

긴급 구조 요청 장치

목차

1. <서론> 주제선정 배경 및 가치판단

2. 주제 구현 방법 및 보조자료

3. 작품제작 계획 및 일정표

1. <서론> 주제선정 배경 및 가치판단

현대 사회에서 안전은 최우선 과제 중 하나입니다. 특히 위험 상황에서의 신속한 대처는 생명을 구할 수 있는 결정적인 역할을 합니다. 그러나 기존의 안전 시스템은 여전히 한계가 있으며, 특히 구조 요청 시스템은 사용자의 개입이 필요한 경우가 많아 응급 상황에서의 대응이 더딘 경우가 있습니다.

범죄 사고와 사회적 안전에 대한 관심은 점점 높아지고 있습니다, 따라서 이 프로젝트는 현대 사회에서의 안전에 대한 새로운 접근 방식을 제시하고, 공공 안전에 기여하는데 큰 역할을 할 것으로 기대됩니다.

2.주제를 선택한 이유 및 해당 근거

1)주제 선정 배경

범죄자의 습격으로부터 피해를 입는 상황은 매우 심각하고 위험합니다. 이러한 상황에서 피해자가 신속하고 효과적으로 구조될 수 있도록 하는 것은 중요한 사회적 과제입니다.

현재의 구조 요청 시스템은 주로 사고나 재난 상황에서의 대응에 중점을 두고 있으며, 범죄에 대한 대비책은 제한적입니다. 이 프로젝트는 범죄자의 습격으로부터 피해를 입은 피해자를 빠르고 효과적으로 구조하기 위한 목적을 가지고 있습니다.

이 장치는 피해자가 위험에 노출되었을 때 자동으로 구조 요청을 하여 구조 대응 시간을 최소화하고, 피해를 최소화할 수 있도록 합니다. 이를 통해 사회적 안전을 강화하고, 범죄로부터 피해를 입은 피해자에게 신속한 지원을 제공하는데 기여할 것으로 기대됩니다.

2) 문제 정의

현재 범죄에 대한 대비책은 신고 앱을 통해 신고를 하거나 호루라기나 스프레이같은 대비책이 존재 하지만 이것들은 모두 피해자가 정상적으로 대처하는 상황에서 좋은 효과를 가집니다. 그러나 범죄자에 의한 피해는 여러 문제가 있습니다.

피해자의 신속한 인지 부족: 범죄자의 습격으로부터 피해를 입은 피해자는 종종 상황에 혼란을 느끼거나 비상 상황임을 인지하지 못할 수 있습니다. 따라서 피해자가 습격을 당했음을 인지하고 구조 요청을 할 수 있는 시간적 여유가 부족할 수 있습니다.

피해자의 소리를 듣지 못하는 경우: 범죄자의 습격으로 인해 피해자가 소리를 지르거나 도움을 요청해도 주변 사람들이 이를 듣지 못할 수 있습니다. 이로 인해 피해자의 구조 요청이 늦어질 수 있습니다.

피해자의 위치 파악 어려움: 범죄자의 습격으로 인해 피해자가 어디에 있는지 파악하기 어려울 수 있습니다. 이로 인해 구조 요청에 대한 대응이 지연될 수 있습니다.

따라서 자동 구조 요청 장치의 개발을 통해 이러한 문제를 해결하고, 범죄로부터 피해를 입은 피해자에게 빠르고 효과적인 지원을 제공할 수 있도록 하는 것이 중요합니다.

3) 해결책의 가치 및 중요성

자동 구조 요청 장치는 범죄자의 습격으로부터 피해를 입은 피해자를 신속하게 구조하여 생명을 보호하는 데 큰 가치와 중요성을 지닙니다.

신속한 대응 및 구조: 자동 구조 요청 장치를 통해 범죄자의 습격으로부터 피해를 입은 피해자는 즉시 구조 요청을 할 수 있습니다. 이는 사건 발생 후 신속한 대응과 구조를 가능하게 하여 피해자의 생명을 보호하는 데 중요한 역할을 합니다.

음성에 의한 작동: 범죄 사고를 당할 때 손발이 자유롭지 못한 경우나 범죄자가 손발을 제압 할 때 음성으로 구조 요청 장치를 작동 시킴으로써 가장 수동적인 상황에서 능동적으로 구조 요청을 할 수 있습니다.

피해자의 안전 보장: 자동 구조 요청 장치는 피해자가 습격을 당한 경우 즉시 구조 요청을 하여 피해자의 안전을 보장합니다. 이는 피해자의 생명뿐만 아니라 신체적, 정신적 안전을 보호하는 데 중요한 역할을 합니다.

구조 시간 단축 및 효율화: 자동 구조 요청 장치를 통해 피해자의 위치를 신속하게 파악하고 구조할 수 있습니다. 이는 구조 시간을 단축하고 구조 작업을 효율화하여 피해자의 생명을 보호하는 데 기여합니다.

따라서 자동 구조 요청 장치는 범죄자의 습격으로부터 피해를 입은 피해자를 구조하는 데 큰 가치를 지니며, 피해자의 생명을 보호하는 데 매우 중요합니다. 이를 통해 범죄에 대한 대비책을 강화하고, 사회적 안전을 보호하는데 기여할 수 있습니다.

긴급 상황 판단의 정확성: 화자 인식 모델

-사용자의 음성을 학습해 사용자를 제외한 기타 환경 소음의 간섭을 최소화

신고 내용의 구체성: GPS/GSM 모듈을 통해 장치가 독단적으로 기타 통신망으로 실시간 위치와 상황 송신 가능

접근성과 대중성:

-사용자의 직접적인 조작을 요구하지 않는, 음성 데이터로 긴급 상황을 식별&신고하는 장치이므로 어떠한 상황에서도 쉽게 접근 가능

-휴대하기 작은 사이즈와 적은 개발 비용으로 대중적으로 이용할 수 있는 제품

4) 주제 선정 근거

범죄를 예방하기 위해 피해자가 신고를 하거나 호신 용품을 들고 다니는 경우가 있다.

그러나 이런 신고 같은 경우는 불시에 발생할 때 피해자가 신고를 하는것에 어려움을 겪을 수 있습니다. 패닉에 빠진 피해자가 신고를 해야 겠다는 생각을 못하는 것은 실제로도 일어나는 일입니다.

호신 용품은 위와 마찬가지로 사용하는 것에 애로사항이 존재하는 경우가 있습니다.

피해자가 혼란과 패닉으로 제대로 사용을 못하는 경우도 존재하며

사용해도 범죄자가 완벽히 제압되지 않습니다 완벽하게 제압이 가능한 전기 충격기

같은경우는 가지고 다니기 위해 여러가지 절차가 필요하며 또한

다른 큰 이유로 이런 호신용품들은 오히려 악용되어 남을 다치게 하거나 호신용품을 사용함에 따라 자신이 처벌 받는 역울한 경우 또한 있습니다.

[https:// HYPERLINK "https://www.youtube.com/watch?v=UM5_ZkdB7-M"www.youtube.com HYPERLINK](https://www.youtube.com/watch?v=UM5_ZkdB7-M)
["https://www.youtube.com/watch?v=UM5_ZkdB7-M"/ HYPERLINK](https://www.youtube.com/watch?v=UM5_ZkdB7-M/)
["https://www.youtube.com/watch?v=UM5_ZkdB7-M"watch?v HYPERLINK](https://www.youtube.com/watch?v=UM5_ZkdB7-M)
["https://www.youtube.com/watch?v=UM5_ZkdB7-M"= HYPERLINK](https://www.youtube.com/watch?v=UM5_ZkdB7-M)
["https://www.youtube.com/watch?v=UM5_ZkdB7-M"UM5_ZkdB7](https://www.youtube.com/watch?v=UM5_ZkdB7-M)
[HYPERLINK "https://www.youtube.com/watch?v=UM5_ZkdB7-M"-M](https://www.youtube.com/watch?v=UM5_ZkdB7-M)

[https:// HYPERLINK](https://www.youtube.com/watch?v=J8RaMbdNz8Y)
["https://www.youtube.com/watch?v=J8RaMbdNz8Y"www.youtube.com](https://www.youtube.com/watch?v=J8RaMbdNz8Y)
[HYPERLINK "https://www.youtube.com/watch?v=J8RaMbdNz8Y"/](https://www.youtube.com/watch?v=J8RaMbdNz8Y)
[HYPERLINK "https://www.youtube.com/watch?v=J8RaMbdNz8Y"watch?v](https://www.youtube.com/watch?v=J8RaMbdNz8Y)
[HYPERLINK "https://www.youtube.com/watch?v=J8RaMbdNz8Y"=](https://www.youtube.com/watch?v=J8RaMbdNz8Y)
[HYPERLINK](https://www.youtube.com/watch?v=J8RaMbdNz8Y)
["https://www.youtube.com/watch?v=J8RaMbdNz8Y"J8RaMbdNz8Y](https://www.youtube.com/watch?v=J8RaMbdNz8Y)

자동 구조 요청 장치의 기존 제품들과의 차별성은 다양한 측면에서 나타납니다.

데시벨 감지 기능: 자동 구조 요청 장치는 특정 데시벨 이상이 발생했을 때만 작동하는 감지 기능을 포함할 수 있습니다. 예를 들어, 소리나 충격이 감지될 경우에만 구조 요청을 하도록 설정할 수 있어서 오인 신호를 줄이고 효율적인 대응이 가능합니다.

위치 파악 기능: 일부 자동 구조 요청 장치는 피해자의 위치를 정확하게 파악할 수 있는 GPS 기능을 내장하고 있습니다. 이는 구조 요청 시 피해자의 위치를 실시간으로 파악하여 빠른 대응이 가능하도록 도와줍니다.

화자 인식 및 특정 단어 지정 기능 : 화자인식 모델을 통해 사용자의 목소리를 인지하도록 해 정확도를 높일 수 있으며, 사전에 위험 상황에 대한 단어들을 사용자가 미리 등록해 놓아 범죄 상황에서 더 도움을 받을 수 있다는 차별점이 있다. (ex 살려주세요 , 도와주세요 등 보편적으로 많이 사용하는 단어들을 기본적으로 등록해두지만 범죄 상황에선 범죄자가 주변에 있을 경우 직접적으로 도움을 요청하는 단어를 말하기에는 피해자가 위기를 느낄 수 있는 부분이 있다. 따라서, 미리 지정 해놓은 다른 단어를 말해 신고되도록 하면 범죄자를 자극해 상황이 커지는 불상사를 막을 수 있다.) 또한 이 기능을 사용하면 사용하는 언어가 다르거나 사람마다 다른 억양 등 다양한 상황에서도 목소리를 미리 사전에 등록하는 이러한 방식을 통해 다양한 상황에서 인지할 수 있어 긴급 상황 인식의 정확성을 높일 수 있다.

정확도 개선에 관한 부분 : 화자인식 모델을 통해 다른 사람들의 대화 소리 같은 주변 환경 소음을 최대한 배제 & GSM 을 통한 SMS 송신 시 신고자의 휴대폰 번호를 같이 보내고 휴대폰 번호로 통화를 요청. >> 통화를 받지 못할 시에는 위급상황임을 인지하고 출발하는 방식.

GPS/GSM 모듈을 통해 장치가 독단적으로 기타 통신망으로 실시간 위치와 상황 등 정보를 송신 가능

긴급 구조 요청 장치에 대한 수요조사 설문(구글폼 링크)

https://docs.google.com/forms/d/e/1FAIpQLSegBdl1JUwwjxtJk0fdql1F3g4J62Ux3Os--JRSMuA7RoT4Yg/viewform?usp=sf_link

<기존 제품들의 기능과 차별점 및 기존 유사 장치들의 수요도 조사>

1. 팸케어 SOS

가격 : 94800원

기능 : 응급 호출/건강 모니터링/다수 위치공유/비상 녹음 및 녹화

단점 : 알림 신호 발신을 위해 핸드폰이 주변에 있어야 함./사용자의 조작 필요/거리 제한

2. 레이홈 SOS

가격 : 72000원

기능 : 긴급 메시지 알림/스마트폰 전용앱

단점 : 스마트폰과 연동이 필요함./사용자의 조작 필요/거리 제한

3. Retekeess 무선 호출기 SOS

가격 : 33500원

기능 : 착용(휴대)이 편리함/벨소리가 다양/수신거리가 길다/볼륨 조절 가능/방수 지원

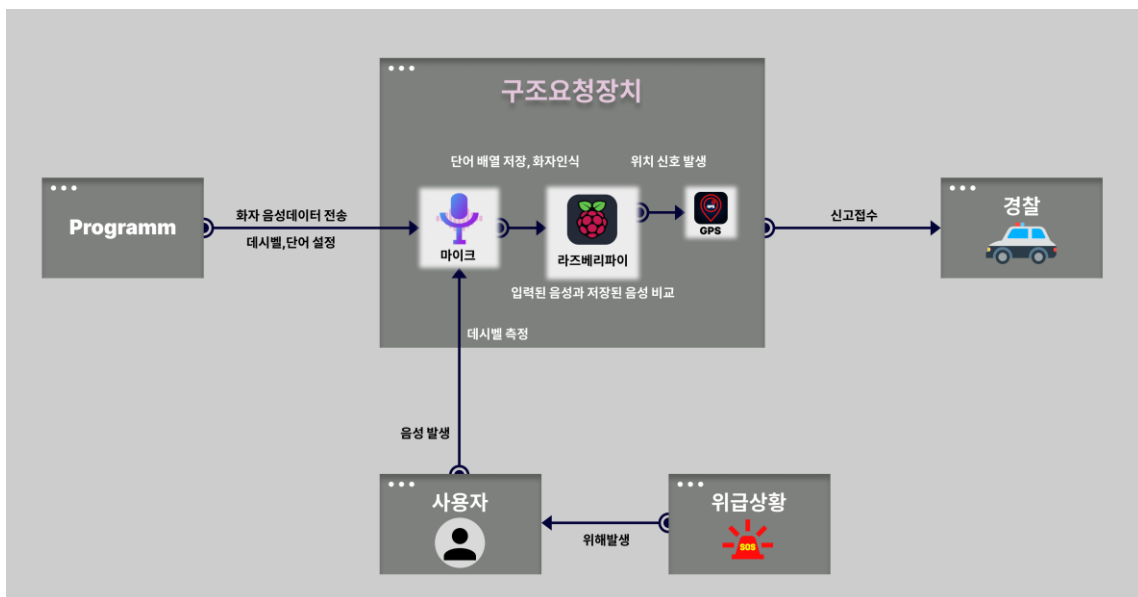
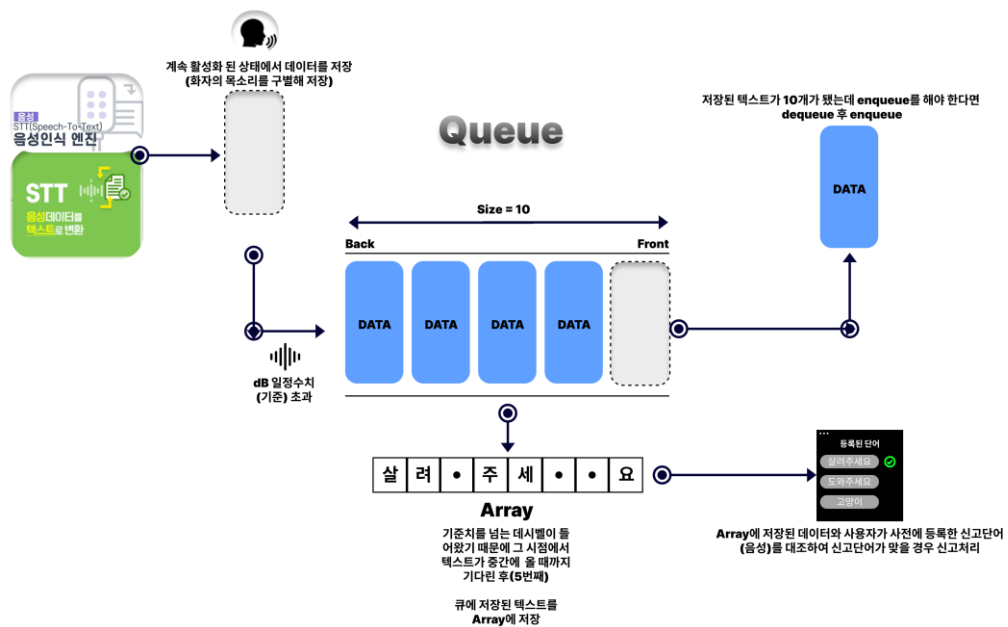
단점 : 수신기와 송신기를 따로 제작해 수신기,송신기 2가지를 모두 들고 다녀야 함.거리 제한에 대한 약평

4. 서울시 지키미 ME - 서울시 지키미 ME/ 24년1월5일부터 서울 거주자에 한정해 범죄 피해자 및 피해 우려자에게 무상 지급하고 상황 종료시 회수>>>거리 제한은 없지만 사용자의 직접적인 조작 필요// 그럼에도 불구하고 많은 시민들에게 압도적인 지지를 얻음/ 특히 자취를 하거나 어린 아이를 키우는 학부모들의 인기가 높다는조사 결과/ 5000명 모집중 2일만에 종료 -<https://mediahub.seoul.go.kr/archives/2010275>
(제품에 대한 기사와 인터뷰)

5. 스마트업 구조요청 장치 (팀 작품) - 사용자 조작 필요없음 / 거리제한 없음 (조사 결과 기존 구조요청 장치들도 기능적 제한이 있었지만 수요도가 높았음. (구글폼으로 지속적인 조사중)

2. 주제 구현방법 및 보조자료

<구현 과정에 대한 이미지>



1)제품 구현 시 필요한 소프트웨어 구성

1. 서론

1.1 배경

- 현대 사회에서 개인의 안전에 대한 중요성이 증대되면서 음성 기반의 위험 감지 및 신고 장치의 수요가 증가하고 있다.

1.2 목적



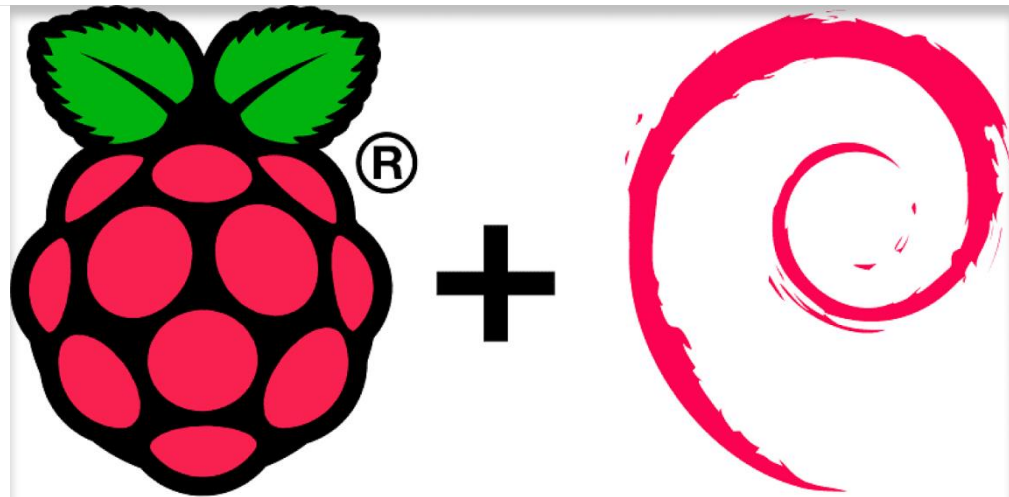
- 본 프로젝트의 목적은 긴급 상황에 돌입한 피해자가 장치의 별다른 조작 없이 자동으로 라즈베리 파이 피코를 활용하여 음성을 감지하고, 특정 구조 요청 음성을 통해 긴급 상황임을 판단, 신속하게 구조 요청과 위치 정보를 경찰서에 전송하는 소형 음성 기반 장치를 개발하는 것이다.

2. 하드웨어 및 소프트웨어 구성

2.1 하드웨어

- 라즈베리 파이 피코 보드
- 마이크로폰 센서
- GPS 모듈
- GSM 모듈
- 배터리 및 충전 장치

2.2 소프트웨어



Raspbian

- Raspbian OS
- 음성 인식 알고리즘
- MQTT 프로토콜을 활용한 통신 소프트웨어
- 위치 식별 및 데이터 전송 알고리즘

3. 음성 감지 및 분석

3.1 음성 인식



- 마이크로폰 센서를 통해 구조 요청 단어(명령어)를 감지하고 특정 명령을 수행하는 음성 인식 모델을 개발한다.
- 화자 인식 모델을 추가해 주변 환경 소음을 최대한 제외시켜 정확성을 높인다.



4. 위치 식별 및 데이터 전송

4.1 GPS 모듈 활용

- GPS 모듈을 활용하여 장치 사용자의 위치 정보를 측정하고 추출한다.

4.2 GSM 를 통한 데이터 전송

- GSM 을 사용하여 위치 정보 및 음성 감지 결과를 사용자의 전화번호와 함께 경찰서 또는 관련 당국에 실시간으로 전송한다.

Message Queuing Telemetry Transport (MQTT) Protocol



5. 소형 사이즈 및 휴대성

5.1 하드웨어 소형화

- 소형화된 음성인식, GPS, GSM 모듈을 통해 소형 사이즈(파운데이션, 텀블러의 크기)의 장치를 설계한다.

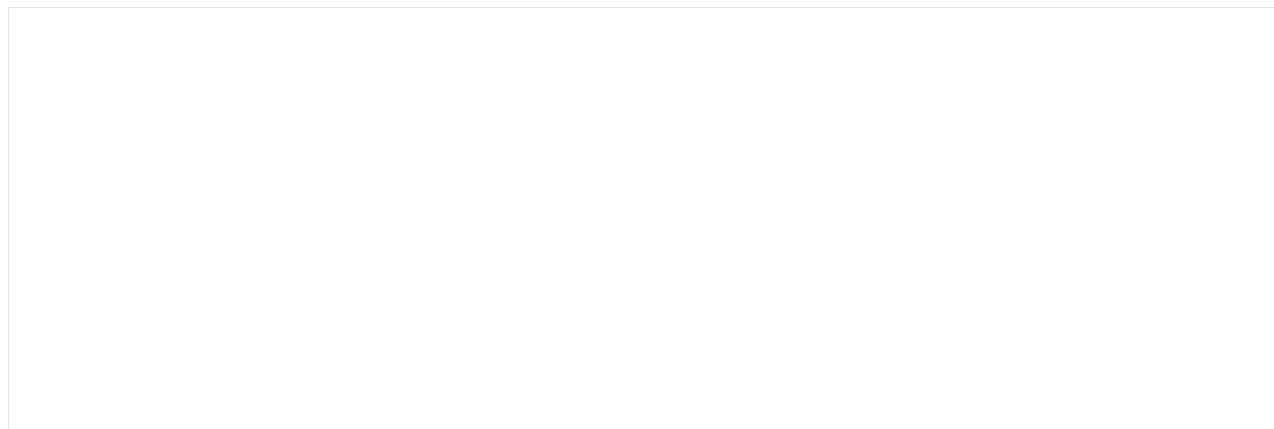
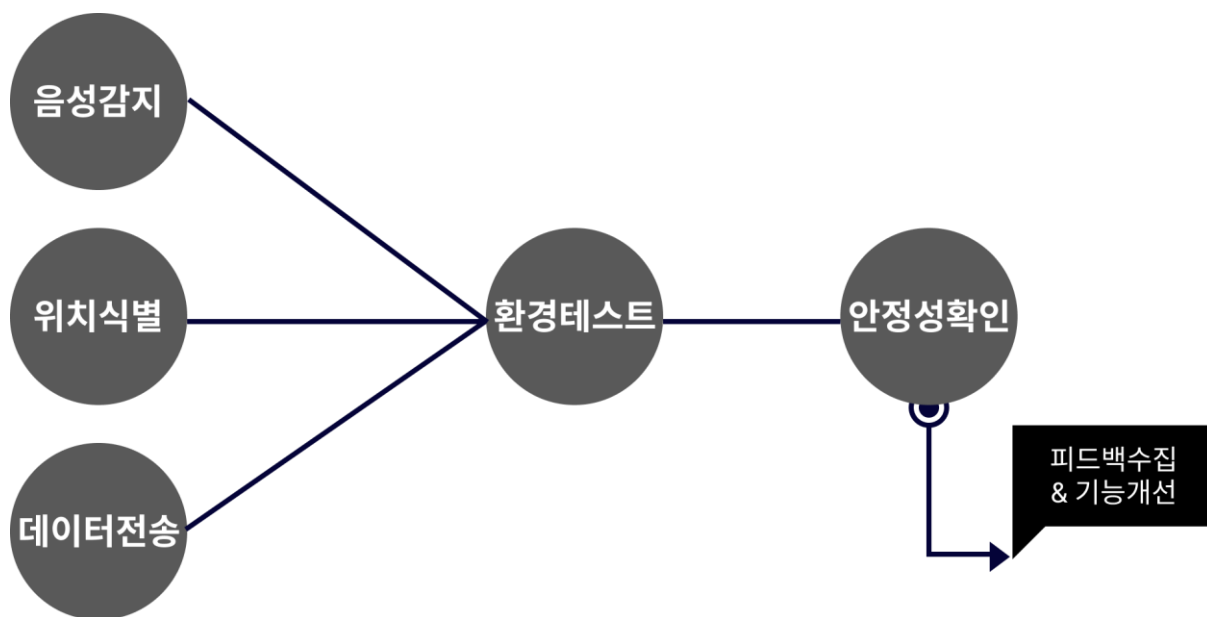
6. 테스트 및 개선

6.1 시뮬레이션 테스트

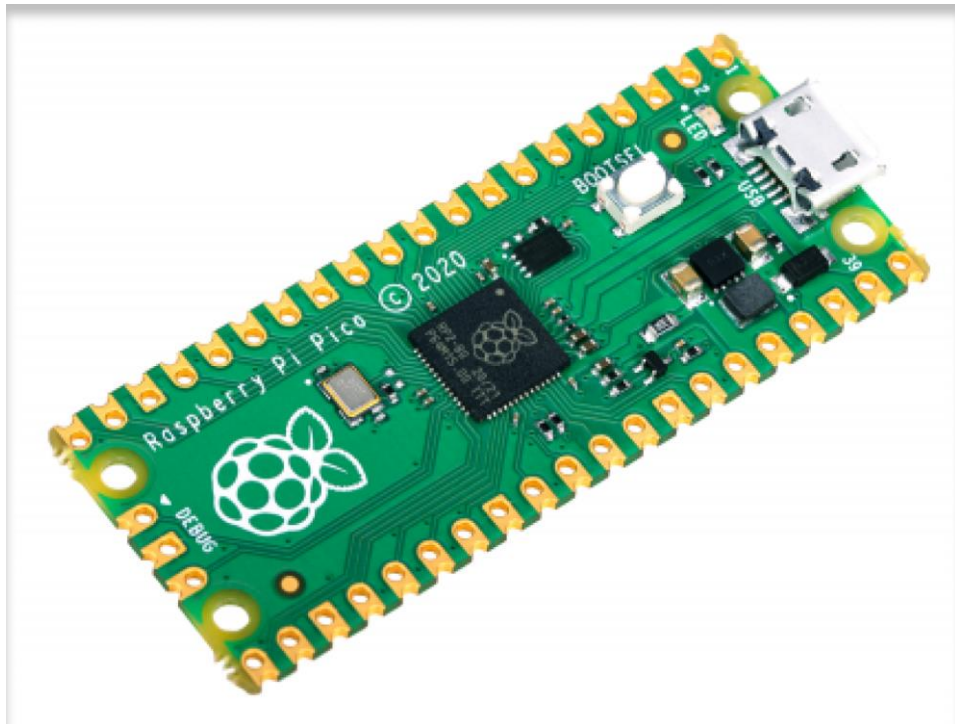
- 음성 감지, 긴급 상황 인식, 위치 식별, 데이터 전송 등의 기능을 시뮬레이션 환경에서 테스트하여 안정성을 확인한다.

6.2 사용자 피드백 수집

- 사용자에게 실제 환경에서 테스트하고 피드백을 수집하여 기능을 개선한다.



라즈베리 파이 피코 보드



라즈베리 파이 Pico 는 소형 마이크로 컨트롤러 보드로, 저렴하면서도 성능이 출중하여 다양한 임베디드 프로젝트에 적합합니다. 이를 사용하는 이유는 다음과 같습니다.

1. 소형 사이즈,경제성:

- 라즈베리 파이 Pico 는 매우 작은 크기를 가지고 있어, 휴대성이 뛰어나며 작은 공간에 적합합니다. 휴대용 음성 기반 장치를 개발할 때 소형 사이즈는 중요한 요소입니다. 또한 저렴한 가격으로 예산 절감 또한 가능합니다.

2. 저전력 소비:

- Pico 는 저전력 마이크로컨트롤러로, 배터리로 구동되는 휴대용 장치에 적합합니다. 효율적인 전력 소비는 긴급 상황에서 장치의 지속적인 사용이 가능하도록 도와줍니다.

3. GPIO 및 다양한 센서 지원:

- Pico 는 다양한 GPIO 핀과 다양한 센서 및 외부 장치를 연결할 수 있는 핀을 제공합니다. 이는 음성 감지, GPS 모듈, GSM 모듈 등을 효과적으로 연결하여 사용할 수 있도록 합니다.

GPS 모듈 및 GSM 모듈 활용

1. GPS 모듈:

- 긴급 상황에서 사용자의 정확한 위치를 파악하여 신속한 대응이 가능하도록 합니다.

2. GSM 모듈:

• 설명 및 원리:

- GSM 모듈은 휴대폰 네트워크와 통신할 수 있는 모듈로, 음성 통화, 문자 메시지, 데이터 통신 등을 지원합니다. 이 모듈은 SIM 카드를 사용하여 통신합니다.

• 활용:

- 긴급 상황 시 경찰서 또는 응급 서비스에 사용자의 위치 및 상황을 전송하는 데 사용됩니다.

3. 활용 시나리오:

- 음성 감지 및 마이크로폰 센서가 사용자의 긴급 상황을 인식하면, 작동하여 위험 상황을 확인합니다.
- GPS 모듈은 사용자의 현재 위치를 식별하고, GSM 모듈을 사용하여 위치 및 음성 감지 정보를 경찰서나 관련 당국에 전송합니다.

4. 소프트웨어 통합:

- GPS 데이터 및 음성 감지 정보를 효과적으로 처리하고 MQTT 또는 다른 통신 프로토콜을 사용하여 데이터를 전송하는 소프트웨어를 개발합니다.

GPS 모듈의 정확도(신뢰성)

5. GPS 모듈:

• GPS 모듈의 정확도에 대해 사용할 센서에 대한 설명

- GPS 2-in-1 및 GSM 기능 모듈
- 쉽게 사용할 수 있도록 Crowtail 인터페이스를 추가
- GSM/GPRS 쿼드 밴드 네트워크를 지원하고 위성 항법을 위해 GPS 기술을 결합한다. 22 개의 추적 센서와 66 개의 수신 채널을 갖추고 있어 높은 수신 감도를 통해 위치 추적, 음성, SMS 및 데이터를 프로젝트에 추가할 수 있도록 한다.
- 최대 오차 범위는 2.5m 안쪽이다

• GPS 모듈의 정확도에 대해 사용할 센서의 특징

- 쿼드 밴드 850/900/1800/1900MHz

- AT 명령으로 제어 (3GPP TS 27.007, 27.005 및 SIMCOM 고급 AT 명령)
- 통합 GPS/CNSS 및 A-GPS 지원
- GPS NMEA 프로토콜 지원
- GPRS 다중 슬롯 class12 연결 : 최대 85.6kbps

GSM 모듈을 활용한 경찰서로 자동 신고

1. GSM 모듈 및 라즈베리 파이 Pico 연결:

- 선택한 GSM 모듈을 라즈베리 파이 Pico 에 연결합니다. 주로 UART (시리얼 통신) 또는 GPIO (일반적으로 AT 명령을 통한 제어에 사용)를 사용하여 연결합니다.

2. SIM 카드 삽입 및 인증:

- GSM 모듈을 사용하기 위해 SIM 카드를 삽입하고, 모듈을 통해 SIM 카드의 PIN 을 인증합니다.

3. GSM 네트워크에 연결:

- GSM 모듈을 사용하여 GSM 네트워크에 연결합니다. 이 단계에서는 모바일 네트워크에 등록되어야 합니다.

4. SMS 메시지 전송:

- 작성한 SMS 메시지를 GSM 모듈을 통해 경찰서의 전화번호로 전송합니다. 이는 AT 명령을 사용하여 수행됩니다.

트리거로써 다시 구조요청을 해제하지 않을 때까지는 GPS 정보를 1 분단위로 자동으로 경찰서에 전송

5. 응답 처리:

- 경찰서로부터의 응답을 처리합니다. 응답은 보통 SMS 로 수신되며, 이를 해석하여 필요한 조치를 취할 수 있습니다.

- AIHub 화자인식 모델 활용 과정



1. 모델 활용 준비작업

- Install Kaldi toolkit <https://kaldi-asr.org/doc/install.html>
- path.sh 에서 KALDI_ROOT 수정
- 다음 디렉토리를 Kaldi 경로에서 실행할 폴더로 복사
 - \$KALDI_ROOT/egs/voxceleb/v1/local
 - \$KALDI_ROOT/egs/sre08/v1/sid
 - \$KALDI_ROOT/egs/wsj/s5/steps
 - \$KALDI_ROOT/egs/wsj/s5/utls

2. 테스트

1) 두 음성 파일간의 유사도 측정

- score_two_wavs.sh <input: wav 파일 1> <input: wav 파일 2> <output : score 저장 파일명>
- e.g., score_two_wav.sh A0003-0000F1012-10120100-00000100.wav C0002-0085F1111-100000_0-00021235.wav plda_score.txt

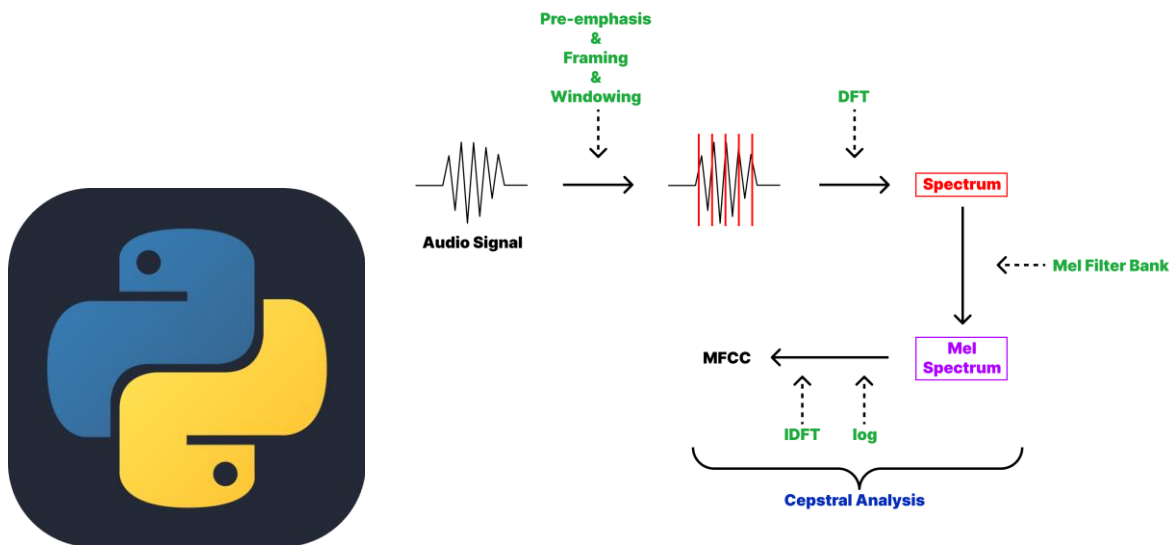
2) 모델 성능 계산(EER)

- 테스트 데이터의 wav.scp, utt2spk, trials 를 생성
 - 테스트에 사용할 wav 파일들을 특정 경로에 복사
 - gen_data_dir.sh <input: 테스트 데이터 DB 경로> <output: 생성할 kaldi input 경로>
 - e.g., gen_data_dir.sh /home/nia/testdata ./input/testdata1
- compute_eer.sh <input: 생성한 kaldi input 폴더> <output: 결과 저장 경로>
- e.g., compute_eer.sh input/testdata1 scores/test1
- 결과 저장 경로 내 test xvectors, eer, scores 파일 생성

데이터셋	활용 모델	오류 지표(EER)	Real-time
AI-Hub (aihub.or.kr)	X-vectors	4.606%	라즈베리파이 피코에서의 구동 자료 없음(직접 구현 필요)

위의 화자인식 방식은 인터넷상 자료가 많지 않아 라즈베리파이 피코에서의 real time 을 예측하기 어렵다.
그렇기 때문에 라즈베리파이 피코에서 구현 후 real time 이 길어질 때를 대비하여 대체할 프로그램을 찾았다.
이 방식은 기존의 화자인식방식보다는 정확도가 낮지만 대략 86%의 준수한 성능을 가졌고 파이썬의 라이브러리만 사용하기 때문에 real time 부분에서는 기존의 방식보다 좋은 성능을 낼 것이 예상된다.

파이썬에서 MFCC 를 통한 화자 인식을 구현하는 과정



1. ****데이터 수집****: 먼저, 화자 인식을 위한 음성 데이터를 수집합니다. 이 데이터는 여러 화자의 음성 샘플을 포함해야 하며, 가능한 많은 다양성을 가질수록 좋습니다.
2. ****MFCC 특징 추출****: 수집한 음성 데이터에서 MFCC(Mel-Frequency Cepstral Coefficients)를 추출합니다. 이는 각 화자의 음성을 대표하는

특징 벡터를 생성하는 과정입니다. Librosa 와 같은 라이브러리를 사용하여 MFCC 를 추출할 수 있습니다.

3. ****데이터 전처리****: 추출된 MFCC 특징 벡터를 사용하여 데이터를 전처리합니다. 이 단계에서는 주로 특징 스케일링이나 차원 축소 등의 기술을 사용하여 데이터를 준비합니다.

4. ****학습 데이터 분할****: 전처리된 데이터를 학습 및 테스트 세트로 분할합니다. 이를 위해 데이터를 랜덤하게 섞고, 일반적으로 학습 세트와 테스트 세트를 80:20 또는 70:30 등의 비율로 분할합니다.

5. ****모델 학습****: 분할된 학습 세트를 사용하여 화자 인식 모델을 학습합니다. SVM(Support Vector Machine), k-NN(k-Nearest Neighbors), 신경망 등 다양한 분류 알고리즘을 사용할 수 있습니다.

6. ****모델 평가****: 학습된 모델을 사용하여 테스트 세트에서 화자 인식을 수행하고, 예측 결과를 실제 레이블과 비교하여 모델의 성능을 평가합니다. 이를 통해 정확도, 정밀도, 재현율 등의 성능 지표를 계산할 수 있습니다.

7. ****모델 튜닝****: 모델의 하이퍼파라미터를 조정하고 다양한 실험을 통해 모델을 향상시킵니다. 이를 통해 최적의 화자 인식 모델을 찾을 수 있습니다.

8. ****실시간 화자 인식****: 최종적으로 개발된 모델을 사용하여 실시간으로 오디오 입력에서 화자를 인식할 수 있습니다. 이를 위해 오디오 입력을 실시간으로 읽고 MFCC 를 추출한 후 모델에 입력으로 전달하여 화자를 인식합니다.

구현 예시 : [\[음성/오디오 신호처리\] Python 을 이용한 deep learning - 화자 인식 : 네이버 블로그 \(naver.com\)](#)

제한된 환경 (라즈베리 파이 피코)에서의 화자인식 모델 적용 방법

1. 사용될 음성 모델의 경량화
2. TensorFlow Lite로의 경량화



원본 모델 준비:

AIHub에서 다운로드한 모델을 사용 :

<https://aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=data&dataSetSn=537>

TensorFlow Lite Micro로의 경량화:

TensorFlow에서 제공하는 tflite 변환 도구를 사용하여 모델을 TensorFlow Lite 형식으로 변환합니다.

예시:

```
tensorflow_lite_converter --saved_model_dir=model --output_file=model.tflite
```

TensorFlow Lite 모델 통합:

TensorFlow Lite 런타임을 사용하여 모델을 통합하고, 이를 PC프로그램이나 라즈베리 파이 피코에서 사용



1. PC 프로그램을 통해 장치에 사용자 등록 (화자 인식)

1.1 사용자 등록 기능 개발:

a. UI 및 사용자 정보 입력 화면 구현:

- PC 프로그램에는 사용자 등록을 위한 화면을 만듭니다.
- 사용자는 이름, ID 등의 정보를 입력할 수 있는 폼을 제공합니다.

b. 사용자 정보 저장 기능 구현:

- 사용자가 입력한 정보를 데이터베이스나 파일 시스템에 저장합니다.
- 사용자에게 할당된 고유한 ID와 화자 특성을 저장합니다.

1.2 화자 인식 알고리즘 개발:

a. 음성 샘플 수집:

- 사용자로부터 몇 번의 음성 샘플을 수집합니다.
- 이 샘플은 화자 특성 학습에 활용됩니다.

b. 화자 인식 모델 훈련:

- 수집한 음성 샘플을 사용하여 화자 인식 모델을 훈련시킵니다.
- 구조 요청장치에도 real time 을 위해 화자 인식 모델을 추가해야 하므로 Tensorflow lite 나 MicroPython 를 사용해 화자인식 모델을 최대한 경량화
-

c. 모델 연동 및 검증:

- 개발한 화자 인식 모델을 PC 프로그램에 연동하고, 등록된 사용자의 음성을 통해 화자를 식별하는 기능을 구현합니다.
- 모델의 성능을 테스트하여 정확성을 확인합니다.

1.3 시스템 통합 및 테스트:

a. 시스템 통합:

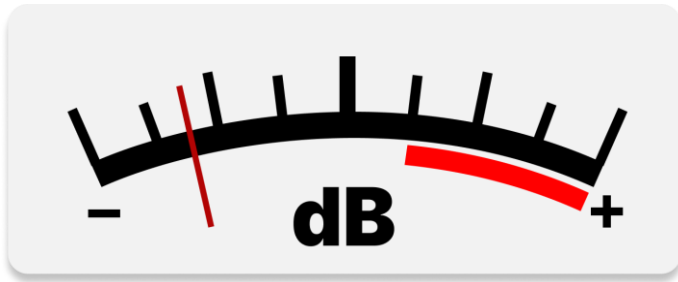
- 사용자 등록 및 화자 인식 모듈을 시스템에 통합합니다.

b. 사용자 등록 테스트:

- 등록된 사용자 정보를 활용하여 화자를 인식하는 기능을 테스트합니다.
- 다양한 사용자 음성을 사용하여 정확성을 검증합니다.

c. UI/UX 테스트:

- 사용자 등록 화면의 UI/UX 를 테스트하여 사용자가 편리하게 등록할 수 있는지 확인합니다.



2. 트리거를 발동시킬 사용자만의 데시벨 등록

2.1 트리거 등록 및 데시벨 설정 기능 개발:

a. UI 구현:

- 사용자가 트리거 및 데시벨을 등록할 수 있는 화면을 프로그램에 추가합니다.
- 사용자는 트리거 단어 및 해당 트리거에 대한 데시벨 값을 입력할 수 있습니다.

b. 트리거 및 데시벨 정보 저장:

- 입력된 트리거와 데시벨 정보를 사용자 정보와 함께 저장합니다.
- 각 사용자에게 여러 트리거를 등록할 수 있습니다.

2.2 시스템 통합 및 테스트:

a. 시스템 통합:

- 새로운 트리거 및 데시벨 등록 기능을 기존 시스템에 통합합니다.

b. 트리거 및 데시벨 설정 테스트:

- 사용자가 등록한 트리거가 정확히 저장되고, 해당 트리거에 대한 데시벨 값이 설정되는지 확인합니다.
- 여러 사용자 및 트리거에 대한 테스트를 진행하여 정확성을 검증합니다.

2.3 문서화 및 배포:

a. 사용자 가이드 업데이트:

- 새로운 기능에 대한 사용자 가이드를 업데이트하여 사용자가 트리거와 데시벨을 등록하는 방법을 설명합니다.

3. 사용자가 트리거로 사용할 단어 등록

3.1 사용자 트리거 등록 기능 개발:

a. UI 구현:

- 사용자가 트리거로 사용할 단어를 등록할 수 있는 화면을 프로그램에 추가합니다.
- 사용자는 원하는 트리거 단어를 입력할 수 있습니다.

b. 트리거 단어 저장:

- 입력된 트리거 단어를 사용자 정보와 함께 저장합니다.
- 각 사용자에게 여러 트리거 단어를 등록할 수 있습니다.

3.2 시스템 통합 및 테스트:

a. 시스템 통합:

- 새로운 트리거 단어 등록 기능을 기존 시스템에 통합합니다.

b. 트리거 단어 등록 테스트:

- 사용자가 등록한 트리거 단어가 정확히 저장되는지 확인합니다.
- 여러 사용자 및 트리거 단어에 대한 테스트를 진행하여 정확성을 검증합니다.

3.3 문서화 및 배포:

a. 사용자 가이드 업데이트:

- 새로운 기능에 대한 사용자 가이드를 업데이트하여 사용자가 트리거 단어를 등록하는 방법을 설명합니다.

4. 위 3가지 데이터를 장치 내부의 특정 메모리로 저장

4.1 데이터 저장 기능 개발:

a. 특정 메모리 할당:

- 장치 내부의 특정 메모리를 할당하고, 이를 사용하여 사용자 등록 정보, 데시벨 설정, 트리거 단어를 저장합니다.

b. 데이터 저장:

- 사용자 등록 정보, 데시벨 설정, 트리거 단어를 해당 특정 메모리에 저장합니다.

4.2 시스템 통합 및 테스트:

a. 시스템 통합:

- 특정 메모리 할당 및 데이터 저장 기능을 기존 시스템에 통합합니다.

b. 데이터 저장 테스트:

- 사용자 정보, 데시벨 설정, 트리거 단어가 정확히 특정 메모리에 저장되는지 확인합니다.
- 여러 사용자 및 설정에 대한 테스트를 진행하여 정확성을 검증합니다.

4.3 문서화 및 배포:

a. 사용자 가이드 업데이트:

- 새로운 기능에 대한 사용자 가이드를 업데이트하여 특정 메모리에 데이터를 저장하는 방법을 설명합니다.

5. 구조 장치의 음성 인식 모듈 선택

5.1 ReSpeaker 2-Mics Pi HAT for Raspberry Pi

특징:

네 개의 마이크로폰을 포함하고 있어 180 도 범위에서 음성을 수집할 수 있습니다.

긴 선형 배열로 구성되어 있어 음성의 방향성을 높일 수 있습니다.

Acoustic Echo Cancellation (AEC) 및 Noise Suppression 과 같은 고급 음성 처리 알고리즘을 내장하고 있습니다.

I2S 인터페이스를 통해 라즈베리 파이 Pico 와 간편하게 연결할 수 있습니다.

5.2 선택 이유:

다양한 음향 환경 대응:

ReSpeaker 2-Mics Pi HAT for Raspberry Pi

는 여러 마이크로폰을 통해 다양한 음향 환경에서 화자를 정확하게 감지할 수 있어 선택하게 되었습니다.

통합된 음성 처리 알고리즘:

AEC 및 Noise Suppression 과 같은 음성 처리 알고리즘이 내장되어 있어 뛰어난 음질과 정확도를 기대할 수 있습니다.

라즈베리 파이 Pico 와의 호환성:

I2S 인터페이스를 통해 간편하게 라즈베리 파이 Pico 와 연결이 가능하며, 제품 간의 호환성이 높습니다.

이러한 이유로 ReSpeaker 2-Mics Pi HAT for Raspberry Pi 를 선택하였습니다

6. 장치 STT기능 상시 작동

6.1 STT 기능 통합:

a. 모듈 연동:

- 선택한 음성 인식 모듈을 구조 장치 시스템에 연동합니다.
- 모듈과 시스템 간의 통신을 설정하고 필요한 API 키 등을 적절히 구성합니다.

b. 상시 작동 설정:

- 시스템이 계속해서 음성을 감지하고, 이를 선택한 STT 모듈로 전달하여 텍스트로 변환하는 기능을 상시 작동하도록 설정합니다.

6.2 시스템 통합 및 테스트:

a. 모듈 통합 및 설정 확인:

- 연동한 STT 모듈이 정확하게 시스템에 통합되었는지 확인하고 설정이 올바르게 적용되었는지 확인합니다.

b. 상시 작동 테스트:

- 장치가 음성을 지속적으로 감지하며 STT 기능이 올바르게 동작하는지 확인하는 테스트를 수행합니다.

7. 특정 데시벨 감지

7.1 데시벨 감지 모듈 통합:

a. 모듈 연동:

- 데시벨을 감지하기 위한 모듈을 시스템에 연동합니다.
- 필요한 하드웨어나 소프트웨어 모듈을 구성하고, 시스템과 통신할 수 있도록 설정합니다.

b. 데시벨 감지 임계값 설정:

- 특정 데시벨 이상을 감지할 때의 임계값을 설정합니다.
- 임계값은 사용자가 설정 가능하도록 설계할 수 있습니다.

7.2 시스템 통합 및 테스트:

a. 데시벨 감지 모듈 통합 확인:

- 연동한 데시벨 감지 모듈이 정확하게 시스템에 통합되었는지 확인합니다.

b. **데시벨 감지 테스트:**

- 시스템이 특정 데시벨 이상을 정확하게 감지하는지 확인하는 테스트를 수행합니다.
- 다양한 데시벨 수준에서 테스트하여 정확성을 검증합니다.

8. 등록된 사용자 화자인식 데이터와 특정 데시벨 이상에서 감지된 음성 데이터 비교 단계

8.1 사용자 인식 및 음성 데이터 처리:

a. **등록된 사용자 정보 조회:**

- 감지된 음성 데이터를 처리하기 전에 등록된 사용자의 화자인식 데이터를 조회합니다.

b. **화자 인식 모듈 활용:**

- 등록된 사용자 정보와 함께 화자 인식 모듈을 활용하여 감지된 음성 데이터의 화자를 판별합니다.

8.2 음성 데이터와 데시벨 비교:

a. **데시벨 감지 확인:**

- 특정 데시벨 이상이 감지되었는지 확인합니다.
- 이 단계는 1 초 이내로 이루어져야 합니다.

b. **데시벨 이상일 경우 데이터 비교:**

- 특정 데시벨 이상이 감지된 경우, 등록된 사용자의 화자인식 데이터와 비교하여 사용자 여부를 확인합니다.

8.3 시스템 통합 및 테스트:

a. 시스템 통합:

- 사용자 인식 및 음성 데이터 처리 모듈을 시스템에 통합합니다.

b. 인식 및 비교 테스트:

- 사용자의 음성이 특정 데시벨 이상으로 감지되었을 때, 화자 인식과 데이터 비교가 정확하게 동작하는지 확인하는 테스트를 수행합니다.

9. 감지한 STT 데이터 텍스트 처리 및 구조 문구 판별 단계

9.1 큐 관리 및 데이터 저장:

a. 한정된 큐 구현:

- 큐를 구현하여 STT 데이터 텍스트를 저장합니다.
- 큐의 크기를 10으로 설정하고, 큐가 가득 차면 enqueue를 하기 전에 dequeue를 수행합니다.

b. 데시벨 감지 이벤트 대기:

- 데시벨이 특정 기준치를 넘을 때까지 큐에 데이터를 계속 저장합니다.
- 이벤트 발생 시, 큐에 저장된 텍스트가 중간에 올 때까지 대기합니다.

9.2 큐 데이터 처리 및 구조 문구 판별:

a. 데시벨 감지 이벤트 발생:

- 특정 기준치를 넘는 데시벨 이벤트가 발생하면, 큐에 저장된 텍스트를 array에 옮깁니다.

b. 트리거 판별:

- array에 옮긴 텍스트에서 등록된 트리거를 연속되지 않더라도 찾아야 하는 조건을 고려하여 판별합니다.
- 화자 인식 데이터와 함께 트리거를 판별하고, 구조 문구가 있는 지 확인합니다.

9.3 시스템 통합 및 테스트:

a. 데이터 처리 모듈 통합:

- 큐 관리 및 데이터 처리 모듈을 시스템에 통합합니다.

b. 트리거 및 구조 문구 판별 테스트:

- 특정 데시벨 이벤트가 발생하면 큐의 데이터를 옮기고, 등록된 트리거와 함께 구조 문구를 판별하는 테스트를 수행합니다.

10. 데이터 삭제 및 경찰서로 정보 전송 단계

10.1 데이터 처리 및 전송:

a. 데이터 존재 여부 확인:

- 구조 문구가 판별되었는지 확인하고, 해당 여부에 따라 데이터 삭제 또는 전송을 결정합니다.

b. 데이터 삭제:

- 구조 문구가 없다면 큐나 저장된 데이터를 삭제합니다.

c. GSM 모듈 통합:

- GSM 모듈을 시스템에 통합하여 실시간 정보 전송을 위한 기반을 마련합니다.

10.2 경찰서로 정보 전송:

a. 전송 데이터 준비:

- 전송할 정보를 GSM 모듈이 처리할 수 있는 형식으로 준비합니다.
- 필요한 정보는 감지된 사건의 내용, 위치 정보, 사용자 정보 등입니다.

b. GSM 통신:

- GSM 모듈을 활용하여 전송 데이터를 경찰서로 전송합니다.
- 안전한 프로토콜을 사용하여 데이터의 정확성과 개인 정보 보호를 고려합니다.

10.3 시스템 통합 및 테스트:

a. 데이터 처리 및 전송 모듈 통합:

- 데이터 삭제 및 경찰서로 정보 전송 모듈을 시스템에 통합합니다.

b. 데이터 삭제 및 전송 테스트:

- 구조 문구가 판별되었을 때, 데이터 삭제 및 경찰서로의 정보 전송이 올바르게 이루어지는지 테스트합니다.

이러한 단계를 통해 전체 시스템 테스트를 완료하고, 최종적으로 라즈베리파이 Pico 를 사용한 음성 기반 긴급 구조 장치를 안정적으로 운영할 수 있게 됩니다.

소형 사이즈 및 휴대성

1.1 전원 및 센서 장치의 소형화

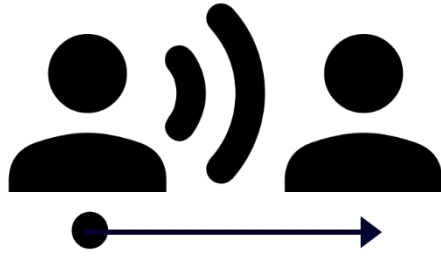
- 휴대용 긴급 구조 장치의 전원, 센서, 통신 장치 등이 소형화되어 휴대용 악세사리에 적절히 적합해야 합니다.

1.2 가벼운 무게

- 장치의 무게는 휴대폰 키링이나 벨트와 같은 악세사리에 부담을 주지 않도록 가벼워야 합니다.
-

평가 보고서를 기반으로 한 디자인 및 구현은 휴대용 긴급 구조 장치를 범죄자들로부터 쉽게 감지되지 않도록 보장할 수 있습니다.

장치와 라즈베리파이 간의 실시간 측정과 신고까지 소요되는 시간



위급 상황을 계산하고 근처 경찰서로 전송하는 시간은 여러 가지 요인에 의해 결정되기 때문에 정확한 값을 제공하기 어렵습니다. 그러나 일반적인 시나리오를 가정하여 계산 가능한 요인들을 고려해볼 수 있습니다.

긴급 상황 발생 :

긴급 상황이 발생하는 시점

음성 신호 수집 및 전처리 :

긴급 상황을 감지한 시스템이 음성 신호를 수집하고 전처리하는 단계 . 이는 음성 데이터의 품질을 향상시키기 위한 잡음 제거 , 특성 추출예상 시간 : 몇 밀리초에서 몇 초 .

긴급 상황 판별 및 분석 :

수집된 음성 데이터를 분석하여 긴급 상황 여부를 판별하는 단계예상 시간 : 몇 밀리초에서 몇 초 .

위치 확인 및 GPS 수집 :

긴급 상황 발생 지점의 위치를 확인하고 GPS 모듈을 통해 현재 위치를 수집하는 단계 .예상 시간 : 몇 초에서 몇 십 초 .

신고 및 통신 :

긴급 상황이 확인되면 해당 정보를 통신 모듈을 통해 관련 당국에 전송하는 단계 . 이는 휴대폰 네트워크 , 인터넷 또는 다른 통신 수단을 사용할 수 있다 .예상 시간 : 몇 초에서 몇 십 초 .

경찰서 응답 및 조치 :

관련된 당국 (경찰서 등)이 신고를 수신하고 적절한 조치를 취하는 단계 . 이 단계에서는 신고를 확인하고 응급 상황에 대응할 수 있도록 조치가 취해짐 .예상 시간 : 몇 분에서 몇 십 분 .

긴급상황 발생시점부터 신고 및 통신까지 걸리는 시간은 1 분내로 가능하다 판단.

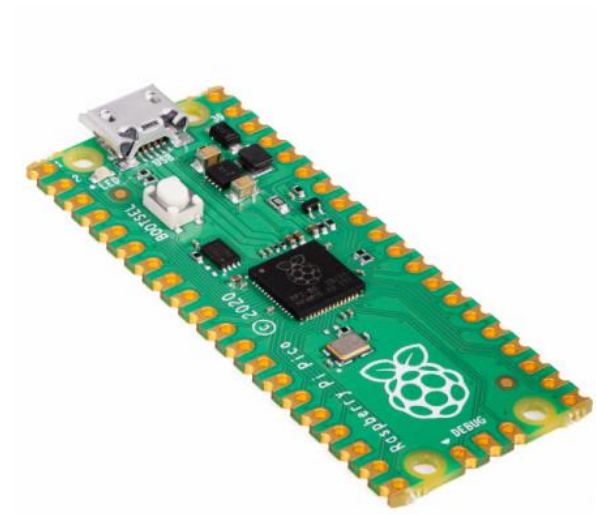
휴대폰으로 경찰서에 신고하면 신고자의 정보, 현재 위치, 상황 설명을 경찰쪽에서 요구하므로 2분 이상의 시간이 걸림.
따라서 구조요청 장치를 통해 다이렉트로 신고자의 실시간 위치와 구조요청 내용을 sms 를 통해 인근 경찰서에 신고하는것이 훨씬 빠름.
경찰서의 응답 및 조치에 걸리는 시간은 외부적 요인이므로 제외

지금 경찰 신고 시스템은 112 로 신고하면 자동으로 경찰서에서 신고자의 위치와 가까운 경찰서로 사건발생을 전달하는 시스템이다.
따라서, 긴급 구조 요청 장치가 스스로 가까운 경찰서를 찾는 방식이 아닌 일단 빠르게 긴급 상황을 감지하여 112 로 신고하는 방식에 초점을 맞춘다.

이러한 요인들을 종합적으로 고려하여 위급 상황을 감지하고 근처 경찰서로 전송하는 데 걸리는 시간을 계산할 수 있습니다. 예를 들어, 감지부터 경찰서로의 전송까지 총 시간은 1분 내로 가능합니다. 이 값은 최적화 및 실제 구현에 따라 단축될 수 있으며, 정확한 값은 제작 과정에서의 테스트 및 측정을 통해 얻어져야 합니다.

2)제품 구현 시 필요한 하드웨어 부품

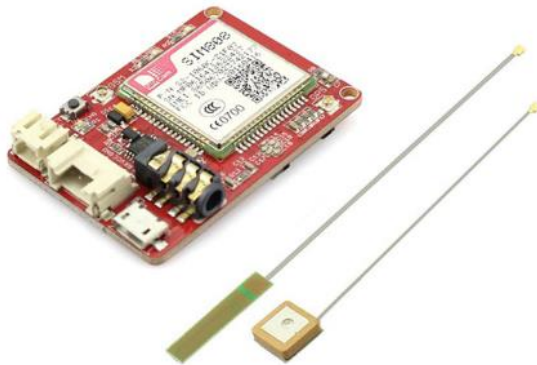
(1) 라즈베리파이 보드



라즈베리파이 피코(Raspberry Pi Pico)

5,400 원

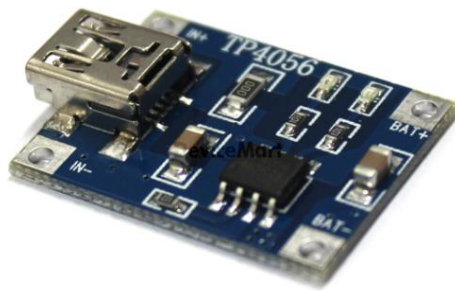
(2) GPS & 통신 모듈



[ELECROW] Crowtail-SIM808 GSM/GPS 모듈

46,400 원

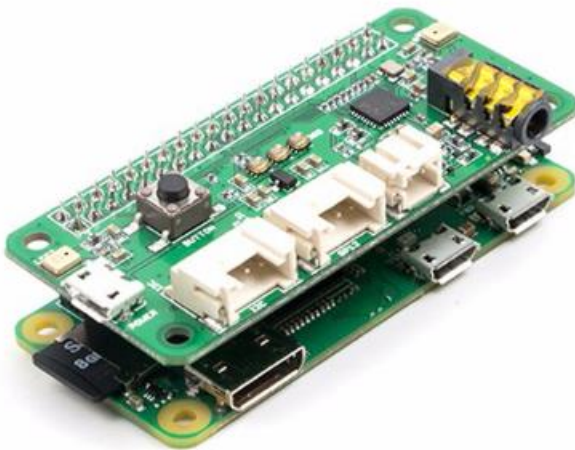
(3) 배터리 충전 모듈



[OEM] 리튬배터리 1A 충전모듈 (Mini-5P)

900 원

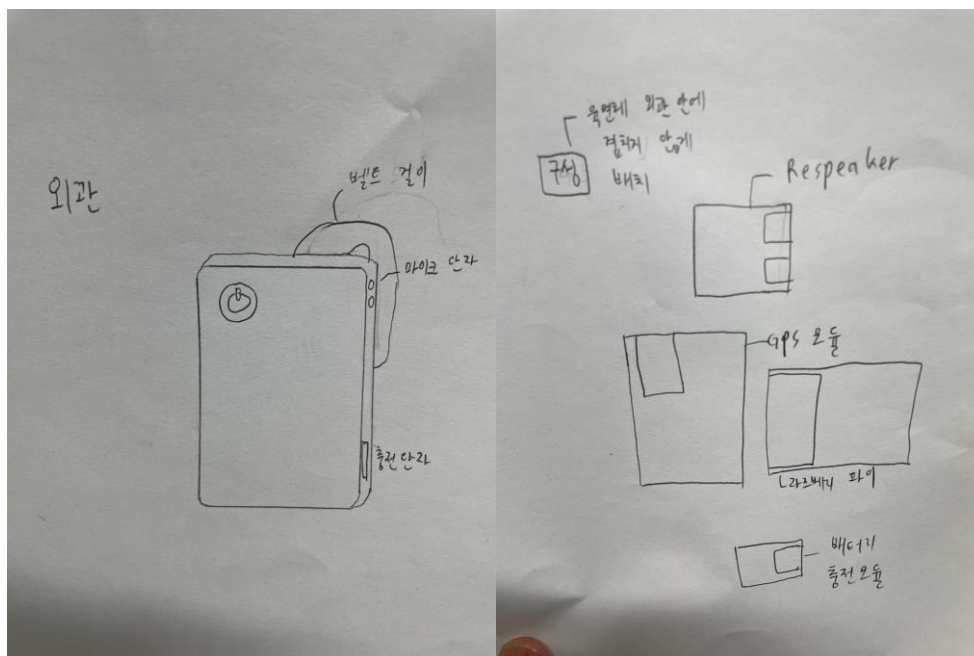
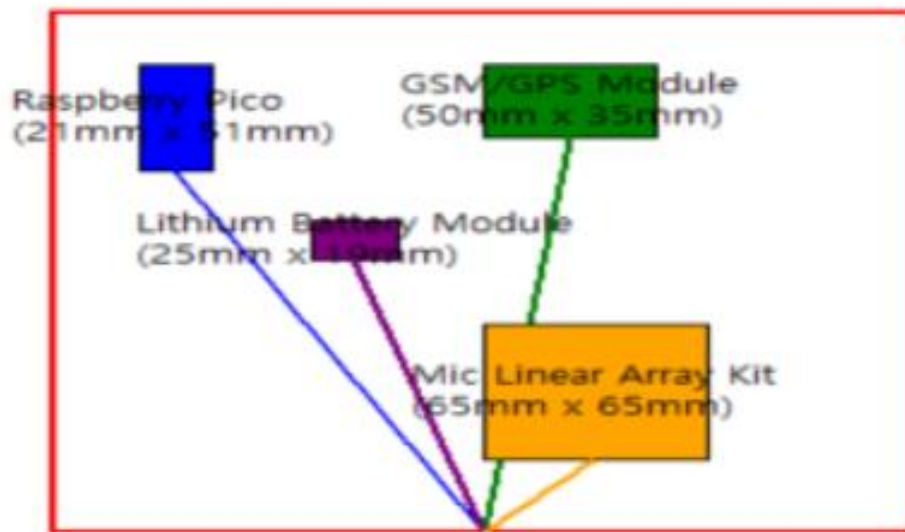
(4) 마이크로폰 센서



[Seeed] ReSpeaker 2-Mics Pi HAT for Raspberry Pi

17,500 원

* 제품 스케치



3. 팀원 간 업무 내용

길진성

회의록 작성
자재 구입
제품 기획
음성 데이터 수집
구조 상황 인지 학습 모델 개발

김현욱

제품 기획
음성데이터 수집
구조 상황 인지 학습 모델 개발
회로 구현 및 시나리오 동작 테스트

김태성

제품 기획
구조 상황 인지 학습 모델 개발
음성데이터 수집
회로 구현 및 시나리오 동작 테스트

정선진

회로 구현 및 시나리오 동작 테스트
구조 상황 인지 학습 모델 개발
음성데이터 수집
제품 기획

4. 작업 제작 추진 계획 및 일정

	1 주	2 주	3 주	4 주	5 주	6 주	7 주	8 주
	물품 구입 및 자세한 계획 수립 & 구조 장치의 하드웨어적 설계도(디자인)제작		회자 인식 코드 구현			음성데이터 수집		중간 고사 기간

	9 주	10 주	11 주	12 주	13 주	14 주	15 주	16 주
	음성 데이터 전처리 & 구조요청 인식 모델 학습				다양한 시나리오에서의 작동 여부 테스트 & 오류 수정			

5. 지원 경비 사용 계획

NO.	물품 이름	금액
1	<i>라즈베리파이 피코(Raspberry Pi Pico)</i>	5,400 원
2	<i>[ELECROW] Crowtail-SIM808 GSM/GPS 모듈</i>	46,400 원
3	<i>[OEM] 리튬배터리 1A 충전모듈 (Mini-5P)</i>	900 원
4	<i>[Seeed] ReSpeaker 2-Mics Pi HAT for Raspberry Pi</i>	17,500 원
5		
6		
7		
8		
9		

	합계	70,200 원
--	----	----------