



# Busan science high school

## 2023 Ocean ICT Festival

## 2023 BOIF

C  
11

QR 코드 영역  
QR 삽입 후  
테두리 삭제

Youtube 영상 QR

## 인공지능 로봇 **염전** 시스템 구축

팀 "오천년의 신비"

2315 윤성빈

2320 최창원

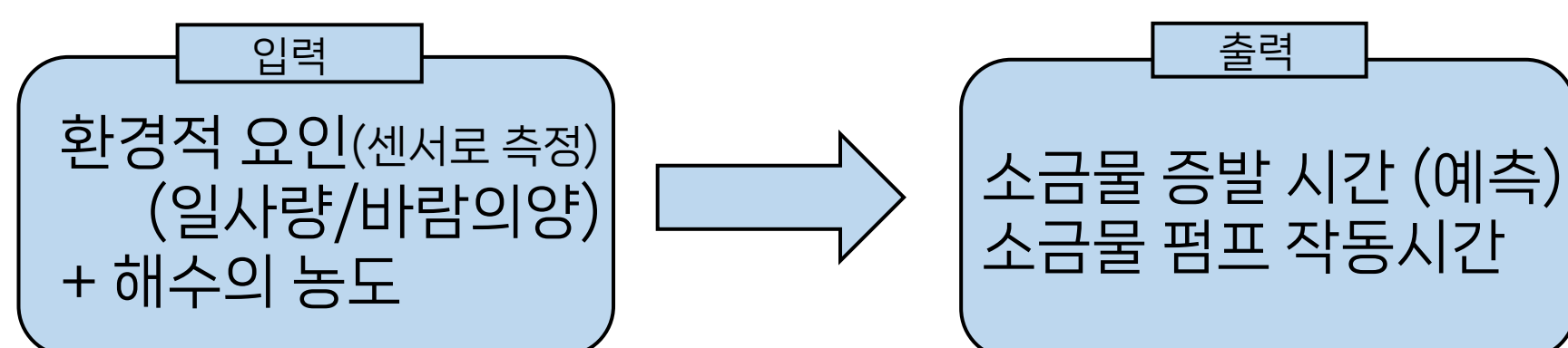
### 탐구 동기

오션 ICT 주제를 찾던 중 몇 년 전 뉴스에서 본 신안 염전 노예 사건이 생각이 났다. 노예가 생기는 이유는 소금 채집이 고된 작업이고, 혼자서는 할 수 없는 스케일이 큰 작업이라 악덕 사장들이 사람을 속여 노예로 부리는 것이라 생각하였다. 2014년에 떠오른 사건이었지만, 1년 전 몇몇 뉴스 취재로 현재진행형으로 일어나고 있다 한다. 또한 현재 후쿠시마 오염수 방출 논란과 관련하여 천일염을 구입하는 사람이 늘어나 이들의 노동량이 더욱 증가하고 있다. 물론 악한 행동을 한 사장 자체의 문제도 있을 것이지만, 염전에서 천일염 채집을 인공지능을 이용한 자동화를 하면 다음과 같은 행위가 감소할 것 같아 주제를 생각하게 되었다.



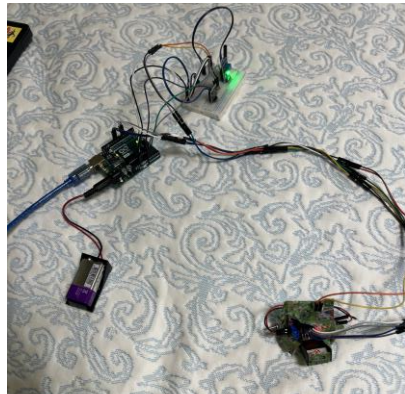
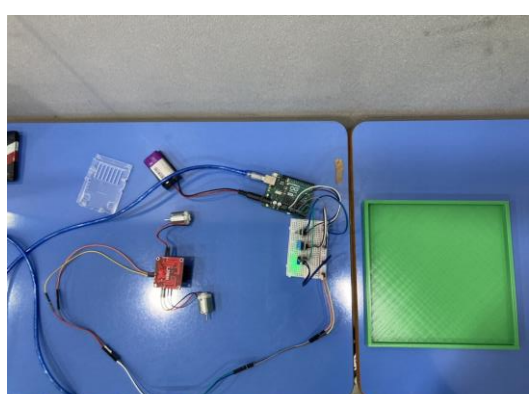
### 입출력

환경적 요인과 소금물(해수)의 농도에 맞게, 인공지능이 소금이 생기는 시간을 예측하여 측정하여야 하므로, 환경적 요인(일사량, 바람의 양)를 입력하여 주면 소금물(해수)의 완전한 증발 시간을 출력하여 바닷물을 효율적으로 끌어올 수 있도록 설계하였다.



### 작품 설계 및 실험

- 염전 로봇 만들기 : 아두이노에 모터드라이브를 몸체로 한 미니 염전 로봇을 만들었다.
- 염전 시스템 제작 : 로봇이 연결된 아두이노를 브레드보드에 연결한 후 모터 드라이버와 조도 센서, 진동 감지 센서를 연결하였다. 원래는 온도 센서도 연결하여 실험을 진행하려 했으나 잘못된 코드 입력으로 실험중 고장으로 사용하지 못하였다. (ㅜㅜ)



빛의 양	조도 센서 값
빛이 없을 때	200
빛 1단	120
빛 2단	95
빛 3단	70

바람	진동센서 값
바람이 없을 때	1023
바람 약	1020
바람 중	1010
바람 강	990

입력받은 센서의 값

- 염전 소금 증발시간 비교 실험 : 이후, 3D 프린터로 염전 형태의 판을 뽑았다. 빛과 바람을 조절해주기 위해 LED등, 선풍기를 사용하였다. 그 후, 제작한 염전 판에 35% 소금물을 40ml씩 넣고 모두 증발되어 소금이 나타나는 데 까지 걸리는 시간을 측정하였다. 조작 변인들을 조절해가며 데이터를 수집하였다. (빛과 바람이 동시에 없는 경우에는 증발에 10시간이 초과되었다.)



바람 x, 빛의 양	시간
빛 1단	4시간 28분
빛 2단	3시간 40분
빛 3단	2시간 21분

바람 약, 빛의 양	시간
빛 1단	1시간 23분
빛 2단	1시간 10분
빛 3단	58분

바람 중, 빛의 양	시간
빛 1단	1시간 2분
빛 2단	57분
빛 3단	52분

바람 강, 빛의 양	시간
빛 1단	49분
빛 2단	43분
빛 3단	38분

빛과 바람에 따른 증발 속도

### 코드 작성

- 앞에서 얻은 실험 결과 값으로 파이썬을 이용하여 회귀분석을 진행하였다.

```
In [1]: import numpy as np
        from sklearn.linear_model import LinearRegression

# 주어진 데이터
A = np.array([100, 95, 70, 120, 95, 70, 120, 95, 70, 120, 95, 70]).reshape(-1, 1)
B = np.array([1023, 1023, 1023, 1020, 1020, 1020, 1010, 1010, 1010, 990, 990, 990]).reshape(-1, 1)
y = np.array([208, 220, 141, 83, 70, 58, 62, 57, 52, 49, 43, 38])

# 피벗수 회귀분석 모델 생성
model = LinearRegression()

# 독립 변수, 종속 변수
X = np.concatenate((A, B), axis=1)

# 모델 학습
model.fit(X, y)

# 회귀계수와 종속 출력
print('회귀 계수 (기울기):', model.coef_)
print('절편:', model.intercept_)

# 새로운 데이터로 예측
new_data = np.array([[100, 1010]]) # 예시: A=100, B=1010인 경우
predicted_result = model.predict(new_data)
print('예측된 결과:', predicted_result)

회귀 계수 (기울기): [ 0.865   3.4151981]
절편: -3459.0031464816907
예측된 결과: [96.84893476]
```

A는 조도센서 값  
B는 진동센서 값  
Y는 그에 따른  
증발 시간(분)이다.

이를 토대로 회귀분석  
하는 코드이고,

아래에 결과 값을 보았을 때, 바람 세기가 빛 세기보다 더 영향이 큰 것을 알 수 있다.

- 그 후, 바람의 세기와 온도를 측정하는 코드를 C++언어로 작성해 아두이노에 입력하였다.

```
int sensorPin = A2;
int PRegistor = A0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(sensorPin, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  int value1 = analogRead(sensorPin);

  Serial.print("Wind Value: ");
  Serial.println(value1);

  int value2 = analogRead(PRegistor);
  Serial.print("Brightness Value: ");
  Serial.println(value2);

  delay(2000);
}
```

이 코드는 센서로 입력받은 조도센서와 진동센서 값을 2초 간격으로 출력하는 코드이다.

아날로그 값으로 입력받은 값은 숫자로 표현된다.

- 회귀분석한 식을 대입하여 바닷물 유입량을 측정하는 코드를 작성하였다.

```
2 const int lightSensorPin = A0; // 조도 센서
3 const int vibrationSensorPin = A2; // 진동 센서
4 const int motorPin = 5; // 바닷물 펌프 모터
5
6 void setup() {
7   pinMode(motorPin, OUTPUT); // 펌프 모터 (방향 출력으로 설정)
8   Serial.begin(9600); // 시리얼 통신 초기화
9 }
10
11 void loop() {
12   int lightValue = analogRead(lightSensorPin); // 조도 센서 값 읽기
13   int vibrationValue = analogRead(vibrationSensorPin); // 진동 센서 값 읽기
14   float motorDuration = 300000 / (0.865 * lightValue + 3.4151981 * vibrationValue - 3459.0031465); // 300[초] / 분자
15
16   Serial.print("조도 값: ");
17   Serial.println(lightValue);
18   Serial.print("진동 값: ");
19   Serial.println(vibrationValue);
20   Serial.print("모터 작동 시간: ");
21   Serial.println(motorDuration);
22   digitalWrite(motorPin, HIGH); // 모터 동작 시작
23   delay(motorDuration); // 모터 동작 시간만큼 대기
24   digitalWrite(motorPin, LOW); // 모터 중지
25
26   delay(60000 * (0.865 * lightValue + 3.4151981 * vibrationValue - 3459.0031465)); // 예상 증발 시간만큼 대기
27 }
28
29
```

위 파이썬 회귀분석코드에서 회귀분석한 결과  $Y = 0.865A + 3.415B - 3459.00...$  라는 식이 나온다. 14번째 줄에서 300(초)에서 위 증발시간(분)을 나누었을 때 대략적으로 펌프를 작동시키기 적절한 시간이 나타나기에 (예를 들어 예상 증발시간이 60분이라면 펌프를 5초동안 작동, 300분이라면 펌프를 1초동안 작동시킨다.) 위와 같은 코드를 작성하였고, 예상 증발시간(ms) 만큼 대기시키는 코드를 작성하였다.

- 마지막으로 로봇이 움직여 소금을 옮기도록 로봇의 모터 작동 코드를 작성하였다. (너무 길어 일부 캡처, 아래 코드가 반복되는 형태)

```
// 모터 드라이버에 펌프
const int out3Pin = 5; // 왼쪽 모터
const int out2Pin = 6; // 오른쪽 모터
const int out4Pin = 9; // 가운데 모터
const int out4Pin = 18; // 오른쪽 모터

void setup() {
  // 펌프 모터
  pinMode(out3Pin, OUTPUT);
  pinMode(out2Pin, OUTPUT);
  pinMode(out3Pin, OUTPUT);
  pinMode(out4Pin, OUTPUT);
  pinMode(out4Pin, OUTPUT);

  // 모터 작동 시간
  analogWrite(out3Pin, 0);
  analogWrite(out2Pin, 0);
  analogWrite(out3Pin, 0);
  analogWrite(out4Pin, 0);

  void loop() {
    // 1초 대기
    delay(1000);

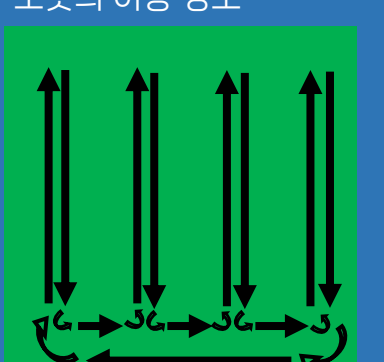
    // 왼쪽 모터로 펌프를 작동 (속도 150)
    analogWrite(out3Pin, 150);
    digitalWrite(out2Pin, LOW);
    digitalWrite(out3Pin, LOW);
    delay(4000); // 4초 대기

    // 오른쪽 모터로 펌프를 작동 (속도 150)
    analogWrite(out3Pin, 0);
    analogWrite(out2Pin, 150);
    digitalWrite(out3Pin, LOW);
    digitalWrite(out4Pin, LOW);
    delay(1000); // 1초 대기

    // 왼쪽 모터로 펌프를 작동 (속도 150)
    analogWrite(out3Pin, 150);
    digitalWrite(out2Pin, LOW);
    digitalWrite(out3Pin, LOW);
    delay(4000); // 4초 대기

    // 오른쪽 모터로 펌프를 작동 (속도 150)
    analogWrite(out3Pin, 0);
    analogWrite(out2Pin, 150);
    digitalWrite(out3Pin, LOW);
    digitalWrite(out4Pin, LOW);
    delay(1000); // 1초 대기
```

로봇의 이동 경로



염전에서 삼을 내린 채로 염전판 위에서 앞뒤로 4개의 레인을 왕복하는 코드이다. 중간중간 회전할 때 왼쪽과 오른쪽 바퀴의 전 후진을 다르게 하여 회전시키는 코드를 넣었다. 또한 모터 작동속도를 디지털값으로 작성하였기에 속도 조절도 가능하다.

### 기대 효과

염전에서 발생하는 비인도적인 일들을 해결할 수 있을 뿐만 아니라, 인부들을 쓰는 것 보다 효율적이고, 평범한 염전에서도 이러한 시스템이 도움이 될 수 있다. 단순히 인력을 대체할 수 있을 뿐만 아니라 회귀분석을 통해 효율적으로 바닷물을 끌어들이는 시스템을 마련하였기에 생산량을 더 증가시킬 수 있다고 생각된다.

실험에서 하지 못했던 온도와 습도 변인을 추가시키면 더 좋은 효과를 얻을 수 있을 것 같다.