

헬라클레스 IoT 수행 보고서

헬라클레스 프로젝트의 IoT파트에서는 라즈베리 파이를 통한 자이로센서와 센서 데이터를 수신할 수 있는 Android APP을 구현했다.

1. 분석 / 설계

분석 / 설계단계에서는 Android APP의 프로토타입을 제작하고 라이더 주행의 안전성을 측정하기 위해 어떤 센서를 이용할지 탐색했다.

우리가 구현하고자 하는 주요 기능은 음성인식을 통한 이벤트 처리와 라즈베리파이와 연결된 자이로센서를 통한 어플리케이션 사용자 평가이다. 어떤 과정으로 진행되는지 알 수 있도록 Fake Application을 통해 헬라클레스의 기획의도를 보여주기로 결정했다.

1.1 Application Prototype

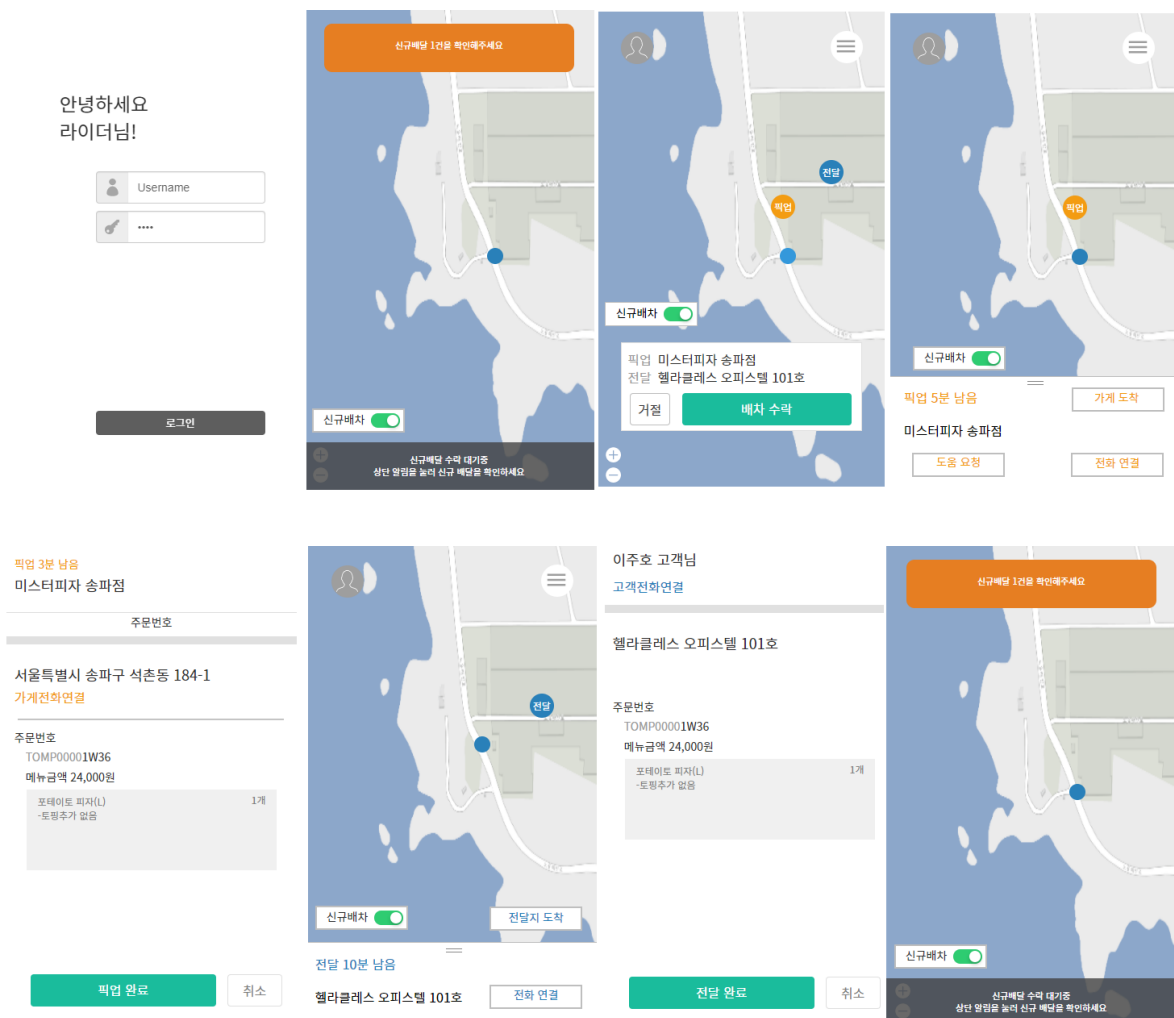


그림 1 어플리케이션 프로토타입

어플리케이션의 프로토타입은 배민커넥트의 AI배차를 모델로 제작했다.

1.2 자이로센서

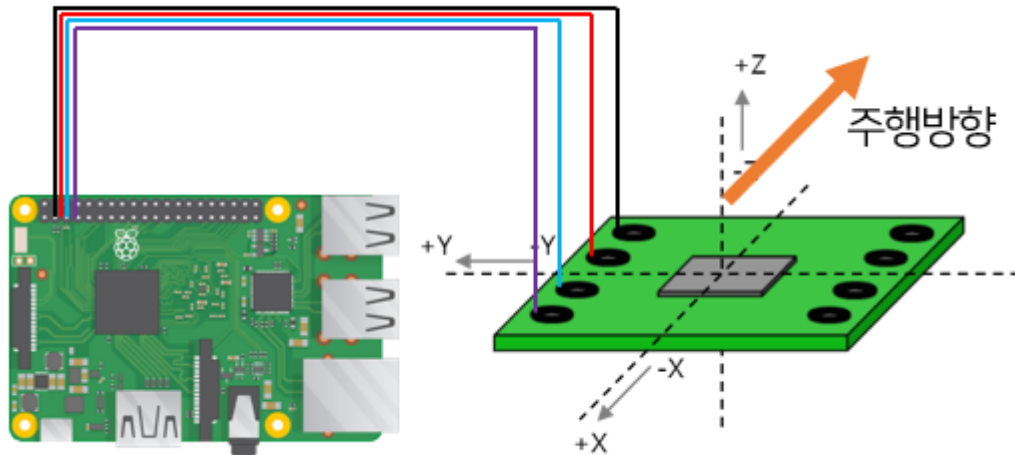


그림 2 라즈베리파이 회로도

B2B사업을 염두에 두고 설계하여 배달 대행 업체의 가방이나 오토바이의 배달통에 라즈베리 파이와 자이로센서를 부착하여 라이더의 주행데이터를 수집하고 이상치를 감지하면 라이더에게 알려줄 수 있도록 했다. 어플리케이션의 사용자는 항상 외부에서 어플리케이션을 이용하기 때문에 MQTT나 AWS IoT Core를 이용하지 않고 블루투스 소켓통신을 통해 이상치가 감지될 때 마다 안드로이드 어플리케이션으로 데이터를 보내주어 어플리케이션에서 이상치를 카운팅할 수 있도록 설계했다.

2. 구현

2.1 Wakeup-word를 통한 음성인식

Wakeup-word를 통해 음성인식을 구현하기 위해 Google SpeechRecognizer API를 이용했다. Google SpeechRecognizer API는 한번 음성을 인식한 후 바로 끝나버리기 때문에 지속적으로 Wakeup-word를 인식할 수 있도록 onResults와 onError에서 음성인식을 다시 시작하도록 구현했다. Wakeup-word는 프로젝트명 헬라클레스에서 따온 “헬라야”로 정했다. SpeechRecognizer에서 “헬라야”라고 인식하면 “말씀해주세요”라는 메시지가 TTS로 출력된다. 이후 녹음이 시작되는데 처음에는 Google MediaRecorder API를 이용해 녹음했는데 파일 인코딩의 문제가 있어 Google AudioRecorder API로 변경했다. AudioRecorder를 통해 1초후 녹음이 시작된다. 이후 4초간 녹음을 하고 pcm형태로 저장이 되는데 DB에서 음성을 처리할 수 있도록 WAV확장자로 인코딩을하여 저장하고 파일을 DB로 전송한 후 삭제한다. 전송된 음성파일은 서버에서 분석하여 사용자의 의도에 따라 인덱싱된 번호를 안드로이드 어플리케이션에서 받아오고 받아온 번호에 따라 알맞은 버튼 처리가 이루어진다.

2.2 자이로 센서 구현

Mpu-6050자이로 센서를 라즈베리 파이에 연결하고 Python을 통해 이상치를 감지한다. 이상치는 자전거 주행을 통해서 모은 데이터를 분석하여 설정했다. 자이로 센서의 x축과 y축값에서 직전의 수치와의 차이가 이상치로 설정한 값보다 크게 나오면 데이터를 블루투스 소켓통신으로 페어링된 안드로이드 기기의 어플리케이션으로 전송한다. 안드로이드 어플리케이션은 데이터를 전송받으면 사용자가 알 수 있도록 “안전주행해주세요”라는 음성메시지를 출력하고, 안드로이드 어플리케이션 내부DB의 alertCount

값을 증가시킨다. 라즈베리 파이와 안드로이드 간 통신에서 MQTT와 AWS IoT Core같은 클라우드 서비스를 이용하지 않은 이유는 배달이 외부에서 이루어지기 때문이다. 외부에서 라즈베리파이의 인터넷 연결이 이루어지려면 핫스팟이나 에그를 이용해야 하는데 거기서 얻는 이점보다 비용이 크다고 판단해 블루투스 소켓통신으로 데이터 전송을 구현했다.

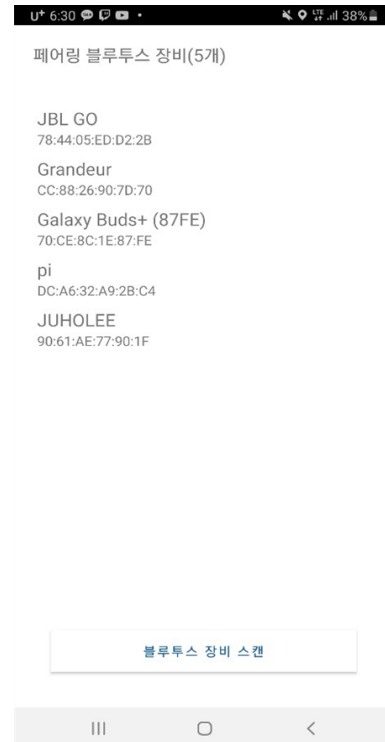


그림 3 APP 페어링 화면

```
{ 'gyro_xout': 228, 'gyro_yout': 313, 'gyro_zout': 63, 'accel_xout': -3672, 'accel_yout': 1872,
  'gyro_xout': 190, 'gyro_yout': 325, 'gyro_zout': 47, 'accel_xout': -3704, 'accel_yout': 1896,
  'gyro_xout': 178, 'gyro_yout': 410, 'gyro_zout': 76, 'accel_xout': -3772, 'accel_yout': 1916,
  'gyro_xout': 145, 'gyro_yout': 345, 'gyro_zout': 269, 'accel_xout': -3648, 'accel_yout': 1816,
  'gyro_xout': 185, 'gyro_yout': 295, 'gyro_zout': 77, 'accel_xout': -3624, 'accel_yout': 1944,
  'gyro_xout': 176, 'gyro_yout': 285, 'gyro_zout': 177, 'accel_xout': -3704, 'accel_yout': 2044,
  'gyro_xout': 198, 'gyro_yout': 337, 'gyro_zout': 19, 'accel_xout': -3684, 'accel_yout': 1916,
  'gyro_xout': 141, 'gyro_yout': 284, 'gyro_zout': 143, 'accel_xout': -3656, 'accel_yout': 1912,
  'gyro_xout': 217, 'gyro_yout': 299, 'gyro_zout': 60, 'accel_xout': -3700, 'accel_yout': 1952,
  'gyro_xout': 210, 'gyro_yout': 289, 'gyro_zout': 83, 'accel_xout': -3640, 'accel_yout': 1984,
  'gyro_xout': 203, 'gyro_yout': 371, 'gyro_zout': 70, 'accel_xout': -3704, 'accel_yout': 1936,
  'gyro_xout': 257, 'gyro_yout': 302, 'gyro_zout': 42, 'accel_xout': -3592, 'accel_yout': 2260,
  'gyro_xout': 167, 'gyro_yout': 310, 'gyro_zout': 33, 'accel_xout': -3748, 'accel_yout': 1940,
  'gyro_xout': 194, 'gyro_yout': 287, 'gyro_zout': 23, 'accel_xout': -3704, 'accel_yout': 1956,
  'gyro_xout': 215, 'gyro_yout': 8, 'gyro_zout': 72, 'accel_xout': -3660, 'accel_yout': 2284,
  'gyro_xout': 174, 'gyro_yout': 317, 'gyro_zout': 96, 'accel_xout': -3728, 'accel_yout': 1924,
  'gyro_xout': 175, 'gyro_yout': 309, 'gyro_zout': 80, 'accel_xout': -3632, 'accel_yout': 1948,
  'gyro_xout': 160, 'gyro_yout': 334, 'gyro_zout': 64, 'accel_xout': -3740, 'accel_yout': 1944,
  'gyro_xout': 203, 'gyro_yout': 290, 'gyro_zout': 98, 'accel_xout': -3744, 'accel_yout': 1976,
  'gyro_xout': 182, 'gyro_yout': 319, 'gyro_zout': 50, 'accel_xout': -3704, 'accel_yout': 1952,
  'gyro_xout': 259, 'gyro_yout': 762, 'gyro_zout': -359, 'accel_xout': -4084, 'accel_yout': 1544,
  'gyro_xout': 315, 'gyro_yout': 1282, 'gyro_zout': 136, 'accel_xout': -4228, 'accel_yout': 2680,
  'gyro_xout': -400, 'gyro_yout': -10363, 'gyro_zout': 30085, 'accel_xout': -3816, 'accel_yout':
  'gyro_xout': 1010, 'gyro_yout': 11635, 'gyro_zout': -32768, 'accel_xout': -5516, 'accel_yout':
  'gyro_xout': 3882, 'gyro_yout': 1698, 'gyro_zout': 1052, 'accel_xout': -3160, 'accel_yout': 152,
  'gyro_xout': -633, 'gyro_yout': -6877, 'gyro_zout': 19570, 'accel_xout': -3320, 'accel_yout': 2,
  'gyro_xout': 523, 'gyro_yout': -2114, 'gyro_zout': -32768, 'accel_xout': -612, 'accel_yout': 88,
  'gyro_xout': -12888, 'gyro_yout': -25129, 'gyro_zout': 32767, 'accel_xout': -16588, 'accel_yout':
  위험주행이 누적되었습니다. 안전주행하세요.
```

그림 4 라즈베리파이 위험주행 경고

3. 개선점

3.1 소음이 큰 환경에서 **Wakeup-word** 호출 문제

멘토링에서 소음이 큰 환경에서 **wakeup-word** 호출이 어렵지 않겠냐는 의견을 받았다. 실제로 소음이 클 때 어플리케이션에서 **wakeup-word** 호출이 잘 이루어지지 않았다. 이 부분은 프로젝트 기간이 얼마 남지 않아 추후에 오토바이, 자전거, 전동 킥보드 등의 핸들이나 헬멧에 버튼을 달아서 음성을 녹음하는 형태로 변환하거나 **wakeup-word** 호출시 노이즈를 제거할 수 있는 방안을 찾아볼 계획이다.

3.2 라이더 평가와 안전을 위한 센서 추가 구현

시간이 남으면 구현하기로 계획했던 기능 중에 초음파 센서, 충격감지 센서의 이용이 있었다.

초음파 센서는 오토바이나 자전거 양옆에 달아서 차들 사이로 위험하게 주행하는 라이더들에게 경고를 해주고 라이더를 평가할 수 있는 기능, 충격감지 센서는 헬멧에 부착하여 헬멧에 일정 이상 충격이 발생하면 119에 신고해줄 수 있는 기능을 추가로 생각했다. 짧은 프로젝트 기간으로 구현하지 못해 추후에 구현할 계획이다.