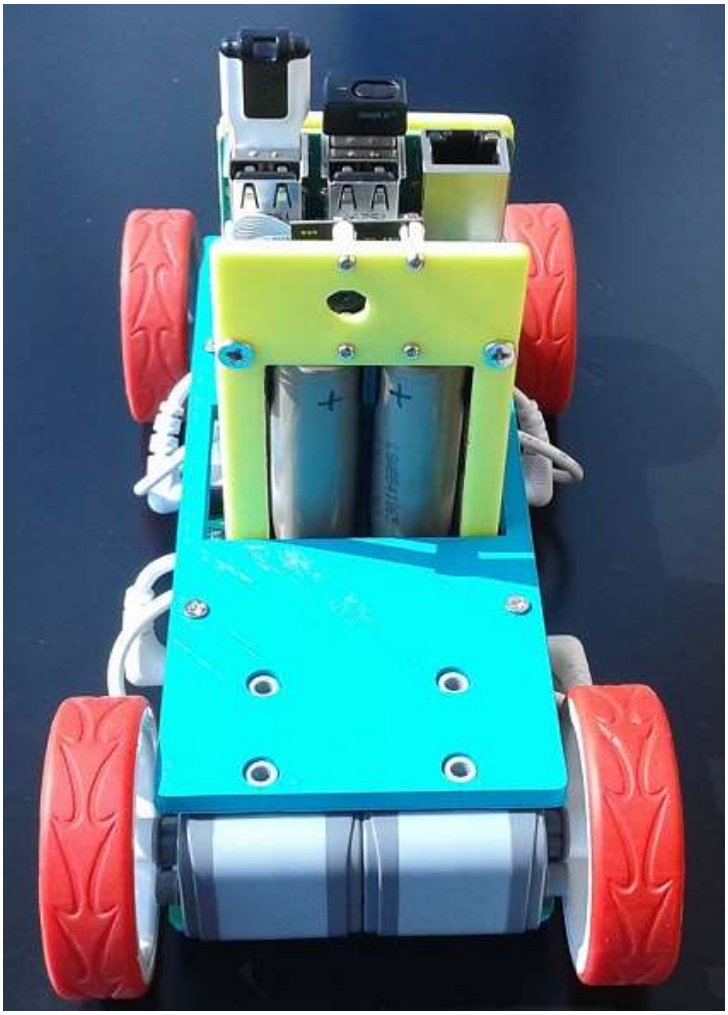
만들면서 배우는

라즈베리파이 라지그 프로젝트

First Edition



커널연구회(www.kernel.bz)

정재준(rgbi3307@nate.com)

저작권

만들면서 배우는 라즈베리파이 라지그 프로젝트

저자 정재준

Copyright © 2012-2016 커널연구회(www.kernel.bz). All rights reserved.
Published by 커널연구회(www.kernel.bz).
서울 금천구 두산로 70 현대지식산업센터 A동 26층 2611호

커널연구회는 리눅스 커널과 자료구조 알고리즘을 연구하고 리눅스 시스템 프로그래밍 및 디바이스드라이버 개발을 통하여 창의적인 프로젝트를 수행하여 IoT 관련 제품들을 만들어 일상생활을 풍요롭고 편리하게 하는데 가치를 두고 있습니다. 아울러 관련 기술들을 교육하여 여러사람들과 공유할 수 있도록 노력하고 있습니다. 커널연구회가 연구 개발한 결과물들은 체계적으로 문서화하여 온라인(www.kernel.bz)상에서 무료 혹은 유료로 제공하고 있습니다. 커널연구회가 제공하는 저작물에는 저작권을 표시하고 있으며 사용자는 저작권 표시를 보존해 주어야 합니다. 커널연구회가 유료로 제공하는 저작물은 사용자에게 개인키(암호)를 부여 하므로 개인키를 타인에게 공개 및 양도하는 일이 없도록 해야 합니다.

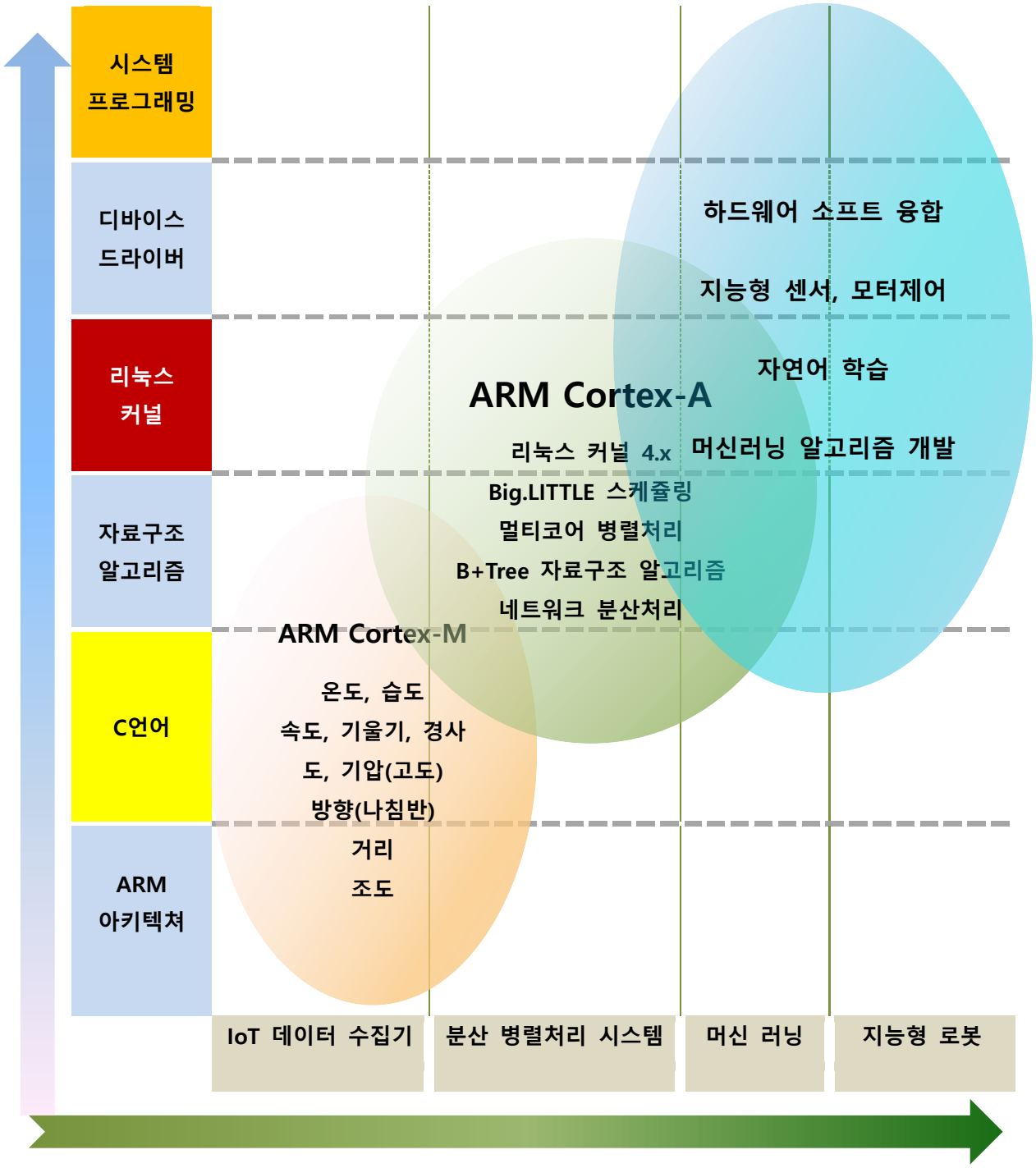
기타 자세한 내용들은 커널연구회 웹사이트(www.kernel.bz)를 방문해 주시기 바랍니다.
감사합니다.

발행인: 정재준
발행처: 커널연구회
출판사등록번호: 제2011-75호
출판사등록일: 2011년 09월 27일
전화: 02-2181-3507 / 팩스: 031-594-5307
ISBN: 978-89-97750-07-8

초판발행일: 2016년 04월 20일

커널연구회 로드맵

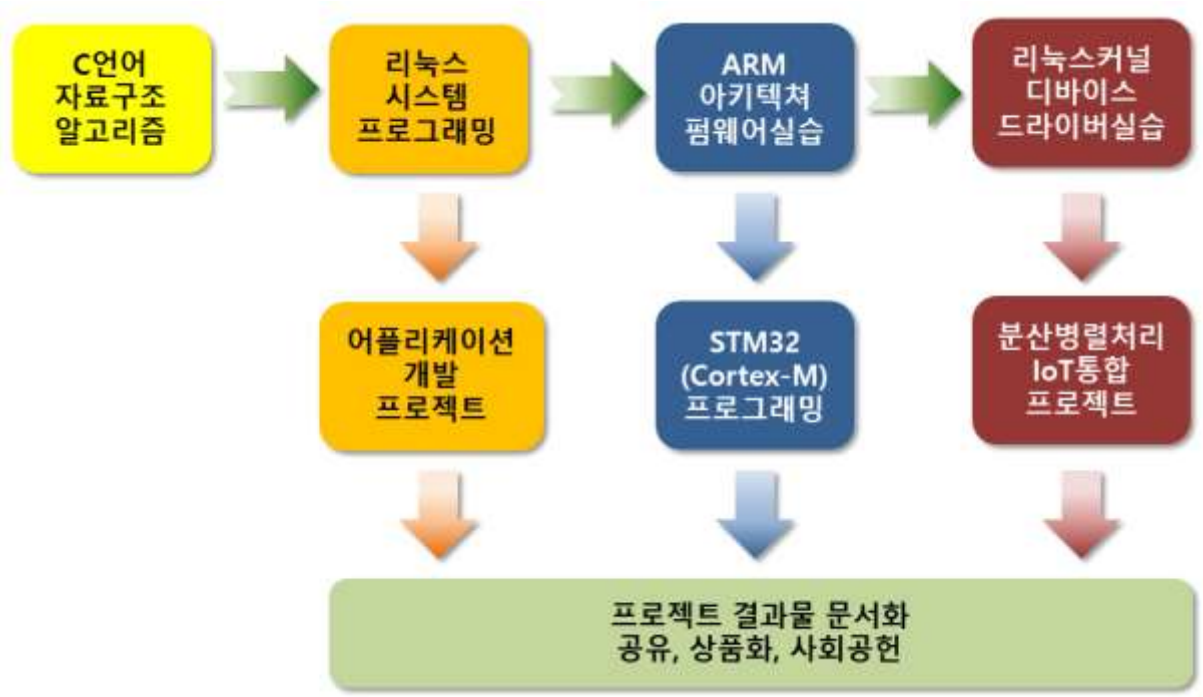
연구개발 및 교육



제품 개발 프로젝트

커널연구회 교육과정 로드맵

커널연구회 교육학원은 다음과 같은 훈련과정 개발 체계를 갖추고 있습니다. 먼저 프로그래밍의 가장 기본이 되는 "C언어와 자료구조 알고리즘"에 대해서 자체적으로 출판한 교재를 통하여 교육생들이 확실이 기초를 다질 수 있도록 교육한후, 이것을 바탕으로 "리눅스 시스템 프로그래밍" 과정을 익혀서 어플리케이션을 개발할 수 있도록 합니다. 그리고 "ARM 아키텍쳐 펌웨어 실습" 과정을 거쳐서 STM32 펌웨어 프로그래밍을 할 수 있는 역량을 배양합니다. 위의 과정들을 마치게 되면 "리눅스 커널과 디바이스드라이버 실습" 과정을 통하여 IoT 환경의 프로젝트를 수행하도록 하는 훈련과정 개발 체계를 갖추고 있습니다. 최종적으로는 교육과정에서 산출한 내용물과 교육 프로젝트 수행 결과물들을 서로 공유하여 사회에 공헌하는 체계로 접근하고 있습니다.



♣ 좀더 자세한 내용들은 커널연구회(www.kernel.bz) 웹사이트를 참조하세요.

라지그 프로젝트 교육안내

만들면서 배우는 라즈베리파이 라지그 프로젝트는 커널연구회의 교육과정을 통해서 자세히 학습할 수 있도록 구성했다. 본 책자는 라지그 프로젝트를 수행하기 위한 여러가지 배경지식에 대해서 설명하는 것이다. 라즈베리파이에는 리눅스가 포팅되어 있으므로 리눅스 환경에서 시스템 프로그래밍하기 위해서 여러가지 개발환경 설치하는 것에 대해서는 자세히 설명하고 있으나, 소프트웨어 구현과 실습에 대해서 책으로 모두 설명하기에는 여러가지 제약사항들이 많은 실정이다. 그래서 소프트웨어를 설명하고 직접 구현 및 실습하는 것은 아래와 같은 커널연구회 교육과정을 통해서 진행하고 있다.

커널연구회 라지그(RaZig) 프로젝트 교육과정에 참여하기 위해서는 C언어와 자료구조 알고리즘 및 리눅스 시스템 프로그래밍에 대해서 어느 정도 사전 지식을 갖추고 있어야 한다. 커널연구회는 C언어와 자료구조 알고리즘 교육 과정과 리눅스 시스템 프로그래밍 교육과정을 마련하고 있으므로 사전 지식이 부족하다면 아래 링크를 참조해 주기 바란다.

커널연구회 교육과정 상세 안내 링크

http://dev.kernel.bz/doc/kernel_bz_edu_detail_intro.htm

라지그 프로젝트는 C언어와 자료구조 알고리즘 및 리눅스 시스템 프로그래밍 과정을 이수하신 분들을 대상으로 아래와 같이 교육을 진행한다. 교육 기간은 2개월이지만, 상황에 따라서 변경될 수 있으므로 커널연구회 웹사이트(http://www.kernel.bz)를 자주 참조해 주기 바란다.

라지그 프로젝트 교육내용

순번	교육 제목	상세 내용
1	라즈베리파이 개발환경	라즈베리파이 개발환경 설정방법 통합개발도구(Code Blocks) 설치 디바이스 포팅 사운드, 비디오, USB WiFi 동글, 카메라 모듈 포팅 GPIO 헤더핀맵 이해
2	리눅스 시스템 프로그래밍	오픈소스 사용법 설명 카메라 제어 프로세스 코딩(쓰레드, 데몬, 시그날) 카메라 동작감시 프로세스 코딩(쓰레드, 데몬, 시그날) 음악 및 동영상 재생 프로세스 코딩(쓰레드, 데몬, 시그날) 인터넷 라디오 플레이 프로세스 코딩(쓰레드, 데몬, 시그날)

3	웹서버프로그래밍	아파치, PHP, MySQL 연동방법 PHP 웹페이지와 C언어 시스템 프로그래밍 연동방법 스마트폰과 USB WiFi 무선 인터페이스 방법
4	모터 제어 프로그래밍	모터 제어 보드의 이해 PWM 인터페이스 이해 및 코딩 스마트폰으로 모터제어 인터페이스 코딩 PHP와 C언어로 로봇제어 프로그래밍 코딩
5	센서 데이터 수집 프로그래밍	라지그 센서 보드의 이해 I2C 인터페이스 이해 및 코딩 스마트폰으로 센서 데이터 수집 인터페이스 코딩 PHP와 C언어로 센서 데이터 통계 및 그래프 처리 기능 코딩

커널연구회의 라지그 프로젝트 교육과정에 참여하기 전에 아래에 대해서 먼저 점검해 보는 것도 필요하다. 아래에서 0, 1, 2 번 사항에 대해서는 어느 정도 선행학습 되어 있어야 한다. 그리고, 3, 4, 5 번 사항은 반드시 필요한 것은 아니지만 알고 있으면 많은 보탬이 될 듯 하다.

사전 점검 내용

0. 커널연구회 라지그 프로젝트에 대해서 사전 정보를 파악하고 있는가?
1. C 언어와 자료구조 알고리즘에 대해서 어느정도 학습했는가.
2. 리눅스 시스템 프로그래밍은 어느정도 알고 있는가?
3. 웹(PHP, MySQL)서버 프로그래밍도 가능한가?
4. 3D 프린터를 사용하기 위해서 3D 도면을 설계해본 경험은 있는가?
5. 하드웨어 회로 설계에 대한 지식은 어느 정도인가?

저자 소개

정재준 (rgbi3307@nate.com) / 커널연구회(www.kernel.bz)

저자는 학창시절 마이크로프로세서 제어 기술을 배웠으며 리눅스 커널을 연구하고 있다. 15년 이상 쌓아온 실무 경험을 바탕으로 "C언어와 자료구조 알고리즘", "리눅스 시스템 프로그래밍", "리눅스 커널과 디바이스드라이버 실습2", "오라클 실무활용 SQL튜닝2"등의 책을 집필하고, 월간임베디드월드 잡지에 다수의 글을 기고 하였다. 또한 "맞춤형 문장 자동 번역 시스템 및 이를 위한 데이터베이스 구축방법 (The System for the customized automatic sentence translation and database construction method)" 라는 내용으로 프로그래밍을 하여 특허청에 특허출원 하였다. 최근에는 서울시 버스와 지하철 교통카드 요금결재 단말기에 들어가는 리눅스 커널과 디바이스 드라이버 개발 프로젝트를 성공적으로 수행했고 여러가지 임베디드 제품을 개발했다. 저자는 스탠포드대학교의 John L. Hennessy 교수의 저서 "Computer Organization and Design" 책을 읽고 깊은 감명을 받았으며, 컴퓨터구조와 자료구조 알고리즘 효율성 연구를 통한 기술서적 집필에 노력하고 있다. 저자는 온라인 상에서 커널연구회(http://www.kernel.bz/) 웹사이트를 운영하며 연구개발, 교육, 관련기술 공유 등을 위해 노력하고 있다.

♣ 저자가 집필한 책들

목차

내용

라즈베리파이 라지그 프로젝트 **1**

저작권 **2**

커널연구회 로드맵 **3**

커널연구회 교육과정 로드맵 **4**

라지그 프로젝트 교육안내 **5**

저자 소개 **7**

목차 **8**

7. 라지그(RAZIG) 프로젝트 **10**

7.1 라즈베리파이 헤더핀 10

7.2 라지그 SENSOR 보드 11

7.3 라지그 MOTOR 보드 13

7.4 라지그 보드 조립 방법 14

7.4.1 Senosr 보드 조립 *14*

7.4.2 Motor 보드 조립 *21*

7.4.3 Senosr와 Motor 보드 연결 *25*

7.5 라지그 기능 소개 27

7.5.1 개인용 PC기능 *29*

7.5.2 카메라 녹화기능 *30*

7.5.3 방범 카메라 기능 *32*

7.5.4 센서 데이터 수집 기능 *34*

7.5.5 음악파일 재생 기능 *36*

7.5.6 동영상 및 라디오방송 듣기 *37*

7.5.7 로봇 자동차 제어기능 *38*

7.6 라지그 프로젝트 교육안내 39

부록B. 커널연구회 교육과정 상세안내 **41**

B1.0 커널연구회 교육과정 로드맵 41

B1.1 C언어와 자료구조 알고리즘 42
B1.2 리눅스 시스템 프로그래밍 46
B1.3 ARM 아키텍쳐, STM32 프로그래밍 50
B1.4 리눅스 커널 자료구조 알고리즘 실습 53
B1.5 리눅스 커널 소스 디버깅 실습 57
B1.6 리눅스 커널 DEVICE TREE 실습 61
커널연구회 교육학원 위치(약도) 65

7. 라지그(RaZig) 프로젝트

커널연구회의 라지그 프로젝트는 직접 만들면서 배우는 과정으로, 커널연구회에서 자체 제작한 라지그(RaZig) 보드를 라즈베리파이의 40 핀 확장 슬롯에 장착하여 다음과 같은 기능을 수행한다.

- **라지그 Senosr 보드: 기울기, 방향, 온도, 습도, 이동속도 측정**
- **라지그 Motor 보드: DC 모터(4 개) 제어, 스피커 소리 출력**

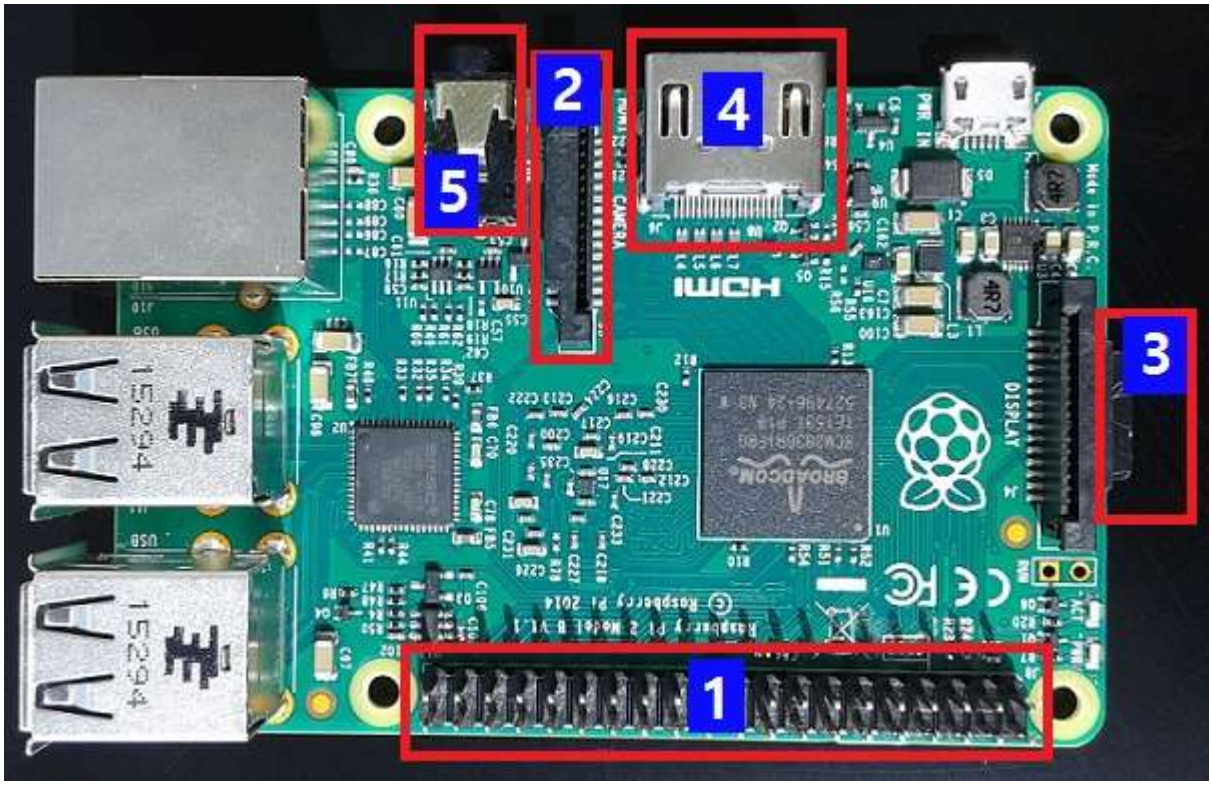
라즈베리파이는 아래와 같이 외부로 연결할 수 있는 40 개의 확장핀을 제공한다. 커널연구회가 제작한 RaZig 보드는 여기에 장착하여 각종 장치들을 제어할 수 있도록 구성한 것이다.

7.1 라즈베리파이 헤더핀

아래의 라즈베리파이 보드에서 1 이 커널연구회 라지그 센서보드와 연결되는 헤더핀(40 핀)이다.

2 에는 카메라가 연결되고 3 은 마이크로 SD 카드, 4 에는 HDMI 케이블이 연결된다.

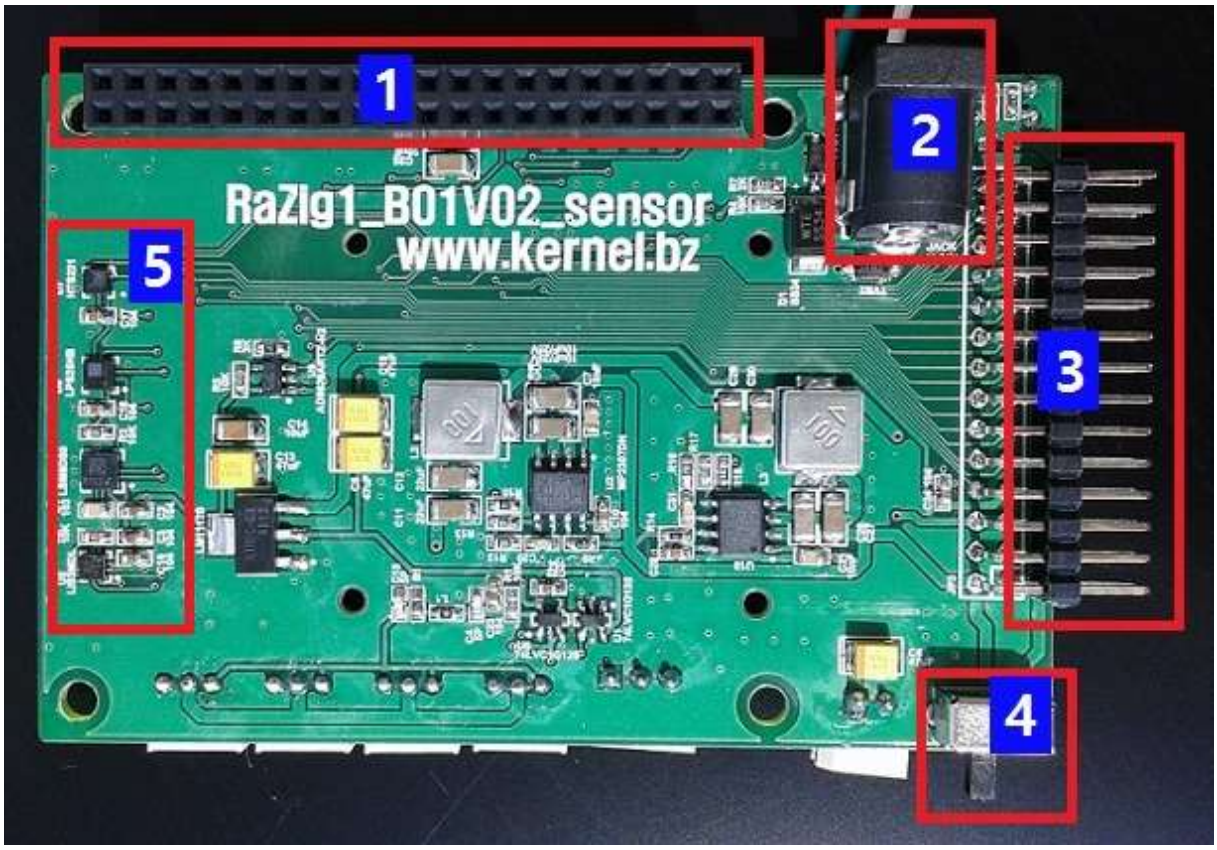
라즈베리파이 보드



7.2 라지그 Sensor 보드

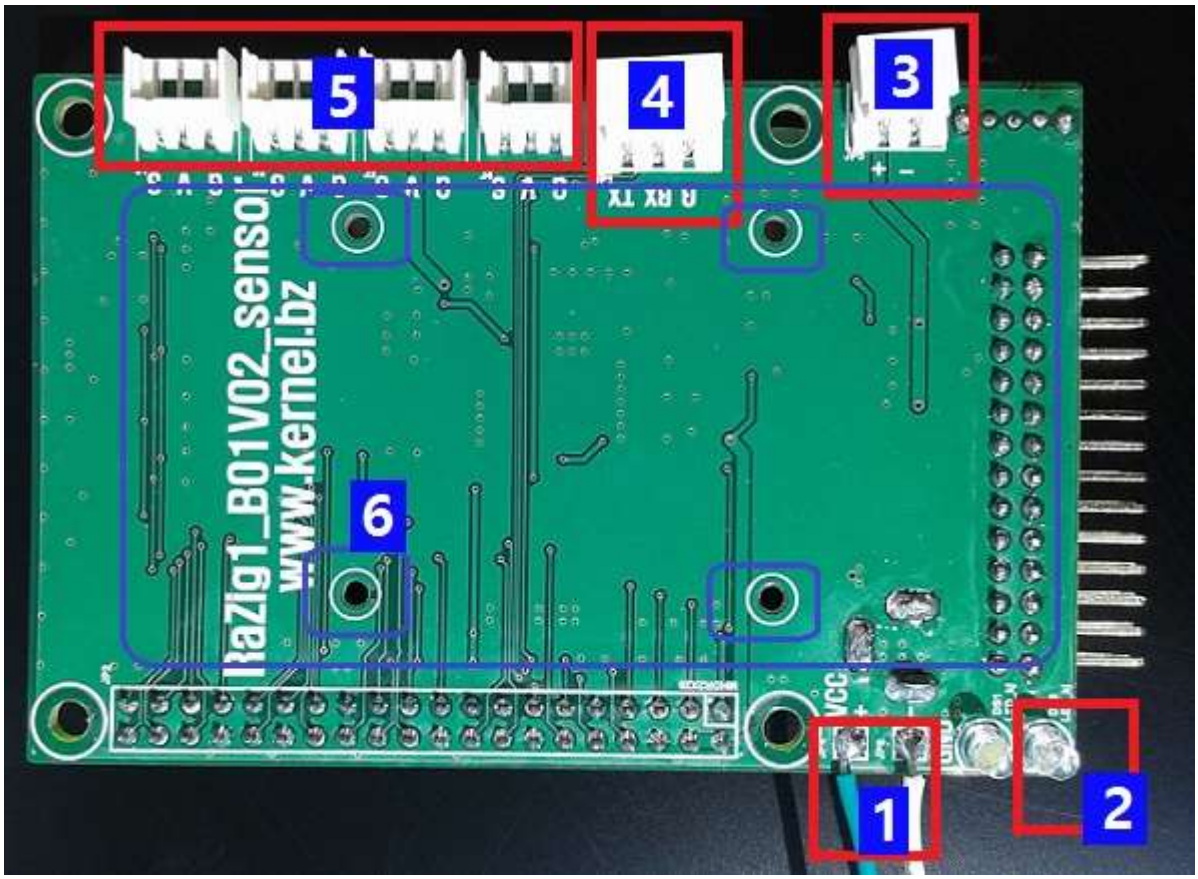
아래부터 커널연구회 라지그 센서 보드의 각 부품별 용법을 요약 설명한다.

커널연구회 RaZig Sensor 보드(하단부)



1	라즈베리파이와 연결하는 40핀 헤더 커넥터
2	외부 어댑터(12V) 전원 입력잭
3	라지그 모터 보드와 연결하는 28핀 헤더핀
4	전원 ON/OFF 스위치
5	센서(온도, 습도, 기울기, 속도, 방향)

커널연구회 RaZig Sensor 보드(상단부)

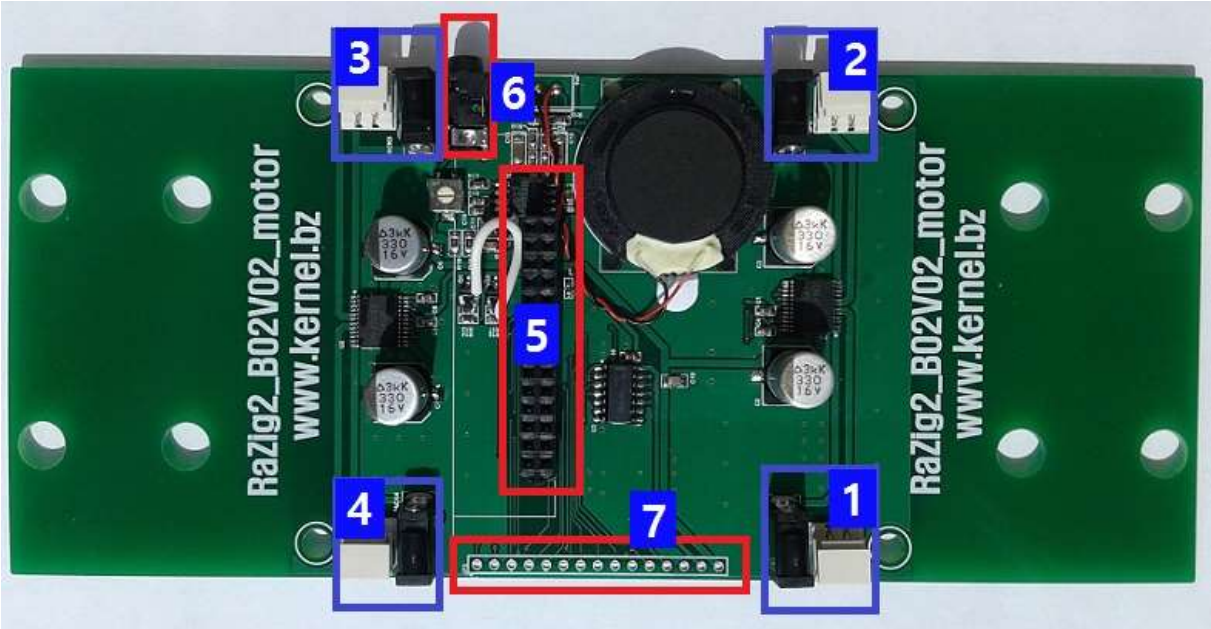


1	배터리 전원 연결선 (7.4V)
2	전원 상태(ON/OFF) 표시 LED
3	전원(5V) 출력용 커넥터
4	시리얼 통신(UART) 커넥터
5	로보티스 다이나믹셀(XL-320) TTL 모터 연결 커넥터
6	배터리 소켓 홀더

7.3 라지그 Motor 보드

아래부터 커널연구회 라지그 모터 보드의 각 부품별 용법을 요약 설명한다.

커널연구회 RaZig Motor 보드



1	DC모터 1번 연결잭
2	DC모터 2번 연결잭
3	DC모터 3번 연결잭
4	DC모터 4번 연결잭
5	라지그 Sensor 보드와 연결하는 28핀 커넥터
6	사운드 입력 잭(라즈베리파이에서 출력되는 사운드를 이곳에 연결)
7	외부 확장용 연결 헤더핀(15핀)

7.4 라지그 보드 조립 방법

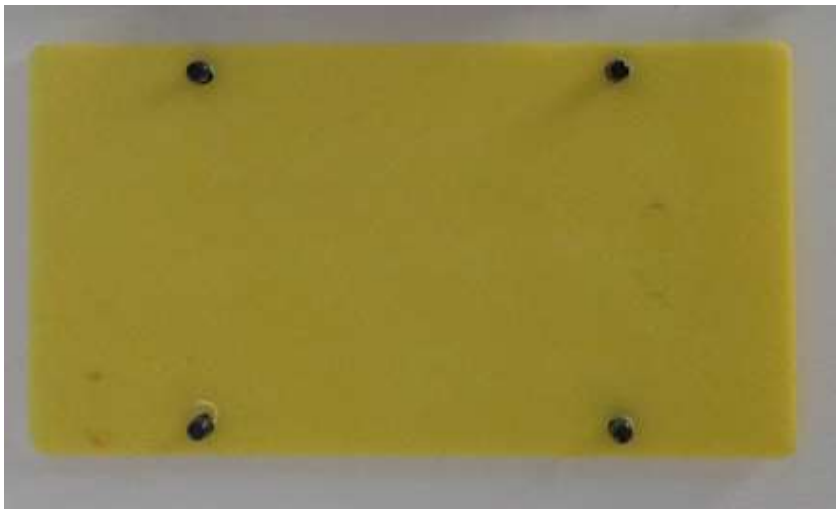
7.4.1 Senosr 보드 조립

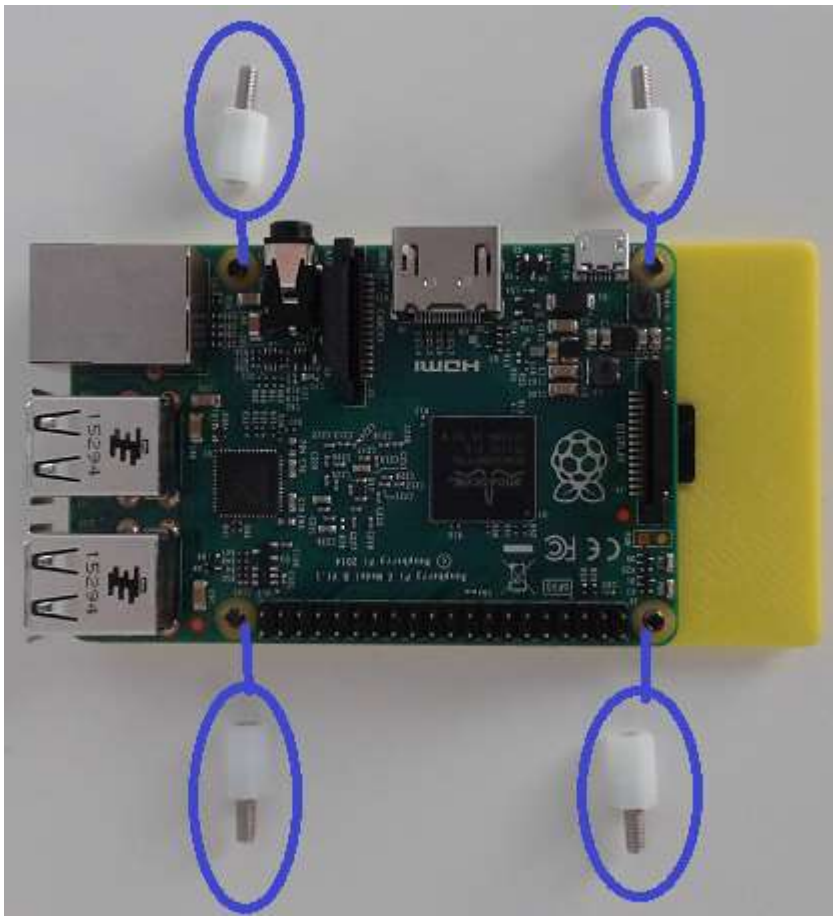
아래부터 라즈베리파이 보드와 라지그 센서보드를 조립하는 과정을 설명한다.

부품 전체

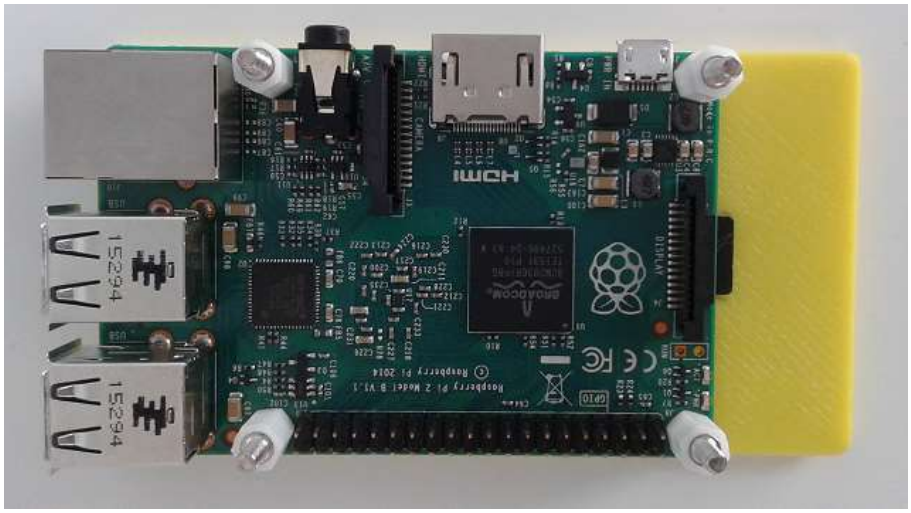
먼저 다음 사진과 같이 밑판에 라즈베리파이 보드를 올려서 너트와 지지대로 고정 시킨다.
지지대의 높이는 10mm 이다.

라즈베리파이 보드와 지지대 조립



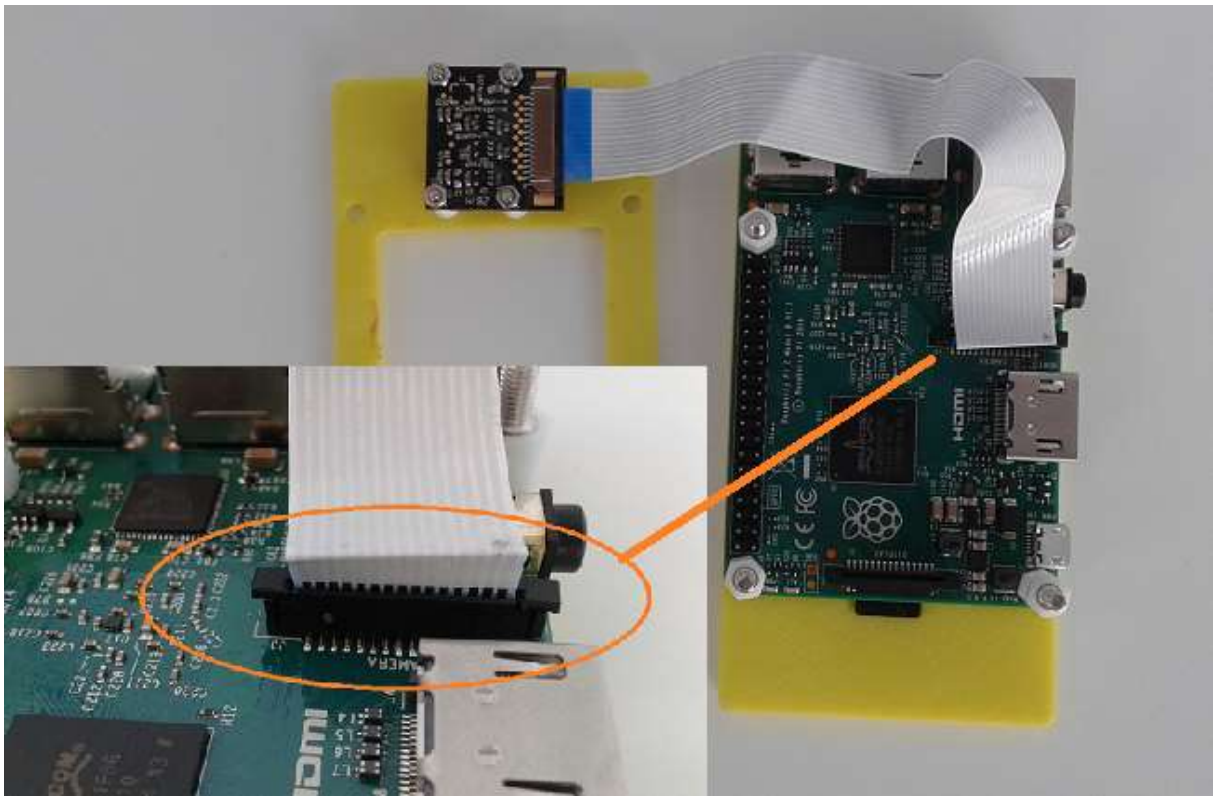


조립된 모습



그리고, 카메라 케이블을 아래 사진을 참고하여 장착한다.

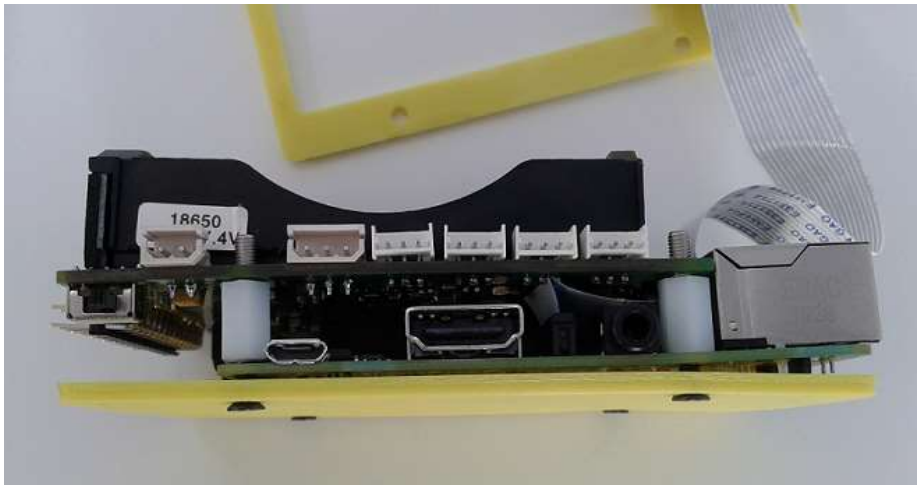
카메라 조립



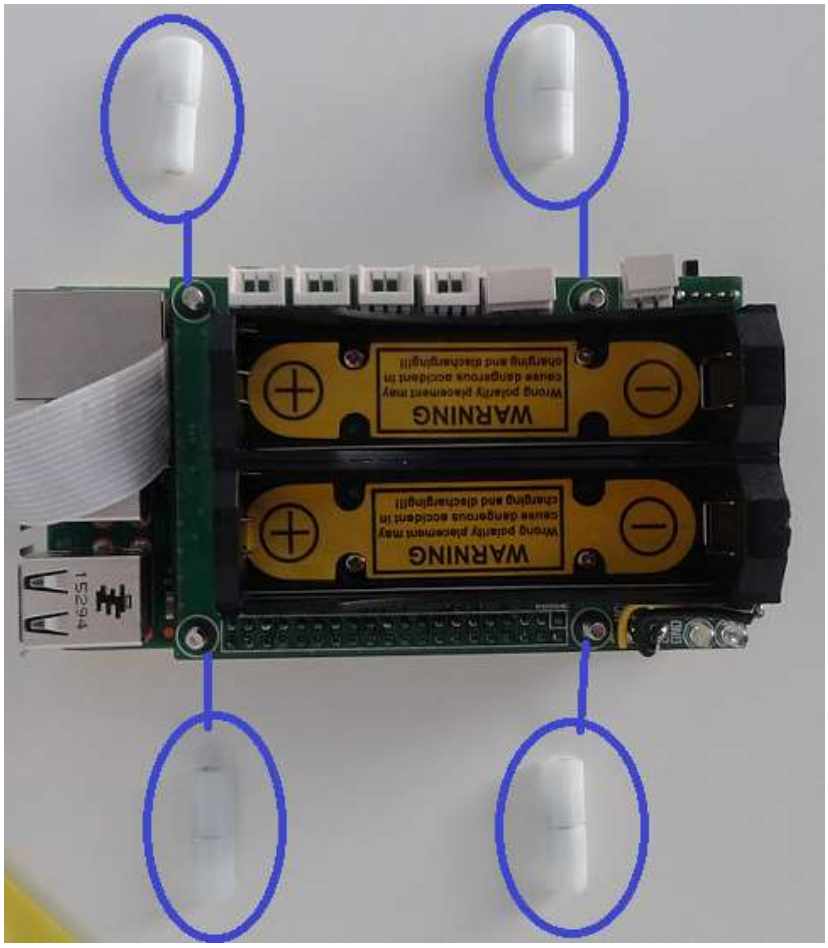
카메라를 장착한후, 아래 사진처럼 배터리 소켓이 이미 장착된 라지그 센서보드를 4 개의 지지대위에 올려서 고정 시킨다.

라지그 Sensor 보드 조립

라지그 센서보드를 조립할 때, 카메라 케이블이 부품들과 지지대 너트에 의해 손상되지 않도록 주의한다.

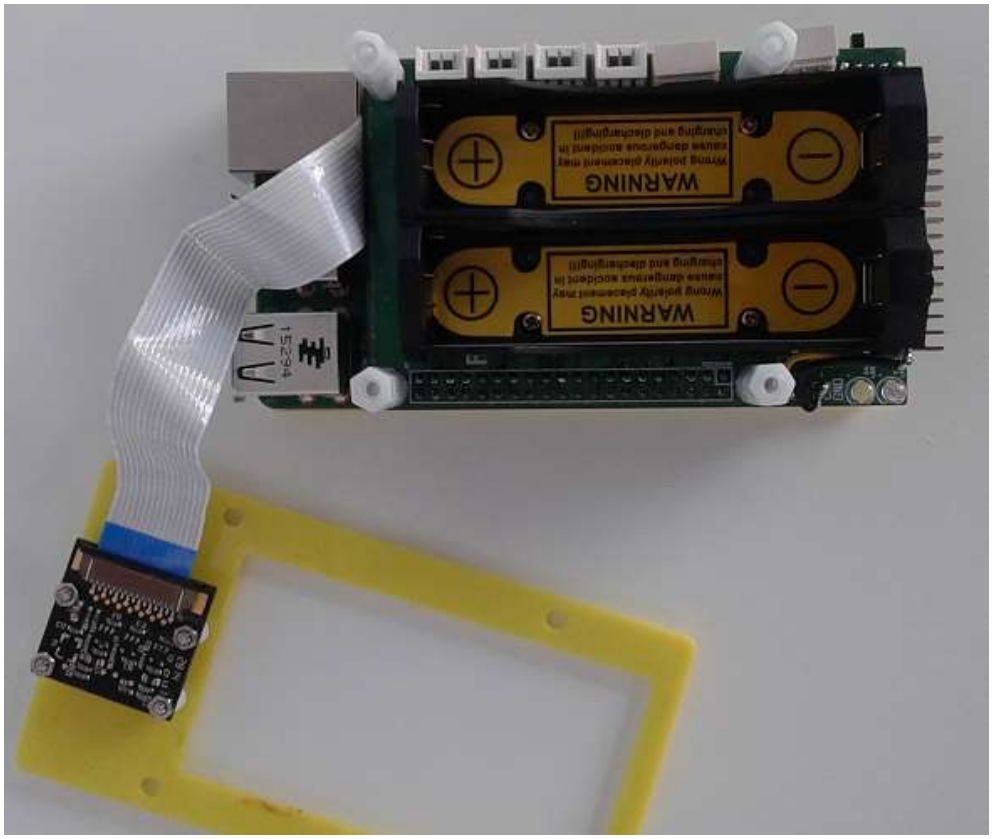


지지대 장착

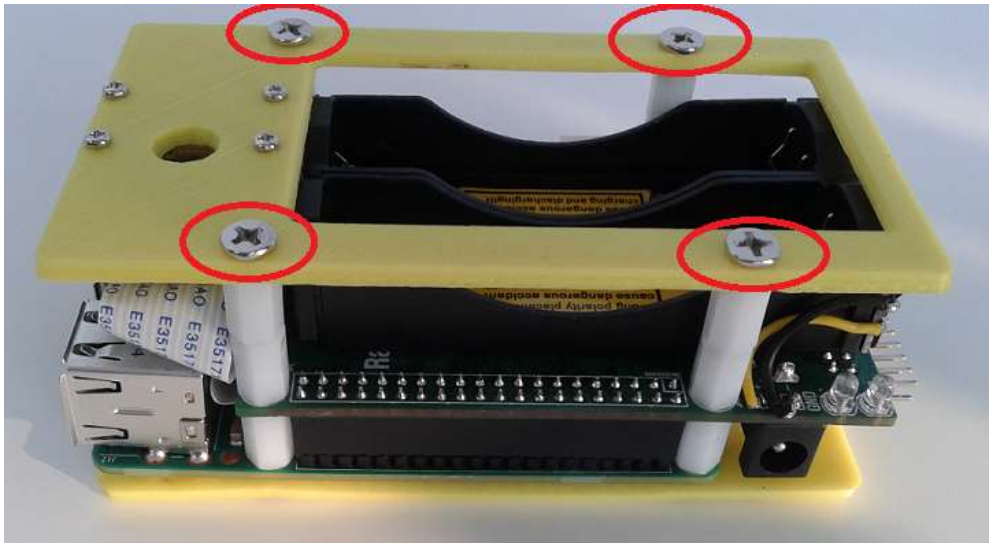


위의 사진과 같이 지지대(높이 20mm)로 라지그 센서 보드를 고정 시킨다.

라지그 Sensor 보드 조립된 모습

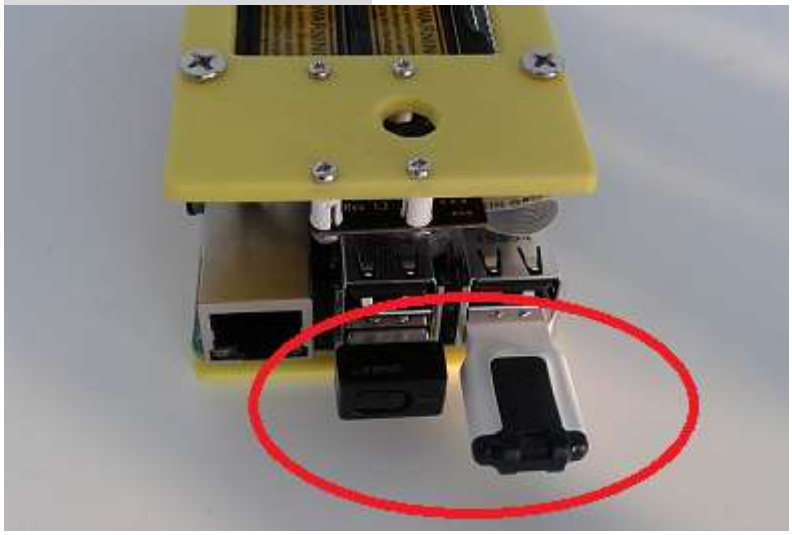


카메라 보드를 상단에 고정하여 조립완료



상단에 카메라를 보드를 올리고 4 개의 너트로 지지대와 함께 고정 시킨다.

WiFi 및 USB 메모리 장착



USB 와이파이 동글과 USB 메모리를 위의 사진처럼 장착한다.

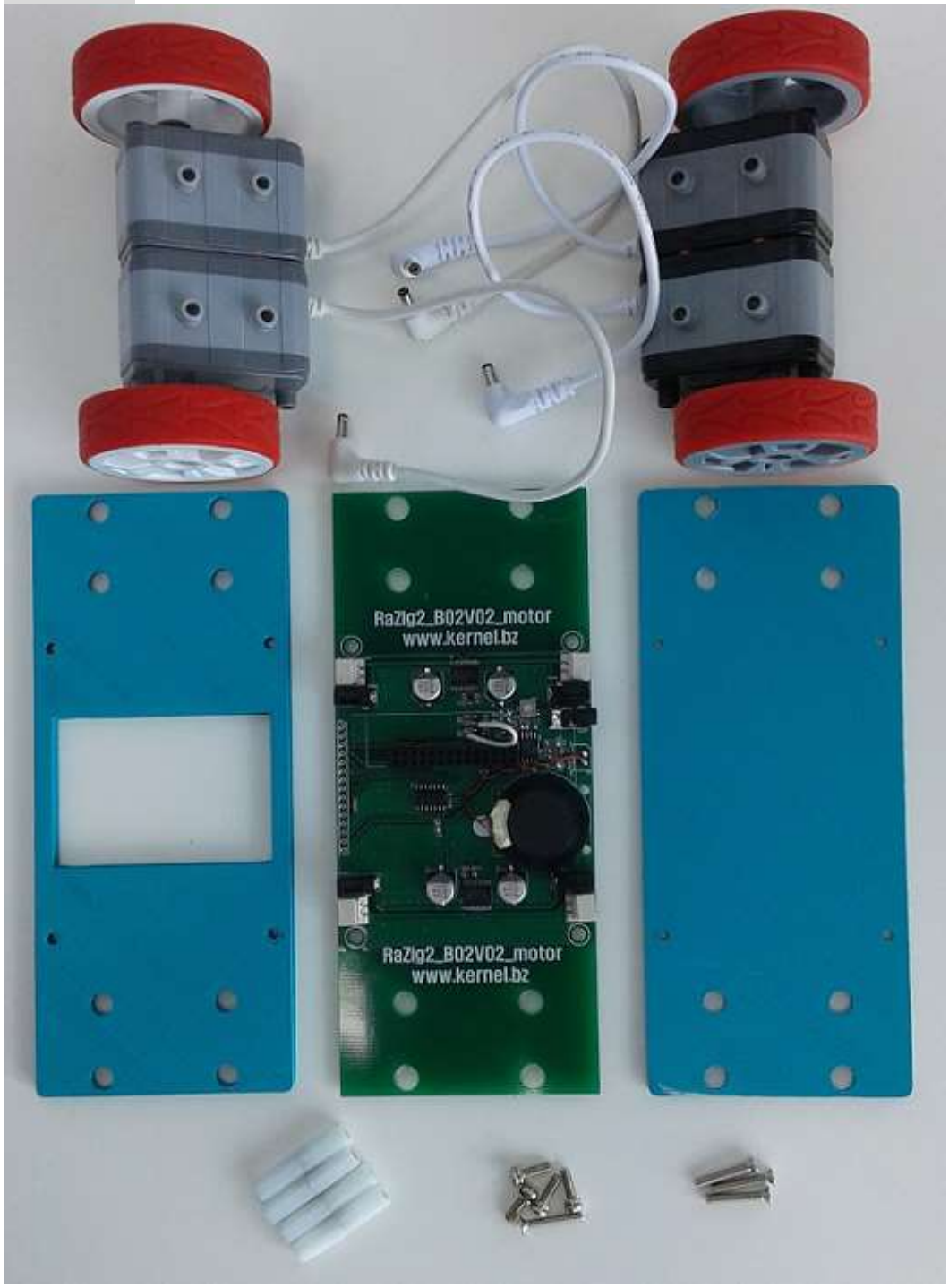
조립 완료된 모습

배터리는 1 개가 3.7V 이며 2 개를 합치면 7.4V 가 된다. 배터리는 외부에 충전용 소켓이 별도로 있으며 이곳에서 충전하여 라지그에 장착한다. 배터리를 장착할때 +/- 극성을 잘 확인하여 – 극성부터 소켓에 끼워 넣도록 한다. 배터리는 충전 및 지속시간이 평균 5 시간정도 된다.

7.4.2 Motor 보드 조립

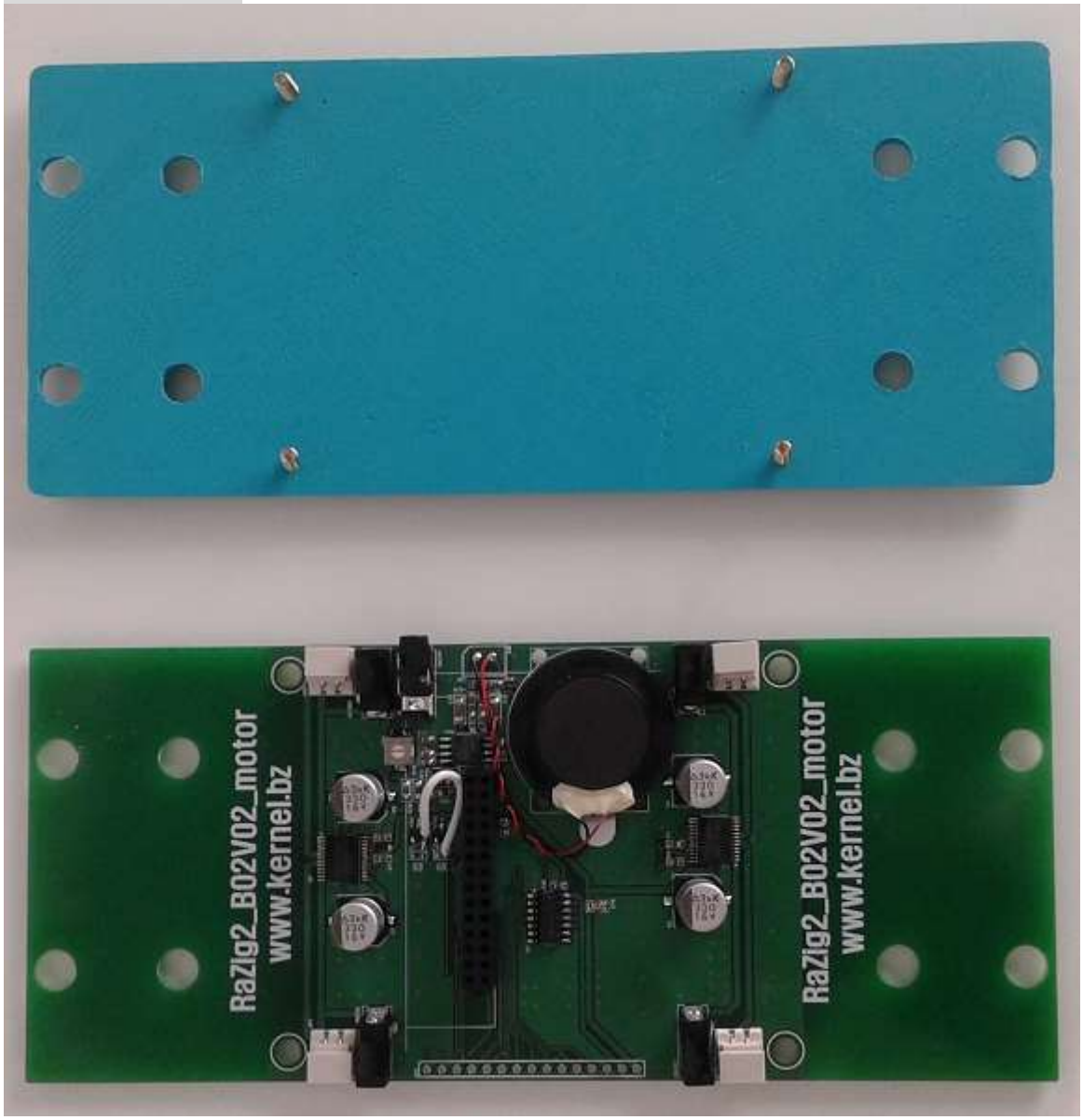
아래부터 라지그 모터 보드를 조립하는 방법에 대해서 설명한다.

부품 전체

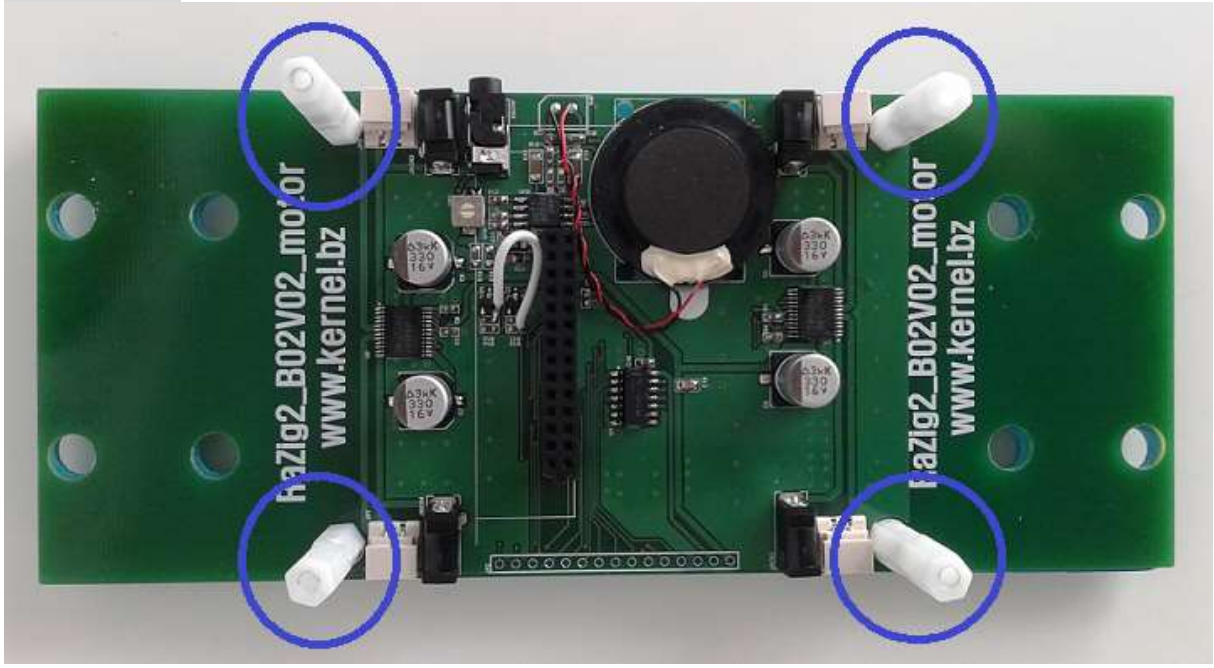


먼저 밑판에 너트를 장착하여 그위에 라지그 모터 보드를 올려서 조립한다.

하단부(밑판) 조립

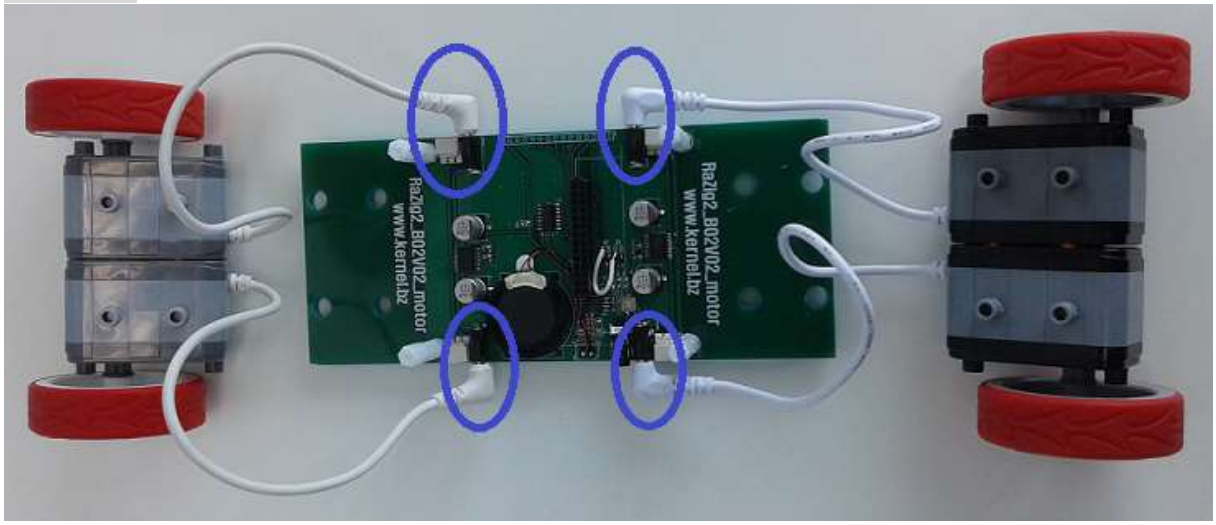


지지대 장착



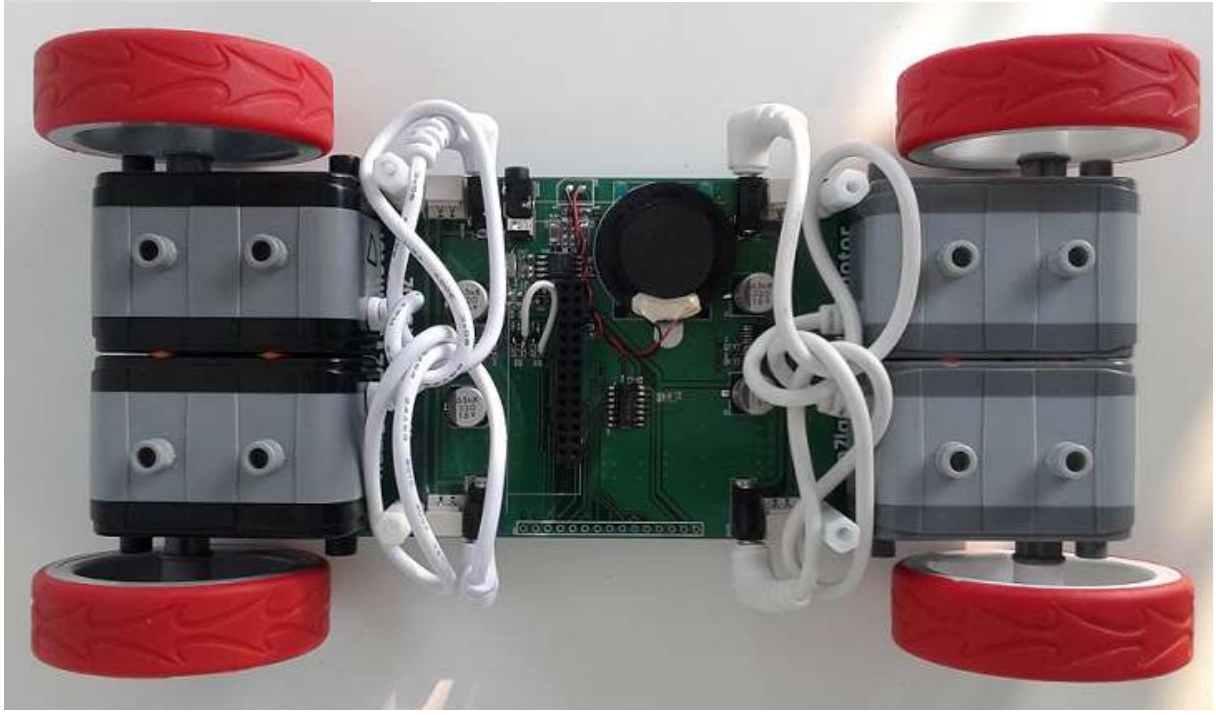
지지대의 높이는 30mm 이다.

모터 조립



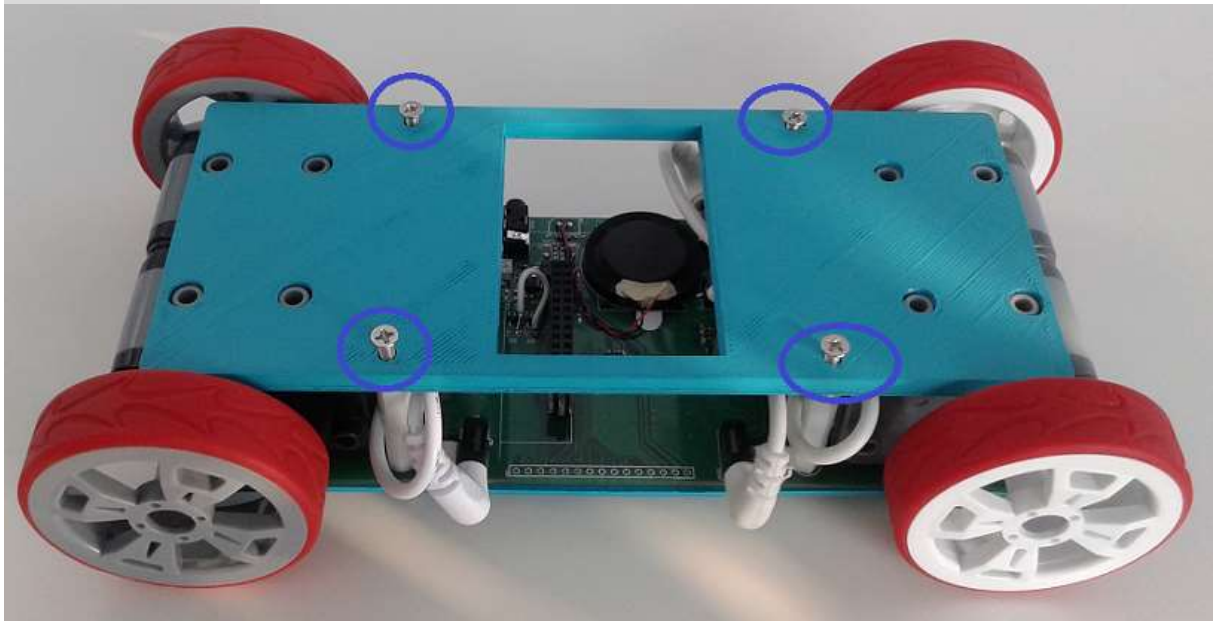
위의 사진처럼, 모터 연결선은 모터가 위치하고 있는 곳의 잭에 연결한다. 모터와 연결잭은 교차되지 않도록, 각각의 위치에 일치되도록 연결한다.

모터 장착 및 연결선 정리



모터 연결선은 위의 사진처럼 서로 묶어서 깔끔하게 정리한다.

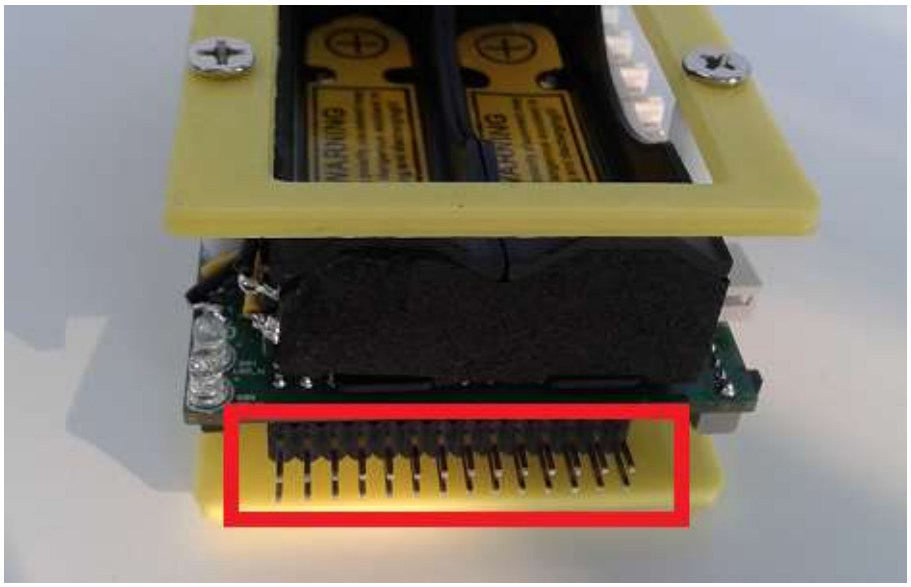
상단부(상판) 조립



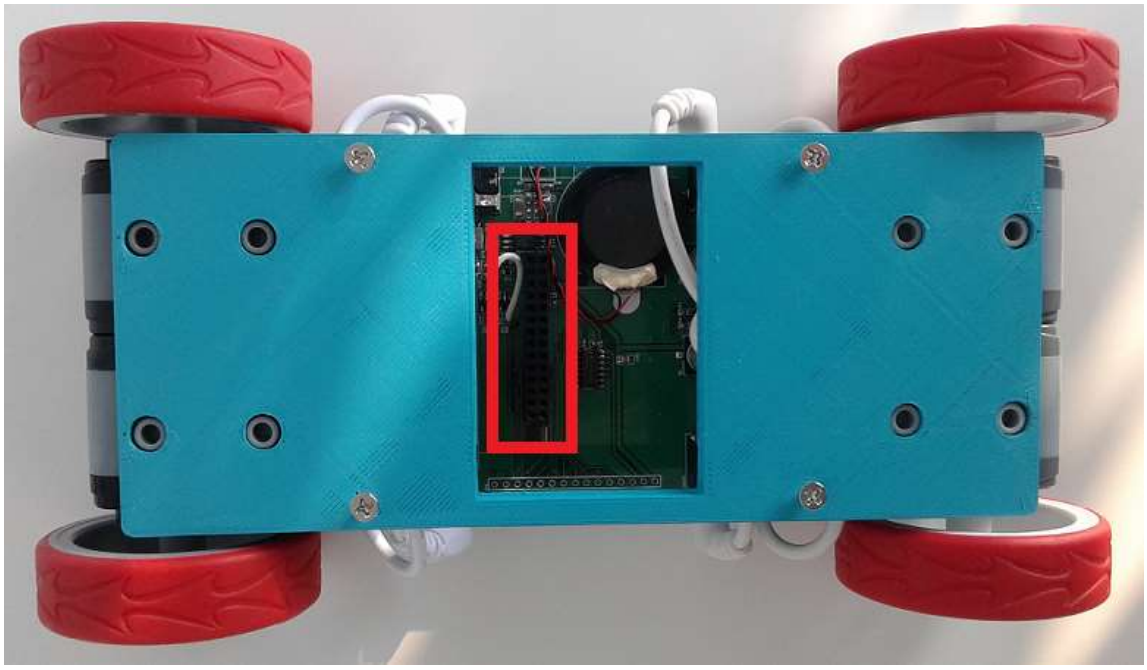
상단부(상판)을 올리고 4 개의 너트를 사용하여 지지대와 함께 고정 한다.

7.4.3 Senosr와 Motor 보드 연결

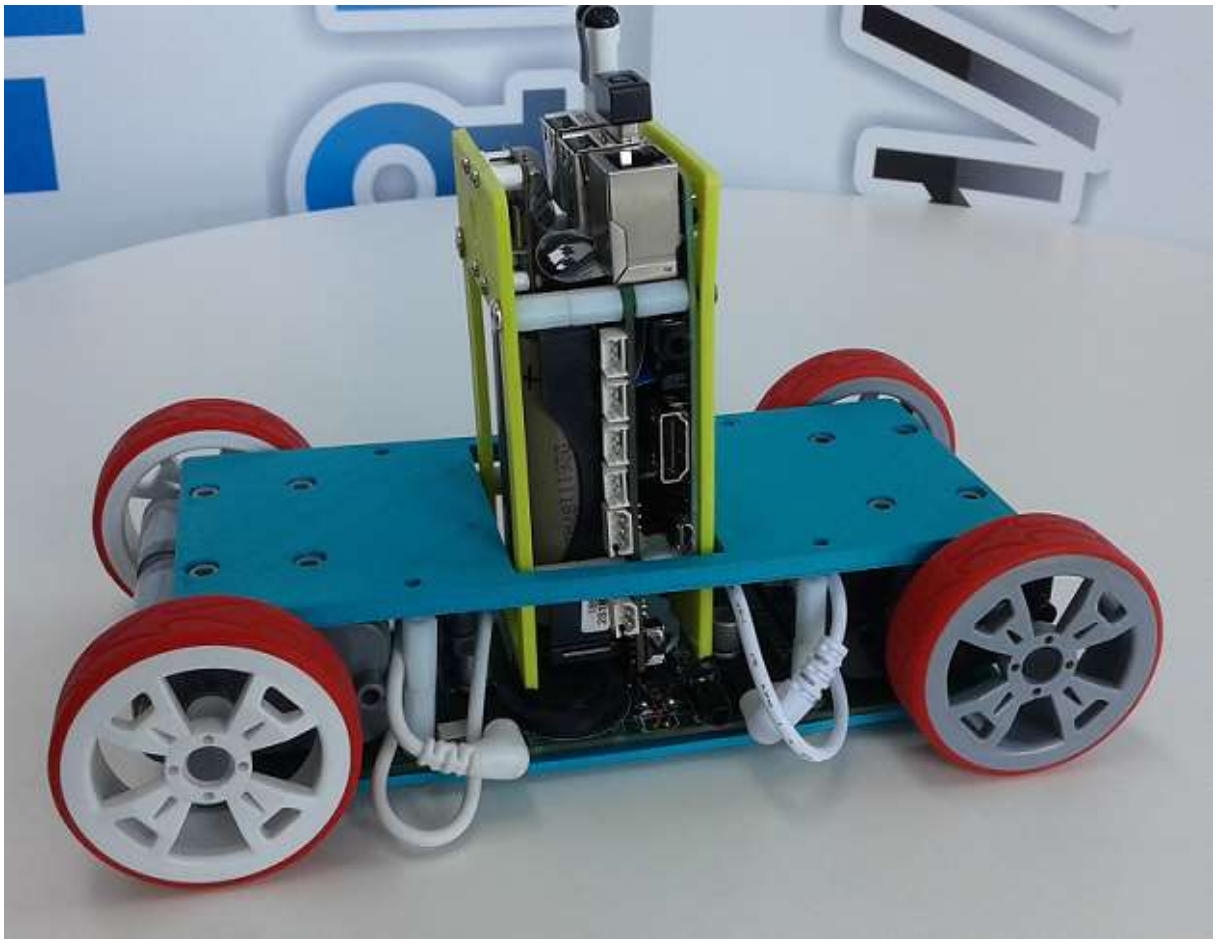
2 개의 모듈로 구성되어 있는 라지그 센서 보드와 모터 보드를 아래와 같이 서로 연결(도킹)할 수 있다.



라지그 센서 보드 하단부에 있는 28 핀 헤더를 라지그 모터 보드의 중앙에 있는 28 핀 소켓에 수직방향으로 끼워 넣어 장착한다.

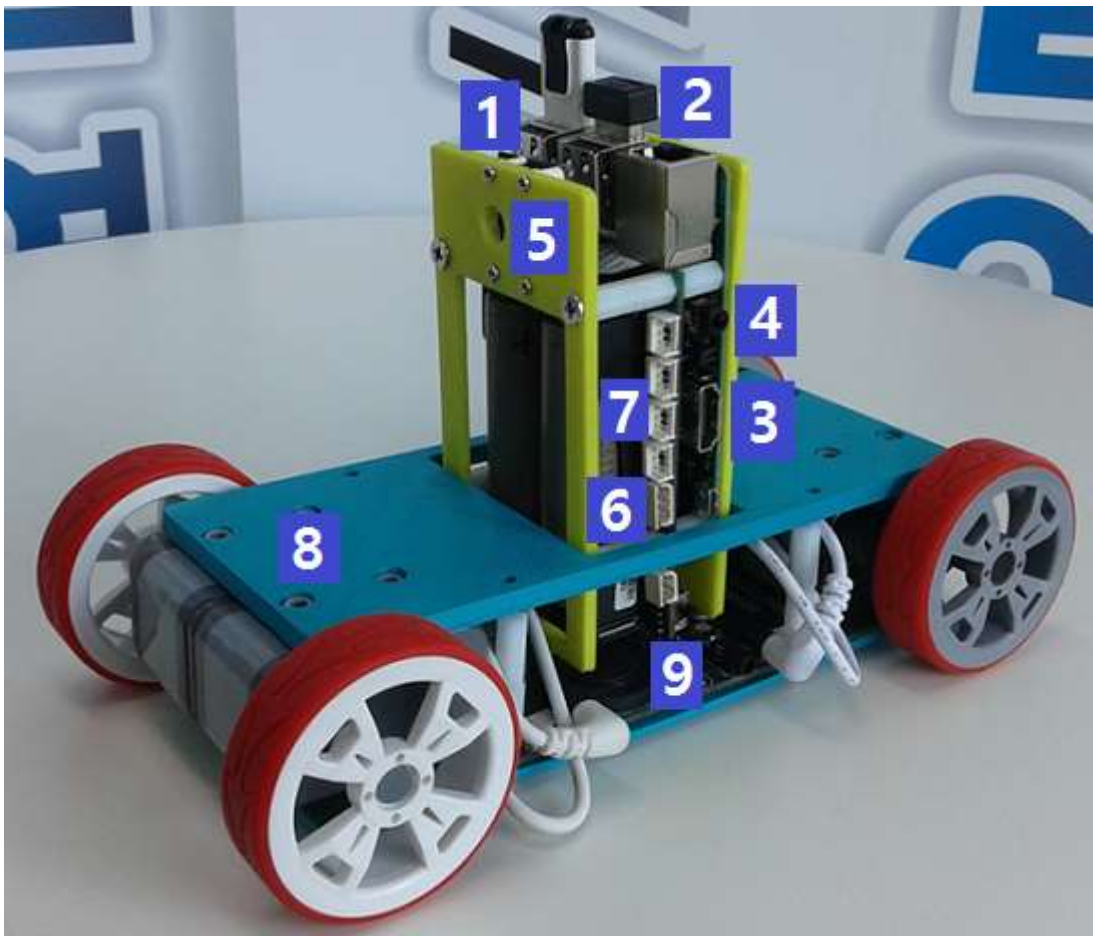


조립 완료된 전체 모습



7.5 라지그 기능 소개

라지그의 외형 기구물은 커널연구회에서 3D 프린터로 출력하여 조립한 것이며 전체적인 외형은 아래 사진과 같다. 모든 부품들을 직접 조립할 수 있도록 했다. 라지그에 구현되어 있는 기능들은 커널연구회의 교육과정을 통해서 사용자들에게 전달 한다. 라지그 제품들도 교육과정 등록을 통해서 교육비와 함께 구매할 수 있다. 아래부터 라지그에 탑재되어 있는 기능들을 요약하여 설명한다.



1	USB 키보드, 마우스, USB 메모리	6	시리얼(UART) 통신
2	WiFi 무선 및 이더넷 유선통신	7	로보티즈 TTL 모터(XL-320) 연결
3	HDMI 출력 (리눅스 PC 기능)	8	DC 모터 4개 사륜구동
4	라즈베리파이 사운드 출력	9	스피커 사운드 입력
5	카메라 영상녹화 및 움직임 감지	9	전원 ON/OFF 스위치

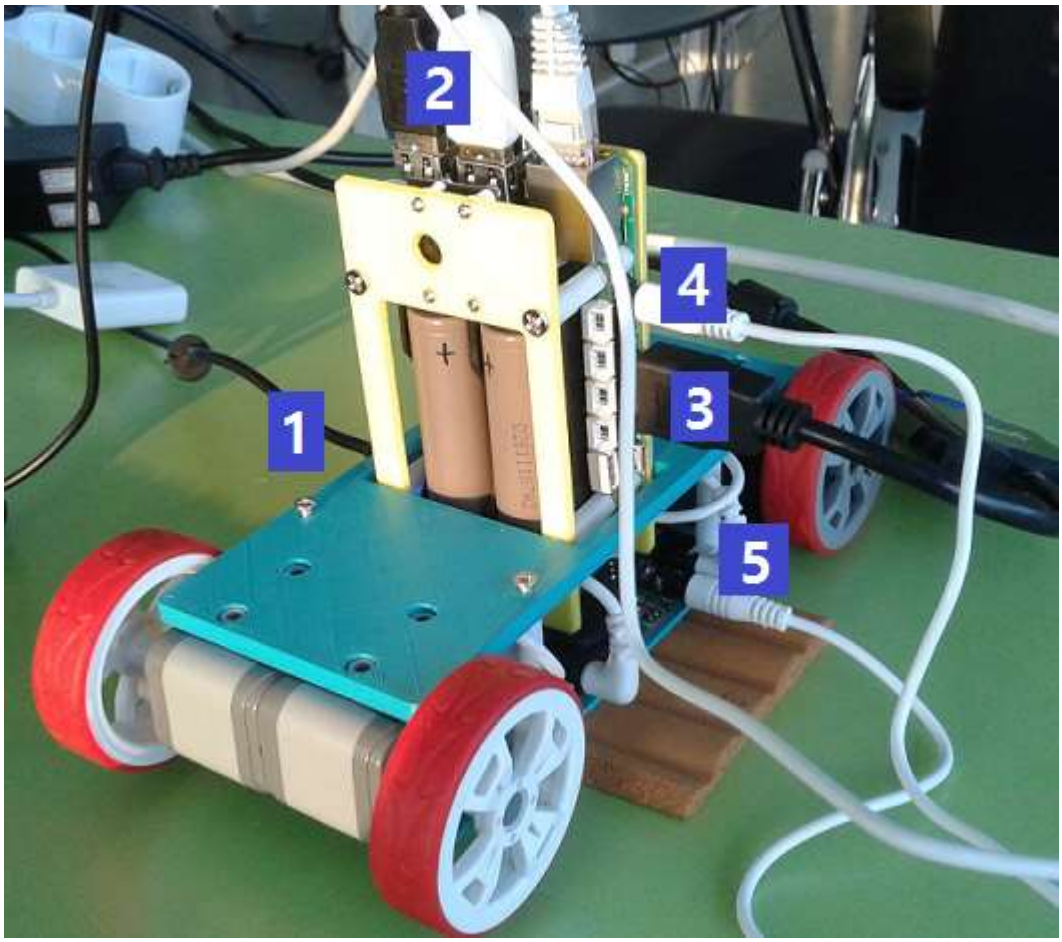
라지그에 탑재된 기능들은 모두 라지그 안에서 동작하고 있으며, 사용자는 휴대폰으로 라지그에 WiFi 무선접속하여 다음과 같은 Graphic User Interface로 기능들을 조작할 수 있다.

라지그 GUI 메인 메뉴

위의 기능들을 아래에서 요약하여 설명한다.

7.5.1 개인용 PC기능

라즈베리파이와 라지그를 인터넷에 연결(유선, 무선 모두 가능)하여 웹서핑을 할 수 있다. 리눅스가 설치되어 있어서 HDMI 출력단자에 외부 모니터를 연결하여 각종 문서작업, 프로그래밍, 게임 등 개인용 PC 기능을 수행할 수 있다.



1 **외부 어댑터(12V) 전원 입력**

2 **USB 키보드 마우스, USB WiFi 동글(무선통신), 이더넷 유선통신**

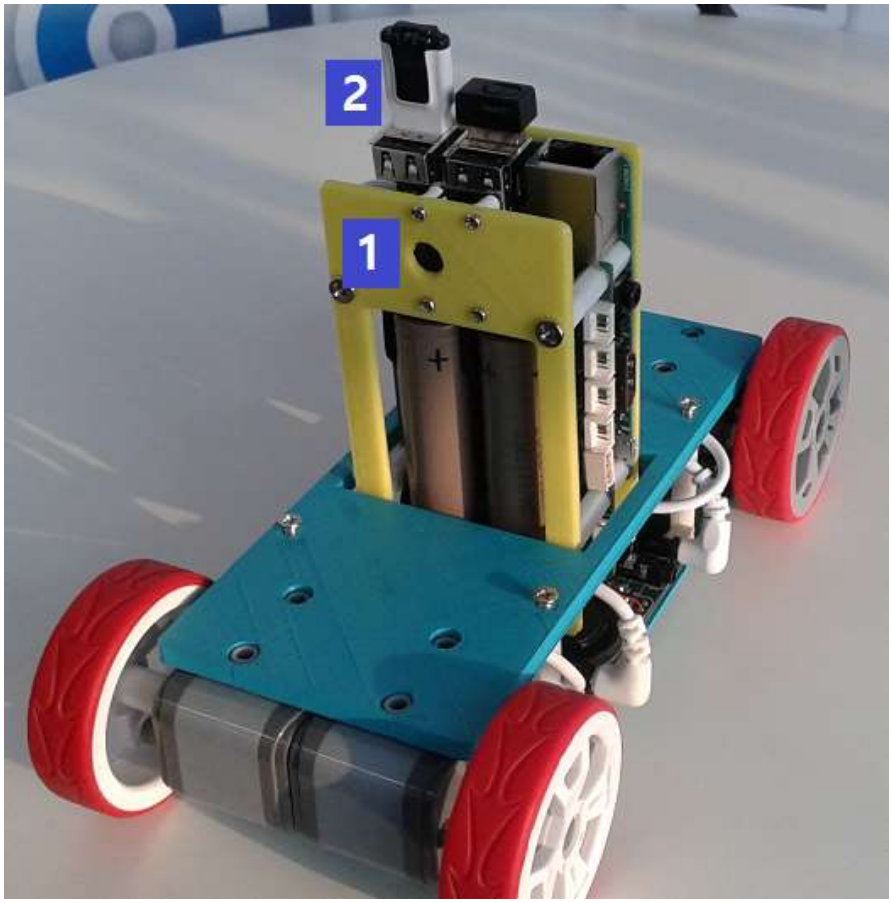
3 **HDMI 화면 출력(리눅스에서 각종 어플리케이션 실행)**

4 **라즈베리파이 사운드 출력**

5 **모터보드 스피커에 사운드 입력**

7.5.2 카메라 녹화기능

아래 사진의 1번에서 카메라 영상을 녹화하여 2번의 USB 메모리에 mp4 파일로 저장하고 이것을 녹화일시별로 재생 및 삭제할 수 있다.



위의 메뉴 기능은 휴대폰으로 라지그에 WiFi 접속하여 작업할 수 있다.

7.5.3 방범 카메라 기능

카메라에 모션 캡쳐기능이 있어서 움직임이 감지 되었을때 영상을 녹화하여 USB에 avi 파일로 저장하고, 저장일시별로 확인 및 삭제할 수 있다.

메인메뉴 **동작감시 파일목록**

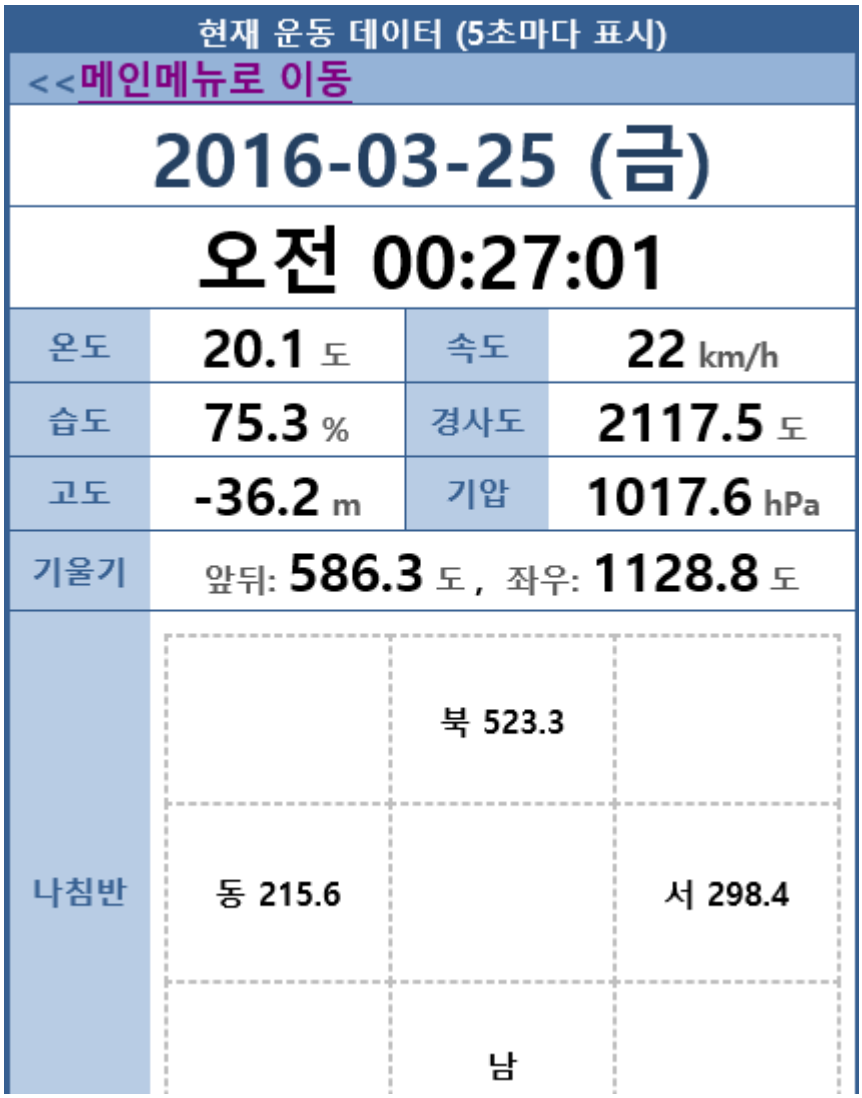
순번	파일명	생성일시
☐ 1	19-20151215171140.avi	2015-12-15 17:12:44
☐ 2	19-20151215171140-03.jpg	2015-12-15 17:12:44
☐ 3	18-20151215170956-02.jpg	2015-12-15 17:10:58
☐ 4	18-20151215170954.avi	2015-12-15 17:10:58
☐ 5	17-20151215165746-03.jpg	2015-12-15 16:58:58
☐ 6	17-20151215165744.avi	2015-12-15 16:58:58
☐ 7	16-20151215163049-01.jpg	2015-12-15 16:33:08
☐ 8	16-20151215163044.avi	2015-12-15 16:33:08
☐ 9	15-20151215162908-04.jpg	2015-12-15 16:30:14
☐ 10	15-20151215162907.avi	2015-12-15 16:30:14
☐ 11	14-20151215155534-03.jpg	2015-12-15 15:56:44

전체선택 삭제하기 메인메뉴

위의 메뉴 기능은 휴대폰으로 라지그에 WiFi 접속하여 작업할 수 있다.

7.5.4 센서 데이터 수집 기능

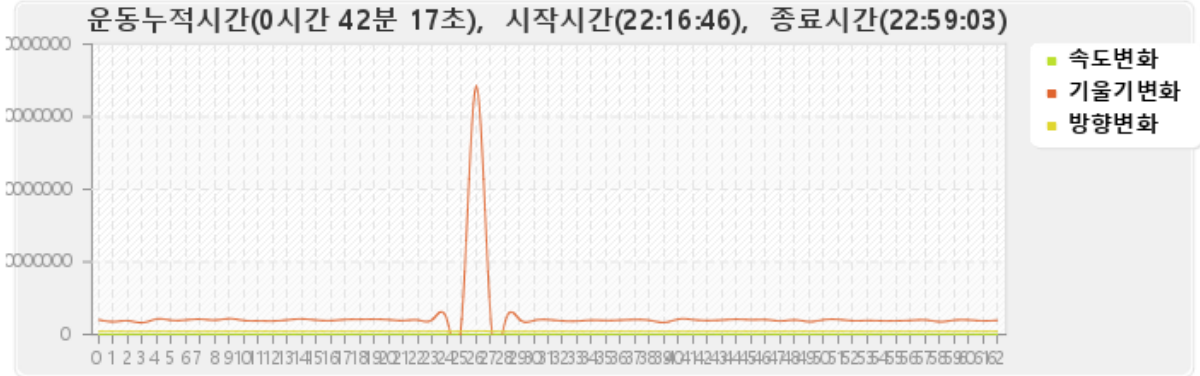
라지그 센서보드에 장착되어 있는 센서들을 사용하여 온도, 습도, 기울기, 이동속도, 방향 정보들을 수집한 데이터를 휴대폰으로 WiFi 접속하여 다음과 같이 확인할 수 있다.



위와 같이 산출된 데이터를 통계처리하여 아래와 같은 그래프로 출력한다.

● 온도, 습도, 기압, 고도 변화 그래프

● 운동량 (X, Y, Z) 변화 그래프



7.5.5 음악파일 재생 기능

즐겨듣는 음악파일을 USB에 저장하여 재생 우선순위를 부여한 순서대로 음악을 들을 수 있다.

메인메뉴 음악듣기 파일목록

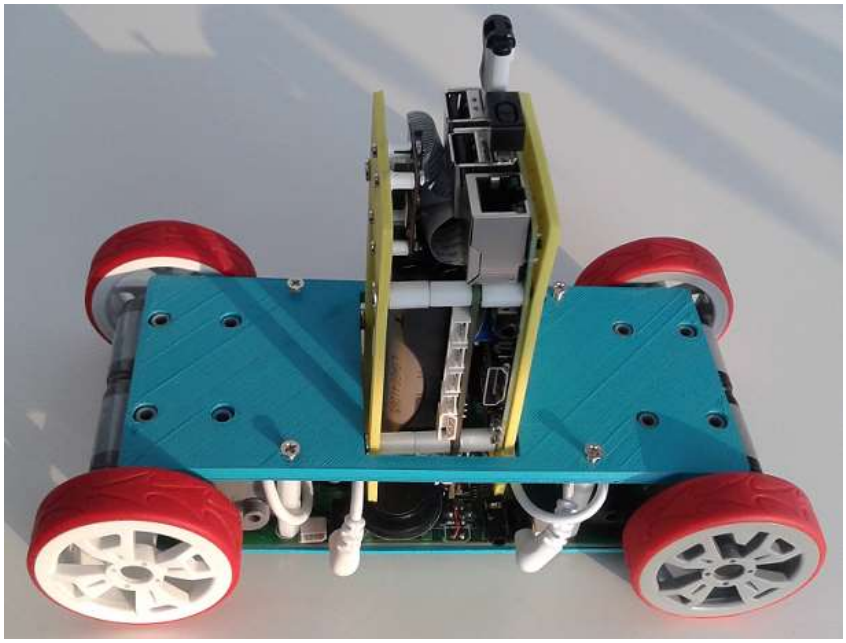
순번	파일명	생성일시
1	Ennio Morricone - Gabriel`s Oboe (Live in Sofia).mp3	2015-10-25 08:22:02
2	nella_fantasia_oboe.mp3	2015-10-25 08:17:28
3	Ennio Morricone - Gabriel`s Oboe.mp3	2015-10-25 08:15:36
4	Sun Woo - Nella Fantasia.mp3	2015-10-25 08:08:40
5	88.트위스티드페이트 - 행운의 여신이 내게 미소를 짓는군.mp3	2014-03-04 00:49:52
6	12.니달리 - 야생을 두려워하게 만들어주지.mp3	2014-03-04 00:44:30
7	9.나서스 - 우리는살고저들은죽을것이다.mp3	2014-03-04 00:44:16
8	8.나서스 - 삶과 죽음의 순환은 계속된다.mp3	2014-03-04 00:43:46
9	37.미스포춘 - 행운은 멍청이를 싫어하는 법이지.mp3	2014-03-04 00:42:52
10	31.마스터이 - 나의검은 당신의 것이오.mp3	2014-03-04 00:42:36
11	32.말파이트 - 바위처럼 단단하게.mp3	2014-03-04 00:42:04

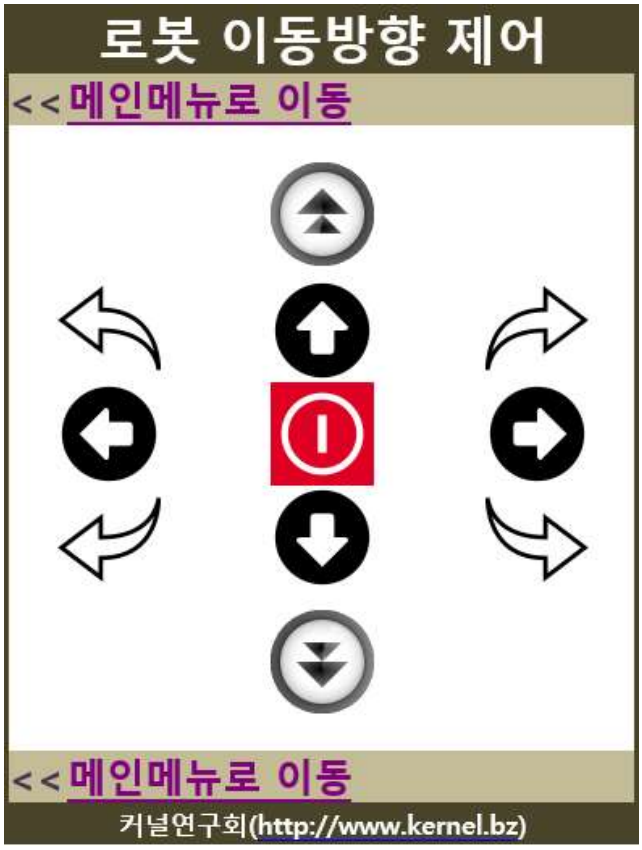
7.5.6 동영상 및 라디오방송 듣기

외부 인터넷이 연결되어 있다면, 인터넷 라디오 채널별로 라디오를 청취할 수 있다.

7.5.7 로봇 자동차 제어기능

라지그 센서보드를 모터보드에 장착하여 DC모터 4개를 사륜구동하여 로봇 자동차의 움직임을 제어할 수 있다.





7.6 라지그 프로젝트 교육안내

커널연구회 라지그(RaZig) 프로젝트에 참여하기 위해서는 C언어와 자료구조 알고리즘 및 리눅스 시스템 프로그래밍에 대해서 어느 정도 사전 지식을 갖추고 있어야 한다. 커널연구회는 C언어와 자료구조 알고리즘 교육 과정과 리눅스 시스템 프로그래밍 교육과정을 마련하고 있으므로 아래 링크를 참조해 주기 바란다.

커널연구회 교육과정 상세 안내 링크

http://dev.kernel.bz/doc/kernel_bz_edu_detail_intro.htm

라지그 프로젝트는 C언어와 자료구조 알고리즘 및 리눅스 시스템 프로그래밍 과정을 이수하신 분들을 대상으로 아래와 같이 교육을 진행한다. 교육 기간은 2개월이나 상황에 따라서 변경될 수 있으므로 커널연구회 웹사이트(http://www.kernel.bz)를 자주 참조해 주기 바란다.

라지그 프로젝트 교육내용

순번	교육 제목	상세 내용
1	라즈베리파이 개발환경	라즈베리파이 개발환경 설정방법 통합개발도구(Code Blocks) 설치 디바이스 포팅 사운드, 비디오, USB WiFi 동글, 카메라 모듈 포팅 GPIO 헤더핀맵 이해
2	리눅스 시스템 프로그래밍	오픈소스 사용법 설명 카메라 제어 프로세스 코딩(쓰레드, 데몬, 시그날) 카메라 동작감시 프로세스 코딩(쓰레드, 데몬, 시그날) 음악 및 동영상 재생 프로세스 코딩(쓰레드, 데몬, 시그날) 인터넷 라디오 플레이 프로세스 코딩(쓰레드, 데몬, 시그날)
3	웹서버프로그래밍	아파치, PHP, MySQL 연동방법 PHP 웹페이지와 C언어 시스템 프로그래밍 연동방법 스마트폰과 USB WiFi 무선 인터페이스 방법

4	모터 제어 프로그래밍	모터 제어 보드의 이해 PWM 인터페이스 이해 및 코딩 스마트폰으로 모터제어 인터페이스 코딩 PHP와 C언어로 로봇제어 프로그래밍 코딩
5	센서 데이터 수집 프로그래밍	라지그 센서 보드의 이해 I2C 인터페이스 이해 및 코딩 스마트폰으로 센서 데이터 수집 인터페이스 코딩 PHP와 C언어로 센서 데이터 통계 및 그래프 처리 기능 코딩

커널연구회 라지그(RaZig) 프로젝트에 참여하기 위해서는 아래와 같은 양식의 수강신청서를 작성하여 커널연구회에 메일(rgbi3307@nate.com)로 신청하면 된다.

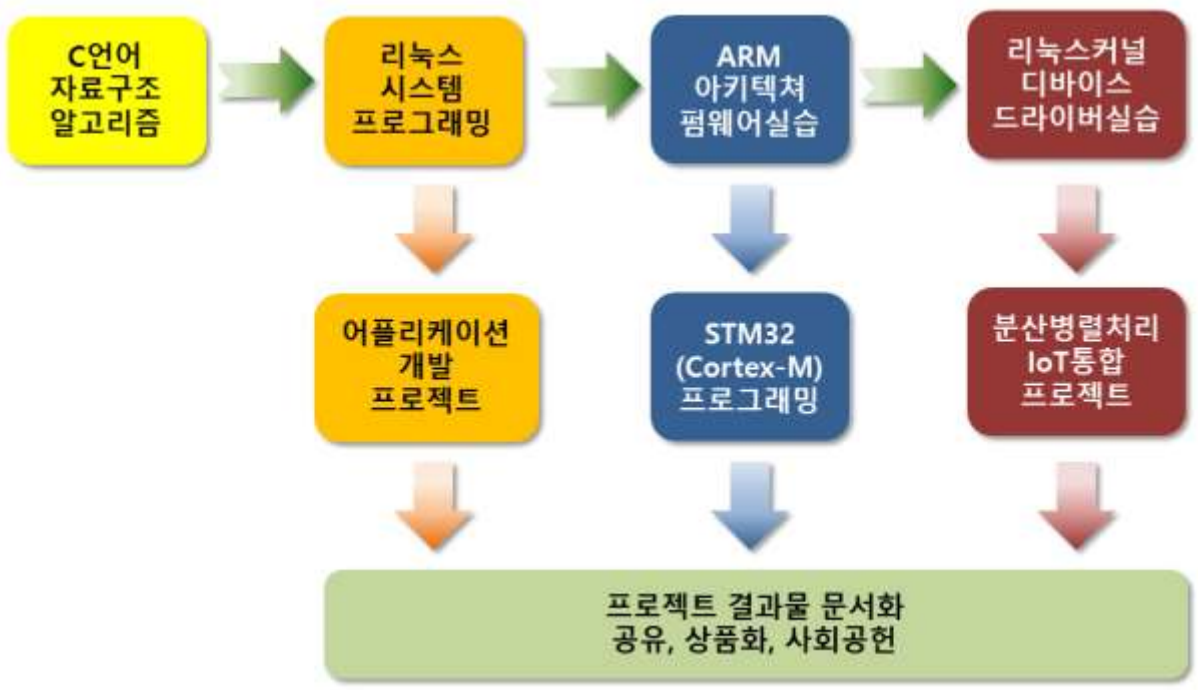
커널연구회 라지그 프로젝트 참가 신청서

* 이름	
* 이메일	
* 전화번호	
직업	**일반(직장)인 / 대학생**
신청 프로젝트명	**커널연구회 라지그(RaZig) 프로젝트**
선행학습수준 (1개 이상 선택)	0. 위의 프로젝트에 대해서 사전지식 (초급/중급/고급) 있음. 1. C언어와 자료구조 알고리즘 (초급/중급/고급) 수준. 2. 리눅스 시스템프로그래밍 (초급/중급/고급) 수준. 3. 웹(PHP, MySQL) 프로그램 (초급/중급/고급) 수준. 4. 3D 프린터 3D 도면 (초급/중급/고급) 수준. 5. 하드웨어 회로 설계 (초급/중급/고급) 수준.

부록B. 커널연구회 교육과정 상세안내

B1.0 커널연구회 교육과정 로드맵

커널연구회 교육학원은 다음과 같은 훈련과정 개발 체계를 갖추고 있습니다. 먼저 프로그래밍의 가장 기본이 되는 "C언어와 자료구조 알고리즘"에 대해서 자체적으로 출판한 교재를 통하여 교육생들이 확실이 기초를 다질 수 있도록 교육한후, 이것을 바탕으로 "리눅스 시스템 프로그래밍" 과정을 익혀서 어플리케이션을 개발할 수 있도록 합니다. 그리고 "ARM 아키텍쳐 펌웨어 실습" 과정을 거쳐서 STM32 펌웨어 프로그래밍을 할 수 있는 역량을 배양합니다. 위의 과정들을 마치게 되면 "리눅스 커널과 디바이스드라이버 실습" 과정을 통하여 IoT 환경의 프로젝트를 수행하도록 하는 훈련과정 개발 체계를 갖추고 있습니다. 최종적으로는 교육과정에서 산출한 내용물과 교육 프로젝트 수행 결과물들을 서로 공유하여 사회에 공헌하는 체계로 접근하고 있습니다.



B1.1 C언어와 자료구조 알고리즘

과정개요

본 과정은 C언어를 체계적으로 학습한 후 자료구조와 알고리즘을 C언어로 실습하여 이해할 수 있는 교육과정입니다. C언어의 역사는 오래 되었고 많은 컴퓨터 프로그래밍 분야에서 사용되고 있습니다. 또한, 자료구조와 알고리즘은 컴퓨터 프로그래밍의 기반을 이루는 초석입니다. 응용은 기반이 되는 기초지식을 토대로 이루어집니다. 오늘날 많은 공학적 응용제품들은 자연과학(수학, 화학, 물리)의 토대 위에서 만들어졌습니다. 컴퓨터 응용 프로그램들 또한 자료구조와 알고리즘 기반에서 코딩 되었으며, C언어는 자료구조와 알고리즘을 구현하는데 유용한 컴퓨터 프로그래밍 언어의 선구자이며 C언어를 잘 학습하면 여러가지 어플리케이션을 작성할 수 있습니다.

과정목표

- **C 언어의 형태, 연산자, 표현, 제어흐름을 이해할 수 있습니다.**
- **조건판단, 반복문, 분기에 대해서 익힐 수 있습니다.**
- **문장과 블록, 함수, 영역(scope)에 대해서 알 수 있습니다.**
- **매크로, 전처리기, 포인터와 주소, 함수, 배열에 대해서 이해합니다.**
- **주소연산, 포인터 배열, 함수 포인터, 복잡한 포인터를 해석 합니다.**
- **객체 지향 코딩, 모듈화, 라이브러화, 구조적 프로그래밍을 익힐 수 있습니다.**
- **반복, 재귀, 포인터, 정렬 알고리즘에 대해서 이해할 수 있습니다.**
- **Linked List, Stack, Queue 를 실습합니다.**
- **Hashing, Binary Search Tree 에 대해서 이해하게 됩니다.**

과정장점

본과정은 강사가 직접 집필한 "C 언어와 자료구조 알고리즘" 교재를 가지고 강의하므로, 교육내용을 확실히 이해할 수 있도록 정확하게 전달합니다. 강사가 실무에서 경험한 다양한 예제들을 가지고 교습하므로 쉽고 재미있게 학습할 수 있습니다.

사전지식

사전 지식이 없어도 C 언어는 학습할 수 있습니다.

강의내용

순서	주요내용	상세설명
1	**형태, 연산자, 표현 이해**	변수명 / 데이터 타입(DATA TYPE)과 크기 / 상수 / 선언 / 산술 연산자들 / 관계와 논리 연산자들 / 형변환 / 증감 연산자 / 비트단위 연산자들 / 할당 연산자와 표현 / 조건 표현 / 연산 순위
2	**제어 흐름 이해**	문장과 블럭 / IF-ELSE / ELSE-IF / SWITCH / 루프 - WHILE, FOR / 루프 DO-WHILE / BREAK 와 CONTINUE / GOTO 와 LABELS
3	**함수와 프로그램 구조**	함수의 기초 / 외부 변수들 / 영역(SCOPE) 규칙들 / 헤더 파일들 / 고정(STATIC) 변수들 / 레지스터 변수들 / 블록 구조 / 초기화 / 재귀(RECURSION) / 전처리기(PREPROCESSOR)
4	**포인터와 배열**	포인터(POINTER)와 주소(ADDRESS) / 포인터와 함수 매개변수 / 포인터와 배열 / 주소 연산 / 문자 포인터와 함수 / 포인터 배열 / 다차원 배열 / 포인터 배열의 초기화 / 포인터 배열과 다차원 배열 / 명령라인 매개변수 / 함수를 가르키는 포인터 / 복잡한 포인터 선언들 이해
5	**구조체 이해**	구조체의 기본 / 구조체와 함수 / 구조체 배열 / 구조체 포인터 / TYPEDEF / UNIONS / 비트 필드 / 구조체 PADDING
6	**반복, 재귀 이해**	반복과 재귀를 통한 알고리즘 효율성 이해
7	**정렬 알고리즘 이해**	Bubble / Insertion / Merge / Quick / Shell / Heap 정렬
8	**Linked List 이해**	Linked List / Stack / Queue 실습
9	**Tree 이해**	Hashing / Binary Search Tree 실습

강의교재

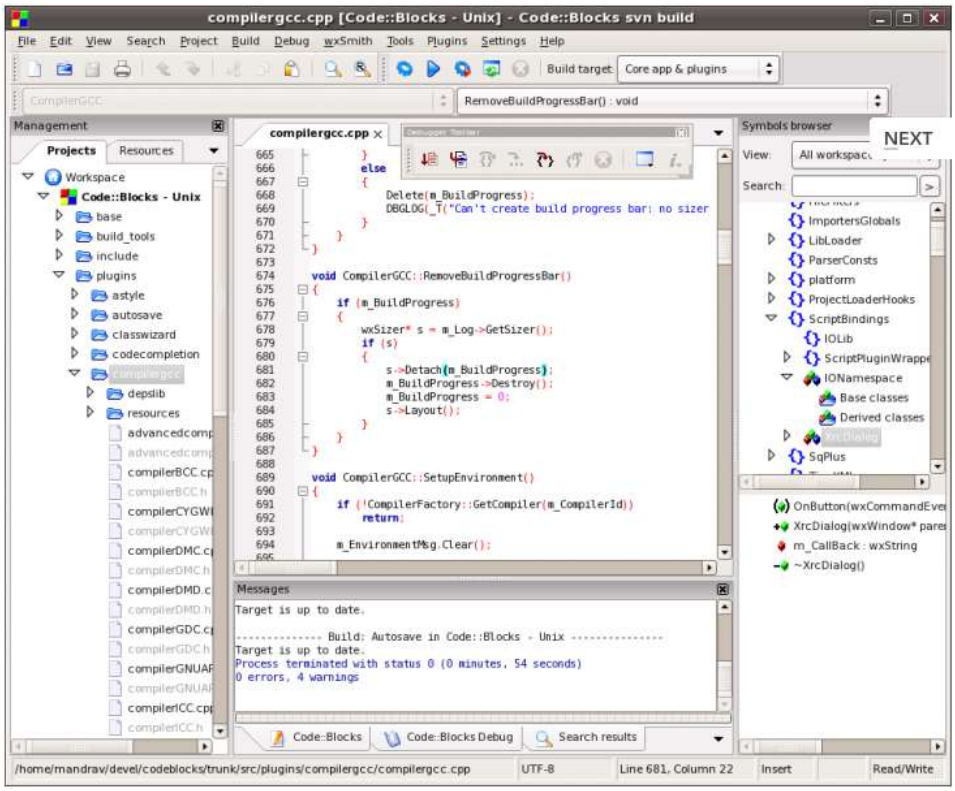
도서명: C 언어와 자료구조 알고리즘
출판사: 커널연구회
저자: 정재준
도서정보 링크(시중서점): http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=16931344

C 언어와 자료구조 알고리즘을 예제 소스코드를 통하여 실습할 수 있도록 구성한 책. C 언어의 형태, 연산자, 표현, 제어흐름, 함수, 배열, 포인터, 구조체등을 잘 설명하고 있으며, 특히 핵심적인 내용을 빠짐없이 잘 전달하고 있다. 아울러 깔끔하게 정리한 예제 소스코드를 직접 실습해 볼 수 있도록 했고, 실행결과를 확인하여 이론적으로 학습한 내용을 직접 확인하여 익히도록 했다. C 언어에 대한 저자의 오랜 경험을 토대로 설명을 아주 쉽고 간결하게 하고 있다. 제 1 부에서 익힌 C 언어를 사용하여 제 2 부에서는 자료구조 알고리즘을 체계적으로 익히도록 유도하고 있다. 특히, 자료구조와 알고리즘을 이론적으로 설명하는데 그치지 않고 C 언어 예제 소스코드를 통하여 독자들이 직접 실습해 볼 수 있도록 내용을 알차게 구성하였다. 자료구조와 알고리즘 설명도 필자의 오랜 경험을 바탕으로 기초가 되는 반복, 재귀, 정렬에서부터 스택, 큐, 링크드 리스트, 해싱, 트리등을 핵심 위주로 잘 설명하고 있다.

실습장비

ARM Cortex-A7(core 4 개) 기반의 라즈베리파이 2 보드에서 gcc 버전 4.9.2 로 실습합니다.

통합개발 환경(IDE)인 CodeBlocks 을 사용하여 소스 편집, 컴파일, 실행, 디버깅 합니다.



B1.2 리눅스 시스템 프로그래밍

과정개요

리눅스 시스템 프로그램은 리눅스 시스템이 지원하는 라이브러리 함수를 시스템 호출하는 방식으로 동작합니다. 리눅스 커널과 GCC 컴파일러가 제공하는 표준 라이브러리를 사용하는 방법을 익혀서 C 언어로 코딩하는 작업이므로 C 언어에 대해서 사전지식이 있으면 충분히 학습할 수 있습니다. 리눅스 시스템 프로그래밍은 모든 프로그래밍의 원천기술에 해당하므로 본 과정을 잘 익혀두면 이것으로부터 파생되는 대부분의 응용프로그래밍을 쉽게 익힐 수 있습니다.

과정목표

- **시스템 호출(System Call)의 개념을 이해할 수 있습니다.**
- **각종 하드웨어 장치들과 입출력(I/O)을 수행하는 방법을 익힐 수 있습니다.**
- **프로세스와 쓰레드에 대해서 익히고 C 언어로 코딩할 수 있습니다.**
- **프로세스간의 신호(Signals)에 대해서 이해할 수 있습니다.**
- **컴퓨터에서 시간을 어떻게 측정하고 관리하는지 코딩할 수 있습니다.**
- **메모리 관리기법(할당, 사용, 해제)을 익힐 수 있습니다.**

과정장점

본과정은 강사가 직접 집필한 "리눅스 시스템 프로그래밍" 교재를 가지고 강의하므로, 교육내용을 확실히 이해할 수 있도록 정확하게 전달합니다. 강사가 실무에서 경험한 다양한 예제들을 가지고 교습하므로 쉽고 재미있게 학습할 수 있습니다.

주요내용

- **시스템 호출(System Call)의 개념 설명**
- **장치 입출력(open, read, write, ioctl, seek) 실습**
- **진보된 입출력(select, poll, scatter, gather) 실습**
- **프로세스(pork, 쓰레드, 데몬, 신호) 실습**
- **시간(get, set, sleep, wait, timer) 실습**
- **메모리(할당, 해제, 매핑, 스택) 실습**

사전지식

C 언어 프로그래밍에 대해서 중급 이상의 실력을 보유하고 있어야 함.

강의내용

순서	주요내용	상세설명
1	**시스템 호출 이해**	open() / read() / write() 실습
2	**장치 입출력(I/O) 실습**	sync() / lseek() / 버퍼입출력 / fopen() / fread() / fwrite() 실습
3	**진보된 입출력 실습**	select() / poll() / scatter() / gather() / mmap() 실습
4	**프로세스 이해 및 실습**	fork() / exec(), exit(), wait() / 쓰레드 / 데몬 / 신호 실습
5	**시간(time) 실습**	time 구조체 / clock, time() / settime(), localtime() / sleep(), usleep() / nanosleep() / alarm(), timer 실습
6	**메모리 실습**	malloc(), calloc(), free() / mmap(), munmap() / alloca() 스택할당 실습

강의교재

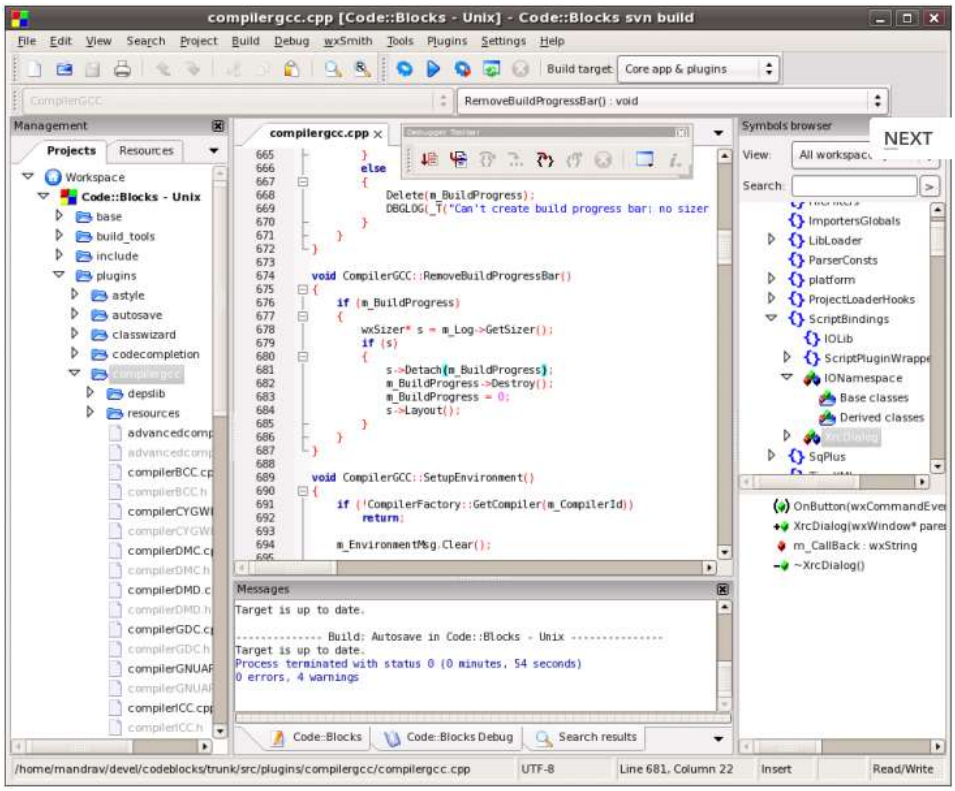
도서명: 리눅스 시스템 프로그래밍
출판사: 커널연구회
저자: 정재준
도서정보 링크(시중서점): http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=16931463

저자가 수년 동안 리눅스 시스템 프로그래밍을 하면서 경험한 내용을 토대로 기술하였다. 리눅스 시스템 프로그래밍을 개념 위주로 설명하는데에 그치지 않고 직접 실습해 볼 수 있는 소스코드를 많이 실었다. 제 1 부에서는 임베디드보드와 부트로더에 대해서 저자가 실습한 내용을 그대로 기술하고 리눅스 커널 포팅에 대해서 설명했다. 제 2 부에서 리눅스 시스템 프로그래밍을 위한 시스템 호출 함수들을 예제 소스코드와 함께 자세히 설명하고 있다. 입출력(I/O), 프로세스, 데몬, 신호, 쓰레드를 구현하는 함수들을 설명하고 이것을 저자가 직접 실습한 소스코드를 함께 실어서 설명했다. 아울러 시간 및 메모리에 대해서도 빠짐없이 기술했다. 특히 마지막 3 부에서 저자가 직접 임베디드 보드에 구현한 영어학습기 예제 소스를 실어서 리눅스 시스템 프로그래밍에 대해서 실습 위주로 익힐 수 있도록 내용을 알차게 구성했다.

실습장비

ARM Cortex-A7(core 4 개) 기반의 라즈베리파이 2 보드에서 gcc 버전 4.9.2 로 실습합니다.

통합개발 환경(IDE)인 CodeBlocks 을 사용하여 소스 편집, 컴파일, 실행, 디버깅 합니다.



B1.3 ARM 아키텍쳐, STM32 프로그래밍

과정개요

시중에서 상품화 되고 있는 임베디드 보드는 대부분 ARM 아키텍쳐로 되어 있습니다. ARM 계열의 임베디드 보드에서 프로그래밍을 하려면 ARM 구조에 대해서 잘 알고 있어야 합니다. 본 교육과정은 ARM 아키텍쳐, 메모리맵, 입출력 장치에 접근하는 주소에 대해서 이해한후, 여러가지 입출력(GPIO, UART, 인터럽트, I2C, SPI, Timer, PWM, ADC, DMA) 장치들을 동작시키는 소스 코드를 설명합니다. 실습을 통해서 이해하게 되므로 현장 업무에 바로 적용할 수 있습니다.

과정목표

- **ARM 아키텍쳐 및 메모리맵 어드레싱에 대해서 이해 합니다.**
- **통합개발환경(IAR)을 설치하고 작업환경을 설치합니다.**
- **GPIO 와 UART 에 대해서 실습합니다.**
- **인터럽트 동작을 이해 합니다.**
- **I2C / SPI 인터페이스를 실습 합니다.**
- **Timer, PWM 동작방식을 이해하고 실습 합니다.**
- **ADC 에 거리측정 센서를 연결하여 DMA 로 읽는 방식을 이해합니다.**

과정장점

본과정은 강사가 현장 업무에서 경험한 내용들을 잘 정리하여 소스 코드 레벨에서 이해하기 쉽도록 교육합니다. 프로그램 소스를 IAR 환경에서 구조적으로 작성하여 수강생들이 소스 코드를 따라가면서 이해할 수 있도록 구성했으므로 현장 업무에도 바로 적용할 수 있습니다.

주요내용

- **ARM 아키텍쳐 이해**
- **메모리맵 이해**
- **개발환경(IAR) 이해**
- **GPIO / UART 실습**
- **인터럽트 / I2C / SPI 실습**
- **Timer, PWM 실습**
- **ADC / DMA 실습**

사전지식

C 언어 프로그래밍에 대해서 중급 이상의 실력을 보유하고 있어야 함.
리눅스 시스템 프로그래밍에 대해서 기본적인 지식이 있어야 함.

강의내용

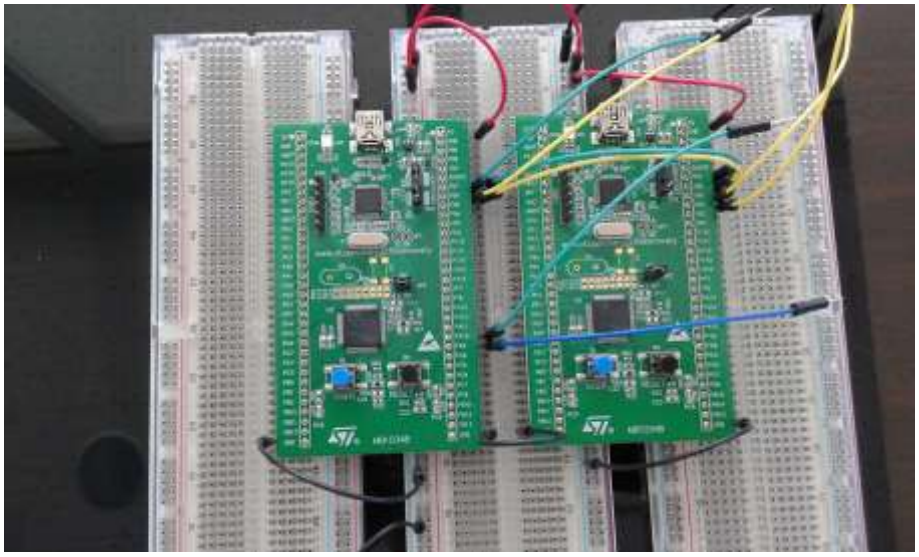
순서	주요내용	상세설명
1	**ARM 아키텍쳐 이해**	ARM Cortex-M 특징 및 구조 이해 ARM Cortex-A 특징 및 구조 이해
2	**메모리맵 이해**	어드레싱 / 메모리맵 / 입출력 장치(I/O) 접근주소
3	**개발환경(IAR) 이해**	IAR 통합개발환경 설치 및 이해
4	**GPIO 실습**	GPIO 초기화 / 핀설정 / 동작모드 / 입출력 실습
5	**UART 실습**	UART 초기화 / 핀설정 / 동작모드 / 입출력 실습
6	**인터럽트 실습**	인터럽트 초기화 / 핀설정 / 동작모드 / 인터럽트 실습
7	**I2C 실습**	I2C 초기화 / 핀설정 / 동작모드 / I2C 입출력 실습
8	**SPI 실습**	SPI 초기화 / 핀설정 / 동작모드 / SPI 입출력 실습
9	**Timer, PWM 실습**	Timer, PWM 초기화 / 핀설정 / 동작모드 / PWM 실습
10	**ADC 실습**	ADC 초기화 / 핀설정 / 동작모드 / ADC 실습
11	DMA 실습	DMA 초기화 / 핀설정 / 동작모드 / DMA 실습

강의교재

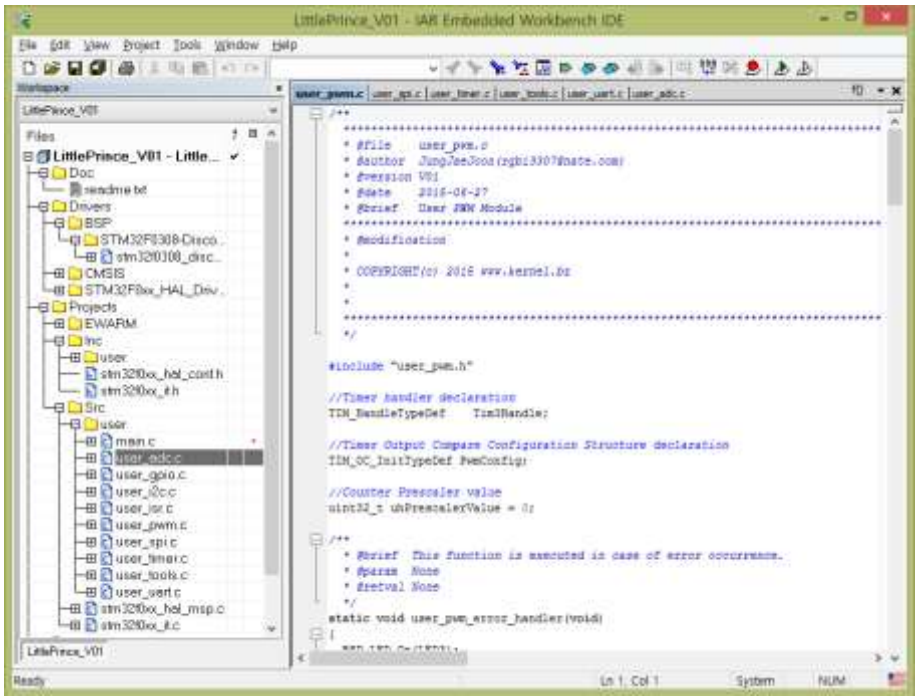
ARM 아키텍쳐 Manual 과 Datasheets 을 교재로하여 프로그램 소스를 직접 실습하면서 강의를 합니다.

실습장비

ARM Cortex-M0 Discovery 보드 2 개를 브래드보드에 연결하여 실습합니다.



통합개발 환경(IDE)인 IAR 을 사용하여 소스 편집, 컴파일, 실행, 디버깅 합니다



B1.4 리눅스 커널 자료구조 알고리즘 실습

과정개요

리눅스 커널은 하드웨어 자원을 빠르고 효율적으로 처리하기 위해 가장 최적화된 자료구조와 알고리즘을 적용하고 있습니다. 이로인해 커널의 자료구조 알고리즘에 대한 깊이있는 지식이 없다면 커널을 제대로 이해할 수 없습니다. 본 과정은 리눅스 커널의 분석과 이해가 필요한 개발자를 위해 커널에 적용된 자료구조와 알고리즘을 분석하고 이해하기 위한 과정입니다.

과정목표

- **리눅스 커널 자료구조 알고리즘 설계에 대해 이해할 수 있습니다.**
- **커널 객체지향 구조체에 대해 이해할 수 있습니다.**
- **커널 링크드 리스트(Doubly Circular Linked List)에 대해 알수 있습니다.**
- **커널 큐(Queues) 자료구조에 대해 알 수 있습니다.**
- **Red-Black Tree 자료구조에 대해 알 수 있습니다.**

과정장점

본 과정은 단순 이론 설명이 아니라 커널 소스를 직접 분석하고 이를 응용하는 실습을 통해 커널에 적용된 자료구조와 알고리즘을 제대로 알수 있도록 실습 위주로 강의합니다.
(커널 소스의 기반이 되는 링크드 리스트와 큐, Red-Black Tree 에 대한 중점 강의).

주요내용

- **리눅스 커널 객체지향 코딩 이해**
- **커널에서 다루는 구조체, 함수 포인터**
- **offsetof, container_of 매크로 이해**
- **Linked-List 이해 및 실습**
- **큐(Queue) 이해 및 실습**
- **Red-Black Tree 이해 및 실습**

사전지식

C 언어 중고급 문법을 알고 있고 특히 구조체와 포인터에 대해서 잘 이해하고 있어야 함.

강의내용

순서	주요내용	상세설명
1	**커널 객체지향 코딩**	커널 구조체 / 구조체 멤버변수 접근방식 이해 구조체 내의 함수 접근방식 이해
2	**객체지향 코드 실습**	구조체 offsetof 매크로 이해 / 구조체 container_of 매크로 이해 / 커널 구조체 실습
3	**Include/linux/list.h**	LIST_HEAD 매크로 이해 / list_init
4	**링크드 리스트 실습**	list_add / list_del / list_replace / list_move / list_last list_cut / list_splice
5	**링크드 리스트 탐색**	list_entry / list_first_entry / list_for_each / list_prepare_entry / hlist
6	**큐(Queues) 실습**	kfifo 구조체 이해 / kfifo_init / kfifo_alloc / kfifo_free __kfifo_put / __kfifo_get
7	**Include/linux/rbtree.h**	Red-Black Tree 구조체 이해 / rb_parent / rb_color rb_is_red / rb_is_black
8	**lib/rbtree.c**	*rb_next / *rb_prev / *rb_first / *rb_last
9	**lib/rbtree.c 탐색**	rb_entry / rb_first / rb_insert / rb_search rb_output / rb_erase
10	**커널 내의 자료구조 알고리즘 적용사례**	task_struct / work_queue / run_queue 이해

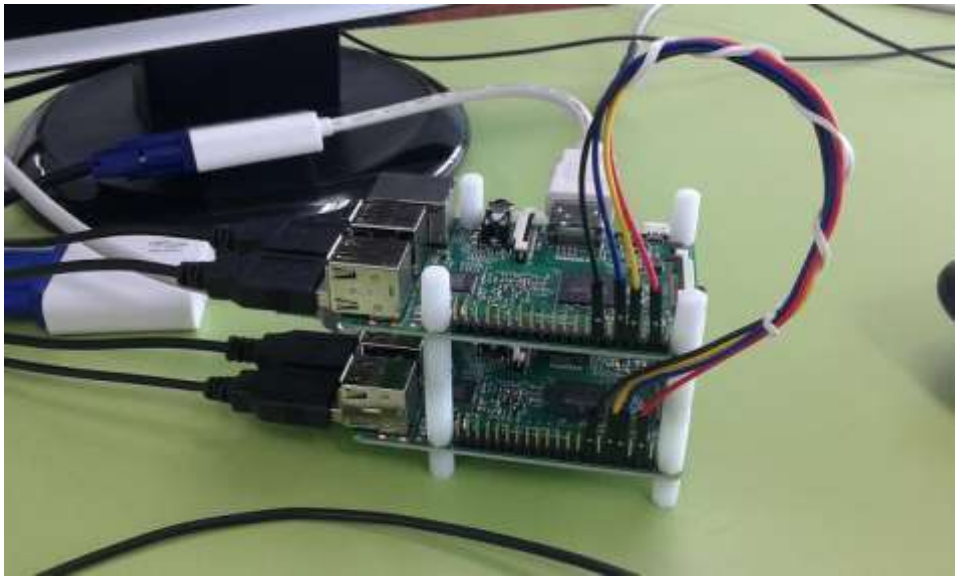
강의교재

도서명: 리눅스 커널과 디바이스드라이버 실습 2
출판사: 커널연구회
저자: 정재준
도서정보 링크(시중서점): http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=27737654

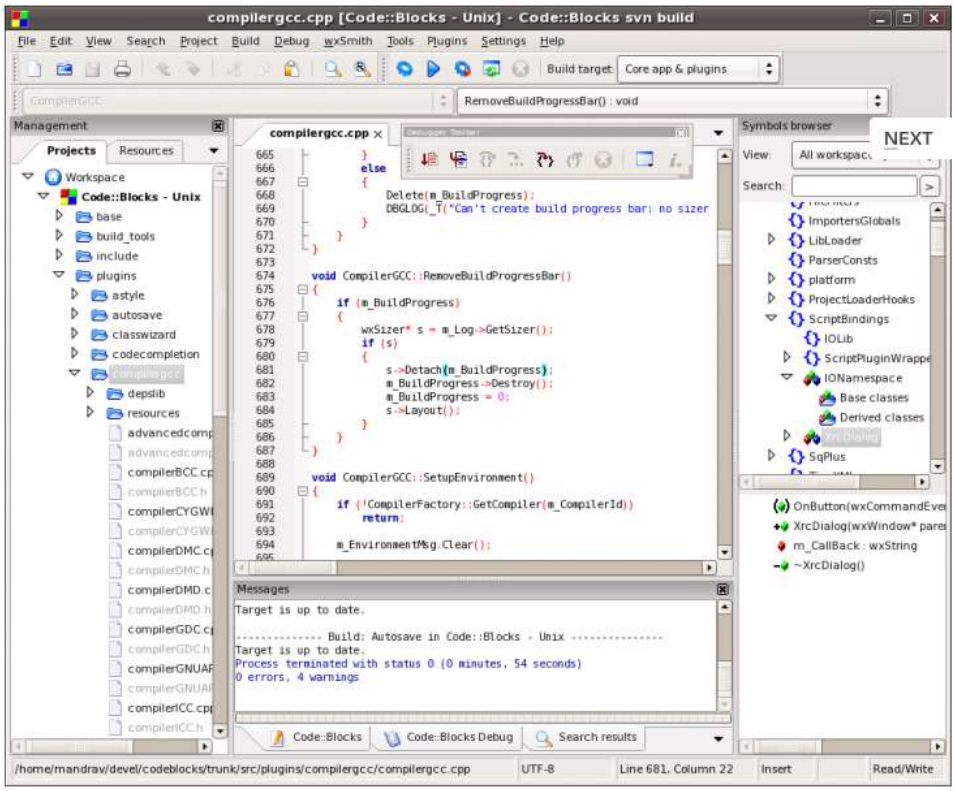
저자가 커널연구회(www.kernel.bz)를 수년동안 운영하면서 리눅스 커널 소스를 분석한 내용 위주로 구성했다. 리눅스 커널 소스를 이론적으로 설명하지 않고 최신 소스를 분석하여 소스 위주로 설명하고자 하는 저자의 꼼꼼한 노력이 돋보이는 책이다. 특히 Second Edtion 에서는 다음과 같은 내용을 대폭 보강했다.
제 4 장 동기화에 대해서 내용을 수정함. 제 6 장 메모리 관리 및 주소 변환에 대해서 내용을 좀더 자세히 보강함. 제 8 장 커널내부 들여다보기에서 커널 부팅 과정을 소스 분석적 관점에서 자세히 설명하고 커널모드와 사용자 모드에 대해서 설명을 좀더 추가함. 제 10 장 기본원리 이해에서 디바이스 드라이버 작성에 대한 기본을 좀더 충실히 기술함. 제 11 장 문자 드라이버들에서 병렬포트 실습 소스 코드를 좀더 자세히 추가하고, 임베디드에서 자주 사용하는 플랫폼 디바이스 드라이버에 대한 설명을 추가함. 제 12 장 직렬통신 디바이스 드라이버에 대한 설명을 좀더 보강함. 제 13 장 IIC 버스 인터페이스에 대한 내용을 새롭게 추가함. 제 14 장 PWM 과 ADC 드라이버에 대한 내용을 새롭게 추가함. 부록 1 리눅스 커널 버전별 특징을 새롭게 추가함.

실습장비

ARM Cortex-A7(core 4 개) 기반의 라즈베리파이 2 보드에서 gcc 버전 4.9.2 로 실습합니다.



통합개발 환경(IDE)인 CodeBlocks 을 사용하여 소스 편집, 컴파일, 실행, 디버깅 합니다.



B1.5 리눅스 커널 소스 디버깅 실습

과정개요

리눅스 커널 소스를 이론적으로만 강의하는 방식이 아니라 커널 소스를 line by line, step by setp으로 디버깅하는 방법을 익혀서 커널 부팅과정, 디바이스드라이버 초기화 및 동작과정, 실무사례에 적용하는 방법등을 직접 실습하면서 익힐 수 있습니다.

과정목표

- **커널 소스 디버깅(menuconfig, kgdb, kgdboc) 방법을 습득할 수 있습니다.**
- **디바이스 드라이버 초기화 및 진입지점 소스를 디거깅하면서 이해할 수 있습니다.**
- **디바이스 드라이버 설계사상 및 동작 흐름을 디버깅할 수 있습니다.**
- **실무사례들을 통하여 커널 소스 디버깅 방법을 좀더 자세히 익힐 수 있습니다.**

과정장점

리눅스 커널 소스를 직접 디버깅하면서 소스 레벨에서 커널의 동작 과정을 확인하면서 커널 소스를 이해할 수 있습니다. 실무사례들(시스템콜, kobject, I2C 인터페이스)을 가지고 커널 소스를 효율적으로 디버깅하는 방법들을 숙지할 수 있고 현장 업무에 바로 적용할 수 있습니다.

주요내용

- **리눅스 커널 소스 디버깅 환경 구축**
- **커널 menuconfig, kgdb, kgdboc 설치 및 사용법**
- **커널 부팅과정 소스 디버깅**
- **디바이스드라이버 초기화 및 진입지점 디버깅**
- **디바이스드라이버 설계사상 및 동작흐름 디버깅**
- **실무사례들로 커널 소스 디버깅**

사전지식

C 언어 중고급 문법을 알고 있고 특히 구조체와 포인터에 대해서 잘 이해하고 있어야 함.
리눅스 커널 자료구조 알고리즘에 대해서 사전 지식이 있어야 함.

강의내용(리눅스 커널 4.1)

순서	주요내용	상세설명
1	**리눅스 커널 소스 디버깅 환경구축**	시리얼 콘솔 연결 / kgdb, gdb 설치 / 커널 menuconfig 설정 실습
2	**kgdboc 사용 실습**	list 명령 실습 / continue, break / info, clear disable 실습 / next, setep, watch / print, display 실습
3	**리눅스 커널 효율적인 디거깅 방법 실습**	Kconfig, Makefile 이해 및 활용기법 debug message 와 로그 활용 및 실습
4	**커널 부팅 디버깅**	init/main.c / strat_kernel() 소스 디버깅 rest_init() / 디바이스드라이버 초기화 과정 디버깅
5	**디바이스드라이버 설계사상 이해 및 디버깅**	drivers/base 이해 / device_driver 구조체 이해 module_init / probe 실습
6	**실무사례 1 시스템콜 디버깅**	open, read, write / Ioctrl copy_to_user() / copy_from_user() 실습
7	**실무사례 2 kobject 디버깅**	kobject 구조체 및 설계사상 이해 kobject_create / attribute 구조체 /sys/kernel 실습
8	**실무사례 3 I2C 인터페이스 디버깅**	i2c-core / i2c-dev / i2c-gpio / i2c-bcm2708 실습

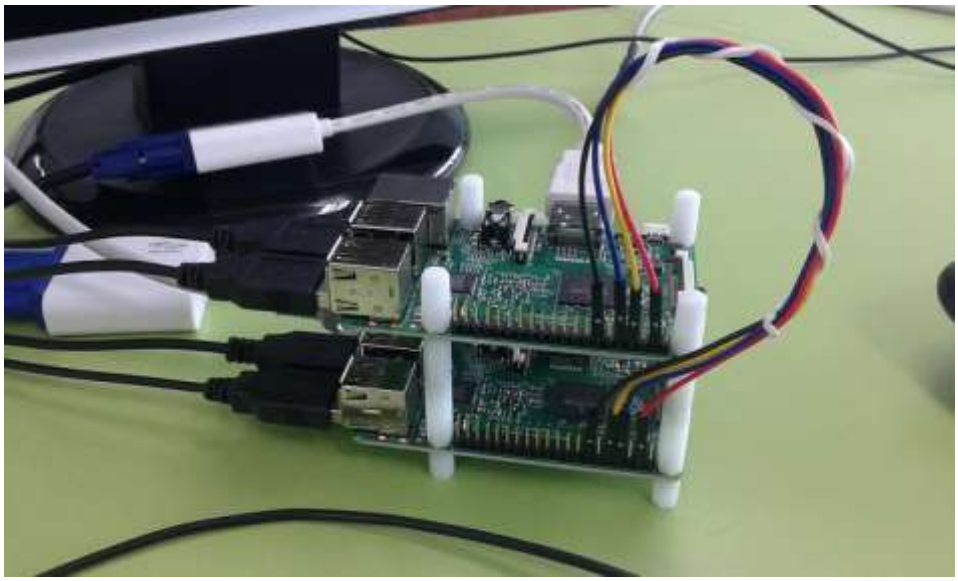
강의교재

도서명: 리눅스 커널과 디바이스드라이버 실습 2
출판사: 커널연구회
저자: 정재준
도서정보 링크(시중서점): http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=27737654

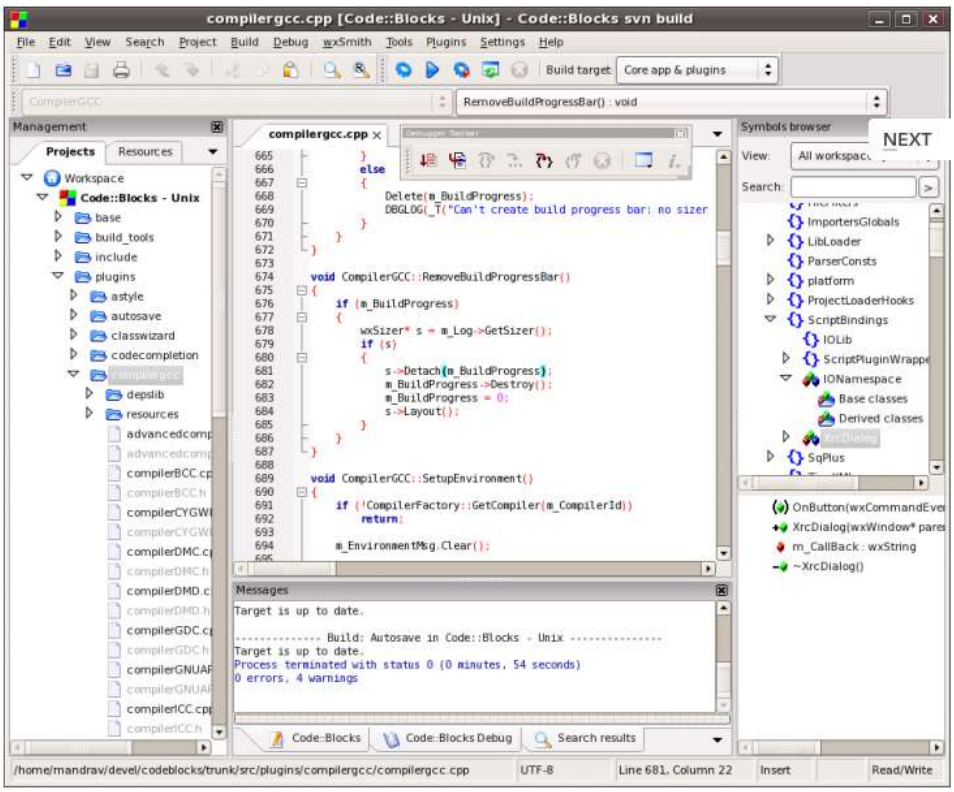
저자가 커널연구회(www.kernel.bz)를 수년동안 운영하면서 리눅스 커널 소스를 분석한 내용 위주로 구성했다. 리눅스 커널 소스를 이론적으로 설명하지 않고 최신 소스를 분석하여 소스 위주로 설명하고자 하는 저자의 꼼꼼한 노력이 돋보이는 책이다. 특히 Second Edtion 에서는 다음과 같은 내용을 대폭 보강했다.
제 4 장 동기화에 대해서 내용을 수정함. 제 6 장 메모리 관리 및 주소 변환에 대해서 내용을 좀더 자세히 보강함. 제 8 장 커널내부 들여다보기에서 커널 부팅 과정을 소스 분석적 관점에서 자세히 설명하고 커널모드와 사용자 모드에 대해서 설명을 좀더 추가함. 제 10 장 기본원리 이해에서 디바이스 드라이버 작성에 대한 기본을 좀더 충실히 기술함. 제 11 장 문자 드라이버들에서 병렬포트 실습 소스 코드를 좀더 자세히 추가하고, 임베디드에서 자주 사용하는 플랫폼 디바이스 드라이버에 대한 설명을 추가함. 제 12 장 직렬통신 디바이스 드라이버에 대한 설명을 좀더 보강함. 제 13 장 IIC 버스 인터페이스에 대한 내용을 새롭게 추가함. 제 14 장 PWM 과 ADC 드라이버에 대한 내용을 새롭게 추가함. 부록 1 리눅스 커널 버전별 특징을 새롭게 추가함.

실습장비

ARM Cortex-A7(core 4 개) 기반의 라즈베리파이 2 보드에서 ARM Native gcc 버전 4.9.2 로 실습합니다. (arm-linux-gnueabihf-gcc). 리눅스 커널 소스 버전은 4.1 입니다.



통합개발 환경(IDE)은 CodeBlocks 을 사용합니다.



B1.6 리눅스 커널 Device Tree 실습

과정개요

Device Tree는 리눅스 커널에 탑재된 장치들에 대한 접근주소(register 기준 주소)를 트리구조로 표준화한 파일입니다. 커널은 장치들을 동작시킬때 해당 장치에 접근하는 주소를 알아야 합니다. 리눅스 커널 3.2 이하의 버전에서는 아키텍쳐별로 장치에 접근하는 주소를 하드 코딩하다보니 커널소스 관리가 복잡해지고 무거워 졌습니다. 그래서 3.2 이후의 커널 버전부터 Device Tree로 표준화 시키면서 4.x 버전부터는 이 규정을 엄격하게 지켜서 디바이스드라이버를 작성해야 합니다.

과정목표

- **Device Tree 소스파일을 작성하는 문법을 이해할 수 있습니다.**
- **각종 하드웨어 장치들을 Device Tree 에 연결하는 방법을 익힐 수 있습니다.**
- **리눅스 커널에서 Device Tree 을 운영하는 소스를 활용할 수 있습니다.**
- **Device Tree 가 적용된 실무사례를 가지고 소스를 분석하고 현장업무에 적용할 수 있습니다.**

과정장점

Device Tree 작성방법을 익히고 리눅스 커널에서 이것을 어떻게 운영하고 있는지 이해할 수 있습니다. 디바이스드라이버를 작성할때 Device Tree 를 작성하는 규정을 이해하고 따라서 하면 리눅스 커널 소스를 효율적으로 코딩할 수 있습니다.

주요내용

- **Device Tree 소스파일 설명**
- **Device Tree 속성값 및 노드 명칭 이해**
- **레지스터 주소 매핑 이해**
- **Device Tree 와 하드웨어 장치들간의 연결관계**
- **이해와 소스코드 실습**
- **Device Tree 검색 및 탐색법 실습**

사전지식

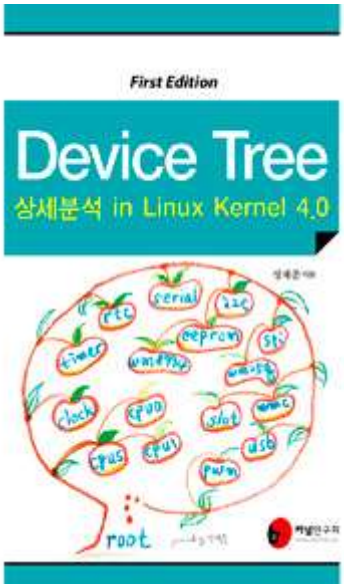
C 언어 중고급 문법을 알고 있고 특히 구조체와 포인터에 대해서 잘 이해하고 있어야 함.
리눅스 커널 자료구조 알고리즘에 대해서 사전 지식이 있어야 함.

강의내용(리눅스 커널 4.1)

순서	주요내용	상세설명
1	**Device Tree 개념 이해**	Device Tree 소스 위치설명 Device Tree 소스 분석법 Device Tree 문법이해
2	**Device Tree 문법 설명**	노드 명칭들 / compatible 속성 이해 주소 및 레지스터 표현법 cell 속성 설명 reg 속성 설명 ranges 속성 설명
3	**하드웨어 장치들을 Device Tree 로 표현하는 방법**	CPU, 메모리, 인터럽트, Clock, MMC, I2C, SPI, UART, PWM, ADC 등의 장치들을 Device Tree 에 표현하는 방법
4	**Device Tree 소스 설명 (/drivers/of)**	address.c / base.c / device.c dynamic.c / irq.c / of_*.c
5	**Device Tree 예제 1**	Device Tree 에 있는 장치들을 가져오는 소스 설명 probe() / of_alias_get_id() 실습
6	**Device Tree 예제 2**	Device Tree 탐색 실습 dtree_search_init() dtree_loop(() dtree_property() dtree_find_all_nodes() 실습

강의교재

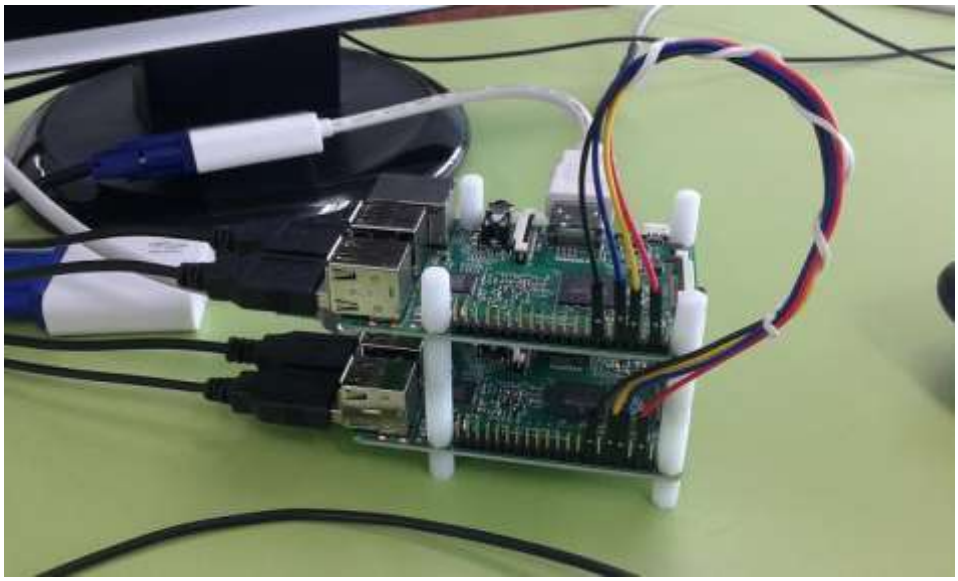
도서명: Device Tree 상세분석 in Linux Kernel 4.0
출판사: 커널연구회
저자: 정재준
도서정보 링크(시중서점): http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=65981744



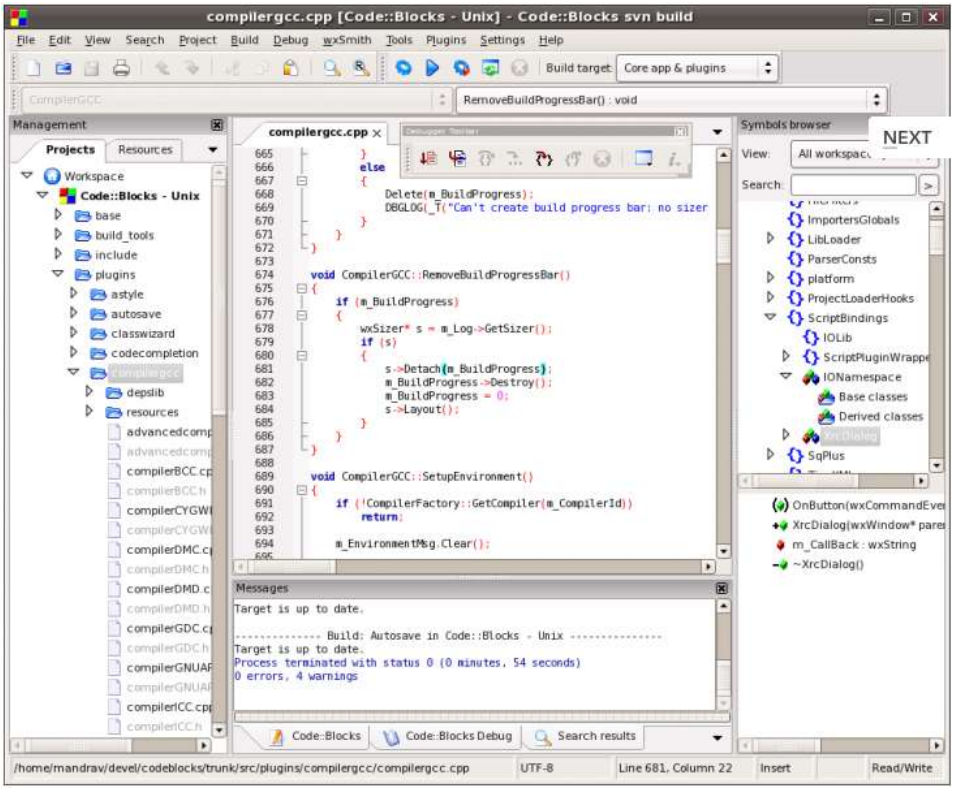
커널연구회는 리눅스 커널과 자료구조 알고리즘을 연구하고 리눅스 시스템 프로그래밍 및 디바이스드라이버 개발을 통하여 창의적인 프로젝트를 수행하여 IoT 관련 제품들을 만들어 일상생활을 풍요롭고 편리하게 하는데 가치를 두고 있습니다. 아울러 관련 기술들을 교육하여 여러사람들과 공유할 수 있도록 노력하고 있습니다. "Device Tree 상세분석 in Linux Kernel 4.0"은 커널연구회가 그동안 수년동안 최신의 리눅스 커널 소스를 분석하고 정리해온 노력들이 그대로 책속에 담겨져 있습니다. 또한 삼성 Exynos5420 이 탑재된 Arndale Octa Board 에 리눅스 커널 4.0 소스와 Device Tree 를 포팅하는 과정을 그대로 따라하면서 실습할 수 있도록 내용을 구성하여 독자분들이 현장 업무에 그대로 적용할 수 있도록 했습니다.

실습장비

ARM Cortex-A7(core 4 개) 기반의 라즈베리파이 2 보드에서 ARM Native gcc 버전 4.9.2 로 실습합니다. (arm-linux-gnueabihf-gcc). 리눅스 커널 소스 버전은 4.1 입니다.



통합개발 환경(IDE)은 CodeBlocks 을 사용합니다.



커널연구회 교육학원 위치(약도)

- **주소:** 서울시 금천구 두산로 70 (독산동 291-1)
  현대지식산업센터 A동 26층 2611호

- **교통: 지하철 1호선 독산역 1번출구**
  (도보 11분, 이동거리 708미터)

♣ 기타 내용들은 커널연구회(www.kernel.bz) 웹사이트를 참조하시기 바랍니다.
♣ 감사합니다.