

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.

• 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

-16 종의 응급상황을 음성/음향 데이터로 총 3,500 시간 이상 구축한

데이터 베이스를 이용하여 긴급 상황 인식 모델 개발

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 예로사항이 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 억울한 경우 또한 있습니다.

호신용품 잘못 쓰면 '특수상해' 처벌 받는다

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

호신용품이 흉기 됐다...신림동 대낮 성폭행 사건에 또 공포

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

5) 기존 제품들과의 차별성

자동 구조 요청 장치의 기존 제품들과의 차별성은 다양한 측면에서 나타납니다.

신속한 인지와 대응: 자동 구조 요청 장치는 범죄자의 습격으로부터 피해를 입은 피해자를 즉시 감지하고 구조 요청을 할 수 있습니다. 이는 기존의 수동적인 구조 요청 시스템과는 달리 피해자가 구조 요청을 할 필요 없이 자동으로 작동하므로 피해자의 신속한 구조를 가능하게 합니다.

특별한 사건 감지 기능: 자동 구조 요청 장치는 특정 사건이 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 & GSM을 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

2. 주제 구현방법 및 보조자료

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



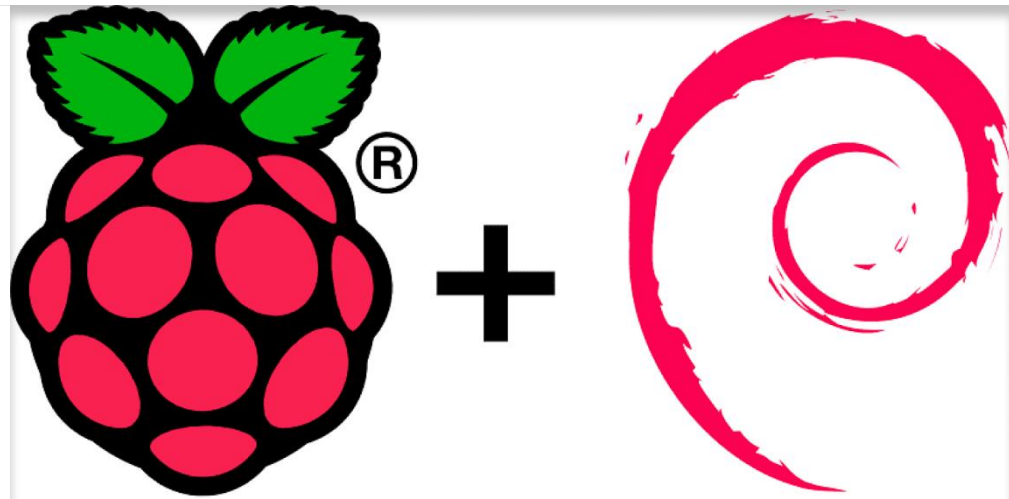
- 본 프로젝트의 목적은 긴급 상황에 돌입한 피해자가 장치의 별다른 조작 없이 자동으로 라즈베리 파이 피코를 활용하여 음성을 감지하고, 특정 구조 요청 음성을 통해 긴급 상황임을 판단, 신속하게 구조 요청과 위치 정보를 경찰서에 전송하는 소형 음성 기반 장치를 개발하는 것이다.
- 또한 범죄자들이 육안으로 구별해 쉽게 탈취 할 수 있는 휴대폰과 달리 구조장치라 생각할 수 없도록 키링이나 벨트와 같은 액세서리류로 위장하는 것이 궁극적인 목적이자 기존 구조 요청 장치들과의 차별점이다.

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.
- 긴급 상황에 대한 음성 데이터 베이스를 통해 신고 여부를 판단하는 모델을 구축한다



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.

5.2 휴대성 강화

- 경량 배터리를 사용하여 휴대성을 높이고, 키링 형태로 디자인하여 사용자가 편리하게 소지할 수 있도록 한다.

6. 전원 관리 및 안전성

6.1 효율적인 전원 관리

- 저전력 소비를 고려하여 장시간 사용이 가능한 효율적인 전원 관리 시스템을 도입한다.(리튬 이온 배터리)

6.2 안전성 강화

- 사용자의 개인 정보 보호 및 장치의 안전을 위해 보안 기능을 강화한다.

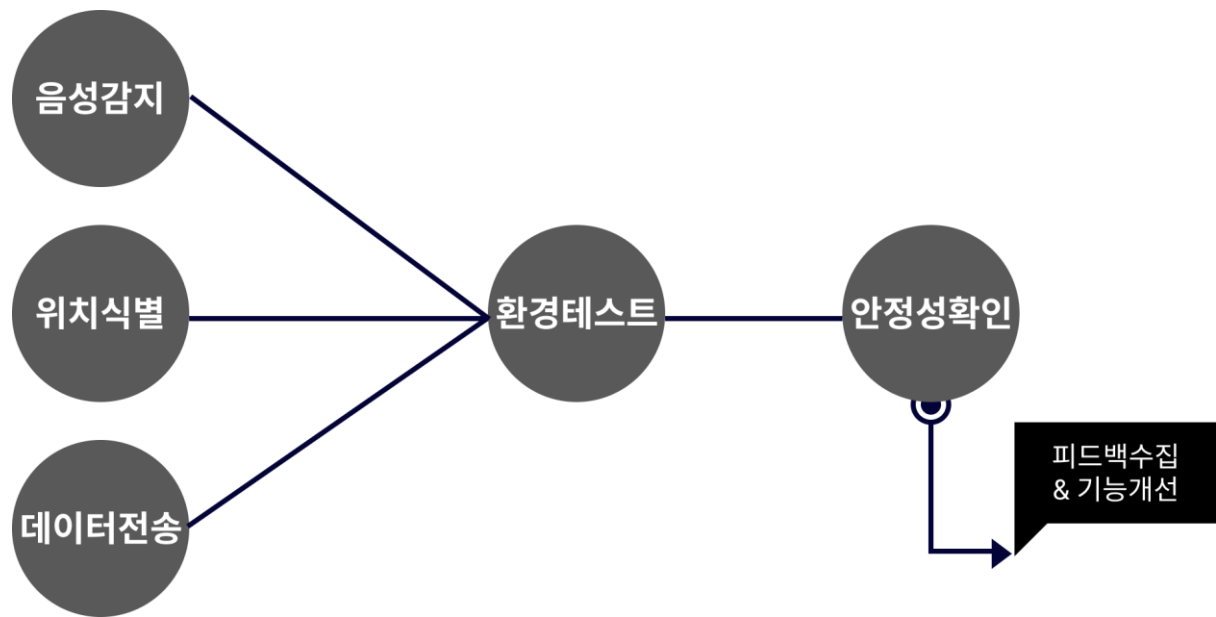
7. 테스트 및 개선

7.1 시뮬레이션 테스트

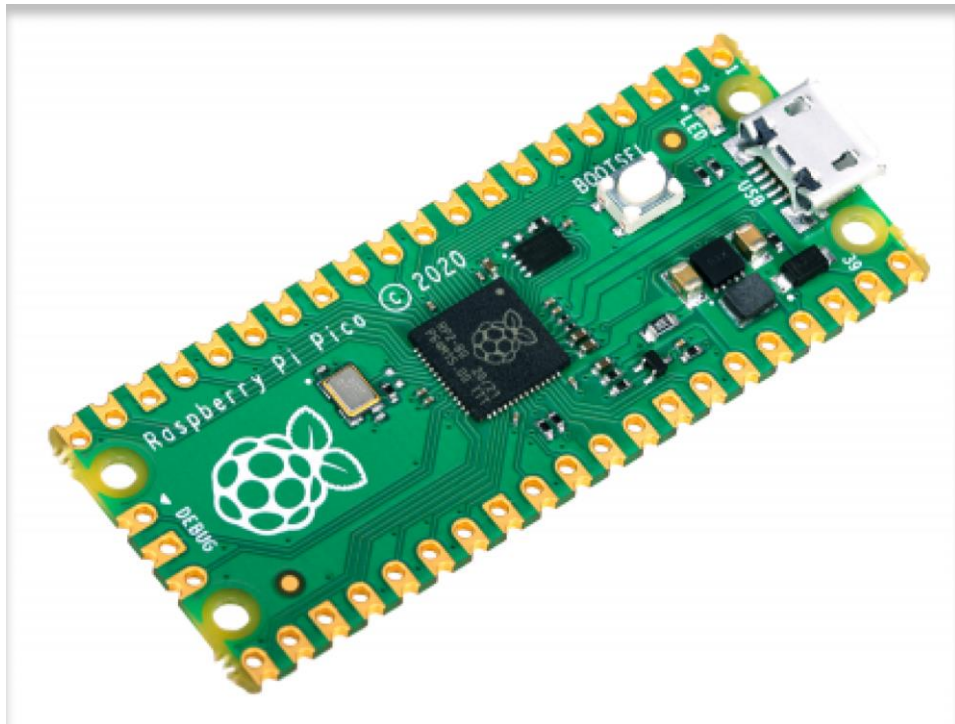
- 음성 감지, 긴급 상황 인식, 위치 식별, 데이터 전송 등의 기능을 시뮬레이션 환경에서 테스트하여 안정성을 확인한다.

7.2 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:

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

2. 경제성:

- 라즈베리 파이 Pico 는 비교적 저렴한 가격에 구매할 수 있습니다. 이는 프로젝트의 예산을 절감하면서도 안정적인 성능을 확보할 수 있는 장점으로 작용합니다.

3. 저전력 소비:

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

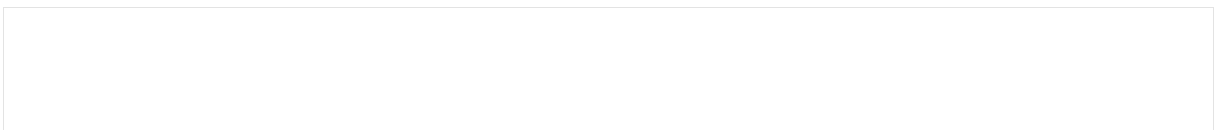
4. GPIO 및 다양한 센서 지원:

- 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.

5. 간편한 프로그래밍 환경:

- MicroPython 과 C/C++를 지원하는 Pico 는 간편한 프로그래밍 환경을 제공합니다. 이는 빠르게 프로토타입을 구축하고 개발을 진행하는 데 도움이 됩니다.

이러한 이유로 라즈베리 파이 Pico 를 선택하여 소형 음성 기반 위험 감지 및 신고 장치를 개발하면, 경제적이면서도 성능이 우수한 솔루션을 얻을 수 있습니다.



GPS 모듈 및 GSM 모듈 활용

1. GPS 모듈:

- **설명 및 원리:**

- GPS 모듈은 위성 신호를 이용하여 장치의 정확한 위치를 식별하는데 사용됩니다. 이 모듈은 위성과의 통신을 통해 경도, 위도 및 고도와 같은 위치 정보를 제공합니다.

- **활용:**

- 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:

- 사용자의 긴급 상황을 인식하면, 음성 감지 및 마이크로폰 센서가 작동하여 위험 상황을 확인합니다.
- GPS 모듈은 사용자의 현재 위치를 식별하고, GSM 모듈을 사용하여 위치 및 음성 감지 정보를 경찰서나 관련 당국에 전송합니다.
- 전송된 정보는 실시간으로 처리되어 긴급 상황에 빠르게 대응할 수 있도록 도움을 줍니다.

4. 소프트웨어 통합:

- 라즈베리 파이 Pico에는 MicroPython이나 C/C++와 같은 프로그래밍 언어를 활용하여 GPS 및 GSM 모듈을 통합하는 소프트웨어를 개발합니다.
- GPS 데이터 및 음성 감지 정보를 효과적으로 처리하고 MQTT 또는 다른 통신 프로토콜을 사용하여 데이터를 전송하는 소프트웨어를 개발합니다.

5. 테스트 및 디버깅:

- GPS 및 GSM 기능이 잘 작동하는지 확인하기 위해 테스트를 수행합니다. 장치를 실제 환경에서 테스트하고 사용자의 위치가 정확하게 전송되는지 확인합니다.
- 소프트웨어에서 발생할 수 있는 버그 및 이슈를 디버깅하고 안정적인 운영을 확인합니다.

이러한 GPS 모듈과 GSM 모듈을 통해 사용자의 위치를 정확하게 파악하고 긴급 상황에 신속한 대응을 가능하게 할 수 있습니다.

GPS 모듈의 정확도(신뢰성)

6. 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.
- 3.7V 리튬배터리로 전원을 공급하고 온보드 Lipoly 배터리 충전 회로가 있어 휴대용 모듈로 제작해 이동 중에도 기능을 수행할 수 있다.
- 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. AT 명령을 사용한 모듈 초기화:

- GSM 모듈은 AT 명령을 사용하여 제어됩니다. 초기화 단계에서는 모듈을 초기화하고 통신 설정을 구성합니다.

3. SIM 카드 삽입 및 인증:

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

4. **GSM 네트워크에 연결:**

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

5. **SMS 메시지 작성:**

- 경찰서에 보낼 SMS 메시지를 작성합니다. 이 메시지에는 긴급 상황을 나타내는 내용이 포함되어야 합니다.

6. **SMS 메시지 전송:**

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

7. **응답 처리:**

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

8. **통신 종료 및 모듈 릴리즈:**

- 통신이 완료되면 GSM 모듈과의 통신을 종료하고, 리소스를 해제합니다.

긴급 구조 요청 장치의 개발 단계

1. 화자 인식 모듈 선택

화자 인식을 위해 라즈베리 파이 Pico 에는 ReSpeaker 4-Mic Linear Array Kit 를 사용하기로 결정하였습니다. 이 모듈은 여러 마이크로폰을 포함하고 있어 화자의 위치를 정확하게 감지할 수 있으며, 뛰어난 음성 신호 처리 성능을 제공합니다. 아래는 모듈 선택에 대한 자세한 내용입니다.

1.1 ReSpeaker 4-Mic Linear Array Kit:

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.

1.2 선택 이유:

다양한 음향 환경 대응:

ReSpeaker 4-Mic Linear Array Kit 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 4-Mic Linear Array Kit 를 선택하여 라즈베리 파이 Pico 에 화자 인식 기능을 추가하게 되었습니다.

2. 마이크론 및 화자 인식 모듈 연결

라즈베리 파이 Pico 에 ReSpeaker 4-Mic Linear Array Kit 및 마이크론을 연결하는 단계는 하드웨어 구성을 마치는 중요한 부분입니다. 아래는 연결 과정에 대한 상세한 내용입니다.

2.1 필요 부품:

ReSpeaker 4-Mic Linear Array Kit

마이크론

Jumper wires

라즈베리 파이 Pico

필요한 도구 및 액세서리

2.2 연결 절차:

ReSpeaker 4-Mic Linear Array Kit 연결:

ReSpeaker 모듈의 I2S 인터페이스를 Pico 의 I2S 핀에 연결합니다.

전원 및 접지(GND) 핀을 연결하여 전원을 공급합니다.

필요한 경우, I2C 인터페이스를 연결하여 추가 기능을 활용할 수 있습니다.

마이크론 연결:

마이크론을 ReSpeaker 모듈에 연결합니다. 보통 마이크론은 PDM(파형 다중화) 출력을 가지고 있으며, 이를 모듈의 마이크 인풋에 연결합니다.

필요한 경우, 마이크론의 전원 및 GND 를 공급합니다.

라즈베리 파이 Pico 연결:

Pico 의 I2S 핀에 ReSpeaker 모듈을 연결합니다.

Pico 와 ReSpeaker 간에 공통 GND 를 설정하여 전기적으로 안정한 연결을 보장합니다.

전원 및 테스트:

전원 공급을 확인하고, Pico 를 컴퓨터에 연결하여 동작 여부를 테스트합니다.

테스트를 통해 마이크론이 ReSpeaker 모듈에서 정상적으로 음성을 수집하는지 확인합니다.

이렇게 하드웨어 연결이 완료되면, 라즈베리 파이 Pico 와 ReSpeaker 4-Mic Linear Array Kit 이 함께 동작하여 음성을 정확하게 수집할 수 있습니다.

3. 라즈베리 파이 Pico 에 필요한 라이브러리 설치

3.1 Pico SDK 설치:

Raspberry Pi Pico SDK 를 공식 GitHub 에서 다운로드합니다.
SDK 를 컴퓨터에 설치하고, Pico 를 연결합니다.
Pico 를 컴파일러와 함께 사용할 수 있도록 환경 변수를 설정합니다.

3.2 ReSpeaker 라이브러리 설치:

ReSpeaker 4-Mic Linear Array Kit 의 공식 GitHub 레포지토리에서
관련 라이브러리를 다운로드합니다.
필요한 파일을 Pico SDK 프로젝트에 추가합니다.

3.3 필수 의존성 설치:

Pico 에서 사용할 수 있는 I2S 인터페이스 관련 라이브러리 및
드라이버를 설치합니다.
필요한 경우, 추가적으로 ReSpeaker 모듈에 필요한 라이브러리를
설치합니다.

3.4 라이브러리 테스트:

ReSpeaker 모듈과 Pico 를 연결한 상태에서 예제 코드를 작성하고
컴파일하여 실행합니다.
음성 입력이 올바르게 인식되는지 확인하고, 화자의 위치가 정확하게
파악되는지 테스트합니다.

이렇게 설치된 라이브러리를 통해 라즈베리 파이 Pico 에서 ReSpeaker
4-Mic Linear Array Kit 을 활용할 수 있게 됩니다.

4. 화자 인식 코드 작성

라즈베리 파이 Pico 에서 ReSpeaker 4-Mic Linear Array Kit 을 이용한 화자 인식을 수행하는 코드를 작성하는 단계입니다. 아래는 코드 작성에 대한 기본적인 개요입니다.

4.1 예제 코드 작성:

Pico SDK 의 예제 코드를 참고하여 ReSpeaker 모듈과의 통합을 위한 기본 코드를 작성합니다.

화자 인식을 지원하는 라이브러리의 예제 코드를 참고하여 화자 식별 및 위치 추정에 필요한 코드를 작성합니다.

4.2 코드 수정 및 테스트:

작성한 코드를 수정하여 ReSpeaker 모듈과의 연동을 정확히 수행하도록 합니다.

예제 코드를 Pico 에 컴파일하고 실행하여, 화자 인식이 올바르게 작동하는지 확인합니다.

4.3 화자 식별 알고리즘 추가:

필요에 따라 화자의 식별 알고리즘을 추가하여 여러 화자를 구분할 수 있도록 코드를 확장합니다.

ReSpeaker 모듈이 제공하는 기능을 최대한 활용하여 화자의 특징을 추출하고 식별합니다.

4.4 화자 인식 모듈 튜닝:

화자 인식의 성능을 높이기 위해 모듈의 설정을 튜닝합니다.

노이즈 제거 및 화자 특징 추출에 관련된 파라미터를 조절하여 최적의 결과를 얻도록 합니다.

작성한 코드를 통해 ReSpeaker 4-Mic Linear Array Kit 을 사용한 화자 인식이 Pico 에서 정확하게 동작하도록 조정합니다.

5. 구조 요청 단어 인식 모듈 선택

라즈베리 파이 Pico 에 구조 요청 단어를 인식하는 모듈을 선택하는 단계입니다. 음성 명령을 텍스트로 변환하는 STT (Speech-to-Text) 엔진을 활용하여 특정 단어를 인식하는 기능을 추가할 수 있습니다. -PocketSphinx 선택

5.1 PocketSphinx 선택 이유:

경량화된 성능:

PocketSphinx 는 경량화된 디자인으로, 리소스가 제한된 환경에서도 효율적으로 동작합니다. 이는 라즈베리 파이 Pico 와의 호환성을 고려한 선택입니다.

제한된 어휘 지원:

프로젝트의 목적이 특정 구조 요청 단어에 대한 인식이므로, PocketSphinx 의 어휘 제한 기능이 적합하다고 판단했습니다.

개발자 커뮤니티:

PocketSphinx 는 활발한 개발자 커뮤니티를 가지고 있어서 문제 해결 및 지원이 용이하다는 장점이 있습니다.

5.2 PocketSphinx 적용 방법:

라이브러리 설치:

PocketSphinx 는 Python 에서 사용 가능한 라이브러리로 제공되므로, 라즈베리 파이 Pico 에 Python 환경을 구축하고 PocketSphinx 라이브러리를 설치합니다.

모델 학습:

PocketSphinx 는 모델 학습이 필요 없는 특징이 있습니다. 그러나 사용자의 발음에 대한 특화된 모델을 만들어 정확도를 높일 수 있습니다.

음성 명령 처리 코드 작성:

PocketSphinx 를 활용하여 라즈베리 파이 Pico 에 음성 명령을 감지하고 처리할 수 있는 Python 코드를 작성합니다. 이 코드는

PocketSphinx 라이브러리의 API 를 활용하여 음성 명령을 인식하고 이에 따른 동작을 수행합니다.

5.3 향후 개선 가능성:

사용자 정의 모델:

PocketSphinx 는 제한된 어휘를 기반으로 동작하지만, 사용자 정의 모델을 통해 더 나은 정확도를 얻을 수 있습니다. 이를 위해서는 추가적인 학습 데이터를 수집하고 모델을 조정하는 과정이 필요합니다.

다양한 언어 지원:

PocketSphinx 는 다양한 언어에 대한 지원이 가능하므로, 프로젝트의 확장성을 고려하여 다양한 언어에 대한 인식을 추가로 구현할 수 있습니다.

연결성 강화:

다양한 통신 모듈을 추가하여 더욱 강화된 연결성을 확보할 수 있습니다. 이는 음성 명령 인식에 대한 신속한 응답과 함께 경찰서로의 효율적인 신고 기능을 제공할 수 있습니다.

5.4 결론:

PocketSphinx 를 선택함으로써 경량화된 환경에서의 음성 명령 인식을 효과적으로 구현할 수 있습니다. 추가적인 모델 학습 및 다양한 언어 지원을 통해 프로젝트의 성능과 다양성을 높일 수 있습니다.

6. 음성 명령 데이터 수집

구조 요청 단어를 인식하기 위해서는 해당 명령에 대한 음성 데이터를 수집. 이는 다양한 사용자의 목소리 및 발음을 포함해야 합니다.

6.1 목적 및 데이터 정의:

목적:

음성 명령 데이터 수집의 목적은 사용자가 장치에 전달할 특정 구조 요청 단어에 대한 음성 패턴을 학습하는 것입니다.

데이터 정의:

수집할 음성 데이터는 다양한 사용자의 목소리, 발음 및 환경 소음을 포함해야 합니다. 또한, 구조 요청 단어의 다양한 발음 및 억양을 포함하여 모델의 일반성을 높이는 것이 중요합니다.

6.2 데이터 수집 도구 및 장비:

마이크로폰:

음성 명령을 수집하기 위해 고품질의 마이크로폰을 사용합니다. 마이크로폰은 주변 소음을 최소화하고 명령어를 명확하게 수집할 수 있는 품질이어야 합니다.

녹음 장비:

수집된 음성 데이터를 저장할 수 있는 녹음 장비가 필요합니다. 휴대용 녹음기나 컴퓨터와 연결된 외장 마이크로폰을 사용할 수 있습니다.

6.3 데이터 수집 절차:

다양한 환경에서 수집:

다양한 환경에서 음성을 녹음하여 실제 사용 환경을 반영합니다. 실내, 실외, 차량 내 등 여러 상황에서 녹음합니다.

다양한 사용자 포함:

여러 사용자의 목소리를 수집하여 모델이 다양한 발음 및 억양을 인식할 수 있도록 합니다.

목표 음성 명령 반복 수행:

구조 요청 단어를 목소리로 반복적으로 발음하며 녹음합니다. 각 사용자가 자연스러운 방식으로 명령을 전달할 수 있도록 유도합니다.

데이터 라벨링:

각 녹음 파일에 대해 해당하는 음성 명령을 라벨링하여 정확한 학습을 위한 데이터 세트를 구축합니다.

6.4 데이터 양과 품질 관리:

데이터 양:

충분한 양의 데이터를 수집하여 모델의 일반성과 정확도를 향상시킵니다. 최소 수백 건 이상의 음성 명령을 수집하는 것이 권장됩니다.

데이터 품질:

수집된 데이터의 품질을 보장하기 위해 노이즈, 에코, 왜곡 등을 최소화하고 고르게 녹음된 데이터를 선별합니다.

6.5 향후 개선 가능성:

실시간 데이터 수집 시스템:

실시간으로 사용자의 음성 명령을 수집하여 지속적으로 데이터를 갱신하는 시스템을 구축하면 모델의 성능을 지속적으로 향상시킬 수 있습니다.

백그라운드 노이즈 대응:

다양한 환경에서 발생하는 백그라운드 노이즈에 대응하는 효과적인 수집 전략을 도입하여 모델의 신뢰성을 높일 수 있습니다.

6.6 결론:

음성 명령 데이터 수집은 모델의 성능에 직접적인 영향을 미치는 핵심 단계입니다. 다양한 상황에서 다양한 사용자의 음성을 수집하고 품질을

보장함으로써, 구조 요청 장치가 효과적으로 음성 명령을 이해하고 처리할 수 있도록 합니다.

7. 음성 명령 데이터 전처리

라즈베리 파이 Pico 에 추가할 구조 요청 음성 명령 데이터를 전처리하는 단계입니다. 전처리는 데이터를 분석 및 학습에 적합한 형태로 가공하는 과정으로, 노이즈 제거, 특징 추출 등이 포함됩니다.

7.1 음성 명령 데이터 수집:

사용자가 장치에 전달할 음성 명령 데이터를 수집합니다.
다양한 사용자의 목소리와 발음을 포함하여 현실적이고 다양한 환경에서의 음성 데이터를 확보합니다.

7.2 데이터 라벨링:

수집한 음성 명령 데이터에 대해 각각의 라벨(정답)을 부여합니다.
각 음성 파일에 해당하는 구조 요청 단어를 라벨로 지정하여 학습 데이터를 구분합니다.

7.3 노이즈 제거:

수집된 데이터에서 노이즈를 제거하는 과정을 수행합니다.
주변 환경 소음, 마이크로폰 잡음 등을 최대한 제거하여 음성 신호의 클린한 형태를 유지합니다.

7.4 특징 추출:

음성 신호에서 학습에 활용할 특징을 추출합니다.
주파수, 에너지, MFCC 등의 특징을 추출하여 모델이 학습하기 적합한 형태로 변환합니다.

7.5 데이터 정규화:

추출된 특징을 정규화하여 모델이 안정적으로 학습할 수 있도록 합니다.
정규화는 특징 값의 범위를 조절하여 학습 알고리즘이 일관된 방식으로 학습하도록 돕습니다.

7.6 데이터 분할:

전처리된 데이터를 학습용과 테스트용으로 나누어 모델의 성능을 평가하기 위한 데이터셋을 구성합니다.

일반적으로 학습용 데이터셋과 테스트용 데이터셋의 비율은 8:2 또는 7:3 으로 설정합니다.

7.7 저장 및 준비:

전처리가 완료된 데이터를 저장하고, 모델 학습을 위해 라즈베리 파이 Pico 에서 활용할 수 있도록 준비합니다.

이렇게 전처리된 음성 명령 데이터는 모델 학습을 위한 준비가 완료된 상태로 사용될 수 있습니다.

8. 음성 명령 인식 모델 학습

라즈베리 파이 Pico 에 구조 요청 단어를 인식하는 모델을 학습하는 단계입니다. 선택한 음성 명령 데이터를 기반으로 모델이 구조 요청 단어를 인식하도록 학습시킵니다.

8.1 모델 선택:

학습에 사용할 모델을 선택합니다. TensorFlow, PyTorch 등의 프레임워크에서 제공하는 음성 명령 인식 모델을 선택

8.2 데이터 로딩:

전처리된 음성 명령 데이터를 모델에 로딩합니다.

학습 데이터는 입력(음성 데이터)과 출력(라벨 또는 구조 요청 단어)으로 이루어져야 합니다.

8.3 학습 설정:

학습을 위한 하이퍼파라미터(learning rate, epoch, batch size 등)를 설정합니다.

이 때, 모델의 복잡성과 학습 데이터의 크기에 맞게 적절한 설정을 고려합니다.

8.4 모델 학습:

설정된 하이퍼파라미터를 기반으로 모델을 학습시킵니다.

학습은 입력 음성 데이터를 모델에 주입하고, 출력이 실제 라벨과 일치하도록 모델의 가중치를 업데이트하는 과정을 반복합니다.

8.5 검증 및 튜닝:

학습된 모델을 검증 데이터셋을 사용하여 평가하고, 필요한 경우 모델을 튜닝합니다.

튜닝은 정확도를 높이기 위해 모델의 구조나 하이퍼파라미터를 조정하는 등의 과정을 포함합니다.

8.6 모델 저장:

학습이 완료된 모델을 저장하여 라즈베리 파이 Pico 에서 사용할 수 있도록 준비합니다.

8.7 모델 테스트:

저장된 모델을 활용하여 테스트 데이터셋이나 라즈베리 파이 Pico 상에서 테스트를 수행합니다.

모델이 구조 요청 단어를 올바르게 인식하는지 확인하고 성능을 평가합니다.

8.8 모델 배포:

테스트를 통과한 모델을 라즈베리 파이 Pico 에 배포하여 구조 요청 단어를 인식하는 기능을 추가합니다.

이렇게 학습된 모델은 라즈베리 파이 Pico 에 통합하여 음성 명령을 인식하고, 구조 요청 단어를 감지할 수 있도록 설정됩니다.

해당 데이터에 대하여 높은 신뢰도를 가진 공개 데이터를 찾았습니다.
따라서 구조 요청 음성 데이터와 화자 인식 데이터는 이것을 사용하도록 합니다.

AI-Hub

(<https://aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=data&dataSetSn=170>)-구조 요청 음성 데이터

(<https://aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=data&dataSetSn=537>)-화자 인식 데이터

9. 라즈베리 파이 Pico 에 음성 명령 인식 코드 작성

라즈베리 파이 Pico 에 구현된 음성 명령 인식 코드는 모델을 활용하여 실제로 음성 명령을 인식하고 처리하는 역할을 합니다.

9.1 모델 로드:

학습된 모델을 라즈베리 파이 Pico 에 로드합니다.

9.2 마이크로폰 및 입력 설정:

Pico 와 연결된 마이크로폰을 사용하여 음성 입력을 받습니다.

9.3 음성 명령 처리:

마이크로폰으로부터 받은 음성 데이터를 모델에 입력으로 전달하여 음성 명령을 추론합니다.

9.4 결과 출력 또는 액션 수행:

모델이 구조 요청 단어를 인식하면, 해당 이벤트에 대한 처리 또는 액션을 수행합니다.

예를 들어, GPS 와 GSM 모듈을 활용하여 경찰서로 구조 요청 SMS 를 전송하는 등의 동작을 구현합니다.

9.5 오류 처리 및 예외 상황 관리:

코드를 통해 오류 처리 및 예외 상황에 대한 적절한 처리를 포함하여 정확성을 높입니다.

9.6 반복 테스트:

음성 명령 인식 코드를 라즈베리 파이 Pico 에 반복적으로 테스트하여 신뢰성을 검증합니다.

9.7 최적화 및 성능 향상:

코드 실행 속도 및 자원 사용을 최적화하여 성능을 향상시킵니다.

9.8 외부 모듈 및 서비스 통합:

외부 모듈이나 서비스와의 통합을 진행합니다. 예를 들어, GPS 모듈, GSM 모듈과의 통합 작업을 추가로 진행합니다.

9.9 전체 시스템 테스트:

모든 기능이 통합된 전체 시스템을 테스트하여 음성 명령이 올바르게 처리되는지 확인합니다.

9.10 코드 배포:

테스트를 통과한 코드를 라즈베리 파이 Pico 에 배포하여 제품을 완성합니다.

이러한 과정을 통해 라즈베리 파이 Pico 는 음성 명령을 신속하게 처리하고, 구조 요청 단어를 인식하여 필요한 액션을 수행할 수 있는 장치로 완성됩니다.

10. 전체 시스템 테스트:

전체 시스템 테스트는 개발한 기능들이 통합되어 원활하게 작동하는지를 확인하는 중요한 단계입니다.

10.1 시나리오 기반 테스트:

다양한 시나리오를 기반으로 음성 명령 인식, 구조 요청 단어 처리, GPS 및 GSM 모듈 기능 등을 테스트합니다.

10.2 라즈베리 파이 Pico 의 자원 사용 테스트:

음성 명령 처리 중에 라즈베리 파이 Pico 의 자원(CPU, 메모리) 사용량을 모니터링하여 효율성을 평가합니다.

10.3 오류 시나리오 테스트:

음성 명령이 정상적으로 처리되지 않거나 오류가 발생하는 시나리오에 대한 테스트를 수행합니다.

10.4 응답 시간 및 신뢰성 테스트:

음성 명령을 입력한 후에서부터 경찰서에 구조 요청 SMS 가 전송되기까지의 응답 시간을 측정하고, 신뢰성을 검증합니다.

10.5 외부 환경에서의 테스트:

다양한 외부 환경에서 테스트하여 환경 변인에 대응할 수 있는지 확인합니다.

10.6 사용자 피드백 수집:

테스트 사용자들로부터 피드백을 수집하여 사용자 경험을 개선하는데 활용합니다.

10.7 보안 테스트:

시스템이 무단 접근으로부터 안전한지를 확인하는 보안 테스트를 수행합니다.

10.8 기능 추가 및 수정:

필요한 경우 기능을 추가하거나 수정하여 최종 제품을 완성합니다.

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

육안으로 식별 여부 확인

1. 위장성 평가
1.1 디자인 및 크기
<ul style="list-style-type: none">휴대용 긴급 구조 장치의 디자인은 휴대폰 키링이나 벨트와 유사하게 디자인되어야 합니다. 크기는 일반적인 악세사리와 유사하게 유지되어야 합니다.
1.2 재질 및 색상
<ul style="list-style-type: none">휴대용 장치의 재질과 색상은 휴대폰 악세사리들과 일치하도록 선택되어야 합니다. 이는 주변에서의 시각적 감지를 어렵게 만듭니다.
1.3 가능한 악세사리
<ul style="list-style-type: none">휴대폰 키링, 벨트, 팔찌 등과 같은 다양한 악세사리에 적용 가능하도록 설계되어야 합니다.
2. 소형 사이즈 및 휴대성
2.1 전원 및 센서 장치의 소형화
<ul style="list-style-type: none">휴대용 긴급 구조 장치의 전원, 센서, 통신 장치 등이 소형화되어 휴대용 악세사리에 적절히 적합해야 합니다.
2.2 가벼운 무게
<ul style="list-style-type: none">장치의 무게는 휴대폰 키링이나 벨트와 같은 악세사리에 부담을 주지 않도록 가벼워야 합니다.
3. 소프트웨어 통합 및 조작성
3.1 사용자 친화적인 인터페이스
<ul style="list-style-type: none">장치의 조작은 간단하며, 사용자가 휴대폰 악세사리로서의 주 기능 외에도 긴급 상황에 신속하게 반응할 수 있도록 설계되어야 합니다.
3.2 알림 및 상태 표시
<ul style="list-style-type: none">휴대용 장치가 작동 중임을 알리는 기능이나 LED 표시등 등을 추가하여 사용자에게 상태를 시각적으로 전달해야 합니다.
4. 사용자의 인지 가능성 테스트
4.1 모의 실험
<ul style="list-style-type: none">휴대용 긴급 구조 장치를 휴대폰 키링이나 벨트와 함께 모의 실험을 통해 여러 환경에서의 인지 가능성을 테스트합니다.
4.2 피드백 수집
<ul style="list-style-type: none">사용자에게 모의 실험을 통해 피드백을 수집하여 실제 상황에서의 인지 가능성을 향상시키는데 활용합니다.
5. 결론 및 향후 계획
5.1 평가 결과

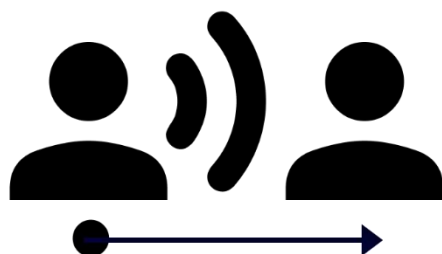
- 수행된 위장성 평가 결과를 종합하여 휴대폰 키링이나 벨트와 같은 악세사리로 위장된 휴대용 긴급 구조 장치의 성공 여부를 평가합니다.

5.2 향후 개선 사항

- 인지 가능성이 부족한 부분에 대한 개선 및 추가적인 테스트가 필요한 경우, 향후 계획을 수립하여 보완합니다.

위장성 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 2 분 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

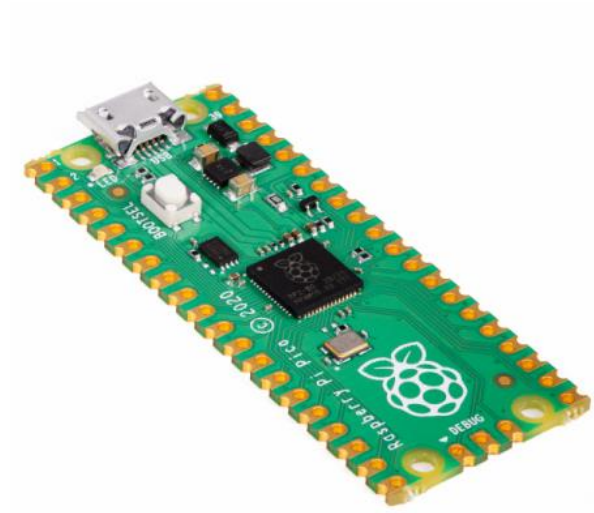
지금 경찰 신고 시스템은 112 로 신고하면 자동으로 경찰서에서 신고자의 위치와 가까운 경찰서로 사건발생을 전달하는 시스템이다.

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 1 분 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

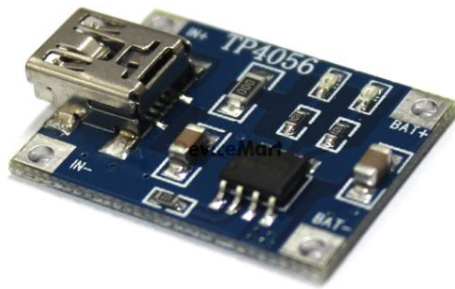
(2) GPS & 통신 모듈



[ELECROW] Crowtail-SIM808 GSM/GPS 모듈

46,400 원

(3) 배터리 충전 모듈



[OEM] 리튬배터리 1A 충전모듈 (Mini-5P)

900 원

(4) 마이크로폰 센서



[Seed] ReSpeaker 4-Mic Linear Array Kit for Raspberry Pi

40,400 원

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 4-Mic Linear Array Kit for Raspberry Pi</i>	40,400 원

5		
6		
7		
8		
9		
	합계	93,100 원