**컴퓨터 응용 설계 및 실험**

**텀 프로젝트**

**4차 보고서**

2020.07.02

인체감지 센서(적외선 센서)와 초음파 센서를 이용한 침입자 확인 시스템

1조

201524582 정희석

201424458 박인철

**목차**

* 프로젝트 주제
* 프로젝트 목표
* 사용 센서 및 모듈
* 구현 예상도
* 구현 사진
* 진행 과정
* 역할 분담
* 실행 결과
* 느낀 점

**1. 프로젝트 주제**

- 초음파 센서와 적외선 인체감지센서를 이용한 방문자 확인 및 위치 추정 시스템

주제선정 이유: 현재 배운 것으로 최대한 가능한 것은 초음파 센서를 이용하고 드라이버를 만드는 것을 배웠으므로 인체감지센서 드라이버를 만들고 WiFi 소켓통신을 이용하여 서버-클라이언트 통신을 이용하여 Host PC에 결과를 출력하도록 할 수 있다. 현재 가지고 있는 Display 출력 문제를 해결하면 카메라 모듈로부터 LCD에 영상을 출력할 수 있을 것 같아서 이 주제를 선정하게 되었다.

**2. 프로젝트 목표**

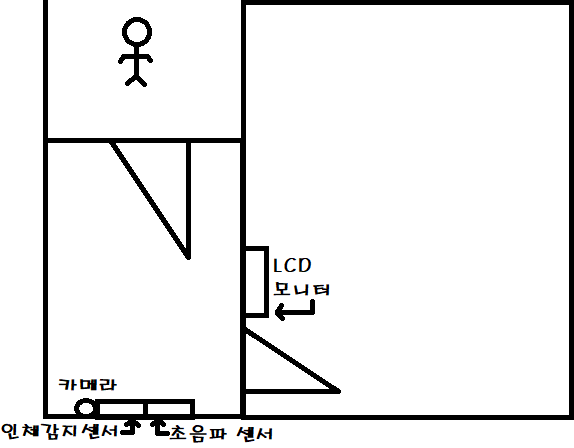
실험 시간에 배운 ACHRO보드 상에서 GPIO를 사용하여 센서를 부착하고 이를 Linux 운영체제에서 조작하며 WiFi 모듈을 통한 소켓 통신을 사용하여 센서의 결과를 보드의 Linux에서 처리하여 소켓 통신을 통해 Host PC에 출력하는 것이 1차 목표이고 최종 목표는 센서 값이 들어온다면(침입이 확인된다면) 보드의 카메라 모듈에 의해 촬영된 것을 디스플레이에 실시간으로 출력하여 확인이 가능하도록 하는 것이다.

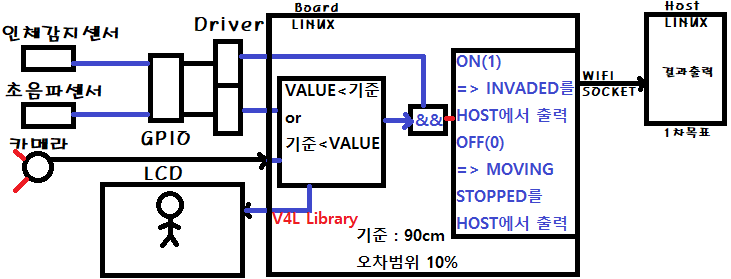
**3. 사용 센서 및 모듈**

초음파 센서, 인체 감지 센서, 카메라 모듈, (ON BOARD)LCD를 사용하였다.

초음파 센서와 인체 감지 센서를 GPIO를 통해서 연결, 각 센서의 드라이버를 구현하고, 카메라 모듈을 위해 OpenCV를 설치, V4L Library를 사용, LCD를 위해 QT Library를 설치하여 사용하였다.

**4. 구현 예상도**

****

****

**5. 구현 사진**

**테이블, 앉아있는이(가) 표시된 사진

자동 생성된 설명**

**ー 카메라 모듈 ー 초음파 센서 모듈 ー 인체 감지 센서 모듈 ー LCD**

**6. 진행 과정**

|  |  |
| --- | --- |
| 목표 날짜 | 구현 내용 |
| ~6/2 | 초음파 센서 처리 => 완료 |
| ~6/9 | 인체 감지 모듈 드라이버 및 처리 => 진행중  Wifi 소켓 통신 구현 => 완료  (소켓 통신 + 초음파센서 진행중)  1차 목표 테스트 => 인체 감지 모듈 미구현으로 미시행 |
| ~6/16 | Display에 카메라 화면 출력(Display 문제 해결)  -> Opencv 설치 과정에서 오류 발생 확인  (소켓 통신 + 초음파센서 동작 확인 완료)  인체 감지 모듈 구현 중 |
| ~6/23 | Display에 카메라 화면 출력(Opencv 오류 해결)  ->카메라 동작 확인 완료  인체 감지 모듈 구현 중 |
| ~6/30 | 카메라 코드 분석 및 프로젝트에 이식 완료  WIFI 접속 외부망 -> 내부망 작업 완료  초음파 센서 값을 통한 카메라 동작 구현 완료 |
| ~7/1 | 인체 감지 센서 드라이버 구현 완료 및 프로젝트에 이식 완료  최종 목표 테스트 |
| ~7/2 | 최종 보고서 작성 |
| 7/3 | 최종 발표 및 시연, 프로젝트 제출 |

(1차 보고서 때 상황)

6/1 월요일 연구실 방문을 통해 인체 감지 모듈을 수령하였으며 현재 4주차의 GPIO 드라이버를 통한 센서 조작을 해보고 있습니다. 이를 소켓통신을 통해서 host PC에 값을 출력하는 것이 6/5일까지의 목표로써 진행하고 있습니다. 초음파 센서의 드라이버와 동작 구현이 완료되면 인체 감지 모듈의 드라이버를 만들고 동작을 확인하는 것이 다음 목표이며 현재 QT Creator를 설치 및 사용 테스트 중입니다.

(2차 보고서 때 상황)

6/3 QT Porting Library, QT Creator 설치 완료 및 디스플레이에 시계 예제 출력 완료

6/5 초음파 센서 드라이버 및 동작 구현 중

6/6 WiFi 소켓 통신 구현 -> ACHRO보드: 서버, HOST 컴퓨터: 클라이언트

=> 이유: HOST 컴퓨터는 VM 가상머신 상에서 동작 하므로 WiFi 소켓 서버를 올려서 동작하도록 하려면 각종 설정이 필요. VMware Workstation 15에서 진행하기 어렵다 판단했기 때문.

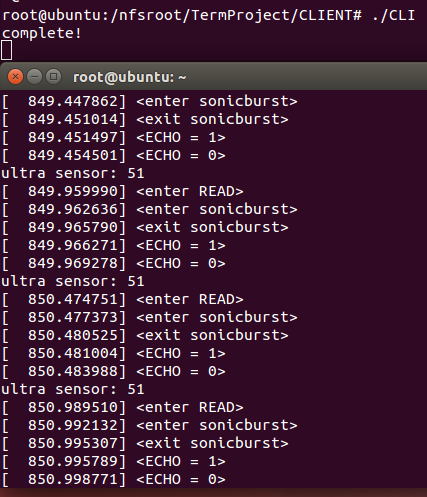
6/7 초음파 센서 드라이버 완료 및 WiFi 소켓 통신과 센서 동작을 하나의 프로그램으로 작성 중

(3차 보고서 상황)

<정상 상태>



>클라이언트에 아무것도 안 뜸 -> 설정 50cm

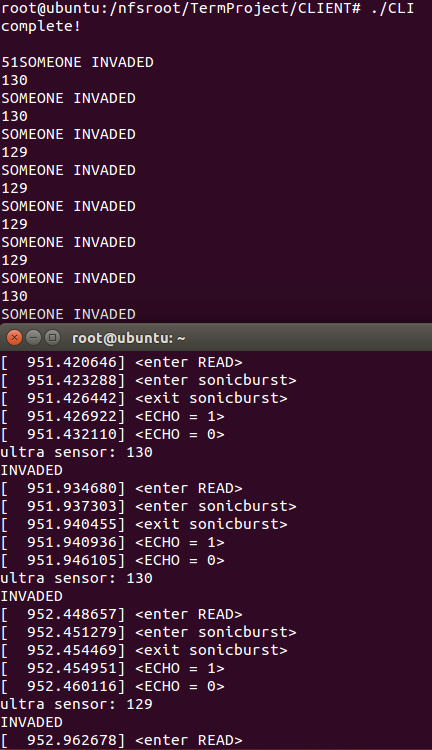


> 보드에서는 확인용으로 거리를 출력하게 함

<도난 상태>



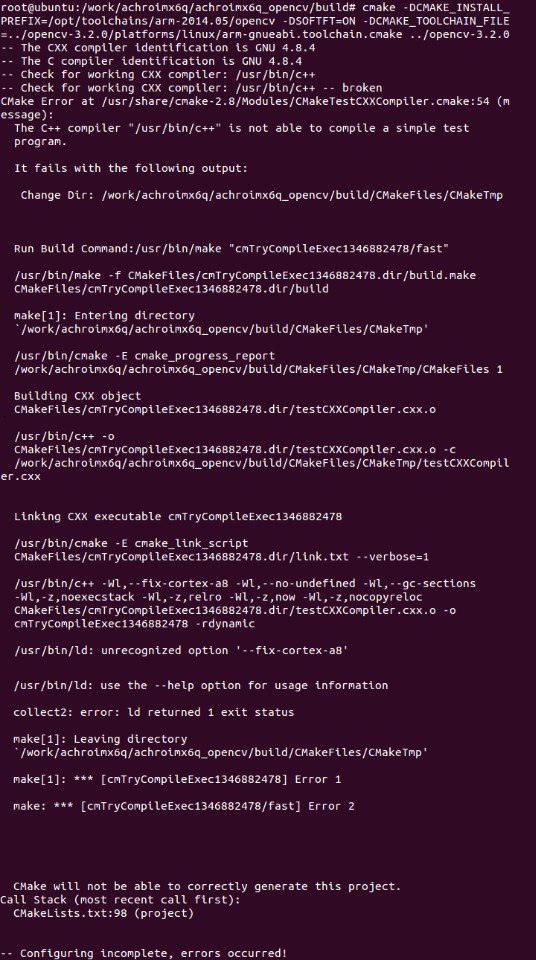
>클라이언트에 거리가 뜨면서 알림을 출력



>보드에는 센서 값과 알림을 확인을 위해 출력하게 하였다.

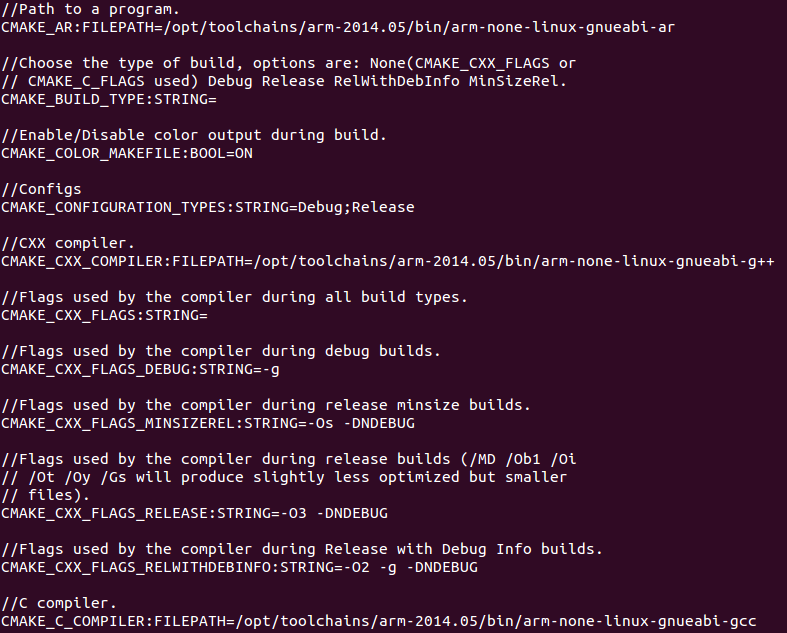
아직 개발 중으로 우선 ultrasonic의 값을 wifi를 통해서 클라이언트에서 출력이 가능함을 확인하는 과정을 진행하였다. 그리고 카메라 모듈을 위한 Opencv의 설치에서 cmake시 발생하는 unrecognized option “–fix-cortex-a8”오류가 있어서 이를 확인하고 있다.

(4차 보고서 상황)



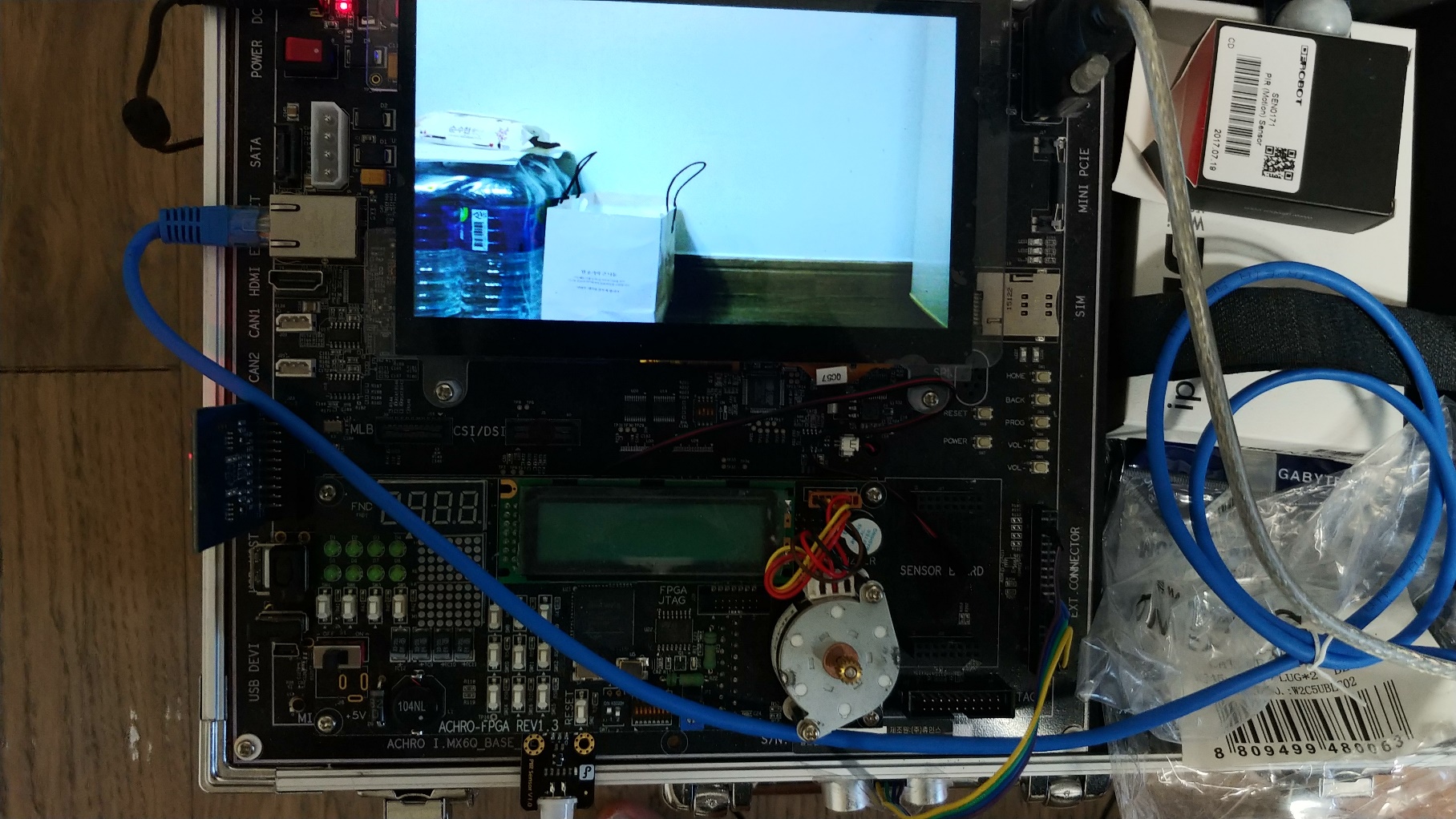
<오류 내역>

cmake시 발생하는 unrecognized option “–fix-cortex-a8”오류가 발생한 원인이 compiler의 위치를 제대로 잡지 못해서 발생하였다. 이를 해결하기 위해 파일을 찾아보다가 build내에 생성된 CMakeCache.txt파일에서의 CXX Compiler, C Compiler, AR의 위치를 재설정해주었다.

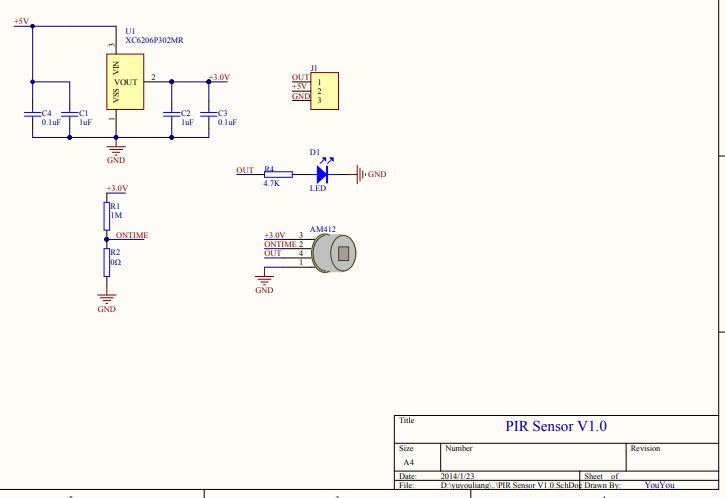


<CMakeCache.txt에서 수정한 부분>

그리고 ppt의 20페이지에 cp -a /usr/local/lib/\* . 에서 opencv의 내부 파일이 없어서 찾아보니 /work/achroimx6q/achroimx6q\_opencv/build/lib에 있어서 이 파일들을 nfs를 이용하여 보드로 옮겼다.

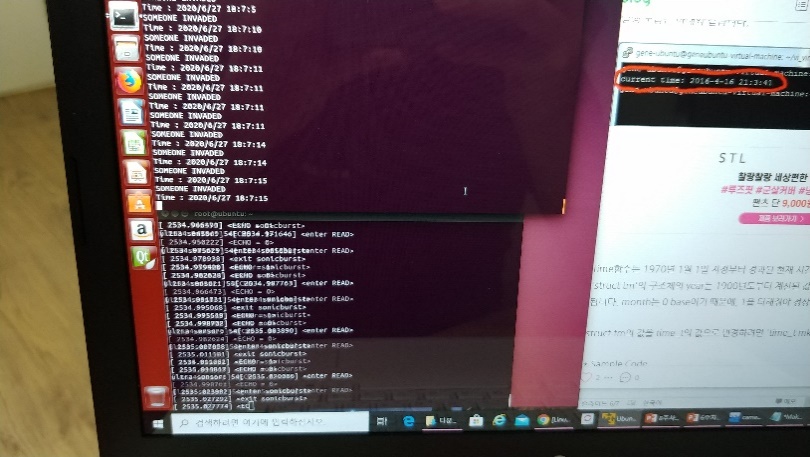


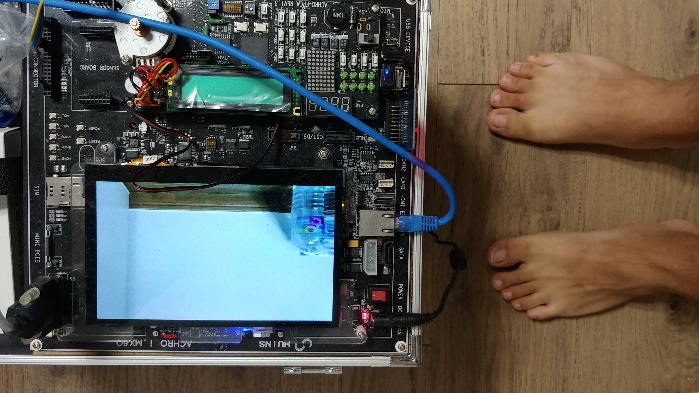
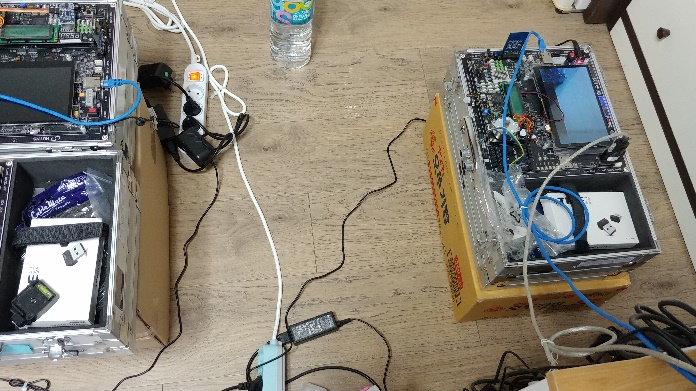
> 8주차 실험자료인 OpenCV를 이용하여 디스플레이에 카메라화면을 띄우는 camera\_test의 실행에 성공하였다. 그리고 새로 받은 센서 SEN0171의 회로도를 구했다. 이 센서는 GPIO2\_4(30)에 OUT, 5V(8)에 +, GND(38)에 -를 연결할 계획이다.



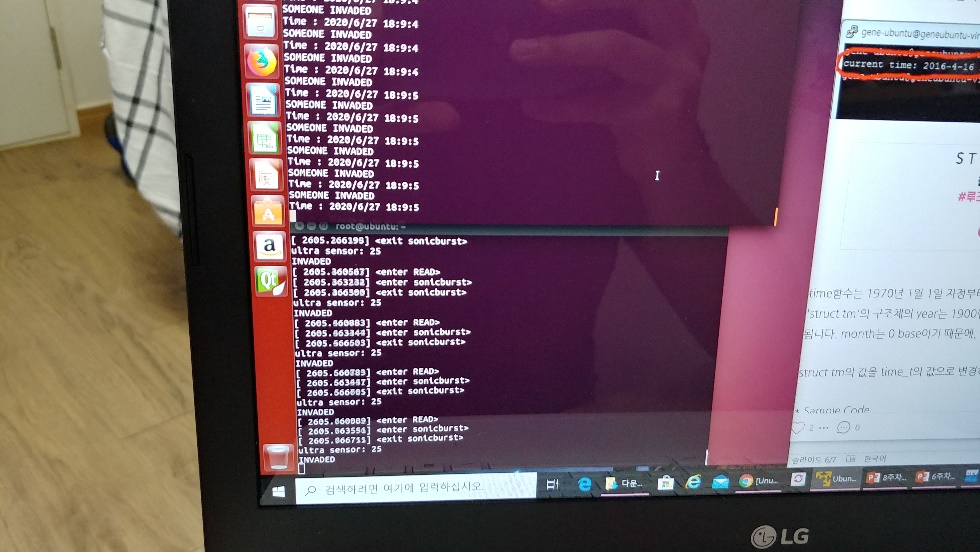
이와 DVD1-SRC의 gpio\_driver의 예시를 분석하여 이 인체감지 센서의 드라이버를 만들 계획이다. 문제는 이 센서를 연결한 점프 케이블이 없어서 주중에 연구실을 방문하여 bread-board와 케이블을 대여할 계획이다.

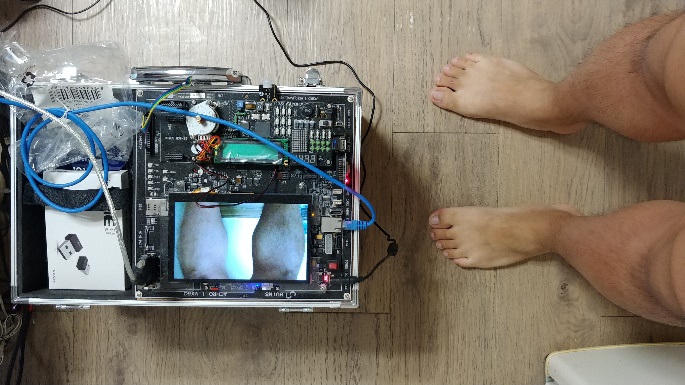
(5차 보고서 상황)

 <-센서 값이 정상범위(40~60) 값은 54,55가 나온다



-> 카메라가 동작하지 않는다.

-> 센서 값이 낮음 -> 중간에 무언가 있는 경우



-> 중간에 물통이 있어서 카메라가 동작

이제 인체감지센서의 드라이버를 만들어서 초음파 센서의 값이 바뀐 후 인체감지센서의 값을 읽어 들여서 카메라를 출력하게 하여도 되고 지금처럼 초음파 센서의 값이 바뀌면 카메라를 동작 시키고 인체감지 센서가 인식하면 host에 wifi로 침입여부를 알려줄 수 있을 것 같다.

(최종 보고서 상황)

**테이블, 앉아있는이(가) 표시된 사진

자동 생성된 설명**

5차 보고서 이후 연구실을 방문하여 점퍼 케이블과 Bread-Board를 대여하여 초음파 센서의 위치를 카메라의 위치로 옮기고 인체 감지 센서와 함께 테이프로 고정, 모든 동작이 구현이 완료되었다.

**7. 역할 분담**

정희석: WiFi 소켓 통신, Display, Camera 모듈 제어, GPIO 드라이버 제작 및 제어

박인철: 초음파 센서, 인체 감지 센서 GPIO 드라이버 제작 및 제어

**8. 실행 결과**

<최종 동작>

컴퓨터이(가) 표시된 사진

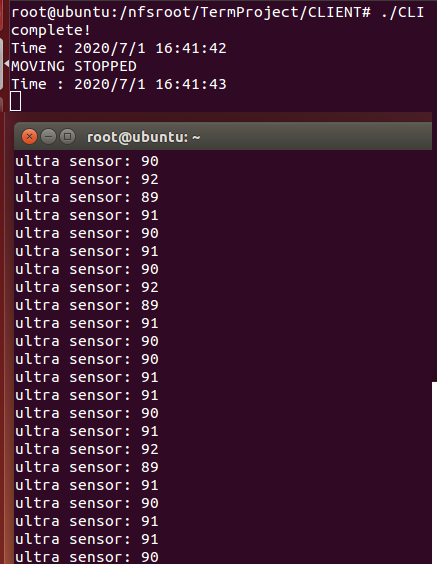
자동 생성된 설명

<좌: CLIENT(HOST) 우: SERVER(BOARD)>

-> 초음파 감지 센서의 값은 기준 값을 90으로 설정하고 센서의 품질을 고려하여 오차범위를 10%로 설정하였다. (이는 구현 과정에서 해당 위치에 설치하였을 때 나왔던 값이 96~ 84를 왔다 갔다 했기 때문) 그리고 중간에 센서의 속도와 print의 속도가 맞지 않아서 이상한 값(음수 값, 정상범위를 한참 넘는 값)이 나오는 것 역시 걸러내었다.

<1. 정상 상태의 동작>

침실, 침대, 방, 테이블이(가) 표시된 사진

자동 생성된 설명

<좌: 보통 상태의 사진, 우 상단: CLIENT(HOST), 우 하단: SERVER(BOARD)

-> 아무것도 없어서 클라이언트에는 아무것도 보내지 않으며 캡처한 화면은 거리를 확인하기 위해 BOARD에 출력시켜서 거리 값을 확인하였다. 이는 이후 초음파 센서의 값을 출력하지 않았을 때 클라이언트와 서버 모두 아무것도 표시 안되는 것을 확인할 수 있었다.

<2. 물건이 놓여있는 경우 or 사람이 움직임이 없는 경우>

앉아있는, 테이블이(가) 표시된 사진

자동 생성된 설명

<물병이 센서 앞에 놓여있는 사진>

앉아있는, 테이블, 쥐고있는, 목재의이(가) 표시된 사진

자동 생성된 설명앉아있는, 테이블, 쥐고있는, 목재의이(가) 표시된 사진

자동 생성된 설명

<좌: CLIENT(HOST), 우: SERVER(BOARD)>

-> 물병은 체온이 36.5도 이상이 아니어서 인체 감지 센서에는 인식이 안되지만 초음파 센서에는 인식이 되므로 카메라는 동작을 하되 클라이언트(HOST)에는 움직임이 멈춰있는 정보를 전달하고 Timestamp와 함께 출력한다.

<3. 사람이 침입한 경우>

사람, 실내, 테이블, 여자이(가) 표시된 사진

자동 생성된 설명컴퓨터이(가) 표시된 사진

자동 생성된 설명컴퓨터이(가) 표시된 사진

자동 생성된 설명

<좌: 사람이 센서 앞에 서있는 사진, 중: CLIENT(HOST), 우: SERVER(BOARD)

-> 사람이 카메라 앞에 있으며 위의 사진은 휴대폰을 초음파 센서에 대고 있었으므로 거리가 가까워짐을 감지하여 카메라가 따라 동작하게 되었고 인체 감지 센서가 감지하여 서버(BOARD)와 클라이언트(HOST) 모두 침입여부를 출력하게 하였다. 특히 CLIENT에서는 해당 Timestamp와 함께 출력하였다.

위의 3가지 경우는 화면(Terminal)에 출력하게 하였지만 이는 linux상에서 실행할 때 (> 출력파일)로 나타내면 로그파일로 남길 수도 있다.

이렇게 원하는 모든 동작을 구현할 수 있었다.

**9. 느낀 점 및 보완할 점**

이번 프로젝트를 진행하면서 배울 수 있었던 것은 지난 학기(3학년 2학기)에 배웠던 임베디드 시스템 과목을 통해서 Cross Compile과 uC/OS 프로그래밍, 임베디드 시스템 설계 및 실험 과목을 통해서 Cross Compile 및 OS없는 임베디드 시스템 프로그래밍을 배운 것과 연결되어 이번 ACHRO-I .MX6Q 보드에 Linux Toolchain을 올려서 Serial통신과 NFS를 배우고 linux상에서 GPIO, FPGA, Bluetooth, WiFi, QT Library, LCD, OpenCV, CAMERA를 어떻게 다루는 지를 배울 수 있었다.

처음에 실험을 따라갈 때는 배웠던 것을 기반으로 실습을 하다가 개발 환경의 차이, 버전 차이 등등의 각종 어려움이 많았다. 이로 인해서 실습을 할 때 의욕도 많이 잃었었고 힘들었다.

특히 이번 학기는 COVID-19로 인한 비대면 실험 실습을 진행하여서 즉각적인 피드백이 어려워서 더더욱 진행이 안되었었다. 하지만 조교님들의 많은 도움을 통해 마지막 프로젝트까지 마무리할 수 있어서 다행으로 생각한다.

무엇보다 프로젝트를 진행하면서 점점 진행되어가는 과정을 확인할 때마다 소소한 성취감을 느낄 수 있어서 좋았다.

그리고 이번 프로젝트에서 아쉬웠던 점, 즉 보완할 점을 생각해 보면 먼저 센서 값이 너무 일정하지가 않고 인체 감지 센서의 동작 속도와 방식은 생각보다 한발 느렸다. 이로 인해 더 좋은 성능은 낼 수 없었다.

그리고 인체 감지 센서의 구현이 가장 마지막으로 진행되어 더 완벽한 목표인 초음파 센서와 함께 인체 감지 센서의 값을 통해 카메라와 LCD, WiFi 통신의 동작이 진행될 수 있었지만 카메라 동작과 LCD는 초음파 센서 값을 기반으로, WiFi 통신은 초음파 센서 값 + 인체 감지 센서의 값을 기반으로 동작하게 되어서 너무 아쉬웠다.

이는 인체 감지 센서의 동작이 온도가 있는 물체가 움직임을 가질 때만 동작한다는 특성 때문에 어려웠던 것도 어느정도 있다.

그리고 센서의 값을 나타내는 속도(print)가 센서에서 값을 읽어오는 속도보다 한참 느려서 병목현상이 발생하여 임의로 동작에 sleep을 넣어 속도를 조절하여야 했던 것 역시 아쉬운 점이다.