

라즈베리파이를 이용한 위험지역 반려동물 감시 카메라 제안서

2071477 c반 박경민

1. 작품개요

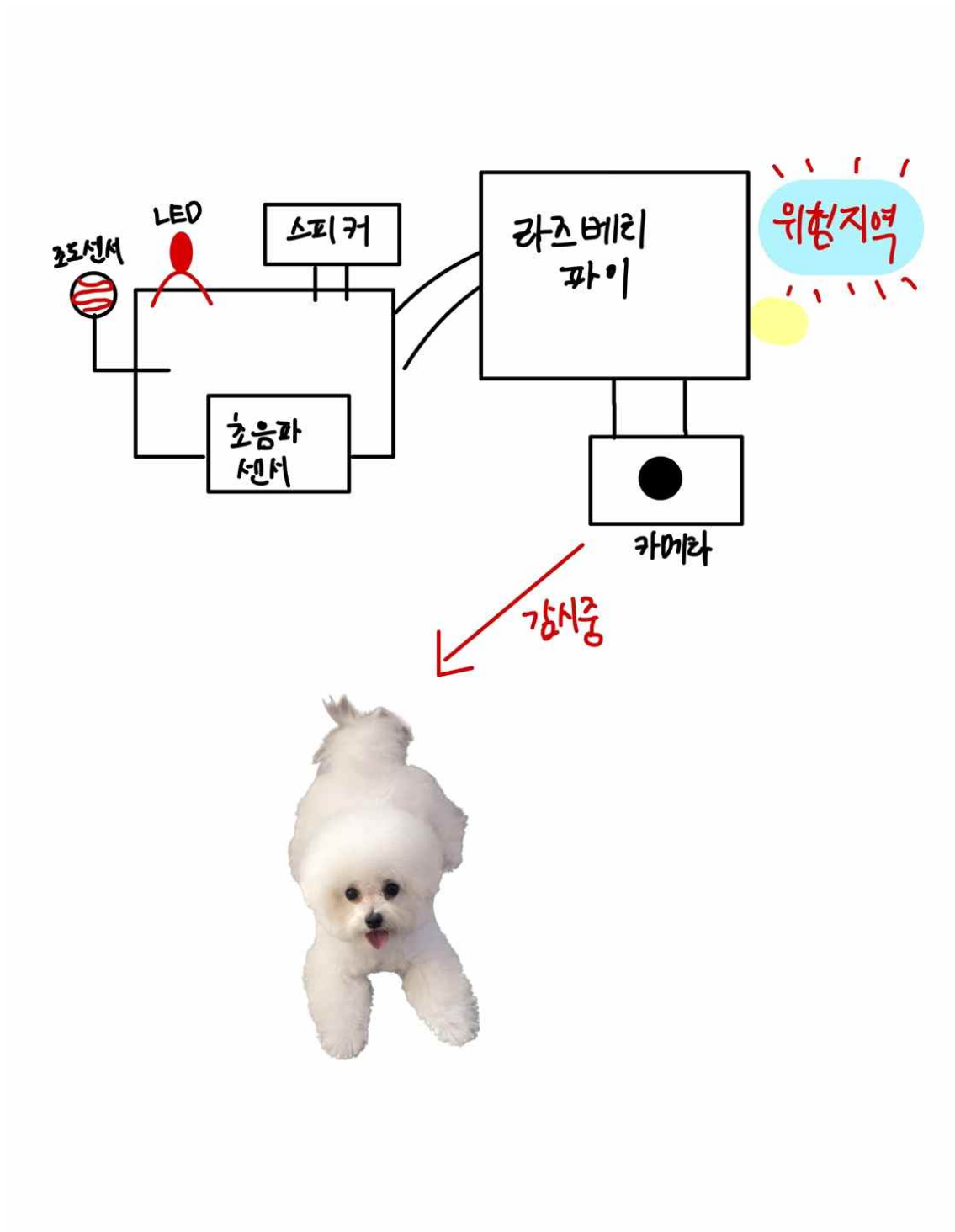
일반 4인가구 이상 가정집 뿐만 아니라 1인가구 시대가 점차 많아지면서 사람들은 반려동물과 함께 지내는 가구가 많아졌습니다. 저 또한 반려동물을 키우는 입장에서 반려동물 보호자들은 본인이 집에 없거나 반려동물을 계속 주시하고 있을 수는 없기 때문에 반려동물에게 무슨일이 생기진 않을까 걱정을 할 수 있고, 실제로도 위험한 물건을 건드린다거나 하면 반려동물에게 자칫 큰 사고가 발생할 수 있습니다.

이러한 이유로 인해 저는 이 프로젝트를 계획 하게 되었습니다.

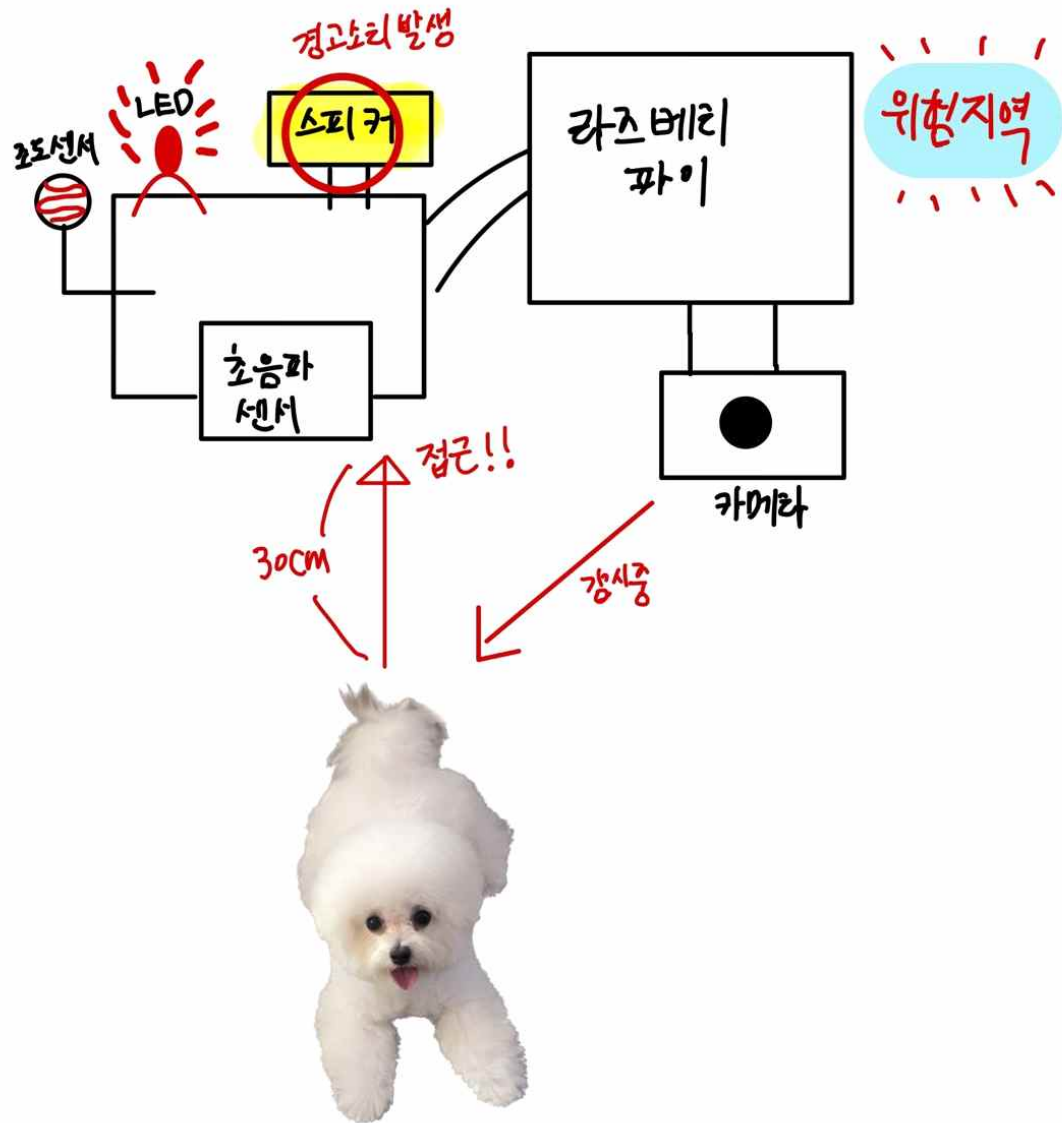
이 라즈베리파이를 이용한 프로젝트에 필요한 것들은 먼저 반려동물에게 위험한 물건이 있는 곳에 라즈베리 파이를 설치합니다. 라즈베리파이는 항상 그 자리에서 카메라를 통해 반려동물을 감시하면서 반려동물이 위험지역으로 가까이 접근을 하게되면 LED 가 켜지고 블루투스 스피커에서 경고음을 울려 반려동물이 접근 하지 못하게 합니다.

접근을 하게 되면 라즈베리 파이는 보호자에게 경고 메시지를 보내주고, 반려동물이 그 장소에서 벗어난다면 안전하다는 메시지를 보내줍니다. 하지만 반려동물이 일정시간이상 그 자리에서 떠나지 않는다면 다시한번 보호자에서 위험 메시지를 보내주어 보호자에서 위험함을 상기시켜주도록 합니다.

2. 시스템 구조



< 평상시 상황 >



< 반려동물 접근 시 상황 >

2. 구현방법

2.1 하드웨어 부분

(각각 한 개씩 사용 합니다. - 블루투스 스피커 별도 구매)

- 라즈베리 파이
- 카메라
- 온 습도센서
- 조도센서
- 초음파센서
- 블루투스 스피커
- LED

* LED

led_red - GPIO 6 #뽕판하단 16

led_yello - GPIO 18 # 뽕판하단 18

* 초음파

trig = GPIO20 #뽕판 상단 19

echo = GPIO16 #뽕판 상단 18

* 온-습도

sda = GPIO 2

scl = GPIO 3

* 조도

5VD #뽕판 하단

SPISCLK = GPIO12

SPIMOSI = GPIO10

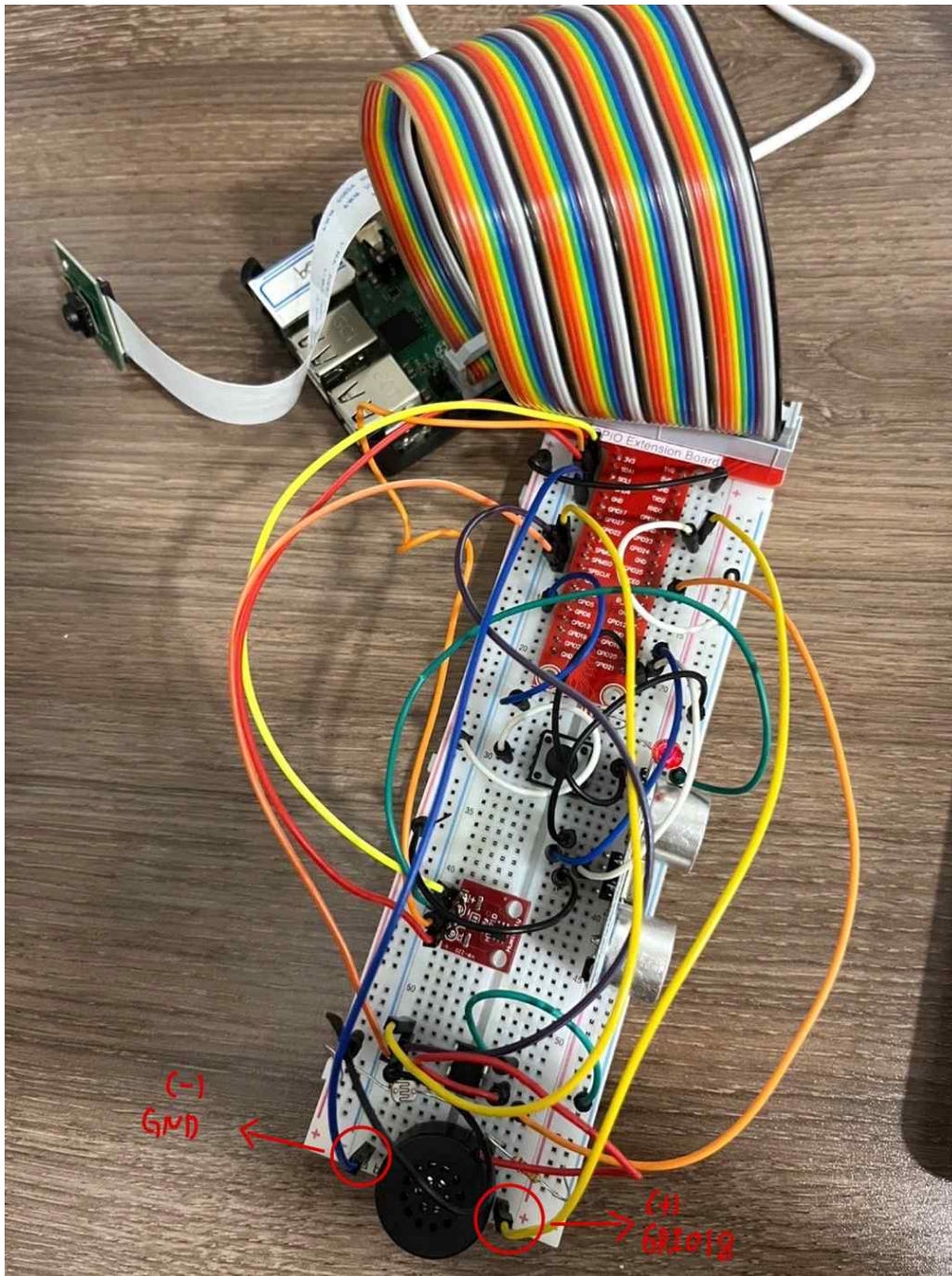
SPIMISO = GPIO11

* 스피커 (피에조 부조 사용)

+극 : GPIO18

-극 : GND

< 하드웨어 부분 사진 (아래) >



2.2 소프트웨어 부분

1. 플라스크 앱(myapp.py)

```
myapp.py > ...
1  from flask import Flask, render_template, request
2  app = Flask(__name__)
3
4  @app.route('/')
5  def index():
6      return render_template('my.html')
7
8  if __name__ == "__main__":
9      app.run(host='0.0.0.0', port=8080)
10
```

2. mqtt를 이용하여 전송하는 파이썬 코드(mqtt.py)

```
mqtt.py > ...
1  #mqtt.py
2
3  import time
4  import paho.mqtt.client as mqtt
5  import sensor
6
7  flag = False
8
9
10 broker_ip = "localhost"
11 client = mqtt.Client()
12
13 client.connect(broker_ip, 1883)
14 client.loop_start()
15
16 def on_connect(client, userdata, flag, rc):
17     client.subscribe("face", qos = 0)
18
19 def on_message(client, userdata, msg) :
20     global flag
21     command = msg.payload.decode("utf-8")
22     if command == "action" :
23         print("action msg received..")
24         flag = True
25
26 client.on_connect = on_connect
27 client.on_message = on_message
28
29
30
31 while True:
32     light = sensor.getLight()
33     client.publish("light",light,qos=0)
34     distance = sensor.measureDistance()
35     client.publish("distance",distance,qos=0)
36     if distance >= 15:
37         sensor.ledOut()
38         sensor.buzzerOnOff()
```

```
mqtt.py > ...
37         sensor.ledOut()
38         sensor.SpeakerOff()
39         client.publish("safe",1,qos=0)
40         imageFileName = sensor.takePicture()
41         client.publish("image",imageFileName,qos=0)
42     else:
43         sensor.ledIn()
44         sensor.Speaker()
45         imageFileName = sensor.takePicture()
46         client.publish("image",imageFileName,qos=0)
47         time.sleep(0.1)
48     print("time...",end=" ")
49     print(flag)
50     client.loop_end()
51     client.disconnect()
52
```

3. 센서를 제어하는 파이썬 코드(sensor.py)


```
sensor.py > ...
1  #sensor.py
2
3  import RPi.GPIO as GPIO
4  import os; import io; import time
5  import picamera
6  import cv2; import numpy as np
7  import Adafruit_MCP3008
8  import busio
9  from adafruit_htu21d import HTU21D
10
11  trig = 20
12  echo =16
13
14  GPIO.setmode(GPIO.BCM)
15  GPIO.setwarnings(False)
16  GPIO.setup(trig,GPIO.OUT)
17  GPIO.setup(echo,GPIO.IN)
18  GPIO.output(trig,False)
19
20  sda=2
21  scl=3
22
23  i2c=busio.I2C(scl,sda)
24  sensor=HTU21D(i2c)
25  mcp = Adafruit_MCP3008.MCP3008(clk=11,cs=8,miso=9,mosi=10)
26
27  fileName = ""
28  stream = io.BytesIO()
29  camera = picamera.PiCamera()
30  camera.start_preview()
31  camera.rotation = 180
32  camera.resolution = (640,480 )
33  camera.framerate = 15
34  camera.resolution = (320, 240)
35  time.sleep(1)
36  mcp=Adafruit_MCP3008.MCP3008(clk=11,cs=8,miso=9,mosi=10)
37
38  led_red = 6
```



```
sensor.py > ...
37
38 led_red = 6
39 led_yello =19
40 buzzer = 18
41 GPIO.setup(led_red,GPIO.OUT)
42 GPIO.setup(led_yello,GPIO.OUT)
43 GPIO.setup(buzzer,GPIO.OUT)
44 pwm = GPIO.PWM(buzzer,262)
45
46
47
48 def getLight():
49     if mcp.read_adc(0)>200:
50         return "밝음"
51     else:
52         return "밝지않음"
53
54 def Speaker():
55     pwm.start(50.0)
56     time.sleep(1.5)
57     pwm.stop()
58
59 def SpeakerOff():
60     pwm.stop()
61
62 def measureDistance():
63     global trig,echo
64     time.sleep(0.5)
65     GPIO.output(trig,True)
66     time.sleep(0.0001)
67     GPIO.output(trig,False)
68
69     while(GPIO.input(echo)==0):
70         # pass
71         pulse_start = time.time()
72     while(GPIO.input(echo)==1):
73         # pass
74         pulse_end=time.time()
```

```
sensor.py > measureDistance
72 while(GPIO.input(echo)==1):
73     # pass
74     pulse_end=time.time()
75
76     pulse_duration = pulse_end - pulse_start
77     return 340*100/2*pulse_duration
78
79
80 def ledIn():
81     onOff = 1
82     controlLED(led_red , onOff)
83     controlLED(led_yello,onOff)
84
85 def controlLED(Led,onOff):
86     GPIO.output(led_red,onOff)
87     GPIO.output(led_yello,onOff)
88
89
90 def ledOut():
91     onOff = 0
92     controlLED(led_red,onOff)
93     controlLED(led_yello,onOff)
94
95
96 def takePicture() :
97     global fileName, stream, camera
98
99     if len(fileName) != 0:
100         os.unlink(fileName)
101
102     stream.seek(0)
103     stream.truncate()
104     camera.capture(stream, format='jpeg', use_video_port=True)
105     data = np.frombuffer(stream.getvalue(), dtype=np.uint8)
106     image = cv2.imdecode(data, 1)
107     haar = cv2.CascadeClassifier('./haarCascades/haar-cascade-files-master/haarcascade_frontalface_default.xml')
108     image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
109
110
111 sensor.py > takePicture
109 data = np.frombuffer(stream.getvalue(), dtype=np.uint8)
110 image = cv2.imdecode(data, 1)
111 haar = cv2.CascadeClassifier('./haarCascades/haar-cascade-files-master/haarcascade_frontalface_default.xml')
112 image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
113 faces = haar.detectMultiScale(image_gray,1.1,3)
114 for x, y, w, h in faces:
115     cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)
116
117     takeTime = time.time()
118     fileName = "./static/%d.jpg" % (takeTime * 10)
119     cv2.imwrite(fileName, image)
120     return fileName
121
122
123 if __name__ == '__main__':
124     while(True):
125         name = takePicture()
126         distance = measureDistance()
127         if mcp.read_adc(0)>200:
128             print("밝음")
129         elif mcp.read_adc(0)>130 and mcp.read_adc(0)<180 :
130             print("황지않음")
131         else:
132             print("어두움")
133             print("distance=%f" % distance)
```

4. 자바 스크립트 파일(mqttio.js)

```

JS mqttio.js > ...
1  var port = 9001 // mosquito의 디폴트 웹 포트
2  var client = null; // null이면 연결되지 않았음
3  var canvas;
4  var context;
5  var img;
6
7  window.addEventListener("load", function() {
8      canvas = document.getElementById("myCanvas");
9      context = canvas.getContext("2d");
10
11      img = new Image();
12      img.onload = function () {
13          context.drawImage(img, 0, 0, 700, 500); // (0,0) 위치에 img의 크기로 그리기
14      }
15  });
16
17  // drawImage()는 "image" 토픽이 도착하였을 때 onMessageArrived()에 의해 호출된다.
18  function drawImage(imgUrl) { // imgUrl은 이미지의 url
19      img.src = imgUrl; // img.onload에 등록된 코드에 의해 그려짐
20  }
21
22  var isImageSubscribed = false;
23
24  function recognize() {
25      if(!isImageSubscribed) {
26          subscribe('image'); // 토픽 image 등록
27          isImageSubscribed = true;
28      }
29      publish('face', 'action'); // 토픽: facerecognition, start 메시지 전송
30  }
31
32  function startConnect() { // 접속을 시도하는 함수
33      clientId = "clientID-" + parseInt(Math.random() * 100); // 랜덤한 사용자 ID 생성
34
35      // 사용자가 입력한 브로커의 IP 주소와 포트 번호 알아내기
36      broker = document.getElementById("broker").value; // 브로커의 IP 주소
37
38      // mqtt.js의 mqtt 객체에 브로커의 주소와 포트 번호 입력

```

```

35 mqttio.js > startConnect
37
38 // id가 message인 DIV 객체에 브로커의 IP와 포트 번호 출력
39 // MQTT 메시지 전송 기능을 모두 가진 Paho client 객체 생성
40 client = new Paho.MQTT.Client(broker, Number(port), clientId);
41
42 // client 객체에 콜백 함수 등록
43 client.onConnectionLost = onConnectionLost; // 접속이 끊어졌을 때 실행되는 함수 등록
44 client.onMessageArrived = onMessageArrived; // 메시지가 도착하였을 때 실행되는 함수 등록
45
46 // 브로커에 접속. 매개변수는 객체 {onSuccess : onConnect}로서, 객체의 프로퍼티는 onSuccess이고 그 값이 onConnect.
47 // 접속에 성공하면 onConnect 함수를 실행하라는 지시
48 client.connect({
49   onSuccess: onConnect,
50 });
51
52
53 var isConnected = false;
54
55 // 브로커로의 접속이 성공할 때 호출되는 함수
56 function onConnect() {
57   isConnected = true;
58   document.getElementById("messages").innerHTML += '<span>반려동물 시스템이 연결되었습니다</span><br/>';
59   subscribe("light");
60   subscribe("safe");
61 }
62
63
64 var topicSave;
65 function subscribe(topic) {
66   if(client == null) return;
67   if(isConnected != true) {
68     topicSave = topic;
69     window.setTimeout("subscribe(topicSave)", 500);
70     return
71   }
72
73   // 토픽으로 subscribe 하고 있음을 id가 message인 DIV에 출력
74   // document.getElementById("messages").innerHTML += '<span>subscribe to: ' + topic + '</span><br/>';

```

```

35 mqttio.js > subscribe
75
76 client.subscribe(topic); // 브로커에 subscribe
77
78
79 function publish(topic, msg) {
80   if(client == null) return; // 연결되지 않았음
81   client.send(topic, msg, 0, false);
82 }
83
84 function unsubscribe(topic) {
85   if(client == null || isConnected != true) return;
86
87   // 토픽으로 subscribe 하고 있음을 id가 message인 DIV에 출력
88   document.getElementById("messages").innerHTML += '<span>Unsubscribing to: ' + topic + '</span><br/>';
89
90   client.unsubscribe(topic, null); // 브로커에 subscribe
91 }
92
93 // 접속이 끊어졌을 때 호출되는 함수
94 function onConnectionLost(responseObject) { // 매개변수인 responseObject는 응답 패킷의 정보를 담은 객체
95   // document.getElementById("messages").innerHTML += '<span>오류 : 접속 끊어짐</span><br/>';
96   if (responseObject.errorCode != 0) {
97     document.getElementById("messages").innerHTML += '<span>오류 : ' + responseObject.errorMessage + '</span><br/>';
98   }
99 }
100
101 // 메시지가 도착할 때 호출되는 함수
102 function onMessageArrived(msg) { // 매개변수 msg는 도착한 MQTT 메시지를 담고 있는 객체
103   console.log("onMessageArrived: " + msg.payloadString);
104
105   // 토픽 image가 도착하면 payload에 담긴 파일 이름의 이미지 그리기
106   if(msg.destinationName == "image") {
107     drawImage(msg.payloadString); // 메시지에 담긴 파일 이름으로 drawImage() 호출. drawImage()는 웹 페이지에 있음
108     document.getElementById("pet").innerHTML += '<span>반려동물을 감시중입니다 </span><br/>';
109   }
110 }
111
112 if(msg.destinationName == "safe") {

```

```

110     }
111
112     if(msg.destinationName == "safe") {
113
114         document.getElementById("pet").innerHTML = '<span>반려동물은 지금 안전합니다 :) </span><br/>';
115     }
116 }
117
118
119 // disconnection 버튼이 선택되었을 때 호출되는 함수
120 function startDisconnect() {
121     client.disconnect(); // 브로커에 접속 해제
122     document.getElementById("messages").innerHTML += '<span>반려동물관리시스템 서비스 종료</span><br/>';
123 }
124

```

4. html 코드(my.html)

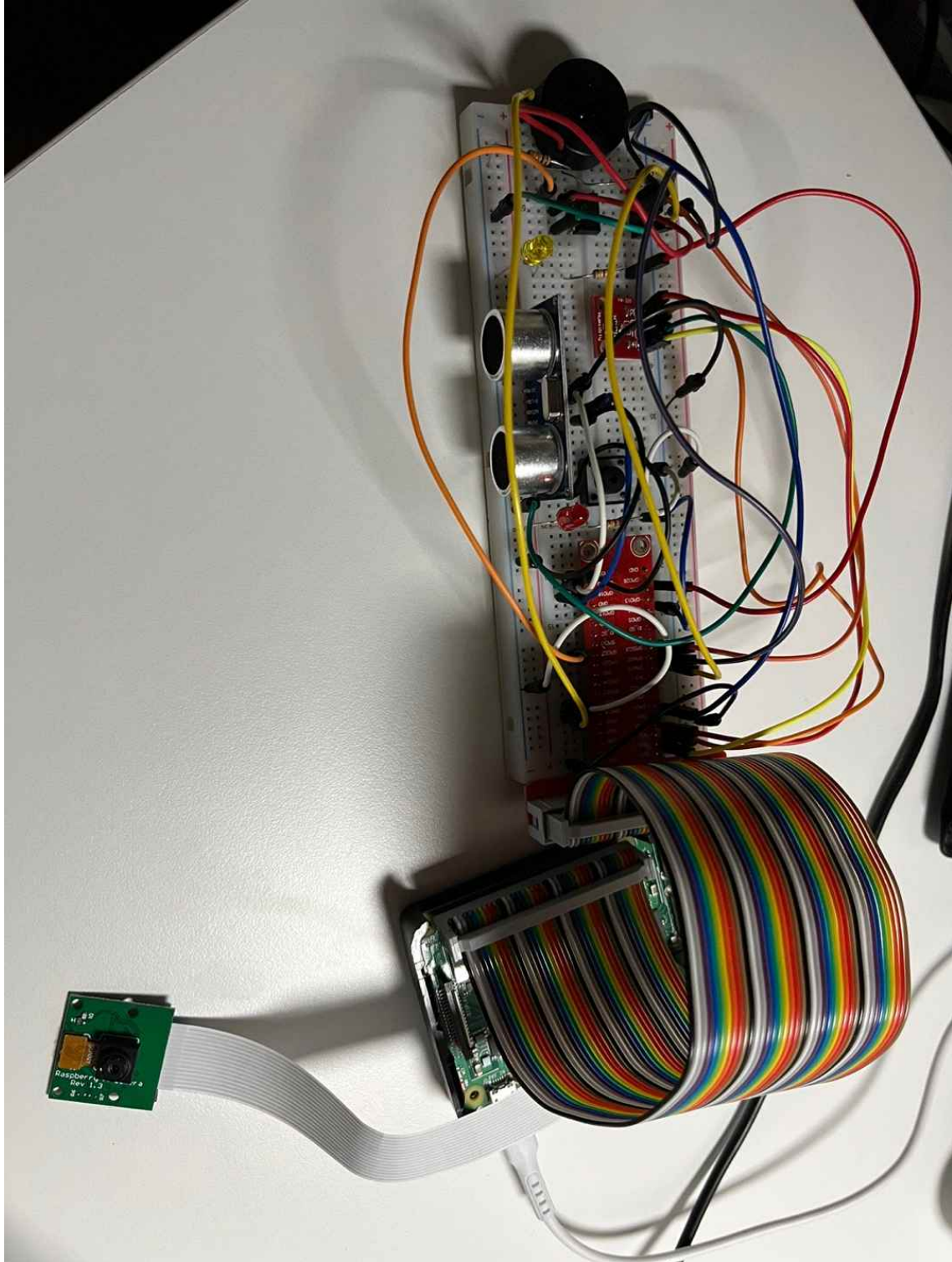
```

<? my.html >...
1  <!doctype html>
2  <html>
3  <head>
4      <link href='ico/favicon.ico' rel='icon'>
5      <meta charset="utf-8">
6      <title>나의 반려동물 관리 시스템</title>
7      <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.2/mqttws31.min.js" type="text/javascript"></script>
8      <script src="/static/mqttio.js" type="text/javascript"></script>
9      <script>
10         window.addEventListener("load", function () {
11             // http://224..129:8080/에서 224...의 IP만 끌어내는 코드
12             var url = new String(document.location);
13             ip = (url.split("/")[1]); // ip = "224...:8080/"
14             ip = (ip.split(":")[0]); // ip = "224..."
15             document.getElementById("broker").value = ip
16         });
17     </script>
18     <style>
19         canvas {background-color: lightblue}
20     </style>
21 </head>
22 <body style = "background-color: orange;">
23     <center>
24         <h1>나의 반려동물 관리 프로그램</h1>
25         <hr>
26         <form id="connection-form">
27             <b>반려동물 주인의 IP 주소 입력:</b>
28             <input id="broker" type="text" name="broker" value=""><br>
29             <input type="button" onclick="startConnect()" value="관리시스템연결">
30             <input type="button" onclick="startDisconnect()" value="관리시스템연결해제">
31         </form>
32     <hr>
33     <form id="subscribe-form">
34         <input type="button" onclick="recognize()" value="관리시스템시작">
35     </form>
36
37     <div style = "color: Green" id = "messages"></div>
38     <hr>
39     <div style = "color: Red" id = "pet"></div>
40     <div style = "color: Red" id = "pet"></div>
41     <hr>
42 </body>
43 </html>
44

```


3. 구현 내용

3.1 최종 회로도

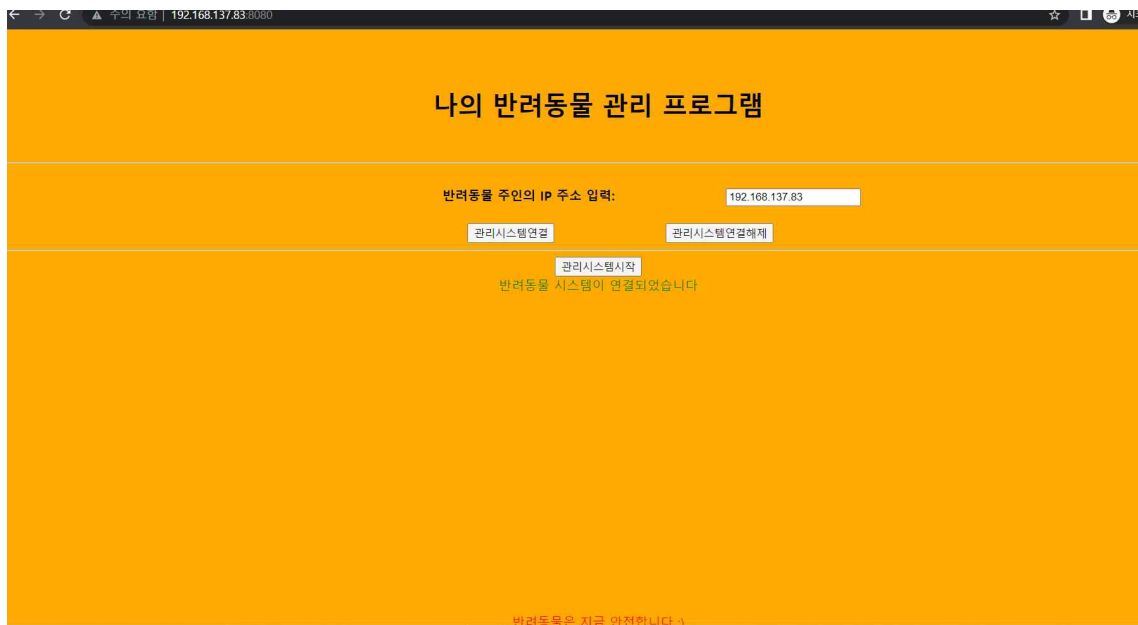


3.2 프로그램 코드 및 설명 /4. 실행 과정 및 결과

- 위 소프트웨어 부분(2.2) 에 게시

4.1 초기

* 웹페이지의 관리시스템연결 버튼을 눌렀을 때 화면
“반려동물 시스템이 연결되었습니다” 라는 문구가 뜬다.



* 웹페이지의 관리시스템시작 버튼을 누르면 카메라가 켜지고 반려동물을 감시하게 된다.
물체가 거리 15이상인 상태에서는 “반려동물은 지금 안전합니다” 와 “반려동물을 감시중입니다” 라는 문구가 번갈아가면서 뜨게된다.

나의 반려동물 관리 프로그램

반려동물 주인의 IP 주소 입력:

192.168.137.83

관리시스템연결

관리시스템연결해제

관리시스템시작

반려동물 시스템이 연결되었습니다



반려동물은 지금 안전합니다 :)

나의 반려동물 관리 프로그램

반려동물 주인의 IP 주소 입력:

192.168.137.83

관리시스템연결

관리시스템연결해제

관리시스템시작

반려동물 시스템이 연결되었습니다

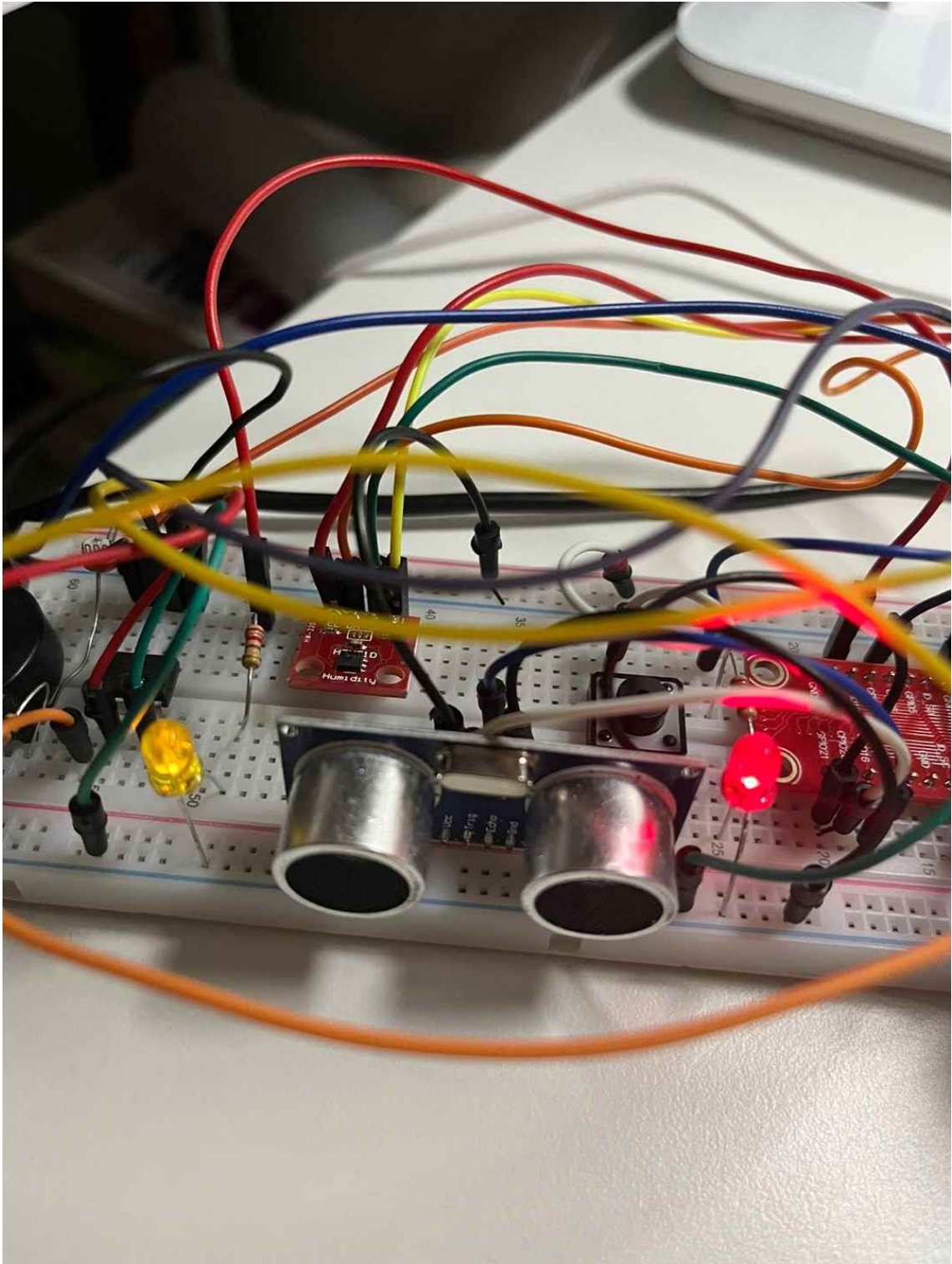


반려동물을 감시중입니다

4.2 반려동물이 접근한 경우

*반려동물이 15이하로 접근하였을 때 빨강과 노란색 LED가 켜지고 피에조부조 스피커가 울려 반려동물에게 경고 소리를 내준다.

“반려동물은 안전합니다” 문구는 이 시간동안은 뜨지 않게 되고, 이로 인해 보호자에서 반려동물이 위험하다는 상황을 인지하게 해준다.



나의 반려동물 관리 프로그램

반려동물 주인의 IP 주소 입력:

192.168.137.83

관리시스템연결

관리시스템연결해제

관리시스템시작

반려동물 시스템이 연결되었습니다



반려동물을 감시중입니다

4.3 연결을 해제했을 경우

*관리시스템 연결 해제 버튼을 누르면 연결이 해제 되게 된다.

나의 반려동물 관리 프로그램

반려동물 주인의 IP 주소 입력:

192.168.137.83

관리시스템연결

관리시스템연결해제

관리시스템시작

반려동물 시스템이 연결되었습니다
반려동물관리시스템 서비스 종료



반려동물은 지금 안전합니다 :)

5. 결론

저는 편입생으로 3학년이지만 전공필수 과목인 모바일 시스템과목을 남들보다는 늦게 수강하게 되었는데, 프로젝트 전까지 여러 과제들을 수행하면서도 사실 프로그램이 돌아가는 코드를 확실하게 이해한다는 느낌을 못 받았었는데 프로젝트를 진행하면서 시스템들이 어떻게 연결되고 어떤 방식으로 돌아가는지 다시 한번 천천히 공부하게 되면서 코드를 완전하게 내 것으로 만들 수 있었던 것 같고, 처음에 프로젝트 진행하면서 코드를 만들며 진행할 때는 이러한 이유 때문에 어려움을 겪었지만, 교수님께서 올려주신 여러 자료들을 참고하고 또 따로 찾아보면서 새로운 지식도 습득하게 되면서 이러한 어려움들을 잘 해결할 수 있었던 것 같습니다. 저한테 이번학기 첫 프로젝트였었는데, 프로젝트를 하면서 처음 생각한대로 완전히 이루어지지 않고 여러 변수가 생길 수 있다는 것을 깨달았고, 충분히 시간이 주어진다고 하여도 프로젝트 준비과정에서 조금더 여러 경우를 생각하면서 계획을 짜야겠다는 생각을 갖게 되었습니다. 그리고 저는 웹트랙이 아니어서 html을 다루는데 어려움이 있었는데, 컴퓨터공학 전공자로서 여러 프로그램을 언제 어디서 다루게 될지 모를 수 있다는 사실을 많이 깨달았고, 시간이 있을 때 웹을 포함한 여러 공부를 해야겠다고 생각했습니다.

