

사물인터넷 이제부터
라즈베리
파이  개정판

사물인터넷을 품은 라즈베리 파이(개정판)

© 2020. 김성우 All Rights Reserved.

1쇄 발행 2020년 8월 12일

지은이 김성우

펴낸이 장성두

펴낸곳 주식회사 제이펍

출판신고 2009년 11월 10일 제406-2009-000087호

주소 경기도 파주시 화동길 159 3층 3-B호 / 전화 070-8201-9010 / 팩스 02-6280-0405

홈페이지 www.jpub.kr / 원고투고 submit@jpub.kr / 독자문의 help@jpub.kr / 교재문의 textbook@jpub.kr

편집팀 이종무, 이민숙, 최병찬, 이주원 / 소통·기획팀 민지환, 송찬수, 강민철 / 회계팀 김유미

진행 이주원 / 교정·교열 박대호 / 내지디자인 이민숙 / 표지디자인 미디어픽스

용지 신승지류유통 / 인쇄 해외정판사 / 제본 광우제책사

ISBN 979-11-90665-39-1(93000)

값 35,000원

- ※ 이 책은 저작권법에 따라 보호를 받는 저작물이므로 무단 전재와 무단 복제를 금지하며, 이 책 내용의 전부 또는 일부를 이용하려면 반드시 저작권자와 제이펍의 서면동의를 받아야 합니다.
- ※ 잘못된 책은 구입하신 서점에서 바꾸어 드립니다.

제이펍은 독자 여러분의 아이디어와 원고 투고를 기다리고 있습니다. 책으로 펴내고자 하는 아이디어나 원고가 있는 분께서는 책의 간단한 개요와 차례, 구성과 저(역)자 약력 등을 메일(submit@jpub.kr)로 보내 주세요.

사물인터넷 이제 꼭 다 라즈베리 파이 개정판



김성우 지음

Jpub
제이퍼블

※ 드리는 말씀

- 이 책에 기재된 내용을 기반으로 한 운용 결과에 대해 저/역자, 소프트웨어 개발자 및 제공자, 제이펍 출판사는 일체의 책임을 지지 않으므로 양해 바랍니다.
- 이 책에 등장하는 각 회사명, 제품명은 일반적으로 각 회사의 등록 상표 또는 상표입니다. 본문 중에는 ™, ©, ® 마크 등이 표시되어 있지 않습니다.
- 이 책에서 사용하고 있는 제품 버전은 독자의 학습 시점이나 환경에 따라 책의 내용과 다를 수 있습니다.
- 이 책에 사용된 모든 부품은 국내/외 관련 온라인 판매처에서 구입하실 수 있습니다.
- 책 내용과 관련된 문의사항은 지은이나 출판사로 연락 주시기 바랍니다.
 - 지은이: <https://cafe.naver.com/openiot>
 - 출판사: help@jpub.kr



차 레

머리말	xiii
이 책의 구성	xv
베타리더 후기	xvii



CHAPTER 1

라즈베리 파이 소개 1

1.1 개요	3
1.2 시스템 구성	3
1.2.1 하드웨어 구성 3 / 1.2.2 주변 장치 5	
1.2.3 소프트웨어 구성 8	
1.3 라즈비안 설치	9
1.4 라즈베리 파이 부팅 및 설정	12
1.4.1 라즈베리 파이 부팅 12	
1.5 라즈비안의 구성	14
1.6 기본 설정	19
1.6.1 장치 설정 19 / 1.6.2 한글 설정 23 / 1.6.3 네트워크 설정 25	
1.7 라즈베리 파이 관련 사이트	25



CHAPTER 2

리눅스 환경 27

2.1 리눅스 접속	29
2.1.1 원격 접속 29 / 2.1.2 그래픽 모드 접속 31	
2.1.3 셸 프롬프트 37	
2.2 리눅스 파일 시스템	37
2.3 기본 명령어	39
2.3.1 파일 목록 보기(ls) 39	
2.3.2 경로와 현재 작업 디렉터리 및 변경 40	
2.3.3 파일 처리 명령어 41	
2.4 일반 명령어	45
2.4.1 도움말 보기 45 / 2.4.2 페이지 단위 파일 출력 46	
2.4.3 사용자와 권한 46 / 2.4.4 기타 파일 명령어 48	



CHAPTER 3

파이썬 환경 81

2.5 시스템 관리 명령	53
2.5.1 관리자 모드 실행 53 / 2.5.2 디렉터리 및 파일 시스템 용량 확인 53	
2.5.3 시스템 자원 관리 55 / 2.5.4 프로세스 관리 56	
2.5.5 네트워크 설정 58 / 2.5.6 시간 스케줄 실행 60	
2.5.7 소프트웨어 패키지 관리 60	
2.6 파일 편집	61
2.6.1 nano 편집기 62 / 2.6.2 vi 편집기 63	
2.7 셸 환경	70
2.7.1 셸 환경설정 70 / 2.7.2 히스토리 기능 73	
2.7.3 메타 문자 74 / 2.7.4 셸 스크립트 실행 77	
3.1 파이썬 개요	83
3.2 파이썬 실행	83
3.3 변수와 자료형	85
3.3.1 변수 85 / 3.3.2 상수와 문자열 85 / 3.3.3 리스트 88	
3.3.4 튜플 88 / 3.3.5 집합 89 / 3.3.6 사전 89 / 3.3.7 기타 자료형 90	
3.4 구문과 프로그램 제어	91
3.5 함수형 프로그래밍	94
3.5.1 함수 개요 94 / 3.5.2 중첩 함수와 재귀 함수 95	
3.5.3 람다 함수 95 / 3.5.4 클로저 96 / 3.5.5 장식자 97	
3.5.6 생성기 98 / 3.5.7 코루틴 99 / 3.5.8 생성기 표현식 100	
3.6 클래스와 객체	100
3.6.1 클래스 개요 100 / 3.6.2 인스턴스 생성 101 / 3.6.3 상속 101	
3.7 입출력과 예외	102
3.7.1 파일 입출력 102 / 3.7.2 pickle 모듈 103	
3.7.3 예외 처리 103 / 3.7.4 with 구문 105	
3.8 동시 실행	106
3.8.1 멀티프로세스 106 / 3.8.2 멀티스레드 107	
3.8.3 비동기 입출력 109	
3.9 모듈	111
3.10 파이게임 모듈	112
3.10.1 파이게임 개요 112 / 3.10.2 간단한 예제 프로그램들 114	
3.10.3 스프라이트를 이용한 슈팅 게임 제작 119	



CHAPTER 4

웹 서버 프로그래밍 123

4.1	웹 서버 개요	125
4.2	웹 서버 설치	127
	4.2.1 Apache 웹 서버 설치 127 / 4.2.2 Lighttpd 웹 서버 설치 129	
	4.2.3 node.js 웹 서버 설치 131	
	4.2.4 Flask 마이크로 웹 응용 프레임워크 설치 132	
	4.2.5 PHP 설치 134	
4.3	파이썬 웹 프로그래밍	135
	4.3.1 HTTP 프로토콜 개요 135 / 4.3.2 HTML 개요 138	
	4.3.3 CGI 개요 139 / 4.3.4 AJAX 144 / 4.3.5 WSGI 인터페이스 146	
4.4	Flask 웹 응용 프레임워크	149
	4.4.1 기본 기능 150 / 4.4.2 요청과 응답 155	
	4.4.3 재접속과 오류 처리 기능 158 / 4.4.4 쿠키와 세션 기능 160	
	4.4.5 로깅과 에러 162 / 4.4.6 Jinja2 템플릿 엔진 163	
	4.4.7 서버 실행 180	
4.5	MariaDB 데이터베이스 설치	183
	4.5.1 MariaDB 소개 및 설치 183 / 4.5.2 MariaDB 사용 186	
	4.5.3 SQL 개요 187 / 4.5.4 MariaDB 자료형 187	
	4.5.5 데이터베이스 생성 및 선택 188 / 4.5.6 테이블 생성 190	
	4.5.7 테이블에 데이터 추가 및 검색 192	
	4.5.8 테이블 데이터 수정 및 삭제 193 / 4.5.9 테이블 구조 변경 195	
	4.5.10 사용자 등록 및 비밀번호 설정 196	
4.6	파이썬 MariaDB 프로그래밍	199
	4.6.1 파이썬 MariaDB 모듈 설치 199	
	4.6.2 MariaDB 서버 접속 및 오류 처리 199	
	4.6.3 SQL 질의 실행 및 결과 집합 처리 201	
	4.6.4 SQL 질의 결과 집합 처리 202	
4.7	SQLite 데이터베이스	204
	4.7.1 SQLite 설치 204 / 4.7.2 SQLite 사용 205	
	4.7.3 SQLAlchemy를 이용한 RDBMS 제어 209	
	4.7.4 웹 서버를 통한 데이터베이스 제어 214	
4.8	REST API 설계	216
	4.8.1 REST API 개요 216	
	4.8.2 REST API를 이용한 스케줄 관리 프로그램 218	



CHAPTER 5

GUI 프로그래밍 223

5.1	그래픽 윈도우 시스템 개요	225
	5.1.1 X 윈도우 개요 225 / 5.1.2 X 윈도우의 특징 227	
5.2	Tkinter 파이썬 GUI 프로그래밍	227
5.3	Qt/PySide2 프로그래밍	234
	5.3.1 Qt 개요 234 / 5.3.2 Qt 및 PySide2 프로그램 개발 환경 235	
	5.3.3 Qt/PySide2 프로그래밍 기초 236	
	5.3.4 Qt/PySide2 위젯 241 / 5.3.5 Qt 기본 위젯들 246	
	5.3.6 대화 상자 249 / 5.3.7 메뉴, 툴바 및 상태 바 위젯 253	
	5.3.8 모델/뷰 위젯 255	
5.4	Qt Designer를 이용한 프로그래밍	261
	5.4.1 Qt Designer 개요 261 / 5.4.2 UI 제작 263	
	5.4.3 정적 UI 프로그램 처리 265 / 5.4.4 동적 UI 프로그램 처리 267	
	5.4.5 채팅 프로그램 제작 267	
5.5	Qt Quick을 이용한 프로그래밍	274
	5.5.1 Qt Quick 개요 274 / 5.5.2 QML 문법 275	
	5.5.3 QML을 이용한 PySide2 GUI 프로그램 282	
	5.5.4 QML을 이용한 채팅 프로그램 제작 288	



CHAPTER 6

하드웨어 제어 297

6.1	GPIO 개요	299
	6.1.1 GPIO 구조 개요 299 / 6.1.2 GPIO 실습에 필요한 부품 301	
	6.1.3 라즈베리 파이 인터페이스 보드 302	
	6.1.4 GPIO 제어 소프트웨어 305	
6.2	GPIO 디지털 입출력 제어	306
	6.2.1 LED 출력 실습 306 / 6.2.2 푸시 버튼 입력 실습 312	
	6.2.3 LED/푸시 버튼 입출력 실습 316	
6.3	GPIO 직렬 전송 제어	319
	6.3.1 직렬 전송 방식 319 / 6.3.2 I2C 장치 실습 319	
	6.3.3 SPI 장치 실습 325 / 6.3.4 One-Wire 장치 연결 331	
	6.3.5 디지털 온도/습도 센서 장치 연결 334	
	6.3.6 USB 및 UART를 통한 외부 장치 연결 335	
6.4	적외선 센서 모듈	340
6.5	디스플레이 장치 실습	346
	6.5.1 LCD1602 장치 실습 346 / 6.5.2 TFT LCD 장치 실습 349	



CHAPTER 7

카메라 장치 활용 359

7.1	카메라 연결 개요	361
7.2	USB 웹캠 연결	361
7.3	파이카메라 연결	364
	7.3.1 파이카메라 개요 364 / 7.3.2 파이캠 유틸리티 365	
	7.3.3 Picamera 모듈 366 / 7.3.4 V4L2 커널 디바이스 드라이버 활용 368	
7.4	motion 프로그램을 통한 원격 감시 시스템 ..	369
7.5	mjpg.streamer 프로그램을 통한 원격 스트리밍	372
7.6	VLC 프로그램을 통한 원격 스트리밍	375
7.7	PIL 및 matplotlib 모듈을 활용한 이미지 처리	376
7.8	OpenCV를 활용한 영상 처리	378
7.9	휴대용 카메라 제작	386



CHAPTER 8

모터 연결 및 RC 카 제어 389

8.1	모터 개요	391
8.2	서보모터 제어	391
8.3	DC 모터 제어	395
8.4	RC 카 제어	397
	8.4.1 RC 카 조립 및 구동 테스트 397 / 8.4.2 웹을 통한 RC 카 제어 401	
	8.4.3 조이스틱 제어 및 AJAX 폴링 방식의 영상 스트리밍 405	



CHAPTER 9

위치 및 방향 측정 장치 활용 407

9.1	GPS 수신기 개요	409
9.2	GPS 수신기 연결	409
9.3	지도 위에 현재 위치 나타내기	413
9.4	날씨 예보 장치 만들기	415
9.5	Navit 내비게이션 프로그램 구동	422
	9.5.1 내비게이션 소프트웨어 설치 422	
	9.5.2 음성 안내 기능 설정 425 / 9.5.3 하드웨어 제작 426	
9.6	IMU 센서 활용하기	428
	9.6.1 만보계 만들기 429 / 9.6.2 전자 나침반 만들기 432	
	9.6.3 AHRS 만들기 434	



CHAPTER 10

아두이노 보드 연결 443

10.1	아두이노 소개	445
10.2	아두이노 프로그래밍	446
10.3	UART를 이용한 연결	449
10.3.1	직접 연결 449	
10.3.2	PyFirmata를 사용한 아두이노 포트 제어 450	
10.4	Xbee 모듈을 이용한 통신	452
10.5	블루투스 및 BLE를 이용한 통신	458
10.5.1	블루투스 개요 458	
10.5.2	블루투스 설정 및 bluez 패키지 설치 460	
10.5.3	FTP 프로파일 467 / 10.5.4 A2DP 오디오 스트리밍 470	
10.5.5	BLE 개요 474 / 10.5.6 BLE 비콘 475	
10.5.7	BLE 장치 간 연결 486 / 10.5.8 블루투스 LED 제어 496	



CHAPTER 11

사물인터넷 보드 활용 501

11.1	사물인터넷 보드 개요	503
11.2	ESP8266 모듈을 이용한 아두이노 프로그래밍	503
11.2.1	ESP8266 개요 503	
11.2.2	AT 펌웨어 테스트 504	
11.2.3	ESP8266 모듈을 활용한 아두이노 프로그래밍 511	
11.3	아두이노 IDE를 활용한 ESP8266 펌웨어 프로그래밍	515
11.3.1	ESP8266 아두이노 코어 설치 및 프로그래밍 515	
11.3.2	AJAX 폴링 방식의 센서 모니터링 520	
11.4	NodeMCU 보드 활용	527
11.4.1	NodeMCU 개요 527	
11.4.2	마이크로파이썬 펌웨어 설치 529	
11.4.3	마이크로파이썬 기본 사용 531	
11.4.4	Thonny 파이썬 IDE 활용 534	
11.4.5	Web REPL 536	
11.4.6	웹 서버를 통한 LED 제어 및 DHT22 온도/습도 센서 모니터링 538	
11.4.7	SSE 방식의 센서 모니터링 543	
11.5	PlatformIO/VSCoDe를 활용한 ESP8266 펌웨어 프로그래밍	548
11.5.1	PlatformIO 개요 548	
11.5.2	VSCoDe 설치하기 552	
11.5.3	웹소켓/SocketIO 방식의 센서 모니터링 556	

- 11.6 ESP32 보드 활용** 566
 - 11.6.1 ESP32 개요 566 / 11.6.2 ESP-IDF 개발 환경 568
 - 11.6.3 다른 개발 환경 571 / 11.6.4 BLE 응용 프로그램 576
- 11.7 MQTT 프로토콜** 581
 - 11.7.1 MQTT 프로토콜 개요 581
 - 11.7.2 MQTT를 이용한 LED 제어 및 DHT22 센서 모니터링 585
 - 11.7.3 MQTT와 SSE/Websocket을 통한 웹 서버 제어 587
- 11.8 LoRaWAN을 활용한 장치 제어 및 모니터링** 592
 - 11.8.1 LoRaWAN 개요 592
 - 11.8.2 게이트웨이 설치 594
 - 11.8.3 애플리케이션 및 LoRa 장치 등록 598
 - 11.8.4 장치 펌웨어 프로그래밍 600
 - 11.8.5 MQTT를 활용한 LoRa 장치 제어 및 모니터링 603



CHAPTER 12

사물인터넷 서비스 활용 611

- 12.1 사물인터넷 개요** 613
- 12.2 WebIOPi 웹 서버 연결** 613
- 12.3 OpenHAB** 621
 - 12.3.1 OpenHAB 개요 및 설치 622
 - 12.3.2 openHAB 구조 628
 - 12.3.3 GPIO 바인딩 및 REST API 테스트 629
 - 12.3.4 TCP 바인딩 및 자동화 테스트 635
 - 12.3.5 알람 및 카메라 기능 테스트 643
- 12.4 Home Assistant 홈 서버** 645
 - 12.4.1 Home Assistant 개요 및 설치 645
 - 12.4.2 기본 구조 및 사용 647
 - 12.4.3 새로운 Home Assistant와 ESPHome 655
- 12.5 클라우드 데이터 저장 서비스** 661
 - 12.5.1 ThingSpeak 661 / 12.5.2 plotly 663
- 12.6 사물인터넷 연동 서비스** 665
 - 12.6.1 IFTTT 서비스 665
 - 12.6.2 Pushbullet 서비스 연동 671
 - 12.6.3 텔레그램 연동 676



CHAPTER 13

음성 인식과 가상 비서 683

13.1	리눅스 사운드 시스템 및 음성 변환 테스트	685
13.2	음성 인식, 합성 및 가상 비서 시스템 개요	688
13.3	음성 인식	688
13.3.1	포켓 스피킹스 689	
13.3.2	구글 음성 인식 API 694	
13.3.3	SpeechRecognition 패키지 695	
13.4	음성 합성	697
13.5	가상 비서	699
13.5.1	구글 어시스턴트 700	
13.5.2	스노우보이 연동 711	
	찾아보기	717



머리말

“

아침에 일어나면 현재 나의 건강 상태가 체크되고,
나의 기호에 따라 기계가 자동으로 식사를 준비하고,
집을 나서면 회사까지 태워 줄 자율주행차가 대기하고 있고,
그날 해야 할 업무가 자동으로 스크린에 나타난다.

”

요즈음 우리는 SF 영화에서 봤던 기술들이 하나둘씩 구현되는 새로운 세상을 만나고 있다. 그 중심에는 사물인터넷 기술이 있다. 센서와 컴퓨터가 내장된 사물들이 인터넷으로 연결되어 각종 서비스를 제공하는 마법 같은 기술을 통틀어서 사물인터넷이라고 일컫는다. 1995년에 처음 사물인터넷이라는 용어가 만들어졌지만, 당시에는 사물에 내장되는 컴퓨터와 센서의 가격이 비싼 데다 한정되고 느린 인터넷 환경 등의 이유로 주목받지 못하였다. 하지만 21세기에 들어와서 반도체 칩의 가격이 급격하게 하락하고, 컴퓨터 성능과 인터넷 속도가 몰라보게 향상되면서 실생활에 사물인터넷 기술의 구현이 가능하게 되었다.

2012년에 처음 만들어진 라즈베리 파이 보드는 아두이노와 더불어 사물인터넷을 위한 소프트웨어를 프로그래밍할 수 있는 대표적인 오픈소스 교육용 컴퓨터다. 이 책은 사물인터넷을 구현하는 데 관심을 가진 대학생을 포함한 일반인들을 대상으로 라즈베리 파이 보드를 활용하여 다양한 장치를 제어할 수 있도록 프로그래밍하는 방법에 관해 다룬다. 기본적인 IT 지식만 있으면 전체적인 내용을 이해할 수 있도록 구성하였지만, 센서를 비롯한 하드웨어와 운영체제를 포함한 컴퓨터 소프트웨어에 관해 어느 정도 알고 있다면 훨씬 이해하기가 쉬울 것이다.

최근 들어 아두이노와 라즈베리 파이가 이끌고 있는 DIY(Do-It-Yourself) 또는 메이커 문화가 붐이다. 또한, 인터넷 커뮤니티와 블로그 등에는 사람들이 공개한 각종 노하우와 오픈소스가 넘쳐나므로 독자들이 사물인터넷 프로그래밍을 익히기에 좋은 시기인 것 같다. 라즈베리 파이를 사용하여 사물인터넷 장치를 만들고 구현하려는 사람들에게 이 책에서 다루는 다양한 기술과 예제가 도움이 될 것으로 기대한다.

감사의 글

지난 몇 년간 라즈베리 파이를 처음 접하고 수업을 진행하면서 이 책을 완성하도록 도움을 준 많은 분에게 감사를 표하고 싶다. 먼저, 저자를 공학자의 길로 이끌어 준 김병국 지도교수님께 감사드린다. 또한, 임베디드 소프트웨어 수업을 수강하면서 라즈베리 파이와 씨름하며 책의 많은 부분이 구현되는 데 도움을 준 사랑스러운 학생들(호용, 동언, 동석, 지형, 아람, 주연, 홍준, 근수, 형석)에게 감사드린다. 네이버 오로카와 pipc 카페에서 활동하며 여러 좋은 정보를 알려 주신 회원님들, 특히 wintersalmon 님께 감사의 말을 전하고 싶다. 그리고 집필과 출판을 독려하며 조언해 주고 편집을 도와준 제이펍 출판사의 대표님 및 편집자, 디자이너께 감사드린다.

마지막으로, 출간을 기뻐해 주신 부모님과 지난 책이 집필되는 동안 옆에서 묵묵히 지켜봐 주고 응원해 준 내 인생의 반려자 현실에게 감사한다. 그리고 희상과 지원, 늘 사랑한다.

지은이 김성우



이 책은 크게 세 부분으로 구분할 수 있다.

1부는 1장부터 5장까지이며, 라즈베리 파이와 프로그래밍을 처음 접하는 초심자도 접근할 수 있도록 리눅스 환경과 파이썬 프로그래밍 언어에 대한 기본적인 사용법을 담았다.

파이썬은 문법이 간결하고, 가독성이 뛰어나며, 마이크로소프트 윈도우와 리눅스 등의 다양한 운영체제에서 구동할 수 있어서 프로그래밍을 처음 접하는 초보자도 쉽게 익힐 수 있다. 만일 리눅스와 파이썬에 익숙한 독자라면 이 부분을 건너뛰어도 된다. 아울러 파이썬으로 웹 서버를 구축하고 데이터베이스와 연동하며 GUI 프로그래밍 방법에 대해서도 다루었다.

2부는 6장부터 9장까지이며, 라즈베리 파이에 센서를 비롯한 각종 장치를 활용하는 방법에 대해 설명하였다.

라즈베리 파이의 GPIO 입출력 포트를 활용하여 LED, LCD, 적외선 리모컨 등의 사용법을 다루었으며, 카메라, 모터, 위치 방향 측정 장치를 활용하는 내용은 따로 장을 나누어 서술하였다.

3부는 10장부터 13장까지이며, 사물인터넷에 대한 고급 기술과 서비스에 대해 소개하였다.

아두이노를 포함하여 최근에 각광받는 저렴한 사물인터넷 보드를 소개하고 라즈베리 파이와 연동하는 방법을 설명하였다. 아울러 라즈베리 파이에 홈 자동화 서버를 설치 활용하며, 클라우드 데이터 서비스 및 연동 서비스를 활용하여 프로그래밍하는 방법에 대해서도 다루었다. 마지막에는 최근 많이 사용되는 음성 인식 기술과 구글 어시스턴트와 같은 인공지능 가상 비서를 활용하여 장치를 제어하는 방법을 수록하였다.

예제 코드 및 Q&A

이 책에서 사용한 예제 코드를 비롯한 각종 자료는 필자가 개설한 네이버 카페(<http://cafe.naver.com/openiot>)나 깃허브^{github} 사이트에서 다운로드할 수 있다. 깃허브에서 다운로드하려면 라즈베리 파이에서 다음과 같은 명령을 실행하면 된다. 아울러 이 책으로 공부하면서 궁금한 점이나 개선해야 할 내용이 있으면 위에 언급한 네이버 카페를 통해 필자에게 연락하기 바란다.

```
$ sudo apt-get install git
$ git clone https://github.com/swkim01/RaspberrypiwithIOT
```



※ 초판과 개정판의 기본 형태가 유사해 초판을 베타리딩하신 분들의 후기를 개정판에도 수록했음을 밝힙니다.

강지훈(피키캐스트)

머신러닝과 AI, 그리고 데이터를 다루는 일을 하다 보니 취미나 업계 동향을 살필 겸해서 IoT 연구 등에 관심을 두게 되었습니다. 그간 다양한 라즈베리 파이 관련 서적이 출간되었지만 IoT와 접목시킨 책이 없었는데, 이 책을 만나서 아주 반가웠고, 이렇게 일반 독자들보다 먼저 읽게 되어 좋았습니다. 조만간 놀랄 만한 제품을 만들 수 있기를 기대해 봅니다.

손정호(한의사)

라즈베리 파이를 활용하기 위해서는 리눅스 환경에 관한 지식이 필요한데, 이 책에서는 리눅스 파일 구조, 셸, 파이썬, Apache, MySQL 등 기본적인 라즈베리 파이 활용에서 사물인터넷 프로그래밍까지 필요한 거의 모든 지식을 망라하고 있습니다. 설명도 자세한 편이라 따라가기 쉬웠습니다.

유형진(데브구루)

몇 년 사이에 라즈베리 파이를 필두로 다양한 오픈 H/W들이 나타나는 것 같습니다. 이 책은 그 중에서도 가장 핫한 라즈베리 파이를 활용할 수 있는 무한한 가능성을 알려 줍니다. 간단하게는 파이썬 프로그래밍 실습부터 다양한 하드웨어 제어(카메라, RC 카 등) 그리고 사물인터넷 활용 방안까지 고루 다루고 있습니다. 신용카드 크기의 라즈베리 파이로 다양한 경험을 원한다면 당장 이 책을 펼쳐 보기를 추천합니다. 다만, 책의 예제 일부가 웹에서 볼 수 있는 것도 있어서 창의적 측면에서 다소 아쉽습니다.

이철민(카카오)

하드웨어를 별로 만져 보지 못한 개발자라서 책을 받고 나서 본문 중 빵판(브레드보드)과 스위치 등이 보여 다소 당황했습니다. 책에서 소개하는 리눅스 기초부터 파이썬 그리고 응용 예제들을 공부하고 독자 나름대로 좀 더 연구한다면 재미있는 IoT 결과물을 만들어 낼 수 있을 것 같습니다. 저는 이 책을 통해 이번에 갓 태어난 아이의 감시(?) 시스템을 만들어 활용 중입니다.

이해진

책 한 권에 너무 많은 내용이 담겨 있어서 설명이 일부 부족해 보이는 것 같았습니다만, 차근차근 따라 하다 보니 별 문제점은 못 느꼈습니다. 오히려 새로 배우게 되는 것들이 많았는데, 카메라, GPS, 웹 서버 실습 부분은 정말 좋았습니다. 이 책과 함께 해당 센서나 모듈을 모두 갖춘 후에 봤더라면 더 효과적인 학습이 되었으리라 생각합니다.

최아연

지금까지는 사물인터넷 프로그래밍을 막연하게만 느끼고 있었는데, 이 책을 통해 보다 구체적으로 알게 되었습니다. 이 책은 단순히 하드웨어적인 부분뿐만 아니라, 라즈베리 파이 세팅부터 리눅스의 기초, 서버, DB 등 구현에 필요한 모든 부분을 다루고 있어서 라즈베리 파이 입문자인 저에게는 많은 도움이 되었습니다.



제이피엣은 책에 대한 애정과 기술에 대한 열정이 뜨거운 베타리더의 도움으로
출간되는 모든 IT 전문서에 사진 감성을 시향하고 있습니다.



1

라즈베리 파이 소개

1장에서는 라즈베리 파이 Raspberry Pi가 무엇이며, 라즈베리 파이를 구동하려면 어떤 것이 필요한지, 그리고 설치하는 어떻게 하는지 알아본다.

값싼 계산 능력을 제공할 때
사람들은 그것을 통해 사업할 방법을 찾는 것으로 밝혀졌다.

- **에번 업튼**(Eben Upton), 라즈베리 파이 제작자

1.1 개요

사물인터넷을 구현하려면 사물에 컴퓨터를 내장하여 센서와 통신 모듈 등을 제어해야 하는데, 이런 목적으로 가장 많이 사용하는 내장형 컴퓨터는 싱글 보드 컴퓨터^{single board computer}다. 싱글 보드 컴퓨터란 마이크로프로세서, 메모리, I/O^{Input/Output} 및 다른 특징을 가진 단일 회로 보드로 만든 컴퓨터를 말한다.* 반면에 일반적인 개인용 컴퓨터는 CPU, 메모리, 메인보드, 비디오 출력 보드 등의 입출력 인터페이스 보드들이 따로 만들어져 확장 슬롯으로 서로 연결되어 있다.

싱글 보드 컴퓨터 중 세계적으로 가장 많이 쓰이는 제품 중 하나가 라즈베리 파이^{Raspberry Pi}다. 라즈베리 파이는 2012년 2월, 영국의 라즈베리 파이 재단에서 교육을 목적으로 제작 발표한 스마트폰 크기의 싱글 보드 컴퓨터다. 이 컴퓨터는 이후 다양한 버전으로 발전하였으며, 우수한 성능과 저렴한 가격, 그리고 무엇보다도 리눅스^{Linux}를 포함한 편리한 개발 환경 덕분에 전 세계의 수많은 개발자가 사용한다.

라즈베리 파이는 여러 버전으로 제작되었는데, 2020년 현재 라즈베리 파이의 모델 A는 25달러, 모델 B/B+는 35달러, 제로 W는 10달러에 판매되고 있다. 국내에서는 정식 수입사인 아이씨뱅크^{ICBanQ}를 포함하여 여러 인터넷 쇼핑몰 사이트에서 구매가 가능하다. 라즈베리 파이의 외형은 그림 1-1과 같다.

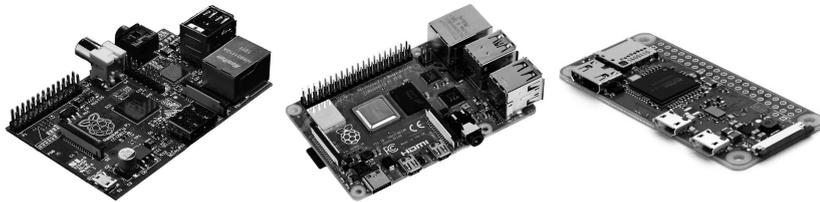


그림 1-1 라즈베리 파이 1B, 4B, 제로 W의 외형

1.2 시스템 구성

1.2.1 하드웨어 구성

라즈베리 파이 1은 미국 반도체 제작회사인 브로드컴^{Broadcom}의 BCM2835 SoC CPU를 장착하고 있는데, 이 칩은 700MHz ARM1176JZF-S 프로세서와 Videocore IV GPU 프로세서를 포함하고 있다. 또한, 모델 A는 256MB, 모델 B는 512MB의 램^{RAM}을 장착하고 있으며, SD 카드 슬롯

* 위키백과를 참고. http://en.wikipedia.org/wiki/Single-board_computer

을 통해 SD 플래시 메모리를 파일 시스템이나 저장 메모리 용도로 사용할 수 있다. 반면에 2015년 2월에 출시한 라즈베리 파이 2는 같은 회사의 BCM2836 SoC CPU를 장착하고, 900MHz 쿼드코어 ARM Cortex A7 코어 프로세서와 1GB의 램을 장착하고 있으며, 일반 SD 카드 대신 마이크로 SD 카드를 사용할 수 있다. 2018년 2월에 출시한 라즈베리 파이 3 B+ 모델은 1.4GHz 64비트 쿼드코어 ARM Cortex-A53 코어 프로세서를 포함한 BCM2837B0 CPU를 장착하였다. 2019년 6월에 출시한 라즈베리 파이 4 B 모델은 1.5GHz 64비트 쿼드코어 ARM Cortex-A72 코어 프로세서를 포함한 BCM2711 CPU를 장착하였다.

라즈베리 파이 보드의 주변 장치로는 USB, HDMI, 이더넷Ethernet, 아날로그 오디오 출력, 전원 포트, GPIO 입출력 포트, DSI, CSI 커넥터 등이 포함된다. 표 1-1은 라즈베리 파이 보드의 종류별 주요 구성 부품과 사양을 나타낸 것이다.

표 1-1 라즈베리 파이의 종류별 사양

모델	Model A		Model B			Zero
버전	1	1	2	3 B+	4	W
Chip	Broadcom BCM2835 SoC full HD Multimedia AP		Broadcom BCM2836 SoC full HD Multimedia	Broadcom BCM2837 SoC full HD Multimedia	Broadcom BCM27117 SoC full HD Multimedia	Broadcom BCM2835 SoC
CPU	700 MHz ARM1176JZF-S Applications Processor		900 MHz ARM Cortex-A7 4xCores AP	1.4GHz ARM Cortex-A53 4xCores AP	1.5GHz ARM Cortex-A72 4xCores AP	1GHz ARM1176JZF-S AP
GPU	Dual Core VideoCore IV® Multimedia Co-Processor					
Memory	256MB SDRAM	512MB SDRAM	1GB SDRAM	1GB SDRAM	1GB/2GB/4GB SDRAM	512MB SDRAM
Ethernet/WiFi	None	onboard 10/100 Ethernet RJ45 jack	onboard 10/100 Ethernet RJ45 jack	onboard gigabit Ethernet RJ45 jack, WiFi/BLE4.2	onboard gigabit Ethernet RJ45 jack, WiFi/BLE5.0	WiFi/BLE4.1
USB	Single USB 2.0 Con	Dual USB 2.0 Con	Quad USB 2.0 Connectors	Quad USB 2.0 Connectors	Dual USB 2.0 Dual USB 3.0 Connectors	Single USB 2.0 Con
Video Output	HDMI(rev 1.3&1.4) Composite RCA(PAL&NTSC)		HDMI(rev 1.3 & 1.4) Composite	HDMI(rev 1.3 & 1.4) Composite	Dual microHDMI Composite	Mini-HDMI, 1080p60
Audio Output	3.5mm jack, HDMI, I2S		3.5mm jack, HDMI, I2S	3.5mm jack, HDMI, I2S	3.5mm jack, Dual uHDMI, I2S	Mini-HDMI
Onboard Storage	SD, MMC, SDIO card slot		Micro SD card slot	Micro SD card slot	Micro SD card slot	Micro SD card slot
GPIO	26	40	40	40	40	40
OS	Linux	Linux	Linux, Android, Win10	Linux, Android, Win10	Linux	Linux, Android, Win10
Power	5V uUSB, 500mA(2.5W)	5V uUSB, 700mA(3.5W)	5V uUSB, 820mA(4.1W)	5V uUSB, 1.13A(5.66W)	5V USB-C, 3.0A(4W)	5V uUSB, 350mA(1.75W)
Dimensions	8.6x5.6x1.5cm	8.6x5.6x1.7cm	8.6x5.6x1.7cm	8.6x5.6x1.7cm	8.6x5.6x1.7cm	6.5x3.0x0.5cm

그림 1-2는 라즈베리 파이의 하드웨어 연결 장치들을 표시한 것이다.

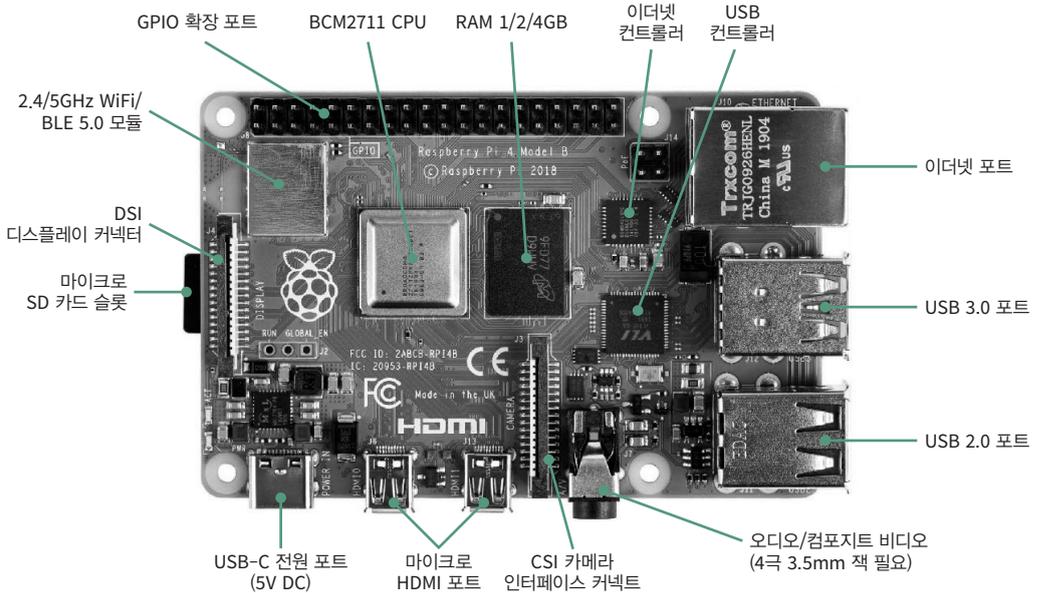


그림 1-2 라즈베리 파이의 주변 연결 장치들

1.2.2 주변 장치

라즈베리 파이를 구동하여 컴퓨터로 사용하려면 기본 보드 외에도 다양한 주변 장치(액세서리)가 필요하다.

1 마이크로 USB/USB Type-C 커넥터 및 전원 어댑터

라즈베리 파이는 전원 공급을 위해 스마트폰 등에 널리 쓰이는 마이크로 USB 또는 USB Type-C 포트를 사용한다. 라즈베리 파이 3까지는 마이크로 USB 5핀 포트를, 라즈베리 파이 4는 USB Type-C 포트를 사용한다. 만약 라즈베리 파이 4를 사용하는데 마이크로 USB 케이블이나 어댑터만 있다면, 마이크로 USB 포트를 Type-C 포트로 변환해 주는 젠더가 필요하다. 또한, 라즈베리 파이 4에서 OS 및 소프트웨어를 안정적으로 구동시키고 싶다면 5V, 3A 이상의 직류 전원을 사용하길 추천한다. 하지만, 외부 장치를 많이 연결하지 않는다면 PC의 USB 포트에 USB-C 커넥터를 연결하여 전원을 공급하거나 별도의 전원 어댑터를 연결하여 사용해도 된다. 별도의 전원 어댑터를 연결하는 경우는 안정적인 전원 공급을 위해 5V, 2A 이상의 용량을 가진 어댑터를 사용하는 것이 바람직하다.



그림 1-3 USB Type-C 케이블, 마이크로 USB 변환 젠더, 마이크로 USB DC 어댑터

2 마이크로 SD 카드

라즈베리 파이 3 또는 4는 리눅스 등의 운영 소프트웨어를 저장하기 위해 하드 디스크 대신 마이크로 SD 카드를 사용한다. 라즈베리 파이에 탑재되는 OS에 따라 다르지만, 라즈비안 OS인 경우에는 보통 8GB 이상의 마이크로 SD 카드를 준비하여 라즈베리 파이의 보드에 있는 SD 카드 슬롯(slot)에 삽입할 수 있어야 한다. 또한, 최초에 PC로부터 OS 이미지 등을 업로드하기 위해 SD 카드 어댑터 등이 필요할 수도 있다.



그림 1-4 마이크로 SD 카드와 SD 카드 어댑터

3 유무선 USB 마우스 및 키보드

라즈베리 파이 보드에는 PC처럼 마우스와 키보드를 연결하는 포트가 별도로 존재하지 않으므로 입력 장치로는 주로 USB형의 마우스와 키보드를 연결하여 사용한다. 라즈베리 파이 1 리비전 revision A, B의 경우는 USB 포트가 2개, 리비전 B+, 2, 3, 4 보드의 경우에는 USB 포트가 4개이므로 USB 카메라 등의 주변 장치를 USB 허브 없이도 추가로 연결할 수 있다. 다만, 더 많은 장치를 연결하려면 별도의 유전원 USB 허브를 사용해야 하는데, 각 USB 포트가 200~300mA 정도의 전류를 사용하므로 2A 정도의 전원 어댑터를 사용하여 충분한 전력을 공급하는 것이 바람직하다.



그림 1-5 무선 키보드 및 마우스

4 HDMI/마이크로 HDMI 케이블 및 모니터

출력 장치로는 HDMI 또는 DVI(Digital Video/Visual Interactive)를 지원하는 LCD 모니터를 사용하는 것이 일반적이다. 라즈베리 파이 3 보드까지는 HDMI 포트를 지원하고, 라즈베리 파이 4 보드는 마이크로 HDMI 포트를 지원한다. 따라서 보드에 따라 HDMI 또는 마이크로 HDMI 케이블을 연결하고, LCD 모니터에 연결하여 프로그램 실행 결과 등을 출력 화면으로 확인할 수 있다.



그림 1-6 HDMI-마이크로 HDMI 케이블, HDMI 케이블, 마이크로 HDMI 변환 어댑터

5 이더넷 케이블 또는 와이파이 USB 동글

라즈베리 파이 3 또는 4 보드를 인터넷에 연결하려면 내장된 WiFi 모듈을 통해 무선 공유기에 연결하거나 보드에 탑재된 이더넷 포트에 유선 랜(LAN) 케이블을 연결하여 공유기/랜 허브와 연결하면 된다. WiFi 모듈이 내장되지 않은 라즈베리 파이 모델은 USB 포트에 WiFi USB 동글 장치를 연결하여 무선 인터넷 공유기와 연결할 수 있다.

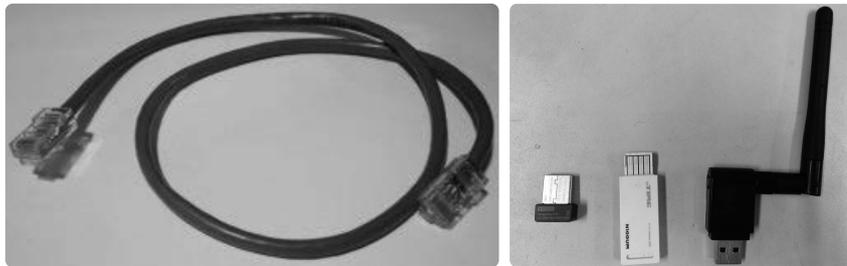


그림 1-7 이더넷 케이블 및 와이파이 동글

6 기타 외부 장치들

이 밖에도, 라즈베리 파이의 여러 연결 포트에 USB 카메라, 파이 카메라, GPIO 확장 보드, 터치 LCD, 블루투스(Bluetooth) 동글, GPS 수신기 등의 다양한 장치를 연결하여 사용할 수 있다.

1.2.3 소프트웨어 구성

라즈베리 파이는 기본적으로 리눅스 운영체제를 사용하여 구동한다(리눅스 외에도 FreeBSD나 RISC OS를 사용하는 것도 가능하다). 일반적인 개인용 컴퓨터에서도 다양한 리눅스 배포판을 실행시킬 수 있듯이 라즈베리 파이도 다양한 리눅스 배포판을 활용하여 구동시킬 수 있다. 라즈베리 파이를 위한 대표적인 배포판은 다음과 같다.

1 라즈비안*

라즈베리 파이에서 가장 많이 사용하는 리눅스 배포판이며, 라즈베리 파이 재단에서 공식적으로 권장하고 있다. 라즈비안Raspbian은 데비안Debian 리눅스를 기반으로 하며, 경량의 LXDE 데스크톱 환경과 함께 웹 브라우저와 파이썬, 스크래치 등의 다양한 도구를 포함하고 있다. raspberrypi.org의 다운로드 페이지에서 공식 배포판을 다운로드할 수 있다. 2020년 현재 라즈비안 배포판은 데비안 Buster Lite 버전과 추천 패키지들이 함께 설치된 Buster 데스크톱 버전, 이렇게 두 가지 버전을 제공한다.

2 우분투

현존하는 리눅스 배포판 중 전 세계적으로 가장 많이 사용하는 리눅스 배포판이며, 우분투Ubuntu 재단에서 라즈베리 파이 모델 B 2, 3를 위한 배포판을 공식적으로 제공하고 있다. Ubuntu Mate 18.04(데스크톱 버전), Snappy Ubuntu Core(최소 설치 버전), Ubuntu Server(서버용) 등이 있다.

3 윈도우 10 IoT Core

마이크로소프트에서 윈도우 10 버전을 라즈베리 파이 2, 3용으로 지원하는 배포판이다.

4 아치 리눅스

아치 리눅스Arch Linux는 ARM 프로세서에 특화된 리눅스 배포판이다. 아주 경량의 리눅스로, 심지어 그래픽 사용자 인터페이스Graphical User Interface, GUI 소프트웨어조차 기본적으로 포함하지 않는다.

5 RISC OS

라즈베리 파이용으로 만들어진 매우 빠르고 작은 시스템이다.

6 OSMC

라즈베리 파이를 XBMC 미디어 센터를 구동하도록 한 리눅스 배포판이다.

* 2020년 6월 이후로 라즈비안 대신 라즈베리 파이 OS라고 부른다.

7 LibreELEC

OSMC처럼 XMBC 미디어 서버용으로 만들어진 임베디드^{embedded} 리눅스 배포판이다.

1.3 라즈비안 설치

라즈베리 파이 보드에 리눅스를 설치하려면 최소 4GB, 클래스 4 이상의 마이크로 SD 카드가 필요하다. 또한, 구동시키고 난 후 다양한 패키지를 추가로 설치하거나 NOOBS를 통해 리눅스 배포판을 설치할 경우에는 넉넉잡아 8GB 이상의 마이크로 SD 카드를 준비하는 것이 좋다.

라즈베리 파이에 리눅스 배포판을 설치하는 방법에는 두 가지가 있다. 첫 번째는 라즈베리 파이 다운로드 페이지로부터 특정한 리눅스 배포판 이미지를 다운로드하여 SD 카드에 직접 설치하는 방법이며, 두 번째는 NOOBS^{New Out Of Box Software}라는 리눅스 설치 프로그램을 먼저 설치한 다음 나중에 리눅스 배포판을 선택적으로 설치하는 방법이다.

1 리눅스 배포판 직접 설치 방식

먼저, 리눅스 배포판 이미지를 직접 설치해 보자. 사용자들이 가장 많이 사용하는 라즈베리 파이 OS인 라즈비안 버스터^{Raspbian Buster} 데스크톱 버전을 대상으로 하겠다. 라즈베리 파이 공식 홈페이지의 다운로드 페이지에서 라즈베리 파이 이미지^{Imager}라는 이미지 설치 프로그램을 다운로드한 후에 호스트 컴퓨터에 설치한다.

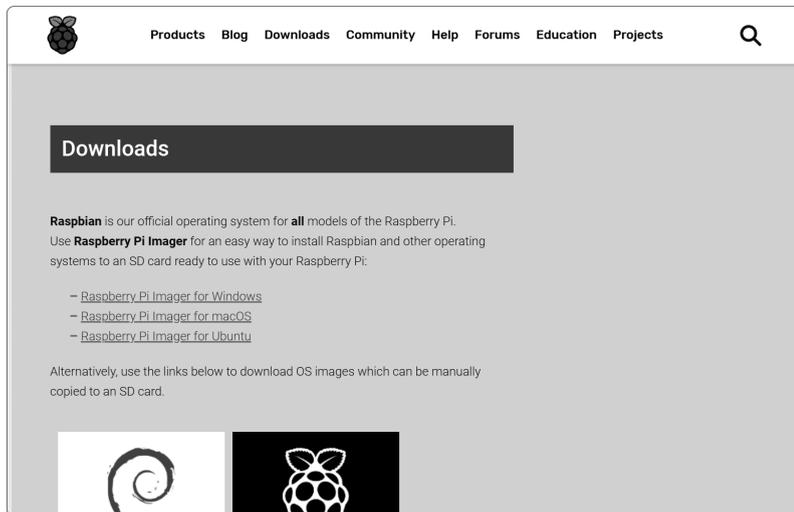


그림 1-8 라즈베리 파이 이미지 다운로드 페이지 화면

마이크로 SD 어댑터에 장착된 마이크로 SD 카드를 호스트 컴퓨터에 연결하고 이미지 프로그램을 실행한다. **운영체제** Operating System 선택 버튼을 클릭하여 **라즈비안** Raspbian을 선택하고, **SD 카드** 선택 버튼을 클릭하여 연결된 SD 카드를 선택한다.

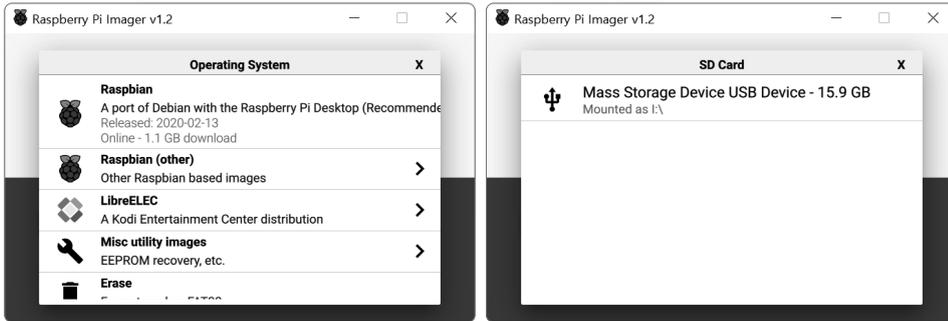


그림 1-9 이미지 프로그램의 운영체제 및 SD 카드 선택 실행 화면

이제 **WRITE** 버튼을 클릭하면 라즈비안 이미지를 SD 카드에 설치한다.

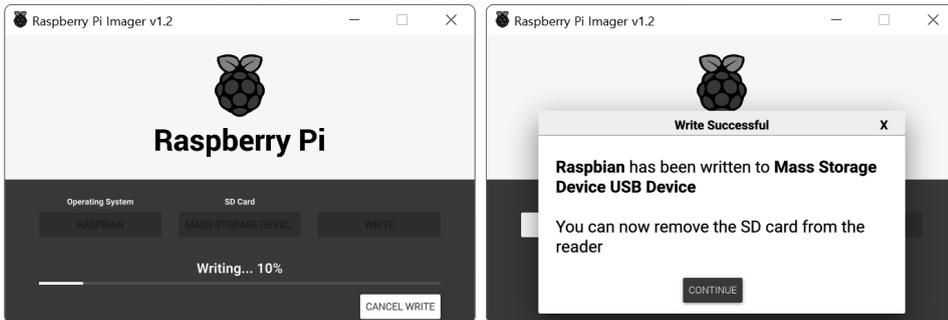


그림 1-10 이미지 프로그램의 라즈비안 설치 화면

아니면 홈페이지에서 라즈비안 이미지 파일을 직접 다운로드하고 어댑터에 장착된 마이크로 SD 카드에 직접 설치한다. 그런데 마이크로 SD 카드에 직접 설치하려면 디스크 이미지 도구가 필요하다. 디스크 이미지 툴은 MS 윈도우, 리눅스, 맥에서 모두 구동되는 balenaEtcher 프로그램을 사용하거나, MS 윈도우인 경우에는 Win32DiskImager 프로그램을* 사용하면 간단하게 처리된다.

* <https://launchpad.net/win32-image-writer>

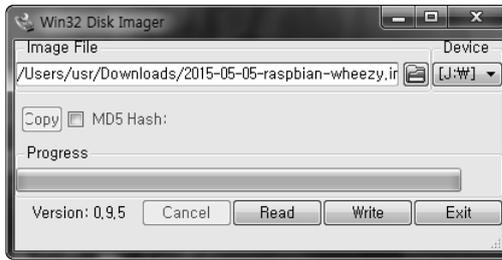


그림 1-11 Win32DiskImager 실행 화면

리눅스나 매킨토시에서는 간단한 셸^{shell} 명령어로 마이크로 SD 카드에 디스크 이미지를 복사할 수 있다. 예를 들어, 리눅스에서 SD 카드의 디바이스^{device} 파일이 /dev/sdd로 설정되어 있다면 다음과 같은 명령을 실행하면 된다.

```
sudo dd bs=1M if=<배포판 이미지 파일> of=/dev/sdd
```

2 NOOBS 설치 방식(직접 배포판을 설치한 경우에는 건너뛴)

이번엔 직접 배포판을 설치하는 대신에 NOOBS 이미지를 설치해 보자. 개인 PC가 윈도우 또는 매킨토시라면 SD 협회의 포맷 도구인 SD Formatter 4.0을 활용한다. SD Formatter 4.0 소프트웨어를 SD 협회 홈페이지*로부터 라이선스 동의하에 다운로드한 다음 설치한다. 설치가 완료되면 준비한 SD 카드를 PC에 삽입하고 SD Formatter 4.0을 실행한다. 초기 화면에서 Option 버튼을 눌러서 Format Type은 FULL(OverWrite)로, Format Size Adjustment는 ON으로 선택한 다음, OK 버튼을 누르면 SD 카드 포맷 과정이 진행된다.

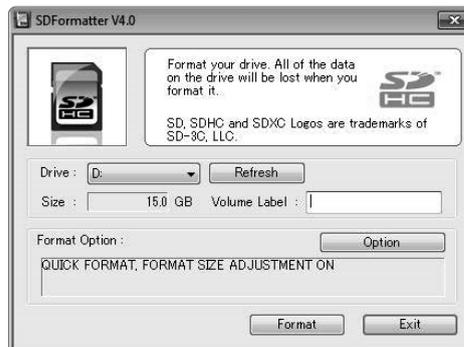


그림 1-12 SD Formatter 4.0 실행 창

* https://www.sdcard.org/downloads/formatter_4/eula_windows

한편, 리눅스를 사용하는 경우에는 gParted(또는 Parted) 프로그램을 사용하여 SD 카드를 FAT 형식으로 포맷하면 된다.

SD 카드가 포맷되었으면 라즈베리 파이 홈페이지의 다운로드 페이지*에 접속하여 NOOBS라는 리눅스 설치 패키지를 다운로드한다. NOOBS 이미지 파일은 zip 압축 파일이며, 용량은 1GB 이상이다. 이 파일의 압축을 풀고 모든 내용을 포맷된 SD 카드에 복사하면 라즈베리 파이에 리눅스를 설치할 준비가 완료된다.

1.4 라즈베리 파이 부팅 및 설정

1.4.1 라즈베리 파이 부팅

마이크로 SD 카드가 준비되면 이제 라즈베리 파이를 부팅시킬 수 있다. 부팅 과정은 다음과 같다. 먼저, NOOBS 또는 라즈비안 OS 이미지가 탑재된 마이크로 SD 카드를 라즈베리 파이 보드의 소켓에 넣는다. 다음으로 USB 키보드, 마우스(무선도 가능)를 연결하고 HDMI 출력 단자에 모니터를 연결한다. 인터넷 연결을 위해 유선랜/무선랜 USB 카드도 연결한다. 마지막으로, 전원 장치에 연결된 마이크로 USB 또는 Type-C 케이블을 보드의 전원 단자에 연결하면 된다.

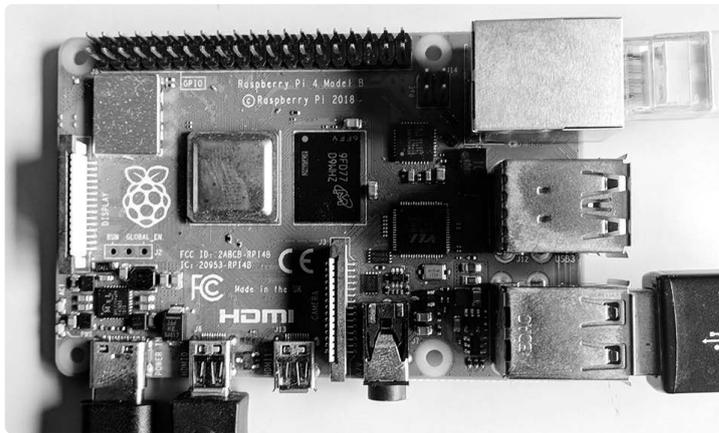


그림 1-13 라즈베리 파이에 여러 장치를 연결한 상태

전원 케이블을 연결하여 부팅하면 NOOBS 이미지에는 그림 1-14와 같은 내용이 스크린에 나타나고, 배포판 이미지에는 그림 1-15와 같이 모니터에 부팅 메시지가 곧바로 나타난다. 이 화면에서

* <https://www.raspberrypi.org/downloads>

마우스를 사용하여 설치하기를 원하는 배포판(복수 선택 가능)을 선택한 다음, Install 버튼을 누르면 해당하는 리눅스 배포판(들)의 이미지가 SD 카드로 재복사되면서 설치가 진행된다. 여기서 라즈비안을 선택한 후 설치를 진행한다. 배포판의 설치가 완료되면 자동으로 재부팅된다.

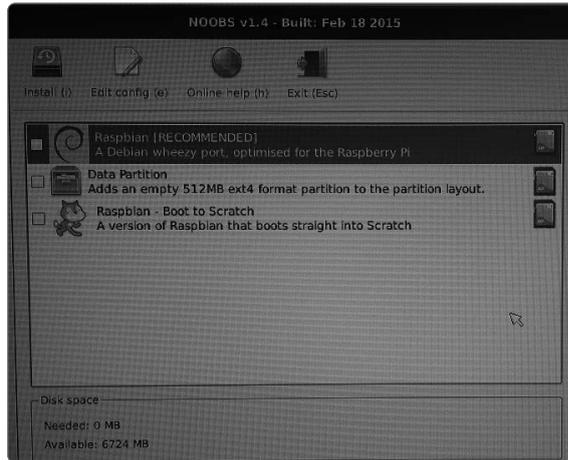


그림 1-14 NOOBS OS 선택 창

라즈비안 OS로 부팅이 시작되면 모니터에 GUI 모드 스플래시 화면이나, 다음과 같이 텍스트 모드로 부팅되는 과정에서 출력되는 로그 메시지를 확인할 수 있다. 리눅스를 부팅하는 도중에 오류가 발생하면 관련된 오류 메시지를 출력하고 작동이 멈춘다.

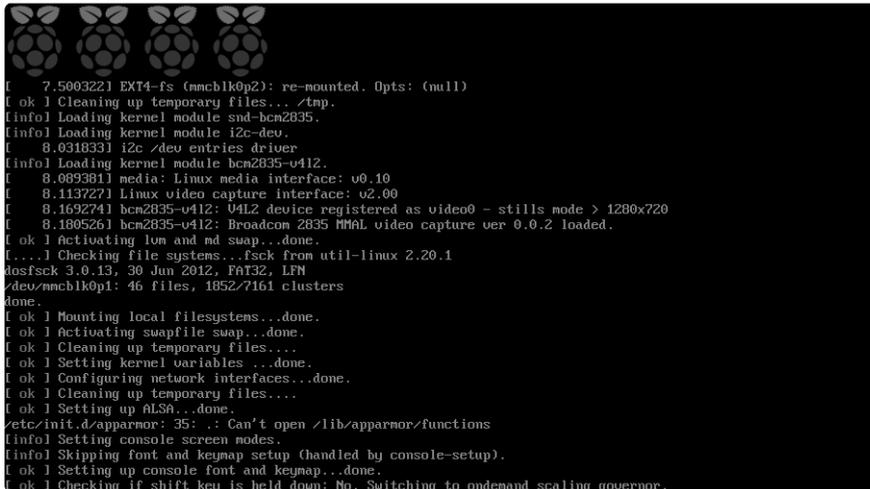


그림 1-15 라즈비안으로 부팅 중인 화면

정상적으로 부팅되면 리눅스 커널이 모든 장치를 초기화하고 루트 파일 시스템을 마운트한 후 GUI 모드에서 다음과 같은 초기 화면이 나타난다.

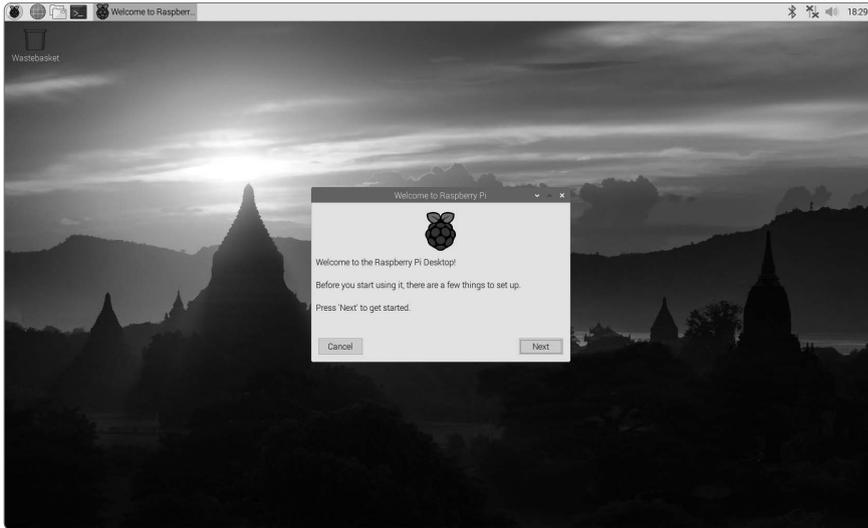


그림 1-16 라즈비안으로 부팅 후 화면

1.5 라즈비안의 구성

라즈비안에는 기본적으로 리눅스를 사용할 수 있는 환경과 패키지들이 포함되어 있다. 그중에서도 사용자가 편리하게 사용할 수 있는 GUI 데스크톱 환경으로 LXDE(Lightweight X11 Desktop Environment)를 수정한 PIXEL(Pi Improved Xwindows Environment, Lightweight)을 사용하고 있다.

PIXEL은 글자 그대로 라즈베리 파이용 경량 X 윈도우 데스크톱 환경을 말한다. 유닉스와 POSIX 표준과 호환하는 리눅스 등의 플랫폼에서는 1980년대 중반에 개발된 X 윈도우를 주요 GUI로 채택하여 사용해 왔으며, 파생된 수많은 데스크톱 환경이 개발되어 왔다. 그중에서도 PIXEL이 채택한 LXDE는 2006년에 발표되어 현재까지도 개발하며 사용하는 오픈소스(open source) 데스크톱 환경이며, 기존의 우분투, 페도라 등과 같이 고성능을 요구하는 데스크톱 리눅스 환경과는 달리 저성능 노트북이나 라즈베리 파이 등에서 원활하게 동작하도록 설계되어 있다. PIXEL은 기존 LXDE 환경에 부팅 화면이 추가되었고 바탕화면(wallpaper)으로 멋진 사진을 제공한다. 또한, 웹 브라우저도 크롬을 사용하며, RealVNC 서버와 부가 기본적으로 설치되어 있어 원격 접속 시에도 편리하다.



그림 1-17 PIXEL GU 환경

GUI 홈 화면을 보면서 LXDE 구성을 학습해 보자. 먼저, 홈 화면은 앱 실행 아이콘들이 포함된 바탕화면과 시작 메뉴 및 작업 표시줄 등이 포함된 상단 메뉴 바(menu bar)로 구성된다. PIXEL LXDE는 기본 윈도우 관리 프로그램으로 Openbox를 사용한다. 이것은 바탕화면의 실행 아이콘을 배치하거나 클릭했을 때 애플리케이션을 실행시키고, 실행된 GUI 프로그램의 창 크기 조절, 이동 등을 제어하며, 하단 메뉴 바 등을 구성하는 기능을 가진다.

상단 메뉴 바의 구성은 다음과 같다. 왼쪽에는 시작 메뉴가 있고, 중앙에는 기본 웹 브라우저, 파일 관리자, 터미널 등이 포함된 툴바(toolbar)와 작업 표시줄이 있으며, 오른쪽에는 시계 등이 있다.

- 시작 메뉴에는 라즈비안에서 기본적으로 제공하는 다양한 프로그램을 메뉴로 포함한다. 자세한 설명은 다음 페이지에 서술하였다.
- 기본 웹 브라우저 아이콘을 선택하면 크롬Chromium 웹 브라우저가 실행된다.
- 파일 관리자 아이콘을 누르면 LXDE의 기본 파일 관리자인 PCManFM이 실행된다. 이것은 MS 윈도우의 파일 탐색기와 기능이 비슷하므로 MS 윈도우에 익숙한 사용자는 쉽게 사용할 수 있을 것이다.
- 상단 메뉴 바의 중앙에 위치한 작업 표시줄은 실행하고 있는 프로그램을 버튼으로 나타내는데, 선택하면 바탕화면에 드러낼 수 있다.
- 상단 메뉴 바의 오른쪽은 네트워크 연결을 나타내는 아이콘, 소리 볼륨을 조절하는 아이콘, CPU 사용량을 모니터링하는 아이콘, 시계 등으로 구성된다.

시작 메뉴는 Programming, Internet, Games, Accessories, Help, Preferences 등으로 구분된 다양한 프로그램과 실행 및 로그아웃 기능을 메뉴 형태로 제공한다. 각 세부 메뉴의 기능은 다음과 같다.



그림 1-18 라즈비안 시작 메뉴

1 Programming

다양한 교육 및 프로그래밍과 관련된 프로그램이 제공된다.

- BlueJ Java IDE는 자바 프로그래밍을 위한 통합개발환경IDE 소프트웨어다.
- GreenFoot Java IDE는 객체지향 개념과 자바 및 2차원 그래픽 응용 프로그램을 만드는 소프트웨어다.
- Mathematica는 수학/과학/공학 분야의 프로그래밍을 할 수 있는 소프트웨어다.
- Node-RED는 주변 장치들과 네트워크를 연결하는 등의 다양한 사물인터넷을 위한 Node.js 프로그래밍을 할 수 있는 소프트웨어다.
- Python이라는 이름으로 IDLEIntegrated DeveLopment Environment이라는 파이썬 프로그래밍을 위한 통합개발환경을 포함한다. IDLE은 Python 2.x 버전에 대한 프로그래밍 도구이고, IDLE 3은 Python 3.x용이다.
- 스크래치Scratch는 미국 MITMassachusetts Institute of Technology 대학에서 만든 교육용 프로그램이다. 이 프로그램은 레고처럼 블록을 조립하는 방식으로 프로그램을 구성하고 실행시킬 수 있다.
- Sense HAT Emulator는 Sense HAT 애드온 보드에 대한 에뮬레이터 프로그램이다.

- Thonny 파이썬 IDE는 라즈비안 PIXEL 환경에 설치된 새로운 파이썬 통합개발환경이다.
- Sonic Pi는 소리를 생성하는 간단한 프로그래밍 소프트웨어다.
- Wolfram은 Mathematica에서 쓰이는 함수/규칙 기반 프로그래밍 언어다.
- 지니_{geany}는 C/C++ 프로그래밍을 위한 통합개발환경이다.

2 Office

- LibreOffice라는 오픈소스 오피스 프로그램이 제공된다.
- Libreoffice Base는 데이터베이스를 만들고 관리한다.
- Libreoffice Calc는 스프레드시트를 만들고 계산, 정보 분석 및 목록 관리를 수행한다.
- Libreoffice Draw는 그리기 도구를 사용하여 순서도 등을 그리는 프로그램이다.
- Libreoffice Impress는 슬라이드 쇼 등의 프레젠테이션을 만들고 편집한다.
- Libreoffice Math는 수식을 입력하고 편집할 수 있는 프로그램이다.
- Libreoffice Writer는 문서를 만들고 편집할 수 있다.

3 Internet

라즈비안의 기본 웹 브라우저인 크롬_{Chromium} 웹 브라우저와 클로_{Claws} 메일 클라이언트, VNC Viewer에 대한 메뉴가 포함되어 있다.

4 Sound & Video

VLC Media Player는 미디어 파일을 재생하는 프로그램이다.

5 Graphic

그림 보기_{Image Viewer} 및 그누 이미지 처리 프로그램_{Gimp} 등이 포함되어 있다. 그림 보기 프로그램은 GPicView라는 PNG, GIF, JPEG 등 여러 형식의 그림을 화면에 출력하는 간단한 프로그램을 나타낸다. Gimp는 화면을 캡처하거나 이미지를 편집하는 프로그램이다.

6 Games

유명한 Minecraft 게임을 Pi에 적용한 Minecraft Pi와 여러 간단한 파이썬 게임을 포함한 Python Games 메뉴가 포함되어 있다.

7 System Tools

Fcitx는 한글/영문 등의 글자 입력 프로그램이다.

8 Accessories

다양한 보조 프로그램을 제공한다.

- Xarchiver 프로그램은 다양한 아카이브/압축 파일을 만들거나 풀어내는 소프트웨어다.
- Calculator는 간단한 계산기 프로그램이다.
- File Manager는 앞에서도 언급하였듯이 PCManFM이라는 파일 관리자 프로그램을 실행시킨다.
- PDF Viewer는 Xpdf라는 PDF 문서를 읽어 화면에 나타내는 프로그램을 제공한다.
- SD Card Copier는 SD 카드를 복사하고 백업하는 응용 프로그램이다.
- Task Manager, 즉 작업 관리자 프로그램으로 LXTask가 등록되어 있다.
- Terminal 프로그램은 LXTerminal이라는 리눅스용 터미널 에뮬레이터다. 뒷부분에서 학습할 리눅스 명령어들은 이 터미널 프로그램에서 주로 실행할 것이다.
- Text Editor는 Leafpad라는 윈도우의 메모장과 같은 간단한 텍스트 편집기다. 다음 장에서 살펴보겠지만, 라즈비안에서는 Leapad 말고도 nano, vim, emacs와 같은 리눅스 터미널 기반의 텍스트 편집기를 사용할 수 있다.

9 Help

사용자에게 도움이 될 만한 정보를 제공한다.

- Debian References는 라즈비안의 모태인 데비안에 관한 설명과 참고 자료를 보여 준다.
- Raspberry Pi Help는 Raspberry Pi Help 주소*에 대한 메뉴인데, 여러 가지 도움이 될 만한 링크를 포함하고 있다. 이 밖에도 Raspberry Pi Resource 주소**, MagPi 라즈베리 파이 잡지 등에 대한 링크가 포함되어 있다.

10 Preferences

데스크톱의 기본 설정 프로그램을 포함한다.

- Add/Remove Software 항목은 PiPackages라는 프로그램으로 새로운 소프트웨어를 설치하거나 기존의 소프트웨어를 제거하는 기능을 수행한다.

* <http://www.raspberrypi.org/help>

** <http://www.raspberrypi.org/resources>

- Appearance Settings 항목은 전체 윈도우의 테마/아이콘/글꼴과 같은 외양을 설정한다.
- Audio Device Settings 항목은 오디오 장치를 설정한다.
- Display Setting 항목은 모니터 디스플레이의 설정 기능을 수행한다.
- Main Menu Editor 항목은 현재 메뉴를 편집하는 기능을 수행한다.
- Raspberry Pi Configuration 항목은 `raspi-config` 명령을 GUI로 구현한 것으로 라즈베리 파이의 로그인 방식, 네트워크 방식, 하드웨어 인터페이스, 지역화 옵션을 설정하는 기능을 수행한다.
- 입력기 항목은 `im-config` 프로그램으로 입력기를 설정한다.

11 Run

리눅스 명령어를 실행하는 기능을 가진다. 리눅스 명령어는 주로 LXTerminal과 같은 터미널 창에서 이루어지지만, 이 메뉴를 통해서도 실행할 수 있다.

12 Shutdown

메뉴를 선택하면 시스템의 로그아웃 및 종료를 나타내는 창이 팝업^{pop-up}된다. 여기서 **Shutdown**은 완전한 종료, **Reboot**는 재부팅, **Logout**은 로그아웃, **Cancel**은 취소 기능이 수행된다.



그림 1-19 Shutdown 설정 화면

1.6 기본 설정

1.6.1 장치 설정

라즈베리 파이의 패스워드, 네트워크, 부팅 모드, 주변 장치 등을 설정하려면 기본 설정 메뉴의 Raspberry Pi Configuration GUI 프로그램을 사용하거나 `raspi-config` 설정 프로그램을 사용하면 된다.

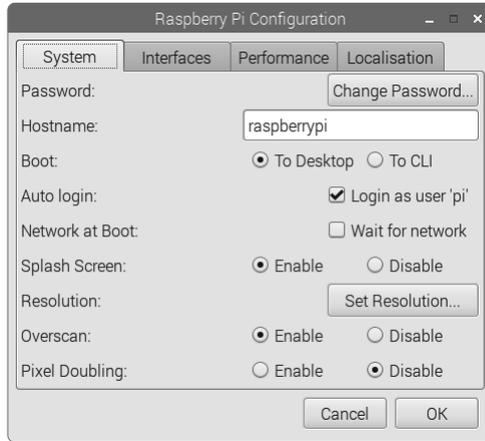


그림 1-20 Raspberry Pi Configuration GUI 프로그램 실행 화면

텍스트 터미널 `LxTerminal`을 실행시켜 보자. 다음과 같은 셸 프롬프트가 나타날 것이다.

```
pi@raspberrypi ~ $
```

- pi: 로그인한 사용자 계정
- raspberrypi: 현재 접속 중인 리눅스 시스템의 호스트 이름
- ~: 현재 작업하고자 하는 디렉터리의 위치

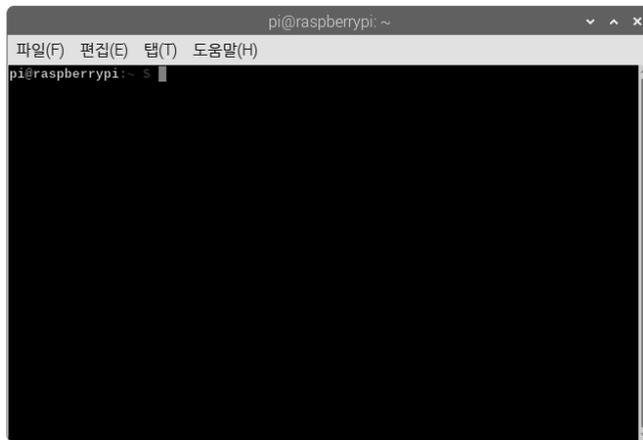


그림 1-21 텍스트 터미널 실행 화면

다음 명령을 입력하면 `raspi-config` 설정 프로그램이 실행된다.

```
$ sudo raspi-config
```

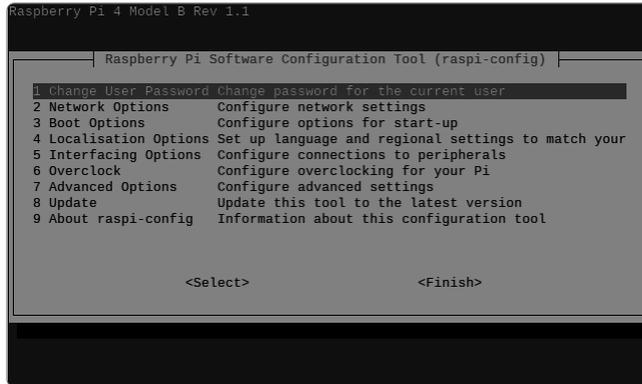


그림 1-22 raspi-config 실행 화면

raspi-config 설정 프로그램은 텍스트 메뉴 방식으로 표 1-2와 같은 기능을 제공한다.

표 1-2 raspi-config 프로그램의 주요 메뉴

메뉴	설명
Change User Password	패스워드 설정(기본 id: pi, 패스워드: raspberry)
Network Options	네트워크 설정(호스트 이름, WiFi SSID/pass, 네트워크 인터페이스 이름 활성화/비활성화)
Boot Options	부팅 모드 설정(데스크톱 GUI/텍스트 CLI, 네트워크 부팅, 스플래시 화면)
Localisation Options	언어, 지역/시간, 키보드, Wifi 국가 등을 설정
Interfacing Options	파이 카메라, SSH, VNC, SPI, I2C, Serial, 1-Wire, 원격 GPIO 등 장치 인터페이스 설정
Overclock	오버 클럭 설정(파이 1-None 700, Modest 800, Medium 900, High 950, Turbo1000MHz, 파이 2-1000MHz, 파이 3-없음)
Advanced Options	파일 시스템 확장, 오버스캔, 메모리 분리, 오디오, 화면 해상도, 픽셀 이중화, GL 드라이버 등 고급 옵션
Update	이 프로그램을 최신 버전으로 갱신
About raspi-config	raspi-config 툴에 대한 설명

여기서 3번 부팅 모드 옵션을 설정해 보자. 부팅 모드 옵션을 선택하여 들어가면 서브 메뉴가 나오는데, 그중에서 첫째 메뉴를 선택하면 다음과 같은 서브 메뉴가 나온다. 여기서 첫째 메뉴는 데스크톱 GUI(X 윈도우) 환경이나 CLI(텍스트 콘솔) 화면으로 부팅하는 등의 설정이고, 둘째 메뉴는 부팅 시에 네트워크 연결을 정하는 설정이다. 셋째 메뉴는 부팅 시 스플래시 화면을 나타낼지 텍스트 화면을 나타낼지를 결정한다.

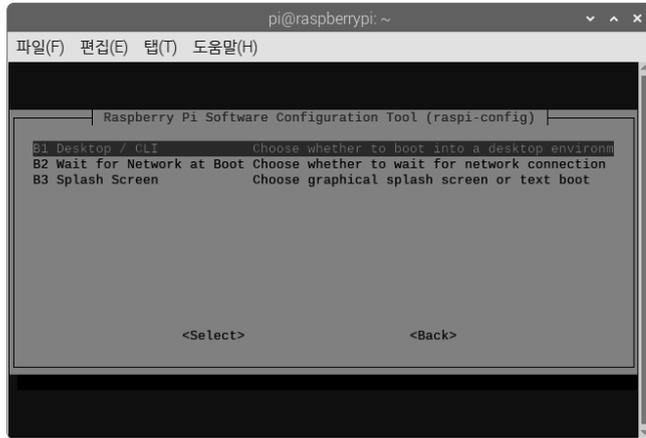


그림 1-23 raspi-config 부팅 옵션 화면

다음, 지역화 옵션에서는 언어, 지역/시간, 키보드의 키를 설정할 수 있다. 라즈베리 파이는 기본적으로 영어/영국/UK 키보드로 설정되어 있으므로 우리나라에 맞게 설정을 바꿔야 한다. 먼저, 언어 설정 Change Locale 메뉴에서 한글을 사용할 수 있도록 ko_KR.EUC-KR과 ko_KR.UTF-8 로케일을 추가한다.

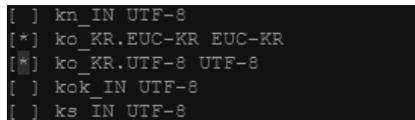


그림 1-24 로케일 추가 옵션 화면

다음으로, 지역/시간을 설정하는 Change Timezone 메뉴에서는 Asia/Seoul을 설정한다.

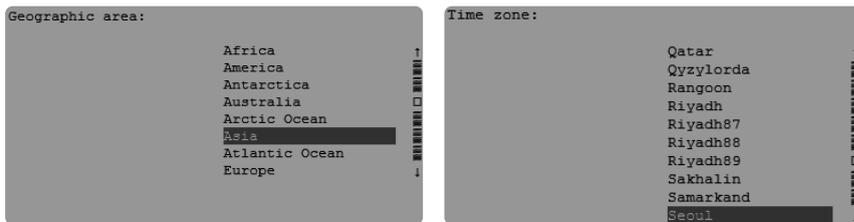


그림 1-25 지역/시간 설정 옵션 화면

다음으로, 키보드 키를 설정하는 Change Keyboard Layout에서는 기본 Generic 105-key(intl) PC 및 en-US 키보드로 설정한다. 기본 UK 키보드 설정에서는 따옴표를 포함해서 몇몇 기호를 나타내는 키 배치가 다르므로 가능하면 변경하는 편이 좋다.

기타 고급 옵션의 메뉴는 표 1-3과 같다.

표 1-3 raspi-config 프로그램의 고급 옵션 메뉴

메뉴	설명
Camera	파이 카메라 설정
SSH	ssh 서버 설정
VNC	그래픽 원격 접속 서버 설정
SPI	커널 SPI 드라이버 활성화 설정
I2C	커널 I2C 드라이버 활성화 설정
Serial	직렬 연결에서 셸 및 커널 메시지 허용 설정
1-Wire	One Wire 인터페이스 설정
Remote GPIO	GPIO의 원격 접속 설정
Expand Filesystem	SD 카드의 파일 시스템 확장
Overscan	모니터 오버스캔 설정
Memory Split	GPU 할당 메모리 설정
Audio	오디오 출력 설정(0 자동, 1 3.5mm 잭, 2 HDMI 포트)
Resolution	화면 해상도 설정(디폴트, 640x480, ...)
Pixel Doubling	2x2 픽셀 매핑 활성화 설정
GL Driver	OpenGL 데스크톱 드라이버 설정(full/Fake KMS, Legacy)

모든 설정이 끝난 후 **Finish** 메뉴를 선택하면 다시 셸로 빠져나온다. 그런 다음, `sudo reboot` 명령을 입력하여 재부팅한다. 만약 라즈베리 파이를 끝내고 싶다면 텍스트 셸에서는 `sudo shutdown -h now`를 입력하고, 그래픽 데스크톱 환경에서는 종료 메뉴에서 **Shutdown**을 선택하면 된다.

1.6.2 한글 설정

라즈베리 파이에서 한글을 사용하도록 설정해 보자. 먼저, GUI 화면 상단의 터미널 아이콘을 클릭하여 텍스트 터미널을 실행시킨 후에 터미널 창에서 다음 명령을 입력하여 시스템 소프트웨어를 갱신한다.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

`fcitx` 한글 입력기는 다음 명령으로 설치한다.

```
$ sudo apt-get install fcitx-hangul
```

아니면 다음과 같은 명령을 실행하면 **ibus** 한글 입력기를 설치할 수 있다.

```
$ sudo apt-get install ibus ibus-hangul
```

다음 명령을 입력하면 한글 폰트를 설치한다.

```
$ sudo apt-get install fonts-unfonts-core
```

한글 입력기를 설정하기 전에 설정 메뉴에서 **Input Method** 프로그램을 실행한 후에 **fcitx**로 입력기를 설정해야 한다. 만약 **fcitx** 항목이 보이지 않으면 시스템 언어를 **en_US**로 바꾸고 다시 설정하면 된다.

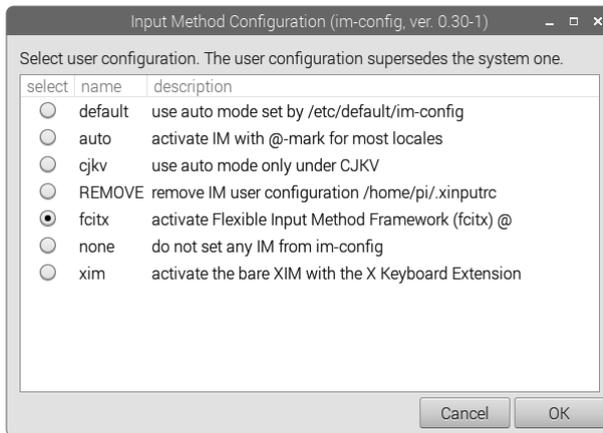


그림 1-26 Input Method 프로그램의 입력기 설정 화면

한글 입력기가 설치되면 화면 상단 메뉴 바의 오른쪽에서 입력 아이콘을 확인할 수 있다. 이 아이콘을 클릭하고 환경설정 창이 뜨면 입력 방식에서 ‘+’ 버튼을 클릭하여 **Hangul** 입력기를 추가하면 된다. 한/영 전환은 **Ctrl**+**Space** 키다. 즉, **Ctrl** 키를 누른 상태에서 **Space** 키를 누르면 된다. 참고로, **raspi-config** 프로그램을 실행하여 **국제화 옵션의 언어 설정** Change Locale 메뉴에서 한글을 사용할 수 있도록 **ko_KR.UTF-8** 로케일을 추가하고 선택해야 한다.

1.6.3 네트워크 설정

인터넷을 사용할 수 있도록 네트워크를 설정해 보자. 유선 인터넷을 사용하는 경우에는 유선 인터넷 케이블을 연결한 후 공유기에 연결하면, 보통은 DHCP 기능을 통해 자동으로 IP가 설정된다. 수동으로 설정하는 경우에는 설정 파일(/etc/network/interfaces)을 수정하거나 다음과 같이 `ifconfig` 명령을 사용하여 설정하면 된다.

```
$ sudo ifconfig eth0 <ip 주소> up
```

무선 WiFi일 때도 리눅스가 WiFi 동글을 인식하면 LXDE X 윈도우 기본 화면에서 오른쪽 상단의 네트워크 설정 아이콘을 클릭하여 공유기에 접속하면 된다. 아니면 다음 명령을 실행하여 `wpa_gui`라는 WiFi 설정 프로그램을 설치하여 활용할 수도 있다.

```
$ sudo apt install wpa_gui
```

WiFi를 수동으로 설정하는 경우에는 다음과 같은 파일을 수정하여 공유기의 SSID와 KEY를 입력하면 된다.

```
$ sudo vi /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="<SSID>"
    psk="<WPA KEY>"
}
```

1.7 라즈베리 파이 관련 사이트

국내외에 라즈베리 파이와 관련된 정보를 얻을 수 있는 수많은 사이트가 존재한다. 라즈베리 파이와 관련된 사이트를 소개하겠다. 먼저, 표 1-4는 라즈베리 파이와 관련해 필요한 정보를 얻을 수 있는 다양한 커뮤니티 사이트 목록이다.

표 1-4 라즈베리 파이 관련 주요 커뮤니티 사이트

분류	주소	설명
해외	www.raspberrypi.org	공식 라즈베리 파이 사이트이며, 공식적인 정보는 물론 커뮤니티 포럼을 통해 각종 관련 정보를 얻을 수 있다.
	learn.adafruit.com	에이다프루트 학습 시스템이며, 다양한 장치 정보를 얻을 수 있다.
	elinux.org	임베디드 리눅스 카페이며, 특히 하드웨어 관련 정보를 얻을 수 있다.
	stackoverflow.com	대표적인 질문 및 응답 사이트이며, 특정 문제에 대한 해결책을 찾을 수 있다.
국내	cafe.naver.com/pipc	라즈베리 파이 네이버 카페이며, 라즈베리 파이를 비롯해 각종 임베디드 보드에 대한 정보를 얻을 수 있다.
	www.oroqa.org	오픈소스 하드웨어&소프트웨어 카페이며, 오픈된 정책으로 다양한 강좌를 볼 수 있다. 특히, 로봇과 관련된 정보가 많다.
	www.rasplay.org	산딸기마을 사이트이며, 다양한 정보가 한글 문서로 많이 수록되어 있다.

표 1-5는 라즈베리 파이 및 관련된 부품들을 판매하는 사이트다.

표 1-5 라즈베리 파이 관련 부품의 주요 판매 사이트

분류	품목	사이트
해외	공식 배포처	<ul style="list-style-type: none"> • www.element14.com • www.allidelec.com • uk.rs-online.com
	공식 부품 및 액세서리	<ul style="list-style-type: none"> • swag.raspberrypi.org
	각종 부품 및 액세서리	<ul style="list-style-type: none"> • www.adafruit.com • www.sparkfun.com • www.waveshare.com
국내	각종 부품 및 액세서리	<ul style="list-style-type: none"> • www.eleparts.com • www.icbanq.com • www.devicemart.co.kr



2

리눅스 환경

리눅스 시스템을 잘 이해하고 프로그래밍하기 위해서는 먼저 사용 환경에 익숙해져야 한다. 리눅스 시스템은 유닉스^{Unix}의 많은 기능을 가져왔으므로 유닉스의 파일 시스템, 텍스트 셸 환경, 명령어 등에 대해서도 잘 이해해야 한다. 이번 장에서는 리눅스를 접속하는 방법, 파일 시스템, 기본 명령어들에 대해 다루기로 한다.

대부분의 좋은 프로그래머는
대중이 돈을 지불하거나 추앙하기를 기대하기 때문이 아니라
재미있기 때문에 프로그래밍을 한다.

- 리누스 토발즈(Linus Torvalds), 리눅스 OS 제작자

2.1 리눅스 접속

리눅스 시스템 접속은 리눅스가 설치된 컴퓨터에 직접 접속하는 방법과 원격으로 접속하는 방법이 있다. 직접 접속하는 방법은 앞 장에서 이미 살펴보았으므로 여기서는 원격 접속 방법을 알아보자.

2.1.1 원격 접속

리눅스는 다중 사용자용으로 만들어졌으므로 하나의 머신_{machine}만 있으면 여러 사람이 네트워크를 통해 원격으로 접속해서 사용할 수 있다. 이런 경우에는 리눅스가 설치된 서버 머신에 사용자 계정을 등록하고 특정 네트워크 접속 프로그램을 사용하여 원격으로 리눅스 시스템에 접속하며, 접속 후에는 그래픽 사용자 인터페이스_{Graphic User Interface, GUI} 또는 기본적인 텍스트 터미널_{text terminal}과 비슷한 환경에서 프로그래밍을 할 수 있다. 프로그래밍을 하기 위해서는 vi 에디터_{editor}와 C 컴파일러_{compiler}인 GCC_{GNU Compiler Collection} 등을 사용하여야 하는데, 자세한 내용은 뒤에서 자세히 살펴보겠다.

원격으로 접속하는 방법 중 텔넷_{telnet}이나 SSH_{Secure SHell} 프로그램을 이용하여 텍스트 터미널 모드에서 원격으로 리눅스 시스템에 접속하는 방법이 있다.

텔넷은 윈도우의 시작 메뉴 실행 창에 telnet을 입력하면 프로그램이 실행되고, 이것으로 리눅스 서버의 IP만 입력하면 쉽게 접속할 수 있는 전통적인 접속 방식이다. 하지만, 전달되는 메시지가 보안이 되지 않는다는 문제점이 있다. 반면에 SSH는 메시지를 암호화하여 전송하므로 비교적 안전하며, 더 많은 사람이 사용한다.

여기서는 MS 윈도우 환경에서 SSH 접속을 지원하는 PuTTY 프로그램을 설치하고, 라즈베리 파이에 접속하는 과정을 알아보자. 먼저, 라즈베리 파이의 설정 프로그램_{raspi-config}을 실행하여 메뉴에서 SSH 항목을 활성화_{enable}시킨다. 그리고 PuTTY 홈페이지*의 Download 링크로부터 puTTY.exe 파일을 다운로드한다. 다운로드한 puTTY.exe를 실행하고, 그림 2-1과 같은 창에서 라즈베리 파이 IP를 입력하고 접속을 시도한다.

* <http://www.chiark.greenend.org.uk/~sgtatham/putty>

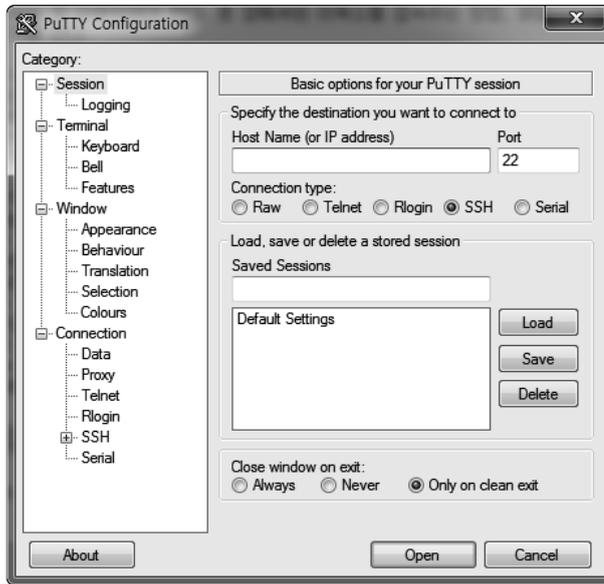


그림 2-1 PuTTY 설정 창

처음 접속을 시도하면 보안을 위한 공개 키(public key) 설정 화면이 나오는데, 이때 **yes**를 선택하면 접속이 되고 로그인 메시지가 나타난다. 로그인 시에는 사용자 ID와 패스워드로 인증 절차를 거치는데, 올바른 경우라면 우리가 명령을 입력할 수 있는 프롬프트와 커서를 보여 준다. **login**에는 자신의 아이디를 넣고 **password**에는 알고 있는 비밀번호를 넣는다. 라즈베리 파이의 기본 아이디는 **pi**이고 패스워드는 **raspberrypi**다.

```
login : pi
password : *****
```

로그인을 하면 텍스트 창에서 리눅스 셸(shell)을 사용할 수 있는 상태가 되며, 다음과 같은 프롬프트(prompt)가 나타난다.

```
pi@raspberrypi ~ $
```

로그인 상태에서 **logout**을 입력하면 시스템에서 빠져나간다. **logout** 외에도 **(Ctrl)+D**, **exit**도 **logout**과 같은 기능을 한다.

2.1.2 그래픽 모드 접속

이번에는 원격 접속 방법 중 그래픽 모드 접속 방법에 대해 알아보자. 그래픽 모드 접속 방법은 리눅스의 X 윈도우 환경을 원격 데스크톱에 나타나게 하여 원격에서 GUI를 직접 사용할 수 있도록 하는 방식이다. 이것을 가능하게 하는 것으로는 VNC Viewer, XRDP, Xmanager, Xming 등과 같은 다양한 프로그램이 있다. 이들은 사실 리눅스 시스템상의 X 윈도우 클라이언트 프로그램을 원격으로 구동시켜 주는 일종의 디스플레이 서버라고 할 수 있다. 이런 전송 방식 중에서 많이 사용하는 방식으로는 잘 알려진 VNC나 윈도우의 원격 데스크톱 프로토콜인 RDP 등이 있다. 여기서는 VNC와 NX에 대해 알아보자.

VNC(Virtual Network Computing)는 그래픽 데스크톱 환경으로 원격 컴퓨터를 제어하는 소프트웨어의 일종이며, 예전부터 오픈소스로 공개되어 많이 사용되어 왔다. 라즈베리 파이의 라즈비안에는 RealVNC를 구동하는 기능이 포함되어 있다.

먼저, 라즈베리 파이에서는 VNC 서버를 설치해야 한다. 설치하는 명령은 다음과 같다.

```
$ sudo apt-get install realvnc-vnc-server
```

이후 VNC 서버를 구동하려면 시작 메뉴에서 1장에서 다룬 라즈베리 파이 설정 메뉴를 클릭하거나, 텍스트 셸에서 `sudo raspi-config`를 실행하고 **Interfaces** 탭에서 **VNC Enable** 항목을 클릭하면 된다. 그러면 오른쪽 상단에 VNC 아이콘이 생성될 것이다. 아이콘을 클릭하면 VNC 뷰어가 연결할 IP 주소를 확인할 수 있다.

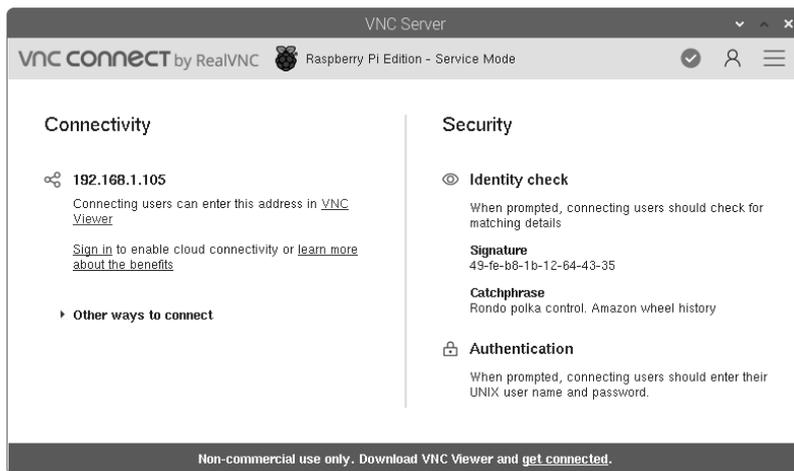


그림 2-2 RealVNC 서버 프로그램 실행 화면

이제 윈도우 등을 사용하는 다른 컴퓨터에 VNC 뷰어 프로그램을 설치해야 한다. 여기서는 RealVNC 뷰어를 설치해 보겠다. RealVNC 홈페이지*로부터 VNC Viewer 프로그램을 다운로드하고 설치한다. 시작 메뉴에서 RealVNC Viewer 프로그램을 찾아 실행하고 Remote Host 입력 창에 라즈베리 파이 IP 주소를 입력하고 Connect 버튼을 클릭하면 패스워드 입력 창이 나타난다. 서버에서 설정해 놓은 패스워드를 입력하면 라즈베리 파이로 접속되어 VNC 화면에 나타날 것이다.

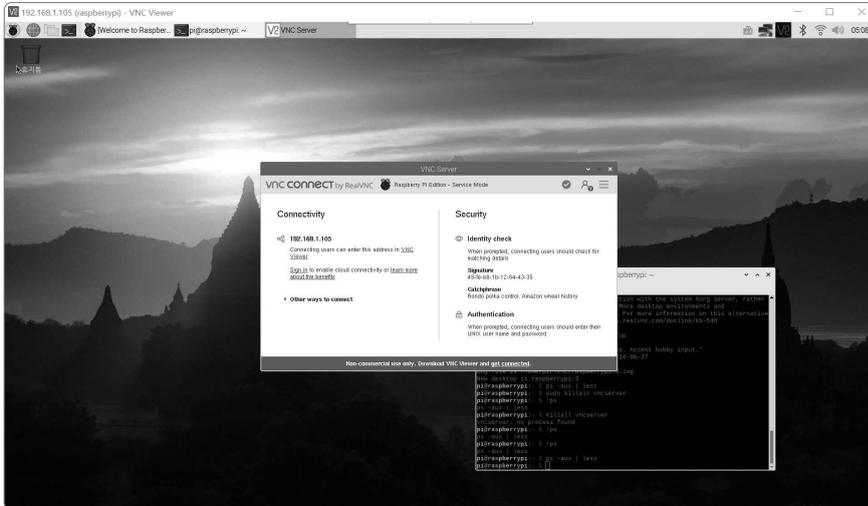


그림 2-3 VNC 접속 화면

그런데, VNC에서는 오디오 기능을 지원하지 않는다. 게다가 창을 움직여 보면 그래픽 처리 속도 역시 썩 빠르지 않다는 걸 느낄 것이다. 다른 데스크톱 GUI 환경이나 하나의 응용 윈도우만 실행하는 기능도 없다. 따라서, 이번에는 이런 기능을 지원하는 NX 방식을 사용해 보기로 하자. NX는 NoMachine에서 만든 전송 프로토콜이다. 이미지 압축 전송 방식을 사용하기 때문에 다른 방식에 비해 비교적 빠른 구동이 가능하다. NX 방식은 현재 상용 정식 버전과 무료 X2Go 버전을 사용할 수 있다.

X2Go부터 사용해 보자. 먼저, 라즈베리 파이에 X2Go 서버를 설치해야 한다. 그 전에 X2Go는 SSH 프로토콜로 접속하므로 라즈베리 파이 설정 프로그램(raspi-config)을 실행하여 고급 설정 메뉴에서 SSH 항목을 활성화(enable)해 둔다. 다음, 텍스트 셸에서 다음과 같은 명령을 실행하여 X2Go 서버를 설치한다. 그러면 X2Go 서버가 서비스로 실행된다.

* <https://www.realvnc.com/en/connect/download/viewer>

```
$ sudo apt install x2goserver
```

X2Go는 GNOME, LXDE, XFCE, Mate, KDE 등 다양한 데스크톱 환경을 사용할 수 있는데, 이런 데스크톱 환경을 간편하게 설치하기 위하여 다음과 같은 명령을 실행한다.

```
$ sudo tasksel
```

여기서, X2Go에서 오류가 거의 없고 가볍고 빠르게 실행할 수 있는 XFCE, LXDE 환경을 선택하여 설치한다.

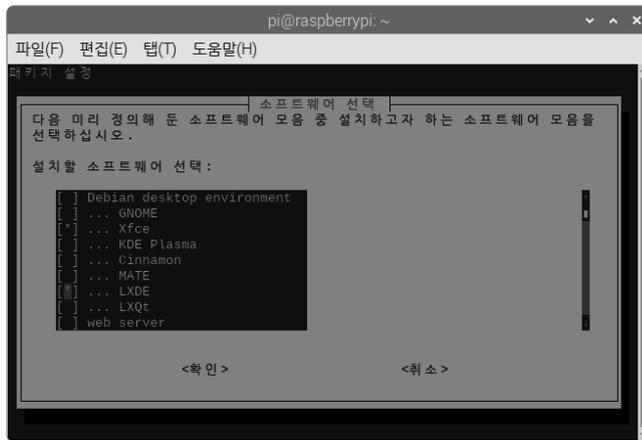


그림 2-4 tasksel 프로그램을 이용한 데스크톱 환경 설치 화면

이제 MS 윈도우 등의 데스크톱 환경에서 X2Go 클라이언트 프로그램을 설치해 보자. X2Go 위키 홈페이지(<https://wiki.x2go.org/>)에서 윈도우용 X2Go 클라이언트 프로그램을 다운로드하고 파일을 클릭하여 설치한다. 설치가 완료되면 바탕화면에 X2Go Client 아이콘이 등록된 것을 확인할 수 있다.

아이콘을 클릭하여 X2Go 클라이언트 프로그램을 실행한 후에 메뉴에서 **New Session**을 생성하면, 그림 2-5와 같은 대화 창이 나타난다. 여기서 세션 이름(예) raspberrypi), 호스트 IP, 로그인 ID(pi), SSH 포트(22), 세션 종류(예) XFCE 또는 LXDE) 등을 설정한 후 저장한다.



3

파이썬 환경

라즈베리 파이 리눅스 환경에서는 C, C++, 자바^{Java}, 파이썬^{Python} 등 다양한 프로그래밍 언어를 사용할 수 있다. 이번 장에서는 이 책에서 가장 많이 활용하는 파이썬 언어에 대해 학습한다. 아울러 파이썬 언어의 문법과 파이썬 모듈 중에서 파이게임^{pygame}에 대해서 살펴본다.

우리가 항상 완벽한 문서를 쓰더라도,
코드를 읽을 수 없다면 거의 재사용할 수 없을 것이다.

- 귀도 반 로섬(Guido van Rossum), 파이썬 언어 제작자

3.1 파이썬 개요

파이썬은 1991년에 귀도 반 로섬 Guido Van Rossum 이 만든 프로그래밍 언어이며, 수많은 장점을 가지고 있어서 전 세계에서 가장 폭넓은 사용자들에게 이용되는 언어 중 하나다. 파이썬의 주요 특징은 다음과 같다.

파이썬은 C, 자바 등과 같은 고급 언어이면서 인터프리터 Interpreter 언어다. C나 자바 언어에서는 프로그램을 작성한 다음 컴파일러 compiler 프로그램이 실행 파일로 변환된 이후에 실행할 수 있다. 반면에 파이썬과 같은 인터프리터 방식은 프로그램을 작성한 다음 곧바로 실행해 결과를 확인할 수 있다. 더구나 파이썬은 인터프리터를 제공하기 때문에 프로그램 전체를 미리 작성할 필요 없이 프로그램 코드를 작성하면서 동시에 실행할 수 있다.

파이썬은 문법적으로 매우 쉽고 직관적이어서 프로그램을 처음 배우는 초보자도 쉽게 배울 수 있다. 그만큼 이해하기 쉽고 코드가 간결하다. 뿐만 아니라, 절차적인 프로그래밍 방식 외에 함수형 프로그래밍 방식과 객체지향적인 클래스도 지원하여, 특정한 기능을 모듈로 패키징화하거나 다른 언어로 된 모듈을 포함하여 확장할 수 있다.

3.2 파이썬 실행

파이썬 프로그램은 파이썬 인터프리터가 실행한다. 파이썬 인터프리터는 보통 텍스트 명령 셸에서 python을 입력하여 실행하거나 IDLE이나 Thonny와 같은 통합개발환경 Integrated Development Environment, IDE 을 통해 실행할 수 있다. 시작 메뉴에서 Thonny IDE 아이콘을 클릭하여 파이썬 인터프리터를 실행시켜 보자(다음부터 python 3.x 버전을 기준으로 설명한다).

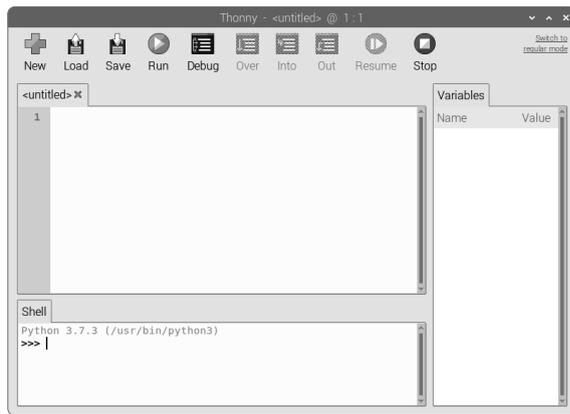


그림 3-1 Thonny 파이썬 IDE 실행 창

이번에는 Thonny IDE를 사용하는 대신에 텍스트 셸에서 다음과 같이 python을 입력해 보자. 파이썬 인터프리터가 실행되면 대화형 셸의 프롬프트 '>>>'가 출력된다.

```
$ python3
```

여기서 다음과 같이 'Hello World'가 출력하는 명령을 입력시켜 보자.

```
>>> print("Hello World")
Hello World
>>> quit()
$
```

파이썬의 대화형 셸은 유효한 표현식을 입력하는 즉시 그 결과를 출력하므로 여러 가지로 유용하다. 예를 들면, 다음과 같은 수학식을 입력하여 계산기로 활용할 수도 있다.

```
>>> 1 + 2 + 3
6
>>> (0 < 1) and (2 < 3)
1
>>> not (1 == 2)
1
```

프로그램을 여러 번 반복해서 실행하고 싶으면 스크립트 파일로 만들어 실행하면 된다. 한 예로, 다음과 같이 파일 편집기를 사용하여 hello.py 파일을 만들어 보자.

코드 3-1 Hello World를 출력하는 hello.py 파이썬 스크립트 파일

```
#!/usr/bin/python3
# helloworld.py
print("Hello World")
```

여기서 '#' 기호는 대부분의 스크립트 언어에서처럼 주석문을 나타낸다. 파일의 첫째 줄의 '#' 주석문은 실행하는 인터프리터를 나타낸다. 여기서는 파이썬 인터프리터를 가리킨다.

파이썬 스크립트를 실행하려면 텍스트 셸에서 다음과 같이 입력하면 된다.

```
$ python3 hello.py
Hello World
```

3.3 변수와 자료형

3.3.1 변수

파이썬에서 변수는 대부분의 고급 언어처럼 알파벳 문자로 시작하여 숫자를 포함해 식별자(이름)로 지정한다. 변수는 '=' 기호를 사용하여 값을 할당한다. 이때, 동적으로 생성되어 값이 할당되므로 미리 자료형을 선언할 필요는 없다. 다음과 같은 코드를 입력해 보자.

```
>>> n = 1
>>> name = 'Kim'
>>> n = n + 2
>>> value = 1.2 * n
>>> print("%f is 1.2 * %d" % (value, n))
3.600000 is 1.2 * 3
```

여기서 변수는 정수, 실수와 같은 숫자는 물론 문자열 등 다양한 자료형을 대입하거나 자료형을 변경시켜 저장할 수 있다는 것을 알 수 있다. 변수는 임의의 식별자, 즉 이름을 사용할 수 있지만, 미리 예약어(keyword)로 정의된 단어는 사용하지 못한다. 파이썬의 예약어는 대략 30여 개인데 다음과 같다.

and	as	assert	break	class	continue	def	del
elif	else	except	exec	finally	for	from	global
if	import	in	is	lambda	nonlocal	not	or
pass	print	raise	return	try	while	with	yield

다음엔 파이썬의 자료형에 대해 알아보자.

3.3.2 상수와 문자열

파이썬은 정수, 실수, 복소수, 부울(boool) 상수와 같은 네 종류의 숫자 상수를 지원한다. 부울 상수는 True 또는 False 값을 가진다. 이들 숫자에 대하여 +, -와 같은 단항 연산과 +, -, *, /, %, **과 같은 덧셈, 뺄셈, 곱셈, 나눗셈 등의 사칙연산, 그리고 나머지, 제곱 연산 등을 기본적으로 사용할 수 있다. 더구나 정수에 대해서는 ~, &, |, ^, >>, <<와 같은 반전과 AND, OR, XOR, 오른쪽 이동(shift right), 왼쪽 이동(shift left)과 같은 비트 연산도 사용할 수 있다.

```

>>> 1 + 1
2
>>> 1234567L - 2345678L
-1111111L
>>> 4 / 5
0
>>> 4.0 / 5.0
0.8
>>> 24.5+2.5j + 5.3+10.7j
(29.8+13.2j)
>>> 4 ** 2
16
>>> 40 & 35
32
>>> 35 << 1
70
>>> 60 >> 2
15

```

숫자 연산 말고도 <(작음), >(큼), ==(같음), !=(다름), >=(크거나 같음), <=(작거나 같음)과 같은 비교 연산자들을 사용할 수 있다. 이 연산자들은 두 값을 비교하여 결과가 참이면 True, 거짓이면 False라는 부울 값을 반환한다.

```

>>> 1 > 1
False
>>> 0 < 2
True
>>> 10L == abs(-10)
True
>>> 100.0 != 100
False

```

파이썬은 ' '(단일따옴표), " "(이중따옴표), ''' '''(삼중따옴표)로 둘러싸인 문자열 상수를 지원한다. 이들 문자열은 인덱스 연산자 []를 사용하여 한 문자 또는 부분문자열을 추출할 수 있다.

```

>>> a = "Hello World"
>>> a[0]
'H'
>>> a[5:8]
'Wo'
>>> a[:2]
'He'
>>> a[-1]

```

```
'd'
>>> a * 2
'Hello WorldHello World'
>>> b = "Goodbye"
>>> a + b
'Hello World Goodbye'
```

문자열에서 가장 중요한 기능 중 하나는 특정한 포맷(형식)으로 쉽게 변환할 수 있다는 점이다. 파이썬에서는 두 가지 방식으로 문자열 포맷을 변환할 수 있다. 첫째는 ‘포맷 문자열 % 변환 값들’ 형식을 사용하여 포맷 문자열을 변환 값들이 적용된 문자열로 변환할 수 있다. 이때, 포맷 문자열 내에 변환 지정자를 정의하여 변환 값들로 매핑할 수 있도록 한다. 여기서 포맷 문자열 내의 변환 지정자는 ‘%[변경자]<문자>’로 표현된다. 여기서 변환 문자는 d(정수), x(16진수), f(부동소수점), s(문자열), c(문자) 등이 가능하다. 변경자는 없어도 되는 옵션인데, 숫자의 경우에는 출력의 최대 폭을 나타내고 괄호로 둘러싸인 키 값을 지정할 수도 있다.

다음과 같은 예를 실행시켜 보자.

```
>>> i = 30
>>> f = 3.141592
>>> s = "python"
>>> print("i=%d, f=%f, s=%s" % (i, f, s))
i=30, f=3.141592, s=python
>>> p = {'name': 'HongGilDong', 'age': 30}
>>> print("name=%(name)s age=%(age)d" % p)
name=HongGilDong age=30
```

여기서 p는 특정한 키 값에 해당하는 값을 저장하는 사전_{dict} 객체를 나타내는 자료형이다. 사전에 대해서는 나중에 다시 설명하겠다.

두 번째 변환 방법은 문자열에 대해 format() 함수를 사용하는 것이다. 형식은 ‘포맷 문자열.format(인자들)’이다. 여기서 포맷 문자열 내의 변환 지정자의 형식은 {숫자}, {이름}이다. {숫자}인 경우 해당하는 format()에 지정된 인자들 중 해당 위치의 인자로 대체된다. {이름}인 경우 인자들 중 같은 이름의 키워드를 가진 인자 값으로 대체된다. 다음 예를 살펴보자.

```
>>> print("i={0} f={1} s={2}".format(30, 3.141592, 'python'))
i=30 f=3.141592 s=python
>>> print("name={name} age={age}".format(name='HongGilDong', age=30))
name=HongGilDong age=30
```



11

사물인터넷 보드 활용

이번 장에서는 사물인터넷을 위한 임베디드 보드 중에서 ESP8266 NodeMCU 및 ESP32 NodeMCU32 보드를 활용하여 LED, 온도/습도 센서 등을 제어하고 웹으로 모니터링하는 방법에 대해 알아본다.

사물인터넷은 많은 제품과 서비스를 만드는 데 드는 한계비용을 거의 제로에 가깝게 생산성을 높여, 사실상 공짜로 만들게 한다.

- 제레미 리프킨(Jeremy Rifkin), 《한계비용 제로 사회》 중에서

11.1 사물인터넷 보드 개요

최근 몇 년 간 누구나 아이디어만 있으면 3D 프린터나 오픈소스 하드웨어 및 소프트웨어를 활용하여 제품을 제작하는 메이커^{maker} 운동 붐이 불면서 그 인기가 계속 확산되고 있다. 이렇게 된 데에는 개발자들의 자발적인 참여 등 여러 이유가 있겠지만, 무엇보다 성능이 우수하면서도 값싼 전자 부품들이 점점 다양하게 개발된 것과, 아두이노를 비롯한 수많은 개발자들이 제공한 오픈소스 하드웨어 및 소프트웨어들로 누구나 쉽게 프로토타입을 만들 수 있게 된 것이 큰 이유일 것이다.

사물인터넷을 위한 임베디드 보드도 마찬가지다. 아두이노가 선구자의 길을 걸었고, 라즈베리파이와 비글본^{BeagleBone}을 비롯한 다양한 보드가 그 길을 따라가고 있다. 이외에도 수많은 보드가 있다. 심지어 10달러 미만의 가격으로 리눅스가 구동되는 보드도 존재한다. 여기서는 이러한 보드들 중 10달러 미만이면서도 와이파이 기능이 포함된 에스프레시프 시스템즈^{Espressif Systems}의 두 종류 MCU 보드를 소개하고 활용 가능성을 살펴보겠다.

11.2 ESP8266 모듈을 이용한 아두이노 프로그래밍

11.2.1 ESP8266 개요

ESP8266은 에스프레시프 시스템즈에서 만든 와이파이 Soc 칩 모듈이다. 이 칩은 텐실리카^{Tensilica}의 80MHz의 Xtensa LX3 RISC CPU를 기반으로 64KB DRAM, 96K SRAM 및 512KB의 SPI 플래시를 포함하고 있으며, TCP/IP 프로토콜 스택이 탑재되어 있다. 또한, 이후에 출시된 ESP-12/ESP-12E ESP8266 모듈은 64KB DRAM, 96K SRAM 및 4MB의 SPI 플래시를 포함하고 있으며 전원 포트를 합쳐 30개의 외부 포트를 통해 11개의 GPIO, ADC 1포트, UART, I2C, SPI, PWM을 사용할 수 있다. 특히 이들 칩은 다양한 모듈로 판매되고 있는데, 대부분의 모듈이 1~5달러에 불과하다.

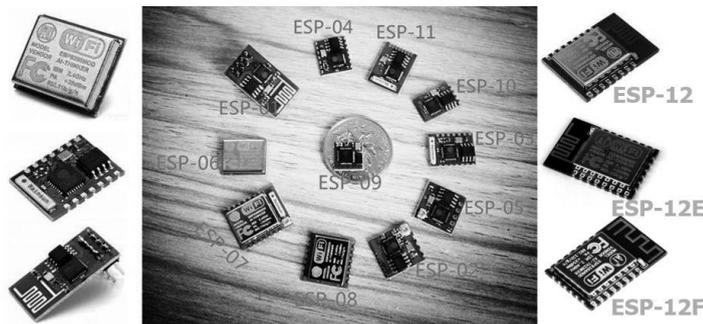


그림 11-1 다양한 ESP8266 모듈

[출처] AliExpress & esp8266 wiki

또한, ESP8266 모듈을 아두이노와 연결할 수 있도록 제작된 ESP8266 와이파이 실드들도 쉽게 구할 수 있다. 예를 들면, 다음과 같은 ESP-12E 제품은 ESP-12E 모듈이 탑재되어 있으며 중국에서 제작되어 가격이 매우 저렴하다. 그런데 이 제품은 불량 부품을 사용한 카피 제품이 있으니 주의해서 사용해야 한다*.

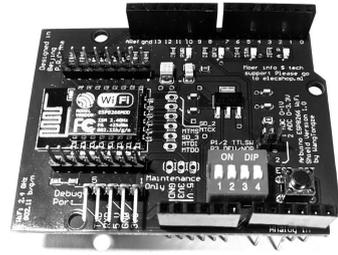


그림 11-2 WangTongZe에서 제작한 ESP8266 와이파이 실드

어쨌든, ESP8266 모듈을 활용하면 여러 방식으로 펌웨어를 프로그래밍할 수 있다. 제조사에서 공식적으로 제공한 AT 명령어 방식의 AT 펌웨어가 탑재된 기본 ESP8266 모듈을 아두이노 보드와 연결해서 사용할 수도 있고, ESP8266 모듈의 MCU 기능을 활용하여 아예 ESP8266만으로 동작하는 펌웨어를 제작할 수도 있다. 또한, Lua 인터프리터^{interpreter}가 내장된 Nodemcu 펌웨어 또는 Micropython 등의 인터프리터를 펌웨어로 탑재하여 스크립트 프로그램을 실행시킬 수도 있다.

11.2.2 AT 펌웨어 테스트

먼저, ESP-01 모듈 또는 ESP8266 와이파이 실드를 테스트해 보자. ESP8266 모듈을 테스트하거나 프로그램을 업로드하기 위해서는 ESP8266 모듈의 UART 핀을 통해 컴퓨터와 연결할 수 있는 별도의 USB-to-Serial 장치가 필요하다. USB-to-Serial 장치가 없다면 대신 아두이노 보드를 활용해도 된다. ESP-01 및 ESP-12E 모듈의 핀 배치는 그림 11-3과 같다.

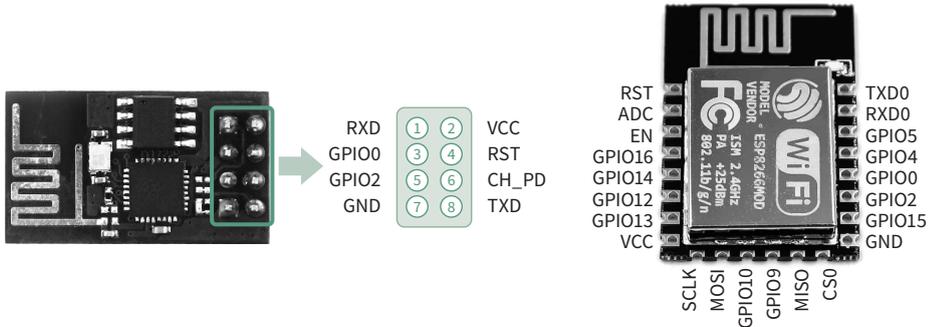


그림 11-3 ESP-01 및 ESP-12E 모듈의 핀 배치

* 보드 상단 오른쪽에 'More' 대신 'Moer'라고 쓰여진 제품은 아두이노와 ESP8266의 UART Rx(D0), Tx(D1) 신호를 연결하는 레벨 컨버터(level converter) 회로에 J1Y 트랜지스터가 아닌 다른 J3Y 칩이 사용하여 제대로 동작하지 않는다. 따라서 실드의 D0, D1 핀을 사용하는 대신 ESP-12E 모듈의 TxD, RxD 핀을 직접 아두이노에 연결해야 한다.

이제 다음과 같은 회로를 제작한다. 이 회로는 부품을 브레드보드에 끼우거나 만능 기판에 납땀하여 구성한다. 회로에 필요한 부품에 대한 설명은 다음과 같다. 먼저, 정전압 레귤레이터 regulator인 LM1117 3.3V 칩을 사용하여 아두이노의 5V 전원을 3.3V로 변환하고 ESP8266 모듈의 전원을 공급한다. ESP8266 모듈의 UART Tx 및 Rx 핀은 아두이노 UNO 보드의 Tx, Rx와 그대로 연결하되, Rx 선은 다음 그림과 같은 전압 분배 회로를 구성하여 5V 입력을 3.3V로 낮추어야 한다. 또, CH_PD 핀도 3.3V로 연결하고 RESET 핀도 스위치를 연결하여 동작하도록 구성한다. 펌웨어를 업로드할 때는 GPIO0 핀을 0V(GND)로 연결해야 하므로 토글 스위치 또는 접퍼와 1k Ω 저항을 사용하여 회로를 구성한다.

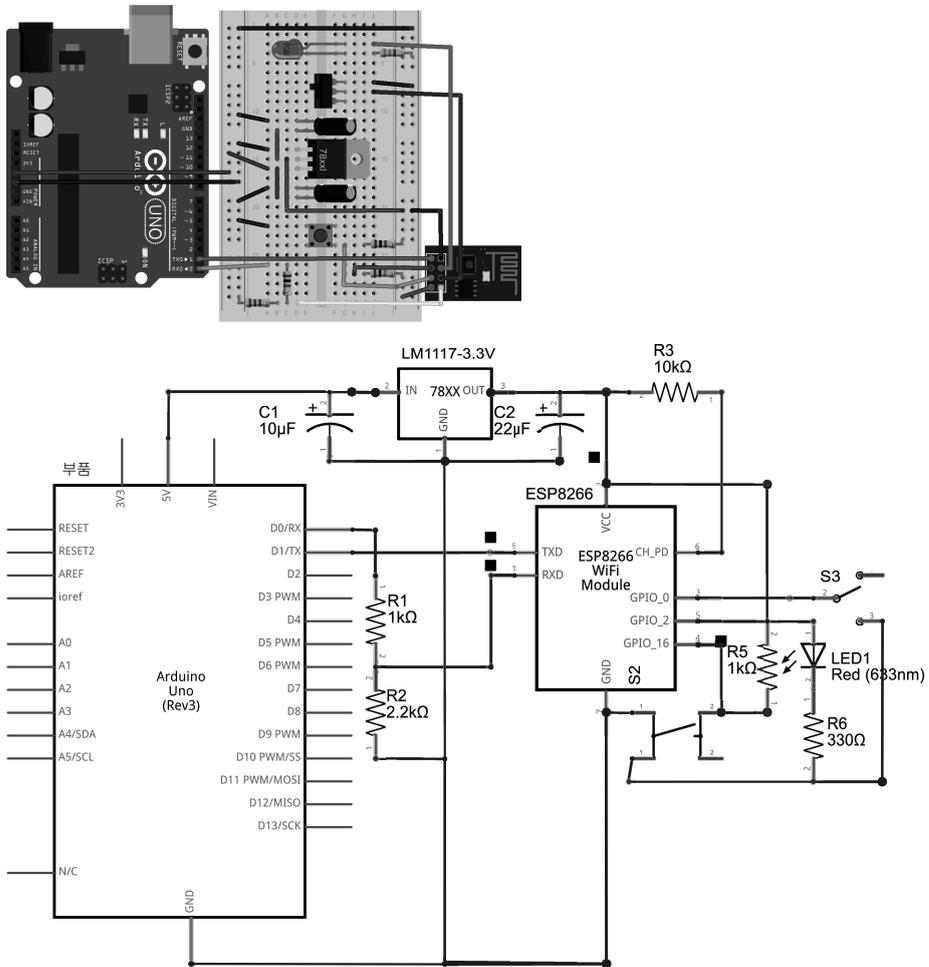


그림 11-4 아두이노와 ESP8266 모듈 연결 회로

만약 ESP8266 와이파이 실드를 사용한다면 실드의 TxD, RxD 핀을 CP2102 UART 보드 또는 아두이노 보드에 연결하면 된다. 이때 실드의 DIP 스위치는 일반 모드이므로 앞의 그림 11-2처럼 P1, P2, P3, P4는 모두 아래로 내려져야 한다.

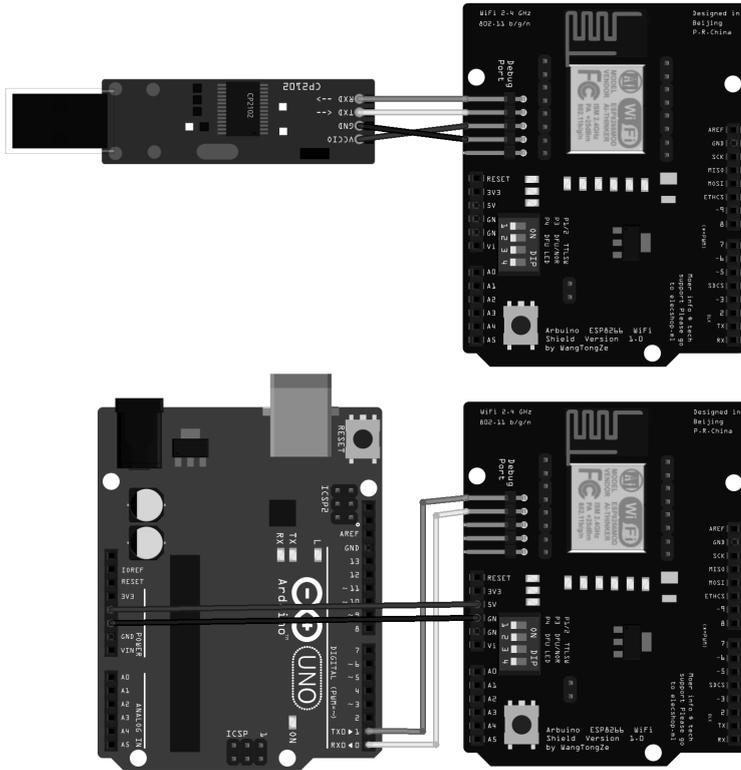


그림 11-5 ESP8266 와이파이 실드와 CP2101 보드 또는 아두이노 연결 회로

아두이노 UNO 보드는 컴퓨터와 ESP8266 모듈 간의 인터페이스 역할만 하므로 전원을 연결하기 전에 일자 드라이버 등을 이용해 아두이노 보드의 atmega328p CPU 칩을 제거해 둔다.

참고로, atmega32u4 칩을 사용하는 아두이노 레오나르도 보드 및 프로 미니 보드는 보드상의 하드웨어 Rx, Tx 핀이 USB-to-Serial 칩과 직접 연결되어 있지 않으므로, 전송되는 데이터를 포워딩하도록 아두이노 스케치 프로그래밍을 해야 한다.

이제 리눅스의 USB 포트를 UART 직렬 콘솔로 연결하는 프로그램인 screen을 설치한다. 설치 명령은 다음과 같다.

```
$ sudo apt-get install screen
```

다음 명령을 입력하여 screen을 실행하고, USB 직렬 포트를 9600 또는 115200 bps/8bit/no parity/1 stop 모드로 연결한다.

```
$ screen /dev/ttyACM0 115200
```

이제 다음과 같은 여러 AT 명령을 입력해 본다. 여기서 AT 명령을 입력할 때마다 CR_{Carriage Return}과 NL_{New Line} 글자를 추가해야 하므로 **Enter** 키를 친 다음에 **Ctrl**+**J** 키를 입력한다.

```
AT
OK
AT+GMR
AT veson:1.3.0.0(Jul 14 2016 18:54:01)
SDK version:2.0.0(5a875ba)
v1.0.0.3
Mar 13 2018 09:35:01
OK
AT+RST
OK
ö Éÿ"-FÖêÊ@*Zè
ready
AT+CWMODE?
+CWMODE:2
OK
AT+CWMODE=3
OK
```

참고로, ESP8266 모듈에서 많이 사용하는 AT 명령을 요약하면 표 11-1과 같다.

표 11-1 ESP8266 모듈의 대표적인 AT 명령어

명령	기능	응답
AT	명령 테스트	OK
AT+GMR	버전 확인	OK+ADDR: <MAC 주소>
AT+RESET	초기화 및 모듈 사용 준비	OK ... ready
AT+CWMODE=<모드>	와이파이 모드 설정 모드=1(스태이션), 2(AP), 3(둘 다)	+CWMODE: <모드> OK

(계속)



12

사물인터넷 서비스 활용

이번 장에서는 라즈베리 파이를 사용하여 다양한 방식의 사물인터넷 서비스를 활용하는 방법을 살펴본다. 구체적으로는 WebIOPi, The Thing System, OpenHAB과 같은 홈 자동화 서버를 설치하고 Xively, Thingspeak, Plotly와 같은 데이터 저장 서비스와 IFTTT, Pushbullet, Telegram과 같은 연동 서비스를 활용해 볼 것이다.

가장 심오한 기술은 사라지는 것이다.
그런 기술은 매일 생활의 직물을 짜서 결국 구별할 수 없게 된다.

- **마크 와이저**(Mark Weiser), 유비쿼터스 컴퓨팅의 아버지

12.1 사물인터넷 개요

위키피디아(Wikipedia)에 따르면 사물인터넷(Internet of Things)이란, 각종 사물에 컴퓨터 칩과 통신 기능을 내장하여 인터넷에 연결하는 기술을 의미한다. 약어로는 짧게 IoT라고 쓴다. 여기서 사물이란 가전제품, 모바일 장비, 웨어러블(wearable) 컴퓨터 등 다양한 임베디드(embedded) 시스템을 말한다.

사물인터넷에 연결되는 사물은 자신을 구별할 수 있는 유일한 아이디를 가져야 하고, 인터넷을 통한 통신 능력을 가져야 하며, 데이터 처리 능력이 있어야 한다. 또한, 필요하면 외부 환경으로부터의 데이터 취득을 위해 센서를 내장할 수 있어야 한다. 라즈베리 파이는 이러한 기능을 모두 가지고 있으므로 사물인터넷에 아주 적합한 장치라고 할 수 있다.

12.2 WebIOPi 웹 서버 연결

WebIOPi는 웹 브라우저를 통해 라즈베리 파이의 모든 GPIO에 연결된 센서를 관리하거나 제어할 수 있도록 설계된 소프트웨어다. 이것은 파이썬으로 가벼운 웹 서버를 구축하고, 자바스크립트로 구성된 웹 클라이언트 모듈을 통해 사용자가 웹 브라우저로 GPIO, 시리얼 등의 장치에 접근하여 제어할 수 있도록 한다. WebIOPi는 GPIO/SPI/I2C 버스에 연결된 아날로그 변환, 온도/습도/조도 센서 등의 30개 이상의 장치를 지원한다. 또한, 안드로이드와 파이 간 또는 파이끼리 통신할 수 있도록 파이썬 및 자바 클라이언트를 제공하고, IoT 프로토콜인 CoAP 서버로도 동작할 수 있다.

WebIOPi를 설치하는 방법은 간단하다. 먼저, WebIOPi 소프트웨어를 다운로드하고 압축을 푼다.

```
$ wget http://downloads.sourceforge.net/project/webiopi/WebIOPi-0.7.1.tar.gz
$ tar xvfz WebIOPi-0.7.1.tar.gz
$ cd WebIOPi-0.7.1
```

WebIOPi-0.7.1 버전에서는 라즈베리 파이 2/3/4에 대해 지원하지 않으므로, 라즈베리 파이 4에 설치하려면 다음과 같은 패치를 적용한다. 패치의 내용은 GPIO 메모리 할당 주소를 라즈베리 파이 4에 맞게 수정하는 부분과 40핀까지 표시하는 기능을 포함한다.

```
$ wget https://github.com/swkim01/webiopi-dht/raw/master/webiopi-0.7.1-rpi2.diff
$ patch -p1 < webiopi-0.7.1-rpi2.diff
```

이제 다음 명령을 실행하여 WebIOPi를 라즈베리 파이에 설치한다.

```
$ sudo ./setup.sh
```

설치 과정에서 마지막 부분에 Weaved를 설치하라는 물음이 나타날 것이다. Weaved 서버는 라즈베리 파이 서버가 공유기 환경의 서브넷에 연결되어 있더라도 가상 주소를 제공하여 외부에서 접속이 가능하도록 해준다. 이 기능은 다른 방식으로 설치하므로 지금은 'n'을 입력하자. 이 부분은 아래에서 다시 설명할 것이다.

설치가 완료되었으면 본격적으로 WebIOPi를 사용해 보자. WebIOPi를 시작하려면 systemd 서비스를 갱신해야 한다. 다음 명령을 입력하여 서비스를 갱신한다.

```
$ sudo systemctl daemon-reload
```

WebIOPi를 시작하려면 다음과 같은 명령을 실행한다.

```
$ sudo /etc/init.d/webiopi start
```

실행을 멈추려면 다음 명령을 실행한다.

```
$ sudo /etc/init.d/webiopi stop
```

부팅 시에 자동으로 실행하도록 하려면 다음과 같은 명령을 실행한다.

```
$ sudo update-rc.d webiopi defaults
```

WebIOPi를 활용하여 개발하는 도중에 실행되는 결과 또는 오류 메시지를 보려면 다음과 같이 WebIOPi 프로그램을 수동으로 실행하면 된다.

```
$ sudo webiopi -d -c /etc/webiopi/config
```

이제 웹 브라우저에서 `http://localhost:8000/` 주소로 접속해 보자. 사용자 이름과 패스워드를 묻는 창이 나타날 것이다. 사용자 이름과 패스워드를 입력하면 메인 홈페이지가 나타난다. `webiopi_password.py` 프로그램으로 사용자 이름과 패스워드를 지정하지 않으면 기본 사용자 이름은 'webiopi', 패스워드는 'raspberrypi'다.

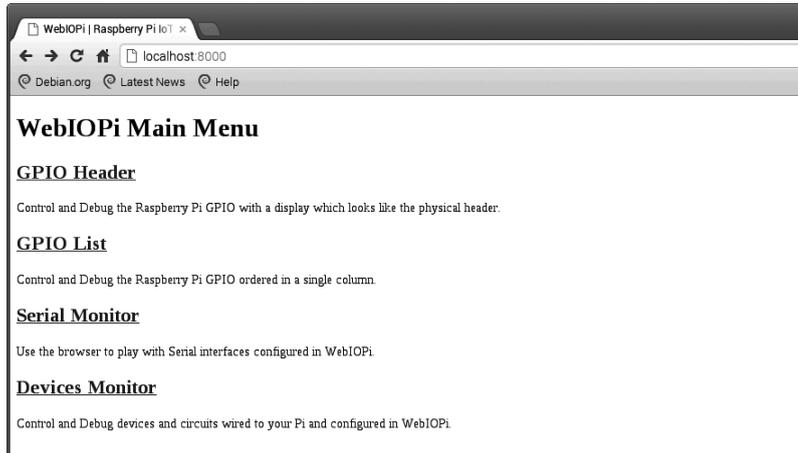


그림 12-1 WebIOPi 접속 화면

GPIO Header 탭을 누르면 다음과 같은 GPIO 핀에 대한 페이지가 나타난다. 여기서 GPIO 핀에 대한 버튼을 눌러 입출력 방향(IN/OUT), 레벨(HIGH/LOW)을 제어할 수 있다.

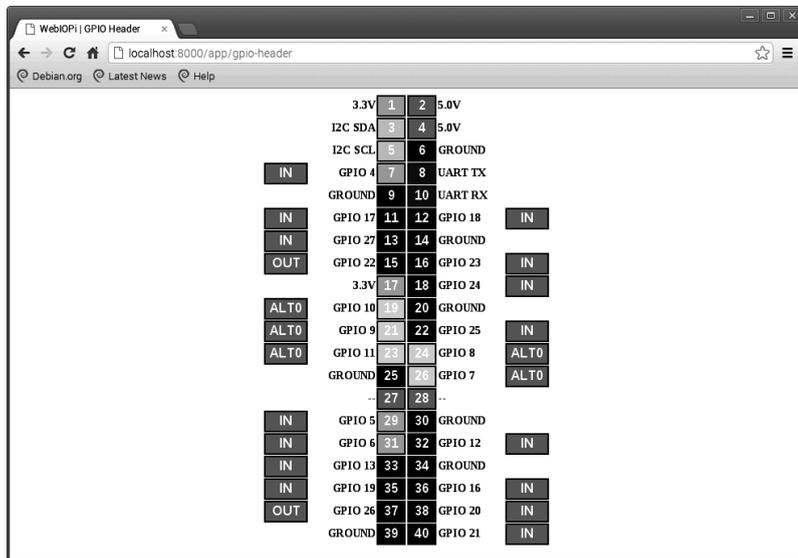


그림 12-2 WebIOPi를 통한 GPIO 제어 화면

이제 가상 사설망 인터넷 접속을 도와주는 `remote.it` 프로그램을 설치하는 방법을 알아보자.

먼저, 설치하기 전에 `www.remote.it`(예전: `www.weaved.com`) 사이트에서 계정을 만든다. 상단의 **Sign Up** 메뉴를 클릭하고, 로그인 이메일 계정과 패스워드를 입력하면 간단히 계정이 만들어질 것이다.

`connectd` 패키지를 설치한다.

```
$ sudo apt update
$ sudo apt install connectd
```

다음, 인스톨러 파일을 실행하면 `connectd` 서비스가 설치된다. 설치하는 도중에 로그인 **Sign In**(1번 메뉴)을 선택하고 `remote.it` 계정 이메일, 패스워드 및 설치할 장치 이름(예 `my_Pi_001`)을 입력해야 한다. 또, `WebIOPi`를 연결하기 위해 `remote.it` 서비스를 **애플리케이션**(1번 메뉴)과 **웹 서비스**(2번 메뉴)를 선택하고, 포트 번호를 디폴트(80)가 아닌 8000으로 입력한다. 서비스 이름은 `webiopi` 등으로 입력하고 끝낸다. 설치가 완료되면 자동으로 `remote.it` 연동 서비스가 실행된다.

```
$ sudo connectdinstaller
```

인터넷이 연결된 곳에서 스마트폰 등으로 웹 브라우저를 열어 `remote.it` 서버에 접속한다. 로그인 한 후에 **Devices** 메뉴를 선택하면 자신의 장치가 나타날 것이다. 장치 이름을 클릭하고 서비스 이름을 클릭하자. 조금 기다린 후에 `remote.it` 서버가 생성한 가상 주소로 접속해 보면 `WebIOPi` 홈페이지가 나타날 것이다.

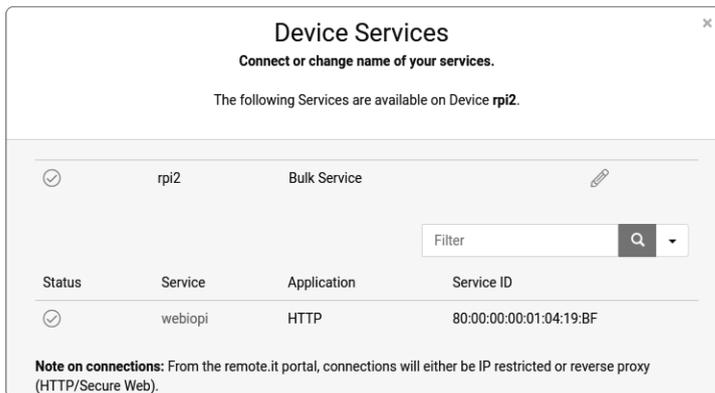


그림 12-3 `remote.it` 서버에 장치가 등록된 웹 페이지 화면

참고로, Connectd 서비스를 중지하고 싶으면 다음과 같은 명령을 입력하면 된다.

```
$ sudo connectd_stop_all
```

이번엔 WebIOPi의 구조에 대해 알아보자. WebIOPi는 기본적으로 HTTP 웹 서버의 기능을 하며, 라즈베리 파이에 연결된 GPIO 및 직렬 포트 등에 연결된 LED, 센서 등의 다양한 장치를 제어하도록 설계되었다. 내부적으로는 서버와 클라이언트 간에 HTTP REST API를 통해 메시지를 주고받지만, IoT 및 센서 네트워크 프로토콜인 CoAP 프로토콜도 지원한다. 또한, 개발자는 파이썬 스크립트를 통해 프로그래밍된 코드를 실행할 수 있는데, 매크로 기능을 활용하면 클라이언트가 임의의 파이썬 함수를 호출할 수도 있다.

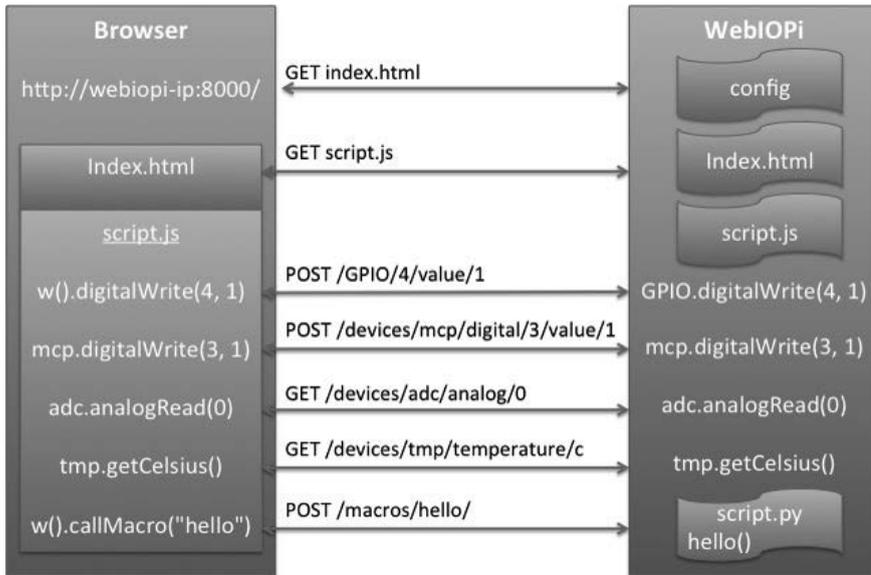


그림 12-4 WebIOPi 통신 순서

[출처] http://webiopi.trouch.com/Tutorial_Basis.html

WebIOPi는 `/etc/webiopi/config` 설정 파일에 지정된 정보를 참고하여 서버를 실행한다. 설정 파일의 형식은 섹션별로 '키=값'의 형태로 저장된다. 설정 가능한 섹션은 표 12-1과 같다.



13

음성 인식과 가상 비서

이번 장에서는 요즘 인기 있는 음성 인식(Voice Recognition)과 인공지능 가상 비서(Artificial Intelligent Virtual Assistant)에 대해 알아본다.

“안시블이 뭘니까, 쉘백?”

“아이디어죠.”

그는 살짝 미소지었다.

“그건 공간 상의 두 지점 사이에 시간 차 없이 통신하는 장치가 될 겁니다.”

- 어슐러 르 권(Ursula K. Le Guin), 《빼앗긴 자들(The Dispossessed)》(1974) 중에서

13.1 리눅스 사운드 시스템 및 음성 변환 테스트

먼저, 라즈베리 파이에서 음성 파일을 변환하거나 출력하기 위해서는 리눅스 시스템의 사운드 시스템에 대해 알 필요가 있다. 리눅스에서는 커널 2.4 버전까지는 OSS(Open Sound System)라는 사운드 드라이버 모듈을 사용하였고, 이후 2.6 버전부터 ALSA(Advanced Linux Sound Architecture) 사운드 드라이버 모듈을 지원하고 있다.

또한, 응용 프로그램에서 사운드 처리를 원활하게 할 수 있게 돕는 몇 가지 사운드 시스템 미들웨어들이 있다. libasound는 ALSA 시스템용 API를 제공하는 C 언어 라이브러리다. 한편, 수많은 응용 프로그램에서 활용하는 PortAudio라는 크로스플랫폼 오픈소스 오디오 입출력 라이브러리도 있다. 그 밖에 네트워크 기능을 지원하는 사운드 서버들도 있다. PulseAudio는 다양한 오디오 소스 입력과 출력을 지원하는 사운드 서버이며, Jack Audio Connection Kit는 오디오 녹음, 효과, 합성을 위한 저지연 오디오 서버인 jackd를 포함한다.

라즈베리 파이에 오디오 입력 및 출력 장치를 연결하고 테스트해 보자. 마이크는 보통 USB 웹캠에 내장된 마이크나 저렴하게 판매되는 USB 마이크 또는 리스피커(Respeaker) 4Mic*와 같은 마이크 실드 보드를 사용하면 된다.



그림 13-1 리스피커(Respeaker) 4Mic 실드 보드가 장착된 모습

리스피커 4Mic 장치를 사용한다면 다음과 같은 명령을 실행하여 리눅스 커널과 디바이스 드라이버를 설치해야 한다.

```
$sudo apt-get update
$sudo apt-get upgrade
$git clone https://github.com/respeaker/seeed-voicecard.git
$cd seeed-voicecard
$sudo ./install.sh
$reboot
```

* http://wiki.seeedstudio.com/ReSpeaker_4_Mic_Array_for_Raspberry_Pi/

스피커로는 PC 스피커를 3.5mm 잭에 연결하거나 블루투스 스피커를 블루투스로 연결하면 된다. 라즈베리 파이에서 3.5mm 잭을 오디오 출력 장치로 설정하려면 다음 명령으로 설정 프로그램을 실행한 후에, 7 Advanced Options ⇨ A4 Audio ⇨ 1 Force 3.5mm ('headphone') jack을 선택하고, 확인하고 끝내면 된다.

```
$ sudo raspi-config
```

라즈베리 파이에서 오디오 입력 장치를 확인하려면 다음과 같은 명령을 입력하면 된다.

```
$ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Ddvice [USB PnP Sound Device], device:0  USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

오디오 출력 장치를 확인하려면 다음과 같은 명령을 입력한다.

```
$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
...
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
...
```

라즈베리 파이는 기본적으로 3.5mm 아날로그 잭과 HDMI인 2개의 오디오 출력 장치를 가진다. 장치를 선택하려면 다음 명령을 입력하면 된다. 여기서 n 값으로 0은 자동(오토), 1은 3.5mm 아날로그 잭, 2는 HDMI를 지정한다.

```
$ amixer cset numid=3 <n>
```

참고로, 오디오 입출력 장치의 볼륨을 조절하려면 amixer를 사용해도 되지만 alsamixer를 사용해도 된다. Pulseaudio 서버를 쓰고 있다면 pavucontrol이라는 GUI 프로그램도 사용할 수 있다.

```
$ alsamixer
```

또한, 간단한 스피커 테스트는 다음 명령을 입력하면 된다.

```
$ speaker-test -t wav
```

다음으로 음성 인식, 합성 등을 프로그래밍하기 위해 기본적인 음성 녹음 기능을 테스트해 보자. 먼저, 마이크를 통해 음성을 녹음하는 기능을 테스트해 본다. 다음과 같은 명령을 입력하면 음성이 녹음된 `temp.wav` 파일이 생성된다. 여기서 `-d` 옵션은 녹음 시간(초)을 나타낸다.

```
$ arecord -d 3 temp.wav
```

만약 `'arecord: main:722: audio open error: 그런 파일이나 디렉터리가 없습니다'`와 같은 오류가 나타나면 다음과 같은 `.asoundrc` 설정 파일을 만들어 주면 된다. 예를 들면, 설정 파일은 재생 `playback`을 위한 스피커는 3.5mm 잭(`card 0`)을 사용하고, 녹음 `record`을 위한 마이크는 USB 마이크 (`card 1`)를 사용하도록 지정한다. 그리고 마이크에 따라 녹음 파일 형식을 16비트로 지정하지 않으면 잡음이 생길 수도 있다는 점을 주의해야 한다.

코드 13-1 음성 입출력 장치를 설정하는 `~/asoundrc` 파일

```
pcm.!default {
    type asym
    playback.pcm {
        type plug
        slave.pcm "hw:0,0"
    }
    capture.pcm {
        type plug
        slave.pcm "hw:1,0"
        slave.format "S16_LE"
    }
}
```

이제 녹음된 `wav` 파일을 재생해 보자. 다음 명령을 입력하고 스피커로부터 음성이 나오는지 확인한다.

```
$ aplay temp.wav
```

13.2 음성 인식, 합성 및 가상 비서 시스템 개요

음성을 듣고 음성으로 응답해 주는 시스템은 보통 다음과 같은 구조로 이루어져 있다. 먼저, 사람의 음성을 마이크로 입력받아 파일로 녹음하고, 반대로 파일을 스피커로 출력할 수 있어야 한다. 이런 기능은 라즈베리 파이에서는 주로 리눅스 커널의 ALSA 프레임워크와 PulseAudio, PortAudio와 같은 음성 처리 라이브러리들이 담당한다. 다음 단계에서는 음성 파일을 텍스트로 변환하는 음성 인식, 즉 STT_{Speech-To-Text} 기능을 통해 내부 시스템으로 전달된다. 이 텍스트는 인공지능 비서 또는 챗봇 등과 같은 가상 비서 프로그램을 통해 IoT 장치를 자동으로 제어하거나 응답 텍스트를 TTS_{Text-To-Speech}로 보낸다. TTS는 텍스트를 음성 파일로 변환하는 음성 합성 기능이며 최종적으로 스피커로 음성을 출력시키게 된다.

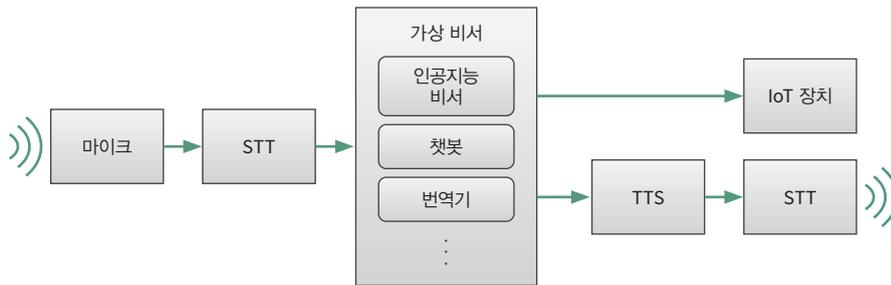


그림 13-2 음성 인식, 합성 및 가상 비서 시스템

실습에 사용된 소스 코드는 다음 명령으로 다운로드할 수 있다.

```
$ git clone https://github.com/swkim01/virtual-assistant
```

13.3 음성 인식

음성 인식_{Speech Recognition}이란 컴퓨터를 사용하여 음성을 텍스트로 인식하고 번역하는 기술을 말하며 다른 말로는 STT_{Speech To Text}라고 한다. 음성 인식 시스템의 구조는 다음과 같다. 마이크를 통해 입력된 음성 신호는 PCM_{Pulse Code Modulation} 형식의 데이터로 저장되고 이 음향 데이터를 필터링한 특성 벡터를 얻는다. 음향 분석 단계에서는 학습과 통계학적인 분석으로 만들어진 음향 모델_{Acoustic Model}을 통해 이 특성 벡터를 음성의 기본 단위인 음소_{phoneme}로 분류한다. 다음으로, 검색 단계에서는 개별 단어들을 모은 사전과 이들 개별 단어들과 문법에 따른 통계적인 확률을 학습한 언어 모델을 통해 가장 가능성이 높은 단어들과 문장을 찾아낸다. 참고

로, 음향 모델과 언어 모델에 많이 쓰이는 기술로는 HMM^{Hidden Markov Model}, DTW^{Dynamic Time Warping}, NN^{Neural Network} 등이 있다. 또한, FST^{Finite State Transducer}는 음향 모델과 언어 모델을 통계적으로 결합한다.

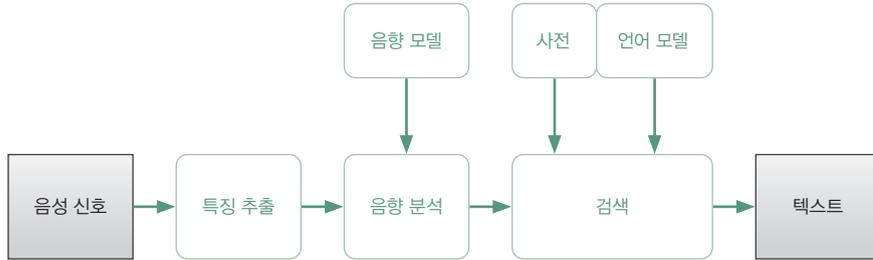


그림 13-3 음성 인식 시스템

대표적인 음성 인식 프로그램으로는 Sphinx/Pocket Sphinx, Julius, HTK, Kaldi, MS Windows Speech API(cloud), Google Speech API(cloud), Wit.ai 등이 있다.

이제 몇 가지 음성 인식 소프트웨어를 설치하고 테스트해 보자.

13.3.1 포켓 스피нк스

먼저, 포켓 스피нк스^{Pocket Sphinx}는 CMU에서 개발한 오픈소스 음성 인식 소프트웨어인 스피нк스^{Sphinx}의 경량 버전이다. 스피нк스는 라이브러리, 음향 모델 및 샘플 프로그램들로 이루어져 있으며, 추가적으로 음향 모델 훈련 프로그램, 컴파일된 언어 모델과 cmudict 발음 사전 및 어휘집을 포함한다.

포켓 스피нк스를 사용하기 위해서는 음향 모델, 언어 모델 및 단어 사전 또는 문법 파일이 필요하다. 음향 모델은 HMM(은닉 마르코프 모델)로 구성하는데, 직접 녹음된 음성 파일을 이용하여 훈련하고 빌드할 수도 있고 이미 빌드된 기존의 모델 파일들을 활용할 수도 있다. 기존의 음향 모델로는 영어, 프랑스어 등에 대한 음향 모델 파일이 있으며 다음 사이트에서 다운로드할 수 있다.

```
https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/
```

직접 HMM 모델을 위해 음성 파일을 훈련하고 빌드하는 과정은 까다로우므로 여기서는 이미 빌드된 음향 모델을 활용하기로 한다. 다음 명령을 입력하여 미국 영어에 대한 음향 모델을 다운로드한다.