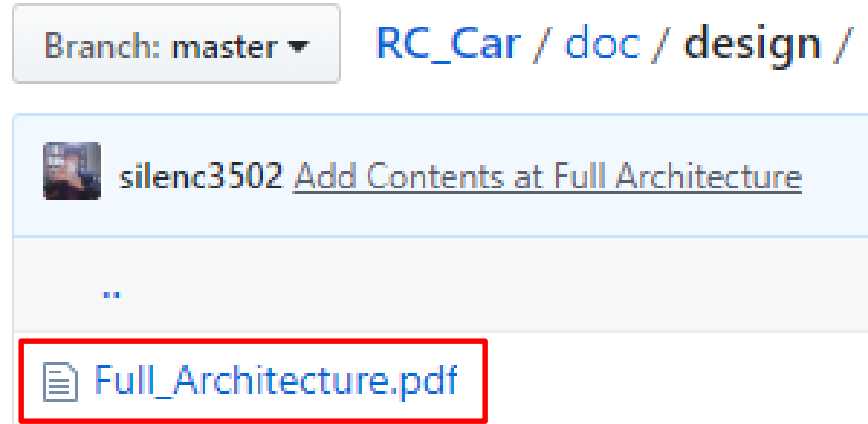


# Xilinx Zynq FPGA, TI DSP, MCU 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee(이상훈)  
[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

# ***Current Mission Progress***


## 전체 아키텍처에 대한 설계 문서











전체 구성에 대한 아키텍처를 이해하기 위해 만든 문서임  
이 문서를 통해 본인이 뭘 해야 하는지 감을 잡길 바람

## DC-DC 컨버터 개발과 관련한 작업들

Branch: master ▾ RC\_Car / circuit / dcdc /

 silenc3502 Merge pull request #32 from HyunwooParkk/master ...

..		
 <a href="#">BUCK CONVERTER .pdf</a>		BUCK CONVERTER
 <a href="#">LC Resonance.pdf</a>		LC Resonance(LC 공진) Circuit
 <a href="#">PSIM PID제어.pdf</a>		PSIM TOOL/PID
 <a href="#">PSIM STUDY 1.pdf</a>		PSIM TOOL STUDY
 <a href="#">PSIM STUDY 2.pdf</a>		PSIM TOOL/PID
 <a href="#">SMPS_PowerElectronics_1.pdf</a>		SMPS Power Electronics
 <a href="#">Switch-Mode Power Supplies.pdf</a>		switch mode power supplies - chapter 2
 <a href="#">prepare.txt</a>		Make DCDC Converter Data Dir

향후 PCB 관련 정보는 PCB 디렉토리에 들어갈 것임

## MCU, DSP, FPGA 관련 문서들

Branch: master ▾

[RC\\_Car](#) / [experiment](#) / [doc](#) /



silenc3502 FTDI USB2CAN

..



esp8266

esp8266-esp-01 datasheet



AM5728\_FTDI\_USB\_2\_CAN.pdf

FTDI USB2CAN



AM5728\_WiFi\_Lab.pdf

AM5728 Based Wi-Fi Lab



Ardu\_Based\_ESP8266.pdf

[Arduino Based Wi-Fi Module\(ESP8266\)\\_Test](#)



CAN\_Test.pdf

Dedicated Doc for Experiment



Configuring CAN TMS570.txt

Dedicated Doc for Experiment



Cortex\_R5\_I2C\_Howto.pdf

Cortex-R5F I2C Howto



DSP\_CAN(using\_Serial\_communication).pdf

How to use CAN2USB module in DSP



FreeRTOS\_Guide.pdf

FreeRTOS manual



HET\_PWM.pdf

HET Based PWM



Pmod\_CAN\_Control\_with\_Zybo.pdf

Pmod CAN Control with Zybo



RTI\_GPIO\_OC\_Config.pdf

RTI Based Common Emitter Circuit



SPI\_Comm.pdf

SPI Communication



SPI\_Loopback.pdf

SPI Lookback Exam



stepmotor\_test.pdf

How to work step motor in MCU




uart\_test.pdf











Dedicated Doc for Experiment

공공기관에서 나오시는 분들을 상대하기 위해 반드시 만들어야 하는 것들임

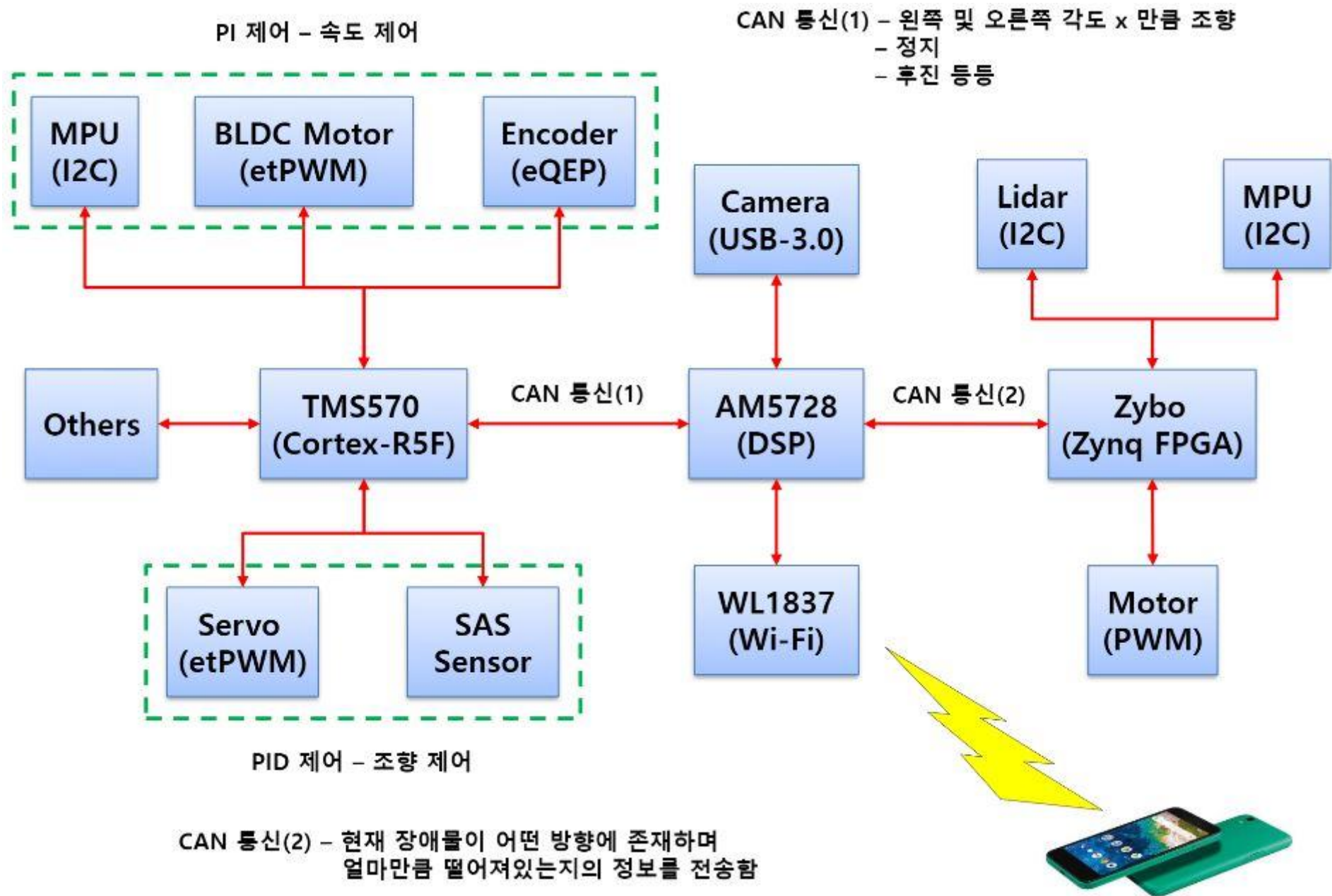
Branch: master ▾ RC\_Car / real\_test /

 silenc3502 NCS - 운영체제 커널분석

..

 prepare	adjust folder
 기술문서개발.txt	NCS - 기술문서개발
 오픈플랫폼활용.txt	NCS - 오픈플랫폼활용
 운영체제커널분석.txt	NCS - 운영체제 커널분석
 임베디드시스템테스팅.txt	NCS - 임베디드시스템테스팅
 펌웨어구현.txt	NCS - 펌웨어 구현
 펌웨어구현환경구축.txt	NCS - 펌웨어 구현 환경 구축
 펌웨어분석.txt	NCS - 펌웨어분석
 펌웨어설계.txt	NCS - 펌웨어 설계
 하드웨어분석.txt	NCS - 하드웨어 분석 시험 및 답안지

시험 문제와 답안을 전부 올려놨으니 대충 60 점만 넘기도록!!!  
나중에 몇몇 사람들 선별해서 인터뷰도 존재함!



영상 신호 처리

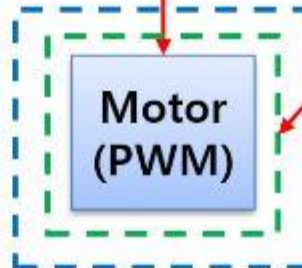
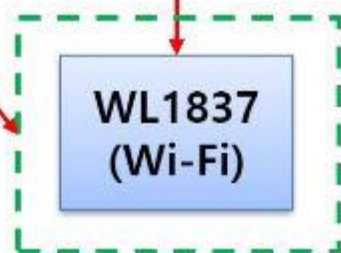
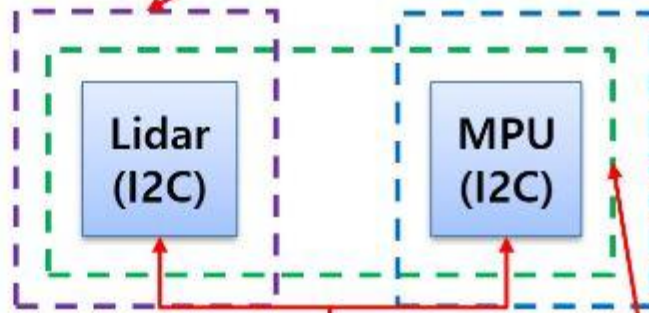
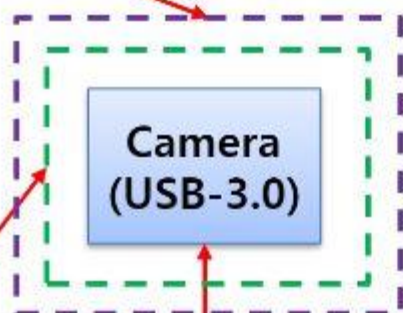
레이더 신호 처리

Device Driver,  
시스템 프로그래밍

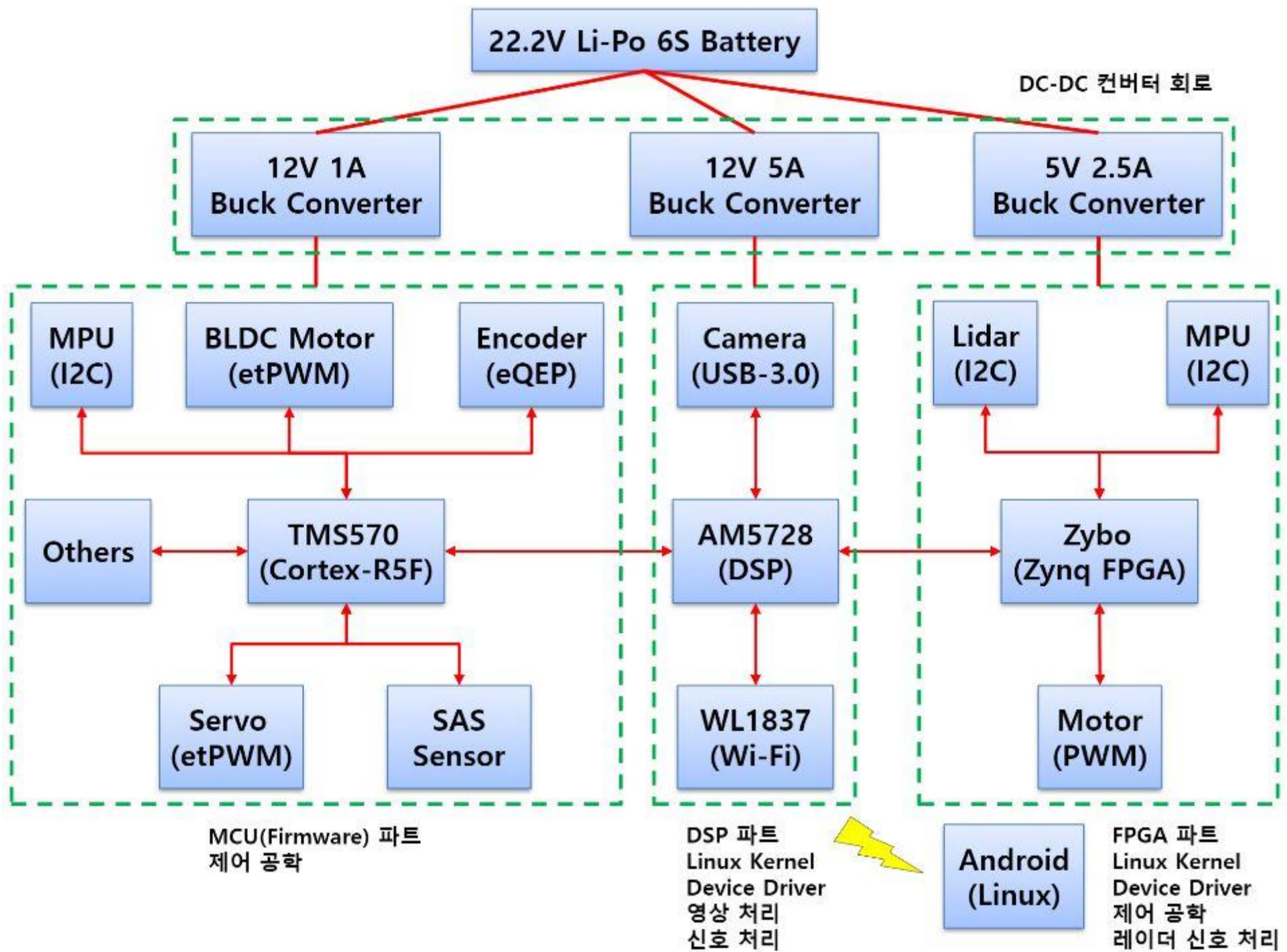
Device Driver,  
시스템 프로그래밍

제어 공학

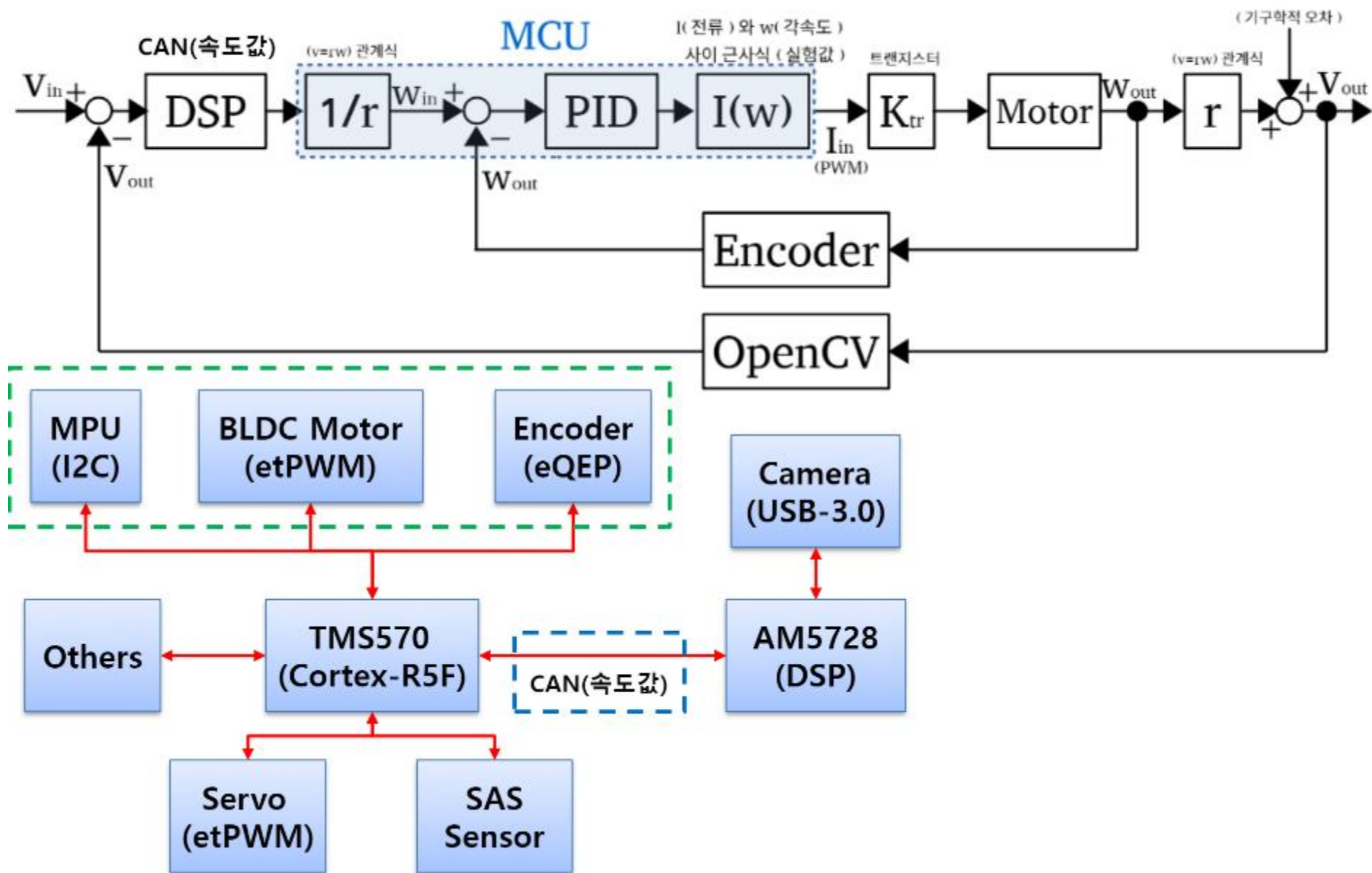
안드로이드 앱



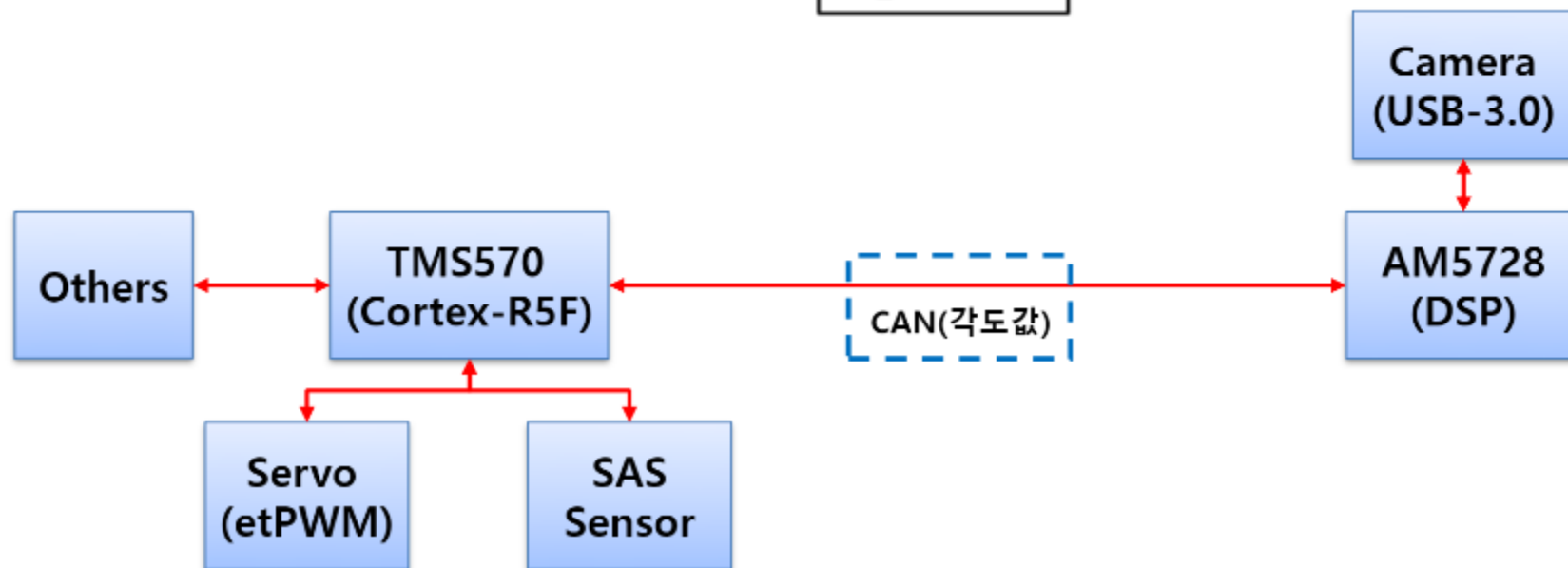
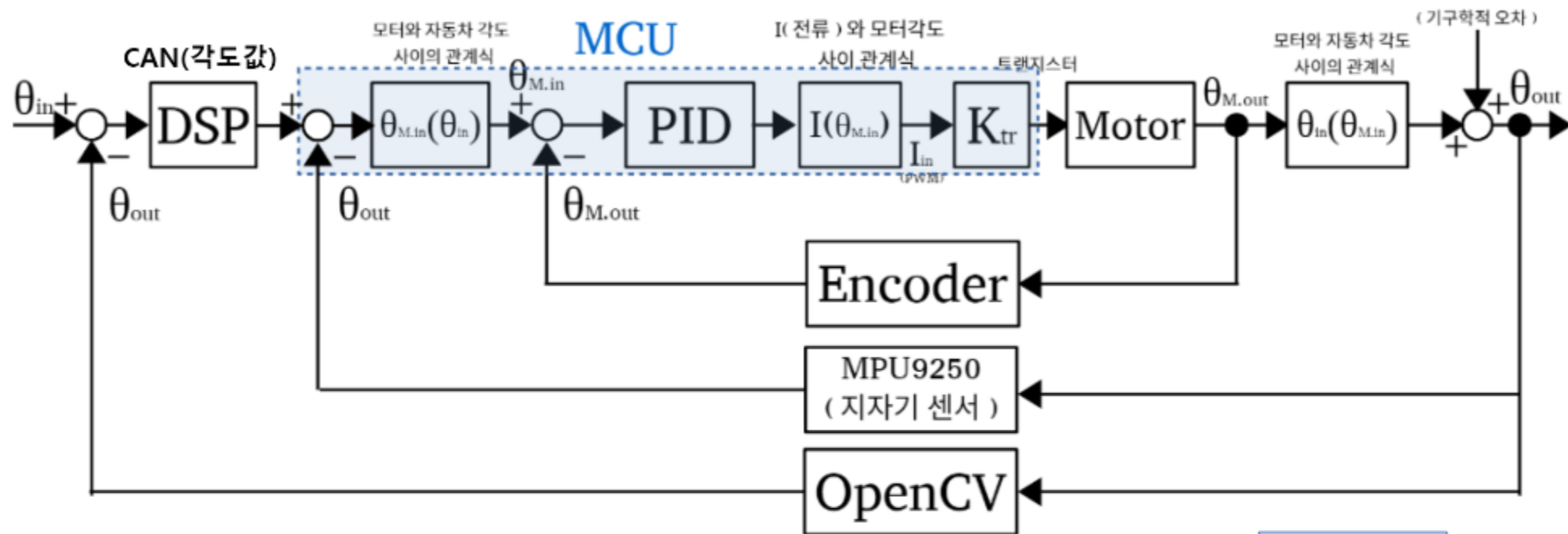




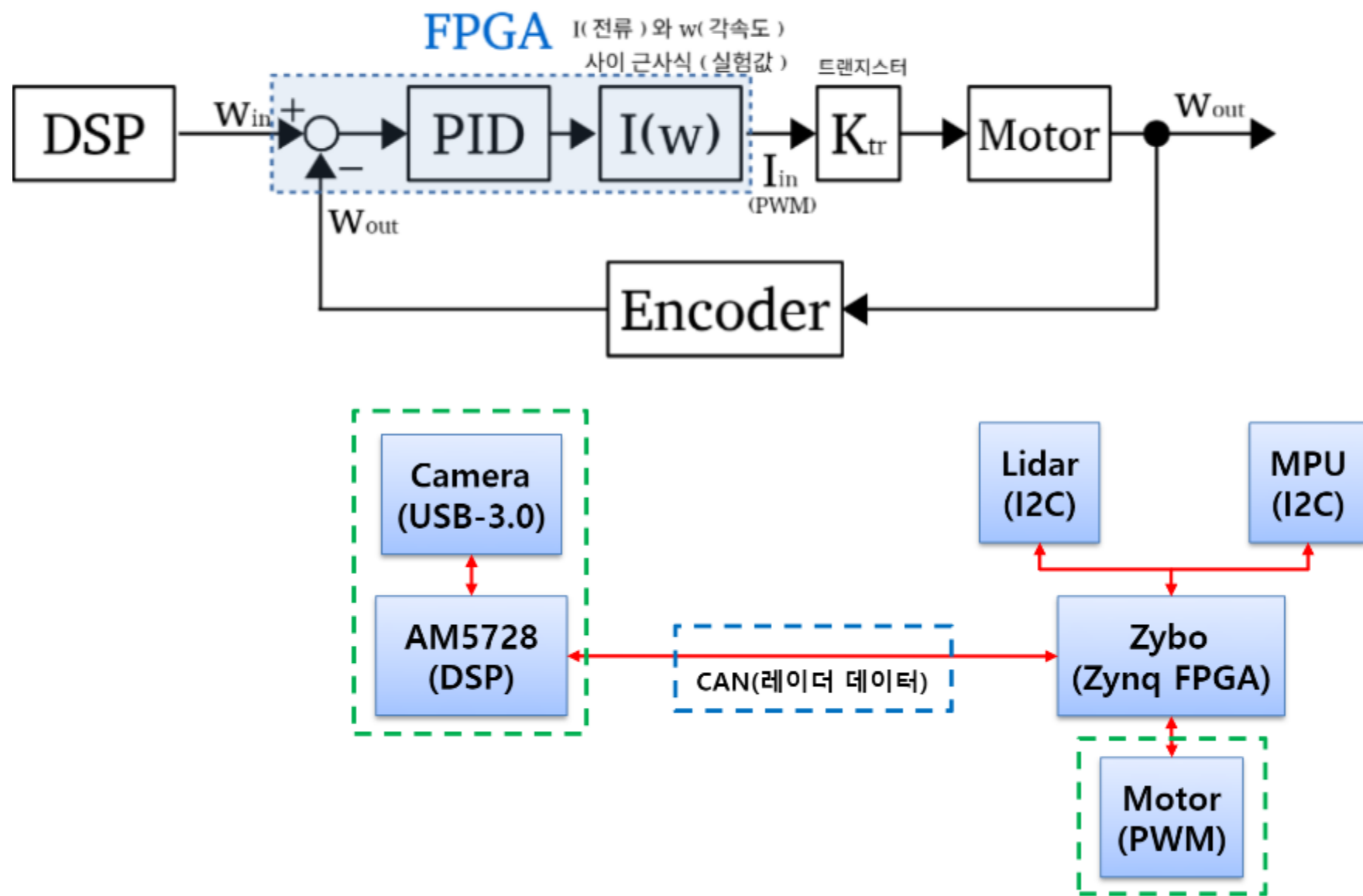
## 자동차 속도 PID 제어



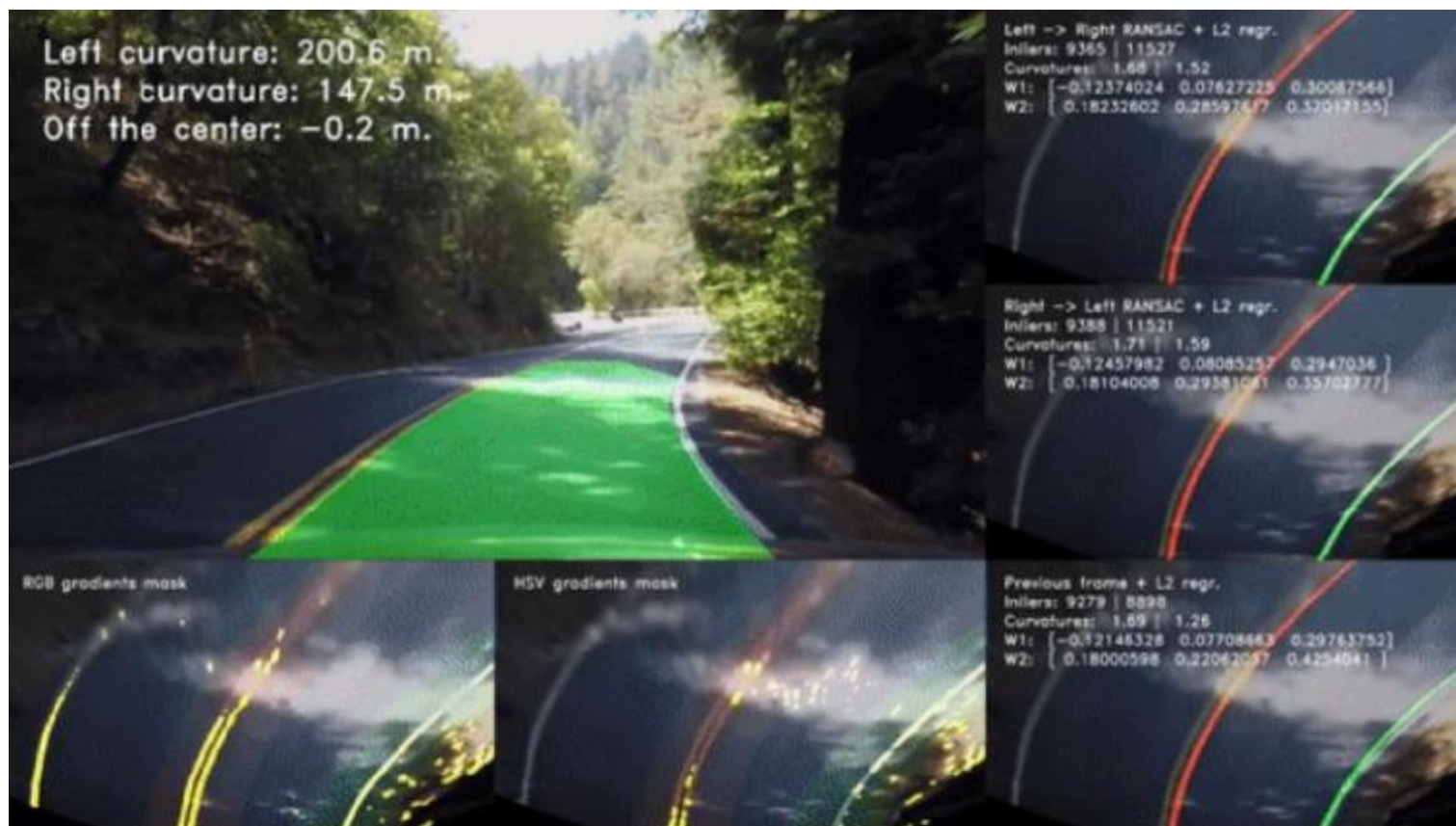
# 자동차 조향 PID 제어



# 라이다 모터 PID 제어









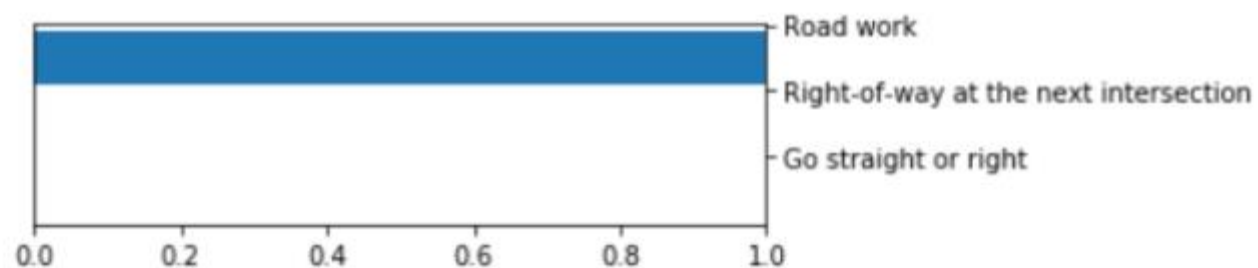
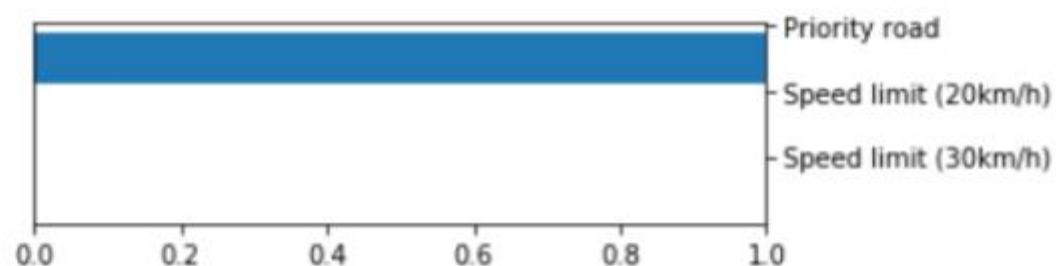
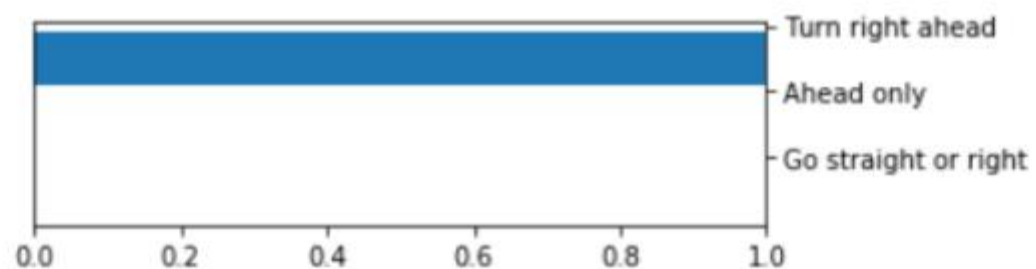
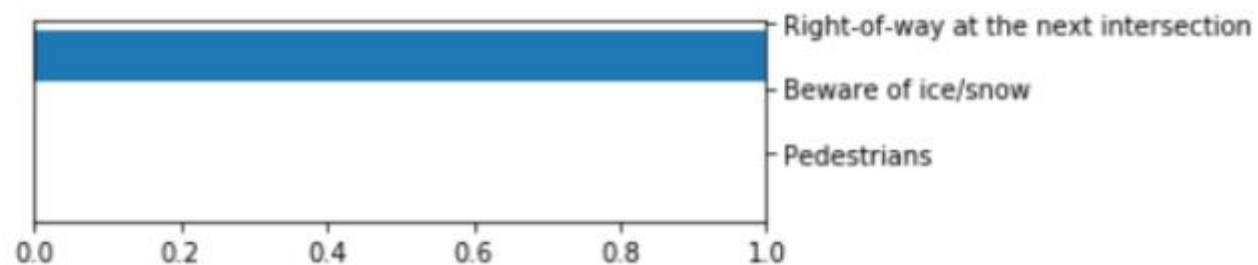
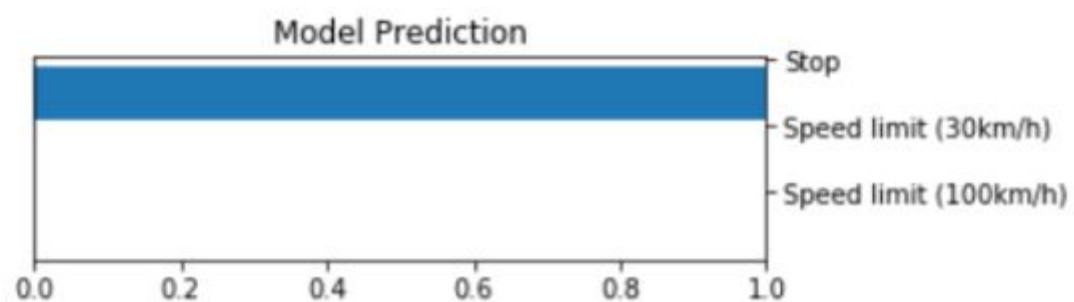
Left Camera Image



Center Camera Image



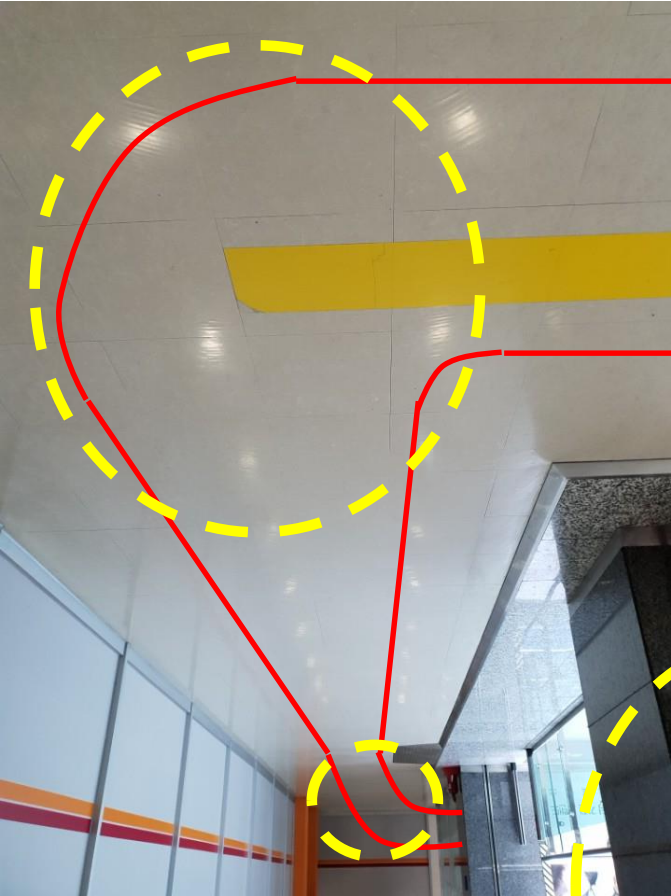
Right Camera Image



***Driving Track***



노란색원(급커브 구간)



# ***Lidar Scan Strategy***

# Lidar Specification

먼저 사용하려는 Lidar 의 스펙을 살펴보도록 한다.

아래는 사용하는 Lidar 의 데이터시트에서 발췌한 내용에 해당한다.

최대 사정 거리가 40 m 이며 정확도는 5 m 미만 구간에선 2.5 cm 의 오차를 가지고

5 m 이상의 구간에선 10 cm 정도의 오차가 발생하기 시작한다.

업데이트 비율은 결국 Target 으로부터 반사되어 돌아오는 신호를 의미하는데 270 Hz 에 해당한다.

## Performance

Specification	Measurement
Range (70% reflective target)	40 m (131 ft)
Resolution	+/- 1 cm (0.4 in.)
Accuracy < 5 m	±2.5 cm (1 in.) typical*
Accuracy ≥ 5 m	±10 cm (3.9 in.) typical Mean ±1% of distance maximum Ripple ±1% of distance maximum
Update rate (70% Reflective Target)	270 Hz typical 650 Hz fast mode** >1000 Hz short range only
Repetition rate	~50 Hz default 500 Hz max

\*Nonlinearity present below 1 m (39.4 in.)

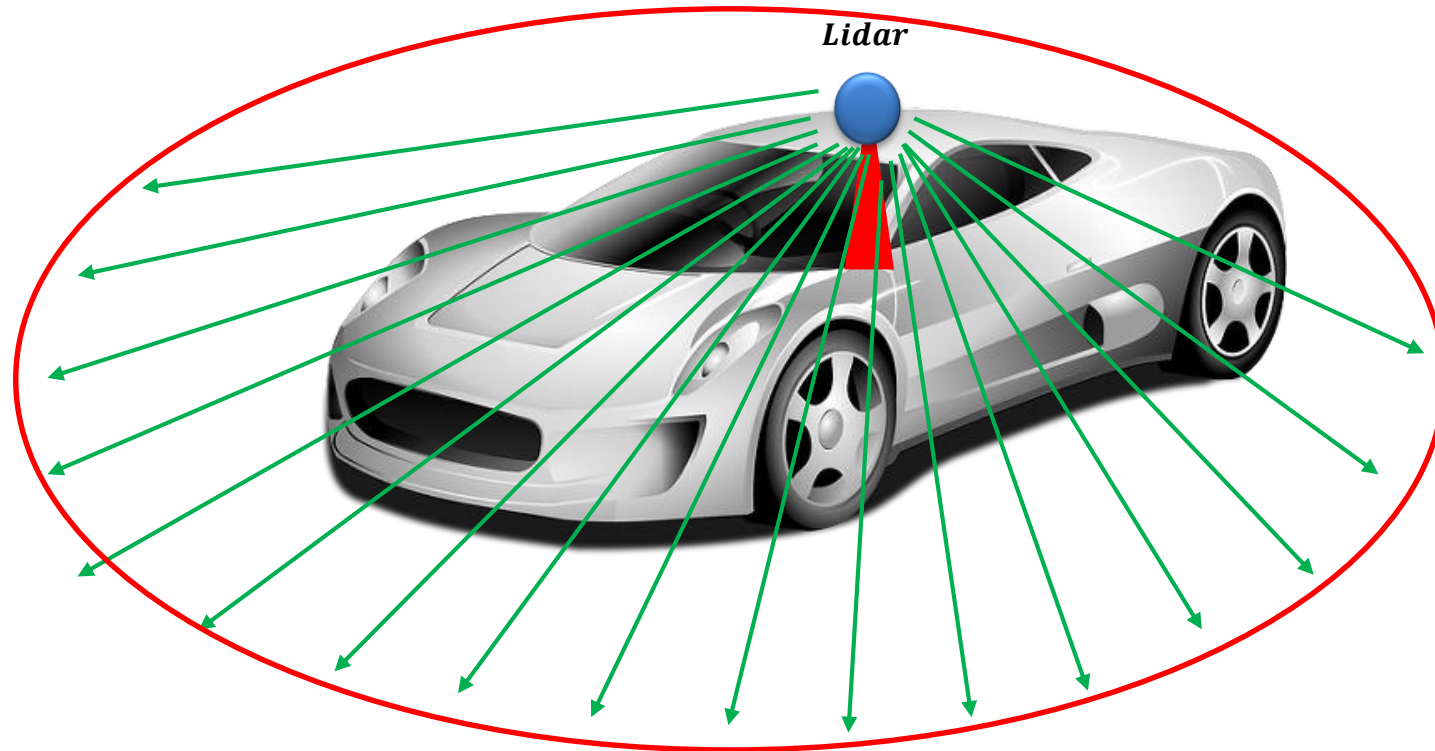
\*\*Reduced sensitivity

차량과 Lidar 를 배치하여 범위에 대해 모델링 해보도록 한다.

Lidar 를 일정한 각도로 기울여서 그 주변을 계속 스캔하도록 만든다.

그렇게 되면 일정한 원 구간이 발생하게 되고 그 영역이 Lidar 의 스캔 영역이 된다.

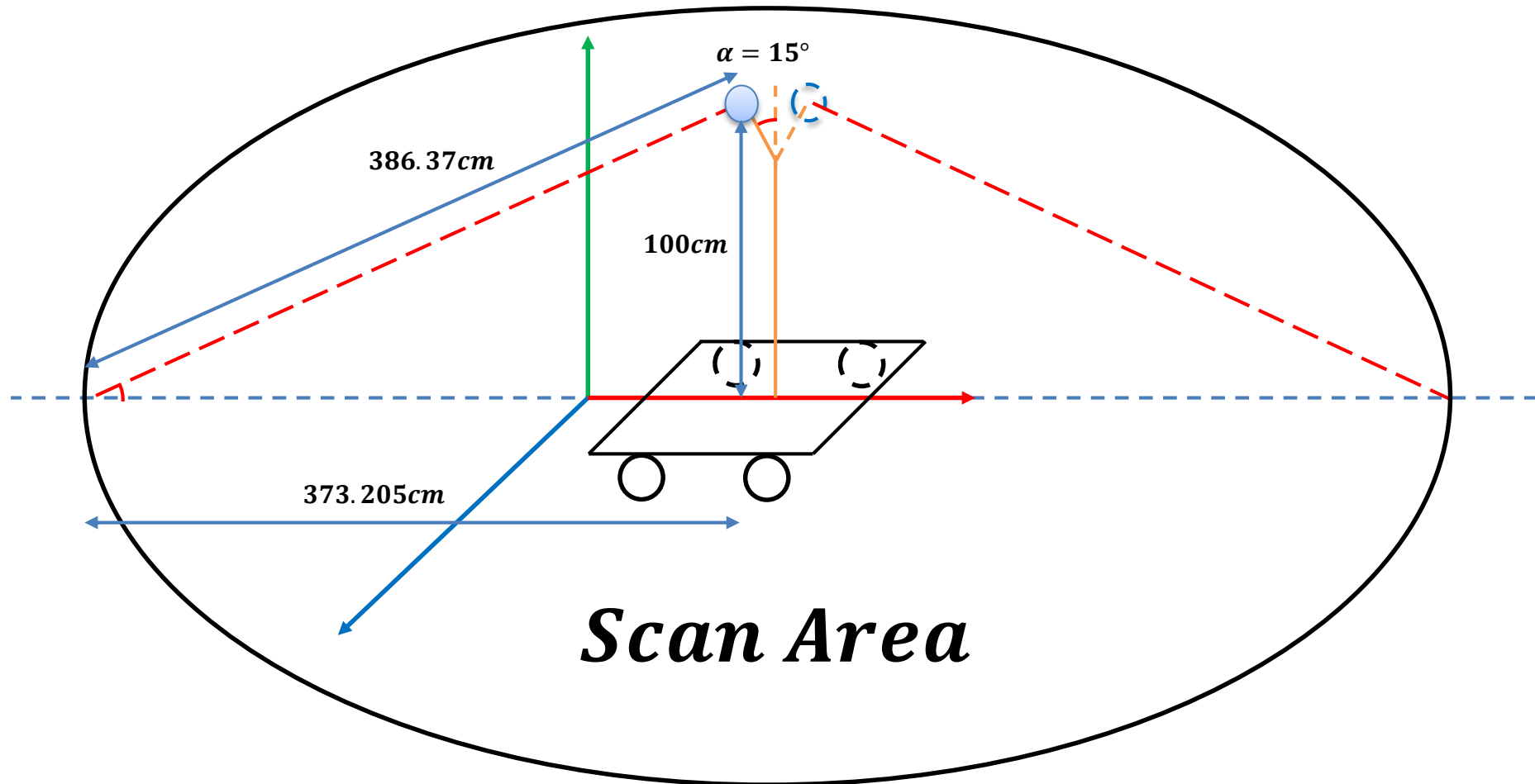
이 영역 내에 물체가 있는지 없는지를 판별하여 영상과 함께 차량을 제어하는데 활용 된다.



**기하학적 해석을 수행하도록 한다.**

$$386.37cm \times \sin(15) = 100cm$$

$$386.37cm \times \cos(15) = 373.205cm$$



# Lidar Scan Speed Calculation

Lidar 가 얼마나 회전해야 하는지 파악해보도록 한다.

우선 필요한 스펙들을 짚욱 열거해본다.

$$h = 1m$$

$$r = 3.732m$$

$$v_{max} = 3m/s$$

$$Lidar\ Update\ Frequency = 270Hz$$

$$Lidar\ Update\ Period = \frac{1}{270} = 0.0037037037 \dots$$

$$x\ rad/s = \frac{x}{2\pi}\ rev/s = \frac{60}{2\pi} x\ RPM$$

이 상태에서 최소 조건을 걸어보자!

Lidar 의 스캔 각을 15 도라고 가정하고 해석을 진행해보도록 한다.

$$15^\circ : 1 = 360^\circ : x \Rightarrow 360 = 15x$$

$$\therefore x = 24$$

0.0037037037... 초 마다 15 도를 회전하니 0.0037037037 ... 에 24 를 곱하면 1 바퀴 회전하는데 걸리는 시간이다.

$$0.0888888 \dots == \frac{24}{270}$$

주파수 횡수만큼 반복되면 1 초가 되므로 위의 역수는 결국 초당 회전수가 된다.

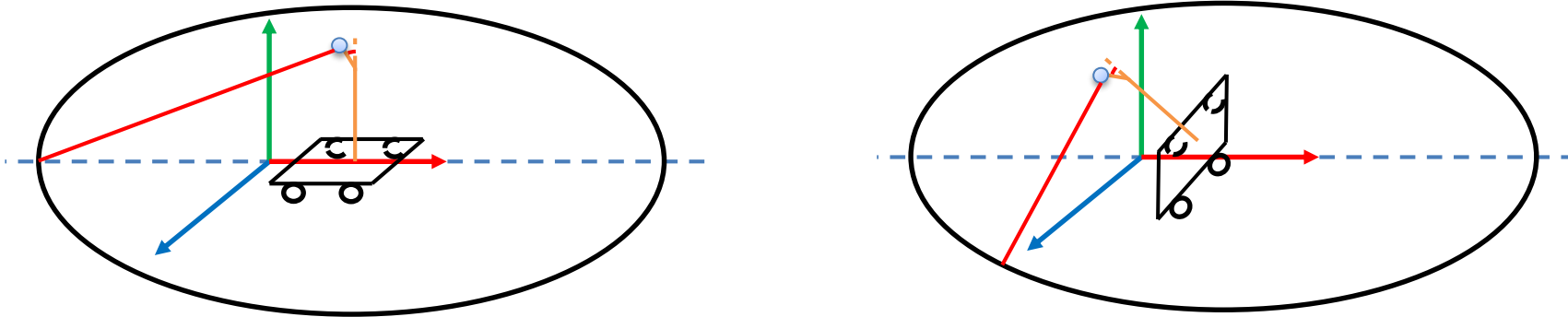
$$\frac{270}{24} = 11.25\ rev/s = 70.68\ rad/s = 675\ RPM$$

# Rotation of the Coordinate System

## 좌표계가 회전한다고 ?!

우리가 Lidar 를 제어함에 있어서 좌표축 자체의 회전에 대해 신경을 많이 써야 한다.  
예로 좌측의 그림을 보면 현 시점에 Lidar 가 스캔하고 있는 영역은  
1 사분면을 지나 2 사분면과 3 사분면의 경계선인 180 도 영역에 해당한다.

반면 차량 자체가 회전을 하게 될 경우를 고려해보도록 하자!

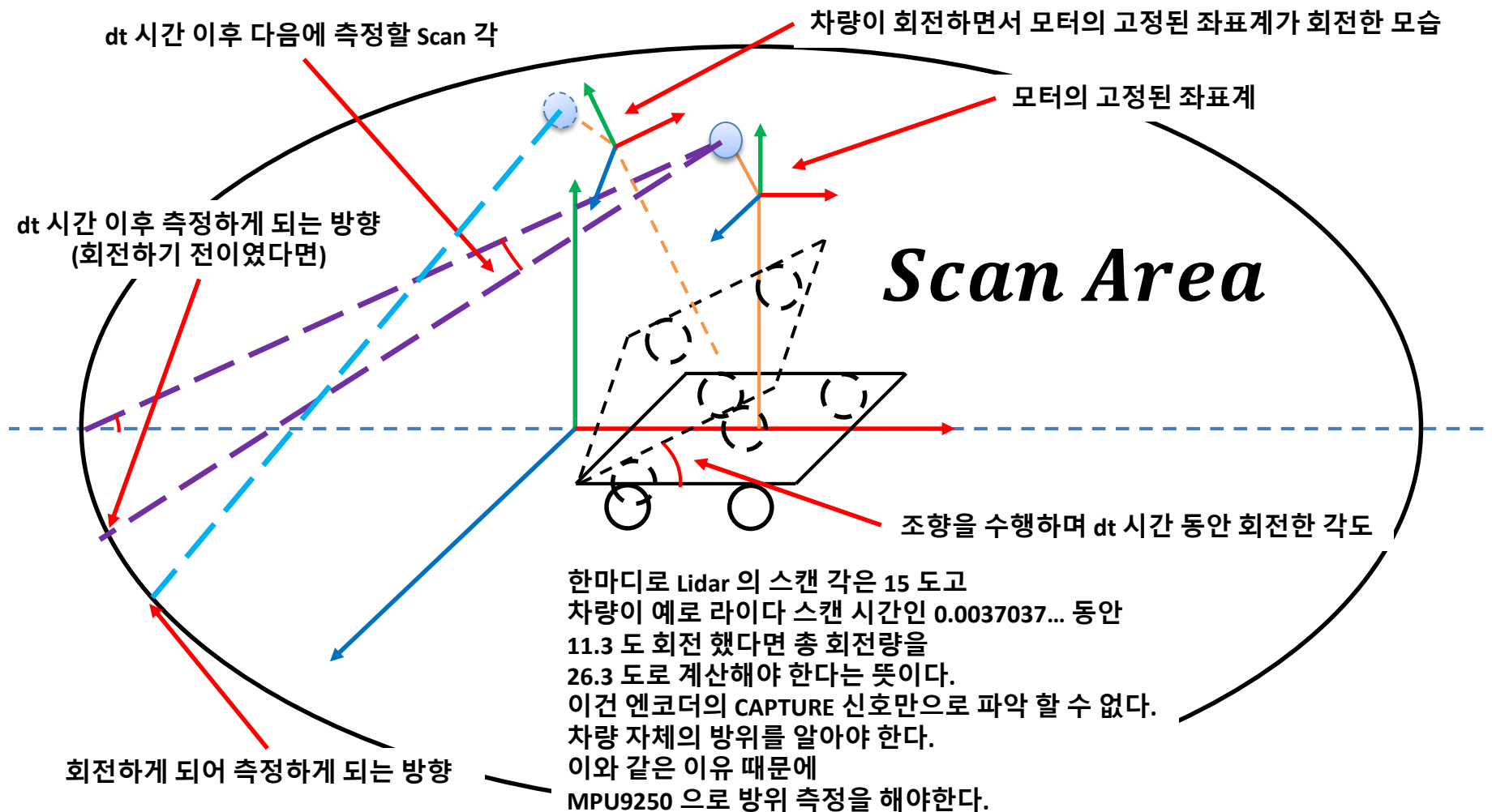


바로 우측 케이스에 해당하는데 차량이 90 도 회전해서 파랑색 축 방향을 향해 달려가고 있다.  
모터는 계속 제자리에서 돌고 있었지만 차량 자체가 90 도 회전을 했기 때문에  
모터의 좌표축 자체가 회전하게 된다는 뜻이다.  
그렇기 때문에 틀어진 만큼 위상을 보정해줘야 한다.  
좌측과 같은 상황이었다는 가정하에 차량이 90 도 틀었다면  
Lidar 는 파랑색 축 방향 즉 270 도 를 스캔하고 있었을 것이다.  
(만약 여기서도 180 도로 계산했다면 잘못된 정보를 가지고 연산을 하고 있으니 분명히 어딘가에 가서 박게 될 것이다)

# Practical Application

실 세계에서 적용을 한다면 어떨까?

모터가 675 RPM 으로 돌고 있으니 차량이 회전하는 90 도의 속도보단 당연히 모터가 회전하는 속도가 훨씬 빠르다.  
물론 그렇지 않다면 직선 주행도 똑바로 못하고 장애물에 쳐박을 것이다.





현재 진행중인 작업은 C++ 의 Nested Template 관련 문서

OpenCV 와 딥러닝을 물리기 위해서라면 반드시 OpenCL 이 필요함

딥러닝을 위한 OpenCL 문서, 그리고 딥러닝을 CPU + DSP 기반에서 C++ 로 작성하기 위한 최소한의 문서,  
추가적으로 수학과 물리 설계 문서를 작성하고 있음.

또한 영상 처리나 딥러닝에 통계학이 들어가며 랜덤 프로세스에 대한 내용이 요구됨.

Lidar 신호 처리를 위한 문서 작업이 향후의 과제에 해당한다.