

IoT 네트워크 labs

---

# 냉장고 프로젝트

---

한예슬

- 01 STEP 01 개발환경 설정
- 02 STEP 02 아두이노 Uno개발
- 03 STEP 03 라즈베리 파이 개발
- 04 STEP 04 웹서버 개발
- 05 STEP 05 DB 개발
- 앱 개발

## Arduino 설치

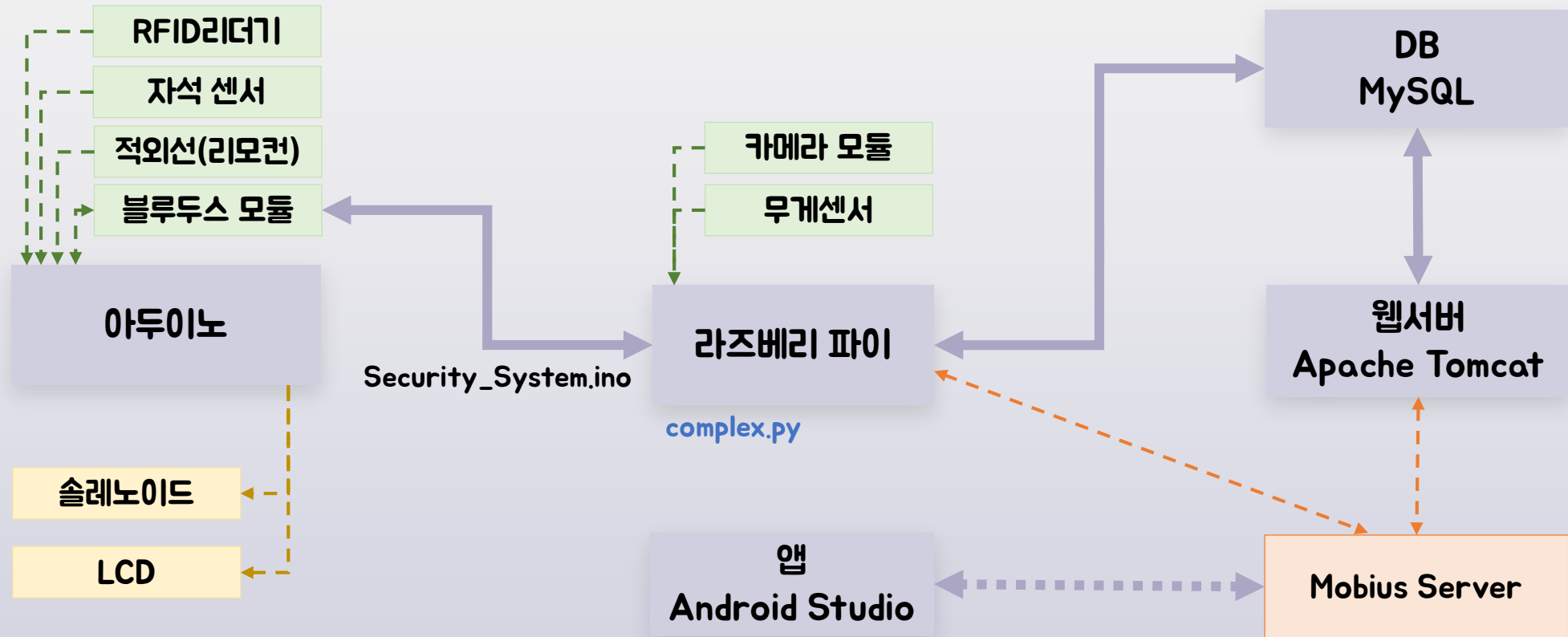
Arduino SW <https://www.arduino.cc/en/main/software>

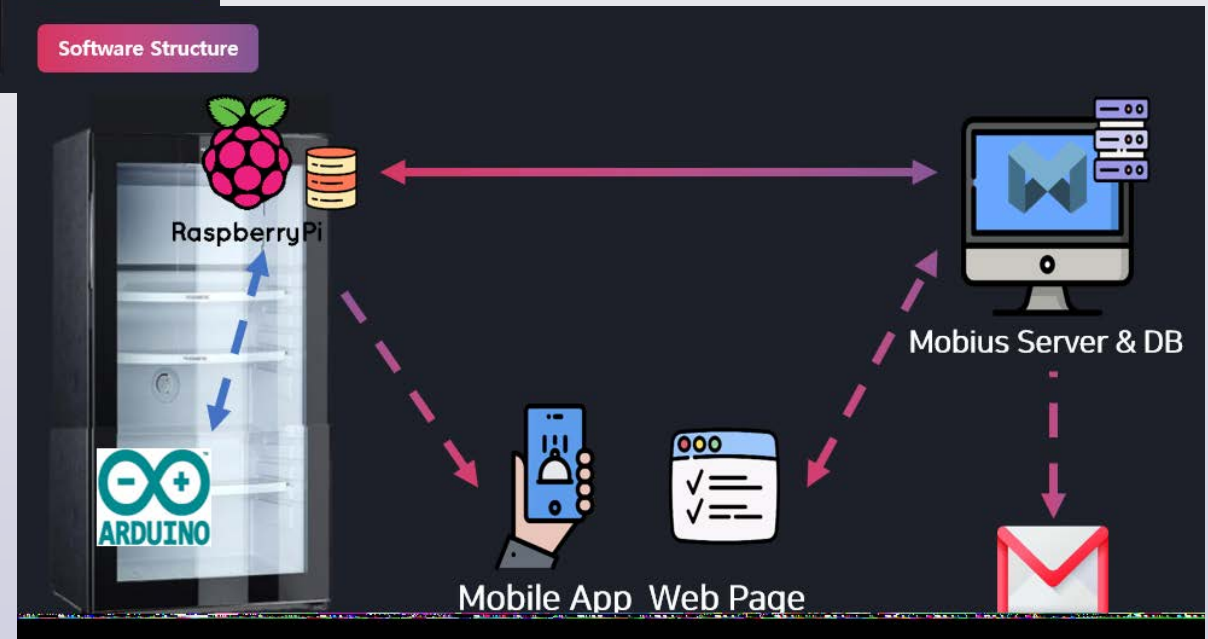
## Raspberry Pi SD카드 포맷

Raspbian <https://www.raspberrypi.org/downloads/raspbian/>

Etcher <https://www.balena.io/etcher/>

가상머신 IP할당 <https://m.blog.naver.com/wideeyed/221087773726>

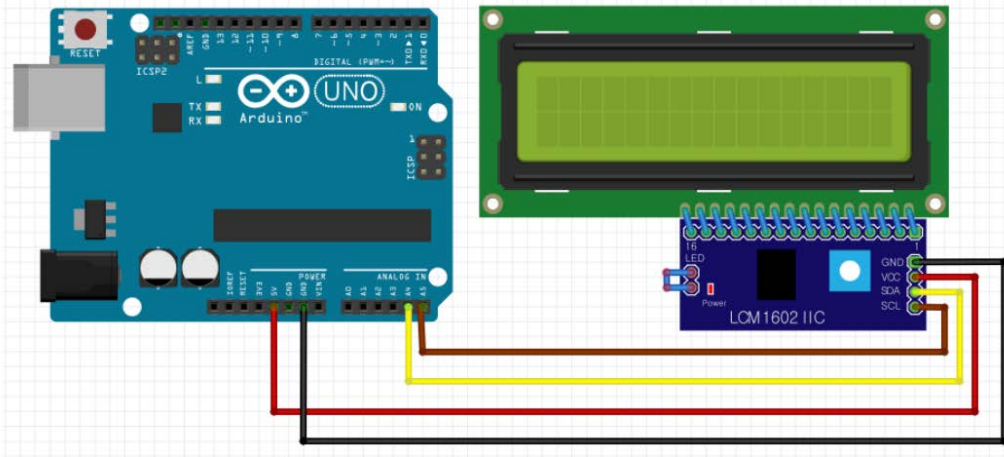




I2C LCD <https://codingrun.com/119>

회로도

- I2C 통신
- 5V



## &lt;I2C LCD 기본 코드:EX15\_I2CLCD.ino&gt;

```
#include <LiquidCrystal_I2C.h> //LiquidCrystal 라이브러리 추가
LiquidCrystal_I2C lcd(0x27, 16, 2); //lcd 객체 선언
```

```
void setup()
{
  lcd.begin(); //LCD 사용 시작
}
```

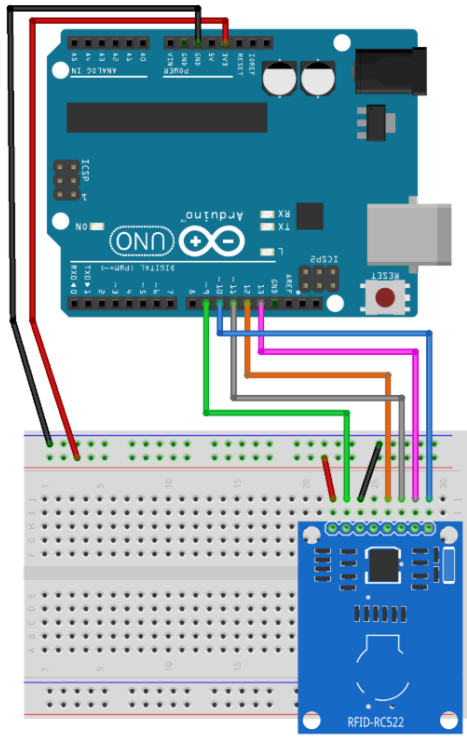
```
void loop()
{
  lcd.setCursor(5, 0); // 커서를 5, 0에 가져다 놓아라. (열, 행)
  lcd.print("Hi ^^"); // 5, 0에 Hi ^^를 출력해라.
  delay(1000); // 1초 대기
  lcd.setCursor(3, 1); // 커서를 3, 1로 가져다 놓아라. (열, 행)
  lcd.print("Codingrun"); // Codingrun을 입력해라.
  delay(1000); // 1초 대기
  lcd.clear(); // 글자를 모두 지워라.
  delay(1000); // 1초 대기
}
```

## RFID 리더기 & 태그 인식하기 <https://blog.naver.com/PostView.nhn?blogId=eduino&logNo=220957847257>

### ■ 아두이노와 연결하기

- 먼저 아래 결선도와 같이 아두이노와 RFID 리더를 연결합니다.

- SPI 통신
- 3.3V



### 자료조사:

RFID와 NFC 차이점 / RFID / NFC / LF(ID) / HF(ID) / UHF(ID) \_

[http://makeshare.org/bbs/board.php?bo\\_table=arduino&wr\\_id=74](http://makeshare.org/bbs/board.php?bo_table=arduino&wr_id=74)

<https://youtu.be/M8BA-phabqE>

### <RFID 기본 코드: RFID\_TEST.ino>

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN 9           // reset핀은 9번으로 설정
#define SS_PIN 10          // SS핀은 10번으로 설정
                          // SS핀은 데이터를 주고받는 역할의 핀( SS = Slave Selector )

MFRC522 mfrc(SS_PIN, RST_PIN);    // MFR522를 사용하기 위해 mfrc객체를 생성해 줍니다.

void setup(){
  Serial.begin(9600);           // 시리얼 통신, 속도는 9600
  SPI.begin();                  // SPI 초기화
                                // (SPI : 하나의 마스터와 다수의 SLAVE(종속적인 역할)간의 통신 방식)

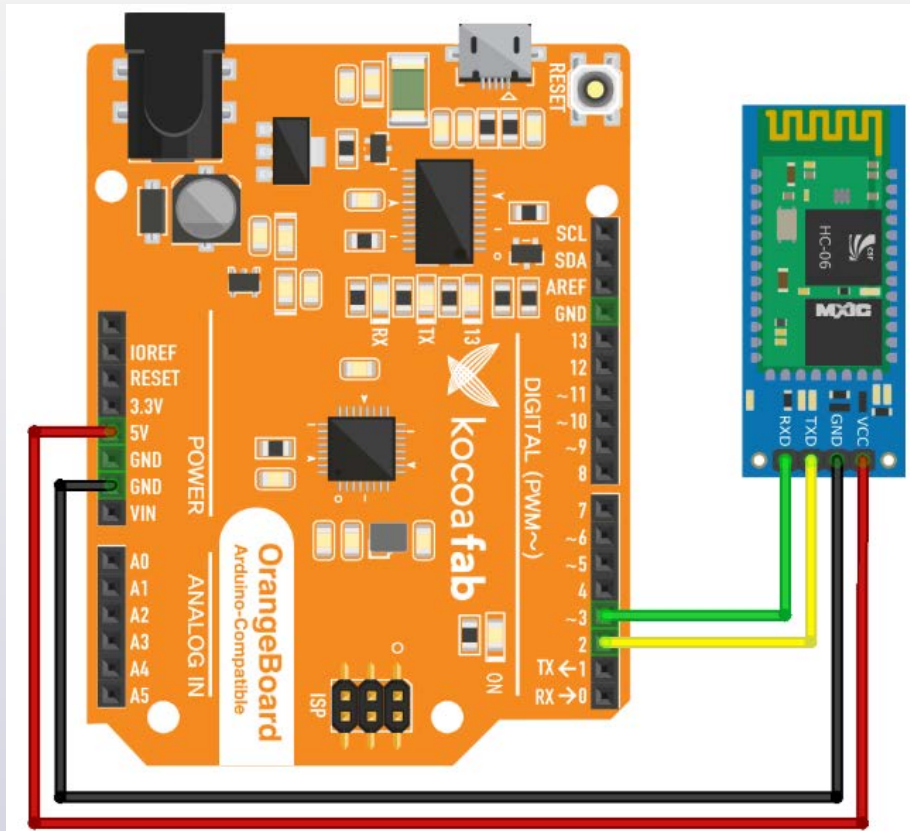
  mfrc.PCD_Init();
}

void loop(){
  if ( !mfrc.PICC_IsNewCardPresent() || !mfrc.PICC_ReadCardSerial() ) {
    // 태그 접촉이 되지 않았을때 또는 ID가 읽혀지지 않았을때

    delay(500);                 // 0.5초 딜레이
    return;                     // return
  }

  Serial.print("Card UID:");    // 태그의 ID출력

  for (byte i = 0; i < 4; i++) { // 태그의 ID출력하는 반복문.태그의 ID사이즈(4)까지
    Serial.print(mfrc.uid.uidByte[i]); // mfrc.uid.uidByte[0] ~ mfrc.uid.uidByte[3]까지 출력
    Serial.print(" ");          // id 사이의 간격 출력
  }
  Serial.println();
}
```

HC-06 <https://kocoafab.cc/tutorial/view/712>

- Serial 통신
- 5V
- AT COMMAND

<HC-06 기본 코드: HC-06\_TEST.ino>

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial mySerial(2, 3); //블루투스의 Tx, Rx핀을 2번 3번핀으로 설정
```

```
void setup() {
  // 시리얼 통신의 속도를 9600으로 설정
  Serial.begin(9600);
  while (!Serial) {
    ; //시리얼통신이 연결되지 않았다면 코드 실행을 멈추고 무한 반복
  }
}
```

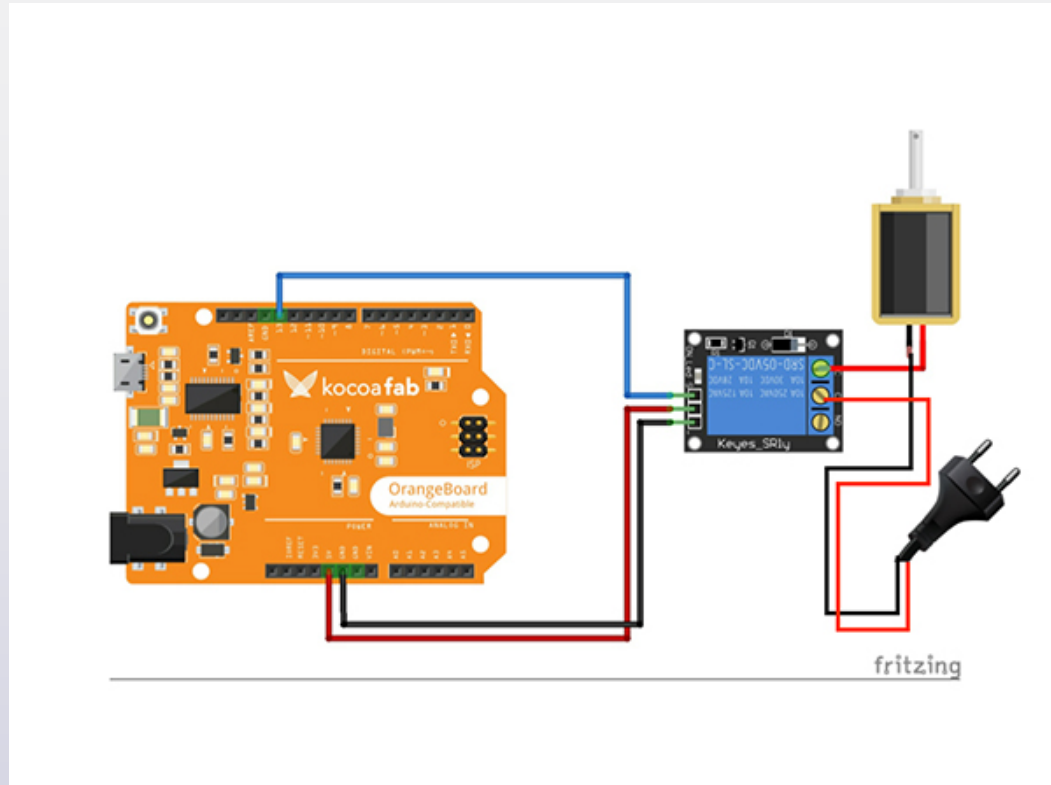
```
Serial.println("Hello World!");
```

```
//블루투스 아두이노의 통신속도를 9600으로 설정
mySerial.begin(9600);
}
```

```
void loop() { //코드를 무한반복합니다.
  if (mySerial.available()) { //블루투스에서 넘어온 데이터가 있다면
    Serial.write(mySerial.read()); //시리얼모니터에 데이터를 출력
  }
  if (Serial.available()) { //시리얼모니터에 입력된 데이터가 있다면
    mySerial.write(Serial.read()); //블루투스를 통해 입력된 데이터 전달
  }
}
```



## 솔레노이드 <https://kocoafab.cc/tutorial/view/345>



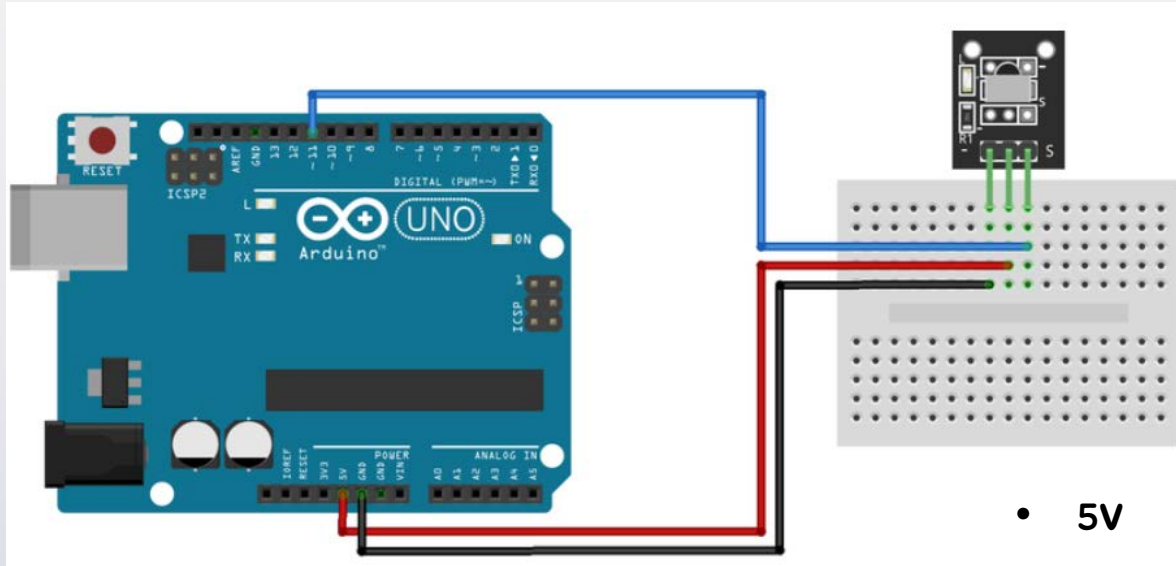
<솔레노이드 기본 코드: Solenoid\_TEST.ino>

```
Int solenoidPin = 5;
```

```
Void setup() {  
  pinMode(solenoidPin, OUTPUT);  
}
```

```
Void loop() {  
  digitalWrite(solenoidPin, HIGH);  
  delay(1000);  
  digitalWrite(solenoidPin, LOW);  
  delay(1000);  
}
```

- 솔레노이드에 따라 5V/12V
- <https://m.blog.naver.com/PostView.nhn?blogId=tkzzang89&logNo=221300992195&proxyReferer=https:%2F%2Fwww.google.co.m%2F>

IR 수신기 & 리모컨 <https://m.blog.naver.com/zeta0807/221047076270>

&lt;IRremote 기본 코드: IRremote\_TEST.ino&gt;

#include &lt;IRremote.h&gt;

int RECV\_PIN = 6;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

void setup()

{

Serial.begin(9600);

// In case the interrupt driver crashes on setup, give a clue

// to the user what's going on.

Serial.println("Enabling IRin");

irrecv.enableIRin(); // Start the receiver

Serial.println("Enabled IRin");

}

void loop() {

if (irrecv.decode(&amp;results)) {

Serial.println(results.value, HEX);

irrecv.resume(); // Receive the next value

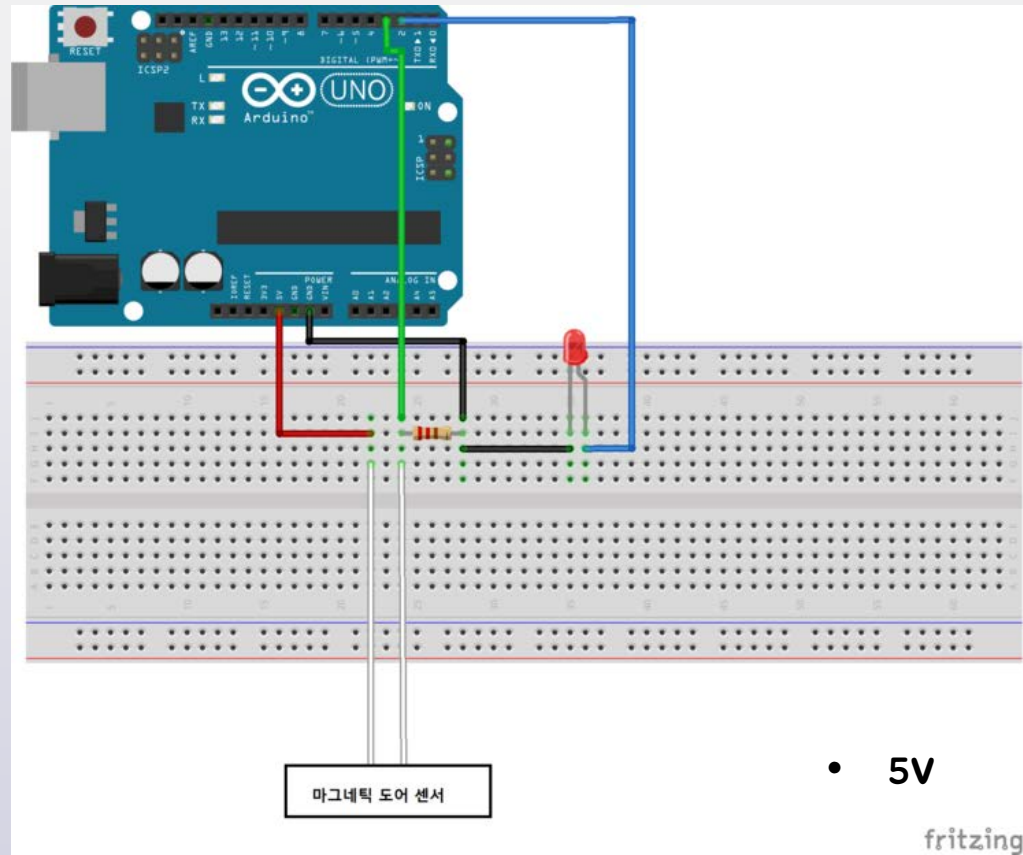
}

delay(100);

}

<https://devicemart.blogspot.com/2019/05/ir.html><https://m.blog.naver.com/PostView.nhn?blogId=nasu0210&logNo=220351153305&proxyReferer=https:%2F%2Fwww.google.com%2F>

마그네틱 센서 <https://m.blog.naver.com/roboholic84/220462617278>



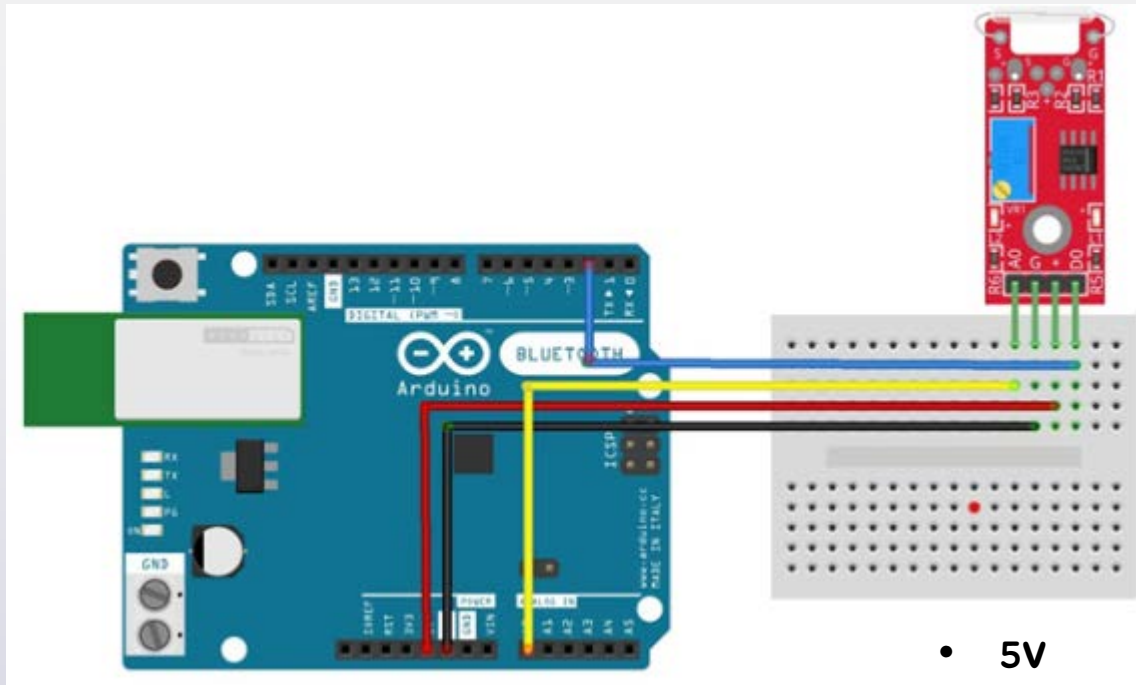
< 마그네틱 센서 기본 코드: Magnetic\_TEST.ino >

```
const int MaganetPin=4;
```

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
  pinMode(MaganetPin,INPUT);  
}
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  int a = digitalRead(MaganetPin);  
  Serial.println(a);  
  delay(100);  
}
```

리드스위치 모듈 [http://makeshare.org/bbs/board.php?bo\\_table=Wiki2&wr\\_id=52](http://makeshare.org/bbs/board.php?bo_table=Wiki2&wr_id=52)



• 5V

<IRremote 기본 코드: Reed\_TEST.ino>

```
void setup() {  
  pinMode(7, INPUT_PULLUP);      // 디지털 3번핀을 입력모드로 설정  
  Serial.begin(9600);            // 시리얼 통신을 시작하며, 속도는 9600으로  
  설정  
}
```

```
void loop() {  
  int a = digitalRead(8);        // 변수 a를 선언하여 디지털 3번핀의 값을 입력  
  Serial.println(a);             // 변수 a를 시리얼 모니터에 출력  
  delay(3000);                  // 1초 대기  
  Serial.println(a);            // 변수 a를 시리얼 모니터에 출력  
}
```

## Raspberry Pi 개발환경 설정

```
sudo apt-get update  
sudo apt-get upgrade
```

### 한글 설정

```
sudo apt-get install fonts-unfonts-core  
sudo apt-get install ibus  
sudo apt-get install ibus-hangul  
ibus engine hangul
```

### nano 에디터 수정

<https://www.thewordcracker.com/miscellaneous/%EB%A6%AC%EB%88%85%EC%8A%A4%EC%9D%98-nano-%EC%97%90%EB%94%94%ED%84%B0-%EB%8B%A8%EC%B6%95%ED%82%A4/>

### Import error

<https://stackoverflow.com/questions/23985163/python3-error-no-module-named-bluetooth-on-linux-mint>

## 블루투스 개발환경 설정

패키지 설치 `sudo apt-get install bluetooth bluez bluez-tools`  
`sudo python3 -m pip install pybluez`

실행 `bluetoothctl` <https://fishpoint.tistory.com/3430>

블루투스 mac 주소 찾기:

1. 블루투스 모듈 이름을 알아볼 수 있게 설정한다.
2. 폰과 먼저 연결한다.(비밀번호는 0000)
3. HC-05 어플 찾아서 mac 주소 확인

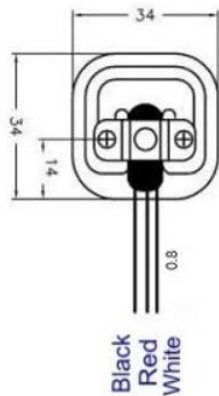
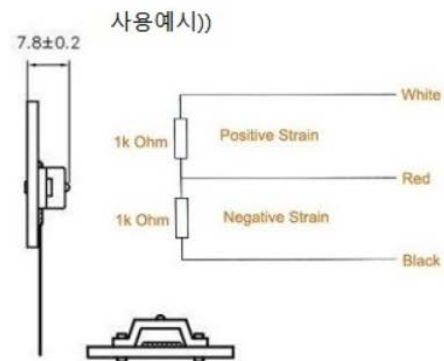
블루투스 페어링 <https://webnautes.tistory.com/979>

## 무게센서

## 상품설명



정격 하중 : 50kg  
 정격 출력 :  $1.0 \pm 0.1 \text{ mV} / \text{V}$   
 비선형 성 : 0.08 % F.S  
 히스테리시스 : 0.1 % F.S  
 반복성 : 0.05 % F.S  
 크리프 (5 분) : 0.05 % F.  
 출력시 온도 효과 :  $0.02 \% \text{ F.S} / ^\circ \text{C}$   
 영점 온도 영향 :  $0.02 \% \text{ F.S} / ^\circ \text{C}$   
 제로 밸런스 :  $\pm 0.1000 \text{ mV} / \text{V}$   
 입력 임피던스 :  $1000 \pm 20 \% \text{ 옴}$   
 출력 임피던스 :  $1000 \pm 10 \% \text{ 옴}$   
 절연 저항 : 2000 Mohm  
 안전 과부하 : 120 % F.S  
 충격 과부하 : 150 % F.S  
 작동 온도 범위 :  $-20 \sim 65 ^\circ \text{C}$   
 권장 전압 : 5 VDC  
 최대 작동 전압 : 8VDC  
 보호 수준 : IP65  
 재질 : 알루미늄 합금  
 케이블 :  $0.8 \times 460 \text{ mm}$



## 구매처

[http://makepcb.co.kr/mall/m\\_search.php?ps\\_mode=search&url=index.php&ps\\_search=%B7%CE%B5%E5%BC%BF&x=0&y=0](http://makepcb.co.kr/mall/m_search.php?ps_mode=search&url=index.php&ps_search=%B7%CE%B5%E5%BC%BF&x=0&y=0)

## Raspberry Pi -> Mobius 서버 전송

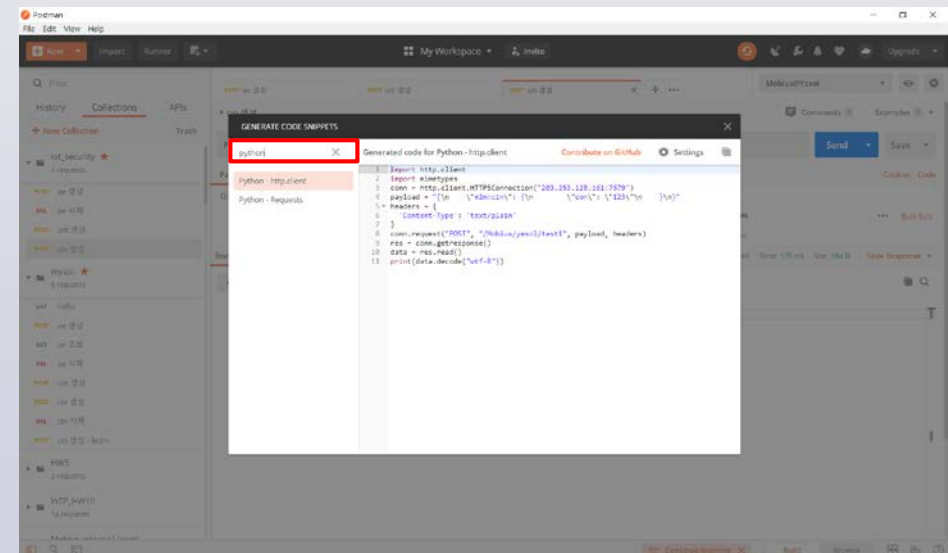
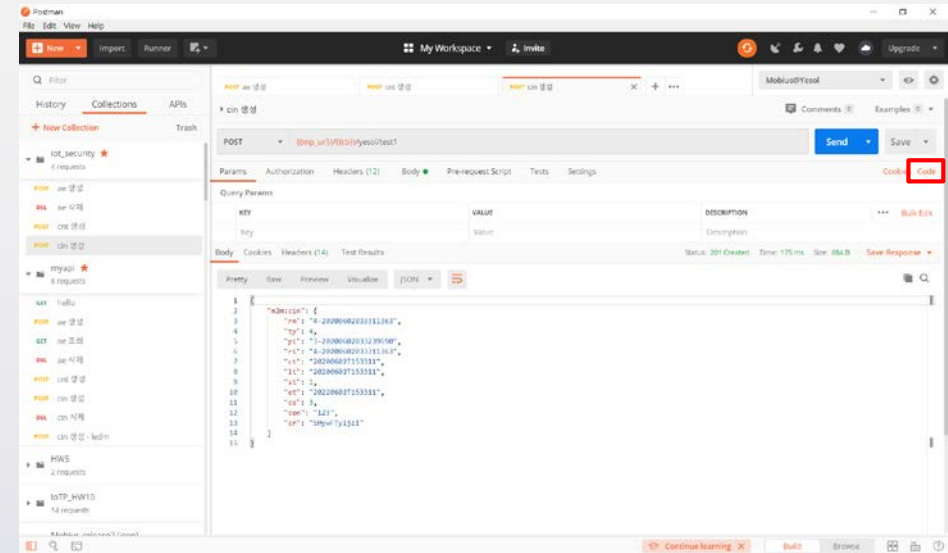
### <파일 타입 별로 전송 방법 쉽게 찾는 방법>

1. Postman을 설치한다.
2. 간단히 모비우스 구조를 만든다.  
방법을 모를 때는 IoT 플랫폼 10주차 강의안을 참고
3. Code를 누른다.
4. 원하는 파일 형식을 검색한다.

### <확인 사이트 주소>

모비우스 링크:

<http://203.253.128.161:7575/#!/monitor>





## python으로 모비우스에 cin 값 올리기

[https://github.com/loTKETI/oneM2M-API-Testing/blob/master/REST\\_API\\_for\\_Mobius\\_Rel2.pdf](https://github.com/loTKETI/oneM2M-API-Testing/blob/master/REST_API_for_Mobius_Rel2.pdf)

```
# post_cin.py

import requests

url = "http://203.253.128.161:7579/Mobius/Yesol/test1"

payload = "{\n    \"m2m:cin\": {\n        \"con\": \"456\"\n    }\n}"

headers = {'Content-Type': 'text/plain;ty=4', 'X-M2M-RI': '12345', 'X-M2M-Origin': 'Syasol'}

response = requests.post(url, headers = headers, data = payload)

print(response.text.encode('utf8'))
```

### 변경사항

1. url에서 **http://** 추가
2. header에서 **ty=4**, **'X-M2M-RI': '12345'**, **'X-M2M-Origin': 'Syasol'** 추가
3. post를 앞으로 빼는 것은 선택인 것 같음



## python으로 모비우스에 cin 값 올리기

[https://github.com/loTKETI/oneM2M-API-Testing/blob/master/REST\\_API\\_for\\_Mobius\\_Rel2.pdf](https://github.com/loTKETI/oneM2M-API-Testing/blob/master/REST_API_for_Mobius_Rel2.pdf)

```
# get_cin_1.py

import requests

url = "http://114.71.221.47:7579/Mobius/Server/led/1a"

payload = {}

headers = {'Content-Type': 'text/plain;ty=4', 'X-M2M-RI': '12345', 'X-M2M-Origin': 'SServer'}

response = requests.get(url, headers = headers, data = payload)

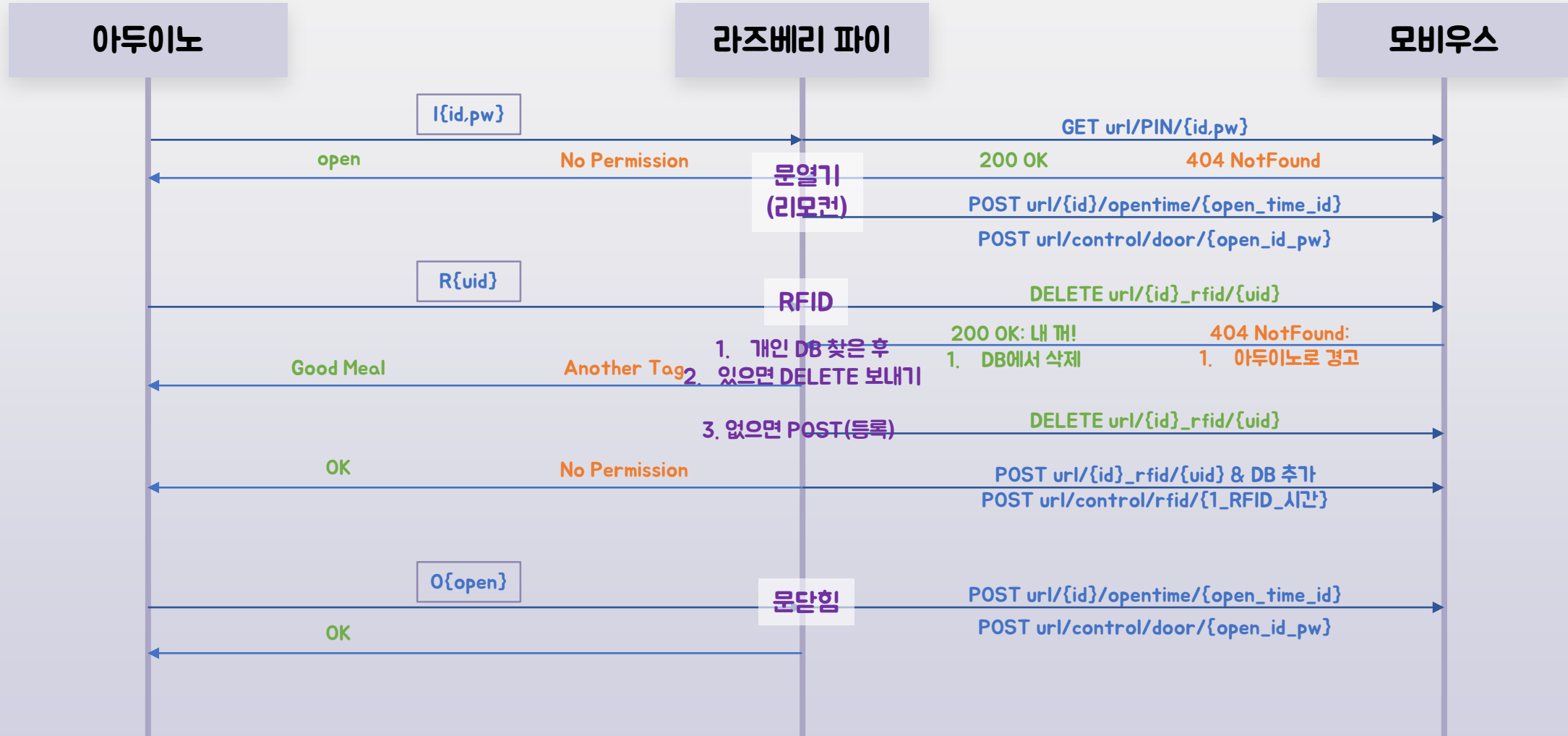
print(response.text.encode('utf8'))
```

### 변경사항

1. url에서 **http://** 추가
2. header에서 **ty=4**, **'X-M2M-RI': '12345'**, **'X-M2M-Origin': 'Syasol'** 추가
3. Get을 앞으로 빼는 것은 선택인 것 같음

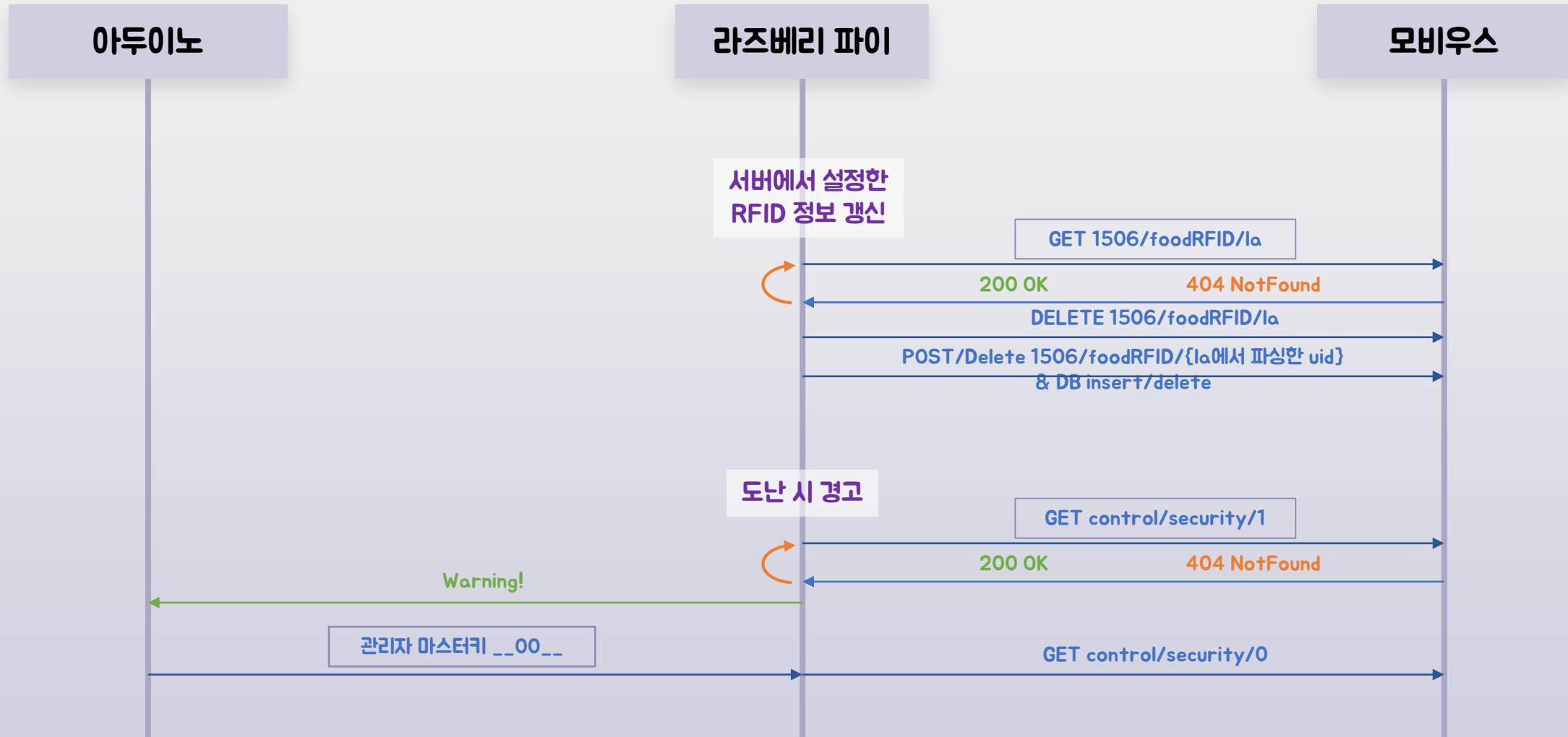
기본 url = http://114.71.221.47:7579/Mobius/

## 데이터 이동 정리(예솔)



기본 url = <http://114.71.221.47:7579/Mobius/>

## 데이터 이동 정리(예솔)

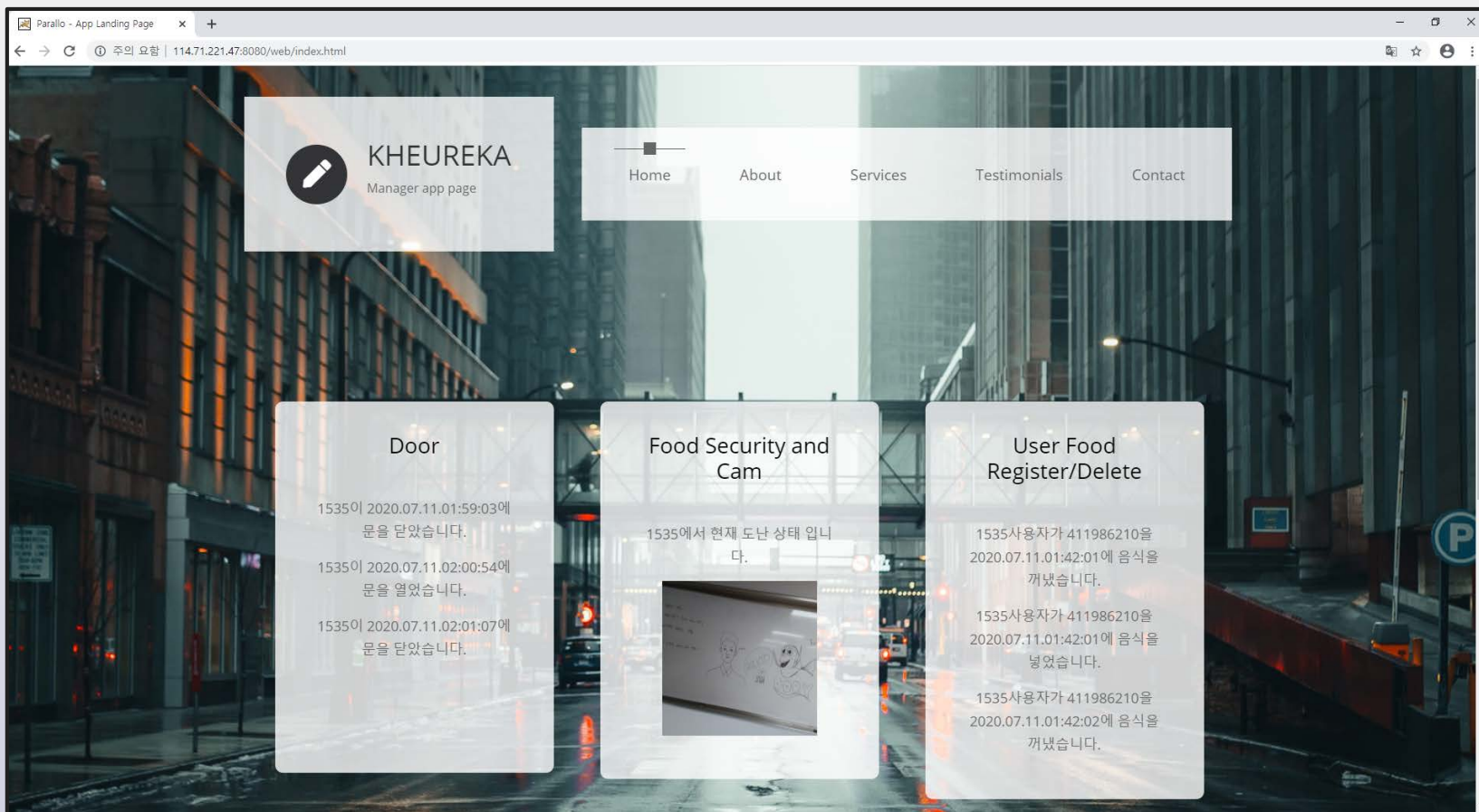


기본 url = <http://114.71.221.47:7579/Mobius/>

## 데이터 이동 정리(예솔)



## 모비우스 서버 설치 (KETI사의 제공하는 서버 사용함) & 테스트를 위해 포스트맨 같이 설치



## MySQL (=MariaDB) //동아리 확인 홈페이지 만들기.pptx 4쪽

- (진짜) 설치
  - `sudo apt-cache search mariadb`
    - Mariadb-server 있는지 찾기
  - `sudo apt-get install mariadb-server`
- 비밀번호 설정:
  - `sudo mysqladmin -u root -p password 비밀번호`
- DB 실행
  - `sudo mysql -u root -p`

- `create database securIoT;`
- `use securIoT`
- `create table RFID(id int not null auto_increment primary key, uid char(13) not null);`
- `select * from RFID;`
- `insert into RFID(uid) value('123123123123');`
- `delete from RFID where uid='123123123123';`

- 기본 설정
  - 데이터베이스 생성 `create database iot_club;`
  - 데이터베이스 목록 보기 `show databases;`
  - 데이터베이스 사용(지정) `use iot_club;`
  - 데이터베이스 삭제 `drop database iot_club;`
  - 테이블 생성 `create table mytable (//테이블 형식);`
  - 테이블 삭제 `drop table mytable;`
  - 테이블 이름 변경 `rename table table1 to table2;`
  - 테이블 업데이트 (수정) `update mytable set club = AI where id=20171535;`
  - 테이블 보기 `select id, name from mytable;`  
`select id, name from mytable where id =20171535;`

## 안드로이드 스튜디오

### 안드로이드 스튜디오 설치

위치: 윈도우

설치 <https://developer.android.com/studio?hl=ko>

참고: 책1: Do it! 안드로이드 앱 프로그래밍

책2: 안드로이드 Kotlin 앱 프로그래밍 가이드



The background is a dark blue gradient. A large, thin blue circle is centered on the slide. Inside this circle is a dashed yellow circle. Two horizontal yellow lines are positioned above and below the text. Several diagonal lines in shades of blue and yellow cross the frame. Small white dots are scattered across the background, resembling stars or data points.

**Thank you**