

4조
IOT PROJECT

“ REALIZATION OF ” STATION

➤ 최종 프로젝트 대비 스테이션 제작

<팀명> : 놀이공원 가고 싶어요

팀장은 # 최수혁 # 이종호 # 이가은 # 조예봄

4조 CONTENTS

➤ PHASE 01

"전체적인 구조"

: "최종 프로젝트"에 대한 간략한 설명

- 이슈 & 택배현황
- 최종 프로젝트란?
- 대표적 사례
- 4가지 순서

➤ PHASE 02

"스테이션에 대해"

: 스테이션에 대한 설명

- 스테이션 내부
- 관람차식 택배보관소
- 응용한 구조
- 하드웨어 제작

➤ PHASE 03

"구현하기"

: 스테이션 구현에 대해

- 사용한 기술 및 부품
- 부품 별 추가설명
- 순서도
- 순서도 추가설명
- 코드
- 마무리

4조 PHASE 01

"전체적인 구조"

스테이션에 대해 설명하기 앞서 “최종 프로젝트”에 대한 간략한 설명을 합니다.

: 이슈 & 택배현황, 최종 프로젝트란?, 대표적 사례, 4가지 순서

Realization of ‘STATION’

01 전체구조

차도없이 조성되는 요즘의 아파트 & 택배종사자와의 마찰



해결하고자하는 문제

‘아파트 갑질’은 모두가 한번씩 들어본 적이 있을 것이다. 요즘, 아파트는 지상의 차도를 없애며 공원으로 조성하고 있으며, 낮은 차고를 가진 차량만이 지하를 이용할 수 있도록 하고 있다.

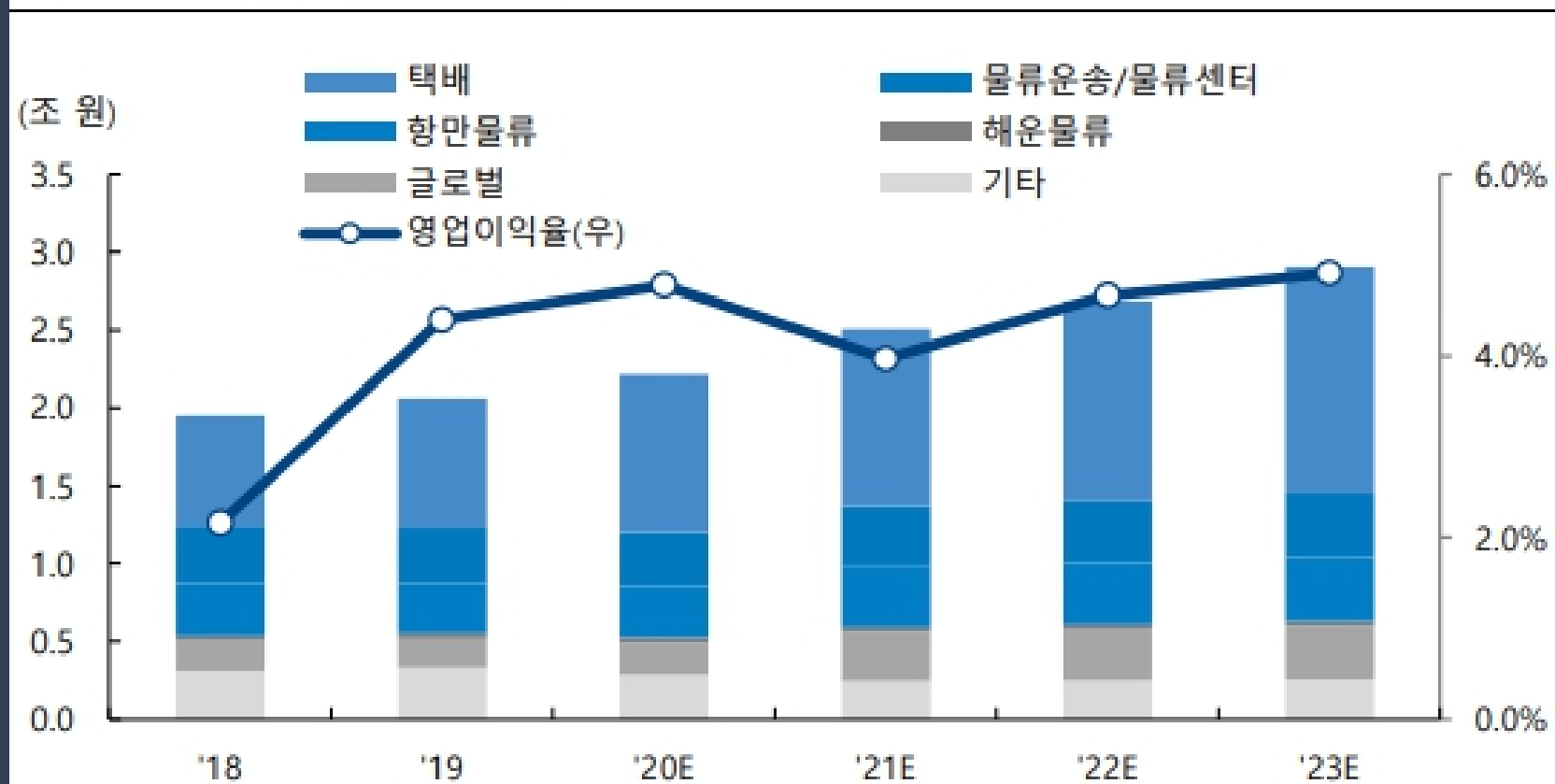
이는 배달 업종 종사자들과의 마찰을 만들어냈고, 이를 대신하는 로봇을 통해 문제를 해결하고자 한다.

https://newsis.com/view/?id=NISI20210414_0017349054

01 전체구조

현황) 1인당 택배 이용건수 : 연간 70회

그림3. 한진 부문별 매출 및 영업이익률 추이



자료: 한진, 하이투자증권

택배의 문제점

날이 갈수록 택배의 이용량은 연간 10%씩 증가하고 있다. 반면 택배환경은 여전히 답보상태이다. 실제로 배송을 할때, 건물 내에서 대기시간이 상당하다고 한다.

추가 정보

1. 전국 총 주택 가운데 아파트의 비율은 62%
2. 아파트 한 동 당 하루 평균 택배의 양: 8~18개



01 전체구조

택배배송 시스템

= "최종 프로젝트"

유통업계에 라스트마일 배송의 중요성이 확대되는 가운데 배송 기술 역시 혁신을 거듭나고 있다. 전체 운송에서 라스트마일 배송이 차지하는 비중은 절반 이상이며, 인력 부족 심화 속 유통업체들은 비용절감과 수익 증대를 위해 첨단기술을 기반으로 하는 배송 시스템을 구축하는 중이다. 그 중에서도 엘리베이터를 통해 층간이동을 하며 배송을 하는 국내기업에는 "WATT"가 있다.

이미 시중에서 사용가능한 로봇을 제작하는 "WATT"의 '제임스'는 층간을 이동하며 서비스를 제공한다. 사진 속의 로봇이 "WATT"의 '제임스'이고, 직접 엘리베이터를 통해 택배를 배송할 수 있다.



01 전체구조

대표적인 사례

국내기업 Watt - 'James'

유통업계에 라스트마일 배송의 중요성이 확대되는 가운데 배송 기술 역시 혁신을 거듭나고 있다. 전체 운송에서 라스트마일 배송이 차지하는 비중은 절반 이상이며, 인력 부족 심화 속 유통업체들은 비용절감과 수익 증대를 위해 첨단기술을 기반으로 하는 배송 시스템을 구축하는 중이다. 그 중에서도 엘리베이터를 통해 층간이동을 하며 배송을 하는 국내기업에는 “WATT”가 있다.

이미 시중에서 사용가능한 로봇을 제작하는 “WATT”의 ‘제임스’는 층간을 이동하며 서비스를 제공한다. 사진 속의 로봇이 “WATT”의 ‘제임스’이고, 직접 엘리베이터를 통해 택배를 배송할 수 있다.

01 전체구조

서비스 구상을 위해 필요한 4가지 단계

STATION 스테이션

: 같은 동/층에 따라 택배를 분류하고 보관합니다

➤ **1단계** 택배를 인식하고 분류하는 단계

➤ **2단계** 제품을 보관하는 단계 (스테이션)

ROBOT 택배배송로봇

: 보관된 택배를 들고 주행 및 배송합니다

➤ **3단계** 택배를 집어서 로봇내부 <-> 로봇외부로 옮기는 단계

➤ **4단계** 로봇이 택배를 들고 원하는 장소로 가는 주행 단계

Realization of 'STATION'

4조 PHASE 02

"스테이션에 대해"

스테이션의 구조에 대해 이야기합니다.

: 스테이션 내부, 관람차식 택배보관소, 응용한 구조&하드웨어 제작

Realization of 'STATION'

스테이션 내부

스테이션의 입구 ➤ 레일 ➤ 관람차식 택배보관소

(1) 송장을 인식하는 스테이션의 입구 : 입구로 들어오는 택배를 인식합니다.

IN할지 OUT할지를 결정 & IN이라면 송장을 인식 및 기억

(2) 레일 : 통과한 택배는 레일을 타고, 바로 '관람차식 택배보관소'로 이동합니다.

(3) 관람차식 택배보관소 : 각도조절 알고리즘을 통해 구동되는 보관소에 저장됩니다.



STRUCTURE

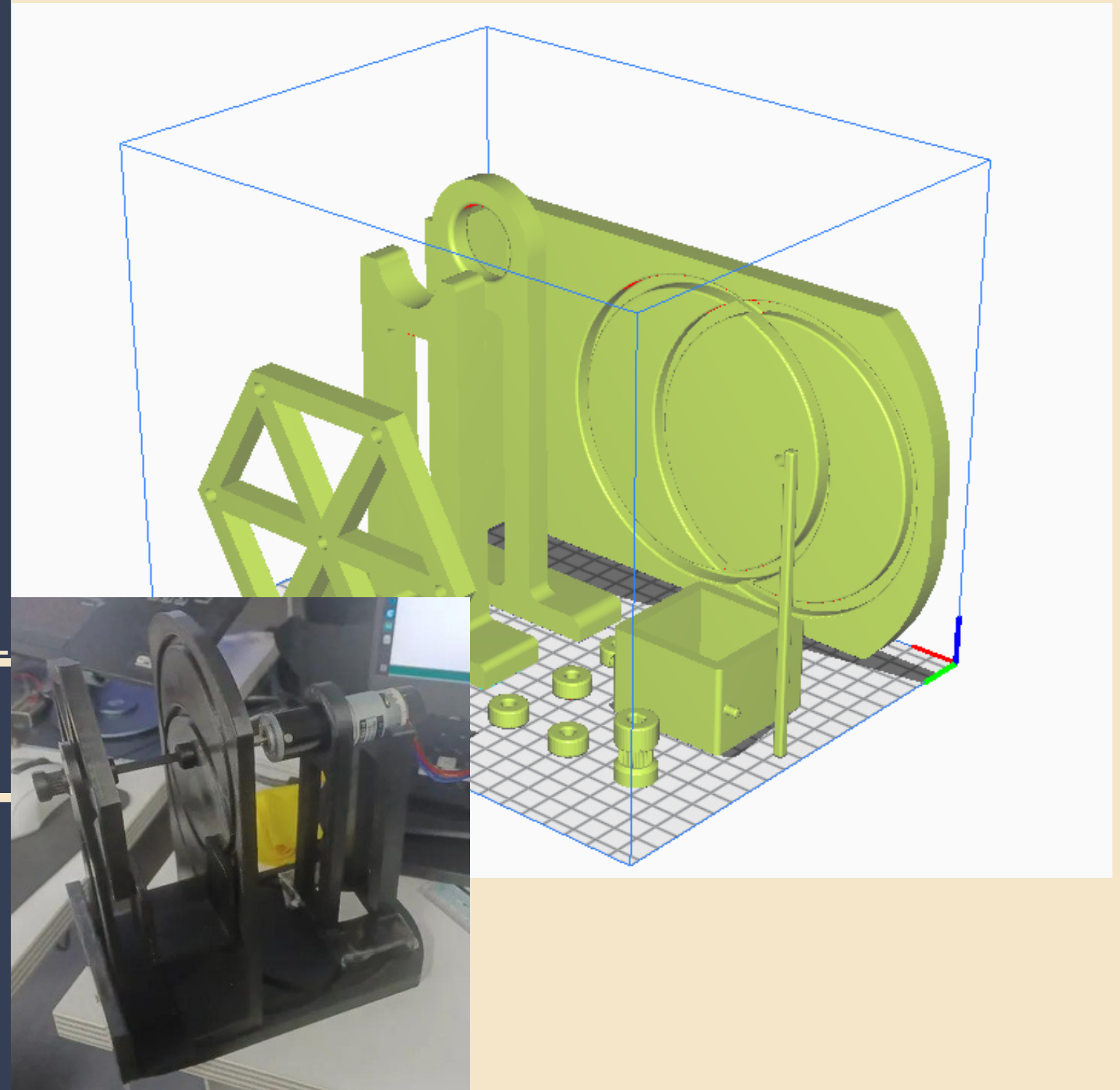
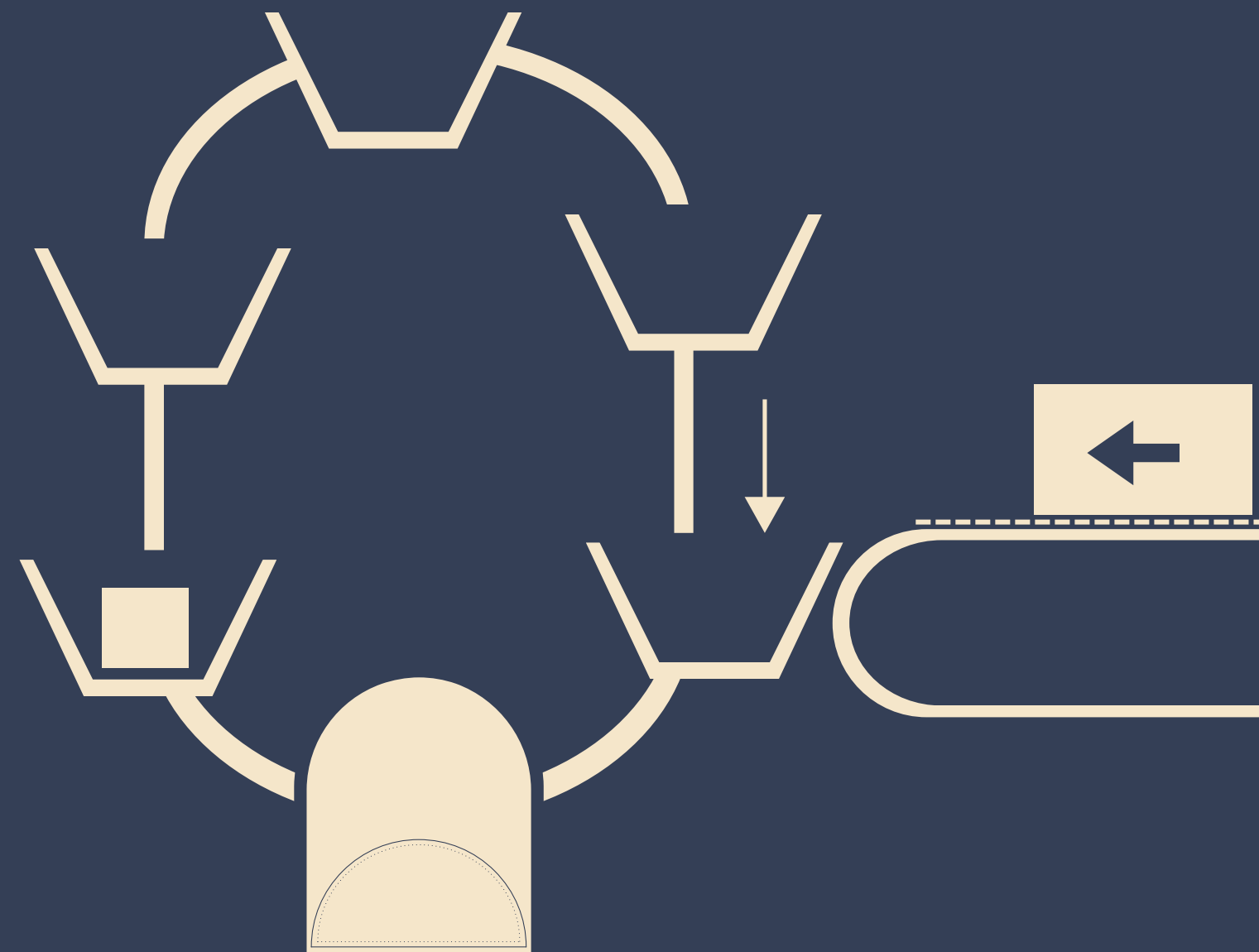
하드웨어 제작 시, 관람차의 구조를 응용했습니다

관람차식 택배보관소

➤ 활용방식과 단계

- (1) 레일을 타고 택배가 옵니다
- (2) 레일의 높이에 맞게 바구니가 위치합니다
- (3) 택배가 바구니에 담기고 난 뒤에 다른 주소의 택배가 온다면, 새 바구니가 다시 레일의 끝에 위치합니다
- (4) 같은 주소의 택배가 온다면, 그 주소의 택배가 담긴 바구니가 다시 회전하여 돌아옵니다.
(이것은 최대 3개까지 적재합니다)
(자세한 알고리즘은 뒤에서)

하드웨어 제작



4조 PHASE 03

"스테이션 구현하기"

아두이노를 통해 스테이션을 구현합니다
: 사용한 기술 및 부품, 부품 별 추가설명, 순서도, 순서도 추가설명, 코드, 마무리

Realization of 'STATION'

사용한 기술 및 부품

아두이노

LCD*

스위치*

LED

초음파 센서*

C++

엔코더 모터*

3D PRINT

리눅스

03 구현하기

사용한 기술 및 부품

||| IN DETAIL |||

➤ 엔코더 모터

관람차식 택배보관소는 회전하며 바구니의 위치를 제어합니다.
이때, 각도를 이용하기 위해 엔코더 모터를 사용합니다.

➤ 초음파 센서

초음파센서를 이용하여 바구니가 레일과 가까워지는 거리를 계산하여 바구니의 위치를 파악합니다.

➤ LCD

실제 레일과 주행로봇을 구현하지 않았기에, '레일을 통해 들어오는 택배'와 '택배를 가져가는 로봇'를 시각적으로 표현하기 위해 사용하였습니다.

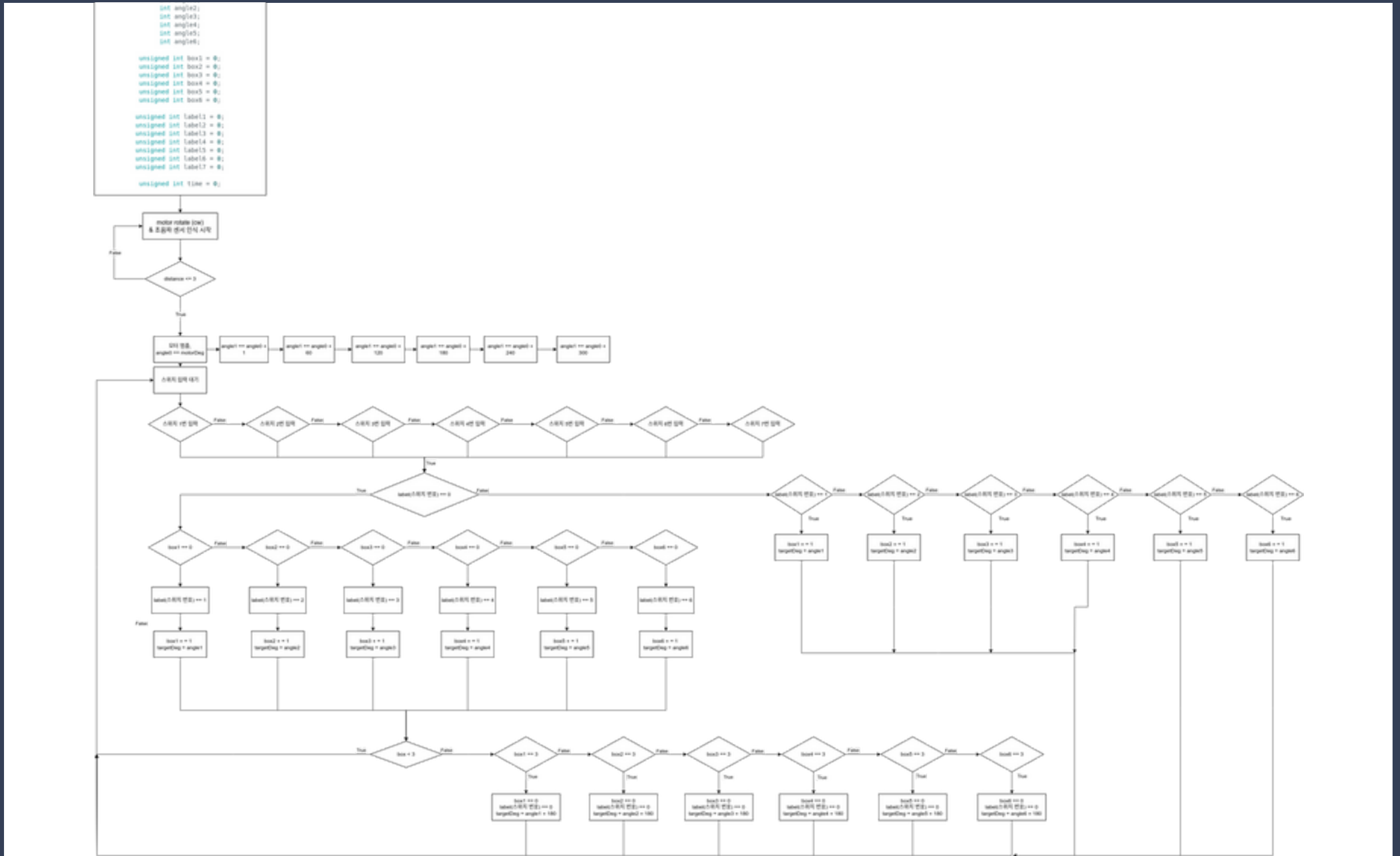
➤ 스위치

택배가 들어오는 신호를 스위치를 이용하여 대신하였습니다.

03 구현하기

순서도

- 이미지로 크게 보기(다음장)
- 사이트로 보기(글씨 클릭)



START

```
const int motorPin = 11;
const int motorPin2 = 8;
const int stop = 0;
const int start = 255;
const int encoderPinA = 2;
const int encoderPinB = 4;
const int TRIG = 6;
const int ECHO = 7;
```

```
int encoderPos = 0;
const float ratio = 360. / 204. / 25.;
float sp = 25;
float targetdeg;
```

```
int angle1;
int angle2;
int angle3;
int angle4;
int angle5;
int angle6;
```

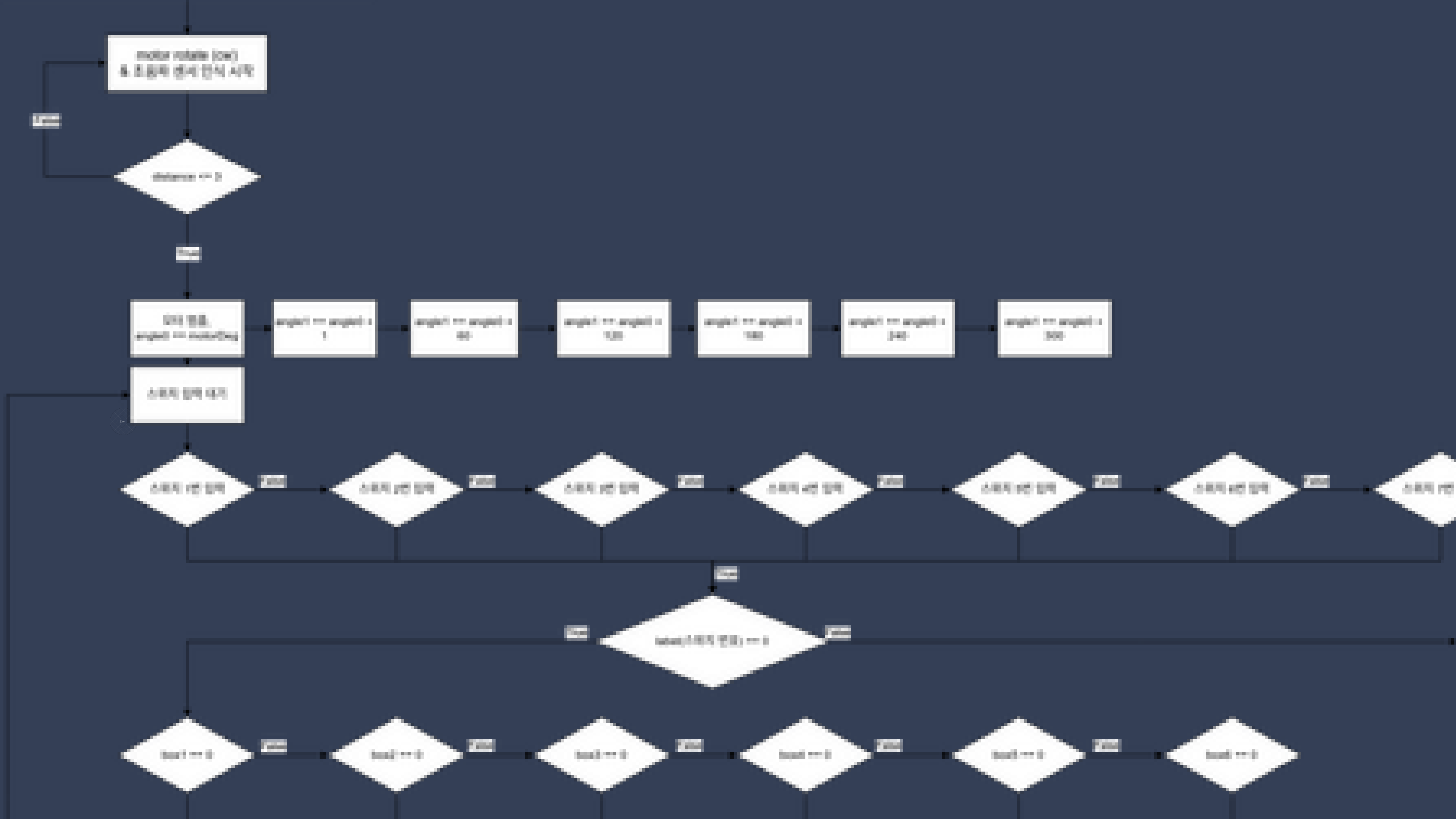
```
unsigned int box1 = 0;
unsigned int box2 = 0;
unsigned int box3 = 0;
unsigned int box4 = 0;
unsigned int box5 = 0;
unsigned int box6 = 0;
```

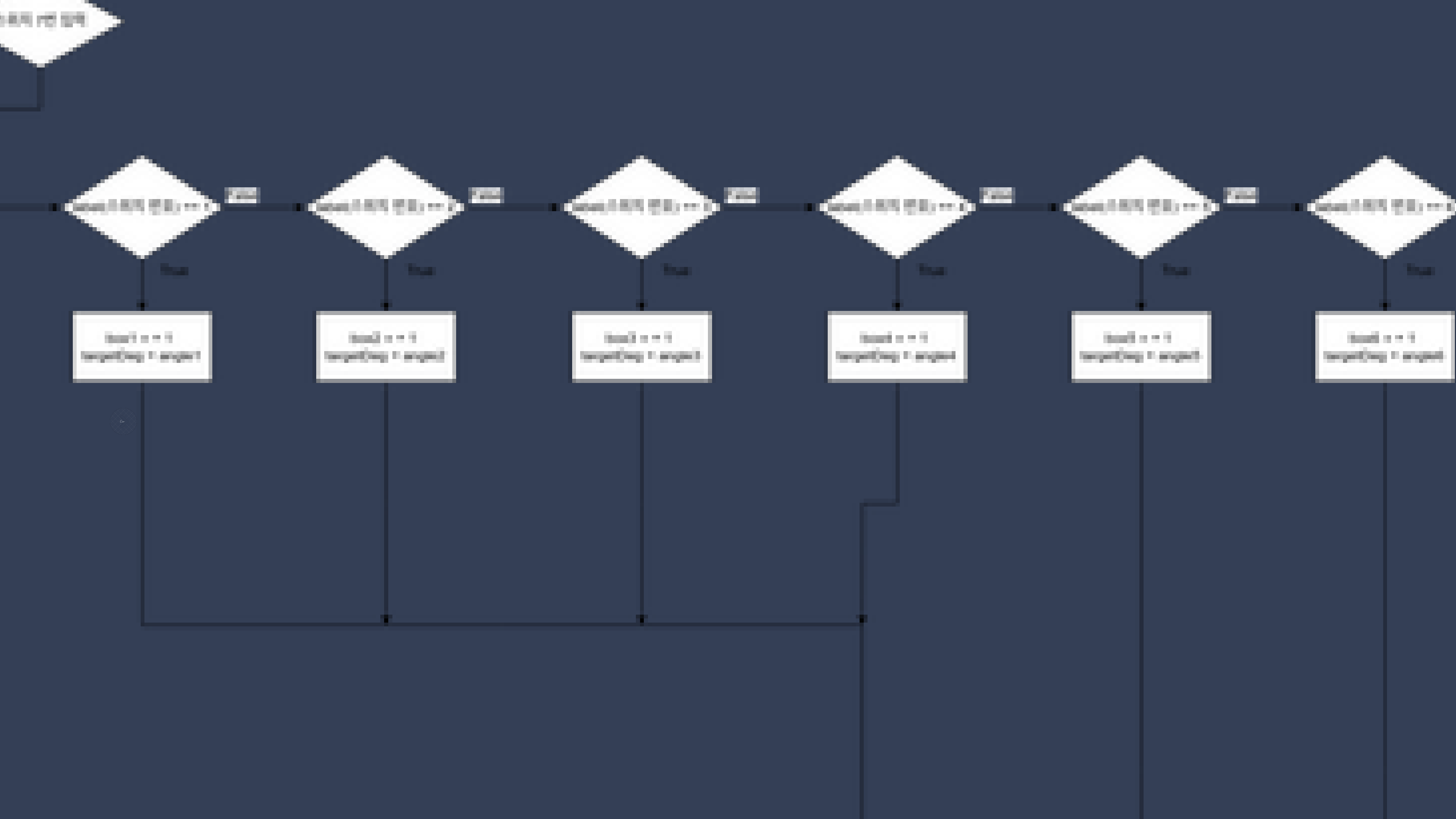
```
unsigned int label1 = 0;
unsigned int label2 = 0;
unsigned int label3 = 0;
unsigned int label4 = 0;
unsigned int label5 = 0;
unsigned int label6 = 0;
unsigned int label7 = 0;
```

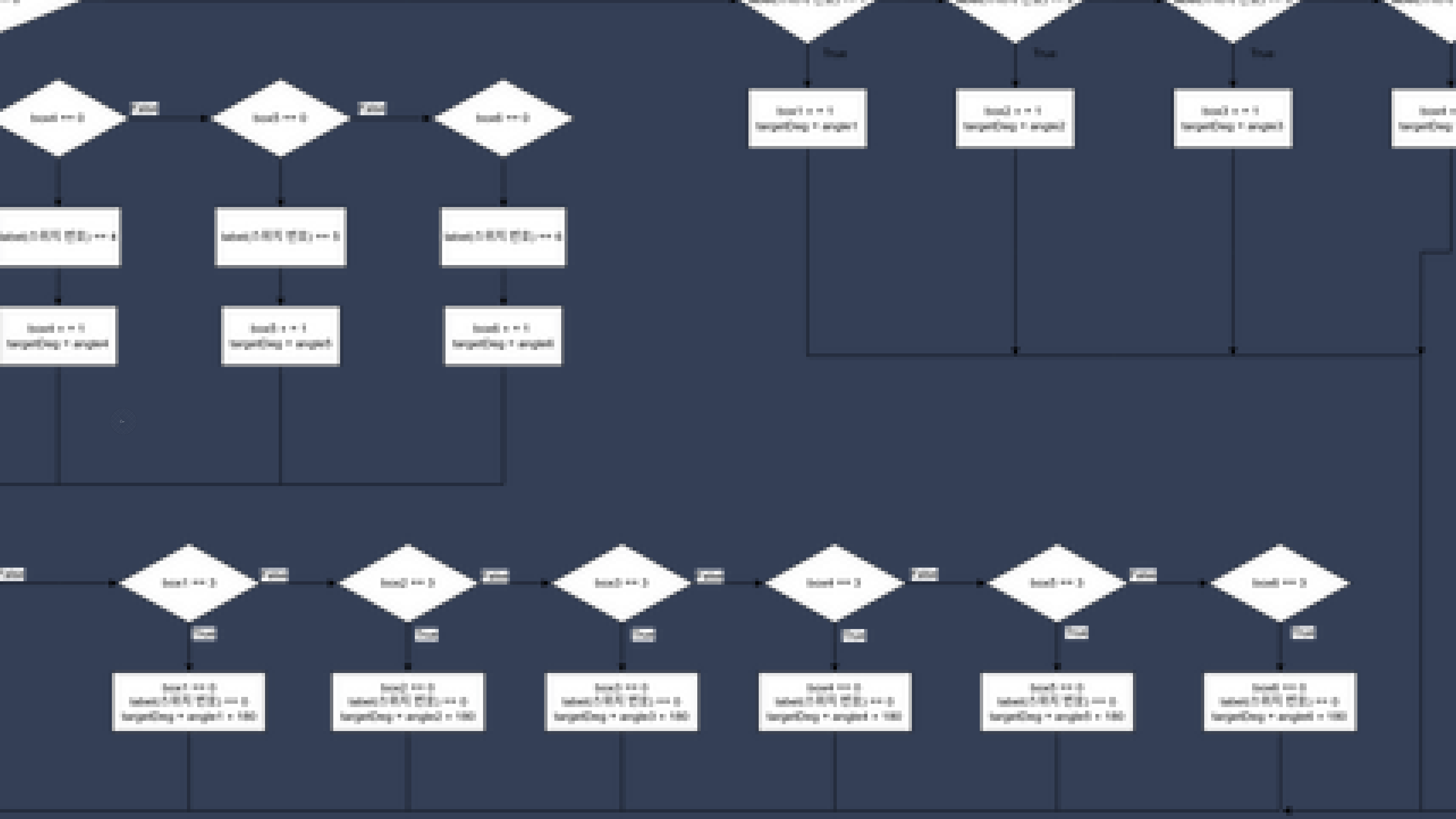
```
unsigned int time = 0;
```

motor rotate (cw)
& 0.5cm의 범위 안에서 회전

distance == 0







03 구현하기

알고리즘 구상

I 순서도 추가설명

ANGLE TOGGLE

처음 물건이 들어오며, 시스템가동이 되는 동시에 첫 바구니 위치의 각도를 'angle'에 저장해둡니다.

DO MOTOR, DO ENCODER

'angle'값이 저장이 됨과 동시에 나머지 바구니마다의 각도가 자동 계산되어 'boxangle'에 한번에 저장이 되며, 이를 이용하여 원하는 (각도의) 바구니로 자유롭게 이동합니다.

ANGLE

바구니에 들어간 택배의 갯수를 셉니다. 하나의 바구니에 최대 3개가 들어가도록 설정해놓았기에 3개들어감과 동시에 로봇을 부릅니다.

BOX

모든 바구니에 택배가 다 차고, 미등록 '라벨링'의 택배가 들어온다면, 로봇을 불러 바구니를 비웁니다.

LABEL

들어오는 택배의 주소에 맞춰, 바구니에게 이름을 지어줍니다. 'labeling'은 바구니의 이름을 저장해놓은 곳입니다.

03 구현하기

CODE

* 이는 기술검증을 위해, 작고 간단하게 구성되었습니다. 아파트에 적용하게 된다면, 규모가 더 커질것으로 예상됩니다.

```
697     if (label1 != 0 && label2 != 0 && label3 != 0 && label4 != 0 && label5 != 0 && label7 != 0)
698     {
699         if (digitalRead(button6) == HIGH)
700         {
701             if (toggle6 == 0)
702             {
703                 toggle6 = 1;
704             }
705         }
706         else
707         {
708             if (toggle6 == 1)
709             {
710                 angleToggle = 2;
711                 Serial.println("S T O P !");
712             }
713         }
714     }
715     else
716     {
717         if (digitalRead(button6) == HIGH)
718         {
719             if (toggle6 == 0)
720             {
721                 toggle6 = 1;
722             }
723         }
724         else
725         {
726             if (toggle6 == 1)
727             {
728                 if (label6 == 0)
```

어려웠던 부분 / 아쉬운 부분

의도한대로 잘 되었는지

팀원들의 의견을 익명으로 모아보았습니다.

- **A** : 전체적 흐름을 순서도로 정리하는 것이 어려웠고, 주문한 부품의 도착시간에 따라서 시도해볼 수 있는 시간이 달라지는 부분이 아쉬웠습니다.

- **B** : delay를 사용하지않고, flag나 time과 같은 변수를 사용함으로써 코드를 많이 줄이고 싶었지만, 알고리즘을 짜는 부분이 어려웠습니다. 다음에 다시 하게 된다면 하드웨어를 구조적으로 더 간단하게 만들고 싶습니다.

- **C** : LCD까지 화려하게 디자인하여, 레일을 타고 들어오는 택배와 로봇마저 표현하고 싶었지만 못한 점이 아쉽습니다.

THANK YOU

Realization of 'STATION'

