최종 결과 보고서

<5조>

201523479 송한솔

201220982 김동규

201220919 강정규

201523475 이아연

* 프로젝트 개요

- 시각장애인들의 보행을 도와주기 위한 스마트 지팡이

* 프로젝트 동기 및 목표

- 설문조사에 따르면 시각장애인들이 불편함을 겪는 일 중 보행 관련 상황이 많다는 것을

알게 되었고, 그러한 불편함을 줄여주기 위해 프로젝트 주제로 선정하게 되었고, 스마트 지팡이를 통해 보행 시 필요한 정보를 알려주어 위험한 상황을 최대한 줄이는 것이 목표이다.

* 분석

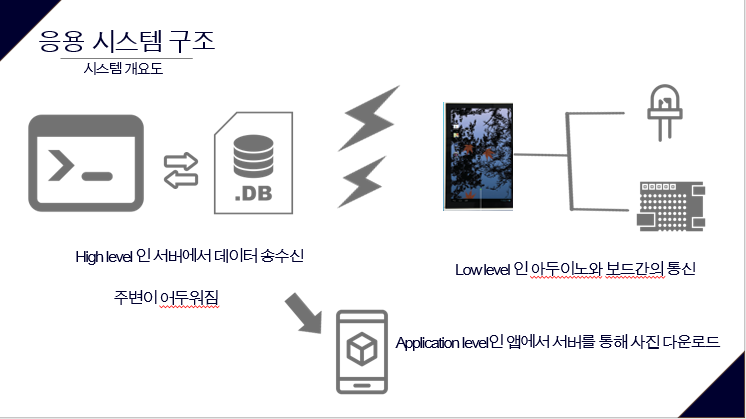
- 사용 시나리오

1. 사용자가 길을 보행 시 정면 혹은 측면의 장애물에 거리가 가까워지면 음성으로 장애물 반대 방향으로 보행하라고 알려주면 사용자는 그 방향에 따라 보행하여 장애물을 피한다.
2. 해가 지는 시간에 보행할 일이 있으면 지팡이에서 LED가 자동으로 켜져서 반대편에서 길을 걷는 타인이나 자동차 운전자가 시각장애인이 보행 중인 것을 인지하고 조심한다.
3. 사용자가 횡단보도 200m 이내 범위에 접근 시 신호등 쪽 송신 센서에서 신호등 데이터 전송을 하여 지팡이에서 신호등 데이터 수신을 한 뒤 사용자가 건널 방향 신호등의 신호 색깔이 무엇인지 음성으로 알려주면 사용자는 미리 인지하고 있다.
4. 사용자가 위험한 상황이나 위치를 카메라로 찍으면 보호자가 스마트폰을 통해 로드뷰처럼 볼 수 있어서 사용자가 처한 상황을 대신 대처한다.

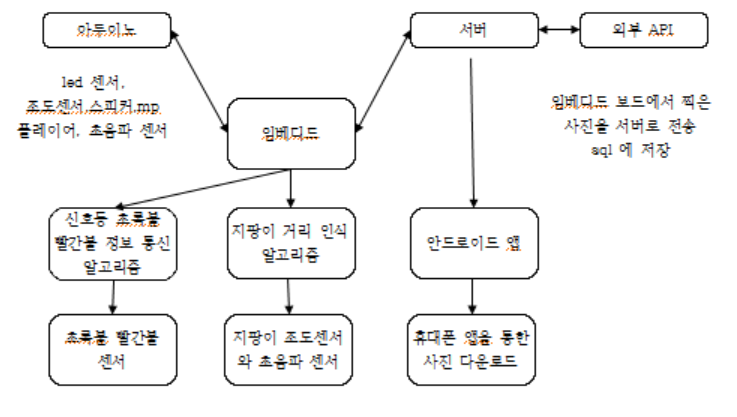
- 기존의 시스템과 차별점

1. 기존의 신호등 시스템과 연계하여 필요한 데이터를 활용하는 접근 방식
2. 사용자가 처한 위험한 상황이나 위치를 사진을 보호자 스마트 폰에서 볼 수 있는 기능

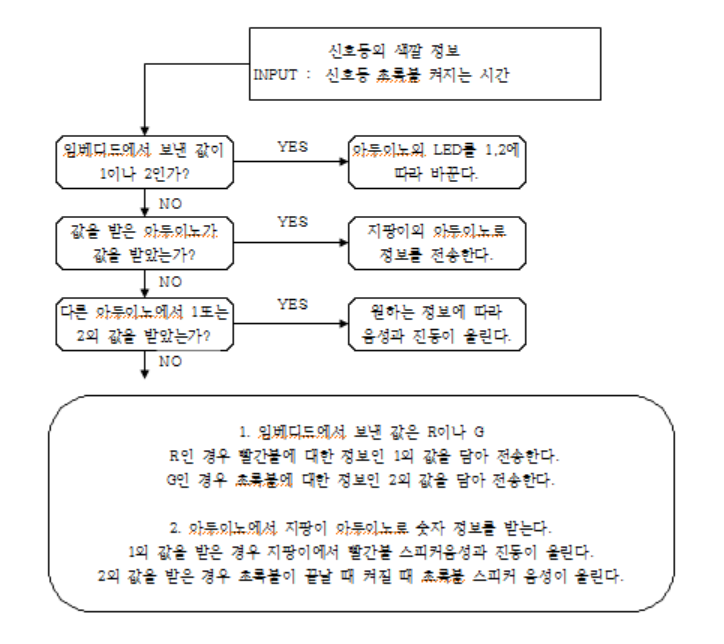
* 응용 시스템 제작 (설계 및 구현)
* 응용 시스템의 구조 (시스템 개요도)

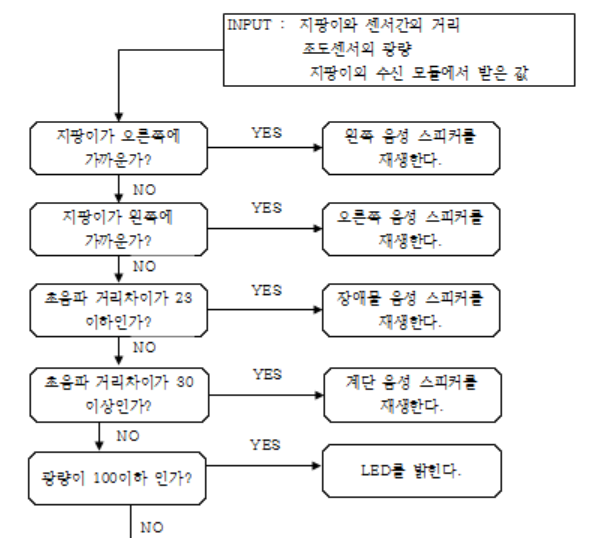


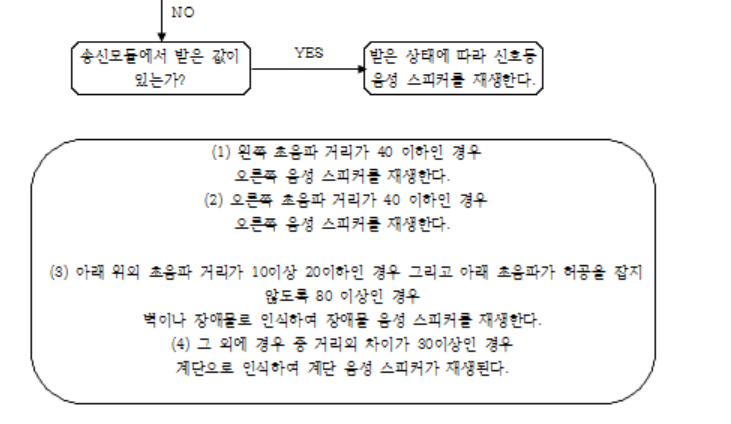
* 전체 알고리즘의 개요



* 세부 알고리즘의 개요 (코드 Copy는 제외!)







* 변동사항
* 기존 제안서 내용에서는 신호등의 불빛을 RGB 색 감지 센서를 통하여 감지 후 데이터를 전송하는 기능이었는데, 기존 신호등 시스템과 연계하여 신호등 데이터를 받아오는 방식으로 바꾸었다.
  + 이 제품이 상용화 되었을 때를 가정하여 바꾼 방식이 현실적으로 가장 실현 가능성이 높기 때문이라고 생각이 들었다.(비용 측면)
* 지팡이에 카메라를 부착하여 사진을 찍고 그 사진을 보호자의 스마트폰에서 볼 수 있는 기능 추가
  + 이 기능이 보호자가 걱정할 수 있는 부분을 해소시켜줄 수 있다고 생각하였기 때문이다.
* 시험평가
* 작성한 프로그램의 컴파일 환경 및 방법, 실행 방법 서술

1. 아두이노
   * 윈도우 or MAC 환경에서 아두이노 프로그래밍 전용 툴을 통하여 프로그래밍 후 업로드를 통해 아두이노에 코드를 업로드 한 뒤 아두이노에 전원을 연결하면 업로드한 코드로 동작한다.
2. 임베디드 보드
   * 리눅스 운영체제 환경(Ubuntu)에서 하나의 디렉토리에 실행파일을 생성할 때 필요한 C코드 파일 및 헤더 파일을 옮긴다.
   * 임베디드 보드에서 실행 가능한 실행 파일을 생성하기 위하여 크로스 컴파일러로 컴파일을 하여야 하므로 명령어 arm-none-linux-gnueani-gcc –pthread –o [생성할 파일이름] [필요한 소스코드파일]로 실행파일로 생성한다.
   * Tftp 를 통하여 보드에 업로드 한 뒤 Putty에서 ./실행파일이름 명령어 입력
3. 서버
   * 우분투서버에서 임베디드 보드와의 데이터 통신은 서버에서 gcc –o [생성할 파일이름] [C 소스코드 파일]로 컴파일 한 후 ./실행파일 명령어를 입력
   * 우분투서버에서 안드로이드와의 데이터 통신은 node server.js 명령어 입력
4. 안드로이드
   * 윈도우 및 mac 운영체제 환경에서 안드로이드 스튜디오 툴을 이용해 .xml, .java, .manifest 파일 3개를 실행하여 컴파일 한다.

* 테스트 결과에 대한 설명, Bug 및 개선할 사항
  + 테스트 결과

1. 지팡이의 초음파 센서가 장애물이 미리 설정한 범위 내에 있는 방향을 인식하여 음성으로 알려준다.
2. 지팡이의 조도 센서를 손으로 잡으면 빛 감지를 통해 빛이 적으므로 LED가 켜진다.
3. 임베디드 보드의 딥스위치 첫번째 스위치를 올리면 oled에 초록불 사진이 켜지고 bus led와 7segment에 남은 시간이 출력되고, buzzer에서 경각심을 주기 위한 소리가 난다. 앞의 디바이스들이 실행되면서 신호등 아두이노에 시리얼 통신으로 초록색 신호라는 데이터를 보내면 신호등의 초록색 LED가 자동으로 켜진다. 그 후 지팡이 아두이노에 아두이노간 데이터 통신으로 신호등 데이터를 송신하면 수신 받은 데이터를 통하여 음성으로 신호등 불빛의 색깔을 알려준다.
4. 임베디드 보드의 딥스위치 두번째 스위치를 올리면 oled에 빨간불 사진이 켜지고 신호등 아두이노에 시리얼 통신으로 빨간색 신호라는 데이터를 보내면 신호등의 빨간색 LED가 켜진다. 그 후 아두이노간 데이터 통신을 통해 지팡이에서 수신 받은 데이터를 통하여 음성으로 신호등 불빛의 색깔을 알려준다.
5. 임베디드 보드의 딥스위치 세번째 스위치를 올리면 임베디드 보드의 카메라가 작동하고 key matrix 2번 버튼을 누르면 사진이 찍히고 사진 파일이 서버에 전송된다. 취소하고 싶으면 1번 버튼을 눌렀을 시 카메라가 종료된다.
6. 안드로이드에서 버튼을 눌러 서버로 request를 보내면 서버로부터 사진을 다운 받을 url을 받아오고 이미지 다운 버튼을 누르면 자동적으로 다운로드가 된다.
   * Bug 및 개선할 사항
7. 사진파일을 전송할 때 파일을 분할해서 보내기 때문에 서버로 사진파일을 전송하는 시간이 wifi 상태에 따라 많이 달라진다. 이미지파일을 한번에 다 보내는 방식으로 개선해야 할 것 같다.
8. 지팡이의 초음파 센서가 모서리 부분에서 동작하면 crosstalk이 일어날 확률이 있다.

* 구현한 사항과 구현하지 못한 사항에 대한 설명 (타당한 이유 제시)
  + 구현한 사항

1. 시각 장애인들이 장애물을 인지하고 벽이나 계단을 인식하여 통행을 돕는 기능

2. 어두운 곳에서 타인에게 인지 시킬 수 있도록 LED를 켜주는 기능

3. 신호등에서 신호정보를 전송하여 지팡이를 통해 신호 정보를 소리로 인지 시켜주는 기능

4. 주변 환경을 지팡이의 카메라로 촬영을 하여 보호자의 스마트폰에 사진 전송해주는 기능

* Challenge Issue : 구현 상에서 어려웠던 점과 극복한 방안에 대한 설명

- 시각장애인을 위한 지팡이와 신호등의 기능을 보완할 수 있는 IOT 에 적합한 장비를 만들어야 하기 때문에 아두이노의 센서와 신호등으로 이용한 임베디드 보드 그리고 안간의 원활하고 정확한 소통과 위험상황에 대한 신속한 대처를 돕기 때문에 원하는 시간에 빠르게 센서와 서버 그리고 앱이 작동하는 것이 가장 중요 했다. 이를 해결하는 것이 가장 중점적인 구현상의 어려움이 되었다.

송한솔 : 위를 바탕으로 원하는 방식으로 구현을 하기 위해서 지팡이의 아두이노가 저희가 생각하는 신호등 센서와 보드와의 통신이 먼저 중요하다고 생각했습니다. 신속한 신호 전달을 주고 받을 수 있게 하고 싶었는데 처음에 그 방식이 잘 전달이 되지 않거나 한쪽이 느리게 받는 등 많은 문제를 일으켰습니다. 또한 센서의 값은 민감하게 작동하지만 실제로 보드에서 가져오는 값을 잘 얻지 못하는 경우가 생기기도 했습니다. 이를 해결하기 위해서 팀원들과 생각해 본 것은 먼저 보드에서 값을 여러 개 보내는 방식이었습니다. 여러번 값을 보내거나 딜레이를 주어서 해결이 되었습니다. 하지만 또 이러한 방식을 하다보니 원하는 값은 빨간불에 해당하는 값인 2나 초록불에 해당하는 값인 1을 각각 한번씩만 얻고 싶었지만 여러개를 얻어오게 될 수 있는 새로운 이슈가 생겼습니다. 이를 해결하기 위해서 아두이노에서 값을 가져올 때 조건을 걸어서 해당 조건에만 한번만 가져오게 할 수 있도록 수정해서 원하는 값을 한번만 얻어오는 신호등 정보를 정확히 한번씩 얻어올 수 있게 해결하였습니다. 또한 지팡이가 주된 기능이다 보니 막바지에는 지팡이 기능에 모두 심혈을 기울이고 정확도와 속도를 높이기 위해 코드를 쓰레드를 사용해 최적화 하거나 지속적인 테스트를 통해 해결하였습니다.

이아연 : 저희가 아두이노 지팡이를 쓰다 보니 보드의 기능을 많이 넣지 않았었습니다. 하지만 시스템프로그래밍 과목자체가 저희가 실습때 배운 기능이 너무 부족하다고 생각되어서 실습 때 사용한 기능을 가능하면 사용하려고 하다보니까 값이 충돌나서 느려지는 경우가 생기기도 했습니다. 이를 해결하기 위해서 쓰레드를 사용해 분산해서 원하는 속도에 모듈이 작동할 수 있게 코드를 수정하고 간단하게 코드를 수정하였습니다. 하지만 코드 수정을 모듈마다 원하는 방식을 다 해결하고 돌리고 나니 비슷한 문제점이 계속 나왔습니다. 이를 해결하기 위해 처음부터 다시 수정을 시작하여 되는 것부터 테스트를 진행해 잘 나오는지 확인하며 안되는 부분을 수정하면서 진행하니까 오류를 찾기가 더 쉬웠습니다. 그리고 아두이노 센서 기능에서 센싱이 잘안되는 경우 무엇이 잘못 된 것인지 알기 쉽지 않았는데 코드를 다시 넣어보기도 하고 선을 다시 잘잡아보기도 하였습니다. 그런데 결국 핀을 꼽는 부분에 문제가 있어 우리가 구현한 코드가 돌아가지 않는 것을 여러 시행착오 끝에 깨닫게 되었고 그를 통해 아두이노 센서가 잘 작동하지 않았던 것들을 해결하였습니다. 또한 실제로 지팡이 기능을 구현하면서 원하는 위치로 갈 때 초음파 거리가 정확히 측정기준대로 움직이지 않았습니다. 이를 해결하기위해 지팡이 계단/벽 알고리즘을 조금 바꿔서 오차범위를 두고 그 위치를 벗어나는 어느정도 부분에서 돌아가게 수정하여 통행보조를 위한 값을 원하는 방식대로 얻어오게 해결하였습니다.

강정규: 보드의 기능과 지팡이의 기능을 중점적으로 두고 프로젝트를 진행하였지만 보호자들의 관점에서 시각장애인의 위험상황을 알려줄 수 있는 것도 좋은 것이라고 생각 했습니다. 그래서 서버를 통해 보드에서 보낸 사진을 앱으로 보내 이를 해결하려 하였습니다. 그런데 실제로 사진을 보드로 찍는 과정자체가 복잡하고 와이파이 연결의 문제로 잘 해결이 되지 않기도 하였습니다. 사진을 전송하는 기능을 사용하려면 일단 사진이 잘찍히고 서버로 전송이 되어야 하는데 사진을 찍고 전송하는 과정에서 그 속도도 느리고 와이파이 연결이 잘잡히지 않아 고생을 많이 하였습니다. 이를 해결하기 위해 자주 쓰는 와이파이를 설정해서 사용하고 잘 잡히는 와이파이를 사용해서 전송이 잘 되지 않는 문제점을 해결하였습니다. 그리고 나니 문제점은 서버였습니다. 서버를 구축하여 앱으로 URL을 전송하는 과정에서 서버가 한번 작동이 아예 되지 않았습니다. 열심히 만든 서버인 만큼 어떻게든 살려보려고 밤을 새서 원인을 찾아 내려고 하였습니다. 하지만 원하는 결과는 얻을 수 없었고 새로 서버를 모두 구축하여 이 문제를 해결할 수 밖에 없었습니다. 또한 서버를 통해 사진을 전송하는 과정에서도 원하는 사진을 바로 받아오는 것이 되지 않았습니다. 분명 똑같은 코드를 통해 작업을 진행하였음에도 불구하고 원하는 사진이 전송되지 않거나 깨지기도 하고 많은 낙심을 했던 부분이 들었던 부분이었습니다. 이를 해결하기 위해 사진을 비트 단위로 보내거나 퍼블릭으로 보내 사진을 받는데 무리가 없게 하려는 노력을 하였고 서버의 과부하를 막기 위해 여러번의 전송 작업과 코드를 조금이라도 단순화해 속도를 높이고 오류를 줄여 원하는 사진을 얻어올 수 있게 수정하여 해결하였습니다.

김동규: 저는 아두이노 지팡이의 기능과 앱을 이용한 보호자가 주변 사진을 받아볼 수 있는 기능을 구현하였습니다. 먼저 아두이노 간 통신을 통해 신호등 정보를 얻어오는 과정에서 분명 신호등 값을 보냈지만 원하는 값을 지팡이 아두이노에서 가져오지 못하는 결과가 나왔습니다. 수신 모듈과 송신 모듈 자체가 라디오 통신을 하기 때문에 원인을 쉽게 찾을 수 없었고 팀프로젝트를 통해 수신모듈선이나 보드 코드 송신모듈선 모두 수정을 해보았고 많은 시행착오 끝에 실제로 해결방안은 수신모듈 핀의 문제였습니다. 그래서 다른 핀에 꽂아 이문제를 생각보다 쉽게 해결할 수 있었지만 하드웨어적인 이러한 문제는 실제로 간단한 문제이지만 원인을 쉽게 알 수 없어 힘들게 해결했던 부분이었습니다. 또한 스피커를 MP 플레이어로 재생하게 하는 문제에서 스피커 자체가 많이 예민하고 저항이나 선 문제인 경우도 많습니다. 그런데 모든 해결방안을 인터넷을 통해 찾아봤지만 해결할 수 없었습니다. 선을 거의 한봉지를 다썻지만 해결하지 못했는데 우연히 보조베터리를 이용하면 전력공급이 잘되어서 잘될 수 있다는 조언을 듣고 보조배터리를 전력으로 이용해 해결할 수 있었습니다. 너무 많은 말썽을 부렸던 스피커여서 팀원들과 가장 많은 소통과 노력을 기울였던 과정이었던 것 같습니다. 또한 앱을 통해 URL을 서버로부터 받아 자동으로 사진이 갤러리에 저장되게 하는 앱을 이성하는데 있어서도 서버에서 URL을 아무리 보내도 잘 전송이 되지 않은 문제가 있어서 처음에는 아예 비트맵 자체를 비트로 받아 저장하기도 하고 사진 전체를 디비에 저장하려고도 했었지만 모두 실패하고 결국 URL을 주고받아 웹서버에서 그 사진을 직접 전송받는 방법으로 해결하였습니다. 모든 과정은 복잡하고 힘들었지만 여러 방법을 고안해내서 해결해 낼 수 있었던 이슈였습니다.