

SMART Handle PROJECT

1 2 1 6 4 7 1 0

박 승 훈

1 2 1 6 4 7 1 1

송 준 혁

CONTENTS

- INTRODUCTION, SCHEDULING, PROJECT PURPOSE
- FLOW DIAGRAM
- PROJECT CODE
- SENSORS
- EOG DATA
- DEMO-VIDEO
- SUMMARY
- FUTURE ENHANCEMENT
- DEVELOPMENT DIRECTION
- SOURCE

INTRODUCTION



- 최근 사망원인 중 심정지로 인한 사망의 비율이 높아지는 추세
- 이 중 급성 심정지 발생률도 지속해서 증가하는 추세

급성 심정지 주요 발생장소

단위: %

가정	50~60
도로·고속도로	8~10
요양기관	5~7
구급차 안	4~6

자료:질병관리본부·소방청

- 급성 심정지 주요 발생 장소는 가정이 50% 이상 이긴 하나, 도로와 고속도로도 10% 정도로 무시할 수 없는 수준

INTRODUCTION

심정지, 골든타임 잡아라...최초목격자 응급처치때 생존 1.85배 ↑

송고시간 | 2019-02-13 14:31



(대구=연합뉴스) 김준범 기자 = 심정지 환자 최초 목격자가 119구급대 도착 전 심폐소생술(CPR)을 하면 환자 생존율이 1.85배 높아지는 것으로 나타났다.

13일 대구시소방안전본부에 따르면 지난해 심정지 환자 발생 1천160건을 분석한 결과 최초 목격자가 심폐소생술을 시행한 건수는 754건(65%)이다.

출처 : 연합뉴스

- 심정지에 관해 초기 대응이 매우 중요
- 급성 심정지 환자의 골든 타임 = 10분



운전 중 급성 심정지에 대처할 수 있는 기술이 필요

INTRODUCTION

안정 시 심박수(남성)

나이대	평균 범위(bpm)	총 범위(bpm)
1세 미만	115~137	102~155
1	107~122	95~137
2-3	96~112	85~124
4-5	84~100	74~112
6-8	76~92	66~105
9-11	70~86	61~97
12-15	66~83	57~97
16-19	61~78	52~92
20-39	61~76	52~89
40-59	61~77	52~90
60-79	60~75	50~91
80세 이상	61~78	51~94







안정 시 심박수(여성)

나이대	평균 범위(bpm)	총 범위(bpm)
1세 미만	118~137	104~156
1	110~125	95~139
2-3	98~114	88~125
4-5	87~104	76~117
6-8	79~94	69~106
9-11	76~91	66~103
12-15	70~87	60~99
16-19	69~85	58~99
20-39	66~82	57~95
40-59	64~79	56~92
60-79	64~78	56~92
80세 이상	64~77	56~93

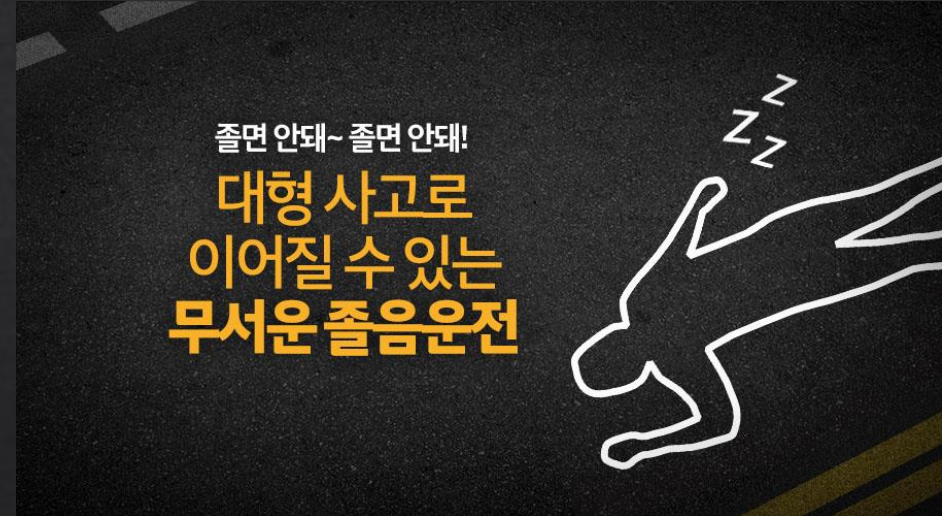
안정 시 심박수는 미국 질병 통제 예방센터에서 발행한 국립 건강 통계 보고서 (2011년, 41호)를 따릅니다.

<HEART RATE>

SCHEDULING

	10월					11월			
	5주차	6주차	7주차	8주차	9주차	10주차	11주차	12주차	13주차
PPG&PR 센서 작동 알고리 즘 및 코드									
경보&경보문 구 알고리즘 및 코드									
중간점검 및 데모									
EOG센서 데이터 취득 및 분석									
EOG센서 연 동 알고리즘 및 코드									
최종 실험 및 점검									

PROJECT PURPOSE



MAIN PURPOSE

1. 핸들에 부착된 PULSE SENSOR를 이용하여 운전자의 심박수 측정을 통해 건강 상태 파악 및 문제 발생시 대형사고 예방
2. EOG SENSOR를 이용한 졸음감지 데이터 분석

FLOW DIAGRAM



FLOW DIAGRAM

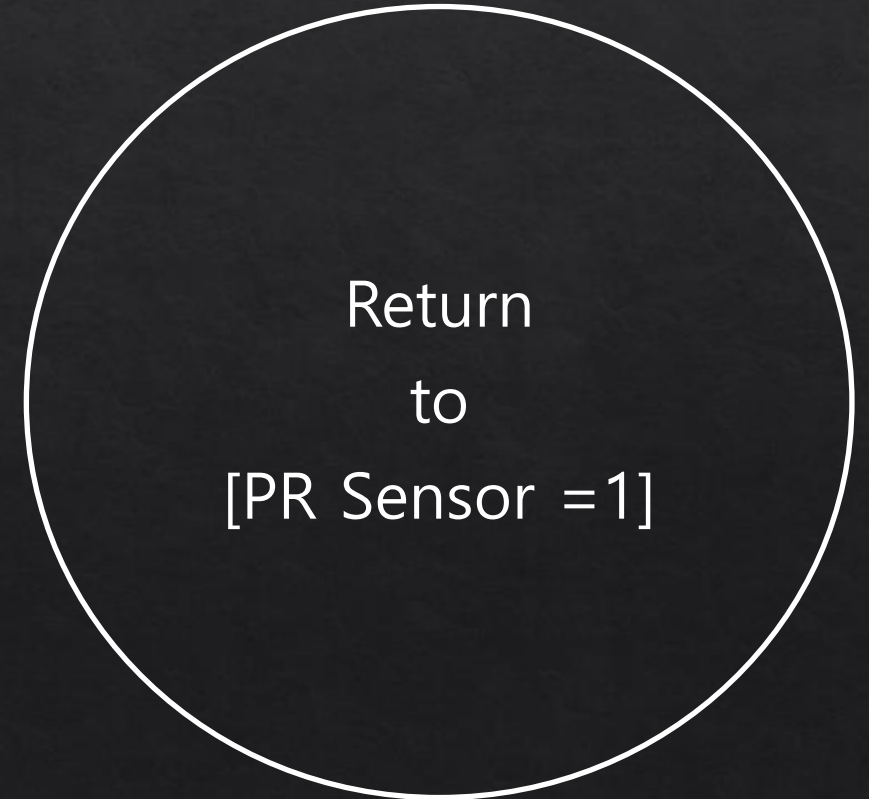
의식이 없는 경우



스스로 대처 할 수 있는 경우



다른 경우



PROJECT CODE

Pulse Sensor

```
void setup() {  
  
    Serial.begin(9600);  
  
    pulseSensor.analogInput(PulseWire);  
    pulseSensor.setThreshold(Threshold);  
    /*PPG 측정을 위한 함수 선언*/  
  
    if (pulseSensor.begin()) {  
        Serial.println("We created a pulseSensor Object !");  
    }  
    /*PPG에서 측정 시작시 시리얼모니터에 안내 */  
}  
  
/*Serial 통신 및 PPG 센서 초기화*/
```

- PPG 센서 측정을 위한 함수 선언
- BPM 측정 시 시리얼 모니터에 안내문 출력

PROJECT CODE

Pressure Sensor

```
void loop() {

    int sensorval1 = analogRead(A2);
    int sensorval2 = analogRead(A3);
    /*압력센서에서 측정되는 값을 analog값으로 받기 위한 변수 선언 */

    int sec=0;
    /*시간 측정을 위한 변수 선언*/

    if(sensorval1>=70 && sensorval2>=70){

        LCD(5,8,"System","On");

        delay(2000);
        lcd.clear();

        sensorval1=0;
        sensorval2=0; /*압력센서 값 초기화*/

        /*압력센서의 값이 양쪽다 70 이상일 때, 즉 핸들을 잡으면 시스템 동작 ( 아래 while 반복문으로 진입 )*/
```

- 각 PR 센서에 70이상의 값이 들어오면 'System On'
- 핸들을 잡으면 시스템 동작

PROJECT CODE

Detect Procedure

```
while(1){

    int myBPM = pulseSensor.getBeatsPerMinute();
    /*PPG에서 인식하는 값을 라이브러리의 함수를 이용하여 myBPM변수에 저장*/
    Serial.println(sec);

    delay(1000);
    sec++;
    /*1초를 delay와 sec++을 이용해 구현*/

    if((sensorval1<5&&sensorval2<5)&&sec>=10)
    {
        /*핸들을 잡지 않은 상태에서 10초 지속*/
        int count=10; /*대기시간 10초 설정*/
        sec=0;

        LCD(5,0,"Please","Hold your handle"); /*LCD에 경고문 출력*/
        delay(2000);

        while(1){

            int sensorval5 = analogRead(A2);
            int sensorval6 = analogRead(A3); /*핸들을 다시 잡을 때를 측정하기 위한 압력센서 변수 설정*/

            Serial.println("상황 파악 중");

            frontLCD(1,1,count,"call in 10 sec");

            Serial.println("Sensorvalue1 : " +String(sensorval5)  + "   Sensorvalue2 : " +String(sensorval6));
            Serial.println(sec);
            noTone(8);
            delay(1000); /*상황실 (시리얼 모니터) 에 현재 핸들의 압력센서 값 출력*/

            count--; /*10초의 대기시간 카운트*/
```

```
if(sensorval5>=20||sensorval6>=20)
{
    lcd.clear();
    break;
} /*1. 운전자가 핸들을 다시 잡을 경우 초기 상태로 진입*/

}

if(count==0)
{
    Serial.println("경보 경보");
    LCD(5,5,"Warning!!","Warning!!");
    for(int j=0;j<10;j++)
    {
        for(int h=0;h<3;h++)
        {
            tone(8,notes[h]);
        }
        delay(500);
    }
    lcd.clear();
    noTone(8);
} /*2. 경고 후 10초가 지나면 경보음 및 경보 문구 출력*/

else if(num1>=20&&num2<=10)
{
    LCD(5,8,"System","OFF");
    delay(1000);
    lcd.clear();

    break;
} /*3. 한쪽 압력센서 만 잡을시 시스템 종료*/
```

- 핸들을 놓을 시 상황파악을 위한 대기시간 10초 입력
- LCD에 경고문 출력
- 10초안에 잡지 못할 시 경보&문구 출력
- 핸들을 한 쪽으로 다시 잡을 시 경보 시스템 종료

PROJECT CODE

Detect Procedure

```
    if((sensorval3>=10||sensorval4>=10)&&(myBPM<=40||myBPM>=150))
/*1. 핸들을 잡고 PPG에서 심박 측정 시 이상 BPM감지 */
    {
        Serial.println("경보 경보");
        frontLCD(5,5,myBPM,"Warning!!");
/*경보 문구 출력 및 전달*/
        for(int i=0;i<3;i++)
        {
            tone(8,notes[i]);
        } /*경보음*/

        sec=0;
        delay(1000);
        lcd.clear();

    } |

    else if((sensorval3>=10||sensorval4>=10)&&(myBPM>40&&myBPM<150))
/*2. 핸들을 잡고 PPG에서 심박 측정 시 정상 BPM감지 */
    {
        Serial.println("이상 없음");
        frontLCD(5,5,myBPM,"Good!!");
        noTone(8);
        sec=0;
        delay(1000);
        lcd.clear();

    }
```

- 핸들을 다시 잡고 정상 BPM이 입력 될 때 LCD에 '이상 없음' 문구 출력

PROJECT CODE

Detect Procedure

```
if (pulseSensor.sawStartOfBeat()) {  
  /*PPG에 측정을 위해 손을 올릴 때 */  
  
  int sensorval3 = analogRead(A2);  
  int sensorval4 = analogRead(A3);  
  /*압력센서에 새로운 값을 받기 위한 변수 선언*/  
  
  Serial.println("BPM : " );  
  Serial.println(myBPM);  
  Serial.println(" ");  
  Serial.println("Sensorvalue1 : " +String(sensorval3)  +"   Sensorvalue2 : " +String(sensorval4));  
  delay(100);  
  /*상황실 (시리얼모니터)에 현재 측정대기 상태 전송*/  
}
```

- PR&PPG 센서 재 동작을 위한 코드

PROJECT CODE

EOG Procedure

```
#include <MsTimer2.h>
int analogPin0 = 0;
int analogPin1 = 1;
int notes[]={262,204,330};
int piezo1 = 6;
int piezo2 = 7;

void setup() {
    Serial.begin(115200);
    pinMode(piezo1, OUTPUT);
    pinMode(piezo2, OUTPUT);
    MsTimer2::set(1000, AnalogAD);
    MsTimer2::start();
    /*EOG 센서 측정을 위한 선언 1000ms=1s 주기*/
}

void loop()
{
    int check=AnalogAD();
    /*앞선 EOG센서의 반환값을 저장하는 check변수*/

    if(check=1)
    {
        for(int i=0;i<5;i++){
            for(int h=0;h<3;h++){
                tone(piezo1,notes[h]);
                tone(piezo2,notes[h]);
            }
        }
        delay(2000);
        noTone(piezo1);
        noTone(piezo2);
        check=0;
    }

    /*check=1 즉 졸음운전 발생시 경보를 울려 졸음운전시 발생할 수 있는 사고를 예방한다.*/
}
```

```
int AnalogAD() {
    int count=0;
    int real_count=0;

    while(1){

        int reading0 = analogRead(analogPin0);
        float Voltage0 = float(reading0)*5/1023;
        Serial.print(Voltage0,4);
        Serial.print(",");
        int reading1 = analogRead(analogPin1);
        float Voltage1 = float(reading1)*5/1023;
        Serial.println(Voltage1,4);
        /*EOG센서의 수직(Voltage0) , 수평(Voltage1)값을 측정한다. 졸음운전시 눈을 감는 이벤트 검출*/
        if(Voltage1>=1.7||Voltage0>=1.7)
        {
            count++;
            real_count=count/100;
        }

        /*취득한 데이터를 기반으로 졸음 운전시 발생하는 이벤트의 평균 값을 구해 count한다.*/
        if(real_count>=20)
        {
            return 1;
        }
    }
    /*졸음 운전시 발생한 값을 count하여 20이 넘었을 때 졸음운전이라 판단 정수 1을 반환한다.*/
}

/*1.EOG 센서의 측정값으로 졸음운전 판단 함수*/
```

SENSORS

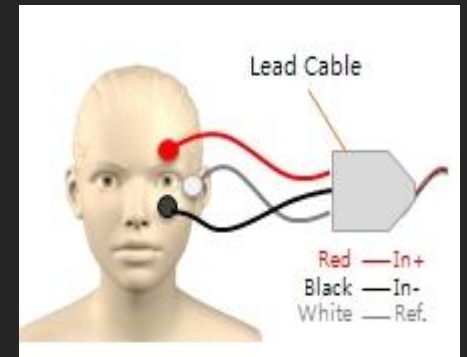
PPG SENSOR

- 혈액의 흐름은 심장의 고동에 맞춰 강해졌다 약해졌다 변화
- 손가락 끝, 귓볼 등의 미세혈관에 적외선을 보내고 반사되는 빛의 양을 감지해 변화하는 혈액의 흐름에 따라 변화하는 빛의 양을 측정하는 원리
- 센서 중앙부에 위치한 녹색 적외선 LED가 적외선을 방출하면 미세혈관에 의해 반사된 빛을 적외선 LED아래 위치한 DETECTOR가 측정



EOG SENSOR

- 눈 주위에 전극을 부착하여 양극을 띤 막이 전극 방향으로 움직일 때는 양 전위, 반대 방향으로 움직일 때는 음 전위가 형성되는 원리를 이용하여 안구 운동을 측정
- 두 눈의 움직임을 각각 기록하기 위해서는 수평과 수직전극을 각각의 눈 주위로 부착하지만, 일반적으로 측정하는 동향 안구운동의 경우에는 양쪽 관자뼈 전극으로 기록하게 된다.



EOG DATA

[Block Diagram]

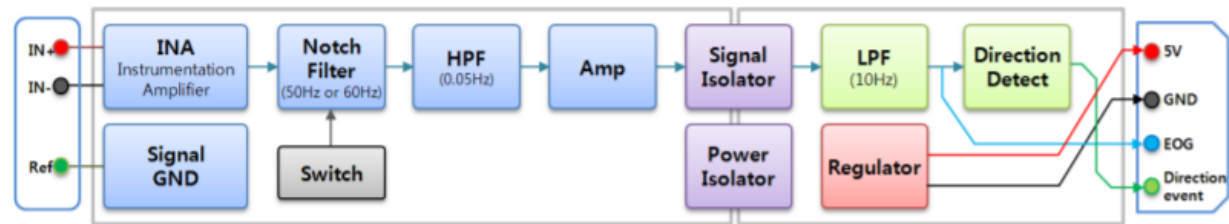
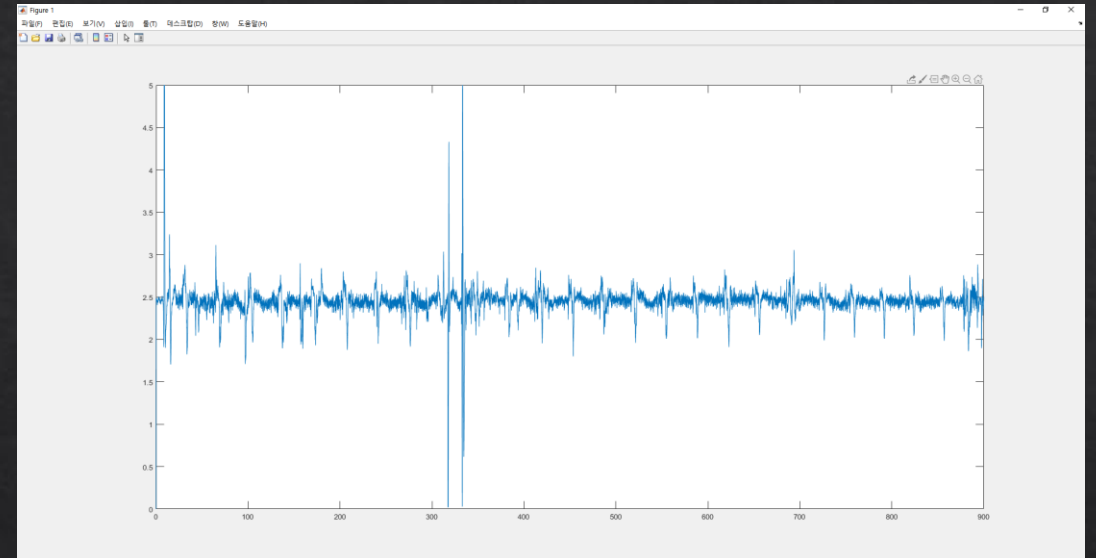


그림 2. PSL-iEOG2 block diagram



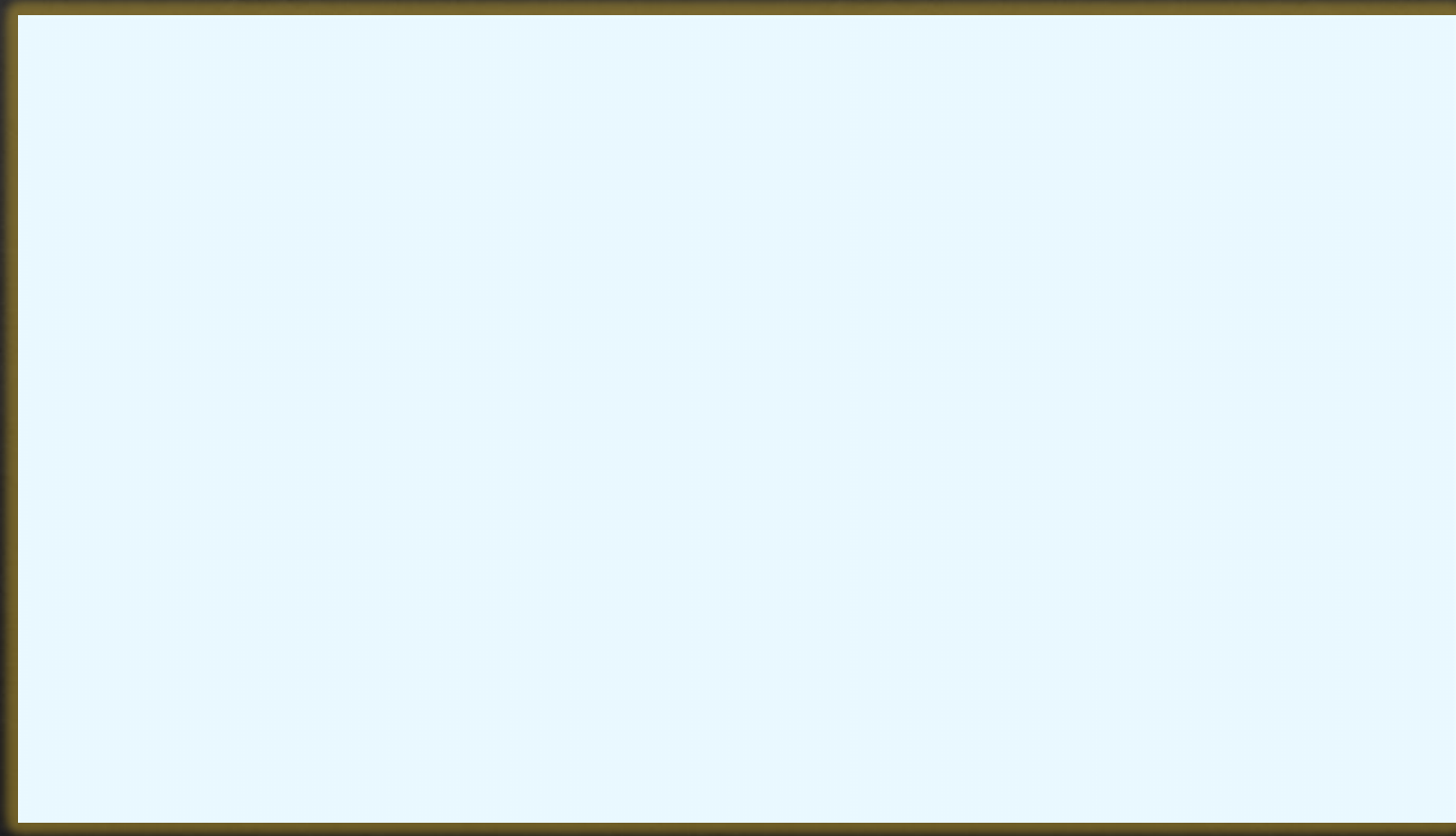
- EOG(Body) → PSL-iEOG2 → PSL-DAQ 데이터 획득 → EOG, Direction event.
- 취득한 RAW DATA를 HPF를 통해 필터링한 뒤 DATA 취득&분석

EOG DATA



- Serial Plotter를 사용하여 이벤트 발생하는 지점 확인
- MATLAB Plot으로 표현

FINAL VEDIO



SUMMARY

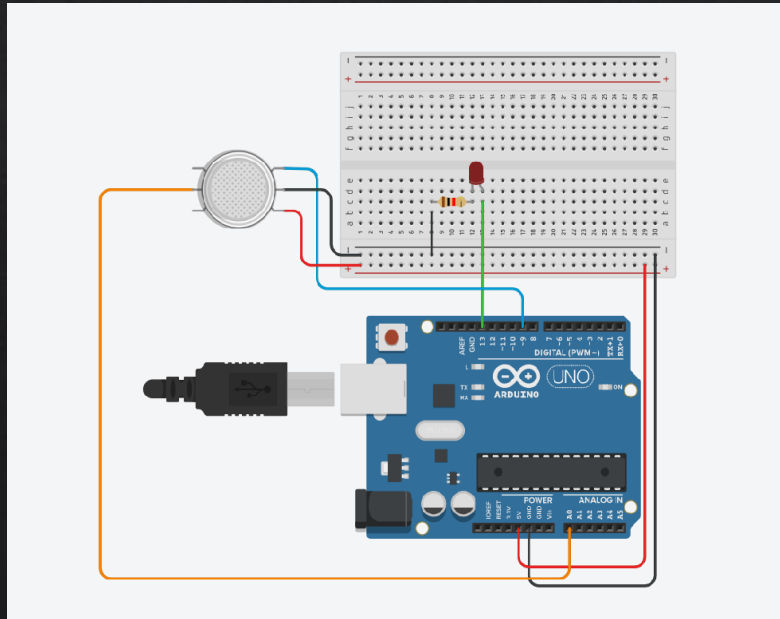
- PROJECT SUMMARY

- PPG 센서를 이용해 운전자 심박수 측정
- BPM 이상 감지시 경고음&경고문구 출력
- EOG 센서를 이용하여 데이터 취득 및 분석

FUTURE ENHANCEMENT

1. GAS SENSOR

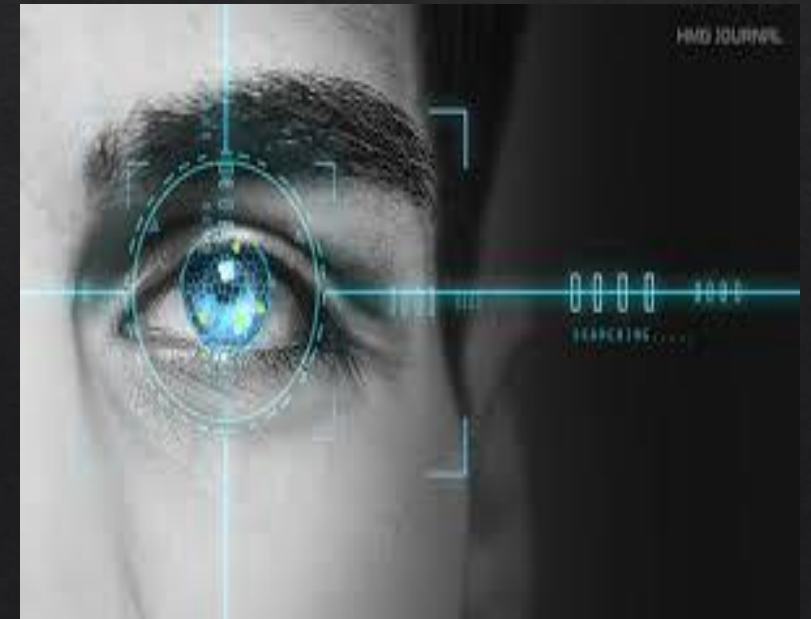
- 아두이노 가스센서를 활용하여 혈중 알코올 농도 측정
- 음주운전 방지



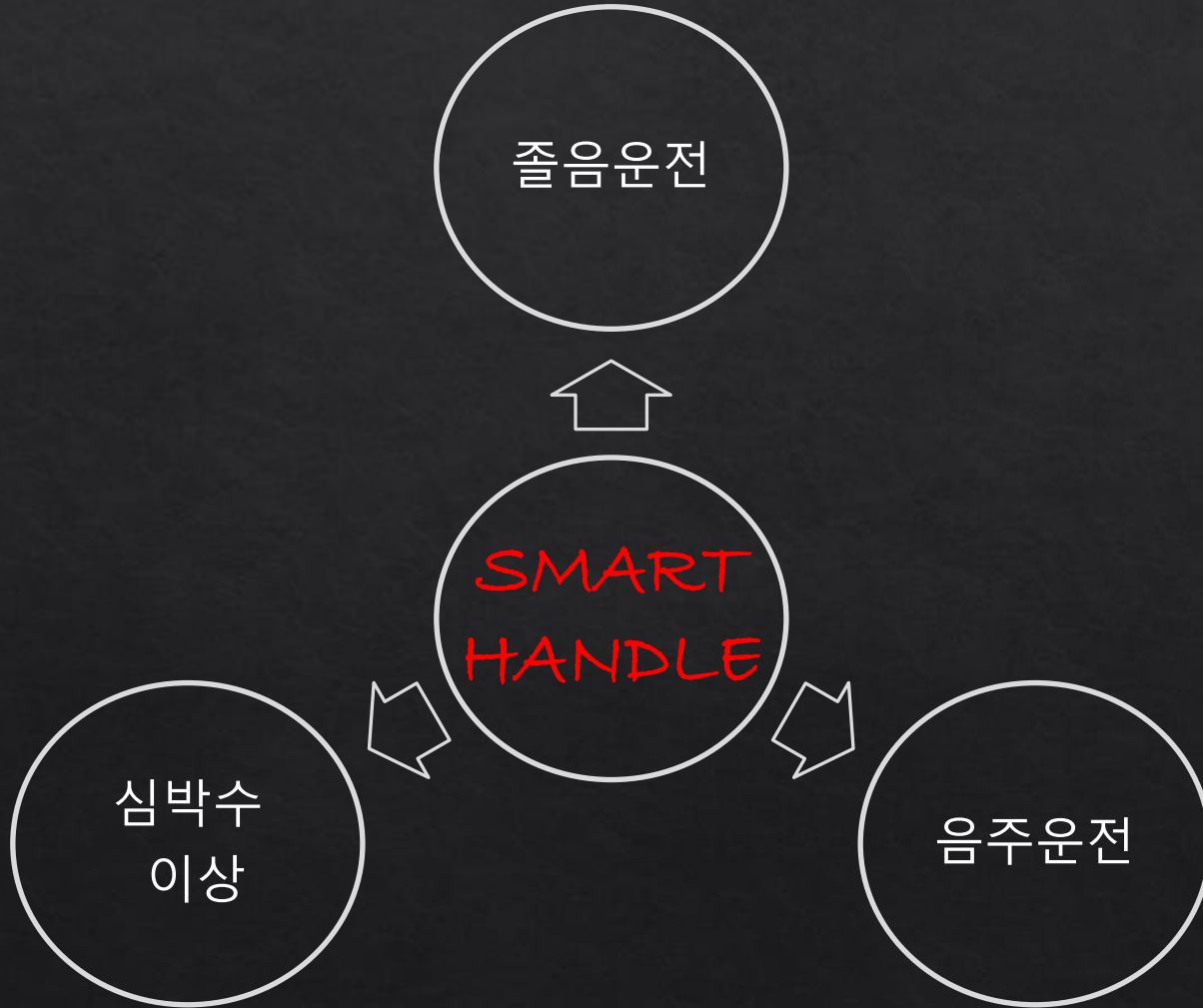
<회로도>

2. EYE TRACKING

- 카메라 모듈을 이용한 동공&눈꺼풀 움직임 감지
- 졸음운전 방지



DEVELOPMENT DIRECTION



SOURCES

- 1) 질병관리본부 소방청
- 2) <https://www.yna.co.kr/view/AKR20190213096200053>
- 3) <http://kormedi.com/1221906/%ED%98%88%EC%95%95%EA%B3%BC-%EC%8B%AC%EB%B0%95%EC%88%98%EC%9D%98-%EB%AC%98%ED%95%9C-%EA%B4%80%EA%B3%84-3%EA%B0%80%EC%A7%80/>
- 4) https://kids.hyundai.com/kidshyundai/safetyEnvironment/learnsafety/pictureCartoonDet.kids;jsessionid=TOqxmWnMzQOaqM7iL95smCSwozq16X8sMyvf8tphToT5NLnVy1oKIT4vp7wfjQNT.HWayKids_WAS02_servlet_engine1?cotnSn=2445
- 5) http://image.ec21.com/company/p/ph/phy/physiolab/upimg/PSL-EOG2_Feature_541x227.jpg
- 6) https://www.google.com/imgres?imgurl=https%3A%2F%2Fblog.hmgjournal.com%2Fimages_n%2Fcontents%2Fbiometrics_190618_02.jpg&imgrefurl=https%3A%2F%2Fnews.hmgjournal.com%2FTech%2F%25EC%2583%259D%25EC%25B2%25B4%25EC%259D%25B8%25EC%258B%259D%25EA%25B8%25B0%25EC%2588%25A0%25EC%2596%25B4%25EB%2594%2594%25EA%25B9%258C%25EC%25A7%2580%25EC%2599%2594%25EC%259D%2584%25EA%25B9%258C&docid=e0I5797duZ4k7M&tbnid=JwYdqY3VvrklwM%3A&vet=12ahUKewiVpr2UrofmAhXHP3AKHV7uBrg4ZBAzKEwwTHoECAEQVg..i&w=711&h=400&bih=750&biw=1536&q=%EB%8F%99%EA%B3%B5%EC%84%BC%EC%84%9C&ved=2ahUKewiVpr2UrofmAhXHP3AKHV7uBrg4ZBAzKEwwTHoECAEQVg&iact=mrc&uact=8

THANK YOU