

# 자동 정밀타격 포탑

팀장 : 김동혁

팀원 : 이동훈

김왕배

정범수

# 역할 분담(1차 시기)

- 동혁 (팀장) : CAN/Networks server, 기구(몸체) 설계, 통신
- 왕배 : (레일건) 회로제작 및 실험, MCU, FPGA
- 범수 : FPGA( Lidar, 절대엔코더), MPU (기구 수평)
- 동훈 : (레이저, 물로켓) 제작 및 실험, MCU, 제어기(속도),  
(+a 기구 설계)

# 프로젝트 목적

## 무인 감시 포탑

출산률 저하에 따른 군 병력 감축으로 인해 무인 시스템의 도입 필요성이 부각되고 있다.

로봇은 항상 일정한 컨디션으로 감시하기 때문에 사람보다 신뢰성이 높다.

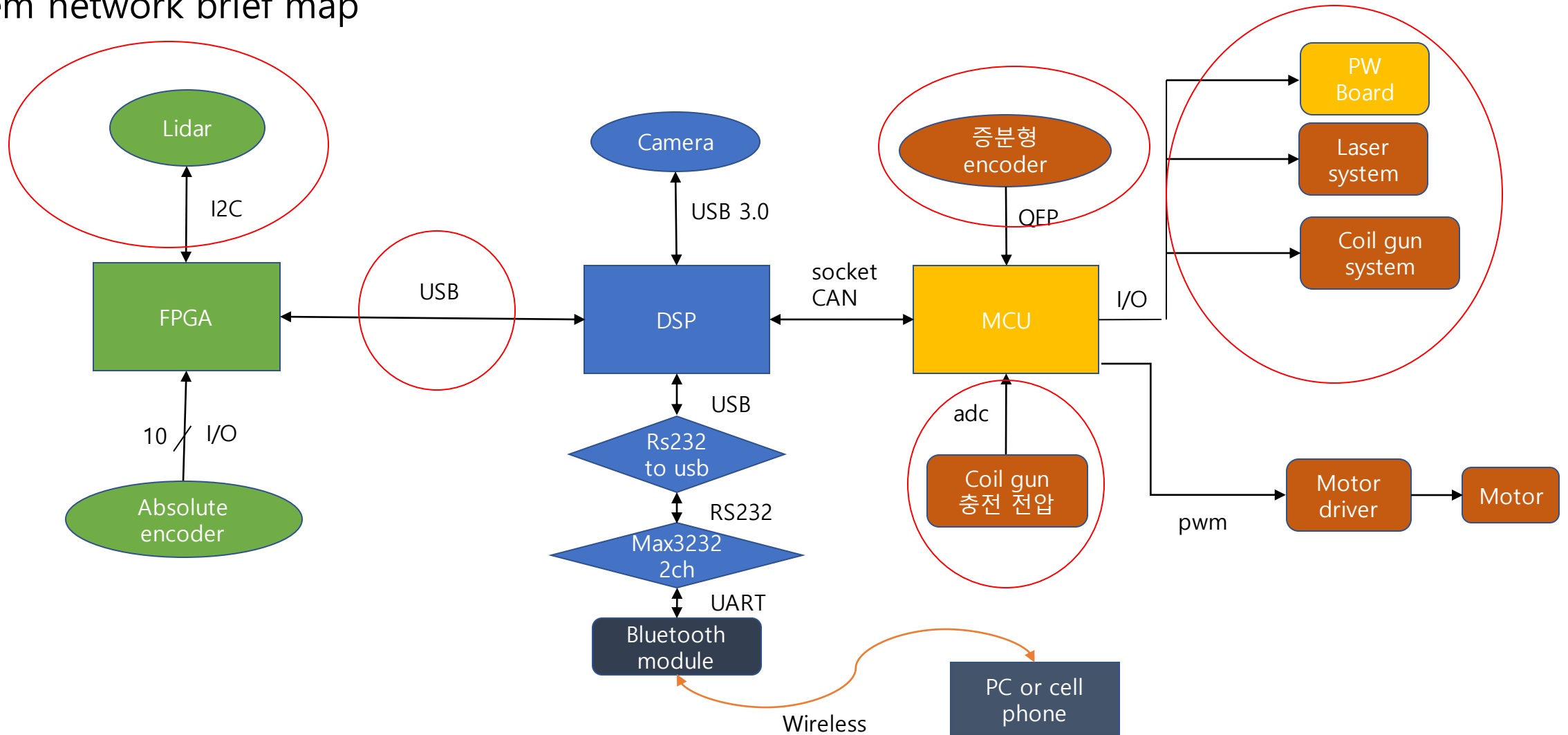
무인 감시 뿐만 아니라 무기를 도입하여 미래형 방위 시스템을 구축하려고 한다.

# 포탑의 사양

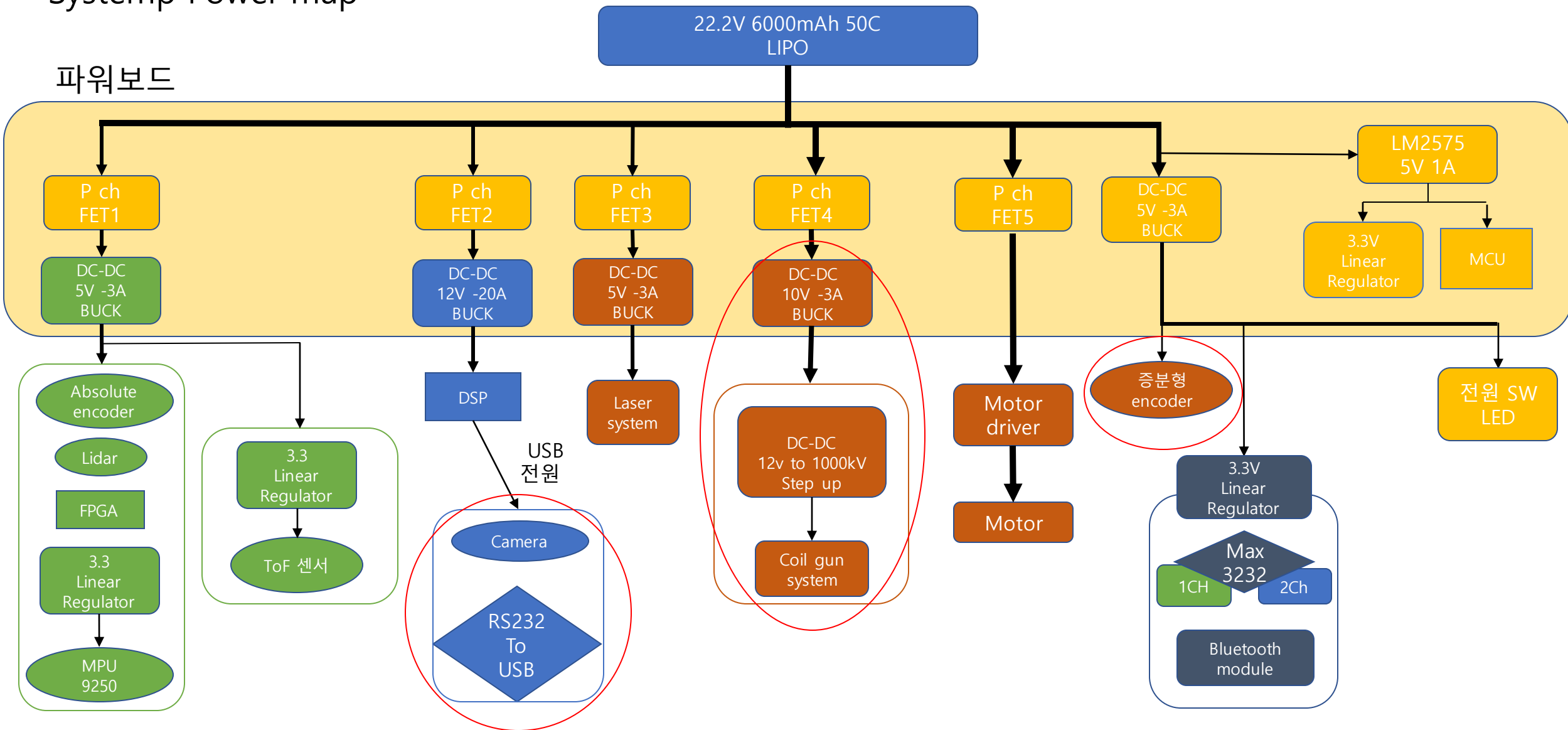
- 포탑에 들어가는 무기 체계는 코일건과 레이저이다.
- 코일건의 경우 포탑의 높이와 같은 높이의 목표물을 추적 타격한다.  
1차 목표 : 정지 목표물. 2차 목표 : 이동 표적
- 포탑의 높이에서 벗어난 목표물은 레이저로 타격한다.
- 사거리는 15m이다.
- 코일건의 경우 거리에 따라 경우 각도를 바꾸어 사격
- 목표물은 검은색 풍선이다.
- 포탑의 기본 회전 속도는 30rpm 이다.
- 포탑의 평형 조절 및 높이 측정은 수동으로 한다.

# 시스템 아키텍처

System network brief map



# System Power map



기구부

# 기구부 요구 사항

1. 슬립링을 쓰지 않도록 회전 부분 설계
2. 각 종 보드 및 센서, 기구들을 담을 수 있는 충분한 크기
3. 동작 시 기구 손상이 일어나지 않도록 내구성 고려
4. 최대한 비용을 줄이기 위해 가지고 있는 부품들을 활용
5. 기구의 회전 속도는 20rpm~60rpm 사이



# 설계 전 결정된 사항



Motor / Encoder Type  
**DC GEARED ENCODER MOTOR**  
**IG42GM W/EC 02TYPE**

감속비 : 1/4 ~ 1/3600 (감속비율 총 21종)  
정격토크 : 1.8kg.cm ~ 30kg.cm  
정격회전수 : 1445 rpm ~ 1.9 rpm  
장착된모터 : DC 24V / 7000 rpm / 34.7 W motor  
엔코더사양 : 38Pulses (19Pulses x 2CH)

02 TYPE MOTOR (DC 24V)	
정격 토크 Rated torque	570 (gf-cm)
정격 회전수 Rated speed	5,900 (RPM)
정격 전류 Rated current	2,100 (mA)
무부하 회전수 No load speed	7,000 (RPM)
무부하 전류 No load current	500 (mA)
정격 출력 Rated output	34.7 (W)

감속기길이(mm) Gear Head L		32.5		39.2		45.9				
중량(g)		499		551		591				
감속비 Reduction ratio		1/4	1/14	1/17	1/24	1/49	1/61	1/84	1/104	1/144
02 TYPE 24V	정격토크(kgf-cm) Rated torque	1.8	5.4	6.6	9.5	16	18	18	20	20
	정격 회전수(RPM) Rated speed	1445	420	340	240	122	102	77.5	63	47
	무부하 회전수 (RPM) No Load speed	1750	500	411	291	142	114	83	67	48

1. 모터 IG42GM 기어비 1/24 모델 사용
  - 기존에 가지고 있던 IG42GM 모터를 사용하기로 함
  - 기어비 1/24 모델 기준으로 토크 9.5kgf-cm

## \*사용 가능 여부 체크

- 질량 약 15kg, 반지름 30cm 원형 구조물을 돌리고 질량은 모두 끝에 점질량으로 몰려 있다고 가정
- 요구 최대 회전 속도가 60rpm(=1rps= $2\pi \text{ rad/s}$ )
- 2초안에 목표속도에 도달한다고 했을 때 최대 각 가속도는  $1\pi \text{ rad/s}^2$
- 해당 각 가속도 일 때 점 질량 가속도는  $1\pi * 0.3 \text{ m/s}^2 = 0.94 \text{ m/s}^2$
- 이때 토크는  $(15 * 0.94 / 9.8) * 30 \text{ kgf-cm} = 43.16 \text{ kgf-cm}$
- 비가 1: 4.5 인 기어 세트 추가 설치 필요

# 설계 전 결정된 사항



- 2. 경창 1011S 캐스터 사용
  - 기존에 가지고 있던 캐스터 사용
  - 높이 4cm 허용하중 16kgf

- \* 사용 가능 여부 체크
  - 회전 판의 중량 약 15kgf라 했을 때 캐스터에 4개에 분산되어 걸리는 중량은 약 3.75kgf임으로 사용 가능

Caster 호칭번호 (CASTER PART NUMBER)		바퀴 사양 (WHEEL SPECIFICATION)					Caster 사양 (CASTER SPECIFICATION)		
회전 (SWIVEL)	고정 (RIGID)	직경 (DIAMETER)	두께 (TREAD WIDTH)	형식 (TYPE)	Bearing 종류 (BEARING)	사용 BOLT (AXLE DIA)	총높이 (OVERALL HEIGHT)	회전반경 (SWIVEL RADIUS)	허용하중 (kgf) (LOAD CAPACITY)
10 11SA1 LP	-	29 (1 - 1/8)	14 (9/16)	Light Duty Plastic	Plain	ø 4.8 (3/16)	40	22	16

사양	장착부 형식 (FASTENING TYPE)	장착부 형상 (FASTENING SHAPE)	CODE 번호 (CODE NO)	장착부 규격 (FASTENING SPECIFICATION)		
				밀판크기 (OVERALL SIZE)	BOLT 취부 간격 (BOLT HOLE SPACING)	조립볼트 규격 (BOLT SIZE)
회전	TOP PLATE (평판형)		1011 SA1 **	30 X 45	35 X 20	M4
			1020 SA1 **	65 X 48	54 X 24	M6
			1025 SA1 **	98 X 70	76,2 X 44,5	M6
			1030 SA1 **	104 X 79	85 X 60	M8
			1040 SA1 **	130 X 102	102 X 76	M8
	THREADED STEM (나사형)		1020 SB1 **	M12 X P1,25 XL44 (1020 SRS만 가능)		

# 설계 전 결정된 사항



- 축 외경 : Ø8mm
- 최대응답주파수 : 35kHz
- 최대허용회전수 : 3000rpm
- 기동토크 : 70gf.cm 이하(0.0069N.m 이하)
- 분해능 : 1024P/R
- 회전방향 : 반시계방향
- 제어출력 : NPN 오픈 콜렉터
- 전원전압 : 5VDC  $\pm 5\%$ (리플 P-P: 5% 이하)
- 접속 케이블사양 : 후면 케이블인출형
- 보호구조 : IP64

## 3. EP50S8-1024-2R-N-5 사용

- 기존에 가지고 있던 EP50S8 절대 엔코더를 사용하기로 함
- 분해능 1024P/R
- 5V전압 입력, NPN 오픈 컬렉터 임으로 레벨 컨버터 및 저항 등 추가적인 회로 필요

## \* 사용 가능 여부

- 회전당 1024 pulse임으로 360도 / 1024 = 0.35도/bit
- 최소 측정치가 0.35도임으로 15m 위치의 물체를 맞춘다고 했을 때  $30 * 0.35 * \frac{\pi}{180} * 100 \text{ cm} = 9.16 \text{ cm}$  측정 오차가 나게 된다.
- 목표물 크기가 20cm정도 임을 감안 했을 때 목표물 타격하기 충분하다.

# 설계 도중 추가 결정된 사항



- 축 외경 : Ø4mm
- 최대응답주파수 : 300kHz
- 최대허용회전수 : 5000rpm
- 기동토크 : 20gf.cm 이하(0.00196N.m 이하)
- 분해능 : 3000P/R
- 제어출력 : 전압
- 전원전압 : 5VDC  $\pm 5\%$ (리플 P-P: 5% 이하)
- 접속 케이블사양 : 후면 케이블인출형
- 보호구조 : IP50

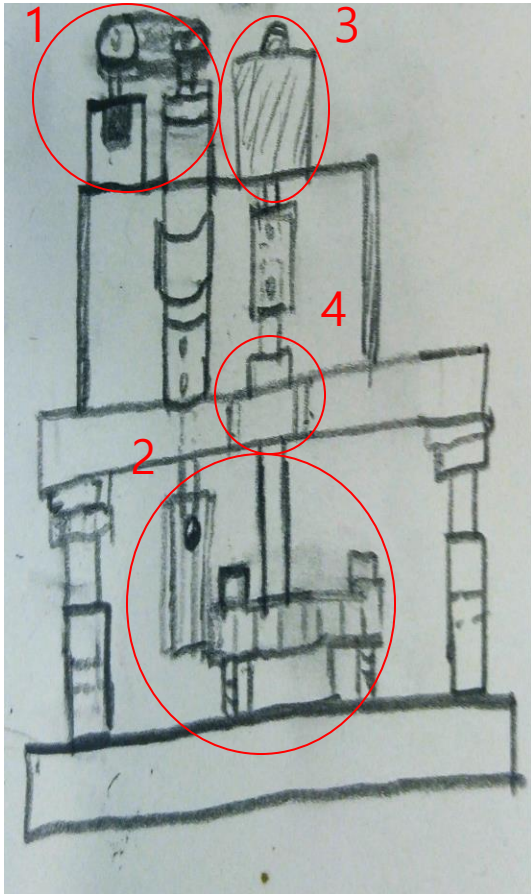
## 1. EP30S4-3000-3-V-5 사용

- 기존에 가지고 있던 EP30S4 엔코더를 사용하기로 함
- 엔코더에 부착 된 분해능으로는 제어가 힘들어 추가 설치하기로 함
- 분해능 3000 pulse/회전
- 5V 전압 출력 임으로 레벨 컨버터 필요

## \* 사용 가능 여부

- 회전당 3000 pulse임으로 측정 주기가 10ms라 했을 때 최소 측정 가능한 모터 rpm은  $\frac{1}{3000} * \frac{1000}{10} * \frac{60}{1} \text{ rpm} = 2 \text{ rpm} = 12^\circ/\text{sec}$
- 기어비가 유성기어 1/24 회전 축 기어 1/7.5임으로
- $\frac{12}{24} * \frac{1}{7.5}^\circ/\text{sec} = 0.07^\circ/\text{sec} = 0.01 \text{ rpm}$  을 제어할 수 있다.

# 기구부 회전 부분 구상도



회전부

## 1. 모터와 엔코더

- 모터 홀센서의 분해능이 부족하여 속도 측정 엔코더 별도로 설치
- 엔코더와 모터는 타이밍 벨트로 연결

- 바닥면과 회전판 사이의 높이 및 기어의 높이 조절 문제로 유성기어 축이 아니라 모터 축에 타이밍 풀리 설치

- 타이밍 풀리 회전 축은 모터의 기어비가 적용이 안됨

## 2. 회전판 기어

- 슬립링을 쓰지 않기 위해 모터 및 모든 부품들이 회전판 위에 올라감
- 회전판을 돌리기 위해서 하나의 기어는 움직이지 않는 판에 고정하고 모터에 기어를 달아서 고정된 기어의 둘레를 도는 형식으로 동작

## 3. 절대 엔코더

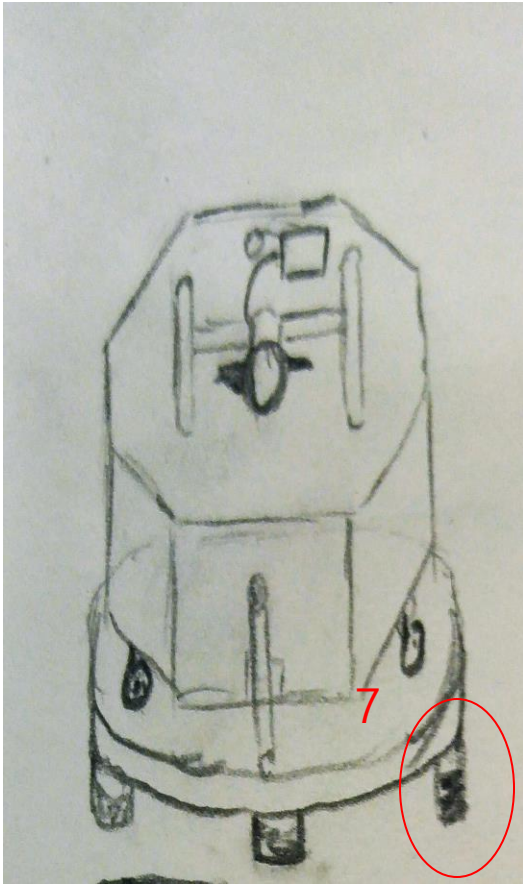
- 회전판의 현재 회전 각도를 절대 엔코더로 측정
- 절대 엔코더의 축과 고정 기어의 축 연결(엔코더 축 고정)
- 회전 시 엔코더의 몸체가 움직임

## 4. 베어링

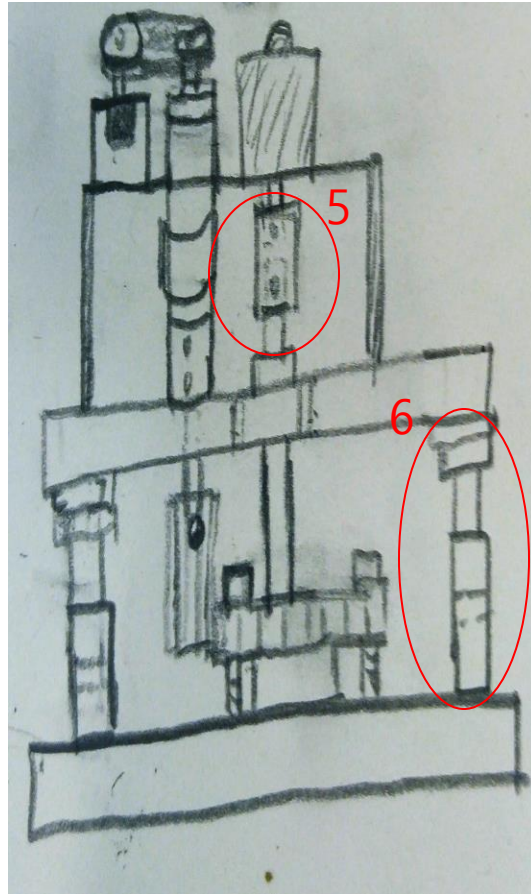
- 고정 축은 베어링으로 회전판의 움직임과 독립적으로 움직이도록 분리



# 기구부 회전 부분 구상도



전체 외형



회전 부

## 5. 커플러 설치

- 고정 기어의 축과 엔코더 축의 직경이 다르기 때문에 커플러 설치
- 엔코더 축에 걸릴 충격 감소 효과

## 6. 바퀴 설치

- 회전판에 바퀴를 설치하여 축에 걸릴 하중을 바퀴로 분산
- 기존에 가지고 있는 높이 4cm의 바퀴 케스터 사용
- 케스터 높이에 맞추어 바닥판과 회전판 사이 높이가 4cm가 되도록 설계

## 7. 높낮이 조절 받침

- 바닥 판에 높낮이 조절 받침을 설치하여 수동으로 포탑의 평형 유지

# 기구부 회전 부분 기어 구입 및 가공



회전 부분 기어 세트

1. 회전 판의 최대 회전 속도는 30rpm으로 설계
2. 모터의 정격속도 및 토크를 고려했을 때 1:7.5의 기어비를 가진 기성 기어 세트를 구입
3. 추가적으로 축을 고정시킬 탭 나사 밑 기어를 바단 판에 고정할 탭 나사 추가 가공

# 기구 바닥부 지지물 구매

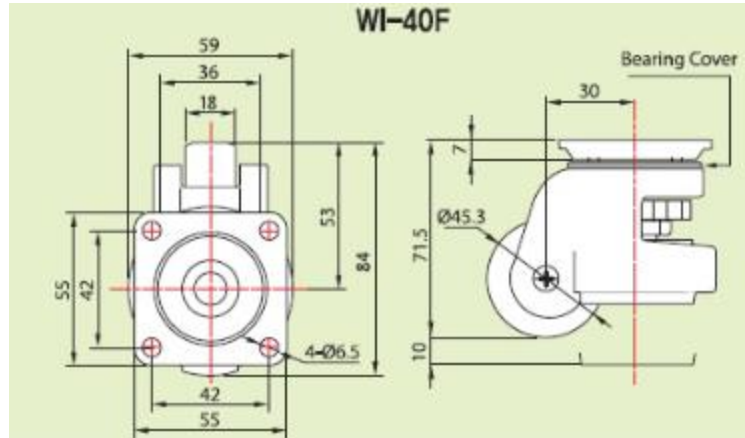


제원(Specification)

- 개당 허용 하중(Load/unit) : 50kg/EA
- 추천 허용 하중(Recommended) : 100kg/4EA
- 자중(Net Weight) : F-TYPE - 0.38kg / S-TYPE - 0.37kg
- 사용온도(Temperature range) : -10 ~ +90°C

특징(Features)

- 높낮이 조절기능
- 손쉬운 이동성·고정성·방진성









1. 포탑의 평형 상태를 조절하기 위하여 높낮이가 조절가능한 캐스터 구매

2. 4개의 WI-40F 사용

3. 4개를 사용했을 때 추천 하중이 100kg임으로 포탑의 예상 하중이 15kg임을 감안했을 때 충분히 버틸 수 있음



# 기구부 축, 타이밍 풀리 구입

<input checked="" type="checkbox"/>	No.	품명 관리 번호 [번호] 미스미 카탈로그 품명 상품명 ([재고]표기가 있는 상품은 재작품) 메이커	수량	단가 (원)	금액 (원)	(출하중소) 품명 ※ (고려도착 및 배송일) 거래일세표 번호	상태
<input checked="" type="checkbox"/>	1	 <a href="#">S8FRV15-70-F10-T14-W13-FC18-G</a> ROTARY SHAFT (98g) MISUMI	1	16,930	16,930	(한국미스미 유통센터) 2019/04/12 ▶ (2019/04/15) LN90170409K2	출하완료
		품명: [우정 코드] 6077070853973					
<input checked="" type="checkbox"/>	2	 <a href="#">HBT15</a> 베어링(원산지:일본)(재고) (125g) MISUMI	1	24,330	24,330	(한국미스미 유통센터) 2019/04/12 ▶ (2019/04/15) LN90170409K2	출하완료
		품명: [우정 코드] 6077070853973					
<input checked="" type="checkbox"/>	3	 <a href="#">SRBM-32-SX15</a> COUPLING SRBM (50g) 설립기준	1	14,710	14,710	(한국미스미 유통센터) 2019/04/12 ▶ (2019/04/15) LN90170409K2	출하완료
		품명: [우정 코드] 6077070853973					
<input checked="" type="checkbox"/>	4	 <a href="#">TBN88MXL025</a> BELT (1g) MISUMI	1	1,740	1,740	(한국미스미 유통센터) 2019/04/12 ▶ (2019/04/15) LN90170409K2	출하완료
		품명: [우정 코드] 6077070853973					
<input checked="" type="checkbox"/>	5	 <a href="#">ATP28MXL025-B-H4</a> STANDARD PULL (9g) MISUMI	1	7,390	7,390	(한국미스미 유통센터) 2019/04/12 ▶ (2019/04/15) LN90170409K2	출하완료
		품명: [우정 코드] 6077070853973					
<input checked="" type="checkbox"/>	6	 <a href="#">ATP20MXL025-B-H5</a> STANDARD PULL (4g) MISUMI	1	6,280	6,280	(한국미스미 유통센터) 2019/04/12 ▶ (2019/04/15) LN90170409K2	출하완료

구입 항목

한국 미스미에서 축과 베어링, 커플러, 타이밍 풀리 세트를 구입

1. 축  
축은 기어의 내경에 맞게 직경 15mm로 구입  
축을 볼트로 고정 시킬 수 있도록 베어링 위치 커플러 위치 등을 고려하여 볼트 고정 부분 마다 평면면취 추가 가공

2. 커플러 및 베어링  
커플러와 베어링은 축 직경에 맞추어 내경 15mm인 것을 구입  
베어링의 경우 회전 판에 고정할 수 있도록 볼트 구멍이 있는 것을 구입

3. 타이밍 풀리 세트  
모터의 기어비가 적용 안된 무부하시 속도 7000rpm과 엔코더 최대 동작 속도 5000rpm을 고려하여 기어비를 7:5로 맞춤

# 기구부 축, 타이밍 풀리 사전 조립



기어, 축, 커플러, 절대  
엔코더 결합



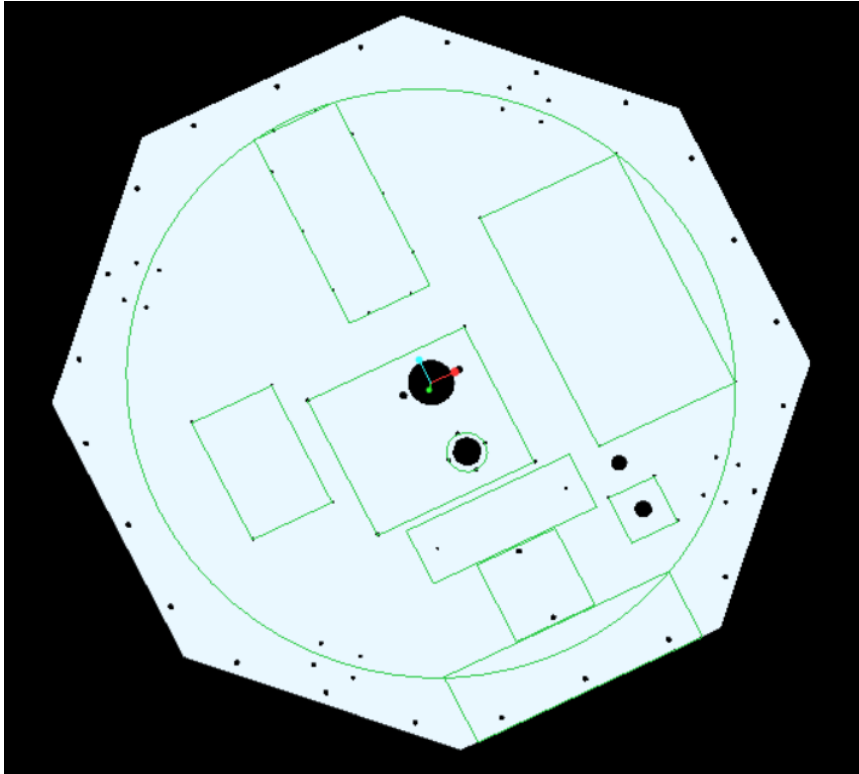
타이밍풀리와 엔코더, 모터 결합

구입한 축, 타이밍 풀리 세트가 절절히 왔는지  
확인하기 위하여 사전 조립

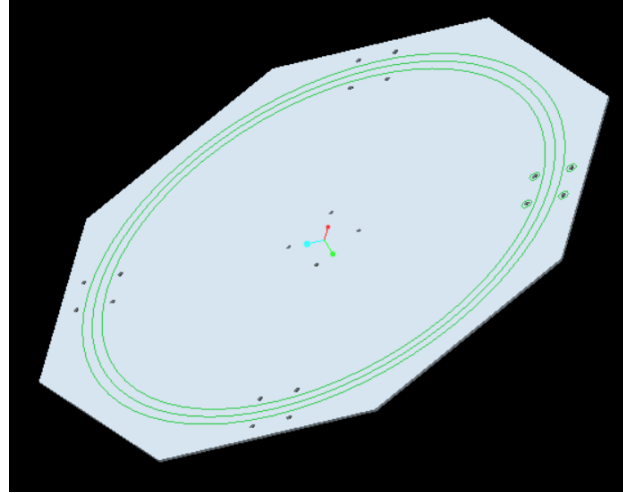
1. 축
  - 고정 기어와 축이 정상적으로 결합되는 것을 확인
  - 베어링 위치와 바닥면 사이의 간격을 재었을 때 평면 면취 가공 위치가 적절하게 배치되었음을 확인
2. 타이밍 풀리
  - 모터와 엔코더에 정상적으로 결합되는 것을 확인
  - 모터의 타이밍 풀리는 모터 축과 끼어 맞춤 식으로 결합되는 것을 확인
  - 엔코더의 타이밍 풀리의 경우 축 고정 나사 탭이 없어서 4mm 나사 탭 추가 가공을 함

# 4주차 진행상황

## 3D 모델링 작업



회전 판



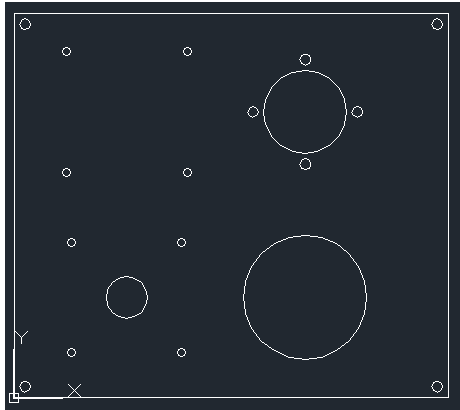
바닥 판

### 진행 상황

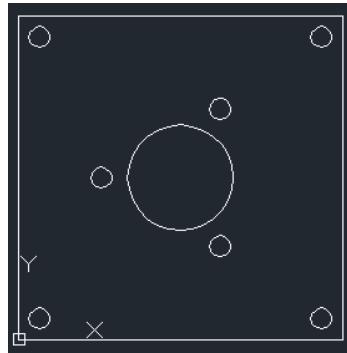
- 장치의 배치에 걸림이 없도록 실측 및 자료 검색을 통하여 장치 배치 공간을 고려하며 3D 모델링.

# 4주차 진행상황

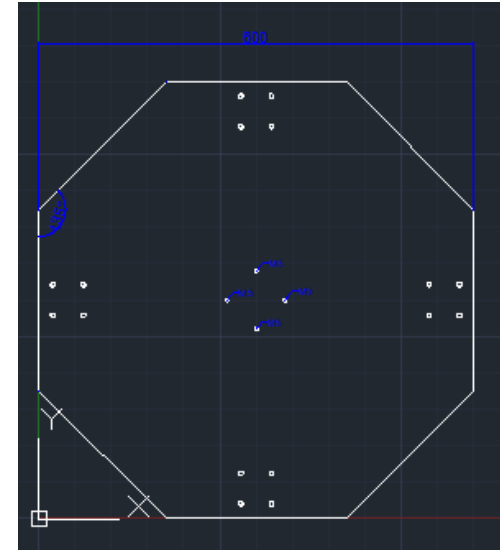
## 도면 작업



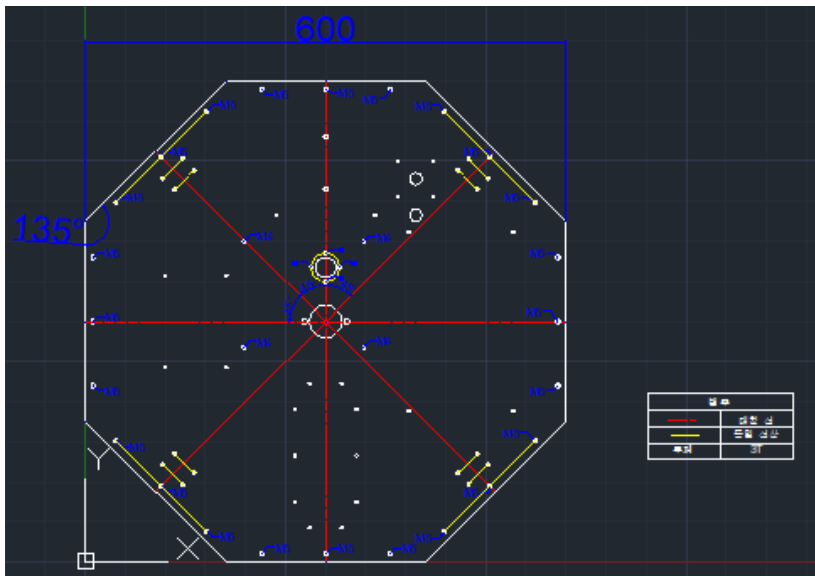
절대 엔코더 고정 판



속도 측정 엔코더 고정 판



바닥 판



회전 판

## 진행 상황

- 3D 모델링을 기반으로 회전부의 각 파트들을 도면으로 옮김.
- 추가 가공이 필요한 탭나사의 경우 M(볼트 직경)로 표시하여 탭을 낼 곳을 표시

# 4주차 진행상황

## 견적 요청

레이저	견적완료	견적요청번호: 800	견적코드: LAS20190426	8,600 원	▼
레이저	견적완료	견적요청번호: 797	견적코드: LAS20190425	26,000 원	▼
레이저	견적완료	견적요청번호: 796	견적코드: LAS20190425	132,900 원	▼
레이저	견적완료	견적요청번호: 794	견적코드: LAS20190425	185,000 원	▼

## 진행상황

- 작성한 도면을 첨부하여 디마이스 마트에 견적 요청
- 가공은 정확도 및 가격을 생각했을 때 레이저 가공을 선택
- 재료는 내구성을 고려하여 회전판은 3T, 바닥 판은 3.5T, 절대 엔코더 지지 판은 2T, 증분형 엔코더 고정 판은 1.5T 알루미늄 사용
- 회전판과 바닥 판에 대해서는 탭 나사 처리 및 아노다이징 요청

## 문제점 및 해결 방안

- 회전판 및 바닥판에 대해서는 크기 때문에 제작 비용이 너무 많이 드는 문제가 있음
- 내부에 들어가는 부품들이 많아 크기를 줄이는데 어려움이 있음
- 다른 업체에도 견적을 요청하여 가격을 비교 비용이 덜 드는 업체를 찾아 제작
- 기타 옆면 및 윗면에 대해서는 아크릴로 제작을 고려

FPGA

# Petalinux 드라이버 분석 및 조사

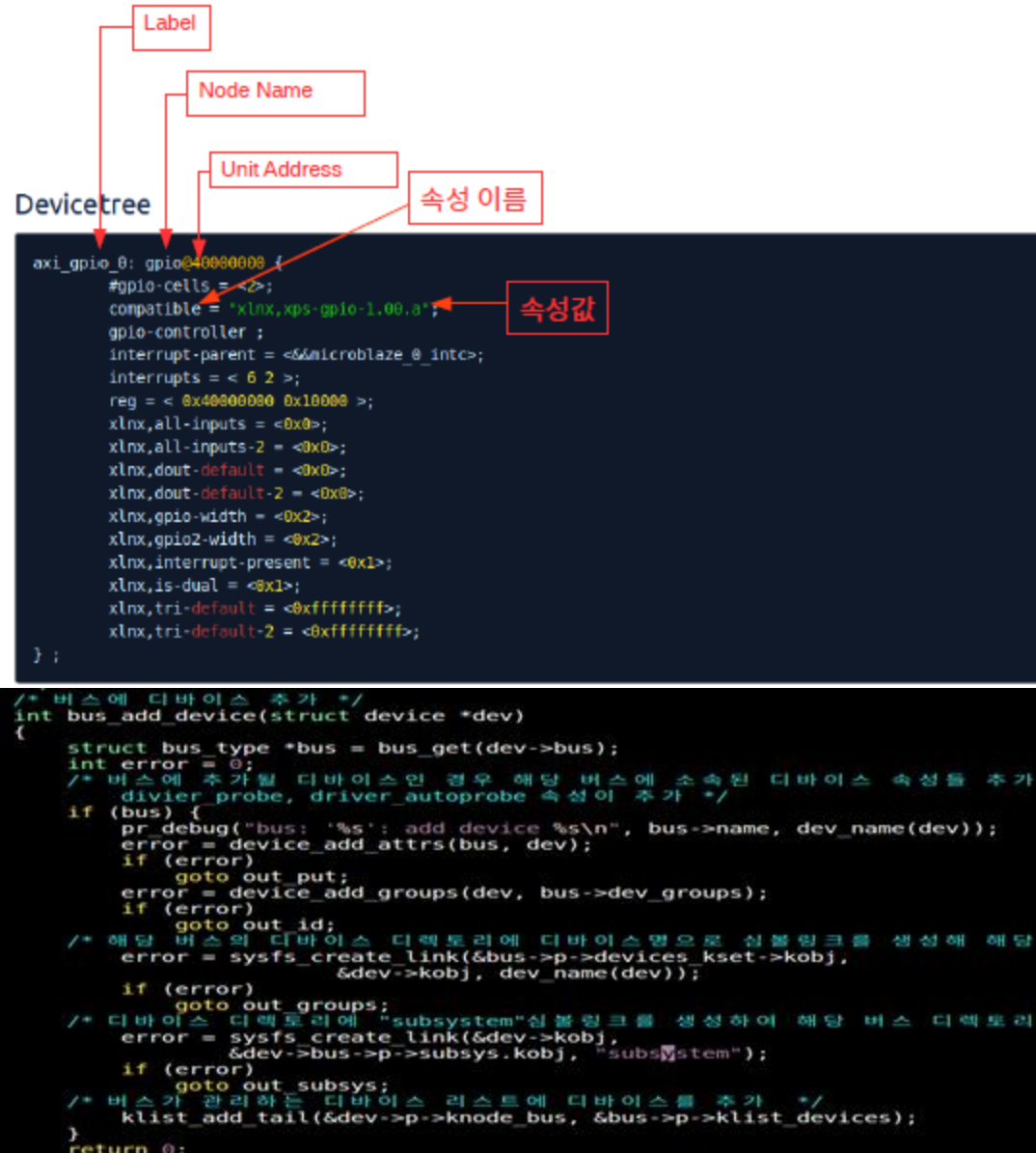
Axi-gpio를 사용하기 위하여  
디바이스 드라이버 작성 필요

디바이스 트리와 드라이버에 관해서  
조사 및 코드 분석 진행

Sysfs 방식으로 Gpio를 구현하려고 했  
으나 실패함

기존 UART통신을 사용하려고 했으나  
시간적으로 구현하기 어렵다고 봄

UART대신 FPGA의 USB포트를 사용하  
고 예비로 PMOD CAN을 사용하기로  
함



## 5주차 진행 상황

프로젝트 5주차  
in list [진행상황](#)

---

☰

### Description

Add a more detailed description...

☒ 4월 22일 월요일

Delete

0%

☐ FPGA GPIO 소스분석
 

Add an item

☒ 4월 23일 화요일

Hide completed items

Delete

100%

☒ ~~FPGA GPIO 10개 핀 LED 출력~~

Add an item

☒ 4월 24일 수요일

Delete

0%

☐ FPGA GPIO 10개 입력받기 진행 중  
☐ 셸프로그래밍 -> 캐릭터 드라이버 프로그래밍 완료
 

Add an item

☒ 4월 25일 목요일

Hide completed items

Delete

100%

☒ ~~FPGA GPIO 10개 핀 중 8개 입력 받음~~



# INPUT

입력 받을 PIN 10개

	Pmod JA	Pmod JB*	Pmod JC	Pmod JD	Pmod JE	Pmod JF
Pmod Type	XADC	High-Speed	High-Speed	High-Speed	Standard	MIO
Pin 1	N15	V8	V15	T14	V12	MIO-13
Pin 2	L14	W8	W15	T15	W16	MIO-10
Pin 3	K16	U7	T11	P14	J15	MIO-11
Pin 4	K14	V7	T10	R14	H15	MIO-12
Pin 7	N16	Y7	W14	U14	V13	MIO-0
Pin 8	L15	Y6	Y14	U15	U17	MIO-9
Pin 9	J16	V6	T12	V17	T17	MIO-14
Pin 10	J14	W6	U12	V18	Y17	MIO-15

```
led-app : input : 000003ff
```

초기값 PIN 10개      0011 1111 1111

```
led-app : input : 000000ff
```

실행. (입력 안 받았는데 PIN(9, 10) 신호 들어옴)

```
led-app : input : 000000fe
```

INPUT V12      0000 1111 1110

```
led-app : input : 000000fd
```

INPUT W16      0000 1111 1101

```
led-app : input : 000000fb
```

INPUT J15      0000 1111 1011

```
led-app : input : 000000f7
```

INPUT H15      0000 1111 0111

```
led-app : input : 000000ef
```

INPUT V13      0000 1110 1111

```
led-app : input : 000000df
```

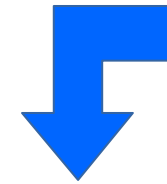
INPUT U17      0000 1101 1111

```
led-app : input : 000000bf
```

INPUT T17      0000 1011 1111

```
led-app : input : 0000007f
```

INPUT Y17      0000 0111 1111



0번부터 입력

# 프로젝트 5주차 문제점

프로젝트 5주차

in list 문제점

Description

Add a more detailed description...

☒

4월 22일 월요일

Hide completed items

Delete

100%

☒

10개 PIN 중 1개만 출력됨

Add an item

☒

4월 23일 화요일

Delete

0%

Add an item

☒

4월 24일 수요일

Delete

0%

☐

실행명령 없이 자동 실행 미완료

Add an item

☒

4월 25일 목요일

Delete

0%

☐

FPGA GPIO 9번 10번핀 INPUT 안됨

☐

petalinux build 안됨

# BUILD 오류

```
ERROR: fsbl-2017.4+gitAUTOINC+77448ae629-r0 do_compile: Function failed: do_compile (log file is located at /home/jbs/FPGA/PTC/led_sw/build/tmp/work/plnx_arm-xilinx-linux-gnueabi/fsbl/2017.4+gitAUTOINC+77448ae629-r0/temp/log.do_compile.21177)
ERROR: Logfile of failure stored in: /home/jbs/FPGA/PTC/led_sw/build/tmp/work/plnx_arm-xilinx-linux-gnueabi/fsbl/2017.4+gitAUTOINC+77448ae629-r0/temp/log.do_compile.21177
Log data follows:
| DEBUG: Executing python function xsct externalsrc_compile_prefunc
| NOTE: fsbl: compiling from external source tree /opt/pkg/petalinux/tools/hsm/data/embeddedsw
| DEBUG: Python function xsct externalsrc_compile_prefunc finished
| DEBUG: Executing shell function do_compile
| Starting xsdk. This could take few seconds... Picked up _JAVA_OPTIONS: -Duser.home=/home/jbs/FPGA/PTC/led_sw/build/tmp/xsctenv
| Eclipse:
| An error has occurred. See the log file
| /home/jbs/FPGA/PTC/led_sw/components/plnx_workspace/fsbl/.metadata/.log.
| XSCTHELPER INFO: Empty Workspace
| Starting xsdk. This could take few seconds... Picked up _JAVA_OPTIONS: -Duser.home=/home/jbs/FPGA/PTC/led_sw/build/tmp/xsctenv
| Eclipse:
| An error has occurred. See the log file
| /home/jbs/FPGA/PTC/led_sw/components/plnx_workspace/fsbl/.metadata/.log.
| timeout while establishing a connection with SDK
|   while executing
| "error "timeout while establishing a connection with SDK""
|   (procedure "getsdkchan" line 111)
|   invoked from within
| "getsdkchan"
|   (procedure "projects" line 35)
|   invoked from within
| "projects -clean -type $type -name $name"
|   (procedure "clean_n_build" line 2)
|   invoked from within
| "clean_n_build bsp $params(bspname)"
|   invoked from within
| "if { $params(ws) ne "" } {
|   #Local Work Space available
|   setws $params(ws)
|   if { [catch {importprojects $params(ws)} result] } {
|     puts "XSCTHELPER IN..."
|   }
|   (file "/home/jbs/FPGA/PTC/led_sw/build/tmp/work/plnx_arm-xilinx-linux-gnueabi/fsbl/2017.4+gitAUTOINC+77448ae629-r0/app.tcl" line 120)
| WARNING: exit code 1 from a shell command.
| ERROR: Function failed: do_compile (log file is located at /home/jbs/FPGA/PTC/led_sw/build/tmp/work/plnx_arm-xilinx-linux-gnueabi/fsbl/2017.4+gitAUTOINC+77448ae629-r0/temp/log.do_compile.21177)
ERROR: Task (/opt/pkg/petalinux/components/yocto/source/arm/layers/meta-xilinx-tools/recipes-bsp/fsbl/fsbl_git.bb:do_compile) failed with exit code '1'
NOTE: Tasks Summary: Attempted 2402 tasks of which 1873 didn't need to be rerun and 1 failed.
```

MCU

# MCU

모터 구동

PWM

eQEP

PID Control

통신

SCI

Ethernet

CAN

I/O

Trigger

ADC

Charge  
Limit

# MCU

모터 구동

PWM

eQEP

PID Control

통신

SCI

Ethernet

CAN

I/O

Trigger

ADC

Charge  
Limit

# 모터 구동

☒ **중분형 엔코더 확인** Hide completed items Delete

100%

☒ 동작 확인

☒ 분해능이 더 큰 엔코더를 연결

☒ 연결 한 뒤 엔코더 동작 확인

☒ Test 코드를 범용성 있게 작성

☒ PID 제어 코드 작성

Add an item

☒ **모터드라이브 확인** Hide completed items Delete

100%

☒ PWM 변경하면서 동작확인 완료

☒ PWM 클럭 1kHz 듀티는 때에 따라 바꿈

Add an item

☒ **DC 24V 모터 작동** Hide completed items Delete

100%

☒ 실험 완료

Add an item

☒ **문제점** Hide completed items Delete

100%

☒ 기존 엔코더의 Resolution이 너무 작음

☒ 모터 속도가 5400rpm으로 돌아야하는데 엔코더 측정 한계가 5000rpm임

Add an item

## Planetary Geared Type DC GEARED ENCODER MOTOR IG42GM W/EC 02TYPE

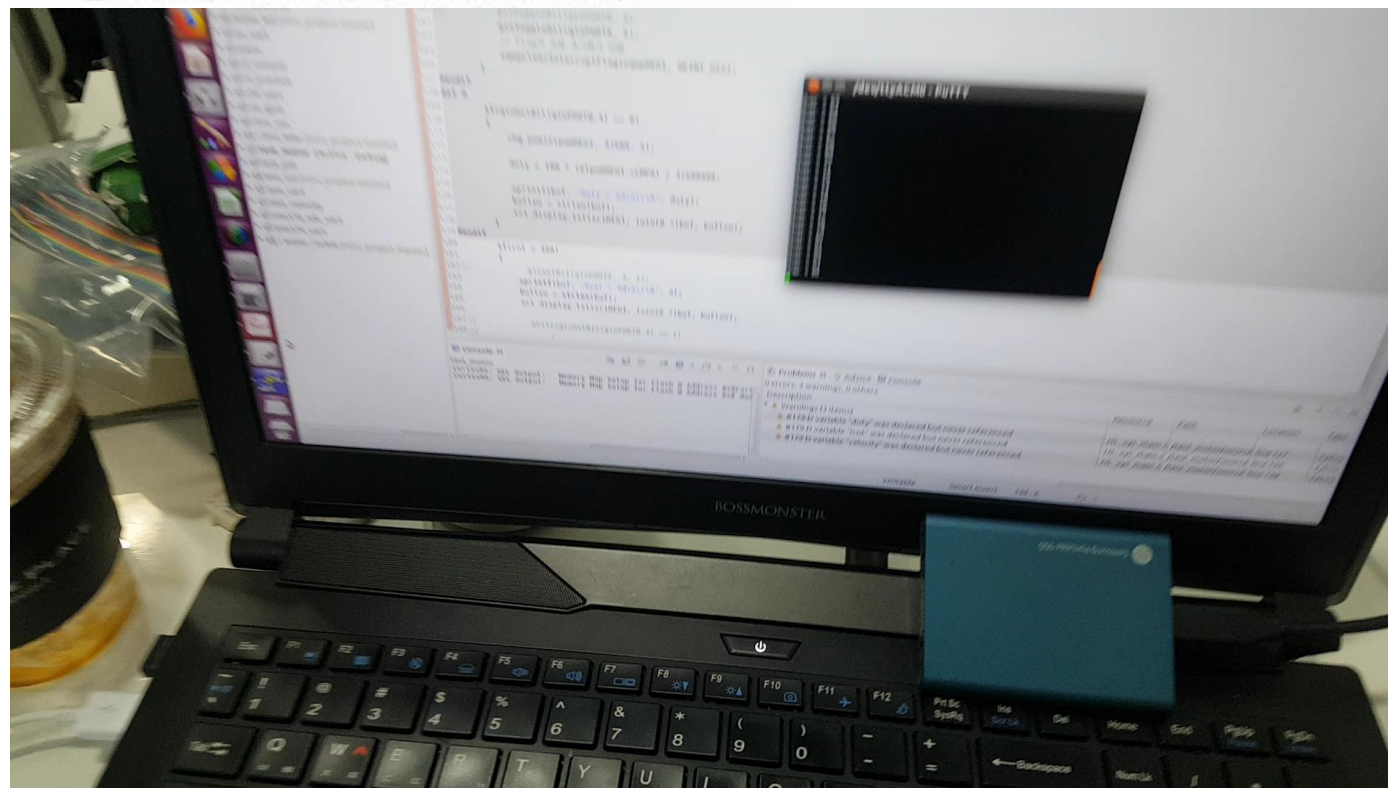
감 속 비 : 1/4 ~ 1/3600 (감속비율 총 21종)

정격토크 : 1.8kg.cm ~ 30kg.cm

정격회전수 : 1445 rpm ~ 1.9 rpm

장착된모터 : DC 24V / 7000 rpm / 34.7 W motor

엔코더사양 : 38Pulses ( 19Pulses x 2CH )





# 모터 구동

- 표기 된 엔코더 사양 : (19 Pulses x 2CH)
- Quadrature Mode
- $19 \text{ Pulses} * 24(\text{감속비}) = 456$
- $456 * 4 = 1824$
- 테스트 시 1바퀴에 QPOSCNT 값 : 480
- $480 / 24 (\text{감속비}) = 20$
- $20 / 4 = 5$

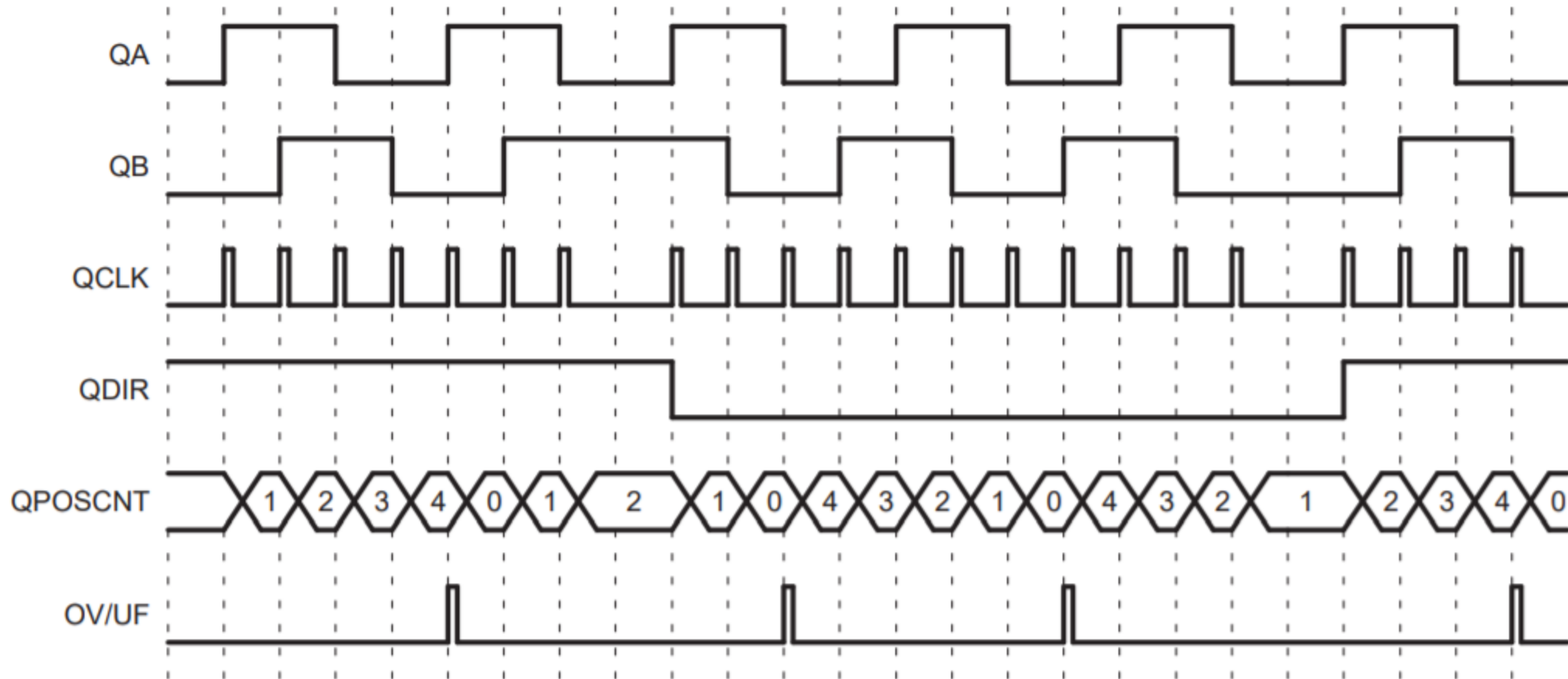


Planetary Geared Type  
**DC GEARED ENCODER MOTOR**  
**IG42GM W/EC 02TYPE**

감 속 비 : 1/4 ~ 1/3600 (감속비율 총 21종)  
정 격 토 크 : 1.8kg.cm ~ 30kg.cm  
정 격 회 전 수 : 1445 rpm ~ 1.9 rpm  
장 착 된 모 터 : DC 24V / 7000 rpm / 34.7 W motor  
엔 코 더 사 양 : 38Pulses ( 19Pulses x 2CH )

# Quadrature Mode

**Figure 34-9. Position Counter Underflow/Overflow (QPOSMAX = 4)**



# 모터 구동

☒ **증분형 엔코더 확인**

Hide completed itemsDelete

100%

☒ 동작 확인
☒ 분해능이 더 큰 엔코더를 연결
☒ 연결 한 뒤 엔코더 동작 확인
☒ Test 코드를 범용성 있게 작성
☒ PID 제어 코드 작성

Add an item

☒ **모터드라이브 확인**

Hide completed itemsDelete

100%

☒ PWM 변경하면서 동작확인 완료
☒ PWM 클럭 1kHz 듀티는 때에 따라 바꿈

Add an item

☒ **DC 24V 모터 작동**

Hide completed itemsDelete

100%

☒ 실험 완료

Add an item

☒ **문제점**

Hide completed itemsDelete

100%

☒ 기존 엔코더의 Resolution이 너무 작음
☒ 모터 속도가 5400rpm으로 돌아야하는데 엔코더 측정 한계가 5000rpm임

Add an item

※커플링의 결합 시 회전축 간의 결합오차(편심, 편각)가 크게 되면 커플링 및 엔코더의 수명이 단축될 수 있으므로 주의하십시오.

## 정격/성능

종류	외경 Ø38mm 축형 INCREMENTAL 코터리 엔코더		
분해능 (P/R)※1	100, 200, 360, 500, 1000, 1024, 3000		
전기적 사양	출력형	A, B, Z 상 (단, Line driver 출력은 A, A, B, B, Z, Z 상)	
	출력위상차	A, B 상 간의 위상차: $\frac{T}{4} = \frac{T}{8}$ (T=A 상의 1주기)	
	제어 출력	Totem pole 출력	• [Low] - 부하전류: 30mA 이하, 잔류전압: 0.4VDC 이하 • [High] - 부하전류: 10mA 이하, 출력전압 (전원전압 5VDC): (전원전압-2.0)VDC 이상 출력전압 (전원전압 12-24VDC): (전원전압-3.0)VDC 이상
		NPN 오픈 콜렉터 출력	부하전류: 30mA 이하, 잔류전압: 0.4VDC 이하
		전압출력	부하전류: 10mA 이하, 잔류전압: 0.4VDC 이하
		Line driver 출력	• [Low] - 부하전류: 20mA 이하, 잔류전압: 0.5VDC 이하 • [High] - 부하전류: -20mA 이하, 출력전압: 2.5VDC 이상
	응답 속도 (상승, 하강)	Totem pole 출력	1μs 이하 (배선길이: 2m, I sink=20mA 일 때)
		NPN 오픈 콜렉터 출력	1μs 이하 (배선길이: 2m, I sink=20mA 일 때)
		전압출력	1μs 이하 (5VDC: 출력저항 820Ω), 2μs 이하 (12-24VDC: 출력저항 4.7kΩ) (배선길이: 2m, I sink=20mA 일 때)
	기계적 사양	Line driver 출력	0.5μs 이하 (배선길이: 2m, I sink=20mA 일 때)
최대응답주파수		300kHz	
전원전압		• 5VDC ±5% (리플 P-P: 5% 이하)      • 12-24VDC ±5% (리플 P-P: 5% 이하)	
소비전류		80mA 이하 (무부하시), Line driver 출력일 경우 50mA 이하 (무부하시)	
절연저항		100MΩ 이상 (전단자와 케이스간 500VDC 메거)	
내전압		750VAC 50/60Hz에서 1분간 (전단자와 케이스간)	
접속방식		후면 배선인출형, 후면 배선인출 키넥터형	
기동토크		20gf·cm (0.002 N·m) 이하	
관성모멘트		20g·cm <sup>2</sup> (2×10 <sup>-6</sup> kg·m <sup>2</sup> ) 이하	
축하중하중		Radial: 2kgf 이하, Thrust: 1kgf 이하	
최대허용회전수※2	5000rpm		
내전동	10-55Hz (1기 1분간) 복전폭 1.5mm X, Y, Z 각 방향 2시간		
내충격	약 50G 이하		
내환경성	사용주위온도	-10~70℃, 보존 시: -25 ~ 85℃	
	사용주위습도	35~85%RH, 보존 시: 35~90%RH	
보호구조	IP50 (IEC 규격)		
배선사양	Ø5mm, 5심 (Line driver 출력: 8심), 2m, 원드 케이블 (AWG24, 소선지름: 0.08mm, 소선수: 40, 절연체 외경: Ø1mm)		
부속품	Ø4mm 커플링		
획득규격	CE (단, Line driver 출력은 제외)		
중량	약 80g		

※1: 이외의 분해능은 주문에 의합니다.

※2: 최대응답회전수 ≥ 최대응답회전수 조건이 되도록 분해능을 선정하십시오. 【 최대응답회전수 (rpm) =  $\frac{\text{최대응답주파수}}{\text{분해능}} \times 60 \text{ sec}$  】

※내환경성의 사용조건은 절방 또는 절로되지 않는 상태입니다.

# 모터 구동

☒ **증분형 엔코더 확인** Hide completed items Delete

100%

☒ 동작 확인

☒ 분해능이 더 큰 엔코더를 연결

☒ 연결 한 뒤 엔코더 동작 확인

☒ Test 코드를 범용성 있게 작성

☒ PID 제어 코드 작성

Add an item

☒ **모터드라이브 확인** Hide completed items Delete

100%

☒ PWM 변경하면서 동작확인 완료

☒ PWM 클럭 1khz 듀티는 때에 따라 바꿈

Add an item

☒ **DC 24V 모터 작동** Hide completed items Delete

100%

☒ 실험 완료

Add an item

☒ **문제점** Hide completed items Delete

100%

☒ 기존 엔코더의 Resolution이 너무 작음

☒ 모터 속도가 5400rpm으로 돌아야하는데 엔코더 측정 한계가 5000rpm임

Add an item



# 모터 구동

☒ **증분형 엔코더 확인** Hide completed items Delete

100%

☒ 동작 확인

☒ 분해능이 더 큰 엔코더를 연결

☒ 연결 한 뒤 엔코더 동작 확인

☒ Test 코드를 범용성 있게 작성

☒ PID 제어 코드 작성

Add an item

☒ **모터드라이브 확인** Hide completed items Delete

100%

☒ PWM 변경하면서 동작확인 완료

☒ PWM 클럭 1kHz 듀티는 때에 따라 바꿈

Add an item

☒ **DC 24V 모터 작동** Hide completed items Delete

100%

☒ 실험 완료

Add an item

☒ **문제점** Hide completed items Delete

100%

☒ 기존 엔코더의 Resolution이 너무 작음

☒ 모터 속도가 5400rpm으로 돌아야하는데 엔코더 측정 한계가 5000rpm임

Add an item

기어비      7.5 : 1 (30rpm)  
              24 : 1 (225rpm)  
              5400 rpm  
= 원판을 30rpm 속도로 돌리기  
위한 모터 속도



# 모터 구동

☒ **증분형 엔코더 확인** Hide completed items Delete

100%

☒ 동작 확인

☒ 분해능이 더 큰 엔코더를 연결

☒ 연결 한 뒤 엔코더 동작 확인

☒ Test 코드를 범용성 있게 작성

☒ PID 제어 코드 작성

Add an item

☒ **모터드라이브 확인** Hide completed items Delete

100%

☒ PWM 변경하면서 동작확인 완료

☒ PWM 클럭 1khz 듀티는 때에 따라 바꿈

Add an item

☒ **DC 24V 모터 작동** Hide completed items Delete

100%

☒ 실험 완료

Add an item

☒ **문제점** Hide completed items Delete

100%

☒ 기존 엔코더의 Resolution이 너무 작음

☒ 모터 속도가 5400rpm으로 돌아야하는데 엔코더 측정 한계가 5000rpm임

Add an item

## 최종 표준 PID 알고리즘

$$\text{Error} = \text{SetPoint} - \text{Input}$$

$$\text{PTerm} = K_p \times \text{Error}$$

$$\text{ITerm} += K_i \times \text{Error} \times dt$$

$$d\text{Input} = \text{Input} - \text{prevInput}$$

$$\text{DTerm} = -K_d \times (d\text{Input} / dt)$$

$$\text{Output} = \text{PTerm} + \text{ITerm} + \text{DTerm}$$



# PID Control

## 최종 표준 PID 알고리즘

Error = SetPoint - Input

PTerm =  $K_p \times \text{Error}$

ITerm +=  $K_i \times \text{Error} \times dt$

dInput = Input - prevInput

DTerm =  $-K_d \times (dInput / dt)$

Output = PTerm + ITerm + DTerm

```
109 #define Kp  2.63;
110 #define Ki  8.4;
111 #define Kd  0.00015;
112
113 int error[2];
114 float ierr;
115 float derr;
116
117 int pwm_CNT; // CMPA로 들어가는 누적값.
118 float P_term;
119 float I_term;
120 float D_term;
121
122 void const_velocity(int preCNT, int setCNT, int *error, float c_time)
123 {
124     error[0] = setCNT - preCNT;
125     ierr += (float)error[0] * c_time;
126     derr = (float)(error[0] - error[1]) / c_time;
127     P_term = (float)error[0] * Kp;
128     I_term = ierr * Ki;
129     D_term = derr * Kd;
130
131     error[1] = error[0];
132
133     pwm_CNT = (int)(P_term + I_term + D_term);
134
135     if(pwm_CNT < 0)
136     {
137         pwm_CNT = -pwm_CNT;
138     }
139     if(pwm_CNT > 37500)
140     {
141         pwm_CNT = 37500;
142     }
143
144     etpwmREG4->CMPA = pwm_CNT;
145 }
146 /*****
```

# MCU

모터 구동

PWM

eQEP

PID Control

통신

SCI

Ethernet

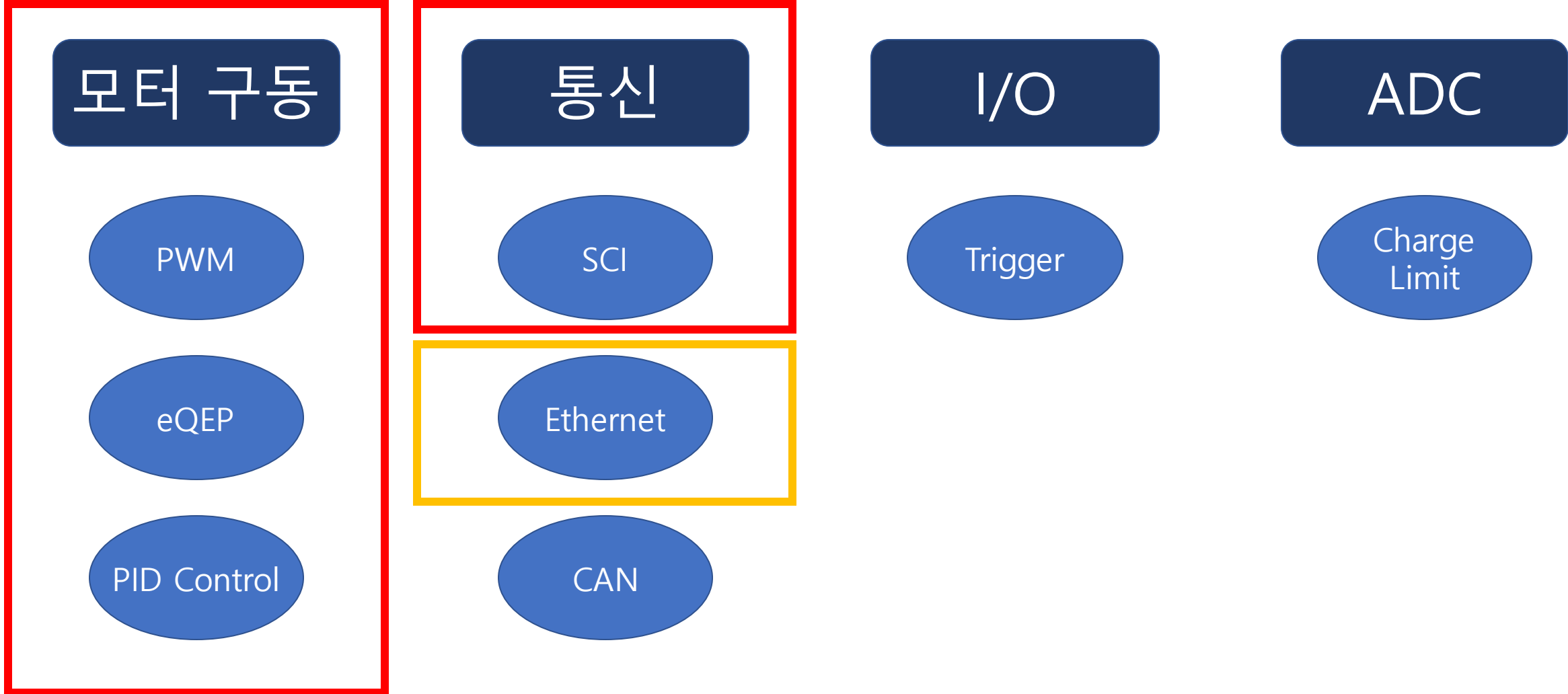
CAN

I/O

Trigger

ADC

Charge  
Limit





# Ethernet

```
/dev/ttyACM0 - PuTTY
DEBUG - Getting PHY ID...SUCCESS
DEBUG - Getting PHY Alive Status...SUCCESS
DEBUG - Getting PHY Link Status...SUCCESS
DEBUG - Setting up Link...SUCCESS

Device IP Address: 192.168.0.104

DEBUG - Getting PHY ID...SUCCESS
DEBUG - Getting PHY Alive Status...SUCCESS
DEBUG - Getting PHY Link Status...SUCCESS
DEBUG - Setting up Link...SUCCESS

Device IP Address: 192.168.0.104

Terminal File Edit View Search Terminal Help
Petalinux environment set to '/opt/pkg/petalinux'
WARNING: /bin/sh is not bash!
bash is Petalinux recommended shell. Please set you
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
WARNING: No tftp server found - please refer to "Pe
e" for its impact and solution
leedh@leedh-N15:~$ putty
leedh@leedh-N15:~$ putty
```

```
/dev/ttyACM0 - PuTTY
HighPri-LED3 666 <1%
Tmr Svc 56 <1%
LowPri-LED2 1331 <1%
UDP/IP 1103082 <1%
EMAC 391 <1%
Initializing GIO...
Initializing IP Layer...
...SUCCESS
Creating Tasks...
Starting Scheduler...
IP Address: 10.42.0.30
Subnet Mask: 255.0.0.0
Gateway IP Address: 10.42.0.1
DNS server IP Address: 208.67.222.222
Task Execution Statistics
-----
DebugTask 1 <1%
LowPri-LED2 30 <1%
IDLE 899716 47%
Tmr Svc 4 <1%
HighPri-LED3 16 <1%
UDP/IP 1011512 52%
EMAC 3 <1%
Task Execution Statistics
-----
DebugTask 28894 1%
IDLE 1799484 63%
HighPri-LED3 31 <1%
Tmr Svc 4 <1%
LowPri-LED2 61 <1%
UDP/IP 1011529 35%
EMAC 25 <1%
Task Execution Statistics
-----
DebugTask 57773 1%
IDLE 2699312 71%
LowPri-LED2 92 <1%
Tmr Svc 4 <1%
HighPri-LED3 46 <1%
EMAC 25 <1%
UDP/IP 1011529 26%
Task Execution Statistics
-----
DebugTask 86648 <1%
IDLE 32668941 96%
HighPri-LED3 62 <1%
Tmr Svc 8 <1%
LowPri-LED2 123 <1%
UDP/IP 1011537 2%
EMAC 31 <1%
Task Execution Statistics
-----
DebugTask 115513 <1%
IDLE 33568771 96%
LowPri-LED2 154 <1%
Tmr Svc 8 <1%
HighPri-LED3 77 <1%
```

```
/dev/ttyACM0 - PuTTY
pwd:
Print Working Directory

[Press ENTER to execute the previous command again]
>

help:
Lists all the registered commands

ps (process status):
Displays a table showing the state of each FreeRTOS task/process.

top (top cpu tasks/processes):
Displays a table showing how much processing time each FreeRTOS task has used.

emacstat:
Display network interface related statistics.

ping <ipaddress> [<optional:bytes to send>]:
for example, ping 192.168.0.3 8, or ping www.example.com

netstat:
Calls FreeRTOS_netstat()

reset:
Performs warm reset.

dir:
Lists the files in the current directory

cd <dir name>:
Changes the working directory

type <filename>:
Prints file contents to the terminal

del <filename>:
Deletes a file (use rmdir to delete a directory)

rmdir <directory name>:
Deletes a directory

copy <source file> <dest file>:
Copies <source file> to <dest file>

pwd:
Print Working Directory

[Press ENTER to execute the previous command again]
>ps
Task State Priority Stack #
-----
UartCmd R 6 414 3
IDLE R 0 102 4
TCPSrv B 3 1448 8
HeartBeat B 0 742 1
IP-task B 5 210 2
UdpCmd B 4 446 7
NtpClient B 4 58 9
LogTask B 2 862 5
EmacRx B 6 340 6

[Press ENTER to execute the previous command again]
```

# MCU

RTOS

TASK 2

모터 구동

PWM

eQEP

PID Control

TASK 3

통신

SCI

Ethernet

CAN

I/O



Trigger


ADC

Charge  
Limit

TASK 1

# RTOS

 **RTOS**  
in list [MCU](#) 

 **Description**  

Add a more detailed description...

☒ **RTOS 기본적인 구현** Hide completed items Delete  
100%

- ☒ 기본 동작 구현
- ☒ PWM + UART
- ☒ 10ms마다 PID하면서 UART
- ☒ UDP 동작 확인
- ☒ TCP 동작 확인

Add an item

☒ **RTOS 환경에서 노트북과 EMAC 통신** Delete  
0%

- ☐ 수업시간에 했던 메시지 보내는 코드 작성해보기
- ☐ PID로 얻은 데이터를 통신

Add an item



☒ **CAN 통신 구현** Delete  
0%


- ☐ 수업시간에 했던 CAN 통신 복습
- ☐ RTOS 환경에서 재구현

Add an item

```
181 /* Task1 */
182 void vTask1(void *pvParameters)
183 {
184     for(;;)
185     {
186         xSocket_t socket = FreeRTOS_socket(FREERTOS_AF_INET, FREERTOS SOCK_DGRAM, FREERTOS_IPPROTO_UDP);
187         if(socket == FREERTOS_INVALID_SOCKET)
188         {
189             sprintf(buf, "SOCKET is INVALID\r\n");
190             sciDisplayText(sciREG1, (uint8_t *)buf);
191         }
192         gpioSetBit(gioPORTB, 6, gioGetBit(gioPORTB, 6) ^ 1);
193         gpioToggleBit(gioPORTA, 5);
194         // Delay는 ms 단위.
195         vTaskDelay(100);
196     }
197 }
220 /* Task3 */
221 void vTask3(void *pvParameters)
222 {
223     for(;;)
224     {
225         sprintf(buf, "CMPA = %d\t Duty = %.1f%%\n\r\0", pwm_CNT, duty);
226         sciDisplayText(sciREG1, (uint8_t *)buf);
227
228         sprintf(buf, "POSCNT = %d\n\r\0", pcnt);
229         sciDisplayText(sciREG1, (uint8_t *)buf);
230
231         sprintf(buf, "setCNT = %d,\t duty = %d\n\r\0", setCNT, (pcnt * 100) / setCNT);
232         sciDisplayText(sciREG1, (uint8_t *)buf);
233
234         sprintf(buf, "Motor_Velocity = %f\n\r\0", motor_vel);
235         sciDisplayText(sciREG1, (uint8_t *)buf);
236
237         sprintf(buf, "Velocity = %f\n\r\0", velocity);
238         sciDisplayText(sciREG1, (uint8_t *)buf);
239
240         vTaskDelay(300);
241     }
242 }
```

# RTOS

 **RTOS**  
in list [MCU](#) 

 **Description**  

Add a more detailed description...

☒ **RTOS 기본적인 구현**

Hide completed items Delete

100%

☒ 기본 동작 구현

☒ PWM + UART

☒ 10ms마다 PID하면서 UART

☒ UDP 동작 확인

☒ TCP 동작 확인

Add an item

☒ **RTOS 환경에서 노트북과 EMAC 통신**

Delete

0%

☐ 수업시간에 했던 메시지 보내는 코드 작성해보기

☐ PID로 얻은 데이터를 통신

Add an item

☒ **CAN 통신 구현**

Delete

0%

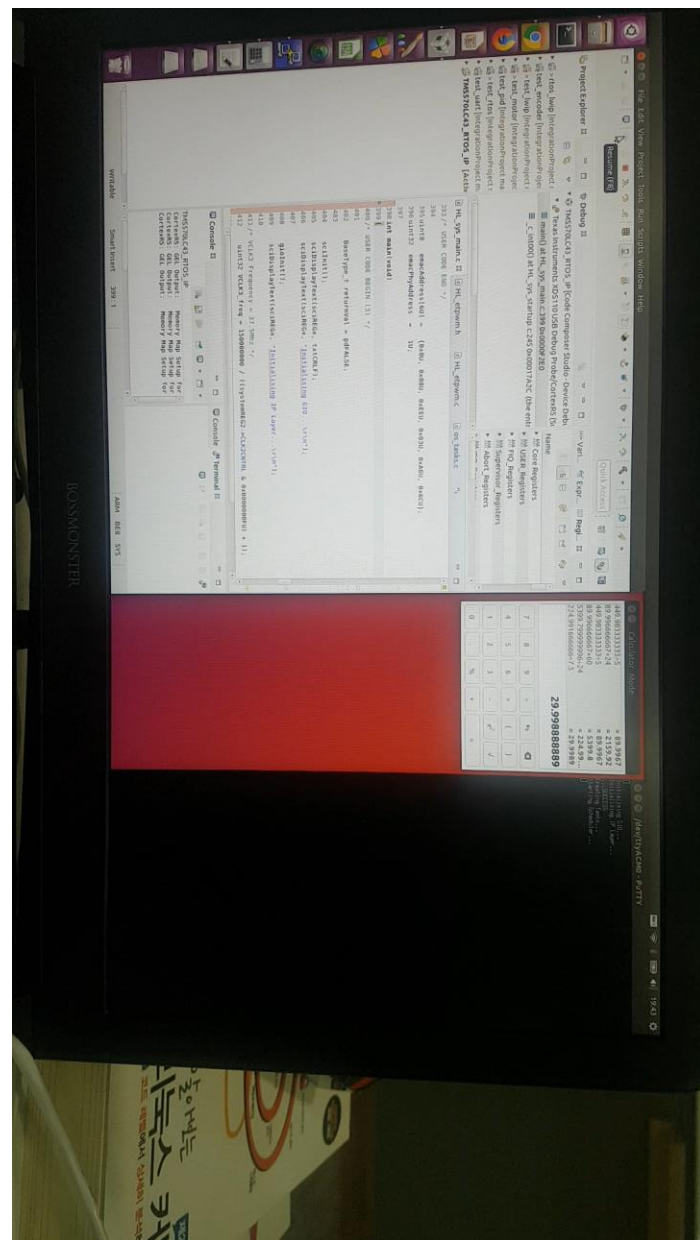
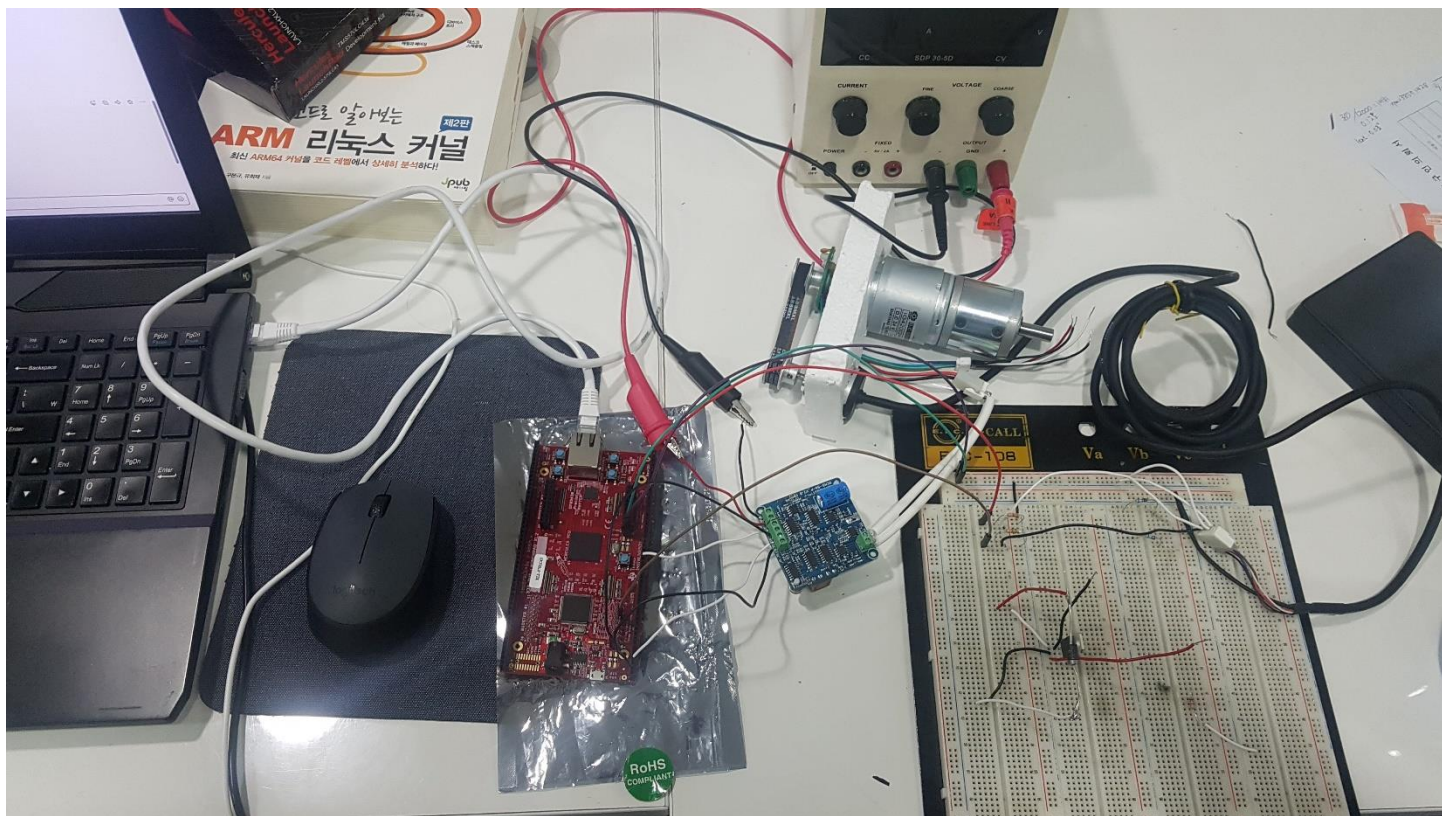
☐ 수업시간에 했던 CAN 통신 복습

☐ RTOS 환경에서 재구현

Add an item

```
199 /* Task2 */
200 void vTask2(void *pvParameters)
201 {
202     for(;;)
203     {
204         if((eqepREG1->QFLG & 0x800) == 0x800)
205         {
206             pcnt = eqepReadPosnLatch(eqepREG1); // 정해놓은 시간동안 들어온 CNT 갯수
207             motor_vel = (((float)pcnt * cpd / c_time) / 360.0) * 7.0 / 5.0) * 60.0; // 모터 엔코더 속도
208             velocity = motor_vel / 24.0 / 7.5; // 원판 돌아가는 속도
209             const_velocity(pcnt, setCNT, error, c_time);
210
211             duty = pwm_CNT * 100 / PWM_freq;
212
213             // Flag가 자동 초기화가 안됨.
214             eqepClearInterruptFlag(eqepREG1, QEINT_Uto);
215             vTaskDelay(10);
216         }
217     }
218 }
```

# 최종 구현 영상



코일건



# Coil Gun

## 개발 진행 및 주의사항

in list [Coil gun](#)

### Description Edit

주의사항!!!

1. 회로 테스트 및 작업시 쇼트 주의
2. 절연장갑 착용!!!
3. 사람에게 총구를 향하지 않기
4. DC-DC convertor 3개 전부 고장  
(2개-과전압, 1개 다른 컨버터와 쇼트로 추정)

### 할일

Hide completed items

Delete

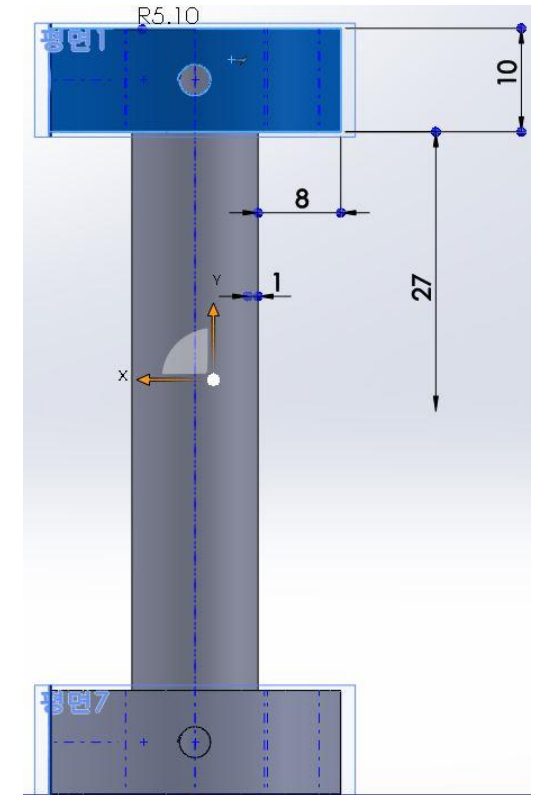
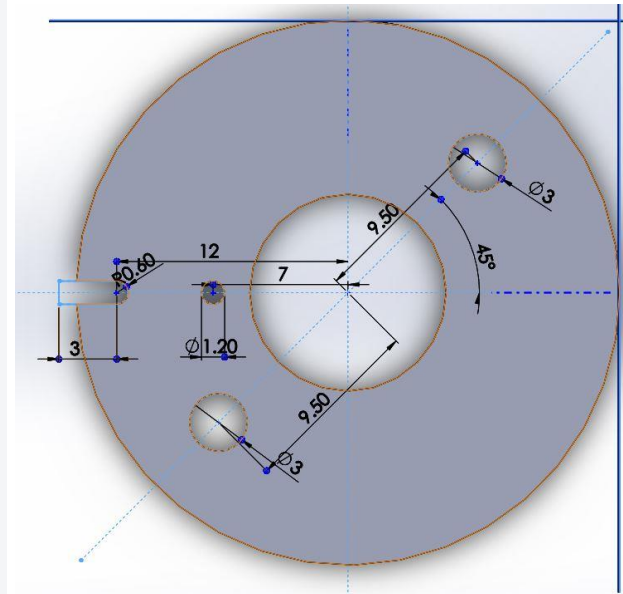
50%

- ☒ dc-dc 테스트
- ☒ 총알 결정 및 구매
- ☐ 총열 보정 및 거리측정
- ☐ 느린 충전속도 개선할것!! // 병렬 사용 중...
- ☐ ne555 구매후 교체
- ☒ Charge Limit을 위한 전압분배 회로 구현

Add an item

Prob 1. 아크릴(강성이 약해 휘어짐)  
Sol 1. 카본(탄소) 소재로 변경

Prob 2. 잦은 Coil Accelerator 수정  
Sol 2. 3D프린터로 보빈을 만들



# Coil Gun

☒ 개발 진행 및 주의사항

in list [Coil gun](#)

≡

Description

Edit

주의사항!!!  
1. 회로 테스트 및 작업시 쇼트 주의  
2. 절연장갑 착용!!!  
3. 사람에게 총구를 향하지 않기  
4. DC-DC convertor 3개 전부 고장  
(2개-과전압, 1개 다른 컨버터와 쇼트로 추정)

☒ 할일

Hide completed items

Delete

50%

☒ dc-dc 테스트

☒ 총알 결정 및 구매

☐ 총열 보정 및 거리측정

☐ 느린 충전속도 개선할것!! // 병렬 사용 중...

☐ ne555 구매후 교체

☒ Charge Limit을 위한 전압분배 회로 구현

Add an item

Prob 1. 아크릴(강성이 약해 휘어짐)  
Sol 1. 카본(탄소) 소재로 변경

Prob 2. 잦은 Coil Accelerator 수정  
Sol 2. 3D프린터로 보빈을 만듦





# Coil Gun

## 개발 진행 및 주의사항

in list [Coil gun](#)

### Description Edit

주의사항!!!

1. 회로 테스트 및 작업시 쇼트 주의
2. 절연장갑 착용!!!
3. 사람에게 총구를 향하지 않기
4. DC-DC convertor 3개 전부 고장  
(2개-과전압, 1개 다른 컨버터와 쇼트로 추정)

### 할일

Hide completed items

Delete

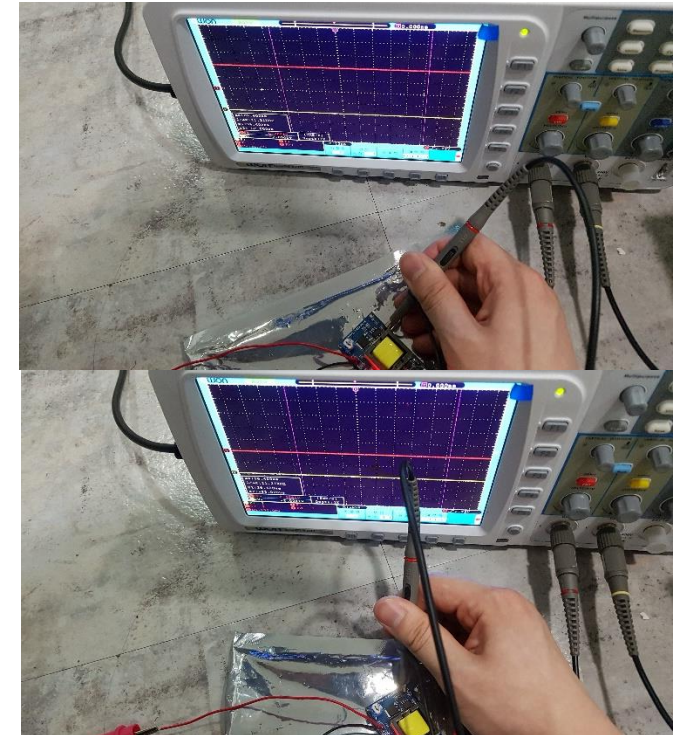
50%

- ☒ dc-dc 테스트
- ☒ 총알 결정 및 구매
- ☐ 총열 보정 및 거리측정
- ☐ 느린 충전속도 개선할것!! // 병렬 사용 중...
- ☐ ne555 구매후 교체
- ☒ Charge Limit을 위한 전압분배 회로 구현

Add an item

Prob 3. 발사 실험 도중 DC-DC 컨버터 고장  
NE555 발진 X -> NE555 구매

Cuz 1. 실험 중 전원 조작 실수로 과전압 인가  
Cuz 2. 컨버터 간 쇼트 의심.



# Coil Gun

## 개발 진행 및 주의사항

in list [Coil gun](#)

### Description Edit

주의사항!!!

1. 회로 테스트 및 작업시 쇼트 주의
2. 절연장갑 착용!!!
3. 사람에게 총구를 향하지 않기
4. DC-DC convertor 3개 전부 고장  
(2개-과전압, 1개 다른 컨버터와 쇼트로 추정)

### 할일

Hide completed items

Delete

50%

✓ *dc-dc 테스트*

✓ *총알 결정 및 구매*

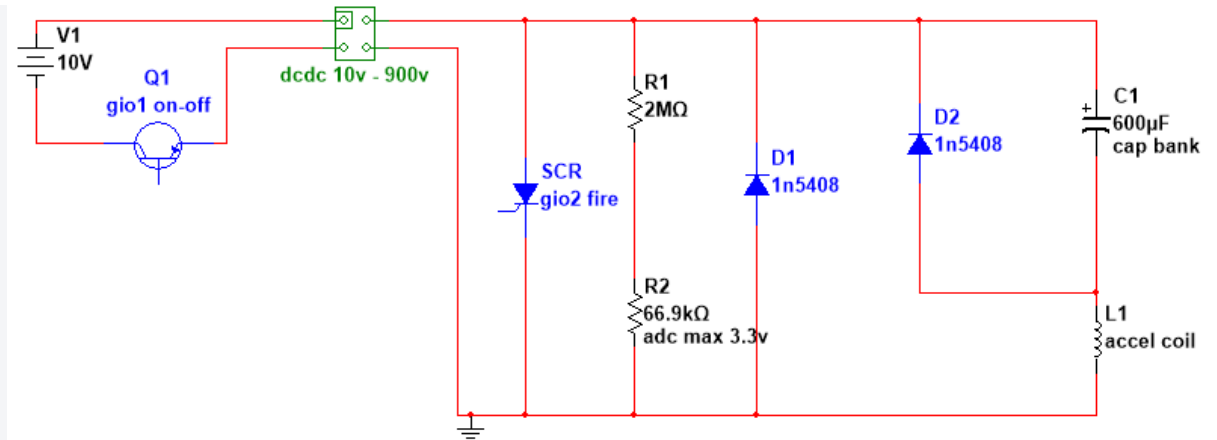
☐ 총알 보정 및 거리측정

☐ 느린 충전속도 개선할것!! // 병렬 사용 중...

☐ ne555 구매후 교체

✓ *Charge Limit을 위한 전압분배 회로 구현*

Add an item



$$R1 = 2.002\text{M}\Omega, R2 = 6.69\text{k}\Omega$$

$$R2 / (R1 + R2) * 1000\text{V} = \text{약 } 3.3\text{V}$$

$V_{r2} = (R2 / (R1 + R2)) * \text{cap-bank-voltage} \dots$  1번  
전압분배한 R2전압  $V_{r2} = (\text{adc} / 4095) * 3.3 \dots$  2번  
adc로 읽은 R2전압 1번과 2번 연립하여 정리

$$\text{cap-bank-voltage} = \text{adc} * 0.24176$$

# Coil Gun 진행 예정

1. NE555 교체 후 테스트
2. 회로 구현시 절연 방법 고민
3. DC-DC 회로 구현 방법(위치, 절연등)
4. 발사 테스트 15m거리 타격 가능여부(전압 및 각도 선정)
5. 차폐 전,후 차이 비교
6. 테스트한 R2전압 adc로 읽기