

// wifi_car

라즈베리파이에 Flask를 이용하여 무선 조정 자동차를 만들어 봤습니다.

// 하드웨어

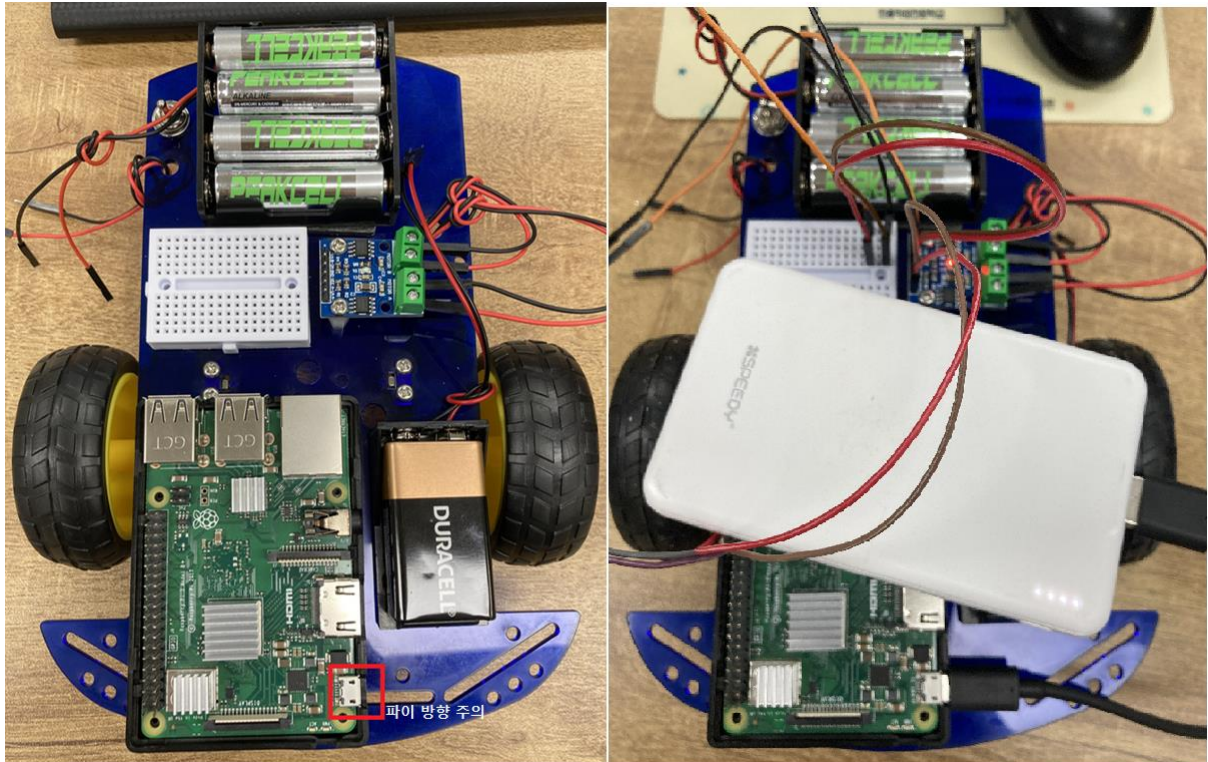
<https://www.devicemart.co.kr/goods/view?no=1385495>



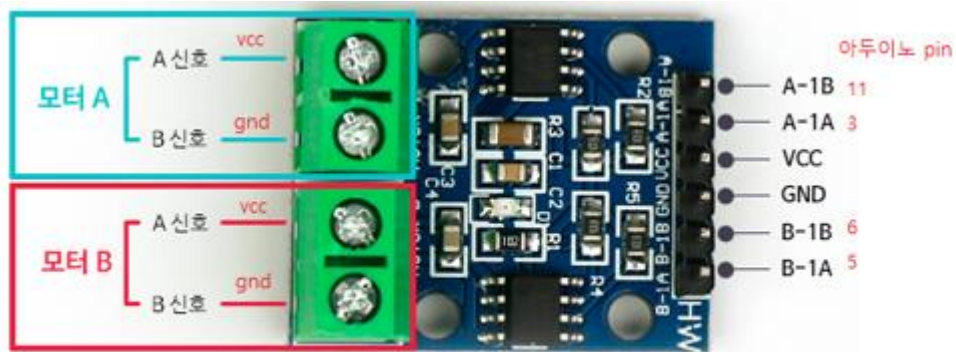
위 키트를 이용하여 블루투스 제어를 확인 했었고요.

이후, 라즈베리파이로 변경하여 내장 블루투스로 제어를 시험 해봤지만,
연결이 불안정하여 wifi 제어로 바로 넘어 갔습니다.

아두이노를 분리하고 그 자리에 아래처럼 라즈베리파이를 장착 합니다.

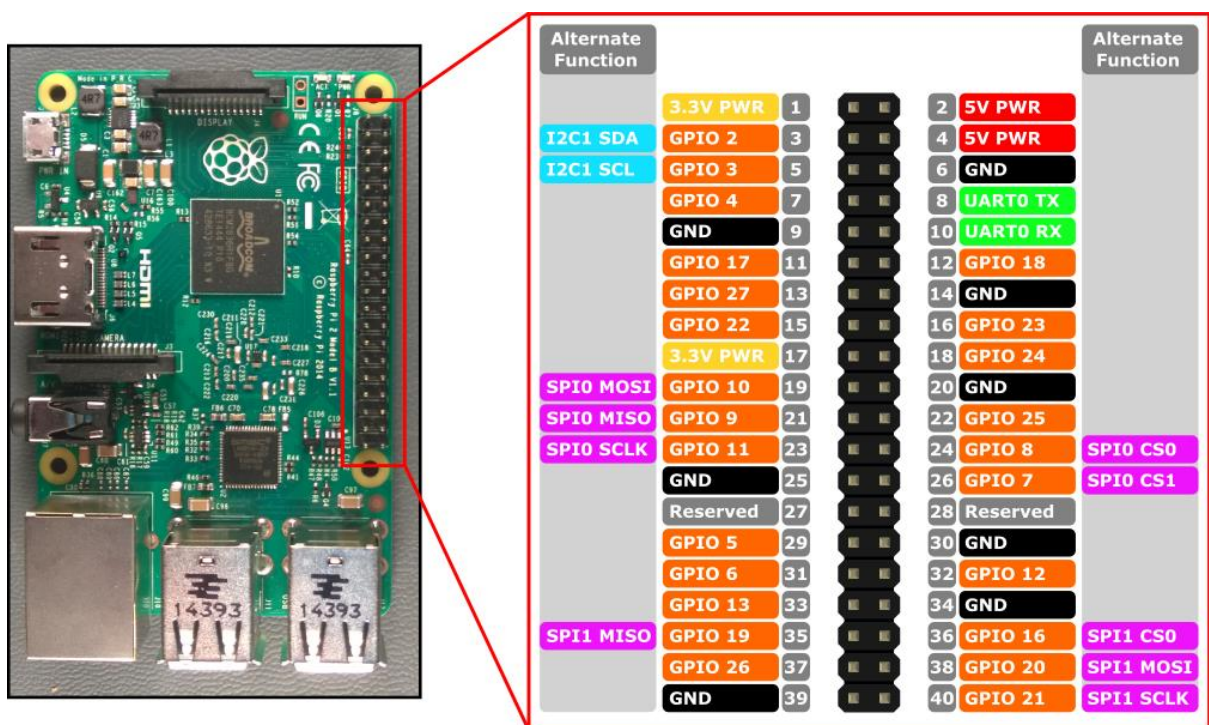
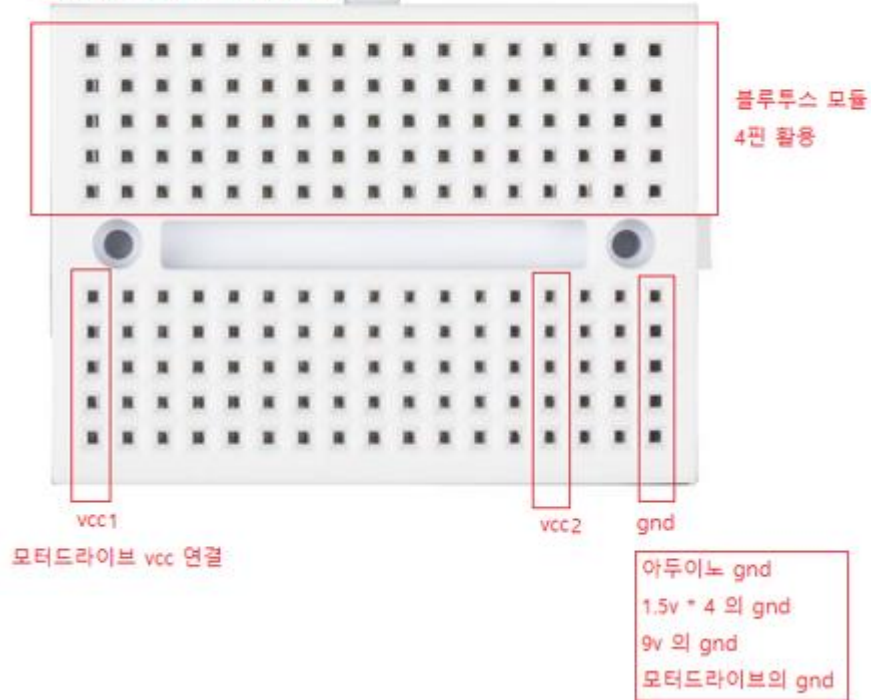


DC 모터는 1.5v 건전지 4개를 연결하여, 모터 드라이버를 통하여 전원 공급해주고, 라즈베리파이에는 휴대용 스마트폰 보조 배터리로 전원을 공급 해줍니다.



L9110s 소형 모터 드라이버 DM141, 단가 1020원

1.5v * 4 의 vcc 는 스위치를 거쳐서 드라이버 모듈의 vcc 를 공급하도록 설정
 즉, 1.5v * 4 의 vcc >> 스위치(+)
 스위치(-) >> 모터드라이브 vcc



// 각 부품을 양면 테이프와 나사로 자리를 고정 했다면,

위 그림을 참고하여 배선 합니다.

저의 경우, 빵판에는

1.5v 건전지 쪽에서 나온 vcc, gnd 와 모터 드라이브에서 나온 vcc, gnd 만 서로 연결 하였습니다.

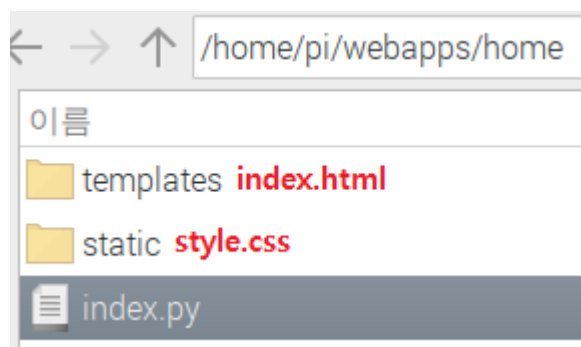
그리고는 라즈베리의 GPIO 5, GPIO 6, GPIO 13, GPIO 26 4개 핀을 모터 드라이브에 연결 했습니다.

위 그림의 11번과 3번핀을 대신하여 26번과 13번을 연결 했습니다.

이렇게 하드웨어 구성은 완료 입니다.

배터리 교체가 용이 하시다면, 무선 상태에서 개발도 좋습니다.

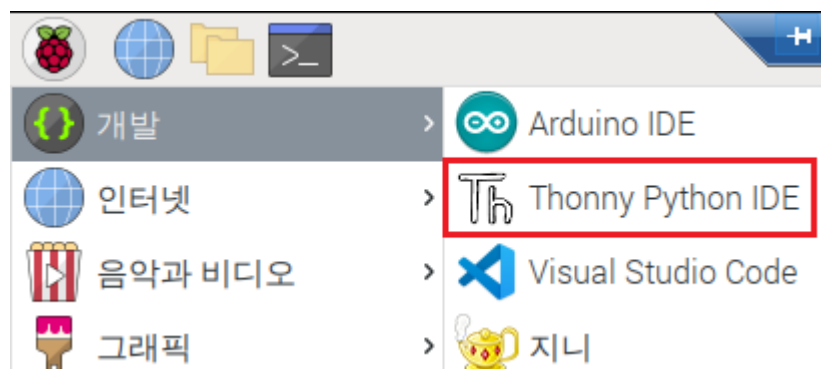
// 소프트웨어



/home/pi 아래에 webapps 폴더를 만들고, 그 안에 다시 home 폴더를 생성합니다.

생성된 home 폴더 아래에 templates, static 폴더를 생성하고, 각각 index.html 과 style.css 파일을 만들고, index.py 로 생성합니다.

라즈베리파이의 개발 >> Thonny Python IDE 를 이용하여, 파이썬 코드를 아래와 같이 작성합니다.



```
// index.py

from flask import Flask, request

from flask import render_template, make_response

import RPi.GPIO as GPIO

app = Flask(__name__)

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

motor_a_a=5

motor_a_b=6

motor_b_a=13

motor_b_b=26


GPIO.setup(5, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(6, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(13, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(26, GPIO.OUT, initial=GPIO.LOW)

motor_a_a_pwm=GPIO.PWM(motor_a_a,50)

motor_a_b_pwm=GPIO.PWM(motor_a_b,50)

motor_b_a_pwm=GPIO.PWM(motor_b_a,50)

motor_b_b_pwm=GPIO.PWM(motor_b_b,50)


@app.route("/")

def home():

    return render_template("index.html")
```



```
@app.route("/moving", methods=["GET"])

def moving():

    state=request.args.get('state')


    try:

        if state=="go":

            GPIO.output(motor_a_a, GPIO.HIGH)

            GPIO.output(motor_a_b, GPIO.LOW)

            GPIO.output(motor_b_a, GPIO.LOW)

            GPIO.output(motor_b_b, GPIO.HIGH)

            #마주보는 모터가 반대로 동작.


            motor_a_b_pwm.ChangeDutyCycle(50)

            motor_b_b_pwm.ChangeDutyCycle(50)

            print("go")

            return make_response("go", 200)


        if state=="back":

            GPIO.output(motor_a_a, GPIO.LOW)

            GPIO.output(motor_a_b, GPIO.HIGH)

            GPIO.output(motor_b_a, GPIO.HIGH)

            GPIO.output(motor_b_b, GPIO.LOW)

            #마주보는 모터가 반대로 동작.


            motor_a_b_pwm.ChangeDutyCycle(50)
```

```
motor_b_b_pwm.ChangeDutyCycle(50)

print("back")

return make_response("back", 200)
```

```
if state=="stop":
```

```
    GPIO.output(motor_a_a, GPIO.LOW)

    GPIO.output(motor_a_b, GPIO.LOW)

    GPIO.output(motor_b_a, GPIO.LOW)

    GPIO.output(motor_b_b, GPIO.LOW)
```

```
    motor_a_a_pwm.ChangeDutyCycle(0)

    motor_a_b_pwm.ChangeDutyCycle(0)

    motor_b_a_pwm.ChangeDutyCycle(0)

    motor_b_b_pwm.ChangeDutyCycle(0)

    print("stop")

    return make_response("stop", 200)
```

```
if state=="left":
```

```
    GPIO.output(motor_a_a, GPIO.HIGH)

    GPIO.output(motor_a_b, GPIO.LOW)

    GPIO.output(motor_b_a, GPIO.HIGH)

    GPIO.output(motor_b_b, GPIO.LOW)

    motor_a_b_pwm.ChangeDutyCycle(50)

    motor_b_b_pwm.ChangeDutyCycle(50)

    print("left")
```

```
return make_response("left", 200)
```

```
if state=="right":
```

```
    GPIO.output(motor_a_a, GPIO.LOW)
```

```
    GPIO.output(motor_a_b, GPIO.HIGH)
```

```
    GPIO.output(motor_b_a, GPIO.LOW)
```

```
    GPIO.output(motor_b_b, GPIO.HIGH)
```

```
    motor_a_b_pwm.ChangeDutyCycle(50)
```

```
    motor_b_b_pwm.ChangeDutyCycle(50)
```

```
    print("right")
```

```
    return make_response("right", 200)
```

```
except expression as identifier:
```

```
    return "fail"
```

```
if __name__ == "__main__":
```

```
    app.run(host="0.0.0.0")
```

// static 폴더에 style.css 는 적당히 스타일을 주었지만, 시크바의 경우, w3school 의 코드를 참조 하였습니다.

https://www.w3schools.com/howto/howto_js_rangeslider.asp

// index.html 은 아래와 같이 속도를 조절하는 시크바를 구성 했지만, 아직 반영하지 못했고, 4방향 이동과 정지만 ajax 처리 하였습니다.

// index.html


```

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
</head>

<body>
    <div class="container">
        <div class="main">
            <table class="table table-striped table-bordered
            table-hover" id="dataTables-example">
                <tbody>
                    <tr>
                        <td colspan=3>
                            <div class="slidecontainer">
                                <input type="range"
min="1" max="100" value="50" class="slider" id="myRange">
                                <p>speed : <span
id="demo"></span></p>
                            </div>
                        </td>
                    <tr>
                    <tr>
                        <td></td>
                        <td><button data-
oper="go">▲</button></td>
                    <tr>
                    <tr>
                        <td><button data-
oper="left">◀</button></td>
                        <td><button data-
oper="stop">■</button></td>
                        <td><button data-
oper="right">▶</button></td>
                    <tr>
                    <tr>
                        <td><button data-
oper="back">▼</button></td>
                        <td><button data-
oper="back">▼</button></td>
                    <tr>
                    <tr>
                        <td colspan=3><input name="result"
readonly="readonly" /></td>
                    <tr>
                </tbody>
            </table>
        </div>
    </div>
    <script src="https://code.jquery.com/jquery-3.6.3.min.js"></script>
    <script>
        $(document).ready(function () {
            var input_result = $('input[name="result"]');
            var input_speed;

```

```

var order = "stop";

function get(order, callback, error) {
    input_speed = $("#myRange").val();
    $.get("/moving", {state: order, speed: input_speed},
function (result) {
        if (result != "") {
            input_result.val(result);
        }
        console.log(result);
    }).fail(function (xhr, status, er) {
        if (error) {
            error(er);
        }
    });
});

$('button[data-oper="go"]').on("click", function (e) {
    order = "go";
    console.log("go");
    get("go");
});

$('button[data-oper="back"]').on("click", function (e) {
    order = "back";
    console.log("back");
    get(order);
});

$('button[data-oper="left"]').on("click", function (e) {
    order = "left";
    console.log("left");
    get(order);
});

$('button[data-oper="right"]').on("click", function (e) {
    order = "right";
    console.log("right");
    get(order);
});

$('button[data-oper="stop"]').on("click", function (e) {
    order = "stop";
    console.log("stop");
    get(order);
});

$("#myRange").change(function () {
    get(order);
});

var slider = document.getElementById("myRange");
var output = document.getElementById("demo");
output.innerHTML = slider.value;

slider.oninput = function () {

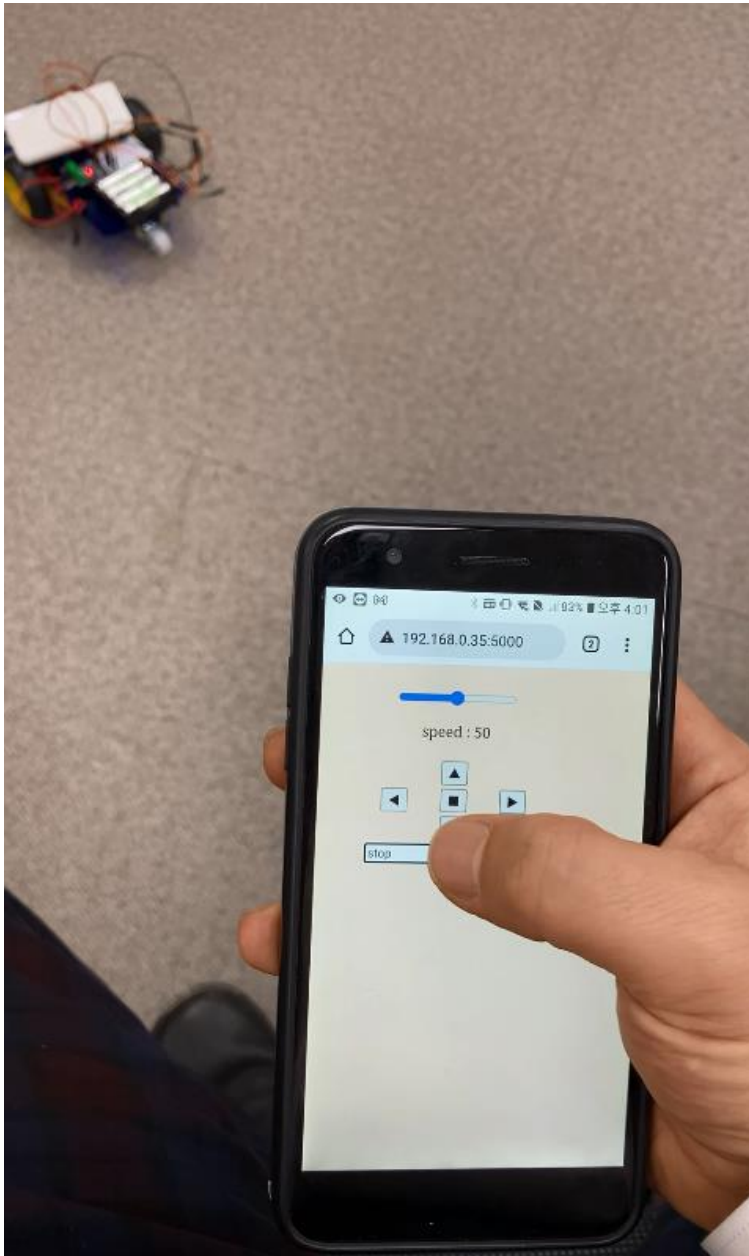
```

```

        output.innerHTML = this.value;
    });
</script>
</body>
</html>

```

// 구동 모습,



// 참조

https://github.com/ntrexeng/Blue_RC_Car/blob/master/Blue_RC_Car/Blue_RC_Car.ino

<https://flask.palletsprojects.com/en/2.2.x/quickstart/#a-minimal-application>

<https://projects.raspberrypi.org/en/projects/python-web-server-with-flask>

<https://learn.microsoft.com/ko-kr/windows/iot-core/learn-about-hardware/pinmappings/pinmappingsrpi>

<https://snyk.io/advisor/python/Flask/functions/flask.request.args>

한울직업전문학교 정재호 교사