



드론으로 배우는
프로그래밍 교실

Ch4. 아두이노 기초2 - 2



01 for문에 대해서	01
for문	02
fading 예제 작성하기	03
fading 예제 최종 구성	04
 02 while문에 대해서	 05
while문	06
do while문	07
while문과 스위치 카운트	08



드론으로 배우는
프로그래밍 교실

초판발행 2016년 9월 23일
지은이 최정애 | 펴낸이 최정애
펴낸곳 WHIT | 주소 안산시 한양대학교로55 창업보육센터 B01
전화 010-5125-2139

Published by WHIT. Printed in Korea
Copyright © 2016 최정애 & WHIT

이 책의 저작권은 최정애와 WHIT에 있습니다.
저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

01 for문에 대해서



for문은 반복문 중 하나로 c언어에서 반복문을 표현할 때 주로 사용됩니다. 일정하게 숫자가 변할 때 각각의 숫자에 대해 다르게 적용시킬 명령이 있을 때 사용됩니다.

예를 들어 배열의 0번째부터 29번째까지 다르게 데이터를 넣어야 할 때 사용하면 편리하게 적용할 수 있습니다.

for문이란?

영어에서 **for**는 ~동안 이라는 뜻으로, 프로그래밍에서도 마찬가지로 정해진 기간동안 명령을 반복하는 반복문으로 사용됩니다.

변수를 선언한 후 그 변수가 정해진 범위 내에서 일정한 수만큼 증가함에 따라 명령을 다르게 할 때 주로 사용됩니다.

예를 들어 다음과 같이 변수 *i*가 0에서 3까지 1씩 증가하면서 명령어를 실행하면 0, 1, 2가 출력됩니다.



<그림1-1> for문 블록 코딩

for문은 아래 그림처럼 **초기값**, **종결값**, **변화조건**으로 구성됩니다. *i=0*처럼 반복의 시작이 어디서부터인지 정하는게 초기값입니다. 종결값은 *i<3* 인 부분입니다. 변화조건은 *i++* 인 부분으로 값을 변화시켜 반복을 제어하게 됩니다.

```
for( i=0 ; i<3 ; i++ )
{
}

```

*i=0*으로 *i*가 3보다 *i*를 1씩
초기화 작을때까지 증가

우선 *i*에 0을 대입하고 명령문을 실행합니다.
명령을 실행한 다음 변화조건을 적용합니다.
이후, 종결값과 *i*를 비교한 후 참이면 다시 명령을 실행하게 됩니다.

fading 예제 작성하기

fading
예제

- 1 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.


ch4_2_1_fading

```

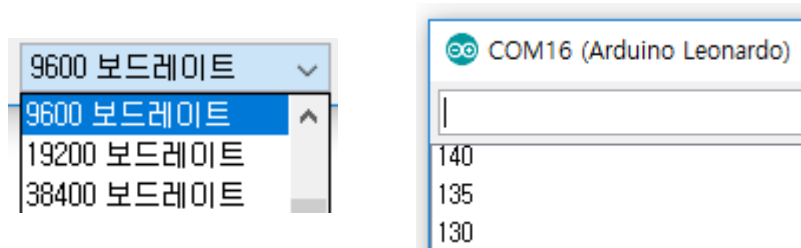
1 void setup() {
2   Serial.begin(9600);
3   pinMode(3, OUTPUT);
4 }
5
6 void loop() {
7   for (int i = 0; i <= 255 ; i += 5) {
8     Serial.println(i);
9     analogWrite(3, i);
10    delay(30);
11  }
12  for (int i = 255; i >= 0 ; i -= 5) {
13    Serial.println(i);
14    analogWrite(3, i);
15    delay(30);
16  }
17 }

```

<그림1-2> fading 예제

- 2  버튼을 눌러 시리얼 모니터를 켭니다.

- 3 보드레이트를 맞춘 후 스위치를 누르고 시리얼 모니터에서 값이 변하는지, LED의 밝기 변화가 있는지 확인합니다.



<그림1-3> 값 확인

fading 예제 해석

```
void setup() {  
  Serial.begin(9600);  
  pinMode(3, OUTPUT);  
}  
  
void loop() {  
  for (int i = 0; i <= 255; i += 5) {  
    //i가 0부터 255까지 5씩 증가  
    Serial.println(i); //i값 출력  
    analogWrite(3, i); // i값만큼 LED밝기 변화  
    delay(30);  
  }  
  for (int i = 255; i >= 0; i -= 5) {  
    //i가 255부터 0까지 5씩 감소  
    Serial.println(i); //i값 출력  
    analogWrite(3, i); //i값만큼 LED밝기 변화  
    delay(30);  
  }  
}
```

02 while문에 대해서



while문은 조건이 맞으면 계속적으로 실행되는 반복문입니다.

보통 조건에 true를 집어넣고 계속적으로 반복시키며 사용합니다.

while문의 제어는 break나 continue 등을 통해 이루어집니다.

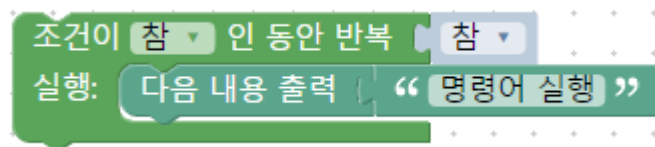
while문은 굉장히 간단한 구조로 광범위하게 사용되기에 꼭 사용법을 알아두어야 합니다.

**while문
이란?**

while문은 for문과 같은 반복문 중 하나로 간단한 구조를 가지고 있습니다.

for문과는 달리 조건에 대한 검사만 하며 조건이 참이면 명령을 반복합니다.

예를 들어 다음과 같이 괄호 안의 조건이 참인 경우만 계속 반복적으로 실행합니다.



<그림2-1> while문 블록 코딩

while문은 다음과 같은 구조를 가지고 있습니다.

```
while( i<3 )
{
    i가 3보다
    작으면 반복
}
```

while문은 조건을 빠져나갈 수 있는 **break;**와 조건 검사로 돌아가는 **continue;**를 통해 제어할 수 있습니다.

break

break를 쓰면 while문의 반복을 멈추고 중괄호를 빠져나갈 수 있습니다.

예를 들어 다음과 같은 코드에서는 **i++**이 실행되지 않습니다.

```
while( true )
{
    break;
    i++;
}
```


continue

continue를 쓰면 해당 지점에서 바로 다시 while문의 조건을 검사한 후 위에서부터 차례대로 명령을 실행합니다.

아래와 같은 코드에서는 i가 2와 같으면 continue에 의해 다시 조건 검사로 이동하기 때문에 i++이 적용되지 않아 반복문이 계속 실행됩니다.

```
int i = 0 ;
while( i < 3 )
{
    if( i == 2 )
    {
        continue;
    }
    i++;
}
```

do while문
이란?

do while문은 while문과 비슷하지만 약간 다릅니다.

while문에서는 조건을 먼저 검사한 후 참이면 명령이 실행되지만 do while문에서는 먼저 명령을 실행한 뒤 조건을 검사하여 반복할지 말지를 결정합니다.

```
do
{
    x++;
    Serial.print(x);
}
while ( x < 3 ) ;
```

while문과 스위치 카운트

while문과
스위치
카운트

- 1 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.


ch4_2_2_while_sw

```

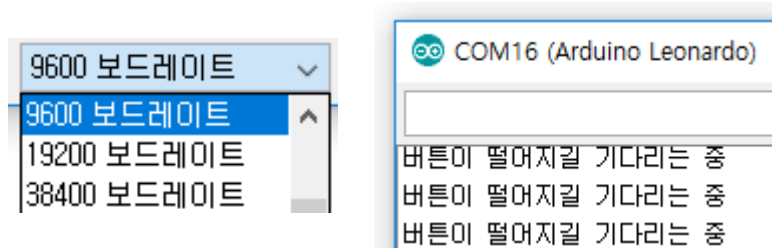
1 int cnt = 0;
2
3 void setup() {
4     Serial.begin(9600);
5     pinMode(2, INPUT);
6 }
7
8 void loop() {
9     if (digitalRead(2) == HIGH) {
10         cnt++;
11         if (cnt > 3) cnt = 0;
12
13         while (digitalRead(2) == HIGH) {
14             Serial.println("버튼이 떨어지길 기다리는 중");
15         }
16
17         Serial.print("버튼 눌림 횟수 : ");
18         Serial.println(cnt);
19     }
20 }

```

<그림2-2> while문과 스위치 카운트

- 2  버튼을 눌러 시리얼 모니터를 켭니다.

- 3 보드레이트를 맞춘 후 스위치를 누르고 시리얼 모니터에서 값이 변하는지 확인합니다.



<그림2-3> 값 확인

while문과 스위치 카운트 해석

```
int cnt = 0; //스위치 눌림 횟수 카운트 변수
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(2, INPUT);  
}
```

```
void loop() {  
  if (digitalRead(2) == HIGH) { //만약 스위치가 눌렀다면  
    cnt++; // cnt값 증가  
    if (cnt > 3) cnt = 0; //만약 cnt가 4이상이면 0으로  
    while (digitalRead(2) == HIGH) { //스위치가 눌린 상태면 반복  
      Serial.println("버튼이 떨어지길 기다리는 중"); //문구 출력  
    }  
    Serial.print("버튼 눌림 횟수 : ");  
    Serial.println(cnt); //스위치 눌림 횟수 출력  
  }  
}
```

아두이노는 빠른 속도로 동작하기 때문에, 스위치를 한 번만 누르려고 해도 순식간에 여러 번 눌리게 됩니다. **while**문을 사용하면, 스위치가 눌러 있는 동안 다음 명령어로 넘어가지 않기 때문에 스위치를 한 번만 누를 수 있게 됩니다.

03 배열



배열이란 변수들을 손쉽게 쓸 수 있는 도구입니다.

대량의 데이터를 다룰 때 배열이 없다면 변수를 일일이 선언해야 할 것입니다. 변수를 일일이 선언한다는 것은 프로그래밍의 취지인 단순 반복 작업의 해결과는 거리가 먼 행동입니다.

배열을 잘 활용하는 것은 초급 프로그래머를 탈출하기 위한 첫 관문입니다.

배열이란?

배열
정의

배열이란 간단히 말해 변수들을 한 묶음으로 묶어 놓은 것입니다. 만약 30명이 있는 1반 학생들의 영어 점수를 변수에 저장한다고 하면 다음과 같이 변수가 30개 필요하게 됩니다.

```
int a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, ...
```

일일이 변수를 선언하는 작업은 매우 귀찮고 반복적인 작업을 필요로 합니다. 이 때 배열을 사용하면 간단히 문제를 해결할 수 있습니다.

```
int    a[30]    ;
```

int의 자료형을 사용하며 a라는 이름의 배열을 지정

변수의 선언부터 간단 해졌습니다. 배열을 만든 후 배열의 각 변수에 접근을 할 때는 다음과 같은 방법을 사용하면 됩니다.

```
a[0]    =    100    ;
```

a의 첫 번째 변수에 100이라는 값을 저장

배열에 들어있는 변수들은 0번부터 시작하여 29번까지 존재하게 됩니다. 배열에 들어있는 값은 다음과 같이 사용하면 됩니다.

```
int    b    =    a[0]    ;
```

int의 자료형을 사용하여 a[0]에 있는 값을 저장
b라는 이름을 지정하여

int형 배열

정수형 배열은 다음과 같이 선언하고 사용할 수 있습니다.

- `int prime[5] = {2, 3, 5, 7, 11};`
- `int three = prime[1];`

`three`라는 변수에는 `prime`이라는 배열의 2번째 값인 숫자3이 들어가게 됩니다.

double형 배열

실수형 배열은 다음과 같이 선언하고 사용할 수 있습니다.

- `double root[4] = {1.414, 1.732, 2.000, 2.236};`
- `double r5 = root[3];`

`r5`라는 변수에는 `root`라는 배열의 4번째 값인 2.236이 들어가게 됩니다.

char형 배열

문자형 배열은 다음과 같이 선언하고 사용할 수 있습니다.

- `char ch[5] = {'h', 'e', 'l', 'l', 'o'};`
- `char h = ch[0];`

`h`라는 변수에는 `ch`라는 배열의 1번째 값인 `h`가 들어가게 됩니다.

위 방식 말고도 아스키 코드값을 바로 넣을 수도 있습니다.

- `char asc[4] = {62, 95, 60, 98};`
- `char b = asc[3];`

위 코드를 넣고 실행시키면 숫자가 아니라 아스키코드표를 통한 문자가 나오게 됩니다.

또한, “”(쌍따옴표)를 통해서도 문자형 배열을 표현할 수 있습니다.

- `char str[6] = "world";`
- `char d = str[4];`

단, “”를 사용하여 문자형 배열을 만들 경우엔 배열의 길이가 문자의 길이보다 1만큼 더 길게 됩니다.

2차원 배열

2차원 배열을 사용하면 행렬을 만들 수 있습니다.

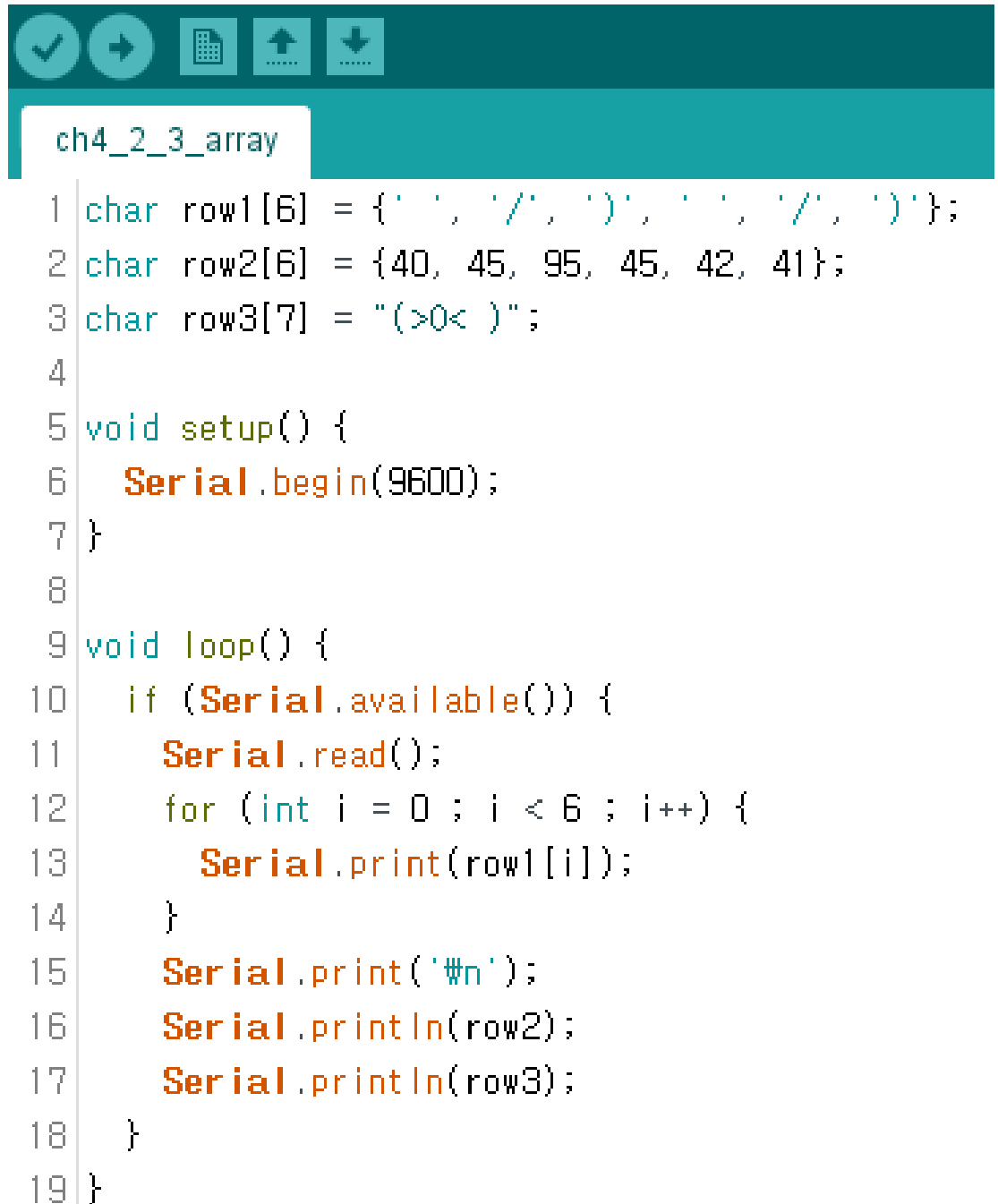
- `int rc[2][3] = {{1, 2, 3}, {4, 5, 6}};`

1	2	3
4	5	6

배열 작성 해보기

배열 작성
해보기

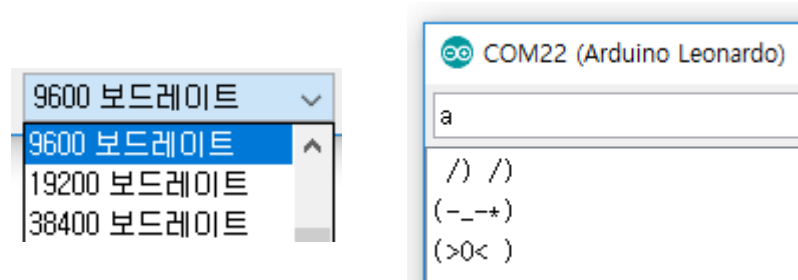
- ① 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.



<그림3-1> 배열 작성 코드

- ②  버튼을 눌러 시리얼 모니터를 켭니다.

- 3 보드레이트를 맞춘 후 시리얼 모니터에서 아무 값을 입력한 후 아두이노로부터 받은 값을 확인합니다.



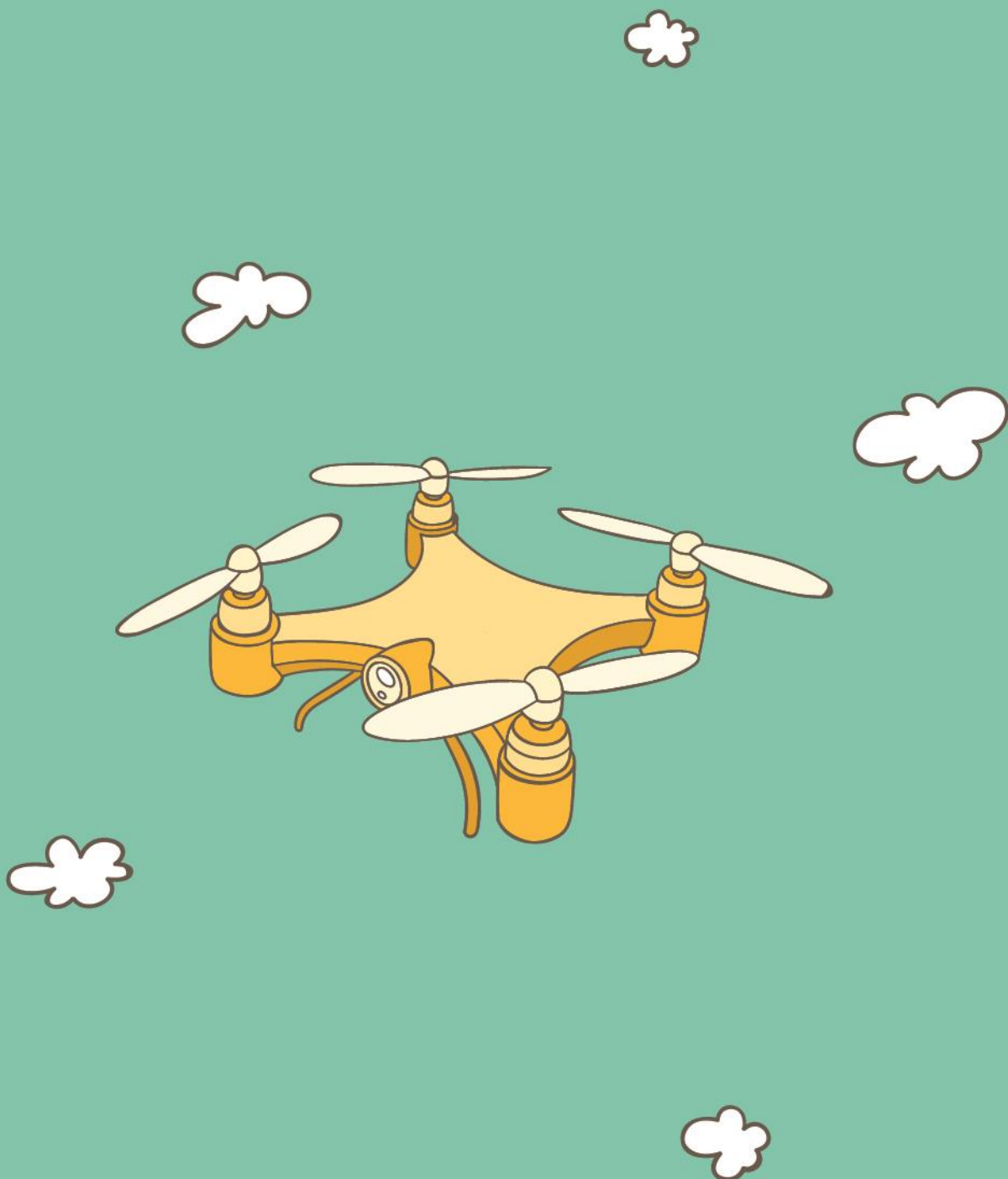
<그림3-2> 값 확인

배열 작성 해석

```
char row1[6] = {' ', '/', ' ', ' ', ' ', ' '}; // 첫 번째 배열  
char row2[6] = {40, 45, 95, 45, 42, 41}; // 두 번째 배열  
char row3[7] = ">O<"; // 세 번째 배열로 문자열 작성
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  if (Serial.available()) {  
    Serial.read();  
    for (int i = 0; i < 6; i++) { // 배열의 길이만큼 반복  
      Serial.print(row1[i]); // 첫 번째 배열의 값들 출력  
    }  
    Serial.print("\n"); // 줄 바꿈  
    Serial.println(row2); // 두 번째 배열 출력  
    Serial.println(row3); // 세 번째 배열 출력  
  }  
}
```

WHIT