

# SMART RC CAR

## -using Raspberry Pi-

Team name : KHar

2018102106 문희준

2018102137 장재운

2018102125 이승현

## Index

1. 프로젝트 소개 ( Project introducing )
2. 프로젝트 목적 ( Project purpose )
3. 관련 이론들 ( Related theories )
4. 프로젝트 진행 과정 ( Process )
5. 결과 및 분석 ( Results & analysis )
6. 기대효과 ( Project effects )

### 1. 프로젝트 소개 (Introducing project)

우리가 제작하는 자동차의 주요 성능은 line following, obstacle avoiding, 그리고 카메라를 이용한 원거리 조종이다. Line following과 light following의 경우 기존의 오픈 소스를 이용해 구현해 봄으로써 전체적인 구조를 이해하기 위한 목적으로 사용하게 되었고, 원격 조정의 경우 Raspbberri pi에 카메라 모듈을 설치하였다. 카메라 모듈에서 받은 정보들은 wifi 안테나를 이용해 웹서버로 전송한 후에, 그것을 다른 디바이스를 이용하여 실시간으로 정보를 얻을 수 있게 프로그래밍하였다. 컨트롤러를 함께 설치하여 무선으로 차를 조종할 수 있도록 설계하여 차체를 움직임과 동시에 차체에 연결되어 있는 카메라의 화면을 실시간으로 볼 수 있도록 하였다.

### 2. 프로젝트 목적 (Project Purpose)

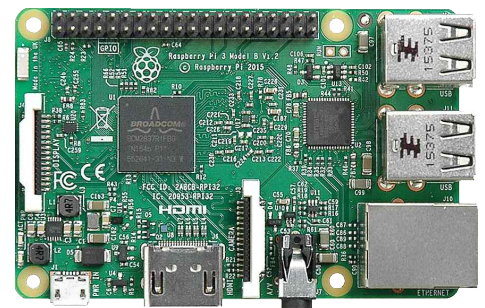
오픈 소스를 사용해 카메라로 실시간 원거리 조정이 가능한 자동차를 구현하는 것을 목적으로 했다. 부가적으로 ultrasonic, line-follower 센서를 부착하여 장애물 피하기, line-following 기능을 추가하였다.

### 3. 관련 이론들 ( Related theories )

- 1) 라즈베리 파이 ( Raspberry pi )

사용한 OS : Raspbian (Wheezy port with faster FP support)

Architecture



라즈베리 파이는 리눅스 커널 기반 운영 체제를 사용한다. Raspbian이라는 라즈베리 파이에 최적화된 데비안 계열의 자유 운영 체제가 현재로서는 가장 권장되는 시스템이다.

## 2) 초음파 센서 (Ultra sonic sensor)



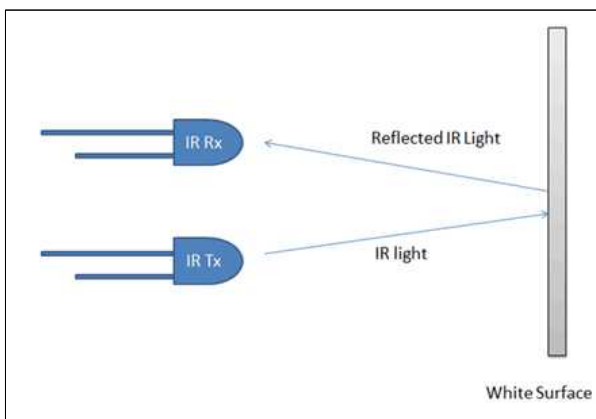
초음파 센서는 40000 헤르츠의 초음파를 공기 중으로 방출 시킨다. 만약 경로에 장애물이 있으면 이것은 다시 모듈로 반사되어 돌아올 것이다. 시간과 속도를 고려하여 계산하면, 거리를 계산할 수 있다.

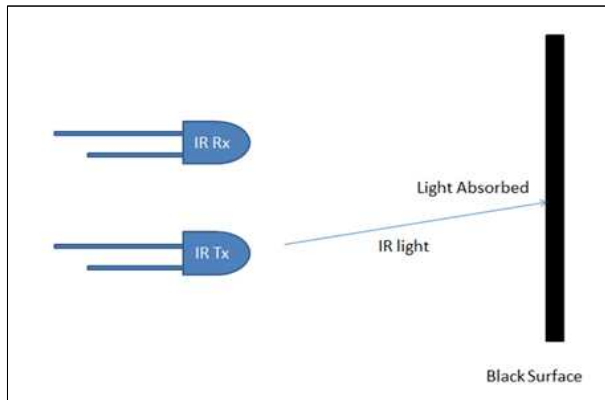
It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

>reference

<https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

## 3) line-following sensor





line follower의 기본 개념은 빛과 관련이 되어있다. 여기서는 흑백표면의 개념이 사용된다. 빛을 흰색 표면에 비출 때는 거의 반사되지만, 검은색 표면에 비출 때는 완전히 흡수된다. 이러한 빛의 특성을 이용한다. 아두이노(라즈베리파이) 센서 기반에서는 IR 전송기와 수신기(사진 다이오드)를 이용한다. 이것은 빛을 쏘고 받을 때 사용된다. IR 전송기는 적외선을 발사한다. 적외선이 흰색 표면에 비춰지면, 적외선은 반사되고 포토다이오드에 다시 감지되어 약간의 전압의 변화를 일으킨다. IR이 검은색 표면에 비춰지면, 빛은 흡수되고 반사된 빛이 존재하지 않는다. 따라서 포토다이오드는 어떠한 빛이나 광선을 감지하지 못한다. 이 센서에서는 하얀색 부분은 1의 출력값을, 검은색 부분은 0의 출력값을 내보낸다.

Concept of working of line follower is related to light. We use here the behavior of light at black and white surface. When light fall on a white surface it is almost full reflected and in case of black surface light is completely absorbed. This behavior of light is used in building a line follower robot. In this arduino based line follower robot we have used IR Transmitters and IR receivers also called photo diodes. They are used for sending and receiving light. IR transmits infrared lights. When infrared rays falls on white surface, it's reflected back and caught by photodiodes which generates some voltage changes. When IR light falls on a black surface, light is absorb by the black surface and no rays are reflected back, thus photo diode does not receive any light or rays. Here in this arduino line follower robot when sensor senses white surface then arduino gets 1 as input and when senses black line arduino gets 0 as input.

>reference

<https://circuitdigest.com/microcontroller-projects/line-follower-robot-using-arduino>

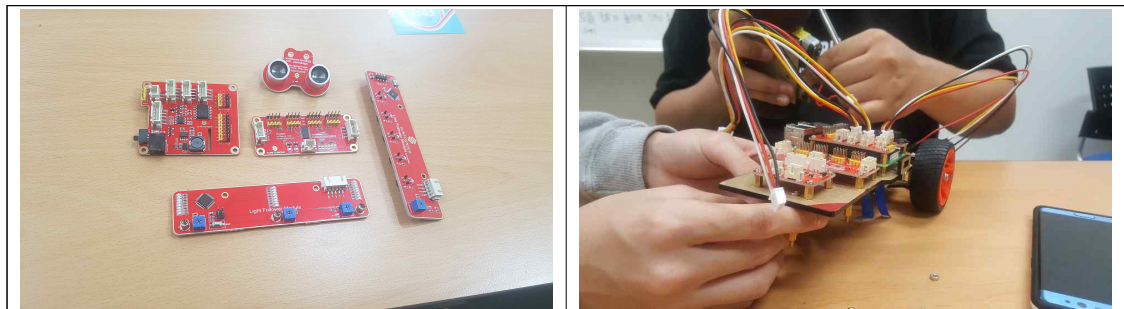
#### 4) 원격 이미지 전송 ( real-time image transfer )

먼저 라즈베리 파이에 usb wifi adapter를 설치하였다. 이 후, 와이파이에 연결한 후, Sunfounder사의 open server를 이용하여 카메라에 얻은 이미지들을 전송한다. 이 후, 컴퓨터나 스마트폰을 이용하여 서버에 연결하고, 라즈베리 파이의 ip address를 입력하면 실시간으로 카메라에 찍힌 정보들이 서버로 전송되어서 원격 조정이 가능하다.

First, we installed a usb wifi adpter on the raspberry pi. After connecting to WiFi, we use Sunfounder's open server to transmit the images to the camera. After that, connect to the server using a computer or smart phone, enter the ip address of Raspberry pi, and the information recorded in the camera in real time can be transmitted to the server and remotely controlled.

## 4. 진행 과정(Process)

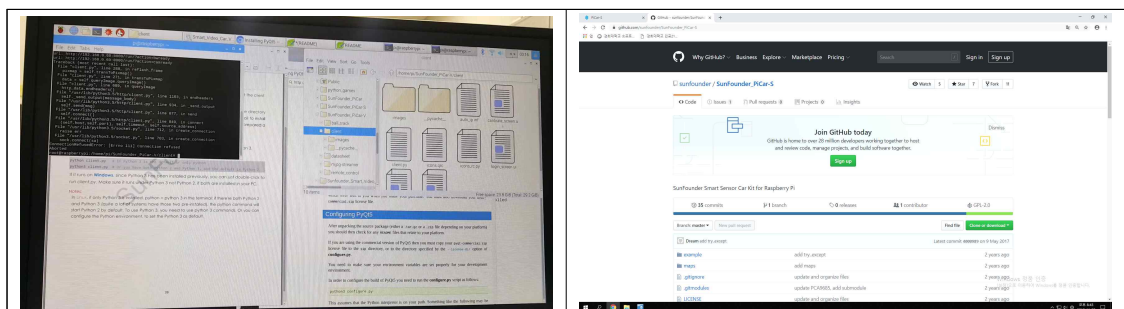
### 1) 하드웨어 조립



하드웨어는 SunFounder사에서 판매하는 line following, ultrasonic avoiding sensor가 포함되어 있는 PiCar-S 키트와 카메라 센서가 포함되어있는 PiCar-V 키트를 조합하여 제작하였다.

차체의 구성은 기본적으로 키트에 포함되어있는 합판에 Raspberry Pi와 3가지 모듈을 연결하고 센서들을 모듈에 연결하는 형태로 이루어졌으며, 그 외에도 뒷바퀴에는 각각 모터를 달아 출력을 높였으며 앞바퀴 역시 입력된 코드에 따라 방향이 전환되도록 만들어졌다. 하드웨어의 조립 자체는 키트를 이용했기 때문에 두 키트를 조합하는 과정을 제외하고는 크게 문제되는 부분 없이 진행할 수 있었다.

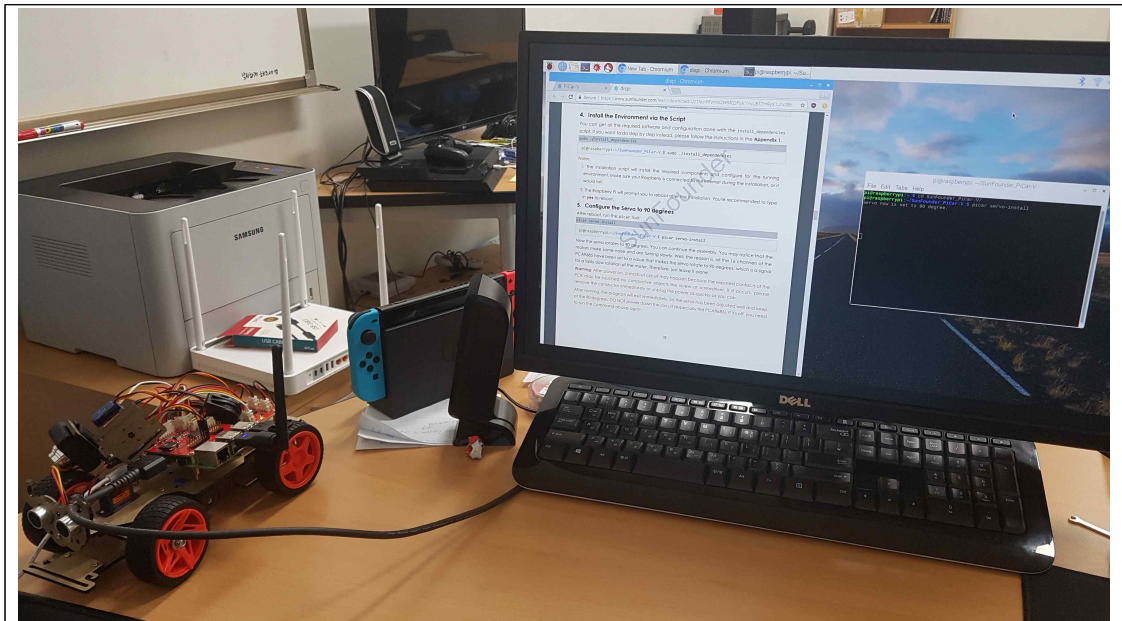
### 2) Open Source Code 설치



Open Source는 GitHub에 있는 데이터를 받아오는 형식으로 진행했다. 우선 Raspberry Pi에 운영체제인 Raspbian을 설치하고, Raspberry Pi를 실행시켜 내부에 있는 터미널을 통해 GitHub에 있는 데이터를 받아왔다. 그 후 받은 Source code를 이용해 필요한 파일을 설치하는 과정을 진행했다. 그런데 이 과정에서 각 과정을 실행하기 위해 필요한 파일들이 제대로 표기되지 않아 직접 코드를 보고 차근차근 과정을 진행해 나갔던 상황이 빈번히 일어나 프로젝트를 진행하는 과정에서 가장 오랜 시간이 걸렸던 작업이었다.

특히 제작한 RC카를 원격 조종하기 위해서는 Open Source를 이용해 Client를 다운받아야했는데, 이 때 필요했던 python3-pip와 pyqt5의 설치 과정에서 설치가 제대로 이루어지지 않을뿐더러 오류가 나는 경우가 많았었다. 설치에는 몇 가지 방법이 있었는데, 이 방법들이 모두 불가능해 여러 방법을 찾아본 결과 불가능했던 이유가 Raspbian에 등록된 계정의 접근성이 낮아 파일의 수정이 불가능하기 때문이라는 사실을 알아낼 수 있었다. 이 과정에서 터미널 내에서 계정을 root 계정으로 전환시켜 접근성을 높이는 방법을 찾아내고, 이 방법을 통해 Client를 설치할 수 있었다.

### 3) 소프트웨어와 하드웨어의 호환

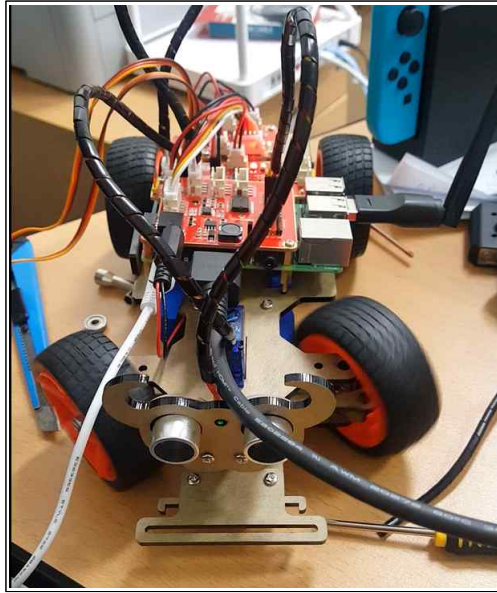


필요한 파일을 설치한 후에는 조립한 하드웨어와 설치한 소프트웨어를 호환시켜 RC카를 작동시키는 작업에 들어갔다. 이 과정에서 어려웠던 부분은 하드웨어를 제대로 작동시키기 위한 수치 조정을 터미널을 통해 직접 진행해야 한다는 부분이었다. 우선 앞바퀴가 직진으로 인식하는 기준이 되는 방향이 실제 직진 방향과 맞도록 각도를 조정해주는 소프트웨어를 반복 실행시켜 각도를 조절하는 과정을 진행했다. 그 후에는 앞바퀴가 좌우로 움직이는 각도를 조정하기 위한 작업과 line following sensor와 ultrasonic avoiding sensor, 앞바퀴와 뒷바퀴가 제대로 인식, 작동을 하는지 확인하기 위한 테스트 과정을 진행했으며, 원격 조종의 경우 Surver를 실행시키고 설치했던 Client를 실행시켜 RC카의 조종과 카메라 센서의 작동 여부를 확인하는 작업을 진행했다.

이 과정에서는 특히 앞바퀴의 각도를 조정하는 과정에서 문제가 많이 생겼다. 수치가 조금만 틀어져도 각도가 완전히 틀어져 방향 조절을 위해 사용하는 장치에 무리가 가서 내부가 타버리는 상황이 발생했기 때문에 최대한 주의를 기울여 각도를 조정할 수밖에 없었다. 그 외의 센서와 바퀴의 경우는 대부분 제대로 조정을 마쳐 원활하게 작동하는 것을 확인할 수 있었다.

### 4) RC카 구동





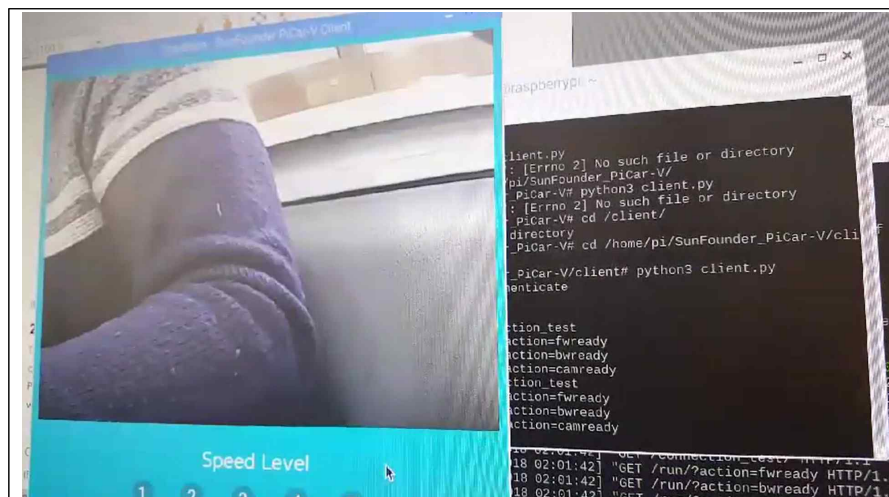
마지막 작업은 RC카를 구동시키는 것이었다. 원격 조종의 경우 Raspbian에서 실행시킨 서버에 다른 기기(노트북, 스마트폰)로 접속하여 원격으로 RC카를 작동시키는 과정을 진행했고, line following의 경우 바닥에 검은색 절연 테이프를 이용해 트랙을 만들어 트랙을 따라 RC카를 이동시켜보았다. 마지막으로 obstacle avoiding의 경우는 경로 상에 장애물을 두고 RC카가 이 장애물을 피해가는지 확인하는 작업을 진행했다.

코드에 따르면 line following의 경우 5개의 센서가 바닥의 경로를 인식해 트랙 옆을 따라가고, obstacle avoiding의 경우는 인식한 거리를 10cm를 기준으로 10cm보다 작으면 turn, 멀면 직진하며 장애물과의 거리가 매우 가까우면 후진을 하는 방식으로 작동한다. 작동시켜본 결과에 대한 언급은 결과 및 분석 부분에서 자세히 언급하도록 하겠다.

## 5. 결과 및 분석 ( Results & analysis )

라즈베리파이를 주 CPU로 wifi 모듈, 카메라, line following sensor, 초음파 센서들을 이용하여 RC카를 세 가지 버전으로 구현해보았다.

### 1) 원격 조종



먼저 wifi 모듈과 카메라를 이용한 기본적인 원격 조종이 가능한 rc car를 구현하였다. wifi 모듈을 이용하여 라즈베리파이와 사용할 서버를 연결해주어 pc뿐만 아니라 모바일 환경에서도 Client를 실행을 한다면 원격 조종이 가능하였고 카메라로부터 정보 또한 정상적으로 받아올 수 있었다. pc에서는 딜레이가 심하지 않아 구동시키는데 큰 문제는 없지만 모바일 환경에서는 딜레이가 심해 구동시키는데 불편함이 있었다. 그리고 wifi가 없는 곳에서는 사용할 수 없다는 한계점도 있다. 때문에 wifi가 없거나 약한 곳에서 대안으로 블루투스 모듈을 이용하는 방법도 있다.

## 2) line tracking



두 번째 버전은 line following sensor를 이용한 line tracking이 가능한 RC카를 구현해 보았다. 이 line following sensor에는 다섯 개의 빛을 감지하는 센서가 있다. 이 센서들이 빛의 세기를 감지하여 일정 값보다 약하면 0으로, 일정 값보다 세면 1로 저장하고 이를 다시 list의 형태로 저장하여 출력해준다. (예시 1,1,1,1,0로 출력되면 다섯 개의 센서 중 가장 오른쪽에 센서에만 빛이 약해졌다는 것을 뜻한다.) 그리고 각각의 리스트일 때 정해진 행동을 하도록 구현하였다. 빛의 세기를 이용하기 때문에 빛의 세기를 더욱 잘 감지할 수 있도록 흰색 바닥에 검정색 테이프를 이용하여 구동시켜보았다. 센서는 잘 작동하였지만 바퀴가 바닥에서 미끄러지는 바람에 제대로 작동하지 않은 문제점이 있었다. 이러한 문제를 해결하기 위해서는 바퀴를 미끄러지지 않는 것으로 교체해야 할 필요가 있어 보인다.

## 3) obstacle avoiding

Ultrasonic avoiding sensor를 이용하여 장애물을 감지하고 피할 수 있는 RC카를 구현해 보았다. 초음파 센서는 초음파를 쏘 장애물에 부딪혀 되돌아오는 시간을 저장하고 시간과 초음파 속도를 이용하여 거리를 계산한 후 이를 출력해준다. 그리고 특정 거리보다 가까울 경우 backward 한 후 turn 하도록 구현하였다. 초음파 센서를 이용하여 장애물과의 거리를 감지하는 것에는 문제가 없었지만 backward 명령을 하기도 전에 뒤로 가거나 장애물과

의 거리가 먼데도 불구하고 앞으로 가지 않는 등 구동하는데 알 수 없는 오류들이 발생하는 문제점이 있었다. 이에 대한 문제점을 파악하는 것이 필요하다.

#### 4) open cv

추가적으로 open cv를 이용하여 RC카가 차선을 인식하고 그 사이로 지나가게 하는 기능을 구현해보려고 시도하였다. 그러나 hough transform은 형광등과 같은 빛에 많은 영향을 받기 때문에 구현하는데 어려움이 있고 실제로 hough transform 자체가 실제 자동차 모델에서 구현되는 기능으로 rc car에서 구현하는데 한계가 있었다. 이러한 것들이 해결되었다고 가정하더라도 차선 사이로 지나가는 코드를 만드는데 어려움이 있어 결국 구현하지 못하게 되었다.

### 6. 기대효과 ( Project effects )

- 1) 원격 조종, line tracking, obstacle avoiding 기능 모두 한 기계에서 구현하는 것이 가능했다. 그러나 동시에 작동하는 것은 불가능했으므로 cpu를 추가적으로 장착하는 등과 같은 해결방법을 통해 보완한다면 더욱 자율 주행 자동차에 가까운 모델을 구현할 수 있을 것이라고 기대된다.
- 2) line tracking 기능을 구현하기 위해 흑백으로만 되어있는 바닥에서 진행하였는데 더욱 향상된 센서를 이용한다면 특정 환경에서만이 아닌 일반 차선에서도 활용될 수 있을 것으로 기대된다.