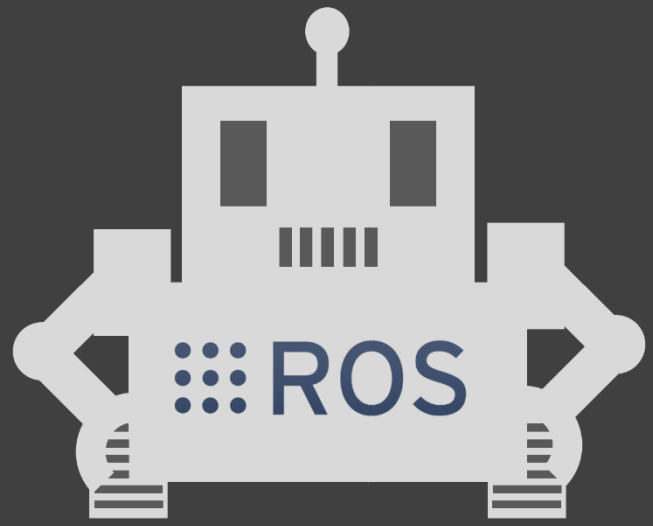
허스키 렌즈

Chapter 2. 허스키 렌즈 아두이노 연결

구선생 로보틱스



Update 20230909

강의 자료 다운로드

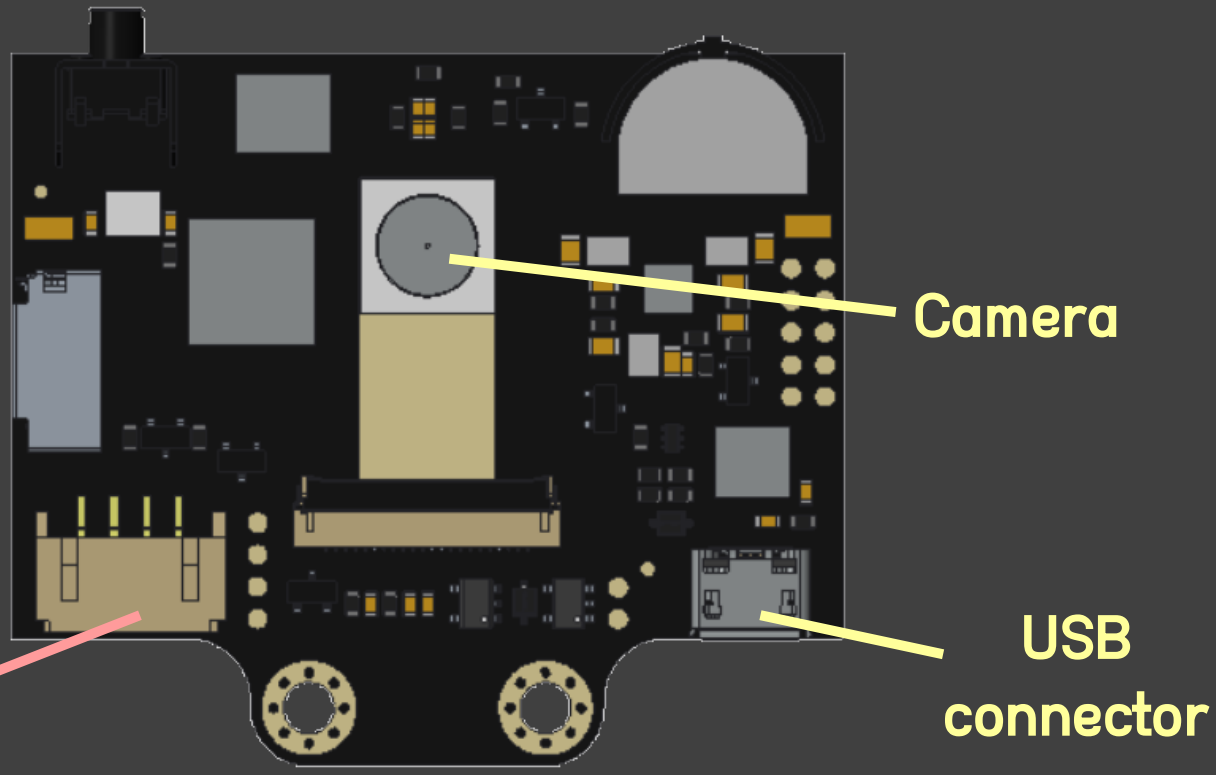
허스키 렌즈 강의자료

https://github.com/DoveSensei/HaskyLensNote

허스키 렌즈의 기능

기능	설명
얼굴 인식	얼굴 윤곽을 감지하고 학습된 얼굴을 인식 및 추적 하는 기능
객체 추적	지정된 개체를 학습하고 추적하는 기능. 하나의 개체만 추적 가능
객체 인식	사물이 무엇인지 인식하고 추적하는 기능
라인 추적	지정된 색상 선을 추적하고 경로를 예측하는 기능
색상 인식	지정된 색상을 학습, 인식 및 추적하는 기능
태그 인식	태그를 감지하고 지정된 태그를 학습, 인식, 추적하는 기능
객체 분류	다양한 물체의 여러 사진을 학습한 다음 내장된 기계 학습 알고리즘을 사용하여 학습하는 기능

허스키 렌즈의 기능



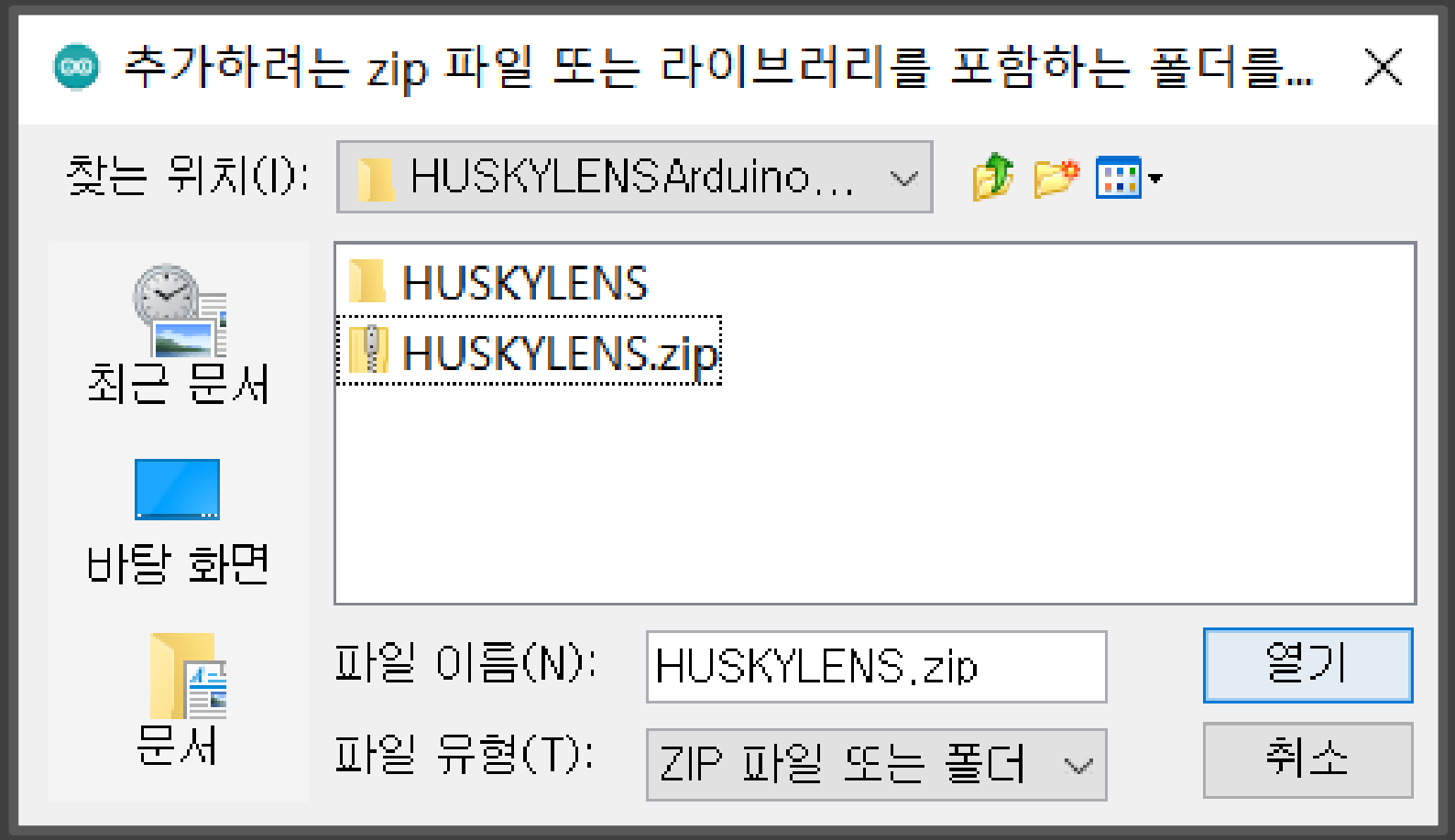
허스키 렌즈의 UART/I2C 를 이용하면
아두이노와 함께 사용할 수 있다

아두이노 IDE 설치

아두이노 IDE 1.8.19
https://drive.google.com/file/d/1rPPIpaOf8Q8_wu0cSPTsRPE62SCoYmQe/view

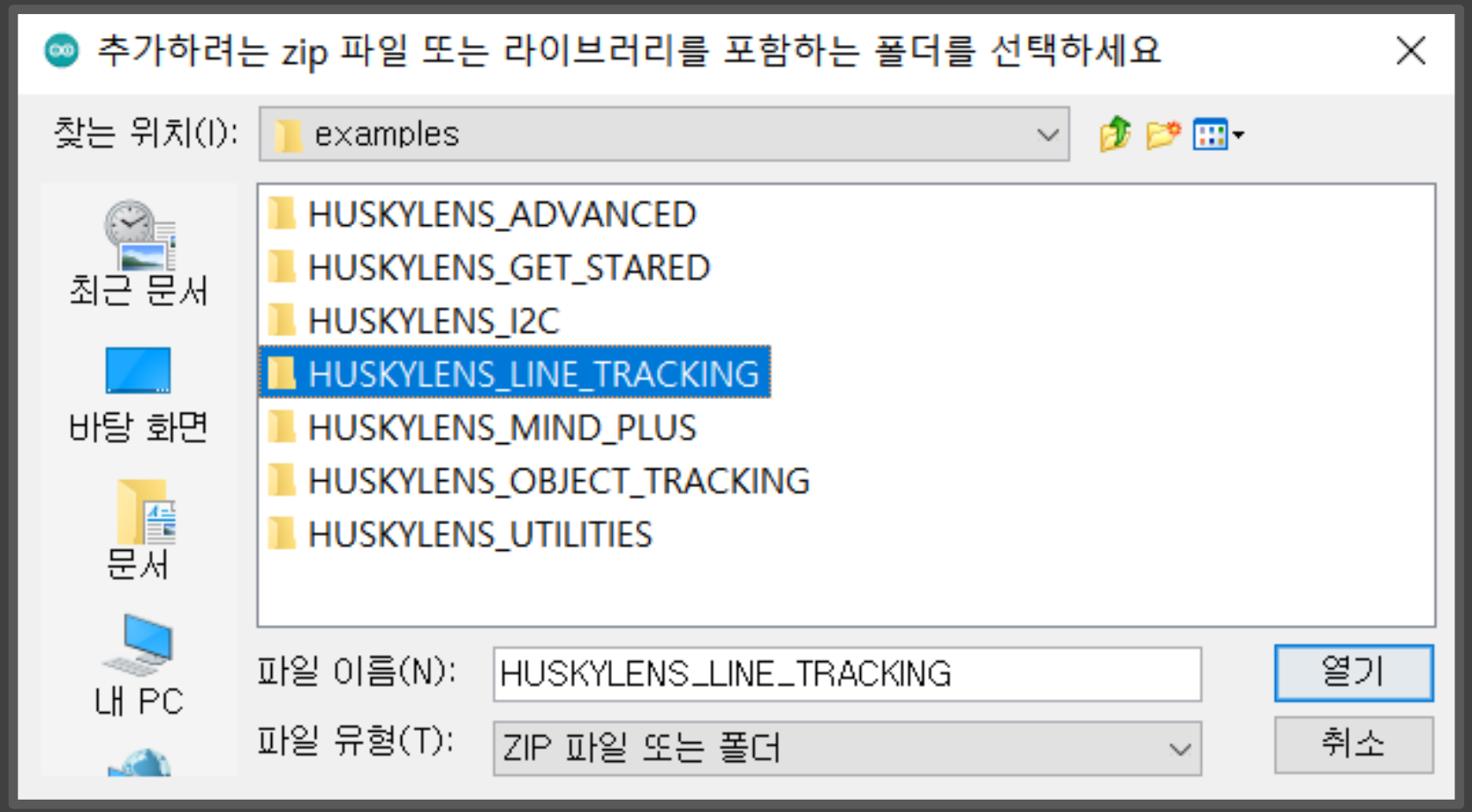
허스키 렌즈 라이브러리 설치 1

1) 스케치 -> 라이브러리 포함하기 -> .ZIP 라이브러리 추가
2) HUSKYLENS.zip 선택 후 열기



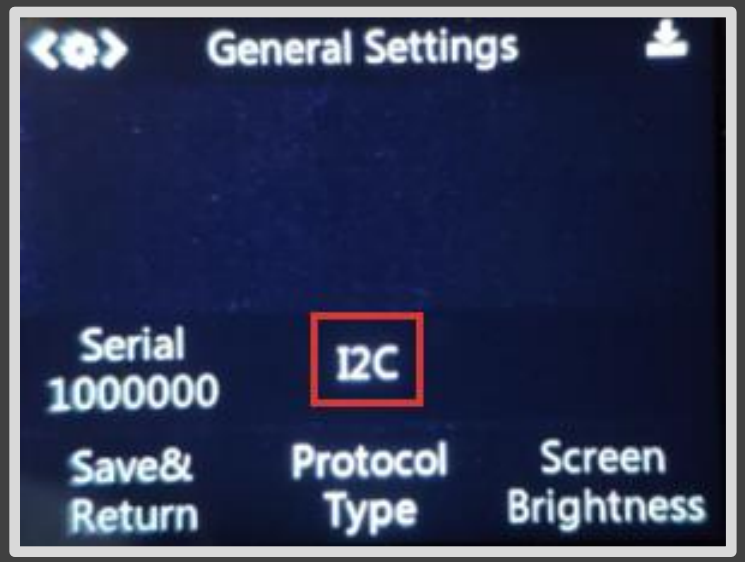
허스키 렌즈 라이브러리 설치 2

1) 스케치 -> 라이브러리 포함하기 -> .ZIP 라이브러리 추가
2) HUSKYLENS -> examples -> HUSKYLENS_LINE_TRACKING 폴더 클릭 후 열기



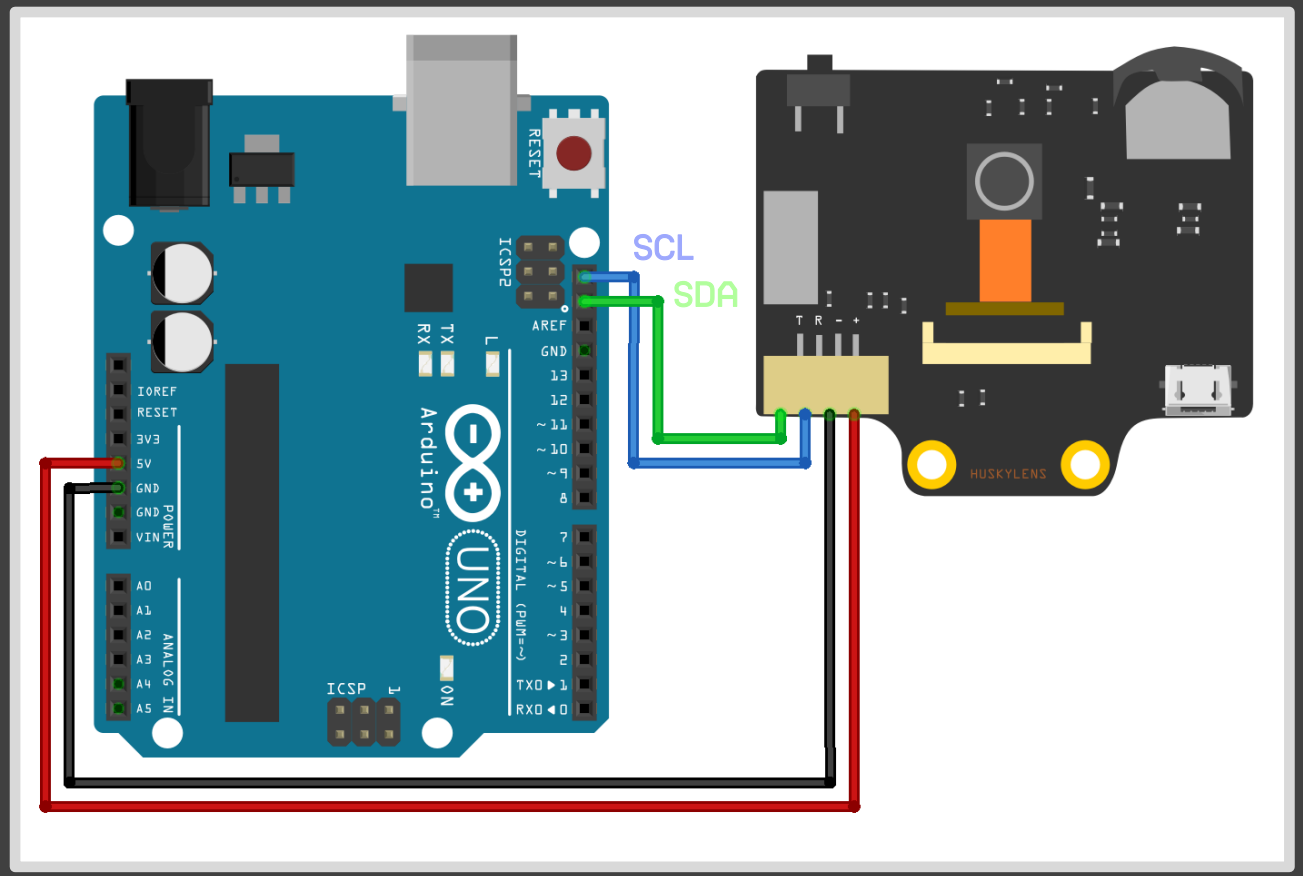
허스키 렌즈 아두이노 연결

1) 허스키 렌즈 통신 설정



General Settings 에서
Protocol Type I2C 설정 후
Save&Return

2) 회로도 연결



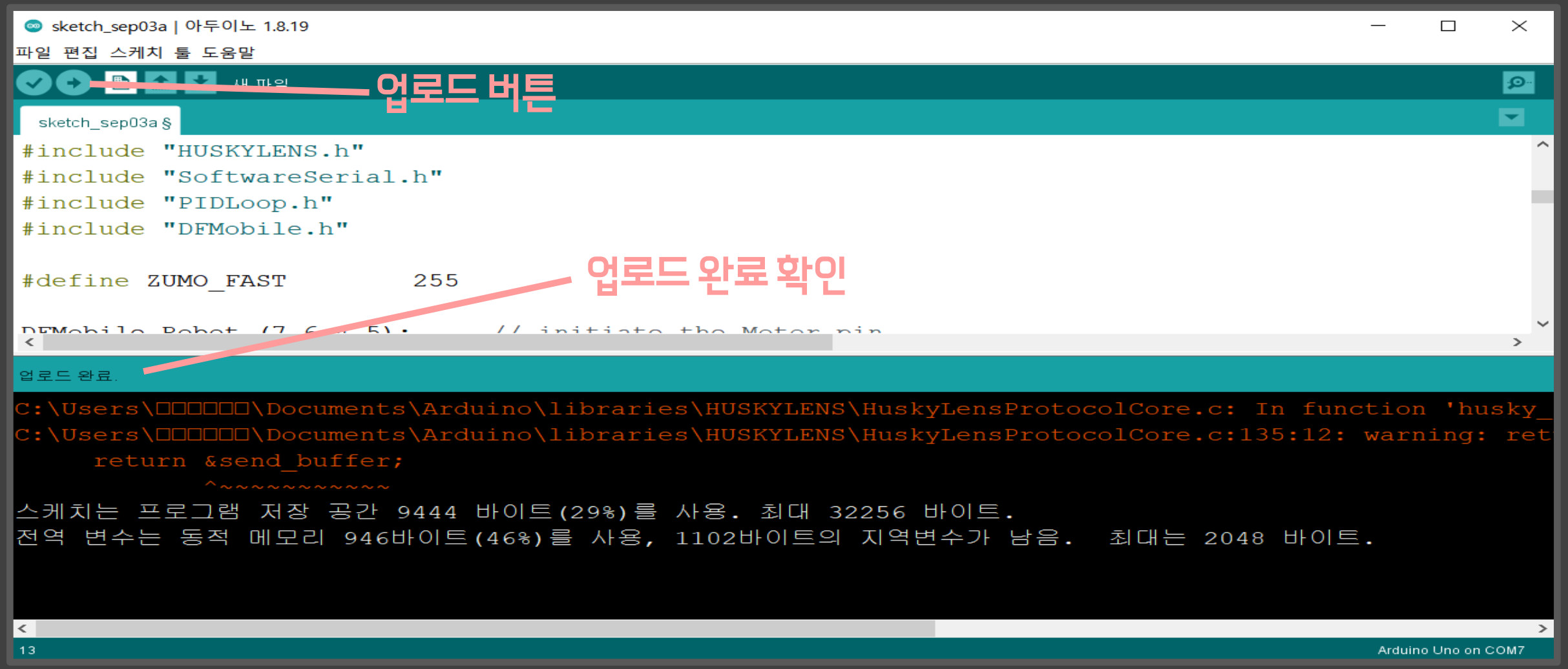
라인 추적 소스코드 실행

1) HUSKYLENS -> examples -> HUSKYLENS_LINE_TRACKING -> HUSKYLENS_LINE_TRACKING.ino 소스코드 복사

```
#include "HUSKYLENS.h"
#include "SoftwareSerial.h"
#include "PIDLoop.h"
#include "DFMobile.h"

#define ZUMO_FAST        255

DFMobile Robot (7,6,4,5);     // initiate the Motor pin
PIDLoop headingLoop(2000, 0, 0, false);
HUSKYLENS huskylens;
//HUSKYLENS green line >> SDA; blue line >> SCL
int ID1 = 1;
void printResult(HUSKYLENSResult result);



void setup() {
    Serial.begin(115200);
    Robot.Direction (HIGH, LOW);  // initiate the positive direction

    Wire.begin();
    while (!huskylens.begin(Wire))
    {
        Serial.println(F("Begin failed!"));
        Serial.println(F("1.Please recheck the \"Protocol Type\" in HUSKYLENS (General Settings>>Protol Type>>I2C)"));
        Serial.println(F("2.Please recheck the connection."));
        delay(100);
    }
    huskylens.writeAlgorithm(ALGORITHM_LINE_TRACKING); //Switch the algorithm to line tracking.
}
int left = 0, right = 0;
```

라인 추적 소스코드 실행

2) 아두이노 IDE에 소스코드 붙여넣기 후 업로드



라인 추적 소스코드 실행

3) 허스키 렌즈 Line Tracking 모드 설정

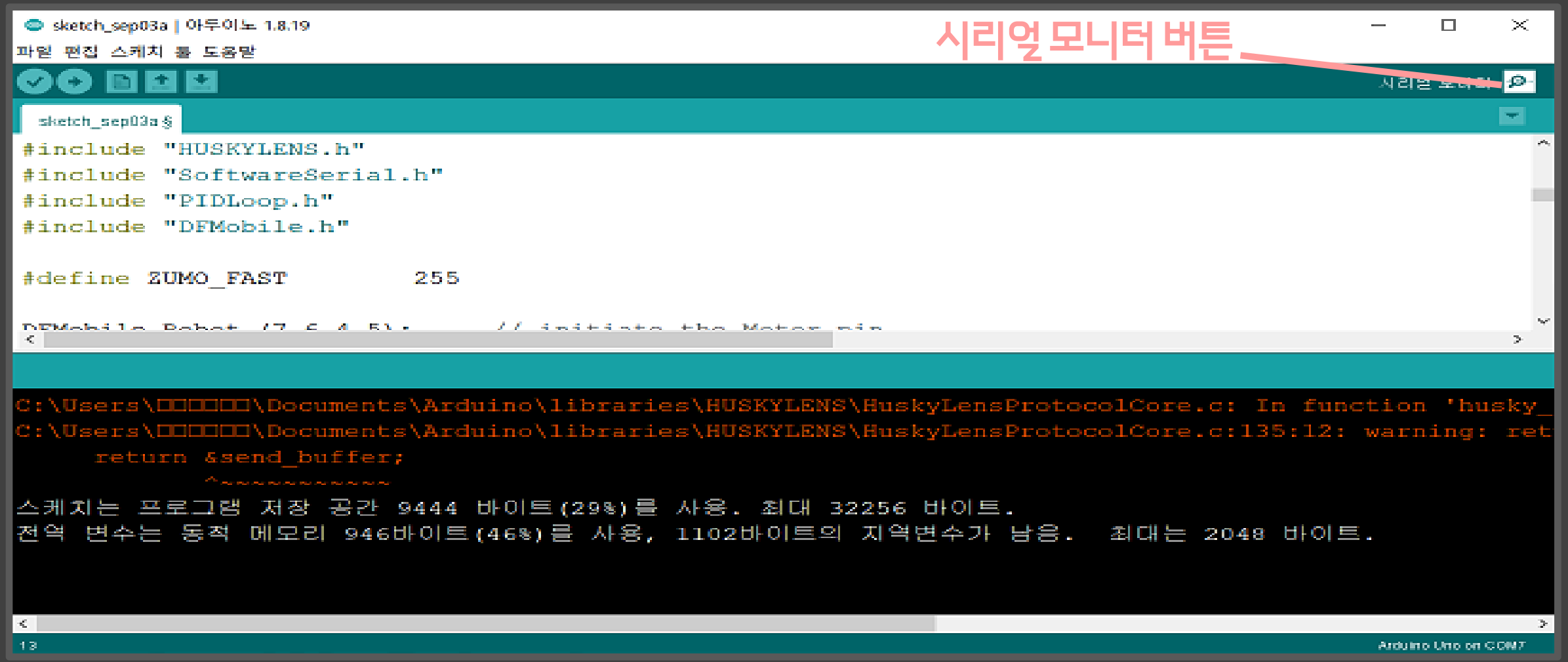
Line Tracking 설정 후
Function Button 길게 눌러
Learn Multiple 비 활성화 후
Save&Return

+ 줄 라인에 두고
Learning Button 누르기

예측 되는 화살표
방향 출력

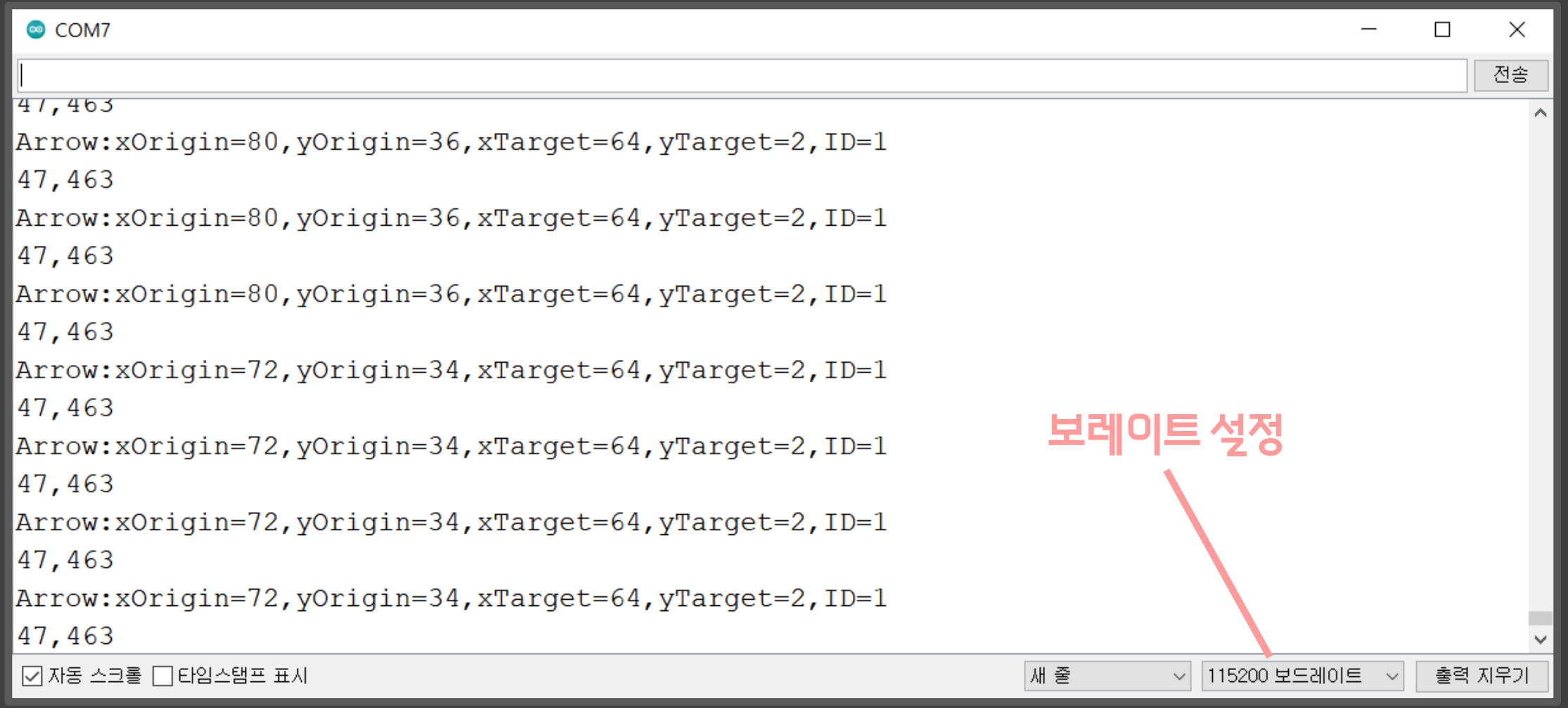
라인 추적 소스코드 실행

4) 시리얼 모니터 클릭



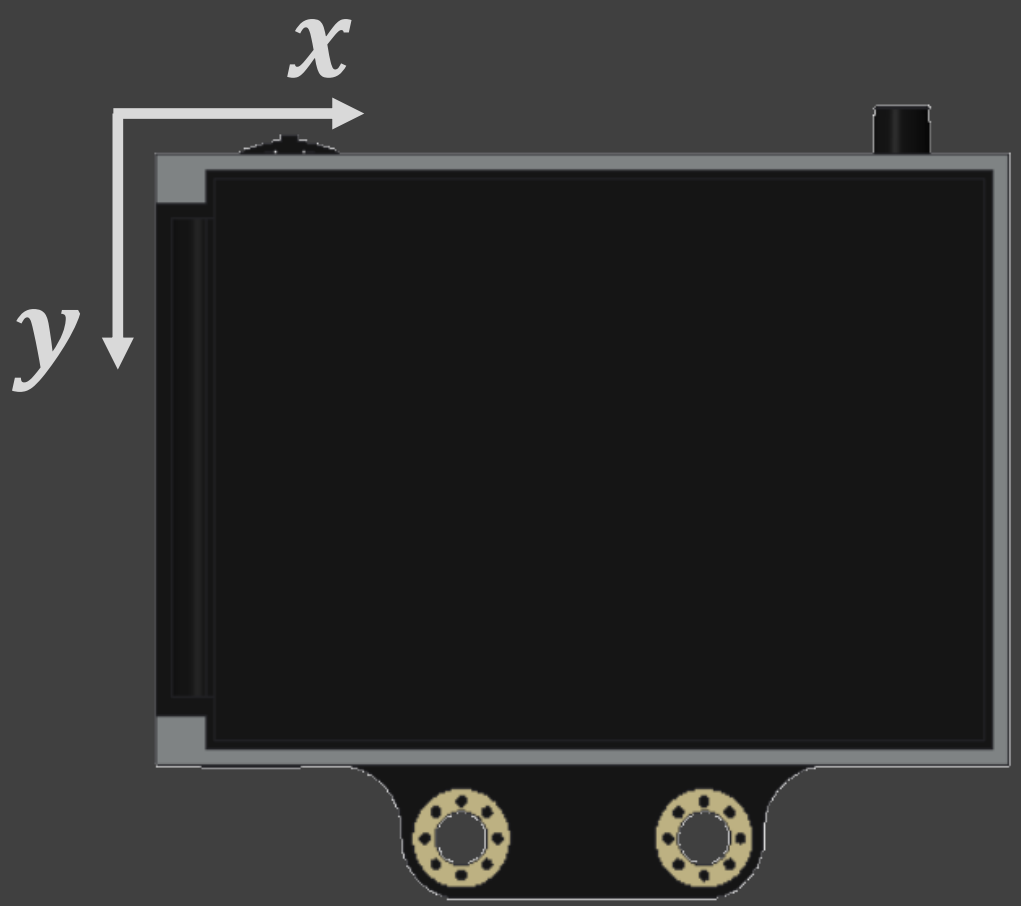
라인 추적 소스코드 실행

5) 시리얼 모니터 115200 보레이트 설정 후, 라인 인식하여 결과 확인



라인 추적 결과 분석

디스플레이 좌표계



$$x = 0 \sim 320$$

$$y = 0 \sim 240$$

라인 추적 결과 분석

시리얼 모니터 메시지

라인이 감지되면 소스코드의 printResult 함수가 호출되어 아래 내용을 출력

```
Arrow:xOrigin=80,yOrigin=36,xTarget=64,yTarget=2,ID=1
```

명칭	설명
xOrigin	화살표의 x좌표 시작 점
yOrigin	화살표의 y좌표 시작 점
xTarget	화살표의 x좌표 끝 점
yTarget	화살표의 y좌표 끝 점
ID	화살표의 ID

참고 자료

허스키 렌즈 문서

https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336#target_0

감사합니다

구선생 로보틱스