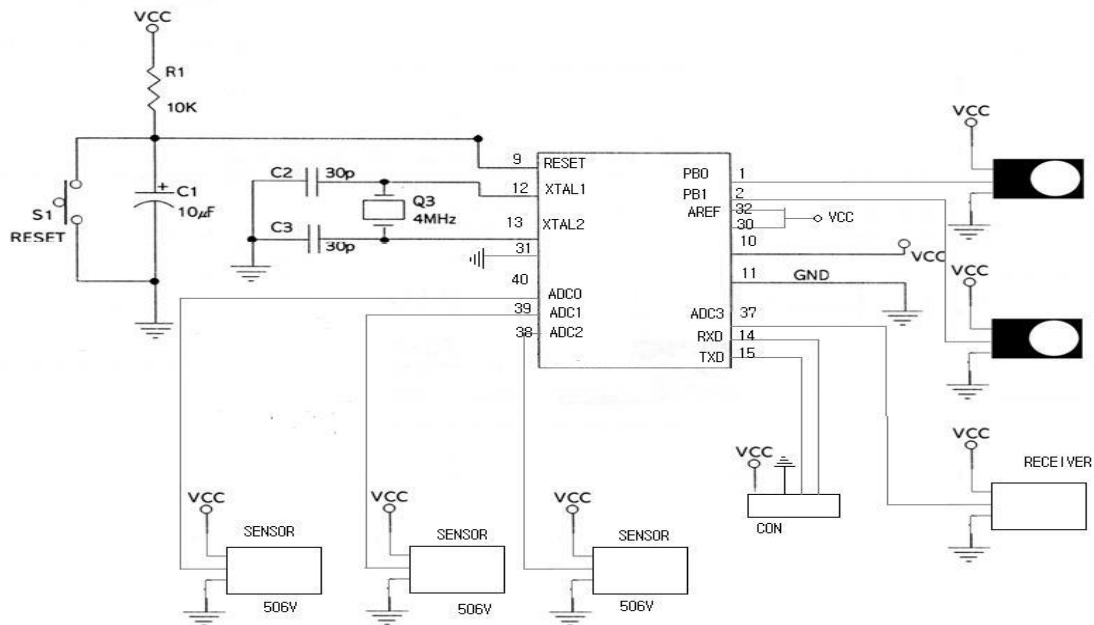
## 3. 제어부 설계

### 3.1. 하드웨어 설계

#### 3.1.1. CONTROLLER 설계(1차)

실제로 선박의 자동제어를 하기위해 필요한 CONTROLLER의 설계이다. 자동제어를 위해서는 먼저 초음파센서를 이용해서 장애물과 선박과의 거리를 측정해야 한다. 측정된 값에 따라 SERVO MOTOR와 변속기에 값을 출력해 주어 선박이 방향을 선회하고 속도를 제어하게 된다.

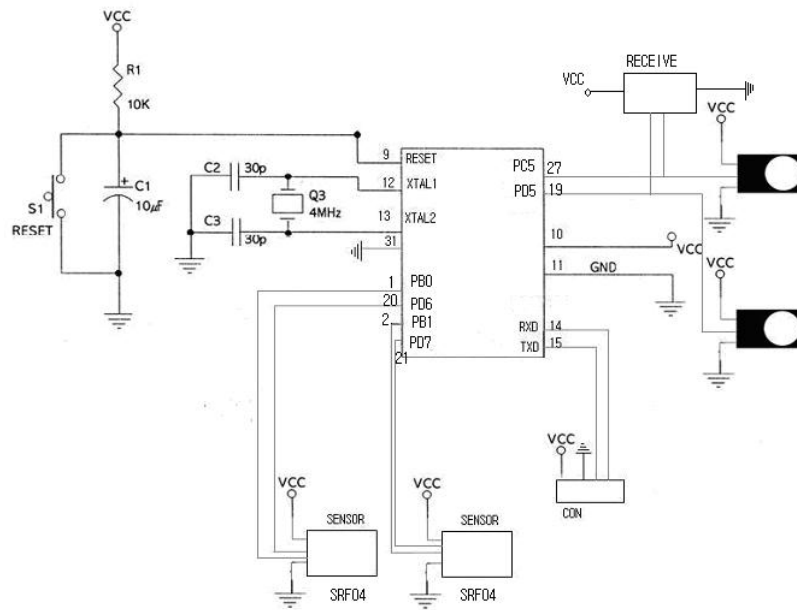
이러한 CONTROLLER를 만들기 위해 회로도를 구성해보았다.



<그림 17> controller 회로도

### 3.1.2. CONTROLLER 설계(2차)

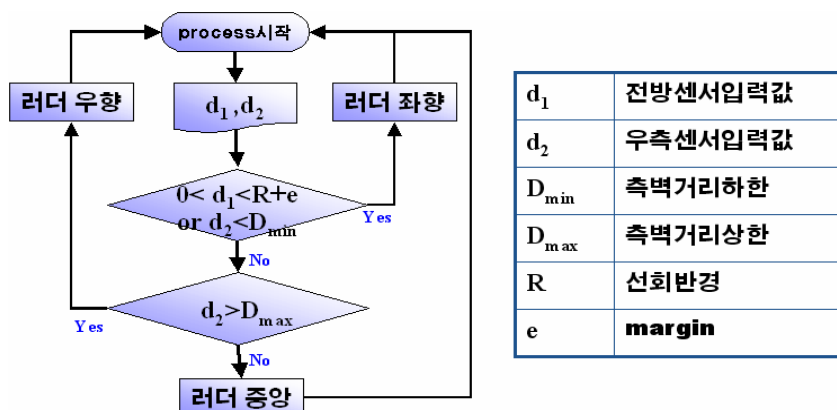
자동/수동 제어 부분의 기계적 스위치로의 변환과 센서를 변경한 탓에 새로운 Controller 기판을 설계하였다. 1차 회로와 비교를 해보면 먼저 센서가 변했다. 1차 설계 때 사용한 센서는 센서에 전압만 공급해주면 다른 작업 없이도 바로 아날로그 신호 값이 나오지만 이 값을 ADC로 받아들여서 변환해주는 작업이 필요하다. 그러나 새로운 센서는 굳이 PORT A에 연결할 필요 없이 아무 포트나 두개의 포트를 할당하여 하나는 센서에 일정한 주기의 펄스파를 주고 하나의 포트에서는 그 값을 읽어 들이는 것이다.



<그림 18> 2차 controller 회로도

### 3.2. 소프트웨어 설계(알고리즘)

주어진 경로로 주행 시 벽과 간격의 멀어지고 가까워짐에 대한 러더 움직임을 알고리즘으로 표현



<그림 19> 주어진 경로를 주행하는 알고리즘



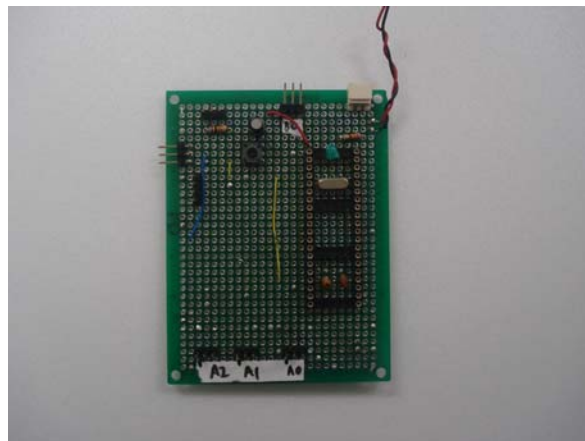
## 4. 제어부 제작

### 4.1. 하드웨어

#### 4.1.1. Controller 제작(1차)

Controller는 자동제어에 핵심적인 회로인 만큼 다른 여러 회로를 만들어 회로에 대한 이해를 높인 후 제작했다. 지금까지 사용했던 모든 부품이 종합적으로 들어가게 만들었다.

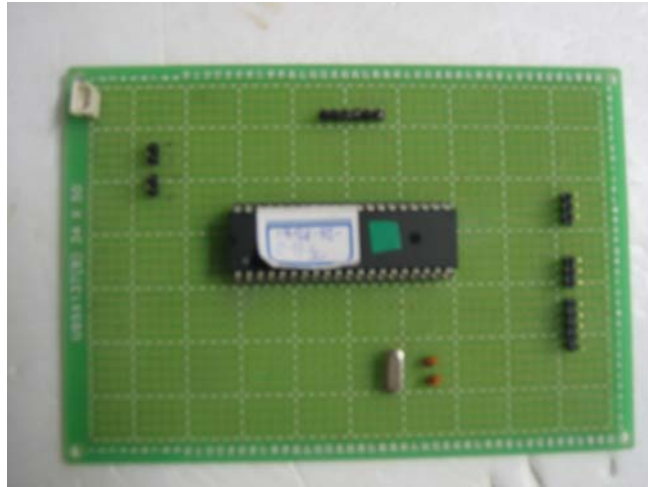
\* 준비물 : ATMEGA8535, 30pF 캐퍼시터 2개, 4Mhz 크리스탈, 10 $\mu$ F 전해콘덴서, 10K $\Omega$ , 저항 3개, 서보모터, 초음파센서 3개, RS232케이블, 전원콘넥터 2개, 커넥터 5개



<그림 20> Controller 의 모습

#### 4.1.2. Controller 제작(2차)

설계 변경에 따라 다시 제작하였다.



<그림 21> 최종 controller 앞면

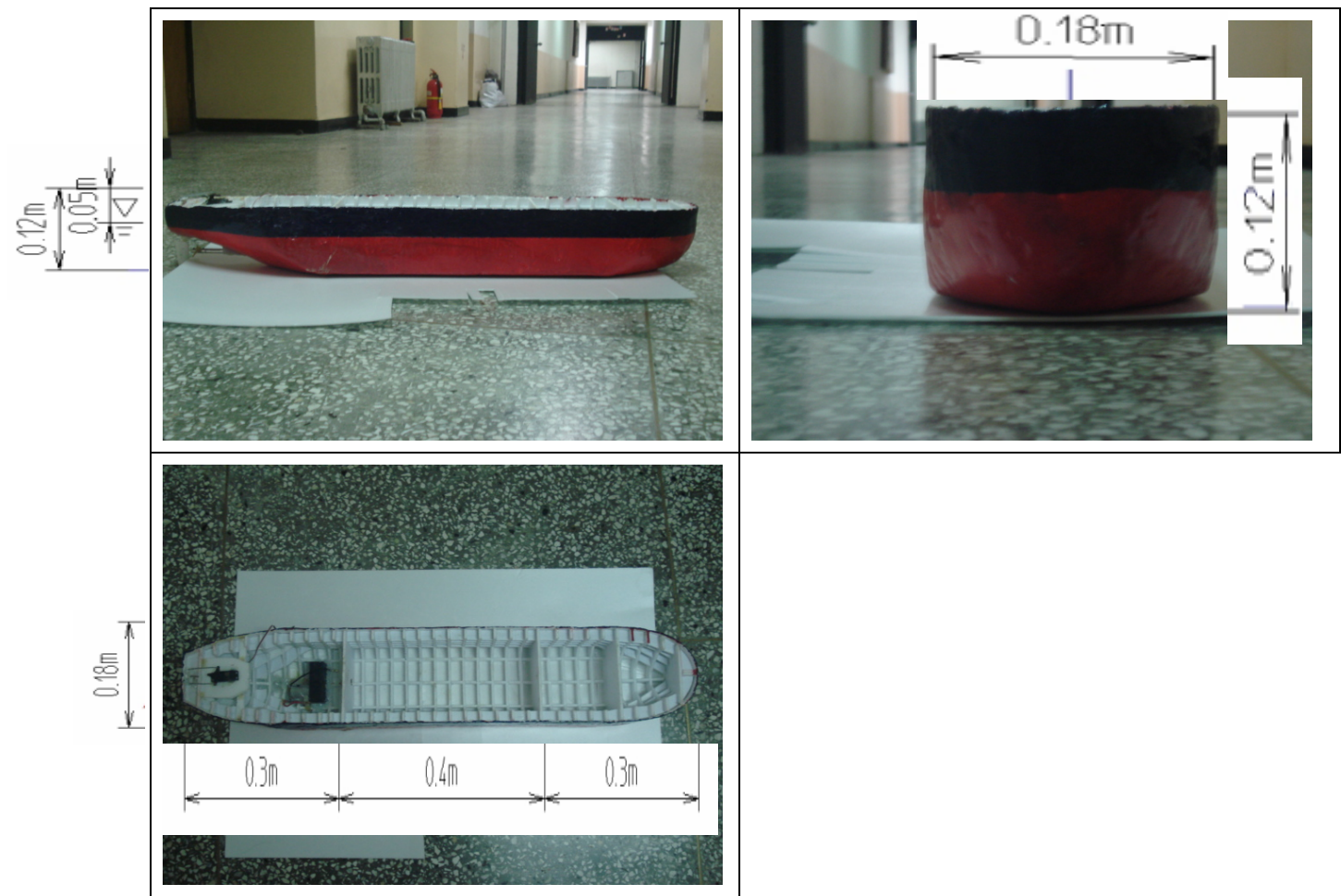
## 4.2. 소프트웨어

### Controller 프로그래밍

1차 Controller 설계에 따른 프로그래밍의 특징은 통상적으로 고안되던 2개의 수신기와 2개의 크리스탈을 통한 자동/수동 전환 스위치를 사용하던 방법에서 벗어나 하나의 수신기만을 이용하여 송신기의 전원이 켜질 때 발생하는 펄스파를 검출하여 소프트웨어 적으로 전환을 처리하는 방법이였다. 하지만 2차 Controller 설계 시 다시 원안으로 돌아가 스위치를 이용하게 되었고 이에 따른 재 프로그래밍을 하게 되었다. 1차 Controller Program에 비해서 달라진 점은 크게 두 부분이다. 1차의 경우 506nv 를 사용해 별다른 설정 없이 ADC 를 통해 변환된 값만 읽어 들이면 센서의 출력 값을 얻을 수 있었던 것과는 달리 따로 trigger pulse 를 입력해주는 부분이 추가 된 것이 그 중 하나이다. 또 하나는 DC motor의 제어가 서보 모터와 같이 pulse 파의 생성을 port 출력을 통해 생성한 것이 아니라 내부 타이머 기능을 이용하여 pwm 파형을 생성하여 DC motor의 rpm을 일정하게 유지하도록 한 점이다.

# 5. Contest

## 5. 완성된 모형선



<그림 22> 완성된 모형선의 개략배치도

<표 7> 경하중량

선체	기관부			추진부	계
2.5(kg)	배터리	모터	센서 및 기타회로	0.05(kg)	3.5 (kg)
	0.5(kg)	0.05(kg)	0.4(kg)		

## 5.2. 1차 Contest

### 5.2.1. 1차 컨테스트 준비

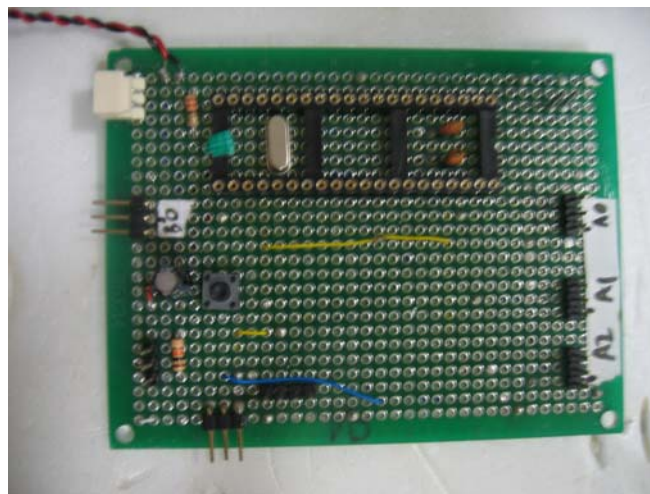
#### 5.2.1.1. 선체부

11월 7일 - 자동제어 부분과 흡수선 긋는 것을 뺀 선체 모든 부분을 완성하고 수조에서 시운전을 하였다. 하지만 물이 새는 것을 발견하고 퍼티작업부터 다시 시작해서 선체 외관을 단단하게 하기로 결정했다.

11월 8일 최종보고서 발표가 끝나자마자 바로 퍼티작업에 들어갔다. 처음에 실패한 것을 경험삼아 좀더 세심하게 골고루 퍼 발랐다. 11월 9일에 바로 에폭시작업을 하였다. 오전에 일찍 에폭시를 발랐더니 저녁때쯤 말라 바로 페인트 칠을 할 수 있었다.

#### 5.2.1.2. 제어부

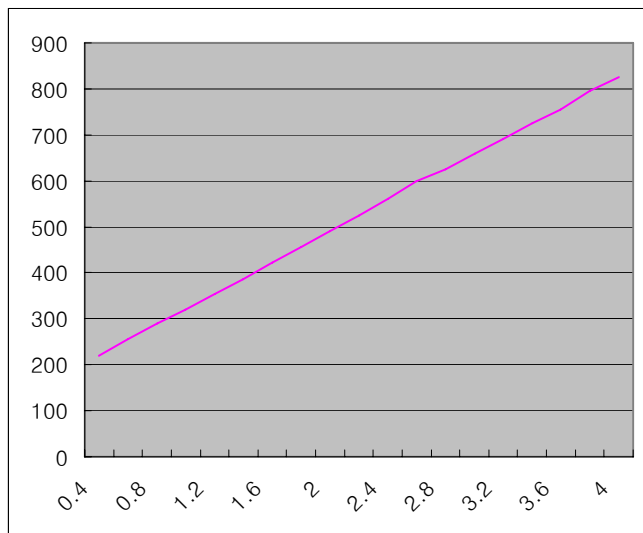
11월 3일 자동제어를 위한 제어기판을 최초로 제작하였다. 기판은 과제를 수행할 때 서보모터를 제어하기 위해 만들었던 기판에 약간의 수정을 해서 만들었다.



<그림 23> 자동제어기판a

11월 7일 작성된 기관으로 센서를 연결하여 RS232를 통해 컴퓨터로 입력 값이 표시되는 간단한 프로그램을 만들어서 센서 입력값을 테스트 해보았다. 그러나 값이 완전 랜덤하게 나왔다. 상황을 해결하기 위해 우리조에서는 센서회사인 센서텍에 연락을 해서 기술자와 통화를 하였으나 센서의 사용법에 대해선 아무것도 알아 낼 수 없었다. 우리는 센서가 불량일 확률이 높다고 생각하고 더 이상 작업을 진행하지 못했다.

11월 9일 기영이형이 모든 조의 대표로 센서텍회사에 직접 갔다. 우리조에서는 기영이형이 새로운 점을 알아오면 바로 새로운 기관을 작성할 수 있도록 용산에가서 부품을 넉넉히 구매해 놓았다. 저녁에 기영이형이 돌아왔으나 센서에는 아무 문제가 없다는 대답만을 들었다고 했다. 그렇지만 12V의 전압을 레귤레이터를 이용해 5V로 낮추어 사용해보자고 했다. 다행히도 같은 전압을 떨어뜨려 사용하니 센서값이 어느정도 의미있는 값을 가지는 것을 알 수 있었다.

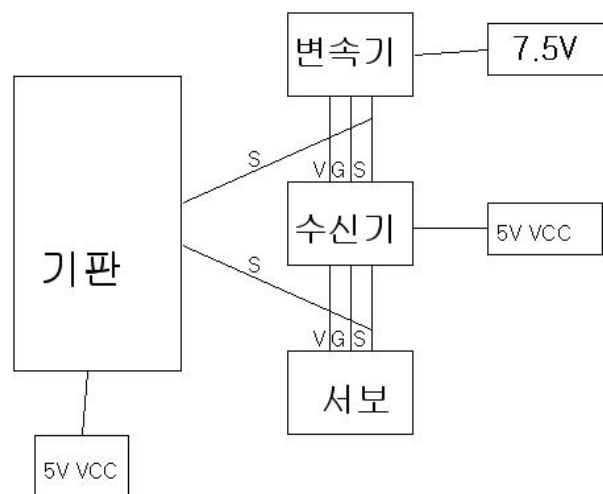


<그림 24> 거리에 따른 센서값

그러나 센서를 테스트 해본 결과 센서와 벽의 각도가 직각일 때에는

센서값과 거리가 위와 같은 비례적인 값을 가졌으나 센서와 벽의 각도가 조금 틀어져도 거리에 훨씬 큰 값이 나오게 되는 것을 알 수 있었다.

11월 10일 이제 남은 일은 자동/수동 변환만 하면 된다고 생각하고 자동/수동 변환에 힘을 쏟았다. 우리조는 다른조와는 달리 자동/수동 전환을 소프트웨어적인 방법으로 하기로 했다. 수신기와 기판을 연결해주어 수동제어를 위한 리모콘이 켜졌을 때 수신기의 펄스값이 바뀌는 것을 이용해서 기판에서 수신기의 바뀐 펄스파를 받아들이면 자동제어를 중단하고 수동으로 전환되는 시스템 이었다. 여러가지 장점이 있는 방법이었지만 신호의 역류 등 여러 기술 상의 문제로 결국 소프트웨어를 통한 자동/수동 제어 변환을 포기하고 스위치를 통해 변환하기로 했다. 결국 센서는 차치하고서라도 기판을 통한 변속기와 서보모터의 제어라는 과제도 해결하지 못 한채 컨테스트에 나갈수 밖에 없었다.



<그림 25> 최초의 제어부 연결

### 5.2.2. 1차 컨테스트 결과 및 향후 과제

컨테스트 결과 우리배가 직진성에서 상당히 우수한 면을 보이며 1등을 했다. 뛰어난 직진성을 보일 수 있었던 점은 먼저 우리배가 다른 배들에

비해 유선형으로 생겼기 때문일 것이다. 또 전날 연습을 해보며 러더를 중앙에 정확하게 맞추는 작업도 중요했다. 또 다른 배들에 비해 우리 배의 프로펠러와 러더의 간격이 좁았던 점이 러더의 효과를 극대화 시켜 준 것 같다.

이 날은 수동제어에서 우리배가 좋은 성능을 보여주기는 했지만 자동제어가 전혀 안 되고 센서의 문제가 하나도 해결되지 않아서 2차 컨테스트 때까지 일주일의 시간동안 이 문제를 해결해야 되는 문제를 남겼다.

### 5.3. 2차 Contest

#### 5.3.1. 2차 Contest 준비과정

1차 Contest에서 우수한 직진 안정성을 검증 받은 덕분에 추가적인 선체부의 보완 작업은 불필요하다고 판단되었다. 해결해야 할 과제들을 정리해보자면 다음과 같았다.

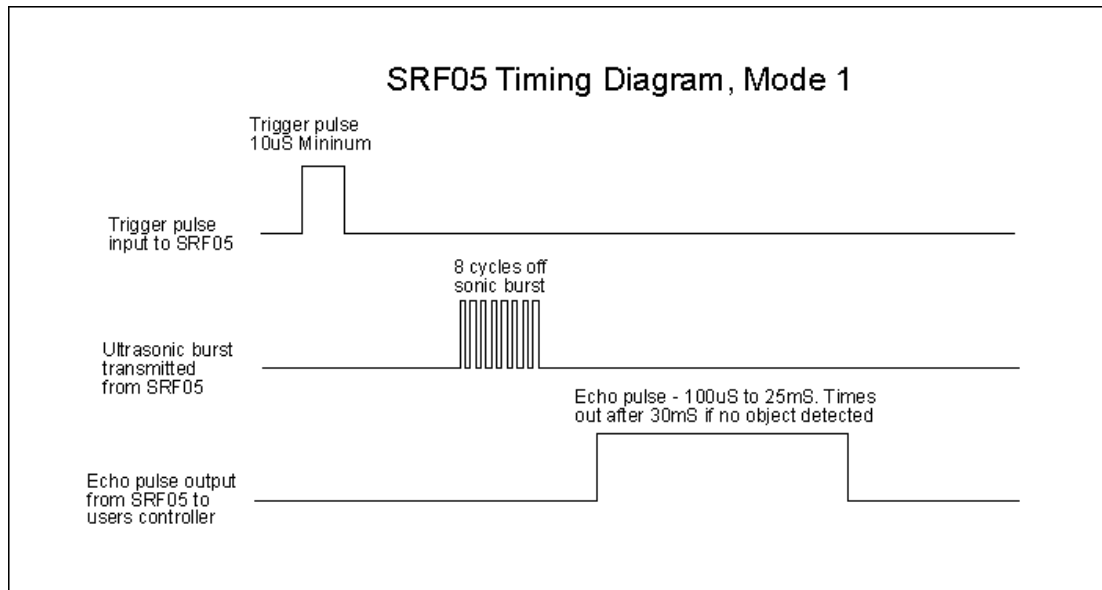
- 센서의 안정적인 구동
- 안정적 자동/수동 전환 방법 확보
- 자동 모드 시 Rudder와 DC motor 의 S/W 제어

1차 Contest 당시 사용하고 있던 506nv 초음파 센서를 변경하기에 앞서 다시 한번 정확히 상태를 체크하기 위해 calibration 작업을 다시 수행하였다. 적외선 센서나 타 종의 초음파 센서로 교체하는 것은 잠시 유보해두고 기존의 506nv 모델을 간단한 시험용 기판에 연결하여 터미널로 출력 값을 다시금 검토하였다. 화요일이 지나가도록 문제는 여전히 발견되었고 그 해답이 보이지 않았다. 센서의 검출면이 수직에서 5도 이상만 비틀어져도 출력 값이 크게 변해 사용할 수 없을 정도의 수치가 나와

버렸고 진동 등의 움직임이 있을 때는 물체가 있음에도 몇 초간 미검출 상태가 지속되면서 1v의 출력이 나와 그 값의 처리가 상당히 곤란했다. 실제 배의 움직임을 예상해볼 때 센서와 벽면이 수직을 이루면서 진행되는 경우는 거의 없을 것이므로 더욱 기존 센서의 활용이 힘든 상황이었다. 수요일, 2차 Contest를 3일 남겨둔 시점에서 우리 조는 센서를 변경할 수밖에 없다는 판단을 내렸고 그 즉시 대체 가능한 타 센서의 조사를 시작했다. 그 결과 적외선 센서와 srf04 라는 초음파 센서가 고려되었는데 적외선센서는 측정 거리가 1.5m가 채 되지 않아 후보에서 제외시켰다. 새로 선택한 srf04 라는 초음파 센서는 미니마우스나 여러 로봇에 흔하게 쓰이는 센서이기 때문에 본 과제에 적합할 것이라 생각했다. 일단 한번 시도해보자 하는 생각으로 센서를 구매한 후 그 작동 시험을 시작하였다.

센서를 교체한 후에는 그 작동에 어려움을 겪었다. 기존의 506nv 센서가 12V전원을 인가하기만 하면 별다른 조작이 필요없이 1~5v를 출력했던것에 비해 새로 구입한 srf04 초음파 센서 모듈은 아래 그림에서 보는 것과 같이 10us 폭의 Trigger pulse를 인가함으로써 모듈에서 초음파를 발생시켜 그 측정이 이루어지는 방식이었다.





**<그림 26> SRF04 작동 원리**

측정이 실시간으로 이루어져야 하고 그 Echo pulse 의 폭에 비례해서 거리가 출력되므로 주기적으로 Trigger pulse를 발생시키는 방법과 Echo pulse의 폭을 측정하는 방법의 연구가 필수적이었다. 시간은 촉박했으나 다른 대안이 없었기 때문에 초심으로 돌아가서 다시 AVR 관련 책들을 참조하면서 배경 이론 공부에 들어갔다. 그 결과 Trigger pulse 는 Atmega8535 chip의 내장 타이머0를 이용, Overflow interrupt를 통해 일정 시간 간격마다 10µs 폭의 pulse 를 발생시키는 데 성공하였다. Echo pulse 폭의 측정은 단시간에 이해하기가 힘든 부분이 많아서 정확한 폭을 측정하는 것은 포기하고 다음과 같은 방법을 고안해 개략적으로 폭의 길이에 비례하는 수치를 얻을 수 있었다.

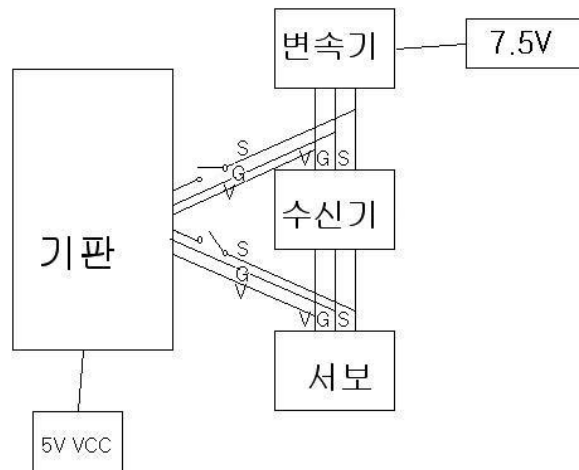
Echo pulse의 폭이 그 측정 거리에 비례해 100µs에서 25ms 까지 출력이 되므로 10µs 마다 Echo pulse 의 출력 값을 읽어 들여서 1일 경우 counting하는 방식을 취해봤더니 거리에 비례하여 증가하는 수치를 얻을 수 있었다. 상당히 rough 한 방식을 이용했음에도 이 센서 모듈이 제공하는 3cm 정도의 분해능을 얻을 수 있었다. 또한 작은 각의 변화에 대해서는 그

출력 값이 크게 영향을 받지 않았고 센서의 움직임에도 안정적으로 작동하였다. 30cm 이내의 근접거리에서도 잘 작동해 기존의 506nm에 비해 안정적인 거리 감지 능력을 보여주었다. 또 한가지 큰 장점은 동작 전원이 5v 인 것이었다. 기존의 센서가 12v에서 작동하여 전원 공급을 12v로 하고 정전압 regulator 를 통해서 회로에 5v를 인가하는 등 애로 사항이 있었던 반면에 srf04를 이용할 경우에는 5v 전원 하나 만으로도 regulator와 같은 추가적인 장치 없이 공통 전원을 공급할 수 있었다. 따라서 자연스럽게 1차 contest 준비 당시 전원의 공급문제로 발생했던 문제들은 더 이상 신경 쓸 필요성이 없어졌다.

센서를 통한 안정적 거리 측정 문제가 가장 큰 노력을 요구하긴 했지만 DC 모터의 제어 문제도 여전히 골칫거리였다. 다른 part들도 마찬가지였지만 독립적으로 제어를 했을 때는 별 문제가 되지 않던 것이 같이 연결할 때 정상적으로 작동하지 않는 경우가 많았던 것이다. DC 모터의 경우도 이와 같은 문제점이 있었다. 항상 동일한 속도로 동작하게끔 programming 했음에도 불구하고 그 동작 속도가 일정하지 못하였고 가끔 돌발적으로 rpm이 높아지는 현상을 발견하였다. 이를 해결하기 위해 rudder의 움직임과 관계 없이 센서의 trigger pulse 를 발생시켰을 때 사용했던 내부 타이머 기능을 이용하여 고속 pwm(Pulse Wave Modulation) 파형을 발생시켜 다른 내부 알고리즘 동작과 독립적으로 항상 균일한 pulse과 입력을 통해 DC motor의 rpm을 안정적으로 유지시킬 수 있었다.

안정적인 자동/수동 전환을 위해서도 많은 노력을 기울였다.

11월 9일 센서를 차치한 자동제어를 위해 계속 새로운 연결을 시도해본 결과 다음과 같이 연결하였을 때 제어가 가장 안정적으로 되는 것을 발견하였다.



<그림 27> 최종 제어부 연결

처음의 연결과 다른 점은 기판과 변속기, 수신기, 서보모터에 연결되는 전원을 공통의 전원과 그라운드를 연결해 주는 것이다.

11월 10일 따로 센서테스트용 기판을 만들어서 센서가 제대로 작동한다는 것을 확인한 뒤, 기존의 기판에 센서를 연결하여 RS232를 통해 센서값을 읽어 보았다. 그 결과 이상하게도 센서값이 제대로 읽히지 않았다. 그 원인은 아직도 파악하지 못하였다. 그래서 테스트용 기판에 약간의 커넥터와 연결을 추가하여 컨트롤러로 만들었다. 최종적으로 완성된 이 기판에서는 센서값을 잘 읽어 들이면서도 서보모터와 변속기의 제어가 완벽하게 이루어 졌다. 이제 적당한 센서 수치만 넣어주면 되는 단계에서 갑자기 기판이 작동하지 않았다. 제대로 작동 되던 기판이 아침 8시 30반에 작동이 되지 않아 당황스럽지 않을 수가 없었다. 어쩔 수 없이 프로그램과 연결을 처음부터 다시 해보았지만 센서값이 읽히지 않았다. 한참을 씨름하다가 결국 기판에는 문제가 없고 RS232가 갑자기 고장 나서 센서값이 PC에 읽히지 않았을 뿐 센서를 포함한 자동제어는 제대로 작동하고 있었다. 컨테스트까지 시간이 얼마 남지 않아서 과전 앞 복도에서 가상의 테스트를 해본 후 정확한 수치를 입력하지 못하고 컨테스트에 나갔다.

### 5.3.2. 2차 Contest 결과 및 향후 과제

1차 Contest의 우수한 결과가 시운전을 통한 최종 조정 작업에 있었다면 2차 Contest의 실망스러운 결과는 자동제어를 통한 시운전의 수행을 해보지 못한 부분에 있었다고 판단된다.

알고리즘은 크게 문제가 없었다고 생각하지만 실제 조타각에 대한 선회 반경을 측정하지 못한 상태에서 ‘적당히’ 수치를 대입한 탓에 실제 Contest에서 너무 과도한 반응을 보여 실패한 것이다. 2차 시도에서 조타각의 변화를 줄여서 다시 시도하였으나 그 조타각도 벽면과의 거리를 유지하면서 주행하기에는 너무도 큰 것이었다. 여러 차례 시운전을 통해서 피드백을 통해 최적 수치를 찾아야 함에도 시간의 제약으로 인해 이를 수행하지 못한 것은 가장 큰 패인으로 분석된다.

우리 조뿐만 아니라 모든 조들의 Contest 모습을 지켜보면서 제대로 된 무인 제어 주행을 위해서는 추가적인 장치의 보완 및 여러 이론 공부도 병행되어야겠구나 하고 생각했다.

먼저 제어 이론에 대한 공부이다. 여러 이론들이 있겠지만 PID 제어 이론에 대한 공부가 필요하다고 판단된다. 단순히 상한, 하한 값을 설정해놓고 양 극단에서만 러더의 방향을 수정하는 것으로는 안정적인 주행이 힘들다. 따라서 시스템의 90%이상을 제어하는데 쓰이는 방식인 PID 제어에 대해 공부를 좀 하여 러더의 각을 조정하는데 활용할 수 있을 것이다.

또한 거리 측정 센서만으로는 제어가 힘들 수밖에 없다. 선수의 방향과 진행 방향의 속도 및 회전 속도 등이 안정적 제어를 위해서는 필수 Input이라고 할 수 있다. 변속기 제어를 통해 속력은 일정하게 유지한다고 해도 선수의 방향과 회전 속도 등은 거리 측정 센서만을 가지고는 알기 힘들다. 측면에 센서 2개를 장치한다고 하더라도 초음파 센서의 특성상 그 정확한 방향을 구하기란 힘들다. 따라서 추가적으로 방향 제어에 이용할 수

있는 자이로 센서 등을 이용해 볼 수 있을 것이다. 물론 이의 사용은 초음파 센서의 제어보다는 훨씬 힘든 작업이 될 것이다. 하지만 정말 노력과 의지만 있다면 도전 해볼만한 과제라 생각한다.

## 6. 역할분담 및 세부일정

### 6.1. 역할분담

<표 8> 역할 분담

선체부	선형설계	김두용, 김영현
	구획배치, 도면생성	김두용, 김영현
	Nesting&cutting	이주현, 심훈섭, 노재욱
	선체조립	이주현, 심훈섭, 노재욱
	선체마감작업	모두
제어부	회로설계	이주현, 심훈섭, 노재욱
	회로제작	이주현, 심훈섭, 노재욱
	무인주행프로그래밍	김두용, 김영현
시험주행&조정		모두

## 6.2.세부일정

<표 9> 세부일정

		10.11					10.28					11.11				
선체부	선형설계															
	구획배치, 도면생성															
	Nesting&cutting															
	선체조립															
	선체마감작업															
제어부	회로설계															
	회로제작															
	무인주행프로그래밍															
시험주행&조정																