

아두이노 레포트

김호권

목 차

1. 아두이노란	... 3
2. 본론	... 3
2-1. 아두이노를 만들기 위한 차량	
2-2. LED	
2-2-1. 비상등	
2-2-2. 점등	
2-2-3. SOS	
2-3. 차량 움직이기	
2-4. LED랑 차량 움직임을 조종할 수 있는 웹 페이지 만들기	
2-5. Keras를 이용하여 경로 따라가기	
2-6. Yolov5를 이용하여 이미지 학습하고 검출하기	
2-7. 학습한 것들을 통합하여 트랙 돌기	
3. 견해	... 12

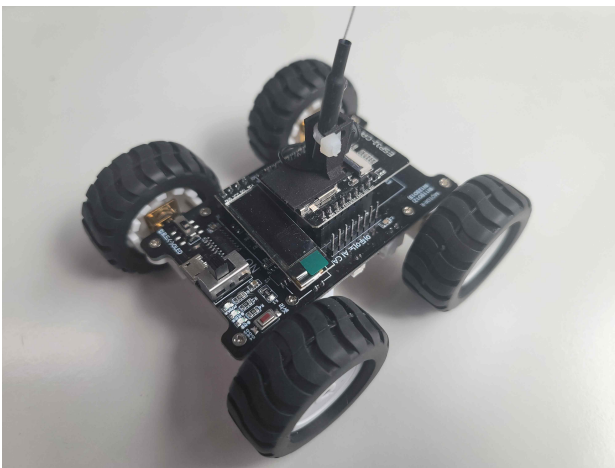
1. 아두이노란

아두이노는 오픈 소스 하드웨어와 소프트웨어 플랫폼으로, 마이크로컨트롤러를 사용하여 다양한 프로젝트를 구현할 수 있는 도구이다. 이를 통해 센서를 활용하여 환경을 감지하거나 모터를 제어하여 움직이는 장치를 만들거나, 빛을 조절하여 조명 시스템을 제어할 수 있다.

아두이노는 사용하기 쉬운 프로그래밍 환경을 통해 아두이노 보드를 프로그래밍하고 다양한 하드웨어 모듈을 활용하여 자신만의 창의적인 프로젝트를 실현할 수 있다.

2. 본론

2-1. 아두이노를 만들기 위한 차량



1. ESP32-CAM
2. PCB 바디
3. OLED
4. N20모터 4개

2-2. LED

2-2-1. 비상등

차량이므로 실제 차량 비상등에 맞춰 LED를 0.3초 마다 깜빡이게 코드를 작성하였다.

```
#define LED_BUILTIN 4

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  delay(3000);
}

void loop() {

  analogWrite(LED_BUILTIN, 25);
  delay(300);
  analogWrite(LED_BUILTIN, LOW);
  delay(300);
}
```



2-2-2. 점등

LED의 밝기가 1.2초 동안 점점 25만큼 올라가 최대 밝기까지 도달했다가 다시 1.2초 동안 밝기가 25씩 줄어들어 최종적으로 꺼지는 코드를 작성하였다.

```
#define LED_BUILTIN 4

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN,OUTPUT);
  delay(3000);
}

void loop() {
  a();
  b();
}
```

```
void a(){
  analogWrite(LED_BUILTIN,0);
  delay(109);
  analogWrite(LED_BUILTIN,25);
  delay(109);
  analogWrite(LED_BUILTIN,50);
  delay(109);
  analogWrite(LED_BUILTIN,75);
  delay(109);
  analogWrite(LED_BUILTIN,100);
  delay(109);
  analogWrite(LED_BUILTIN,125);
  delay(109);
  analogWrite(LED_BUILTIN,150);
  delay(109);
  analogWrite(LED_BUILTIN,175);
  delay(109);
  analogWrite(LED_BUILTIN,200);
  delay(109);
  analogWrite(LED_BUILTIN,225);
  delay(109);
  analogWrite(LED_BUILTIN,250);
  delay(109);
  analogWrite(LED_BUILTIN,255);
  delay(109);
}

void b(){
  analogWrite(LED_BUILTIN,250);
  delay(109);
  analogWrite(LED_BUILTIN,225);
  delay(109);
  analogWrite(LED_BUILTIN,200);
  delay(109);
  analogWrite(LED_BUILTIN,175);
  delay(109);
  analogWrite(LED_BUILTIN,150);
  delay(109);
  analogWrite(LED_BUILTIN,125);
  delay(109);
  analogWrite(LED_BUILTIN,100);
  delay(109);
  analogWrite(LED_BUILTIN,75);
  delay(109);
  analogWrite(LED_BUILTIN,50);
  delay(109);
  analogWrite(LED_BUILTIN,25);
  delay(109);
}
```

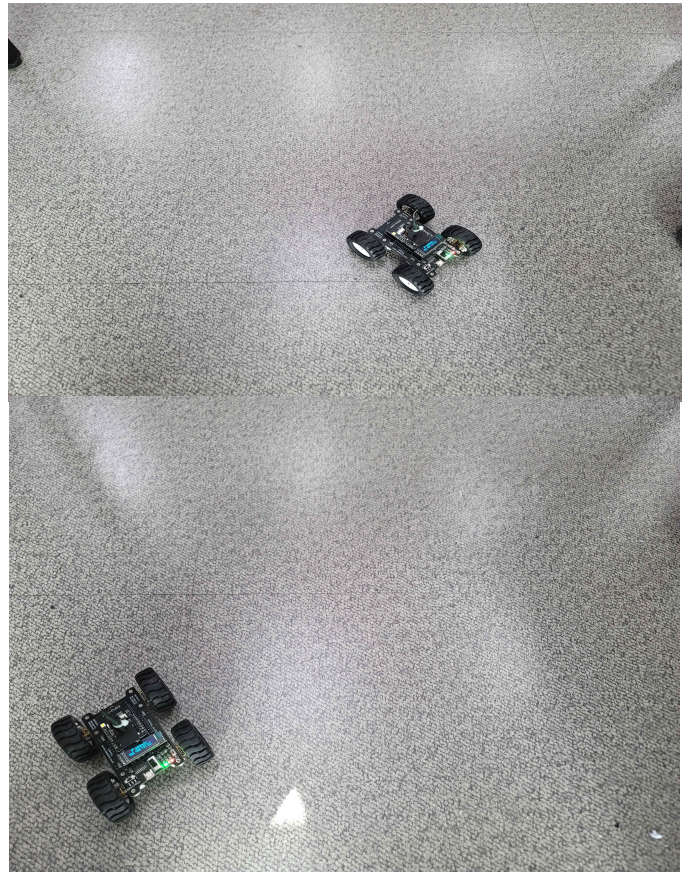
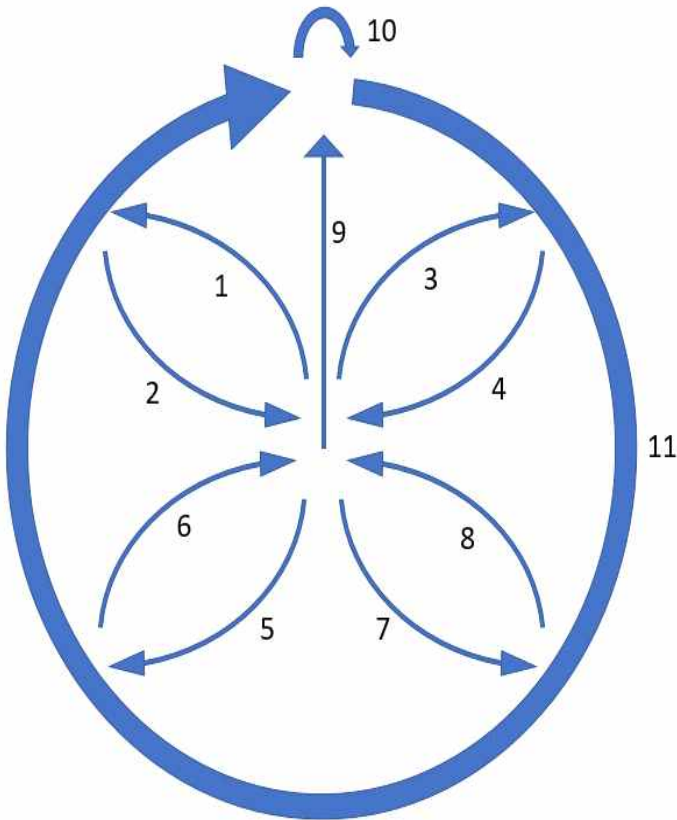
2-2-3. SOS

LED를 SOS의 모스부호형식으로 깜빡여 차량이 SOS를 하는 코드를 작성했다.

<pre>analogWrite(LED_BUILTIN, 5); delay(80); analogWrite(LED_BUILTIN, LOW); delay(90);</pre>	<pre>analogWrite(LED_BUILTIN, 5); delay(200); analogWrite(LED_BUILTIN, LOW); delay(90);</pre>	<pre>analogWrite(LED_BUILTIN, 5); delay(80); analogWrite(LED_BUILTIN, LOW); delay(90);</pre>
<pre>analogWrite(LED_BUILTIN, 5); delay(80); analogWrite(LED_BUILTIN, LOW); delay(90);</pre>	<pre>analogWrite(LED_BUILTIN, 5); delay(200); analogWrite(LED_BUILTIN, LOW); delay(90);</pre>	<pre>analogWrite(LED_BUILTIN, 5); delay(80); analogWrite(LED_BUILTIN, LOW); delay(90);</pre>
<pre>analogWrite(LED_BUILTIN, 5); delay(80); analogWrite(LED_BUILTIN, LOW); delay(200);</pre>	<pre>analogWrite(LED_BUILTIN, 5); delay(200); analogWrite(LED_BUILTIN, LOW); delay(110);</pre>	<pre>analogWrite(LED_BUILTIN, 5); delay(80); analogWrite(LED_BUILTIN, LOW); delay(800);</pre>
S	O	S

2-3. 차량 움직이기

차량에 아두이노 코드를 업로드 하여 정해진 대로 움직이는 코드를 작성했다.



차량 사진은 8번과 11번의 움직임을 하고 있다.

```
int TIME_INTERVAL = 3000;
int speed = 255;
String rData;

void loop() {
  if (Serial.available() > 0) {
    //speed = Serial.parseInt();
  }
  if (speed >= 0 && speed <= 255) {
    delay(1000);

    rData="left: ";
    display.clear();
    display.setTextAlignment(TEXT_ALIGN_LEFT);
    display.drawString(0,10,rData + String(speed));
    display.display();
    Serial.println(speed);
    car_left(speed);
    delay(TIME_INTERVAL*2);

    rData="back_left: ";
    display.clear();
    display.setTextAlignment(TEXT_ALIGN_LEFT);
    display.drawString(0,10,rData + String(speed));
    display.display();
    Serial.println(speed);
    car_back_left(speed);
    delay(TIME_INTERVAL*2);
```

1. Time interval은 3초, speed는 255를 기본 값으로 설정

2. 각 방향에 맞는 함수 작성

3. 순서대로 함수 발동하도록 코드 작성
(left → back_left → right → back_right → back_left → left → back_right → right → go → turn → circle)

4. 업로드 후 작동 되는지 확인

실행 결과 정상적으로 11번까지 진행하는 것을 확인 하였다.

2-4. LED랑 차량 움직임을 조종할 수 있는 웹 페이지 만들기

차량을 와이파이에 연결하고 JAVA를 아두이노와 연결하여 LED와 차량 움직임을 조종할 수 있고 카메라 화면을 스트리밍하는 웹페이지를 만드는 코드를 작성했다.

```
static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
        return res;
    }
}
```

```
void startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    config.server_port = 80;

    httpd_uri_t cmd_uri = {
        .uri       = "/action",
        .method    = HTTP_GET,
        .handler   = cmd_handler,
        .user_ctx  = NULL
    };
    httpd_uri_t stream_uri = {
        .uri       = "/stream",
        .method    = HTTP_GET,
        .handler   = stream_handler,
        .user_ctx  = NULL
    };
    if (httpd_start(&camera_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(camera_httpd, &cmd_uri);
    }
    config.server_port += 1;
    config.ctrl_port += 1;
    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(stream_httpd, &stream_uri);
    }
}
```

카메라 스트리밍을 처리하는 아두이노 코드

```
function led_start(num){
    let url = "built_in/" + num;

    $.ajax({
        "url": url,
        "method": "GET",
        "success" : function(result){
            alert(result);
        }
    });
}

function move(move_text){
    let url = "move/" + move_text;

    $.ajax({
        "url": url,
        "method": "GET",
        "success" : function(result){
            alert(result);
        }
    });
}
```

```
@GetMapping("/motor/{direct}")
@ResponseBody
public String motor(@PathVariable("direct") String direct) {
    String result = "";
    String url = "/motor/" + direct;
    String method = "GET";

    HttpURLConnection conn = getHttpURLConnection(url, method);

    result = getHttpResponse(conn);

    return result;
}

@GetMapping("/led/led_start/{num}")
@ResponseBody
public String led_start(@PathVariable int num) {
    String result = "";

    String url = "/led/led_start/" + num;
    String method = "GET";
    HttpURLConnection conn = getHttpURLConnection(url, method);

    result = getHttpResponse(conn);

    return result;
}
```

STS에서 차량 조종 웹페이지 기능 중 LED와 차량 조종 함수와 컨트롤 코드

RC CAR



IP Address

LED 깜빡이기

SOS

비상
등

점등

차량 조작

좌전
진

전진

우전
진

좌회
전

정지

우회
전

좌후
진

후진

우후
진

메세지 칸

만들어진 웹페이지

2-5. Keras를 이용하여 경로 따라가기

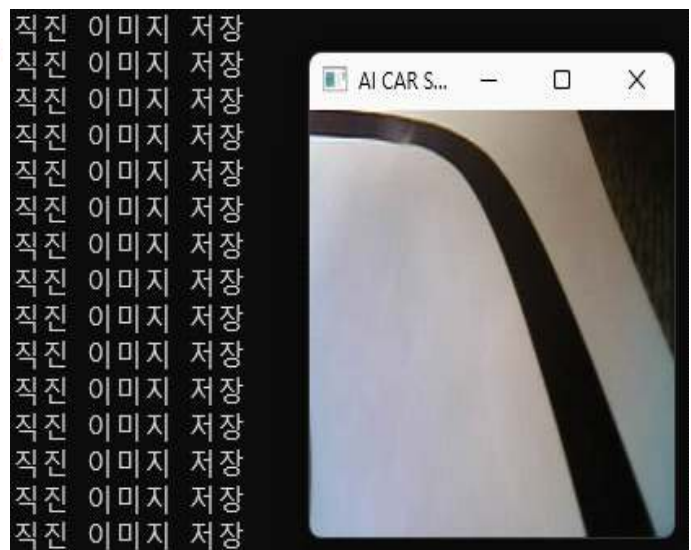
차량을 와이파이에 연결하고 경로를 따라 이미지 저장을 한 후에 티처블머신으로 학습 후 Keras를 이용하여 경로를 자동으로 따라가는 코드를 작성했다.

```
#자동차 조종
if key in input:
    urlopen('http://' + ip + "/action?go="+input[key])
    car_state = input[key]
    print(input[key])

#주행 이미지 저장
file_name = uuid.uuid1().hex+'.png'

if car_state == 'forward':
    print("직진 이미지 저장")
    cv2.imwrite(f'01_go/go_'+file_name,img)
elif car_state == 'left':
    print("왼쪽 이미지 저장")
    cv2.imwrite(f'02_left/left_'+file_name,img)
elif car_state == 'right':
    print("오른쪽 이미지 저장")
    cv2.imwrite(f'03_right/right_'+file_name,img)

if key == ord('q'): break
```



이미지 저장 코드와 콘솔에 나오는 이미지 저장 완료 메세지

```

model = load_model(r"D:\workspaces\arduino\keras_model.h5", compile=False)
class_names = open(r"D:\workspaces\arduino\labels.txt", "r").readlines()

def image_process_thread():
    global img
    global image_flag
    while True:
        if image_flag == 1:
            img = np.asarray(img, dtype=np.float32).reshape(1, 224, 224, 3)
            img = (img / 127.5) - 1

            # Predict the model
            prediction = model.predict(img)
            index = np.argmax(prediction)
            class_name = class_names[index]
            confidence_score = prediction[0][index]
            percent = int(str(np.round(confidence_score * 100))[:-2])

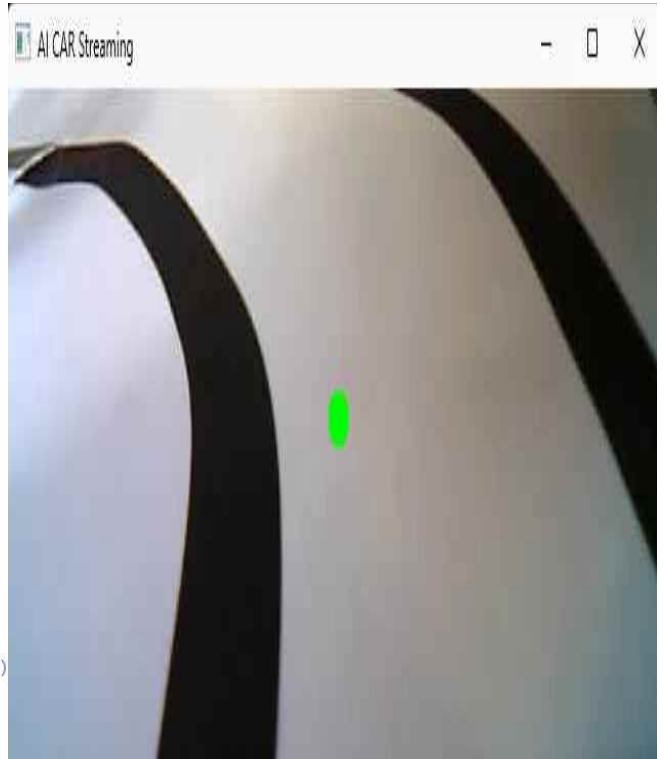
            if "go" in class_name[2:] and percent >= 95:
                print("직진:", str(np.round(confidence_score * 100))[:-2], "%")
                urlopen('http://' + ip + "/action?go=forward")

            elif "left" in class_name[2:] and percent >= 85:
                print("왼쪽:", str(np.round(confidence_score * 100))[:-2], "%")
                urlopen('http://' + ip + "/action?go=left")

            elif "right" in class_name[2:] and percent >= 85:
                print("오른쪽:", str(np.round(confidence_score * 100))[:-2], "%")
                urlopen('http://' + ip + "/action?go=right")

            image_flag = 0

```



티처블머신으로 학습한 모델을 이용한 자율주행 코드와 자율주행 확인

2-6. YOLOv5를 이용하여 이미지 학습하고 검출하기

차량을 와이파이에 연결하고 표지판을 이용하여 이미지 저장을 한 후에 학습 후 yoloV5를 이용하여 이미지를 검출하고 이미지에 따라 행동하는 코드를 작성했다.

```

if os.path.isdir('images') is False:
    os.mkdir("images")

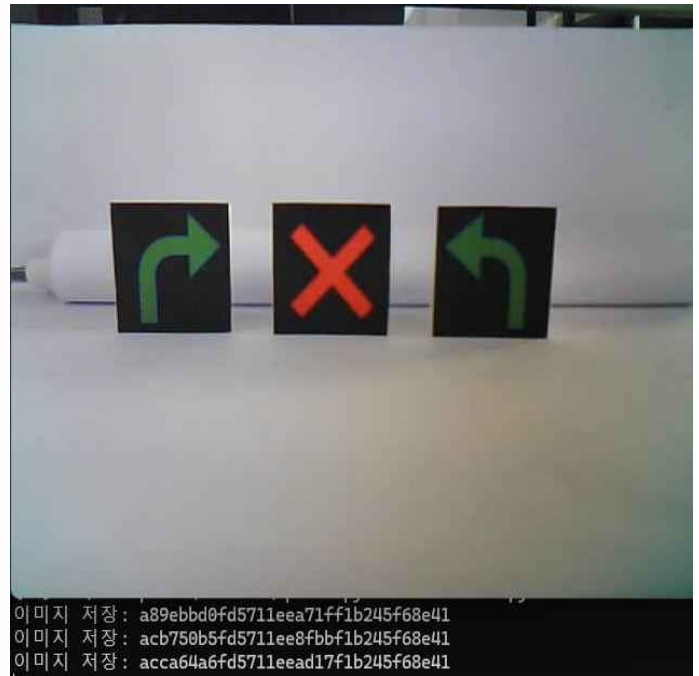
image_uuid = uuid.uuid1().hex
while True:
    buffer += stream.read(4096)
    head = buffer.find(b'\xff\xd8')
    end = buffer.find(b'\xff\xd9')

    try:
        if head > -1 and end > -1:
            jpg = buffer[head:end+2]
            buffer = buffer[end+2:]
            img = cv2.imdecode(np.frombuffer(jpg, dtype=np.uint8), cv2.IMREAD_UNCHANGED)

            cv2.imshow("AI CAR Streaming", img)

            key = cv2.waitKey(1)
            if key == ord('q'):
                break
            elif key == ord('s'):
                print("이미지 저장:", image_uuid)
                cv2.imwrite(f'images/image_{image_uuid}.png', img)
                image_uuid = uuid.uuid1().hex

```



이미지 저장 코드와 이미지 저장하고 있는 장면



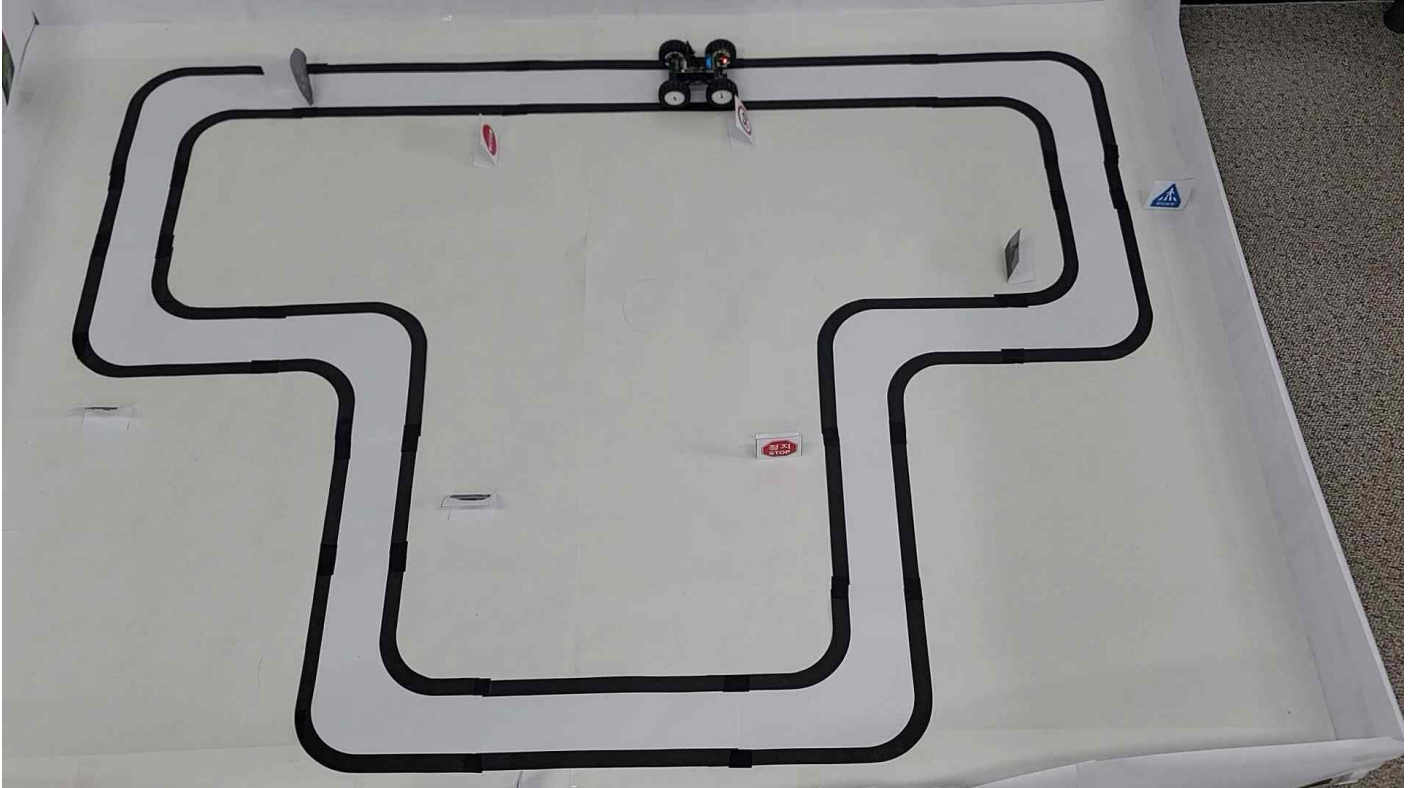
학습하여 검출된 화면



이미지 rock를 감지하고 멈춘 화면

2-7. 학습한 것들을 통합하여 트랙 돌기

이전에 학습한 트랙과 이미지를 종합해서 트랙을 도는 코드를 작성하여 실행하였다.



직접 만든 트랙에 학습한 표지판을 세워두고 실행시키는 장면

3. 견해

수업의 일환으로 아두이노에 대해 이해하고 그 일환으로 아두이노를 이용하여 버튼을 누르면 LED가 작동하고 연결된 차량이 움직이고 그 차량에 있는 카메라로 화면을 스트리밍 하는 웹페이지를 제작했다.

그리고 최종프로젝트에서 아두이노를 이용하여 직접 만든 트랙과 이미지를 학습하고 실제로 이미지 검출 및 동작을 진행하였다.

아두이노는 전자 제품을 개발하고 프로그래밍하는 데 매우 유용한 도구로 널리 사용됩니다. 사용하기 쉽고 학습하기 쉽다는 장점으로 처음 사용해보는 나도 이해하기 쉽고 재밌게 개발했으며 내가 생각한 대로 움직이는 것을 보면서 하나의 기쁨도 느낄 수 있었다.

이런 아두이노를 사용하는 실제 사례로는 스마트 홈 시스템, 로봇 제어, 사물 인터넷 장치 등 많은 곳에서 사용하고 있다.

그러나 아두이노는 제한적인 성능을 가지고 있기에 대규모 프로젝트에는 적합하지 않으며 전문적으로 응용하려는 분야에서는 다른 특화된 플랫폼을 사용하는 것이 바람직하다 할 수 있다.