

# 임베디드 시스템 설계 및 실험

14주차 TinyML 2

## 4조

윤소현 김동윤 유주연 팜민두옹

## 실험 목표

- ESP 기반 MCU 펌웨어 포팅의 이해
- ESP-IDF 프레임워크 활용
- TensorFlow Lite 및 TensorFlow Lite for micro에 대한 이해

## 실험 내용

1. 환경 설정
  - OS : Windows
  - Visual Studio Code
  - VScode-esp-idf-extension 설치
  - 시리얼 포트 설정
2. 13주차와 같이 TinyWebTrainer으로 모델 학습 & 변환
  - Motion : 4가지 이상의 서로 다른 클래스
3. SDK Configuration (Gamba labs edu kit의 하드웨어 스펙에 맞도록 설정)
4. 모델 변경
5. 펌웨어 초기화

## 주요 개념 : ESP-IDF

**ESP-IDF(ESP IoT Development Framework)**는 ESP32 및 ESP8266 칩을 프로그래밍하기 위한 실시간 운영체제(FreeRTOS) 기반의 IoT 개발 프레임워크

- ESP-IDF는 ESP32와 ESP8266 같은 칩의 하드웨어 기능을 활용할 수 있는 라이브러리를 제공
- FreeRTOS 기반으로 멀티태스킹과 실시간 운영 체제를 지원하며, OTA 업데이트와 같은 IoT 특화 기능을 포함
- 오픈소스로 제공되어 커스터마이징과 커뮤니티 지원이 가능하며, 스마트홈, 환경 센서 등의 IoT 개발에 적합

## 실험 과정

### Todo #1 : 환경설정

1. 다음과 같이 환경 설정을 해준다.

- OS : Windows
- Visual Studio Code
- VScode-esp-idf-extension 설치
- 시리얼 포트 찾기

### Todo #2 : 모델 학습

- 트레이너로 모션을 선택하고 새로운 모델을 학습시킨다.
- 왼쪽, 오른쪽, 위, 아래 4개의 클래스를 생성하고 데이터를 수집했다.

⑤ 학습 시작 버튼 클릭으로 모델 학습 가능

⑥ 모델 테스트 페이지에서 모델 성능 검증 가능

- 변환 / 전송 과정을 통해 tflite를 추출하고 터미널에서 `xxd.exe -i 모델파일명.tflite > 바꿀파일명.c`

명령어를 통해 c 파일을 생성한다.

- 보드가 연결된 시리얼 포트를 확인하고 타겟 보드를 esp32s3으로 설정한다.

## Todo #3 : SDK Configuration

- ESP-IDF terminal 실행 후 pnu-exp-motion 폴더로 이동

- `Idf.py build` 수행

- `Idf.py menuconfig` 명령어 수행 후 보드에 대한 설정 진행

- SPI Flash의 데이터 전송 모드 : Quad I/O로 설정

- 칩에 연결된 외부 Flash memory의 크기 4MB로 설정

- 외부 PSRAM을 SPI 인터페이스로 사용하도록 설정

- PSRAM의 데이터 전송 모드 : 8비트로 설정

## 실험 결과



- 새로 만든 모델의 위, 아래, 오른쪽, 위 모션 테스트를 해본 결과 대체적으로 정확한 결과가 나왔다.