

# Tinker board

IoT를 위한 최선의 엣지 단말기



MAKER SPACE  
**G·CAMP**

# Android Debug Bridge



# Android App install - Android SDK platform install

## 1. SDK tool 다운로드

### 다운로드

Android 개발자는 Android 스튜디오의 **SDK Manager** 또는 **sdkmanager** 명령줄 도구에서 최신 SDK 플랫폼 도구를 가져와야 합니다. 그래야만 도구가 나머지 Android SDK 도구와 함께 올바른 위치에 저장되고 쉽게 업데이트됩니다.

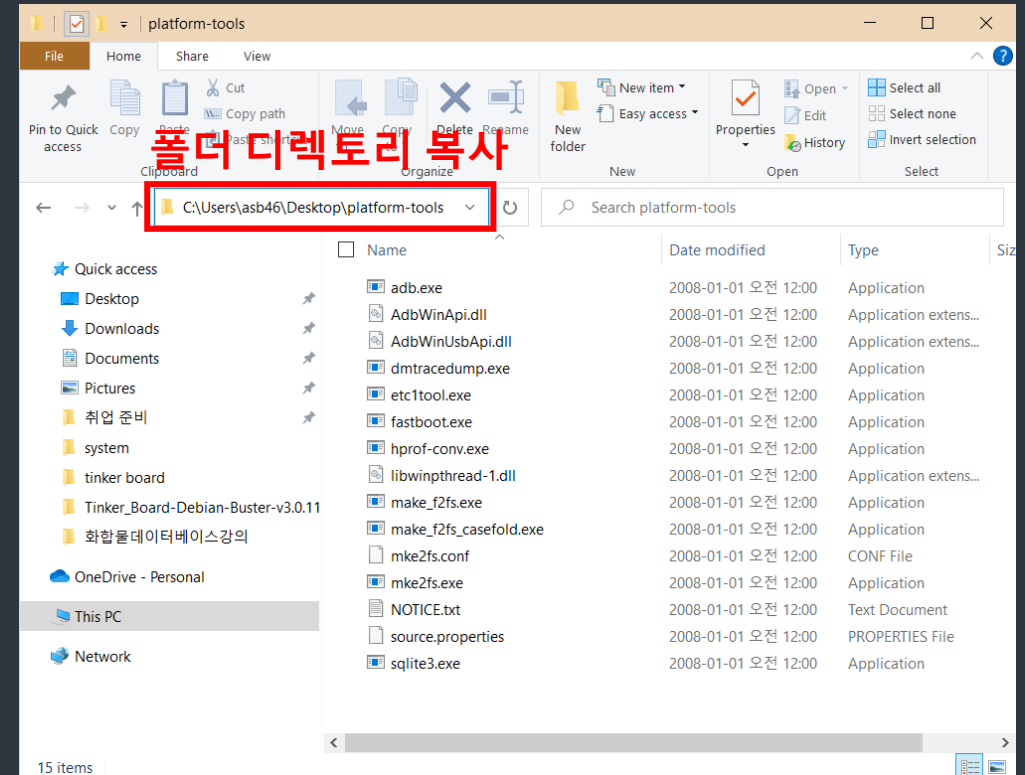
그러나 명령줄 도구만 필요한 경우 다음 링크를 사용하세요.

- [Windows용 SDK 플랫폼 도구 다운로드](#)
- [Mac용 SDK 플랫폼 도구 다운로드](#)
- [Linux용 SDK 플랫폼 도구 다운로드](#)

이러한 링크는 변경되지 않지만, 항상 최신 버전의 도구를 가리킵니다.

[Android SDK 설치 링크](#)

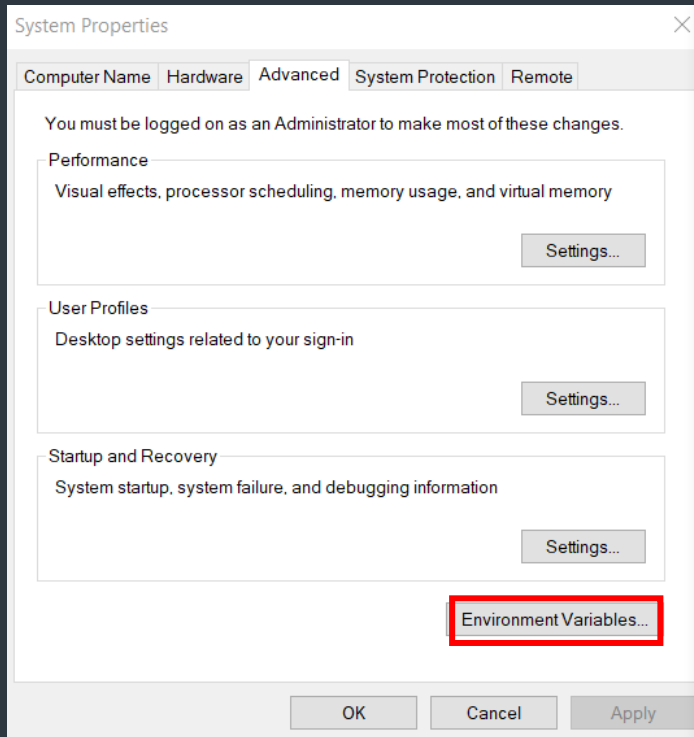
## 2. 다운로드 후 압축풀고 폴더 위치 복사



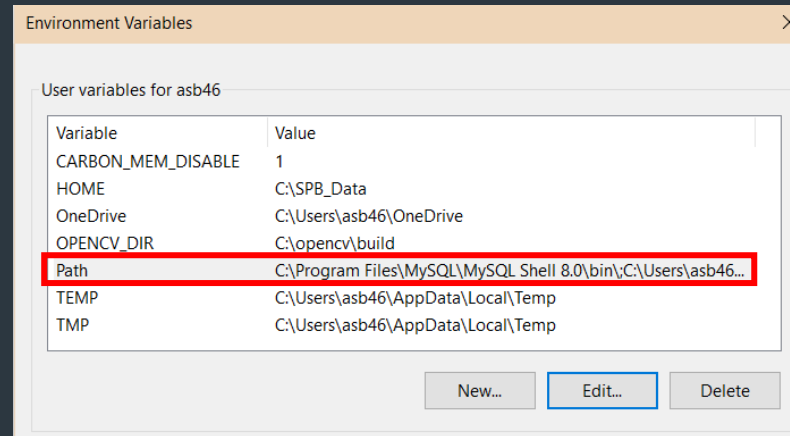
# Android App install - Android SDK platform install

## 3. SDK tool 다운로드

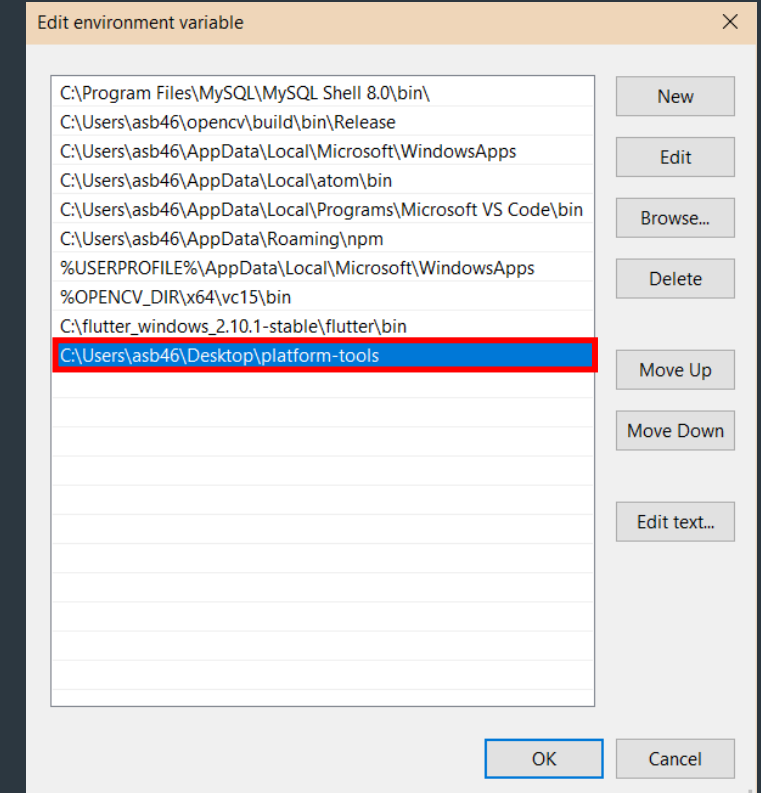
### 환경 변수 클릭



### Path 선택 후 “Edit” 클릭

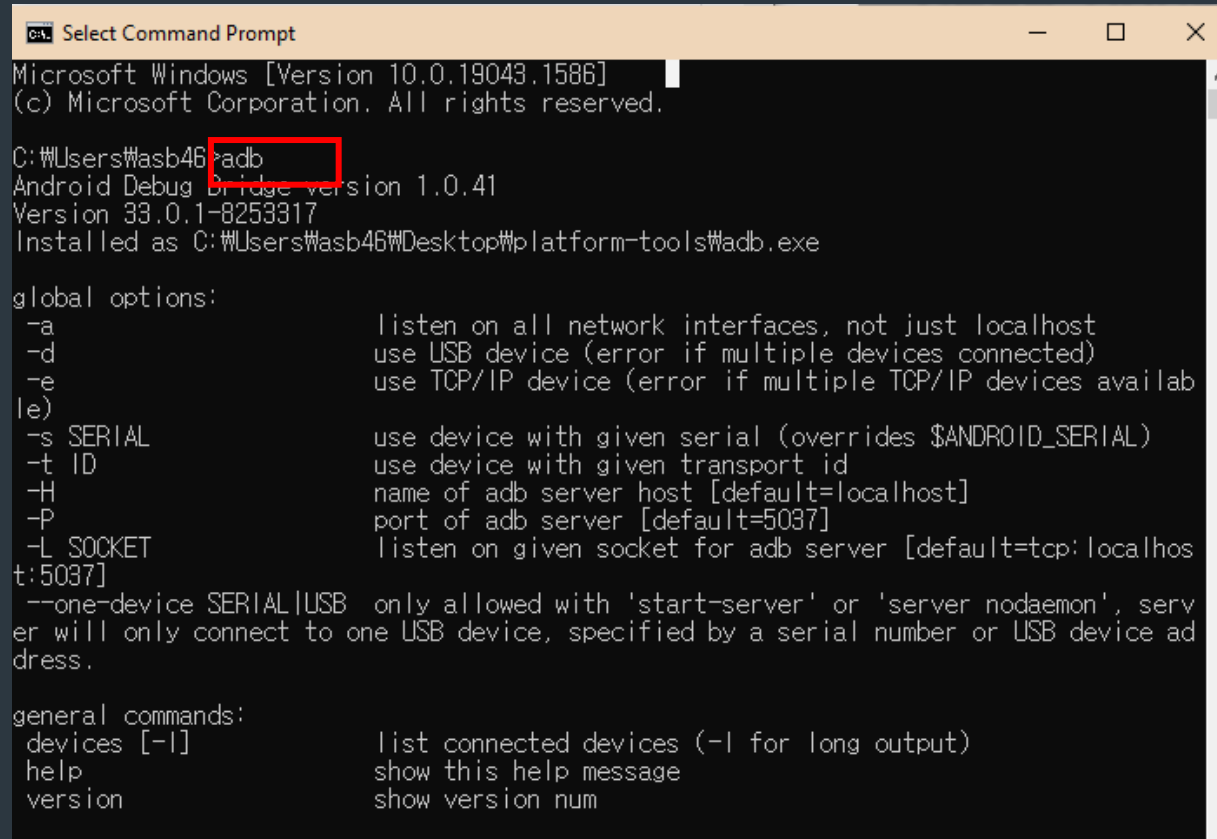


### 복사한 Android SDK 폴더 위치 추가



# Android App install - Android SDK platform install

## 3. adb 커맨드 동작 테스트



```
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\wasb46>adb
Android Debug Bridge version 1.0.41
Version 33.0.1-8253317
Installed as C:\Users\wasb46\Desktop\platform-tools\adb.exe

global options:
-a          listen on all network interfaces, not just localhost
-d          use USB device (error if multiple devices connected)
-e          use TCP/IP device (error if multiple TCP/IP devices available)
-s SERIAL   use device with given serial (overrides $ANDROID_SERIAL)
-t ID       use device with given transport id
-H          name of adb server host [default=localhost]
-P          port of adb server [default=5037]
-L SOCKET   listen on given socket for adb server [default=tcp:localhost:5037]
--one-device SERIAL|USB only allowed with 'start-server' or 'server nodaemon', server will only connect to one USB device, specified by a serial number or USB device address.

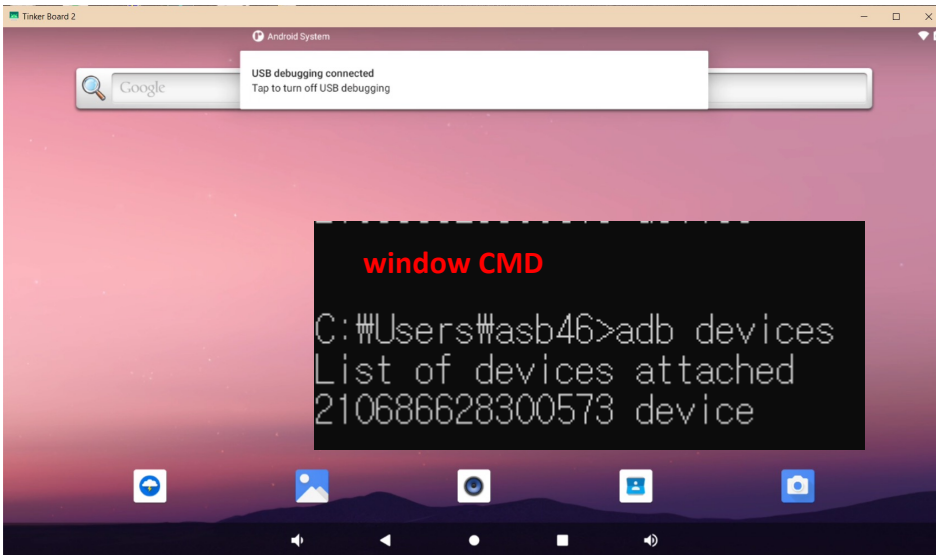
general commands:
devices [-l] list connected devices (-l for long output)
help        show this help message
version     show version num
```

# Android Debug Bridge (ADB)

ADB를 통해 안드로이드 디바이스와 PC가 통신하는 방법



개발 PC와 디바이스를 USB로 연결

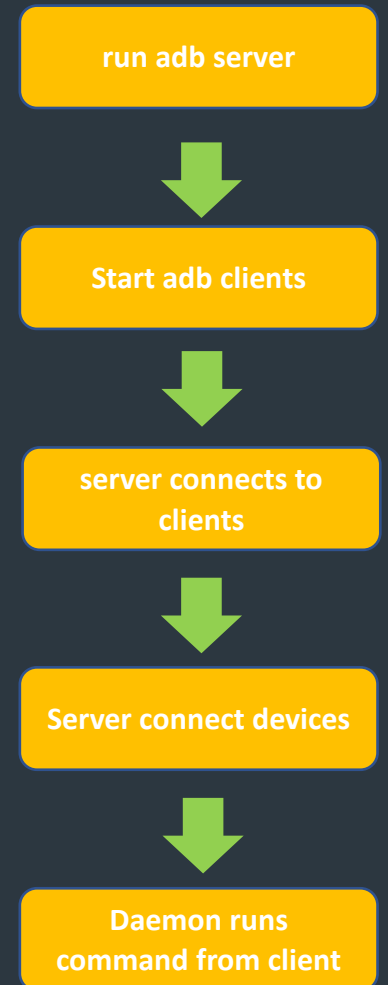
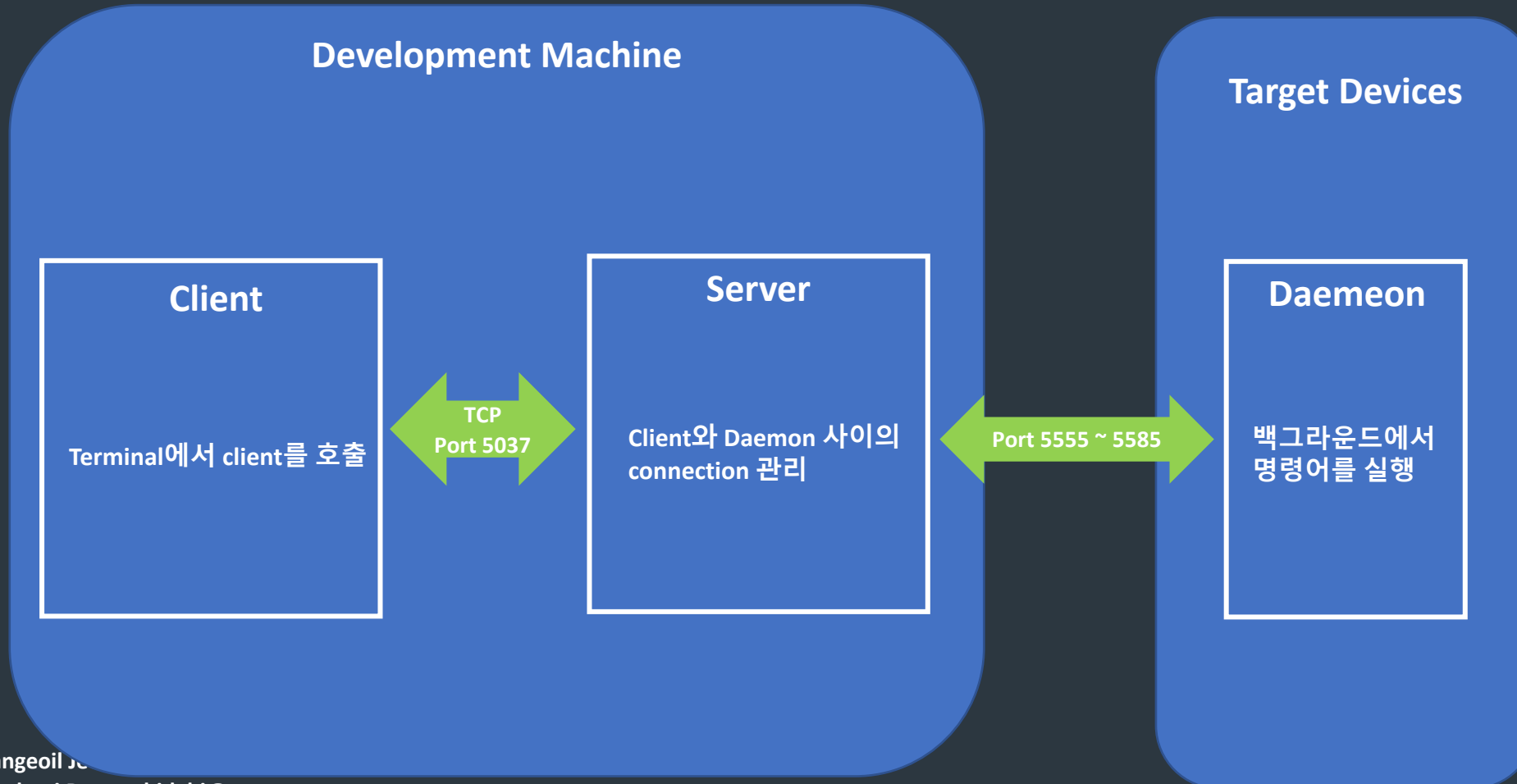


ADB를 이용하여 개발 PC와 디바이스를 무선으로 연결

```
PS C:\Users\asb46\Desktop> cd system
PS C:\Users\asb46\Desktop\system> adb connect 192.168.219.103
cannot connect to 192.168.219.103:5555: No connection could be made because
61)
PS C:\Users\asb46\Desktop\system> adb connect 192.168.219.103:5555
cannot connect to 192.168.219.103:5555: No connection could be made because
61)
PS C:\Users\asb46\Desktop\system> adb connect 192.168.219.103:45393
connected to 192.168.219.103:45393
PS C:\Users\asb46\Desktop\system>
```

# Android Debug Bridge (ADB)

## How ADB works?



# Scrapy 미러링 다운로드 및 실행

## 다운로드 링크

다운로드 링크 접속하여 윈도우 버전 다운로드

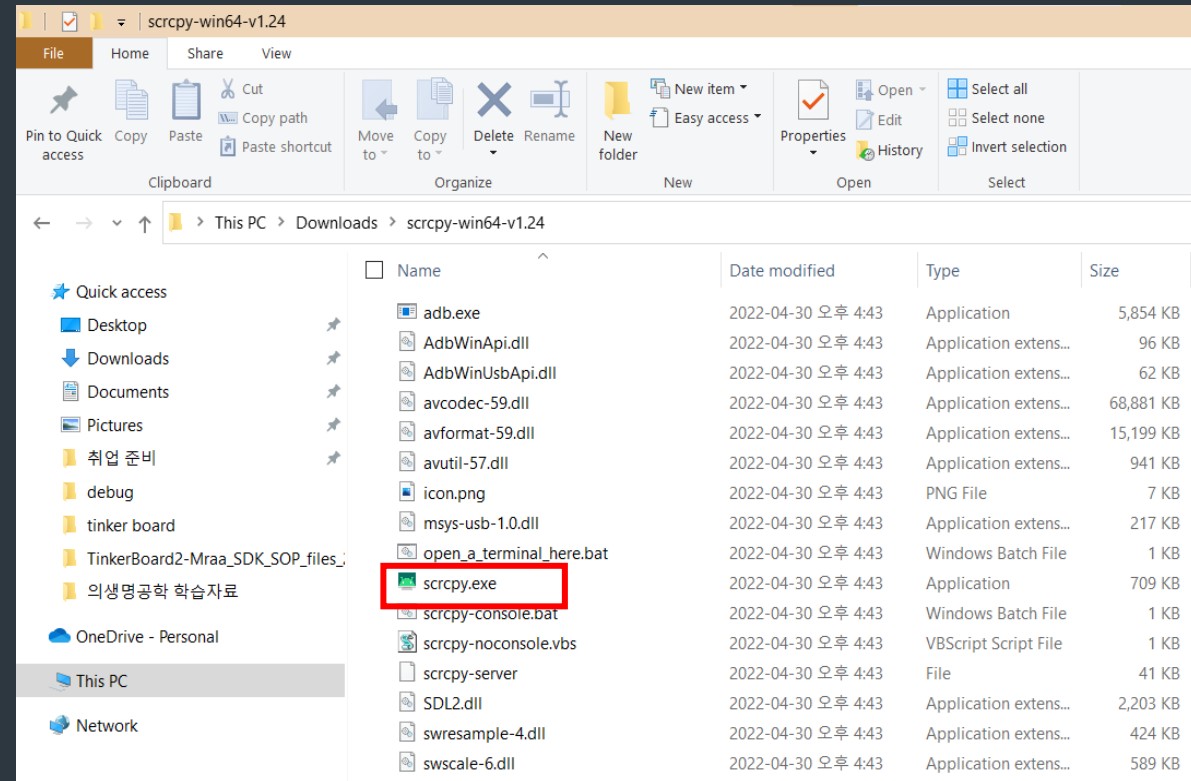
### Get the app

#### Summary

- Linux: `apt install scrapy`
- Windows: [download](#)
- macOS: `brew install scrapy`

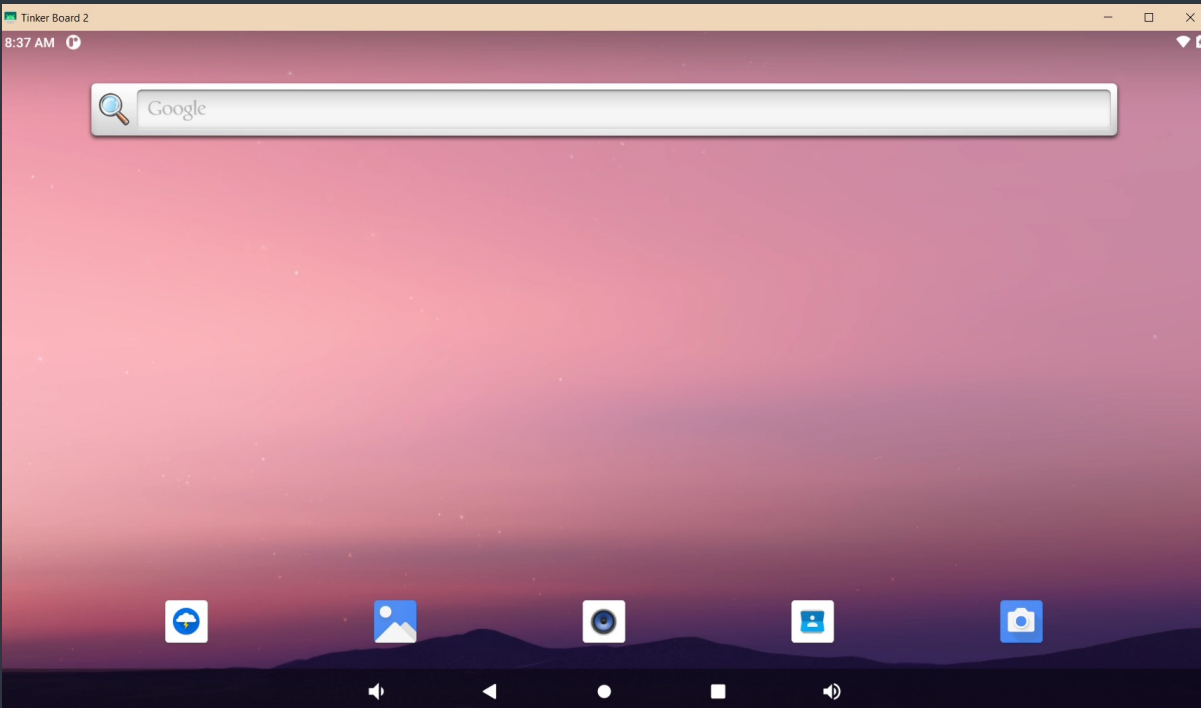
Build from sources: [BUILD \(simplified process\)](#)

## 다운로드 폴더에서 scrapy.exe 실행





# Scrapy 미러링 다운로드 및 실행



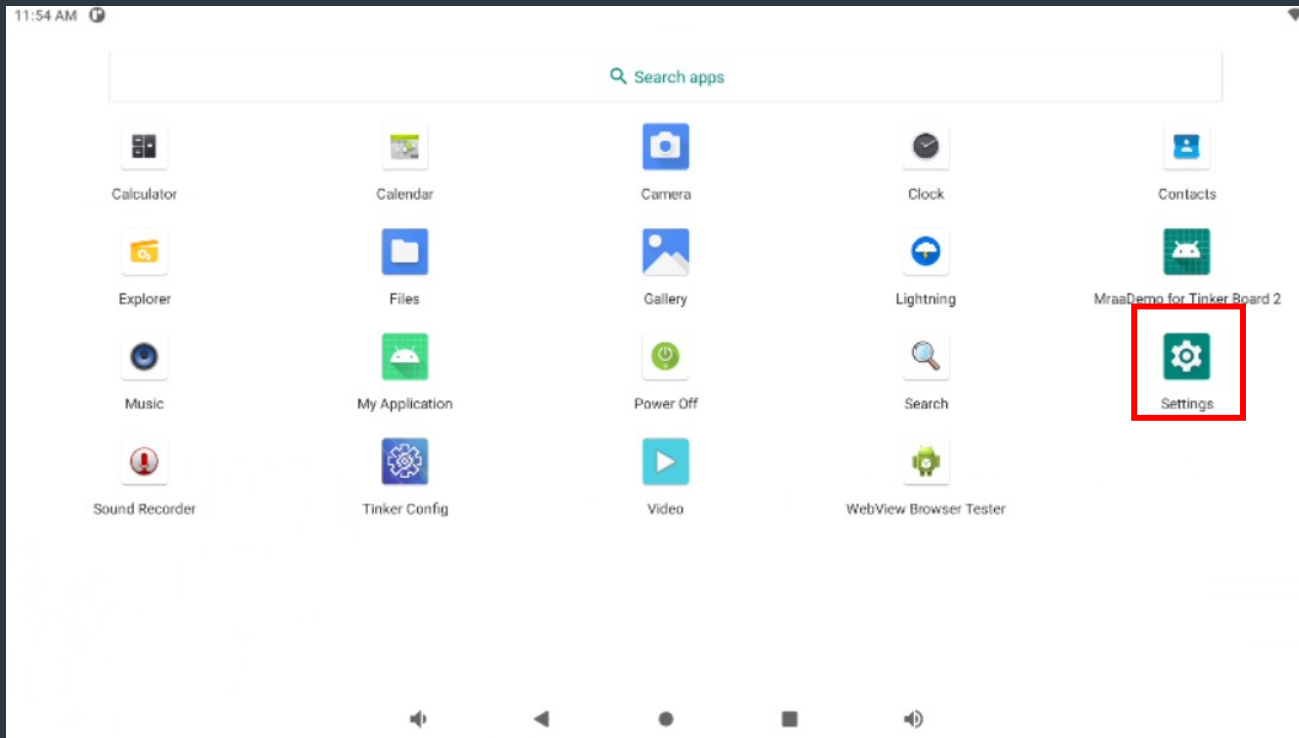
scrapy 실행시, cmd 화면이 켜졌다 바로 꺼지면 USB 포트 연결의심

## 실행안될 시 해결 방법

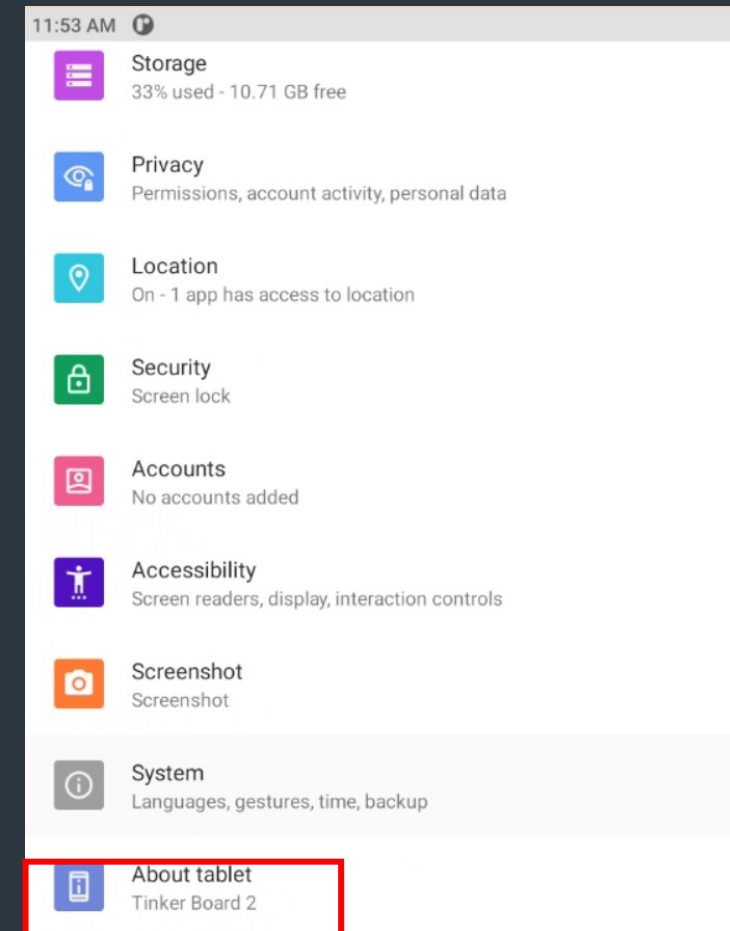
- cmd 창에서 adb devices로 연결 상태 확인
- 팅커보드에서 USB debugging mode 되어 있는 지 확인
- ADB 다운로드 및 환경변수 설정 확인

# Tinker Board Debug Mode 활성화

## 1. settings 클릭

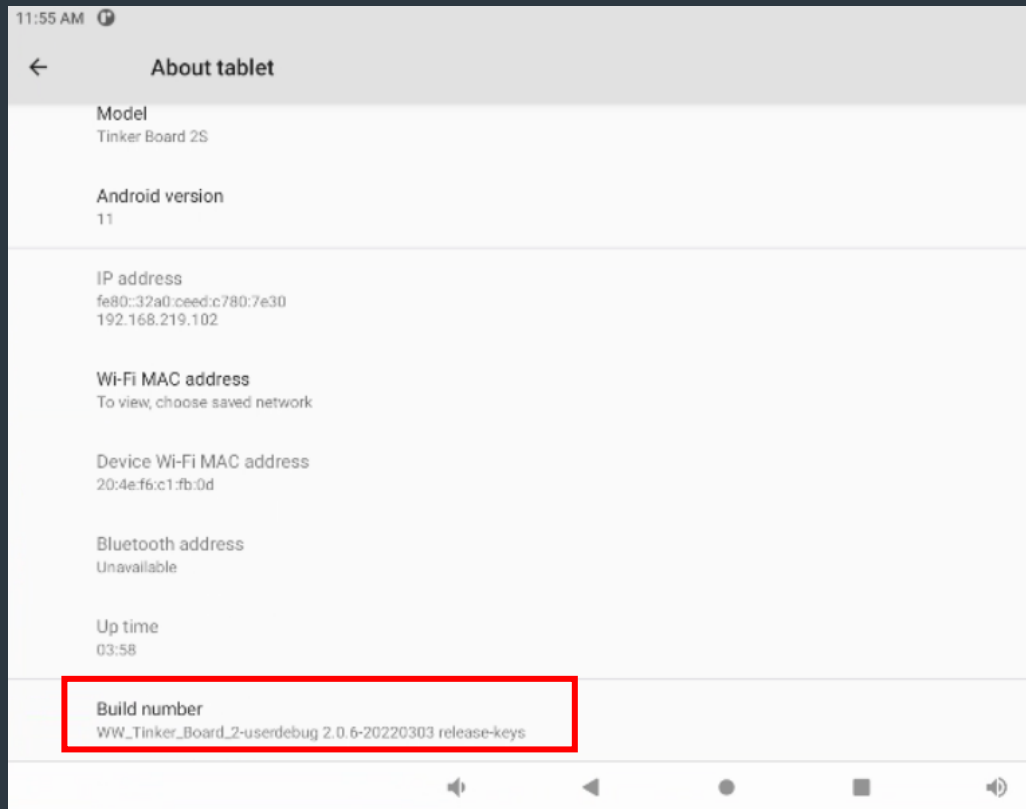


## 2. About\_tablet 클릭

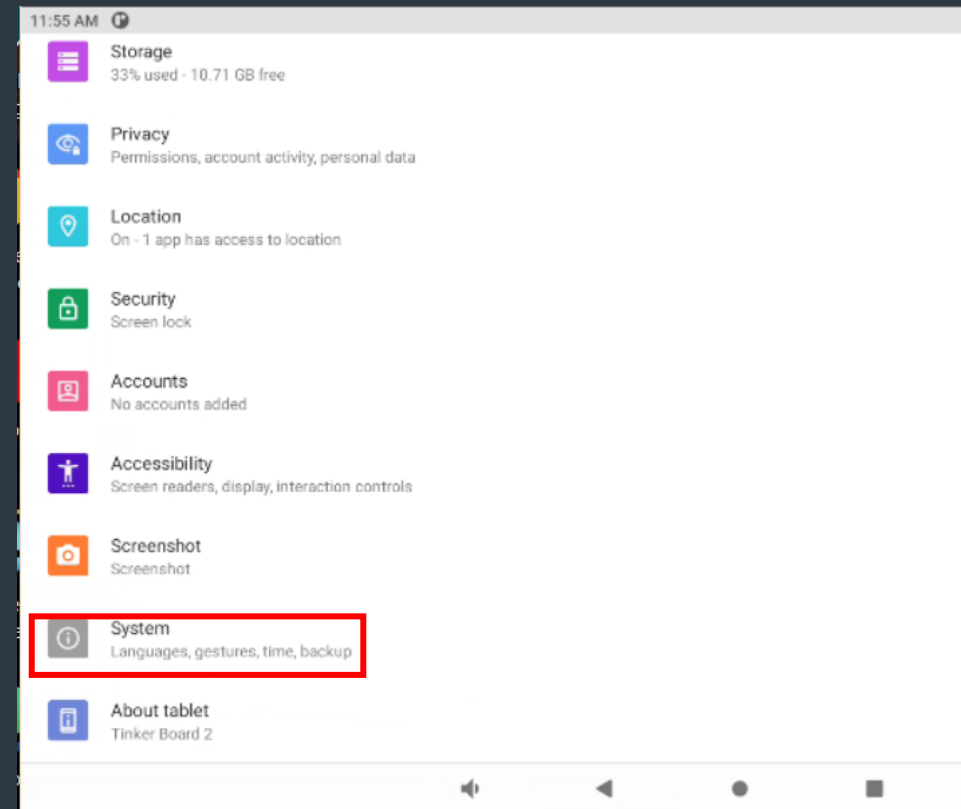


# Tinker Board Debug Mode 활성화

## 3. Build number 10번 연속 클릭해서 Debug mode 활성화

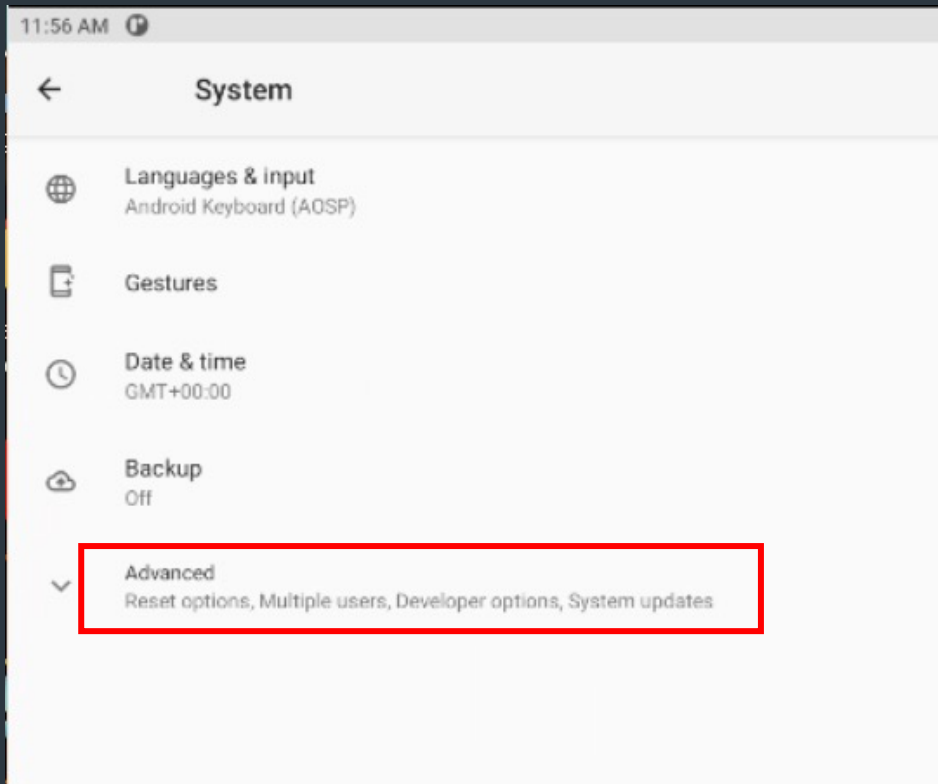


## 4. System 클릭

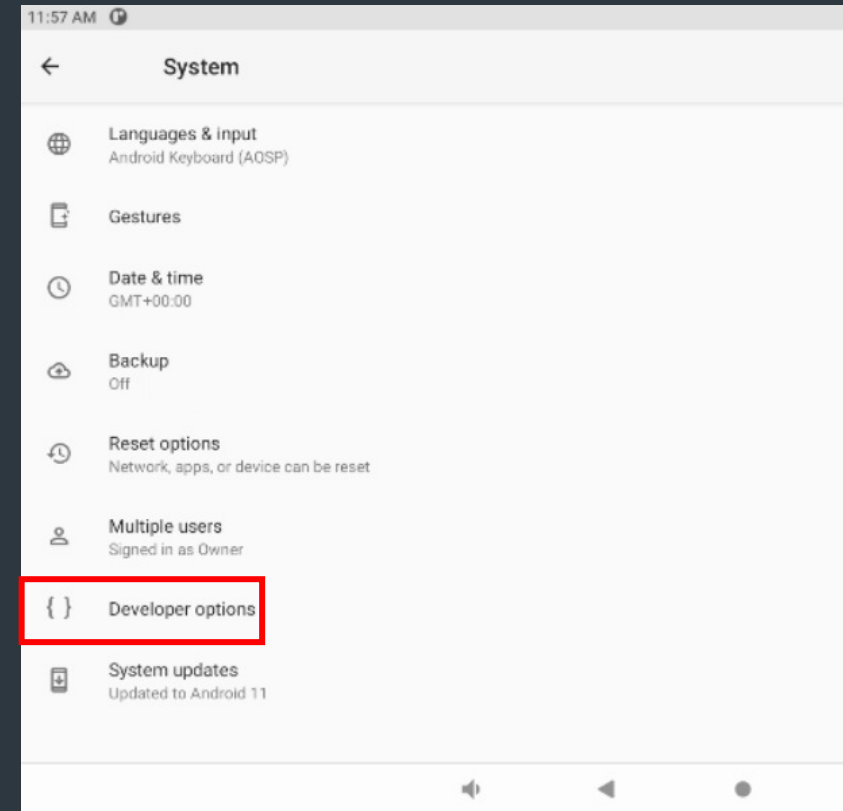


# Tinker Board Debug Mode 활성화

## 5. Advanced 클릭

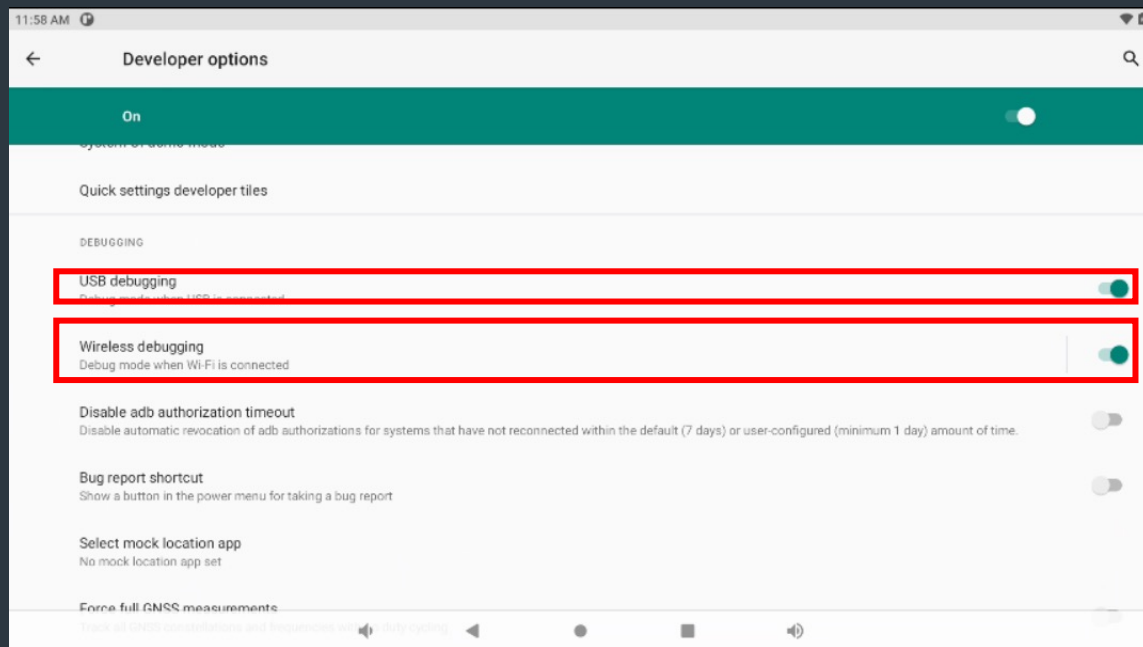


## 6. Developer Option 클릭

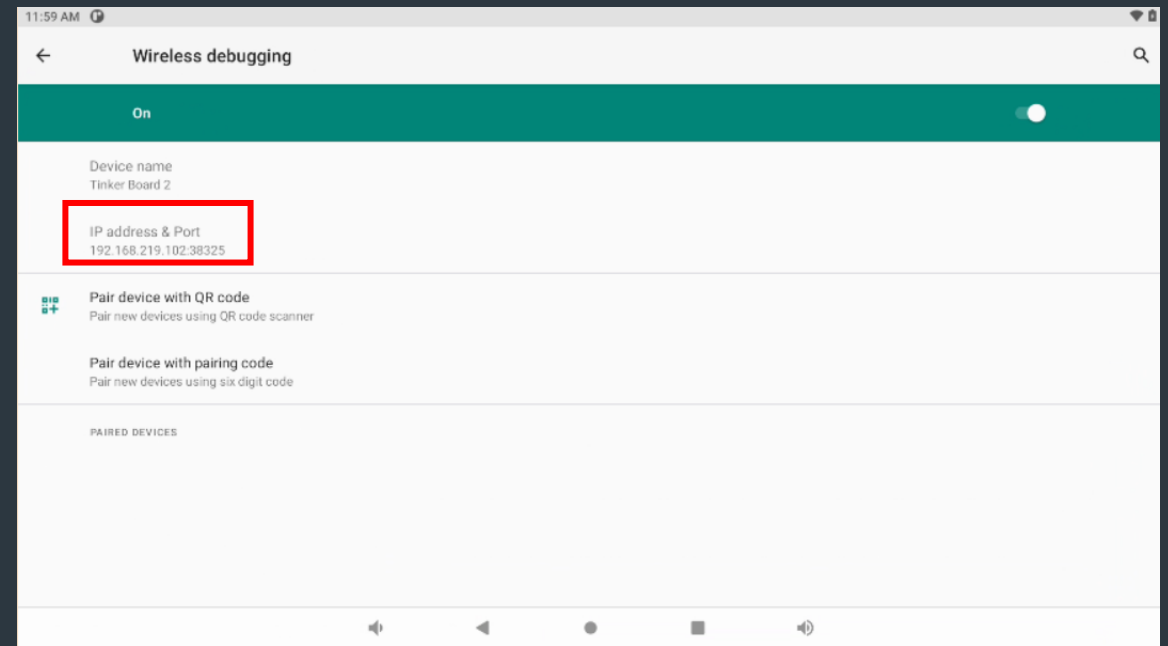


# Tinker Board Debug Mode 활성화 (wireless option)

## 7. USB debugging or Wireless Debugging 활성화

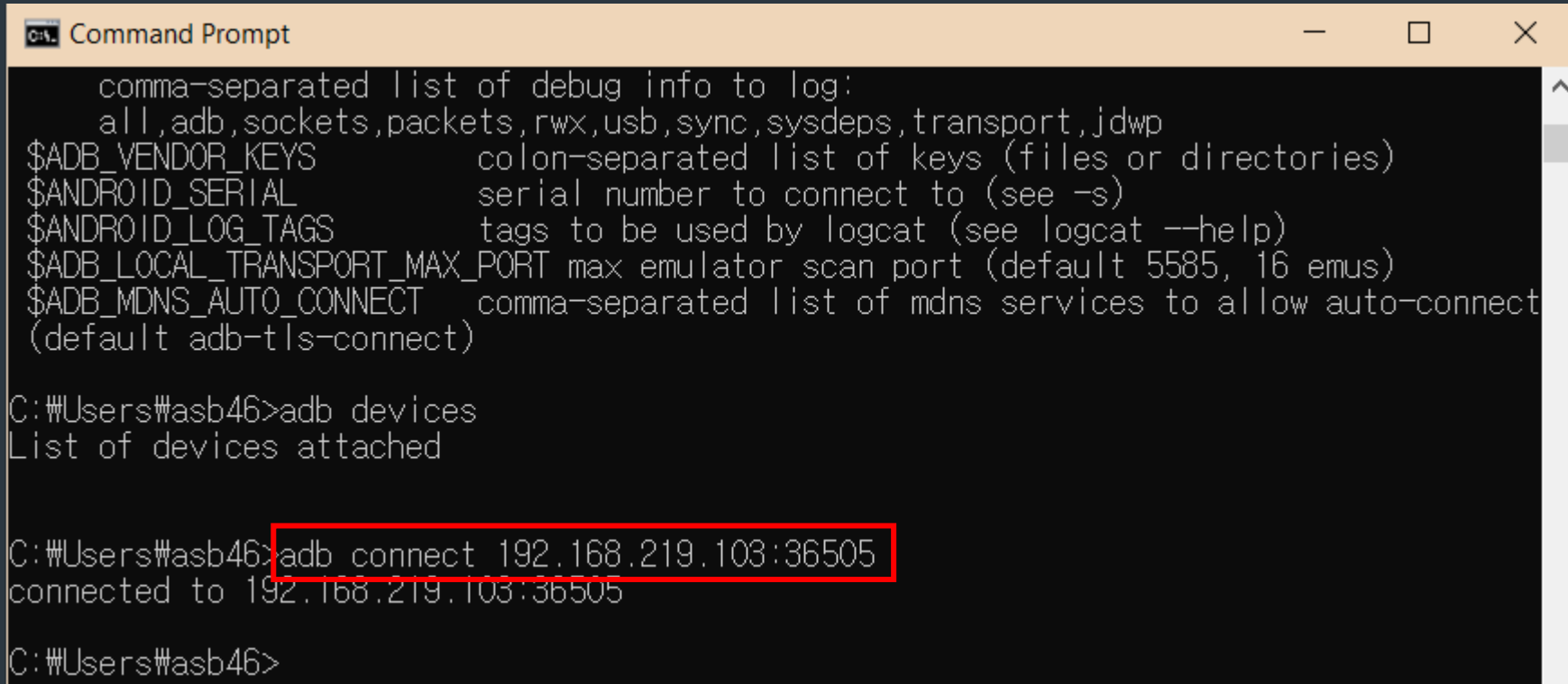


## 8. IP address & Port 확인



# ADB 커맨드를 이용하여 PC와 보드를 연결 (wireless option)

adb connect "ip\_address:port\_number"

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window has a standard Windows title bar with minimize, maximize, and close buttons. The background is black with white text. The text inside the window shows the following: A list of ADB options: comma-separated list of debug info to log: all,adb,sockets,packets,rwx,usb,sync,sysdeps,transport,jdwp; \$ADB\_VENDOR\_KEYS: colon-separated list of keys (files or directories); \$ANDROID\_SERIAL: serial number to connect to (see -s); \$ANDROID\_LOG\_TAGS: tags to be used by logcat (see logcat --help); \$ADB\_LOCAL\_TRANSPORT\_MAX\_PORT: max emulator scan port (default 5585, 16 emus); \$ADB\_MDNS\_AUTO\_CONNECT: comma-separated list of mdns services to allow auto-connect (default adb-tls-connect). Then, the command "adb devices" is entered, and the output "List of devices attached" is shown. Next, the command "adb connect 192.168.219.103:36505" is entered, and the output "connected to 192.168.219.103:36505" is shown. The command prompt ends with "C:\Users\asb46>". The command "adb connect 192.168.219.103:36505" and its output are highlighted with a red rectangular box.

```
Command Prompt
  comma-separated list of debug info to log:
  all,adb,sockets,packets,rwx,usb,sync,sysdeps,transport,jdwp
$ADB_VENDOR_KEYS      colon-separated list of keys (files or directories)
$ANDROID_SERIAL       serial number to connect to (see -s)
$ANDROID_LOG_TAGS     tags to be used by logcat (see logcat --help)
$ADB_LOCAL_TRANSPORT_MAX_PORT max emulator scan port (default 5585, 16 emus)
$ADB_MDNS_AUTO_CONNECT comma-separated list of mdns services to allow auto-connect
(default adb-tls-connect)

C:\Users\asb46>adb devices
List of devices attached

C:\Users\asb46>adb connect 192.168.219.103:36505
connected to 192.168.219.103:36505

C:\Users\asb46>
```

# ADB 커맨드

adb devices	adb server와 연결된 디바이스 목록을 보여줌
adb devices -l	연결된 디바이스의 이름을 보여줌

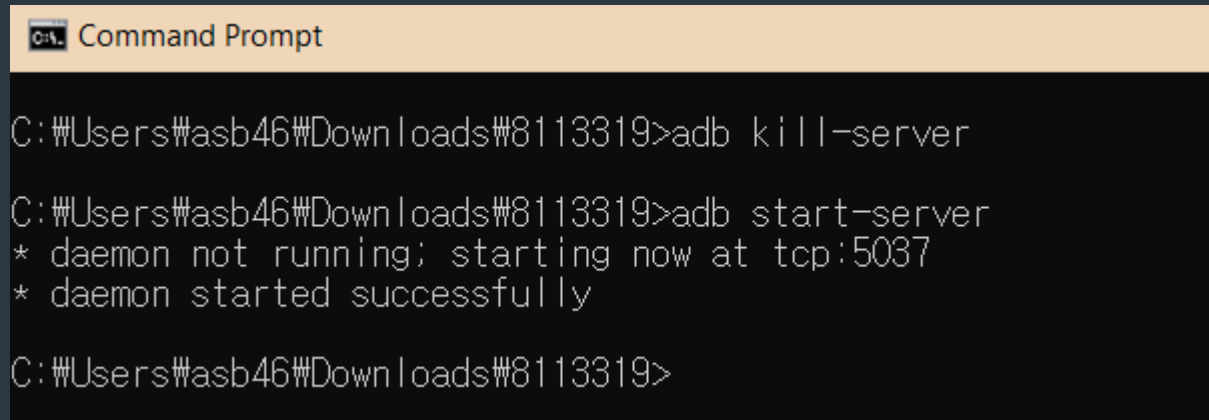


```
Command Prompt
C:\Users\asb46>adb devices
List of devices attached
192.168.219.103:41321    device

C:\Users\asb46>adb devices -l
List of devices attached
192.168.219.103:41321    device product:WW_Tinker_Board model:
WW_Tinker_Board device:WW_Tinker_Board transport_id:1
```

# ADB 커맨드

adb kill-server	adb server 연결을 종료
adb start-server	adb server 연결 시작



```
Command Prompt
C:\Users\asb46\Downloads\8113319>adb kill-server

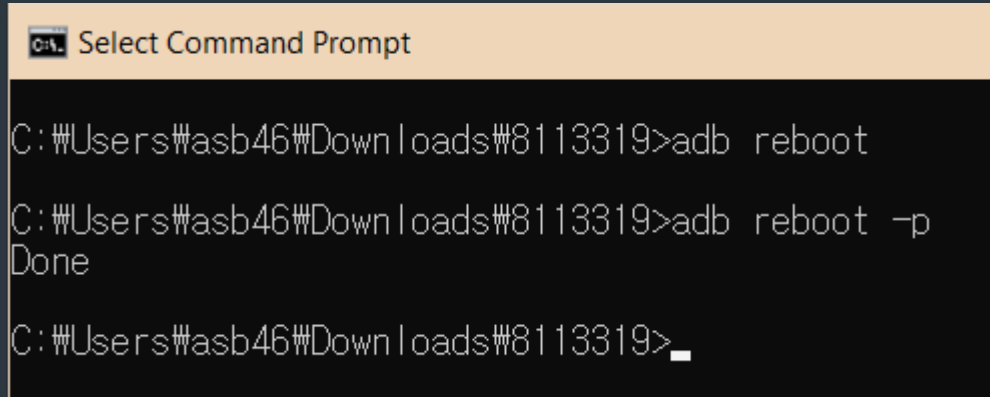
C:\Users\asb46\Downloads\8113319>adb start-server
* daemon not running; starting now at tcp:5037
* daemon started successfully

C:\Users\asb46\Downloads\8113319>
```



# ADB 커맨드

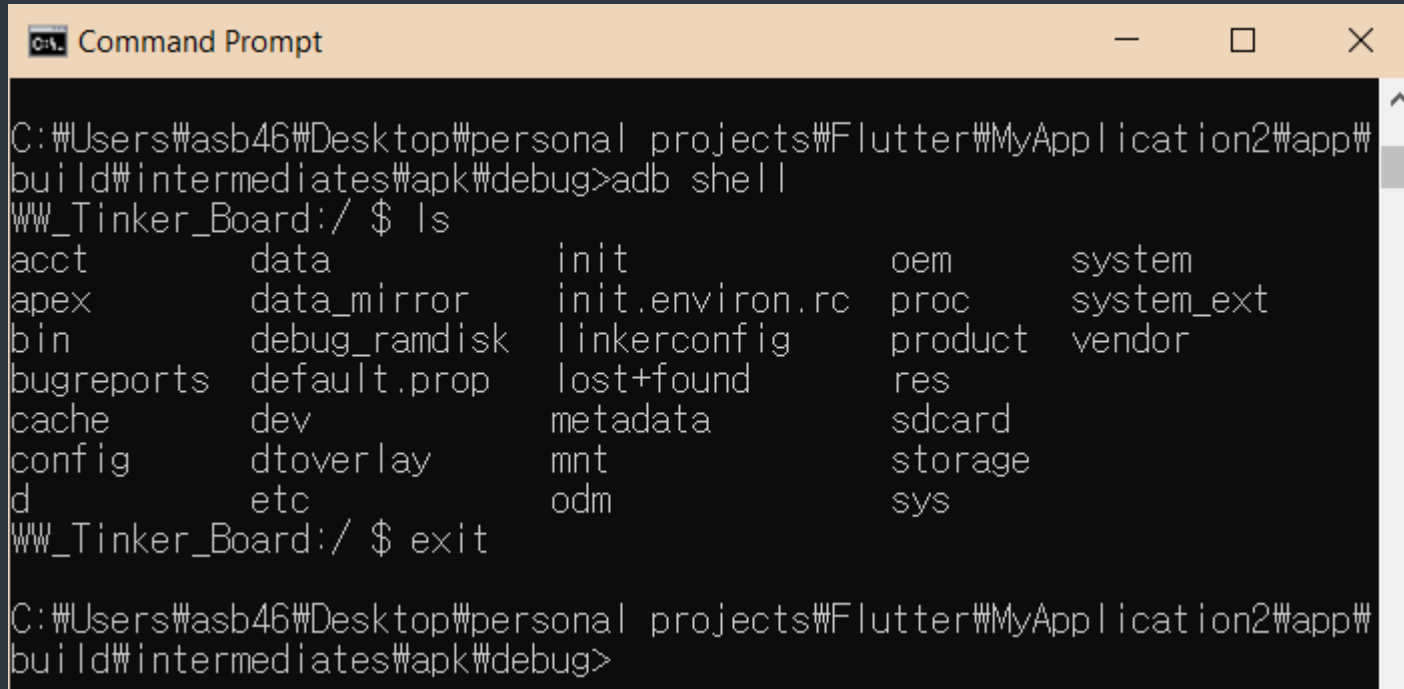
adb reboot	안드로이드 디바이스 reboot
adb reboot -p	안드로이드 디바이스 shutdown



```
C:\Users\asb46\Downloads\8113319>adb reboot  
C:\Users\asb46\Downloads\8113319>adb reboot -p  
Done  
C:\Users\asb46\Downloads\8113319>_
```

# ADB 커맨드

- adb shell                      Android device 상에서 terminal 모드로 접속
- adb shell pm list packages    list all applications installed
- adb shell pm list packages -3   list all the third-party applications



```
C:\Users\asb46\Desktop\personal_projects\Flutter\MyApplication2\app\build\intermediates\apk\debug>adb shell
WW_Tinker_Board:/ $ ls
acct          data          init          oem          system
apex          data_mirror  init.environ.rc  proc         system_ext
bin           debug_ramdisk linkerconfig  product      vendor
bugreports    default.prop  lost+found    res
cache         dev          metadata      sdcard
config        dtoverlay    mnt           storage
d            etc          odm           sys
WW_Tinker_Board:/ $ exit

C:\Users\asb46\Desktop\personal_projects\Flutter\MyApplication2\app\build\intermediates\apk\debug>
```

# ADB 커맨드

adb root	restarts adb with root permissions
adb remount	put “/system” in writable mode (should’ve done before pushing files)

```
Command Prompt - adb shell
C:\Users\asb46\Downloads\8113319>adb pull /dtoverlay/text.txt
/dtoverlay/text.txt: 1 file pulled, 0 skipped.

C:\Users\asb46\Downloads\8113319>adb root
restarting adbd as root

C:\Users\asb46\Downloads\8113319>adb remount
remount succeeded

C:\Users\asb46\Downloads\8113319>adb push text.txt /dtoverlay/
text.txt: 1 file pushed, 0 skipped. 0.0 MB/s (12 bytes in 0.001s)
```

# ADB 커맨드

adb push                  client에서 daemon으로 파일 전송

adb pull                  daemon에서 client으로 파일 전송

```
Command Prompt - adb shell
C:\Users\asb46\Downloads\8113319>adb pull /dtoverlay/text.txt
/dtoverlay/text.txt: 1 file pulled, 0 skipped.

C:\Users\asb46\Downloads\8113319>adb root
restarting adbd as root

C:\Users\asb46\Downloads\8113319>adb remount
remount succeeded

C:\Users\asb46\Downloads\8113319>adb push text.txt /dtoverlay/
text.txt: 1 file pushed, 0 skipped. 0.0 MB/s (12 bytes in 0.001s)
```

# ADB 커맨드

adb install apk\_name            apk 파일을 안드로이드 디바이스에 install  
adb uninstall package\_name    install된 패키지를 uninstall

```
Command Prompt
C:\Users\asb46\Downloads\8113319>adb install Android-MraaDemo_tinkerboard2.apk
Performing Streamed Install
Success

C:\Users\asb46\Downloads\8113319>adb shell pm list packages -3
package:com.tinkerboard.mraademo
package:com.example.myapplication

C:\Users\asb46\Downloads\8113319>adb uninstall com.tinkerboard.mraademo
Success

C:\Users\asb46\Downloads\8113319>_
```

# ADB 활용 - Google Store Install

# ADB 커맨드 – Tinker edge R의 Google playstore 설치

## <구글 apk 파일 설치 방법>

1. 카카오톡으로 받은 파일을 압축을 풀어주세요.
2. 검색창에 CMD를 쳐주세요.
3. adb devices 를 입력하여 연결이 잘 되어있는지 확인을 합니다.
4. adb root를 입력하여 android 에 root 권한을 줍니다.
5. Adb remount를 입력하여 system partition에 쓰기 권한을 줍니다.
6. adb push를 이용하여 apk 파일을 /system/priv-app에 넣어줍니다.
7. adb shell을 이용하여 카카오톡에서 받은 파일이 잘 설치 되어있는지 확인을 합니다.
8. cd /system/priv-app
9. ls
10. 다운된것을 확인이 되면 exit를 입력하여 Shell에서 빠져나옵니다.

## <구글 apk 파일 설치 예시>

```
C:\Users\Deil>adb devices
List of devices attached
DXSV3AJ5LM    device

C:\Users\Deil>adb root
restarting adbd as root

C:\Users\Deil>adb remount
remount succeeded

C:\Users\Deil>cd C:\Users\Deil\Desktop\강의 자료\google_store

C:\Users\Deil\Desktop\강의 자료\google_store>adb push "com.android.vending_25.9.19-21_0_PR_380694501-82591910_minAPI21(ar
m64-v8a,armeabi-v7a,x86,x86_64)(nodpi)_apkmirror.com" /system/priv-app
adb: error: cannot stat 'com.android.vending_25.9.19-21_0_PR_380694501-82591910_minAPI21(ar
m64-v8a,armeabi-v7a,x86,x86_64)(nodpi)_apkmirror.com': No such file or directory

C:\Users\Deil\Desktop\강의 자료\google_store>adb push "com.android.vending_25.9.19-21_0_PR_380694501-82591910_minAPI21(ar
m64-v8a,armeabi-v7a,x86,x86_64)(nodpi)_apkmirror.com.apk" /system/priv-app
com.android.vending_25.9.19-21_0_PR_380694501-82591910_min...le pushed, 0 skipped. 27.5 MB/s (45835474 bytes in 1.588s)
```

6. **adb push** "com.android.vending\_12.4.14-all\_0\_PR\_219234122-81241400\_minAPI16(armeabi,armeabi-v7a,mips,mips64,x86,x86\_64)(nodpi)\_apkmirror.com.apk" **/system/priv-app**

# ADB 커맨드 – Tinker edge R의 Google playstore 설치

## <라이브러리 설치 방법>

1. 압축파일 안의 lib 폴더에 들어갑니다
2. 검색창에 CMD를 쳐주세요.
3. Adb devices 를 입력하여 연결이 잘 되어있는지 확인을 합니다.
4. Adb root를 입력하여 android 에 root 권한을 줍니다.
5. Adb remount를 입력하여 system partitio에 쓰기 권한을 줍니다.
6. Adb push를 이용하여 apk 파일을 /system/lib/에 넣어줍니다.
7. adb shell을 이용하여 카카오톡에서 받은 파일이 잘 설치 되어있는지 확인을 합니다.
8. cd /system/lib/
9. ls
10. 다운된것을 확인이 되면 exit를 입력하여 Shell에서 빠져나옵니다.

## <구글 lib 파일 설치 예시>

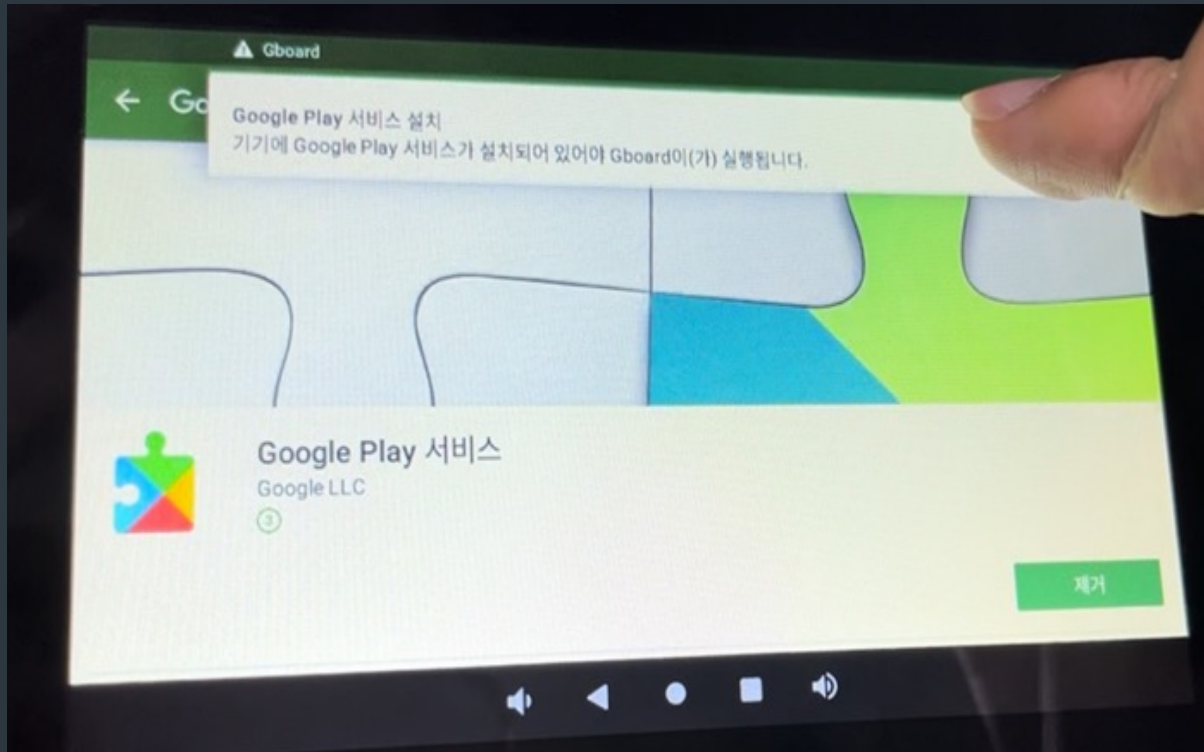
1. X
2. X
3. x
4. x
5. X
6. **adb push** "libAppDataSearch.so" **/system/lib/**

11. **adb reboot**를 입력하여 재부팅을 해줍니다.



# ADB 커맨드 – Tinker edge R의 Google playstore 설치

<Google playstore 연결 성공시>



팅커보드 엣지알에 play store 가 설치된것을 확인 하였습니다.