

# 캡스톤디자인1

---

## 꽃병

2290027 인채빈 (팀장)

2090004 구동해

2090059 허진경

2290031 김은선



## 02. 목차

---

**01** 주제 소개

**02** 기능 소개

**03** 오류 및 해결 과정

# 00. 역할분담

## 인채빈(팀장)

- 발표자료 (ppt, 포스터)제작
- 외관 제작
- 보고서 작성
- 하드웨어 조립
  - 센서파트 (초음파, 카메라 인식, MP3 모듈)
- 아두이노회로 자료조사
  - 각종모듈 회로연결

## 구동해

- 외관 디자인
- 외관 제작
- 하드웨어 조립
  - 모듈파트 (도트LED, 스피커, 서브모터)
- 모터연결 및 스피커 출력코드 자료조사

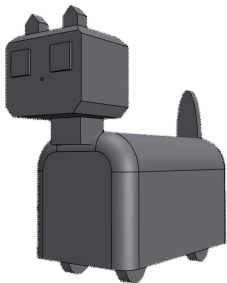
## 허진경

- 앱 개발
  - 블루투스로 강아지와 핸드폰 연결
- AI 카메라 학습 시키기
- 아두이노 코딩
  - 카메라, 모터 코딩
- 카메라 코드에 대한 자료조사

## 김은선

- 발표
  - 중간발표 및 최종발표
- 아두이노 코딩
  - 센서를 통한 LED 출력 코딩
- 아두이노 코드 자료조사
  - LED 입출력코드
  - 초음파센서 입력코드

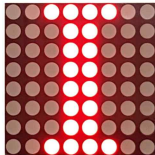
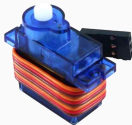
# 01. 주제 소개



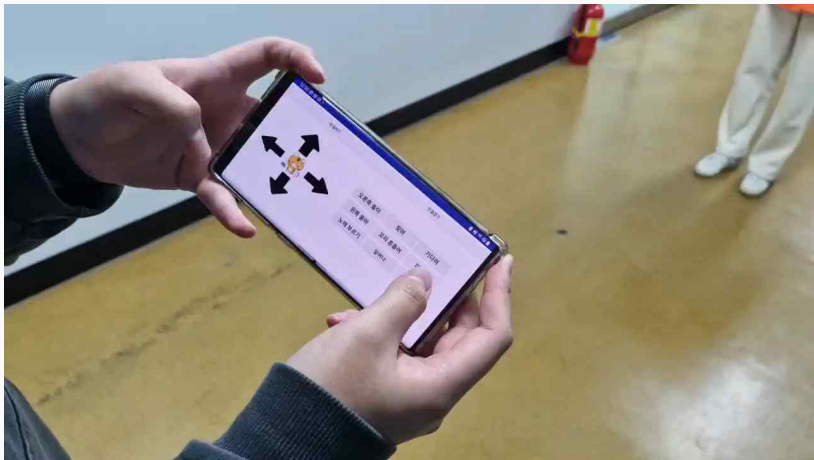
## 01 구현방법

- 입력장치 > 아두이노 > 출력장치

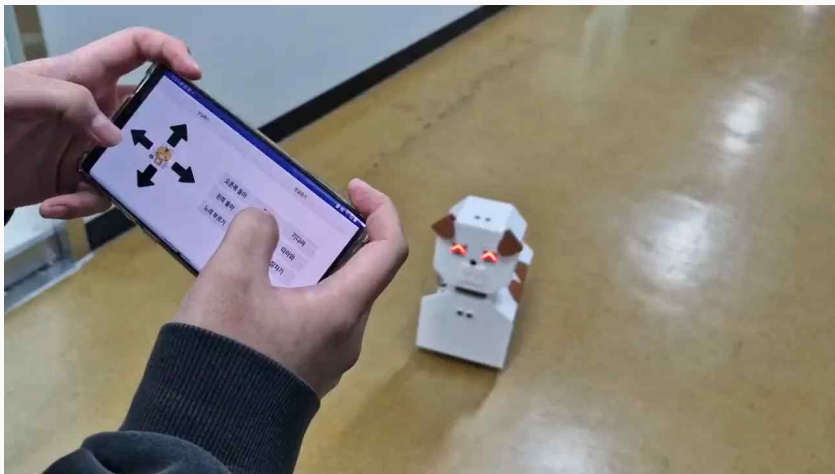
## 02 주요장치



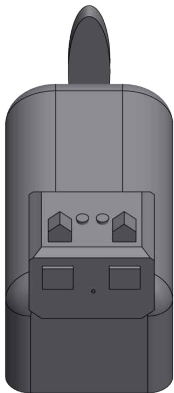
## 02. 동작 - 카메라



## 02. 동작 - 블루투스



## 03. 로봇 기능소개



### 01 소리 구현 방법

- 아두이노 mp3모듈에 강아지 소리 녹음 후 스피커로 재생
- 초음파 센서로 손을 감지해 스피커를 통한 소리 재생

### 02 움직임 구현 방법

- 초음파 센서로 손을 감지해 LED를 통해 눈 웃음표정 구현
- pixycam2를 통해 사물을 인식, 모터바퀴와 보조바퀴를 통해 방향 상관없이 사물을 따라다님.
- 초음파 센서로 물체를 감지해 서보모터를 통하거나 블루투스로 연결된 스마트폰 어플로 꼬리 움직임 구현
- 블루투스로 연결된 스마트폰 어플로 한바퀴 도는 움직임 구현

## 04. 카메라

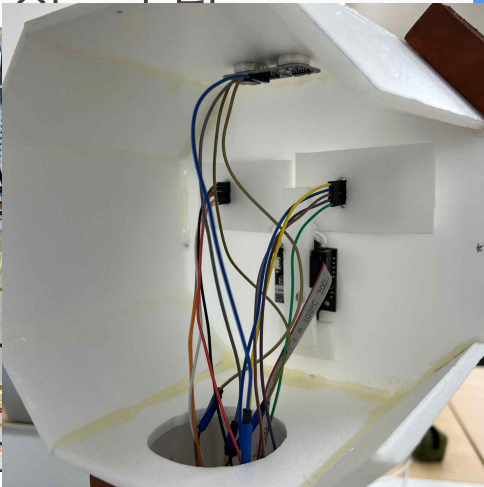
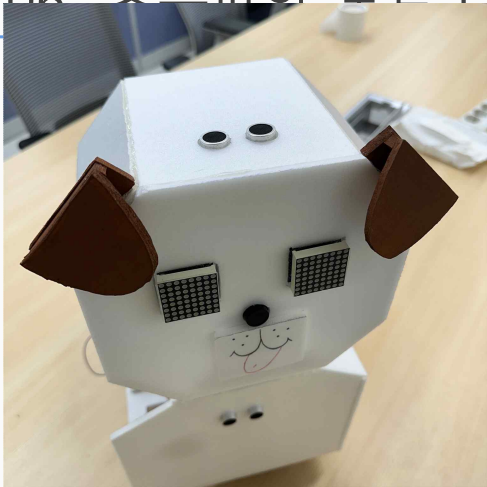




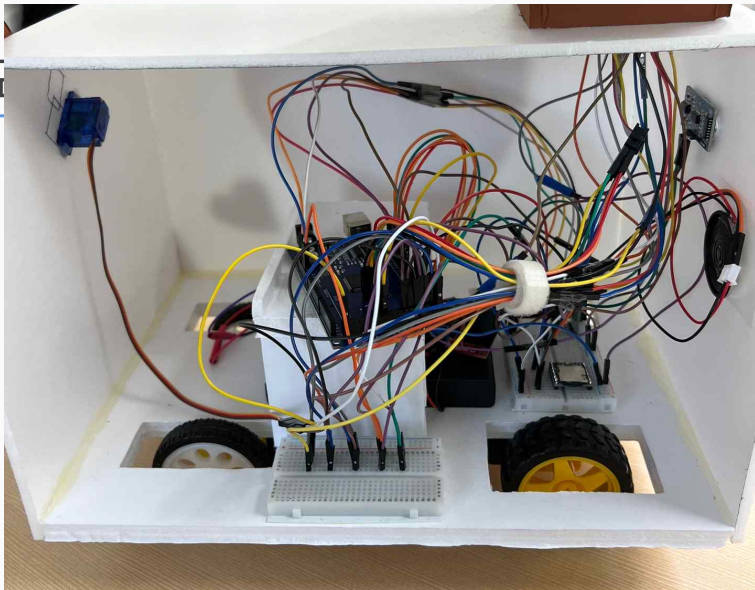
## 05. 앱 제작



## 06 쿠키파와 드림 LED, 서버무터



07. 5



```

// 맵핑 함수
float mapfloat(long x, long in_min, long in_max, long out_min, long out_max) {
    return (float)(x - in_min) * (out_max - out_min) / (float)(in_max - in_min) + out_min;
}

// Pixy 카메라 데이터 읽기
float pixyCheck() {
    // Pixy2에서 감지한 블록 개수 가져오기
    int blocks = pixy.ccc.getBlocks();

    if (blocks > 0) {
        // 첫 번째 블록 정보 가져오기
        signature = pixy.ccc.blocks[0].m_signature; // 블록의 서명 정보
        height = pixy.ccc.blocks[0].m_height; // 블록의 높이
        width = pixy.ccc.blocks[0].m_width; // 블록의 너비
        x = pixy.ccc.blocks[0].m_x; // 블록의 x 좌표 (화면상의 위치)
        y = pixy.ccc.blocks[0].m_y; // 블록의 y 좌표 (화면상의 위치)

        // 중심 좌표 계산 (블록 중심 좌표를 바로 사용)
        cx = x; // 블록의 x 중심 좌표
        cy = y; // 블록의 y 중심 좌표

        // 중심 좌표 정규화 (-1 ~ 1로 변환)
        float normalizedCX = ((float)cx - 158.0) / 158.0; // 0~316을 -1~1로 변환 (x축)
        float normalizedCY = ((float)cy - 104.0) / 104.0; // 0~208을 -1~1로 변환 (y축)

        // 정규화된 값이 -1과 1 사이에 있도록 제한
        normalizedCX = constrain(normalizedCX, -1, 1);
        normalizedCY = constrain(normalizedCY, -1, 1);

        // 블록의 면적 계산 (면적 = 너비 * 높이)
        area = width * height;

        // 디버깅 출력: 정규화된 중심 좌표 출력
        Serial.print("Normalized CX: ");
        Serial.print(normalizedCX);
        Serial.print(" Normalized CY: ");
        Serial.println(normalizedCY);

        return normalizedCX; // 정규화된 중심 좌표 반환 (x값만 반환)
    } else {
        // 블록이 감지되지 않았을 경우
        Serial.println("No object detected");
        return 0; // 객체 미검출 시 0 반환
    }
}

```

```

void handleBluetooth() {
    if (BLUETOOTH_SERIAL.available()) {
        String text = BLUETOOTH_SERIAL.readStringUntil(0x0A);
        Serial.print("Received Command: ");
        Serial.println(text);

        if (text == "1") moveRobot(90, 90); // 앞으로 이동
        else if (text == "2") moveRobot(255, -130); // 오른쪽 회전
        else if (text == "3") moveRobot(-130, 255); // 왼쪽 회전
        else if (text == "4") moveRobot(-90, -90); // 뒤로 이동
        else if (text == "10"){ //왼쪽 돌아
            moveRobot(-130, 255);
            delay(6500);
            moveRobot(0, 0);
        }
        else if (text == "13"){ //오른쪽 돌아
            moveRobot(255, -130);
            delay(6500);
            moveRobot(0, 0);
        }
        else if (text == "11") { //찾어
            // DFPlayer 트랙 재생 디버깅
            Serial.println("Attempting to play track 1");
            myDFPlayer.play(1); // 1번 트랙 재생
            Serial.println("Playing track 1");

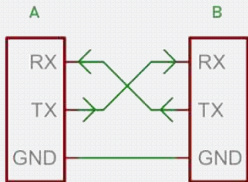
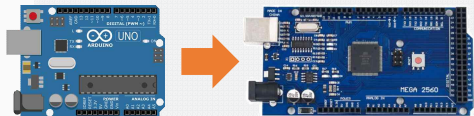
            Serial.println("Command 11: Showing custom3 LED");

            // custom3 패턴을 5초 동안 출력
            unsigned long startMillis = millis(); // 시작 시간 기록
            while (millis() - startMillis < 2700) { // 5000ms (5초) 동안 반복
                showLED(custom3, 1); // custom3 출력
            }
        }
    }
}

```

# 09. 오류 및 해결과정

## 01 보드변경



## 02 카메라

```
// 맵핑 함수
float mapfloat(long x, long in_min, long in_max, long out_min, long out_max) {
    return (float)(x - in_min) * (out_max - out_min) / (float)(in_max - in_min) + out_min;
}

// pixy 카메라 데이터 읽기
float pixycheck() {
    // pixy에서 감지한 블록 개수 가져오기
    int blocks = pixy.ooo.getBlocks();

    if (blocks > 0) {
        // 첫 번째 블록 정보 가져오기
        signature = pixy.ooo.blocks[0].m_signature; // 블록의 서명 정보
        height = pixy.ooo.blocks[0].m_height; // 블록의 높이
        width = pixy.ooo.blocks[0].m_width; // 블록의 너비
        x = pixy.ooo.blocks[0].m_x; // 블록의 x 좌표 (화면상의 위치)
        y = pixy.ooo.blocks[0].m_y; // 블록의 y 좌표 (화면상의 위치)

        // 중심 좌표 계산 (블록 중심 좌표를 바로 사용)
        cx = x; // 블록의 x 중심 좌표
        cy = y; // 블록의 y 중심 좌표

        // 중심 좌표 정규화 (-1 ~ 1로 변환)
        float normalizedCX = ((float)cx - 150.0) / 150.0; // 0~316를 -1~1로 변환 (x축)
        float normalizedCY = ((float)cy - 104.0) / 104.0; // 0~208을 -1~1로 변환 (y축)

        // 정규화된 값이 -1과 1 사이에 있도록 제한
        normalizedCX = constrain(normalizedCX, -1, 1);
        normalizedCY = constrain(normalizedCY, -1, 1);

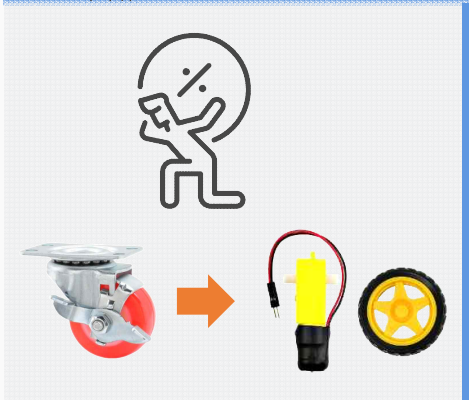
        // 블록의 면적 계산 (면적 = 너비 * 높이)
        area = width * height;

        // 디버깅 출력: 정규화된 중심 좌표 출력
        Serial.print("Normalized CX: ");
        Serial.print(normalizedCX);
        Serial.print(" Normalized CY: ");
        Serial.println(normalizedCY);

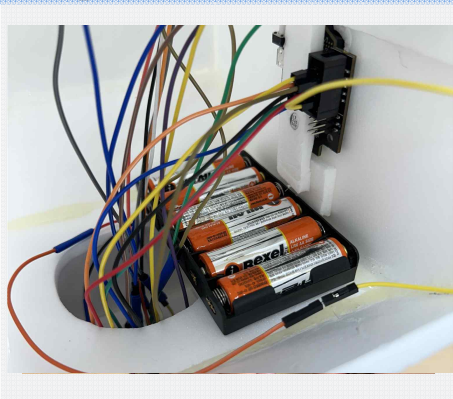
        return normalizedCX; // 정규화된 중심 좌표 반환 (x값만 반환)
    } else {
        // 블록이 감지되지 않았을 경우
        Serial.println("No object detected");
        return 0; // 객체 미검출 시 0 반환
    }
}
```

# 09. 오류 및 해결과정

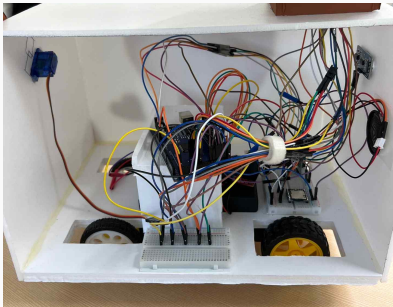
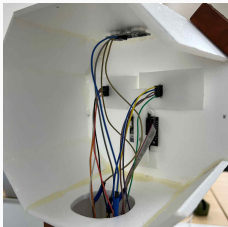
## 03 모터바퀴



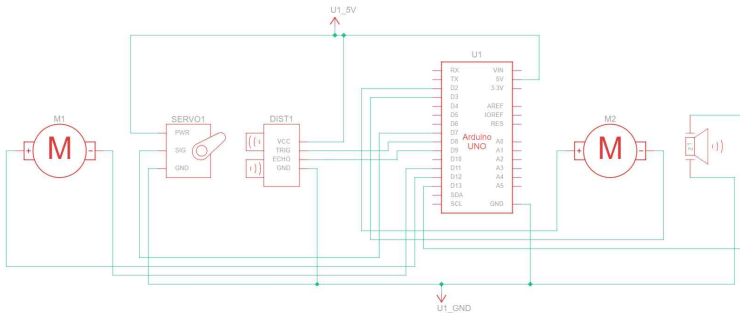
## 04 전압문제



## 13. 제작 완료



# 07. 아두이노 설계도





# 15. 진행도

추진일정		구분				9월				10월				11월				12월			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
과제선정																					
자료조사																					
감아지 설계																					
앱 제작																					
아두이노 코딩																					
감아지 제작																					
개발모듈구현	카메라 모듈																				
	센서 모듈																				
	LED 모듈																				
개별 테스트	블루투스																				
	스피커																				
	모터 테스트																				
	센서 테스트																				
	LED 테스트																				
	카메라 사물인식																				
통합구현	앱 LED 구현																				
	앱 모터 구현																				
통합테스트																					
최종발표																					

	계획
	수행

# Q & A

