

<https://doi.org/10.7236/IIBC.2018.18.2.149>

IIBC 2018-2-18

라즈베리파이 기반 다기능 RC카 컨트롤러

Raspberry-based multi-function RC car controller

이명균*, 이용수**, 김정준***

Myoung-Gyun Lee*, Yong-Soo Lee**, Jeong-Joon Kim***

요약 현재 RC카는 남녀노소를 불문하고 정밀한 컨트롤과 질주 본능을 느끼고 싶어하는 모든 사람이 참여할 수 있는 하나의 문화로써 자리잡게 되었다. 또한 RC카는 직접 차량을 운전하면서 느끼는 체감을 작고 가벼운 몸체 하나로 경험할 수 있다는 것을 이유로 많은 사람들에게 주목받고 있다. 이러한 트렌드에 맞춰 다양한 개발 회사들은 각각의 환경에 맞는 주행, 속도 개선과 같은 기능적인 부분과 그 정교함에 초점을 두어 개발하고 있다. 물론, 주행 성능만으로도 굉장한 파급 효과를 불러올 수 있지만, RC카를 가지고 할 수 있는 것은 부품 변경, 모터 개조 정도 밖에 없다. 따라서, 라즈베리파이 기반의 RC카를 제작해 어플리케이션으로 컨트롤 하고 센서 인식을 통한 다양한 기능의 추가와 웹캠을 사용해 얻을 수 있는 영상 정보를 통한 주행을 할 수 있는 RC카를 개발한다. 작은 RC카 한 대로 사람이 진입할 수 없는 다양한 건설, 사고 현장에서 움직이는 로봇으로써의 사용과 사용자들의 흥미를 이끌 수 있을 만한 기능을 겸비할 수 있도록 연구를 수행한다.

Abstract Now RC cars have become a culture where everyone who wants to feel precise control and rushing instinct regardless of sex, young or old can participate. In addition, RC cars have attracted a lot of people because they can feel the feeling of driving while driving a car in a small, lightweight body. In line with these trends, various development companies are focusing on functional aspects such as driving and speed improvement for each environment and their sophistication. Of course, driving performance alone can bring a tremendous ripple effect, but what you can do with a RC car is only part replacement and motor modification. Therefore, we will develop RC car based on raspberry pie, control by application, add various functions through sensor recognition, and develop RC car that can travel through video information that can be obtained using web cam. As a small RC car, we carry out research that can combine the functions that can be used as a robot that moves at various construction sites and accident sites where people can not enter, and which can attract users' interest.

Key Words : RC cars, raspberry pie, application, sensor recognition, video information

1. 서 론

한 때, 선풍적인 인기를 끌었던 미니카와 달리 직접 조종하고 경쟁할 수 있는 현재의 RC카는 단순히 장난감

넘어 동호회인이 50만명에 육박할 정도로 많은 사람들이 좋아하는 취미로써 자리 잡았으며, 전국적으로 대회도 개최되는 등 스포츠의 일부분으로써 자리하고 있다. 이처럼 사람들의 관심이 많아지면서, 단순히 평지의 트랙

*정회원, 한국산업기술대학교 컴퓨터공학과

**정회원, 여주대학교 컴퓨터정보과

***정회원, 한국산업기술대학교 컴퓨터공학과

접수일자: 2018년 3월 20일, 수정완료: 2018년 4월 5일

게재확정일자: 2018년 4월 6일

Received: 20 March, 2018 / Revised: 5 April, 2018

Accepted: 6 April, 2018

***Corresponding Author: jkim@kpu.ac.kr

Dept. of Computer Engineering, Korea Polytechnic University, Korea.

에서만 주행하는 것이 아니라 MTB 자전거 처럼 산악용 RC카도 개발되는 등 여러 환경에서의 주행이 가능해졌고, 직접 컨트롤 할 수 있다는 데에 큰 이점을 가지며, 컨트롤러에 따른 주행의 정교함도 추가되고 있습니다. 이처럼 주행의 성능과 정교함에 초점을 두어 다양하게 개발되고 있으며, RC카의 발전만큼 선호하는 사용자 수와 연령대의 폭이 증가하고 있다^[1,2].

RC카의 파급력은 실제 차량에서 착안한 RC카가 속도감도 느낄수 있고 장난감처럼 작고 휴대할 수 있다는 점이 주된 이유인데, 이러한 장점을 살려 인공지능 로봇처럼 건설,사고현장에 인간이 진입할수 없는 공간을 탐사하는 것을 주된 목적으로 하여 다양한 기술을 접목시킨 RC카를 생각해보았다. 먼저, 라즈베리파이를 기반으로 하여 Wi-fi 환경에서 어플리케이션을 통한 RC카의 컨트롤을 주 목적으로 하였으며, Java를 사용하여 라즈베리파이와 어플 간의 통신을 하였다^[3,4]. 다음으로, 라즈베리파이 내부의 제어선을 통해 모터를 제어하고 웹 캠의 스트리밍 기능을 통해 얻을 수 있는 영상 정보를 실시간으로 어플리케이션에 보내주어, 사용자는 카메라에서 보내주는 스트리밍 데이터를 보며 주행하며, 또한 카메라의 딜레이나 통신 상태의 불량에 대비해 초음파 센서를 이용한 장애물 인식을 통해, 사방의 장애물들을 사전 탐지해 정지하는 기능을 수행하고, 부가적인 기능으로는, 자이로스코프 센서를 통한 기울기 인식을 통해 VR기기와 스마트폰을 연결하면 카메라로 보이는 뷰에 따라 좌우측으로 기울이면서 주행할 수 있는 기능을 넣어 사용자들이 공적인 목적이 아니라 개인이 흥미롭게 가지고 놀수있는 장난감으로써의 기능을 제공한다^[5-8].

본 논문은 다음과 같이 구성된다. 2장에서는 본 연구에서 사용한 언어와 환경에 대한 개념 및 활용 방향에 대해 설명하고 3장에서는 본 연구를 진행하기 위한 설계과정에 대해 기술하였으며, 4장에서는 구현과정, 5장에서는 결론을 기술했다 한다.

II. 관련 연구

1. Linux

아래의 그림은 본 논문에서 사용하는 라즈베리파이의 OS인 Linux 커널의 구조에 대해 보여주고 있다^[5,6].

커널은 컴퓨터의 가장 기본적인 각 장치들을 관리하

고 제어하기 위한 소프트웨어이다. 컴퓨터가 부팅되면서 GRUB 과 같은 부트로더에 의해서 메모리로 로드되어 컴퓨터가 꺼질 때까지 항상 메모리에 상주해서 컴퓨터의 각 장치들을 관리하고 제어하는 역할과 사용자들과 의사소통을 지속적으로 하게 된다.

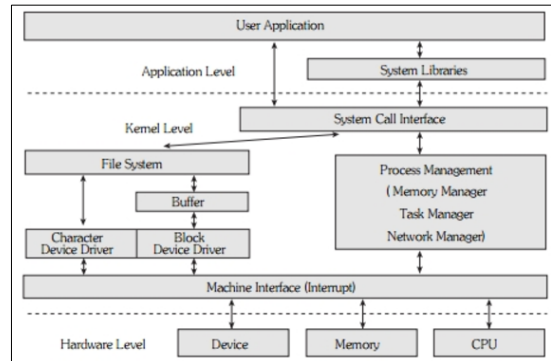


그림 1. Linux 커널의 구조

Fig. 1. Structure of the Linux kernel

이처럼 핵심적인 기능을 맡고 있는 것이 커널이지만 커널은 운영체제 그 자체로는 볼 수 없다. 하지만, 리눅스의 커널 의존도가 높아지듯, 점점 운영체제의 거의 모든 기능에 빠질 수 없는 소프트웨어로 자리하면서 최근 일 반적인 커널들은 운영체제라는 말로 통용된다.

리눅스 커널은 그림 1처럼 크게 응용프로그램, 시스템 콜 인터페이스, 커널, 하드웨어 네 가지로 구성되어있다.

응용프로그램은 사용자 모드에서 구동되는 프로그램으로 직접 사용자와 소통하는 역할을 하고 시스템 콜 인터페이스는 응용프로그램에게 커널의 서비스를 제공하는 인터페이스이다.

2. Android

안드로이드의 기본 구조는 아래 그림처럼 크게 다섯 부분의 레이어(Layer)로 나뉜다.

안드로이드는 리눅스라는 오픈소스 운영체제의 소스 코드를 기반으로 개발된 모바일 디바이스용 운영체제로, 리눅스와 비슷한 시스템 체계를 따르고 있는 무료 배포 라이선스이다.

Linux kernel 레이어는 안드로이드 가장 하단의 리눅스 커널로 인해, 리눅스 기반을 가지고 있다고 할 수 있으며, linux의 core 부분과 휴대폰 사용에 필요한 부분만 가지고 있다.

Libraries 레이어는 SQLite SSL OpenGL 등의 주요 라이브러리를 포함한다.



그림 2. 안드로이드의 기본 구조
 Fig. 2. Basic structure of Android

Android Runtime 레이어에는 구글의 자바가상머신인 Dalvik과 Core library 들이 위치하고 있으며, 대부분의 라이브러리들이 C++로 되어있다.

Application Framework 레이어에는 안드로이드 SDK에 의해 제공되는 API 라이브러리가 emulator 등의 툴들이 위치하고 있다.

Applications 레이어는 사용자들과 직접 소통하는 곳으로 안드로이드 자체의 기본 앱이나, 사용자가 설치한 앱들을 가리킨다.

3. Java

그림 3은 플랫폼 독립적인 Java의 구조를 보여준다.

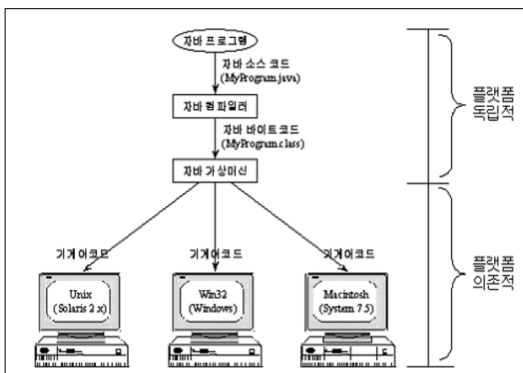


그림 3. 플랫폼 독립적인 Java
 Fig. 3. Platform-independent Java

자바는 객체지향 언어이며, 컴파일 시 높은 신뢰성을 특징으로, 자바 컴파일러가 생성하는 바이트코드를 통해 어떤 플랫폼에서도 사용할 수 있다는 점에서 이식성이 매우 뛰어난 언어라는 것을 알 수 있다.

플랫폼이란 프로그램이 실행되는 하드웨어 또는 운영체제와 같은 소프트웨어적인 환경을 말한다. 하지만 자바 플랫폼은 하드웨어와 무관하게 동작하는 오직 소프트웨어적인 플랫폼이란 점에서 다른 플랫폼과 달라 자바 가상머신과 자바 API를 별도로 갖는다.

그림의 진행방향 처럼 자바 언어로 작성한 자바 프로그램은 자바 컴파일러를 이용하여 자바 바이트코드로 컴파일 되고, 이 자바 바이트코드는 자바 가상머신에 의해 해석되어 실행되는데, 이때 자바 가상머신은 자바 바이트코드에 대한 해석기 즉, 인터프리터로 동작하게 된다. 이렇게 자바 프로그램은 컴파일 방식 및 인터프리터 방식이 모두 적용된다는 것을 확인할 수 있다.

III. 본 론

1. 세부 설계 및 구현

아래 그림은 RC카의 전체적인 작동 흐름을 보여준다.

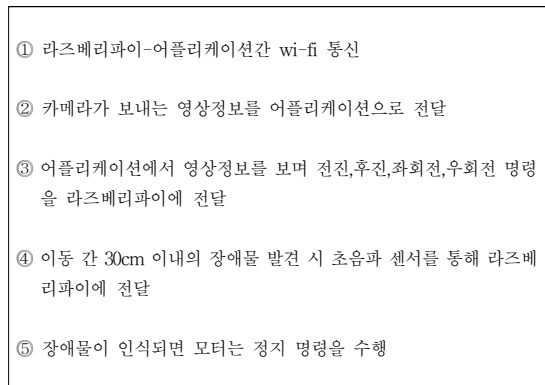


그림 4. RC 카 작동 과정
 Fig. 4. RC car operation process

우선, RC카를 구동시키기 위해 앞서 외부기기(카메라, 초음파 센서, 모터드라이버)를 사용하기 위해 라즈베리파이 내부에 필요한 라이브러리들을 설치한다. 필요한 라이브러리 및 실행 파일의 설치가 끝나면, 라즈베리파이 내부에서 먼저 실행해 정상적으로 기능을 수행하는지 확인한다.

라즈베리파이 내부의 환경 구축이 끝나면 그림 4처럼 크게 다섯 가지 단계를 거쳐 어플리케이션으로 조종할 수 있는 RC카를 개발할 수 있게 된다.

2. 시스템 구성

아래 그림은 RC카가 작동하면서 사용되는 시스템 시나리오의 흐름을 보여준다.

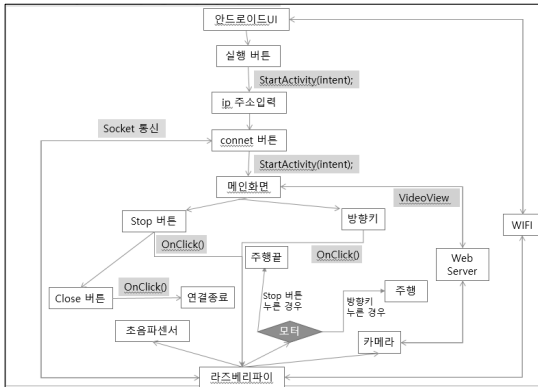


그림 5. RC카의 시스템 시나리오
Fig. 5. System scenario of RC car

그림 5는 RC카의 기능을 사용하기 위한 주요 함수들을 나타내고 각 버튼과 기능이 어떠한 일련의 과정들을 통해 수행되는지를 나타내는 흐름도이다.

3. 어플리케이션 준비

아래 그림 6은 실제 어플리케이션을 실행했을 때 나오는 화면을 구축하기 위한 xml 파일이다.

```
<WebView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/webView"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_above="@+id/editTextIPAddress" />

<Button
    android:id="@+id/buttonConnect"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:text="Connect"
    android:textColor="#FFFFFF"
    android:layout_marginBottom="11dp"
    android:layout_above="@+id/textViewStatus"
    android:layout_alignParentLeft="true" />
```

그림 6. activity_main.xml
Fig. 6. activity_main.xml

그림 6은 카메라에서 전송 받을 화면을 나타낼 WebView와 Button을 추가하기 위한 설정 값들을 나타낸 소스 코드의 일부이다. 위 그림과 같은 코드를 실행함으로써 지정한 곳으로 버튼 및 화면이 배치되고 각각의 기능을 수행할 수 있다.

다음은 그림 6을 실행시켰을 때 나오게 될 화면이다.



그림 7. Main 화면
Fig. 7. Main Screen

그림 7처럼 카메라가 전송하는 영상 정보를 표시할 WebView 란과 각 버튼들이 배치되어 있는 것을 확인할 수 있다.

4. 라즈베리파이-어플리케이션 통신

아래 그림은 어플리케이션을 통해 수신 받은 데이터를 TCP를 통해 라즈베리파이로 송신하는 java 파일이다.

그림 8은 받은 데이터의 크기와 데이터 형식을 지정해주는 소스 코드로, 전달 받은 데이터를 라즈베리파이로 정상적으로 송신하면 try-catch문을 빠져나올 수 있게 된다.

```

public rcvthread(logger logger, Socket socket){
    this.logger = logger;
    flag = 1;
    this.socket = socket;
}

public void setFlag(int setflag) { flag = setflag; }

public void run() {
    while(flag == 1){
        try{
            rcvBufSize = socket.getInputStream().read(rcvBuf);
            rcvData = new String(rcvBuf, 0, rcvBufSize, "UTF-8");

            if (rcvData.compareTo("[close]")==0){
                flag = 0;
            }
            logger.log("Receive Data : " + rcvData);
        } catch (IOException e){
            e.printStackTrace();
        }
        logger.log("Exit loop");
    }
}
    
```

그림 8. rcvthread.java
 Fig. 8. rcvthread.java

다음으로, 아래 그림은 메인이 되는 파일로써, 이벤트 발생이나 조건에 따라 데이터를 만들고 앞서 생성한 rcvthread를 호출해 통신을 시작하는 역할을 하는 java 파일이다.

```

public void onClick(View arg0) {
    if(arg0 == buttonConnect)
    {
        imm.hideSoftInputFromWindow(editTextIPAddress.getWindowToken(), 0);

        try{
            if(socket!=null)
            {
                socket.close();
                socket = null;
            }

            server = editTextIPAddress.getText().toString();
            socket = new Socket(server, port);
            outs = socket.getOutputStream();

            rcvThread = new Thread(new rcvthread(logger, socket));
            rcvThread.start();
            logger.log("Connected");
            buttonConnect.setVisibility(View.INVISIBLE);
            editTextIPAddress.setVisibility(View.INVISIBLE);
        } catch (IOException e){
            logger.log("Fail to connect");
            buttonConnect.setVisibility(View.VISIBLE);
            editTextIPAddress.setVisibility(View.VISIBLE);
            e.printStackTrace();
        }
    }
}
    
```

그림 9. teleopclient.java
 Fig. 9. teleopclient.java

그림 9는 전체 코드의 일부로써, Connect 버튼을 눌러 연결에 성공했을 때를 나타내는 소스 코드로, 앞서 설명한 그림 8의 rcvthread를 호출해서 입력한 ip와 지정해준

포트번호를 이용해 소켓 통신을 진행하고, 연결에 성공 혹은 실패 했을 때 다루지는 함수가 들어 있다.

그림 8과 그림 9 두 java 파일은 Android studio에서만 사용하는 것이 아니라 라즈베리파이 내부의 Linux에서도 동일한 형태의 자바 코드로 작성해 컴파일과 실행을 진행해야 서로간의 연결에 성공할 수 있다.

5. 모터 제어

모터는 모터드라이버를 경유하여 라즈베리파이의 GPIO 핀과 연결된다. GPIO는 제어 및 전원을 담당하며, 하나의 단자가 입출력 과정을 동시에 수행할 수 없다. 이에 따라, 모터의 전원 입출력 및 제어의 구분을 명확히 하고, GPIO와 연결된 모터를 구동시키기 위해 java 코드 작성 시 gpio 관련 패키지들을 import 해주어야 한다.

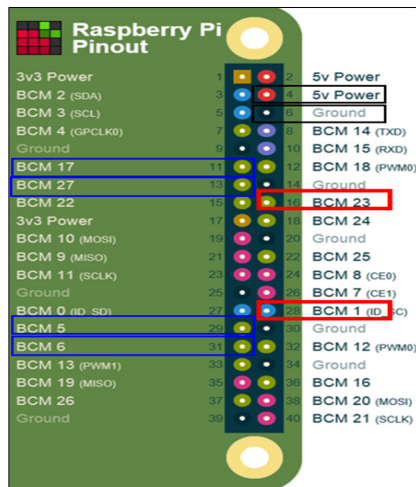


그림 10. 라즈베리파이의 GPIO pin
 Fig. 10. GPIO Pin of Raspberry Pie

그림 10이 개발한 RC카가 사용하는 라즈베리파이3 버전이 가지는 GPIO pin의 현황이다. Power라고 쓰여있는 부분이 전원, BCM이 제어선을 의미한다.

```

buttonConnect = (Button)this.findViewById(R.id.buttonConnect);
buttonClose = (Button)this.findViewById(R.id.buttonClose);
buttonUp = (Button)this.findViewById(R.id.buttonUp);
buttonLeftTurn = (Button)this.findViewById(R.id.buttonLeftTurn);
buttonRightTurn = (Button)this.findViewById(R.id.buttonRightTurn);
buttonDown = (Button)this.findViewById(R.id.buttonDown);
buttonStop = (Button)this.findViewById(R.id.buttonStop);
    
```

그림 11. 설정한 변수와 xml 파일의 연동
 Fig. 11. Interlock of set variables and xml file

그림 11은 사전에 설정한 버튼 변수들을 아무런 기능을 할 수 없던 그림 6에서 만든 xml 파일과 연동시켜 실제 버튼으로써 작동할 수 있도록 하는 코드이다.

6. 카메라 기능

어플리케이션을 통해 직접 보면서 조종할 수 있도록 웹 캠을 이용해 어플리케이션으로 실시간 전송하는 기능을 구현하기 위해, motion 패키지의 설치 및 어플리케이션으로 보내기 위한 설정의 추가 등을 진행하였다.

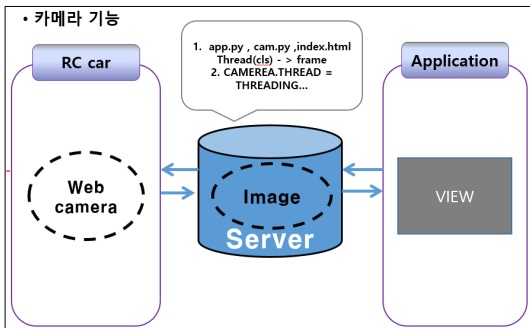


그림 12. 영상정보의 전달 과정
Fig. 12. Transmission process of video information

그림 12는 실제 RC카의 웹 캠에서 촬영되는 영상을 서버를 거쳐 어플리케이션으로 전달되는 과정이 담긴 구조도이다.

```
WebView webView = (WebView)findViewById(R.id.webView);
webView.setWebViewClient(new WebViewClient());
webView.setBackgroundColor(255);
//xx
webView.getSettings().setLoadWithOverviewMode(true);
webView.getSettings().setUseWideViewPort(true);
```

그림 13. 영상 데이터를 Client로 전송하는 과정
Fig. 13. Process of transmitting video data to Client

그림 11의 과정처럼 그림 6에서 만들었던 WebView와 java 코드를 연동시킨다. 그리고 실제 카메라로 들어오는 데이터를 서버를 거쳐 클라이언트(어플리케이션)으로 보내주는 과정을 나타내는 코드가 그림 13이다.

7. 초음파 센서

초음파 센서는 아래 그림 14처럼 초음파의 출력부와 입력부로 나뉘어 출력부에서 발생시킨 초음파가 물체에

닿아 반사되어 입력부로 돌아올 때까지의 시간을 통해 거리를 산출한다.

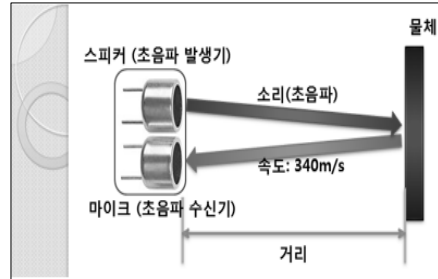


그림 14. 초음파 센서의 작동 원리
Fig. 14. Operation source of ultrasonic sensor

```
digitalWrite (M1, HIGH);
digitalWrite (M2, LOW);
digitalWrite (M3, HIGH);
digitalWrite (M4, LOW);

if (distance <= 30)
{
digitalWrite (M1, LOW);
digitalWrite (M2, HIGH);
digitalWrite (M3, LOW);
digitalWrite (M4, HIGH);
delay(1000);}
```

그림 15. 초음파-모터 연동
Fig. 15. Ultrasound - motor interlock

초음파 센서의 기능이 정상 작동되는지 사전에 확인하려면, 다음 그림 15와 같이 초음파 센서를 통해 측정된 거리 값에 따른 조건문을 주고 모터의 핀을 HIGH에서 LOW로 바꿔주어 설정한 거리 값에 따른 모터의 움직임을 컨트롤 할 수 있다.

8. 어플리케이션 실행

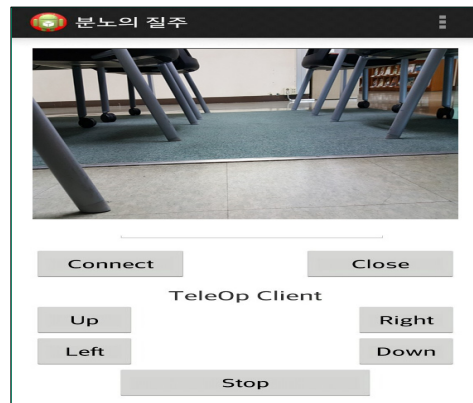


그림 16. 어플리케이션 실행화면
Fig. 16. Execution screen of application

그림 16은 모든 설정이 끝나고 라즈베리파이를 연결 대기 상태로 설정한 후, 라즈베리파이의 고유 ip를 입력해, Connect 버튼을 누르게 되면 라즈베리파이와 어플리케이션 양방간의 연결에 성공해 다음과 같이 카메라를 이용한 영상 정보가 어플리케이션의 상단으로 실시간 전송되고, Up, Left, Right, Down 버튼이 활성화 되면서 네개의 버튼을 통해 RC카는 주행할 수 있게 된다.

VI. 결론

보다 빠른 RC카들은 개발되고 있지만, 보다 다양한 기능을 가지는 RC카의 개발은 눈에 띄지 않는다. 따라서, 본 논문에서는 RC카의 성능 증가가 아닌 기능 구축에 초점을 두고 개발을 시작하였다. 앞서 서술한대로 라즈베리파이 기반의 RC카를 제작하여, 라즈베리파이와 추가로 개발한 어플리케이션을 사용할 스마트폰이 동일한 Wi-fi 환경 내에서 서로 통신할 수 있도록 구현하였다. 서로간의 연결이 이뤄지면, RC카 전면부에 달린 카메라를 통한 영상 정보들이 어플리케이션에 실시간으로 출력되었고, 해당 실시간 영상 정보를 눈으로 확인하면서, 라즈베리파이로 제어하는 4개의 모터들을 구동해 전진, 후진, 좌회전, 우회전이 가능하였으며, 초음파 센서와 물체간의 간격이 30cm이내가 되면 정지하는 기능까지 구현하였다.

하지만, 흥미를 위해 사용하려고 시도했던 자이로스코프 센서를 통한 기술기의 정도에 따른 RC카의 움직임을 제어해보려고 하였으나 현재 사용하는 모터가 DC모터라서 모터의 정밀제어가 힘들다는 문제점을 파악했다. 따라서, 조사 끝에 차후 DC모터 대신 모터의 회전 반경이 180도까지 정밀하게 움직이는 서보 모터를 사용해 모터를 제어한다면 문제가 해결될 것으로 보인다.

References

- [1] Jung, S, J and Sung, K, "Basic Technique for Navigation and Aviation : Management and Security of User in Linux Server", The Korea Navigation Institute, The Journal of Korea Navigation Institute 19(6). pp.587-594. 2015.12.
DOI: 10.12673/jant.2015.19.6.587
- [2] Kim, Y, S, "Feasibility Study of Real-Time Programs in Linux", Korean Institute of Information Technology, The Journal of Korean Institute of Information Technology 10(8). pp.127-134. 2012.8.
- [3] Jeon, Y, H and Ahn, H, S, "Android Based Ubiquitous Interface for Controlling Service Robots", The Institute of Internet, Broadcasting and Communication, The Journal of The Institute of Internet, Broadcasting and Communication 10(3). pp.35-41. 2010.6.
- [4] Cho, H, S and Lee, H, J, "The Android-based Bluetooth Device Application Design and Implementation", Intelligent Transport Systems, The Journal of Intelligent Transport Systems 11(1). pp.72-85. 2012.2.
DOI: 10.12815/kits.2012.11.1.072
- [5] No, S, Y, Shin, S, Y, Park, S, J and Lee, J, C, "Implementation of Java API for Embedded System", Korea Society Of Computer Information, The Journal of Korea Society Of Computer Information 16(1). pp.297-300. 2008.
- [6] Kim, G, D, Won, S, Y and Kim, H, S, "An Object Recognition Performance Improvement of Automatic Door using Ultrasonic Sensor", The Institute Of Electronics and Information Engineers, Journal of the Institute of Electronics and Information Engineers 54(3). pp.97-107. 2017.3.
DOI: 10.5573/ieie.2017.54.3.97
- [7] Jeong, S, Lee, J, J and Jung, W, K, "A Indoor Management System using Raspberry Pi", Korea Academia-Industrial cooperation Society, The Journal of Korea Academia-Industrial cooperation Society 17(9). pp.745-752. 2016.9.
DOI : 10.5762/KAIS.2016.17.9.745
- [8] Kim, Y, W, Byun, J, W, Choi, S, H, Park, S, Y and Park, S, J, "A Usability Analysis Method of Mobile Application using User Behavior Logs", Korea Institute Of Information Scientists and

Engineers, The Journal of Korea Institute Of Information Scientists and Engineers 39(2). pp.91-98. 2012.2.

저자 소개

이 명 균(정회원)



• Myoung Gyun Lee received his MS in Computer Science at Korea Polytechnic University in 2018. His research interests include Database Systems, BigData, etc.

이 용 수(정회원)



• Yong-soo Lee received his MS in Computer Science at Konkuk University in 1989. In 2015, he received his PhD in Information & Control Engineering at Kwangwoon University. He is currently a professor at the Department of Computer Information at Yeosu Institute of Technology. He is the Member of the Korea Institute of Internet, Broadcasting & Communication (IIBC). His research interests include Database Systems, Data Mining, BigData, Wireless Sensor Networks and Ubiquitous Sensor Network (USN), etc.

김 정 준(정회원)



• Jeong Joon Kim received his BS and MS in Computer Science at Konkuk University in 2003 and 2005, respectively. In 2010, he received his PhD in at Konkuk University. He is currently a professor at the department of Computer Science at Korea Polytechnic University. His research interests include Database Systems, BigData, Semantic Web, Geographic Information Systems (GIS) and Ubiquitous Sensor Network (USN), etc.