


---

# TC275 DC Motor PI Control

정세인, 신재환



# 목차

## 01

### 프로젝트 구성

요구사항  
요구사항 명세서

## 03

### 프로젝트 결과

구현  
단위 테스트  
통합 테스트

## 02

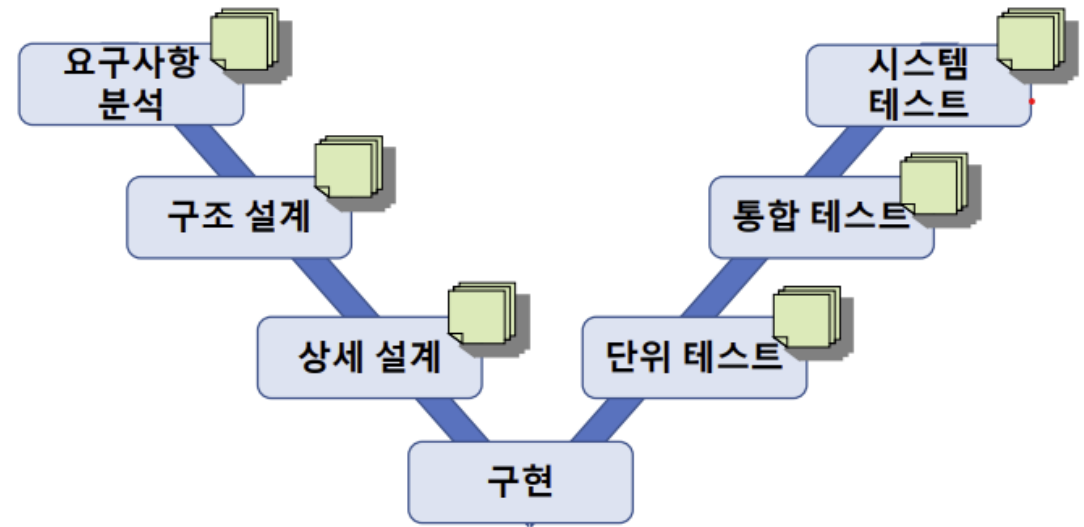
### 개발과정

구조 설계  
하드웨어 구성도  
동작 흐름도  
상세 설계

## 04

### 프로젝트 시연

시연 영상



# 프로젝트 구성

## 요구사항

### PI 속도 제어 실습

- 요구 사항
  - 속도 ref 프로파일을 아래와 같이 설정
  - 4초 후 실행
  - 15초간 상승 하강
  - 7초간  $\omega_d$  정상상태 구간 제어



- $\omega_d$  정상 상태 속도 Case 3
  - $2\pi \text{ rad/s}$  (초당 한 바퀴)
  - $4\pi \text{ rad/s}$
  - $5\pi \text{ rad/s}$

구현해 보기

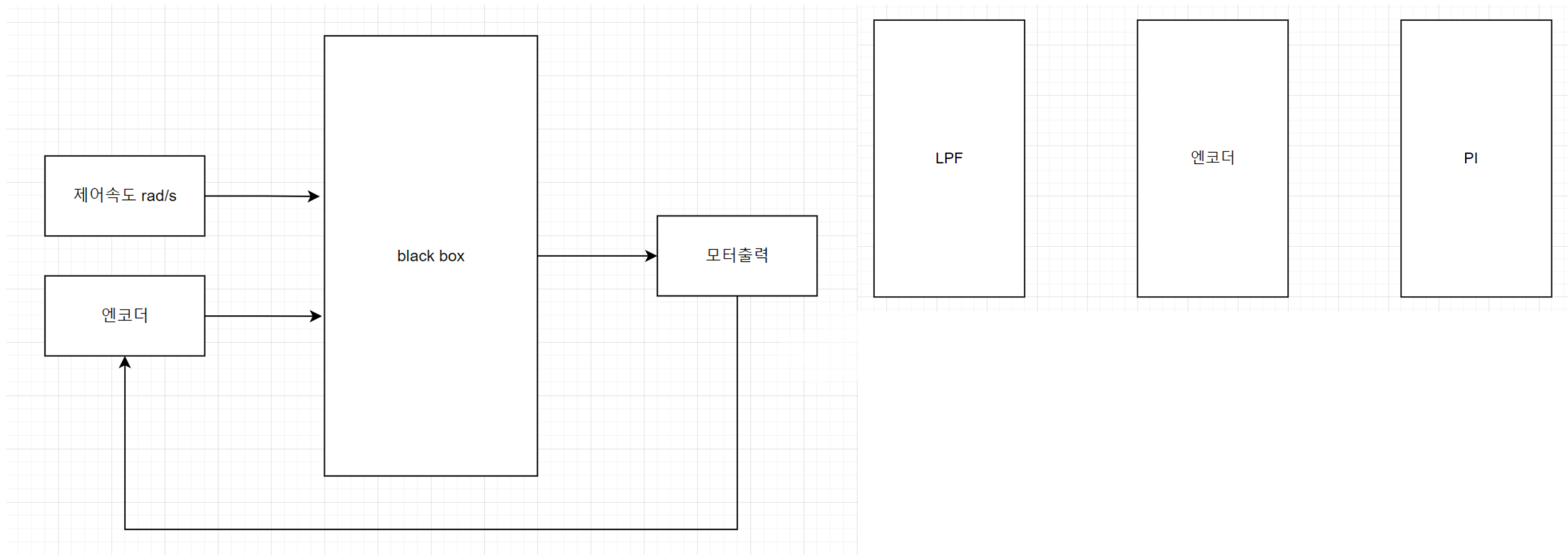
# 프로젝트 구성

## 요구사항 명세서

Req. ID	Summary	중요도	요청자/일 자	승인/비고
Reqspect-001	0초부터 4초까지 $0\pi$ rad/s로 모터속도 설정한다.	중	신재환/2024. 2. 22	
Reqspect-002	4초부터 19초구간동안 $0\pi$ rad/s~특정 $\pi$ rad/s까지 linear하게 상승한다.	상	신재환/2024. 2. 22	
Reqspect-003	19초부터 26초구간동안 특정 $\pi$ rad/s를 정상 상태 구간으로 유지한다.	상	정세인/2024. 2. 22	
Reqspect-004	26초부터 41초구간동안 특정 $\pi$ rad/s~ $0\pi$ rad/s까지 linear하게 감소한다.	상	신재환/2024. 2. 22	
Reqspect-005	특정속도는 $2\pi, 4\pi, 5\pi$ rad/s로 설정한다.	중	정세인/2024. 2. 22	
Reqspect-006	Pi제어를 통해 모터 속도를 제어한다.	상	신재환/2024. 2. 22	
Reqspect-007	LPF를 이용해 노이즈를 제거하여 정밀 제어한다.	상	정세인/2024. 2. 22	

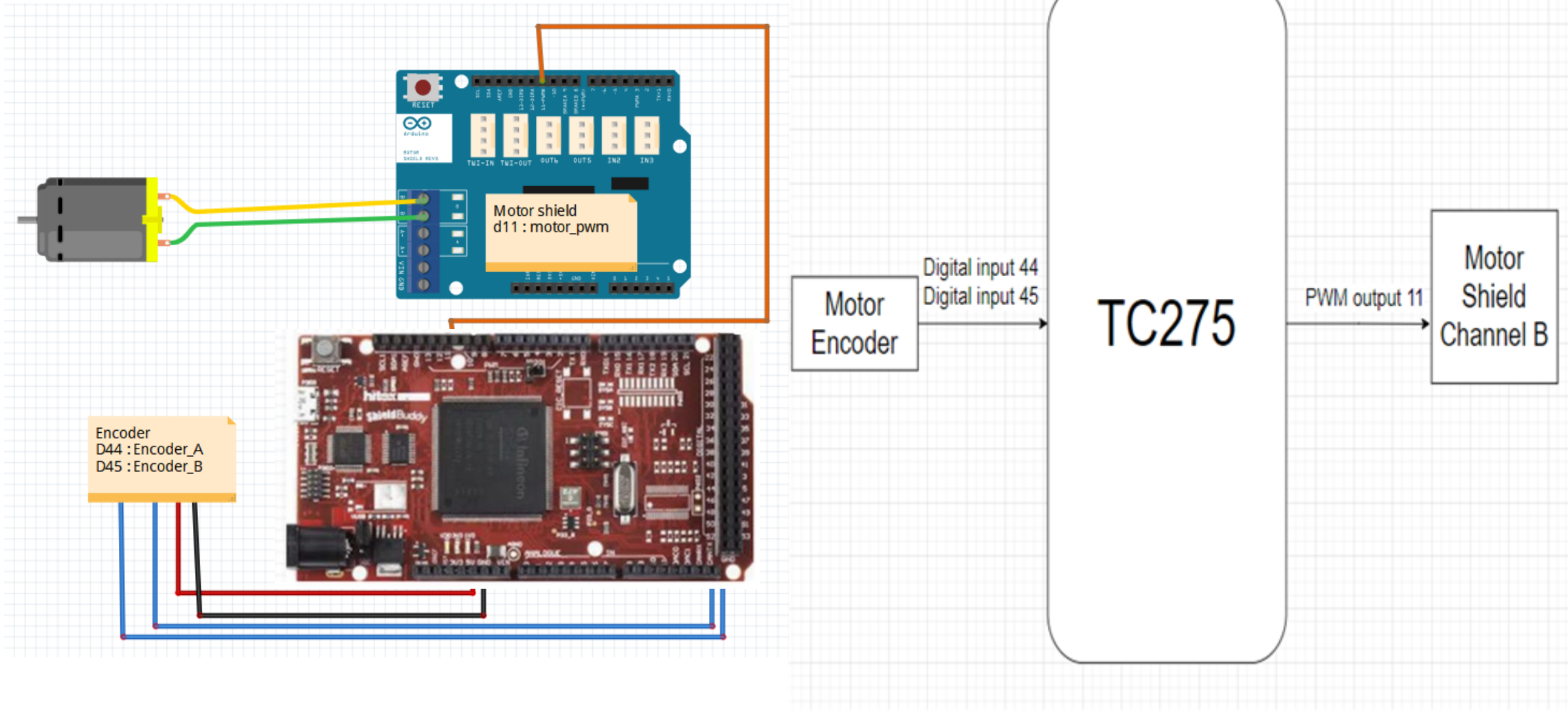
# 개발과정

## 구조 설계



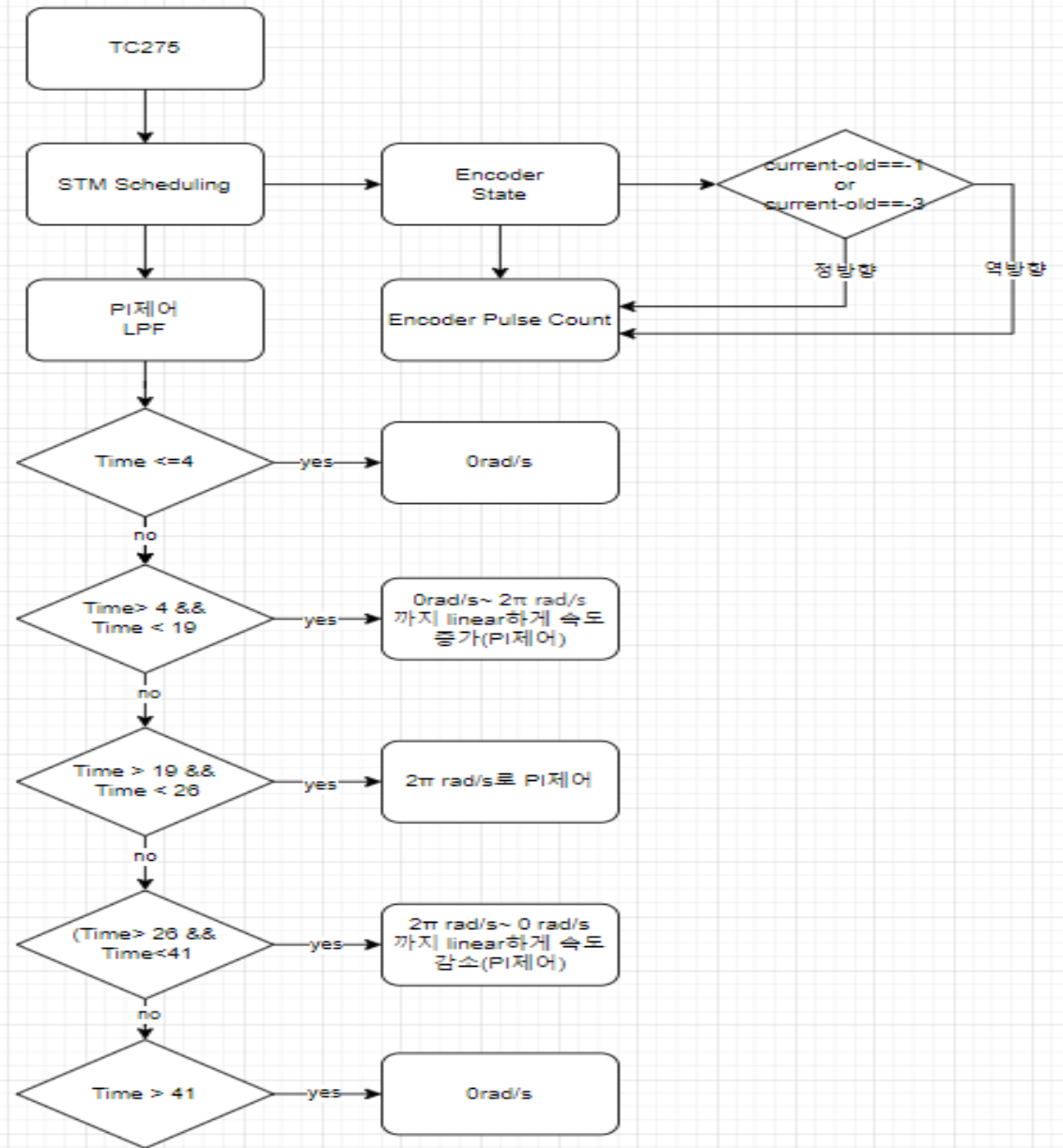
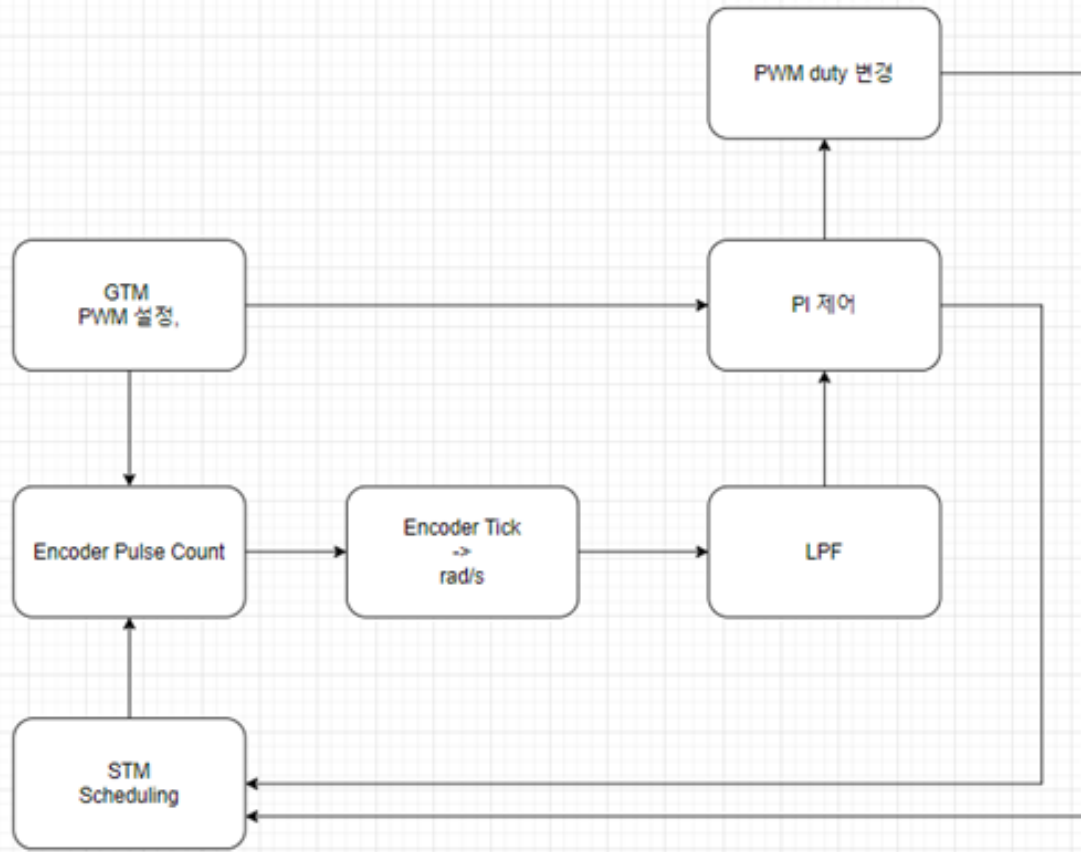
# 프로젝트 구성

## 하드웨어 구성도



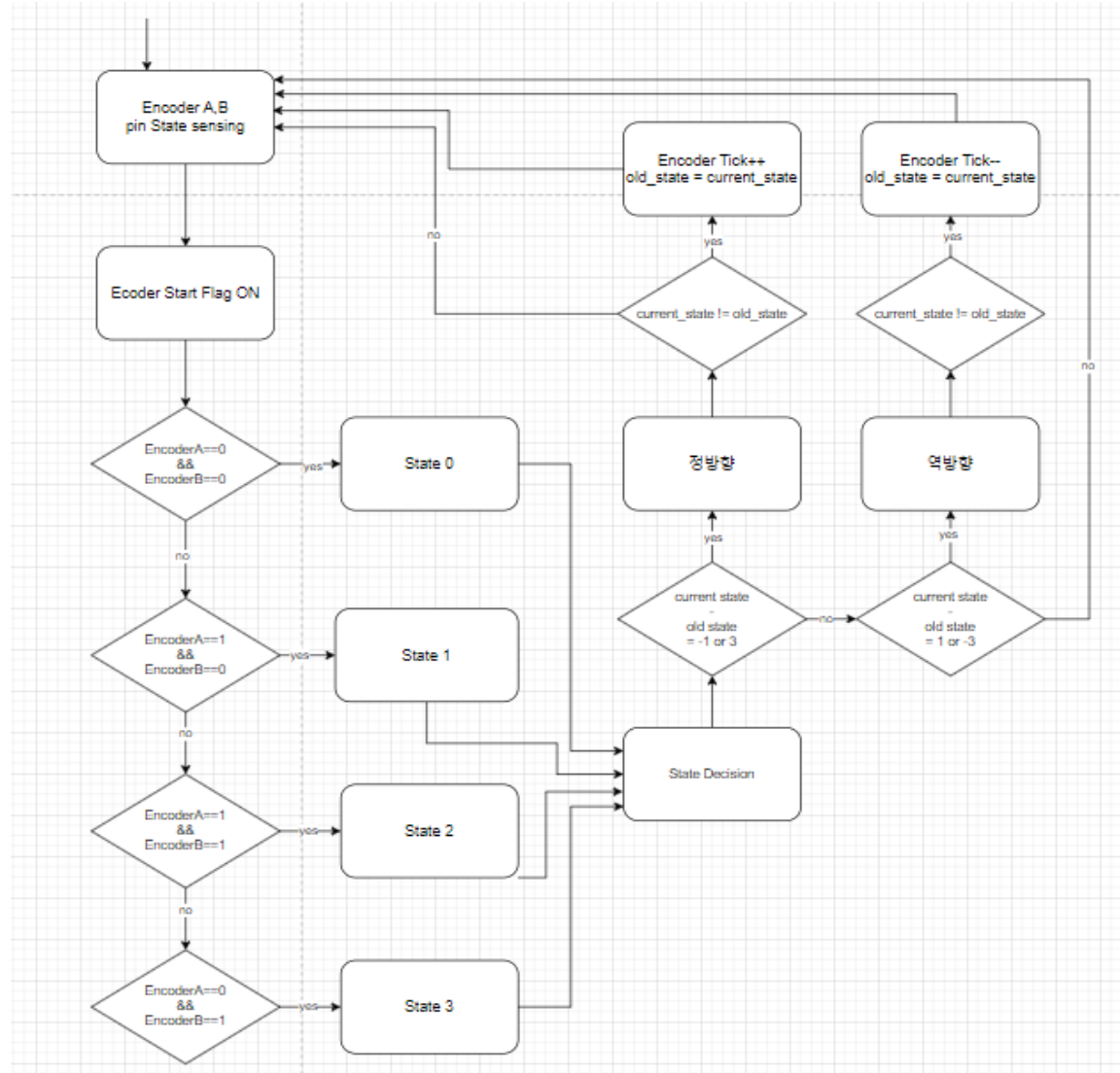
# 프로젝트 구성

## 동작 흐름도



# 개발과정

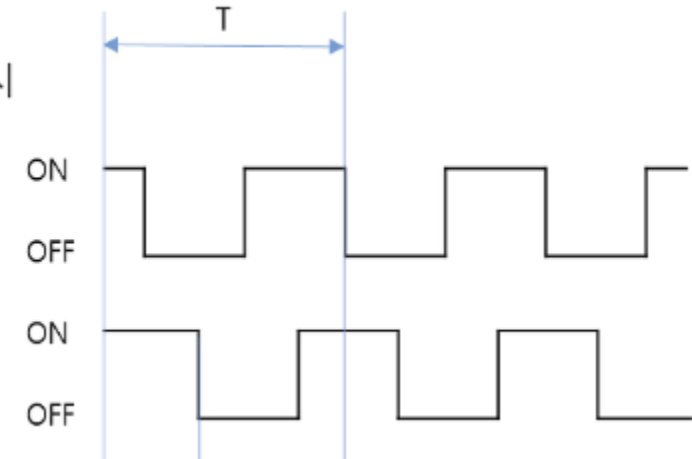
## 상세 설계(엔코더)



### ● 정회전 시

A상

B상





# 프로젝트 결과

## 구현

```
98 static void Apnotask(void) {
99     EncoderA = IfxPort_getPinState(ENCA);
100     EncoderB = IfxPort_getPinState(ENCB);
101     if((EncoderA == 0) && (EncoderB == 0)) {
102         EncoderState = 0;
103     }
104     if((EncoderA == 1) && (EncoderB == 0)) {
105         EncoderState = 1;
106     }
107     if((EncoderA == 1) && (EncoderB == 1)) {
108         EncoderState = 2;
109     }
110     if((EncoderA == 0) && (EncoderB == 1)) {
111         EncoderState = 3;
112     }
113
114     if(EncoderFlag == 0) {
115         OldEncoderState = EncoderState;
116         EncoderFlag = 1;
117     }
118
119     if((EncoderState-OldEncoderState == 1 || EncoderState-OldEncoderState == -3 )){
120         dir =0; //정방향 가정
121     }
122     if((EncoderState-OldEncoderState == -1 || EncoderState-OldEncoderState == 3)){
123         dir =1; // 역방향 가정
124     }
125
126     if(OldEncoderState != EncoderState){
127         switch(dir){
128             case 0:
129                 EncoderTicks++;
130                 break;
131             case 1:
132                 EncoderTicks--;
133                 break;
134         }
135     }
```

```
136     if(T > 19 && T < 26){
137         if(OldEncoderState != EncoderState){
138             switch(dir){
139                 case 0:
140                     SteadyStateTicks++;
141                     break;
142                 case 1:
143                     SteadyStateTicks--;
144                     break;
145             }
146         }
147     }
148     OldEncoderState = EncoderState;
149     theta = (EncoderTicks *2 * 3.141592)/(4*12);
150 }
```

# 프로젝트 결과

## 구현

```
152 static void AppTaskIms(void){
153     msec++;
154     T = 0.001*msec;
155
156     w = (float32)(theta - theta_old)/Ts;
157     w = LPF(w_old, w, 160, Ts);
158     w_old = w;
159     //W_RPM = (uint32)(60*w/(2*3.141592));
160     theta_old = theta;
161
162     if(T<=4)
163     {
164         w_ref = 0;
165     }
166     else if(T > 4 && T < 19)
167     {
168         w_ref = 0.06667*Wd*(2*3.141592)*(T-4);
169     }
170     else if(T >= 19 && T <= 26)
171     {
172         w_ref = Wd*(3.141592*2);
173     }
174     else if(T > 26 && T<41)
175     {
176         w_ref = Wd*(3.141592*2)-0.06667*Wd*(2*3.141592)*(T-26);
177     }
178     else if(T > 41)
179     {
180         w_ref=0;
181     }
182     if(T <= 4)
183     {
184         Vin = 0;
185         EncoderTicks = 0;
186     }
```

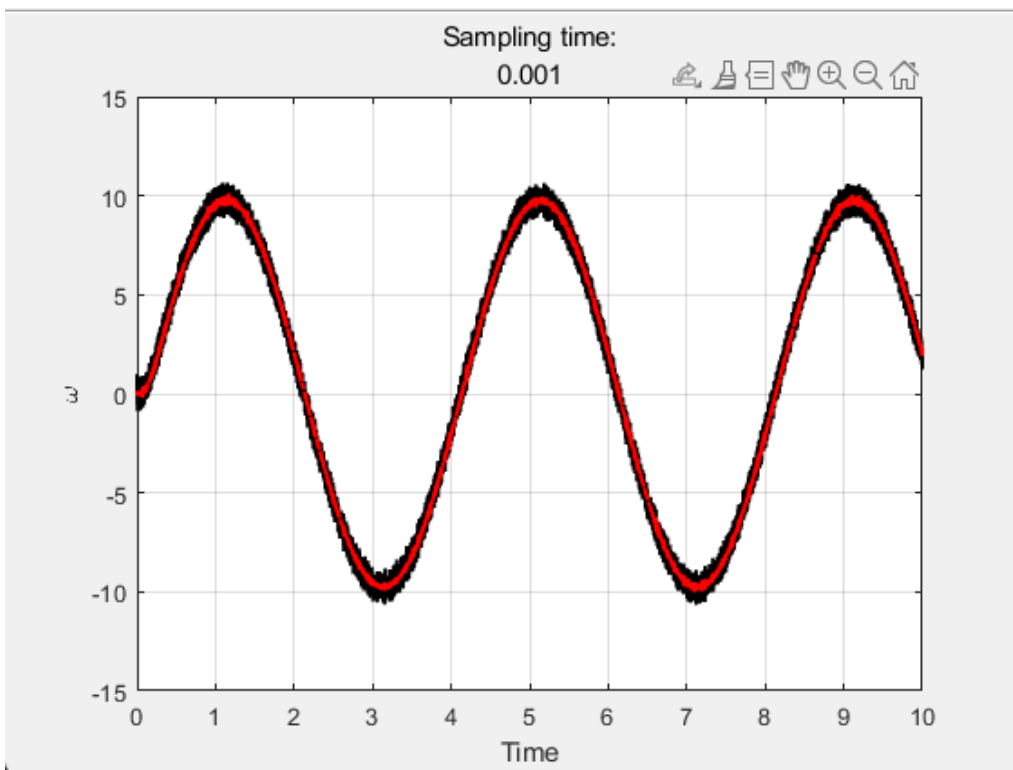
```
187     else if(T > 4 && T < 41)
188     {
189         error_w = w_ref - w;
190         error_w_int = error_w_int_old + (error_w)*Ts;
191         error_w_int_old = error_w_int;
192
193         if(error_w_int > 10)
194         {
195             error_w_int = 10;
196         }
197
198         Vin = (kp*error_w+ki*error_w_int);
199
200         if(Vin > 11)
201         {
202             Vin = 11;
203         }
204         else if(Vin < 0)
205         {
206             Vin = 0;
207         }
208     }
209     else if(T > 41)
210     {
211         Vin = 0;
212     }
213     DUTY[0] = Vin/12;
214     g_GtmTomPwmHl.tOn[0] = DUTY[0];
215     GtmTomPwmHl_run();
216 }
```

# 프로젝트 결과

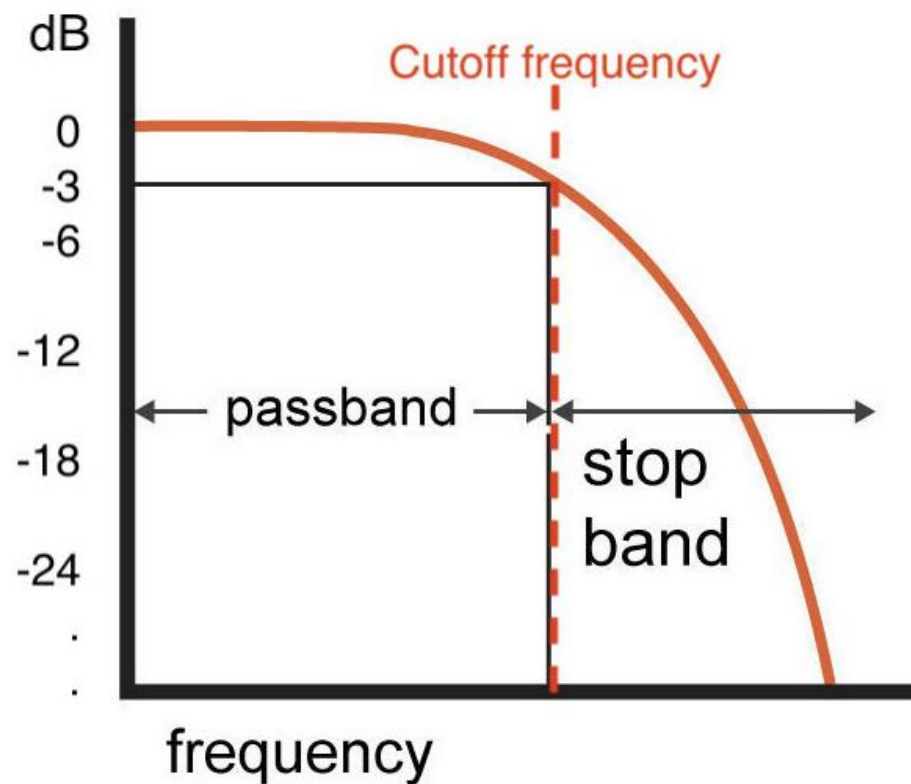
## 구현

```
93 float32 LPF(float32 Y_fil_d, float32 U, uint32 cf, float32 Ts) {  
94     float32 Y_fil;  
95     Y_fil = (1-Ts*cf)*Y_fil_d+Ts*cf*U;  
96     return Y_fil;  
97 }
```

## LPF



## Low-pass



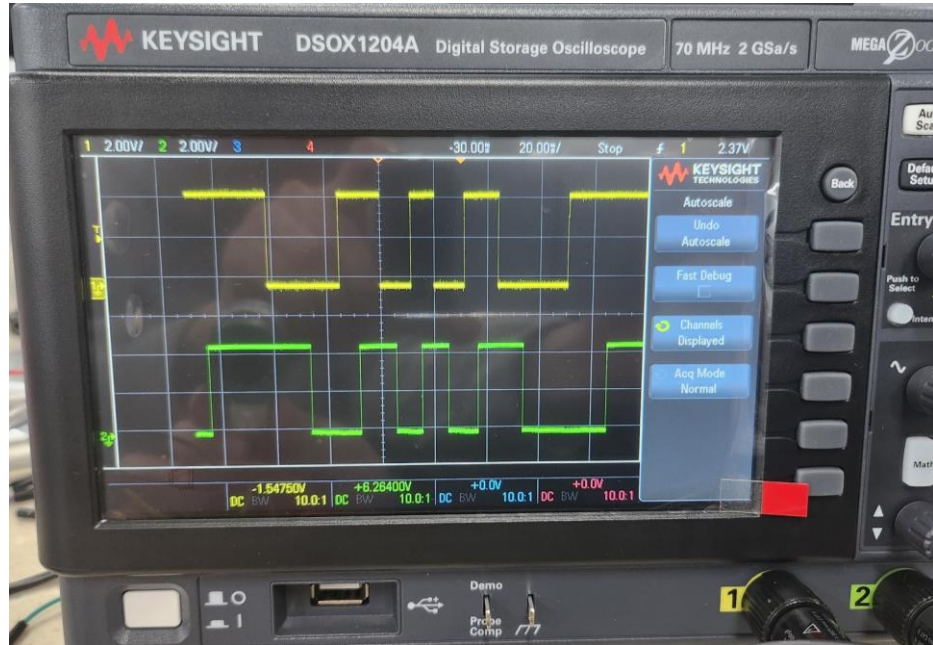
# 프로젝트 결과

## 단위 테스트

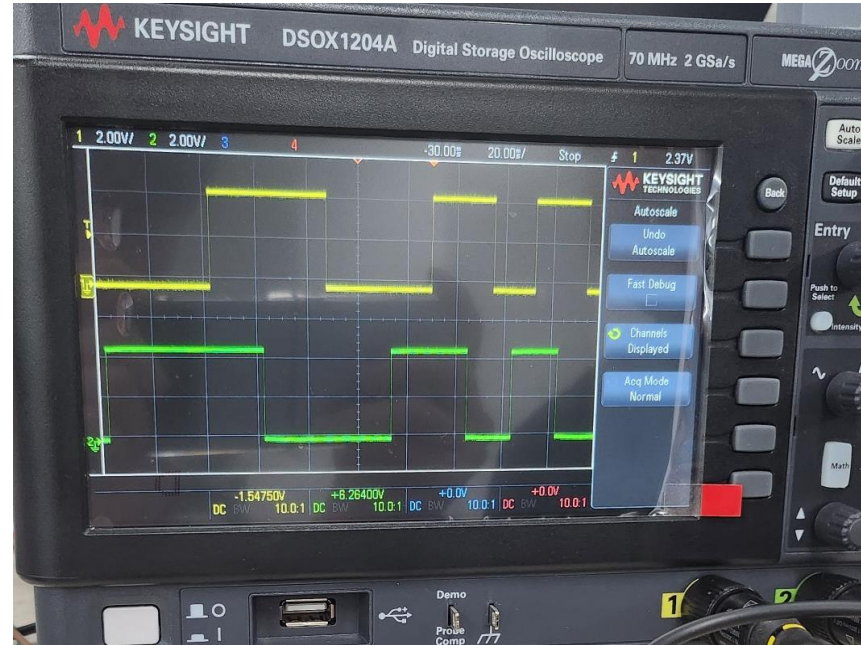
Req. ID	Summary	Tester	Date	검증 방법	결과
U-Test-001	EncoderTicks=0으로 초기화한 후Motor을 정방향으로 한바퀴 돌린 후 EncoderTicks=48이 되는지 확인	신재환	2024. 2. 22	모터를 손으로 정방향 한바퀴 돌림.	O
U-Test-002	EncoderTicks=0으로 초기화한 후Motor을 역방향으로 한바퀴 돌린 후 EncoderTicks=-48이 되는지 확인	신재환	2024.2.22	모터를 손으로 역방향 한바퀴 돌림	O
U-Test-003	EncoderTicks=0으로 초기화한 후Motor을 정방향으로 한바퀴 돌린 후, 다시 역방향으로 한바퀴 돌린 후 EncoderTicks=0가 되는지 확인	정세인	2024. 2. 22	모터를 손으로 정방향 한바퀴, 역방향 한바퀴 돌림	O
U-Test-004	Duty 설정 후 Motor 회전 확인	신재환	2024. 2. 22	Duty = 0.3	O
U-Test-005	0s~4s까지 DC Motor의 Angular_Velocity를 0rad/s로 설정 후 모터동작확인	정세인	2024. 2. 22	Wref=0, Duty =0	O
U-Test-006	Matalb을 이용해 노이즈가 낀 신호를 만들고 설계한 LPF를 통과시켜 cutoff frequency 도출	신재환	2024. 2. 22	Ts = 0.001, Noise = 0.001*rand LPF function	O

# 프로젝트 결과

## 단위 테스트



엔코더 정방향



엔코더 역방향

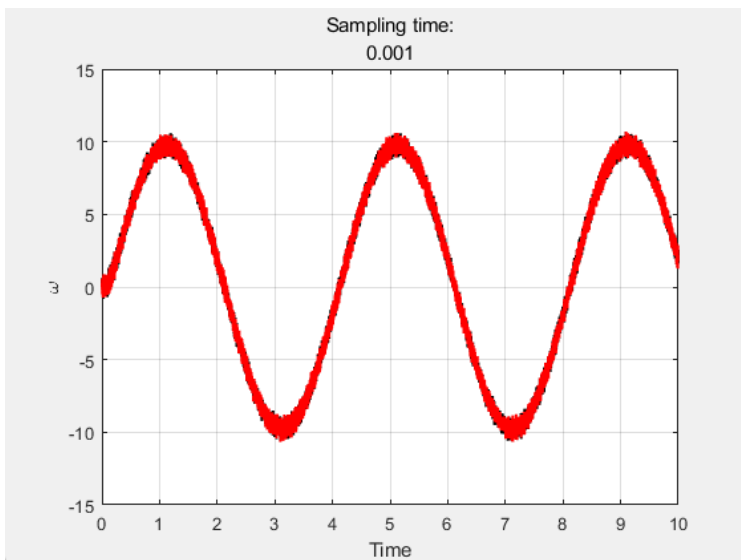
# 프로젝트 결과

## 단위 테스트

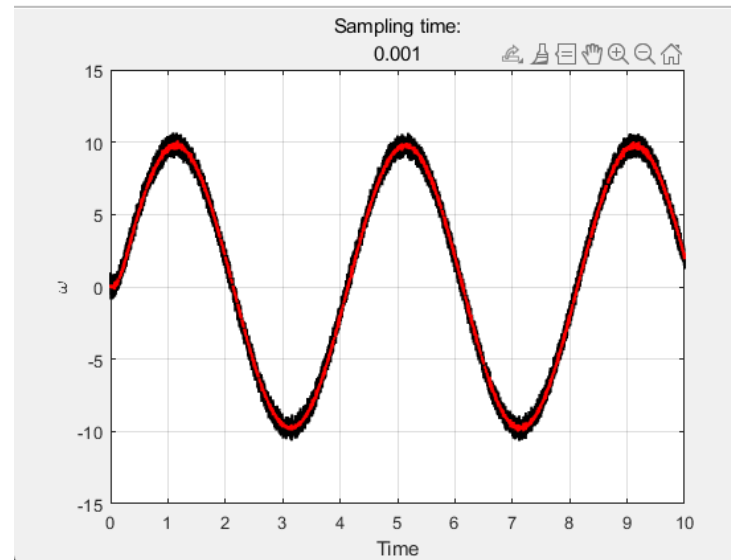
Name	Type	Value
'Appsheduling.c'::T	float	49.724
'Appsheduling.c'::msec	unsigned long	49724
'Appsheduling.c'::EncoderTicks	signed long	2594
'Appsheduling.c'::Vin	float	0
'Appsheduling.c'::SteadyStateTi	signed long	828

Name	Type	Value
'Appsheduling.c'::T	float	46.195
'Appsheduling.c'::msec	unsigned long	46195
'Appsheduling.c'::EncoderTicks	signed long	2591
'Appsheduling.c'::Vin	float	0
'Appsheduling.c'::SteadyStateTi	signed long	840

Cutoff-frequency 1000



Cutoff-frequency 160



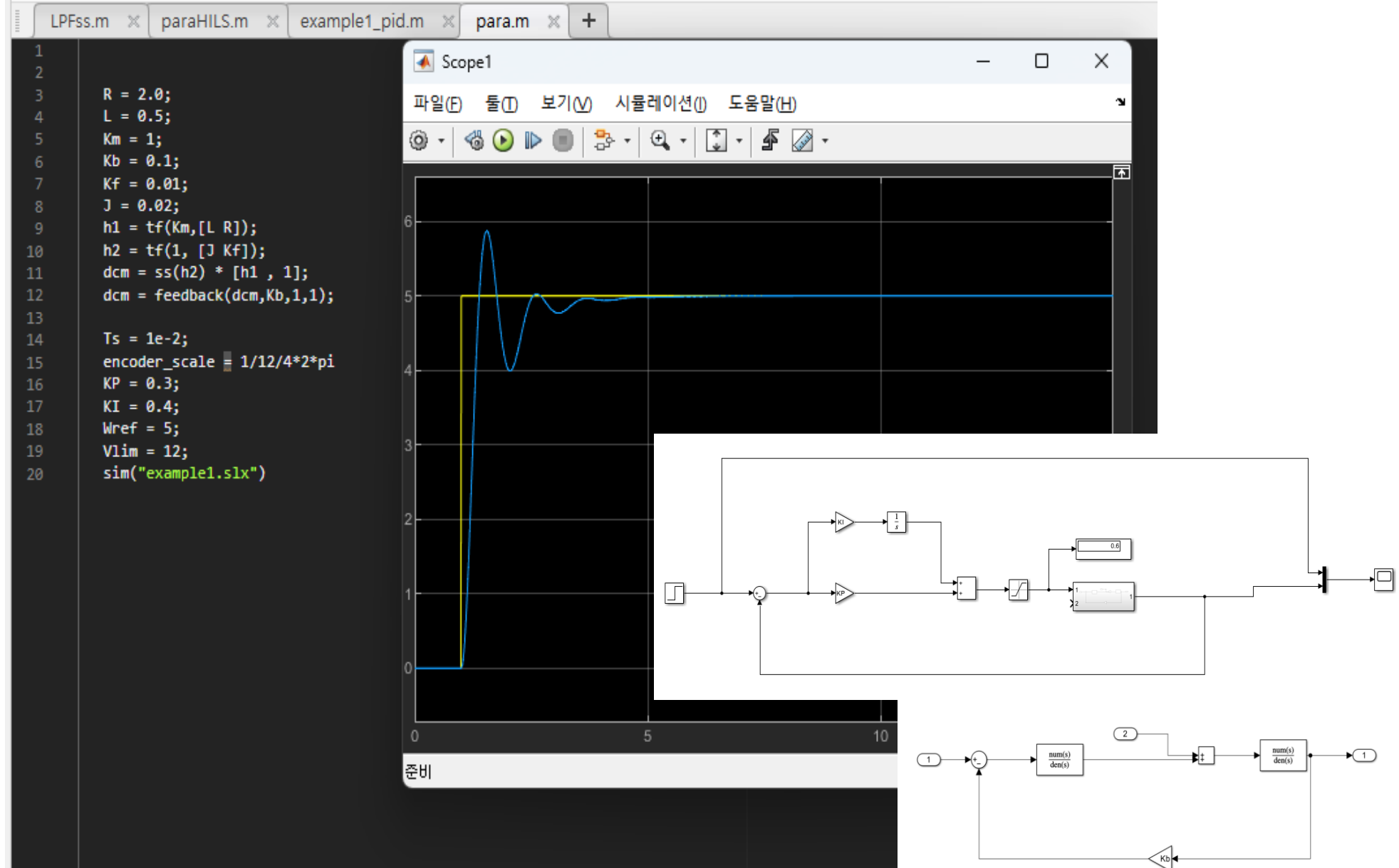
# 프로젝트 결과

## 단위 테스트

Req. ID	Summary	Tester	Date	검증 방법	결과
U-Test-007	Matlab, Simulink를 통해 DC Motor모델링후 KI, KP를 가변하면서 Scope로 PI제어된 그래프 확인. 최적 튜닝값 도출.	신재환	2024. 2. 22	KP= 0.01~50 KI= 0.01~50	O
U-Test-008	SILS서 얻은 cutoff frequency, KI,KP를 적용해 초기속도 0rad/s로 시작해0s~10s까지 $2\pi$ rad/s로 PI제어. STM 10ms Scheduling 이용.	정세인	2024. 2. 22	kP = 0.01~50 KI = 0.01~50 Cutoff frequency = variable, Ts = 0.01	X
U-Test-009	STM 1ms Scheduling를 이용해 Test-008과 같이 PI제어	신재환	2024. 2. 22	Ts=0.001	X
U-Test-010	초기속도 0rad/s로 시작해 0s~8s까지 linear 하게 $2\pi$ rad/s까지 속도 증가	정세인	2024. 2. 22	Ts=0.001 kP = variable KI = variable	O
U-Test-011	LPF 고정, KP, KI값을 변경 후 초기속도 0rad/s로 시작해 0s~8s까지 linear 하게 $2\pi$ rad/s까지 속도 증가, 변화 관측	신재환	2024. 2. 22	Ts=0.001, kP = 0.01~50 KI = 0.01~50, CFL= 500	O
U-Test-012	LPF 변경, KP, KI값을 고정 후 초기속도 0rad/s로 시작해 0s~8s까지 linear 하게 $2\pi$ rad/s까지 속도 증가, 변화 관측	정세인	2024. 2. 22	Ts=0.001 kP = 0.3, KI = 0.4 CFL= 50~1000	O

# 프로젝트 결과

## 단위 테스트



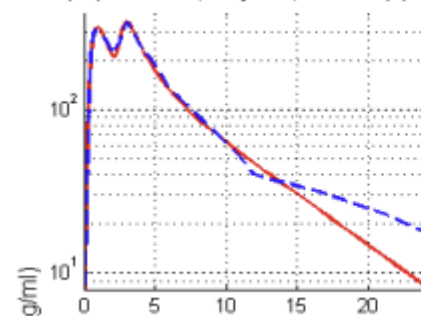


# 프로젝트 결과

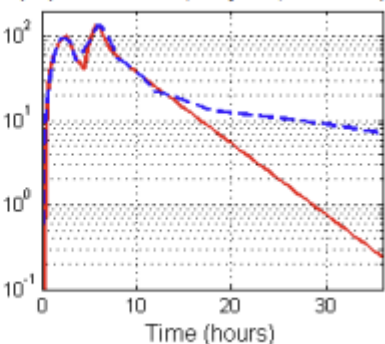
## 단위 테스트

```
187 else if(T > 4 && T < 41)
188 {
189     error_w = w_ref - w;
190     error_w_int = error_w_int_old + (error_w)*Ts;
191     error_w_int_old = error_w_int;
192
193     if(error_w_int > 10)
194     {
195         error_w_int = 10;
196     }
197
198     Vin = (kp*error_w+ki*error_w_int);
199
200     if(Vin > 11)
201     {
202         Vin = 11;
203     }
204     else if(Vin < 0)
205     {
206         Vin = 0;
207     }
208 }
209 else if(T > 41)
210 {
211     Vin = 0;
212 }
213 DUTY[0] = Vin/12;
214 g_GtmTomPwmH1.tOn[0] = DUTY[0];
215 GtmTomPwmH1_run();
216 }
```

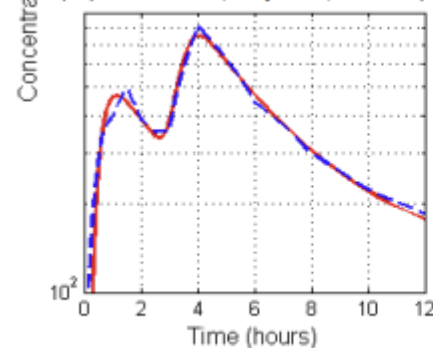
(4A) Data Set 1, Subject G, Model PI(2)



(4B) Data Set 2, Subject C, Model VA(2)



(4C) Data Set 3, Subject A2, Model PI(2)



— Response of fitted model  
- - - Linear interpolation  
between measurement points

출처: <https://www.walshmedicalmedia.com/open-access/modelling-the-double-peak-phenomenon-jbb.1000068.pdf>

# 프로젝트 결과

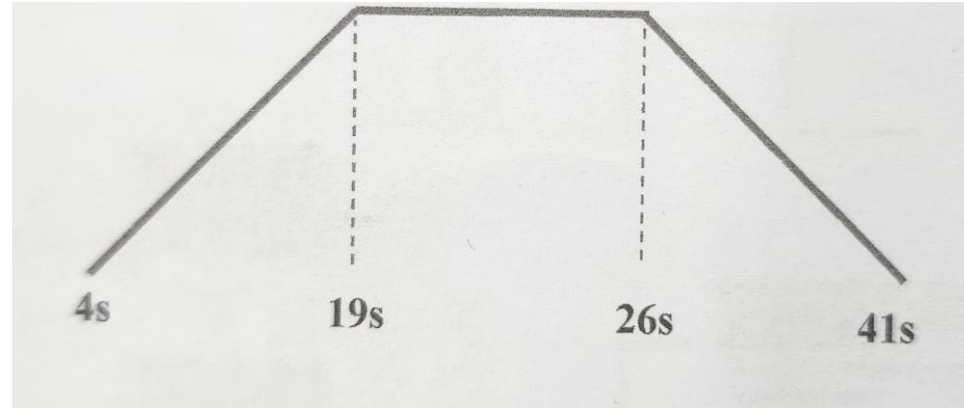
## 통합 테스트

Req. ID	Summary	Tester	Date	검증 방법	결과
I-Test-001	초기상태 0 rad/s, 0s~4s까지 0rad/s로 모터설정, duty 반영 후 동작	신재환	2024. 2. 22	EncoderTicks=0인지 확인 Wd = $2\pi$ rad/s	O
I-Test-002	초기상태 0 rad/s, 0s~4s까지 0rad/s, 4s ~ 19s 동안 0 rad/s 부터 $2\pi$ rad/s 까지 linear 하게 속도 증가	정세인	2024. 2. 22	4s~9s ,9s~14s,14sf~19s 세 구간 별로 작동하는 CNT 설정 후 CNT가 linear하게 증가 하 는지 비교 Wd = $2\pi$ rad/s	O
I-Test-003	초기상태 0 rad/s, 0s~4s까지 0rad/s, 4s ~ 19s 동안 0 rad/s 부터 $2\pi$ rad/s 까지 linear 하게 속도 증가, 19s~26s까지 $2\pi$ rad/s 동일 속 도로 회전하게 모터 동작	신재환	2024. 2. 22	19s~26s에만 카운트하는 SteadyStateTicks 설정 후 $48*7*1 = 336$ 이 나오는지 확 인 Wd = $2\pi$ rad/s	O
I-Test-004	초기상태 0 rad/s, 0s~4s까지 0rad/s, 4s ~ 19s 동안 0 rad/s 부터 $2\pi$ rad/s 까지 linear 하게 속도 증가, 19s~26s까지 $2\pi$ rad/s 동일 속 도, 26s~41s까지 linear하게 0rad/s로 속 도 감소	정세인	2024. 2. 22	26s~31s ,31s~36s,36s~41s 세 구간 별로 작동하는 CNT 설정 후 CNT가 linear하게 감 소 하는지 비교 Wd = $2\pi$ rad/s	O
I-Test-005	목표 각속도를 변경하면서 I-Test-001 ~I- Test-004 반복	신재환	2024. 2. 22	Wd = $4\pi$ rad/s Wd = $5\pi$ rad/s	O

# 프로젝트 결과

## 통합 테스트

```
137     if(T > 19 && T < 26){
138         if(OldEncoderState != EncoderState){
139             switch(dir){
140                 case 0:
141                     SteadyStateTicks++;
142                     break;
143                 case 1:
144                     SteadyStateTicks--;
145                     break;
146             }
147         }
148     }
```

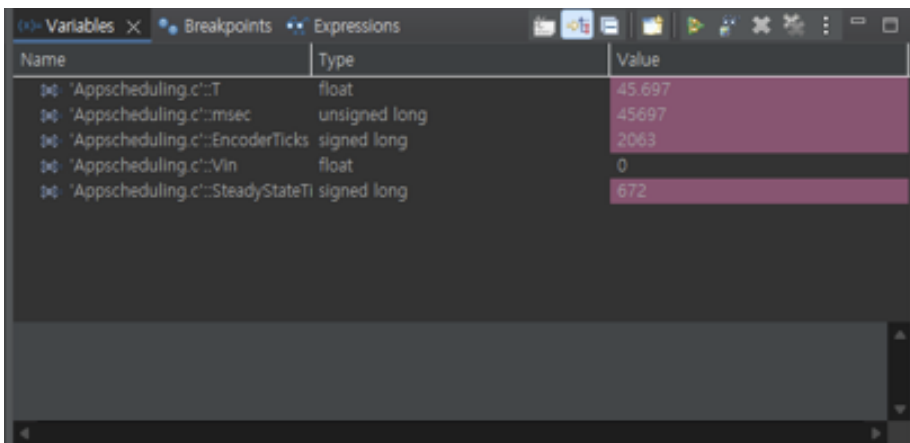


Variables			Variables			Variables		
Name	Type	Value	Name	Type	Value	Name	Type	Value
'Appscheduling.c'::T	float	43.866	'Appscheduling.c'::T	float	44.848	'Appscheduling.c'::T	float	44.19
'Appscheduling.c'::msec	unsigned long	43866	'Appscheduling.c'::msec	unsigned long	44848	'Appscheduling.c'::msec	unsigned long	44190
'Appscheduling.c'::EncoderTicks	signed long	1008	'Appscheduling.c'::EncoderTicks	signed long	1008	'Appscheduling.c'::EncoderTicks	signed long	1008
'Appscheduling.c'::Vin	float	0	'Appscheduling.c'::Vin	float	0	'Appscheduling.c'::Vin	float	0
'Appscheduling.c'::SteadyStateTi	signed long	336	'Appscheduling.c'::SteadyStateTi	signed long	337	'Appscheduling.c'::SteadyStateTi	signed long	336

$$2\pi \text{ rad/s} = 48 \times 7 = 336$$

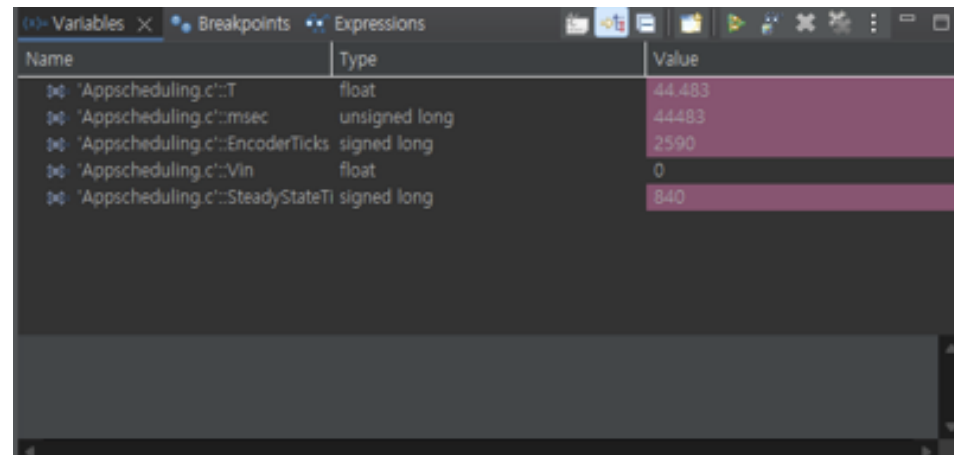
# 프로젝트 결과

## 통합 테스트



Name	Type	Value
'Apscheduling.c':T	float	45.697
'Apscheduling.c':msec	unsigned long	45697
'Apscheduling.c':EncoderTicks	signed long	2063
'Apscheduling.c':Vin	float	0
'Apscheduling.c':SteadyStateTi	signed long	672

$$4\pi \text{ rad/s} = 48 \cdot 7 \cdot 2 = 672$$



Name	Type	Value
'Apscheduling.c':T	float	44.483
'Apscheduling.c':msec	unsigned long	44483
'Apscheduling.c':EncoderTicks	signed long	2590
'Apscheduling.c':Vin	float	0
'Apscheduling.c':SteadyStateTi	signed long	840

$$5\pi \text{ rad/s} = 48 \cdot 7 \cdot 2.5 = 840$$



# 프로젝트 시연

시연 영상

