

2020년 1학기 클라우드 컴퓨팅  
(IoT 클라우드 서비스 개발)

프로젝트 보고서

제목 : NodeMCU 와 Firebase를 이용한  
클라우드 기반 IoT 응용 서비스 개발

성명 : 설재혁

반 : 2-1

학번 : 201929196



소속 : 동의과학대학교 컴퓨터정보과

지도교수 : 김종현

## 1. 클라우드 기반 IoT 원격 LED 제어하기

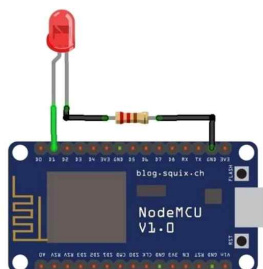
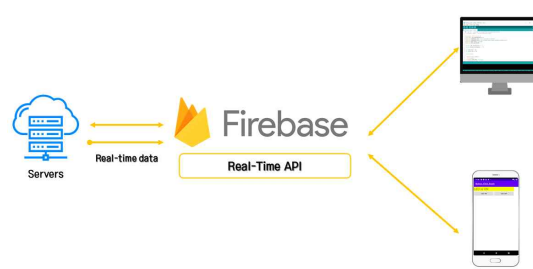
### 1.1 개발 환경

“클라우드 기반 IoT 원격 LED 제어하기” 수행에 사용된 개발 환경은 아래와 같습니다.

	
Arduino	FireBase

### 1.2 개발 내용

- 전체 시스템 구조, 전체 시스템 작동 원리

	
전체 시스템 회로 구성도	전체 시스템 작동 원리

### 1.3 개발 결과물

- 중요 소스코드

```
1 // 아두이노 Firebase 라이브러리
2 #include <ESP8266WiFi.h>
3 #include <FirebaseArduino.h>
4
5 // 데이터베이스 호스트 이름
6 #define FIREBASE_HOST "fir-led-control-5b0e3"
7 // 데이터베이스 비밀번호
8 #define FIREBASE_AUTH "dz8tzaw5AD0qQfPf5qqECjmiEqNt63aE2mf9kYXj"
9 // Wi-Fi 설정
10 #define WIFI_SSID "iptime"
11 #define WIFI_PASSWORD ""
12
13 String red_fireStatus = ""; //Firebase DB로부터 받은 값 저장
14 String blue_fireStatus = ""; //Firebase DB로부터 받은 값 저장
15
16 void setup()
17 {
18   Serial.begin(9600); //통신 속도 설정
19   delay(1000); // 지연 시간 설정
20   pinMode(red_led, OUTPUT); //red_led를 OUTPUT으로 지정
21   pinMode(blue_led, OUTPUT); //blue_led를 OUTPUT으로 지정
```

```

34 Serial.println();
35 Serial.print("Connected to ");
36 Serial.println(WIFI_SSID);
37 Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); //Firebase에 연결
38 Firebase.setString("RED_LED_STATUS", "OFF"); //RED_LED의 초기 LED 상태를 OFF로 설정
39 Firebase.setString("BLUE_LED_STATUS", "OFF"); //BLUE_LED의 초기 LED 상태를 OFF로 설정

42 void loop()
43 {
44   red_fireStatus = Firebase.getString("RED_LED_STATUS"); //Firebase 데이터베이스로부터 RED_LED의 LED 상태 값 읽음
45   if (red_fireStatus == "ON") { // FireBase 데이터베이스로부터 받은 LED 상태 값이 "ON"이면 LED 켜기
46     Serial.println("RED LED Turned ON"); // Serial Monitor에 "Led Turned ON" 문자열 출력
47     digitalWrite(red_led, HIGH); // red_led LED ON
48   }
49   else if (red_fireStatus == "OFF") { //Firebase 데이터베이스로부터 RED_LED의 LED 상태 값 읽음
50     Serial.println("RED LED Turned OFF"); // Serial Monitor에 "Led Turned OFF" 문자열 출력
51     digitalWrite(red_led, LOW); // red_led LED OFF
52   } else {
53     Serial.println("Command Error! Please send ON/OFF");
54   }
55
56   delay(2000);
57
58   blue_fireStatus = Firebase.getString("BLUE_LED_STATUS"); //Firebase 데이터베이스로부터 BLUE_LED의 LED 상태 값 읽음
59   if (blue_fireStatus == "ON") { // FireBase 데이터베이스로부터 받은 LED 상태 값이 "ON"이면 LED 켜기
60     Serial.println("BLUE LED Turned ON"); // Serial Monitor에 "Led Turned ON" 문자열 출력
61     digitalWrite(blue_led, HIGH); // blue_led LED ON
62   }
63   else if (blue_fireStatus == "OFF") { //Firebase 데이터베이스로부터 BLUE_LED의 LED 상태 값 읽음
64     Serial.println("BLUE LED Turned OFF"); // Serial Monitor에 "Led Turned OFF" 문자열 출력
65     digitalWrite(blue_led, LOW); // blue_led LED Off
66   } else {
67     Serial.println("Command Error! Please send ON/OFF");
68   }
69
70   delay(2000);
71 }

```

## 1.4 깃허브 주소

<https://github.com/alwas7767/CloudComputing.git>

## 2. 클라우드 기반 실시간 IoT 온습도 측정 시스템 제작

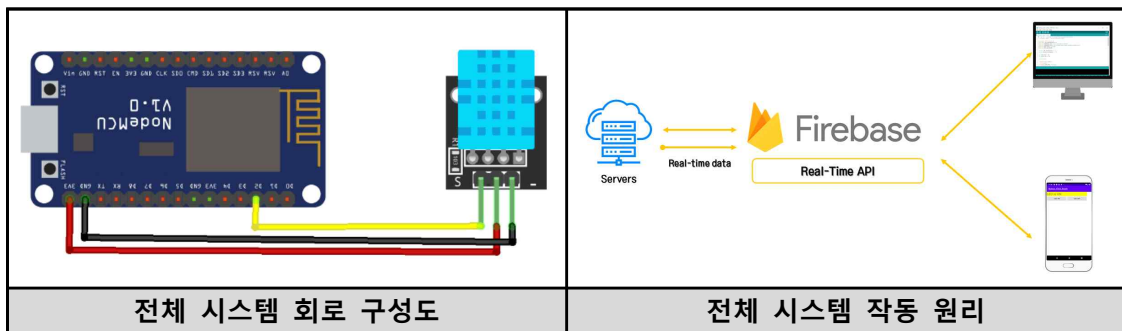
### 1.1 개발 환경

“클라우드 기반 실시간 IoT 온습도 측정 시스템 제작” 수행에 사용된 개발 환경은 아래와 같습니다.

			
Arduino	FireBase	Android Studio	Brackts

### 1.2 개발 내용

- 전체 시스템 구조, 전체 시스템 작동 원리



### 1.3 개발 결과물

```

1 // 아두이노 Firebase 라이브러리
2 #include <ESP8266WiFi.h>
3 #include <WiFiUdp.h>
4 #include <FirebaseArduino.h>
5 #include <DHT.h>
6
7 #define DHTPIN D4
8 #define DHTTYPE DHT11
9 //데이터베이스 호스트 이름
10 #define FIREBASE_HOST "hello-firebase-7bc0f"
11 //데이터베이스 비밀번호
12 #define FIREBASE_AUTH "AcsnvIaMQJ64pvRRKRJVjWp9lEdxZztGg8ANQmWB"
13 //Wi-Fi 설정
14 #define WIFI_SSID "iptime"
15 #define WIFI_PASSWORD ""
16
17 DHT dht(DHTPIN, DHTTYPE);
18
19 void setup() {
20   Serial.begin(9600); // 통신속도 설정
21   // Wi-Fi 연결
22   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
23   dht.begin(9600);
24
25   Serial.print("connecting"); //Serial Monitor에 "connecting" 문자열 출력
26   while (WiFi.status() != WL_CONNECTED) {
27     Serial.print("."); //Serial Monitor에 "." 문자열 출력
28     delay(500); //지연시간 설정
29   }
30   Serial.println();
31   Serial.print("connected: "); //Serial Monitor에 "connected:" 문자열 출력
32   Serial.println(WiFi.localIP());
33
34   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

```

```

39 void loop() {
40   float temp = dht.readTemperature();
41   float humi = dht.readHumidity();
42   Serial.print("Temperature = "); //Serial Monitor에 "Temperature =" 문자열 출력
43   Serial.println(temp);
44   Serial.print("Humidity = "); //Serial Monitor에 "Humidity = " 문자열 출력
45   Serial.println(humi);
46
47   // Firebase.setFloat("temperature", temp);
48   // Firebase.setFloat("humidity", humi);
49
50   // Firebase.pushFloat("temperature", temp);
51   // Firebase.pushFloat("humidity", humi);
52
53   StaticJsonBuffer<200> jsonbuffer;
54   JsonObject& root = jsonbuffer.createObject();
55   root["temperature"] = temp;
56   root["humidity"] = humi;
57   String name = Firebase.push("logDHT", root);
58
59   // handle error
60   if (Firebase.failed()) {
61     Serial.print("pushing /logs failed:");
62     Serial.println(Firebase.error());
63     return;
64   }
65   Serial.println("pushed: /logDHT/");
66   Serial.println(name);
67
68   delay(5000);
69 }

```

## 1.4 깃허브 주소

<https://github.com/alwas7767/CloudComputing.git>

### 3. 개발 후기

**느낀 점** : 이번 학기에는 코로나 19 때문에 수업에 대해 많은 영향이 있어 매우 안타깝다는 생각이 듭니다. 그러나, 아두이노 부품을 통해 많은 시행착오를 겪으면서 하나의 결과물을 만드는 것도 내가 더 나은 사람이 될 수 있는 발걸음이라고 생각하고 열심히 하였습니다. 이번 학기에는 수업 영향이 많아 힘든 수업이었지만, 한 학기를 무사히 잘 마무리할 수 있게 되어 좋습니다.

**향후 개선점** : 클라우드컴퓨팅실습 과제 수행 중 에러가 수도 없이 많이 발생하였습니다. 그러다 보니, 에러 고치는데만 신경을 쓰고 과제 완성도는 신경을 미처 쓰지 못한 것 같아 매우 아쉽다는 생각이 듭니다. 향후에는 과제 완성도와 에러를 둘 다 신경 쓸 수 있도록 하겠습니다.

**추가해서 공부하고 싶은 분야** : FireBase 연동 / FireBase를 이용한 모바일 어플 제작 등에 대해 더 공부해보고 싶습니다.