

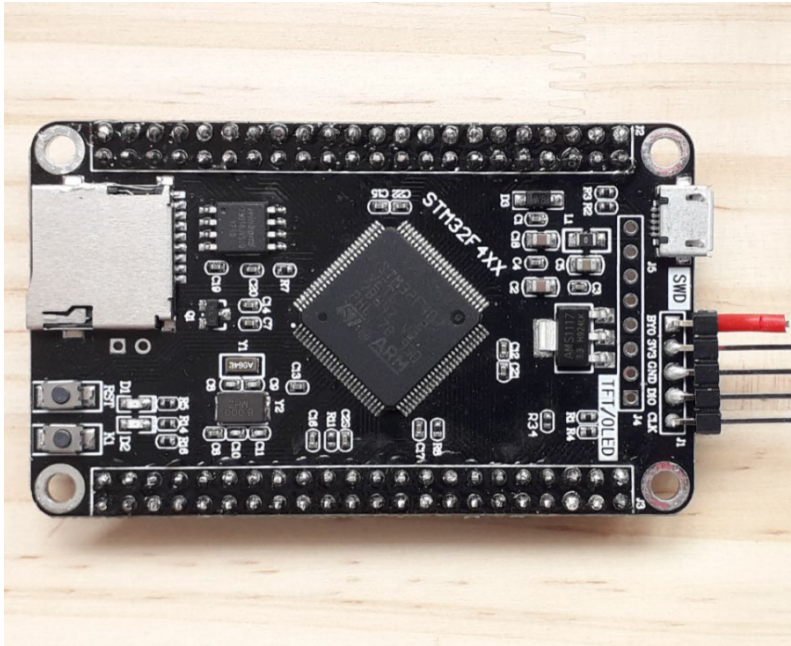
## 사전준비 prework.pdf

2022-06-20 지규선

### ■ 사전 준비 0 단원 별 준비물

### ■ 사전 준비 1 **stm32F407VET Breakout** 보드 완성하기

빨간 전선 피복? 가끔 st-LINK V2 를 반대로 연결하는 경우가 있는데요, 그 예방차원에서 기준을 표시해 놓은 거죠.

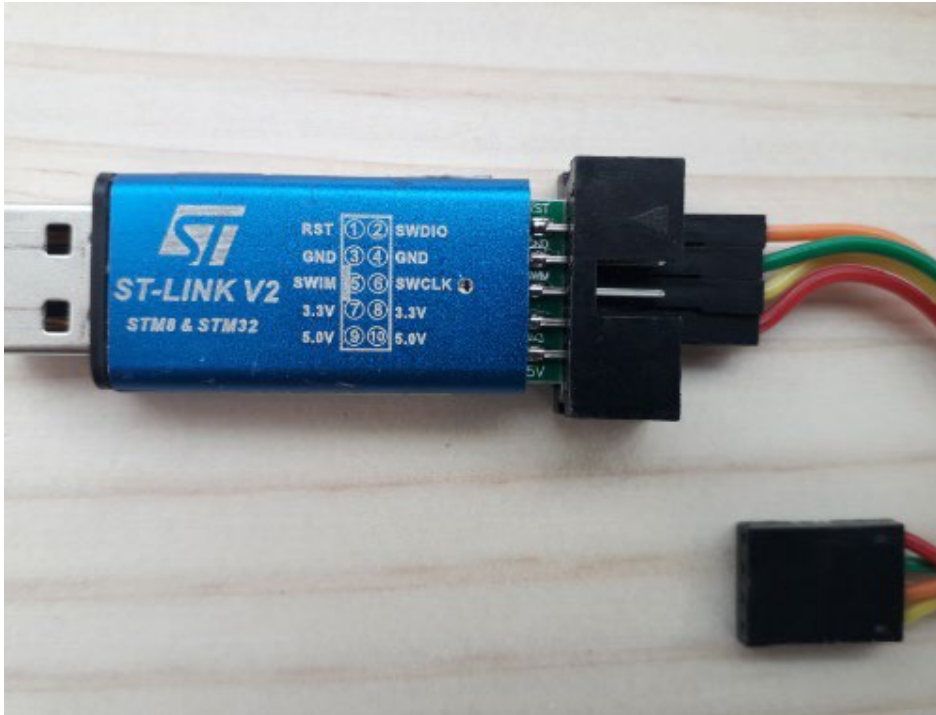


### ■ 사전 준비 2 업로드 테스트

#### 2-1. st-Link V2 세팅

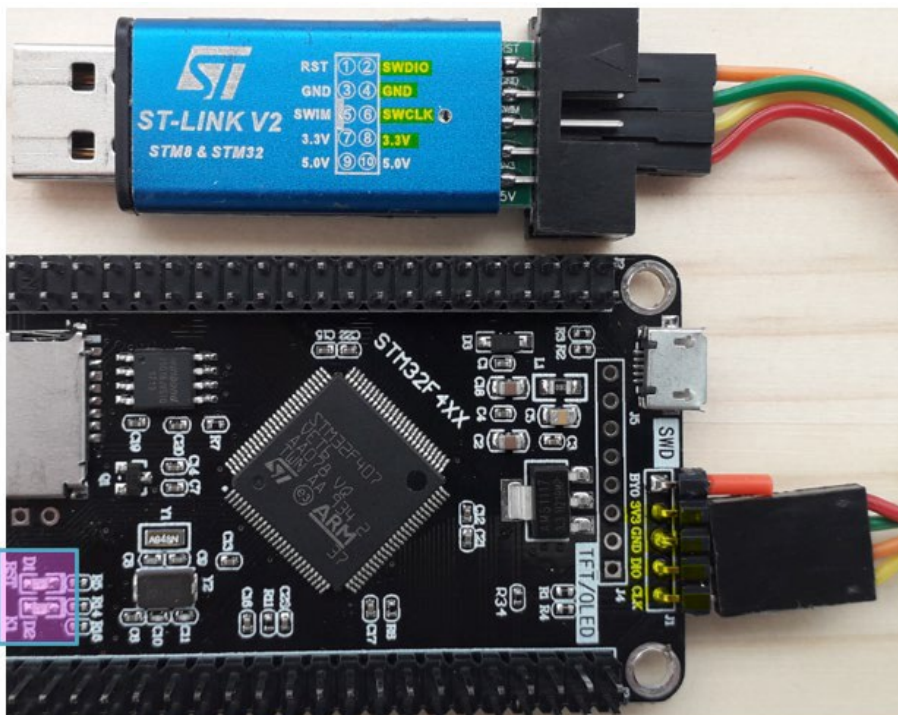
SWIM | SWD? SWIM 은 stm8 용입니다. 우리가 사용하는 임베디드용 보드는 stm32 이므로 SWD 에 연결합니다. st-Link V2 를 구매하면 점퍼와이어도 함께

오는데요, 3V3 : 빨강, GND: 초록, IO: 주황, SWCLK: 노랑으로 연결합니다.  
물론 사용자에 따라 달라질 수 있어요.

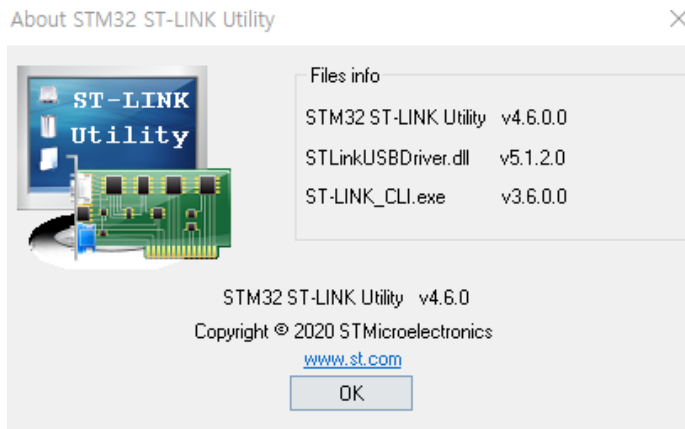


## 2-2. stm32F407VET6 과 st-Link V2 연결

사진의 좌하단 한 개의 LED 만 켜지면 OK!



## 2-3. st-Link Utility 설치



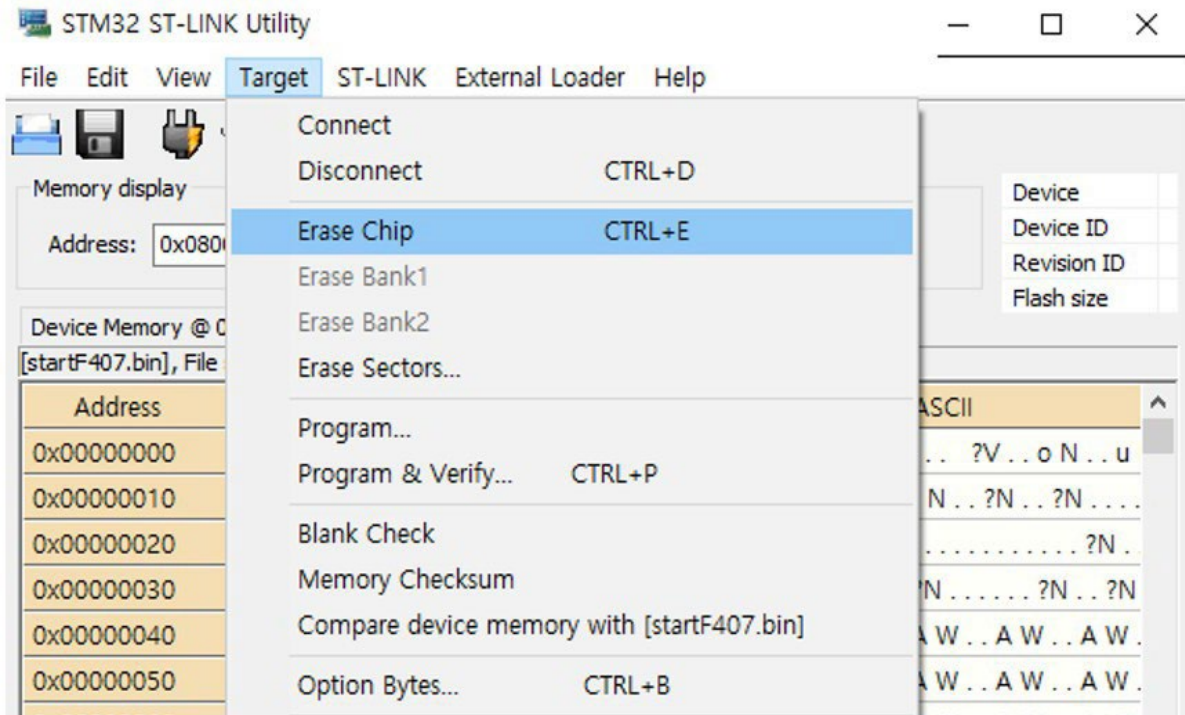
만일 dll 파일 문제가 생기면 Microsoft Visual C++ 2015(~ 2018) Redistributable Update 를 설치하세요. **꼭 32 비트용으로요!** 아마도 st-LINK Utility 가 32 비트 S/W 인가 봅니다.

Microsoft Visual C++ 2015 Redistributable Update 3 RC

Select Language: English ▼ Download

<input type="checkbox"/>	File Name	Size
<input type="checkbox"/>	vc_redist.x64.exe	14.3 MB
<input checked="" type="checkbox"/>	vc_redist.x86.exe	13.5 MB

## 2-4. (st-Link utility 를 이용한) stm32F407VET6 보드 초기화



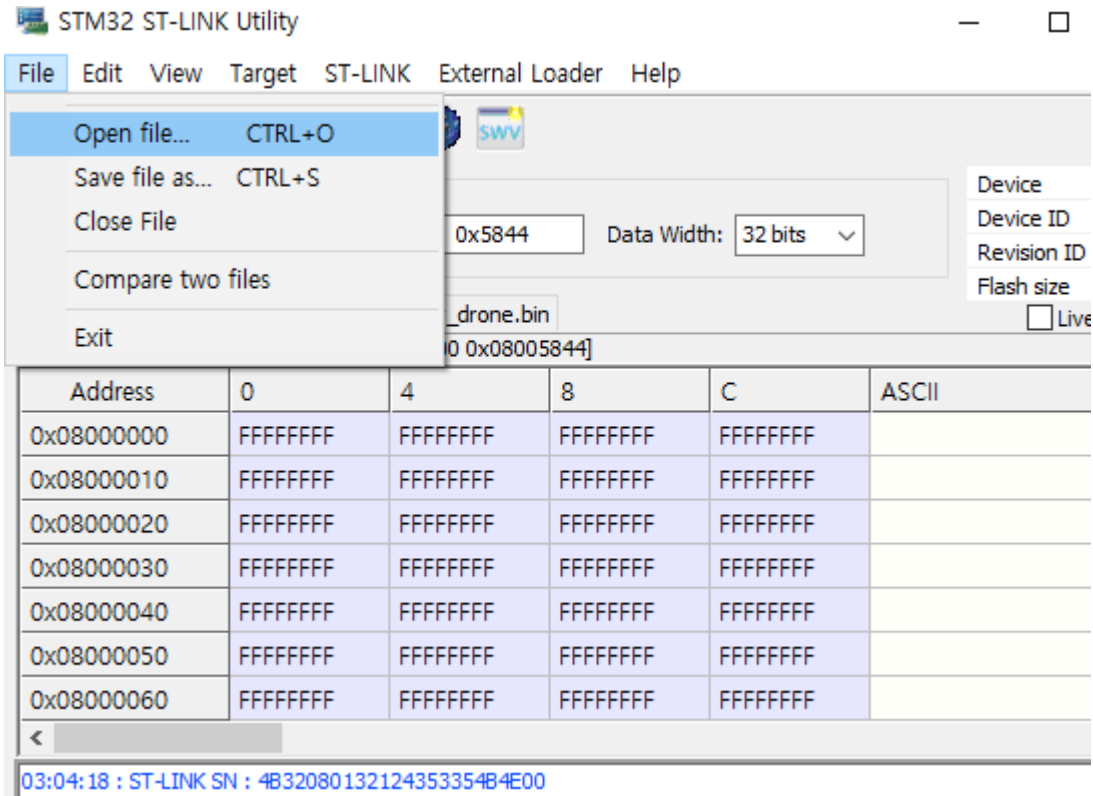
[Target] - [Erase Chip]하면, 다음처럼 stm32F407VET | VGT Flash 메모리가 초기화됩니다.

Address	0	4	8	C
0x08000000	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x08000010	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x08000020	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x08000030	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x08000040	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x08000050	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x08000060	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x08000070	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x08000080	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF

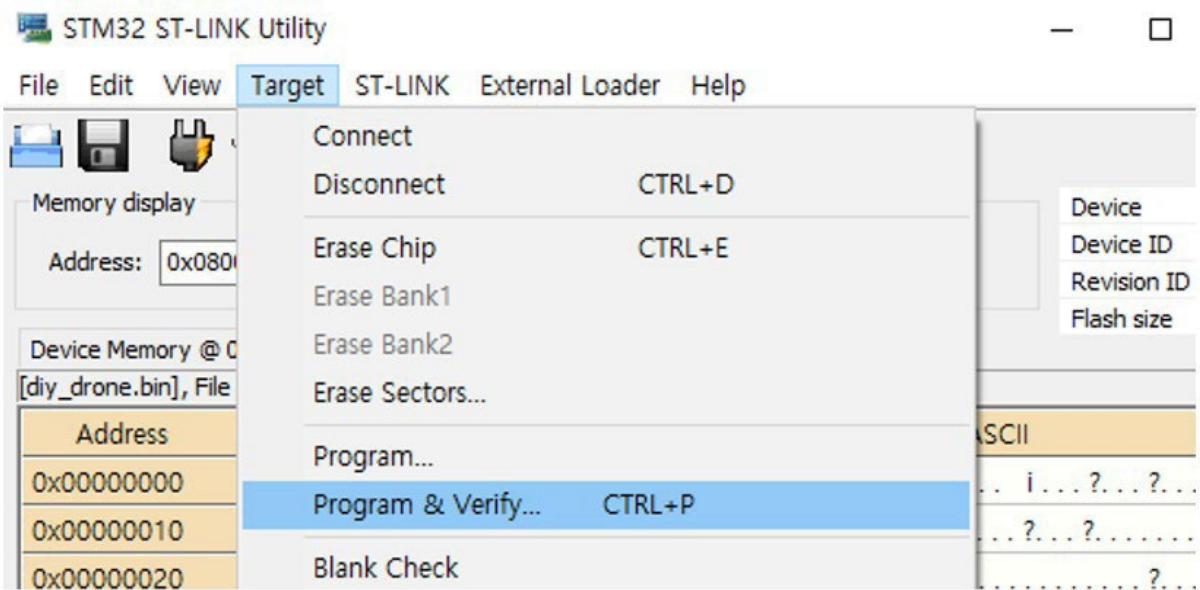
이제부터 stm32CubeIDE 에서 만들어진 펌웨어(Hex | Bin 파일)를 PC -> FC 로 업로드를 할 수 있어요.

## 2-5. 업로드

- ① 업로드할 펌웨어(Hex | Bin 파일)를 선택하세요.



- ② [Target] - [Program & Verify] 하면 업로드가 완료됩니다.





### ■ 사전 준비 3 stm32CubeIDE 설치

(2022. 6 월 현재) 1.9.0 버전이네요. 자신의 OS 에 맞게 다운로드 및 설치하세요.

### ■ 사전 준비 4 아두이노 설치

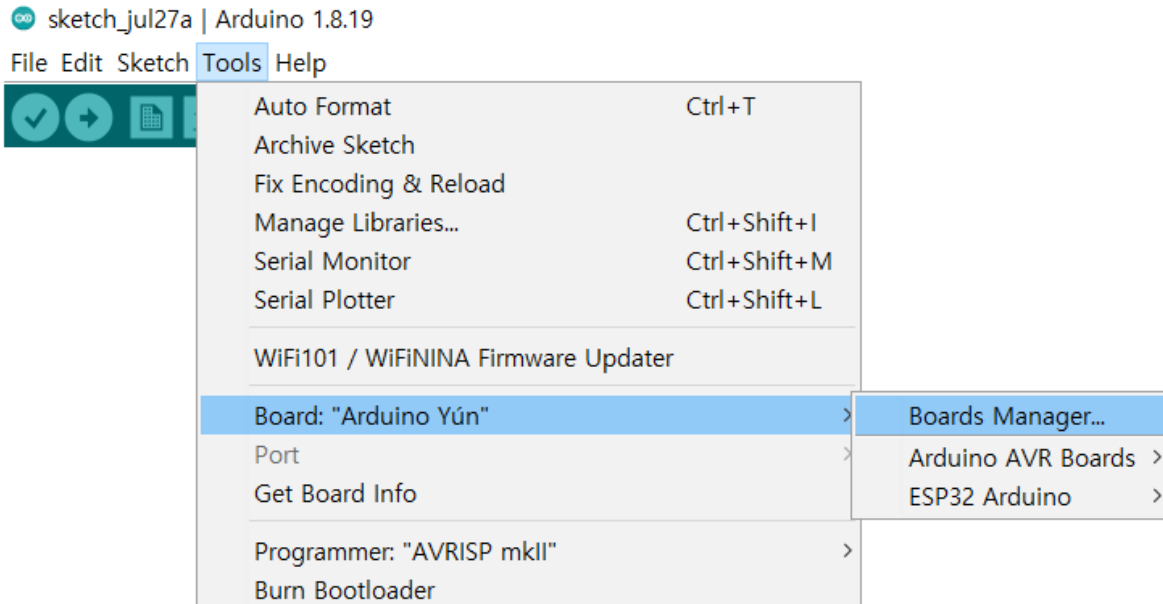
#### 4-1. esp32 용 라이브러리 추가

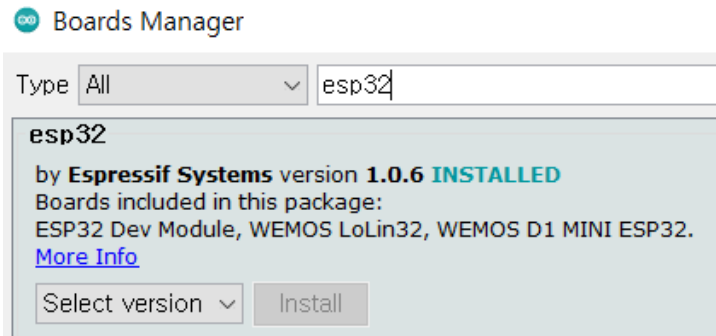
① 아두이노 IDE [환경설정]에서 [추가적인 보드 URLs]에 다음 줄을 넣고 [확인]합니다.

- [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

추가적인 보드 매니저 URLs `https://dl.espressif.com/dl/package_esp32_index.json`

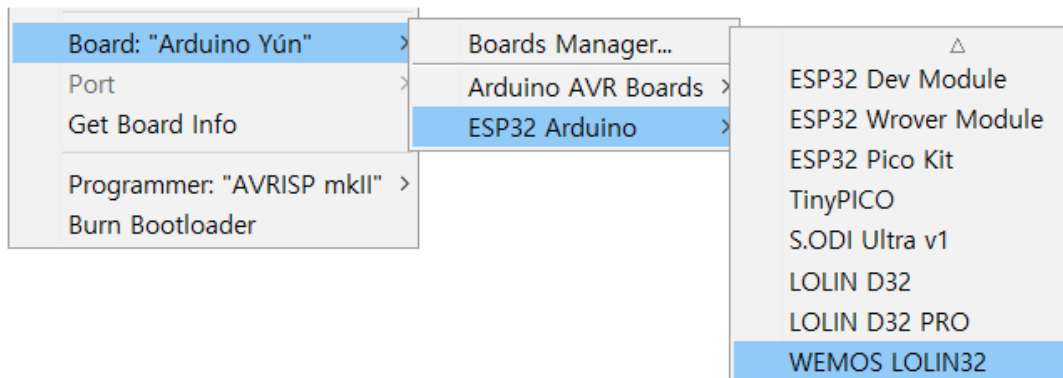
② [툴] - [보드] - [보드매니저]에서 **esp32**를 설치합니다.





③ [툴] - [보드] - [esp32 Arduino]에서 **WEMOS LOLIN32**를 선택합니다.

업로드 속도, CPU 클럭 등은 기본값을 그대로 사용합니다.






#### 4-2. esp32 용 WebSocket 라이브러리 추가

<https://github.com/Links2004/arduinoWebSockets>

① 다운로드하고 압축을 풀어 아두이노 [라이브러리] 폴더에 넣습니다.

(E:) > arduino-1.8.19 > libraries

 Adafruit\_Circuit\_Playground  
 arduinoWebSockets-master  
 Bridge

참, 아두이노를 다시 시작해야 새 라이브러리가 적용됩니다.

## ■ 사전 준비 5 안드로이드 스튜디오

### 5-1. [안드로이드 스튜디오](#) 설치

Android 스튜디오 Bumblebee (2021.1.1) Canary

### 5-2. 구글 맵 사용을 위한 앱키 얻기

우리는 다중 목적지 비행을 구현하기 위해 구글맵이 필요합니다.

#### > 새 프로젝트 만들기

① [API-Key 얻는 사이트](#)에 들어가요. 구글 계정이 필요합니다.

② [Select a project]를 클릭합니다.



③ [NEW PROJECT]를 클릭합니다.

Select a project  **NEW PROJECT**

④ 프로젝트 이름을 넣고 CREATE 버튼을 클릭합니다. 그럼 새 프로젝트가 만들어져요.

Project name \*  ?

Location \*  BROWSE

Parent organization or folder

**CREATE** CANCEL



## &gt; 출금 계좌(청구계정) 만들기

① (새 프로젝트가 만들어지면) 출금계좌를 새로 만들라는 창이 떠요

## No active billing accounts

You cannot enable billing for this project because all your billing accounts are closed.  
Please visit the billing section to correct your account state, then try again to enable billing.

CANCEL

MANAGE BILLING ACCOUNTS

참고로, 매월 \$200 크레딧을 무료로 제공한답니다. 결국, 일반 사용자들의 경우, 예전처럼 비용 부담 없이 그대로 사용할 수 있어요.



② 출금 계좌를 새로 만드세요.

CREATE ACCOUNT

Filter

Status : Active

Enter property name or value

③ 원하는 계좌명을 넣고 [다음]으로 넘어 갑니다.

Name \*

Google\_why\_billing

Country \*

South Korea

CONTINUE

CANCEL

- ④ 나의 청구계정 내용을 확인하고 마무리합니다.

**SUBMIT AND ENABLE BILLING**

참, (출금 계좌용) 카드 등록 과정도 거치게 되는데, 여기서는 생략했어요.

- ⑤ (새로운 화면에서) 청구계정 관리를 클릭합니다.

**Billing account** [Manage](#)

Google Why Billing, [REDACTED]

7CA9B7

**Organization**

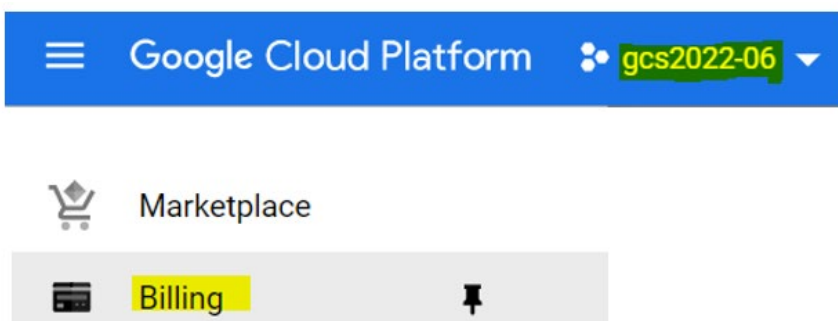
No organization

- ⑥ 청구계정(출금계좌)에 연결된 프로젝트가 없네요. 구글 맵 프로젝트를 사용하려면 내 계좌를 연결해 놓으라는 거죠.

## Projects linked to this billing account

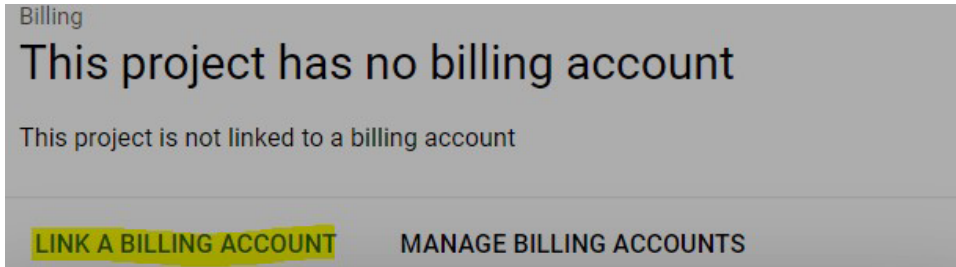
Project name	Project ID	Actions ?
There are no projects linked to this billing account.		

- ⑦ 내가 만든 프로젝트를 선택하고 [Billing]을 클릭합니다.



⑧ 내 카드와 프로젝트를 연결합니다.

- LINK A BILLING ACCOUNT
- 내 청구계정 이름 선택
- SET ACCOUNT



### Set the billing account for project "gcs-app"

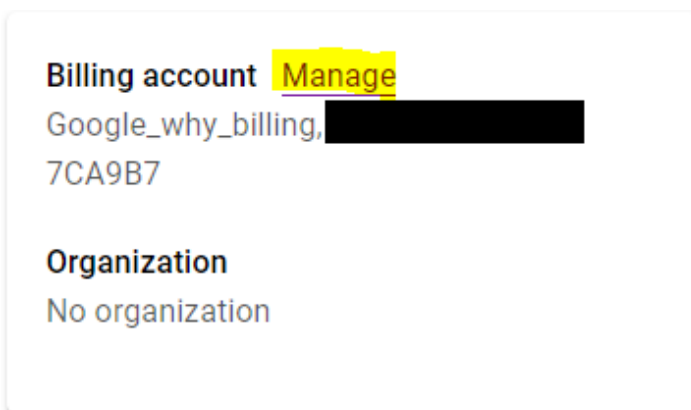
Billing account \*

Google\_why\_billing ▼ ?

Any charges for this project will be billed to the account you select here.

CANCEL [SET ACCOUNT](#)

⑨ 다시 청구계정 관리를 클릭해보면 둘이 서로 연결된 것을 볼 수 있어요.



### Projects linked to this billing account

Project name	Project ID	Actions ?
gcs-app	gcs-app-351802	⋮

## > 앱 키(App Key) 얻기

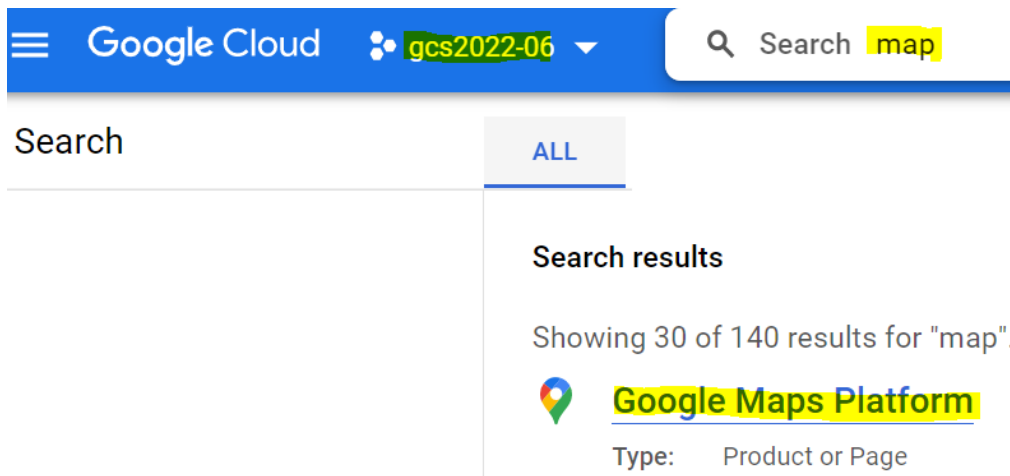
프로젝트를 만들어 내 계좌를 연동시켜 놓았으니, 이제 앱 키를 만들 수 있습니다.

- ① 프로젝트명을 선택하고,



- ② 'credentials'로 들어 갑니다.

credentials 찾기가 쉽지 않을 때는,  
[내 프로젝트]를 선택한 다음 'map'을 검색하여 Google Maps Platform을 클릭하세요.



- ③ 앱 제한 여부를 정해 앱키를 만듭니다.

- ❶ 앱을 아무 컴퓨터에서나 만들어 배포할 수 있게 함
- ❷ (우리의 경우) 앱을 특정 컴퓨터에서만 만들어 배포할 수 있게 함

❶ 앱 제한 없음

### Application restrictions

☒ None

### API restrictions

API restrictions specify the enabled APIs that this key can call

☒ Don't restrict key  
This key can call any API

☐ Restrict key

Note: It may take up to 5 minutes for settings to take effect

SAVE

CANCEL

## ② 앱 제한 있음

### Application restrictions

☒ Android apps

### Restrict usage to your Android apps

Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps

com.copter1.new20,  
01:3E:11: [REDACTED] AC:CC:B1:34:61 [REDACTED]

▼

ADD AN ITEM

인증서 지문

### API restrictions

API restrictions specify the enabled APIs that this key can call

☒ Don't restrict key  
This key can call any API

☐ Restrict key

Note: It may take up to 5 minutes for settings to take effect

SAVE

CANCEL

④ (인증서 지문을 넣고) 저장하면 앱키가 바로 생성됩니다.

## > 인증서 지문 얻는 법

- ① [API-Key 얻는 사이트](#)에 들어갑니다.
- ② [Credentials] – [Maps API Key]를 클릭합니다.
- ③ [Debug certificate fingerprint]에서 인증서 지문을 복사(Copy to clipboard)합니다.



## Debug certificate fingerprint

For Linux or macOS:

```
$ keytool -list -v -keystore ~/.android/debug.keystore -alias and
```

For Windows:

```
$ keytool -list -v -keystore "%USERPROFILE%\android\debug.keystore"
```

Copy to clipboard

- ④ 윈도우 검색창에서 cmd를 실행시켜, .android 폴더로 이동합니다.

```
C:\Users\gyoos\android>
```

- ⑤ 클립보드에 복사해 두었던 인증서 지문을 붙인 다음 엔터합니다.  
 ⑥ 출력된 결과중에서 SHA1 을 복사하여 앱의 인증서 지문에 붙여 넣습니다.

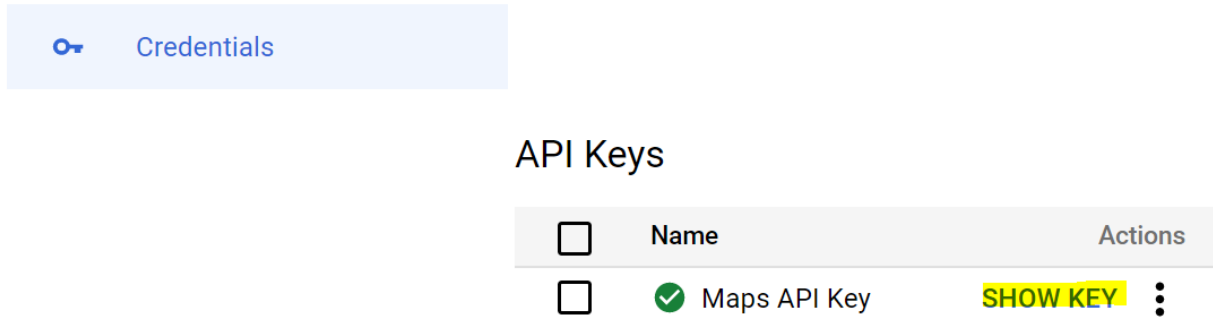
```
Certificate fingerprints:
  SHA1: 01:3E:11:11:97:32:AC:CC:B1:34:61:6E:9
  SHA256: 15:37:FD:27:B6:64:16:51:53:34:D1:30
Signature algorithm name: SHA1withRSA (weak)
Subject Public Key Algorithm: 2048-bit RSA key
```

com.copter1.new20,  
 01:3E:11 [REDACTED] AC:CC:B1:34:61 [REDACTED]

ADD AN ITEM

## > 앱 키(App Key) 활용

① [Credentials] – [SHOW KEY]에서 앱키를 복사합니다.



② [안드로이드 스튜디오] [AndroidManifest.xml] 앱키를 붙입니다.

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value=" " />
```

### 5-3. 안드로이드 스튜디오에서 앱 만들기

앱 사용을 특정 PC로 제한했다면,  
이제부터 앱을 새로 만들 때는 ‘앱 제한에 사용된 패키지 이름’을 반드시 사용해야 합니다.

- com.copter1.new20

