ToC
(Team of Chasing Car)

[미래형 무인 주행 자동차]

프로젝트 매니저 : 이상훈

김웅일, 김주황, 김준혁, 이영상, 송명규, 최고봉

CONTENTS

프로젝트 소개

팀명	**TOC (Team Of Chasing car)**
프로젝트명	미래형 무인주행 자동차
프로젝트 소개	라인 트래킹, 충돌방지, 모터 정밀제어가 가능한 Remote Control 자동차
사용 Board	TI AM5728, AD8283, HSC-ADC-EVALCZ, ZYBO, STM32F407LGT6, WeMos D1 mini V2 WIFI IoT Development Board
사용기술	영상처리, 초음파 센서, PID 제어, Wi-Fi 무선통신
팀원구성	김준혁, 김웅일, 김주황, 이영상, 송명규, 최고봉

업무분담

김웅일	김주황	김준혁	이영상	송명규	최고봉
Radar	Radar	Remote Controller	Firmware	Firmware	영상처리
Blind Spot Detection(BSD)	Blind Spot Detection(BSD)	Wi-Fi 통신	RTOS	공업수학(제어공학관련)	UART 통신
		스마트폰 Application	모터제어	모터제어	SPI 통신
		TCP/IP Socket 통신			

무선통신 리모콘 (김준혁)

2016.06.03
Wi-Fi 통신을 위한 모듈 선정
ESP-12F 칩을 탑재한 WeMos D1 mini V2

2016.06.09
Linux 상에서 Wi-Fi AP On하여 TCP/IP Server Open

2016.06.16
TCP/IP Application 활용하여 모듈에 Client Connect

2016.06.23
Client로 모듈로 Open한 Server에 메시지 전송 구현

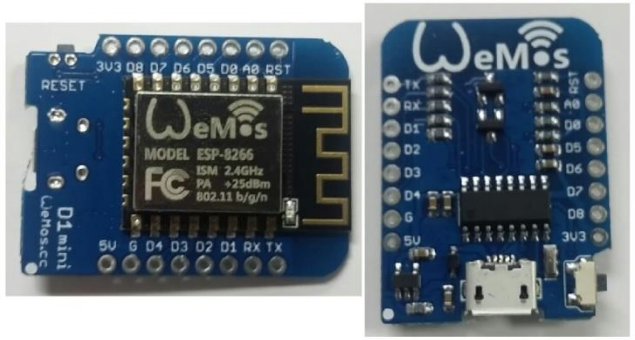
2016.06.30
스마트폰 버튼 Interface로 해당 버튼 터치시 메시지가 전송되도록
Android Application 초기버전 제작

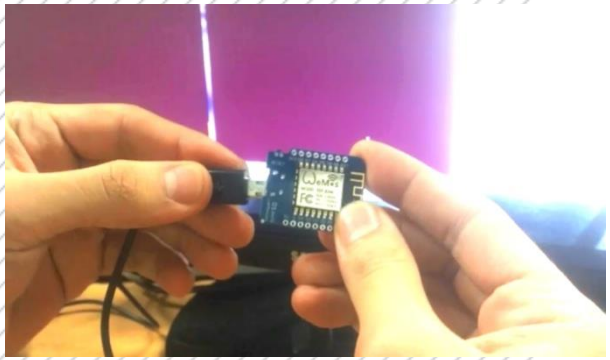
2016.07.04
TCP/IP Socket 통신 구현하여 수신부 측에서 전달 받은 메시지 문자열로 구분하여 처리 및 명령어 실행할 수 있도록 프로그램 구현

2016.07.10
리모컨으로 조종하고자 하는 대상이 자동차이기 때문에 단순한 버튼의 터치 기능 이외에 추가적으로 누르고 있을 시에 지속적으로 동작하도록 하기위한 연속처리가 가능하도록 Application 수정, Update

2016.07.13 현재
리모컨 쪽에서 서버에 제대로 접속이 되었는지 확인 할 수 있도록 구성,
지금까지 PC에 설치되어 있는 리눅스 상에서 환경 구성과 동작 확인을 하였으며,
이제부터 실제로 적용된 AM5728 보드상에서 동작 할 수 있도록 진행

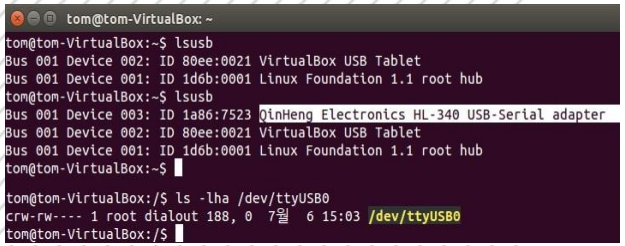
1. Wi-Fi 무선통신을 위한 부품 선정





- WeMos D1 mini V2 WIFI IoT Development Board (ESP8266-12F 탑재)
- TI AM5728 보드와 스마트폰과의 Wi-Fi 무선통신을 위한 mini board
- 무선랜 기능과 Soft AP 기능을 지원하여 Server와 Client로 모두 동작이 가능
- 입력 전압은 3.3V 이며 Micro USB 포트로 전원 공급 및 다른 장치와 USB Serial 통신

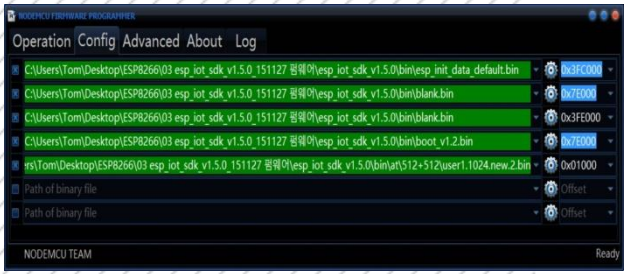
2. 모듈 사용을 위한 환경설정



5. 32Mbit Flash, map: 512KB + 512KB

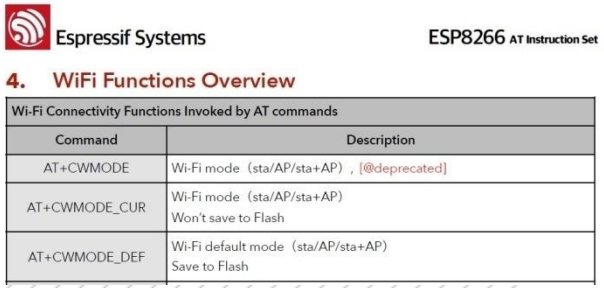
Use Espressif Flash download tool and select Flash size : 32Mbit.

bin	Address	Description
esp_init_data_default.bin	0x3FC000	Optional. Stores default RF parameter values.
blank.bin	0x7E000	Initialize Flash user parameter area, more details in Appendix.
blank.bin	0x3FE000	Initialize Flash system parameter area, more details in Appendix.
boot.bin	0x00000	In \bin\at
user1.1024.new.2.bin	0x01000	In \bin\at\512+512



- PC 및 다른 장치에서 동작이 가능하도록 FTDI의 드라이버를 설치하여 장치 활성화
- 개발 및 동작 환경을 위한 최신버전의 Firmware 다운로드
  (esp_iot_sdk_v1.5.0)
- 다운로드한 Firmware를 설치를 위하여 Flash 프로그램으로 Offset과 파일을 설정하여 Firmware Flash

3. Linux 상에서 AP 설정과 Server Open

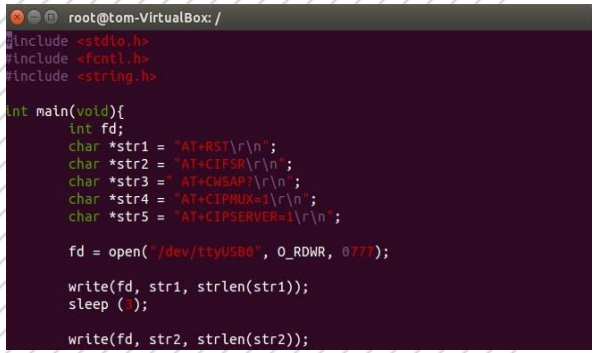


Espressif Systems ESP8266 AT Instruction Set

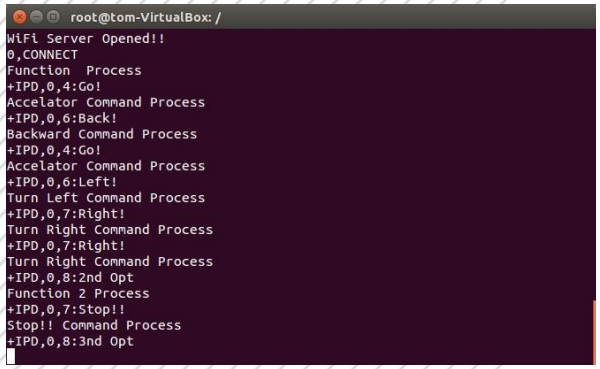
4. WiFi Functions Overview

Wi-Fi Connectivity Functions Invoked by AT commands

Command	Description
AT+CWMODE	Wi-Fi mode（sta/AP/sta+AP）, [@deprecated]
AT+CWMODE_CUR	Wi-Fi mode（sta/AP/sta+AP） Won't save to Flash
AT+CWMODE_DEF	Wi-Fi default mode（sta/AP/sta+AP） Save to Flash



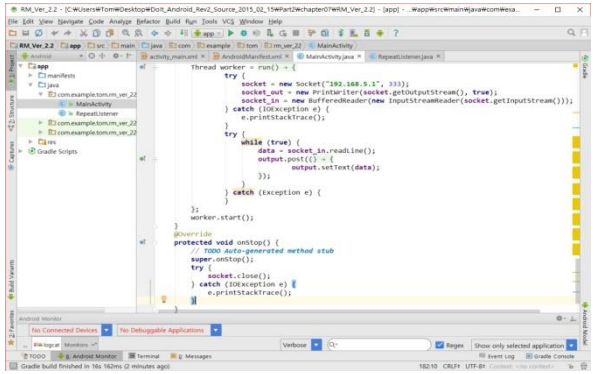
```
root@tom-VirtualBox: /
#include <stdio.h>
#include <fcntl.h>
#include <string.h>

int main(void){
        int fd;
        char *str1 = "AT+RST\r\n";
        char *str2 = "AT+CIFSR\r\n";
        char *str3 = "AT+CWSAP?\r\n";
        char *str4 = "AT+CIPMUX=1\r\n";
        char *str5 = "AT+CIPSERVER=1\r\n";

        fd = open("/dev/ttyUSB0", O_RDWR, 0777);

        write(fd, str1, strlen(str1));
        sleep (3);

        write(fd, str2, strlen(str2));
```



```
root@tom-VirtualBox: /
WiFi Server Opened!!
0,CONNECT
Function  Process
+IPD,0,4:Go!
Accelator Command Process
+IPD,0,6:Back!
Backward Command Process
+IPD,0,4:Go!
Accelator Command Process
+IPD,0,6:Left!
Turn Left Command Process
+IPD,0,7:Right!
Turn Right Command Process
+IPD,0,7:Right!
Turn Right Command Process
+IPD,0,8:2nd Opt
Function 2 Process
+IPD,0,7:Stop!!
Stop!! Command Process
+IPD,0,8:3nd Opt
```

- ESP8266의 기능으로 AT Commands 를 사용하여 AP, Server 설정 (WiFi mode, AP Configuration, Address & Port, TCP Server 등)
- Linux 상에서 장치가 연결되었을 때 초기화 및 기본 세팅, TCP정보 확인, AP정보 확인, Server Open 등의 기능을 수행하도록 기본 프로그램 작성
- Open된 TCP Server에 Client로 부터의 메시지가 전송 되었을때 메시지 출력하도록 Buffer Reader 프로그램 작성

4. Android 리모컨 Application 제작



- 스마트폰으로 자동차의 원격 수동 조종이 가능하도록 Wi-Fi 통신 할 수 있는 Application을 제작
(Android Studio 상에서
Java언어를 사용하여 구현)

- TCP/IP 통신을 위한 Socket을 만들어 장치쪽에 Open한 서버에 IP와 Port 번호로 Client로 접속

- 조종을 위한 버튼을 구성하여 버튼 클릭시 해당 메시지를 Server 쪽으로 전송하도록 프로그램 작성

영상처리(최고봉)

- Open CV 함수의 문제점

Open CV의 Canny 필터나 Sobel 필터를 사용하여 에지를 검출한 후

허프 공간에서 직선성분을 찾아 차선을 검출하는 방법은 차선 이외의 에지들이 많이 존재할 경우 오검출률이 높아지는 단점이 있음

차선의 특징 추출

- 차선의 특징 추출을 위한 가정

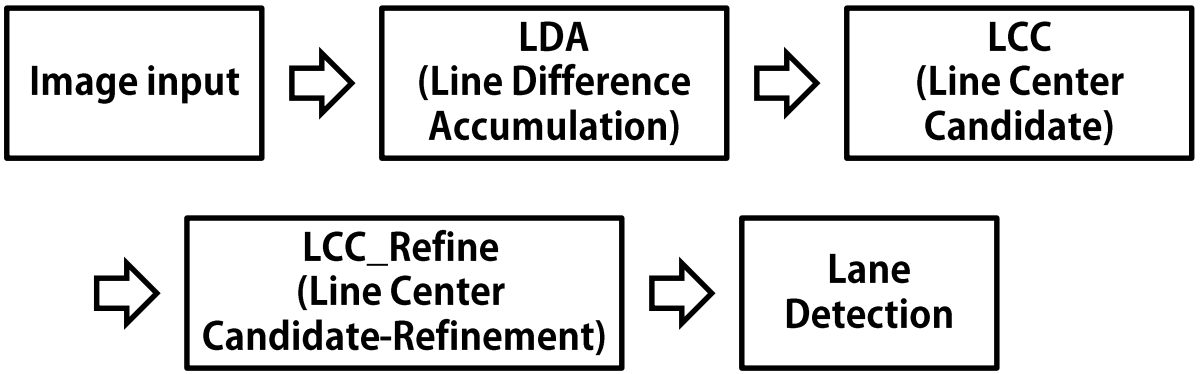
  1) 차선은 항상 도로보다 밝다.

  2) 차선은 일정한 크기의 폭을 지닌다.

  3) 운전 방향과 차선의 방향은 평행하다.

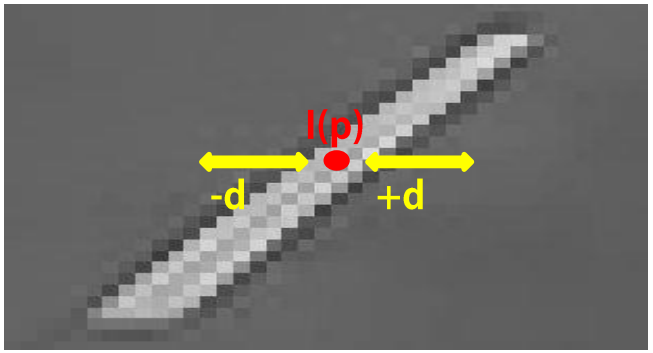
위의 가정은 밝기 변화 무관하다.

- Flow diagram



Line Difference Accumulation (LDA)

- 차선 특징 추출의 예시



(그림 1)

- 입력 영상을 아래의 식(1)을 이용하여 그레이 이미지로 변환

$$G(x,y) = (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \quad (1)$$

- 변환된 영상 G(x, y)를 식(2)를 적용하여(그림1과 같이) 밝기 값 차이를 가지는 영상 I(x, y)를 구함

$$I(x,y) = 2 \times G(x,y) - G(x+d,y) - G(x-d,y) \quad (2)$$

Line Difference Accumulation (LDA)

- I(x, y)의 밝기 차이 값이 음수일 경우 가정 1에 위배되므로 식(3)에서 제거

$$D(x,y) = \begin{cases} 0 & , \text{ if } (I(x,y) < I(x+d,y) \text{ or } (I(x,y) < I(x-d,y) \\ Val(x,y) & , \text{ otherwise} \end{cases} \quad (3)$$

- 실세계에서는 차선이 넓이가 일정하지만 영상에서는 차선의 넓이가 Y축에 비례하여 점점 커짐
- 따라서 d는 아래식과 같이 가중치 요소를 적용하여 점점 커져야 함

$$d_{min} = \mu \times (y - V_y)$$
$$d_{max} = \mu \times (y - V_y) + 3 \quad (4)$$

- 최종 차선 특징 추출 영상은 아래 식과 같음

$$H(x,y) = \sum_{d=d_{min}}^{d_{max}} D(x+d,y) + \sum_{d=d_{min}}^{d_{max}} D(x-d,y) \quad (5)$$

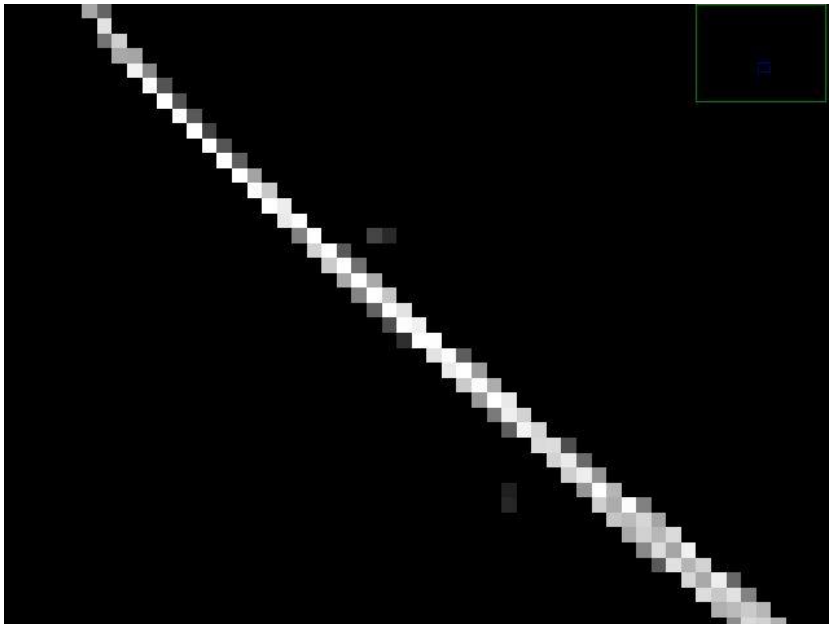
차선의 특징 추출결과

(그림 2. 원본 Gray 영상)

(그림 3. 차선의 후보 영역(LDA))

차선 후보 영역의 특징

(그림 4. 원본 영상)



(그림 5. 확대 이미지)

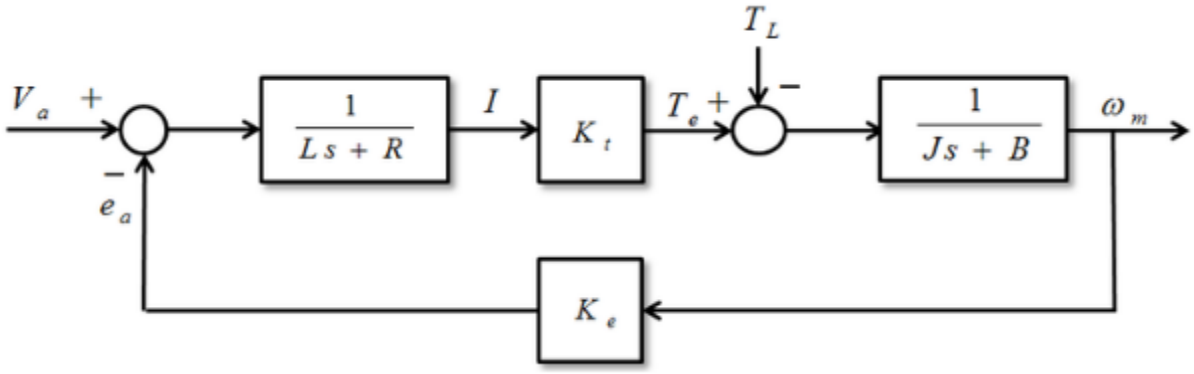
- 검출된 차선 후보 영역은 그림 5에서 보이는 것처럼 중앙이 가장 큰 값을 가짐
- 상대적인 밝기 값을 비교하기 때문에, 그림자 영역이나 석양 영역에서 낮아진 그레이 영상의 밝기 값은 차선 후보 검출에 영향을 주지 않음

모터제어
(이영상)

2016/5/30 : STM32의 예제 분석 시작
2016/6/2 : PWM을 사용하여 LED 제어 성공
2016/6/4 : ADC관련 예제를 수행해보고 외부의 가변저항의 값을 읽어오는 것을 수행해보았지만 실패.
2016/6/5 : ADC의 채널설정이 잘못된 것을 확인하고 해결함
2016/6/6 : 모터의 rpm을 측정하기 위해 Hall센서를 읽어오는 것을 시작
2016/6/9 : 외부 인터럽트를 사용하여 Rising Edge를 검출 시 count하는 것을 성공
2016/6/10 : FreeRTOS의 설정을 시작함
2016/6/13 : Task를 나누고 같은 우선순위를 주었음에도 하나의 Task만 돌아가는 문제가 발생
2016/6/17 : 모터제어를 위한 모델링을 시작
2016/6/22 : 모터의 제어를 위한 PI 제어기의 설계를 시작
2016/6/27 : 모터의 Parameter가 없어 직접 측정과 계산을 수행함
2016/7/05 : Parameter의 계산과 PI제어기의 이득을 결정하기 위하여 matlab으로 simulation 중임.

2016/7/07 : 제어기의 설계를 완료함.
속도제어기에 PI & IP 를 함께 쓰는 방식으로 구현했지만 PI제어기로만 구현한 것과 결과의 차이가 보이지 않음
전류가 1초 delay가 생기는데 이유를 알 수 없음
2016/7/12 : 제어기의 코딩을 끝냄.
2016/7/13 : 전류제어기에 Feedback 해줄 전류를 모터드라이버에서 받아오는 중.

DC모터 모델링

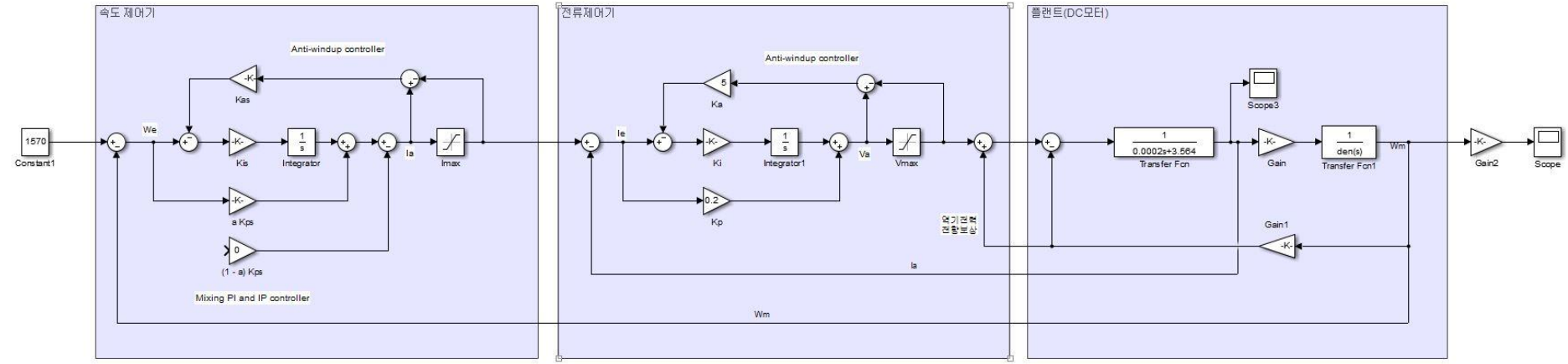


DC모터의 구동시스템 블록도

$$\frac{\omega_m(s)}{v(s)} = \frac{k_T}{(L_a s + R_a)(Js + B) + k_T k_e}$$

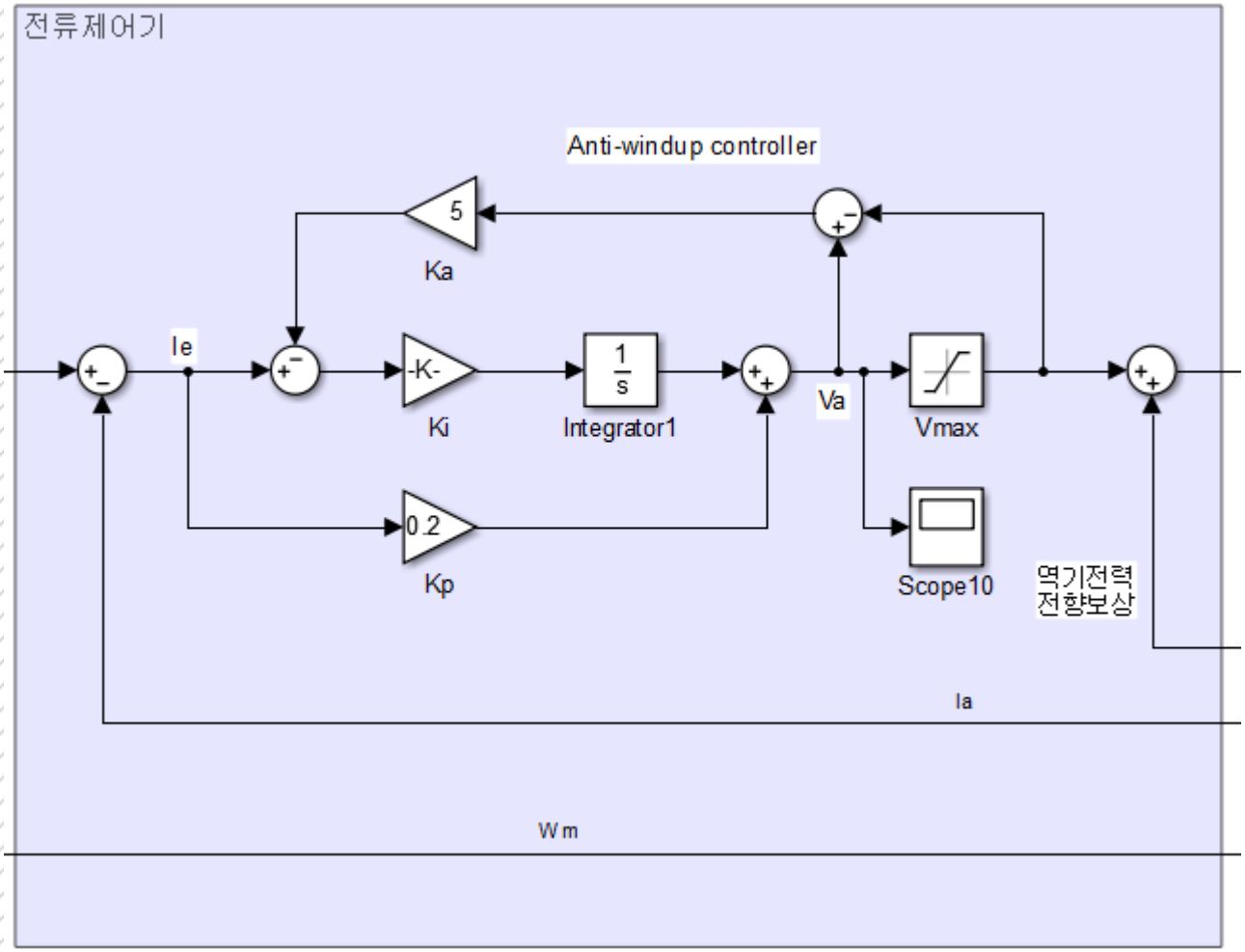
전압과 각속도의 관계식

PID 제어기 설계



PI 전류제어기와 PI & IP 속도제어기

PID 제어기 설계



전류 제어기 블록도

PI 전류제어기 설계

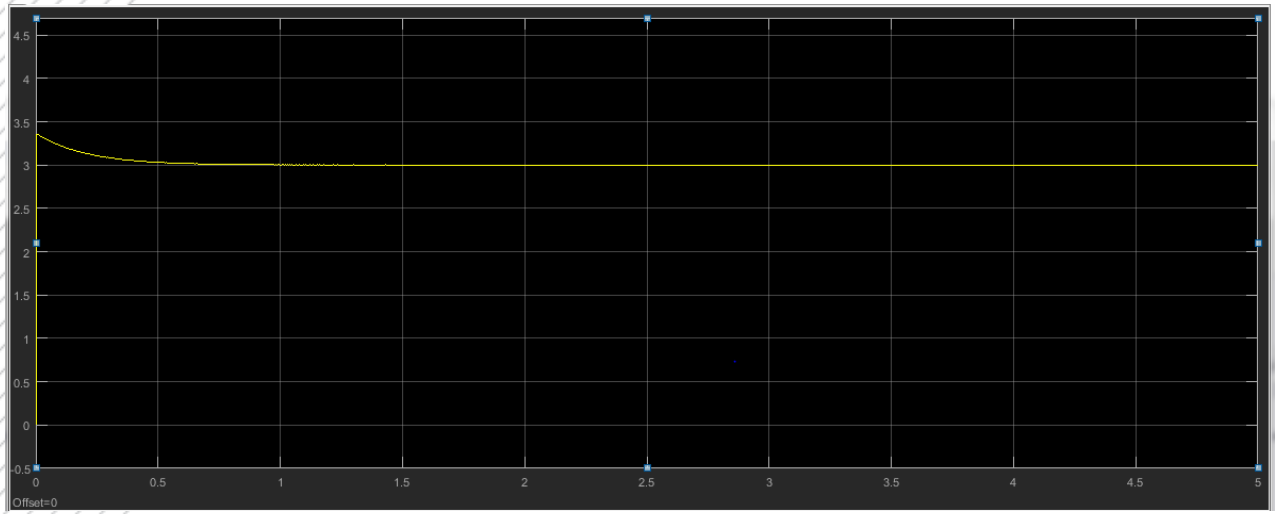
$$K_p = L_a \omega_{cc}$$

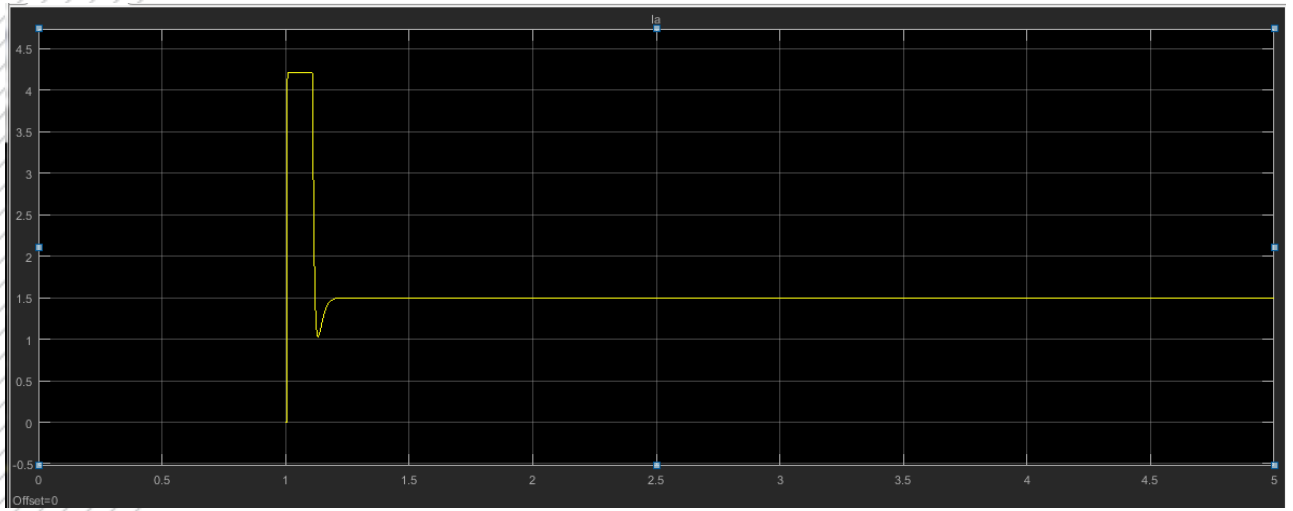
$$K_i = R_a \omega_{cc}$$

$$K_a = 1 / K_p \text{ ,} \quad \left(\frac{1}{3K_p} < K_a < \frac{3}{K_p} \text{ 범위에서 선정} \right)$$

제어기 Gain

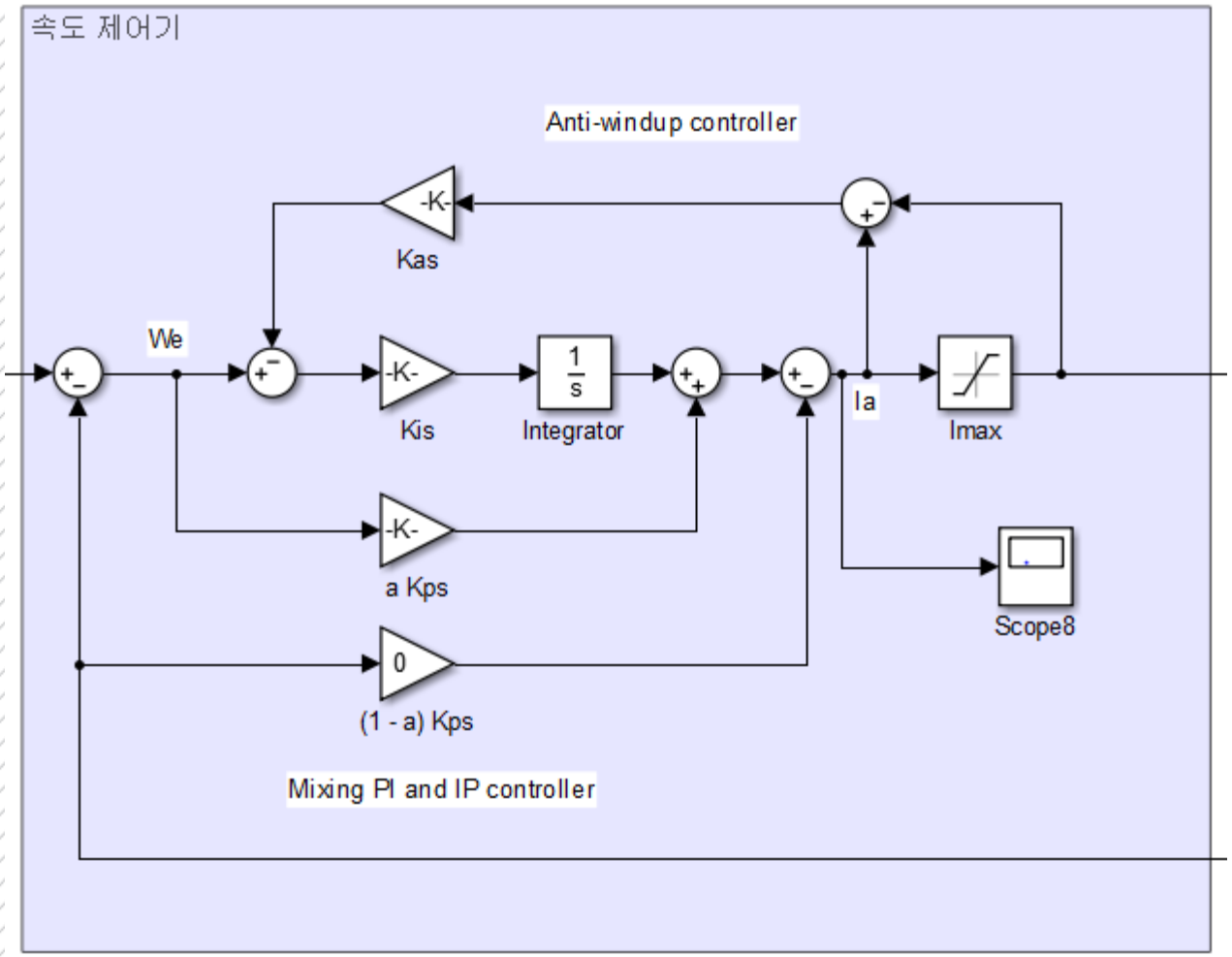
PI 전류제어기 설계





MATLAB 시뮬레이션 결과

PI & IP 속도제어기 설계



속도 제어기 블록도

PI & IP 속도제어기 설계

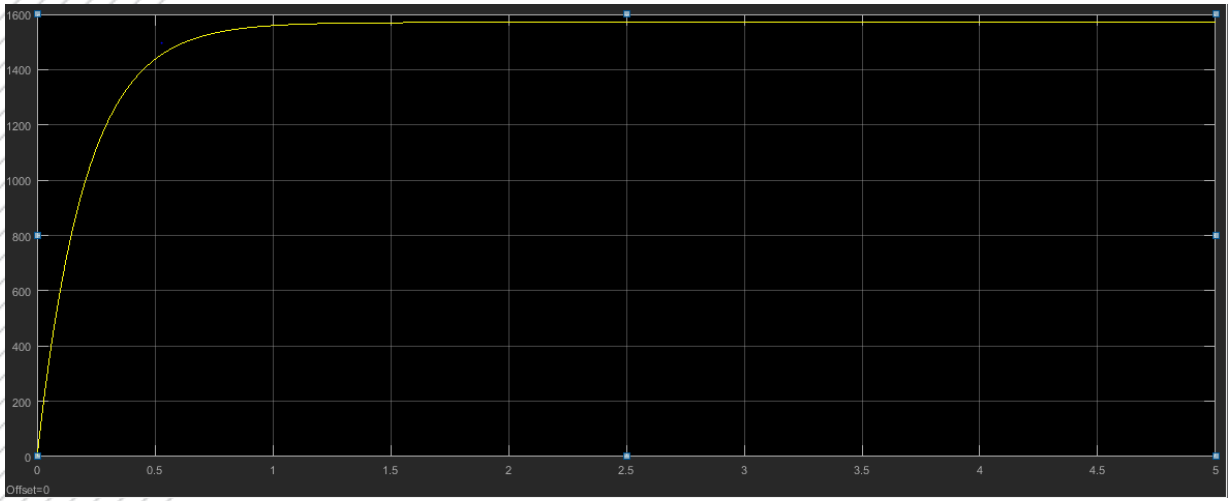
$$K_{ps} = \frac{J\omega_{cs}}{K_T}$$

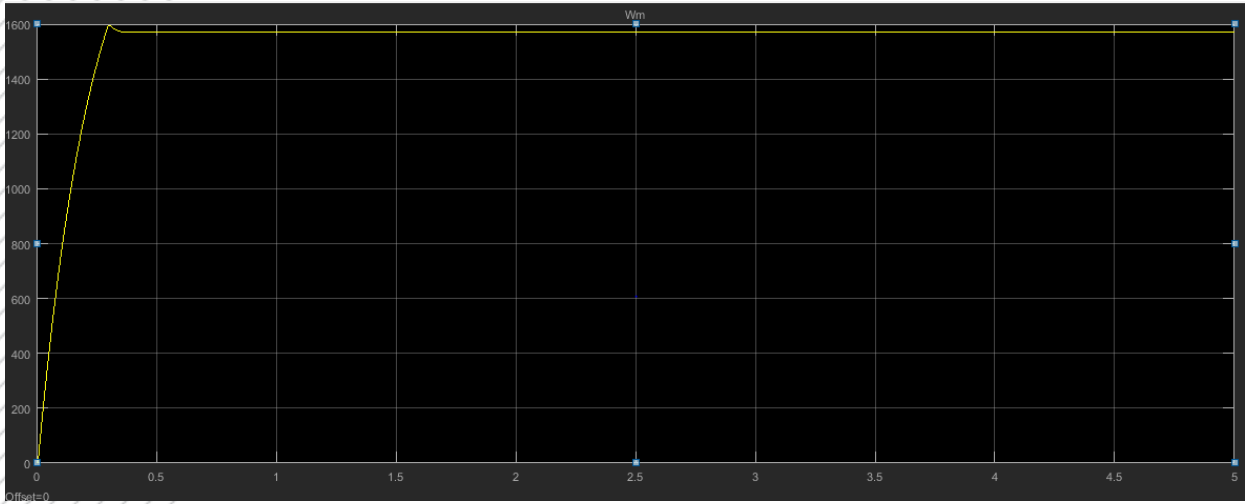
$$K_{is} = \frac{J\omega_{cs}^2}{5K_T}$$

$$K_{as} = \frac{1}{K_{ps}}$$

제어기 Gain

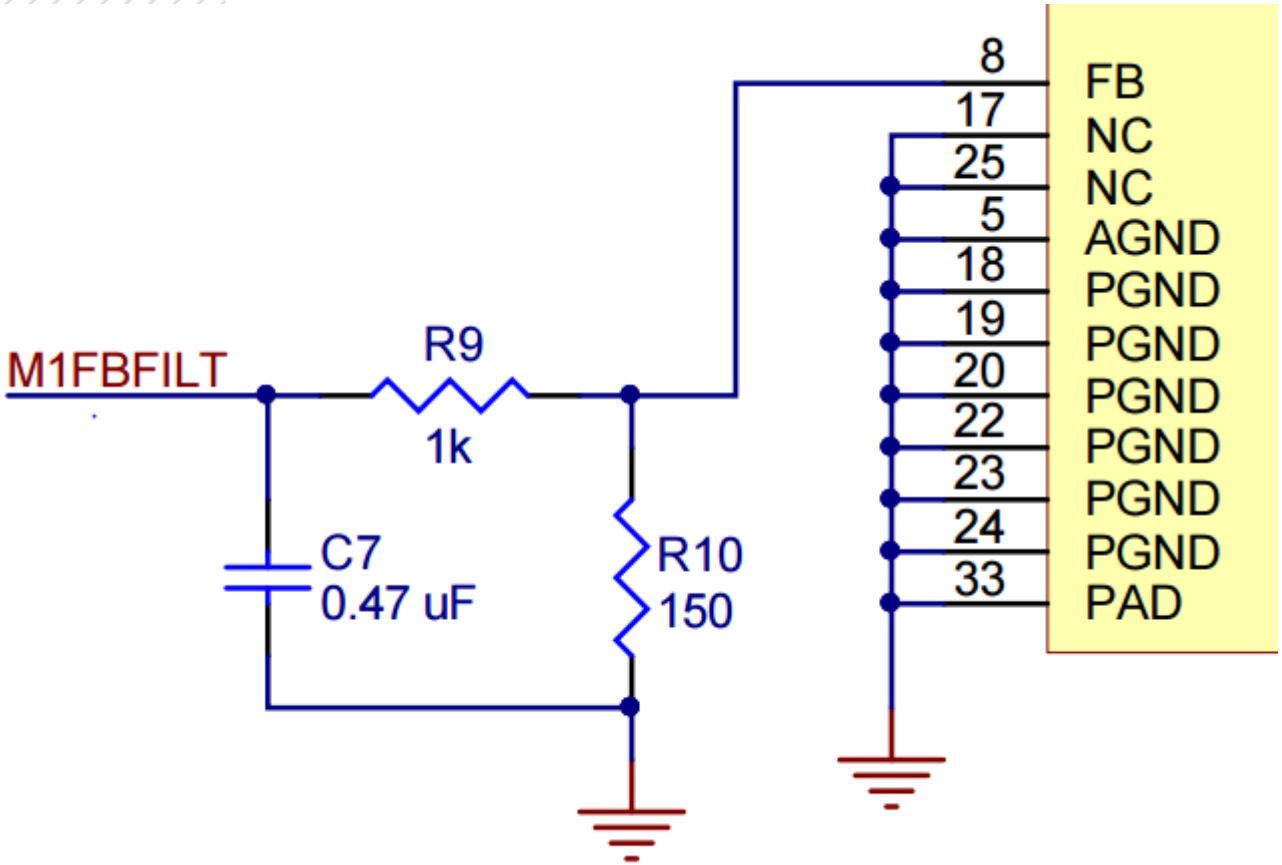
PI 속도제어기 설계





MATLAB 시뮬레이션 결과

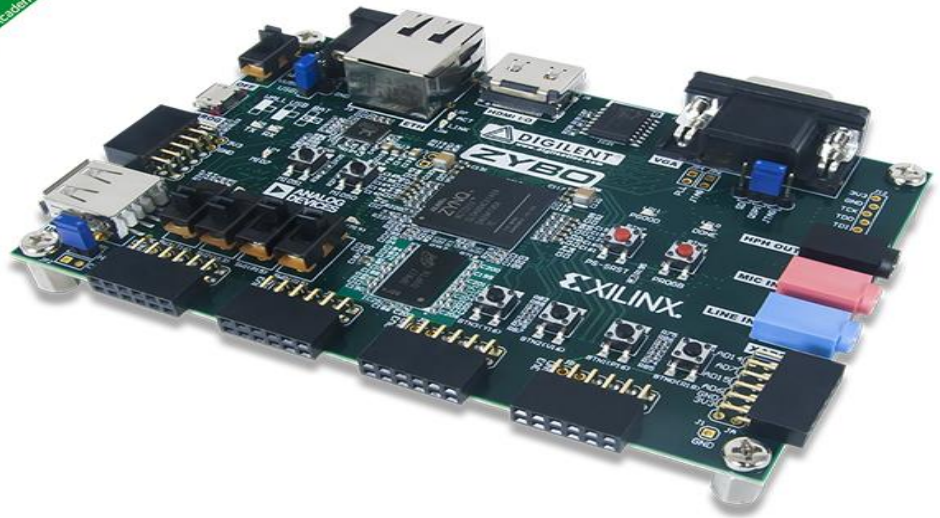
전류 Feedback



모터 드라이버 전류 Feedback 부분 회로도

초음파 센서
(김웅일, 김주황)

ZYBO
HC-SR04



진행상황

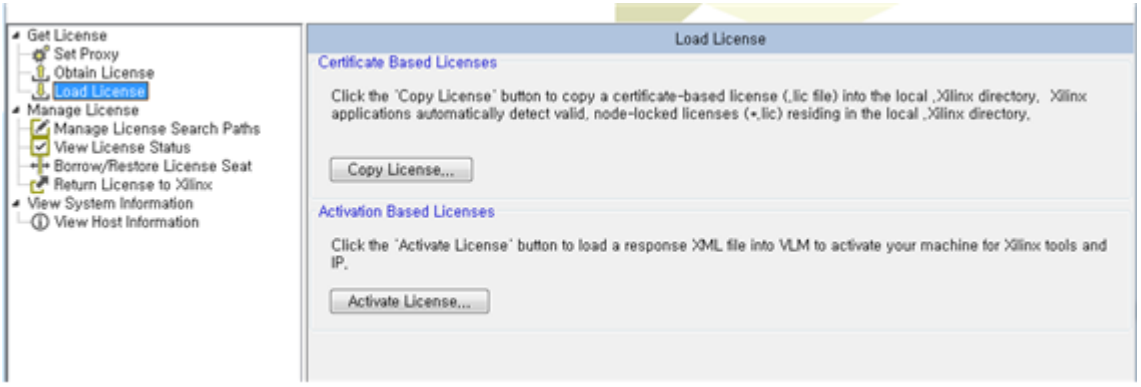
ZYBO board의 구성 및 사용 방법 조사

LED 제어

HC-SR04(초음파 센서)를 활용하여 물체감지 시도

진행상황

ZYBO board의 구성 및 사용 방법 조사

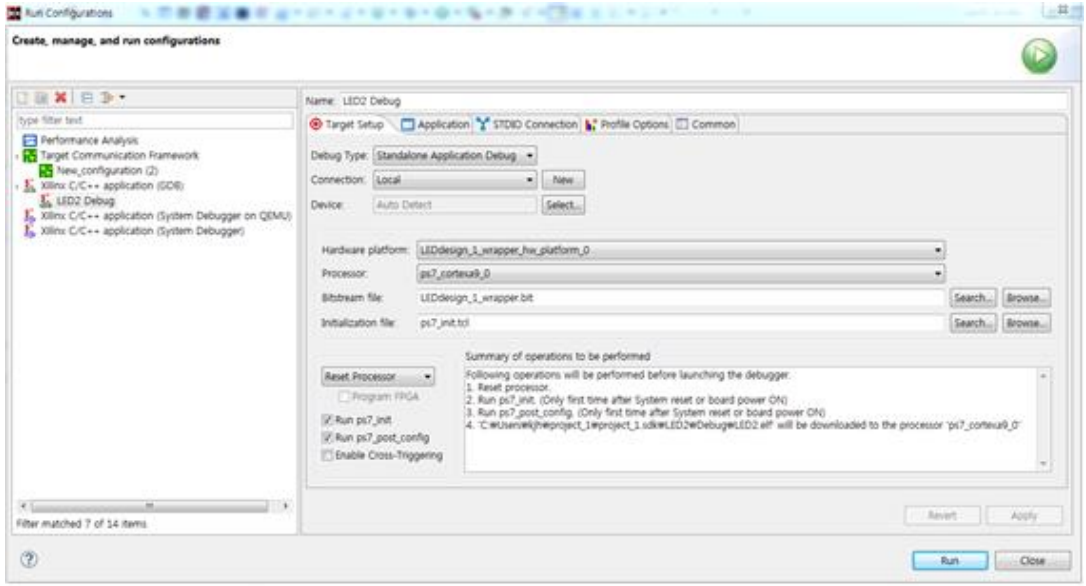


초기 Vivado 설치시 문제발생
해결법 : Load License가서 Copy License를 해주면 됨

LED 제어

HC-SR04(초음파 센서)를 활용하여 물체감지 시도

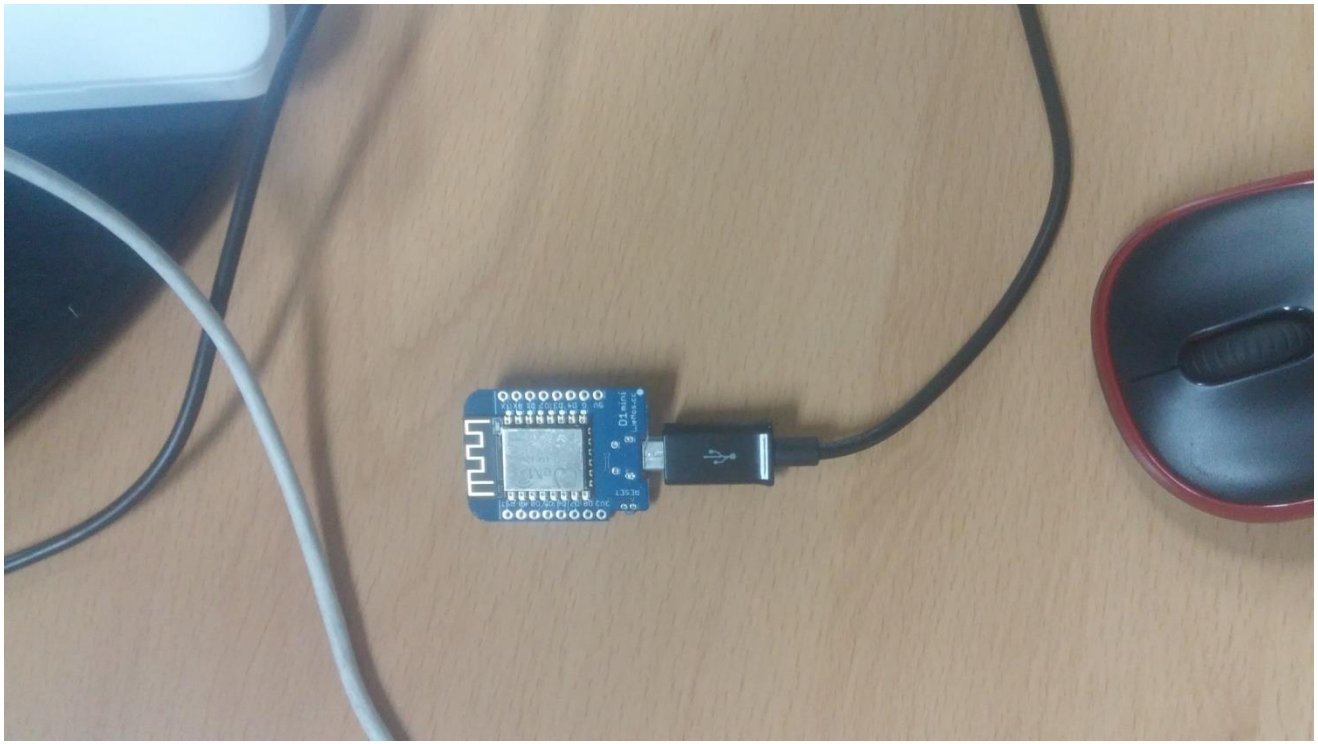
진행상황



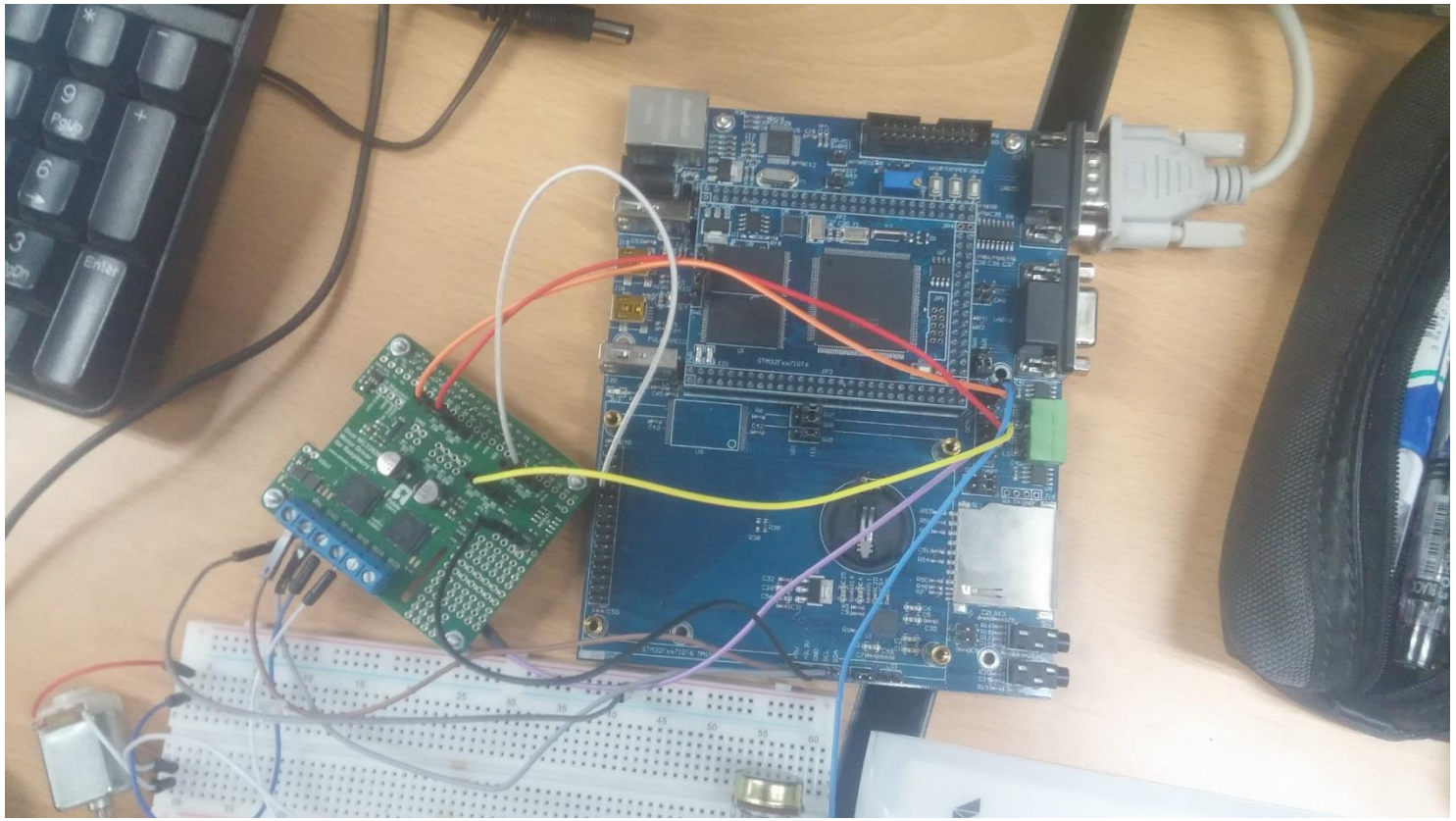
디버깅시 오류발생

영상처리 보드

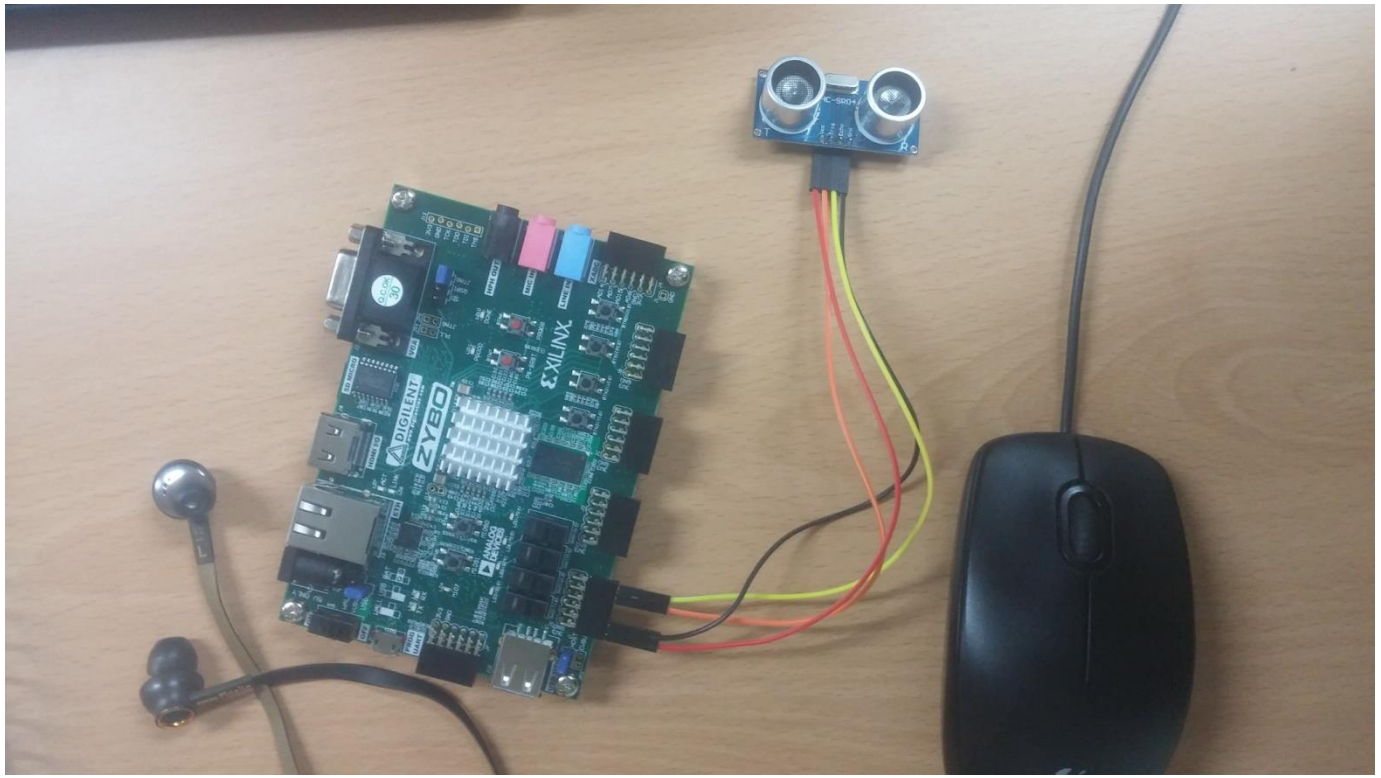
WI-FI 모듈

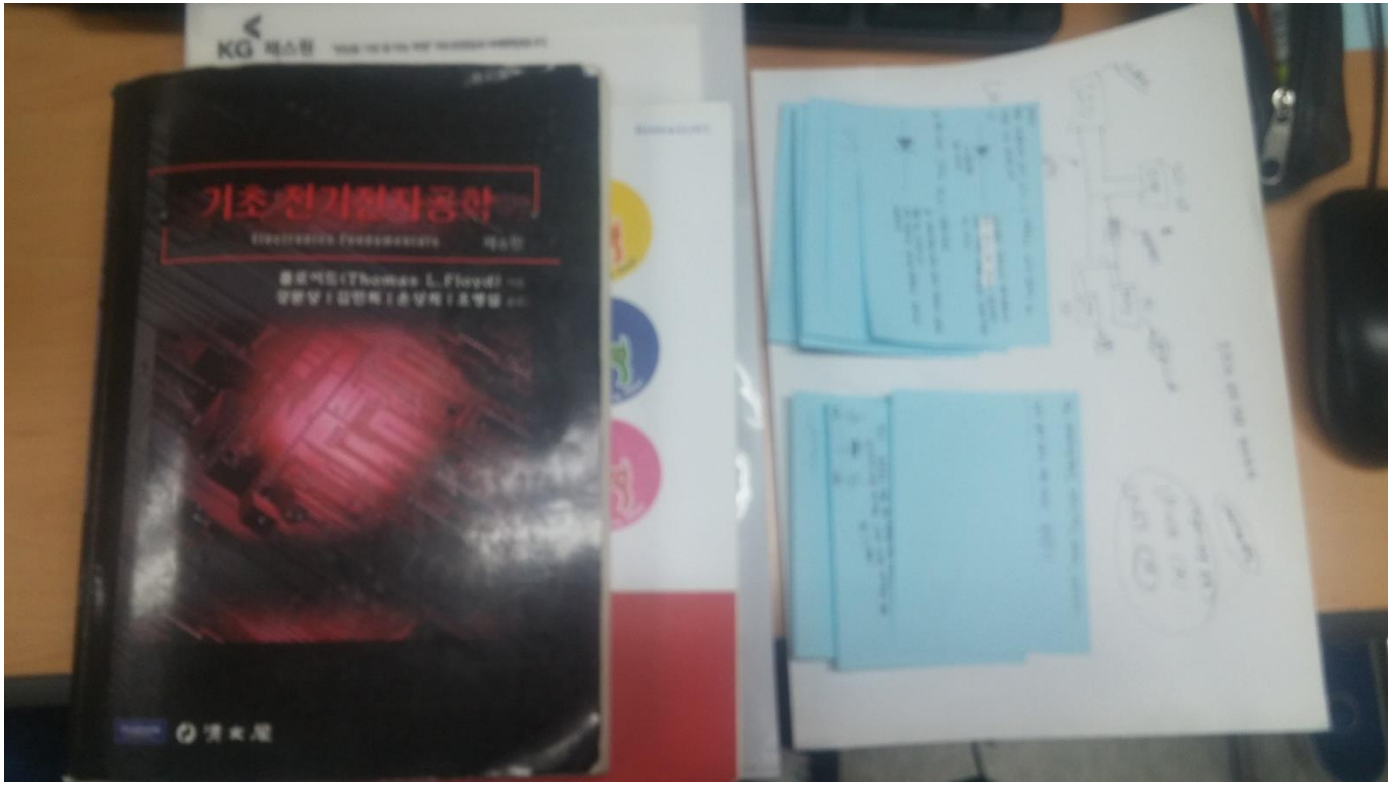


STM 보드 (이동,조향)



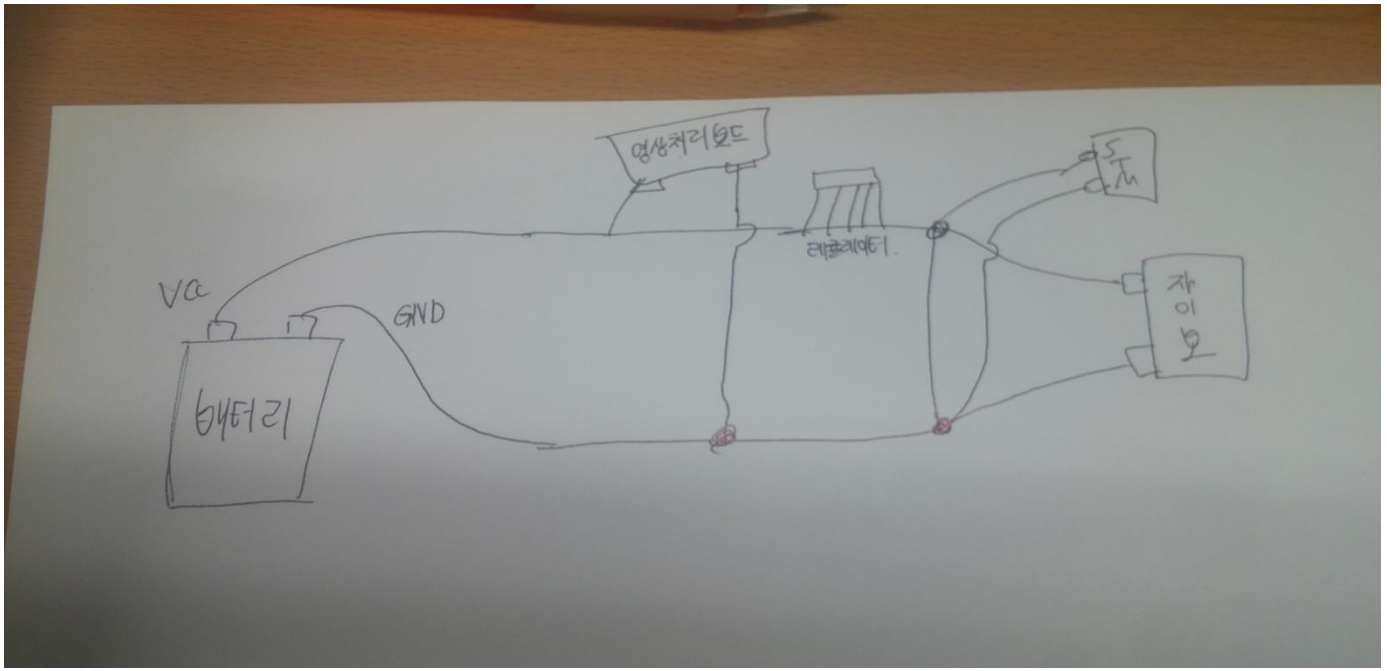
자이보 보드(초음파)





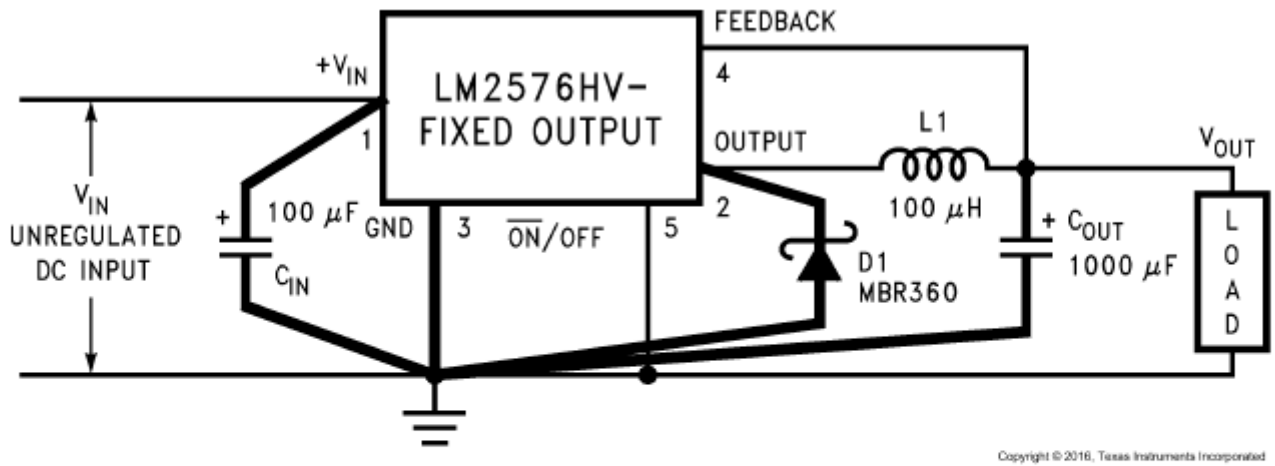
회로이론을 위해 공부한것.

전원설계 구상도



주의 !! :그림에서와 다르게 실제 전원설계를 구성할때에는 각보드에 따로 연결을 할것입니다. 위의 그림은 제 개인적인 주관에 의해 이상적으로 생각하여 구상한것으로 실제와는 많이 다를 수 있습니다.

LM2576 레귤레이터 구성도



C_{IN} — 100-µF, 75-V, Aluminum Electrolytic
C_{OUT} — 1000-µF, 25-V, Aluminum Electrolytic
D_1 — Schottky, MBR360
L_1 — 100 µH, Pulse Eng. PE-92108
R_1 — 2 k, 0.1%
R_2 — 6.12 k, 0.1%

Figure 26. Fixed Output Voltage Versions

7. 수행일정

	1주	2주	3주	4주	5주	6주	7주	8주	9주	10주
최고봉	영상처리									
						DSP 제어 & 보드간 통신				
김주황 김웅일	레이더 이론									
						FMCW방식을 이용한 BSD				
이영상	Firmware									
		RTOS								
			Motor Control							
						통신 & 회로 & 필터				
송명규	데이트 시트 분석									
			공업수학							
						회로이론&회로설계				
김준혁	모듈 선정 및 Wi-Fi AP 설정									
			Android Application 제작 및 스마트폰 연동							
						TCP/IP 소켓통신 구현 및 수신부 메시지, 명령어 처리				
									전체 Flow Test 및 시스템 연동	