

2019지능형모형차 경진대회 보고서

학 교	단국대학교
팀 명	단보르기니
유 형	본 경기
팀 장	이진호 (전자전기공학부)
팀 원	고승일 (전자전기공학부) 박찬희 (전자전기공학부) 이장훈 (전자전기공학부) 김원석 (전자전기공학부)

1. 개요

1.1 설계 배경

4차 산업 혁명의 시작과 동시에 AI가 많은 관심을 받기 시작했다. AI를 다양한 방면으로 사용하기 위해 많은 기술들이 개발되고 있는데 그 중 대표적인 기술이 자율주행 시스템이다. 자율주행 자동차란 운전자의 조작이 없이 스스로 운행이 가능한 자동차를 말한다. 현재 많은 기업들이 이미 부분적인 자율주행 시스템을 상업화한 단계이며 5G 시대가 다가옴에 따라 클라우드를 이용한 원격 조종 등 다양한 가능성을 보이기 시작했다. 이에 자율주행 자동차를 만들면서 기존의 기술과 새로운 기술을 접목시키면서 공학인으로써 한발 나아갈 수 있는 기회가 될 거라고 생각했다.

1.2 설계 목표

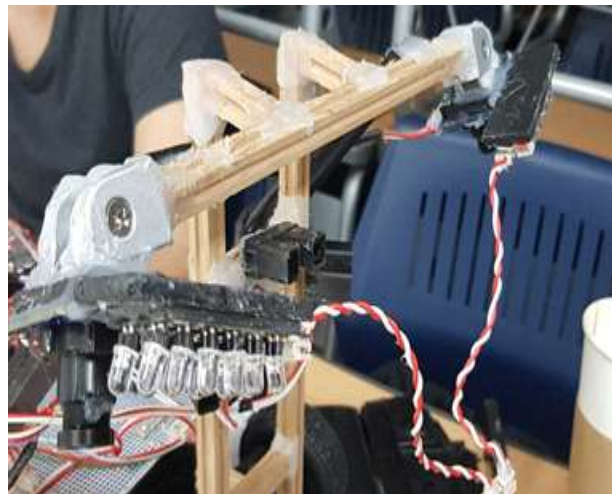
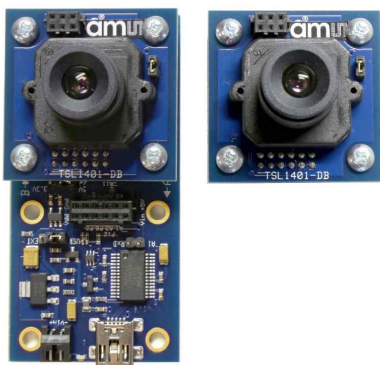
- 팀원 간에 역할 분담을 바탕으로 개개인의 역량을 발휘할 수 있도록 제작을 진행
- Interrupt Scheduling을 통해서 안정적인 시스템 구현
- Line Scanning Camera 와 거리 센서에 Median Filter를 적용하여 안정적인 ADC값 추출
- Position 방식을 이용하여 곡률에 따른 좌, 우측 추종 정도 결정
- 속도 제한 구간(School Zone)에서 장애물 감지를 통한 차선 변경
- 정확한 속도 제어로 장애물과 충돌하지 않고 정지(AEB 구현)

2. 설계 내용

2.1 하드웨어 구성

2.1.1 Sensor

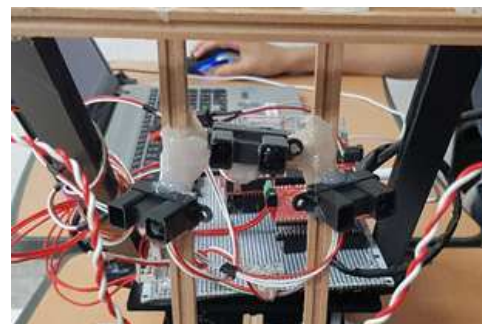
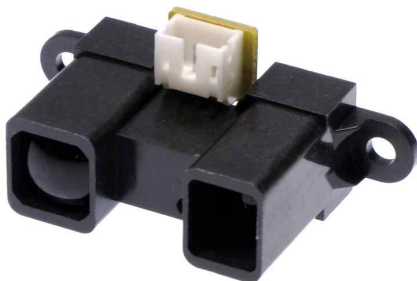
TSL1401-DB (Line Scan Camera)



<그림 2-1> TSL-1401-DB(좌) 와 차량에 장착된 모습(우)

차선을 검출하기 위해서 TSL1401-DB 2개를 사용하였다. 각 센서당 왼쪽, 오른쪽 라인을 검출하며 TSL1401-DB는 밝기에 따라 값이 변하는 센서인 만큼 빛의 변화에 민감한데 이를 대비해 LED 조명을 달아 모든 환경에서 작동할 수 있도록 하였다. 또한 센서가 값을 받을 때 센서값의 잡음을 처리하기 위해서 OPamp를 이용하여 처리를 하였다.

GP2Y0A02(적외선 거리센서)



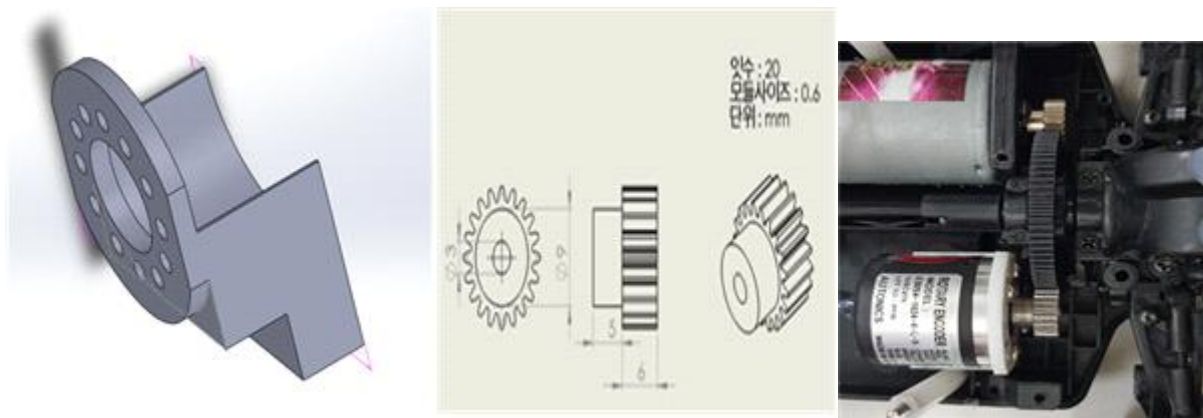
<그림 2-2 > GP2Y0A02(좌) 와 차량에 장착된 모습(우)

SHARP 사의 2Y0A02 적외선 센서를 사용하였다. 언덕 구간에서의 오작동을 방지하기 위해 지상에
서 약 25cm 높이로 설치하였고 직진구간에서의 장애물 검출 및 AEB를 위해서 정면으로 1개, 곡선 구
간에서의 장애물 검출을 위해 양 옆으로 30도 정도 휘어서 오른쪽 1개, 왼쪽 1개 총 3개를 사용하였다.

2.1.2 구동부



<그림 2-3> RS540SH(좌) 와 IFX007T(우)



<그림 2-4> 엔코더 기어, 엔코더 가드, 엔코더 장착된 모습

모터는 RS540SH로 기본적으로 제공되는 모터를 사용하였고, 모터드라이버는 IFX007T 모듈을 사용
하였다. 엔코더는 Autonics 사의 E30S4-1024-6-L-5의 모터에 장착된 기어에 맞춰서 모듈사이즈
0.6 기어를 제작하여 장착했고, SolidWorks로 엔코더가드를 제작하여서 3D프린터로 출력하여 차량에
장착하였다.

2.1.3 조향부



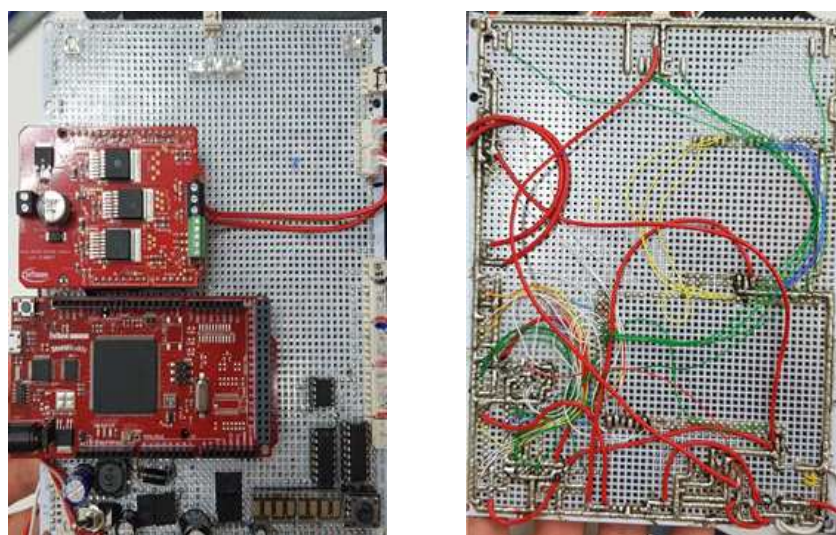
<그림 2-5> 조향부 서보모터

서보모터는 기존의 차량에 부착되어 제공되는 모델을 사용하였다.

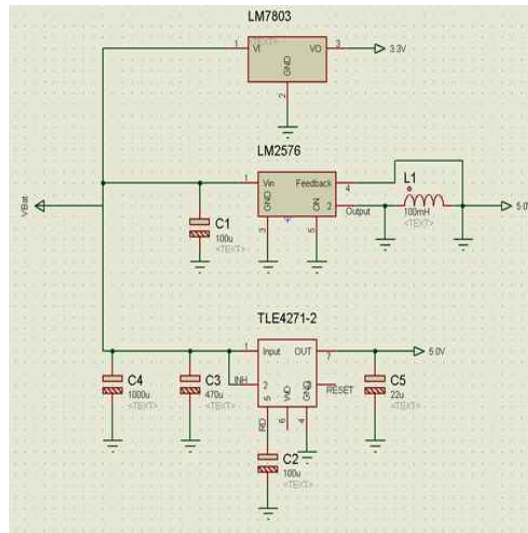
2.1.4 회로

전원부 구성은 간섭이 일어나지 않도록 하기 위해 센서부, 센서 인식을 위한 LED, 모터부 전원 레귤레이터를 따로 구현하였으며, 센서부는 LM2576, 모터부는 Infineon사의 TLE-4271-2, LED는 LM7803을 사용하였다. 회로 구성은 여러 상황에 대비하기 위해 만능기판에다가 납땜을 이용해서 구현하였다.

LM2576은 PWM 방식이기 때문에 발열설계를 하지 않았고, TLE-4271-2와 LM7803은 주변온도 30℃에서 작동한다는 가정 하에 발열 설계하여 방열판을 장착했다.



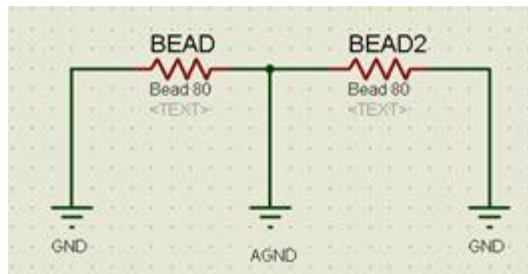
<그림 2-6> 메인보드 전면부, 후면부



<그림 2-7> 전원부 회로도

2.1.4.1 센서부

LM2576을 이용하여 상호 간섭을 배제하고, Ripple Noise를 최소화하기 위해서 커패시터와 인덕터를 이용해서 전원부 설계를 하였고, GND의 간섭을 최소화하기 위해 Bead저항을 이용하여 Analog Ground(AGND)를 만들어서 잡음을 최소화 하였다.



<그림 2-8> AGND 회로도

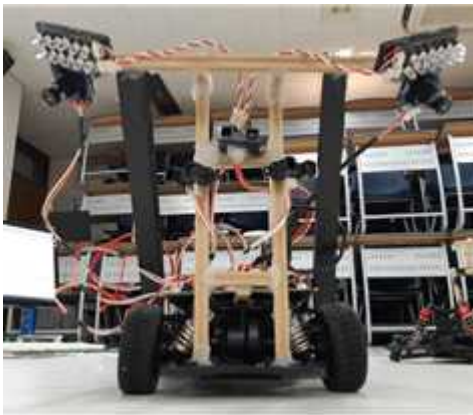
2.1.4.2 모터부

잡음 제거를 위해 GND를 가능한 한 넓게 깔았으며, RS540SH 모터가 저전압 고전류 제어를 하기 때문에 안정적인 전류 공급을 위해 18AWG 규격의 두꺼운 테플론 선을 사용하였다.



<그림 2-9> 모터부 점프선

2.1.5 전체 외관



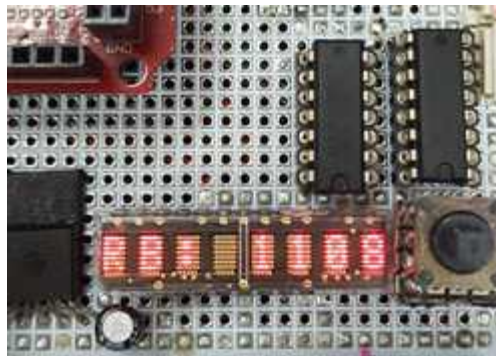
<그림 2-10> 차량 전면부와 측면부

차량 크기 규정에 맞추어서 크기를 가로 300mm, 세로 450mm, 높이 300mm 이내로 설계하였다.

2.2 소프트웨어 구성

2.2.1 Debugging Environment

자율주행 알고리즘 개발의 편의를 위하여 Infineon사에서 제공하는 iLLD를 기반으로 HCMS-2912(VFD)를 이용해서 디버깅 환경을 구축하였다. 별도의 통신을 하지 않고, 초기 설정에서 메뉴를 구현하여 센서, 엔코더, 라인, 서보모터, 속도 등을 관리 할 수 있게 하였다.



<그림 2-11> VFD 사용하는 모습

2.2.1 Interrupt Scheduler

자율주행 알고리즘을 개발하면서, 가장 핵심이 되는 부분은 Interrupt Scheduler 설계인데, 인터럽트를 1ms 주기로 하면서 구동부, 조향부, 센서부 제어를 안정적으로 하면서 다음 인터럽트에 영향을 미치는 Pending 현상을 방지하기 위해서 처리 시간까지 감안하여 자체적으로 설계를 하였다.

Time	Work
0ms	Line Detecting, Motor Control, Distance Detecting
1ms	Line Sensor Value Processing, Motor Control, Distance Detecting
⋮	⋮
29ms	Motor Control, Distance Detecting
30ms	Motor Control, Distance Detecting
31ms	Line Detecting, Motor Control, Distance Detecting
32ms	Line Sensor Value Processing, Motor Control, Distance Detecting

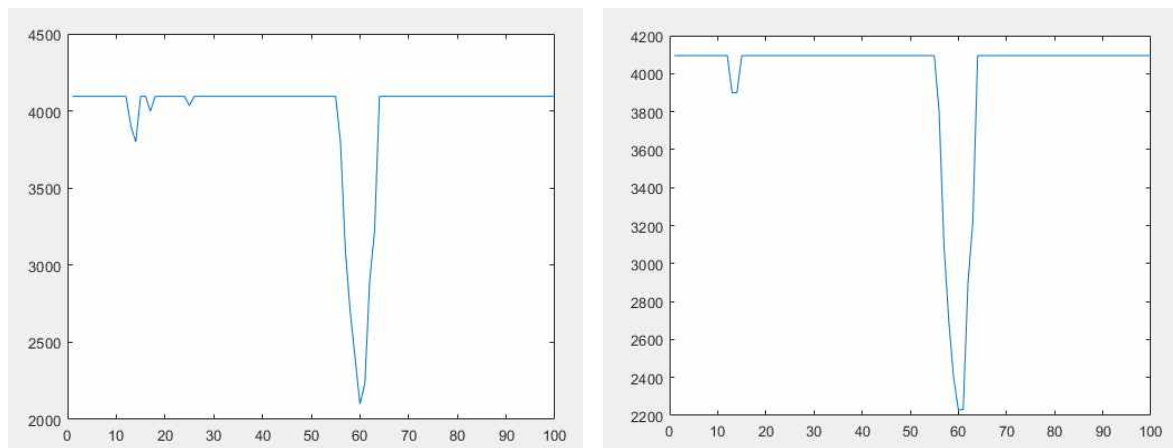
<표 1> Interrupt Scheduler

Line Camera가 값을 정확히 받게 하기 위해서 30ms마다 값을 받게 하였고, 보다 정밀한 제어를 위해 값을 받고난 이후에 센서값을 처리해서 라인을 추출하는데 사용하였다. 모터와 거리센서 값은

1ms마다 처리하여서 정밀한 속도 제어와 정확한 장애물 회피를 할 수 있게 하였다.

2.2.2 Camera Signal Processing

Line Camera 로 받은 128개의 Pixel 값을 Median Filtering을 통해서 Ripple Noise를 제거 하였다. 검은색과 흰색의 명확한 값 차이를 주기 위해서 라인을 놓치지 않는 선에서 센서값을 주는 시간에 일정 시간 Delay를 주어서 보다 명확하게 값을 인식하게 하였다. 환경에 따라서 값이 달라질 수 있는 가능성을 고려하여 128개의 Pixel 중에 가운데 100개의 Pixel을 추출했다.



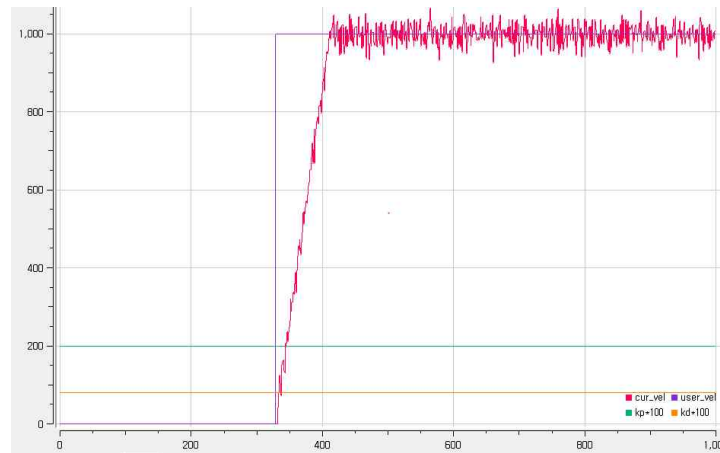
<그림 2-12> Median Filter 적용 전, 후

2.2.3 Line Detecting

Line Camera를 이용하여 받은 값을 이용해서, 100개의 Pixel에 주어진 값들 중에 최댓값(라인 외)과 최솟값(라인)을 추출하여 그 값을 127등분 하여 일정 수준이하이면 라인, 이상이면 라인이 아니라는 기준을 세웠고, 100개의 Pixel을 20개의 Sector로 나눠서 검출된 라인이 각 센서에서 어느 Sector 잡히는 지 인지 할 수 있게 하였다.

2.2.4 속도 제어

속도제어를 위하여 PID 제어를 선택하였다. PID 제어는 현재 값과 추종 값의 오차를 비례-적분-미분 과정을 통하여 더 빠르고 유연한 제어가 가능해진다. 엔코더로 측정된 모터 회전수를 기어비와 바퀴 반지름을 통해서 얻은 상수로 계산을 통하여 제어 주기를 바탕으로 속도로 환산하여 사용했다. 보다 빠른 추종을 위해서 가속도 개념을 추가하였다. 위의 그림은 1000mm/s 의 속도를 2000mm/s^2 의 가속도로 추종하는 그래프이다. 정확히 0.5초 만에 추종하는 모습을 볼 수 있다.

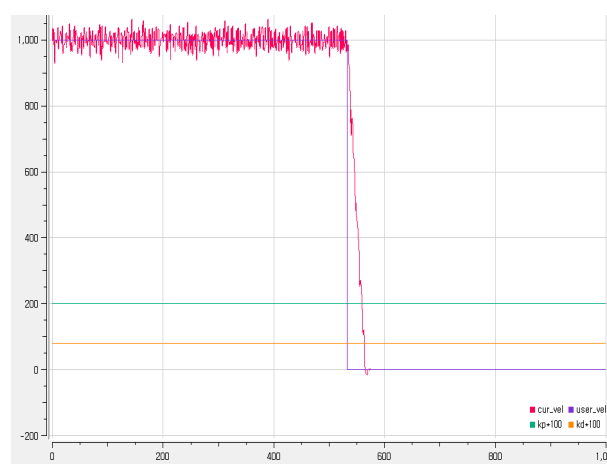


<그림 2-13> PID제어를 통한 속도 추종

또한, 모터를 제어할 때 PWM 값을 SR0, SR1 레지스터 값의 변경을 통해서 자체적으로 PWM의 duty Cycle을 결정하여 모터의 양쪽 핀에 On, Off 방식으로 적용하여 한쪽에만 pwm이 들어가도록 하여 방향 제어를 하였다. 엔코더의 값도 보다 정밀하게 하기 위해서 Resolution을 66536으로 설정하여 사용했다.

2.2.5 AEB

AEB의 경우 속도와 가속도 제어를 이용해서 구현하였다. 거리 센서로 장애물을 인식하고 나서 1ms주기로 돌아가는 모터 제어 함수를 통해서 속도를 가속도만큼 계속 감속시켜서 0mm/s까지 감속시키고 모터 출력을 차단한다.



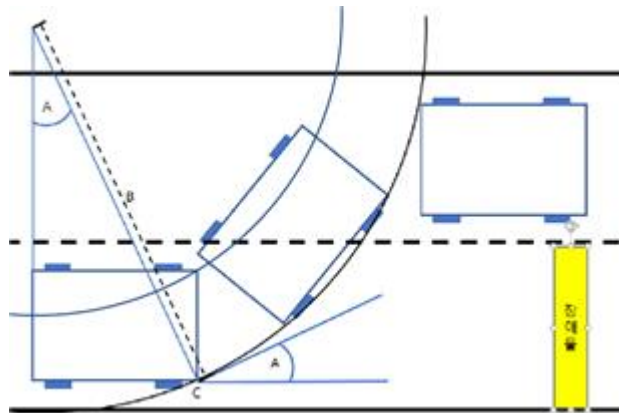
<그림 2-14> AEB를 통한 정지 속도 추종

2.2.6 School Zone Detecting

횡단보도의 경우에는 라인만 검출 할 때 보다 Pixel의 개수가 많아지고, 그것을 기준으로 하여서 라인을 검출할 때 보다 라인이라고 인식된 Sector의 개수가 많을 때 School Zone으로 인식하도록 하였고, 똑같은 상황이 발생하였을 때 School Zone을 벗어나도록 구현하였다.

2.2.7 장애물 회피

장애물을 인식하는 거리를 에커만 조향 방식을 이용해서 벗어날 수 있는 최소한의 거리를 계산하고, 그 거리보다 조금 더 여유로운 값을 주어서 안정적인 회피 주행이 가능하도록 하였다, 이에 더해, 조향부의 더 정밀한 제어를 위해 각도단위보다는 서보모터의 최소 펄스를 최소단위로 해서 duty cycle을 결정했다.



<그림 2-15> 장애물 회피 시뮬레이션

장애물을 인식하고 나서 Interrupt Counting을 이용하여 일정 시간동안 장애물이 없는 차선으로 꺾고 다시 라인으로 돌아오기 위해 Interrupt Counting을 사용하여 일정시간동안 중앙으로 돌아온다. 회피 주행을 하는 동안에 거리센서를 이용해서 장애물이 있는지 없는지 확인을 하면서, 벗어나고 난 후에 라인을 다시 인식하면서 조향한다.

2.2.8 Lane Tracking

필터 처리된 100개의 Pixel값들은 20개의 Sector로 나누어서 라인을 검출하였고, 각 Sector마다 가중치(무게값)를 주어서 라인을 따라 조향할 수 있게 하였다. 무게값은 실제로 라인카메라가 인식하는 Pixel간 거리를 측정하여서 에커만 조향이론을 바탕으로 결정하였다.

3. 주요 장치 이론 및 적용 방법

3.1 전원부 레귤레이터 방열 설계

$$\theta_{ja} = \theta_{jc} + \theta_{ca} + \theta_{sa} \cdots \cdots \text{식 (1)}$$

$$\theta_{ja} = (T_j - T_a) / P_d \cdots \cdots \text{식 (2)}$$

$$T_j = (\theta_{jc} + \theta_{cs} + \theta_{sa})P_d + T_a \cdots \cdots \text{식 (3)}$$

(θ_{ja} : 반도체접합부에서 대기까지의 열 저항, θ_{jc} : 반도체 접합부에서 디바이스 케이스까지의 열 저항,

θ_{ca} : 디바이스케이스에서 대기까지의 열 저항, θ_{cs} : 절연 시트지 저항(통상적으로 1.5°C/W),

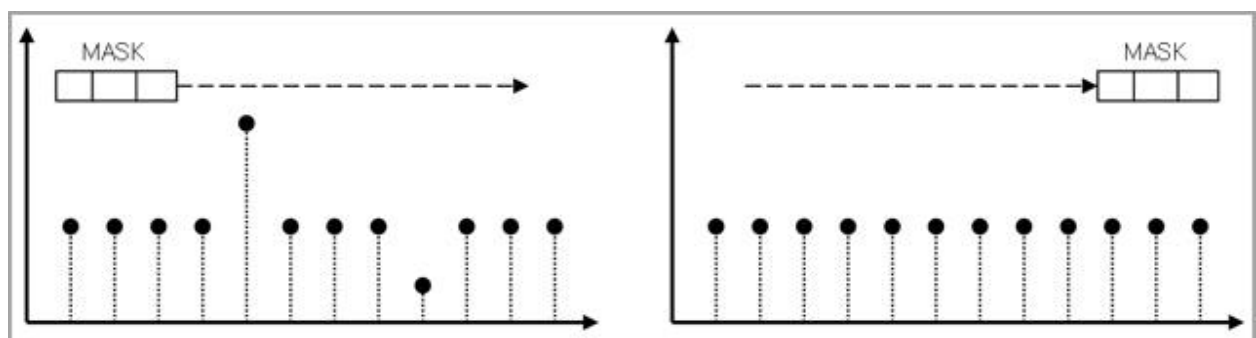
θ_{sa} : 방열판의 열 저항(소자 사용 조건 충족 요건), P_d : 소자가 열로 소비하는 전력(W),

T_j : 반도체소자의 접합부 한계 온도, T_a : 반도체 소자를 사용하는 실제온도)

TLE-4271-2는 최대 0.5A, 평균적으로 0.3A정도의 전류가 흘렀고, LM7803에는 평균적으로 0.5A가 흘렀는데 DataSheet를 참고해 각 상수를 얻어서 식(3)을 바탕으로 소자가 견딜 수 있는 한계 온도를 계산하여 방열판을 TLE-4271-2에는 소형 방열판 1개, LM7803에는 중형 1개, 소형 1개를 장착했다.

3.2 Median Filter(메디안 필터)

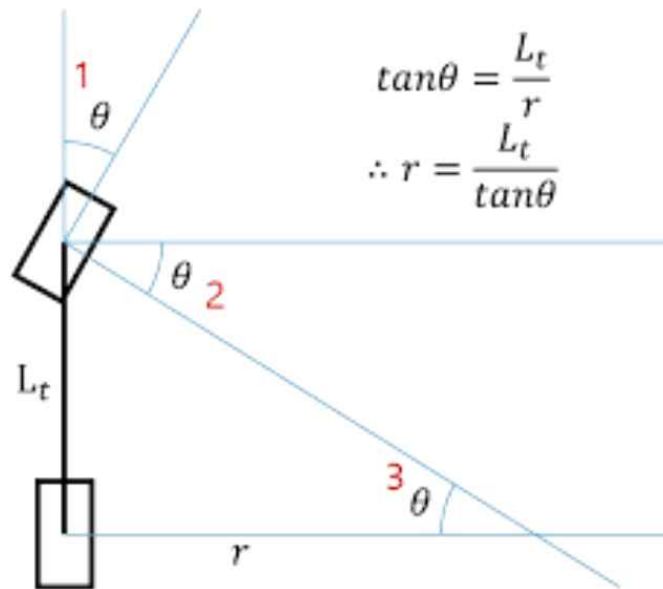
미디언 필터(Median Filter)는 주어진 마스크 영역의 값들을 크기 순서대로 정렬한 후 중간 크기의 값을 선택하는 필터이다.



<그림 3-1> Median Filtering

3.3 Ackerman-Jantoud tupe (에커만 조향 이론)

무게값을 통해 계산된 가중치(포지션)을 통해서 현재 위치에서 어느 정도 움직여야 하는지 최소 이동 거리와, 최소 조향각을 계산하여서 라인을 검출할 때 마다 계산하여 Line Tracking을 구현하였다.



4. 결론 및 토의

4.1 결론

다음과 같이 설계 목표를 달성하였다.

- 센서 제어, 모터 제어를 원하는 시간에 작동하도록 Interrupt Scheduler를 구현하였다.
- Line Camera의 Pixel을 Sector를 나눠서 주어진 무게값을 통해서 Line Tracking을 한다.
- 엔코더 값을 통해서 원하는 속도에 PID제어를 통해서 도달할 수 있다.
- 장애물 인식을 거리 센서를 통해 하고, 인터럽트 시간을 통해서 정확하게 회피한다.
- PID제어를 통해서 스쿨 존에서 가감속, 장애물을 만났을 때 감속을 할 수 있다.

4.2 토의

개발을 진행하면서 가장 중요한 것이 센서라고 생각하였고, 센서를 Filtering 하기 위한 다양한 방법들을 고민해보았다. 하드웨어 적인 필터링 방식과 소프트웨어적인 필터링 방식이 있었지만, 스케줄링과 연산량을 감안해 보았을 때 Median Filter를 선택하였지만 다음 기회에는 다른 필터링 방식도 써보고 싶다.

또한 대회장이든 테스트하는 곳이든 환경에 따라서 변하지 않는 구동을 하는 자율주행 자동차를 만들고 싶었는데 LED와 센서 커팅을 통해서 구현을 하고자 했지만 그래도 환경의 영향을 무시하지 못했던 것 같아 아쉽다.

지능형 모형차를 제작하면서 소프트웨어와 하드웨어의 결합이 생각보다 어렵다는 것을 많이 깨달았으며, 아무리 소프트웨어를 잘 구성해도 하드웨어가 받쳐주지 않으면 의미가 없고, 하드웨어를 아무리 안정적으로 구성해도 소프트웨어가 비효율적이라면 의미가 없다는 것을 깨닫게 되었고, 그로 인해 하드웨어적으로든, 소프트웨어 적으로든 완성된 작품을 만들고자 하였다. 그러한 과정 속에서 서로 의견충돌이 많았지만 최종적으로 완성이라는 목표를 향해 열정을 가지고 와준 팀원 모두에게 감사한다.

[첨부양식 1]


전자 부품 목록

제조사	부품명	수량	사용목적
AMS	TSL1401EVALKIT	2	차선 및 트랙 검출
SHARP	GP2Y0A02	2	장애물 검출
Mabuchi Motor	RS-540SH	1	차량 구동용 모터
Autonics	30S4-1024-6-L-5	1	모형차 속도 제어
플랫폼에 포함	Servo Motor	1	모형차 조향 제어
Infineon	TLE-4271-2	1	모터제어용 5V 레귤레이터
Infineon	ShieldBuddy	1	메인 보드
TEN-POWER	LM7803	1	LED-용 3.3V 레귤레이터
On-Semiconductor	LM2576	1	센서부 5V 레귤레이터
Broadcom	HCMS-2912	1	메뉴 조작용 VFD
Texas Instruments	AN26LS32	1	엔코더 보호용 소자
Texas Instruments	LM358	1	센서 OPamp
Texas Instruments	SN74LS08	1	클럭 발생용 AND게이트
Infineon	IFX007T Module	1	모터제어용 모터드라이브

[첨부양식 2]

차체 변경 목록

*변경 내용 및 변경 사유

 <p>엔코더 장착대를 설치하기 위해서 구동 기어 덮개를 일부 절단하였고 차체를 일부 절단하였다. 엔코더 장착대를 고정하기 위해서 바닥에 구멍을 뚫었다.</p>	<ol style="list-style-type: none"> 1. 변경사유 엔코더 장착을 위한 차체 변형 2. 변경 근거 조항 엔코더 장착을 목적으로 차체 일부를 절단하는 것은 허용하지만 그 이외의 목적으로 플랫폼을 절단하거나 변형하는 것을 금지한다.
 <p>메인보드 지지대를 설치하기 위해서 차체 바닥에 구멍을 뚫었다.</p>	<ol style="list-style-type: none"> 1. 변경사유 메인보드 지지대를 장착을 위한 구멍 뚫음 2. 변경사유 근거 조항 부품 장착을 목적으로 플랫폼에 구멍을 뚫는 것은 허용되지만 그 이외의 목적으로 플랫폼을 절단하거나 변형하는 것을 금지한다.
	<ol style="list-style-type: none"> 1. 변경사유 차체 길이 규정 준수를 위한 차량 후미부분 제거 2. 변경 근거 조항 http://race.acelab.org/community/notice.php?pagenumber=1&at=view&idx=371