



드론으로 배우는
프로그래밍 교실

Ad2. 슬롯 머신 만들기



01 슬롯 머신 만들기1	01
슬롯머신이란?	02
스위치로 랜덤 숫자 출력하기	03
슬롯머신 코드 함수화	05
02 슬롯 머신 만들기2	08
777과 랜덤값 보이기	08
슬롯 머신 커스터마이징	13
03 슬롯 머신 만들기3	14
슬롯 머신 최종 코드 변형	15
슬롯 머신 최종 코드	17
슬롯 머신 만들기 업그레이드	19



드론으로 배우는
프로그래밍 교실

초판발행 2016년 9월 23일
지은이 이상준 | 펴낸이 CodingBird
펴낸곳 WHIT | 주소 안산시 한양대학교로55 창업보육센터 B01

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

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

01 슬롯 머신 만들기1



슬롯 머신은 사용자가 버튼을 눌러 특정 숫자를 맞추면 보상을 주는 오락기입니다.

이러한 슬롯머신은 특정 숫자가 나올 확률을 조절할 수 있는 소프트웨어를 탑재하고 있습니다.

인형뽑기 등과 같은 오락기에서도 이러한 확률 조절 SW가 쓰입니다.

슬롯머신이란?

슬롯머신

슬롯머신은 동전을 투입구에 넣고 레버를 당기면 화면의 무늬 또는 숫자가 돌아가는데, 만약 같은 무늬가 일렬로 나오면 상금을 얻을 수 있는 게임입니다.



<그림1-1> 슬롯 머신

위의 그림에서 **SPIN**이란 버튼을 누르면 첫 번째 그림이 랜덤으로 정해집니다. 이후, **SPIN**버튼을 두 번 더 누르면 나머지 2개의 그림도 정해지고, 세 개의 그림이 일치하면 상금을 얻게 됩니다.



<그림1-2> 777

우리는 다음과 같이 게임을 바꾸어 진행합니다.

- **SPIN**버튼 : 스위치
- 세 그림 : 숫자(1~9)
- 세 숫자 중 1개라도 7이 나오면 상금
- 세 숫자 모두 7이 나오면 상금

스위치로 랜덤 숫자 출력하기

스위치로
랜덤 숫자
출력하기


- 1 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.

```

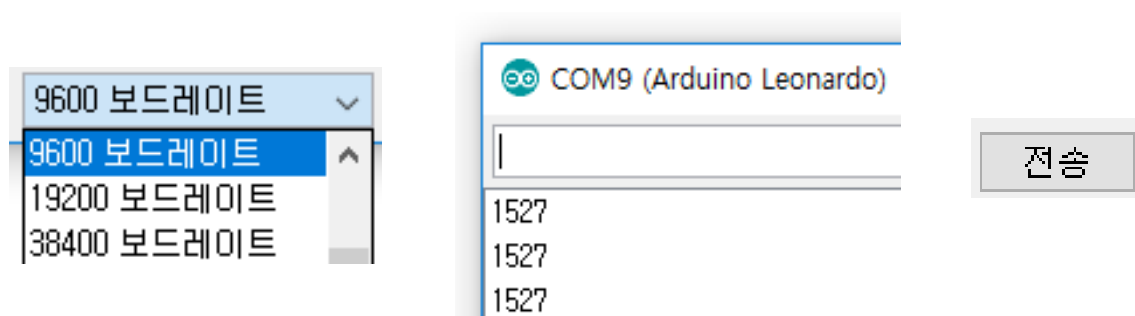
slot_machine_1
1 int cnt = 0;
2 int num[4] = {0,};
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(3, OUTPUT);
7     pinMode(2, INPUT);
8     randomSeed(analogRead(5));
9 }
10
11 void loop() {
12     if (digitalRead(2) == HIGH) {
13         while (digitalRead(2) == HIGH);
14         cnt++;
15         if (cnt > 3) cnt = 0;
16         num[cnt] = random(1, 10);
17         delay(50);
18     }
19     Serial.print(num[0]);
20     Serial.print(num[1]);
21     Serial.print(num[2]);
22     Serial.println(num[3]);
23 }

```

<그림1-3> 스위치로 랜덤 숫자 출력하기

- 2  버튼을 눌러 시리얼 모니터를 켭니다.

- 3 보드레이트를 맞춘 후 스위치를 눌러 숫자가 변하는지 확인합니다.



<그림1-4> 시리얼 통신 확인

슬롯 머신 코드 해석

```
int cnt = 0; // 눌린 횟수 저장 변수
int num[4] = {0,}; // 랜덤 숫자 저장 배열

void setup() {
  Serial.begin(9600); // 시리얼 통신 시작
  pinMode(3, OUTPUT); // 3번 핀 출력으로 설정
  pinMode(2, INPUT); // 2번 핀 입력으로 설정
  randomSeed(analogRead(5)); // 랜덤 시드 설정
}

void loop() {
  if (digitalRead(2) == HIGH) { // 만약 스위치가 눌렸다면
    while (digitalRead(2) == HIGH); // 떨어 때까지 대기
    cnt++; // 카운트 증가
    if (cnt > 3) cnt = 0; // 카운트가 4 이상이면 0으로
    num[cnt] = random(1, 10); // 1부터 9 사이의 랜덤 값 저장
    delay(50); // 스위치 채터링 방지
  }
  Serial.print(num[0]); // 랜덤 숫자 출력
  Serial.print(num[1]); // 랜덤 숫자 출력
  Serial.print(num[2]); // 랜덤 숫자 출력
  Serial.println(num[3]); // 랜덤 숫자 출력
}
```

꿀TIP

채터링이란?

스위치를 누를 때
매우 짧은 시간동안
스위치가 떨어졌다
붙었다 하여 여러 번
눌리는 현상

슬롯머신 코드 함수화

슬롯 머신
코드 함수화

- 1 다음과 같이 코드를 변형하여 아두이노에 업로드합니다.

slot_machine_2

```

1 int cnt = 0;
2 int num[4] = {0,};
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(3, OUTPUT);
7     pinMode(2, INPUT);
8     randomSeed(analogRead(5));
9 }
10
11 void loop() {
12     saveNumber();
13     printNumber();
14 }
15
16 void saveNumber() {
17     if (digitalRead(2) == HIGH) {
18         while (digitalRead(2) == HIGH);
19         cnt++;
20         if (cnt > 3) cnt = 0;
21         num[cnt] = random(1, 10);
22         delay(50);
23     }
24 }
25
26 void printNumber() {
27     Serial.print(num[0]);
28     Serial.print(num[1]);
29     Serial.print(num[2]);
30     Serial.println(num[3]);
31 }

```

<그림1-5> 슬롯 머신 코드 함수화

바뀐 부분

함수화 과정

함수화는 `loop()`문 안의 코드를 함수로 따로 빼내어, `loop()`문을 간결하게 만들어줌으로서 가독성을 높이는 작업입니다.

다음과 같은 과정을 통해 함수화합니다.

숫자 저장

숫자 출력


```
11 void loop() {  
12   if (digitalRead(2) == HIGH) {  
13     while (digitalRead(2) == HIGH);  
14     cnt++;  
15     if (cnt > 3) cnt = 0;  
16     num[cnt] = random(1, 10);  
17     delay(50);  
18   }  
19   Serial.print(num[0]);  
20   Serial.print(num[1]);  
21   Serial.print(num[2]);  
22   Serial.println(num[3]);  
23 }  
  
11 void loop() {  
12   saveNumber();  
13   printNumber();  
14 }  
15  
16 void saveNumber() {  
17   if (digitalRead(2) == HIGH) {  
18     while (digitalRead(2) == HIGH);  
19     cnt++;  
20     if (cnt > 3) cnt = 0;  
21     num[cnt] = random(1, 10);  
22     delay(50);  
23   }  
24 }  
25  
26 void printNumber() {  
27   Serial.print(num[0]);  
28   Serial.print(num[1]);  
29   Serial.print(num[2]);  
30   Serial.println(num[3]);  
31 }
```

우선, `loop()`문 안의 구조를 살펴보면 숫자를 `num[]` 배열에 저장하는 부분과, 저장된 숫자를 출력하는 부분으로 나뉩니다.

`num[]` 배열에 저장하는 부분을 `saveNumber()` 함수로, 저장된 숫자를 출력하는 부분을 `printNumber()` 함수로 빼냅니다.

이후, `loop()`문 안에 `saveNumber()` 함수와 `printNumber()` 함수를 써줍니다.

사용자 입력
받기 함수화
해석

- 2  버튼을 눌러 시리얼 모니터를 켭니다.
- 3 보드레이트를 맞춘 후 스위치를 눌러 숫자가 변하는지 확인합니다.

```
int cnt = 0; // 눌린 횟수 저장 변수
int num[4] = {0,}; // 랜덤 숫자 저장 배열

void setup() {
  Serial.begin(9600); // 시리얼 통신 시작
  pinMode(3, OUTPUT); // 3번 핀 출력으로 설정
  pinMode(2, INPUT); // 2번 핀 입력으로 설정
  randomSeed(analogRead(5)); // 랜덤 시드 설정
}

void loop() {
  saveNumber(); // saveNumber()함수 실행
  printNumber(); // printNumber()함수 실행
}

void saveNumber() { // saveNumber()함수 정의
  if (digitalRead(2) == HIGH) { // 만약 스위치가 눌렸다면
    while (digitalRead(2) == HIGH); // 땔 때 까지 대기
    cnt++; // 카운트 증가
    if (cnt > 3) cnt = 0; // 카운트가 4 이상이면 0으로
    num[cnt] = random(1, 10); // 1부터 9 사이의 랜덤 값 저장
    delay(50); // 스위치 채터링 방지
  }
}

void printNumber() { // printNumber()함수 정의
  Serial.print(num[0]); // 랜덤 숫자 출력
  Serial.print(num[1]); // 랜덤 숫자 출력
  Serial.print(num[2]); // 랜덤 숫자 출력
  Serial.println(num[3]); // 랜덤 숫자 출력
}
```

02 슬롯 머신 만들기2



슬롯 머신은 사용자가 버튼을 눌러 특정 숫자를 맞추면 보상을 주는 오락기입니다.

이러한 슬롯머신은 특정 숫자가 나올 확률을 조절할 수 있는 소프트웨어를 탑재하고 있습니다.

인형뽑기 등과 같은 오락기에서도 이러한 확률 조절 SW가 쓰입니다.

777과 랜덤값 보이기

1 다음과 같이 코드를 변형합니다.

777
검사하는
코드

slot_machine_3

```
1 int cnt = 0;
2 int num[4] = {0,};
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(3, OUTPUT);
7     pinMode(2, INPUT);
8     randomSeed(analogRead(5));
9 }
10
11 void loop() {
12     saveNumber();
13     printNumber();
14     if (num[1] == 7 && num[2] == 7 && num[3] == 7 && cnt == 3) {
15         Serial.println("!!!! JACKPOT !!!!");
16         digitalWrite(3, HIGH);
17     }
18 }
```


바뀐 부분

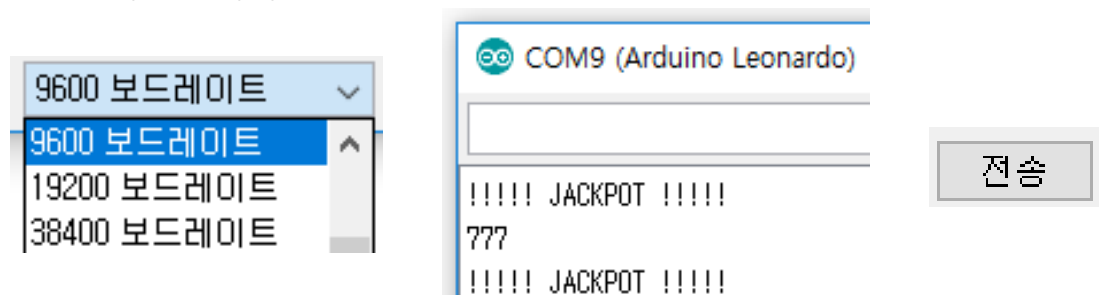
<그림2-1> 777 조건 검사를 하는 코드

위와 같이 코드를 구성하면, num[] 배열에 들어있는 값을 조사하여 777과 같은지 확인하고, 문구를 출력할 수 있습니다.

바뀐 부분

```
19
20 void saveNumber() {
21     if (digitalRead(2) == HIGH) {
22         while (digitalRead(2) == HIGH);
23         cnt++;
24         if (cnt > 3) cnt = 0;
25         num[cnt] = random(1, 10);
26         delay(50);
27     }
28 }
29
30 void printNumber() {
31     if (cnt == 0) {
32         Serial.print(random(10));
33         Serial.print(random(10));
34         Serial.print(random(10));
35     } else if (cnt == 1) {
36         Serial.print(num[1]);
37         Serial.print(random(10));
38         Serial.print(random(10));
39     } else if (cnt == 2) {
40         Serial.print(num[1]);
41         Serial.print(num[2]);
42         Serial.print(random(10));
43     } else if (cnt == 3) {
44         Serial.print(num[1]);
45         Serial.print(num[2]);
46         Serial.print(num[3]);
47     }
48     Serial.println();
49 }
```

- 2  버튼을 눌러 시리얼 모니터를 켜고 스위치를 눌러 변화를 살펴 봅니다.
- 3 보드레이트를 맞춘 후 스위치를 눌러 숫자가 변하는지 확인합니다.



<그림2-2> 시리얼 통신 확인

```
int cnt = 0; // 눌린 횟수 저장 변수
int num[4] = {0,}; // 랜덤 숫자 저장 배열

void setup() {
  Serial.begin(9600); // 시리얼 통신 시작
  pinMode(3, OUTPUT); // 3번 핀 출력으로 설정
  pinMode(2, INPUT); // 2번 핀 입력으로 설정
  randomSeed(analogRead(5)); // 랜덤 시드 설정
}

void loop() {
  saveNumber(); // saveNumber()함수 실행
  printNumber(); // printNumber()함수 실행
  if (num[1] == 7 && num[2] == 7 && num[3] == 7 && cnt == 3) {
    // 777이면
    Serial.println("!!!! JACKPOT !!!!"); // 문구 출력
    digitalWrite(3, HIGH); // LED 켜기
  }
}
```

```

void saveNumber() { // saveNumber() 함수 정의
  if (digitalRead(2) == HIGH) { // 만약 스위치가 눌렀다면
    while (digitalRead(2) == HIGH); // 땔 때 까지 대기
    cnt++; // 카운트 증가
    if (cnt > 3) cnt = 0; // 카운트가 4 이상이면 0으로
    num[cnt] = random(1, 10); // 1부터 9 사이의 랜덤 값 저장
    delay(50); // 스위치 채터링 방지
  }
}

void printNumber() { // printNumber() 함수 정의
  if (cnt == 0) { // cnt가 0일 경우
    Serial.print(random(10)); // 랜덤 출력
    Serial.print(random(10)); // 랜덤 출력
    Serial.print(random(10)); // 랜덤 출력
  } else if (cnt == 1) { // cnt가 1일 경우
    Serial.print(num[1]); // 데이터 출력
    Serial.print(random(10)); // 랜덤 출력
    Serial.print(random(10)); // 랜덤 출력
  } else if (cnt == 2) { // cnt가 2일 경우
    Serial.print(num[1]); // 데이터 출력
    Serial.print(num[2]); // 데이터 출력
    Serial.print(random(10)); // 랜덤 출력
  } else if (cnt == 3) { // cnt가 3일 경우
    Serial.print(num[1]); // 데이터 출력
    Serial.print(num[2]); // 데이터 출력
    Serial.print(num[3]); // 데이터 출력
  }
  Serial.println();
}

```

위와 같이 **printNumber()** 함수를 바꾸면, 값이 정해져 있는 자릿수의 숫자는 정해져 있는 값이 출력되고, 값이 아직 안정해진 자릿수의 숫자는 랜덤값이 출력되게 됩니다.

슬롯 머신 커스터마이징

슬롯 머신
변형하기

- 1 다음 loop문 안의 코드를 자신이 원하는 대로 바꾸어, 숫자에 따른 보상을 정하세요.

```

11 void loop() {
12   saveNumber();
13   printNumber();
14   if (num[1] == 7 && num[2] == 7 && num[3] == 7 && cnt == 3) {
15     Serial.println("!!!! JACKPOT !!!!");
16     digitalWrite(3, HIGH);
17   }
18 }

```

변형할 부분

<그림2-3> 슬롯 머신 변형하기

- 2 다음 saveNumber()의 코드를 자신이 원하는 대로 바꾸어, 숫자가 나올 확률을 조정하세요.

```

20 void saveNumber() {
21   if (digitalRead(2) == HIGH) {
22     while (digitalRead(2) == HIGH);
23     cnt++;
24     if (cnt > 3) cnt = 0;
25     num[cnt] = random(1, 10);
26     delay(50);
27   }
28 }

```

변형할 부분

<그림2-4> 슬롯 머신 변형하기

03 슬롯 머신 만들기3



슬롯 머신은 사용자가 버튼을 눌러 특정 숫자를 맞추면 보상을 주는 오락기입니다.

이러한 슬롯머신은 특정 숫자가 나올 확률을 조절할 수 있는 소프트웨어를 탑재하고 있습니다.

인형뽑기 등과 같은 오락기에서도 이러한 확률 조절 SW가 쓰입니다.

슬롯 머신 최종 코드 변형

슬롯 머신
최종

바뀐 부분

① loop문 안에 들어갈 코드입니다.

```
11 void loop() {
12   saveNumber();
13   printNumber();
14   if (num[1] == 7 && num[2] == 7 && num[3] == 7 && cnt == 3) {
15     Serial.println("!!!! JACKPOT !!!!");
16     digitalWrite(3, HIGH);
17   } else if ((num[1] == 7 || num[2] == 7 || num[3] == 7) && cnt == 3) {
18     Serial.println("!!!! rucky !!!!");
19     digitalWrite(3, HIGH);
20   } else if (num[1] == 1 && num[2] == 1 && num[3] == 2 && cnt == 3) {
21     Serial.println("!!!! police !!!!");
22     digitalWrite(3, HIGH); delay(100);
23     digitalWrite(3, LOW); delay(100);
24   }
25 }
```

<그림3-1> 슬롯 머신 최종 코드

② saveNumber() 부분 변경 코드입니다.

바뀐 부분

```
27 int rotto_count = 0;
28 void saveNumber() {
29   if (digitalRead(2) == HIGH) {
30     while (digitalRead(2) == HIGH);
31     cnt++;
32     if (cnt > 3) cnt = 0;
33     num[cnt] = random(1, 10);
34
35     if (cnt == 0) rotto_count++;
36     if (rotto_count > 3 && cnt != 0) {
37       num[cnt] = 7;
38       if (cnt == 3) rotto_count = 0;
39     }
40
41     delay(50);
42   }
43 }
```

바뀐 부분

<그림3-2> 슬롯 머신 최종 코드

슬롯 머신 최종 해석

```
if (num[1] == 7 && num[2] == 7 && num[3] == 7 && cnt == 3) {  
    Serial.println("!!!! JACKPOT !!!!"); // 777이면 문구 출력  
    digitalWrite(3, HIGH); // LED 켜  
} else if ((num[1] == 7 || num[2] == 7 || num[3] == 7) && cnt == 3) {  
    Serial.println("!!!! rucky !!!!"); // 한 개라도 7이면 문구 출력  
    digitalWrite(3, HIGH); // LED 켜  
} else if (num[1] == 1 && num[2] == 1 && num[3] == 2 && cnt == 3) {  
    Serial.println("!!!! police !!!!"); // 112면 문구 출력  
    digitalWrite(3, HIGH); delay(100); // LED 켜  
    digitalWrite(3, LOW); delay(100); // LED 끄  
}
```

```
int rotto_count = 0; // 로또 카운트 저장 변수  
void saveNumber() { // saveNumber() 함수 정의  
    if (digitalRead(2) == HIGH) { // 만약 스위치가 눌렀다면  
        while (digitalRead(2) == HIGH); // 떴을 때 까지 대기  
        cnt++; // 카운트 증가  
        if (cnt > 3) cnt = 0; // 카운트가 4 이상이면 0으로  
        num[cnt] = random(1, 10); // 1부터 9 사이의 랜덤 값 저장  
  
        if (cnt == 0) rotto_count++; // 로또 카운트 증가  
        if (rotto_count > 3 && cnt != 0) { // 로또 카운트가 3을 넘으면  
            num[cnt] = 7; // 7 저장  
            if (cnt == 3) rotto_count = 0; // 로또 카운트 초기화  
        }  
  
        delay(50);  
    }  
}
```

슬롯 머신 최종 코드

```

slot_machine_4
1 int cnt = 0;
2 int num[4] = {0,};
3
4 void setup() {
5   Serial.begin(9600);
6   pinMode(3, OUTPUT);
7   pinMode(2, INPUT);
8   randomSeed(analogRead(5));
9 }
10
11 void loop() {
12   saveNumber();
13   printNumber();
14   if (num[1] == 7 && num[2] == 7 && num[3] == 7 && cnt == 3) {
15     Serial.println("!!!! JACKPOT !!!!");
16     digitalWrite(3, HIGH);
17   } else if ((num[1] == 7 || num[2] == 7 || num[3] == 7) && cnt == 3) {
18     Serial.println("!!!! rucky !!!!");
19     digitalWrite(3, HIGH);
20   } else if (num[1] == 1 && num[2] == 1 && num[3] == 2 && cnt == 3) {
21     Serial.println("!!!! police !!!!");
22     digitalWrite(3, HIGH); delay(100);
23     digitalWrite(3, LOW); delay(100);
24   }
25 }

```

<그림3-3> 슬롯 머신 최종 코드

```

26
27 int roto_count = 0;
28 void saveNumber() {
29     if (digitalRead(2) == HIGH) {
30         while (digitalRead(2) == HIGH);
31         cnt++;
32         if (cnt > 3) cnt = 0;
33         num[cnt] = random(1, 10);
34
35         if (cnt == 0) roto_count++;
36         if (roto_count > 3 && cnt != 0) {
37             num[cnt] = 7;
38             if (cnt == 3) roto_count = 0;
39         }
40
41         delay(50);
42     }
43 }
44
45 void printNumber() {
46     if (cnt == 0) {
47         Serial.print(random(10));
48         Serial.print(random(10));
49         Serial.print(random(10));
50     } else if (cnt == 1) {
51         Serial.print(num[1]);
52         Serial.print(random(10));
53         Serial.print(random(10));
54     } else if (cnt == 2) {
55         Serial.print(num[1]);
56         Serial.print(num[2]);
57         Serial.print(random(10));
58     } else if (cnt == 3) {
59         Serial.print(num[1]);
60         Serial.print(num[2]);
61         Serial.print(num[3]);
62     }
63     Serial.println();
64 }


```

<그림3-4> 슬롯 머신 최종 코드

슬롯 머신 만들기 업그레이드

슬롯 머신
업그레이드

for문과 배열을 사용하면 다음과 같이 코드를 줄일 수 있습니다.



```

45 void printNumber() {
46   if (cnt == 0) {
47     Serial.print(random(10));
48     Serial.print(random(10));
49     Serial.print(random(10));
50   } else if (cnt == 1) {
51     Serial.print(num[1]);
52     Serial.print(random(10));
53     Serial.print(random(10));
54   } else if (cnt == 2) {
55     Serial.print(num[1]);
56     Serial.print(num[2]);
57     Serial.print(random(10));
58   } else if (cnt == 3) {
59     Serial.print(num[1]);
60     Serial.print(num[2]);
61     Serial.print(num[3]);
62   }
63   Serial.println();
64 }

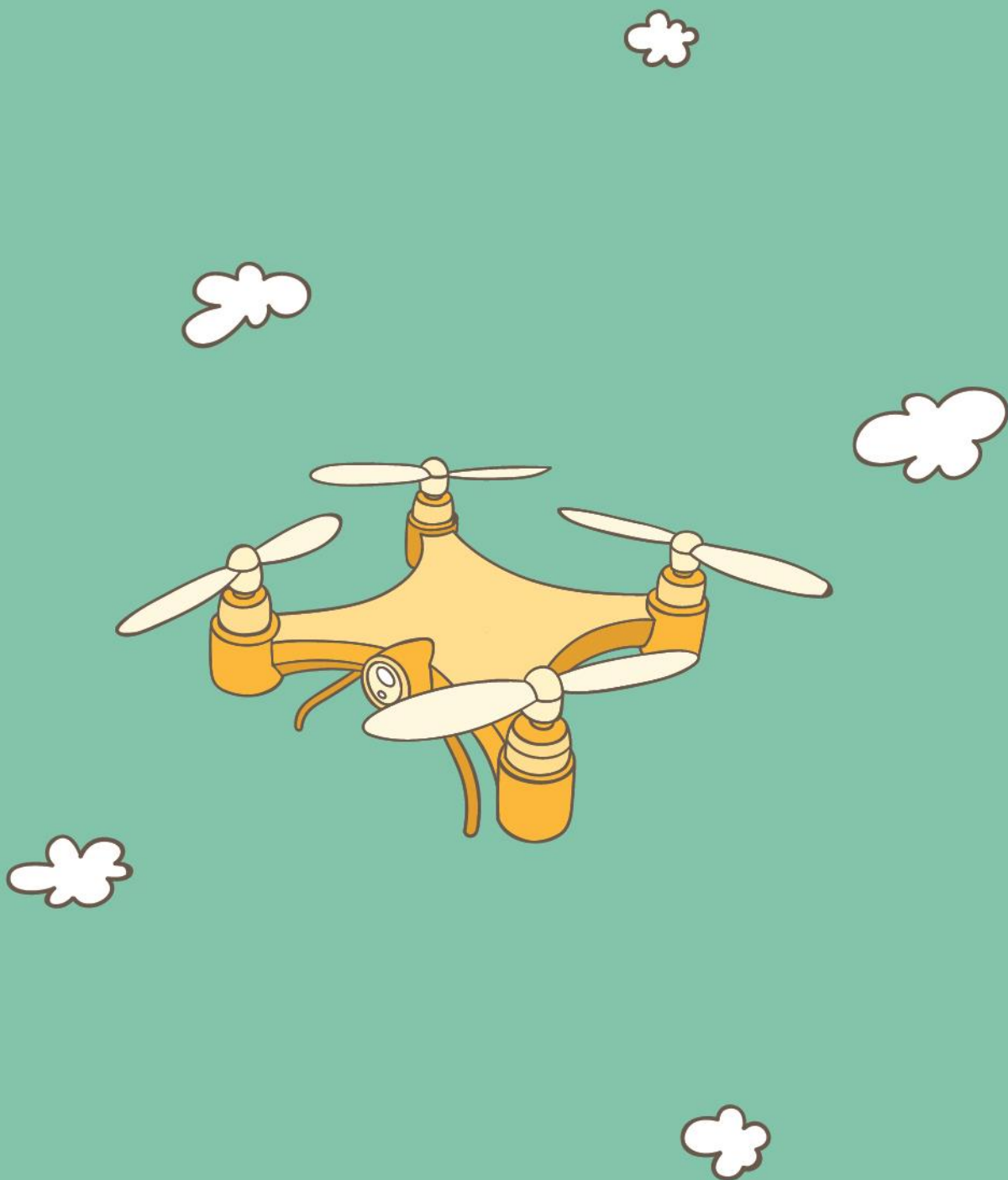
```

```

45 void printNumber() {
46   for (int i = 0; i < cnt; i++) {
47     Serial.print(num[i]);
48   }
49   for (int i = 0 ; i < 3 - cnt ; i++) {
50     Serial.print(random(10));
51   }
52   Serial.println();
53 }

```

<그림3-5> 슬롯 머신 최종 코드 업그레이드



WHIT