

지하철 자리 확인

아두이노 챌린지



B.W

2021132010

김유진

2021134042

최이설

CONTENTS



01 제작동기

02 · 작품설명

03 · 동작설명

04 · 마무리

제작동기



통학을 하는 사람들에게는 지하철의 자리가 매우 중요하다. 왜냐하면 자신이 가는 목적지까지 편안하게 앉아서 가기 위해서이다. 그러기 위해서는 지하철의 호칸의 자리에 사람이 있는지 없는지 확인할 수 있는 앱이 있으면 좋겠다는 생각을 하였다.



블루투스 모듈은 10m안에서 통신이 가능하고 와이파이에는 블루투스와 같은 제한이 없다. 따라서 와이파이를 사용하는 것이 더 이득이다.


02

작품 설명



작품 설명

디자인



지하철 의자의 형상을 확실하게 보기 위해 간단하게 제작을 하였고, 또한 감압센서와 조도 센서, 아두이노 회로 구성을 직접 확인할 수 있도록 뒤에 덮개를 제작하였습니다.

작품 설명(작품1)

지하철 자리 확인



✓ 사용한 부품: 아두이노 우노, 블루투스 모듈, 조도센서, 브레드 보드, 점퍼선, 건전지

✓ 블루투스 모듈을 사용해 정보를 전달하도록 하고, 앱인벤터를 사용해 앱을 구축한 다음 서로 간의 정보를 주고 받을 수 있도록 조정한다.

작품 설명(작품2)

지하철 자리 확인



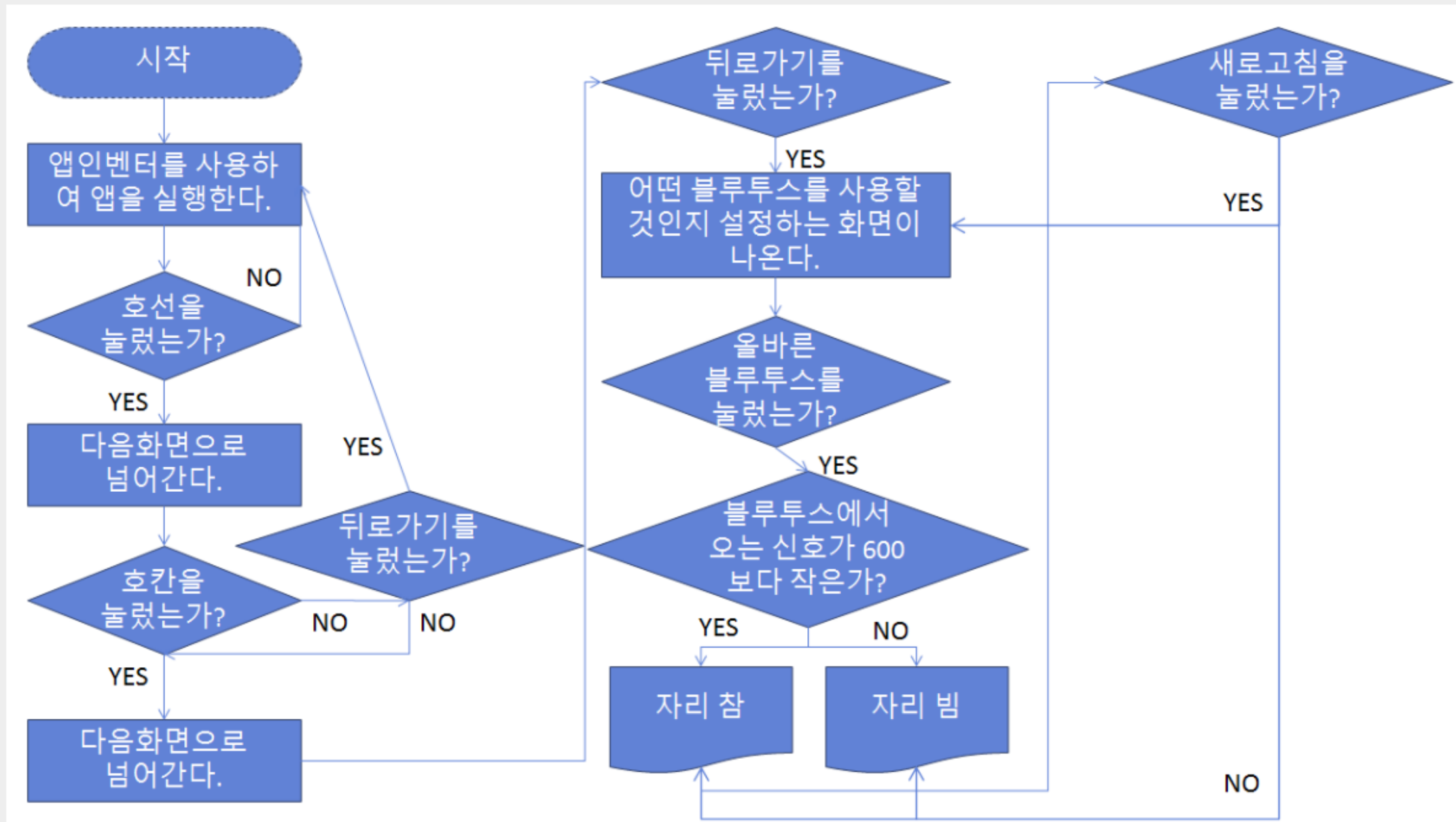
✓
사용한 부품: 아두이노 우노, 와이파이 보드 D1, 감압센서, 조도센서, 브레드 보드, 점퍼선, 건전지

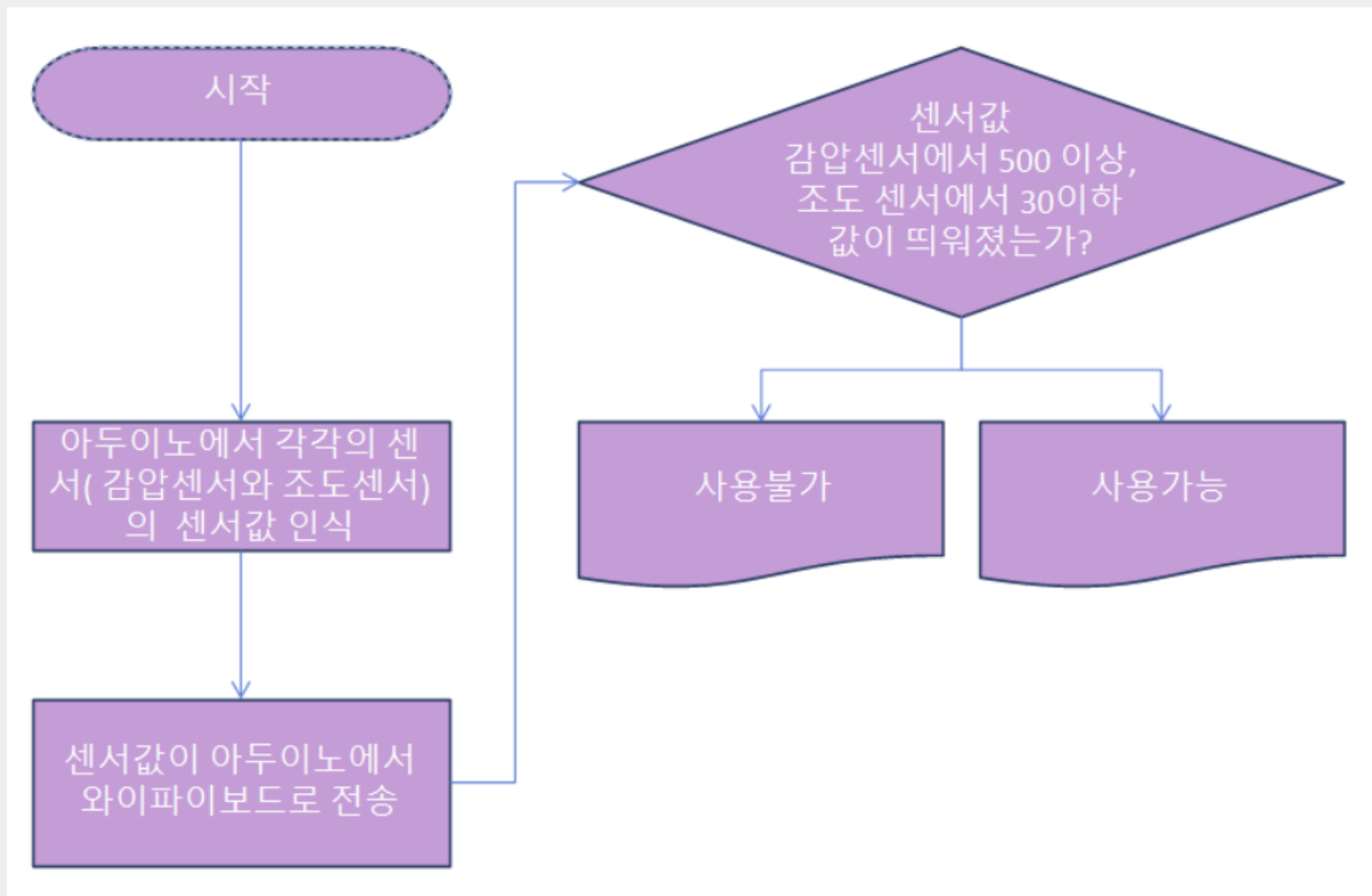
✓
와이파이 보드를 사용하여 각각의 보드에 입력되는 값을 앱으로 넘긴 뒤 각 자리에 대한 값을 확인할 수 있게 제작

03

동작 설명







Thank you

질문있으면 해주시길 바랍니다.



작품1

```
#include <SoftwareSerial.h> //외부의 라이브러리를 쓰기 위해서 아두이노에서 미리 만들어진 함수를 설정
SoftwareSerial mySerial(2, 3); //블루투스 핀 설정
int light = A0; //정수의 숫자를 저장하기 위한 변수와 그 변수가 아두이노의 어디에 해당하는지
void setup() {
  Serial.begin(9600); //시리얼 통신 9600 통신 속도로 시작
  mySerial.begin(9600);
}
void loop() {
  int m_light = analogRead(light); //지정한 아날로그 핀에서 light의 값을 읽는다.
  Serial.println(m_light); //시리얼 통신에 위의 m_light 값을 보내준다.
  mySerial.println(m_light);
  delay(1000);
}
```



○아이에 숨겨진 컴포넌트 나타내기

전화 크기 (505,320)



- subway_number
 - TableArrangement1
 - Button3
 - Button4
 - Button5
 - Button7
 - Button8
 - Button9
 - Button2
 - Button6
 - Button1

- subway_number
 - 스크린설명
 - 기본값:
 - 수평정렬
 - 기본값: 3
 - 수직정렬
 - 기본값: 1
 - 배경색
 - 기본값:
 - 배경이미지
 - 기본값:
 - BigDefaultText
 - 기본값:
 - 스크린닫기애니메이션
 - 기본값:

언제 Button1 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1

언제 Button2 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1

언제 Button3 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1

언제 Button4 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1

언제 Button5 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1

언제 Button6 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1

언제 Button7 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1

언제 Button8 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1

언제 Button9 .클릭했을때

실행 다른 스크린 열기 스크린 이름 Screen1



작품1-앱인벤터

○보여줄거진 컴포넌트 나타내기
전화 크기 (506.320) ▾



- Screen1
 - TableArrangement1
 - Button1
 - Button3
 - Button4
 - Button5
 - Button6
 - Button7
 - Button8
 - Button9
 - Button10
 - Button2
 - Button11

Screen1

스크린설정

강조색

기본값

수정방법

가운데.3 ▾

수직정렬

위.1 ▾

접이줄

subway

배경색

기본값

배경이미지

언제 Button1 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Button2 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Button3 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Button4 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Button5 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Button11 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 subway_number ▾

언제 Button6 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Button7 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Button8 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

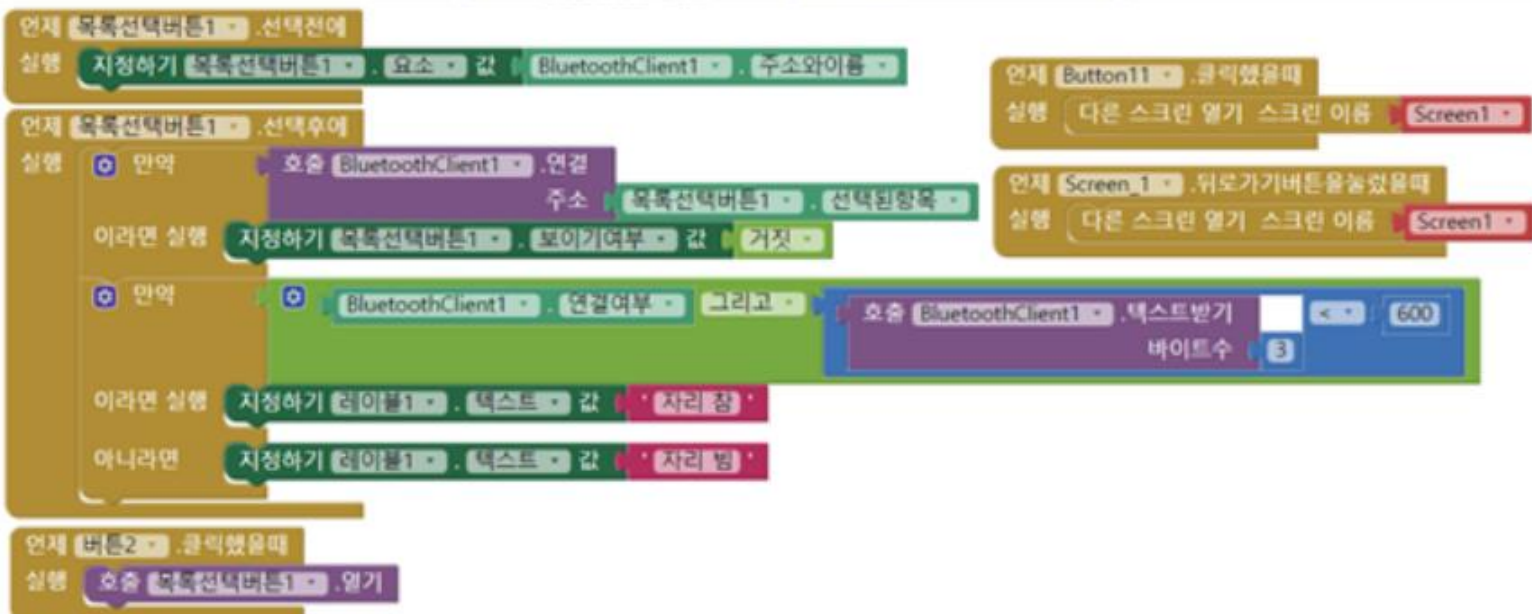
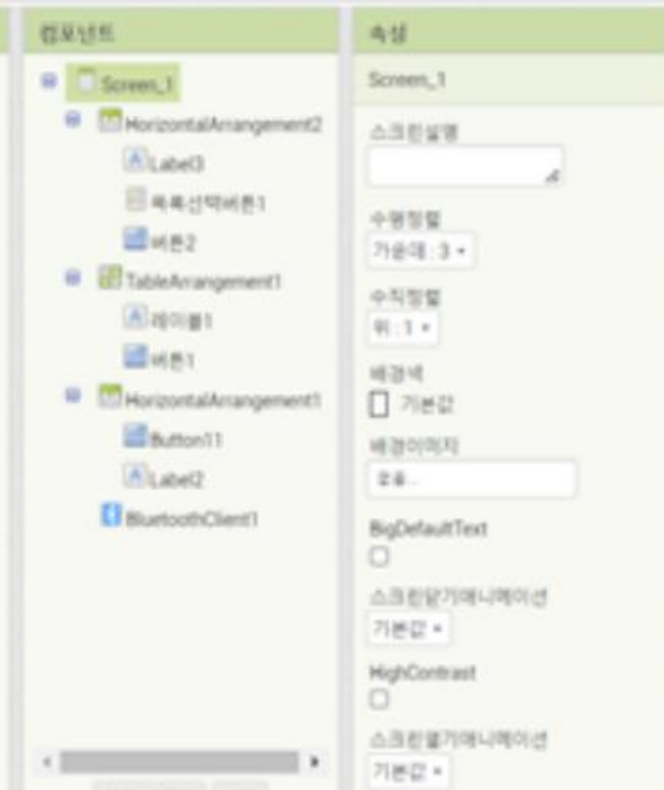
언제 Button9 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Button10 ▾ .클릭했을때
실행 다른 스크린 열기 스크린 이름 Screen_1 ▾

언제 Screen1 ▾ .뒤로가기버튼을눌렀을때
실행 다른 스크린 열기 스크린 이름 subway_number ▾



작품1-앱인벤터



작품1-앱인벤터

작품2

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial(3,4); //시리얼 통신을 위한 객체선언
int L_sensor_0 = A0;
int W_sensor_1 = A1;
int L_sensor_2 = A2;
int L_sensor_3 = A3;
int value_0, value_1, value_2, value_3;
int mid, result;
void setup() {
  Serial.begin(9600); //시리얼 모니터
  mySerial.begin(9600); //와이파이 보드 시리얼
}
void loop() {
  value_0=analogRead(L_sensor_0); //센서값들을 각각의 value 그릇에 저장
  value_1=analogRead(W_sensor_1);
  value_2=analogRead(L_sensor_2);
  value_3=analogRead(L_sensor_3);
  if(value_0 >= 30) mid = 2000; //첫번째 자리에 사람이 없을 때
  else mid = 1000; //첫번째 자리에 사람이 있을 때
  if(value_1 <= 500) mid = mid + 200; //두번째 자리에 사람이 없을 때
  else mid = mid + 100; //두번째 자리에 사람이 있을 때
  if(value_2 >= 30) mid = mid + 20; //세번째 자리에 사람이 없을 때
  else mid = mid + 10; //세번째 자리에 사람이 있을 때
  if(value_3 >= 30) mid = mid + 2; //네번째 자리에 사람이 없을 때
  else mid = mid + 1; //네번째 자리에 사람이 있을 때
```

```
switch(mid){
  case 1111: result = 70; break; //F XXXX
  case 1112: result = 71; break; //G XXXO
  case 1121: result = 72; break; //H XXOX
  case 1122: result = 73; break; //I XXOO
  case 1211: result = 74; break; //J XOXX
  case 1212: result = 75; break; //K XOXO
  case 1221: result = 76; break; //L XOOX
  case 1222: result = 77; break; //M XOOO
  case 2111: result = 80; break; //P OXXX
  case 2112: result = 81; break; //Q OXXO
  case 2121: result = 82; break; //R OXOX
  case 2122: result = 83; break; //S OXOO
  case 2211: result = 84; break; //T OOOX
  case 2212: result = 85; break; //U OOXO
  case 2221: result = 86; break; //V OOOX
  case 2222: result = 87; break; //W OOOO
  default: result = 63; //? ERRO 값 출력
}
//시리얼 모니터 내용을 블루투스 측에 write
mySerial.write((char)result);
Serial.println((char)result);
delay(1000);
result = result - result; //result 값 초기화
}
```



작품2-아두이노


```

#include<SoftwareSerial.h>
SoftwareSerial mySerial(D4, D5); //시리얼 통신을 위한 객체선언
#include <ESP8266WiFi.h>
const char* ssid = "1201-2.4G-1"; // 사용 중 인 와이파이 이름
const char* password = "kpu123456!"; // 와이파이 패스워드
WiFiServer server(80);
int result = 0;
void setup() {
  Serial.begin(9600); //시리얼 모니터
  mySerial.begin(9600); //아두이노 시리얼
  delay(10);
  Serial.println();
  // Connect to WiFi network
  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin("1201-2.4G-1", "kpu123456!");
  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  // Start the server
  server.begin();
  Serial.println("Server started");
  // Print the IP address
  Serial.println(WiFi.localIP());
}

void loop() {
  WiFiClient client = server.available();
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println("Connection: close");
  client.println("Refresh: 1"); // 자동으로 웹페이지 새로고침 (1초 설정)
  client.println();
  client.println("<!DOCTYPE html>");
  client.println("<html xmlns='http://www.w3.org/1999/xhtml'>");
  client.println("<head>\n<meta http-equiv='Content-Type' content='text/html; charset=utf-8' />");
  //client.println("<meta http-equiv='Content-Type' content='text/html; charset=utf-8' />");
  //client.println("<head>\n<meta charset='UTF-8'>");
  client.println("<title>B.W</title>"); // 웹 서버 페이지 제목 설정
  client.println("</head>\n<body>");
  client.println("<center>"); // 웹 페이지 내용 중앙 정렬
  client.println("<H1>지하철 이용 현황</H1>"); // 페이지 내용 설정
  client.println("<br>");
  if(mySerial.available() > 0){
    result = mySerial.read(); //아두이노에서 전송받은 데이터 result그릇에 저장
  }
}

```



작품2-와이파이 보드

```
//웹 페이지 내용 작성
if (result<80) client.print("<H3>1번 : 사용 불가</H3> ");
else client.print("<H2>1번 : 사용 가능</H2> ");
client.println("<br>");
if((result%10)<=3) client.print("<H3>2번 : 사용 불가</H3> ");
else client.print("<H2>2번 : 사용 가능</H2> ");
client.println("<br>");
if(((result%10)==0)||((result%10)==1)) client.print("<H3>3번 : 사용 불가</H3> ");
else client.print("<H2>3번 : 사용 가능</H2> ");
client.println("<br>");
if(result%2 != 0) client.print("<H3>4번 : 사용 불가</H3> ");
else client.println("<H2>4번 : 사용 가능</H2>");
client.println("<br>");
delay(1000);
client.println("<br>");
client.println("<br>");
client.println("<H1> 김유진, 최이설 </H1>"); // 페이지 내용 설정
client.println("<pre>");
client.print("</body>\n</html>");
}
```

