

소프트웨어 프로젝트 2

뱅뱅 제어 (Bang Bang Control)

2024년 2학기

국민대학교

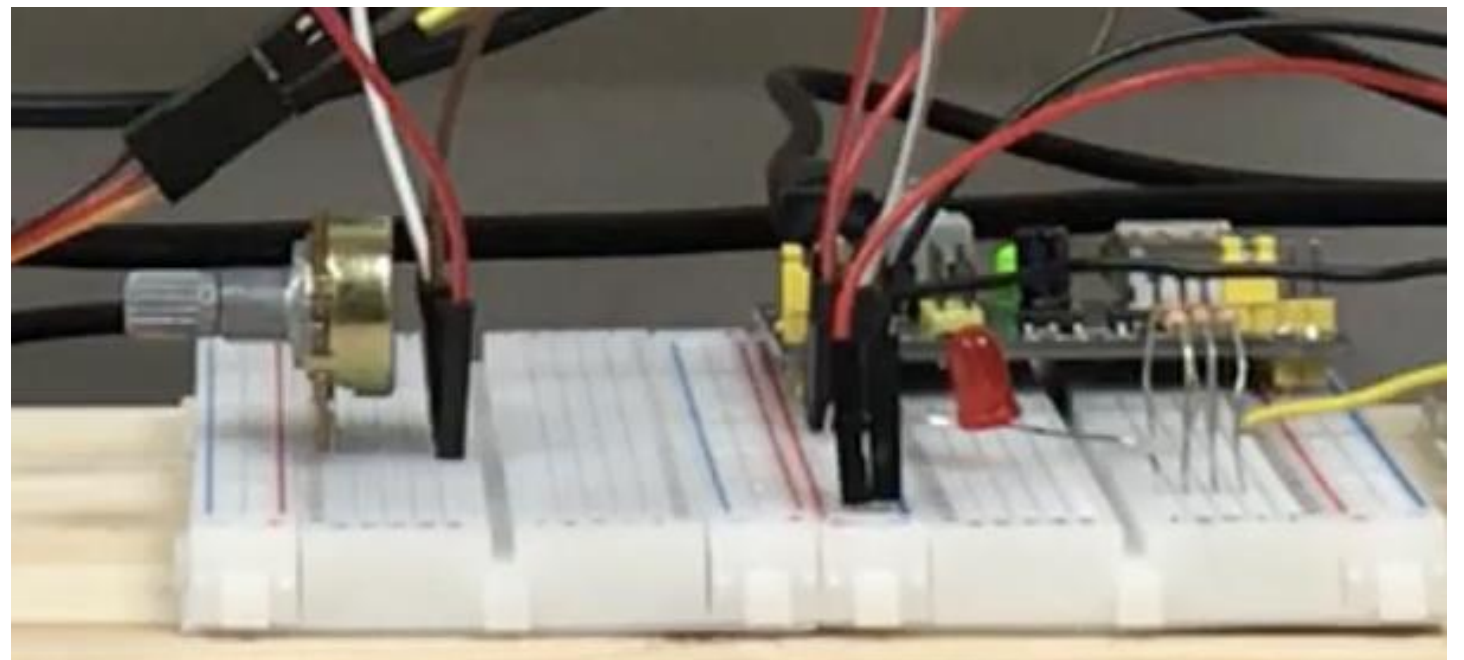
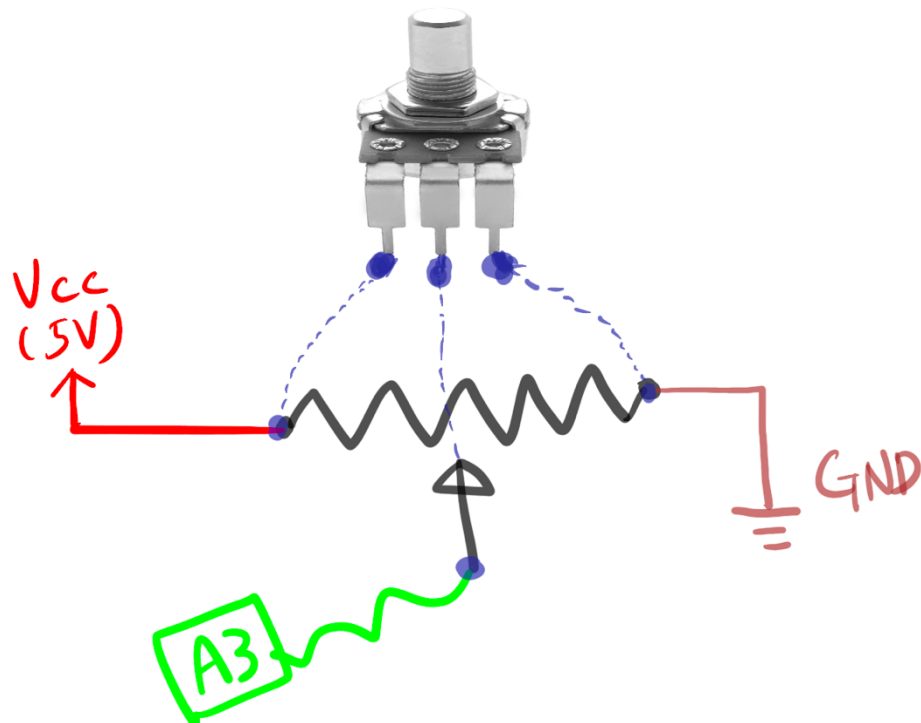
소프트웨어학부/인공지능학부

주용수, 최진우, 한재섭, 허대영

{ysjoo, jaeseob, jnwochoi, dyheo}@kookmin.ac.kr

회로 구성

- `#define PIN_LED 9` // 뱅뱅 제어 확인용 LED 연결
- `#define PIN_SERVO 10` // 레일플레이트 제어용 서보
- `#define PIN_IR A0` // 적외선 센서
- `#define PIN_VAR A3` // 서보 조정용 가변저항
 - 브레드보드에 직접 삽입하여 설치할 것



실습 1: 레일플레이트 최대 가동범위 확인

- 프레임워크 설계에 따른 레일 플레이트의 최대 기울기 범위 확인

- 전원을 끈 상태에서 서보혼(=로워암)을 서보 기어에 **임시 조립**
- 서보혼을 돌려 레일플레이트를 수평으로 맞추고 (탁구공으로 확인)

내벽에 B 위치를 연필로 표기

A 14

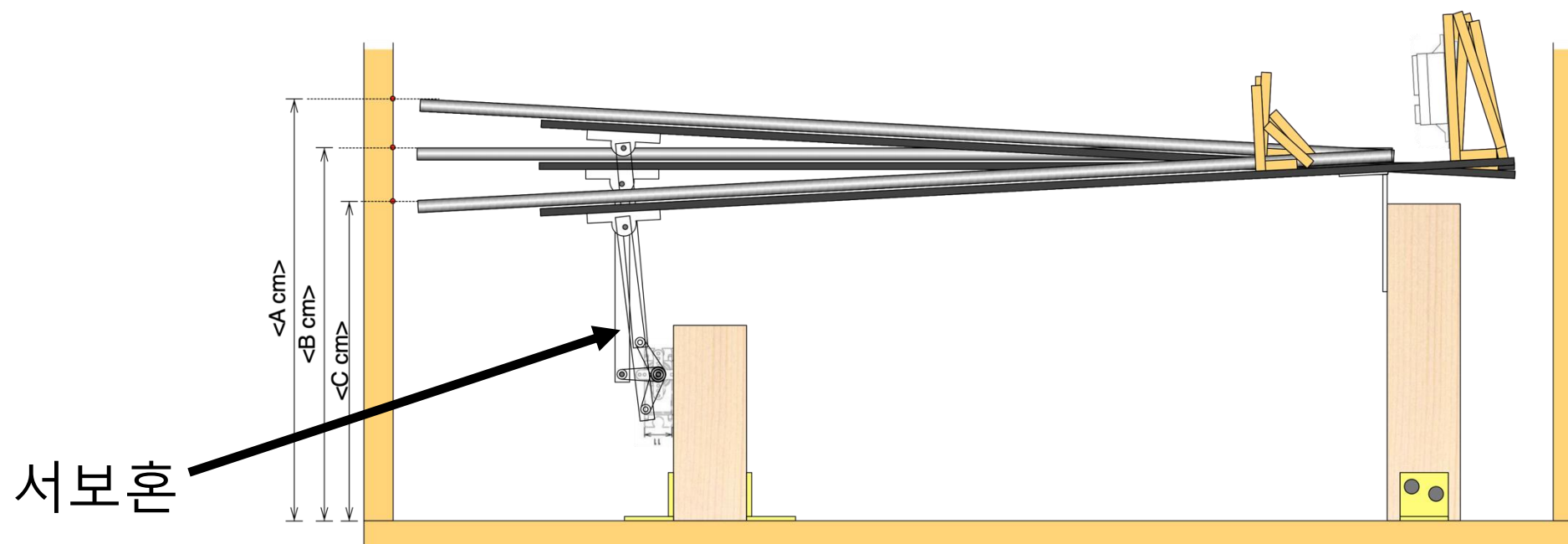
- 서보혼을 손으로 천천히 돌리면서 최대 가동범위 A, C를 연필로 표기

B 14.9

- (A-B)와 (B-C)의 값이 다를 경우 A를 줄이거나 C를 키워서

C 16.5 -> 16

상하 가동 폭이 대칭이 되도록 조절할 것 => $2B = A+C$ 가 되어야 함



실습 2: 서보혼-서보 조립

- 서보 조정 코드 준비

- https://www.dropbox.com/scl/fi/od2tvhi9cepxfedqhjx95/22_servo_range_adj.ino?rlkey=genmixack3051fvk2ov2ytw2q&dl=0

- 서보혼 분리 후 전원 연결 및 위 코드 실행

- 서보혼이 연결되지 않은 상태(**중요**)에서 서보 기어를 `_DUTY_NEU`으로 초기화
- 위 샘플 코드의 `_DUTY_NEU` 초기값은 1500 us로 설정되어 있음

- 서보혼-서보 조립

- 아두이노 전원이 들어와 있는 상태에서 조립할 것
- 레일이 최대한 수평에 가깝게 되도록 조립
 - 수평이 완벽하게 맞지는 않아도 됨 (실습 3에서 미세 조정)
- **주의: 조립 과정에서 가변저항 노브(knob)를 절대로 조작하지 말 것**

실습 3: 서보 펄스폭 범위 측정

- 레일플레이트 가동범위에 따른 서보 펄스폭 측정
 - 아두이노 및 전원공급모듈의 **전원을 끈 상태에서** 가변저항 노브를 중간 위치로 옮김
 - 노브를 시계방향 최대, 반시계방향 최대로 돌려서 최대 회전 범위를 확인한 후 적당히 중간 지점에서 멈출 것
 - 전원을 연결한 후 레일플레이트 끝이 A, B, C에 위치하도록 노브를 조심스럽게 돌릴 것
 - 노브를 끝까지 돌려도 A 또는 C 위치까지 도달하지 않는 경우 다시 전원을 끄고 노브의 초기 위치를 재조정할 것
 - **주의: 노브를 빨리 돌려서 가동범위를 초과할 경우 서보암이 파손됨**

실습 3: 서보 펄스폭 범위 측정

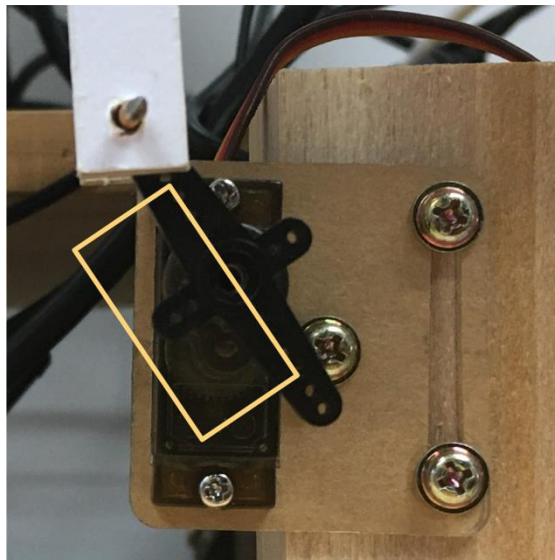
- A, B, C 위치에 해당하는 서보 duty 측정
 - 위치별로 시리얼 모니터의 duty 값을 확인
 - DUTY_NEU를 측정할 때는 레일 플레이트 위에 탁구공을 놓고 좌우로 굴러가면서 최적의 수평 위치를 찾을 것
 - 서보 중심축 각도 측정
 - 각도기가 없을 경우 사진 촬영하여 파워포인트 등 프로그램으로 측정

31
343 -17

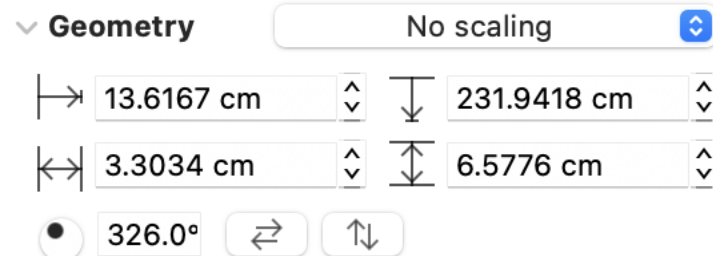
B 1330

A 1640

C 1040



레일플레이트 높이	서보 펄스폭 (us)	서보 중심축 각도
B (수평)	<u>DUTY_NEU</u> = ?	0°
A (낮을 때)	<u>DUTY_MAX</u> = ?	D°
C (높을 때)	<u>DUTY_MIN</u> = ?	E°



과제: Bang-Bang Control

- 탁구공의 거리 x 에 따라 레일 기울기 제어
 - $x < 175$ mm: 탁구공이 센서로부터 먼 쪽으로 움직이도록 서보 제어
 - $x > 175$ mm: 탁구공이 센서와 가까운 쪽으로 움직이도록 서보 제어

```
27 // Target Distance
28 #define _DIST_TARGET    175 //
```

```
90 // bang bang control
91 if (dist_ema > _DIST_TARGET) {
92     duty_target = _DUTY_NEU - _BANGBANG_RANGE;
93     digitalWrite(PIN_LED, 0);
94 } else if (dist_ema < _DIST_TARGET) {
95     duty_target = _DUTY_NEU + _BANGBANG_RANGE;
96     digitalWrite(PIN_LED, 1);
97 }
```


과제: Bang-Bang Control

- 코드 템플릿

- https://www.dropbox.com/scl/fi/frkwildcmt1q5kk8vxbs7/22_bangbangcontrol.ino?rlkey=tvvrbcfxyzc0puv53vp1k9qsb&dl=0

- 파라미터/상수 설정 (필수: ???를 ‘자신만의’ 값으로)

- Line 9-11: 이전 실습에서 확인 최적의 INTERVAL 값으로 설정
 - 이후에 제어/모니터 상황에 따라 거리 측정, 시리얼 출력 INTERVAL 조정
- Line 14: 적당한 ema_alpha 값 설정
- Line 18-20: 실습 3에서 찾아낸 duty 값들로 수정
- Line 22: 실습 3에서 찾아낸 ID - EI°로 수정
- Line 23 : 적당한 Servo 각속도로 수정
- Line 25: bangbang 제어 범위 설정
- Line 57: 적당한 시리얼 통신 속도 설정
- Line 87: `ir_sensor_filtered(n, p, 0)`의 **n, p**를 적당한 값으로 설정
- Line 139: 10주 2강에서 얻은 식으로 `volt_to_distance()` 함수 수정

과제: Bang-Bang Control

```
8 // Event interval parameters
9 #define _INTERVAL_DIST      ??? // distance sensor interval (unit: ms)
10 #define _INTERVAL_SERVO     ??? // servo interval (unit: ms)
11 #define _INTERVAL_SERIAL    ??? // serial interval (unit: ms)

13 // EMA filter configuration
14 #define _EMA_ALPHA ??? //

17 // Servo adjustment - Set _DUTY_MAX, _NEU, _MIN with your own numbers
18 #define _DUTY_MAX ??? // 2000
19 #define _DUTY_NEU ??? // 1500
20 #define _DUTY_MIN ??? // 1000
21
22 #define _SERVO_ANGLE_DIFF   ??? // Replace with |D - E| degree
23 #define _SERVO_SPEED        ??? // servo speed
24
25 #define _BANGBANG_RANGE     ??? // duty up and down for bangbang control
```

과제: Bang-Bang Control

```
56    // initialize serial port
57    Serial.begin(???);
```

```
86    // get a distance reading from the distance sensor
87    dist_filtered = volt_to_distance(ir_sensor_filtered(???, ???, 0));
```

```
135 float volt_to_distance(int a_value)
136 {
137     // Replace line into your own equation
138     // return (6762.0 / (a_value - 9) - 4.0) * 10.0;
139     return ???;
140 }
```

과제: Bang-Bang Control

- 서보에 공급되는 제어 펄스 폭은 다음과 같음
 - $\text{_DUTY_NEU} + \text{_BANGBANG_RANGE}$
 - $\text{_DUTY_NEU} - \text{_BANGBANG_RANGE}$
- 탁구공이 잘 구르지 않을 경우
 - _BANGBANG_RANGE 값을 키워볼 것
- 탁구공이 멈춤장치 또는 케이스 측벽에 부딪칠 경우
 - _BANGBANG_RANGE 값을 조금씩 줄여볼 것
 - 한 쪽에만 부딪칠 경우 _DUTY_NEU 값을 미세 조정해볼 것
- 과제 제출시에는 탁구공이 멈춤장치 또는 케이스에 부딪쳐도 무방함(감점 없음)

과제: Bang-Bang Control

- 그리고
 - 탁구공과의 거리 측정 주기와 서보 속도의 관계가 제어에 영향이 있음
 - BangBang 제어는 원래 잘 안됨 !!!
 - 잘 되면 좋지만,
 - 주 목적은 각 INTERVAL, 서보속도, BANGBANG 제어량과의 관계를 확인하는 확인하는 것..
- ➔ 실제 PID 제어에 활용

과제: Bang-Bang Control

- 제출물
 - 완성한 코드 GitHub 주소 (파일명: 22_bbc_학번.ino)
 - 영상 YouTube link (일부공개로 설정)
 - 제출 마감: 13주 1강 수업 전까지
 - 과제코드: 22P11
- 영상 촬영시 주의사항
 - 탁구공 움직임과 LED가 나오도록
 - Shorts로 올리지 마세요(반드시 가로 방향으로 촬영할 것).
 - 프레임워크 밑판 측면에 학번, 성명 기입할 것 (아래 사진 참조)

