

2018학년도 1학기

임베디드 시스템 소프트웨어

- 침입자 탐지 시스템 -



과목	임베디드 시스템 소프트웨어
담당교수	진현욱 교수님
팀	Team3
이름	김상원, 권오승, 김제헌
제출일	2018.06.25.

목차

1. 배경	3
1.1. 주제 소개	3
1.2. 주제 선정 이유	3
2. 목적	3
2.1. 개발 목표	3
2.2. 개발 목적	3
3. 설계	4
3.1. 시스템 구조	4
4. 구현	5
4.1. Pi1 (파이 간 신호제어)	5
4.2. Pi2 (초음파 센서, 조도 센서)	5
4.3. Pi3 (카메라, LED, 스피커, 모터)	6
4.4. 결과물 종합	7
5. 시연	9
5.1. 시연 시나리오	9
5.2. Youtube 링크	9
6. 일정관리	9
7. 역할분담	9

1. 배경

1.1. 주제 소개

이번에 저희 팀이 개발한 프로젝트의 주제는 '침입자 감지 시스템'입니다.
내용은 심플합니다. 도둑, 강도 등의 무단 침입자가 집 안에 들어섰을 때, 이를 감지할 수 있도록 해주는 시스템입니다.

1.2. 주제 선정 이유

이번 임소 프로젝트의 요구사항 중, 최소 3가지 이상의 센서 또는 액추에이터를 사용해야 했는데, 그 중 2개는 초음파 센서와 조도 센서[=빛 센서]로 고정되었습니다.

본래 하고 싶었던 시나리오[미세먼지 측정-환기 알림 시스템]가 있었지만, 앞의 두 개의 센서를 넣자니 시나리오가 맞지 않게 되어서 못하게 되었습니다. 그 밖에도 초음파 센서까지는 어떻게든 이용할 수 있어도, 조도 센서는 일반적으로 사용되지는 않았습니다. 그래서 조도 센서가 이용될만한 주제[=무언가를 수행하는데 있어서, 밝고 어두움이 꼭 판별되어야 되는 어떤 주제]가 무엇이 있을까 고민하다가 이 주제를 선정하게 되었습니다.

2. 목적

2.1. 개발 목표

기본적으로는 초음파 센서가 항상 탐지를 하고 있습니다. 그러다가 무언가 물체가 감지되면, 조도 센서가 작동해서 현재 방의 상태가 밝은지 어두운지를 체크하고, 밝으면 바로 사진을 찍고, 어두우면 LED 등을 켜 이후에 사진을 찍게 되며, 그와는 별개로 스피커로 경고음이 울리며, 모터가 동작해서 출입문이 닫히게 되는 구조입니다. 찍힌 사진은 서버로 전송되어 침입기록이 남게 됩니다.

2.2. 개발 목적

이것에 대한 개발 의의를 설명하자면, 귀중품 또는 접근제한 구역에 설치하여 침입 시도 및 접근 시도를 탐지할 수 있고, 예를 들어 도둑 등의 침입자가 집안에 들어왔다가 나간 후 나중에 이 기록을 확인해보으로써 누가 침입했는지를 확인할 수 있으며 이러한 자료를 증거 자료로 활용함으로써 추후 후속조치도 가능한 등의 효과가 있을 것으로 예상됩니다.

여기서 조도 센서 및 LED에 대해서 추가적인 언급을 해보겠습니다. 나중에 침입이 감지되었다 하더라도, 만약 실내 환경이 아주 어둡다면 설령 사진을 찍어도 범인의 얼굴이 제대로 찍히지가 않습니다.

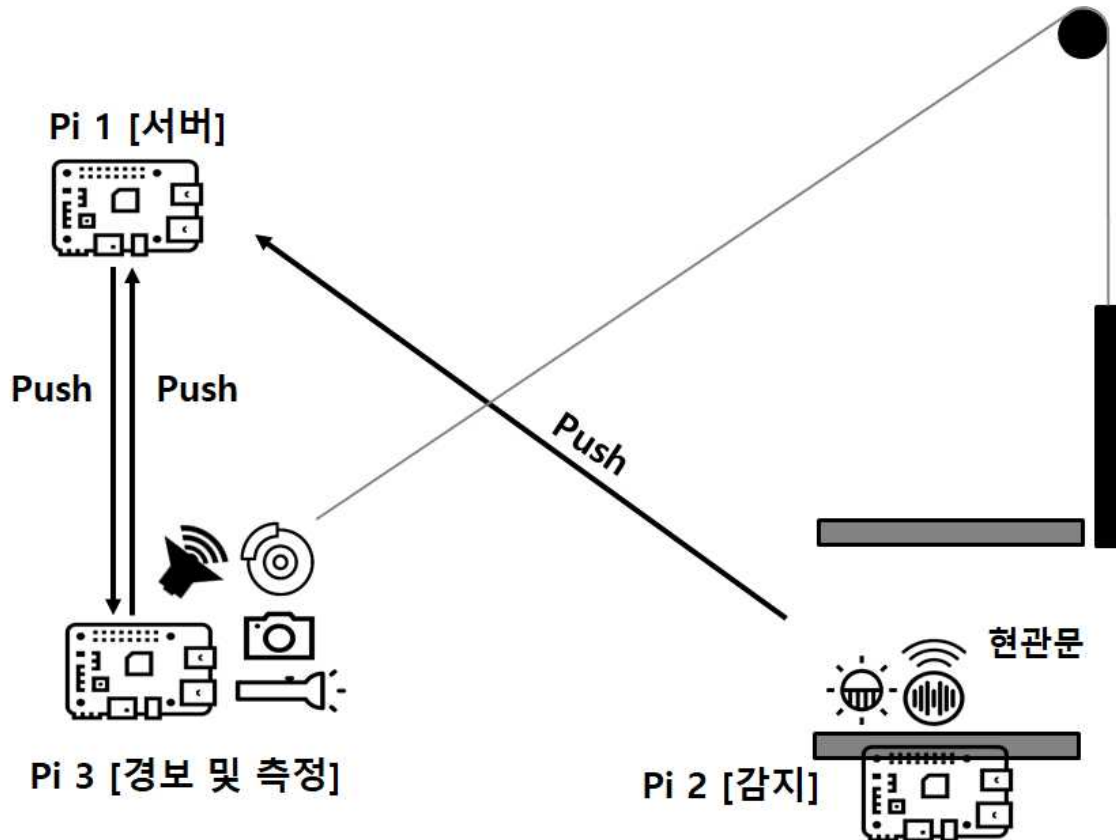
하지만, 이러한 환경을 조도 센서가 현재 방의 상태를 파악하고, 일정 기준 이상으로 어두울 때 LED 등이 켜진다면, 보다 선명한 범인의 얼굴을 찍을 수가 있게 되는 의의가 있습니다.

3. 설계

3.1. 시스템 구조

이를 위한 시스템 구조에 대해서 설명해 보겠습니다.

앞서 언급한대로 저희는 초음파 센서, 카메라 모듈, 빛 센서, LED, 스피커, 모터 모듈로 총 6 종류의 장비를 사용합니다.



라즈베리파이1 : 별도의 센서 및 모듈은 연결하지 않는다.

- 안전한 곳에서 위치하고, 메인 서버처럼 동작할 예정이며, 파이2와 파이3을 중개한다.
- 침입 기록을 관리한다.

라즈베리파이2 : 초음파 센서, 조도 센서 연결

- 적당한 위치에서 초음파 센서를 주기적으로 실행시키며, 침입자를 감지하면 빛 센서의 측정값에 따라, 결과 메시지를 파이1로 전달한다.

라즈베리파이3 - LED, 카메라 모듈, 스피커 모듈, 모터 모듈

- 파이1로부터 신호를 받은 값에 따라, LED를 켤 수도 있으며, 카메라 모듈을 통해 사진을 찍어서 파이1로 보낸다. 그와 별개로 경고음을 울리며, 모터가 동작해서 출입문을 닫는다.

4. 구현

4.1. Pi1 (파이 간 신호제어)

파이 간 신호제어

- 앞서 3에서 언급한 대로, 파이1은 안전한 곳에서 위치하고 파이2와 파이3을 중개하는 등의 메인 서버처럼 동작한다는 설정이라서 별도의 센서 및 모듈은 연결하지 않았습니다.
- 기본적으로 파이 간 신호제어, 즉 소켓 통신에 있어서 개발언어는 파이썬을 이용하였습니다.
- 팀원 중 한명의 노트북을 Wi-Fi 핫스팟(공유기)로 만들었고, 3개의 파이가 이 망에 접속하게 함으로써, 서로 통신이 되도록 했습니다.
- Pi2, Pi3으로부터 각각 접속을 받고 대기하다가, Pi2로부터 침입자 탐지 신호를 받으면, 이를 Pi3으로 알려주게 되며, Pi3으로부터 찍힌 사진 값을 받아서, capture 폴더에 현재 시각으로 파일 이름을 저장하게 됩니다.
- 파일을 저장한 이후, 서버의 사용자가 임의로 어떤 신호를 p2와 pi3에게 보내면, pi2와 pi3는 각각 침입자 탐지 재개와, 닫힌 문을 열게 됩니다.

파이 간 통신 프로토콜

- 파이 간 프로토콜

수신자	송신자	전송메시지	의미
pi2 [client1]	pi1 [server]	1111	detection and now bright
		1100	detection and now dark
		0000	no detection
pi1 [server]	pi3 [client2]	1111	detection and now bright
		1100	detection and now dark

4.2. Pi2 (초음파 센서, 조도 센서)

초음파 센서

- 수업 실습자료인 '실습10_RaspberryPiSensors.pdf'의 7Page를 보면, Trigger신호를 발생 시키고, 10ns이후에 8번의 초음파 신호를 전송하게 되고, 초음파 신호가 돌아왔을 때 echo값이 발생하게 되며, 이때까지의 시간 차이 값을 이용해서 거리를 측정한다고 나와 있습니다.
- 이를 통해서 일단, Trigger 및 Echo신호를 막아 두었다가, 원할 때 Trigger 신호를 발생 시키고, 10ns이후에 do_gettimeofday()를 통해서 신호를 보낸 시각을 측정하며, echo값을 읽기 모드로 대기시키고, echo값을 받아서 인터럽트가 발생하는 순간의 시각을 측정해서 거리를 구하도록 구현하였습니다. 이때, 인터럽트가 발생하기 전까지의 대기는 wait_queue를 써서 구현했습니다.
- 이렇게 ultrasonic.c를 구현하였고, 이를 get_ultrasonic_distance() 함수로 만들어서 이용하는 응용프로그램인 ultrasonic_lib.c를 구현했습니다.
- 별도로 Makefile를 통한 ultrasonic.ko와 mknod.sh 파일도 있습니다.

조도센서

- 수업 이론자료인 '10_Serial_Busses.pdf'의 SPI 부분을 보면, SPI의 Full-duplex data transfer에 대한 내용이 있습니다. 리눅스 커널 spi드라이버 /dev/spidev0.0를 이용하였고, 채널 정보를 포함한 3바이트의 버퍼를 linux/spi/spidev.h의 spi_ioc_transfer 구조체에 송신, 수신 버퍼로 지정하고, 그 외 설정을 라즈베리파이 환경에 맞춰 설정한 후 SPI_IOC_MESSAGE를 이용하여 full-duplex 통신을 하였습니다. 여러 시험을 통해 정한 값 1000 이하인 경우 어둡다고 판단하여 정보를 return합니다.
- 3바이트 전송 바이트의 내용은 처음 5비트 0으로 채우고 start bit, single/diff bit와 D2, D1, D0 비트를 통해 채널을 정합니다. 이후 이어지는 남은 비트는 빛 센서의 응답결과가 뒤에서부터 12비트 입력되게 됩니다. 따라서 0 5비트, start bit, single/diff bit, D2가 첫 번째 바이트에, D1, D0가 두 번째 바이트에 적히게 되고, 두 번째 바이트와 세 번째 바이트에 뒤에서부터 12비트에 결과가 적힙니다. SPI의 bits_per_word 속성을 8비트로 지정하여 한 번에 한 바이트씩 읽도록 하였습니다.
- 이렇게 light_lib.c를 구현했습니다.

C언어 소스코드를 파이썬 코드에 적용

- 이러한 C언어를 파이썬 코드에 적용시키기 위해서, gcc로 so파일을 만들었습니다.
- 작성한 C언어 소스코드인 ultrasonic_lib.c 및 light_lib.c를 컴파일하여 리눅스에서 파이썬이 이용 가능한 .so 파일로 빌드하면 해당 모듈을 import해서 사용할 수가 있습니다.

4.3. Pi3 (카메라, LED, 스피커, 모터)

카메라 모듈

- 수업 실습자료인 '실습10_RaspberryPiSensors.pdf'를 참고해서 카메라로부터 사진을 찍을 수 있게 했습니다.
- 찍힌 사진은 screenshot 폴더에 임시로 저장되며, 서버인 Pi1로 전송되게 됩니다.

LED

- 수업 실습자료인 '실습05_GPIO.pdf'를 참고해서 LED의 불을 켜고 끌 수 있게 했습니다.
- Pi1로부터 받은 값을 고려해서, 밝을 때는 불이 켜지지 않고, 어두울 때, 불이 켜지도록 했습니다.

스피커 모듈

- 수업 실습자료인 '실습10_RaspberryPiSensors.pdf'를 참고해서 일정 시간동안, 경고음이 울리도록 했습니다.

모터 모듈

- 수업 실습자료인 '실습11_스테퍼 모터.pdf'를 참고해서 특정 각도를 움직일 수 있는 함수를 통해서 우드락 모형의 문을 열고 닫을 수 있게 했습니다.

PI3 모듈 종합

- 앞서 언급한, 카메라, LED, 스피커, 모터 모듈을 이용하는 pi3.c, pi3.h를 작성하였고, 이를 pi3_turnOn(), pi3_turnOff(), pi3_doorSpeakerOn() 함수로 만들어서 이용하는 응용 프로그램인 pi3_app.c를 구현했습니다.
- 별도로 Makefile를 통한 pi3.ko와 mknod.sh 파일도 있습니다.

C언어 소스코드를 파이썬 코드에 적용

- 이러한 C언어를 파이썬 코드에 적용시키기 위해서, gcc로 so파일을 만들었습니다.
- 작성한 C언어 소스코드인 pi3_app.c를 컴파일하여 리눅스에서 파이썬이 이용 가능한 .so 파일로 빌드하면 해당 모듈을 import해서 사용할 수가 있습니다.

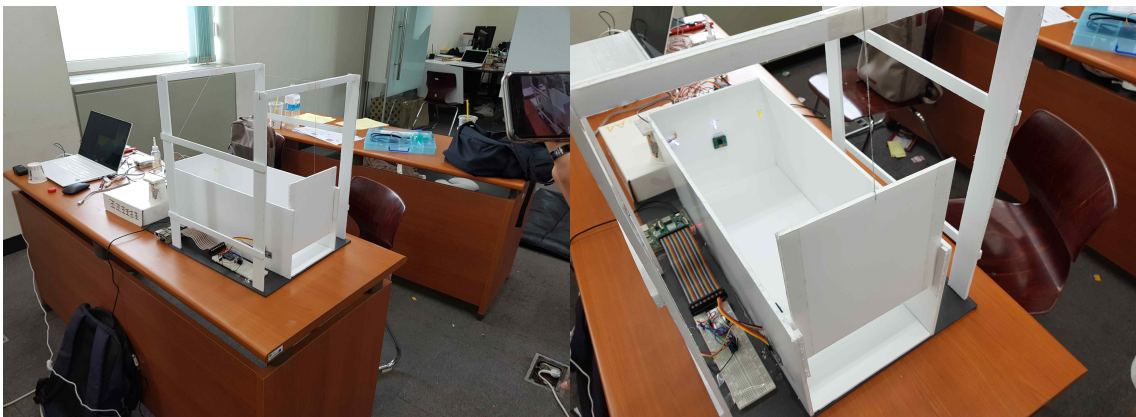
4.4. 결과물 종합

우드락 모형

- 시뮬레이션 환경을 구축하기 위해서 우드락으로 다음과 같은 모형을 만들었습니다.
- 왼쪽 이미지는 전체 우드락 모형의 이미지이며, 오른쪽 이미지는 정면에서 바라본 우드락 모형의 이미지입니다. [정면 : 카메라 & LED, 좌측면 : 초음파센서 & 빛센서]



- 물체가 탐지 시, 사진을 찍고, 경보음과 함께 차단문이 내려가게 됩니다.
- 왼쪽 이미지는 밝을 때의 상황이며, 오른쪽 이미지는 어두울 때의 상황[빛 센서를 주먹 쥐고 측정함]으로 각각 LED가 꺼졌고, 켜진 상황입니다.



- 이렇게 찍힌 사진은 pi1의 서버로 전송되서 capture폴더에 다음과 같이 남게 됩니다.

```
pi@raspberrypi:~/capture $ ls
2018-06-22_10:25:14_00.jpg  2018-06-22_10:25:35_00.jpg  2018-06-22_10:26:00_00.jpg  2018-06-22_10:28:32_00.jpg  2018-06-22_12:10:20_00.jpg  2018-06-22_13:54:50_00.jpg  2018-06-25_06:18:34_00.jpg
2018-06-22_10:25:17_00.jpg  2018-06-22_10:25:30_00.jpg  2018-06-22_10:26:03_00.jpg  2018-06-22_10:28:35_00.jpg  2018-06-22_12:16:53_00.jpg  2018-06-22_13:55:11_00.jpg  2018-06-25_06:20:00_00.jpg
2018-06-22_10:25:20_00.jpg  2018-06-22_10:25:41_00.jpg  2018-06-22_10:26:06_00.jpg  2018-06-22_10:28:41_00.jpg  2018-06-22_13:28:25_00.jpg  2018-06-25_05:12:04_00.jpg  2018-06-25_06:21:47_11.jpg
2018-06-22_10:25:23_00.jpg  2018-06-22_10:25:44_00.jpg  2018-06-22_10:26:09_00.jpg  2018-06-22_10:28:44_00.jpg  2018-06-22_13:43:47_00.jpg  2018-06-25_05:14:10_00.jpg  2018-06-25_06:23:27_11.jpg
2018-06-22_10:25:26_00.jpg  2018-06-22_10:25:47_00.jpg  2018-06-22_10:26:12_00.jpg  2018-06-22_12:08:24_00.jpg  2018-06-22_13:45:55_00.jpg  2018-06-25_05:22:09_11.jpg
2018-06-22_10:25:29_00.jpg  2018-06-22_10:25:51_00.jpg  2018-06-22_10:26:15_00.jpg  2018-06-22_12:09:54_00.jpg  2018-06-22_13:46:46_00.jpg  2018-06-25_05:24:19_00.jpg
2018-06-22_10:25:32_00.jpg  2018-06-22_10:25:57_00.jpg  2018-06-22_10:26:18_00.jpg  2018-06-22_12:10:10_00.jpg  2018-06-22_13:47:25_00.jpg  2018-06-25_06:16:59_00.jpg
pi@raspberrypi:~/capture $
```


5. 시연

5.1. 시연 시나리오

1. Pi1의 서버를 실행시키고, 연이어서 Pi2, Pi3의 클라이언트를 실행시켜서 서버에 접속시킵니다.
2. Pi2에서는 3초 간격으로 초음파 센서가 작동됩니다.
3. Pi2에서 측정 값이 일정 수치 이하로 떨어졌을 때, 물체가 탐지된 것으로 판단하며, 이 때 조도 센서가 작동되서 현재 방의 상태가 밝은지 어두운지를 판단합니다.
4. Pi2는 이 값을 Pi1로 보내게 되며, Pi1은 이를 보고 Pi3으로 알리게 됩니다.
5. Pi3은 해당 신호에 따라 LED를 켜지 말지를 결정하게 되며, 그것과는 별개로 사진을 찍고, 스피커로 경고음이 나오게 되며, 모터가 작동해서 출입문이 닫히게 됩니다.
6. 찍힌 사진은 Pi1인 서버로 전송되며, 이 데이터는 침입기록으로써 남게 됩니다.
7. 파일을 저장한 이후, 서버의 사용자가 임의로 어떤 신호를 p2와 pi3에게 보내면, pi2와 pi3는 각각 침입자 탐지 재개와, 닫힌 문을 열게 됩니다.

5.2. Youtube 링크

임베디드 소프트웨어 시연 영상 1 [밝을 때] : <https://youtu.be/gt3u9KbyXg0>

임베디드 소프트웨어 시연 영상 2 [어두울 때] : <https://youtu.be/ENT74C1JVdl>

6. 일정관리

일정	내용	보충설명
~06.03	중간발표 준비	프로젝트 팀 정하기 및 프로젝트 주제, 대략적인 방향에 대한 고민
06.04	중간발표	프로젝트 제안서 발표
06.05~06.08	설계	프로그램의 구현 부분에 대한 실질적인 고민 및 설계
06.08~06.13	구현	각자 맡은 부분을 프로그래밍 및 유닛 테스트
06.13~06.24	통합 및 테스트, 보고서	각자 구현한 모듈을 합치고 테스트, 추가 모형 제작 최종 보고서 작성
06.25	최종발표	시연 및 최종보고서 제출

7. 역할분담

- 김상원 : 중간발표, Pi1 (파이 간 신호 제어), C언어 파이썬 모듈 연결, 조도 센서
- 김제현 : Pi2 (초음파 센서, 조도 센서)
- 권오승 : Pi3 (카메라, 모터, 스피커, LED), 시연용 우드락 모형 제작