



최종보고서

사용자 지향 IoT 쇼핑카트 Line Tracing 시스템 (IS_LTS)

2019. 06. 13

한국외국어대학교

정보통신공학과

2팀(Shopporter)

문서 정보

구 분	소 속	성 명	날 짜	서 명
작성자	한국외국어대학교	박영준	2019. 06. 13	
	한국외국어대학교	송진호	2019. 06. 13	
	한국외국어대학교	김현석	2019. 06. 13	
	한국외국어대학교	김효민	2019. 06. 13	
검토자	한국외국어대학교	박영준	2019. 06. 13	
	한국외국어대학교	송진호	2019. 06. 13	
	한국외국어대학교	김현석	2019. 06. 13	
	한국외국어대학교	김효민	2019. 06. 13	
사용자				
승인자	한국외국어대학교	홍진표		



머리말

본 문서는 사용자 지향 IoT 카트에 대한 서비스와 시스템 구성을 설명하고 있다. Line Tracing 모듈과 Ultra Sonic 센서를 이용해 Line을 따라가며 장애물 발견 시 급정거하는 시스템을 Cart에 적용하여 User(사용자)는 안전하고 편리한 쇼핑을 할 수 있다. 또한 Data base를 통해 지난 쇼핑 목록을 확인할 수 있다. 관리자는 사용자의 쇼핑목록과 인기있는 상품, 없는 상품, 상품현황 등을 알 수 있다.



목 차

목 차	4
1. 개요	6
1.1 목적	6
1.2 서비스 시나리오	7
1.3 용어 및 약어	7
2. 기능 소개	8
2.1 카트 기능 개요	8
2.1.1 자율주행	8
2.1.2 급정거 기능	8
2.1.3 구매물품 확인	8
2.2 Application 기능 소개	9
3. 시스템 구성도	10
3.1 전체 시스템 구성도	10
3.2 소프트웨어	11
3.2.1 Python	11
3.2.2 Android Studio	11
3.2.3 Arduino IDE	11
3.2.4 Amazon EC2	12
3.2.5 WSGI	12
3.2.6 SQLite	12
3.3 하드웨어	13
3.3.1 Arduino	13
3.3.2 Raspberry pi	13
3.3.3 Smart Phone	14
4. 네트워크 구성과 통신 방식	16
4.1 Arduino-RaspberryPi간 Serial 통신	16
4.2 RaspberryPi-Web Server간 Socket 통신	16
4.3 Web Server-Android간 Socket 통신	17
5. 시스템 상세 설계	18
5.1 상세 설계도	18
5.2 Sensor, Actuator 회로도	18
5.2.1 Rfid	18

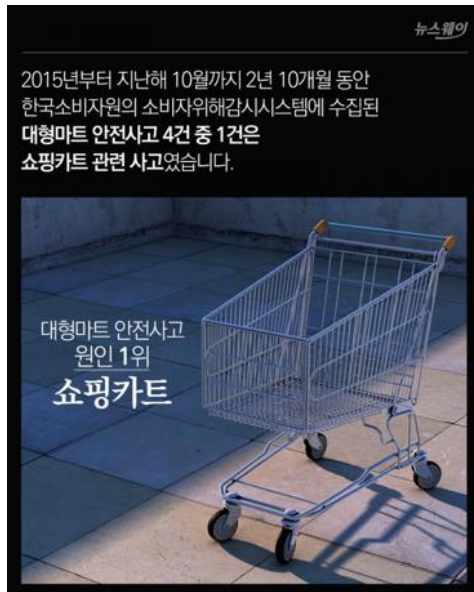
최종보고서: 사용자 지향 IoT 쇼핑카트 Line Tracing 시스템 (User oriented IoT Shopping Cart Line Tracing System)

5.2.2 Motor Control Module & DC Motor	18
5.2.3 UltraSonic Sensor.....	19
5.3 IS Server(Web & Database)	19
6. 적용방안 및 기대효과 (SWOT 분석).....	19
7. 향후 개발 사항	20
7.1 결제 시스템 추가.....	20
7.2 쇼핑 안내 기능 추가.....	20
7.3 Beacon	20
8. 프로젝트 세부추진계획 및 세부일정	21
9. 팀원 담당업무	21
10. Source Code	21

1. 개요

본 장에서는 IoT 쇼핑 카트(Shopperter)의 목적과 사용된 핵심기술의 약어를 소개합니다.

1.1 목적



[대형마트 시설별 안전사고 현황]				
(단위 : 건, %)				
구분	2015년	2016년	2017.10월	합계
쇼핑카트	80 (26.8)	47 (26.8)	39 (21.8)	166 (25.5)
무빙워크(에스컬레이터)	90 (30.2)	41 (23.4)	28 (15.6)	159 (24.4)
바닥 및 계단	42 (14.1)	20 (11.4)	30 (16.8)	92 (14.1)
상중 및 진열대	12 (4.0)	22 (12.6)	12 (6.7)	46 (7.1)
문	18 (6.0)	8 (4.6)	19 (10.6)	45 (6.9)
주차장	20 (6.7)	4 (2.3)	11 (6.1)	35 (5.4)
유아용 놀이시설	7 (2.4)	8 (4.6)	12 (6.7)	27 (4.1)
편의시설 및 의자	7 (2.4)	7 (4.0)	5 (2.8)	19 (2.9)
엘리베이터	3 (1.0)	4 (2.3)	5 (2.8)	12 (1.8)
유모차 및 휠체어	6 (2.0)	1 (0.6)	1 (0.6)	8 (1.2)
지게차 및 적재물	3 (1.0)	1 (0.6)	2 (1.1)	6 (0.9)
기타 시설물*	10 (3.4)	12 (6.8)	15 (8.4)	37 (5.7)
합계	298 (100.0)	175 (100.0)	179 (100.0)	652 (100.0)

* 셀프포장대, 테이프커팅기, 벽재, 시더리, 기타 자재류 등

오늘 날 대형마트에서 사용하고 있는 쇼핑 카트는 단순히 사물을 옮기기 위해 바구니 형태의 철제 바구니에 바퀴를 달아 사람이 직접 움직여야 하는 수동적인 카트입니다. 이러한 수동성으로 인해 쇼핑 카트를 이용하는 환경이나 이용자의 주의여부에 따라 안전사고도 종종 발생합니다. 실제로 대형마트에서 일어나는 대부분의 사고는 어린이들과 쇼핑 카트의 충돌로 인한 사고입니다. 이러한 현황에서 저희는 쇼핑 카트가 자동으로 움직이며 쇼핑에 도움되는 방법을 궁리해보았습니다. 저희는 이러한 과정 속에서 하드웨어적인 개선방법과 소프트웨어적인 개선방법을 도출했습니다. 하드웨어적인 개선방안은 카트에 자율주행 기능을 넣어 사람이 직접 카트를 운행함으로써 생기는 사고의 가능성을 줄이는 것입니다. 소프트웨어적인 개선방안은 현 구조는 원래 쇼핑이 다 끝난 뒤 Cashier에서 계산하는데 이 구조를 마트 이용객이 구매 물품을 카트에 담은 그 장소가 Cashier와 같은 기능을 할 수 있도록 만들려고 합니다. 즉 이용객은 Cashier에서 자신의 차례를 기다린 뒤 구매물품을 수동적으로 확인하며 계산할 필요없이 Shopporter와 연계된 앱을 통해 구매물품을 확인하고 금액확인이 가능하게 하려합니다. 이러한 개선 방안들을 통해 Shopporter은 간단하지만 쇼핑에 도움되는 여러 강력한 기능들을 가진 IoT 쇼핑 카트입니다.



1.2 서비스 시나리오

시나리오1:

주부 K씨는 항상 오전에 집안일을 마친 뒤 오후에 장을 보러 가는데 장을 보다 가족들이 좋아하는 음식을 하나 둘 챙기다 보면 항상 예산이 초과가 되어 걱정이다. 장을 다 본 뒤 계산대를 가니 자신처럼 장을 보러 온 주부들이 계산을 하려고 줄을 서있는데 다들 계산할 물품이 많아서 생각보다 오랜 시간을 기다려야 해서 불편함을 느낀다.

시나리오1에서의 Shopporter의 활용:

쇼핑을 하면서 App을 통해 결제 금액을 확인 할 수 있어 추가 지출을 막을 수 있으며 App에서 총금액의 확인이 가능 하기에 Cashier에서 일일이 물품을 확인하며 결제금액을 도출 할 필요가 없어져 계산대의 빠른 손님 받기가 가능하다.

시나리오2:

아이들과 함께 장보는 것을 좋아하는 주부 L씨는 오늘도 아이들을 데리고 장을 보러 왔지만 아직 걷지 못하는 아기와 호기심이 많아 눈만 떴면 사라지는 6살 아이가 있어 마음 편히 장보는게 불가능하다. 심지어 6살 아이는 저번 달 마트에서 코너길에서 갑자기 튀어나온 카트와 부딪혀 이마에 7바늘을 꿰매는 상처를 입어 더 조심스럽다.

시나리오2에서의 Shopporter의 활용:

Shopporter은 자율주행 기능을 탑재하고 있어 카트를 직접 끌며 주의를 기울일 필요가 없어 쇼핑에 더 집중 할 수 있으며 같이 온 아이들의 케어에 더 신경 쓸 수 있을 것이다. 초음파 센서를 통한 전방 장애물과의 거리를 상시 측정하므로 충돌 가능성이 있는 거리에 물체가 있을 시 급정거를 하므로 설령 충돌이 일어나더라도 카트의 속력이 더해져 더 크게 다치는 일이 없어질 것이다.

1.3 용어 및 약어

용어 및 약어	풀이	비고
AWS	Amazon Web Service – 클라우드에서는 컴퓨팅 파워, 스토리지 옵션, 네트워킹 및 데이터 베이스 같은 다양한 인프라 서비스를 제공	
RFID	Radio-Frequency Identification 주파수를 이용해 ID를 식별하는 방식으로 일명 전자태그로 불린다.	
Shopporter	본 프로젝트의 이름이자 만든 Device의 이름으로 Shopping 과 Supporter 가 합쳐서 쇼핑을 도와주는 기기임을 의미한다.	
SP	팀 이름이자 스마트 카트의 이름인 Shopporter 의 약자로 2Team만의 고유 시스템을 정의할 때 쓰인다.	

2. 기능 소개

본 장에서는 Shopporter의 기능소개와 Shopporter가 가진 application 기능을 소개합니다.

2.1 카트 기능 개요

2.1.1 Line Tracing 기능

아두이노에서 라인 트레이서 모듈을 통해 얻은 값을 통한 DC Motor 의 제어로 정해진 길을 따라 이동하는 라인 트레이싱 기능을 탑재했습니다.

2.1.2 급정거 기능

라즈베리파이의 초음파 센서를 통해 카트 전방의 장애물과의 거리를 읽고 충돌 위험이 있는 거리감지시 아두이노의 DC Motor에 신호를 주어 급정거.

2.1.3 구매물품 확인

RFID 모듈을 통해 확인한 물품의 정보를 웹서버로 보내고 앱을 통해 총 구매물품 및 가격의 확인이 가능합니다.

2.2 차별성

	기존 카트	ShopPorter
충돌 사고 위험	높음	매우 낮음
분실 위험	높음	매우 낮음



쇼핑에 필요한
정보 제공



고객 관리
시스템



빅데이터
수집 가능

2.2 Application 기능 소개



로그인 뷰는 사용자의 아이디와 비밀번호를 확인하고 어플에 로그인을 할 수 있는 뷰이다.

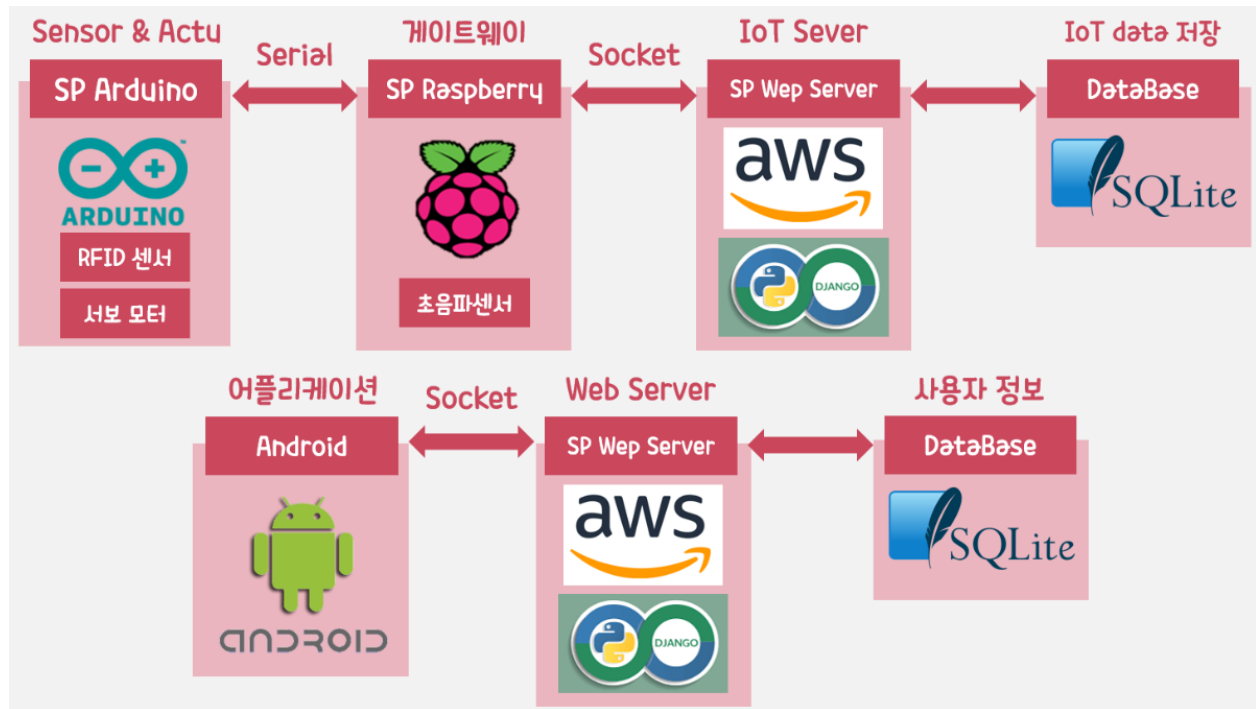
아이디와 비밀번호가 맞으면 메뉴 뷰로 이동하여 쇼핑을 시작할 지 로그아웃을 할지 고르는 메뉴 뷰가 나온다. 메뉴 뷰에서 쇼핑시작 버튼을 누르면 카트 뷰로 이동하게 된다.

카트 뷰에서는 사용자가 태그에 찍은 RFID값을 실시간으로 출력하여 현재 카트 안에 있는 상품이 무엇인지 알 수 있고 쇼핑 완료 버튼을 누르면 웹서버로 쇼핑 기록을 보낸다.

3. 시스템 구성도

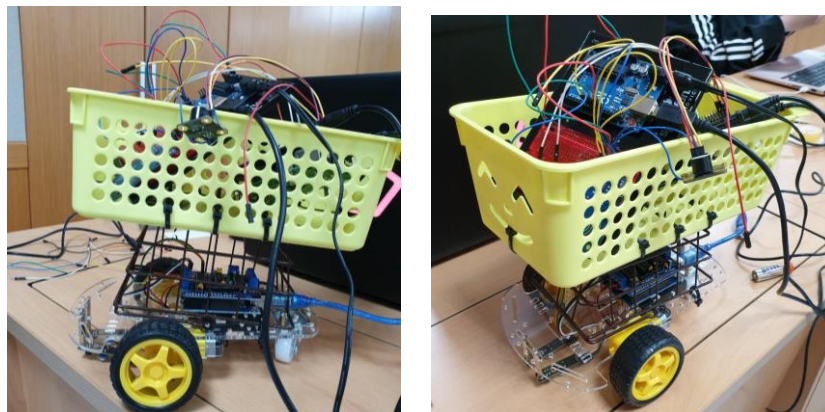
본 장에서는 전체 시스템 구성, 세부 시스템 구성 그리고 시스템을 구성하는 개별 소프트웨어, 하드웨어, 센서 기술한다.

3.1 전체 시스템 구성도

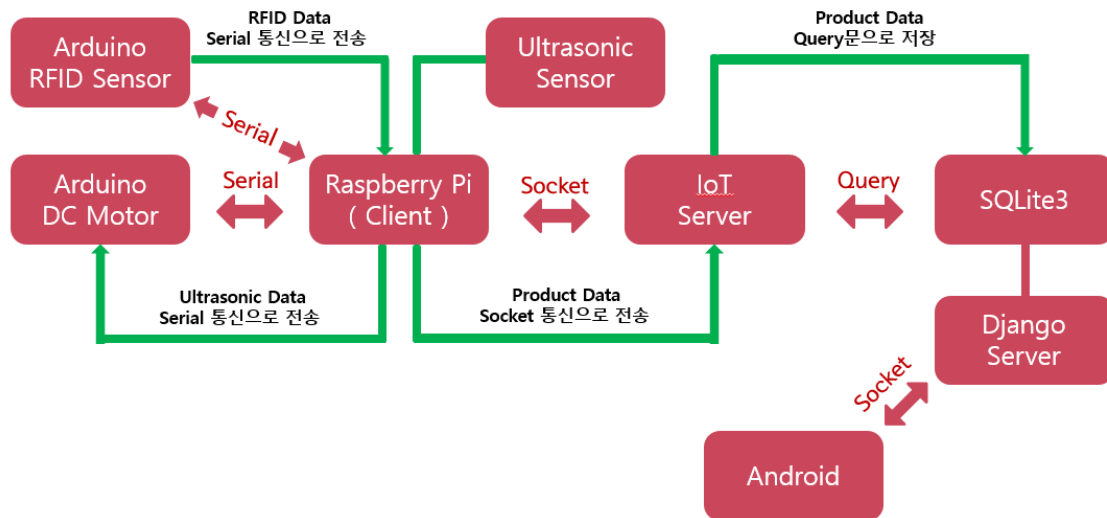


<시스템 구성도>

Arduino 의 DC Motor가 라인 트레이싱 시스템을 통해 자율 주행을 하다 Raspberry Pi에서 읽어온 UltraSonic 값이 일정 수치 이하일 시 Serial 통신을 통해 신호를 보내주어 급정거를 하게 한다. Arduino 에서 RFID 센서로 값을 받아 데이터를 Serial 통신으로 Raspberry Pi로 보내고 이 값을 Socket 통신으로 IoT Server로 전송해서 Query문을 통해 Django 에 내장된 SQLite을 통해 저장하여 DB를 만들고 DB에 저장된 데이터를 Android 와 Socket 통신합니다.



< Shopperter 완성품 >



<데이터 흐름도>

3.2 소프트웨어

3.2.1 Python (with PyCharm and Raspberry Pi)



타이핑 범용 프로그래밍 언어로, 펄 및 루비와 자주 비교된다. 다양한 플랫폼에서 쓸 수 있고, 라이브러리(모듈)가 풍부하여, 대학을 비롯한 여러 교육 기관, 연구 기관 및 산업계에서 이용이 증가하고 있다. 또 파이썬은 순수한 프로그램 언어로서의 기능 외에도 다른 언어로 쓰인 모듈들을 연결하는 풀언어(glue language)로써 자주 이용된다. 실제 파이썬은 많은 상용 응용 프로그램에서 스크립트 언어로 채용되고 있다. 도움말 문서도 정리가 잘 되어 있으며, 유니코드 문자열을 지원해서 다양한 언어의 문자 처리에도 능하다. 기본적으로 인터프리터 위에서 실행될 것을 염두에 두고 설계된 언어이다.

3.2.2 Android Studio



구글이 안드로이드 앱 개발을 위해 JetBrains 사의 IntelliJ IDEA를 기반으로 만든 통합 개발 환경이다. 2014년 10월부터 이클립스를 대체해 안드로이드의 공식 IDE가 되었다. 무료로 제공되고 있으며 Windows, macOS, 리눅스를 지원한다. 2017년 5월부터 안드로이드의 공식 프로그래밍 언어로 채택된 Kotlin도 사용 가능하다.

3.2.3 Arduino IDE



아두이노 통합개발환경(Arduino IDE)은 편집기, 컴파일러, 업로더 등이 합쳐진 소프트웨어 환경이다. '아두이노 소프트웨어'라고도 불린다. 이와 더불어 기타 개발에 필요한 각종 옵션 및 라이브러리 관리를 할 수 있다. 아두이노 프로그램 실행 시, 개인용 컴퓨터와 시리얼 통신을 할 수 있는 가상 시리얼 모니터를 제공한다. 보통 USB를 통해 업로드를 하므로 아두이노 보드는 USB를 UART 통신으로 바꾸는 방법이 제공되고, MCU가 실행할 때는 이 UART 통신을 이용하여 필요한 통신을 할 수 있다. 이렇게 되려면 아두이노의 MCU는 부트로더가 올라가 있어야 한다.

3.2.4 Django



Django는 파이썬으로 만들어진 무료 오픈소스 웹 애플리케이션 프레임워크(web application framework)로 쉽고 빠르게 웹사이트를 개발할 수 있도록 돕는 구성요소로 이루어진 웹 프레임워크 고도의 데이터베이스 기반 웹사이트를 작성하는 데 있어서 수고를 더는 것이 장고의 주된 목표이다. 장고는 컴포넌트의 재사용성(reusability)과 플러그인화 가능성(pluggability), 빠른 개발 등을 강조하고 있다.

3.2.4 Amazon EC2



Amazon Elastic Compute Cloud EC2는 아마존닷컴의 클라우드 컴퓨팅 플랫폼 아마존 웹 서비스의 중앙부를 이루며, 사용자가 가상 컴퓨터를 임대 받아 그 위에 자신만의 컴퓨터 애플리케이션들을 실행할 수 있게 한다. EC2는 사용자가 아마존 머신 이미지(AMI)로 부팅하여 아마존이 "인스턴스"라 부르는 가상 머신을 원하는 소프트웨어를 포함하여 구성할 수 있게 하는 웹 서비스를 제공함으로써 스케일링이 가능한 애플리케이션 배치를 장려한다. 사용자는 필요하면 서버 인스턴스를 만들고 시작하고 종료할 수 있으며, 실행 중인 서버에 대해 시간 당 지불하므로 일래스틱이라는 용어를 사용하게 된다. EC2는 사용자에게 레이턴시 최적화와 높은 수준의 다중화를 허용하는 지리학적 인스턴스 위치에 대한 통제 기능을 제공한다

3.2.5 WSGI

웹 서버 게이트웨이 인터페이스(WSGI, Web Server Gateway Interface)는 웹서버와 웹 애플리케이션의 인터페이스를 위한 파이썬 프레임워크다. 기존의 파이썬 웹 애플리케이션 프레임워크는 웹서버를 선택하는데 있어서 제약이 있었다. 보통 CGI, FastCGI, mod_python 과 같은 커스텀API 중에 하나만 사용할 수 있도록 디자인 되었는데, WSGI는 그에 반하여 low-level로 만들어져서 웹서버와 웹 애플리케이션, 프레임워크간의 벽을 허물었다. WSGI는 서버와 게이트웨이, 애플리케이션과 프레임워크 양단으로 나뉘어져 있다. WSGI 리퀘스트를 처리하려면, 서버단에서 환경정보와 콜백함수를 애플리케이션단에 제공해야 한다. 애플리케이션은 그 요청을 처리하고 미리 제공된 콜백함수를 통해 서버단에 응답한다.

3.2.6 SQLite

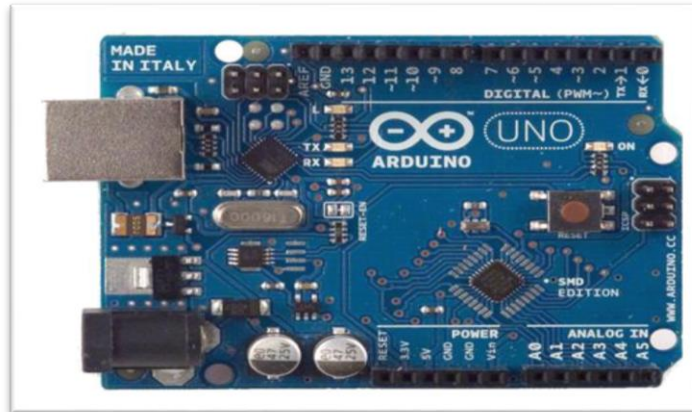


SQLite는 MySQL나 PostgreSQL과 같은 데이터베이스 관리 시스템이지만, 서버가 아니라 응용 프로그램에 넣어 사용하는 비교적 가벼운 데이터베이스이다.

일반적인 RDBMS에 비해 대규모 작업에는 적합하지 않지만, 중소 규모라면 속도에 손색이 없다. 또 API는 단순히 라이브러리를 호출하는 것만 있으며, 데이터를 저장하는 데 하나의 파일만을 사용하는 것이 특징이다. 구글 안드로이드 운영 체제에 기본 탑재된 데이터베이스이기도 하다.

3.3 하드웨어

3.3.1 Arduino



아두이노는 오픈 소스를 기반으로 한 단일 보드 마이크로컨트롤러로 완성된 보드(상품)와 관련 개발 도구 및 환경을 말한다. 아두이노는 다수의 스위치나 센서로부터 값을 받아들여, LED나 모터와 같은 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 물건을 만들어 낼 수 있다.

이번 프로젝트에서는 카트의 구동을 위한 Dc Motor의 제어, Rfid 모듈의 활용을 위해 쓰였다.

3.3.2 Raspberry pi



라즈베리 파이는 영국 잉글랜드의 라즈베리 파이 재단이 개발한 신용카드 크기의 싱글 보드 컴퓨터이다. 라즈베리 파이는 그래픽 성능이 뛰어나면서도 가격이 저렴한 것이 특징이다. 라즈베리파이는 키보드, 모니터 등을 뺀 단일 보드만으로 구성됐다. 다시 말해 컴퓨터 일부 부품인 셈이다. 누군가 보면 미완성으로 그친 제품일 수 있지만, 프로그래머에게는 나만의 컴퓨터를 만들 수 있는 좋은 재료가 된다. 컴퓨터는 이미 보드에 정해진 기능이 있고, 확장할 수 없다. 하지만 라즈베리파이는 사용자가 원하는 대로 기능을 확장하거나 용도를 변경할 수 있다. 모니터와 마우스를 연결하고 그 안에 운영체제를 설치하면 라즈베리파이는 일반 PC가 되고, 그 안에서 문서를 작성하거나 웹 브라우저를 실행할 수 있다. 카메라 모듈을 연결하면 디지털 카메라가 된다. 각종 센서 모듈을

연결하면 사물인터넷 제품을 만들 수 있고, 게임기 버튼과 디스플레이를 결합하면 휴대용 게임기도 만들 수 있다.

이번 프로젝트에서는 급 정지를 위한 초음파 센서 값을 받는 것 과 아두이노와의 Serial 통신을 통해 센서들의 값을 받고 이를 Socket 통신을 통해 Web Server에 전달하는 GateWay 역할을 한다.

3.3.3 Smart Phone



휴대전화에 인터넷 통신과 정보검색 등 컴퓨터 지원 기능을 추가한 지능형 단말기로서 사용자가 원하는 애플리케이션을 설치할 수 있는 것이 특징이다. 기능면에서 휴대전화와 컴퓨터가 결합된 형태이다.

이번 프로젝트에서는 App 을 설치하여 이용객들이 구매 물품, 가격 확인 기능을 Shopporter을 통해 활용할 수 있도록 해주는 역할을 한다.

4. 네트워크 구성과 통신 방식

4.1 Arduino – RaspberryPi 간 Serial 통신



2개의 Arduino 가 각각 USB 케이블로 Raspberry Pi 에 연결되어 있는데 Rfid 모듈을 사용하는 Arduino는 Rfid 태그가 찍힐 때마다 지정해 놓은 신호를 Serial 통신으로 Raspberry Pi 에 보내준다. 이 신호를 받은 Raspberry Pi 는 신호에 따른 코드가 돌아간다.

Raspberry Pi에서 초음파 센서로부터 일정 거리 이하라는 데이터를 받을 시 Serial 통신으로 Arduino에 지정한 신호를 보내주어 Arduino로 하여금 급정거를 할 수 있도록 한다.

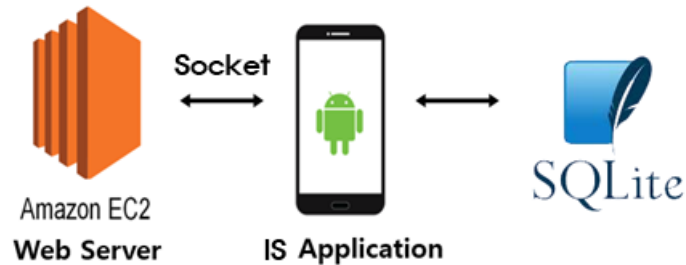
4.2 RaspberryPi – Web Server간 Socket 통신



IS Gateway는 내장된 무선랜을 이용하여 wifi에 접속한다. 무선 네트워크 환경이 조성된 후, AWS EC2의 가상 컴퓨팅 인스턴스의 IP를 목적지로 하여 IoT Socket통신 환경을 구축한다.

IoT Socket통신으로 주고 받는 데이터는 Arduino와 RaspberryPi의 Serial통신을 통해 전송된 RFID 값을 Django의 SQLite에 database를 저장한다.

4.3 Web Server – Android 간 Socket



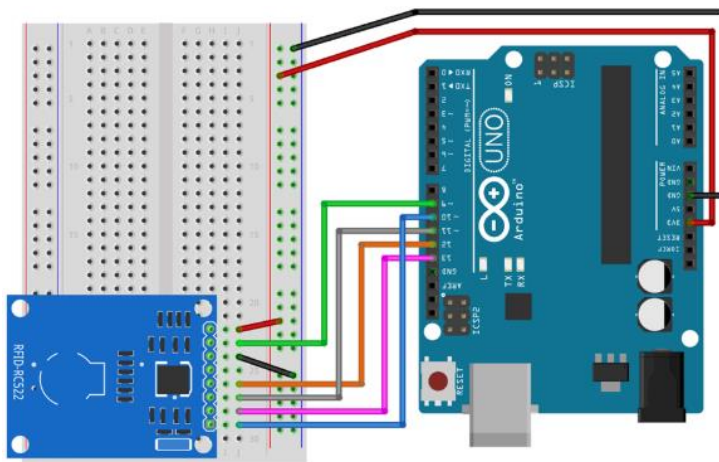
AWS에서 제공하는 EC2의 가상 컴퓨팅 인스턴스 IP를 Server Host IP Server로 지정하고 안드로이드 Application을 Client로 Socket통신 환경을 구축한다. Socket통신을 기반으로 사용자가 구매하고자 하는 쇼핑 목록 데이터를 실시간으로 전송해서 안드로이드 Application에 올려주어 쇼핑 목록을 확인할 수 있다. 추가된 쇼핑 목록은 안드로이드 자체 DB인 SQLite에 저장된다.

5. 시스템 상세 설계

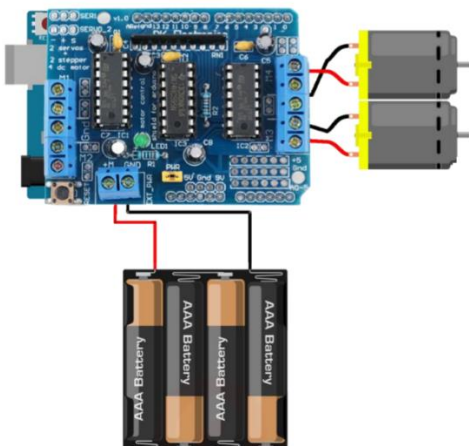
5.1 상세 설계도

5.2 Sensor, Actuator 회로도

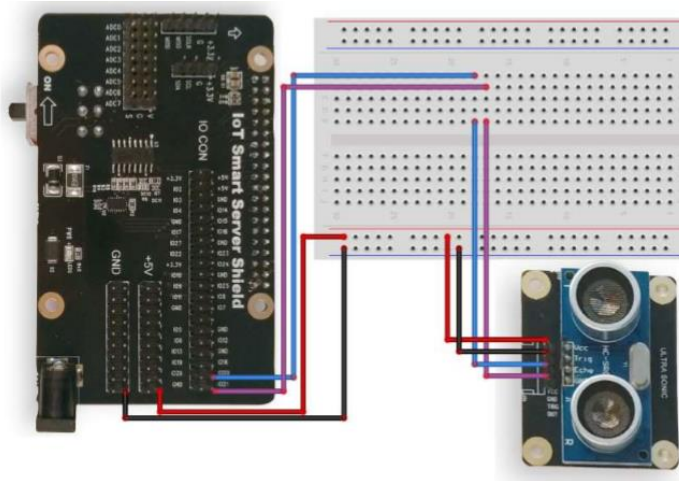
5.2.1 Rfid (Arduino)



5.2.2 Motor Control Module & DC Motor (Arduino)



5.2.3 UltraSonic Sensor (Raspberry Pi)



5.3 P Server(Web & Database)

IS Server는 IS SensorNode에서 사용자가 원하는 물건을 RFID 태그를 통해 들어온 데이터가 Django의 SQLite DB와, 안드로이드의 SQLite DB에 각각 저장된다. Django로 만들어진 웹페이지에서 AWS EC2 서버에 올라간 admin으로 접속하면 저장된 database 목록을 확인할 수 있고 수정도 할 수 있다. Server와 Raspberry Client는 RequestHandler와 response로 data를 주고 받을 수 있으며 안드로이드 Application과 Server도 동시에 통신하기 위해 thread를 생성하였고 Python 다 대 1 Socket 통신을 구현하였다.

6. 적용방안 및 기대효과(SWOT 분석)

S: 현재 활용 되고 있는 쇼핑 카트와 비교하여 높은 안정성과 다양한 기능을 가지고 있다.

W: 기존의 카트를 모두 교체해야 하고 기존카트 대비 비싸기에 구매충인 중소마트 사장들이 부담을 느낄 수 있다.

O: 대형마트에서 대기업과 연계하여 1대당 1억원 상당의 스마트카트를 운영할 계획을 가지고 있다. Shopporter는 그보다 훨씬 저렴한 가격에 제작이 가능하고 필수기능들을 모두 탑재했다.

T: 대기업에서 스마트카트의 제작을 하고있기에 기술력에서 차이가 날 수 있다.

SO(강점 , 기회) 전략 : 대기업에서 제작하는 스마트 카트에 비해 저렴하면서 필수 기능을 모두 탑재했기 때문에 기존 카트를 사용하는 마트에도 이를 어필 할 수 있다.



7. 향후 개발 사항

7.1 결제 시스템 추가

현재는 물품의 Rfid를 읽어 총액을 합산하는 정도의 기능만 탑재되어 있지만 추후 결제 단말기를 장착함으로써 결제 시스템을 지원해 매장의 Cashier 인건비를 절약하고 소비자는 결제를 위해 기다리는 시간을 아낄 수 있다. 또한 휴대폰으로 연계되는 웹 사이트를 통해 네이버페이, 삼성페이, 계좌이체 등 인터넷 상에서 사용되는 다양한 결제 방식을 채용 가능하다.

7.2 쇼핑 안내 기능 추가

매장 내 물품 배치도와 음성 인식 기능을 탑재해 고객이 필요한 물품을 말하거나 입력할 시 카트가 알아서 해당 물품의 위치로 안내해주어 처음 방문한 고객도 헤매지 않고 쇼핑이 가능함.

7.3 Beacon

Beacon 을 활용하여 이용자의 스마트폰에 설치된 앱과의 거리를 측정하여 이용자를 자동으로 다니며 쇼핑할 수 있게 함.



8. 프로젝트 세부추진계획 및 세부일정

프로젝트 기간		2019.05.17 ~ 2019.06.20				
구분	추진내용	프로젝트 기간				
		1주	2주	3주	4주	5주
계획	아이디어 회의					
분석	아이디어 관련 구체적인 자료조사 & 역할분담					
개발	개발환경 구축					
	Android 어플리케이션 개발					
	아두이노 Sensor 제어 프로그래밍					
	Django를 활용한 Web 개발					
	개발 파트 연동 및 프로토타입 제작					
	테스트, 최종 보고서 작성					
	시연 및 프로젝트 종료					

9. 팀원 담당 업무

이름	역할
박영준 🏰	Web Server 개발 및 DB 구축
송진호	Raspberry Pi & Web Server간 Socket 통신
김현석	Arduino & Raspberry Pi 제어 및 제품 제작
김효민	Android Application 제작

10. Source Code

별도 첨부