
Arduino UNO R4 WiFi

Arduino IoT Cloud 사용 매뉴얼 - 응용



DEVICEMART

www.devicemart.co.kr

UNO R4 WiFi 보드의 내장 LED 매트릭스를
IoT 클라우드에서 제어해보는 응용 예제입니다.

1. 준비

하드웨어

[아두이노 UNO R4 WiFi](#)

소프트웨어

[아두이노 IoT 클라우드](#)

매뉴얼 - 설정 편을 먼저 진행하신 후

매뉴얼 - 응용 편을 진행해 주세요.

 [Arduino UNO R4 WiFi IoT Cloud 사용 매뉴얼 - 설정](#)

2. 변수 설정

IoT Cloud의 Things에 접속해 새 변수를 지정합니다.

ADD VARIABLE을 클릭하여

Name을 지정해주고 **Integer Number** 타입으로 지정해 줍니다.

변수는 원하는 이름을 지정하여 설정할 수 있습니다.

여기에서는 “**Led_Matrix**”라는 이름으로 설정하였습니다.

ThingsDashboardsDevicesIntegrationsTemplatesUPGRADE

Untitled

Cloud Variables

Variables are what you use to store data like temperature or a smart light switch.

ADD VARIABLE

Set webhook

Add variable

Name

Led_Matrix

 Sync with other Things 

Integer Number eg. 1

Declaration

`int led_Matrix;` 

Variable Permission 

☒ Read & Write

☐ Read Only

Variable Update Policy 

☒ On change

☐ Periodically

Threshold

0

ADD VARIABLE

CANCEL

Device

R4

bceb-47ed-8396-...

INO R4 WiFi

ne

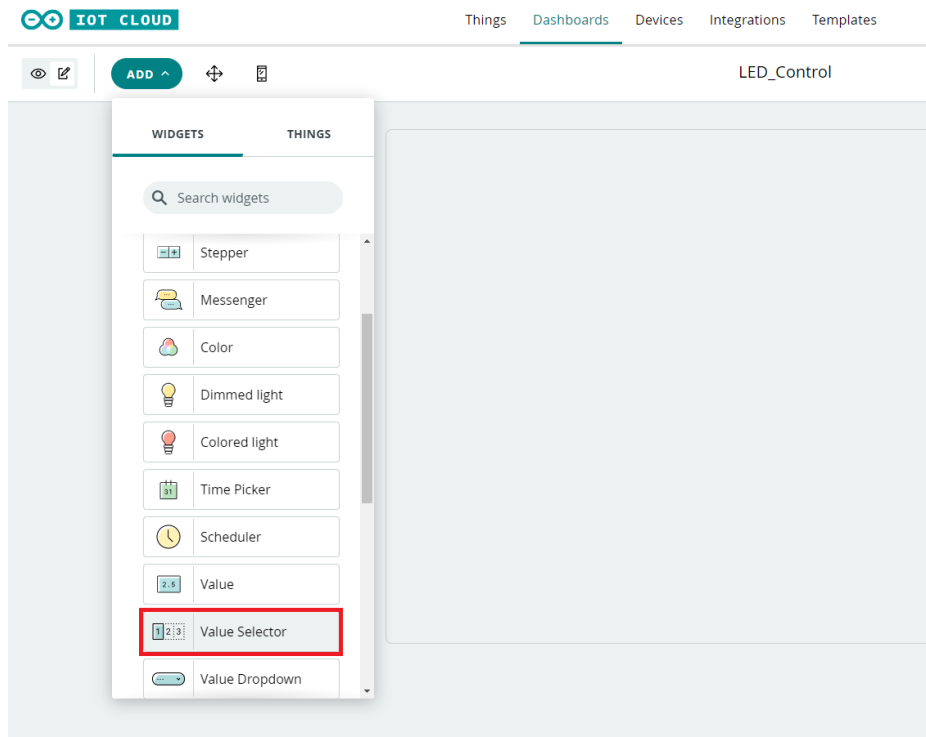


 Detach

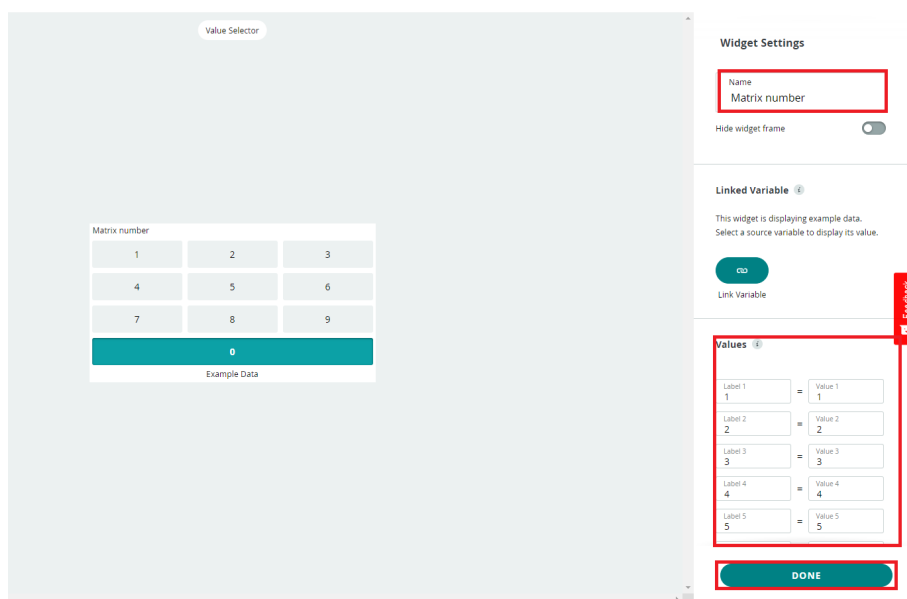
57b8...

3. 위젯 추가

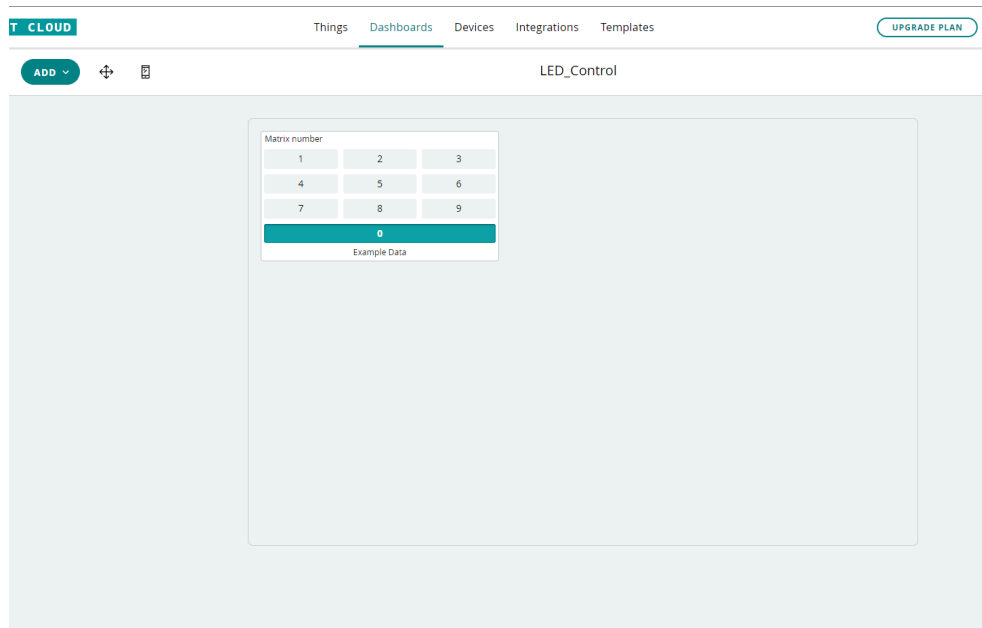
새 위젯을 추가할 **Dashboards**로 이동합니다.
대시보드 화면에서 **“Value Selector”** 위젯을 선택합니다.



새 위젯과 연결하려는 **Name**을 변경한 후
Values 값을 지정하고 **DONE**을 클릭합니다.



완료하면 대시보드 화면이 다음과 같이 보입니다.

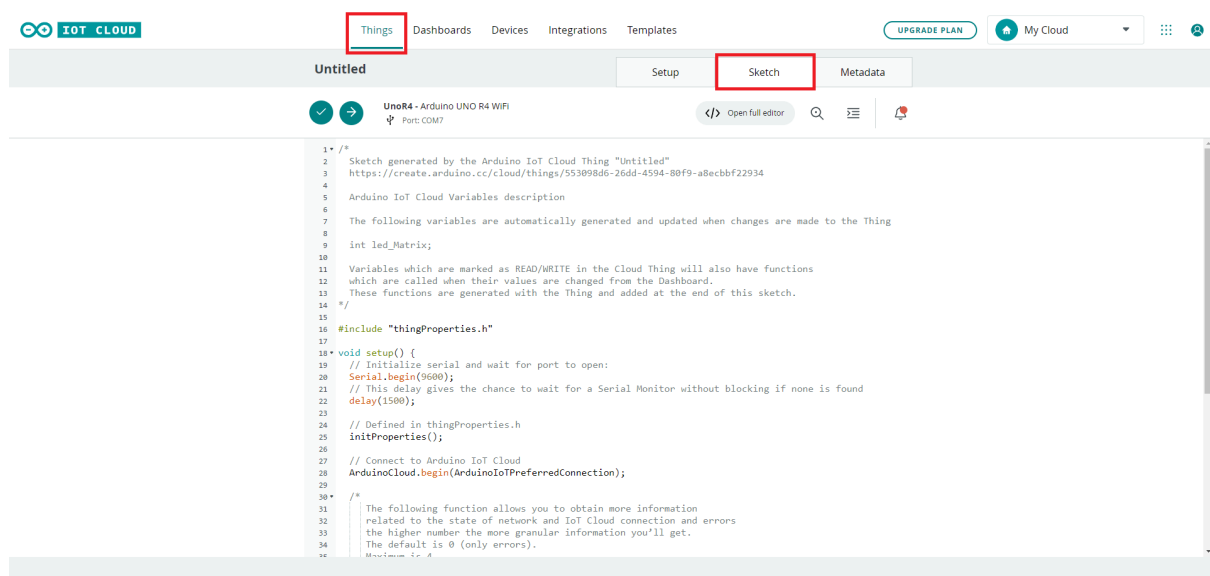


4. Sketch

Arduino IoT Cloud는 프로젝트에 사용할 대부분의 코드를 작성해주어 사용자가 편리하게 이용할 수 있도록 도움을 줍니다.

사용자 정의 코드만 간단하게 수정하면 프로젝트를 완성할 수 있습니다.

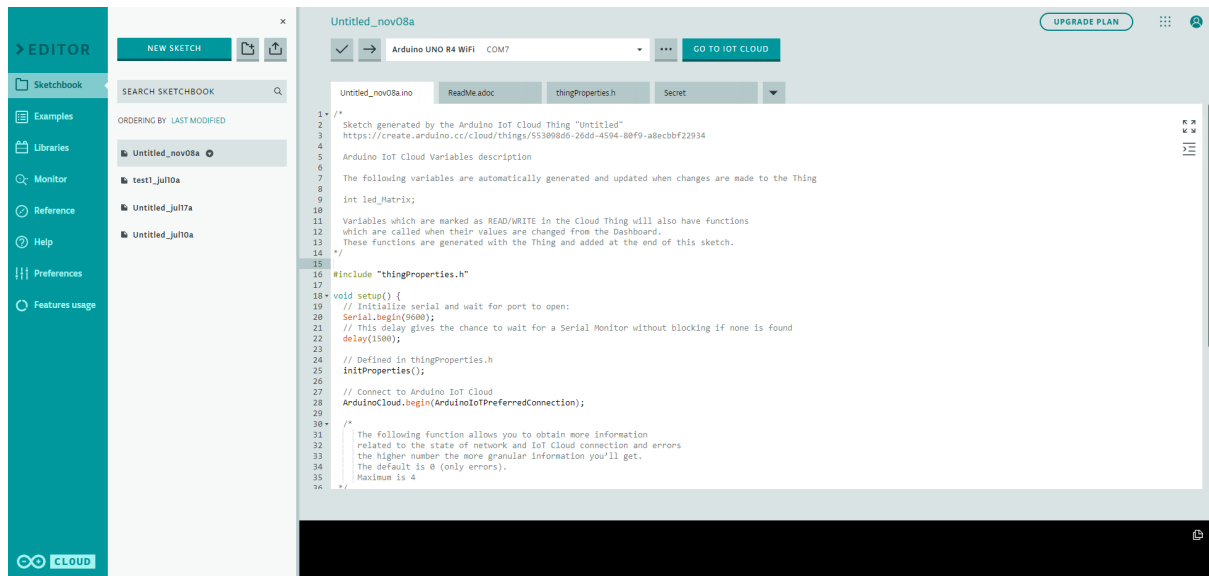
먼저 **Things** 탭으로 이동하여 **Sketch** 탭을 클릭해보면 구성항 정보를 기반으로 작성된 스케치를 확인할 수 있습니다.



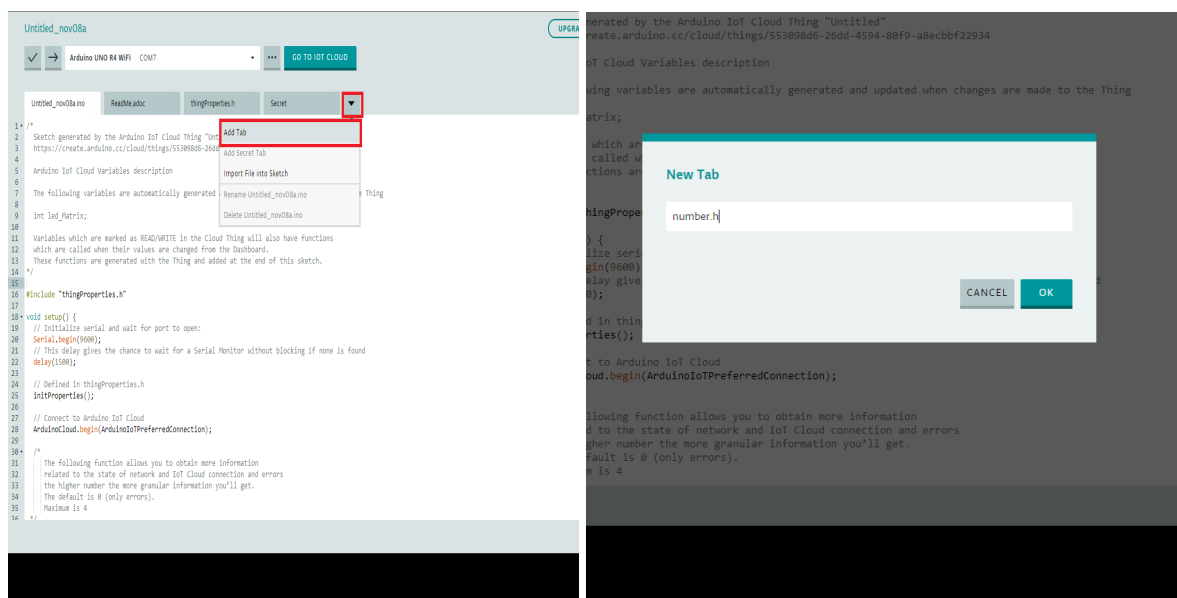
Open full editor를 클릭하면 수정 가능한 온라인 편집기로 이동합니다.



아래는 온라인 편집기 화면입니다.



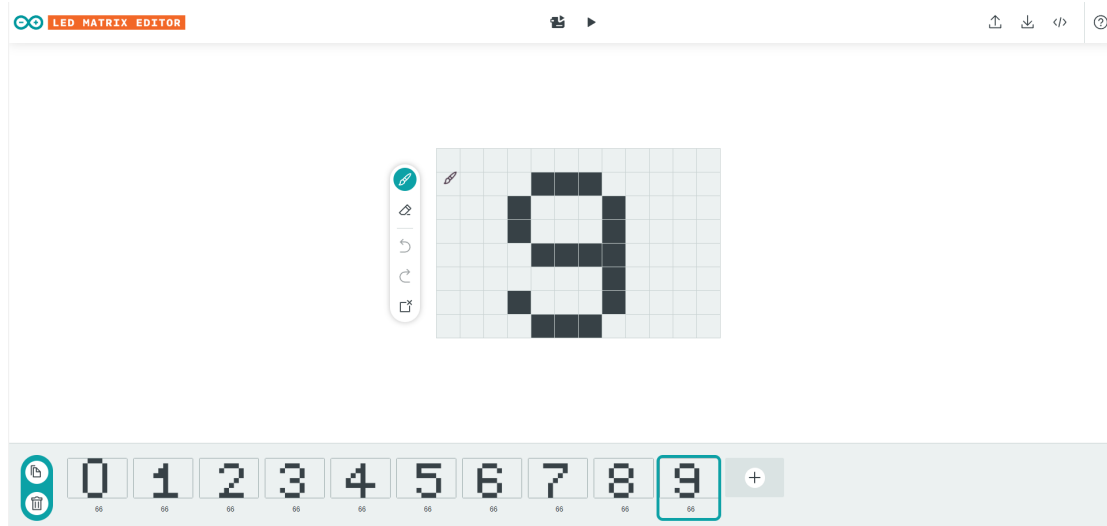
Add Tab을 클릭하여 새 탭을 추가합니다.
새 탭의 이름은 “number.h” 로 지정합니다.



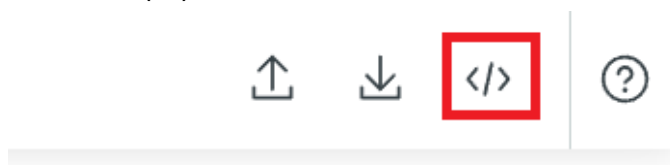
새로 추가한 탭에는 매트릭스 제어 코드를 입력해야 합니다.

매트릭스 제어를 위해 숫자 행렬을 생성해주어야 하는데
Arduino에서 제공하는 **LCD** 매트릭스 편집 도구를 이용하면 0~9까지 숫자
행렬을 쉽게 생성할 수 있습니다.

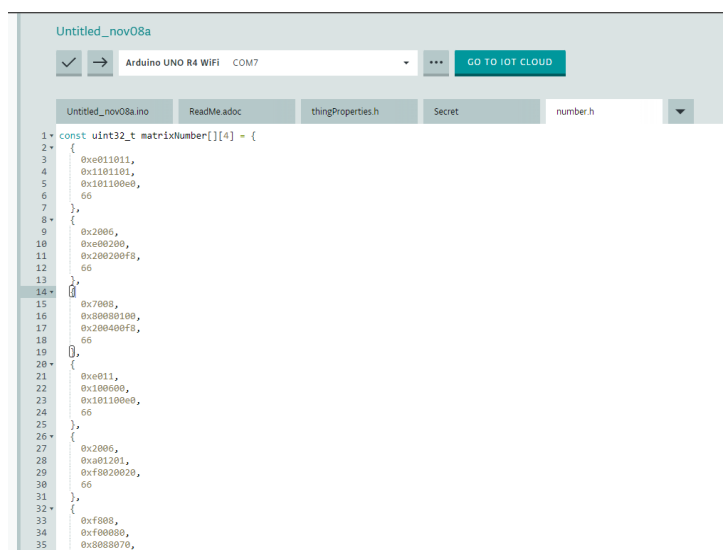
<https://ledmatrix-editor.arduino.cc/>



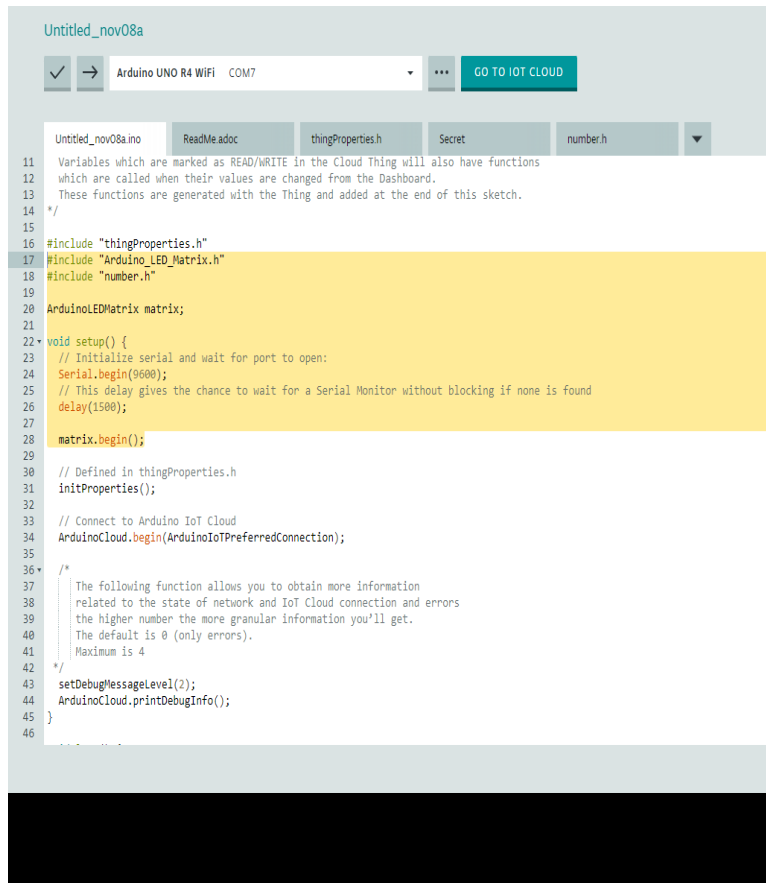
완성 후 저장을 하면
헤더파일(.h) 포맷의 파일이 저장됩니다.



이 코드를 IoT 편집기에서 생성했던
“number.h” 파일에 붙여넣어 줍니다.



그 다음 기본 스케치를 수정해 줍니다.



"**Arduino_LED_Matrix.h**" 라이브러리와 새로 생성했던 "**number.h**" 파일을 포함시켜 주기 위해 다음과 같이 작성합니다.

```
#include "Arduino_LED_Matrix.h"
#include "number.h"
```

LED 매트릭스를 사용하기 위해 객체를 생성합니다.

```
ArduinoLEDMatrix matrix;
```

실행을 위해 초기화 코드를 **setup** 함수에 작성해 줍니다.

```
matrix.begin();
```

loadFrame 메서드를 사용하면 IoT Cloud가 사용자에게 전달하는 번호를 디스플레이에 쓸 수 있습니다

```
void onLedMatrixChange() {
  matrix.loadFrame(matrixNumber[led_matrix]);
}
```

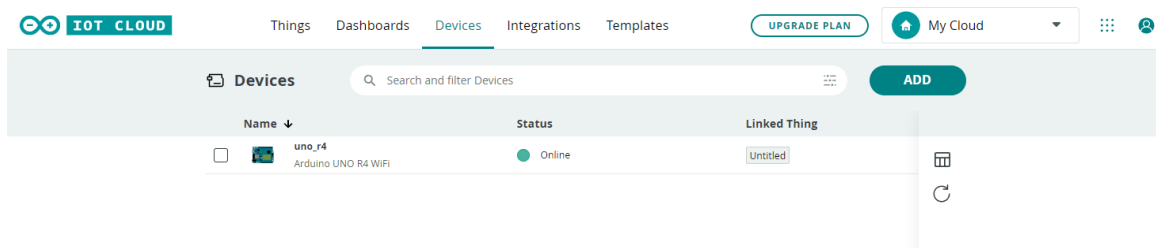
```

46
47 void loop() {
48   ArduinoCloud.update();
49   // Your code here
50
51 }
52
53
54
55
56 /*
57  Since LedMatrix is READ_WRITE variable, onLedMatrixChange() is
58  executed every time a new value is received from IoT Cloud.
59  */
60 void onLedMatrixChange() {
61   // Add your code here to act upon LedMatrix change
62   matrix.loadFrame(matrixNumber[led_Matrix]);
63 }

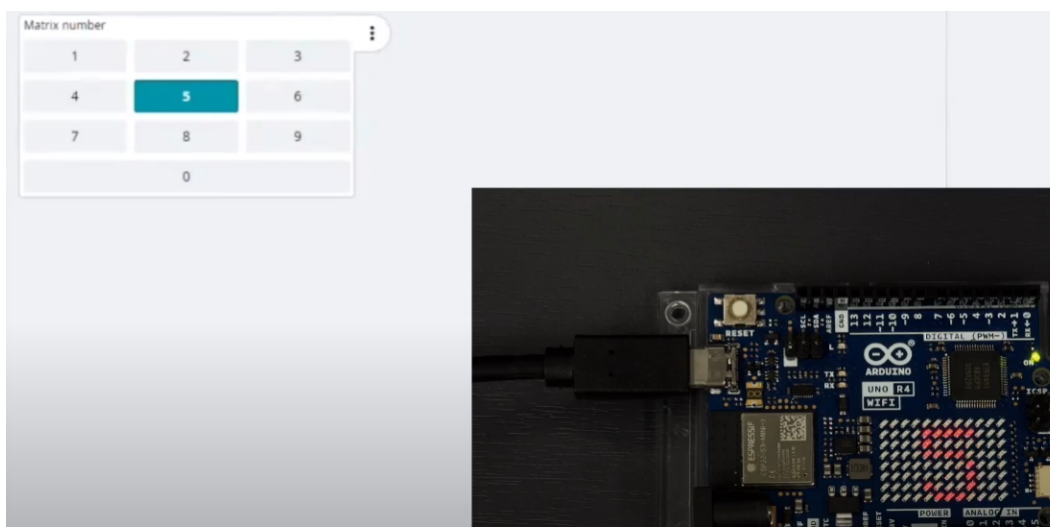
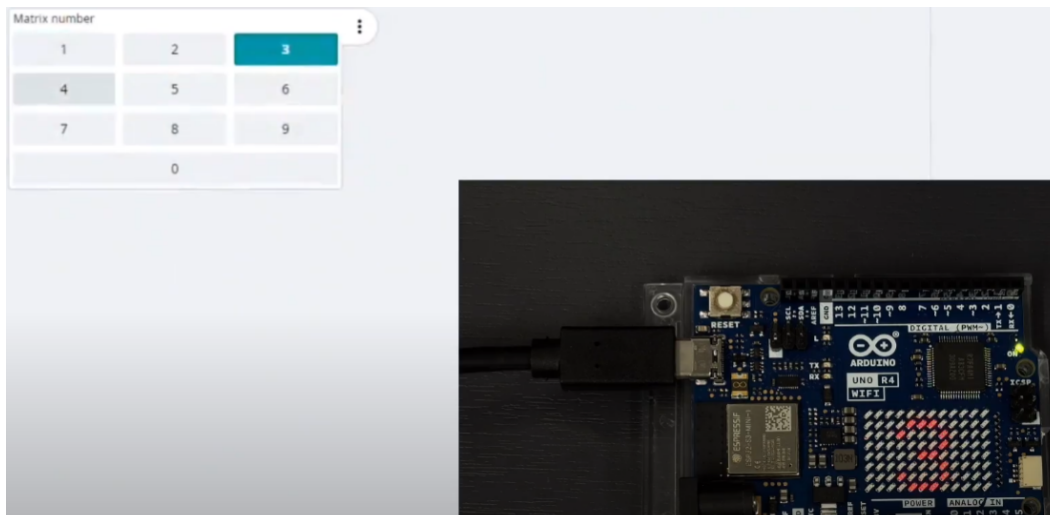
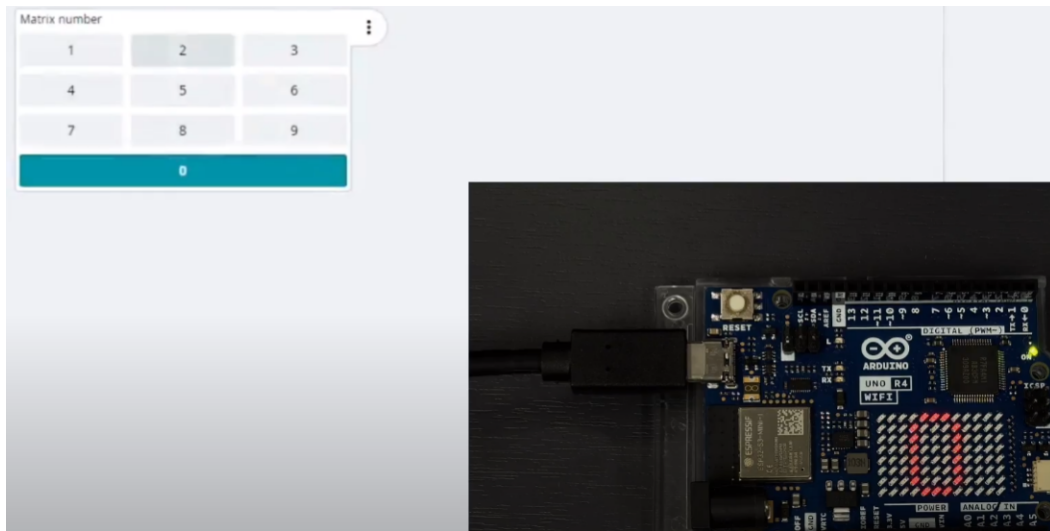
```

코드 수정을 완료 하였으면 스케치를 보드에 업로드 합니다.

Arduino Uno R4 WiFi IoT Matrix로 전송하고 장치가 다시 온라인 상태가 될 때까지 기다립니다.



5. 실행



실행 영상 :

https://drive.google.com/file/d/1G-Db_LUAIUZPhFsBlp0uxf-UaVW-l63b/view?usp=sharing