

야간
전술
행군 자세
복구 지원 시스템



육군 제3야전군사령부 상병 장동원

대부분이 얘기하는 군대에서 힘든 훈련



유격 훈련



혹한기 훈련

그 훈련을 하면서 알게 되는 공통점



두 훈련 다 **행군**을 함



부상, 안전사고의 가능성이
매우 높은 훈련

행군

야간 행군의 경우
시야 확보, 기온 하강 등의 요인

안전사고 가능성 상승



야간 행군 간 발생하는 문제점

1. 군장 균형이 안 맞는 경우에 대한 파악이 어려움

군장 균형이 안 맞는 상황이 지속 될 경우,
무게가 불균형하게 가해지면서 부상, 안전사고의 원인이 됨

2. 지휘 계통이 행군 인원의 상태를 빠르게 파악하기 어려움

지휘관 또한 행군에 참여하고 있는 상황이기 때문에
행군 인원들의 상태를 빠르게 파악하기 어려움 => 안전사고 가능성 상승

3. 과학화된 다른 군 훈련과 다르게 훈련 결과에 대한 분석이 어려움

달성 여부로 평가되는 훈련인 만큼 훈련 결과에 대한 피드백이 어려움

이 문제를
Arduino로
해결할 수 있지 않을까?



그렇게 1인팀으로 나오게 되었습니다

초기 계획

1. 군장 균형이 안 맞는 경우에 대한 파악이 어려움

3축 자이로센서를 이용해 군장이 기울어져 있을 때 경고를 주거나 군장을 고쳐 준다면?

2. 지휘 계통이 행군 인원의 상태를 빠르게 파악하기 어려움

심박센서와 블루투스를 이용해 행군 인원의 상태를 지휘관에게 빠르게 알려 준다면?

3. 과학화된 다른 군 훈련과 다르게 훈련 결과에 대한 분석이 어려움

3축 자이로센서에서 수집된 데이터를 코스 분석 등을 통해 다음 훈련에 반영한다면?

계획 수정 소요

1. 군장 균형이 안 맞는 경우에 대한 파악이 어려움

~~3축 자이로센서~~를 이용해 군장이 기울어져 있을 때 경고를 주거나 군장을 고쳐 준다면?

2. 지휘 계통이 행군 인원의 상태를 빠르게 파악하기 어려움

~~심박센서와 블루투스~~를 이용해 행군 인원의 상태를 지휘관에게 빠르게 알려 준다면?

3. 과학화된 다른 군 훈련과 다르게 훈련 결과에 대한 분석이 어려움

~~3축 자이로센서에서 수집된 데이터~~를 코스 분석 등을 통해 다음 훈련에 반영한다면?

수정된 계획

1. 군장 균형이 안 맞는 경우에 대한 파악이 어려움

기울기 센서를 이용해 군장이 기울어져 있을 때 경고를 주거나 군장을 고쳐 준다면?

2. 지휘 계통이 행군 인원의 상태를 빠르게 파악하기 어려움

심박센서와 부저, LED를 이용해 행군 인원의 상태를 지휘관에게 빠르게 알려 준다면?

3. 과학화된 다른 군 훈련과 다르게 훈련 결과에 대한 분석이 어려움

심박센서에서 수집된 데이터를 코스 분석 등을 통해 다음 훈련에 반영한다면?

구현 과정



매직라이트
컵 모듈



프로세싱



심박 센서

군장 평형 상태 파악

```
#include<Servo.h>
Servo servo;

int servo_pin = 8;
int left_LED = 3;
int left_pin = 4;
int right_LED = 13;
int right_pin = 12;
int min_value = 544;
int max_value = 2400;
int now_degree = 90;

void setup() {
    pinMode(left_LED, OUTPUT);
    pinMode(right_LED, OUTPUT);
    pinMode(left_pin, INPUT);
    pinMode(right_pin, INPUT);

    servo.attach(servo_pin, min_value, max_value);
    servo.write(90);

    Serial.begin(9600);
}
```

```
void loop() {
    bool left_sensor = digitalRead(left_pin);
    bool right_sensor = digitalRead(right_pin);

    if(left_sensor == LOW && right_sensor == HIGH){
        now_degree = now_degree + 8;
        servo.write(now_degree);
        digitalWrite(left_LED, HIGH);
        digitalWrite(right_LED, LOW);
        Serial.println("left");
    }
    if(left_sensor == HIGH && right_sensor == LOW){
        now_degree = now_degree - 8;
        servo.write(now_degree);
        digitalWrite(left_LED, LOW);
        digitalWrite(right_LED, HIGH);
        Serial.println("right");
    }
    if(left_sensor == HIGH && right_sensor == HIGH){
        servo.write(now_degree);
        digitalWrite(left_LED, HIGH);
        digitalWrite(right_LED, HIGH);
        Serial.println("alright");
    }
    if(left_sensor == LOW && right_sensor == LOW){
        servo.write(now_degree + 15);
        servo.write(now_degree - 15);
        servo.write(now_degree);
        digitalWrite(left_LED, HIGH);
        digitalWrite(right_LED, HIGH);
        Serial.println("strange");
    }
    delay(300);
}
```

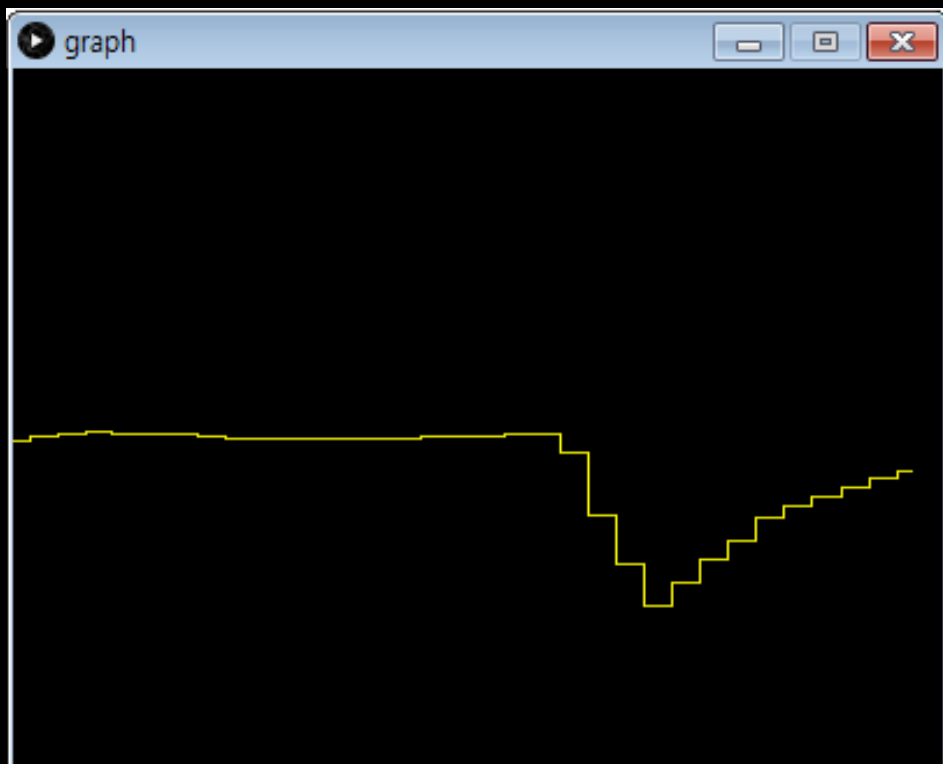
행군 인원 상태 파악

```
int raw_value = analogRead(A0)-900;
if(raw_value < minval){
    raw_value = 0;
}
double value = alpha * old_value + (1 - alpha) * raw_value;

Serial.println (value);
old_value = value;
```

```
if(raw_value == 0){
    analogWrite (redpin, 60);
    analogWrite (greenpin, 30);
    time_check = millis();
    digitalWrite(7, LOW);
}
else if(flag == true){
    if(value < 85 || value > 120){
        analogWrite (redpin, 100);
        analogWrite (greenpin, 0);
        time_now = millis();
        if(time_now - time_check > 10000){
            digitalWrite(7, HIGH);
            Serial.println("Danger!!!");
        }
    }
    else{
        analogWrite (redpin, 0);
        analogWrite (greenpin, 100);
        flag = false;
        time_check = millis();
        digitalWrite(7, LOW);
    }
}
else if(flag == false && value < 85 || value > 120){
    analogWrite (redpin, 100);
    analogWrite (greenpin, 0);
    time_check = millis();
    flag = true;
}
else{
    analogWrite (redpin, 0);
    analogWrite (greenpin, 100);
    time_check = millis();
    digitalWrite(7, LOW);
}
delay (200);
```

심박센서 그래프 그리기



```
import processing.serial.*;

Serial port;
int x_position = 1;
float value = 0;
float old_value = 0;

void setup () {
  size(400, 300);
  port = new Serial(this, Serial.list()[2], 9600);
  port.bufferUntil('\n');

  background(0);
}

void draw () {
  stroke(255, 255, 0);
  line(x_position, height - old_value, x_position, height - value);
  if(x_position >= width){
    x_position = 0;
    background(0);
  }
  else {
    x_position++;
  }
  old_value = value;
}

void serialEvent (Serial port) {
  String input = port.readStringUntil('\n');

  if (input != null) {
    input = trim(input);
    value = float(input);
    println(value);
    value = map(value, 0, 200, 0, height);
  }
}
```


시연 영상

한계점



저에게 시간과 예산이 조금만 더 있었더라면...

- **군장을 모터로 직접 움직이는 것은 한계가 있음**

모터 무게, 군장 용량 등의 문제로 불가능 (제어 시연을 위해서 이런 형태로 제작)

- **ON/OFF 센서(매직 라이트 컵) / 서보모터로는 세부 제어 불가능**

외부 전원이 없었기 때문에 스텝모터 사용 불가능

- **심박센서로 수집한 데이터만으로는 한계가 있음**

리얼타임 모듈, 3축 자이로 센서 등 좀 더 다양한 정보가 필요함

- **블루투스 센서, 3축 자이로 센서, 외부 전원이 있었다더라면...**

더 나아가

- 3축 자이로 센서를 이용한 행군간 자세 분석

자세가 무너지는 코스 구간을 분석해 향후 행군 훈련에 반영

- 블루투스 모듈을 이용한 3축 자이로 센서 데이터 전송 및 시각화

블루투스를 이용해 지휘관의 단말기에 데이터를 바로 보내준다면?

- 모터를 이용한 군장 평형 제어 방식 개선

군장을 들고 캘리브레이션 버튼을 누르면 3축자이로 센서 값과 삼각법을 이용해 평형을 맞추기 위해 군장 끈을 당겨야 하는 길이를 파악하고 모터로 그만큼의 길이를 당기게 함

감사합니다