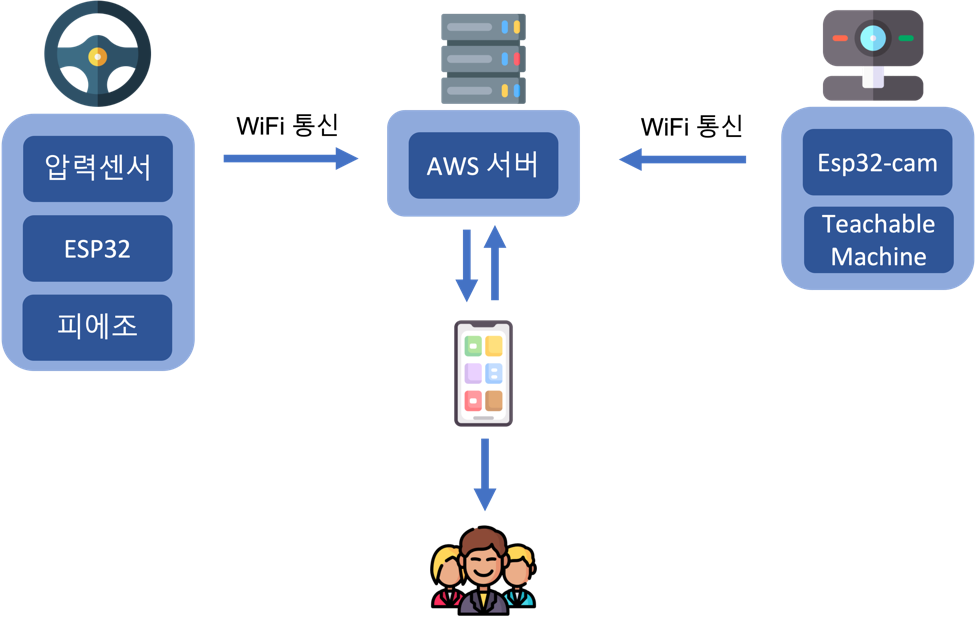
**IoT Eyes Checking  
- 압력센서와, AI 카메라를 활용한 졸음 운전 방지 시스템 -**

****

**IoT임베디드 오픈플랫폼   
Term Project 최종 보고서**

**컴퓨터 시스템과 201945023 김영호  
컴퓨터 시스템과 202145024 간영훈**

1. **제품 개요**
   1. **아이디어 배경**최근 화물연대의 안전 운임제 연장 주장과 관련된 파업으로 더 부각되듯, 안전 운전의 중요성과 졸음 운전의 위험성이 지속적으로 언급되어 왔습니다.  
      대한산업안전협회에 의하면 곧 다가올 여름 휴가철에 하루 평균 약 4건의 사고가 더 발생한다고 발표하였습니다. 동시에 장시간 운전이 많아지면서 집중력 저하, **졸음** 등으로 교통사고가 발생할 위험성이 높아진다는 것이 위험의 요인입니다.   
        
      (출처: [(사)대한산업안전협회 공식 블로그](https://blog.naver.com/safety1964/222772349382))
   2. **시스템 개발 필요성**
      1. 졸음운전의 원인  
         졸음운전은 장거리 운전을 할 때 운전자에게 고도의 집중력을 요하기 때문에, 그동안 쌓여온 신체적, 정식적 피로가 겹쳐 발생합니다.   
         위와 같이 수면부족으로 인한 피로, 장시간 운전으로 추가적으로 쌓인 피로, 차량 내산소부족, 식사로 인해 생긴 식곤증 등 여러 요인의 결합입니다.
      2. 구조적 문제  
         신체에서 발생하는 문제를 판단하고 경고해 줄 무언가가 없는 것이 문제입니다.  
         조수석에 앉은 동승인이 운전자의 상태를 연속적으로 지켜볼 수 없고 정확하게 졸음의 상태를 구분할 수 없기 때문에, 사고를 방지하지 못할 수 있습니다.  
         운전자가 혼자 운전하는 것이면 더욱이 위험성을 더 올라가게 됩니다.
   3. **졸음운전의 위험성**  
      졸음운전은 굉장히 많은 위험들을 불러들입니다.  
      첫째로 위험 요소 발견이 늦어진다는 점입니다.  
      둘째로 급브레이크로 인해 조작 위험성을 증가시킨다는 점입니다.  
      셋째로 발의 감각이 둔해져서 속도 조절이 어려워집니다.  
      넷째로 방향감각, 속도감이 없어져 사고의 위험을 대폭 높인다는 점입니다.  
      시속 100km/h속도로 1초만 졸아도 약 28m를 수면상태로 주행하게 됩니다. 4초를 졸면100m 이상을 운전자의 의지와 관련없게 주행하게 됩니다. 이에 앞차와 추돌사고가 발생할 확률이 커지며, 정체구간에서 사고가 발생하면 더 큰 인명피해가 발생할 수 있습니다. 치사율은 18.5%로 과속으로 인한 교통사고로 인한 치사율(7.8%)보다 약 2.4배 높은 것으로 나타났습니다.  
      위와 같은 위험이 지속되면 운전자의 사망사고로 이어질 수 있습니다.
   4. **IoT Eyes Checking의 개발 목표**
      1. 졸음운전 감지 기능  
         운전자가 졸음운전을 하고 있는지 감지하는 기능입니다.
      2. 졸음운전 경고 기능  
         운전자가 졸음운전 상태라고 시스템이 인식할 시 외부 신호로 경고하는 기능입니다.
      3. 졸음운전 신고 기능  
         외부 신호로 운전자에게 지속적으로 경고하여도, 운전자의 상태가 개선되지 않거나 응답하지 않을 경우 운전자의 상태를 신고하는 기능입니다.
   5. **문제점 해결 및 기대효과**
      1. 운전자의 졸음운전 예방과 안전사고 감소  
         졸음운전을 사전에 방지함으로써 안전사고를 예방하고, 감소시켜 사회적 비용을 감소시킬 수 있습니다.  
         이는 소방, 경찰, 구급, 산업 등 많은 사회적 비용이 절약됩니다.
      2. 졸음 운전으로 인한 사망사고 감소  
         사망사고 예방으로 인해 사회적인 문제로 지적되는 산업 인력 감소와 사회적 비용 감소에 효과적입니다.
2. **제품 설계**
   1. **제품 구성**
      1. **시스템 구성**  
         운전자의 졸음 상태를 알기 위해 핸들 부분과 엑셀 부분에 압력센서를 이용하여 압력 값을 받아오게 하였으며,

액셀을 밟은 상태에서 핸들의 센서 값이 없으면 피에조가 울리고 10초 후 자동으로 신고를 할 수 있으며 10초 전에 핸들에 있는 압력센서에 일정 값이 전달되면 카운트가 끝나고 피에조가 멈춥니다.

만약 10초가 지나면 서버에 “119”라는 키워드가 넘어가 자동으로 신고하도록 만들었습니다.

Esp32-cam을 사용해 운전자의 얼굴을 인식하여 졸음 여부를 확인합니다. 그러기 위해서는 Teachable Machine 모델을 만들어 cam에서 받아온 이미지를 통해 결과값을 받아와 결과값에 따라 운전자의 졸음 여부를 확인합니다.

만약 운전자가 졸고 있는 상태면 “Show”라는 키워드를 서버에 전송하여 자동 신고를 하도록 만들었습니다.

앱에서는 웹 뷰를 사용하였고 화면에는 보호자 등록 버튼과 긴급 신고 버튼이 존재하며 서버를 통한 자동신고가 들어오면 즉시 화면이 전환되도록 만들었습니다.

* + 1. **IoT Eyes Checking 시스템 설계 주요 구성품**  
       전자기기이(가) 표시된 사진

       자동 생성된 설명검은색이(가) 표시된 사진

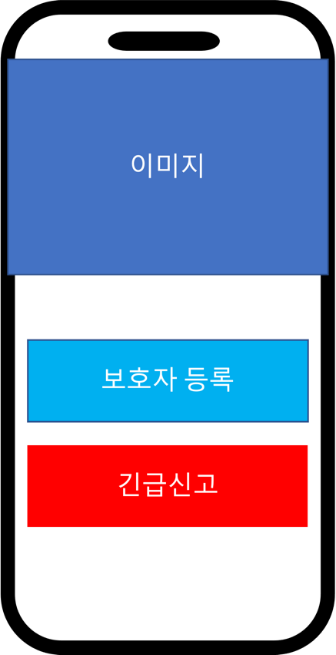
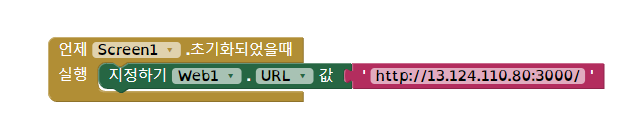
       자동 생성된 설명텍스트, 전자기기, 회로이(가) 표시된 사진

       자동 생성된 설명  
       ESP32: ESP8266의 상위 기종이며 ESP8266에서 부족했던 입출력 핀이 대폭 늘어났으며WiFi & Bluetooth을 지원하는 보드이다.

압력센서(Force Sensor): 기능 중에 핵심적인 기능을 담당하고 있으며 운전자의 압력 값을 가져올 때 사용한다.

부저센서(피에조):전기적 신호를 이용하여 소리를 내는 부품이다. 프로젝트에서는 필요조건에 따라 사용된다.

ESP32-CAM: ESP32보드와 같이 WiFi모듈을 지원하며 카메라를 따로 부착이 가능하다.

* + 1. **앱인벤터**
       - **UI디자인**
         1. 화면 구성  
            첫 번째 화면  
            애플리케이션을 실행하면 나오는 메인 화면입니다. 메인 화면에는 2가지의 버튼을 배치해 두었습니다.
         2. 두 번째 화면  
            메인 화면에서 “보호자 등록”버튼을 누르면 화면이 전환되며 보호자의 연락 정보를 등록할 수 있는 폼이 보여게 됩니다.  
            폼에 작성 후 등록을 누르면 연락처가 저장됩니다.
         3. 세 번째 화면  
            메인 화면에서 “긴급신고”버튼을 누르면 화면이 전환되며 3번째와 같은 화면이 보이게 됩니다. 3번째 화면은 주어진 조건에 따라 현재 어떤 상태인지 확인이 가능하게 텍스트가 변합니다.
       - **앱인벤터 컴포넌트**
         1. **메인화면**  
            Node.js를 AWS서버에 올려 사용하고 있기 때문에 모든 기능은 웹에서 처리하고 그 웹주소를 앱인벤터에서 웹뷰를 사용하여 보여주는 방식으로 구현하였습니다.
    2. **앱인벤터 서버처리**
       - **메인 페이지 처리**  
         서버를 실행 시킬 때 먼저 autoReport라는 전역 변수의 값이 119면 show.ejs라는 웹페이지를 보여주고 아니면 index.ejs라는 웹페이지를 보여줍니다  
         Ex) autoReport 값이 119가 맞다면 show.ejs를 보여준다.  
         텍스트이(가) 표시된 사진

         자동 생성된 설명
       - **보호자 등록 페이지**텍스트이(가) 표시된 사진

         자동 생성된 설명  
         메인 페이지에 있는 보호자 등록 버튼을 누르면 보호자 등록 페이지로 넘어가는 동시에 “/edit”로 들어오면 edit.ejs를 보여줍니다.  
           
         텍스트이(가) 표시된 사진

         자동 생성된 설명  
         Edit.ejs에서 보호자 연락 정보를 작성한 후 등록이 완료되면 POST방식으로 서버에 값을 전송하게 됩니다. 서버에서는 data라는 변수에 받아온 값을 저장하고 phone이라는 전역변수에 다시 저장하게 됩니다. 저장하고 난 후 다시 메인 페이지로 이동합니다.
       - **긴급 신고 페이지**텍스트이(가) 표시된 사진

         자동 생성된 설명  
         메인 페이지에 있는 긴급 신고버튼을 누르면 긴급 신고 페이지로 넘어가는 동시에 “/report”경로로 들어오면 report.ejs를 보여주며 phone이라는 변수 값을 같이 가지고 넘어갑니다.  
         텍스트이(가) 표시된 사진

         자동 생성된 설명  
         report.ejs에서 phone값을 if문으로 확인후 값이 있으면 웹페이지의 텍스트를 전화번호를 함께 보여주며 작성해준다.
       - **자동 신고시스템**텍스트이(가) 표시된 사진

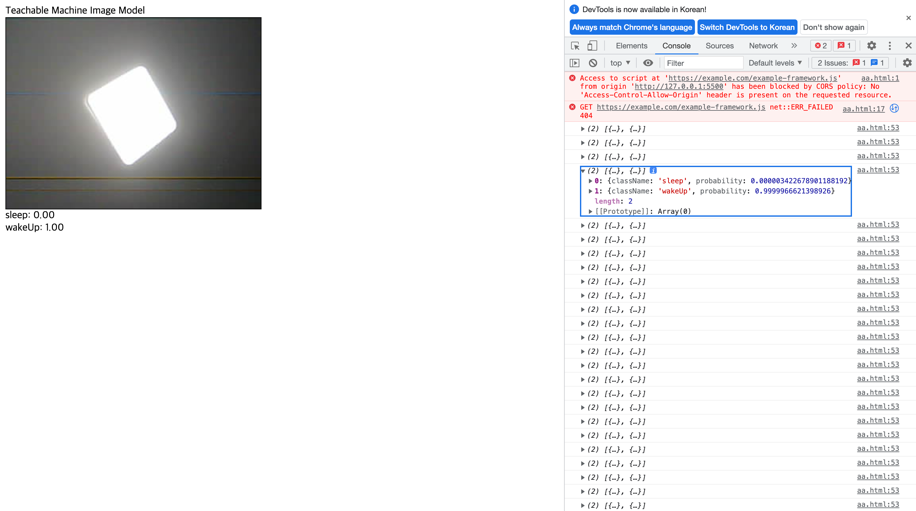
         자동 생성된 설명  
         아두이노에서 값을 보낼 URL을 지정하고 반복문을 통하여 10초 카운트르 시작합니다. 만약 10초 카운트가 끝날 경우 wifi\_send()라는 메소드를 이용하여 서버에 파라미터 형식으로 값을 보내 “119”라는 값을 보내게 됩니다
       - **확인 버튼(초기화)**  
           
           
         **텍스트이(가) 표시된 사진

         자동 생성된 설명** Report.ejs, show.ejs 페이지에 있는 확인버튼을 누르면 값은 받아오지 않지만 서버에 있는 전역변수를 초기화하여 기능을 초기화하고 if문을 사용해 조건에 맞으면 메인 페이지로 이동하게 끔 만들었습니다.

자동 신고 시스템이나 보호자의 전화번호를 저장하고 난 후에 전역변수에 값을 저장하다 보니 초기화를 하지 않을 경우 프로그램상 필요하지 않을 때 기능이 실행될 때가 있어 초기화가 꼭 필요 했습니다.

* + - * **ESP32-cam 이미지 처리 (Teachable Machine)**  
        텍스트이(가) 표시된 사진

        자동 생성된 설명  
        setInterval함수를 사용하여 0.5초마다 cam에서 받아온 주소를 사용하여 임의로 이미지를 가져와 웹페이지에 있는 이미지 태그에 계속해서 붙여준다.  
        텍스트이(가) 표시된 사진

        자동 생성된 설명  
        위에 코드를 보면 maxPredictions에서 가져온 값을 반복문을 이용하여 풀어줍니다.  
          
        콘솔에서 찍히는 값을 보면 cam에서 가져오는 이미지를 티쳐블 머신 모델과 비교하여 운전자의 졸음 여부를 확인 한 값을 가져 오는걸 알 수 있습니다.   
        여기서 저희는 운전자가 졸음 운전을 하는지 알아야 하기 때문에

Sleep값을 받아와 0.65값보다 크면 count를 증가 시키고 count가 5가 되면 ajax를 사용하여 서버에 “show”라는 키워드를 보냅니다.

반대로 0.65보다 작을 경우에는 count를 증가 하지만 count가 2면 서버에 “No”라는 키워드를 보내게 됩니다.

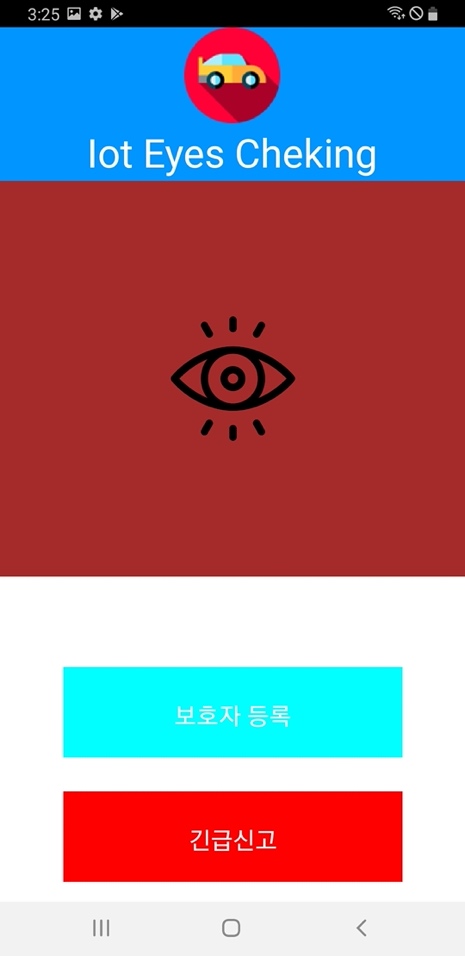
* + - * **“Show”&”No” 판단하기**텍스트, 화면, 스크린샷이(가) 표시된 사진

        자동 생성된 설명Ajax를 이용하여 서버에 값을 보내면 받아오는 값을 cam이라는 변수에 저장을 합니다.

텍스트, 화면이(가) 표시된 사진

자동 생성된 설명전역변수 cam에 저장한 값에 따라 return 해주는 문자를 다르게 해줍니다  
  
텍스트이(가) 표시된 사진

자동 생성된 설명  
아두이노에서 “/cam”에서 return된 문자를 받아옵니다. 만약 받아온 문자가 “Show”일 경우에는 state상태를 1로 변경하여 부저센서를 울리도록 하였습니다

* + - * **실제 화면 UI**텍스트이(가) 표시된 사진

        자동 생성된 설명지도이(가) 표시된 사진

        자동 생성된 설명지도이(가) 표시된 사진

        자동 생성된 설명

**아두이노 회로**

* + - * 회로 설계

|  |  |
| --- | --- |
|  | Esp32-cam 와이파이 블루투스 보드 Esp32-cam-mb 마이크로 Usb 직렬 포트 Ch340g Ov2640 2mp 카메라 모듈  듀얼 모드 Arduino - Buy Esp32-cam-mb Product on Alibaba.com |
| - 핸들 압력센서는 36번PIN에 연결한다. - 페달 압력센서는 39번PIN에 연결한다. - 피에조는 15번PIN에 연결한다. - 핸들 및 페달 핀과 연결되는 GND 사이 저항은 220 ohm(빨빨갈)을 사용한다.  ESP32와 ESP32-CAM보드 칩셋에 내장된 와이파이 모듈로 와이파이를 연결한다. | |

* + - * 핀맵

|  |  |  |  |
| --- | --- | --- | --- |
| Force Sensor\_1 | 36Pin | 220 ohm | GND |
| Force Sensor\_2 | 36Pin |
| Force Sensor\_3 | 39Pin |
| Buzzer | 15Pin | - |

* 1. **제품 설명**

|  |  |
| --- | --- |
|  |  |
| ESP32-본체 | ESP32-CAM |

* + 1. 실제제품
    2. 실제 적용제품

|  |  |
| --- | --- |
|  |  |
| 압력센서가 부착된 핸들 | 압력센서가 부착된 페달 |

* + 1. 동작 설명  
       1. 운전상태와 핸들의 압력으로 졸음 운전을 구분합니다>  
       페달을 밟고 있을 때 핸들에 압력이 없을 시 피에조가 10초간 울립니다.  
       운전자는 10초 안에 핸들을 바로 잡아야 하며 시간내 바로잡지 않을 시 서버에 운전자가 사고가 발생함 신호를 보내며 어플리케이션에 위치 정보가 표시됨과 함께 신고가 완료됩니다.

2. 핸들을 잡은 상태에서 발생하는 졸음운전을 캐치하기 위해 ESP32-CAM이 운전자의 얼굴을 계속적으로 스캔합니다.  
스캔한 영상은 서버로 보내지게 되며, 서버에서는 티쳐블머신에서 학습시킨 모델과 대조하며 결과값을 도출해내게 됩니다.  
대조값을 받은 서버에서는 수면 상태가 0.7 이상이 3-5초 이상 지속될 경우 Show신호를 보내, 위와 같이 10초간 피에조를 울리게 되며, 마찬가지로 일정 이상 압력을 주지 않은 이상 소음은 멈추지 않으며 10초가 지나면 사고임을 감지하고, 서버에 신호를 보내 사고가 발생했음을 알리고 어플리케이션에 위치정보가 표시됨과 함께 신고가 완료됩니다.

* 1. 유사 제품

|  |  |
| --- | --- |
|  | 특징 : 운전자의 눈동자를 적외선 카메라로 촬영하여 시선 정보를 이용해 운전자의 상태를 측정한다.  장점 : 운전자의 시선과 눈깜빡임을 감지해 졸음 운전을 방지할 수 있다.  단점 : 각도에 따라 적외선 카메라가 눈동자를 못할 수 있으며, 주변이 밝은 주간에는 적외선이 눈동자를 감지하지 못해 무용지물이다. 마찬가지로 야간이라도 차량 내 밝은 빛이 있다면 인식하기 힘들다. |
| 제이큐 사에서 출시된 야간 졸음방지기 | |

* 1. **사용 부품에 대한 DataSheet**
     1. Esp32 :

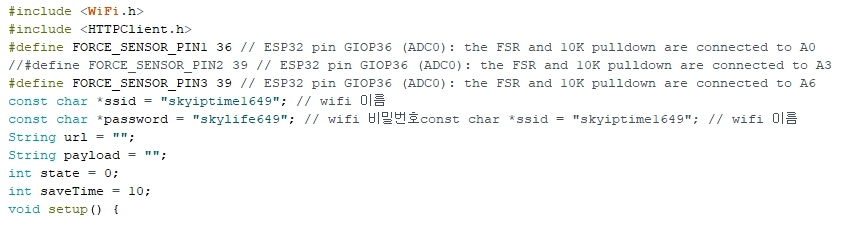
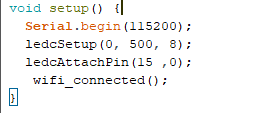
|  |
| --- |
|  |
| Power supply = 2.3V to 3.6V |
| ESP8266이 인기를 얻고나서 후속작으로 내놓은 상위 기종으로 WIFI뿐만 아니라 Bluetooth 4.2도 기본으로 지원하고 처리속도도 빨라 졌습니다. 무엇 보다도 ESP8266에서 부족했던 범용 입출력 핀이 대폭 늘어나서 활용범위가 커진 것이 장점입니다. |

* + 1. Esp32-Cam

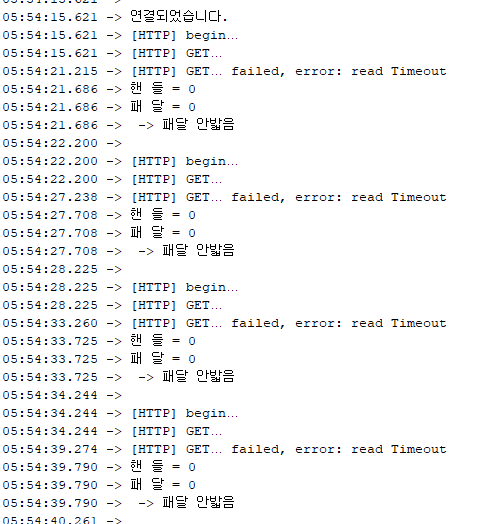
|  |
| --- |
|  |
| Power supply = 5V |
| ESPressif Systems라는 회사에서 개발하여 인기를 얻고 있는 ESP32 기본 모듈에 아예 카메라까지 내장하여 싸고 편리하게 활용할 수 있도록 판매되는 제품이다. 즉 아두이노보다 몇 배 빠른 프로세서 + WIFI와 블루투스 + Camera를 하나의 보드에 통합하고 여기에 SD카드슬롯까지 부착된 모듈이 기존 아두이노 모듈 하나의 가격과 비슷한 가격으로 판매되는 것이다. 이를 이용하면 아주 저렴하고 간단하게 CCTV녹화기, 동영상스트리밍장치, Remote CAR 동영상 송출장치 등을 개발할 수 있게 된다. |

* + 1. FSR 406

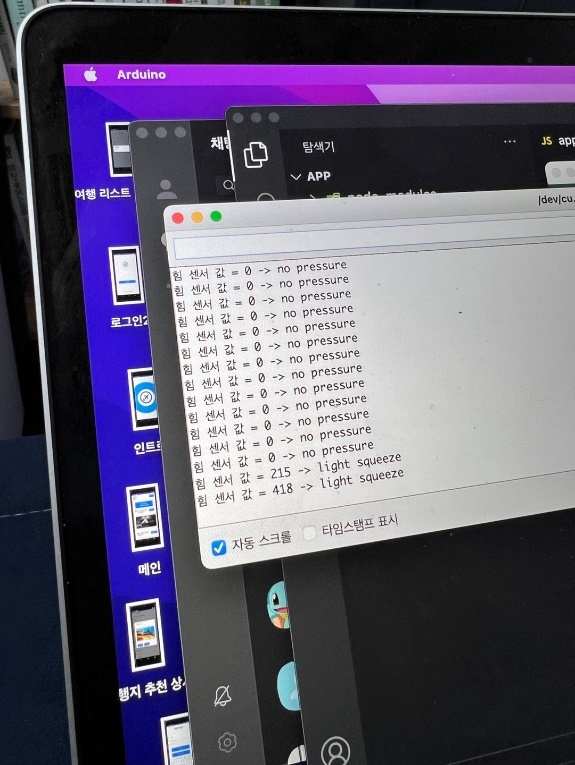
|  |  |
| --- | --- |
|  |  |
| Force Sensitivity Range 0.1 - 10.02 Newtons | |

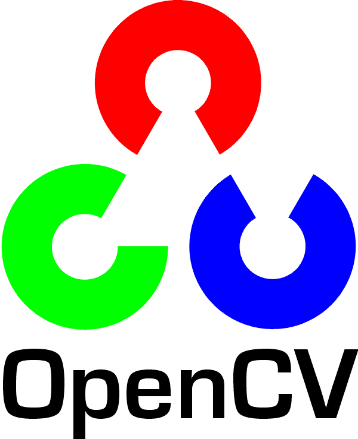
* 1. **코드**
     1. 아두이노 코드


* + 1. 아두이노 테스트 결과  
       

! - 테스트 결과 : 압력센서들이 서로 합선없이 기본값 0이 잘 출력되는 것을 확인할 수 있다. (합선되거나 GND포트가 죽을 시 압력센서 값이 굉장히 튀는 현상이 발견됨)

테스트 사진  
 

보완점  
1. 카메라 및 AI

카메라로 촬영된 영상을 AI머신에 입히기 위해서 티쳐블 머신을 활용하였지만, 이미지파일도 작고 모델도 완벽하지 않아서인지 구분을 제대로 하지 못하였습니다.

그러나 티쳐블머신이 아니라OpenCV를 사용해야 한다는 것을 뒤늦게 알아버린 터라 OpenCV를 활용하지 못하고 화면을 가린것과 가리지 않은것으로 대처하였습니다.

이는 차후 보완사항으로 눈의 좌표값을 구해 계속적으로 학습시켜 상태를 구분할 수   
있도록 하는 것이 최종 목표입니다.

2. 데이터 베이스 저장  
 서버에서 받아온 값을 데이터 베이스에 저장하여 기록하는 것을 중간 목표로

추가하였지만, 시간의 부족함과 기본 지식 부족으로 완성하지 못하였습니다.

이 또한 추후 보완 사항으로 지정하여 완성하는 것이 목표입니다.



**전시회 포스터**

